



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA LA IDENTIFICACIÓN Y CONTROL DE UNA PLATAFORMA EN SUSPENSIÓN MEDIANTE LABVIEW

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

ÍNDICE GENERAL

DOCUMENTO Nº 1 : MEMORIA

DOCUMENTO Nº 2 : ANEXO DE PROGRAMACIÓN

DOCUMENTO Nº 3 : ANEXO DE DISEÑO

DOCUMENTO Nº 4 : MANUAL DE USUARIO

DOCUMENTO Nº 5 : PLIEGO DE CONDICIONES

DOCUMENTO Nº 6 : PRESUPUESTO

RESUMEN

El presente proyecto trata del desarrollo de una aplicación, a través de la plataforma de diseño LabVIEW, para controlar el proceso de una plataforma en suspensión. En las páginas siguientes se expondrán los pasos seguidos para la modelización del proceso y el posterior diseño del controlador de dicho modelo.

Gracias al programa MatLab se consigue desarrollar un modelo a partir de los datos obtenidos por un ensayo. Con este modelo se obtendrán los parámetros del controlador correspondiente que serán introducidos en la aplicación creada en LabVIEW que nos facilitará el control del proceso.

La aplicación contará con distintos tipos de funcionamiento entre los cuales se encuentran: el control manual, control por PID, control por Red...que complementarán los distintos usos de la aplicación.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°1 - MEMORIA**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

MEMORIA

1.	OBJETO DEL PROYECTO	1
2.	ANTECEDENTES DEL PROYECTO.....	1
2.1	Descripción General del Sistema.....	1
2.1.1	Descripción del Prototipo.....	2
2.2	LabVIEW.....	12
2.2.1	Introducción.....	12
2.2.2	Ventajas de Utilizar LabVIEW.....	12
2.3	Control por Computador.....	13
2.3.1	Introducción.....	13
2.3.2	Componentes del Control por Computador.....	13
2.3.3	Actividades y Ventajas Realizadas por el Control del Computador:	15
2.4	Introducción a los Controladores.....	16
2.4.1	Entrada.....	17
2.4.2	Controlador.....	17
2.4.3	Tipos de Controlador.....	17
2.4.4	Planta.....	19
3.	Motivación y Justificación.....	19
3.1	Motivación.....	19
3.2	Justificación.....	20
4.	DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA.....	20
4.1	Implantación del Control Manual mediante LabVIEW.....	21
4.1.1	Función de Parado.....	22
4.2	Implantación del Control Mediante PID a través de LabVIEW.....	22
4.3	Obtención del Modelo y Diseño del Controlador PID.....	23
4.3.1	Obtención del Modelo.....	23
4.3.2	Diseño del Controlador PID.....	24
4.4	Implementación de diversas Herramientas.....	24
4.4.1	Perfil.....	24
4.4.2	Autotuning.....	25
5.	BIBLIOGRAFÍA.....	26

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

1. OBJETO DEL PROYECTO.

El proyecto está destinado a diseñar una aplicación, desarrollada en LabVIEW, que controle y regule la altura a la que se encuentra una plataforma en suspensión mediante un ventilador controlado vía PC.

2. ANTECEDENTES DEL PROYECTO.

El prototipo utilizado para la realización del proyecto pertenece al Departamento de Ingeniería de Sistemas y Automática (DISA) de la Universidad Politécnica de Valencia. Está compuesto por un tubo de plástico que dispone de una plataforma en su interior que será elevada mediante el aire que genera un ventilador situado en la base del tubo.

Para poder controlar el ventilador es necesario un computador así como una tarjeta de adquisición para la comunicación entre el PC y el proceso. Se pretende conseguir un sistema que gestione y controle el proceso de manera que pueda ser manipulado por el usuario. A continuación se procede a analizar más en detalle cada parte del proyecto.

2.1 Descripción General del Sistema.

Como podemos observar en el esquema siguiente (Ilustración 1 - Esquema del sistema) nuestro sistema no solo está compuesto del PC y del tubo, sino que además dispone de etapas intermedias entre ambos. Esto es debido a que el ordenador no puede procesar los datos que emite el detector y éstos han de ser procesados previamente para la adecuada lectura del PC. Para ello se empleará una tarjeta de adquisición de datos con sus respectivos conversores de A/D y D/A.

El funcionamiento del sistema empieza con la lectura de datos realizada por el sensor, este devuelve una señal analógica que ha de ser traducida a señal digital para que la TAD pueda leer los datos. Una vez leídos estos son enviados al ordenador, el cual procesará la señal y devolverá otra señal en función de la señal recibida. Esta nueva señal llegará al ventilador a través de la TAD y tras pasar por un conversor que traducirá la señal de digital a analógica.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

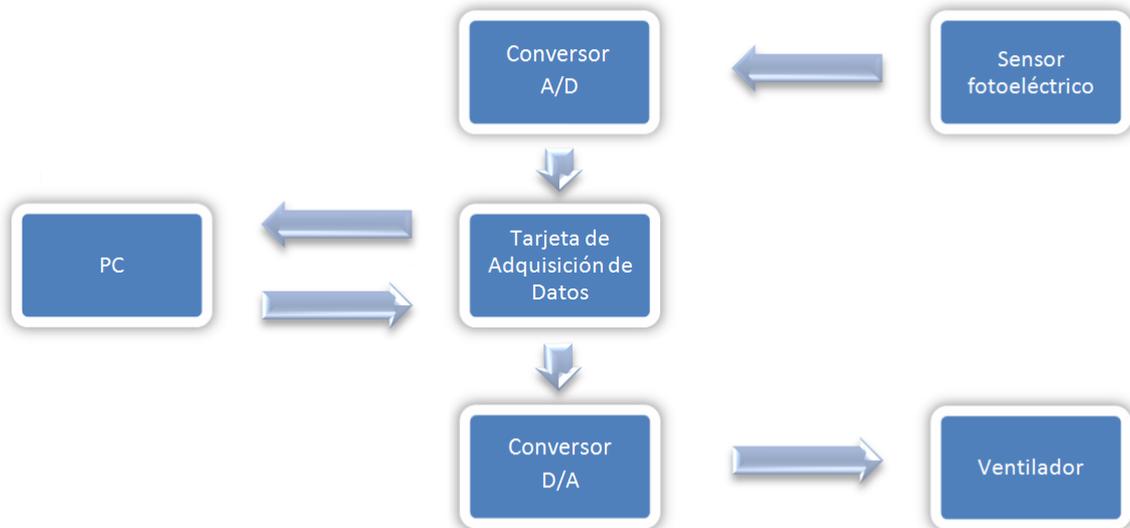


Ilustración 1 - Esquema del sistema

2.1.1 Descripción del Prototipo.

Físicamente, se pueden distinguir de un simple vistazo cada uno de sus componentes. A continuación se presenta una imagen (Ilustración 2 - Imagen del prototipo) en la cual se puede apreciar claramente los distintos elementos mencionados previamente, y alguno más:

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

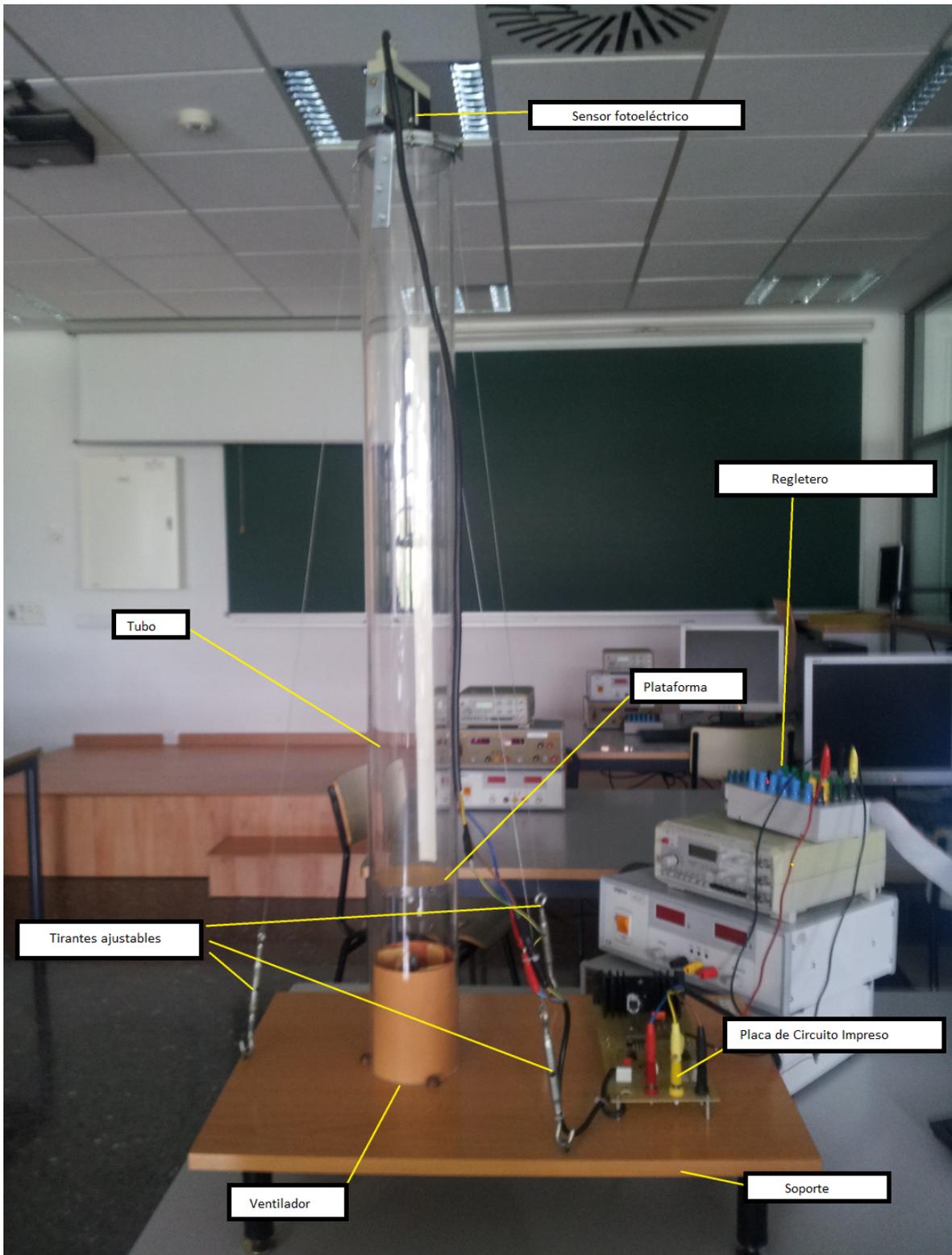


Ilustración 2 - Imagen del prototipo

- **Detector fotoeléctrico:** Sensor situado en la parte superior del tubo que proporciona un valor dependiendo de la distancia a la que se encuentre respecto a un objeto situado en la trayectoria de su haz.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

- **Zona muerta de detección:** esta es la zona que dista del sensor menos del mínimo que detecta. En esta zona el sensor devolverá el mismo valor para todas las lecturas, es aconsejable no tomar medidas en esta zona durante el ensayo.
- **Tirantes ajustables:** tensores para fijar, estabilizar y equilibrar el tubo. Se necesita un grado de perpendicularidad con el suelo para evitar roce y desequilibrios de la plataforma con las paredes internas del tubo.
- **Cilindro deslizante:** plataforma que propulsaremos en nuestro experimento. Provista de un peso en la parte inferior debido a la ligereza del material con el que está hecho.
- **Tubo de PVC transparente:** contiene la plataforma que será elevada. Delimita el espacio en el que transcurre el experimento.
- **Zona de comportamiento acusadamente no lineal:** el estar demasiado cerca del ventilador y tan lejos del sensor provoca errores en la medida provocando una no linealidad que explicaremos más adelante.
- **Placas de circuito impreso:** Placa donde se encuentran las conexiones con la fuente de alimentación mediante regletero, la salida de datos del sensor fotoeléctrico y la entrada del ventilador.
- **Placa soporte:** tabla de madera con cuatro patas regulables, ajustables para nivelar la superficie de nuestro prototipo.
- **Motor/ventilador:** aparato eléctrico que se encargará de propulsar la plataforma a lo largo del tubo.
- **Fuente de alimentación:** Pese a no salir en la imagen, debajo de la torre de aparatos eléctricos entre los que se encuentra la tarjeta de adquisición, se encuentra la fuente de alimentación. Por ello se decide ajuntar una imagen como información adicional :

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)



Ilustración 3 - Fuente de alimentación

2.1.1.1 El Detector fotoeléctrico.

Para obtener la altura a la que se encuentra la plataforma se utiliza un sensor fotoeléctrico con salida analógica, concretamente el modelo XUJ-K803538 de la firma TELEMECANIQUE. Para la configuración del sensor, se indicarán los pasos posteriormente.

Este sensor funciona mediante la técnica de la triangularización lo cual permite la identificación de la altura de la plataforma deslizando sin importar el color de ésta.



Ilustración 4 - Sensor fotoeléctrico

Al ser un sensor óptico, el ruido generado por perturbaciones ajenas a nuestro sistema se ha de tener en cuenta. El polvo, el humo, la suciedad son perturbaciones que perjudican a la sensibilidad de nuestro sistema, por eso es importante tener limpio el sistema y elegir correctamente el ambiente en el que se desarrollará el ensayo, también será necesario aplicar un filtro para el ruido de 50Hz creado por la luz de los tubos fluorescentes del laboratorio.

El sensor devuelve una señal analógica que puede variar de los 0-10V o bien de 4-20mA.

A continuación, en la (Ilustración 5 - Gráficas del sensor) (proporcionadas por los fabricantes) se muestra el comportamiento que sigue el sensor, teóricamente lineal entre 20-80cm pero durante el ensayo nos daremos cuenta de que no es lineal en absoluto. Se puede configurar de ambas formas.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

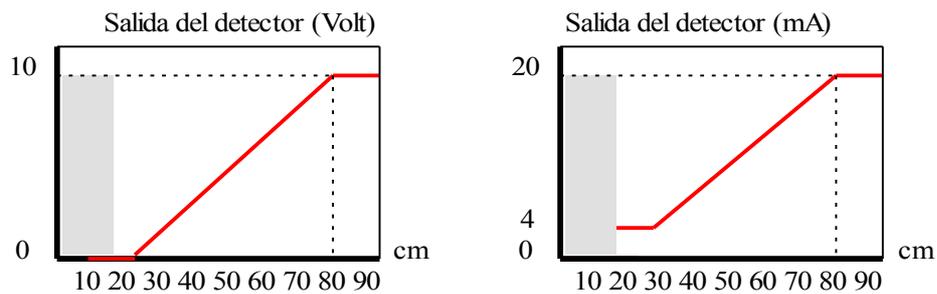


Ilustración 5 - Gráficas del sensor

2.1.1.2 El Ventilador.

La propulsión de la plataforma se logra mediante un ventilador situado en la parte inferior del tubo. Su rango de actuación va desde los 0V a los 12V. Dado que la tarjeta de adquisición trabaja en el rango de 0-10V será necesario utilizar un amplificador no inversor de ganancia 1,2.

Se descubre tras realizar numerosos experimentos que la entrada para que el sistema se encuentre en reposo, ni gane ni pierda altura (o para que no oscile tanto en altura) es relativamente variada al empezar el proyecto esta rondaba los 6.875 V, mientras que al finalizar el proyecto alcanzaba los 7.07 V, esto puede ser debido a los cambios climáticos sufridos, una variación en la temperatura ambiente de unos casi 10 grados y probablemente una variación en la presión además del uso prolongado del ventilador. El valor de la tensión para que la plataforma esté en reposo será de utilidad a lo largo del proyecto.

Para evitar el sobrecalentamiento del ventilador y los posibles daños que se puedan ocasionar, se ha de limitar la tensión a la que se le somete. También hay que comprobar que no hay ningún objeto debajo del soporte o del ventilador, ya que este podría dificultar el funcionamiento de este.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.1.1.3 Placa de circuito impreso.

Disponemos de una placa que cuenta con entradas de tensión a 15V mediante un regletero que ira conectado a la fuente de alimentación y conexiones con la TAD que serán: la conexión al detector fotoeléctrico y la conexión al ventilador. Como se puede ver en la imagen (Ilustración 6 - Placa de circuito impreso) la entrada de tensión de 24V se ha eliminado debido al poco uso de ésta. Será el encargado de enviar la tensión al sensor fotoeléctrico y al ventilador. El interruptor que se encuentra en la parte central con las opciones PC y PID, tendrá que permanecer en la opción PC a lo largo de todo el proyecto. Como se puede observar a pesar de tener 4 conexiones, solo se utilizarán las dos de la izquierda que serán las encargadas de transmitir los datos del sensor fotoeléctrico a la TAD y de la TAD al ventilador.

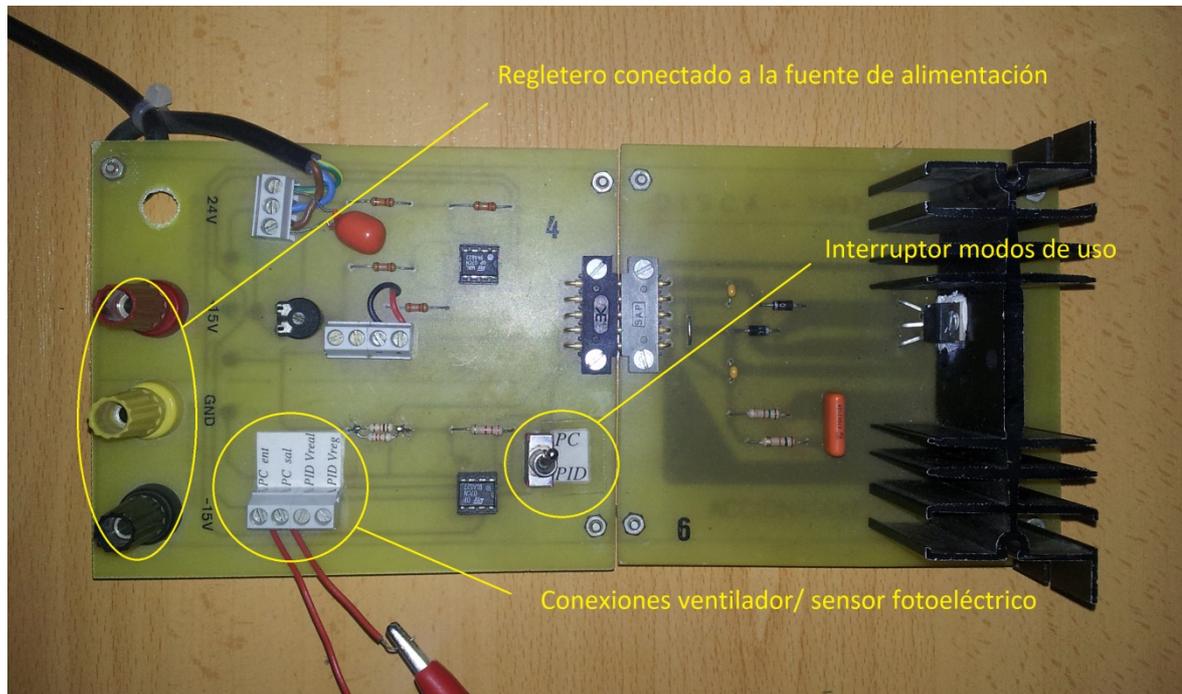


Ilustración 6 - Placa de circuito impreso

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.1.1.4 Tarjeta de Adquisición de Datos.

La tarjeta de adquisición a utilizar será la del modelo PCI-9112 de AD-LINK que podemos encontrar instalada en los ordenadores de los laboratorios del DISA. Es necesaria además la conexión de un regletero para la realización de los ensayos. En la imagen siguiente se muestra dicha Tarjeta de Adquisición (Ilustración 7 - Tarjeta de Adquisición de Datos):



Ilustración 7 - Tarjeta de Adquisición de Datos

Reunir conversores A/D y D/A, entradas y salidas digitales, contador y un temporizador, en una sola tarjeta de adquisición, junto a su económico precio, hacen que éste modelo sea idóneo tanto para experimentos en laboratorio como para procesos industriales.

A continuación se muestran las tablas con las características de la tarjeta de adquisición de datos PCI-9112 de AD-LINK

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

- Entradas analógicas (Convertidor Analógico /Digital)

Canales:	16 canales con masa común.
Resolución:	12 bits
Rangos de Entrada:	Bipolar: (-10v a 10v), (-5v a 5v). Configurables mediante <i>jumpers</i> .
Tensión Límite:	+30v y -30v, máximas continuas.
Tipo de Conversión:	Aproximaciones sucesivas
Convertidor:	HADC574z
Velocidad de conversión:	100 KHz (máx.)
Precisión:	0.015% de la lectura ± 1 bit
Linealidad:	± 1 bit
Modo de disparo:	Programa, temporizador y externo
Transferencia de datos:	Programa (<i>polling</i>), interrupciones o DMA

- Salida analógica (Convertidor Digital/Analógico)

Canales:	2 canales.
Resolución:	12 bits.
Rango de salida:	0-5v / 0-10v (con referencia interna) ± 10 v con referencia externa.
Tensión de Referencia:	Interna: -5v / -10 v. (± 0.05 v) Externa: DC o AC, ± 10 v máxima.
Tipo de conversión:	Multiplificador monolítico de 12 bits.
Dispositivos analógicos:	AD7541AKN o equivalentes.
Linealidad:	$\pm 1/2$ bit.
Carga de salida:	5 mA máx
Tiempo de establecimiento:	30 microsegundos

- Entrada digital

Canal:	16 bits.
Nivel:	Compatible TTL.
Tensión de entrada:	Nivel bajo: 0.8 v máximo. Nivel alto: 2.0 v mínimo.
Carga de entrada:	Nivel bajo: 0.4 mA máximo a 0.5 v. Nivel alto: 2.0 mA mínimo a 2.7 v.

Ilustración 8 - Especificaciones técnicas de la TAD (1)

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

- **Salida Digital**

Canal:	16 bits.
Nivel:	Compatibles TTL.
Tensión de salida:	Nivel bajo: admite hasta 8 mA a 0.5 v máx. Nivel alto: proporciona hasta 0.4 mA a 2.4 v mín.

- **Temporizador/Contador programable**

Dispositivo:	INTEL 8254
Contadores:	3 canales, 16 bits (2 para generar disparos, 1 disponible para el usuario).
Entrada (Gate):	Compatible TTL/DTL/CMOS.
Base de tiempos:	2 MHz (cristal de cuarzo del chip).
Salida/Disparo:	Según se programen los contadores en función de la frecuencia del cristal. (0.00046 Hz – 0.5 MHz.)

- **Interrupciones**

Nivel:	IRQ3 a IRQ15 seleccionable por <i>jumper</i> (JP5y JP6).
Habilitación:	Vía S0, S1 y S2 del registro de control.

- **Canal de DMA**

Nivel:	1 o 3 seleccionable por <i>jumper</i> . (JP8y JP9).
Habilitación:	Vía S0, S1 y S2 del registro de control.

- **Especificaciones generales**

Consumo:	+5v: típico 500 mA, máximo 1A +12v: típico 50 mA, máximo 100mA -12v: típico 14 mA, máximo 20mA
Conector de Entrada/Salida:	Conector 20 pines. Adaptador disponible para convertirlo en un conector de 37 pines de tipo D.
Dirección base de Entrada/Salida:	Se requieren 16 direcciones de E/S. La dirección base es configurable a través de conmutadores.
Temperatura de funcionamiento:	Desde 0 °C hasta 50 °C
Temperatura en embalaje:	Desde -20 °C hasta 65 °C
Peso:	243 gramos

Ilustración 9 - Especificaciones técnicas de la TAD (2)

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.1.1.5 Regletero

Se dispone de un Regletero que cuenta con 2 modos de funcionamiento; el ACL 8112 con: 6 entradas de +/- 10V, 2 salidas de 0-10V, otras 2 salidas de +/- 10V y 2 masas; y el ACL 6126 de: 1 masa, 3 salidas a +/- 5V y 3 más a +/- 10V.



Ilustración 10 - Regletero

2.1.1.6 Alimentación y Cableado.

Las conexiones entre el detector fotoeléctrico, la placa con circuito integrado, la fuente de alimentación y la tarjeta de adquisición se realizan mediante cables con bananas o pinzas en los extremos. Un cable banana-banana triple entre la fuente de alimentación y la placa de circuito impreso, un cable banana-banana entre la masa de la fuente de alimentación y la masa de la TAD y cables pinza-banana para las conexiones entre la TAD y Placa de Circuito Integrado.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.2 LabVIEW

2.2.1 Introducción

Una de las exigencias del proyecto era que la aplicación fuese diseñada mediante LabVIEW. Por tanto, antes de empezar el proyecto, se realizará una explicación sobre LabVIEW, ventajas e inconvenientes que esta plataforma plantea y el amplio abanico de posibilidades que nos brinda.

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) es una plataforma de diseño virtual que salió al mercado por primera vez en 1986 y que a pesar de estar destinado en principio para máquinas MAC, fue adaptada para Windows en el año 1992.

2.2.2 Ventajas de Utilizar LabVIEW

La principal característica de la plataforma de diseño de LabVIEW es que utiliza un lenguaje de programación gráfico o lenguaje G (de gráfico). Este lenguaje permite al ingeniero que lo utiliza, programar de forma rápida e intuitiva. Gracias a este lenguaje el código y las conexiones entre las herramientas se pueden apreciar cómodamente, pudiendo reordenar las herramientas y agruparlas al antojo del programador. Queda a disposición del usuario crear un código ordenado y fácil de leer.

Al no necesitar de amplios conocimientos de programación, es un software orientado para todos los públicos y de un aprendizaje rápido. Dispone de una biblioteca extensa repleta de numerosas VIs (Virtual Instrument) cada una con una función distinta. Estas herramientas virtuales son capaces de complementarse entre ellas y lograr cumplir las necesidades del programador. Ordenadas por funcionamiento dentro del menú de la biblioteca, también dispone de un buscador para una búsqueda más cómoda aún. Además dispone de tutoriales en los que se puede apreciar el funcionamiento de cada una de las herramientas disponibles, lo que ofrece una capacidad autodidacta de esta plataforma.

Junto a todas estas numerosas herramientas se pueden encontrar también herramientas de Entrada/Salida lo que facilita la comunicación con periféricos sin necesidad de descargar bibliotecas adicionales para realizar la conexión.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

Otra ventaja añadida a LabVIEW es que además de disponer de un código ordenado y claro, es que la interfaz también dispone de estas características. Totalmente personalizable y con numerosas posibilidades, esta interfaz permite desde ordenar las herramientas por grupos, hasta definir los colores de los elementos de la interfaz, permitiendo crear un entorno visual amigable e intuitivo para el futuro usuario del programa. (Para ver un ejemplo de la interfaz acudir al Manual del usuario).

Además permite la reutilización de códigos de otras aplicaciones de LabVIEW, permitiendo así el poder utilizar aplicaciones realizadas previamente para realizar una nueva aplicación. Para acabar, mencionar que debido a la popularidad de su lenguaje se han ido añadiendo múltiples librerías de aplicaciones creadas por los usuarios de LabVIEW con más VIs aún que pueden ser descargadas gratis o pagando.

2.3 Control por Computador

2.3.1 Introducción

El primer concepto de control por computador operando en tiempo real se remonta al año 1950, Brown y Campbell. Aparece un computador en un control realimentado en una gráfica de un documento descrito por ellos. No obstante pasaron 8 años hasta que apareció el primer sistema computarizado llevado a la industria por Louisiana Power and Light Company en USA, no obstante no era un sistema de control. El primer sistema de control de lazo cerrado por computador fue utilizado por Texaco Company en Texas.

2.3.2 Componentes del Control por Computador

En la imagen siguiente se muestra un diagrama de un sistema de control por computador con sus respectivas componentes:

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

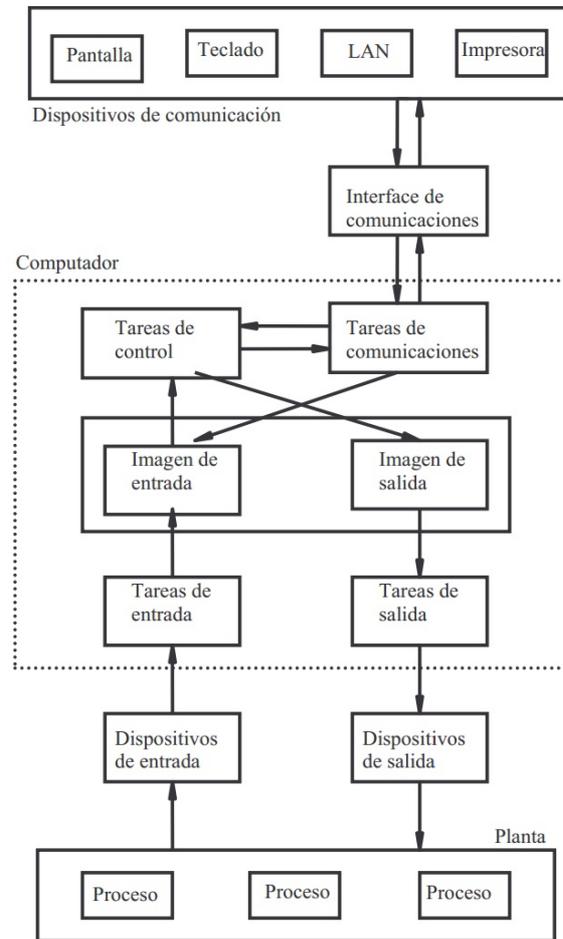


Ilustración 11 - Diagrama de Control por Computador

- **Planta:** corresponde al proceso o conjunto de procesos que se llevan a cabo.
- **Dispositivos de entrada:** son los encargados de transmitir la información de lo ocurrido en el proceso al computador, estos datos corresponden a los valores de las variables controladas bien sean señales digitales o analógicas.
- **Dispositivos de salida:** son los encargados de transmitir información del computador para que el sistema de control actúe sobre el proceso, al igual que los dispositivos de entrada, también pueden ser señales digitales o analógicas.
- **Tareas de entrada:** tratan las señales obtenidas del proceso, por ejemplo linealización, cambio de escala, filtraje, etc...

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

- **Tareas de salida:** al igual que las tareas de entrada, estas se encargaran de traducir los datos para los dispositivos de salida, tales como: generación de alarmas, cambio de escala, etc.
- **Tareas de control:** algoritmos de control (tales como PID, ON/OFF...) utilizados por el sistema sobre los lazos de control para los que se le haya programado.
- **Tareas de comunicación:** son las empleadas para adecuar las condiciones del sistema de cara a la interpretación del usuario. También son necesarias para realizar modificaciones sobre las condiciones del trabajo en planta.
- **Interfaz de comunicaciones:** adapta las señales generadas por el computador a los distintos dispositivos de comunicación para que estos puedan transmitir la información obtenida del proceso a los usuarios.

2.3.3 Actividades y Ventajas Realizadas por el Control del Computador:

2.3.3.1 *Actividades:*

Las tareas realizadas gracias al control por computador son las siguientes:

- Adquisición de datos
- Control secuencial
- Lazos de Control Digital Directo
- Control supervisor
- Análisis de datos
- Almacenamiento de datos.
- Interfaz hombre-máquina

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.3.3.2 Ventajas:

El control por computador está destinado para cumplir con:

- Eficiencia de la operación
- Facilidad de la operación
- Seguridad
- Producto de mayor calidad
- Reducción del desperdicio
- Reducción del impacto ambiental debido a la producción
- Reducción del tiempo de producción
- Control secuencial

2.4 Introducción a los Controladores.

Para entender mejor el trabajo realizado en este proyecto se ha decidido adjuntar una explicación breve sobre lo que son los controladores y que funciones tienen en el sistema. En la siguiente imagen (Ilustración 12 - -Diagrama de un Sistema) se muestra un diagrama sobre un sistema de lazo cerrado, en él se detallan las distintas partes que tiene un sistema y los cambios que experimenta a lo largo de su recorrido.



Ilustración 12 - -Diagrama de un Sistema

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.4.1 Entrada.

A nuestro sistema se le introduce una señal denominada Referencia, será el valor que deseamos que alcance nuestra variable de salida. Esta señal entrara en un comparador que efectuará la diferencia entre la salida deseada y la salida medida a este resultado se le llamará señal de error y será lo que entre en nuestro controlador o regulador.

2.4.2 Controlador.

La señal de error entrará en el controlador para que a partir del valor del error, el controlador devuelva una variable de salida regulada $u(t)$ correspondiente al valor del error entrante.

2.4.3 Tipos de Controlador.

Dependiendo de cómo funcione un controlador, se podrá distinguir entre hasta cuatro tipos distintos de controlador:

2.4.3.1 *Controlador Proporcional.*

O también conocido como controlador P, es el más simple de los cuatro. A partir de la entrada error obtiene la salida proporcional del error, en otras palabras:

$$U_p(k) = K_c * E(k)$$

Obtiene la salida a partir del producto del error con un parámetro proporcional.

2.4.3.2 *Controlador Proporcional-Integral.*

O también conocido como controlador PI, este controlador añade una acción integral al controlador previo. Esta acción se logra teniendo en cuenta el valor de la señal de salida del ciclo anterior, es decir:

$$U_i(k) = U_i(k - 1) + \frac{K_c * T_s}{T_i} * E(k)$$

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

Como se puede apreciar en la ecuación, ahora la ecuación será la suma de dos términos. El mencionado anteriormente valor de salida del ciclo anterior será utilizado en el ciclo actual, para ello habrá que crear una variable de memoria (se explica de forma detenida en el anejo de programación). Por otro lado una proporcional del error que tendrá en cuenta la proporción (o relación) entre el tiempo de muestreo y el tiempo de integración (aparte del coeficiente proporcional). El tener los dos coeficientes por separado significa que aun en caso de que el error sea igual a cero, nuestro integrador será igual al integrador del ciclo anterior.

Este controlador contará con ambos coeficientes, el proporcional y el integral.

$$U(t) = U_p(k) + U_i(k)$$

2.4.3.3 Controlador Proporcional-Derivativo.

O también conocido como controlador PD, este controlador añade la acción derivativa al controlador. Esta acción tendrá en cuenta la diferencia entre el error del ciclo anterior y el del ciclo actual, lo que equivale a:

$$U_d(k) = K_c * T_d * \frac{E(k) - E(k - 1)}{T_s}$$

Este controlador sin embargo solo dispondrá de un solo coeficiente por lo tanto cuando el error y el error anterior sean iguales nuestro coeficiente derivativo será cero. Nótese que ahora el coeficiente es inversamente proporcional a la relación entre el tiempo de muestreo y el tiempo derivativo. Al igual que el controlador proporcional el controlador derivativo contará con el parámetro proporcional:

$$U(t) = U_p(k) + U_d(k)$$

2.4.3.4 Controlador Proporcional-Integral-Derivativo.

O también conocido como controlador PID, este controlador añade la acción integral junto con la acción derivativa al controlador. Esta acción tendrá los tres coeficientes:

$$U(t) = U_p(k) + U_i(k) + U_d(k)$$

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

2.4.4 Planta.

El sistema enviará la tensión (o VM) al ventilador y este propulsará la plataforma en relación a la tensión suministrada. Entonces, el sensor fotoeléctrico realizará una lectura y devolverá la altura a la que se encuentra la plataforma.

Esta señal de salida se utilizará, mediante una realimentación, para obtener el error del siguiente ciclo. Si se ha realizado correctamente el modelo y el correspondiente controlador, entonces nuestro proceso funcionará correctamente.

3. Motivación y Justificación.

3.1 Motivación

Se ha elegido la temática de control de entre todos los proyectos, debido a su gran atractivo. Al cursar las asignaturas del departamento de DISA en el curso de tercero de GITI, Sistemas Automáticos, Tecnología Automática y de cuarto curso, Laboratorio de Automática y Control fue lo que más interés despertó y motivó su elección sin duda alguna. El poder controlar procesos industriales, diseñar el código de los programas que las máquinas siguen, además de ser intuitivo, resultó gratificante.

Hoy en día la gente utiliza la electrónica y los sistemas de control casi sin darse cuenta, por ejemplo el mando a distancia de la televisión, un microondas, un semáforo, una taladradora, fresadora, un largo etcétera que muchas personas desconocen de su funcionamiento pero que sin ellos la vida diaria sería mucho más compleja.

Desde el descubrimiento de los sistemas de control, estos han ido mejorando a ritmo exponencial. Existen infinidad de procesos de control, desde un programa sencillo como puede ser el temporizador de un microondas, hasta el control de un proceso industrial como puede ser el funcionamiento de una fresadora automática. Siendo esta una materia en la que se puede profundizar y puede suponer verdaderos retos para aquel que decide investigar esta temática.

Tras realizar el proyecto, los conocimientos obtenidos en las asignaturas quedan reforzados y asentados, de tal manera que la visión sobre control cambia completamente y lo que parecía complejo y laborioso en su momento, se vuelve un asunto simple al que simplemente hay que dedicarle tiempo y paciencia.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

3.2 Justificación

De entre todas las ramas del Departamento de Ingeniería de Sistemas y Automática: Automática, Robótica, Filtrado y Control; se decide por Control debido a que trata los procesos industriales de forma individual y es una forma excelente de empezar en el mundo de los procesos industriales. Saber cómo programar lazos de control en procesos industriales ayuda a comprender de cierta forma el resto de ramas. Asimismo, se planteaba un entorno de programación en código gráfico, un lenguaje inédito que se trató en la asignatura de Laboratorio de Automática y Control que terminó resultando peculiar.

Se eligió de entre el resto de proyectos debido a que pese a parecer un proceso simple, al tratar con corrientes de viento puede llegar a resultar complejo. Además el resto de proyectos ofrecidos no acabó despertando tanto interés como este proyecto

4. DESCRIPCIÓN DE LA SOLUCIÓN ADOPTADA.

Para el desarrollo del controlador se han utilizado varios programas, para identificar el modelo, diseñar los controladores y para aplicarlo. Entre ellos el más utilizado ha sido el LabVIEW, mediante el cual se ha diseñado el controlador. Por otra parte, se ha utilizado MatLab para el diseño del modelo del sistema y para la posterior obtención de nuestros parámetros del controlador, se emplearan las herramientas Ident y Rltool que incorpora el programa.

Para una mejor interacción con el programa se realiza una interfaz amigable y clara con los controles necesarios. Como se puede observar en la imagen siguiente (Ilustración 13 - Interfaz de control):

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

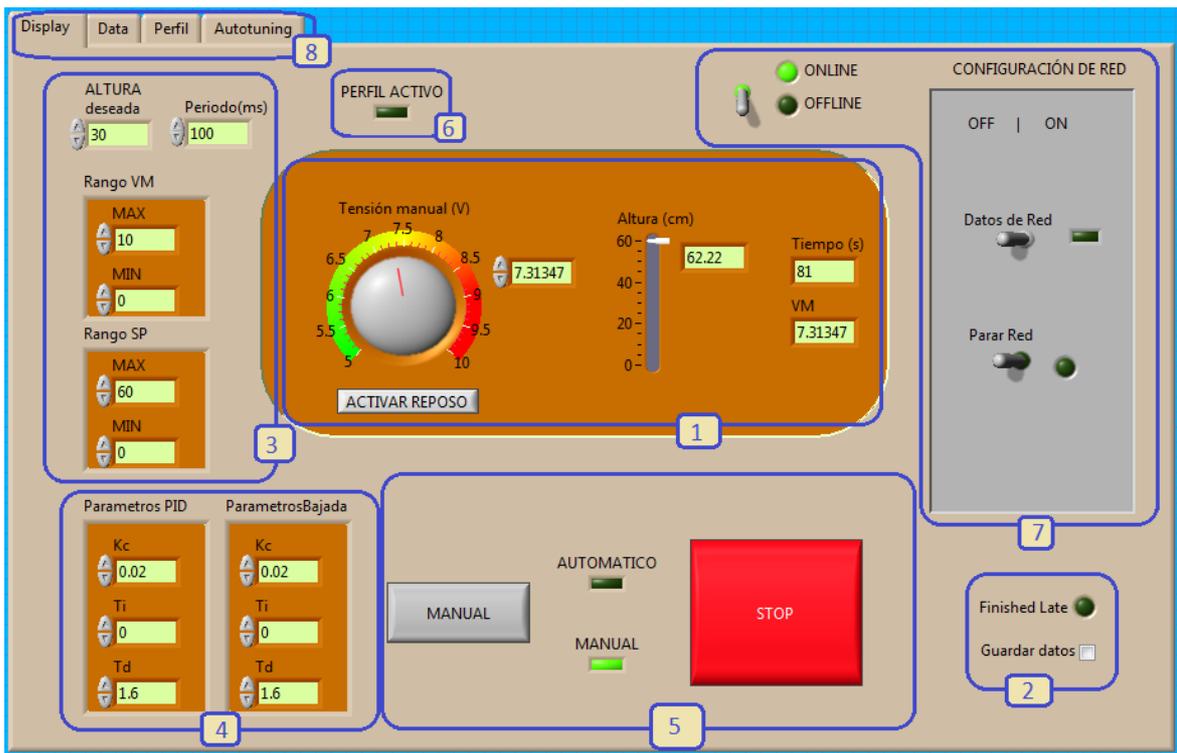


Ilustración 13 - Interfaz de control

Así, de un vistazo se pueden identificar diversas herramientas de las que se dispone. Desde el control manual hasta el control del proceso mediante PID. En esta memoria se ha realizado una explicación breve y sencilla del trabajo realizado a lo largo del proyecto. Para obtener más información acerca del trabajo realizado, se ruega remitirse a los anexos adjuntos.

4.1 Implantación del Control Manual mediante LabVIEW.

Se decide introducir un sistema de control manual sobre el prototipo para que el usuario pueda controlar de forma directa la tensión introducida en el ventilador. La forma de control de la tensión será mediante una ruleta o dial y con una pantalla digital que permite la introducción directa de la tensión en ésta además de mostrar la tensión suministrada al ventilador en todo momento. En la imagen anterior (Ilustración 13 - Interfaz de control) se puede apreciar dicha ruleta y a su lado una pipeta, que será la encargada de mostrar la altura a la que se encuentra la plataforma mediante la aguja y el panel digital de la derecha.

La entrada de datos al sistema es mucho más simple que la salida de datos, el rango de las unidades de salida (Voltios) no necesita de traducción alguna, ya que la TAD emite ya en unidades de tensión que coinciden además con el mismo valor que la entrada del

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

ventilador. No obstante, la salida de datos del dispositivo viene dado en una escala de valores del -2.35 al 9.3 uds, este pequeño inconveniente se soluciona mediante la inserción de un sistema de ‘traducción’, un interpolador que transforma la lectura de nuestro sensor en un valor del 0-60 cm (que realmente son -0.7 y 62.2 aproximadamente), la altura a la que se encuentre nuestra plataforma.

Junto a la programación del modo manual se instaura el modo de grabación y el modo gráfico. El modo de grabación procede a grabar los datos siempre y cuando la variable de condición esté activada mientras que el modo gráfico mostrará por pantalla los datos que alcanzan la tensión y la altura a lo largo del proceso mediante dos gráficas. También se han adjuntado distintos paneles y leds con información útil para el usuario y además un botón de reposo, que cambiará la tensión suministrada por la tensión en la que la plataforma se mantiene a una altura razonablemente constante.

4.1.1 Función de Parado.

Durante la realización del proyecto se ha observado los golpes y el desgaste que sufre la plataforma por los cambios bruscos de tensión. Debido a que el programa grababa el último valor de la tensión se decidió poner la tensión a cero antes de finalizar el programa. No obstante, la plataforma descendía en caída libre.

Por precaución se ha añadido un sistema de apagado que tras presionar el botón de paro, durante 7 segundos se suministra una tensión inferior a la denominada tensión de reposo, para que la plataforma descienda suavemente hasta la parte inferior del tubo, transcurridos los 7 segundos, el ventilador se apaga. Es posible modificar este perfil y cambiar el tiempo que tarda o la tensión suministrada durante ese tiempo. De hecho se recomienda al usuario que actualice la tensión si esta no es inferior a la tensión de reposo.

4.2 **Implantación del Control Mediante PID a través de LabVIEW.**

Además de la función manual también se desea instaurar una función automática para que el dispositivo funcione sin la intervención directa de un usuario.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

El programa pedirá la altura deseada (o SetPoint) a la que queremos que se sitúe la plataforma (y se introducirá en **cm**). Se ha realizado el diseño de una herramienta VI que actúe como PID para éste proyecto. Necesitará de la alimentación de datos, la Variable Medida Manual, el ManAuto, el SetPoint, la Variable Controlada, los rangos de SP y Variable Medida, el periodo (T_s) y los parámetros de nuestro PID, y devolverá el valor correspondiente de la Variable Manipulada. Gracias a la configuración de este PID, hemos podido incluir ambas formas de uso, manual y mediante controlador. Se ha habilitado un interruptor ManAuto para cambiar de una a otra.

La función del controlador PID funciona gracias a las ecuaciones introducidas (explicación detallada dentro del anexo de programación y de diseño) pero han de obtenerse los parámetros del PID adecuados. Para ello se realizará un estudio y el correspondiente diseño a nuestro modelo.

4.3 Obtención del Modelo y Diseño del Controlador PID.

4.3.1 Obtención del Modelo.

Para la obtención del modelo de nuestro sistema, utilizaremos el programa MatLab, en concreto la herramienta Ident, la cual nos permitirá extraer el modelo de nuestro sistema mediante unos datos que se deberán introducir.

Estos datos se conseguirán gracias a la acción de grabar (mencionada antes en el apartado de función manual). Se procederá a la realización de un ensayo con escalones en la VC, el cual consistirá en desplazar la plataforma desde el punto inferior del tubo hasta al superior. Una vez arriba se dejará reposar unos segundos y se procederá a realizar una caída libre mediante un escalón negativo. Se realizan varios escalones de diferentes magnitudes para poder extraer diferentes modelos y poder elegir el que más nos convenga (se realiza una explicación más detallada en el documento Anexo de Diseño).

Existen distintas configuraciones para el modelo, el que más se adapta sin embargo, es el de primer orden con integrador; además si se le añade retraso o 'delay' se puede observar que el modelo se adapta casi perfectamente a los datos obtenidos del ensayo. Tras el trabajo realizado en la herramienta de MatLab Ident se obtiene nuestro modelo:

$$G(s) = \frac{K_p}{s(1 + T_{p1} * s)} : K_p = 49.353; T_{p1} = 1.6104;$$

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

4.3.2 Diseño del Controlador PID.

Una vez obtenido el modelo de nuestro sistema, será necesario diseñar su controlador correspondiente. Para esta tarea, se utilizará la herramienta de MatLab SISOTool, la cual nos permite desarrollar el sistema de las raíces y modificarlo a nuestro antojo.

Esta herramienta permite una obtención de los parámetros de los controladores bastante sencilla y gráfica. Tras examinar los comportamientos del sistema con un controlador P y uno PD, se opta por el PD, debido a la lentitud del controlador P.

Pese a no utilizarse en nuestro proyecto, nuestro programa es capaz de diferenciar entre escalones positivos y negativos, esto quiere decir que, en caso de comportarse de forma distinta entre subir y bajar (una diferencia entre la referencia y la altura actual de la plataforma positiva o negativa), el sistema se podría configurar para un mejor comportamiento.

Esta herramienta se implementó previamente al modelado, tras éste se observó que era innecesaria, pero como no interfería en los cálculos ni el proceso así que se dejó instaurada. Tras el trabajo realizado en la herramienta de MatLab SISOTool se obtiene nuestro controlador:

$$G_{R_{PD}}(s) = \frac{K_p(1 + T_d * s)}{1} : K_p = 0.0145; T_d = 1.6104;$$

4.4 **Implementación de diversas Herramientas.**

Para una mejor complementación del sistema y un mejor funcionamiento del proceso se decide añadir diversas herramientas que pueden servir de ayuda al usuario.

4.4.1 Perfil.

Para la comprobación del funcionamiento del PID se decide instaurar un perfil, el perfil está configurado para que varíe la referencia introducida sin necesidad de que la cambie el usuario (la instauración y configuración pueden verse con más detalle en el anejo de programación). Para activar el perfil el programa ha de estar en modo automático.

Pese a ser utilizado como herramienta de comprobación, el perfil puede ser modificado para introducir un perfil de funcionamiento que el usuario desee. A continuación se muestran dos gráficas en las que se pueden apreciar el funcionamiento del perfil:

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

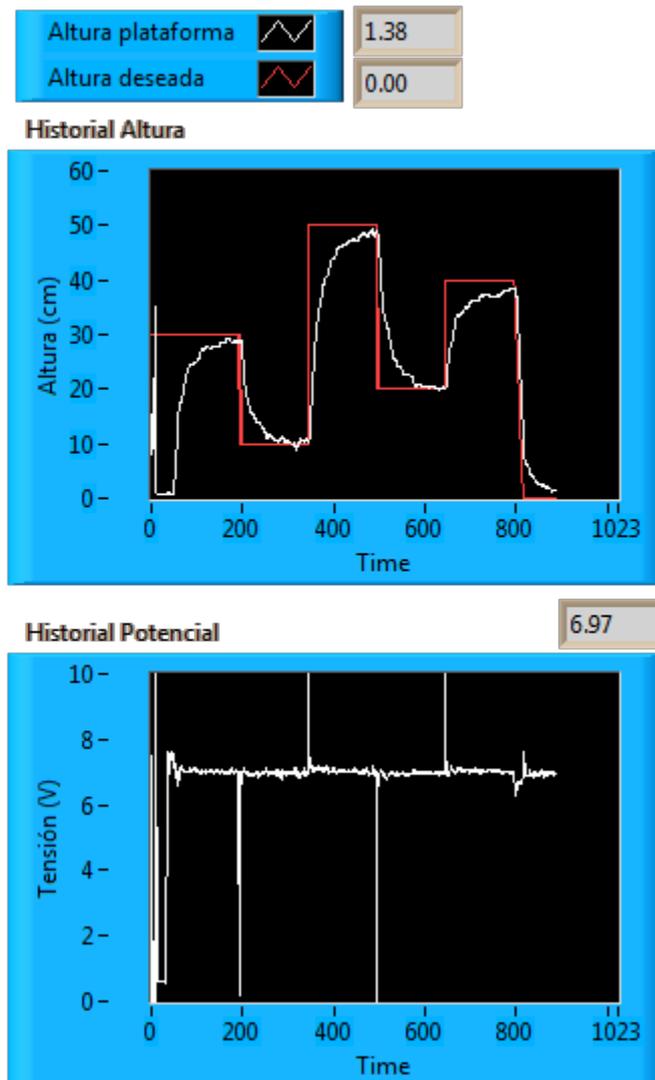


Ilustración 14 – Comprobación mediante Perfil

4.4.2 Autotuning

Para que el usuario pueda apreciar la diferencia entre un controlador PID configurado manualmente y uno configurado por una máquina, se introduce una herramienta VI que proporcionaba ya el programa para poder realizar un contraste entre ambos. El primer contra que se observa será que el autotuning no dispone de controladores PD para configurarlos. Dejando solo la opción de emplear un controlador P que resultará de comportamiento lento. En el documento Manual del Usuario se explica con mayor determinación el funcionamiento del Autotuning.

Desarrollo de un Sistema de Control de la Posición de una Plataforma en Suspensión mediante LabVIEW (Memoria)

5. BIBLIOGRAFÍA

La mayor parte de la información se ha obtenido de Internet, se ha contrastado con otras fuentes no escritas aquí para contrastar la veracidad de estas.

Apuntes asignatura: “Laboratorio de Automática y Control: Apartado de Control”

Autor: Juan Manuel Herrero Durá

Título: Manual de Labview

Autor: NI INSTRUMENTS <http://www.ni.com/pdf/manuals/320999e.pdf>

Título: Sistemas de control

http://www.juntadeandalucia.es/averroes/~23005153/d_tecnologia/bajables/2%20bachillerato/SISTEMAS%20AUTOMATICOS%20DE%20CONTROL.pdf

Wikipedia:

Sistemas de control:

http://es.wikipedia.org/wiki/Sistema_de_control

LabVIEW:

<http://es.wikipedia.org/wiki/LabVIEW>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°2 - ANEXO DE
PROGRAMACIÓN**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

ANEXO DE PROGRAMACIÓN

1.	INTRODUCCIÓN A LABVIEW	1
2.	CONTROL MANUAL	1
2.1	VIs para la conexión con la Tarjeta de Adquisición de Datos.....	1
2.2	Linealización de los Datos del Sensor	2
2.3	La Interfaz Básica del Control Manual.....	3
2.4	Muestreo de Datos mediante Gráficas	4
2.5	Sistema de Grabado.....	5
2.6	Sistema de Apagado.....	6
3.	CONTROL AUTOMÁTICO.....	6
3.1	Implementación de un Controlador PID al Sistema	7
3.1.1	Construcción de una Herramienta PID	7
3.1.2	Preconfiguración de la Herramienta PID.....	9
3.1.3	Configuración de la Herramienta PID	10
3.2	Comprobación mediante Perfil.....	11
4.	FUNCIONES ADICIONALES.....	12
4.1	Autotuning	12
4.2	Control por Red.....	14

Debido a las demandas de nuestro cliente, la programación del control de nuestro dispositivo, se realizará mediante LabVIEW. Este anexo se ha realizado para la explicación detallada de la construcción de nuestro programa diseñado. Además se detallará el funcionamiento de los distintos modos de funcionamiento de los que dispone: control manual, control por PID, autotuning, control por red, planificación de ganancia...

1. INTRODUCCIÓN A LABVIEW

La plataforma LabVIEW (acrónimo de Laboratory Virtual Instrument Engineering Workbench) nos permite diseñar programas mediante un lenguaje de programación gráfico. Una de las ventajas que presenta esta plataforma es que posee una amplia variedad de herramientas preinstaladas con el programa, además ofrece la posibilidad de añadir nuevas bibliotecas para adaptar la comunicación a dispositivos externos, como en nuestro caso, la Tarjeta de Adquisición PCI 9112 de AD-LINK. A continuación se realizará una explicación del programa desarrollado, de su interfaz y de sus diversas herramientas:

2. CONTROL MANUAL

2.1 VIs para la conexión con la Tarjeta de Adquisición de Datos.

Para la creación del control manual del programa será necesario la lectura y la escritura a través de la tarjeta gráfica. Para poder realizar la escritura y la lectura es necesario disponer de una librería adicional para LabVIEW, dicha librería fue proporcionada por los responsables del laboratorio. Fue instalada previamente en los ordenadores del laboratorio en la dirección: C:\ADLINK\DAQ-LVIEW PnP\LLB

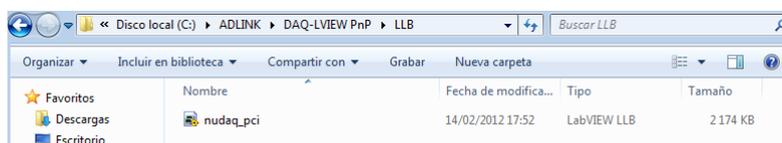


Ilustración 1 - Dirección de la Librería

En esta librería se encuentran las herramientas necesarias para realizar la conexión entre el ordenador y la tarjeta de adquisición. Las herramientas entrada/salida de nuestra tarjeta de adquisición se encuentran en esta librería, una vez introducidas en nuestro código solo es necesario introducir el canal y el número del dispositivo. En nuestro caso, debido a las conexiones realizadas en el regletero, el canal será 0 y el n° será 1.

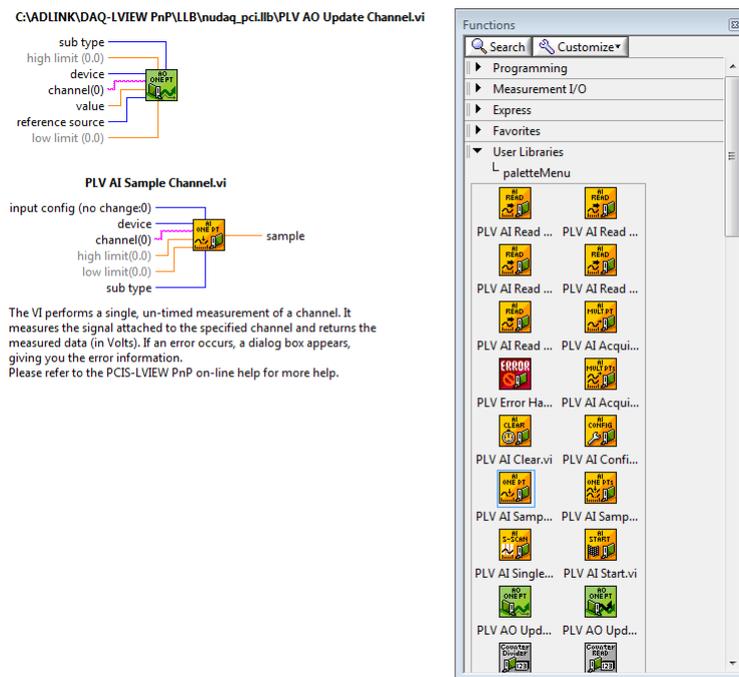


Ilustración 2 - VIs Entrada/Salida

Una vez instalados los nuevos VI (Virtual Instrument) se dispone a analizar la señal obtenida.

2.2 Linealización de los Datos del Sensor

Tras analizar la lectura de datos, nos damos cuenta de que la señal que envía el sensor fotoeléctrico, no está calibrada y no es lineal, por ello, los datos obtenidos del sensor deben ser procesados para que en la interfaz se muestren la altura en cm y en el rango de 0-60 cm en los que viene dividido el tubo.

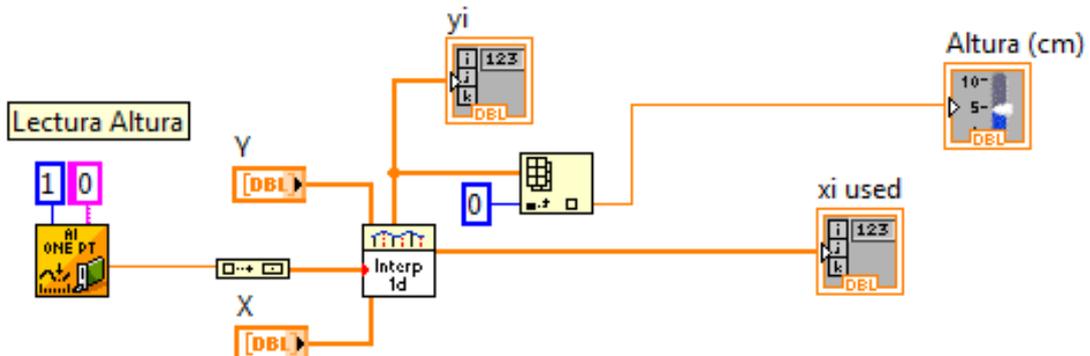


Ilustración 3 - Linealización de la Señal de Entrada

Se ha adaptado una herramienta de LabVIEW para que proporcione la lectura del sensor, los datos leídos son introducidos a un interpolador que devolverá la altura real a la que se encuentra la plataforma. La altura será mostrada en pantalla en un indicador que se ha explicado anteriormente y en la segunda página de la interfaz donde se haya también el interpolador y sus datos. La explicación de la interfaz es algo que concierne más al manual del usuario que a esta sección, por lo tanto para más información remítase al manual.

Mediante una serie de ensayos se han realizado diversas medidas a diferentes alturas y se ha anotado la lectura del sensor correspondiente. La herramienta interpoladora calculará la altura a la que se encuentra la plataforma mediante una interpolación entre estos datos. Se han introducido 10 valores con sus equivalentes alturas para una mayor precisión a la hora de estimar de la herramienta a través de los datos. Mostrará también la lectura del sensor y la altura que le corresponde. (Siendo Y_i la altura correspondiente a X_i la lectura del sensor).

2.3 La Interfaz Básica del Control Manual.

Una de las funciones implantadas en nuestro programa es la posibilidad de controlar la tensión suministrada al ventilador directamente. Aunque el programa empezó tan solo con funcionamiento manual, su desarrollo ha ido combinando todos los funcionamientos. Tras los numerosos cambios al código, la acción manual acaba resultando un poco mezclada con el programa entero, no obstante, a continuación se muestran imágenes del código del control manual:

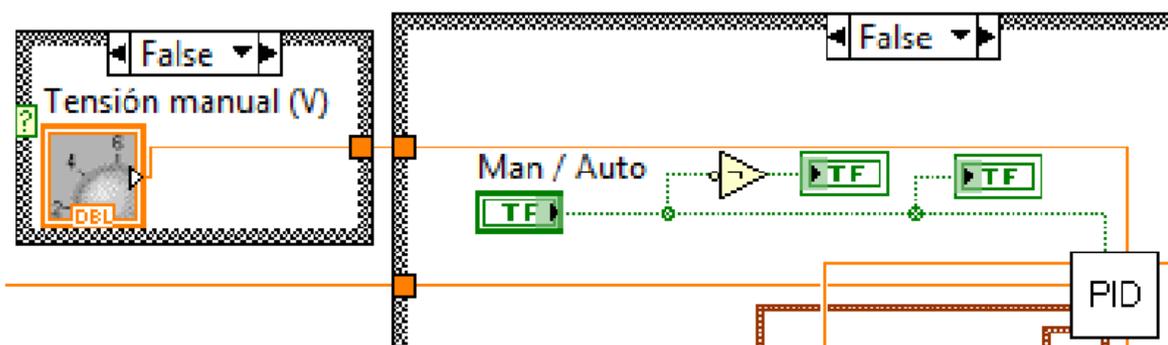


Ilustración 4 - Control Manual (Salida)

En la (Ilustración 4 - Control Manual (Salida)) se puede observar una dial, es una herramienta visual que sirve para introducir una tensión situada en el rango representado (5-10) ya que se desea una precisión mayor en torno a 7V que será la tensión necesaria para el reposo.

Es preciso evitar tensiones excesivamente grandes o pequeñas puesto que hacen que nuestro sistema actúe violentamente; ante una entrada menor a 5V, el sistema desciende prácticamente en caída libre y con más de 9V la plataforma asciende tan rápido que golpea bruscamente los límites instalados en la parte superior (más adelante se procede a establecer límites superiores e inferiores al sistema); por ello, se decide imponer límites en la ruleta para evitar un cambio brusco involuntario debido a una mala selección con el ratón.

Límite inferior de 5 para evitar caídas libres y límite 10 porque es el máximo admisible (también es posible introducir la tensión en la pantalla digital, para más información consultar en Manual del Usuario). El código creado permite actuar de modo manual, cuando los interruptores 'AUTOTUNING' y 'Man/Auto' están ambos apagados.

Al estar el interruptor Autotuning apagado, se activa el PID diseñado, y dentro de sus funciones, el PID funcionará en estado manual. Por otra parte, con el interruptor en la parte central-derecha de la imagen encontramos un indicador con forma de tubo o pipeta que mostrará la altura a la que se encuentra la plataforma analógicamente y se mostrará el valor digital en la pantalla de su lado.

2.4 Muestreo de Datos mediante Gráficas

Tras la obtención de los datos de entrada del sensor (VC altura) y de los datos de salida (VM tensión), se decide representarlos en un gráfico para un mejor entendimiento del proceso. Estas gráficas mostrarán el valor de la variable manipulada y de la variable controlada en cada ciclo. Más adelante se añadirá junto a la variable controlada la Referencia, para un mejor contraste. Tal como muestra la imagen siguiente (Ilustración 5 - Gráficas de Datos):

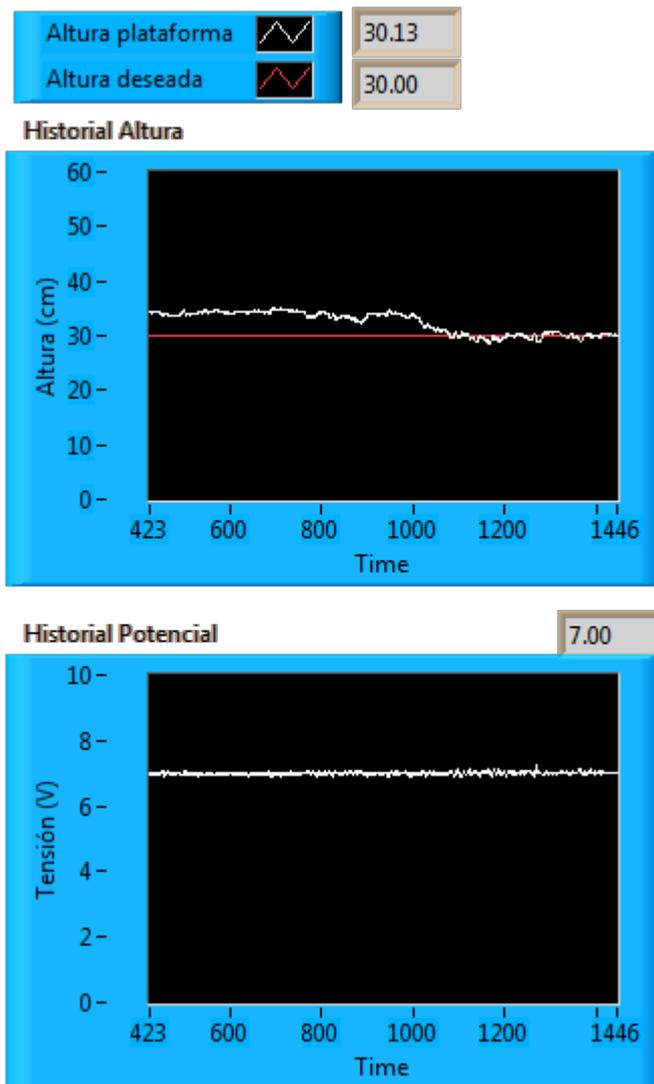


Ilustración 5 - Gráficas de Datos

En la gráfica de abajo podemos observar los registros de la tensión que se ha ido suministrado al ventilador (VM) durante el ensayo, mientras que en la gráfica de arriba podemos observar la altura que la plataforma alcanzaba, además la altura que se desea alcanzar también viene representada en la misma gráfica para un mejor contraste (la altura deseada es aquella a la que queremos que el automático sitúe nuestra plataforma, más adelante se dará una explicación más extensa).

2.5 Sistema de Grabado.

Junto a la introducción de las gráficas, se añade además un código para grabar los resultados obtenidos en el ensayo. Gracias a ello se podrá diseñar un modelo del sistema a partir de los datos obtenidos. (Más información en el Anejo de Diseño)

3.1 Implementación de un Controlador PID al Sistema

3.1.1 Construcción de una Herramienta PID

Para la realización del PID, debido a su extensión, se ha decidido crear una VI auxiliar a nuestro código (también adjuntada). Para ello se emplearán las ecuaciones siguientes:

$$\begin{aligned}U_p(k) &= K_c * E(k) \\U_i(k) &= U_i(k - 1) + \frac{K_c * T_s}{T_i} * E(k) \\U_d(k) &= K_c * T_d * \frac{E(k) - E(k - 1)}{T_s} \\U(t) &= U_p(k) + U_i(k) + U_d(k)\end{aligned}$$

Como se ha explicado en la Memoria, estas ecuaciones son la definición de nuestra herramienta, con ella crearemos un proporcional que nos permita obtener el valor de la variable medida en función de la referencia introducida. Como podemos ver en la imagen siguiente (Ilustración 7 - PID diseñado):

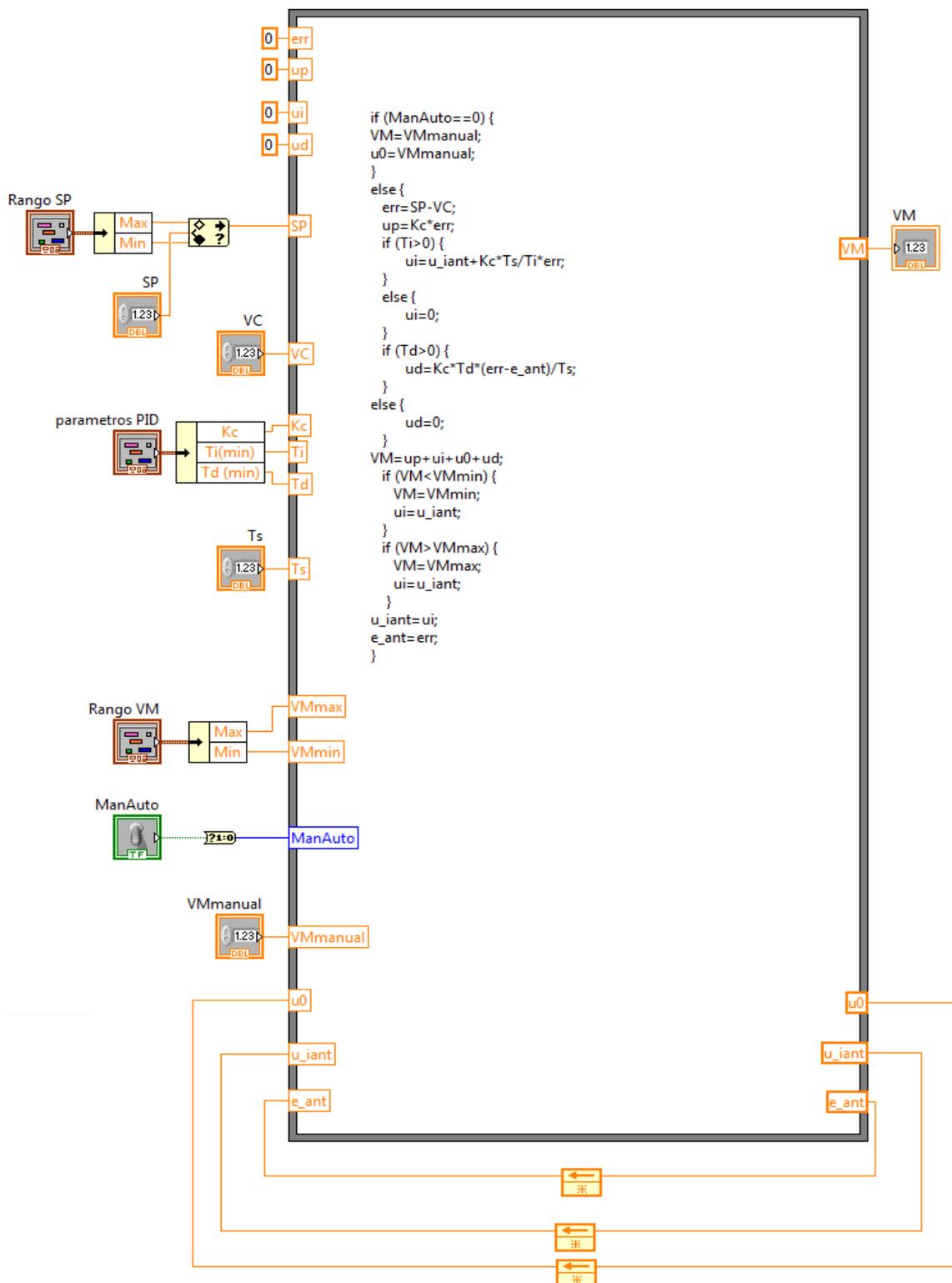


Ilustración 7 - PID diseñado

3.1.2 Preconfiguración de la Herramienta PID

Dispone de una configuración que le permite la introducción de los parámetros proporcional, derivativo e integral y mediante ellos obtener el valor correspondiente a una referencia entrada. Esta herramienta funciona mediante programación en C. Las ecuaciones se modifican para que coincidan con el nombre de nuestras variables y se añade la opción de si un parámetro está a cero, el código de éste no se active.

Para el funcionamiento de las acciones integral y derivativa son necesarias unas memorias auxiliares para grabar el valor de las variables error y salida del ciclo anterior que deberán ser inicializadas a cero al principio del código pero que gracias a la VI Feedback Node no será necesario ya que este se encarga de inicializarlo a cero.

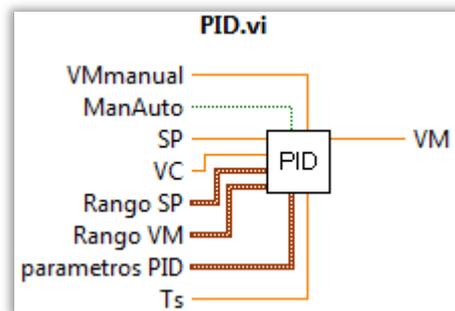


Ilustración 8 - SubVI PID

Tras la creación de nuestro PID, se procede a la preconfiguración de éste. Para poder hacer uso de una herramienta VI mediante otra externa, se necesita designar las entradas y salidas pertinentes. Las entradas serán las siguientes: Rango SP, Rango VM, parámetros PID, VManual, ManAuto, SP, VC y T_s (periodo). Por otra parte de salida solo habrá VM.

La explicación del porqué de las entradas y salidas es la siguiente:

Las restricciones de rango vienen dadas para evitar que se alcancen valores superiores a los establecidos, se limita la VM para evitar sobrecargas en el ventilador, y el SP porque es contraproducente ya que el tubo solo dispone de 60 cm de largo.

Se introducen los parámetros K_c , T_i y T_d mediante el cluster 'parámetros PID', en los cuales introduciremos los parámetros obtenidos en el apartado de diseño (Anexo de Diseño)

Se introducen también VManual y ManAuto, para que pueda funcionar nuestro PID de manera manual. ManAuto será la variable booleana que sirva para cambiar entre manual y

automático (0 – Manual y 1 – Automático) y VManual tendrá el valor que queramos que tenga la tensión al actuar de forma manual. Al estar desactivada ManAuto la VM de salida será igual a VManual

Por otra parte, SP, VC y Ts son introducidas para que el PID pueda funcionar, son parámetros de funcionamiento que, junto a los parámetros PID variarán el comportamiento del controlador. SP (setpoint o punto de funcionamiento) indicará la altura a la que queremos que se posicione nuestra plataforma, VC (o Variable Controlada) será el valor actual de dicha altura y Ts será el tiempo transcurrido.

Por último, y no por ello menos importante, VM (o Variable Manipulada) la cual será proporcionada por nuestro controlador, tras realizar los cálculos necesarios.

Quedará tal como se muestra en la figura (Ilustración 8 - SubVI PID), con las entradas necesarias a la izquierda, abajo y arriba y la salida a la derecha. 8 entradas y 1 salida,

3.1.3 Configuración de la Herramienta PID

La instauración de la VI PID queda tal que:

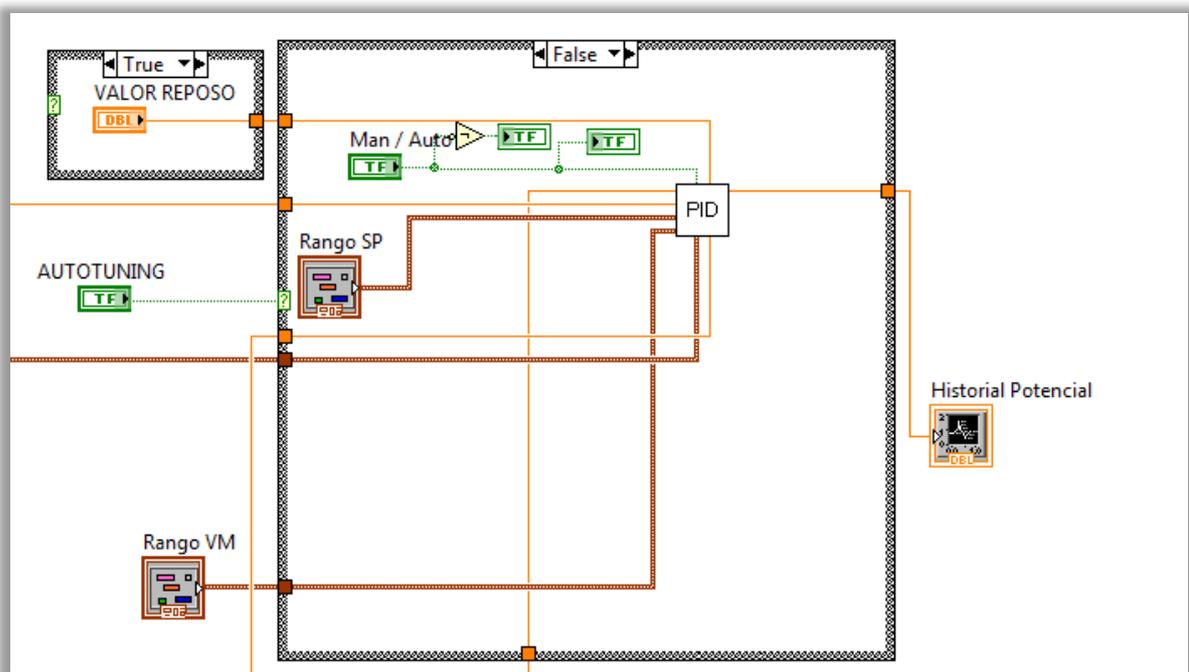


Ilustración 9 - Función PID

En la imagen se puede distinguir entre las distintas entradas que dispone el PID (en la imagen que aparece al final del documento se puede apreciar con mejor claridad), la entrada referencia, la VManual... todas las entradas mencionadas anteriormente. La salida VM, conecta con un gráfico y con la VI encargada de enviar la información enviada al ventilador

Para que el sistema fuera capaz de distinguir entre escalones positivos y escalones negativos, se introdujo un Case Structure ligado a un comparador entre la referencia y la variable controlada para que en caso de que la condición (SP-VC) fuera mayor que cero estaríamos ante un escalón positivo y si fuera menor que cero ante uno negativo. En el anexo de diseño se especifican los valores de estos parámetros.

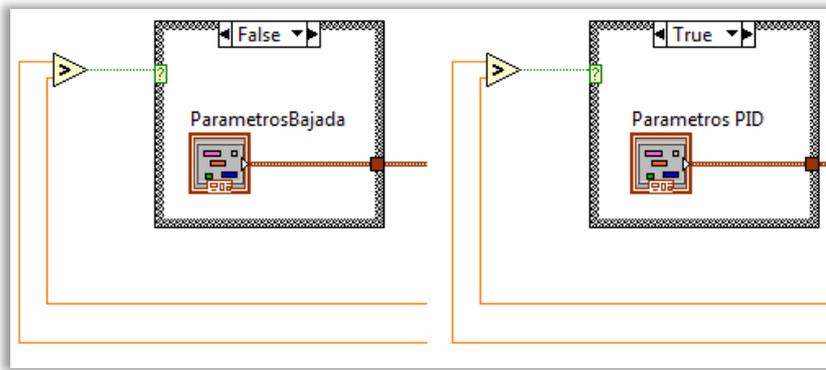


Ilustración 10 - Distinción entre escalones

3.2 Comprobación mediante Perfil.

Para la comprobación del funcionamiento del PID, se introducirá una herramienta capaz de suministrar distintos valores de salida para distintos tiempos.

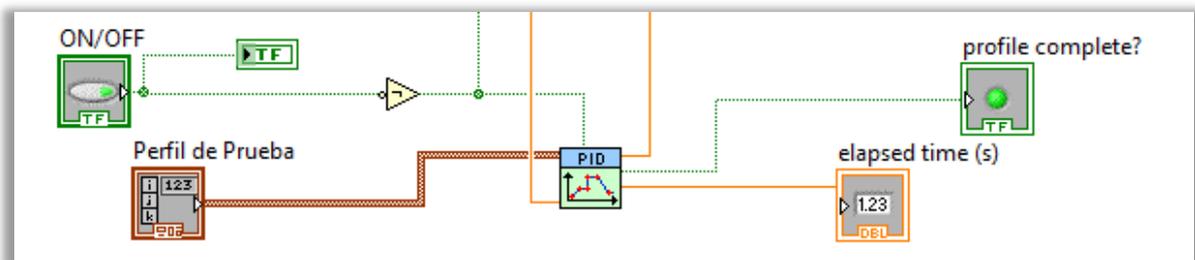


Ilustración 11 - Perfil de Prueba

Se dispone al sistema de un interruptor para encender la función del perfil. Se mostrará por pantalla una señal luminosa para avisar al usuario de que el perfil se está llevando a cabo. Durante este tiempo el perfil introducirá el valor que se le ha configurado para que suministre. En la Tab denominada perfil se encuentra la configuración del perfil y puede ser configurada no solo para realizar un test de comprobación sino también para seguir una serie de datos suministrada por el usuario. Una vez finalizado el perfil hay que presionar de nuevo el interruptor ON/OFF para que deje de leer el perfil.

4. FUNCIONES ADICIONALES.

Para la mejora del producto final se ha decidido implantar dos herramientas auxiliares al proceso, que pueden mejorar su funcionamiento. Entre ellas podemos encontrar el autotuning y el control por red. Son distintos modos de uso para que el usuario pueda escoger el modo en el que quiere que funcione la máquina.

4.1 Autotuning

Se ha creado este modo de empleo para que el usuario pueda utilizar otro controlador distinto al nuestro, un controlador creado por una herramienta VI auxiliar que facilita LabVIEW. En el ensayo se procede a realizar una prueba de dicha subVI para contrastar los resultados entre utilizar el controlador diseñado o el control que le proporcione el programa.

Para ello, se ha creado un Case Structure para que funcione o bien el PID diseñado o bien el obtenido por la herramienta autotuning. Para que el autotuning funcione correctamente es necesario administrarle unos parámetros previos aproximados (PID gains), los parámetros del autotuning (autotuning parámetros) y dos interruptores que activarán su funcionamiento. Devolverá los parámetros modificados como PID gains out, los cuales se realimentarán a los que se han puesto en la entrada (PID gains).

Uno de los grandes inconvenientes de esta herramienta es que solo puede aproximar controladores P, PI, y PID. Al ser un modelo con integrador no es conveniente añadir otro integrador, el sistema se vuelve más complejo, además de que introducir otro integrador no es necesario ya que no existe error de posición. Por lo tanto solo se podría elegir un Proporcional resultando un coeficiente proporcional demasiado bajo (como se puede ver en el anejo de diseño).

El programa funciona mediante la introducción de escalones a la referencia, de altura y repeticiones especificadas en los autotuning parameters, en los parámetros también se especificará la naturaleza del sistema (o bien rápida o bien lenta). Primero el programa calcula el ruido del sistema y tras ello procede a la introducción de escalones y tras examinar cómo reacciona el proceso, devolverá unos valores.

Esta herramienta se puede y se debe ejecutar más de una vez, para que mejore la aproximación de estos parámetros, una vez que los parámetros dejen de variar significa que los parámetros no se pueden aproximar más. Por lo tanto serán estos últimos parámetros los obtenidos de la herramienta que se considerarán los parámetros del programa.

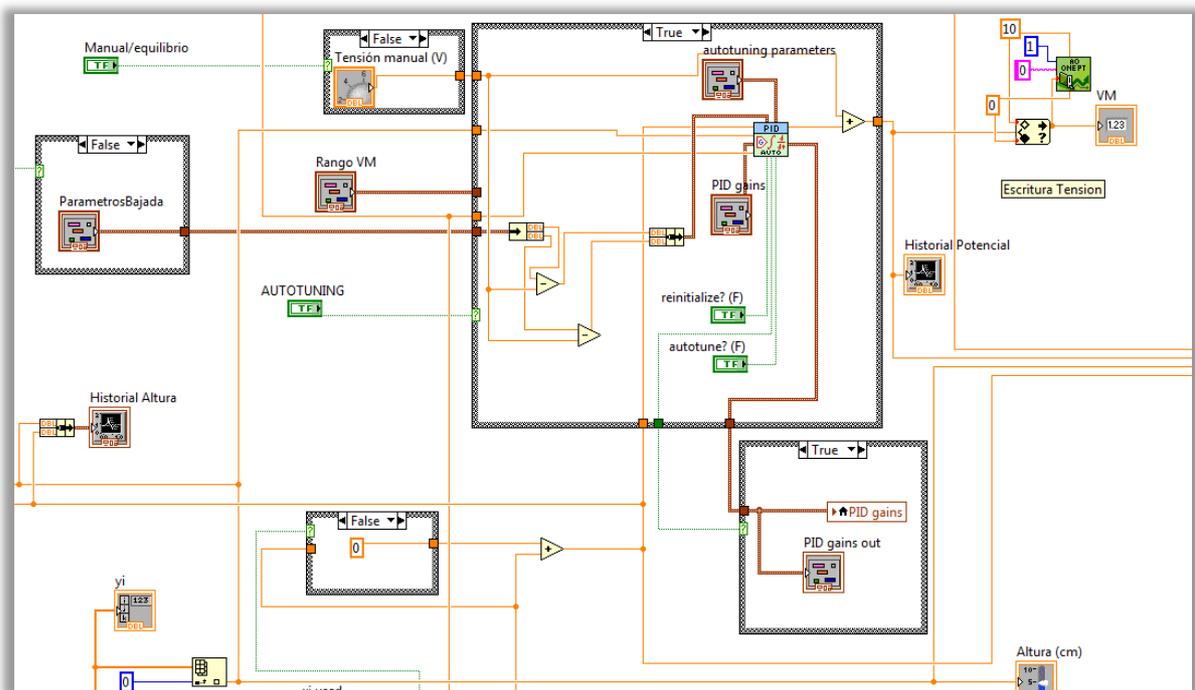


Ilustración 12 - Autotuning

En la imagen anterior (Ilustración 12 – Autotuning) se puede apreciar la integración de la función en el código y la unión con los parámetros que la configuran.

Tras obtener los parámetros mediante el autotuning y contrastarlos con los que hemos obtenido que el PID diseñado se adapta mejor al comportamiento del sistema. No obstante se deja a juicio del usuario el emplear esta herramienta si así lo desea.

4.2 Control por Red.

Para poder controlar el prototipo de forma remota es necesario implementar una VI para poder recibir datos de otro ordenador conectado mediante Ethernet. Será menester además disponer de un programa (que también se adjuntará) para poder enviar los datos. Mientras que un ordenador envía los datos (el servidor) el otro, el que dispone del prototipo será el que reciba los datos (el cliente). Ambas herramientas han sido creadas para las conexiones de red y adaptadas al programa para su funcionamiento.

En la siguiente imagen (Ilustración 13 - Control por Red (Servidor) se muestra el código del servidor, y las modificaciones realizadas para una mejora adaptación a nuestro programa, como la inserción de los límites de la entrada de los valores (de 0 a 60) y un redimensionamiento a los elementos más importantes (la ruleta y el botón de paro)

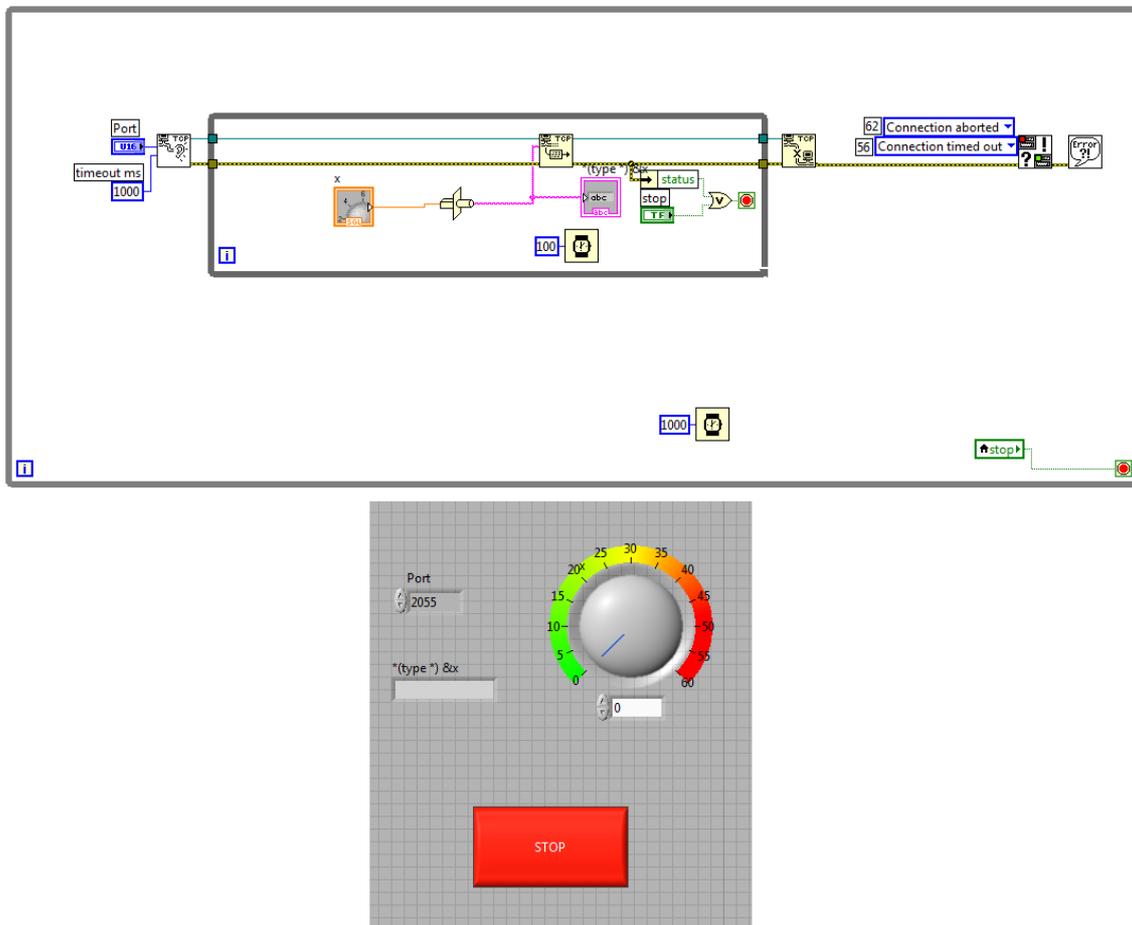


Ilustración 13 - Control por Red (Servidor)

Es necesario (como se indica en el Manual del Usuario) que el servidor esté conectado antes de encender el cliente para que la conexión se realice correctamente, debido a la escritura del código.

Para el cliente se ha añadido un Time Loop aparte para que se produzcan simultáneamente con el proceso. Asimismo, será necesaria la introducción de nuevos Case Structure para poder compatibilizar la conexión con red con el resto de modos de uso. El primer problema sucede cuando se intenta arrancar el programa sin un servidor. Al no encontrarlo, el programa devuelve error y será difícil controlar el proceso. Como consecuencia será necesario encerrar la parte del código que solicite la entrada de datos del servidor dentro de un Case Structure, para que en caso de no haber servidor, esta comprobación no se lleve a cabo.

En el programa fuente se introduce una variable externa para poder lograr el uso de los datos obtenidos por red. Será introducida de nuevo en un Case Structure para que el usuario decida si usar o no los datos recibidos por red.

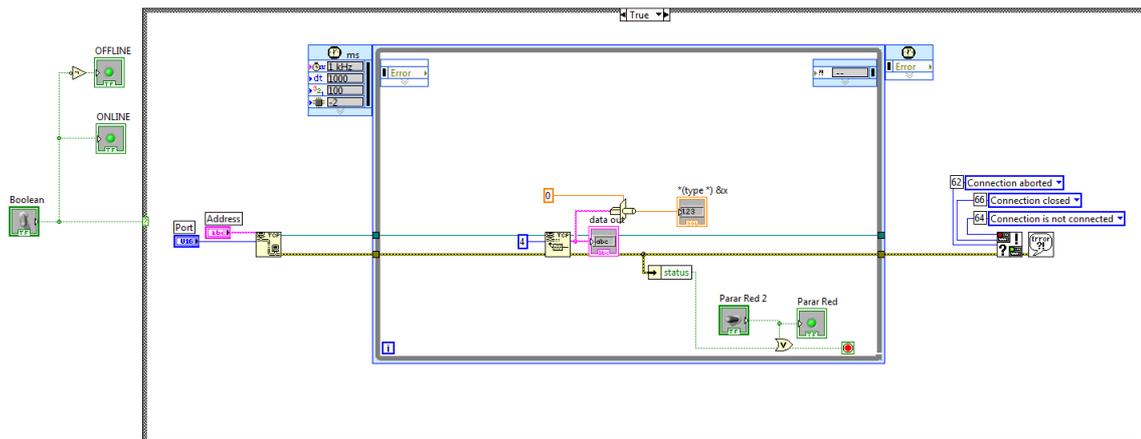


Ilustración 14 - Control por Red (Servidor)

Como se puede observar en la imagen, se ha añadido un botón de paro que estará relacionado con el botón de paro general. Esta relación se lleva a cabo para que ambas estructuras terminen a la vez. No obstante el botón de paro de esta nueva estructura, parará de forma crítica nuestro proceso, pese a poder ocultarse, se decide dejarlo visible y que sea decisión del usuario el parar o no las comunicaciones por red.

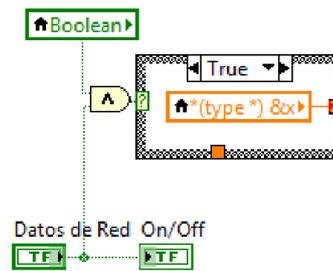


Ilustración 15 - Datos por Red

Se han añadido dos interruptores booleanos para que el usuario pueda elegir si quiere recibir los datos y usarlos o solo recibirlos. Es necesario tener ambos interruptores encendidos para que se produzca la utilización de los datos recibidos, en el caso de que 'datos de red' este apagado y 'boolean' este encendido los datos se recibirán pero no serán utilizados, pero si control por red está apagado no se utilizará la variable de datos por red.

Pese a que se ha logrado un funcionamiento casi sin errores, en ocasiones al conectar el programa sin haber conectado previamente el servidor, puede dar error al finalizar el programa. No obstante, el programa se lleva a cabo sin ningún fallo o dificultad. Para el correcto funcionamiento del programa, el usuario tiene que verificar que los tres interruptores de la configuración por red se encuentren apagados antes de poner en marcha el programa.

Para finalizar se adjuntan tres imágenes del código completo, en ellas se puede apreciar el código de control por red y el código principal con las diferentes opciones de los Case Structure

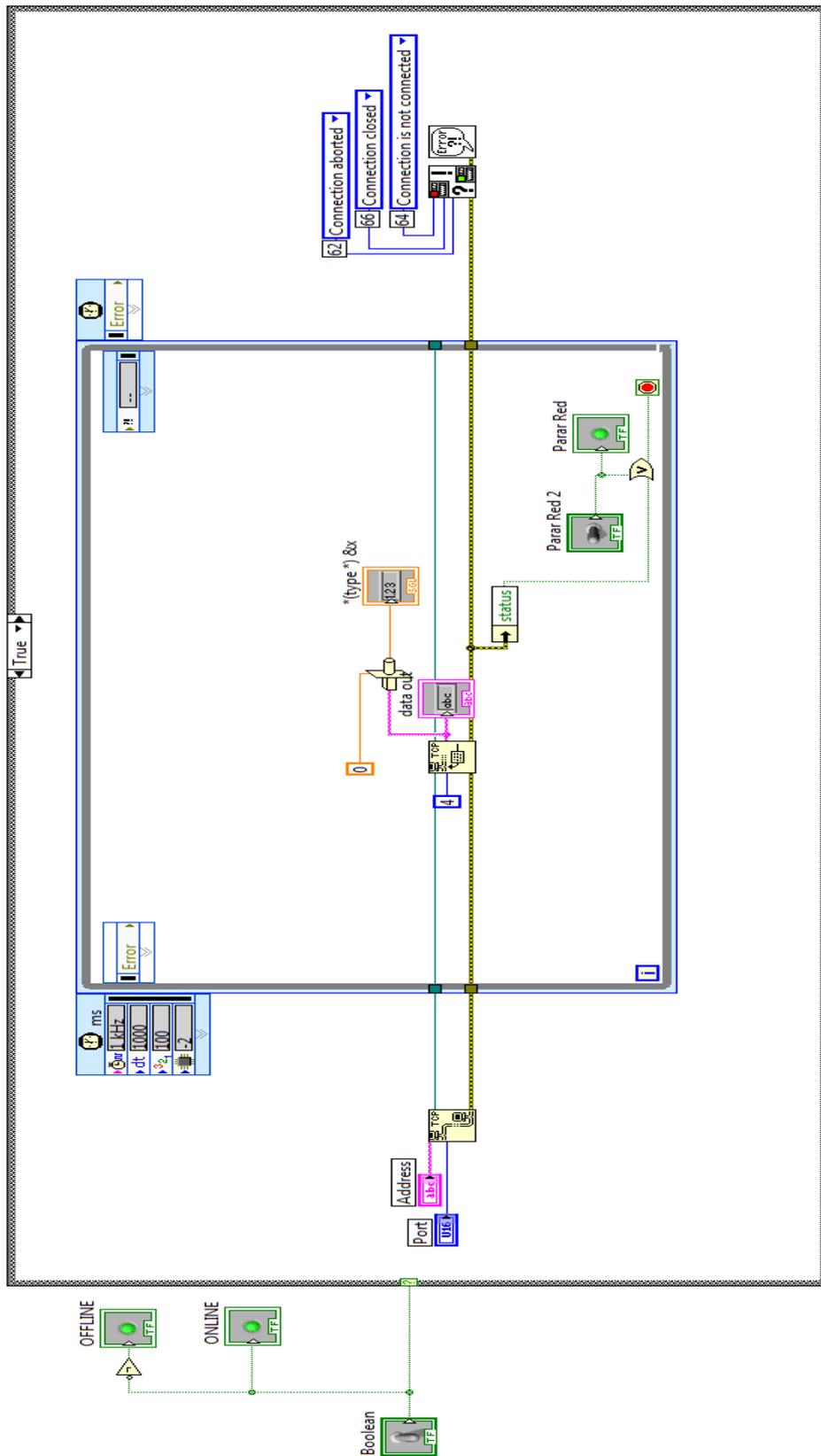


Ilustración 15 - Código de Control por Red

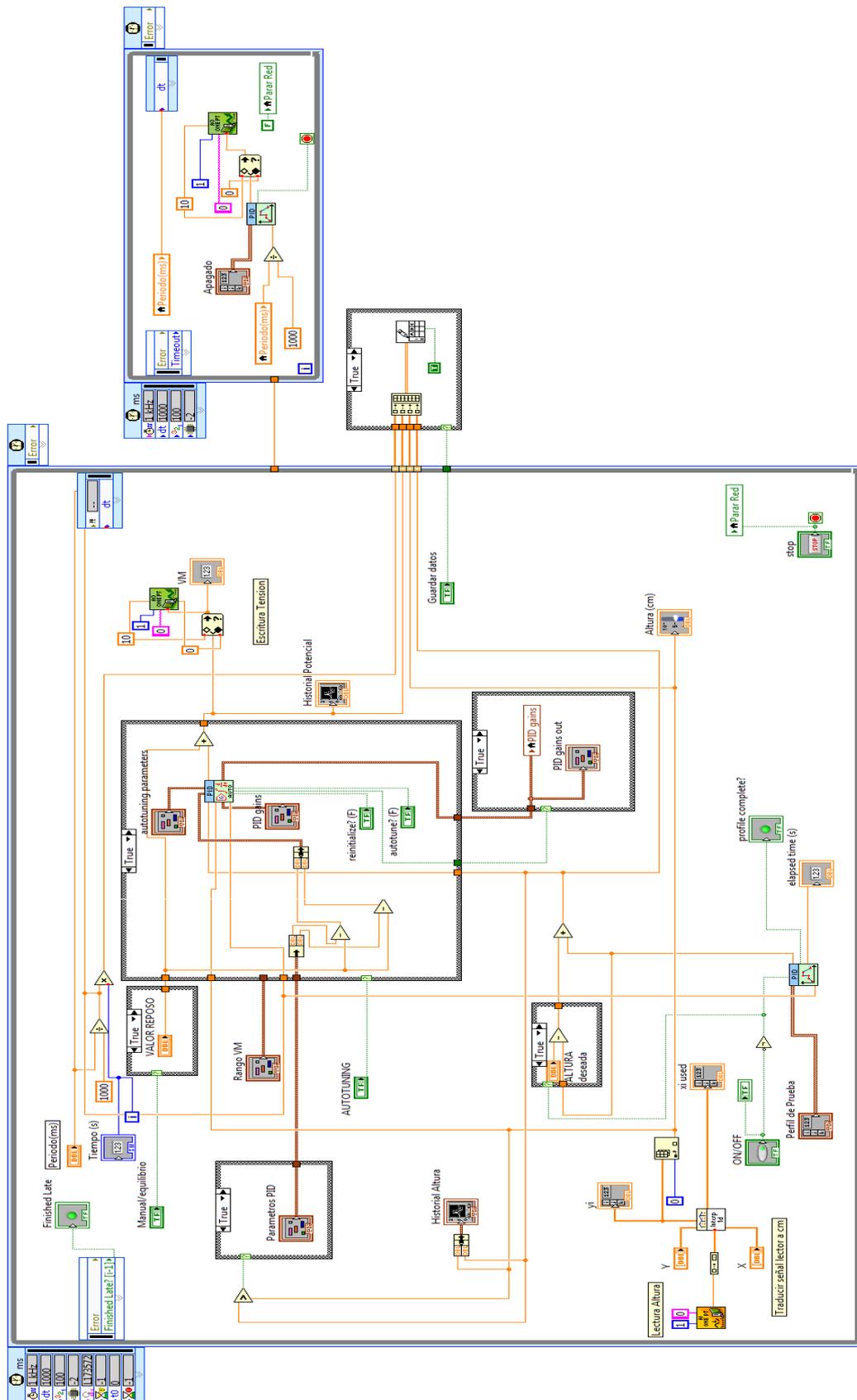


Ilustración 16 - Código Principal (1)

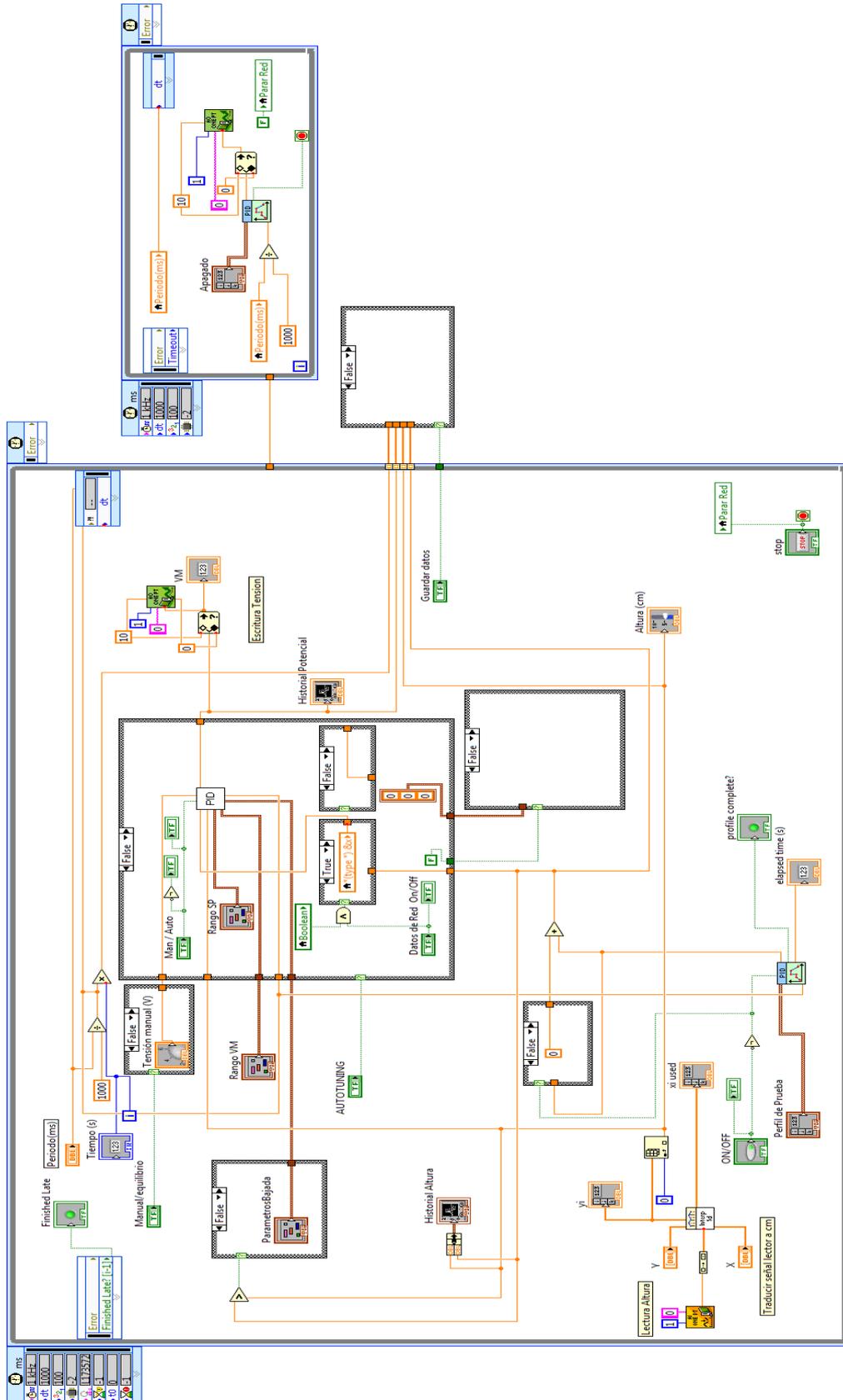


Ilustración 17 - Código Principal (2)



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°3 - ANEXO DE DISEÑO**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

ANEXO DE DISEÑO

1.	LINEALIZACIÓN DE LOS DATOS DE LA ENTRADA.....	1
2.	OBTENCIÓN DE DATOS E IDENTIFICACIÓN DEL MODELO.....	1
2.1	Obtención de Datos.....	1
2.2	Identificación del Modelo.....	5
3.	OBTENCIÓN DE LOS PARÁMETROS DEL CONTROLADOR.....	9
4.	COMPROBACIÓN DEL PROPORCIONAL.....	11

En este documento se realizará la parte de diseño perteneciente al proyecto. Primero se obtendrá un modelo del sistema, para después obtener el controlador que corresponde a dicho modelo. Para ello se utilizará el programa MatLab, en concreto la herramienta Ident y la herramienta SISOTool.

1. LINEALIZACIÓN DE LOS DATOS DEL SENSOR.

Tras instalar las herramientas de lectura, se procede a realizar un ensayo y tras observar los resultados se descubre que los números que adquiere la variable que se ha asignado para la salida no son correspondientes a los que se estimaban. Los datos oscilarán de -2.5 a 9.3, el sensor devuelve un valor mayor conforme la plataforma se aproxima a él. Por lo tanto se asigna al valor -2.35 la altura más baja, que no será cero si no -0.7 ya que la regla que mide la altura no cubre todo el rango del tubo y a 9.3 se le asignará 62.5cm.

No obstante, tras la realización de diversos ensayos se percibe que el sensor fotoeléctrico no funciona de manera lineal, esto hace que sea necesario tomar diversos datos entre el límite inferior y superior.

Se ha procedido a realizar un ensayo para la toma de datos y se ha situado la plataforma a distintas alturas y se ha anotado los diferentes valores obtenidos. Se decide que con un total de 10 valores la herramienta aproxima con un error despreciable. Gracias a estos datos y a la herramienta VI de LabVIEW Interpolate 1D, al introducir un valor del -2.35 al 9.3 nos dará su correspondiente en cm.

Esta herramienta será necesaria para poder manipular correctamente la lectura del sensor, además facilita la lectura y el estudio del sistema gracias a la traducción a centímetros. En el anejo de programación se adjuntan imágenes y se realiza una explicación de ellas.

2. OBTENCIÓN DE DATOS E IDENTIFICACIÓN DEL MODELO

2.1 Obtención de Datos.

Para poder llevar a cabo la identificación del modelo, es necesario obtener previamente los datos del ensayo. Para ello emplearemos la herramienta de LabVIEW Write To Spreadsheet File que se encuentra en la sección Programming > File I/O dentro de la librería de Functions. Ésta generará un archivo de texto con los datos a la que la

conectemos. Se decide conectarle el tiempo, la variable manipulada y la variable controlada, además también obtendrá la altura deseada (VC deseada) pese a no ser usada en el transcurso del diseño. Véase figura (Ilustración 1 - Función Guardar Datos)

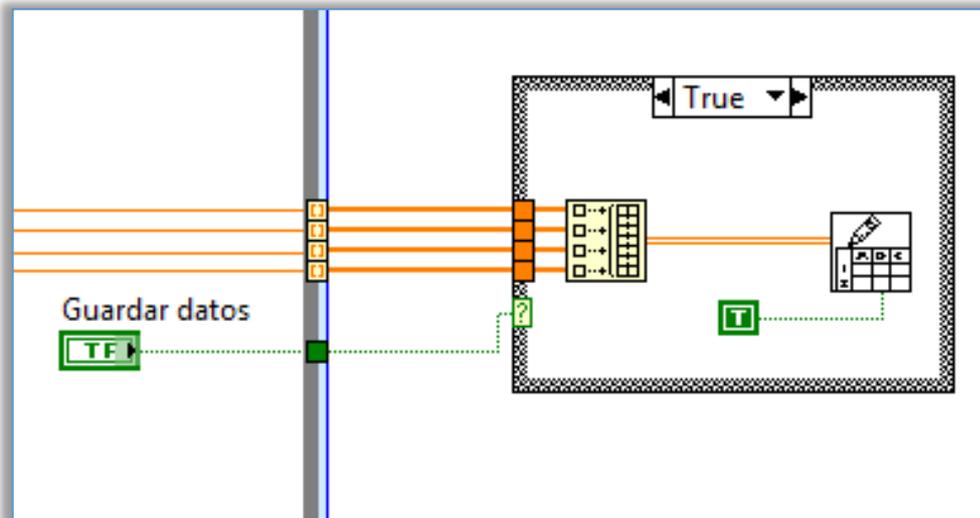


Ilustración 1 - Función Guardar Datos

NOTA: se ha añadido la opción de poder deshabilitar el guardado de los datos (mediante un Case Estructure), debido a que en futuros ensayos no será menester producir el archivo de texto, en la interfaz vendrá dado tal que:



Ilustración 2 – Habilitar Guardar Datos

Si se encuentra marcada la casilla, el programa producirá el txt tras finalizar el proceso. Es necesario para un mejor análisis introducir una entrada escalón y producir una ascensión completa, es decir, mover la plataforma desde la posición más baja 0cm (-0.7cm realmente), hasta los 60cm (62.5cm) sin variar el valor de la tensión hasta alcanzar el tope y viceversa, estando en la posición máxima introducir un escalón negativo para que descienda hasta la posición inferior de nuevo, se realiza esta secuencia tres veces para poder tener diferentes parámetros entre los que escoger. Esta es la gráfica del ensayo con el que se ha obtenido el modelo (Ilustración 3 – Gráfica de altura y tensión de los datos del ensayo):

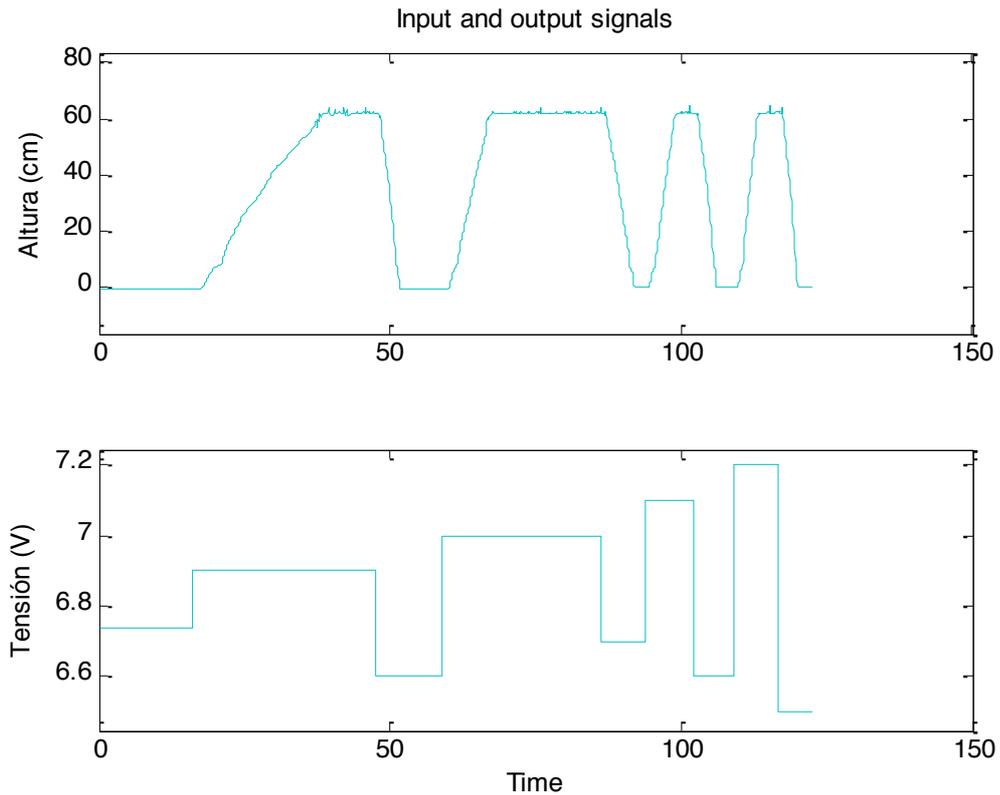


Ilustración 3 – Gráfica de altura y tensión de los datos del ensayo

Una vez obtenido el archivo de texto con todos los datos, se procede a la identificación del modelo, para ello emplearemos la herramienta de MatLab, Ident (Ilustración 4 – Introducir los datos al Ident).

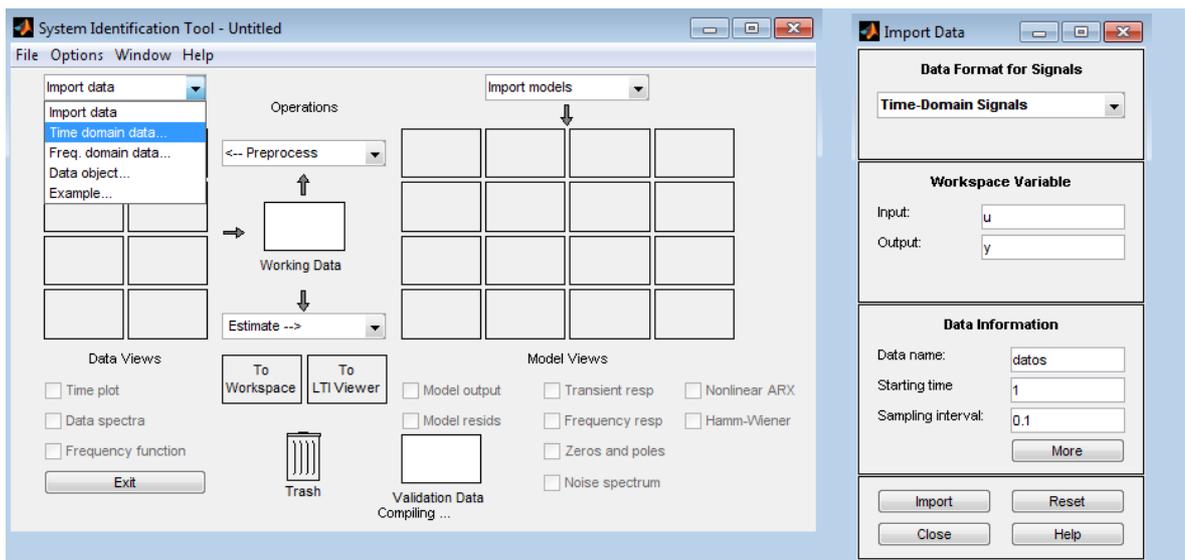


Ilustración 4 – Introducir los datos al Ident

Mediante el selector de rango aislamos los distintos escalones en la VM y sus correspondientes reacciones en la VC. Es necesario no seleccionar la rampa entera, dejar fuera de rango el sector en el que la plataforma encuentra un límite (superior o inferior) tal como se muestra en la figura (Ilustración 5 – Seleccionar el rango de los escalones)

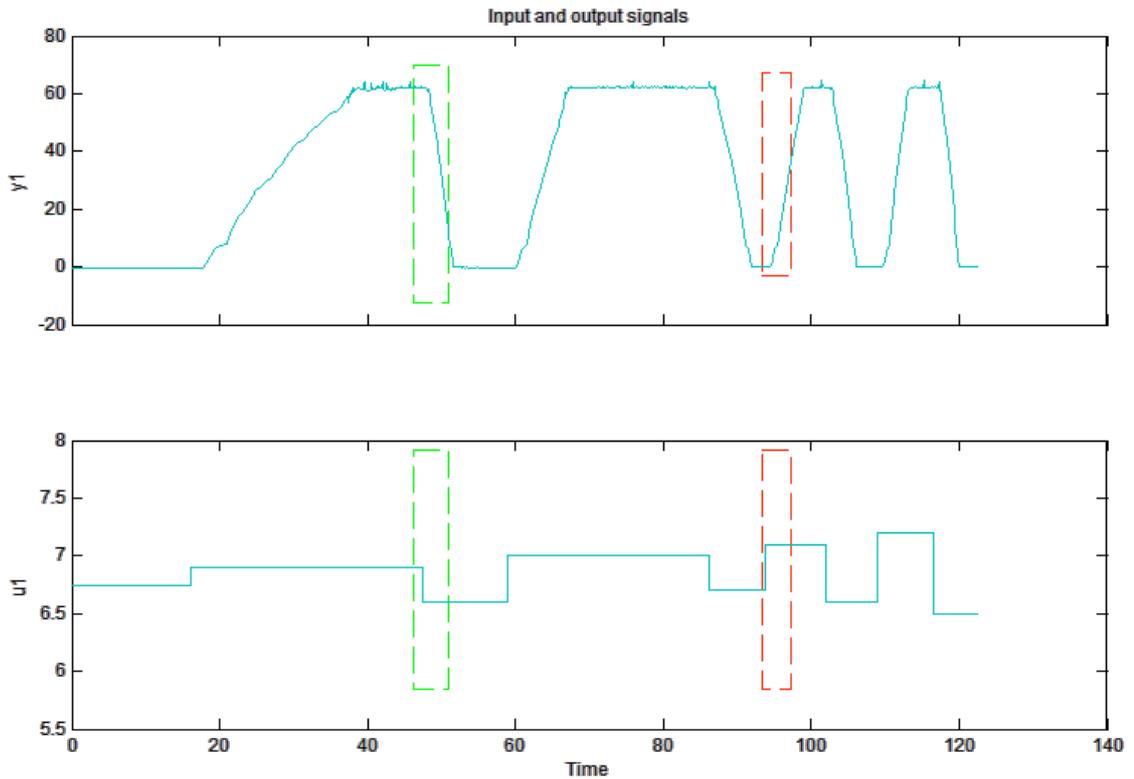


Ilustración 5 – Seleccionar el rango de los escalones

2.2 Identificación del Modelo

Tras seleccionar los diferentes escalones que hemos obtenido de nuestro ensayo, se procede a restarles el valor inicial para que la referencia de este escalón no sea relativa a un valor, sino que lo sea a cero, que el valor final sea igual a la magnitud del escalón. Para ello, el conjunto de datos que forman los escalones han de ser devueltos al Workspace, arrastrando nuestro conjunto del Data Views al recuadro que aparece en la parte central izquierda de la ventana de la herramienta Ident. Una vez en Workspace se procede con los comandos:

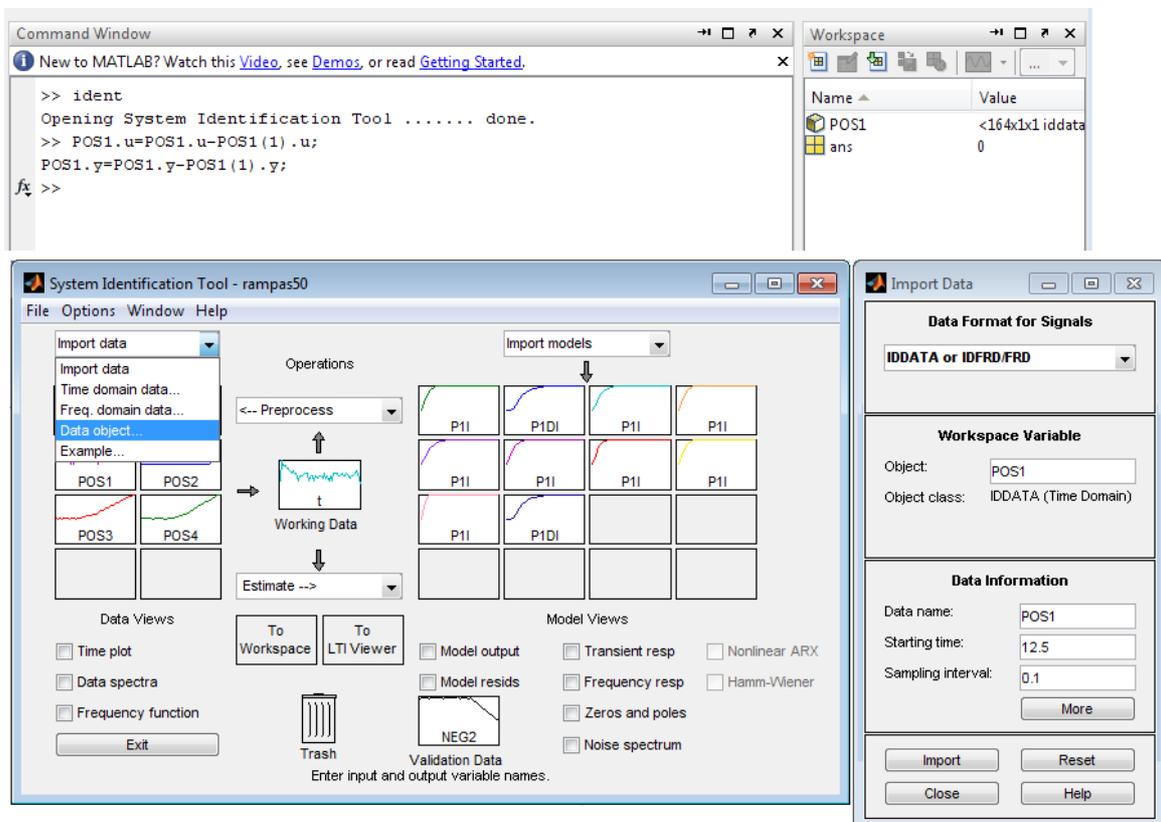


Ilustración 6 – Modificación de los datos escalón para el proceso de éstos

Mediante el creador de modelos, seleccionamos un modelo de primer orden con integrador. No obstante al resultar poco aproximados y con un coeficiente de coincidencia relativamente bajo, se opta por añadir un ‘delay’ o retraso al modelo diseñado. Ver (**¡Error! No se encuentra el origen de la referencia.**) Al ser un sistema tan sensible a las perturbaciones (las corrientes generadas dentro del tubo son casi completamente aleatorias e imposibles de prever), provocando que muchos de los modelos obtenidos no sean válidos, por eso se han realizado varias secuencias de escalones previamente.

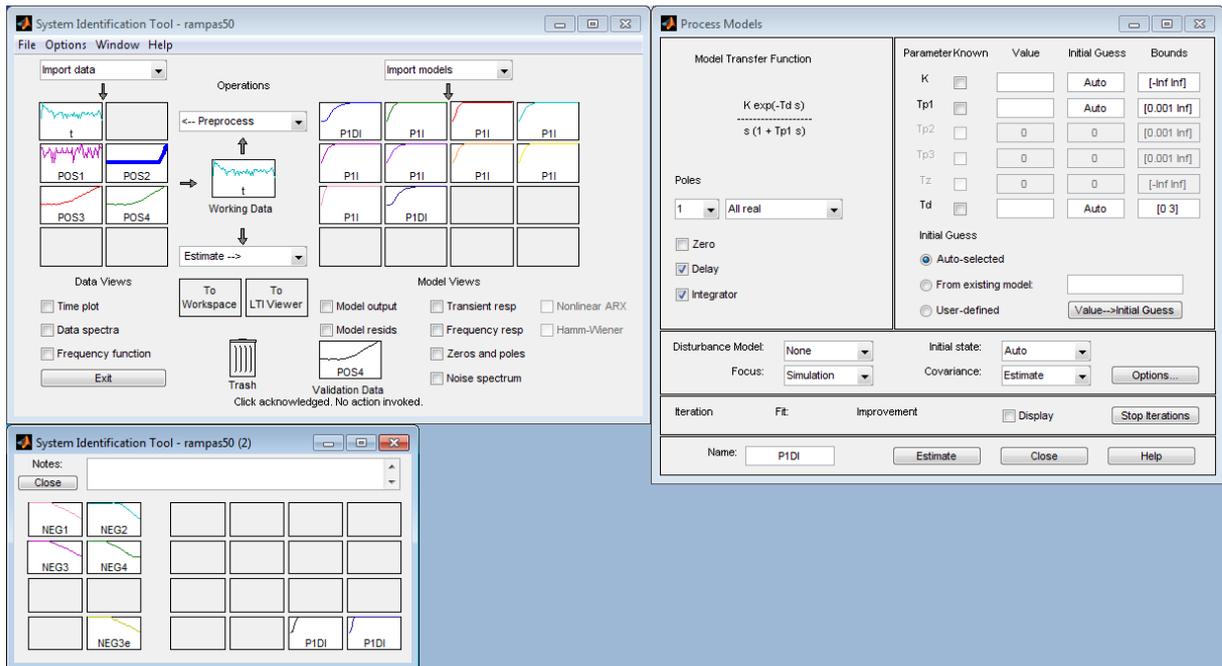


Ilustración 7 – Definición del modelo

Podemos observar la creación de numerosos modelos pero tras el contraste de la aproximación de éstos (Ilustración 8 – Gráficas resultantes de los escalones positivos) nos damos cuenta de que unos se adaptan mucho mejor que otros La herramienta Ident nos dará los parámetros de nuestro modelo, es decir, los parámetros que tendría nuestro sistema si fuese un modelo ideal, al haber seleccionado un modelo de orden 1 con integrador se obtienen los coeficientes Kp y Ti correspondientes.

En las siguiente figura (Ilustración 8 – Gráficas resultantes de los escalones positivos) se muestra el análisis de los modelos obtenidos por todos los escalones contrastados con los escalones positivos aislados de los datos del ensayo. Para un mejor análisis de las gráficas, consultar modelo a color.

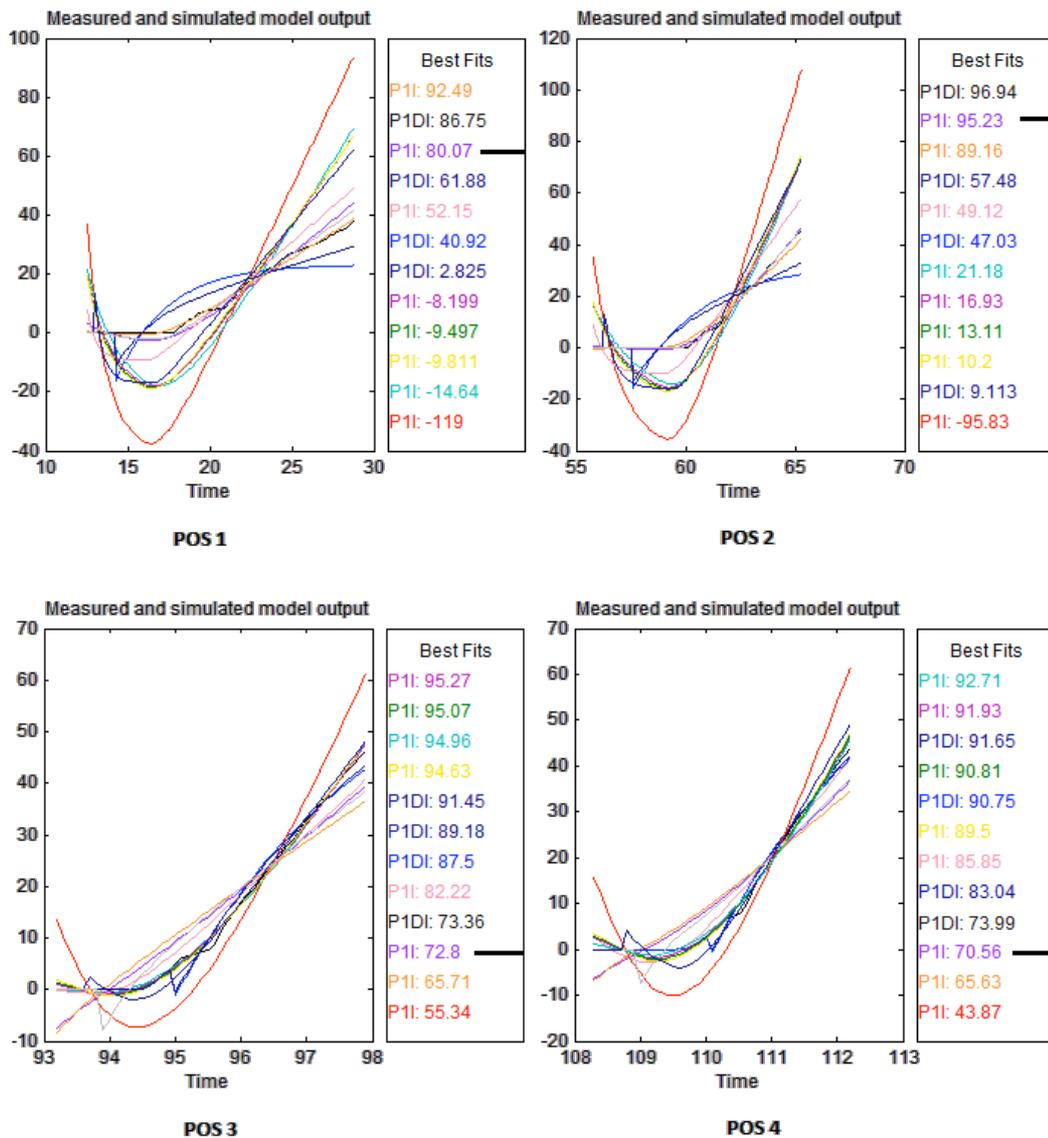


Ilustración 8 – Gráficas resultantes de los escalones positivos

Los pequeños picos y variaciones anómalas son debidos a errores del programa que toma valores iniciales distintos para cada modelo.

El resultado corresponde con el modelo con parámetros sin retraso (o ‘delay’, es el que mejor se adapta a los cuatro casos y tiene los siguientes parámetros:

$$G(s) = \frac{K_p}{s(1 + T_{p1} * s)} : K_p = 49.353; T_{p1} = 1.6104;$$

En la siguiente figura (Ilustración 9 – Gráficas resultantes de los escalones negativos) se muestra el análisis de los modelos obtenidos por todos los escalones contrastados ahora con los escalones negativos aislados de los datos del ensayo:

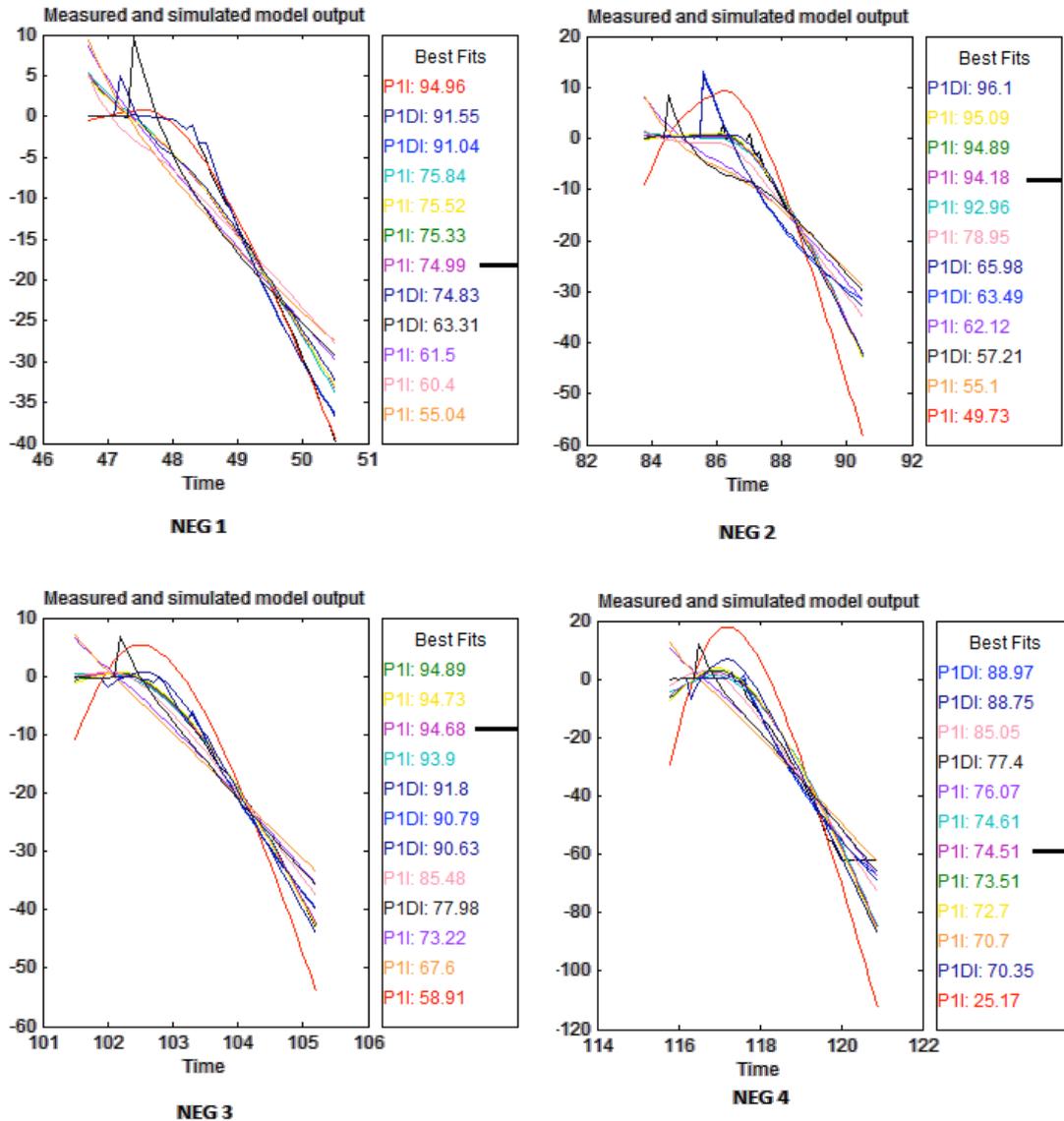


Ilustración 9 – Gráficas resultantes de los escalones negativos

En la imagen se ha marcado el modelo del apartado anterior, se puede observar que se ajusta a los escalones negativos también con una precisión bastante buena en comparación con el resto de modelos. Siendo solo superado por el modelo de color verde oscuro. De coeficientes: ($K_p=48.807$ y $T_{p1}= 1.4926$) que son de rango parecido a nuestro modelo seleccionado. Por lo tanto, se confirma que el modelo se adapta a nuestro sistema.

Se puede apreciar distintos modelos en las imágenes y es que se están comparando los modelos obtenidos de todos los escalones y además se ha añadido algún modelo con retraso, que son los que en la imagen se marcan como P1DI, para ello también se investigarán los dos mejores de ellos. Los parámetros de estos modelos son los siguientes: $K_p=24.24$ y $T_{p1}=0.86783$ y $T_d=0.71034$ y el otro con $K_p=41.603$ y $T_{p1}=0.48721$ y $T_d=0.49023$.

Para poder utilizar la herramienta SISOTool, es necesario introducir el modelo siguiendo la siguiente Transformada de Fourier:

$$G(s) = \frac{K_p}{s * (1 + T_{p1} * s)} * e^{-T_d * s}$$

Donde K_p será la variable proporcional del modelo, T_{p1} será el parámetro integrador del modelo y R_d será el 'delay' que tenga (o en el caso del primer modelo, no). Como los modelos con retraso son más difíciles de introducir en la herramienta SISOTool, se realizará una aproximación a:

$$G(s) = \frac{K_p}{s * (1 + T_{p1} * s) * (1 + T_d * s)}$$

Ahora que ya se dispone de los modelos del proceso, se procede a obtener los respectivos controladores

3. OBTENCIÓN DE LOS PARÁMETROS DEL CONTROLADOR.

Tras obtener los parámetros del modelo, procedemos a su introducción en otra herramienta de MatLab que nos permitirá obtener los coeficientes de nuestro controlador, esta herramienta es la SISOTool, que mediante el lugar de las raíces nos permitirá diseñar un controlador que se adapte a nuestras necesidades. En la siguiente figura se puede observar los comandos necesarios para la introducción de los modelos y la activación de la herramienta SISOTool (**¡Error! No se encuentra el origen de la referencia.**):

```
>> G=tf([49.353],[1.6104 1 0])      >> G=tf([41.603],[0.238845 0.97744 1 0])

Transfer function:                  Transfer function:
   49.35                            41.6
-----                            -----
 1.61 s^2 + s                       0.2388 s^3 + 0.9774 s^2 + s

>> rltool(G)
```

Ilustración 10 – Comando de SISOTool

Aparece la ventana de la herramienta SISO y en ella se puede apreciar el lugar de las raíces del modelo. Para empezar se decide hacer un proporcional solamente, así no será necesario añadir nada al sistema de momento. Si se desplazan los polos hasta prácticamente 0.621 se puede obtener el máximo valor del proporcional, 0.0031451 que el modelo nos deja utilizar, como se puede apreciar en la figura (Ilustración 11 – SISOTool con proporcional solamente).

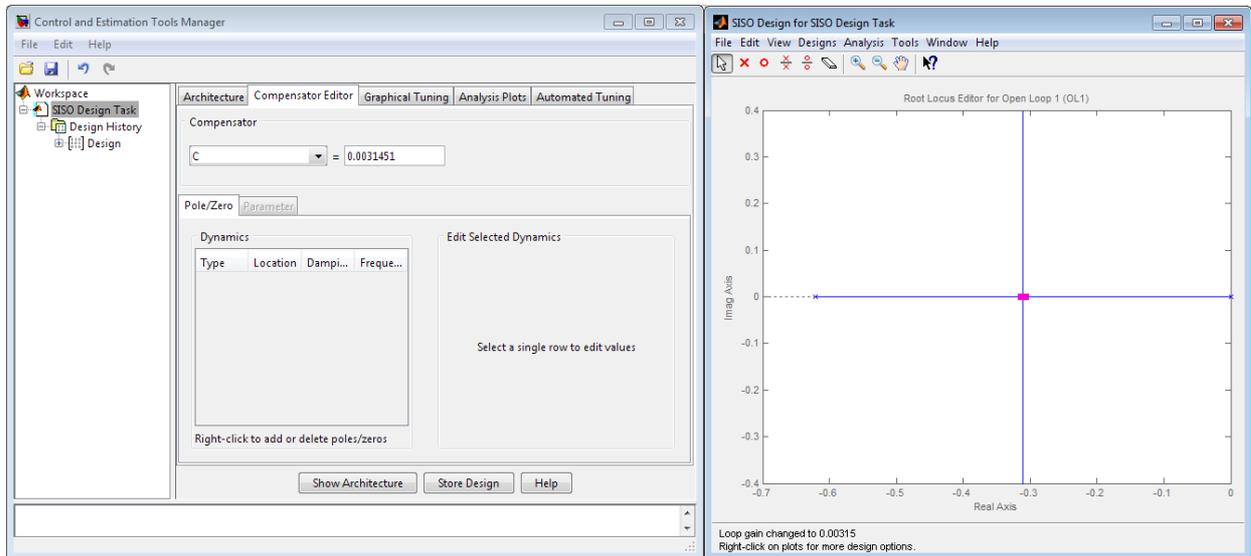


Ilustración 11 – SISOTool con proporcional solamente

Al realizar una comprobación simplemente con la acción proporcional se puede apreciar que el sistema es excesivamente lento (información predecible debido a la magnitud del parámetro proporcional). Se procede entonces a la adición de una acción derivada para mejorar el controlador. Este nuevo cero introducido se situará coincidente con el polo que dispone el modelo anulándolo y dejando el nuevo sistema tal como se puede apreciar en la figura (Ilustración 12 – SISO tras añadirle acción derivativa)

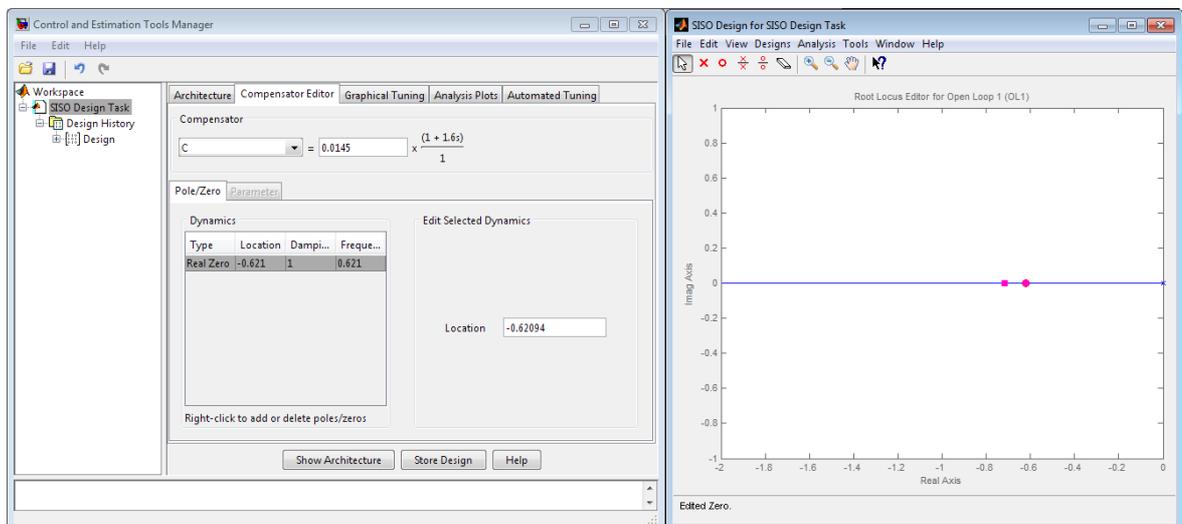


Ilustración 12 – SISO tras añadirle acción derivativa

Observamos pues, que el nuevo sistema dispondrá de acción proporcional y derivativa (PD) cuyos valores serán los siguientes: $T_d=1.61$ y $K=0.0145$, sin embargo la K podrá ser variada hasta incluso 0.2. El valor de K no podrá ser excesivamente elevado para no producir sobreoscilación o inestabilidad

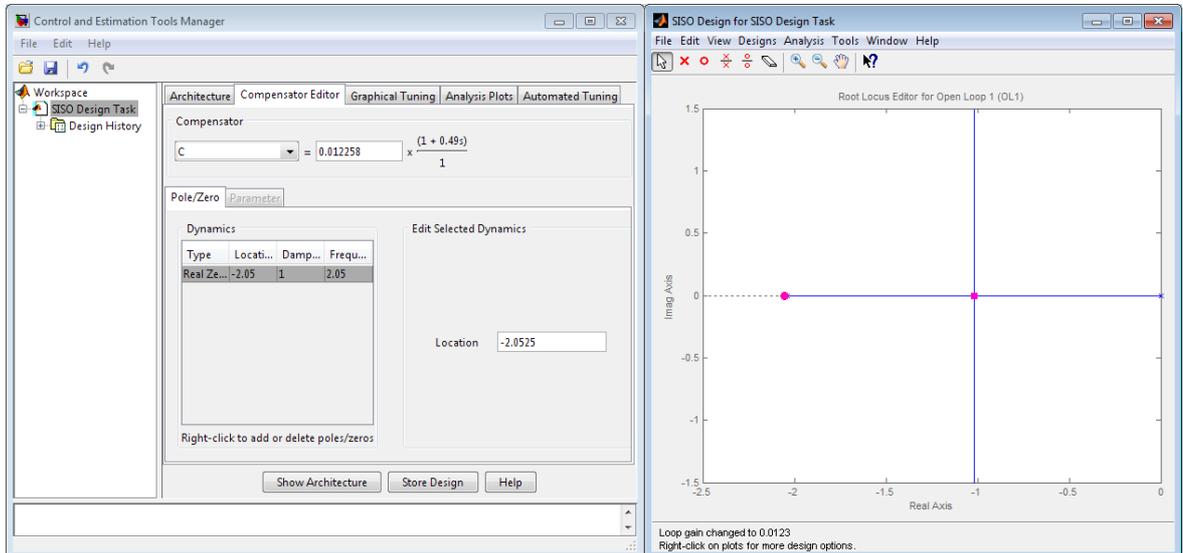


Ilustración 13 – Diseño de un segundo proporcional.

Este segundo proporcional ha sido implementado directamente con un derivador debido a los resultados del proporcional anterior. Dando como resultados una $K_c=0.012334$ y un $T_d=0.49$.

Tras una exhaustiva comparación entre ambos proporcionales se decide implantar los coeficientes de nuestro primer proporcional.

Estos datos serán utilizados en el programa de LabVIEW para la creación del controlador PD que se encargará de regular la tensión de salida en función de la función de entrada.

4. COMPROBACIÓN DEL PROPORCIONAL

Al no disponer de tiempo de establecimiento determinado, se considera un tiempo de establecimiento no superior a 6 segundos ya que se dispone de un sistema lento, se procede entonces a realizar una comprobación a través de las SISOTool y la opción de introducir un escalón, se obtiene del modelo sin retraso y del segundo controlador (añadiendo delay):

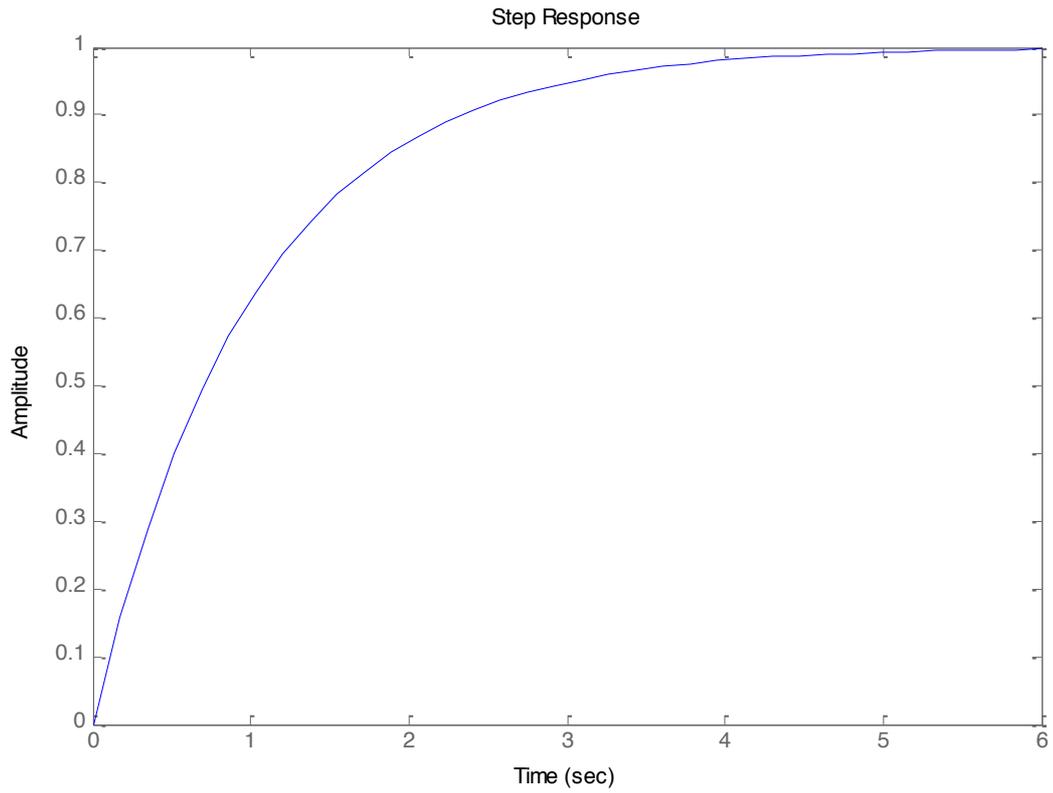


Ilustración 15 – Respuesta ante Escalón del Primer Controlador

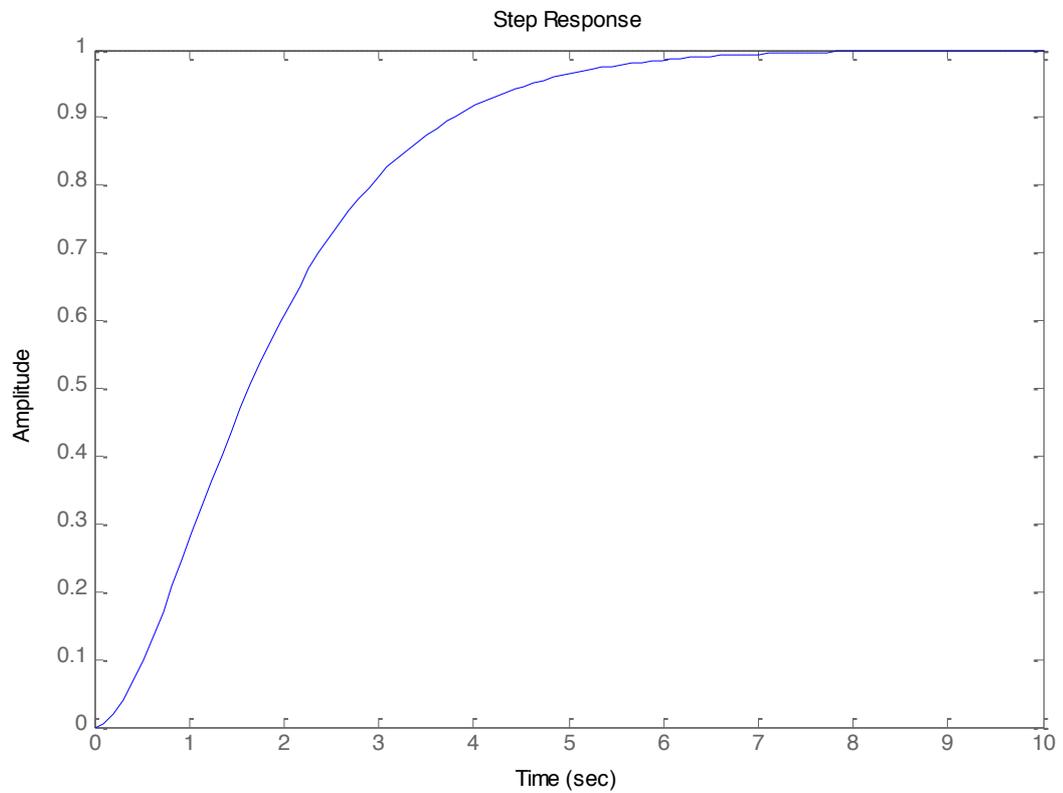


Ilustración 14 – Respuesta ante Escalón del Segundo Controlador

Al contrastar una gráfica con otra se puede apreciar que la primera es capaz de responder más rápido ante una entrada escalón, por lo tanto, se escogerán estos parámetros para ambos controladores. Pudiendo cambiar estos parámetros si el usuario lo desea en la primera página de la interfaz (consultar Manual del Usuario).



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°4 - MANUAL DEL USUARIO**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

MANUAL DEL USUARIO

- 1. REQUISITOS MÍNIMOS DEL PC 1
- 2. INTERFAZ..... 1
 - 2.1 Ventana Display 2
 - 2.2 Ventana Data..... 5
 - 2.3 Ventana Perfil. 6
 - 2.4 Ventana Autotuning 7

En este documento se explicará la interfaz de la aplicación, los diferentes modos de uso de los que dispone y como usar cada uno de ellos. Se llevará a cabo una pequeña introducción al programa, lo que incluirá la descripción de las ventanas, la navegación entre ellas, las herramientas de la interfaz. Tras esta introducción se dará paso a la explicación de la aplicación en sí y de sus distintas formas de uso.

1. REQUISITOS MÍNIMOS DEL PC

Para poder ejecutar el correctamente la aplicación, la computadora debe cumplir los siguientes requisitos:

- Microprocesador Pentium III / Celeron 866 MHz como mínimo y un Pentium 4/M en caso de querer utilizar el programa LabVIEW entero.
- Memoria RAM mínima de 256 Mb para la ejecución del programa diseñado y 1GB en caso de querer utilizar el programa LabVIEW.
- Resolución de Pantalla superior a 1024x768 píxeles.
- Conexión a Internet para poder realizar la descarga del programa.
- Disco Duro, debe tener como mínimo 407MB para la ejecución del programa realizado, además se recomienda tener más de 3.5 GB en caso de desear instalar LabVIEW completo.
- Tarjeta de Adquisición de datos, se recomienda la utilización de una tarjeta de del fabricante ADLink modelo PCI-9112 compatible
- Periféricos, Teclado y Ratón.

Las conexiones necesarias entre el ordenador se realizarán mediante un regletero (las conexiones entre este y el resto de elementos vendrán explicados en la Memoria).

2. INTERFAZ

La interfaz del programa está compuesta por una serie de ventanas: Display, Data, Perfil y Autotuning, a las que se puede acceder a través de las pestañas, como se explica más adelante. En los siguientes apartados se explicará las funciones de cada una de ellas.

2.1 Ventana Display

A continuación se muestra una captura de la primera página de la carpeta creada en el Front Panel, se procede con la explicación de esta:

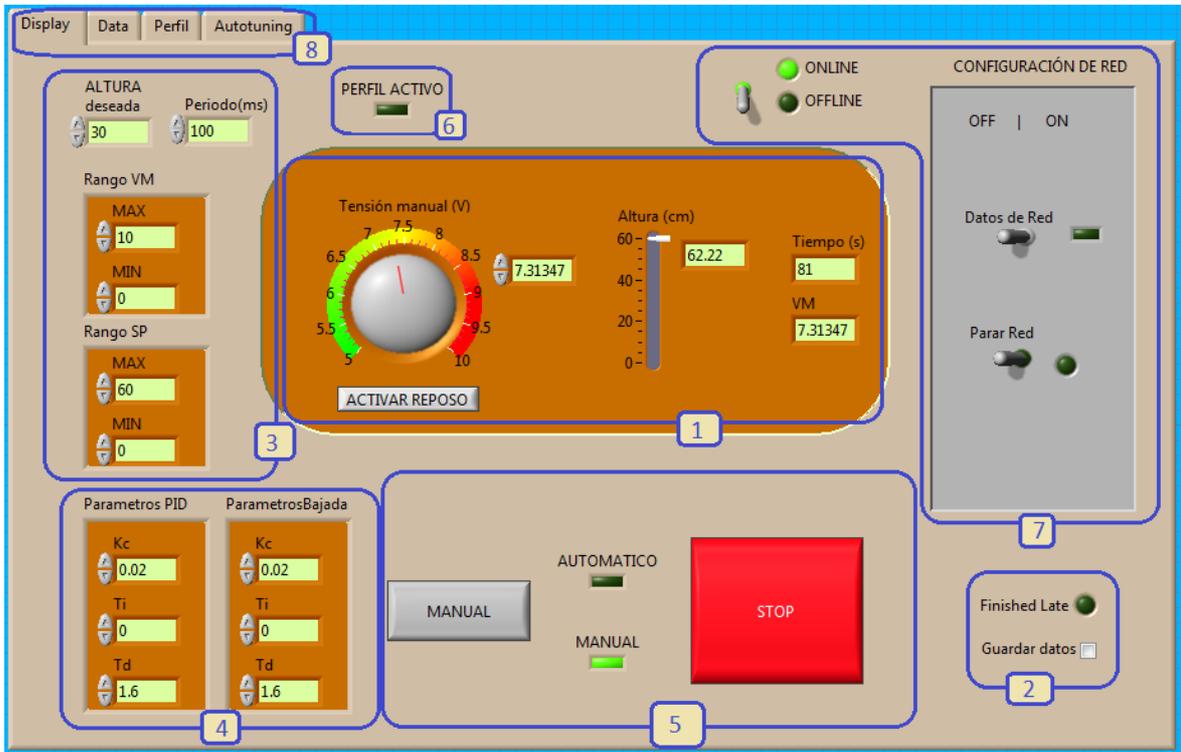


Ilustración 1 - Primera Página

- 1) **Panel de control:** Aquí se sitúa el control manual de nuestra aplicación. Mediante el control de tipo dial se regula la tensión suministrada, también es posible introducir el valor en el control numérico de su lado. Debajo del dial se encuentra un botón 'Activar Reposo' que sirve para anular el funcionamiento del dial y suministrar la tensión de reposo (que será configurable en la segunda página) En la parte central derecha se encuentra una barra que mostrará la altura a la que se encuentra la plataforma gracias a la aguja que señalará la altura, se ha facilitado también un panel digital que muestra la altura mediante cifras, para una mejor apreciación de esta. Por último, en la parte más alejada de la derecha se encuentra un panel con el tiempo transcurrido desde el inicio del ensayo y otro panel con el valor de la Variable Manipulada (en otras palabras la tensión).

- 2) **Opción de guardado de datos y led de alarma:** Como su nombre bien indica, la casilla de Guardar datos, nos permitirá guardar los datos que realicemos durante un ensayo, dejando a elección del usuario si guardar los datos o no. Si la casilla está marcada, al final del ensayo saltará una ventana con la opción de elegir la dirección y el nombre donde se desean guardar los datos. La señal luminosa de Finished Late se encenderá si durante el ensayo la estructura temporal (Time Loop) no completa el ensayo en el tiempo especificado.
- 3) **Configuración del PID y Referencia:** En este espacio se encontrarán los dos rangos que se introducen en el PID (comentado en el apartado de programación), se recomienda no modificar estos parámetros al usuario ya que podrían dañar el prototipo o producir el malfuncionamiento de la aplicación. En la parte superior se encuentran dos paneles que servirán para introducir la Referencia (ALTURA deseada) y en el otro el tiempo de muestreo por ciclo (o Periodo) en milisegundos.
- 4) **Parámetros del PID:** se ha decidido separar estos dos clusters del resto de parámetros del PID, porque a diferencia de los anteriores, estos parámetros no dañaran la estructura al menos de forma directa. Los datos que vienen preconfigurados son los datos del controlador PID diseñado. No obstante, se pueden variar si el usuario lo considera necesario, pero un cambio de parámetros puede afectar al funcionamiento de la aplicación provocando sobreoscilamientos, inestabilidad... pudiendo provocar que la plataforma golpee los topes inferiores y superiores.
- 5) **Interruptores:** Aquí se encuentran los interruptores principales de la aplicación. El botón de la izquierda es el encargado de cambiar el modo de funcionamiento de la aplicación de Manual a Automático. Se han añadido dos Leds para ayudar a diferenciar al usuario el modo de funcionamiento en el que se encuentra. A la derecha se encuentra el botón de Paro. Debido al código programado, el botón de paro no realizará un paro inmediato, si no que activará una secuencia en la cual el sistema se situará en una tensión inferior a la de Reposo durante 7 segundos, dándole tiempo a la plataforma de descender lentamente para después reducir la tensión a cero y detener el programa.
- 6) **Perfil activo:** Se introduce una señal luminosa de que el 'Perfil de Prueba' se está llevando a cabo. Este perfil se puede reconfigurar por el usuario y está disponible en la tercera página)
- 7) **Configuración en Red:** Siendo esta la parte más compleja de la interfaz, se llevará a cabo una explicación en profundidad. Junto a esta aplicación se adjunta el servidor también:

- a. **Botón ONLINE / OFFLINE:** Sirve para alternar entre el modo de funcionamiento conectado a la red o sin conexión. Si se va a trabajar sin conexión a la red este botón tiene que estar en posición OFFLINE antes de ejecutar la aplicación. Si por el contrario se va a trabajar con conexión a Red y control mediante otro computador, el interruptor deberá estar en posición ONLINE antes de su encendido.

 - b. **Datos de Red:** Durante el funcionamiento en red se puede distinguir entre elegir los datos recibidos vía red o, por el contrario, seguir realizando el control desde la aplicación. Este interruptor podrá conmutar entre los datos enviados por el servidor o utilizar los datos suministrados por el usuario mediante esta aplicación (0- Datos a través de esta aplicación 1- Datos a través de la Red). Esta función solamente recibe la Referencia deseada desde otra máquina será necesario activar el modo Automático para que escuchar realmente los datos.

 - c. **Parar Red:** Este interruptor sirve para desactivar la comunicación con el servidor, no obstante almacena el último valor recibido y la variable permanecerá con ese valor. Para terminar realmente de utilizar los datos recibidos por red, será necesario apagar el interruptor de Datos de Red.
- 8) **Pestañas:** Para navegar entre las diferentes páginas de la carpeta basta con clicar en la pestaña a la que se desea ir.

2.2 Ventana Data

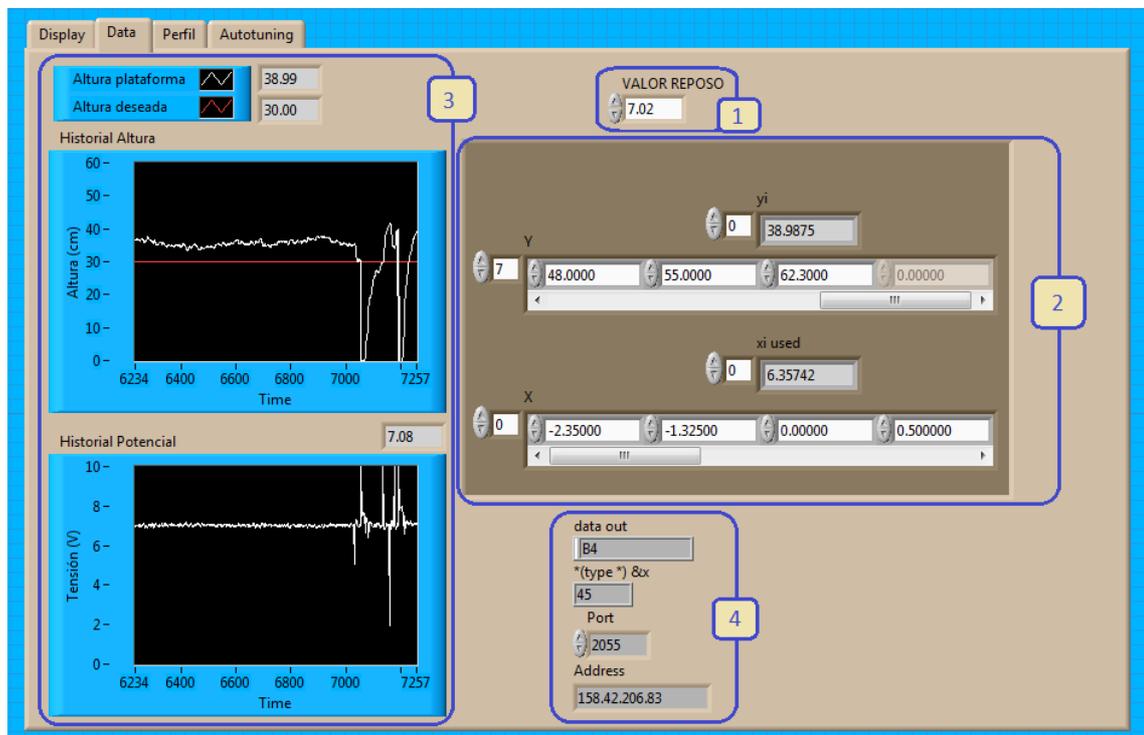


Ilustración 2 - Segunda Página

- 1) **Valor Reposo:** Este es el panel donde se podrá cambiar el valor de la tensión e introducir el valor de la tensión para el cual la plataforma no oscila u oscila lo menos posible. Antes de proceder con la aplicación el usuario deberá introducir el valor de dicha tensión ya que debido a la naturaleza de este proceso (un ventilador que propulsa la plataforma a través del aire ambiental) este valor oscilará, ya sea por temperatura, altitud del lugar... Para ello se enciende la aplicación en modo manual y se oscila levemente el valor que encontramos predeterminado hasta encontrar el nuevo valor.
- 2) **Interpolador:** Estos son los datos con los cuales hemos obtenido la traducción de la salida del sensor a centímetros, se recomienda al usuario no manipular estos datos.
- 3) **Gráficas:** En esta página se encuentran las gráficas que muestrean los datos que van alcanzando, en la gráfica de arriba, la referencia y la altura actual y en la de abajo la tensión. Se muestrean los datos en cada ciclo y en los paneles digitales aparece el valor (en cifras) de cada variable en el momento actual.

- 4) **Configuración de Red:** En estos paneles se configura la conexión por red, en 'Address' se introducirá la IP del servidor al que deseamos conectarnos, y en 'Port' el puerto al que se conecten servidor y cliente.

2.3 Ventana Perfil.

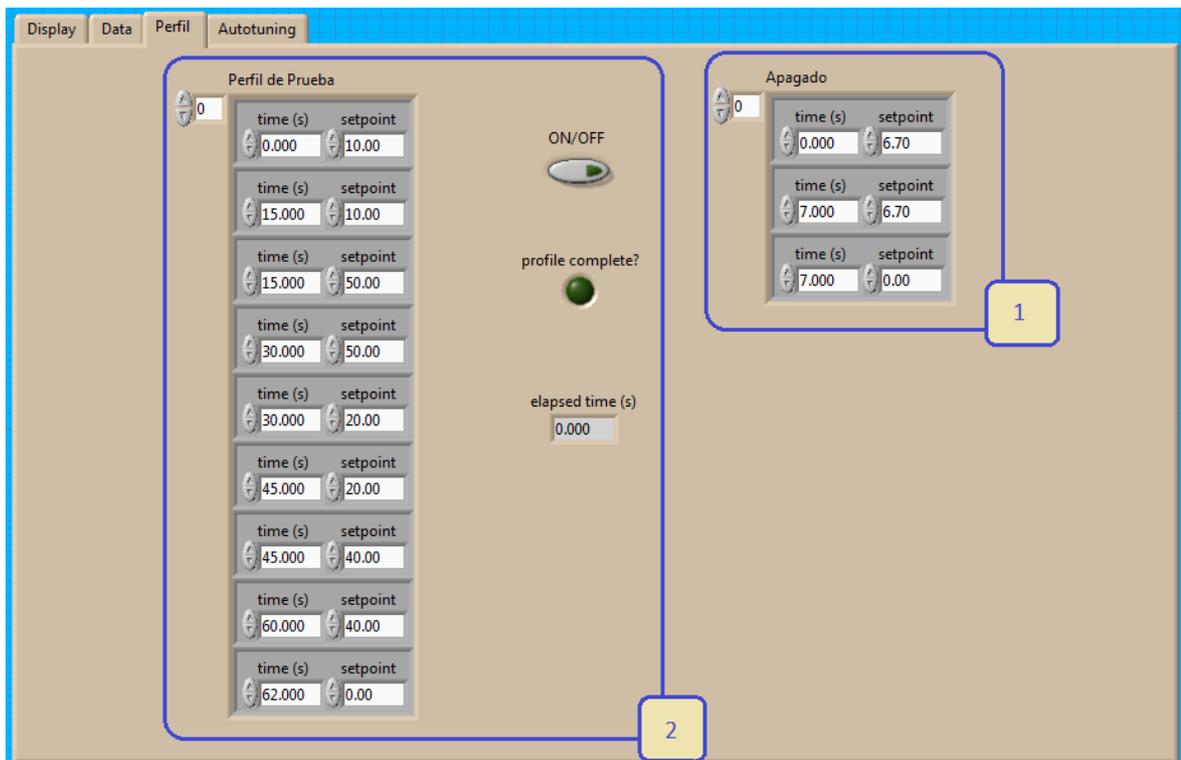


Ilustración 3 - Tercera Página

- 1) Perfil Apagado: Aquí se encuentra la configuración del perfil mencionado anteriormente. Como se puede apreciar en la imagen esta preconfigurado para administrar una tensión levemente inferior a la de Reposo durante 7 segundos. La plataforma descenderá lentamente y tras 7 segundo se apagará. En caso de que la tensión de Reposo sea inferior a la indicada, el usuario deberá cambiarla.
- 2) Perfil de pruebas: Perfil diseñado para la comprobación de nuestro PID, también se puede utilizar con otros fines, se deja a disposición del usuario el cambiarlo a su antojo. El interruptor de su derecha 'ON/OFF' es el encargado de realizar el perfil. Una vez terminado el perfil, se ha de apagar el interruptor para poder manipular la referencia de nuevo.

2.4 Ventana Autotuning

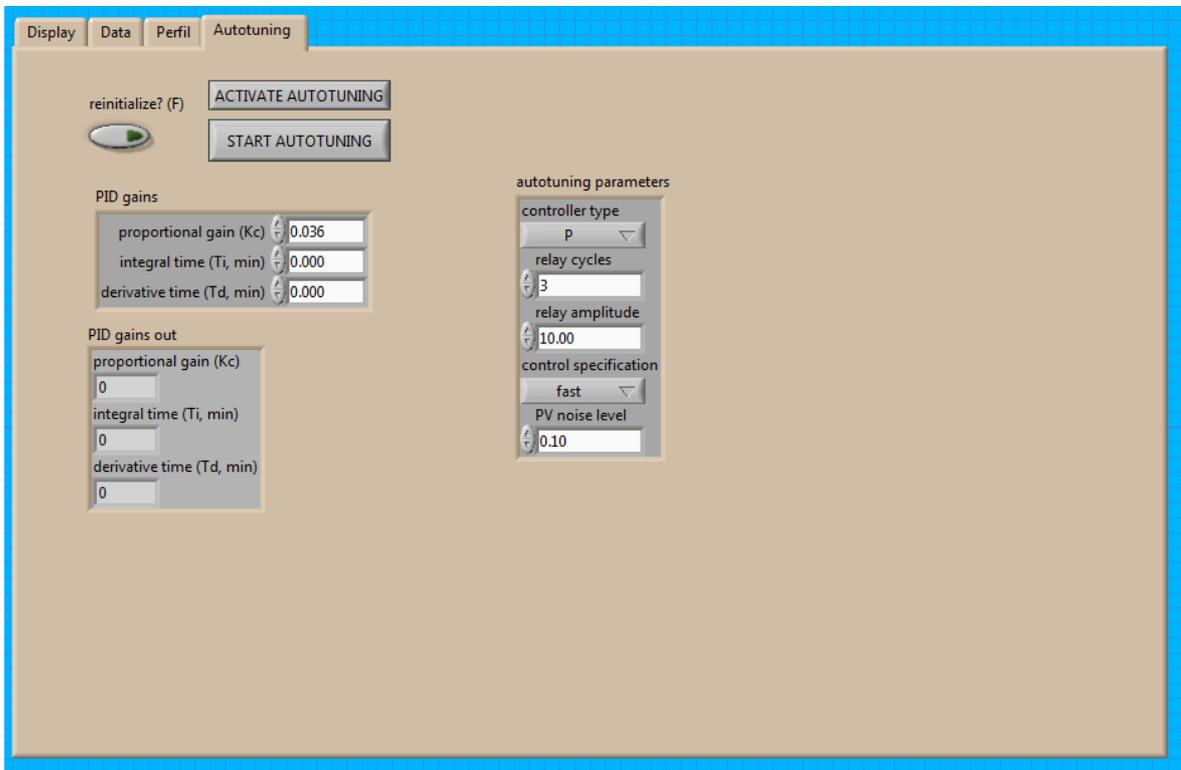


Ilustración 4 - Cuarta Página

En esta última página se muestran los diferentes controles que posee el autotuning.

Se ha incluido la función Autotuning por si el usuario decide utilizar los parámetros que le proporcione esta herramienta en lugar de los diseñados.

Para hacer funcionar la herramienta de autotuning habrá que estabilizar primero el sistema y después activar la tecla 'Activate Autotuning' para cambiar de funcionamiento a autotuning y después activar la tecla 'Start Autotuning' para dar comienzo al autotuning.

Aparecerá esta ventana (Ilustración 5 - Autotuning) y el usuario solo deberá seguir las instrucciones mostradas:

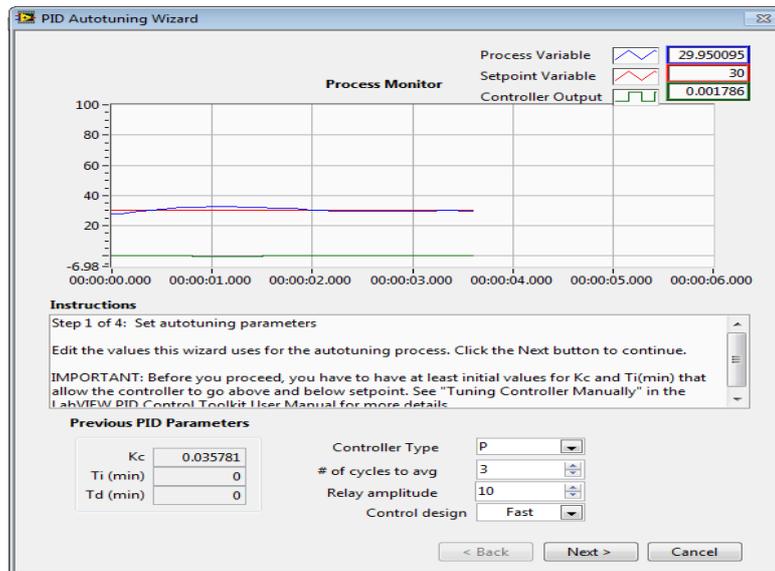


Ilustración 5 - Autotuning

Al final aparecerán los parámetros diseñados por la herramienta y será decisión del usuario usarlos o no.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°5 - PLIEGO DE
CONDICIONES**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

PLIEGO DE CONDICIONES

1.	PLIEGO DE CONDICIONES GENERALES.....	1
1.1	Pliego de Condiciones Facultativas.....	1
1.1.1	Programación de Trabajos.....	1
1.1.2	Ejecución Defectuosa y Modificaciones.....	1
1.1.3	Recepción Definitiva.....	2
1.2	Pliego de Condiciones Económicas.....	2
1.2.1	Fianzas.....	2
1.2.2	Composición de Precios.....	3
1.2.3	Mejoras.....	3
1.2.4	Revisión de Precios.....	3
2.	PLIEGO DE CONDICIONES PARTICULARES.....	4
2.1	Condiciones de Funcionamiento del Sistema.....	4
2.1.1	Limitación por Equipo Hardware.....	4
2.1.2	Limitaciones Bajo Sistemas Operativos.....	5
2.1.3	Control del Sistema.....	5
2.2	Condiciones de Instalación.....	5
2.3	Condiciones del Programa.....	5
2.4	Condiciones de Mantenimiento.....	6
2.4.1	Condiciones de Mantenimiento del Programa Informático.....	6
2.4.2	Condiciones de Mantenimiento del Soporte Magnético.....	6
2.5	Condiciones de Garantía.....	7

1. PLIEGO DE CONDICIONES GENERALES.

Se describen bajo este título una serie de normas de carácter general y obligado cumplimiento en cualquier tipo de proyecto de estas características.

Distingamos dos apartados:

- Pliego de condiciones facultativas.
- Pliego de condiciones económicas.

1.1 Pliego de Condiciones Facultativas.

Como principales obligaciones, al cliente se le exige conocer:

- Las leyes.
- El proyecto en su totalidad.

1.1.1 Programación de Trabajos.

El programador se compromete a realizar el proyecto dentro de los plazos acordados con el cliente, después de una revisión exhaustiva del mismo.

1.1.2 Ejecución Defectuosa y Modificaciones.

El cliente tiene la responsabilidad de revisar el funcionamiento del programa suministrado y comunicar los aspectos que, a juicio de él, no estén bien, extendiéndose dicha responsabilidad a aquellos defectos no detectados pero existentes.

Las modificaciones que el cliente crea oportuno realizar se verán, en cualquier caso, grabadas por su correspondiente incremento de precio, ya que se considerarán como cambios no imprescindibles para el normal funcionamiento del programa y realizados por sus gustos particulares. Atendiendo a esta última razón, el proyectista se compromete a alterar el contenido del programa siempre que no perjudique su ejecución.

Las tarifas a aplicar, en caso de modificaciones, serán las mismas que para el resto del proyecto, siguiendo las normas publicadas en el Boletín Oficial del Estado.

En el improbable caso de que las anomalías detectadas se deban a una mala o defectuosa programación, el proyectista se compromete a realizar las modificaciones que estime convenientes para subsanarlas, sin que esto origine costo económico alguno para el comprador.

1.1.3 Recepción Definitiva.

Terminado el plazo de ejecución del proyecto, se procederá a la recepción definitiva del mismo.

1.2 **Pliego de Condiciones Económicas.**

Seguidamente se describen las relaciones económicas que deben regir entre el cliente y el proyectista, para cada uno de los siguientes apartados:

- Fianzas.
- Composición de precios.
- Mejoras.
- Revisión de precios.

1.2.1 Fianzas.

Las condiciones de pago del proyecto realizado son las siguientes:

El comprador depositará, en el momento de adjudicación del proyecto y antes del inicio de los trabajos de programación una fianza como garantía, por una cantidad equivalente al 5% del valor total del presupuesto del proyecto.

En caso de demora en el abono de la fianza, esta se incrementará en un 5% semanal.

El resto del valor total del proyecto será abonado una vez realizada la instalación y comprobación del correcto funcionamiento del programa por el usuario.

Tras la concertación del período máximo de realización del proyecto entre el proyectista y el cliente, se ha acordado que el proyectista abone una cantidad del 1% del valor total del proyecto, por cada día de demora sobre la fecha acordada con el cliente.

Se ha estimado convenientemente realizar un mantenimiento conjunto del programa informático, durante un período de tres años después de la entrega del mismo. Durante este período la garantía del producto será total.

1.2.2 Composición de Precios.

Los precios aplicados al proyecto están en consonancia con los dispositivos generales acordados en los convenios en los que tiene ámbito. Así, las tarifas empleadas por la mano de obra corresponden a las vigentes para un Ingeniero Técnico Industrial realizando tareas de programación.

1.2.3 Mejoras.

Las ampliaciones y mejoras que el cliente estime oportuno realizar en el programa, supondrán un aumento en el importe total del proyecto, de acuerdo con el total que supusiese realizarlas y las tarifas ya estipuladas.

1.2.4 Revisión de Precios.

Como el tiempo que puede transcurrir en la redacción de este proyecto y su entrega y aceptación por parte del cliente, puede ser largo, si transcurre el período de tiempo tabulado por el Código Oficial de Peritos e Ingenieros Técnicos Industriales de Valencia (C.O.P.I.T.I.), se procederá a una revisión de precios en el acto de entrega del mismo. Para ello se utilizarán las fórmulas y tablas dispuestas a tal fin en el Boletín Oficial del Estado del año correspondiente.

2. PLIEGO DE CONDICIONES PARTICULARES.

Bajo este título se desarrollan los aspectos relacionados con las características más específicas de un proyecto como el presente, que por tratarse de un programa informático, reúne una serie de cualidades, que lo diferencian de la mayoría.

2.1 Condiciones de Funcionamiento del Sistema.

Incluye los mínimos exigidos para el correcto funcionamiento del programa en cuanto a hardware, y software, así como la configuración básica del equipo.

2.1.1 Limitación por Equipo Hardware.

La aplicación software desarrollada, sólo puede funcionar en sistemas PC que reúnan las condiciones hardware que se indican a continuación:

- Microprocesador Pentium III / Celeron 866 MHz como mínimo y un Pentium 4/M en caso de querer utilizar el programa LabVIEW entero.
- Memoria RAM mínima de 256 Mb para la ejecución del programa diseñado y 1GB en caso de querer utilizar el programa LabVIEW.
- Resolución de Pantalla superior a 1024x768 píxeles.
- Conexión a Internet para poder realizar la descarga del programa.
- Disco Duro, debe tener como mínimo 407MB para la ejecución del programa realizado, además se recomienda tener más de 3.5 GB en caso de desear instalar LabVIEW completo.
- Tarjeta de Adquisición de datos, se recomienda la utilización de una tarjeta de del fabricante ADLink modelo PCI-9112 compatible
- Periféricos, Teclado y Ratón.

Estos requisitos son los mínimos requeridos para el correcto funcionamiento del programa incluyendo las operaciones de simulación que realiza, por tanto los que se deben incluir en el Pliego de Condiciones como información al Cliente.

2.1.2 Limitaciones Bajo Sistemas Operativos.

Los requisitos de software son los siguientes:

- Sistema Operativo Windows 8/Windows 7/Vista/XP.

2.1.3 Control del Sistema.

Se realizará de la forma más sencilla posible, para ello se han tomado todas las medidas necesarias para que el cliente se pueda desenvolver por el programa de forma cómoda, simplemente debe saber desenvolverse bajo el entorno gráfico Windows, ya que el programa a incorporado todos los controles de los que dispone un sistema visual como es Windows.

- Botones de control de Funciones sencillas.
- Espacios diferenciados y ordenados.
- Ayuda disponible sobre fondo de pantalla.
- Y por supuesto el uso de Ratón.

2.2 Condiciones de Instalación.

El programa puede ser instalado fácilmente por el cliente, sin apenas tener unos vanos conocimientos del entorno gráfico Windows, ya que en el manual de usuario será explicado detalladamente el modo de instalación del programa.

2.3 Condiciones del Programa.

La adjudicación del proyecto da derecho, exclusivamente a una licencia del mismo. Cada licencia consta de un disquete, el cual contienen el programa en su totalidad.

Si el usuario precisa de otra copia del programa, a parte de la suministrada, por razones ajenas al funcionamiento de la misma, está terminantemente prohibida la transferencia o copia de un ordenador a otro. En este caso, se deberá proceder a la compra de otra licencia de venta por parte del usuario al técnico realizador del proyecto.

En el precio de esta/s nueva/s copia/s se valorarán los coste directos de producción del software sin tener en cuenta desplazamientos y horas de puesta en marcha, siempre y cuando la instalación se realice por parte del cliente. En dicho caso, cualquier anomalía en la instalación que produzca un funcionamiento incorrecto del programa, anula el derecho de garantía posventa de la copia.

2.4 Condiciones de Mantenimiento.

Este apartado incluye el mantenimiento del programa informático y el material de soporte magnético en el que se suministra.

2.4.1 Condiciones de Mantenimiento del Programa Informático.

El programa informático no necesita revisión por parte de ningún técnico, salvo en el caso de modificaciones en la configuración del hardware o del software. Así mismo, si se actualiza o modifica la versión del equipo y procediera modificar alguna característica o ubicación del programa, dicha revisión deberá efectuarse por parte de un técnico designado por el realizador del proyecto. En este caso regirá el apartado económico “1.2.3.- Mejoras” para tasar el costo asignado, y además devengaran derechos por desplazamiento al lugar donde esté ubicado el equipo, y si procediera, dietas, tal como establece la ley vigente para este tipo de instalaciones.

Queda terminantemente prohibida la alteración del programa y, si se comprobara que esté ha sido modificado, ello acarrearía la pérdida total de la garantía del producto así como cualquier derecho sobre actualizaciones, mejoras y/o revisiones.

2.4.2 Condiciones de Mantenimiento del Soporte Magnético.

Las condiciones de mantenimiento aquí detalladas son las habituales en este tipo de discos ópticos. Se debe mantener el disco alejado de las radiaciones solares, no exponerlo a temperaturas elevadas, no deslizar la superficie legible sobre superficies abrasivas.

2.5 Condiciones de Garantía.

Si por cualquier razón se produjera una pérdida accidental de algún fichero se podrá, previa autorización del proyectista, copiar de nuevo sólo los citados ficheros, o si fuese durante los tres años a partir de la fecha de adjudicación del proyecto dada la limitación de los discos flexibles convencionales, el programador se compromete a suministrar otra copia del paquete completo en un plazo máximo de 6 días a partir de la fecha de notificación de la pérdida por parte del cliente, a la que deberán acompañar los discos estropeados para su comprobación.

De todo ello se desprende, por tanto, que el período de vigencia de la garantía es de 3 años. Más allá, el proyectista no se hará cargo de los gastos ocasionados por posibles anomalías, por otra parte lógicas por el paso del tiempo. Si transcurrido este plazo se requiriera otra copia del programa, se deberá volver a negociar el acuerdo, aunque por condiciones del cliente, se otorgarán importantes beneficios en forma de descuentos y rapidez de entrega. Por último señalar además, que la garantía se establece automáticamente en el momento de la firma del contrato de compraventa, sin ningún otro tipo de trámite por parte del comprador del producto de software objeto del proyecto.

Conviene recordar que cualquier cambio o variación en el programa por parte del cliente, así como su copia, préstamo o instalación no autorizada, anula completamente cualquier condición de garantía, expuesta anteriormente. Esto incluye también la utilización indebida del disco y/o sus ficheros, bien sea por trabajar bajo condiciones, entornos o configuraciones no recomendadas o por negligencia en su utilización

El programador no se hace responsable, en ningún caso, de los perjuicios que se pueden desprender de la utilización indebida del programa informático, tanto materiales como daños a terceras personas.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

**DESARROLLO DE UNA APLICACIÓN PARA
LA IDENTIFICACIÓN Y CONTROL DE UNA
PLATAFORMA EN SUSPENSIÓN MEDIANTE
LABVIEW
DOCUMENTO N°6 - PRESUPUESTO**

AUTOR: VÍCTOR GARCÍA LLOBREGAT

TUTOR: JUAN MANUEL HERRERO DURÁ

Curso Académico: 2013-14

PRESUPUESTO

1.	INTRODUCCIÓN.....	1
2.	ESTUDIO ECONÓMICO.....	2
2.1	Costos de la Mano de Obra.....	2
2.1.1	Costos de Mano de Obra Directa.....	2
2.1.2	Costes de Mano de Obra Indirecta	3
2.1.3	Costo Total de la Mano de Obra.....	3
2.2	Gastos Generales.....	4
2.2.1	Gastos de Mantenimiento	4
2.2.2	Gastos por Necesidad de Material	4
2.2.3	Total Gastos Generales.....	4
2.3	Amortizaciones	5
2.3.1	Amortización del Hardware.....	5
2.3.2	Amortización del Software	5
2.3.3	Amortización Total:.....	5
3.	PRESUPUESTO FINAL.....	6

1. INTRODUCCIÓN.

El proyecto presente trata sobre el desarrollo de una aplicación por tanto, al no emplear material adicional para la creación de un producto final, nuestro proyecto se basará, en resumidas cuentas, en horas de trabajo realizadas por un ingeniero, el coste del hardware entregado, más licencias, permisos y demás tipos de impuestos.

Los honorarios del ingeniero son fácilmente estimados. Basta con obtener el valor medio del salario de un técnico especializado en programación por lo que supone una gran ventaja a la hora de realizar la factura. No obstante, también hay que estimar el valor del material empleado en el diseño del programa, lo que resulta en el presupuesto total.

Este se divide en cuatro partes:

- Costos de la mano de obra.
- Gastos generales.
- Amortizaciones.
- Gestión de fuentes.

2. ESTUDIO ECONÓMICO.

Para poder tasar correctamente nuestro proyecto, hay que investigar el valor de las distintas partes.

2.1 Costos de la Mano de Obra.

En esta parte del presupuesto se definen el coste económico ocasionado por el trabajo del ingeniero.

Se distinguen dos subtipos:

- Costos de mano de obra directa
- Costos de mano de obra indirecta

2.1.1 Costos de Mano de Obra Directa

En este subtipo se identifican los gastos ocasionados directamente por el ingeniero, es decir, el coste total del tiempo que el Ingeniero ha dedicado a la programación más al diseño del modelo y su respectivo controlador.

2.1.1.1 Coste del Tiempo Dedicado al Diseño por Parte del Ingeniero

Se procederá a estimar el tiempo invertido y los costes conforme a los estándares de rendimiento habitual en una empresa de software para el trabajo de diseño no es necesaria la intervención del ingeniero:

- Diseño y análisis
- Programación
- Comprobacion y corrección de errores

Se establece un salario de 30€ la hora de Ingeniero y 15€ la hora de Técnico y se procede al cálculo de las horas empleadas:

- Montaje y preconfiguración del sistema: 10h Técnico.
- Modelización de los sistemas: 30h Ingeniero
- Diseño y análisis de controladores: 40h Ingeniero
- Programación: 100h Ingeniero
- Estudio y búsqueda de información: 20h Ingeniero
- Comprobación del programa y corrección de errores: 20h Técnico
- Redacción de informe y memoria: 30h Técnico
- Redacción de anejos y manual de usuario: 30h Ingeniero

Total: 60h Técnico + 220h Ingeniero

Lo que asciende a un total de: 6600€ + 900€ = 7500€

2.1.1.2 Costo de la Paga Extraordinaria

Suponiendo una paga extraordinaria del equivalente al salario de dos meses al año. La parte correspondiente al trabajo realizado será por tanto:

Horas trabajadas al mes:

$$8h * 22días = \frac{176h}{mes} \rightarrow \left(\frac{2112h}{año} \right)$$

$$Salario \text{ al mes ingeniero} = 176 * 30 = 5280€$$

$$Salario \text{ al mes técnico} = 176 * 15 = 2640€$$

Ahora:

$$\frac{220}{2112} = 0.10416 \text{ años trabajados por el Ingeniero}$$

$$\frac{60}{2112} = 0.0284 \text{ años trabajados por el Técnico}$$

Sumando así un coste adicional de:

$$0.104 * 2 * 5280 = 1100€$$

$$0.0284 * 2 * 2640 = 150€$$

2.1.1.3 Coste total de la Mano de Obra Directa

El coste total ocasionado por la mano de obra directa será igual a:

$$7500 + 1250 = 8250€$$

2.1.2 Costes de Mano de Obra Indirecta

Al tratarse de un proyecto individual llevado a cabo por tan solo un Ingeniero, no habrá ningún coste debido a mano de obra indirecta. La Oficina Técnica estará formada por un solo Ingeniero que trabajará también como Técnico.

2.1.3 Costo Total de la Mano de Obra

El coste total de la mano de obra será igual a: 8250€

2.2 Gastos Generales

Dentro de los gastos generales se distinguirá entre dos tipos:

- Gastos de mantenimiento
- Gastos por necesidad de material

No se considerarán los costes generados por las necesidades de software y hardware como gastos generales debido a que no se realizará un uso exclusivo, no obstante, se obtendrán los costes ocasionados por el mantenimiento de estos.

2.2.1 Gastos de Mantenimiento

En este apartado se incluirán los gastos debidos al mantenimiento del hardware y también los gastos de mantenimiento de la Oficina Técnica.

Se considerará que los gastos de mantenimiento son equivalentes al 2% del coste total de la mano de obra.

$$2\% \text{ de } 8250 = 165\text{€}$$

2.2.2 Gastos por Necesidad de Material

Los gastos materiales incluirán el ordenador con las especificaciones técnicas necesarias para el funcionamiento del programa, el prototipo y la tarjeta de adquisición:

Ordenador de sobremesa:	315€
Monitor:	75€
Teclado y Ratón	9€
Tarjeta de adquisición	500€
Prototipo	150€
Impresión y encuadernación	8€
TOTAL:	957€

Los requisitos mínimos del ordenador vienen explicados en el Pliego de Condiciones

2.2.3 Total Gastos Generales

Entre los gastos de mantenimiento y los gastos materiales, la suma total de los gastos generales alcanza los

$$165\text{€} + 957\text{€} = 1122\text{€}$$

2.3 Amortizaciones

Se dividirán las amortizaciones en dos grupos:

- Amortizaciones del hardware
- Amortizaciones del software

2.3.1 Amortización del Hardware.

Proporcionalmente al coste, contando con que a los 5 años se amortiza la inversión,

$$\frac{280 \text{ horas}}{5 \text{ años}} = \frac{280h}{2112 * 5h} = 2.6515\%$$

Se considera que el valor del hardware empleado (equipos e instalaciones informáticas) en el desarrollo del proyecto ronda los 1200€

$$1200€ * 2.6515\% = 31.82€$$

El uso de una fuente de alimentación con un precio de 220€ y 15 años de amortización.

$$220 * \frac{2.615\%}{3} = 1.94€$$

Lo que suma un total de: 33.76€.

2.3.2 Amortización del Software

Se establece que el tiempo que se ha utilizado el software equivale a 1,6 meses. El periodo de amortización del software es de aproximadamente unos 3 años. Durante el proyecto, se han empleado los siguientes programas:

Windows 8.1 (incluido en el ordenador):	-	
Windows Office:	70€/año	9.33€
MatLab:	2000€/3años	88.89€
LabVIEW:	2630€/3años	116.89€
TOTAL:		215.11€

2.3.3 Amortización Total:

El valor total de la amortización asciende a:

$$33.76 + 215.11 = 248.87€$$

3. PRESUPUESTO FINAL

Tras el cálculo del coste de los distintos apartados del proyecto, se procede al recuento y suma de éstos:

Mano de obra:	8,250.00€
Gastos generales:	1,122.00€
Amortización:	248.87€
<hr/>	
Suma total del presupuesto de ejecución del material:	9,620.87€

Y tras aplicar los correspondientes impuestos:

Presupuesto de ejecución material:	9,620.87€
Gastos generales (15% del PEM):	1,443.13€
Beneficio Industrial (10% del PEM):	962.09€
<hr/>	
SubTotal Presupuesto Desarrollo	12,026.09€
IVA (21% del STPD):	2,525.48€

Esto suma un **PRESUPUESTO FINAL** de: **14,551.57€**

El presupuesto del proyecto asciende a un total de **Catorce mil quinientos cincuenta y un euros y cincuenta y siete céntimos**

Valencia 29 de junio de 2014