



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

AUTOR: Pérez Jaén, Juan

TUTOR: Herrero Durá, Juan Manuel

Curso Académico: 2013-14

A mi familia y amigos, sin vosotros no habría llegado hasta aquí.



RESUMEN DEL PROYECTO.

En el presente proyecto se han fijado como objetivos: el diseño del sistema de control de una maqueta de un helicóptero con dos grados de libertad (correspondientes a los ángulos de cabeceo y rotación) y el desarrollo de una aplicación en LabVIEW que permita el control de dicha maqueta.

Como paso previo a la realización de los ensayos, se han diseñado unas aplicaciones de LabVIEW que permitan realizar los ensayos. Para el diseño de estas aplicaciones se han calculado las rectas de conversión de los sensores. Se ha procedido a la linealización de la ganancia del proceso de cabeceo, ya que se observa una fuerte no linealidad que puede afectar a la identificación del modelo lineal. También se elimina la zona muerta del motor de rotación por la misma razón.

En primer lugar, se han realizado los ensayos necesarios para la identificación de los modelos del sistema y se ha decidido dividir la zona de funcionamiento del motor principal (cabeceo) en seis zonas, esto implica que es necesario la realización de cinco modelos, el motor de rotación se modelará con un único modelo para el rango de funcionamiento.

Tras realizar los ensayos, se ha procedido a la identificación de los modelos del sistema, el modelo elegido para el cabeceo es un modelo de tercer orden, éste consta de dos polos complejos y un cero, el modelo de rotación escogido es un primer orden con integrador.

El diseño escogido para los reguladores del motor principal es un regulador que cancela los polos complejos, con una constante del regulador que fije un tiempo de establecimiento menor de 4 segundos. El regulador escogido para el motor de rotación es un PD, con el cero que cancela el polo estable del sistema y una la constante que fija un tiempo de establecimiento menor de 4 segundos.



Desarrollo de una aplicación para la identificación y control de un helicóptero mediante LabVIEW



Una vez diseñados los reguladores, se procede al diseño de la aplicación de control de la maqueta. Para el control de los motores es necesario un PID; se ha decidido programar uno propio y no usar el que viene por defecto en LabVIEW. Se programa un PID con antiwindup para el control del motor de cabeceo y se implementa otro PID para el control de rotación, similar pero cambiando los límites de saturación del antiwindup.

Una vez programado el PID, se ha programado el planificador de ganancia para interpolar los parámetros del PID al más adecuado en cada punto de funcionamiento.

Para el uso de la maqueta se deben implementar cuatro modos de funcionamiento, éstos son:

- Automático, el helicóptero sigue la referencia que se le introduce mediante la interface de control.
- Perfil, el helicóptero sigue un perfil programado por el usuario mediante el panel de control en la pestaña de configuración.
- Modo Manual, en el cual la tensión que le llega a los motores la fija el usuario en el panel de control principal.
- Joystick, en el cual el helicóptero es controlado mediante un joystick que se encuentra situado en el mismo equipo o en otro conectado a la misma red Ethernet.

El control del helicóptero mediante el joystick se implementa mediante comunicación TCP si se encuentra en otro equipo, de tal forma que sea posible controlar el helicóptero de manera remota mediante el joystick. Con el joystick también es posible cambiar el modo de funcionamiento del helicóptero entre los cuatro posibles, gracias a los pulsadores que éste lleva incorporados.

Los objetivos de este proyecto se han alcanzado de manera adecuada, se ha conseguido controlar un proceso con polos complejos como dinámica predominante, se ha desarrollado una aplicación en LabVIEW que permite el control de la maqueta, tanto desde el mismo ordenador como desde otro.



ÍNDICE GENERAL

DOCUMENTO Nº 1: MEMORIA.

DOCUMENTO Nº 2: PRESUPUESTO.

DOCUMENTO Nº 3: PLIEGO DE CONDICIONES.

DOCUMENTO Nº 4: ANEXO DEL USUARIO.

DOCUMENTO Nº 5: ANEXO DE DISEÑO.

DOCUMENTO Nº 6: ANEXO DE PROGRAMACIÓN.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 1: MEMORIA

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



ÍNDICE DE CONTENIDO

1 INTRODUCCIÓN.....	7
1.1 OBJETO DEL PROYECTO.....	7
1.2 OBJETIVOS.....	8
1.3 MOTIVACIÓN.....	9
1.4 JUSTIFICACIÓN.....	9
1.5 ANTECEDENTES DEL CONTROL AUTOMATICO.....	10
1.5.1 Sistemas de control automático.....	10
1.5.2 Regulador PID.....	12
1.5.3 Introducción a LabVIEW.....	14
1.6 DESCRIPCIÓN DEL EQUIPO.....	15
1.6.1. Descripción de la maqueta de simulación.....	16
1.6.2 Tarjeta de adquisición de datos.....	18
2 PREPARACIÓN Y DISEÑO PREVIO.....	19
2.1 PREPARACIÓN.....	19
2.2 DISEÑO DEL CONTROLADOR PID.....	20
3 IDENTIFICACIÓN Y DISEÑO DE LOS REGULADORES.....	22
3.1 PROCESO DE CABECEO.....	23
3.1.1 Ensayos cabeceo zona 0°.....	24
3.1.2 Identificación cabeceo zona 0°.....	25
3.1.3 Diseño del regulador cabeceo zona 0°.....	27
3.1.4 Resumen reguladores cabeceo.....	30



3.2 PROCESO ROTACIÓN.	31
3.2.1 Ensayos.....	31
3.2.2 Identificación rotación.	32
3.2.3 Diseño del regulador rotación.....	32
4 DESARROLLO DE LA APLICACIÓN.	33
4.1 BUCLES DE ENCENDIDO Y APAGADO DE LA MAQUETA.	34
4.1.1 Bucle de encendido.	35
4.1.2 Bucle de apagado.	35
4.2 PLANIFICACIÓN DE LA GANANCIA.....	36
4.3 PERFIL DE LA REFERENCIA.	37
4.4 CONFIGURACIÓN DEL JOYSTICK.....	39
4.4.1 Preparación.	40
4.4.2 Modo de funcionamiento joystick.	41
5 APLICACIÓN FINAL.....	42
6 FUTURAS AMPLIACIONES.....	43
7 CONCLUSIONES.	44
8 BIBLIOGRAFÍA.	46



ÍNDICE DE FIGURAS.

Figura 1	Maqueta de un simulador de cabeceo y rotación de un helicóptero....	7
Figura 2	Entradas y salidas de un sistema.....	11
Figura 3	Estructura de control en bucle cerrado.	11
Figura 4	Ecuaciones continuas y discretizadas.....	13
Figura 5	Versión de LabVIEW empleada en el desarrollo del proyecto.....	15
Figura 6	Versión de Matlab empleada en el desarrollo del proyecto.....	15
Figura 7	Parte superior de la maqueta de simulación.	16
Figura 8	Parte inferior de la maqueta de simulación.....	17
Figura 9	Bornera de conexiones de la tarjeta de adquisición de datos.....	18
Figura 10	Gráfica linealización de la ganancia	19
Figura 11	Ecuaciones de linealización en el código de LabVIEW.	20
Figura 12	PID desarrollado en LabVIEW.....	20
Figura 13	Algoritmo PID ISA y ecuaciones continuas y discretas.....	21
Figura 14	Diagrama de bloques del seguido para el control del proceso.....	22
Figura 15	Ensayo válido para la identificación del modelo en la zona de 0°.	24
Figura 16	Subensayos zona 0°.	25
Figura 17	Modelo identificado.	26
Figura 18	Lugar de las raíces del modelo con las especificaciones.....	27
Figura 19	Lugar de las raíces con controlador.	28
Figura 20	Regulador zona 0°.....	29
Figura 21	Validación del regulador.....	29
Figura 22	Ensayo de rotación.	31
Figura 23	Diagramas de bloques de la aplicación.	33
Figura 24	Diagrama de bloques de los bucles de encendido y parada.....	34
Figura 25	Bucle de parada código de LabVIEW.....	35
Figura 26	Diagrama de bloques del interpolador de los parámetros del PID....	36
Figura 27	Interpolate.vi	36
Figura 28	Diagrama de bloques del funcionamiento del perfil.....	37
Figura 29	svi-perfil.vi	38



Figura 30 Perfil predefinido en la aplicación.....	38
Figura 31 Joystick empleado en el control.....	39
Figura 32 Rectas de conversión de los sensores del joystick.....	40
Figura 33 Lógica de control del joystick.vi	40
Figura 34 Funcionamiento joystick.	41
Figura 35 Lógica de control.	42



ÍNDICE DE TABLAS.

<i>Tabla 1 Modelos identificados zona 0°</i>	26
Tabla 2 Media modelos y modelo final.....	26
Tabla 3 Regulador zona 0°.....	29
Tabla 4 Modelos del proceso de cabeceo.....	30
Tabla 5 Reguladores proceso cabeceo.....	30
Tabla 6 Modelos del proceso de rotación.....	32
Tabla 7 Parámetros control rotor	32

1 INTRODUCCIÓN.

1.1 OBJETO DEL PROYECTO.

El presente proyecto tiene como finalidad el control de una maqueta de un helicóptero con dos grados de libertad (cabeceo y rotación), para ello será necesario actuar sobre los motores de rotación y cabeceo del helicóptero. También tiene como propósito el desarrollo de una aplicación en LabVIEW, capaz de gestionar y controlar el helicóptero. En la Figura 1 se puede ver una imagen de la maqueta de simulación que se ha empleado en el proyecto.



Figura 1 Maqueta de un simulador de cabeceo y rotación de un helicóptero



1.2 OBJETIVOS

Los objetivos del presente proyecto son:

- Identificación y control de un proceso con polos complejos como dinámica dominante. El regulador diseñado debe de cumplir: sobreoscilación menor del 4,3%, tiempo de establecimiento menor de 4 segundos y error de posición nulo, para un cambio en la referencia.
- Identificación y control del proceso de rotación que presenta un modelo de primer orden con integrador. El regulador diseñado debe de cumplir las condiciones detalladas a continuación: tiempo de establecimiento menor de 4 segundos y error de posición nulo, para un cambio en la referencia.
- Implementación de un PID mediante las ecuaciones necesarias, se ha de programar e implementar un PID propio.
- Desarrollo de una aplicación en LabVIEW para el control de la maqueta, ésta ha de controlar el cabeceo y la rotación con los controles calculados en los apartados anteriores.
- Utilización de un joystick para el control remoto de la maqueta.
- Implementación de diversos modos de funcionamiento de la maqueta, se han de implementar cuatro modos de funcionamiento en la aplicación:
 - ◆ Automático, la referencia que se le introduce manualmente.
 - ◆ Perfil, el helicóptero sigue un perfil programado por el usuario.
 - ◆ Modo Manual, se controla la tensión de los motores directamente.
 - ◆ Joystick, en el cual el helicóptero es controlado mediante un joystick, éste se puede encontrar en el mismo equipo o en otro.



1.3 MOTIVACIÓN

El desarrollo del presente proyecto tiene la motivación de la aplicación y profundización de los conocimientos adquiridos en las tres asignaturas de control y automática impartidas en el grado en tecnologías industriales. SAU (sistemas automáticos), TAU (tecnología automática) y LAC (laboratorio de automática y control). También la aplicación y mejora de los conocimientos adquiridos en las asignaturas de electrónica y programación, éstas son: tecnología informática industrial, sistemas electrónicos y tecnología electrónica.

Así como la adquisición de conocimientos útiles en la programación en LabVIEW, ya que en la carrera es un programa que solamente se trata en unas prácticas, profundización en los conocimientos de Matlab y uso de las herramientas de control que éste incorpora, como pueden ser Ident Tool, rltool. Con la finalidad de preparar el desarrollo del master industrial especializado en control y automática.

1.4 JUSTIFICACIÓN

Se ha elegido el control de un helicóptero para desarrollar y ampliar los conocimientos adquiridos de control de procesos, ya que la complejidad del proceso de cabeceo con polos complejos dominantes supone un reto al no haber tratado con ningún proceso similar anteriormente. El control de la rotación presenta el reto de ser un proceso con integrador.

Se ha escogido el desarrollo de una aplicación en LabVIEW para el control de la maqueta de simulación de un helicóptero para desarrollar los conocimientos adquiridos de programación en LabVIEW, ya que es un programa ampliamente extendido en la industria debido a que es muy versátil.



1.5 ANTECEDENTES DEL CONTROL AUTOMÁTICO.

En este apartado se va a realizar una breve introducción a los antecedentes del control por computador, ya que tratar de manera detallada estos antecedentes queda fuera del alcance del proyecto.

1.5.5 Sistemas de control automático

La Ingeniería de Control tradicionalmente ha estado vinculada al sector industrial para el control de sistemas físicos. Pero últimamente, ésta se ha extendido a la extensa mayoría. De este modo, la Ingeniería de Control se presenta como una disciplina con múltiples aplicaciones.

El control en el siglo XXI presenta múltiples disciplinas, por ejemplo, si deseamos analizar y controlar un sistema continuo se puede modelar en cuanto a su respuesta temporal mediante la utilización de la función de transferencia o mediante la técnica de variables de estado. Así se representan dos posibilidades de representación una “externa” y otra “interna” para sistemas de mayor complejidad. Ambas posibilidades agrupan tras de sí métodos de cálculo básicamente diferentes, dando lugar en el caso de los sistemas de control a las denominadas “teoría clásica de control” y “teoría moderna de control”. La teoría de control clásica se caracteriza por el análisis y el diseño mediante métodos de cálculo basados en transformadas de Laplace y la respuesta del sistema, fundamentalmente en el dominio de la frecuencia. La teoría moderna de control se ha desarrollado como consecuencia de la aparición de los ordenadores, gracias a su tremendo potencial de cálculo.

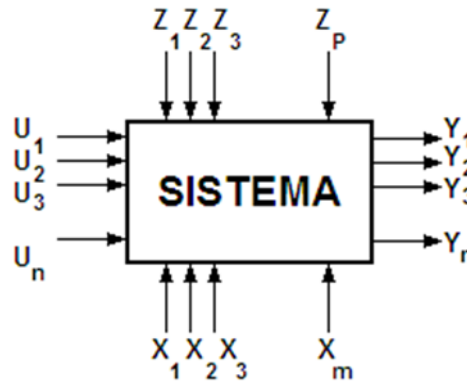


Figura 2 Entradas y salidas de un sistema.

En comparación con la teoría moderna, la teoría clásica tiene la ventaja de hacer menos énfasis en las técnicas matemáticas y más en la comprensión clara del comportamiento físico del sistema. Por otra parte, en determinadas circunstancias, la solución clásica no solamente es la más sencilla sino la más adecuada.

La teoría clásica de control estudia el comportamiento dinámico del sistema ante las perturbaciones, a partir de las relaciones existentes entre las variables de entrada y las de salida, caracterizando así, su comportamiento externo. La teoría moderna de control caracteriza el sistema mediante las variable de estado; esto es, por su comportamiento interno.

Los sistemas físicos son representados para su estudio por modelos integrados por componentes ideales, cuyo comportamiento pueda ser definido mediante expresiones matemáticas. Seleccionado el modelo y comprobado que existe suficiente coincidencia entre los resultados calculados y el comportamiento real del sistema, puede ser utilizado para sustituir al sistema real en la realización de estudios de tipo teórico.

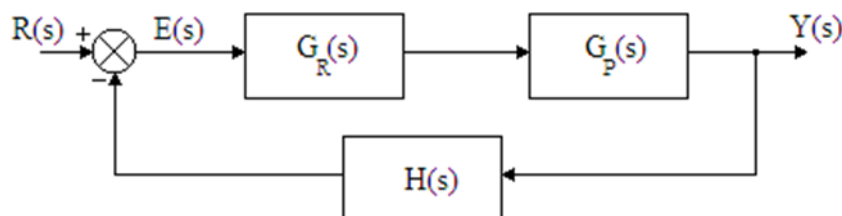


Figura 3 Estructura de control en bucle cerrado.



La estructura de control más extendida es el control en bucle cerrado, en la Figura 3 se puede ver como la señal de salida es realimentada, para restarse sobre la acción de control y llevar la salida a su valor correcto. Para esto es necesario medir continuamente la señal de salida para poder compararla con la referencia o consigna, de forma que cuando exista una discrepancia entre ambas el regulador reaccione en el sentido de anularla.

Dentro de los sistemas de regulación, los de tipo lineal representan a un grupo de gran importancia, en primer lugar porque son los más comúnmente utilizados y por otra parte porque es precisamente para este tipo de sistemas para los que se ha desarrollado los métodos matemáticos de cálculo más potentes. A ello contribuye también el que el comportamiento de determinados elementos no lineales y su influencia sobre el sistema, en numerosos casos, solamente puede ser estudiado aproximando dichos elementos por otro u otros tipos, de comportamiento lineal.

1.5.2 Regulador PID.

Los controladores tipo PID son con mucha diferencia los más utilizados en la industria de control de procesos. Su sencillez, versatilidad y capacidad para resolver los problemas básicos y requisitos de funcionamiento modestos hacen de ellos una herramienta imprescindible en el control de procesos.

A lo largo de las últimas décadas los controladores PID han sobrevivido a diferentes cambios tecnológicos que van desde los primeros controladores desarrollados a partir de elementos neumáticos hasta los desarrollados con microprocesadores pasando por las válvulas electrónicas, los dispositivos analógicos, los transistores y los circuitos integrados. La incorporación de los microprocesadores ha tenido un impacto muy grande ya que ha permitido que los controladores PID se enriquezcan con más funciones sin perder ninguna de sus propiedades, así se les ha añadido posibilidades de ajuste automático de los parámetros, posibilidades de ejecución de reglas lógicas o automatismos secuenciales.



La estructura de un regulador PID viene caracterizada por la forma:

$$u(t) = K \left[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d(\tau) + T_d \frac{de(t)}{dt} \right]$$

Donde $u(t)$ es la variable controlada y $e(t)$ es el error expresado como $e(t) = r(t) - y(t)$. La ecuación muestra los tres tipos de acciones de control: la acción proporcional K , la acción integral y la acción derivativa. Para ajustar el comportamiento del controlador se dispone de tres parámetros; la ganancia proporcional K , el tiempo integral T_i y el tiempo de la derivada T_d .

La función de transferencia del controlador PID es la siguiente:

$$\frac{U(s)}{E(s)} = K \left[1 + \frac{1}{T_i s} + T_d s \right] = \frac{K(1 + T_i s + T_i T_d s^2)}{T_i s}$$

Para el diseño de reguladores PID, ajustando los tres parámetros K , T_i y T_d . Para ello se parte de unas especificaciones estáticas (error de posición) y de unas especificaciones dinámicas (tiempo de establecimiento y sobreoscilación). Con el fin de poder implementar un regulador PID en un computador se ha de discretizar la ecuación del regulador PID para convertirla en un algoritmo de control y sea fácil de programar.

$$U(t) = K_c \left(E(t) + \frac{1}{T_i} \int_0^t E(\tau) d\tau + T_d \frac{dE(t)}{dt} \right)$$

$$U_p(t) = K_c E(t)$$

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

$$U_i(t) = K_c \frac{1}{T_i} \int_0^t E(\tau) d\tau$$



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

$$U_d(t) = K_c T_d \frac{dE(t)}{dt}$$

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

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

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

Figura 4 Ecuaciones continuas y discretizadas.



1.5.3 Introducción a LabVIEW.

LabVIEW es una plataforma de programación gráfica que ayuda a ingenieros a escalar desde el diseño hasta pruebas y desde sistemas pequeños hasta grandes sistemas. Ofrece integración sin precedentes con software legado existente, IP y hardware al aprovechar las últimas tecnologías de cómputo. LabVIEW ofrece herramientas para resolver los problemas de hoy en día y la capacidad para la futura innovación, más rápido y de manera más eficiente (Instruments, s.f.).

LabVIEW es una plataforma estándar en la industria de test y medida, para el desarrollo de sistemas de prueba y control de instrumentación; en el campo de la automatización industrial, para la adquisición de datos, análisis, monitorización y registro, así como para el control y monitorización de procesos; en el área de visión artificial, para el desarrollo de sistemas de inspección en producción o laboratorio. En los últimos años ha crecido en nuevos campos de trabajo como simulación, diseño de control y sistemas embebidos en tiempo real.

LabVIEW incorpora un lenguaje desarrollado por National Instruments, es un lenguaje de programación gráfico de propósito general, como es el Lenguaje C o Basic, pero con la característica que es totalmente gráfico. Está basado en la programación modular, lo que permite crear tareas muy complicadas a partir de módulos o submódulos mucho más sencillos.

LabVIEW permite la creación de instrumentos virtuales (.vi), la ventaja de LabVIEW es que permite crear archivos ejecutables en ordenadores que no tengan instalados el programa.

1.6 DESCRIPCIÓN DEL EQUIPO.

El equipo empleado en el desarrollo de la aplicación ha sido una maqueta de simulación del cabeceo y rotación de un helicóptero y un ordenador equipado con una tarjeta de adquisición de datos DAQ-LVIEW PnP PCI-9112 de Adlink y la bornera de conexionado de ésta. El equipo ha de ser capaz de ejecutar con soltura los programas

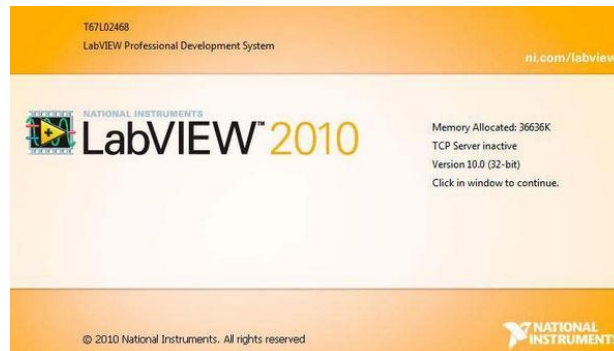


Figura 5 Versión de LabVIEW empleada en el desarrollo del proyecto

LabVIEW y Matlab, que tendrá instalados.

La versión del software utilizada de LabVIEW, desarrollado por National Instruments, ha sido la versión 2010.

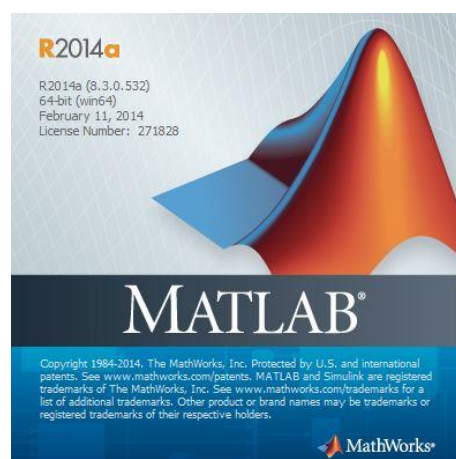


Figura 6 Versión de Matlab empleada en el desarrollo del proyecto.

La versión de Matlab, desarrollado por mathworks, empleada ha sido la 2014a, en la Figura 6 se puede ver la pantalla de inicio del mismo.

1.6.1. Descripción de la maqueta de simulación.

La maqueta de simulación es la que aparece en la Figura 1. Las partes que la componen se detallan en los siguientes puntos.

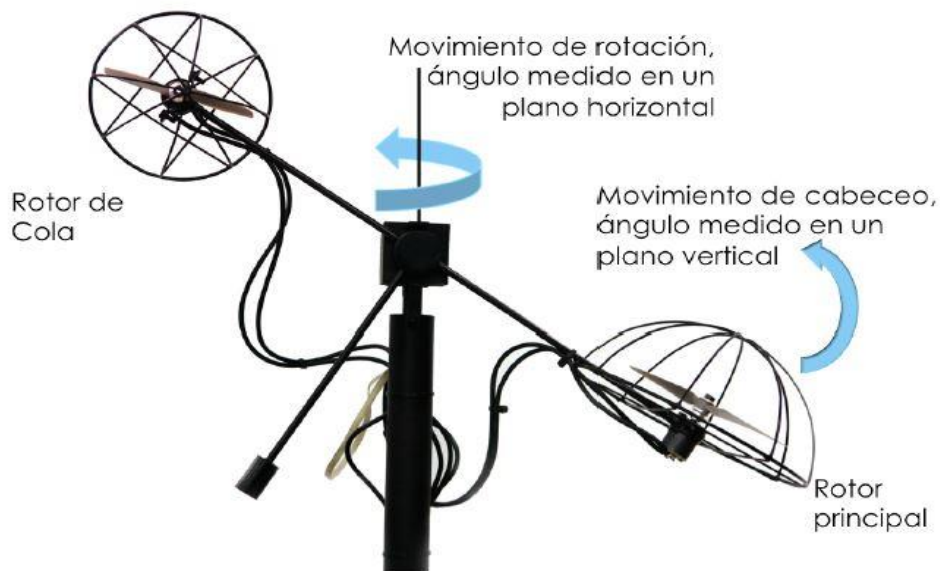


Figura 7 Parte superior de la maqueta de simulación.

La parte superior de la maqueta, Figura 7, está compuesta por lo que sería el cuerpo del helicóptero real, en ésta se encuentran los motores de cabeceo (rotor principal) y el motor de rotación (rotor de cola), esta maqueta solo permite la simulación de dos grados de libertad (cabeceo y rotación), un helicóptero real dispone de nueve grados de libertad.

También se puede apreciar la presencia de los cables de alimentación de los motores, éstos interfieren en la simulación añadiendo dinámica que el helicóptero real no presenta, para disminuir la influencia de los cables al mínimo, se ha limitado el rango de funcionamiento de la maqueta a $\pm 90^\circ$ de rotación y $\pm 30^\circ$ de inclinación.

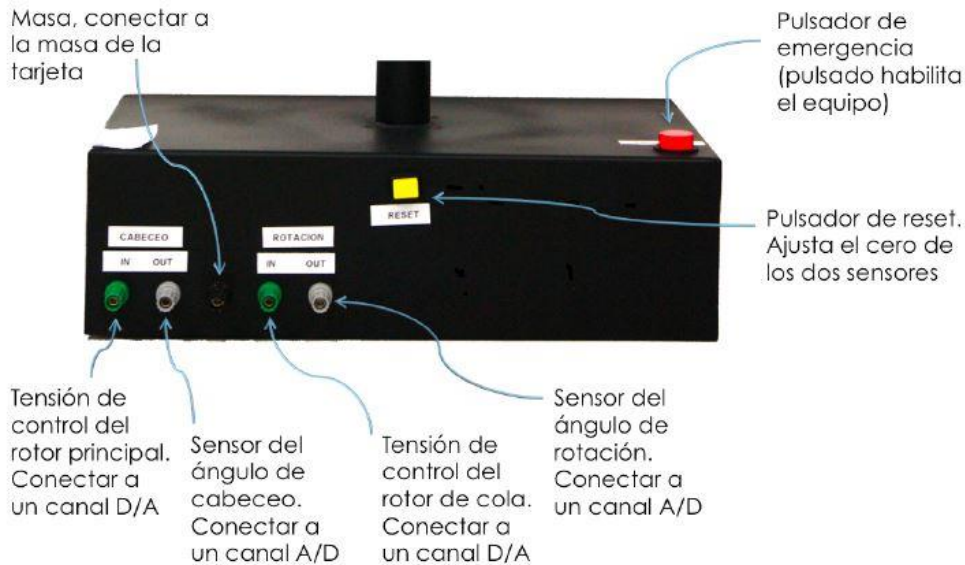


Figura 8 Parte inferior de la maqueta de simulación.

En la parte inferior de la maqueta, Figura 8, lleva implementada la parte de potencia necesaria para su control. En esta parte se encuentran localizados los bornes de conexión y los pulsadores. Su función se describe a continuación:

- Tensión de control del cabeceo: En este borne se conecta la tensión de control del motor de cabeceo, esta tensión será analógica y tendrá un valor entre 0V y 10V.
- Sensor del ángulo de cabeceo: Este borne se conectará a una entrada analógica, el rango de tensión proporcionada por el sensor es $\pm 10V$.
- Tensión de control del sensor de cola: Se ha de conectar la tensión del motor de rotación, esta tensión será analógica y tendrá un valor comprendido entre $\pm 10V$.
- Sensor del ángulo de rotación: Se ha de conectar a una entrada analógica de la tarjeta, el valor de la tensión proporcionado está comprendido entre $\pm 10V$.
- Masa: Se ha de conectar a masa de la tarjeta de adquisición de datos.
- Pulsador RESET: Ajusta la posición de equilibrio del helicóptero, fija el cero de los sensores.
- Botón de emergencia: Pone en funcionamiento la maqueta de simulación.

1.6.2 Tarjeta de adquisición de datos.

La tarjeta de adquisición de datos empleada es la DAQ-LVIEW PnP PCI-9112 de Adlink, se encuentra instalada en el interior del ordenador, se utiliza una bornera de conexiones que dispone de seis entradas analógicas, dos salidas analógicas y cinco salidas digitales. Las salidas analógicas disponibles son dos, pero se dispone de cuatro conexiones: dos en las que la salida está en el intervalo 0-10V y otras dos en las que la salida está en el rango de $\pm 10V$. Las salidas a nivel de programación son las mismas, solo que para la utilización de las segundas, si se escribe un valor de 0, la salida proporcionará -10V y si se escribe un valor de 5 la salida proporcionará una tensión de 0V.



Figura 9 Bornera de conexiones de la tarjeta de adquisición de datos.

Las conexiones se realizarán de la siguiente forma: el sensor de cabeceo se conectará a E0, el sensor de rotación a E1. El motor de cabeceo se conectará a S0 de 0-10V y el motor de rotación a S1 de $\pm 10V$, el borne de masa se conectará a cualquiera de los tres disponibles.

2 PREPARACIÓN Y DISEÑO PREVIO.

Para el correcto desarrollo de los ensayos, han sido necesarios la realización de diversos cálculos y el desarrollo de unas aplicaciones de LabVIEW con la finalidad de la realización de los ensayos y posterior guardado de los datos. Todo esto es previo a la identificación, desarrollo de los controladores y de la aplicación principal.

En primer lugar, se escoge el periodo de ejecución que se empleará en todos los bucles de control programados en LabVIEW, también se empleará para los cálculos en Matlab. El periodo seleccionado para el control del helicóptero es 50 milisegundos, es necesario un periodo de muestreo relativamente rápido para controlar un proceso como el controlado en el helicóptero.

2.1 PREPARACIÓN.

Lo primero que se ha realizado es el cálculo de las rectas de conversión de los sensores de cabeceo y rotación. La recta conversión del sensor de cabeceo es $Y=9*X$ y la del sensor de rotación es $Y=14*X$, las unidades de Y son grados y las de X voltios.

En la Figura 10 se puede ver la gráfica con la ganancia del sistema de cabeceo y la recta de la ganancia lineal.

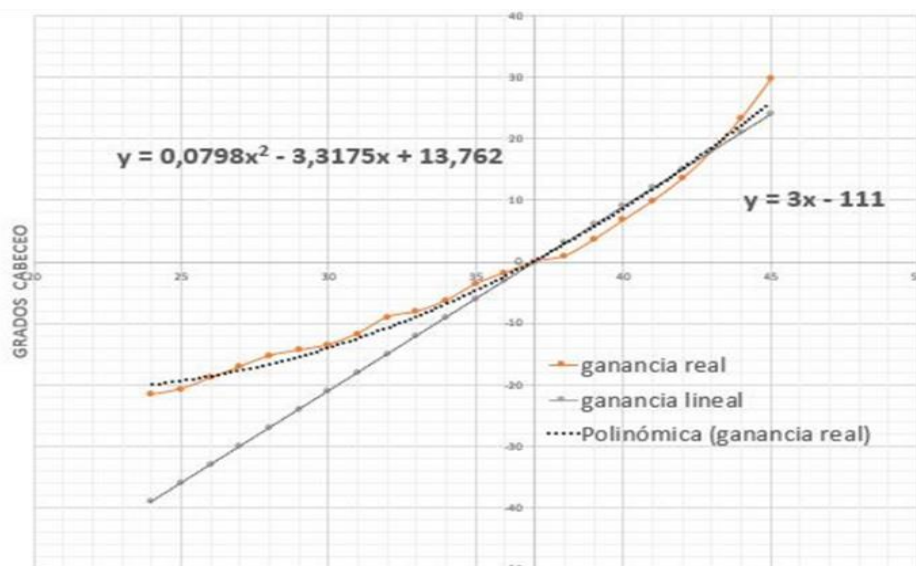


Figura 10 Gráfica linealización de la ganancia

Ha sido necesaria la linealización de la ganancia del proceso de cabeceo, ya que se observa una gran no linealidad que puede afectar al proceso de obtención de un modelo lineal mediante identificación. Para ello se han realizado diversas medidas de la posición de cabeceo para diversos valores de tensión y se ha procedido al cálculo de las ecuaciones necesarias para convertir la ganancia en lineal, este cálculo se ha tratado con más detalle en el anexo de diseño. En la Figura 11 se ven las ecuaciones de linealización.

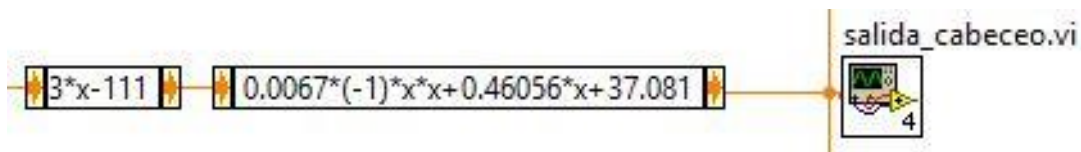


Figura 11 Ecuaciones de linealización en el código de LabVIEW.

En la platilla de rotación se ha procedido a la eliminación de la zona muerta del motor.

Una vez realizados estos preparativos, se puede proceder a la identificación de los modelos.

2.2 DISEÑO DEL CONTROLADOR PID.

Antes de realizar el control de la posición del helicóptero es necesaria la programación de un PID. Se decide programar un PID propio y no la utilización del PID disponible en la toolbox de LabVIEW.

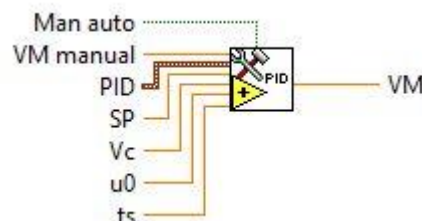


Figura 12 PID desarrollado en LabVIEW.

Se desarrolla un PID propio que permite el control de los motores. Se usarán dos de estos PIDs para controlar los motores, cada uno se ha modificado para el control del motor oportuno.

La ecuación programada es la ecuación del algoritmo PID ISA. Para el control por ordenador es necesario discretizar la ecuación, ya que el control por ordenador se realiza de forma discreta, mientras que la realidad es continua. En la Figura 13 se pueden ver las ecuaciones continuas del controlador y las ecuaciones discretizadas.

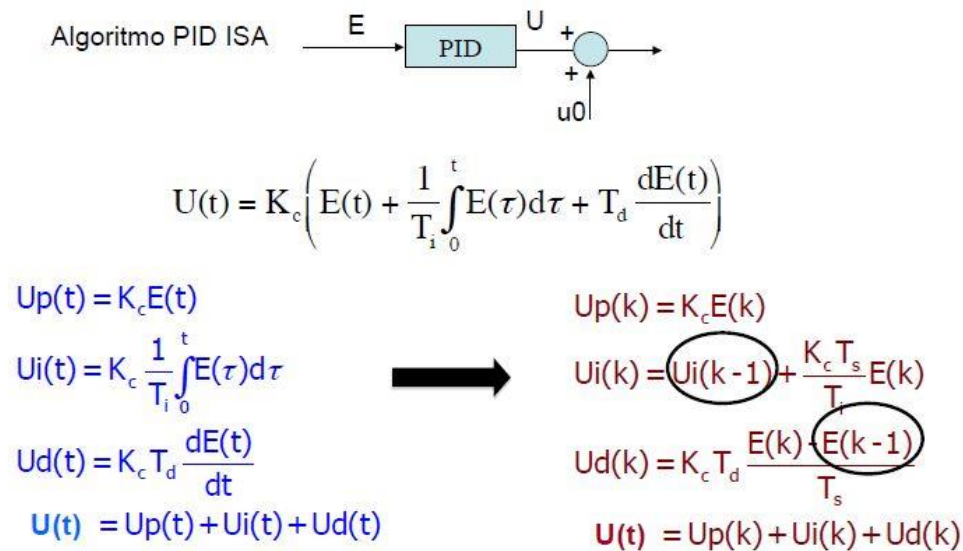


Figura 13 Algoritmo PID ISA y ecuaciones continuas y discretas.

Como se puede ver en la figura 10, para la implementación de estas ecuaciones es necesario el conocimiento del error en el ciclo de ejecución anterior. Como en LabVIEW no hay direcciones de memoria física para el guardado de las mismas, éste se realiza mediante la realimentación de variables disponibles en LabVIEW. Para más detalle ir al anexo de programación.

3 IDENTIFICACIÓN Y DISEÑO DE LOS REGULADORES.

Una vez disponemos de las aplicaciones para la realización de los ensayos, que darán los datos para la identificación de los modelos, procedemos a la identificación y diseño de los reguladores.

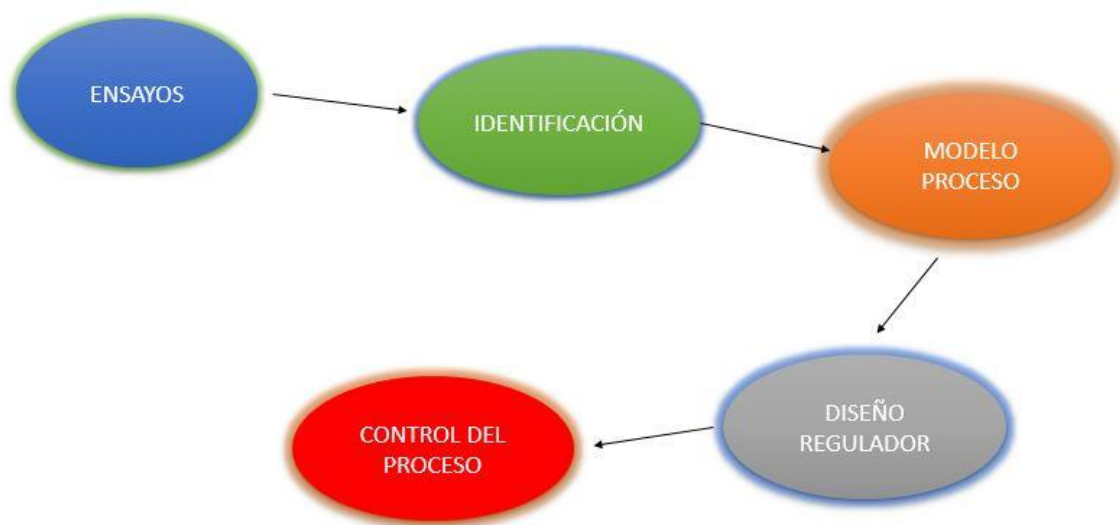


Figura 14 Diagrama de bloques del seguido para el control del proceso.

El proceso que se seguirá para controlar los procesos de cabeceo y rotación se describe en el diagrama de bloques de la Figura 14.

Los procesos que se identificarán y controlarán en el presente proyecto son el cabeceo y la rotación de la maqueta del helicóptero. En los próximos apartados se describe el procedimiento seguido, para más información de los cálculos y procedimientos seguidos leer el anexo de diseño.



3.1 PROCESO DE CABECEO

Para el cabeceo se ha comprobado, con los ensayos previos, que cambia el comportamiento del sistema dependiendo de la inclinación del helicóptero, por lo que se ha decidido dividir el rango de funcionamiento en seis zonas, lo que conlleva el diseño de cinco reguladores. Esto permitirá, mediante la planificación de ganancia, que el sistema funcione con un regulador más adecuado en cada punto de funcionamiento, en lugar de si únicamente se controlara con un único regulador.

Los puntos en los que se diseñarán estos reguladores y, por tanto, en los que se realizarán los ensayos serán: -28° , -15° , 0° , 15° y 25° . Estos puntos están dentro del rango de funcionamiento fijado, que es $\pm 30^\circ$.

Como el procedimiento y cálculos realizados para el diseño de los cinco reguladores ha sido el mismo, para simplificar la memoria y no saturar la misma con imágenes y datos diferentes pero con la misma información, se detalla a continuación los ensayos, identificación, modelo, y regulador diseñado para la zona de funcionamiento de 0° . Al final de este apartado se detallarán los modelos identificados y reguladores diseñados para cada zona de funcionamiento.

El modelo que se ha identificado para el proceso de cabeceo es un sistema de tipo cero (no tiene integrador) y de tercer orden con tres polos, dos de ellos complejos, que fijan la dinámica dominante del sistema y un polo real. El regulador para el control mediante un PID se diseña por cancelación de los polos complejos del sistema.

En el anexo de diseño se dan otras posibles soluciones para la identificación y control, así como la justificación de por qué se han seleccionado estas soluciones.

3.1.1 Ensayos cabeceo zona 0°.

Para que un ensayo sea válido para la identificación es necesario llevar el sistema al punto de equilibrio en el cual se quiere la identificación del modelo y se realizan escalones de valor un 5% en la variable manipulada, después se espera a que el sistema se estabilice y vuelve al punto de equilibrio. En este proyecto, para la identificación se han realizado en cada punto de equilibrio dos escalones positivos y dos negativos, la variable controlada no ha de saturar para que un ensayo sea válido, esto quiere decir que no ha de salir del rango de funcionamiento.

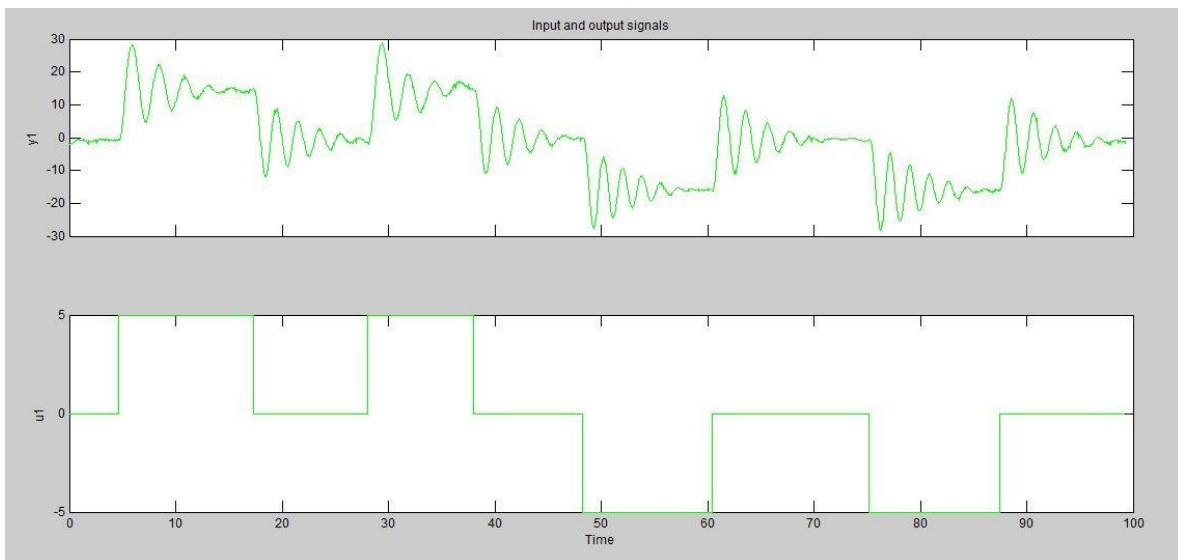


Figura 15 Ensayo válido para la identificación del modelo en la zona de 0°.

En la Figura 15, se pueden ver los datos válidos empleados para la identificación del modelo en la zona de 0°.

3.1.2 Identificación cabeceo zona 0°.

Para la identificación se ha utilizado Ident Tool de Matlab. Ident permite la identificación de los modelos; el operario ha de indicar a Ident el tipo de modelo que ha de identificar, en nuestro caso el modelo parece a simple vista un segundo orden con dos polos complejos sin retardo. Tras realizar varias identificaciones, vemos que los modelos que más se aproximan a nuestro sistema son los modelos de tercer orden con dos polos complejos y un polo real.

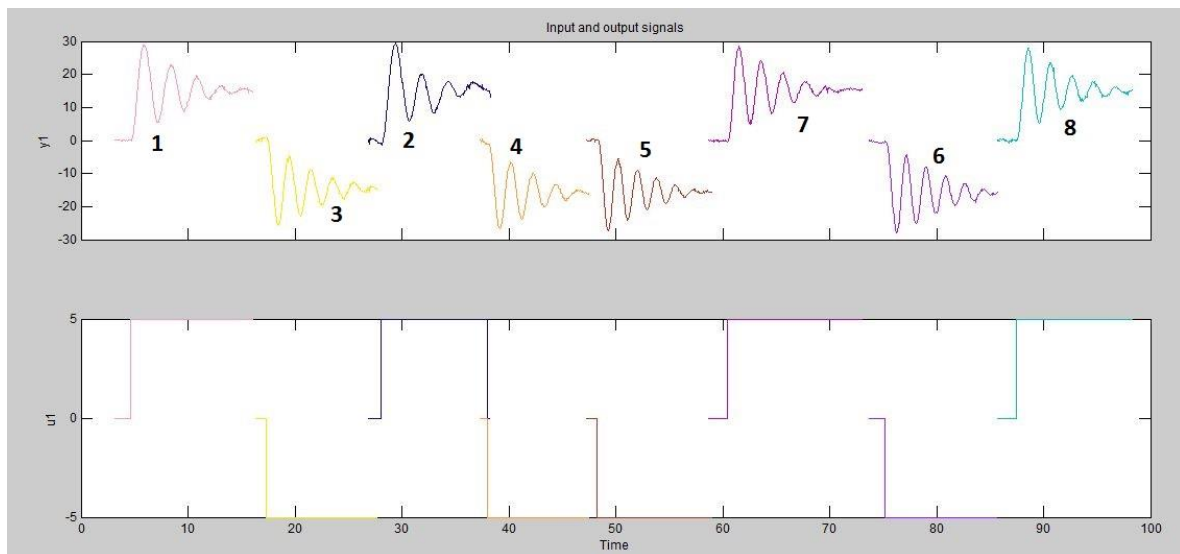


Figura 16 Subensayos zona 0°.

En la Figura 16 se ven los datos del ensayo de 0 ° dividido en subensayos y procesado para realizar la identificación de los modelos. Se han numerado atendiendo al proceso realizado para la identificación.

Para que el modelo sea lo más preciso posible se realizan ocho identificaciones, como se dispone de cuatro escalones se realizará la identificación con un escalón y la validación con el otro escalón similar. De esta manera, se realizará la media de los ocho modelos para la obtención de los parámetros del modelo en la zona de funcionamiento, que en este caso es 0°.

En la Figura 17 se puede ver la forma que tiene la función de transferencia del modelo identificado, se pueden ver los parámetros que proporciona la identificación.

$$G(s) = \frac{Kp}{(1+2*Zeta*Tw*s+(Tw*s)^2)(1+Tp3*s)}$$

Figura 17 Modelo identificado.

En la Tabla 1 se pueden ver los ocho modelos identificados para 0°.

escalón ¹	k	tw	zeta	tp3	% 1 ²	% 2 ³
1-2	3,0957	0,39303	0,090117	0,060563	78,52	86,08
2-1	3,0699	0,39432	0,10002	0,08902	84,54	79,53
3-4	2,9242	0,32256	0,074267	0,14559	76,51	90,53
4-3	3,1664	0,32955	0,082513	0,10502	76,77	89,65
5-6	3,1243	0,28741	0,06905	0,14977	88,17	90,82
6-5	3,1456	0,29097	0,060712	0,13197	87,41	91,46
7-8	3,0966	0,32829	0,075854	0,07749	91,78	90,6
8-7	3,0752	0,32628	0,077593	0,08564	90,11	92,37

Tabla 1 Modelos identificados zona 0°.

En la Tabla 2 se indica la media de los escalones positivos y negativos, así como los parámetros del modelo final para la zona de 0° de inclinación.

	k	tw	zeta	tp3
media escalón positivo	3,084	0,3604	0,0858	0,0781
media escalón negativo	3,090	0,3076	0,0716	0,1330
modelo final	3,087	0,3340	0,0787	0,1056

Tabla 2 Media modelos y modelo final.

¹ El nombre del escalón indica el subensayo utilizado para la identificación y el empleado para la validación del modelo identificado.

² % 1 indica el porcentaje correspondiente a la validación con los segundos datos empleados, en el primer caso serían los datos del subensayo 2.

³ % 2 indica el porcentaje correspondiente a la validación con los mismos datos empleados para identificar.

3.1.3 Diseño del regulador cabeceo zona 0°.

El diseño de los reguladores se realiza mediante el uso de rtool de Matlab. Una vez identificado el modelo de cada zona se procede al diseño del mismo. Para el diseño del regulador se procede introduciendo el modelo en rtool y poniendo las condiciones que debe de cumplir el regulador.

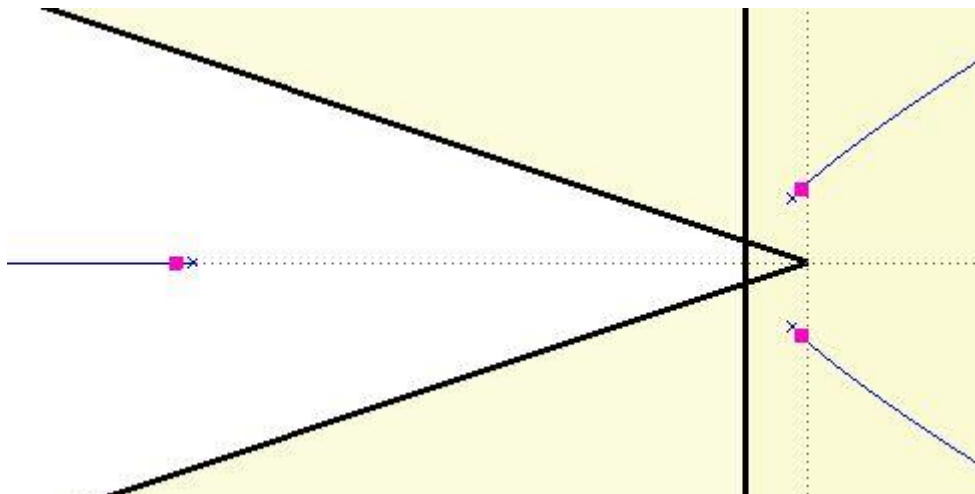


Figura 18 Lugar de las raíces del modelo con las especificaciones.

En la Figura 18 Lugar de las raíces del modelo con las especificaciones. se puede ver el lugar de las raíces del modelo con la zona de especificaciones, como se deduce del funcionamiento de un helicóptero. Los polos complejos están relativamente cerca del origen y son la dinámica dominante, lo que hace que el sistema se vuelva inestable fácilmente.

Para el control del proceso es necesario que el controlador disponga de un integrador, ya que las especificaciones exigen error de posición nulo. Esto lo proporciona un tipo de sistema igual a uno o superior y nuestro modelo es del tipo cero, como conclusión, nuestro regulador tiene que tener un integrador. Una vez colocado el integrador la única manera de controlar el proceso y estabilizarlo es colocar los dos ceros de los que dispone el PID.

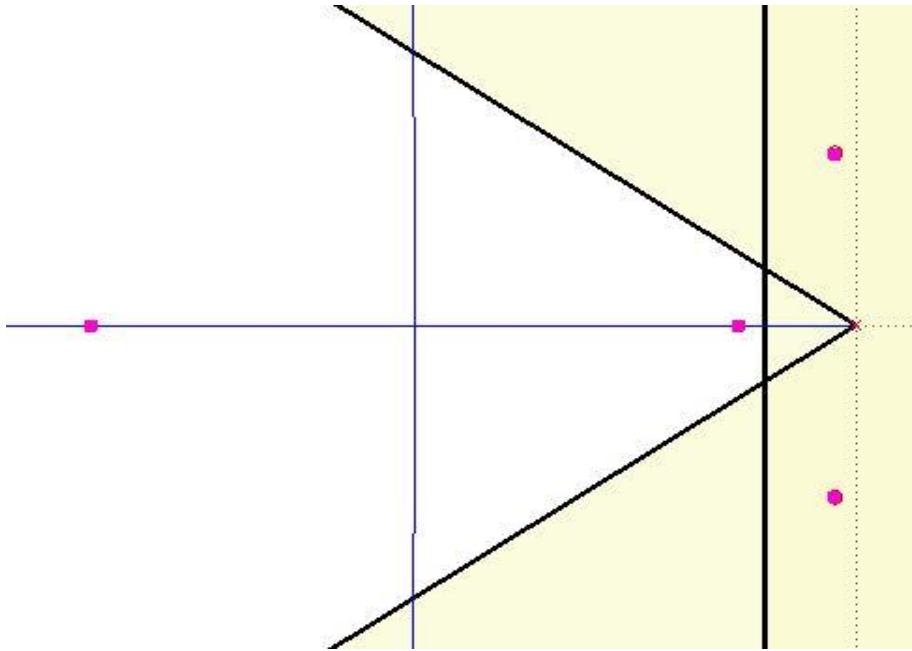


Figura 19 Lugar de las raíces con controlador.

En la Figura 19 se puede ver el lugar de las raíces tras colocar el regulador elegido, los ceros del PID son complejos y están “encima” de los polos complejos del sistema, por lo que el regulador cancela los polos complejos. La constante del regulador se selecciona para que el polo del integrador entre en la zona de especificaciones, de ésta manera cumpliremos el tiempo de establecimiento.

La sobreoscilación teórica es nula, ya que los polos del sistema en bucle cerrado se encuentran en el eje real, sin embargo la cancelación de los polos complejos puede no ser exacta, por lo que el sistema puede presentar algo de sobreoscilación, esta se comprobará en la validación experimental sobre la maqueta.

En la Figura 20, se ve el regulador final para la zona de 0°. El regulador es un PID completo en el que los ceros son complejos.

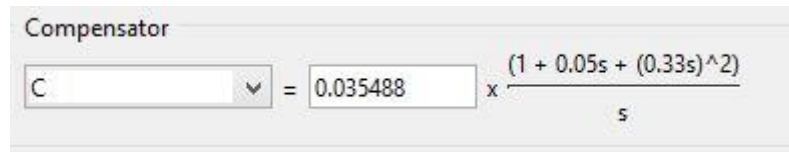


Figura 20 Regulador zona 0°.

Los parámetros del PID calculados a partir del regulador son los indicados en la Tabla 3.

ZONA REGULADOR	Kr	Ti	Td
0 GRADOS	0,162	0,052	2,120

Tabla 3 Regulador zona 0°.

Para la validación del regulador se disponen de herramientas como Simulink de Matlab, sin embargo en el presente proyecto se ha recurrido a la simulación experimental sobre la maqueta, para ello se ha repetido el ensayo de identificación pero con el PID implementado.



Figura 21 Validación del regulador.

Como se puede ver en la Figura 21, el regulador cumple las especificaciones marcadas en los objetivos, error de posición nulo, tiempo de establecimiento menor a 4 segundos y la sobreoscilación menor al 4,3%.

3.1.4 Resumen reguladores cabeceo.

El procedimiento seguido para la identificación del modelo para la zona de 0° se ha repetido cuatro veces para la identificación de los demás modelos, en la Tabla 4 se pueden ver los modelos identificados.

ZONA MODELO	k	tw	zeta	tp3
25 GRADOS	3,0430	0,4245	0,1232	0,0990
15 GRADOS	3,0430	0,4245	0,1232	0,0990
0 GRADOS	3,0872	0,0526	0,1116	0,1056
(-) 15 GRADOS	3,1905	0,2960	0,0592	0,1250
(-) 28 GRADOS	2,8899	0,2806	0,0569	0,1468

Tabla 4 Modelos del proceso de cabeceo.

Una vez se han identificado los modelos, se procede al diseño de los reguladores siguiendo la misma metodología empleada para 0°, los reguladores se diseñarán por cancelación de los polos complejos en todos los casos.

ZONA REGULADOR	Kr	Ti	Td
25 GRADOS	0,345	0,1675	1,745
15 GRADOS	0,318	0,105	1,722
0 GRADOS	0,162	0,053	2,121
(-) 15 GRADOS	0,112	0,035	2,502
(-) 28 GRADOS	0,092	0,032	2,464

Tabla 5 Reguladores proceso cabeceo.

En la Tabla 5 se pueden ver los parámetros de los cinco reguladores diseñados. Como se puede apreciar, los parámetros del regulador cambian en función de la zona de funcionamiento, por lo que se puede confirmar que se ha hecho bien en dividirla en varias, ya que la ganancia del proceso y la posición de los polos cambian.

3.2 PROCESO ROTACIÓN.

Para el modelado y diseño del regulador de la maqueta de simulación se realizará un único regulador a partir de los dos ensayos realizados.

3.2.1 Ensayos.

Para la identificación del modelo de rotación es más difícil realizar un ensayo válido, ya que al ser un modelo rápido con integrador el sistema satura rápidamente (gira y se sale del rango de funcionamiento). Tras varios intentos se realizan ensayos válidos.

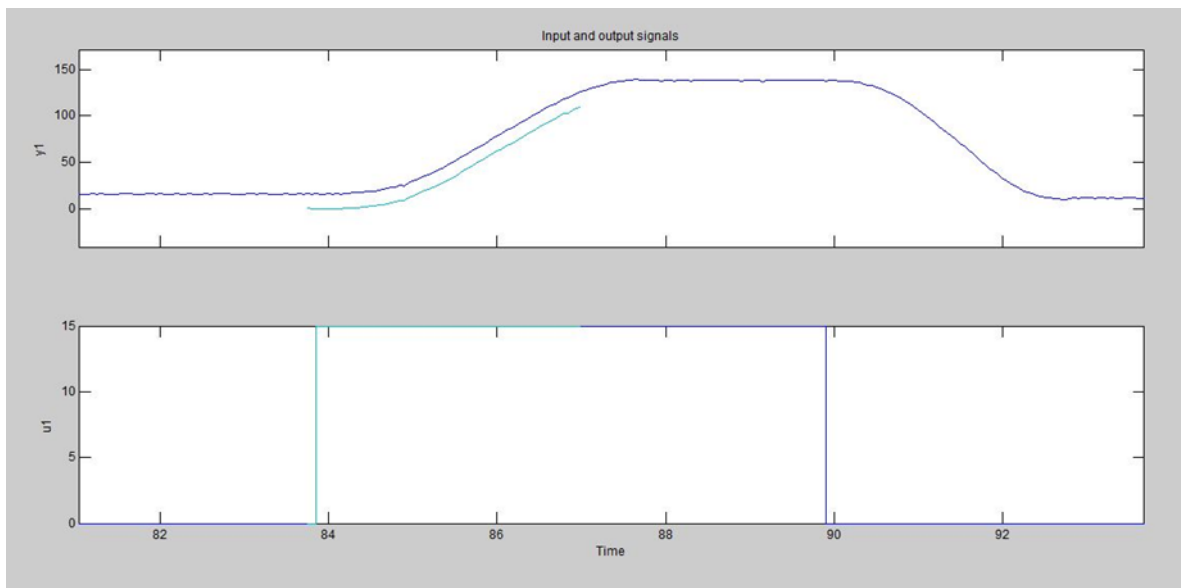


Figura 22 Ensayo de rotación.

En la Figura 22 se pueden ver los datos del ensayo utilizado para el modelado.

3.2.2 Identificación rotación.

Una vez realizados los ensayos, se procede a la identificación del modelo. El modelo identificado será un primer orden con integrador sin retardo. Con Ident se realiza una identificación para un escalón positivo y otra para un escalón negativo y se realiza la media de estos. En la Tabla 6 se pueden ver los parámetros modelados para el proceso de rotación y la media de los dos modelos para obtener el modelo utilizado para el diseño del regulador.

	Kp	Tp
Subida	11,3	0,766
Bajada	12,0	1,096
Modelo rotor	11,6	0,931

Tabla 6 Modelos del proceso de rotación.

El modelo identificado es un primer orden con integrador.

$$G(s)_{rotación} = \frac{11.6}{s(0.931 + s)}$$

3.2.3 Diseño del regulador rotación.

La solución adoptada para el control del movimiento de rotación de la maqueta ha sido un PD que se diseña sin cancelación, se usa rtool para el diseño del regulador, el regulador PD tiene la forma:

$$PD_{rotor} = Kr(1 + Tds) = K(s + b)$$

El PD los parámetros del controlador se muestran en la tabla:

	Kr	Td
rotación	0.5	1.096

Tabla 7 Parámetros control rotor

La función de transferencia del mismo es:

$$PD_{rotor} = 0.45(1 + 1.096s)$$

4 DESARROLLO DE LA APLICACIÓN.

Una vez se dispone de los reguladores necesarios para el control de la maqueta de simulación y de los preparativos para el desarrollo de la aplicación (*PID.vi*, rectas de los sensores, linalización y eliminación de zona muerta) se ha procedido al desarrollo de la aplicación que cumple con los objetivos marcados para la misma:

- Controlar el movimiento de rotación y cabeceo de la maqueta de simulación.
- Ofrecer cuatro modos de control: manual, automático, describir un perfil y control mediante el joystick.
- El control del joystick debe poder realizarse desde otro ordenador, o desde el mismo.

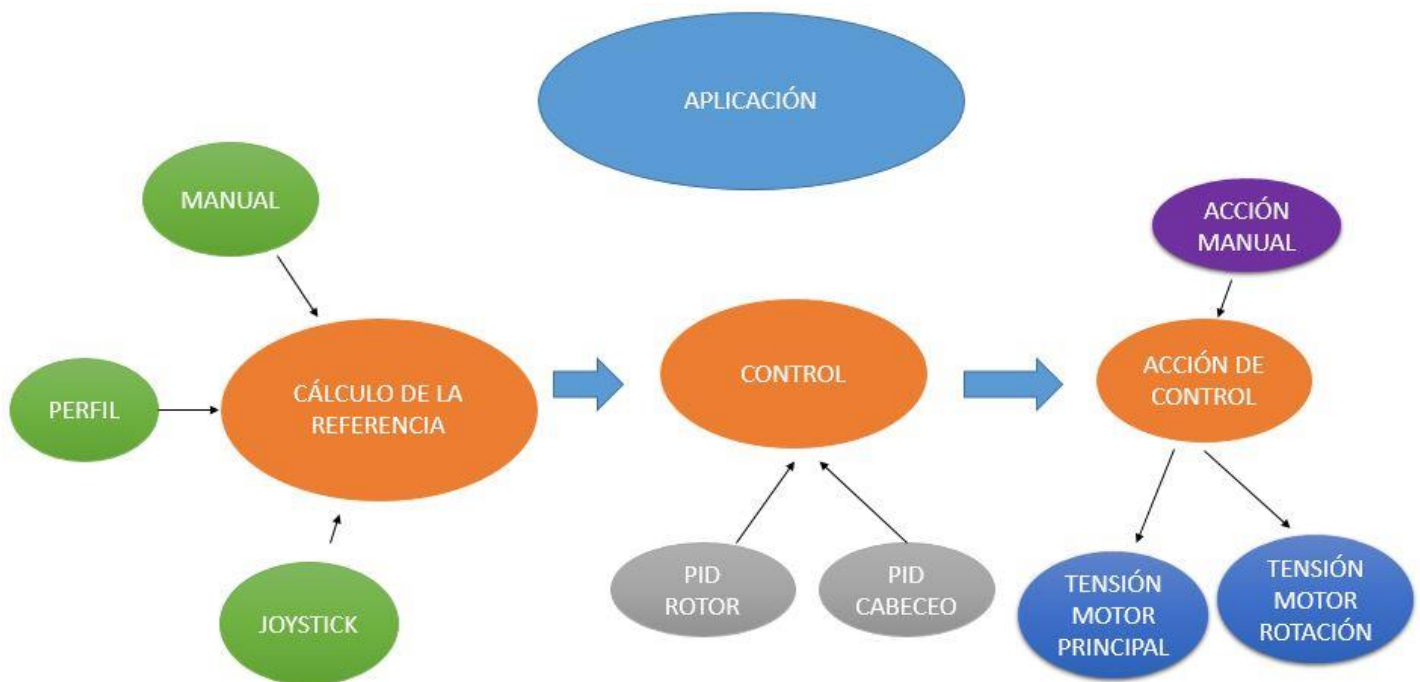


Figura 23 Diagramas de bloques de la aplicación.

El funcionamiento de la aplicación se resume en la Figura 23, donde se muestra un diagrama de bloques, simplificando el funcionamiento de la aplicación.

4.1 BUCLES DE ENCENDIDO Y APAGADO DE LA MAQUETA.

Se ha desarrollado un bucle de encendido y otro de apagado de la maqueta, ya que tras varios ensayos se ha observado que sin estos la maqueta se ponía en marcha de manera muy brusca y no se apagaba a no ser que el último valor de la tensión en los motores fuera cero.

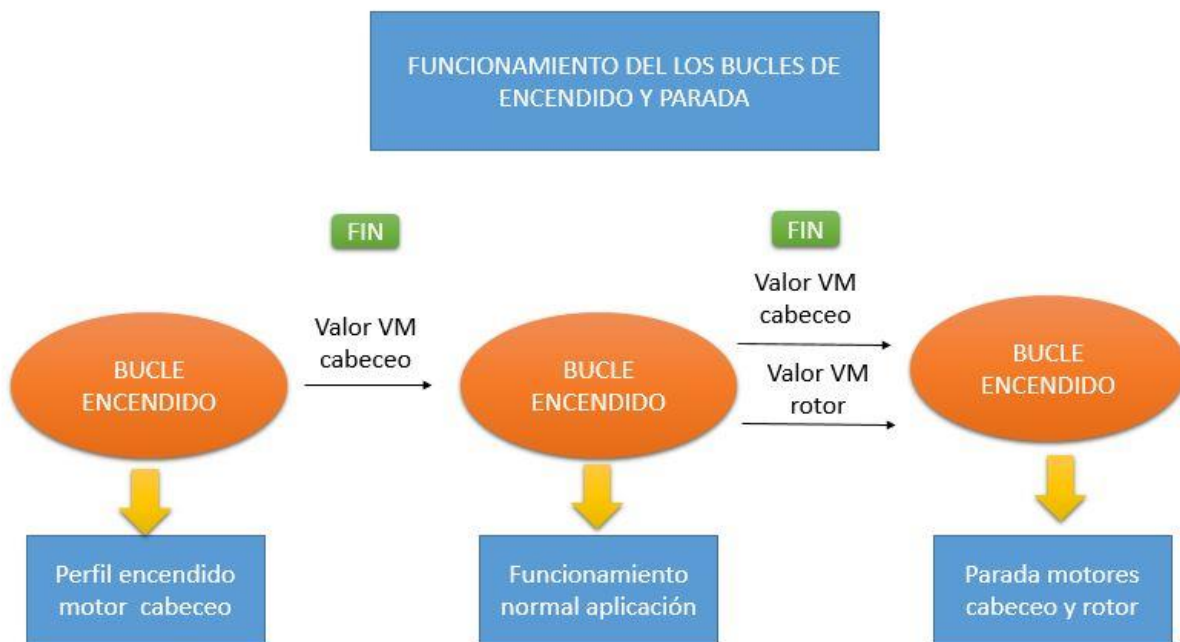


Figura 24 Diagrama de bloques de los bucles de encendido y parada.

En la Figura 24 se puede ver el diagrama de bloques del funcionamiento de los bucles de encendido y apagado de la maqueta de simulación. Es necesario pasar el último valor de la tensión de los motores (VM) al bucle siguiente. Se puede ver cómo, una vez puesta en marcha la maqueta, el bucle de encendido pasa el último valor de VM del cabeceo al bucle principal, y como el bucle principal pasa el valor de las VMs al bucle de parada.

4.1.1 Bucle de encendido.

El bucle de encendido se ejecuta antes del bucle principal, este realiza un perfil en la tensión del motor de cabeceo hasta llevarlo a la posición de equilibrio. El controlador PID no se encuentra en funcionamiento. Se ha implementado un perfil que realiza una rampa desde cero hasta la tensión en el punto de equilibrio, que en el caso de la maqueta ensayada es 3.7V. Una vez adquiere esta tensión, realiza una recta de valor constante en 3.7V de cuatro segundos de duración. Una vez finaliza el perfil, se para el bucle de inicio y pasa a ejecutarse el principal.

El motor de rotación no es necesario ponerlo en funcionamiento previamente, ya que éste parte de la posición de equilibrio sin la puesta en marcha del mismo y, al ser un sistema con integrador, no se podría poner en funcionamiento con un perfil sin un PID para controlar el movimiento del mismo.

4.1.2 Bucle de apagado.

El bucle de parada realiza dos perfiles: uno a partir del último valor en la tensión del motor de cabeceo (VM cabeceo) hasta cero, siguiendo una rampa que dura cuatro segundos, más una recta en cero que dura otros cuatro segundos. El otro perfil desarrolla el mismo perfil con el motor de rotación, solo que éste es a partir del último valor de VC del rotor. En la Figura 25 se ve el aspecto que tiene este bucle en labview.

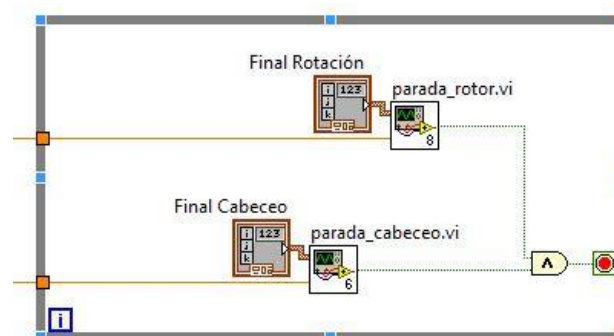


Figura 25 Bucle de parada código de LabVIEW.

4.2 PLANIFICACIÓN DE LA GANANCIA.

Se han calculado los cinco modelos que se utilizarán para la planificación de la ganancia. Los parámetros de los cinco reguladores se pasan a *interpolador.vi*, que es el encargado de interpolar los parámetros de los reguladores PID y proporcionar los parámetros más adecuados en cada instante de ejecución de la aplicación.



Figura 26 Diagrama de bloques del interpolador de los parámetros del PID.

En la Figura 26, vemos el diagrama de bloques del planificador de la ganancia, éste interpolará linealmente los parámetros del PID entre los pasados y proporcionará los valores del PID adecuado en cada punto de funcionamiento.

El usuario puede modificar estos valores en la pestaña de configuración de la interface de control de la aplicación, en el manual del usuario se pueden encontrar más detalles. En cuanto al funcionamiento de *interpolador.vi*, éste se puede encontrar detallado en el anexo del manual del programador.



Figura 27 Interpolate.vi

4.3 PERFIL DE LA REFERENCIA.

En el modo de funcionamiento de Perfil, la aplicación hace que el helicóptero realice unos movimientos predeterminados siguiendo unos perfiles en las referencias de rotación e inclinación que el usuario ha definido en el panel de control.

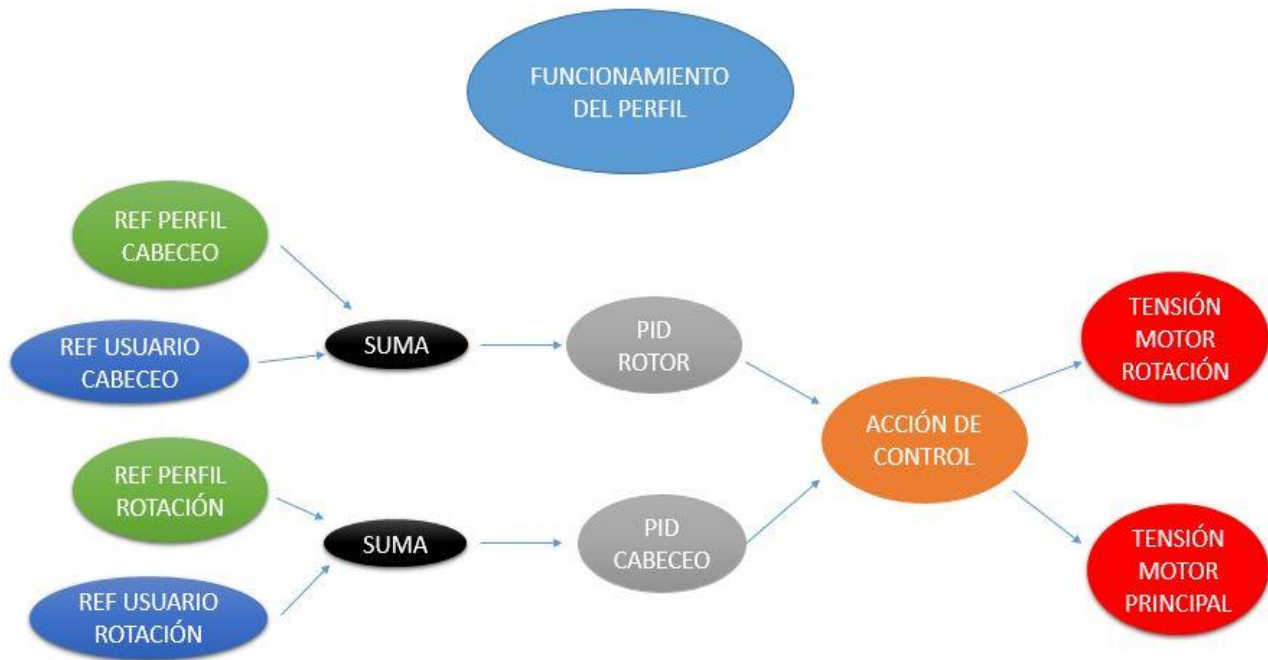


Figura 28 Diagrama de bloques del funcionamiento del perfil

En la Figura 28 podemos ver el diagrama de bloques del funcionamiento del perfil, éste suma la referencia correspondiente del perfil, a la referencia introducida por el usuario, por lo que permite realizar perfiles en distintos puntos de equilibrio. Si el punto de equilibrio está cerca de los límites de funcionamiento no hay problema, ya que la referencia satura una vez alcanzado el límite.

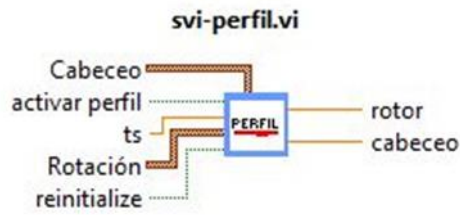


Figura 29 svi-perfil.vi

En la Figura 29 se ve el .vi que se ha programado para crear el perfil en las referencias del helicóptero. Este se activará cuando se active el modo de funcionamiento Perfil. Una vez el perfil se ha completado, si se desea realizar de nuevo es necesario reiniciar el .vi.

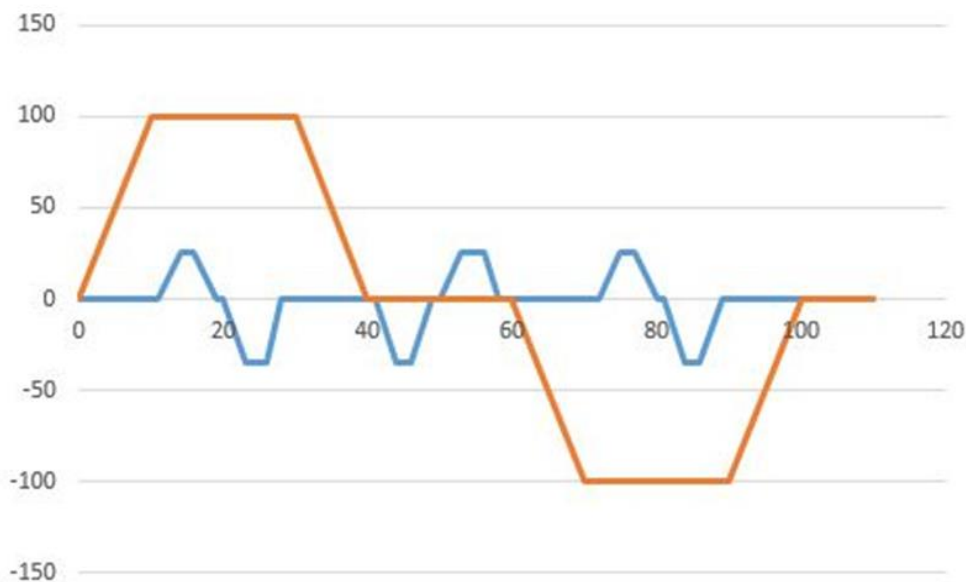


Figura 30 Perfil predefinido en la aplicación.

En la Figura 30 se puede ver el perfil que se ha definido por defecto en la aplicación, en el podemos ver la referencia de rotación en rojo y la referencia de la inclinación del helicóptero en azul. Los movimientos que se han predefinido son independientes, cuando uno se mueve el otro se queda quieto; la duración del perfil es de unos dos minutos aproximadamente.

4.4 CONFIGURACIÓN DEL JOYSTICK

Se dispone de un Joystick para el funcionamiento el control del helicóptero, este se puede ver en la Figura 31.



Figura 31 Joystick empleado en el control.

El joystick dispone de más funciones de las que se van a emplear, solamente se usarán la inclinación del “eje Y” para la inclinación del helicóptero y la rotación del “eje Z” para la rotación del helicóptero. Se han utilizado cuatro pulsadores para cambiar de modo de funcionamiento desde el joystick.

4.4.1 Preparación.

Para el uso del joystick se ha necesitado la instalación de los controladores en el equipo, para el uso de los sensores se han calculado las rectas de conversión del valor proporcionado por el joystick a grados de inclinación o de rotación, dependiendo del caso. El cálculo ha sido inmediato ya que el sensor da un valor simétrico de ± 3200 en todo el rango de movimiento para los dos sensores empleados, por lo que al ser los rangos de funcionamiento del helicóptero simétricos, basta con realizar la división de los límites para obtener la recta de conversión, se puede ver en la Figura 32.

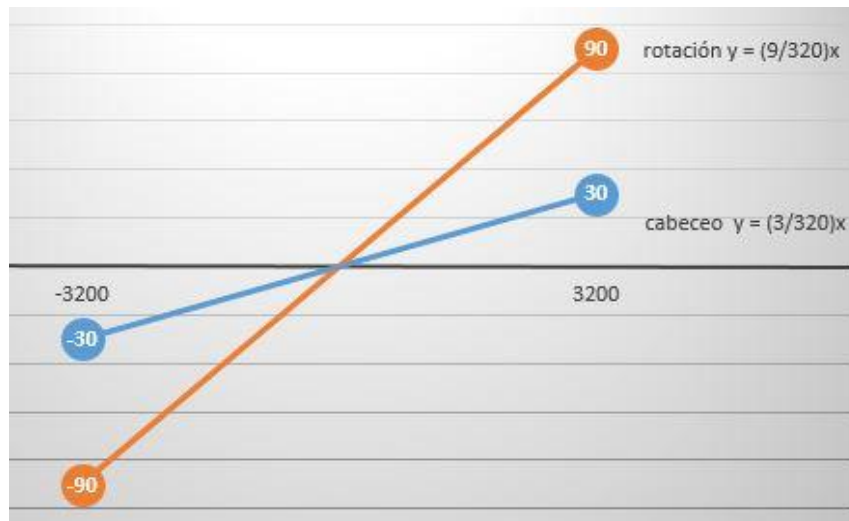


Figura 32 Rectas de conversión de los sensores del joystick.

Para el uso de los pulsadores ha sido necesario el diseño de un .vi que convierte los pulsadores en interruptores y solamente permite la activación de una salida, para que solamente esté un modo de funcionamiento activado en cada instante. En la Figura 33 se puede ver este .vi, los pulsadores empleados del joystick son el 7, 8, 11 y el 12.



Figura 33 Lógica de control del joystick.vi

4.4.2 Modo de funcionamiento joystick.

En el uso del joystick se han contemplado dos posibilidades: que se encuentre en el mismo equipo en el que se realiza el control del helicóptero, o que se encuentre en otro equipo conectado a la misma red.

En el primer caso se ejecutará un bucle de lectura del joystick de forma paralela al bucle principal y pasará los datos mediante variables locales.

En el segundo caso, el mismo bucle de lectura que en el caso anterior se ejecutará en otro equipo, en el cual pasará los datos a un bucle de comunicación TCP que hará el papel de servidor y transmitirá los datos por Ethernet. En el equipo principal se encontrará un bucle de comunicación TCP que hará el papel de cliente y leerá los datos del servidor.

El cambio de funcionamiento se elegirá antes del inicio de la aplicación, y una vez seleccionado un modo de funcionamiento no se podrá cambiar a no ser que se reinicie la aplicación, esto es debido a que para el funcionamiento de comunicación TCP es necesario que el bucle comience a la vez que el principal.

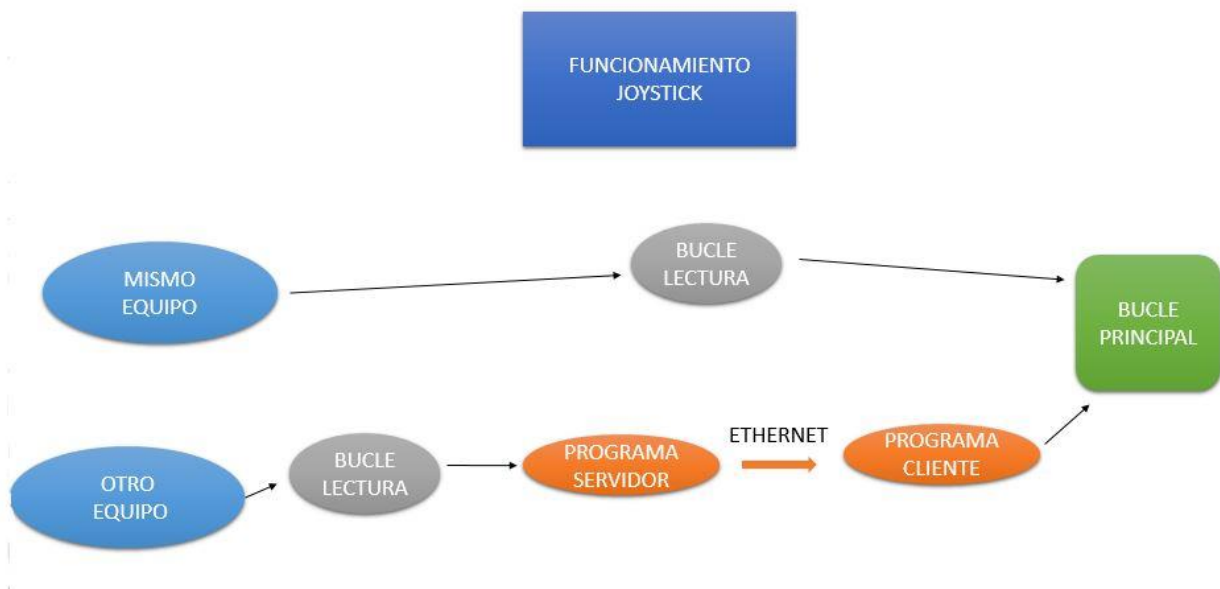


Figura 34 Funcionamiento joystick.

5 APLICACIÓN FINAL.

La aplicación final que se ha desarrollado, y que se entrega junto con el presente proyecto, engloba todo lo realizado en el presente proyecto. Lleva predefinidos los valores de los reguladores desarrollados y los valores en los perfiles de encendido, apagado, perfil de la referencia, etc.

Se ha añadido una parte lógica de control que permite cambiar de modo de funcionamiento y modo de control joystick o mediante el panel de control.

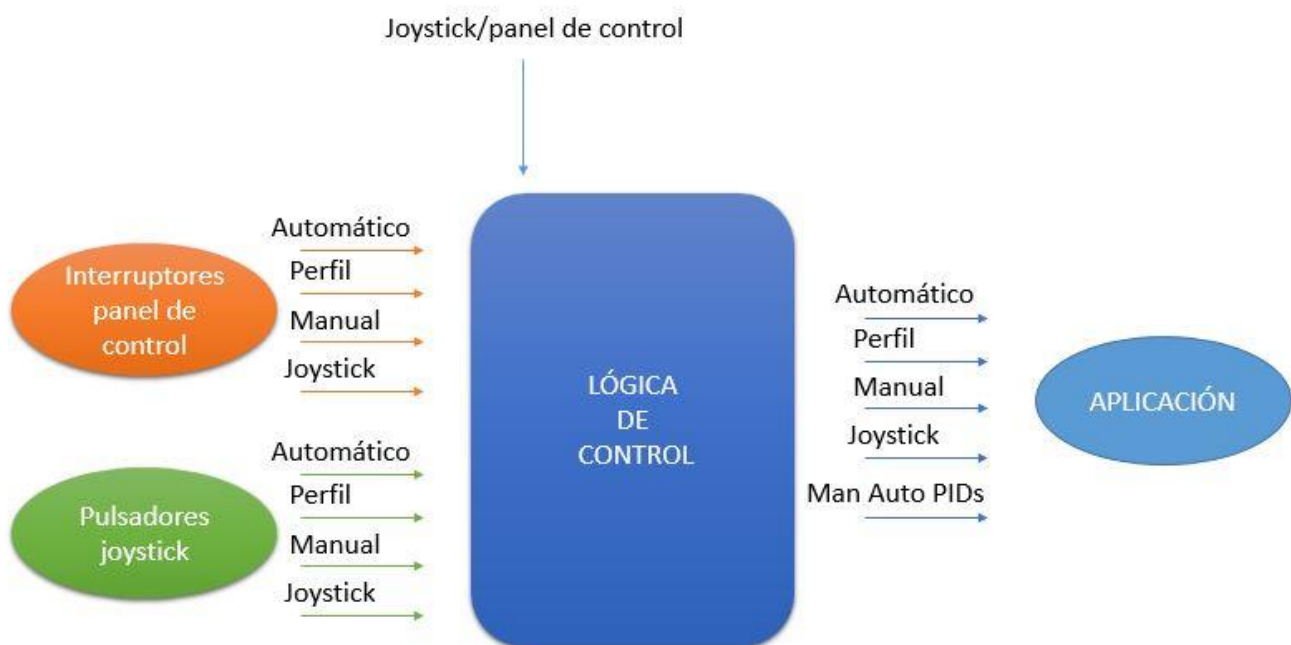


Figura 35 Lógica de control.

El diagrama de bloques que se muestra en la Figura 35 muestra la lógica que selecciona la entrada con la que se desea controlar la maqueta, las salidas de ésta son los cuatro modos de funcionamiento y la señal que controla la activación de los PIDs, que estarán activados en todos los modos de control excepto en el manual.



6 FUTURAS AMPLIACIONES.

Una vez finalizado el presente proyecto, alcanzados los objetivos propuestos al comienzo, se presentan algunas mejoras que se implementarán en el futuro:

- Planificación de la ganancia en tres dimensiones en el motor de cabeceo (rotor principal) del helicóptero, de esa manera el controlador no dependería solamente del ángulo de inclinación como en la aplicación desarrollada, sino que dependería también del ángulo de rotación del helicóptero. Se ha observado que el regulador no se comporta igual en la zona de equilibrio (0° de rotación) que en la zonas límites del rango de funcionamiento ($\pm 90^\circ$ de rotación), por lo que se ha pensado en dividir esta zona en cuatro zonas de funcionamiento e interpolar primero los reguladores en función del ángulo de rotación y, posteriormente, en función del ángulo de inclinación. Esto queda fuera del presente proyecto por limitaciones de tiempo, ya que supondría el desarrollo de cuatro veces los reguladores diseñados para el modelo de inclinación.
- Mediante comunicación TCP, con el joystick poder controlar dos o más helicópteros a la vez, o ser capaz de seleccionar uno en concreto. Se utilizarán los pulsadores del joystick que no se han empleado en el control de un solo helicóptero. Este punto ha quedado fuera del presente proyecto por estar fuera de las limitaciones del mismo.
- Control mediante una Tablet de la posición del helicóptero. LabVIEW permite el control y monitorización de proyectos de forma remota mediante comunicación inalámbrica, en este punto se implementaría un módulo de la aplicación ejecutable por una Tablet para el control del helicóptero. Este punto queda fuera del presente proyecto por limitaciones de tiempo y de recursos disponibles.
- Otra posible mejora que se ha propuesto para mejorar el control en el futuro es la implementación de una estructura de control avanzada que permita eliminar las perturbaciones debidas a la interacción de los movimientos, cuando el helicóptero rota a la vez que realiza un movimiento de cabeceo.

7 CONCLUSIONES.

En el presente proyecto se han cumplido los objetivos fijados al comienzo del mismo, a continuación se detalla cómo se han cumplido los mismos y las conclusiones más relevantes:

- identificado correctamente el modelo de cabeceo del helicóptero y desarrollado un control eficaz del mismo. La solución adoptada para el regulador es la cancelación de los polos complejos, dinámica dominante del proceso. Para mejorar este control se implementa la planificación de ganancia, que permite ajustar los parámetros del regulador en función de la inclinación del helicóptero, para así ajustar mejor los parámetros a cada posición del mismo. Cumpliendo los requisitos impuestos para el regulador: Eliminando prácticamente la sobreoscilación del sistema. Cumpliendo el tiempo de establecimiento menor de cuatro segundos. Error de posición nulo en régimen permanente.-
- El control de rotación, identificado un modelo y diseñada un regulador que cumple los requisitos impuestos en los objetivos: Error de posición nulo y tiempo de establecimiento menor de 4 segundos.
- Desarrollados los PIDs para el control de los motores, un PID para cada motor, implementadas las ecuaciones discretas del algoritmo PID ISA, incluida la acción antiwindup para que la acción integral no acumule el error cuando la variable manipulada sature.
- Desarrollada la aplicación en LabVIEW capaz de controlar el funcionamiento de la maqueta. Ésta incorpora el encendido y apagado de la maqueta, condición que no estaba en los objetivos pero que, tras tratar con la maqueta, se realizar un encendido y apagado controlado de la misma.



- La aplicación incorpora los cuatro modos de funcionamiento:
 - Modo Manual, en el cual es el usuario el que introduce el valor de tensión que llega a los motores.
 - Modo Automático en el cual el usuario introduce la referencia a la que quiere que llegue el helicóptero y los controladores PID llevan el helicóptero a esa posición.
 - Perfil en la referencia, en la cual la maqueta describe un perfil en las tres dimensiones que define el usuario.
 - Modo Joystick en el cual el helicóptero sigue los movimientos del joystick.
- También implementada la opción de elegir si se quiere realizar el control con el joystick desde el mismo ordenador o se desea controlarlo desde otro equipo.
- Comunicación TCP, de tal modo que el helicóptero se puede controlar desde otro equipo conectado a la misma red Ethernet, en la cual se encuentra conectado el equipo de control de la maqueta de simulación.



8 BIBLIOGRAFÍA.

Instruments, N. (s.f.). *Software de Desarrollo de Sistemas NI LabVIEW*. Obtenido de
<http://www.ni.com/labview/esa/>

J. L: Guzmán, R. c. (1996). *Sistemas de Control Automatico*. Prentice Hall.

K. J. Aström, T. (2009). *Control PID Avanzado*. Prentice hall.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 2: PRESUPUESTO

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



INDÍCE DE CONTENIDO

1 INTRODUCCIÓN.....	4
2 PRESUPUESTO DE EJECUCIÓN MATERIAL	5
2.1 CÁLCULO DE LOS RECURSOS HUMANOS.	5
2.1.1 Personal directamente implicado.	5
2.1.1.1 Cálculo del coste por hora.....	6
2.1.1.2 Coste total personal directamente implicado.	6
2.1.2 Personal indirectamente implicado.	7
2.1.2.1 Cálculo del precio por hora.	7
2.1.2.2 Coste total personal indirectamente implicado.	8
2.1.3 Gasto recursos humanos.....	8
2.2 AMORTIZACIÓN DE SOFTWARE.	8
2.2.1 Datos para la amortización del software.	9
2.2.2 Gastos en software.....	9
2.3 GASTOS EN HARHWARE.	9
2.3.1 Compra de equipo.....	10
2.3.2 Amortización de equipos.	10
2.3.4 Gastos totales en hardware.	10
2.4 COSTES ADMINISTRATIVOS.....	10
2.4.1 Cálculo gastos administrativos.....	11
2.5 TOTAL PRESUPUESTO DE EJECUCIÓN MATERIAL.....	11
3 PRESUPUESTO DE EJECUCIÓN POR CONTRATA.	12
4 PRESUPUESTO BASE DE LICITACIÓN.	12



INDÍCE DE TABLAS

Tabla 1 cálculo del coste hora para el graduado en tecnologías industriales	6
Tabla 2 cálculo del precio por hora del técnico de mantenimiento	7
Tabla 3 Gastos de amortización del software	9
Tabla 4 Hardware de control.....	10
Tabla 5 Gastos de amortización del hardware.....	10
Tabla 6 Presupuesto ejecución material.....	11
Tabla 7 Presupuesto de ejecución por contrata.	12
Tabla 8 Presupuesto base de licitación.....	12



1 INTRODUCCIÓN.

En el presente documento se detallará el presupuesto de la aplicación y del equipo de control. Al tratarse de una aplicación informática de control no tendrá el formato de un presupuesto clásico, como puede ser el presupuesto de un proyecto para la construcción de una nave industrial.

Para estimar el presupuesto, se ha supuesto que el proyecto ha sido encargado a una empresa de desarrollo de software industrial. A partir de este supuesto se va a realizar el estudio económico pertinente.

El punto de partida será que un cliente nos encarga el desarrollo de una aplicación de control para una maqueta de simulación del movimiento de rotación y cabeceo de un helicóptero, nosotros le entregaremos la aplicación de control y los equipos necesarios para realizar el control de la misma. No se contempla la adquisición de la maqueta de simulación, ya que es suministrada por el cliente, sin embargo se presupuesta el equipo necesario para realizar el control.

Procedemos a detallar el cálculo del presupuesto, en éste se incluye el coste de los Recursos Humanos, costes de los materiales empleados en el proyecto y los costes directos e indirectos para dar lugar al presupuesto de ejecución material. A éste se añadirán el beneficio industrial y los gastos generales dando lugar al presupuesto de contrata. Finalmente, tras añadir el IVA tendremos el presupuesto de licitación del proyecto.

El presupuesto de ejecución material se dividirá en cuatro unidades, siendo éstas:

- Recursos humanos.
- Amortización del software.
- Gastos en hardware.
- Gastos administrativos.

En cada unidad se incluirán los costes directos oportunos, sin embargo los costes indirectos no se incluirán puesto que se consideran dentro de los gastos generales.



2 PRESUPUESTO DE EJECUCIÓN MATERIAL

Para el cálculo del presupuesto se ha realizado un estudio económico que permite valorar el coste del proyecto, teniendo en cuenta todos los recursos implicados en la ejecución del mismo.

2.1 CÁLCULO DE LOS RECURSOS HUMANOS.

En este apartado se ha incluido tanto el personal directamente implicado en el proyecto, un graduado en tecnologías industriales, como los técnicos de mantenimiento de los equipos utilizados. El personal administrativo estará reflejado más adelante en los costes administrativos.

2.1.1 Personal directamente implicado.

El personal directamente implicado en el desarrollo de este proyecto ha sido un graduado en tecnologías industriales, que se dedica exclusivamente a este proyecto a lo largo de todo un mes.



2.1.1.1 Cálculo del coste por hora

El cálculo del coste por hora del graduado en tecnologías industriales se detalla en la Tabla 1.

CONCEPTOS	TOTAL
Salario bruto año	21.000
Gratificaciones extra (junio, Navidad y vacaciones)	5.250
Seguridad social (Base cotización, contingentes comunes, accidentes trabajo, desempleo, fondo garantía social y formación profesional)	6.300
Varios (horas extra, dietas,...)	1.250
TOTAL ANUAL	33.800
A FACTURAR:	
Por Jornada (220 días laborables en un año)	153,64
Por hora (8 horas laborables)	19,21

Tabla 1 cálculo del coste hora para el graduado en tecnologías industriales

2.1.1.2 Coste total personal directamente implicado.

El graduado en tecnologías industriales está un mes trabajando en el desarrollo de la aplicación, haciendo en total unas 176 horas hábiles. La facturación del graduado es de **176x19.21=3.380€**

Total gasto mano de obra directa =3.380€



2.1.2 Personal indirectamente implicado.

En este punto incluiremos los técnicos de mantenimiento del equipo empleado en el transcurso del proyecto.

Este personal está formado por dos técnicos, uno encargado de la instalación y actualización del software de los equipos informáticos y otro encargado del mantenimiento del hardware utilizado.

2.1.2.1 Cálculo del precio por hora.

Los dos técnicos cuestan lo mismo, por lo que se facturan las horas de los dos técnicos en un mismo concepto que es técnico de mantenimiento (detallado en la tabla 2).

CONCEPTOS	TOTAL
Salario bruto año	15.000
Gratificaciones extra (junio, Navidad y vacaciones)	3.750
Seguridad social (Base cotización, contingentes comunes, accidentes trabajo, desempleo, fondo garantía social y formación profesional)	4.504
Varios (horas extra, dietas,...)	950
TOTAL ANUAL	24.204
A FACTURAR:	
Por Jornada (220 días laborables en un año)	110
Por hora (8 horas laborables)	13,75

Tabla 2 cálculo del precio por hora del técnico de mantenimiento

Las horas de mantenimiento se reducen a dos días de preparación de los equipos (entre actualización de software e instalación del hardware) y cinco horas de mantenimiento a la semana durante la duración del mismo.



2.1.2.2 Coste total personal indirectamente implicado.

Dos días más cuatro semanas de mantenimiento hacen un total de 36 horas.

Los técnicos de mantenimiento dedican treinta y seis horas, por lo que la facturación de estos será de **36x13,75=495€**.

Total gasto mano de obra indirecta =495€

2.1.3 Gasto recursos humanos.

Los costes totales de personal, sin tener en cuenta al personal administrativo, es el coste del graduado en tecnologías industriales más el coste de los técnicos de mantenimiento.

Total gasto mano de obra =3.875€

2.2 AMORTIZACIÓN DE SOFTWARE.

Para estimar el coste del software utilizado, tendremos en cuenta que será la suma de la amortización de los programas empleados en el proyecto, estos costes variarán para cada programa, ya que no depende únicamente del precio de la licencia, sino que también depende del número de proyectos realizados cada año con dichos programas.



2.2.1 Datos para la amortización del software.

En la tabla siguiente se detallan los precios y el número de proyectos que se estima que se realizarán en un año.

Software	Precio € licencia/año	Nº de proyectos/ año	Valor a amortizar por proyecto €
LabVIEW	1080	6	180
Office 365	70	10	7
Matlab	1000 ¹	10	100
Windows 8	100 ²	10	10

Tabla 3 Gastos de amortización del software

2.2.2 Gastos en software.

La suma de los costes de amortización de todos los programas.

Total gasto en software 297€

2.3 GASTOS EN HARHWARE.

Para el desarrollo del proyecto se ha empleado un ordenador personal para la programación y procesado de los datos. Por otro lado como el cliente nos entrega la maqueta de simulación sin ningún equipo de control se adquiere el equipo de control necesario. Éste se entregara al cliente junto con la maqueta del helicóptero.

¹ La licencia de Matlab cuesta 2000€, pero en la empresa se actualiza cada dos años.

² El equipo se actualizó a Windows 8 por 300€, se renovará con el equipo a los 3 años.

2.3.1 Compra de equipo.

Equipo necesario para realizar el control de la maqueta.

Hardware	Precio €
PC	500
Tarjeta adquisición de datos	250
Joystick	30

Tabla 4 Hardware de control.

2.3.2 Amortización de equipos.

La amortización del equipo de programación se detalla en la Tabla 5.

Hardware	Precio € licencia/año	Tiempo de amortización (años)	Nº de proyectos/ año	Valor a amortizar por proyecto €
Ordenador programación	750	3 ³	10	25

Tabla 5 Gastos de amortización del hardware

2.3.4 Gastos totales en hardware.

El montante total al que asciende el gasto en hardware es:

Total coste amortización hardware 1055€.

2.4 COSTES ADMINISTRATIVOS.

En este apartado se reflejarán los costes de administración derivados de la tramitación y adjudicación del proyecto por parte de la empresa.

El cálculo de los costes derivados de la administración del proyecto por parte de la empresa se realiza a estima y se supone aproximadamente un 20% del coste de mano de obra.

³ Normalmente son 5 años, pero al tratarse de una empresa de desarrollo de software se actualizan los equipos antes.



2.4.1 Cálculo gastos administrativos

En este apartado se incluirán los gastos de tramitación del presupuesto y los costes administrativos derivados de la tramitación del proyecto.

Para el cálculo de estos gastos se recurre a presupuestarlos como el 20% de los gastos en personal directo e indirecto.

Gastos totales administrativos $3380 \cdot 0,2 = 676\text{€}$

2.5 TOTAL PRESUPUESTO DE EJECUCIÓN MATERIAL.

El presupuesto del proyecto se calcula sumando los valores obtenidos en los apartados anteriores del estudio económico:

Total recursos humanos	3380
Total amortización software	397
Total amortización hardware	1055
Total administrativo	676
Total	5508

Tabla 6 Presupuesto ejecución material.

El presupuesto de ejecución material asciende a un total de cinco mil quinientos ocho euros.

3 PRESUPUESTO DE EJECUCIÓN POR CONTRATA.

El presupuesto de ejecución por contrata se calcula a partir del presupuesto de ejecución material.

Descripción	rendimiento	Importe €
PRESUPUESTO DE EJECUCIÓN MATERIAL		5508
Gastos Generales	13%	716
Beneficio Industrial	6%	330
TOTAL		6554

Tabla 7 Presupuesto de ejecución por contrata.

El presupuesto de ejecución por contrata asciende a un total de seis mil quinientos cincuenta y cuatro euros.

4 PRESUPUESTO BASE DE LICITACIÓN.

Por último lugar, para obtener el presupuesto final del proyecto se ha de añadir el I.V.A, que en el momento de realización de este presupuesto está en un 21%.

Descripción	rendimiento	Importe €
PRESUPUESTO DE EJECUCIÓN POR CONTRATA		6554
I.V.A	21%	1376
TOTAL		7930

Tabla 8 Presupuesto base de licitación.

El presupuesto base de licitación asciende a un montante total de siete mil novecientos treinta euros.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 3: PLIEGO DE CONDICIONES

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



ÍNDICE

1	PLIEGO DE CONDICIONES GENERALES.....	4
1.1	PLIEGO DE CONDICIONES FACULTATIVAS.....	4
1.1.1	Programación de trabajos.....	4
1.1.2	Ejecución defectuosa y modificaciones.....	5
1.1.3	Recepción definitiva.....	5
1.2	PLIEGO DE CONDICIONES ECONÓMICAS.....	6
1.2.1	Fianzas.....	6
1.2.2	Composición de precios.....	7
1.1.3	Mejoras.....	7
1.1.4	Revisión de precios.....	7
2	PLIEGO DE CONDICIONES PARTICULARES.....	8
2.1	CONDICIONES DE FUNCIONAMIENTO DEL SISTEMA.....	8
2.1.1	Limitación por equipo hardware.....	8
2.1.2	Limitaciones bajo sistemas operativos.....	9
2.1.3	Control del sistema.....	9
2.2	CONDICIONES DE INSTALACIÓN.....	9
2.3	CONDICIONES DE CARA AL PROGRAMA.....	10
2.4	CONDICIONES DE MANTENIMIENTO.....	10
2.4.1	Condiciones de mantenimiento del programa informático.....	10
2.4.2	Condiciones de mantenimiento de la maqueta de simulación.....	11
2.5	CONDICIONES DE GARANTÍA.....	11



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.

La contrata se compromete a realizar el proyecto dentro de los plazos acordados con el cliente, después de una revisión exhaustiva del mismo. La duración del proyecto es de un mes y medio una vez entregada la maqueta por parte del cliente a la contrata.



1.1.2 Ejecución defectuosa y modificaciones.

El cliente tiene la responsabilidad de revisar el funcionamiento de la aplicación suministrada y comunicar los aspectos que, a juicio de él, no considere correctos, extendiéndose dicha responsabilidad a aquellos defectos no detectados pero existentes.

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

Las tarifas a aplicar, en caso de modificaciones, serán las mismas que para el resto del proyecto, siguiendo lo establecido entre la empresa y el cliente.

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 entrega por parte de la empresa de la aplicación desarrollada junto con la maqueta de simulación y equipo necesario para el control por computador de la misma.



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 30% del valor total del presupuesto del proyecto.

En caso de demora en el abono de la fianza, esta se incrementará en un 3% semanal.

El resto del valor total del proyecto será abonado una vez realizada la instalación y comprobación del correcto funcionamiento de la aplicación.

Una vez transcurridos mes y medio tras la entrega de la maqueta, periodo fijado por el cliente y la empresa, se ha acordado que la empresa desarrolladora abone una cantidad del 0.5% 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 de la aplicación informática, durante un período de un año después de la entrega. 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 en la empresa para un graduado en tecnologías industriales realizando tareas de programación y para los técnicos de mantenimiento.

1.1.3 Mejoras.

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

1.1.4 Revisión de precios.

Como el tiempo que puede transcurrir en la aprobación 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 Ingenieros Industriales de Valencia, se procederá a una revisión de precios en el acto de entrega del mismo.



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 una aplicación informática desarrollada en LabVIEW, 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 de la aplicación 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 4/M o equivalente.
- Memoria RAM mínima de 1 GB.
- Conexión USB 2.0.
- Espacio libre en el Disco Duro de 100 MB.
- Tarjeta de Adquisición de datos, compatible con LabVIEW.
- Periféricos, Teclado y Ratón.
- Sistema operativo Windows 7.



Estos requisitos son los mínimos requeridos para el correcto funcionamiento de la aplicación incluyendo las operaciones de simulación que realiza, por tanto serán 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 vista o superior.

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 la aplicación de forma cómoda, simplemente debe saber desenvolverse bajo el entorno gráfico Windows, ya que la aplicación ha 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.

De esta manera el cliente no tiene por qué saber programar ni tener conocimientos de LabVIEW

2.2 CONDICIONES DE INSTALACIÓN.

La aplicación puede ser instalada fácilmente por el cliente, sin apenas tener unos vanos conocimientos del entorno gráfico Windows, ésta será inmediata ya que se entrega el ejecutable .exe.



2.3 CONDICIONES DE CARA A LA APLICACIÓN.

La adjudicación del proyecto da derecho, exclusivamente, a una licencia del mismo.

Si el cliente precisa de otra licencia de la aplicación, a parte de la suministrada, por razones ajenas al funcionamiento de la misma, 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 costes 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 de la aplicación anula el derecho de garantía posventa de la copia.

2.4 CONDICIONES DE MANTENIMIENTO.

Este apartado incluye el mantenimiento de la aplicación desarrollada en LabVIEW, el mantenimiento de la maqueta de simulación correrá a cargo del cliente.

2.4.1 Condiciones de mantenimiento de la aplicación informática.

La aplicación no necesita revisión por parte de ningún técnico, salvo en el caso de modificaciones en la configuración del hardware o del software.

Queda terminantemente prohibida la alteración de la misma y, si se comprobara que está ha sido modificada, 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 de la maqueta de simulación.

La contrata desarrolladora del software se compromete a salvaguardar la integridad de la maqueta entregada por el cliente para el desarrollo de la aplicación de control. Se devolverá al cliente en las mismas condiciones en las que éste la entregó. En caso de no ser así, será compensado económicamente por la empresa en función de los daños sufridos.

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 el año a partir de la fecha de adjudicación del proyecto, éste podrá ser descargado de la wad de la empresa de nuevo.

De todo ello se desprende, por tanto, que el período de vigencia de la garantía es de un año. 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 actualizaciones de la aplicación, 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 la aplicación por parte del cliente, así como su copia, préstamo o instalación no autorizada, anula completamente cualquier condición de garantía expuesta anteriormente.

La empresa no se hace responsable, en ningún caso, de los perjuicios que se puedan desprender de la utilización indebida de la aplicación, 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 IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 4: MANUAL DEL USUARIO

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



ÍNDICE DE CONTENIDO

1 INTRODUCCIÓN.....	4
2 REQUISITOS PARA REALIZAR EL CONTROL.....	4
2.1 HARDWARE.....	4
2.1.1 Ordenador.....	5
2.1.2 Tarjeta de adquisición de datos.....	5
2.2 SOFTWARE.....	5
3 USO DE LA APLICACIÓN.....	6
3.1 PESTAÑA DE CONTROL.....	7
3.2 PESTAÑA DE GRÁFICAS.....	9
3.3 PESTAÑA DE CONFIGURACIÓN.....	10
3.4 PESTAÑA PARÁMETROS PIDs.....	12
4 BIBLIOGRAFIA.....	13



ÍNDICE DE FIGURAS

Figura 1 Pestañas del panel de control.	6
Figura 2 Pestaña de control.....	7
Figura 3 Indicadores del funcionamiento del joystick.	7
Figura 4 Panel de control principal.....	8
Figura 5 Pestaña de gráficas.....	9
Figura 6 Pestaña de configuración.	11
Figura 7 Pestaña parámetros PIDs.	12



1 INTRODUCCIÓN.

En el presente documento se detalla cómo usar la aplicación desarrollada en el proyecto, de manera que un usuario sin conocimientos de control ni de programación en LabVIEW pueda controlar el helicóptero si ha leído este manual.

Así mismo podrá configurar y definir los perfiles de encendido, apagado y el realizado por el helicóptero en el modo perfil. Se podrá modificar los parámetros de los reguladores definidos por defecto, pero esto solo se aconseja a usuarios avanzados en el control por computador.

2 REQUISITOS PARA REALIZAR EL CONTROL.

En este apartado se expondrán los requisitos de hardware y software que debe cumplir el equipo con el que se ejecute la aplicación para realizar el control. En el presupuesto se ha incluido el coste del equipo que se recomienda para este fin.

2.1 HARDWARE.

Para realizar el control de la maqueta con la aplicación desarrollada en este proyecto son necesarios un ordenador, una tarjeta de adquisición de datos y un joystick.



2.1.1 Ordenador.

El equipo necesario para la ejecución de la aplicación debe cumplir las especificaciones **mínimas** necesarias para ejecutar el archivo .exe de la aplicación creado con LabVIEW, a continuación se resumen (National Instruments, s.f.):

- Procesador, Pentium 4 o equivalente.
- RAM: 256 MB.
- Sistema Operativo: Windows Vista (32 bits y 64 bits) o superior.
- Espacio en Disco, 407 MB

Para la ejecución del programa completo es necesario un equipo con LabVIEW instalado. Los requisitos de LabVIEW son superiores a los indicados.

2.1.2 Tarjeta de adquisición de datos.

La tarjeta de adquisición que se recomendada es la tarjeta DAQ-LVIEW PnP PCI-9112 de Adlink (usada en el desarrollo de la aplicación), se recomienda el uso de ésta. En caso de usar otra tarjeta no se garantiza el funcionamiento de la aplicación, ya que ésta se ha configurado para los driver la tarjeta PIC-9112. Dicha tarjeta ha de disponer de dos entradas analógicas y de dos salidas analógicas libres para la conexión del helicóptero.

2.2 SOFTWARE.

Para la utilización del archivo .exe que contiene la aplicación será necesario que el equipo disponga de los drivers de la tarjeta instalados y que posea un sistema operativo Windows vista o superior. XP no se incluye, ya que Microsoft no da soporte a éste sistema.

En el desarrollo de la aplicación se han usado diversos programas, pero no será necesario tener instalados éstos para la ejecución de la aplicación mediante el .exe. Sin embargo, si se dispone del permiso del desarrollador y se desea modificar la aplicación es necesario la instalación de éstos. En el anexo de programación se puede consultar qué programas se han usado y los procedimientos seguidos.

3 USO DE LA APLICACIÓN.

El panel de control de la aplicación se ha dividido en cuatro pestañas. Éstas se han dividido de forma que la navegación por las mismas sea amena y fluida. La interface del usuario es simple, para que con un simple golpe de vista se vean los controles. Como la finalidad de la aplicación es el control de helicóptero y para ello se debe de atender al funcionamiento real del mismo, no se han implementado pantallas dinámicas que requieran la atención del usuario.

Las cuatro pestañas del panel del usuario son: control, graficas, configuración y parámetros PIDs. En la Figura 1, se pueden ver las cuatro pestañas de control.

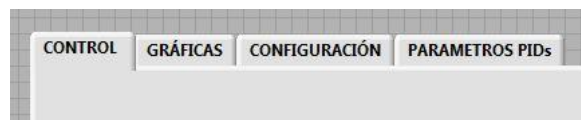


Figura 1 Pestañas del panel de control.

3.1 PESTAÑA DE CONTROL.

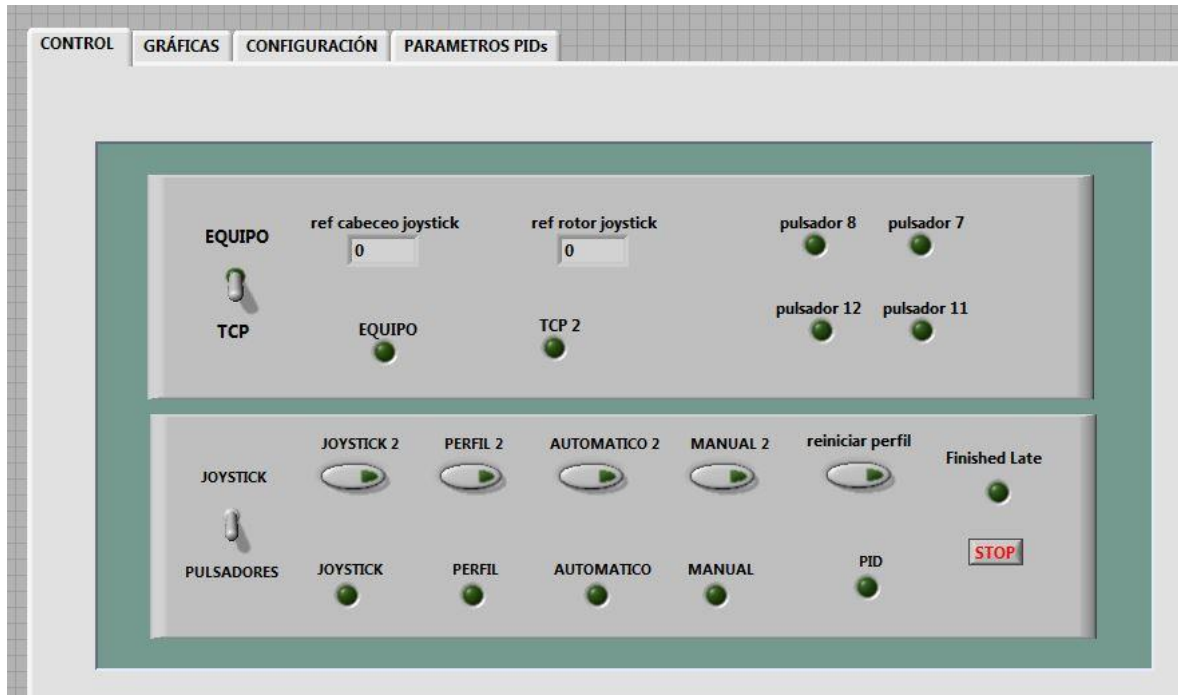


Figura 2 Pestaña de control.

En la pestaña de control se controla el modo de funcionamiento y se selecciona el modo de control mediante el joystick, si este se encuentra en el mismo equipo o se comunica por TCP.

La posición del joystick ha de fijarse antes de ejecutar la aplicación, ya que una vez puesta en marca no se puede cambiar y, si se ha seleccionado la posición errónea, la aplicación no funcionará y dará error al ejecutarse.

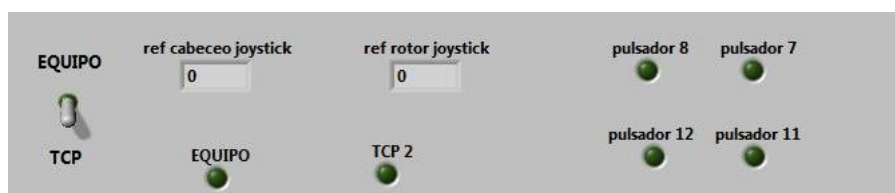


Figura 3 Indicadores del funcionamiento del joystick.

En la Figura 3 se observa el panel de control que permite seleccionar la posición del joystick y los indicadores que muestran el funcionamiento del mismo. En estos indicadores se puede ver si se encuentra en el mismo equipo o si se comunica por TCP, así como el valor de la referencia que éste manda y el pulsador que se ha pulsado en último lugar en el joystick para seleccionar el modo de funcionamiento.

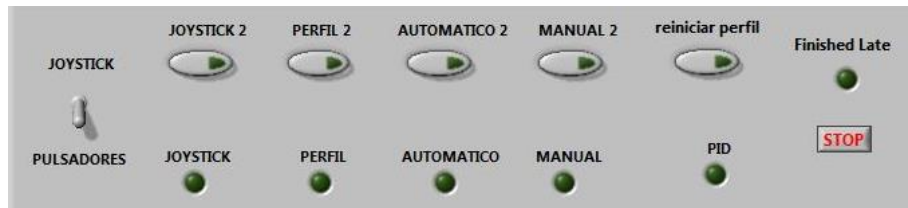


Figura 4 Panel de control principal.

En la Figura 4 se puede ver el panel de control principal de la aplicación, en éste se puede seleccionar el modo de control, pulsadores o joystick. Este modo de control sí se puede cambiar durante la ejecución de la aplicación. Los pulsadores para la selección del modo de funcionamiento también se encuentran en este panel de control, así como los pulsadores de **reiniciar el perfil** y el pulsador de **stop** para la finalización de la aplicación.

Los led muestran el modo de funcionamiento que se encuentra activado y el led PID muestra si los controladores PID están controlando el proceso, o si el control lo realiza el usuario (este led se encontrará encendido para todos los modos de funcionamiento excepto para el modo manual).

3.2 PESTAÑA DE GRÁFICAS.

En la pestaña de gráficas podemos ver la evolución de las variables en tiempo real. También se encuentran los controles de la referencias para el modo de control automático y los controles de la tensión de los motores para el modo de funcionamiento manual.

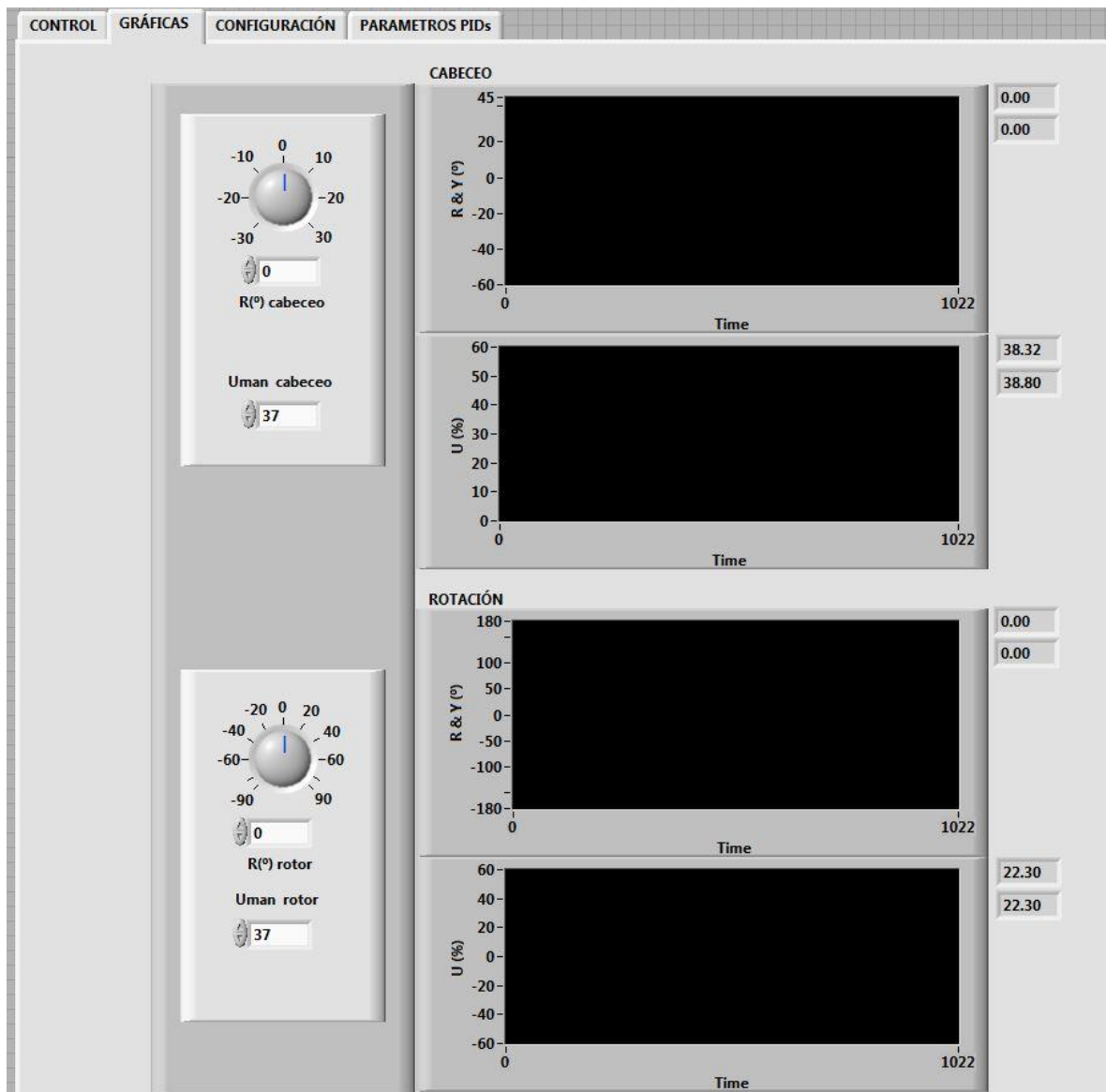


Figura 5 Pestaña de gráficas.



En las dos primeras gráficas se muestran las variables correspondientes al cabeceo del helicóptero. En la primera se ven en rojo los grados de inclinación real y el valor de la referencia introducido en blanco. En la segunda se ve en rojo la tensión que le llega al motor y en blanco el valor de la tensión sin linealizar.

En las dos últimas gráficas se ven los valores correspondientes a las variables del movimiento de rotación del helicóptero. En la primera de las dos se ve en rojo el valor de los grados de rotación real y en blanco el valor de la referencia. En la última se ve la acción de control del motor de rotación, en rojo el valor de la tensión que le llega al motor y en blanco el valor de la tensión antes de eliminar la zona muerta del motor.

3.3 PESTAÑA DE CONFIGURACIÓN.

En la pestaña de configuración se introducen los valores de los perfiles y los datos necesarios para poder realizar la comunicación por TCP. En esta pestaña se configuran los perfiles que se realizan en el encendido y apagado del helicóptero. Se recomienda dejarlos fijados por defecto, pero si el usuario desea modificarlos es posible.

Se puede definir los puntos del perfil que realizará el helicóptero en el modo de funcionamiento perfil. Estos perfiles sí se pueden modificar sin problemas por parte del usuario. Si se desean modificar hay que tener en cuenta que se han de introducir los puntos por los que el usuario desea que pase el helicóptero, debe de definir la posición y el tiempo de dicho punto.

La ampliación realizará una interpolación lineal entre dichos puntos, se pueden realizar escalones fijando dos posiciones con un mismo tiempo. Los puntos han de introducirse ordenados en el tiempo.

En la Figura 6 se puede ver el aspecto de la pantalla de configuración. En primer lugar, se encuentran los perfiles del modo de funcionamiento perfil, los perfiles de los bucles de encendido y apagado se encuentran debajo de éstos. También se encuentra el control del periodo de ejecución de la aplicación, se recomienda no cambiarlo.

Por último, se encuentran los controles para realizar la comunicación por TCP. Se ha de fijar la dirección IP del equipo en el cual se ejecuta la aplicación servidor y el puerto en el cual se han de leer los datos. Estos datos han de ser definidos por el usuario antes de ejecutar el programa con el joystick situado en otro equipo.

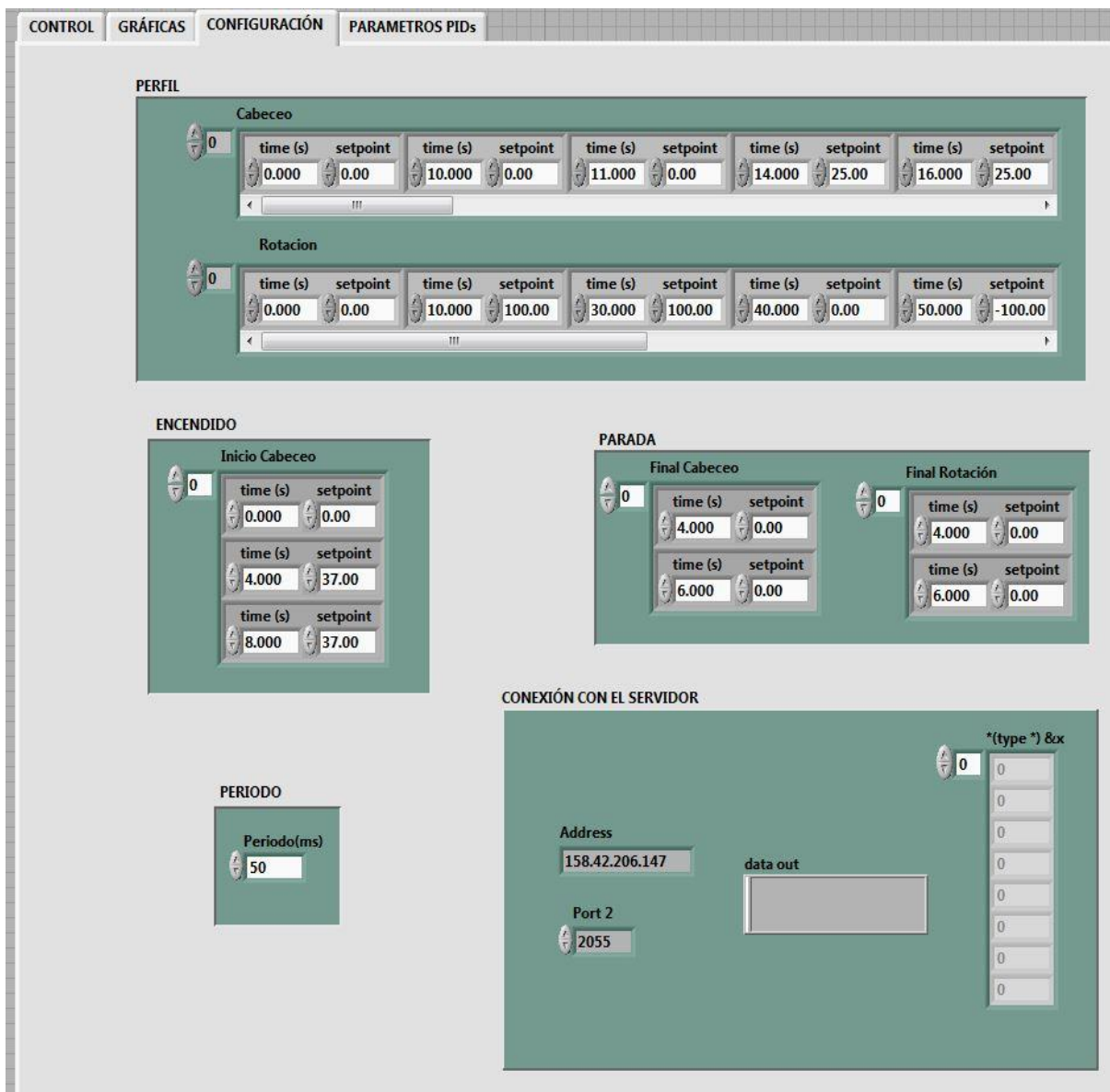


Figura 6 Pestaña de configuración.

3.4 PESTAÑA PARÁMETROS PIDs.

En la Figura 7 vemos la última pestaña del panel de control, en ésta se encuentran los controles para definir el valor de los PIDs. Los valores que se han fijado por defecto son los que se han calculado para la maqueta de simulación ensayada, en el proyecto y nos recomienda que un usuario sin experiencia los modifique.

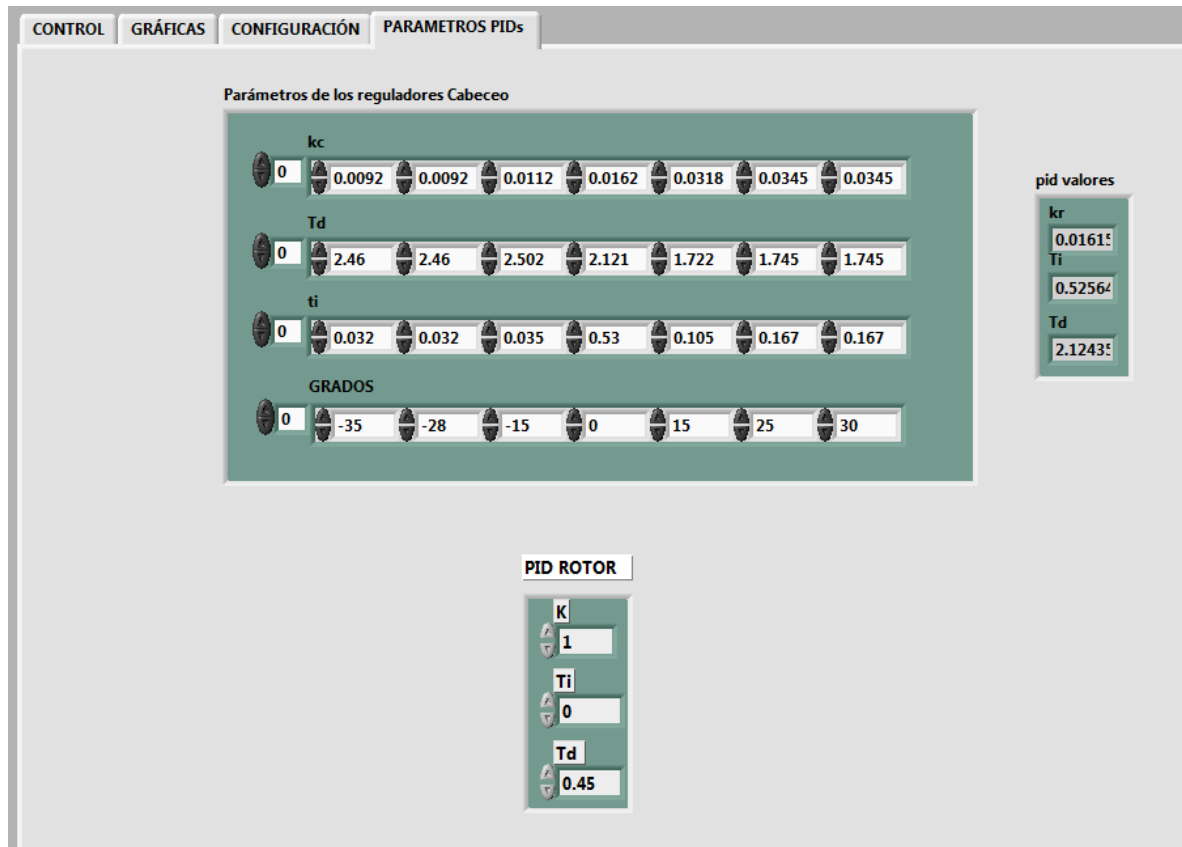


Figura 7 Pestaña parámetros PIDs.

Si se desea usar la aplicación con otra maqueta, se recomienda seguir los pasos descritos en el manual de diseño y calcular los reguladores de la nueva maqueta, ya que no se asegura un correcto funcionamiento de estos reguladores con otra maqueta.

A la derecha de la interface que permite definir los valores para la planificación de ganancia hay un indicador que nos muestra el valor de los parámetros del PID, que se ha interpolado para el control de la inclinación en cada momento.

En la parte inferior se definen los valores del controlador para la rotación.



4 BIBLIOGRAFIA.

National Instruments. (s.f.). <http://www.ni.com>. Obtenido de
<http://www.ni.com/labview/requirements/esa/>



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 5: ANEXO DE DISEÑO

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



ÍNDICE

DOCUMENTO Nº 5: ANEXO DE DISEÑO	0
1 INTRODUCCIÓN Y OBJETIVOS.	6
2 METODOLOGÍA SEGUIDA.	7
2.1 LABVIEW.....	7
2.2 MATLAB.....	8
2.2.1 Ident Tool.	8
2.2.2 Rltool	10
3 CONEXIÓN DE LA MAQUETA.....	11
4 CALIBRACIÓN Y CÁLCULOS PREVIOS.	12
4.1 CALIBRACIÓN DE LOS SENSORES.....	12
4.1.1 Sensor de Cabeceo del Helicóptero.	13
4.1.2 Sensor de Rotación del helicóptero.	13
4.2 Rango de funcionamiento de los motores.....	14
4.2.1 Motor de cabeceo del Helicóptero	14
4.2.2 Motor de Rotación del Helicóptero	15
4.3 LINEALIZACIÓN DE LA GANANCIA DEL PROCESO DE CABECEO	15
5 CONTROL PROCESO CABECEO	17
5.1 ELECCIÓN DE UNA SOLUCIÓN PARA LA IDENTIFICACIÓN Y CONTROL.	17
5.1.1 Realización de ensayos.....	18
5.1.2 Modelado con Ident Tool.....	18
5.1.3 Elección del controlador.	20
5.2 ZONA -28°.....	22
5.2.1 Realización de un ensayo válido en -28°.....	22
5.2.2 Cálculo del modelo para -28°.....	23
5.2.3 Cálculo del regulador para -28°.....	25



5.3 ZONA -15°.....	26
5.3.1 Cálculo del modelo para -15°.....	26
5.3.2 Cálculo del regulador para -15°.....	27
5.4 ZONA 0°.....	27
5.4.1 Cálculo del modelo para 0°.....	27
5.4.2 Cálculo del regulador para 0°.....	28
5.2 ZONA 15°.....	29
5.5.1 Cálculo del modelo para 15°.....	29
5.5.2 Cálculo del regulador para 15°.....	30
5.2 ZONA 25°.....	30
5.6.1 Cálculo del modelo para 25°.....	30
5.6.2 Cálculo del regulador para 25°.....	31
5.7 RESUMEN DEL PROCESO DE CABECEO.....	32
6 PROCESO DE ROTACIÓN.....	33
6.1 ENSAYO DE ROTACIÓN.....	33
6.2 MODELADO DE LA ROTACIÓN.....	34
6.3 DISEÑO DEL REGULADOR DE LA ROTACIÓN.....	35
7 CÁLCULOS PARA EL USO DEL JOYSTICK.....	37
8 BIBLIOGRAFÍA.....	38



ÍNDICE DE FIGURAS

Figura 1 Código aplicación cabeceo.	7
Figura 2 Ventanas de Ident Tool.	8
Figura 3 Ensayo valido dividido en subensayos para Ident Tool.....	9
Figura 4 Ventanas de Rltool	10
Figura 5 Parte inferior de la maqueta del helicóptero.....	11
Figura 6 Parte superior de la maqueta.....	12
Figura 7 Parte superior de la maqueta.....	12
Figura 8 Gráficas ganancia real y ganancia lineal.....	16
Figura 9 Ecuaciones de liberalización en el código de LabVIEW.....	17
Figura 10 Escalones para la selección del modelo.	18
Figura 11 Ventana de Ident con los modelos identificados.	19
Figura 12 Validación del modelo con Ident.....	19
Figura 13 Lugar de las raíces modelo cabeceo.....	20
Figura 14 Lugar de las raíces modelo cabeceo con un PI.....	20
Figura 15 Lugar de las raíces cabeceo sin cancelación.....	21
Figura 16 Lugar de las raíces cabeceo con cancelación.	21
Figura 17 Datos del ensayo para modelar en la zona de -28°	22
Figura 18 Subensayos zona -28°	23
Figura 19 Ensayo rotor.	33
Figura 20 Validación de los modelos de rotación.	34
Figura 21Lugar de las raíces modelo rotación.	36
Figura 22 Rectas de conversión de los sensores del joystick.....	37



ÍNDICE DE TABLAS

Tabla 1 Modelos -28°	24
Tabla 2 Parámetros modelos 28°	24
Tabla 3 Parámetros controlador -28°	25
Tabla 4 Modelos -15°	26
Tabla 5 Parámetros modelos -15°	26
Tabla 6 Parámetros controlador -15°	27
Tabla 7 Modelos 0°	27
Tabla 8 Parámetros modelos 0°	28
Tabla 9 Parámetros controlador 0°	28
Tabla 10 Modelos 15°	29
Tabla 11 Parámetros modelos 15°	29
Tabla 12 Parámetros controlador 15°	30
Tabla 13 Modelos 25°	30
Tabla 14 Parámetros modelos 25°	31
Tabla 15 Parámetros controlador 25°	31
Tabla 16 Modelos del proceso de cabeceo.	32
Tabla 17 Reguladores proceso cabeceo.	32
Tabla 18 Modelo de rotación.	35
Tabla 19 Controladores de rotación.	36



1 INTRODUCCIÓN Y OBJETIVOS.

En el presente documento se ha procedido a detallar los cálculos y ensayos realizados durante la elaboración del trabajo final de grado y se han detallado y justificado las soluciones adoptadas.

Este documento no explica el código de LabVIEW, éste se explica con detalle en los manuales del programador y en el manual de usuario. A continuación, en el presente documento, se procederá a detallar los cálculos necesarios para controlar el helicóptero. Cada maqueta tiene calibrados los sensores de manera diferente y, por supuesto, cada maqueta tiene una inercia y dinámica diferentes, por lo que los datos obtenidos solamente servirán para poder controlar dicha maqueta. Ésta es la usada en los ensayos realizados para el modelado y diseño de los reguladores, que controlan los movimientos de la maqueta.

Para utilizar la aplicación desarrollada en otra maqueta será necesario replicar los ensayos y cálculos detallados en este anexo. Una vez obtenidos los nuevos reguladores, se han de cambiar los datos de éstos.

Un mínimo cambio en el funcionamiento de uno de los motores, o en la posición de un cable, ocasiona una variación significativa en los modelos del sistema, lo que ocasiona que los reguladores diseñados no se comporten como se indica en la memoria.

2 METODOLOGÍA SEGUIDA.

En este apartado se detallan las metodologías utilizadas en los programas empleados para el desarrollo del proyecto, se indica la forma de procesar y tratar los datos para obtener como solución unos controladores válidos para el proceso.

2.1 LABVIEW.

Los ensayos realizados para la obtención de los datos, necesarios para la identificación y posterior diseño de los reguladores, se han de realizar a partir de las aplicaciones desarrolladas con este fin. En el anexo del programador se detalla la programación de la aplicación para los ensayos de cabeceo y rotación. Se trata de unos programas básicos en LabVIEW que nos permiten realizar variaciones en la variable manipulada para obtener la respuesta del sistema y el posterior guardado de los datos, si lo consideramos un ensayo válido (Figura 1).

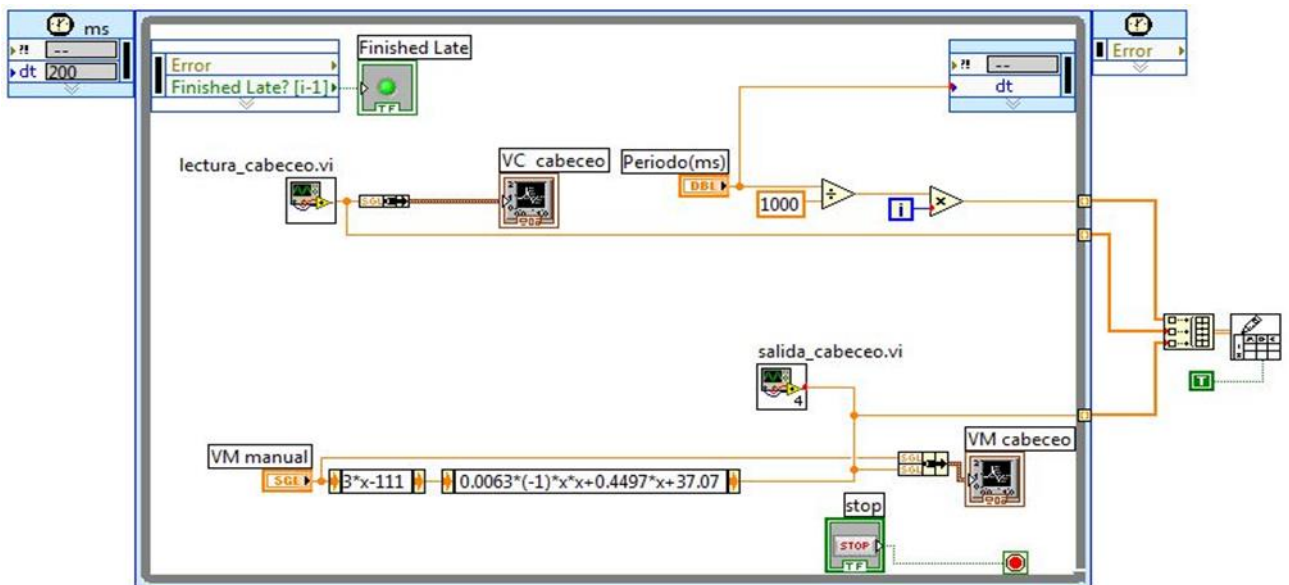


Figura 1 Código aplicación cabeceo.

2.2 MATLAB

Para la identificación y modelado de los sistemas y el posterior diseño de los reguladores, se emplearán diversas herramientas incluidas en Matlab:

- Ident Tool para la identificación de los modelos.
- Rltool para el diseño de los reguladores.

Se podría utilizar la herramienta Simulink de Matlab para la validación de los reguladores, sin embargo, para los ensayos y modelos utilizados, el diseño por rltool y el posterior ensayo sobre la maqueta serán suficientes.

2.2.1 Ident Tool.

Los datos de los ensayos realizados en LabVIEW se procesan en Matlab, para la identificación que realizamos en este documento serán necesarios escalones en la variable manipulada, de esta manera en la salida del sistema (variable controlada) aparecerá la dinámica que deseamos identificar. Para que estos datos sean válidos para la identificación en Ident Tool han de cumplir ciertos requisitos. Si un ensayo dado no los cumple todos será necesario repetir el ensayo, éstos son:

- El escalón en la variable manipulada no ha de saturar la variable controlada.
- El sistema tiene que partir de un punto de equilibrio.
- Antes de realizar otro escalón, el sistema ha de alcanzar el régimen permanente.

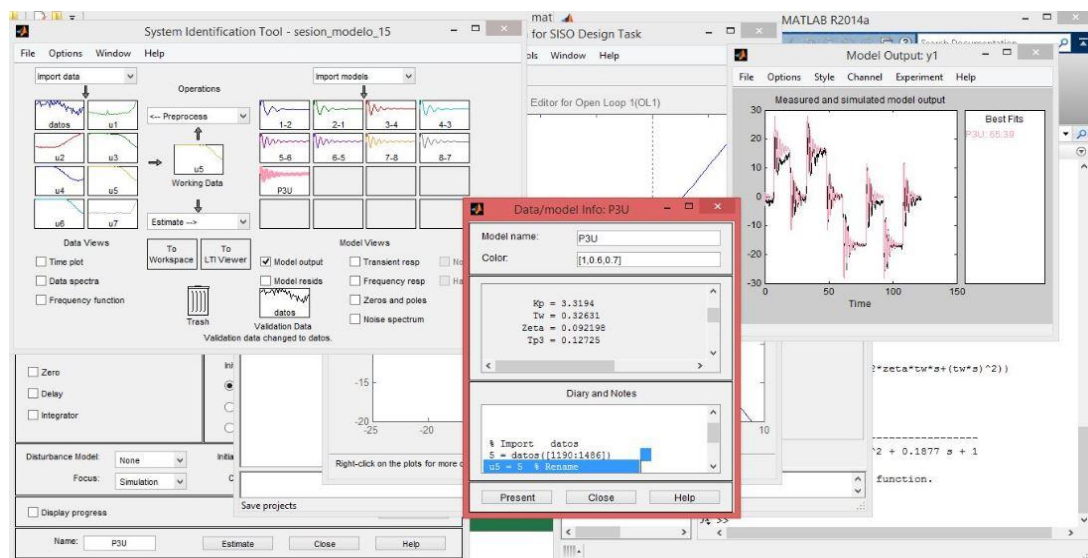


Figura 2 Ventanas de Ident Tool.

Posteriormente, cuando un ensayo es válido, se comienza el procesado de los datos para el uso de los mismos en Ident Tool. Cada ensayo se divide de manera que cada conjunto de datos contenga un solo escalón, de esta manera, un mismo ensayo puede contener diversos escalones en la señal manipulada, agilizando así la toma de datos. Cada conjunto de datos se denominará subensayo. En la Figura 3 se aprecia como quedará un ensayo dividido en subensayos.

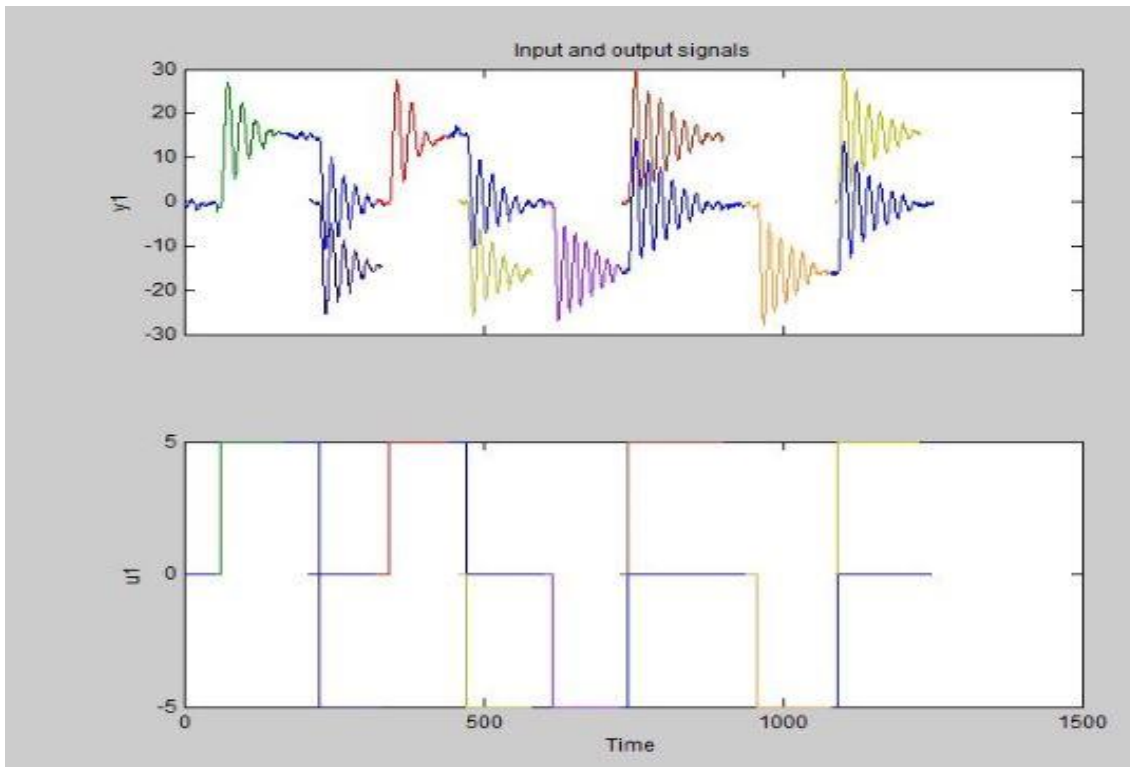


Figura 3 Ensayo valido dividido en subensayos para Ident Tool.

Además de dividir un ensayo en subensayos, para que los datos se puedan usar en Ident Tool, es necesario que todos los escalones partan del reposo, esto quiere decir que cada subensayo parte de cero, tanto CV como VM. Una vez tenemos los datos en Ident Tool, procedemos al modelado el sistema.

2.2.2 Rltool

Una vez identificado el modelo mediante Ident Tool, se procede a diseñar el regulador mediante rltool. Para ello se introduce el modelo y se añaden las especificaciones dinámicas y estáticas (sobreoscilación máxima y tiempo de establecimiento mínimo).

Una vez se ha definido la zona de especificaciones sobre el lugar de las raíces, se ha procedido a diseñar un regulador que cumpla dichas especificaciones, este regulador se comprueba experimentalmente sobre la maqueta.

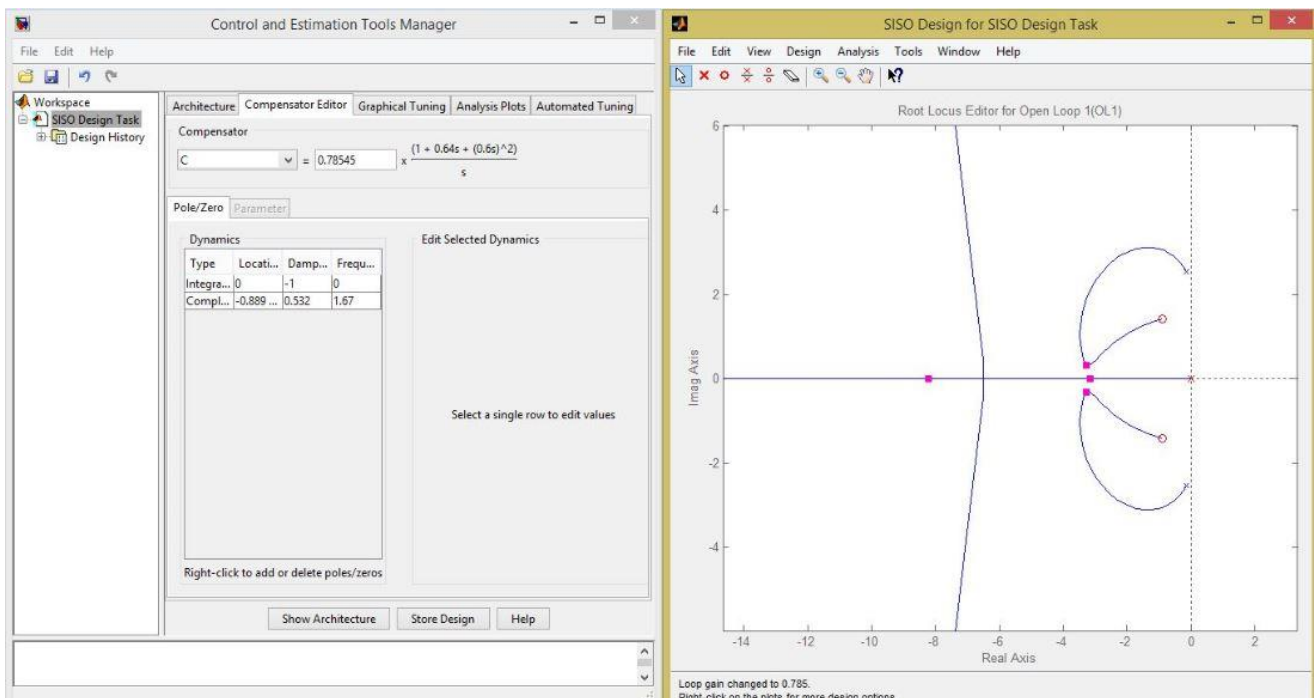


Figura 4 Ventanas de Rltool

Una vez implementado el regulador, se observa cuál es el comportamiento real de dicho regulador, y se da por válido o no según sea el comportamiento de éste.

3 CONEXIÓN DE LA MAQUETA.

La maqueta que se observa en la figura 1 ha de conectarse a la tarjeta de adquisición de datos (DAQ-LVIEW PnP PCI-9112 de Adlink). Dicha tarjeta ha de disponer y tener libres dos entradas analógicas de $\pm 10V$ y dos salidas analógicas: una de $\pm 10V$ para el motor de rotación y otra de $0V$ a $+10V$ para el motor de cabeceo del helicóptero.

Las conexiones se realizan siguiendo las indicaciones de la Figura 5.

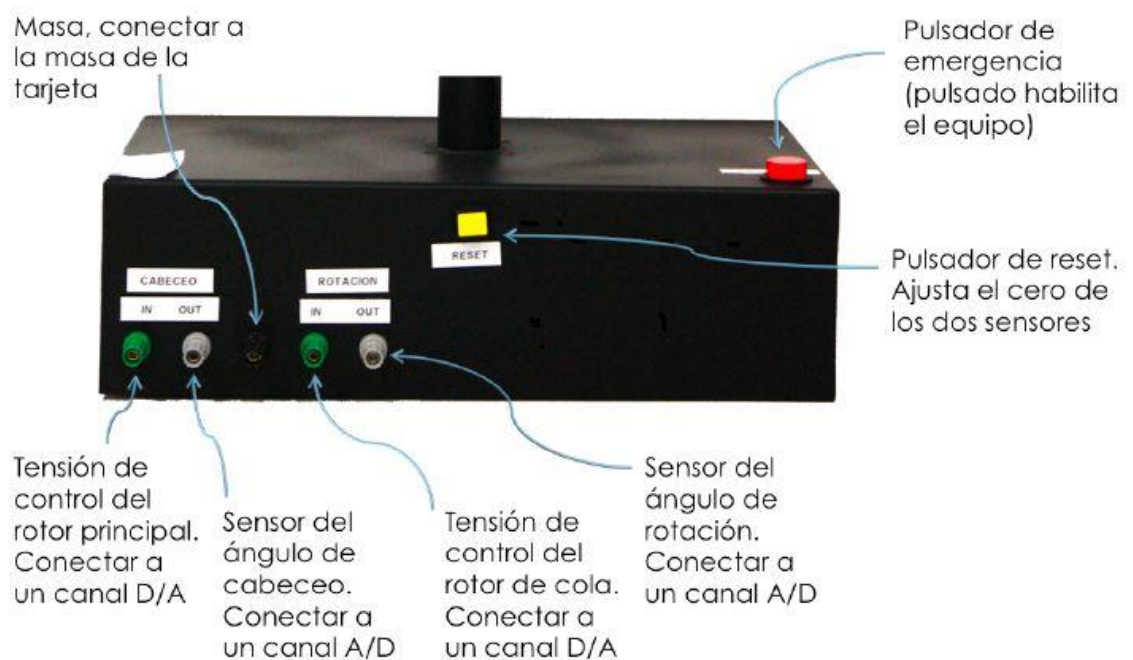


Figura 5 Parte inferior de la maqueta del helicóptero.

La función del interruptor rojo de emergencia es la de habilitar el equipo para la realización de las simulaciones.

La función del pulsador amarillo, denominado reset, es resetear los valores de los sensores y fijar el cero en la posición en la que se encuentre el helicóptero. Se utilizará para fijarlo en la posición horizontal a cero grados.

4 CALIBRACIÓN Y CÁLCULOS PREVIOS.

En este apartado se han detallado los cálculos previos a la realización de los ensayos, éstos son necesarios para definir las rectas de conversión de los sensores, rango de funcionamiento de las variables manipuladas y posibles correcciones necesarias. Este paso es importante ya que de los resultados dependerá que los datos obtenidos sean válidos para la identificación.

4.1 CALIBRACIÓN DE LOS SENSORES.

La maqueta dispone de dos sensores que miden el ángulo de cabeceo y rotación del helicóptero.

En la Figura 6 se muestran los movimientos que miden estos sensores.

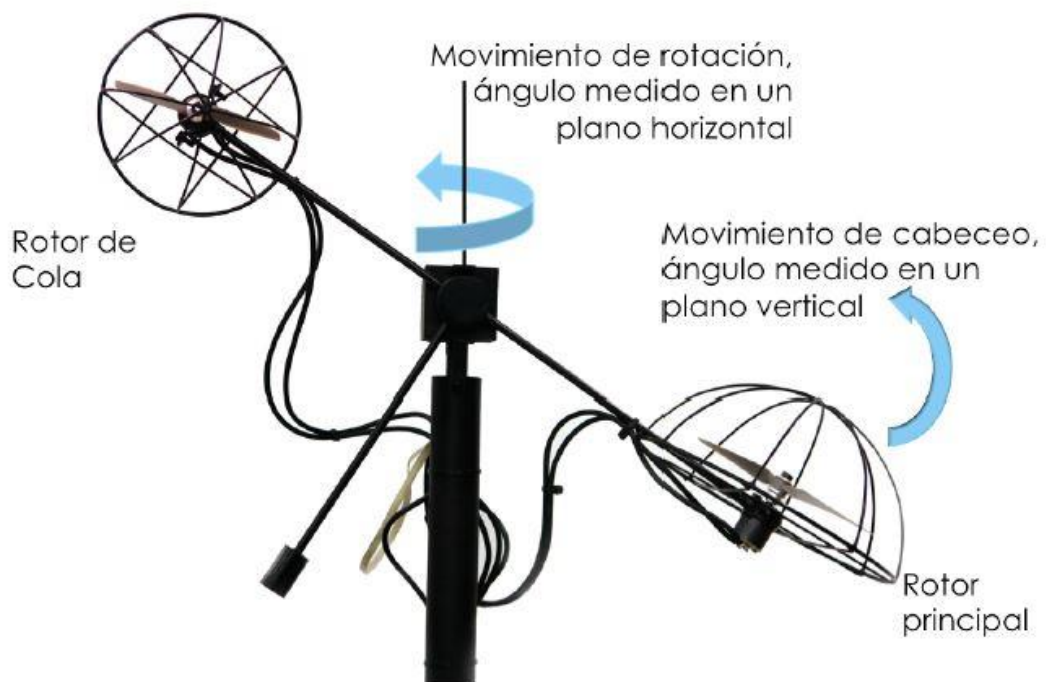


Figura 6 Parte superior de la maqueta.



4.1.1 Sensor de Cabeceo del Helicóptero.

El sensor de cabeceo de la maqueta de simulación proporciona una señal analógica que varía entre $\pm 10V$, para convertir esta señal a grados es necesario realizar una recta de conversión, para ello se realiza la medida en tres puntos:

- Primer punto: El sensor proporciona una tensión de 0V cuando la maqueta se encuentra en la posición que el botón reset le define como posición de equilibrio, para el correcto funcionamiento esta posición se define como cero grados (horizontal).
- Segundo punto: Para medir este punto es necesario un porta ángulos, la medida se realizará a 45 grados de la horizontal, en este punto el sensor marca 5V.
- Tercer punto: Procediendo como en el segundo punto, se coloca la maqueta a 45 grados con la horizontal y se anota la tensión marcada por el sensor, es este caso -5V.

La recta de conversión resultante de los tres puntos medidos es $Y=9*X$, siendo Y los grados de rotación de la maqueta y X la tensión marcada por el sensor.

4.1.2 Sensor de Rotación del helicóptero.

El sensor de rotación de la maqueta de simulación proporciona una señal analógica que varía entre $\pm 10V$, para convertir esta señal a grados girados es necesario realizar una recta de conversión, para ello se realiza la medida en tres puntos:

- Primer punto: El sensor proporciona una tensión de 0V cuando la maqueta se encuentra en la posición que el botón reset le define como posición de equilibrio. Para el correcto funcionamiento, esta posición la define la posición que fija el tornillo de sujeción.
- Segundo punto: Para medir este punto es necesario un porta ángulos, la medida se realizará a 90 grados hacia la derecha del punto de equilibrio. En este punto el sensor marca 5V.



- Tercer punto: Procediendo como en el segundo punto, la maqueta se colocará a 90 grados hacia la izquierda del punto de equilibrio y se anota la tensión marcada por el sensor. En este caso -5V.

La recta de conversión resultante de los tres puntos medidos es $Y=14*X$, siendo Y los grados de rotación de la maqueta y X la tensión marcada por el sensor.

4.2 Rango de funcionamiento de los motores.

Los motores se conectan a las salidas de la tarjeta de adquisición de datos siguiendo lo indicado en el punto 3 de este anexo.

4.2.1 Motor de cabeceo del Helicóptero.

El motor de cabeceo está controlado por una señal analógica que puede variar entre 0V y 10V, para 0V el motor está completamente parado y para 10V el motor funciona a plena potencia.

Tras realizar ciertos ensayos preliminares y observar el comportamiento de la maqueta, se ha llegado a la decisión de limitar la tensión máxima del motor a 5V, para garantizar la integridad del motor y la estabilidad del helicóptero, ya que se aprecia cómo tensiones superiores a 5 voltios ocasionan la inestabilidad del sistema y someten al motor a esfuerzos innecesarios para el correcto funcionamiento de la maqueta. Por lo que el rango de funcionamiento del motor de cabeceo se limita al rango [0V, 5V].



4.2.2 Motor de Rotación del Helicóptero.

El motor de rotación está controlado por una señal analógica que puede variar entre $\pm 10V$, para $0V$ el motor está completamente parado, para $10V$ el motor funciona a plena potencia hacia la derecha y para -10 el motor funciona a plena potencia hacia la izquierda.

Al igual que sucede con el motor de cabeceo, tras realizar ciertos ensayos preliminares y observar el comportamiento de la maqueta, se ha llegado a la decisión de limitar la tensión máxima del motor a $5V$ y la mínima a $-5V$, para garantizar la integridad del motor y la estabilidad del helicóptero.

4.3 LINEALIZACIÓN DE LA GANANCIA DEL PROCESO DE CABECEO.

Al realizar los ensayos preliminares se ha observado como la ganancia en el proceso de cabeceo en la maqueta de simulación no es lineal. Así, al principio, para variaciones relativamente grandes en la tensión no se ha modificado apenas el ángulo de inclinación. Mientras que una vez alcanzados ángulos superiores a 25 grados la ganancia aumenta de manera exponencial.

Para evitar la no linealidad, que puede afectar al correcto funcionamiento del controlador PID, se opta por linealizar la ganancia del proceso antes de continuar con los ensayos. Para linealizar la ganancia es necesario realizar diferentes medidas a través de todo el rango de funcionamiento. El rango para el ángulo de cabeceo (variable controlada) es $[-30^\circ, 30^\circ]$.

El punto de equilibrio, en el cual el helicóptero se sitúa a 0° , es a la tensión de $3.7V$. Partiendo de ese punto como referencia, se realiza la medición de los grados a los que se sitúa la maqueta para tensiones constantes, dando como resultado la curva de la Figura 8. Ésta se ha aproximado mediante una ecuación de segundo grado, dando como resultado $y=0,0798x^2 - 3,3175x+13,762$. Partiendo de ésta se han calculado las ecuaciones para convertir la ganancia en lineal.

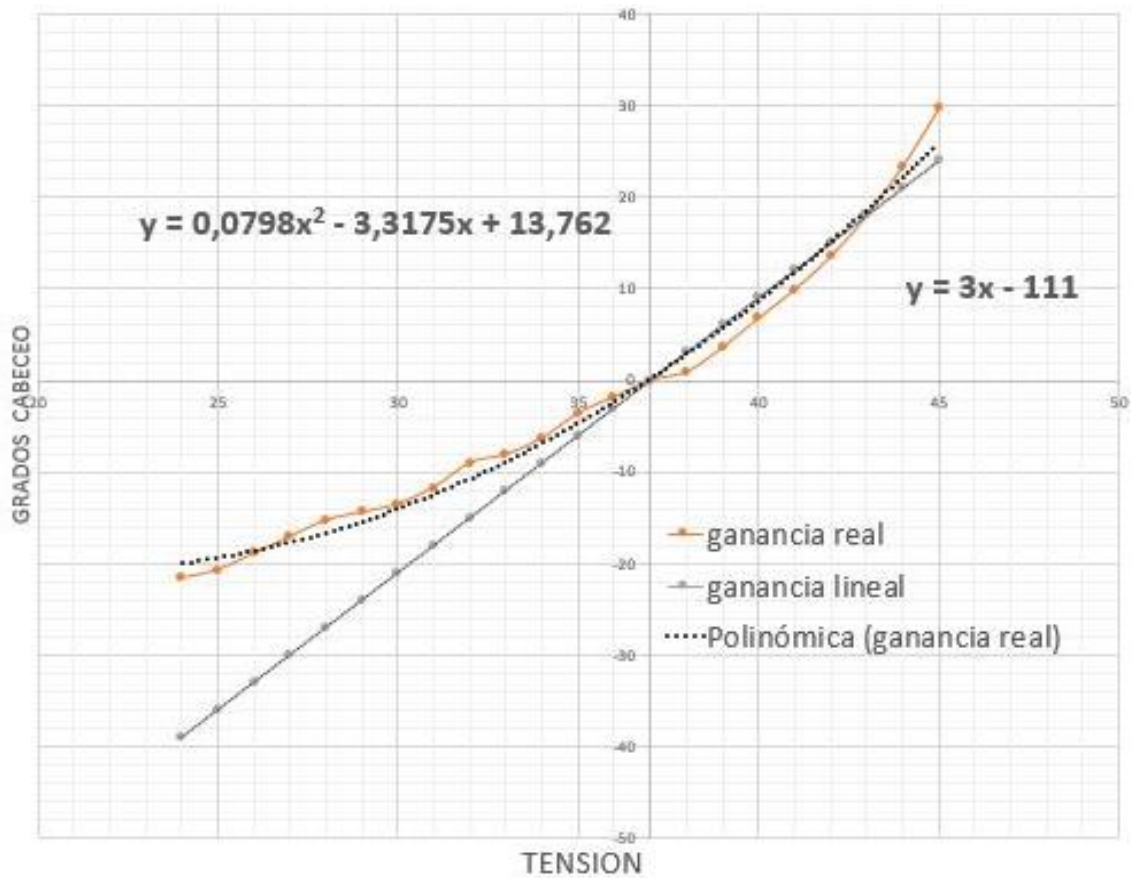


Figura 8 Gráficas ganancia real y ganancia lineal.

Para convertir la ganancia a lineal se han empleado dos ecuaciones que se situarán en el código de LabVIEW antes de mandar la tensión a la maqueta. La ganancia lineal da como resultado una variación de 3° por cada 0.1V de variación en la tensión del motor.

La primera ecuación transforma la tensión proporcionada por el programa a los grados que se obtendrían si la ganancia fuera lineal y la segunda convierte esos grados en la tensión que se debe dar realmente para conseguir éstos (inversa a la curva aproximada de la ganancia) $x = -0,0067y^2 + 0,46056y + 37,081$.

En la Figura 9 se muestra como quedará una vez implementada la linealización en LabVIEW.

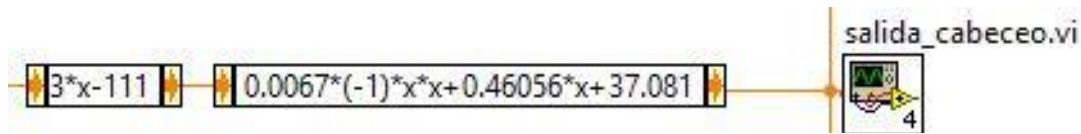


Figura 9 Ecuaciones de liberalización en el código de LabVIEW.

5 CONTROL PROCESO CABECEO

Para implementar el planificador de ganancia se identifican cinco modelos y, por lo tanto, se diseñan cinco reguladores. Las zonas de funcionamiento seleccionadas para realizar estos controles son: -28° , -15° , 0° , 15° y 25° . A continuación, se detallan los cálculos y metodología seguida en Matlab para obtener los modelos y reguladores en cada zona.

5.1 ELECCIÓN DE UNA SOLUCIÓN PARA LA IDENTIFICACIÓN Y CONTROL.

Para la selección del modelo a identificar se realizarán varias identificaciones sobre un escalón de prueba y se seleccionará el modelo que más se aproxime al sistema. Una vez seleccionado el modelo, se realizarán varios reguladores y se seleccionará el adecuado.

5.1.1 Realización de ensayos.

Para el correcto identificado del modelo del sistema son necesarios los datos de un ensayo válido. Para la selección de una solución de identificación válida, se ha realizado un ensayo que tiene como posición de equilibrio la horizontal (0°).

La posición de equilibrio se alcanza con una tensión en la salida del motor de 3.7V, para el ensayo se realizarán escalones de un valor igual al 5% en VM (U1).

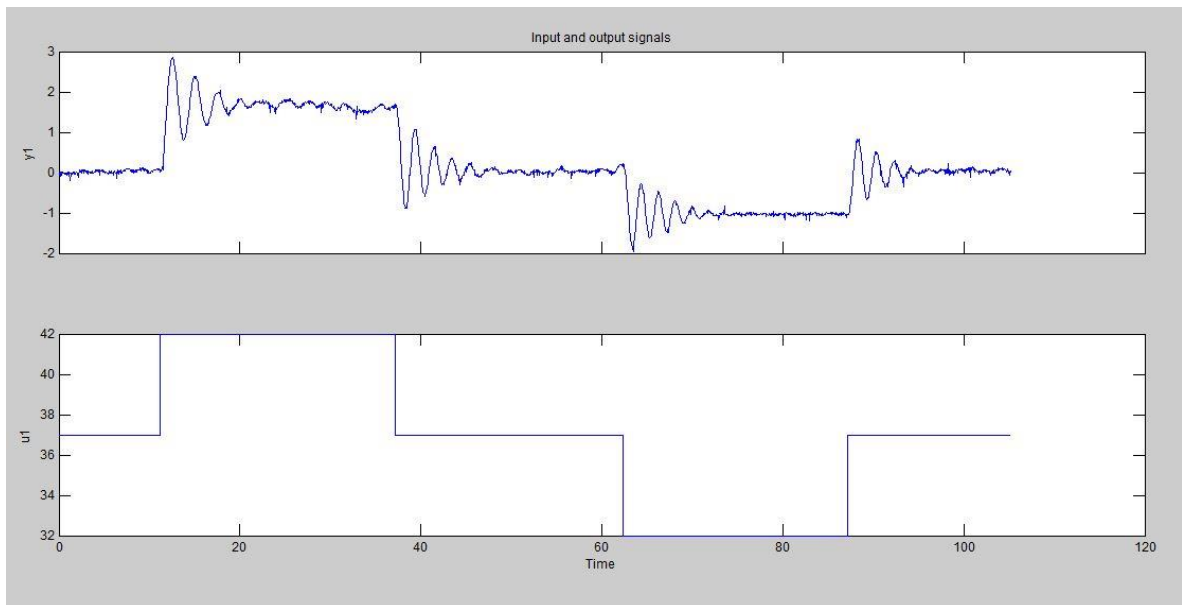


Figura 10 Escalones para la selección del modelo.

5.1.2 Modelado con Ident Tool.

El modelo de partida para la identificación es un modelo de segundo orden con polos complejos, se realizan tres modelados:

- Segundo orden con polos complejos.
- Tercer orden con polos complejos y polo real.
- Tercer orden con polos complejos, polo real y un cero.

No se consideran los modelos con retardo, ya que la respuesta del sistema es inmediata.

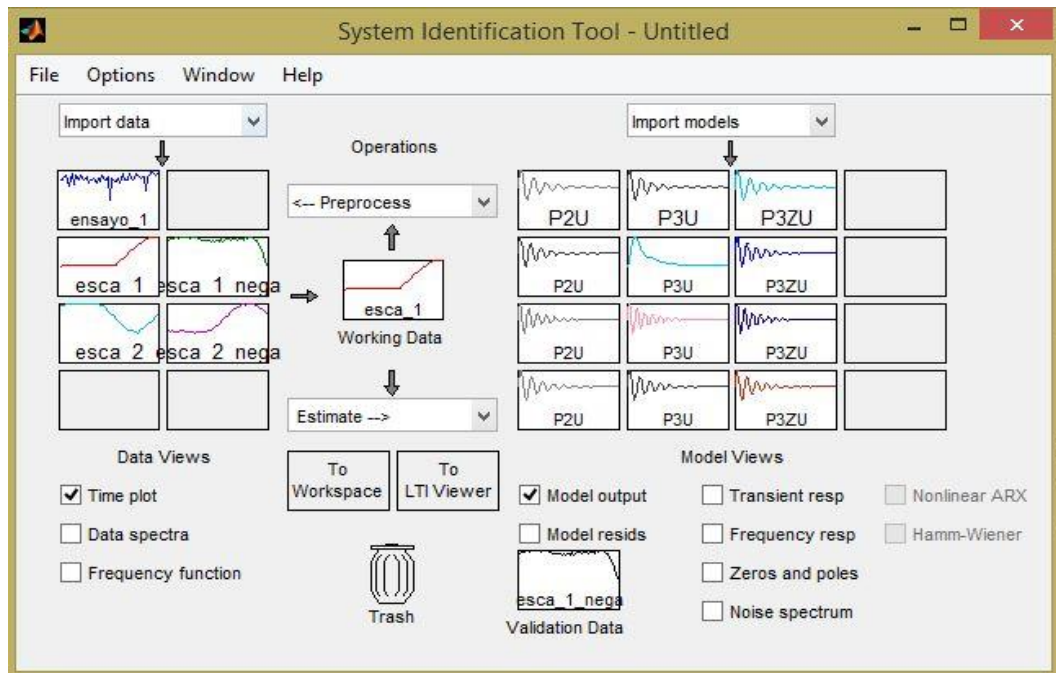


Figura 11 Ventana de Ident con los modelos identificados.

Se realiza la identificación de los tres modelos en los cuatro escalones disponibles en el ensayo de prueba y se llega a la conclusión de que el modelo indicado para el sistema de cabeceo del helicóptero es el modelo de tercer orden sin cero.

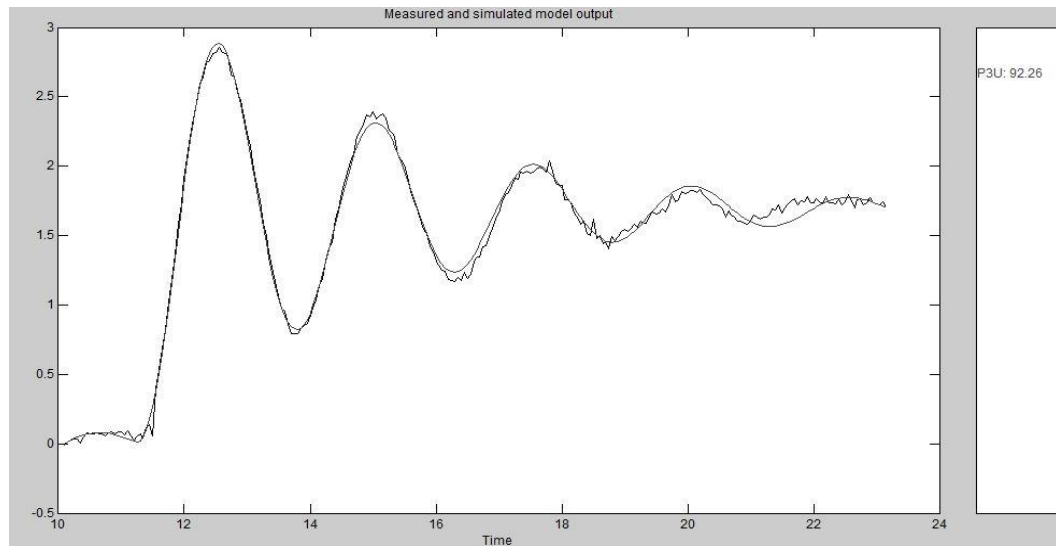


Figura 12 Validación del modelo con Ident.

Se ha realizado la validación del modelo mediante la herramienta de simulación que incorpora Ident. Como se puede ver en la Figura 12, el modelo proporciona un 92% de coincidencia con el sistema, por lo que se procede a identificar los modelos en cada zona de funcionamiento.

5.1.3 Elección del controlador.

Para el cálculo del regulador, se utiliza la herramienta Rltool de Matlab una vez se ha seleccionado el modelo del sistema como un tercer orden con dos polos complejos y un polo real. Se procede a introducir el modelo y las restricciones que debe cumplir el sistema en bucle cerrado. El lugar de las raíces del modelo es el mostrado en la Figura 13.

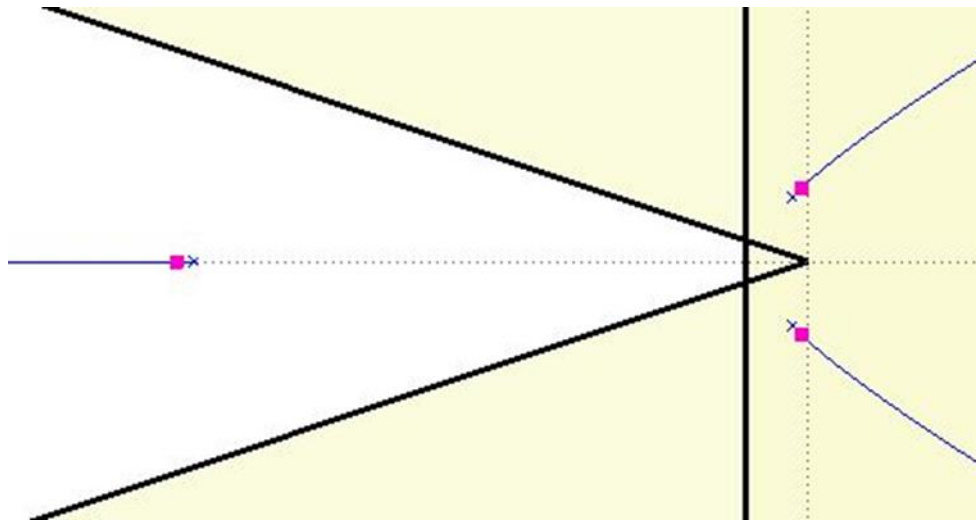


Figura 13 Lugar de las raíces modelo cabeceo.

Los reguladores seleccionados deben introducir los polos en bucle cerrado dentro de la zona de especificaciones. El regulador diseñado será un PID completo, ya que necesitamos error de posición nulo y un PI no permite controlar el sistema, como se ve en la Figura 14.

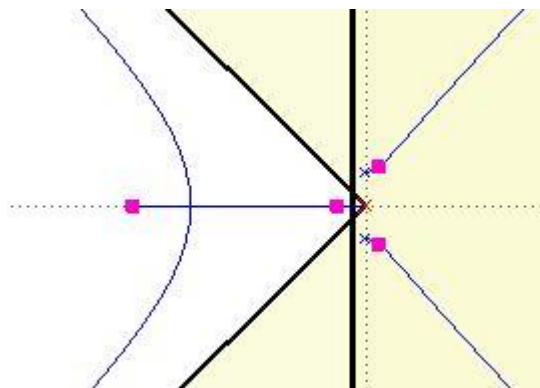


Figura 14 Lugar de las raíces modelo cabeceo con un PI

Se diseñan dos posibles soluciones de reguladores:

- Cancelación de los polos complejos.
- Ceros complejos que no cancelen los del sistema.

En la Figura 15 y en la Figura 16, se pueden ver los dos lugares de las raíces tras colocar los reguladores.

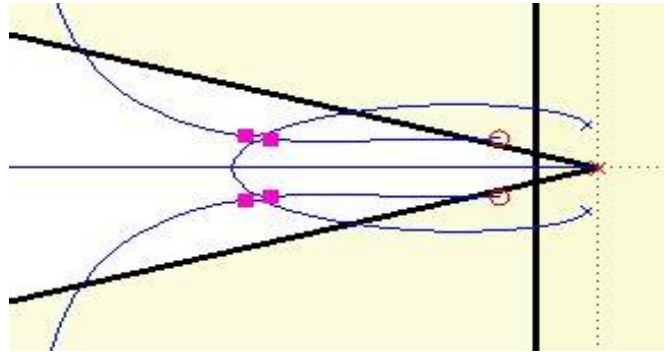


Figura 15 Lugar de las raíces cabeceo sin cancelación.

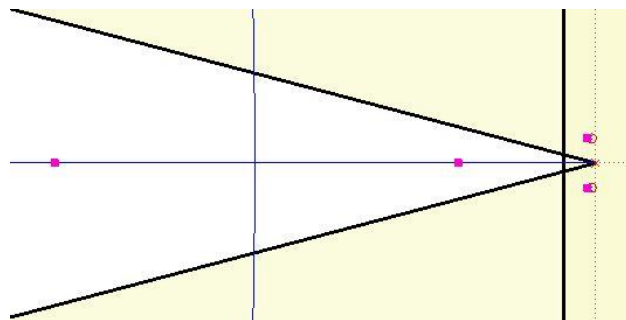


Figura 16 Lugar de las raíces cabeceo con cancelación.

Los reguladores se han validado sobre la maqueta (se realiza una prevalidación con la herramienta rtool) y el regulador diseñado por cancelación da mejores resultados, por lo que el regulador seleccionado para realizar el control de cabeceo del helicóptero es un PID por cancelación de los polos complejos.

5.2 ZONA -28° .

En este apartado se detallan los cálculos realizados en esta zona de funcionamiento: ensayo, modelado y diseño del controlador.

5.2.1 Realización de un ensayo válido en -28° .

Para la realización de un ensayo válido para identificar el modelo en este rango de funcionamiento se ha procedido de la siguiente manera.

Se ha situado la maqueta en -28° como punto de equilibrio. Este se alcanza con una tensión en la salida de 2.8V. Se ha procedido a realizar escalones de valor $\pm 5\%$ en la variable manipulada. Los datos obtenidos se muestran en la Figura 17, estos datos son importados a Ident para la posterior identificación.

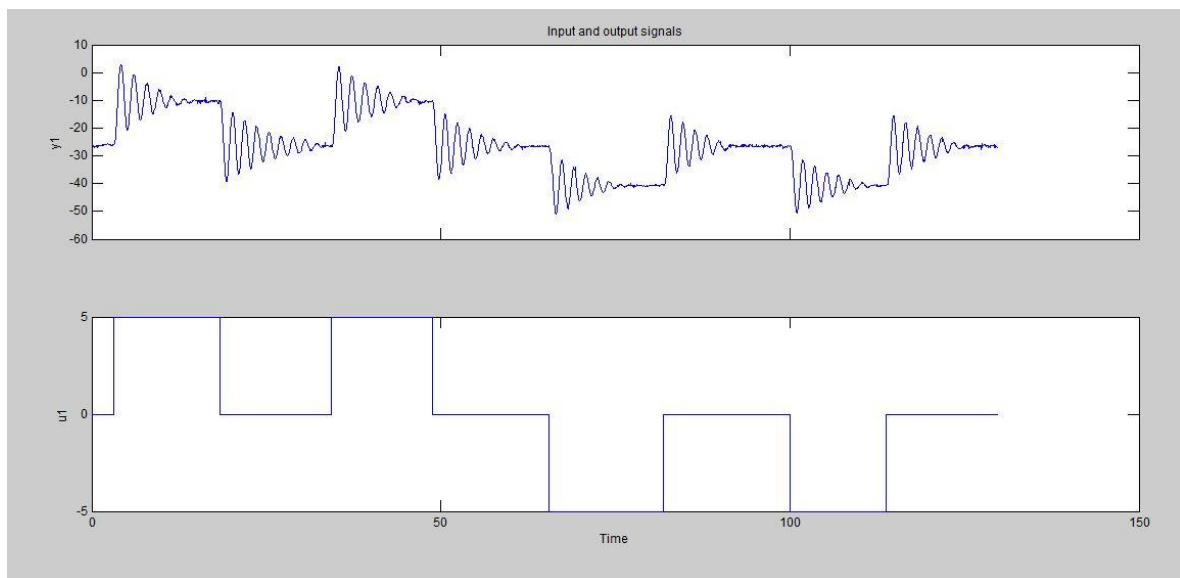


Figura 17 Datos del ensayo para modelar en la zona de -28° .

Para mayor precisión en los modelos obtenidos, todos los escalones se realizan por duplicado, de esta manera se dispone de más datos para validar e identificar en Ident.

5.2.2 Cálculo del modelo para -28°.

Para la identificación de modelos con Ident Tool procedemos a dividir el ensayo en subensayos, como se muestra en la Figura 18.

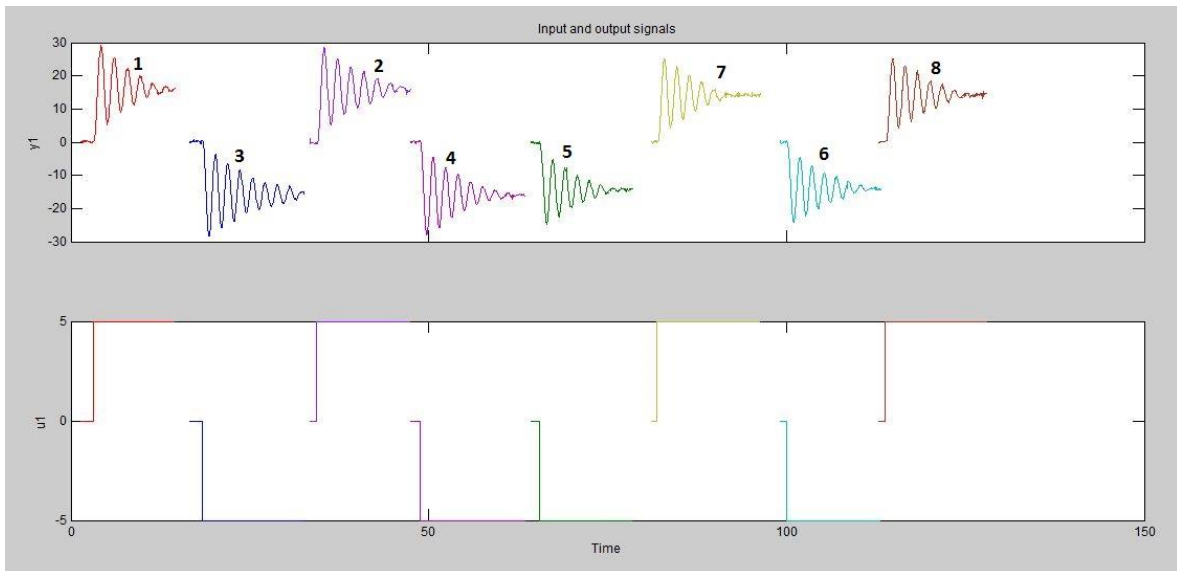


Figura 18 Subensayos zona -28°.

Disponemos de ocho subensayos dentro del ensayo, éstos se agrupan de dos en dos. Así disponemos de dos parejas de escalones positivos 1-2, 5-6 y dos de escalones negativos 3-4 y 7-8.

Procedemos a identificar los modelos. Se identificarán ocho modelos, el modelo que se ha seleccionado para el sistema de cabeceo ha sido un modelo de tercer orden con dos polos complejos y un polo real, la función de transferencia del modelo es:

$$G(s) = \frac{Kp}{(1 + Zeta * Tw * s + (Tw * s)^2)(1 + Tp3 * s)}$$



Las parejas de escalones se han empleado para disponer de datos de identificación y validación diferentes. En la Tabla 1 se muestran los parámetros de los ocho modelos.

escalón ¹	k	tw	zeta	tp3	Validación %
1-2	3,2166	0,30217	0,073869	0,1	79,17
2-1	3,2096	0,2937	0,055965	0,14443	88,77
3-4	3,1466	0,27626	0,041216	0,14441	86,66
4-3	2,1651	0,27735	0,054749	0,117	88
5-6	2,884	0,27169	0,059893	0,1747	88,44
6-5	2,8047	0,2714	0,053816	0,16496	87,37
7-8	2,8346	0,27601	0,06181	0,16513	87,42
8-7	2,8576	0,27608	0,054135	0,16371	86,14

Tabla 1 Modelos -28°.

Para la obtención del modelo final se ha realizado la media entre los modelos:

	k	tw	zeta	tp3	%
Media escalones positivos	3,03	0,286	0,0614	0,143	85,3
Media escalones negativos	2,76	0,274	0,0524	0,150	87,6
Modelo -28°	2,88	0,280	0,0569	0,146	86,4

Tabla 2 Parámetros modelos 28°.

En la siguiente ecuación se ve la función de transferencia del modelo en -28°:

$$G(s)_{-28^\circ} = \frac{2.88}{(1 + 0.32 * s + 0.079 * s^2)(0.146 + s)}$$

¹ 1-2 significa que se ha identificado con el escalón 1 y se ha validado con los datos de 2, así en los 8 modelos.



5.2.3 Cálculo del regulador para -28°.

Partiendo del modelo identificado para -28°, se diseña un regulador PID. Los ceros se fijan por cancelación de los polos complejos y la constante se fija para el cumplimiento del tiempo de establecimiento. La fórmula de un PID es la mostrada a continuación:

$$PID = \frac{Kp (s + a)(s + b)}{s} = \frac{Kr (Td * s^2 + s + 1/Ti)}{s}$$

El PID por cancelación para -28° es:

$$PID_{-28^\circ} = \frac{Kp (0.079s^2 + 0.32s + 1)}{s} = \frac{0.0092 (2.46 * s^2 + s + 1/0.32)}{s}$$

En la Tabla 3 se indican los parámetros del controlador en la zona de -28°.

ZONA REGULADOR	Kr	Ti	Td
-28 grados	0.0092	0,032	2,46

Tabla 3 Parámetros controlador -28°.

5.3 ZONA -15°.

En la zona de -15° se procede de forma similar a lo realizado en la zona de -28°. Se realiza un ensayo válido partiendo de -15° como punto de equilibrio, esto se logra fijando la tensión del motor en 3.2V. Se realizan cuatro escalones, dos positivos y dos negativos y se guardan los datos. El ensayo se carga en Matlab y se procesan los datos, se divide en subensayos y se normalizan para la identificación en Ident Tool.

5.3.1 Cálculo del modelo para -15°.

El modelo identificado es un tercer orden con dos polos complejos y un polo real, se identifican los 8 modelos. En la Tabla 4 se muestran los parámetros de los mismos.

escalón	k	tw	zeta	tp3	Validación %
1-2	3,23	0,327	0,0681	0,101	85,8
2-1	3,14	0,326	0,0757	0,077	88,2
3-4	3,03	0,292	0,0554	0,138	88,5
4-3	3,06	0,291	0,0602	0,145	89,1
5-6	3,33	0,275	0,0471	0,147	88,3
6-5	3,25	0,274	0,0529	0,143	88,6
7-8	3,21	0,291	0,0580	0,123	89,7
8-7	3,29	0,292	0,0558	0,127	89,7

Tabla 4 Modelos -15°.

	k	tw	zeta	tp3	%
Media escalones positivos	3,21	0,309	0,0644	0,107	88,4
Media escalones negativos	3,17	0,283	0,0539	0,143	88,6
Modelo -15°	3,19	0,296	0,0592	0,125	88,5

Tabla 5 Parámetros modelos -15°.

En la Tabla 5 se muestran los parámetros del modelo identificado para -15° y en la siguiente ecuación la forma de la función de transferencia final para -15°:

$$G(s)_{-15^\circ} = \frac{3.19}{(1 + 0.35 * s + 0.088 * s^2)(0.125 + s)}$$

5.3.2 Cálculo del regulador para -15°.

Partiendo del modelo identificado para -15°, se diseña un regulador. Los ceros se fijan por cancelación de los polos complejos y la constante se fija para el cumplimiento del tiempo de establecimiento. El PID por cancelación para -15° es:

$$PID_{-15^\circ} = \frac{Kp (0.088s^2 + 0.35s + 1)}{s} = \frac{0.0112 (2.5 * s^2 + s + 1/0.35)}{s}$$

En la Tabla 6 se indican los parámetros del controlador en la zona de -15°.

	Kr	Ti	Td
-15 grados	0.0112	0,035	2,502

Tabla 6 Parámetros controlador -15°.

5.4 ZONA 0°.

En la zona de 0° se procede de forma similar. Se realiza un ensayo válido partiendo de 0° como punto de equilibrio, esto se logra fijando la tensión del motor en 3.7V. Se realizan cuatro escalones, dos positivos y dos negativos y se guardan los datos. El ensayo se carga en Matlab y se procesan los datos, se divide en subensayos y se normalizan para la identificación en Ident Tool.

5.4.1 Cálculo del modelo para 0°.

escalón	k	tw	zeta	tp3	Validación %
1-2	3,096	0,393	0,090	0,061	78,5
2-1	3,070	0,394	0,100	0,089	84,5
3-4	2,924	0,323	0,074	0,146	76,5
4-3	3,166	0,330	0,083	0,105	76,7
5-6	3,124	0,287	0,069	0,150	88,1
6-5	3,146	0,291	0,061	0,132	87,4
7-8	3,097	0,328	0,076	0,077	91,7
8-7	3,29	0,292	0,0558	0,127	89,7

Tabla 7 Modelos 0°.

En la Tabla 7 se pueden ver los parámetros de los ocho modelos identificados para 0°.

	k	tw	zeta	tp3	%
Media escalones positivos	3,084	0,360	0,086	0,078	86,2
Media escalones negativos	3,090	0,308	0,072	0,133	82,2
Modelo 0°	3,087	0,334	0,079	0,106	84,2

Tabla 8 Parámetros modelos 0°.

En la Tabla 8 se muestran los parámetros del modelo identificado para 0° y en la siguiente ecuación la forma de la función de transferencia final.

$$G(s)_{0^\circ} = \frac{3.09}{(1 + 0.52 * s + 0.111 * s^2)(0.106 + s)}$$

5.4.2 Cálculo del regulador para 0°.

Partiendo del modelo identificado para 0°, se diseña un regulador. Los ceros se fijan por cancelación de los polos complejos y la constante se fija para el cumplimiento del tiempo de establecimiento. El PID por cancelación para 0° es:

$$PID_{0^\circ} = \frac{Kp (0.111s^2 + 0.52s + 1)}{s} = \frac{0.0162 (2.121 * s^2 + s + 1/0.053)}{s}$$

En la Tabla 9 se indican los parámetros del controlador en la zona de 0°.

	Kr	Ti	Td
0 grados	0.0162	0,053	2,121

Tabla 9 Parámetros controlador 0°.

5.2 ZONA 15°.

En la zona de 15° se procede de forma similar. Se realiza un ensayo válido partiendo de 15° como punto de equilibrio, esto se logra fijando la tensión del motor en 4.2V. Se realizan cuatro escalones, dos positivos y dos negativos y se guardan los datos. El ensayo se carga en Matlab y se procesan los datos, se divide en subensayos y se normalizan para la identificación en Ident Tool.

5.5.1 Cálculo del modelo para 15°.

escalón	k	tw	zeta	tp3	Validación %
1-2	2,75	0,560	0,1945	0,012	64,3
2-1	3,03	0,588	0,1720	0,019	72,66
3-4	2,95	0,394	0,0933	0,245	79,5
4-3	3,06	0,406	0,1337	0,174	81,97
5-6	3,32	0,326	0,0922	0,127	74,17
6-5	3,02	0,327	0,0910	0,131	73,91
7-8	2,96	0,393	0,0995	0,057	71,39
8-7	3,26	0,401	0,1098	0,027	71,39

Tabla 10 Modelos 15°.

En la Tabla 10 se pueden ver los parámetros de los ocho modelos identificados para 15°.

	k	tw	zeta	tp3	%
Media escalones positivos	3,00	0,486	0,1439	0,029	69,9
Media escalones negativos	3,09	0,363	0,1025	0,169	77,4
Modelo 15°	3,04	0,424	0,1232	0,099	73,7

Tabla 11 Parámetros modelos 15°.

En la Tabla 11 se muestran los parámetros del modelo identificado para 15° y en la siguiente ecuación la forma de la función de transferencia del modelo final.

$$G(s)_{15^\circ} = \frac{3.043}{(1 + 0.105 * s + 0.18 * s^2)(0.099 + s)}$$

5.5.2 Cálculo del regulador para 15°.

Partiendo del modelo identificado para 15°, se diseña un regulador. Los ceros se fijan por cancelación de los polos complejos y la constante se fija para el cumplimiento del tiempo de establecimiento. El PID por cancelación para 15° es:

$$PID_{15^\circ} = \frac{K_p (0.18s^2 + 0.105s + 1)}{s} = \frac{0.0318 (1.722 * s^2 + s + 1/0.105)}{s}$$

En la Tabla 12 se indican los parámetros del controlador en la zona de 15°.

	Kr	Ti	Td
15 grados	0.0318	0.105	1.722

Tabla 12 Parámetros controlador 15°.

5.2 ZONA 25°.

En la zona de 25° se procede de forma similar. Se realiza un ensayo válido partiendo de 25° como punto de equilibrio, esto se logra fijando la tensión del motor en 4.7V. Se realizan cuatro escalones, dos positivos y dos negativos y se guardan los datos. El ensayo se carga en Matlab y se procesan los datos, se divide en subensayos y se normalizan para la identificación en Ident Tool.

5.6.1 Cálculo del modelo para 25°.

escalón	k	tw	zeta	tp3	Validación %
1-2	3,096	0,393	0,090	0,061	78,5
2-1	3,070	0,394	0,100	0,089	84,5
3-4	2,924	0,323	0,074	0,146	76,5
4-3	3,166	0,330	0,083	0,105	76,7
5-6	3,124	0,287	0,069	0,150	88,1
6-5	3,146	0,291	0,061	0,132	87,4
7-8	3,097	0,328	0,076	0,077	91,7
8-7	3,29	0,292	0,0558	0,127	89,7

Tabla 13 Modelos 25°.

En la Tabla 13 se pueden ver los parámetros de los ocho modelos identificados para 25°.

	k	tw	zeta	tp3	%
Media escalones positivos	3,094	0,398	0,0978	0,97	86,2
Media escalones negativos	3,090	0,308	0,072	0,133	82,2
Modelo 25°	3,025	0,4895	0,1452	0,09458	84,2

Tabla 14 Parámetros modelos 25°.

En la Tabla 11 se muestran los parámetros del modelo identificado para 25° y en la siguiente ecuación la forma de la función de transferencia del modelo final.

$$G(s)_{25^\circ} = \frac{3.08}{(1 + 0.52 * s + 0.111 * s^2)(0.106 + s)}$$

5.6.2 Cálculo del regulador para 25°.

Partiendo del modelo identificado para 25°, se diseña un regulador. Los ceros se fijan por cancelación de los polos complejos y la constante se fija para el cumplimiento del tiempo de establecimiento. El PID por cancelación para 25° es:

$$PID_{25^\circ} = \frac{Kp (0.111s^2 + 0.52s + 1)}{s} = \frac{0.0162 (2.121 * s^2 + s + 1/0.053)}{s}$$

En la Tabla 6 se indican los parámetros del controlador en la zona de 25°.

	Kr	Ti	Td
25 grados	0.0345	0,1675	1.745

Tabla 15 Parámetros controlador 25°.

5.7 RESUMEN DEL PROCESO DE CABECEO.

Los modelos identificados para los cinco puntos de funcionamiento seleccionados para realizar la planificación de la ganancia se recogen en la Tabla 16.

ZONA MODELO	k	tw	zeta	tp3
25 GRADOS	3,025	0,4895	0,1452	0,09458
15 GRADOS	3,0430	0,4245	0,1232	0,0990
0 GRADOS	3,0872	0,0526	0,1116	0,1056
- 15 GRADOS	3,1905	0,2960	0,0592	0,1250
- 28 GRADOS	2,8899	0,2806	0,0569	0,1468

Tabla 16 Modelos del proceso de cabeceo.

En la Tabla 17 se recogen los cinco controladores diseñados por cancelación de los polos complejos.

ZONA REGULADOR	Kr	Ti	Td
25 GRADOS	0,345	0,1675	1,745
15 GRADOS	0,318	0,105	1,722
0 GRADOS	0,162	0,053	2,121
- 15 GRADOS	0,112	0,035	2,502
- 28 GRADOS	0,092	0,032	2,464

Tabla 17 Reguladores proceso cabeceo.

Se puede apreciar la variación en los parámetros de los controladores desarrollados para el control de la inclinación del helicóptero, se puede concluir que se ha procedido de manera correcta al dividir el rango de funcionamiento en zonas.

6 PROCESO DE ROTACIÓN.

El proceso de rotación se controlará con un único regulador, por lo que se realizará un ensayo válido y se modelará a partir de éste.

El proceso de rotación presenta un integrador, por lo que al aplicar un escalón en la variable manipulada, la variable controlada realiza una pendiente hasta saturar. El sistema es relativamente rápido, por lo que la realización de un ensayo válido se complica para este proceso.

6.1 ENSAYO DE ROTACIÓN.

El procedimiento para realizar el ensayo es: se ha parado el motor y se ha realizado un escalón del 15% en la tensión del rotor. Una vez llega a saturar, se realiza un escalón en -15%.

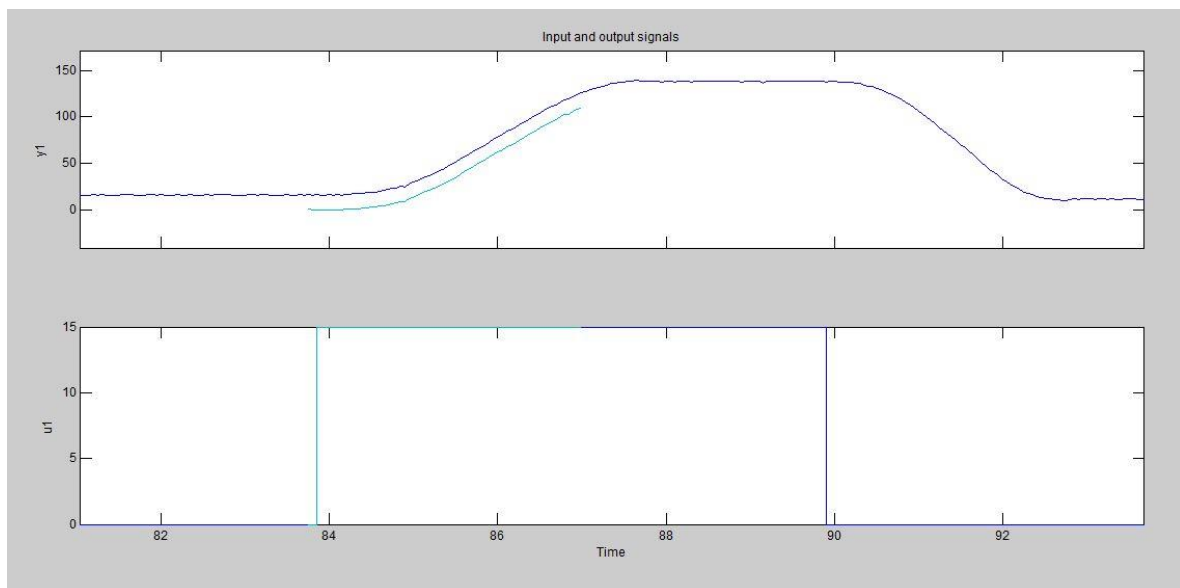


Figura 19 Ensayo rotor.

Una vez normalizados los datos se procede a la identificación.

6.2 MODELADO DE LA ROTACIÓN.

Una vez tenemos un ensayo válido para modelar, procedemos a la identificación del proceso de rotación, se plantearán tres modelos posibles:

- Primer orden con integrador: El sistema posee la dinámica de este modelo.
- Primer orden con integrador y con retardo: Se aprecia algo de retardo en las gráficas de los datos.
- Segundo orden con integrador y retardo.

En la Figura 20 se puede ver la validación de los tres modelos identificados. El modelo que menos se ajusta es el que presenta retardo. Por el contra, el modelo que más se ajusta es el modelo de primer orden. La solución adoptada, por tanto, para el modelado de rotación es un modelo de primer orden con integrador.

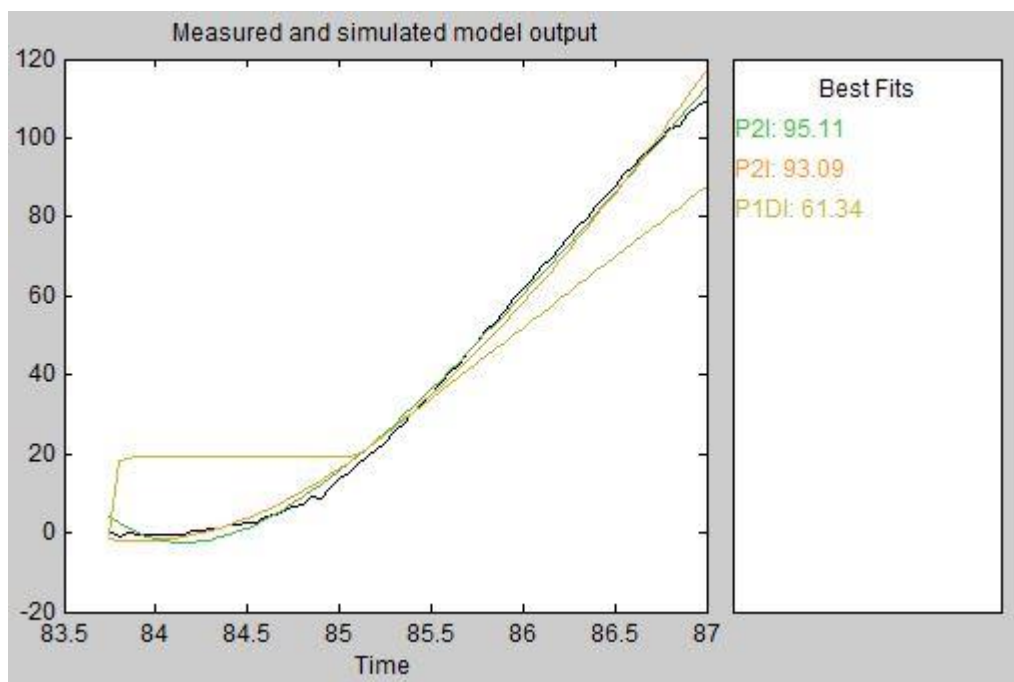


Figura 20 Validación de los modelos de rotación.

La función de transferencia del sistema es:

$$G(s)_{rotación} = \frac{Kp}{s(Tp + s)}$$



Se realizan dos identificaciones: una para la rampa de subida y otra para la rampa de bajada. Para obtener el modelo del sistema se realiza la media de estos modelos.

	Kp	Tp
Subida	11,3	0,766
Bajada	12,0	1,096
Modelo rotor	11,6	0,931

Tabla 18 Modelo de rotación.

En la Tabla 18 se pueden ver los parámetros identificados mediante la rampa de subida y la de bajada. Como se puede ver, los parámetros varían significativamente, pero no demasiado. La función de transferencia del modelo es:

$$G(s)_{rotación} = \frac{11.6}{s(0.931 + s)}$$

El modelo final es un primer orden con integrador.

6.3 DISEÑO DEL REGULADOR DE LA ROTACIÓN.

Se procede al diseño del regulador, una vez se tiene el modelo para la rotación del helicóptero. Como el modelo ha sido un primer orden con integrador, para el control del mismo será suficiente con un PD, ya que el error nulo en régimen permanente lo asegura el integrador del sistema.

Se diseñan dos reguladores como posibles soluciones para controlar la rotación:

- Cancelación del polo estable del sistema.
- Sin cancelación del polo del sistema.

El lugar de las raíces del que parte el sistema es el mostrado en la Figura 20.

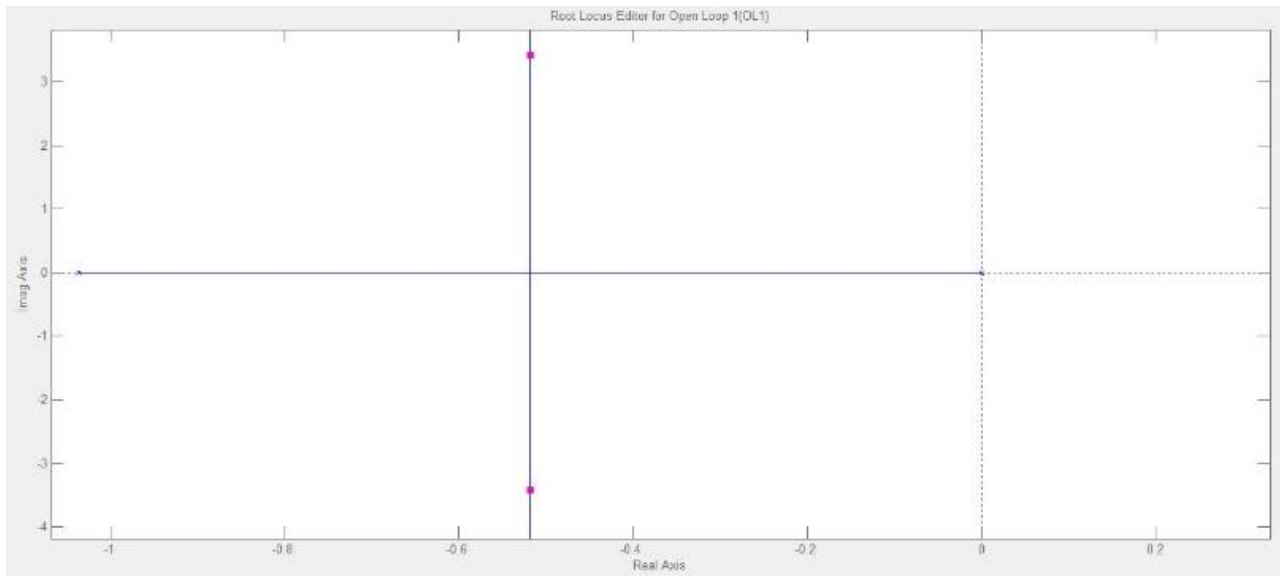


Figura 21 Lugar de las raíces modelo rotación.

La fórmula del regulador PD es:

$$PD_{rotor} = Kr(1 + Tds) = K(s + b)$$

Los reguladores diseñados por cancelación y sin cancelación son los indicados en la Tabla 19.

	Kr	Td
con cancelación	0.9	0.766
sin cancelación	0.45	1.096

Tabla 19 Controladores de rotación.

La validación de los controladores se realiza sobre la maqueta, el regulador que mejor se comporta es el diseñado sin cancelación.

El PD que controla el movimiento de rotación es:

$$PD_{rotor} = 0.45(1 + 1.096s)$$

7 CÁLCULOS PARA EL USO DEL JOYSTICK.

Para el uso del joystick no son necesarios demasiados cálculos. Basta con calcular las rectas de conversión del valor proporcionado por el joystick a los grados de inclinación o rotación.

El cálculo ha sido inmediato, ya que los dos sensores usados dan un valor simétrico de ± 3200 en todo el rango de movimiento. Por lo que, al ser simétricos los rangos de funcionamiento del helicóptero, basta con realizar la división de los límites para obtener la recta de conversión. Se pueden ver en la Figura 22.

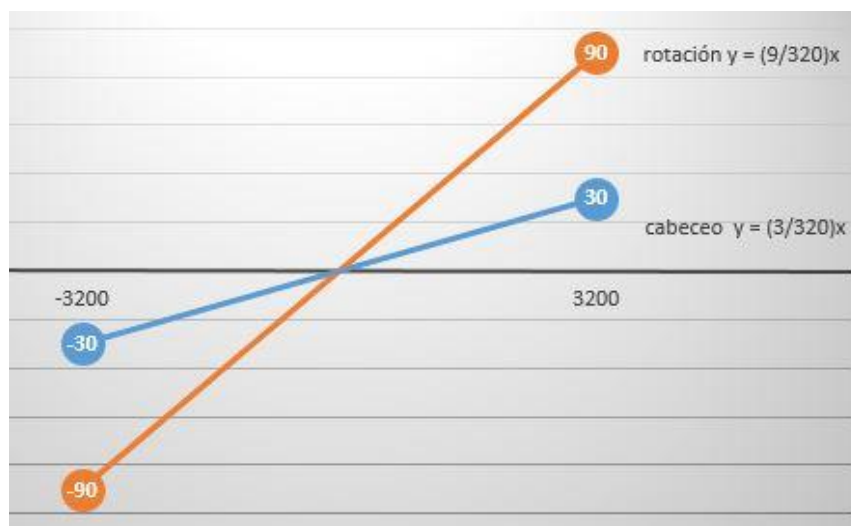


Figura 22 Rectas de conversión de los sensores del joystick.

La recta que se utilizará para el sensor de cabeceo será $Y = (3/320) \cdot X$ y la del sensor de rotación será $Y = (9/320) \cdot X$. Éstos son los únicos cálculos necesarios para la utilización del joystick.



8 BIBLIOGRAFÍA.

Instruments, N. (s.f.). *Software de Desarrollo de Sistemas NI LabVIEW*. Obtenido de
<http://www.ni.com/labview/esa/>

J. L: Guzmán, R. c. (1996). *Sistemas de Control Automatico*. Prentice Hall.

K. J. Aström, T. (2009). *Control PID Avanzado*. Prentice hall.



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



ESCUELA TÉCNICA
SUPERIOR INGENIEROS
INDUSTRIALES VALENCIA

TRABAJO FIN DE GRADO EN INGENIERÍA EN TECNOLOGÍAS INDUSTRIALES

DESARROLLO DE UNA APLICACIÓN PARA IDENTIFICACIÓN Y CONTROL DE UN HELICÓPTERO MEDIANTE LABVIEW

DOCUMENTO Nº 6: ANEXO DE PROGRAMACIÓN

AUTOR: PÉREZ JAÉN, JUAN

TUTOR: HERRERO DURÁ, JUAN MANUEL

Curso Académico: 2013-14



INDÍCE DE CONTENIDO

1 INTRODUCCIÓN Y OBJETIVOS DEL ANEXO.....	6
2 DISEÑO Y PROGRAMACIÓN PREVIA.....	7
2.1 APLICACIÓN DE ENSAYOS DE CABECEO.....	7
2.2 APLICACIÓN DE ENSAYOS ROTACIÓN.....	9
2.3 LECTURA Y ESCRITURA DE LA TARJETA.....	10
2.3.1 LECTURA GRADOS INCLINACIÓN DE CABECEO.....	10
2.3.2 LECTURA GRADOS DE ROTACIÓN.....	11
2.3.3 ESCRITURA SALIDA DE CABECEO.....	12
2.3.4 ESCRITURA SALIDA DE ROTACIÓN.....	12
2.4 ENCENDIDO Y APAGADO.....	13
2.4.1 BUCLE DE ENCENDIDO DEL MOTOR DE CABECEO.....	13
2.4.2 BUCLE DE APAGADO DE LOS MOTORES.....	15
2.5 ELIMINACIÓN ZONA MUERTA MOTOR ROTACIÓN.....	18
3 PROGRAMACIÓN DEL PID EN LABVIEW.....	19
3.1 PID CONTROL DE CABECEO.....	20
3.2 PID CONTROL ROTACIÓN.....	22
4 IMPLEMENTACIÓN DEL PLANIFICADOR DE GANANCIA EN EL CABECEO.....	23
5 PERFIL SETPOINT EN LA REFERENCIA.....	26
6 PROGRAMACIÓN JOYSTICK.....	28
6.1 CONFIGURACIÓN.....	28
6.2 BUCLE DE LECTURA DEL JOYSTICK.....	30
6.3 PROGRAMACIÓN DE LA LÓGICA DE CONTROL.....	31



6.4 COMUNICACIÓN TCP.....	32
6.4.1 PROGRAMA SERVIDOR.....	32
6.4.2 PROGRAMA CLIENTE.	32
7 PROGRAMA FINAL DE LA APLICACIÓN.....	33
7.1 CÁLCULO DE LA REFERENCIA.....	34
7.2 ACCIÓN DE CONTROL.	35
8 BIBLIOGRAFÍA.	36



INDÍCE DE FIGURAS

Figura 1 Código aplicación cabeceo.	7
Figura 2 Panel control aplicación cabeceo.	8
Figura 3 Código aplicación rotación.	9
Figura 4 Código lectura posición cabeceo.	10
Figura 5 Lectura cabeceo.vi	10
Figura 6 Código lectura rotor	11
Figura 7 Lectura rotor.vi	11
Figura 8 Código de salida del cabeceo.	12
Figura 9 Código de salida rotor.	12
Figura 10 Inicio cabeceo.	13
Figura 11 Código inicio maqueta.	14
Figura 12 Bucle de inicio.	14
Figura 13 Bucle para del programa.	15
Figura 14 Parada_cabeceo.vi	15
Figura 15 Código parada_cabeceo.vi	16
Figura 16 Perfil parada cabeceo.	16
Figura 17 Parada_rotor.vi	17
Figura 18 Código parada_rotor.vi	17
Figura 19 Perfil parada rotor.	17
Figura 20 Código para la eliminación de la zona muerta del rotor true y false.	18
Figura 21 Bucle de control.	19
Figura 22 Ecuaciones PID discretizadas.	19
Figura 23 Aspecto final del PID.vi	20
Figura 24 Código PID en LabVIEW	21
Figura 25 PID control rotación.	22
Figura 26 Evolución de K_r en función de los grados de inclinación.	23
Figura 27 Interpolador.vi	24
Figura 28 Código del interpolador.	25
Figura 29 NI_Gmath:interpolate 1D.vi	25



Figura 30 svi-perfil.vi	26
Figura 31 Código del perfil.	26
Figura 32 Panel de control del perfil.	27
Figura 33 Perfil programado.	27
Figura 34 Joystick.	29
Figura 35 Bucle de lectura del joystick.	30
Figura 36 Lógica control joystick.	31
Figura 37 Diagrama de funcionamiento de la aplicación.....	33
Figura 38 Código para el cálculo de la referencia.	34
Figura 39 Diagrama del cálculo de la referencia.	35
Figura 40 Diagrama de control de la aplicación.	35



1 INTRODUCCIÓN Y OBJETIVOS DEL ANEXO.

En el presente documento se va a explicar y detallar la programación realizada en LabVIEW, de tal modo que un programador pueda seguir el funcionamiento y desarrollo de la aplicación. No se realizará una introducción a LabVIEW, ya que el presente documento va dirigido a programadores que posean las nociones básicas de LabVIEW.

LabVIEW, al ser un programa de programación gráfica, no precisa del conocimiento de C ni de ningún lenguaje de programación típico, pero para la aplicación que se describe en este anexo se ha empleado C en diversos Formula Note de LabVIEW, por lo que será necesario el conocimiento básico de C.

El objetivo del presente documento es que un programador sea capaz de entender el programa y, si es necesario, modificarlo. Cabe mencionar que el código de LabVIEW puede llegar a ser algo caótico, por eso en la aplicación se han definido variables locales y eliminado el mayor número de cables posibles. En el presente documento se comenta por partes el código, haciendo referencia a las variables locales definidas.

Del mismo modo que es necesario el conocimiento básico de LabVIEW y C, serán igualmente necesarias unas nociones básicas de control. Se han implementado diversas técnicas de control, como pueden ser los dos PID's implementados mediante código C, en un Formula Note de LabVIEW. La explicación desarrollada puede quedar algo escueta para el programador, si éste no tiene las nociones básicas de control por ordenador.

2 DISEÑO Y PROGRAMACIÓN PREVIA.

Previamente al diseño de la aplicación principal se han desarrollado unas aplicaciones para la realización de los ensayos y se han diseñado unos bucles de encendido y apagado de la maqueta de simulación.

2.1 APLICACIÓN DE ENSAYOS DE CABECEO.

En primer lugar, se ha diseñado una aplicación que será la utilizada para la realización de los ensayos de cabeceo que proporcionarán los datos, para la identificación y posterior diseño de los reguladores.

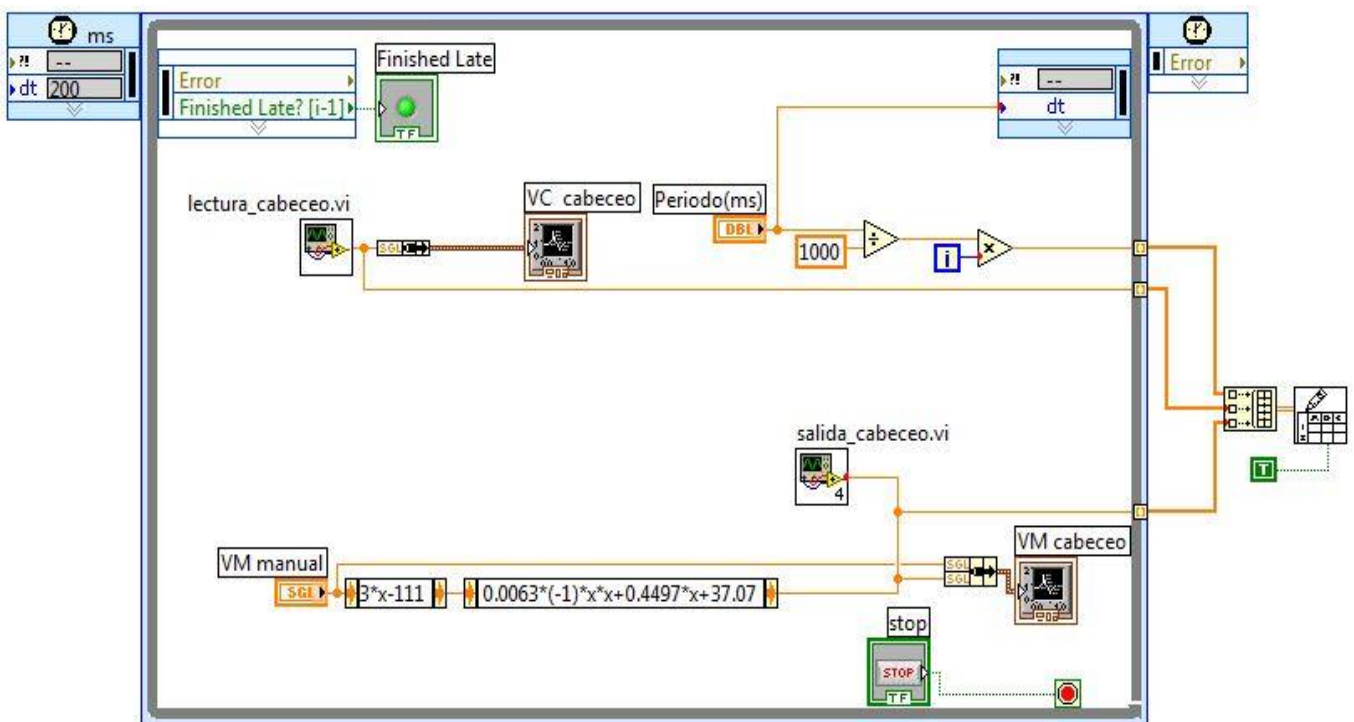


Figura 1 Código aplicación cabeceo.

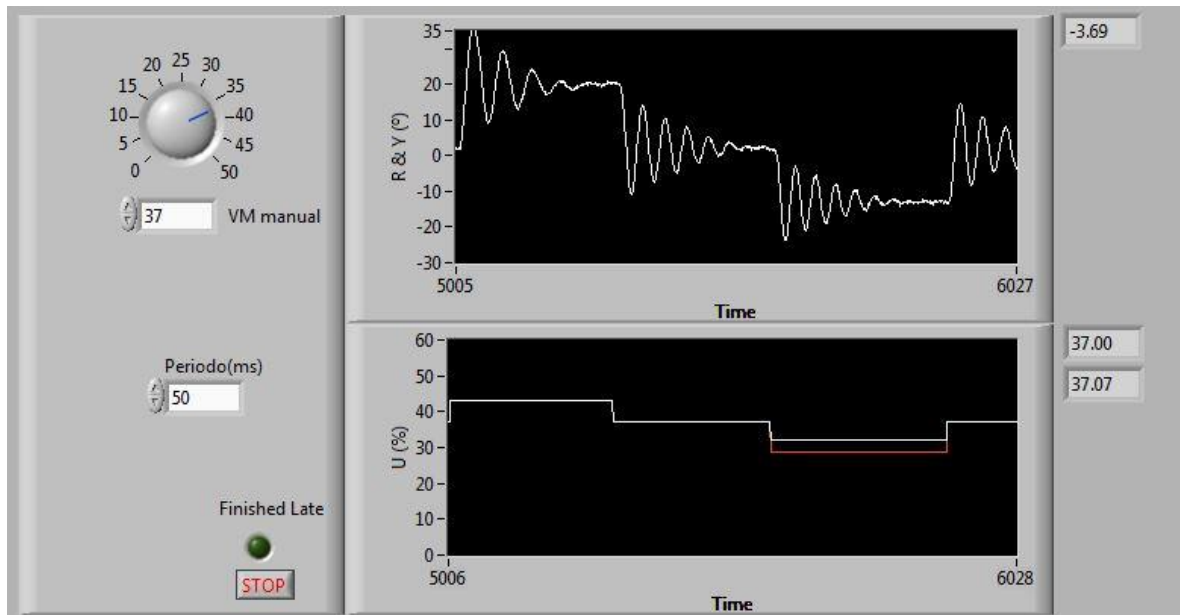


Figura 2 Panel control aplicación cabeceo.

En la Figura 1 y en la Figura 2 se pueden ver las pantallas de la aplicación. Se puede apreciar cómo se han usado las ecuaciones de linealización de la ganancia y los .vi diseñados para la lectura y escritura del cabeceo, el diseño de estos se detalla más adelante en este mismo documento.

El guardado de los datos necesarios del ensayo (t , VC y VM), se realiza mediante la composición de una matriz que tiene como columnas las variables guardadas, mediante el Build Array se crea la matriz. El posterior guardado de ésta con el nombre y lugar que el usuario decida se realiza mediante el "Write to Spreadsheet File.vi".

2.2 APLICACIÓN DE ENSAYOS ROTACIÓN.

La aplicación para el ensayo necesario para identificar el modelo de rotación se realizará del mismo modo que la aplicación para los ensayos de inclinación.

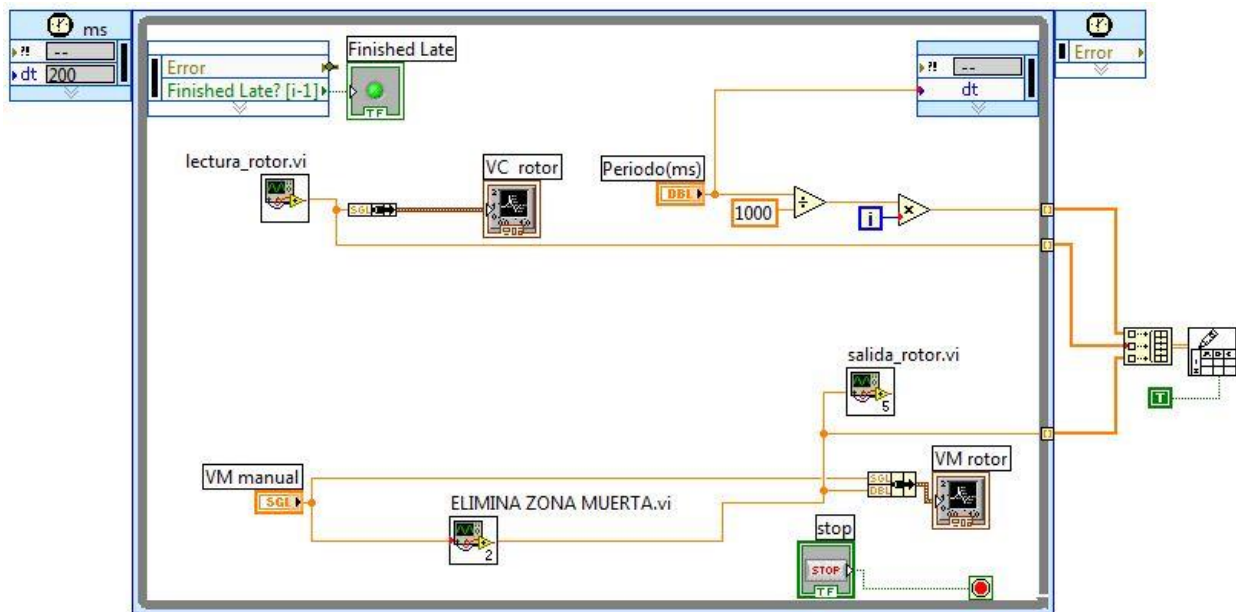


Figura 3 Código aplicación rotación.

Se han modificado varias partes del código:

- Se ha sustituido las ecuaciones de linealización por el .vi de eliminación de la zona muerta del motor, detallado en el apartado 2.5 de este mismo anexo.
- Los .vi de lectura y escritura se han sustituidos por los correspondientes para el rotor.

El panel de control es similar al de la aplicación de cabeceo, ya que no es necesario realizar ningún cambio en el panel de control, únicamente se ha modificado el código.

2.3 LECTURA Y ESCRITURA DE LA TARJETA.

Para simplificar el código de LabVIEW, ya que se utilizará en diversas ocasiones, se han desarrollado cuatro .vi para la lectura de los sensores y la escritura en las salidas de la tarjeta de adquisición, que están configuradas para nuestra aplicación. Esto quiere decir que tienen implementadas las rectas de conversión y definidos los canales.

2.3.1 LECTURA GRADOS INCLINACIÓN DE CABECEO.

Este es un .vi que nos da el valor de la inclinación en grados. En la Figura 4 se muestra el código, donde se puede apreciar la simplicidad del mismo.

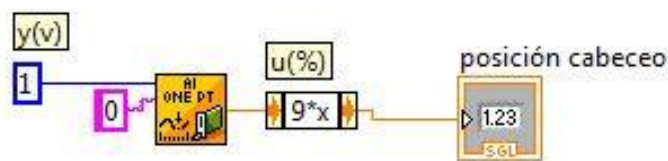


Figura 4 Código lectura posición cabeceo.

En la Figura 4 se puede ver como el .vi proporciona la posición de inclinación en grados del helicóptero. La recta de conversión es la calculada para el ángulo de inclinación y el canal de lectura es el 0, como se indica en el apartado 1.5.3 Tarjeta de adquisición de datos de la memoria.

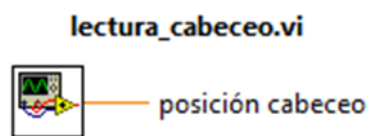


Figura 5 Lectura cabeceo.vi

2.3.2 LECTURA GRADOS DE ROTACIÓN.

Al igual que el apartado anterior, se puede apreciar la simplicidad del código. Este es un .vi que nos proporciona el valor de la rotación en grados.

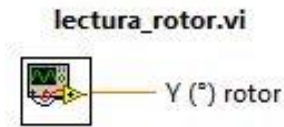


Figura 7 Lectura rotor.vi

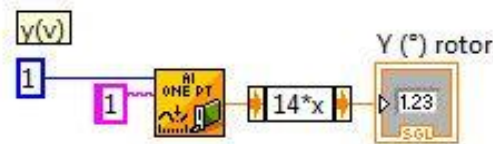


Figura 6 Código lectura rotor

Como se puede ver en la Figura 6. La ecuación de conversión implementada es la calculada para el sensor de rotación, y el canal de lectura es el 1, que es donde se ha conectado la salida del sensor de rotación.

Como se puede apreciar en los dos últimos casos, la complejidad del código es casi inexistente, lo que hace pensar que no es necesario realizar el encapsulado. Sin embargo, se ha tomado la decisión de encapsular todo lo posible con la finalidad de facilitar al máximo la lectura e interpretación del código de la aplicación final.

2.3.3 ESCRITURA SALIDA DE CABECEO.

Para la escritura de la tensión de cabeceo en la tarjeta de adquisición, se ha generado el siguiente vi.

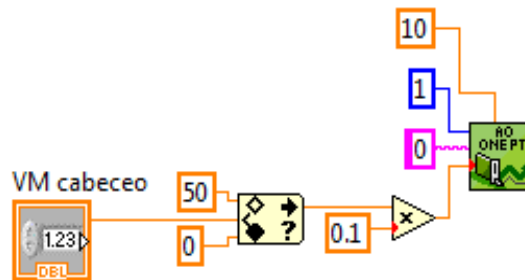


Figura 8 Código de salida del cabeceo.

En la Figura 8 se puede ver el código que se ha encapsulado en “salida cabeceo.vi”, en él se puede apreciar cómo la señal VM se limita entre 0 y 50. También se puede ver cómo se convierte de decivoltios a voltios multiplicando por 0.1. Se procede a escribir en la salida 1 (canal 0) de la tarjeta, que es donde se ha conectado el motor de cabeceo.

2.3.4 ESCRITURA SALIDA DE ROTACIÓN.

En la Figura 9 se puede ver en código de escritura de la salida de rotación, en él se puede ver como la señal se limita entre ± 50 y se pasa de decivoltios a voltios.

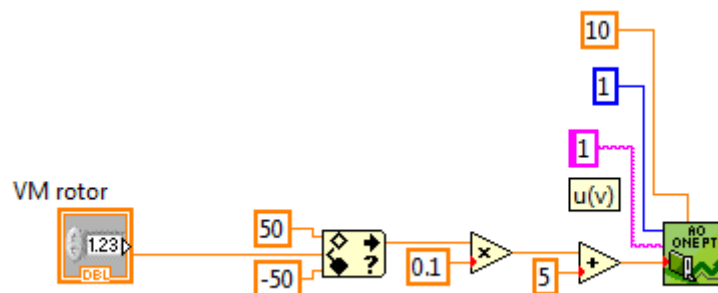


Figura 9 Código de salida rotor.

Se puede ver cómo a la señal se le suma 5, como se detalla en el apartado 1.5.3 de la memoria, esto es porque en la tarjeta de adquisición de datos el rotor se conecta a la salida analógica 2 de $\pm 10V$, para escribir en esa salida de la tarjeta hay que escribir en el canal 1 y el valor de 0V equivale a escribir 5V.

2.4 ENCENDIDO Y APAGADO.

Para el correcto control de la maqueta, se ha decidido la implementación de un bucle de encendido que se ejecutará antes del bucle principal y un bucle de apagado que se ejecutará una vez finaliza éste.

2.4.1 BUCLE DE ENCENDIDO DEL MOTOR DE CABECEO.

La finalidad de este bucle es la de poner la maqueta en la posición de equilibrio antes de la puesta en funcionamiento del bucle principal, para que la puesta en marcha sea suave y controlada.

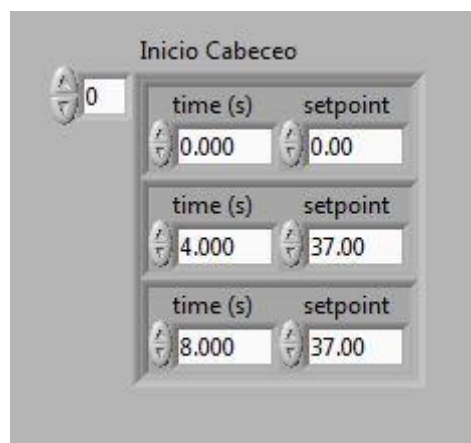


Figura 10 Inicio cabeceo.

Este bucle solamente pone en funcionamiento el motor de cabeceo, llevando a éste a su posición de equilibrio. Lo hace mediante la implementación de un perfil en la tensión del motor. Como se puede ver en la Figura 10. El perfil tiene una duración de ocho segundos y consta de dos partes:

- La primera describe una rampa en la tensión del motor que va desde 0V a 3.7V y tiene una duración de cuatro segundos.
- La segunda parte del perfil es una recta de valor 3.7V para dar tiempo a la maqueta a que se estabilice.

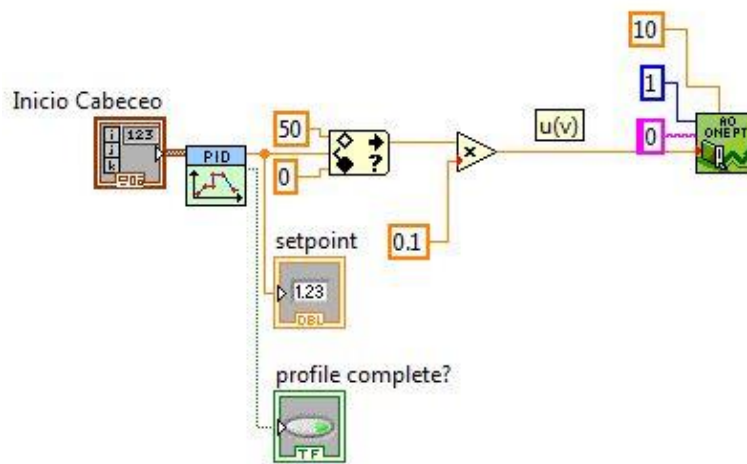


Figura 11 Código inicio maqueta.

En la Figura 11 se puede ver como se ha implementado el perfil en la tensión de cabeceo mediante el uso de Perfil Profiel.vi disponible en la toolbox de PIDs de LabVIEW.



Figura 12 Bucle de inicio.

En la Figura 12 se puede ver como el bucle finaliza cuando el perfil se ha completado. Cabe mencionar que este bucle pasa al principal el valor de la u_0 , necesaria en el PID de control del cabeceo.

2.4.2 BUCLE DE APAGADO DE LOS MOTORES.

En el bucle de apagado de los motores de la maqueta, se toma el último valor de la tensión de los motores y se realiza un perfil de la referencia entre ese valor y cero, el bucle no finaliza hasta que los dos motores están parados.

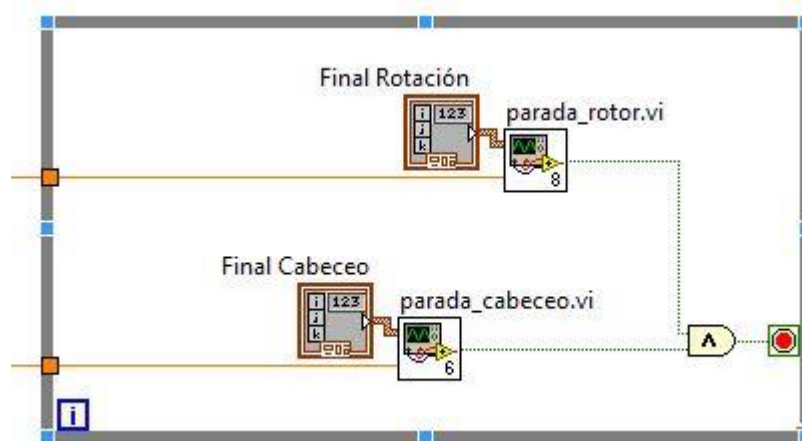


Figura 13 Bucle para del programa.

En la Figura 13 se puede ver como el bucle de parada del programa está compuesto por dos .vi, uno para la parada del motor de cabeceo y otro para la parada del motor de rotación. Hay que pasar a estos .vi el último valor de la tensión del motor correspondiente y el bucle finalizará cuando los dos motores estén parados.

Para la parada del motor de cabeceo es necesario pasar un Cluster con el perfil que se desea hacer y el último valor de VM.



Figura 14 Parada_cabeceo.vi

Para realizar el perfil entre el último valor de VM y cero es necesario introducir el último valor de VM en la primera posición del Cluster, para ello se recurre a las operaciones necesarias en LabVIEW. El perfil se implementa mediante un perfil de la referencia, de la misma manera que se implementó el perfil de encendido.

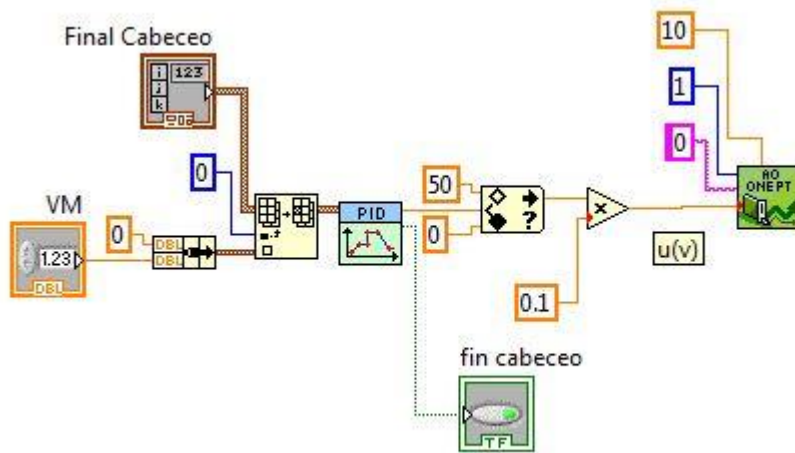


Figura 15 Código parada_cabeceo.vi

El perfil que se ha programado es una pendiente de cuatro segundos entre el último valor de VM y cero, más una recta de dos segundos en cero. En la Figura 16 se puede ver como se han definido los puntos del perfil, a falta del primer punto que se introduce mediante el código.

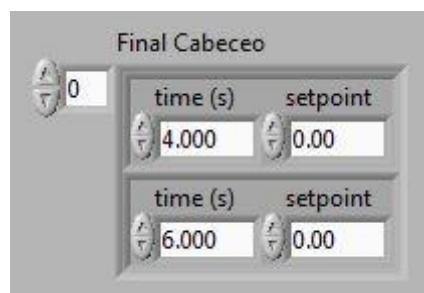


Figura 16 Perfil parada cabeceo.

Para la parada del motor de rotación de la maqueta de simulación se ha procedido de forma similar, es necesario pasar a *parada_rotor.vi* el último valor de VM del rotor.

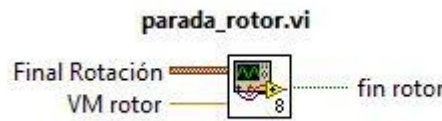


Figura 17 Parada_rotor.vi

El código de LabVIEW es similar al empleado para la parada del motor de cabeceo de la maqueta.

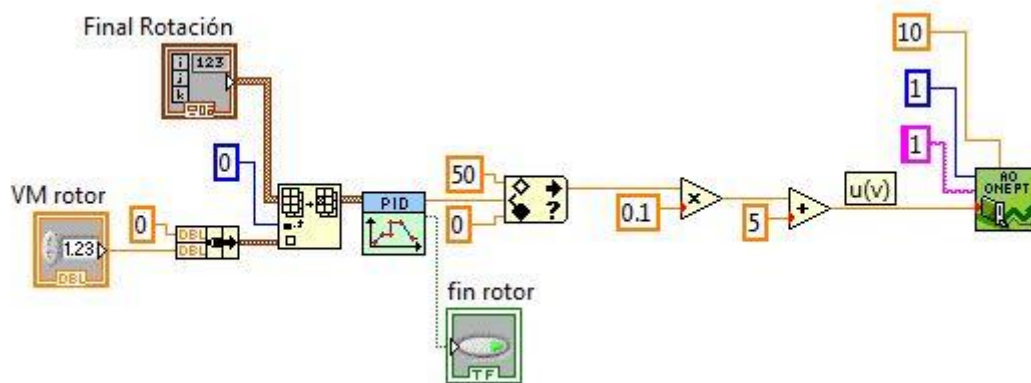


Figura 18 Código parada_rotor.vi

El perfil es una pendiente de cuatro segundos entre el último valor de VM rotor y cero, más una recta de dos segundos en cero. En la Figura 19 se puede ver como se ha definido los puntos del perfil, a falta del primer punto que se introduce en el código.

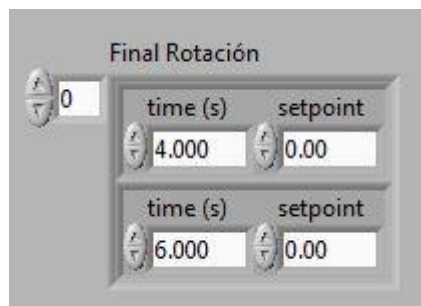


Figura 19 Perfil parada rotor.

2.5 ELIMINACIÓN ZONA MUERTA MOTOR ROTACIÓN.

En el motor de rotación se ha observado la existencia de una zona muerta en la que éste no funciona, se ha decidido eliminarla. Esta zona para la cual el motor está parado está comprendida entre -0.5V y 1.4V. Para la eliminación del intervalo se ha implementado un .vi con esa finalidad.

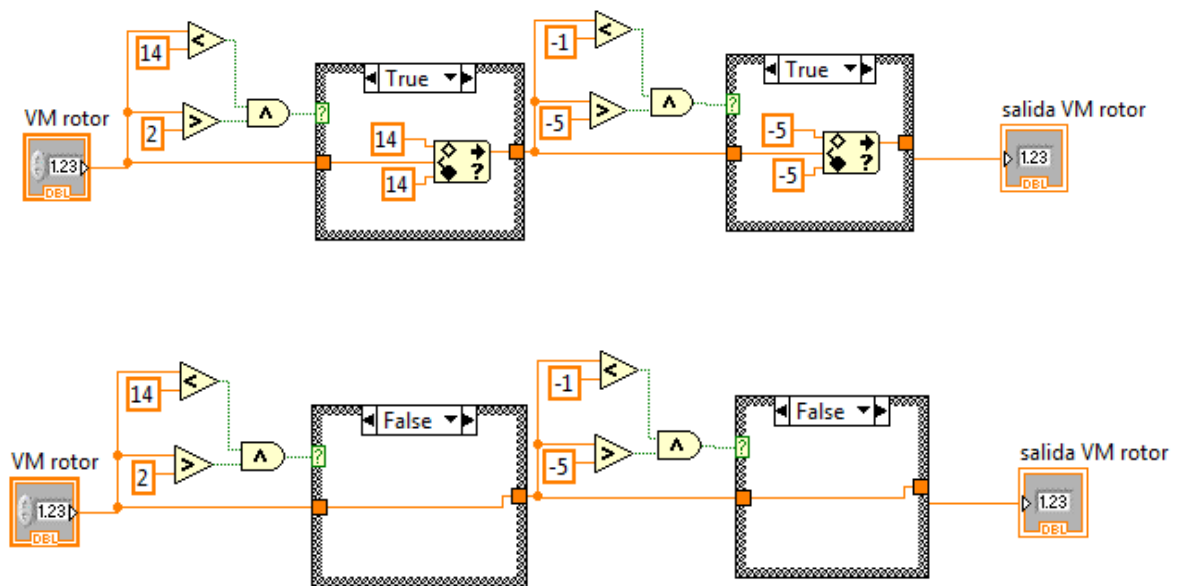


Figura 20 Código para la eliminación de la zona muerta del rotor true y false.

La eliminación de la zona se consigue mediante la implementación de dos case Structure:

- Si la tensión está comprendida entre 0.2V y 1.4V, se aplica una tensión de 1.4V.
- Si la tensión está comprendida entre -0.1V y -0.5V la tensión aplicada es 0.5V.
- Si la tensión está comprendida entre 0V y 0.1V se deja apagado el motor.

La eliminación de la zona muerta permite un mejor funcionamiento del motor de rotación, ya que disminuye la inercia del movimiento de rotación originada por la existencia de la misma.

3 PROGRAMACIÓN DEL PID EN LABVIEW.

Se ha optado por la programación de nuestro propio PID y no utilizar el que trae por defecto la toolbox de PIDs de LabVIEW, para poder controlar mejor el funcionamiento del mismo.

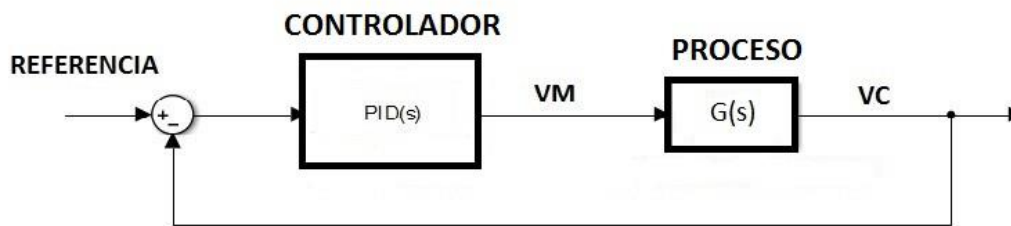


Figura 21 Bucle de control.

En la Figura 21 podemos ver el bucle de control de un PID. La ecuación programada en el código del PID es la ecuación discretizada, ya que para la ejecución por ordenador únicamente se pueden programar ecuaciones discretas.

$$U(t) = K_c \left(E(t) + \frac{1}{T_i} \int_0^t E(\tau) d\tau + T_d \frac{dE(t)}{dt} \right)$$

$$U_p(t) = K_c E(t)$$

$$U_i(t) = K_c \frac{1}{T_i} \int_0^t E(\tau) d\tau$$

$$U_d(t) = K_c T_d \frac{dE(t)}{dt}$$

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

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

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

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

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

Figura 22 Ecuaciones PID discretizadas.

3.1 PID CONTROL DE CABECEO.

Se ha procedido a la programación de un PID en LabVIEW. Para ello se ha dispuesto un encapsulado de todo el código dentro de un .vi que más tarde será usado en el código de la aplicación. A este .vi se han de pasar los parámetros necesarios para el cálculo de la acción de control. Estos parámetros son:

- El valor de la variable controlada en grados (VC).
- El valor de la variable manipulada si se desea controlar en manual (VM manual).
- Modo de funcionamiento, manual o automático (Man Auto).
- El valor de la VM anterior, para iniciar la acción de control (u_0),
- Los parámetros del controlador PID (K_r , T_i y T_d).
- Periodo de ejecución (t_s).
- El valor de la referencia (SP).
- El valor de la acción de control es la salida VM en decivoltios.

Todos estos parámetros están mostrados en la Figura 23.

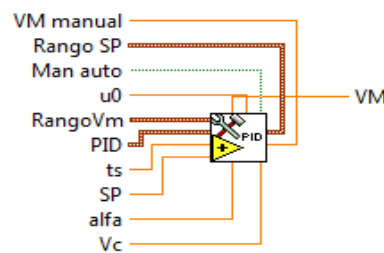


Figura 23 Aspecto final del
PID.vi

En la Figura 24 se puede ver el código. Se puede apreciar como para el control es necesario el conocimiento de variables de ciclos de ejecución anteriores, que serán variables internas. En LabVIEW, al no disponer de direcciones de memoria física, las variables se pasarán mediante realimentación; las variables que han de ser realimentadas son: El error anterior, la acción u_0 (que se inicializa al valor de u_0 del bucle de inicio) y el valor de u_i anterior (acción integral).

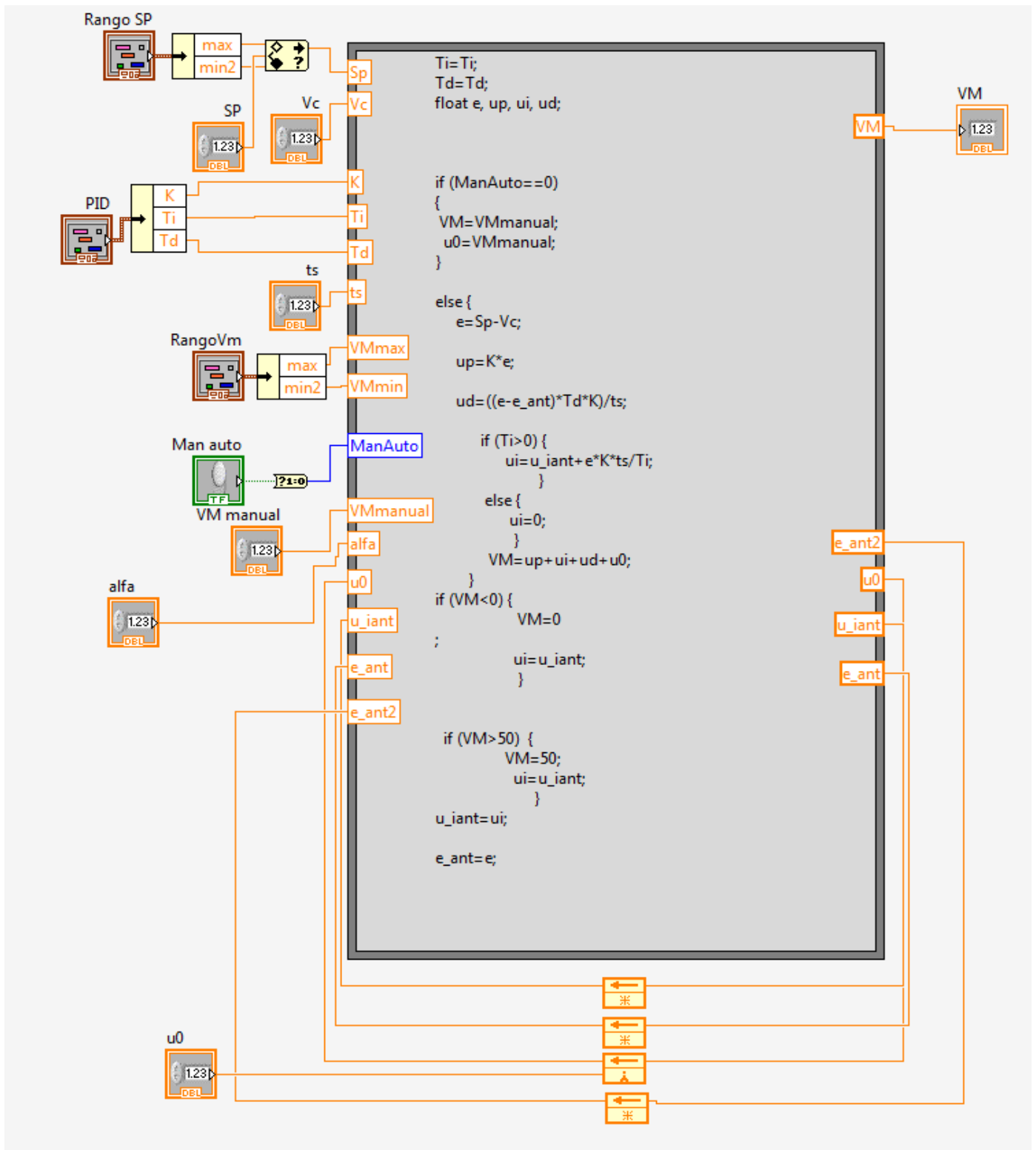


Figura 24 Código PID en LabVIEW

3.2 PID CONTROL ROTACIÓN

El PID que se ha implementado para el control del ángulo de cabeceo de la maqueta del helicóptero sirve también para el control de la rotación, con unas pequeñas modificaciones:

- Variando los límites del antiwindup: en el rotor, la señal satura para el valor de $\pm 5V$, cambiando el límite inferior del mismo se obtiene el controlador de la rotación.
- En este caso no es necesario para la variable u_0 , ya que el motor de rotación estará parado, antes de la puesta en marcha del mismo control o en modo manual pasando este por el mismo PID.

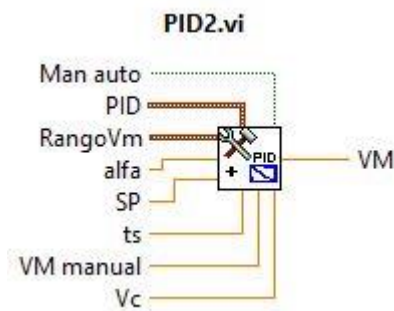


Figura 25 PID control rotación.

La salida el PID2 se conecta al .vi de eliminación de la zona muerta.

4 IMPLEMENTACIÓN DEL PLANIFICADOR DE GANANCIA EN EL CABECEO.

Como se ha detallado en la memoria, en el cabeceo se implementa un planificador de la ganancia para variar los parámetros del PID en función del ángulo de cabeceo, ya que el modelo varía significativamente, por lo que el controlador debe variar también. El rango de funcionamiento se ha dividido en seis, para esta división es necesario el diseño de cinco reguladores. El cálculo y diseño de los mismos se detalla en el anexo de diseño. En este documento detallamos el procedimiento y código necesario para la implementación de dicha planificación en LabVIEW.

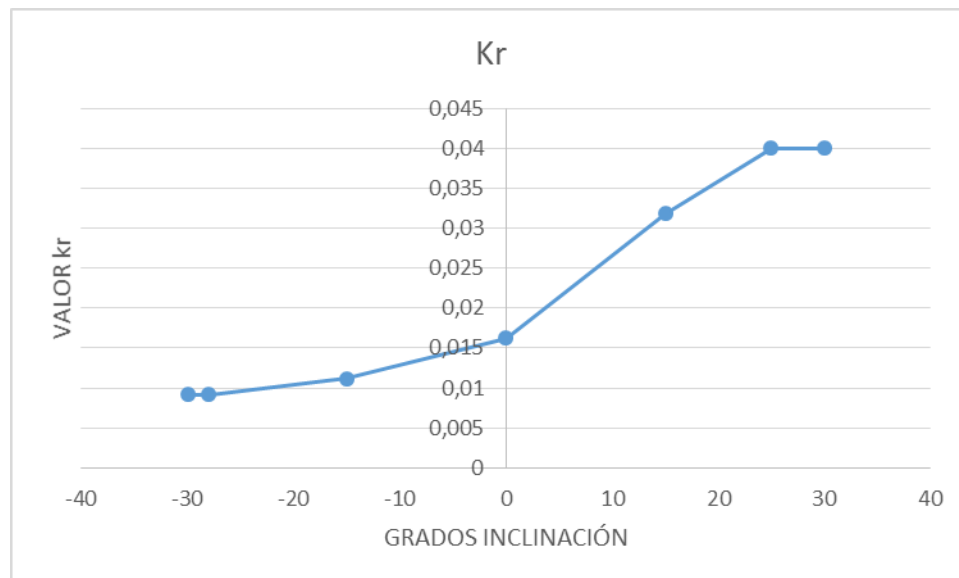


Figura 26 Evolución de Kr en función de los grados de inclinación.

En la Figura 26 se muestra un ejemplo de cómo evoluciona la constante del regulador en función de los grados de inclinación de la maqueta de simulación, se puede apreciar la necesidad del planificador de la ganancia.

ZONA REGULADOR	Kr	Ti	Td
25 GRADOS	0,354	0,105	1,722
15 GRADOS	0,318	0,105	1,722
0 GRADOS	0,162	0,053	2,121
(-) 15 GRADOS	0,112	0,035	2,502
(-) 28 GRADOS	0,092	0,032	2,464

Tabla 1 Reguladores proceso cabeceo.

En la Tabla 1 se muestra un resumen de los reguladores diseñados en el anexo de diseño. Se puede observar cómo hay un considerable cambio en la ganancia, mayor cuanto más se eleva la maqueta de simulación (llega a multiplicarse 2,5 veces), y en la posición de los polos, definida por el valor de Ti y Td.

En LabVIEW, al planificador de la ganancia se le ha llamado interpolador, ya que su función es tomar el valor actual de VC e interpolar el regulador adecuado para ese punto de funcionamiento.

Hay que pasar un Cluster con todos los reguladores y este nos devuelve un Cluster con los parámetros del regulador a utilizar y el valor de VC para el cual debe interpolar el valor del regulador. En la Figura 27 vemos el aspecto que toma el vi.



Figura 27 Interpolador.vi

En la Figura 28 se puede observar la composición del código de interpolación. En primer lugar, se extrae del Cluster de reguladores los vectores con los parámetros de éstos, se pasa cada vector a un *NI_Gmath:interpolate 1D.vi* que, tras pasarle el valor de VC, nos da un vector que, en la posición cero, tiene el valor del parámetro interpolado para el valor de VC de ese instante. Se unen los parámetros en un nuevo Cluster, que es el que se le pasará al PID.

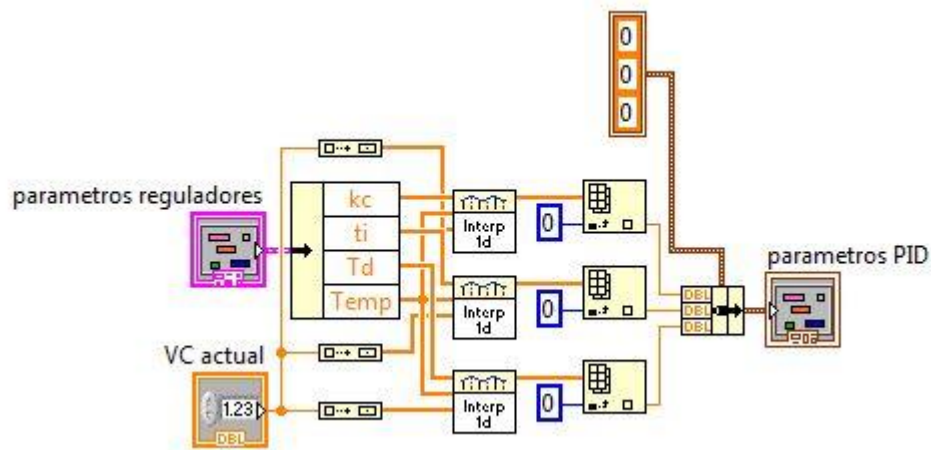


Figura 28 Código del interpolador.

En la Figura 29 vemos con más detalle el *NI_Gmath:interpolate 1D.vi* de LabVIEW. Este .vi tiene implementadas muchas más funciones de las que se han empleado en la interpolación lineal.

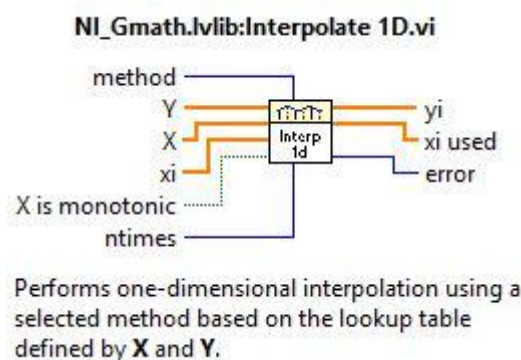


Figura 29 *NI_Gmath:interpolate 1D.vi*

5 PERFIL SETPOINT EN LA REFERENCIA

Se ha implementado un perfil Setpoint, éste modifica la referencia de los dos motores simultáneamente, dando la sensación de movimiento dinámico.

Para realizar el perfil se ha creado *svi-perfil.vi*. Como se puede ver en la Figura 30, es necesario pasar a este .vi los Cluster con los puntos del perfil, periodo de ejecución del bucle (ts), la señal de activación del perfil y, una vez se ha completado el perfil, la señal para reiniciar el perfil.

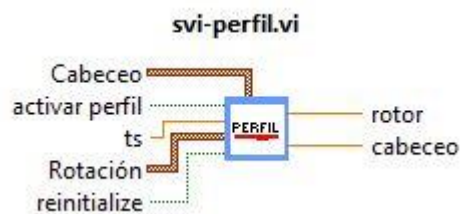


Figura 30 svi-perfil.vi

En el código de este .vi se puede apreciar cómo se ha implementado dentro de un case Structure para activar y desactivar el perfil en función del modo de funcionamiento en el que se esté. Una vez se ha completado el perfil, es necesario reiniciar para volver a ejecutarlo. En la Figura 31 se ve el código programado para el caso en el cual el perfil esté activado.

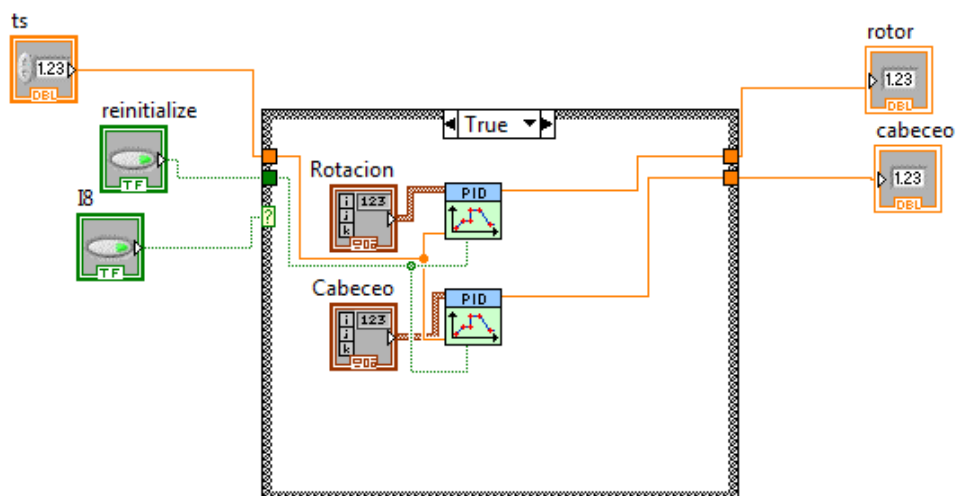


Figura 31 Código del perfil.

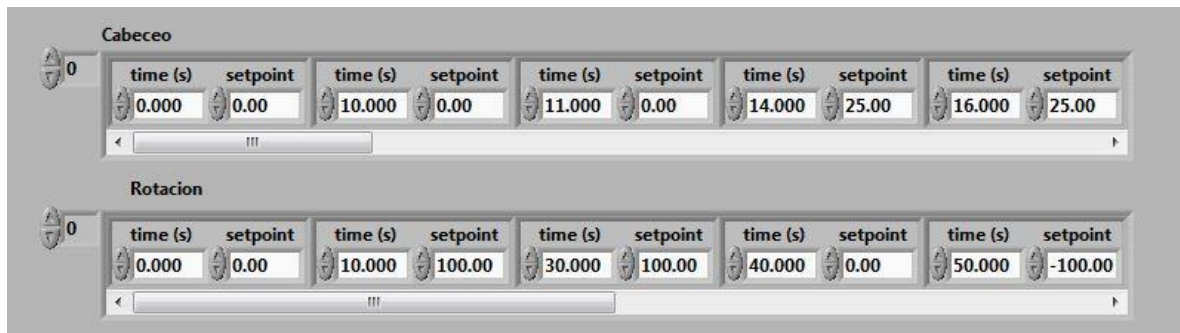


Figura 32 Panel de control del perfil.

Este .vi modifica el valor de la referencia en cada ciclo de ejecución, por eso es necesario pasar el valor de t_s . En la Figura 32 se pueden ver los puntos iniciales que componen el perfil, el usuario puede modificar fácilmente estos puntos para que el helicóptero describa el perfil que desee. Como se aprecia, hay dos Cluster de puntos, uno para el motor principal de inclinación y otro para el motor de rotación.



Figura 33 Perfil programado.

En la Figura 33 se puede ver el perfil que se ha implementado por defecto en la aplicación, en rojo tenemos el ángulo de rotación que describe el motor de rotación y en azul el ángulo de inclinación de la maqueta. Se puede ver que una vez ha cesado el movimiento de rotación comienza el movimiento de inclinación. El desarrollo del perfil completo tiene una duración cercana a los dos minutos.

6 PROGRAMACIÓN JOYSTICK

Se ha dispuesto de un joystick para el control del helicóptero, para el uso del mismo es necesario realizar una configuración previa, un cálculo de las rectas de conversión de los sensores y una selección de los sensores y pulsadores que se van a emplear. El joystick se colocará en el mismo equipo o en otro diferente, el cual se comunicará con el equipo que desarrolla el control mediante comunicación TCP. El funcionamiento del joystick debe ser el mismo independientemente de donde se encuentre, para ello se transmitirán por red el valor de la referencia para el control de inclinación y rotación, así como el valor de los pulsadores que se han empleado en la selección de modo de funcionamiento del mismo.

6.1 CONFIGURACIÓN.

Para el control de la maqueta con el joystick se seleccionan los sensores que se utilizarán, ya que el joystick dispone de sensores para los movimientos de los tres ejes y rotación en los mismos.

Como en la maqueta disponemos de dos grados de libertad, que son el cabeceo y la rotación, se han seleccionado en el joystick la inclinación del "eje y" por la inclinación del helicóptero y la rotación en el "eje z" para el movimiento de rotación de la maqueta.

Las rectas de conversión de tensión de los sensores a grados de inclinación y de rotación ya han sido calculadas en el anexo de diseño y son: para el sensor de inclinación del eje y es $y = (3/320) * x$ siendo x el valor proporcionado por el joystick e y el valor de la inclinación en grados. Para el sensor de rotación del eje z es $y = (9/320) * x$ siendo x el valor proporcionado por el joystick e y el valor de la rotación en grados.

La instalación del joystick en el equipo y el reconocimiento del mismo por LabVIEW es realmente sencillo, ya que LabVIEW lo reconoce inmediatamente una vez se han instalados los driver del joystick en el equipo¹. Una vez instalados los driver se procede a la programación del bucle de lectura del joystick que se utilizará en este mismo equipo o en otro para la comunicación TCP.



Figura 34 Joystick.

Además de la referencia de la inclinación y la rotación se ha implementado el uso de cuatro pulsadores del joystick, para permitir el cambio del modo de funcionamiento desde el joystick sin necesidad de interactuar con el equipo de control. Los cuatro pulsadores permiten cambiar a los cuatro modos de funcionamiento: control de la referencia con el joystick, modo perfil, que la maqueta describa el perfil programado, modo automático, en el cual la maqueta sigue la referencia introducida en el ordenador y por último, modo manual, en el cual el usuario introduce directamente el valor de la tensión que llegará a los motores. El valor de los pulsadores se transmitirá por TCP o, si está en el mismo equipo, por variables locales. En la Figura 34 se ve el joystick y los pulsadores que se han usado y la función que desempeñan.

¹ LabVIEW está preparado para trabajar con la mayoría del equipamiento de control comercial y lo reconoce sin problemas.

6.2 BUCLE DE LECTURA DEL JOYSTICK.

Se ha programado un bucle para la lectura del Joystick. Este bucle pasa por variables locales el valor de la referencia proveniente el joystick, ya en grados de inclinación y rotación, y el valor de los pulsadores de control. Como se puede ver en la Figura 35 el bucle se ejecuta con un periodo similar al bucle de control principal y se leen del joystick el valor de inclinación del “eje y” (Y axis) y el valor de la rotación del “eje z” (Z axis rotation). También se leen los valores de los cuatro pulsadores empleados en el control, que el joystick transmite en la posición 7, 8, 11 y 12 del vector compuesto por todos los pulsadores disponibles en el joystick.

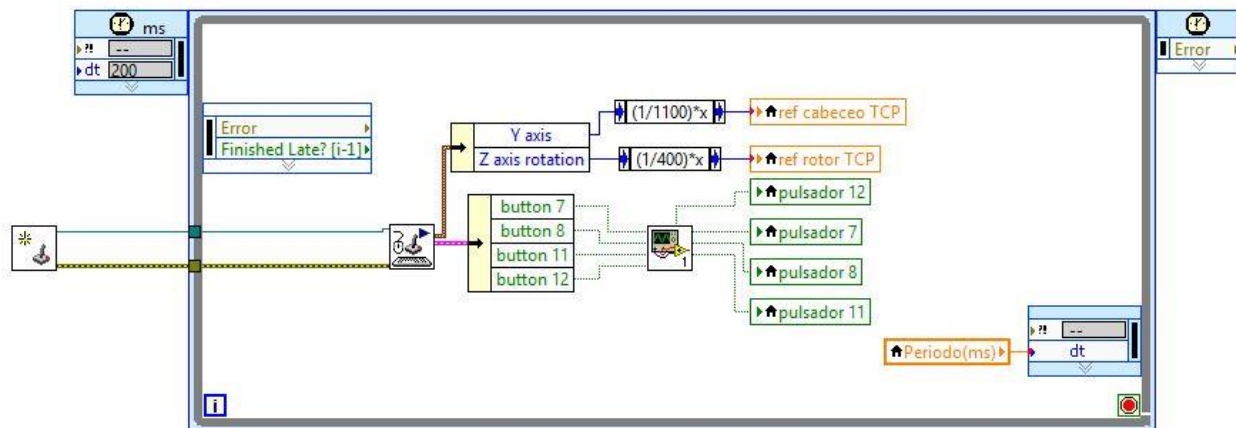


Figura 35 Bucle de lectura del joystick.

La lectura de los valores del joystick se realizará con el mismo bucle, solo que una vez se transmite el valor por variable local directamente al bucle principal de la aplicación y, en el caso de comunicación TCP, se transmite a un bucle que actúa como servidor para la transmisión de los mismos al cliente que se encuentra en la aplicación principal.

6.3 PROGRAMACIÓN DE LA LÓGICA DE CONTROL

Para el uso de los pulsadores nos encontramos con el problema de que son pulsadores y no podemos modificar la dinámica como se haría si fueran controles de LabVIEW. Para cambiar la dinámica y convertirlos en interruptores, aprovechando también para solventar el inconveniente de la posibilidad de que el usuario pulse varios controles a la vez, se ha implementado un fragmentado. Un .vi denominado *lógica_joystick.vi* que realiza ésta conversión y permite únicamente la activación de un control.

En la Figura 36 se puede ver cómo funciona la lógica de control del joystick: toma el valor de los pulsadores en el ciclo de ejecución, entra en el if del pulsador que esté pulsado y da el valor de 1 a ese pulsador y de 0 al resto, de tal manera que el último if en ejecutarse da el valor de 1 al pulsador correspondiente y 0 a los demás. No importa el número de pulsadores que estén pulsados, el último en ejecutarse pone a cero el resto de salidas. Luego, por lógica de LabVIEW, se convierte a salidas booleanas de control.

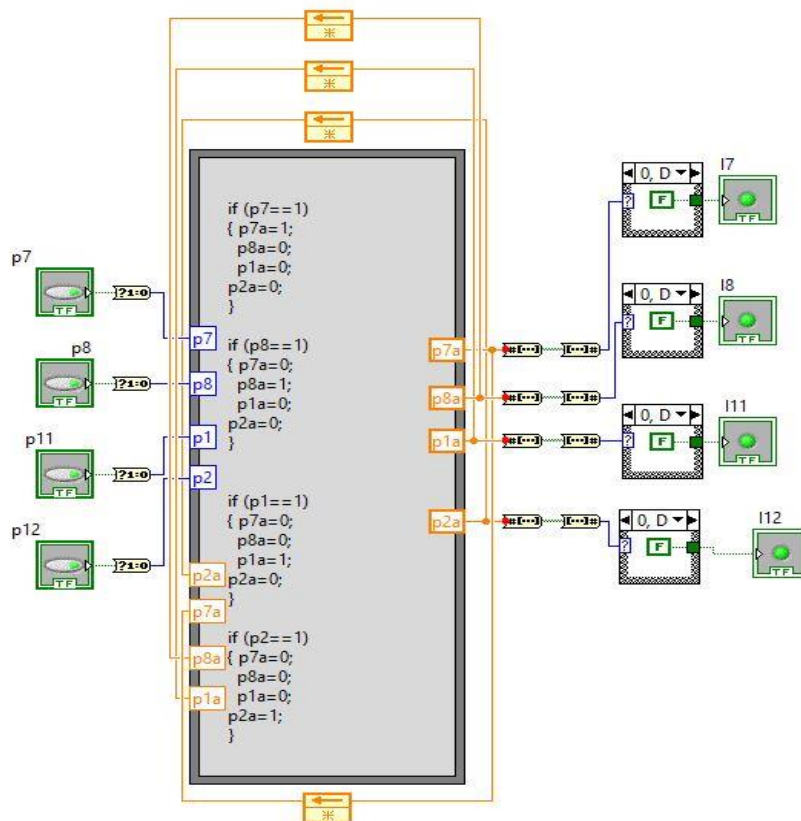


Figura 36 Lógica control joystick.



6.4 COMUNICACIÓN TCP.

Para comunicación TCP es necesario el desarrollo de dos pequeñas aplicaciones: una que funcione como servidor y transmita los valores que se desean transmitir y otra que funcione como cliente, que lea los valores que se han transmitido mediante el servidor. En el caso de la aplicación desarrollada, el servidor será en el que se encuentra el joystick y el cliente se encontrará en la aplicación principal.

6.4.1 PROGRAMA SERVIDOR.

Para el programa servidor, establece la comunicación a través del puerto del equipo que le indica el usuario. Una vez establecida la comunicación, el bucle transforma los caracteres que se desean transmitir en una cadena de bits y los transmite mediante la red Ethernet, en el programa cliente habrá que indicar la cantidad de caracteres que se han transmitido para que los descifre.

6.4.2 PROGRAMA CLIENTE.

El programa que desempeña la función de cliente lee del puerto del equipo identificado mediante la dirección IPE proporcionada al programa, por lo que es necesario el conocimiento de la dirección IPE del equipo que ejecuta el programa servidor. Una vez se ha leído la cadena de bits transmitida, hay que proceder a la descifrado de la misma, para ello hay que indicar la composición y número de datos transmitidos. En este caso se trata de un vector con seis datos, cada dato está compuesto por 8 bits, por lo que hay que indicar un total de 48 bits de lectura.

7 PROGRAMA FINAL DE LA APLICACIÓN.

El funcionamiento de la aplicación principal incorpora todo lo desarrollado hasta ahora en un único programa de LabVIEW. El funcionamiento global de la aplicación se detalla continuación.

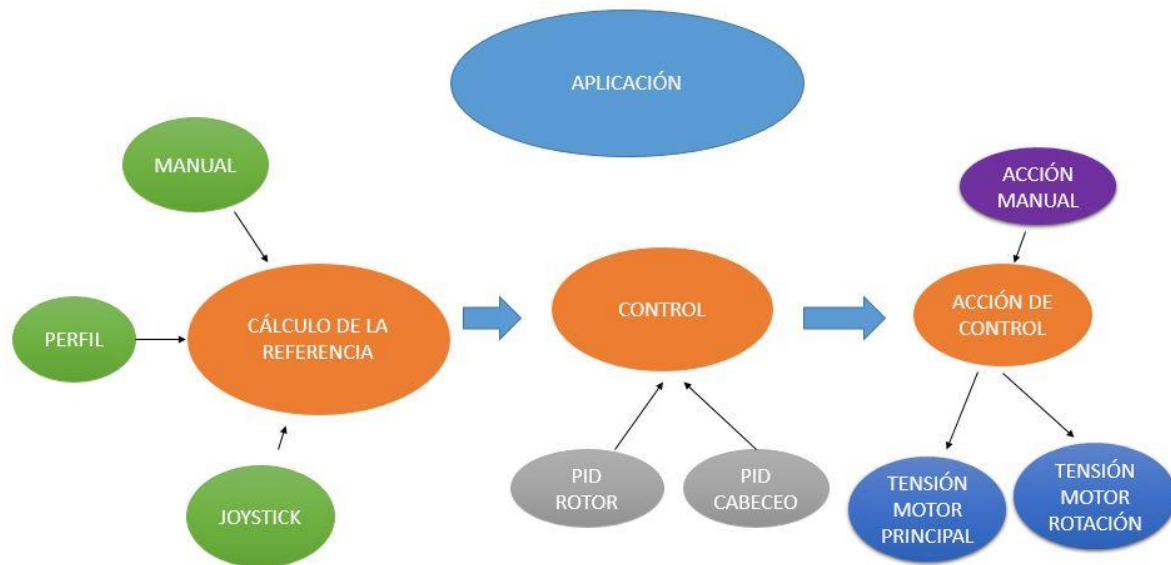


Figura 37 Diagrama de funcionamiento de la aplicación.

En la Figura 37 se ve el diagrama de la aplicación. En primer lugar se calcula el valor de la referencia. Ésta depende del modo de funcionamiento, puede ser introducida directamente por el usuario, puede provenir el perfil, puede venir del joystick y éste, a su vez, se puede encontrar en el mismo equipo o en otro diferente y ser pasada por comunicación TCP.

Una vez definido el valor de la referencia se procede al cálculo de la acción de control de los dos motores. Una vez se ha calculado la acción de control se procede a modificar la tensión de los motores, esta tensión también puede ser definida en modo manual directamente por el usuario.

7.1 CÁLCULO DE LA REFERENCIA.

Al disponer de tres modos de funcionamiento que nos proporcionan una referencia es necesario desarrollar una lógica de control que nos proporcione una única referencia que pasaremos a los controladores PID.

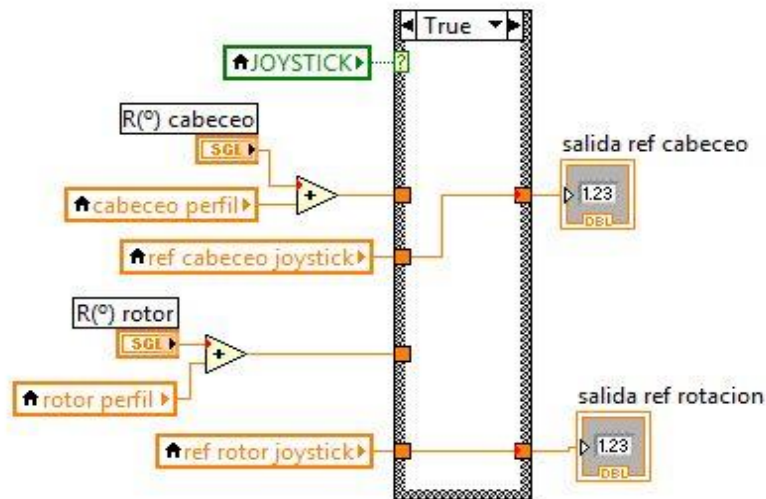


Figura 38 Código para el cálculo de la referencia.

En la Figura 38 se puede ver el código encargado de controlar la referencia. En primer lugar, se suma a la referencia introducida en el panel de control la referencia proveniente del perfil, como la referencia del perfil será diferente a cero, solamente en el caso de que el mismo éste activado no dará problemas. En segundo lugar, y una vez calculada esta referencia, se pasa a un Case Structure que pasará el valor de la referencia a los PIDs, el valor que pasará dependerá de que esté el modo joystick activado o no. Si está el joystick activado, toma el valor de la referencia del joystick y lo pasa a la salida, si no está activado el joystick, pasa el valor procedente de la suma anterior a la salida.

En la Figura 38 podemos ver el diagrama de bloques que sigue el cálculo de la referencia que deben seguir los controladores. Este diagrama de bloques es el implementado mediante el código de la Figura 39.

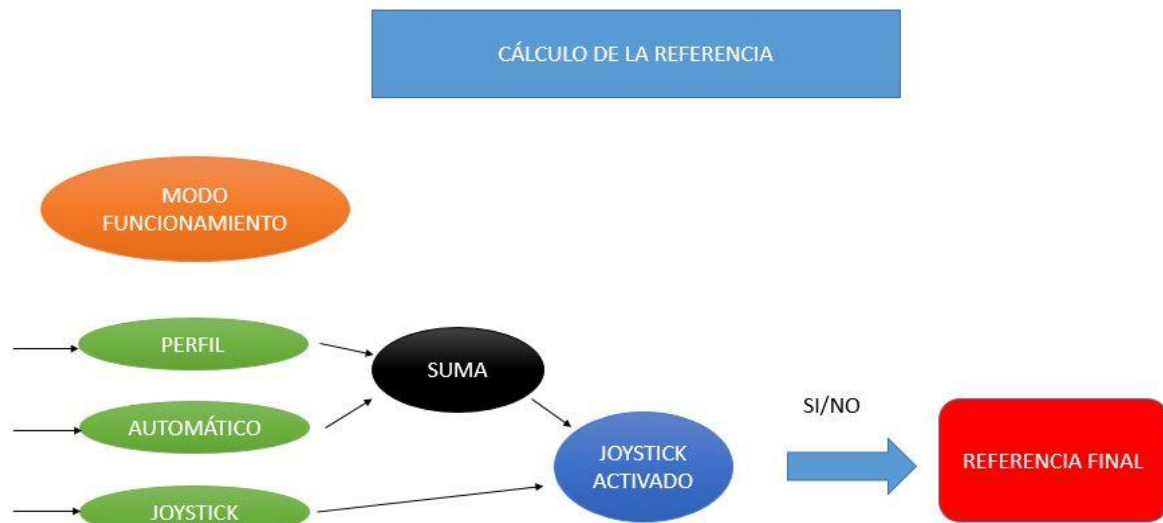


Figura 39 Diagrama del cálculo de la referencia.

7.2 ACCIÓN DE CONTROL.

Una vez se ha calculado el valor de la referencia, la aplicación procede a calcular la acción de control aplicada a la variable manipulada, para que la maqueta de simulación siga a la referencia marcada. En la Figura 40 se puede ver el diagrama de bloques del control.

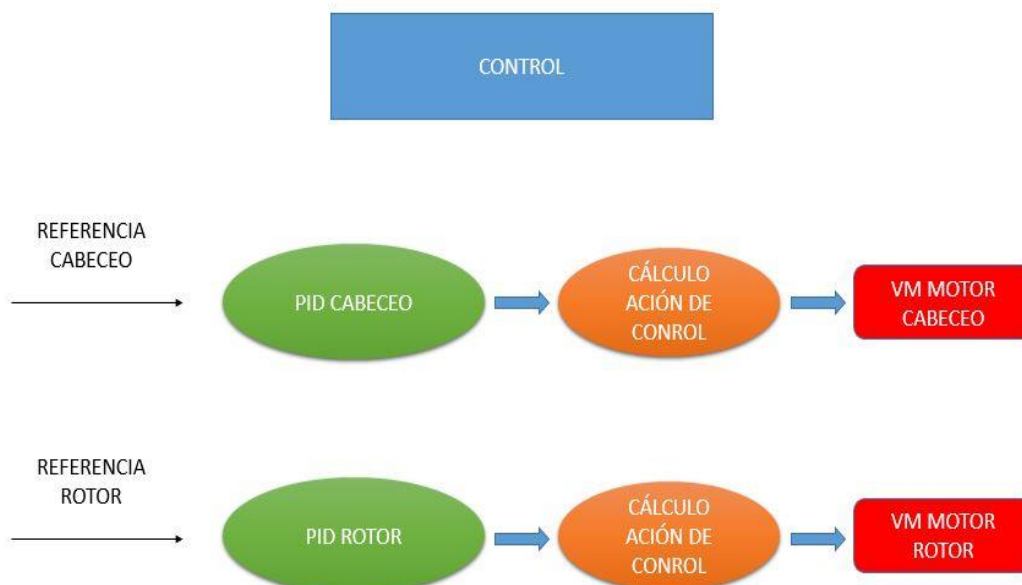


Figura 40 Diagrama de control de la aplicación.



8 BIBLIOGRAFÍA.

Instruments, N. (s.f.). *Software de Desarrollo de Sistemas NI LabVIEW*. Obtenido de
<http://www.ni.com/labview/esa/>

J. L: Guzmán, R. c. (1996). *Sistemas de Control Automatico*. Prentice Hall.

K. J. Aström, T. (2009). *Control PID Avanzado*. Prentice hall.