

# MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

## TRABAJO FIN DE MÁSTER

### **DESARROLLO DE CONTROLADORES AVANZADOS PARA SISTEMAS MULTIVARIABLE EN EL ENTORNO LABVIEW**

**Alumno/Alumna**

*Dorronsoro, López de  
Maturana, Miguel*

**Director/Directora**

*Artaza, Fano, Fernando*

**Departamento**

*Sistemas y automática*

**Curso académico**

*2017/2018*

*Bilbao, 21 de septiembre de 2018*



## **RESUMEN TRILINGÜE**

### **Resumen:**

*Este trabajo de fin de máster (TFM) consiste en el desarrollo de distintos algoritmos de control mediante la aplicación LabVIEW, además de una tarjeta de interfaz NI myRIO de National Instruments (NI) para el control en tiempo real de tres sistemas distintos: un sistema carro, un sistema péndulo Gantry (Grúa) y un sistema péndulo invertido. Los algoritmos están basados en controladores PID y de realimentación de estados, por asignación de polos y LQR. Así mismo, se crea una interfaz de usuario que permite interactuar con los sistemas y realizar pruebas con los algoritmos de control mencionados eligiendo los parámetros correspondientes, así como hacer simulaciones sobre sus modelos teóricos.*

---

### **Palabras clave:**

*TFM, LabVIEW, NI myRIO, HMI, control, tiempo real, péndulo, invertido, Gantry, LQR, PID.*

---

### **Laburpena:**

*Master amaierako lan (MAL) hau LabVIEW aplikazioaren bidez, baita National Instruments-ko (NI) NI myRIO interfazeko txartela baten bidez hiru sistema ezberdinerako erreal denborako kontrolerako, hainbat kontrol algoritmo eraikitzean datza: gurdi sistema bat, Gantry pendulu (Garabi) sistema bat eta pendulu alderantzikatu sistema bat. Algoritmoak PID eta estatuen berrelikadura oinarrituta daude, poloen esleipenaren eta LQR-ren bitartez. Halaber, erabiltzaile interfaze bat egin da, sistemekin elkarri eragiteko eta dagokion parametroak aukeratuz lehen aipatutako kontrol algoritmoekin probak egiteko aukera ematen duena, baita modelo teorikoetan simulazioak egin ere.*

---

### **Hitz gakoak:**

*MAL, LabVIEW, NI myRIO, HMI, kontrol, denbora erreala, pendulu, alderantzikatu, Gantry, LQR, PID.*

---

### **Abstract:**

*This end of master thesis (EMT) consists in the development of different control algorithms using LabVIEW software, as well as an NI myRIO interface board from National Instruments (NI) for the real time control on three different systems: a cart system, a Gantry pendulum (Crane) system and an inverted pendulum system. The algorithms are based on PID and state feedback, by means of poles placement and LQR controllers. Furthermore, a user interface which allows to interact with the systems and, selecting the corresponding parameters, to perform tests with the previously mentioned control algorithms, as well as perform simulations on the theoretical models.*

---

### **Keywords:**

*EMT, LabVIEW, NI myRIO, HMI, control, real time, pendulum, inverted, Gantry, LQR, PID.*

---

## ÍNDICE

1. INTRODUCCIÓN.....	1
2. CONTEXTO.....	2
3. OBJETIVOS Y ALCANCE.....	4
4. BENEFICIOS DEL PROYECTO .....	5
5. ANÁLISIS DE ALTERNATIVAS .....	7
5.1. OBTENCIÓN DEL MODELO.....	7
5.2. TIPOS DE CONTROLADOR INCLUIDOS.....	8
5.2.1. CONTROLADOR PID .....	8
5.2.2. CONTROLADOR POR REALIMENTACIÓN DEL ESTADO.....	9
5.2.3. CONTROL PREDICTIVO.....	11
5.2.4. REDES NEURONALES.....	11
5.3. TIEMPO DE MUESTREO .....	12
5.4. CÁLCULO DE LAS DERIVADAS .....	13
5.5. LECTURA Y ESCRITURA.....	18
5.6. ALGORITMOS DE AUTOERECCIÓN .....	20
6. METODOLOGÍA.....	21
6.1. DESCRIPCIÓN DEL SISTEMA .....	24
6.1.1. MAQUETA.....	24
6.1.2. UNIVERSAL POWER MODULE (UPM).....	25
6.1.3. TARJETA NI MYRIO.....	26
6.1.4. ORDENADOR .....	27
6.1.5. ESQUEMA DE INTERCONEXIÓN.....	27
6.2. CONEXIONES ENTRE TARJETA Y MAQUETA.....	28
6.2.1. ENCODERS .....	28
6.2.2. MOTOR .....	28
6.2.3. DETALLES DE LAS CONEXIONES .....	29
6.2.4. PRUEBA DE LAS CONEXIONES.....	31
6.3. VARIABLES PRINCIPALES DEL SISTEMA Y RANGOS.....	33
6.4. MODELOS TEÓRICOS.....	34
6.4.1. MODELO DEL MOTOR Y EL CARRO.....	34
6.4.2. MODELO DEL PÉNDULO GANTRY.....	36
6.4.3. MODELO DEL PÉNDULO INVERTIDO .....	41
6.4.4. MODIFICACIONES SOBRE EL MODELO MATEMÁTICO .....	42
6.5. DISEÑO DE CONTROLADORES .....	47
6.5.1. SISTEMA CARRO .....	47

6.5.2. SISTEMA PENDULO GANTRY .....	56
6.5.3. SISTEMA PENDULO INVERTIDO.....	76
6.5.4. CONCLUSIONES .....	110
6.6. DESARROLLO DE LA APLICACIÓN DE USUARIO .....	111
6.6.1. PROGRAMA DE ORDENADOR Y HMI .....	113
6.6.2. PROGRAMA DE CONTROL EN TIEMPO REAL .....	123
6.6.3. LIBRERÍA EN RED .....	132
6.6.4. PROCESO DE DESARROLLO.....	136
6.6.5. PRODUCTO FINAL .....	137
7. PLANIFICACIÓN .....	150
8. DESCARGO DE GASTOS.....	156
9. CONCLUSIONES FINALES .....	159
10. BIBLIOGRAFÍA.....	161
ANEXOS.....	1
ANEXO I: MODELOS TEÓRICOS .....	1
1. Modelo del motor y el carro.....	2
2. Modelo del péndulo Gantry .....	5
3. Modelo del péndulo invertido .....	11
ANEXO II: PARÁMETROS DEL SISTEMA .....	1
ANEXO III: GANANCIAS ÓPTIMAS DE LOS CONTROLADORES.....	1
1. Sistema carro.....	2
2. Sistema péndulo Gantry .....	2
3. Sistema péndulo invertido.....	2
4. Autoerección.....	3
ANEXO IV: MANUAL DE USUARIO DE LA APLICACIÓN .....	1
1. Introducción .....	4
2. Descripción .....	5
3. Conexión de la tarjeta .....	9
4. Carga de la aplicación en tiempo real .....	11
5. Instalación de la aplicación para ordenador .....	14
6. Preparación de la maqueta IP02 de Quanser .....	15
7. Uso de la aplicación para ordenador.....	18
ANEXO V: CÓDIGO DE LA APLICACIÓN .....	1
1. HMI.vi .....	2
2. Carro Prep.vi.....	3
3. Pendulo Gantry Prep.vi.....	7
4. Pendulo Invertido Prep.vi.....	11

5. Carro PID Monitorizacion.vi .....	16
6. Carro SS Monitorizacion.vi.....	18
7. Pendulo Gantry PID Monitorizacion.vi.....	20
8. Pendulo Gantry SS Monitorizacion.vi.....	22
9. Pendulo Invertido PID Monitorizacion.vi.....	24
10. Pendulo Invertido SS Monitorizacion.vi.....	26
11. Carro PID Analisis.vi .....	28
12. Carro SS Analisis.vi .....	31
13. Pendulo Gantry PID Analisis.vi .....	34
14. Pendulo Gantry SS Analisis.vi .....	37
15. Pendulo Invertido PID Analisis.vi .....	40
16. Pendulo Invertido SS Analisis.vi.....	44
17. Inicio.vi.....	48
18. Carro PID RT.vi .....	50
19. Carro SS RT.vi.....	53
20. Pendulo Gantry PID RT.vi.....	56
21. Pendulo Gantry SS RT.vi.....	59
22. Pendulo Invertido PID RT.vi.....	62
23. Pendulo Invertido SS RT.vi.....	68

## LISTA DE FIGURAS

<i>Figura 1. Logo de la Escuela de Ingeniería de Bilbao .....</i>	<i>2</i>
<i>Figura 2. Maqueta IP02 de Quanser .....</i>	<i>2</i>
<i>Figura 3. Logo de National Instruments y su software LabVIEW.....</i>	<i>3</i>
<i>Figura 4. Puente grúa .....</i>	<i>5</i>
<i>Figura 5. Ejemplos de aplicación del péndulo invertido a sistemas reales .....</i>	<i>6</i>
<i>Figura 6. Posibles modos de obtención del modelo .....</i>	<i>7</i>
<i>Figura 7. Modo de obtención del modelo seleccionado.....</i>	<i>8</i>
<i>Figura 8. Esquema del controlador PID .....</i>	<i>9</i>
<i>Figura 9. Esquema de control por realimentación del estado.....</i>	<i>10</i>
<i>Figura 10. Topología de una red neuronal .....</i>	<i>12</i>
<i>Figura 11. Bloque en LabVIEW para comprobar la observabilidad .....</i>	<i>13</i>
<i>Figura 12. Esquema del observador de orden completo continuo.....</i>	<i>14</i>
<i>Figura 13. Esquema del observador de orden completo discreto.....</i>	<i>15</i>
<i>Figura 14. Esquema del observador de orden reducido discreto .....</i>	<i>16</i>
<i>Figura 15. Diagrama de Bode del filtro derivativo.....</i>	<i>17</i>
<i>Figura 16. Módulo preprogramado de encoder para NI myRIO.....</i>	<i>18</i>
<i>Figura 17. Código del módulo preprogramado de encoder para NI myRIO.....</i>	<i>18</i>
<i>Figura 18. Bloques elementales de encoder para NI myRIO.....</i>	<i>19</i>
<i>Figura 19. Módulo preprogramado de salida analógica para NI myRIO .....</i>	<i>19</i>
<i>Figura 20. Código del módulo preprogramado de salida analógica para NI myRIO ....</i>	<i>19</i>
<i>Figura 21. Bloques elementales de salida analógica para NI myRIO .....</i>	<i>20</i>
<i>Figura 22. Work Breakdown Structure del proyecto .....</i>	<i>22</i>
<i>Figura 23. Proceso para “Análisis de los sistemas”, “Obtención de modelos” y “Diseño de controladores” .....</i>	<i>23</i>
<i>Figura 24. Maqueta IP02 de Quanser .....</i>	<i>24</i>
<i>Figura 25. Componentes de la maqueta IP02 de Quanser .....</i>	<i>25</i>
<i>Figura 26. Módulo UPM.....</i>	<i>26</i>
<i>Figura 27. Tarjeta NI myRIO de National Instruments.....</i>	<i>26</i>
<i>Figura 28. Esquema de interconexión de los sistemas .....</i>	<i>27</i>
<i>Figura 29. Cableado de los encoders .....</i>	<i>28</i>
<i>Figura 30. Conexión en los pines de NI myRIO .....</i>	<i>30</i>
<i>Figura 31. Tarjeta NI myRIO con las conexiones realizadas .....</i>	<i>31</i>
<i>Figura 32. Front Panel del VI de prueba de las conexiones .....</i>	<i>32</i>
<i>Figura 33. Código del VI de prueba de las conexiones .....</i>	<i>32</i>
<i>Figura 34. Esquema del modelo del motor y el carro .....</i>	<i>34</i>
<i>Figura 35. Esquema del modelo del péndulo Gantry .....</i>	<i>36</i>
<i>Figura 36. Esquema del modelo del péndulo invertido.....</i>	<i>41</i>
<i>Figura 37. Tensión aplicada al carro.....</i>	<i>43</i>
<i>Figura 38. Desplazamiento del carro .....</i>	<i>43</i>
<i>Figura 39. Detalle del desplazamiento del carro .....</i>	<i>44</i>
<i>Figura 40. Voltaje aplicado al carro.....</i>	<i>44</i>
<i>Figura 41. Desplazamiento del carro .....</i>	<i>45</i>
<i>Figura 42. Detalle del desplazamiento del carro .....</i>	<i>45</i>
<i>Figura 43. Estructura del controlador PID para el carro .....</i>	<i>47</i>
<i>Figura 44. Desplazamiento del carro en la prueba.....</i>	<i>48</i>
<i>Figura 45. Tensión aplicada al carro en la prueba .....</i>	<i>48</i>
<i>Figura 46. Estructura de un controlador de realimentación del estado con integración</i>	<i>49</i>
<i>Figura 47. Integración del error implementada en LabVIEW.....</i>	<i>50</i>

<i>Figura 48. Estructura del controlador de realimentación del estado para el carro .....</i>	<i>50</i>
<i>Figura 49. Desplazamiento del carro en la prueba.....</i>	<i>52</i>
<i>Figura 50. Tensión aplicada al carro en la prueba .....</i>	<i>52</i>
<i>Figura 51. Velocidad del carro en la prueba.....</i>	<i>53</i>
<i>Figura 52. Desplazamiento del carro en la prueba.....</i>	<i>54</i>
<i>Figura 53. Tensión aplicada al carro en la prueba .....</i>	<i>54</i>
<i>Figura 54. Velocidad del carro en la prueba.....</i>	<i>55</i>
<i>Figura 55. Esquema de PIDs conectados en cascada para el péndulo Gantry .....</i>	<i>56</i>
<i>Figura 56. Estructura del controlador PID para el péndulo Gantry .....</i>	<i>57</i>
<i>Figura 57. Salida del sistema en la prueba .....</i>	<i>58</i>
<i>Figura 58. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>58</i>
<i>Figura 59. Desplazamiento del carro en la prueba.....</i>	<i>59</i>
<i>Figura 60. Ángulo del péndulo Gantry en la prueba .....</i>	<i>59</i>
<i>Figura 61. Salida del sistema en la prueba .....</i>	<i>60</i>
<i>Figura 62. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>60</i>
<i>Figura 63. Desplazamiento del carro en la prueba.....</i>	<i>61</i>
<i>Figura 64. Ángulo del péndulo Gantry en la prueba .....</i>	<i>61</i>
<i>Figura 65. Estructura del control por realimentación del estado para el péndulo Gantry .....</i>	<i>62</i>
<i>Figura 66. Salida del sistema en la prueba .....</i>	<i>63</i>
<i>Figura 67. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>63</i>
<i>Figura 68. Desplazamiento del carro en la prueba.....</i>	<i>64</i>
<i>Figura 69. Ángulo del péndulo Gantry en la prueba .....</i>	<i>64</i>
<i>Figura 70. Velocidad del carro en la prueba.....</i>	<i>65</i>
<i>Figura 71. Velocidad angular del péndulo Gantry en la prueba.....</i>	<i>65</i>
<i>Figura 72. Salida del sistema en la prueba .....</i>	<i>66</i>
<i>Figura 73. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>66</i>
<i>Figura 74. Desplazamiento del carro en la prueba.....</i>	<i>67</i>
<i>Figura 75. Velocidad del carro en la prueba.....</i>	<i>67</i>
<i>Figura 76. Ángulo del péndulo Gantry en la prueba .....</i>	<i>68</i>
<i>Figura 77. Velocidad angular del péndulo Gantry en la prueba.....</i>	<i>68</i>
<i>Figura 78. Salida del sistema en la prueba .....</i>	<i>69</i>
<i>Figura 79. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>70</i>
<i>Figura 80. Desplazamiento del carro en la prueba.....</i>	<i>70</i>
<i>Figura 81. Velocidad del carro en la prueba.....</i>	<i>71</i>
<i>Figura 82. Ángulo del péndulo Gantry en la prueba.....</i>	<i>71</i>
<i>Figura 83. Velocidad angular del péndulo Gantry en la prueba.....</i>	<i>72</i>
<i>Figura 84. Salida del sistema en la prueba .....</i>	<i>73</i>
<i>Figura 85. Tensión aplicada al péndulo Gantry en la prueba .....</i>	<i>73</i>
<i>Figura 86. Desplazamiento del carro en la prueba.....</i>	<i>74</i>
<i>Figura 87. Velocidad del carro en la prueba.....</i>	<i>74</i>
<i>Figura 88. Ángulo del péndulo Gantry en la prueba .....</i>	<i>75</i>
<i>Figura 89. Velocidad angular del péndulo Gantry en la prueba.....</i>	<i>75</i>
<i>Figura 90. Estructura del controlador PID para el péndulo invertido.....</i>	<i>77</i>
<i>Figura 91. Desplazamiento del carro en la prueba.....</i>	<i>78</i>
<i>Figura 92. Ángulo del péndulo invertido en la prueba .....</i>	<i>78</i>
<i>Figura 93. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>79</i>
<i>Figura 94. Desplazamiento del carro en la prueba.....</i>	<i>79</i>
<i>Figura 95. Ángulo del péndulo invertido en la prueba .....</i>	<i>80</i>
<i>Figura 96. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>80</i>

<i>Figura 97. Estructura del controlador por realimentación del estado para el péndulo invertido.....</i>	<i>81</i>
<i>Figura 98. Desplazamiento del carro en la prueba.....</i>	<i>82</i>
<i>Figura 99. Ángulo del péndulo invertido en la prueba.....</i>	<i>82</i>
<i>Figura 100. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>83</i>
<i>Figura 101. Velocidad del carro en la prueba.....</i>	<i>83</i>
<i>Figura 102. Velocidad angular del péndulo Gantry en la prueba.....</i>	<i>84</i>
<i>Figura 103. Desplazamiento del carro en la prueba.....</i>	<i>84</i>
<i>Figura 104. Ángulo del péndulo invertido en la prueba.....</i>	<i>85</i>
<i>Figura 105. Detalle del ángulo del péndulo invertido en la prueba.....</i>	<i>85</i>
<i>Figura 106. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>86</i>
<i>Figura 107. Velocidad del carro en la prueba.....</i>	<i>86</i>
<i>Figura 108. Velocidad angular del péndulo invertido en la prueba.....</i>	<i>87</i>
<i>Figura 109. Desplazamiento del carro en la prueba.....</i>	<i>88</i>
<i>Figura 110. Ángulo del péndulo invertido en la prueba.....</i>	<i>88</i>
<i>Figura 111. Detalle del ángulo del péndulo invertido en la prueba.....</i>	<i>89</i>
<i>Figura 112. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>89</i>
<i>Figura 113. Velocidad del carro en la prueba.....</i>	<i>90</i>
<i>Figura 114. Velocidad angular del péndulo invertido en la prueba.....</i>	<i>90</i>
<i>Figura 115. Desplazamiento del carro en la prueba.....</i>	<i>91</i>
<i>Figura 116. Ángulo del péndulo invertido en la prueba.....</i>	<i>91</i>
<i>Figura 117. Detalle del ángulo del péndulo invertido en la prueba.....</i>	<i>92</i>
<i>Figura 118. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>92</i>
<i>Figura 119. Velocidad del carro en la prueba.....</i>	<i>93</i>
<i>Figura 120. Velocidad angular del péndulo invertido en la prueba.....</i>	<i>93</i>
<i>Figura 121. Desplazamiento del carro en la prueba.....</i>	<i>94</i>
<i>Figura 122. Ángulo del péndulo invertido en la prueba.....</i>	<i>95</i>
<i>Figura 123. Detalle del ángulo del péndulo invertido en la prueba.....</i>	<i>95</i>
<i>Figura 124. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>96</i>
<i>Figura 125. Velocidad del carro en la prueba.....</i>	<i>96</i>
<i>Figura 126. Velocidad angular del péndulo invertido en la prueba.....</i>	<i>97</i>
<i>Figura 127. Desplazamiento del carro en la prueba.....</i>	<i>98</i>
<i>Figura 128. Ángulo del péndulo invertido en la prueba.....</i>	<i>98</i>
<i>Figura 129. Detalle del ángulo del péndulo invertido en la prueba.....</i>	<i>99</i>
<i>Figura 130. Tensión aplicada al péndulo invertido en la prueba.....</i>	<i>99</i>
<i>Figura 131. Velocidad del carro en la prueba.....</i>	<i>100</i>
<i>Figura 132. Velocidad angular del péndulo invertido en la prueba.....</i>	<i>100</i>
<i>Figura 133. Convenio de ángulos.....</i>	<i>101</i>
<i>Figura 134. Tensión aplicada al carro para la autoerección.....</i>	<i>102</i>
<i>Figura 135. Ángulo Theta del péndulo en la autoerección.....</i>	<i>103</i>
<i>Figura 136. Ángulo Theta del péndulo en la autoerección.....</i>	<i>106</i>
<i>Figura 137. Tensión aplicada al carro en la autoerección.....</i>	<i>106</i>
<i>Figura 138. Energía del péndulo durante la autoerección.....</i>	<i>107</i>
<i>Figura 139. Ángulo Theta del péndulo en la autoerección.....</i>	<i>108</i>
<i>Figura 140. Tensión aplicada al carro en la autoerección.....</i>	<i>108</i>
<i>Figura 141. Energía del péndulo durante la autoerección.....</i>	<i>109</i>
<i>Figura 142. Arquitectura del programa de la aplicación.....</i>	<i>111</i>
<i>Figura 143. Ventana del proyecto de la aplicación.....</i>	<i>112</i>
<i>Figura 144. Arquitectura del programa de ordenador.....</i>	<i>113</i>
<i>Figura 145. Front Panel de HMI.vi.....</i>	<i>114</i>

<i>Figura 146. Código de HMI.vi .....</i>	114
<i>Figura 147. Front Panel de los VI de preparación .....</i>	115
<i>Figura 148. Código de Pendulo Invertido Prep.vi I.....</i>	116
<i>Figura 149. Código de Pendulo Invertido Prep.vi II.....</i>	117
<i>Figura 150. Front Panel de los VI de monitorización.....</i>	118
<i>Figura 151. Código de Pendulo Invertido SS Monitorizacion.vi I.....</i>	119
<i>Figura 152. Código de Pendulo Invertido SS Monitorizacion.vi II.....</i>	119
<i>Figura 153. Front Panel I de los VI de análisis.....</i>	120
<i>Figura 154. Front Panel II de los VI de análisis.....</i>	120
<i>Figura 155. Código de Pendulo Invertido SS Analisis.vi I.....</i>	121
<i>Figura 156. Código de Pendulo Invertido SS Analisis.vi II.....</i>	121
<i>Figura 157. Arquitectura del programa de control en tiempo real.....</i>	123
<i>Figura 158. Código de Inicio.vi.....</i>	124
<i>Figura 159. Código de Pendulo Gantry PID RT.vi I.....</i>	125
<i>Figura 160. Código de Pendulo Gantry PID RT.vi II.....</i>	126
<i>Figura 161. Código de Pendulo Invertido SS RT.vi I.....</i>	127
<i>Figura 162. Código de Pendulo Invertido SS RT.vi II.....</i>	128
<i>Figura 163. Código de Pendulo Invertido SS RT.vi III.....</i>	129
<i>Figura 164. Tarjeta NI myRIO lista para iniciar la aplicación de ordenador .....</i>	130
<i>Figura 165. Tarjeta NI myRIO lista para iniciar la prueba en tiempo real.....</i>	130
<i>Figura 166. Tarjeta NI myRIO ejecutando el control en tiempo real sobre la maqueta IP02 .....</i>	131
<i>Figura 167. Configuración del Shared Variable Engine.....</i>	134
<i>Figura 168. Configuración de la función FIFO de las variables .....</i>	135
<i>Figura 169. Creación de una nueva aplicación en tiempo real.....</i>	137
<i>Figura 170. Información de la aplicación en tiempo real .....</i>	138
<i>Figura 171. Archivos incluidos en la aplicación en tiempo real.....</i>	139
<i>Figura 172. Carga de la aplicación en tiempo real en la tarjeta NI myRIO .....</i>	140
<i>Figura 173. Creación de una nueva aplicación para ordenador .....</i>	141
<i>Figura 174. Información de la aplicación para ordenador.....</i>	142
<i>Figura 175. Archivos incluidos en la aplicación para ordenador.....</i>	142
<i>Figura 176. Creación de un nuevo instalador.....</i>	143
<i>Figura 177. Información del instalador .....</i>	144
<i>Figura 178. Carpeta de destino del instalador.....</i>	145
<i>Figura 179. Aplicaciones incluidas en el instalador.....</i>	145
<i>Figura 180. Accesos directos creados en la instalación .....</i>	146
<i>Figura 181. Software adicionales incluidos en la instalación.....</i>	147
<i>Figura 182. Requerimientos del sistema de destino.....</i>	148
<i>Figura 183. Aspecto final de la ventana de proyecto.....</i>	149
<i>Figura 184. Diagrama de Gantt de las tareas principales del proyecto .....</i>	152
<i>Figura 185. Diagrama de Gantt de todas las tareas del proyecto.....</i>	154

## **LISTA DE TABLAS**

Tabla 1. Lista de componentes de la maqueta IP02 de Quanser .....	25
Tabla 2. Identificación de los cables de los encoders.....	28
Tabla 3. Detalles de las conexiones.....	29
Tabla 4. Variables del sistema .....	33
Tabla 5. Caracterización de los encoders .....	33
Tabla 6. Valores del algoritmo de autoerección para el péndulo corto .....	105
Tabla 7. Valores del algoritmo de autoerección para el péndulo largo .....	107
Tabla 8. Variables de la librería Librería PC.lvlib .....	122
Tabla 9. Variables de la librería en red Librería En Red.lvlib .....	133
Tabla 10. Lista de tareas del proyecto .....	151
Tabla 11. Carga de trabajo de las tareas principales del proyecto .....	155
Tabla 12. Lista de mano de obra y materiales con sus cargas horarias y cantidades	156
Tabla 13. Precios unitarios, vida útil y uso anual de los conceptos .....	157
Tabla 14. Tasas de amortización de los materiales .....	157
Tabla 15. Descargo de gastos del proyecto .....	158

## **ABREVIATURAS**

1. **NI:** National Instruments, empresa desarrolladora del software y tarjeta de adquisición de datos utilizados.
2. **UPM:** Universal Power Module, módulo desarrollado por Quanser, que sirve de amplificador de potencia de la señal de control al motor de accionamiento del carro.
3. **DC:** corriente continua, del inglés Direct Current.
4. **SISO:** sistema de una entrada y una salida, del inglés Simple Input Simple Output system.
5. **SIMO:** sistema de una entrada y múltiples salidas, del inglés Simple Input Multiple Output system.
6. **MIMO:** sistema de múltiples entradas/salidas, del inglés Multiple Input Multiple Output system.
7. **CDM:** Centro De Masas, punto geométrico de un sistema de masas, discreto o continuo, que dinámicamente se comporta como si en él estuviera aplicada la resultante de las fuerzas externas al sistema.
8. **FPGA:** Field-Programmable Gate Array o matriz de puertas programables.
9. **LQR:** Linear Quadratic Regulator, controlador basado en la minimización del error cuadrático.
10. **PID:** Proportional Integral Derivative, algoritmo de control de salida proporcional al error, a su integral y a su derivada.
11. **USB:** Universal Serial Bus, conexión serie ampliamente utilizada.
12. **VI:** Virtual Instrument, archivo de programa de LabVIEW.
13. **SVE:** Shared Variable Engine, herramienta que en LabVIEW se encarga del tratamiento de las variables compartidas, tanto de su escaneo como de su actualización.

## **1. INTRODUCCIÓN**

Este documento contiene el trabajo de fin de máster “Desarrollo de controladores avanzados para sistemas multivariable en el entorno LabVIEW” realizado por Miguel Dorronsoro López de Maturana y dirigido por D. Fernando Artaza Fano durante el curso académico 2017/2018 en la Escuela de Ingeniería de Bilbao.

Las asignaturas de Automática y Control impartidas en Escuelas Técnicas utilizan muchas herramientas matemáticas y algoritmos necesarios para explicar las diversas teorías de control. Estas imparticiones están apoyadas con ejemplos y ejercicios, pero que también se realizan de una manera teórica sobre el papel.

El alumno, no obstante, rara vez tiene acceso a experimentar con maquetas y sistemas reales para aplicar los conocimientos teóricos adquiridos. Ello se debe a la dificultad de implementar estas prácticas en un laboratorio y medir los resultados de las mismas.

Por otro lado, el sistema de Péndulo Invertido es el paradigma de la dificultad de control, ya que se trata de un sistema complejo, altamente inestable, multivariable y no lineal.

Por estas razones en este proyecto se diseña un banco de pruebas que permita al usuario experimentar y afianzar los conocimientos adquiridos sobre Teoría de Control Automático.

El banco de pruebas estará basado en la maqueta IP02 de un péndulo invertido, fabricado por Quanser, y en la aplicación software LabVIEW de National Instruments, empresa que también suministra la tarjeta NI myRIO, en la que se ejecutará el control en tiempo real.

En el inicio de este documento se presenta el proyecto, desarrollando los apartados apartados de contexto, objetivos y alcance y beneficios del mismo.

Posteriormente, se analizan algunas alternativas existentes para llevar a cabo el proyecto, y se justifican las elecciones tomadas. Seguidamente, se describe la metodología seguida para llevar a cabo el trabajo.

Tras este análisis técnico, se muestra la planificación según la que se ha ejecutado el proyecto y los costes del mismo, en el descargo de gastos.

Finalmente, se hace un resumen de lo esencial en el apartado de conclusiones. Además, se incluyen cinco anexos: uno en el que se desarrolla la obtención de los modelos matemáticos de los sistemas de estudio, otro en el que se recogen los parámetros de los sistemas, otro en el que se recogen los valores óptimos de las ganancias de los controladores que son objeto de estudio, un manual de usuario para la aplicación producto del proyecto y, por último, el código de la aplicación en LabVIEW.

## 2. CONTEXTO

Este trabajo nace como respuesta a la necesidad del alumno de realizar un trabajo de fin de máster, requisito necesario para obtener el graduado en el Máster en Ingeniería Industrial de la Universidad del País Vasco. Para ello, se contacta con D. Fernando Artaza Fano, profesor de la Escuela de Ingeniería de Bilbao y director del trabajo, quien propone dicho trabajo.

El proceso se lleva a cabo durante el segundo curso del máster (2017-2018), a partir de febrero de 2018, teniendo como objetivo su finalización para finales de septiembre de 2018, de tal manera que pueda ser entregado en la convocatoria de septiembre del mismo año, y presentado en octubre.

El trabajo se desarrolla en uno de los laboratorios de control de la Escuela de Ingeniería de Bilbao, donde se encuentran los software y hardware necesarios para la consecución del trabajo.



Figura 1. Logo de la Escuela de Ingeniería de Bilbao

En la ejecución de este trabajo de fin de máster, se utilizan la **maqueta IP02** de Quanser, que permite probar los sistemas carro, péndulo Gantry (Grúa) y péndulo invertido, la aplicación **LabVIEW**, para diseñar y desarrollar controladores y aplicaciones para la maqueta y la **tarjeta NI myRIO**, para implementar dichos controladores en la maqueta.

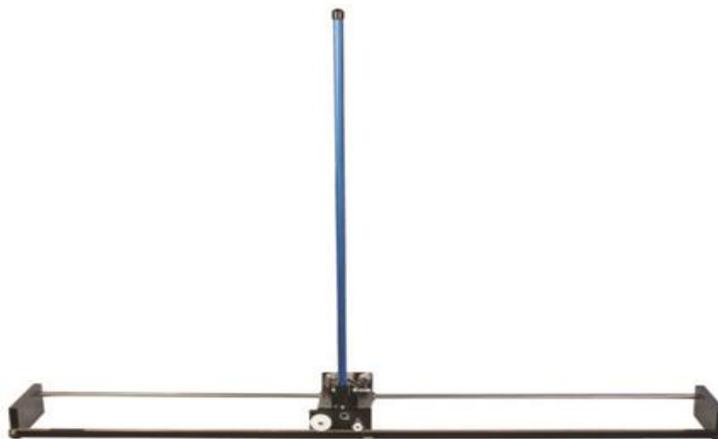


Figura 2. Maqueta IP02 de Quanser

**National Instruments** es la empresa suministradora de la aplicación LabVIEW, ampliamente usada por ingenieros para generar prototipos e implementar aplicaciones de registro, diagnóstico, monitorización y control en una gran variedad de entornos

industriales. National Instruments también proporciona la tarjeta NI myRIO para adquisición de datos (sensores) y generación de señales de control de acuerdo a los algoritmos de control, todo ello, además, en tiempo real.



*Figura 3. Logo de National Instruments y su software LabVIEW*

En LabVIEW destaca el propio lenguaje de programación de la aplicación, que se constituye mediante bloques funcionales y cables de conexión virtuales entre los mismos, haciendo posible un mayor rendimiento en la ejecución de programa, debido a que varios procesos paralelos pueden ser tratados simultáneamente por distintos procesadores de la computadora.

Entre las distintas aplicaciones en la industria, LabVIEW es utilizado para:

- Sistemas de prueba de Aerogeneradores.
- Monitorización de variables de redes eléctricas para control dinámico (Smart Grid).
- Monitorización de variables de máquinas para seguimiento de estado, mantenimiento predictivo, etc...
- Diseño de máquinas inteligentes.
- Robótica.

En el campo de la Docencia y Desarrollo LabVIEW permite:

- Aprendizaje de herramientas de desarrollo de sistemas embebidos.
- Aprendizaje de programación FPGA.
- Realización de prácticas de simulación y control.

Debido a que el alumno no tiene conocimiento alguno de LabVIEW al iniciar el proyecto, el estudio de la documentación y aprendizaje del software y su lenguaje de programación constituyen las primeras tareas a llevar a cabo.

Finalmente, cabe destacar que como producto final de este trabajo se creará una herramienta docente que podrán usar alumnos de la Escuela de Ingeniería de Bilbao para implementar distintos controladores sobre la maqueta IP02 y sobre modelos teóricos.

## **3. OBJETIVOS Y ALCANCE**

El objetivo inicial de este trabajo de fin de máster es, por tanto, utilizar el software LabVIEW de National Instruments para desarrollar diferentes controladores para tres sistemas diferentes, un sistema carro-servomotor, un sistema de péndulo invertido y un sistema de péndulo Gantry (Grúa), que pueden ser configurados en la misma maqueta IP02. La conexión y el control de ésta se realizarán mediante la tarjeta NI myRIO de National Instruments.

Además de dar los valores óptimos para los parámetros de estos controladores, se desarrollará, utilizando el mismo programa, un banco de pruebas que permita al usuario experimentar y afianzar los conocimientos sobre Teoría de Control adquiridos durante las clases lectivas de la especialidad.

Una vez desarrollado el banco de pruebas, el usuario podrá seleccionar diferentes modos de control, dependiendo del sistema a controlar: modo carro, modo péndulo Gantry o modo péndulo invertido, así como experimentar con diferentes algoritmos de control como LQR, PID, ... De esta forma, el usuario podrá comparar los resultados de las distintas estrategias de control, observar los efectos de las ganancias asignadas en cada caso, comparar entre modelo teórico y sistema real, ... etc., todo ello por medio de una interfaz gráfica de usuario.

En resumen, el principal objetivo final del proyecto es crear una herramienta didáctica que:

- Ayude a estudiar y comparar 3 sistemas distintos, de distinto número de entradas y salidas, diferentes dinámicas...
- Ayude a entender la diferencia entre distintos tipos de controlador.
- Ayude a ver la diferencia entre modelo matemático y sistema real.
- Sea útil para practicar en la sintonización de un controlador.

Para facilitar el uso de esta interfaz de usuario, se incluirá un anexo que contenga un manual de usuario para la misma, explicando las opciones que ésta ofrece y cómo se debe utilizar, así como la conexión de la tarjeta a la maqueta y al ordenador en el que se aloje la aplicación de la interfaz de usuario.

## 4. BENEFICIOS DEL PROYECTO

En este apartado se explican los beneficios que el proyecto trae consigo.

Por un lado, académicamente este trabajo es muy beneficioso para el alumno. Para empezar, le ofrece la posibilidad de trabajar con tres sistemas reales distintos; concretamente, poder trabajar con un servo lineal, un sistema muy extendido en la industria (por ejemplo, en máquina-herramienta, accionamientos para manipulación de cargas, etc...), un péndulo Gantry, que se asemeja a los objetivos de control de un puente-grúa, y un péndulo invertido, un sistema muy inestable y SIMO (Single Input Multiple Output), resultando muy instructivo para el alumno. El control de un péndulo invertido tiene múltiples aplicaciones, que van desde el control de equilibrio y movimiento de Robots Bípedos, vehículos de 2 ruedas, despegue y navegación de cohetes, etc...

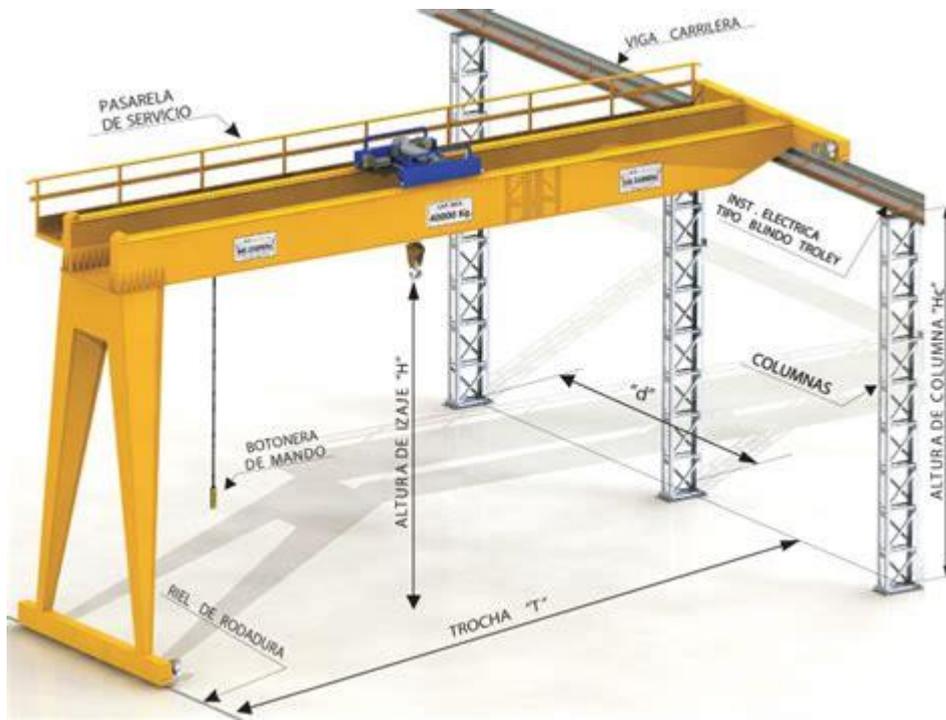


Figura 4. Puente grúa



Figura 5. Ejemplos de aplicación del péndulo invertido a sistemas reales

Además, el trabajo permite al alumno trabajar con el software de *LabVIEW*, un software complejo pero muy potente, y muy extendido en la industria, especialmente para monitorización y automatización. También se trabaja con la tarjeta *NI myRIO*, parte clave para la sensorización, actuación y el control en tiempo real, esta última característica imprescindible para el control en sistemas rápidos y altamente inestables.

Así mismo, la posibilidad de utilizar distintos tipos de controlador para un mismo sistema, permite al alumno obtener un criterio comparativo en cuanto al funcionamiento de los mismos, sus prestaciones y dificultad de sintonización.

Por otro lado, como ya se ha mencionado, el producto final de este trabajo es un programa de interfaz de usuario que permite estudiar los tres sistemas mencionados, utilizar sobre los mismos distintos controladores y comparar el modelo matemático con el sistema real; es decir, una herramienta didáctica que podrá ser utilizada en la Escuela de Ingeniería de Bilbao para que los alumnos puedan apreciar las diferencias entre modelo y sistema real, entre los tres sistemas de estudio, entre los distintos tipos de controlador o aprender a sintonizar un controlador utilizando el modelo teórico.

Finalmente, cabe destacar que para usar la maqueta en los ordenadores que hay hoy en día en los laboratorios del Departamento de Sistemas y Automática de la Escuela de Ingeniería de Bilbao (y, en general, en cualquier ordenador moderno) es necesario el uso de una tarjeta de adquisición de datos distinta a la original que se suministraba con la maqueta de Quanser, debido a un cambio en los puertos de adquisición de datos de los ordenadores. Por lo tanto, la adecuación de la tarjeta *NI myRIO* permitirá seguir utilizando estas maquetas.

## 5. ANÁLISIS DE ALTERNATIVAS

En este apartado se presentan las distintas alternativas posibles en diferentes aspectos y actividades del proyecto que se han presentado durante el desarrollo del mismo, justificando las razones por las que se han elegido unas y por las que se han descartado las otras.

### 5.1. OBTENCIÓN DEL MODELO

Existen diferentes maneras de obtener el modelo de un sistema:

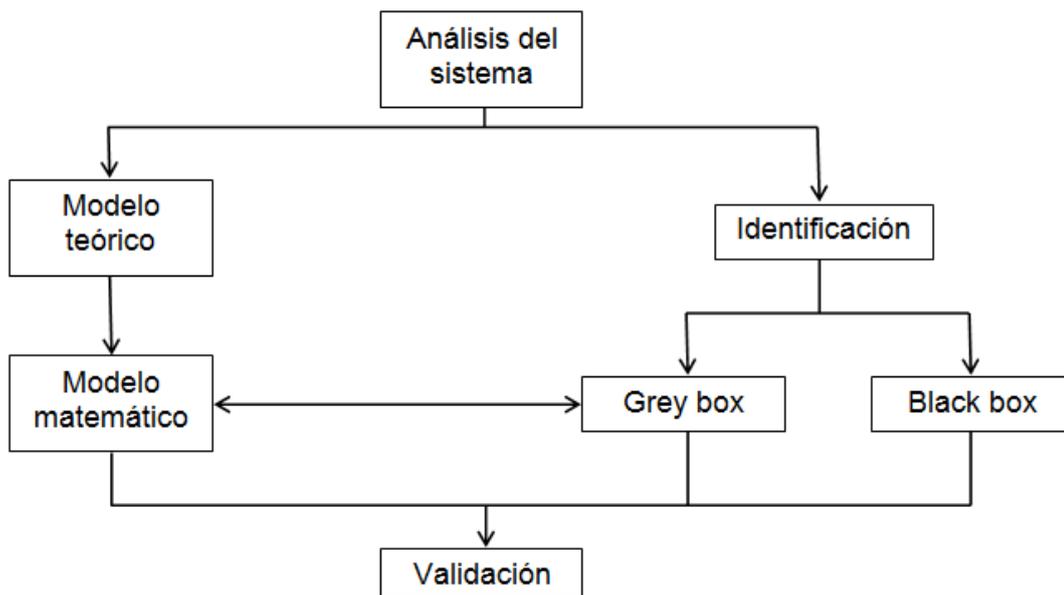


Figura 6. Posibles modos de obtención del modelo

Se puede optar por identificar el sistema, es decir, obtener un modelo de forma experimental. Mientras que la estrategia “Black box” trata el sistema como una caja negra, realizando la identificación a partir de las salidas obtenidas en función de las entradas aplicadas, la estrategia “Grey box” utiliza datos conocidos del modelo matemático (como la forma de una ecuación que representa al sistema) y, de la misma manera que el anterior, identifica los parámetros desconocidos.

Sin embargo, debido a la simplicidad de los sistemas, documentación existente sobre péndulos invertidos en general y a la detallada información sobre los valores de los parámetros del sistema (que se puede consultar en *ANEXO II: PARÁMETROS DEL SISTEMA*), se decide realizar un modelo matemático del sistema. Posteriormente, en una etapa de validación de los modelos, comparando los sistemas reales con sus respectivos modelos, se corregirán determinados parámetros que puedan distar de su valor real.

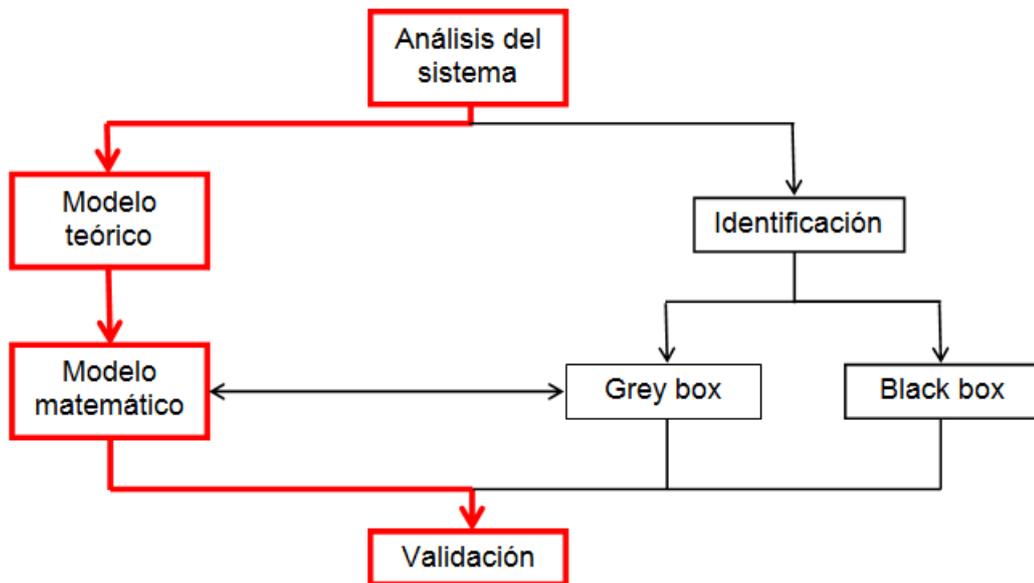


Figura 7. Modo de obtención del modelo seleccionado

## 5.2. TIPOS DE CONTROLADOR INCLUIDOS

Existen muchas estrategias de control, y dado que no hay restricciones en cuanto a cuáles usar en este proyecto, es preciso elegir, tanto las que se van a analizar para el sistema como las que se van a incluir en la aplicación de usuario que luego se desarrollará.

### 5.2.1. CONTROLADOR PID

Aunque el controlador PID no puede considerarse un controlador avanzado, debido a su extendido uso en la industria y a su sencillez, se decide incluir este algoritmo que, como muestra la ecuación (1) se compone de una acción proporcional al error, otra proporcional a la integral del error y otra proporcional a la derivada del mismo.

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt + K_c T_d \frac{de(t)}{dt} \quad (1)$$

En la Figura 8,  $K_i = \frac{K_c}{T_i}$  y  $K_d = K_c T_d$ .

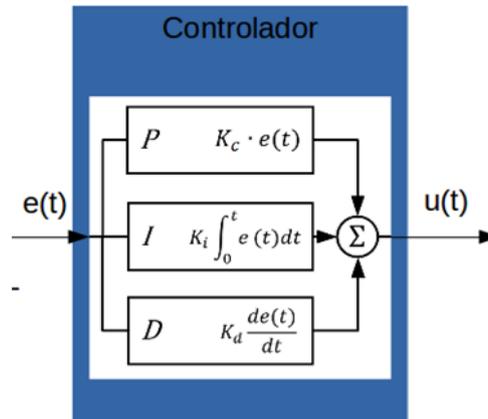


Figura 8. Esquema del controlador PID

No obstante, este algoritmo no tiene la capacidad y potencia de otros controladores porque no utiliza todas las variables de estado del sistema.

### **5.2.2. CONTROLADOR POR REALIMENTACIÓN DEL ESTADO**

Expresar un sistema por medio de un sistema de estados es una técnica muy utilizada que permite el desarrollo de muchas estrategias de control que se basan en la medición del estado del sistema.

Un sistema de estados se construye representando las ecuaciones del sistema por medio de un número de variables de estado y sus derivadas. En general, en el caso continuo, sus ecuaciones matriciales son:

$$\dot{x}(t) = A(t) x(t) + B(t) u(t) \quad (2)$$

$$y(t) = C(t) x(t) + D(t) u(t) \quad (3)$$

siendo  $x(t)$  el vector de estado, que contiene las variables de estado,  $u(t)$  el vector de las entradas al sistema,  $y(t)$  el vector de salidas del sistema y el resto de términos matrices que determinan la relación entre los anteriores.

Por lo tanto, en su concepción más simple, se puede usar una realimentación del estado multiplicado por una matriz de ganancias  $K$  para controlar el sistema, como se muestra en la *Figura 9*.

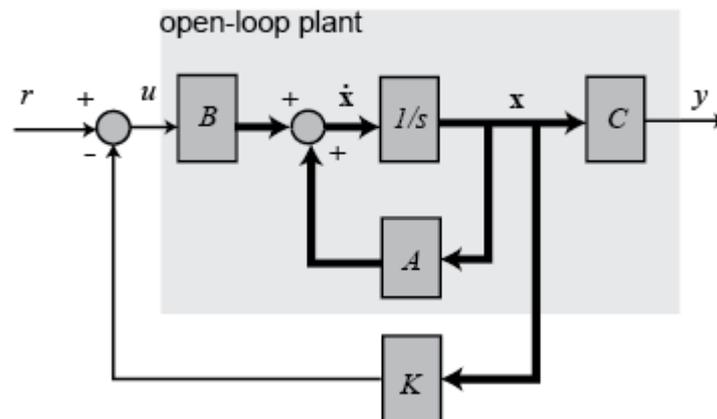


Figura 9. Esquema de control por realimentación del estado

El controlador por realimentación de estado es, por tanto, un algoritmo potente que utiliza todas las variables de estado debidamente ponderadas mediante una matriz de ganancias  $K$ . Existen diferentes métodos para seleccionar la matriz  $K$  adecuada, como se indica a continuación.

### **5.2.2.1. ASIGNACIÓN DE POLOS**

Este método consiste en seleccionar los polos deseados del sistema en lazo cerrado, es decir, se elige el comportamiento del conjunto sistema-controlador seleccionando los polos del mismo y, por lo tanto, su dinámica (tiempo de establecimiento, amortiguamiento, sobreoscilación...). Elegidos los polos, se calcula  $K$  con facilidad, a partir de la igualdad deseada entre el sistema con el estado realimentado y el sistema ficticio cuyos polos son los deseados.

Se decide incluir este método, ya que es interesante sintonizar un controlador de esta manera, es decir, seleccionando la dinámica deseada en el sistema.

### **5.2.2.2. CONTROLADOR LQR**

El método LQR (Linear Quadratic Regulator) es una estrategia de control óptimo, que calcula  $K$  en base a la minimización de la siguiente función de coste:

$$J = \int_0^{\infty} (x^T Q x + u^T R u) dt \quad (4)$$

donde  $x$  y  $u$  son el vector de estado y el vector de entradas, respectivamente;  $x^T$  y  $u^T$  sus respectivos vectores transpuestos; y  $Q$  y  $R$  son las matrices de peso que ponderan el estado y las entradas respectivamente. Por lo tanto, dando un valor alto a un término de la diagonal de  $Q$  se da importancia a que ese estado siga a la referencia sin error, mientras que en el caso de  $R$  se da importancia a la minimización de la energía utilizada en esa entrada, es decir, se limita la misma.

Cabe destacar que la matriz  $Q$  debe ser simétrica semidefinida positiva, mientras que  $R$  debe ser simétrica definida positiva.

Existen ponderaciones que desdobl原因 la ponderación del estado en la función de coste en una ponderación del estado final y una ponderación del estado a lo largo de la trayectoria. Sin embargo, se va a utilizar el supuesto anterior, en el que existe la misma ponderación para ambos.

El problema se soluciona minimizando la función  $J$  en  $x$  y  $u$  y, en el caso de sistemas lineales, a partir de la ecuación de Ricatti resultante se puede calcular la acción de control óptima  $u_{opt}$ , que en este caso tiene la forma de una realimentación de estado, por lo que también se puede calcular la matriz de ganancias  $K$ , que apenas varía en el tiempo y puede considerarse constante en este supuesto.

Esta estrategia resulta muy interesante desde el punto de vista académico, ya que no es tan sencilla como las anteriores, pero es una estrategia de control óptimo. Además, permite apreciar cómo cambia la salida y la trayectoria del estado en general, si se modifica el peso asignado en cada término. Es por esto que también se decide incluir este controlador.

De las estrategias que se analizan para cada uno de los sistemas de estudio, todas para las que se pueda diseñar un controlador adecuado serán incluidas en la aplicación de usuario.

### **5.2.3. CONTROL PREDICTIVO**

Se trata de técnicas de control óptimo, es decir en las que se minimiza una función de coste, como en el caso del controlador LQR, pero en este caso la optimización se realiza en cada instante de muestreo, teniendo en cuenta los valores futuros (predichos) de la respuesta en función de la acción de control que se aplica.

Se decide no utilizar este método por su complejidad, porque el alumno no tiene experiencia con él y porque la carga de computación de estos métodos es muy alta, por lo que podría entrar en conflicto con la alta frecuencia de muestreo que se requiere por ejemplo para controlar el péndulo invertido.

### **5.2.4. REDES NEURONALES**

Las redes neuronales son un paradigma de aprendizaje y procesamiento automático inspirado en la forma en que funciona el cerebro humano. Mediante múltiples ensayos, son capaces de “aprender” y ajustar un algoritmo de control a un sistema en estudio.

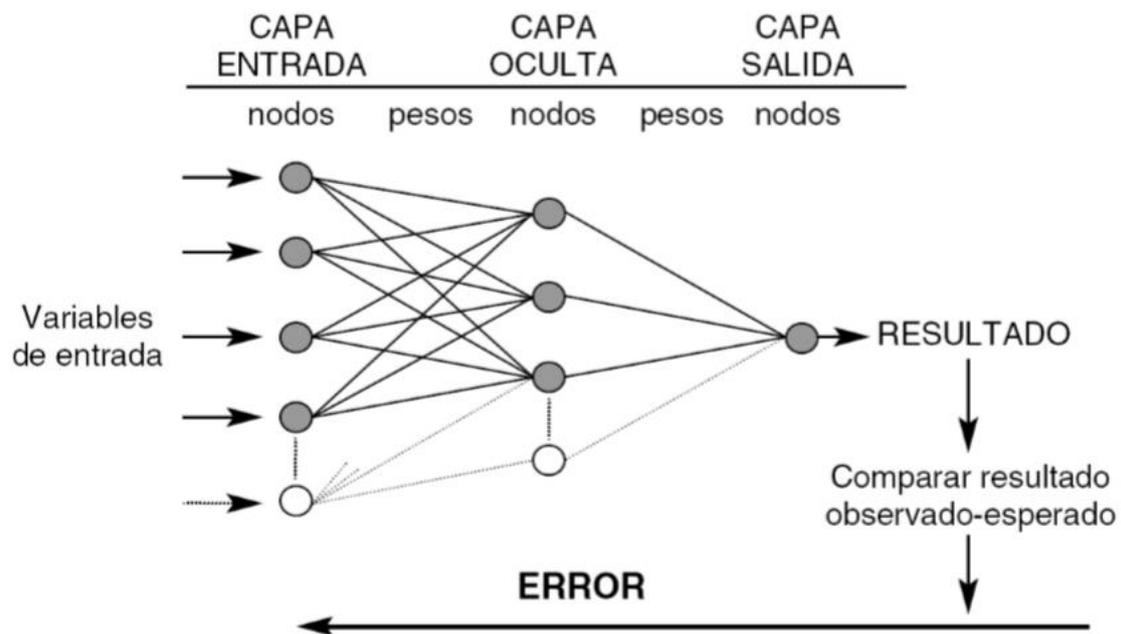


Figura 10. Topología de una red neuronal

En este caso, las redes neuronales se consideran una opción muy sofisticada para los sistemas de estudio, que no son tan complejos en cuanto a su dinámica. Por lo tanto, se decide no utilizar esta estrategia.

### **5.3. TIEMPO DE MUESTREO**

Por la naturaleza de la programación en LabVIEW y el módulo NI myRIO de control que se usa, es necesario seleccionar un tiempo de muestreo con el que trabajar, es decir, cada cuánto tiempo se leen entradas, se ejecuta el algoritmo de control en la tarjeta NI myRIO y se actualizan salidas.

Teniendo en cuenta el teorema de Nyquist, la frecuencia de muestreo  $f_s$  debe ser mayor que el doble del ancho de banda de la señal, en este caso las lecturas de los encoders.

La señal que determina la posición horizontal del carro, así como la del péndulo no son excesivamente rápidas, por lo que una frecuencia de 50 muestras por segundo ( $f_s = 50 \text{ Hz}$ ) podría ser suficiente. Sin embargo, dado que el sistema de péndulo invertido es muy inestable, se necesita una acción de control rápida para mantener el equilibrio inestable, necesiándose por lo tanto un tiempo de muestreo pequeño. Tras pruebas, se determina que  $T_s = 0.002 \text{ s}$ , o lo que es lo mismo,  $f_s = 500 \text{ Hz}$  es un buen valor para el péndulo invertido.

Por simplicidad, aunque no se necesite un muestreo tan rápido en los otros sistemas, se usa la misma frecuencia de muestreo, ya que se comprueba que no es demasiado alta como para saturar la CPU de la tarjeta NI myRIO.

## 5.4. CÁLCULO DE LAS DERIVADAS

El control por realimentación de estados utiliza los 4 estados del sistema que son el desplazamiento del carro, el ángulo del péndulo y sus respectivas derivadas.

Las dos primeras variables son medibles (por los encoders), pero las derivadas no, por lo que deben ser obtenidas por derivación de las dos primeras o bien estimadas.

Para estimarlas, una opción es utilizar un observador del estado. Un observador es un algoritmo que estima el estado del sistema, parcial o totalmente, a partir del conocimiento de la entrada y la salida, conocidas en este caso. Además, es una opción que, por lo general, no presenta las discontinuidades y el ruido que pueden aparecer en la derivación de la señal.

Para utilizar observadores es condición necesaria y suficiente que el estado sea completamente observable.

Una vez obtenido el modelo, proceso que se explica en el apartado 8.4. *MODELOS TEÓRICOS*, se puede comprobar si el sistema es observable mediante la matriz de observabilidad, verificando que no es singular. En LabVIEW, mediante el bloque "CD Observability Matrix.vi" de la *Figura 11*, se comprueba que todos los sistemas de estudio son observables.

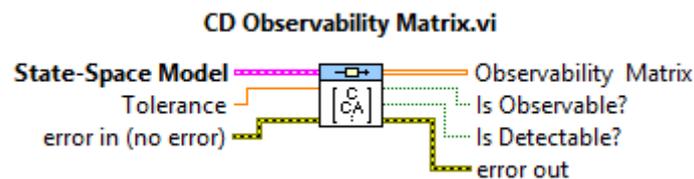


Figura 11. Bloque en LabVIEW para comprobar la observabilidad

En el caso del observador de orden completo, se estima el estado entero, considerado desconocido al inicio. Mediante una matriz  $L$  se ajusta la dinámica del observador, es decir, la velocidad con la que el error de observación ( $y(t) - \tilde{y}(t)$ ) tiende a cero, siendo  $\tilde{y}(t)$  la salida estimada por el observador, a partir del estado estimado  $\tilde{x}(t)$ .

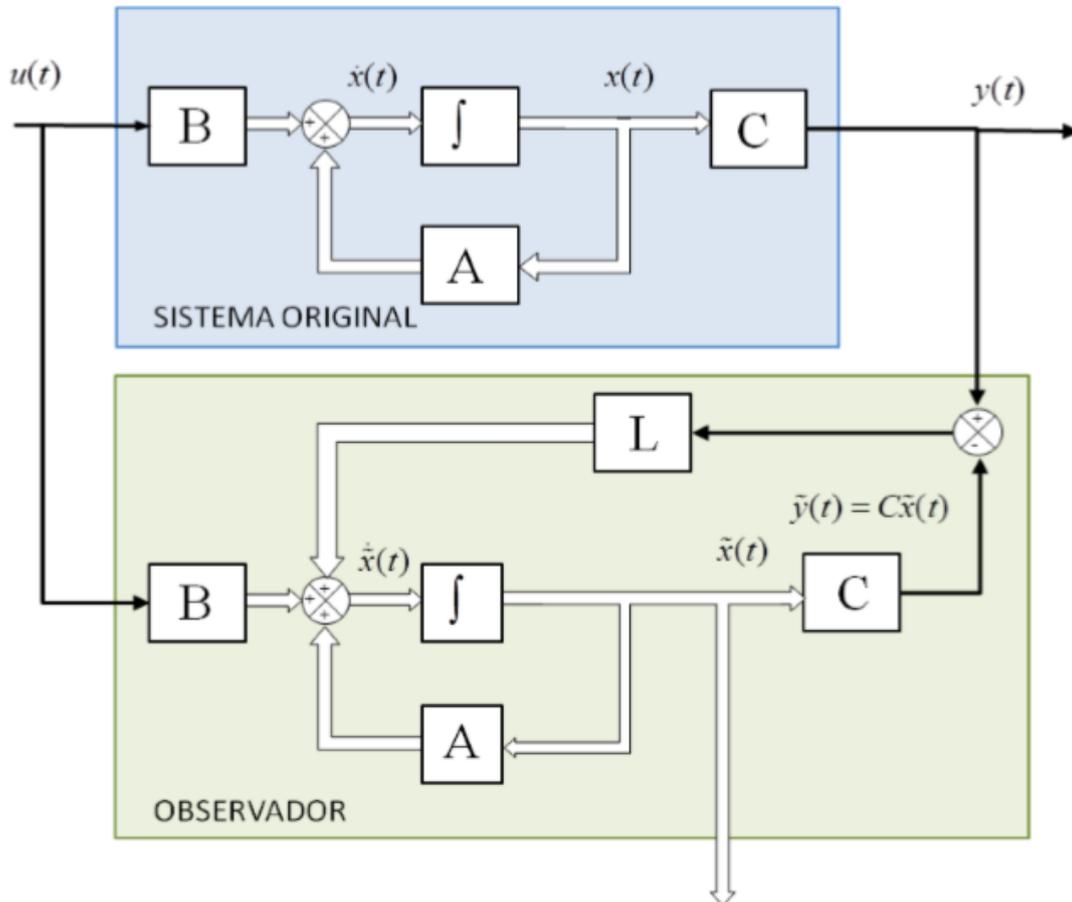


Figura 12. Esquema del observador de orden completo continuo

La ecuación (5) da el algoritmo del observador, a partir del cual se calcula el estado estimado  $\tilde{x}(t)$ . Una vez calculado, a efectos prácticos, éste es el estado que se realimenta en un controlador por realimentación del estado.

$$\dot{\tilde{x}}(t) = A\tilde{x}(t) + Bu(t) + L(y(t) - C\tilde{x}(t)) = (A - LC)\tilde{x}(t) + Bu(t) + Ly(t) \quad (5)$$

En este caso, el observador a utilizar debería ser discreto, aunque su algoritmo es muy parecido, y viene dado por la ecuación (6).

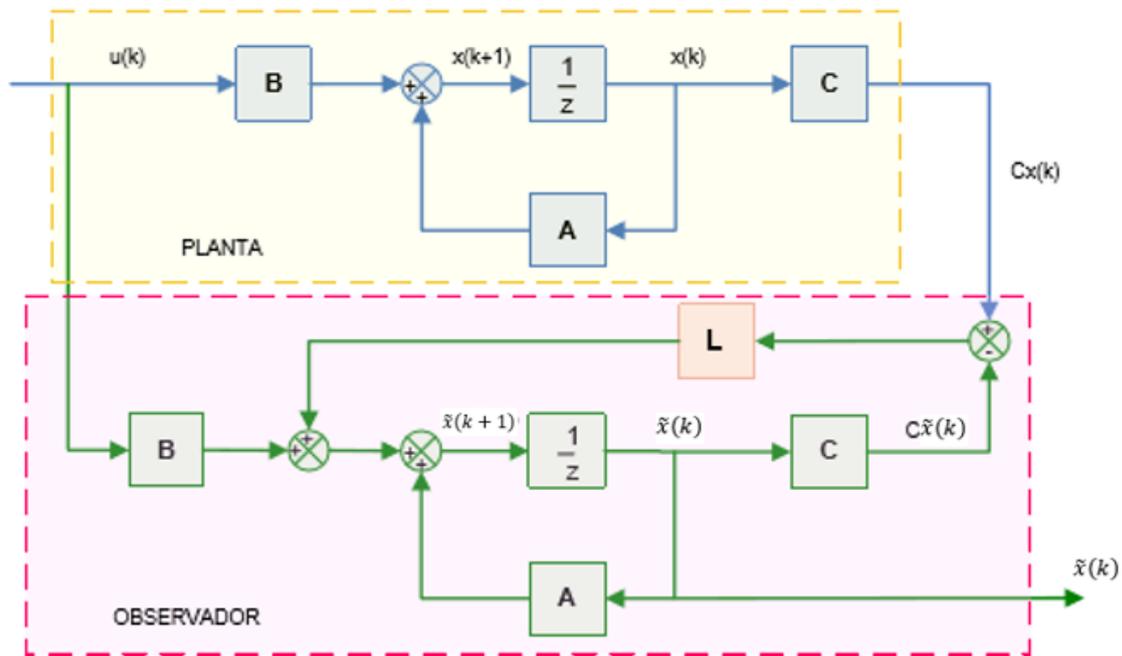


Figura 13. Esquema del observador de orden completo discreto

$$\tilde{x}(k + 1) = A \tilde{x}(k) + Bu(k) + L(y(k) - C\tilde{x}(k)) \quad (6)$$

El problema de un observador de orden completo en este caso, en el que se conocen dos variables de estado, es que se desecha información, estimándose de nuevo éstas dos variables, y sólo se utilizan las variables medidas para conocer la salida del sistema. Por lo tanto, no es un buen aprovechamiento de los recursos disponibles.

En el caso del observador reducido, esto no es un problema, ya que se utiliza la información de las dos variables de estado conocidas. Sin embargo, el algoritmo es notablemente más complejo, como se muestra en la figura siguiente.

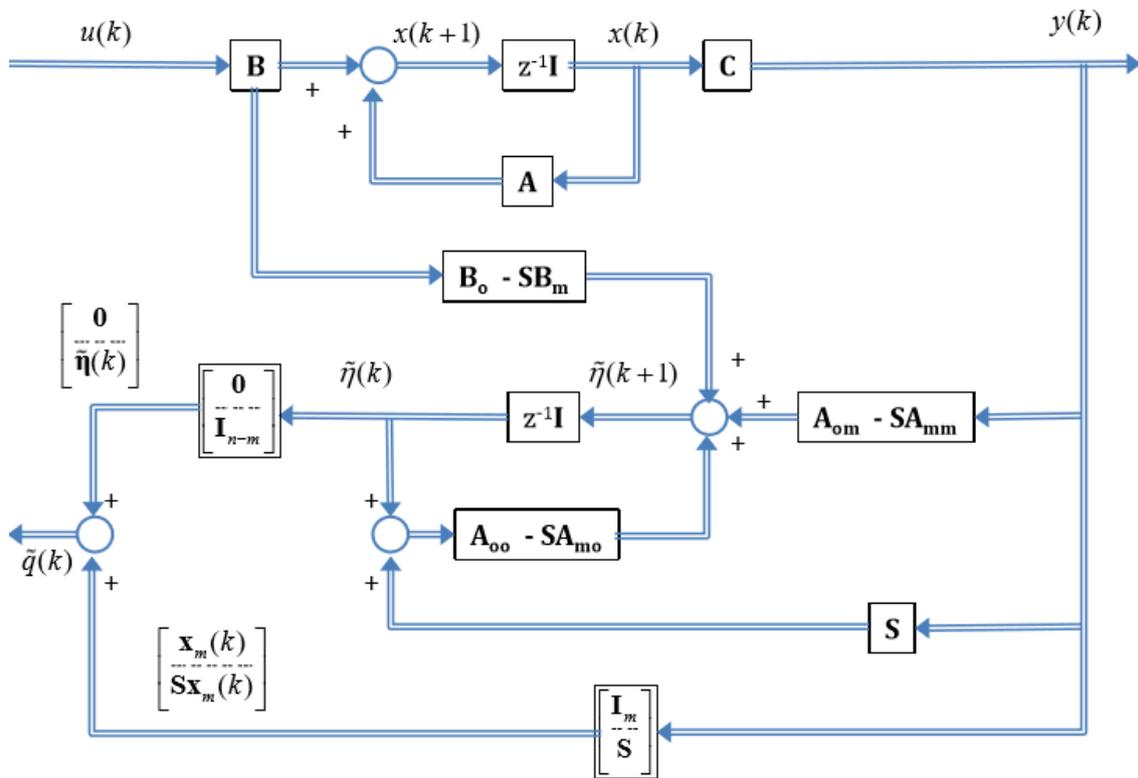


Figura 14. Esquema del observador de orden reducido discreto

Así mismo, cabe destacar que los observadores están basados en el modelo real, por lo que, si éste no es muy fiel al sistema real, nunca se eliminará el error de estimación.

Por todas las razones anteriores y buscando la sencillez, se decide derivar las señales.

Las técnicas de derivación pueden ser de varias formas, pero en todos los casos se requiere un filtrado adicional porque la función derivación tiende a generar y amplificar el ruido de presente en la señal.

Este fenómeno se agrava debido a la discretización de las señales obtenidas de los encoders.

Una primera opción es obtener las derivadas mediante cálculo de diferencias finitas (ecuación (7)), aplicando posteriormente un filtro a las señales así obtenidas.

$$\dot{x}(k) = \frac{x(k) - x(k-1)}{T_s} \quad (7)$$

Sin embargo, tras varias pruebas realizadas, los resultados no son del todo buenos, principalmente debido al ruido que introduce dicho cálculo, así que se opta por utilizar la siguiente función aplicada a la señal.

$$H(s) = [s] \left[ \frac{\omega_c^2}{s^2 + 2\zeta\omega_c s + \omega_c^2} \right] = \frac{\omega_c^2 s}{s^2 + 2\zeta\omega_c s + \omega_c^2} \quad (8)$$

La función anterior, propuesta por Quanser, tiene dos propósitos, además de la propia función de derivación, añade un filtro pasa-bajos que permite eliminar las altas frecuencias, con una frecuencia de corte  $f_c$  y donde

$$\omega_c = 2 \pi f_c \quad (9)$$

$$\zeta = 0.9 \quad (10)$$

A continuación se muestra la función para  $f_c = 10 \text{ Hz}$  y su diagrama de Bode.

$$H(s) = \frac{3948s}{s^2 + 113s + 3948} \quad (11)$$

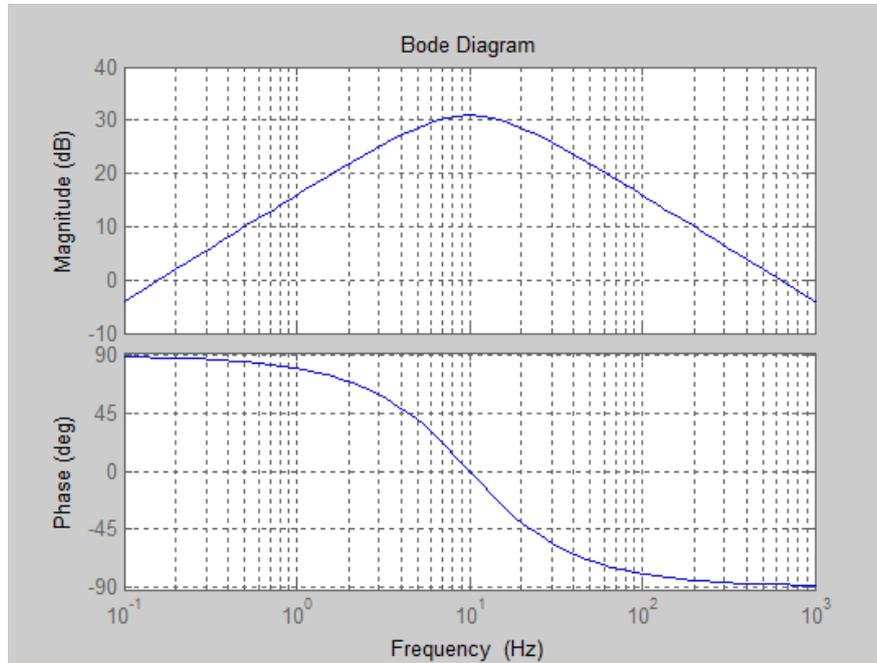


Figura 15. Diagrama de Bode del filtro derivativo

Como puede verse, para frecuencias inferiores a 10Hz,  $H(s)$  funciona derivando la señal de entrada. A partir de esa frecuencia, sin embargo, se comporta como un filtro pasa bajos.

La elección de la frecuencia del filtro debe ser tal que dicha frecuencia sea lo más baja posible, pero sin llegar a afectar a la dinámica y control del sistema.

Tras pruebas realizadas, se concluye que para el funcionamiento del carro y carro-péndulo en modo Gantry (grúa), la frecuencia de 10Hz es la más adecuada.

Sin embargo, en el modo péndulo invertido, al ser el sistema fuertemente inestable, y requerir acciones más rápidas de control, se utiliza una frecuencia superior, de 20Hz.

La función  $H(s)$  es discretizada para su uso en forma de ecuación en diferencias finitas en el código. Para un  $T_s = 0.002 \text{ s}$  se tiene, para 10 Hz:

$$H(z) = \frac{\dot{X}(z)}{X(z)} = \frac{7.049z - 7.049}{z^2 - 1.784z + 0.7977} = \frac{7.049z^{-1} - 7.049z^{-2}}{1 - 1.784z^{-1} + 0.7977z^{-2}} \quad (12)$$

siendo  $X(z)$  la transformada en  $z$  de la variable y  $\dot{X}(z)$  su derivada filtrada. Por lo tanto:

$$\dot{x}(k) = 1.784 \dot{x}(k-1) - 0.7977 \dot{x}(k-2) + 7.049 x(k-1) - 7.049 x(k-2) \quad (13)$$

Para 20 Hz:

$$H(s) = \frac{15790s}{s^2 + 226.2s + 15790} \quad (14)$$

$$H(z) = \frac{\dot{X}(z)}{X(z)} = \frac{25.14z - 25.14}{z^2 - 1.586z + 0.6361} = \frac{25.14z^{-1} - 25.14z^{-2}}{1 - 1.586z^{-1} + 0.6361z^{-2}} \quad (15)$$

$$\dot{x}(k) = 1.586 \dot{x}(k - 1) - 0.6361 \dot{x}(k - 2) + 25.14 x(k - 1) - 25.14 x(k - 2) \quad (16)$$

En el bucle de control en tiempo real se implementan las ecuaciones en diferencias finitas de los filtros, para cuyos parámetros se parte de valores nulos al inicio del bucle.

## 5.5. LECTURA Y ESCRITURA

Para la captura en LabVIEW de las señales de los encoders cableadas a la tarjeta NI myRIO existen dos opciones.

La aplicación myRIO Toolkit de LabVIEW proporciona unos módulos de encoder preprogramados para los que únicamente hay que indicar la entrada de encoder en la tarjeta y el tipo de encoder, en este caso, de cuadratura. LabVIEW muestra el código de este módulo preprogramado, compuesto de bloques elementales.

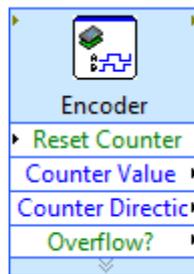


Figura 16. Módulo preprogramado de encoder para NI myRIO

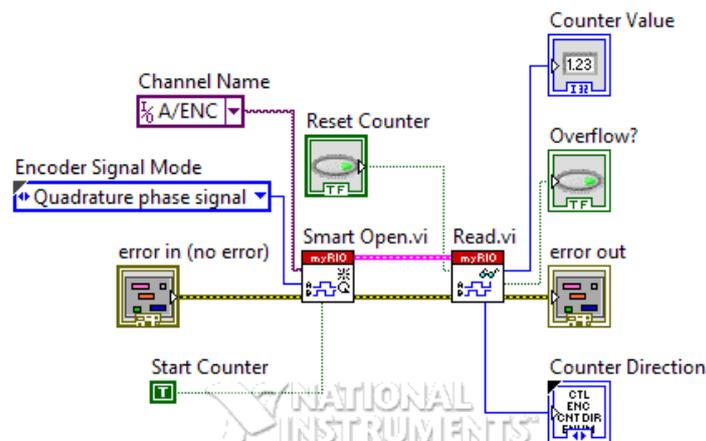


Figura 17. Código del módulo preprogramado de encoder para NI myRIO

Otra opción es utilizar directamente estos bloques elementales, que se muestran en la Figura 18. Se decide utilizar esta segunda opción por mayor flexibilidad en el código y

por lo tanto, en general, tiempo más corto de ejecución, ya que no se realizan comprobaciones innecesarias en cada lectura, como si es preciso reiniciar el encoder, etc.

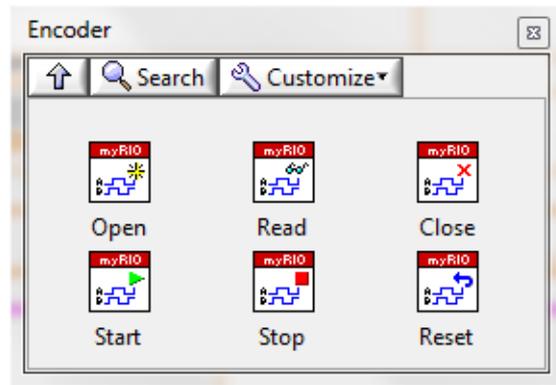


Figura 18. Bloques elementales de encoder para NI myRIO

Se utilizan los bloques “Open” y “Reset” para abrir la referencia a la entrada de encoder y reiniciar la misma al inicio de cada programa. Posteriormente, durante el bucle de control se utiliza “Read” para leer la señal de entrada. Finalmente, se usa “Close” para cerrar la referencia al final de programa.

Lo mismo ocurre en el caso de la escritura de la señal de control. Se desecha la opción de los módulos preprogramados para salidas analógicas de la tarjeta, para las que únicamente hay que seleccionar el canal, y se utilizan los bloques elementales disponibles para las mismas.

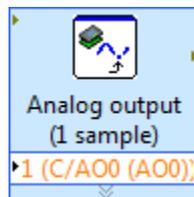


Figura 19. Módulo preprogramado de salida analógica para NI myRIO

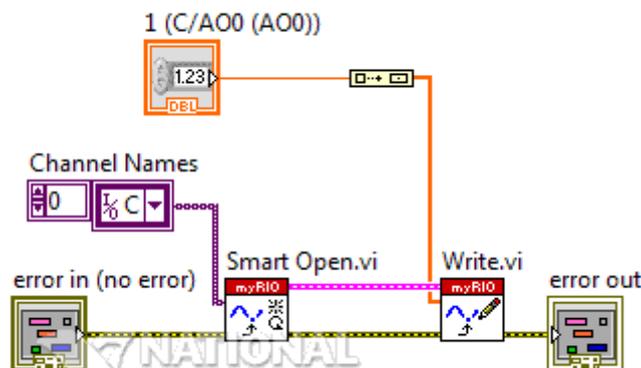


Figura 20. Código del módulo preprogramado de salida analógica para NI myRIO

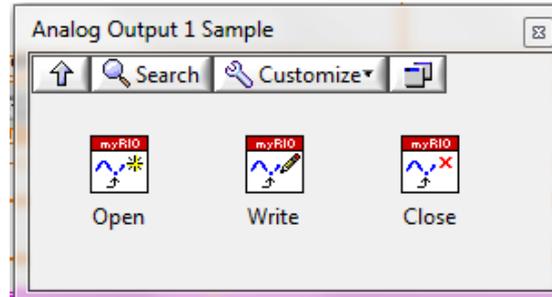


Figura 21. Bloques elementales de salida analógica para NI myRIO

En este caso, se utiliza “Open” para abrir la referencia al correspondiente puerto de salida analógica; luego, en el bucle de control, se utiliza “Write” para escribir el valor en Voltios en la salida analógica; después, al cerrar el programa, se utiliza “Write” de nuevo para escribir un voltaje nulo en la salida analógica y asegurar que no hay voltaje residual en la salida al finalizar el programa; finalmente se usa “Close” para cerrar la referencia.

### **5.6. ALGORITMOS DE AUTOERECCIÓN**

El término “autoerección” se refiere al movimiento de oscilación controlada del péndulo con el objetivo de, partiendo de su posición inferior de reposo, colocarlo hacia arriba, en la posición de péndulo invertido. Una vez en esa posición se produce una conmutación al modo de control de equilibrio de péndulo invertido mediante uno de los controladores anteriormente descritos.

Se barajan dos opciones para conseguir la autoerección: una basada en una excitación a la frecuencia natural del sistema carro-péndulo y otra basada en el control de la energía total del mismo.

La excitación del sistema con una onda senoidal a la frecuencia de resonancia es un algoritmo más sencillo, ya que se “puls” el sistema a una frecuencia que provoca oscilaciones crecientes en amplitud. Tiene, no obstante, el inconveniente de que no se controla fácilmente la velocidad angular del péndulo cuando éste alcanza la parte superior, por lo que, si esa velocidad es excesiva, será difícil que el algoritmo de control de equilibrio tome el control una vez producida la conmutación del modo de control.

La opción del control de la energía total es más compleja, pero permite conseguir que el péndulo llegue a la parte superior con una velocidad pequeña.

En el apartado 6. *METODOLOGÍA* se detallan estos métodos, las pruebas realizadas para elegir entre ambos y los respectivos resultados.

## **6. METODOLOGÍA**

A continuación, se presenta la WBS (Work Breakdown Structure) del proyecto, que incluye las tareas del mismo y da una idea de la forma en la que se va a proceder para llevarlo a cabo.

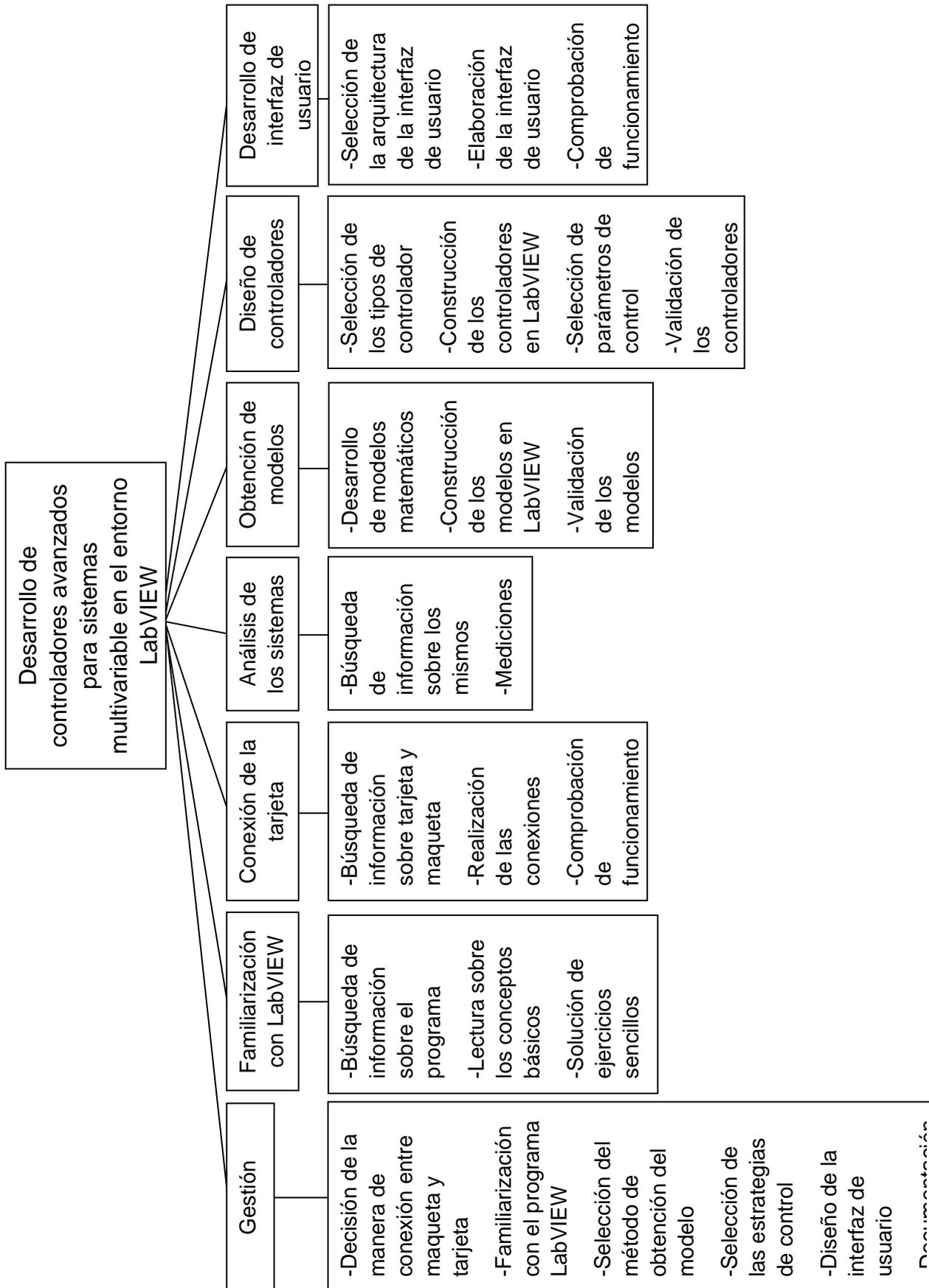
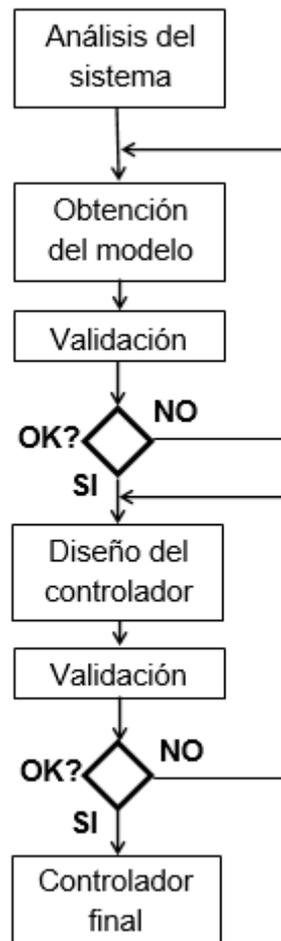


Figura 22. Work Breakdown Structure del proyecto

Los apartados “Análisis de los sistemas”, “Obtención de modelos” y “Diseño de controladores” se realizan para los tres sistemas de estudio. En concreto, para estas tres tareas del proyecto, se sigue el proceso que muestra la *Figura 23* siguiente. Dicho proceso es iterativo, ya que, tanto tras la obtención de un modelo como de un controlador, es necesario que éste sea validado. Si el resultado no es satisfactorio, se debe repetir el paso anterior; si, por el contrario, el resultado es satisfactorio, se avanza al siguiente paso.



*Figura 23. Proceso para “Análisis de los sistemas”, “Obtención de modelos” y “Diseño de controladores”*

Por un lado, para validar los modelos, se comparan la respuesta del modelo y la respuesta del sistema real ante la misma excitación en lazo abierto. Por otro lado, en el caso de los controladores, tras su prueba en el modelo, se validan en el sistema real, valorándose la respuesta de éste en cuanto a estabilidad y efectividad en el seguimiento a una referencia.

En este documento, no se detallan todos los intentos de validación, ya que son muy numerosos, debido al alto número de parámetros y la variedad de posibles valores para éstos; además, en la mayoría de casos, estas pruebas no aportan información relevante.

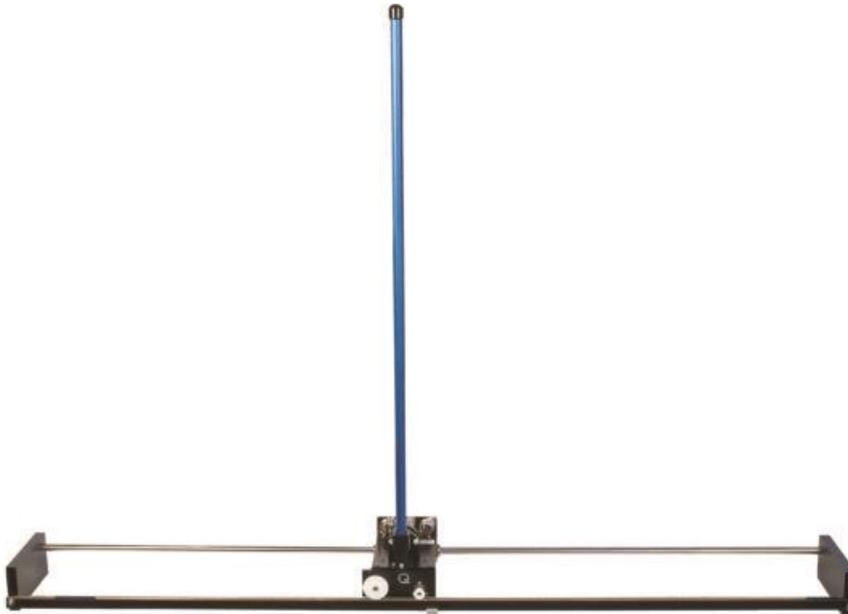
## **6.1. DESCRIPCIÓN DEL SISTEMA**

A continuación, se describen las distintas partes que componen el total del proyecto.

### **6.1.1. MAQUETA**

La maqueta desarrollada por Quanser, que sirve como soporte de los tres sistemas que se estudian en este documento, consta básicamente de una pista sobre la que puede moverse un carro accionado por un motor de corriente continua, y al que se pueden acoplar dos péndulos articulados de distinta longitud en un eje rotativo montado en el mismo, y perpendicular al mismo y a la pista. Además, se dispone de una masa que se puede añadir al carro.

La maqueta incluye dos encoders, uno para medida de la posición del carro y otro para medida de la posición del péndulo.



*Figura 24. Maqueta IP02 de Quanser*

En la *Figura 25* se muestran todos los componentes de la maqueta, enumerados y nombrados en la *Tabla 1*.

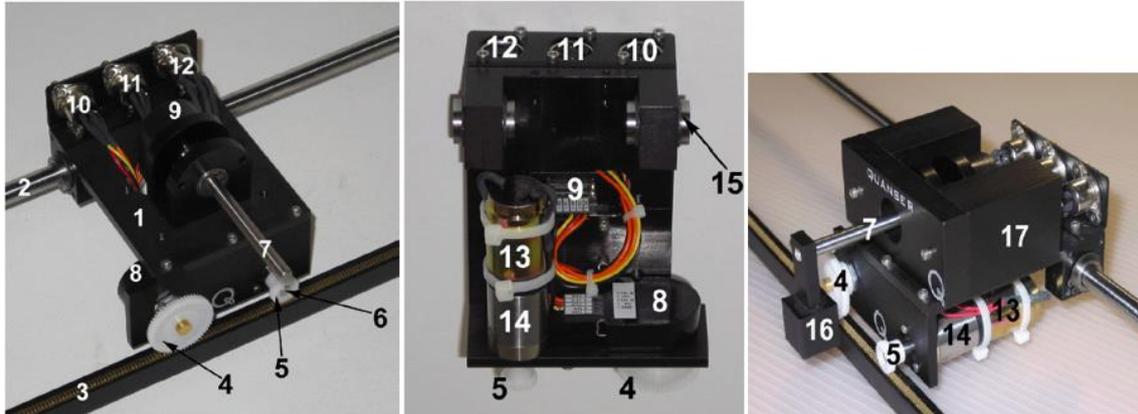


Figura 25. Componentes de la maqueta IP02 de Quanser

ID	DESCRIPCIÓN
1	Carro de la maqueta
2	Eje-guía de acero inoxidable
3	Cremallera
4	Piñón de accionamiento del encoder de posición del carro
5	Piñón motriz del carro
6	Eje del piñón motriz del carro
7	Eje de giro del péndulo
8	Encoder de posición del carro
9	Encoder de posición del péndulo
10	Conector del encoder de posición del carro
11	Conector del encoder de posición del péndulo
12	Conector de alimentación al motor de DC
13	Motor de DC
14	Motorreductor planetario
15	Cojinete de deslizamiento lineal
16	Soporte del péndulo
17	Masa supletoria del carro

Tabla 1. Lista de componentes de la maqueta IP02 de Quanser

Las características de los encoder y del motor de DC de la maqueta se describen en el siguiente apartado, en el que se detalla la conexión entre la maqueta y la tarjeta.

### 6.1.2. UNIVERSAL POWER MODULE (UPM)

Puesto que la señal de control procedente de la tarjeta NI myRIO no tiene la potencia (corriente) necesaria para mover el motor, se dispone de un módulo amplificador Universal Power Module (UPM) por el que pasa la señal de control para ser amplificada en potencia antes de llegar al motor. La resistencia del cable de conexión entre la UPM y el motor determina la amplificación en voltaje, que en este caso es de valor 1, es decir, el voltaje no varía. La salida de la UPM puede suministrar, sin embargo, una corriente de hasta 1A al motor.



Figura 26. Módulo UPM

### **6.1.3. TARJETA NI MYRIO**

La implementación del control en tiempo real y la adquisición de datos se realiza mediante una tarjeta NI myRIO de National Instruments, concretamente el modelo 1900.

Los programas de control se cargan desde el ordenador en esta tarjeta, que los ejecuta para el control de la maqueta. La conexión a la maqueta en lo que se refiere a las señales de control y de los sensores requiere de una adaptación de los conectores que se explica más adelante.



Figura 27. Tarjeta NI myRIO de National Instruments

## 6.1.4. ORDENADOR

Para llevar a cabo el proyecto, tanto el producto final como las pruebas pertinentes para su total desarrollo, se utiliza un ordenador Dell Optiplex 9020 con Windows 7. El ordenador se usa para desarrollar los programas de control y cargarlos posteriormente a la tarjeta NI myRIO para su ejecución en tiempo real. El ordenador permite, además, monitorizar los resultados, mediante la comunicación por variables entre ordenador y tarjeta, realizada a través de un cable USB (comunicación serie), que también podría hacerse mediante WiFi. Para estas tareas, se necesitan las aplicaciones software LabVIEW 2015, myRIO Toolkit 2015 y LabVIEW Runtime 2015 de National Instruments, que están disponibles en los discos de instalación que se incluyen con el suministro de la tarjeta NI myRIO.

## 6.1.5. ESQUEMA DE INTERCONEXIÓN

A continuación, se muestra el esquema de interconexión entre los distintos subsistemas que conforman el sistema general, ya explicados en sus respectivos apartados.

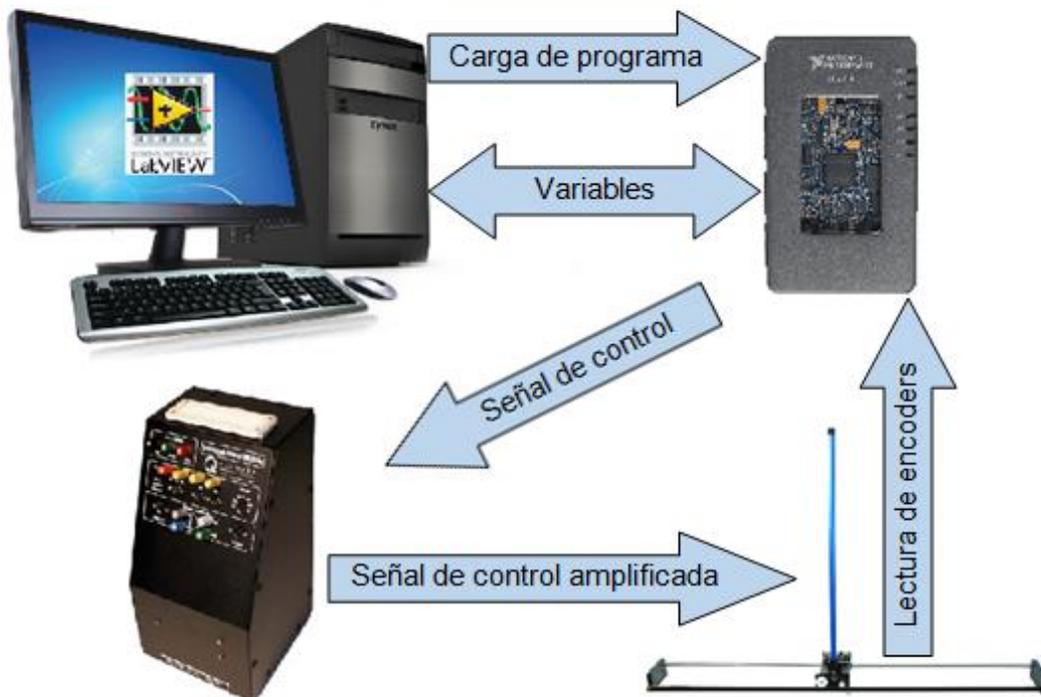


Figura 28. Esquema de interconexión de los sistemas

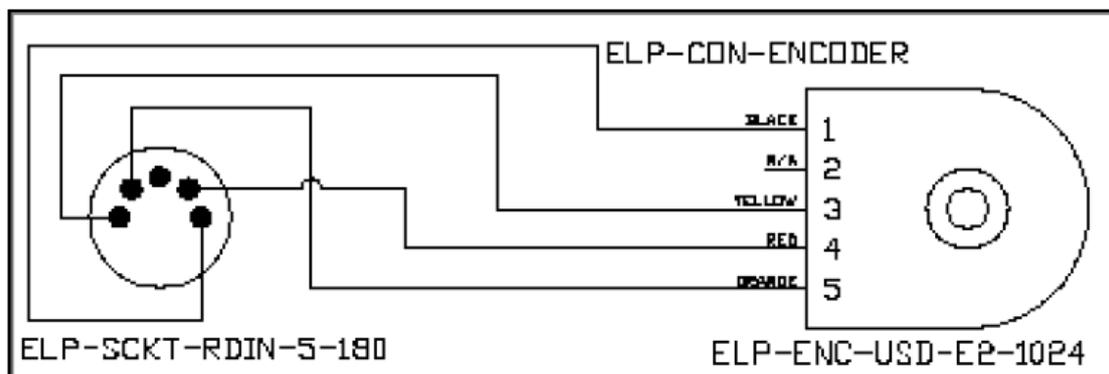
## 6.2. CONEXIONES ENTRE TARJETA Y MAQUETA

Dado que se utiliza la tarjeta NI myRIO para el desarrollo del proyecto, es necesario adaptar las conexiones de la maqueta al UPM y al ordenador, evitando así el uso de la tarjeta de adquisición de datos desarrollada por Quanser, que ya no es compatible con los puertos de datos de los ordenadores modernos.

Se requiere, por tanto, adaptar las conexiones de entrada de los dos encoders a la tarjeta NI myRIO y de la salida de la misma la señal de control al UPM que gobierna el motor DC.

### 6.2.1. ENCODERS

Los encoders presentes en la maqueta son el modelo ELP-ENC-USD-E2-1024 de US Digital, y presentan la configuración de cables y conector mostrados en la *Figura 29* y la *Tabla 2*.



*Figura 29. Cableado de los encoders*

Cable	Color
Tierra	Negro
Canal A (incremento)	Amarillo
+ 5V	Rojo
Canal B (decremento)	Naranja

*Tabla 2. Identificación de los cables de los encoders*

Por lo tanto, de los cuatro pines del conector, dos se utilizan para la alimentación a 5 V del encoder y los otros dos para incrementos y decrementos en el conteo.

Para la conexión a la tarjeta, se adaptan conectores hembra de cinco pines, como los presentes en la tarjeta original de Quanser. De estos cinco pines del conector hembra, se sacan cables de los cuatro correspondientes a las funciones del encoder previamente mencionadas, que se conectan a la tarjeta mediante conector de bornas de tornillo. Las conexiones se detallan en el apartado 6.2.3. *DETALLES DE LAS CONEXIONES*.

### 6.2.2. MOTOR

El motor de corriente continua encargado del movimiento del carro es un Faulhaber Coreless DC Motor (2338S006), que se alimenta mediante una señal de corriente continua que proporciona el UPM tras amplificar la señal de control analógica de su entrada.

La tensión nominal permanente del motor es de  $\pm 6$  V, admitiendo picos de tensión mayores durante breves instantes. No obstante, la señal enviada al motor no debe exceder el valor de  $\pm 15$  V, para evitar daños en el mismo.

Para transmitir dicha señal, se utiliza un conector coaxial macho, al que se conecta su correspondiente conector hembra cuyos cables de salida se conectan a la tarjeta NI myRIO, tierra analógica y tensión activa analógica, la cual puede tomar tensión positiva o negativa respecto a tierra, permitiendo así el giro del motor en ambos sentidos.

### 6.2.3. DETALLES DE LAS CONEXIONES

CONECTOR	COLOR DE CABLE	FUNCIÓN	PIN EN NI myRIO
Entradas de pulsos de encoders  	Amarillo	Entrada "A" de pulsos de encoder	ENCX.A
	Marrón	Entrada "B" de pulsos de encoder	ENCX.B
	Rojo	Positivo "5V" alimentación de encoder	5V
	Negro	Negativo alimentación de encoder	DGND
Salida analógica para control de motor  	Rojo	Activo de Salida analógica	AO0
	Negro	Tierra de Salida analógica	AGND

Tabla 3. Detalles de las conexiones

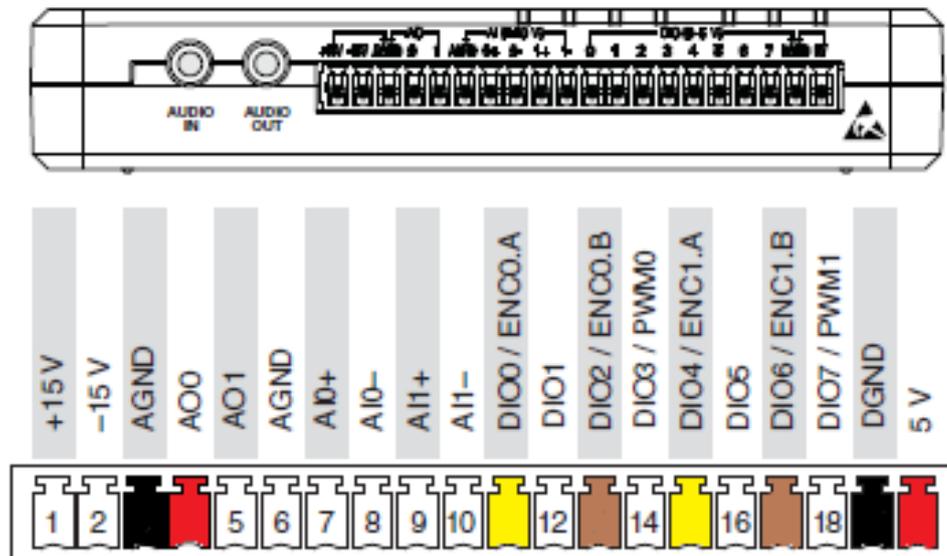


Figura 30. Conexión en los pines de NI myRIO

En la *Tabla 3* se detalla la función de cada cable en los conectores y el pin de la tarjeta NI myRIO en la que se conecta, mientras que en la *Figura 30* se ubican dichos pines en el conector C de dicha tarjeta. Cabe destacar que ambos encoders comparten los pines de alimentación (19 y 20). En la entrada “encoder 0” se conecta el encoder que determina la posición horizontal del carro, y en la entrada “encoder 1” el que determina la posición angular del péndulo.

El cableado no se realiza directamente en los pines de la tarjeta, sino en un mini-borneo extraíble, equipado con bornas de tornillo, como el que se puede ver en la *Figura 31*, permitiéndose así el traslado de las conexiones a otra tarjeta NI myRIO simplemente insertando el mini-borneo extraíble en el conector C de la nueva tarjeta.

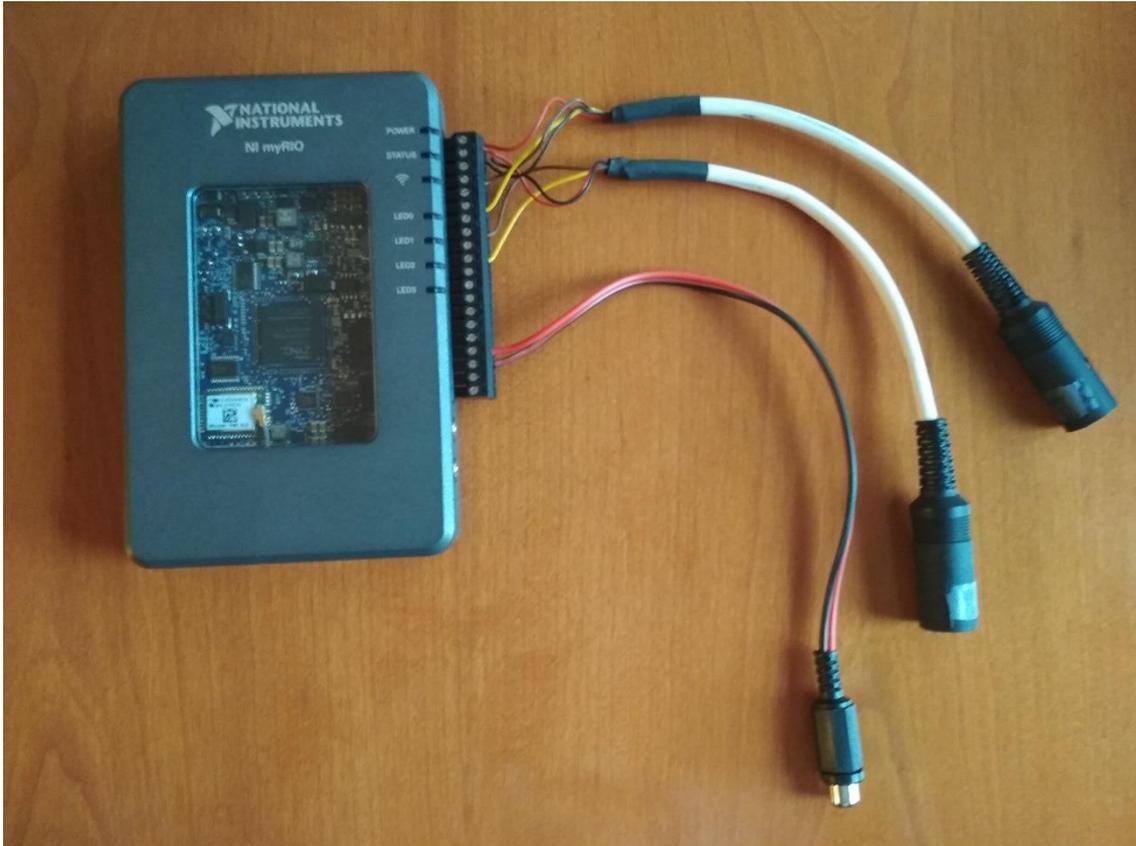


Figura 31. Tarjeta NI myRIO con las conexiones realizadas

#### **6.2.4. PRUEBA DE LAS CONEXIONES**

Para realizar una prueba de las conexiones y comprobar la correcta lectura de las entradas y salidas de la tarjeta, se desarrolla un programa simple en LabVIEW utilizando los bloques de función de la herramienta myRIO Toolkit para salida analógica y encoder.

En el caso de la salida analógica, únicamente se configura el bloque seleccionando de qué pin de la tarjeta se va a obtener dicha salida, mientras que para los encoders sólo se selecciona que sean de cuadratura y, posteriormente, qué entradas de las reservadas para dicho fin en la tarjeta se van a utilizar.

Aunque se usan los bloques preprogramados por National Instruments para esta prueba, ya se ha explicado que en el desarrollo de la aplicación se usan los bloques de funciones elementales disponibles para los encoders.

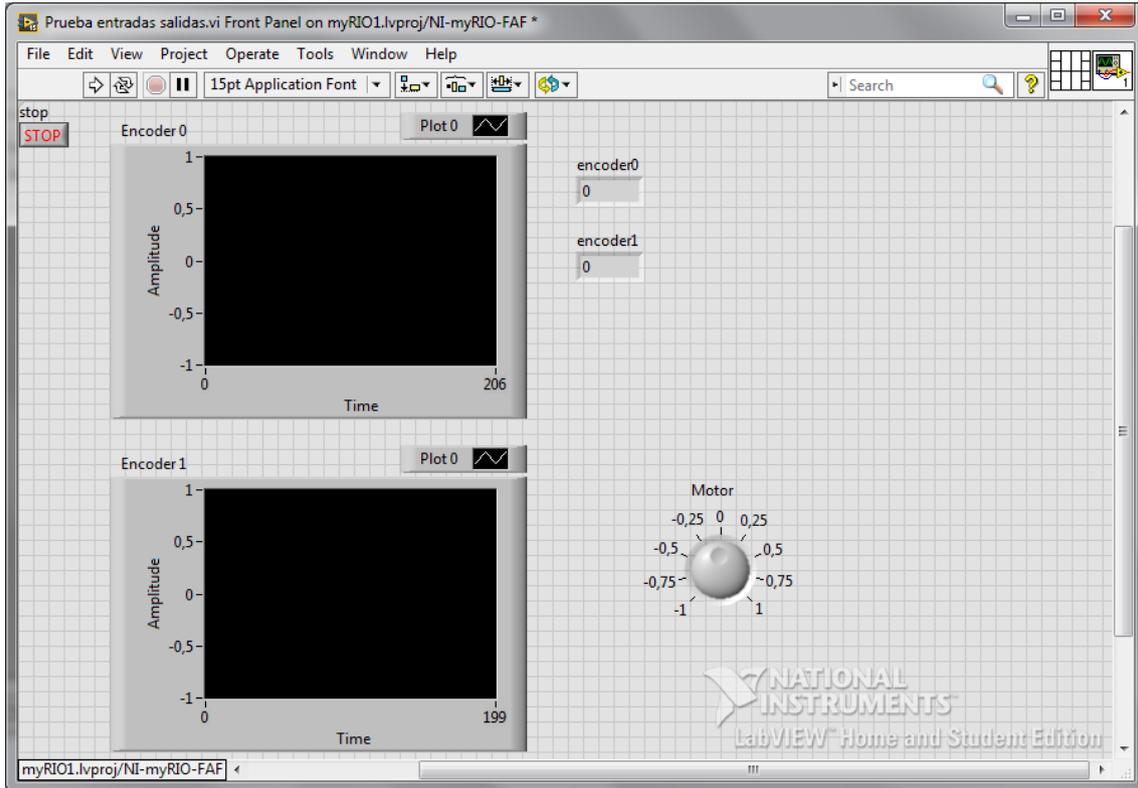


Figura 32. Front Panel del VI de prueba de las conexiones

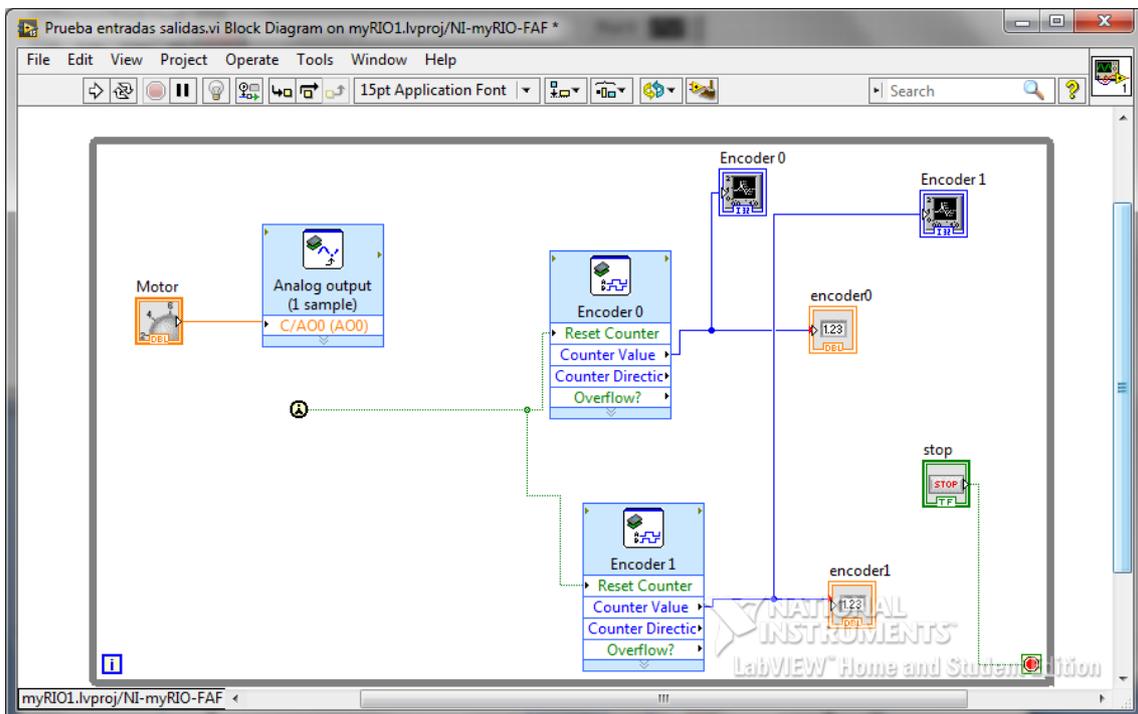


Figura 33. Código del VI de prueba de las conexiones

Mediante una ruleta del interfaz gráfico se modifica la señal de control enviada al motor entre -1 y 1 V de DC y se comprueba que el carro se mueve correctamente en ambos sentidos. Además, moviendo tanto el carro como el eje del péndulo en ambos sentidos de giro, se comprueba el correcto funcionamiento de los dos encoders.

### **6.3. VARIABLES PRINCIPALES DEL SISTEMA Y RANGOS**

Tras mediciones en la maqueta, se determinan los siguientes rangos y referencias de las variables principales del sistema:

<b>Magnitud</b>	<b>Valor</b>	<b>Unidad</b>
Longitud de pista útil	843	mm
Longitud del carro	95	mm
Carrera útil	748	mm

*Tabla 4. Variables del sistema*

Utilizando el programa desarrollado para probar las entradas y salidas de la tarjeta, se determina el valor de los encoders para las posiciones de interés, correspondiéndose para el encoder del carro (encoder 0) con los dos extremos de su carrera útil tomando como referencia el punto medio de la misma. En el caso del encoder del péndulo (encoder 1), y tomando como referencia la posición vertical inferior del mismo, se comprueba la correcta lectura cuando el péndulo alcanza la posición vertical superior en ambos sentidos de giro.

<b>Encoder</b>	<b>Valores en unidad</b>	<b>Valores en incrementos</b>	<b>Ganancia</b>
Encoder 0	[-374, 374] mm	[16445,-16445]	-43.9706 pulsos/mm
Encoder 1	$[-\pi, \pi]$ rad	[2048,-2048]	-651.8986 pulsos/rad

*Tabla 5. Caracterización de los encoders*

En la *Tabla 5* se muestra la correspondencia entre unidades de la magnitud medida por los encoders y los valores en cuentas que éstos suministran, obteniéndose además valores de interés concretos como los límites de trabajo del encoder 0 y el valor de la posición deseada en el péndulo invertido. Los signos negativos en la ganancia significan que el encoder cuenta en el sentido opuesto al establecido como positivo en el modelo.

El resto de parámetros del sistema se encuentran disponibles en *ANEXO II: PARÁMETROS DEL SISTEMA*, al final de este documento.

## 6.4. MODELOS TEÓRICOS

En este apartado se explica el desarrollo seguido para la obtención de los modelos matemáticos de los tres sistemas de estudio. Para un estudio más detallado, se puede consultar el ANEXO I: MODELOS TEÓRICOS de este documento.

### 6.4.1. MODELO DEL MOTOR Y EL CARRO

Se calcula el modelo matemático que relaciona el voltaje aplicado al motor de corriente continua en el carro con la fuerza resultante sobre el conjunto motor-carro para desplazarse por la pista.

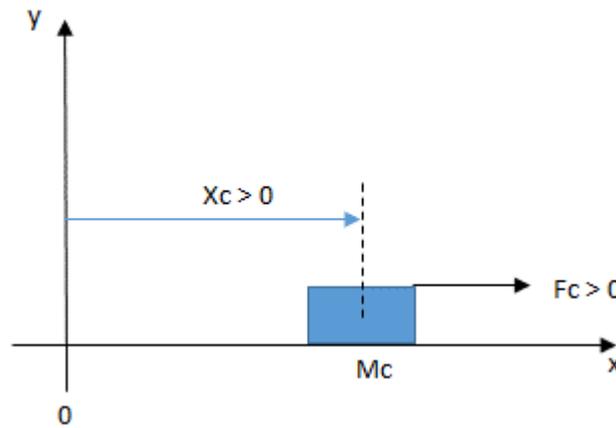


Figura 34. Esquema del modelo del motor y el carro

Aplicando balance de fuerzas al movimiento del carro se obtiene la ecuación (17)

$$F_c(t) - B_{eq} \dot{x}_c(t) - F_{ai}(t) = M_c \ddot{x}_c(t) \quad (17)$$

donde  $F_c(t)$  es la fuerza de tracción en el eje del piñón motriz del carro,  $B_{eq}$  el coeficiente de rozamiento viscoso,  $F_{ai}(t)$  la fuerza inercial del rotor,  $M_c$  la masa del carro y  $x_c(t)$  la posición del carro respecto a su posición inicial.

La expresión de la fuerza inercial del rotor tiene la siguiente forma:

$$F_{ai}(t) = \frac{\eta_g K_g^2 J_m \ddot{x}_c(t)}{r_{mp}^2} \quad (18)$$

Siendo  $K_g$  la relación del reductor,  $\eta_g$  el rendimiento del reductor,  $r_{mp}$  el radio del piñón motriz y  $J_m$  el momento de inercia del rotor.

Por otro lado, la ecuación que rige el comportamiento del motor es la que se indica a continuación:

$$V_m(t) - R_m I_m(t) - L_m \frac{dI_m(t)}{dt} - E_{mf}(t) = 0 \quad (19)$$

donde  $V_m(t)$  es la tensión aplicada al motor,  $R_m$  la resistencia de su devanado,  $I_m(t)$  la intensidad que circula por él,  $L_m$  su inductancia y  $E_{mf}(t)$  la fuerza contraelectromotriz. Como en el caso del motor de la maqueta la inductancia es mucho menor que la resistencia, el término que acompaña a  $L_m$  se desprecia, quedando así una expresión más sencilla que permite despejar  $I_m(t)$  de una manera más simple.

$$V_m(t) - R_m I_m(t) - E_{mf}(t) = 0 \quad (20)$$

$$I_m(t) = \frac{V_m(t) - E_{mf}(t)}{R_m} = \frac{V_m(t) - K_m \omega_m(t)}{R_m} \quad (21)$$

La fuerza de tracción viene dada por la ecuación (22), mientras que el par motor  $T_m(t)$ , viene dado por la ecuación (23), mediante el rendimiento del motor  $\eta_m$ , la constante par-corriente  $K_t$  y la intensidad del motor.

$$F_c(t) = \frac{\eta_g K_g T_m(t)}{r_{mp}} \quad (22)$$

$$T_m(t) = \eta_m K_t I_m(t) \quad (23)$$

De las ecuaciones (21), (22) y (23) se obtiene la expresión:

$$F_c(t) = \frac{\eta_g \eta_m K_t K_g [V_m(t) - K_m \omega_m(t)]}{r_{mp} R_m} \quad (24)$$

Como, además, se sabe que

$$\omega_m(t) = \frac{K_g \dot{x}_c(t)}{r_{mp}} \quad (25)$$

La expresión final de la fuerza de tracción en función del voltaje aplicado es

$$F_c(t) = \frac{\eta_g \eta_m K_t K_g [r_{mp} V_m(t) - K_m K_g \dot{x}_c(t)]}{r_{mp}^2 R_m} \quad (26)$$

Sustituyendo las expresiones de  $F_c(t)$  y  $F_{ai}(t)$  en la ecuación original y operando:

$$\frac{\eta_g \eta_m K_t K_g V_m(t)}{r_{mp} R_m} - \left[ \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} + B_{eq} \right] \dot{x}_c(t) - \left[ \frac{\eta_g K_g^2 J_m}{r_{mp}^2} + M_c \right] \ddot{x}_c(t) = 0 \quad (27)$$

Aplicando a la expresión (27) transformada de Laplace y reordenando los términos se llega a la siguiente expresión

$$\frac{\eta_g \eta_m K_t K_g}{r_{mp} R_m} V_m(s) = \left[ \frac{\eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m}{r_{mp}^2 R_m} s + \frac{\eta_g K_g^2 J_m + r_{mp}^2 M_c}{r_{mp}^2} s^2 \right] x_c(s) \quad (28)$$

Por lo que entre la alimentación de tensión al motor y la posición del carro existe la siguiente función de transferencia:

$$G(s) = \frac{x_c(s)}{V_m(s)} = \frac{\eta_g \eta_m K_t K_g r_{mp}}{s \left[ (\eta_g K_g^2 J_m R_m + r_{mp}^2 M_c R_m) s + \eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m \right]} \quad (29)$$

Ésta es la expresión que caracteriza el primer sistema de estudio, el carro. El sistema también puede ser representado por sus ecuaciones de estado, a partir de la ecuación (27).

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \dot{x}_c(t) \end{bmatrix}, u = V_m(t) \text{ e } y = x_c(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{(\eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m)}{R_m (\eta_g K_g^2 J_m + r_{mp}^2 M_c)} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{(\eta_g \eta_m K_t K_g) r_{mp}}{R_m (\eta_g K_g^2 J_m + r_{mp}^2 M_c)} \end{bmatrix}$$

$$C = [1 \quad 0] \quad (30)$$

Se valora añadir al modelo matemático un tiempo muerto, que podría generarse debido a la fricción estática entre el carro y la pista. Sin embargo, tras pruebas en el sistema se determina que dicho tiempo muerto es despreciable, por lo que finalmente no es introducido.

#### 6.4.2. MODELO DEL PÉNDULO GANTRY

En el caso de la inclusión del péndulo Gantry en el carro, el nuevo modelo se obtiene partiendo de las expresiones de la fuerza y la posición del carro calculadas anteriormente.

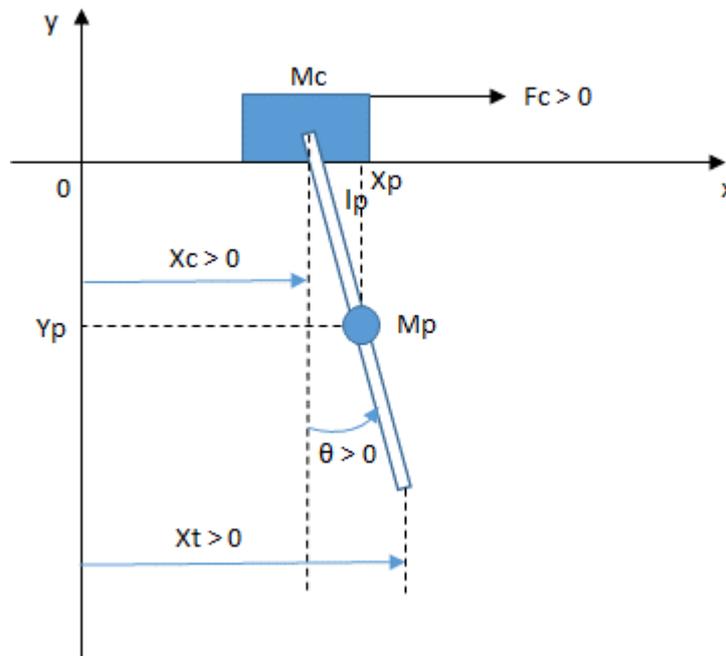


Figura 35. Esquema del modelo del péndulo Gantry

En la Figura 35 se representa el sistema de forma simplificada, mostrando las variables de salida del mismo, el ángulo  $\theta$  del péndulo y la posición del carro  $x_c$ , la masa del péndulo  $M_p$ , la distancia  $l_p$  desde la articulación hasta su centro de gravedad, así como sus coordenadas  $x_p$  e  $y_p$ .

Para plantear el modelo del sistema se utilizan las ecuaciones de Euler-Lagrange (31), donde  $q_i$  son las variables independientes del sistema, en este caso  $\theta$  y  $x_c$ ;  $L$  el Lagrangiano, que se define, según la ecuación (32), como la resta entre la energía cinética total del sistema  $T_T$  y la energía potencial total del sistema  $V_T$ ; y  $Q_i$  son las

fuerzas generalizadas, que son aquellas fuerzas externas que pueden producir un cambio en la coordenada  $q_i$ .

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (31)$$

$$L = T_T - V_T \quad (32)$$

En el caso del sistema de estudio, las ecuaciones son las de la expresión (33), donde  $B_p$  es el coeficiente de rozamiento viscoso de la articulación del péndulo.

$$\begin{cases} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_c} \right) - \frac{\partial L}{\partial x_c} = Q_{x_c} = F_c(t) - B_{eq} \dot{x}_c(t) \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q_{\theta} = -B_p \dot{\theta}(t) \end{cases} \quad (33)$$

Dado que el carro se mueve en un plano horizontal, se toma como referencia dicho plano para la energía potencial, por lo que sólo el péndulo contribuye a ésta, siendo  $g$  la constante de gravedad en la superficie terrestre.

$$V_T = -M_p g l_p \cos \theta(t) \quad (34)$$

La energía cinética del sistema es la suma de la energía cinética del carro  $T_c$  y la del péndulo  $T_p$ . En ambos casos, hay un término debido a la traslación y otro a la rotación, en el caso del carro debido al motor. Su expresión final es la siguiente:

$$\begin{aligned} T_T = \frac{1}{2} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \dot{x}_c(t)^2 + M_p l_p \cos \theta(t) \dot{\theta}(t) \dot{x}_c(t) \\ + \frac{1}{2} (M_p l_p^2 + J_p) \dot{\theta}(t)^2 \end{aligned} \quad (35)$$

Sustituyendo las derivadas parciales del Lagrangiano en las ecuaciones de Euler-Lagrange (31) y operando se obtiene el siguiente sistema de ecuaciones.

$$\begin{cases} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \ddot{x}_c(t) + (M_p l_p \cos \theta(t) \ddot{\theta}(t) - M_p l_p \sin \theta(t) \dot{\theta}(t)^2) + B_{eq} \dot{x}_c(t) = F_c(t) \\ M_p l_p \cos \theta(t) \ddot{x}_c(t) + (M_p l_p^2 + J_p) \ddot{\theta}(t) + M_p g l_p \sin \theta(t) + B_p \dot{\theta}(t) = 0 \end{cases} \quad (36)$$

Además se linealiza la expresión (36) en torno al punto de equilibrio, es decir, para  $\theta(t) \simeq 0$ , lo que implica  $\cos \theta(t) \simeq 1$  y  $\sin \theta(t) \simeq \theta(t)$ . Si, adicionalmente, se desprecia el término de orden superior ( $\dot{\theta}(t)^2$ ) porque la velocidad angular del péndulo es muy pequeña en torno al punto de equilibrio se obtiene la expresión (37).

$$\begin{cases} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \ddot{x}_c(t) + M_p l_p \ddot{\theta}(t) + B_{eq} \dot{x}_c(t) = F_c(t) \\ M_p l_p \ddot{x}_c(t) + (M_p l_p^2 + J_p) \ddot{\theta}(t) + M_p g l_p \theta(t) + B_p \dot{\theta}(t) = 0 \end{cases} \quad (37)$$

Despejando en la segunda ecuación de (37)  $\ddot{\theta}(t)$  y sustituyendo la expresión resultante en la primera ecuación de (37), despejando y simplificando  $\ddot{x}_c(t)$  se llega a (39).

$$\ddot{\theta}(t) = \left( -\frac{M_p l_p}{M_p l_p^2 + J_p} \right) \ddot{x}_c(t) - \left( \frac{M_p g l_p \theta(t) + B_p \dot{\theta}(t)}{M_p l_p^2 + J_p} \right) \quad (38)$$

$$\ddot{x}_c(t) = \frac{(M_p l_p^2 + J_p) F_c(t) + M_p^2 g l_p^2 \theta(t) + M_p l_p B_p \dot{\theta}(t) - (M_p l_p^2 + J_p) B_{eq} \dot{x}_c(t)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \quad (39)$$

Ya que la expresión (39) no depende de  $\ddot{\theta}(t)$ , si se sustituye en la (38) y se simplifica, se obtiene la ecuación (40), con lo que las derivadas segundas de las dos variables de salida del sistema se encuentran desacopladas.

$$\ddot{\theta}(t) = \frac{-\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p \theta(t) + M_p l_p B_{eq} \dot{x}_c(t) - \left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) B_p \dot{\theta}(t) - M_p l_p F_c(t)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \quad (40)$$

A partir de las expresiones (39) y (40) se pueden deducir las ecuaciones de estado del sistema, que tendrán la siguiente forma y los valores que se detallan a continuación.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \theta(t) \\ \dot{x}_c(t) \\ \dot{\theta}(t) \end{bmatrix}, u = F_c(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Siendo,

$$a_{32} = \frac{M_p^2 g l_p^2}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{33} = -\frac{\left(M_p l_p^2 + J_p\right) B_{eq}}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{34} = \frac{M_p l_p B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{42} = -\frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{43} = \frac{M_p l_p B_{eq}}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{44} = - \frac{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) B_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{(M_p l_p^2 + J_p)}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\ \frac{M_p l_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (41)$$

El modelo de ecuaciones de estado representado en (41) tiene como salidas del sistema las deseadas, pero la fuerza del carro es la entrada al mismo, mientras que es deseable tener la tensión aplicada en el motor como entrada, ya que será la acción de control que se introduzca al sistema. Por lo tanto, sustituyendo la expresión (24) en (39) y (40), se obtienen las expresiones de  $\dot{x}_c(t)$  y  $\ddot{\theta}(t)$  en función de la tensión del motor.

Si se calculan en este caso, las ecuaciones de estado quedan como se muestra a continuación.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \quad \text{donde } x = \begin{bmatrix} x_c(t) \\ \theta(t) \\ \dot{x}_c(t) \\ \dot{\theta}(t) \end{bmatrix}, u = V_m(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Siendo,

$$a_{32} = \frac{M_p^2 g l_p^2}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2}$$

$$a_{33} = - \frac{(M_p l_p^2 + J_p) \left( B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} \right)}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2}$$

$$a_{34} = \frac{M_p l_p B_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2}$$

$$\begin{aligned}
 a_{42} &= - \frac{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) M_p g l_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{43} &= \frac{M_p l_p \left( B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} \right)}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{44} &= - \frac{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) B_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 B &= \begin{bmatrix} 0 \\ 0 \\ \frac{\left( M_p l_p^2 + J_p \right) \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[ \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2 \right]} \\ \frac{M_p l_p \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[ \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2 \right]} \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (42)
 \end{aligned}$$

Aunque tanto el sistema péndulo invertido como el sistema péndulo Gantry corresponden a sistemas SIMO, Single Input ( $V_m(t)$ ) y Multiple Output ( $x_c(t)$  y  $\theta(t)$ ), en el caso del péndulo Gantry se simula el movimiento de un puente-grúa en el que el objetivo es minimizar las oscilaciones en la carga, es decir, en el extremo del péndulo.

Por ello, la variable a controlar es  $x_t(t)$  de la *Figura 35*:

La salida que interesa, por tanto, no es  $y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$ , sino  $y = x_t(t)$ , tratando por lo tanto el sistema como SISO, Single Input ( $V_m(t)$ ), Single Output ( $x_t(t)$ ).

De acuerdo a lo anterior se puede calcular la nueva matriz C para  $y = x_t(t)$  La variable  $x_t(t)$  se define según la ecuación (28).

$$x_t(t) = x_c(t) + 2 l_p \sin \theta(t) \quad (43)$$

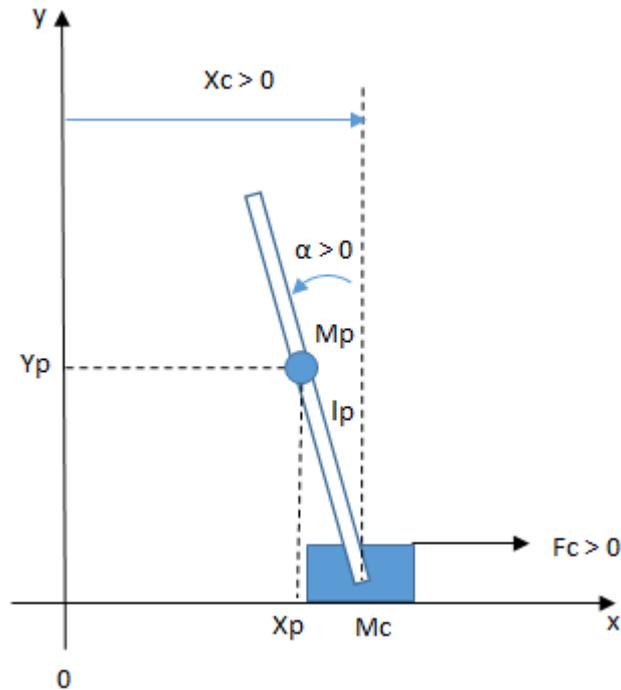
Linealizando en torno al punto de equilibrio, se puede asumir que el ángulo es igual al seno del mismo, por lo que se obtiene la expresión (44), a partir de la cual se deduce C.

$$x_t(t) = x_c(t) + 2 l_p \theta(t) \quad (44)$$

$$C = [1 \quad 2 l_p \quad 0 \quad 0] \quad (45)$$

### 6.4.3. MODELO DEL PÉNDULO INVERTIDO

En el caso del péndulo invertido, el planteamiento es el que se muestra en la *Figura 36*, análogo al caso anterior pero con una referencia distinta para el ángulo, que en este caso se denomina  $\alpha(t)$ .



*Figura 36. Esquema del modelo del péndulo invertido*

Por lo tanto, desarrollando las ecuaciones de la misma manera que en el caso del péndulo Gantry, se obtiene el modelo para el caso del péndulo invertido, dado por el siguiente sistema de ecuaciones de estado. Se puede apreciar que únicamente cambian algunos signos de los términos con respecto al modelo anterior, debido al cambio en la referencia del ángulo.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \alpha(t) \\ \dot{x}_c(t) \\ \dot{\alpha}(t) \end{bmatrix}, u = V_m(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \alpha(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Donde,

$$a_{32} = \frac{M_p^2 g l_p^2}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$\begin{aligned}
 a_{33} &= - \frac{(M_p l_p^2 + J_p) \left( B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} \right)}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{34} &= - \frac{M_p l_p B_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{42} &= \frac{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) M_p g l_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{43} &= - \frac{M_p l_p \left( B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} \right)}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 a_{44} &= - \frac{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) B_p}{\left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2} \\
 B &= \begin{bmatrix} 0 \\ 0 \\ \frac{(M_p l_p^2 + J_p) \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[ \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2 \right]} \\ \frac{M_p l_p \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[ \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) J_p + \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) M_p l_p^2 \right]} \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (46)
 \end{aligned}$$

#### 6.4.4. MODIFICACIONES SOBRE EL MODELO MATEMÁTICO

Como ya se ha comentado anteriormente, la posibilidad de añadir un tiempo muerto al sistema se descarta. Sin embargo, al modelo con el que se simula el sistema se le añade una banda muerta, es decir, un rango de voltaje alrededor de cero para el cual el sistema no responde, debido probablemente a la fricción del piñón con la cremallera de la pista.

Para determinar esta zona muerta se aplica una rampa de voltaje al sistema carro, positiva en un caso y negativa en el otro, registrándose la respuesta  $x_c(t)$  del carro.

Como muestran las siguientes figuras, la zona muerta en el caso de voltaje positivo se extiende hasta 0,26 V, es decir, que entre 0V y 0,26V el carro no se mueve.

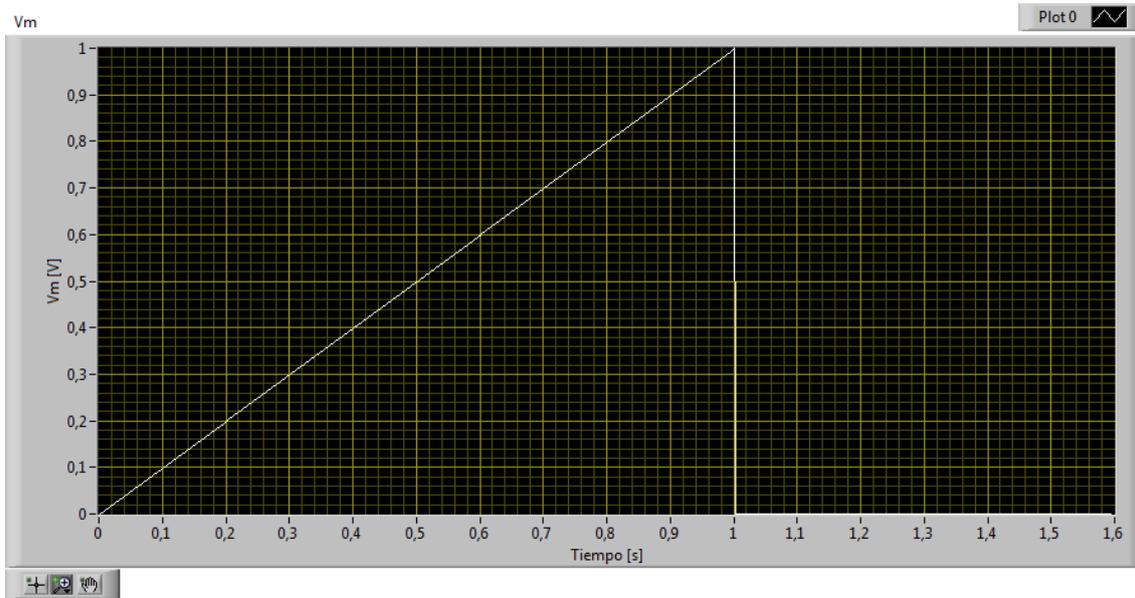


Figura 37. Tensión aplicada al carro

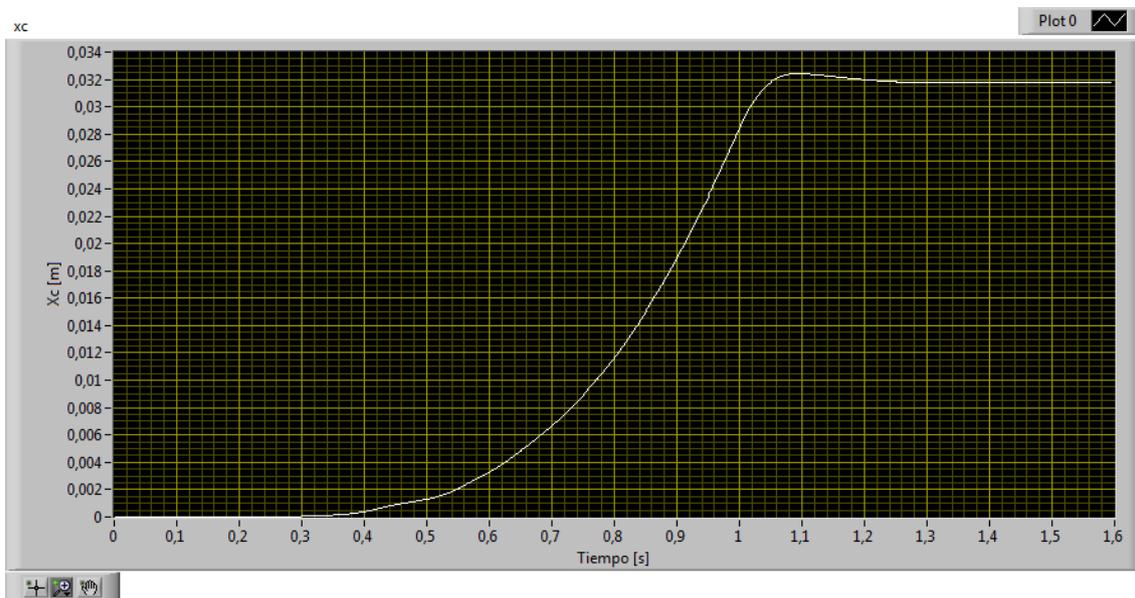


Figura 38. Desplazamiento del carro

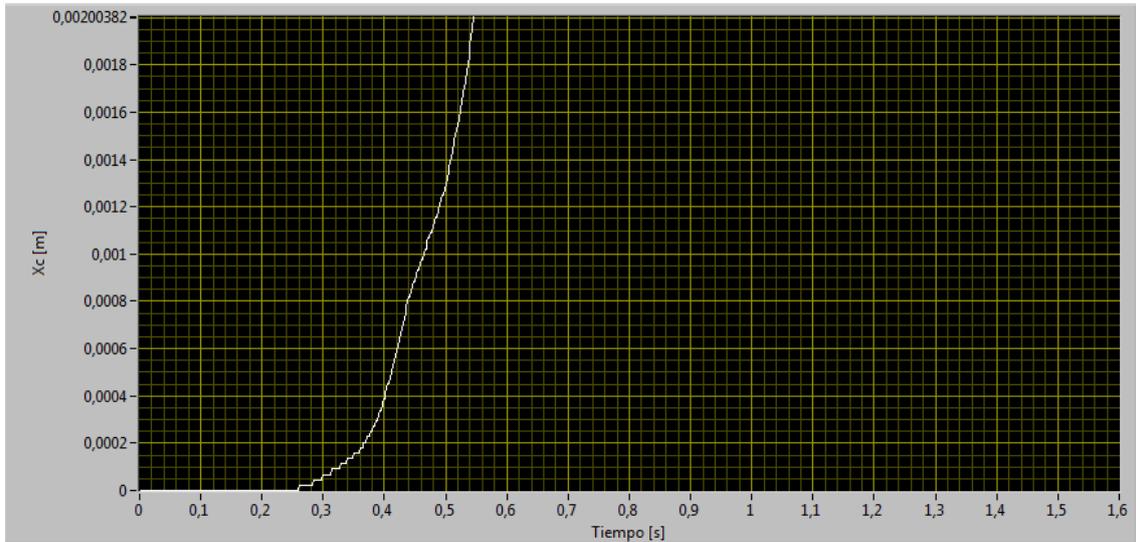


Figura 39. Detalle del desplazamiento del carro

En las siguientes figuras, se puede observar que para voltajes negativos la zona muerta se extiende hasta -0,1V.

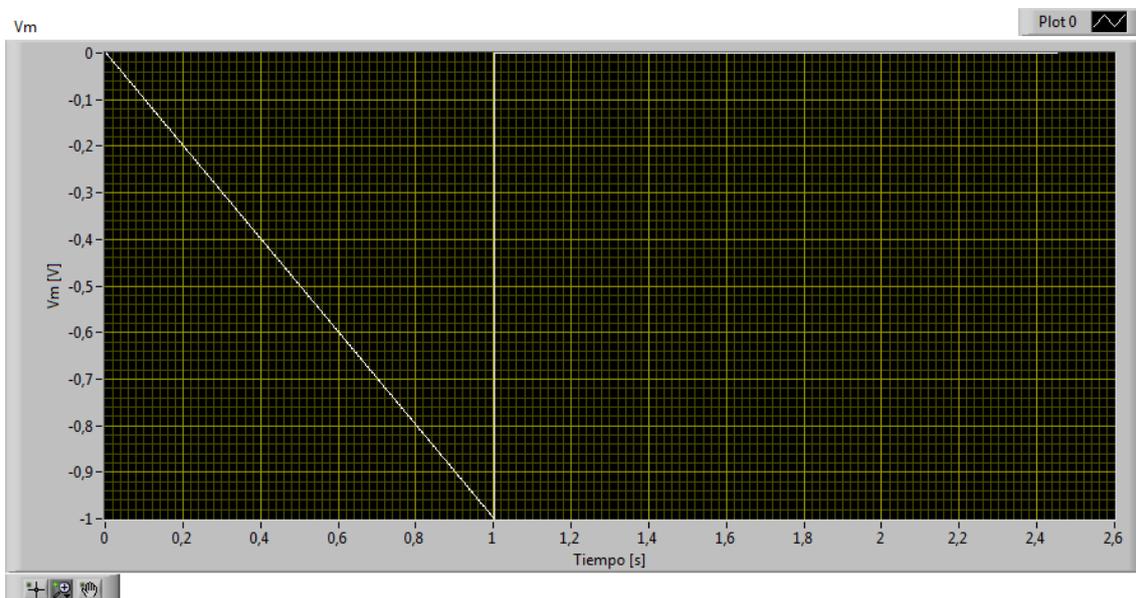


Figura 40. Voltaje aplicado al carro

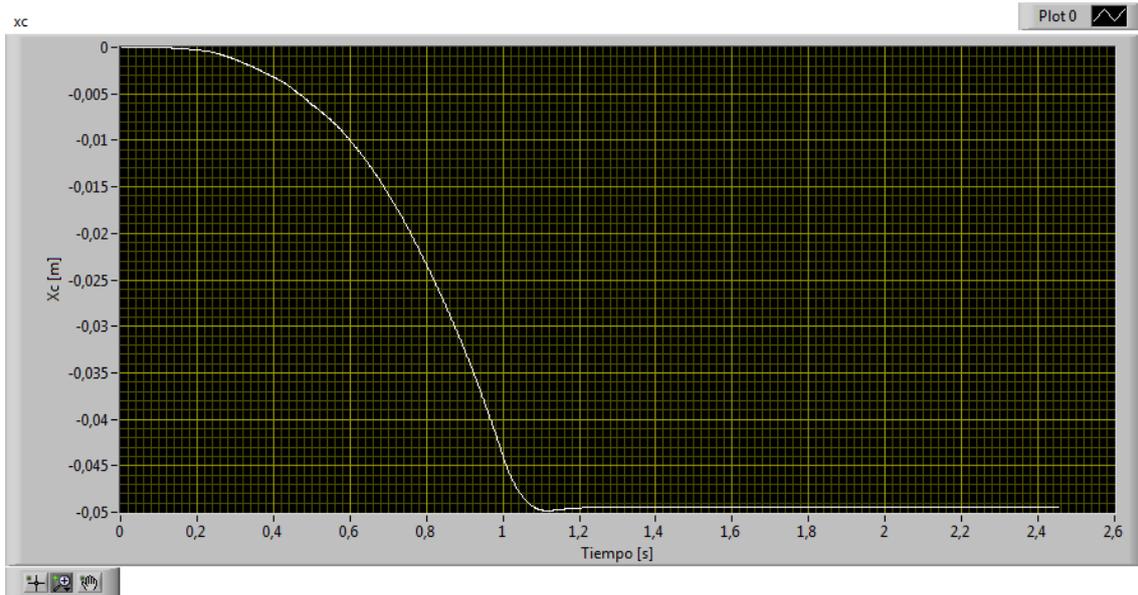


Figura 41. Desplazamiento del carro

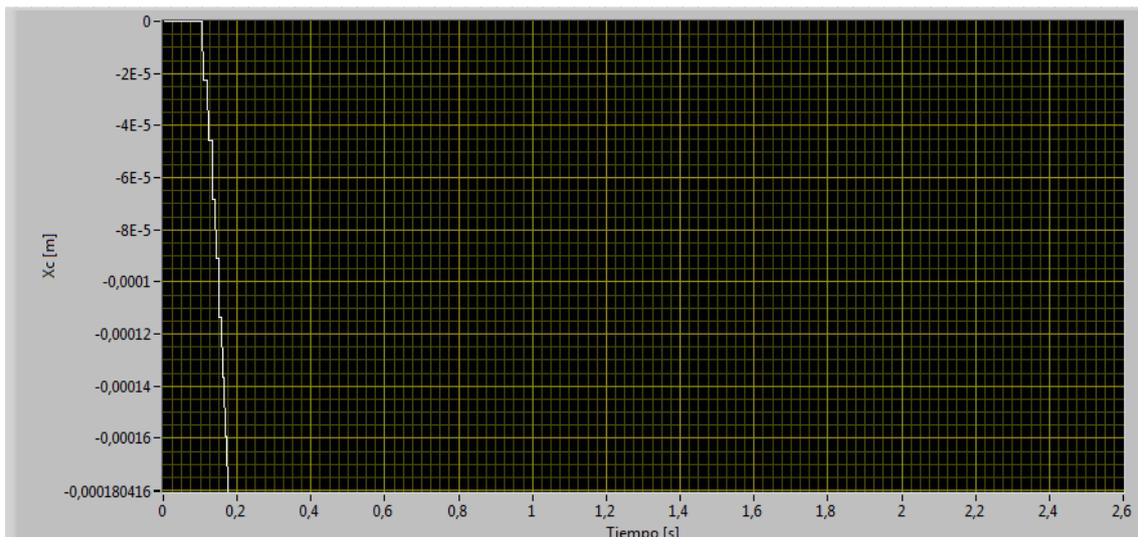


Figura 42. Detalle del desplazamiento del carro

Sin embargo, teniendo en cuenta que en diferentes pruebas se obtienen valores ligeramente diferentes, por uniformidad se decide ubicar en el modelo la zona muerta entre los límites -0,25V y 0,25V.

Por otro lado, aunque los valores de los parámetros de la maqueta IP02 vienen dados por Quanser, tras realizar pruebas en el sistema real, se validan los modelos modificando algunos parámetros de la misma, sobre todo aquellos difíciles de medir como rozamientos, rendimientos..., etc. Para un listado detallado de los parámetros, tanto ajustados como sin ajustar (originales de Quanser, refiérase al ANEXO II: PARÁMETROS DEL SISTEMA.

El coeficiente de rozamiento de la articulación del péndulo  $B_p$  se decide considerar nulo, ya que apenas existe rozamiento en ésta. Además el rendimiento del motor  $\eta_m$  se ajusta a un valor mayor, de 0,9, debido a que con estos valores el modelo se ajusta más a los resultados reales obtenidos en las pruebas realizadas.

Por último, es importante mencionar que el alto desgaste del piñón motriz hace que éstos no sean sistemas invariantes en el tiempo, sino que a medida que el desgaste avanza, la respuesta del sistema varía. Se trata de una gran dificultad a la hora de modelar los sistemas, ya que los parámetros seleccionados no son igual de buenos a lo largo de la evolución de un piñón.

## 6.5. DISEÑO DE CONTROLADORES

En este apartado se diseñan controladores válidos para ser incluidos en la aplicación de usuario que se realiza posteriormente. Para ello, se busca su estructura y parámetros óptimos, de manera que se garantice que los controladores incluidos son realizables, y que existen valores que consigan estabilizar el sistema.

El proceso de elección de los parámetros de cada controlador se realiza como un proceso de prueba y error sobre el sistema real, partiendo de valores válidos aproximados obtenidos en el modelo.

En todos los casos, la señal de control  $u$  se limita entre 8V y  $-8V$ , por medio de un saturador, para evitar daños en el motor.

Cabe destacar que todos los controladores se han sintonizado para el sistema sin añadir la masa extra disponible.

### 6.5.1. SISTEMA CARRO

El objetivo de control del sistema carro consiste en controlar únicamente la posición del carro.

#### 6.5.1.1. Controlador PID

Ya que en este sistema únicamente se ha de controlar la variable  $x_c$  la estructura del controlador es de la siguiente forma:

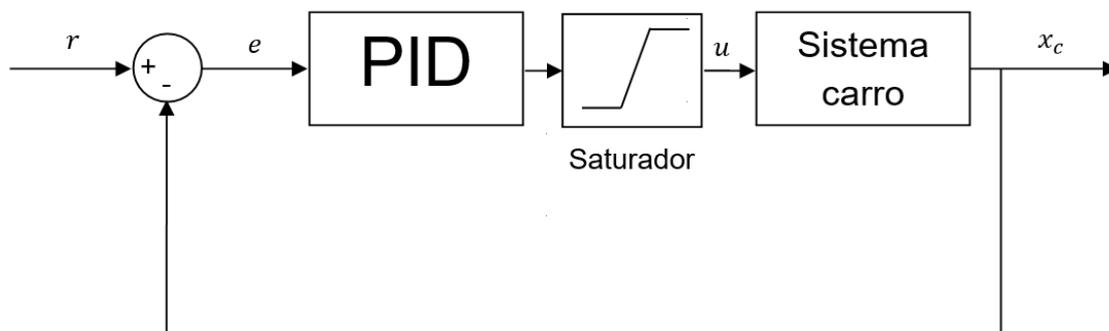


Figura 43. Estructura del controlador PID para el carro

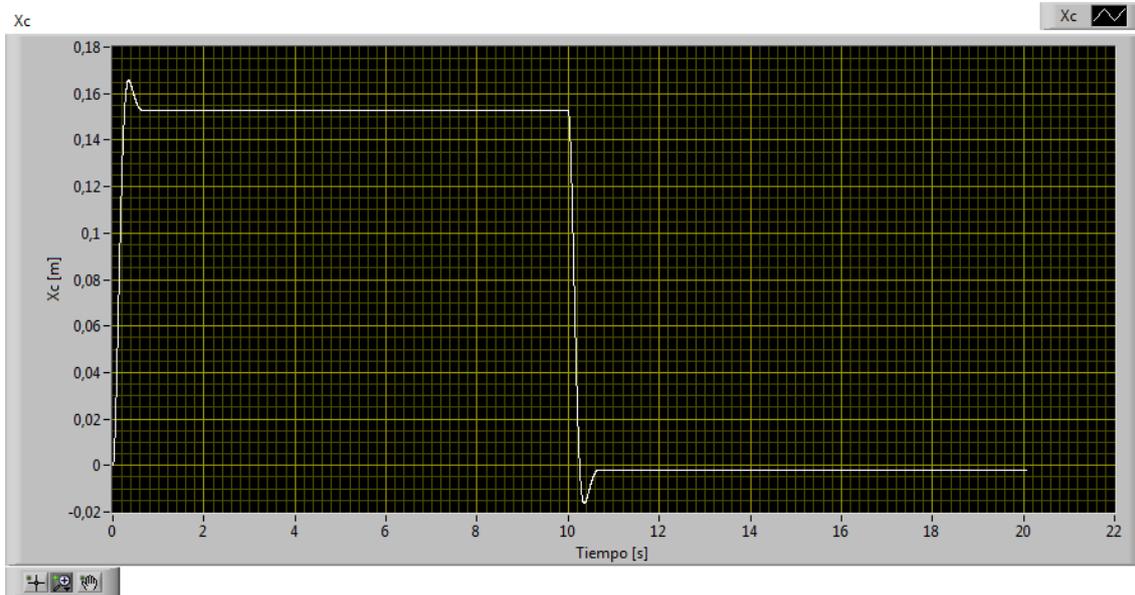
Aunque según el modelo matemático del carro, éste es un integrador puro, por lo que no debería tener error estacionario a una entrada escalón, sin la necesidad, por tanto, de una acción integral, debido a la banda muerta existente que se ha explicado en el apartado anterior es necesario añadir una acción integral para garantizar que no haya error estacionario. Esto ocurre debido a que la banda muerta es una no-linealidad que distorsiona la dinámica del modelo matemático original.

Los parámetros del controlador que se fijan por defecto en la aplicación son los que se muestran a continuación, y se han obtenido mediante prueba y error dada la simplicidad y estabilidad del sistema carro.

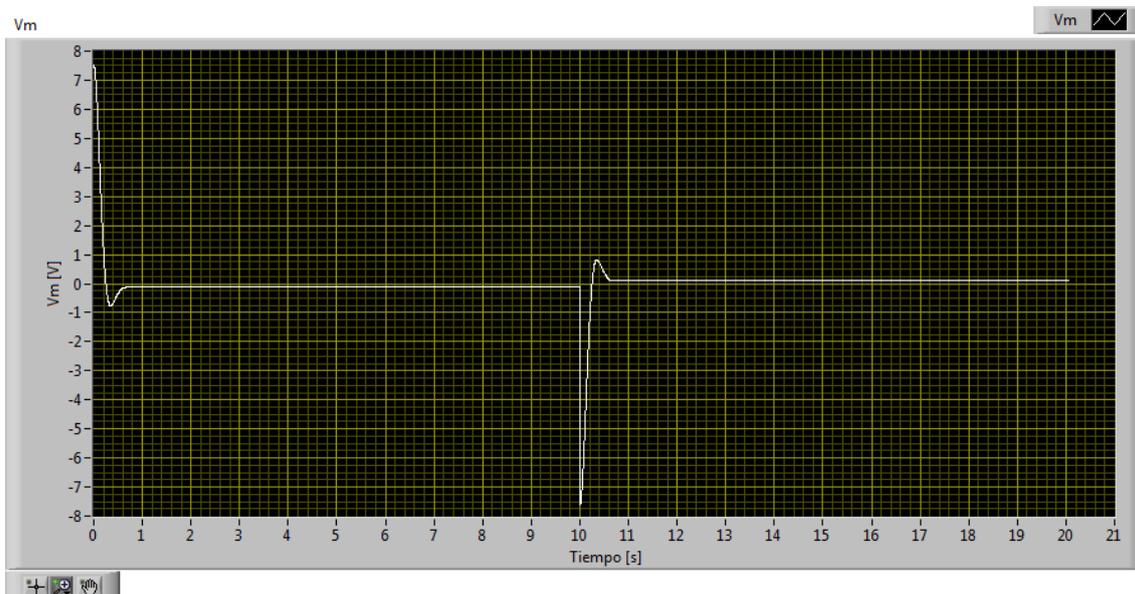
$$PID: \quad K_c = 50 \text{ V/m} \quad T_i = 5,010 \text{ min} \quad T_d = 0 \text{ min} \quad (47)$$

En las figuras siguientes se muestra la respuesta del sistema real  $x_c$  ante una entrada de onda cuadrada de 0,15 metros y la señal de control  $V_m$ . Si las especificaciones del

sistema no admitiesen rebasamiento, se puede encontrar otro conjunto de valores de ganancias para conseguirlo.



*Figura 44. Desplazamiento del carro en la prueba*



*Figura 45. Tensión aplicada al carro en la prueba*

### **6.5.1.2. Controlador de realimentación del estado por asignación de polos**

Como se ha indicado anteriormente, la asignación de polos en lazo cerrado permite definir la dinámica deseada del sistema.

Al igual que en el caso anterior, se requiere integración para eliminar el error estacionario, que en el caso de un controlador por realimentación de estado tiene la forma que se observa en la *Figura 46*.

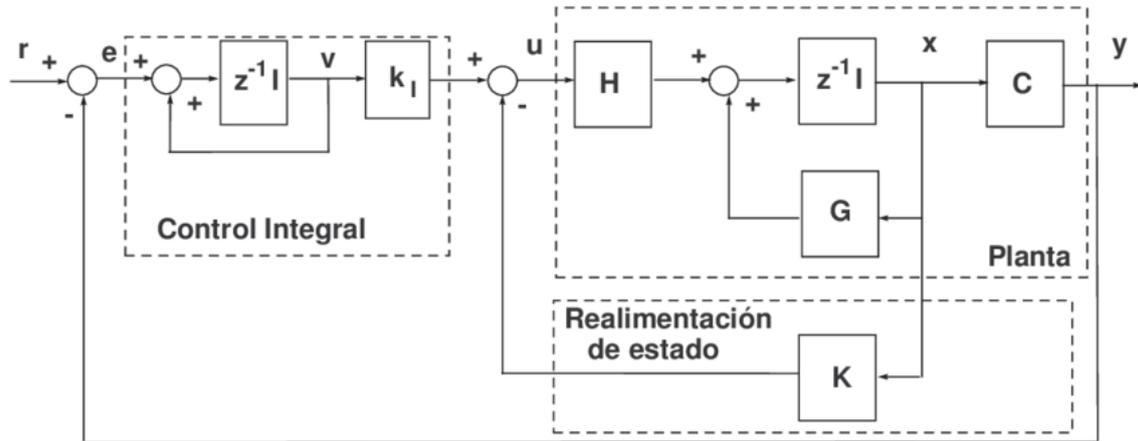


Figura 46. Estructura de un controlador de realimentación del estado con integración

Para calcular las matrices  $K$  y  $K_i$  en un controlador por realimentación del estado es necesario ampliar el sistema considerando la variable  $v$ , la integral del error, una variable de estado más. Es decir, en este caso, el sistema pasa de dos a tres variables de estado. Por lo tanto, si en el sistema original había que seleccionar dos polos para fijar la dinámica del sistema en lazo cerrado, ahora son necesarios tres.

Se puede deducir que el sistema ampliado, o aumentado, viene dado por la siguiente ecuación:

$$\begin{bmatrix} \dot{x}(t) \\ \dot{v}(t) \end{bmatrix} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ v(t) \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u(t) + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (48)$$

siendo  $A$ ,  $B$  y  $C$  las matrices del sistema de estados, y la referencia  $r$  constante.

Si se analiza la dinámica del error de las variables de estado se llega a la ecuación (50), definiéndose el mismo como:

$$e(t) = \begin{bmatrix} x_e(t) \\ v_e(t) \end{bmatrix} = \begin{bmatrix} x(t) - x(\infty) \\ v(t) - v(\infty) \end{bmatrix} \quad (49)$$

$$\dot{e}(t) = \hat{A} e(t) + \hat{B} u(t) \quad (50)$$

donde  $\hat{A} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix}$  y  $\hat{B} = \begin{bmatrix} B \\ 0 \end{bmatrix}$  son las matrices ampliadas del sistema.

Se puede resolver la matriz de ganancias del sistema ampliado  $\hat{K}$  mediante varios métodos, por ejemplo mediante la fórmula de Ackermann, que se puede implementar fácilmente en Matlab o LabVIEW.

La fórmula de Ackermann viene dada por la ecuación (51), donde  $\phi(A)$  es la ecuación característica deseada evaluada en  $A$ , que se muestra en la expresión (52), donde  $\alpha_i$  son sus coeficientes.

$$K = [0 \ 0 \ \dots \ 0 \ 1] [B \ AB \ \dots \ A^{n-1}B]^{-1} \phi(A) \quad (51)$$

$$\phi(A) = A^n + \alpha_1 A^{n-1} + \dots + \alpha_{n-1} A + \alpha_n I \quad (52)$$

Para el comando utilizado en LabVIEW, sin embargo, únicamente es necesario especificar los polos deseados y el sistema calcula automáticamente la ecuación característica deseada del sistema y con ella la matriz  $K$  de ganancias.

Una vez resuelta para el sistema aumentado se obtiene  $\hat{K}$ , y se sabe por la manera en que se define este sistema que:

$$\hat{K} = [K \mid -K_i] \quad (53)$$

siendo  $K$  una matriz con el mismo número de columnas que la dimensión del sistema original, ya que se utiliza para la realimentación del estado, y  $K_i$  la matriz de constantes de integración (normalmente sólo existirá una), con tantas constantes de integración como integraciones sobre distintas variables se realicen.

En este caso,  $K$  es un vector de dimensión 2 y  $K_i$  una única constante de integración, para la integración de la variable  $x_c$ .

En el algoritmo de control, para este caso y los casos de realimentación del estado siguientes, se toma el error constante a lo largo del tiempo de muestreo en cada ciclo, sumándose al error de integración acumulado en las iteraciones anteriores al principio de cada iteración.

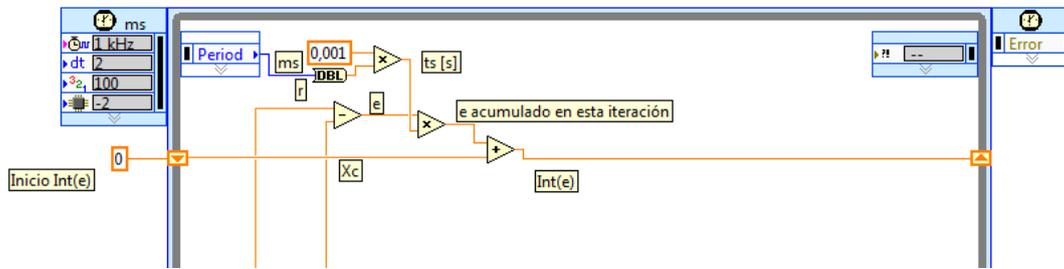


Figura 47. Integración del error implementada en LabVIEW

En la Figura 47, el valor que llega a la flecha de la derecha del bucle temporizado se transfiere a la siguiente iteración, partiendo desde la flecha de la izquierda del mismo bucle. Este elemento es conocido en LabVIEW como un *Shift Register*.

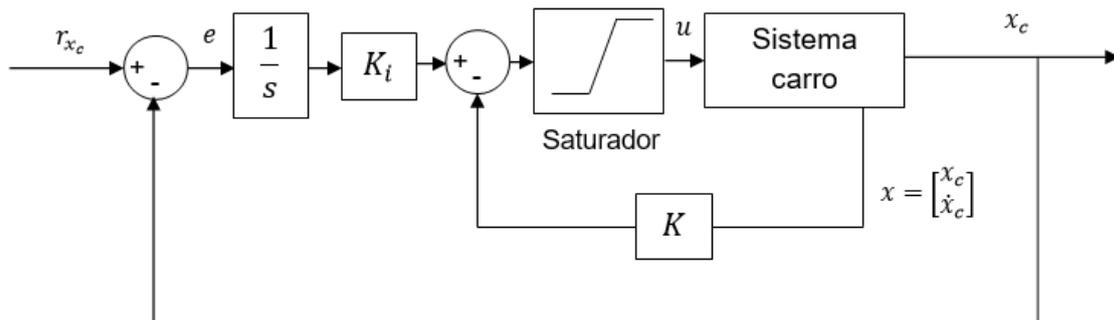


Figura 48. Estructura del controlador de realimentación del estado para el carro

Para sintonizar el controlador, en este caso, se seleccionan dos polos dominantes complejos conjugados que definan la dinámica del sistema en lazo cerrado, mientras que el tercer polo se elige arbitrariamente, alejado de los anteriores, y situado en el eje real negativo, para que no tenga apenas influencia en la dinámica.

Las especificaciones de la respuesta se definen, por tanto, asimilando el sistema en lazo cerrado a un sistema de segundo orden:

- $SO$ : rebasamiento máximo ante entrada escalón.
- $T_s$  = Tiempo de establecimiento del sistema dentro del  $\pm 2\%$  del valor final.

En un sistema de segundo orden con amortiguamiento  $\xi$  y pulsación  $\omega_n$ , los polos  $p_1$  y  $p_2$  vienen dados por:

$$p_1 = -\xi \omega_n + \beta \omega_n j \quad (54)$$

$$p_2 = -\xi \omega_n - \beta \omega_n j \quad (55)$$

Donde

$$\beta = \sqrt{1 - \xi^2} \quad (56)$$

Además,  $\xi$  y  $\omega_n$  están relacionados con  $SO$  y  $T_s$  mediante las fórmulas:

$$\xi = \frac{\ln^2\left(\frac{SO}{100}\right)}{\sqrt{\pi^2 + \ln^2\left(\frac{SO}{100}\right)}} \quad (57)$$

$$\omega_n = \frac{4}{\xi T_s} \quad (58)$$

Por lo tanto, los polos conjugados dominantes son:

$$p_{1,2} = -\xi \omega_n \pm \beta \omega_n j \quad (59)$$

El tercer polo se elige como se ha explicado previamente.

Para un  $SO = 0,01\%$  y  $T_s = 1s$  se obtienen los polos conjugados  $p_{1,2} = -4 \pm 1,364 j$ , y asignando  $p_3 = -15$ , la fórmula de Ackermann da el siguiente valor para  $\hat{K}$ .

$$\hat{K} = [41,04 \quad -0,13 \quad -79,76] \quad (60)$$

donde  $K = [41,04 \quad -0,13]$  y  $K_i = 79,76$ .

En las siguientes figuras se muestra la respuesta del sistema real  $x_c$ , su derivada  $\dot{x}_c$  y la acción de control  $V_m$ , ante una referencia de onda cuadrada.

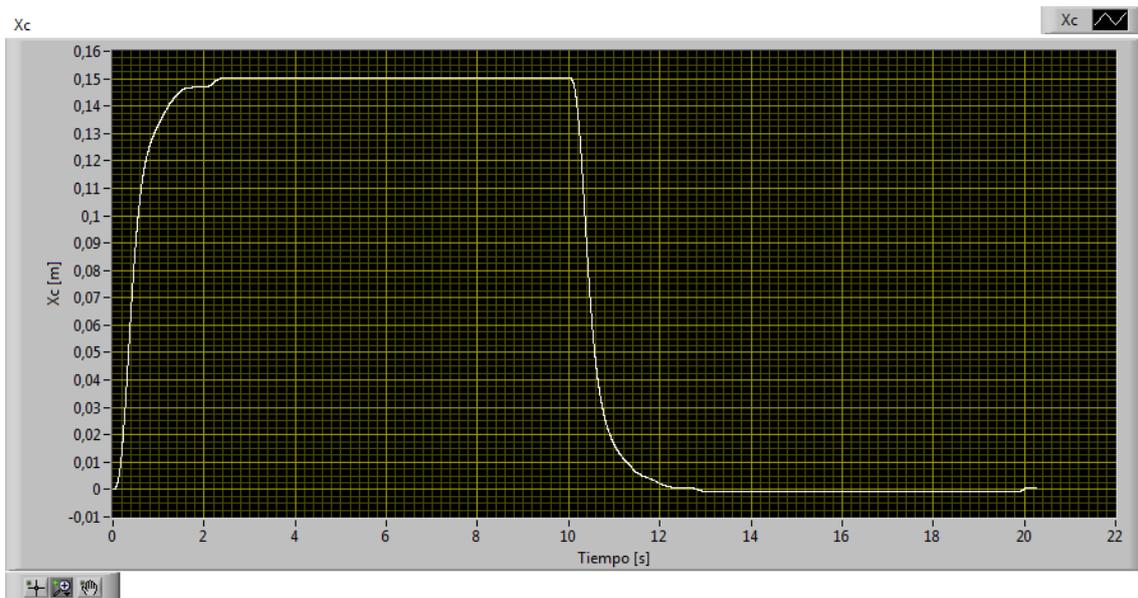


Figura 49. Desplazamiento del carro en la prueba

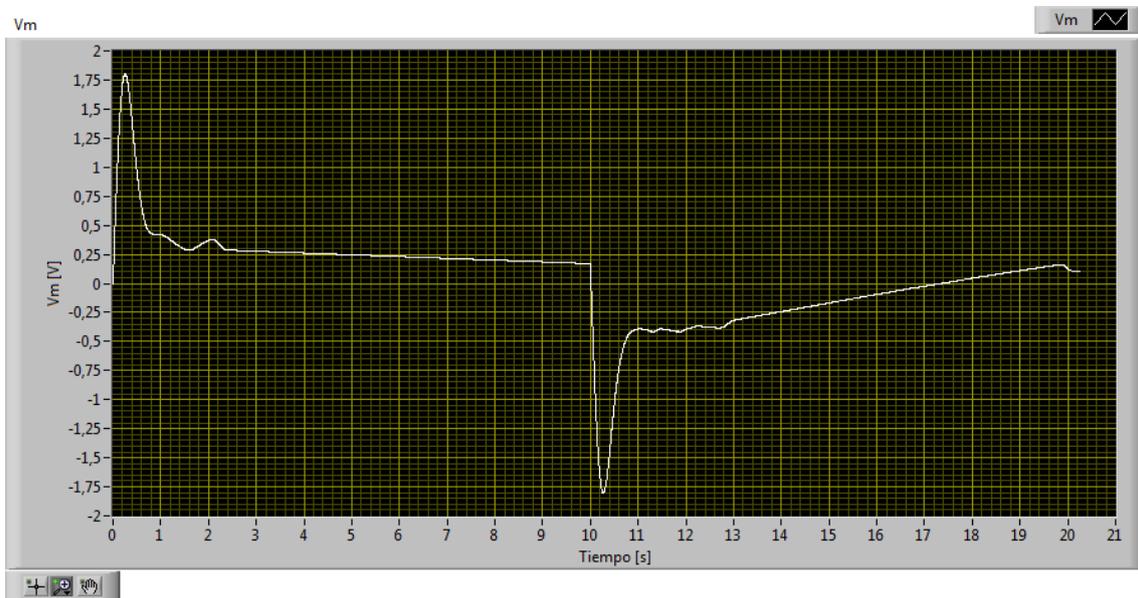


Figura 50. Tensión aplicada al carro en la prueba

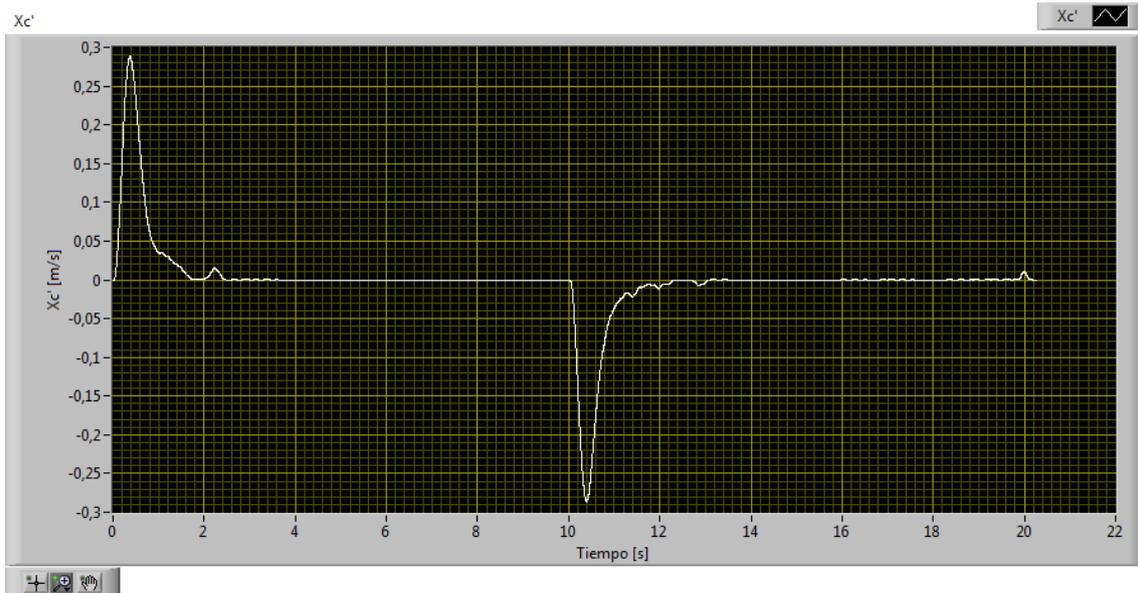


Figura 51. Velocidad del carro en la prueba

### 6.5.1.3. Controlador de realimentación del estado LQR

Como ya se ha explicado en el apartado 5.2.2. *CONTROLADOR POR REALIMENTACIÓN DEL ESTADO*, en este caso el esquema del controlador es el mismo que en el caso anterior. Sólo cambia la manera en que se calcula la matriz de realimentación del estado  $\hat{K}$  del sistema ampliado, y por lo tanto, también las matrices  $K$  y  $K_i$ .

Seleccionando las matrices  $Q$  y  $R$ , es sencillo ejecutar el algoritmo de optimización LQR mediante Matlab o LabVIEW. La matriz  $Q$  tiene la misma dimensión que el sistema ampliado, es decir, en este caso tres. La matriz  $R$  tiene la misma dimensión que el número de entradas al sistema, en este caso una.

Para este sistema, se pondera especialmente la minimización del error de  $x_c$ , dando valores grandes a los términos de  $x_c$  y de la integral del error de  $x_c$ ,  $Q_{11}$  y  $Q_{33}$  respectivamente. Por el contrario, se le asigna poco peso a  $\dot{x}_c$ .

$$Q = \begin{bmatrix} 200 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 3000 \end{bmatrix} \quad (61)$$

El peso asignado a  $V_m$ , mediante la matriz  $R$ , permite limitar la acción de control, evitando que tome valores excesivos. En este caso, se escoge un peso pequeño porque no interesa limitarla, ya que no existen especificaciones de energía en el control del sistema.

$$R = 0,3 \quad (62)$$

Con los valores  $Q$  y  $R$  anteriores, se utiliza la función LQR en LabVIEW, que encuentra la acción de control óptima minimizando la función de coste del algoritmo, y por lo tanto, devuelve la matriz de realimentación del estado del sistema ampliado que se muestra a continuación.

$$\hat{K} = [51,38 \quad 2,30 \quad -100] \quad (63)$$

Por consiguiente, los valores de las ganancias del sistema son las siguientes.

$$K = [51,38 \quad 2,30] \quad (64)$$

$$K_i = 100 \quad (65)$$

En las figuras mostradas a continuación, se observan la respuesta del sistema real  $x_c$ , su derivada  $\dot{x}_c$  y la acción de control  $V_m$ , ante una referencia de onda cuadrada de 0,15 metros.

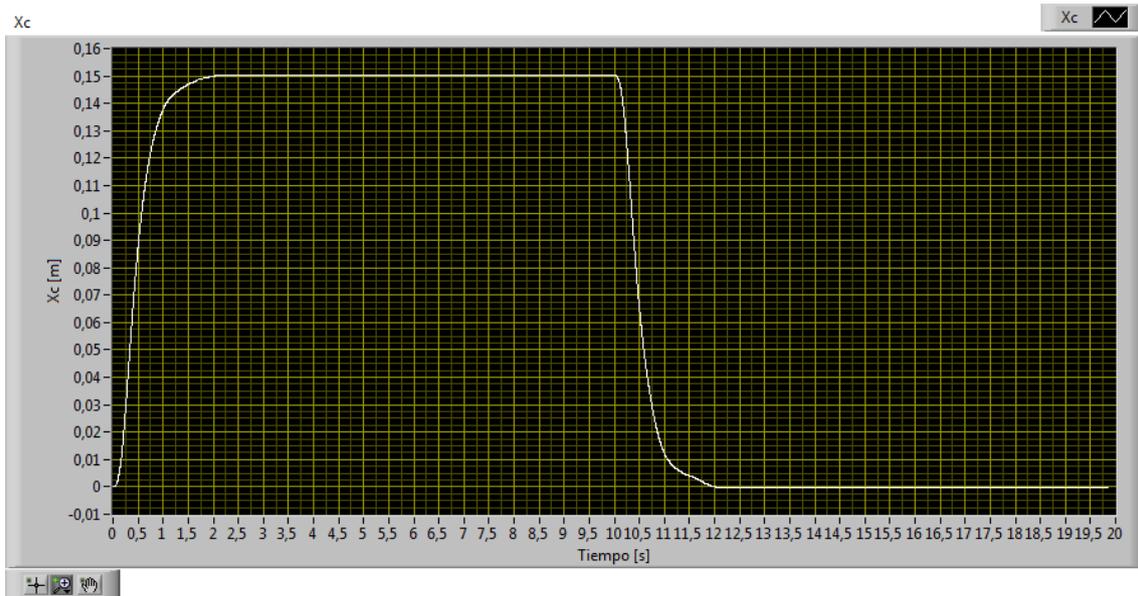


Figura 52. Desplazamiento del carro en la prueba

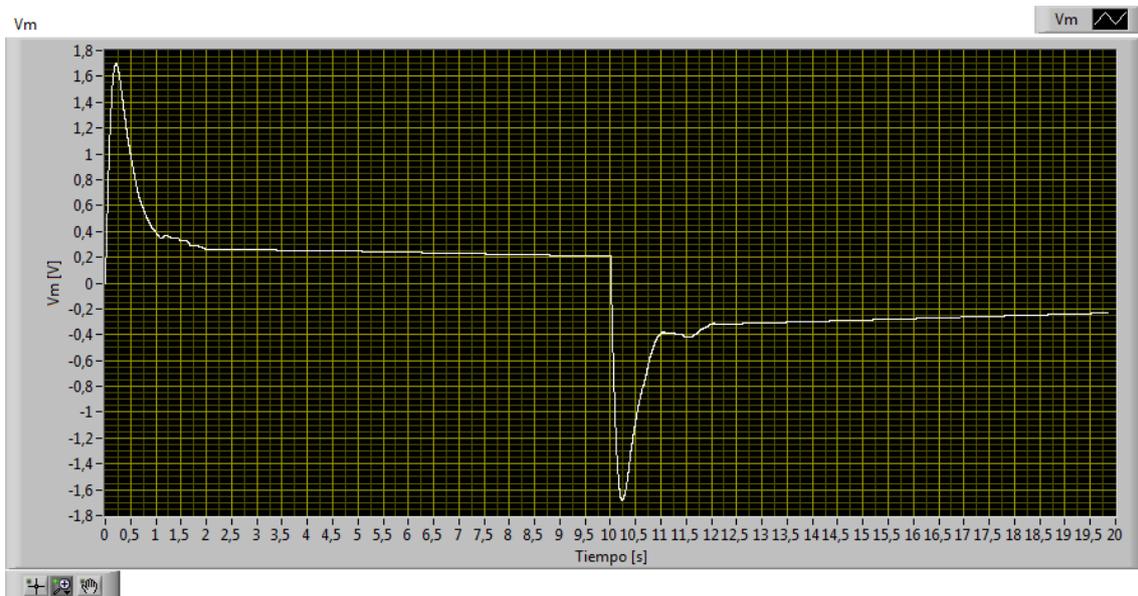


Figura 53. Tensión aplicada al carro en la prueba

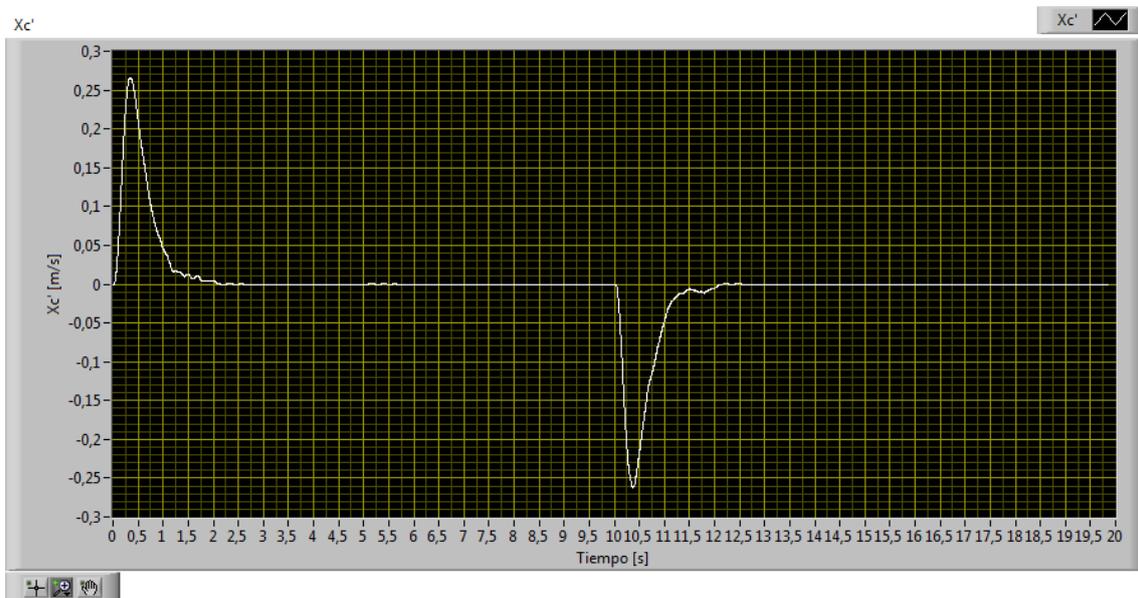


Figura 54. Velocidad del carro en la prueba

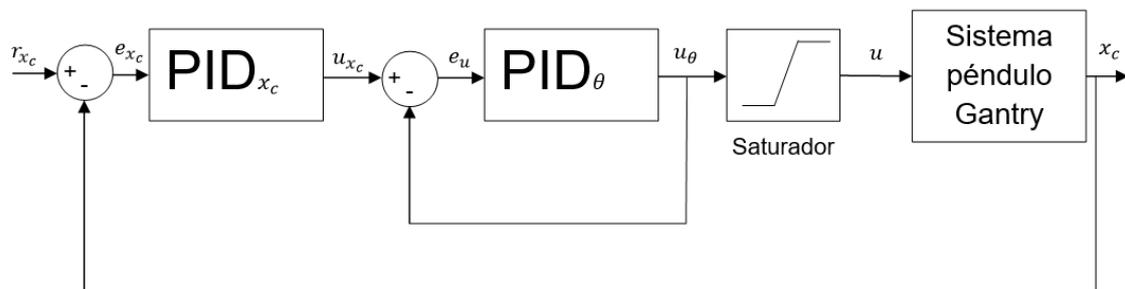
## 6.5.2. SISTEMA PENDULO GANTRY

En este caso se simula el movimiento de una carga suspendida de un puente-grúa, en el que el objetivo es que la posición de la carga se desplace hasta una coordenada horizontal determinada y con la menor oscilación posible del péndulo.

### 6.5.2.1. Controlador PID

En este sistema, aunque la salida considerada del sistema es  $x_t = x_c + 2 l_p \theta$ , es importante la forma en la que se consigue seguir a la referencia, ya que podrían darse valores altos de  $\theta$ , lo que es inadmisibles en el funcionamiento de un sistema de este tipo. Por lo tanto, se desecha la idea de utilizar un único algoritmo PID que trabaje con el error de la señal  $x_t$ , calculada a partir de las medidas de  $x_c$  y  $\theta$ .

En consecuencia, se utilizan dos algoritmos PID, uno para la variable  $x_c$ , para la que se establece la referencia deseada, y otro para la variable  $\theta$ , cuya referencia es cero. En cuanto a la configuración del esquema de control con dos algoritmos PID, se prueba a conectarlos en cascada, con el PID de  $\theta$  como lazo interno, ya que es más rápido que el del desplazamiento del carro  $x_c$ , como en el esquema de la *Figura 55*.



*Figura 55. Esquema de PIDs conectados en cascada para el péndulo Gantry*

Sin embargo, no se consigue un resultado del todo satisfactorio en el control, por lo que se decide utilizar el esquema propuesto en la *Figura 56*, en el que se presenta una arquitectura de los dos controladores PID conectados en paralelo, en el que se suman las salidas de ambos para obtener la señal de control.

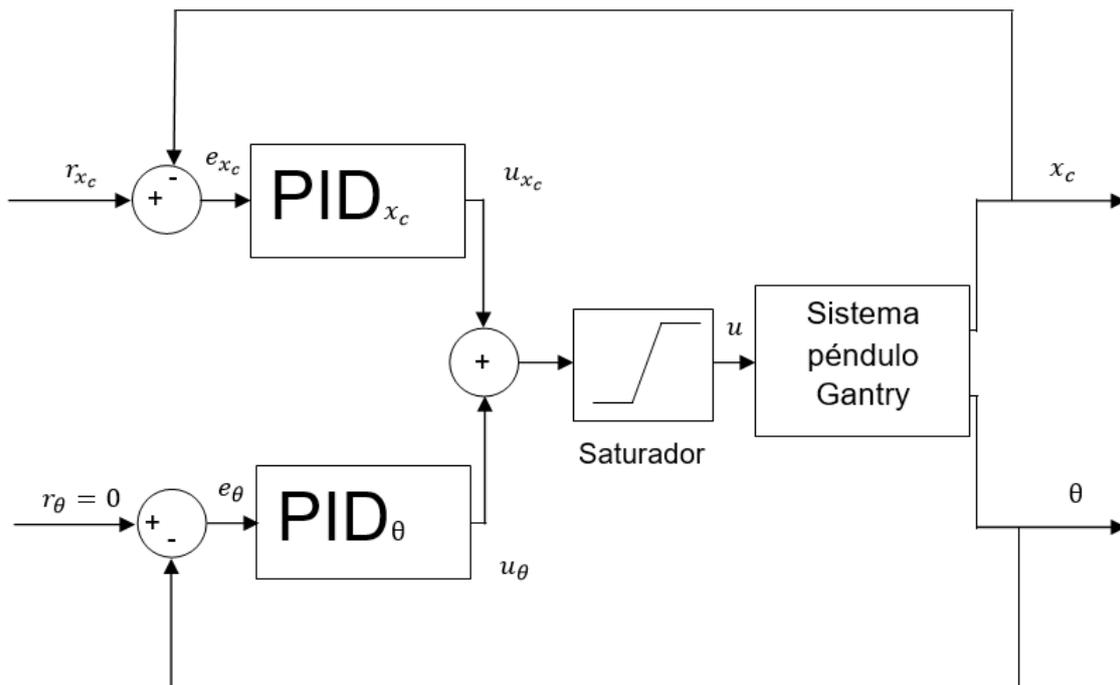


Figura 56. Estructura del controlador PID para el péndulo Gantry

Para el sistema péndulo Gantry, también es necesario utilizar integración en  $x_c$  para asegurar un error nulo en estado estacionario ante entrada escalón. La integración en  $\theta$  no es tan importante, ya que el sistema tiende naturalmente a  $\theta = 0$ .

Para la selección de las ganancias de ambos controladores se tiene en cuenta lo siguiente:

- El controlador de  $\theta$  debe ser más rápido que el de  $x_c$  para evitar oscilaciones grandes del péndulo, por tanto, la constante  $K_c$  del PID de  $\theta$  se elige mayor que la del PID de  $x_c$ .
- De acuerdo a las consideraciones de integración, el parámetro  $T_i$  del PID de  $x_c$  debe ser menor (acción integral mayor) que el mismo parámetro del PID de  $\theta$ .
- Los valores de  $T_d$  en ambos casos se eligen mediante el método de prueba y error, aunque en todo caso no se requieren valores elevados de este parámetro.
- Los signos de  $K_c$  se eligen para que la realimentación sea negativa.

### **Péndulo Corto**

Teniendo en cuenta las consideraciones anteriores, se seleccionan para el péndulo corto los siguientes valores:

$$PID_{x_c}: \quad K_c = 20 \text{ V/m} \quad T_i = 0,4 \text{ min} \quad T_d = 2,5 * 10^{-5} \text{ min} \quad (66)$$

$$PID_{\theta}: \quad K_c = -30 \text{ V/rad} \quad T_i = 5 \text{ min} \quad T_d = 3,33 * 10^{-6} \text{ min} \quad (67)$$

Los resultados en el sistema real para una entrada con forma de onda cuadrada de 0,15 metros de amplitud son los que se muestran en las siguientes figuras. Se registran la salida del sistema  $x_t$ , las variables  $x_c$  y  $\theta$ , y la acción de control  $V_m$ .

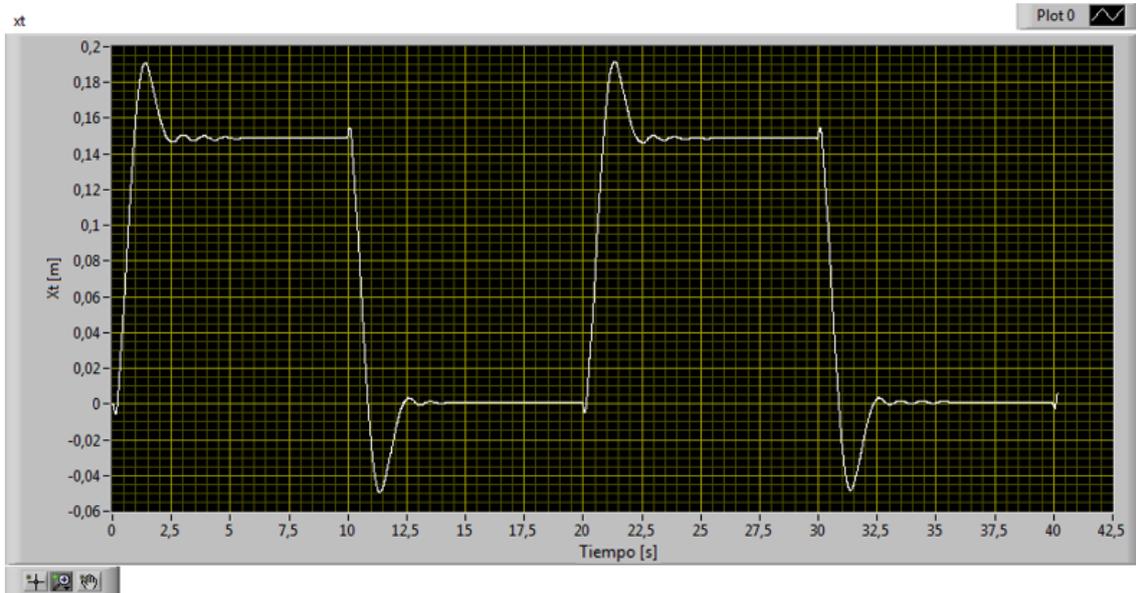


Figura 57. Salida del sistema en la prueba

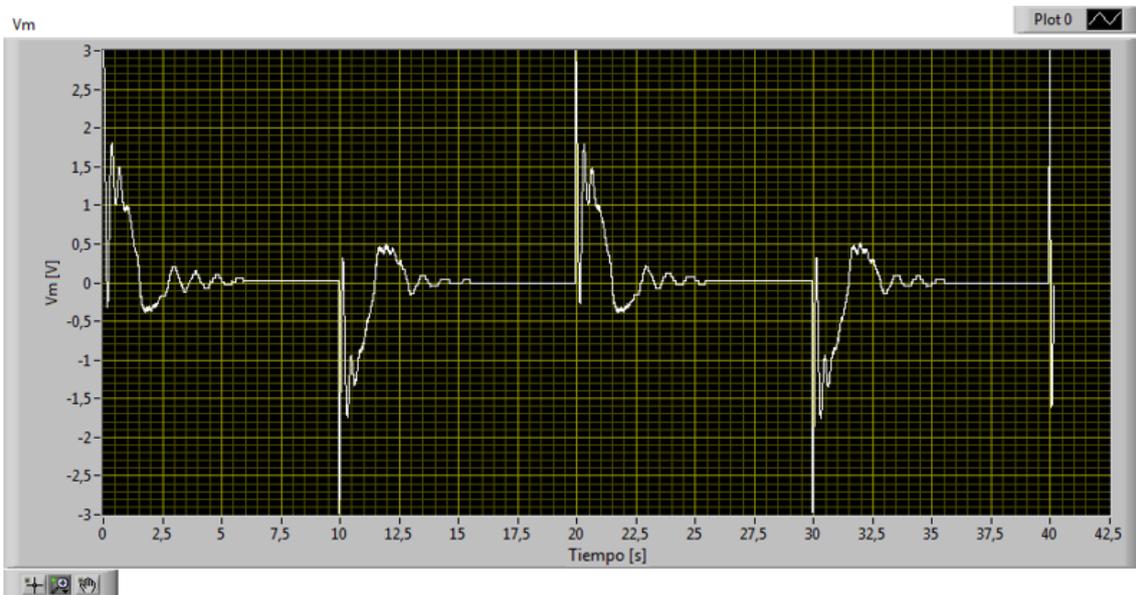


Figura 58. Tensión aplicada al péndulo Gantry en la prueba

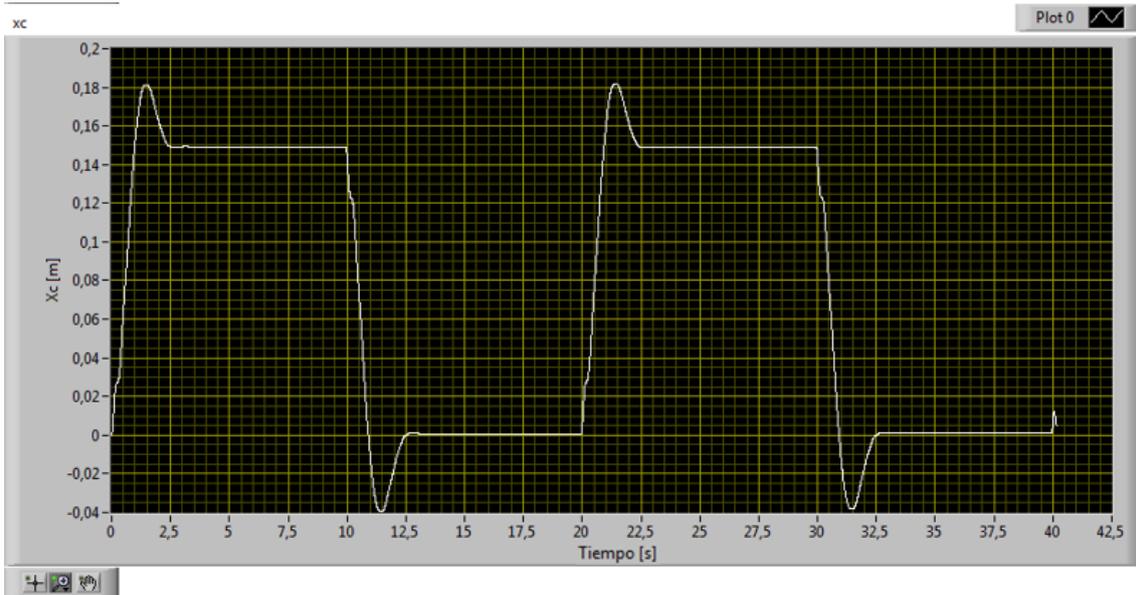


Figura 59. Desplazamiento del carro en la prueba

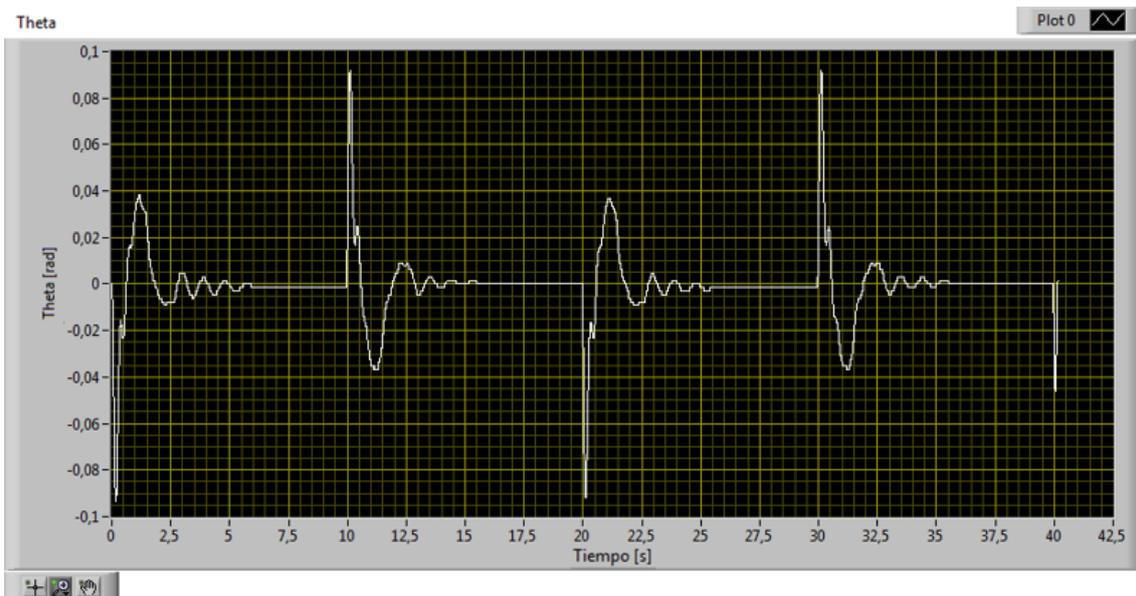


Figura 60. Ángulo del péndulo Gantry en la prueba

### **Péndulo Largo**

En el caso del péndulo largo, los mismos parámetros para los controladores PID resultan ser válidos.

$$PID_{x_c}: \quad K_c = 20 \text{ V/m} \quad T_i = 0,4 \text{ min} \quad T_d = 2,5 * 10^{-5} \text{ min} \quad (68)$$

$$PID_{\theta}: \quad K_c = -30 \text{ V/rad} \quad T_i = 5 \text{ min} \quad T_d = 3,33 * 10^{-6} \text{ min} \quad (69)$$

De nuevo, para una onda cuadrada de 0,15 metros de amplitud, los resultados en el sistema real para la salida del sistema  $x_t$ , las variables  $x_c$  y  $\theta$ , y la acción de control  $V_m$  son los que se muestran a continuación.

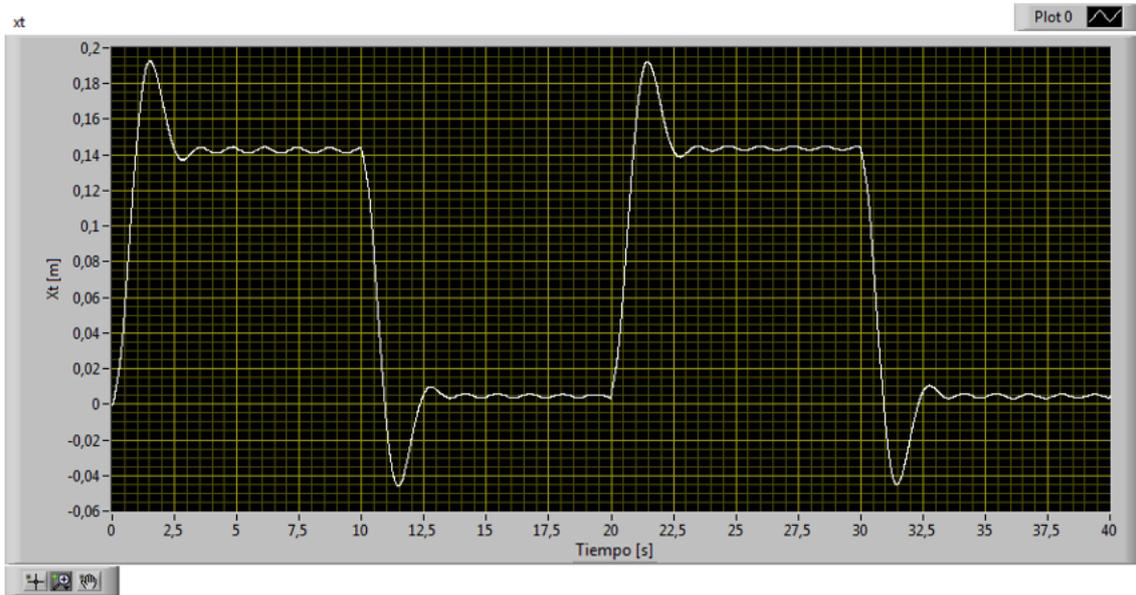


Figura 61. Salida del sistema en la prueba

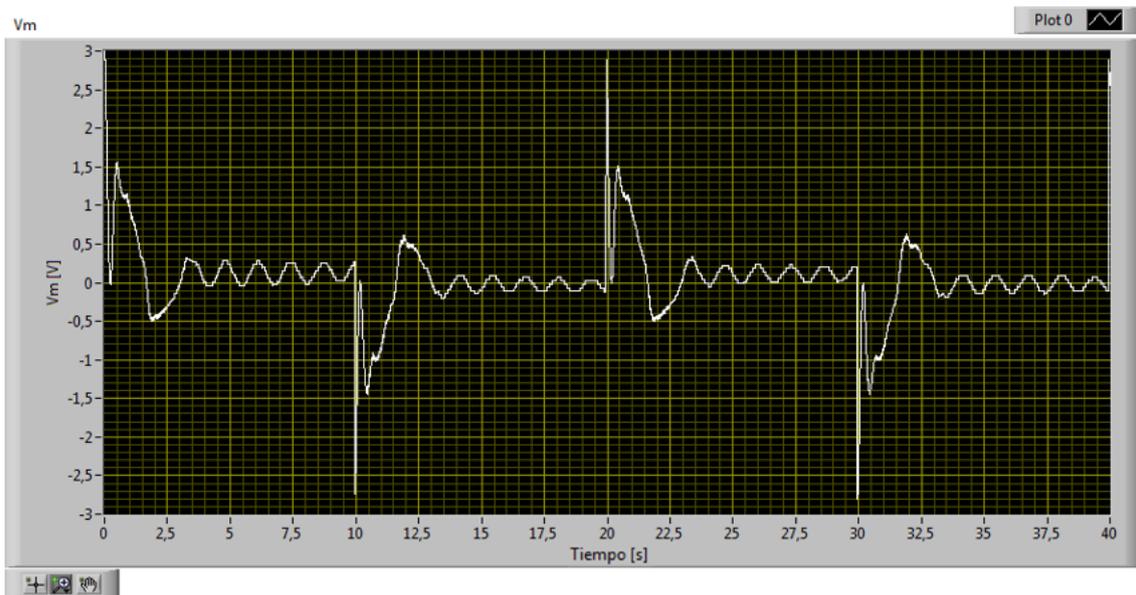


Figura 62. Tensión aplicada al péndulo Gantry en la prueba

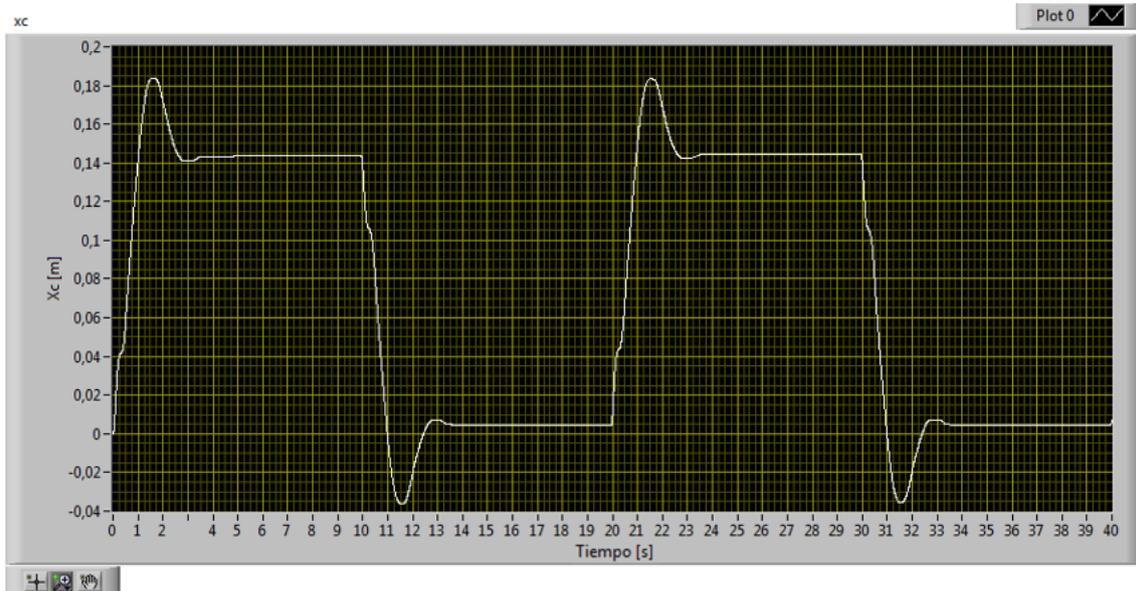


Figura 63. Desplazamiento del carro en la prueba

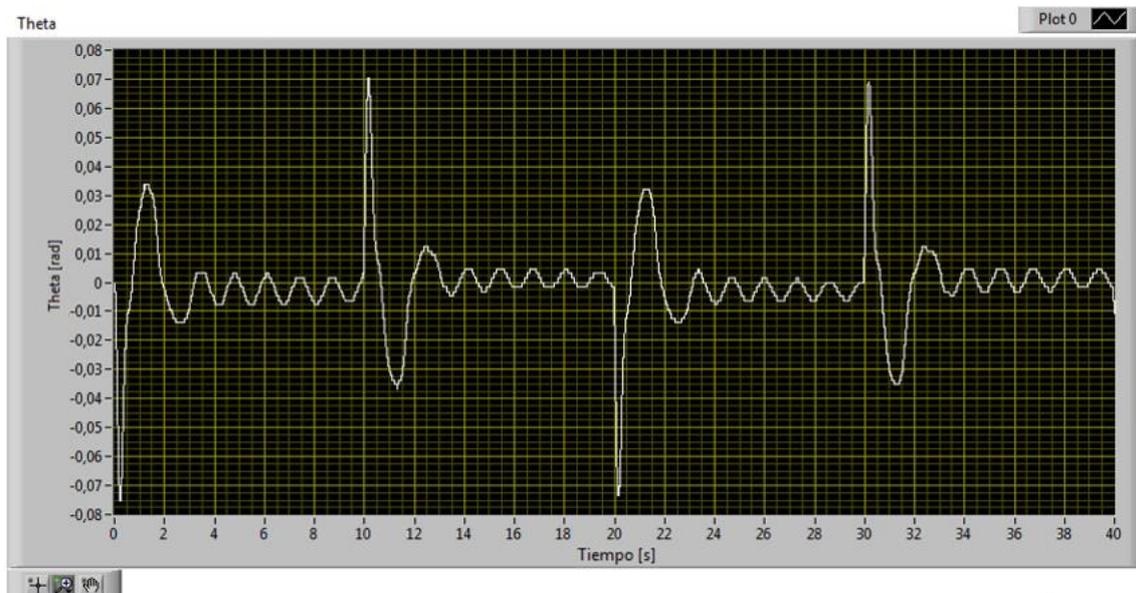


Figura 64. Ángulo del péndulo Gantry en la prueba

### **6.5.2.2. Controlador de realimentación del estado por asignación de polos**

En este caso también es necesario utilizar integración, por lo que el sistema de estados se ve aumentado en una dimensión, de cuatro a cinco.

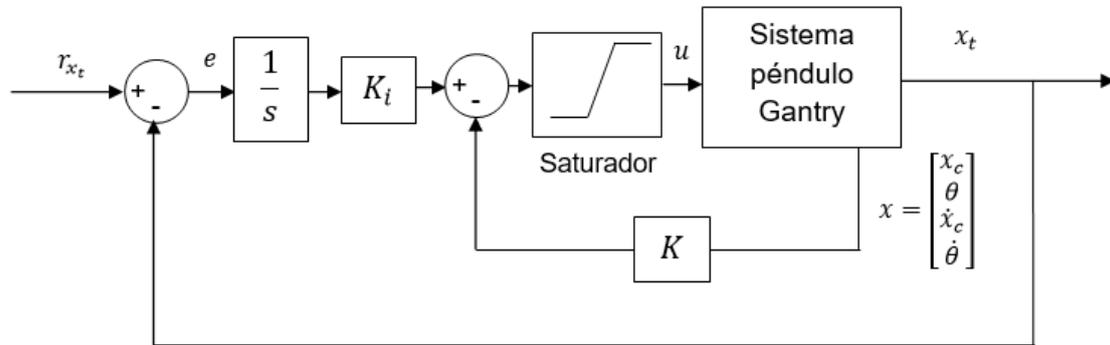


Figura 65. Estructura del control por realimentación del estado para el péndulo Gantry

Para sintonizar el controlador, en este caso, se seleccionan dos polos complejos conjugados para definir la dinámica del sistema en lazo cerrado, mientras que los otros tres se eligen arbitrariamente, más alejados, y en el eje real negativo.

### Péndulo Corto

Haciendo las mismas consideraciones que para el sistema carro, para  $SO = 0,01\%$  y  $T_s = 1s$  se obtienen los polos conjugados  $p_{1,2} = -4 \pm 1,364j$ , y asignando

$$p_3 = -7 \quad (70)$$

$$p_4 = -11 \quad (71)$$

$$p_5 = -15 \quad (72)$$

la fórmula de Ackermann da el siguiente valor de

$$\hat{K} = [127,07 \quad -19,98 \quad 31,44 \quad 4,75 \quad -169,77] \quad (73)$$

donde  $K = [127,07 \quad -19,98 \quad 31,44 \quad 4,75]$  y  $K_i = 169,77$ .

Con estos valores, para una onda cuadrada de 0,15 metros de amplitud, se obtienen sobre el sistema real los resultados que se muestran a continuación. En este caso, se registran la salida del sistema  $x_t$ , la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\theta$  y sus respectivas derivadas.

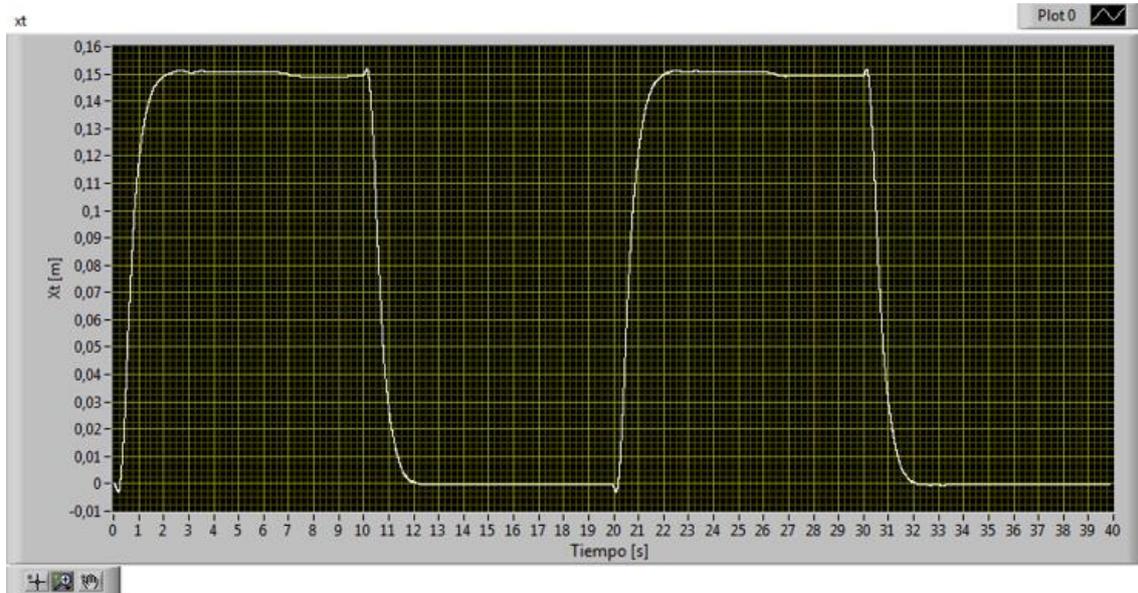


Figura 66. Salida del sistema en la prueba

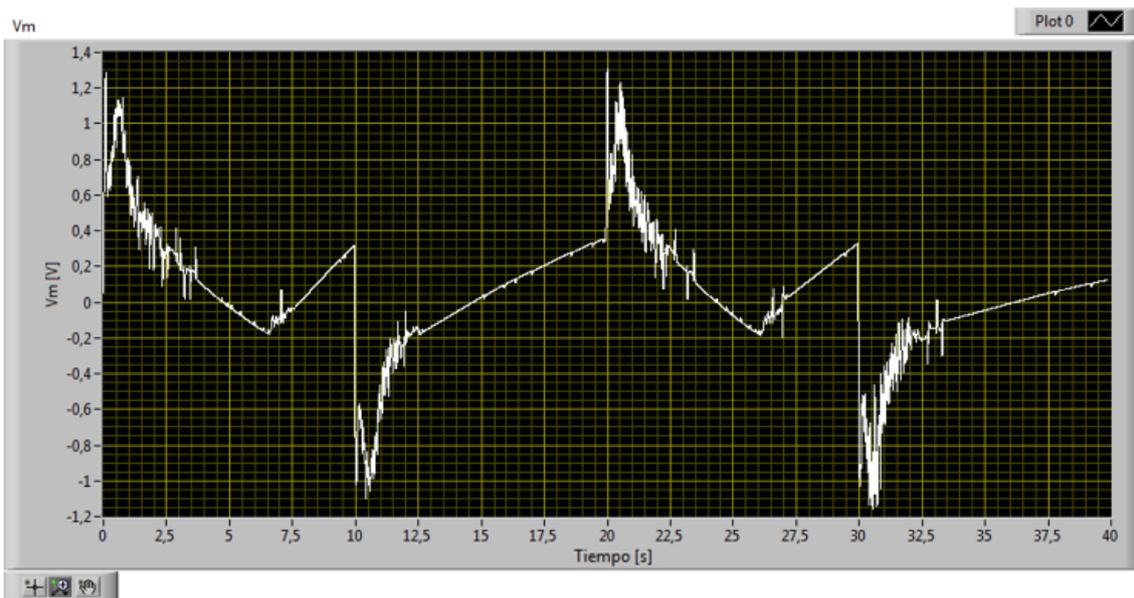


Figura 67. Tensión aplicada al péndulo Gantry en la prueba

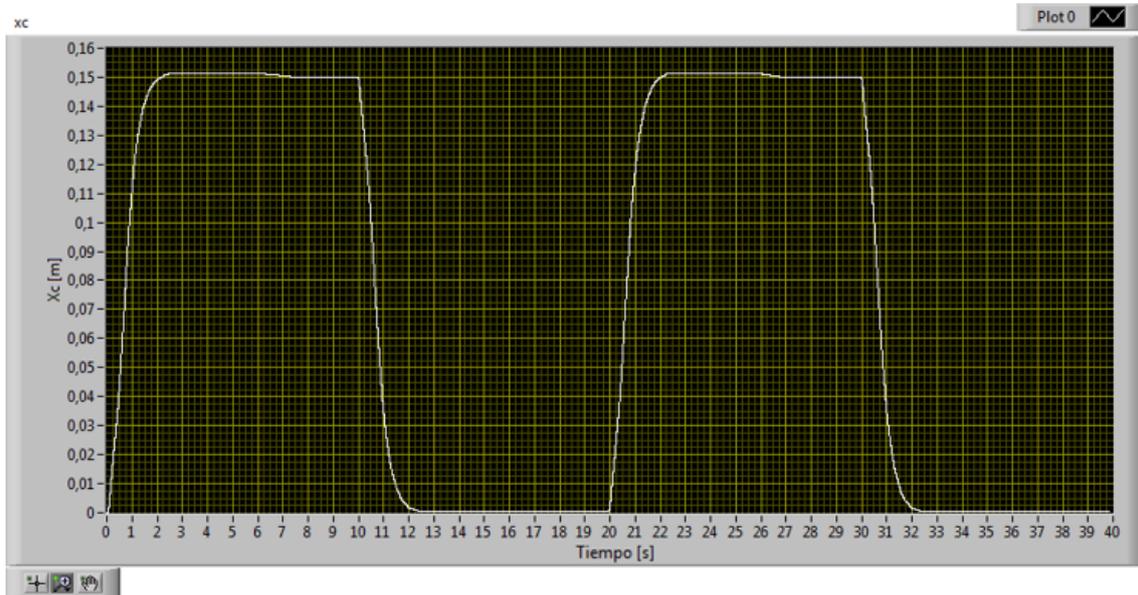


Figura 68. Desplazamiento del carro en la prueba



Figura 69. Ángulo del péndulo Gantry en la prueba

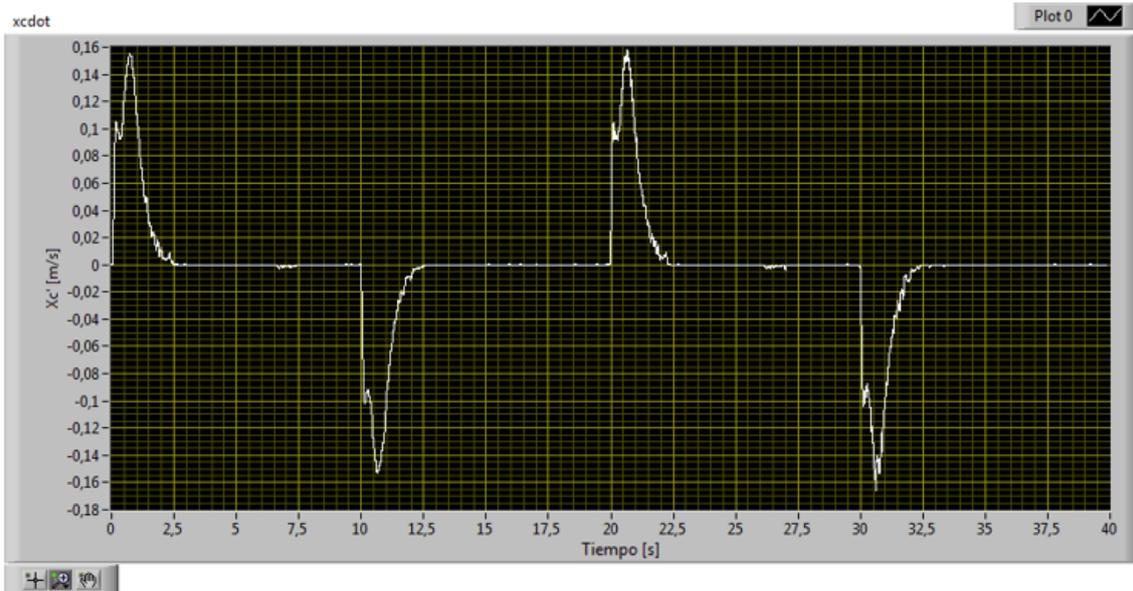


Figura 70. Velocidad del carro en la prueba

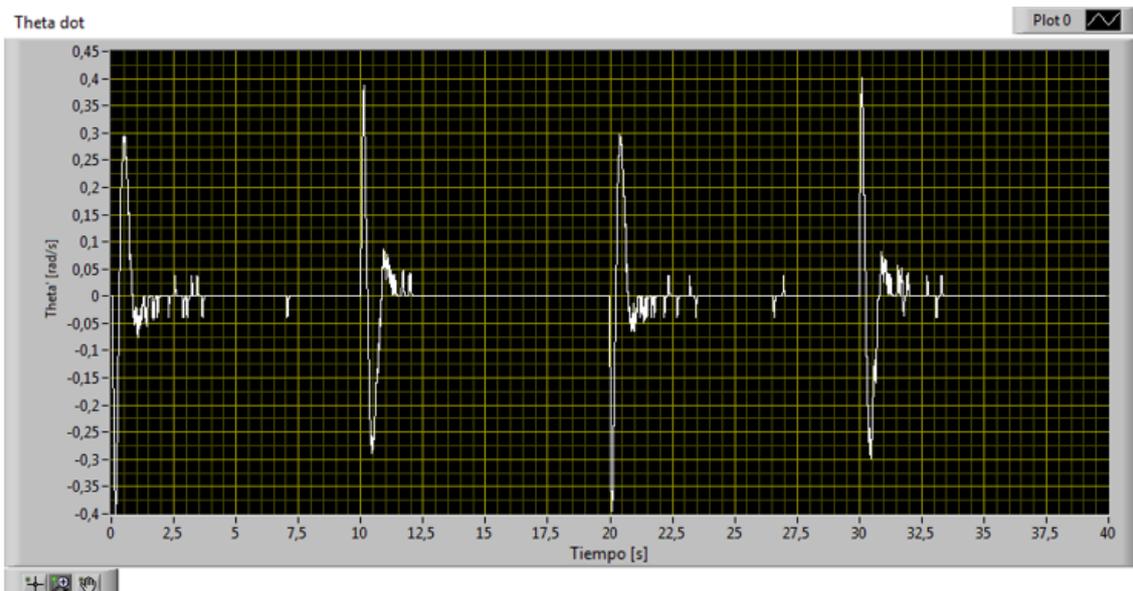


Figura 71. Velocidad angular del péndulo Gantry en la prueba

### Péndulo Largo

En el caso del péndulo largo, se fijan los mismo requisitos para la dinámica del sistema en lazo cerrado  $SO = 0,01\%$  y  $T_s = 1s$  y, por lo tanto, se obtienen los mismos polos conjugados  $p_{1,2} = -4 \pm 1,364j$ . Además, los otros tres polos que se eligen también coinciden con los seleccionados en el caso del péndulo corto.

$$p_3 = -7 \quad (74)$$

$$p_4 = -11 \quad (75)$$

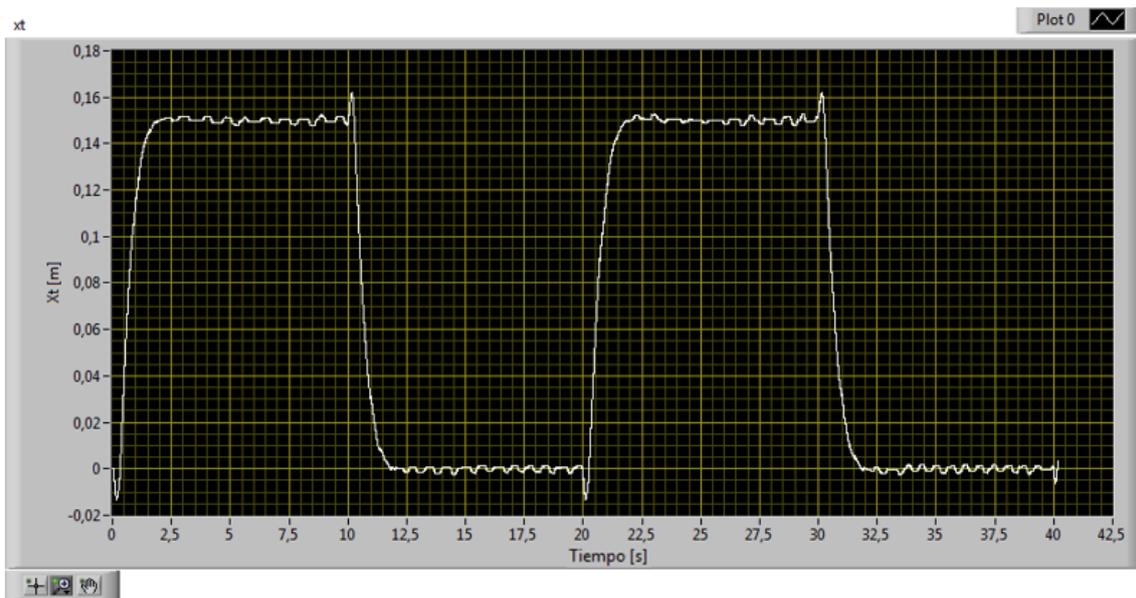
$$p_5 = -15 \quad (76)$$

Sin embargo, ésta vez la fórmula de Ackermann da un valor distinto de la matriz de ganancias del sistema aumentado:

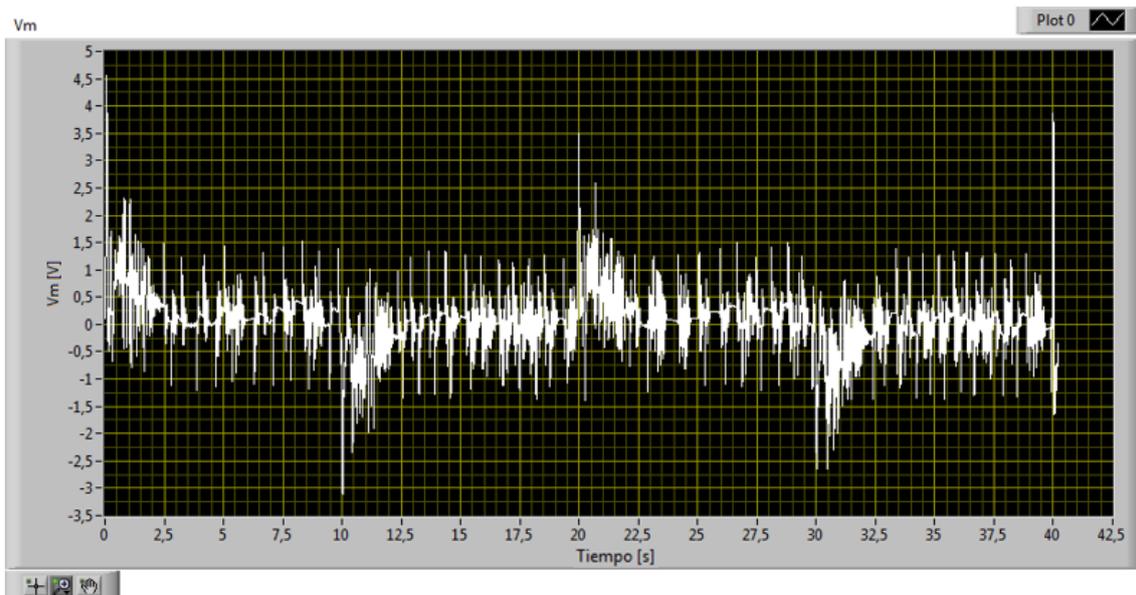
$$\hat{K} = [283,20 \quad 16,73 \quad 83,08 \quad 32,85 \quad -378,44] \quad (77)$$

donde  $K = [283,20 \quad 16,73 \quad 83,08 \quad 32,85]$  y  $K_i = 378,44$ .

Con una onda cuadrada de 0,15 metros de amplitud como referencia, se muestran en las figuras siguientes la salida del sistema  $x_t$ , la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\theta$  y sus respectivas derivadas, obtenidas en el sistema real.



*Figura 72. Salida del sistema en la prueba*



*Figura 73. Tensión aplicada al péndulo Gantry en la prueba*

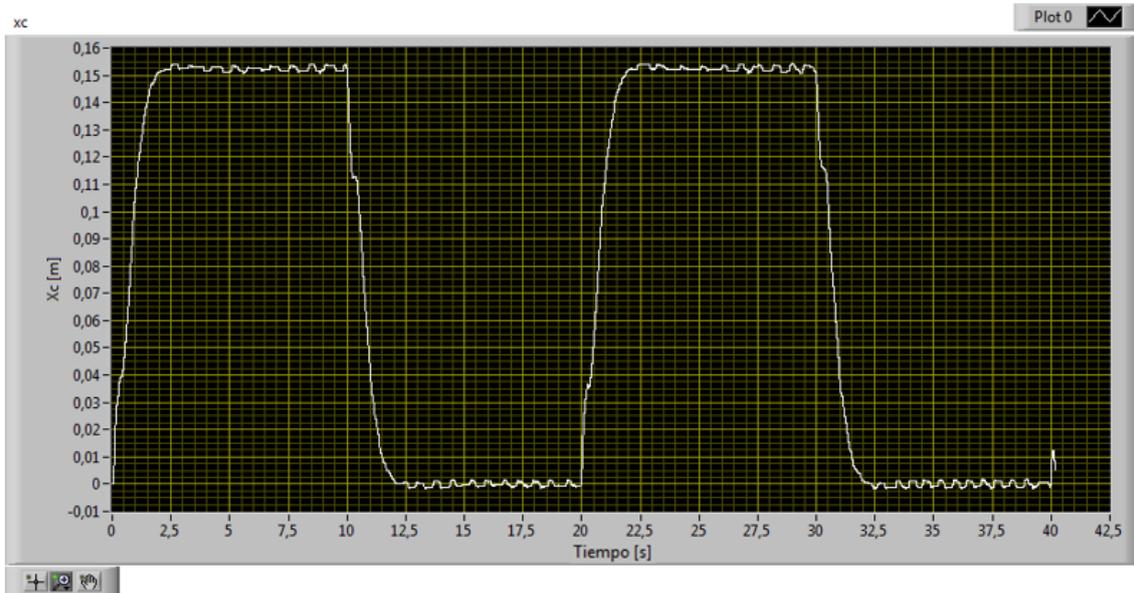


Figura 74. Desplazamiento del carro en la prueba

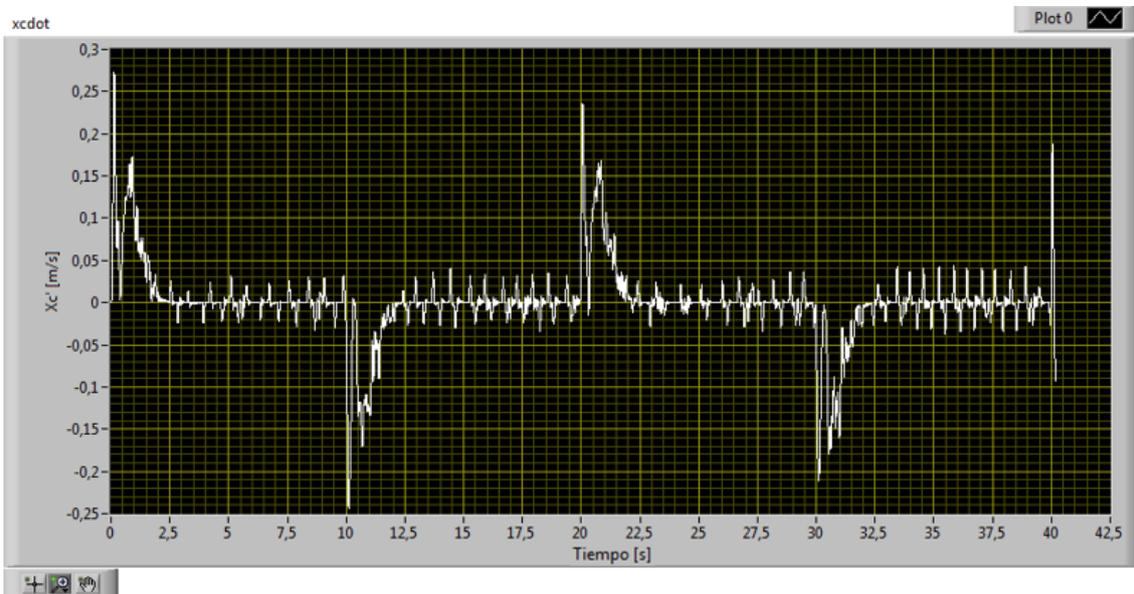


Figura 75. Velocidad del carro en la prueba

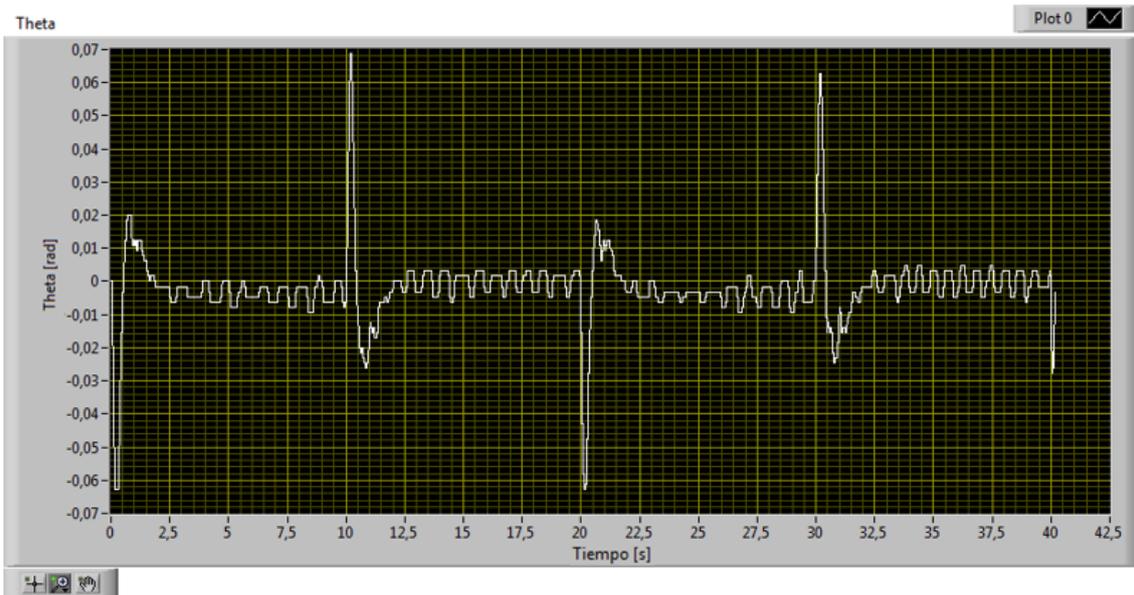


Figura 76. Ángulo del péndulo Gantry en la prueba

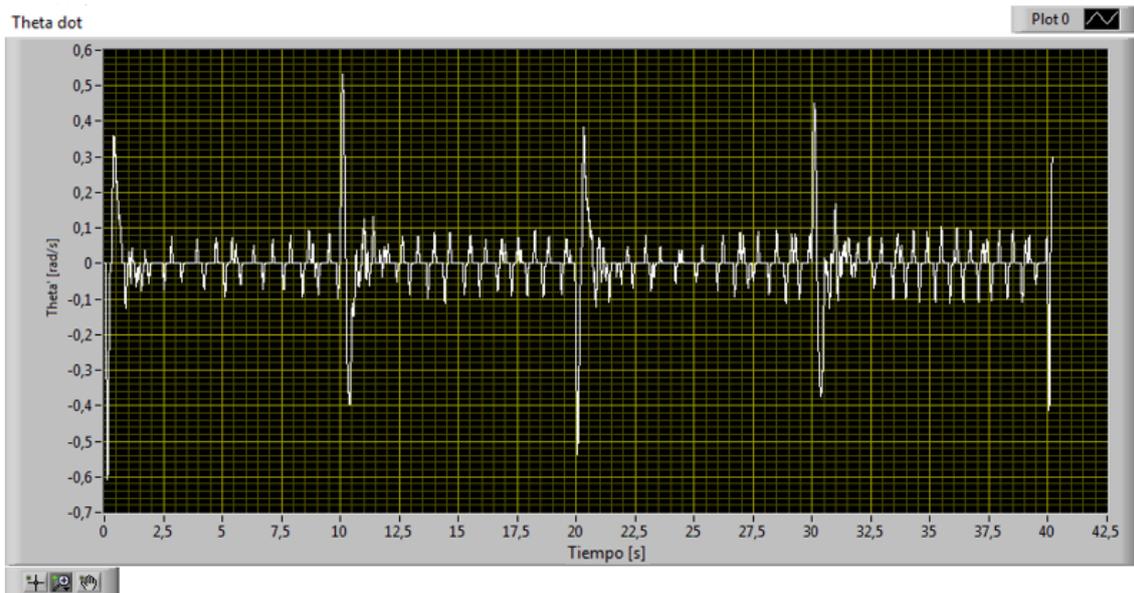


Figura 77. Velocidad angular del péndulo Gantry en la prueba

### **6.5.2.3. Controlador de realimentación del estado LQR**

El esquema del controlador es el mismo que en el caso anterior, como ya se ha señalado para el sistema carro, únicamente cambian las ganancias del mismo.

La matriz  $Q$  tiene dimensión cinco en este caso, la misma que el sistema ampliado. La matriz  $R$ , como tiene la misma dimensión que el número de entradas al sistema, en este caso vuelve a ser de dimensión uno.

### Péndulo Corto

En este sistema, se penaliza el error de  $x_c$  y  $\theta$ , que son las variables que definen la salida  $x_t$ , y de la integral del error de esta última. Por el contrario, se les asigna poco peso a las derivadas.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 500 \end{bmatrix} \quad (78)$$

Para  $V_m$ , de nuevo, se escoge un peso pequeño porque no interesa limitarla.

$$R = 0,3 \quad (79)$$

Con los valores  $Q$  y  $R$  anteriores, se utiliza la función LQR en LabVIEW, que da como resultado la siguiente matriz de ganancias de realimentación del estado del sistema aumentado.

$$\hat{K} = [47,50 \quad -40,70 \quad 16,00 \quad -1,80 \quad -40,80] \quad (80)$$

Por lo tanto, los valores de las ganancias del sistema son las siguientes.

$$K = [47,50 \quad -40,70 \quad 16,00 \quad -1,80] \quad (81)$$

$$K_i = 40,80 \quad (82)$$

A continuación, se presentan las figuras con los resultados obtenidos sobre el sistema real para una referencia de onda cuadrada de 0,15 metros de amplitud. Se pueden observar la salida del sistema  $x_t$ , la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\theta$  y sus respectivas derivadas.

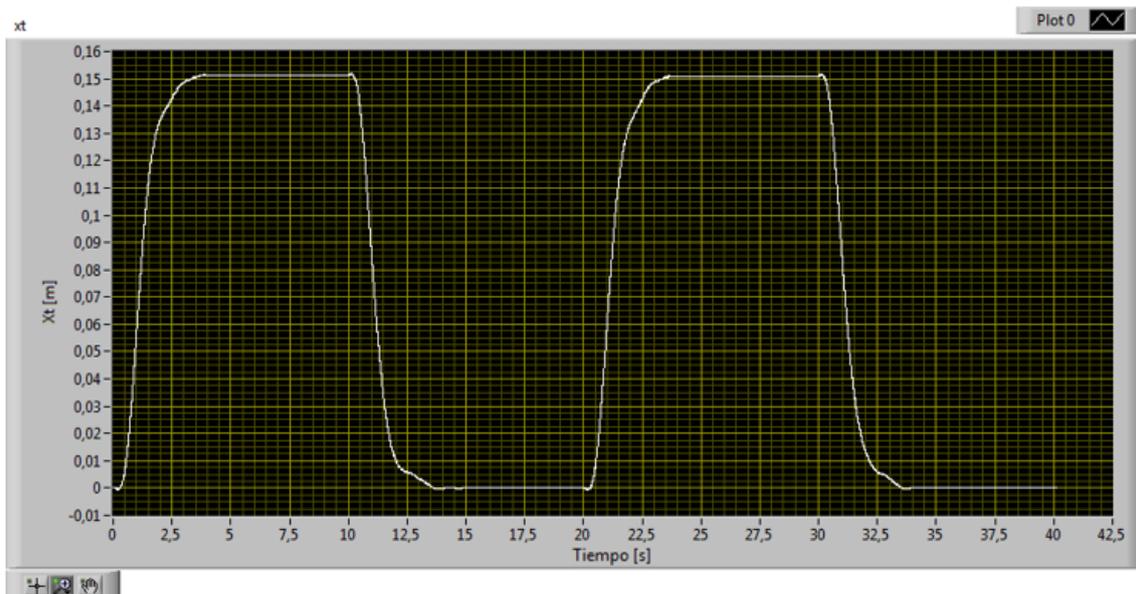


Figura 78. Salida del sistema en la prueba

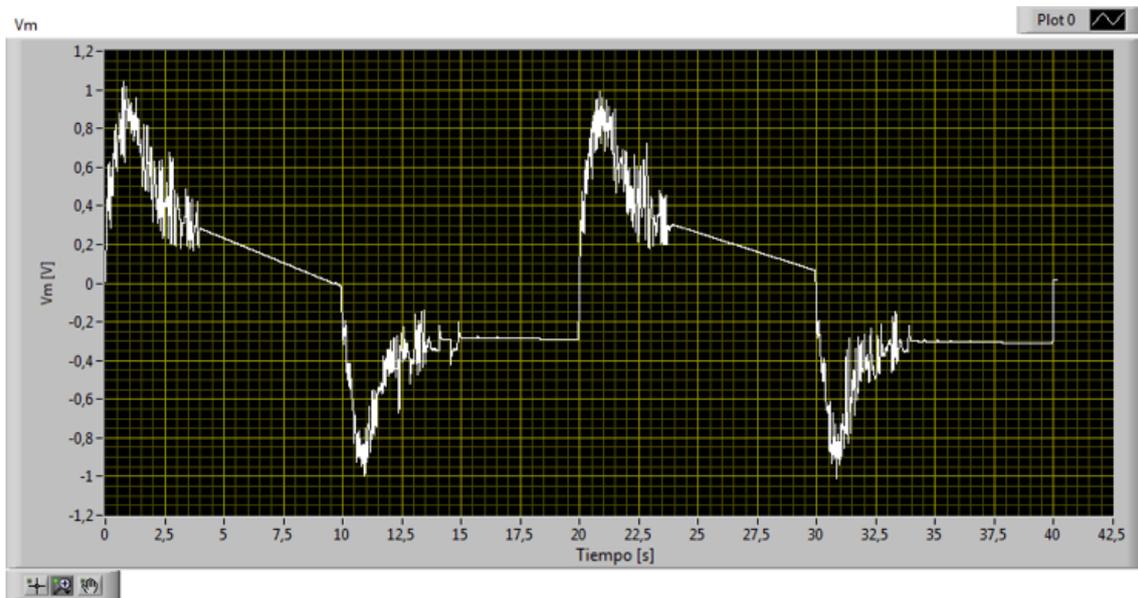


Figura 79. Tensión aplicada al péndulo Gantry en la prueba

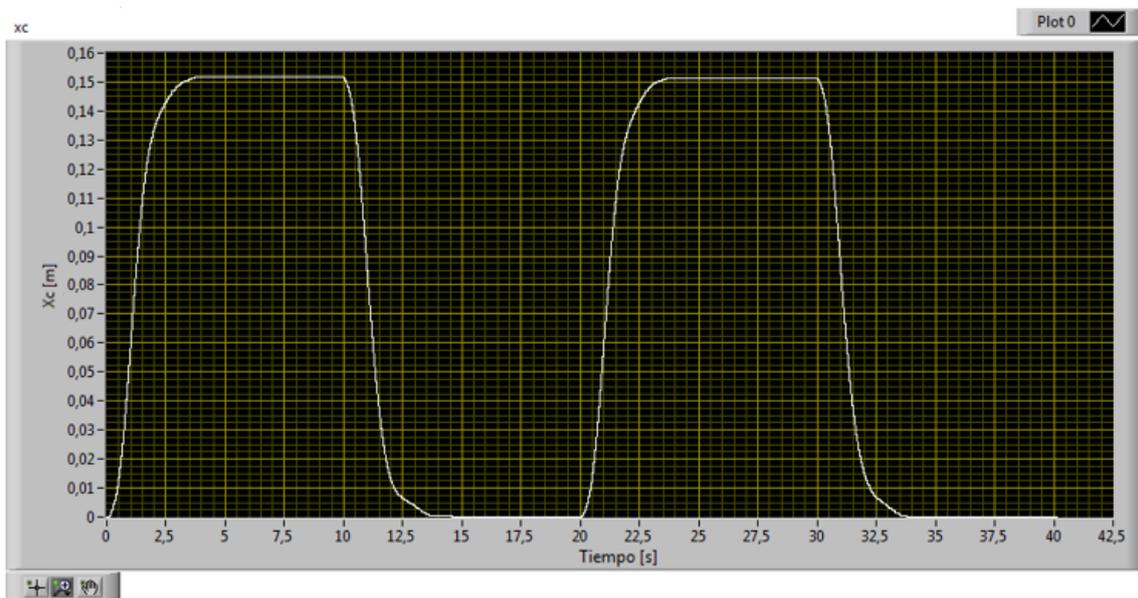


Figura 80. Desplazamiento del carro en la prueba

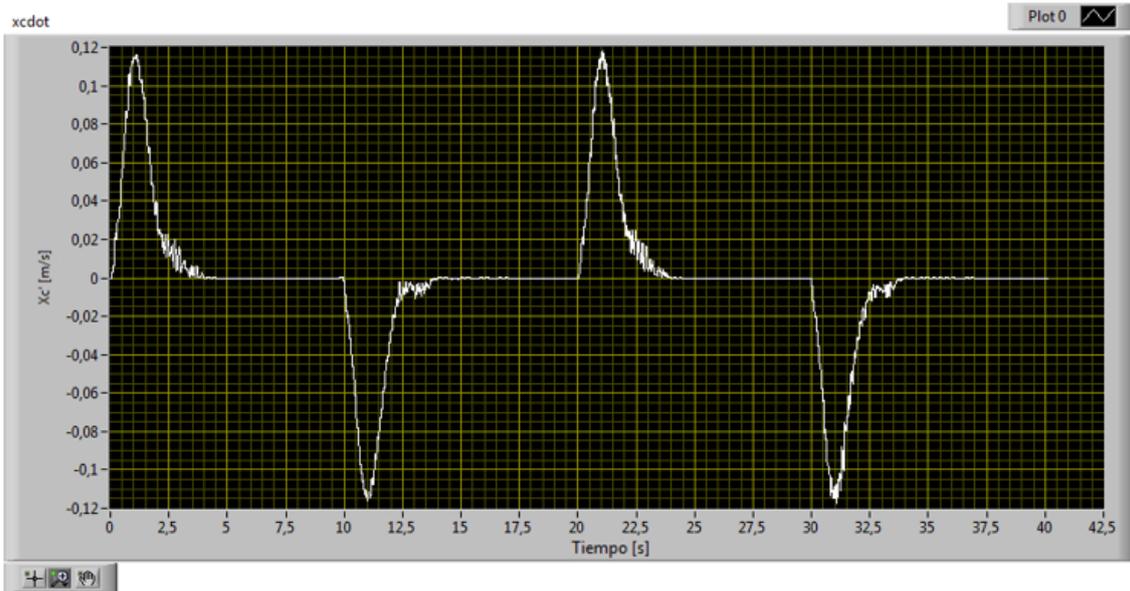


Figura 81. Velocidad del carro en la prueba

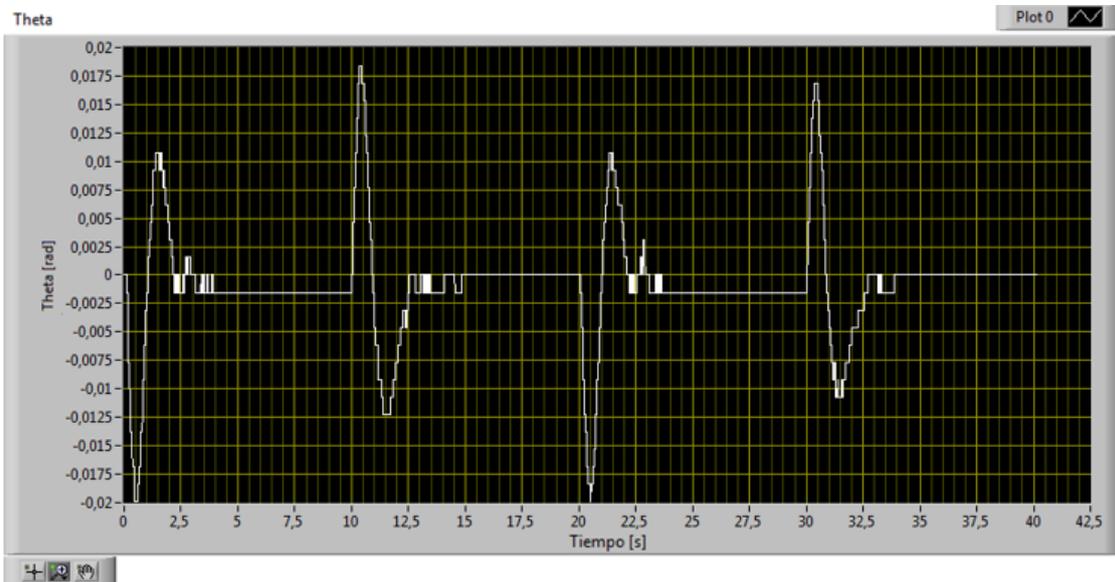


Figura 82. Ángulo del péndulo Gantry en la prueba

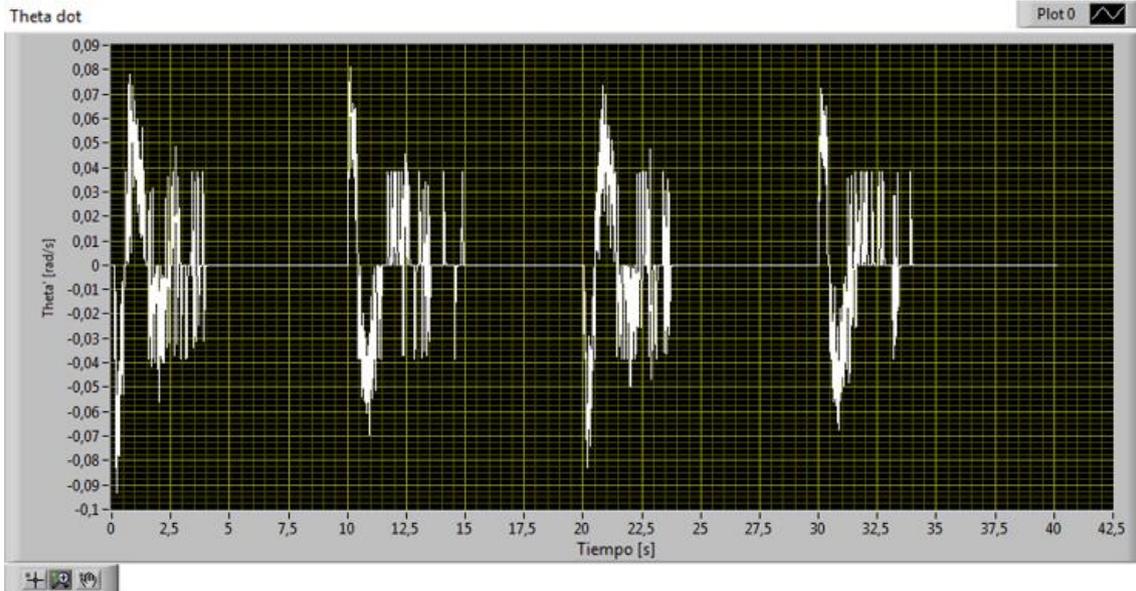


Figura 83. Velocidad angular del péndulo Gantry en la prueba

### **Péndulo Largo**

Siguiendo el mismo criterio que para el péndulo corto, la matriz  $Q$  se define con los mismos valores que en el caso anterior.

$$Q = \begin{bmatrix} 100 & 0 & 0 & 0 & 0 \\ 0 & 100 & 0 & 0 & 0 \\ 0 & 0 & 4 & 0 & 0 \\ 0 & 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 0 & 500 \end{bmatrix} \quad (83)$$

Para  $V_m$  se elige un valor de ponderación pequeño, pero esta vez distinto, debido al proceso de prueba y error en el que, con este valor, se consiguen mejores resultados.

$$R = 0,05 \quad (84)$$

Con los valores  $Q$  y  $R$  anteriores, se vuelve a usar la función LQR en LabVIEW, para obtener la matriz de ganancias de realimentación del estado del sistema aumentado.

$$\hat{K} = [113,42 \quad -77,72 \quad 45,97 \quad 7,31 \quad -100] \quad (85)$$

Por lo tanto, los valores de las ganancias del sistema son las siguientes.

$$K = [113,42 \quad -77,72 \quad 45,97 \quad 7,31] \quad (86)$$

$$K_i = 100 \quad (87)$$

De nuevo, las figuras siguientes muestran la salida del sistema  $x_t$ , la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\theta$  y sus respectivas derivadas, obtenidas en una prueba sobre el sistema real con una onda cuadrada de 0,15 metros de amplitud como referencia.

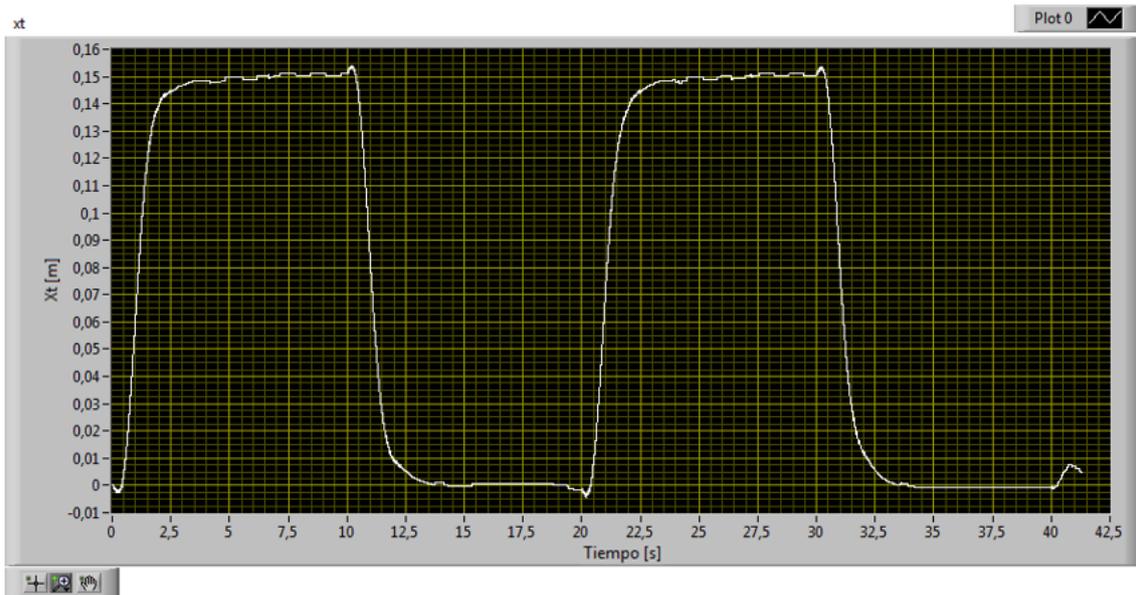


Figura 84. Salida del sistema en la prueba

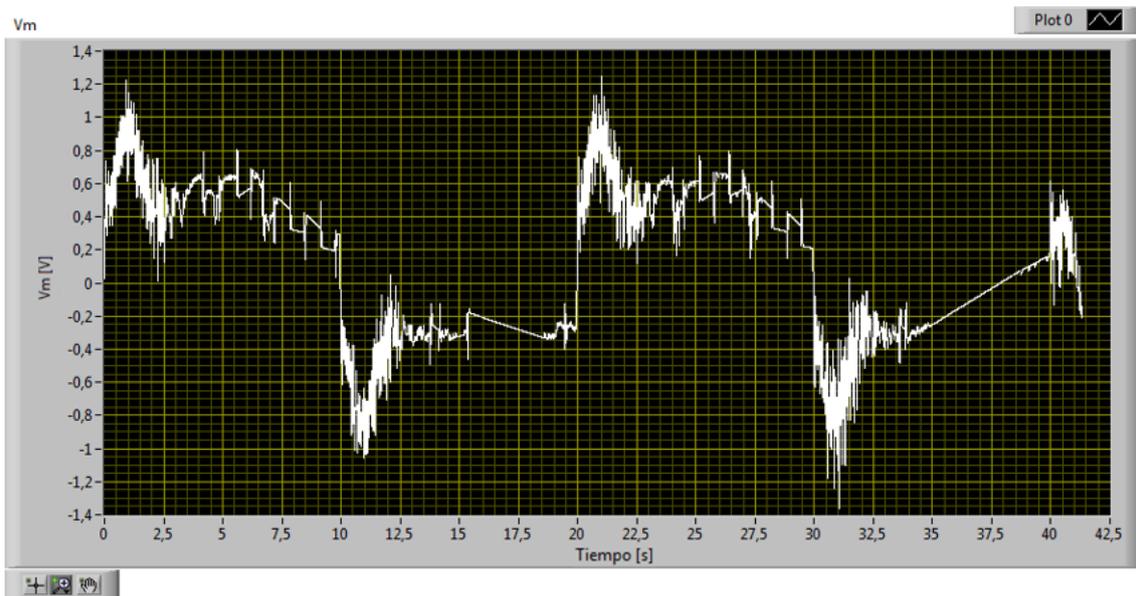


Figura 85. Tensión aplicada al péndulo Gantry en la prueba

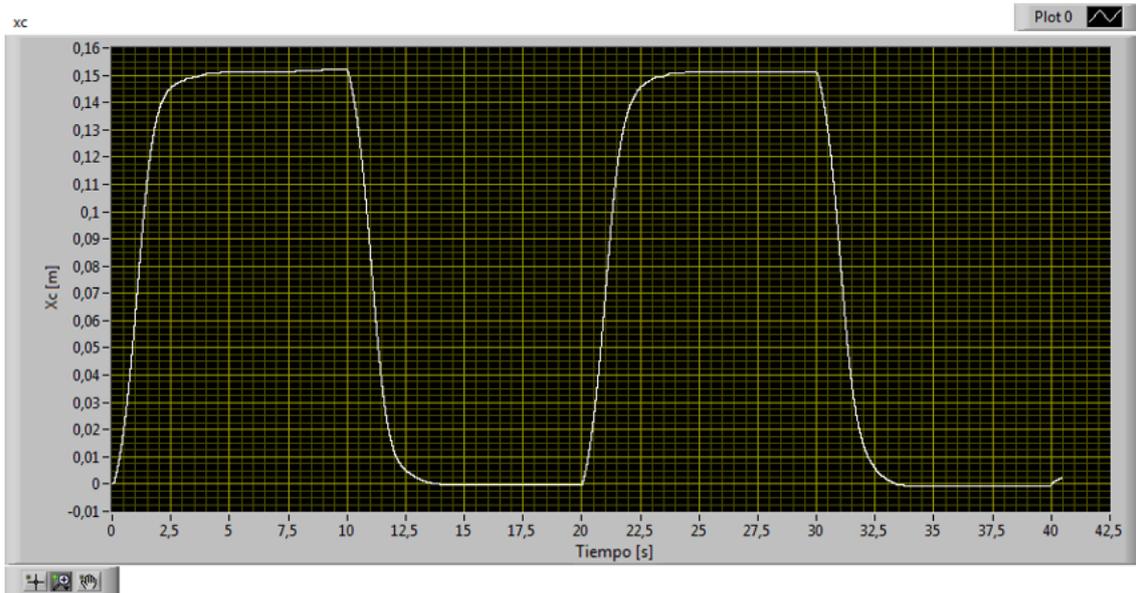


Figura 86. Desplazamiento del carro en la prueba

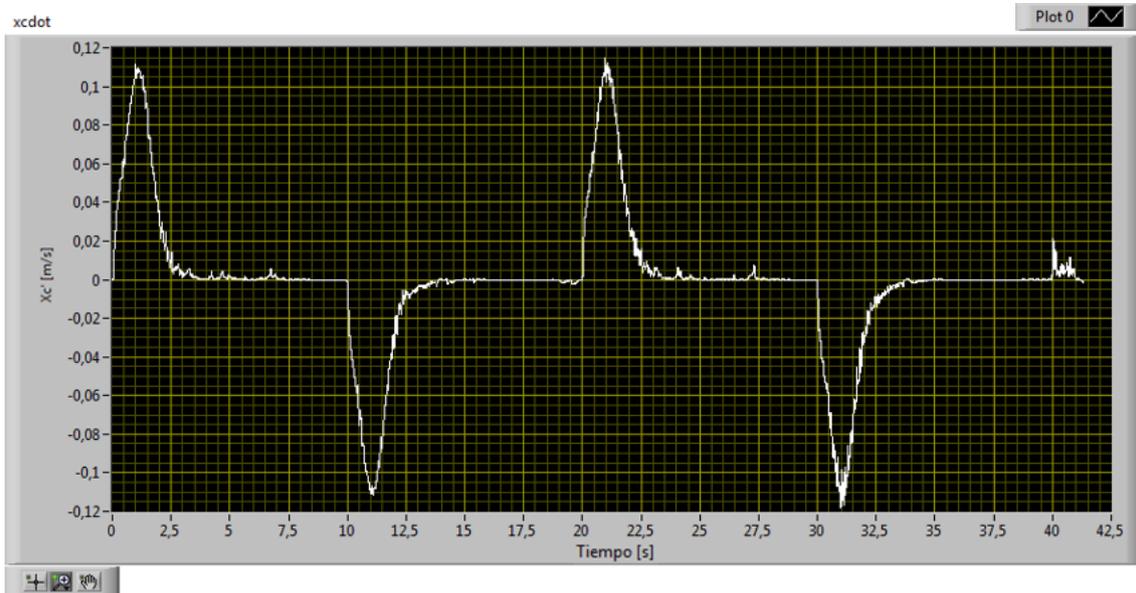


Figura 87. Velocidad del carro en la prueba

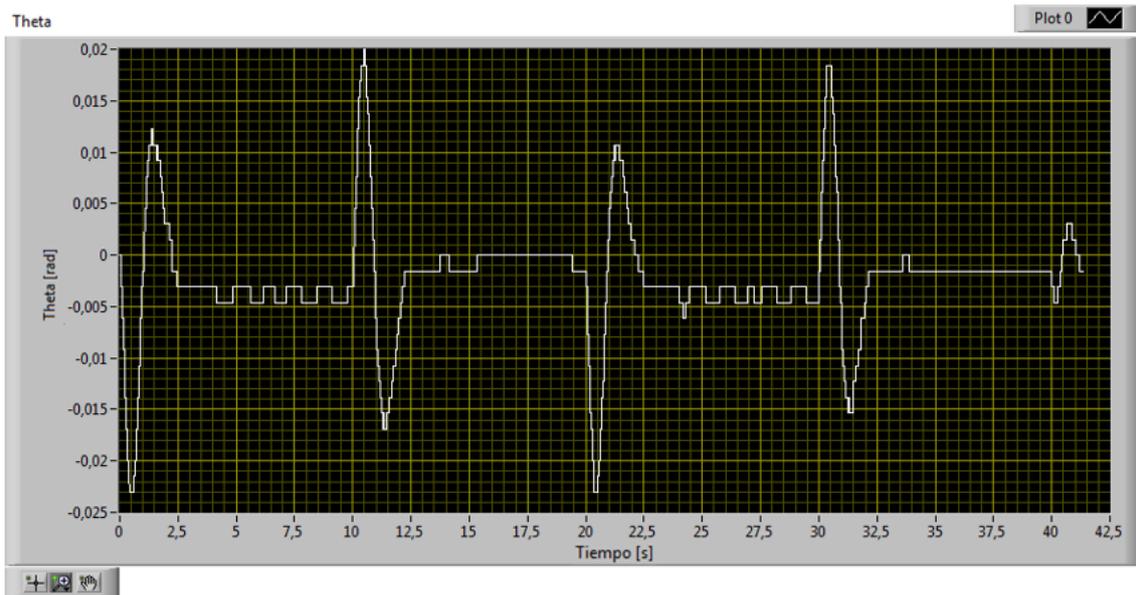


Figura 88. Ángulo del péndulo Gantry en la prueba

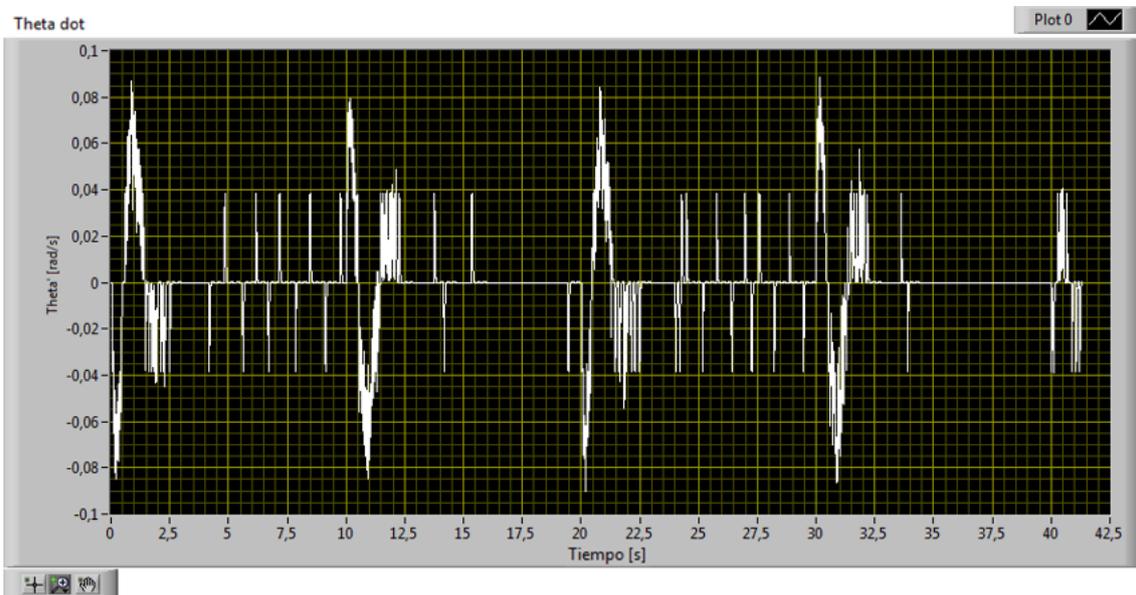


Figura 89. Velocidad angular del péndulo Gantry en la prueba

## **6.5.3. SISTEMA PENDULO INVERTIDO**

El control del sistema “péndulo invertido” requiere como primera acción llevar al péndulo a esa posición partiendo del péndulo en su posición inferior de reposo. Esto puede hacerse bien manualmente, bien automáticamente mediante un algoritmo llamado “autoerección”.

Una vez que el péndulo está en la posición vertical superior se aplican los distintos algoritmos de control para mantener el péndulo en esa posición superior.

En el caso de la autoerección, la estrategia consiste en ejecutar primeramente un algoritmo cuya finalidad es llevar el péndulo desde su posición de reposo inferior a la posición vertical superior.

En ese instante, se produce una conmutación entre este algoritmo y los que tienen como objetivo mantener el péndulo en la posición vertical superior. Entre estos últimos se utilizarán algoritmos PID, LQR, etc...

Primeramente, se explican estos últimos algoritmos, ya que se han diseñado primero, llevando el péndulo a la posición superior manualmente. Como las señales de control generadas para el control del péndulo invertido en modo de control de balanceo son muy ruidosas, debido a la gran inestabilidad del sistema, se introduce en estos controladores un filtro pasa bajos con frecuencia de corte a 40Hz después de la salida del controlador y antes del saturador.

### **6.5.3.1. Controlador PID**

En este sistema, es necesario controlar dos variables,  $x_c$  y  $\alpha$ , siendo el control de ambas crítico para el correcto funcionamiento del sistema. Por lo tanto, y dado que el control en cascada vuelve a resultar insatisfactorio, se utiliza la misma estructura que en el caso del sistema péndulo Gantry, con dos algoritmos PID, uno para cada variable, siendo la suma de las acciones de control generadas por éstos la que actúa sobre el motor. Por supuesto, mientras que la referencia para  $x_c$  es el movimiento del carro que se desea, la referencia del ángulo  $\alpha$  ha de ser cero. Como se ha indicado, la acción de control, antes de llegar al saturador, se filtra con un filtro pasa bajos con una frecuencia de corte a 40Hz.

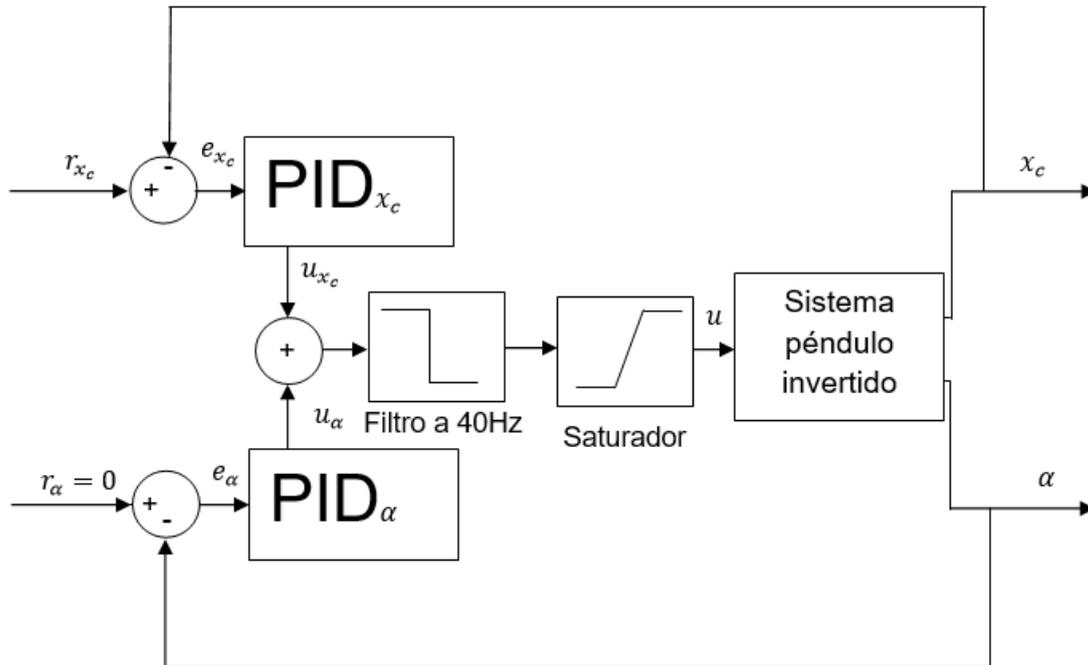


Figura 90. Estructura del controlador PID para el péndulo invertido

Para el sistema péndulo invertido, también es necesario utilizar integrar en  $x_c$  para asegurar un error nulo en estado estacionario ante entrada escalón. La integración en  $\alpha$  no es tan importante, ya que el controlador tratará de mantener  $\alpha = 0$ , ya que, de otra manera, el sistema no sería estable.

Para la selección de las ganancias de ambos controladores se pueden hacer las mismas consideraciones que para el péndulo Gantry.

### **Péndulo Corto**

Teniendo en cuenta esas consideraciones, se seleccionan para el péndulo corto los siguientes valores:

$$PID_{x_c}: \quad K_c = -51 \text{ V/m} \quad T_i = 1,35 \text{ min} \quad T_d = 0,01 \text{ min} \quad (88)$$

$$PID_{\alpha}: \quad K_c = 54 \text{ V/rad} \quad T_i = 500 \text{ min} \quad T_d = 0,0024 \text{ min} \quad (89)$$

Los resultados en el sistema real para una entrada con forma de onda cuadrada de 0,15 metros de amplitud son los que se muestran en las siguientes figuras. Se registran las salidas del sistema  $x_c$  y  $\alpha$ , y la acción de control  $V_m$ .

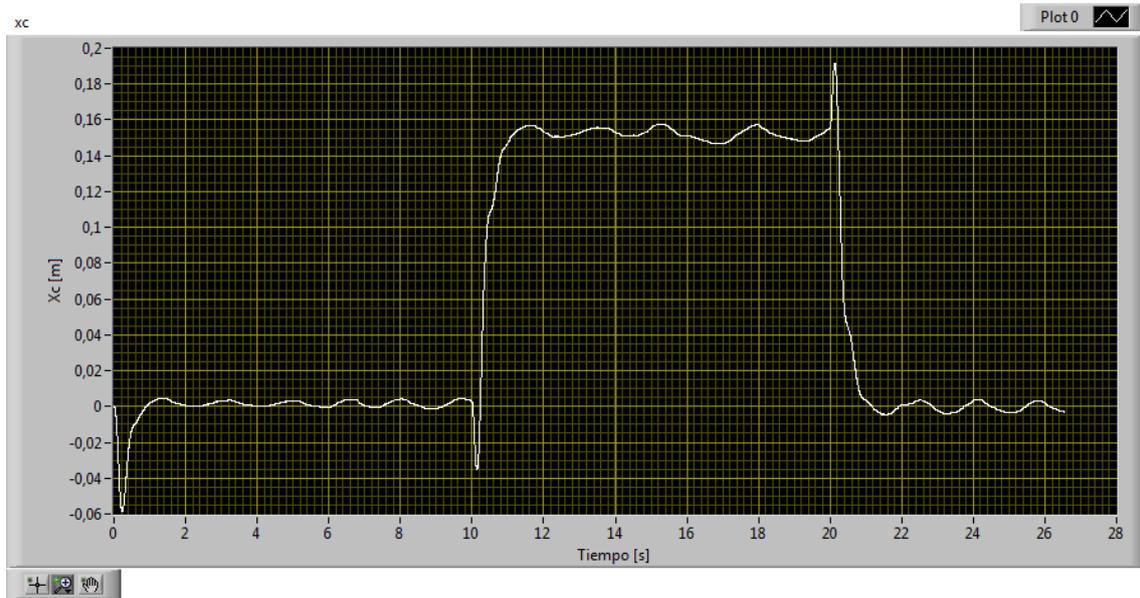


Figura 91. Desplazamiento del carro en la prueba

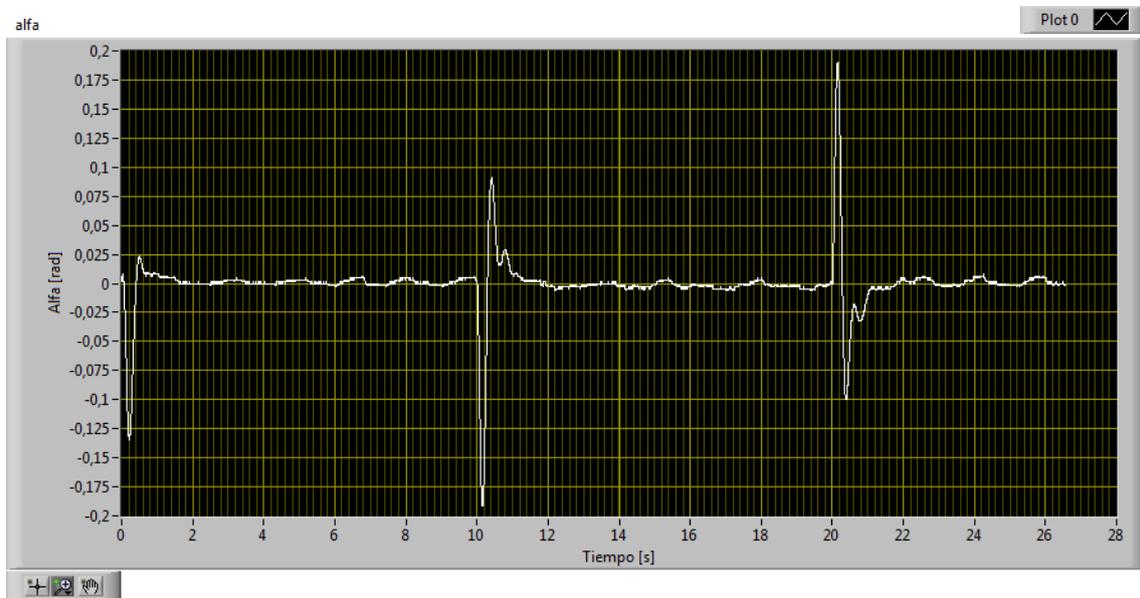


Figura 92. Ángulo del péndulo invertido en la prueba

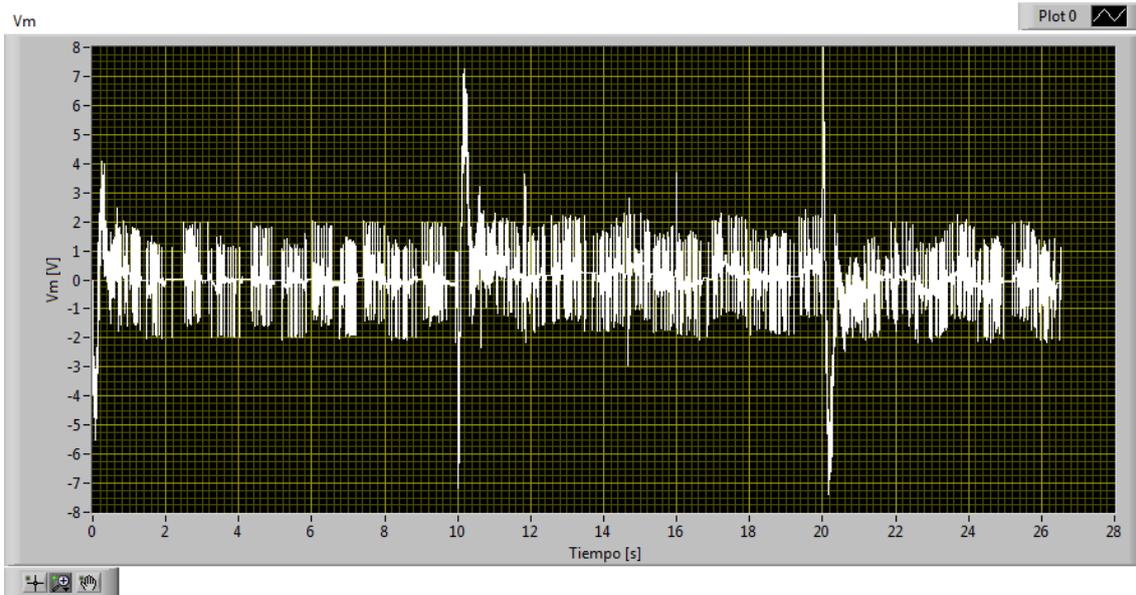


Figura 93. Tensión aplicada al péndulo invertido en la prueba

### Péndulo Largo

En el caso del péndulo largo, los valores seleccionados son los siguientes.

$$PID_{x_c}: \quad K_c = -50 \text{ V/m} \quad T_i = 5 \text{ min} \quad T_d = 0,018 \text{ min} \quad (90)$$

$$PID_{\alpha}: \quad K_c = 110 \text{ V/rad} \quad T_i = \infty \text{ min} \quad T_d = 0,0035 \text{ min} \quad (91)$$

A continuación, se muestran las salidas del sistema  $x_c$  y  $\alpha$ , y la acción de control  $V_m$  en el sistema real para una entrada con forma de onda cuadrada de 0,15 metros de amplitud.

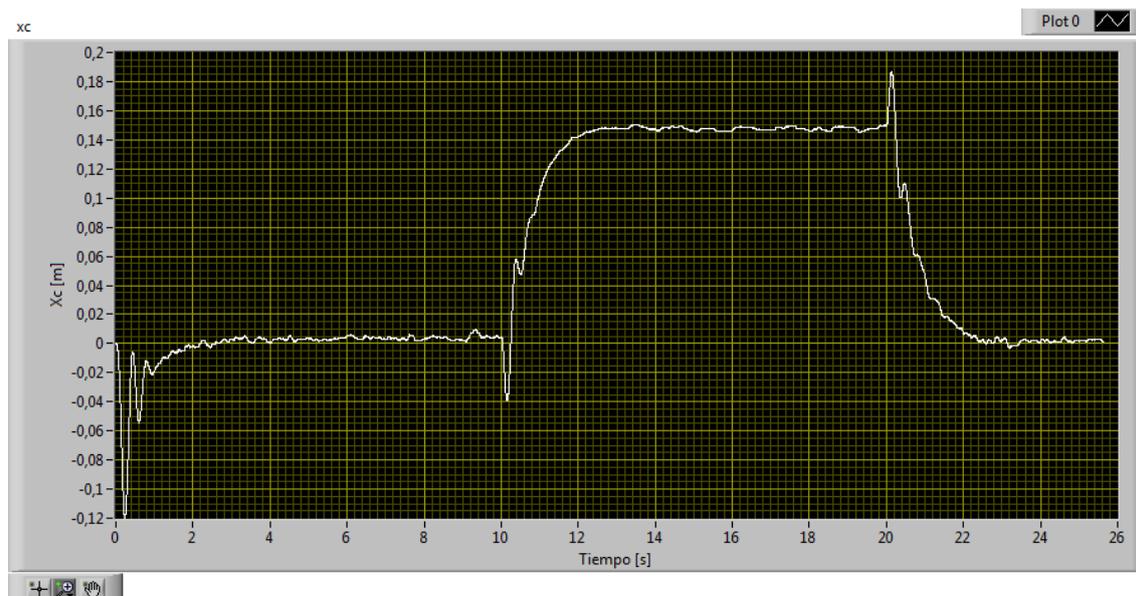


Figura 94. Desplazamiento del carro en la prueba

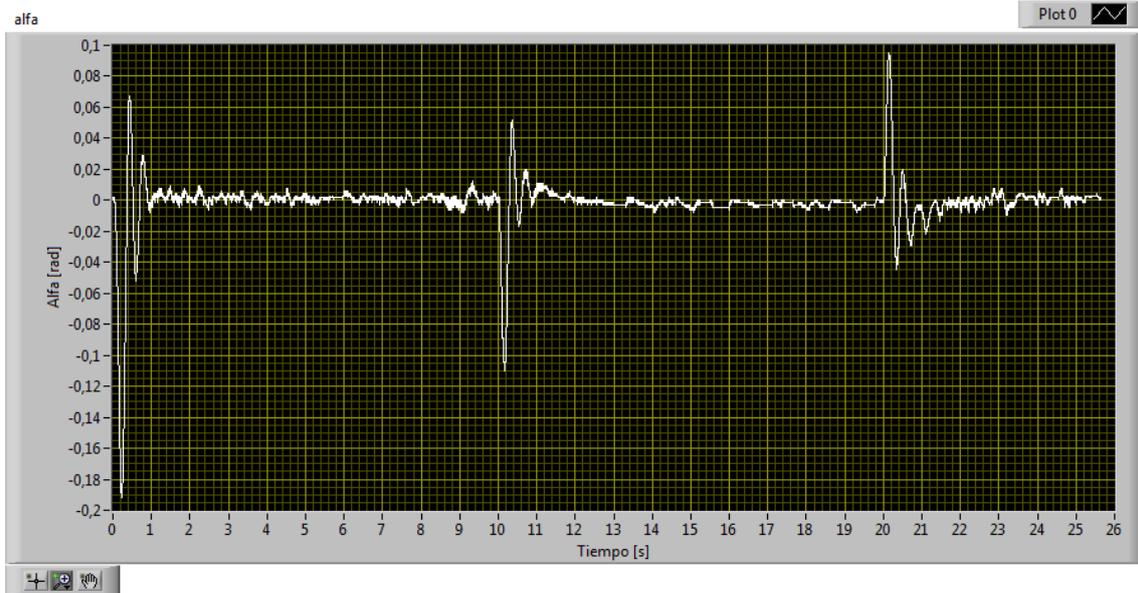


Figura 95. Ángulo del péndulo invertido en la prueba

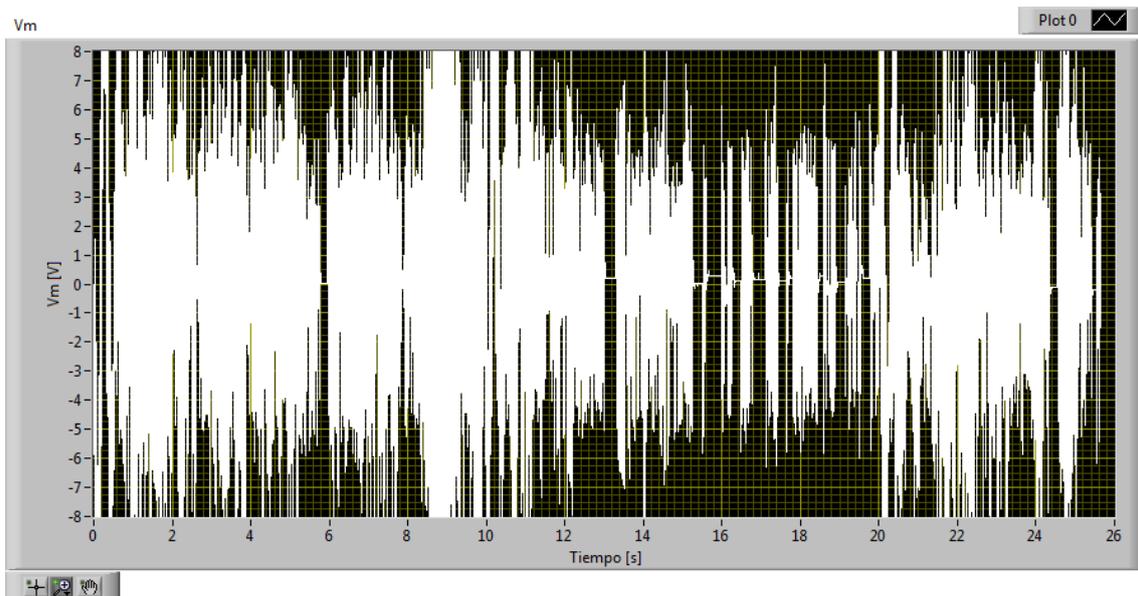


Figura 96. Tensión aplicada al péndulo invertido en la prueba

### 6.5.3.2. Controlador de realimentación del estado por asignación de polos

A pesar de lo dicho anteriormente, se ha probado una doble integración, añadiendo dos estados al sistema, las integrales del error de  $x_c$  y  $\alpha$ . Sin embargo, en las simulaciones realizadas con el modelo, el valor correspondiente a la constante de integración del ángulo  $\alpha$  obtenido mediante la fórmula de Ackermann, que calcula la matriz de ganancias del sistema aumentado, ha sido siempre cero. Esto se debe a que el control elimina el error estacionario porque de otra manera, si existiera un error estacionario en el ángulo  $\alpha$ , el sistema sería inestable.

Por lo tanto, únicamente se integra en la variable  $x_c$ , y el esquema de control es muy parecido al caso del péndulo Gantry, aunque en vez de realimentarse  $x_t$ , que no existe en este sistema, se realimenta  $x_c$ . Además, como ya se ha mencionado en la

obtención de los modelos, en este sistema el ángulo se denomina  $\alpha$ . Por último, la acción de control se filtra antes del saturador con un filtro pasa bajos a 40 Hz.

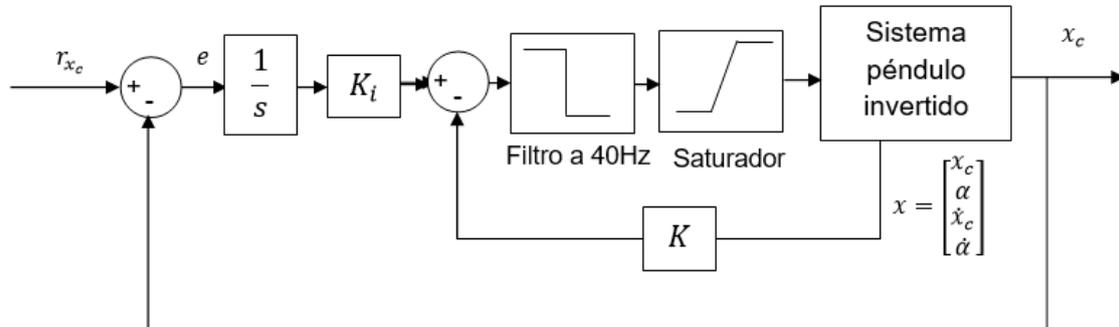


Figura 97. Estructura del controlador por realimentación del estado para el péndulo invertido

En este sistema, de nuevo, han de elegirse cinco polos para definir la dinámica del mismo en lazo cerrado. Como en el sistema anterior, se eligen dos polos conjugados dominantes mediante la elección del sobreoscilamiento y el tiempo de establecimiento deseados, y los otros tres polos se escogen alejados y en el eje real negativo.

### Péndulo Corto

Para el péndulo corto, se eligen  $SO = 0,05\%$  y  $T_s = 2s$ , valores con los que se obtienen los polos conjugados  $p_{1,2} = -2 \pm 0,827j$ . Los otros tres polos se eligen en el eje real negativo.

$$p_3 = -7 \quad (92)$$

$$p_4 = -11 \quad (93)$$

$$p_5 = -15 \quad (94)$$

La fórmula de Ackermann da el siguiente valor de la matriz de ganancias del sistema aumentado:

$$\hat{K} = [-51,41 \quad 54,14 \quad -30,70 \quad 7,76 \quad 44,53] \quad (95)$$

donde  $K = [-51,41 \quad 54,14 \quad -30,70 \quad 7,76]$  y  $K_i = -44,53$ .

La respuesta obtenida en el sistema real para referencia de  $x_c$  cero se muestra en las figuras siguientes, representándose la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\alpha$  y sus respectivas derivadas.

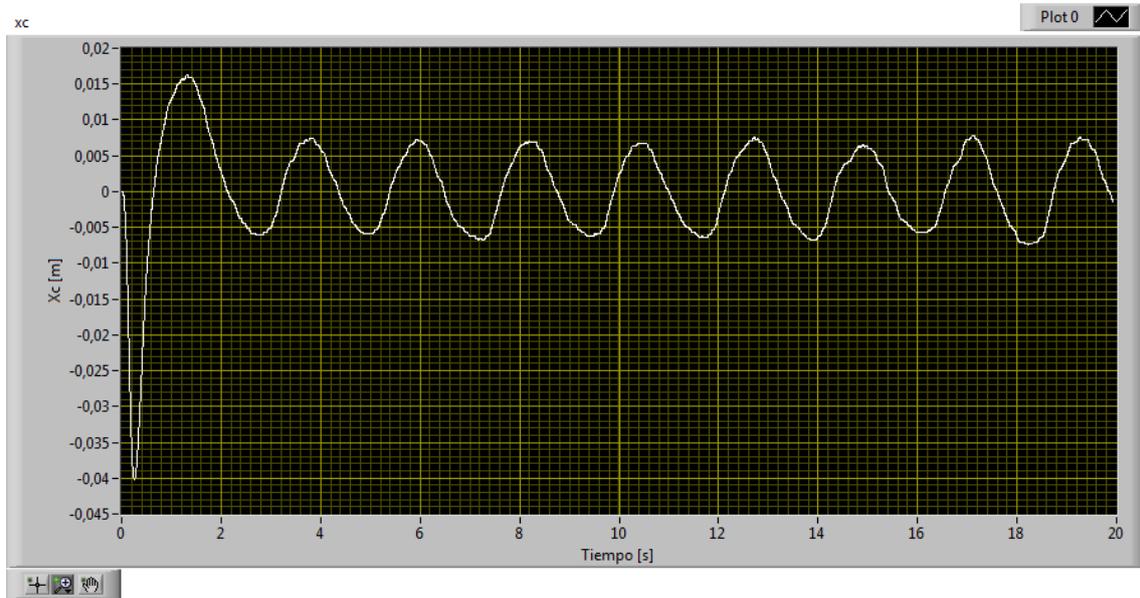


Figura 98. Desplazamiento del carro en la prueba

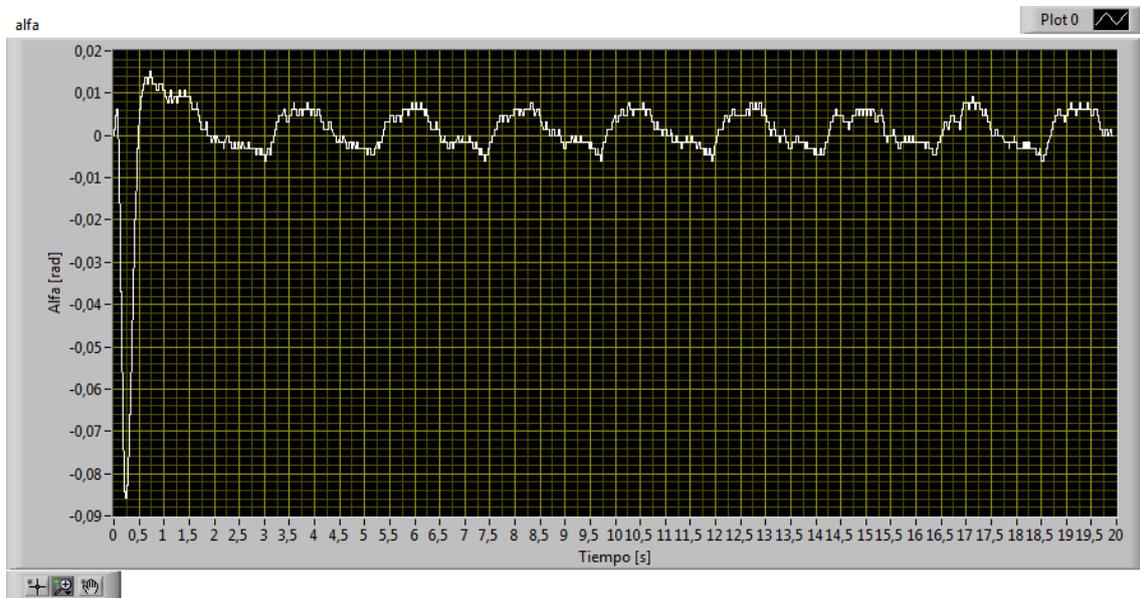


Figura 99. Ángulo del péndulo invertido en la prueba

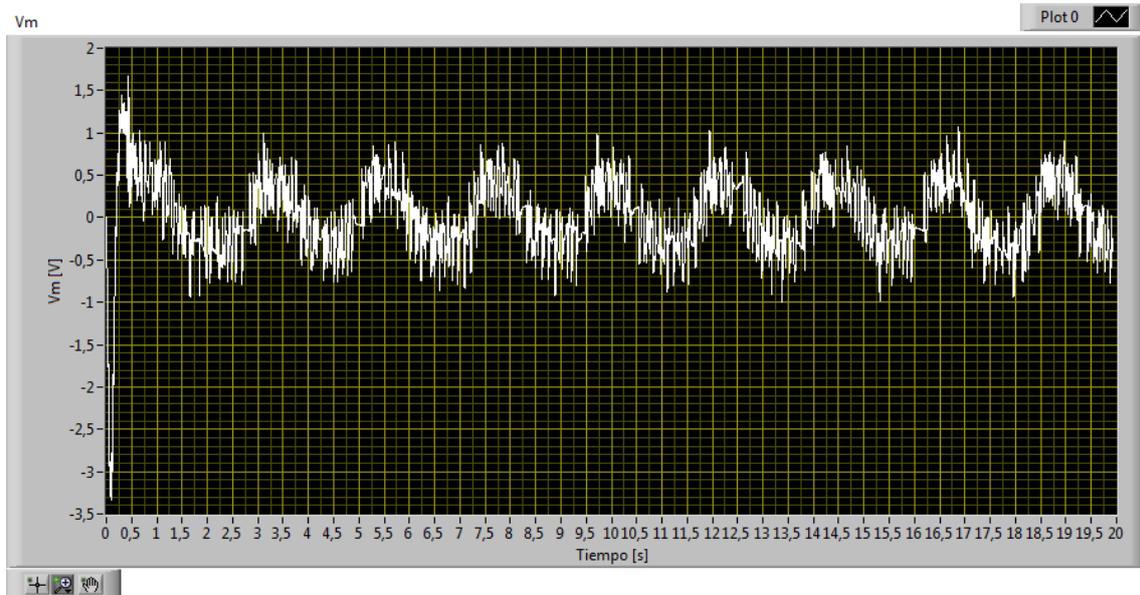


Figura 100. Tensión aplicada al péndulo invertido en la prueba

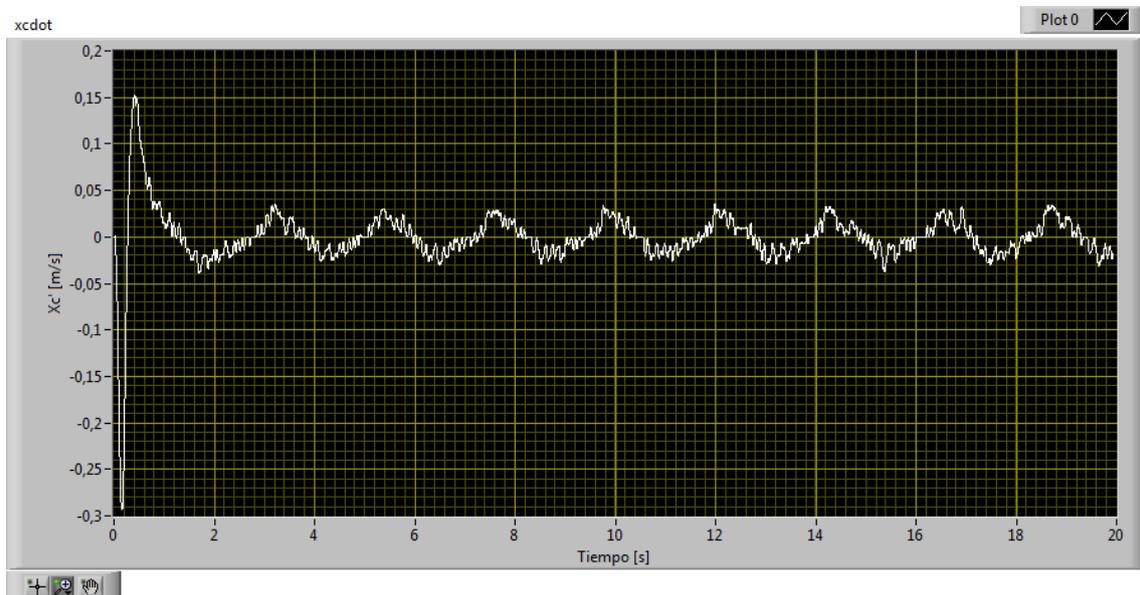


Figura 101. Velocidad del carro en la prueba

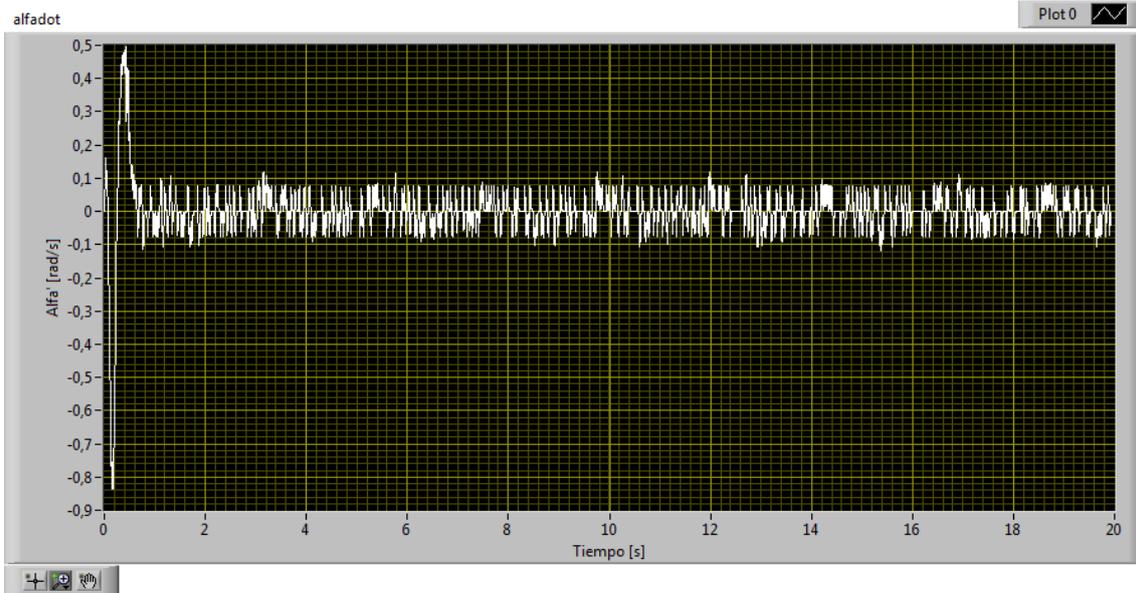


Figura 102. Velocidad angular del péndulo Gantry en la prueba

Si, por el contrario, se decide permitir un sobreoscilamiento mayor, se pueden seleccionar las especificaciones  $SO = 2\%$  y  $T_s = 2,2s$ . Estos valores dan los polos conjugados  $p_{1,2} = -1,818 \pm 1,460j$ . El resto de polos se mantienen.

Con la fórmula de Ackermann se calcula el siguiente valor de la matriz de ganancias del sistema aumentado:

$$\hat{K} = [-50,09 \quad 52,96 \quad -30,02 \quad 7,59 \quad 51,69] \quad (96)$$

donde  $K = [-50,09 \quad 52,96 \quad -30,02 \quad 7,59]$  y  $K_i = -51,69$ .

En las figuras siguientes se muestran la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\alpha$  y sus respectivas derivadas en el sistema real. La respuesta obtenida es para referencia de onda cuadrada de amplitud de 0,15 metros.

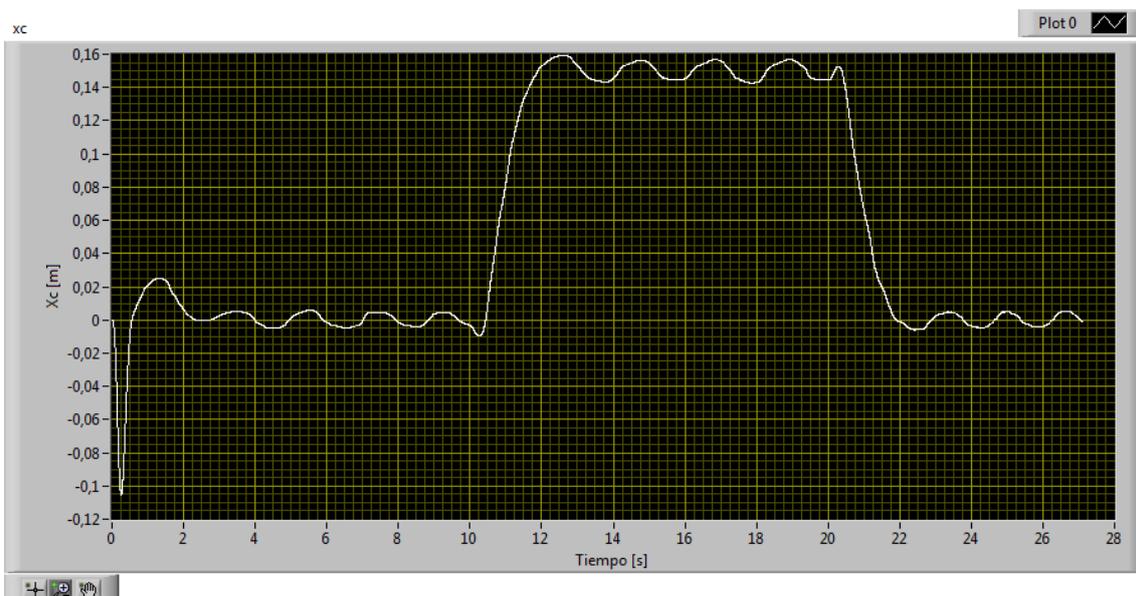


Figura 103. Desplazamiento del carro en la prueba

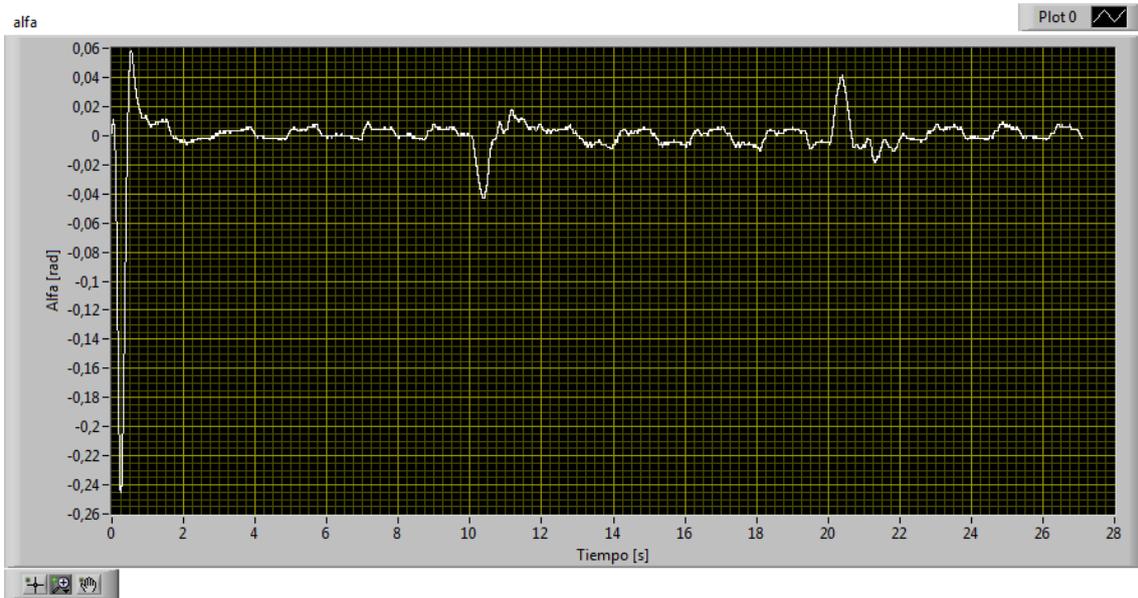


Figura 104. Ángulo del péndulo invertido en la prueba

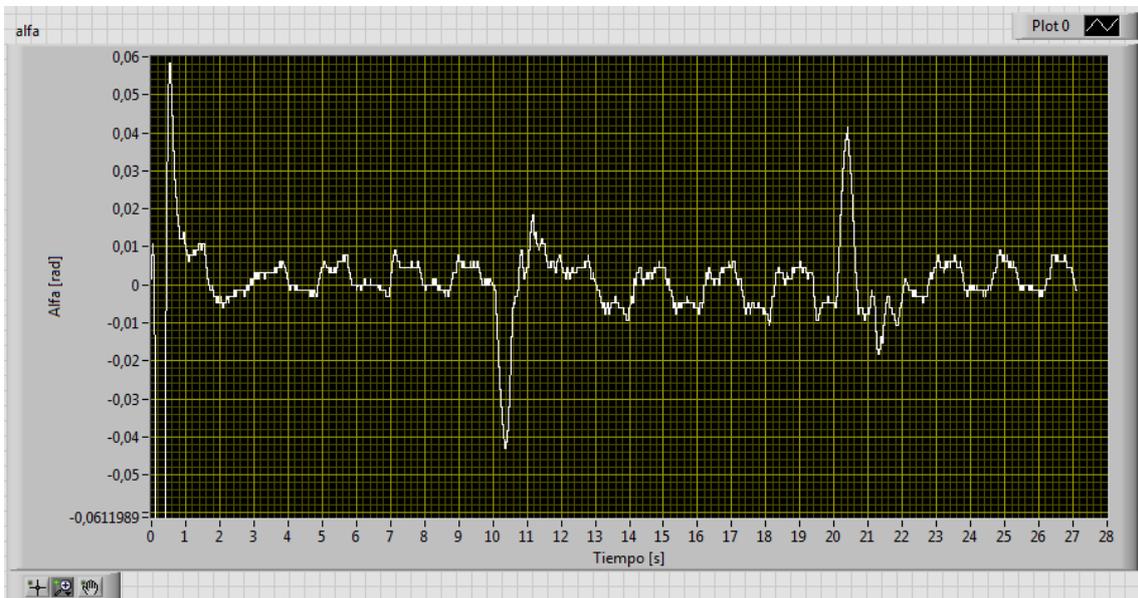


Figura 105. Detalle del ángulo del péndulo invertido en la prueba

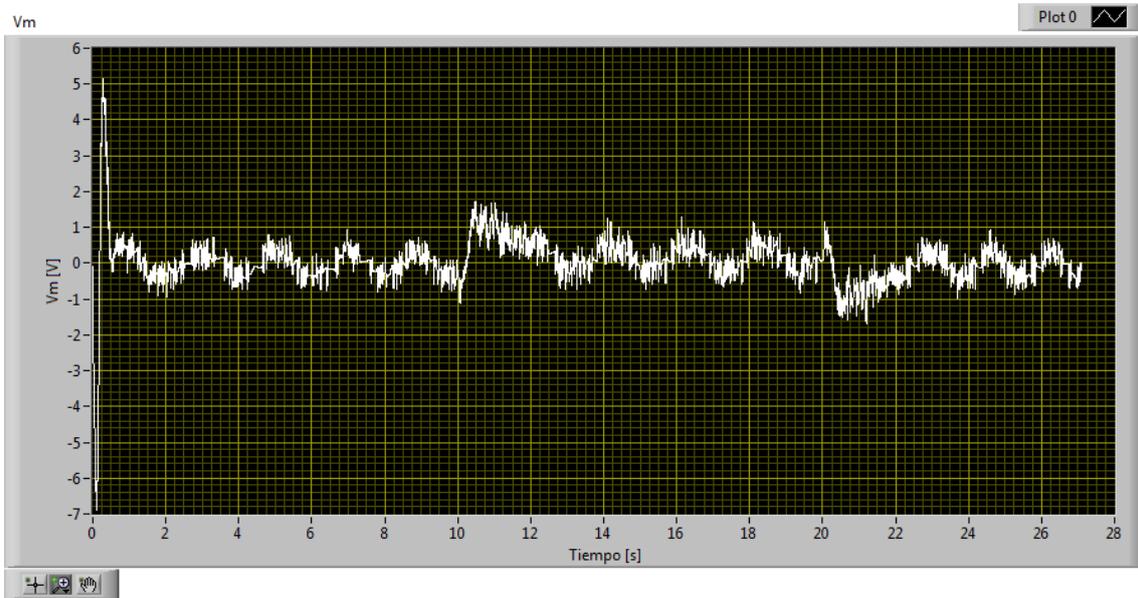


Figura 106. Tensión aplicada al péndulo invertido en la prueba

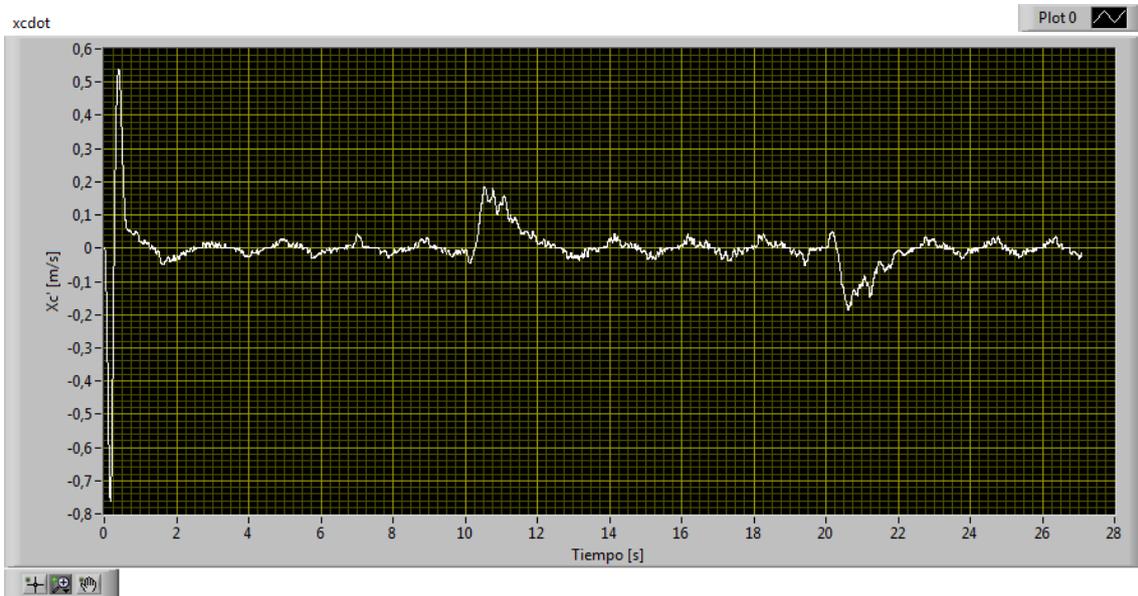


Figura 107. Velocidad del carro en la prueba

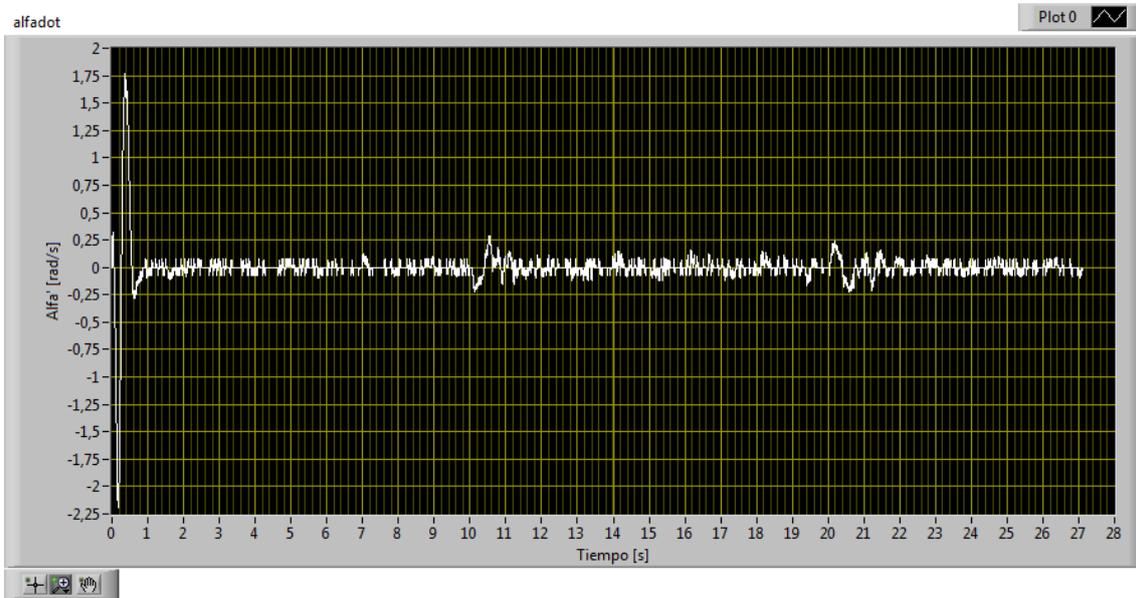


Figura 108. Velocidad angular del péndulo invertido en la prueba

### **Péndulo Largo**

En el caso del péndulo largo, se eligen las especificaciones de  $SO = 0,05\%$  y  $T_s = 2s$ , valores con los que se obtienen los polos conjugados  $p_{1,2} = -2 \pm 0,827j$ . Los otros tres polos se eligen, como siempre, se eligen alejados en el eje real negativo.

$$p_3 = -7 \quad (97)$$

$$p_4 = -11 \quad (98)$$

$$p_5 = -15 \quad (99)$$

La resolución de la fórmula de Ackermann devuelve el valor de la matriz de ganancias del sistema aumentado que se muestra a continuación.

$$\hat{K} = [-114,57 \quad 142,57 \quad -62,29 \quad 30,38 \quad 99,25] \quad (100)$$

donde  $K = [-114,57 \quad 142,57 \quad -62,29 \quad 30,38]$  y  $K_i = -99,25$ .

La respuesta obtenida en el sistema real ante una referencia de onda cuadrada de 0,15 metros de amplitud se muestra en las figuras siguientes, representándose la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\alpha$  y sus respectivas derivadas.

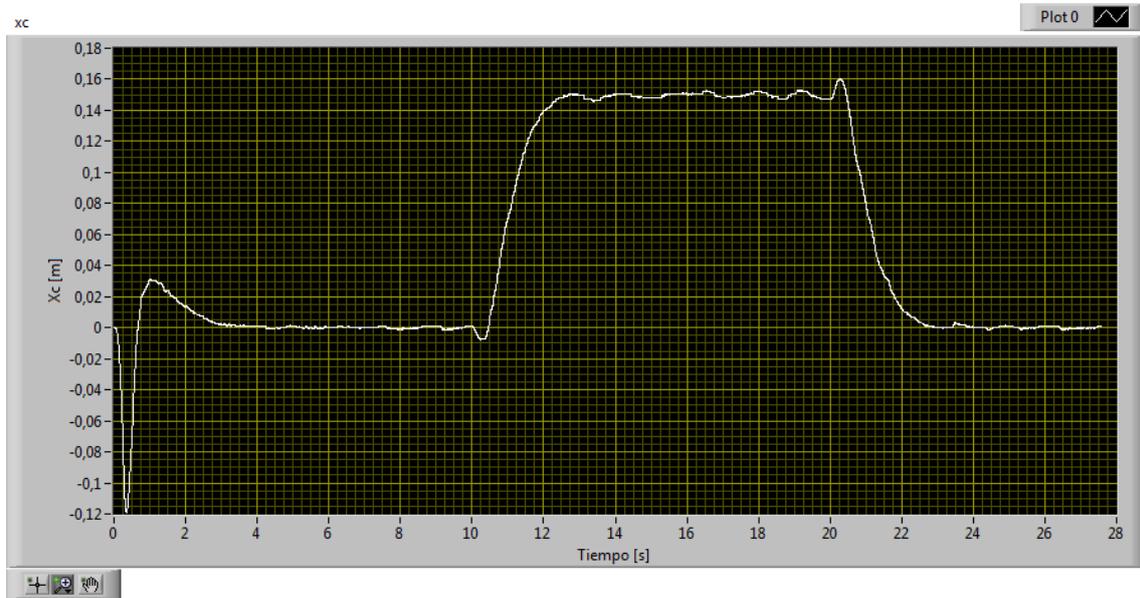


Figura 109. Desplazamiento del carro en la prueba

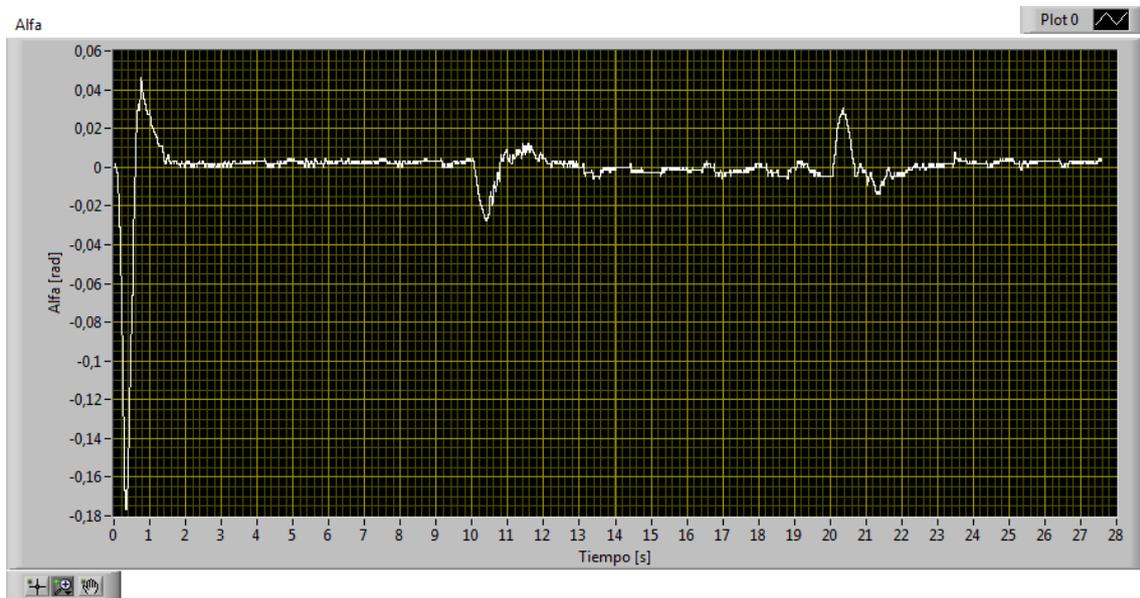


Figura 110. Ángulo del péndulo invertido en la prueba

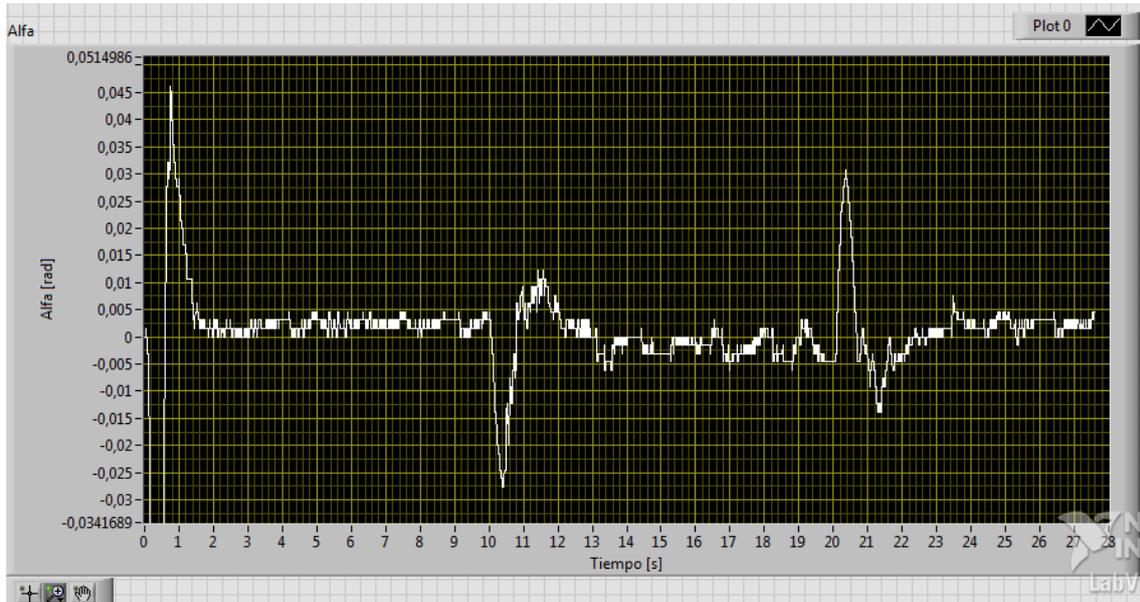


Figura 111. Detalle del ángulo del péndulo invertido en la prueba

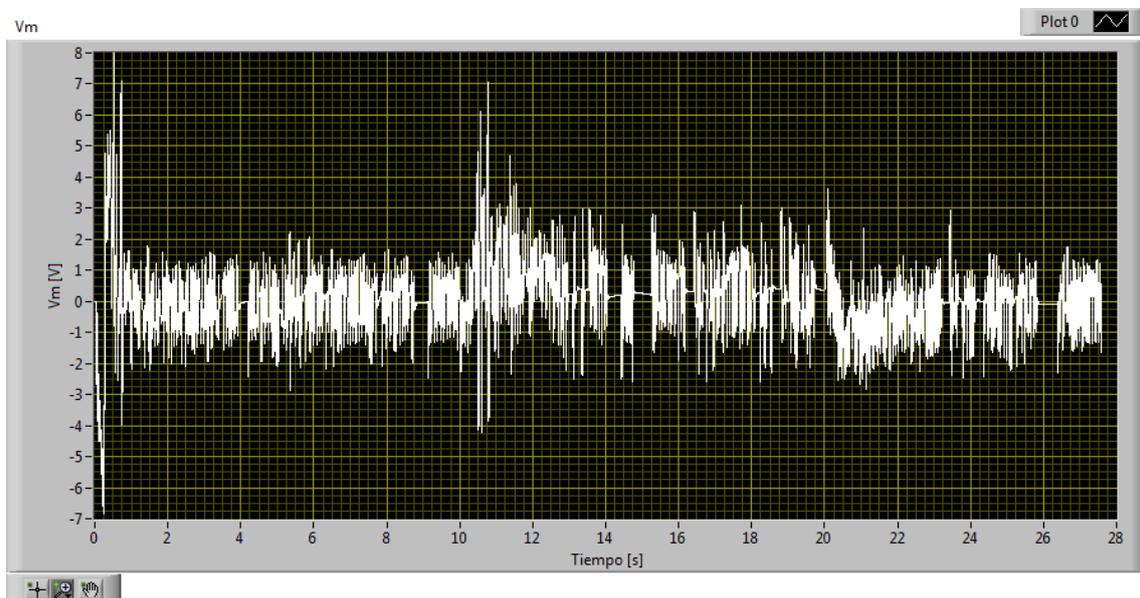


Figura 112. Tensión aplicada al péndulo invertido en la prueba

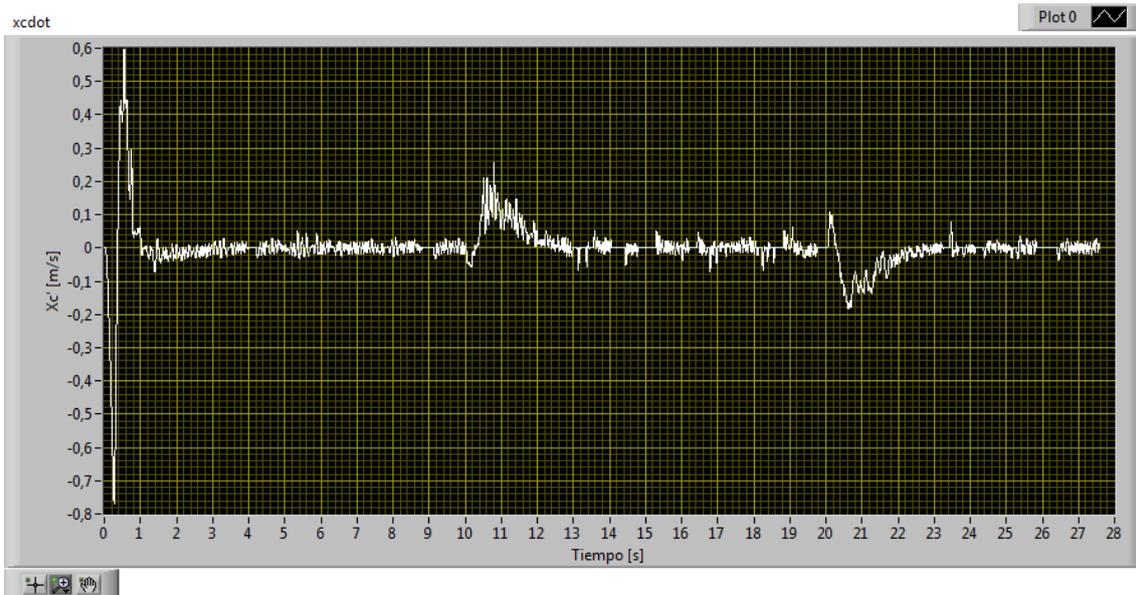


Figura 113. Velocidad del carro en la prueba

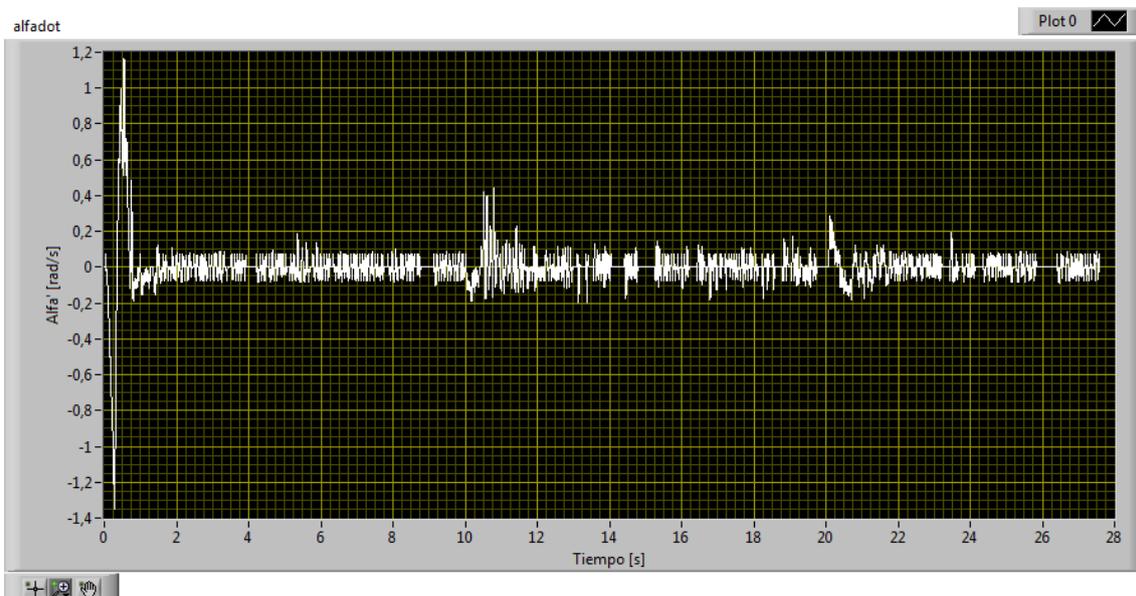


Figura 114. Velocidad angular del péndulo invertido en la prueba

Sin embargo, si se decide permitir mayor sobreoscilamiento, existen otras ganancias válidas como, por ejemplo, las que se consiguen para  $SO = 2\%$  y  $T_s = 3s$ , que corresponden a los polos conjugados  $p_{1,2} = -1,333 \pm 1,071 j$ . Los otros tres polos se mantienen en su ubicación.

$$p_3 = -7 \quad (101)$$

$$p_4 = -11 \quad (102)$$

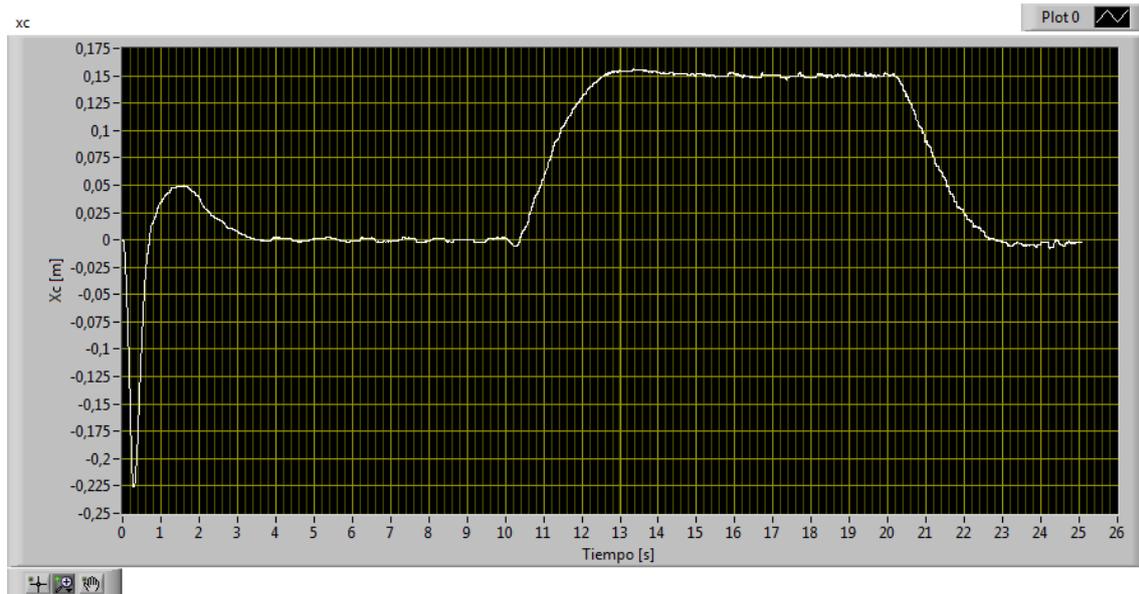
$$p_5 = -15 \quad (103)$$

Resolviendo la fórmula de Ackermann, se halla el valor de la matriz de ganancias del sistema aumentado siguiente.

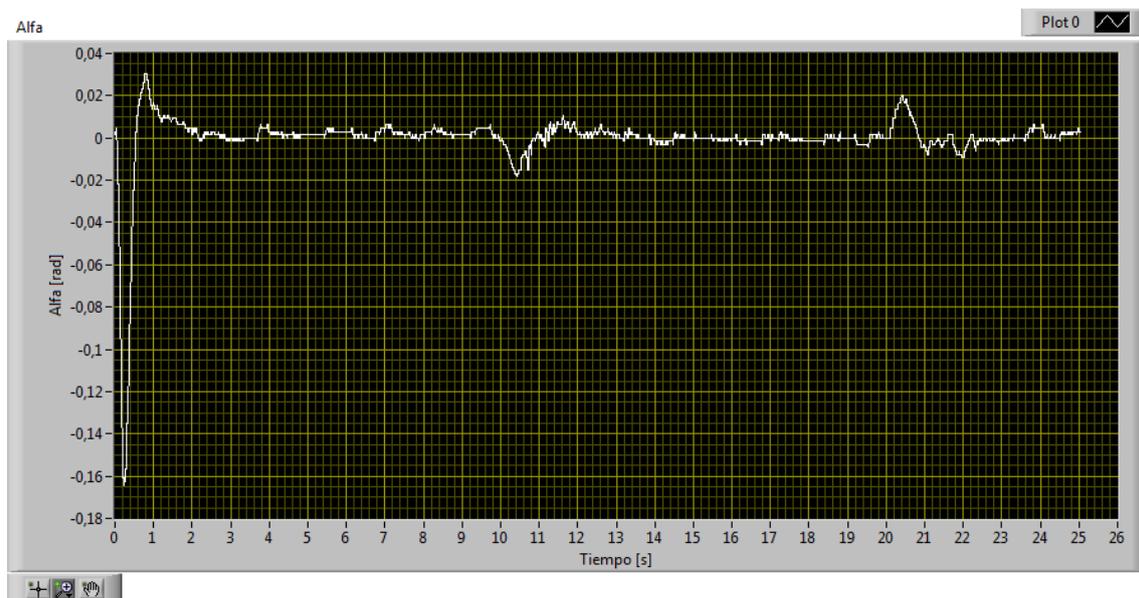
$$\hat{K} = [-75,10 \quad 116,98 \quad -51,06 \quad 25,20 \quad 61,95] \quad (104)$$

donde  $K = [-75,10 \quad 116,98 \quad -51,06 \quad 25,20]$  y  $K_i = -61,95$ .

A continuación, se muestra la respuesta obtenida en el sistema real ante una referencia de onda cuadrada de 0,15 metros de amplitud. En las figuras se incluyen la acción de control  $V_m$  y el estado, compuesto por  $x_c$ ,  $\alpha$  y sus respectivas derivadas.



*Figura 115. Desplazamiento del carro en la prueba*



*Figura 116. Ángulo del péndulo invertido en la prueba*

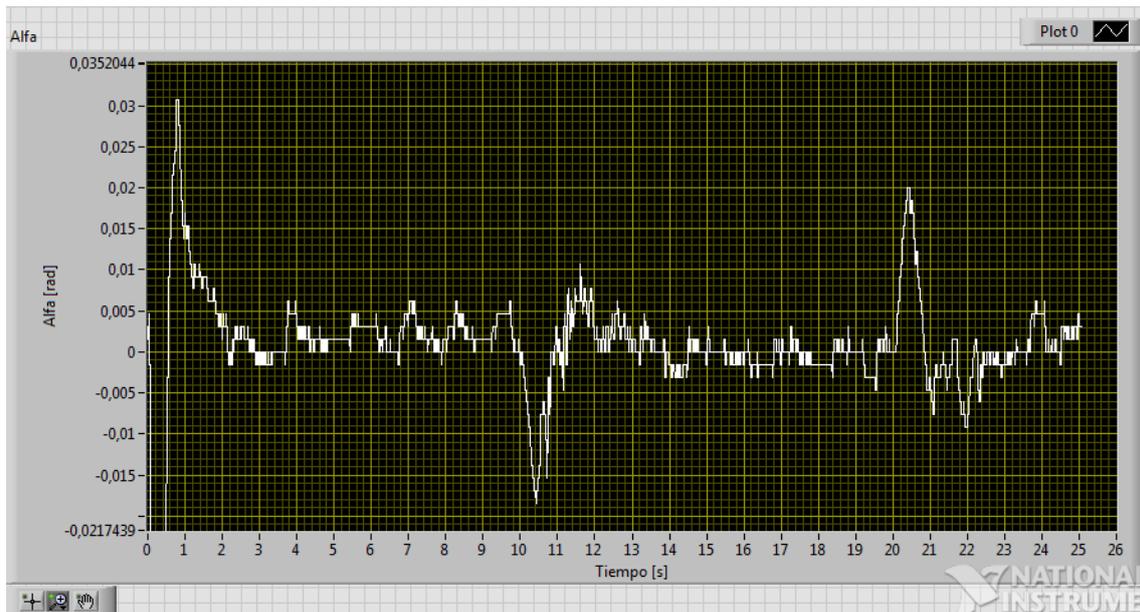


Figura 117. Detalle del ángulo del péndulo invertido en la prueba

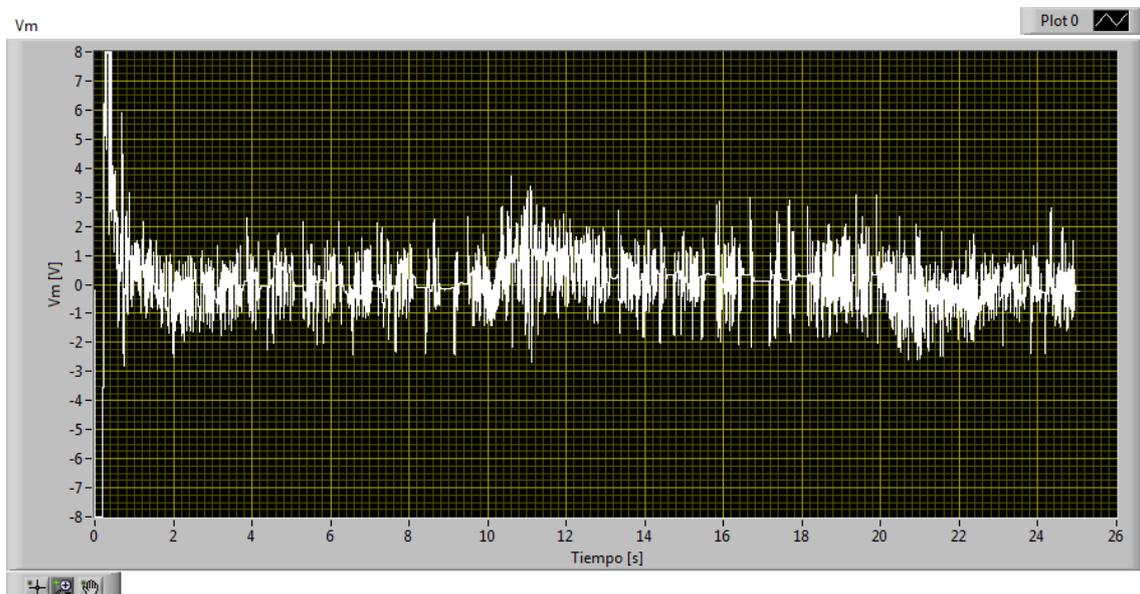


Figura 118. Tensión aplicada al péndulo invertido en la prueba

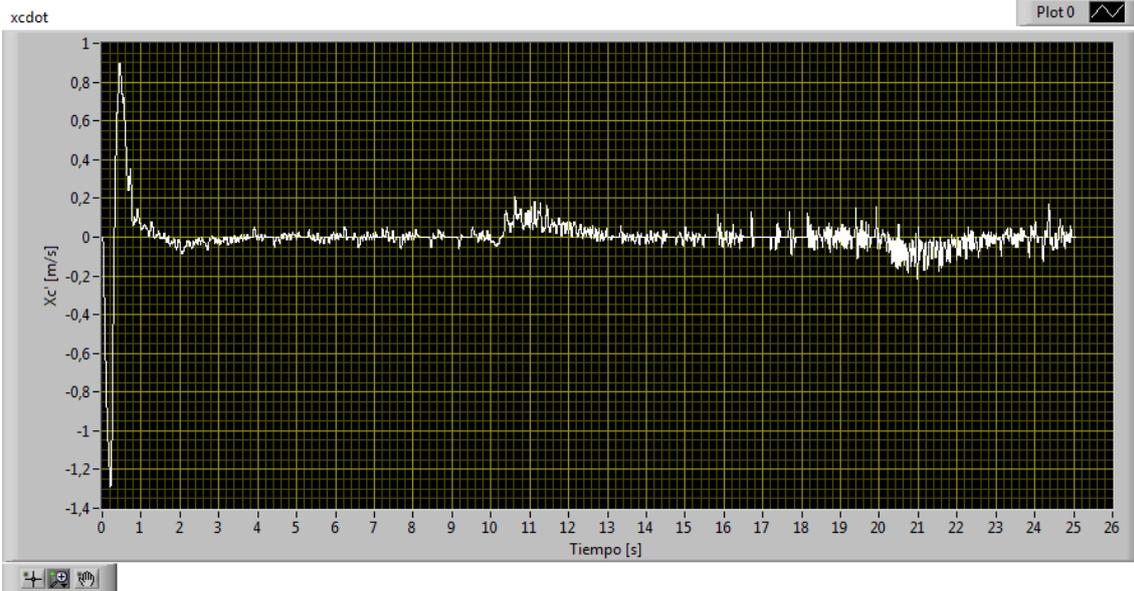


Figura 119. Velocidad del carro en la prueba

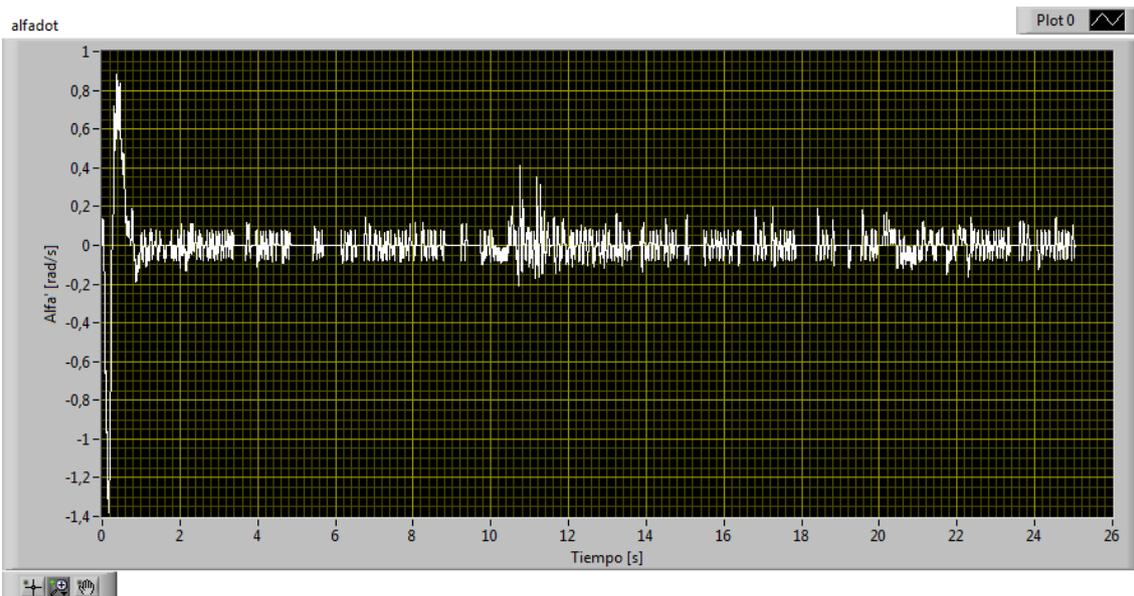


Figura 120. Velocidad angular del péndulo invertido en la prueba

### 6.5.3.3. Controlador de realimentación del estado LQR

Se ha intentado de nuevo la doble integración, en  $x_c$  y  $\alpha$ , con el mismo resultado, por lo que se descarta la de  $\alpha$ , y el esquema de control es el mismo que en el caso de asignación de polos.

#### Péndulo Corto

En este sistema, sobre todo, se penaliza el error de  $x_c$  y  $\alpha$ , ya que ambas son salidas del sistema. La primera es la variable que debe seguir a la referencia, y la segunda es vital para la estabilidad del sistema. Además, también se penaliza la integral del error de la primera, para que éste sea lo menor posible. Por el contrario, se les asigna poco peso a las derivadas, que no son vitales para el objetivo de control.

$$Q = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 60 \end{bmatrix} \quad (105)$$

Para  $V_m$ , se escoge un peso pequeño porque no interesa limitarla, igual que en ocasiones anteriores.

$$R = 0,05 \quad (106)$$

Con los valores anteriores de  $Q$  y  $R$ , se utiliza la función LQR en LabVIEW, que devuelve como resultado la siguiente matriz de ganancias de realimentación del estado del sistema aumentado.

$$\hat{K} = [-50,86 \quad 85,39 \quad -36,24 \quad 13,36 \quad 34,64] \quad (107)$$

Por lo tanto, los valores de las ganancias del sistema son las siguientes.

$$K = [-50,86 \quad 85,39 \quad -36,24 \quad 13,36] \quad (108)$$

$$K_i = -34,64 \quad (109)$$

A continuación, se presentan las figuras con los resultados obtenidos sobre el sistema real ante referencia nula para  $x_c$  y  $\alpha$ . Se pueden observar las salidas del sistema  $x_c$  y  $\alpha$ , sus respectivas derivadas y la acción de control  $V_m$ .

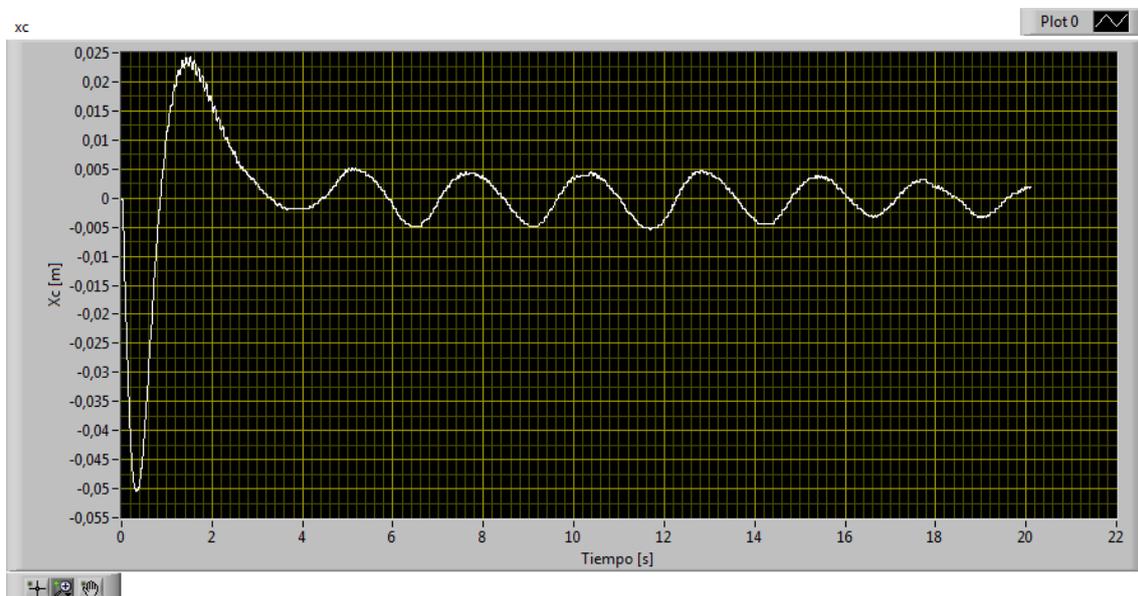


Figura 121. Desplazamiento del carro en la prueba

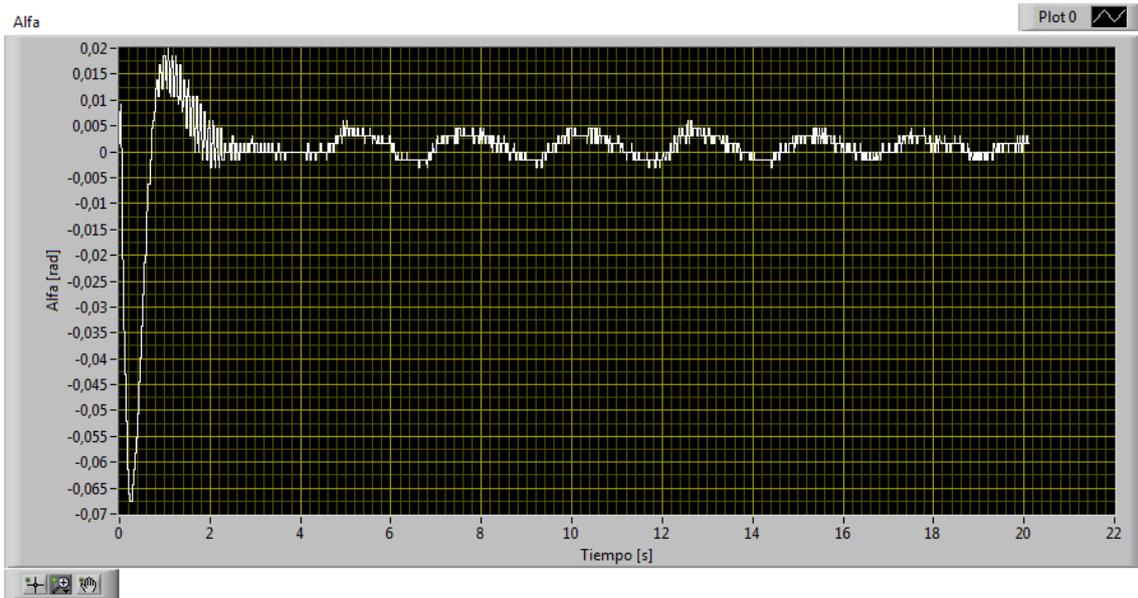


Figura 122. Ángulo del péndulo invertido en la prueba.

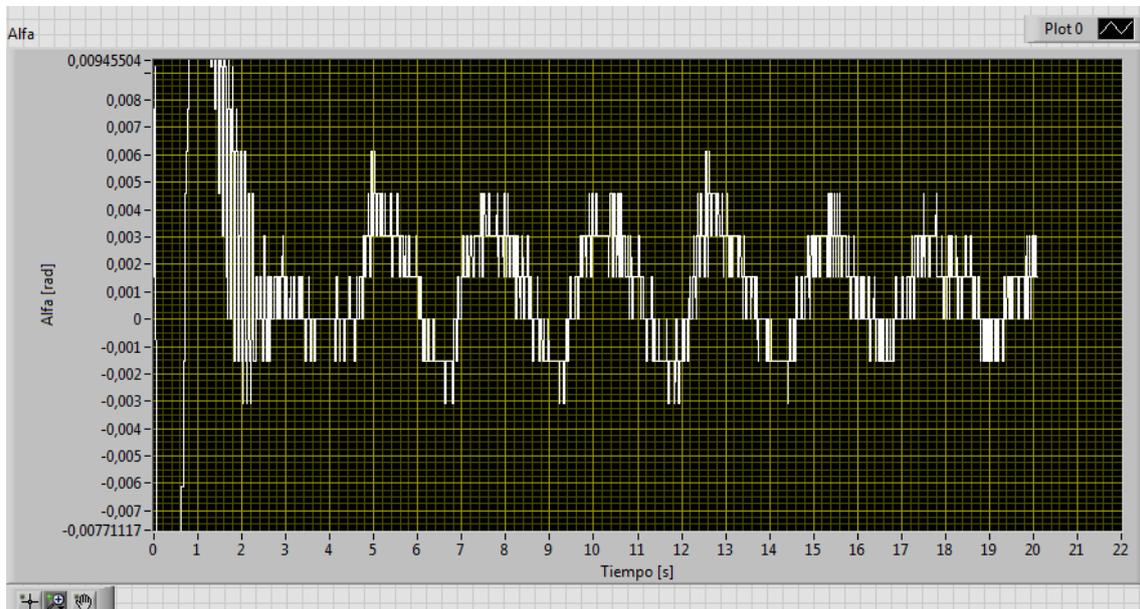


Figura 123. Detalle del ángulo del péndulo invertido en la prueba

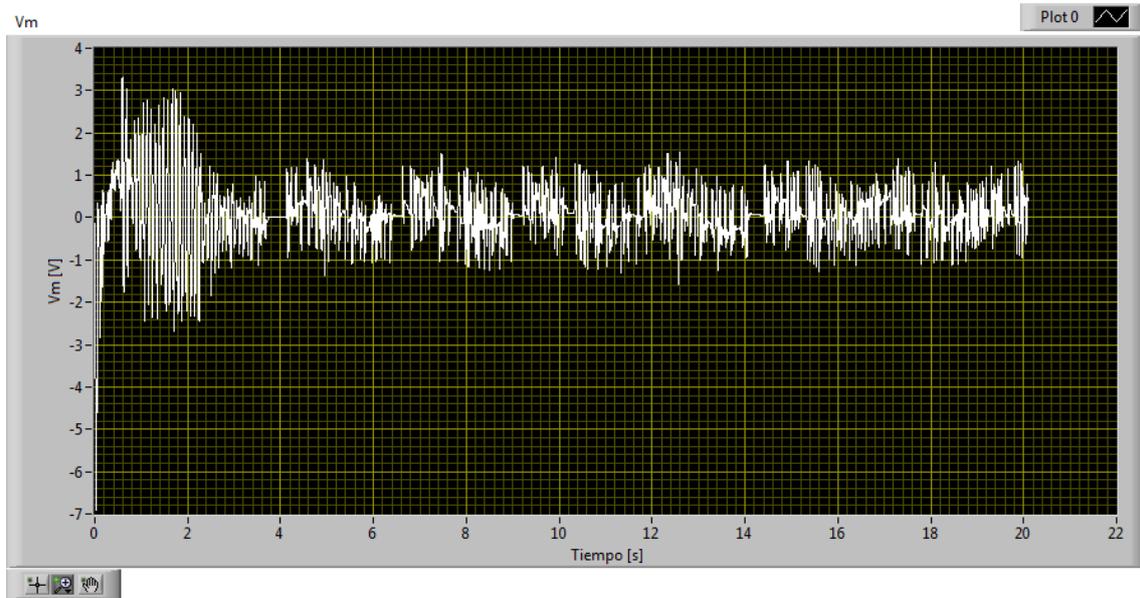


Figura 124. Tensión aplicada al péndulo invertido en la prueba

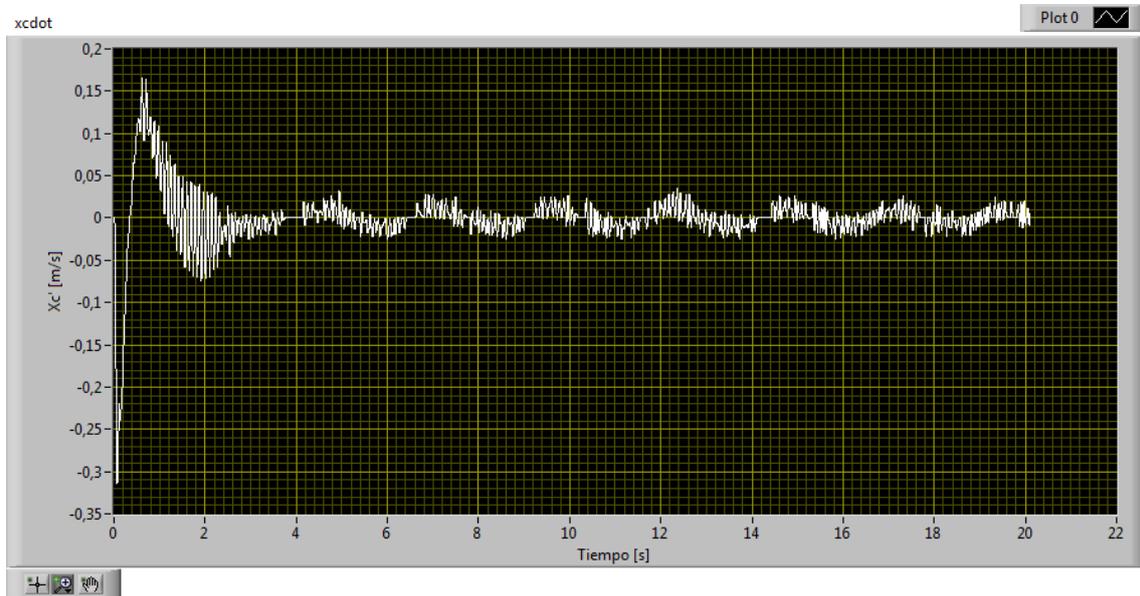


Figura 125. Velocidad del carro en la prueba

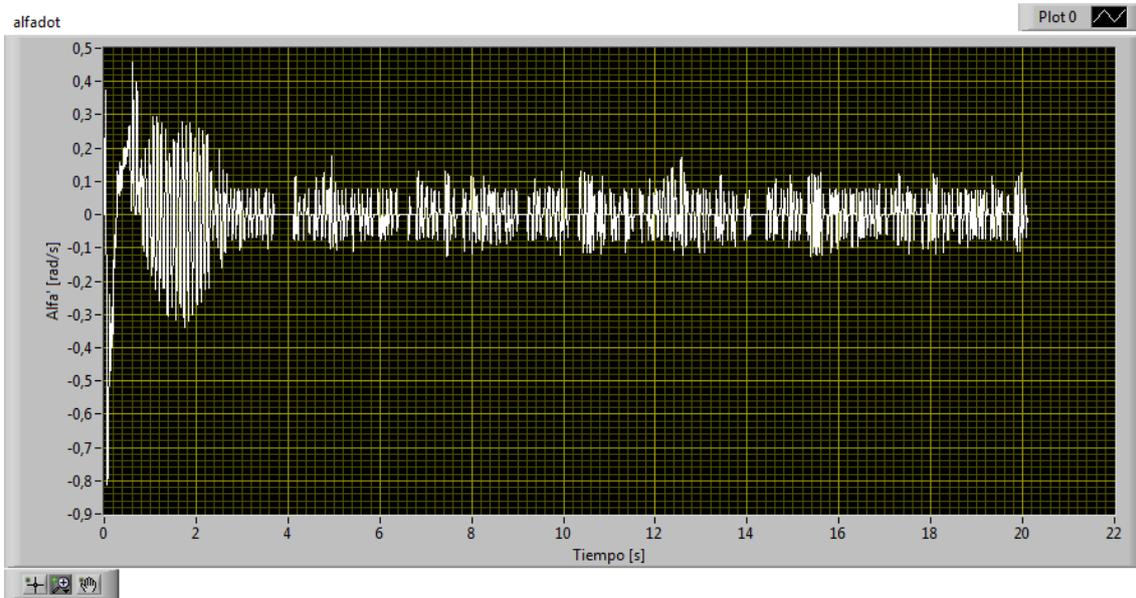


Figura 126. Velocidad angular del péndulo invertido en la prueba

### Péndulo Largo

Dado que el criterio es el mismo que en el caso del péndulo corto, y los resultados son buenos, se seleccionan las mismas matrices en este caso.

$$Q = \begin{bmatrix} 30 & 0 & 0 & 0 & 0 \\ 0 & 30 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 0 & 60 \end{bmatrix} \quad (110)$$

$$R = 0,05 \quad (111)$$

Con  $Q$  y  $R$ , se calcula la siguiente matriz de ganancias de realimentación del estado del sistema aumentado, usando la función LQR de LabVIEW.

$$\hat{K} = [-54,02 \quad 114,15 \quad -41,81 \quad 24,20 \quad 34,64] \quad (112)$$

Por lo tanto, los valores de las ganancias del sistema son las siguientes.

$$K = [-54,02 \quad 114,15 \quad -41,81 \quad 24,20] \quad (113)$$

$$K_i = -34,64 \quad (114)$$

A continuación, se presentan los resultados obtenidos sobre el sistema real, de nuevo ante referencia nula. Se pueden observar las salidas del sistema  $x_c$  y  $\alpha$ , sus respectivas derivadas y la acción de control  $V_m$ .

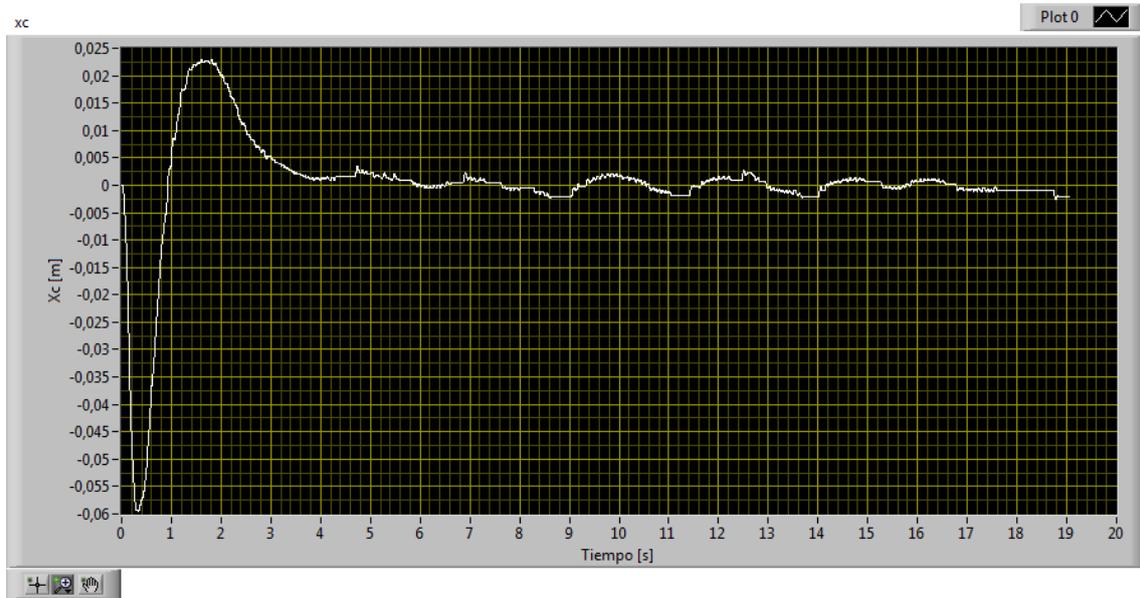


Figura 127. Desplazamiento del carro en la prueba

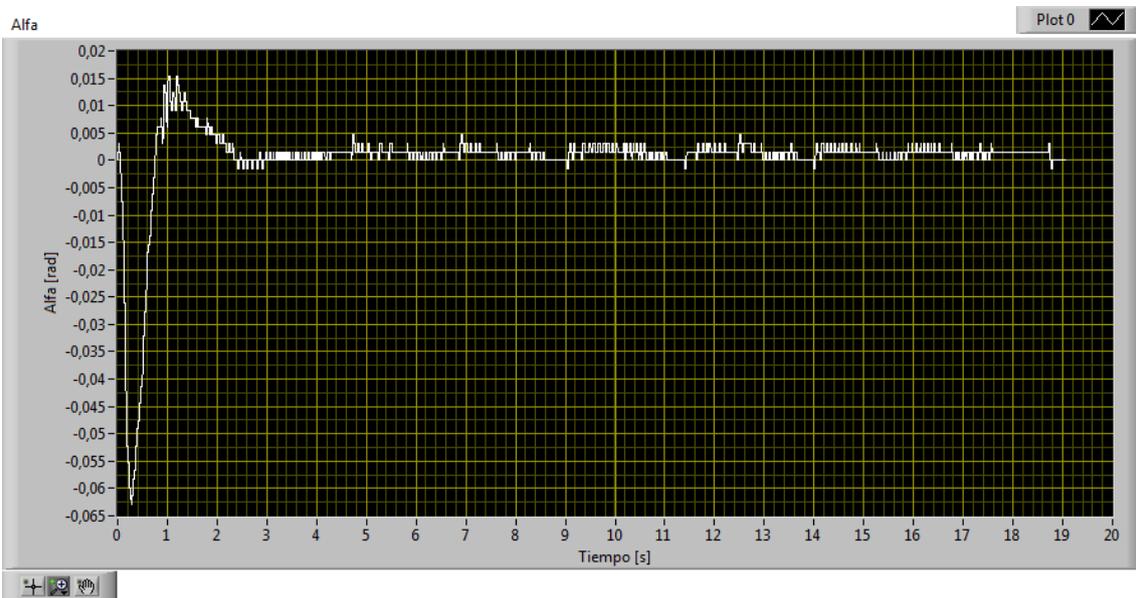


Figura 128. Ángulo del péndulo invertido en la prueba

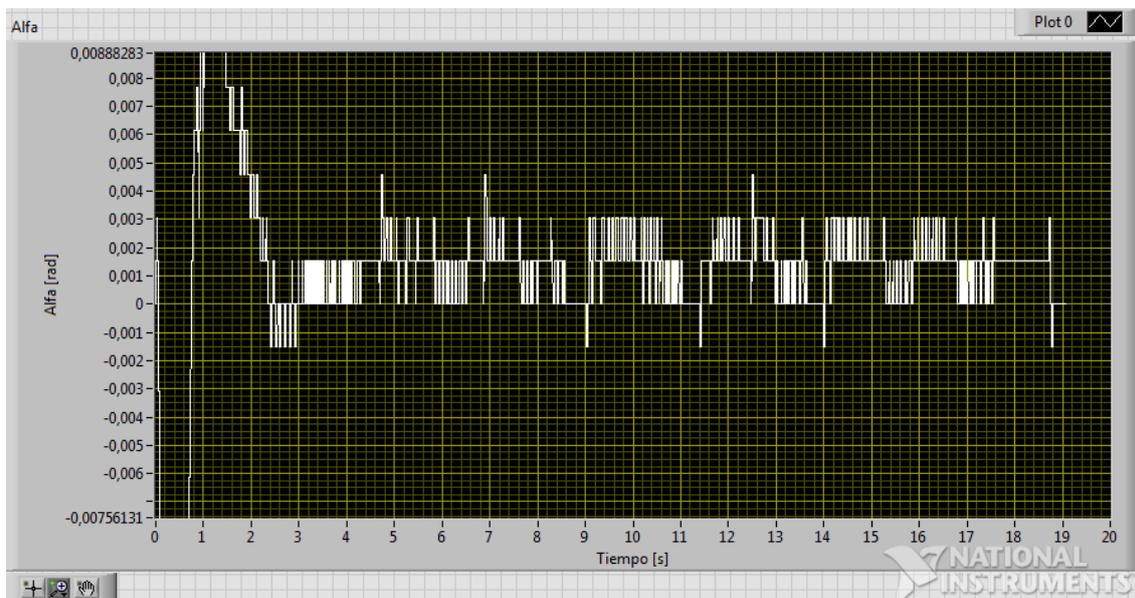


Figura 129. Detalle del ángulo del péndulo invertido en la prueba

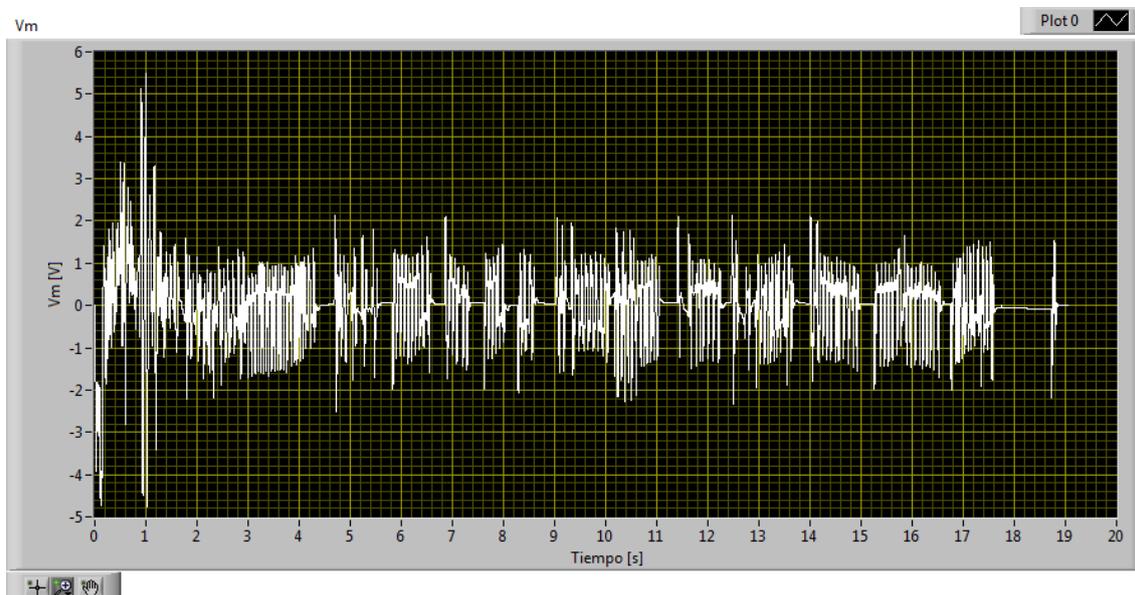


Figura 130. Tensión aplicada al péndulo invertido en la prueba

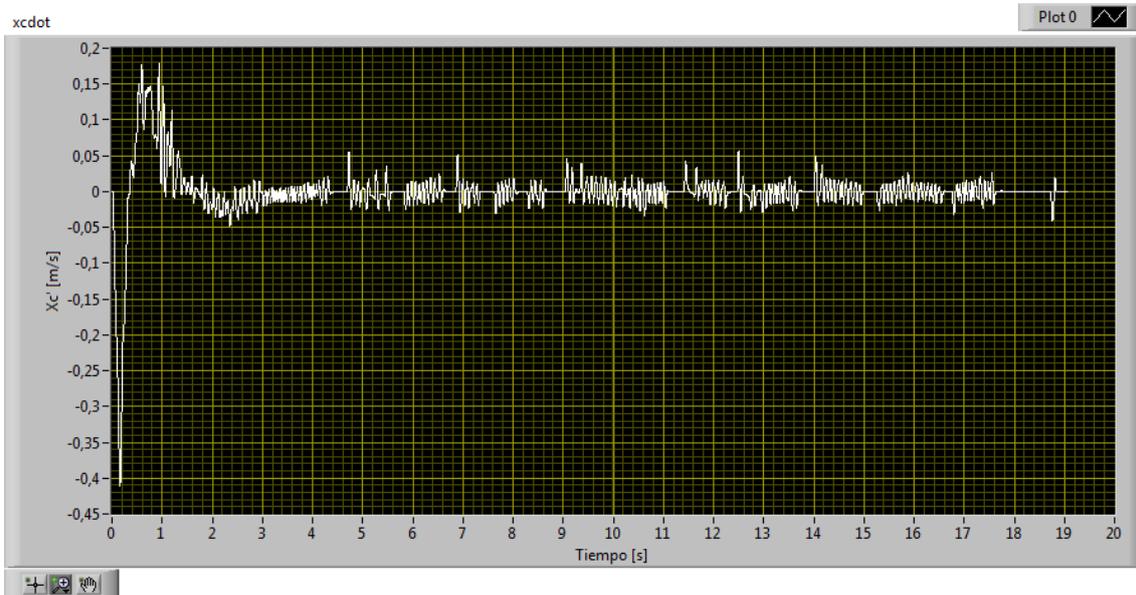


Figura 131. Velocidad del carro en la prueba

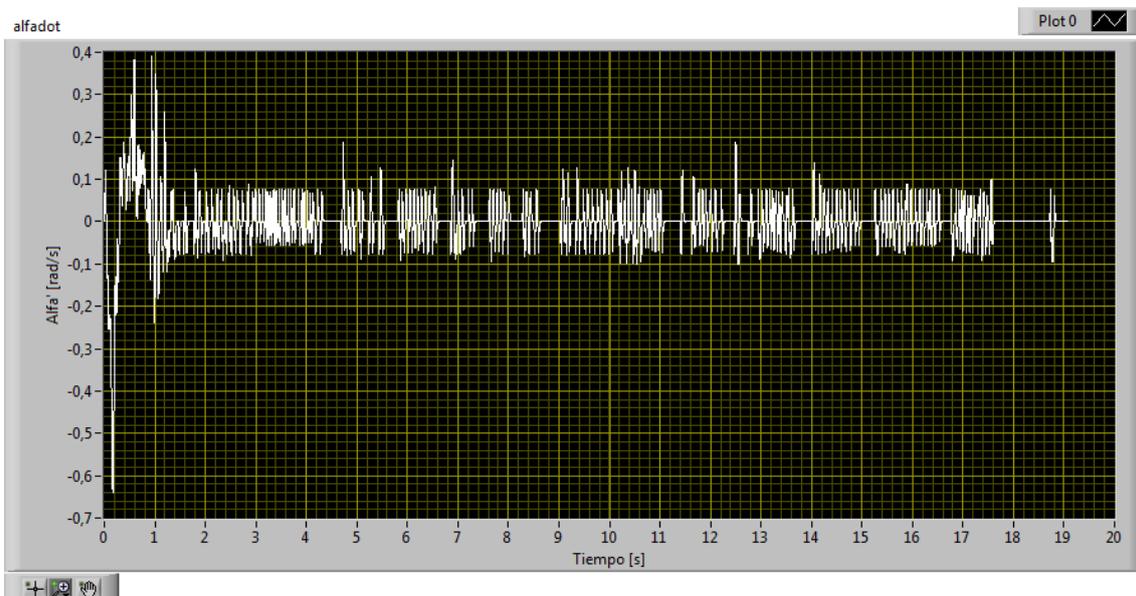


Figura 132. Velocidad angular del péndulo invertido en la prueba

#### **6.5.3.4. Autoerección**

Como ya se ha mencionado antes, la autoerección es el movimiento por el cual el péndulo pasa de manera autónoma de su posición de reposo, en la posición vertical inferior, a la posición de péndulo invertido. Por lo tanto, para la puesta en marcha y control del péndulo se distinguen tres fases: reposo, autoerección y control de balanceo. Esta última se refiere a los controles explicados en los apartados anteriores, cuyo objetivo es mantener al péndulo en posición invertida.

### **Convenio de ángulos y signos en modos Autoerección y Balanceo**

Por razones prácticas, interesa manejar diferentes referencias para la posición angular del péndulo.

En el modo de autoerección, se adopta el ángulo  $\theta(t)$  como variable, que mide el ángulo del péndulo respecto a su posición inferior de reposo.

Una vez que entra en funcionamiento el modo de control de balanceo, el ángulo del péndulo que se maneja es el que toma como referencia la posición vertical superior, de acuerdo a la figura siguiente:

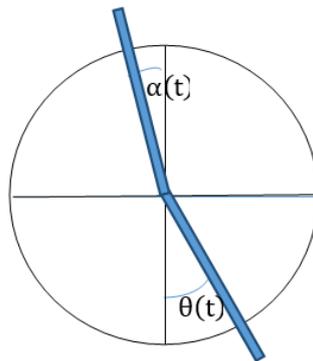


Figura 133. Convenio de ángulos

Se requiere, por tanto, una función que convierta el ángulo  $\theta(t)$  leído por el encoder, en el ángulo alfa, necesario para el posterior control de balanceo.

La conversión se realiza mediante bloques aritméticos y booleanos que implementan las siguientes funciones:

Si  $(\pi - \theta(t)) \leq \pi$ , entonces,

$$\alpha(t) = -(\pi - \theta(t)) \quad (115)$$

En caso contrario

$$\alpha(t) = (\pi + \theta(t)) \quad (116)$$

### **Control de autoerección**

El objetivo es llevar el péndulo desde su posición de reposo (péndulo en posición inferior) hasta su posición vertical superior, en unas condiciones de posición ( $\alpha(t)$ ) y velocidad angular ( $\dot{\alpha}(t)$ ) que permitan al modo balanceo tomar el control. Una velocidad angular excesiva puede imposibilitar que los controladores de balanceo (LQR, PID, ...) cumplan el objetivo de mantener estable el péndulo en la posición vertical en el momento de realizar el cambio del modo de autoerección al modo de control de balanceo.

Se han tenido en cuenta distintas estrategias de control, como se indica a continuación.

- Excitación a frecuencia de resonancia del sistema

Se trata de excitar al sistema (carro y péndulo) mediante una señal senoidal de tensión aplicada al motor de las siguientes características:

$$V_m = A \sin(\omega t) \quad (117)$$

donde  $\omega = 2\pi f_r$ , siendo  $f_r$  la frecuencia de resonancia del conjunto carro-péndulo y  $A$  es la amplitud de la excitación requerida para comunicar al sistema la energía necesaria que lleve el péndulo a su posición vertical.

La frecuencia de resonancia se ha obtenido experimentalmente desplazando el péndulo hasta una posición inicial y soltándolo para permitir que oscile libremente. El resultado obtenido para el péndulo corto es de  $f_r = 0.83 \text{ Hz}$ .

Ésta es la frecuencia de oscilación del péndulo y también la horizontal del carro, ya que durante la oscilación del péndulo también se produce una oscilación en el carro en un su movimiento horizontal.

Excitando el sistema con una señal senoidal de esa frecuencia el sistema es fuertemente inestable, generándose oscilaciones. Para que éstas crezcan en amplitud hasta alcanzar el péndulo la parte superior, se ha comprobado que se requiere una amplitud mínima  $A = 4 \text{ V}$  para el péndulo corto.

En la figura siguiente se muestra el resultado obtenido. Como puede verse, a los 3,5 segundos desde el inicio de las oscilaciones, el péndulo alcanza la posición vertical superior, rebasándola incluso.

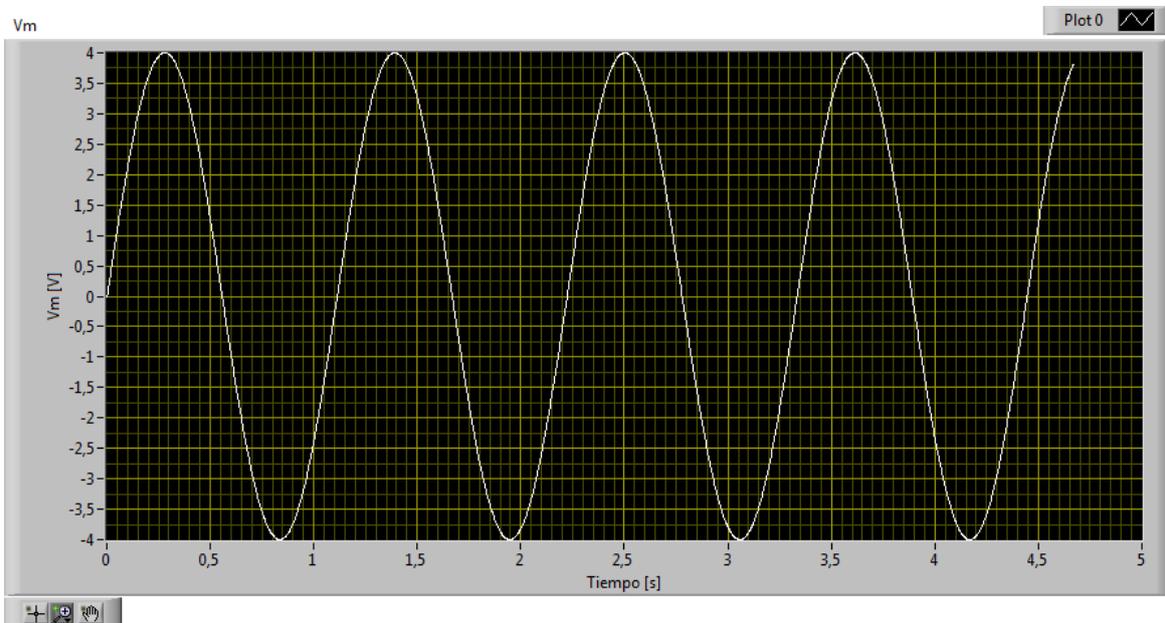


Figura 134. Tensión aplicada al carro para la autoerección

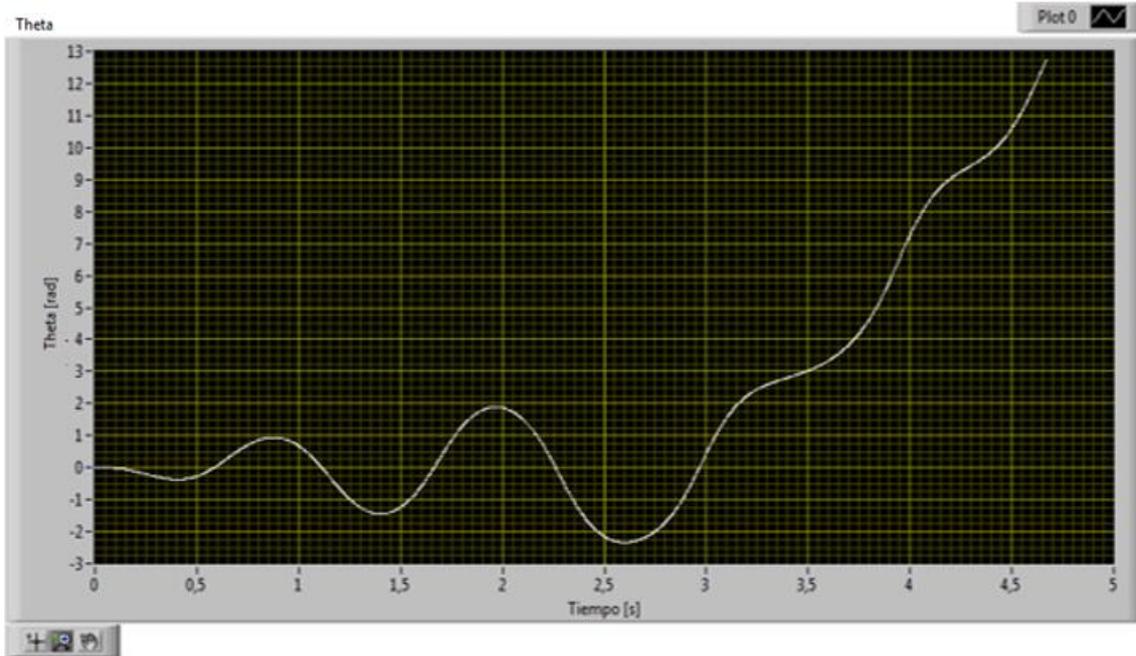


Figura 135. Ángulo Theta del péndulo en la autoerección

Para el péndulo largo, los valores son  $f_r = 0.57 \text{ Hz}$  y  $A = 7 \text{ V}$ .

El modo de control anteriormente descrito tiene, no obstante, los siguientes inconvenientes:

- La amplitud de excitación es la misma a lo largo de todo el proceso de autoerección, con independencia de que el péndulo esté iniciando el movimiento, en el que requiere más energía, o haya adquirido ya grandes oscilaciones, momento en que requiere poca energía adicional.
- El ajuste de las oscilaciones de forma que el péndulo llegue a la parte superior con poca velocidad debe hacerse mediante el ajuste de la amplitud  $A$ , pero pequeños cambios en el sistema, como fricciones, etc... pueden hacer que el resultado no sea repetitivo, haciendo que, para la misma amplitud  $A$  en algunos casos el péndulo no llegue a la parte superior o por el contrario lo haga a velocidad excesiva, dificultando así el cambio de modo a control de balanceo.

Por dichas razones se prefiere utilizar un control basado en energía como propone Quanser, y que se explica a continuación.

• Control basado en la energía total del péndulo

Se basa en un control proporcional al error existente entre la energía total del péndulo  $E$  (cinética más potencial) y la referencia de energía total deseada  $E_r$ .

La ecuación de control es la que se indica a continuación:

$$u = \text{satumax}[\mu \text{sign}(\dot{\theta}(t) * \cos(\theta(t)))(E - E_r)] \quad (118)$$

donde  $u$  es la aceleración lineal a comunicar al carro, limitada a unos valores  $\pm u_{max}$ , determinados mediante una función de saturación;  $\theta(t)$  es el ángulo del péndulo respecto a su posición de reposo inferior;  $\mu$  es el módulo de la ganancia

proporcional, que permite el ajuste del controlador; y  $E$  es la energía total del péndulo en cada instante (variable a controlar), cuyo valor es:

$$E = E_k + E_p = \frac{1}{2} J_{po} \dot{\theta}(t)^2 + M_p g l_p (1 - \cos(\theta(t))) \quad (119)$$

$$J_{po} = \frac{1}{12} M_p (2 l_p)^2 + M_p l_p^2 = \frac{4}{3} M_p l_p^2 \quad (120)$$

$J_{po}$  es el momento de inercia del péndulo respecto a la articulación del péndulo en el carro (extremo de péndulo), a diferencia de  $J_p$  manejado anteriormente en los modelos matemáticos del sistema, que está referido a su centro de masa.

Como puede verse, el control proporcional no es lineal porque la ganancia,  $\mu \text{sign}(\dot{\theta}(t) * \cos(\theta(t)))$ , cambia de signo en función de  $\dot{\theta}(t)$  y  $\cos(\theta(t))$ .

Por otro lado, puesto que lo que se pretende es que el péndulo llegue a la parte superior con velocidad nula o muy pequeña, el valor de  $E_r$  debe ser el correspondiente solamente a la energía potencial en la parte superior del péndulo, cuyo valor es:

$$E_r = E_p = M_p g (2 l_p) \quad (121)$$

Para el péndulo corto el valor es de  $E_r = 0.4 J$  y para el péndulo largo de  $E_r = 1.5 J$ .

Para establecer la ley de control (118) en función de la tensión aplicada al motor  $V_m$ , es preciso relacionar  $V_m$  y  $u$ .

De acuerdo a la ecuación (122)

$$\frac{\eta_g \eta_m K_t K_g V_m(t)}{r_{mp} R_m} - \left[ \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} + B_{eq} \right] \dot{x}_c(t) - \left[ \frac{\eta_g K_g^2 J_m}{r_{mp}^2} + M_c \right] \ddot{x}_c(t) = 0 \quad (122)$$

Haciendo  $\ddot{x}_c(t) = u$ , y llamando

$$M_{eq} = \left[ M_c + \frac{\eta_g K_g^2 J_m}{r_{mp}^2} \right] \quad (123)$$

$$V_m = \frac{u M_{eq} r_{mp} R_m}{\eta_g \eta_m K_t K_g} + \left[ \frac{K_g K_m}{r_{mp}} + \frac{r_{mp} R_m B_{eq}}{\eta_g \eta_m K_t K_g} \right] \dot{x}_c(t) \quad (124)$$

Las ecuaciones de la ley de control son, por tanto:

$$u = \text{satmax}[\mu \text{sign}(\dot{\theta}(t) * \cos(\theta(t)))(E - E_r)] \quad (118)$$

$$V_m = \frac{u M_{eq} r_{mp} R_m}{\eta_g \eta_m K_t K_g} + \left[ \frac{K_g K_m}{r_{mp}} + \frac{r_{mp} R_m B_{eq}}{\eta_g \eta_m K_t K_g} \right] \dot{x}_c(t) \quad (124)$$

Como puede verse, con el péndulo en reposo, como la velocidad del péndulo es cero, la función signo es cero y, por lo tanto,  $u$  también. De esta manera, nunca se iniciaría el proceso de autoerección. Por lo tanto, en la programación se incluye un

pequeño impulso de -5V para sacar al péndulo del reposo. Además, en el caso de que el péndulo no esté completamente en reposo, se asegura hacia qué lado comienza la autoerección.

Como el controlador así definido es de característica proporcional (P), se produce un error permanente estacionario, de forma que el péndulo no alcanza el valor de consigna deseado  $E_r$ . Además, la aproximación de  $E$  a  $E_r$  es de forma asintótica, por lo que la fase de autoerección se prolonga excesivamente en el tiempo. Para corregir estos fenómenos, se establecen valores de  $E_r$  superiores a los teóricamente requeridos de 0,4 y 1,5 Julios, eligiendo 0,8 y 1,9 Julios respectivamente.

Por otro lado, se observa que la aportación a la acción de control del término  $\left[ \frac{r_{mp} R_m B_{eq}}{\eta_g \eta_m K_t K_g} \right] \dot{x}_c(t)$  es muy pequeño, por lo que dicho término se suprime por sencillez.

Éste es, finalmente, el algoritmo elegido para la autoerección en la aplicación debido a su mejor comportamiento, ya que tiene en cuenta la velocidad con la que el péndulo llega a su posición superior.

### **Sintonización del algoritmo**

A pesar de que este algoritmo utiliza parámetros del sistema para calcular la energía del mismo, debido a las diferencias entre modelo y sistema real es necesario ajustar algunos parámetros del algoritmo de autoerección basado en la energía total del péndulo.

### **Péndulo Corto**

Los valores de los parámetros del sistema ajustados para el péndulo corto son los que se muestran en la *Tabla 6* siguiente.

Parámetro	Valor	Unidades
$M_p$	0.129	[Kg]
$l_p$	0.155	[m]
$J_{po}$	0.004	[Kgm <sup>2</sup> ]
$E_r$	0.8	[J]
$B_{eq}$	0	[Ns/m]
$K_m$	0.0103	[V/rad/s]
$u_{max}$	20	[m/s <sup>2</sup> ]
$\mu$	10	[kg <sup>-1</sup> m <sup>-1</sup> ]

*Tabla 6. Valores del algoritmo de autoerección para el péndulo corto*

En las siguientes figuras se puede apreciar la evolución del ángulo  $\theta$  y la acción de control  $V_m$  durante la fase de autoerección y de control de balanceo, en torno a la posición cero. Además, se muestra la energía  $E$  del péndulo, únicamente durante la fase de autoerección.

Se observa que la conmutación de un modo a otro se produce poco antes de los 19 segundos transcurridos en la prueba, cuando el péndulo alcanza los 4 julios mencionados antes, y cuando el ángulo es cercano a 180°, es decir,  $\pi$  radianes (en valor absoluto). En ese momento el valor de  $\theta$  permanece constante.

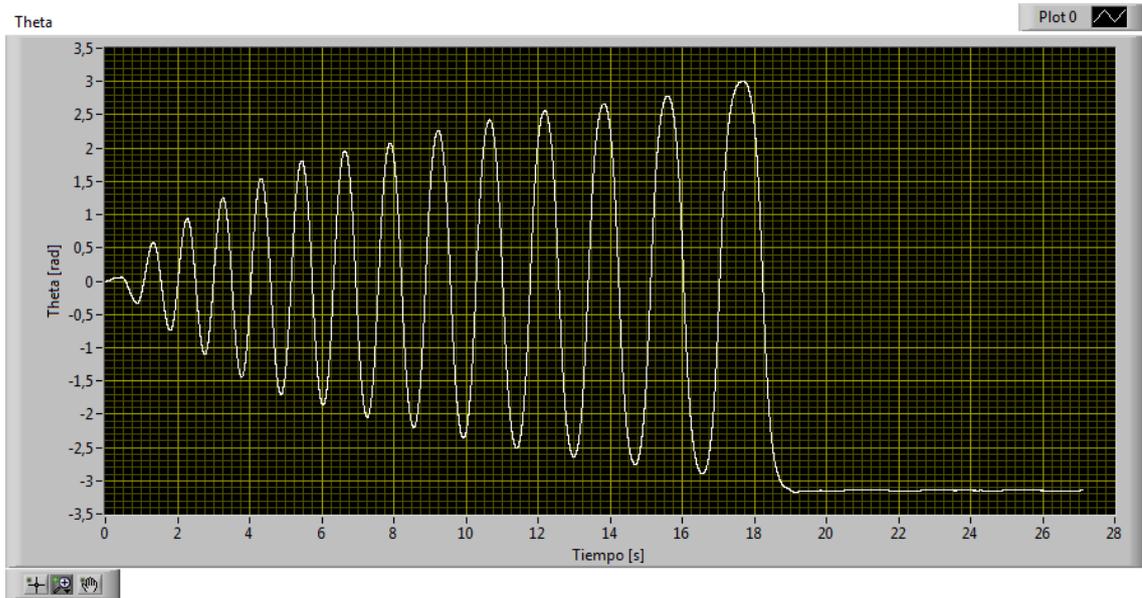


Figura 136. Ángulo Theta del péndulo en la autoerección

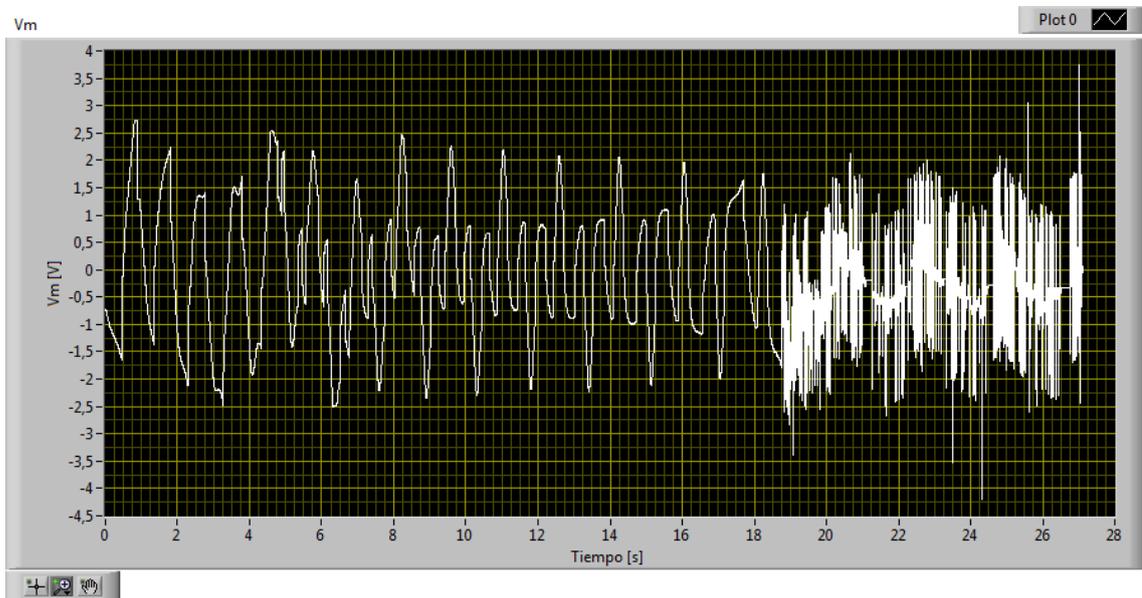


Figura 137. Tensión aplicada al carro en la autoerección

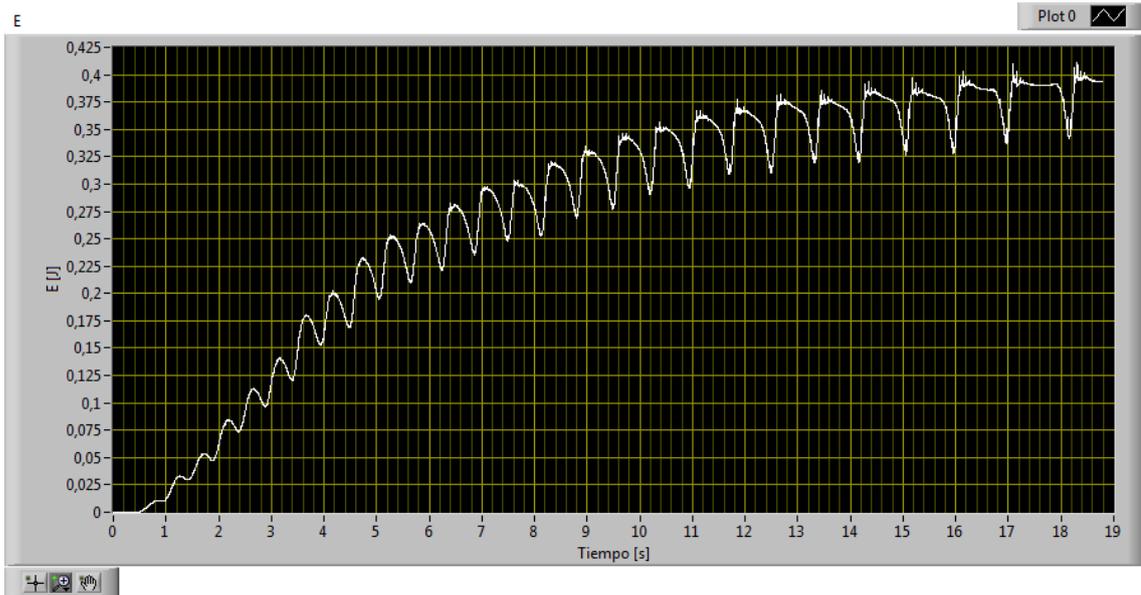


Figura 138. Energía del péndulo durante la autoerección

Para el péndulo corto, las pruebas se inician 150 milímetros a la izquierda de la posición central de la pista, para que el carro no se salga de la misma al realizar la conmutación, que genera desplazamientos notables en el carro antes de estabilizarse.

### Péndulo Largo

En el caso del péndulo largo, los valores de los parámetros para la autoerección son los indicados en la *Tabla 7* siguiente.

Parámetro	Valor	Unidades
$M_p$	0.23	[Kg]
$l_p$	0.33	[m]
$J_{po}$	0.0253	[Kgm <sup>2</sup> ]
$E_r$	1.9	[J]
$B_{eq}$	0	[Ns/m]
$K_m$	0.017	[V/rad/s]
$u_{max}$	32	[m/s <sup>2</sup> ]
$\mu$	18	[kg <sup>-1</sup> m <sup>-1</sup> ]

Tabla 7. Valores del algoritmo de autoerección para el péndulo largo

Se muestra la evolución del ángulo  $\theta$  y la acción de control  $V_m$  durante la fase de autoerección y de control de balanceo, en torno a la posición cero, en las figuras inferiores. Además, se representa la energía  $E$  del péndulo, únicamente durante la fase de autoerección.

Se observa que la conmutación de un modo a otro se produce en torno a los 8,5 segundos transcurridos en la prueba, cuando el péndulo alcanza los 1,5 julios antes mencionados, y cuando el ángulo es cercano a 180°, es decir,  $\pi$  radianes (en valor absoluto).

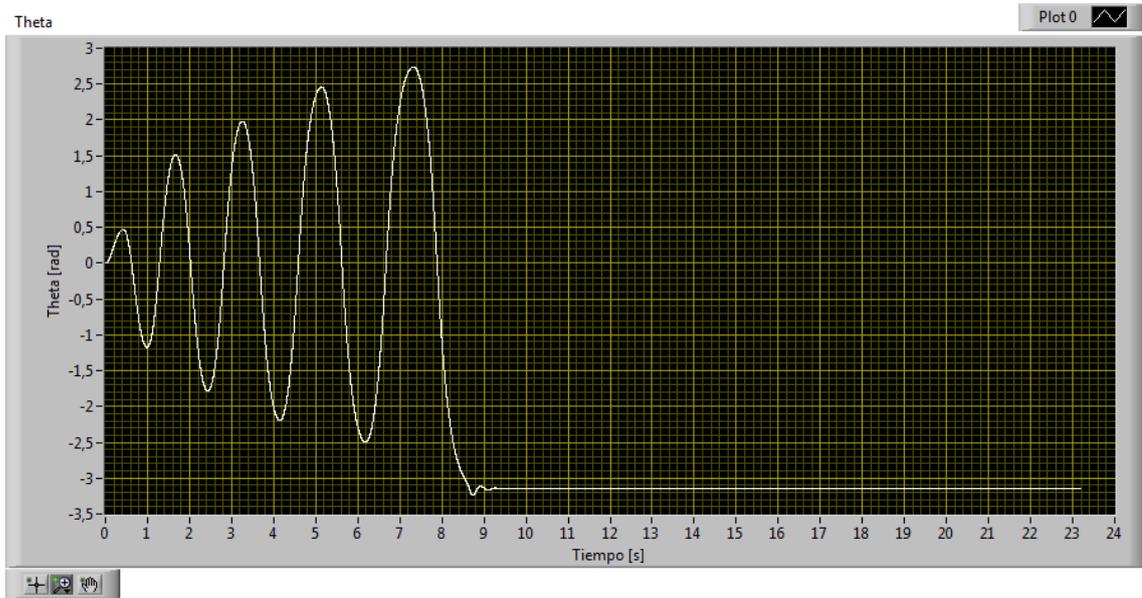


Figura 139. Ángulo Theta del péndulo en la autoerección

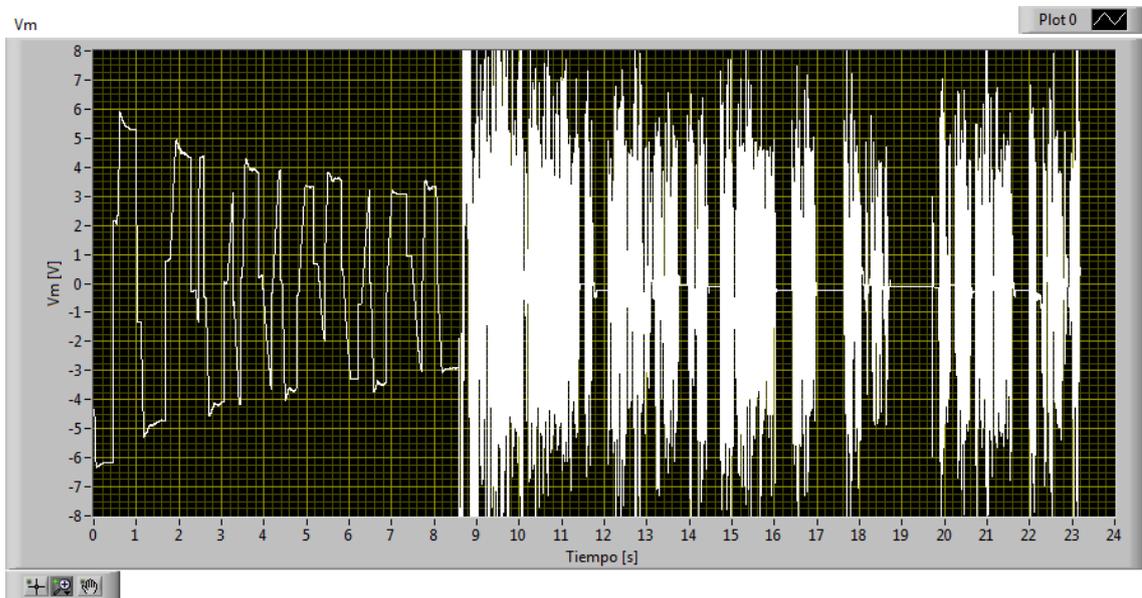


Figura 140. Tensión aplicada al carro en la autoerección

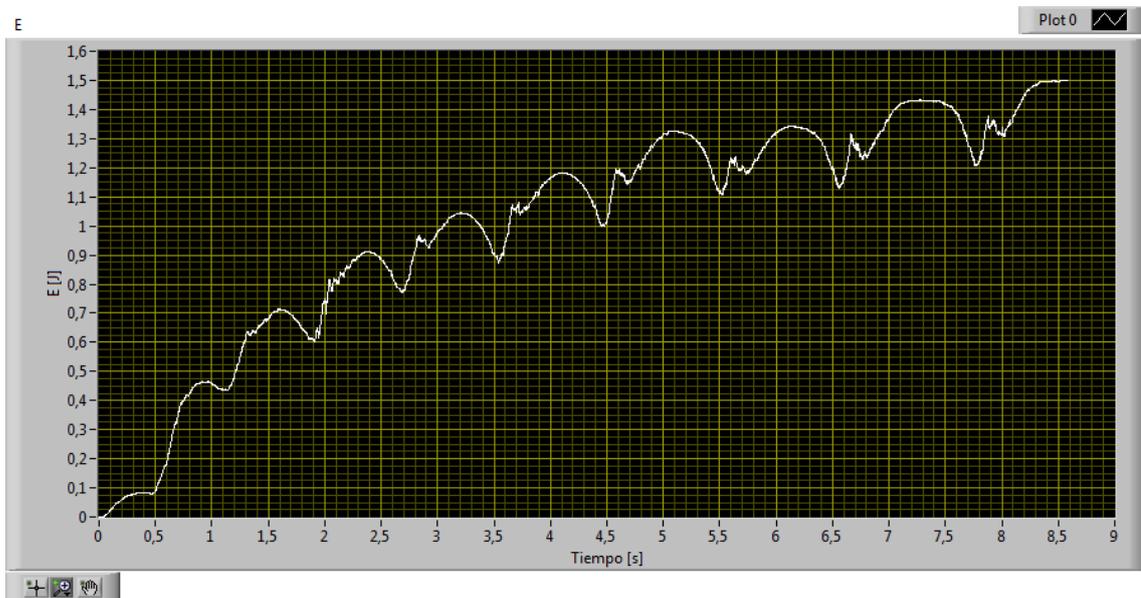


Figura 141. Energía del péndulo durante la autoerección

Cabe destacar que como el péndulo largo necesita una carrera mayor para iniciar la autoerección, las pruebas se inician 200 milímetros a la derecha de la posición central de la pista, para que el carro no se salga de la misma.

### **Ángulo de conmutación**

Se denomina intervalo de ángulo de conmutación al intervalo del ángulo  $\alpha$  en el que el sistema de control permite el cambio de modo, de autoerección a balanceo.

Debido a las características de cada control de balanceo y su comportamiento en la conmutación, se decide utilizar ángulos de conmutación distintos para controladores PID y controladores por realimentación del estado.

En los controladores PID el intervalo de ángulo  $\alpha$  de conmutación es  $\alpha = [-5^\circ, 5^\circ]$ , mientras que en los controladores por realimentación del estado es  $\alpha = [-10^\circ, 10^\circ]$ , es decir,  $5^\circ$  y  $10^\circ$  antes de alcanzar la posición superior del péndulo, respectivamente.

### **6.5.4. CONCLUSIONES**

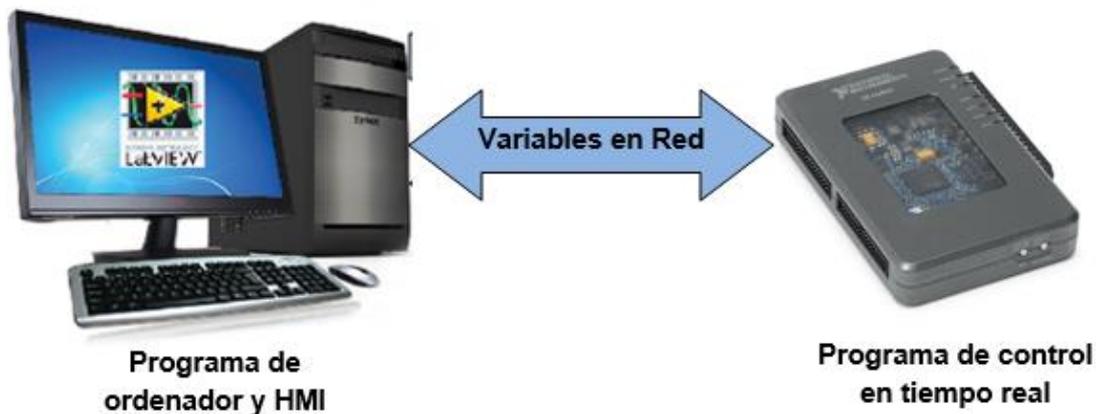
Como conclusión del diseño y pruebas realizadas con los diferentes sistemas y controladores, puede decirse:

- El sistema péndulo invertido, por su naturaleza inestable, es el más difícil de controlar y, contra lo que pudiese pensarse de forma intuitiva, el péndulo corto presenta más dificultades de control que el péndulo largo.
- Entre los controladores probados para péndulo Gantry y péndulo invertido, el algoritmo PID tiene un peor comportamiento que el resto debido a sus limitaciones en el manejo de variables, ya que no utiliza todas las variables de estado, como sí lo hacen los controladores de realimentación de estados.
- El algoritmo PID parece, así mismo, menos robusto, habiéndose comprobado que es muy sensible en el ajuste de la ganancia  $T_d$  (tiempo derivativo).
- En relación a los controladores de realimentación de estados, tanto LQR como por asignación de polos, el primero tiene un comportamiento ligeramente mejor.
- Durante las múltiples pruebas realizadas en el laboratorio, se ha observado que el piñón motriz sufre desgaste, debido a que está hecho de material plástico. A medida que el piñón se desgasta, varían sus características (dimensiones, holguras, ...), lo cual hace que los sistemas en estudio no sean invariantes en el tiempo. Por esa razón, las ganancias presentadas como óptimas en un momento determinado, pueden no serlo a medida que el piñón sufre ese desgaste, necesitándose un reajuste de los controladores. El controlador de autoerección es especialmente sensible a este fenómeno. Por esa razón, se proporciona en la aplicación de interfaz de usuario la alternativa de levantar manualmente el péndulo cuando lo que se busca es solamente experimentar con los controladores de balanceo.

## **6.6. DESARROLLO DE LA APLICACIÓN DE USUARIO**

La aplicación o interfaz de usuario se plantea de forma que, con la tarjeta NI myRIO conectada a ordenador y maqueta, el usuario sea capaz de controlar los experimentos desde el ordenador, seleccionando parámetros, comienzo y parada de los experimentos, etc.

Para conseguirlo, se plantea la arquitectura de la *Figura 142*, en la que existe un programa ejecutado en el ordenador, otro programa en la tarjeta NI myRIO, que se encarga del control en tiempo real de la maqueta, y una librería de variables en red compartidas vía USB, encargadas de coordinar la ejecución de ambos programas y transferir datos de uno a otro. Además, existe otra librería en el programa de ordenador, para la comunicación exclusiva entre distintos subprogramas del programa alojado en éste, sin necesidad de comunicarse con la tarjeta NI myRIO.



*Figura 142. Arquitectura del programa de la aplicación*

Ambos programas están formados por distintos subprogramas, llamados VI (Virtual Instrument) en LabVIEW, que se ejecutan o no dependiendo de las elecciones tomadas por el usuario de la interfaz de usuario.

Se pueden observar en la *Figura 143* los subprogramas VI y las librerías existentes en el proyecto de LabVIEW que conforman la aplicación desarrollada. Bajo la pestaña "My Computer" se ve el código incluido en el programa de ordenador, mientras que bajo la pestaña "NI-myRIO-FAF (172.22.11.2)" se ve la parte del programa cargado a la tarjeta NI myRIO: Cabe destacar que, aunque la librería en red "Librería En Red.lvlib" contiene variables que se utilizan en ambos programas, dicha librería debe ser alojada en uno de los dos dispositivos; en este caso se ha elegido cargarla en la tarjeta NI myRIO. La librería exclusiva del programa de ordenador "Librería PC.lvlib" forma parte de la aplicación ejecutada en el ordenador.

A continuación, se describe la arquitectura del programa y su código. Para una visión más detallada de este último, se puede referir al *ANEXO V: CÓDIGO*, presente al final de este documento.

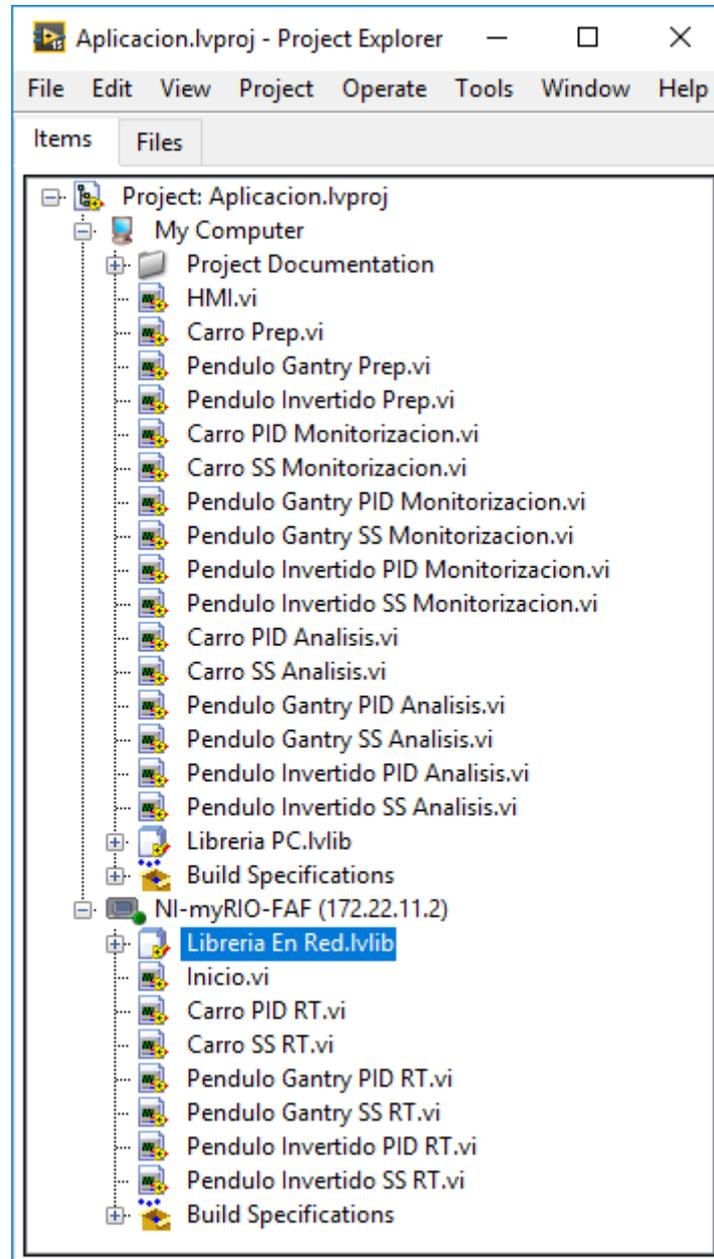


Figura 143. Ventana del proyecto de la aplicación

### 6.6.1. PROGRAMA DE ORDENADOR Y HMI

El programa de ordenador es el encargado de la ejecución de la interfaz de usuario. Se compone de los VI que se muestran en la *Figura 144* siguiente, junto a la dependencia entre ellos.

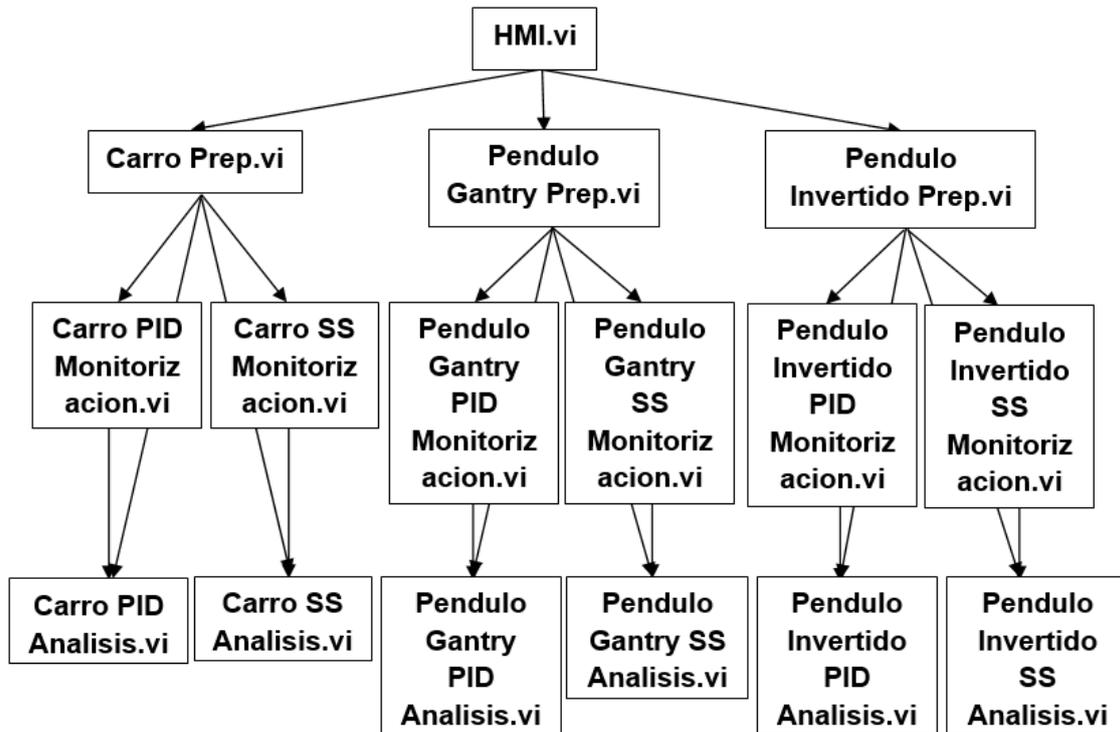


Figura 144. Arquitectura del programa de ordenador

#### 6.6.1.1. HMI.vi

La aplicación se inicia en este VI, en el que se muestran al usuario tres botones para elegir entre los tres sistemas de estudio: carro, péndulo Gantry o péndulo invertido. Además, existe un botón de fin de la aplicación.



Figura 145. Front Panel de HMI.vi

Básicamente, el código consiste en la inicialización de ciertas variables y un bucle “while” en el que se espera a un evento, mediante el cambio de valor de uno de los botones. En función del botón seleccionado, se carga el VI de preparación correspondiente o se finaliza el bucle y, por lo tanto, el programa. A continuación, se puede observar el código, o *Block Diagram*, del VI. A continuación, se muestra el código de “Inicio.vi”.

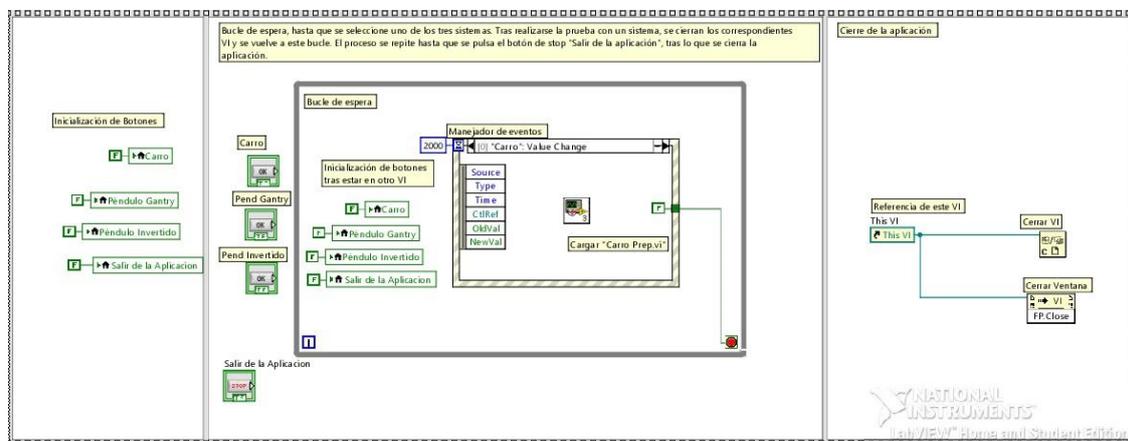


Figura 146. Código de HMI.vi

### 6.6.1.2. VI de preparación

Existen tres VI de preparación, uno para cada sistema: “Carro Prep.vi”, “Pendulo Gantry Prep.vi” y “Pendulo Invertido Prep.vi”. En estos VI, el usuario puede introducir los datos de la prueba que se desea realizar: modificar los parámetros del modelo; elegir entre prueba sobre el modelo, el sistema real o ambas; elegir el tipo de referencia entre rampa, escalón u onda cuadrada; y elegir el tipo de controlador, PID, asignación de polos o LQR, y sus parámetros. Así mismo, existen indicadores que proporcionan datos sobre el sistema. Al terminar se debe pulsar el botón “Avanzar a control”. Si, en cambio, se quiere volver a la pantalla anterior existe un botón que lo permite.

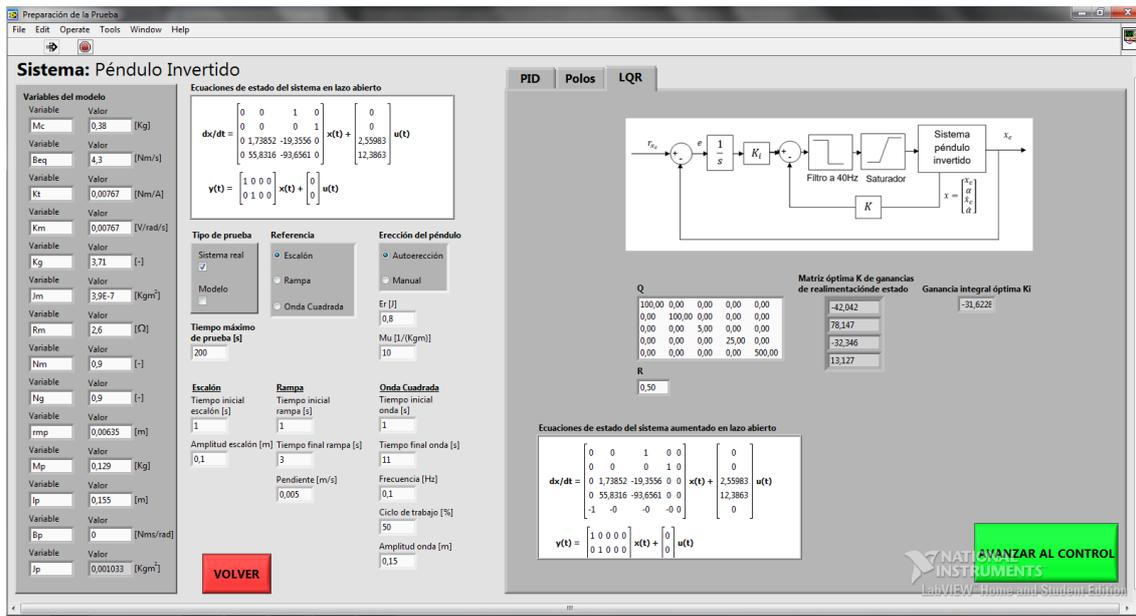


Figura 147. Front Panel de los VI de preparación

El código parte de una inicialización de ciertas variables, principalmente botones. Posteriormente, se inicia un bucle “while”, temporizado para no saturar la CPU, en el que se comprueba periódicamente la actualización de los controles y se actualizan los indicadores.

Finalmente, se transfieren los parámetros necesarios para la prueba a sus variables o variables en red correspondientes y, en función de la prueba seleccionada, se carga un VI. Si se ha seleccionado sistema real, se carga un VI de monitorización. Si, por el contrario, no se ha seleccionado, se carga un VI de análisis. Si el controlador seleccionado es PID, se carga el VI correspondiente a PID; y si se ha seleccionado asignación de polos o LQR, se carga el VI correspondiente a SS (State Space), espacio de estados. Si se ha pulsado el botón de volver, no se carga ningún subprograma, y se cierra este VI. Esta decisión se realiza mediante sentencias “Case” anidadas. Tras la ejecución del subprograma, el VI se cierra.

Se limitan los valores de los controles en el panel frontal entre los valores admitidos, de los que se proporciona una lista en el ANEXO IV: MANUAL DE USUARIO DE LA APLICACIÓN, en su apartado 7.2.4. Restricciones de los valores de los controles. Además, se comprueba que se haya seleccionado al menos alguna de las dos opciones, sistema real o modelo, que no haya incongruencias en los tiempos de prueba y los de la referencia, que los polos complejos estén acompañados de su conjugado y que la matriz Q sea simétrica semidefinida positiva.

A modo de ejemplo, a continuación, se muestra el código correspondiente a “Pendulo Invertido Prep.vi”.

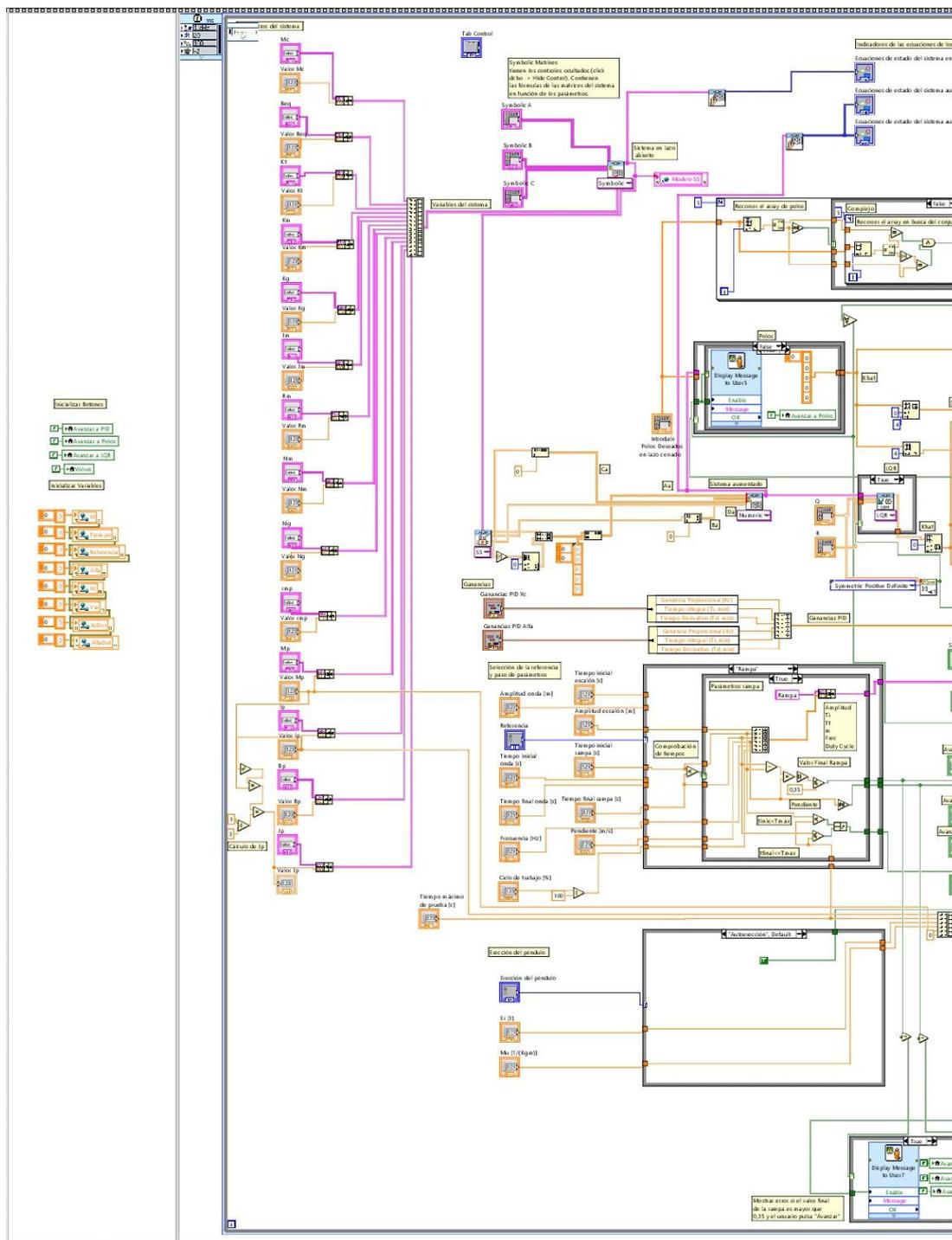


Figura 148. Código de Pendulo Invertido Prep.vi I

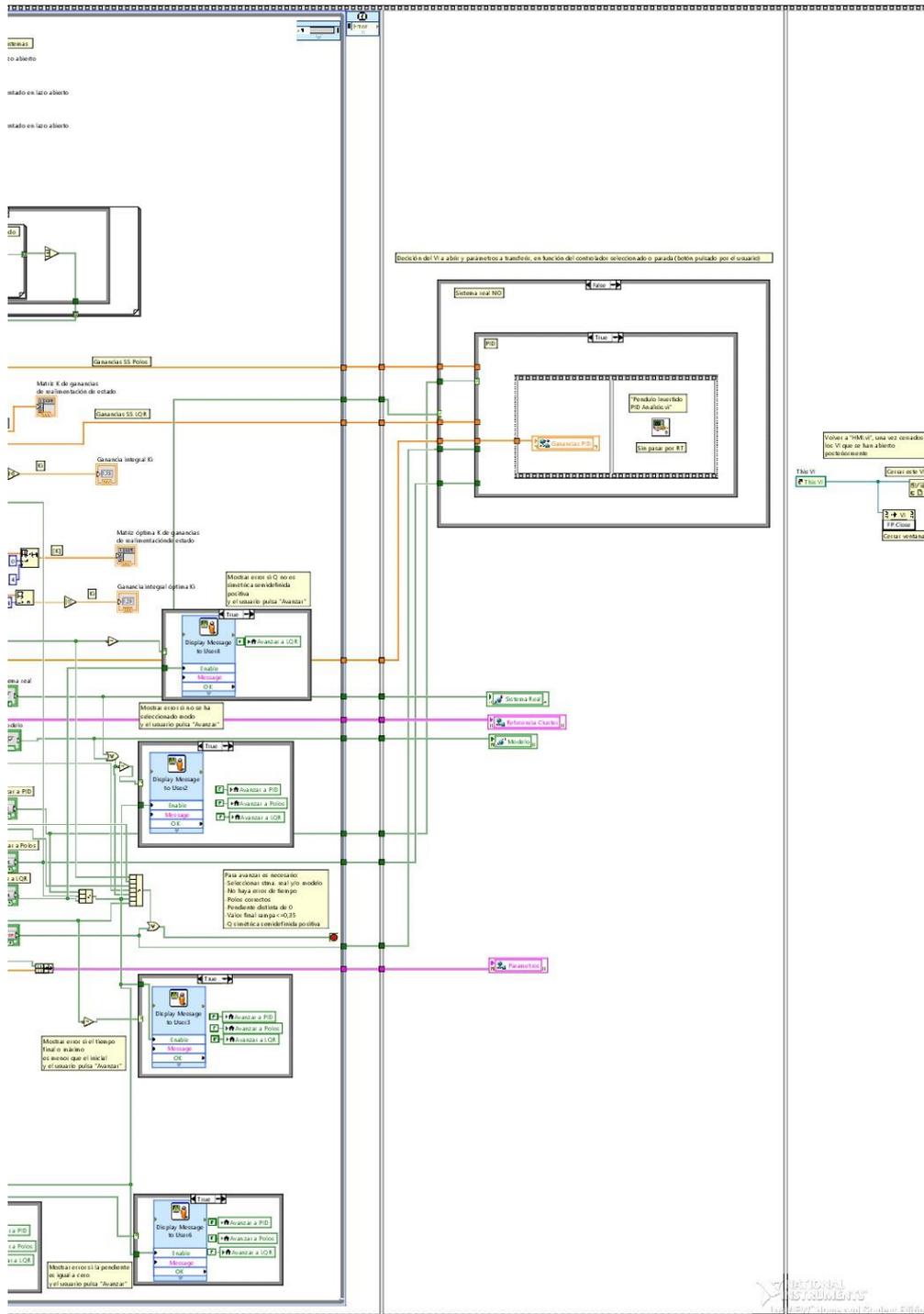


Figura 149. Código de Pendulo Invertido Prep.vi II

### 6.6.1.3. VI de monitorización

Existen seis VI de monitorización, dos para cada sistema, uno para PID y otro para controladores basados en espacio de estados (SS). Estos VI permiten monitorizar mediante gráficos en tiempo real las pruebas llevadas a cabo sobre el sistema real, así como controlar el inicio y el final de la prueba. Además, muestran indicadores de paradas de emergencia para final de carrera o un ángulo excesivo, así como indicadores de fase de erección o control de balanceo en el caso del péndulo invertido.

Por último, existe un botón que permite avanzar al análisis de la prueba una vez terminada la misma.

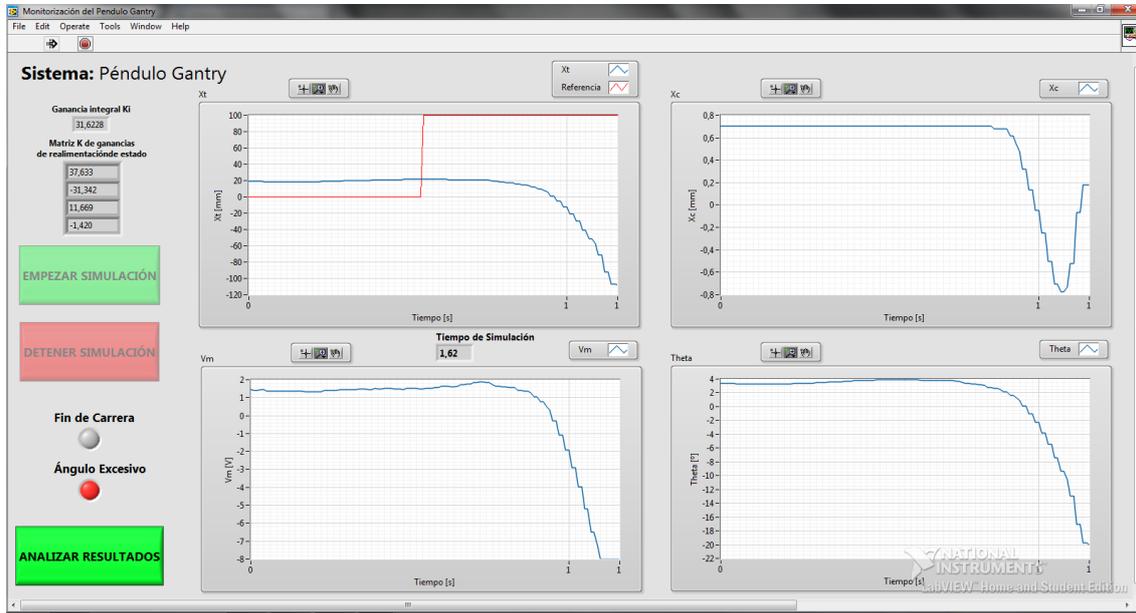


Figura 150. Front Panel de los VI de monitorización

El código comienza con la inicialización de indicadores y la carga de variables, entre ellas la escritura de la subvariable *VI en RT* en la variable *Control RT*, que selecciona el VI correspondiente en el programa en tiempo real. Posteriormente, mediante una estructura de “evento”, se espera a que el usuario pulse el botón de inicio de simulación, cuya orden también se transfiere al programa en la tarjeta NI myRIO por medio de la variable *Control RT*.

Tras la orden, se inicia un bucle “while” temporizado en el que se lee periódicamente la variable *Valores Actualizados*, que contiene los últimos valores registrados en la prueba en tiempo real, y se muestran en sus correspondientes gráficas. Además, se comprueban las condiciones de parada, que son el accionamiento del botón de detención de la simulación o una parada de emergencia, por ángulo excesivo, fin de carrera o rebasamiento del tiempo límite, registrados en el programa en tiempo real y leídos de la variable *Control RT*.

Una vez acabado el proceso de monitorización, se transfiere el modo de parada mediante la variable *Control RT* al programa en tiempo real, para que en caso de que la orden se haya dado desde el botón de detención se finalice el bucle de control en tiempo real. A continuación, se espera mediante una estructura de evento a que el usuario pulse el botón de avanzar al análisis de los resultados. Finalmente, cuando esto ocurre, se carga el VI de análisis que corresponde. Más tarde, cuando la subrutina acaba, se cierra este VI.

A modo de ejemplo, a continuación, se muestra el código correspondiente a “Péndulo Invertido SS Monitorización.vi”.

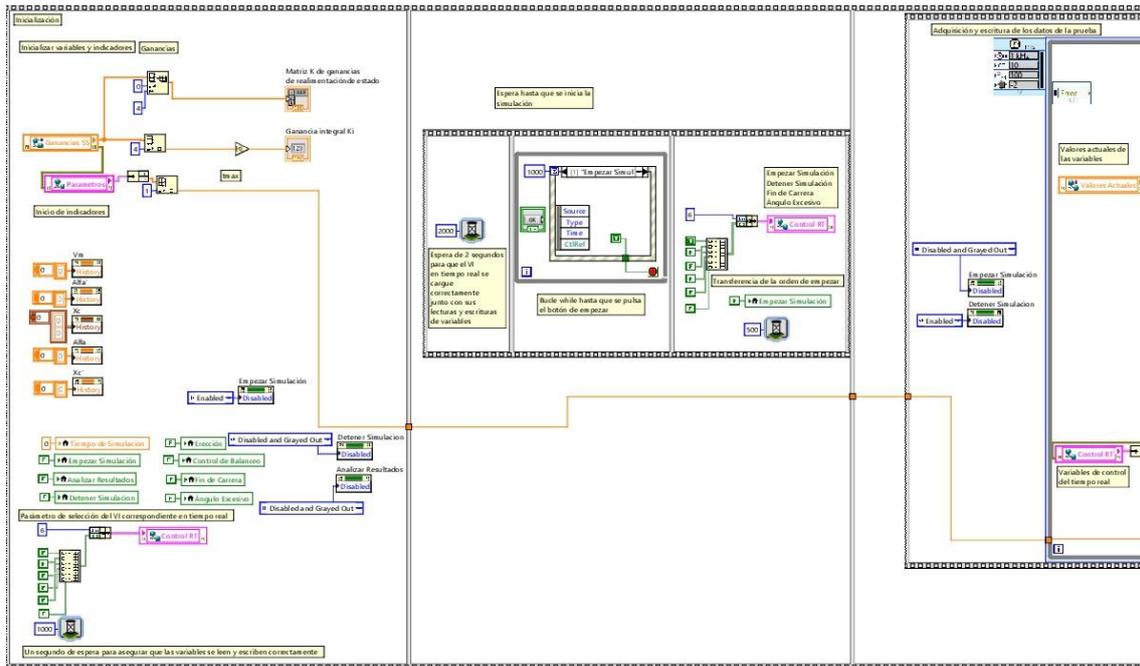


Figura 151. Código de Pendulo Invertido SS Monitorizacion.vi I

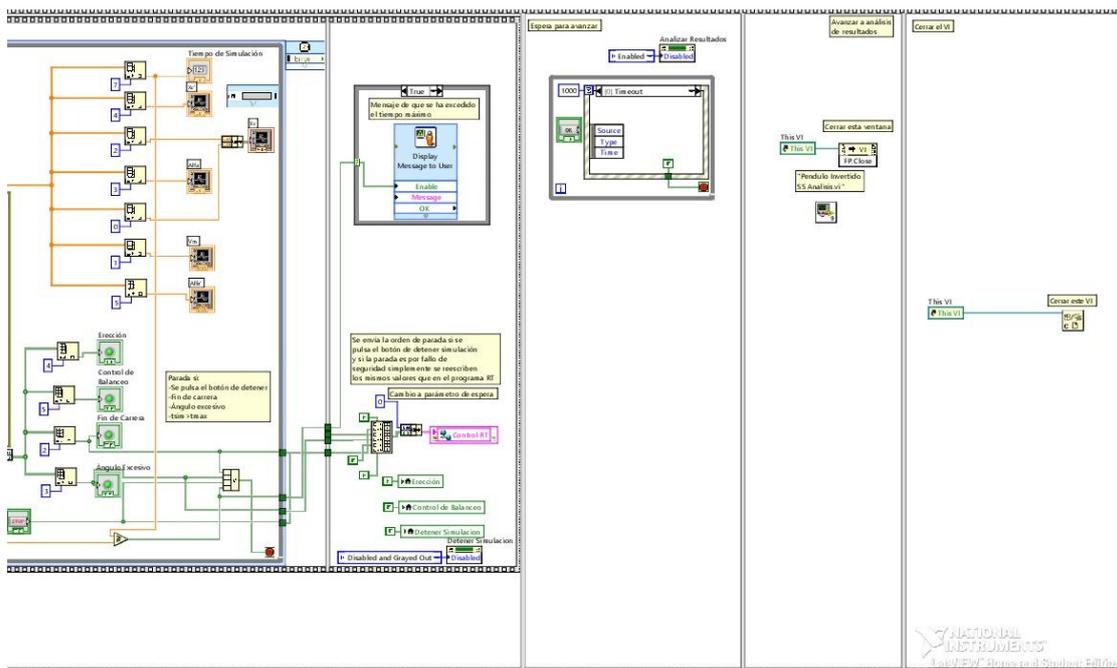


Figura 152. Código de Pendulo Invertido SS Monitorizacion.vi II

### 6.6.1.4. VI de análisis

El programa cuenta con seis subprogramas VI de análisis, dos para cada sistema dependiendo del controlador elegido, como en el caso de la monitorización. Estos VI muestran gráficas en las que se exponen los resultados de la prueba, ya sea para el modelo, para el sistema real o para ambos, dependiendo de la opción seleccionada. Además, existe un botón para terminar con la prueba, y volver a la pantalla de inicio, en la que se puede elegir el sistema de estudio. En el caso del péndulo invertido, se muestra también el tiempo de conmutación entre el modo de erección y el modo de control de balanceo.

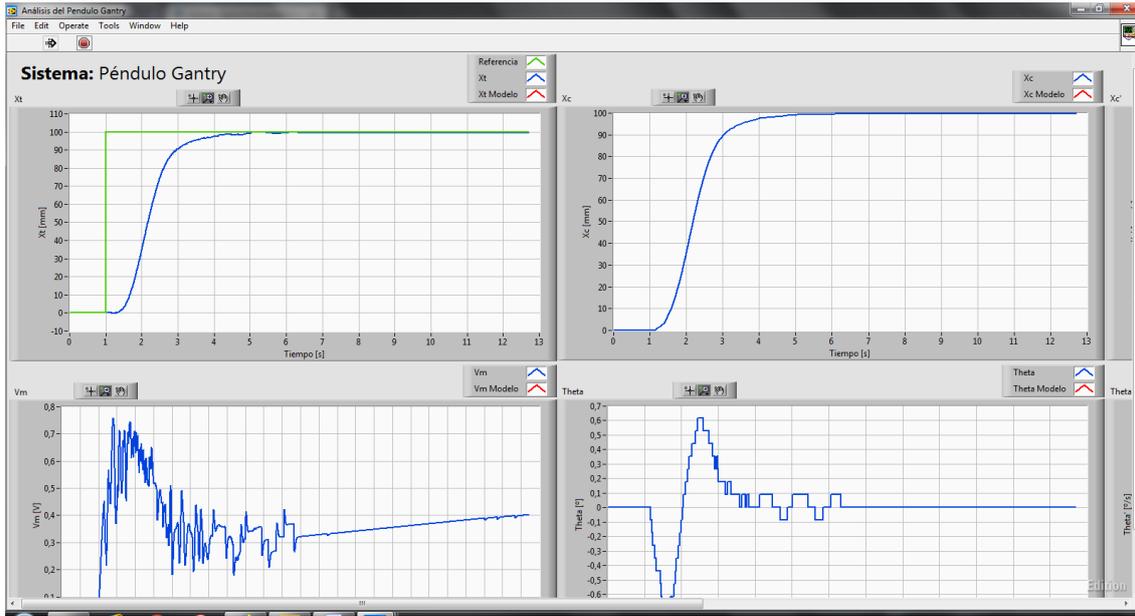


Figura 153. Front Panel I de los VI de análisis

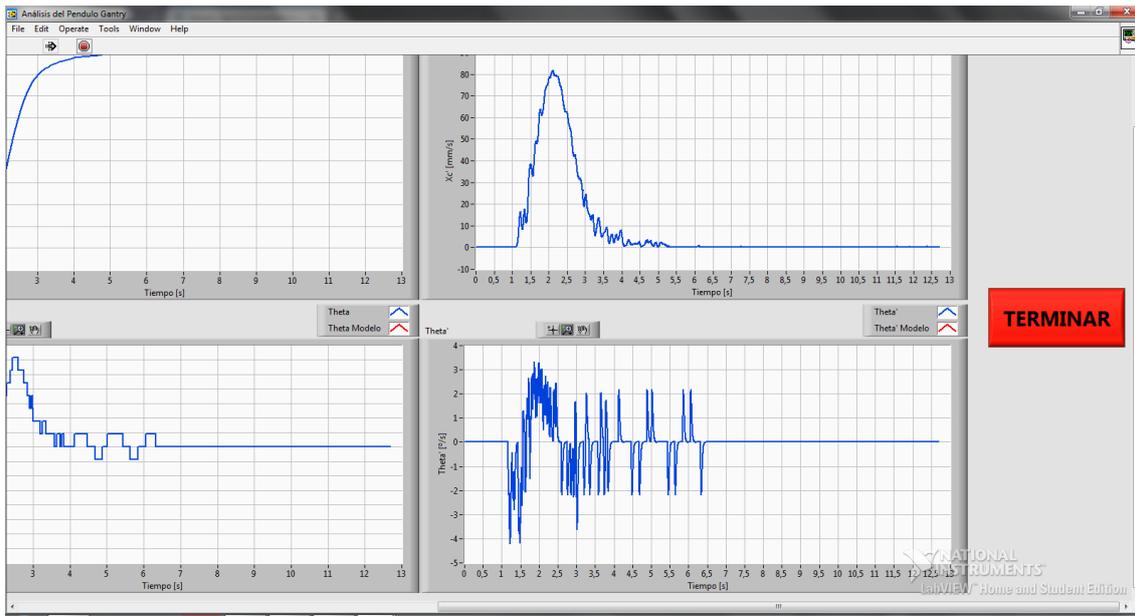


Figura 154. Front Panel II de los VI de análisis

En la ejecución, primeramente, se cargan todas las variables de las pruebas realizadas. La variable booleana *Modelo*, contenida en la librería del ordenador, se utiliza para determinar mediante una sentencia case si se realiza o no simulación sobre el modelo. En caso de que su valor sea verdadero, se realiza mediante un bucle de control y simulación, la prueba sobre el modelo y se registran los resultados en distintos vectores. A continuación, mediante dos sentencias “Case” anidadas, se selecciona la representación de resultados adecuada en las gráficas, en función del modo seleccionado (modelo, sistema real o ambos). Finalmente, se espera mediante una estructura de “evento” a que se pulse el botón terminar, tras lo que se cierra el VI.

A modo de ejemplo, a continuación, se muestra el código correspondiente a “Péndulo Invertido SS Analisis.vi”.

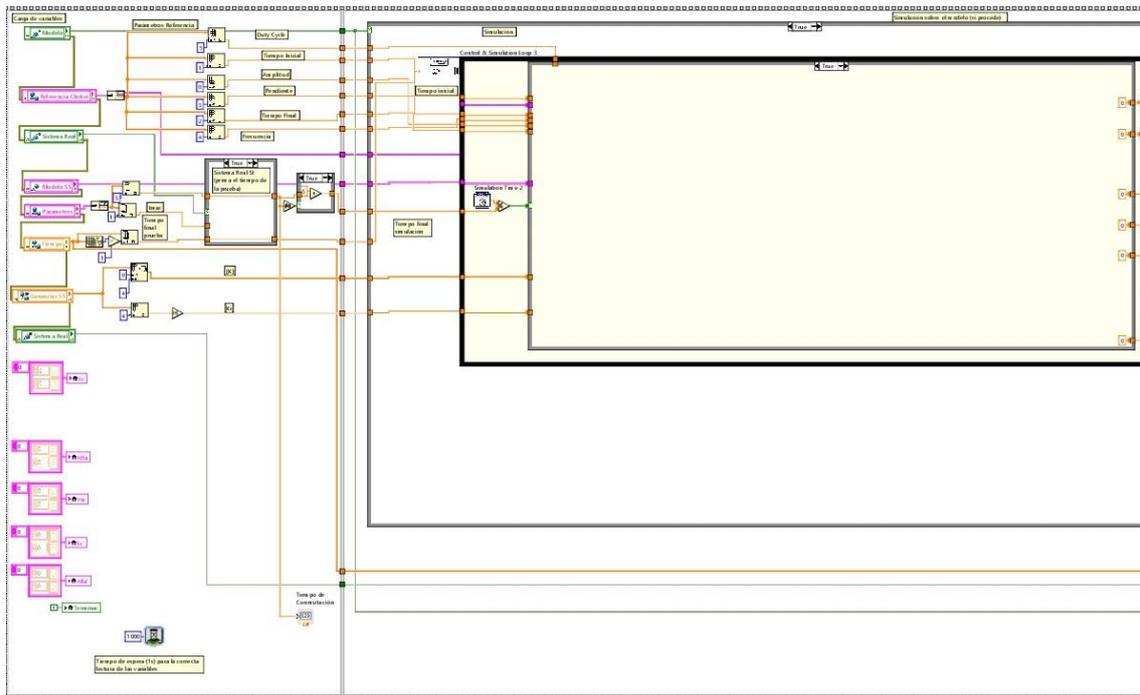


Figura 155. Código de Pendulo Invertido SS Analysis.vi I

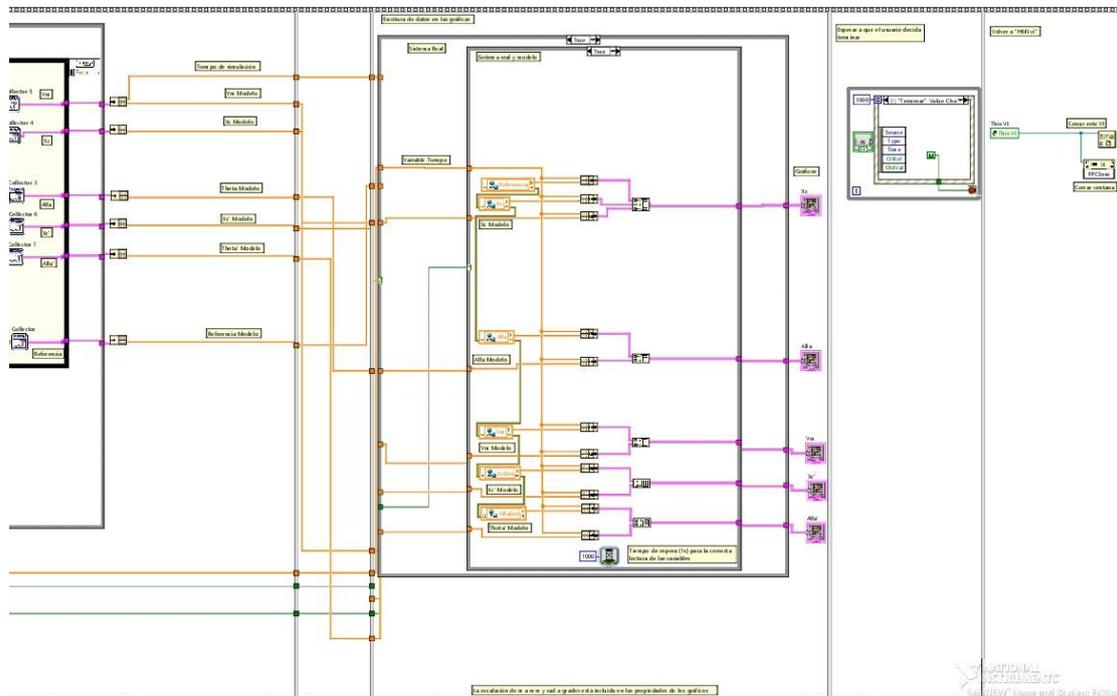


Figura 156. Código de Pendulo Invertido SS Analysis.vi II

### **6.6.1.5. Librería PC.lvlib**

En la *Tabla 8* siguiente, se listan las variables en la librería a la que se accede únicamente desde el ordenador y, por lo tanto, no está compartida en red. Estas variables se llaman Single Process Shared Variables en LabVIEW.

<b>VARIABLE</b>	<b>TIPO</b>
<b>Modelo</b>	Boolean
<b>Sistema Real</b>	Boolean
<b>Modelo SS</b>	Cluster

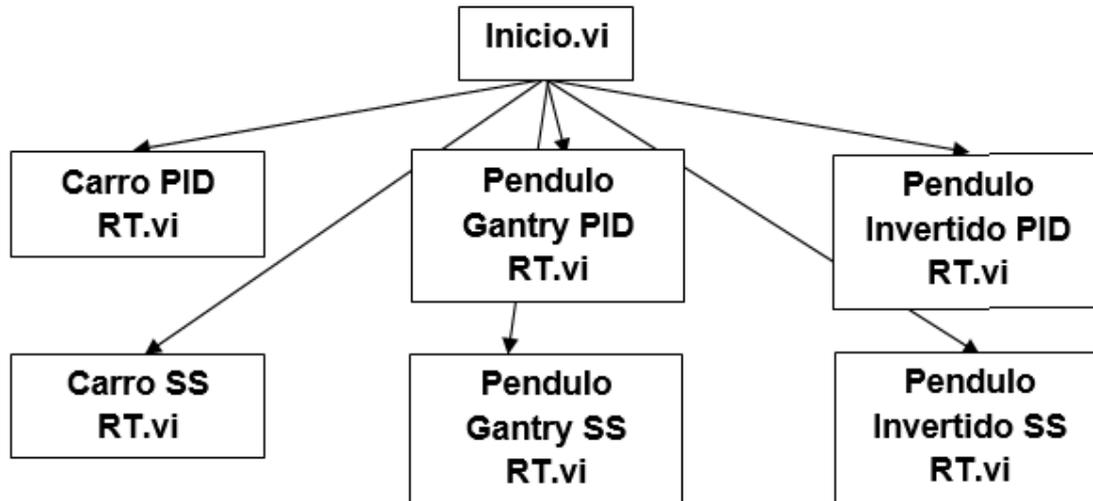
*Tabla 8. Variables de la librería Librería PC.lvlib*

Las variables *Modelo* y *Sistema Real* son variables booleanas que registran el tipo de prueba seleccionada por el usuario, es decir, si se ha seleccionado hacer la prueba sobre el modelo y/o sobre el sistema real, respectivamente.

La variable *Modelo SS* almacena el sistema de estados generado a partir de los parámetros del modelo establecidos por el usuario y las ecuaciones que definen cada sistema, utilizado posteriormente para la simulación sobre el modelo.

### 6.6.2. PROGRAMA DE CONTROL EN TIEMPO REAL

Se trata del programa alojado en la memoria de la tarjeta NI myRIO, y que comienza su ejecución automáticamente cuando ésta se alimenta o reinicia. En este caso, no existe ninguna pantalla del programa que se presente en la pantalla del ordenador. Los VI que integran el programa y su dependencia mutua es la que se muestra en la *Figura 157* que se encuentra a continuación.



*Figura 157. Arquitectura del programa de control en tiempo real*

#### 6.6.2.1. Inicio.vi

Este es el VI principal de la aplicación en tiempo real, en el que se inicia la misma y desde el que se gestiona la llamada al VI que se encarga del control en tiempo real de cada sistema y tipo de controlador.

El código, tras una breve espera para que el SVE se inicie correctamente, inicializa las variables de control en tiempo real y los LEDs indicadores del estado del programa. Posteriormente, se entra en un bucle while en el que se lee la variable *VI en RT* que determina el control en tiempo real que se va a ejecutar, es decir, el VI que se va a ejecutar. Esta variable se lee sólo cuando hay un Timeout (programado para un segundo) y cuando hay un cambio en la misma, de modo que no se sobrecarga el procesador. Tras la ejecución del VI correspondiente, se vuelve al bucle de "Inicio.vi", que no termina hasta que se apaga la tarjeta.

A continuación, se muestra el código de "Inicio.vi".

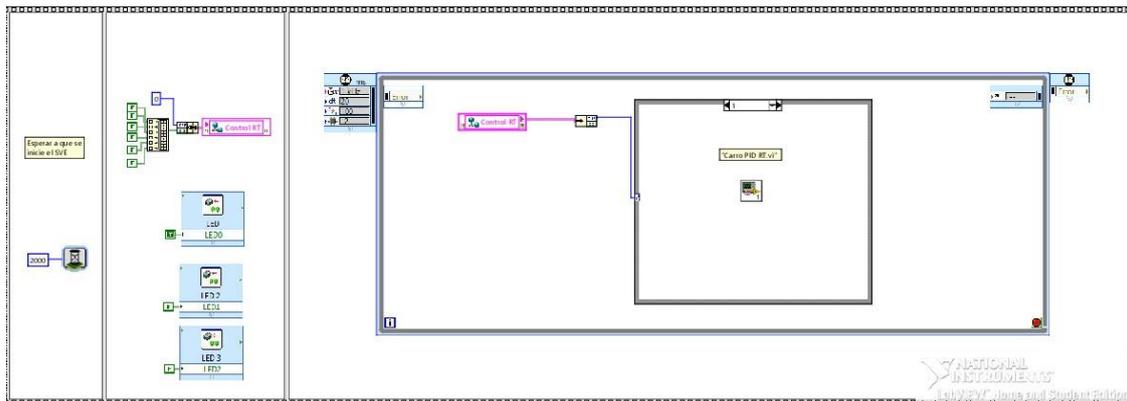


Figura 158. Código de Inicio.vi

### 6.6.2.2. VI de control en tiempo real

Existen seis VI distintos para el control en tiempo real, dos por sistema, uno para el control con algoritmos PID y otro para el control por realimentación del estado (asignación de polos y LQR).

Cada VI de este tipo comienza con la carga de los parámetros necesarios para la prueba y la inicialización de los registros. Además, se abren las referencias de las entradas y salidas del sistema (encoders y salida analógica) y se resetean los encoders.

Posteriormente, se pasa a un bucle “while” de espera, en el que se lee la variable *Control RT*, esperando la orden de comenzar la simulación. El bucle sólo se ejecuta cada segundo o cuando se detecta un cambio en la variable. A continuación, el programa se introduce en el bucle temporizado, cuyo periodo es el periodo de muestreo, que ejecuta el control.

En el bucle de control se realiza la lectura de los encoders, se calculan las derivadas mediante la ecuación en diferencias finitas de los filtros derivativos, se genera la referencia seleccionada, se ejecuta el algoritmo de control, se escriben la señal de control en la salida analógica que le corresponde, se actualizan los valores actuales de las variables del sistema, se aumenta el buffer que funciona de registro de la prueba y se comprueba que no se ha cumplido ninguna condición de parada de la prueba. Cuando esto último ocurre, se detiene el bucle de control.

Una vez finalizado el control, se escribe un voltaje nulo en la salida analógica de la tarjeta, para realizar una parada controlada de la maqueta y se cierran las referencias de las entradas y salidas. Finalmente, se escriben los registros en sus correspondientes variables y el modo de parada en la variable *Control RT*.

A modo de ejemplo, a continuación, se muestra el código correspondiente a “Pendulo Gantry PID RT.vi”.

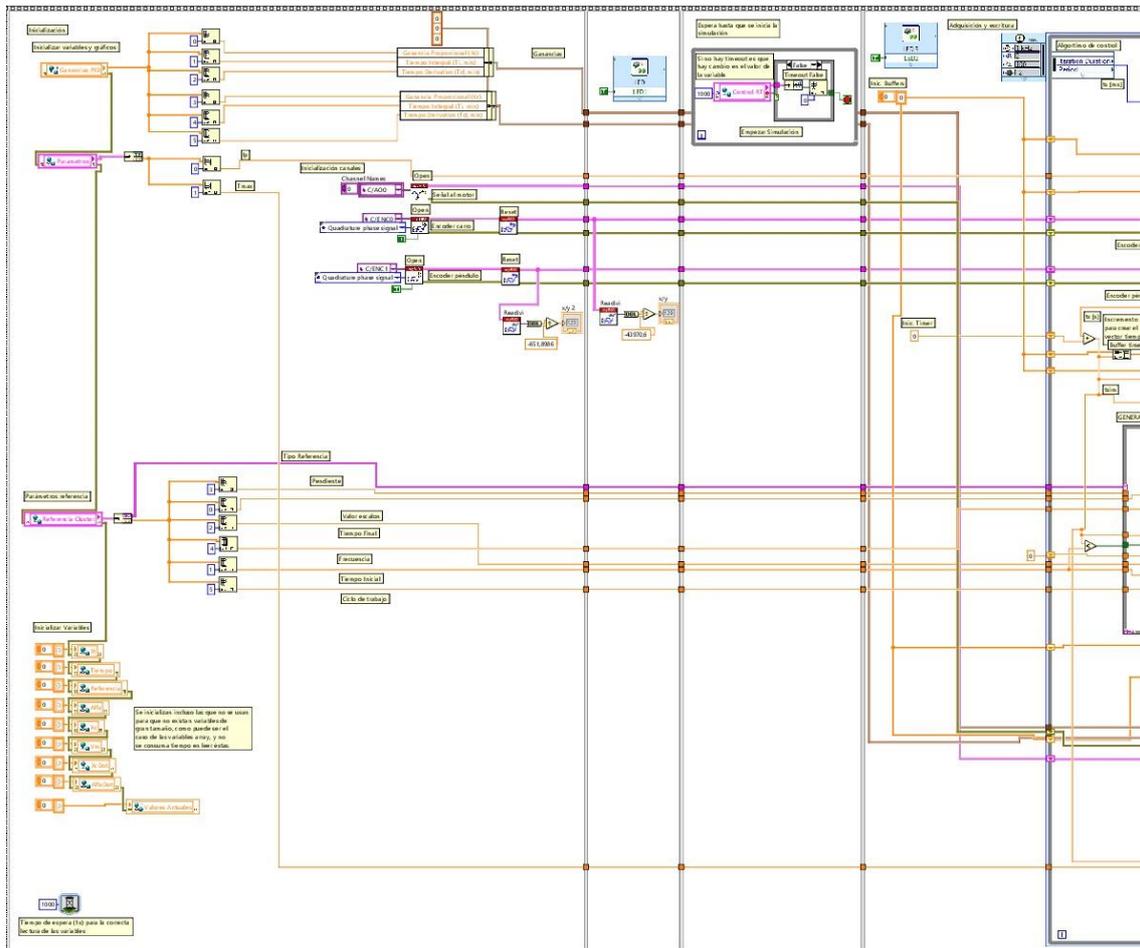


Figura 159. Código de Pendulo Gantry PID RT.vi I

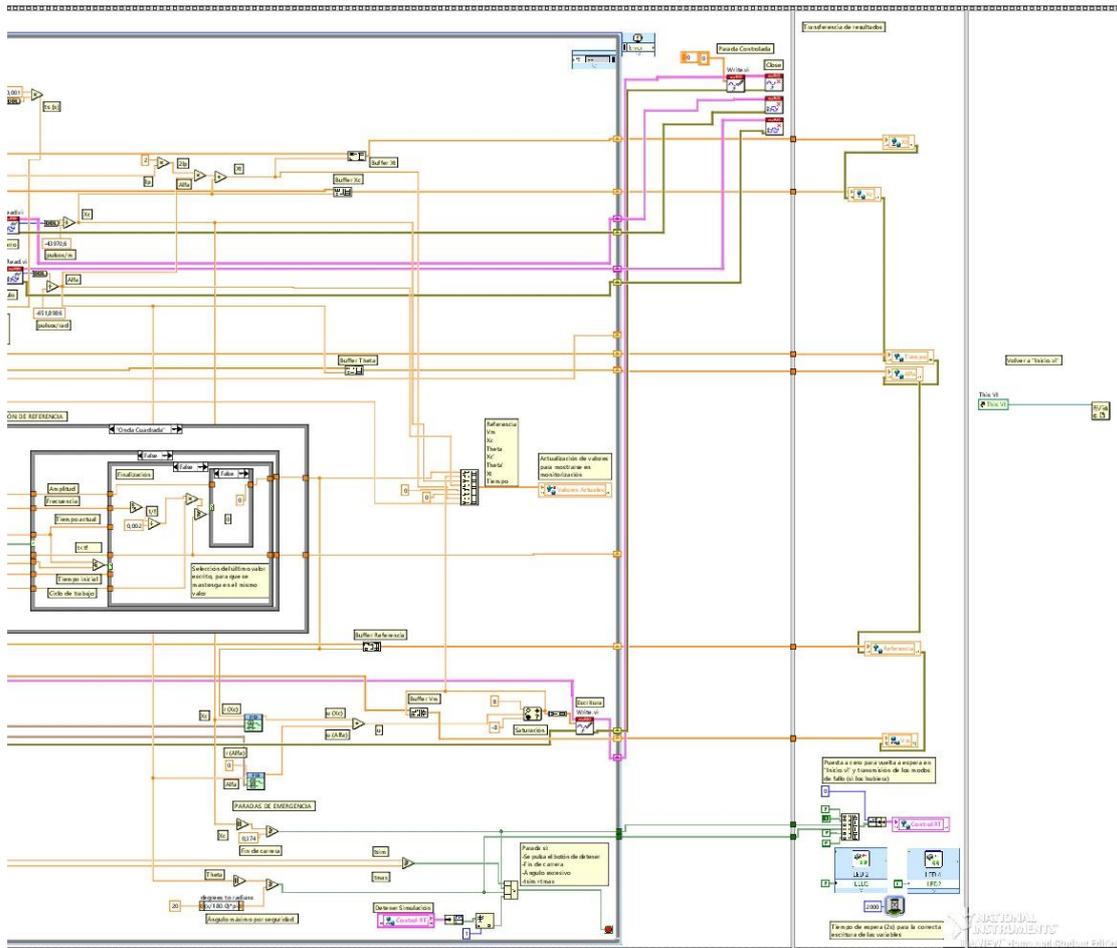


Figura 160. Código de Pendulo Gantry PID RT.vi II

En el caso de los VI correspondientes al sistema péndulo invertido, existe una fase adicional entre la inicialización de parámetros y el bucle de control. Se trata de un bucle temporizado en el que se lleva a cabo la erección del péndulo, manual o automática, según lo haya seleccionado el usuario.

Esta fase se ejecuta en un bucle temporizado como el mencionado anteriormente. En él se leen los valores de los encoders y se calculan las derivadas y, mediante una sentencia "Case", se ejecuta el algoritmo de control de autoerección, que determina el voltaje a escribir en la tarjeta o, si se ha seleccionado la erección manual, se escribe voltaje nulo.

Según el valor de la variable  $l_p$ , se determina si se da el caso de péndulo corto o péndulo largo. En función de este valor, no sólo se seleccionan los valores del algoritmo de autoerección, sino que también se añaden -0,15 metros o 0,2 metros al valor leído por el encoder, dado que, como se ha mencionado en el apartado 6.5.3.4. *Autoerección*, para el péndulo corto se inician las pruebas 150 milímetros hacia la izquierda de la posición central de la pista y para el péndulo largo 200 milímetros hacia la derecha de la misma.

Además, aunque sólo es crítico para el péndulo largo, se programa un impulso inicial de voltaje negativo (-5V) que dura 20 milisegundos. De esta manera, se asegura que el movimiento del carro comienza hacia la izquierda de la pista.

El bucle termina cuando el ángulo del péndulo es mayor que el valor fijado,  $170^\circ$  o  $175^\circ$  dependiendo del control elegido, y el péndulo está lo suficientemente vertical como para comenzar el control de balanceo. El tiempo de conmutación entre ambos controladores se registra dentro de la variable *Parametros*.

Así mismo, tras la conmutación, se mantiene el sistema durante cinco segundos en la posición de la conmutación para garantizar la estabilidad en la misma, y posteriormente se inicia la prueba con la referencia seleccionada por el usuario.

En este caso, tanto para registrar el ángulo  $\alpha$  en la erección como para usarlo en el algoritmo de control de balanceo, es necesario hacer la conversión de ángulo que se explica en el apartado 6.5.3.4. *Autoerección*.

A modo de ejemplo, a continuación, se muestra el código correspondiente a “Pendulo Invertido SS RT.vi”.

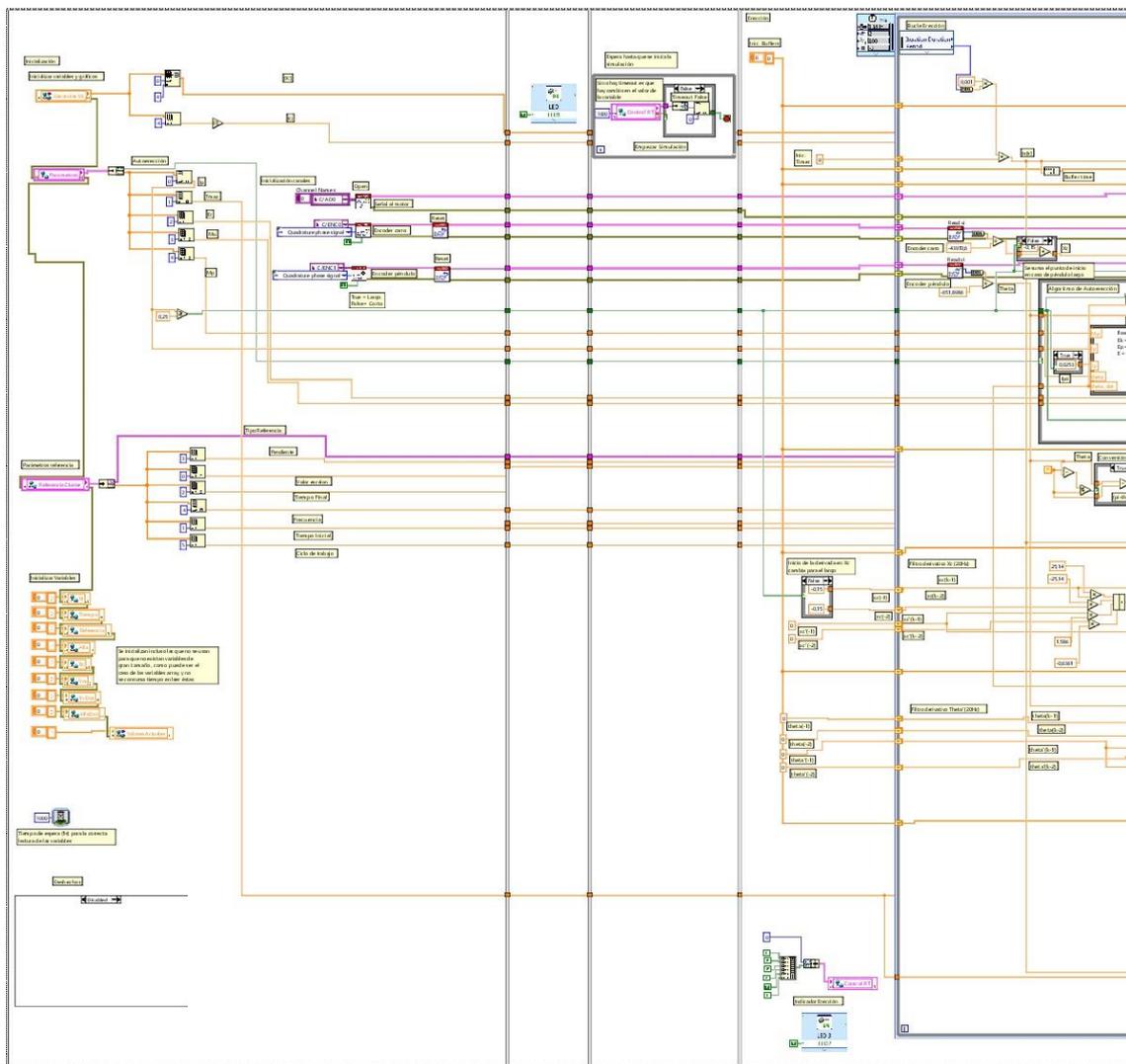


Figura 161. Código de Pendulo Invertido SS RT.vi I

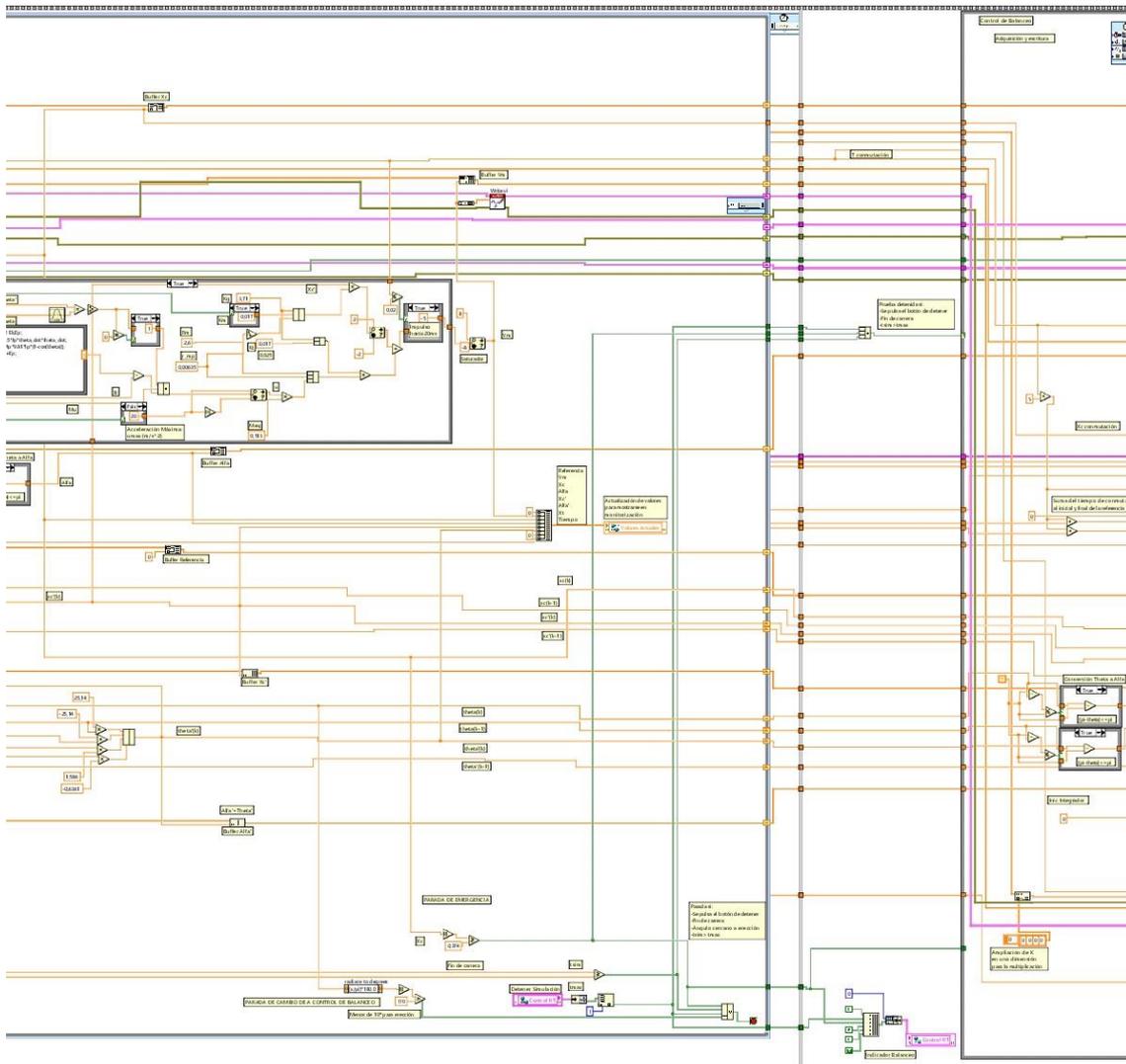


Figura 162. Código de Pendulo Invertido SS RT.vi II

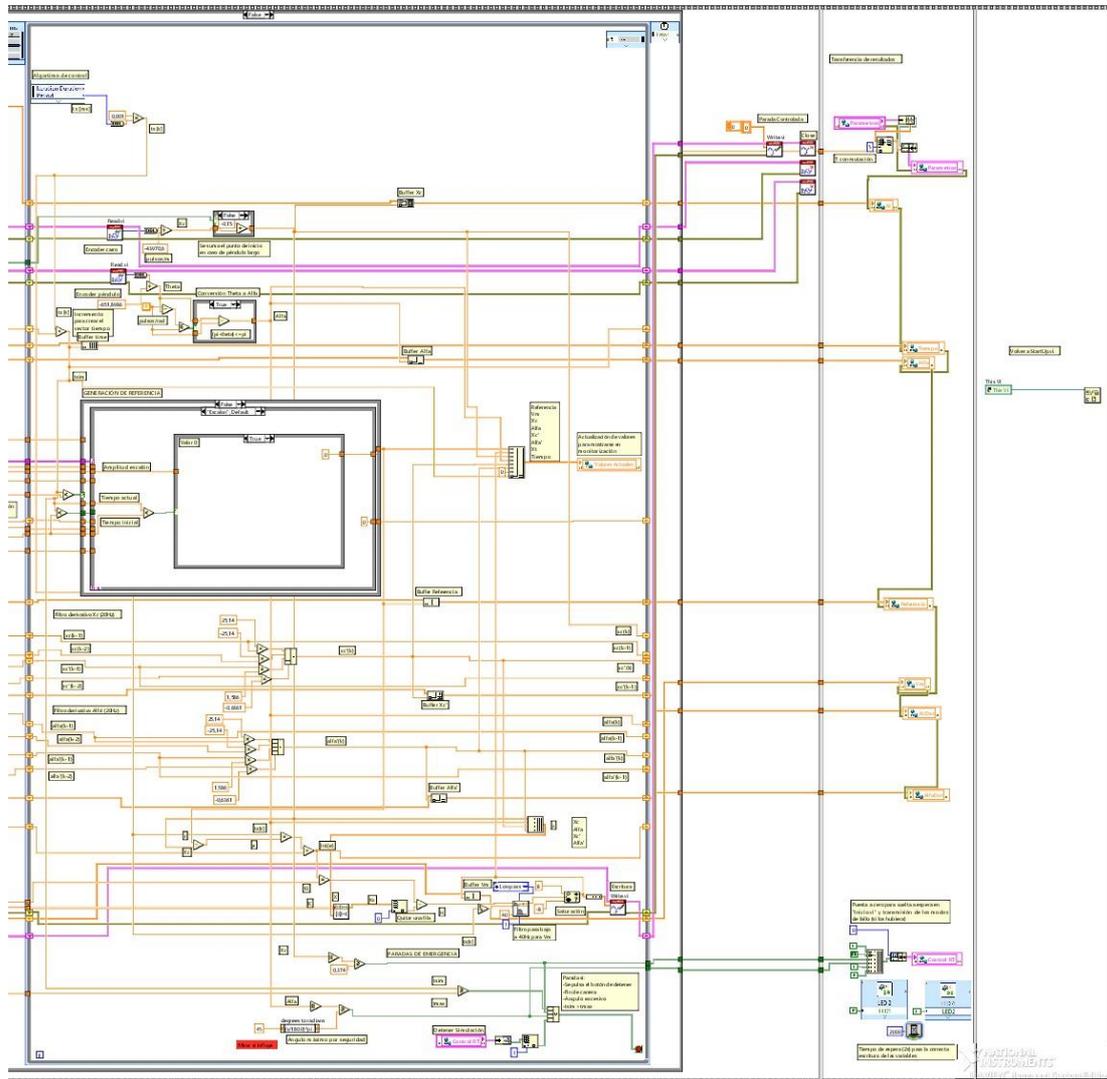


Figura 163. Código de Pendulo Invertido SS RT.vi III

### 6.6.2.3. LEDs indicadores de la ejecución

Para hacer un seguimiento de la ejecución del programa, tanto en las pruebas realizadas para su correcto funcionamiento como de cara a que el usuario compruebe que la ejecución es correcta, se utilizan LEDs de la tarjeta.

El LED0 se enciende al iniciar la tarjeta, en cuanto ésta está preparada para recibir la orden de iniciar uno de los seis sistemas en tiempo real. Permanece encendido durante toda la ejecución del programa.

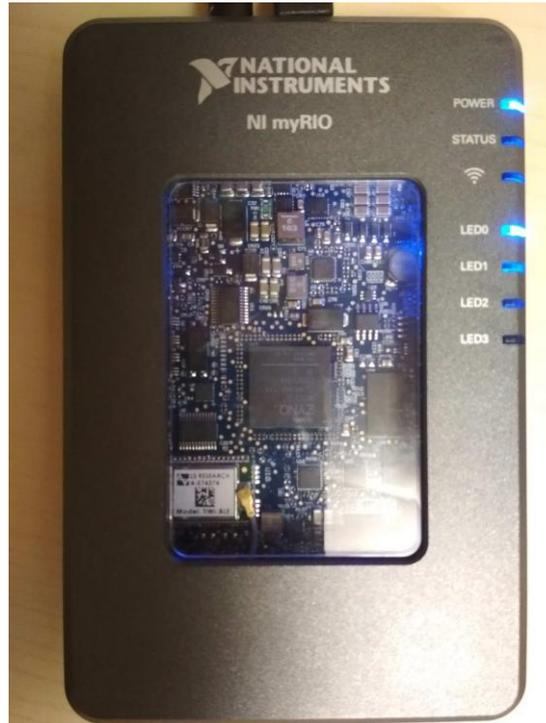


Figura 164. Tarjeta NI myRIO lista para iniciar la aplicación de ordenador

El LED1 se activa cuando el programa se encuentra en el subprograma en tiempo real que se debe ejecutar y con la inicialización y lectura de parámetros realizada, es decir, cuando la tarjeta está lista para recibir la orden de inicio de la simulación. Permanece encendido hasta que se cierra el VI, tras acabar la rutina de control.

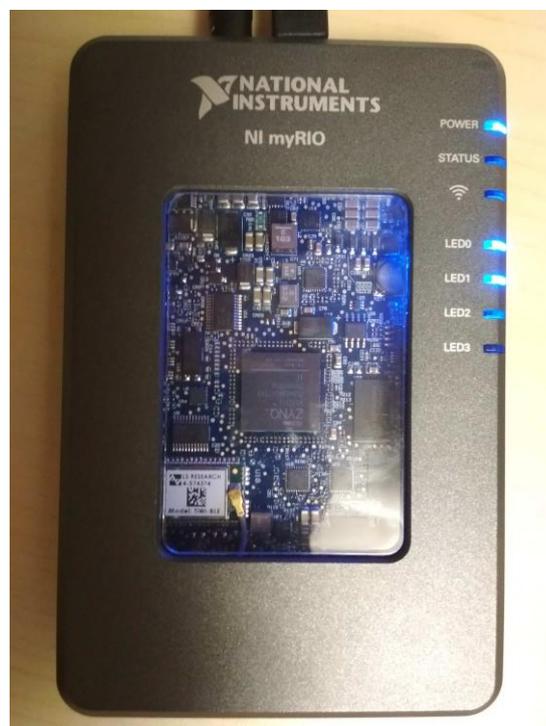


Figura 165. Tarjeta NI myRIO lista para iniciar la prueba en tiempo real

El LED2 se activa cuando comienza el bucle de control de un subprograma en tiempo real. Permanece encendido hasta que el control en tiempo real termina.

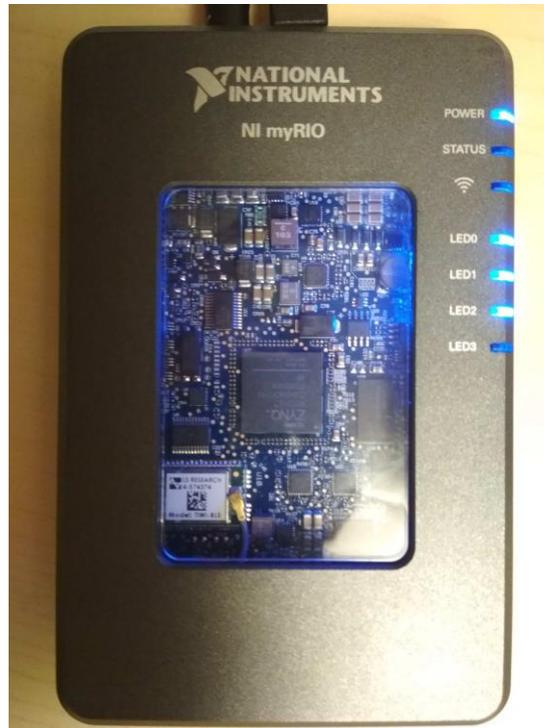


Figura 166. Tarjeta NI myRIO ejecutando el control en tiempo real sobre la maqueta IP02

### 6.6.3. LIBRERÍA EN RED

A continuación, en la *Tabla 5*, se listan las variables en red de la aplicación, llamadas Network-Published Shared Variables en LabVIEW. En LabVIEW únicamente existen los nombres de las variables, listados en la columna izquierda de la tabla. Sin embargo, debido a que muchas de éstas son compuestas, se ha incluido un desglose indicando qué parámetros o subvariables están incluidos en cada variable.

VARIABLE	TIPO	SUBVARIABLES	TIPO	SUBVARIABLES	TIPO
<b>Valores actuales</b>	Array of Double	Referencia Actual	Double		
		$V_m$ Actual	Double		
		$X_c$ Actual	Double		
		Ángulo Actual	Double		
		$X_c'$ Actual	Double		
		Ángulo' Actual	Double		
		$X_t$ Actual	Double		
<b>Control RT</b>	Cluster	VI en RT	Integer 32		
		Control Booleanos RT	Array of Boolean	Empezar Simulación	Boolean
				Detener Simulación	Boolean
				Fin de Carrera	Boolean
				Ángulo Excesivo	Boolean
				Erección	Boolean
				Balanceo	Boolean
<b>Ganancias PID</b>	Array of Double	$K_c$ de $X_c$	Double		
		$T_i$ de $X_c$	Double		
		$T_d$ de $X_c$	Double		
		$K_c$ de Ángulo	Double		
		$T_i$ de Ángulo	Double		
		$T_d$ de Ángulo	Double		
<b>Ganancias SS</b>	Array of Double	$K$	Array of Double	$K_1$	Double
				$K_2$	Double
				$K_3$	Double
				$K_4$	Double
		$-K_i$	Double		

Referencia Cluster	Cluster	Tipo	String		
		Parámetros	Array of double	Amplitud	Double
				T final	Double
				Pendiente	Double
				Frecuencia	Double
				Duty Cycle	Double
Parametros	Cluster	Autoerección	Boolean		
		Parámetros Generales	Array of Double	$l_p$	Double
				T máximo	Double
				$E_r$	Double
				$\mu$	Double
				$M_p$	Double
		T conmutación	Double		
Alfa	Array of Double				
AlfaDot	Array of Double				
Referencia	Array of Double				
Tiempo	Array of Double				
Vm	Array of Double				
Xc	Array of Double				
XcDot	Array of Double				
Xt	Array of Double				

Tabla 9. Variables de la librería en red Librería En Red.Ivlib

La variable *Valores Actuales* es un vector que almacena el valor instantáneo de cada variable del bucle de control en cada iteración, y se utiliza para, con un tiempo de bucle menor, mostrar los valores de las variables del sistema en los VI de monitorización del programa del ordenador, de manera que el usuario pueda ver la evolución del sistema mientras una prueba en tiempo real está en marcha.

La variable *Control RT* se utiliza para controlar la ejecución de las pruebas en tiempo real. Por un lado, dispone de una variable de tipo entero que, según el valor que se le asigna, llama a la subrutina en tiempo real adecuada, en función de la elección del usuario. Por otro lado, contiene un vector con varios parámetros booleanos de comandos e indicadores que se encuentran en la pantalla en la monitorización de una prueba en tiempo real.

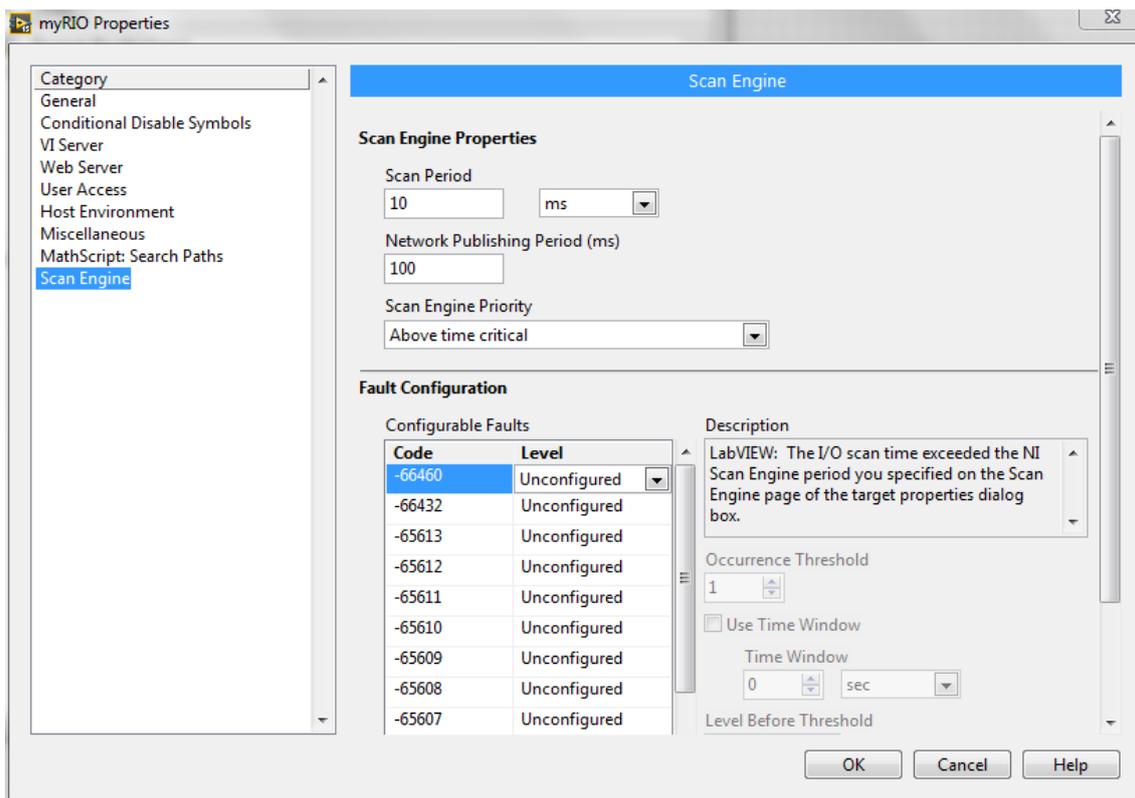
La variable *Ganancias PID*, como su nombre indica, contiene las ganancias introducidas por el usuario de los dos algoritmos PID que pueden llegar a usarse sobre un solo sistema, en el caso de los sistemas péndulo Gantry y péndulo invertido. Por otro lado, la variable *Ganancias SS* almacena la ganancia  $K$  de realimentación del

estado y la ganancia integral  $K_i$  cambiada de signo. En el caso del sistema carro, dos espacios del vector son ceros, ya que  $K$  sólo tiene dos valores.

En cuanto a *Referencia Cluster* es un variable que contiene el tipo de referencia seleccionada y los principales parámetros que la definen. Se utiliza para, tras ser introducida por el usuario, generarla en el programa en tiempo real. De la misma forma funciona la variable *Parametros*, aunque en este caso sirve para pasar parámetros de la prueba y del proceso de autoerección. La variable booleana *Autoerección* establece si la erección será automática o manual, por ejemplo. Además, contiene el tiempo máximo al que se limita la prueba y que, en caso de seleccionar únicamente una prueba sobre el modelo, es el tiempo de la simulación.

El resto de variables, que son vectores, no tienen una longitud definida, ya que almacenan los registros de las pruebas en tiempo real de las distintas variables de los sistemas, y la duración de éstas es variable. Se usan para representar los resultados en la fase de análisis.

El Shared Variable Engine (SVE) es la herramienta que maneja las variables en red. Se establecen los tiempos de escaneo y actualización de las variables que se muestran en la *Figura 167*.



*Figura 167. Configuración del Shared Variable Engine*

En las variables cuya longitud es conocida y constante, se activa la función FIFO, para garantizar que no se acceda a las variables de dos lugares del programa a la vez, y se pierda la información o funcionalidad de uno de los accesos. Es importante fijar, como se muestran en la *Figura 168* para la variable *Valores Actuales*, el número de elementos del vector, para no sobrescribir la variable.

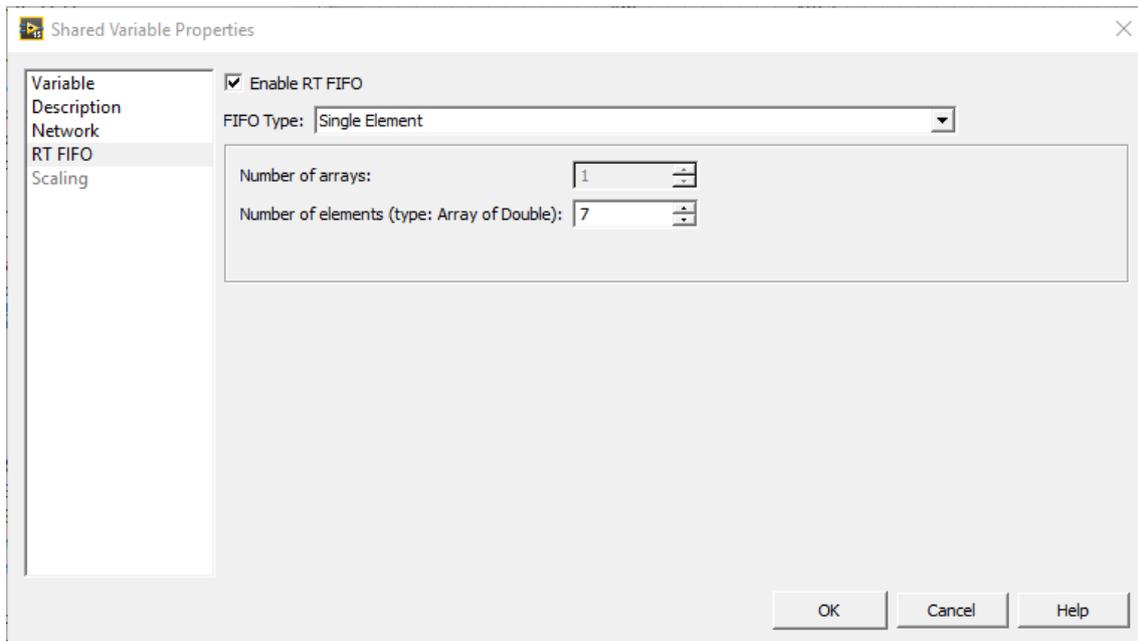


Figura 168. Configuración de la función FIFO de las variables

Las variables de longitud indefinida, es decir, los registros de las variables del sistema, se inician a cero al inicio de cada VI en tiempo real, aunque no se usen en dicho VI, para garantizar que no se quede un vector de gran tamaño almacenado en una variable y que al escanearse ralentice el proceso de actualización de variables. Con el mismo propósito de no ralentizar la actualización de variables, se agrupan las variables en Clusters o Arrays para minimizar el número de las mismas.

#### **6.6.4. PROCESO DE DESARROLLO**

El proceso seguido para desarrollar la aplicación es el siguiente.

Primeramente, se ha planteado la arquitectura explicada anteriormente. Una vez decidida la misma, se desarrolla “HMI.vi”. La estrategia seguida es la de elaborar la secuencia de subprogramas para un sistema y un controlador y, una vez obtenidos, extrapolar los programas a los homólogos para distintos controladores y sistemas. Por lo tanto, posteriormente, se lleva a cabo “Pendulo Gantry Prep.vi”, después “Pendulo Gantry PID Monitorizacion.vi”, más tarde “Pendulo Gantry PID RT” y, por último, “Pendulo Gantry PID Analisis.vi”. A lo largo del desarrollo de cada VI, se van probando pequeñas partes del mismo, comprobando el correcto funcionamiento de cada función o conjunto de bloques y depurándolo para obtener el comportamiento deseado.

Una vez desarrollada esta parte del programa, se comprueba el correcto funcionamiento del mismo, y se corrigen los errores.

Finalmente, se extrapola el código de esa rama del programa al resto, con las pertinentes adaptaciones, ampliación de funciones, corrección de parámetros y dimensiones... etc. Se realizan pruebas finales de funcionamiento de toda la aplicación, buscando modos de fallo, y se corrigen los errores.

## 6.6.5. PRODUCTO FINAL

Se busca que el producto final sea lo más accesible posible y que el software LabVIEW no sea necesario para ejecutar el programa. Por supuesto, es necesario suministrar un producto cuyo código no se pueda modificar.

Para conseguir esto, se crean aplicaciones a partir de los VI que conforman el programa en LabVIEW.

### 6.6.5.1. Aplicación en tiempo real

Se crea una aplicación en tiempo real a partir del programa en tiempo real, que se ejecuta en la tarjeta.

Para crearla, hay que seleccionar una nueva *Build Specification* del tipo *Real-Time Application*, haciendo clic derecho en *Build Specifications* bajo la tarjeta NI myRIO, como se muestra en la *Figura 169* siguiente.

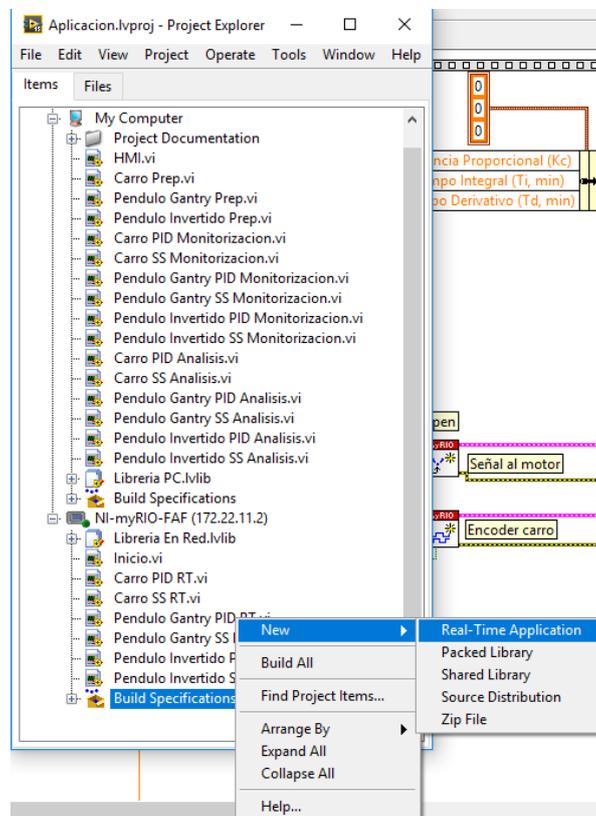
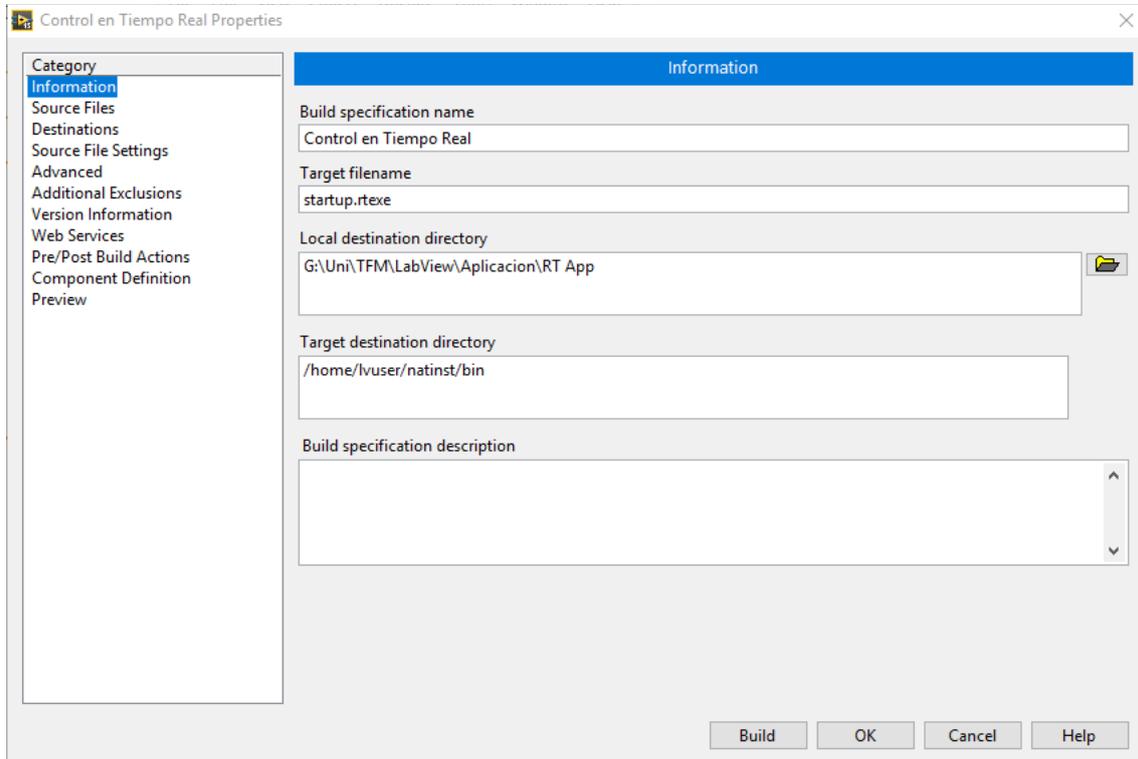


Figura 169. Creación de una nueva aplicación en tiempo real

Posteriormente, hay que fijar las propiedades de la compilación de esta nueva aplicación.

Primeramente, se especifica el nombre de la aplicación, en este caso *Control en Tiempo Real*. Además, es importante que el nombre de la aplicación en el target, es decir, en la tarjeta NI myRIO sea *startup.rtxe*, ya que es necesario ese nombre para que se ejecute nada más encender la tarjeta. A continuación, han de definirse el directorio en el que se guarda la aplicación dentro del ordenador de desarrollo y en el que se guardará al ser cargada en la tarjeta. Este último no se modifica con respecto

al directorio predeterminado. En la *Figura 170* siguiente se puede observar como rellenar estos campos.



*Figura 170. Información de la aplicación en tiempo real*

En la pestaña *Source Files*, se seleccionan los VI del proyecto que forman parte de la aplicación. En *Startup Vis* se debe incluir el VI con el que da comienzo el programa, en este caso "Inicio.vi". El resto de VI y librerías se cargan en *Always Included*, como se muestra en la *Figura 171* inferior.

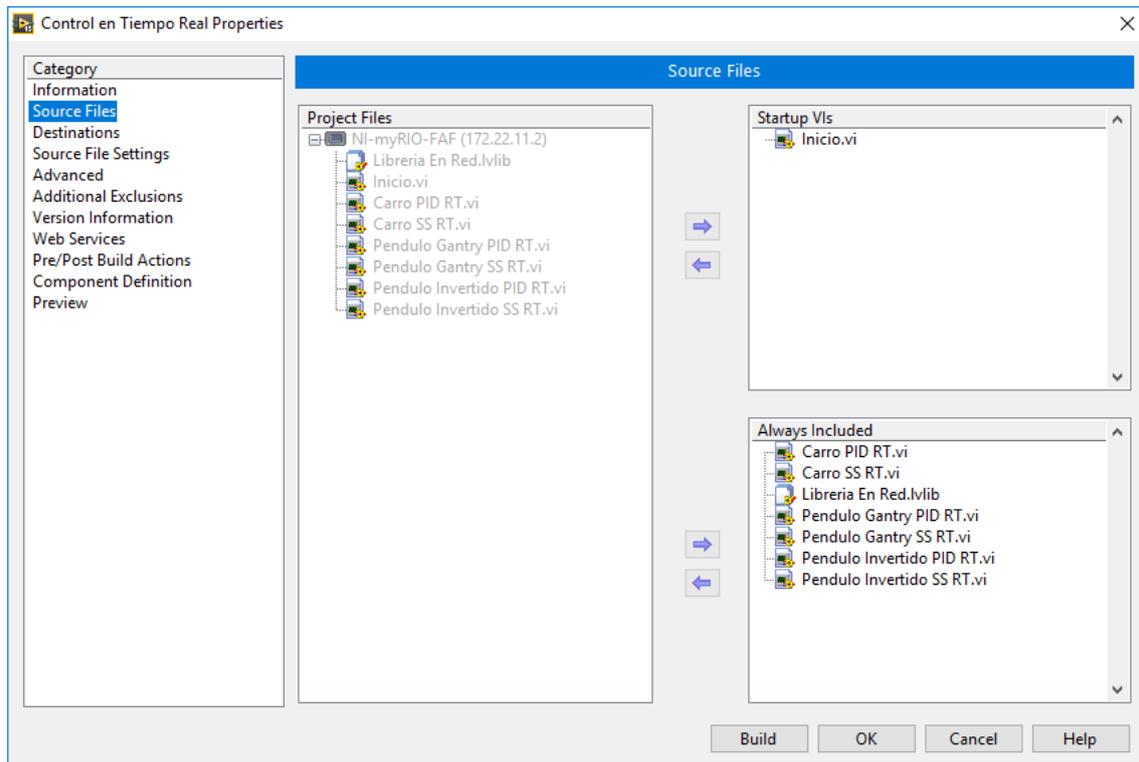


Figura 171. Archivos incluidos en la aplicación en tiempo real

Por último, se selecciona la opción *Build* para construir la aplicación en tiempo real.

La carga se hace desde la ventana de proyecto en LabVIEW, haciendo clic derecho sobre la aplicación ya creada, y seleccionando la opción *Run as startup*, como se muestra en la *Figura 172* a continuación. Tras la carga, aparece un diálogo que indica que es necesario resetear la tarjeta, en el que se pulsa la opción *Yes*.

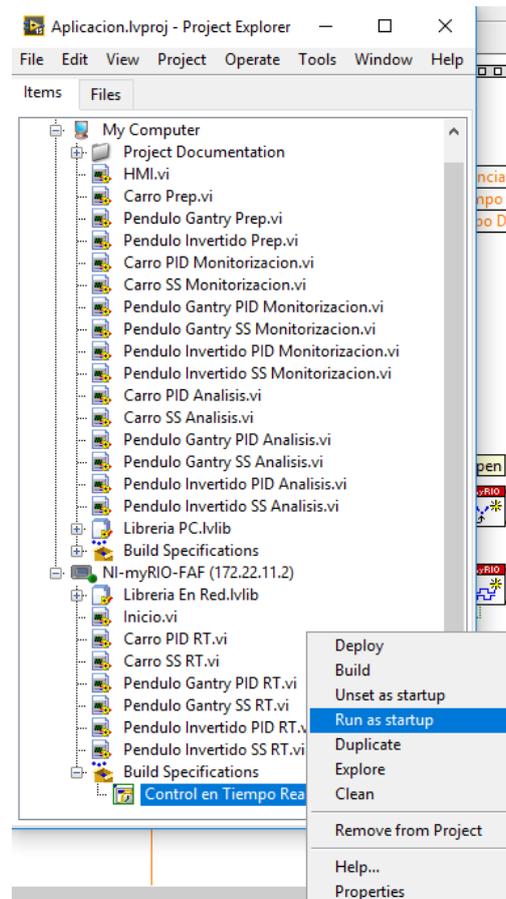


Figura 172. Carga de la aplicación en tiempo real en la tarjeta NI myRIO

Esta aplicación se suministra directamente cargada en la tarjeta Ni myRIO con la que se desarrolla el proyecto. Además, se incluye el archivo de la aplicación en tiempo real por si se desea cargar a otra tarjeta o, en algún momento, se borra el código en la tarjeta de este proyecto. Tras hacer la carga, es necesario resetear la tarjeta, como ya se ha indicado.

A pesar de que la aplicación se puede cargar sin el software de LabVIEW, si es necesario hacerlo, National Instruments recomienda que se haga desde la ventana de proyecto en LabVIEW, como se ha explicado anteriormente. Por esta razón, dentro del producto final también se incluye el proyecto en el que se ha desarrollado el código.

### **6.6.5.2. Aplicación para ordenador**

Como en el caso anterior, se crea una aplicación con el programa que se ejecuta en el ordenador.

En este caso, para construir la aplicación, hay que hacer clic derecho en *Build Specifications* bajo el ordenador de desarrollo y seleccionar una nueva *Application (EXE)*. La Figura 173 siguiente muestra cómo proceder.

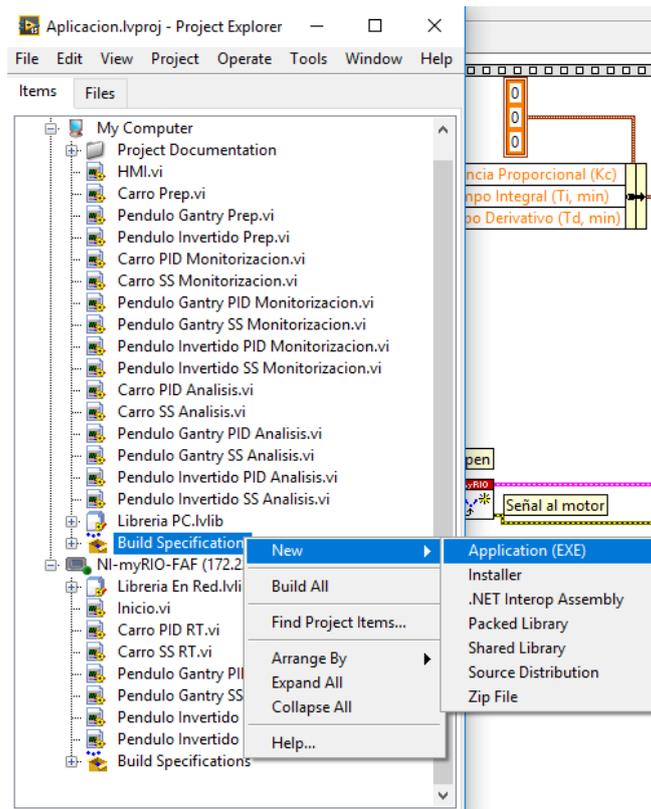


Figura 173. Creación de una nueva aplicación para ordenador

Para empezar, como muestra la *Figura 174*, es necesario seleccionar el nombre de la aplicación, en la casilla en la que pone *Target Filename*, en este caso *Sistema de Analisis de Controladores Avanzados para la Maqueta IP02 de Quanser.exe*. Además, se especifica el directorio donde se guardará la aplicación.

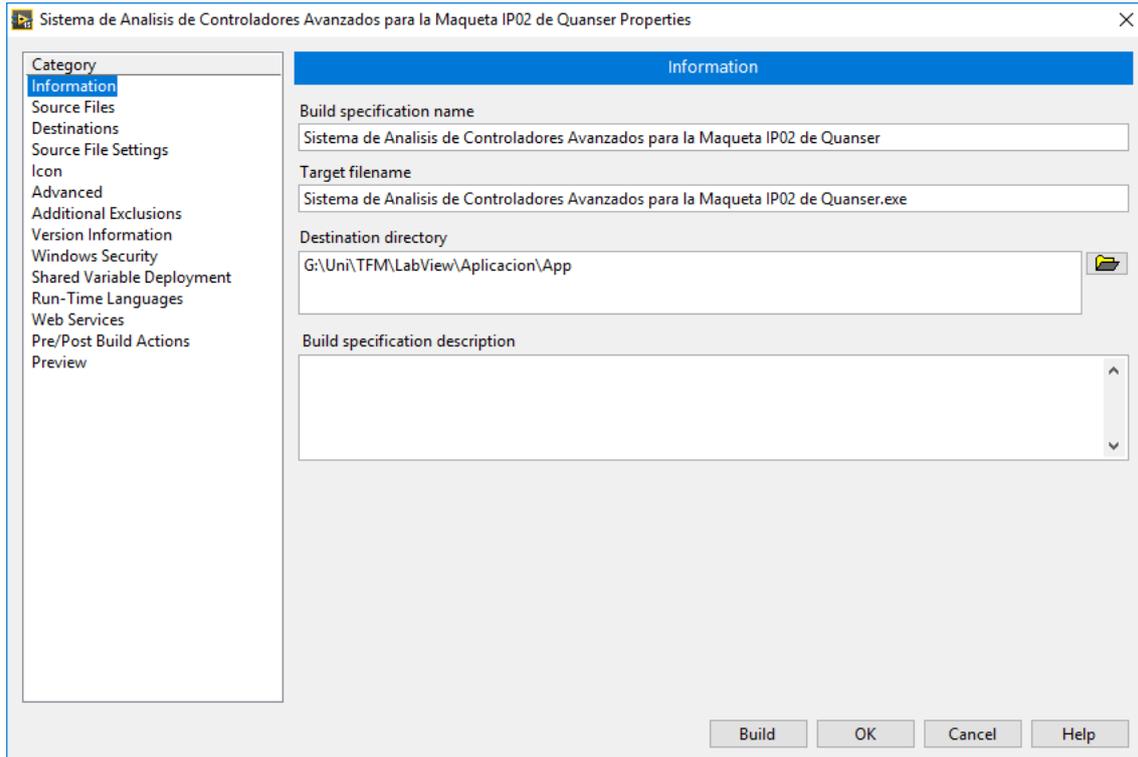


Figura 174. Información de la aplicación para ordenador

De nuevo, en la pestaña *Source Files*, se ubica bajo *Startup VIs* el VI con el que comenzará la ejecución de la aplicación, en este caso “HMI.vi”. El resto, junto con librerías, se ubican bajo *Always Included*.

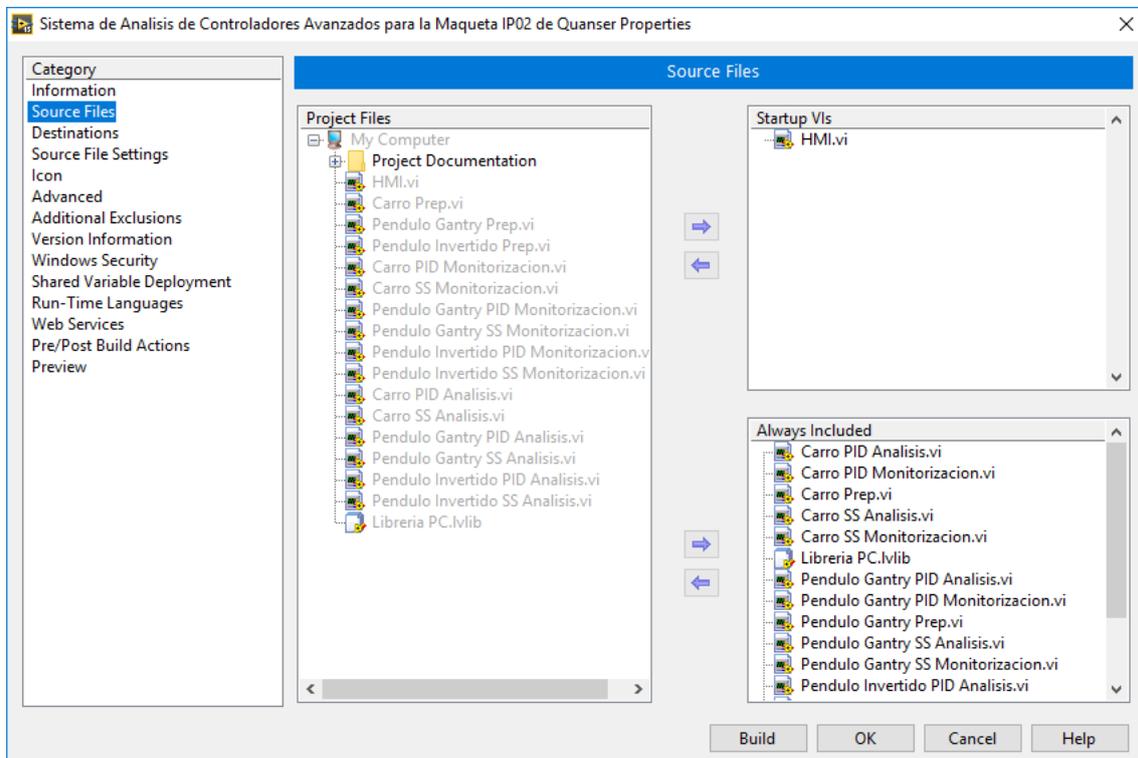


Figura 175. Archivos incluidos en la aplicación para ordenador

Una vez configurados, se hace clic en el botón *Build*, para construir la aplicación, que se guardará en el directorio seleccionado. Una vez construida, se podrá ejecutar desde ahí.

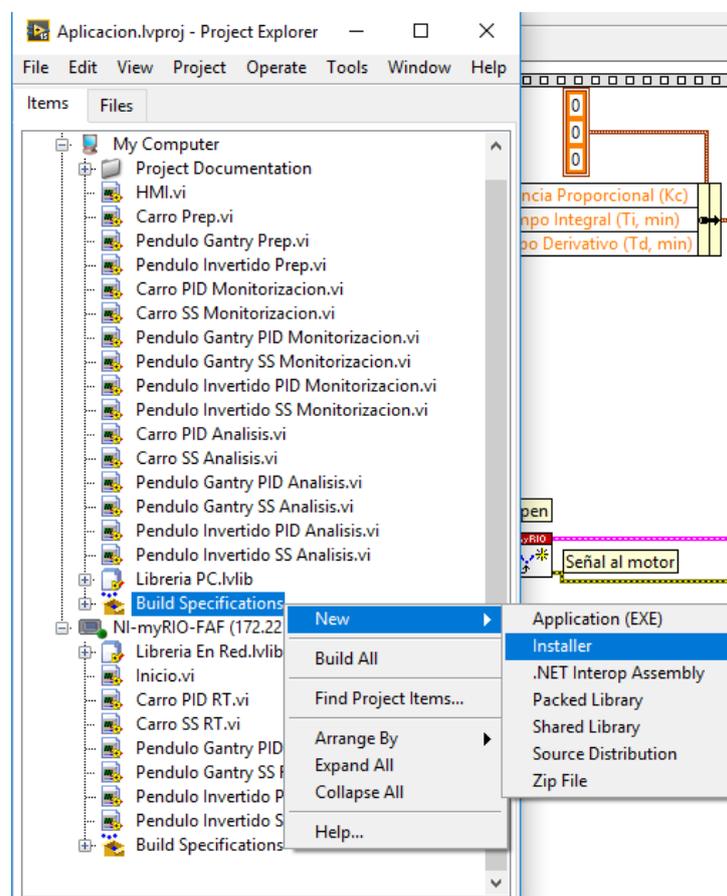
### **6.6.5.3. Instalador de la aplicación para ordenador**

La aplicación para ordenador permite que el programa se ejecute, aunque el proyecto de desarrollo ya no se encuentre en el ordenador. Sin embargo, aún son necesarios ciertos elementos de software de LabVIEW para que ésta se ejecute correctamente. Por lo tanto, si se desea ejecutar la aplicación en un ordenador que no cuenta con LabVIEW instalado, no será posible.

Para solucionar este problema, se crea un instalador de la aplicación, que incluye la instalación de los elementos software necesarios para ejecutar la aplicación en cualquier ordenador y permitir, por consiguiente, su distribución. Los elementos software necesarios son:

- *NI LabVIEW Runtime 2015*, para la correcta ejecución de programas en tiempo real y comunicación con targets que trabajan en tiempo real.
- *NI CompactRIO Runtime 15.0*, para tener instalados los drivers correctos para la comunicación USB de la tarjeta NI myRIO con el ordenador.

Como se muestra en la *Figura 176*, el instalador se crea haciendo clic derecho en *Build Specifications* bajo el ordenador de desarrollo y seleccionando un nuevo *Installer*.



*Figura 176. Creación de un nuevo instalador*

Bajo *Product Name*, se escribe el nombre deseado para el archivo, en este caso *INSTALADOR de Sistema de Analisis de Controladores Avanzados para la Maqueta IP02 de Quanser*, y en *Installer destination*, el directorio donde éste se guardará.

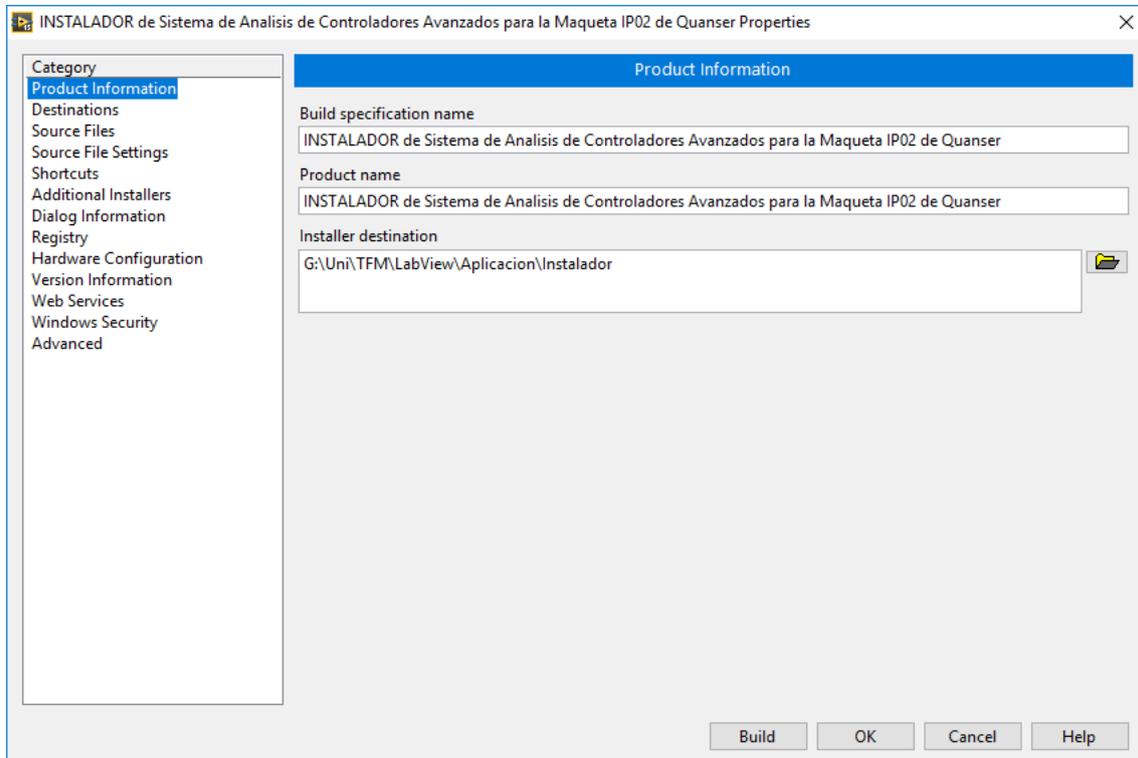


Figura 177. Información del instalador

En *Destinations*, se define el directorio en el que se instalará la aplicación en el ordenador de destino, en este caso en *Program Files > Sistema de Analisis de Controladores Avanzados para la Maqueta IP02 de Quanser*.

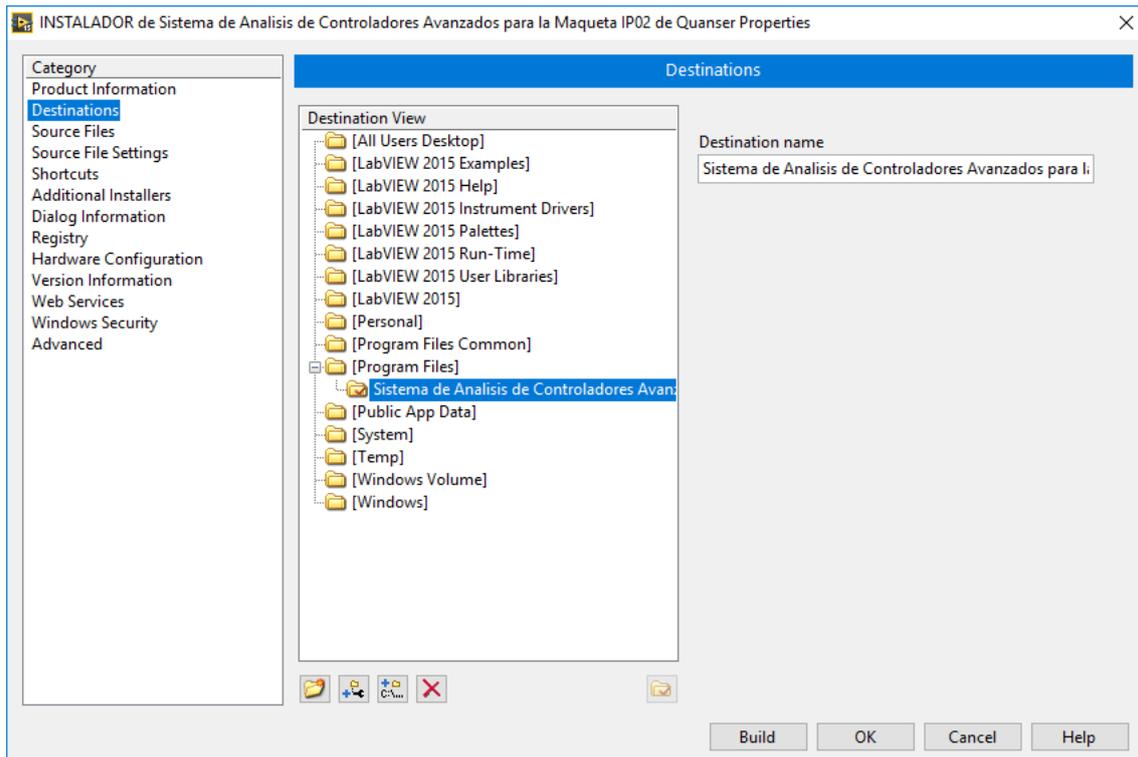


Figura 178. Carpeta de destino del instalador

En la pestaña *Source Files*, se debe incluir la aplicación ya creada, ubicada dentro de *Build Specifications*, que se instalará.

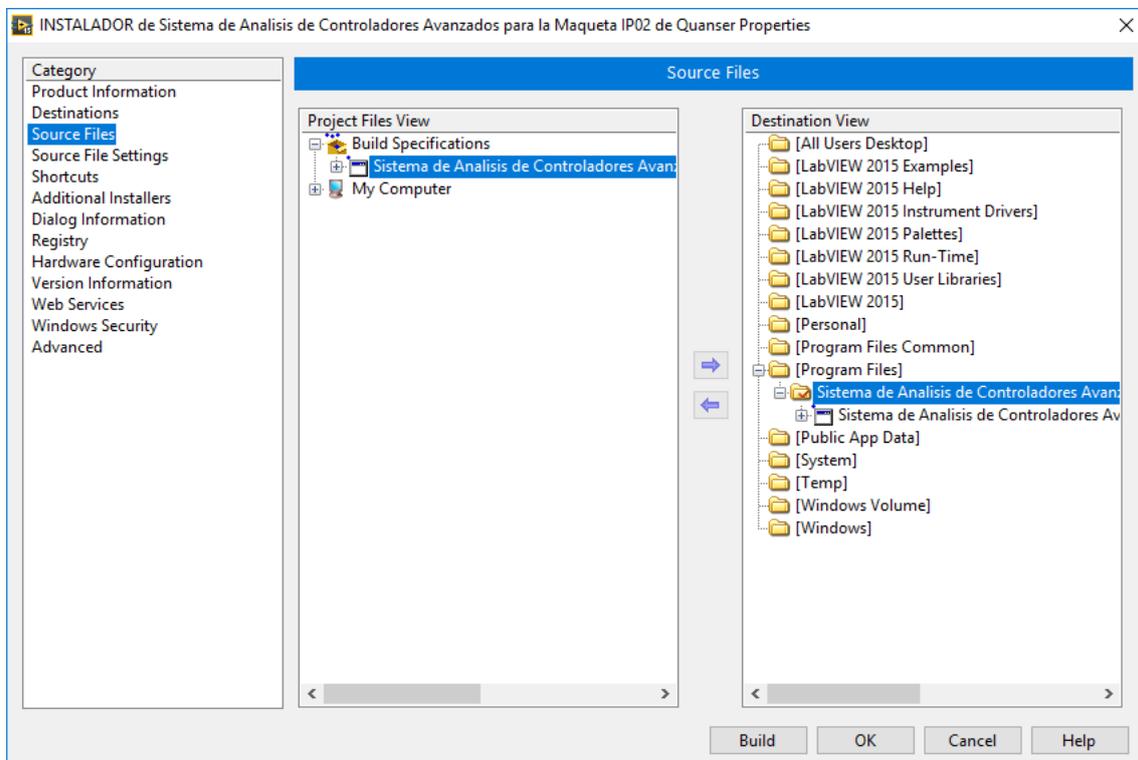


Figura 179. Aplicaciones incluidas en el instalador

En la opción *Shortcuts*, se especifican los accesos directos que se crearán en la instalación, en este caso, el que viene predeterminado en la carpeta de la instalación y otro en el escritorio.

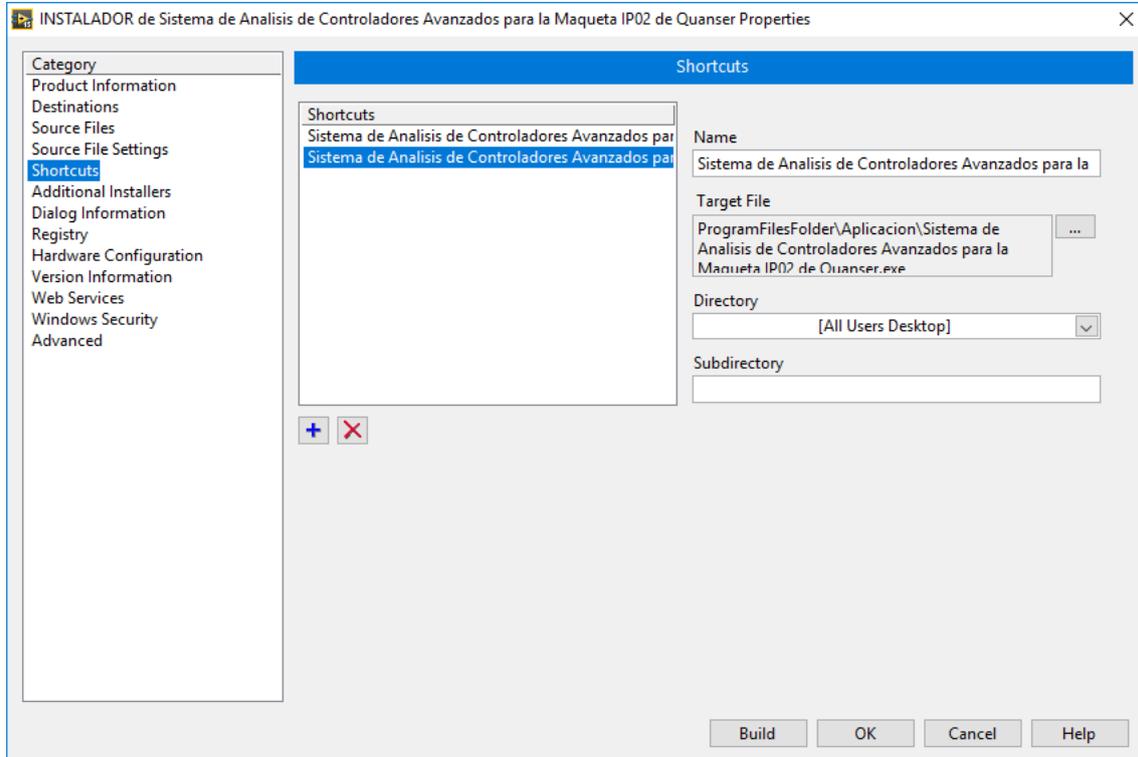


Figura 180. Accesos directos creados en la instalación

Los paquetes software adicionales que incluye el instalador se seleccionan en *Additional Installers*. En este caso, como se ha mencionado antes, se incluyen *NI LabVIEW Runtime 2015* y *NI CompactRIO Runtime 15.0*.

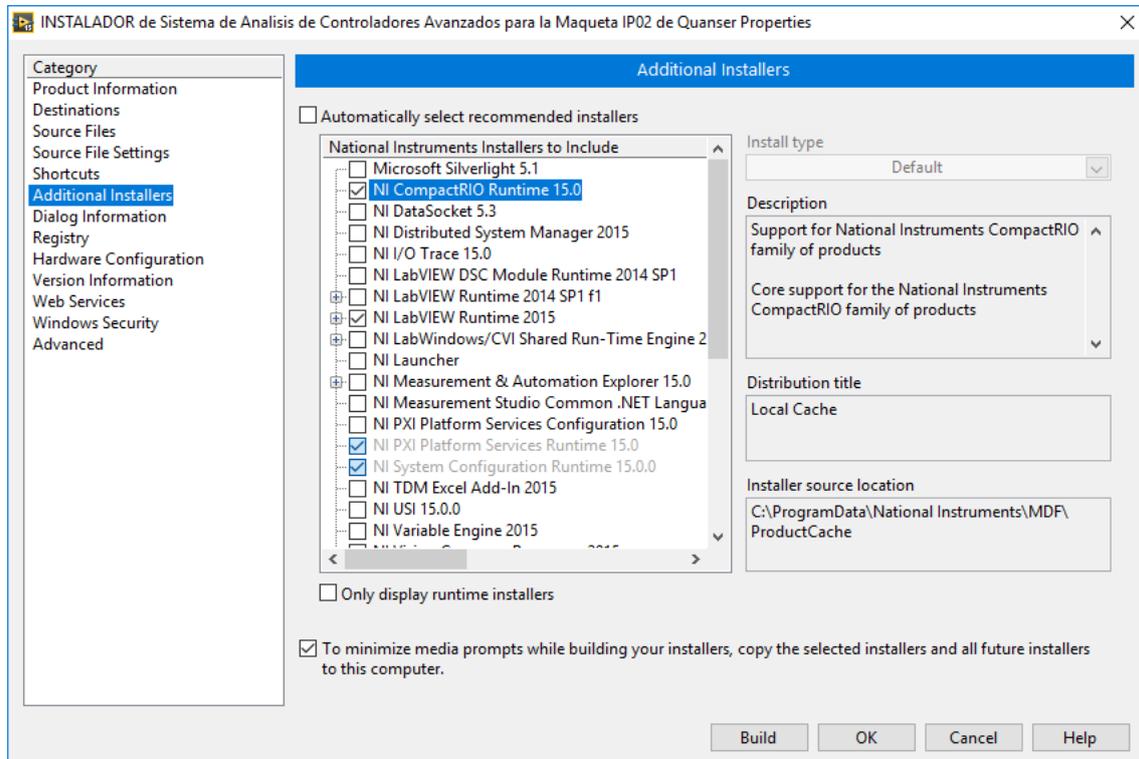


Figura 181. Software adicionales incluidos en la instalación

Por último, en *Advanced*, es importante comprobar que está desmarcada la casilla *Require the LabVIEW 2015 32-bit development system*, para que no sea necesario LabVIEW en el ordenador de destino. Además, se mantiene el apartado *System Requirements* en *Windows XP Service Pack 3 or later*, para que sea compatible con las versiones de Windows XP o mayores.

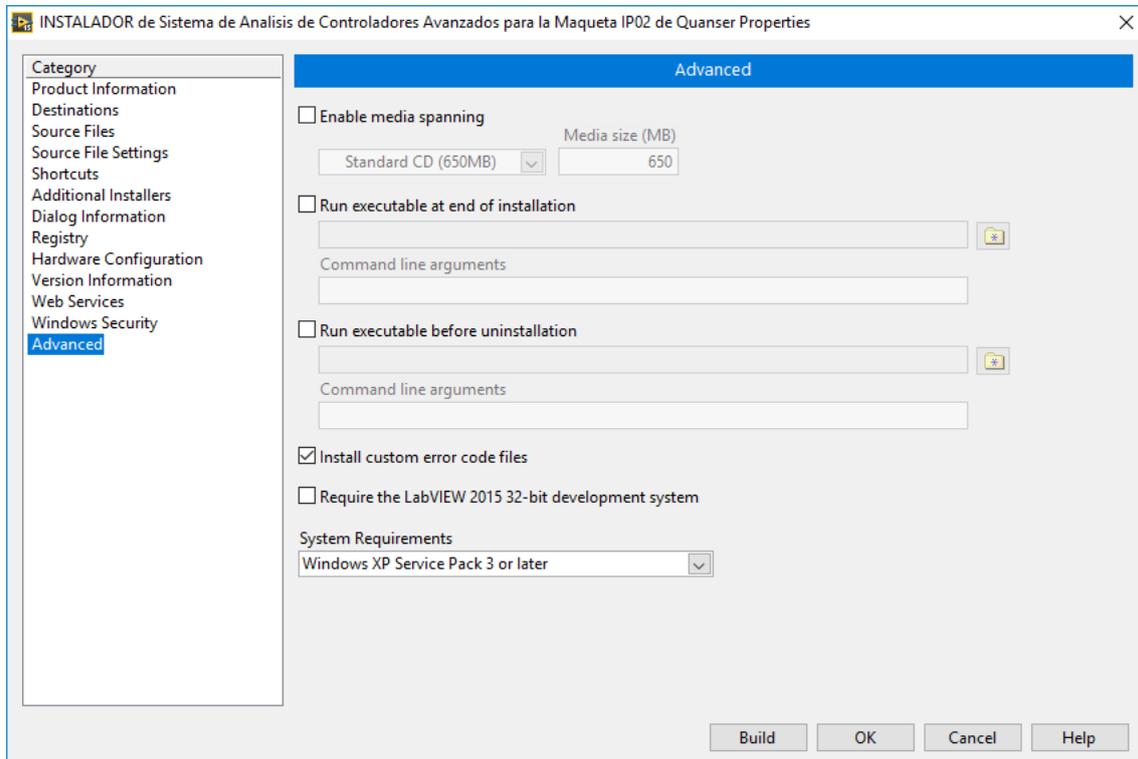


Figura 182. Requerimientos del sistema de destino

Una vez configurado el instalador, se hace clic en el botón *Build*, con lo que se generará el instalador en el directorio elegido, y podrá ser lanzado con el archivo *setup.exe*, contenido en la carpeta *Volume* del directorio. Para llevar a cabo la compilación, es necesario introducir en la bandeja de disco del ordenador el DVD *LABVIEW 2015 MYRIO SOFTWARE BUNDLE, DVD 1* que se suministra con la tarjeta, para que se pueda añadir el software *NI CompactRIO Runtime 15.0* al paquete de instalación.

En la *Figura 183* inferior, se muestra el aspecto final del proyecto, con sus correspondientes *Build Specifications*.

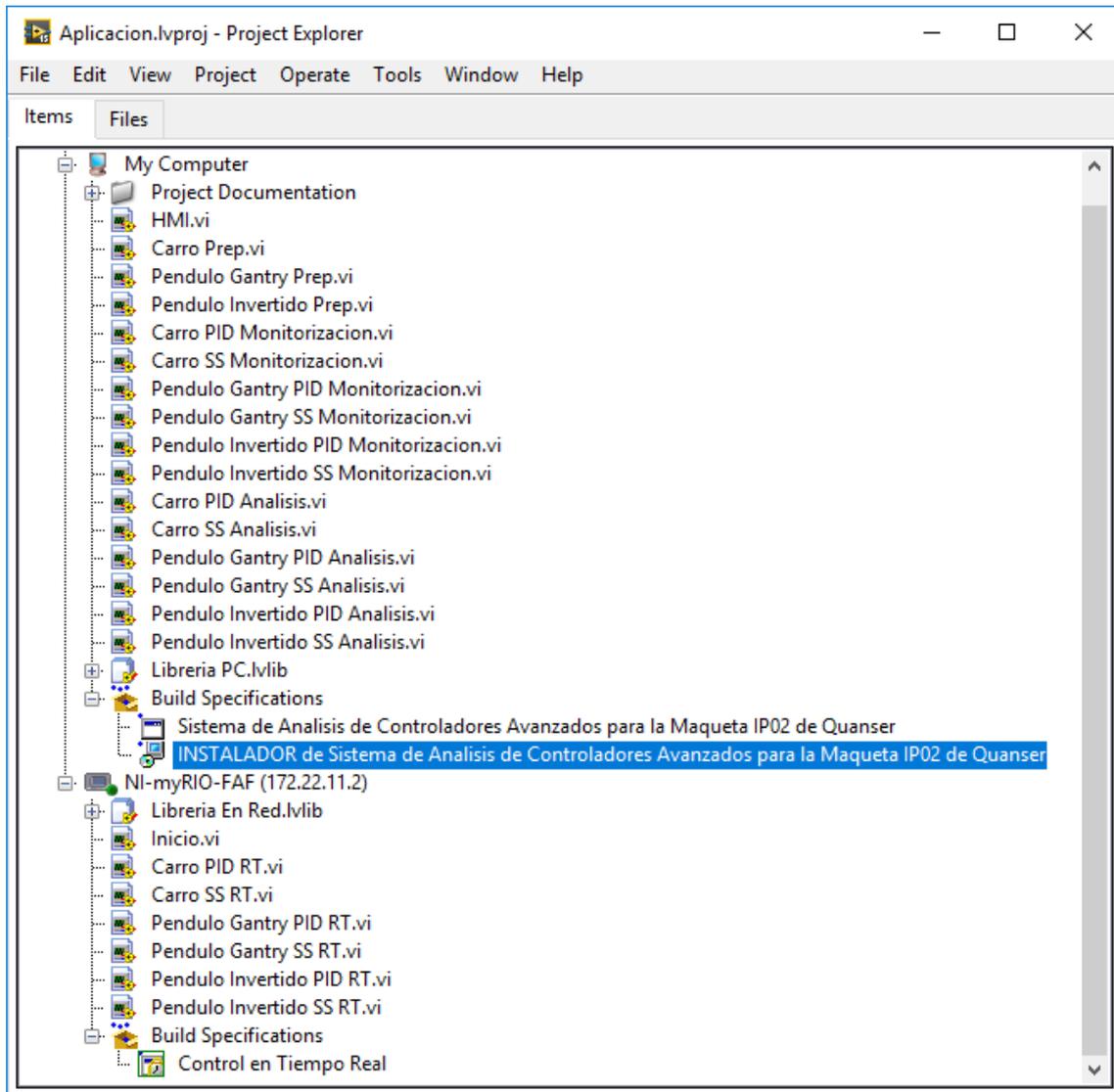


Figura 183. Aspecto final de la ventana de proyecto

## **7. PLANIFICACIÓN**

En este apartado se presenta el plan de proyecto seguido.

El proyecto comienza el lunes 12 de febrero de 2018 y finaliza el jueves 20 de septiembre de 2018. La planificación descrita más abajo se basa en un calendario laboral de lunes a viernes de 6 horas diarias, excepto unas pocas jornadas concretas que son de 9 horas. Se ha excluido del grupo de días laborales un total de 28 días de vacaciones.

A continuación, se muestra la lista de tareas del proyecto.

## DESARROLLO DE CONTROLADORES AVANZADOS PARA SISTEMAS MULTI VARIABLES EN EL ENTORNO LABVIEW

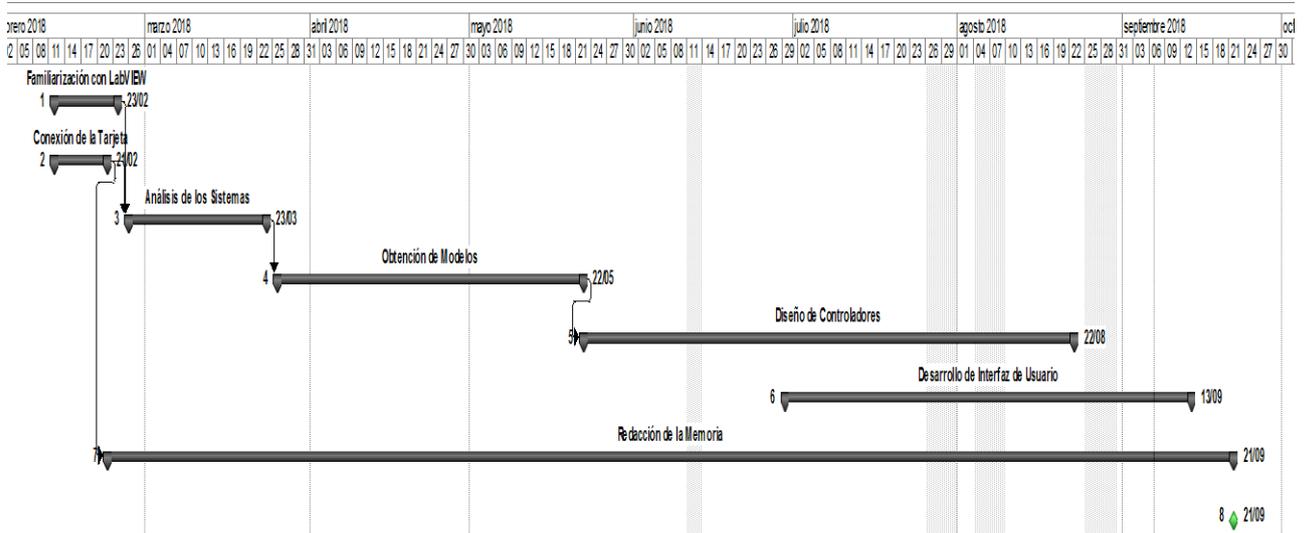
Id	EDT	Nombre de tarea	Duración	Comienzo	Fin	Predecesoras
1	1	<b>Familiarización con LabVIEW</b>	10 días	lun 12/02/18	vie 23/02/18	
2	1.1	Búsqueda de Información	3 días	lun 12/02/18	mié 14/02/18	
3	1.2	Lectura sobre Conceptos Básicos	3 días	lun 12/02/18	mié 14/02/18	
4	1.3	Solución de Ejercicios	7 días	jue 15/02/18	vie 23/02/18	3;2
5	2	<b>Conexión de la Tarjeta</b>	8 días	lun 12/02/18	mié 21/02/18	
6	2.1	Búsqueda de Información	5 días	lun 12/02/18	vie 16/02/18	
7	2.2	Realización de las Conexiones	2 días	lun 19/02/18	mar 20/02/18	6
8	2.3	Comprobación de Funcionamiento	1 día	mié 21/02/18	mié 21/02/18	7
9	3	<b>Análisis de los Sistemas</b>	20 días	lun 26/02/18	vie 23/03/18	1;5
10	3.1	Búsqueda de Información	15 días	lun 26/02/18	vie 16/03/18	
11	3.2	Mediciones	5 días	lun 19/03/18	vie 23/03/18	10
12	4	<b>Obtención de Modelos</b>	40 días	lun 26/03/18	mar 22/05/18	9
13	4.1	Desarrollo de Modelos Matemáticos	15 días	lun 26/03/18	vie 13/04/18	
14	4.2	Construcción de los Modelos en LabVIEW	10 días	lun 16/04/18	mar 01/05/18	13
15	4.3	Validación de los Modelos	15 días	mar 01/05/18	mar 22/05/18	14
16	4.4	<b>Modelos Obtenidos</b>	0 días	mar 22/05/18	mar 22/05/18	15
17	5	<b>Diseño de Controladores</b>	55 días	mar 22/05/18	mié 22/08/18	12
18	5.1	Selección de Tipos de Controlador	10 días	mar 22/05/18	mar 05/06/18	
19	5.2	Construcción de los Controladores en LabVIEW	15 días	mar 05/06/18	vie 29/06/18	18
20	5.3	Selección de los Parámetros de Control	30 días	vie 29/06/18	mié 22/08/18	19
21	5.4	Validación de los Controladores	30 días	vie 29/06/18	mié 22/08/18	19
22	5.5	<b>Controladores Obtenidos</b>	0 días	mié 22/08/18	mié 22/08/18	21
23	6	<b>Desarrollo de Interfaz de Usuario</b>	40 días	vie 29/06/18	jue 13/09/18	19
24	6.1	Selección de la Arquitectura de la Interfaz de Usuario	15 días	vie 29/06/18	jue 19/07/18	
25	6.2	Elaboración de la Interfaz de Usuario	20 días	vie 20/07/18	jue 06/09/18	24
26	6.3	Comprobación de Funcionamiento	25 días	vie 20/07/18	jue 13/09/18	24
27	6.4	<b>Interfaz de Usuario Terminada</b>	0 días	jue 13/09/18	jue 13/09/18	26
28	7	<b>Redacción de la Memoria</b>	133 días	jue 22/02/18	vie 21/09/18	8;5
29	7.1	Redacción de Apartados Generales de la Memoria	13,5 días	jue 22/02/18	lun 17/09/18	
30	7.2	Redacción sobre Conexión de la Tarjeta	4 días	jue 22/02/18	jue 01/03/18	
31	7.3	Redacción sobre los Sistemas de Estudio	7 días	lun 19/03/18	vie 30/03/18	30;10
32	7.4	Redacción sobre los Modelos	2 días	lun 16/04/18	jue 03/05/18	31;13
33	7.5	Redacción sobre los Controladores	6 días	mar 05/06/18	jue 05/07/18	32;18
34	7.6	Redacción sobre la Interfaz de Usuario	3 días	vie 14/09/18	mar 18/09/18	33;26
35	7.7	Redacción del Manual de Usuario para la Interfaz de Usuario	3 días	mié 19/09/18	vie 21/09/18	34;26
36	7.8	<b>Documento Finalizado</b>	0 días	vie 21/09/18	vie 21/09/18	35;29;30;31;32;33;3
37	8	<b>Fin de Proyecto</b>	0 días	vie 21/09/18	vie 21/09/18	27;36

Tabla 10. Lista de tareas del proyecto

Como muestra la *Tabla 10* en color azul, el proyecto se divide en siete tareas principales:

- 1. Familiarización con LabVIEW
- 2. Conexión de la Tarjeta
- 3. Análisis de los Sistemas
- 4. Obtención de Modelos
- 5. Diseño de Controladores
- 6. Desarrollo de Interfaz de Usuario
- 7. Redacción de la Memoria

En el diagrama de Gantt que se muestra en la *Figura 184*, se puede apreciar mejor la dependencia entre las tareas principales mencionadas. El proyecto comienza con las tareas de *Familiarización con LabVIEW* y *Conexión de la Tarjeta*, que empiezan simultáneamente y se desarrollan en paralelo. Una vez finalizadas ambas, se pone en marcha la tarea de *Análisis de los Sistemas*, que, cuando finaliza, da comienzo a la de *Obtención de Modelos*. Una vez concluida esta última, se inicia la tarea de *Diseño de Controladores*. Cuando la subtarea *Construcción de los Controladores en LabVIEW* llega a su fin, se inicia la tarea de *Desarrollo de Interfaz de Usuario*, que se realiza en paralelo con la tarea de *Diseño de Controladores* hasta que ésta última termina. Por último, existe una tarea de *Redacción de la Memoria*, que comienza tras la conclusión de la tarea de *Conexión de la Tarjeta* y se lleva a cabo simultáneamente con el resto de tareas hasta la conclusión del proyecto, que sucede al finalizar esta tarea.



*Figura 184. Diagrama de Gantt de las tareas principales del proyecto*

En el diagrama de Gantt completo de la *Figura 185*, en rojo se pueden ver las tareas críticas del proyecto, y el camino crítico que conforman. Éstas son las tareas cuyo retraso implica un retraso del fin del proyecto. Este camino está conformado por las tareas *Familiarización con LabVIEW*, *Análisis de los Sistemas* y *Obtención de Modelos* en su totalidad. La tarea de *Conexión de la Tarjeta* no es crítica, ya que desarrollándose en paralelo con *Familiarización con LabVIEW* dura menos que ésta última.

En el caso de *Diseño de Controladores* y *Desarrollo de Interfaz de Usuario*, al ocurrir simultáneamente, sólo es crítica parte de la tarea, ya que es el final de la subtarea

*Construcción de los Controladores en LabVIEW* la que da inicio a *Desarrollo de Interfaz de Usuario*. Además, en *Desarrollo de Interfaz de Usuario*, las subtareas *Elaboración de la Interfaz de Usuario* y *Comprobación de Funcionamiento* también se realizan simultáneamente, aunque esta última finaliza cinco días después de que termine la anterior, retrasándose el final de la segunda si se retrasa el de la primera, por lo que ambas son críticas. Por último, se puede observar que la redacción de la memoria, excepto en su fase final, una vez finalizada la interfaz de usuario, no es crítica.

Aunque la *Validación de los Modelos* se realiza en solitario, esta subtarea incluye sucesivas pruebas y correcciones en el modelo. La *Validación de los Controladores*, sin embargo, se realiza simultáneamente con la elección de parámetros, ya que, en este caso, se realizan sucesivas elecciones de parámetros y pruebas.

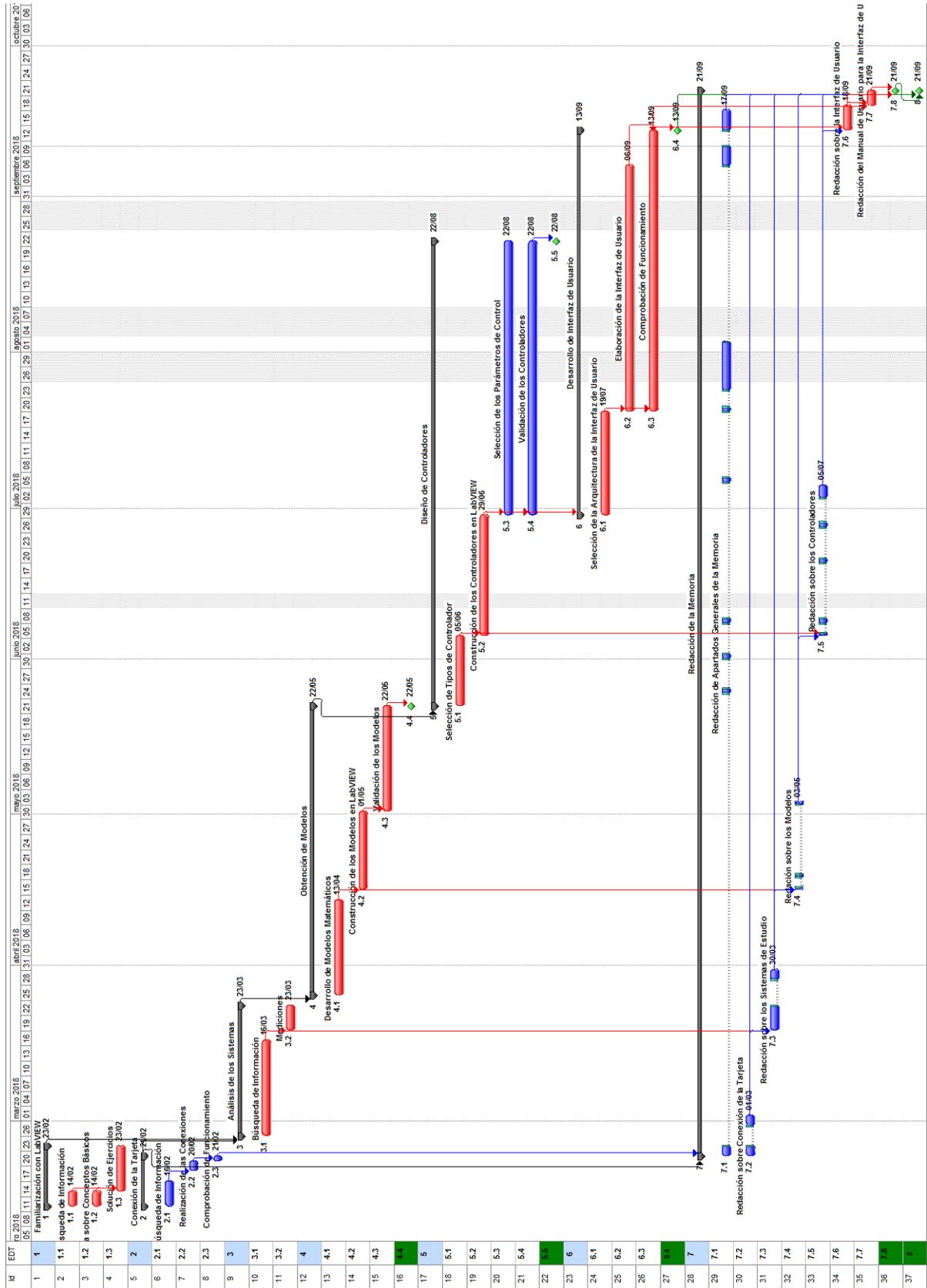


Figura 185. Diagrama de Gantt de todas las tareas del proyecto

Así mismo, en la *Tabla 10* anterior, se muestran los hitos del proyecto en color verde, que son los siguientes:

- 4.4. Modelos Obtenidos
- 5.5. Controladores Obtenidos
- 6.4. Interfaz de Usuario Terminada
- 7.8. Documento Finalizado
- 8. Fin de Proyecto

Los hitos del proyecto se corresponden con la finalización de las tareas que dan un resultado tangible como consecuencia de las mismas, como es el caso de los modelos, los controladores, la interfaz de usuario y la memoria del proyecto. Además, existe un hito final de *Fin de Proyecto*, que únicamente sirve para marcar el fin del mismo.

En la *Tabla 11*, que se muestra a continuación, se puede observar la carga de trabajo de cada una de las tareas principales del proyecto. Sumando las horas de duración de cada tarea, puede verse que la carga de trabajo total del proyecto ha sido de 853 horas, bastante superior a las 600 horas que se estiman para un trabajo de fin de máster, pero que se justifican por el desconocimiento inicial del software LabVIEW que, aunque se ha mitigado con la primera tarea del proyecto, ha influido en el resto de fases, en las que la complejidad del uso del software ha ido en aumento y, por lo tanto, ha habido que utilizar conceptos que no estaban contemplados en la primera tarea del proyecto.

Id	EDT	Nombre de tarea	Trabajo	Duración	Comienzo	Fin
1	1	Familiarización con LabVIEW	33 horas	10 días	lun 12/02/18	vie 23/02/18
5	2	Conexión de la Tarjeta	23 horas	8 días	lun 12/02/18	mié 21/02/18
9	3	Análisis de los Sistemas	105 horas	20 días	lun 26/02/18	vie 23/03/18
12	4	Obtención de Modelos	229 horas	40 días	lun 26/03/18	mar 22/05/18
17	5	Diseño de Controladores	218 horas	55 días	mar 22/05/18	mié 22/08/18
23	6	Desarrollo de Interfaz de Usuario	144 horas	40 días	vie 29/06/18	jue 13/09/18
28	7	Redacción de la Memoria	101 horas	133 días	jue 22/02/18	vie 21/09/18
37	8	Fin de Proyecto	0 horas	0 días	vie 21/09/18	vie 21/09/18

*Tabla 11. Carga de trabajo de las tareas principales del proyecto*

Por último, se observa que las tareas que mayor tiempo han requerido han sido las de *Obtención de Modelos* y *Diseño de Controladores*, con más de 200 horas cada una. Esto se debe en parte a la necesidad de validar tanto modelos como controladores, en un proceso iterativo de pruebas y correcciones hasta dar con el resultado deseado, que ha requerido mucho tiempo.

## 8. DESCARGO DE GASTOS

En este apartado se detallan los gastos necesarios para llevar a cabo el proyecto y su justificación.

Primeramente, en la *Tabla 12* siguiente, se definen la mano de obra y materiales utilizados para este trabajo, así como sus cargas horarias y cantidades.

Carga Horaria			
Nº Ref.	Concepto	Cantidad	
<b>1.</b>	<b>MANO DE OBRA</b>	<b>Tiempo [h]</b>	
1.1.	Horas de director de proyecto	80	
1.2.	Horas de ingeniero industrial	853	
1.3.	Horas de técnico	20	
<b>2.</b>	<b>MATERIALES</b>	<b>Unidades</b>	<b>Uso [h/ud]</b>
2.1.	Ordenador personal completo	1	853
2.2.	Tarjeta NI myRIO con licencias de software LabVIEW 2015, myRIO Toolkit 2015 y LabVIEW Runtime 2015	1	700
2.3.	Maqueta IP02 de Quanser Inc. y accesorios	1	500
2.4.	Licencia de software de Matlab 2013b	1	100
2.5.	Conectores y cables	1	500

*Tabla 12. Lista de mano de obra y materiales con sus cargas horarias y cantidades*

En la mano de obra, además de las horas del tutor del proyecto y el alumno, se incluyen las del técnico de laboratorio que realizó, guiado por las especificaciones dadas por el alumno, el trabajo de adaptación de los cables para la conexión de la tarjeta.

En cuanto a los materiales, además de un ordenador personal completo, el set de National Instruments que incluye la tarjeta NI myRIO y las pertinentes licencias de software, la maqueta IP02 y los citados conectores y cables, se incluye la licencia del software Matlab, que se utiliza para hacer determinados cálculos de una forma más sencilla que en LabVIEW.

Posteriormente, en la *Tabla 13* siguiente, se detallan los precios unitarios de los conceptos mencionados anteriormente, así como la vida útil y el uso anual en el caso de los materiales, utilizados para calcular las amortizaciones. Se puede apreciar que el ordenador y el software Matlab se utilizan durante muchas horas a lo largo del año, no únicamente en este proyecto, mientras que el uso de la tarjeta y la maqueta fuera del proyecto es bastante menor, y en el caso de los cables nulo.

Precios Unitarios				
Nº Ref.	Concepto	Precio Unitario		
<b>1.</b>	<b>MANO DE OBRA</b>	<b>Precio [€/h]</b>		
1.1.	Horas de director de proyecto	60,00		
1.2.	Horas de ingeniero industrial	40,00		
1.3.	Horas de técnico	20,00		
<b>2.</b>	<b>MATERIALES</b>	<b>Precio [€/ud]</b>	<b>Vida [años]</b>	<b>Uso [h/año]</b>
2.1.	Ordenador personal completo	900,00	10	1500
2.2.	Tarjeta NI myRIO con licencias de software LabVIEW 2015, myRIO Toolkit 2015 y LabVIEW Runtime 2015	600,00	5	600
2.3.	Maqueta IP02 de Quanser Inc. y accesorios	6000,00	15	600
2.4.	Licencia de software de Matlab 2013b	2000,00	5	1200
2.5.	Conectores y cables	40,00	10	600

*Tabla 13. Precios unitarios, vida útil y uso anual de los conceptos*

Los materiales mencionados, ya que han sido utilizados para clases, trabajos y proyectos anteriores, y se volverán a utilizar en otros venideros, se incluyen en amortizaciones.

Con la información dada, es fácil calcular las tasas de amortización de los distintos bienes, mediante la división de su precio unitario entre el número de horas de uso a lo largo de su vida útil.

Tasas de Amortización		
Nº Ref.	Concepto	Tasa de amortización [€/h]
<b>2.</b>	<b>MATERIALES</b>	
2.1.	Ordenador personal completo	0,0600
2.2.	Tarjeta NI myRIO con licencias de software LabVIEW 2015, myRIO Toolkit 2015 y LabVIEW Runtime 2015	0,2000
2.3.	Maqueta IP02 de Quanser Inc. y accesorios	0,6667
2.4.	Licencia de software de Matlab 2013b	0,3333
2.5.	Conectores y cables	0,0067

*Tabla 14. Tasas de amortización de los materiales*

En el apartado de gastos se incluyen los costes de impresión de documentación para la realización del proyecto. También se tienen en cuenta los desplazamientos desde Vitoria, lugar de residencia del alumno, al laboratorio, estimándose el coste en 120€ mensuales durante 8 meses.

A continuación, se presenta el descargo de gastos completo del proyecto.

CONCEPTO	Nº UNIDADES	UNIDAD	COSTE UNITARIO		COSTE TOTAL
<b>HORAS INTERNAS</b>					<b>39.320,00 €</b>
Director de Proyecto	80	h	60,00	€/h	4.800,00 €
Ingeniero Industrial	853	h	40,00	€/h	34.120,00 €
Técnico	20	h	20,00	€/h	400,00 €
<b>AMORTIZACIONES</b>					<b>557,85 €</b>
Ordenador Personal	853	h	0,0600	€/h	51,18 €
Tarjeta NI myRIO y software asociado	700	h	0,2000	€/h	140,00 €
Maqueta IPO2 de Quanser Inc. y accesorios	500	h	0,6667	€/h	333,33 €
Licencia de Matlab	100	h	0,3333	€/h	33,33 €
Conectores y cables	500	h	0,0067	€/h	3,33 €
<b>GASTOS</b>					<b>980,00 €</b>
Impresión de Documentos					20,00 €
Desplazamiento al Laboratorio	8	meses	120	€/mes	960,00 €
<b>SUBCONTRATAIONES</b>					<b>0,00 €</b>
<b>SUBTOTAL 1</b>					<b>40.857,85 €</b>
Costes Indirectos				4%	1.634,31 €
<b>SUBTOTAL 2</b>					<b>42.492,16 €</b>
Costes Financiero				4%	1.699,69 €
<b>TOTAL</b>					<b>44.191,85 €</b>

Tabla 15. Descargo de gastos del proyecto

Los costes indirectos representan los gastos en conceptos no directamente relacionados con el proyecto, pero que han de tenerse en cuenta. Entre ellos se encuentran la electricidad utilizada, tanto para iluminación como para el funcionamiento de los equipos, la conexión a internet, la limpieza del laboratorio, el programa antivirus y mantenimiento del ordenador y la climatización del laboratorio, por citar algunos.

Por último, se tienen en cuenta los costes financieros que se generan del préstamo que se solicita para llevar a cabo el proyecto.

El coste total del proyecto asciende a **44.191,85€**. Cabe destacar que el coste de la mano de obra supera notablemente al resto de partidas, ya que los equipos y licencias a utilizar ya están en posesión de la universidad, y sus tasas de amortización son bajas por su largo tiempo de vida y de uso. Además, la gran cantidad de horas de ingeniero industrial, es decir, trabajadas por el alumno, hacen que este concepto suponga más del 75% del coste total.

## **9. CONCLUSIONES FINALES**

Como conclusiones finales del proyecto, se puede decir lo siguiente:

- La elección del presente proyecto se ha hecho con el objetivo de crear una herramienta de experimentación en técnicas de control automático sobre un sistema real (maqueta IP02 de Quanser).
- La configuración de péndulo invertido es el paradigma de la dificultad de control, ya que se trata de un sistema complejo, altamente inestable, multivariable, no lineal e incluso no invariante en el tiempo, como se ha mencionado.
- Con la conexión adaptada para la tarjeta NI myRIO, dicha maqueta podrá seguir usándose en el futuro a pesar de que la tarjeta de interfaz original de Quanser ya no es compatible con los ordenadores modernos.

En lo que a control se refiere se ha llegado a las siguientes conclusiones:

- De los tres sistemas objeto del proyecto, el péndulo invertido, por su naturaleza inestable, es el más difícil de controlar, siendo el péndulo corto el que presenta más dificultades de control.
- El algoritmo PID tiene un peor comportamiento que el resto de controladores debido a sus limitaciones en el manejo de variables, ya que no utiliza todas las variables de estado como sí lo hacen los controladores de realimentación de estados.
- El algoritmo PID parece, así mismo, menos robusto.
- En relación a los controladores de realimentación de estados estudiados, LQR y asignación de polos, el primero tiene un comportamiento ligeramente mejor.
- Se ha observado que el piñón motriz sufre un rápido desgaste con el uso, lo cual hace que los sistemas en estudio no sean invariantes en el tiempo, requiriendo reajustes de parámetros de control a medida que el piñón se desgasta. El controlador de autoerección es especialmente sensible a este fenómeno.

En cuanto a la aplicación creada:

- Permite hacer pruebas sobre el sistema real y simulaciones sobre el modelo matemático de los sistemas carro, péndulo Gantry y péndulo invertido.
- Permite ver los cambios en el modelo matemático modificando los parámetros del mismo.
- Permite comparar tres tipos de algoritmo de control: PID, realimentación del estado por asignación de polos y LQR.

- Se suministra un instalador que permite que la aplicación se ejecute en un ordenador sin el software LabVIEW.

En lo relativo a la planificación y recursos:

- Aunque el necesario proceso de aprendizaje del manejo de la herramienta LabVIEW ha incrementado la duración del proyecto, ha supuesto un notable beneficio personal para el alumno.
- El proyecto comenzó el 12 de febrero de 2018 y finalizó el jueves 20 de septiembre de 2018, suponiendo 853 horas de trabajo para el alumno.
- El coste total del proyecto ha sido de **44.191,85€**, suponiendo las horas trabajadas por el alumno más del 75% del mismo.

Por último, en los anexos a este documento se incluye un manual de usuario para facilitar el uso de la aplicación creada en este proyecto.

## **10. BIBLIOGRAFÍA**

- [1] National Instruments, «Fundamentos del Entorno de LabVIEW» [En línea]. Disponible en: <http://www.ni.com/getting-started/labview-basics/esa/environment> [Consultado última vez: 4-Abr-2018].
- [2] National Instruments, «Herramientas Comunes en LabVIEW» [En línea]. Disponible en: <http://www.ni.com/getting-started/labview-basics/esa/tools> [Consultado última vez: 4-Abr-2018].
- [3] National Instruments, «Estructuras de Ejecución en LabVIEW» [En línea]. Disponible en: <http://www.ni.com/getting-started/labview-basics/esa/execution-structures> [Consultado última vez: 4-Abr-2018].
- [4] National Instruments, «Estructuras de Datos en LabVIEW» [En línea]. Disponible en: <http://www.ni.com/getting-started/labview-basics/esa/data-structures> [Consultado última vez: 4-Abr-2018].
- [5] National Instruments, «Pasar Datos entre Iteraciones de Ciclo en LabVIEW» [En línea]. Disponible en: <http://www.ni.com/getting-started/labview-basics/esa/shift-registers> [Consultado última vez: 4-Abr-2018].
- [6] National Instruments, «Diseñar Interfaces de Usuario Personalizadas en NI LabVIEW con su NI USB-TC01» [En línea], 18-Mar-2010. Disponible en: <http://www.ni.com/tutorial/10728/es/#toc1> [Consultado última vez: 4-Abr-2018].
- [7] Quanser, «IP01 and IP02 User Manual», Año 2008.
- [8] National Instruments, «NI myRIO-1900 USER GUIDE AND SPECIFICATIONS», Año 2013.
- [9] National Instruments, «NI myRIO Workshop», Versión 14.04.
- [10] National Instruments, «Soporte de Software para NI myRIO» [En línea], 7-Nov-2013. Disponible en: <http://www.ni.com/product-documentation/14603/es/> [Consultado última vez: 4-Abr-2018].
- [11] US Digital, «E2 Optical Kit Encoder», 22-Ago-2006.
- [12] Quanser, «Universal Power Module 1503, 1506, 2405, 2405-PWM User Manual», Año 2006.

- [13] Quanser, «Quanser Accessories: UPM-15-03, UPM-24-05, and UPM-180-25B», Año 2006.
  
- [14] Ricardo Fernández del Busto y Ezeta, «Análisis y Diseño de Sistemas de Control Digital», 21-May-2013.
  
- [15] National Instruments, <<Use Shared Variables Effectively (Real-Time Module)>> [En línea], Marzo de 2017. Disponible en: [http://zone.ni.com/reference/en-XX/help/370622R-01/lvrtbestpractices/rt\\_bp\\_svars/](http://zone.ni.com/reference/en-XX/help/370622R-01/lvrtbestpractices/rt_bp_svars/) [Consultado última vez: 17-Jul-2018].
  
- [16] National Instruments, <<LabVIEW. Control Design User Manual>>, Jun-2009.
  
- [17] Daniel Rodríguez Ramírez, Carlos Bordóns Alba, <<Apuntes de ingeniería de control. Análisis y control de sistemas en espacio de estado. Identificación de sistemas. Control adaptativo. Control predictivo.>>, 5-May-2005.

## **ANEXOS**

### **ANEXO I: MODELOS TEÓRICOS**

En este documento se detallan los cálculos, simplificaciones y suposiciones realizados para desarrollar los modelos matemáticos de los tres sistemas que son objeto de estudio. Este anexo es una ampliación del apartado 6.4. *MODELOS TEÓRICOS*, contenido en el apartado 6. *METODOLOGÍA* del documento principal.

## 1. Modelo del motor y el carro

En este apartado se obtiene el modelo matemático que relaciona el voltaje aplicado al motor de corriente continua en el carro con la fuerza resultante sobre el conjunto motor-carro para desplazarse por la pista.

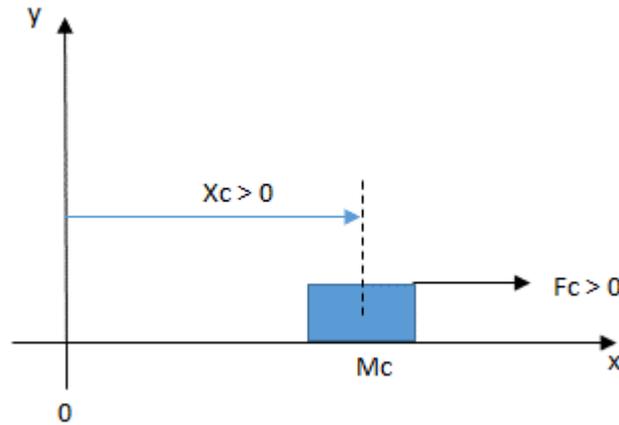


Figura 1. Esquema del modelo del motor y del carro

Aplicando balance de fuerzas al movimiento del carro se obtiene la ecuación (1)

$$F_c(t) - B_{eq} \dot{x}_c(t) - F_{ai}(t) = M_c \ddot{x}_c(t) \quad (1)$$

donde  $F_c(t)$  es la fuerza de tracción en el eje del piñón motriz del carro,  $B_{eq}$  el coeficiente de rozamiento viscoso,  $F_{ai}(t)$  la fuerza inercial del rotor,  $M_c$  la masa del carro y  $x_c(t)$  la posición del carro respecto a su posición inicial.

La expresión de la fuerza inercial del rotor se deduce de las siguientes ecuaciones:

$$F_{ai}(t) = \frac{\eta_g K_g T_{ai}(t)}{r_{mp}} \quad (2)$$

$$T_{ai}(t) = J_m \ddot{\theta}_m(t) \quad (3)$$

$$\theta_m(t) = \frac{K_g x_c(t)}{r_{mp}} \quad (4)$$

y tiene la siguiente forma:

$$F_{ai}(t) = \frac{\eta_g K_g^2 J_m \ddot{x}_c(t)}{r_{mp}^2} \quad (5)$$

Siendo  $T_{ai}(t)$  el par de inercia del rotor,  $K_g$  la relación del reductor y, por tanto, la multiplicación de ambas el par de inercia del rotor visto en el piñón de salida,  $\eta_g$  el rendimiento del reductor,  $r_{mp}$  el radio del piñón motriz,  $J_m$  el momento de inercia del rotor y  $\theta_m(t)$  el ángulo girado por el rotor.

Por otro lado, teniendo en cuenta el circuito equivalente del motor mostrado en la figura siguiente:

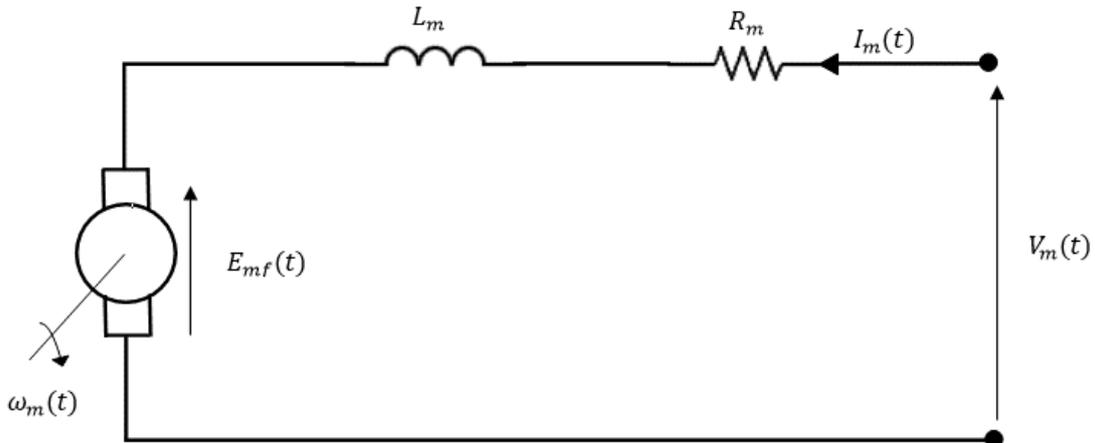


Figura 2. Esquema eléctrico del motor

la ecuación que rige el comportamiento del motor es la que se indica a continuación:

$$V_m(t) - R_m I_m(t) - L_m \frac{dI_m(t)}{dt} - E_{mf}(t) = 0 \quad (6)$$

donde  $V_m(t)$  es la tensión aplicada al motor,  $R_m$  la resistencia de su devanado,  $I_m(t)$  la intensidad que circula por él,  $L_m$  su inductancia y  $E_{mf}(t)$  la fuerza contraelectromotriz. Como en el caso del motor de la maqueta la inductancia es mucho menor que la resistencia, el término que acompaña a  $L_m$  se desprecia, quedando así una expresión más sencilla que permite despejar  $I_m(t)$  de una manera más simple.

$$V_m(t) - R_m I_m(t) - E_{mf}(t) = 0 \quad (7)$$

$$I_m(t) = \frac{V_m(t) - E_{mf}(t)}{R_m} = \frac{V_m(t) - K_m \omega_m(t)}{R_m} \quad (8)$$

Esta última expresión de la ecuación (8) se obtiene teniendo en cuenta que la fuerza contraelectromotriz es igual a la velocidad angular del motor  $\omega_m(t)$  multiplicada por la constante electromotriz  $K_m$ .

Teniendo en cuenta que la fuerza de tracción viene dada por la ecuación (9), análoga a la ecuación de la fuerza inercial del rotor (2) pero involucrando el par motor  $T_m(t)$ , que viene dado por la ecuación (10), mediante el rendimiento del motor  $\eta_m$ , la constante par-corriente  $K_t$  y la intensidad del motor. Sustituyendo la expresión (8) en (10), y la expresión resultante en (9), se obtiene la ecuación (11).

$$F_c(t) = \frac{\eta_g K_g T_m(t)}{r_{mp}} \quad (9)$$

$$T_m(t) = \eta_m K_t I_m(t) \quad (10)$$

$$F_c(t) = \frac{\eta_g \eta_m K_t K_g [V_m(t) - K_m \omega_m(t)]}{r_{mp} R_m} \quad (11)$$

Como, además, se sabe que

$$\omega_m(t) = \frac{K_g \dot{x}_c(t)}{r_{mp}} \quad (12)$$

La expresión final de la fuerza de tracción es

$$F_c(t) = \frac{\eta_g \eta_m K_t K_g [r_{mp} V_m(t) - K_m K_g \dot{x}_c(t)]}{r_{mp}^2 R_m} \quad (13)$$

Sustituyendo (5) y (13) en (1) se obtiene:

$$\frac{\eta_g \eta_m K_t K_g [r_{mp} V_m(t) - K_m K_g \dot{x}_c(t)]}{r_{mp}^2 R_m} - B_{eq} \dot{x}_c(t) - \frac{\eta_g K_g^2 J_m \ddot{x}_c(t)}{r_{mp}^2} = M_c \ddot{x}_c(t) \quad (14)$$

$$\frac{\eta_g \eta_m K_t K_g V_m(t)}{r_{mp} R_m} - \left[ \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m} + B_{eq} \right] \dot{x}_c(t) - \left[ \frac{\eta_g K_g^2 J_m}{r_{mp}^2} + M_c \right] \ddot{x}_c(t) = 0 \quad (15)$$

Aplicando a la expresión (15) transformada de Laplace y reordenando los términos se llega a la expresión (16).

$$\frac{\eta_g \eta_m K_t K_g}{r_{mp} R_m} V_m(s) = \left[ \frac{\eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m}{r_{mp}^2 R_m} s + \frac{\eta_g K_g^2 J_m + r_{mp}^2 M_c}{r_{mp}^2} s^2 \right] x_c(s) \quad (16)$$

Por lo que entre la alimentación de tensión al motor y la posición del carro existe la siguiente función de transferencia:

$$G(s) = \frac{x_c(s)}{V_m(s)} = \frac{\eta_g \eta_m K_t K_g r_{mp}}{s \left[ (\eta_g K_g^2 J_m R_m + r_{mp}^2 M_c R_m) s + \eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m \right]} \quad (17)$$

Ésta es la expresión que caracteriza el primer sistema de estudio, el carro. El sistema también puede ser representado por sus ecuaciones de estado, a partir de la ecuación (15).

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \quad \text{donde } x = \begin{bmatrix} x_c(t) \\ \dot{x}_c(t) \end{bmatrix}, u = V_m(t) \text{ e } y = x_c(t)$$

$$A = \begin{bmatrix} 0 & 1 \\ 0 & -\frac{(\eta_g \eta_m K_t K_g^2 K_m + B_{eq} r_{mp}^2 R_m)}{R_m (\eta_g K_g^2 J_m + r_{mp}^2 M_c)} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ \frac{(\eta_g \eta_m K_t K_g) r_{mp}}{R_m (\eta_g K_g^2 J_m + r_{mp}^2 M_c)} \end{bmatrix}$$

$$C = [1 \quad 0] \quad (18)$$

Se valora añadir al modelo matemático un tiempo muerto, que podría generarse debido a la fricción estática entre el carro y la pista. Sin embargo, tras pruebas en el sistema se determina que dicho tiempo muerto es despreciable, por lo que finalmente no es introducido.

## 2. Modelo del péndulo Gantry

En el caso de la inclusión del péndulo Gantry en el carro, el nuevo modelo se obtiene partiendo de las expresiones de la fuerza y la posición del carro calculadas anteriormente.

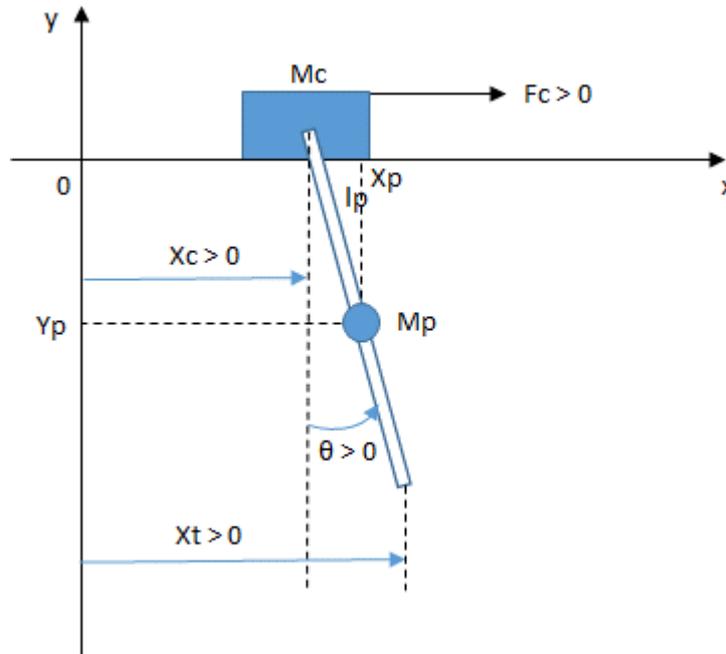


Figura 3. Esquema del modelo del péndulo Gantry

En la *Figura 3* se representa el sistema de forma simplificada, mostrando las variables de salida del mismo, el ángulo  $\theta$  del péndulo y la posición del carro  $x_c$ , la masa del péndulo  $M_p$ , la distancia  $l_p$  desde la articulación hasta su centro de gravedad, así como sus coordenadas  $x_p$  e  $y_p$ .

Para plantear el modelo del sistema se utilizan las ecuaciones de Euler-Lagrange (19), donde  $q_i$  son las variables independientes del sistema, en este caso  $\theta$  y  $x_c$ ;  $L$  el Lagrangiano, que se define, según la ecuación (20), como la resta entre la energía cinética total del sistema  $T_T$  y la energía potencial total del sistema  $V_T$ ; y  $Q_i$  son las fuerzas generalizadas, que son aquellas fuerzas externas que pueden producir un cambio en la coordenada  $q_i$ .

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}_i} \right) - \frac{\partial L}{\partial q_i} = Q_i \quad (19)$$

$$L = T_T - V_T \quad (20)$$

En el caso del sistema de estudio, las ecuaciones son las de la expresión (21), donde  $B_p$  es el coeficiente de rozamiento viscoso de la articulación del péndulo.

$$\begin{cases} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}_c} \right) - \frac{\partial L}{\partial x_c} = Q_{x_c} = F_c(t) - B_{eq} \dot{x}_c(t) \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) - \frac{\partial L}{\partial \theta} = Q_{\theta} = -B_p \dot{\theta}(t) \end{cases} \quad (21)$$

Dado que el carro se mueve en un plano horizontal, se toma como referencia dicho plano para la energía potencial, por lo que sólo el péndulo contribuye a ésta, siendo  $g$  la constante de gravedad en la superficie terrestre.

$$V_T = -M_p g l_p \cos \theta(t) \quad (22)$$

La energía cinética del sistema es la suma de la energía cinética del carro  $T_c$  y la del péndulo  $T_p$ . En ambos casos, hay un término debido a la traslación y otro a la rotación, en el caso del carro debido al motor.

$$T_T = T_c - V_p \quad (23)$$

Para el carro:

$$T_c = \frac{1}{2} M_c \dot{x}_c(t)^2 + \frac{1}{2} J_m \omega_m(t)^2 \quad (24)$$

$$T_c = \frac{1}{2} M_c \dot{x}_c(t)^2 + \frac{1}{2} \frac{J_m K_g^2 \dot{x}_c(t)^2}{r_{mp}^2} = \frac{1}{2} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) \dot{x}_c(t)^2 \quad (25)$$

La ecuación (25) se obtiene sustituyendo la expresión (12) en  $\frac{1}{2} J_m \omega_m(t)^2$ .

Para el péndulo:

$$T_p = \frac{1}{2} M_p \dot{l}_p(t)^2 + \frac{1}{2} J_p \dot{\theta}(t)^2 \quad (26)$$

Aunque el modelo se expresa en función de  $J_p$ , éste se calcula con la siguiente fórmula:

$$J_p = \frac{1}{12} M_p (2 l_p)^2 = \frac{1}{3} M_p l_p^2 \quad (27)$$

Teniendo en cuenta las siguientes relaciones, que se pueden deducir del esquema del sistema:

$$\dot{l}_p(t)^2 = \dot{x}_p(t)^2 + \dot{y}_p(t)^2 \quad (28)$$

$$x_p(t) = x_c(t) + l_p \sin \theta(t) \quad (29)$$

$$y_p(t) = -l_p \cos \theta(t) \quad (30)$$

Derivando respecto al tiempo las dos últimas ecuaciones se obtienen las expresiones (31) y (32), respectivamente.

$$\dot{x}_p(t) = \dot{x}_c(t) + l_p \cos \theta(t) \dot{\theta}(t) \quad (31)$$

$$\dot{y}_p(t) = l_p \sin \theta(t) \dot{\theta}(t) \quad (32)$$

Por lo tanto, sustituyendo (28), (31) y (32) en (26), y posteriormente sustituyendo (25) y (26) en (23), se obtiene una expresión de la energía cinética total con la siguiente forma.

$$T_T = \frac{1}{2} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} \right) \dot{x}_c(t)^2 + \frac{1}{2} M_p \left[ \left( \dot{x}_c(t) + l_p \cos \theta(t) \dot{\theta}(t) \right)^2 + \left( l_p \sin \theta(t) \dot{\theta}(t) \right)^2 \right] + \frac{1}{2} J_p \dot{\theta}(t)^2 \quad (33)$$

Operando y reordenando términos se llega a la expresión (34).

$$T_T = \frac{1}{2} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \dot{x}_c(t)^2 + M_p l_p \cos \theta(t) \dot{\theta}(t) \dot{x}_c(t) + \frac{1}{2} (M_p l_p^2 + J_p) \dot{\theta}(t)^2 \quad (34)$$

Por lo tanto, finalmente, el Lagrangiano y sus derivadas parciales quedan de la siguiente manera.

$$L = \frac{1}{2} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \dot{x}_c(t)^2 + M_p l_p \cos \theta(t) \dot{\theta}(t) \dot{x}_c(t) + \frac{1}{2} (M_p l_p^2 + J_p) \dot{\theta}(t)^2 + M_p g l_p \cos \theta(t) \quad (35)$$

$$\frac{\partial L}{\partial \dot{x}_c(t)} = \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \dot{x}_c(t) + M_p l_p \cos \theta(t) \dot{\theta}(t) \quad (36)$$

$$\frac{\partial L}{\partial x_c(t)} = 0 \quad (37)$$

$$\frac{\partial L}{\partial \dot{\theta}(t)} = M_p l_p \cos \theta(t) \dot{x}_c(t) + (M_p l_p^2 + J_p) \dot{\theta}(t) \quad (38)$$

$$\frac{\partial L}{\partial \theta(t)} = -M_p l_p \sin \theta(t) \dot{\theta}(t) \dot{x}_c(t) - M_p g l_p \sin \theta(t) \quad (39)$$

Sustituyendo las derivadas parciales del Lagrangiano en las ecuaciones de Euler-Lagrange (21) y operando se obtiene el siguiente sistema de ecuaciones.

$$\begin{cases} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \ddot{x}_c(t) + (M_p l_p \cos \theta(t) \ddot{\theta}(t) - M_p l_p \sin \theta(t) \dot{\theta}(t)^2) + B_{eq} \dot{x}_c(t) = F_c(t) \\ M_p l_p \cos \theta(t) \ddot{x}_c(t) + (M_p l_p^2 + J_p) \ddot{\theta}(t) + M_p g l_p \sin \theta(t) + B_p \dot{\theta}(t) = 0 \end{cases} \quad (40)$$

Si además se linealiza la expresión (40) en torno al punto de equilibrio, es decir, para  $\theta(t) \simeq 0$ , lo que implica  $\cos \theta(t) \simeq 1$  y  $\sin \theta(t) \simeq \theta(t)$ , se obtiene la expresión (41).

$$\begin{cases} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \ddot{x}_c(t) + M_p l_p \ddot{\theta}(t) - M_p l_p \theta(t) \dot{\theta}(t)^2 + B_{eq} \dot{x}_c(t) = F_c(t) \\ M_p l_p \ddot{x}_c(t) + (M_p l_p^2 + J_p) \ddot{\theta}(t) + M_p g l_p \theta(t) + B_p \dot{\theta}(t) = 0 \end{cases} \quad (41)$$

Adicionalmente, si se desprecia el término de orden superior porque la velocidad angular del péndulo es muy pequeña en torno al punto de equilibrio.

$$\begin{cases} \left( M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p \right) \ddot{x}_c(t) + M_p l_p \ddot{\theta}(t) + B_{eq} \dot{x}_c(t) = F_c(t) \\ M_p l_p \ddot{x}_c(t) + (M_p l_p^2 + J_p) \ddot{\theta}(t) + M_p g l_p \theta(t) + B_p \dot{\theta}(t) = 0 \end{cases} \quad (42)$$

Despejando en la segunda ecuación de (42)  $\ddot{\theta}(t)$  se obtiene (43), y sustituyendo (43) en la primera ecuación de (42) y despejando  $\ddot{x}_c(t)$  se llega a (44).

$$\ddot{\theta}(t) = \left( -\frac{M_p l_p}{M_p l_p^2 + J_p} \right) \ddot{x}_c(t) - \left( \frac{M_p g l_p \theta(t) + B_p \dot{\theta}(t)}{M_p l_p^2 + J_p} \right) \quad (43)$$

$$\ddot{x}_c(t) = \frac{F_c(t) + \frac{M_p^2 g l_p^2 \theta(t) + M_p l_p B_p \dot{\theta}(t)}{M_p l_p^2 + J_p} - B_{eq} \dot{x}_c(t)}{M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p - \frac{M_p^2 l_p^2}{M_p l_p^2 + J_p}} \quad (44)$$

La última expresión se puede simplificar ligeramente, llegando a la forma que se muestra a continuación.

$$\ddot{x}_c(t) = \frac{(M_p l_p^2 + J_p) F_c(t) + M_p^2 g l_p^2 \theta(t) + M_p l_p B_p \dot{\theta}(t) - (M_p l_p^2 + J_p) B_{eq} \dot{x}_c(t)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \quad (45)$$

Ya que la expresión (45) no depende de  $\ddot{\theta}(t)$ , si se sustituye en la (43) y se simplifica, se obtiene la ecuación (46), con lo que las derivadas segundas de las dos variables de salida del sistema se encuentran desacopladas.

$$\ddot{\theta}(t) = \frac{-\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p \theta(t) + M_p l_p B_{eq} \dot{x}_c(t) - \left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) B_p \dot{\theta}(t) - M_p l_p F_c(t)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \quad (46)$$

A partir de las expresiones (45) y (46) se pueden deducir las ecuaciones de estado del sistema, que tendrán la siguiente forma y los valores que se detallan a continuación.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \theta(t) \\ \dot{x}_c(t) \\ \dot{\theta}(t) \end{bmatrix}, u = F_c(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Siendo,

$$a_{32} = \frac{M_p^2 g l_p^2}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{33} = -\frac{(M_p l_p^2 + J_p) B_{eq}}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{34} = \frac{M_p l_p B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{42} = -\frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{43} = \frac{M_p l_p B_{eq}}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{44} = - \frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{(M_p l_p^2 + J_p)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\ \frac{M_p l_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \end{bmatrix}$$

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (47)$$

El modelo de ecuaciones de estado representado en (47) tiene como salidas del sistema las deseadas, pero la fuerza del carro es la entrada al mismo, mientras que es deseable tener la tensión aplicada en el motor como entrada, ya que será la acción de control que se introduzca al sistema. Por lo tanto, sustituyendo la expresión (13) en (45) y (46), se obtienen las expresiones de  $\ddot{x}_c(t)$  y  $\ddot{\theta}(t)$  en función de la tensión del motor.

Si se calculan en este caso, las ecuaciones de estado quedan como se muestra a continuación.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \theta(t) \\ \dot{x}_c(t) \\ \dot{\theta}(t) \end{bmatrix}, u = V_m(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Siendo,

$$a_{32} = \frac{M_p^2 g l_p^2}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$a_{33} = - \frac{\left(M_p l_p^2 + J_p\right) \left(B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m}\right)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2}$$

$$\begin{aligned}
 a_{34} &= \frac{M_p l_p B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{42} &= -\frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{43} &= \frac{M_p l_p \left(B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m}\right)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{44} &= -\frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 B &= \begin{bmatrix} 0 \\ 0 \\ \frac{\left(M_p l_p^2 + J_p\right) \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2\right]} \\ \frac{M_p l_p \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2\right]} \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (48)
 \end{aligned}$$

Aunque tanto el sistema péndulo invertido como el sistema péndulo Gantry corresponden a sistemas SIMO, Single Input ( $V_m(t)$ ) y Multiple Output ( $x_c(t)$  y  $\theta(t)$ ), en el caso del péndulo Gantry se simula el movimiento de un puente-grúa en el que el objetivo es minimizar las oscilaciones en la carga, es decir, en el extremo del péndulo.

Por ello, la variable a controlar es  $x_t(t)$  de la *Figura 3*:

La salida que interesa, por tanto, no es  $y = \begin{bmatrix} x_c(t) \\ \theta(t) \end{bmatrix}$ , sino  $y = x_t(t)$ , tratando por lo tanto el sistema como SISO, Single Input ( $V_m(t)$ ), Single Output ( $x_t(t)$ ).

De acuerdo a lo anterior se puede calcular la nueva matriz C para  $y = x_t(t)$  La variable  $x_t(t)$  se define según la ecuación (49).

$$x_t(t) = x_c(t) + 2 l_p \sin \theta(t) \quad (49)$$

Linealizando en torno al punto de equilibrio, se asume que el ángulo es igual al seno del mismo, por lo que se obtiene la expresión (50), a partir de la cual se deduce C.

$$x_t(t) = x_c(t) + 2 l_p \theta(t) \quad (50)$$

$$C = [1 \quad 2 l_p \quad 0 \quad 0] \quad (51)$$

### 3. Modelo del péndulo invertido

En el caso del péndulo invertido, el planteamiento es el que se muestra en la *Figura 4*, análogo al caso anterior pero con una referencia distinta para el ángulo, que en este caso se denomina  $\alpha(t)$ , que hace que las ecuaciones (29) y (30) cambien en un signo.

$$x_p(t) = x_c(t) - l_p \sin \alpha(t) \quad (52)$$

$$y_p(t) = l_p \cos \alpha(t) \quad (53)$$

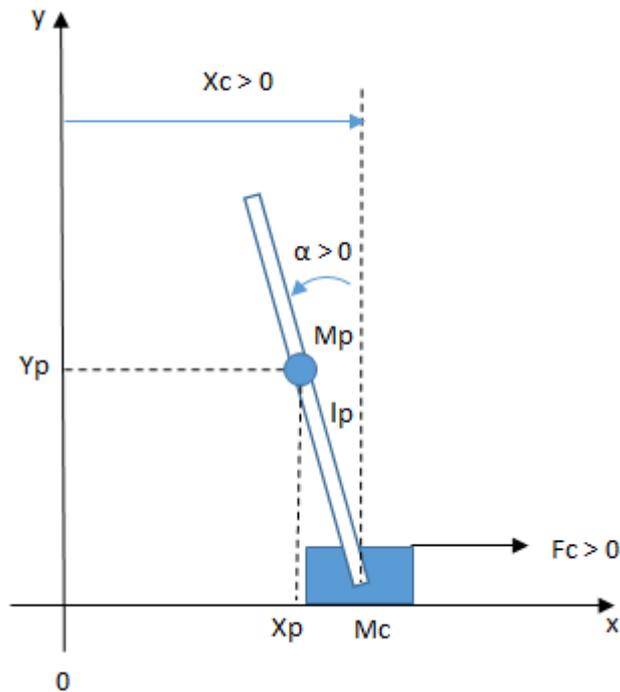


Figura 4. Esquema del modelo del péndulo invertido

Por lo tanto, desarrollando las ecuaciones de la misma manera que en el caso del péndulo Gantry, se obtiene el modelo para el caso del péndulo invertido, dado por el siguiente sistema de ecuaciones de estado. Se puede apreciar que únicamente cambian algunos signos de los términos con respecto al modelo anterior.

$$\begin{cases} \dot{x} = A x + B u \\ y = C x \end{cases} \text{ donde } x = \begin{bmatrix} x_c(t) \\ \alpha(t) \\ \dot{x}_c(t) \\ \dot{\alpha}(t) \end{bmatrix}, u = V_m(t) \text{ e } y = \begin{bmatrix} x_c(t) \\ \alpha(t) \end{bmatrix}$$

$$A = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & a_{32} & a_{33} & a_{34} \\ 0 & a_{42} & a_{43} & a_{44} \end{bmatrix}$$

Donde,

$$\begin{aligned}
 a_{32} &= \frac{M_p^2 g l_p^2}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{33} &= - \frac{\left(M_p l_p^2 + J_p\right) \left(B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m}\right)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{34} &= - \frac{M_p l_p B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{42} &= \frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) M_p g l_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{43} &= - \frac{M_p l_p \left(B_{eq} + \frac{\eta_g \eta_m K_t K_g^2 K_m}{r_{mp}^2 R_m}\right)}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 a_{44} &= - \frac{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) B_p}{\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2} \\
 B &= \begin{bmatrix} 0 \\ 0 \\ \frac{\left(M_p l_p^2 + J_p\right) \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2\right]} \\ \frac{M_p l_p \eta_g \eta_m K_t K_g}{r_{mp} R_m \left[\left(M_c + \frac{J_m K_g^2}{r_{mp}^2} + M_p\right) J_p + \left(M_c + \frac{J_m K_g^2}{r_{mp}^2}\right) M_p l_p^2\right]} \end{bmatrix} \\
 C &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad (54)
 \end{aligned}$$

## **ANEXO II: PARÁMETROS DEL SISTEMA**

En este anexo se presentan los valores de los parámetros del sistema que han sido utilizados para calcular los modelos teóricos. Cabe destacar que se muestran dos masas de carro y dos coeficientes de rozamiento de carro distintos, debido a una masa extra que se puede añadir al mismo. Además, para el péndulo se presentan dos valores distintos, ya que existen dos péndulos que se pueden acoplar al carro, uno corto y otro de mayor longitud.

<b>Carro</b>	<b>ID</b>	<b>Unidades</b>	<b>Valor</b>
Masa	$M_c$	[Kg]	0.38
Masa (con masa extra)	$M_c$	[Kg]	0.745
Coefficiente de rozamiento viscoso (sólo carro y péndulo corto)	$B_{eq}$	[Ns/m]	4.3
Coefficiente de rozamiento viscoso (carro con masa extra y/o péndulo largo)	$B_{eq}$	[Ns/m]	5.4

<b>Péndulo</b>			<b>Corto/Largo</b>
Masa	$M_p$	[Kg]	0.129/0.23
Distancia desde la articulación al CDM del péndulo	$l_p$	[m]	0.155/0.33
Coefficiente de amortiguamiento viscoso de la articulación del péndulo	$B_p$	[Nms/rad]	0/0
Momento de inercia del péndulo	$J_p$	[Kgm <sup>2</sup> ]	0.00103/0.00835

<b>Motor</b>			
Tensión nominal del motor	$V_m$	[V]	6.0
Constante Par-Corriente	$K_t$	[Nm/A]	0.00767
Constante contra electromotriz	$K_m$	[V/rad/s]	0.00767
Resistencia de la armadura del motor	$R_m$	[Ω]	2.60
Inductancia de la armadura del motor	$L_m$	[mH]	0.18
Eficiencia del motor	$\eta_m$	[-]	0.9
Momento de inercia del motor	$J_m$	[Kgm <sup>2</sup> ]	3.90E-07

<b>Transmisión</b>			
Relación reductora	$K_g$	[-]	3.71
Eficiencia del reductor	$\eta_g$	[-]	0.9
Radio del piñón motriz	$r_{mp}$	[m]	0.00635

Algunos parámetros han sido corregidos con respecto a los que proporciona Quanser durante el proceso de validación del modelo, principalmente parámetros difíciles de medir como rozamientos o que han podido cambiar con el tiempo, como rendimientos. A continuación, se presentan los valores originales proporcionados por Quanser, aunque los usados en los modelos teóricos han sido los de la tabla anterior.

<b>Carro</b>	<b>ID</b>	<b>Unidades</b>	<b>Valor</b>
Masa	$M_c$	[Kg]	0.38
Masa (con masa extra)	$M_c$	[Kg]	0.745
Coeficiente de rozamiento viscoso (sólo carro)	$B_{eq}$	[Ns/m]	4.3
Coeficiente de rozamiento viscoso (carro con masa extra)	$B_{eq}$	[Ns/m]	5.4

<b>Péndulo</b>	<b>Corto/Largo</b>		
Masa	$M_p$	[Kg]	0.129/0.23
Distancia desde la articulación al CDM del péndulo	$l_p$	[m]	0.155/0.33
Coeficiente de amortiguamiento viscoso de la articulación del péndulo	$B_p$	[Nms/rad]	<u>0.0024/0.0024</u>
Momento de inercia del péndulo	$J_p$	[Kgm <sup>2</sup> ]	0.00103/0.00835

<b>Motor</b>			
Tensión nominal del motor	$V_m$	[V]	6.0
Constante Par-Corriente	$K_t$	[Nm/A]	0.00767
Constante contra electromotriz	$K_m$	[V/rad/s]	0.00767
Resistencia de la armadura del motor	$R_m$	[ $\Omega$ ]	2.60
Inductancia de la armadura del motor	$L_m$	[mH]	0.18
Eficiencia del motor	$\eta_m$	[-]	<u>0.69</u>
Momento de inercia del motor	$J_m$	[Kgm <sup>2</sup> ]	3.90E-07

<b>Transmisión</b>			
Relación reductora	$K_g$	[-]	3.71
Eficiencia del reductor	$\eta_g$	[-]	0.9
Radio del piñón motriz	$r_{mp}$	[m]	0.00635

Se subrayan los parámetros que han sido corregidos.

### **ANEXO III: GANANCIAS ÓPTIMAS DE LOS CONTROLADORES**

En este anexo se presentan resumidas las ganancias óptimas obtenidas a lo largo del proyecto, para facilitar el uso de la aplicación de usuario creada.

Para las matrices  $Q$  que se presentan para los controladores LQR, se muestra únicamente la diagonal principal de la matriz, siendo el resto de sus términos nulos.

### 1. Sistema carro

Control		Kp	Ti(min)	Td(min)
PID		50	5.010	0
Control	Entradas	K		Ki
Polos	$p_{1,2} = -4 \pm 1,364 j$ $p_3 = -15$	41.04	-0.13	79.76
LQR	Q(200 1 3000) R=0.3	51.38	2.30	100

### 2. Sistema péndulo Gantry

Control	Péndulo	Xc			Alfa		
		Kp	Ti(min)	Td(min)	Kp	Ti(min)	Td(min)
2 PID	Corto	20	0.4	$2.5 \times 10^{-5}$	-30	5	$3.33 \times 10^{-6}$
	Largo	20	0.4	$2.5 \times 10^{-5}$	-30	5	$3.33 \times 10^{-6}$

Control	Péndulo	Entradas	K			Ki	
LQR	Corto	Q(100 100 4 10 500) R=0.3	47.50	-40.70	16.00	-1.80	-40.80
	Largo	Q(100 100 4 10 500) R=0.05	113.42	-77.72	45.97	7.31	100
Polos sin SO	Corto	$p_{1,2} = -4 \pm 1,364 j$ $p_{3,4,5} = -7, -11, -15$	127.07	-19.98	31.44	4.75	169.77
	Largo	$p_{1,2} = -4 \pm 1,364 j$ $p_{3,4,5} = -7, -11, -15$	283.20	16.73	83.08	32.85	378.44

### 3. Sistema péndulo invertido

Control	Péndulo	Xc			Alfa		
		Kp	Ti(min)	Td(min)	Kp	Ti(min)	Td(min)
2 PID	Corto	-51	1.35	0.01	54	500	0.0024
	Largo	-50	5	0.018	110	Inf	0.0035

Control	Péndulo	Entradas	K				Ki
LQR	Corto	Q(30 30 1 2 60) R=0.05	-50.86	85.39	-36.24	13.36	-34.64
	Largo	Q(30 30 1 2 60) R=0.05	-54.02	114.15	-41.81	24.20	-34.64
Polos sin SO	Corto	$p_{1,2} = -2 \pm 0,827 j$ $p_{3,4,5} = -7, -11, -15$	-51.41	54.14	--30.70	7.76	-44.53
	Largo	$p_{1,2} = -2 \pm 0,827 j$ $p_{3,4,5} = -7, -11, -15$	-114.57	142.57	-62.29	30.38	-99.25
Polos con SO	Corto	$p_{1,2} = -1,818 \pm 1,460 j$ $p_{3,4,5} = -7, -11, -15$	-50.09	52.96	-30.02	7.59	-51.69
	Largo	$p_{1,2} = -1,333 \pm 1,071 j$ $p_{3,4,5} = -7, -11, -15$	-75.10	116.98	-51.06	25.20	-61.95

#### 4. Autoerección

##### Corto

Parámetro	Valor	Unidades
$M_p$	0.129	[Kg]
$l_p$	0.155	[m]
$E_r$	0.8	[J]
$\mu$	10	[kg <sup>-1</sup> m <sup>-1</sup> ]

##### Largo

Parámetro	Valor	Unidades
$M_p$	0.23	[Kg]
$l_p$	0.33	[m]
$E_r$	1.9	[J]
$\mu$	18	[kg <sup>-1</sup> m <sup>-1</sup> ]

## **ANEXO IV: MANUAL DE USUARIO DE LA APLICACIÓN**

Este documento explica al usuario de la aplicación desarrollada en este proyecto cómo se debe utilizar y qué posibilidades ofrece la misma.

# Sistema de Análisis de Controladores Avanzados para la Maqueta IP02 de Quanser

## Manual de Usuario

## **ÍNDICE**

1. Introducción .....	4
2. Descripción .....	5
2.1. Maqueta .....	5
2.2. Universal Power Module (UPM) .....	6
2.3. Tarjeta NI myRIO .....	7
2.4. Ordenador .....	8
2.5. Esquema de interconexión .....	8
3. Conexión de la tarjeta .....	9
4. Carga de la aplicación en tiempo real .....	11
5. Instalación de la aplicación para ordenador .....	14
6. Preparación de la maqueta IP02 de Quanser .....	15
7. Uso de la aplicación para ordenador.....	18
7.1. Inicio.....	19
7.2. Preparación de la prueba .....	19
7.3. Control en tiempo real y monitorización.....	26
7.4. Análisis de resultados.....	35
7.5. Ejecución anormal de la aplicación.....	39

## **1. Introducción**

Este documento es un manual de usuario para la aplicación *Sistema de Análisis de Controladores Avanzados para la Maqueta IP02 de Quanser*, ubicada en los laboratorios de la Escuela de Ingeniería de Bilbao.

Esta aplicación tiene por objetivo la realización de pruebas sobre la maqueta IP02 de Quanser, que modela los sistemas carro, péndulo Gantry y péndulo invertido, además de hacer simulaciones sobre modelos matemáticos de éstos sistemas.

A continuación, se describe el sistema, sus parámetros más importantes y cómo utilizar la aplicación y las posibilidades que ofrece.

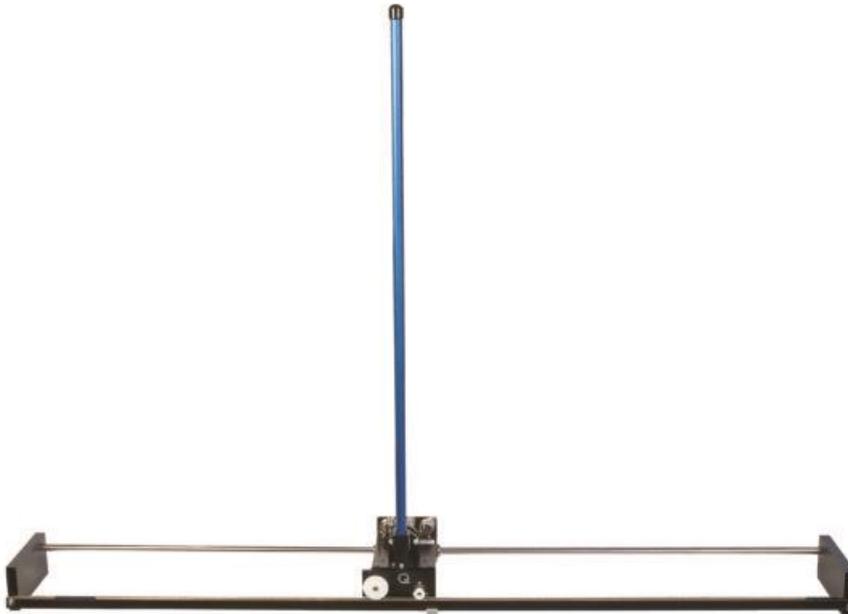
## **2. Descripción**

A continuación, se describen las distintas partes que componen el total del proyecto.

### **2.1. Maqueta**

La maqueta IP02 de Quanser, que sirve como soporte de los tres sistemas que se estudian en este documento, consta básicamente de una pista sobre la que puede moverse un carro accionado por un motor de corriente continua, y al que se pueden acoplar dos péndulos articulados de distinta longitud en un eje rotativo montado en el mismo, y perpendicular al mismo y a la pista. Además, se dispone de una masa que se puede añadir al carro.

La maqueta incluye dos encoders, uno para medida de la posición lineal del carro en la pista y otro para medida de la posición angular del péndulo.



*Figura 1. Maqueta IP02 de Quanser*

En la *Figura 2* se muestran todos los componentes de la maqueta, enumerados y nombrados en la *Tabla 1*.

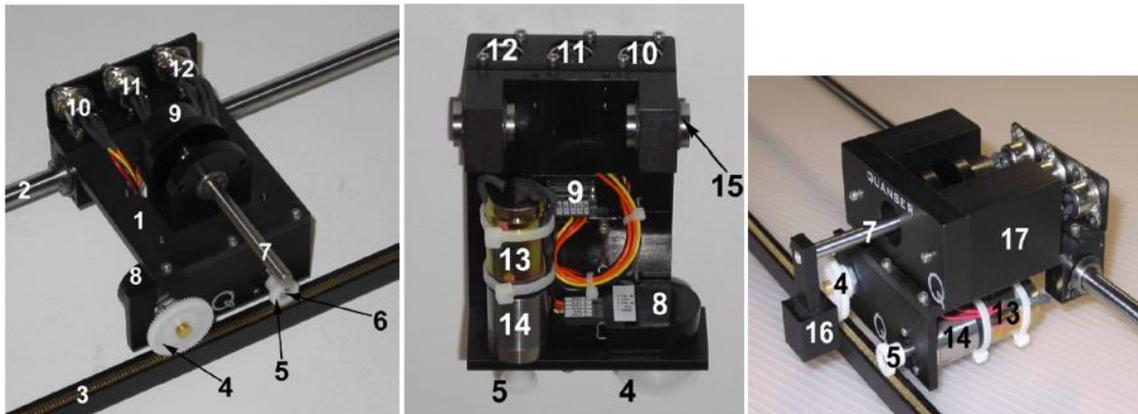


Figura 2. Componentes de la maqueta IP02 de Quanser

ID	DESCRIPCIÓN
1	Carro de la maqueta
2	Eje-guía de acero inoxidable
3	Cremallera
4	Piñón de accionamiento del encoder de posición del carro
5	Piñón motriz del carro
6	Eje del piñón motriz del carro
7	Eje de giro del péndulo
8	Encoder de posición del carro
9	Encoder de posición del péndulo
10	Conector del encoder de posición del carro
11	Conector del encoder de posición del péndulo
12	Conector de alimentación al motor de DC
13	Motor de DC
14	Motorreductor planetario
15	Cojinete de deslizamiento lineal
16	Soporte del péndulo
17	Masa supletoria del carro

Tabla 1. Lista de componentes de la maqueta IP02 de Quanser

## **2.2. Universal Power Module (UPM)**

Se dispone de un módulo amplificador Universal Power Module (UPM) cuya misión es amplificar en potencia la señal de control al motor. La resistencia del cable de conexión entre la UPM y el motor determina la amplificación en voltaje, que en este caso es de valor 1, es decir, el voltaje no varía. La salida de la UPM puede suministrar, sin embargo, una corriente de hasta 1A al motor.



Figura 3. Módulo UPM

### **2.3. Tarjeta NI myRIO**

La implementación del control en tiempo real y la adquisición de datos se realiza mediante una tarjeta NI myRIO de National Instruments, concretamente el modelo 1900.

Los programas de control en tiempo real son ejecutados por esta tarjeta.



Figura 4. Tarjeta NI myRIO de National Instruments

## 2.4. Ordenador

En el ordenador se aloja la aplicación de interfaz de usuario. Para utilizar la aplicación, es necesario disponer de una versión de Windows XP o superior.

La tarjeta NI myRIO se conecta vía USB al ordenador.

## 2.5. Esquema de interconexión

A continuación, se muestra el esquema de interconexión entre los distintos subsistemas que conforman el sistema general, y el flujo de programas y señales.

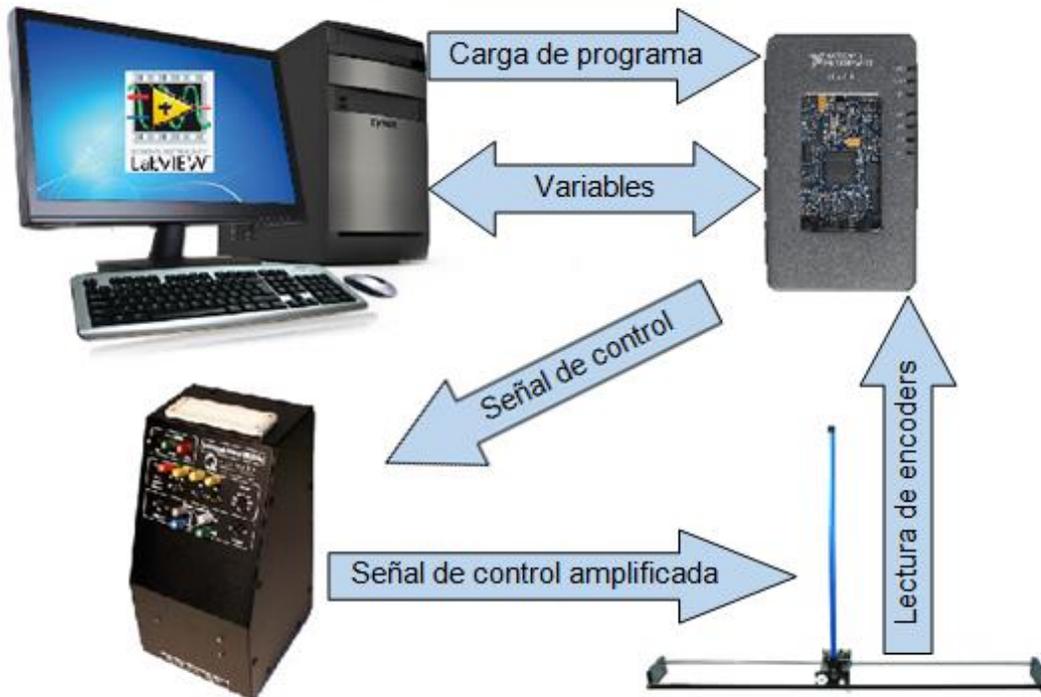


Figura 5. Esquema de interconexión de los sistemas

### 3. Conexión de la tarjeta

Para poder utilizar la aplicación para pruebas en tiempo real, es necesario conectar una tarjeta NI myRIO vía USB al ordenador y alimentarla.



Figura 6. Tarjeta NI myRIO de National Instruments

Tras la conexión, es posible, si se tiene el software de LabVIEW para el desarrollo de aplicaciones en la tarjeta NI myRIO instalado, que aparezca el siguiente cuadro de diálogo. Únicamente hay que cerrarlo, haciendo clic en la esquina superior de la ventana, o seleccionando la opción “Do Nothing”.

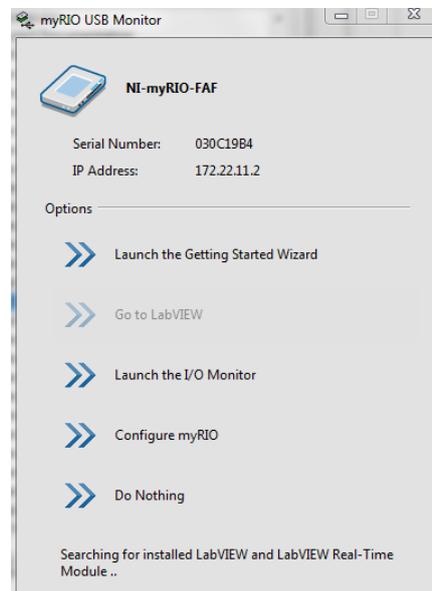


Figura 7. Cuadro de diálogo al conectar la tarjeta NI myRIO

Una vez alimentada la tarjeta, se debe esperar a que el LED indicador LED0 se encienda para empezar a utilizar la aplicación para ordenador.



Figura 8. Tarjeta NI myRIO lista para iniciar la aplicación de ordenador

Aunque se vaya a utilizar la aplicación para hacer pruebas únicamente sobre modelos matemáticos (simulación), es imprescindible conectar la tarjeta.

## 4. Carga de la aplicación en tiempo real

La aplicación en tiempo real, que contiene el programa que realiza el control en tiempo real sobre la maqueta, se debe cargar en la tarjeta NI myRIO que se vaya a utilizar. Si se utiliza la tarjeta denominada “NI-myRIO-FAF”, que se utilizó para desarrollar las aplicaciones, la aplicación ya se encuentra cargada y empezará a ejecutarse cuando se alimente la tarjeta.

Sin embargo, en el caso de usarse otra tarjeta NI myRIO o si por cualquier causa se haya eliminado la aplicación en tiempo real de la tarjeta original, será necesario cargar la aplicación en tiempo real en la tarjeta.

Para ello, existen modos de carga que no necesitan del programa LabVIEW de National Instruments, que se pueden encontrar en la página web de esta empresa. Sin embargo, dado que National Instruments recomienda que se haga desde LabVIEW y que, con cualquier tarjeta NI myRIO se suministra una licencia del software y el propio software, se explica a continuación cómo hacerlo desde este entorno.

### Solution

If you have a LabVIEW executable file and do not have the LabVIEW Development Environment, you can still deploy your startup executable to PharLap, VxWorks, and Linux targets through one of the available solutions below.

1. If you are using identical hardware you can deploy the executable on a duplicate image via [the Replication and Deployment \(RAD\) Utility](#). Instructions for how to install and use RAD are included in the previous link.
- 2) If the executable has been compiled for the correct target, you can FTP the file onto your real-time controller:
  - Place the executable in the `c:\ni-rt\startup\` directory.
  - Set the following two settings in the `NI-RT.ini` file or `lvrt.conf` file on the target:

```
RTTarget.ApplicationPath=c:\ni-rt\startup\startup.rtexe  
RTTarget.LaunchAppAtBoot=True
```

where `startup.rtexe` is the name of your executable file.

### Additional Information

While this method works, the recommended method for deploying a LabVIEW Real-Time startup executable is to [do so through the LabVIEW Project](#).

*Figura 9. Alternativas que ofrece National Instruments para la carga de una aplicación en NI myRIO*

Primeramente, tras alimentar la tarjeta y conectarla vía USB al ordenador, se debe abrir, haciendo doble clic, el proyecto en LabVIEW, llamado *Aplicacion.lvproj*, que se encuentra ubicado en la carpeta *Aplicacion* que se suministra al usuario.

Nombre	Fecha de modifica...	Tipo	Tamaño
App	14/09/2018 9:54	Carpeta de archivos	
documentation	05/07/2018 12:21	Carpeta de archivos	
Instalador	14/09/2018 10:59	Carpeta de archivos	
RT App	14/09/2018 12:43	Carpeta de archivos	
Aplicacion.aliases	19/09/2018 20:31	Archivo ALIASES	1 KB
Aplicacion.lvps	19/09/2018 20:31	Archivo LVLPS	1 KB
Aplicacion.lvps	04/07/2018 13:31	Archivo LVLPS	2 KB
Aplicacion.lvproj	19/09/2018 20:31	LabVIEW Project	61 KB
Carro PID Analisis	17/09/2018 11:13	LabVIEW Instrume	346 KB
Carro PID Monit	19/09/2018 20:16	LabVIEW Instrume	122 KB
Carro PID RT.vi	19/09/2018 20:31	LabVIEW Instrume	247 KB
Carro Prep.vi	19/09/2018 10:56	LabVIEW Instrume	271 KB
Carro SS Analisis.vi	17/09/2018 11:09	LabVIEW Instrume	384 KB
Carro SS Monitorizacion.vi	19/09/2018 20:16	LabVIEW Instrume	125 KB
Carro SS RT.vi	19/09/2018 8:42	LabVIEW Instrume	255 KB
HMI.vi	19/09/2018 10:17	LabVIEW Instrume	528 KB
Inicio.vi	19/09/2018 8:43	LabVIEW Instrume	119 KB

Figura 10. Aplicación de LabVIEW “Aplicacion.lvproj”

En la ventana de proyecto, y bajo el directorio *Build Specifications* de la tarjeta NI myRIO, se encuentran las especificaciones para la aplicación en tiempo real, con el nombre *Control en Tiempo Real*. Se debe hacer en ésta clic con el botón derecho del ratón y seleccionar la opción *Build*, para compilar el programa.

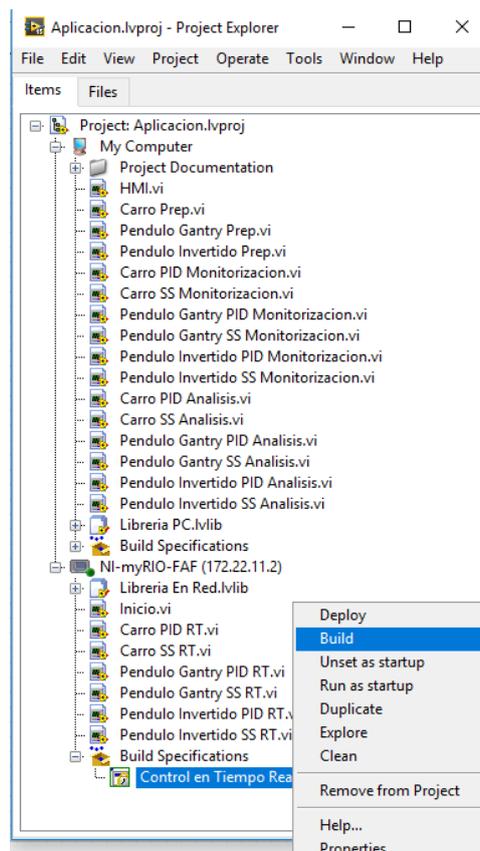


Figura 11. Compilación de la aplicación en tiempo real

La carga también se hace desde la ventana de proyecto en LabVIEW, haciendo clic derecho de nuevo sobre *Control en Tiempo Real*, la aplicación ya creada, y

seleccionando la opción *Run as startup*, para que el programa se ejecute cada vez que se inicie la tarjeta.

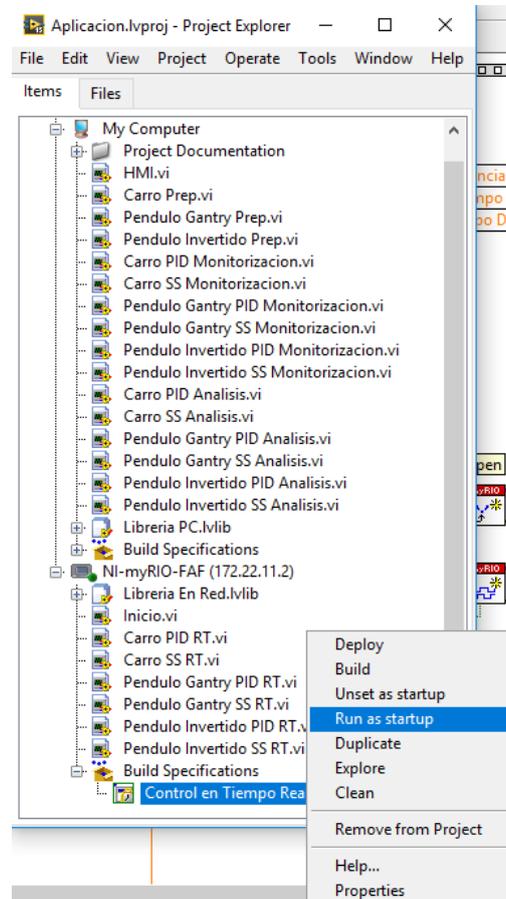


Figura 12. Carga de la aplicación en tiempo real

Tras la carga, aparece un cuadro de diálogo que indica que es necesario resetear la tarjeta, en el que se pulsa la opción Yes.

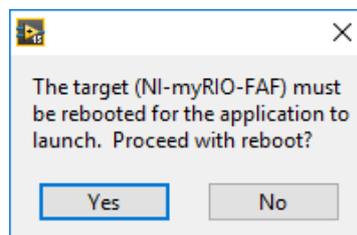


Figura 13. Cuadro de diálogo para el reinicio de la tarjeta NI myRIO

Una vez reiniciada la tarjeta, el programa se comenzará a ejecutar, y lo hará cada vez que ésta se inicie.

En caso de que se quiera cargar directamente con uno de los otros métodos, se suministra la aplicación en tiempo real ya compilada en la carpeta *Aplicacion > RT App > home > lvuser > natinst > bin* con el nombre de *startup.rtexe*, necesario para que al ser cargada se ejecute cada vez que se inicie la tarjeta.

## **5. Instalación de la aplicación para ordenador**

La aplicación se suministra en la carpeta *Aplicacion >App* con el nombre *Sistema de Análisis de Controladores Avanzados para la Maqueta IP02 de Quanser.exe*. Se puede lanzar directamente haciendo doble clic en ella, siempre y cuando se disponga en el ordenador de LabVIEW.

Si el ordenador en el que se quiere ejecutar la misma **NO** dispone de LabVIEW, se debe recurrir al instalador de la aplicación, ubicado en *Aplicacion > Instalador > Volume*, con el nombre *setup.exe*. Éste se encargará de, además de instalar la aplicación, instalar el software necesario para el correcto funcionamiento de la misma sin LabVIEW.

Se debe hacer doble clic en el archivo para lanzar el instalador. Sólo es necesario seguir las instrucciones del mismo para completar la instalación. Si el software necesario que se ha mencionado ya estuviera instalado en el ordenador, el instalador no lo instalará.

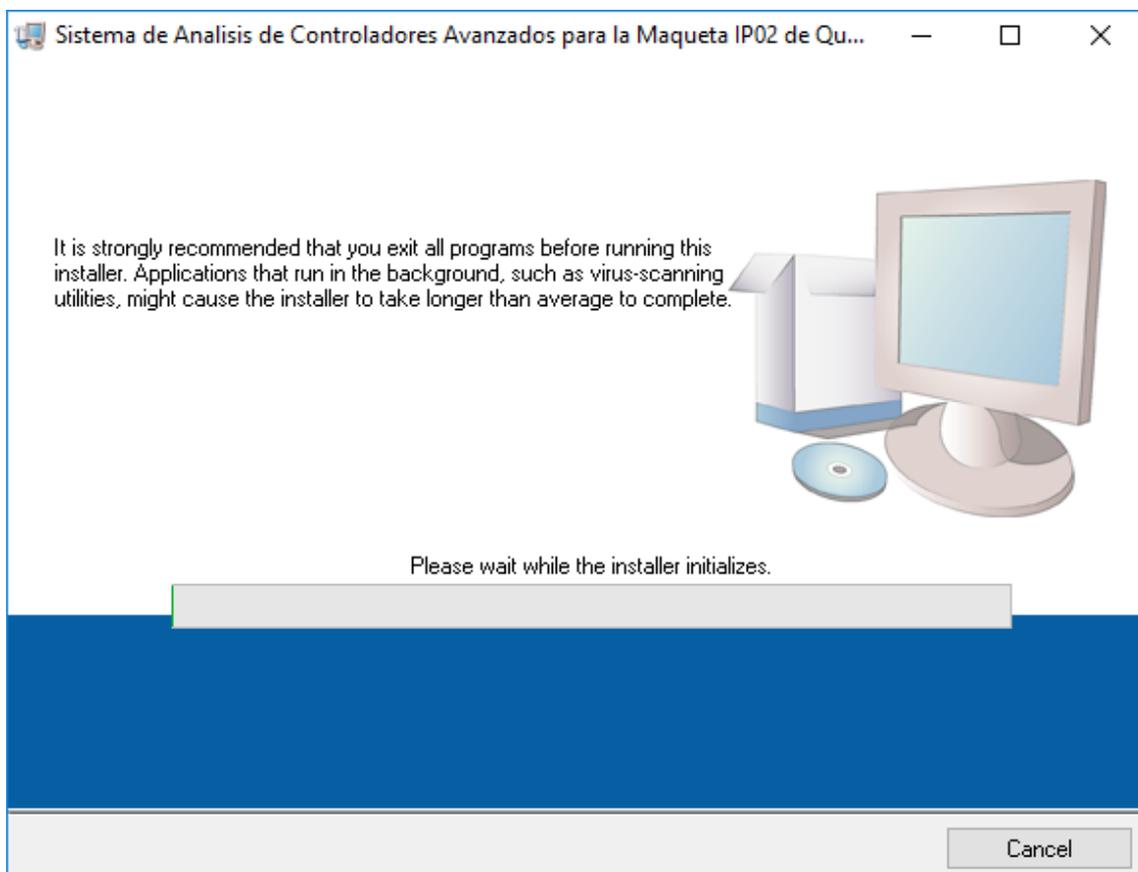


Figura 14. Ventana del instalador de la aplicación

Por defecto, la aplicación se instala en la carpeta *Archivos de programa > Sistema de Análisis de Controladores Avanzados para la Maqueta IP02 de Quanser*, aunque el destino se puede cambiar durante la instalación.

Se crearán dos accesos directos durante la instalación, uno en la carpeta de destino y otro en el escritorio.

## **6. Preparación de la maqueta IP02 de Quanser**

Para poder utilizar la maqueta IP02, es necesario alimentar y encender el módulo UPM, disponible en el laboratorio, que se muestra en la figura siguiente. Este módulo sirve para amplificar la potencia de la señal de control. El conector de la señal de control saliente de la tarjeta se conectará al conector señalado en rojo en la figura inferior mediante cable de conector de cinco pines a conector coaxial. Del conector marcado en azul se enviará la señal de control amplificada en potencia hacia el motor mediante cable de conector de seis pines a conector de cuatro pines.



*Figura 15. Conectores del módulo UPM*

El conector de la señal de control saliente de la tarjeta es el conector concéntrico que se muestra en la figura siguiente.



*Figura 16. Conector de la señal de control saliente de la tarjeta NI myRIO*

Por otro lado, los conectores de los encoders en la tarjeta tienen la forma que se muestra en la figura siguiente.



Figura 17. Conector de encoder en la tarjeta NI myRIO

Cada uno de los conectores de los encoders en la tarjeta tiene un identificador, un 0 para el “encoder0” o encoder del carro y un 1 para el “encoder1” o encoder del péndulo. Ocurre lo mismo en la maqueta. Se deberán conectar mediante cable de cinco pines los dos terminales de cada encoder, cero con cero y uno con uno. El terminal restante en la figura inferior, situado a la derecha, es el de alimentación al motor.

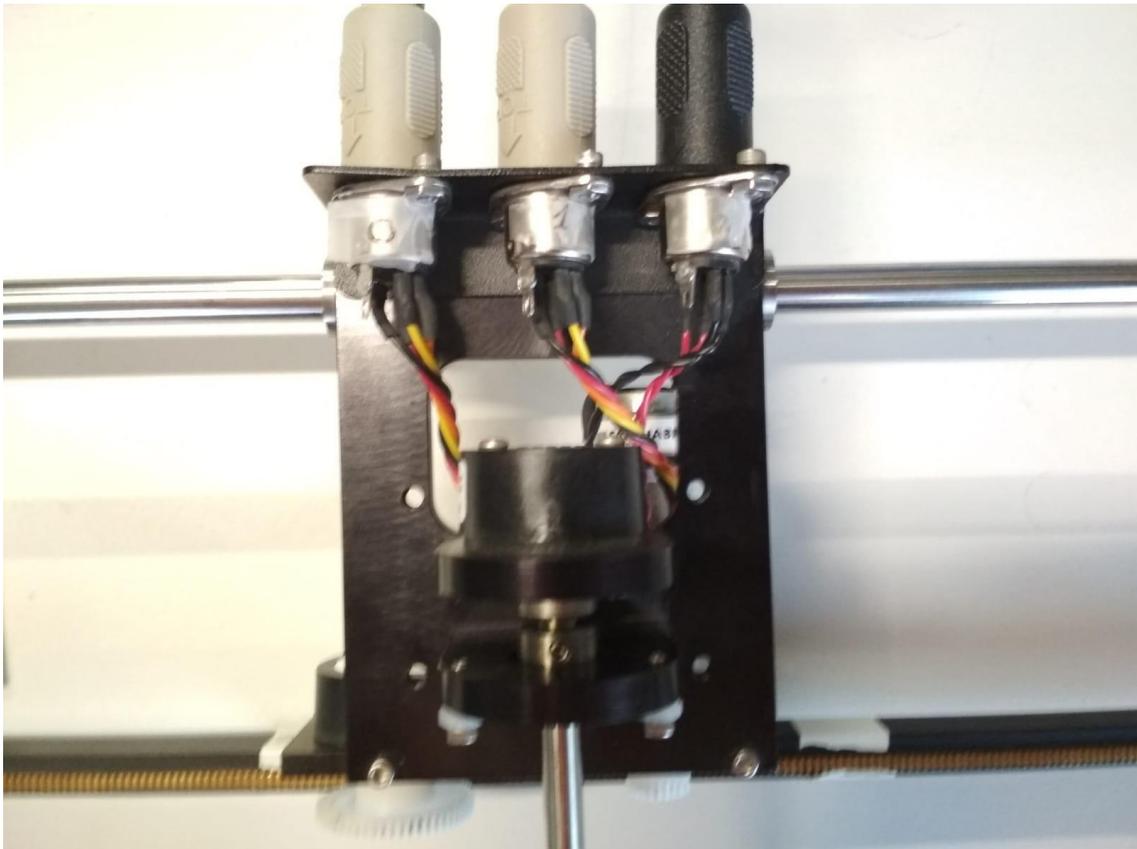


Figura 18. Conectores en la maqueta IP02 de Quanser

Todos los cables están disponibles en el laboratorio, identificados como pertenecientes a la maqueta IP02 correspondiente.

Es importante fijar bien la pista a la superficie de apoyo, evitando que deslice sobre ésta. Para ello, se recomienda incluso poner peso en los extremos, ya que acciones de control muy ruidosas pueden hacer vibrar la pista, y por lo tanto provocar un

desplazamiento de ésta. También se recomienda calzar el apoyo que se encuentra hacia la mitad de la pista, ya que es posible que no haga bien contacto con la superficie de apoyo y, por lo tanto, el piñón motriz del carro no engrane bien en la cremallera de la pista.

Este último detalle puede resultar crítico en controles como los del sistema péndulo invertido. Además, no hacerlo influirá negativamente en el desgaste del piñón motriz.

## 7. Uso de la aplicación para ordenador

Con la maqueta IP02 ya preparada, lo primero que se debe hacer antes de comenzar a usar la aplicación, como ya se ha indicado antes, es alimentar y conectar la tarjeta con el programa ya cargado, y esperar a que se encienda el LED indicador LED0 de la misma, que confirma que el programa está en marcha.

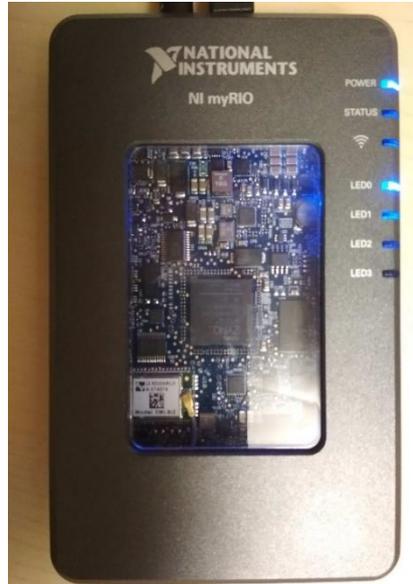


Figura 19. Tarjeta NI myRIO lista para iniciar la aplicación de ordenador

Una vez esto ocurre, se puede lanzar la aplicación *Sistema de Análisis de Controladores Avanzados para la Maqueta IP02 de Quanser.exe*. Es posible que, al iniciarla, se muestre en pantalla el siguiente mensaje.

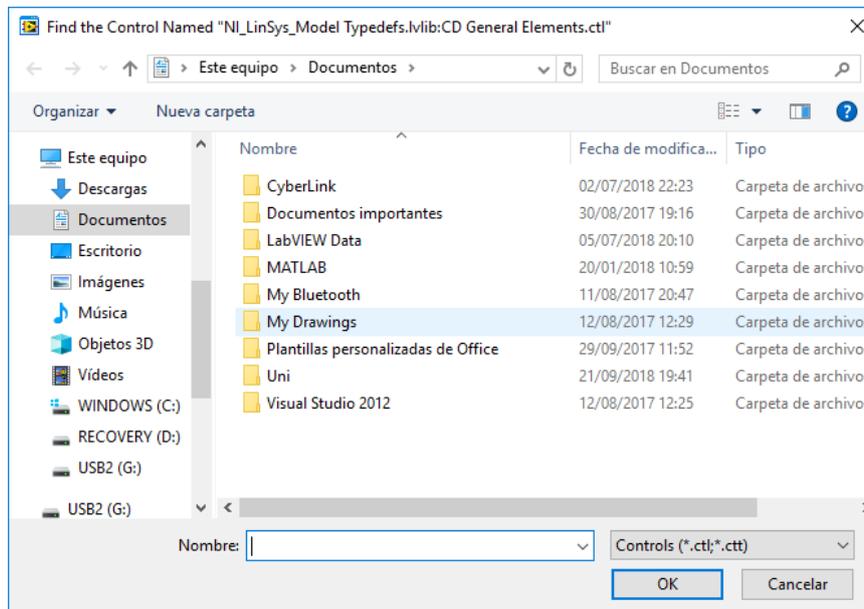


Figura 20. Posible cuadro de diálogo al iniciar la aplicación

Simplemente es necesario pulsar el botón “Cancelar”, y la aplicación se iniciará a continuación.

## 7.1. Inicio

Al iniciar la aplicación, el usuario se encuentra en la pantalla inicial, que da a elegir entre los tres sistemas que se pueden modelar con la maqueta IP02 de Quanser. Pulsando cualquiera de los tres, se avanza a una pantalla de preparación de la prueba sobre el sistema correspondiente. Además, existe un botón “Salir”, que permite terminar y cerrar la aplicación.



Figura 21. Pantalla de inicio de la aplicación

## 7.2. Preparación de la prueba

La siguiente pantalla será una de las tres que se muestran a continuación, dependiendo del sistema elegido por el usuario para las pruebas.

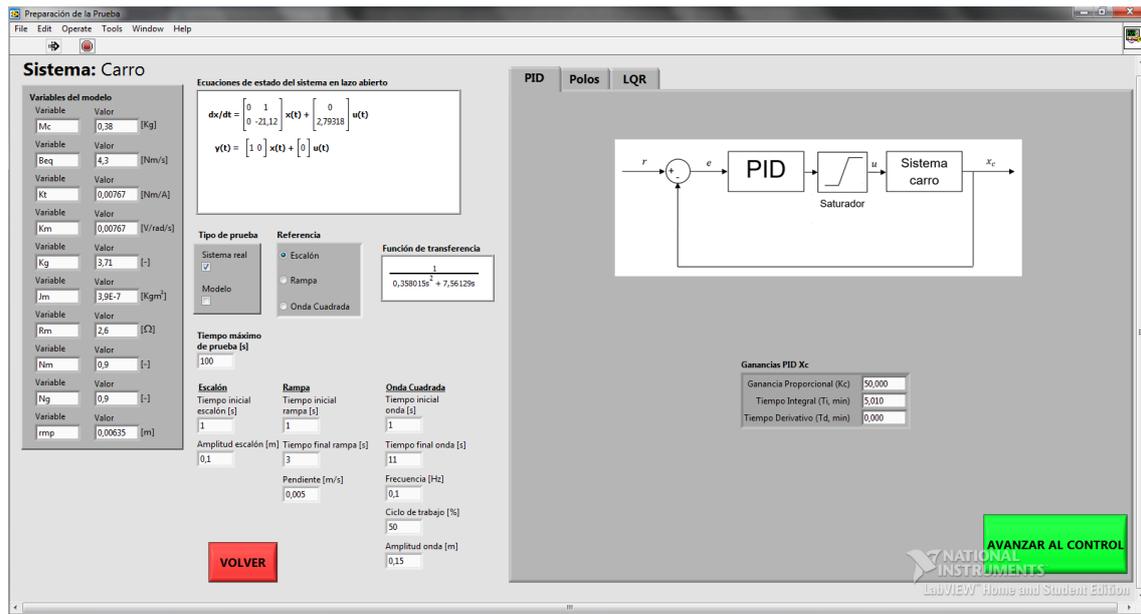


Figura 22. Pantalla de preparación de la prueba para el sistema carro

## **7.2.1. Parámetros del modelo**

En la parte izquierda, se encuentran los valores que definen el modelo matemático del sistema elegido, que puede modificar el usuario. Los valores predeterminados con los que se inicia la aplicación son los correspondientes al carro sin añadir la masa extra y al péndulo corto, en el caso del péndulo Gantry y del péndulo invertido. Los valores por defecto propuestos por el desarrollador son los incluidos en el *ANEXO II: PARÁMETROS DEL SISTEMA*.

En la parte superior central, se muestran las ecuaciones de estado del sistema y, en el caso del sistema carro, su función de transferencia, calculadas a partir de los valores introducidos por el usuario para el modelo.

## **7.2.2. Parámetros de prueba**

Además, el usuario debe seleccionar si hacer la prueba sobre el sistema real, sobre el modelo o sobre ambos, marcando las opciones deseadas. Es necesario marcar al menos una opción. De lo contrario, el programa no permitirá el avance a la siguiente pantalla y, al intentarlo, dará el siguiente mensaje de error.

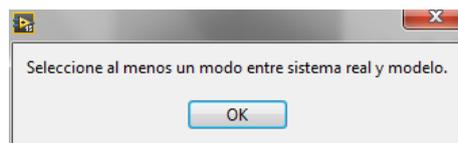


Figura 23. Mensaje de error al no seleccionar ningún tipo de prueba

Justo bajo la selección del tipo de prueba, se debe especificar el tiempo máximo de la misma. Este tiempo determina cuándo se detendrá una prueba en tiempo real automáticamente, aunque el usuario no dé la orden de parada y, si únicamente se selecciona una prueba sobre el modelo, el tiempo final de esta prueba. Si se seleccionan ambos, el modelo se simulará hasta el tiempo final de la prueba en tiempo real.

Así mismo, el usuario debe seleccionar la referencia que desea introducir al sistema en la prueba, pudiendo elegir entre una referencia tipo escalón, una tipo rampa o una tipo onda cuadrada. Se debe tener en cuenta que el origen de la pista para la medida del desplazamiento del carro comienza en el centro de la misma.

Si se quiere referencia nula, se recomienda poner el valor de amplitud del escalón a cero. Más abajo, se pueden seleccionar los parámetros de cada tipo de referencia.

En el caso del escalón, hay que definir el tiempo en que se produce y el valor de su amplitud.

Para la rampa, que empezará siempre con un valor amplitud nulo, hay que definir su tiempo inicial, su tiempo final y su pendiente. Si se quiere un valor final concreto, es fácil calcularlo de la siguiente manera:  $Valor\ Final = m (t_f - t_i)$ . Si se intenta avanzar habiendo seleccionado esta referencia con pendiente cero, el programa devolverá el siguiente mensaje de error.

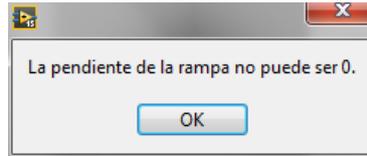


Figura 24. Mensaje de error al seleccionar una pendiente igual a cero

Por último, la onda cuadrada será entre cero y un valor de amplitud seleccionado. Es necesario que se definan los tiempos inicial y final de la onda, su frecuencia, el ciclo de trabajo (Duty Cycle), que es el porcentaje del periodo que la onda toma el valor no nulo de amplitud, y dicho valor de amplitud.

Si no existe congruencia entre tiempos iniciales y finales de la referencia seleccionada, o entre éstos y el tiempo máximo de la prueba, no se podrá avanzar a la siguiente pantalla y, al intentarlo, se mostrará el siguiente mensaje de error.

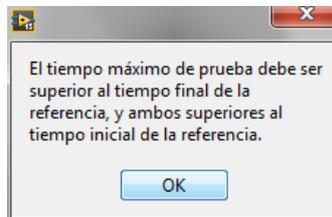


Figura 25. Mensaje de error por incongruencia en los tiempos

### **7.2.3. Selección de controlador y parámetros**

Por otro lado, en la parte derecha de la pantalla, se encuentran los controladores disponibles para ser implementados en el sistema elegido. En los tres casos, se pueden seleccionar haciendo clic en la pestaña correspondiente, un controlador con algoritmos PID, otro con realimentación del estado por asignación de polos e integración y con realimentación del estado por LQR e integración. En la parte inferior derecha de los tres, se encuentra el botón que permite avanzar a la prueba seleccionada.

Todos los controles de esta aplicación incluyen un saturador entre -8V y 8V para garantizar que el motor de DC del carro no sea dañado por acciones de control excesivamente altas.

Además, los controladores implementados sobre el péndulo invertido, debido a que habitualmente se generan acciones de control muy ruidosas por la inestabilidad inherente del sistema, incluyen un filtro pasa bajos de 40Hz previo al saturador mencionado.

#### **7.2.3.1. Controladores PID**

Para el control basado en algoritmos PID, se muestra en la pantalla correspondiente el esquema del mismo para cada sistema. Debajo de éste, se pueden seleccionar los parámetros que definen el o los algoritmos PID presentes en el controlador. El algoritmo se define en este programa por la ganancia proporcional  $K_c$ , el tiempo integral  $T_i$  y el tiempo derivativo  $T_d$ , según la ecuación que se muestra a continuación.

$$u(t) = K_c e(t) + \frac{K_c}{T_i} \int_0^t e(t) dt + K_c T_d \frac{de(t)}{dt} \quad (1)$$

Por supuesto, estos tiempos no podrán ser negativos, aunque sí pueden asignarse valores como cero o infinito.

### 7.2.3.2. Controladores por asignación de polos

En el control de realimentación del estado por asignación de polos con integración, de nuevo se muestra en la pantalla el esquema del mismo en cada sistema. Debajo de ese diagrama, se pueden introducir los polos en lazo cerrado deseados para definir la dinámica del sistema y la aplicación calculará las ganancias, de realimentación del estado  $K$  e integral  $K_i$ , para el sistema correspondiente. Más abajo, se muestran las ecuaciones de estado del sistema ampliado, en el que se incluye la integral del error de la variable realimentada.

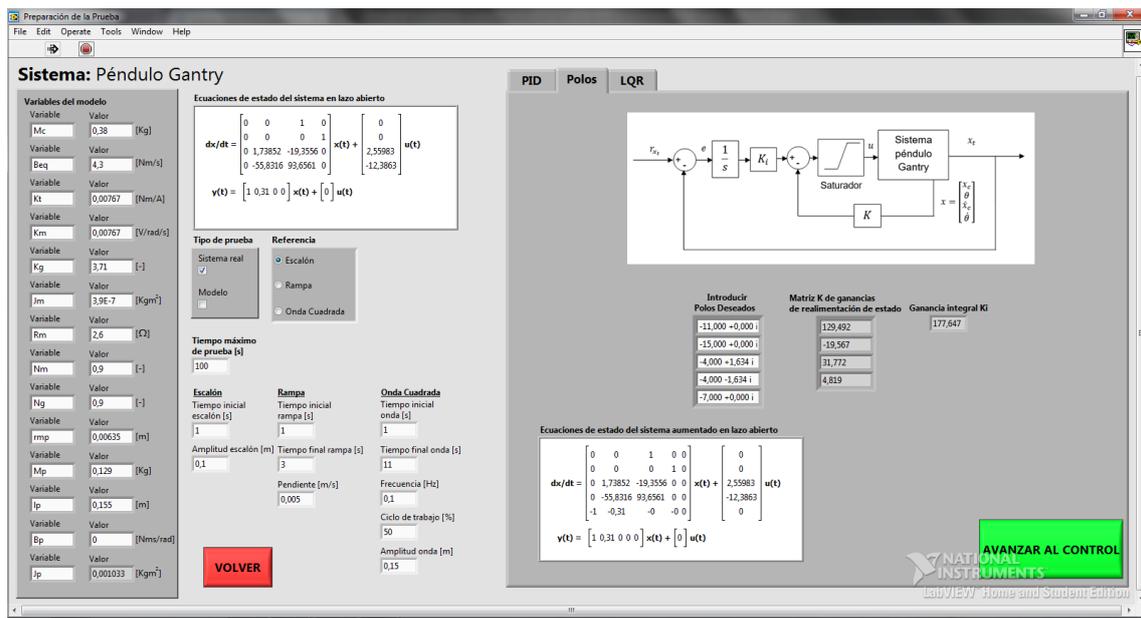


Figura 26. Pantalla de preparación de la prueba para el sistema péndulo Gantry

La única restricción respecto a los polos elegidos es que los polos complejos deben ir acompañados de su conjugado. Si esta condición no se cumple, al intentar avanzar se mostrará el siguiente mensaje de error.

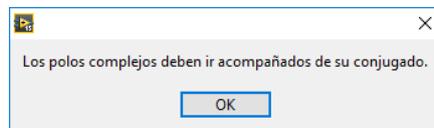


Figura 27. Mensaje de error al no incluir el conjugado de un polo complejo

Sin embargo, se recuerda que, aunque programáticamente están permitidos, los polos en el semiplano derecho, es decir, con parte real positiva, harán que los sistemas sean inestables.

### 7.2.3.3. Controladores LQR

La pestaña del controlador de realimentación del estado por LQR es muy similar a la del controlador por asignación de polos. En la parte superior de las pantallas correspondientes se muestra el esquema del controlador de cada sistema, mientras que en la parte inferior se dan las ecuaciones de estado del sistema ampliado, incluyendo la integral del error de la variable realimentada. Entre ambos, se

encuentran las matrices  $Q$  y  $R$  que el usuario definirá para que la aplicación calcule las ganancias, de realimentación del estado  $K$  e integral  $K_i$ .

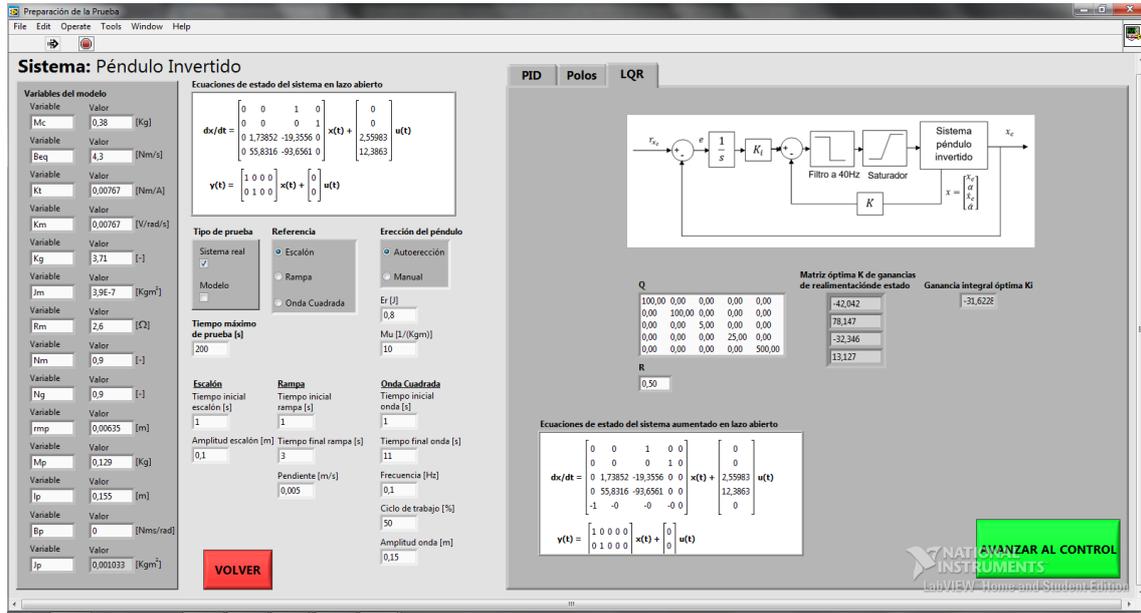


Figura 28. Pantalla de preparación de la prueba para el sistema péndulo invertido

Las matrices  $Q$  y  $R$ , que sirven para ponderar el error del estado y el valor de la acción de control respectivamente, deben ser ambas simétricas. Además,  $R$  debe ser definida positiva, por lo que el valor más pequeño permitido para ella es de 0,01. Por otro lado,  $Q$  también debe ser semidefinida positiva, condición que, si se incumple y se intenta avanzar al control, hará que se muestre el siguiente mensaje de error.

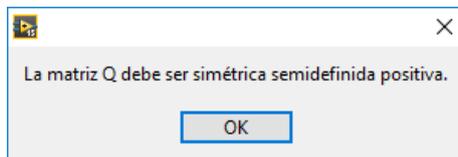


Figura 29. Mensaje de error al seleccionar una matriz  $Q$  que no es simétrica semidefinida positiva

### 7.2.4. Restricciones de los valores de los controles

Todos los valores de los controles se reinician a sus valores predeterminados al iniciar la aplicación. Sin embargo, entre distintas pruebas para el mismo sistema se guardan los últimos valores seleccionados por el usuario.

Para finalizar, en la tabla siguiente, se muestran las restricciones aplicadas a los valores de los controles que puede introducir el usuario.

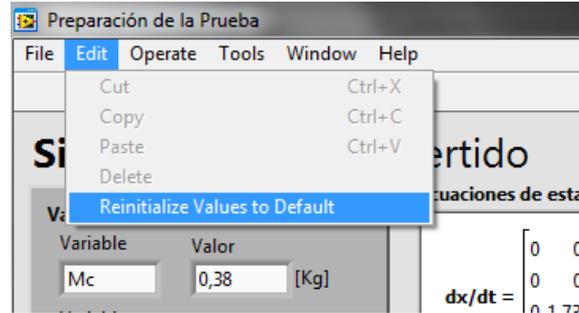
Control	Restricciones	Respuesta ante error
$M_c$	$0 < M_c < \infty$	Valor límite
$B_{eq}$	$0 \leq B_{eq} < \infty$	Valor límite
$K_t$	$0 < K_t < \infty$	Valor límite
$K_m$	$0 < K_m < \infty$	Valor límite
$K_g$	$0 < K_g < \infty$	Valor límite
$J_m$	$0 < J_m < \infty$	Valor límite
$R_m$	$0 < R_m < \infty$	Valor límite

$N_m (\eta_m)$	$0 < N_m \leq 1$	Valor límite
$N_g (\eta_g)$	$0 < N_g \leq 1$	Valor límite
$r_{mp}$	$0 < r_{mp} < \infty$	Valor límite
$M_p$	$0 < M_p < \infty$	Valor límite
$l_p$	$0 < l_p < \infty$	Valor límite
$B_p$	$0 \leq B_p < \infty$	Valor límite
Tiempo máximo de prueba [s]	$0 \leq T_{max} < \infty$	Valor límite
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Tiempo inicial escalón [s]	$0 \leq T_i < \infty$	Valor límite
	$T_i < T_{max}$	Mensaje de error al intentar avanzar
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Amplitud escalón [m]	$-0,35 \leq A \leq 0,35$	Valor límite
Tiempo inicial rampa [s]	$0 \leq T_i < \infty$	Valor límite
	$T_i < T_{max}$	Mensaje de error al intentar avanzar
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Tiempo final rampa [s]	$0 \leq T_f < \infty$	Valor límite
	$T_f \leq T_{max}$	Mensaje de error al intentar avanzar
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Pendiente [m/s]	$m \neq 0$	Mensaje de error al intentar avanzar
	$-0,2 \leq m \leq 0,2$	Valor límite
Valor final rampa [m]	$ m (T_f - T_i)  \leq 0,35$	Mensaje de error al intentar avanzar
Tiempo inicial onda [s]	$0 \leq T_i < \infty$	Valor límite
	$T_i < T_{max}$	Mensaje de error al intentar avanzar
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Tiempo final onda [s]	$0 \leq T_f < \infty$	Valor límite
	$T_f \leq T_{max}$	Mensaje de error al intentar avanzar
	<i>Incremento mínimo = 0,002</i>	Valor más cercano
Frecuencia [Hz]	$0 \leq f \leq 50$	Valor límite
Ciclo de trabajo [%]	$0 \leq D \leq 100$	Valor límite
Amplitud de onda [m]	$-0,35 \leq A \leq 0,35$	Valor límite
$E_r [J]$	$0 \leq E_r < \infty$	Valor límite
$\mu [1/(Kgm)]$	$0 \leq \mu \leq 100$	Valor límite
Tiempo Integral ( $T_i$ , min)	$0 \leq T_i < \infty$	Valor límite
Tiempo Derivativo ( $T_d$ , min)	$0 \leq T_d < \infty$	Valor límite
Introducir polos deseados	Los polos complejos deben ir acompañados de su conjugado	Mensaje de error al intentar avanzar
Q	Simétrica semidefinida positiva	Mensaje de error al intentar avanzar
R	$0,01 \leq R < \infty$ (Simétrica definida positiva)	Valor límite

Tabla 2. Restricciones de los parámetros de la prueba

### **7.2.5. Reinicio a los valores predeterminados**

Si, durante la preparación de la prueba, se quieren restablecer los valores predeterminados de una de las pantallas de preparación, se debe hacer clic en la pestaña “Edit” que se encuentra en la parte superior izquierda de la pantalla y seleccionar “Reinitialize Values to Default”, como se muestra en la figura siguiente.



*Figura 30. Reinicio de los parámetros a su valor predeterminado*

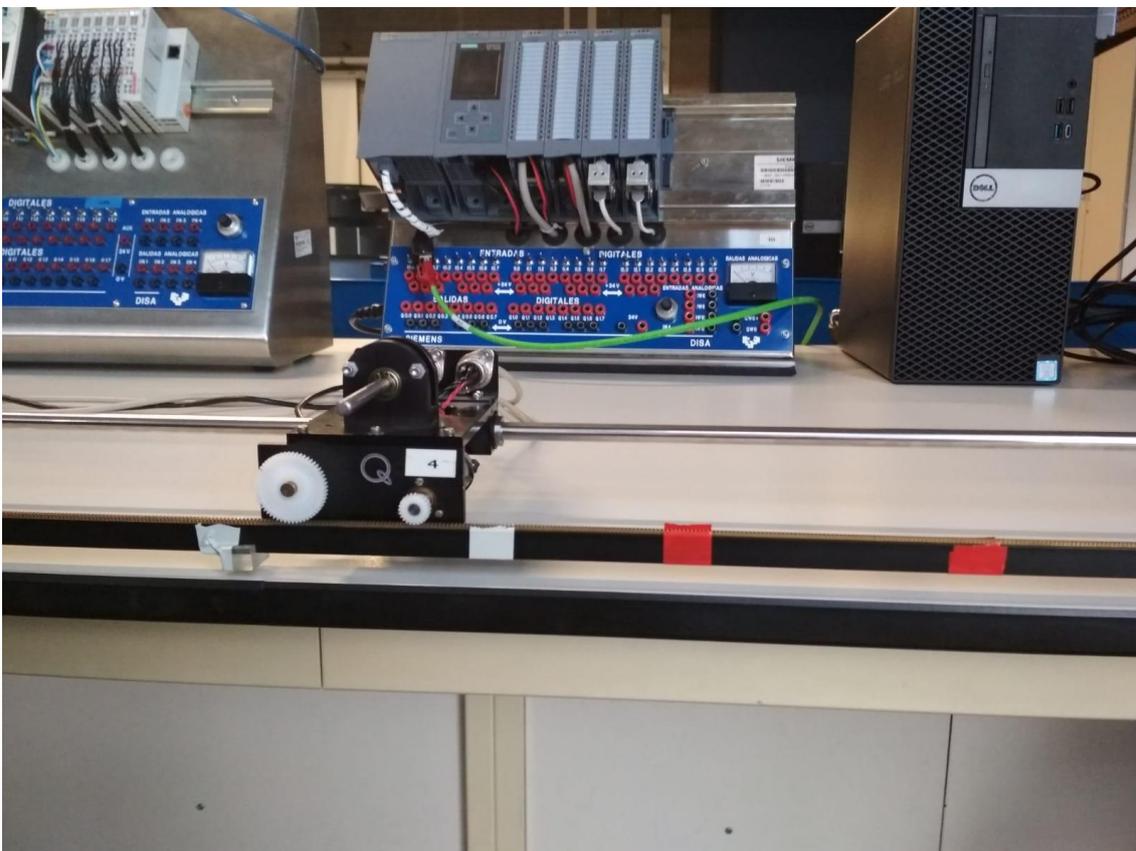
Se ha de tener en cuenta que se restablecerán todos los valores de la pantalla correspondiente.

## 7.3. Control en tiempo real y monitorización

Una vez seleccionados los parámetros de la prueba en la ventana anterior, ésta estará lista para llevarse a cabo. Sin embargo, si se ha seleccionado actuar sobre el sistema real, **antes de presionar el botón de “Avanzar a Control”**, es necesario preparar la maqueta para la prueba correspondiente.

Llegados a este punto, además de las consideraciones ya comentadas en el apartado 5. *Preparación de la maqueta IP02 de Quanser* de este documento, es necesario colocar adecuadamente el carro en el punto de inicio estipulado para cada prueba.

Para **todas las pruebas EXCEPTO** para el sistema péndulo invertido, la prueba se iniciará con el carro situado entre las marcas blancas de la maqueta, en el centro de la pista. La siguiente figura muestra el sitio exacto en la maqueta.



*Figura 31. Punto de inicio de la prueba para todos los sistemas excepto péndulo invertido*

Sin embargo, las pruebas con el sistema **péndulo invertido con péndulo largo** se llevarán a cabo colocando el carro entre las marcas rojas de la maqueta a su inicio, ligeramente a la derecha del centro de la pista, para permitir la correcta erección del mismo.

A pesar de que no sea necesario en ese aspecto, se iniciará la prueba con el carro en dicha posición, aunque se seleccione la erección manual. Sin embargo, se debe tener en cuenta que la referencia para el valor de las entradas al sistema (escalón, rampa...) sigue siendo el centro de la pista.

En la figura siguiente se muestra este segundo punto de inicio, marcado en rojo.

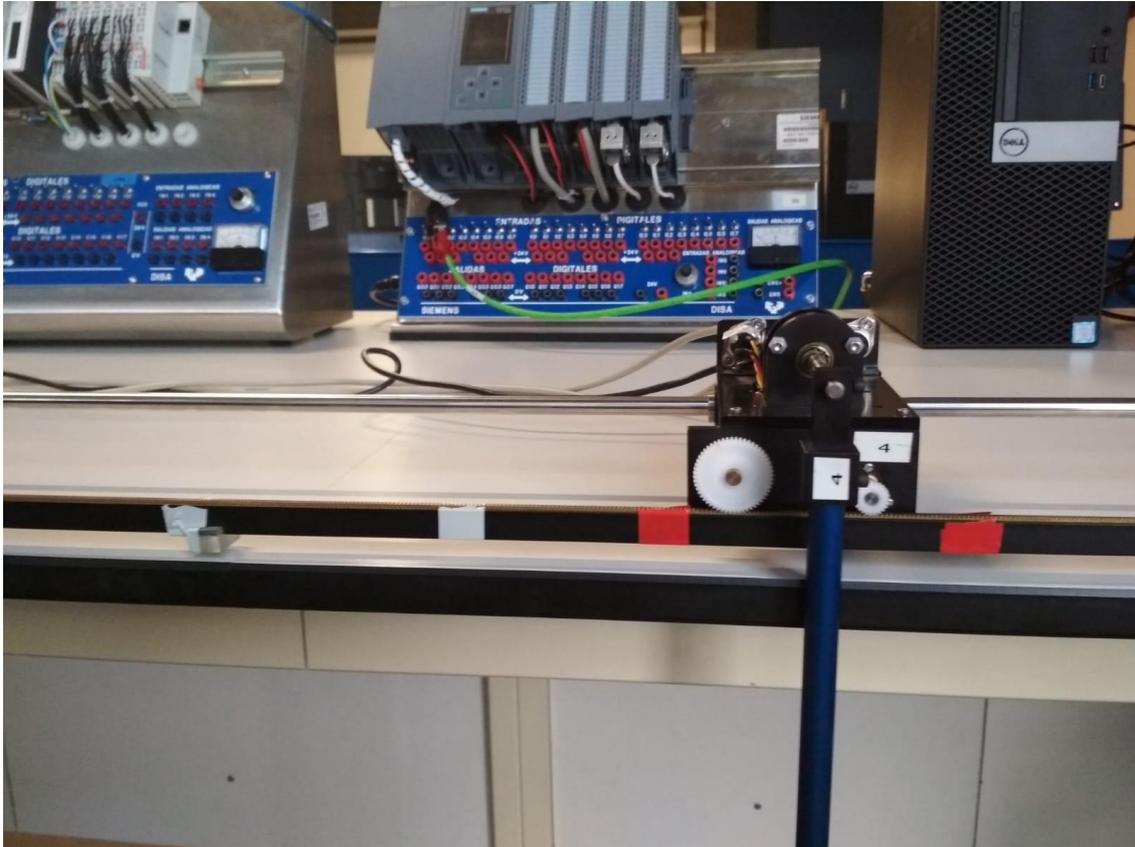


Figura 32. Punto de inicio de la prueba para el péndulo invertido con péndulo largo

Por otro lado, las pruebas con el sistema **péndulo invertido con péndulo corto** se llevarán a cabo colocando el carro entre las marcas verdes de la maqueta a su inicio, ligeramente a la izquierda del centro de la pista, para permitir la correcta erección del mismo.

De nuevo, aunque no sea necesario en ese aspecto, se iniciará la prueba con el carro en dicha posición, aunque se seleccione la erección manual, y ha de tenerse en consideración que la referencia para el valor de las entradas al sistema (escalón, rampa...) sigue siendo el centro de la pista.

En la figura siguiente se muestra este tercer punto de inicio, marcado en verde.

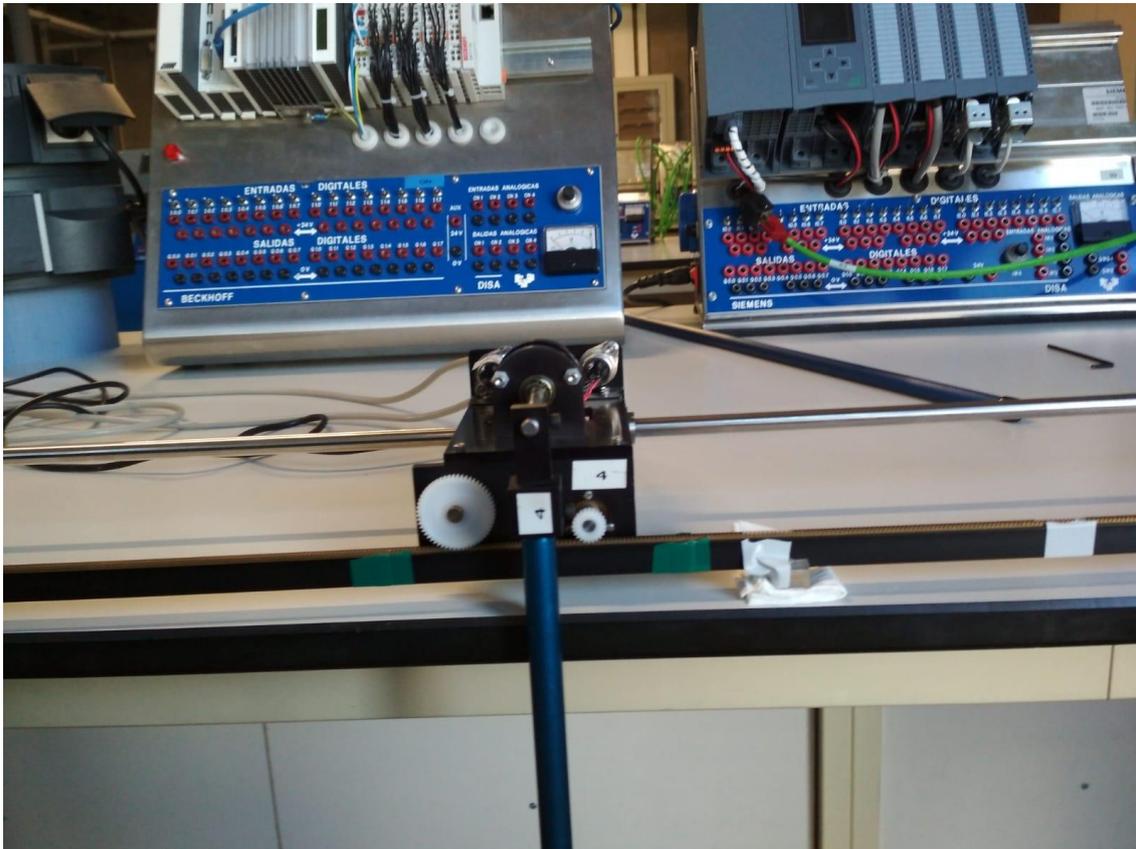


Figura 33. Punto de inicio de la prueba para el péndulo invertido con péndulo corto

Además, en todas las **pruebas en las que se use un péndulo**, éste deberá comenzar la prueba vertical hacia abajo y en reposo, como se muestra en la figura superior.

Una vez ubicado el carro en el lugar de inicio adecuado de la prueba, y el péndulo en reposo, se puede empezar la prueba pulsando el botón “Avanzar a Control”.

Se abrirá una pantalla de monitorización y control de la prueba. El LED1 de la tarjeta NI myRIO se encenderá, indicando que la tarjeta está lista para recibir la orden de inicio de la prueba.

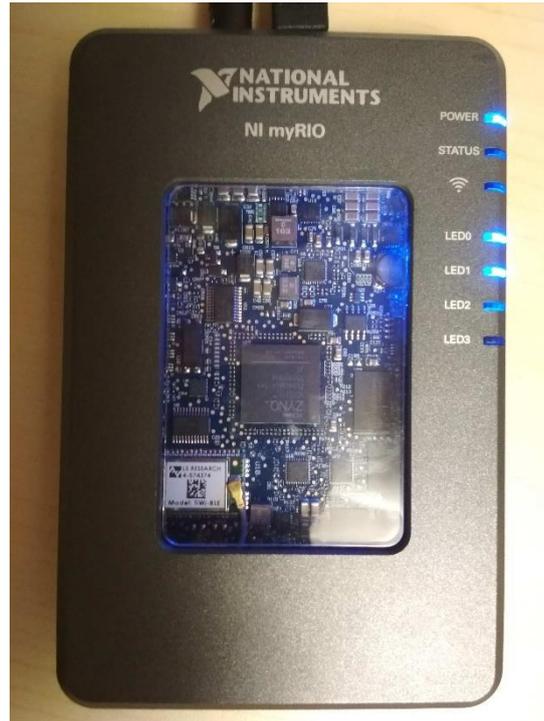


Figura 34. Tarjeta NI myRIO lista para iniciar la prueba en tiempo real

Para comenzar la prueba se deberá pulsar el botón “Empezar Simulación”, mientras que el resto de botones no estarán aún disponibles. Además, se mostrarán las ganancias, ya sean PID o de realimentación del estado, seleccionadas en el paso anterior.

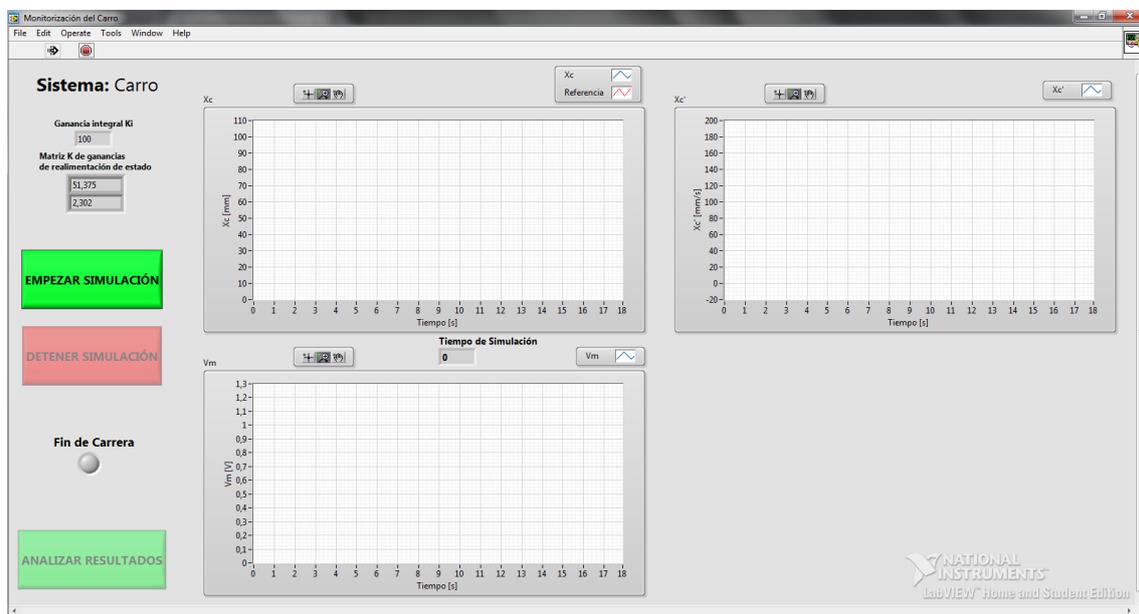


Figura 35. Pantalla de monitorización para el sistema carro esperando a empezar la prueba

Al iniciar la prueba, se encenderá el LED2 de la tarjeta NI myRIO, que indica que el control en tiempo real está en marcha.

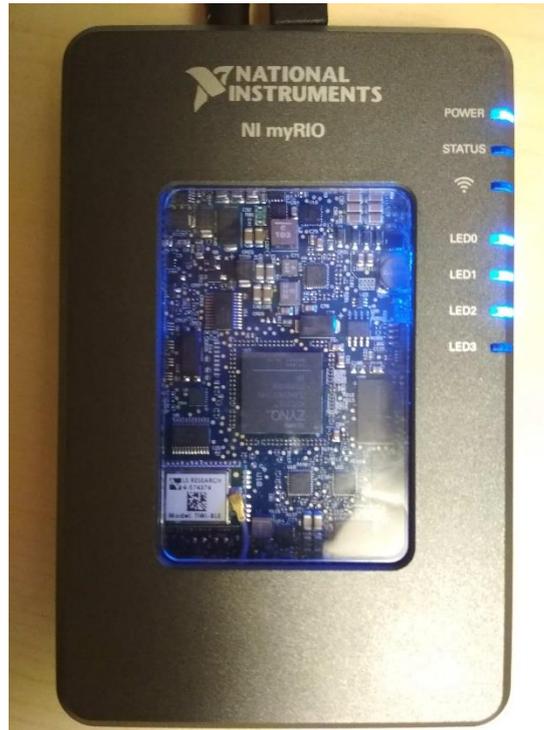


Figura 36. Tarjeta NI myRIO ejecutando el control en tiempo real sobre la maqueta IP02

Tras iniciar la prueba, se podrá visualizar en tiempo real la evolución de las variables del sistema en los gráficos que aparecen en la pantalla, además del tiempo de simulación o prueba, mientras la tarjeta está ejecutando el control en tiempo real sobre la maqueta. Aunque la representación es muy fiel, pueden existir pequeños desfases de tiempo, debido al tiempo de actualización de las variables. Por lo tanto, en la pantalla de análisis posterior a la prueba, se podrán ver las gráficas con los datos exactos registrados.

Cuando el usuario lo desee, podrá finalizar la prueba pulsando el botón “Detener Simulación”, mientras que los otros dos botones estarán desactivados.

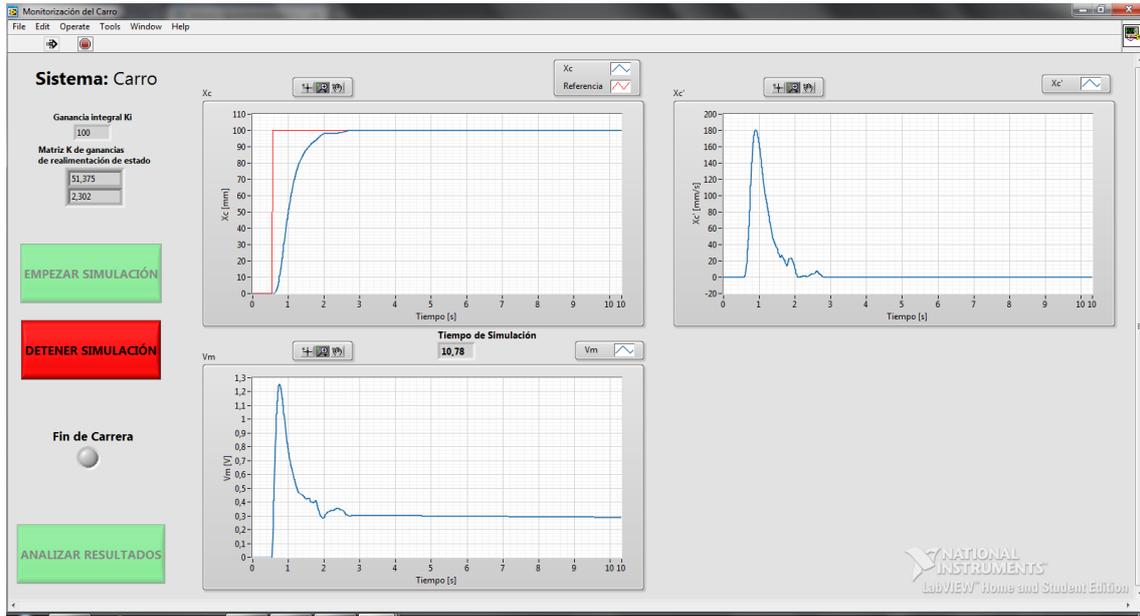


Figura 37. Pantalla de monitorización para el sistema carro durante la prueba

Cabe destacar que, en ciertos casos, debido al gran número de variables a mostrar, no se podrán mostrar todas a la vez en pantalla. En dichos casos, se podrá desplazar la pantalla con la barra inferior, para mostrar el resto de variables.

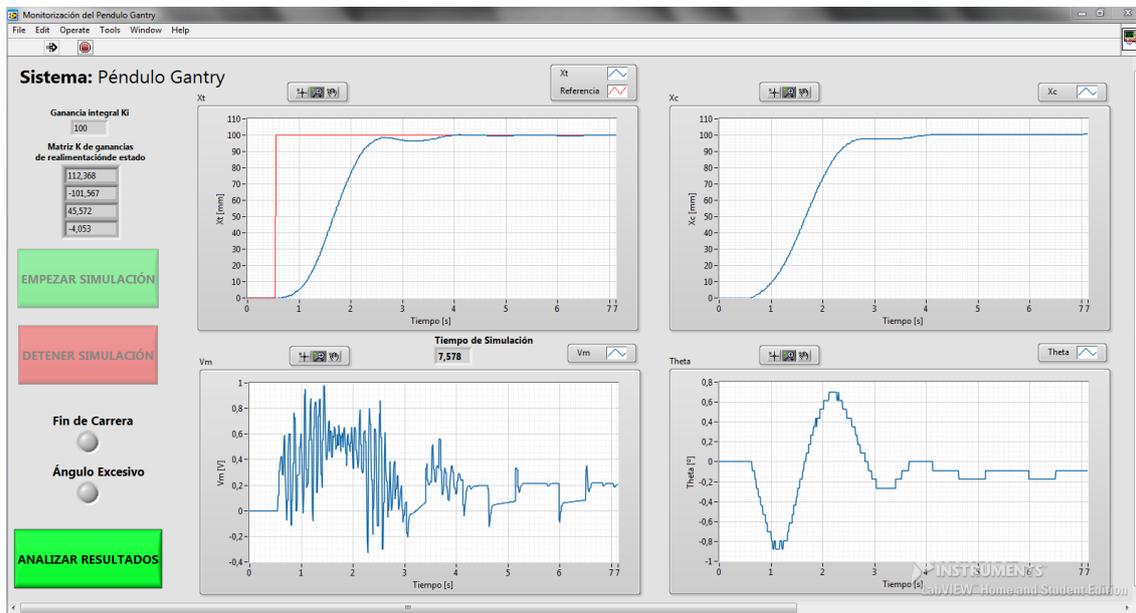


Figura 38. Pantalla de monitorización para el sistema péndulo Gantry I

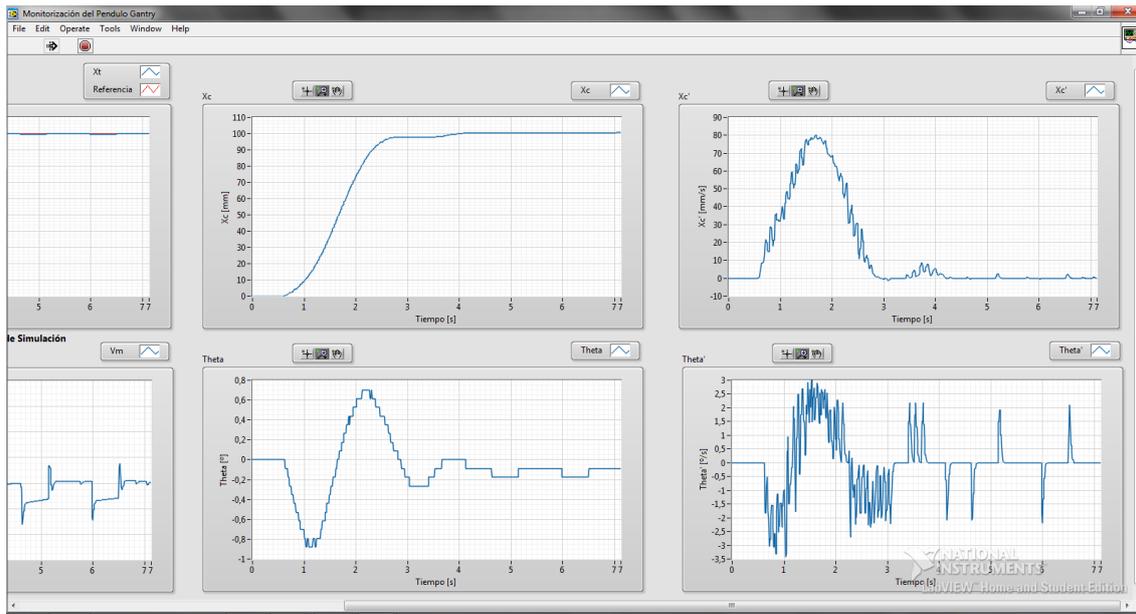


Figura 39. Pantalla de monitorización para el sistema péndulo Gantry II

Es posible que la prueba no vaya según lo esperado, caso frecuente a la hora de intentar sintonizar un controlador. Si el carro se sale de la carrera útil de la pista, la prueba se detendrá automáticamente, y se encenderá el indicador de “Fin de Carrera” en color rojo, como se muestra en la siguiente figura.

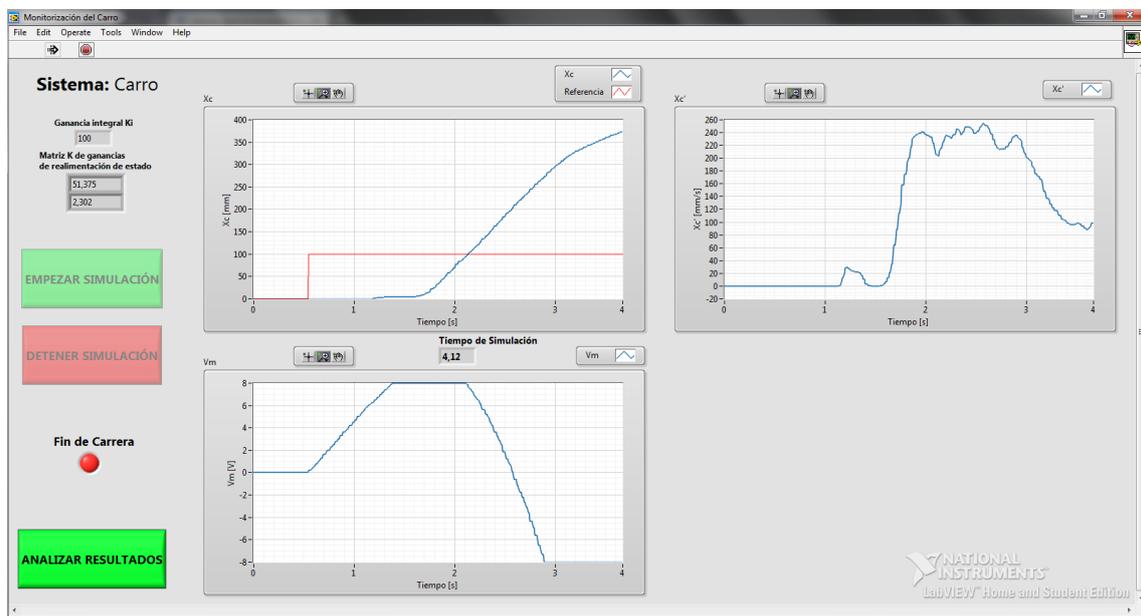


Figura 40. Indicador de fin de carrera

Lo mismo ocurre en el caso de sistemas con péndulo, para el caso de que el ángulo del mismo se considere excesivo o peligroso, siendo indicativo de que el control no va bien. En este caso, se encenderá en rojo el indicador de “Ángulo Excesivo” en la pantalla, antes de detenerse la prueba automáticamente.

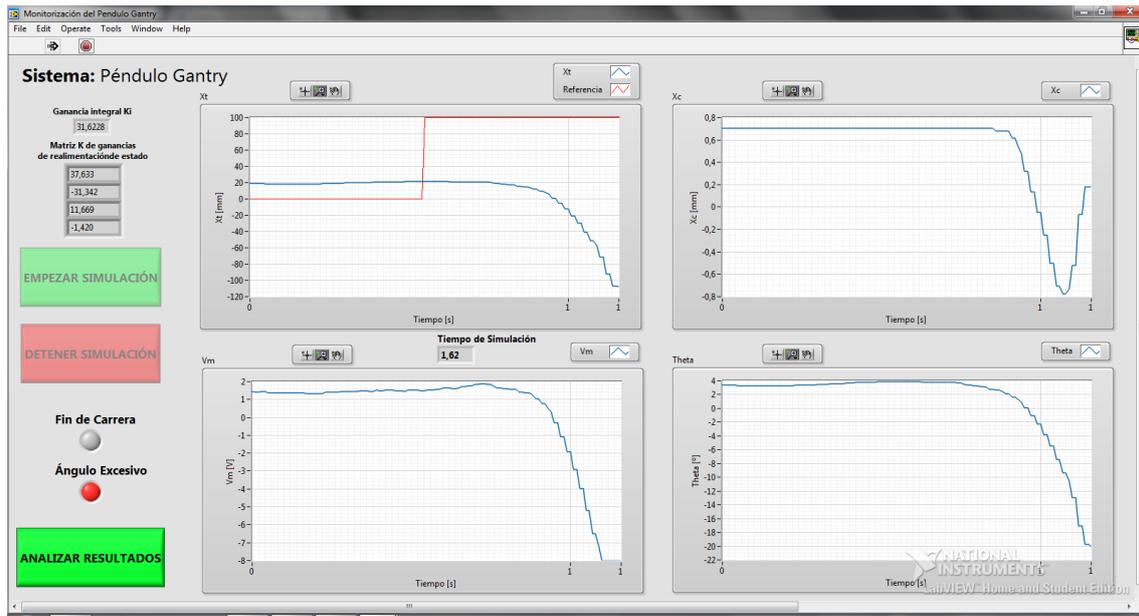


Figura 41. Indicador de ángulo excesivo

Así mismo, si el tiempo de simulación excede el tiempo máximo establecido en la pantalla anterior, se mostrará el siguiente mensaje en pantalla y se detendrá la prueba.

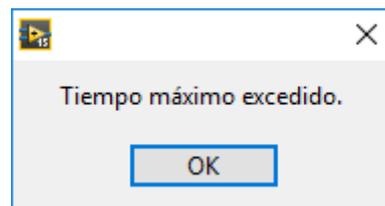


Figura 42. Mensaje de tiempo máximo excedido

En las pantallas de péndulo invertido, además, existen dos indicadores que dan información acerca de la fase en la que se encuentra el sistema. Tras el inicio de la prueba, se encenderá la luz que indica que el sistema se encuentra en fase de erección, automática o manual, y, una vez hecha la conmutación, con el péndulo en su posición superior, se indicará que el sistema está en la fase de control de balanceo, como muestran las figuras siguientes.

En el caso de seleccionarse el modo de erección manual, el usuario deberá llevar manualmente el péndulo a su posición superior, haciéndolo de forma lenta y no brusca, sobre todo al acercarse al ángulo de conmutación. Se recomienda hacerlo cogiendo el péndulo desde su extremo. Una vez se llegue al ángulo de conmutación, se entrará en la fase de balanceo, y el usuario podrá soltar el péndulo, que se sostendrá gracias al control.

Tanto en erección manual, como en autoerección, el péndulo se mantendrá durante cinco segundos en la posición en la que se haya producido la conmutación para garantizar que se estabiliza, y posteriormente comenzará la referencia seleccionada por el usuario.

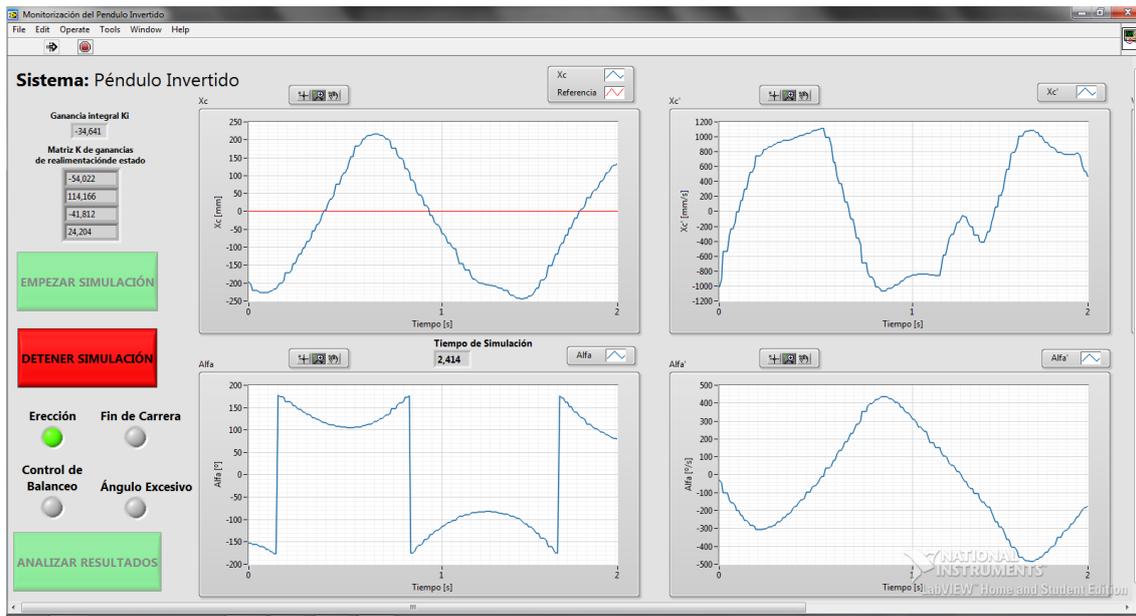


Figura 43. Indicador de fase de erección

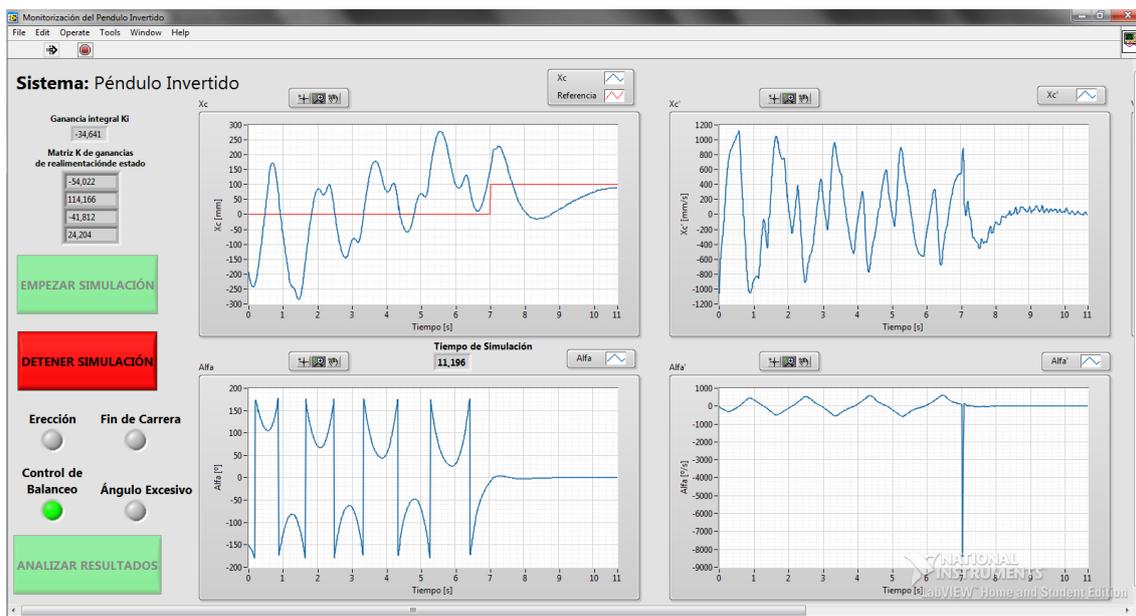


Figura 44. Indicador de fase de control de balanceo

Tras finalizar la prueba, sea de la manera que sea, se podrán observar los resultados, hasta que se pulse el botón “Analizar Resultados”, con el que se pasará a la siguiente pantalla, en la que se representarán los resultados exactos de la prueba.

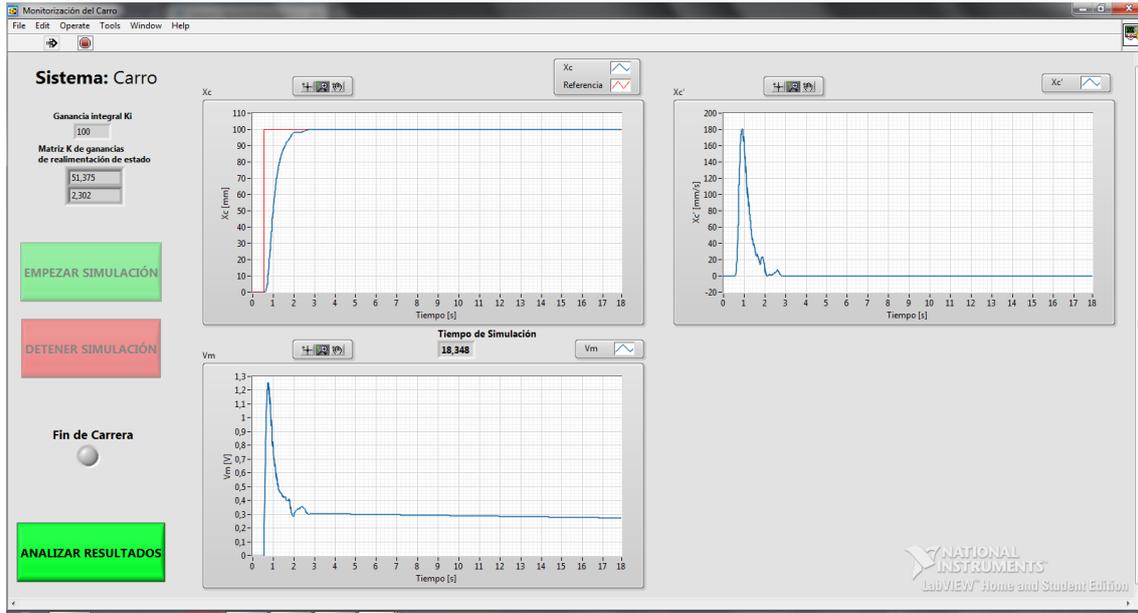


Figura 45. Pantalla de monitorización para el sistema carro tras la prueba

Además, una vez finalizada la prueba, en la tarjeta NI myRIO se apagarán el LED1 y el LED2, dejando únicamente encendido el LED0, que indica que la tarjeta está lista para una nueva prueba.

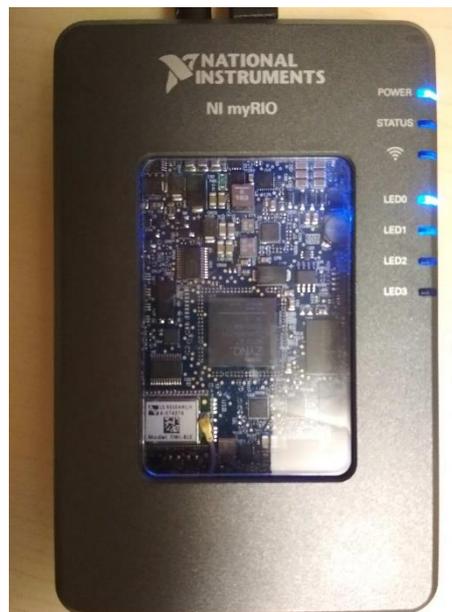


Figura 46. Tarjeta NI myRIO lista para iniciar la aplicación de ordenador

## 7.4. Análisis de resultados

En esta pantalla se representarán los resultados de las pruebas realizadas, del sistema real, del modelo o de ambos, según se haya seleccionado en la preparación de la prueba. Además, se representará la referencia introducida al sistema. Finalmente, en el extremo derecho de la pantalla, se ubica el botón "Terminar", que cerrará todas las pantallas salvo la de inicio, en la que se podrá volver a seleccionar de nuevo el sistema de estudio.

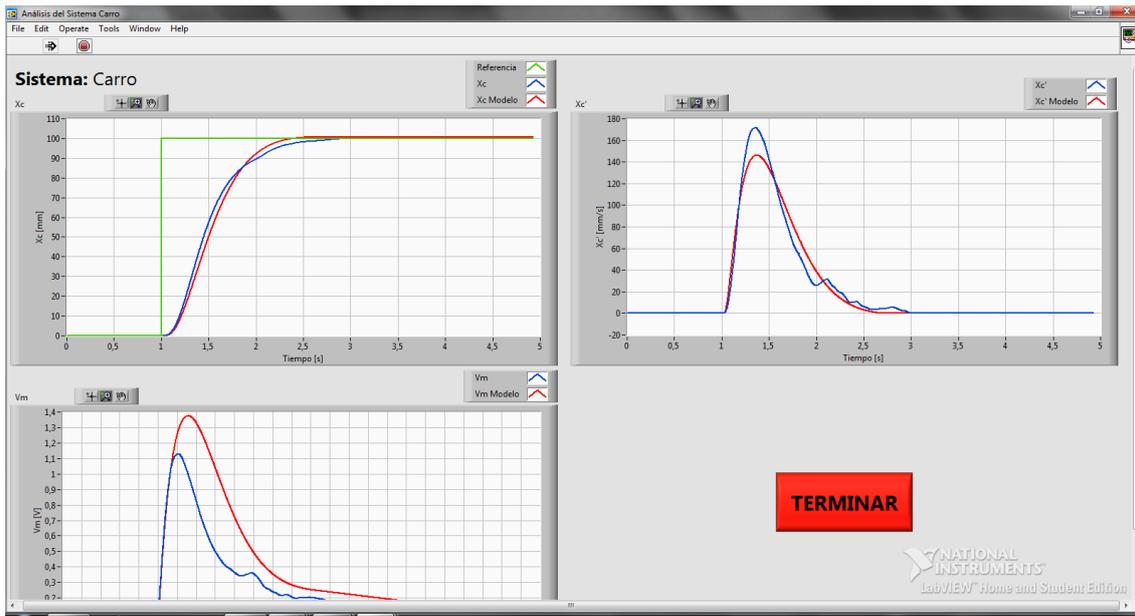


Figura 47. Pantalla de análisis de resultado para el sistema carro

Cabe destacar que, en ciertos casos, debido al gran número de variables a mostrar, no se podrán mostrar todas a la vez en pantalla. En dichos casos, se podrá desplazar la pantalla con la barra inferior, para mostrar el resto de variables y, al final, el botón "Terminar".

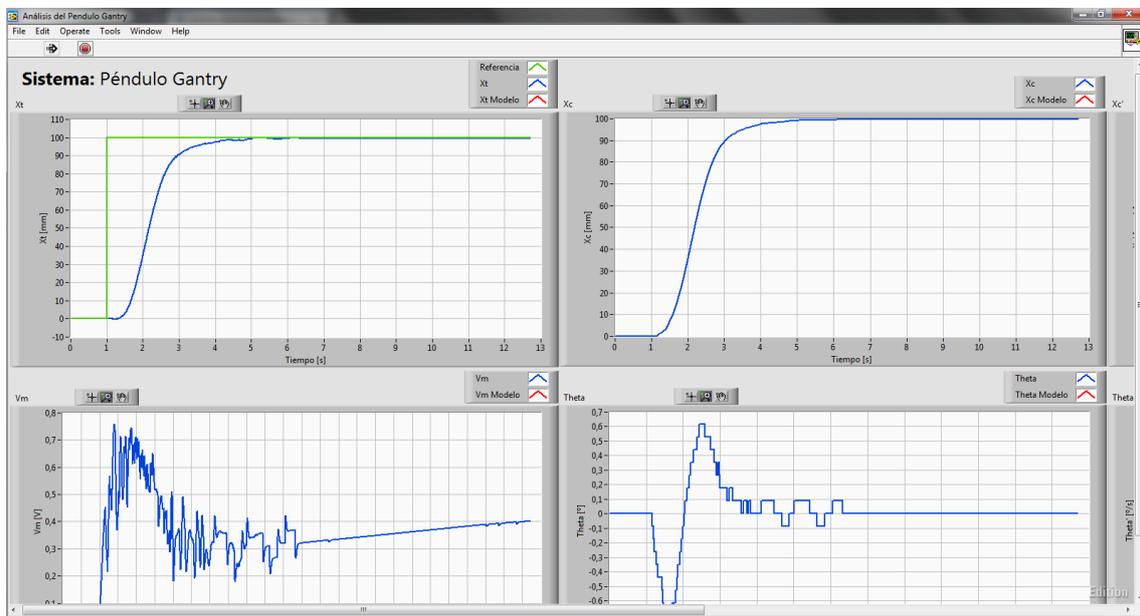


Figura 48. Pantalla de análisis de resultado para el sistema péndulo Gantry I

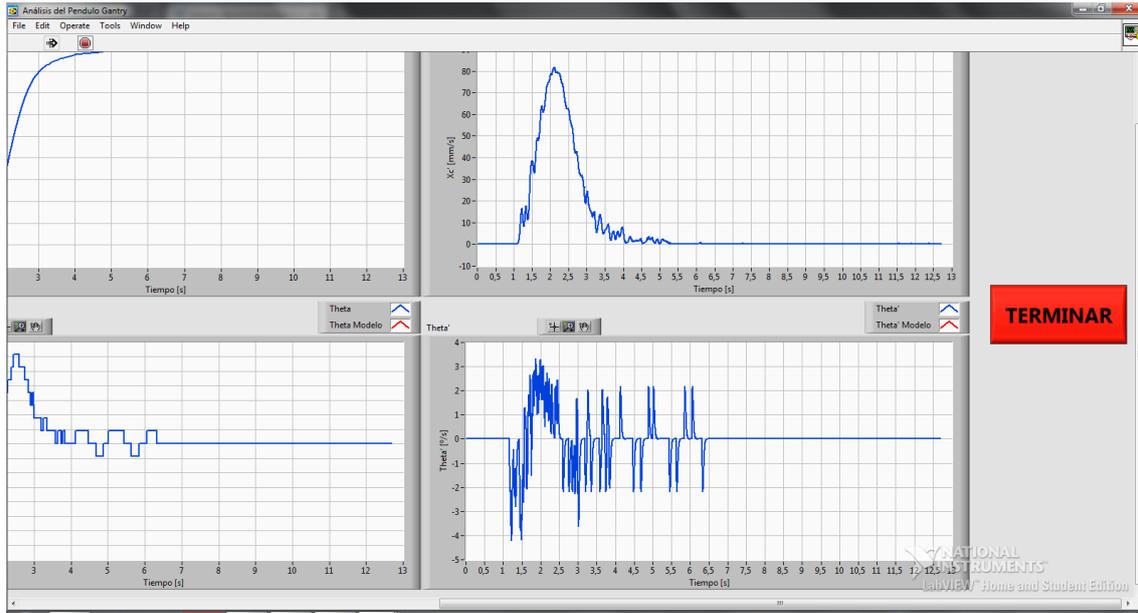


Figura 49. Pantalla de análisis de resultado para el sistema péndulo Gantry II

Para poder observar en detalle los gráficos, tanto en este caso como en el de monitorización, se incluyen herramientas sobre ellos. En la pantalla de monitorización, únicamente pueden usarse al finalizar la prueba.

La herramienta central permite hacer zoom sobre el gráfico, de las distintas maneras que indican los dibujos que se despliegan al hacer clic sobre ella. En la siguiente figura, de izquierda a derecha, primero la fila superior y después la inferior:

- Zoom vertical y horizontal
- Zoom horizontal
- Zoom vertical
- Autoajustar al tamaño de los datos
- Acercar al hacer clic sobre el gráfico
- Alejar al hacer clic sobre el gráfico

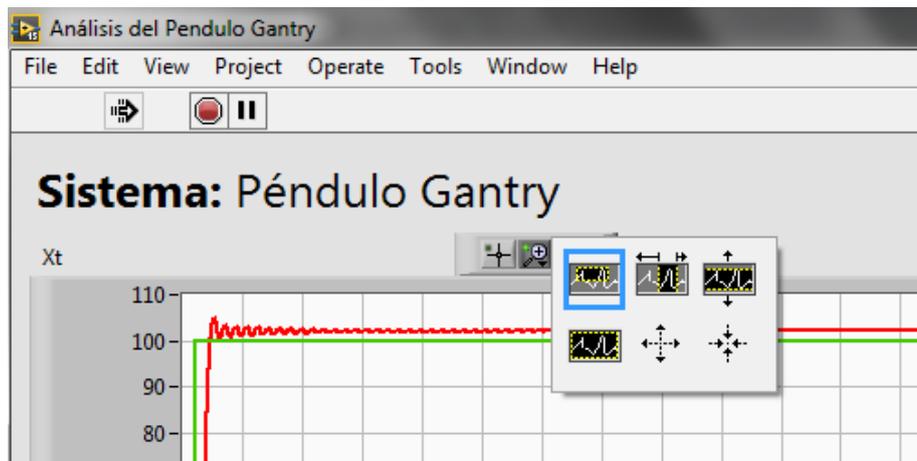
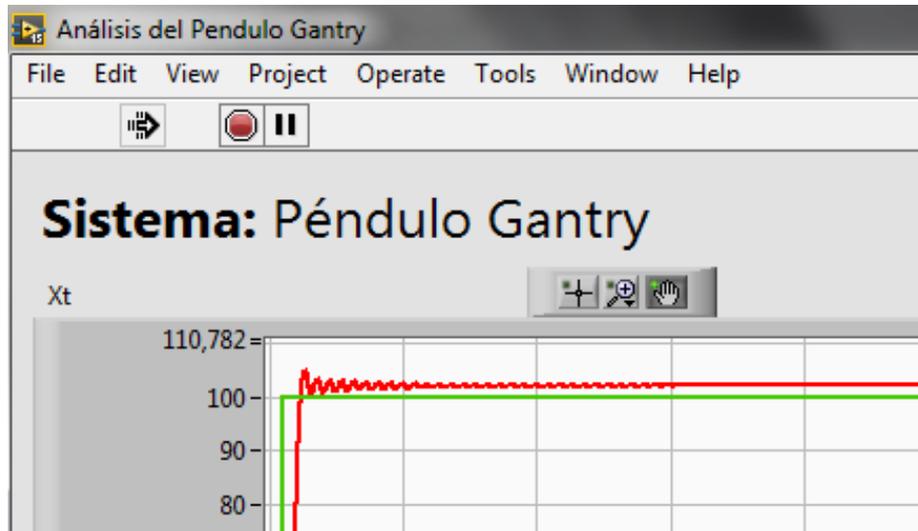


Figura 50. Herramienta central de gráfico

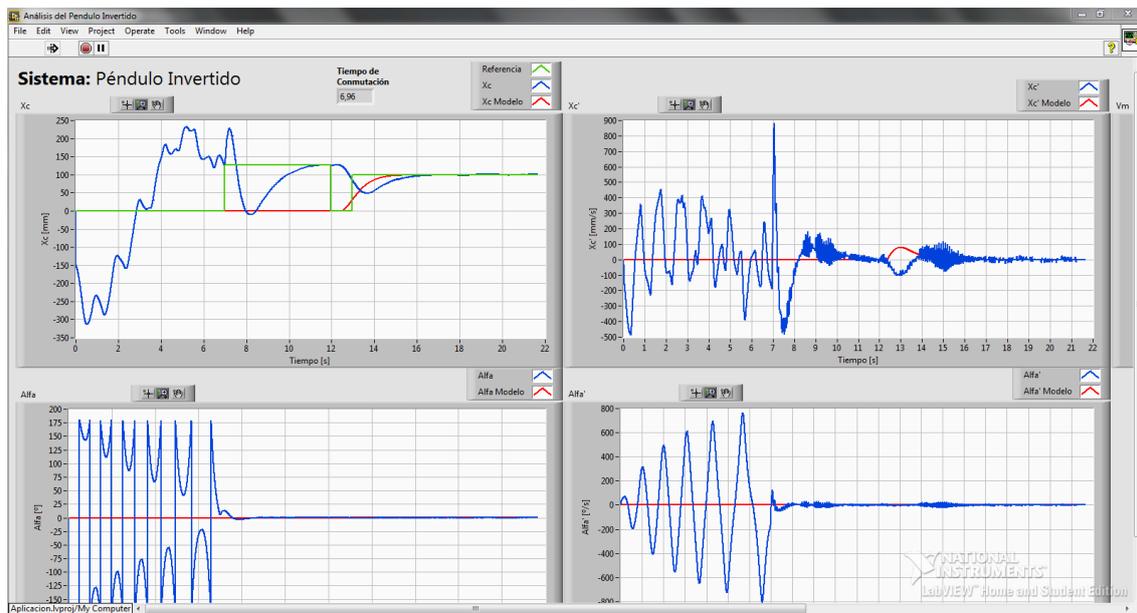
Seleccionando la herramienta de la derecha el usuario puede moverse por el gráfico arrastrando la pantalla del mismo al mantener pulsado el botón izquierdo del ratón y mover el mismo.

Por último, presionar en la herramienta de la izquierda únicamente sirve para tener disponible el cursor para moverse por el gráfico.



*Figura 51. Herramientas de gráfico*

En el caso del sistema péndulo invertido, la autoerección no se simula sobre el modelo. Si únicamente se ha seleccionado el modelo, la simulación comienza con el péndulo erecto. Si se seleccionan modelo y sistema real, la simulación comenzará cinco segundos (tiempo para estabilizarse) tras el tiempo de conmutación registrado en la prueba, que se mostrará en pantalla.

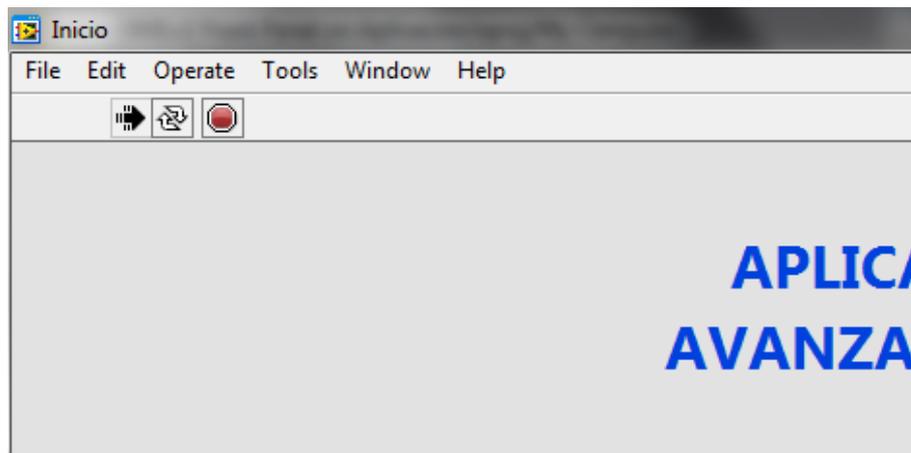


*Figura 52. Pantalla de análisis de resultados de sistema real y modelo para el péndulo invertido*

### **7.5. Ejecución anormal de la aplicación**

Aunque la aplicación está programada para que no se produzcan errores en la ejecución, si el usuario decidiera cerrar una ventana cuando no corresponde, el programa podría salirse de su ejecución normal, y bloquearse la aplicación de ordenador y/o la aplicación en tiempo real en la tarjeta.

Si se bloquease la aplicación en tiempo real, se recomienda ir a la ventana de inicio de la aplicación y pulsar el botón rojo que se encuentra en la parte superior izquierda, y que se muestra en la figura siguiente. Después de hacer esto, se deberá cerrar la ventana de la aplicación y reiniciar la misma para volver a usarla.



*Figura 53. Detención del programa ante ejecución anormal*

Si es la tarjeta la que se queda a mitad de una ejecución en tiempo real, es decir, con alguna de los indicadores LED1 o LED2 encendidos, lo primero será apagar el módulo UPM para garantizar que no se actúa sin control sobre la maqueta, ya que podría dañarla. Posteriormente, para preparar la tarjeta antes de llevar a cabo otra prueba, será necesario desconectar la alimentación y volver a conectarla, esperando para ello a que vuelva a quedar encendido únicamente el LED0. Así mismo, se debe cerrar y volver a abrir la aplicación del ordenador, aunque ésta no hubiera sufrido bloqueo (este último caso es improbable).

## **ANEXO V: CÓDIGO DE LA APLICACIÓN**

En este anexo se incluye el código de cada archivo VI (Virtual Instrument) de la aplicación, tanto de la aplicación de ordenador como de la aplicación en tiempo real, con comentarios en el mismo.

## 1. HMI.vi

Home/Student Edition

Page 1

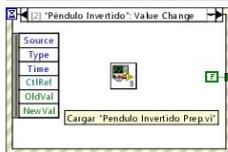
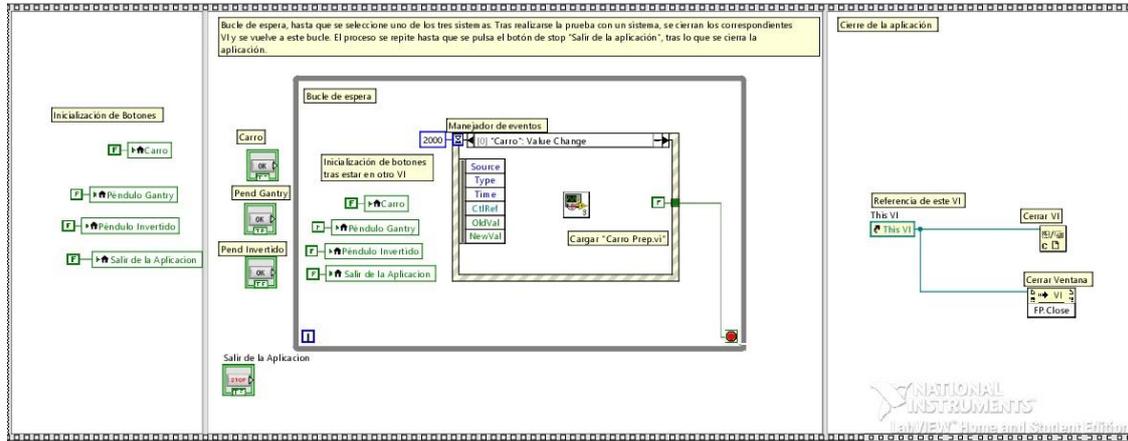


Inicio

G:\Uni\TFM\LabView\Aplicacion\HMI.vi

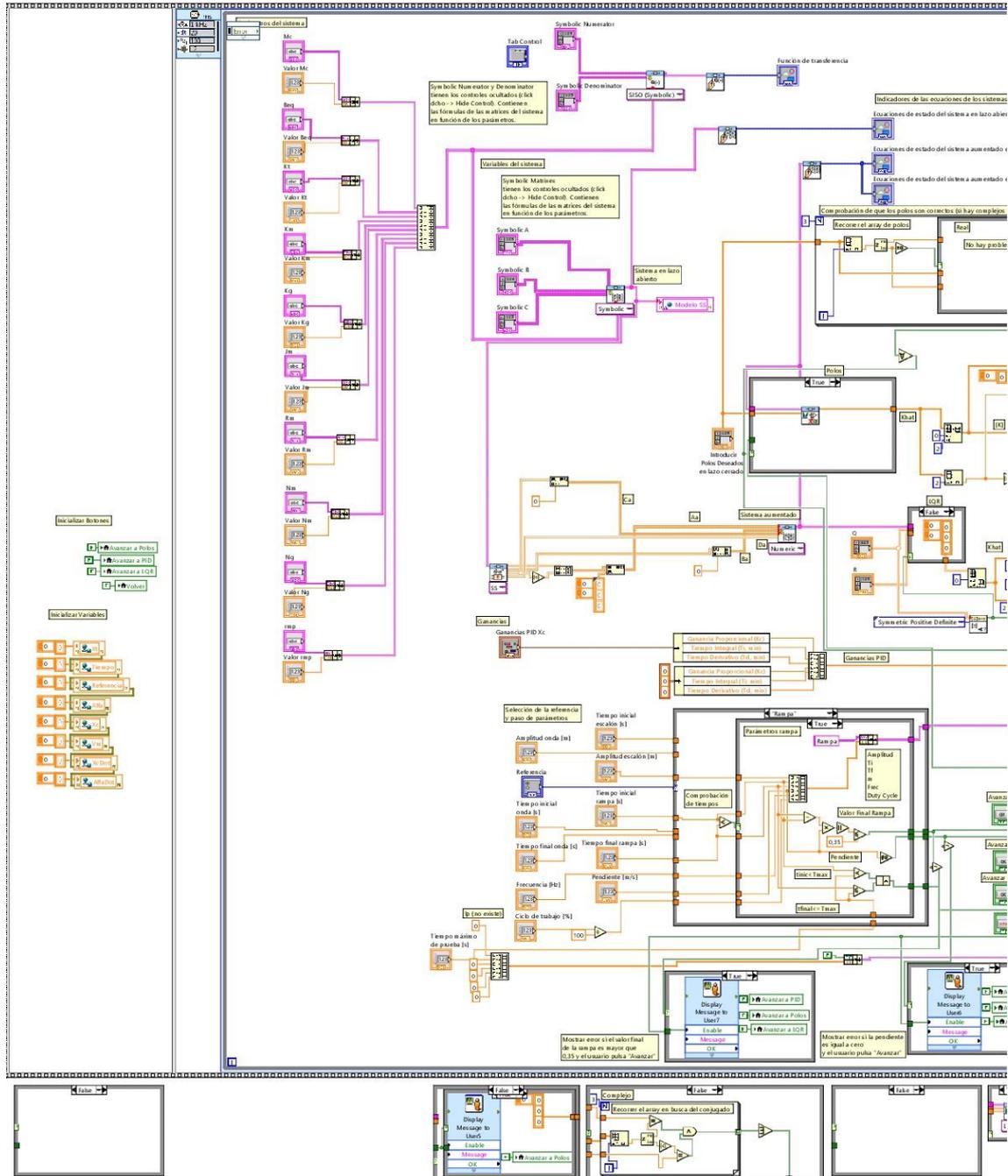
Last modified on 21/09/2018 at 13:14

Printed on 24/09/2018 at 9:26

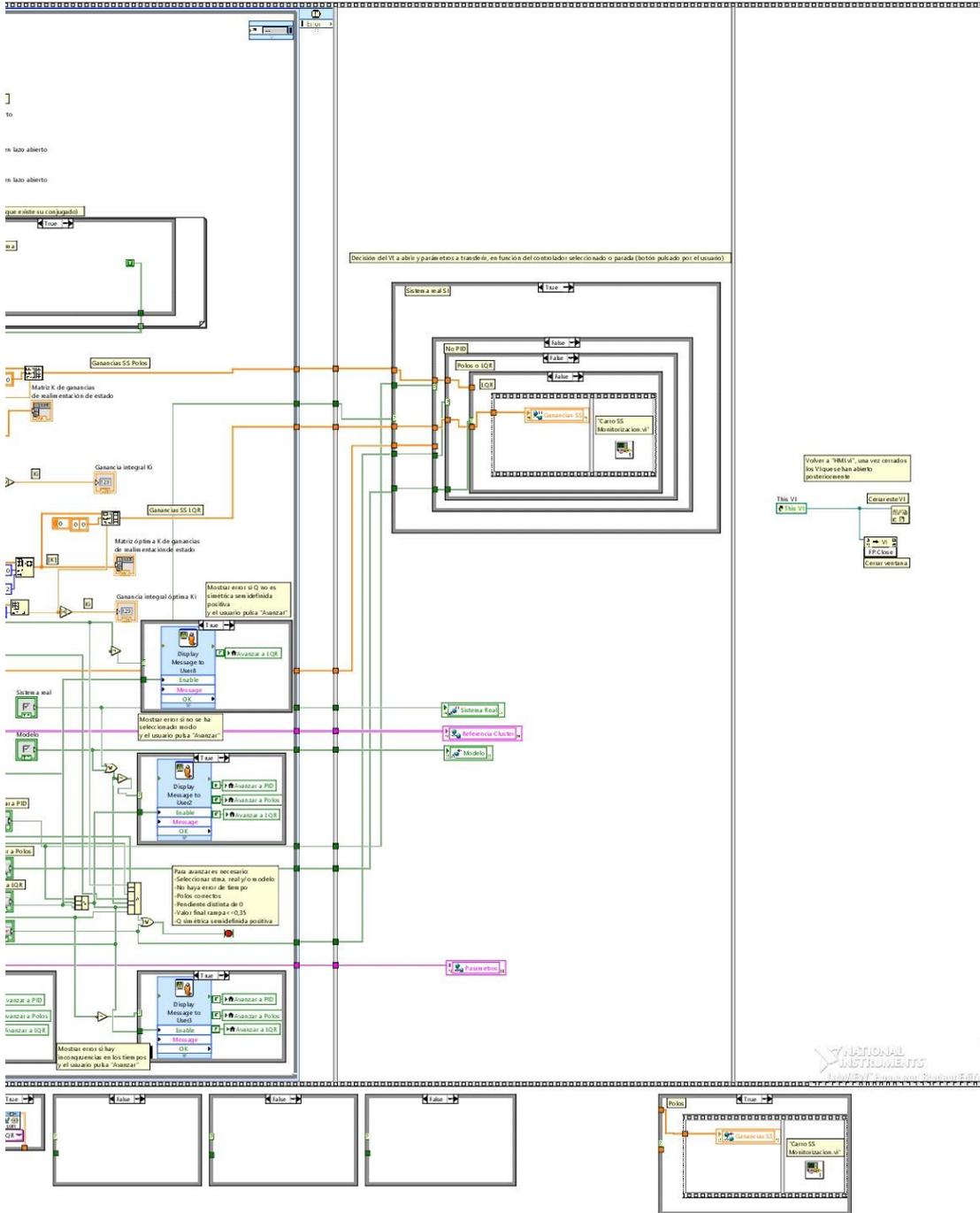


## 2. Carro Prep.vi

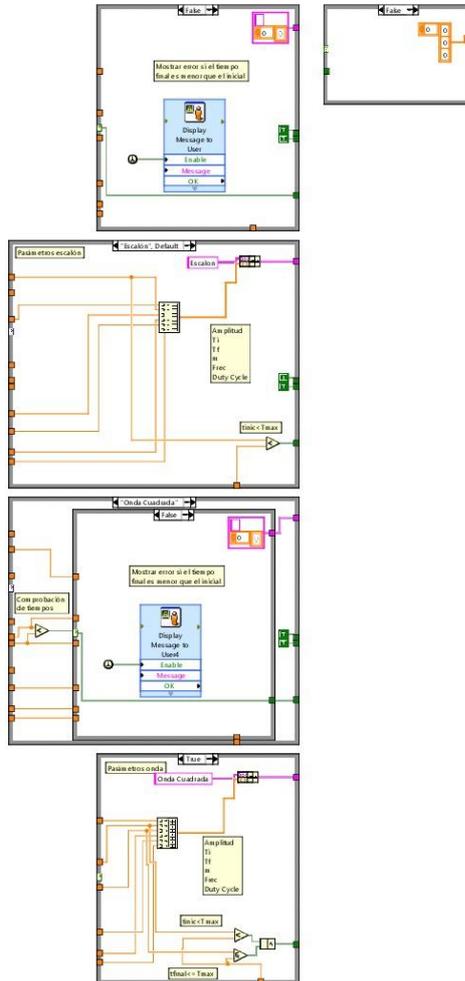
Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Carro Prep.vi  
Last modified on 21/09/2018 at 16:02  
Printed on 24/09/2018 at 9:34



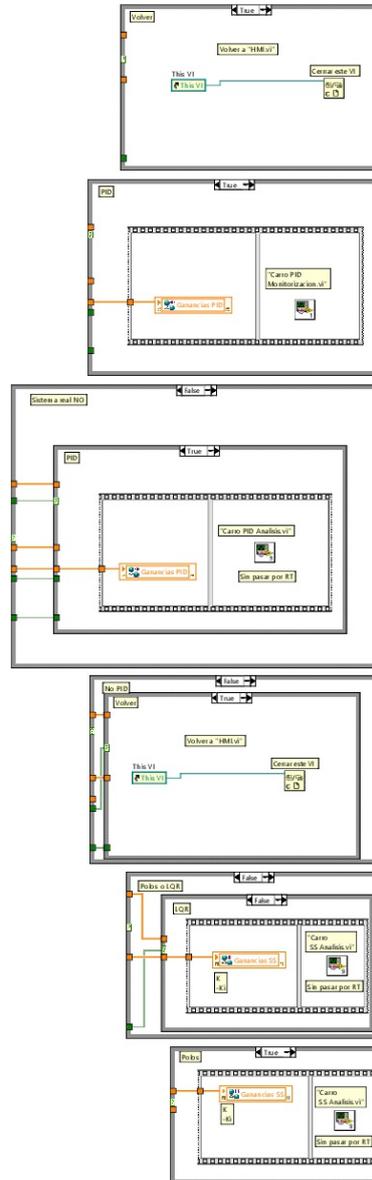
Home/Student Edition  
 Preparación de la Prueba  
 G:\Uni\TFM\LabView\Aplicacion\Carro Prep.vi  
 Last modified on 21/09/2018 at 16:02  
 Printed on 24/09/2018 at 9:34



Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Carro Prep.vi  
Last modified on 21/09/2018 at 16:02  
Printed on 24/09/2018 at 9:35

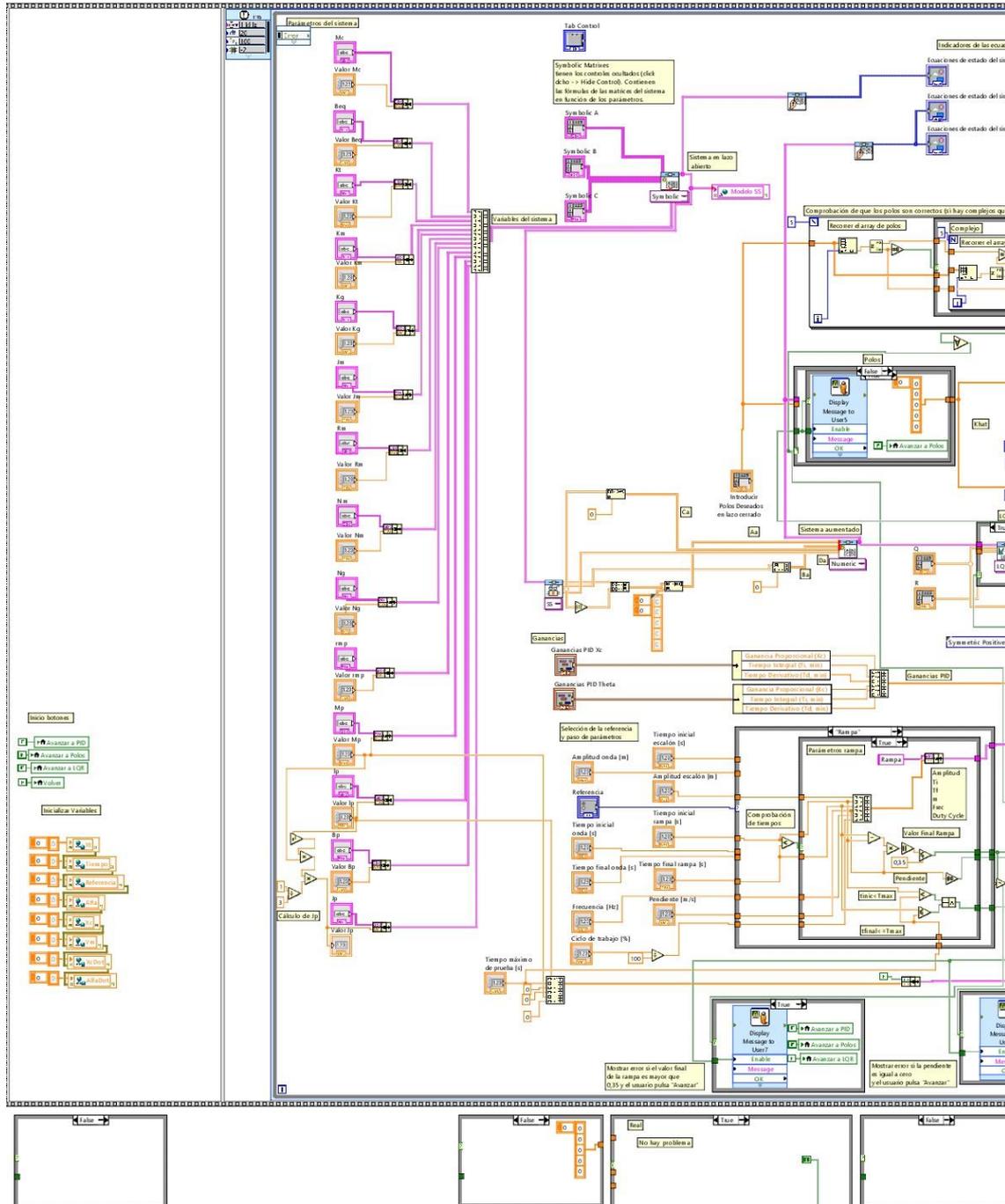


Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Carro Prep.vi  
Last modified on 21/09/2018 at 16:02  
Printed on 24/09/2018 at 9:35

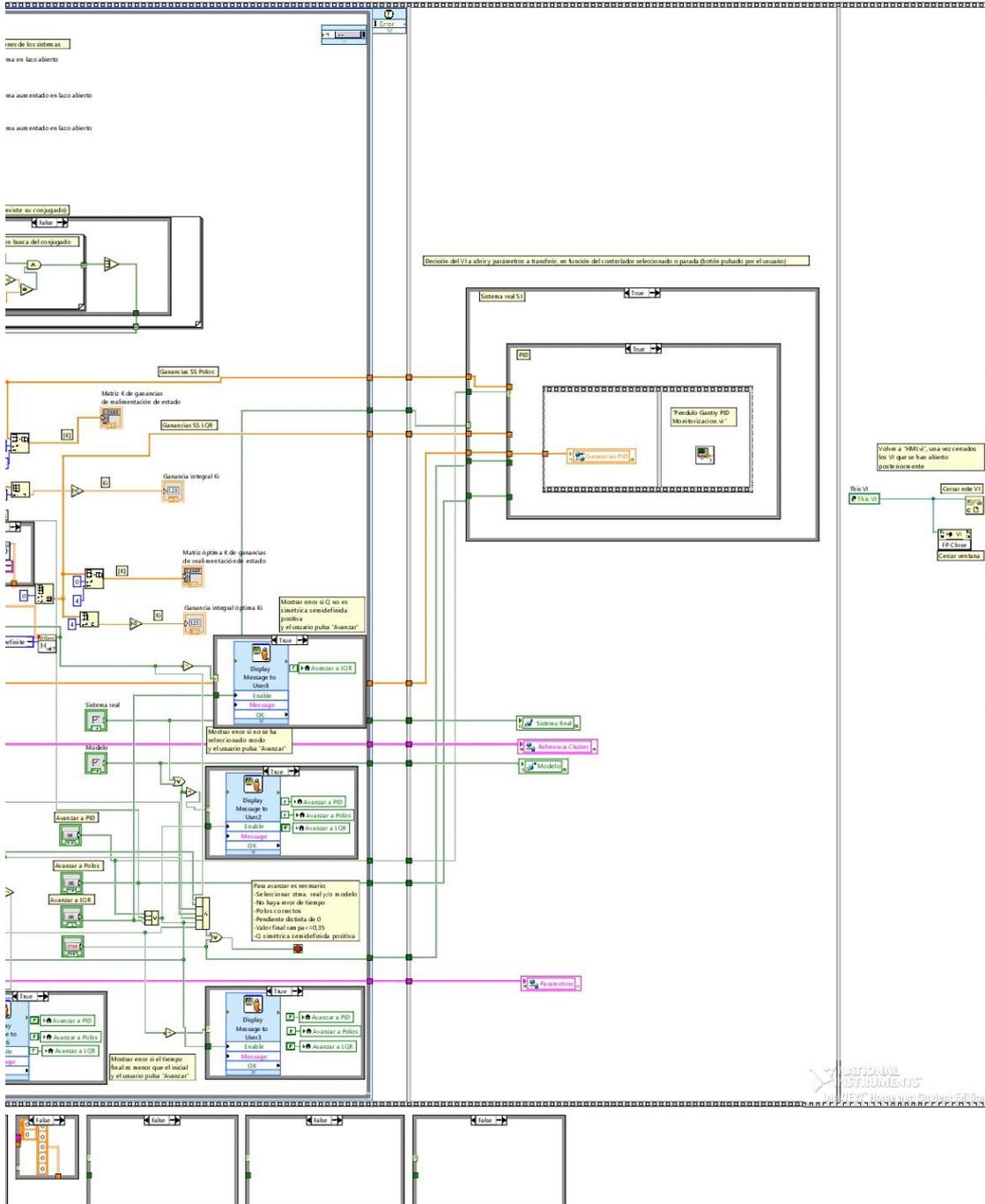


## 3. Pendulo Gantry Prep.vi

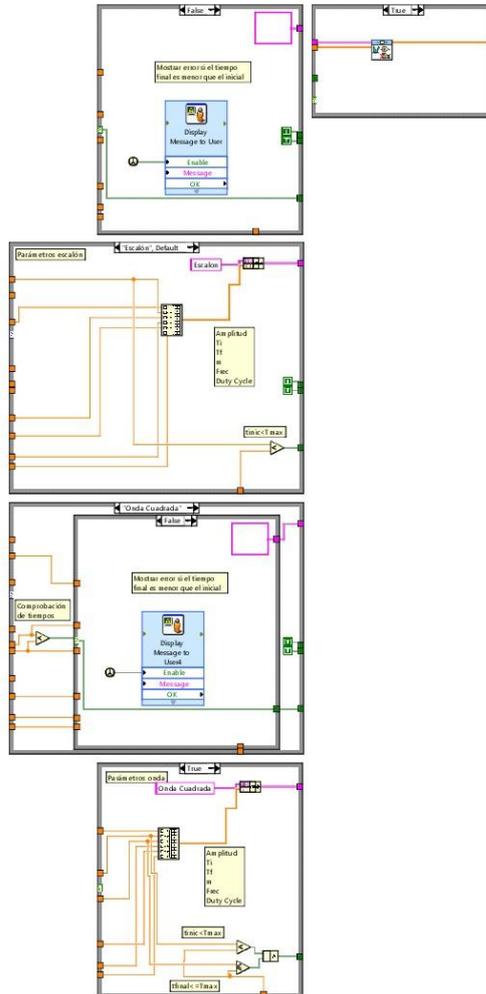
Home/Student Edition  
 Preparación de la Prueba  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry Prep.vi  
 Last modified on 24/09/2018 at 14:19  
 Printed on 24/09/2018 at 18:46



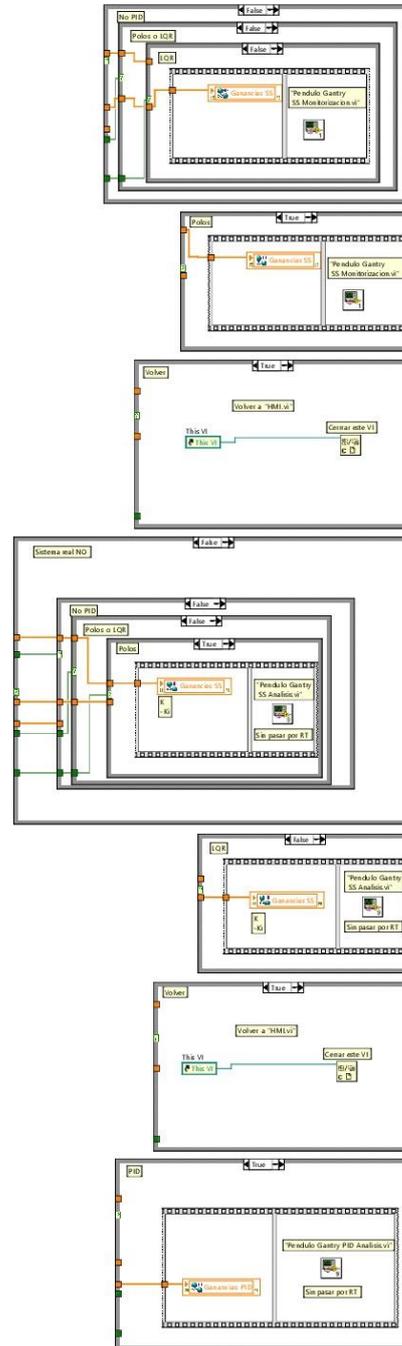
Home/Student Edition  
 Preparación de la Prueba  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry Prep.vi  
 Last modified on 24/09/2018 at 14:19  
 Printed on 24/09/2018 at 18:46



Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry Prep.vi  
Last modified on 24/09/2018 at 14:19  
Printed on 24/09/2018 at 18:46

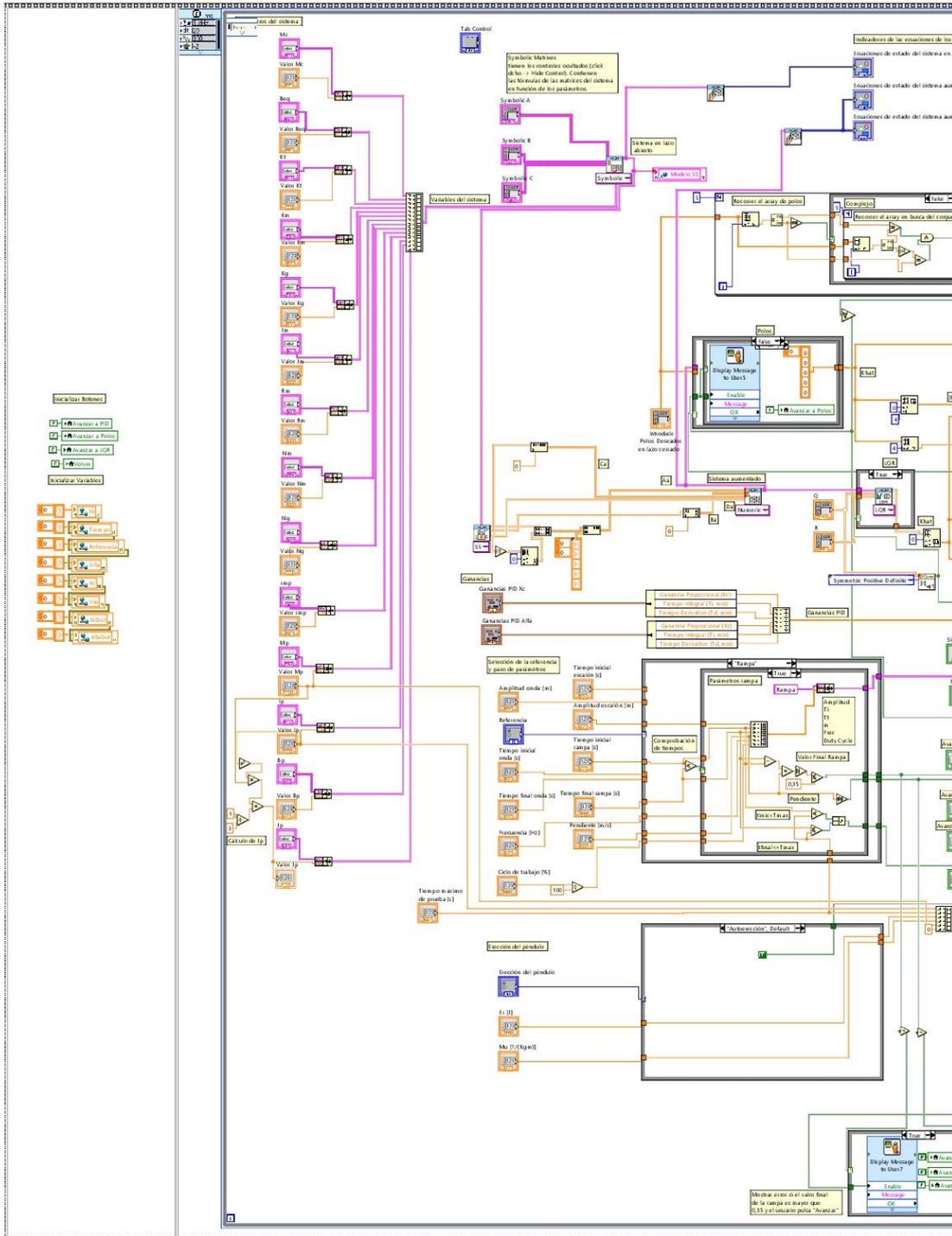


Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry Prep.vi  
Last modified on 24/09/2018 at 14:19  
Printed on 24/09/2018 at 18:46



## 4. Pendulo Invertido Prep.vi

Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido Prep.vi  
Last modified on 24/09/2018 at 14:22  
Printed on 24/09/2018 at 18:50





Home/Student Edition

Page 3

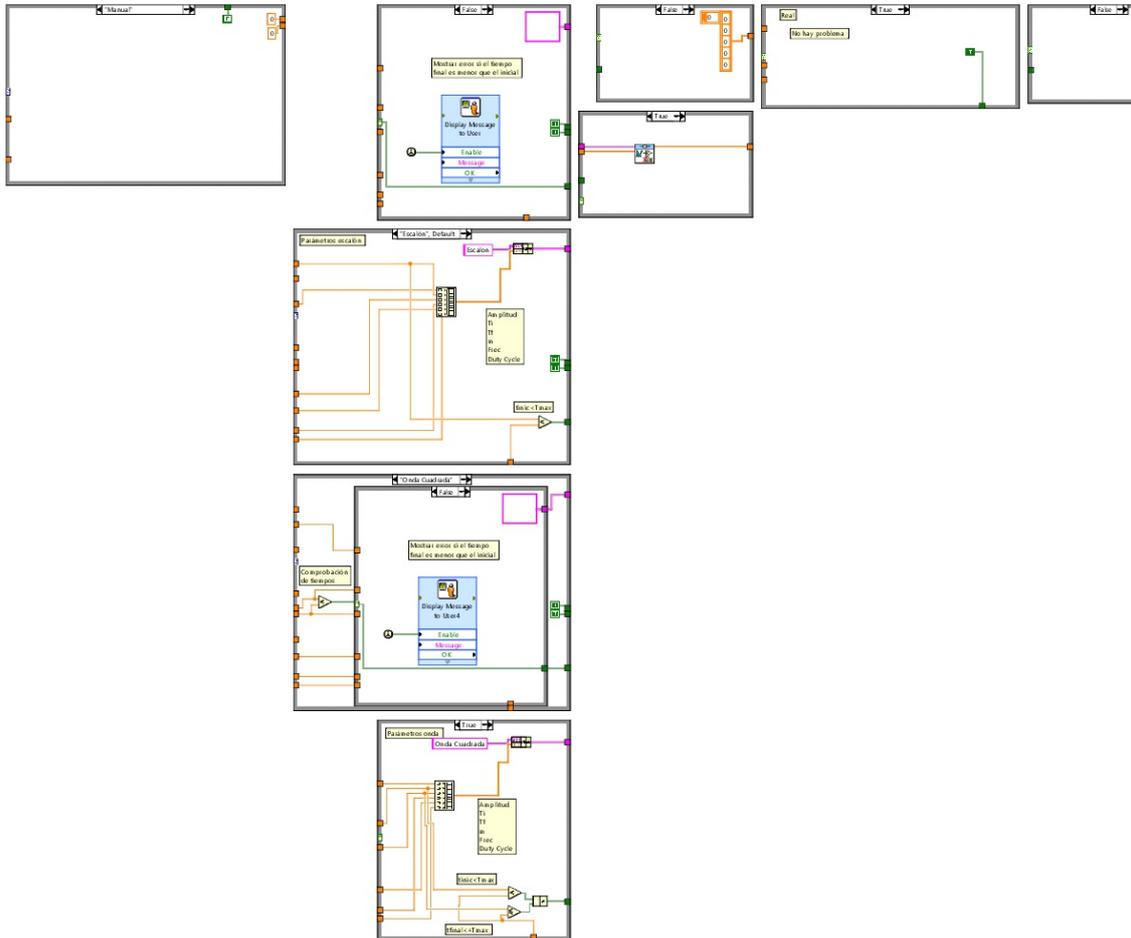


Preparación de la Prueba

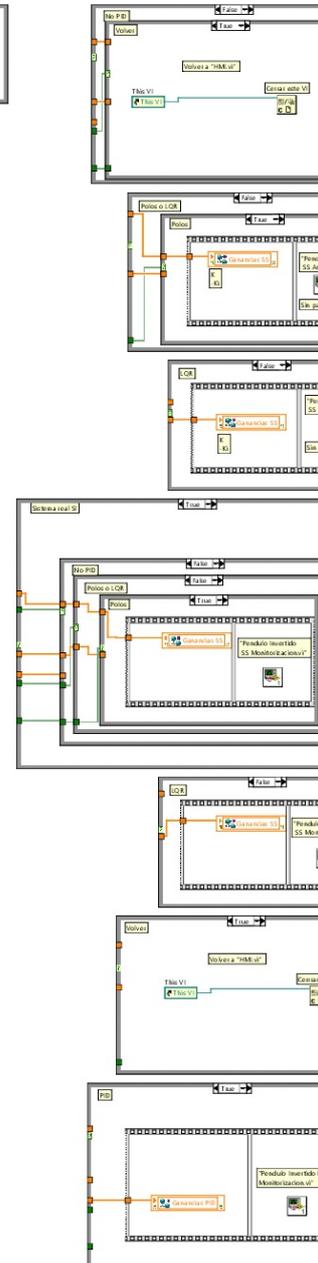
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido Prep.vi

Last modified on 24/09/2018 at 14:22

Printed on 24/09/2018 at 18:51



Home/Student Edition  
 Preparación de la Prueba  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido Prep.vi  
 Last modified on 24/09/2018 at 14:22  
 Printed on 24/09/2018 at 18:51



Home/Student Edition  
Preparación de la Prueba  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido Prep.vi  
Last modified on 24/09/2018 at 14:22  
Printed on 24/09/2018 at 18:51



## 5. Carro PID Monitorizacion.vi

Home/Student Edition

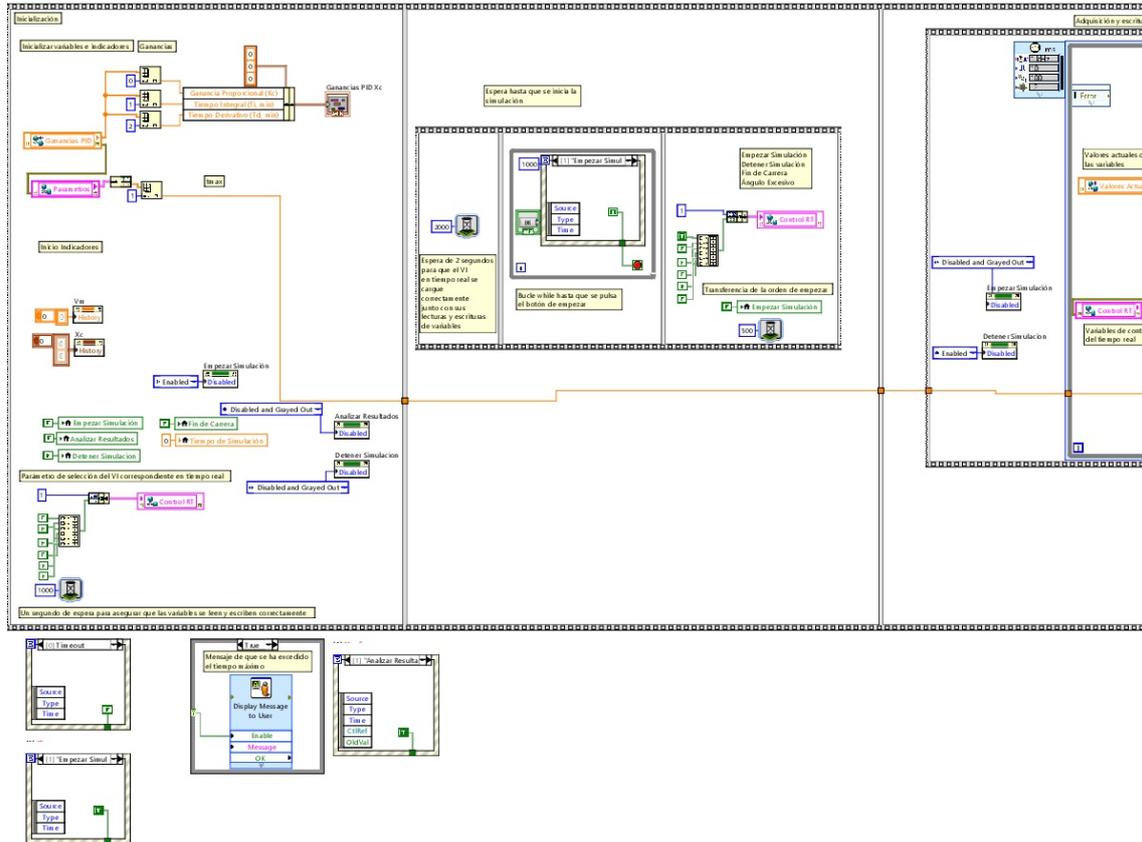
Monitorización del Carro

G:\Uni\TFM\LabView\Aplicacion\Carro PID Monitorizacion.vi

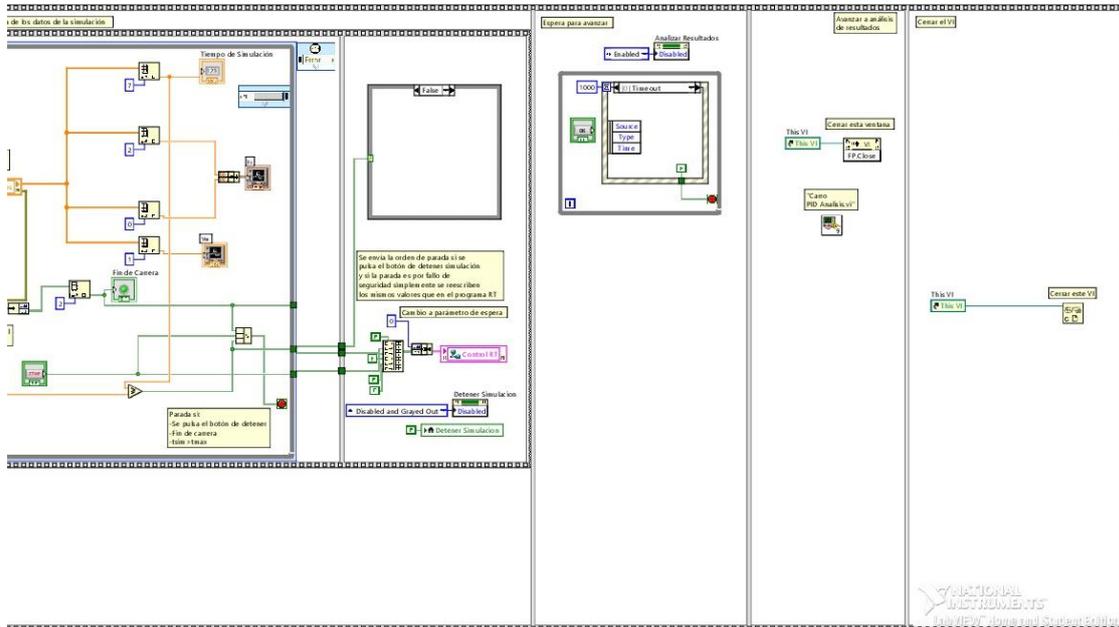
Last modified on 24/09/2018 at 19:07

Printed on 24/09/2018 at 19:07

Page 1

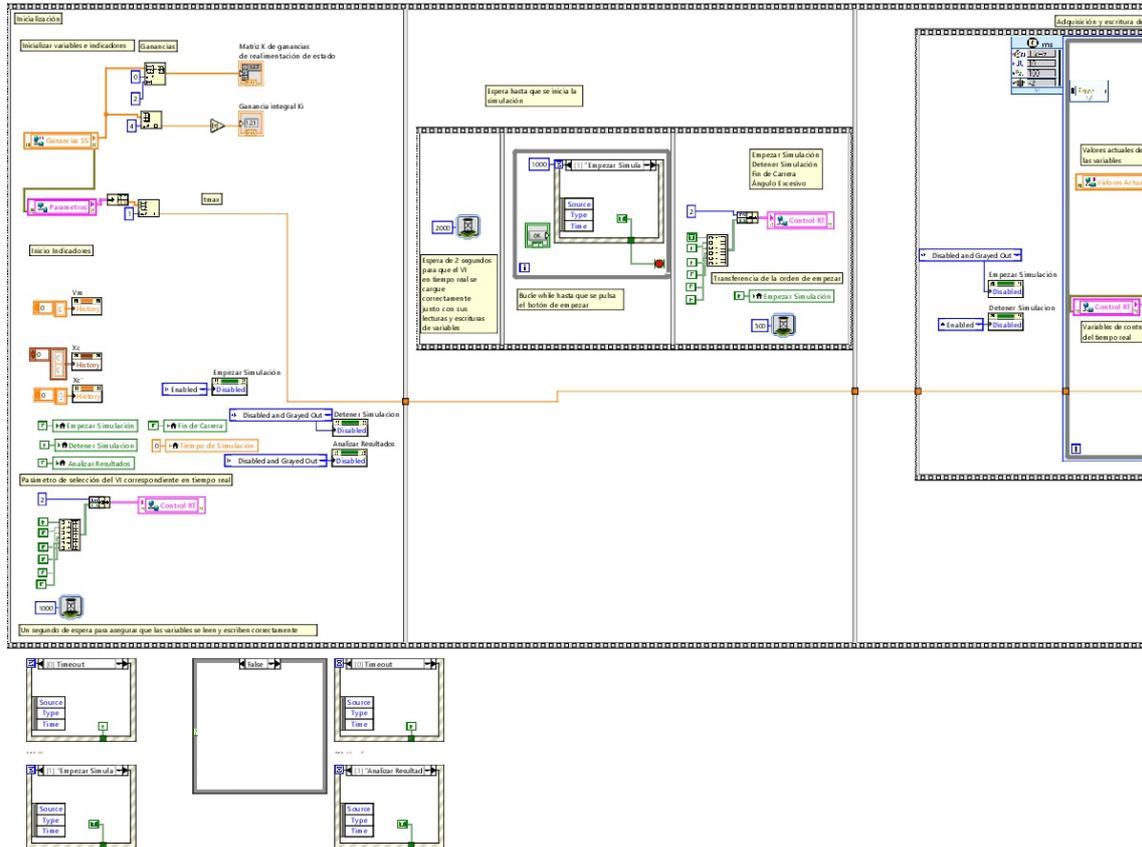


Home/Student Edition  
 Monitorización del Carro  
 G:\Uni\TFM\LabView\Aplicacion\Carro PID Monitorizacion.vi  
 Last modified on 24/09/2018 at 19:07  
 Printed on 24/09/2018 at 19:07

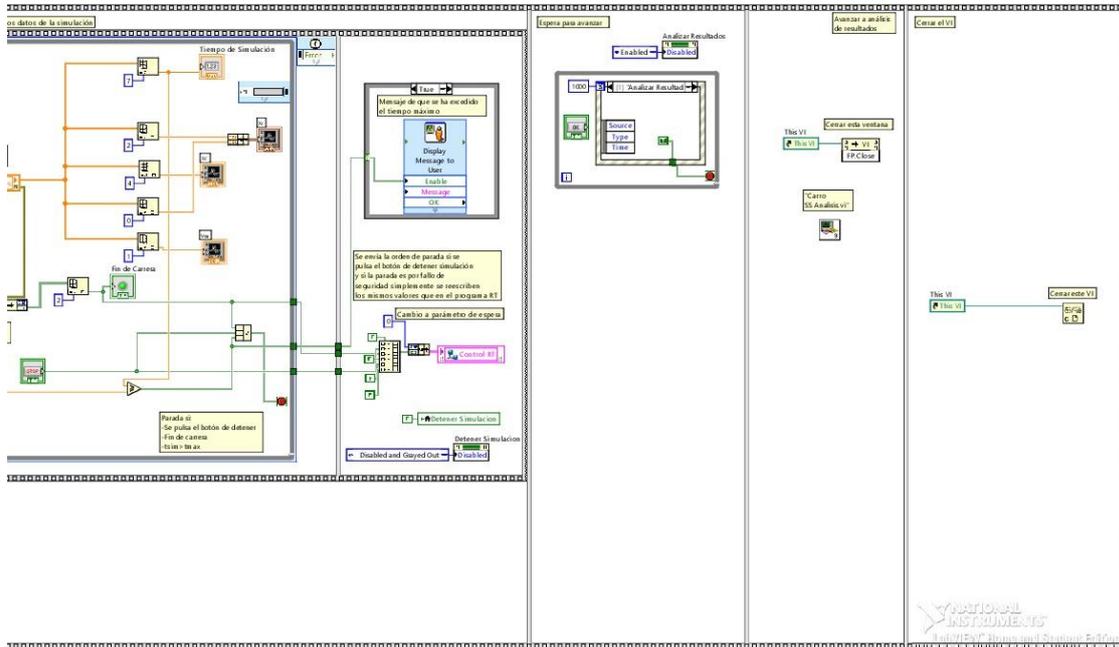


## 6. Carro SS Monitorizacion.vi

Home/Student Edition  
 Monitorización del Carro  
 G:\Uni\TFM\LabView\Aplicacion\Carro SS Monitorizacion.vi  
 Last modified on 24/09/2018 at 19:09  
 Printed on 24/09/2018 at 19:10



Home/Student Edition  
 Monitorización del Carro  
 G:\Uni\TFM\LabView\Aplicacion\Carro SS Monitorizacion.vi  
 Last modified on 24/09/2018 at 19:09  
 Printed on 24/09/2018 at 19:10



## 7. Pendulo Gantry PID Monitorizacion.vi

Home/Student Edition

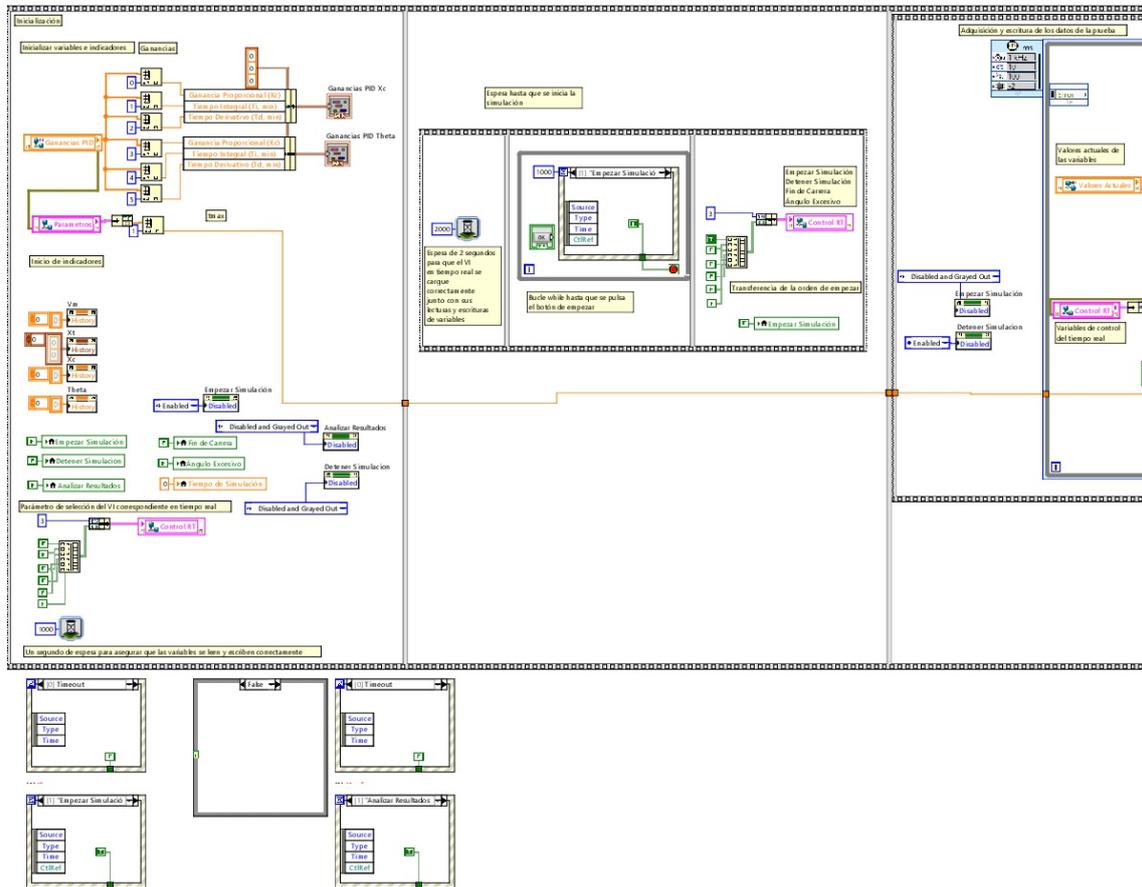
Monitorización del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry PID Monitorizacion.vi

Last modified on 24/09/2018 at 19:11

Printed on 24/09/2018 at 19:12

Page 1



Home/Student Edition

Page 2

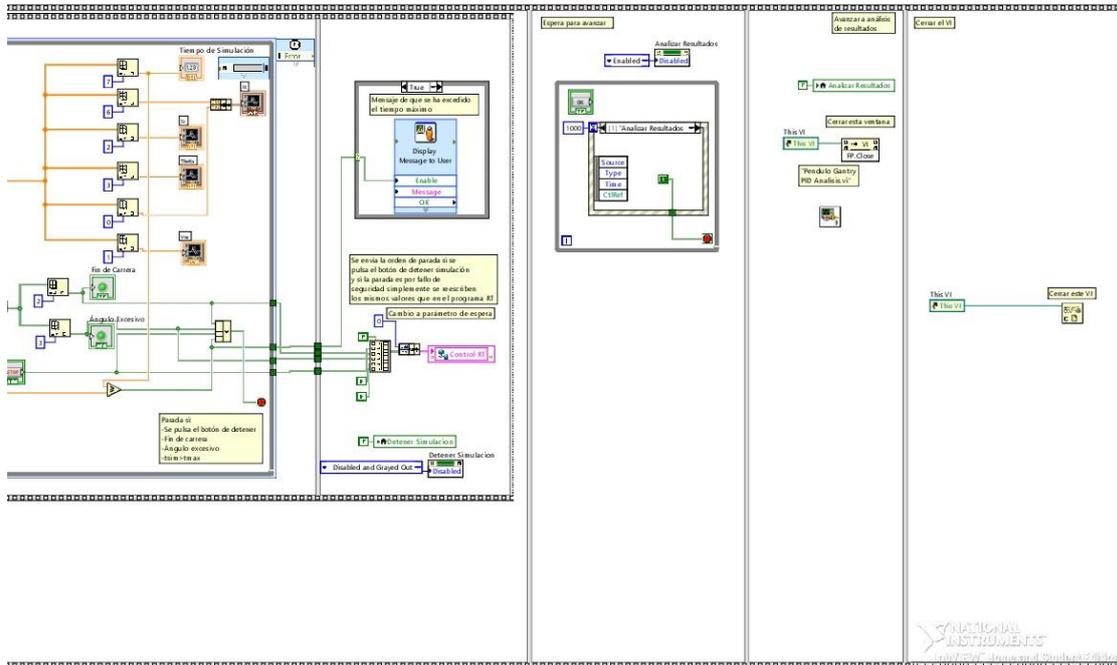


Monitorización del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry PID Monitorizacion.vi

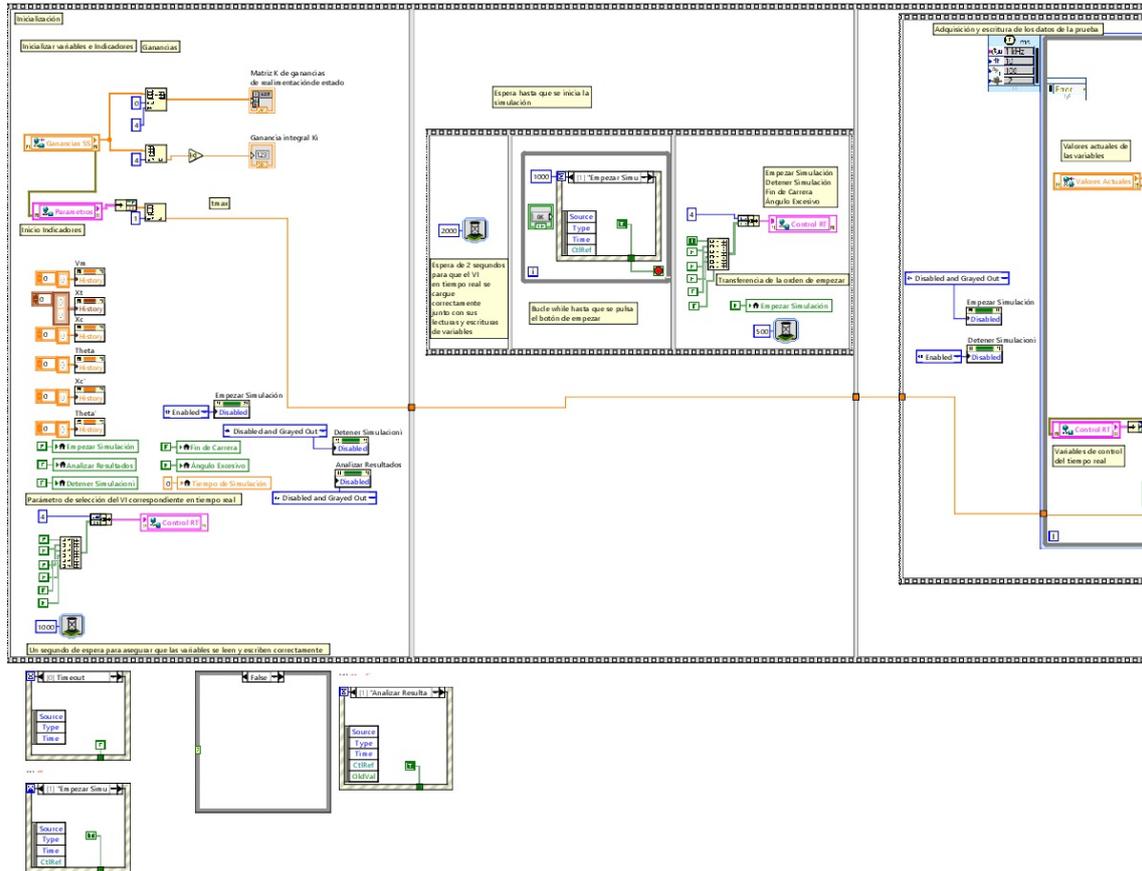
Last modified on 24/09/2018 at 19:11

Printed on 24/09/2018 at 19:12



## 8. Pendulo Gantry SS Monitorizacion.vi

Home/Student Edition  
 Monitorización del Pendulo Gantry  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry SS Monitorizacion.vi  
 Last modified on 24/09/2018 at 19:15  
 Printed on 24/09/2018 at 19:15



Home/Student Edition

Page 2

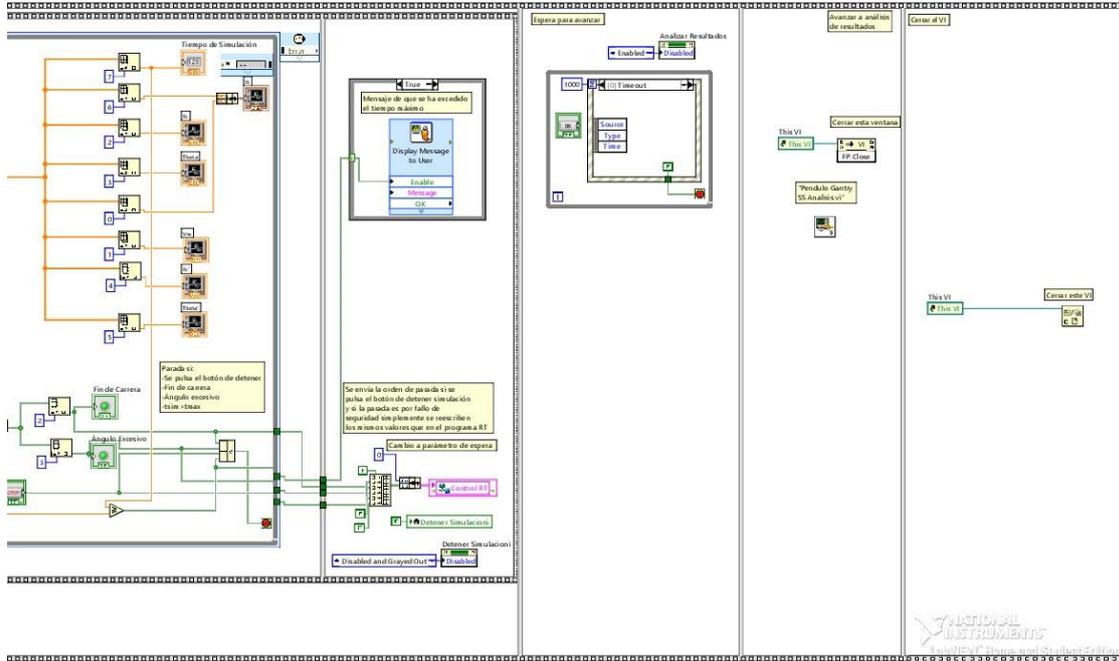


Monitorización del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry SS Monitorizacion.vi

Last modified on 24/09/2018 at 19:15

Printed on 24/09/2018 at 19:15



## 9. Pendulo Invertido PID Monitorizacion.vi

Home/Student Edition

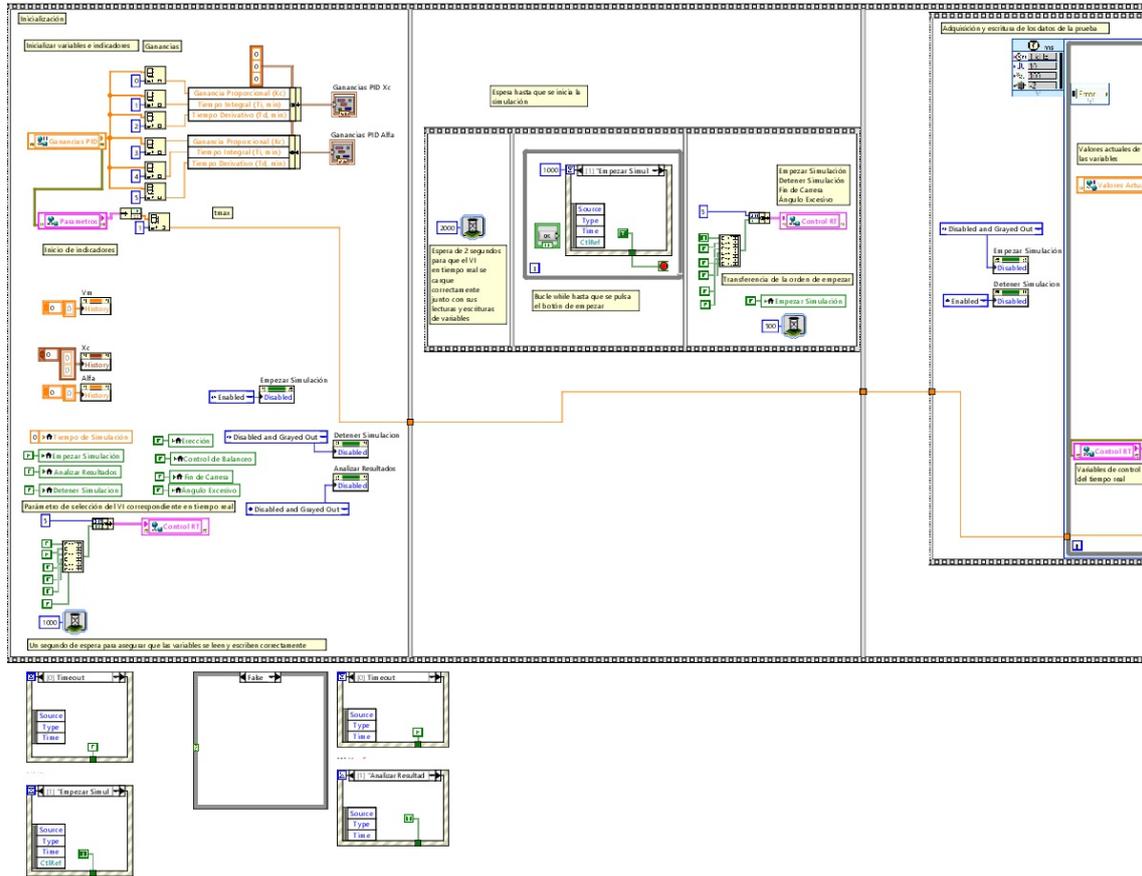
Monitorización del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID Monitorizacion.vi

Last modified on 24/09/2018 at 19:18

Printed on 24/09/2018 at 19:18

Page 1



Home/Student Edition

Page 2

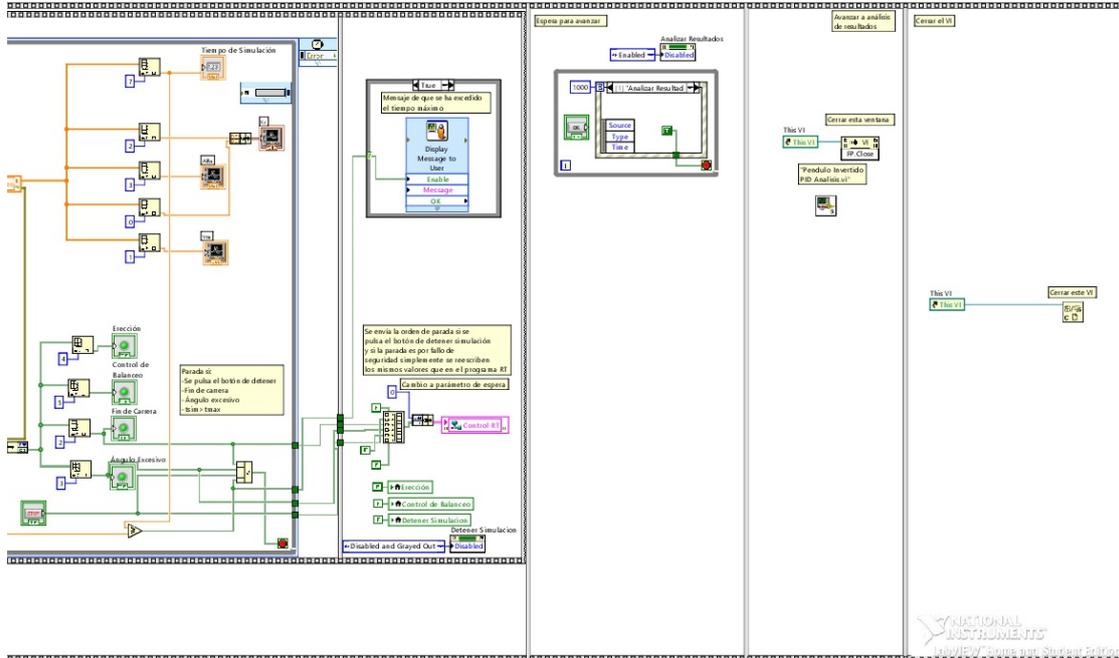


Monitorización del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID Monitorizacion.vi

Last modified on 24/09/2018 at 19:18

Printed on 24/09/2018 at 19:18



## 10. Pendulo Invertido SS Monitorizacion.vi

Home/Student Edition

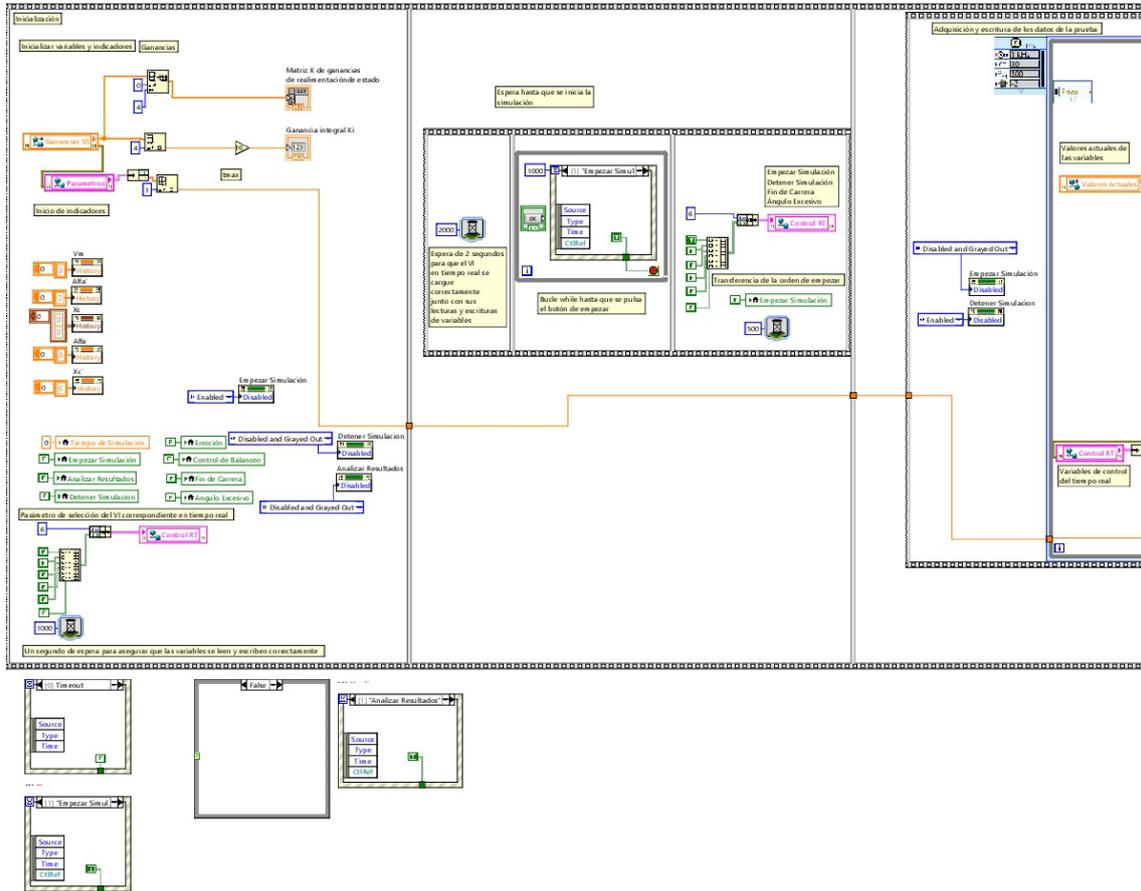
Monitorización del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS Monitorizacion.vi

Last modified on 24/09/2018 at 19:20

Printed on 24/09/2018 at 19:21

Page 1



Home/Student Edition

Page 2

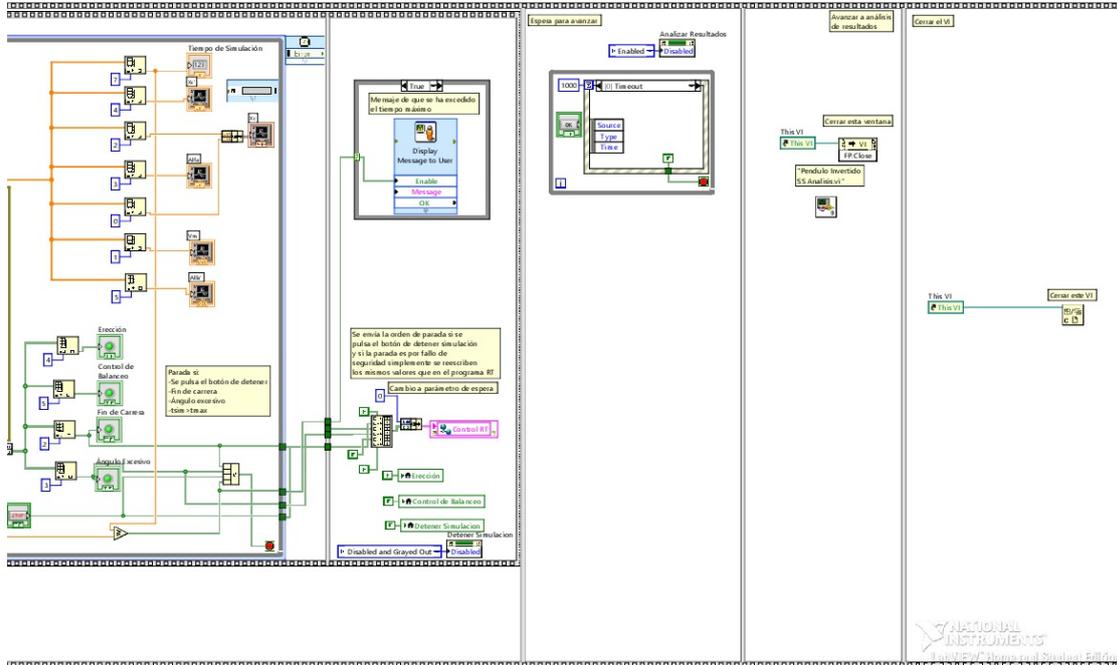


Monitorización del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS Monitorizacion.vi

Last modified on 24/09/2018 at 19:20

Printed on 24/09/2018 at 19:21



## 11. Carro PID Analysis.vi

Home/Student Edition

Page 1

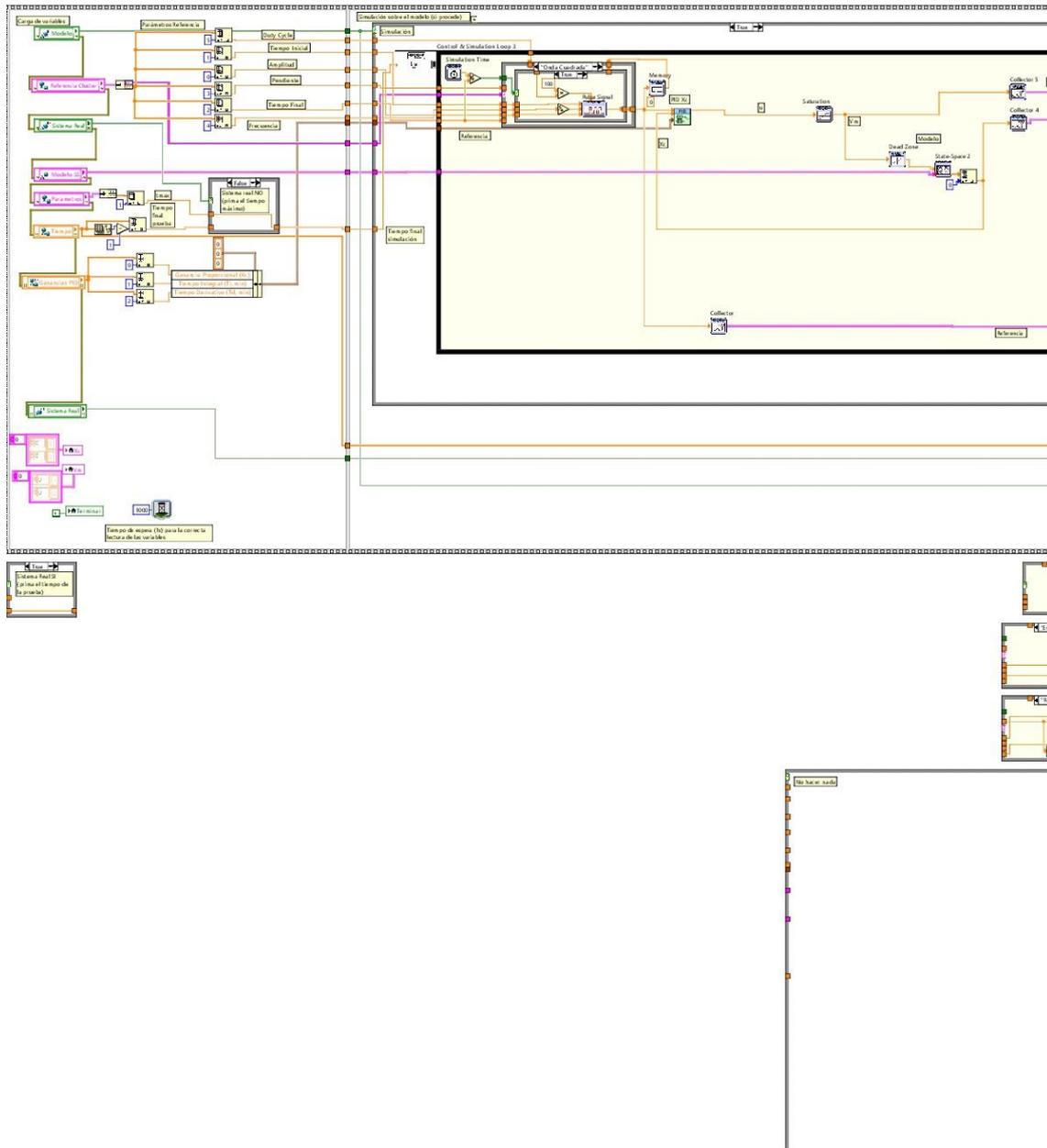


Analisis del Carro

G:\Uni\TFM\LabView\Aplicacion\Carro PID Analysis.vi

Last modified on 20/09/2018 at 12:03

Printed on 24/09/2018 at 19:23



Home/Student Edition

Page 2

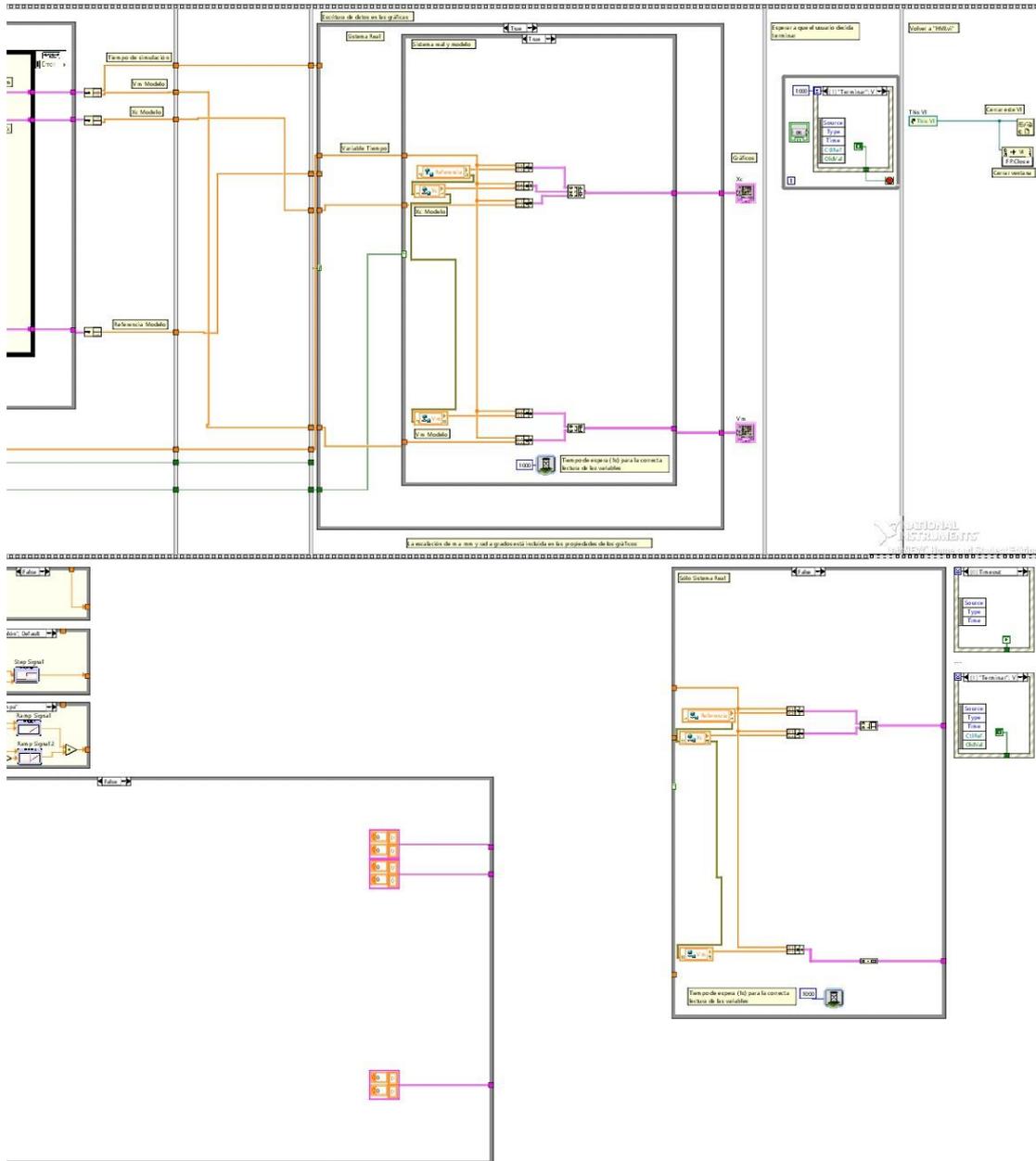


Analisis del Carro

G:\Uni\TFM\LabView\Aplicacion\Carro PID Analisis.vi

Last modified on 20/09/2018 at 12:03

Printed on 24/09/2018 at 19:23



Home/Student Edition

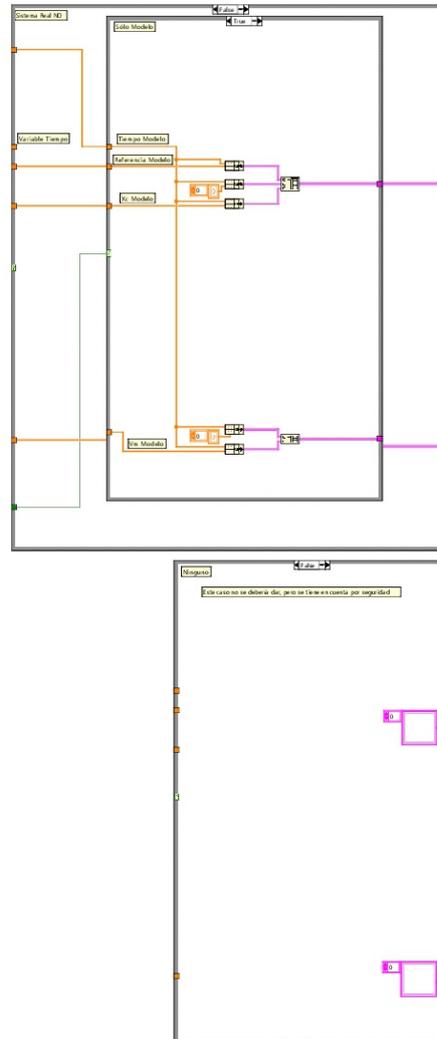
Analisis del Carro

G:\Uni\TFM\LabView\Aplicacion\Carro PID Analisis.vi

Last modified on 20/09/2018 at 12:03

Printed on 24/09/2018 at 19:23

Page 3



## 12. Carro SS Analisis.vi

Home/Student Edition

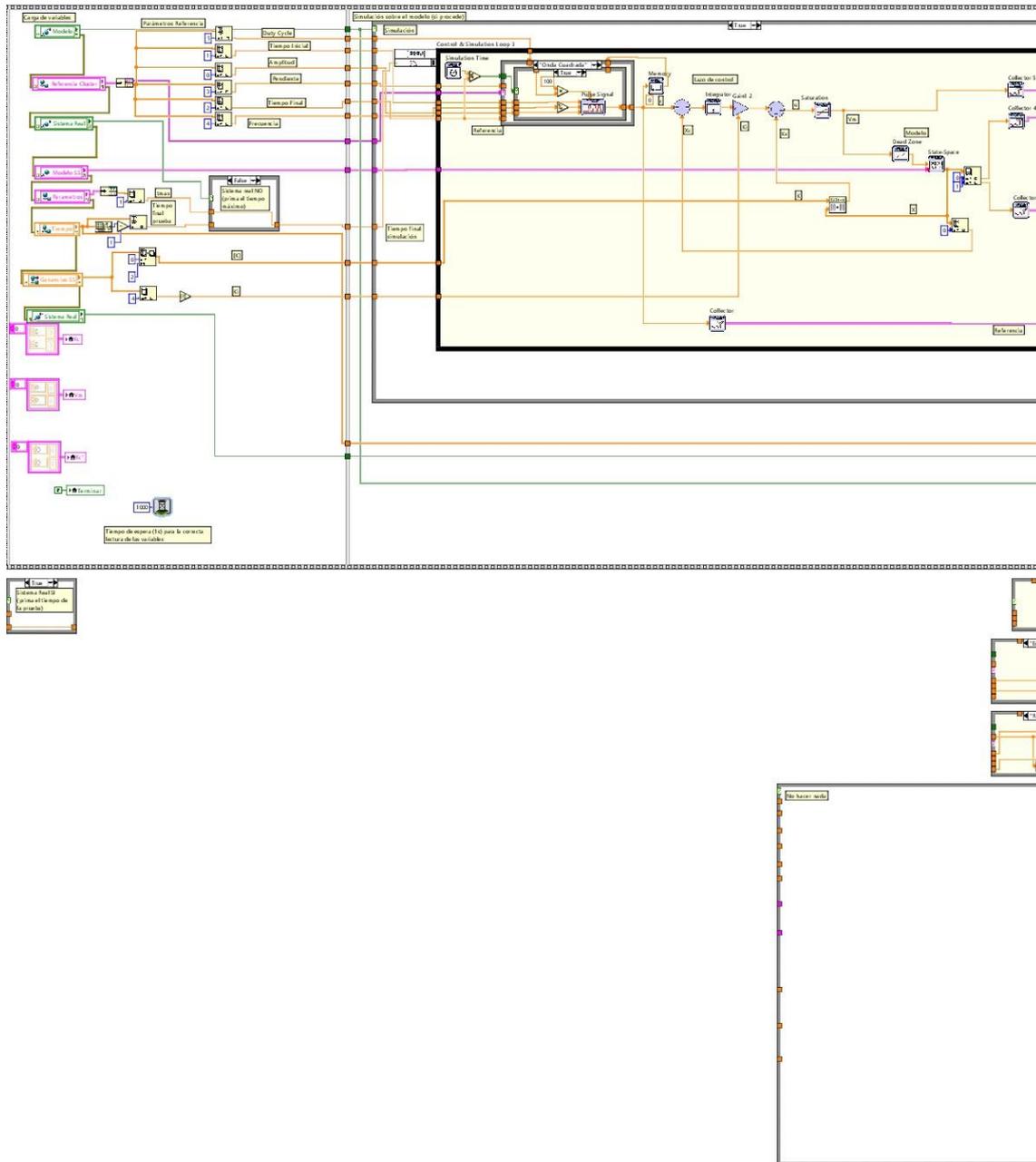
Análisis del Sistema Carro

G:\Uni\TFM\LabView\Aplicacion\Carro SS Analisis.vi

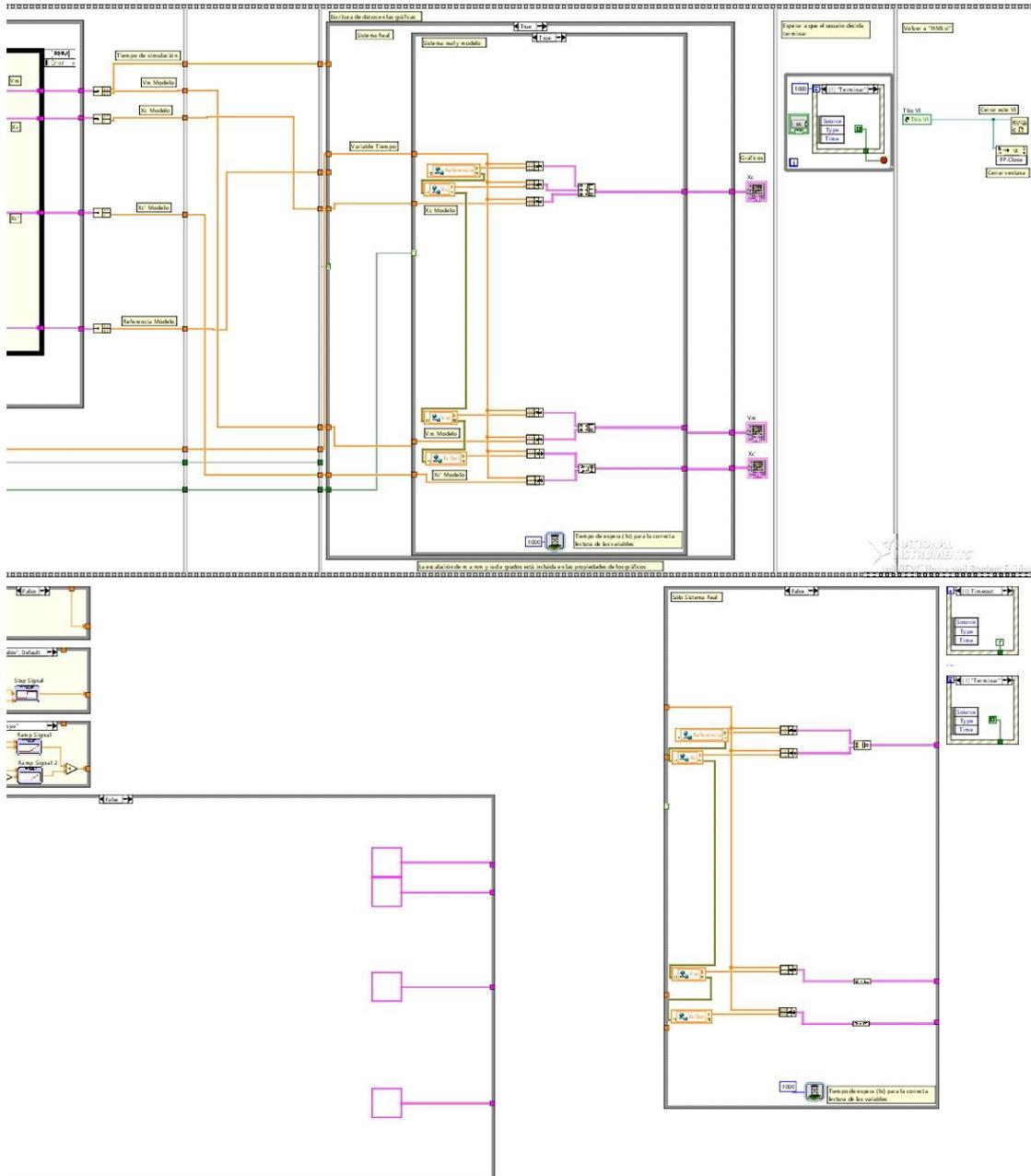
Last modified on 17/09/2018 at 11:09

Printed on 24/09/2018 at 19:25

Page 1



Home/Student Edition  
 Análisis del Sistema Carro  
 G:\Uni\TFM\LabView\Aplicacion\Carro SS Analysis.vi  
 Last modified on 17/09/2018 at 11:09  
 Printed on 24/09/2018 at 19:25





## 13. Pendulo Gantry PID Analysis.vi

Home/Student Edition

Page 1

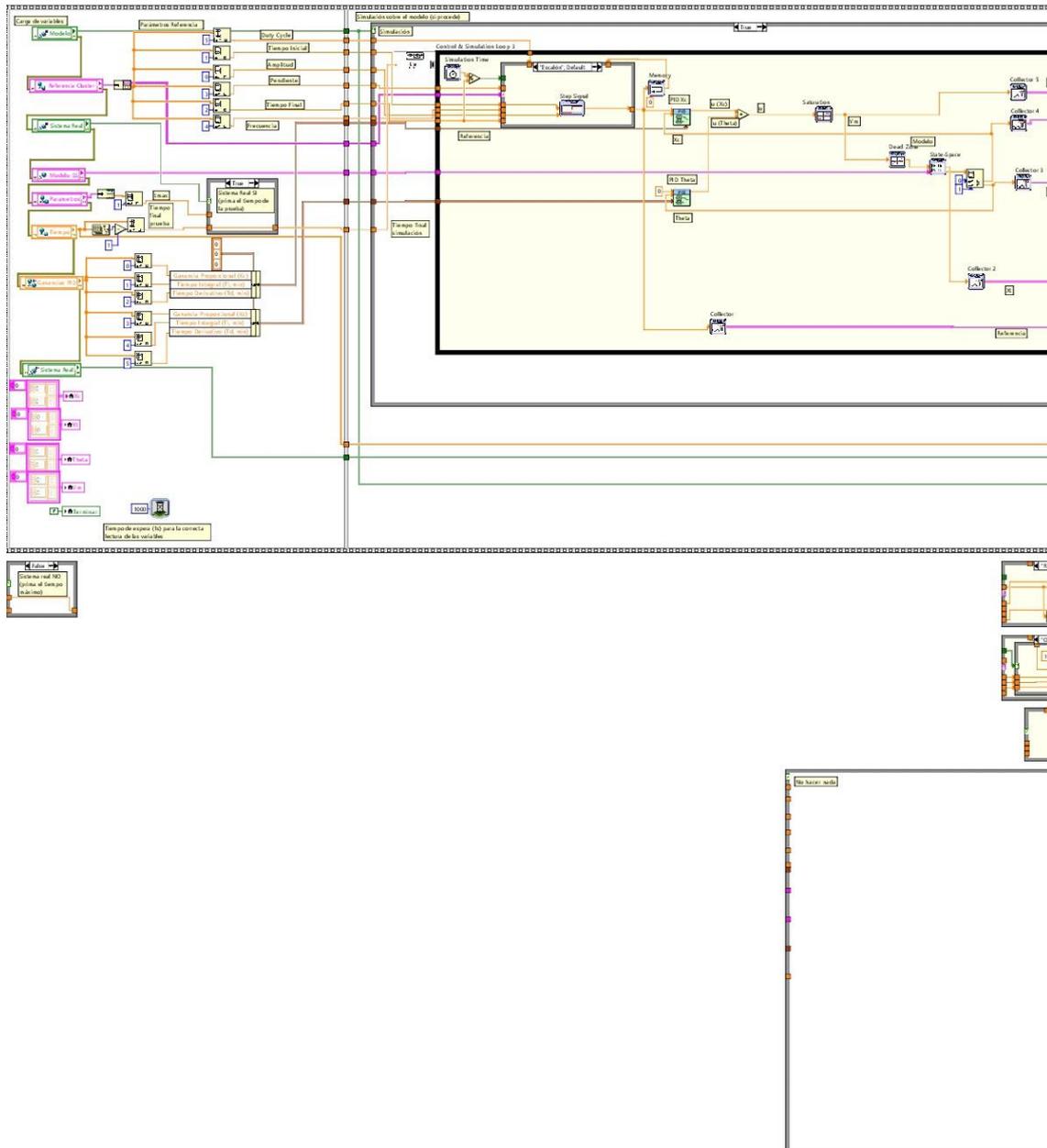


Análisis del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry PID Analysis.vi

Last modified on 17/09/2018 at 11:12

Printed on 24/09/2018 at 19:29



Home/Student Edition

Page 2

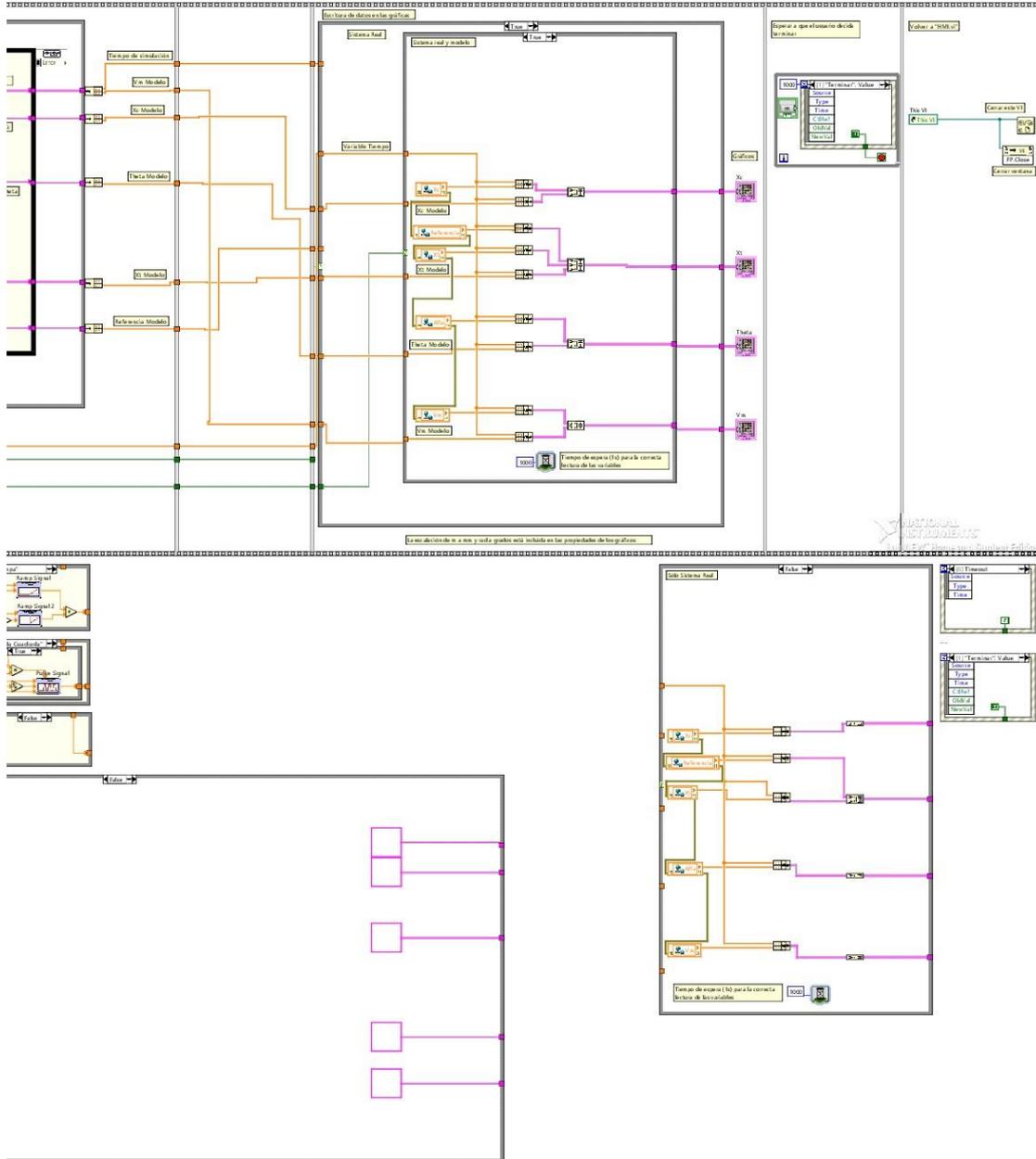


Análisis del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry PID Analisis.vi

Last modified on 17/09/2018 at 11:12

Printed on 24/09/2018 at 19:29



Home/Student Edition

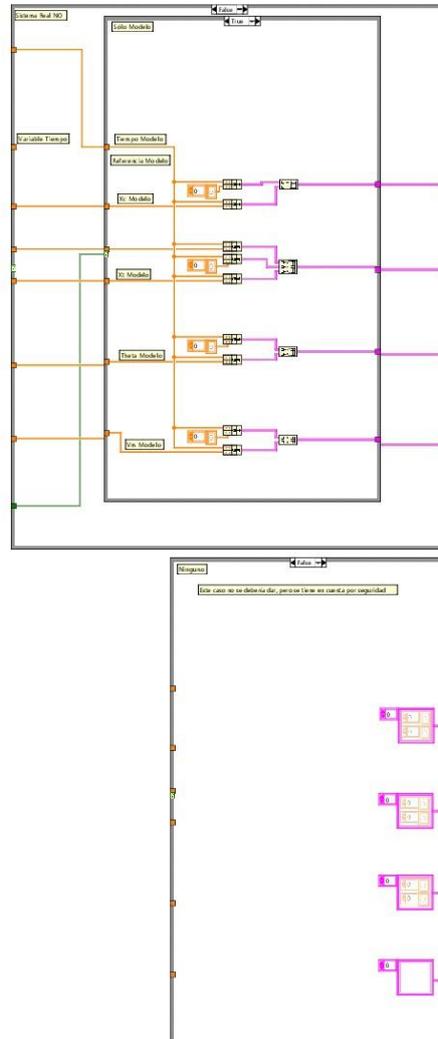
Análisis del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry PID Analisis.vi

Last modified on 17/09/2018 at 11:12

Printed on 24/09/2018 at 19:29

Page 3



## 14. Pendulo Gantry SS Analysis.vi

Home/Student Edition

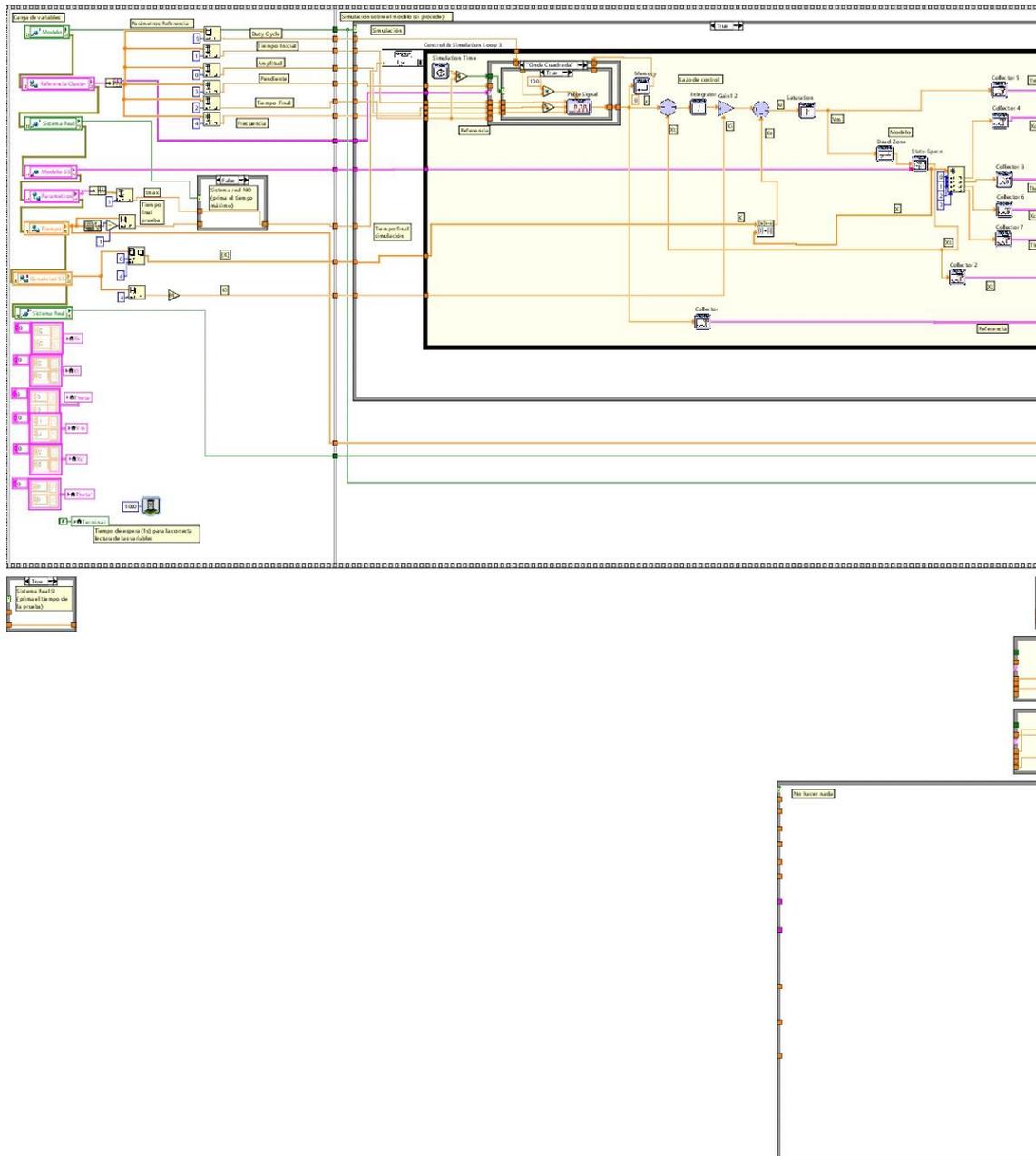
Análisis del Pendulo Gantry

G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry SS Analysis.vi

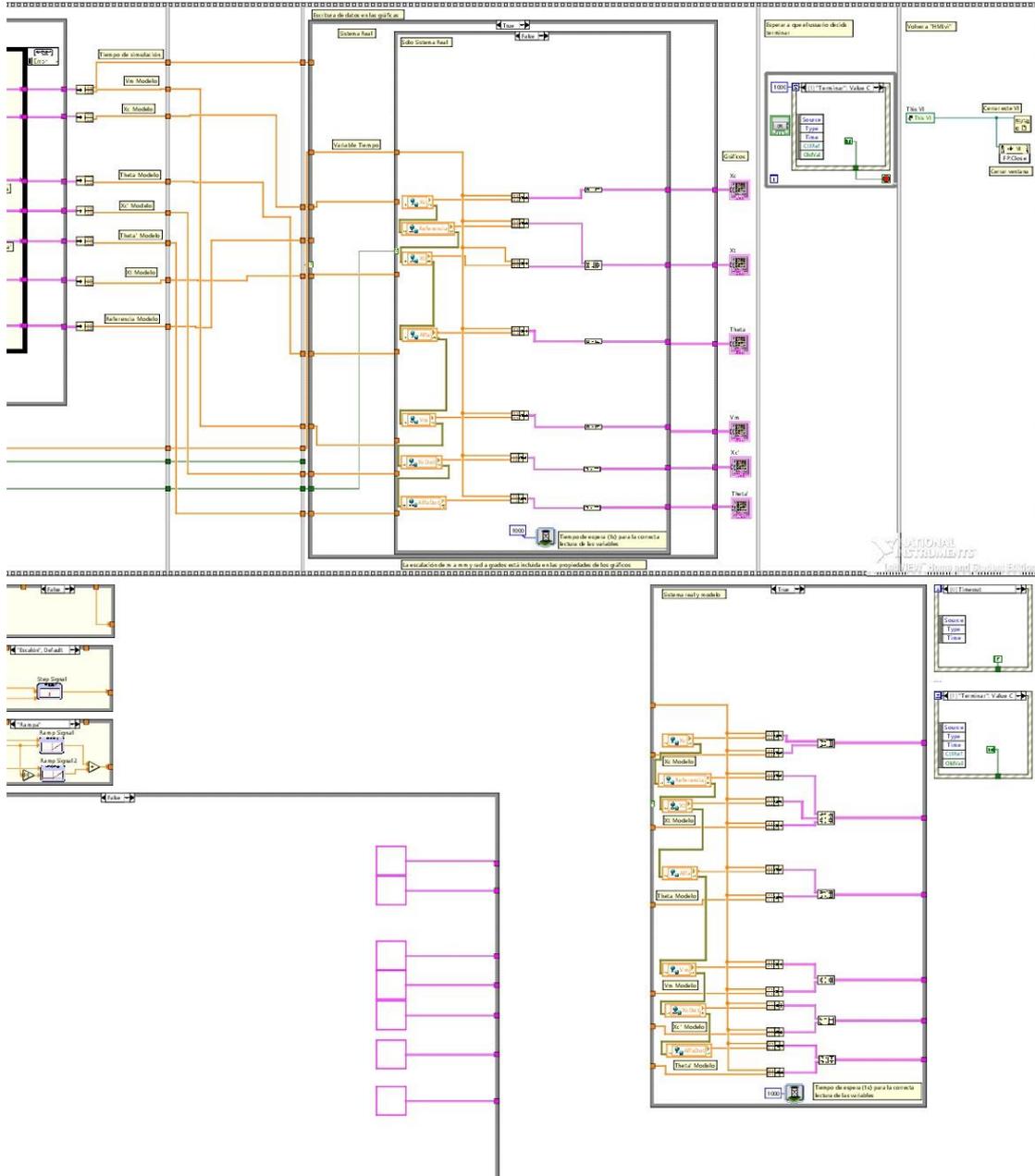
Last modified on 17/09/2018 at 11:12

Printed on 24/09/2018 at 19:31

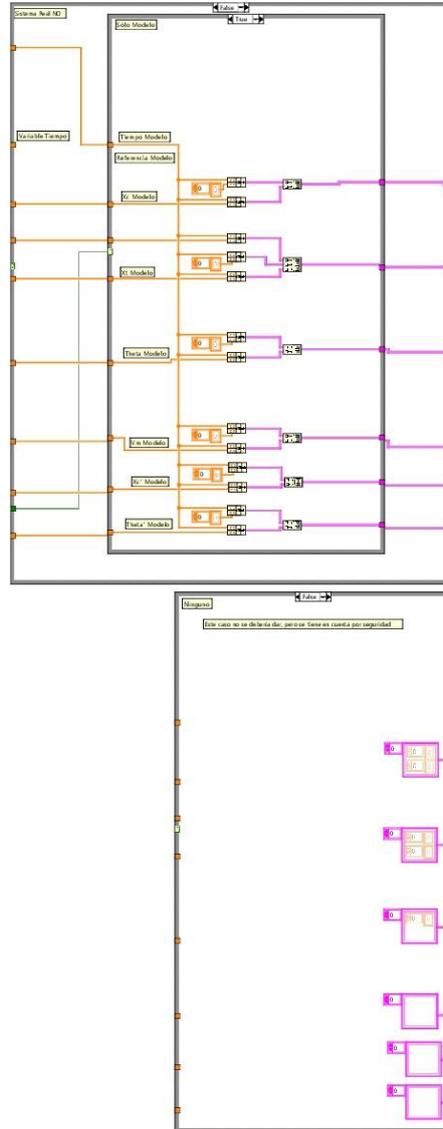
Page 1



Home/Student Edition  
 Análisis del Pendulo Gantry  
 G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry SS Analisis.vi  
 Last modified on 17/09/2018 at 11:12  
 Printed on 24/09/2018 at 19:31



Home/Student Edition  
Análisis del Pendulo Gantry  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry SS Analisis.vi  
Last modified on 17/09/2018 at 11:12  
Printed on 24/09/2018 at 19:31



## 15. Pendulo Invertido PID Analisis.vi

Home/Student Edition

Page 1

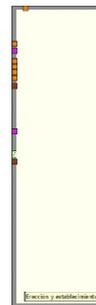
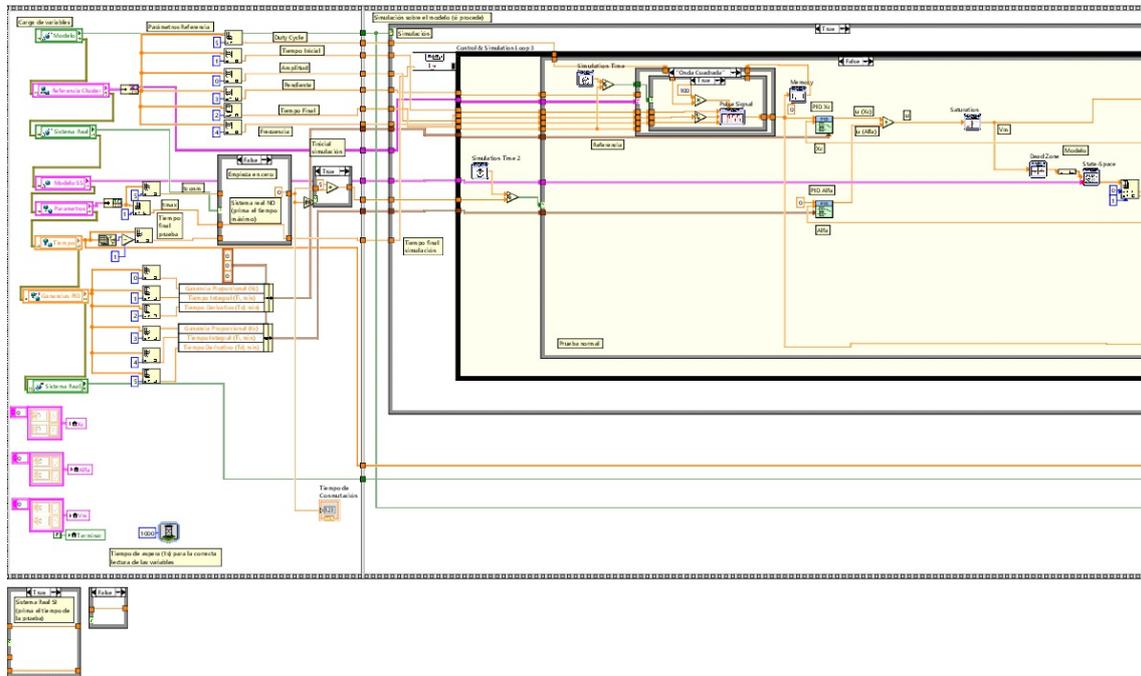


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID Analisis.vi

Last modified on 21/09/2018 at 16:26

Printed on 24/09/2018 at 19:34



Home/Student Edition

Page 2

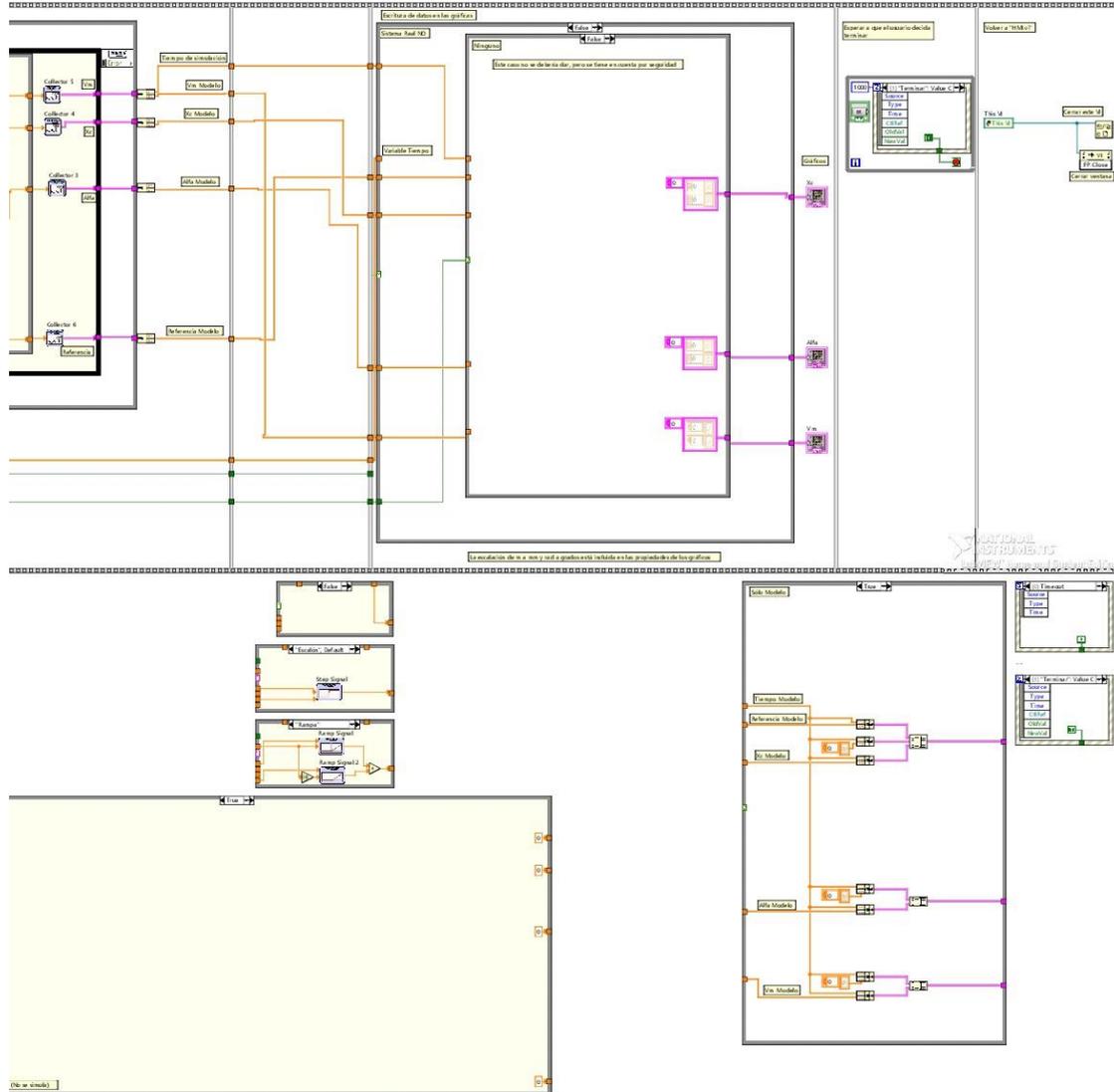


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido PID Analisis.vi

Last modified on 21/09/2018 at 16:26

Printed on 24/09/2018 at 19:34



Home/Student Edition

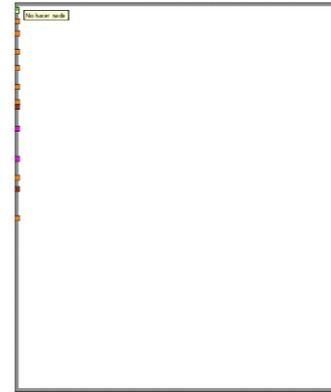
Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido PID Analisis.vi

Last modified on 21/09/2018 at 16:26

Printed on 24/09/2018 at 19:34

Page 3



Home/Student Edition

Page 4

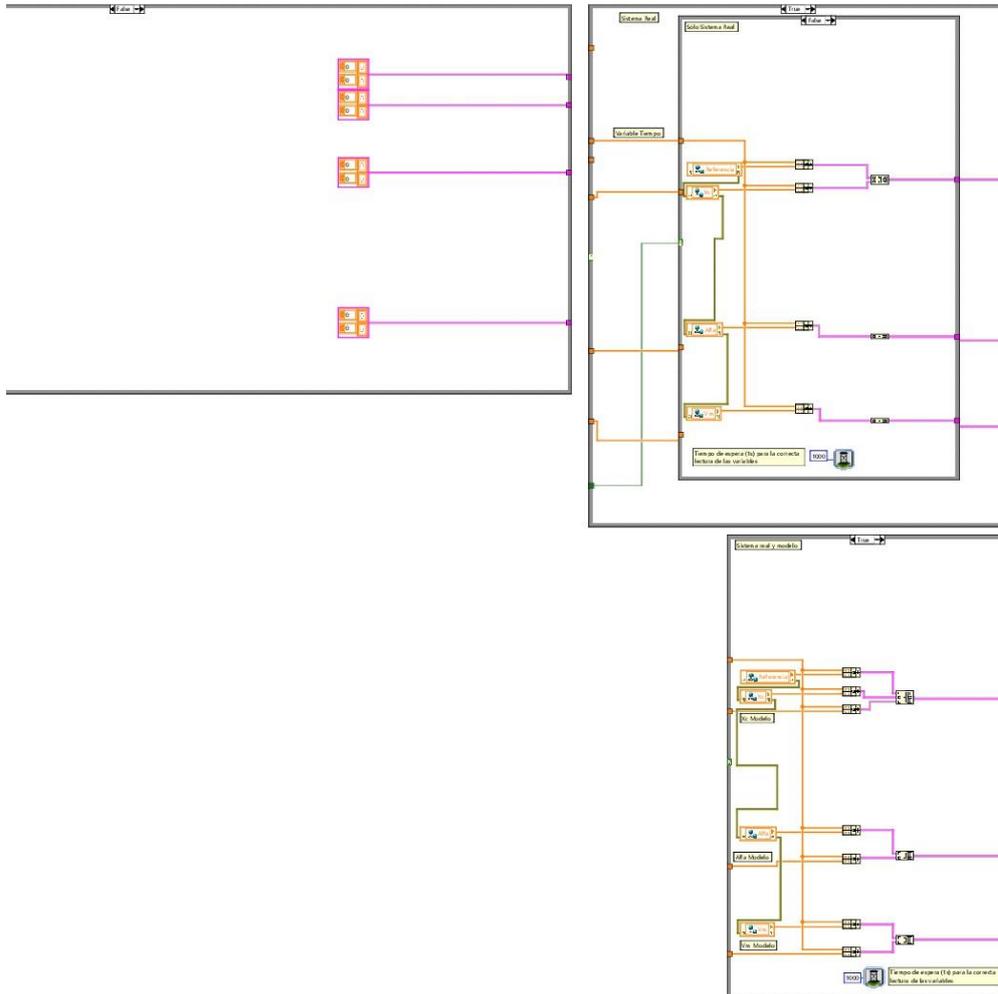


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido PID Analisis.vi

Last modified on 21/09/2018 at 16:26

Printed on 24/09/2018 at 19:34



## 16. Pendulo Invertido SS Analysis.vi

Home/Student Edition

Page 1

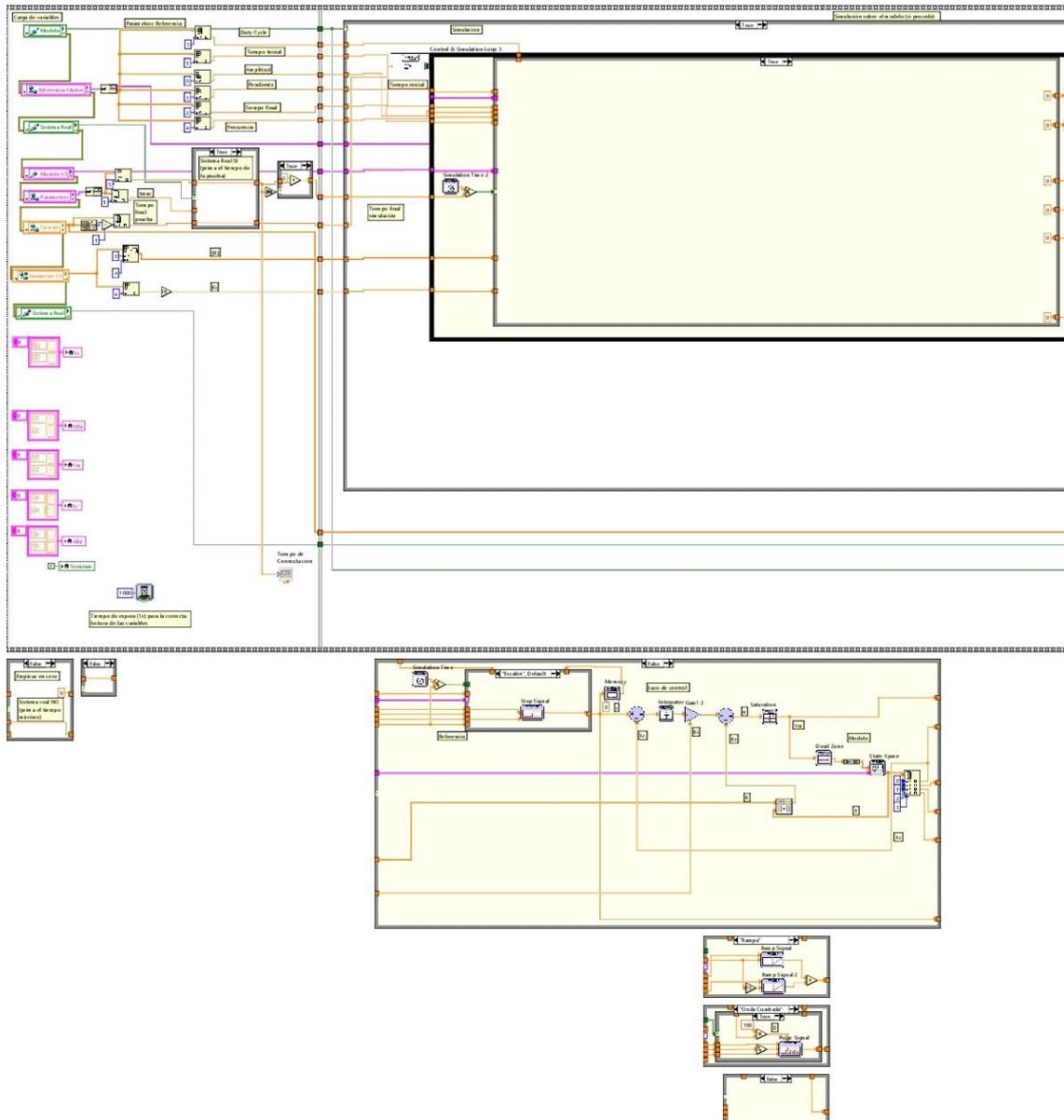


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido SS Analisis.vi

Last modified on 21/09/2018 at 16:31

Printed on 24/09/2018 at 19:36



Home/Student Edition

Page 2

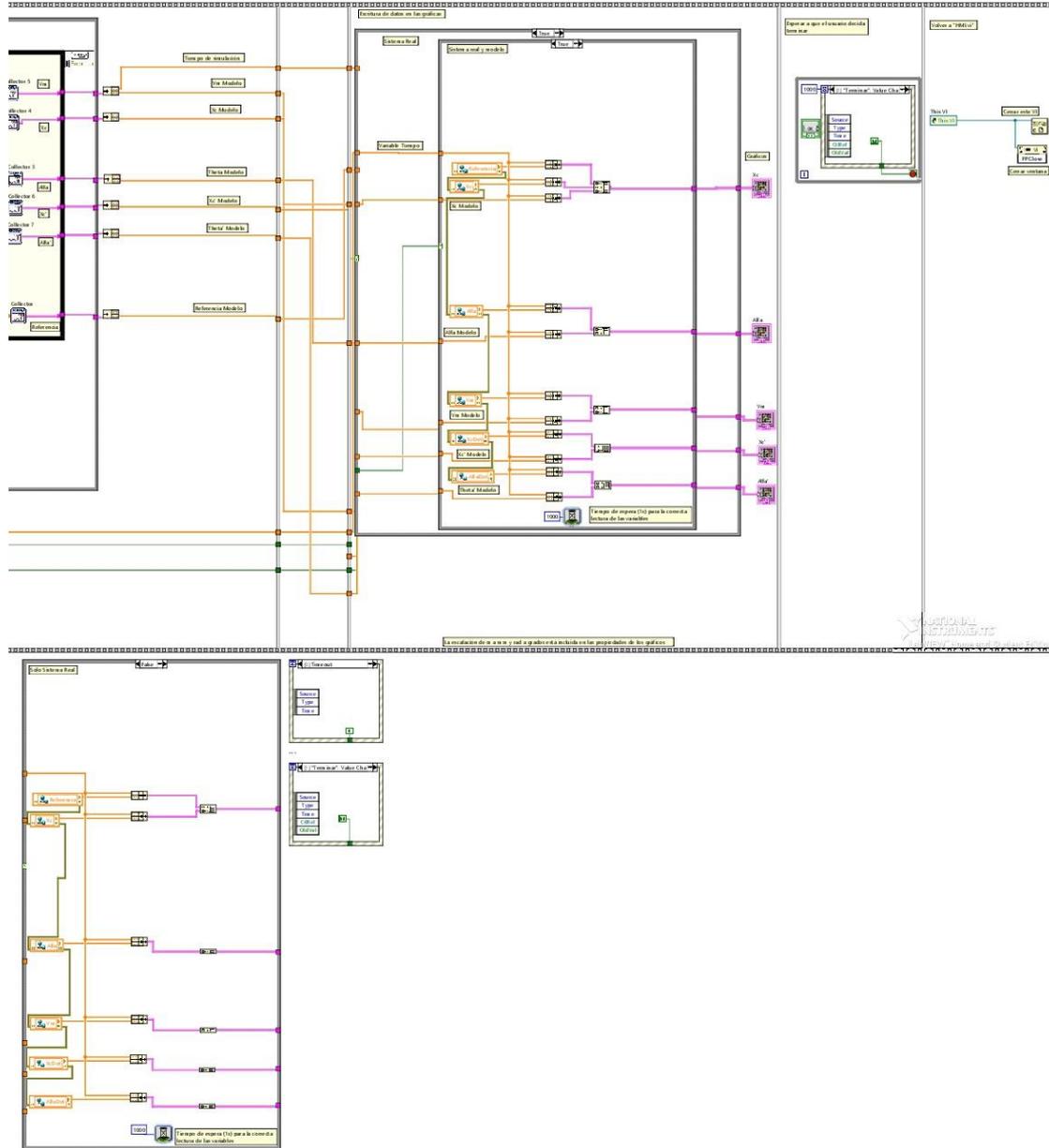


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS Analisis.vi

Last modified on 21/09/2018 at 16:31

Printed on 24/09/2018 at 19:36



Home/Student Edition

Page 3

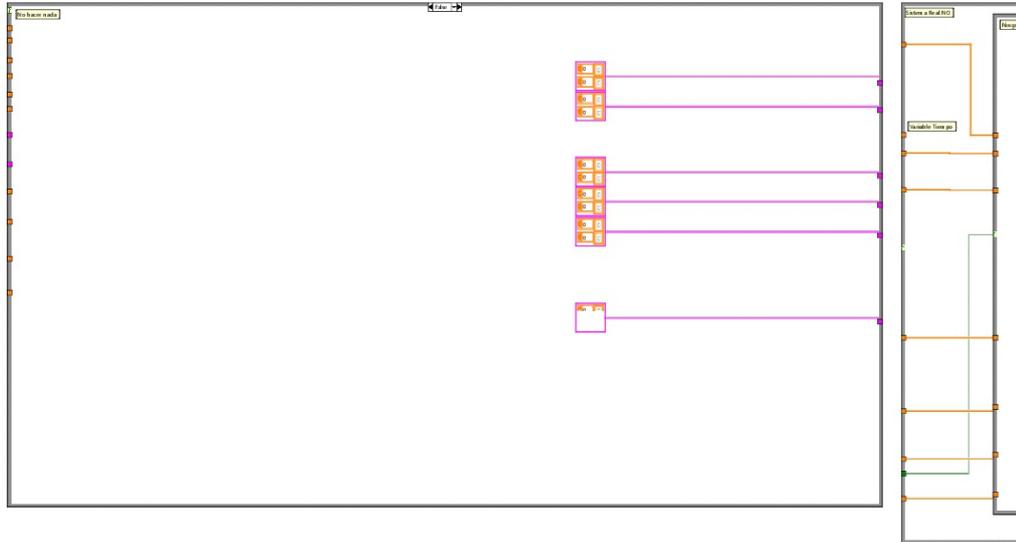


Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido SS Analisis.vi

Last modified on 21/09/2018 at 16:31

Printed on 24/09/2018 at 19:36



Home/Student Edition

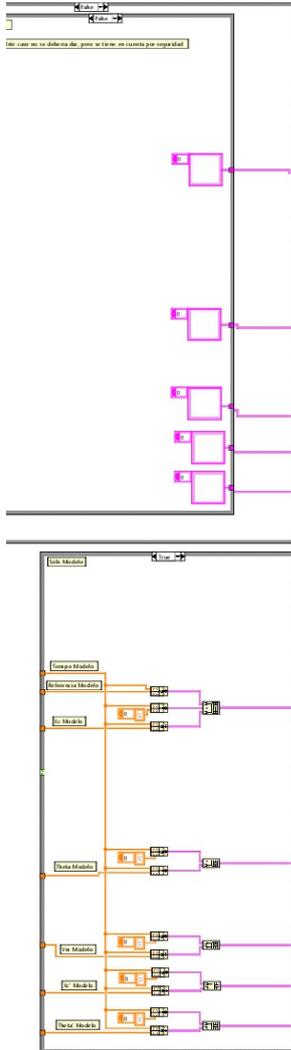
Análisis del Pendulo Invertido

G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido SS Analisis.vi

Last modified on 21/09/2018 at 16:31

Printed on 24/09/2018 at 19:36

Page 4



## 17. Inicio.vi

Home/Student Edition

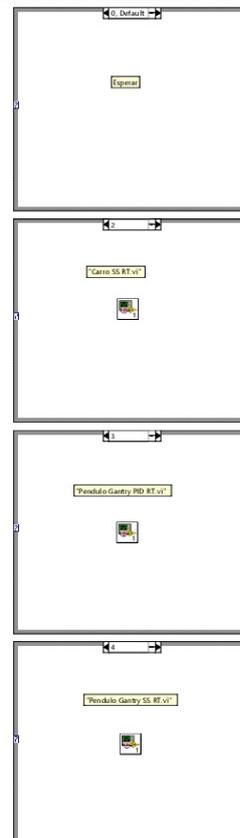
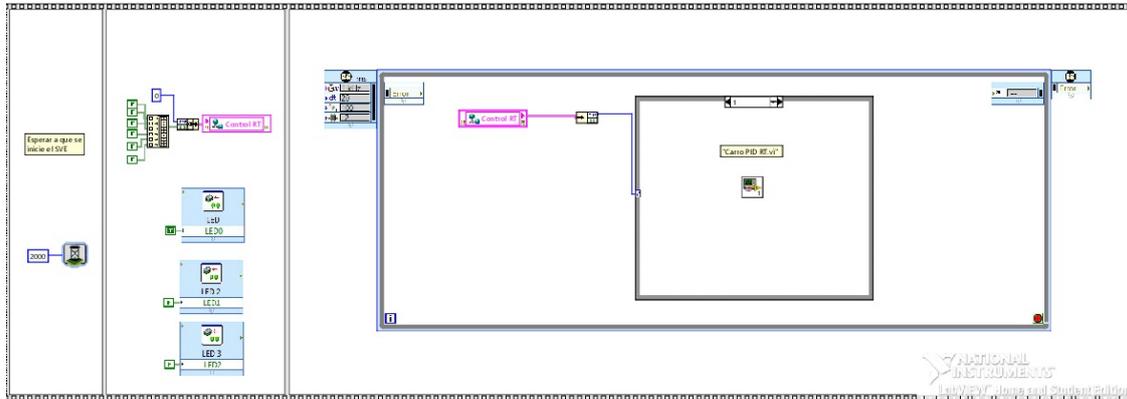
Inicio.vi

G:\Uni\TFM\LabView\Aplicacion\Inicio.vi

Last modified on 21/09/2018 at 11:27

Printed on 24/09/2018 at 19:53

Page 1



Home/Student Edition

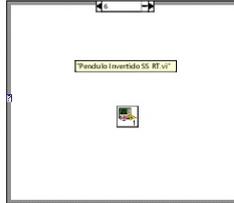
Inicio.vi

G:\Uni\TFM\LabView\Aplicacion\Inicio.vi

Last modified on 21/09/2018 at 11:27

Printed on 24/09/2018 at 19:53

Page 2



## 18. Carro PID RT.vi

Home/Student Edition

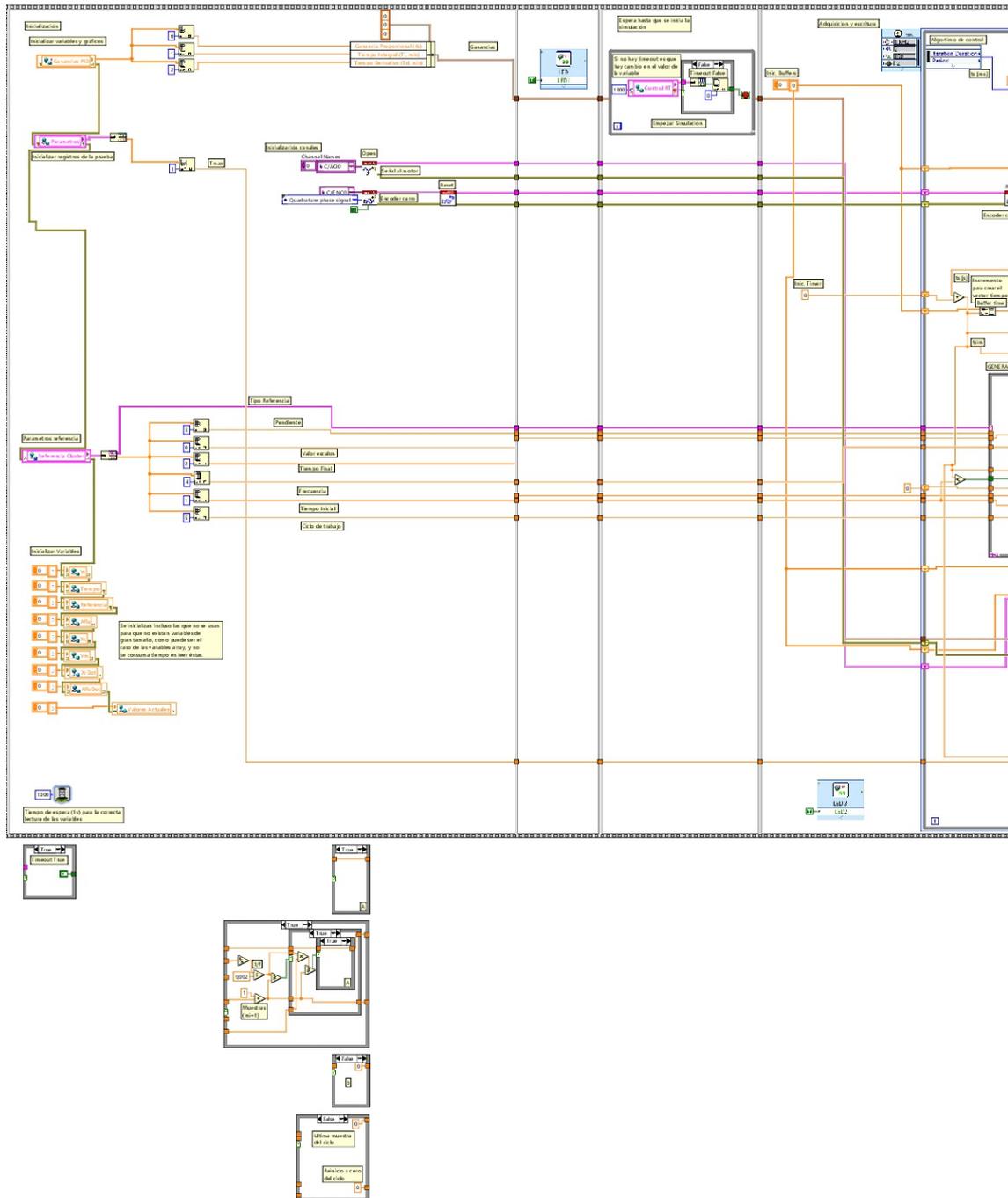
Carro PID RT.vi

G:\Uni\TFM\LabView\Aplicacion\Carro PID RT.vi

Last modified on 24/09/2018 at 19:54

Printed on 24/09/2018 at 19:55

Page 1



Home/Student Edition

Page 2

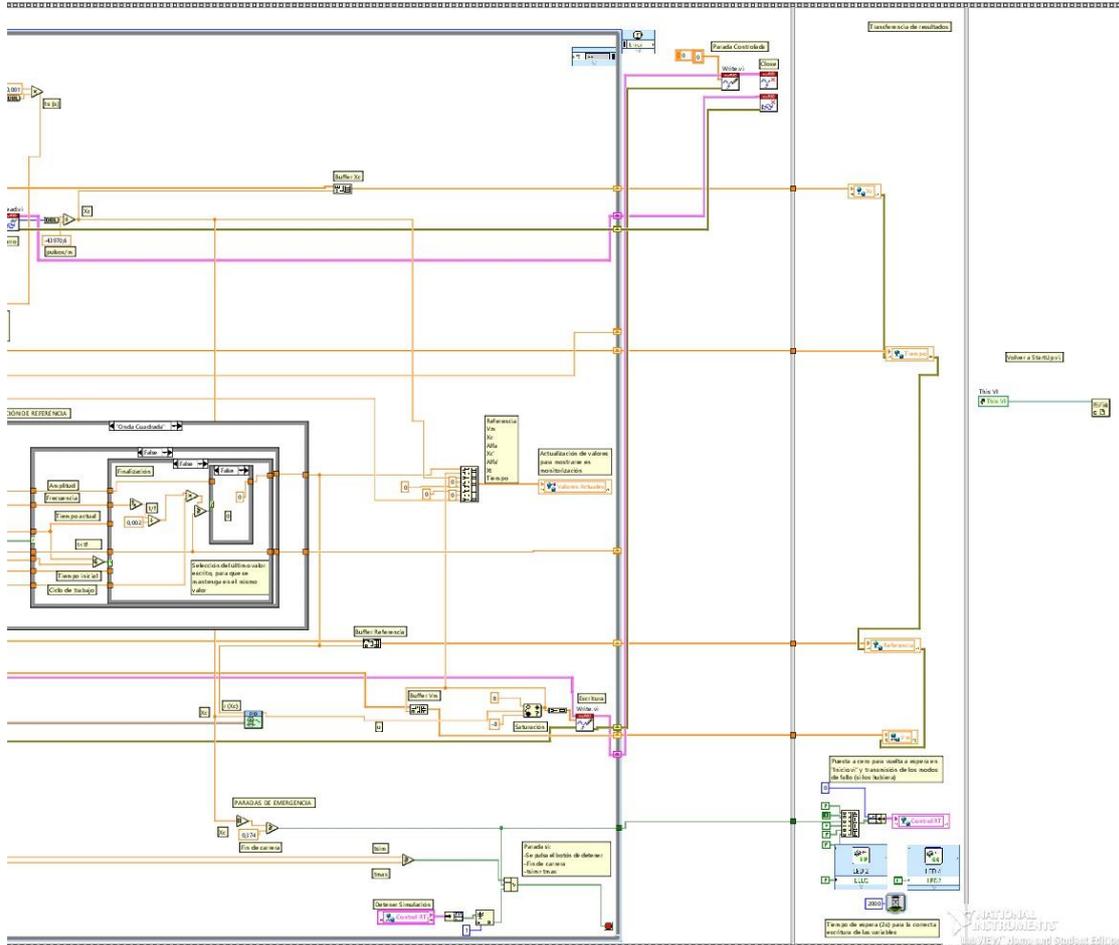


Carro PID RT.vi

G:\Uni\TFM\LabView\Aplicacion\Carro PID RT.vi

Last modified on 24/09/2018 at 19:54

Printed on 24/09/2018 at 19:55



Home/Student Edition

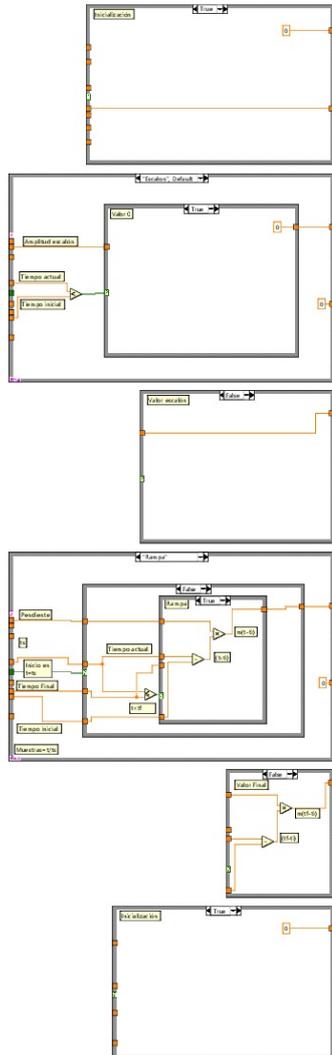
Carro PID RT.vi

G:\Uni\TFM\LabView\Aplicacion\Carro PID RT.vi

Last modified on 24/09/2018 at 19:54

Printed on 24/09/2018 at 19:55

Page 3



## 19. Carro SS RT.vi

Home/Student Edition

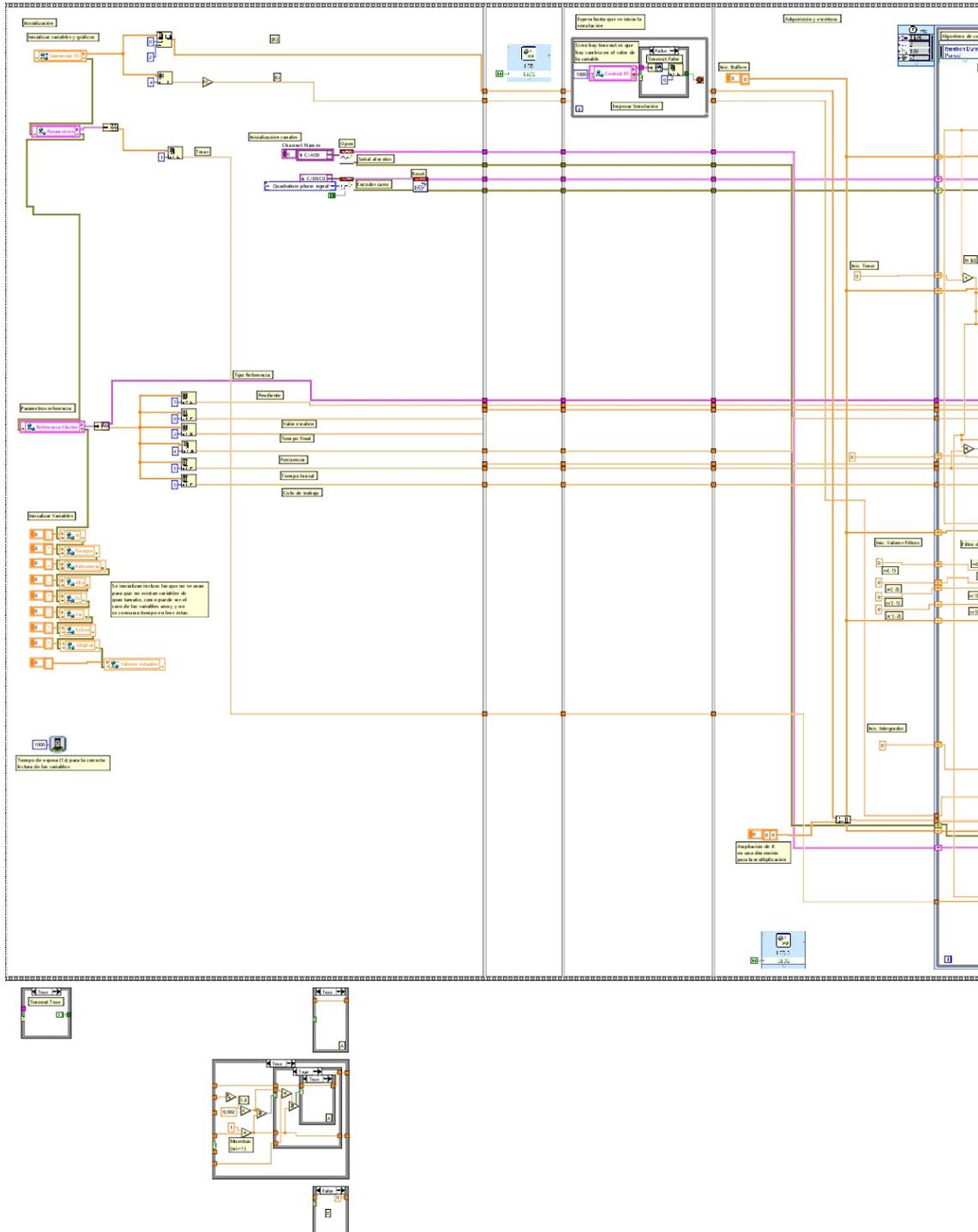
Carro SS RT.vi

G:\Uni\TFM\LabView\Aplicacion\Carro SS RT.vi

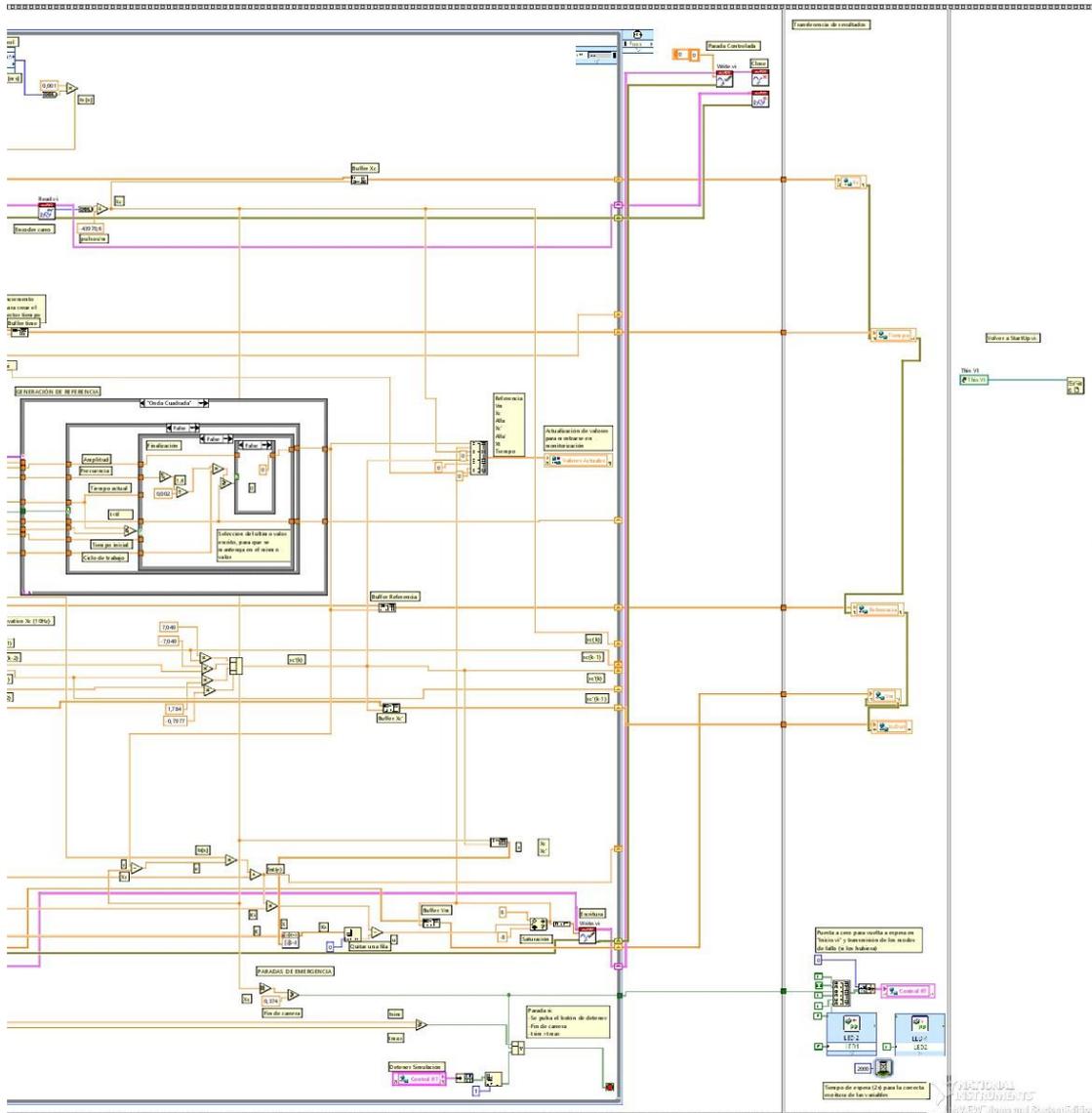
Last modified on 24/09/2018 at 19:54

Printed on 24/09/2018 at 19:57

Page 1



Home/Student Edition  
Carro SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Carro SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 19:58



Home/Student Edition

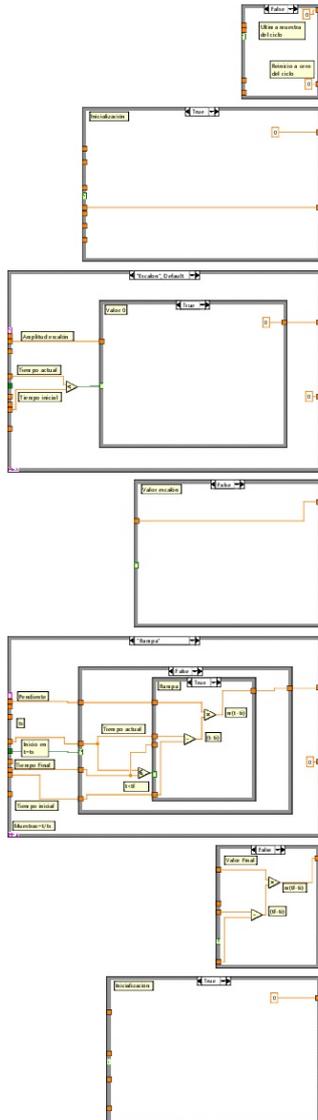
Carro SS RT.vi

G:\Uni\TFM\LabView\Aplicacion\Carro SS RT.vi

Last modified on 24/09/2018 at 19:54

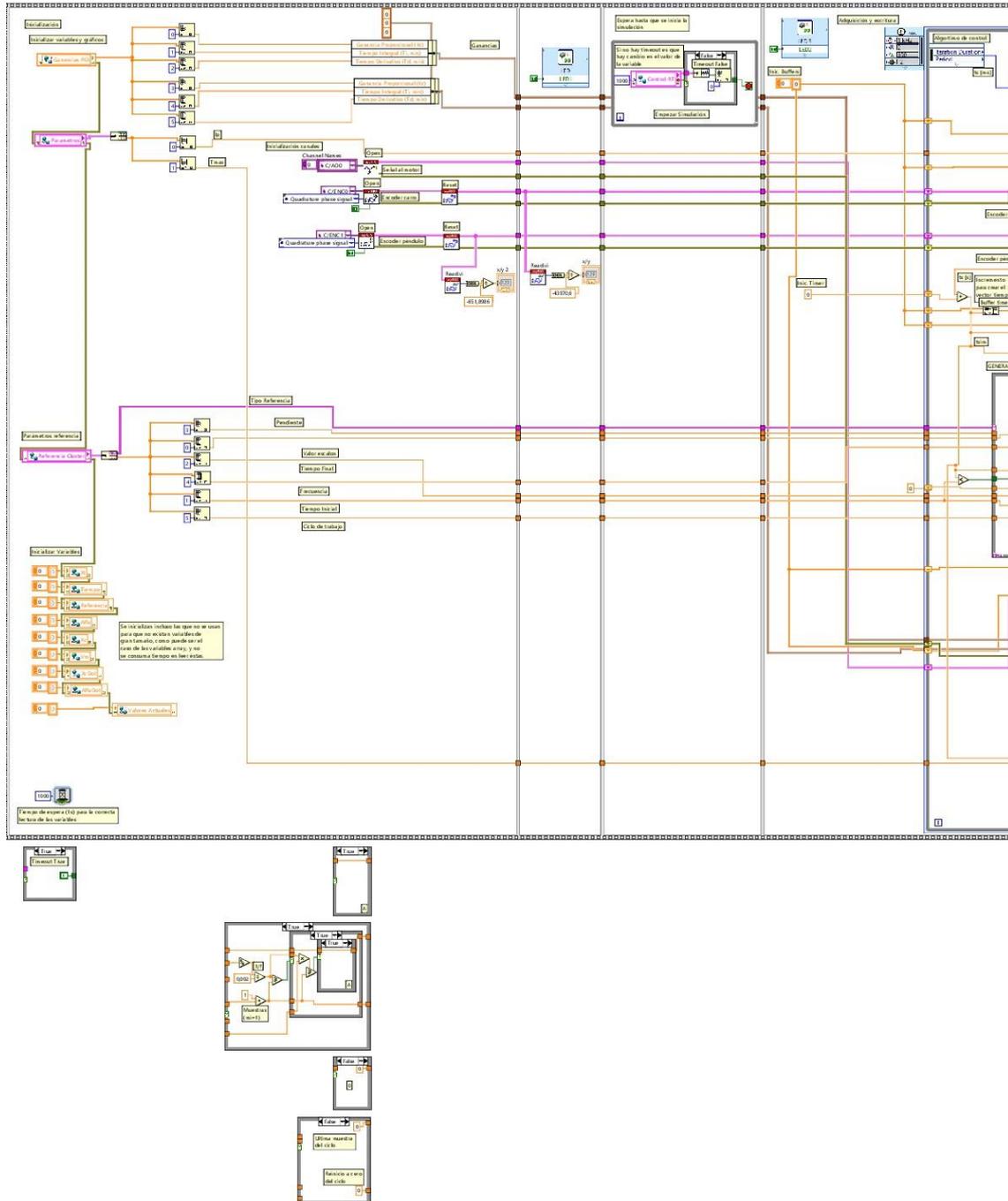
Printed on 24/09/2018 at 19:58

Page 3

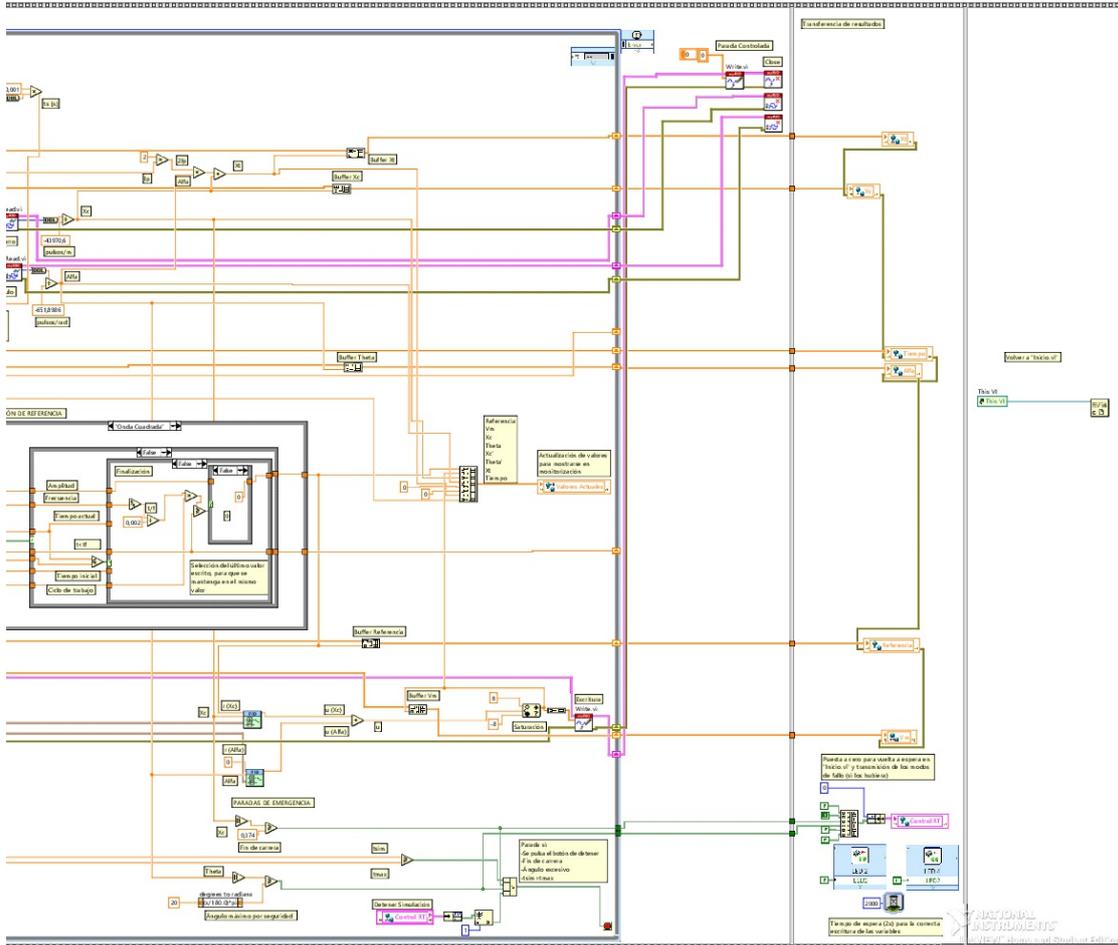


## 20. Pendulo Gantry PID RT.vi

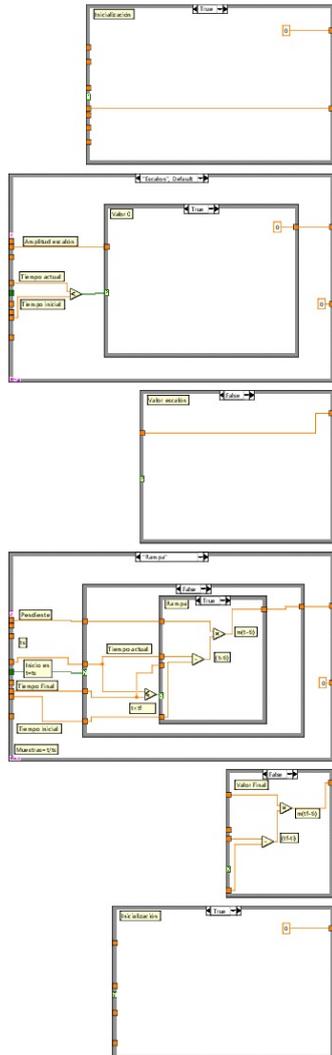
Home/Student Edition  
 Pendulo Gantry PID RT.vi  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry PID RT.vi  
 Last modified on 24/09/2018 at 19:54  
 Printed on 24/09/2018 at 20:00



Home/Student Edition  
Pendulo Gantry PID RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:00

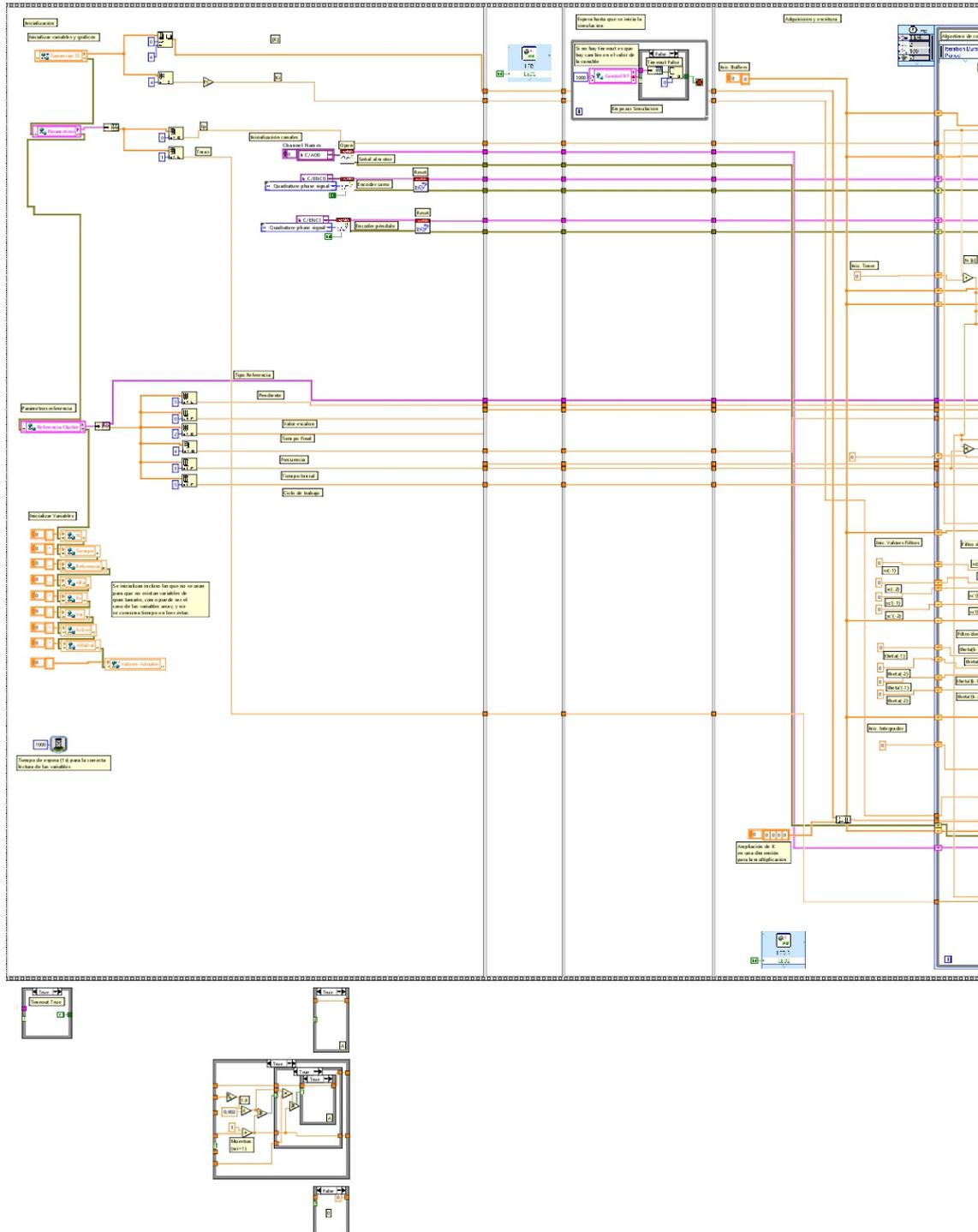


Home/Student Edition  
Pendulo Gantry PID RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Gantry PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:00

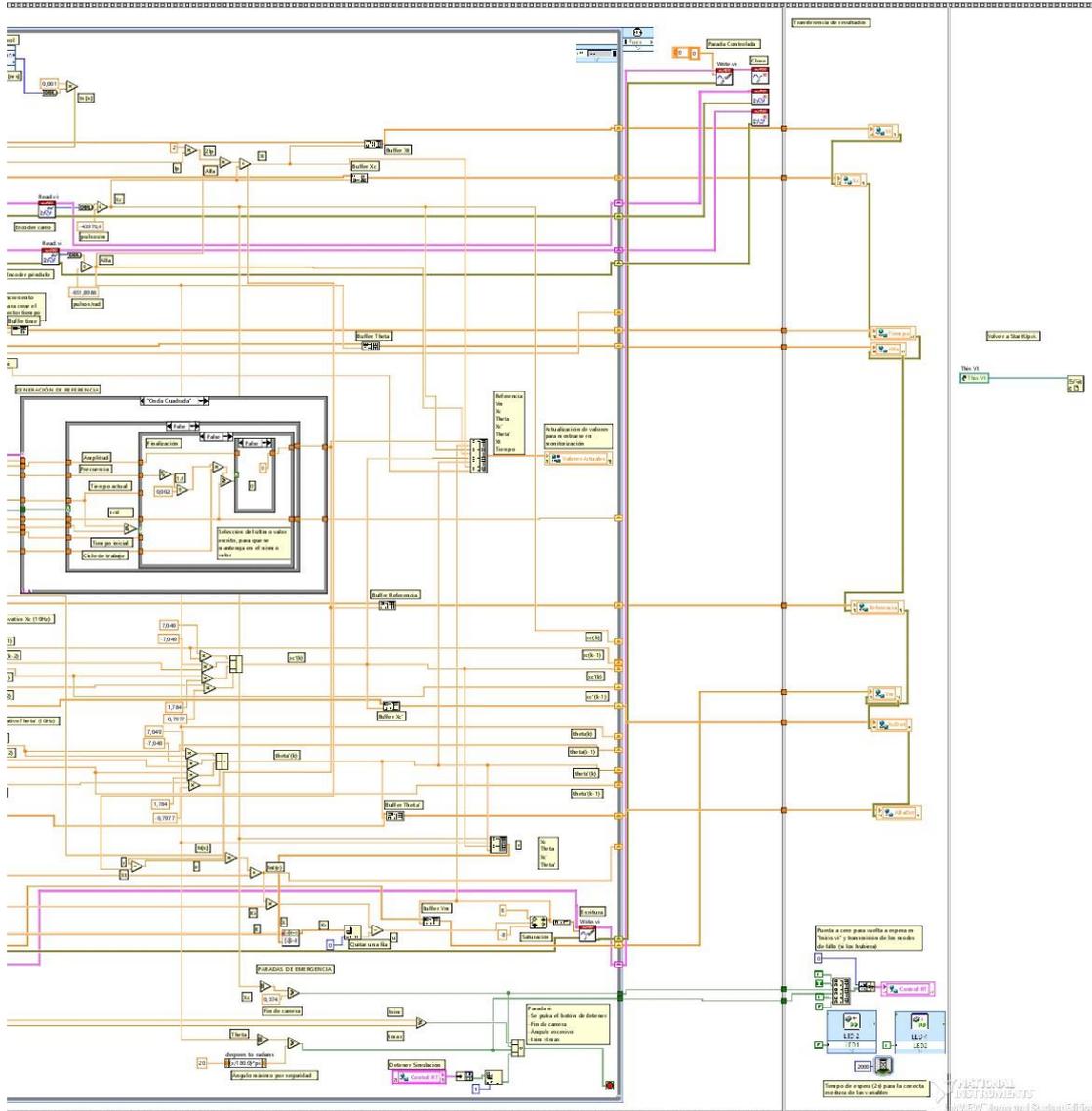


## 21. Pendulo Gantry SS RT.vi

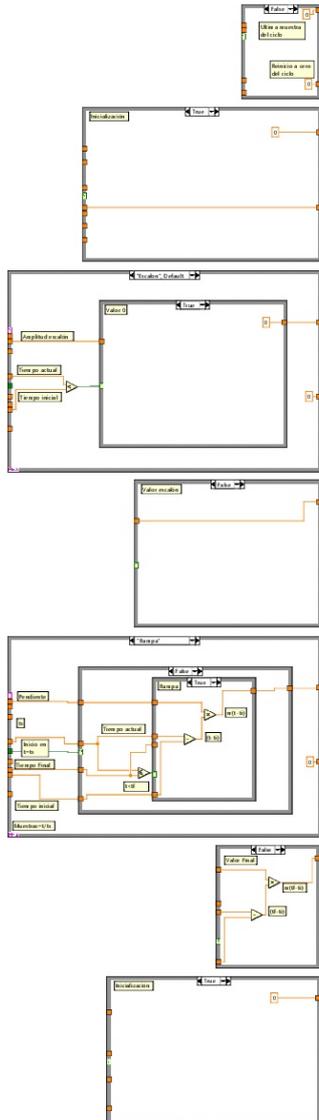
Home/Student Edition  
 Pendulo Gantry SS RT.vi  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry SS RT.vi  
 Last modified on 24/09/2018 at 19:54  
 Printed on 24/09/2018 at 20:03



Home/Student Edition  
Pendulo Gantry SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:03

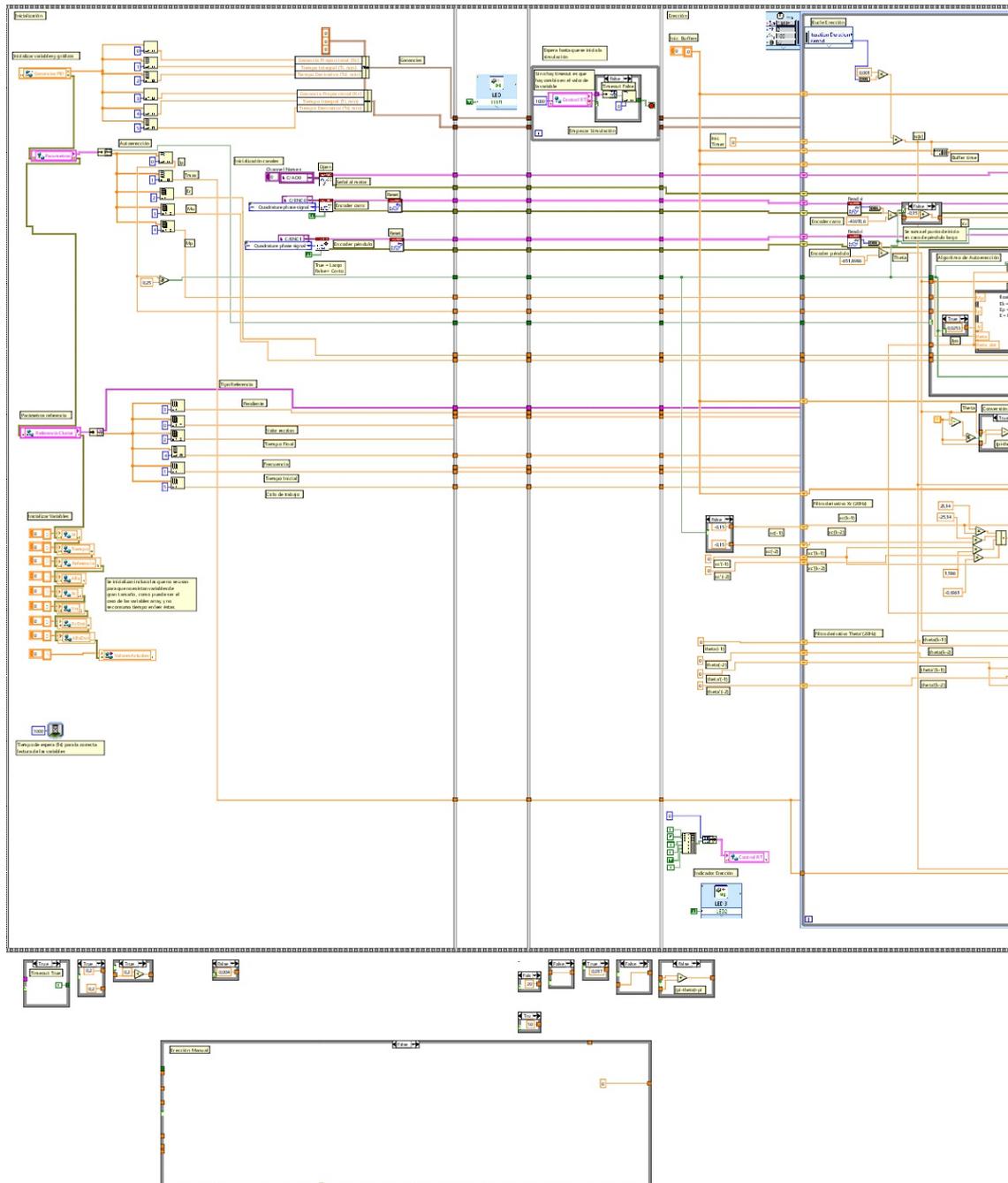


Home/Student Edition  
Pendulo Gantry SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Gantry SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:03

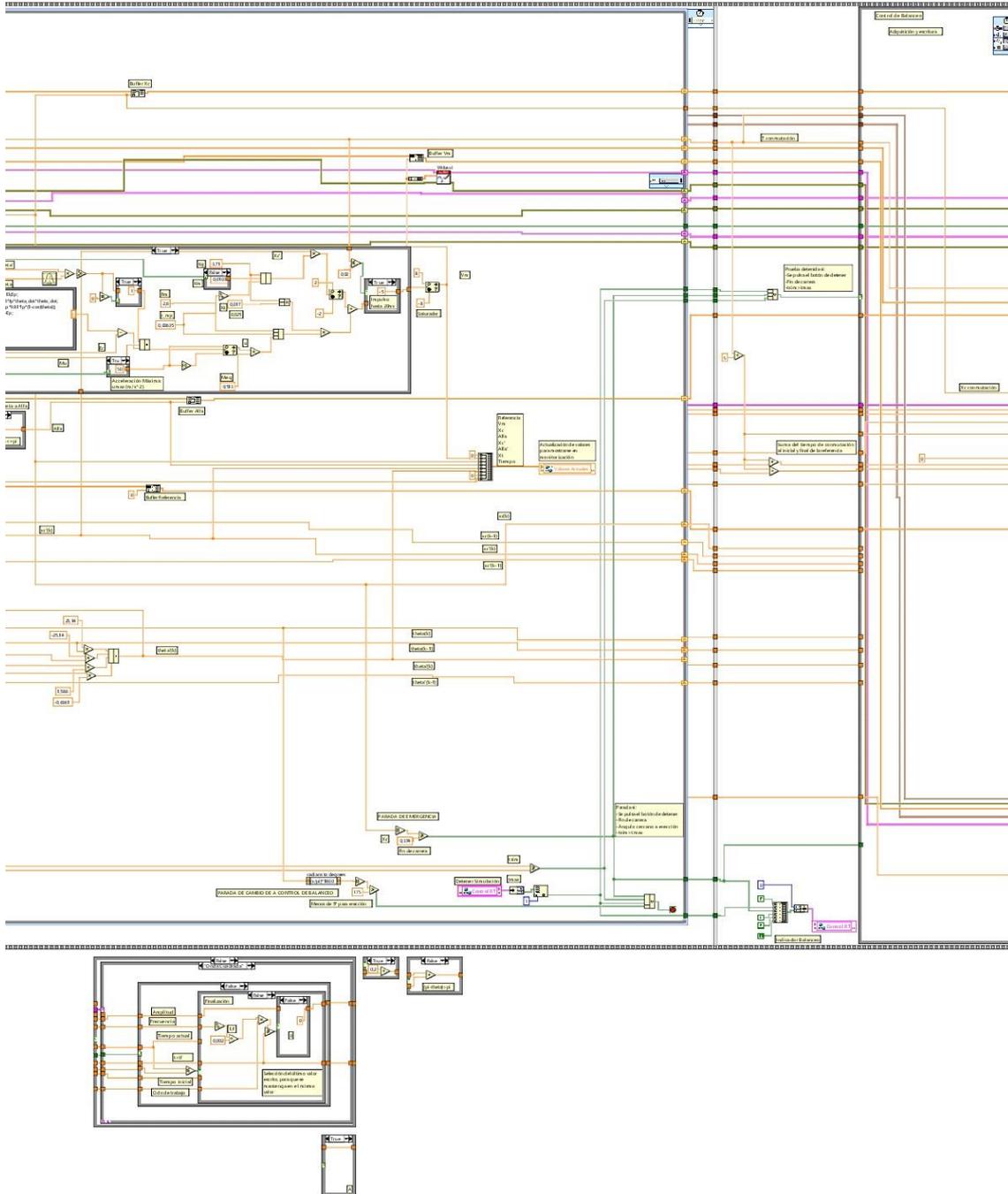


## 22. Pendulo Invertido PID RT.vi

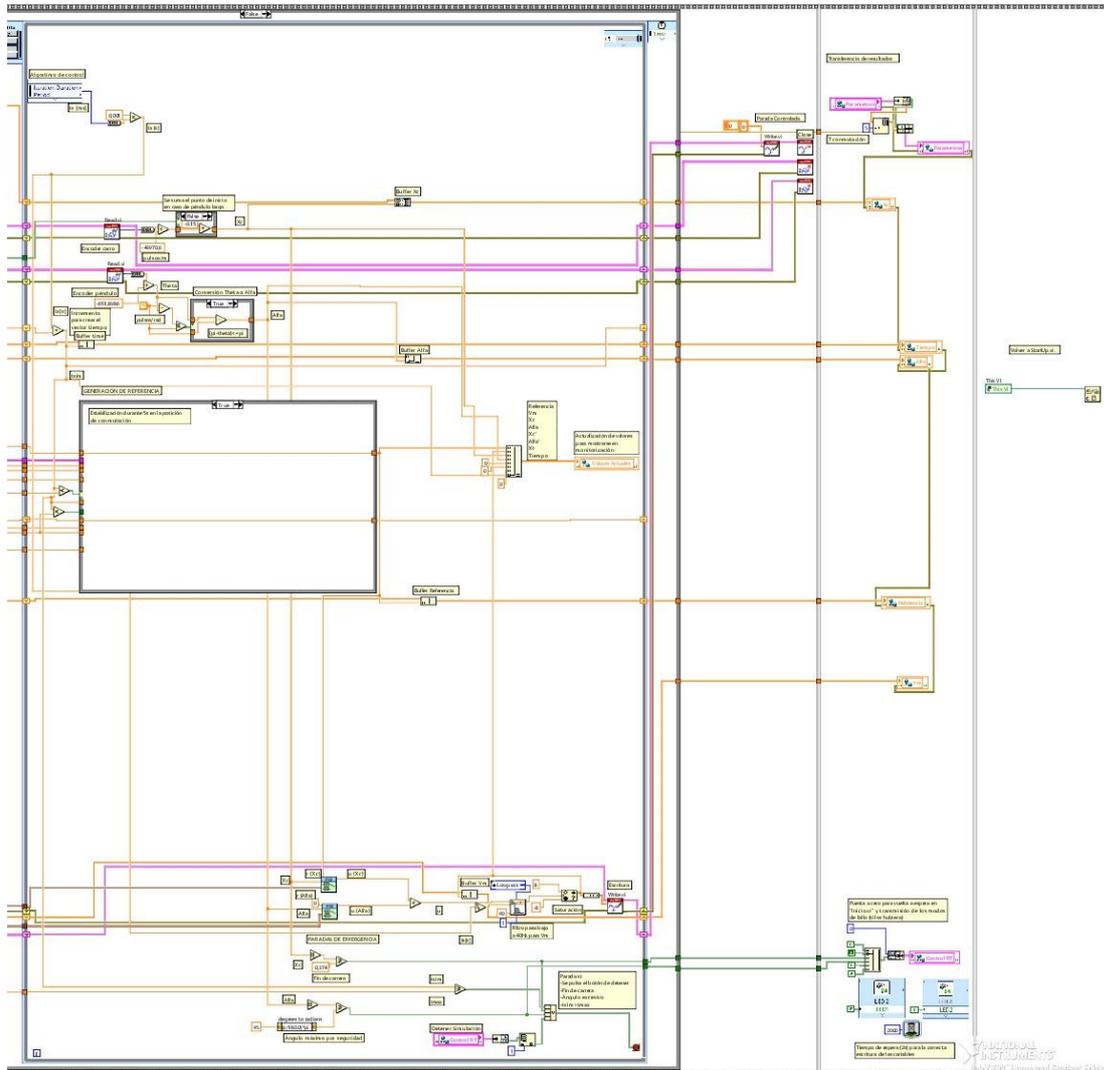
Home/Student Edition  
 Pendulo Invertido PID RT.vi  
 G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID RT.vi  
 Last modified on 24/09/2018 at 19:54  
 Printed on 24/09/2018 at 20:05



Home/Student Edition  
Pendulo Invertido PID RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:05



Home/Student Edition  
Pendulo Invertido PID RT.vi  
G:\Uni\FM\LabView\Aplicacion\Pendulo Invertido PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:05



Home/Student Edition

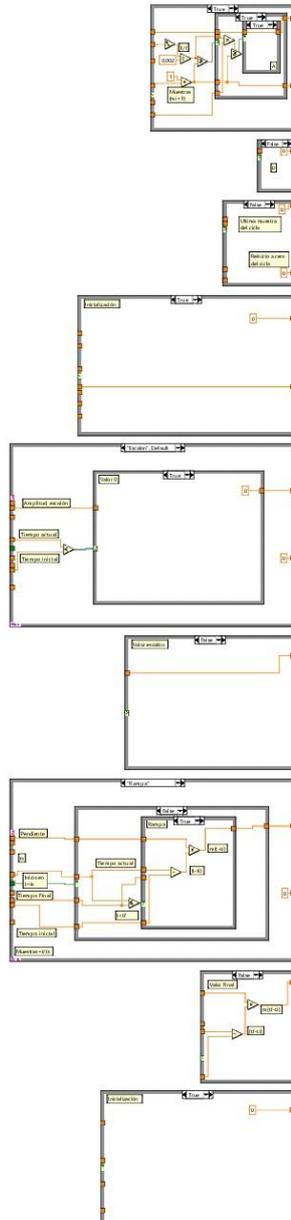
Pendulo Invertido PID RT.vi

G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido PID RT.vi

Last modified on 24/09/2018 at 19:54

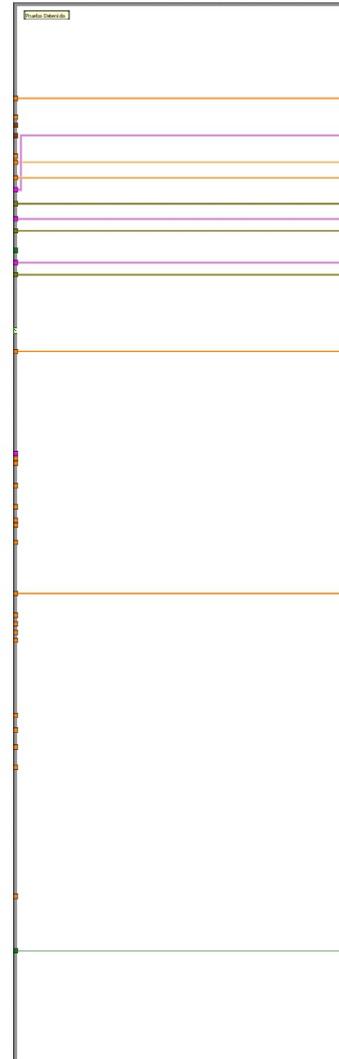
Printed on 24/09/2018 at 20:05

Page 4



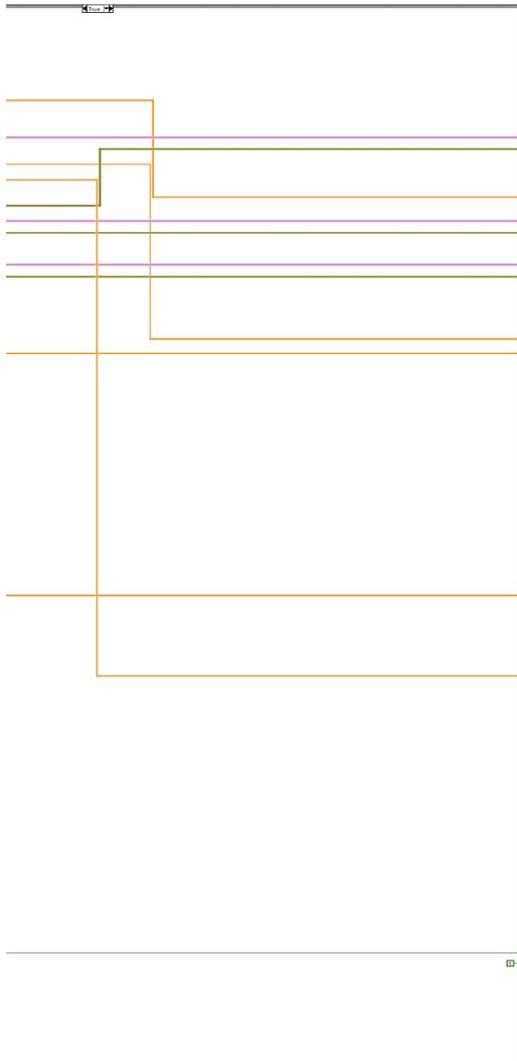
Home/Student Edition  
Pendulo Invertido PID RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:05

Page 5



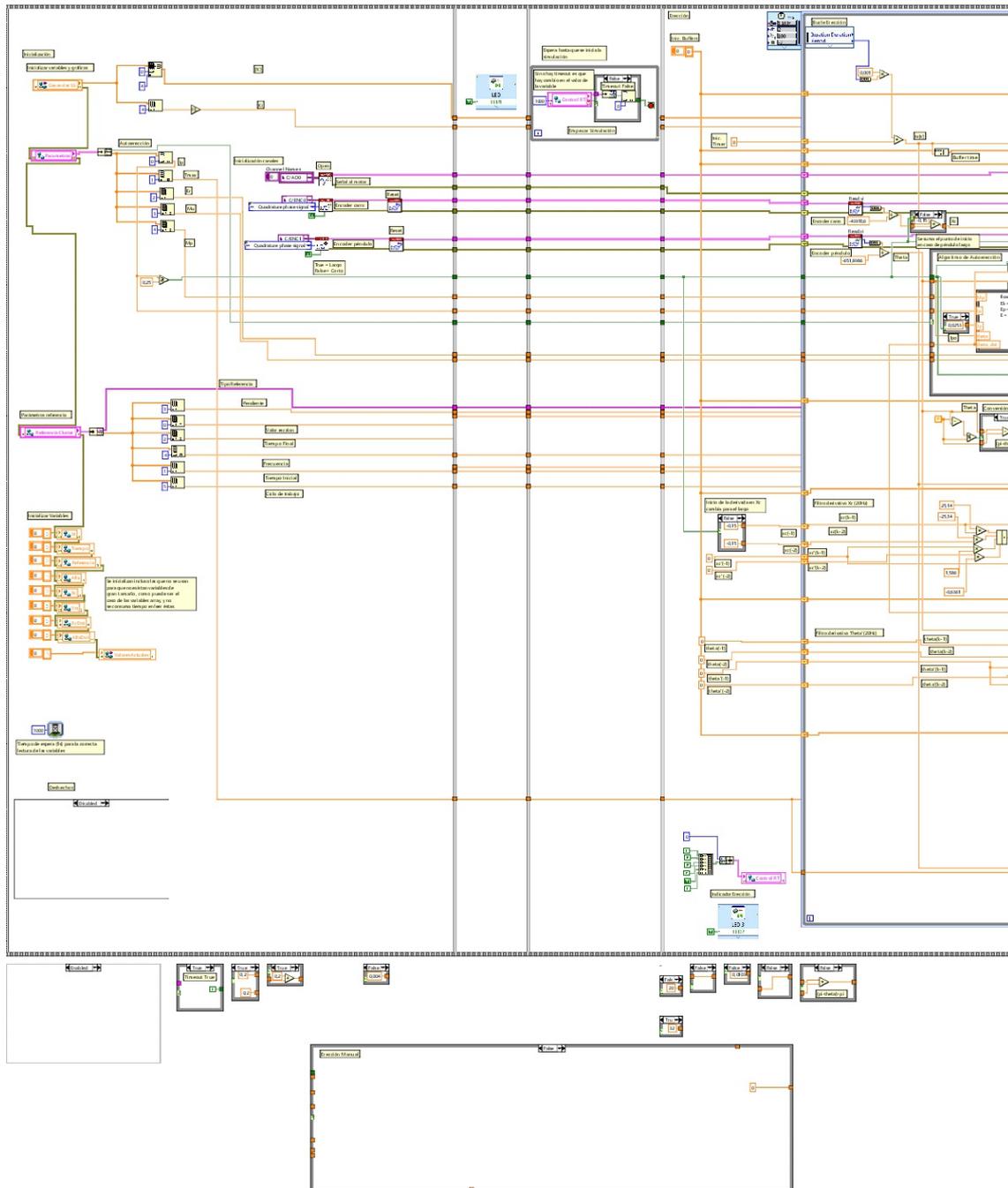
Home/Student Edition  
Pendulo Invertido PID RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido PID RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:05

Page 6

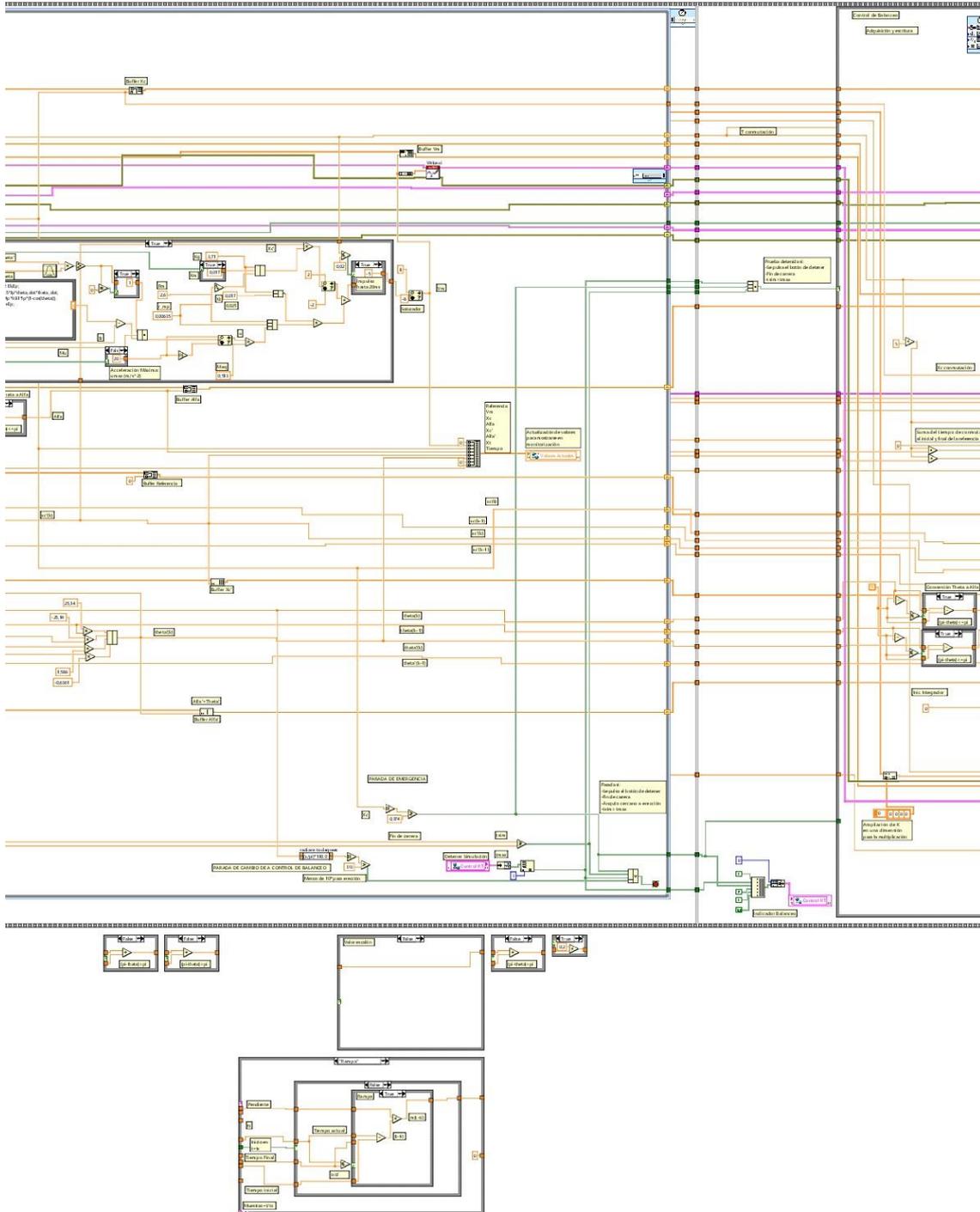


## 23. Pendulo Invertido SS RT.vi

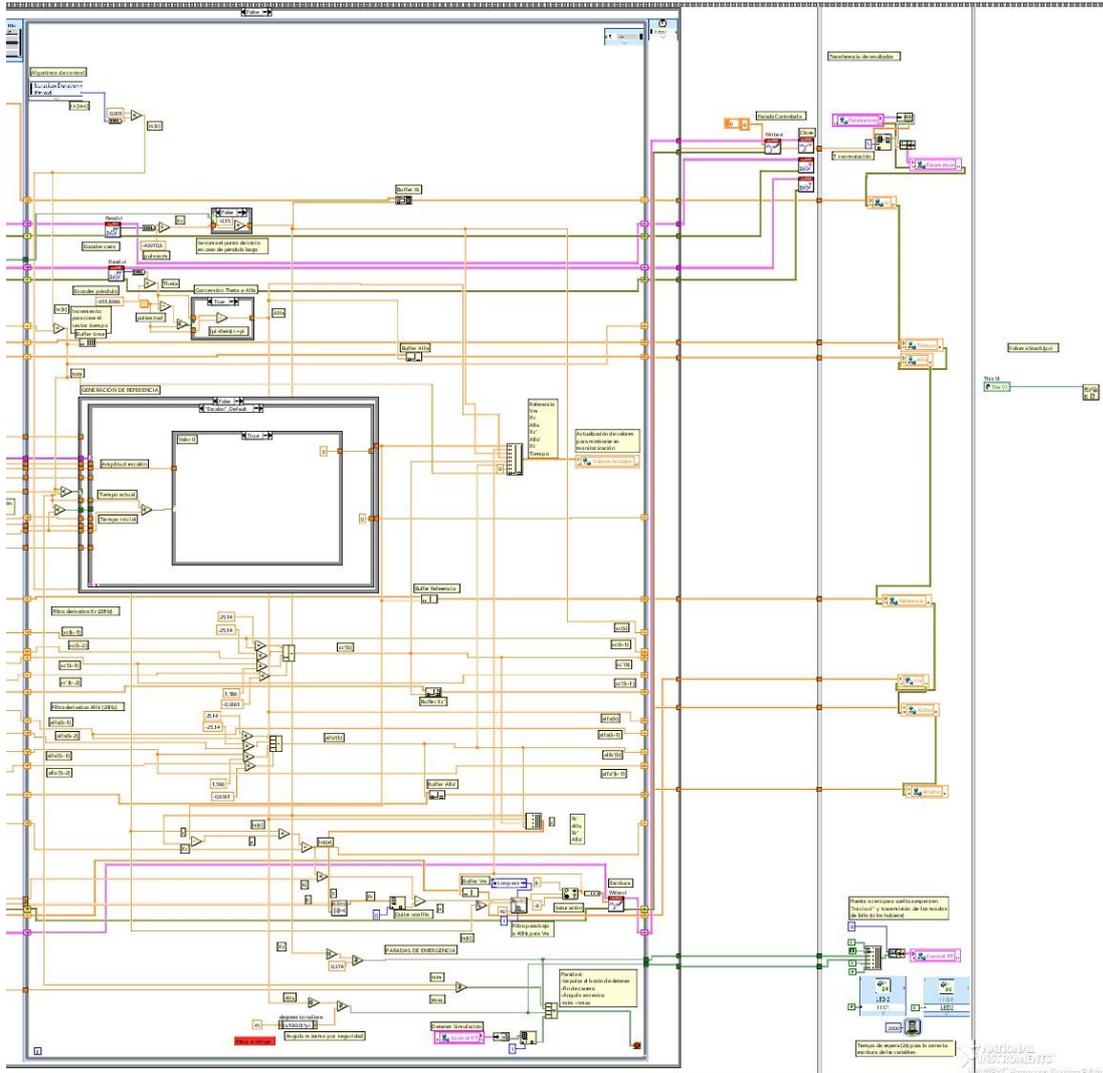
Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09



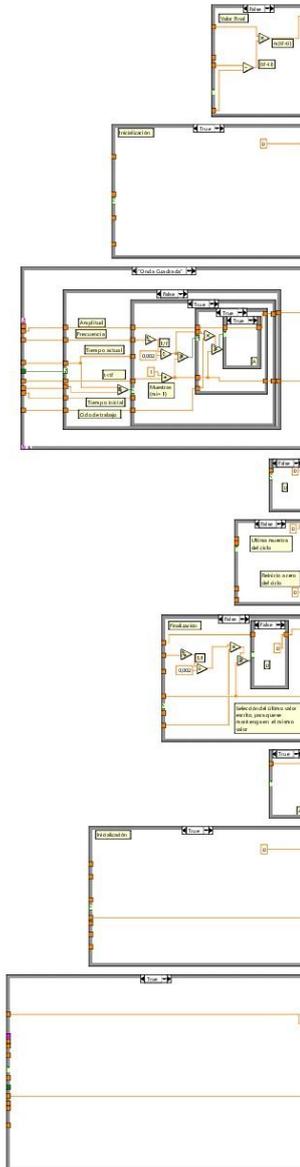
Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09



Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\FM\LabView\Aplicacion\Pendulo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09

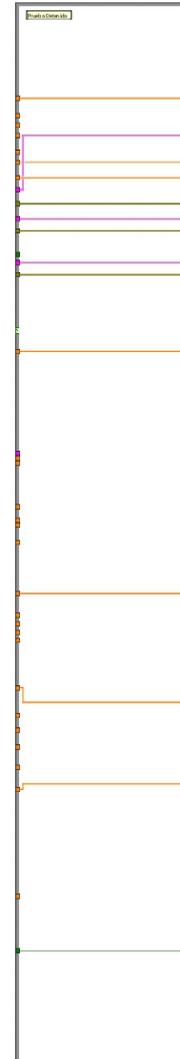


Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendulo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09



Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09

Page 5



Home/Student Edition  
Pendulo Invertido SS RT.vi  
G:\Uni\TFM\LabView\Aplicacion\Pendolo Invertido SS RT.vi  
Last modified on 24/09/2018 at 19:54  
Printed on 24/09/2018 at 20:09

Page 6

