UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE GUAYAQUIL

CARRERA: INGENIERÍA ELECTRÓNICA

TRABAJO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE:
INGENIERO ELECTRÓNICO

TEMA:

DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA PRÁCTICAS DE CONTROLADORES PID CONVENCIONALES

AUTORES:
ALEXANDRA GABRIELA VALENZUELA COBOS
FÉLIX GERARDO ESCANDÓN

TUTOR:
ING MÓNICA MIRANDA MSc

GUAYAQUIL, MARZO DEL 2019

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA

Los conceptos desarrollados en el presente proyecto de titulación, el desarrollo del tema, el análisis y respectivas conclusiones de este, corresponden exclusivamente a **Alexandra Gabriela Valenzuela Cobos** con cedula de identificación 0951832013 y **Félix Gerardo Escandón** con cedula de identificación 0959469370 y el patrimonio intelectual del mismo a la **Universidad Politécnica Salesiana**

Guayaquil, marzo del 2019

(f) Alexandra Valenzuela C.I.0951832013	(f) Félix Escandón C.I. 095946937

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN

Nosotros **Alexandra Gabriela Valenzuela Cobos** con cédula de identificación 0951832013 y **Félix Gerardo Escandón** con cédula de identificación 0959469370, manifestamos a nuestra voluntad y cedemos a la **Universidad Politécnica Salesiana** la titularidad sobre los derechos del actual proyecto de titulación en virtud de que somos los autores del tema: **"DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA PRÁCTICAS DE CONTROLADORES PID CONVENCIONALES"**, mismo que ha sido realizado para la obtención del título de**: INGENIERO ELECTRÓNICO,** quedando la universidad facultada para ejercer y usar los derechos cedidos anteriormente

En aplicación a lo determinado en la Ley de Propiedad Intelectual Del Ecuador, en nuestra condición de autores nos reservamos los derechos morales de la obra anteriormente citada.

En concordancia, suscribimos este documento en el momento que hagamos entrega del trabajo final en formato impreso y digital a la Biblioteca de la Universidad Politécnica Salesiana.

Guayaquil, marzo del 2019

(f) Alexandra Valenzuela C.I.0951832013	(f) Félix Escandón C.I. 095946937

CERTIFICADO DE DIRECCION DEL TRABAJO DE TITULACIÓN

Yo, Msc. Mónica Miranda certifico que bajo mi dirección y asesoría fue desarrollado el proyecto técnico de titulación **DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA PRÁCTICAS DE CONTROLADORES PID CONVENCIONALES**, realizado por la Srta. **Alexandra Gabriela Valenzuela Cobos** y Sr. **Félix Gerardo Escandón** para la obtención del título de**: INGENIERO ELECTRÓNICO**

Por medio de la presente certifico que el documento cumple con los requisitos establecidos en el Instructivo para el Estructura y Desarrollo de Trabajos de Titulación para pregrado de la Universidad Politécnica Salesiana.

En virtud de lo anterior, autoriza su presentación y aceptación como una obra autentica y de valor académico

Guayaquil, marzo 2019

_____

(f) Mónica Miranda

DEDICATORIA

Dedico este proyecto de grado a todos mis seres queridos, a mi padre a mi madre a mis hermanas y a mi novia, por compartir conmigo la aventura de vivir y seguir hasta el final una carrera.

A mis extraordinarios amigos, las vivencias en el aula no hubieran sido las mismas sin ellos allí.

A mi compañera de tesis, me hubiera vuelto loco de no ser por su habilidad para mantener la calma cuando todo va en contra y decir, “tu tranquilo yo me encargo”.

De manera muy personal se la dedico también a quienes creyeron en mí, como esperanza al futuro, gritando "TODO ES POSIBLE, solo es cuestión de tiempo"

Félix Escandón

DEDICATORIA

Dedicado a Dios, por ser el inspirador para continuar en este proceso de obtener uno de los anhelos más anhelados.

A mi familia, principalmente a mis padres: Milton y Cinthia; quienes han sido un pilar fundamental en mi formación como profesional, por ser mi ejemplo quienes me enseñaron que con trabajo y perseverancia se encuentra el éxito además de brindarme la confianza, consejos y recursos para lograrlo.

A mis hermanos: Diego, Milton y Barbie por su cariño y apoyo incondicional, este logro también es de ustedes.

A abuela María que con sus oraciones, consejos y palabras de aliento hicieron de mí una mejor persona y que de una u otra forma me acompañan en todos mis sueños y metas.

Y por último a esos verdaderos amigos; Jean Carlos, Félix, Emilio y Jessica con quienes compartí todos estos años juntos, quienes se convirtieron en amigos de vida y aquellos que serán mis colegas.

Gabriela Valenzuela

AGRADECIMIENTO

Agradezco a mis padres por todo su apoyo y oportunidades que me ayudaron a tomar, así como su guía y consejos muchas veces muy acertados.

A todos mis profesores por los conocimientos que me brindaron, sus experiencias y los retos que pusieron ante mí para superarme una y otra vez a lo largo de mis estudios.

A mi tutora quien estuvo allí para cuando se necesitaba de una ayuda en el momento menos esperado.

A mis amigos quienes creyeron en mi en todo momento y me enseñaron lo que significa tener amigos de verdad, amistades que valen oro.

Félix Escandón

AGRADECIMIENTO

A mi familia, por haberme dado la oportunidad de formarme en esta prestigiosa universidad y haber sido mi apoyo durante todo este tiempo.

A mi tutora, por haberme guiado, no solo en la elaboración de este trabajo de titulación, sino a lo largo de mi carrera universitaria y haberme brindado el apoyo para desarrollarme profesionalmente y seguir cultivando mis valores.

A mis amigos; de manera especial a mi compañero en este proyecto Félix, con quien compartí dentro y fuera del aula gracias por todo tu apoyo, paciencia y diversión.

Gabriela Valenzuela

RESUMEN

AÑO	ALUMNOS	DIRECTOR DEL PROYECTO	TEMA DEL PROYECTO
2019	• FELIX GERARDO ESCANDÓN • ALEXANDRA GARBIELA VALENZUELA COBOS	ING. MÓNICA MIRANDA MSc.	DISEÑO E IMPLEMENTACION DE MÓDULOS PARA PRÁCTICAS DE CONTROLADORES PID CONVENCIONALES

Este proyecto técnico de titulación, “DISEÑO E IMPLEMENTACIÓN DE MÓDULOS PARA PRÁCTICAS DE CONTROLADORES PID CONVENCIONALES” tiene como objetivo realizar un conjunto de programas y plantas que pudieran servir como un acercamiento más amigable a la implementación de los controles PID en sistemas reales.

A lo largo de este documento se irán mostrando: los diseños, programas, manuales y prácticas realizadas para ayudar al estudiante a desarrollar sus potenciales en el manejo de los controladores PID, así como también el poder experimentar con algunas de las herramientas que existen para realizar el control de una planta desde un computador y los diferentes aspectos que intervienen en el diseño de una planta, las características de ella y qué tipo de opciones se tienen a la hora de implementarla.

ABSTRACT

YEAR	STUDENTS	PROJECT DIRECTOR	PROJECT THEME
2019	• FELIX GERARDO ESCANDÓN • ALEXANDRA GARBIELA VALENZUELA COBOS	ING. MÓNICA MIRANDA MSc.	DESIGN AND IMPLEMENTATION OF MODULES FOR CONVENTIONAL PID CONTROLLER PRACTICES

This technical project of university degree. “DESIGN AND IMPLEMENTATION OF MODULES FOR CONVENTIONAL PID CONTROLLER PRACTICES” has as purpose of making a set of programs and plants that could serve as a more friendly approach to the implementation of PID controls in real life systems.

Throughout this document will be shown: the designs, programs, manuals and practices made to help the student to develop their potential in the handling of PID controllers, as well as to be able to experiment with some of the tools that exist to perform the control of a plant from a computer and the different aspects involved in the design of a plant, the characteristics of it and what kind of options are available when implementing it.

ÍNDICE GENERAL

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA II

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA DEL TRABAJO DE TITULACIÓN III

CERTIFICADO DE DIRECCION DEL TRABAJO DE TITULACIÓN..... IV

DEDICATORIA V

AGRADECIMIENTO VII

RESUMEN..... IX

ABSTRACT X

ÍNDICE DE FIGURAS.....XIV

INTRODUCCIÓN..... 1

1.EL PROBLEMA 2

1.1. PLANTEAMIENTO DEL PROBLEMA 2
1.2 IMPORTANCIA Y ALCANCE DEL PROYECTO..... 2
1.3 DELIMITACIÓN 2
1.4 OBJETIVO GENERAL..... 3
1.5 OBJETIVOS ESPECÍFICOS..... 3

2. MARCO TEÓRICO 4

2.1 EL PRINCIPIO DE REALIMENTACIÓN:..... 4
2.2 SISTEMA DE CONTROL CONTINUO Y DISCONTINUO..... 5
2.2.1 Sistema de control continuo..... 5
2.2.2 Sistema de control discontinuo (ON-OFF) 6
2.3 Sistema de control lineales y Sistemas de control no lineales..... 6
2.4 TIPOS DE SISTEMAS DE CONTROL 7
2.4.1 Siso (Single Input Output)..... 7
2.4.2 Miso (Múltiple Input Single Output) 7
2.4.3 Simo (Single Input Multiple Output)..... 7
2.4.4 Mimo (Multiple Input Multiple Output)..... 8
2.5 TIPOS DE CONTROLADORES..... 8
2.5.1 Control Proporcional (P):..... 8
2.5.2 Control Integral (I):..... 9
2.5.3 Control Proporcional-Integral (PI):..... 9
2.5.4 Control Derivativo: 10
2.5.5 Control Proporcional Derivativo:.....10
2.5.6 Control PID, Proporcional, Integral y Derivativo: 11
2.6 SINTONIZACIÓN PID:.....11
2.7 PROCESOS DE IDENTIFICACIÓN DE SISTEMAS:.....11

3. MARCO METODOLOGÍCO 12

3.1 METODOLOGÍA DE LA INVESTIGACIÓN 12
3.2 TÉCNICA DE INVESTIGACIÓN 12
3.3 PROCEDIMIENTO DEL PROYECTO 12
3.3.1 Primera etapa (Investigación del proyecto) 12
3.3.2 Segunda etapa (Equipos y diseño) 13
3.3.3 Tercera etapa (Construcción de módulos y creación de prácticas) 13
3.3.4 Cuarta etapa (Finalización de las instalaciones, pruebas finales) 13
3.4 IMPLEMENTACIÓN DEL PROYECTO 14

3.4.1 Diseño del proyecto 14
3.4.2 Creación de la estructura 15
3.4.3 Diseño de los ambientes virtuales 18
3.4.4 Diseño de las prácticas 18
3.5 PROCESO DE VALIDACIÓN 19
3.5.1 Pasos para la validación 19
3.5.1.1 Elaboración de validación para circuito RC en Matlab 19
3.1.2 Elaboración de validación para circuito RC en LabVIEW 21
3.6 VALIDACIÓN MOTOR DE PASO 22
3.7 PLANTA TÉRMICA 1 26
3.8 PLANTA TÉRMICA 2 28
3.9 PLANTA LEVITADORA 1 29
3.9 PLANTA LEVITADORA 2 33

4. PRÁCTICAS DE LABORATORIO 35

4.1 PRÁCTICA #1 "INTRODUCCIÓN" 35
4.1.1 Objetivos 35
4.1.2 Toolkit LabVIEW-Arduino 35
4.1.3 Toolkit LINX 36
4.1.4 Programar el Arduino 37
4.1.5 Herramientas del Toolkit LabVIEW-Arduino 38
4.1.6 Estructura de un programa 42
4.1.7 Herramienta del Toolkit LINX 43
4.1.8 Estructura de un programa 45
4.2 PRÁCTICA #2 "MODELADO DE SISTEMAS" 46
4.2.1 Objetivos 47
4.2.2 VI de Modelado 47
4.2.3 Secuencia del proceso 50
4.2.4 Para Practicar 51
4.2.5 Modelado Planta térmica 2 57
4.2.6 Modelado Planta Levitadora 1 58
4.2.7 Modelado Motor de Paso 59
4.3 PRÁCTICA #3 "VALORES PID & SISOTOOL" 60
4.3.1 Objetivos 60
4.3.2 Sisotool 61
4.3.2.1 Ecuaciones PID 66
4.3.3 VI Valores de PID 67
4.3.4 PID Planta Motor de Paso 68
4.4 PRÁCTICA #4 "CONTROLADOR PID EN TIEMPO CONTINUO" 71
4.4.1 Objetivos 71
4.4.2 Teoría PID 71
4.4.3 Para Practicar - Parte 1 76
4.4.4 Para Practicar - Parte 2 82
4.4.5 Control PID de la planta Motor de Paso 87
4.5 PRÁCTICA #5 "PID DISCRETO" 90
4.5.1 Objetivos 90
4.5.2 PID discreto 90
4.5.3 VI de PID Z 91
4.5.4 Para Practicar 92
4.5.5 PID Discreto de Planta Térmica 100
4.6 PRÁCTICA#6 "LEVITADOR EÓLICO" 102
4.6.1 Objetivos 102
4.6.2 VI modelado LE 102
4.6.3 Programación interna del VI de modelado LE 104

4.6.4 Toma de muestras de planta Levitador Eólico. 106
4.6.5 Realización de modelados varios.....109
4.6.6 Linealización y simulación de comportamiento 111
4.6.7 VI Práctica PID de planta LE.....112
4.6.8 Programación interna del VI Práctica PID de la planta Levitadora 113
4.6.9 Realización de controles PID 115
4.6.10 Control P.....116
4.6.11 Control PI.....116
4.6.12 Control PID 116

5. RESULTADOS 121

5.1 MOTOR DE PASO.....121
5.2 PLANTAS TÉRMICAS 1 Y 2 122
5.3 PLANTAS LEVITADORAS 1 Y 2 122

6. CONCLUSIONES Y RECOMENDACIONES 124

6.1 CONCLUSIONES 124
6.2 RECOMENDACIONES 124

7. BIBLIOGRAFÍA.....125

8. ANEXOS.....126

ÍNDICE DE FIGURAS

Figura 1 Diagrama de bloques del sistema de control de un proceso 4
Figura 2 Sistemas continuos y discontinuos 5
Figura 3 Esquema de un calentador de agua a vapor..... 5
Figura 4 Esquema de calentador de agua eléctrico 6
Figura 5 Sistema Siso 7
Figura 6 Sistema Miso 7
Figura 7 Sistema Simo 7
Figura 8 Sistema Mimo 8
Figura 9 Funciones más usadas para identificación de sistemas 11
Figura 10 Árbol de causas y efectos.....12
Figura 11 Diseño del puente.....14
Figura 12 Diseño de la pista del puente.....14
Figura 13 Vista lateral del gabinete plástico.....15
Figura 14 Vista superior de la Planta Térmica 15
Figura 15 Vista isométrica de la Planta Térmica.....16
Figura 16 Vista superior de la Planta Motor de Paso 16
Figura 17 Vista isométrica de la Planta Motor de Paso.....17
Figura 18 Vista isométrica la Planta de Levitación.....17
Figura 19 Archivos del proyecto.....18
Figura 20 Ventana de la librería a usar 20
Figura 21 Circuito Iniciador 20
Figura 22 Diseño del circuito en Simulink para validación de circuito RC 20
Figura 23 Señal de entrada 21
Figura 24 Comparación del comportamiento Real vs Simulado 21
Figura 25 Error medio cuadrático.....21
Figura 26 VI validación circuito RC 21
Figura 27 Recolección de datos.....22
Figura 28 Error medio cuadrático.....22
Figura 29 VI de control manual motor de paso 22
Figura 30 Ventana Ident 23
Figura 31 Señal de salida 23
Figura 32 Señal de entrada 23
Figura 33 Diferentes modelos con su porcentaje de aproximación 24
Figura 34 Función de transferencia utilizada “nlhw1” 24
Figura 35 Exportación de datos 24
Figura 36 Error medio cuadrático.....25
Figura 37 Validación con modelo de orden 2 25
Figura 38 Error medio cuadrático.....25
Figura 39 Ventana Ident 26
Figura 40 Señal de salida 26
Figura 41 Señal de entrada 26
Figura 42 Función de transferencia “tf7” 27
Figura 43 Conjunto de muestras 1, con su proximidad 27
Figura 44 Conjunto de muestras 2, con su proximidad 27
Figura 45 Validación de la planta térmica 1 27
Figura 46 Error medio cuadrático.....28
Figura 47 Ventana Ident 28
Figura 48 Señal de salida 28
Figura 49 Señal de entrada 28
Figura 50 Conjunto de muestras, con su proximidad 29
Figura 51 Función de transferencia “tf10” 29
Figura 52 Validación Planta térmica.....29

Figura 53 Error medio cuadrático....29
Figura 54 Respuesta de la simulación....30
Figura 55 Señal de salida....30
Figura 56 Señal de entrada....30
Figura 57 Ventana Ident....31
Figura 58 Primer conjunto de muestras con su proximidad....31
Figura 59 Segundo conjunto de muestras con su proximidad....31
Figura 60 Función de transferencia 13....32
Figura 61 Función de transferencia 12....32
Figura 62 Validación para el caso 1....32
Figura 63 Error medio cuadrático caso 1....32
Figura 64 Validación para el caso 2....33
Figura 65 Respuesta de la simulación....33
Figura 66 Señal de salida....33
Figura 67 Señal de entrada....34
Figura 68 Conjunto de muestras con su proximidad....34
Figura 69 Función de transferencia tf4....34
Figura 70 Respuesta de la simulación....34
Figura 71 Error medio cuadrático....34
Figura 72 Toolkit para Arduino....35
Figura 73 Proceso de descarga LINX....36
Figura 74 Proceso de instalación de LINX....36
Figura 75 Toolkits a utilizar....37
Figura 76 Carpeta librería LabVIEW/Arduino....37
Figura 77 Archivos dentro de la carpeta LIFA_Base....38
Figura 78 Herramientas de Toolkit Arduino....38
Figura 79 Ejemplos del Toolkit Arduino....39
Figura 80 Elementos dentro de Low Level....39
Figura 81 Elementos dentro de Bus Specific....40
Figura 82 Elementos dentro de la pestaña Sensors....40
Figura 83 Elementos dentro la pestaña BlinkM....41
Figura 84 Elementos dentro de la pestaña Servo....41
Figura 85 Elementos dentro de la pestaña Stepper Motor....41
Figura 86 Elementos dentro de la pestaña LCD....42
Figura 87 Elementos dentro de la pestaña Utility....42
Figura 88 Pasos para acceder a Toolkit LINX....43
Figura 89 Pasos para acceder a la herramienta Analog....43
Figura 90 Opciones dentro de la herramienta Analog....44
Figura 91 Opciones dentro de la herramienta Digital....44
Figura 92 Opciones dentro de la herramienta PWM....44
Figura 93 Herramientas dentro de las casillas I2C SPI UART....45
Figura 94 Ubicación del sensor Ultrasónico....45
Figura 95 Herramientas dentro de la pestaña Utilies....45
Figura 96 Front Panel del VI Modelado....47
Figura 97 Visualización de valores de entrada y salida del VI....48
Figura 98 Controles del VI....48
Figura 99 Selección de pines a usa....49
Figura 100 Control de Vin y Vout a modo simulado....49
Figura 101 Controladores e Indicadores a usar en la parte simulada....49
Figura 102 Esquema del circuito....51
Figura 103 Paso 1 de la práctica....51
Figura 104 Paso 2 de la práctica....52
Figura 105 Paso 3 de la práctica....52
Figura 106 Herramienta de visualización....52

Figura 107 Visualización mejorada de la práctica 53
Figura 108 Comportamiento de un circuito RC 53
Figura 109 Identificación de la planta.....54
Figura 110 Colocación del nombre de Entrada y Salida.....54
Figura 111 Colocación del intervalo del tiempo de muestreo 55
Figura 112 Obtención de las muestras 55
Figura 113 Obtención de un estimado de la función de transferencia.....56
Figura 114 Colocación de numero de polos y ceros de la función de transferencia 56
Figura 115 Obtención de la función de transferencia.....57
Figura 116 Muestreo de la Planta Térmica 2 57
Figura 117 Ventana Ident 58
Figura 118 Conjunto de muestras con su proximidad 58
Figura 119 Función de transferencia "tf10" 58
Figura 120 Muestreo Planta Levitadora 1 58
Figura 121 Ventana Ident 59
Figura 122 Conjunto de muestras con su proximidad 59
Figura 123 Función de transferencia "tf12" 59
Figura 124 Muestreo Planta Motor de Paso.....59
Figura 125 Ventana Ident 60
Figura 126 Conjunto de muestras con sus proximidades.....60
Figura 127 Función de Transferencia "tf4".....60
Figura 128 Ventana del comando Sisotool 61
Figura 129 Ventana Architecture 61
Figura 130 Ventana System Data.....62
Figura 131 Ventana Sample Time Conversion.....62
Figura 132 Ventana Compensator Editor.....63
Figura 133 Ventana Graphical Tunning 63
Figura 134 Ventana Analysis Plots 63
Figura 135 Ventana Automated Tunning 64
Figura 136 Opciones de métodos de diseño.....64
Figura 137 Especificaciones de los tipos de controlador.....64
Figura 138 Función de transferencia.....65
Figura 139 Exportación de datos 65
Figura 140 Valores que se exportara.....66
Figura 141 Front Panel de la práctica PID 67
Figura 142 Block Diagram de la práctica PID.....68
Figura 143 Colocación de datos 68
Figura 144 Selección del grafico a utilizar.....68
Figura 145 Elección del tipo de grafico 69
Figura 146 Agregar polos y zeros.....69
Figura 147 Gráficos del lugar geométrico de las raíces, y su respuesta al paso 69
Figura 148 Datos del control PID 70
Figura 149 Exportación de datos 70
Figura 150 Valores de control PID 70
Figura 151 Front Panel Vi PID en continuo.....73
Figura 152 Visualizador de la Función de transferencia y controlador 73
Figura 153 Visualizador de los valores en tiempo real 73
Figura 154 Controles de las Ganancias y Set Point 74
Figura 155 Descripción de colores de las señales 75
Figura 156 Botones que controlan el VI 75
Figura 157 Seleccionadores de pines.....75
Figura 158 Aviso del VI.....76
Figura 159 Control PID con Saturación.....76
Figura 160 Control PID sin Saturación.....77

Figura 161 Aumento del tiempo de integración sin saturación 77
Figura 162 Aumento del tiempo de integración con saturación 78
Figura 163 Segundo aumento de integración 78
Figura 164 Desactivado el control PID.....79
Figura 165 Aumento de la ganancia proporcional a 2 79
Figura 166 Aumento de la ganancia proporcional a 4 79
Figura 167 Aumento de la ganancia proporcional a 10 80
Figura 168 Control PID con tiempo de 1 80
Figura 169 Control PID con un tiempo integral de 0.5.....81
Figura 170 Control PID con un valor bajo de ganancia Kp.....81
Figura 171 Control PID cambiando el Set Point.....82
Figura 172 Ejemplos de valores PID.....82
Figura 173 Esquema del circuito.....83
Figura 174 Función de transferencia en Ident.....84
Figura 175 Respuesta con la función Step.....85
Figura 176Simulador LabVIEW vs Sisotool.....85
Figura 177 Simulación con Gear's Method 86
Figura 178 Simulación con Runge-Kutta 4.....86
Figura 179 Control PID para un circuito RLC 87
Figura 180 Modelado de la Planta Motor de Paso 88
Figura 181 Función de transferencia a usar.....88
Figura 182 Obtención de los valores PID.....88
Figura 183 Control PID de la Planta Motor de Paso.....89
Figura 184 Grafica del lugar geométrico de las raíces y de su respuesta al paso...89
Figura 185 Front panel del VI para el control PID en Z 91
Figura 186 Block Diagram de Control 91
Figura 187Block Diagram Control PID en z parte 1.....92
Figura 188 Block Diagram Control PID en z parte 2.....92
Figura 189 Diseño del circuito.....92
Figura 190 Función de transferencia.....93
Figura 191 Señales de entrada, salida y aproximación. 93
Figura 192 Grafica de Polos y Ceros 94
Figura 193 Cambio a tiempo discreto 94
Figura 194 Grafica del control PID en Z.....94
Figura 195 Características de la planta.....95
Figura 196 Cambiando valor de ganancias.....95
Figura 197 Respuesta del controlador 95
Figura 198 Grafica con un integrador puro una ganancia unitaria.....96
Figura 199 Comparación de graficas 96
Figura 200 Control PI.....97
Figura 201 PI del sistema vs Respuesta de la planta.....97
Figura 202 Control proporcional unitario 98
Figura 203 Cambio de valores de las ganancias.....98
Figura 204 Señal de control P.....98
Figura 205 Señal ganancia unitaria 99
Figura 206 Valor del compensador en sisotool 99
Figura 207 Control PI en sisotool.....99
Figura 208 Simulación LabVIEW vs Sisotool. 100
Figura 209 Planta R.....100
Figura 210 Ganancias del control PID 101
Figura 211 PID con un tiempo de ejecución de 0.1 s 101
Figura 212 PID con un tiempo de ejecución de 1 s 101
Figura 213 PID con un tiempo de ejecución de 3 s 102
Figura 214 Front Panel VI Levitador Eólico.....103

Figura 215 Block Diagram parte 1 105
Figura 216 Block Diagram parte 2 105
Figura 217 Block Diagram parte 3 106
Figura 218Pasos para la comunicación entra LabVIEW y Arduino 107
Figura 219 Comportamiento de la planta 107
Figura 220 Primera recolección de datos.....108
Figura 221 Segunda recolección de datos.....108
Figura 222 Comand Windows.....109
Figura 223 Filtrado de señales.....109
Figura 224 Comparación de resultados 110
Figura 225 Seleccionar la aproximación más exacta 110
Figura 226 Respuesta con Step.....111
Figura 227 Diferentes modelos 112
Figura 228 Front Panel VI para control PID Levitador Eólico 113
Figura 229 Block Diagram parte 1 114
Figura 230 Block Diagram parte 2 115
Figura 231 Control P.....116
Figura 232 Control PI.....116
Figura 233 Control PID 116
Figura 234 Señal de Salida.....117
Figura 235 Señal de Entrada 117
Figura 236 Ventana Ident 117
Figura 237 Resultado de modelo no lineal 117
Figura 238 Análisis de proceso de mejora de la señal 118
Figura 239 Resultados de modelos no lineales.....118
Figura 240 Resultados de funciones transferencias.....119
Figura 241 Comparación de simulaciones lineales 119
Figura 242 Simulacion en Sisotool, con la respuesta de PI.....119
Figura 243 PID utilizado.....120
Figura 244 Valores de las ganancias del PID 120
Figura 245 Planta con el controlador PID 120
Figura 246 Resultado Planta Motor de Paso.....121
Figura 247 Resultado Planta Motor Térmica 1 y 2 122
Figura 248 Resultado de la Planta Levitadora 1 y 2.....123
Figura 249 Módulos de las Plantas Térmicas 126
Figura 250 Módulos de las Plantas Levitadoras.....126
Figura 251 Modulo de la Planta Motor de Paso 127

Introducción

En la práctica de la ingeniería frecuentemente persiste la necesidad de trasladar un sistema físico hasta un punto explícito con un indiscutible valor de precisión. Un sistema de control es el conjunto de métodos y herramientas que ordenan un sistema físico hasta los escenarios esperados.

Dentro de la vida cotidiana son cada vez más los aspectos en que se han visto incluidos los sistemas de control, ya sea en la producción industrial, sistemas de transporte, líneas de ensamble, sistemas de seguridad, etc.

En la mayoría de éstos, la exactitud y certidumbre de los procesos realizados es de transcendental importancia, y en ellas el control juega un papel importante. El fundamento del control automático radica en calcular una operación correctiva en función de la diferencia entre el estado presente del proceso y el estado esperado.

Esto se conoce como control realimentado y está conformado por tres bloques básicos: un proceso, planta o sistema a controlar, un sensor y un controlador.

El controlador es un conjunto de elementos (mecánico, eléctrico, neumático, entre otros) que capta en su entrada el error de referencia y hace una ley de control para provocar la señal correctiva de entrada del proceso.

El presente trabajo propone ayudar en la asimilación del aspecto practico de la enseñanza de los controladores tipo PID realizándose este en tres diferentes tipos de módulos (temperatura, posición con sus respectivas simulaciones), haciendo uso de herramientas LabVIEW y Matlab, así como Arduino ya que son los entornos más conocidos.

1. El PROBLEMA

1.1. Planteamiento del problema

En la Universidad Politécnica Salesiana sede Guayaquil, existe la necesidad de tener una forma didáctica y eficiente para poder realizar la tutoría y aprendizaje de los controladores PID y todo lo que estos acarrean, desde el modelado de plantas hasta el proceso de control.

Al término actual en este documento (2019), en la Universidad Politécnica Salesiana sede Guayaquil, cuenta únicamente con métodos de enseñanzas teórico, simulado del control PID, al igual que el proceso de modelación de plantas.

Dada esta problemática, los estudiantes que cursan la carrera de Ingeniería Electrónica, encuentran difícil la aplicación y uso del control PID junto con el modelado de plantas fuera del ambiente virtual. Debido que la Universidad carece de los módulos físicos en los cuales se puedan realizar las prácticas.

1.2 Importancia y alcance del proyecto

El conectar la teoría con la practica resulta fundamental en el proceso aprendizaje, sobre todo en el caso de los controles PID. En la actualidad los métodos para esto resultan en ocasiones ser poco intuitivos para poder ser manipulados con destrezas por los estudiantes.

Este proyecto brinda medios didácticos para la conexión de la teoría con la práctica de los controladores PID, junto a un conjunto de prácticas auto guiadas que abarcan los procesos de: Toma de muestras, Modelado de sistemas, Diseño de un control PID en tiempo continuo y Diseño de un control PID en tiempo discreto.

Pasando a su vez por los conceptos de PWM así como del manejo de herramientas de Matlab como "Ident" y "Sisotool".

1.3 Delimitación

Destinado a ser utilizado de manera principal en la materia de "Control 1" y en el laboratorio de Control Automático, de la Universidad Politécnica Salesiana, sede Guayaquil.

1.4 Objetivo General

Implementar 5 módulos para prácticas relacionadas a controladores PID convencionales junto con 5 prácticas para el uso de estos.

1.5 Objetivos específicos

- Diseñar la estructura de los módulos
- Seleccionar los dispositivos de control a utilizar, sensores de velocidad y temperatura, junto con su respectivo motor y fuente de calor a utilizar.
- Recopilar la información teórica afín a los controladores PID a proporcionar dentro de las prácticas.
- Desarrollarlos complementos virtuales de las prácticas en los ambientes de LabVIEW y Matlab.
- Construir las tarjetas electrónicas y carcasas de los módulos.
- Realizar los programas a usar en los controladores seleccionados.
- Elaborar manuales de uso de las diversas practicas
- Realizar las prácticas para el uso de los módulos a implementar.

2. MARCO TEÓRICO

El control PID se volvió una herramienta estándar en los procesos de control desde su creación en el año de 1940.

En la actualidad más del 95% de los procesos de control son realizados usando un tipo de control PID, el más usado es el control PI.

El control PID es un componente esencial en los sistemas de control distribuidos, y es normalmente combinado con selectores, funciones lógicas, funciones secuenciales y bloques de funciones simples para poder formar complicados sistemas de automatización usados en una variedad de procesos. [1]

Los controles PID tienen por fama el ser robustos por ello suelen ser usados con frecuencia en la industria, adicional a esto son de estructura sencilla lo cual permite un fácil uso de estos, aunque esto lleva a que no respondan bien a plantas con sistemas no lineales o con variables que fluctúen bruscamente de manera constante. [2]

2.1 El principio de realimentación:

De manera sintetizada se utiliza un proceso de respuesta proporcional inversa, lo cual se refiere a que si tu variable monitoreada (del proceso) baja del punto de referencia, se va a producir como respuesta un incremento en la variable controlada por el proceso, vuelta de esta llegar a superar el punto de referencia, producirá una reducción en la variable controlada.

A este tipo de realimentación se le conoce como realimentación negativa por la relación de crecimiento y reducción inversa de la variable de control, respecto de la variable monitoreada [3]

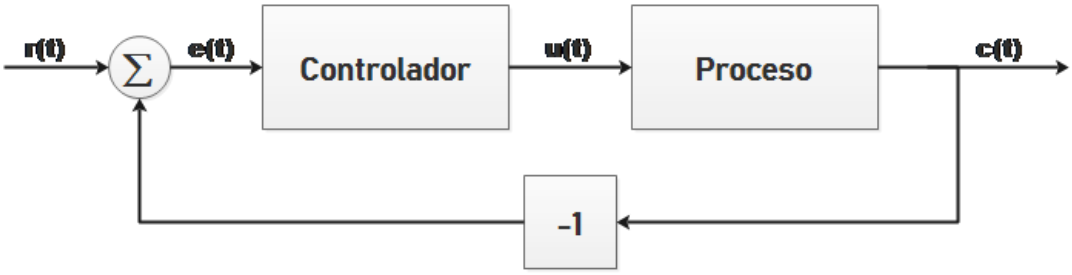


Figura 1 Diagrama de bloques del sistema de control de un proceso [3]

.

|

2.2 Sistema de control continuo y discontinuo

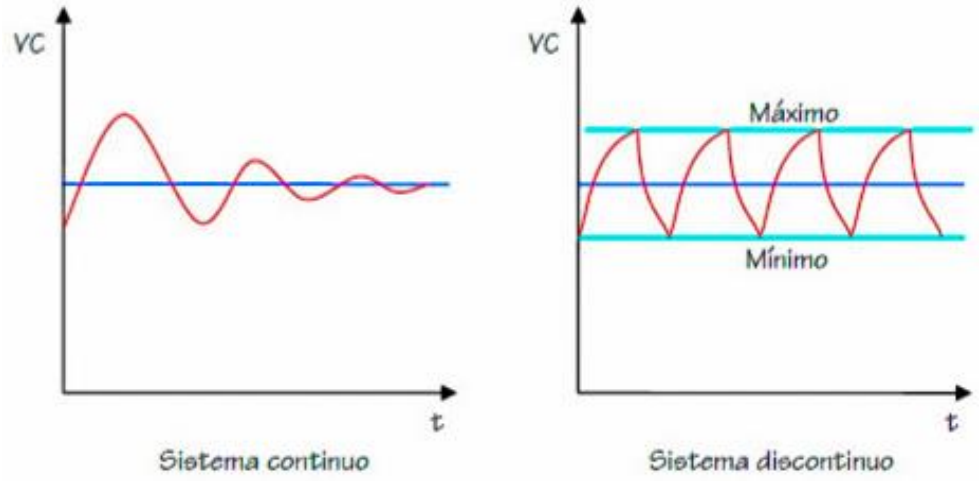


Figura 2 Sistemas continuos y discontinuos [4]

2.2.1 Sistema de control continuo

Es aquel donde la variable que se manipula no efectúa continuamente. Se puede tomar como ejemplo el controlador de la temperatura con vapor.
Este instrumento calienta el agua y con la ayuda del serpentín por el cual pasa el flujo de vapor aumentado o disminuido por una válvula de control, la cual se abre o cierra según como desee el controlador.
La abertura de la válvula puede tomar una infinidad de valores, entre estar abierta y cerrada entonces es continuo. [4]

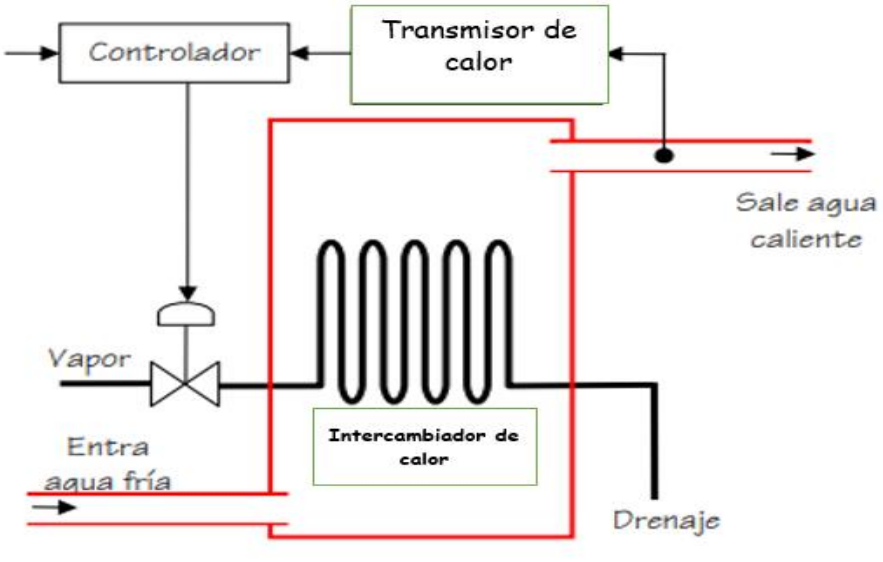


Figura 3 Esquema de un calentador de agua a vapor [4]

2.2.2 Sistema de control discontinuo (ON-OFF)

Es aquel que solo puede tomar un único valor máximo o mínimo. Este solo funciona abierto o cerrado.
Como aplicación se tendría un sistema de calentador de agua mediante una resistencia eléctrica la cual se conecta y se desconecta, según el valor del termostato, el cual en esta aplicación sería un instrumento de medición, controlador y elemento final de control. Ya que cuando llegue a la temperatura indicada se desconecta el circuito eléctrico apagando la resistencia eléctrica. [4]

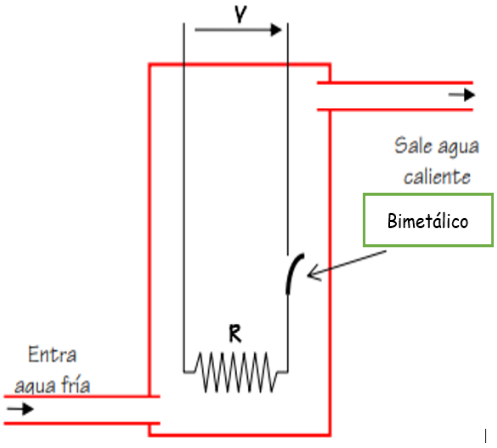


Figura 4 Esquema de calentador de agua eléctrico [4]

2.3 Sistema de control lineales y Sistemas de control no lineales

Se llama sistema lineal aquel que tiene relación la entrada y la salida y se puede definirse con una ecuación lineal.

$$y = ax + b \tag{1}$$

Además, un sistema lineal se puede aplicar el principio de superposición, este establece que la respuesta que se obtiene por dos funciones excitadoras distintas es la suma de las respuestas individuales.

$$\text{Si } s = f(e) \text{ entonces } e = a + b \ \rightarrow s = f(a) + f(b) \tag{2}$$

Cabe decir que los sistemas reales son todos no lineales, sin embargo, en muchas ocasiones la extensión del cambio de variables del sistema no es amplia y se puede linealizar el sistema dentro de un rango relativamente estrecho de valores de variables.

2.4 Tipos de sistemas de control

2.4.1 Siso (Single Input Output)

Costa de una sola entrada y una sola salida, este tipo de sistema es el más eficaz ya que es donde se pierde menos la señal. [5]

Figura 5 Sistema Siso [5]

2.4.2 Miso (Múltiple Input Single Output)

Cuando se consta de un sistema con múltiples señales de entradas o control y una señal de salida [5]

Figura 6 Sistema Miso [5]

2.4.3 Simo (Single Input Multiple Output)

Sistema el cual solo se tiene una señal de entrada y múltiples señales de salida. [5]

Figura 7 Sistema Simo [5]

2.4.4 Mimo (Multiple Input Multiple Output)

Sistema el que consta con múltiples entradas y múltiples salidas. [5]

Figura 8 Sistema Mimo [5]

2.5 Tipos de controladores

2.5.1 Control Proporcional (P):

En este controlador cuenta con las siguientes formulas, variables y constantes:

$$e(t)\ Error$$
$$u(t)\ Salida\ del\ controlador$$
$$K_p\ constante\ de\ ganancia\ proporcional$$

$$u(t) = K_p e(t) \tag{3}$$

$$\frac{U(s)}{E(s)} = K_p \tag{4}$$

Se entiende por control proporcional al acto de controlar la variable monitoreada usando una relación proporcional a la señal de error que es el resultado de restar del punto de referencia el valor de la variable monitoreada, y este asignarlo a la variable controlada o señal de control como se muestra en la formula (3).

El aumento de ganancia en este controlador tiene como efecto una reducción en el error de este estacionario, no obstante, no se puede eliminar del todo el mismo dado que el control va a presentar oscilaciones dentro de los límites tolerables para el error, dichas oscilaciones aumentaran en dimensión conforme se use una ganancia de mayor magnitud, volviendo al sistema inestable en algunos casos. [6]

2.5.2 Control Integral (I):

Una de las características más importantes de este tipo de control es que ayuda a eliminar el error en estado estacionario y al mismo tiempo brinda estabilidad al sistema reduciendo las oscilaciones, esto lo logra usando la integral del valor de error, consiguiendo así un registro del progreso de este a través del tiempo. [6]

Con este controlador se añaden las siguientes ecuaciones:

$$u(t) = K_i \int_0^t e(t)dt \tag{5}$$

$$\frac{U(s)}{E(s)} = \frac{K_i}{s} \tag{6}$$

Adicional a esto es importante notar que el control integral agrega un polo en el origen (la s) dentro de la función de transferencia (4) en lazo abierto, lo cual repercute en la eliminación del error en estado estacionario cuando se ingresa una entrada escalón como se muestra en las siguientes ecuaciones:

$$e_{ss} = \frac{1}{1+K_p} \tag{7}$$

$$K_p = \lim_{s\to 0} G_c G_p = \lim_{s\to 0} \frac{K_i}{s} G_p(s) \to \infty \tag{8}$$

$$\therefore\ e_{ss} \to 0$$

Por otro lado, el control integral también tiende a aumentar el sobre impulso y con ello la estabilidad relativa de la planta, por ello se recomienda utilizar otras acciones de control en conjunto con esta misma. [6]

2.5.3 Control Proporcional-Integral (PI):

En esta se unen los dos métodos de control antes tratados en este documento, combinándolos para lograr combinar las virtudes de ambos consiguiendo tanto estabilidad como reducción del error en estado estacionario, las fórmulas resultantes de este proceso son las siguientes:

$$u(t) = K_p e(t) + K_i \int_0^t e(t)dt \tag{9}$$

$$u(t) = K_p \left[e(t) + \frac{K_i}{K_p} \int_0^t e(t)dt \right] \tag{10}$$

$$u(t) = K_p \left[e(t) + \frac{1}{T_i} \int_0^t e(t)dt \right] \tag{11}$$

$$\frac{U(s)}{E(s)} = K_p \left[1 + \frac{K_i/K_p}{s} \right] \tag{12}$$

$$\frac{U(s)}{E(s)} = K_p \frac{s+(K_i/K_p)}{s} \quad (13)$$

$$\frac{U(s)}{E(s)} = K_p \frac{s+1/T_i}{s} \quad (14)$$

El tiempo integral (T_i) es el tiempo necesario para que la parte integral del control haga igual la parte proporcional al error.
Hay que notar que el control PI agrega:

Un polo en el origen s = 0
Y un cero en s = $-K_i/K_p$.

2.5.4 Control Derivativo:

Esta consiste en utilizar un valor proporcional al obtenido de realizar la derivada del error:

$$u(t) = K_d \frac{de(t)}{dt} \quad (15)$$

Usando el control derivativo se obtiene la cualidad de poder realizar un control de anticipación, ya que esta permite obtener la tendencia de crecimiento o decrecimiento de la señal de error, pero esto también le hace vulnerable al ruido ya que esta actuara de manera agresiva si detecta un cambio brusco en la señal de error. [6]

2.5.5 Control Proporcional Derivativo:

Al combinarse las características del control proporcional con las del derivativo, se tiene como resultado una respuesta rápida con poco sobre impulso y mucha precisión en el sistema, pero de excederse se puede volver muy lenta.
La salida de control adquiere la siguiente forma:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} \quad (16)$$

$$u(t) = K_p \left[e(t) + \frac{K_d}{K_p} \frac{de(t)}{dt} \right] \quad (17)$$

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} \right] \quad (18)$$

T_d es la variable que representa al tiempo derivativo; Este tipo de control genera un cero en
s = $-1/K_d$.

2.5.6 Control PID, Proporcional, Integral y Derivativo:

Este control es la combinación final de todos los anteriores vistos, el balance ideal y maleable entre velocidad, estabilidad y eficiencia en términos de bajos sobre picos, el resultado total de esta mezcla en la variable controlada es la siguiente:

$$u(t) = K_p e(t) + K_d \frac{de(t)}{dt} + K_i \int_0^t e(t)dt \quad (19)$$

$$u(t) = K_p \left[e(t) + T_d \frac{de(t)}{dt} + \frac{1}{T_i} \int_0^t e(t)dt \right] \quad (20)$$

$$\frac{U(s)}{E(s)} = K_p \left[1 + sT_d + \frac{1}{sT_i} \right] \quad (21)$$

2.6 Sintonización PID:

EL proceso de sintonización consiste en obtener el valor más adecuado según las necesidades establecidas de las variables que afectaran a la función de salida y de transferencia del controlador y de la planta respectivamente (K_p, T_i, T_d).
Para realizar este proceso se tiene varias opciones entre ellas el método empírico y el método Ziegler-Nichols. [7]

2.7 Procesos de identificación de sistemas:

La realización de este proceso se centra en el obtener mediante una variedad de métodos un modelo matemático de la planta, normalmente la función de transferencia.

Para esto se suele tener señales de entrada que cumplan con diversas condiciones como: que tiene que ser superior al ruido, pero no demasiado grande para tener control de esta y debe de aportar información suficiente con la cual trabajar. [8]

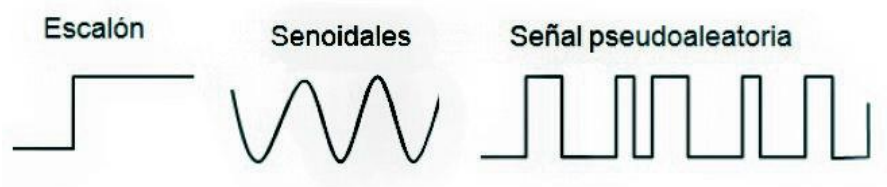


Figura 9 Funciones más usadas para identificación de sistemas. [8]

3. MARCO METODOLOGÍCO

3.1 Metodología de la investigación

Este proyecto, busca la creación de módulos didácticos para ser usados en tutorías sobre controladores PID convencional y su respectivo folleto de prácticas en el software LabVIEW.
La finalidad es que el estudiante, no solo conozca la parte teórica del mismo, si no que desarrolle su práctica, poniendo a prueba los conocimientos antes dados.

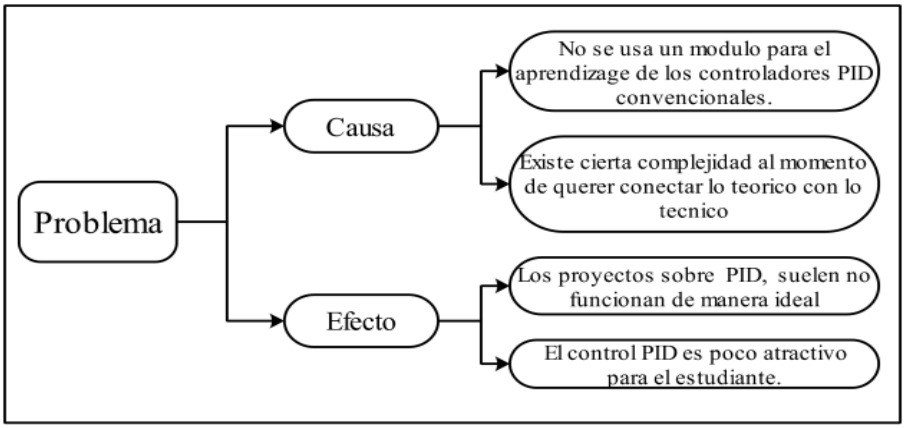


Figura 10 Árbol de causas y efectos

El método empleado para elaborar el proyecto es el método lógico inductivo, el cual indica que se parte de casos particulares y se adquiere conocimientos generales.

3.2 Técnica de investigación

Constituye al conjunto de mecanismos, y recursos dirigidos a recolectar conservar, analistas y transmitir los datos sobre lo que se investiga.

3.3 Procedimiento del proyecto

3.3.1 Primera etapa (Investigación del proyecto)

Se empezó estableciendo los componentes necesarios para cumplir con las expectativas, luego de esto fueron enlistados y cotizados los diferentes instrumentos de medición, ejecución, alimentación, control, entre otros recursos que serían necesarios.

3.3.2 Segunda etapa (Equipos y diseño)

Se verificó la disponibilidad de estos y su inter-compatibilidad, y se realizaron diversas evaluaciones y cambios, según estos fueron siendo necesarios para poder asegurar un funcionamiento directo con pocos requerimientos de acoples adicionales con el fin de simplificar las conexiones y optimizar el funcionamiento de las futuras plantas y programas.

3.3.3 Tercera etapa (Construcción de módulos y creación de prácticas)

Una vez adquiridos los materiales se procedió a realizar los prototipos con los cuales poder hacer las pruebas y comprobar el funcionamiento de sus contrapartes virtuales, así como las diferentes posibilidades de experimentación que estos brindaran.
Se realiza los diseños en AutoCAD, de los cases de los módulos para tener una visión más específica de lo que procederá a realizar.

3.3.4 Cuarta etapa (Finalización de las instalaciones, pruebas finales)

Se finaliza el completo ensamblaje del módulo con todos sus elementos, comienza la prueba del equipo con todos sus elementos, se monitorea su correcto funcionamiento en varios días. Se corrigen errores y soluciona problemas en el módulo.

3.4 IMPLEMENTACIÓN DEL PROYECTO

3.4.1 Diseño del proyecto

Para el diseño e implementación del prototipo se procede a establecer las practicas a realizar, iniciando con la creación de los circuitos en el software PROTEUS, verificando si cumple con los requerimientos establecidos.

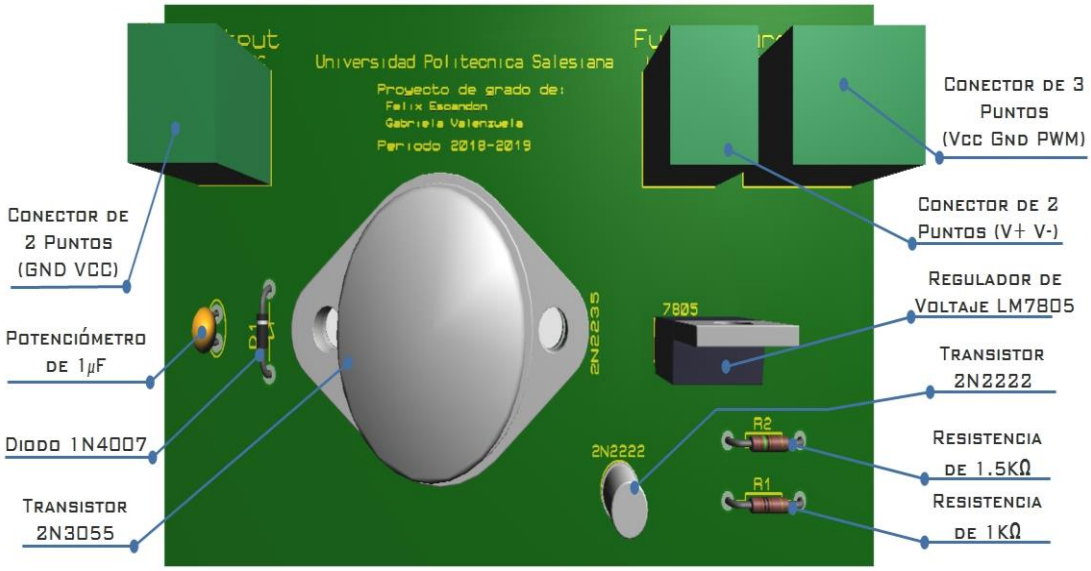


Figura 11 Diseño del puente

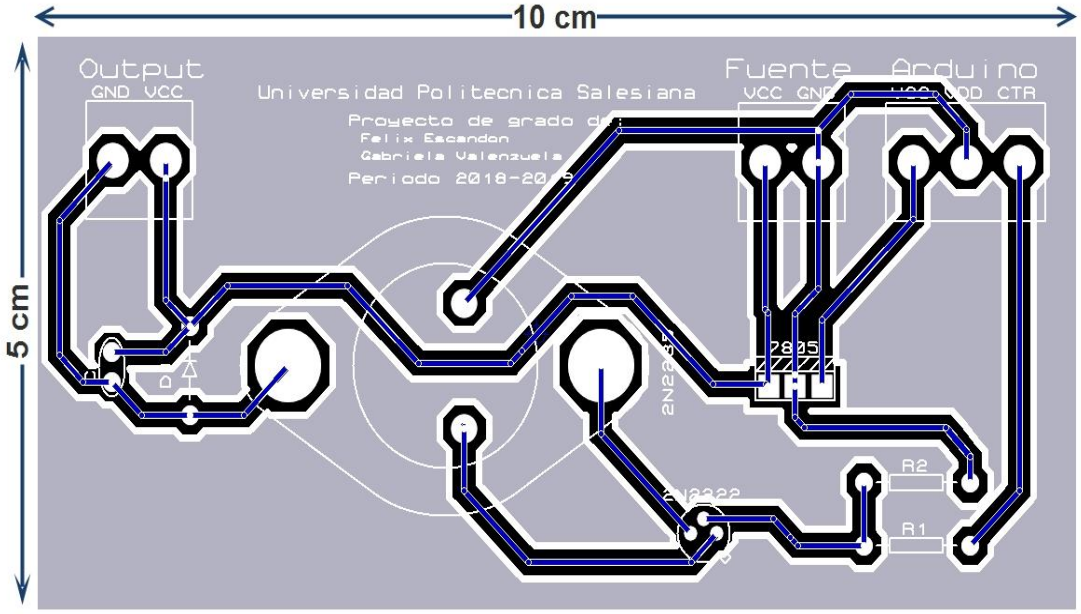


Figura 12 Diseño de la pista del puente

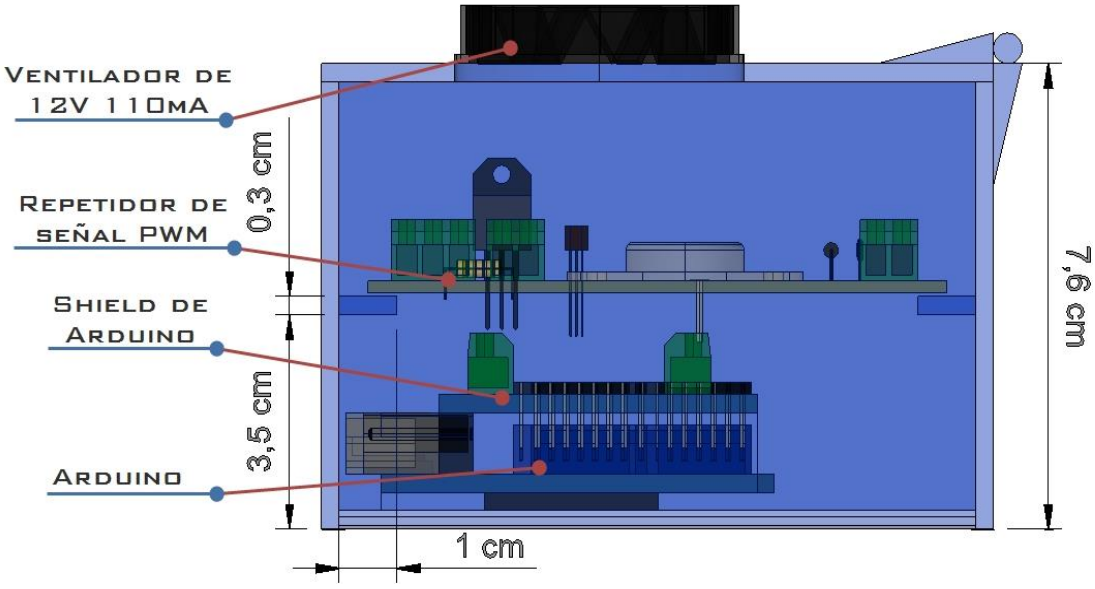


Figura 13 Vista lateral del gabinete plástico

3.4.2 Creación de la estructura

Para los diseños de las estructuras físicas se utilizó el software AutoCAD con el cual se diseñaron diferentes diseños dependiendo del módulo que se iba a realizar, estos diseños fueron enviados a empresas que pudiesen tanto imprimirlos en 3D o realizar estas estructuras en acrílico según se requiriese.

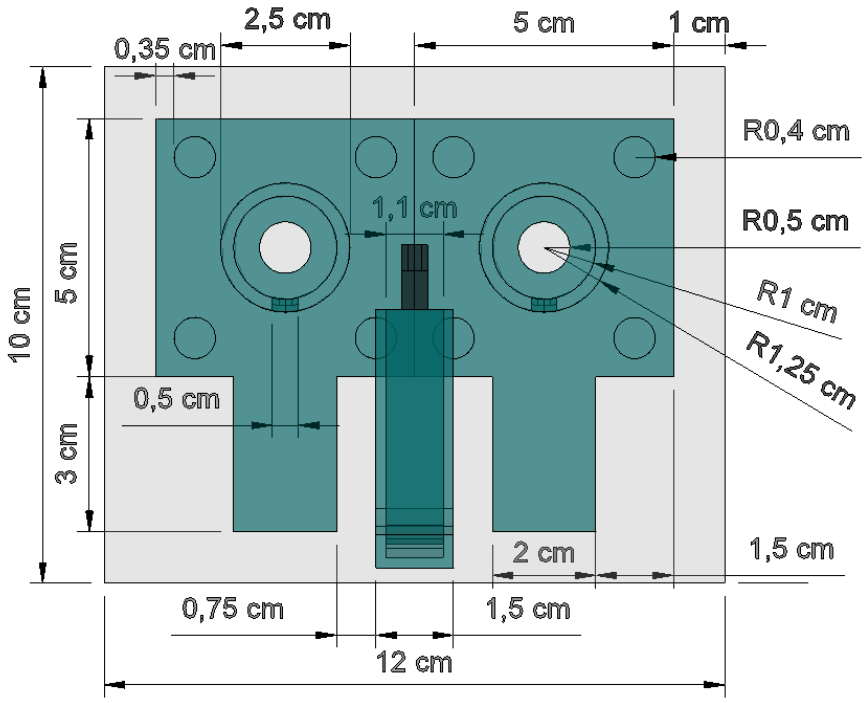


Figura 14 Vista superior de la Planta Térmica

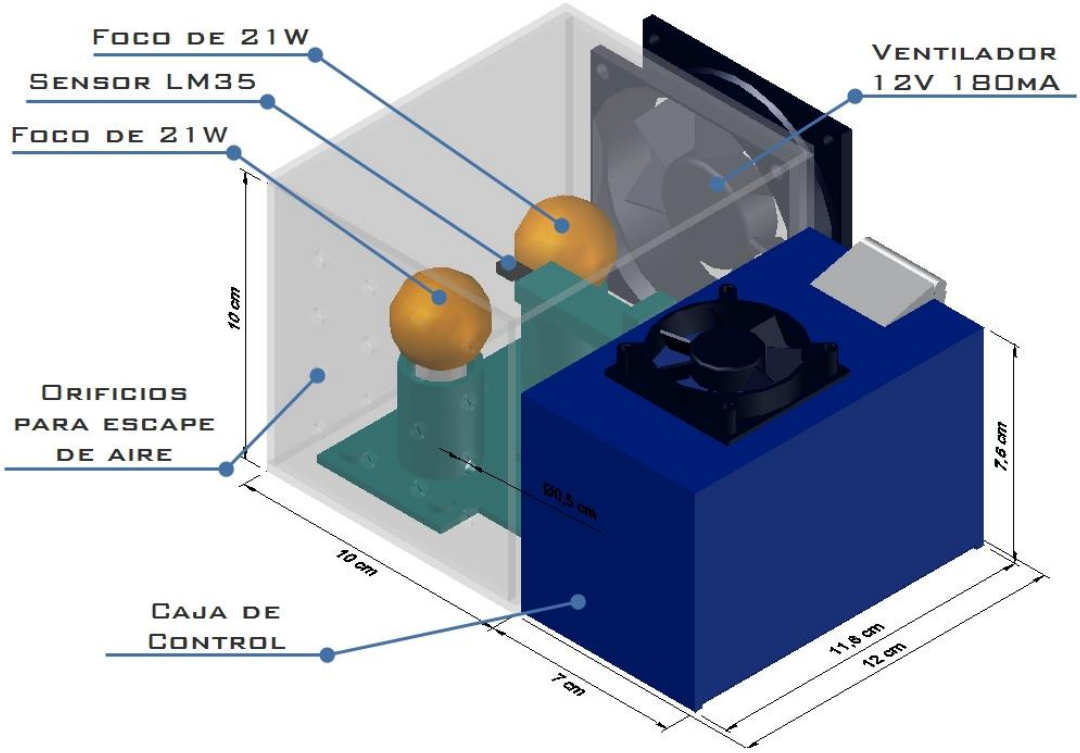


Figura 15 Vista isométrica de la Planta Térmica

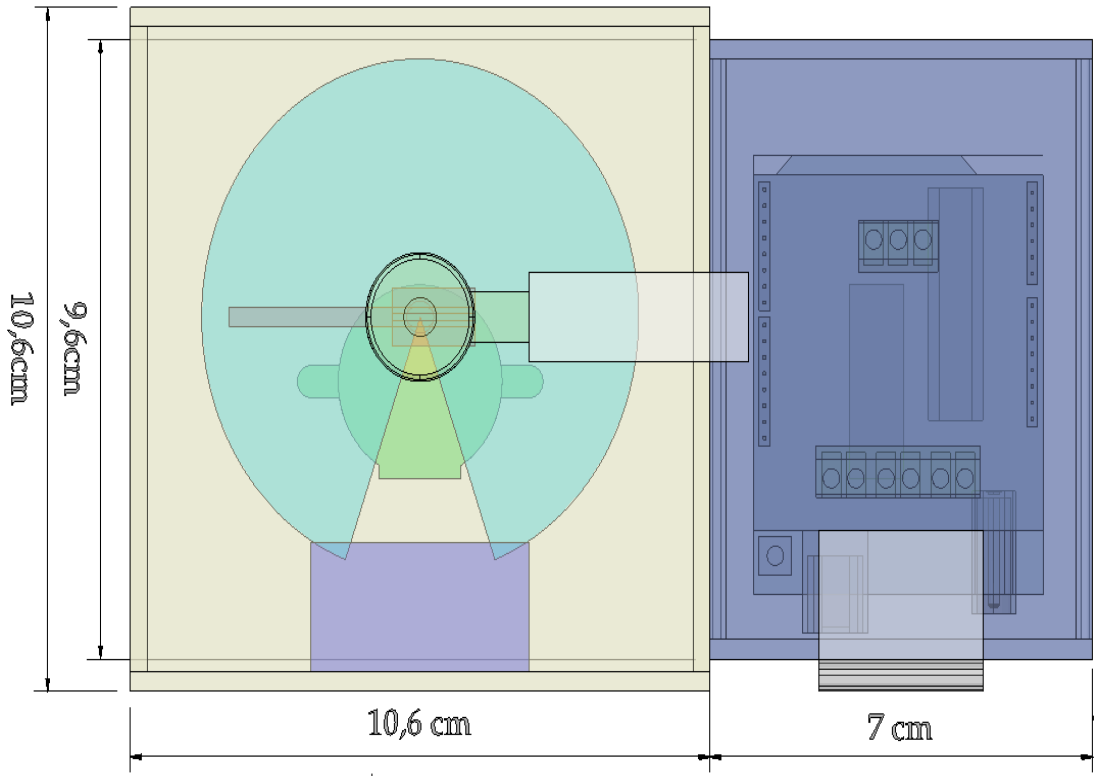


Figura 16 Vista superior de la Planta Motor de Paso

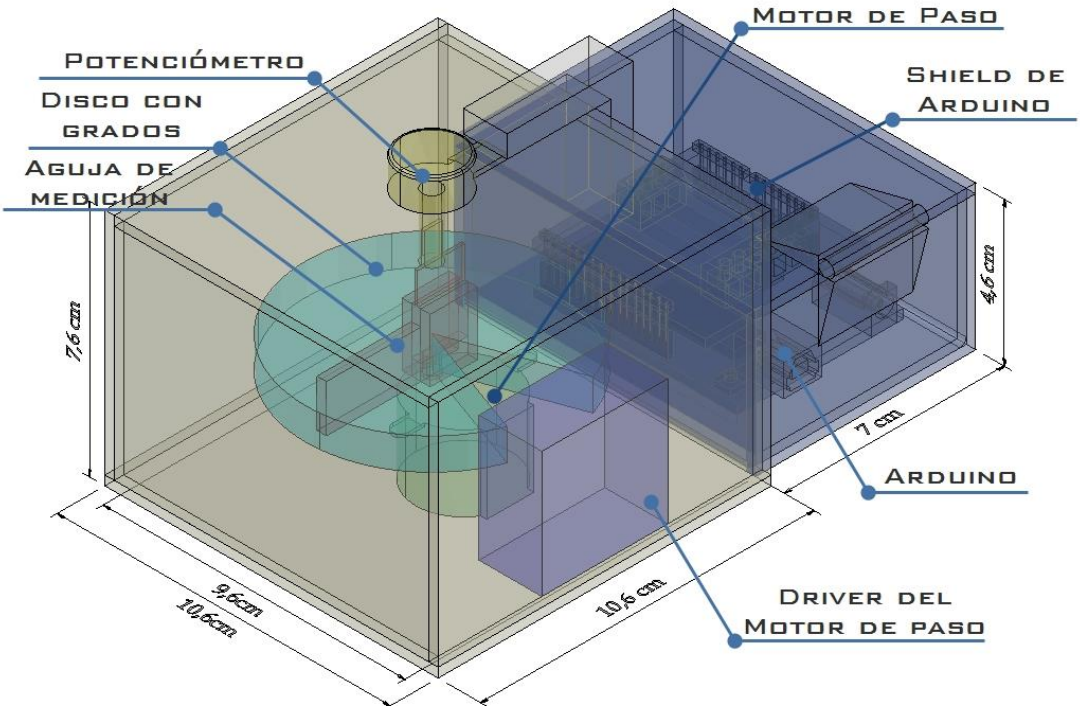


Figura 17 Vista isométrica de la Planta Motor de Paso

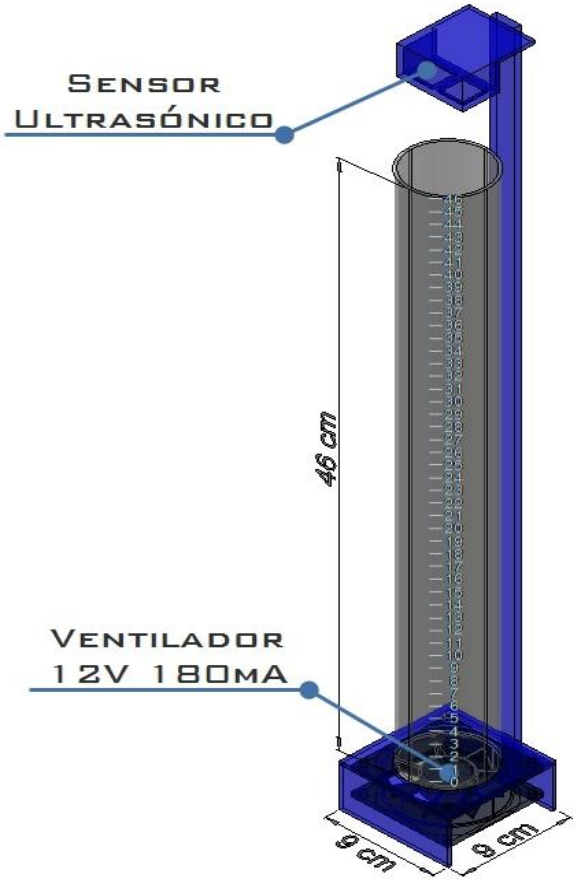


Figura 18 Vista isométrica la Planta de Levitación

3.4.3 Diseño de los ambientes virtuales

Dentro de lo referente a lo virtual se deberá asegurar contar con los programas que se desean utilizar para generar todos los ambientes y herramientas virtuales que se contemplen necesarios para el funcionamiento del proyecto, así como las diversas librerías que estos requieran para su comunicación y la utilización de los componentes a utilizar como:

- Toolkit LabVIEW-Arduino
- Control Desing and Simulation
- Math script advance
- Linx de MakerHub

Teniendo listos los prototipos y los programas iniciales y conservando un adecuado orden y fragmentación de los diferentes archivos y componentes para su clara identificación y modificación, se procederá de manera sistemática y ordenada, junto con un continuo archivar a modo de historial de los procedimientos a ejecutar conforme estos van pasando por las etapas de edición hasta llegar a su forma deseada.
Para lograr esto se recomienda tener dos o tres carpetas separadas, en estas tener conjuntos de archivos a manera de versiones para conservar un historial el cual revisar en caso de obtener fallas en el proceso, y una carpeta principal en la cual se encuentre de forma ordenada tal como se busca entregar en donde estén los archivos finales divididos en subcarpetas según se requiera, de forma opcional la tercera carpeta será utilizada para almacenar los archivos diagramas, diseños, entre otros, que se vayan generando durante el proceso.

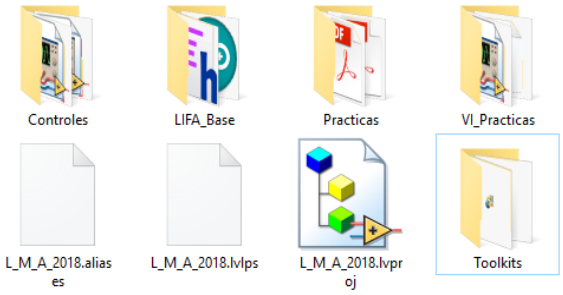


Figura 19 Archivos del proyecto

3.4.4 Diseño de las prácticas

Teniendo ensambladas las partes y los prototipos formados ya como un producto terminado, se realizarán las pruebas de cada uno y posterior a esto se realizarán las diversas prácticas en Microsoft Word que contengan las

instrucciones, información, temáticas y demás datos necesarios para cumplir con las tutorías de los diversos temas a tratar en la práctica.
Durante el proceso es posible que se genere información adicional y se abarquen temas que originalmente no pertenecían a los vislumbrados para las practicas, para esto de querer agregarlos para enriquecer el proyecto se deberá adaptar y reorganizar los datos y prácticas.

- Criterio de Estabilidad de Ruth Hurwitz
- Introducción
- Modelado
- Valores de PID
- PID
- PID discreto
- Levitador Eólico
- Señal PWM
- Sistemas No Lineales

3.5 Proceso de validación

Dentro del proceso de control, y generación de modelos que emulen el comportamiento de una determinada planta.

Dado que los modelos no son ideales, se busca determinar el marguen de diferencia que existe entre la respuesta de estos modelos y la de la planta real, para esto una de las herramientas que se utilizan es la del error medio cuadrático.

3.5.1 Pasos para la validación

3.5.1.1 Elaboración de validación para circuito RC en Matlab

Se utilizará una librería para Simulink de Matlab que permite controlar al Arduino por medio de bloques de programa, de nombre "ArduinoIO".

Para probar este conjunto de programas realizados utilizando esta librería, se realizará un circuito RC sencillo y se obtuvo su modelo con el VI de Modelado General, esto debido a las limitaciones que tiene esta librería para el control de sensores.

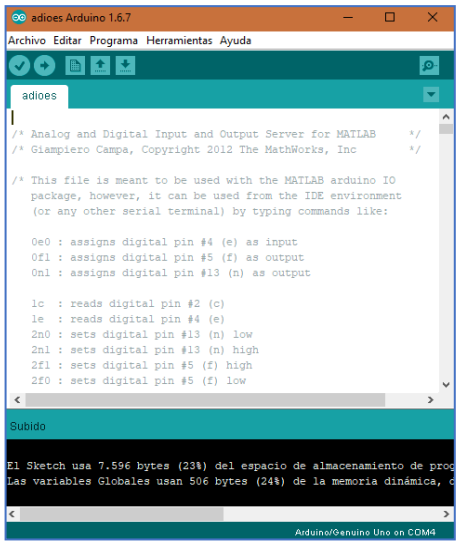


Figura 20 Ventana de la librería a usar

Luego de esto, se utilizará el programa de nombre "Iniciador" el cual tiene como función el llevar a un valor de 0 el pin de control previo a realizar la simulación y comparación con la planta real.

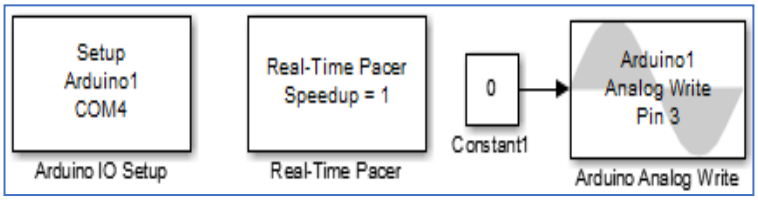


Figura 21 Circuito Iniciador

En la figura se muestra programa de Simulink de nombre Validación permite realizar la comparación de los valores en tiempo real y adicionalmente genera un listado con la resta al cuadrado de cada uno de los valores obtenido a lo largo de la simulación, para luego de esto realizar su suma y división para la cantidad de valores obtenidos.

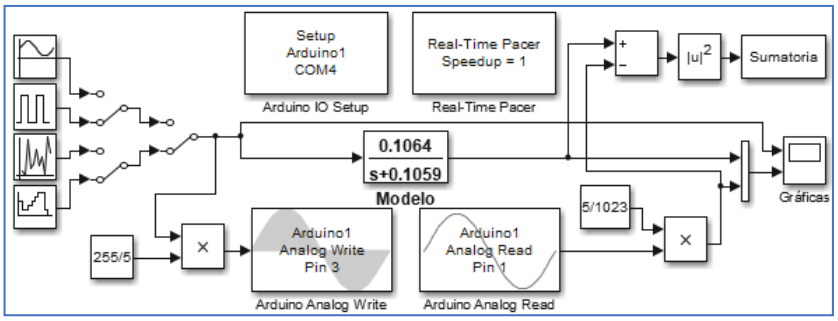


Figura 22 Diseño del circuito en Simulink para validación de circuito RC

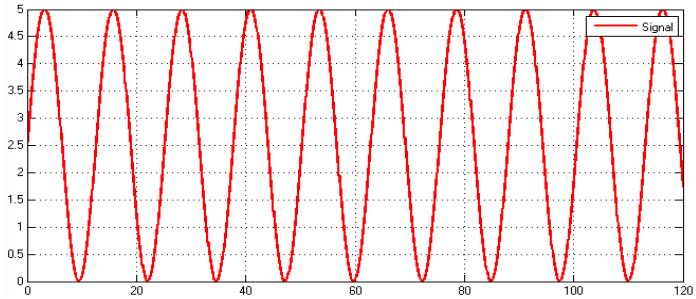


Figura 23 Señal de entrada

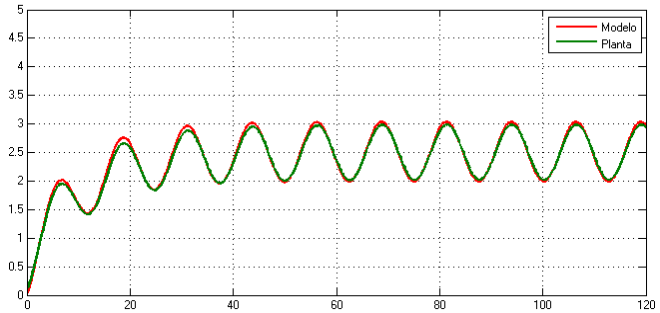


Figura 24 Comparación del comportamiento Real vs Simulado

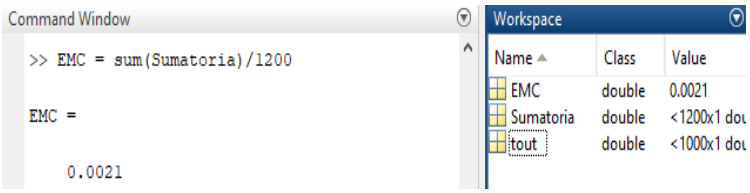


Figura 25 Error medio cuadrático

3.1.2 Elaboración de validación para circuito RC en LabVIEW

Se utilizará las dos librerías empleadas lo largo del proyecto de LINX y Arduino-LabVIEW,

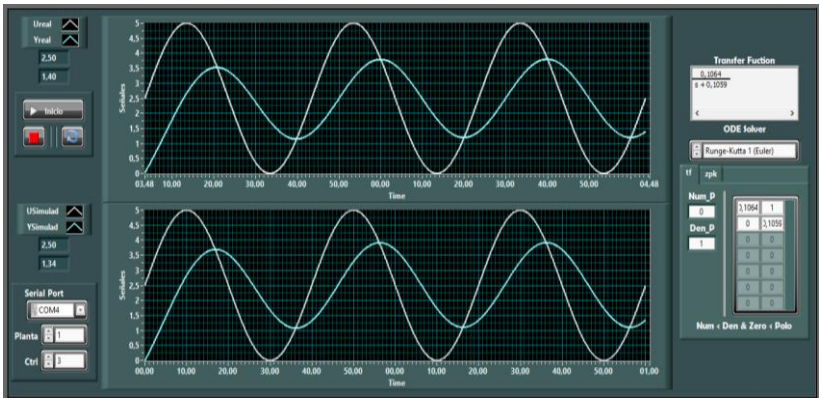


Figura 26 VI validación circuito RC

	A	B	C	D	E	F	G	H	I
1	Time - Ureal	Señales - Ureal	Time - Yreal	Señales - Yreal		Time - USimu	Señales - USimu	Time - YSimu	Señales - YSimulada
2	3,57	2,539	3,48	0,029		0	2,5	0	0
3	3,67	2,579	3,58	0,049		0,1	2,539	0,1	0,027
4	3,77	2,618	3,68	0,078		0,2	2,579	0,2	0,053
5	3,87	2,657	3,78	0,103		0,3	2,618	0,3	0,08
6	3,97	2,696	3,88	0,127		0,4	2,657	0,4	0,107
7	4,07	2,735	3,98	0,151		0,5	2,696	0,5	0,134
8	4,17	2,774	4,08	0,176		0,6	2,735	0,6	0,162
9	4,27	2,813	4,18	0,205		0,7	2,774	0,7	0,189
10	4,37	2,852	4,28	0,229		0,8	2,813	0,8	0,217
11	4,47	2,891	4,38	0,254		0,9	2,852	0,9	0,244
12	4,57	2,93	4,48	0,278		1	2,891	1	0,272
13	4,67	2,968	4,58	0,308		1,1	2,93	1,1	0,3
14	4,77	3,007	4,68	0,332		1,2	2,968	1,2	0,328
15	4,87	3,045	4,78	0,356		1,3	3,007	1,3	0,356
16	4,97	3,084	4,88	0,381		1,4	3,045	1,4	0,384
17	5,07	3,122	4,98	0,405		1,5	3,084	1,5	0,412
18	5,17	3,16	5,08	0,435		1,6	3,122	1,6	0,441
19	5,27	3,197	5,18	0,459		1,7	3,16	1,7	0,469
20	5,37	3,235	5,28	0,488		1,8	3,197	1,8	0,498

Figura 27 Recolección de datos

Error Medio Cuadratico
Señal de salida
0,00847811
Error Medio Cuadratico
Señal de entrada
0,000771033

Figura 28 Error medio cuadrático

3.6 Validación Motor de paso

Se utilizará la versión 2 del VI para controlar manualmente el motor de paso para asegurar que este empiece desde 30 grados que es el punto 0 de referencia para el rango de operación que va a utilizarse.

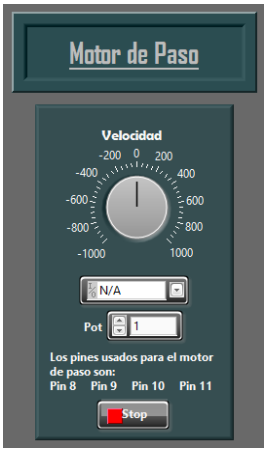


Figura 29 VI de control manual motor de paso

Luego de esto se utilizará el VI de modelado para motor a pasos, con el fin de obtener un conjunto de muestras con el cual desarrollar un modelo por identificación utilizando ident de Matlab.

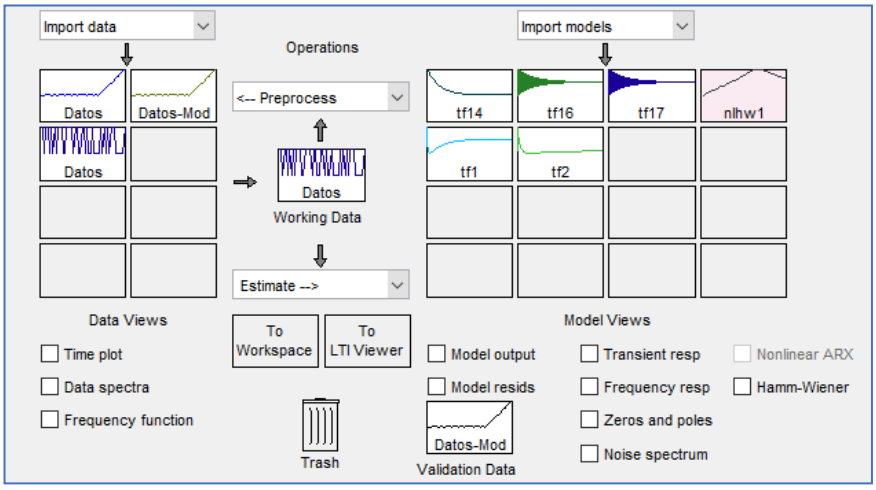


Figura 30 Ventana Ident

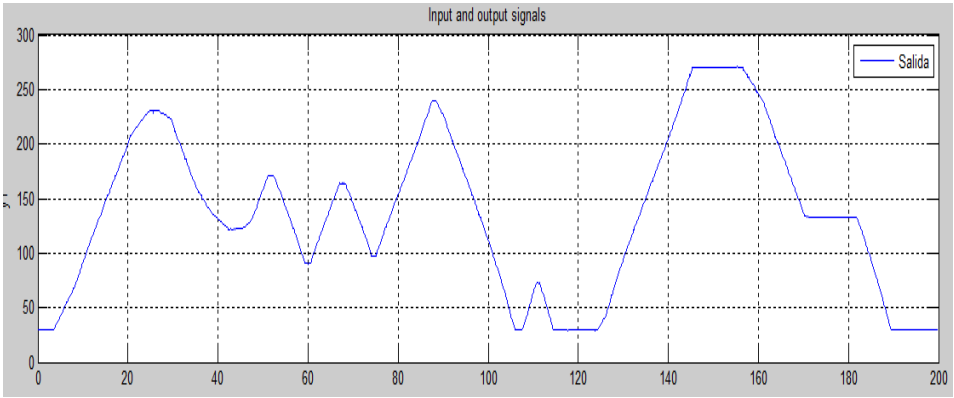


Figura 31 Señal de salida

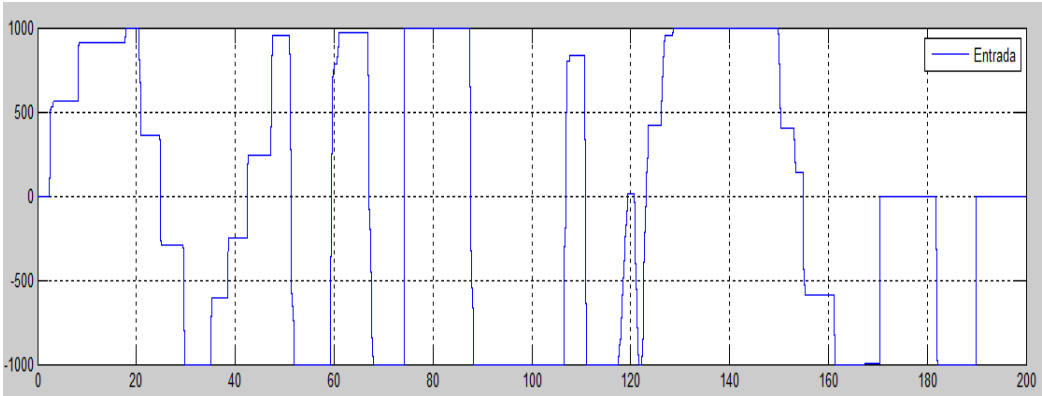


Figura 32 Señal de entrada

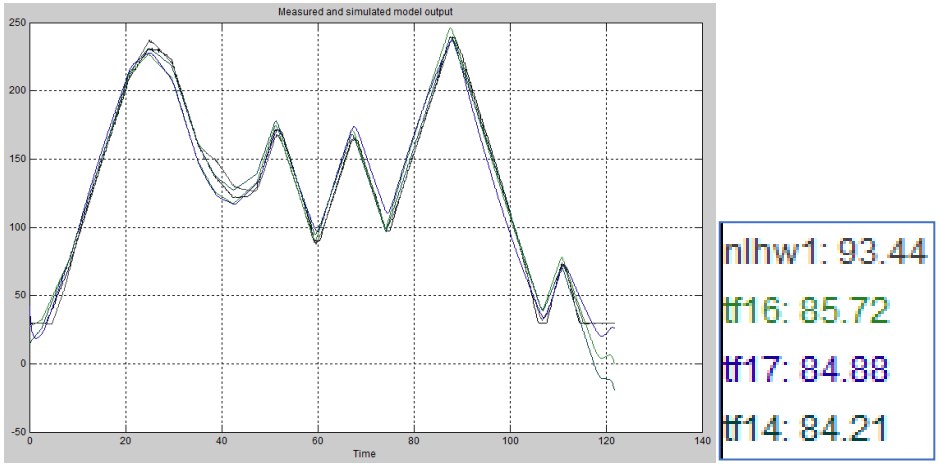


Figura 33 Diferentes modelos con su porcentaje de aproximación

Se utilizará por motivos la función con mayor aproximación obtenida

```
  From input "u1" to output "y1":
     0.01113 s^2 + 0.008653 s - 9.076e-05
  ----------------------------------------
  s^3 + 0.7861 s^2 + 4.249e-05 s + 3.312e-05
Name: tf16
```

Figura 34 Función de transferencia utilizada “nlhw1”

Finalmente, para comprobar el modelo obtenido se realiza la comparación y obtención de datos para realizar el cálculo del error medio cuadrático.

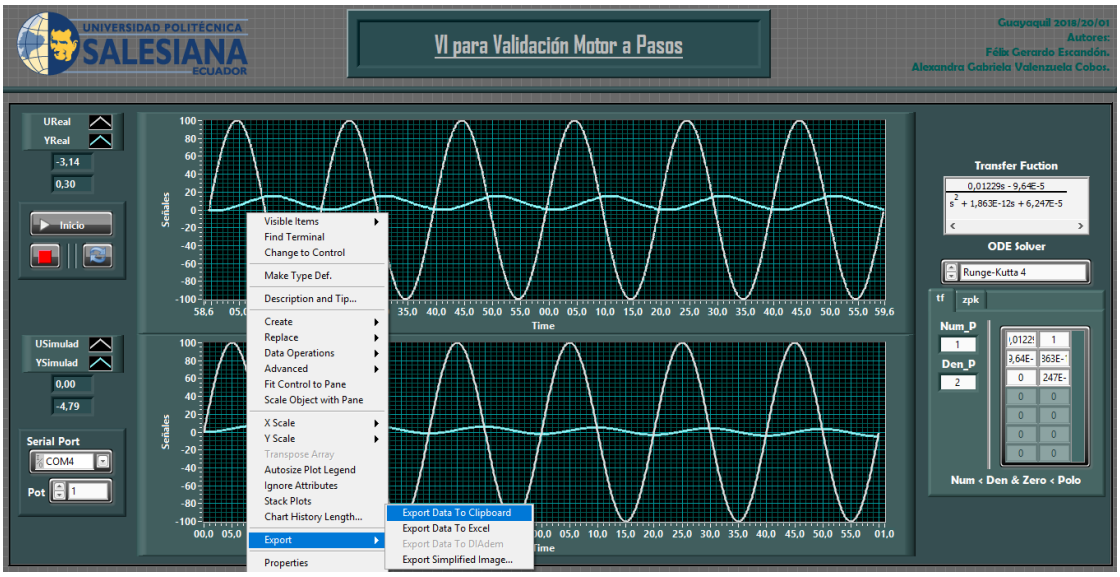


Figura 35 Exportación de datos

Para esto se realiza la exportación de los datos de las gráficas a una hoja de EXEL, y en esta se realiza por medio de fórmulas la obtención primero de un conjunto de valores restados y elevados al cuadrado y luego la suma y división de estos para un valor de 1200 muestras.

Error Medio Cuadratico
Señal de salida
46,111474
Error Medio Cuadratico
Señal de entrada
1,72275E-28

Figura 36 Error medio cuadrático

De esta manera se puede obtener y realizar varias muestras y cambiar solamente las tablas de datos originales, dando mayor dinamismo al proceso, pudiendo evaluarse varias funciones de manera rápida.
En ocasiones el comportamiento de un modelo difiere al evaluarse de esta manera, siendo en este caso un modelo de menor orden dio un mejor resultado que el modelo seleccionado, cuyas respuestas se muestran a continuación.

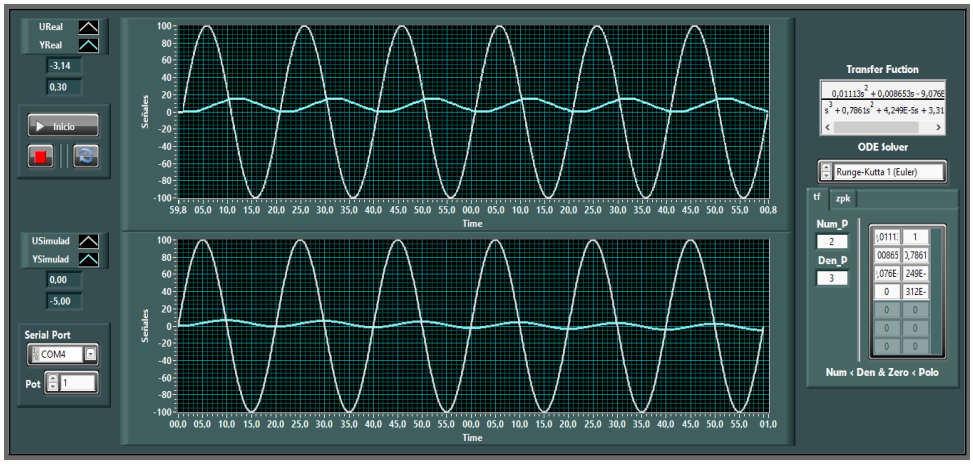


Figura 37 Validación con modelo de orden 2

Error Medio Cuadratico
Señal de salida
54,67787882
Error Medio Cuadratico
Señal de entrada
1,72275E-28

Figura 38 Error medio cuadrático

3.7 Planta térmica 1

En el caso de la planta térmica se utilizará dos conjuntos de muestras para la evaluación y obtención del modelo a utilizar, así como la validación de este, tal como se muestra en las imágenes a continuación, cabe recalcar que se realizará un filtrado y recorte de la señal para evaluar las secciones más adecuadas de estas.

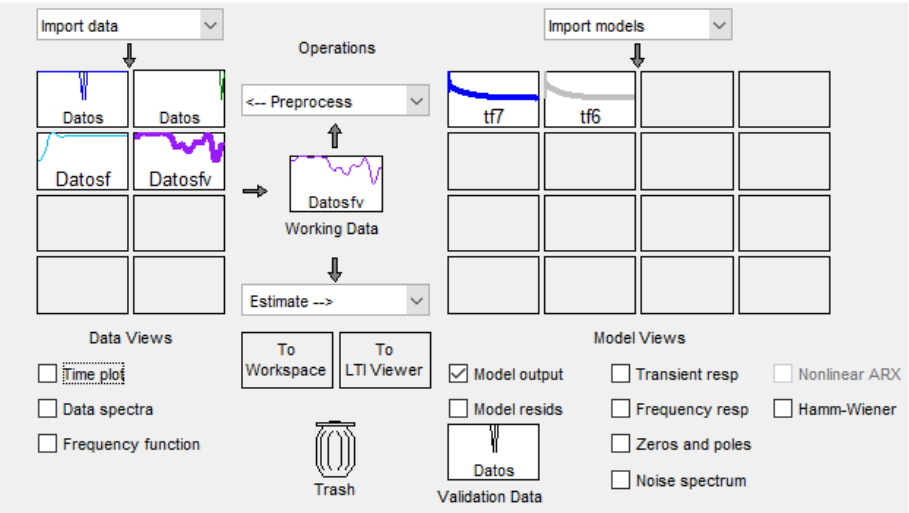


Figura 39 Ventana Ident

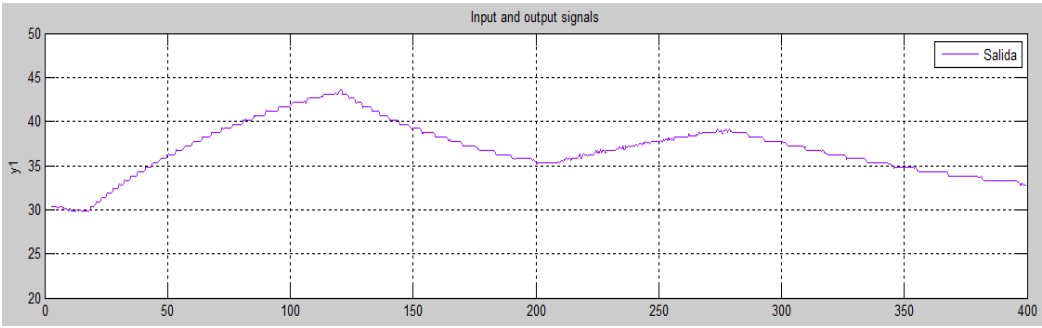


Figura 40 Señal de salida

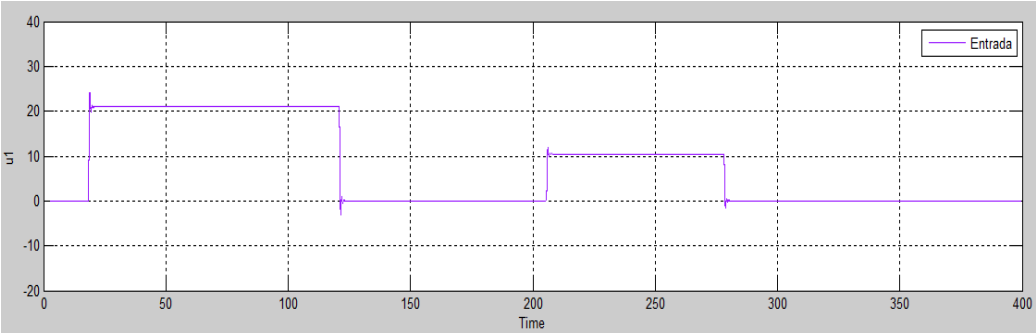


Figura 41 Señal de entrada

```
From input "u1" to output "y1":
  0.0024 s^2 + 0.02352 s + 0.0002566
  ----------------------------------------
  s^3 + 1.983 s^2 + 0.03998 s + 4.186e-05
ame: tf7
```

Figura 42 Función de transferencia “tf7”

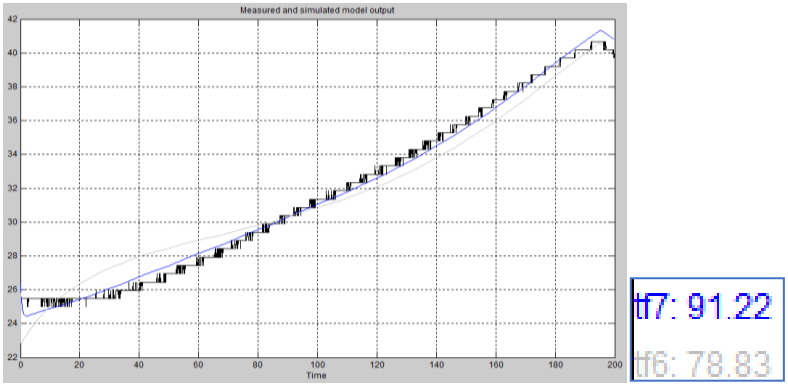


Figura 43 Conjunto de muestras 1, con su proximidad

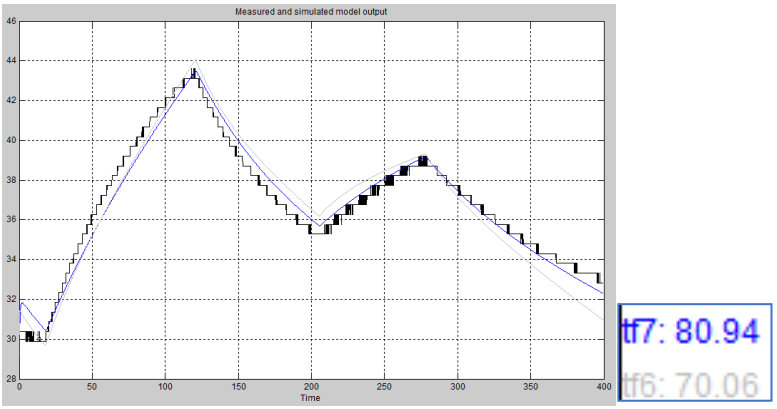


Figura 44 Conjunto de muestras 2, con su proximidad

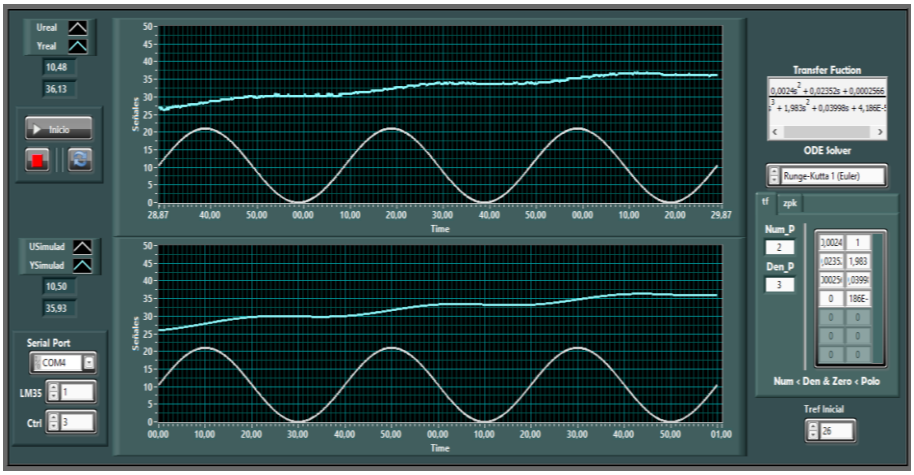


Figura 45 Validación de la planta térmica 1

Figura 46 Error medio cuadrático

3.8 Planta térmica 2

Se realiza proceso similar al anterior. Dando los siguientes resultados

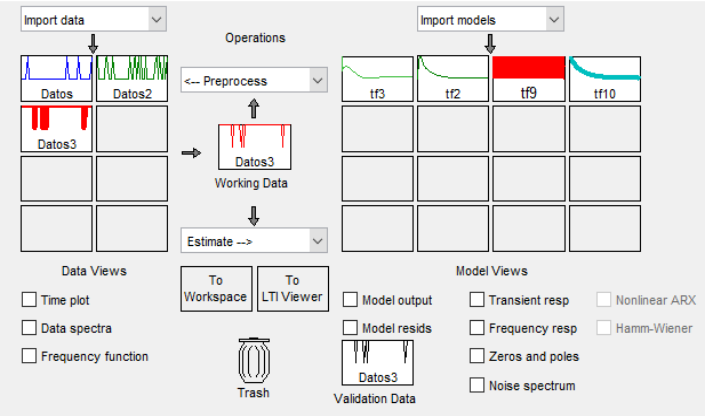


Figura 47 Ventana Ident

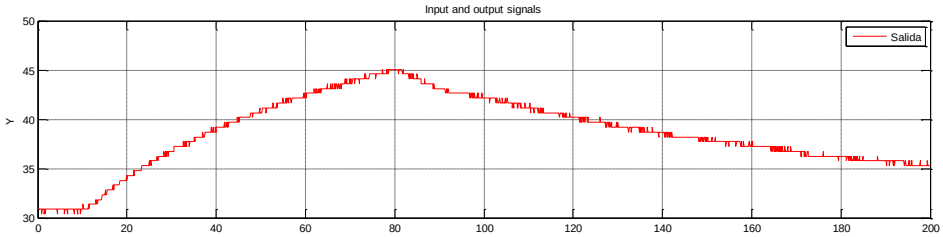


Figura 48 Señal de salida

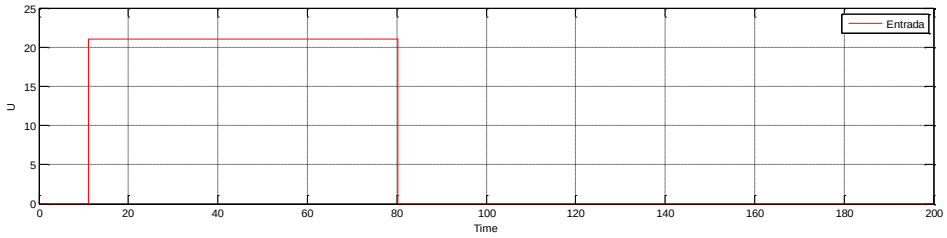


Figura 49 Señal de entrada

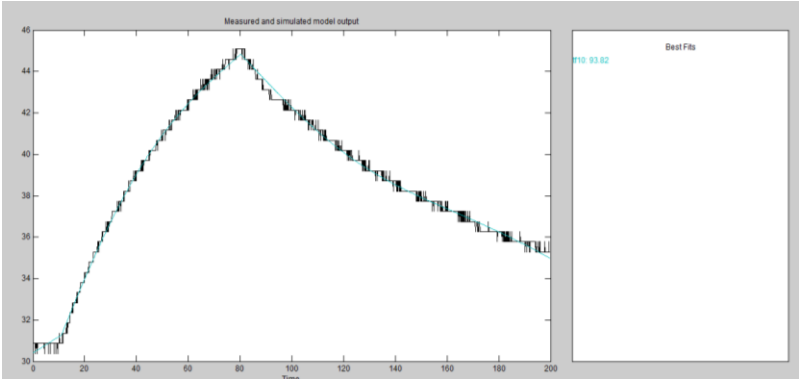


Figura 50 Conjunto de muestras, con su proximidad

```
From input "U" to output "Y":
   0.01135 s^2 + 0.000287 s + 2.504e-05
  ----------------------------------------
  s^3 + 0.02334 s^2 + 0.002111 s + 4.133e-06
```

Figura 51 Función de transferencia “tf10”

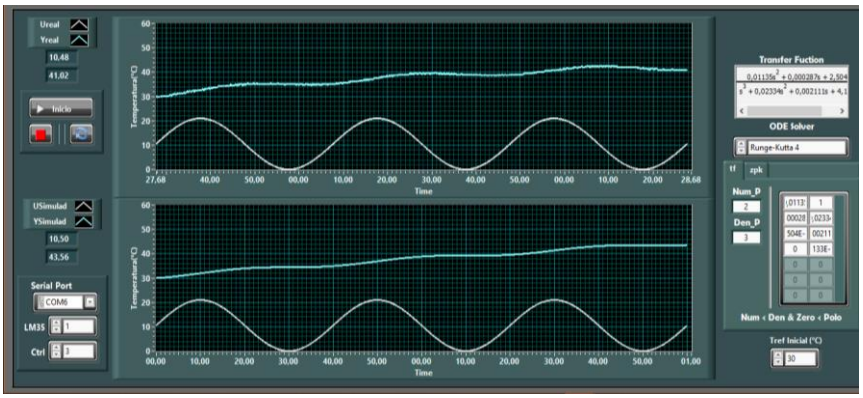


Figura 52 Validación Planta térmica

Error Medio Cuadratico
Señal de salida
0,916937736
Error Medio Cuadratico
Señal de entrada
0,013597393

Figura 53 Error medio cuadrático

3.9 Planta Levitadora 1

En esta planta se obtuvo los datos de la respuesta del sistema al recibir dos niveles de tensión diferentes los cuales fueron 7 y 5 voltios, representados por los valores de 2 y 0 en la barra de voltaje de entrada

Figura 54 Respuesta de la simulación

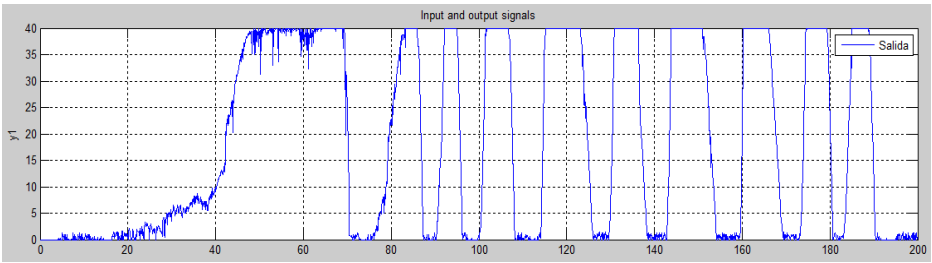


Figura 55 Señal de salida

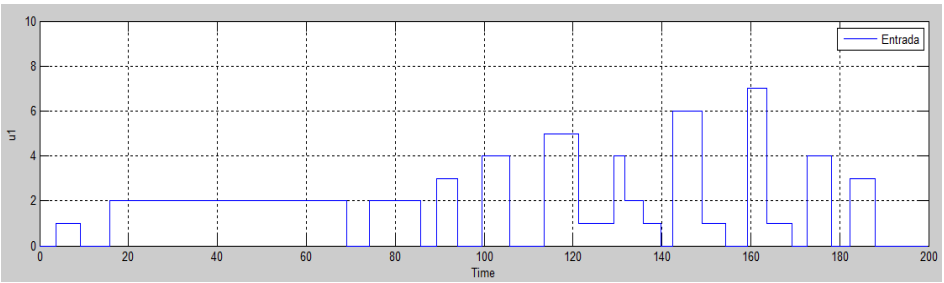


Figura 56 Señal de entrada

Teniendo las muestras necesarias y haciendo varias pruebas se obtuvieron varios modelos con los cuales se procedió a su evaluación en el VI de validación.

Cabe destacar que, si bien algunos modelos tenían valores más altos de aproximación, su desempeño al realizar la simulación con una señal diferente demostraba que no eran tan buenos, mientras que otros de similar pero menor porcentaje, dieron un mejor desempeño en replicar el comportamiento de la planta.

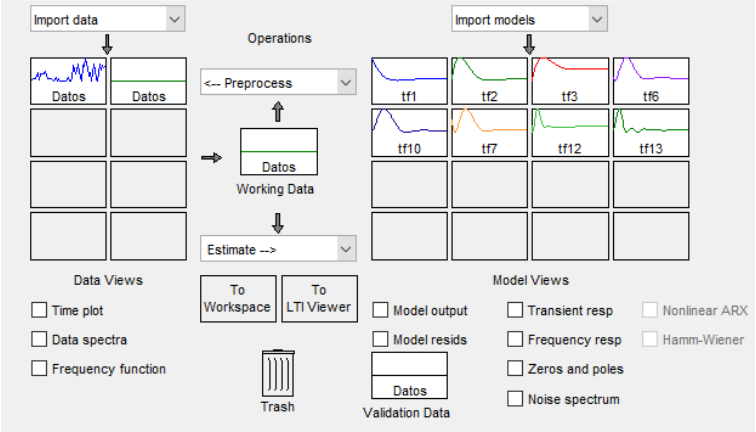


Figura 57 Ventana Ident

Los modelos que se muestran en imagen de la ventana del ident se realizaron utilizando ambos conjuntos de muestras y comparando con los mismos su desempeño, las tres primeras señales fueron hechas utilizando el primer conjunto de datos, de color azul, mientras que las demás señales se hicieron utilizando el segundo conjunto de muestras.

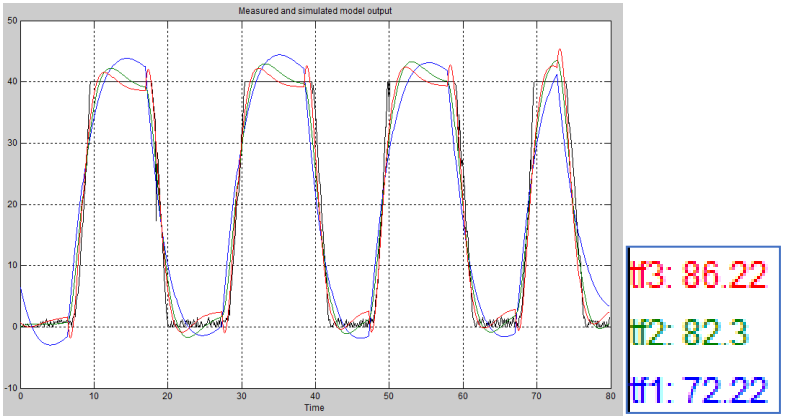


Figura 58 Primer conjunto de muestras con su proximidad

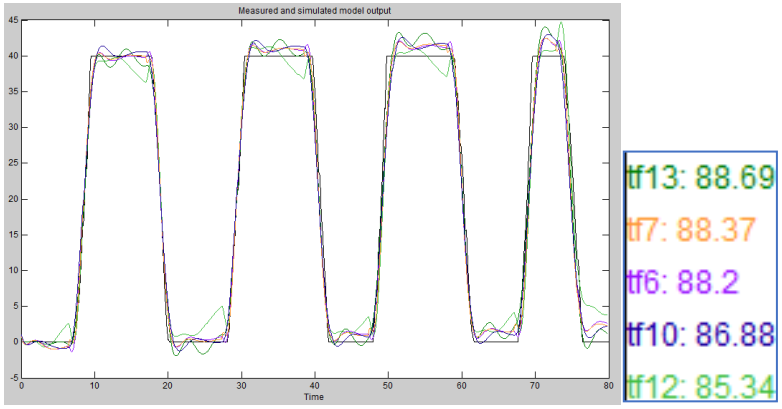


Figura 59 Segundo conjunto de muestras con su proximidad

De estos dos grupos de funciones se determinó el uso de tf1 y tf12 por su comportamiento más acercado al de la planta al evaluar estos en Matlab utilizando el comando "plot" y realizando una simulación lineal con la señal a la que se sometería la planta real y la simulada para la comparación y posterior obtención del error medio cuadrado.

```
From input "u1" to output "y1":
     7.266 s + 0.5523
  ------------------------
  s^2 + 0.3522 s + 0.05137
```

Figura 60 Función de transferencia 13

```
From input "u1" to output "y1":
          -20.63 s^3 + 26.64 s^2 + 6.546 s + 0.3379
  ------------------------------------------------------------
  s^5 + 2.095 s^4 + 4.081 s^3 + 2.412 s^2 + 0.5252 s + 0.03136
Name: tf12
Continuous-time identified transfer function.
```

Figura 61 Función de transferencia 12

Para el caso del primero se obtuvieron los siguientes resultados.

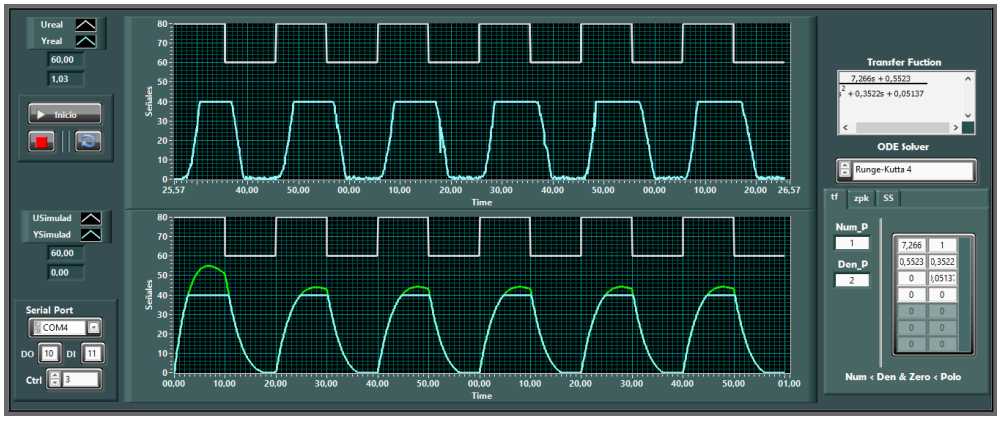


Figura 62 Validación para el caso 1

Error Medio Cuadratico
Señal de salida
48,8770076
Error Medio Cuadratico
Señal de entrada
9,666666667

Figura 63 Error medio cuadrático caso 1

Con el segundo modelo seleccionado se obtuvieron los siguientes resultados

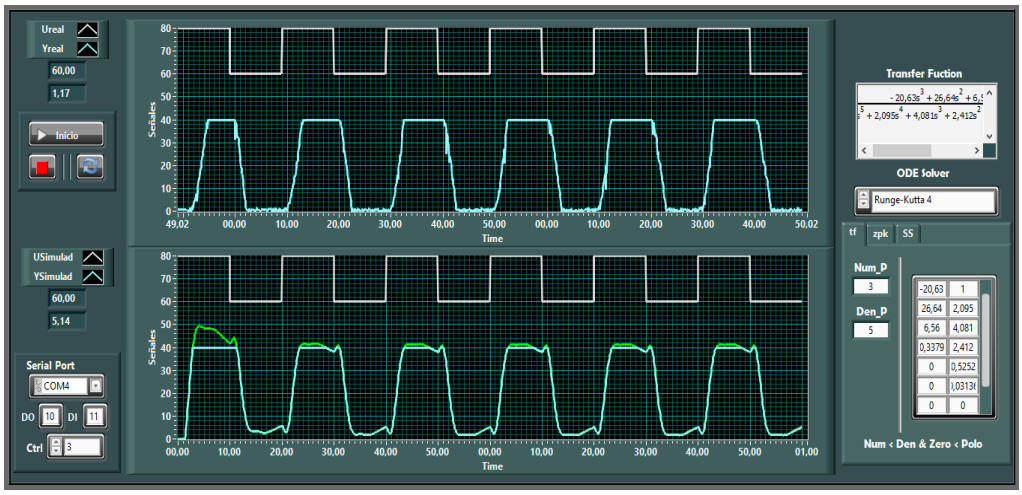


Figura 64 Validación para el caso 2

3.9 Planta Levitadora 2

Se realiza proceso similar al anterior. Dando los siguientes resultados

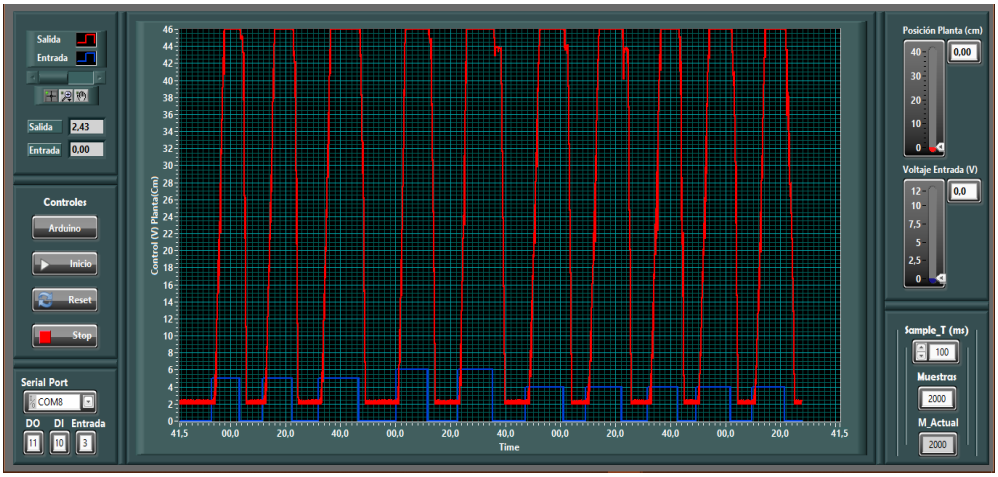


Figura 65 Respuesta de la simulación

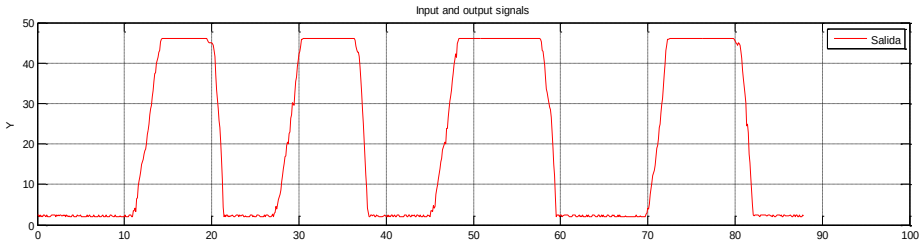


Figura 66 Señal de salida

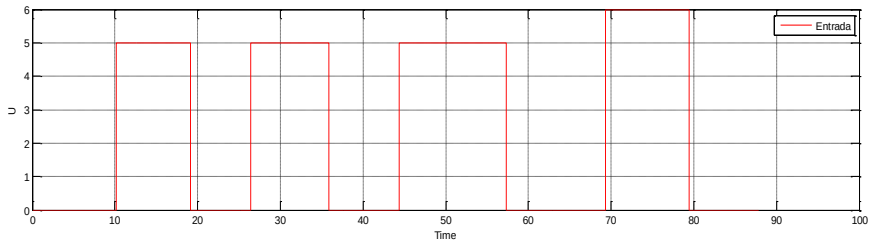


Figura 67 Señal de entrada

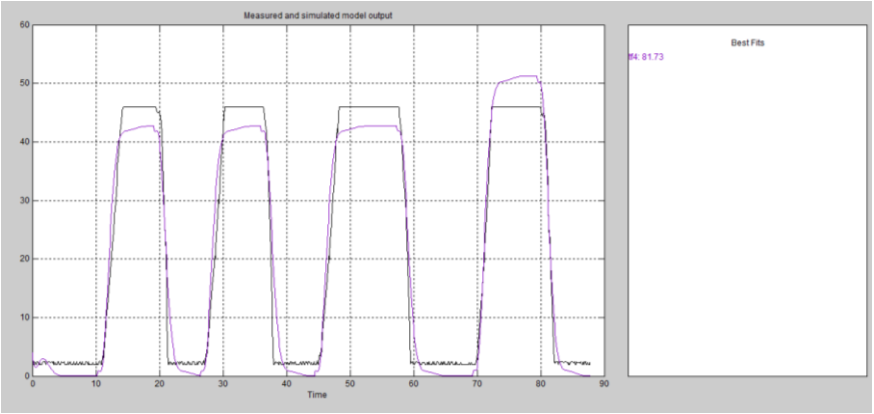


Figura 68 Conjunto de muestras con su proximidad

```
From input "U" to output "Y":
   1.909 s^2 - 7.978 s + 24.39
--------------------------------
s^3 + 2.915 s^2 + 4.909 s + 2.854
```

Figura 69 Función de transferencia tf4

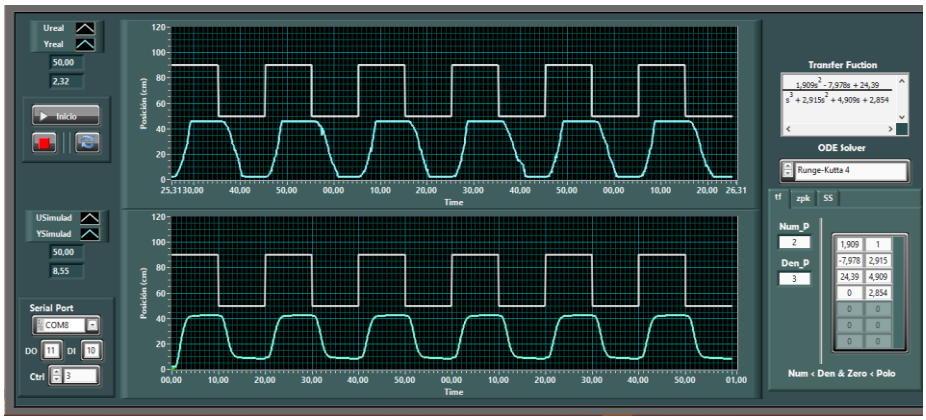


Figura 70 Respuesta de la simulación

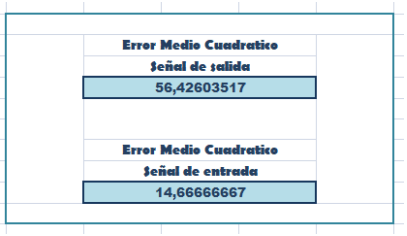


Figura 71 Error medio cuadrático

4. Prácticas de Laboratorio

4.1 Práctica #1 “Introducción”

4.1.1 Objetivos

- Identificar los Toolkits añadidos en LabVIEW.
- Repasar los recursos de la librería de LabVIEW-Arduino.
- Construir un programa básico utilizando la librería LabVIEW-Arduino.
- Repasar los recursos de la librería LINX de MakerHub.
- Construir un programa básico utilizando la librería de LINX.
- Repasar los conocimientos elementales de Ingeniería de control.

4.1.2 Toolkit LabVIEW-Arduino

Este Toolkit permite realizar una comunicación serial entre un Arduino y LabVIEW, por ello durante el proceso los puertos 0 y 1 digitales del Arduino son usadas en la comunicación con LabVIEW por medio de un puerto USB, entre otras opciones.

La forma más fácil de asegurar que la instalación de este Toolkits es correcta, es revisando en VI Pack Age Manager, el cual es indicado instalar en el paso 1 de la imagen anterior, y en el paso 2 se indica la realización de la descarga e instalación del Toolkits de LabVIEW-Arduino.

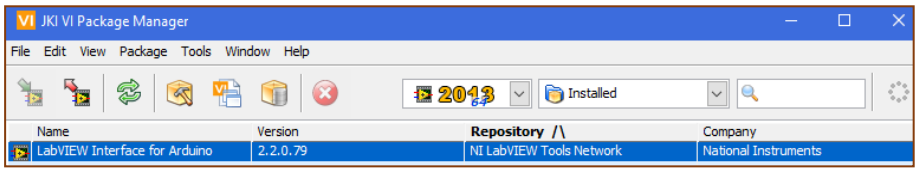


Figura 72 Toolkit para Arduino

Como puede apreciarse, se está utilizando el Toolkit para LabVIEW 2013, de igual manera se utiliza ese mismo LabVIEW, aclarando que se usa la versión de 32 bits y no la de 64 bits, por un asunto de utilidad dado que el otro Toolkit que se va a usar solo existe para esa versión hasta la fecha en que fue desarrollado este proyecto.

No obstante, el Toolkit de LabVIEW-Arduino esta tanto para la versión de 32 bits como para la de 64 bits, y es indiferente cuál de las dos se desee usar si no se va a dar uso de los otros Toolkits exclusivos de la versión de 32 bits; ya que ambas versiones pueden ser usadas para controlar a través de LabVIEW a un Arduino.

4.1.3 Toolkit LINX

De manera adicionar en los VI de prácticas más avanzadas se da uso a una librería diferente, esta librería viene de mano de un grupo de desarrolladores llamados MakerHub los cuales ofrecen la librería de LINX.

Esta librería se la puede obtener e instalar de manera similar a la anterior vista, para empezar, se deberá seleccionar la opción de "Descarga de LINX". Luego de esto se seleccionará "Down load Toolkit" con lo cual se abrirá la ventana de JKI VI Pack Age Manager.

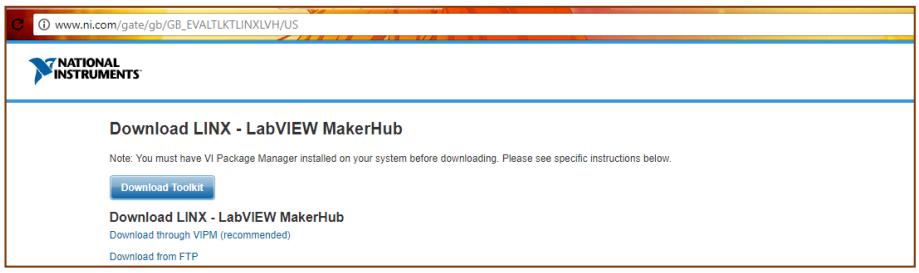


Figura 73 Proceso de descarga LINX

JKI se encargará del proceso de descarga e instalación, solicitando la confirmación de continuación al usuario y el aceptar con los términos de uso.

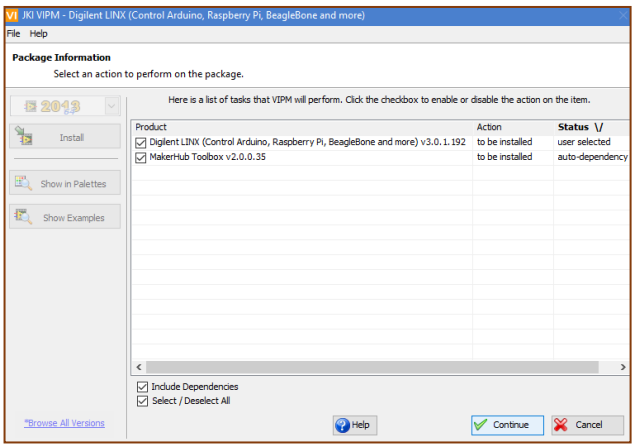


Figura 74 Proceso de instalación de LINX

Al finalizarse el proceso se deberá contar con los Toolkits que se muestran a continuación.

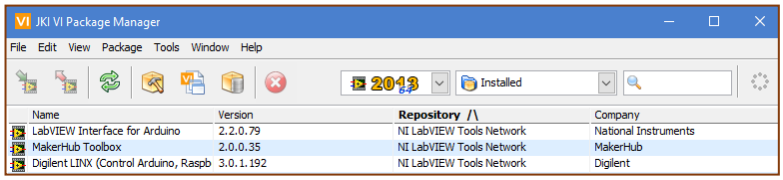


Figura 75 Toolkits a utilizar

Los otros dos Toolkits necesarios para ejecutar los VI a utilizar para el desarrollo de estas prácticas son:

- Control Design and Simulation
- Math script RT advance

4.1.4 Programar el Arduino

Una vez se haya realizado la instalación de las herramientas en LabVIEW quedará un paso más antes de poder empezar a trabajar, y ese es la programación del Arduino para que este pueda entender los comandos de LabVIEW, recibir y enviar información utilizando el Toolkit de LabVIEW Arduino; Para realizar esto se deberá tener instalado el software de Arduino, para poder abrir el código que posteriormente se va a cargar en el Arduino, este se encuentra en una carpeta llamada LIFA base.

Figura 76 Carpeta librería LabVIEW/Arduino

Esta carpeta se encuentra por defecto en la carpeta del software de LabVIEW Arduino en el disco local c y en la carpeta de National Instruments, pero para facilitar el proceso de búsqueda, se generó una copia de esta en la carpeta del proyecto para poder acceder al archivo con mayor facilidad.

Dentro de esta carpeta se encuentran los siguientes archivos

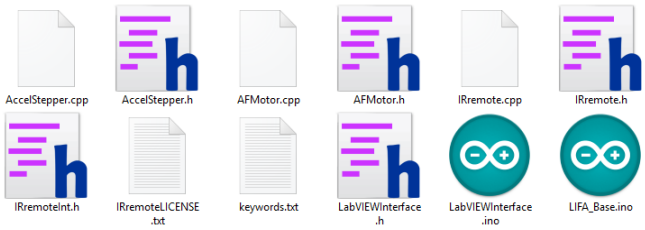


Figura 77 Archivos dentro de la carpeta LIFA_Base

De todos estos archivos se deberá abrir el archivo de nombreLIFA_Base.ino, el cual deberá de ser cargado al Arduino que se fuera a utilizar para que este pueda recibir los comandos que vengan de LabVIEW y habiéndolo subido ya podrá ser usado el Arduino para realizar la conexión con LabVIEW y realizar programas de control, entre otras tantas posibilidades que brinda este Toolkit.
Nota: En el caso de los VI que utilizan LINX el proceso se realiza al inicio de cada uno accediendo al Wizard de Maker hub desde la pestaña de Tools.

4.1.5 Herramientas del Toolkit LabVIEW-Arduino

El conjunto de herramientas que se encuentran en este Toolkit podrá ser utilizadas para la realización de programas varios en los que, utilizando LabVIEW como maestro y Arduino como esclavo, o visto de otra manera, LabVIEW como el programa y Arduino como el ejecutor de dicho programa, se realice el proceso de control o toma de muestras entre otras tantas posibilidades.

Estas herramientas cuentan con bloques de programa especializados para muestreo, otros para control y otros más para sensores dedicados, junto con varios bloques que llevan a varios VI de ejemplo sobre el uso de estos.
Se empezará por un vistazo general a estas herramientas.

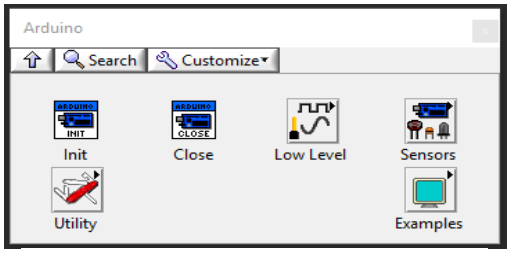


Figura 78 Herramientas de Toolkit Arduino

A primera vista se ve la pantalla de ejemplos, la de utilidades, la de sensores de bajo nivel.

Al colocar el cursor sobre la casilla de nombre "Exámenes".

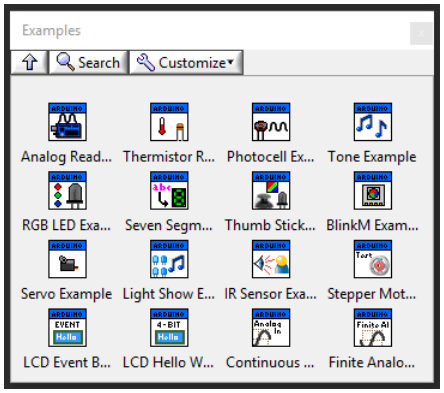


Figura 79 Ejemplos del Toolkit Arduino

En esta ventanilla se puede apreciar una variedad de ejemplos que el Toolkit ofrece al usuario para poder explorar y conocer el funcionamiento de las herramientas disponibles, así como a su vez realizar pruebas rápidas de diferentes elementos como por ejemplo un Led RGB.

Al poner el cursor sobre la casilla de nombre "Low Level".

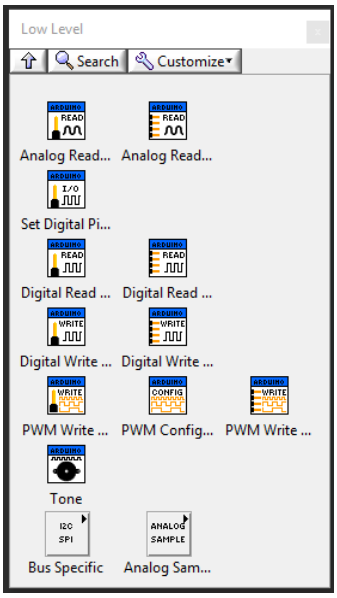


Figura 80 Elementos dentro de Low Level

En esta pestaña se puede apreciar los bloques de lectura y escritura digital y analógica respectivamente, tanto individual como por puertos, la escritura analógica como tal se da utilizando el PWM que está limitado a pines específicos del Arduino, y de forma básica trabaja a 500Hz.

También se puede apreciar un bloque de programa llamado Tone y los de Bus Specific y Analog Sample

Al poner el cursor sobre la casilla de nombre "Bus Specific y Analog Sample".

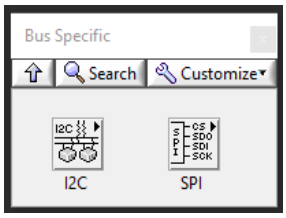


Figura 81 Elementos dentro de Bus Specific

Dentro de Bus Specific se encuentran las herramientas para realizar comunicación I2C y SPI

Al poner el cursor sobre la casilla de nombre "Sensors"

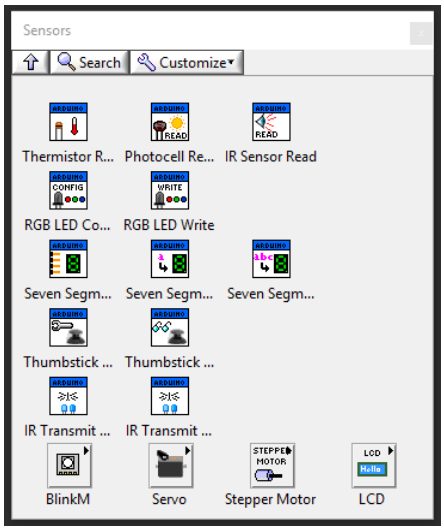


Figura 82 Elementos dentro de la pestaña Sensors

Como puede apreciarse, en esta sección se encuentran todas las herramientas dedicadas a los diferentes elementos contemplados en la librería, para facilitar el uso de estos en los diferentes programas que se fueran a realizar, a su vez también tiene cuatro pestañas más para unos elementos que requieren cierto trabajo extra en su implementación dentro del código.

La casilla de BlinkM tiene las herramientas necesarias para utilizar un BlinkM a continuación se realizará una visualización de cuáles son estas herramientas.

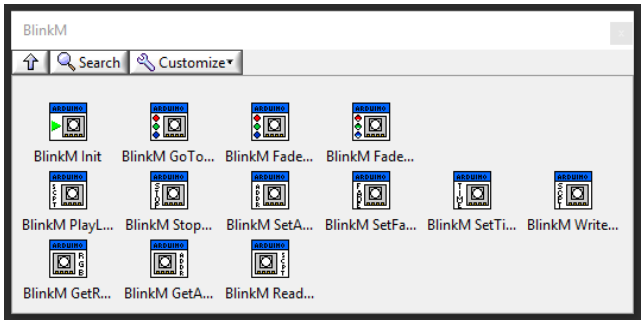


Figura 83 Elementos dentro la pestaña BlinkM

En el caso de la casilla de Servo se tiene las siguientes herramientas.

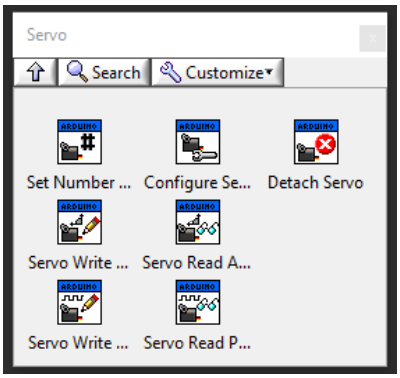


Figura 84 Elementos dentro de la pestaña Servo

Dentro de los bloques de programa se tiene la posibilidad de configurar desde el número de servomotores hasta la de desconectarlo, leer su posición, entre otras.

De forma similar al servomotor, se tiene las herramientas para poder utilizar un motor a pasos.

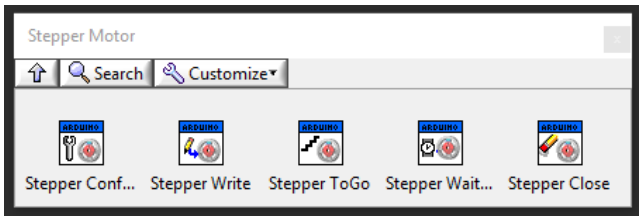


Figura 85 Elementos dentro de la pestaña Stepper Motor

En el caso del Stepper Motor se tiene considerado el uso de un Shield de Arduino para su mejor manejo.

Como conjunto de bloques de programa dedicados se tiene los del LCD.

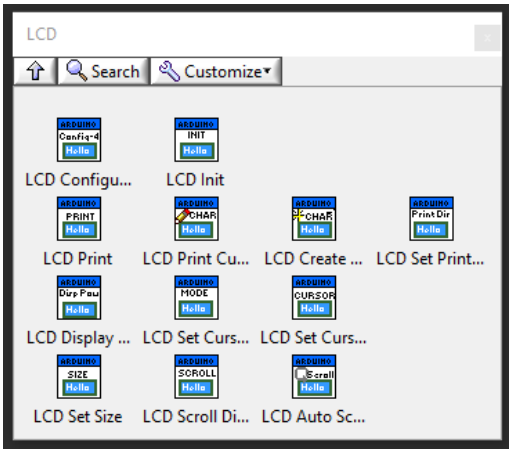


Figura 86 Elementos dentro de la pestaña LCD

Finalmente, dentro de lo que a las herramientas del Block Diagram, se tiene la pestaña de "Utility".

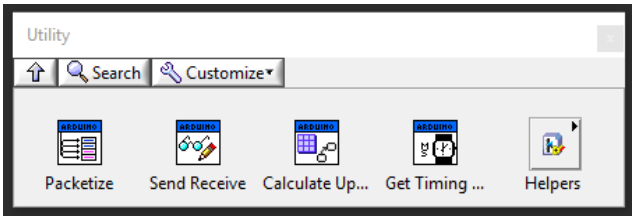


Figura 87 Elementos dentro de la pestaña Utility

Esta parte final contempla las opciones pensadas para programas más complejos que requieran procesos de análisis de tiempo, envío de paquetes entre otras opciones, pero que por lo general en programas pequeños suelen no ser usados, ya que estos mismos se encuentran implementados en el código de los anteriores bloques de programa revisados en esta práctica.
Las cuales permiten tener una interacción con los valores descritos por cada una en los diferentes programas que sean realizados.

4.1.6 Estructura de un programa

Para utilizar este conjunto de herramientas es importante saber que estas requieren una estructura básica para su correcto funcionamiento.

- Se empezará por la revisión dentro de la casilla de ejemplos de algunos de los ejemplos que esta contiene.
- Se realizará un análisis a su estructura.

- Luego será realizada una comparación entre las diferentes estructuras.
- Finalmente deberán de ser identificadas sus similitudes.

Se realizará la revisión de cada uno por separado para posteriormente ver en qué se parecen y en qué se diferencian, que elementos tienen y donde están estos ubicados además de cómo están conectados entre ellos.

4.1.7 Herramienta del Toolkit LINX

Para acceder a las herramientas del Toolkit de LINX se deberá dar ir a la pestaña de MakerHub la cual se encuentra junto con los demás bloques de programación de LabVIEW, dando clic derecho en la ventana de "Block Diagram" del VI.

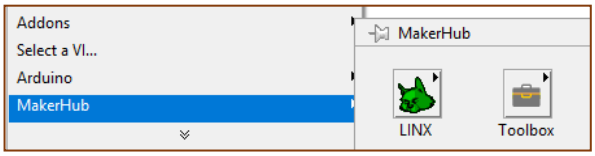


Figura 88 Pasos para acceder a Toolkit LINX

Estas herramientas se encuentran divididas en dos secciones, la primera corresponde a las herramientas que serán utilizadas para elaborar los programas a nivel básico, similar a como ocurría en el Toolkit de LabVIEW-Arduino, y la segunda a las herramientas adicionales para programas más complejos las cuales engloban bloques de programa para ruteo de paquetes

Dentro de la casilla de "Peripherals" se encuentran las herramientas correspondientes a todas las señales de entrada y salida referentes a los pines y tipos de comunicaciones del Arduino, entre estos de manera especial serán utilizados los de "Analog, Digital y PWM" dada su gran utilidad en los procesos de control que serán realizados en prácticas de más adelante.

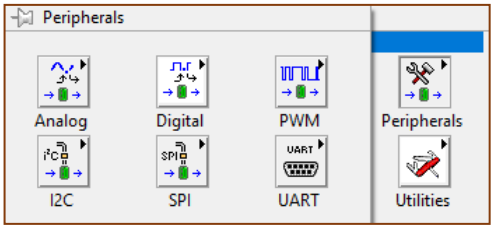


Figura 89 Pasos para acceder a la herramienta Analog

En "Analog" se encuentran las opciones para escritura y lectura de los pines analógicos del Arduino, así como la selección de un voltaje de referencia diferente.

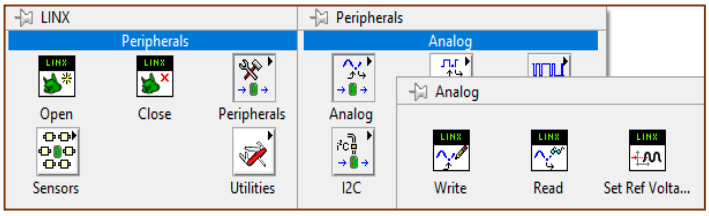


Figura 90 Opciones dentro de la herramienta Analog

Su contraparte en "Digital" contiene los bloques de programa para todos los pines digitales del Arduino y estos mismos al estar ubicados, permiten la configuración por puertos o conjuntos de estos, así como el calcular el ancho de pulso de una señal y enviar una señal cuadrada.

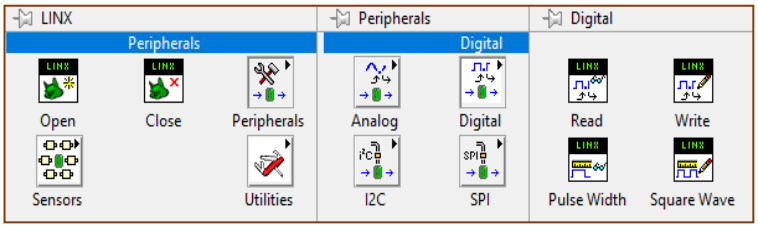


Figura 91 Opciones dentro de la herramienta Digital

Dentro de PWM se tienen las opciones para la escritura de una señal PWM, la cual es comúnmente utilizada para los procesos de control digitales, está controlada utilizando el porcentaje del ciclo positivo a utilizar como variable, esta variación está dada en un rango de 0 a 1 siendo 1 el 100% y 0 el 0%.

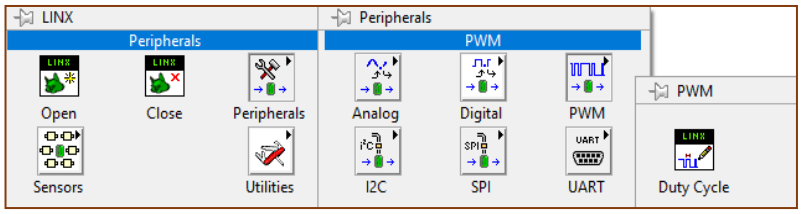


Figura 92 Opciones dentro de la herramienta PWM

En la casilla de I2C, SPI y UART se encuentran los bloques de programa necesarios para la codificación, configuración y uso de estas comunicaciones con las opciones que se muestran a continuación.

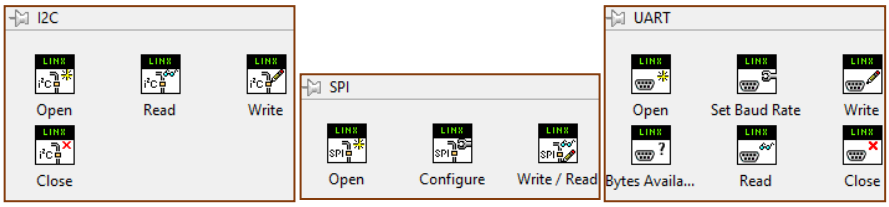


Figura 93 Herramientas dentro de las casillas I2C SPI UART

En el bloque de "Sensors" se ubican todos los bloques de programa del Toolkit que interaccionan con sensores conocidos de manera más sencilla dado que ya vienen configurados para el uso de estos, un ejemplo de esto es el sensor ultrasónico ubicado en la casilla de "Distance".

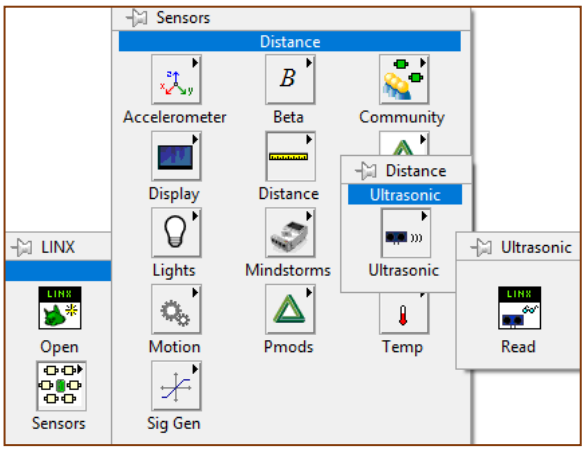


Figura 94 Ubicación del sensor Ultrasónico

Por último, en esta sección se encuentra dentro de la casilla de "Utilities" varios bloques de programa dedicados a labores de personalización y configuraciones.

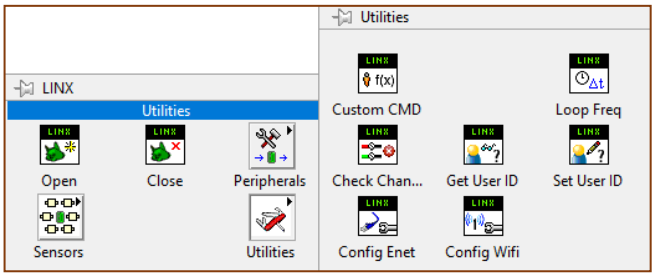


Figura 95 Herramientas dentro de la pestaña Utilies

4.1.8 Estructura de un programa

En el caso de LINX se deberán seguir las mismas mecánicas y estructuras aprendidas previamente con el Toolkit de LabVIEW-Arduino, dado que este comparte los mismos requerimientos.

Cabe destacar que este a diferencia del Toolkit de LabVIEW-Arduino, no cuenta con una sección de ejemplos pero que no hace tanta falta dado que cuenta con una descripción de lo que hacen sus elementos al seleccionar la ayuda en LabVIEW.

No hay manera de decir que nomás se podría considerar elemental como conocimiento para teoría de control, pero si se puede estimar o realizar un acercamiento personal a dichos conocimientos, a continuación, un listado con los conocimientos considerados necesarios por los autores de este proyecto de grado para el uso de este.

- Modelado matemático de sistemas físicos.
- Modelado matemático de sistemas electrónicos.
- Criterios de estabilidad de Routh-Hurwitz.
- Tipos de controladores P - PI – PID.
- Sintonización de controles P-PI-PID.
- Compensadores en Adelanto y Atraso.
- Sistemas no lineales.
- Control PID discreto.

Estos temas serán tratados a lo largo de las diversas prácticas, pero es importante el invertir un poco de tiempo obteniendo un conocimiento más extenso de dichos temas y tener varias fuentes para consultar y consolidar dicho conocimiento.

4.2 Práctica #2 “Modelado de sistemas”

4.2.1 Objetivos

- Visualizar y entender el funcionamiento del VI.
- Comprender el proceso de muestreo.
- Realizar la identificación de una planta.

4.2.2 VI de Modelado

Se empezará con una vista general de lo que se encontrará en el VI de Modelado, en él se podrá usar una de las entradas analógicas del Arduino para realizar la lectura de valores de una planta, dicha planta a su vez podrá ser alimentada por el Arduino o en su defecto ser comandada por el mismo.

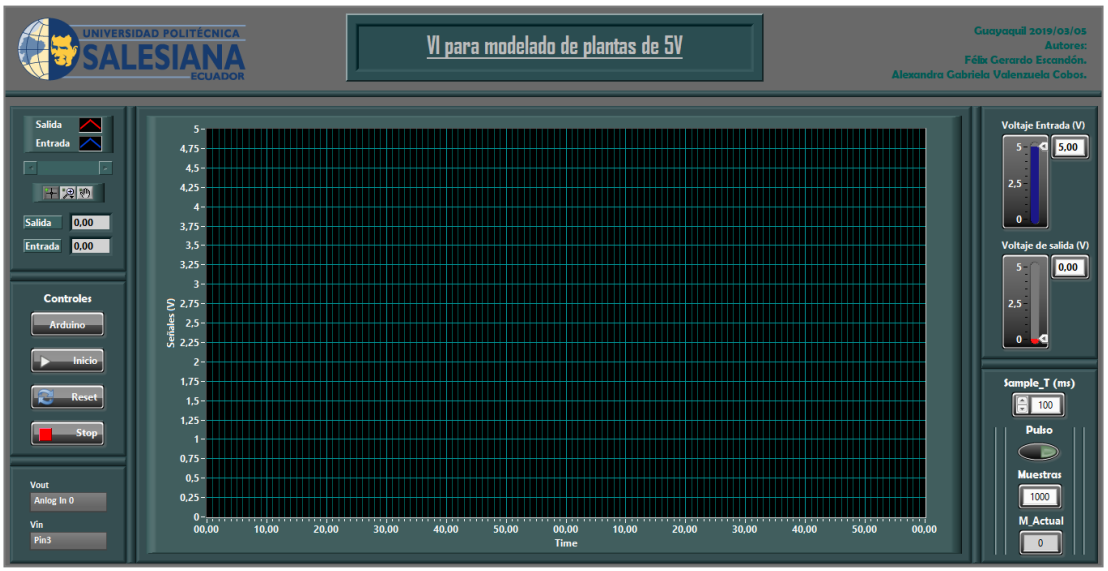


Figura 96 Front Panel del VI Modelado

Este VI consta principalmente de una pantalla central la cual esta calibrada para que el valor de voltaje sea de -1V a 6V, puesto que el Arduino puede realizar un rango de medición analógica de 0 a 5V, por otro lado, la barra de tiempo esta puesta para mostrar los minutos y segundos transcurridos, para facilitar la obtención de valores de tiempo de los gráficos dispuestos en la pantalla.

Luego se puede apreciar que hay un conjunto de controles e indicadores tanto a derecha como a izquierda de la pantalla, A continuación, se realizará un acercamiento a los controles que se encuentran a la izquierda de la pantalla y se indicara la función que cumple cada uno en el proceso de muestreo y modelado de plantas.

Figura 97 Visualización de valores de entrada y salida del VI

Esta sección del VI es meramente informativa, con la pequeña adición de una barra de herramientas para manipular la pantalla hacer acercamientos entre otras cosas, también permite ver el valor actual de la variable medida y de la variable controlada.

Figura 98 Controles del VI

Aquí se encuentran los controles que gobiernan el conjunto de procesos internos del VI.

El primer botón permite alternar entre el uso del Arduino o un modo simulado en el cual el usuario utilizando los controles ubicados a la derecha, dibuja la señal.

El segundo botón da inicio al proceso de muestreo y subsecuentes procesos que culminan en la generación de la función de transferencia de la planta.

El tercer botón es usado para guardar los datos en el archivo Datos.mat que contendrá los valores enviados al escritorio de Matlab y los modificados a lo largo del proceso, este botón se deberá presionar luego de haber realizado el modelado y se hayan enviado al escritorio la o las funciones de transferencia de la planta, cerrando el ciclo del proceso y permitiendo volver a utilizar el botón de inicio para volver a realizar todo el proceso nuevamente.

Finalmente, el cuarto botón permite finalizar el proceso en cualquier momento, pero por asuntos de diseño es necesario dar clic al botón de reste para terminar de cerrar el proceso totalmente.

Figura 99 Selección de pines a usa

Estos dos menús permiten seleccionar los pines de entrada analógica y salida digital que se fueran a utilizar, estos están nombrados en referencia a la señal que representan.

Figura 100 Control de Vin y Vout a modo simulado

Estos controles permiten manipular la señal que se va a registrar cuando se realice el modelado en modo simulado.

El primero comanda el valor simulado de entrada de la planta y el segundo el valor simulado de lo que sería leído por el Arduino como salida de la planta.

Figura 101 Controladores e Indicadores a usar en la parte simulada

En esta sección se encuentran varios controles y un indicador, los cuales son importantes tanto para el proceso simulado como al usar una planta real.
El primer control permite poner el tiempo de muestreo (En milisegundos).

El segundo es un botón cuyo valor positivo equivale a 5v y el negativo equivale a 0v puesto que se está utilizando la salida del Arduino como salida digital y no como PWM.

El tercer control permite ubicar el número de muestras que se van a realizar, mientras que el cuarto indica en cual número de muestra se encuentra el proceso, lo cual es principalmente útil para tener una noción del número de muestras que quedan por ser tomadas.

4.2.3 Secuencia del proceso

Se empezará realizando las conexiones que va a utilizar en el Arduino y la planta, tanto de sensores como de pines de lectura y control, de igual manera se deberán de seleccionar los pines a utilizar en el VI.

Acto siguiente se deberán configurar el tiempo de muestreo, numero de muestras y el estado de inicio del botón de pulso, luego de esto se dará clic al botón de inicio, para empezar a realizar el muestreo, una vez finalizado el proceso habrá que esperar un momento a que se despliegue la ventana de "ident" de Matlab, luego se generará la función de transferencia y esta deberá ser enviada al escritorio, finalmente se dará clic al reste para guardar los datos.

Los pines digitales del Arduino están configurados para iniciar todos como salida y energizados, por esta razón suele darse el caso de que al realizarse la conexión por primera vez con el Arduino sin haberse configurado el mismo previamente, suceda que la planta se encienda al máximo de su capacidad, para evitar que esto de problemas, se puede utilizar el “VI de Practica” para preconfigurarlo o en su defecto hacer un primer proceso de muestreo muy pequeño configurando el botón de pulso como apagado para asegurar que empiece así al volver a iniciar el proceso nuevamente.

De querer realizar medición de una planta desconectada por completo del sistema de control, es importante coordinar que el encendido sea realizado al mismo tiempo que se da inicio al proceso de muestreo teniendo encendido el botón de pulso para poder tener una sincronía entre las barras de tiempo y la señal de alimentación.

4.2.4 Para Practicar

Deberá de realizarse el siguiente circuito en un Protoboard, los materiales que se necesitaran son los siguientes:

- 1 Protoboard.
- 1 potenciómetro de 5kΩ.
- 1 capacitor de 1000µF o similar.
- cables para conectar.
- 1 Arduino uno, junto con su cable USB.

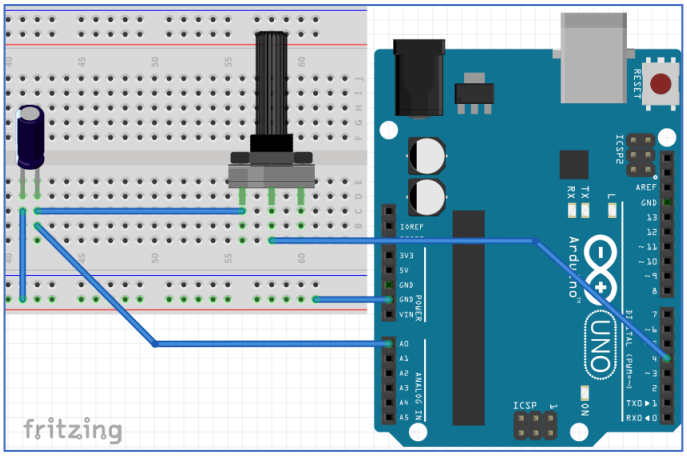


Figura 102 Esquema del circuito

El VI de modelado y lo vamos a configurar según el tipo de conexión que fuéremos a usar; luego, con un Arduino configurado con la librería como se muestra en la práctica #1, realizamos la conexión USB con la computadora, y empezamos a correr el VI.

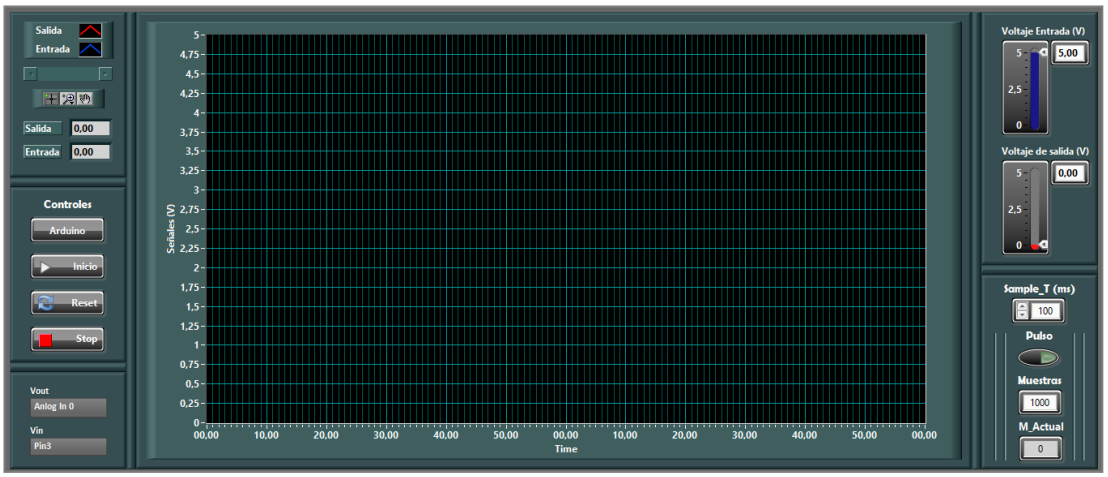


Figura 103 Paso 1 de la práctica

Luego de dar inicio, se puede apreciar que se obtiene un resultado como este, y que luego de un tiempo aparecerá la ventana de ident.

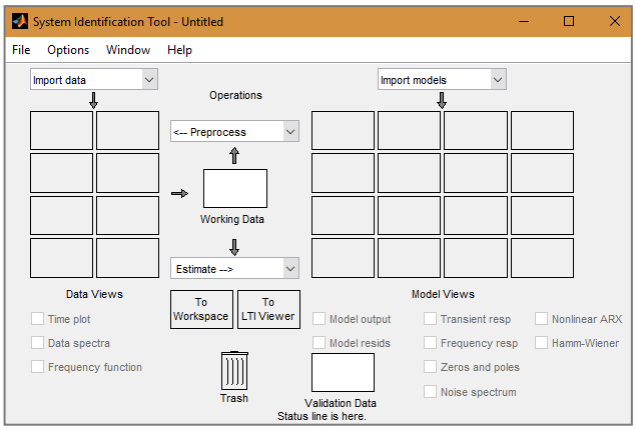


Figura 104 Paso 2 de la práctica

Pues bien, ahora se deberá minimizar la pantalla del ident, dar clic en el botón de reste, y en el circuito se deberá bajar la resistencia del potenciómetro al mínimo, luego de esto se volverá a dar clic al botón de inicio.

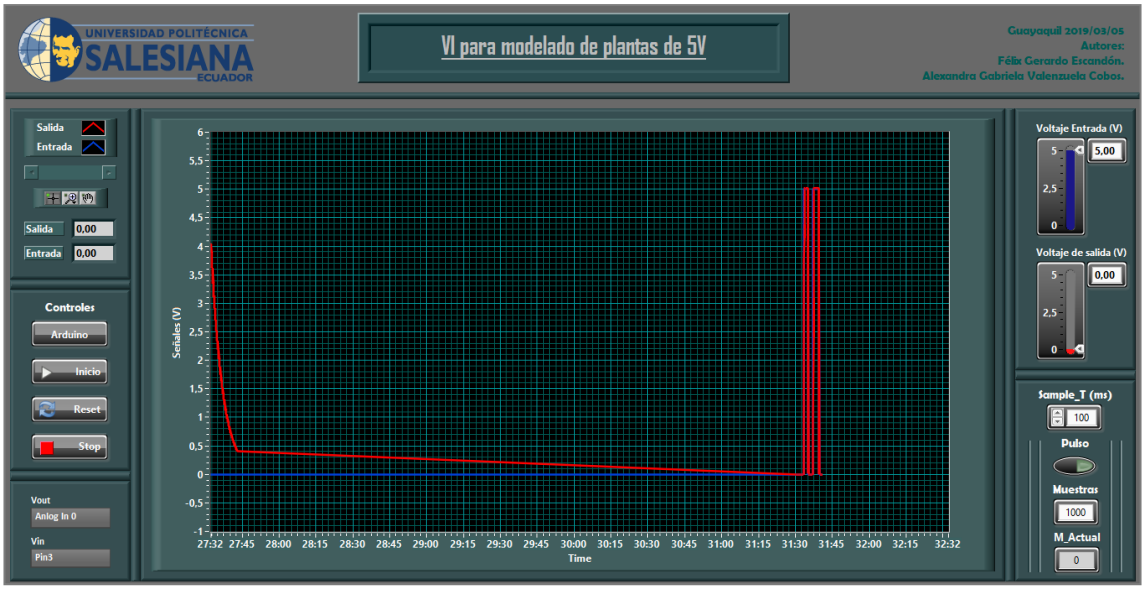


Figura 105 Paso 3 de la práctica

Posiblemente se obtenga un resultado similar al mostrado, para solucionar esto se realizará un acercamiento horizontal utilizando las herramientas de edición de la esquina superior izquierda y así mejorar la visualización de los valores medidos.

Figura 106 Herramienta de visualización

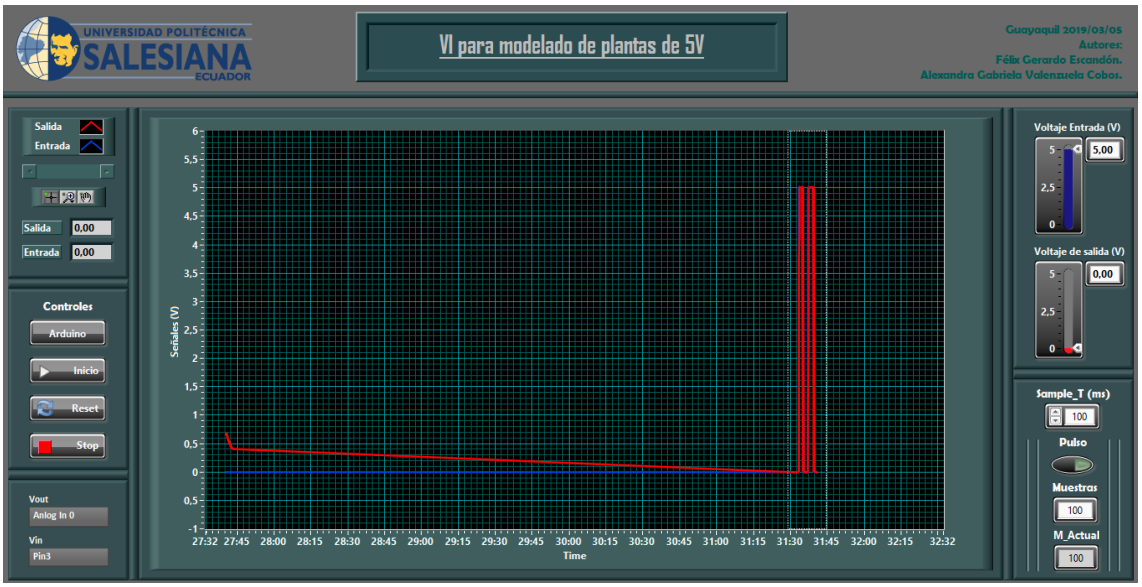


Figura 107 Visualización mejorada de la práctica

De esta manera se conseguirá tener una visualización más optima de los datos y se podrá ver las pequeñas perturbaciones en la señal, una de las causas de esto es debido a que el sistema carece casi por completo de resistencia y el capacitor puede descargarse y cargarse demasiado rápido, lo cual genera una perdida rápida del voltaje recibido.

Cuando se usa un PWM esta distorsión es mucho más notoria, pero como se está usando una salida digital, esto casi no puede percibirse.

Bien, ahora se deberá aumentar ligeramente la resistencia y analizar cómo cambia la señal de salida.

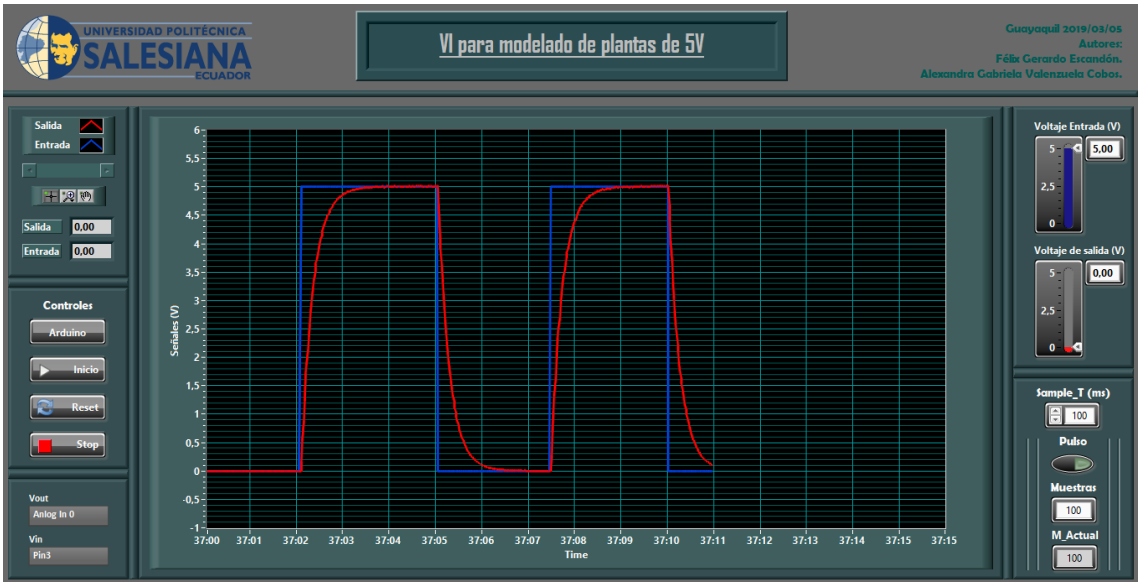


Figura 108 Comportamiento de un circuito RC

Esta respuesta ya es más parecida a la utilizada generalmente para ejemplificar el comportamiento de un circuito RC; Bien, ahora se deberá realizar la generación de la función de transferencia de esta planta.

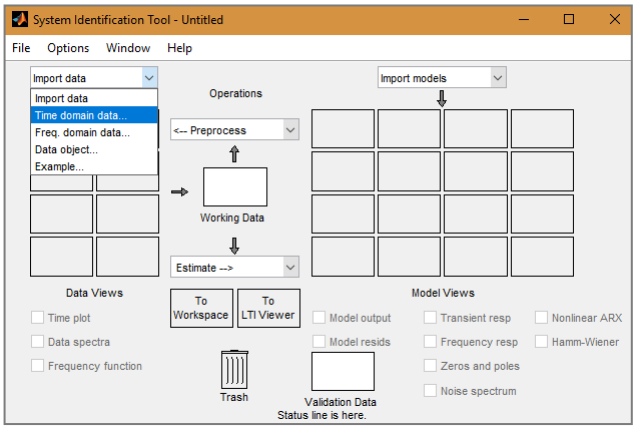


Figura 109 Identificación de la planta

Deberá de abrir la ventana de ident en la barra de tareas de la computadora, dar clic en la pestaña de nombre “Import data”, y posterior a esto seleccionar “Time domain data”.

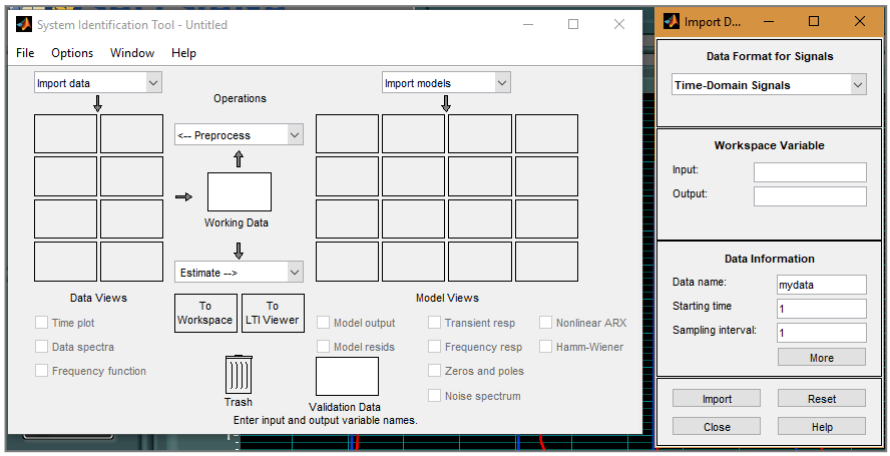


Figura 110 Colocación del nombre de Entrada y Salida

Esto abrirá una nueva ventana, en la cual se deberán ingresar los datos que se muestran a continuación, hay que tener en cuenta que donde dice “Sampling interval” aunque no se especifique, la unidad de tiempo utilizada por ident está dada en segundos, por lo cual siendo que el proceso de muestreo fue cada 100ms, deberá de ubicarse 0.1 para que este coincida con la unidad de tiempo utilizada en ident.

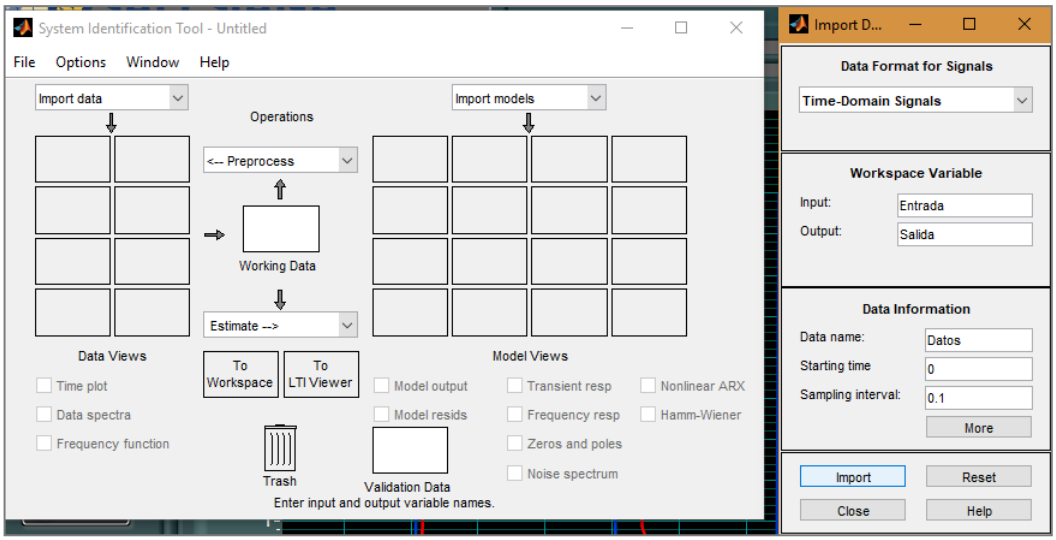


Figura 111 Colocación del intervalo del tiempo de muestreo

El archivo Datos.mat es el archivo en el que se guardan los datos que se van generando en el VI luego de presionar el botón de reste, este el archivo puede encontrarse en una carpeta del proyecto al igual que una serie de parejas de archivos adicionales con los valores de las señales de entrada y salida, en el archivo Datos.mat tiene el nombre de Entrada y Salida respectivamente.

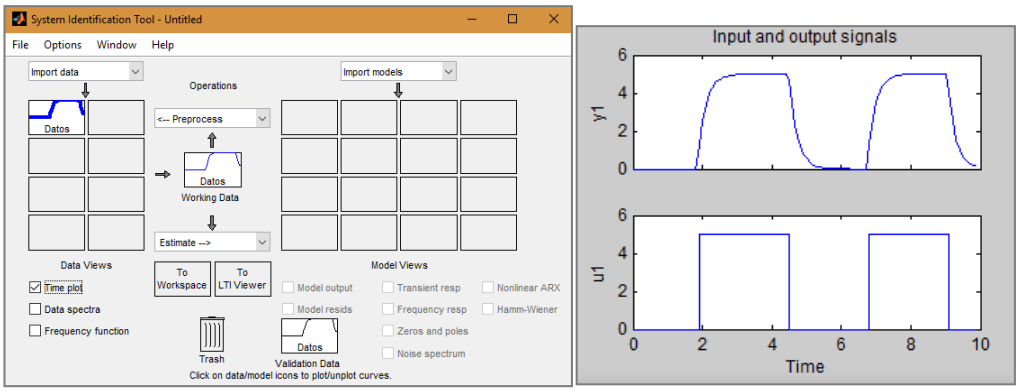


Figura 112 Obtención de las muestras

Una vez se hayan ingresado dichos valores se podrá apreciar que se importa a ident la misma señal de entrada y salida ingresada del VI, y se puede apreciar que como se tomaron 100 muestras a 100ms, se tiene un total de 10s de señal con la cual se puede empezar a trabajar en la modelación de una función de transferencia.

Para esto se dará clic en la pestaña de nombre "Estimate" y luego de esto se elegirá la opción para función de transferencia, cabe recalcar que existen otras opciones como se puede apreciar en la imagen a continuación, pero que, para efecto de esta práctica, solo será utilizada la primera opción.

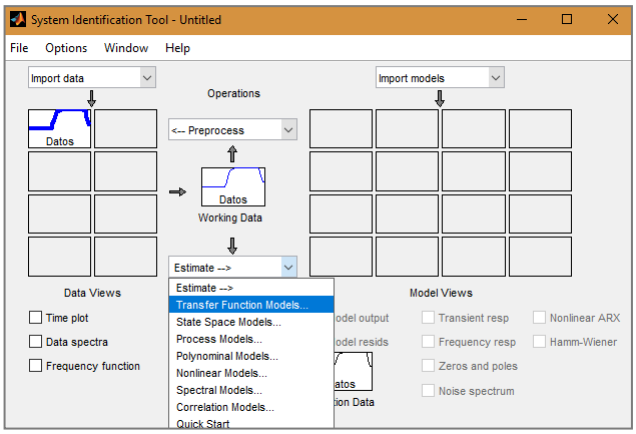


Figura 113 Obtención de un estimado de la función de transferencia

Al dar clic en esta opción, aparecerá otra pestaña más, en esta ventana se podrá configurar el número de polos y de ceros que se desea que tenga la función de transferencia a generar, junto con la opción de realizarla en tiempo continuo o en tiempo discreto, junto con otras dos opciones más de configuración avanzada, que se anima al estudiante a revisar, investigar y experimentar, de momento se generará el modelado en base al modelo matemático de un circuito RC el cual tiene 1 polo y 0 ceros.

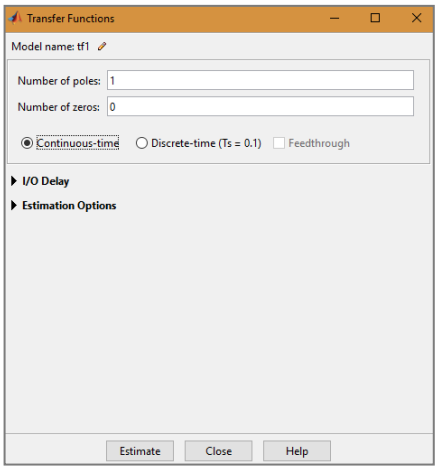


Figura 114 Colocación de numero de polos y ceros de la función de transferencia

Y listo, ident genera de manera automática una función de transferencia según las configuraciones dadas, y dando clic a uno de los cuadros se puede desplegar ventanas de información respecto de la señal que tengamos seleccionada, como se puede apreciar, se muestra la respuesta del modelo y su % de aproximación a la señal muestra, el cual es bastante alto, pero no perfecto.

Se puede dar doble clic en el modelo tf1 para desplegar sus características y ver la forma de la función de transferencia, pero también se puede mantener el clic para poder arrastrarlo y llevarlo al cuadro que dice "To workspace" , una vez que se realice eso va a mostrarse un mensaje en la parte baja de la

ventana a modo de confirmación de que fue enviado exitosamente, luego de esto se deberá dar clic al botón de reste para realizar el guardado de los datos. De manera adicional, también puede realizarse un guardado manual de los datos de ident desde la pestaña de "File".

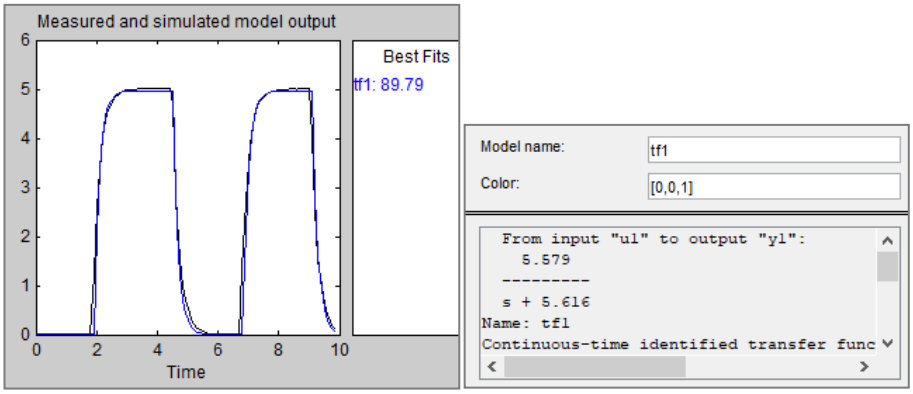


Figura 115 Obtención de la función de transferencia

Con esto se puede dar por terminado el proceso, es importante no olvidar dar click en el botón de reset para guardar la función de transferencia de la planta que se ha obtenido, en el archivo Datos.mat, de lo contrario no se podra usar esta información en los demas VI.

4.2.5 Modelado Planta térmica 2

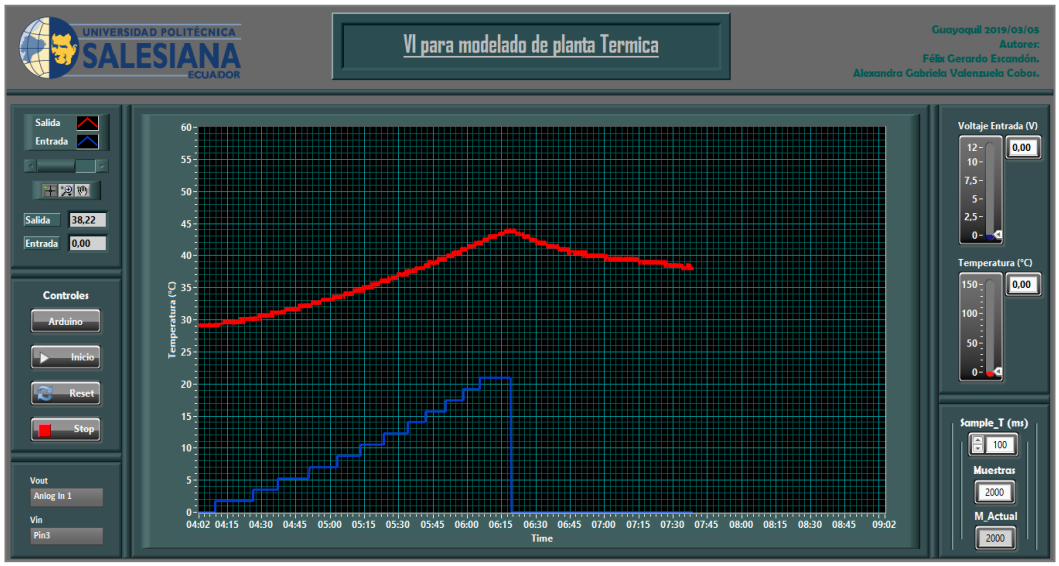


Figura 116 Muestreo de la Planta Térmica 2

Se realizo un total de tres muestreos de 2000 muestras a la "Planta Térmica 2" utilizando una luminaria, y se elaboró con estos diferentes funciones de transferencia en ident escogiendo tf10.

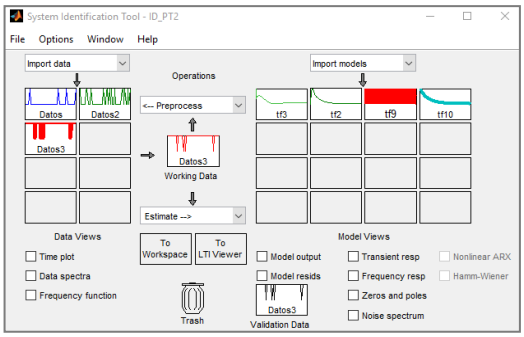


Figura 117 Ventana Ident

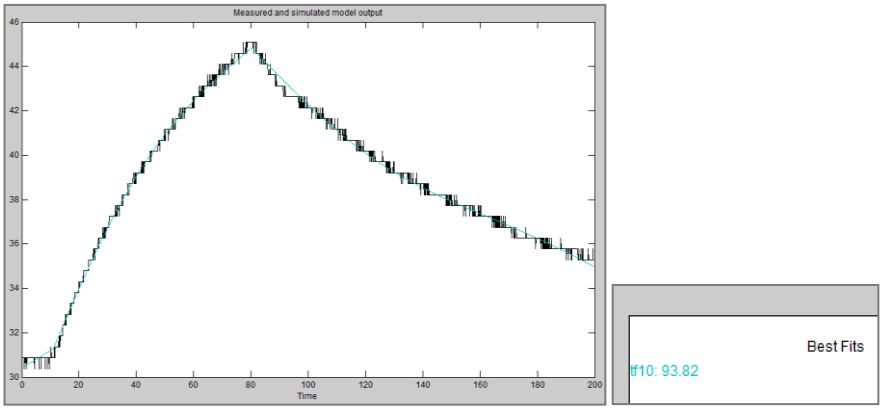


Figura 118 Conjunto de muestras con su proximidad

```
  From input "U" to output "Y":
     0.01135 s^2 + 0.000287 s + 2.504e-05
  ----------------------------------------
  s^3 + 0.02334 s^2 + 0.002111 s + 4.133e-06
Name: tf10
Continuous-time identified transfer function.
```

Figura 119 Función de transferencia "tf10"

4.2.6 Modelado Planta Levitadora 1

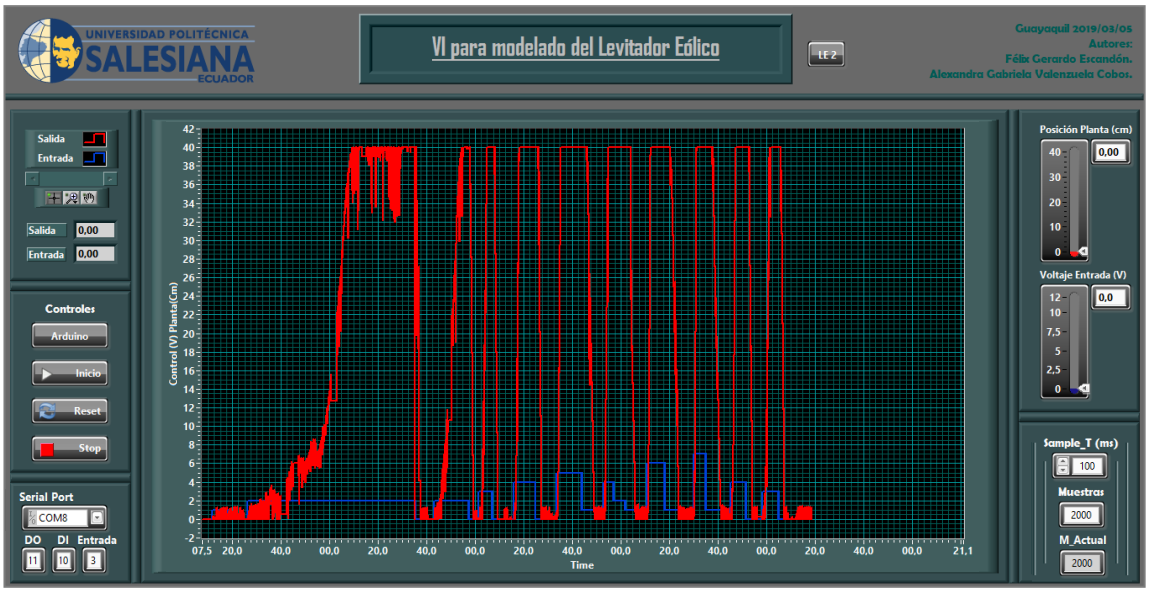


Figura 120 Muestreo Planta Levitadora 1

Las plantas "Planta Levitadora 1" y "Planta Levitadora 2" son sistemas no lineales, por ello se requiere tener un proceso diferente al momento de obtener un modelo de una de estas por ello también se da uso a sistemas no lineales, ya que los sistemas lineales limitan el comportamiento de estas plantas.

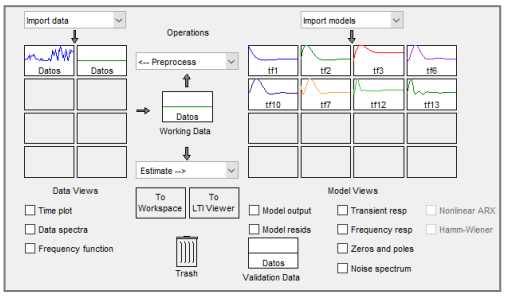


Figura 121 Ventana Ident

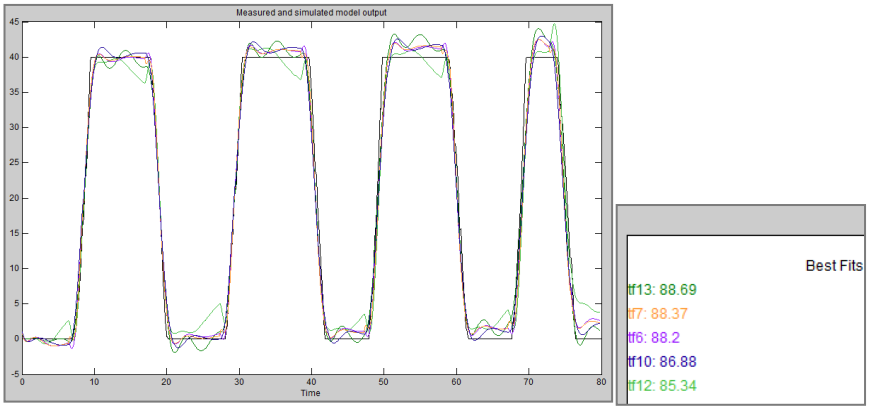


Figura 122 Conjunto de muestras con su proximidad

```
  From input "u1" to output "y1":
           -20.63 s^3 + 26.64 s^2 + 6.546 s + 0.3379
  ---------------------------------------------------------
  s^5 + 2.095 s^4 + 4.081 s^3 + 2.412 s^2 + 0.5252 s + 0.03136
Name: tf12
Continuous-time identified transfer function.
```

Figura 123 Función de transferencia "tf12"

4.2.7 Modelado Motor de Paso

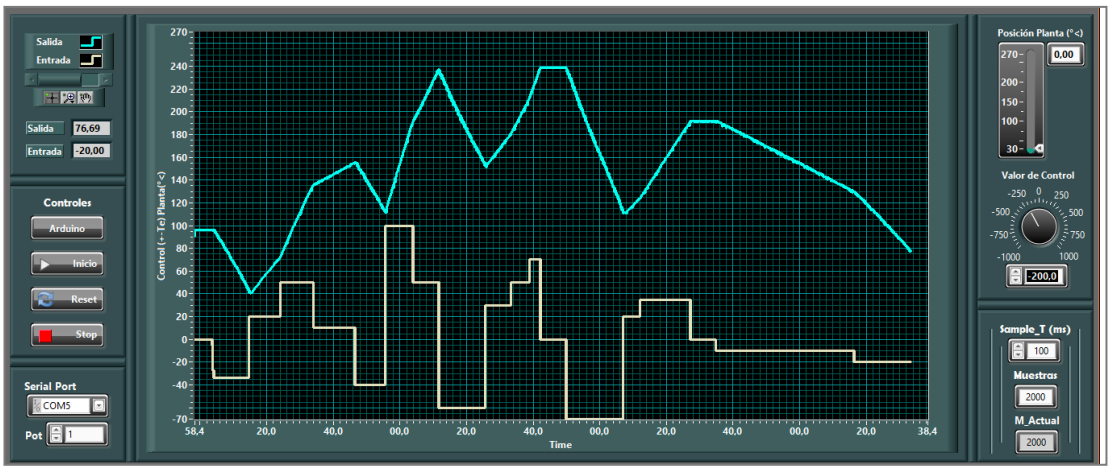


Figura 124 Muestreo Planta Motor de Paso

La planta Motor de paso debido a su diseño, se puede obtener una variación de velocidad en términos de tiempo de espera entre paso y paso, el cual es directamente proporcional al valor mostrado en el valor de control, siendo 1000 el equivalente a 0.01s de espera que es el valor mínimo de espera que tiene entre paso y paso; teniendo un comportamiento lineal.

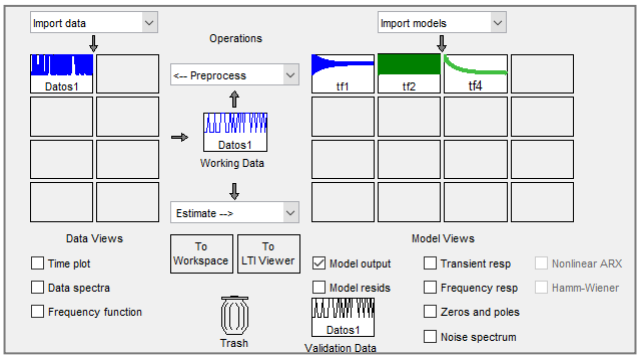


Figura 125 Ventana Ident

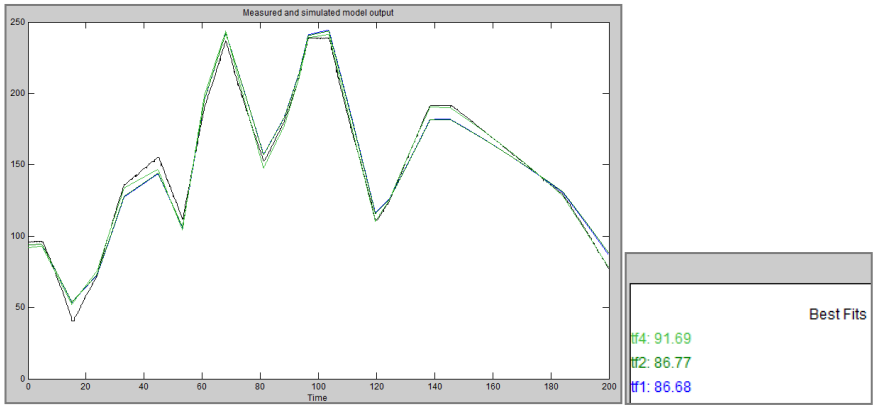


Figura 126 Conjunto de muestras con sus proximidades

```
From input "U" to output "Y":
   0.01258 s^2 + 5.231e-05 s + 1.299e-05
  ----------------------------------------
  s^3 + 0.007518 s^2 + 0.0007695 s + 5.643e-07
Name: tf4
Continuous-time identified transfer function.
```

Figura 127 Función de Transferencia "tf4"

4.3 Práctica #3 “Valores PID & Sisotool”

4.3.1 Objetivos

- Revisar las ventanas de Sisotool de Matlab.
- Visualizar las funciones del PID dado por Sisotool.
- Realizar los cálculos para obtener los valores de las constantes del PID que diseñemos.
- Comprobar la eficacia de nuestros cálculos con el VI de Valores de PID.
- Realizar un VI para la obtención de los valores PID usando la segunda forma en que se nos entrega la función del PID.

4.3.2 Sisotool

Sisotool es un conjunto de herramientas que combinan una amplia gama de gráficas, algoritmos de control, procesos de control, opciones de configuración, entre otras tantas herramientas usadas para los procesos de control, todo comprimido y espléndidamente distribuido y configurable haciendo uso de ventanas que se pueden abrir y cerrar.

Las partes de sisotool que serán repasadas van a ser aquellas usadas en las diversas prácticas que serán realizadas, pero se anima a la experimentación e indagación de aquellas que no se vean en esta práctica, dada su gran utilidad en diversos procesos de control; Se empezará realizando una visualización a la pantalla principal, tal como se muestra al utilizar el código "sisotool" en Matlab.

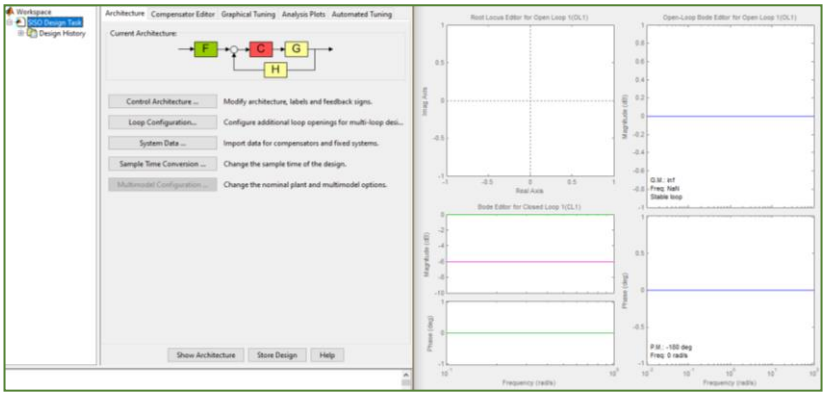


Figura 128 Ventana del comando Sisotool

Van a abrirse dos ventanas, la del lado izquierdo es la principal, mientras que la del lado derecho corresponde a una ventana que está relacionada con la pestaña de "Graphical Tuning"; a continuación, serán revisadas tres de las opciones que se tienen en la primera pestaña "Architecture" de la ventana principal.

Al presionar este botón, se va a abrir una ventana en al cual se puede ver las diversas arquitecturas que pueden ser usadas para desarrollar los procesos de control, así como cambiar diversas características de estos.

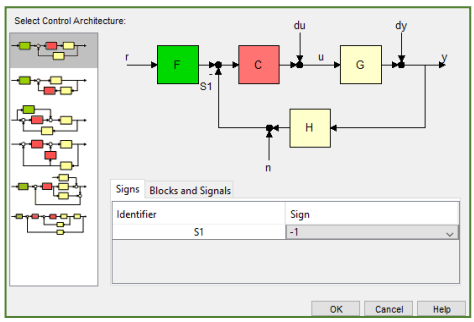


Figura 129 Ventana Architecture

El siguiente botón es el de "System Data" el cual da las opciones de seleccionar el sistema, el valor del compensador de realimentación, el controlador, y un compensador de la señal de entrada, o lo que es lo mismo, seleccionar las variables correspondientes a los bloques que se ven en la imagen anterior de la arquitectura de control utilizada.

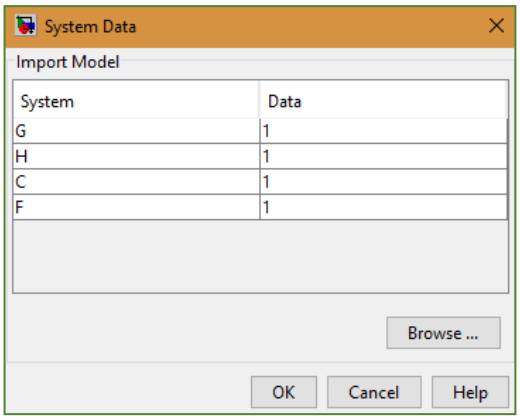


Figura 130 Ventana System Data

El tercero de los botones que se revisarán se trata del de "Sample Time Conversion" el cual es el encargado de contener las variables referentes al tiempo y es el que será utilizado cuando se quiera realizar un control discreto, ya que se puede modificar el tiempo de muestreo y seleccionar el tipo de método de conversión que se va a utilizar, por defecto son los que se muestran en pantalla.

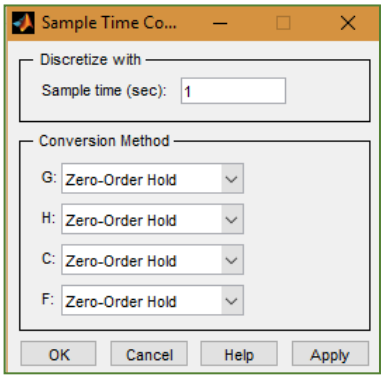


Figura 131 Ventana Sample Time Conversion

La segunda pestaña corresponde a "Compensator Editor" aquí se puede realizar el diseño de manera segmentada del controlador que se quiera realizar, ya que permite agregar integradores, diferenciadores, polos, ceros, además de compensadores en adelanto y atraso respectivamente, adicionalmente se muestra la forma que tiene el controlador.

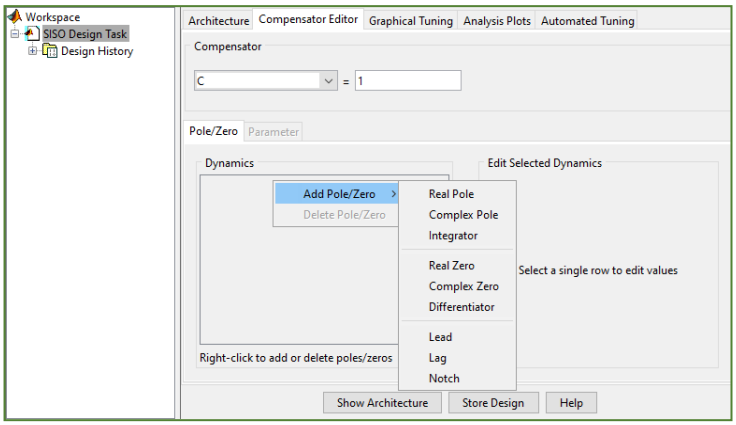


Figura 132 Ventana Compensator Editor

La tercera pestaña llamada "Graphical Tuning" brinda las herramientas para modificar la pantalla izquierda que se abre por defecto al utilizar sisotool.

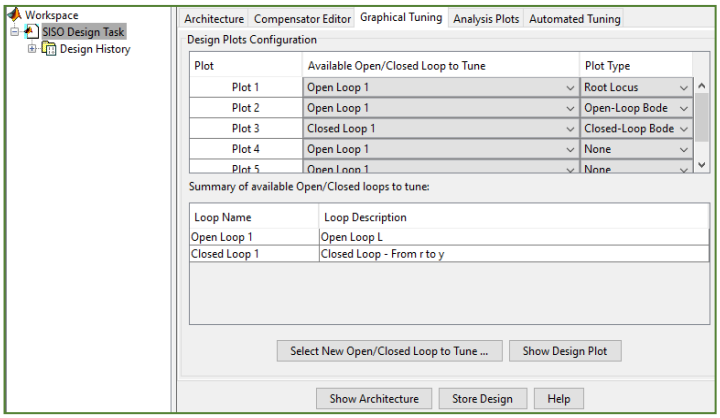


Figura 133 Ventana Graphical Tunning

En la cuarta pestaña se tiene el conjunto de herramientas de "Analysis Plots", estas herramientas brindan la posibilidad de abrir nuevas pestañas con los gráficos que se muestran en el listado seleccionando de manera específica cuales respuestas se quiere para dichos gráficos.

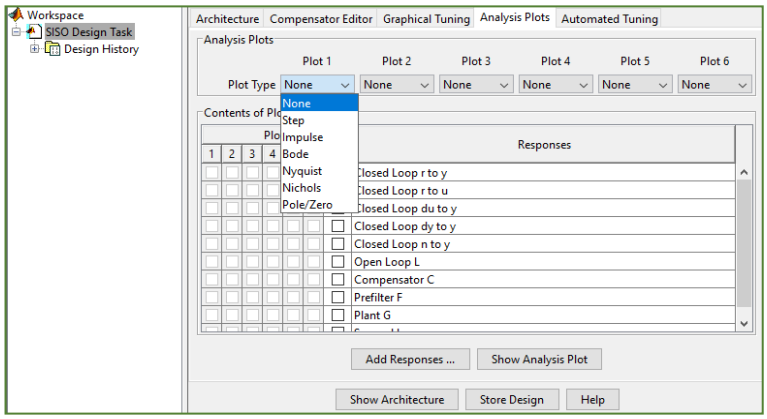


Figura 134 Ventana Analysis Plots

Finalmente se encuentra la pestaña de "Automated Tuning", en esta pestaña se puede seleccionar el tipo de control que se requiere y hacer la estimación de los valores de las constantes del control de manera automática.

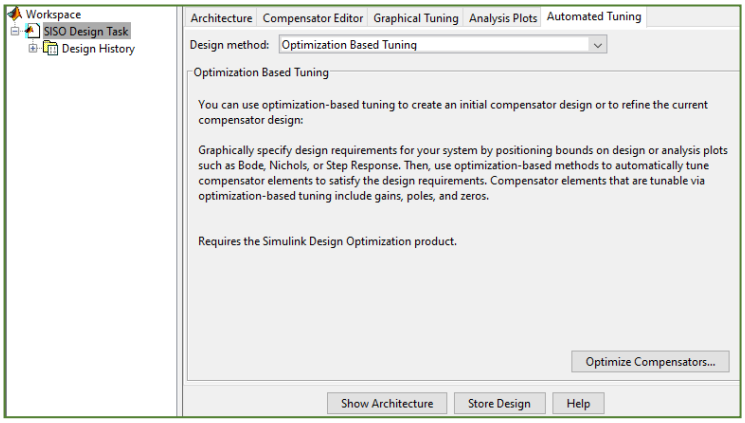


Figura 135 Ventana Automated Tunning

Se puede seleccionar entre las opciones que se muestran, pero para lo correspondiente a esta práctica se va a dar uso únicamente del "PID Tuning".

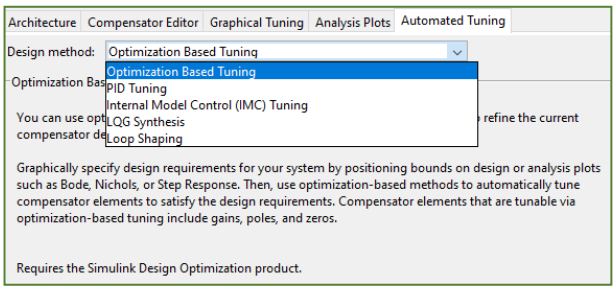


Figura 136 Opciones de métodos de diseño

Dentro de este existen varias opciones que van desde un control Proporcional hasta el PID con filtro derivativo, todos seleccionables y también se encuentra la opción de elegir entre una respuesta robusta o una adaptación con fórmulas clásicas, como puede verse también se tiene una caja de selección para poner o quitar el filtro derivativo del controlador, así como el cómo se desea que sea realizado el diseño del controlador, si de manera automática o manual, utilizando unas barras que aparecen al seleccionar este último.

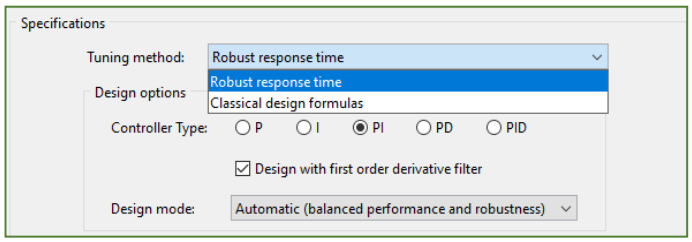


Figura 137 Especificaciones de los tipos de controlador

Al dar por primera vez en "Update Compensator" se generará el control seleccionado para la planta ingresada de la manera seleccionada, en este caso de manera automática con un balance entre la respuesta y la robustez del sistema y un control PID sin filtro derivativo, el cual se muestra en el recuadro de "Compensator" y también se muestra en la pestaña de "Compensator Editor" con la diferencia que en esa pestaña será posible modificar dichos valores; Otra forma de modificarlo es moviendo las polos y ceros de manera manual en la ventana de "Graphical Tuning" mientras se visualizan las variaciones en la gráfica de respuesta al paso de la pestaña de "Analysis Plots".

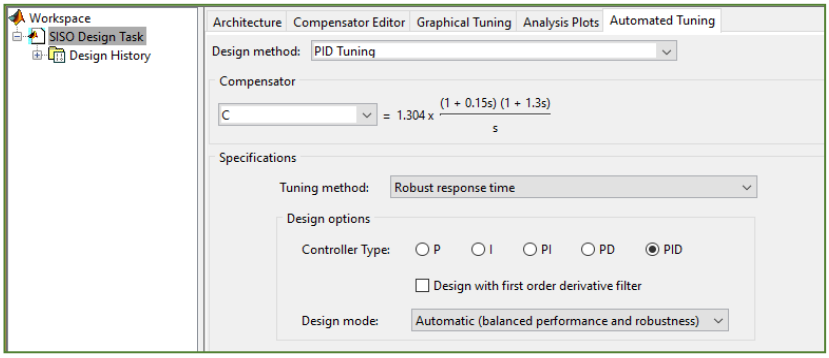


Figura 138 Función de transferencia

Cuando se tenga el controlador deseado, dicho controlador podrá ser exportado al escritorio de Matlab, para poder guardarlo en un archivo Datos.mat, y poder ser utilizado más tarde o en alguna otra aplicación u VI, esto desde File-Export.

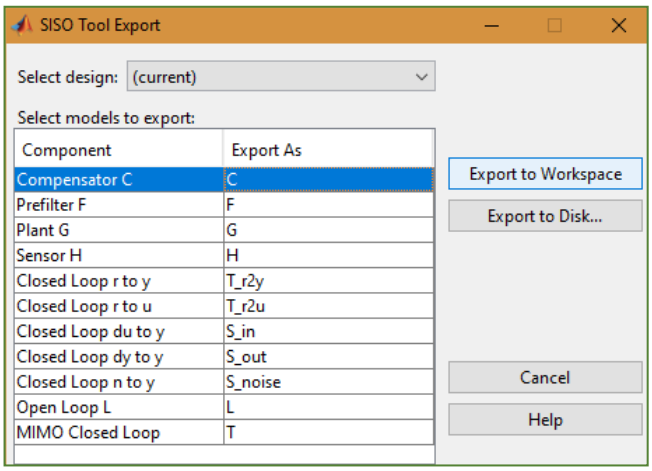


Figura 139 Exportación de datos

Adicional a esto también se podrá seleccionar la opción de "Store design" y este generará un conjunto de descripciones del diseño de control que se ha creado, junto con sus valores tal como se exporta al Workspace de Matlab.

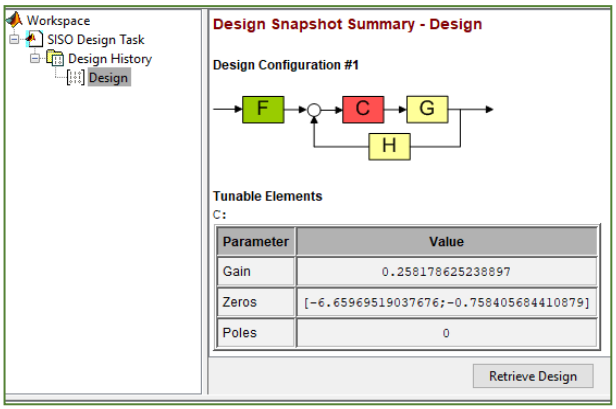


Figura 140 Valores que se exportara

Esta forma de expresar el controlador es una manera elaborada y tratada de manera matemática internamente por sisotool, mediante fórmulas matemáticas es posible llegar a obtener esta misma expresión en términos diferentes

4.3.2.1 Ecuaciones PID

Se deberá empezar por la formula general de PID

$$C(s) = K_p\left(1 + \frac{1}{T_i * s} + T_d * s\right) \tag{22}$$

$$C(s) = K_p + K_i * \frac{1}{s} + K_d * s \tag{23}$$

Y es importante tener en mente las relaciones que hay entre sus variables.

$$Kp \quad Ki = \frac{Kp}{T_i} \quad Kd = Kp * T_d \tag{24}$$

En sisotool, se presentan dos formas diferentes de representar al controlador, estas, para poder obtener el valor de las diferentes ganancias por individual conduce a la necesidad de llevar a las mismas a una forma más simple y menos mezclada de dichas variables y formula.

Una de las maneras de conseguir esto es a través de la siguiente forma de la ecuación:

$$C(s) = \frac{K_d * s^2 + K_p * s + K_i}{s} \tag{25}$$

Y partiendo de esta forma se puede tomar dos caminos diferentes para obtener dos ecuaciones expresadas en términos diferentes, pero que al final siguen conteniendo las mismas variables, se verá estas dos formas y cómo llegar a ellas por individual.

Forma 1 al multiplicar la ecuación por $\frac{T_d}{T_d}$ $\quad C(s) = K_p * T_d \frac{s^2+\frac{1}{T_d}*s+\frac{1}{T_d*T_i}}{s}$ (26)

que es presentada por Matlab de forma factorizada $C(s) = z\frac{(s+x)+(s+y)}{s}$ (27)

Forma 2 al multiplicar la ecuación por $\frac{T_i}{T_i}$ $\quad C(s) = \frac{K_p}{T_i}\frac{T_d*T_i*s^2+T_i*s+1}{s}$ (28)

que se presenta por sisotool de forma factorizada $C(s) = c\frac{(1+a*s)+(1+b*s)}{s}$ (29)

Como se puede apreciar, la manera en que es mostrada la función no da mucha información sobre cual variable es cual, pero de procesarla un poco, se llega a dos formas que permiten partir ya sea desde Ti o Td para encontrar el resto de los valores.

4.3.3 VI Valores de PID

Figura 141 Front Panel de la práctica PID

Práctica Valores de PID, es un vi comodín genera los valores de las constantes al agregarle un PID, se lo utiliza como método de comprobación de un proceso manual.

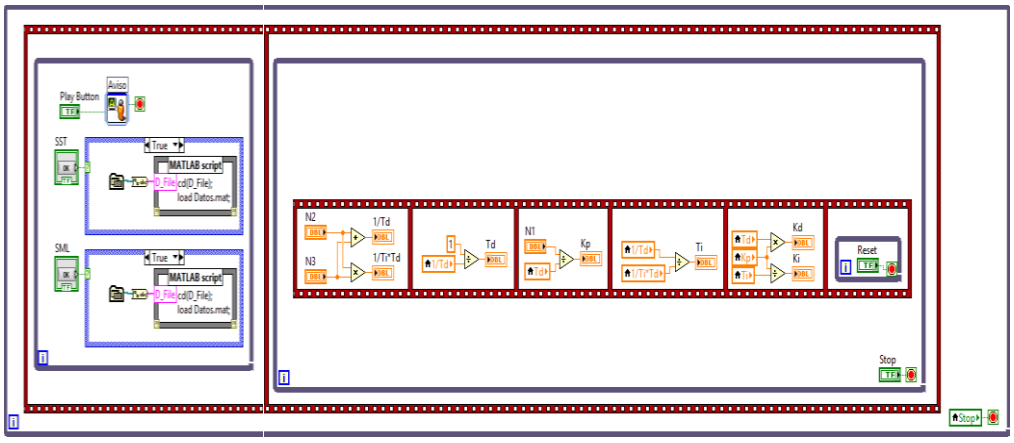


Figura 142 Block Diagram de la práctica PID

Imitando este proceso, pero aplicando los conocimientos de la Forma 2, realizar un VI que pueda obtener los valores de las variables para la segunda forma en que se presenta la función del control PID y comparar el resultado de estos valores resolviendo de manera manual en una hoja aparte.
De forma adicional comparar dichos valores tanto usando la forma 1 como la forma 2 de la función de control para comprobar las diferencias que existen.

4.3.4 PID Planta Motor de Paso

Se dará uso de un modelo obtenido para la planta "Motor de Paso". En sisotool y se cargaron los datos de la función de transferencia (figura 127) y se seleccionó en F el setpoint que se usará como referencia, en este caso 100; Se desactivaron todas las opciones salvo Root locus además se seleccionó la respuesta al paso y las opciones para que se grafiquen las señales de control y de salida.

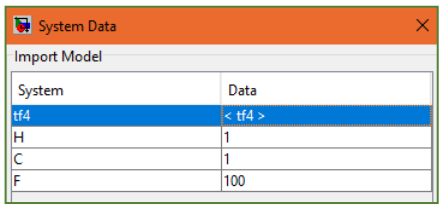


Figura 143 Colocación de datos

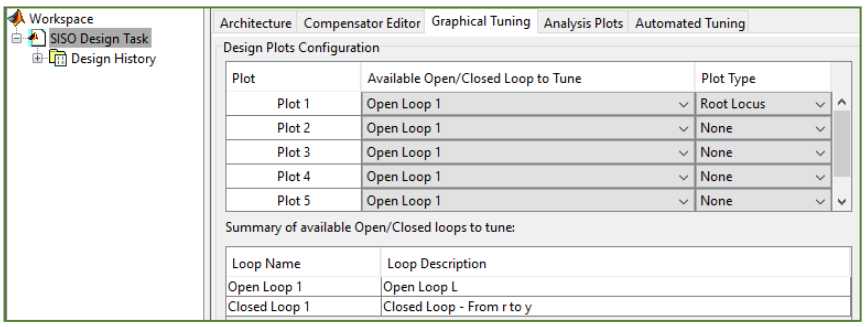


Figura 144 Selección del grafico a utilizar

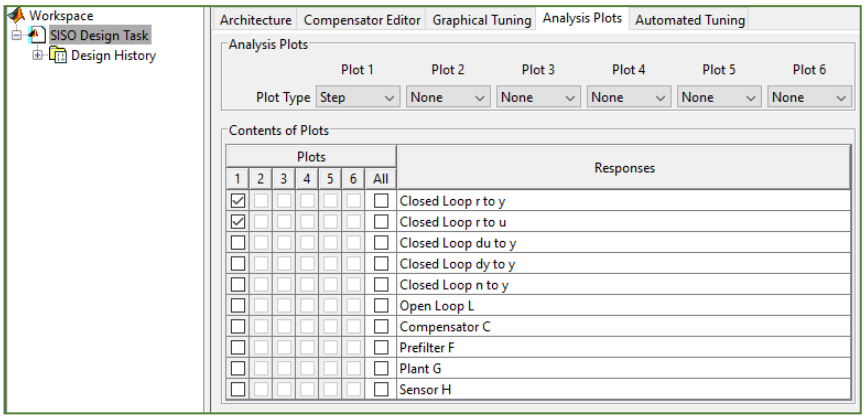


Figura 145 Elección del tipo de grafico

Se agregaron dos zeros y un integrador para formar un PID

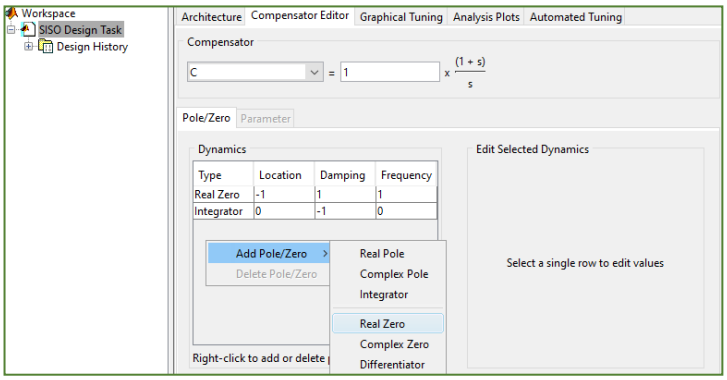


Figura 146 Agregar polos y zeros

Luego ubicando a cada extremo de la pantalla se selecciona el lugar de las raíces y la ganancia del controlador de manera gráfica desde la ventana donde se muestra el Robot Locus.

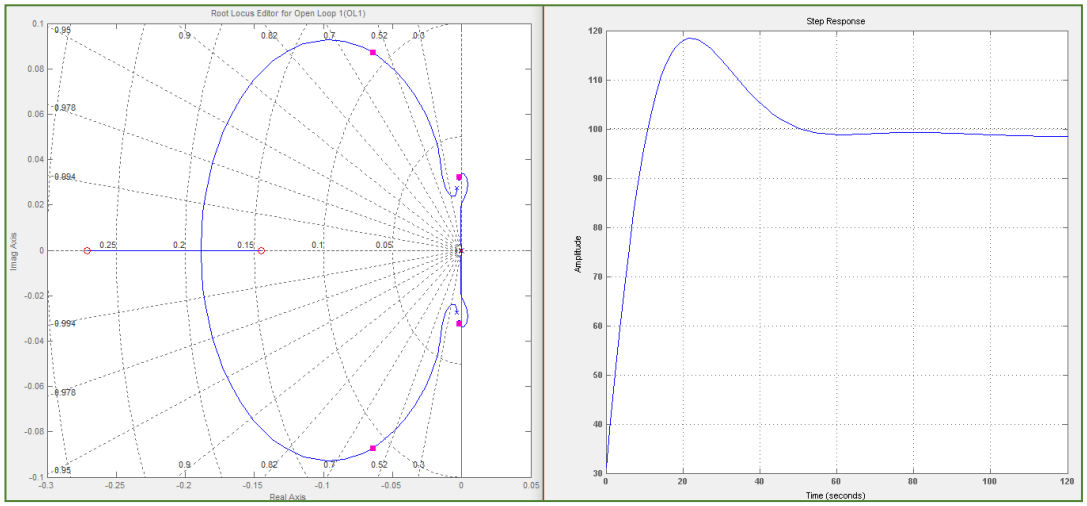


Figura 147 Gráficos del lugar geométrico de las raíces, y su respuesta al paso

Obteniéndose así un PID completo el cual es exportado al Workspace de Matlab para que se muestre en la forma requerida para obtener sus valores desde el VI de "Valores de PID".

Nota: Se recomienda guardar el diseño con el botón de "Store Design" y el " SISO Design Task" para no perder los controles elaborados.

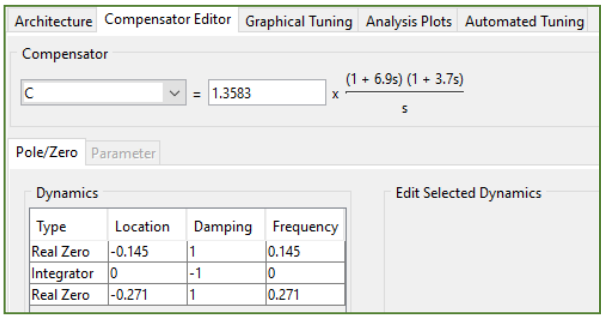


Figura 148 Datos del control PID

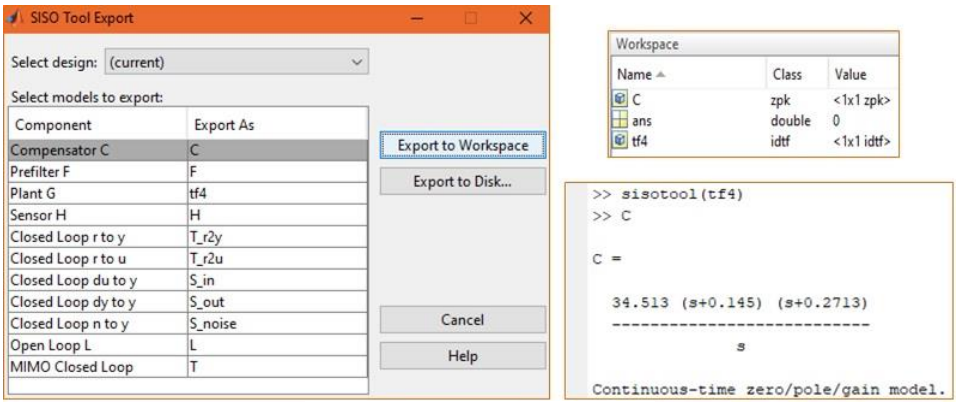


Figura 149 Exportación de datos

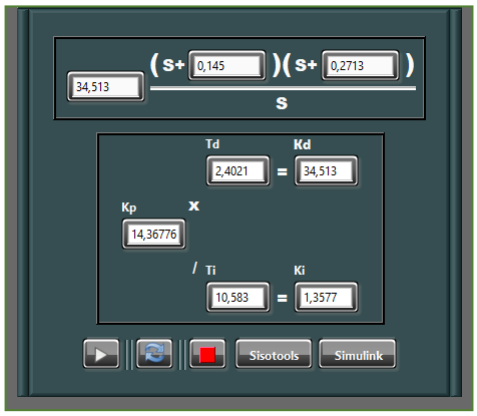


Figura 150 Valores de control PID

Y listo, con eso ya se tendría listo los valores del PID a implementar en el VI de PID de Motor de Paso.

4.4 Práctica #4 “Controlador PID en tiempo continuo”

4.4.1 Objetivos

- Repasar los puntos elementales del control PID.
- Ver las herramientas que brindan los VI de PID.
- Utilizar el VI de PID para realizar el control de una planta simulada.
- Utilizar el VI de PID para realizar el control de una planta real.

4.4.2 Teoría PID

Control PID es uno de los tipos de control más populares y fácil de comprender, este se divide en tres ramas que pueden o no estar juntas al mismo tiempo, junto con una serie de posibles estructuras y combinaciones, pero en este documento se va a centrar en solo una de estas estructuras y en las 4 posibles combinaciones que pueden ser realizadas utilizando la misma.

Donde u es el valor o señal de entrada, e es la señal del error que resulta de la diferencia entre la salida obtenida y la entrada del sistema y finalmente esta y la salida del sistema.

Luego se verá que existen tres ganancias, $Kp \;\; Ki \;\; Kd$ estas corresponden a proporcional, integral y derivativa de allí las siglas de este tipo de control.

En esta configuración se puede variar el tipo o combinación que se usa, variando directamente el valor de la constante que se deseen utilizar o silenciar.

De manera básica se tiene el control Proporcional, este escala la señal del error y permite llegar más cerca del valor de u denominado set point, que es el valor al que se desea que llegue la planta, no obstante, con solo un control proporcional no se puede eliminar el error de estado estacionario que es la diferencia que queda entre el valor deseado y el obtenido cuando la respuesta de control llega al punto estable y deja de haber cambios en la señal de salida.

Para solucionar este error de estado estacionario se suele implementar una compensación integral, la cual va a ir creciendo mientras la señal de error se mantenga por arriba de 0 o decreciendo en su defecto si la señal está por encima del valor deseado; Por esta capacidad el controlador PI (Proporcional integral) es uno de los más simples pero funcionales para implementar como controlador para una planta, cuando la velocidad no es importante; El llamado tiempo de establecimiento T_s y la constante de tiempo T, que son los valores

usualmente vistos cuando se mira la velocidad de respuesta de un control, donde la constante de tiempo es el tiempo que toma a la señal medida llegar al 63,2% del valor deseado (set point) y tiempo de establecimiento es el tiempo transcurrido desde que inicia el proceso hasta que llega al 95%del valor del set point o tres veces el valor de la constante de tiempo.

Para solucionar los problemas de velocidad de respuesta lo que se suele hacer es implementar un control con compensación derivativa, que resulta del análisis de la ratio de cambio del valor del error, al ver a qué ritmo este decrementa o incrementa y en base a eso aumentar o disminuir la respuesta del control, esto se consigue obteniendo la derivada de la curva del error que da la pendiente en cada punto evaluado.

De modo que por los puntos citados anteriormente el control PID resulta bastante competente para la labor de control ya que ataca varios de los problemas más usuales de una manera muy eficiente, no obstante, el punto clave de este control es el elegir las constantes correctas para obtener un comportamiento deseado ya que, de elegir mal, se va a tener un sistema inestable e ineficiente.

Estas constantes están relacionadas entre sí de la siguiente manera.

$$Kp \quad Ki = \frac{Kp}{T_i} \quad Kd = Kp * T_d \tag{30}$$

$$c(t) = K_p e(t) + \frac{K_p}{T_i}\int_0^t e(\tau)d\tau + K_p T_d \frac{de(t)}{dt} \tag{31}$$

Si se aplica la transformada de Laplace se obtiene las siguientes ecuaciones.

$$C(s) = K_p\left(1 + \frac{1}{T_i * s} + T_d * s\right) \tag{32}$$

$$C(s) = K_p + K_i * \frac{1}{s} + K_d * s \tag{33}$$

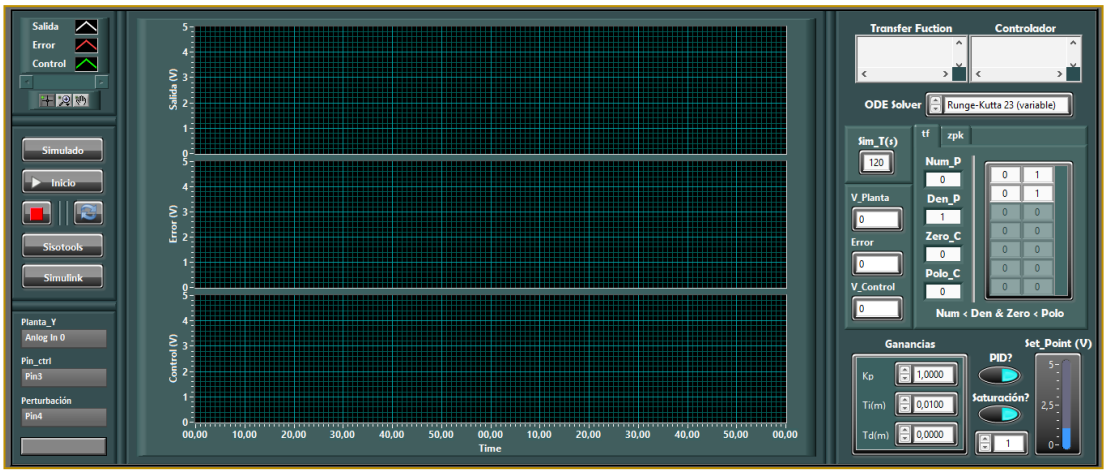


Figura 151 Front Panel Vi PID en continuo

En el centro se puede apreciar una pantalla dividida en tres partes, cada franja corresponde a cada una de las variables que se están analizando, esto para poder realizar una comparación entre cada una y evitar confusiones.

A cada lado de la pantalla central se encuentran los diferentes controles necesarios para las configuraciones y simulaciones, de manera similar a como ocurre con el VI de modelado, se puede intuir el funcionamiento de algunos, pero se puede notar que hay muchos más controles y botones que en el VI de modelado.

Se empezará viendo desde el lado derecho las herramientas que tiene este VI.

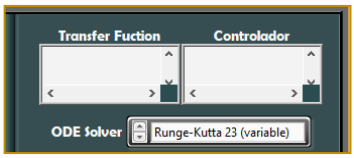


Figura 152 Visualizador de la Función de transferencia y controlador

En esta sección se encuentran dos pantallas pequeñas en las cuales se van a escribir la función de transferencia y el controlador a utilizar en las simulaciones, junto con un menú para elegir el tipo de modo de resolución que se desea utilizar para las simulaciones.

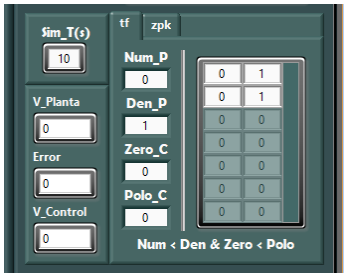


Figura 153 Visualizador de los valores en tiempo real

Luego se tiene la parte de visualización de valores donde se pueden ver en tiempo real los valores de salida de la planta, de la señal de control y de la señal del error, arriba de estos se tiene el tiempo de simulación en segundos y a su derecha se encuentran dos pestañas, solo se puede usar una a la ves y la que este seleccionada es la que va a utilizarse en modo simulación, en ellas se encuentran unos bloques de números que permiten ingresar cuantos polos y ceros se tengan o los grados del numerador y denominador como se puede apreciar en la imagen, los controles debajo de "Num_P y Den_P" están implementados para poder realizar la inclusión de una función de control, por ejemplo, implementar un control en adelanto o atraso.

Luego se tiene la matriz en la cual se va a escribir de arriba a abajo los números desde mayor grado en caso de tf, en caso de zpk es indiferente, sin dejar espacio se escribirían los números para el controlador, hay una leyenda abajo de la matiz que indica a que corresponde cada columna, como se muestra, se escribe numerador menor a denominador y se muestra que se tiene un numerador de grado 0 y un denominador de grado 1, y en la matriz se tiene 0 en las dos casillas de la columna del numerador y dos 1 en la del denominador formando así la función de transferencia $\frac{1}{s+1}$

Figura 154 Controles de las Ganancias y Set Point

Por último el lado derecho se tiene los controles para las ganancias del PID que se fueran a utilizar, un botón que permite activar o desactivar los valores del PID haciendo que por default se tenga una planta en lazo cerrado con una ganancia Kp igual a 1 y Ki como Kd igual a 0, el otro botón permite activar o no la saturación para ver si se puede implementar esos valores en la simulación dado que el Arduino que tiene una salida saturada de 0 a 5 v por diseño estructural, y finalmente la barra de set point junto con un control numérico para cambiar con precisión entre diferentes valores puntuales.
Como dato adicional a tomar en cuenta, los valores están dispuestos como Ti y Td en minutos, es decir, para simular se utilizarán los valores de Ti y Td normales en segundos, pero al momento de usarlo en el Arduino, deberemos de dividir por 60 dicho valor, para que este en minutos, y pueda ser utilizado, esto es debido a que se utiliza un bloque de programa de PID el cual trabaja de esta manera

Los controles del lado izquierdo del VI

Figura 155 Descripción de colores de las señales

Igual que en el Vi de modelado se cuenta con un visualizador de los colores y nombres de los gráficos mostrados en pantalla, junto con una barra de herramientas para la manipulación de dichos gráficos.

Figura 156 Botones que controlan el VI

Luego se tiene los botones que controlan la secuencia, pero en estos se encuentran dos botones adicionales, uno de ellos permite abrir "sisotool" que es una herramienta muy poderosa de Matlab que permite realizar una serie de pruebas diferentes a una planta cargada en el sistema, esta planta se carga de manera automática si el archivo Datos.mat se encuentra en la carpeta de trabajo, que por default es la misma carpeta donde se encuentran todos los VI, el segundo abre "Simulink" de la misma manera y carga los datos de Datos.mat al escritorio de trabajo.

Figura 157 Seleccionadores de pines

Finalmente están los controles para la selección de pines en el Arduino, hay que tener en cuenta que para el pin de control será necesario utilizar solo los pines que tengan este símbolo ~ alado de ellos ya que estos son los que permiten el uso del PWM, y que no pueden ser usados los pines 0 y 1 ya que estos son utilizados para la comunicación por USB con LabVIEW, por ello esta

preconfigurado en el pin 3 del Arduino, y luego se encuentra un selector llamado perturbación que es una salida digital que se conectará ya sea a un transistor para crear un swich electrónico o a cualquier aparato que pueda generar una perturbación al sistema, y se pulsará el botón de abajo para activar esta perturbación.

4.4.3 Para Practicar - Parte 1

Se empezará por utilizar una planta genérica para ver cómo funciona el proceso de simulación y practicar un poco los conceptos repasados sobre PID, al terminar de configurar todo, y dar inicio, se encontrará con que aparece un mensaje de aviso, sobre que se desactivaran dos de los botones de control, además si se utilizase un valor de 0 en el tiempo integral, se obtendrá otro mensaje adicional; Cada que se quiera ver los cambios que se han hecho en modo simulado, se deberá de dar clic en el botón de reste, para que se vuelva a realizar la simulación.

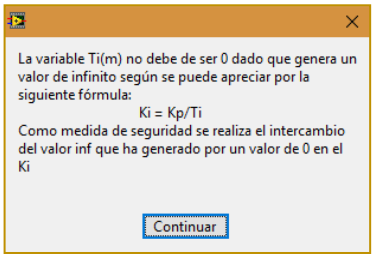


Figura 158 Aviso del VI

Como se puede apreciar la señal que se muestra es el resultado de evaluar la función de transferencia de la planta, mostrada en uno de los recuadros, con un controlador adicional de 1, que no afecta el proceso, y un control PID saturado, que pretende generar una respuesta que llega casi a 50v, se quitará la saturación para ver cómo cambia la gráfica.

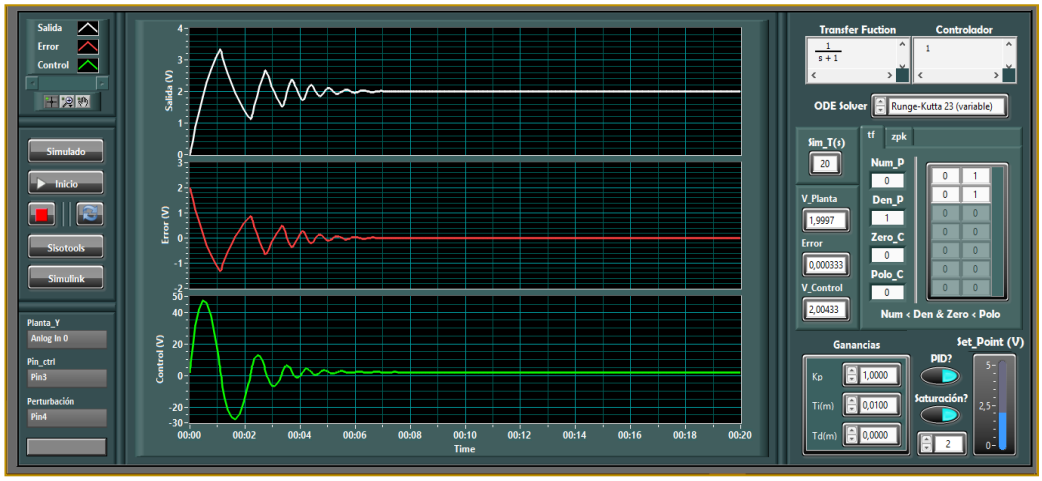


Figura 159 Control PID con Saturación

Cuando se desactiva la saturación, se puede apreciar que la acción de voltaje negativo del control genera un efecto en la planta, y cambia la dinámica de esta, pero este sigue siendo con control PI mal configurado.

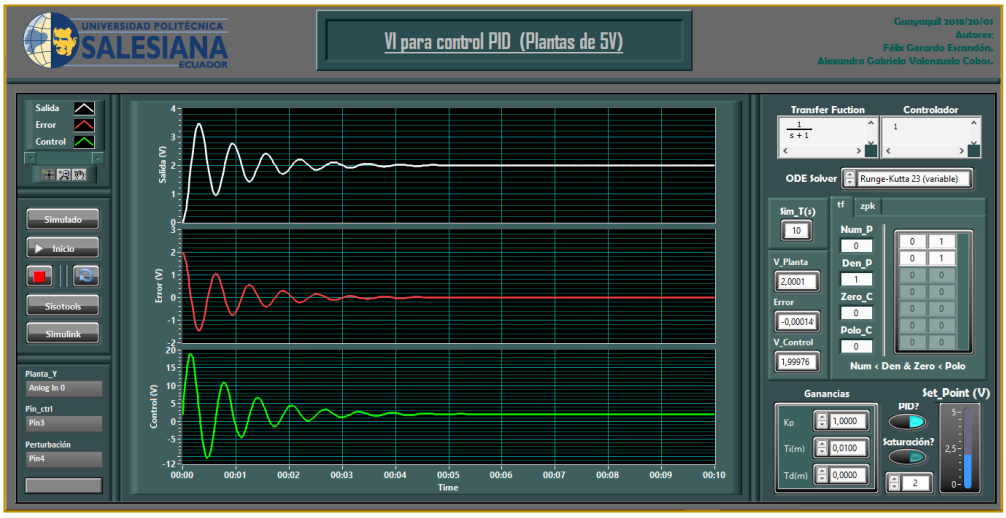


Figura 160 Control PID sin Saturación

Aumentando 10 veces el valor del tiempo de integración, se puede ver que la señal se comporta de una mejor manera, pero aún se mantiene fuera del rango de voltaje que se puede aportar con el Arduino, pero se puede notar que ahora la señal de control no tiene valores negativos.

Antes de seguir modificando se deberá comprobar que no exista mucha diferencia cuando se agregue el saturado.

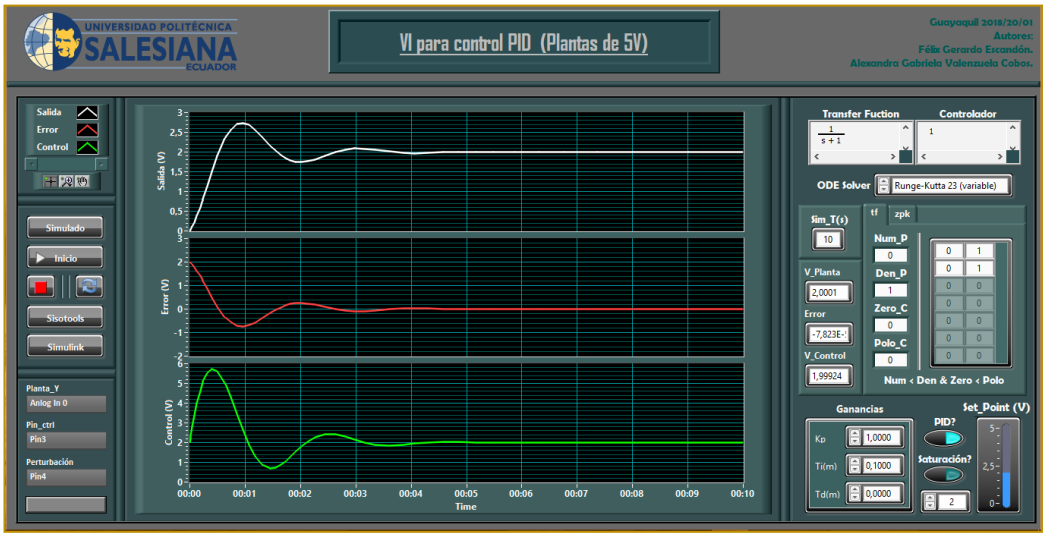


Figura 161 Aumento del tiempo de integración sin saturación

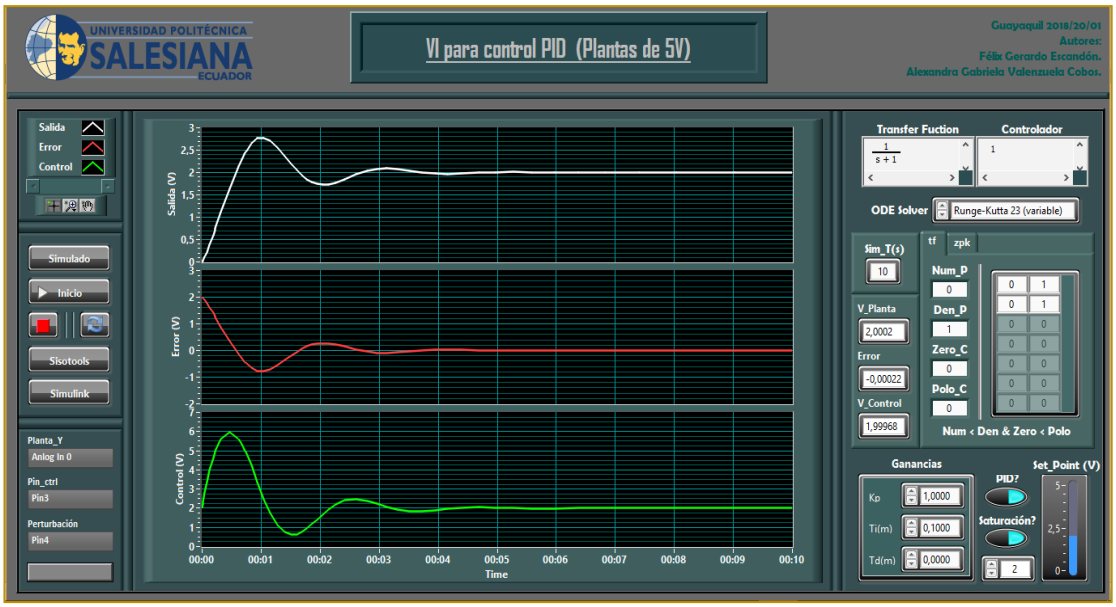


Figura 162 Aumento del tiempo de integración con saturación

Como se puede apreciar, no hay mucha diferencia, ya que esta es de apenas un voltio por un tiempo de medio segundo.

Al aumentar nuevamente el tiempo de integración se ve que la respuesta es completamente diferente, y que la planta llega al valor deseado en unos 4 segundos, ahora se deberá buscar reducir ese tiempo, pero antes se revisará que sucede si se utiliza un control netamente proporcional.

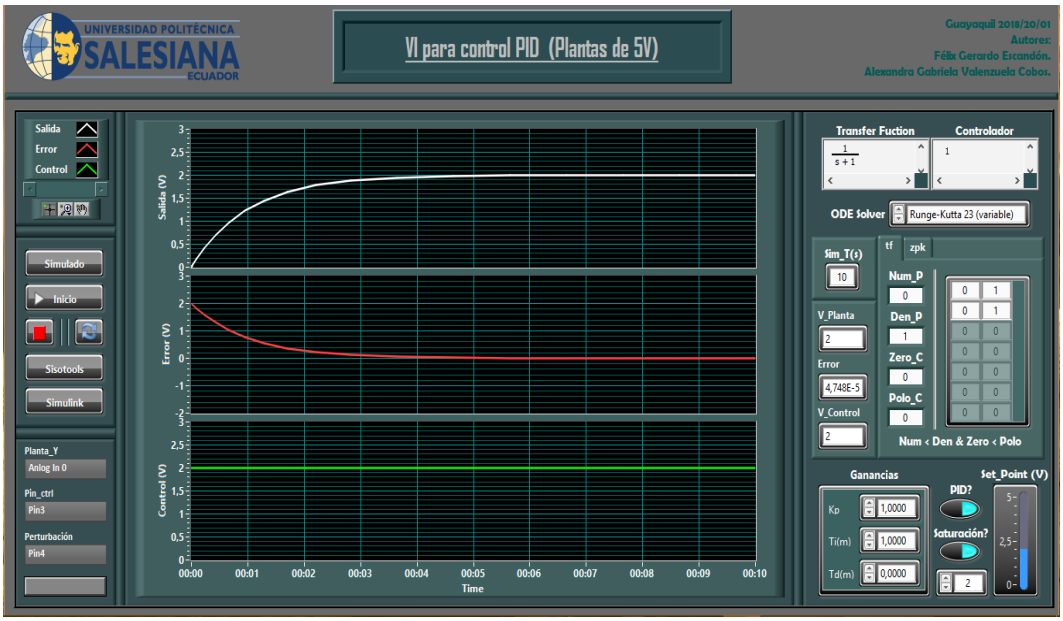


Figura 163 Segundo aumento de integración

Se reemplazará el control PI, reemplazándolo por un control proporcional igual a 1 y con 0 en Ti y Td, al desactivar el PID con el botón de "PID".

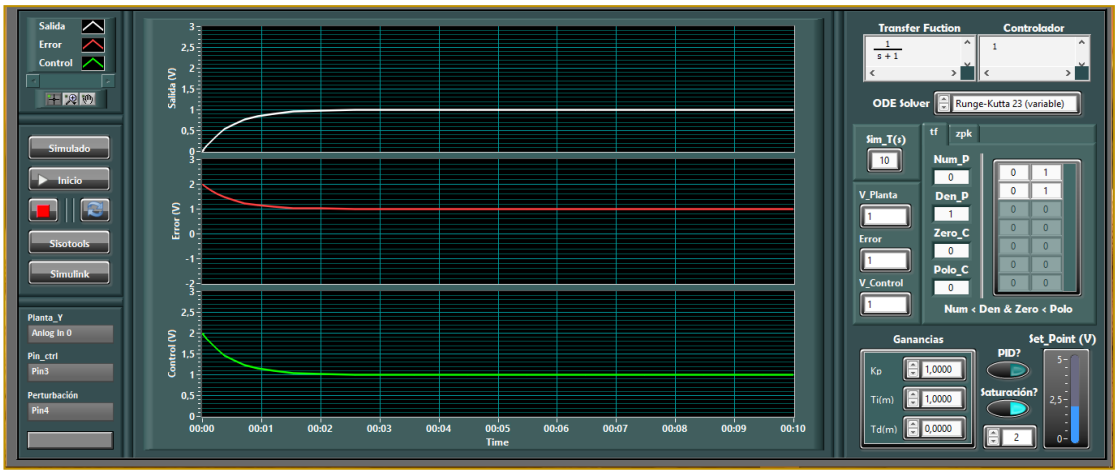


Figura 164 Desactivado el control PID

Como se puede ver, la planta solo puede llegar a la mitad del valor del set point, y que al tener desactivado el uso de las ganancias del el PID, o uso del PID por defecto que es el control proporcional de ganancia 1.

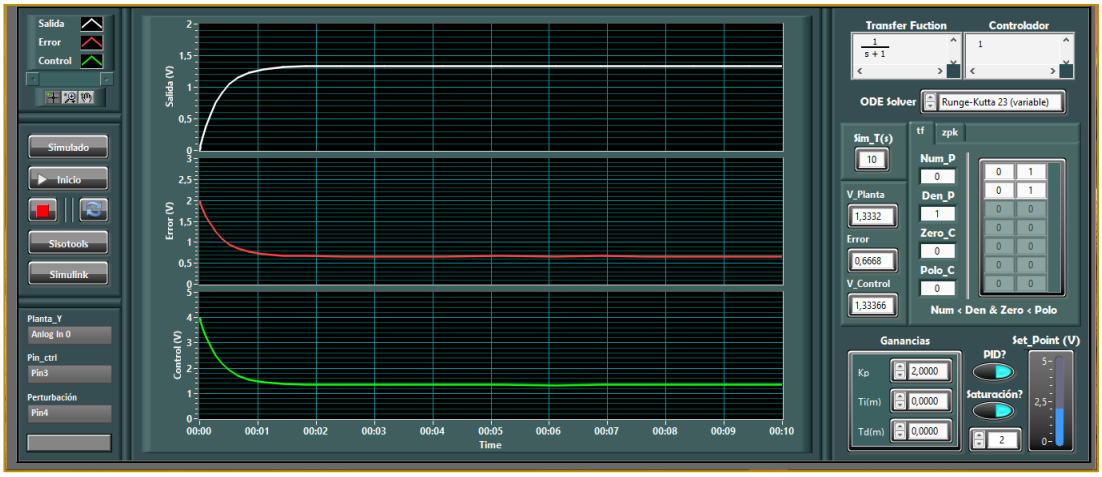


Figura 165 Aumento de la ganancia proporcional a 2

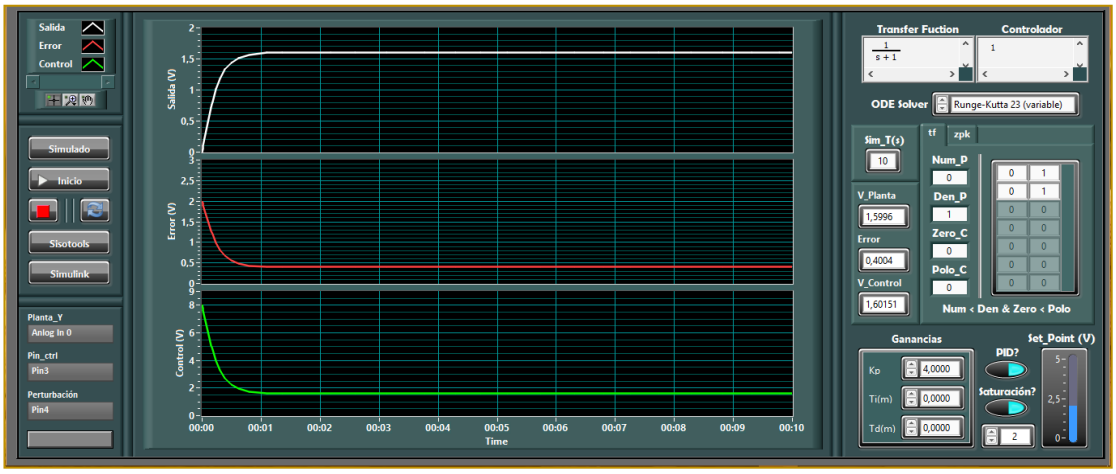


Figura 166 Aumento de la ganancia proporcional a 4

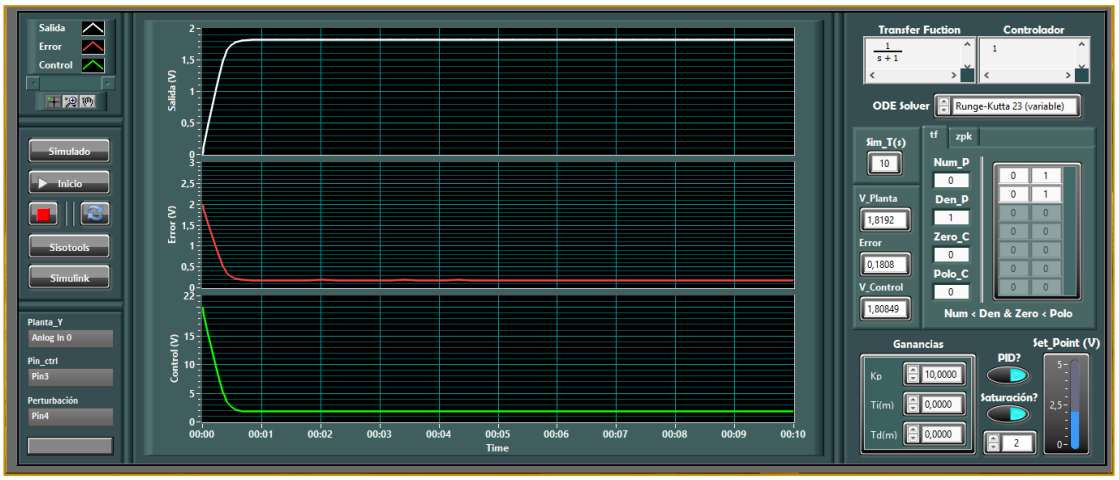


Figura 167 Aumento de la ganancia proporcional a 10

Como se puede apreciar, el control proporcional no es suficiente para conseguir el resultado que se desea, y dado que se encuentra saturado el sistema, este se comporta de forma similar en cada uno de los casos.
Habiéndose comprobado el comportamiento de la planta simulada con un control Proporcional se deberá regresar al control anterior pero ahora se deberá de utilizar el doble de la ganancia y un valor de 1 en el tiempo integral, para comprobar que sucede al aumentar la ganancia proporcional sin variar el tiempo integral.

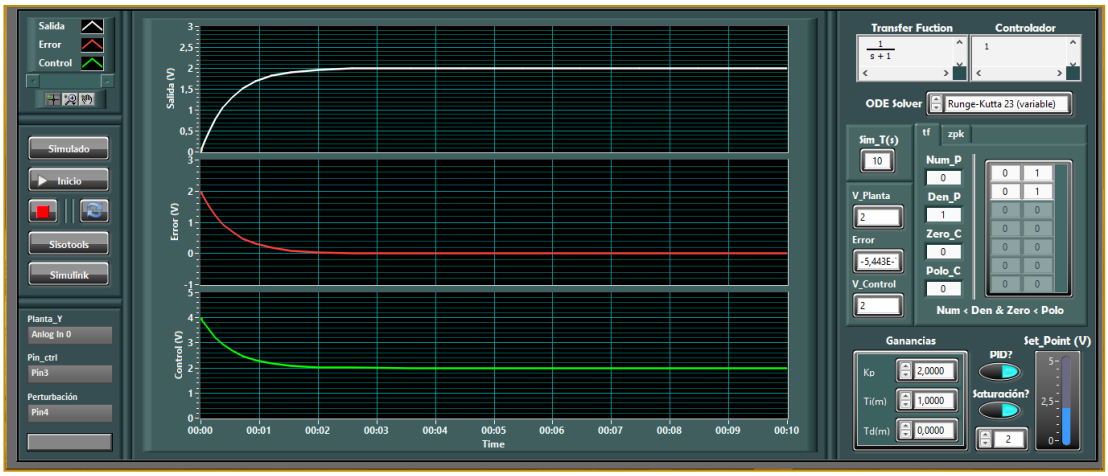


Figura 168 Control PID con tiempo de 1

Esto redujo el tiempo en el que la planta llega al set point a la mitad, solo con aumentar el valor de la ganancia proporcional y conservando el valor del tiempo integral, dado que el valor de ganancia proporcional también afecta el valor de la ganancia integral.
Al reducir el tiempo integral se puede apreciar que se empieza a generar un sobre pico en el sistema.

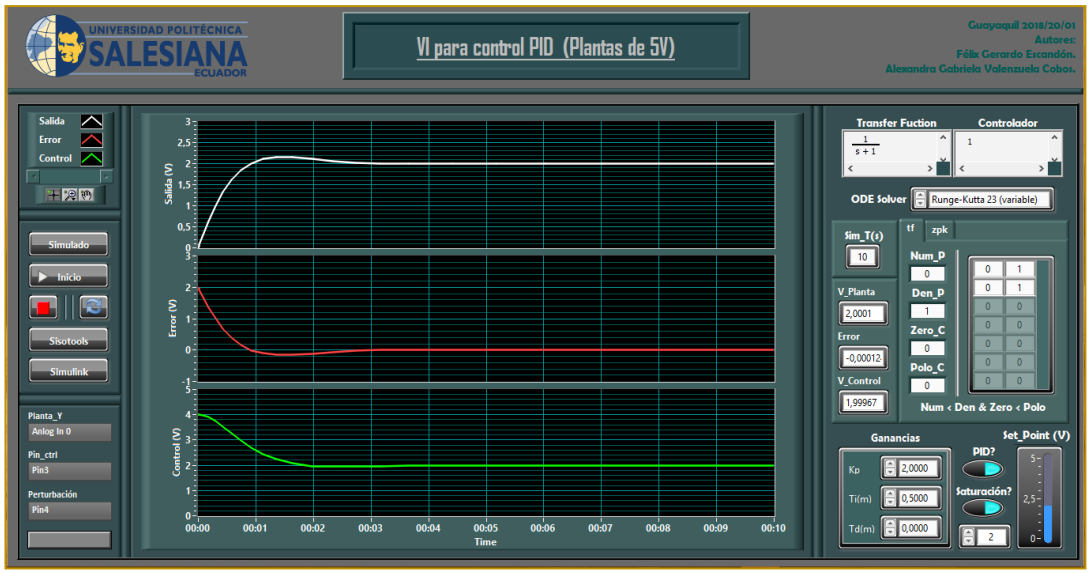


Figura 169 Control PID con un tiempo integral de 0.5

Y mientras más se baja el valor, mayor es el sobre pico, bajando el valor de ganancia proporcional.

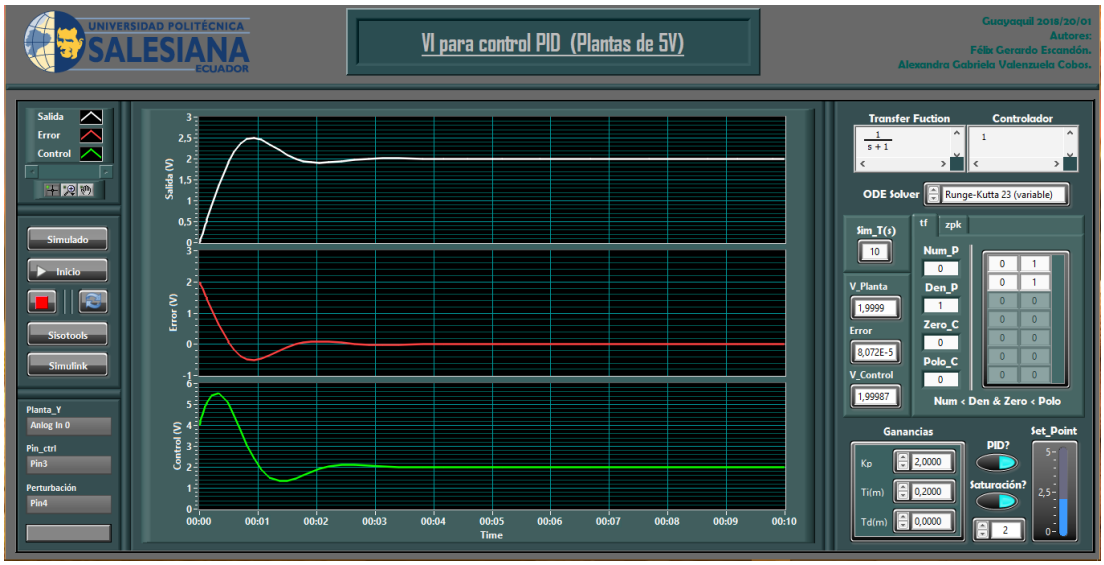


Figura 170 Control PID con un valor bajo de ganancia Kp

Al ir variando los valores de las ganancias se puede llegar a tener un control mucho más eficiente, esta señal tiene un parecido a la señal que resulta de tener Kp=2 y Ti=1, pero la diferencia clave se encuentra en la señal de control, de compararlas se podrá notar que este control, apenas necesita 2,5v para conseguir lo que el otro control consigue empezando con 4v, y además de eso, este tiene un comienzo mucho más suave a la vez que conserva la respuesta de tiempo y gana eficiencia en términos de voltaje de alimentación.

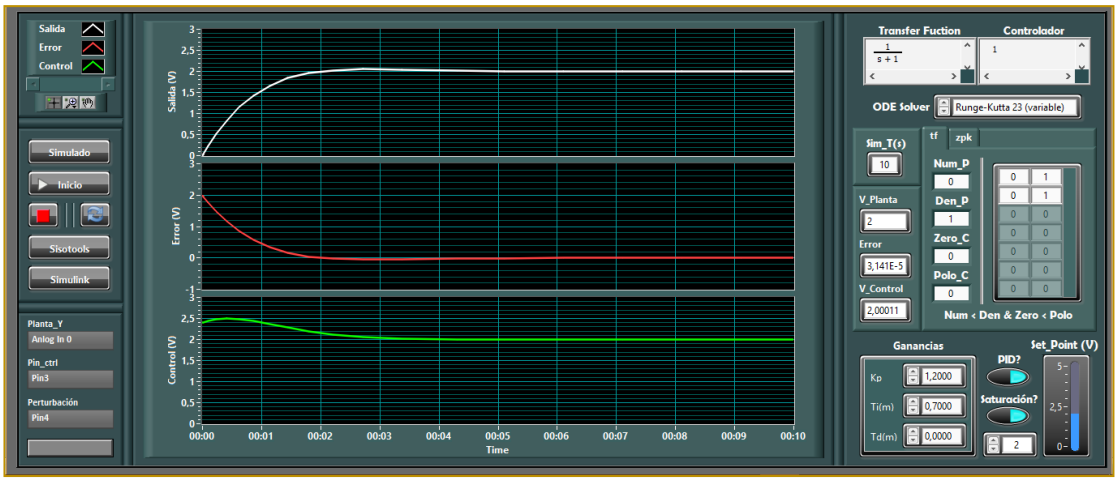


Figura 171 Control PID cambiando el Set Point

Si se cambia el set point se apreciará que el comportamiento del control y de la planta se mantienen iguales, solo se escalan al nuevo valor objetivo, esto permite trabajar un control PID con valores de set point pequeños, para luego ser utilizado con valores mucho más grandes.

A continuación, algunos ejemplos de diferentes valores para las funciones de transferencia que se pueden crear en el VI.

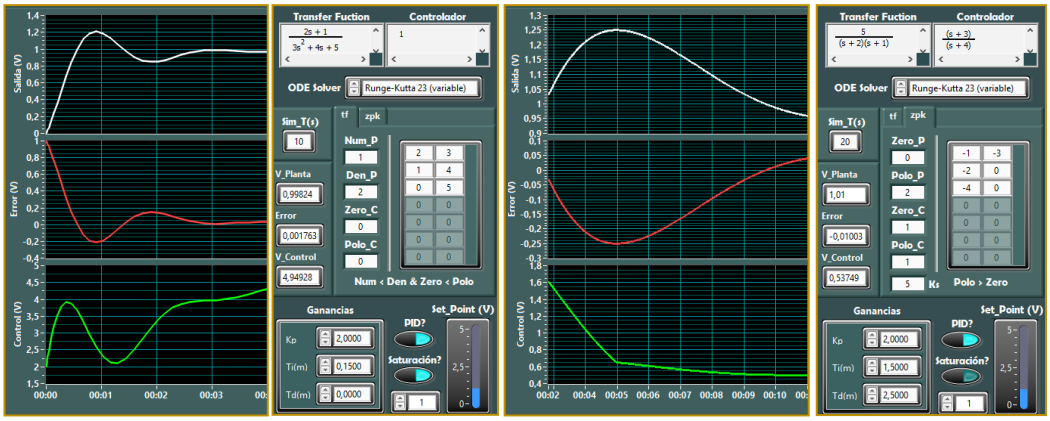


Figura 172 Ejemplos de valores PID

En el segundo ejemplo se puede ver que el valor de Ks corresponde a la variable K en el proceso de crear una función de la forma ZPK

4.4.4 Para Practicar - Parte 2

Se empezará por realizar el modelado de la planta que se desea controlar, de ser posible también se deberá realizar el modelado matemático de la misma para comparar ambas funciones.

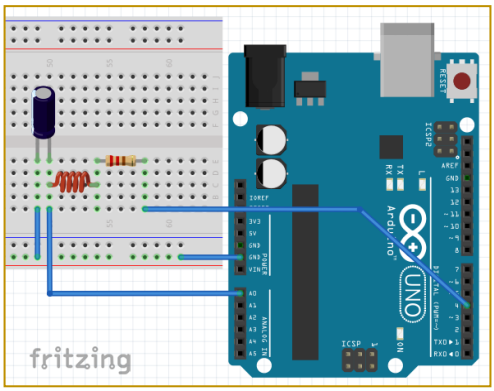


Figura 173 Esquema del circuito

Los materiales para utilizar en el circuito que se muestra son los siguientes:

- Una resistencia de 33000Ω
- Un capacitor de $100\mu F$
- Un inductor de $2200\mu H$

Este es el circuito (planta) que será utilizado para realizar la demostración del proceso de control, el cual es un circuito RLC al cual le corresponde el modelado matemático que se muestra a continuación.

Los circuitos RLC están gobernados por la siguiente formula, resultante del análisis de sus componentes distribuidos en serie en un circuito.

Siendo que el voltaje de entrada del circuito corresponde a la suma del voltaje de todos sus elementos se puede obtener la primera ecuación y sacando la transformada de Laplace se obtiene la segunda ecuación en términos de S.

$$V_{in}(t) = R * i(t) + L * \frac{di}{dt} + \frac{1}{C}\int i(t)dt \tag{34}$$

$$V_{in}(s) = R * I(s) + L * s * I(s) + \frac{1}{C} * \frac{1}{s} I(s) \tag{35}$$

y dado que el elemento cuyo voltaje será medido para realizar el control es el capacitor, entonces se tiene que el voltaje de salida corresponde al voltaje del capacitor.

$$V_{out}(s) = \frac{1}{C} * \frac{1}{s} I(s) \tag{36}$$

De donde la función de transferencia de este circuito seria la relación entre la salida y la entrada expresada de la siguiente forma.

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{\frac{1}{C}*\frac{1}{s}I(s)}{I(s)*\left[R+L*s+\frac{1}{C}*\frac{1}{s}\right]} \tag{37}$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{C}*\frac{1}{s}*\left[\frac{1}{R+L*s+\frac{1}{C*s}}\right] \tag{38}$$

$$\frac{V_{out}(s)}{V_{in}(s)} = \frac{1}{C*L*s^2+C*R*s+1} \tag{39}$$

Con esta función de transferencia se puede concluir que se necesitará de una función que tenga dos polos y ningún Zero para poder realizar una comparación adecuada entre la función modelada en "ident" y la resultante del modelado matemático.

Como se puede apreciar existen diferencias entre ambas funciones de transferencia, pero de realizarse una división para acomodar las funciones a una misma forma se podrá ver varias similitudes ligeras entre sus valores.

```
tf1 =

  From input "u1" to output "y1":
         2127
  --------------------
  s^2 + 6534 s + 2125

Name: tf1
Continuous-time identified transfer function.

Parameterization:
   Number of poles: 2   Number of zeros: 0
   Number of free coefficients: 3
   Use "tfdata", "getpvec", "getcov" for parameters and their uncertainties.

Status:
Estimated using TFEST on time domain data "Datos".
Fit to estimation data: 99.05% (simulation focus)
FPE: 0.0002458, MSE: 0.0002415
```

Figura 174 Función de transferencia en Ident

$$tf1 = \frac{1}{0{,}47m*s^2+3{,}072*s+0{,}99} \tag{40}$$

$$Modelado\ Mat = \frac{1}{2{,}2\mu*s^2+3{,}3*s+1} \tag{41}$$

Una vez se haya obtenido la función de transferencia de la planta se podrá pasar a la herramienta sisotool para diseñar el control PID añadiendo dos zeros y un integrador.

Una de las formas de seleccionar los valores de la ubicación de los polos es de manera intuitiva viendo cómo se comporta la curva de respuesta al paso.

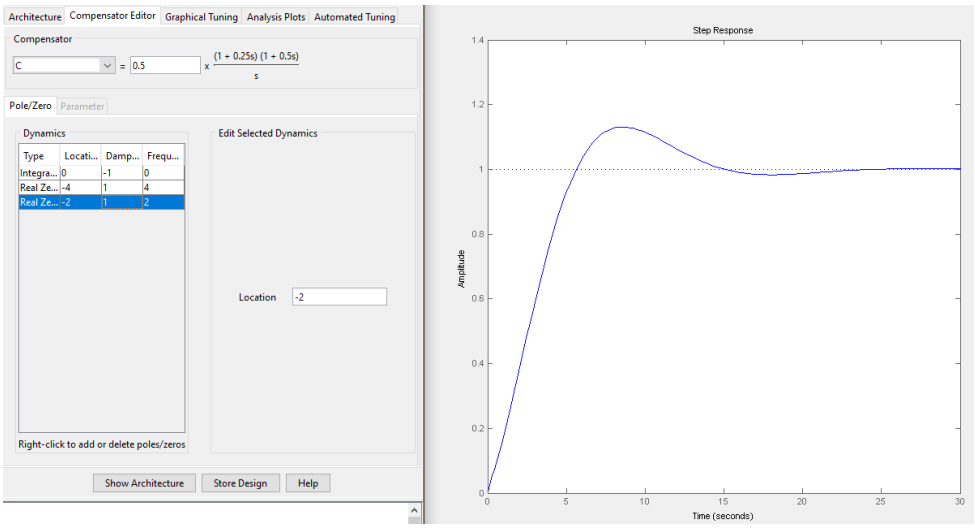


Figura 175 Respuesta con la función Step

Luego de esto se realizará la exportación de dicha función, y se pasará a obtener los valores de esta utilizando el VI de "Valores de PID", o en su defecto obtener los valores de las constantes utilizando las fórmulas vistas en el PDF de "Valores de PID y Sisotool".

Teniendo los valores de las constantes del controlador PID se puede realizar una última prueba realizando una simulación utilizando la función dibujada por Sisotool y compararla con la dibujada por el simulador de LabVIEW utilizando las constantes del PID y la función de transferencia que se obtuvo del proceso de modelado.

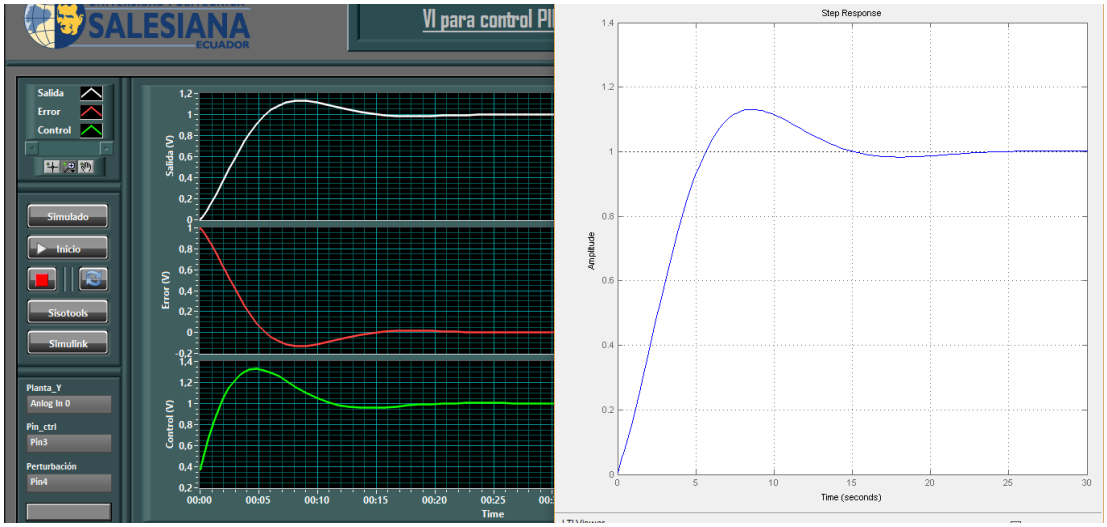


Figura 176Simulador LabVIEW vs Sisotool

Como se puede apreciar los resultados de ambas funciones son iguales hasta donde se puede comparar con solo la vista, pero hay un detalle importante para tener en cuenta de este proceso, y es el importante rol que cumple el selector llamado "ODE solver" el cual, dependiendo de la función simulada y el solver seleccionado puede llegar incluso a no mostrar la simulación.

Escogiendo el "Gear'sMethod (Variable)" se puede ver que si se genera la simulación de manera ideal.

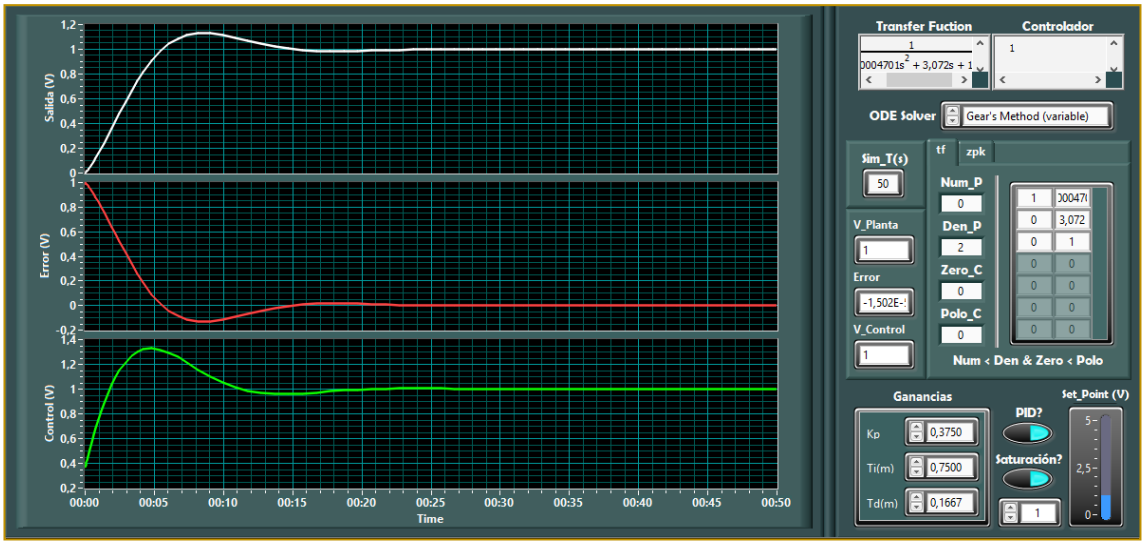


Figura 177 Simulación con Gear's Method

Mientras que al utilizar el "Runge-Kutta 45(Variable)" se puede ver que las funciones se van al infinito y no se consigue una gráfica que sea de utilidad.

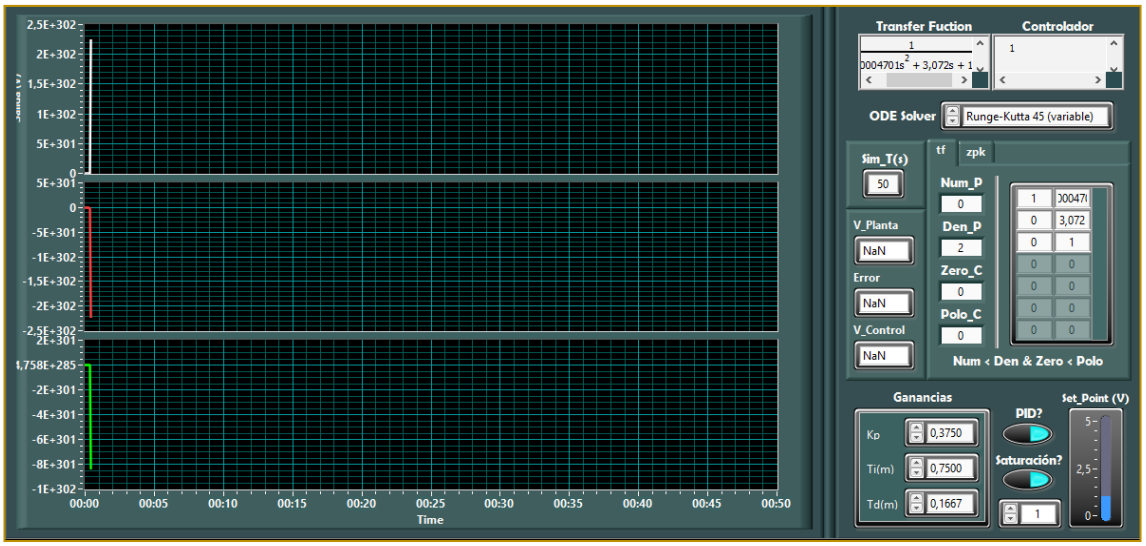


Figura 178 Simulación con Runge-Kutta 4

Ahora solo queda el paso final, se deberá dividir los valores de Ti y Td para que estos estén de acuerdo con el sistema utilizado en el bloque de control que se utiliza para realizar el control con el Arduino, este indica que Ti y Td deben estar en minutos, lo cual lleva a realizar una división para 60 de dichos valores.

Para Ti

```
» 0.75/60

ans =

    0.0125
```

Para Td

```
» 0.16667/60

ans =

    0.0028
```

Y como se puede apreciar la señal de control, de error y de salida son iguales a las simuladas, concluyendo así que el proceso fue 100% exitoso de principio a fin, y que ahora se tiene un control PID para un circuito RLC que funcionará para todos los niveles de voltaje que se seleccionen hasta el que el valor de sobre pico llegue a 5v,luego de esto ya no se podrá utilizar dadas las limitaciones físicas del Arduino.

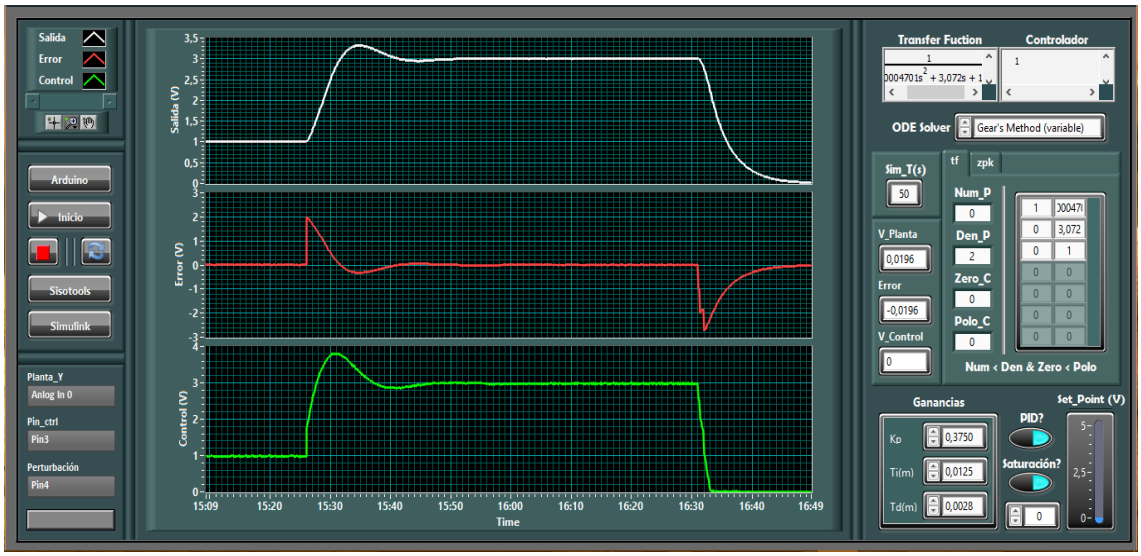


Figura 179 Control PID para un circuito RLC

Es importante destacar que aun cuando el control funciona, esto no indica que sea el mejor, por lo que se motiva a buscar una mejor configuración y realizar varias pruebas según sea necesario o según los objetivos impuestos para este control en términos de tiempo de establecimiento o valor de sobre pico entre otros.

4.4.5 Control PID de la planta Motor de Paso

Para esta demostración se utilizó la planta "Motor de Paso", como una continuación de la **“Práctica 3**
Se inicia obteniendo un modelo para la planta, el cual se muestra en la figura 180.

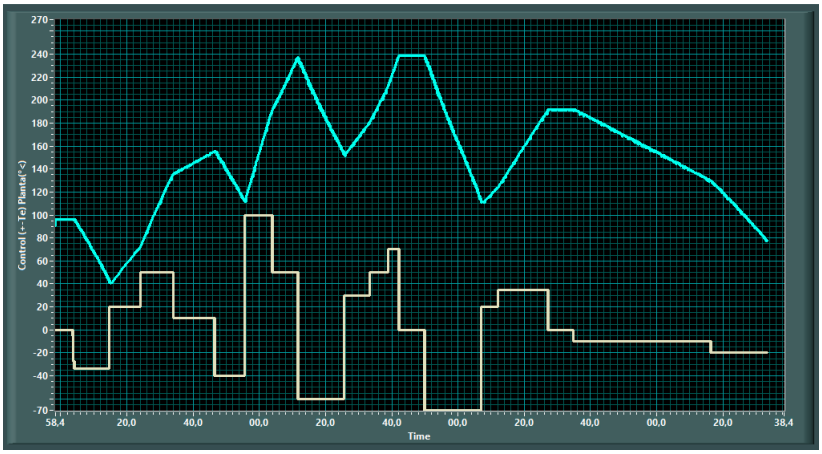


Figura 180 Modelado de la Planta Motor de Paso

Luego de esto se obtuvo en ident con los datos del modelo, una función de transferencia asociada.

```
  From input "U" to output "Y":
     0.01258 s^2 + 5.231e-05 s + 1.299e-05
  ---------------------------------------------
  s^3 + 0.007518 s^2 + 0.0007695 s + 5.643e-07
Name: tf4
Continuous-time identified transfer function.
```

Figura 181 Función de transferencia a usar

La cual se utilizó en sisotool para construir un control PID que fuese útil para la planta en cuestión, posteriormente se utilizó el VI "Valores de PID" para obtener los valores de las constantes del PID para poder ingresarlas en el VI de "Práctica_PID - Planta MPaso".

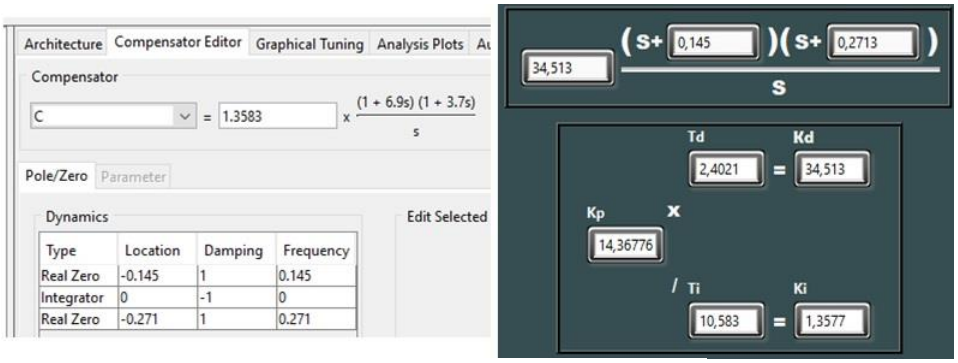


Figura 182 Obtención de los valores PID

Con lo cual se obtuvo el siguiente resultado, en el cual se puede apreciar una similitud en el sobre pico mostrado en sisotool, y también se puede apreciar pequeñas variaciones de un paso en torno al valor del set point.

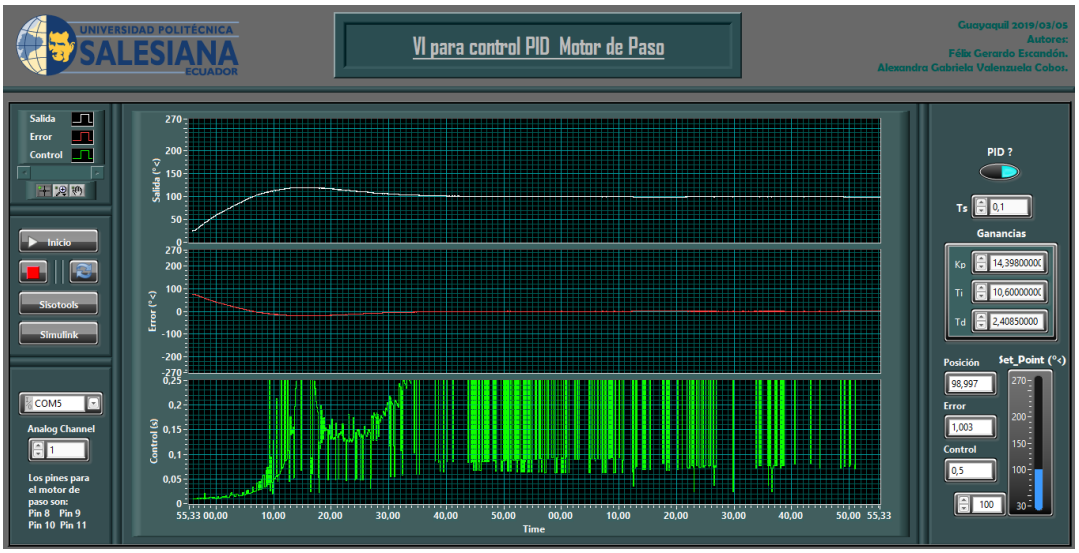


Figura 183 Control PID de la Planta Motor de Paso

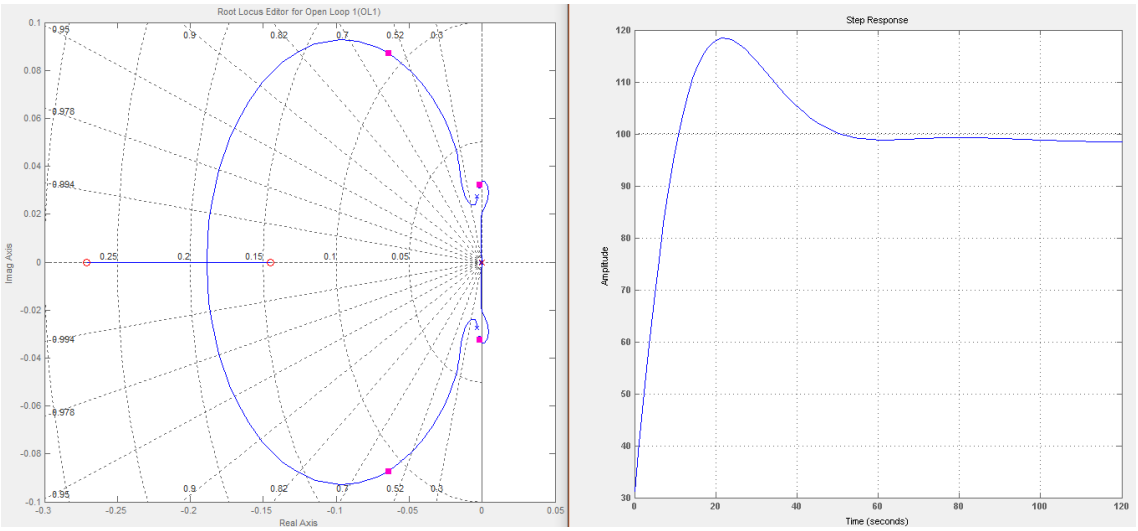


Figura 184 Grafica del lugar geométrico de las raíces y de su respuesta al paso

4.5 Práctica #5 “PID Discreto”

4.5.1 Objetivos

- Ver las diferencias entre el control PID en tiempo continuo con el discreto.
- Utilizar el VI de Modelado para obtener un modelo en Z de una planta.
- Utilizar el VI de PID en Z para realizar el control de una planta.

4.5.2 PID discreto

La diferencia clave del control PID en tiempo continuo y el control PID en tiempo discreto radica en un punto especifico, y este es precisamente el tiempo.
El control PID en tiempo continuo es un control más simple de implementar, el cual puede ser realizado utilizando circuitos, opanes, resistencias y capacitores diversos para conseguir el efecto deseado, mientras que el control PID en tiempo discreto un control digital, es decir que requiere del uso de un sistema más complejo para ser implementado, esto lleva a que sea un control que requiere más elementos involucrados en el proceso de control, pero que permite tener un rango de control mucho más amplio y preciso.

Esta diferencia en el proceso lleva a ver el proceso como interacciones realizadas de manera secuencial cada x tiempo por un computador, por lo cual la manera en que se utiliza la ecuación del control PID cambia y se vuelve de la siguiente manera.

$$c(k) = K_p e(k) + \frac{K_p}{T_i} \sum_{k=0}^{\infty} e(k) * T + K_p T_d \{e(kT) - e(k-1)\}/T \qquad (42)$$

Donde k indica el numero de muestra actual, T indica el tiempo de muestreo y como se puede apreciar, la integral cambia por una sumatoria del valor del error actual multiplicado por el tiempo T que ha pasado, mientras que la derivada se vuelve una resta entre el error actual con el anterior dividido para el tiempo.

Existen otros tipos de aproximaciones que se pueden aplicar, como, por ejemplo, en lugar de utilizar netamente el valor del error actual, utilizar un promedio entre el valor anterior y el actual, así como también hay una aproximación llamada de velocidad, que realiza una comparación con el doble de variable de tiempo, resultando en una función PID discreta diferente a la mostrada previamente.

4.5.3 VI de PID Z

Se revisarán para empezar los pequeños cambios que tiene este VI respecto a su contraparte en el tiempo, el cual difiere solamente en la forma usada para graficar las señales y en la adición de un control numérico para seleccionar el tiempo de muestreo, luego de esto veremos la parte de programación grafica por bloques correspondiente al proceso de simulación y al proceso de control PID discreto utilizando el Arduino.

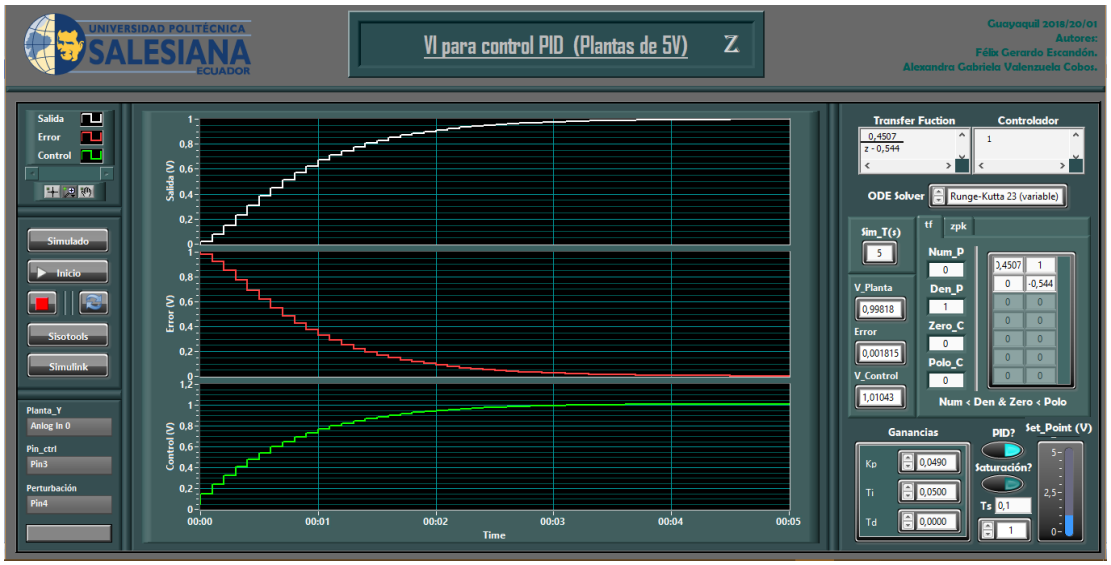


Figura 185 Front panel del VI para el control PID en Z

En el proceso de simulación es utilizado el toolkit de "Control Design and Simulation" para realizar un proceso de control simulado, para el cual se contempla como variables externas el tiempo de muestreo, el tiempo de simulación, y el tipo de resolución que se desea utilizar para la aproximación de las señales, internamente, se puede ver que se controla el valor del set point, que existe una selección entre un valor por defecto de las cargas del controlador, y un selector para la adición de una interferencia, el control PID contiene un proceso de evaluación para Ti = 0, un ZOH, un saturador y un selector que recibe la función de transferencia de la planta y finalmente el graficador.

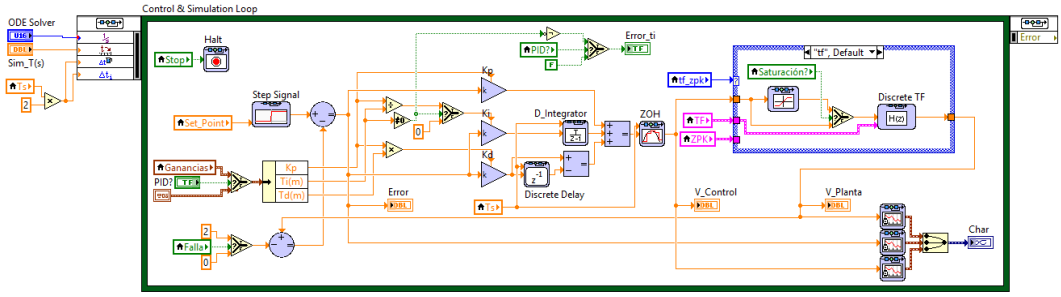


Figura 186 Block Diagram de Control

La contraparte del modo simulado es el proceso de control utilizando el Arduino, el cual es más sencillo, pero más extenso que el anterior.

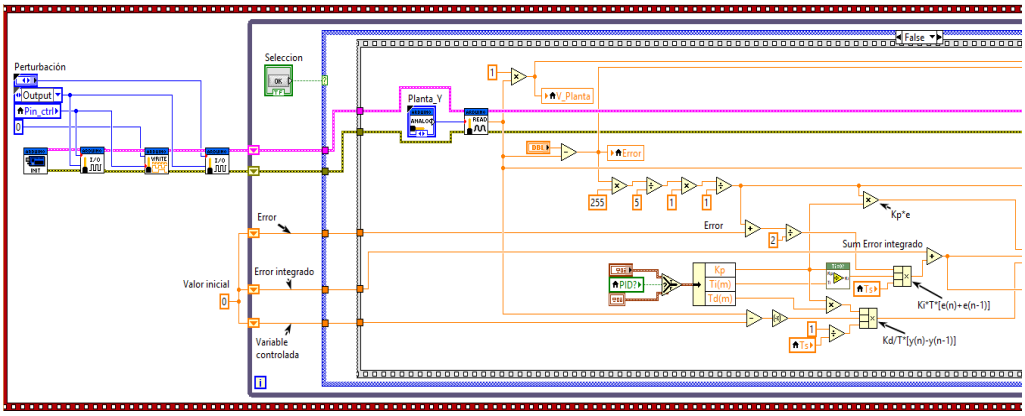


Figura 187Block Diagram Control PID en z parte 1

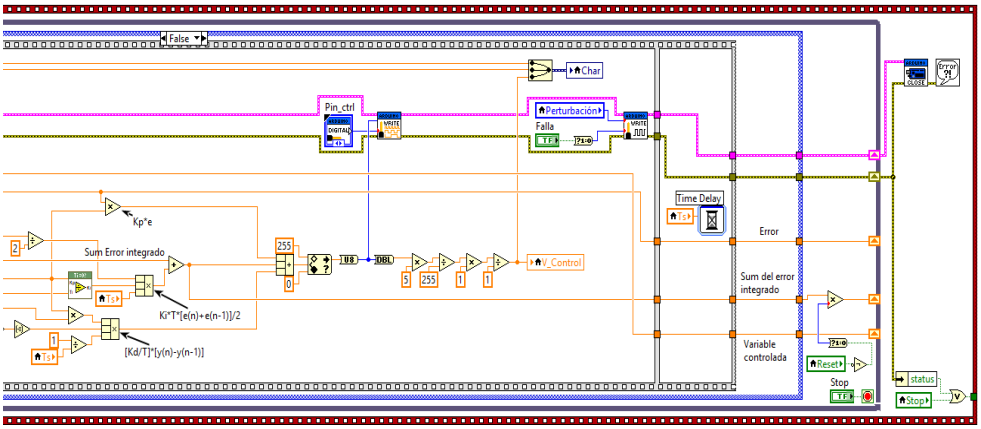


Figura 188 Block Diagram Control PID en z parte 2

4.5.4 Para Practicar

Será utilizado el siguiente circuito para esta práctica.

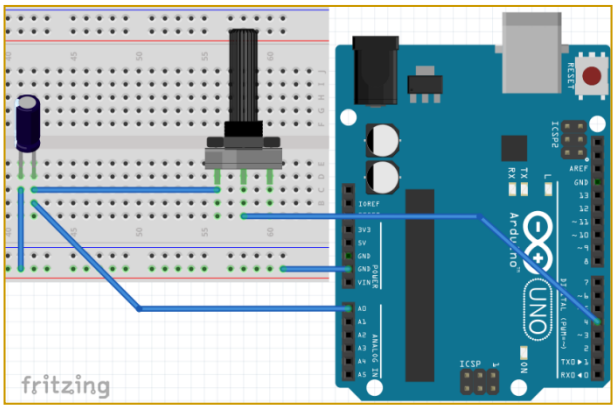


Figura 189 Diseño del circuito

Los materiales para utilizar son los siguientes:

- Un potenciómetro de $\mathbf{5}\boldsymbol{K}\Omega$.
- Un capacitor de $\mathbf{1000}\mu\boldsymbol{F}$.

La cantidad de resistencia que sea utilizada se recomienda sea suficientemente alta para que el circuito tenga un valor de voltaje estable; Posterior a la conexión se empezará por realizar el proceso de modelado, para esta demostración en particular se está utilizando un tiempo de muestreo de 0.2s, la resistencia medida en el potenciómetro es de $4{,}5K\Omega$ y el proceso de muestreo fue realizado utilizando el VI de Modelado.
La función tf1 será realizada con las características que se aprecian en las imágenes siguientes.

```
From input "u1" to output "y1":
                0.04342
z^(-1) * ---------------
            1 - 0.9566 z^-1
Name: tf1
```

Figura 190 Función de transferencia

Luego de esto será necesario revisar que tanto se aproxima la respuesta del modelo a la señal real, y revisar la ubicación de los polos del sistema para determinar si este es estable o inestable.
Tener en cuenta que para que un sistema en z sea estable, sus polos y zeros deben estar dentro del circulo unitario sin importar su signo.

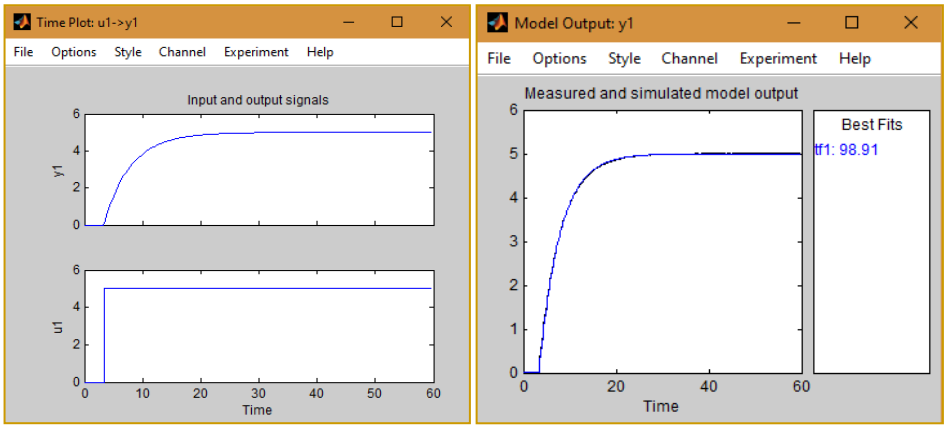


Figura 191 Señales de entrada, salida y aproximación.

Como se puede apreciar, se tiene una proximidad muy grande del 99,91%, y sus polos están dentro del círculo de la unidad, junto con un cero en el origen. Y el circuito en lazo abierto, llega a la señal de referencia en unos 20s, lo cual cambia una vez que se lo ingrese a un sistema de control en lazo serrado como se va a ver más adelante.

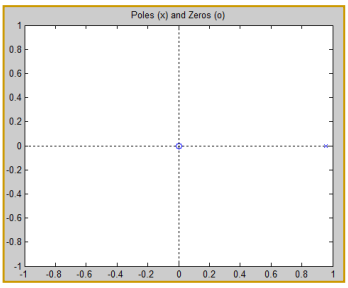


Figura 192 Grafica de Polos y Ceros

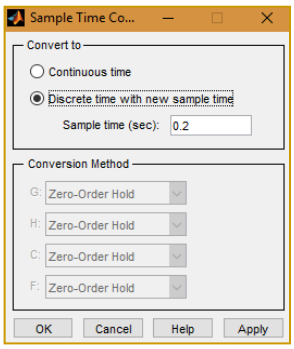


Figura 193 Cambio a tiempo discreto

El siguiente paso es abrir el VI de PID en Z e introducir los valores de la función de transferencia tf1 en el caso de este ejemplo, y luego antes de dar al botón de Play e iniciar el proceso de simulación, se deberá primero tener un control PI discreto que utilizar, esto debido a que es el óptimo para esta planta en cuestión.

Importante tener en cuenta también el configurar tanto la planta G que debe tener el valor de tf1, y el cambiar el sistema para que sea en tiempo discreto como se muestra en la imagen.

Una vez se tenga esto se deberá de configurar para que se despliegue la venta de respuesta al paso para poder ver la respuesta al paso del sistema en lazo cerrado con ganancia unitaria.

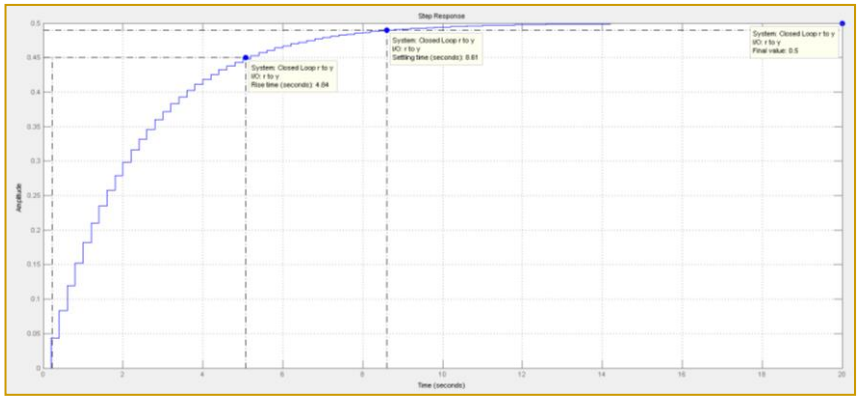


Figura 194 Grafica del control PID en Z

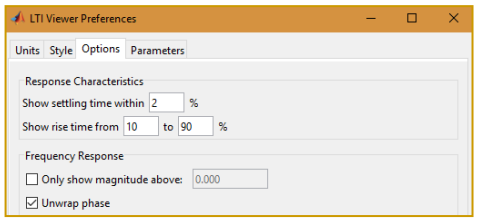


Figura 195 Características de la planta

Al hacer esto se puede ver que la planta tiene un tiempo de subida de poco más de 4,5s y un tiempo de establecimiento de casi 9s y que el error en estado estacionario es de 0,5v es decir la mitad del valor del set point.
Con el fin de obtener un conocimiento más profundo de cómo se comporta esta planta frente a diferentes partes del control PI, se empezará seleccionando una ganancia de 3 y se analizará la respuesta.

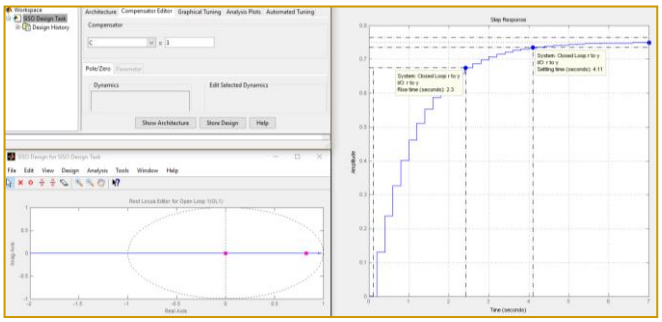


Figura 196 Cambiando valor de ganancias

Como se puede apreciar, solo con un control proporcional se puede conseguir que se mejoren los tiempos de respuesta del control para llegar a un valor, con lo cual se puede deducir que si lo que se desea es que la planta responda en un tiempo específico para llegar a un valor, entonces con solo un control proporcional se podría obtener lo que se desea; Pero es importante analizar la respuesta que arroja el controlador.

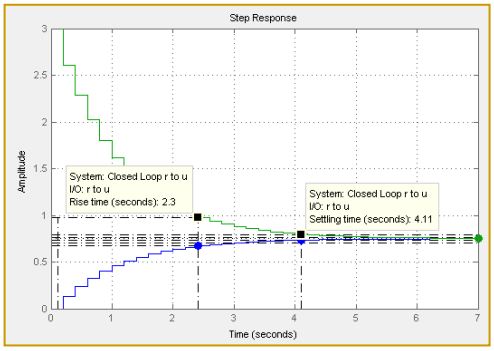


Figura 197 Respuesta del controlador

En la señal verde se puede apreciar que la señal de control empieza en 3 por la amplificación del error en 3 y va reduciéndose conforme el error se reduce también y la señal se acerca al set point.

Analizara la respuesta de la planta al agregar un integrador puro al sistema con una ganancia unitaria al control.

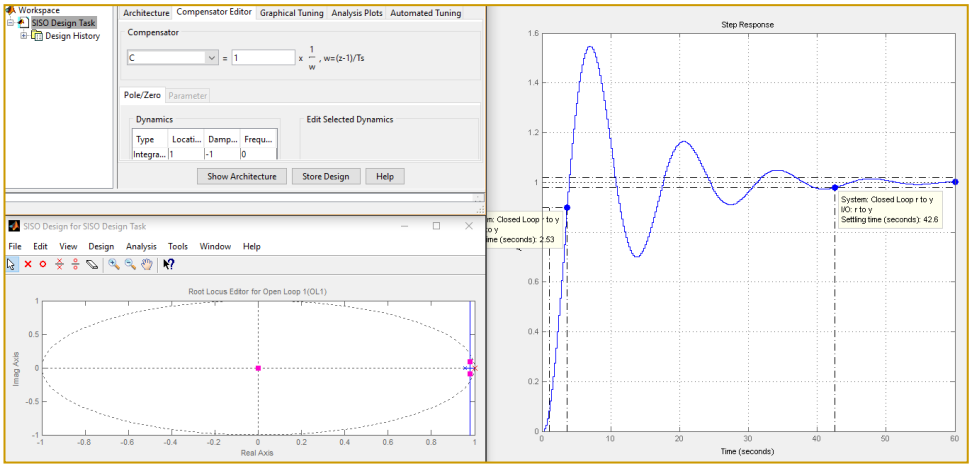


Figura 198 Grafica con un integrador puro una ganancia unitaria

Al haberse agregado el integrador puro en el control, cambia completamente la respuesta de la planta, se reduce el tiempo de subida, pero el tiempo de establecimiento se ve aumentado drásticamente, en casi 5 veces el valor original, y se puede ver que aparece un sobre pico muy importante de más del 50%, no obstante, se puede ver que la planta llega al valor del set point, es decir, el error en estado estacionario se ve reducido.

Analizando la señal de control se puede ver que es menos alta que la del control proporcional en su valor más alto, pero que esta empieza en un valor de 0 y va aumentando de manera continua conforme el error se mantiene alto y luego a la inversa generando una oscilación que se va atenuando al paso del tiempo, pero esto no es un control PI completo.

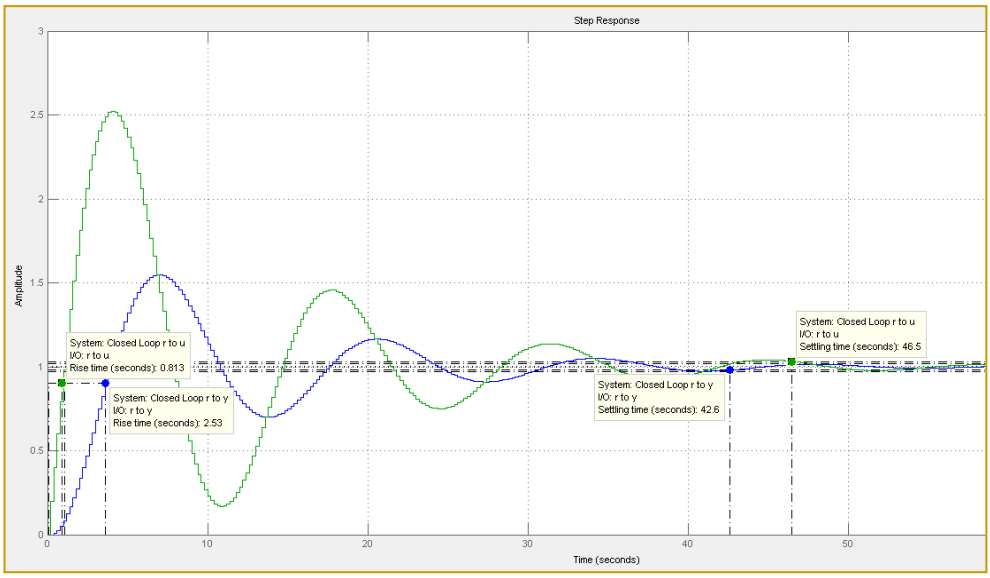


Figura 199 Comparación de graficas

Ahora deberá de ser agregado, un Zero real, lo cual completará la estructura del control PI, y analizarse la respuesta de este.

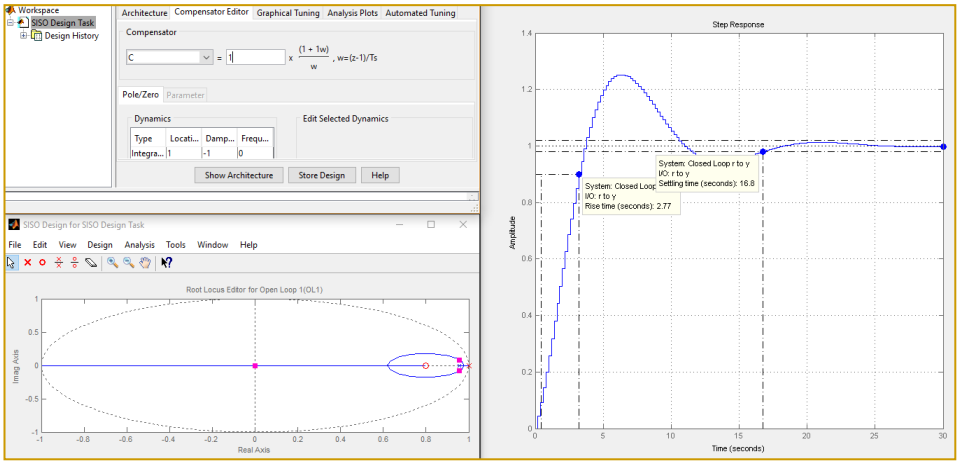


Figura 200 Control PI

Pues se puede ver que el sobre pico se reduce, así como el tiempo de establecimiento, lo cual indica que este control (PI), permite a diferencia del control proporcional, eliminar el error en estado estacionario, pero a costo de aumentar el tiempo de establecimiento y un sobre pico.
Antes de empezar a hacer cambios, se realizará una comparación de la señal de control PI del este sistema, respecto de la respuesta de la planta.

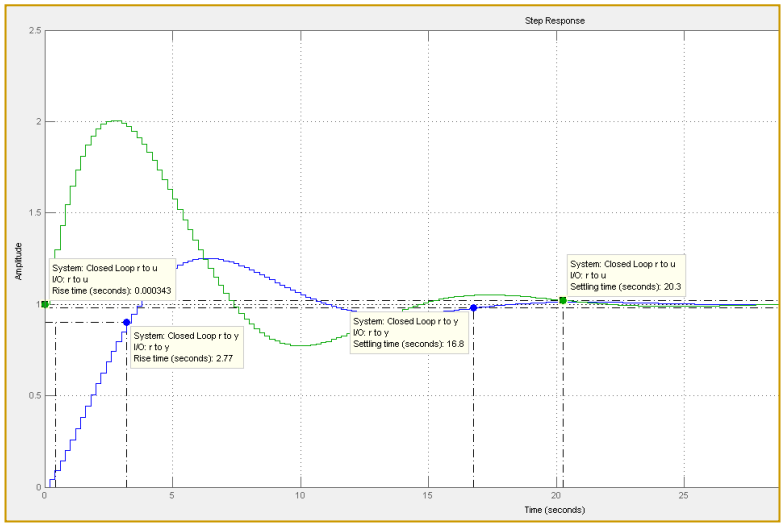


Figura 201 PI del sistema vs Respuesta de la planta

Al agregar el zero, se reduce el máximo pico de la señal de control respecto a la que se mostró cuando solo se tenía el integrador; Se pasará ahora a la ubicación del zero y el valor de la ganancia, buscando reducir el sobre pico y el tiempo de establecimiento de la planta, que son mayores a los que se tiene estando en lazo cerrado con ganancia unitaria (Un control Proporcional unitario).

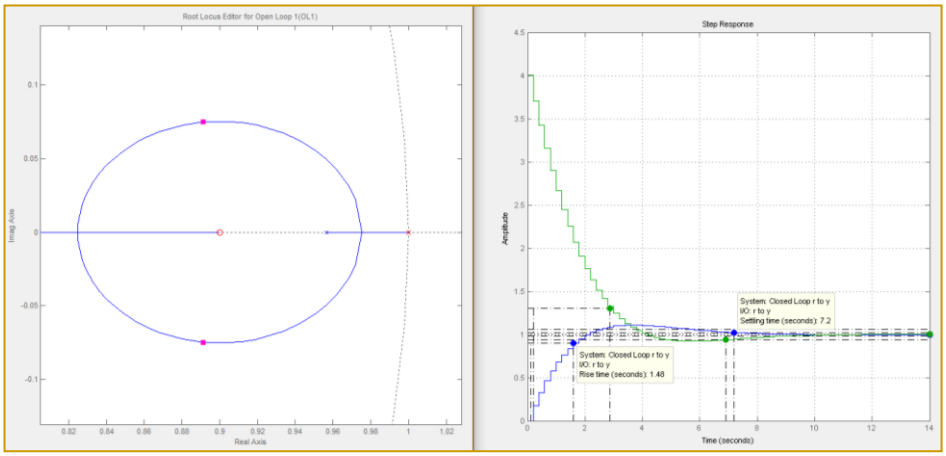


Figura 202 Control proporcional unitario

Con estos valores se obtuvieron los resultados mostrados; Como resultado se redujeron el tiempo de subida y el de establecimiento, a costo de un ligero sobre pico del 15% aproximadamente.
Utilizando los procesos aprendidos en "Valores del PID" se puede determinar: Kp = 4 y Ti = 2.

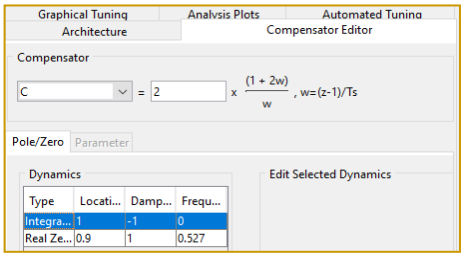


Figura 203 Cambio de valores de las ganancias

Se va a realizar una comparación final entre la señal con control PI vs la señal únicamente con ganancia unitaria en lazo cerrado.

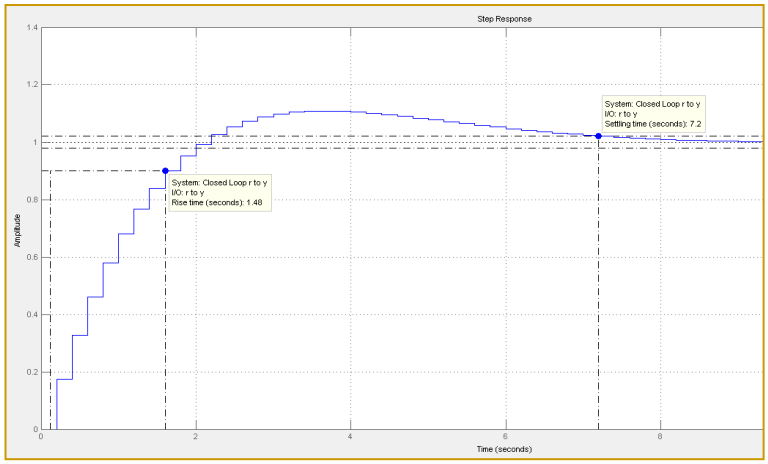


Figura 204 Señal de control P

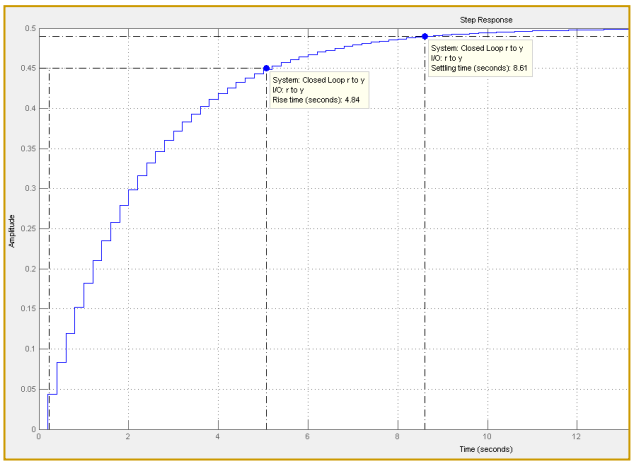


Figura 205 Señal ganancia unitaria

Ahora se obtendrá un control de forma automática por parte de sisotool bajo configuraciones por defecto cuando se selecciona un control PI para la planta y se comparará con los resultados obtenidos para encontrar las diferencias con el control que fue generado y determinar los puntos claves de la viabilidad de estos y cual resulta mejor para lo que se busca.

```
>> C

C =

  4 (z-0.9)
  ---------
    (z-1)

Sample time: 0.2 seconds
Discrete-time zero/pole/gain model.
```

```
>> C

C =

  0.6497 (z-0.8762)
  -----------------
         (z-1)

Sample time: 0.2 seconds
Discrete-time zero/pole/gain model.
```

Figura 206 Valor del compensador en sisotool

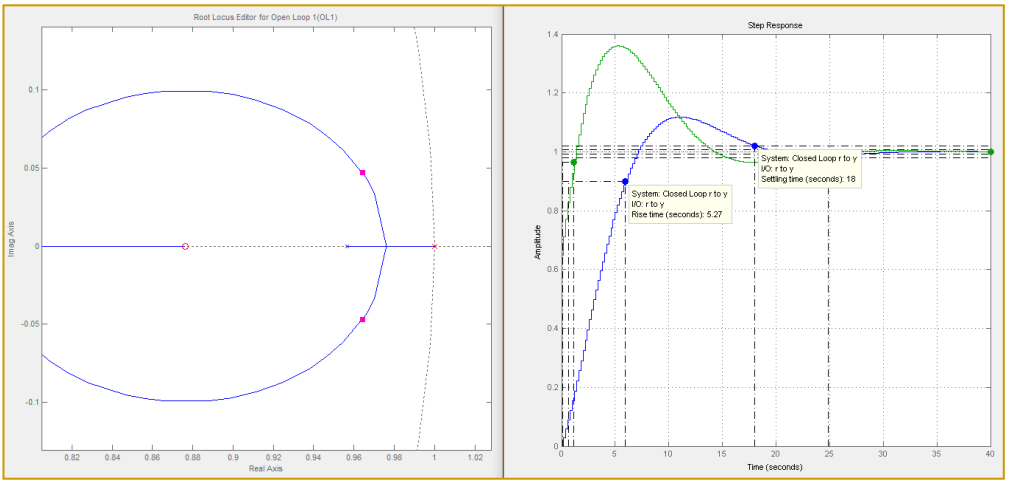


Figura 207 Control PI en sisotool

Si bien este proceso tiene una respuesta de control mucho más suave, el sistema es mucho más lento y con un sobre pico de valor similar, concluyendo con esto queda el realizar la simulación en LabVIEW para comparar las respuestas.

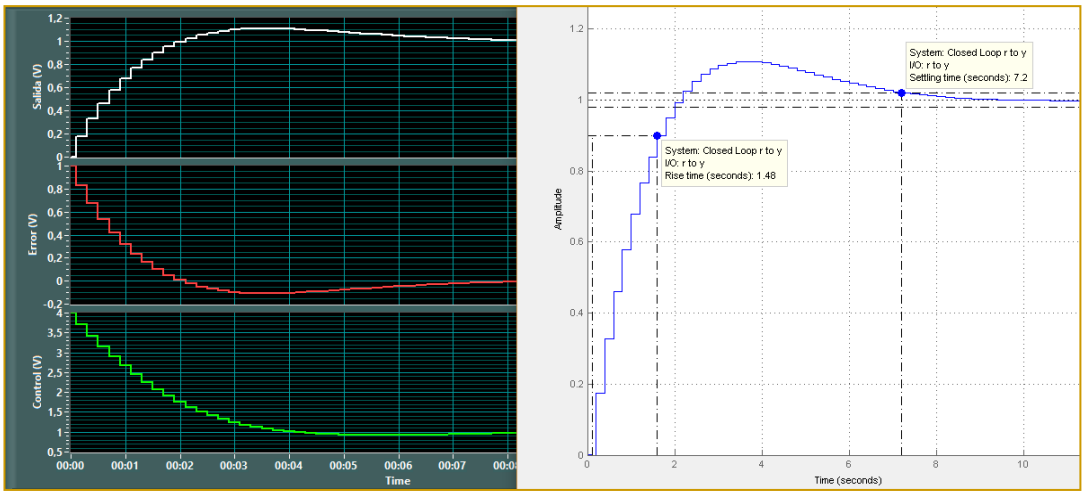


Figura 208 Simulación LabVIEW vs Sisotool.

Se obtiene el mismo resultado, de modo que los valores de Kp y Ti que se obtuvieron son los correctos y se puede proceder a probar el control en la planta real, para lo cual se va a detener y volver a iniciar el VI, luego se configurará el set point en 0 antes de dar al inicio, se da clic al botón de inicio, y se espera a que la planta este en 0 y el error también, para poder resetear el valor acumulado de compensación integral se presionará por un momento el botón de reste y posterior a esto se podrá cambiar a 1 el valor del set point y ver el controlador en acción.

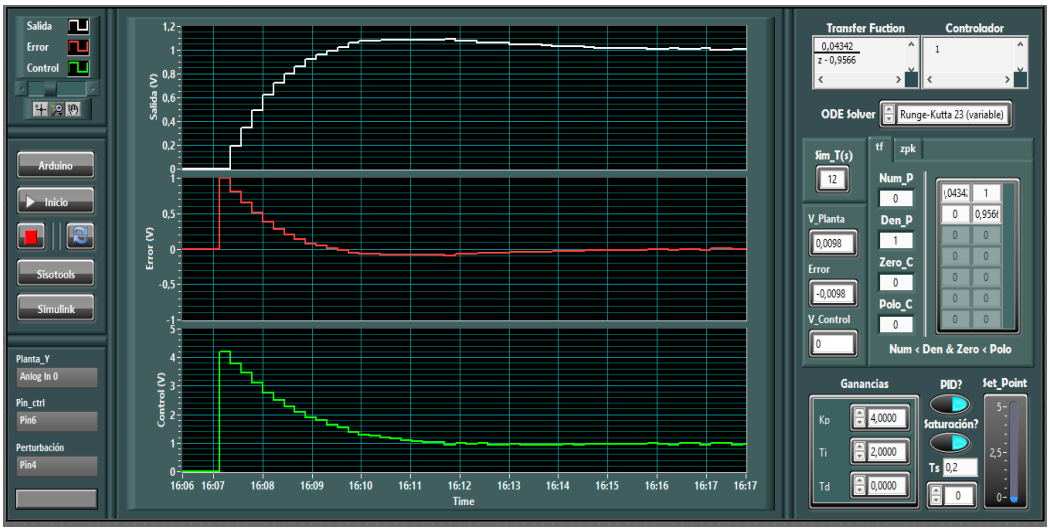


Figura 209 Planta R

4.5.5 PID Discreto de Planta Térmica

Para la demostración de este proceso se utilizará la planta térmica, dado que esta posee una sensibilidad más baja a los cambios en el tiempo de muestreo del control que se implemente en esta, debido a lo lento de su proceso de cambio.

Uno de los puntos fuertes de los controles discretos es el poder seleccionar de manera activa los lapsos de tiempos en los cuales se deben realizar los ajustes a un sistema; debido a esto se decidió mostrar un mismo control PID siendo utilizado con diferentes periodos de ajustes.

El control PID utilizado fue el siguiente.

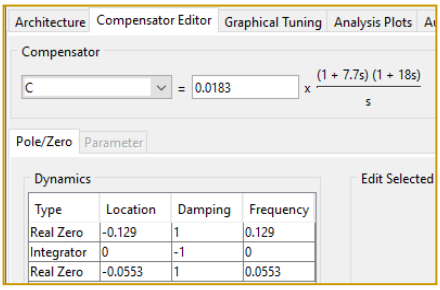


Figura 210 Ganancias del control PID

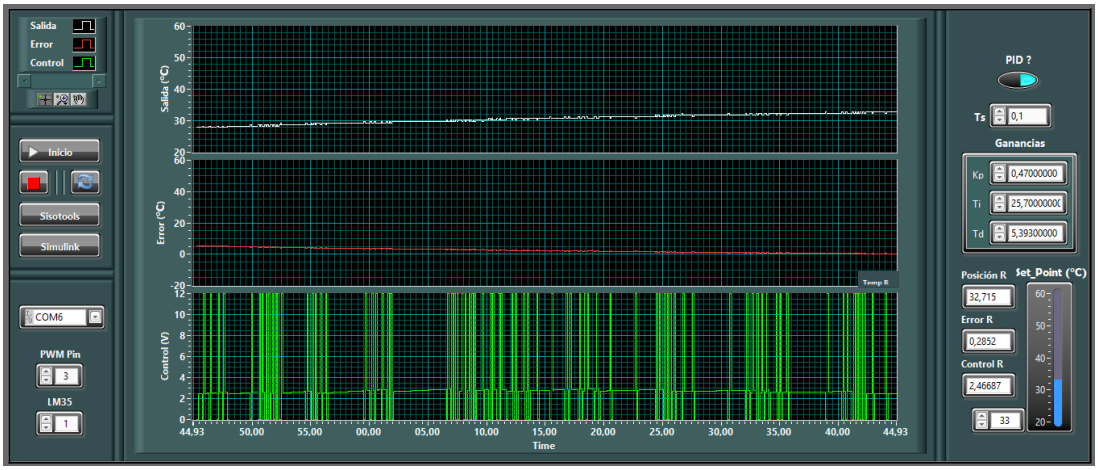


Figura 211 PID con un tiempo de ejecución de 0.1 s

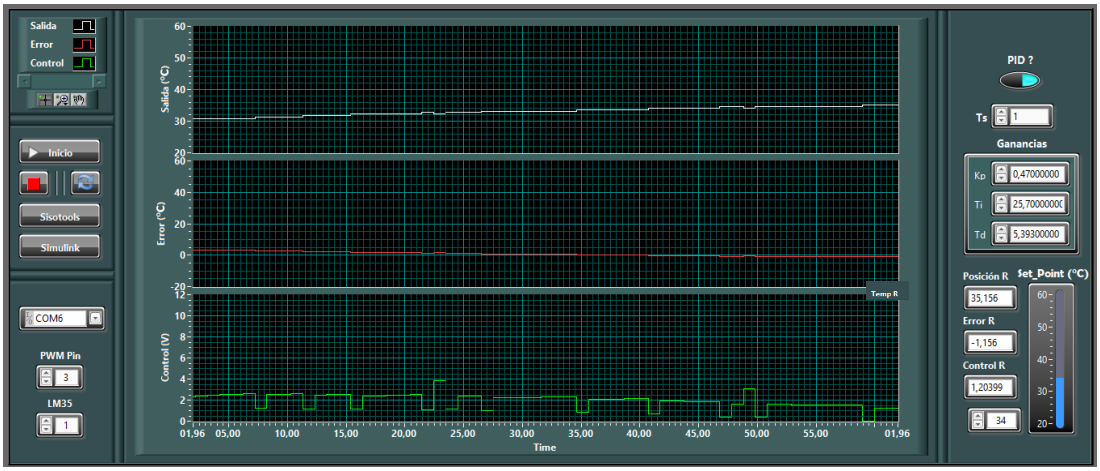


Figura 212 PID con un tiempo de ejecución de 1 s

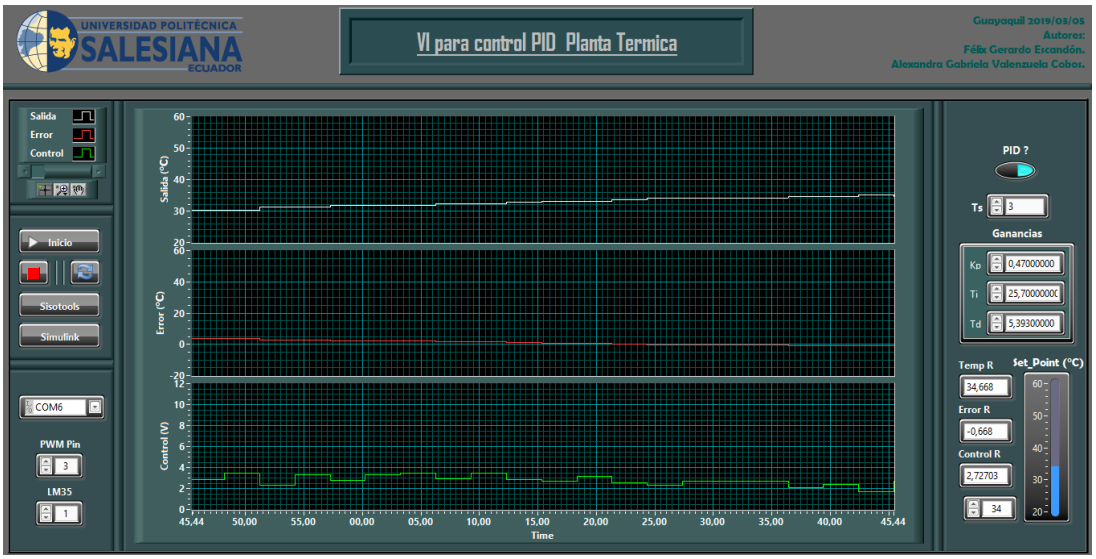


Figura 213 PID con un tiempo de ejecución de 3 s

Como puede observarse, al variar la variable Ts se consigue una señal de control diferente, pero con el mismo patrón, solo que esta evalúa el estado de la planta en un tiempo Ts, con ello según el control y sistema a controlar se puede conseguir sistemas más estables con variaciones en su entrada de menor frecuencia en su variación.

4.6 Práctica#6 “Levitador Eólico”

4.6.1 Objetivos

- Revisar y comprender el funcionamiento del VI personalizado para el modelado de la planta de levitación eólica.
- Obtener conjuntos de muestras para realizar el modelado de la planta.
- Revisar y comprender el funcionamiento del VI personalizado para el control PID de la planta de levitación eólica.
- Obtener varios modelos no lineales para la planta.
- Realizar la linealización de los modelos no lineales.
- Hacer las simulaciones del comportamiento de la planta.
- Realizar diferentes controles utilizando los modelos linealizados en sisotool.
- Comparar los resultados en las simulaciones con los reales.

4.6.2 VI modelado LE

Al ejecutar el VI modelado LE se abrirá la ventana que se muestra a continuación, con los controles para configurar el proceso de muestreo.

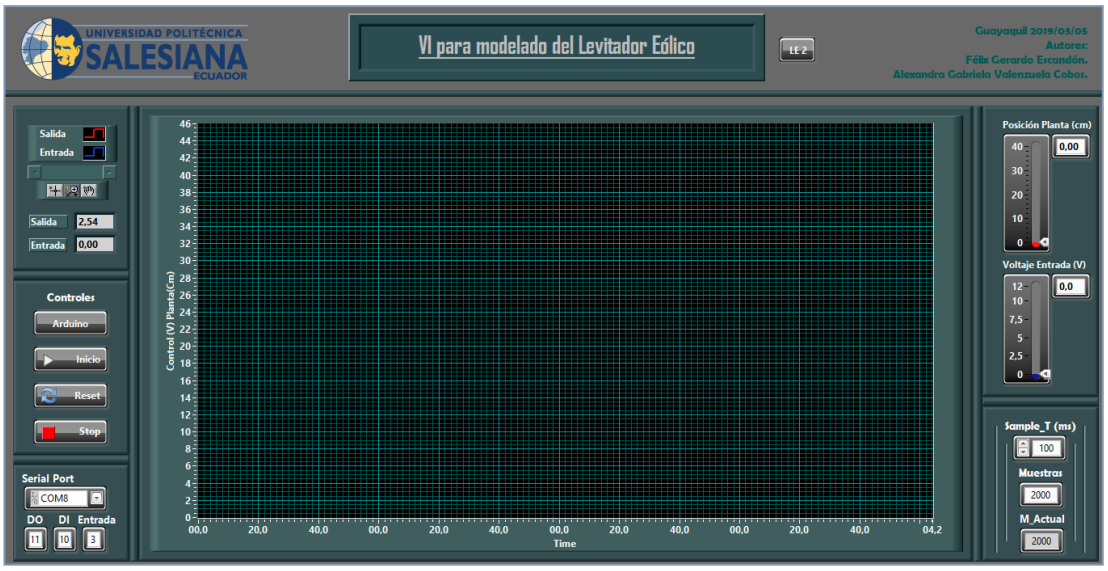


Figura 214 Front Panel VI Levitador Eólico

En el centro se grafica la señal de Entrada de la planta y la de Salida, la barra de tiempo se encuentra en segundos y milisegundos para ayudar al análisis y junto al encabezado del VI se cuenta con un selector para configurar si se utilizará la Planta 1 o 2.

Del lado derecho se encuentran los controles Posición Planta y Voltaje Entrada que corresponden a la posición de la planta al realizar el modelado en modo de simulación y al valor de voltaje de entrada que se enviará la planta, debajo de estos se encuentran los controles de tiempo de muestreo y numero de muestras.

Del lado izquierdo se encuentran los controles para los gráficos del Chart central y debajo de estos los que controlan la secuencia del proceso, empezando por el botón de nombre Arduino el cual cambia entre el modo simulado y el uso de Arduino conectado a la planta real, siguiendo con el botón Inicio para iniciar el proceso de toma de muestras, el botón Reset que permite guardar los cambios hechos al archivo Datos.mat y reiniciar el proceso de toma de muestras hasta antes de dar clic en el botón de Inicio, y finalmente el botón de Stop que detiene el proceso.

Debajo de estos se encuentran los controles de configuración para las conexiones con el Arduino, empezando por el listado de puertos COM en el que se selecciona el puerto COM actual al que se encuentra conectado el Arduino, luego de estos se tiene los pines correspondientes a la señal PWM de control del Arduino hacia la planta y los pines de Echo (Di) y Trigger (Do) del sensor ultrasónico.

4.6.3 Programación interna del VI de modelado LE

Los componentes claves del proceso de modelado son los elementos provistos por MakerHub del paquete de LINX los cuales permiten a LabVIEW controlar, utilizando el Arduino, elementos que trabajan por debajo de su velocidad de procesamiento de 1ms, en este caso el sensor ultrasónico trabaja en el orden de los picosegundos.

Debido a que esto se realiza al inicio del programa la configuración necesaria para el uso de esta librería, dentro del código se realiza el seteado inicial por defecto de los elementos utilizados para el proceso de secuencia y remodelado, para que sea asegurado los diferentes conjuntos de muestras que se obtengan.

Seguido de esto se abre la estructura While Loop en la cual se ejecutará el proceso, dentro de esta se encuentra una estructura Flat Sequence encargada del realizar de manera secuencial el proceso y permitiendo junto con el lazo while la realización de un siclo repetitivo controlado que empieza solicitando el cambio de estado del botón de Inicio.

Acto seguido se puede observar un segundo While Loop en el cual se realiza la toma de muestras, regido por un temporizador Elapsed Time que recibe el resultado en segundos de los milisegundos divididos para 1000 con lo cual activa cada vez que pase ese valor de tiempo el bloque de secuencia donde se escriben en un archivo .lvm los valores medidos de entrada y salida de la planta y aumentando el contador de muestras tomadas ubicado dentro de la estructura Case Structure ubicada hacia la derecha del lazo while.

En la esquina superior izquierda se encuentra otra estructura Case Structure que tiene por función el realizar la toma de muestras de la planta y el seteo de los valores de entrada a la planta por parte del Arduino utilizando los elementos de la librería LINX y los demás controles antes descritos.

Se realiza un acondicionamiento del valor medido por el sensor para así se ajuste a la medida utilizada en el tubo para visualizar el efecto del controlador, para ello se hace uso de un subvi con que contiene la ecuación de la recta.

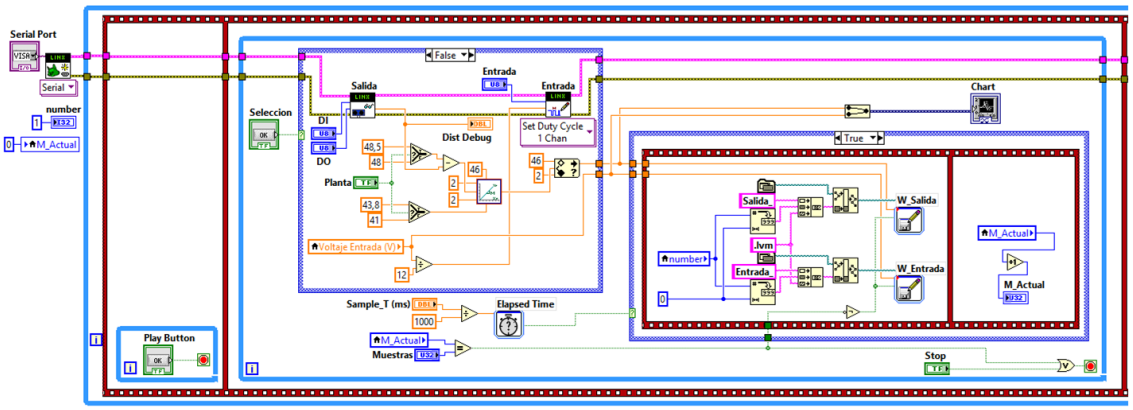


Figura 215 Block Diagram parte 1

Continuando la secuencia se realiza la lectura de los archivos .lvm creados y la inserción de es tos en el archivo Datos.mat seguido de la ejecución del comando ident, ambos procesos son realizados utilizando la estructura Matlab script con la condición de que el botón de paro no esté activado, luego de esto se llega a un nuevo punto de espera el cual nos permite trabajar en ident los diferentes modelos y exportarlos al archivo Datos.mat antes de que se guarden los cambios realizados.

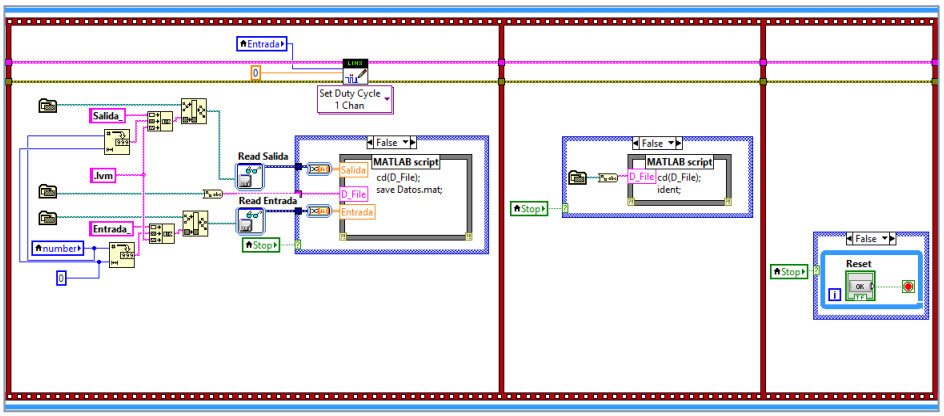


Figura 216 Block Diagram parte 2

Una vez se finalizan los procesos de modelado y se presiona el botón de Reset, se produce en la secuencia siguiente el guardado de los cambios realizados y permite volver a empezar la secuencia nuevamente desde la solicitud del botón de Inicio.

Fuera del lazo while se realiza el cierre de la conexión entre LabVIEW y Arduino con el cierre de las herramientas LINX.

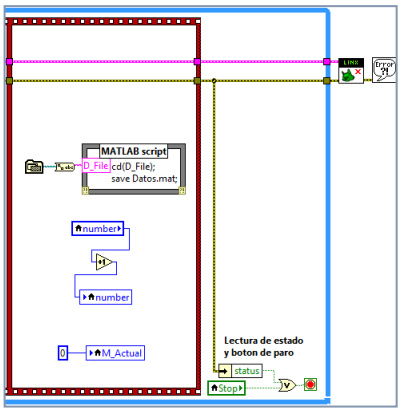
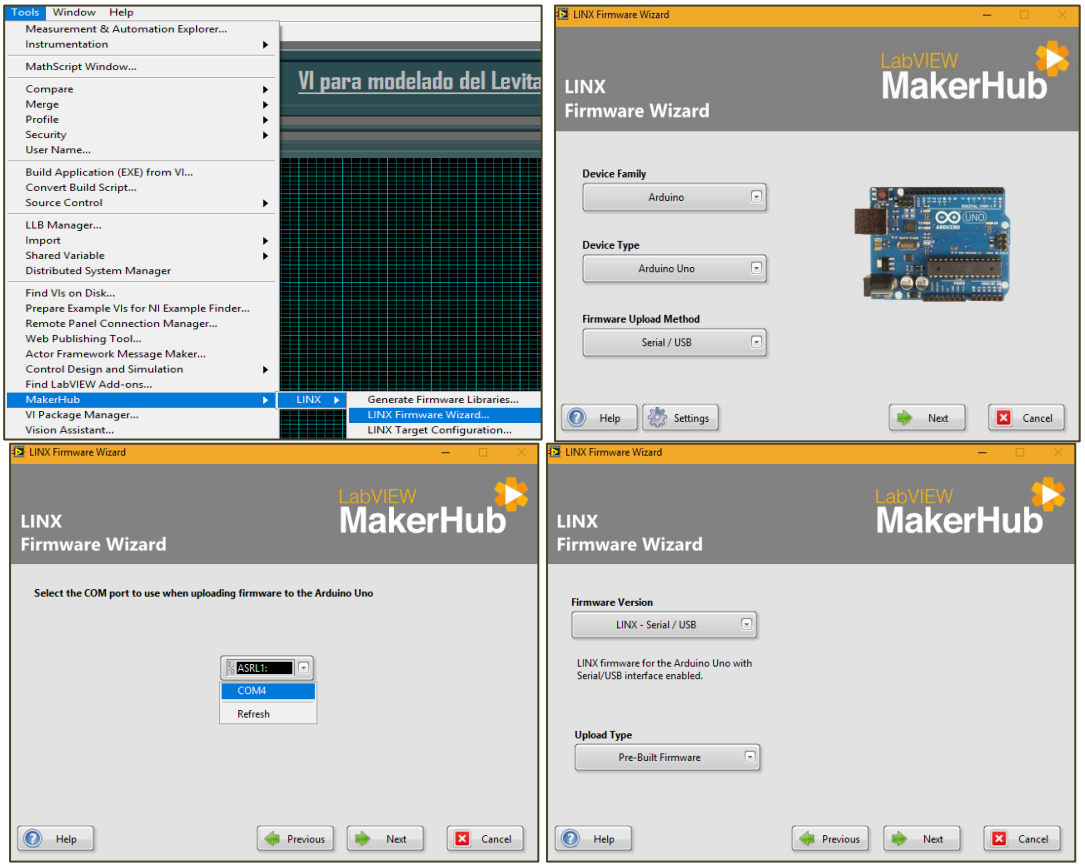


Figura 217 Block Diagram parte 3

4.6.4 Toma de muestras de planta Levitador Eólico.

Se empezará realizando la programación de Arduino utilizando el Wizard de MakerHub tal como se muestra en la siguiente secuencia de imágenes, con el fin de poder realizar la comunicación entre LabVIEW y Arduino, los valores que se muestran pueden variar de computador en computador, así como con el uso de un tipo diferente de Arduino



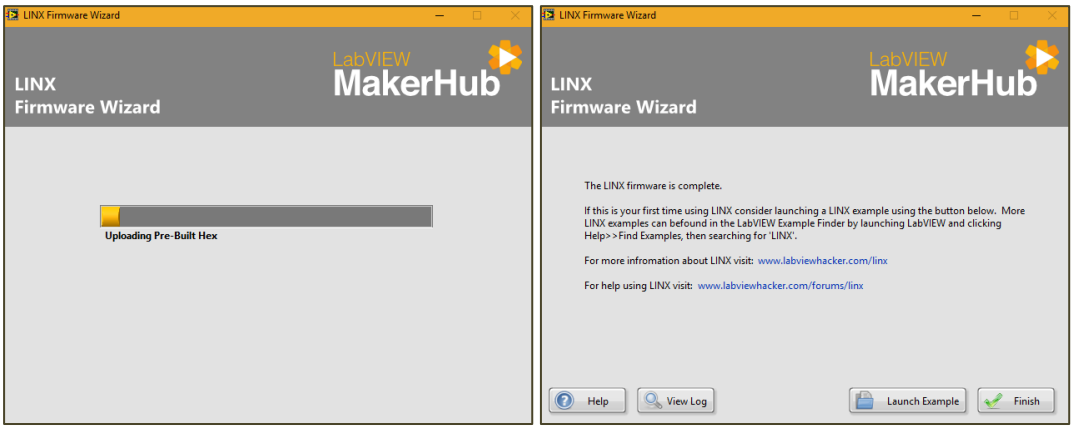


Figura 218Pasos para la comunicación entra LabVIEW y Arduino

Luego de esta configuración se realizará la selección del puerto COM en la ventana principal del VI según corresponda y asegurándonos de que sea el mismo que se usó en la programación del Arduino desde el Wizard.

Luego de esto y de realizar las conexiones según corresponda, se puede proceder con la recopilación de muestras de la planta funcionando a diferentes niveles de tensión y en tiempos diferentes.

Figura 219 Comportamiento de la planta

Una vez se termine la toma de muestras según el tiempo de muestreo y el número de muestras luego de un momento se abrirá automáticamente la ventana de ident en donde se podrá proceder a la importación de los datos que se acaban de recopilar.
Es posible el agregar datos adicionales para dar mayor riqueza a los gráficos en términos de variables al acceder a las opciones adicionales.
Se recomienda tomar más de un conjunto de muestras para poder realizar comparaciones y tener una mayor cantidad de información con la cual realizar los modelados.

En el caso de esta demostración, fueron realizadas dos recopilaciones tal como se muestra, es importante tomar en cuenta que gracias al diseño del VI, es posible obtener un segundo conjunto de muestras seguido de importar el primer conjunto al presionar reste y luego inicio, para volver a importar de la misma manera el nuevo conjunto de muestras recopilado.

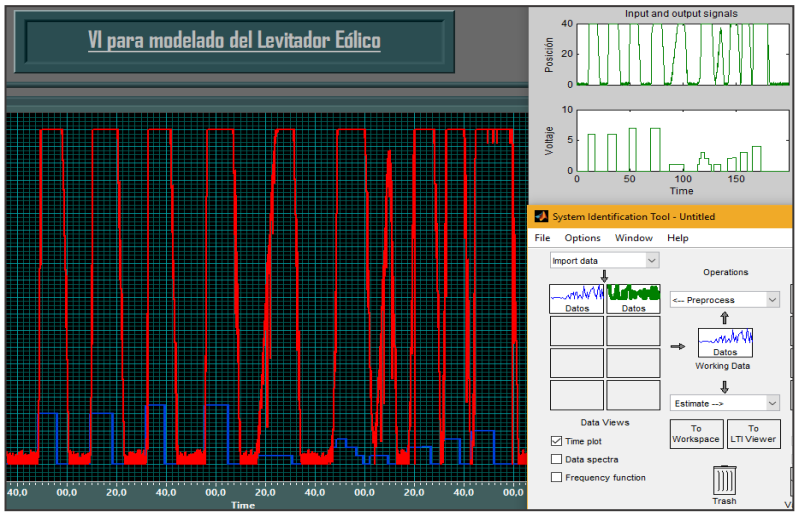


Figura 220 Primera recolección de datos

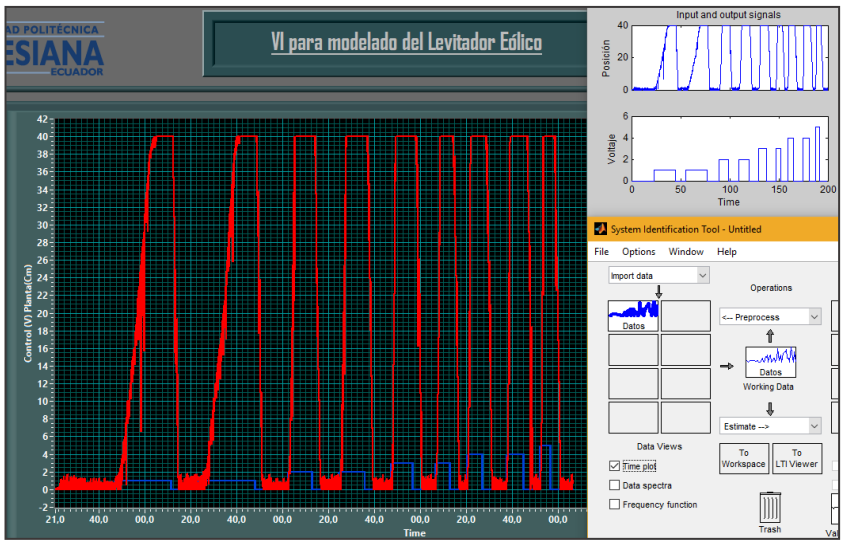


Figura 221 Segunda recolección de datos

No obstante, es posible que durante la toma de muestras se produzcan errores, como cuando se interfiere la medición entre otras tantas, para esto es importante mantenerse atento y ubicar estos momentos para evitar usarlos durante el modelado, ya que serán motivo de errores e imprecisiones.

Se recomienda también se guarde el archivo de Ident a partir de este punto en adelante, para obtener mayor libertar para el resto del proceso.

4.6.5 Realización de modelados varios.

Para el proceso de modelado se utilizará ident, de manera adicional se contará con una ventana de comandos de Matlab con el cual se podrán utilizar los comandos de linap, pero hay un detalle importante a tomar en cuenta, esto es que tal como se encuentran los dos conjuntos de muestras, no será posible realizar un modelado optimo, por ello utilizar Matlab en su totalidad resulta mucho más efectivo para trabajar las variables y comandos a utilizar.

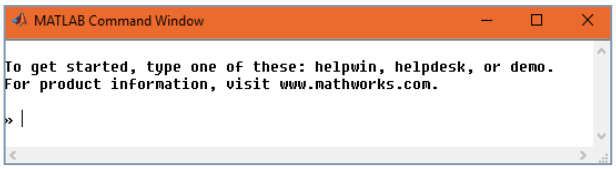


Figura 222 Comand Windows

Como paso inicial se realizará un pulido de las muestras, para esto se utilizará filtrados y acondicionamientos manuales de las muestras, con el objetivo de volverlas lo más ideales posibles.

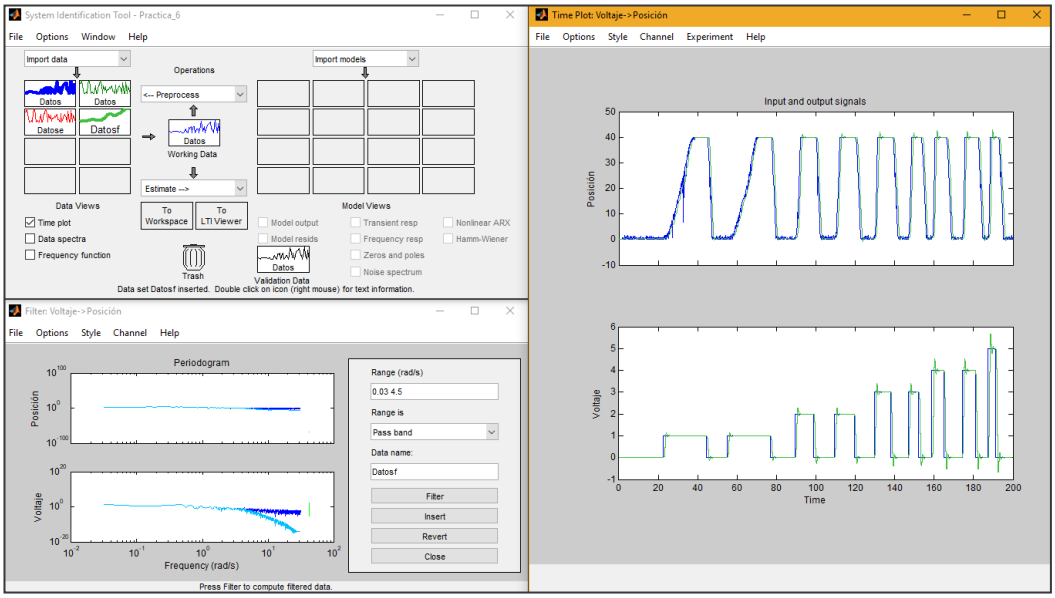


Figura 223 Filtrado de señales

Cuando se tenga el conjunto de muestras filtrado, se puede hacer un ajuste manual para compensar diferentes distorsiones colaterales provocados por el filtrado, de manera manual, utilizando Matlab para poder comparar los resultados entre si, tal como se ve en la imagen siguiente.

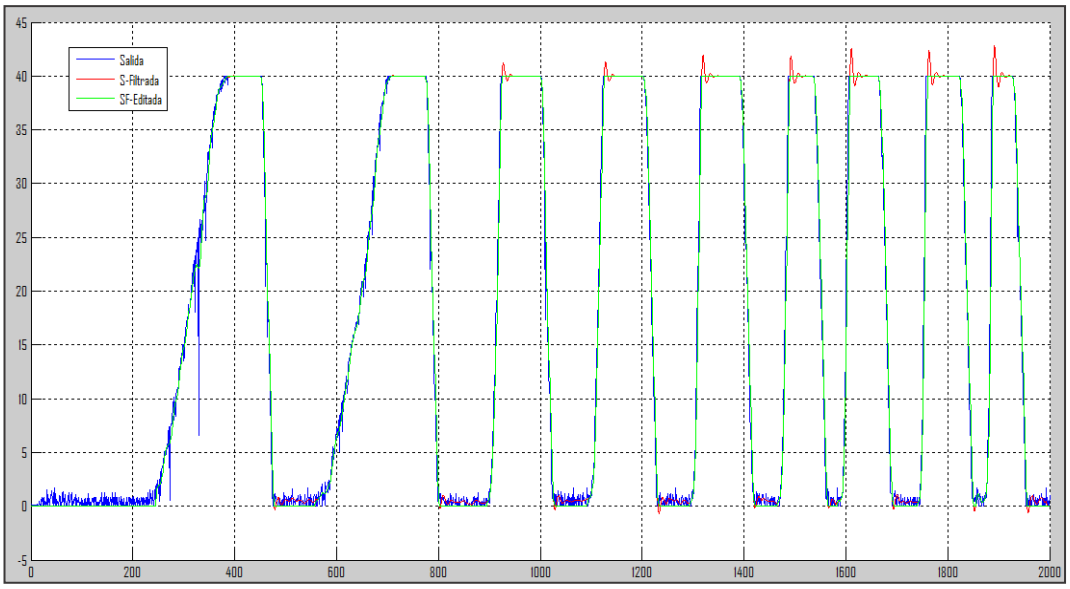


Figura 224 Comparación de resultados

Una vez se tiene ya la señal editada a lo más optimo posible se puede proceder a sacar los diferentes modelados, los resultados de estos dependen únicamente de la habilidad del modelador y la riqueza de los datos con los que se realice el modelado, así como la planta que se desea modelar.

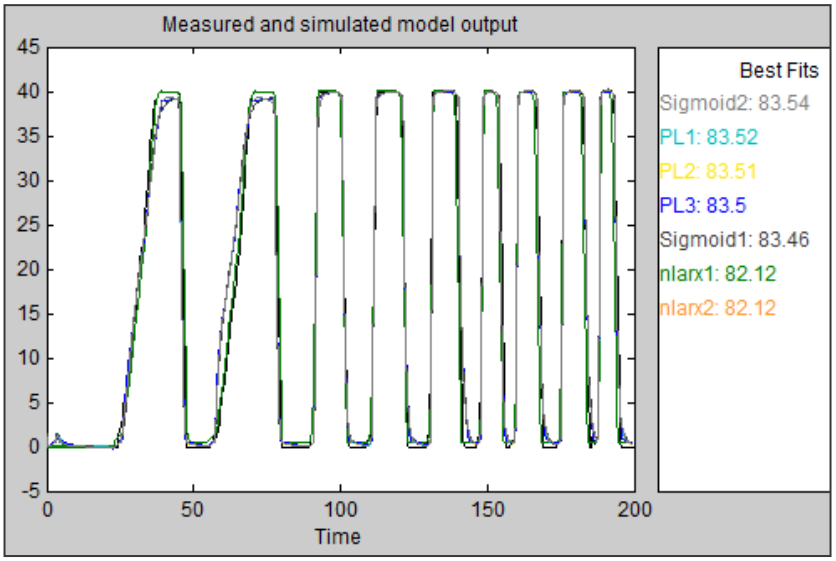


Figura 225 Seleccionar la aproximación más exacta

En la imagen anterior se puede apreciar diferentes modelos realizados al conjunto de muestras obtenidos de la planta, todos estos imitan el comportamiento muestreado de la planta con más del 80% de aproximación, con lo cual, aunque no sean exactos, se aproximan lo suficiente para realizar una simulación del comportamiento de la planta.

No todas las opciones con las que se cuentan son igualmente viables, y no siempre se llega a un valor alto de aproximación a la primera, por ello se recomienda experimentar de manera metódica y sistemática tomando en cuenta varios valores diferentes en cada opción, así como varias combinaciones de estas e ir anotando las realizadas para determinar los mejores valores y combinaciones.

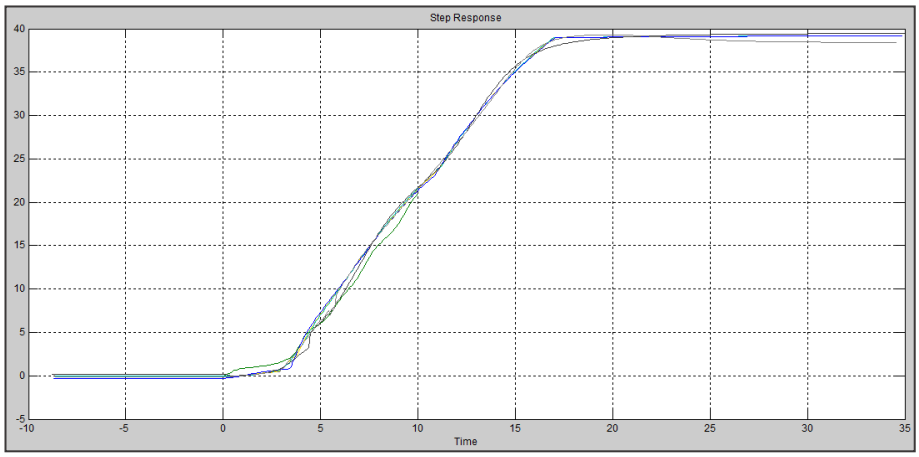


Figura 226 Respuesta con Step

Luego de ver los valores de aproximación, se recomienda también ver la respuesta al paso de los diferentes modelos ya que en esta se puede apreciar el comportamiento de los modelos al recibir valores concretos de entrada, que de manera siguiente pueden ser comparados con los datos de la planta real.

4.6.6 Linealización y simulación de comportamiento

El siguiente paso del proceso es la linealización de los modelos no lineales obtenidos, esto es con el objetivo de tener un modelo lineal con el cual sea posible realizar las simulaciones en sisotool, ya que este solo admite sistemas lineales para sus simulaciones.

Esta parte del proceso requiere simular la anterior parte, la eficiencia del resultado depende del modelador y del modelo a linealizar.

Para esta parte del proceso se va a hacer selección de tres de los modelos que se obtuvieron en el paso anterior y se buscara el modelo lineal que mejor se acerque a la respuesta que tendría la planta real, teniendo en cuenta como criterio de selección, el tiempo que le toma al sistema lineal llegar a 40cm.

```
>> G1 = linapp(PL1,0,4.8695,50)

G1 =
Discrete-time OE model:  y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = 0.087 z^-1 + 0.1543 z^-2

  F(z) = 1 - 1.693 z^-1 + 0.5338 z^-2 + 0.1749 z^-3

Name: PL1
Sample time: 0.1 seconds

Parameterization:
   Polynomial orders:   nb=2   nf=3   nk=1
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for parameters

Status:
Model modified after estimation.
>> step(G1)
>> G3 = linapp(Sigmoid1,0,4.58,50)

G3 =
Discrete-time OE model:  y(t) = [B(z)/F(z)]u(t) + e(t)
  B(z) = 0.07581 z^-1 + 0.02271 z^-2

  F(z) = 1 - 2.624 z^-1 + 2.325 z^-2 - 0.6955 z^-3

Name: Sigmoid1
Sample time: 0.1 seconds

Parameterization:
   Polynomial orders:   nb=2   nf=3   nk=1
   Number of free coefficients: 5
   Use "polydata", "getpvec", "getcov" for parameters

Status:
Model modified after estimation.
>> step(G3)
>> G5 = linapp(nlarx1,0,40,500)

G5 =
Discrete-time ARX model:  A(z)y(t) = B(z)u(t) + e(t)
  A(z) = 1 - 0.8288 z^-1

  B(z) = 4.217 z^-1

Name: nlarx1
Sample time: 0.1 seconds

Parameterization:
   Polynomial orders:   na=1   nb=1   nk=1
   Number of free coefficients: 2
   Use "polydata", "getpvec", "getcov" for parameters

Status:
Model modified after estimation.
>> step(G5)
```

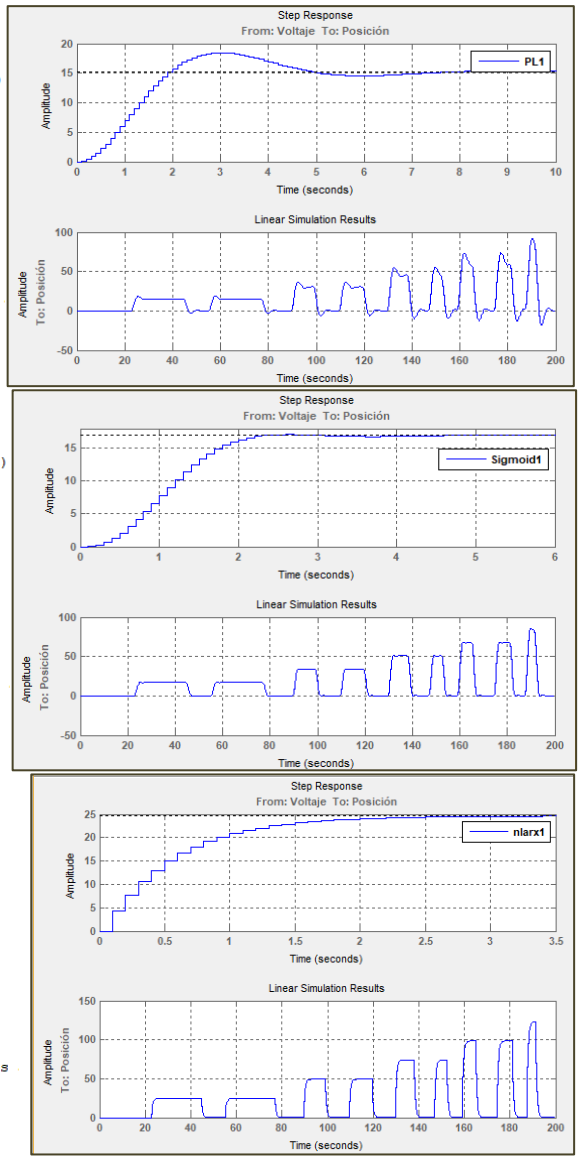


Figura 227 Diferentes modelos

Se recomienda durante este paso, el ir separando los modelos que tengan un resultado favorable según el criterio del modelador, otra de las opciones que se tiene son las de realizar una simulación lineal para saber la respuesta del modelo lineal o en su defecto utilizar sisotool para comparar la respuesta a un control proporcional con una K obtenida de la comparación del error inicial y el rango de voltaje de control del que se dispone

4.6.7 VI Práctica PID de planta LE

En el caso del VI para PID de la planta Levitador Eólico se tiene la siguiente ventana principal la cual guarda similitud con las vistas en anteriores prácticas, pero está más destinada a el uso exclusivo de la planta para la que se le adapto.

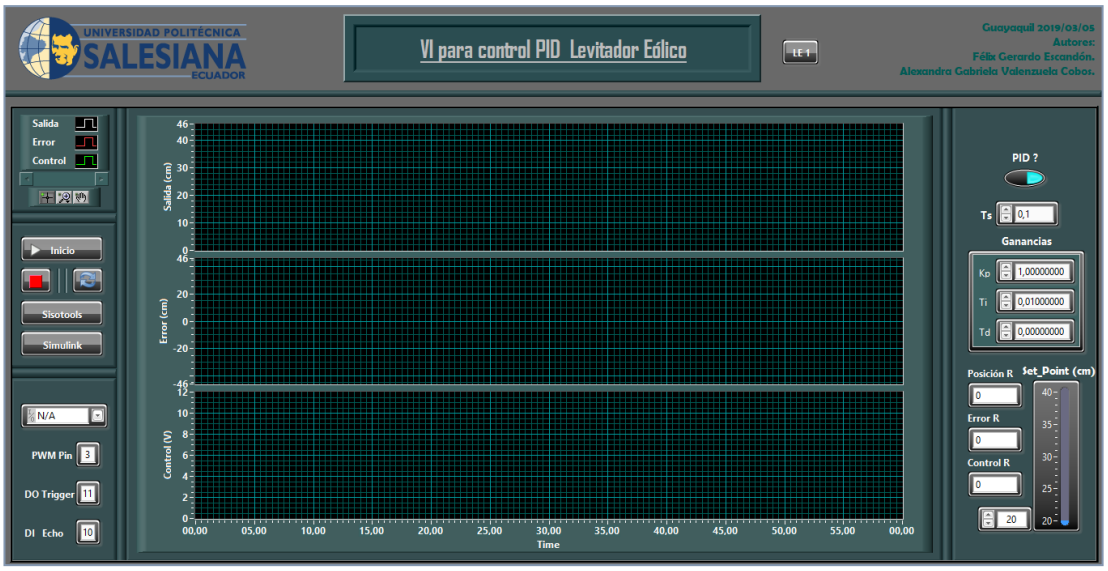


Figura 228 Front Panel VI para control PID Levitador Eólico

En el centro se grafican las señales de Salida, Error y Control vs el tiempo en segundos y milisegundos y junto al encabezado del VI se cuenta con un selector para configurar si se utilizará la Planta 1 o 2.

Los botones del lado izquierdo corresponden a la secuencia de ejecución de manera similar a la de modelado con la diferencia de que, en este caso, el botón de reset cumple la función de devolver a 0 los valores acumulados del error integral, debajo de estos se encuentran los controles para las conexiones entre Labview y Arduino.

Estos controles de la derecha empiezan con un Botón que activa y desactiva las constantes del PID ingresadas en el bloque de variables de Ganancias del PID, en medio de estos se encuentra el tiempo de muestreo a usar y debajo de estos tres controles se encuentran los indicadores de los valores finales del gráfico y el Set Point que está limitado entre los valores de 20 y 40.

4.6.8 Programación interna del VI Práctica PID de la planta Levitadora

El código empieza con un lazo while general que agrupa todo el código, dentro de este se encuentra una estructura secuencial que en su primera casilla tiene programada una pausa con el extra de poder realizar la ejecución de los códigos de Sisotool y Simulink incluyendo en el proceso la apertura de los datos del archivo Datos.mat durante la espera del cambio de señal en el botón de Inicio.

En la casilla siguiente de esta se empieza con la configuración inicial de los valores correspondientes a la señal de entrada de la planta, la configuración

del puerto COM y apertura de las herramientas de LINX y los valores iniciales del Error, Error integrado y Variable controlada volviéndolos 0.

Luego de esto se ve el inicio del lazo while en el que se genera la secuencia encargada de realizar el control PID según sea configurado, se realiza primero la lectura del valor de la planta y se acopla su valor para que este dentro del rango de control que se tiene para reducir el valor de ruido en los estados de inicio y saturación de la planta.

Siguiendo con la secuencia se realiza una separación de la señal acoplada medida de la planta para que sea graficada y obtener de esta el error y la diferencia entre la señal actual y la anterior, así como el valor de la variable controlada que es el valor de la planta actual que se vuelve el valor anterior para cuando se genere la siguiente iteración.

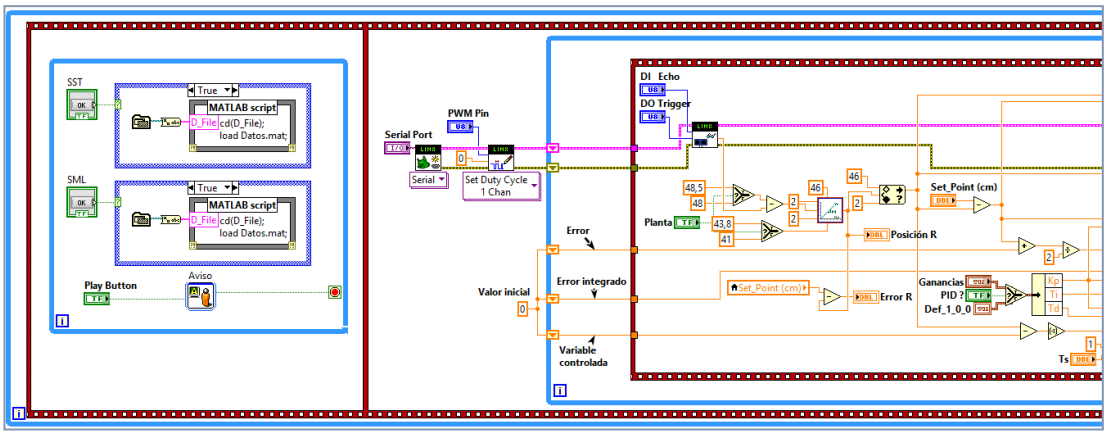


Figura 229 Block Diagram parte 1

De igual forma puede verse la selección entre el uso de los valores de Ganancias PID y los de un conjunto por defecto que contiene [1, 0, 0] como valores, y la consecuente subdivisión por nombre de estos, para pasar a generar las operaciones indicadas para los controles PID en términos de Kp Ki y Kd.

Al final de estos procesos se realiza la saturación de la señal de control dentro de un rango de 0 a 12 para posterior a esto realizar su división al total para tener un valor entre 0 y 1 correspondiente al ancho del ciclo positivo de la señal PWM de control.

En el bloque que continua se realiza la pausa según el tiempo de muestreo para el proceso de control PID discreto que se requiere usar por el tipo de modelado que se realiza.

Del otro lado del lazo while se puede ver la condición negada que se utiliza para mantener o volver a 0 los valores acumulados del error integral, fuera de este lazo while se realiza el cierre de las herramientas LINX y se termina el proceso de control PID.

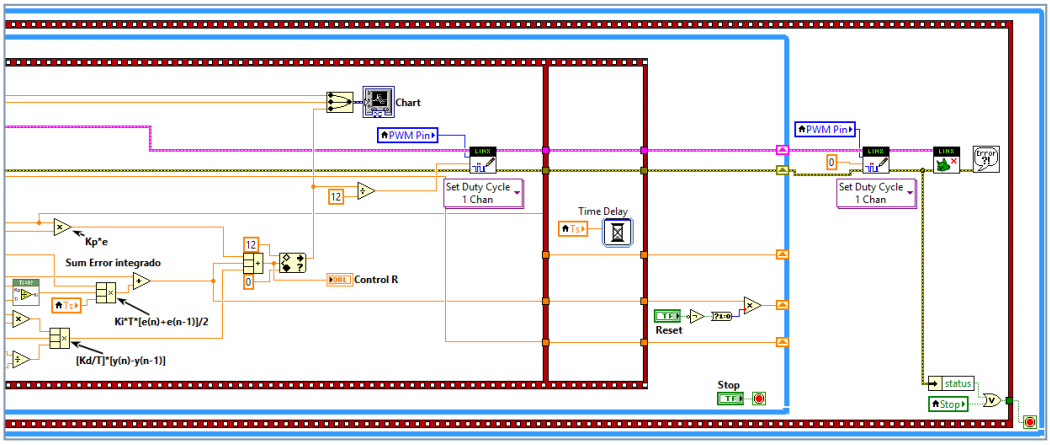


Figura 230 Block Diagram parte 2

4.6.9 Realización de controles PID

Como primer control se tendrá el que se usó a modo de comprobación, el cual constaba de un control proporcional de 0.1, luego deberá ser realizado los controles PI y PID cuyos valores se podrán modificar con la ayuda de sisotool para ver a que rango de valores es posible acceder teniendo en cuenta el rango de voltajes de 0 a 12 voltios con los que se alimenta de forma efectiva la planta.

Dentro de las opciones que se tiene también se encuentra la de generar controles de manera automática y utilizar las herramientas de sisotool para ajustar sus valores de manera manual e ir viendo cómo cambia la respuesta No siempre el control PID completo es el mejor para una planta, cada planta responde de mejor o peor forma a los diferentes controles posibles

4.6.10 Control P

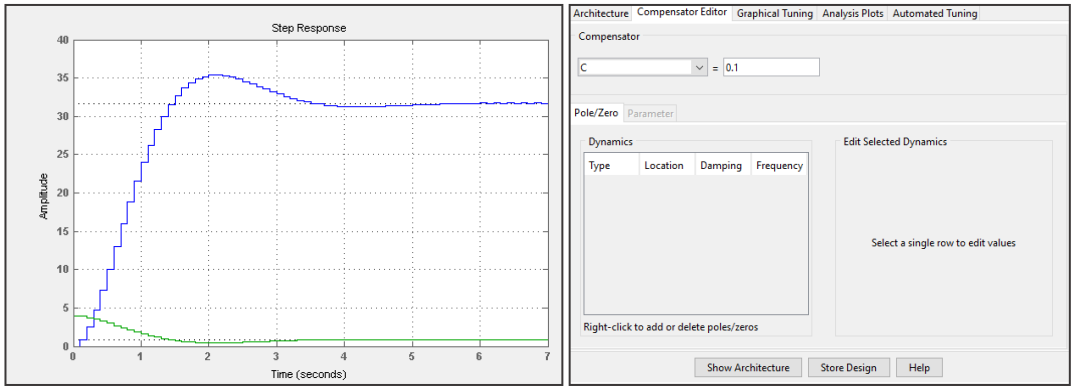


Figura 231 Control P

4.6.11 Control PI

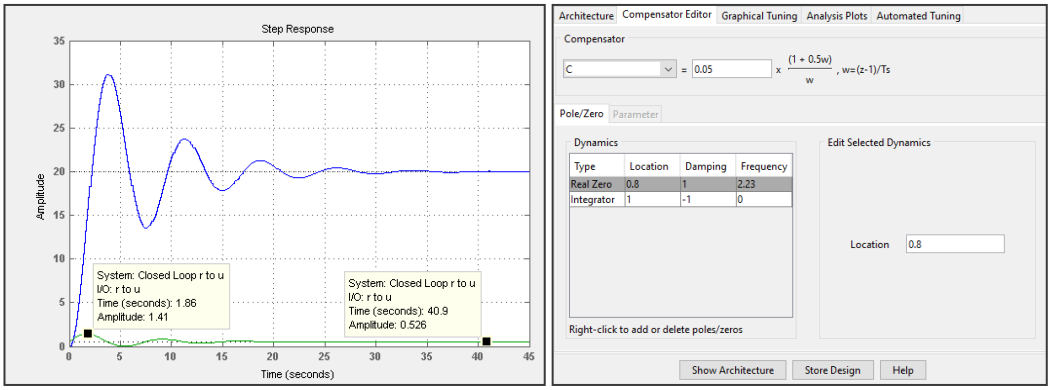


Figura 232 Control PI

4.6.12 Control PID

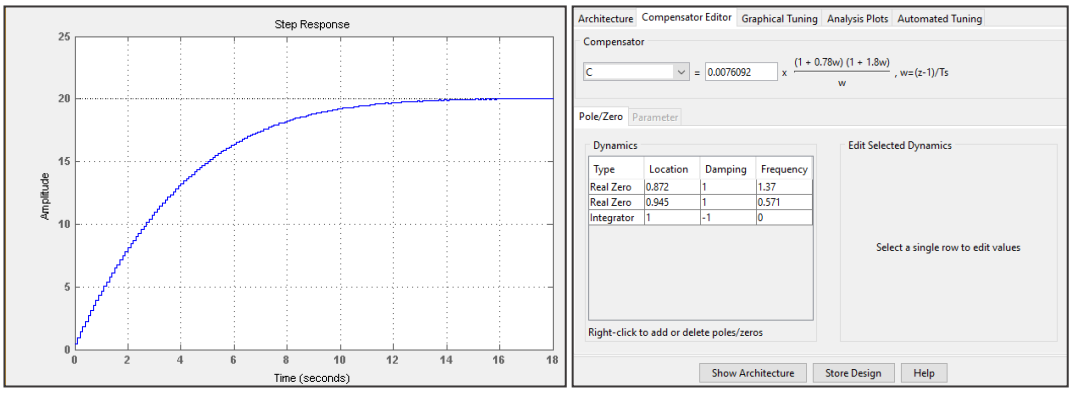


Figura 233 Control PID

Como demostración de este proceso, a continuación, se adjuntan capturas del proceso completo de principio a fin con la implementación de un control PID en la "Planta Levitadora #2"

Se empezó obteniendo el conjunto de muestras con el que se obtendrá el modelo con ident.

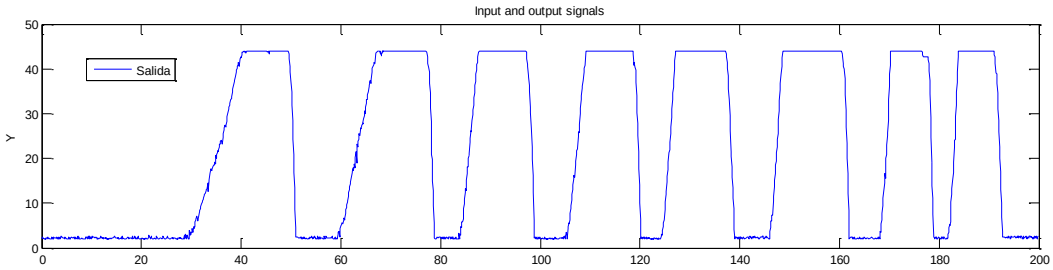


Figura 234 Señal de Salida

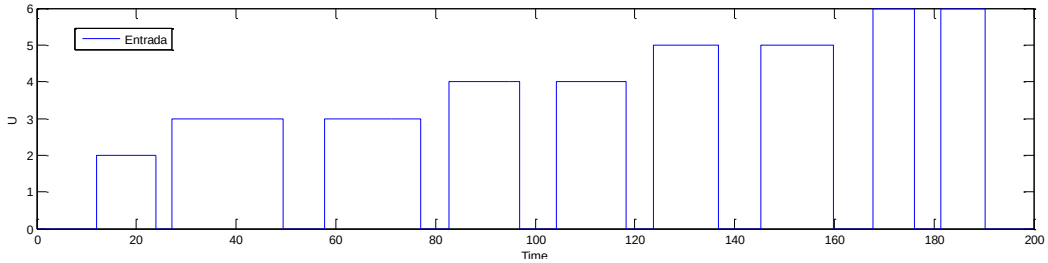


Figura 235 Señal de Entrada

Luego se ingresó estos datos en ident y se procedió a un primer intento de modelado con las muestras sin modificar para ver el resultado, para esto se utilizó un modelado no lineal.

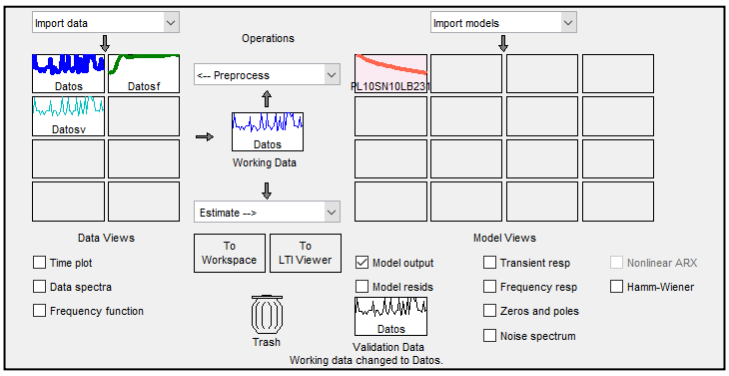


Figura 236 Ventana Ident

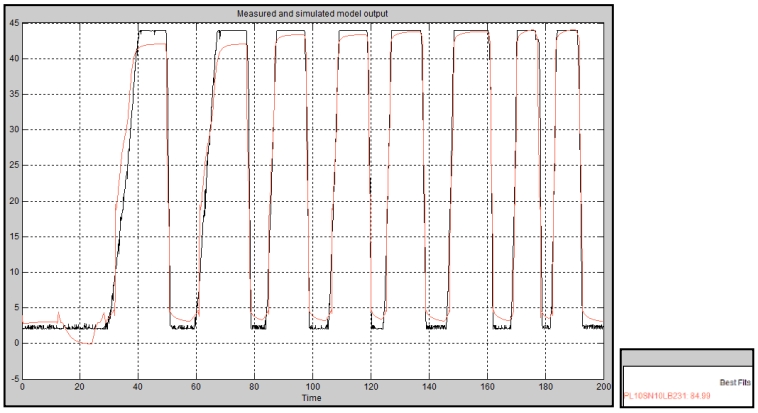


Figura 237 Resultado de modelo no lineal

Siguiendo con el método, se realiza el filtrado y pulido de la señal, así como la comparación de los resultados obtenidos.

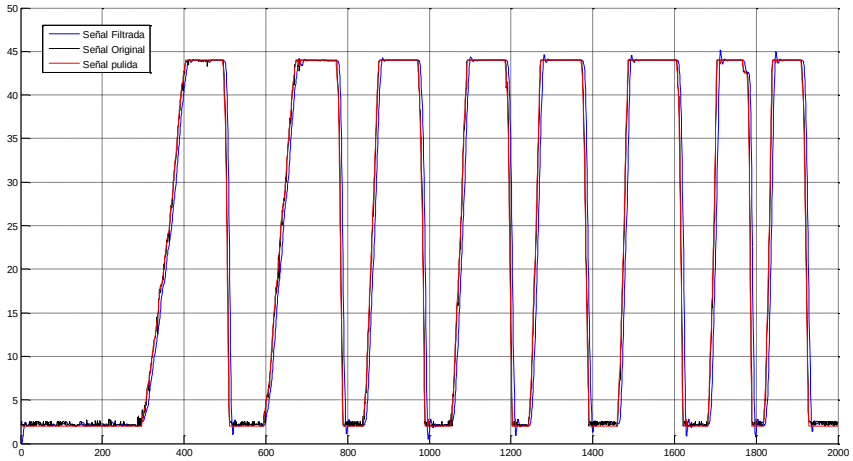


Figura 238 Análisis de proceso de mejora de la señal

Finalmente se vuelve a intentar obtener un modelo no lineal utilizando la nueva señal que se obtuvo y se compara los resultados.

Nota: En el PID sobre "Mejora de Conjunto de Muestras" se aborda de manera más profunda el proceso de filtrado y mejoramiento del conjunto de muestras.

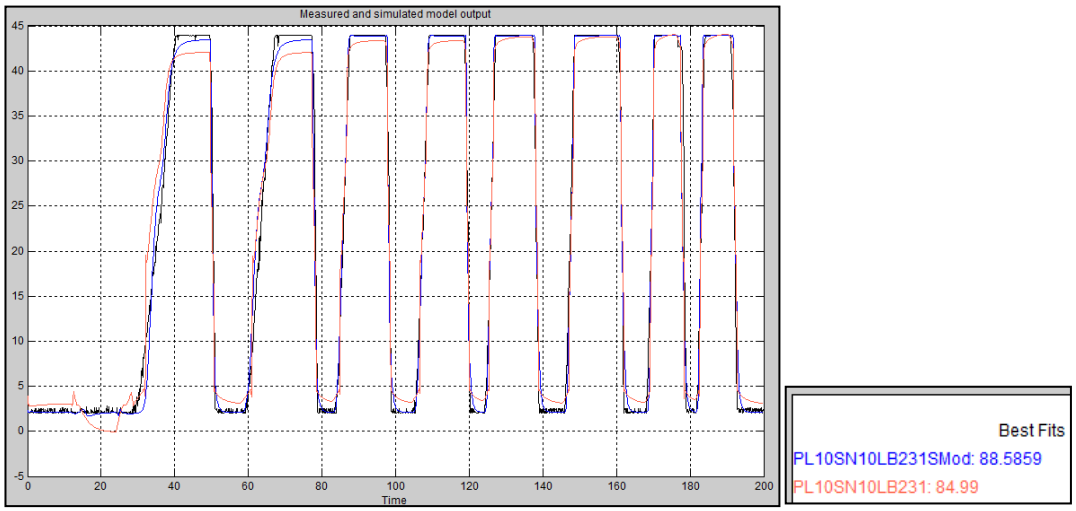


Figura 239 Resultados de modelos no lineales

Con el nuevo conjunto de muestras se hace también un acercamiento lineal a la planta y se obtiene un modelo lineal desde ident utilizando solo una sección de la muestra, este se lo compara con uno obtenido utilizando el primer conjunto de muestras, de igual manera se compara los resultados del modelo linealizado en Matlab.

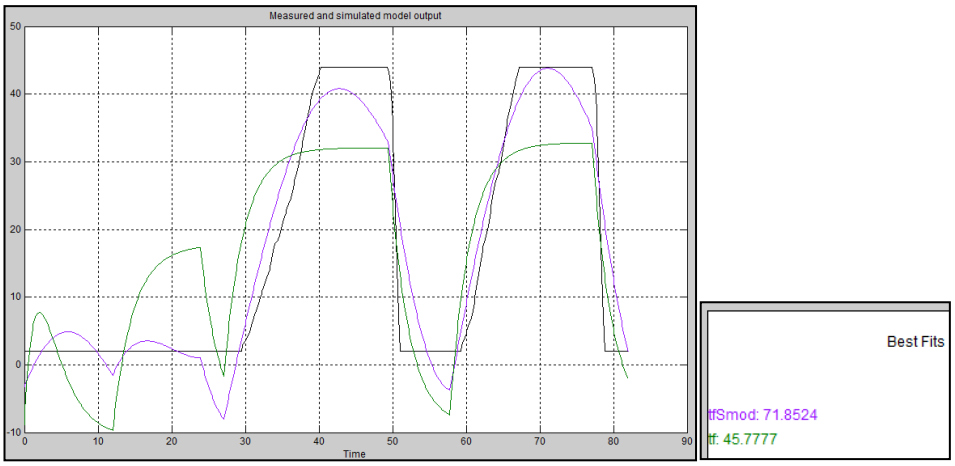


Figura 240 Resultados de funciones transferencias

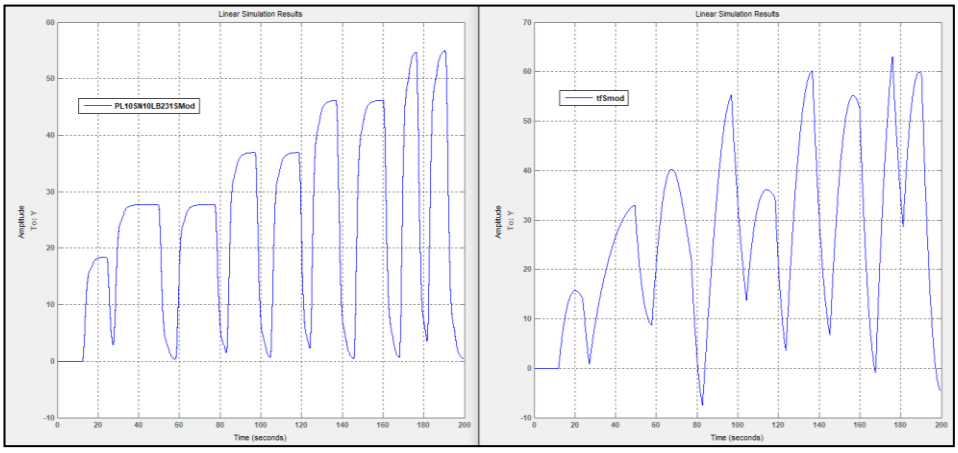


Figura 241 Comparación de simulaciones lineales

Finalmente, en Matlab utilizando "sisotool" se procede a realizar un controlador PID, para implementar en la planta real, para ello se utilizó el modelo linealizado "PL10SN10LB231SMod", con el cual se obtuvieron los siguientes resultados.

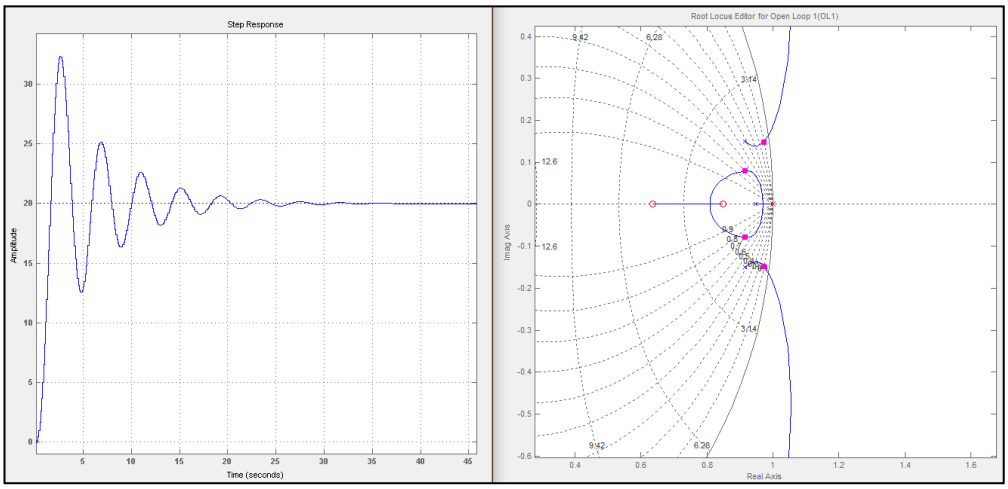


Figura 242 Simulacion en Sisotool, con la respuesta de PI

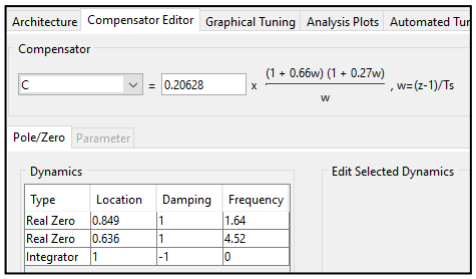


Figura 243 PID utilizado

Utilizando el script de nombre Valores PID se obtuvo de manera rápida los valores del control que se diseñó en sisotool.

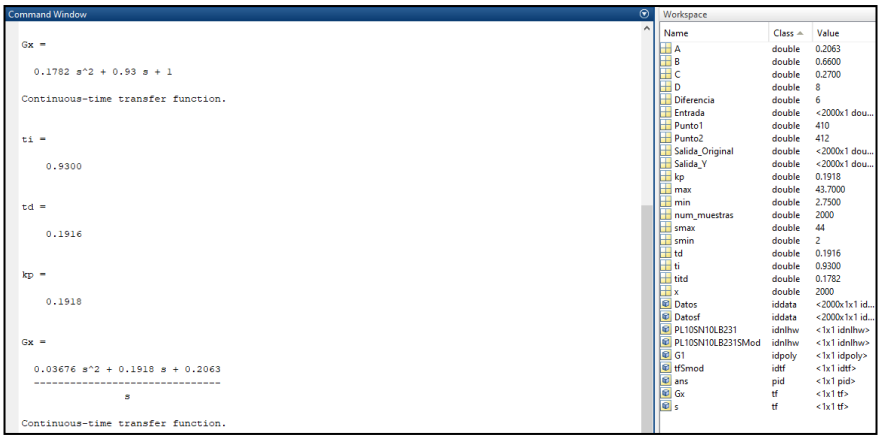


Figura 244 Valores de las ganancias del PID

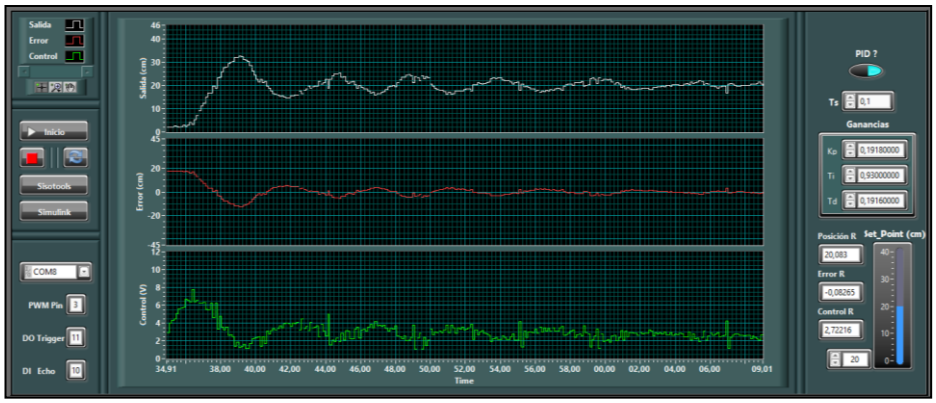


Figura 245 Planta con el controlador PID

Como puede apreciarse, el PID implementado tiene un comportamiento similar al simulado utilizando el modelo linealizado del modelo no lineal obtenido en ident a partir del conjunto de muestras pulido, cuyo origen en el VI de "Practica_Modelado LE" permitió realizar un controlador PID eficiente sin necesidad de estar realizando ajustes en la planta, debido al alto porcentaje de aproximación que se obtuvo en el modelo no lineal, y al buen resultado obtenido en la linealización del mismo.

5. Resultados

Cada tipo de planta posee características únicas asociadas a su comportamiento y elemento físico a controlar, así como los valores medidos con los cuales se analiza el estado de la misma, dotando así a estas de diferentes aptitudes para el proceso de aprendizaje de los controles PID.

5.1 Motor de paso

La planta cuyo proceso se centra en el uso de un motor de paso como ente a controlar, presenta por un lado un sistema muy estable, capaz de llegar a un error en estado estacionario casi nulo con solo un control proporcional. No obstante, se tiene una limitante en el ente físico usado para su control, el cual es el tiempo de espera entre paso y paso.

Este tiempo de espera afecta de manera directa a la velocidad del desplazamiento de la aguja, dicho tiempo no puede ser menor a un valor mínimo especificado por el diseñador lo cual limita la respuesta máxima en el tiempo que puede tomarle al motor en llevar al sistema del punto A al punto B en determinado momento.
Para ejemplificar este comportamiento se diseñó un PID que tuviese un sobrepico con la finalidad de que la velocidad de inicio en el sistema fuese la máxima y se mantuviese de esa manera el mayor tiempo posible.

Kp	14,3980
Ti	10,6000
Td	2,4085

Tabla 1 PID Implementado Motor de Paso

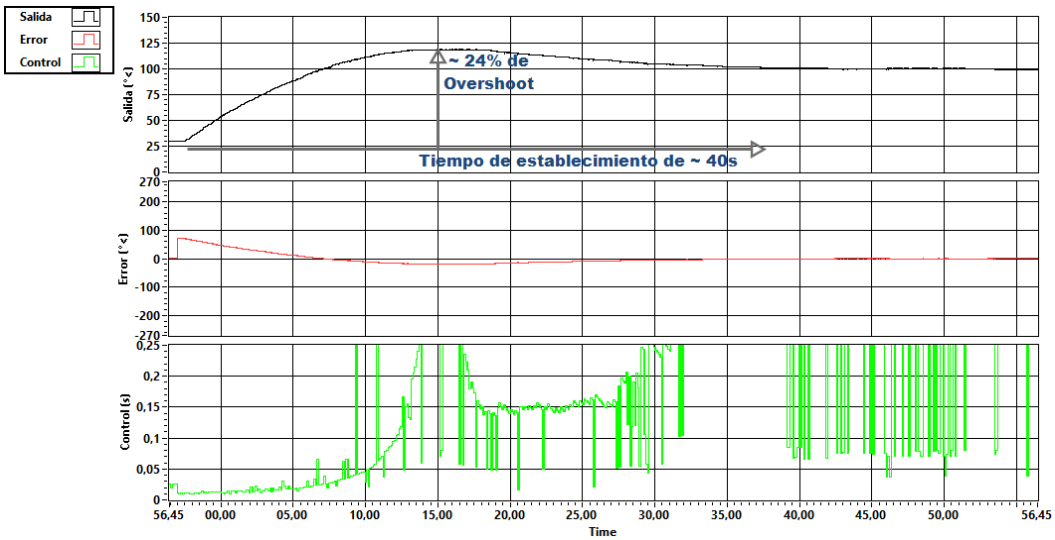


Figura 246 Resultado Planta Motor de Paso

5.2 Plantas térmicas 1 y 2

De característica muy lenta, las plantas térmicas 1 y 2 presentan un ejemplo de lo que es un sistema de respuesta lenta que requiere en consecuencia un control que tome en cuenta dicho comportamiento, y atención por parte del estudiante al diseñar el mismo.

Debido a su lenta respuesta, al simular su comportamiento se suele comerte el error de intentar el uso de controles cuyo valor de control es muy alto, esto debido ya sea a la ganancia o al valor proporcional utilizado, por ello es importante el diseñar un control cuyos valores de control se mantengan dentro del rango de diseño.

A consecuencia de esto se realizó un control con ganancia baja cuya respuesta se encuentra dentro de los rangos de voltaje que se dispone y con una respuesta lenta, característica intrínseca de este sistema.

Kp	0,4700
Ti	25,7000
Td	5,3930

Tabla 2 PID Implementado Planta Térmica

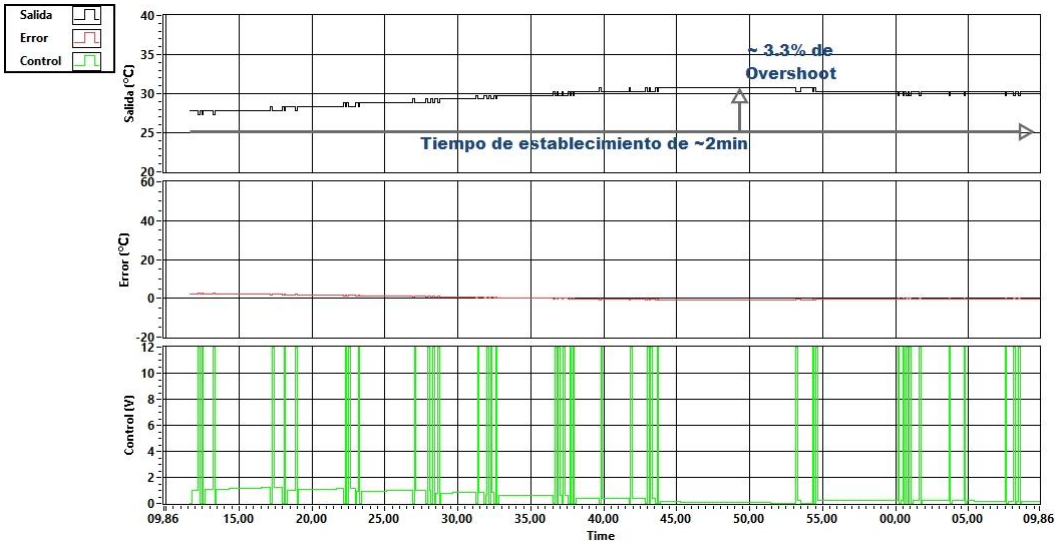


Figura 247 Resultado Planta Motor Térmica 1 y 2

5.3 Plantas Levitadoras 1 y 2

Sistemas no lineales de respuesta rápida, que requieren según el objeto de medición de diferentes valores base para elevarse o mantenerse levitando, debajo de estos cae por efecto de gravedad.

Para su control se buscó utilizar un PID más equilibrado en donde por un lado se asegure valores mínimos de control que se encuentren por arriba de los valores necesarios para que el objeto levite, dejando a su vez espacio para que se llegue a aquellos valores que hacen que este caiga.

De manera adicional también se hace uso de un valor proporcional no muy alto que permita eliminar el error en estado estacionario y un valor derivativo adecuado para la velocidad con la que se desplaza el objeto, así como para compensar por pequeños ruidos provocados por la oscilación del objeto medido.

Kp	0,1918
Ti	1,1000
Td	0,2916

Tabla 3 PID Implementado Planta Levitadoras

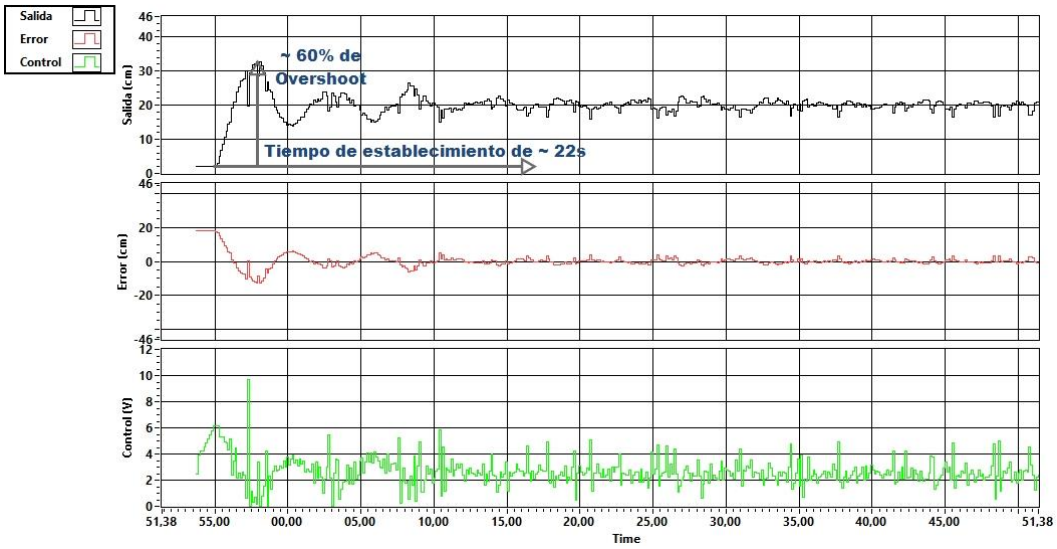


Figura 248 Resultado de la Planta Levitadora 1 y 2

6. Conclusiones y Recomendaciones

6.1 Conclusiones

- En el transcurso del proyecto de grado se culminó con éxito las implementaciones 3 tipos de módulos distintos, cada uno con características útiles para distintos aspectos presentes en los procesos de control PID

- Basándonos en la experiencia universitaria, se optó con por el uso de dispositivos didácticos y de fácil acceso como son: Arduino, sensor ultrasónico, sensor de temperatura LM35, motor de paso, potenciómetros, ventiladores dc y luminarias dc.

- Cada una de las prácticas desarrolladas tiene un tipo de entorno virtual distinto todas estas vistos en el capítulo 4. Se logró implementar con éxito varios VI's a usar en diferentes módulos

- Cada una de la práctica cuenta con una descripción integral sobre el funcionamiento de los Vi's asociados a la misma, así como una breve información teórica sobre los temas tratados a fin de volverlas autodidactas, ya que son un complemento a las clases dadas.

6.2 Recomendaciones

Es importante tener en cuenta en qué tipo de materiales se van a trabajar los diseños, y que tipo de tecnologías serán necesarias para estos, en el caso de este proyecto, se dio uso de impresiones 3D, así como talleres especializados en acrílicos y aluminio.

Llevar un orden por carpetas y subcarpetas, así como organizar los archivos con nombres y fechas representativos, separar las imágenes de los archivos de investigación entre otras alternativas, resultan de vital importancia, porque lo que empieza como un archivo de texto, termina convirtiéndose en decenas de versiones una mejor que la anterior y contar con ese historial de avance es primordial.

El proyecto en su estado actual admite la adición de nuevas plantas, y la inclusión de nuevas librerías para el control de estas, así como el uso de otros tipos de controladores, esto debido a que lo implementado está lejos de ser definitivo, y en tecnología siempre se están dando avances de los cuales futuros ingenieros pueden dar uso para mejorar lo que se ha logrado con este proyecto.

7. Bibliografía

[1] K. J. Astrom, Diseños de Sistema de Control, 2002.
[2] V. Mazzone, Control Automaticos 1, Quilmes, 2002.
[3] M. A. Moreno, Apuntes de Control PID, La Paz, 2001.
[4] J. F. DULHOSTE, Control de Procesos para Ingenieros, Merida, 2010.
[5] L. Pedrini, «Sistemas de Control,» Ciudad de Mexico, 2011.
[6] W. Bolton, Ingenieria de Control, Alfaomega, 2009.
[7] V. G. Garay, Acciones Basicas de Control, Bilbao.
[8] A. M. Bueno, Sistema de control Automatico, Alicante, 2011.
[9] Best Seller, Metodologia de la Investigacion, Mexico, 2006.

8. Anexos

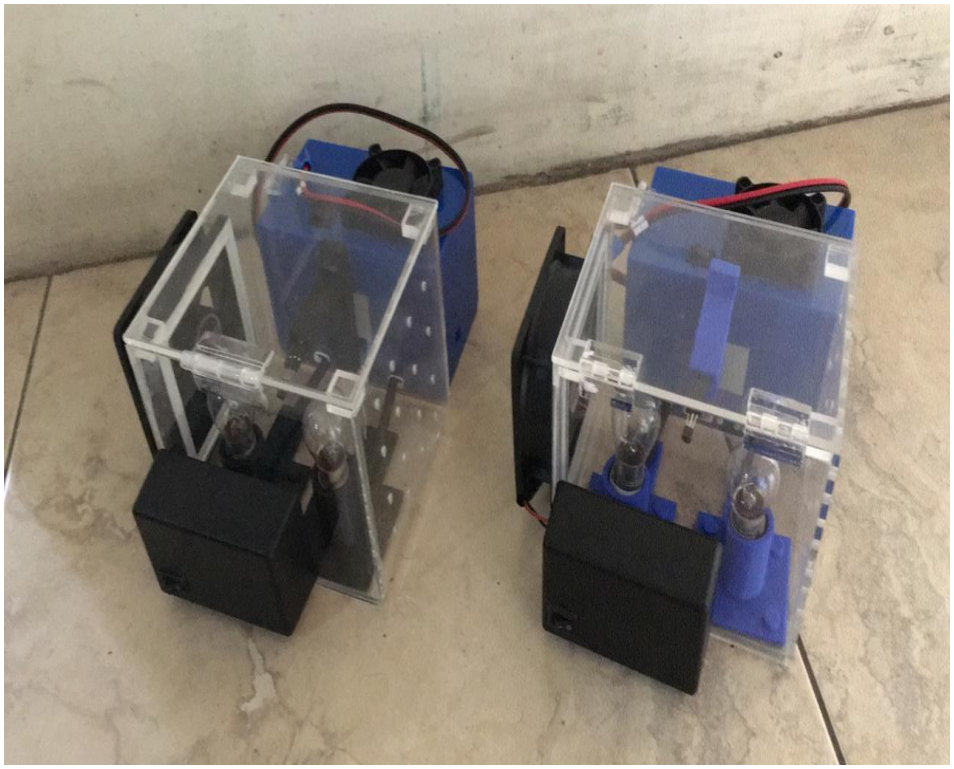


Figura 249 Módulos de las Plantas Térmicas

Figura 250 Módulos de las Plantas Levitadoras

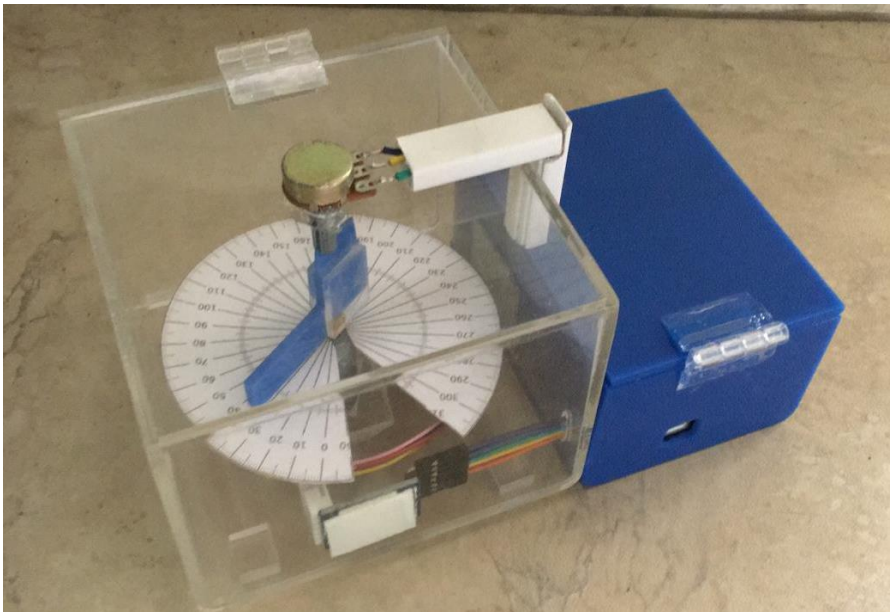


Figura 251 Modulo de la Planta Motor de Paso