



CARRERA DE INGENIERÍA ELECTRÓNICA

PROYECTO DE TITULACIÓN PREVIO A LA OBTENCIÓN DEL TÍTULO DE
INGENIERO ELECTRÓNICO

PROYECTO TÉCNICO:

“DISEÑO E IMPLEMENTACIÓN DE ALGORITMOS DE CONTROL DIFUSO Y
PID ADAPTATIVO STR PARA EL SISTEMA MIMO DE DOBLE ROTOR 33-220
DE FEEDBACK PARA EL LABORATORIO DE CONTROL AUTOMÁTICO DE
LA UNIVERSIDAD POLITÉCNICA SALESIANA SEDE GUAYAQUIL”

AUTORES:

MARCOS ENRIQUE MIRANDA ORELLANA
JAIME EDISON SOLIS DELGADO

TUTOR:

Ing. ORLANDO BARCIA Msc.

GUAYAQUIL – ECUADOR

2018

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA.

Los conceptos ejecutados, los diagramas ilustrados, los análisis realizados y las conclusiones obtenidas con relación al presente trabajo son de exclusiva responsabilidad de los autores, Marcos Enrique Miranda Orellana y Jaime Edison Solis Delgado, y el patrimonio intelectual del proyecto corresponde a la Universidad Politécnica Salesiana.

Guayaquil, agosto del 2018

Jaime Edison Solis Delgado

C.I.: 0924362965

Marcos Enrique Miranda Orellana

C.I.:0927828218

CERTIFICADO DE CESIÓN DE DERECHOS.

Nosotros, MARCOS ENRIQUE MIRANDA ORELLANA con cédula de identidad número 0927822718 y JAIME EDISON SOLIS DELGADO con cédula de identidad número 0924362965, declaramos de nuestra voluntad y libre elección que cedemos a la Universidad Politécnica Salesiana, los derechos patrimoniales del proyecto técnico con el nombre de “Diseño e implementación de algoritmos de control difuso y PID adaptativo STR para el sistema MIMO de doble rotor 33-220 de Feedback para el Laboratorio de Control Automático de la Universidad Politécnica Salesiana sede Guayaquil”, realizada como trabajo de titulación para la institución en mención. Para la obtención del título de Ingeniero Electrónico.

Según lo establecido en la Ley de Propiedad Intelectual y su normativa institucional vigente, en nuestra condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato impreso y digital a la biblioteca de la Universidad Politécnica Salesiana

Guayaquil, agosto del 2018

Jaime Edison Solis Delgado

C.I.: 0924362965

Marcos Enrique Miranda Orellana

C.I.:0927828218

CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.

Por medio de la presente constancia que los Sres. Jaime Edison Solis Delgado y Marcos Enrique Miranda Orellana han desarrollado y elaborado satisfactoriamente el proyecto final de titulación, que se ajusta a las normas establecidas por la Universidad Politécnica Salesiana, por tanto, autorizo su presentación para los fines legales pertinentes.

Ing. Orlando Barcia MSc.

DIRECTOR DEL PROYECTO

DEDICATORIA.

A mis padres Cesar Solis y Carmita Delgado y a mis hermanos, sobrino y en especial a mi primo Luis Eduardo que ya no se encuentra conmigo, quienes me han acompañado durante toda la vida y a lo largo de este proceso de estudios y me han impulsado y motivado siempre a poder conseguir mis metas y han forjado en mis muchos valores que aplico durante mi rutina diaria y de los cuales deseo implementarlos en mi vida laboral y profesional.

Jaime E. Solis D.

DEDICATORIA.

Dedico este trabajo a mis padres, Fausto y Patricia; a mi abuelita Lidia, quienes siempre me enseñaron la importancia de instruirse y esforzarse por ser mejor.

También lo dedico a mi esposa Pamela e hijo Ammón que cada mañana al despertar me suscitan trabajar por un mundo mejor.

Marcos E. Miranda O.

AGRADECIMIENTO.

Agradezco a mi familia por siempre estar presente y guiarme a lo largo de mi carrera universitaria.

Agradezco a aquellos compañeros que pudieron brindarme su ayuda con temas relacionados a mi carrera y mi vida personal.

Agradezco a mi novia por estar siempre conmigo en todo el proceso de titulación.

Jaime E. Solis D.

AGRADECIMIENTO.

Mi primer agradecimiento es a Dios por las bendiciones que me han permitido llegar a completar este trabajo y mi grado.

A mi familia los incontables sacrificios por verme materializar mis proyectos y metas.

A mi compañero en este trabajo por su ánimo constante en las dificultades que afrontamos.

A mis maestros que incondicionalmente me ayudaron y guiaron en el desarrollo de este proyecto.

Marcos E. Miranda O.

AGRADECIMIENTO GENERAL.

Agradecemos a la “Universidad Politécnica Salesiana” por las facilidades otorgadas y la autorización de trabajar en sus laboratorios.

Esto nos permitió adquirir nuevos conocimientos de gozar como alumnos de tan prestigiosa institución.

Gratificamos en todo lo que vale a las autoridades de esta honorable institución, por brindarnos la oportunidad de incorporarnos al programa de superación académica el cual nos permitió estudiar esta ingeniería.

A nuestro tutor Ing. Orlando Barcia, sus conocimientos, sus orientaciones, su paciencia y motivación, que han sido fundamentales para la realización de nuestro proyecto, estamos muy agradecidos por todo el tiempo dedicado.

Jaime y Marcos

RESUMEN.

Año	Alumnos	Director de Proyecto Técnico	Tema del Proyecto
2018	Marcos Enrique Miranda Orellana	Ing. Orlando Barcia MSc.	“Diseño e implementación de algoritmos de control difuso y PID adaptativo STR para el sistema MIMO de doble rotor 33-220 de Feedback para el laboratorio de control automático de la universidad politécnica salesiana sede Guayaquil.”
	Jaime Edison Solis Delgado		

El reciente proyecto técnico de titulación, tiene como objetivo general, la implementación de algoritmos de control difuso y PID adaptativo STR para el sistema MIMO de doble rotor 33-220 (TRMS) de Feedback utilizando Matlab / Simulink.

Este proyecto de titulación ayuda a la comprensión de los diferentes tipos de control a los estudiantes además con asignaturas como teoría de control I, II y III, adicionalmente con informática industrial de la Universidad Politécnica Salesiana sede Guayaquil para la aplicación didáctica del módulo comparando los diferentes tipos de controladores que hay en la actualidad y que son de suma importancia en los procesos industriales.

Este proyecto forma la base para futuras investigaciones ya que implementa dos controladores robustos y realiza un análisis de los mismos con el fin de determinar, en función de sus características principales, la idoneidad de su uso en sistemas con una dinámica similar.

Palabras clave: control adaptativo, STR, PID, control difuso, identificación de sistemas, MIMO, control descentralizado.

ABSTRACT.

Year	Students	Technical project manager.	Project theme.
2018	Marcos Enrique Miranda Orellana	Ing. Orlando Barcia MSc.	“Design and implementation of fuzzy control algorithms and adaptive PID STR for the Twin rotor MIMO system 33-220 feedback for the automatic control laboratory of the Universidad Politécnica Salesiana in Guayaquil.”
	Jaime Edison Solis Delgado		

The recent technical degree project has as general objective the implementation of fuzzy control algorithms and adaptive PID STR for the Twin rotor MIMO system 33-220 (TRMS) Feedback using Matlab / Simulink.

This degree project helps students to understand the different types of control with subjects such as control theory I, II and III, additionally with industrial computing from the Universidad Politécnica Salesiana in Guayaquil. for the didactic application of the module comparing the different types of controllers that currently exist and that are of utmost importance in industrial processes.

This project forms the basis for future research since it implements two robust controllers and carries out an analysis of them in order to determine, based on their main characteristics, the appropriateness of their use in systems with similar dynamics.

Key words: adaptive control, STR, PID, fuzzy control, system identification, MIMO, decentralized control.

ÍNDICE DE CONTENIDO.

CERTIFICADO DE RESPONSABILIDAD Y AUTORÍA.....	II
CERTIFICADO DE CESIÓN DE DERECHOS.....	III
CERTIFICADO DE DIRECCIÓN DEL TRABAJO DE TITULACIÓN.....	IV
DEDICATORIA.....	V
AGRADECIMIENTO.....	VII
AGRADECIMIENTO GENERAL.....	IX
RESUMEN.....	X
ABSTRACT.....	XI
ÍNDICE DE CONTENIDO.....	XII
ÍNDICE DE TABLAS.....	XIV
ÍNDICE DE FIGURAS.....	XV
ABREVIATURAS Y SIMBOLOGÍA.....	XVII
INTRODUCCIÓN.....	1
1 EL PROBLEMA.....	2
1.1 Descripción del problema.....	2
1.2 Importancia y alcance.....	2
1.3 Delimitación del problema.....	3
1.3.1 Delimitación temporal.....	3
1.3.2 Delimitación espacial.....	3
1.3.3 Delimitación académica.....	3
1.4 Objetivos.....	4
1.4.1 Objetivo general.....	4
1.4.2 Objetivos específicos.....	4
2 MARCO TEÓRICO.....	5
2.1 Twin Rotor MIMO System.....	7
2.2 Identificación de sistemas.....	7
2.2.1 Señal Binaria Pseudo-aleatoria.....	8
2.2.2 Modelo Polinomiales.....	10
2.2.3 Mínimos cuadrados.....	10
2.2.4 Error del modelado.....	13
2.3 Controlador difuso.....	13
2.3.1 Funciones de membresía.....	14
2.3.2 Reglas.....	15
2.3.3 Fuzzy PD+I.....	16
2.4 Controlador Adaptativo.....	17
2.4.1 Asignación de polos.....	18
3 MARCO METODOLÓGICO.....	21
3.1 Modelo del TRMS.....	22
3.1.1 Modelo matemático del TRMS.....	22
3.1.2 Identificación experimental del TRMS.....	25
3.2 Diseño del controlador difuso.....	32
3.2.1 Control difuso para Pitch.....	32
3.2.2 Control difuso para Yaw.....	37
3.3 Diseño del controlador PID adaptativo STR.....	42
3.3.1 Controlador adaptativo para el Pitch.....	42
3.3.2 Controlador adaptativo para Yaw.....	46
4 RESULTADOS.....	49

4.1	Control fuzzy	49
4.2	Controlador autoajustable PID	50
5	ANÁLISIS DE RESULTADOS	51
5.1	Validación de la identificación experimental y del modelo matemático del TRMS	51
5.2	Pruebas con el controlador Fuzzy PD+I.....	53
5.3	Pruebas con el controlador PID STR	54
5.4	Pruebas con entrada escalera	54
5.5	Pruebas con perturbaciones	56
	CONCLUSIONES.	58
	RECOMENDACIONES.	59
	REFERENCIAS BIBLIOGRÁFICAS.....	60
	ANEXOS	62

ÍNDICE DE TABLAS.

Tabla 1 Modelos Polinomiales	10
Tabla 2: Parámetros del TRMS	24
Tabla 3: Variables lingüísticas de entrada.	32
Tabla 4: Variables lingüísticas de salida.	33
Tabla 5: Reglas para el controlador Difuso.	33
Tabla 6: Variables lingüísticas de entrada.	38
Tabla 7: Variables lingüísticas de salida.	38
Tabla 8: Reglas para el controlador Difuso.	38
Tabla 9 Error relativo RMS de el modelado lineal y el no lineal.	53
Tabla 10 Promedio de voltaje de los controladores con una entrada paso.	54
Tabla 11 Promedio de voltaje de los controladores con una entrada escalera.	55
Tabla 12 Promedio de voltaje de los controladores con perturbación	57

ÍNDICE DE FIGURAS.

Figura 1 Módulo de trabajo, a la izquierda el PC # 8 y a la derecha el	2
Figura 2 Universidad Politécnica Salesiana, sede Guayaquil, ubicación Geográfica..	3
Figura 3 Propiedad de escalado (a). Propiedad de aditividad (b)	5
Figura 4 Sistema en lazo cerrado	6
Figura 5 Modelo esquemático del TRMS	7
Figura 6 Proceso de identificación de sistemas	8
Figura 7 Diseño de PRBS con longitud de 31 periodos de muestreo	9
Figura 8 Señal PRBS con $N = 5$, $T_s = 0.1$ y $p = 2$. Realizada en Matlab.....	9
Figura 9 Estructura del controlador difuso	14
Figura 10 Principales funciones de membresía, a) triangular. b) trapezoidal. c) y d) gaussianas	15
Figura 11 Controlador Fuzzy PD+I	16
Figura 12 Controlador Autoajutable (STR)	17
Figura 13 Diagrama de bloques del controlador PID para el método propuesto	19
Figura 14 Metodología del proyecto	21
Figura 15 Modelo fenomenológico del TRMS	22
Figura 16 Señal PRBS para identificación de la variable Pitch Principal y Pitch Cruzado.	25
Figura 17 Señal PRBS para identificación de la variable Yaw Principal y Yaw Cruzado.	25
Figura 18 Esquema en Simulink para obtener datos de la variable Pitch principal y cruzada.	26
Figura 19 Datos obtenidos experimentalmente para el ángulo principal de Pitch.	26
Figura 20 Identificación del modelo Pitch principal en Ident de Matlab.....	27
Figura 21 Datos obtenidos experimentalmente para el ángulo cruzado de Pitch.	27
Figura 22 Identificación del modelo Pitch cruzado en Ident de Matlab.	28
Figura 23 Esquema en Simulink para obtener datos de la variable Yaw principal y cruzada.	29
Figura 24 Datos obtenidos experimentalmente para el ángulo principal de Yaw.	29
Figura 25 Identificación del modelo Yaw principal en Ident de Matlab.....	30
Figura 26 Datos obtenidos experimentalmente para el ángulo cruzado de Yaw.	30
Figura 27 Identificación del modelo Yaw cruzado en Ident de Matlab.	31
Figura 28 Modelo lineal del TRMS en lazo abierto en Simulink.	31
Figura 29 Salida del modelo lineal con una señal de prueba paso en Matlab.....	32
Figura 30 Esquema del controlador difuso para ángulo Pitch.	34
Figura 31 Funciones de membresía de entrada del error para el ángulo Pitch.	34
Figura 32 Funciones de entrada de la derivada del error para el ángulo Pitch.	34
Figura 33 Funciones de membresía de salida para el motor del ángulo Pitch.	35
Figura 34 Gráfica de la superficie del controlador difuso para ángulo Pitch.....	35
Figura 35 Diseño del control PID para fuzzy en el movimiento Pitch.	36
Figura 36 Esquema en Simulink del modelo lineal con controlador Fuzzy Incremental para ángulo Pitch.....	36
Figura 37 Esquema en Simulink del modelo no lineal con controlador Fuzzy Incremental para ángulo Pitch.....	37
Figura 38 Respuesta a entrada paso y señal de control para ángulo Pitch.	37
Figura 39 Esquema del controlador difuso para ángulo Yaw.....	39
Figura 40 Funciones de membresía de entrada del error para el ángulo Yaw.....	39

Figura 41 Funciones de entrada de la derivada del error para el ángulo Yaw.....	40
Figura 42 Funciones de membresía de salida para el motor del ángulo Yaw.....	40
Figura 43 Gráfica de la superficie del controlador difuso para ángulo Yaw.....	40
Figura 44 Diseño del control PID para fuzzy en el movimiento Yaw.....	41
Figura 45 Esquema en Simulink del modelo no lineal con controlador Fuzzy Incremental para ángulo Yaw.	41
Figura 46 Esquema en Simulink del modelo lineal con controlador Fuzzy Incremental para ángulo Yaw.	42
Figura 47 Respuesta a entrada paso y señal de control para ángulo Yaw.	42
Figura 48 Gráfica de los polos y ceros de la función de transferencia linealizada del Pitch.	44
Figura 49 Esquema en Simulink del modelo no lineal con PID adaptativo STR para el ángulo Pitch.....	45
Figura 50 Esquema en Simulink del modelo lineal con PID adaptativo STR para el ángulo Pitch.....	45
Figura 51 Respuesta a entrada paso con control PID adaptativo STR para ángulo Pitch.	46
Figura 52 Esquema en Simulink del modelo no lineal con PID adaptativo STR para el ángulo Yaw.....	47
Figura 53 Esquema en Simulink del modelo lineal con PID adaptativo STR para el ángulo Yaw.....	47
Figura 54 Respuesta a entrada paso con control PID adaptativo STR para ángulo Yaw.....	48
Figura 55 Gráfica de la superficie obtenida para el ángulo Pitch y para el ángulo Yaw.	49
Figura 56 Respuesta del controlador fuzzy clásico en el modelo no lineal.....	49
Figura 57 Parámetros utilizados en el controlador PID STR para pitch y yaw.....	50
Figura 58 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada paso. ...	51
Figura 59 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada Pulso... ..	52
Figura 60 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada sinusoidal.	52
Figura 61 Respuesta al paso con el control Fuzzy PD+I con la señal de control.	53
Figura 62 Respuesta al paso con el control PID STR con la señal de control.	54
Figura 63 Respuesta a señal escalera del ángulo Pitch con la señal de control.	55
Figura 64 Respuesta a señal escalera del ángulo Yaw con la señal de control.	55
Figura 65 Respuesta con perturbaciones del ángulo Pitch con la señal de control....	56
Figura 66 Respuesta con perturbaciones del ángulo Yaw con la señal de control.	57

ABREVIATURAS Y SIMBOLOGÍA.

ARMAX	Auto regresión de promedio móvil con entrada exógena
ARX	Auto regresión con entrada exógena
BJ	Box Jenkins
OE	Error de salida
PRBS	Señal Binaria Pseudoaleatoria
RMSE	Raíz del error cuadrático medio
MIMO	Múltiples entradas y múltiples salidas
TRMS	Twin Rotor MIMO System
ξ	Factor de amortiguamiento
w_n	Frecuencia natural
Pitch	Movimiento de cabeceo del TRMS
Yaw	Movimiento en alabeo del TRMS
θ_y, θ_p	Ángulo yaw y pitch
u_y, u_p	Voltaje de los motores yaw y pitch
PID	Proporcional Integral Derivativo
STR	Regulador autoajustable
$P(k)$	Matiz de covarianza
LTI	Lineal e invariante en el tiempo
.mdl	Tipo de archivo para guardar datos en Simulink
.m	Tipo de archivo para guardar datos en Matlab
$\tau_1; \tau_2$	Momento de los rotores de pitch y yaw
t_r	Tiempo de subida

t_s	Tiempo de establecimiento
M	Sobrepico maximo

INTRODUCCIÓN.

El módulo (Twin Rotor MIMO System o TRMS), es un sistema de alta complejidad, esto se debe a la alta no linealidad, acoplamiento cruzado significativo entre sus dos ejes y aerodinámica compleja lo cual lo convierte en un problema interesante para su análisis.

Este tipo de sistemas son de interés didáctico, aunque tiene varias diferencias a su semejante en la industria, la formulación del modelo matemático es basada en ecuaciones diferenciales, con lo que este estudio tiene como ventaja la analogía que este tipo de sistemas tiene con procesos reales, siendo aún estos de mayor complejidad.

El control PID lineal se puede apreciar en varios tipos de problemas industriales. Su principal desventaja es que para ajustar sus valores se deben realizar pruebas de fallo y error. La eficacia de un controlador PID lineal depende de que se ajusten sus parámetros entorno a un objetivo [1].

Al presentarse más incremento de innovación tecnológica, se generan sistemas cada vez más complejos donde el uso de controladores requiere más precisión con la finalidad de lograr una propuesta más eficaz ante un problema más complejo [2].

En la propuesta planteada del proyecto se busca evaluar dos modelos: PID adaptativo autoajutable y Fuzzy para el control del TRMS por medio del uso de controladores descentralizado y tomando en cuenta solo los movimientos principales de cada uno de esos rotores (Pitch y Yaw).

El planteamiento y la elaboración de ambos controladores son con la finalidad de elaborar al final una comparación de ambos y demostrar cuál de ellos es más eficaz en respuesta a este tipo de sistemas.

1 EL PROBLEMA.

1.1 Descripción del problema

En la actualidad, el Laboratorio de Control Automático del Bloque E de la Universidad Politécnica Salesiana Sede Guayaquil, posee un sistema de control TWIN ROTOR MIMO SYSTEM (TRMS) 33-220 (MIMO es de múltiples entradas y múltiples salidas, por sus siglas en inglés) de la empresa FEEDBACK [1]; este módulo posee la capacidad de integrar los diversos tipos de sistema de reguladores, contando actualmente con un controlador proporcional integral derivativo (PID) lineal y uno no lineal [2]. El módulo TRMS al ser un sistema aerodinámico esta propenso a varias perturbaciones lo cual dificulta su control, siendo un problema al implementar controladores lineales porque no se adaptan al sistema.



Figura 1 Módulo de trabajo, a la izquierda el PC # 8 y a la derecha el sistema TRMS

Con base en los conocimientos de Teoría de control I, II y III, Electiva II, Informática Industrial y adicionando Control Adaptativo y aerodinámica al área de estudio, con la utilización del PC-8 del laboratorio y el software ya instalado de Matlab/Simulink; se responderá a la problemática con la implementación de dos reguladores, uno de lógica difusa y otro de PID adaptativo tipo regulador auto ajustable (Self-Tuning Regulator, STR por sus siglas en inglés).

1.2 Importancia y alcance

Es muy significativo conocer la importancia de los controladores difuso y PID adaptativo en el área de control ya que con su aplicación se podrá entender mejor el uso de estos sistemas en la elaboración de procesos y en la eficiencia de su funcionamiento en el área industrial.

Este proyecto va dirigido especialmente para los alumnos de las materias de control I, II y III y la materia informática industrial para la carrera de ingeniería electrónica de la Universidad Politécnica Salesiana sede Guayaquil.

El proyecto aportará a los docentes de las materias anteriormente mencionadas a poder enseñar de una forma mucho más didáctica los diferentes tipos de control existentes en el área, lo cual implica que los estudiantes podrán reforzar los conocimientos teóricos aprendidos en las aulas.

1.3 Delimitación del problema

1.3.1 Delimitación temporal

El proyecto técnico implementado tuvo un tiempo de duración de 17 meses y fue culminado en el año 2018.

1.3.2 Delimitación espacial

Este proyecto fue desarrollado en las instalaciones de la Universidad Politécnica Salesiana Sede Guayaquil, bloque E, en el laboratorio de Control. En la Figura 2 Universidad Politécnica Salesiana, sede Guayaquil, ubicación Geográfica se muestra la ubicación de la Universidad Politécnica Salesiana.

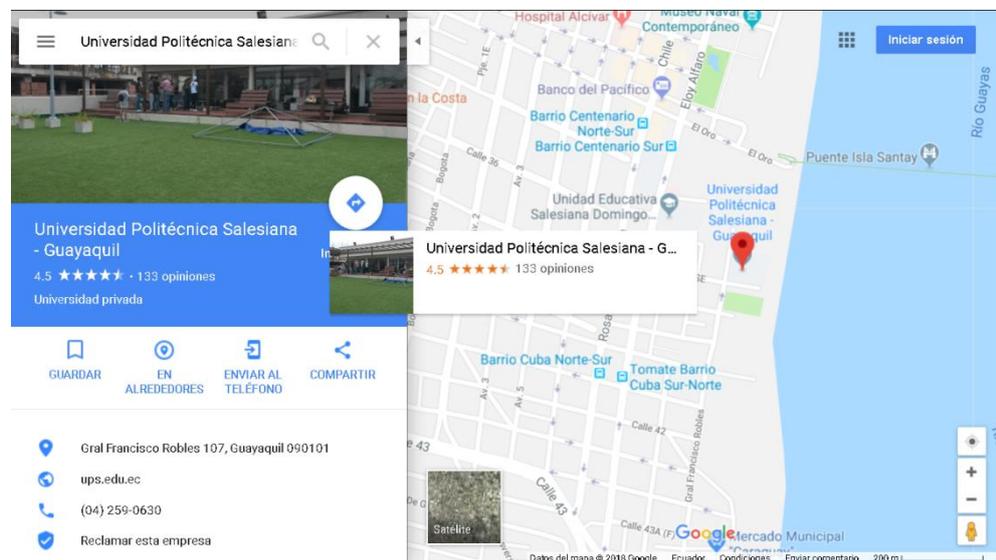


Figura 2 Universidad Politécnica Salesiana, sede Guayaquil, ubicación Geográfica

1.3.3 Delimitación académica

El proyecto consiste en la implementación de algoritmos de control difuso y PID adaptativo STR mediante Matlab/Simulink para el sistema TWIN ROTOR MIMO SYSTEM (TRMS) 33-220 de la empresa Feedback para el laboratorio de control ubicado en el bloque E de la Universidad Politécnica Salesiana sede Guayaquil. Otros tipos de controles quedan fuera del alcance del documento.

1.4 Objetivos

1.4.1 Objetivo general

Diseñar e implementar algoritmos de control difuso y PID adaptativo STR para el sistema MIMO de doble rotor 33-220 (TRMS) de Feedback utilizando Matlab/Simulink.

1.4.2 Objetivos específicos

- Diseñar reglas y funciones para el controlador difuso.
- Obtener la respuesta del sistema con la implementación el controlador difuso.
- Diseñar parámetros para el controlador PID del sistema.
- Diseñar algoritmos de control adaptativo tipo STR en el controlador PID.
- Obtener la respuesta del sistema con la implementación el controlador PID adaptivo.
- Evaluar los algoritmos de control propuestos a través del análisis temporal y el seguimiento de referencia del sistema Twin Rotor Mimo System 33-220.
- Comparar los algoritmos de control y establecer el más eficiente.

sistema, m es el orden del numerador en la función de transferencia, define los ceros del sistema.

Otra representación del sistema es la de espacio de estados que se representa de la siguiente manera, considerando los factores, o como en los libros encontramos, variables intermedias del sistema como x_1, x_2, \dots, x_n , siendo el subíndice n el número total de variables [3].

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t)\end{aligned}$$

Donde, A , es la matriz de estado, B , es la matriz de entrada, C , es la matriz de salida, D , es la matriz de transmisión directa, x , es el vector de estados o de variables de estado. De la anterior ecuación a través de la transformada Laplace se puede llegar a la equivalencia a la función de transferencia, con ayuda de la matriz identidad I .

$$G(s) = C(sI - A)^{-1}B + D \quad (2)$$

Para los sistemas de control, se utiliza el sistema en lazo cerrado y abierto, el primero que se muestra en la Figura 4 Sistema en lazo cerrado, se vale de la utilización de la señal de salida del sistema para restarla con la entrada, que es llamada también referencia o setpoint, utilizando la diferencia entre estas, el error, para controlar el sistema; por el contrario el control de lazo abierto no considera la salida del sistema valiéndose solamente de la entrada de este.

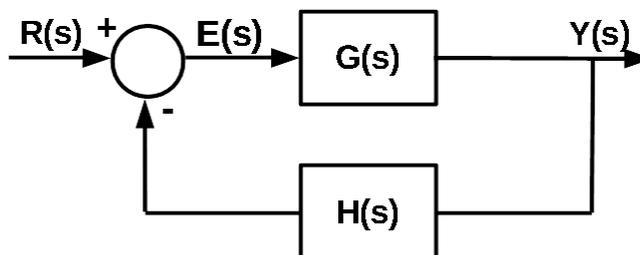


Figura 4 Sistema en lazo cerrado [3]

Donde, $R(s)$ es el setpoint, o punto de referencia deseado, $E(s)$ es el error entre la referencia, $Y(s)$ es la salida y $H(s)$ es la función de transferencia del sensor lector de la salida del sistema. Mientras que matemáticamente se representa con la siguiente ecuación.

$$\frac{Y(s)}{R(s)} = \frac{G(s)}{1 + G(s)H(s)} \quad (3)$$

El sistema de control que utilizaremos es en lazo cerrado que tiene la ventaja de mejor adaptación ante cambios y perturbaciones y necesitan menos recalibración, sin menospreciar las virtudes de los sistemas en lazo abierto que son menos costosos y mucho más fácil en su construcción y mantenimiento, además son útiles cuando no hay problemas de estabilidad [3].

2.1 Twin Rotor MIMO System

Es un módulo elaborado por la empresa de Feedback en donde se permite simular el sistema dinámico de un helicóptero, este contiene un pivote en donde se produce el giro de los ejes tanto vertical como horizontal, además de poseer dos motores considerados como Pitch y Yaw uno principal y el otro de cola [1], en este sistema se genera un efecto cruzado como lo muestra el modelo esquemático en la Figura 5 Modelo esquemático del TRMS.

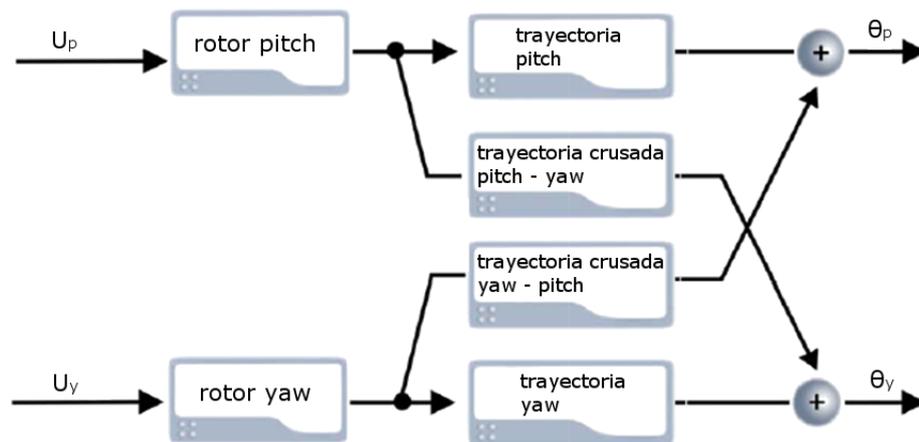


Figura 5 Modelo esquemático del TRMS [1].

El módulo Feedback en su hardware muestra dos botoneras una de inicio y una de paro de emergencia, además de la fuente de alimentación, posee un Encoder HEDS-9000 con el que se obtiene los valores de los ángulos, usa también una tarjeta PCI-1711 la cual realiza la comunicación del Software con el Hardware, este módulo trabaja conectado a una computadora o PC en donde se realizan las simulaciones correspondientes, físicamente el TRMS se lo observa en la Figura 1.

2.2 Identificación de sistemas.

El modelado de sistemas se desarrolla a partir de las expresiones matemáticas, en este estudio sería del sistema TRMS, desarrollado en la unidad 3. Muchas veces es complejo obtener un modelo matemático por las variables físicas del sistema que podrían variar, además de variables que no se consideran en las operaciones matemáticas [4]. Otra forma de obtener el modelo es mediante la identificación del sistema, es decir, mediante la recolección de datos que describen el sistema, la entrada y la salida [6].

La Figura 6 Proceso de identificación de sistemas. muestra el proceso que se usa en la identificación experimental:

1. Adquisición de datos, con sistemas de varias entradas y varias salidas se debe definir como se tomarán, que entrada interactúa con que salida; también escoger la señal de excitación, de este paso depende toda la identificación, se necesita conocer el funcionamiento del sistema y si no se logra reflejar ese

funcionamiento en los datos no será posible obtener un modelo que se asemeje [7].

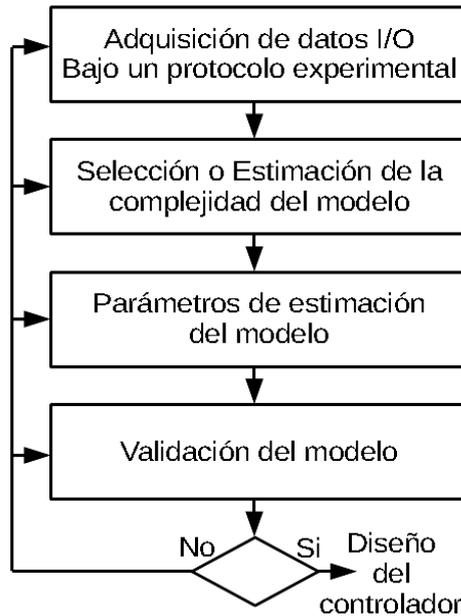


Figura 6 Proceso de identificación de sistemas [6].

2. Complejidad del modelo, definir el orden del modelo que deseamos manejar y que cumpla con una proximidad adecuada al sistema real, además del empleo de los diferentes algoritmos de estimación sean estos en tiempo real o no.
3. Parámetros de estimación, se obtienen los parámetros que definen la función de transferencia del sistema, en algunos métodos de estimación es necesario definir parámetros iniciales.
4. Validación del modelo, se debe comparar la respuesta del modelo obtenido con los datos de estimación y corroborar una aproximación adecuada, posterior a esto realizar comparaciones con el sistema físico y obtener un error tolerable.

2.2.1 Señal Binaria Pseudo-aleatoria.

La señal binaria Pseudo-aleatoria (Pseudo Random Binary Signal, PRBS, por sus siglas en inglés), es una secuencia que a lo largo su periodo tiene variaciones de ancho de pulso en apariencia aleatoria, a semejanza del ruido blanco en tiempo discreto. La señal se genera con registros de desplazamiento realimentados, como se muestra en la Figura 7 Diseño de PRBS con longitud de 31 periodos de muestreo., su longitud depende del número de registros(N), definidas por $2^N - 1$.

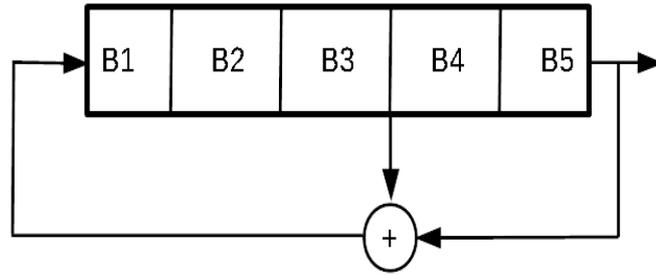


Figura 7 Diseño de PRBS con longitud de 31 periodos de muestreo [6].

Por lo menos un registro debe tener como valor inicial diferente a cero [6], las ecuaciones que definen la señal son la duración de la prueba, L , y la máxima duración del pulso pseudo-aleatorio, t_{im} .

$$t_{im} = pNT_s > t_R \quad (4)$$

$$L > p2^{N-1}T_s \quad (5)$$

Donde, T_s , es periodo de muestreo, t_R , es el tiempo de subida (rise time en inglés) del sistema, p , es un sub-múltiplo del periodo de frecuencia ($f_{PRBS} = f_s / p$), para evitar valores altos de registros que pueden complicado de implementar.

Al configurar la magnitud se debe considerar el ruido residual y que este sea menor a la señal PRBS. En este aspecto la longitud de la prueba aumenta la relación señal/ruido es demasiado pequeña [6]. En la Figura 8 Señal PRBS con $N = 5$, $T_s = 0.1$ y $p = 2$. Realizada en Matlab se puede observar la señal PRBS.

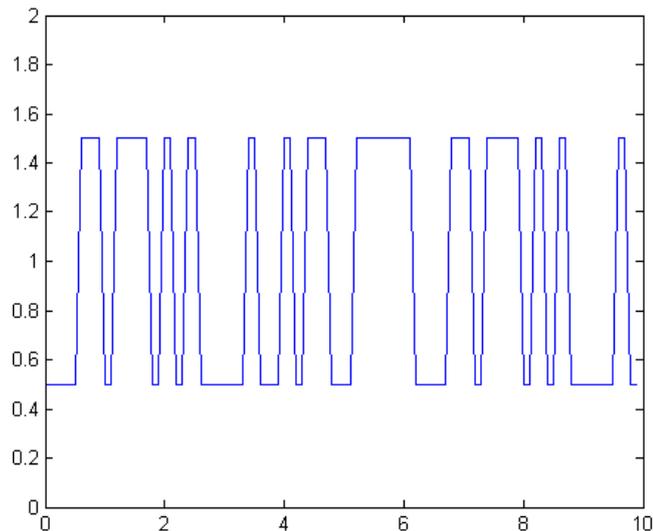


Figura 8 Señal PRBS con $N = 5$, $T_s = 0.1$ y $p = 2$. Realizada en Matlab.

2.2.2 Modelo Polinomiales.

De forma generalizada se muestra en la siguiente ecuación el modelo polinómico que relaciona la salida con la entrada y la perturbación, generalmente un ruido blanco $e(k)$ [8].

$$A(z^{-1})y(k) = \frac{B(z^{-1})}{F(z^{-1})}u(k) + \frac{C(z^{-1})}{D(z^{-1})}e(k) \quad (6)$$

$A(z^{-1})$, $B(z^{-1})$, $C(z^{-1})$, $D(z^{-1})$ y $F(z^{-1})$ son polinomios expresados en z^{-1} , el operador de atraso, como se muestra en la siguiente ecuación, algunos autores lo denominan operador de transferencia [4] [8].

$$B(z) = \sum_{k=1}^{\infty} b(k)z^{-k}$$

En varios sistemas existe un retardo n_r en el muestreo entre la entrada y la salida, un grupo importante coeficientes de B se cancelan, es decir son cero [4].

$$\begin{aligned} B(z^{-1}) &= b_{n_r}z^{-n_r} + b_{n_r+1}z^{-n_r-1} + \dots + b_{n_r+n_b-1}z^{-n_r-n_b+1} \\ &= z^{-n_r}\tilde{B}(z^{-1}), \quad b_{n_r} \neq 0 \end{aligned}$$

Este retardo es considerado al seleccionar el modelo desde las librerías de matlab/simulink. En la expresión (6) para el desarrollo teórico, es decir n_r es 1.

Los principales modelos polinómicos aparecen en la siguiente tabla.

Tabla 1 Modelos Polinomiales [4]

Modelo	Polinomios usados
Autorregresivo con una entrada exógena (ARX)	AB
Autorregresivo de media móvil con una entrada exógena (ARMAX)	ABC
Box-Jenkins (BJ)	BCDF
Error de Salida (OE)	BF

No se debe confundir con la nomenclatura que usa la representación en espacio de estados.

2.2.3 Mínimos cuadrados.

El método de mínimos cuadrados (Least Squares, LS por sus siglas en inglés) se basa en el análisis de regresión de los datos de entrada y salida del sistema y obtener los

parámetros para su modelo [9]. Desarrollada a partir del modelo ARX con $n_a = n_b = n$ y $n_r = 0$; introducimos el vector de parámetros y el vector de regresión [9].

$$\theta^T(k) = [a_1, a_2, \dots, a_n, b_1, b_2, \dots, b_n] \quad (7)$$

$$\Phi^T(k) = \begin{bmatrix} -y(k-1), -y(k-2), \dots, -y(k-n), \\ u(k-1), u(k-2), \dots, -u(k-n) \end{bmatrix} \quad (8)$$

Con base a la definición de (7) el valor del error se expresa con la siguiente ecuación.

$$e = y - R\theta$$

La variable y es un vector de manera vertical, al igual que la variable e , ambas de longitud $N - n$, siendo N la cantidad de muestras obtenidas de la entrada y salida del sistema, iniciando desde la iteración $n + 1$, R es una matriz de longitud $(N - n, 2n)$ de la siguiente forma.

$$R = \begin{bmatrix} -y(n) & -y(n-1) & \dots & -y(1) & u(n) & \dots & u(1) \\ -y(n+1) & -y(n) & \dots & -y(2) & u(n+1) & \dots & u(2) \\ \vdots & & & & & & \vdots \\ -y(N-1) & -y(N-2) & \dots & -y(N-n) & u(N-1) & \dots & u(N-n) \end{bmatrix}$$

El método busca minimiza el valor del error, lo cual se logra con la siguiente ecuación [10].

$$\hat{\theta} = (R^T R)^{-1} R^T y \quad (9)$$

El algoritmo de mínimos cuadrados recursivos. (Recursive Least Squares, RLS por sus siglas en ingles), tiene la característica de que puede ser usado en tiempo real, se usa la predicción tanto de la salida, como del error.

$$\hat{y}(k) = \theta^T(k-1)\Phi(k) \quad (10)$$

$$\hat{e}(k) = y(k) - \hat{y}(k) \quad (11)$$

El siguiente paso es la corrección con el valor actual [11], ya que al ser un método en tiempo real se toman a cada k instante, el valor real del sistema, este proceso se obtiene mediante las siguientes ecuaciones [12]:

$$\hat{\theta}(k) = \theta(k-1) + \frac{P(k-1)\Phi(k)}{1 + \xi(k)} \hat{e}(k) \quad (12)$$

$$P(k) = P(k-1) - \frac{P(k-1)\Phi(k)\Phi^T(k)P(k-1)}{1 + \xi(k)} \quad (13)$$

Donde P es la matriz de covarianza y la variable auxiliar ξ está dada en la siguiente expresión.

$$\xi = \Phi^T(k)P(k-1)\Phi(k)$$

Se expone del RLS dos métodos más, con olvido exponencial (RLS-ef, donde ef se hace referencia a exponential forgetting por sus siglas en ingles) y con olvido direccional adaptativo (RLS-adf, donde adf se hace referencia a adaptive directional forgetting por sus siglas en ingles) [12].

$$\hat{\theta}(k) = \theta(k-1) + \frac{P(k-1)\Phi(k)}{\phi + \xi(k)} \hat{e}(k) \quad (14)$$

$$P(k) = \frac{1}{\phi} \left(P(k-1) - \frac{P(k-1)\Phi(k)\Phi^T(k)P(k-1)}{\phi + \xi(k)} \right) \quad (15)$$

$$P(k) = P(k-1) - \frac{P(k-1)\Phi(k)\Phi^T(k)P(k-1)}{\varepsilon^{-1}(k) + \xi(k)} \quad (16)$$

Actualizando el vector de parámetros, (12), y la matriz de covarianza,(13), con base a los métodos RLS-ef y RLS-adf, obtenemos un nuevo vector de parámetros, (14) y una nueva matriz de covarianza, (15), para el primero; para el segundo el vector de parámetros de mantiene igual a (12) y se tiene una nueva matriz de covarianza, (16), que se muestran a continuación:

Las variables auxiliares para (16) y las variables de las que estas a su vez dependen son expuestas a continuación:

$$\varepsilon(k) = \phi(k) - \frac{1 - \phi(k)}{\xi(k)}$$

$$[\phi(k)]^{-1} = 1 + (1 + \rho)[\ln(1 + \xi(k))] + \left[\frac{(v(k-1) + 1)\eta(k)}{1 + \xi(k) + \eta(k)} - 1 \right] \frac{\xi(k)}{1 + \xi(k)}$$

$$\eta(k) = \frac{\vartheta(k)}{\lambda(k-1)}$$

$$v(k) = \phi(k)(v(k-1) + 1)$$

$$\lambda(k) = \phi(k) \left[\lambda(k-1) + \frac{\vartheta(k)}{1 + \xi(k)} \right]$$

Cuando un sistema tiene parámetros que varían con el tiempo, es decir no son LTI, la utilización de estos métodos funciona, ya que se consideran las iteraciones más recientes con mayor prioridad o peso de las que se han tomado.

El método RLS-ef que depende de ϕ , conocido como el factor de olvido, este es un número mayor a 0 y menor igual a 1, mientras más bajo es el factor de olvido menos influencia toman los valores anteriores. En el método RLS-adf el valor de ϕ es variable

en cada iteración, influenciado por el error, se adapta a los cambios que se muestran en la entrada y salida del sistema [12].

Los valores iniciales de $\theta(0)$ y $P(0)$, se lo determina por el conocimiento del sistema, la matriz de covarianza generalmente inicia de la forma G^*I , donde G es una ganancia que mientras mayor es esta menor es la importancia de los valores de estimación inicial, I es una matriz identidad de la misma longitud que el vector de parámetros del sistema [12].

2.2.4 Error del modelado.

En la validación del modelo se lo realiza mediante el cálculo del error como se explicó anteriormente, para este es usualmente utilizada la raíz del error cuadrático medio (Root Mean Square Error, RMSE, por sus siglas en inglés) como se muestra en varios estudios [2], [13] y [14]; el RMSE se calcula mediante la siguiente ecuación.

$$RMSE = \sqrt{\sum_{i=1}^N \frac{(ys_i - ym_i)^2}{N}}$$

Donde ys es la salida del sistema y ym es la salida del modelo, mientras N es la cantidad de muestras obtenidas; para una mejor interpretación de los resultados se emplea la versión relativa del RMSE (%RMSE) que se lo obtiene dividiendo el RMSE para la media de ys , es decir, la salida del sistema [14].

$$\%RMSE = \frac{RMSE}{\bar{y}_s} * 100 \quad (17)$$

2.3 Controlador difuso.

Más conocido como fuzzy logic (en inglés), este hace uso de la lógica difusa del cual la principal diferencia a la lógica convencional es el valor que toma, es decir, no se otorga un valor absoluto de verdadero o falso (0, 1) si no un valor intermedio conocidos como valor de pertenencia. En la lógica difusa se hace uso de variables lingüísticas, tales como “lento”, “más lento” o “menos lento”, que son expresiones ambiguas; que tratan de asemejarse a la mente humana [15]. Matemáticamente en su forma discreta se representa de la siguiente manera.

$$\tilde{A} = \left\{ \sum_i \frac{\mu_A(x_i)}{x_i} \right\}$$

Donde \tilde{A} es el conjunto difuso que es también la variable lingüística, $\mu(X_i)$ representa el valor que toma la entrada X_i que sería el valor de pertenencia. La operación sumando se interpreta como unión y no como suma, además X_i en el denominador indica el valor de la entrada del sistema al cual se le asignó un valor de pertenencia y no se debe interpretar como una división [15].

Los operadores que se usan en la lógica difusa se realizan matemáticamente con operadores $\max(\tilde{A}, \tilde{B})$; $\min(\tilde{A}, \tilde{B})$, y $1-(\tilde{A})$, para realizar la unión, intersección y complemento respectivamente.

La estructura del controlador se puede apreciar en la Figura 9 Estructura del controlador difuso., que consta de cuatro bloques:

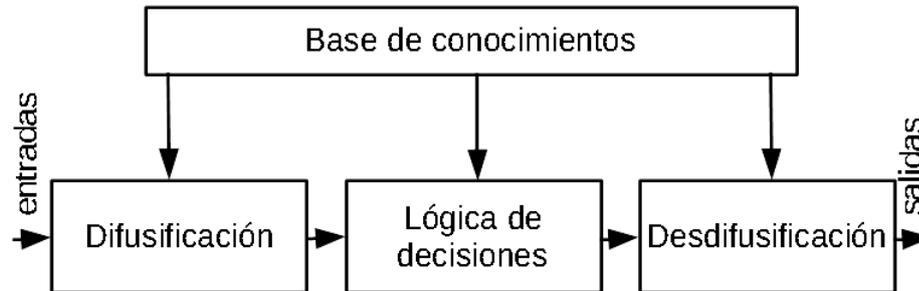


Figura 9 Estructura del controlador difuso [15].

- Difusificación, este recoge los datos de la entrada para convertirlos en valores difusos, esto en base a las funciones de membresía que más adelante se analizará.
- Base de conocimientos, es la que dicta el conocimiento del sistema y las acciones de control que pueden tomar, así como las reglas de control que define a estas.
- Lógica de decisiones, es quien procesa los datos de la base de conocimiento para dar la acción de control, esta simula el procesamiento mental humano.
- Desdifusificación, es la contraparte de la difusificación, ya que acondiciona la salida del controlador fuzzy para que a la entrada del sistema se realice el control programado.

2.3.1 Funciones de membresía.

Las funciones de membresía, MF, es una representación de los elementos del conjunto de entrada, con base al grado de pertenencia que tienen estos elementos, es decir, la MF denota el valor de pertenencia que tiene un valor de entrada en dicha MF. El conjunto de entrada previamente delimitado se lo conoce como universo discurso [16].

La Figura 10 Principales funciones de membresía, a) triangular. b) trapezoidal. c) y d) gaussiana. muestra algunas funciones de membresía más comunes, estas realizadas en Matlab. A estas funciones se le asigna una variable lingüística. Con base a la Figura 10 Principales funciones de membresía, a) triangular. b) trapezoidal. c) y d) gaussiana. a, también podemos realizar la representación matemática en tiempo continuo de la siguiente forma.

$$f(x) = \begin{cases} 3 \leq x \leq 6 \rightarrow \frac{x-3}{6-3} \\ 6 \leq x \leq 8 \rightarrow \frac{x-8}{6-8} \\ 0 \text{ en todo lo demas} \end{cases}$$

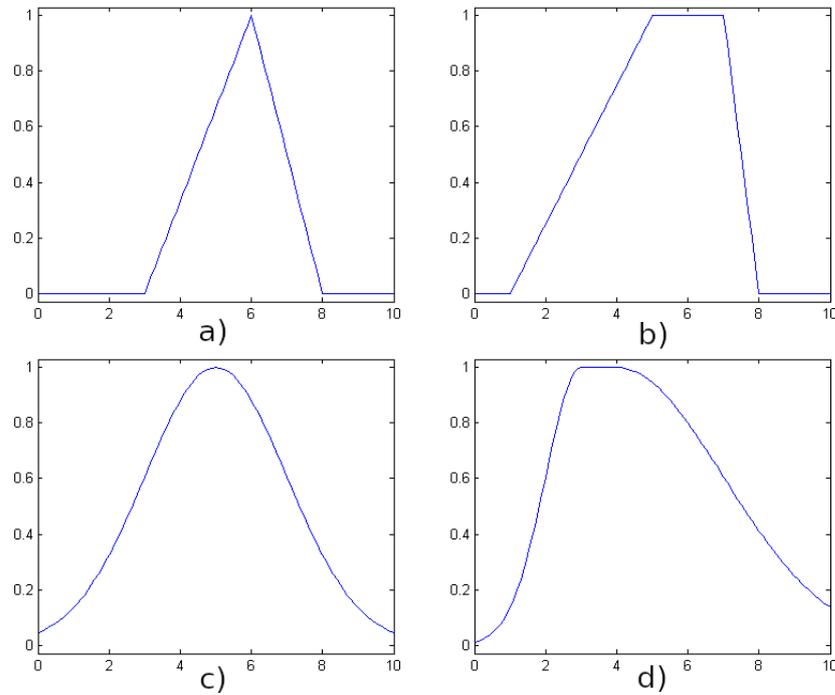


Figura 10 Principales funciones de membresía, a) triangular. b) trapezoidal. c) y d) gaussiana [16].

2.3.2 Reglas.

En las reglas se hace uso de los operadores AND (conjunción o intersección, \wedge), OR (unión o disyunción, \vee), NOT (complemento, \sim) [16], y se usa la forma IF – THEM, como se muestra en a continuación.

Si entrada 1 es \tilde{A} y entrada 2 es \tilde{B} entonces salida es \tilde{C}

Primero se relaciona el universo discurso de las diferentes variables lingüísticas, se realiza matemáticamente esta relación difusa mediante el producto cartesiano.

$$\mu_R = \mu_{A \times B}(x_1, x_2) = \min(\mu_A(x_1), \mu_B(x_2))$$

Pero esta expresión solo, es una representación de las reglas diseñadas, no las aplica ni las resuelve para esto es necesario realizar una composición que se define de la siguiente expresión.

$$T = R \cdot S$$

$$\mu_T(x_1, x_3) = \max(\min(\mu_R(x_1, x_2), \mu_S(x_2, x_3)))$$

Donde se R y S, son relaciones difusas pero no necesariamente lo deben ser ambas, y T es la composición entre estos, este método es conocido como composición max-min. Todo esto desarrolla el método de inferencia Mamdani, esto con la adición de un método de desdifusificación como el del centroide o centro de área expuesto en la siguiente expresión.

$$salida = \frac{\sum \mu(x) \cdot x}{\sum \mu(x)}$$

2.3.3 Fuzzy PD+I.

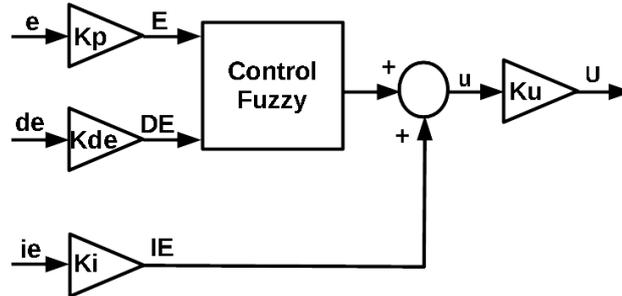


Figura 11 Controlador Fuzzy PD+I [17].

Utilizando el controlador fuzzy en control realimentado lo que recibimos en la entrada es el error por lo cual resulta simple el implementar un controlador con base en el control PID.

$$u = k_p \left(e + \frac{1}{T_i} \int_0^t e dt + T_d \frac{de}{dt} \right)$$

Y su equivalente en tiempo discreto sería:

$$u(k) = k_p \left(e(k) + \frac{T_s}{T_i} \sum_{i=1}^k e(i) + T_d \frac{e(k) - e(k-1)}{T_s} \right) \quad (18)$$

El controlador fuzzy PD+I es una implementación del controlador PID, partiendo de un controlador PD adicionando con un sumador la parte integral, este se integra a las entradas del controlador fuzzy [15], [17], como se muestra en la Figura 11, de donde podemos partir para obtener las siguientes ecuaciones con base en (18).

$$U(k) = [f(Kp e(k), Kde \cdot de(k)) + Ki \cdot ie(k)] Ku$$

Donde la función f representa el controlador fuzzy, es decir, el mapeo difuso que hace el controlador de la entrada y la salida, con lo que al realizar una aproximación lineal y acomodando los al modelo de la (18) obtenemos:

$$U(k) = KpKu \left[e(k) + \frac{Kde}{Kp} \frac{e(k) - e(k-1)}{T_s} + \frac{Ki}{Kp} \sum_i e(i) T_s \right]$$

$$\begin{aligned}
 k_p &= K_p * K_u \\
 T_d &= \frac{K_{de}}{K_p} \\
 \frac{1}{T_i} &= \frac{K_i}{K_p}
 \end{aligned}
 \tag{19}$$

Con las igualdades de (19) podemos implementar entre los varios métodos para ajustar los valores del controlador PID y adecuarlo al controlador fuzzy PD+I.

2.4 Controlador Adaptativo.

El control adaptativo, es un regulador no lineal que adiciona al sistema en lazo cerrado, una lectura del sistema, es decir, de los parámetros del sistema, con lo cual se desarrolla la modificación del controlador convencional que se emplea junto a este, con el fin de mejorarlo frente a cambios en el sistema y a las perturbaciones que este pueda estar sufriendo [18].

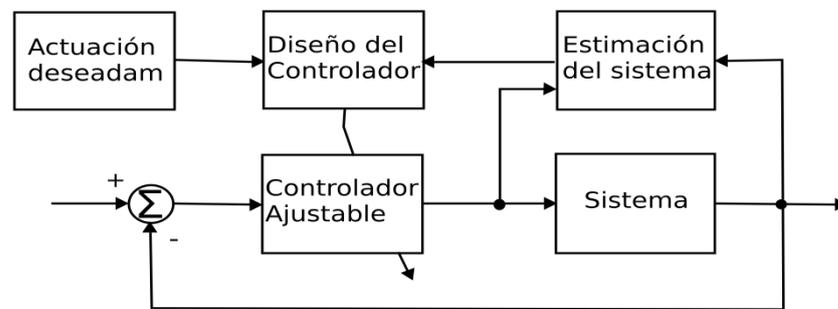


Figura 12 Controlador Autoajustable (STR) [18].

El controlador STR es un tipo de controlador adaptativo, el cual se observa en la Figura 12 Controlador Autoajustable (STR)., esta busca una estimación de la planta para que con base a los cambios observados y un referente de control, que se denomina actuación deseada, rediseñar los parámetros del controlador convencional para llegar a un regulador capaz de adaptarse al sistema; este tipo de controlador STR es conocido como de estructura explícita [11], mientras que los de estructura implícita obtiene los datos del sistema desde el cual hace una estimación de los parámetros del controlador [9].

La estructura explícita a estudiar sería la siguiente:

1. Estimación del sistema, busca los parámetros del sistema usando un método en línea que se realiza con algoritmos recursivos como el de RLS, algunas de sus diferentes variantes vistas anteriormente desde (12) a (16).
2. Diseño del controlador, el diseño del controlador recoge los parámetros del sistema y los datos para la actuación deseada del controlador, de entre estos algoritmos se usa, en este trabajo, la colocación de polos que más adelante se desarrolla.

3. Controlador ajustable, este es el que realiza la acción de control, en este trabajo el controlador es un PID con filtro en la acción derivativa.

Partiendo del control PID en tiempo continuo en la transformada de Laplace, al añadir un filtro de primer orden, filtro de capacidad simple [9], se llega a lo siguiente:

$$K(s) = \frac{U(s)}{E(s)} = k_p \left(1 + \frac{1}{T_i s} + \frac{T_d s}{T_f s + 1} \right)$$

Donde $T_f = T_d/\alpha$; $\alpha = \langle 3; 20 \rangle$. Aplicando la fórmula de conversión expuesta por [9], obtenemos la forma discreta y recursiva respectivamente mostrada a continuación:

$$K(z) = \frac{U(z)}{E(z)} = \frac{Q(z^{-1})}{P(z^{-1})} \quad (20)$$

$$u(k) = p_1 u(k-1) + p_2 u(k-2) + q_0 e(k) + q_1 e(k-1) + q_2 e(k-2) \quad (21)$$

2.4.1 Asignación de polos.

La función de transferencia en lazo cerrado con base a la Figura 4 Sistema en lazo cerrado y (3), añadiendo (20) que es el controlador, utilizando la función de transferencia discreta del sistema y tomando $H(z) = 1$, sería la siguiente

$$\frac{Y(z)}{R(z)} = \frac{B(z^{-1})Q(z^{-1})}{A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1})}$$

Siendo el denominador de ésta la ecuación característica

$$D(z^{-1}) = A(z^{-1})P(z^{-1}) + B(z^{-1})Q(z^{-1}) \quad (22)$$

Tomando en cuenta que la función característica de segundo orden es la siguiente

$$s^2 + 2\xi\omega_n s + \omega_n^2 = 0 \quad (23)$$

Se puede escribir (22) con base en (23) de lo cual se obtiene

$$D(z^{-1}) = 1 + d_1 z^{-1} + d_2 z^{-2} \quad (24)$$

Entonces a comparar (24) con (23) se obtiene las siguientes relaciones.

$$\begin{aligned} d_2 &= \exp(-2\xi\omega_n T_0) \\ d_1 &= -2\exp(-\xi\omega_n T_0) \cos\left(\omega_n T_0 \sqrt{1 - \xi^2}\right); \xi \leq 1 \\ d_1 &= -2\exp(-\xi\omega_n T_0) \cosh\left(\omega_n T_0 \sqrt{\xi^2 - 1}\right); \xi > 1 \end{aligned} \quad (25)$$

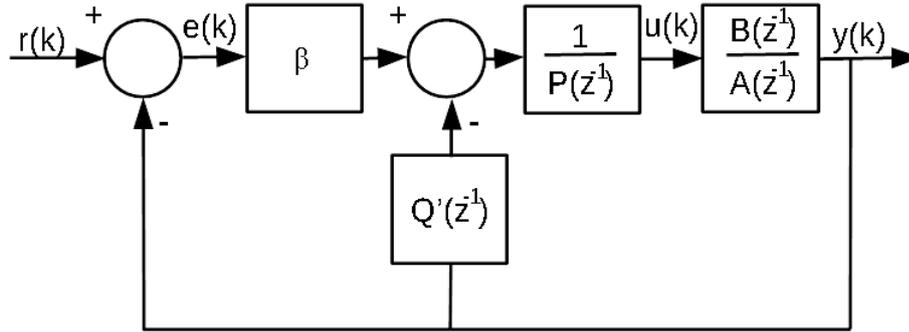


Figura 13 Diagrama de bloques del controlador PID para el método propuesto [9]

En la Figura 13 Diagrama de bloques del controlador PID para el método propuesto se muestra el diagrama del controlador que se usa en el método propuesto, el controlador toma la siguiente forma.

$$K(z) = \frac{\beta E(z) - Q'(z^{-1})Y(z)}{P(z^{-1})} \quad (26)$$

Donde los polinomios Q' y P toman la forma.

$$\begin{aligned} P(z^{-1}) &= (1 - z^{-1})(1 + \gamma z^{-1}) \\ Q'(z^{-1}) &= (1 - z^{-1})(q_0' + q_2' z^{-1}) \end{aligned} \quad (27)$$

De forma recursiva de (26) y utilizando (27).

$$\begin{aligned} u(k) &= -(\gamma - 1)u(k - 1) + \gamma u(k - 2) + \beta w(k) - \\ &[(q_0' + \beta)y(k) - (q_0' + q_2')y(k - 1) + q_2'y(k - 2)] \end{aligned} \quad (28)$$

La función de transferencia en lazo cerrado de este sería:

$$K(z) = \frac{\beta B(z^{-1})}{A(z^{-1})P(z^{-1}) + B(z^{-1})[Q'(z^{-1}) + \beta]} \quad (29)$$

Al resolver la ecuación característica D correspondiente a (29) para obtener las variables del controlador según Bobál, Böhm, Fessl, y Macháček [9] se obtiene:

$$\begin{aligned}
q_0' &= q_2' \left(\frac{b_1}{b_2} - \frac{a_1}{a_2} \right) - \frac{a_2}{b_2} \\
q_2' &= \frac{s_1}{r_1} \\
\gamma &= q_2' \frac{b_2}{a_2} \\
\beta &= \frac{d_1 + 1 - a_1 - \gamma - b_1 q_0'}{b_1}
\end{aligned} \tag{30}$$

Donde s_1 y r_1 están definidas por las ecuaciones:

$$\begin{aligned}
s_1 &= a_2(b_2(a_1(b_1 + b_2) + b_1(d_2 - a_2) - b_2(d_1 + 1)) - a_2 b_1^2) \\
r_1 &= (b_1 + b_2)(a_1 b_1 b_2 - a_2 b_1^2 - b_2^2)
\end{aligned}$$

La librería, Self-Tuning Controller Simulink Library (STCSL), diseñada por la Tomas Bata University in Zlín en el sitio de internet www.utb.cz/stctool [9], esta se puede obtener desde el enlace citado en las referencias bibliográfica bajo el ítem [19]. Librería la cual se utiliza en este proyecto y aplica los conocimientos expuestos.

3 MARCO METODOLÓGICO.

Se describen los métodos, procedimientos y técnicas que se utilizaron para recolectar y procesar la información y presentar resultados. Se justifica brevemente su elección relacionándolo con el objetivo y el problema. Se menciona quiénes diseñaron los instrumentos y su aplicación.

Se procedió a utilizar el método deductivo, estudiando casos similares en los que se implementaron en el módulo de Feedback, donde se probaron algoritmos de control semejantes a los propuestos. Con los que se obtuvieron las bases de diseño de los controladores difuso y de PID adaptativo.

El método analítico se aplicó para descomponer y observar el comportamiento del sistema para el desarrollo de las reglas y funciones del controlador difuso, también para la estimación de la planta para el controlador autoajustable (STR) con lo que se diseñó el controlador PID adaptativo.

Para la evaluación de los reguladores se empleó el método sintético, para integrar las partes que se analizaron y desarrollaron con el método analítico y así se determinó el regulador más eficiente.

El desarrollo que se sigue se puede apreciar en la Figura 14.

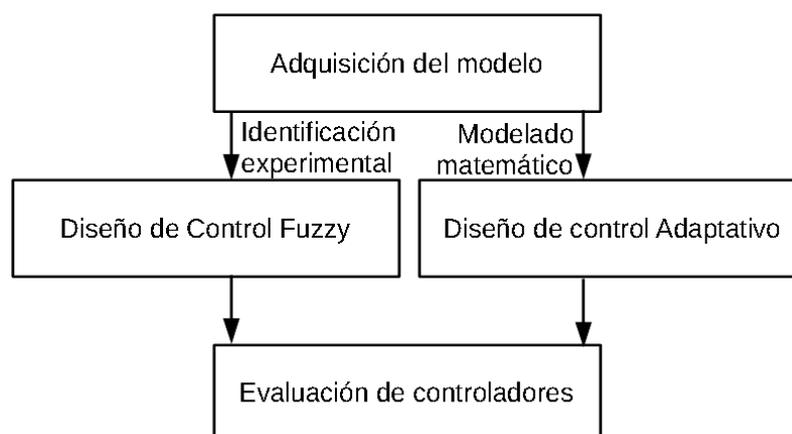


Figura 14 Metodología del proyecto

- En la adquisición del modelo se procedió a utilizar los momentos que rigen el movimiento del sistema, lo que sirve como datos iniciales que necesita el método recursivo para la identificación del controlador adaptativo, paralelo a esto con la identificación experimental se obtuvo el modelo lineal que es útil para el diseño del controlador fuzzy
- Al diseño del control fuzzy clásico con el método de Mamdani, el desarrollo de las funciones de membresía y las reglas, se incorporó el control PID, utilizando la parte proporcional y derivativa del error para el diseño del control fuzzy clásico, utilizando Matlab y el modelo lineal donde se adquirió los valores de las ganancias del controlador PID.

- En el diseño del controlador adaptativo, este es un regulador autoajutable, que utiliza PID como controlador, se implementó el algoritmo de asignación de polos por lo que se tomó un sistema de segundo orden como base para que se realice el control. Con base al modelo matemático del sistema se hallaron los valores iniciales para implementar el método de identificación de RLS-adf; del sistema de segundo orden, se utiliza ξ y w_n para desarrollar el control.
- Para la evaluación de los controladores se tomó en cuenta el tiempo de establecimiento, al 5%, además de la eficiencia que mostró la señal de control de cada controlador

3.1 Modelo del TRMS.

Como se indicó anteriormente para conocer el sistema que se tiene como objetivo su control, se basa en los modelos que se obtendrán mediante el análisis matemático y la identificación del sistema.

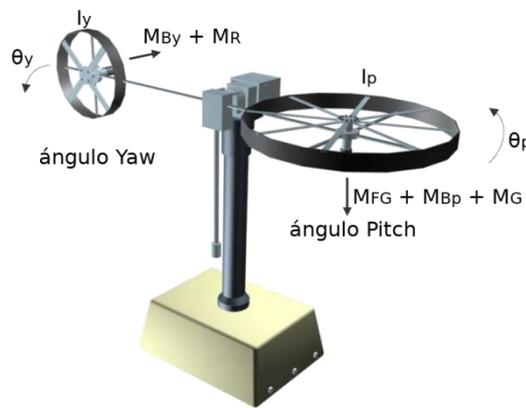


Figura 15 Modelo fenomenológico del TRMS [1].

3.1.1 Modelo matemático del TRMS

El modelo matemático del sistema es desarrollado a partir de la Figura 15. Donde la entrada del sistema son el voltaje de los motores que mueven el eje horizontal y el eje vertical (u_y y u_p respectivamente), estas con un rango de voltaje de 2.5 a -2.5 voltios, mientras que las salidas que se pretende controlar son los ángulos de inclinación de dichos ejes (θ_y para el horizontal y θ_p para el vertical).

El movimiento vertical se define con base a los momentos que la afectan

$$I_p \ddot{\theta}_p = M_1 - M_{FG} - M_{Bp} - M_G$$

Donde

$$\begin{aligned}
M_1 &= a_1 \cdot \tau_1^2 + b_1 \cdot \tau_1 && \text{característica estática no lineal} \\
M_{FG} &= M_G \cdot \sin(\theta_p) && \text{momento de gravedad} \\
M_{Bp} &= B_{1p} \cdot \dot{\theta}_p + \frac{0.0326}{2} \cdot \sin(2\theta_p) \dot{\theta}_p^2 && \text{momento de fuerza de fricción} \\
M_G &= K_{gy} \cdot M_1 \cdot \dot{\theta}_y \cos(\theta_p) && \text{momento giroscópico}
\end{aligned}$$

El momento del motor con base en el motor y el circuito eléctrico se aproxima a una función de primer orden usando la transformada de Laplace [20].

$$\tau_1 = \frac{k_1}{T_{11}s + T_{10}} \cdot u_p$$

Para el movimiento horizontal la ecuación es.

$$I_y \ddot{\theta}_y = M_2 - M_{By} - M_R$$

Dónde:

$$\begin{aligned}
M_2 &= a_2 \cdot \tau_2^2 + b_2 \cdot \tau_2 && \text{característica estática no lineal} \\
M_{By} &= B_{1y} \cdot \dot{\theta}_y && \text{momento de fuerza de fricción}
\end{aligned}$$

El momento de reacción cruzada se aproxima.

$$M_R = \frac{k_c(T_0s + 1)}{T_p s + 1} \cdot M_1$$

El momento del motor es:

$$\tau_2 = \frac{k_2}{T_{21}s + T_{20}} \cdot u_y$$

Se Utiliza el modelo simplificado en espacio de estado que se describen Pratap y Purwaren [20], con lo cual se partirá para el desarrollo del control adaptativo ya que este necesita de la información del sistema para tener un mejor desempeño.

$$\frac{d}{dt}\theta_p = \dot{\theta}_p$$

$$\begin{aligned} \frac{d}{dt}\dot{\theta}_p = & \frac{a_1}{I_p}\tau_1^2 + \frac{b_1}{I_p}\tau_1 - \frac{M_g}{I_p}\sin(\theta_p) + \frac{0.0326}{2I_p}\sin(2\theta_p)\dot{\theta}_y^2 - \frac{B_{1p}}{I_p}\dot{\theta}_p \\ & - \frac{k_{gy}}{I_p}a_1\cos(\theta_p)\dot{\theta}_y\tau_1^2 - \frac{k_{gy}}{I_p}b_1\cos(\theta_p)\dot{\theta}_y\tau_1 \end{aligned}$$

$$\frac{d}{dt}\theta_y = \dot{\theta}_y$$

$$\frac{d}{dt}\dot{\theta}_y = \frac{a_2}{I_y}\tau_2^2 + \frac{b_2}{I_y}\tau_2 - \frac{B_{1y}}{I_y}\dot{\theta}_y - \frac{1.75}{I_y}k_c a_1\tau_1^2 - \frac{1.75}{I_y}k_c b_1\tau_1 \quad (31)$$

$$\frac{d}{dt}\tau_1 = -\frac{T_{10}}{T_{11}}\tau_1 + \frac{k_1}{T_{11}}u_p$$

$$\frac{d}{dt}\tau_2 = -\frac{T_{20}}{T_{21}}\tau_2 + \frac{k_2}{T_{21}}u_y$$

$$y = [\theta_p \ \theta_y]^T$$

Los parámetros físicos del sistema se exponen en la tabla 2

Tabla 2: Parámetros del TRMS [2]

Parámetros	Valores
I_p - momento de inercia del motor vertical	$6.8 \times 10^{-2} \text{ kg m}^2$
I_y - momento de inercia del motor horizontal	$2 \times 10^{-2} \text{ kg m}^2$
a_1 - parámetro de característica estática	0.0135
b_1 - parámetro de característica estática	0.081
a_2 - parámetro de característica estática	0.02
b_2 - parámetro de característica estática	0.09
M_g - momento de gravedad	0.32 N m
B_{1p} - parámetro de la función del momento de fricción	$6 \times 10^{-3} \text{ N m s}^2/\text{rad}$
B_{1y} - parámetro de la función del momento de fricción	$1 \times 10^{-1} \text{ N m s}^2/\text{rad}$
K_{gy} - parámetro del momento giroscópico	0.05 s/radianes
k_1 - ganancia del motor vertical	1.1
k_2 - ganancia del motor horizontal	0.8
T_{11} - parámetro del denominador del motor vertical	1.1
T_{10} - parámetro del denominador del motor vertical	1
T_{21} - parámetro del denominador del motor horizontal	1
T_{20} - parámetro del denominador del motor horizontal	1
T_p - parámetro del momento de reacción cruzada	2
T_0 - parámetro del momento de reacción cruzada	3.5
k_c - ganancia del momento de reacción cruzada	-0.2

3.1.2 Identificación experimental del TRMS.

Se realiza la identificación del sistema mediante la herramienta ident de Matlab, la cual me permite obtener el modelo del sistema dinámico por medio de la recopilación de datos de entrada y salida del TRMS, se la utiliza para la creación del modelo lineal.

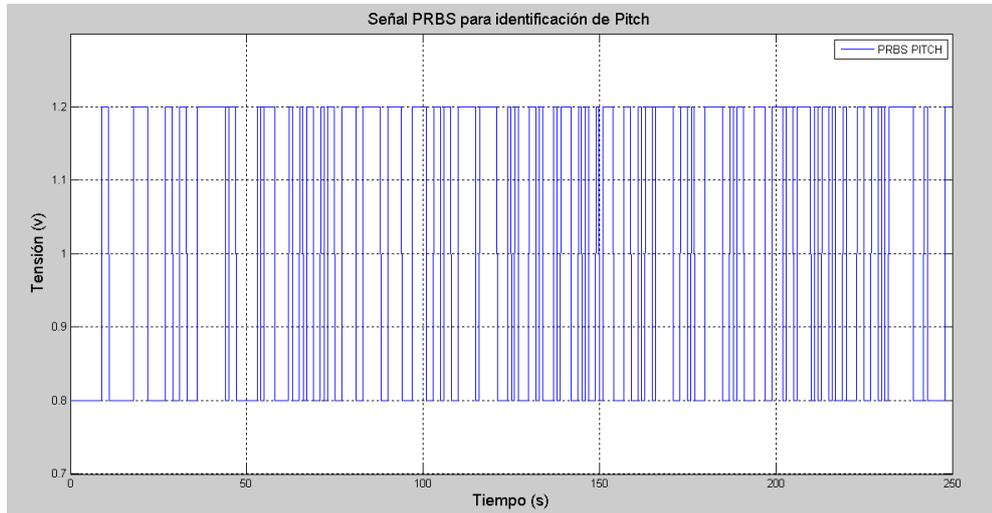


Figura 16 Señal PRBS para identificación de la variable Pitch Principal y Pitch Cruzado.

Para la creación del modelo lineal, es necesario realizar la identificación de los 4 modelos que componen el sistema de doble rotor por medio de una señal de entrada que pueda obtener la mayor cantidad de valores que componen la estructura, en cuyo caso se elige la función PRBS.

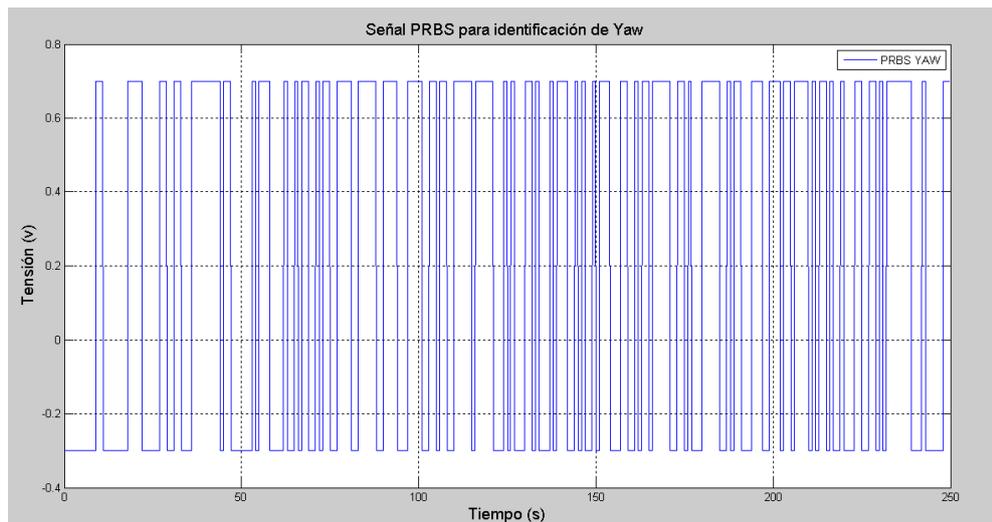


Figura 17 Señal PRBS para identificación de la variable Yaw Principal y Yaw Cruzado.

Se realiza el diseño de la señal PRBS para el motor en Pitch utilizando una función de Matlab dado por el autor Landau y Zito [6]. Para su previa elaboración es necesario conocer el tiempo de subida del sistema a identificar para lo cual tenemos que $T_r = 1.319s$, con esto, mediante la (4) obtenemos los siguientes parámetros $N=10$, $P=4$ y $T_s=0.1s$, además con los datos obtenidos de la (5) se diseña la señal que se muestra en la Figura 16 Señal PRBS para identificación de la variable Pitch Principal y Pitch Cruzado..

Realizando el mismo procedimiento para la entrada del motor en Yaw con $T_r = 2.401s$ se obtiene los siguientes parámetros $N=10$, $P=4$ y $T_s=0.1s$, mostrando la señal resultante en la Figura 17.

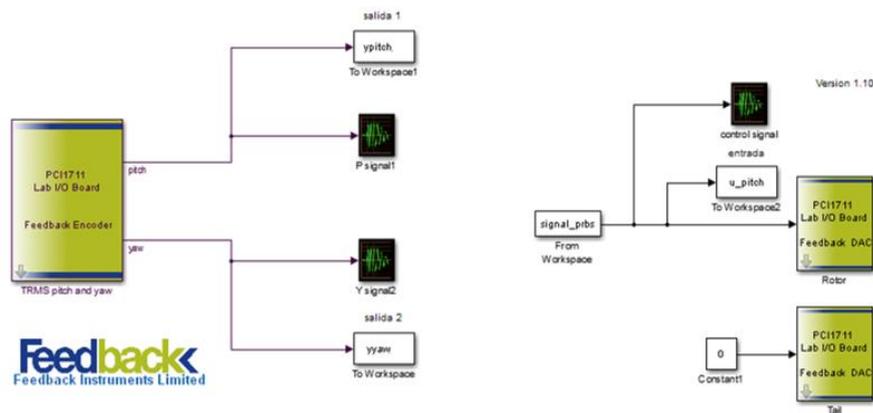


Figura 18 Esquema en Simulink para obtener datos de la variable Pitch principal y cruzada.

3.1.2.1 Modelo Pitch principal y cruzado

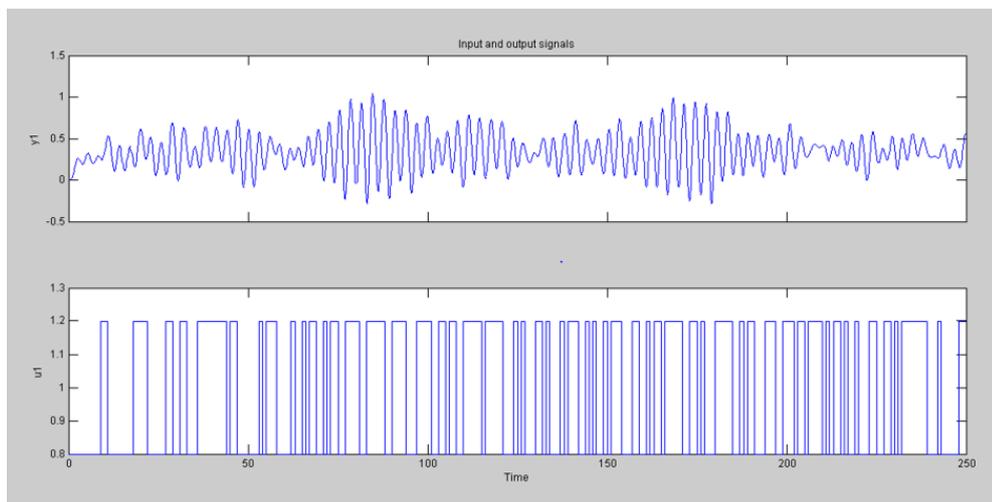


Figura 19 Datos obtenidos experimentalmente para el ángulo principal de Pitch.

Para la identificación del modelo dinámico principal Pitch se realiza la medición de los datos de entrada u_p y de salida θ_p , como se pudo apreciar en la estructura o esquema del TRMS en la Figura 5, existe un acoplamiento cruzado entre las entradas y salidas

del sistema por lo que se desea obviarlo para la obtención de datos, por lo que se hace que la entrada $u_y = 0$. El esquema realizado en Simulink se lo aprecia en la Figura 18 Esquema en Simulink para obtener datos de la variable Pitch principal y cruzada. y los datos obtenidos en la Figura 19.

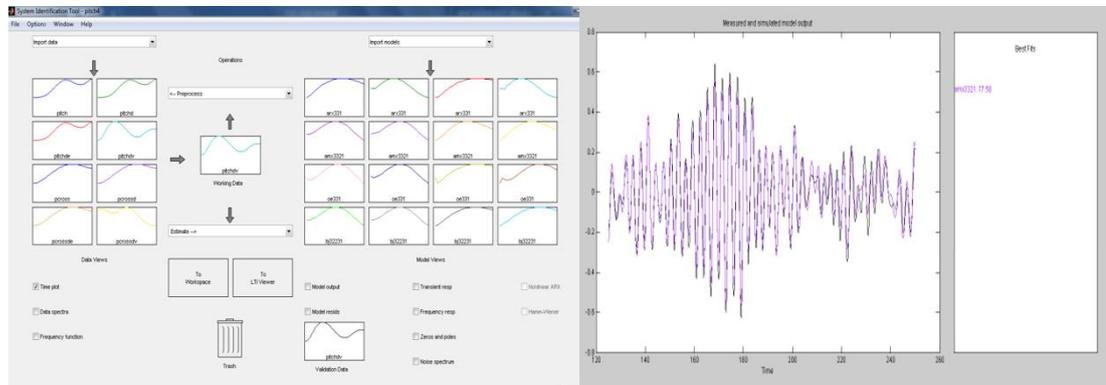


Figura 20 Identificación del modelo Pitch principal en Ident de Matlab.

Con la adquisición de estos datos se obtiene el modelo lineal del sistema mediante la utilización de la herramienta de ident /Matlab con la implementación de modelos polinomiales. (ARX, ARMAX, BJ, OE) vistos en la Tabla 1. Con el modelo paramétrico Autorregresivo de media móvil con una entrada exógena (ARMAX) se obtuvo el mejor desempeño 77.58%, véase Figura 20 Identificación del modelo Pitch principal en Ident de Matlab., con una función de transferencia de 3 polos y 2 ceros mostrada a continuación.

$$\frac{0.01009s^2 + 0.04488s + 2.116}{s^3 + 1.194s^2 + 4.508s + 4.973} \quad (32)$$

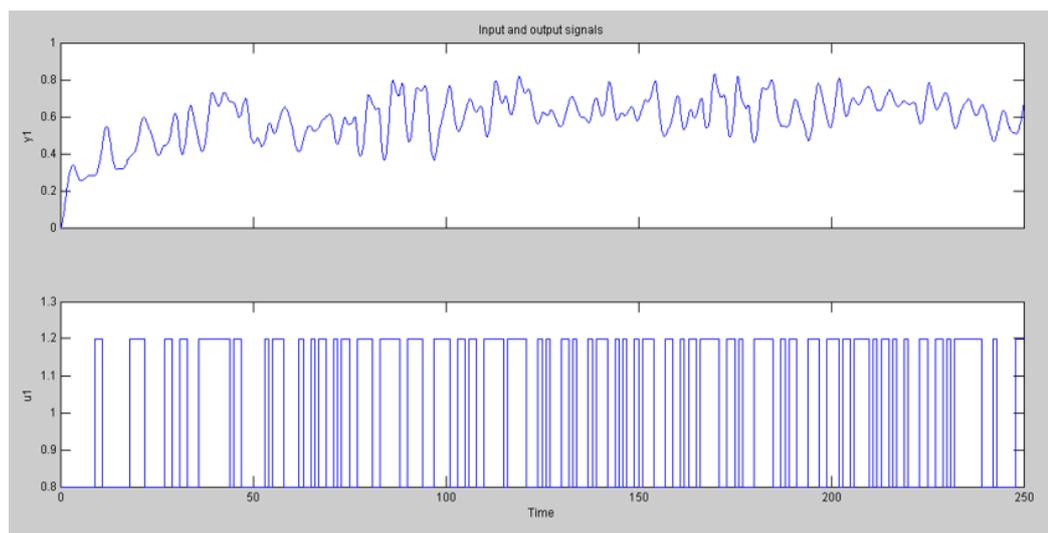


Figura 21 Datos obtenidos experimentalmente para el ángulo cruzado de Pitch.

Para la identificación del modelo dinámico del acoplamiento cruzado de Pitch, se realiza la medición de los datos de entrada u_p y de salida θ_y , se desea eliminar la

iteración entre u_y y θ_y por lo que se desea obviarlo también para la obtención de datos, por lo que se hace que la entrada $u_y = 0$. El esquema realizado en Simulink se lo aprecia en la Figura 18 Esquema en Simulink para obtener datos de la variable Pitch principal y cruzada. y los datos recogidos en la Figura 21 Datos obtenidos experimentalmente para el ángulo cruzado de Pitch..

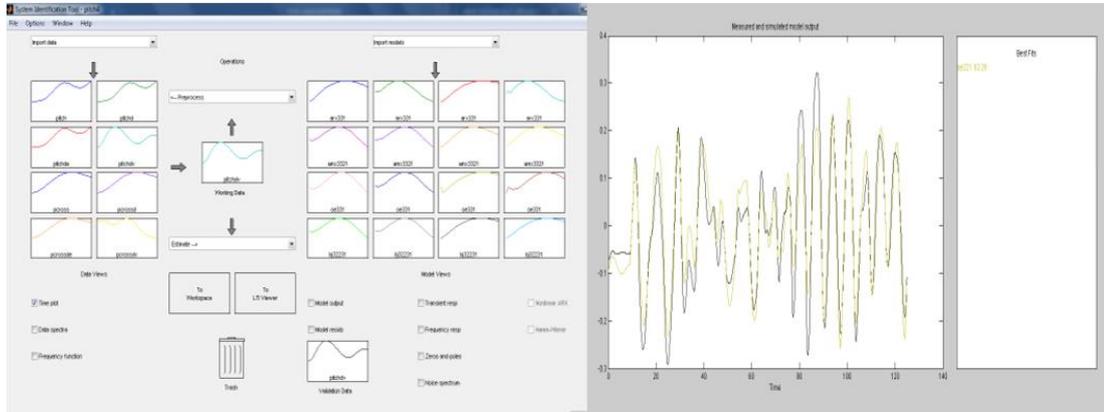


Figura 22 Identificación del modelo Pitch cruzado en Ident de Matlab.

Con la adquisición de estos datos se obtiene el modelo lineal del sistema mediante la utilización de la herramienta de ident /Matlab con la implementación de modelos polinomiales. (ARX, ARMAX, BJ, OE). Con el modelo paramétrico Error de Salida (OE) se obtuvo el mejor desempeño 63.29%, véase Figura 22 Identificación del modelo Pitch cruzado en Ident de Matlab., con una función de transferencia de 2 polos y 1 ceros mostrada continuación.

$$\frac{0.1003s + 0.2129}{s^2 + 0.3164s + 0.7567} \quad (33)$$

3.1.2.2 *Modelo Yaw principal y cruzado*

Para la identificación del modelo dinámico Yaw principal, se realiza la medición de los datos de entrada u_y y de salida θ_y , como existe un acoplamiento cruzado entre las entradas y salidas del sistema por lo que se desea obviarlo también para la obtención de datos, por lo que se hace que la entrada $u_p = 0$. El esquema realizado en Simulink se lo aprecia en la Figura 23 Esquema en Simulink para obtener datos de la variable Yaw principal y cruzada. y los datos recogidos en la Figura 24 Datos obtenidos experimentalmente para el ángulo principal de Yaw..

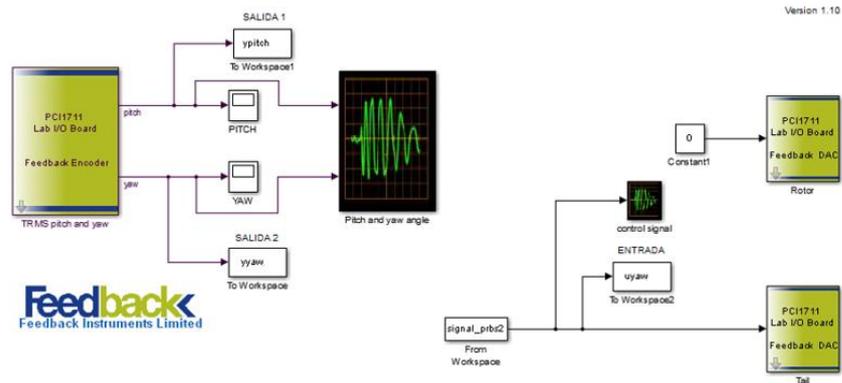


Figura 23 Esquema en Simulink para obtener datos de la variable Yaw principal y cruzada.

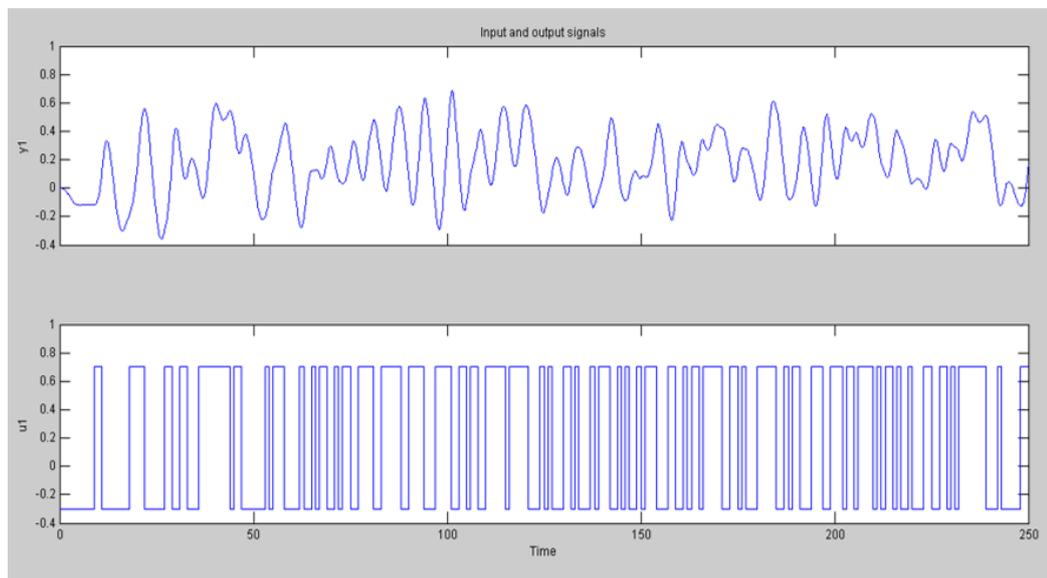


Figura 24 Datos obtenidos experimentalmente para el ángulo principal de Yaw.

Con la adquisición de estos datos se obtiene el modelo lineal del sistema mediante la utilización de la herramienta de ident /Matlab con la implementación de modelos polinomiales. (AR, ARX, ARMAX, BJ, OE).

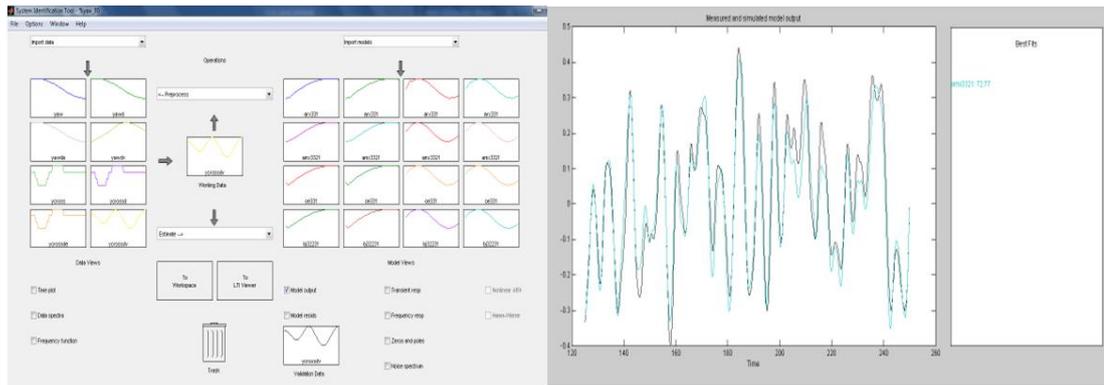


Figura 25 Identificación del modelo Yaw principal en Ident de Matlab.

Con el modelo paramétrico Autorregresivo de media móvil con una entrada exógena (ARMAX) se obtuvo el mejor desempeño 72.77% con una función de transferencia de 3 polos y 2 ceros mostrada en la siguiente ecuación.

$$\frac{0.0009585s^2 + 0.05354s + 0.6662}{s^3 + 2.261s^2 + 1.802s + 1.219} \quad (34)$$

Para la identificación del modelo dinámico del acoplamiento cruzado de Yaw, se realiza la medición de los datos de entrada u_y y de salida θ_p , se desea eliminar la iteración entre u_p y θ_p por lo que se desea obviarlo también para la obtención de datos, por lo que se hace que la entrada $u_p = 0$. El esquema realizado en Simulink se lo aprecia en la Figura 23 Esquema en Simulink para obtener datos de la variable Yaw principal y cruzada. y los datos en la Figura 26 Datos obtenidos experimentalmente para el ángulo cruzado de Yaw..

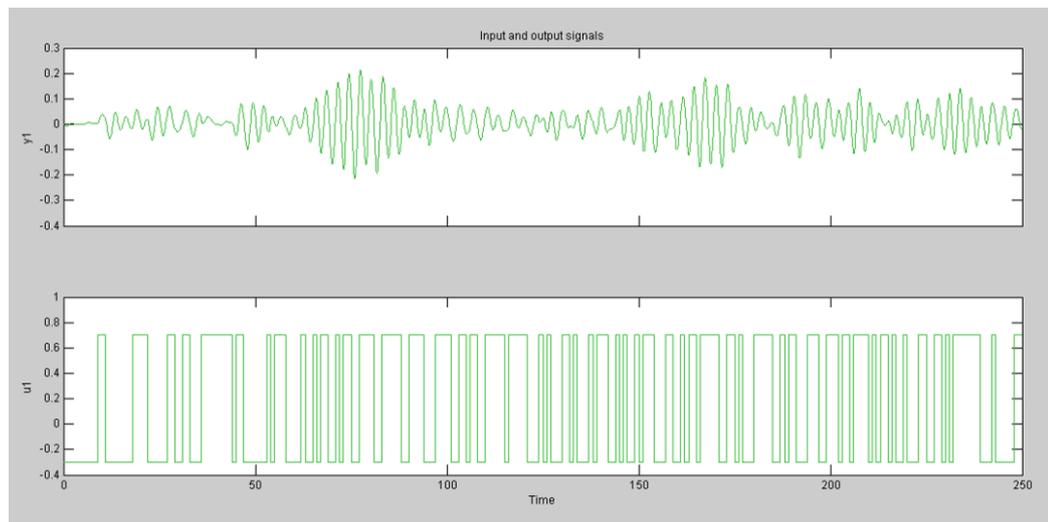


Figura 26 Datos obtenidos experimentalmente para el ángulo cruzado de Yaw.

Con la adquisición de estos datos se obtiene el modelo lineal del sistema mediante la utilización de la herramienta de ident /Matlab con la implementación de modelos polinomiales. (AR, ARX, ARMAX, BJ, OE).

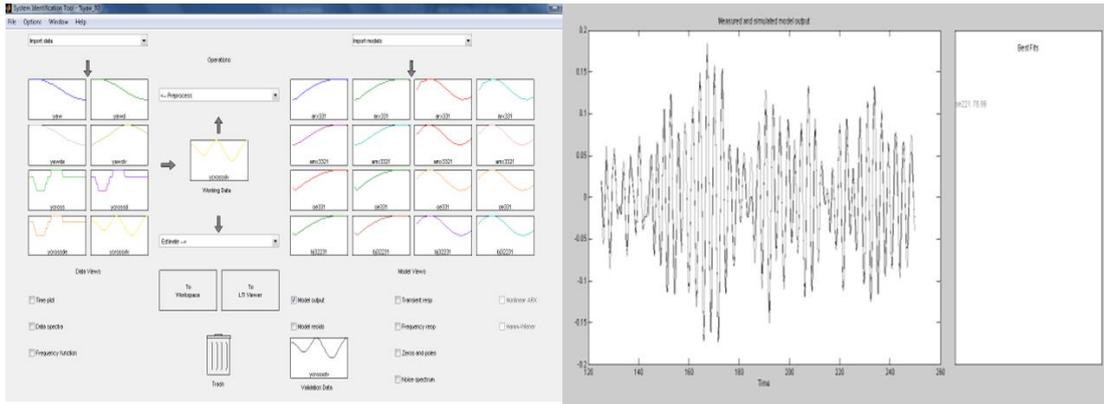


Figura 27 Identificación del modelo Yaw cruzado en Ident de Matlab.

Con el modelo paramétrico Error de Salida (OE) se obtuvo el mejor desempeño 76.99% con una función de transferencia de 2 polos y 1 cero mostrada en la siguiente ecuación.

$$\frac{0.04027s + 0.1465}{s^2 + 0.1156s + 4.826} \quad (35)$$

3.1.2.3 Elaboración del modelo Lineal

Después de encontrar las funciones de transferencia de las 4 variables que intervienen en el sistema, se transfieren los resultados a Simulink para proceder a la respectiva simulación en lazo abierto.

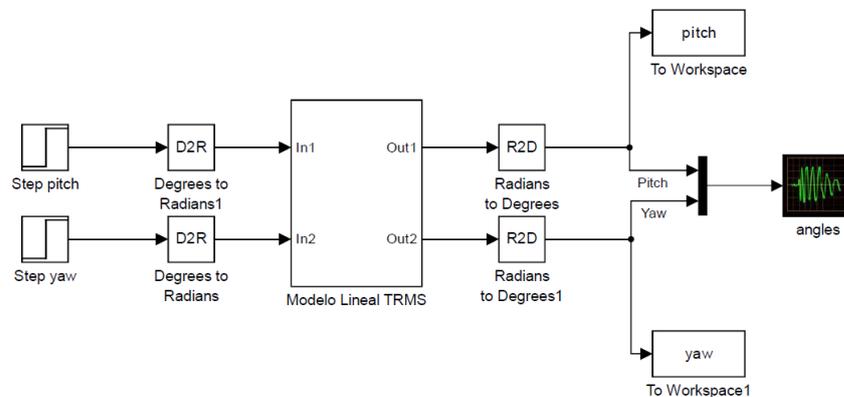


Figura 28 Modelo lineal del TRMS en lazo abierto en Simulink.

El bloque “Modelo Lineal TRMS” se puede observar en Anexo 3. El modelo lineal del TRMS obtenido de los datos experimentales se define en la siguiente función de transferencia del sistema MIMO con base en las ecuaciones (32) a (35).

$$G_{TRMS} = \left\{ \begin{array}{l} \frac{0.01009s^2 + 0.04488s + 2.116}{s^3 + 1.194s^2 + 4.508s + 4.973} \quad \frac{0.04027s + 0.1465}{s^2 + 0.1156s + 4.826} \\ \frac{0.1003s + 0.2129}{s^2 + 0.3164s + 0.7567} \quad \frac{0.0009585s^2 + 0.05354s + 0.6662}{s^3 + 2.261s^2 + 1.802s + 1.219} \end{array} \right\} \quad (36)$$

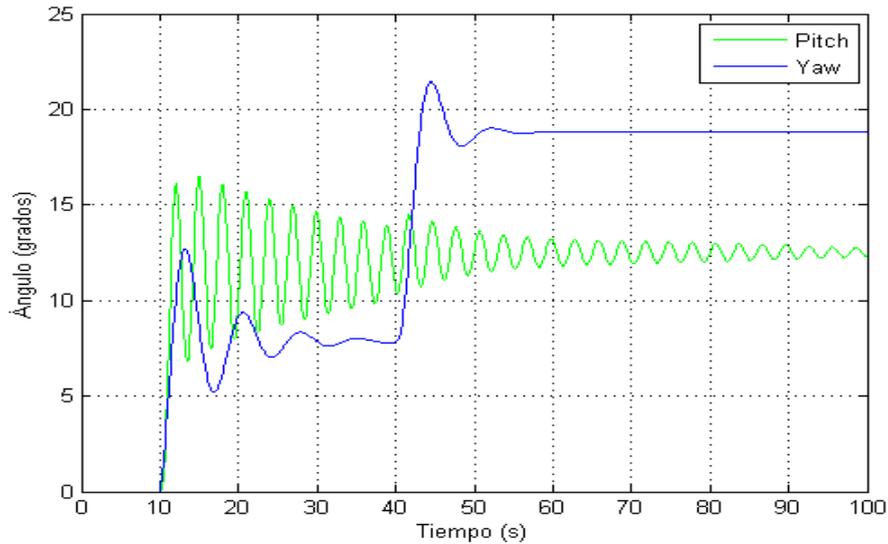


Figura 29 Salida del modelo lineal con una señal de prueba paso en Matlab.

Como se puede observar en la Figura 29, en la respuesta al paso del modelo lineal existe una interacción considerable entre la entrada u_p y la salida θ_y y en la entrada u_y y la salida θ_p existe una interacción mínima, esto demuestra el funcionamiento del sistema TRMS. Para el presente trabajo se desarrolla la creación de controladores descentralizados para el control de los ángulos principales del sistema: Pitch y Yaw por lo que la creación del bloque en Simulink es para mostrar el movimiento cruzado que aparece en ambos grados de libertad cuando funcionan de manera conjunta.

3.2 Diseño del controlador difuso.

3.2.1 Control difuso para Pitch.

Para el diseño del controlador difuso se debe tener en cuenta la cantidad de entradas y salidas a utilizar, en este caso para el control del ángulo Pitch se proceden a utilizar dos entradas que corresponden a la entrada del error (e_{Pitch}) y la segunda que corresponde a la derivada del error (de_{Pitch}) y a su respectiva salida (main-rotor-signal).

Se consideran 9 funciones de membresía para las variables que identificarán a las entradas y se le asignan a cada una las siguientes etiquetas NG, N, NP, NMP, C, PMP, PP, P, PG cuyo significado es descrito a continuación:

Tabla 3: Variables lingüísticas de entrada.

NG	Negativo Grande
N	Negativo
NP	Negativo Pequeño
NMP	Negativo Muy Pequeño
C	Cero
PMP	Positivo Muy Pequeño
PP	Positivo Pequeño
P	Positivo

PG	Positivo Grande
-----------	-----------------

Se consideran 11 funciones de membresía para las variables que identificarán a la salida y se le asignan a cada una las siguientes etiquetas NMG, NG, N, NP, NMP, C, PMP, PP, P, PG, PMG cuyo significado es descrito a continuación:

Tabla 4: Variables lingüísticas de salida.

NMG	Negativo Muy Grande
NG	Negativo Grande
N	Negativo
NP	Negativo Pequeño
NMP	Negativo Muy Pequeño
C	Cero
PMP	Positivo Muy Pequeño
PP	Positivo Pequeño
P	Positivo
PG	Positivo Grande
PMG	Positivo Muy Grande

Las reglas para el controlador son expuestas en la siguiente tabla.

Tabla 5: Reglas para el controlador Difuso.

De/e	NG	N	NP	NMP	C	PMP	PP	P	PG
NG	NMG	NMG	NG	NG	NG	N	NP	NMP	C
N	NMG	NG	NG	N	N	NP	NMP	C	PMP
NP	NG	NG	N	NP	NP	NMP	C	PMP	PP
NMP	NG	N	N	NP	NMP	C	PMP	PP	P
C	NG	N	NP	NMP	C	PMP	PP	P	PG
PMP	N	NP	NMP	C	PMP	PP	P	PG	PG
PP	NP	NMP	C	PMP	PP	P	PG	PG	PMG
P	NMP	C	PMP	PP	P	PG	PG	PMG	PMG
PG	C	PMP	PP	P	PG	PG	PMG	PMG	PMG

En Matlab comenzamos el diseño del controlador fuzzy agregando una entrada adicional para la derivada del error con el método de mandani.

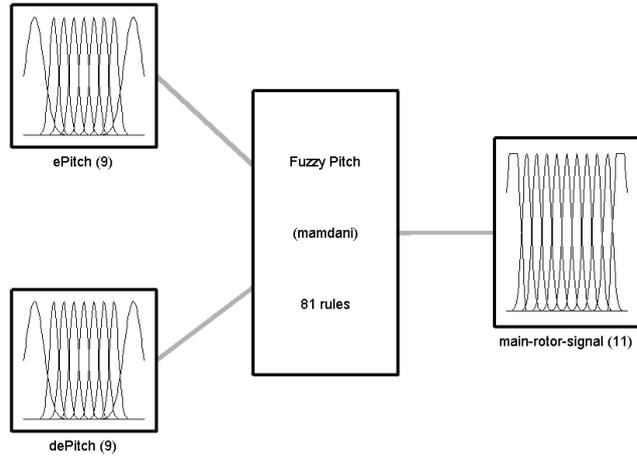


Figura 30 Esquema del controlador difuso para ángulo Pitch.

Se configura las funciones de membresía para el error como se muestran a continuación.

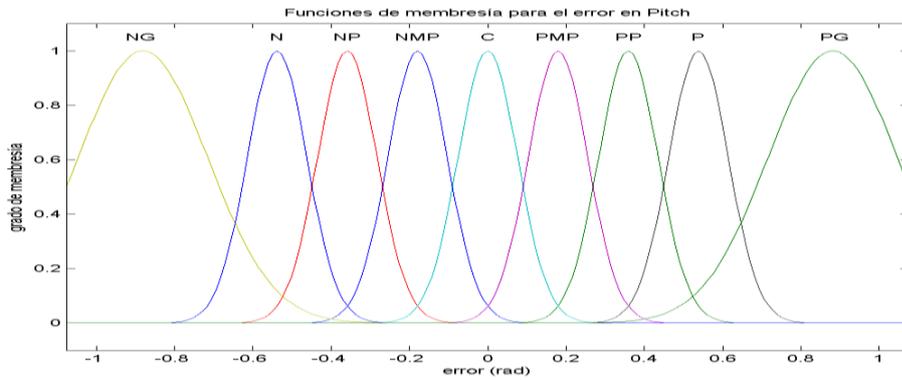


Figura 31 Funciones de membresía de entrada del error para el ángulo Pitch.

Se continúa con las funciones de membresía de la entrada correspondiente a la derivada del error.

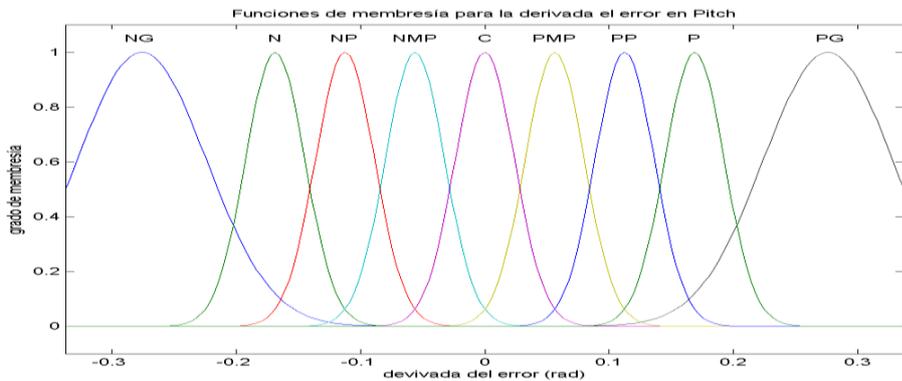


Figura 32 Funciones de entrada de la derivada del error para el ángulo Pitch.

Para las funciones de membresía para la salida que en este caso es el motor para el movimiento pitch o vertical se usan las siguientes.

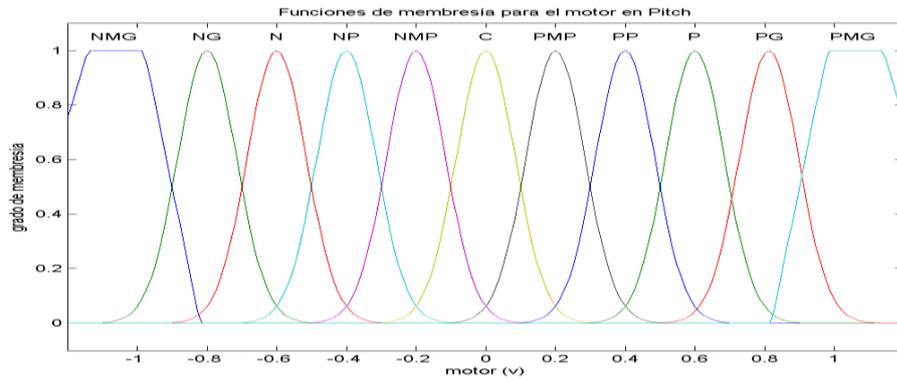


Figura 33 Funciones de membresía de salida para el motor del ángulo Pitch.

También se ingresan las reglas que se detallaron anteriormente, con lo que se finaliza el diseño del controlador fuzzy para pitch, siguiente es la gráfica que detalla la superficie que describe el comportamiento del controlador

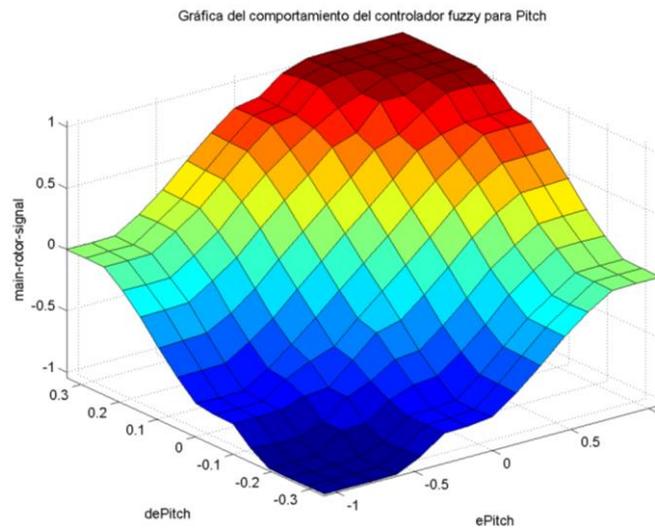


Figura 34 Gráfica de la superficie del controlador difuso para ángulo Pitch.

Valiéndose del esquema del fuzzy PD+I que podemos ver en Figura 11, se diseña el control PID mediante sisotool, utilizando el modelo lineal, con el fin de conseguir reducir el sobre pico y establecer un tiempo de estabilización aceptable.

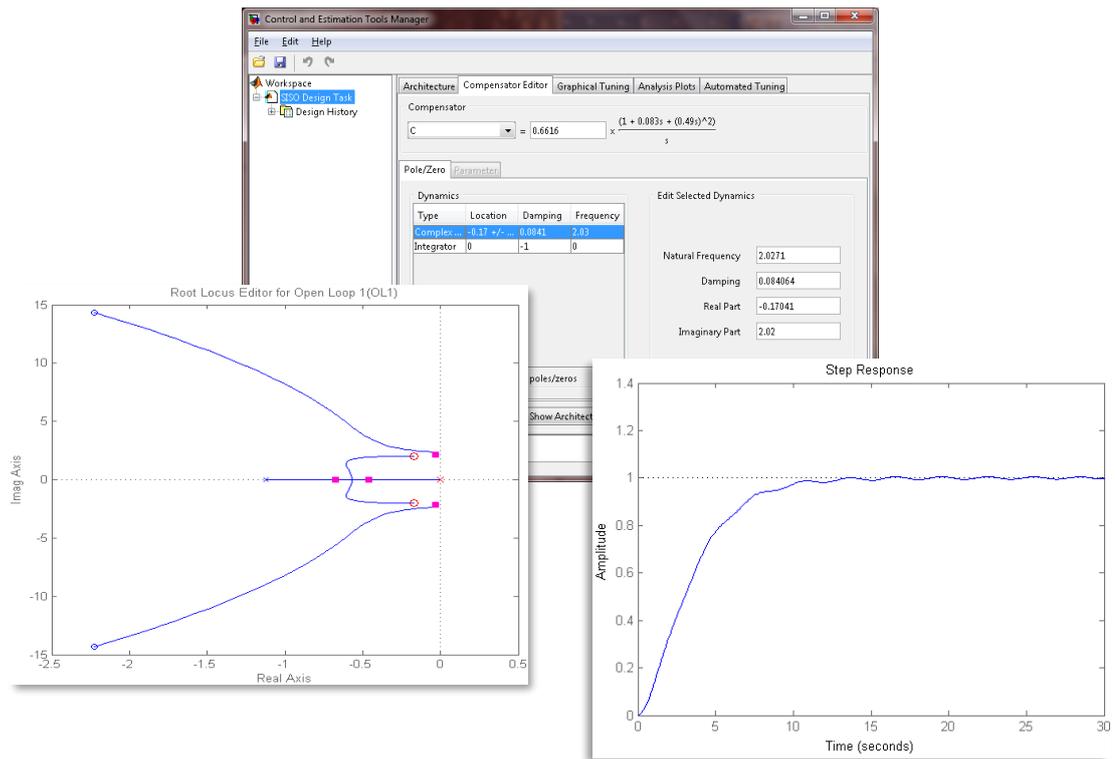


Figura 35 Diseño del control PID para fuzzy en el movimiento Pitch.

Se han los valores característicos del controlador para que con base a (19), obtener los parámetros para el control fuzzy, para simplificar se asume $K_i = 0.8$, para el diseño en cuestión se usaron los valores.

$$K_p = 0.0664; K_{de} = 0.0161; K_u = 0.827 \quad (37)$$

Se realiza la simulación del modelo lineal y del modelo no lineal, este último se describe en el Anexo 4, la implementación en Simulink de estos con el controlador para Pitch se encuentran en la Figura 36 y la Figura 37 respectivamente.

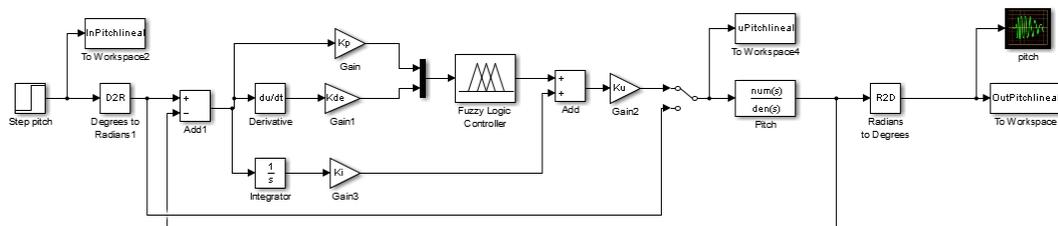


Figura 36 Esquema en Simulink del modelo lineal con controlador Fuzzy Incremental para ángulo Pitch.

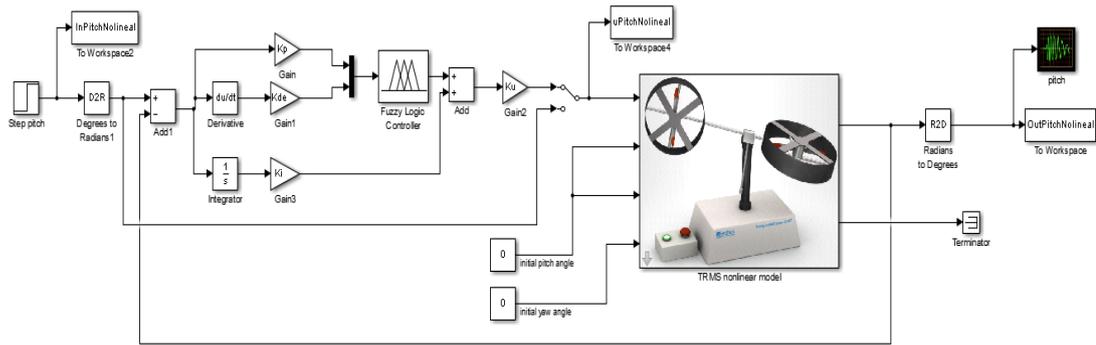


Figura 37 Esquema en Simulink del modelo no lineal con controlador Fuzzy Incremental para ángulo Pitch.

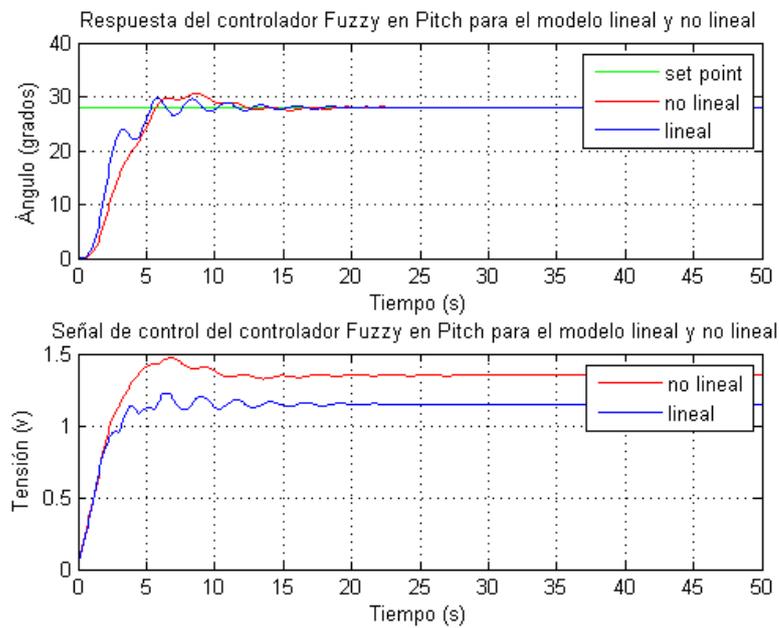


Figura 38 Respuesta a entrada paso y señal de control para ángulo Pitch.

En la Figura 38 se puede apreciar que los resultados obtenidos en la simulación, son favorables debido a que el sobre pico está en un margen aceptable, el desempeño en ambos modelos es aceptable, las pruebas son realizadas para un ángulo de 20° .

3.2.2 Control difuso para Yaw.

Para el diseño del controlador difuso se debe tener en cuenta la cantidad de entradas y salidas a utilizar, en este caso para el control del ángulo Yaw se proceden a utilizar dos entradas que corresponden a la entrada del error (e_{Yaw}) y la segunda que corresponde a la derivada del error (de_{Yaw}) y a su respectiva salida (tail-rotor-signal). Se consideran 13 funciones de membresía para las variables que identificarán a las entradas y se le asignan a cada una las siguientes etiquetas NMMG, NMG, NG, N, NP, NMP, C, PMP, PP, P, PG, PMG, PMMG cuyo significado es descrito a continuación:

Tabla 6: Variables lingüísticas de entrada.

NMMG	Negativo Mucho Más Grande
NMG	Negativo Más Grande
NG	Negativo Grande
N	Negativo
NP	Negativo Pequeño
NMP	Negativo Muy Pequeño
C	Cero
PMP	Positivo Muy Pequeño
PP	Positivo Pequeño
P	Positivo
PG	Positivo Grande
PMG	Positivo Mas Grande
PMMG	Positivo Mucho Más Grande

Se consideran 15 funciones de membresía para las variables que identificarán a la salida y se le asignan a cada una las siguientes etiquetas NDG, NMMG, NMG, NG, N, NP, NMP, C, PMP, PP, P, PG, PMG, PMMG, PDG cuyo significado es descrito a continuación:

Tabla 7: Variables lingüísticas de salida.

NDG	Negativo Demasiado Grande
NMMG	Negativo Mucho Más Grande
NMG	Negativo Más Grande
NG	Negativo Grande
N	Negativo
NP	Negativo Pequeño
NMP	Negativo Muy Pequeño
C	Cero
PMP	Positivo Muy Pequeño
PP	Positivo Pequeño
P	Positivo
PG	Positivo Grande
PMG	Positivo Mas Grande
PMMG	Positivo Mucho Más Grande
PDG	Positivo Demasiado Grande

El diseño respectivo de las reglas para el controlador se lo realiza de la siguiente manera:

Tabla 8: Reglas para el controlador Difuso.

De/e	NMMG	NMG	NG	N	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG
NMMG	NDG	NDG	NDG	NDG	NDG	NDG	NMMG	NMG	NG	N	NP	NMP	C
NMG	NDG	NDG	NDG	NDG	NDG	NMMG	NMG	NG	N	NP	NMP	C	PMP
NG	NDG	NDG	NDG	NDG	NMMG	NMG	NG	N	NP	NMP	C	PMP	PP

N	NDG	NDG	NDG	NMMG	NMG	NG	N	NP	NMP	C	PMP	PP	P
NP	NDG	NDG	NMMG	NMG	NG	N	NP	NMP	C	PMP	PP	P	PG
NMP	NDG	NMG	NMG	NG	N	NP	NMP	C	PMP	PP	P	PG	PMG
C	NMMG	NMG	NG	N	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG
PMP	NMG	NG	N	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG	PDG
PP	NG	N	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG	PDG	PDG
P	N	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG	PDG	PDG	PDG
PG	NP	NMP	C	PMP	PP	P	PG	PMG	PMMG	PDG	PDG	PDG	PDG
PMG	NMP	C	PMP	PP	P	PG	PMG	PMMG	PDG	PDG	PDG	PDG	PDG
PMMG	C	PMP	PP	P	PG	PMG	PMMG	PDG	PDG	PDG	PDG	PDG	PDG

En Matlab comenzamos el diseño del controlador fuzzy agregando una entrada adicional para la derivada del error con el método de mandani.

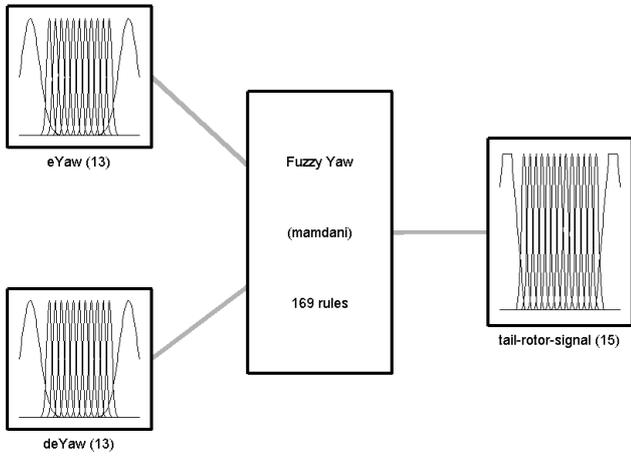


Figura 39 Esquema del controlador difuso para ángulo Yaw.

Se configura las funciones de membresía para el error como se muestran a continuación.

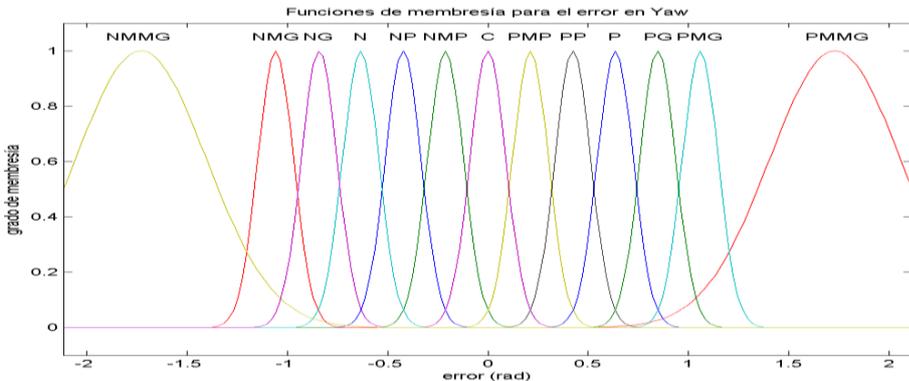


Figura 40 Funciones de membresía de entrada del error para el ángulo Yaw.

Se continúa con las funciones de membresía de la entrada correspondiente a la derivada del error.

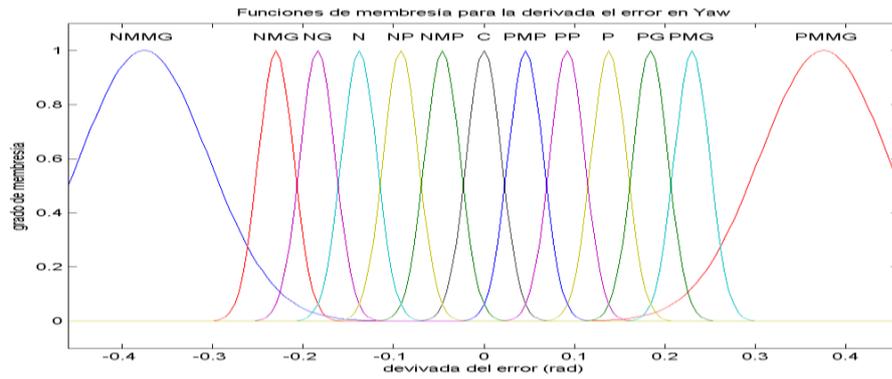


Figura 41 Funciones de entrada de la derivada del error para el ángulo Yaw.

Para las funciones de membresía para la salida que en este caso es el motor para el movimiento pitch o vertical se usan las siguientes.

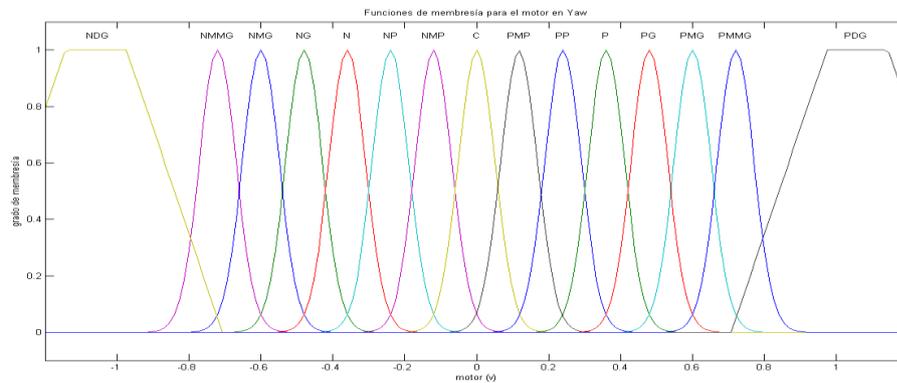


Figura 42 Funciones de membresía de salida para el motor del ángulo Yaw.

También se ingresa las reglas que se detallaron anteriormente, con lo que se finaliza el diseño del controlador fuzzy para Yaw, siguiente es la gráfica detalla la superficie que describe el comportamiento del controlador.

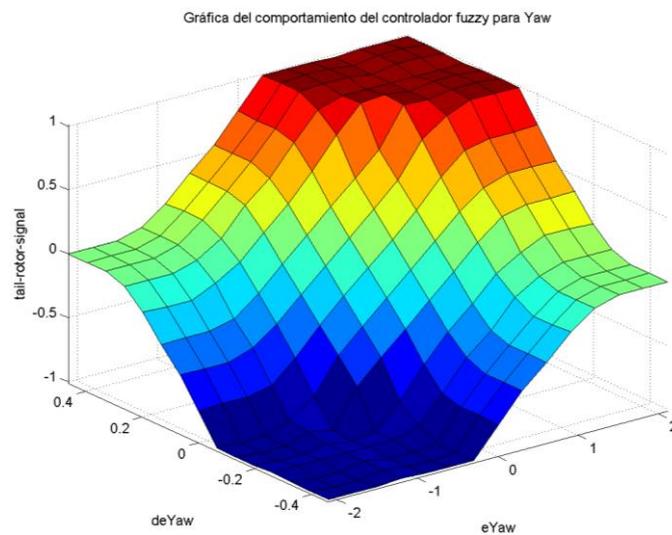


Figura 43 Gráfica de la superficie del controlador difuso para ángulo Yaw.

De igual manera que el diseño del controlador difuso para el ángulo Pitch, también se utiliza sisotool y el modelo lineal de Yaw, con el fin de conseguir reducir el sobre pico y establecer un tiempo de estabilización aceptable.

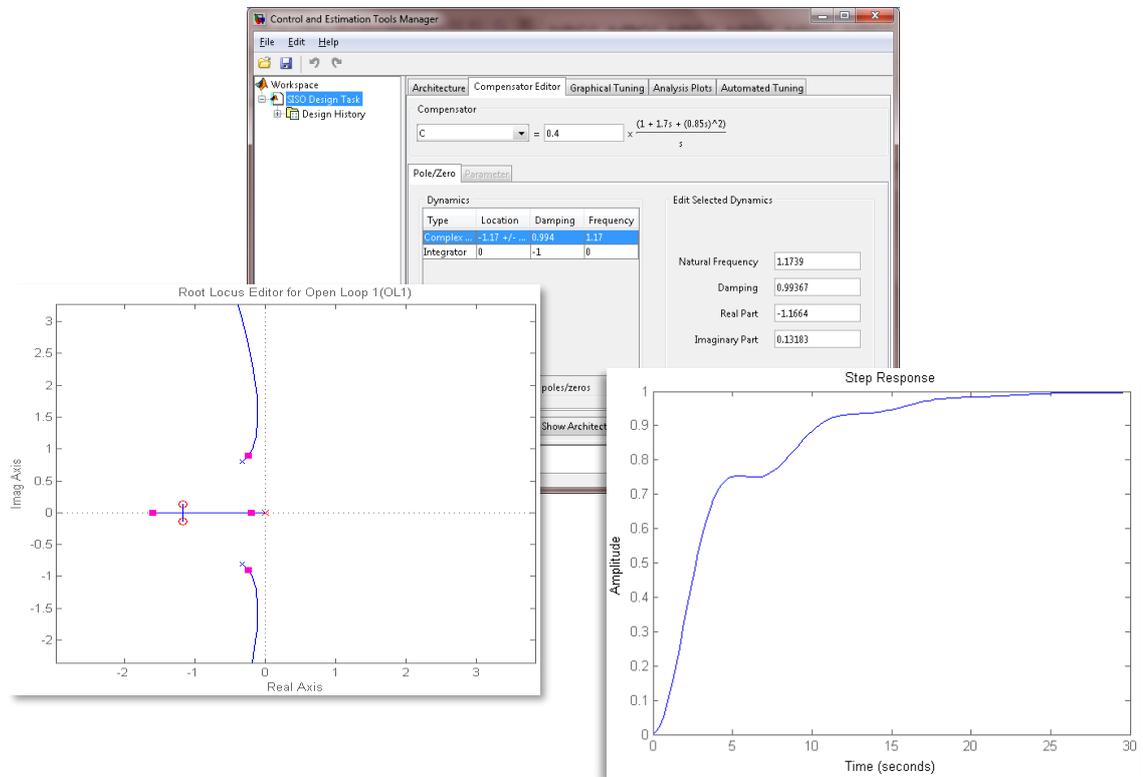


Figura 44 Diseño del control PID para fuzzy en el movimiento Yaw.

Los valores de las constantes K_e , K_{de} y K_u son obtenidos mediante (19), asumiendo $K_i = 1$, para simplificar los cálculos. Para el diseño en cuestión se usaron los valores

$$K_p = 1.693; K_{de} = 1.2286; K_u = 0.4 \quad (38)$$

Se realiza la simulación del modelo no lineal y del modelo lineal, la implementación en Simulink de estos con el controlador para Yaw se encuentra en la Figura 45 y la Figura 46 respectivamente.

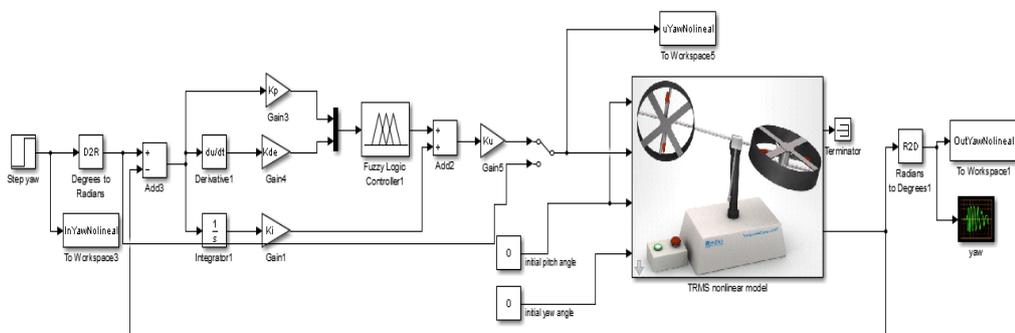


Figura 45 Esquema en Simulink del modelo no lineal con controlador Fuzzy Incremental para ángulo Yaw.

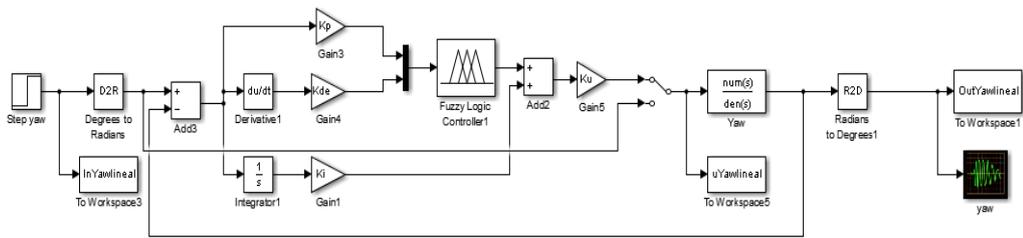


Figura 46 Esquema en Simulink del modelo lineal con controlador Fuzzy Incremental para ángulo Yaw.

En la Figura 47 se puede apreciar que los resultados obtenidos en la simulación son favorables debido a la similitud de sus gráficas con el sistema real, el desempeño del modelo lineal es más cercano al proceso auténtico.

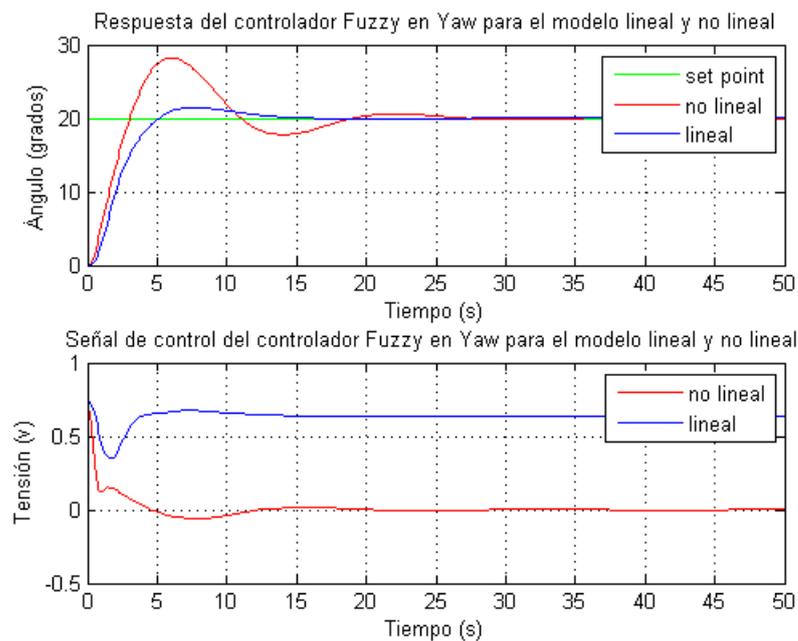


Figura 47 Respuesta a entrada paso y señal de control para ángulo Yaw.

3.3 Diseño del controlador PID adaptativo STR.

3.3.1 Controlador adaptativo para el Pitch

Para los datos iniciales desde el modelo matemático, en (31)(30), el vector de estado $[\theta_p \ d\theta_p \ \theta_y \ d\theta_y \ \tau_1 \ \tau_2]$ es reemplazado por $[x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6]$ entonces se aproxima, para que se utilicen los estados que intervienen en la salida θ_p . Haciendo que $u_y=0$ se simplifica a los estados $[x_1 \ x_2 \ x_5]$; se calcula el punto de equilibrio, para lo cual se cumple que la derivada de los estados $\dot{x} = 0$ [2].

$$\begin{aligned}
u_p &= U_1 \\
x_1 = X_1(U_1) &= \arcsin\left(\frac{k_1}{M_g T_{10}} \left(\frac{a_1 k_1}{T_{10}}\right) U_1^2 + b_1 U_1\right) \\
x_2 &= 0 \\
x_5 &= \frac{k_1}{T_{10}}
\end{aligned}$$

Con base al punto de equilibrio se linealiza con el método extendido [21] al sistema y se obtiene los valores en la representación de espacio de estados.

$$A = \begin{bmatrix} 0 & 1 & 0 \\ -\frac{M_g}{I_p} \cos(X_1(U_1)) & -\frac{B_{1p}}{I_p} & \frac{2a_1}{I_p} X_5(U_1) + \frac{b_1}{I_p} \\ 0 & 0 & -\frac{T_{10}}{T_{11}} \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ \frac{k_1}{T_{11}} \end{bmatrix}$$

$$C = [1 \quad 0 \quad 0]$$

$$D = 0$$

Para llevarlo a función de transferencia se debe de utilizar (2) y se definen variables auxiliares [2]

$$\begin{aligned}
c_1 &= -\frac{M_g}{I_p} \cos(X_1(U_1)) \\
c_2 &= -\frac{B_{1p}}{I_p} \\
c_3 &= -\frac{T_{10}}{T_{11}} \\
c_4 &= \frac{2a_1}{I_p} X_5(U_1) + \frac{b_1}{I_p}
\end{aligned}$$

Con lo que se obtiene la función de transferencia lineal para el movimiento Pitch

$$G_{U_1}(s) = \frac{\frac{k_1}{T_{11}} c_4}{s^3 - s^2(c_2 + c_3) + s(c_3 c_2 - c_1) + c_3 c_1}$$

Experimentalmente obtenemos el valor de linealización, $U_1 = 1.45$ ya que a este valor de voltaje en el movimiento Pitch mantiene el pivote vertical que resulta con la función de transferencia siguiente

$$G_{U_1}(s) = \frac{1.628}{s^3 + 0.9973s^2 + 4.125s + 3.677}$$

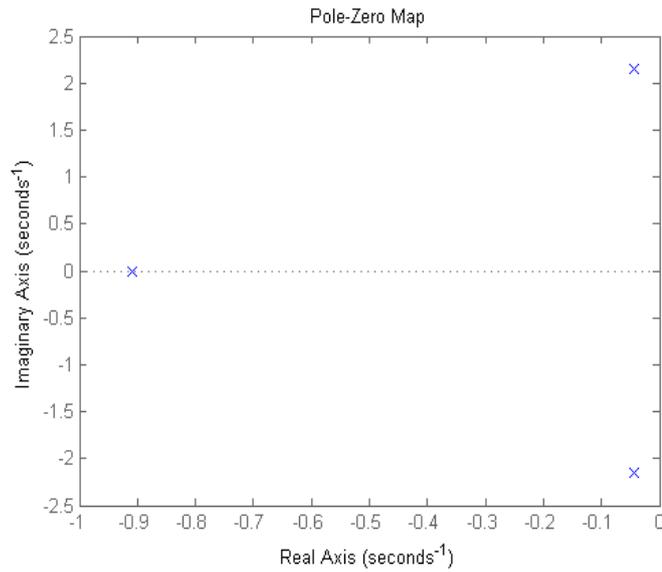


Figura 48 Gráfica de los polos y ceros de la función de transferencia linealizada del Pitch.

Entonces se grafica los polos y ceros de esta en Matlab, mostrada en la Figura 48. En la gráfica se muestra el polo en -0.9 como un polo despreciable, ya que se encuentra muy lejos de los polos dominantes del sistema, entonces se procede a eliminarlo y adecuar la ganancia para desarrollar la ecuación equivalente de segundo orden.

$$G_{U_{1eq}}(s) = \frac{1.791}{s^2 + 0.8824s + 4.045}$$

Con base a esta función de transferencia de segundo orden se discretiza con Matlab con un periodo de muestreo de 0.5, este obtenido al observar el comportamiento del sistema, basado en un margen de error de 5%

$$D_p(z) = \frac{0.2027z + 0.1996}{z^2 - 1.048z + 0.9568} \quad (39)$$

Continuamos con el diseño de los polos para el algoritmo de asignación de polos tomando como referencia $t_s = 4.5$ y $M = 0.1$ y basándose en sus ecuaciones respectivas se obtiene:

$$\xi = 0.9382 ; w_n = 0.9474 \quad (40)$$

Con estos valores es posible la desarrollar del controlador PID STR detallado anteriormente y con la implementación de la librería STCSL se utiliza el bloque pp2b_1.mdl este se detalla desde el Anexo 5 hasta el Anexo 8. Este sería el diagrama con el modelo no lineal.

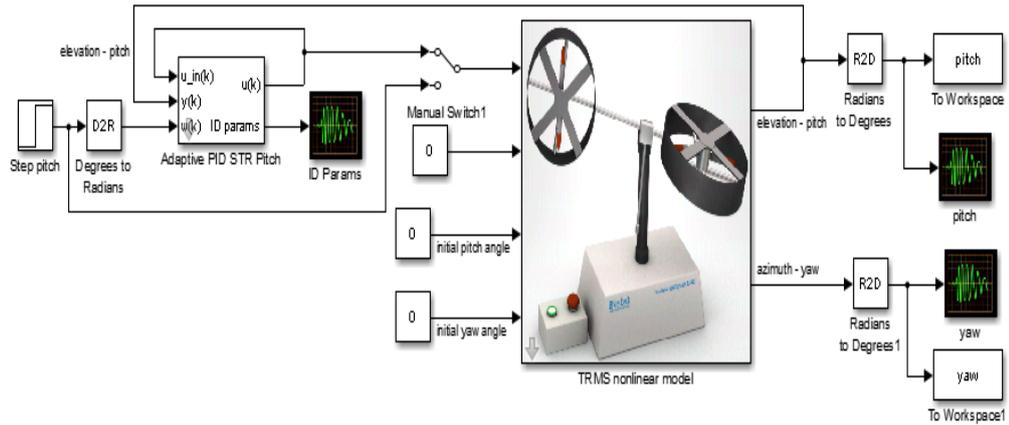


Figura 49 Esquema en Simulink del modelo no lineal con PID adaptativo STR para el ángulo Pitch.

Seguidamente se muestra el diagrama con el modelo lineal obtenido con la identificación experimental.

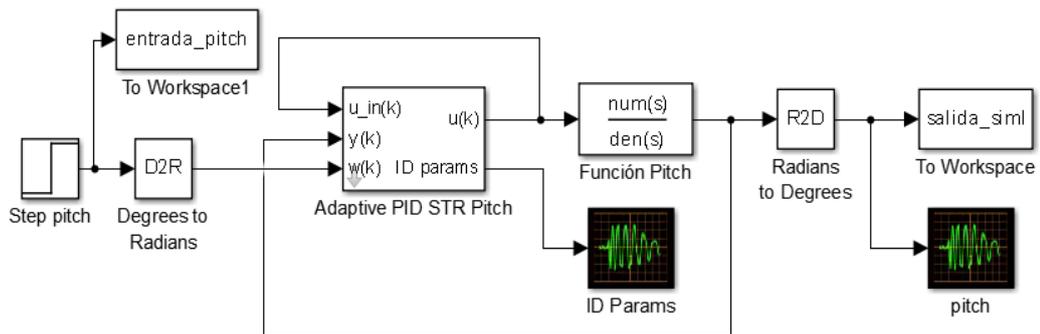


Figura 50 Esquema en Simulink del modelo lineal con PID adaptativo STR para el ángulo Pitch.

La respuesta al paso de la Figura 49 y la Figura 50 se muestra en la Figura 51.

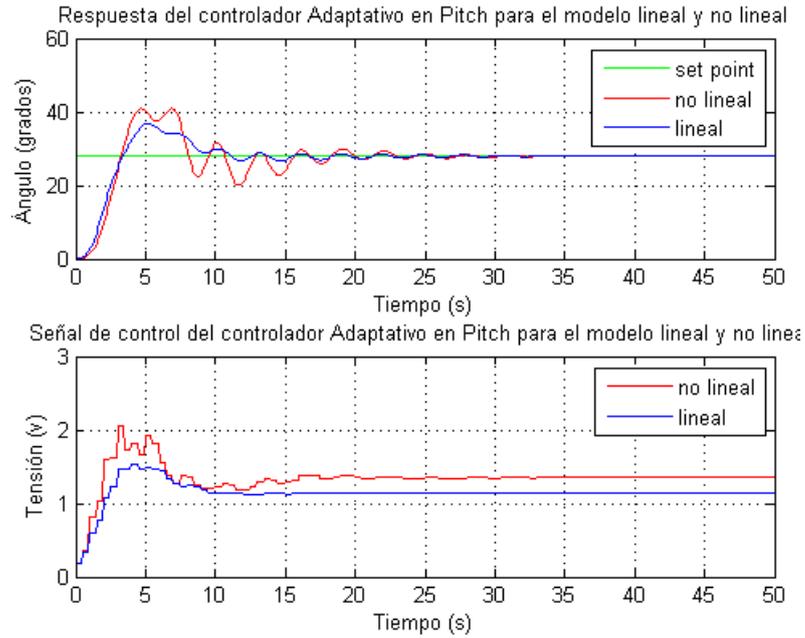


Figura 51 Respuesta a entrada paso con control PID adaptativo STR para ángulo Pitch.

3.3.2 Controlador adaptativo para Yaw

Con el movimiento de Yaw el modelo matemático no representa el sistema [2], por lo que se utiliza la del camino principal de Yaw de la identificación experimental, se observa las características que rigen un sistema de segundo orden para obtener una función de transferencia que podamos discretizar, los datos obtenidos, $t_r = 3.747$, $M=2.1$, y el valor final igual a 2.1875 mediante con los cuales se usa una función de Matlab dado por el autor Landau y Zito [6], esta función se detalla en el Anexo 9, con estos valor obtenidos resulta a la siguiente función de transferencia.

$$G_{yr}(s) = \frac{1.064}{s^2 + 1.155s + 0.4866}$$

Con esta función podemos discretizar con Matlab a un $T_s = 0.1$, basándose en las semejantes observaciones del sistema con respecto a Yaw, con lo que se obtiene.

$$D_y(z) = \frac{0.004489z + 0.004341}{z^2 - 1.9z + 0.9041} \quad (41)$$

Continuamos con el diseño de los polos para el algoritmo de asignación de polos tomando como referencia $t_s = 4.5$ y $M = 0.1$ y basándose en sus ecuaciones respectivas se obtiene:

$$\xi = 0.9382; \omega_n = 0.9474 \quad (42)$$

Igual que se hizo con el Pitch, con estos valores es posible la desarrollar el controlador PID STR, apoyándose en la implementación de la librería STCSL se utiliza el bloque

pp2b_1.mdl cuyo diagrama detallado se pueden observar en los anexos. Este sería el diagrama con el modelo no lineal para Yaw.

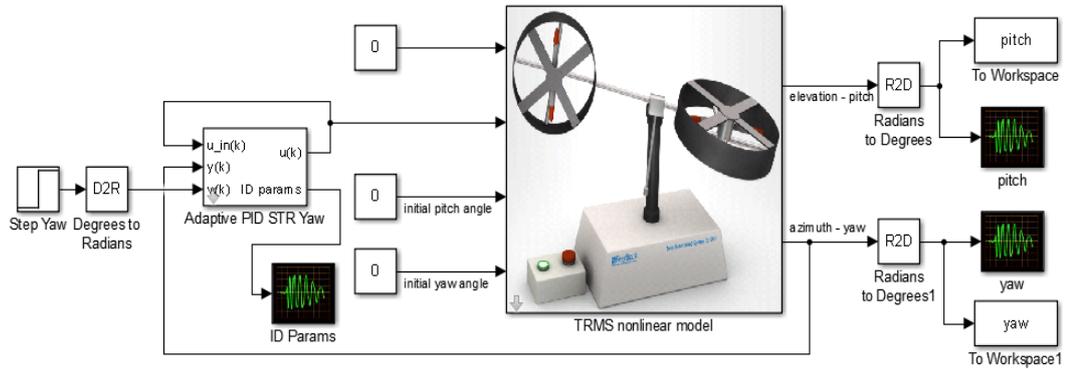


Figura 52 Esquema en Simulink del modelo no lineal con PID adaptativo STR para el ángulo Yaw

Seguidamente se muestra el diagrama con el modelo lineal obtenido con la identificación experimental

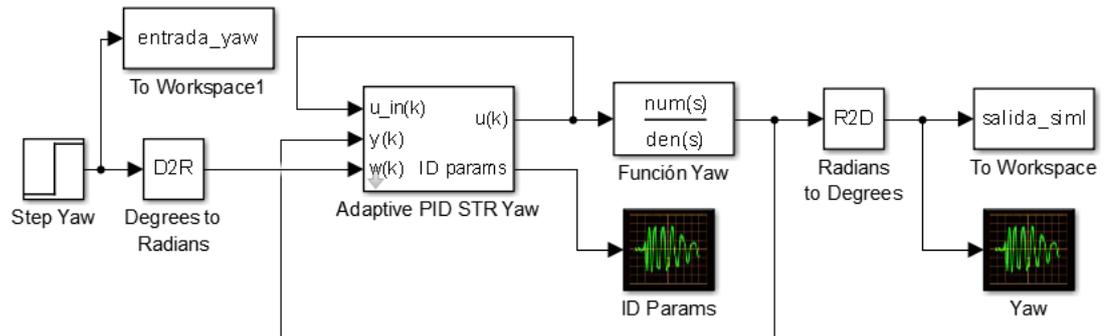


Figura 53 Esquema en Simulink del modelo lineal con PID adaptativo STR para el ángulo Yaw.

La respuesta al paso de la Figura 52 y la Figura 53 se muestra en la Figura 54.

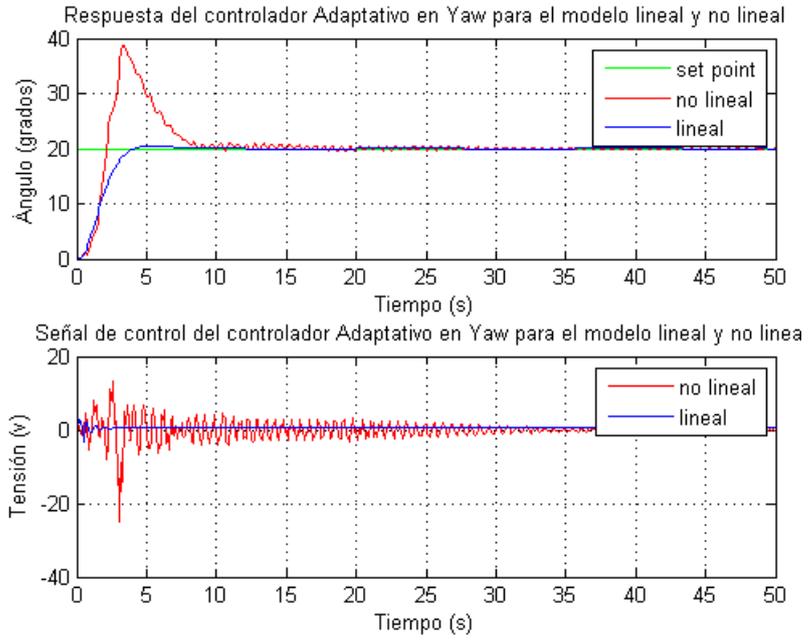


Figura 54 Respuesta a entrada paso con control PID adaptativo STR para ángulo Yaw.

4 RESULTADOS

4.1 Control fuzzy

Para el diseño del controlador fuzzy se realizó 70 funciones de membresía en total, 29 para el movimiento en Pitch y 41 para el movimiento en Yaw. Con respecto a las reglas implementadas son: 81 reglas para Pitch y 169 reglas para el Yaw, es decir 250 reglas para el control del sistema. El resultado de la elaboración de las reglas y funciones de membresía obtenidas para cada ángulo se puede apreciar en el siguiente modelo de superficie.

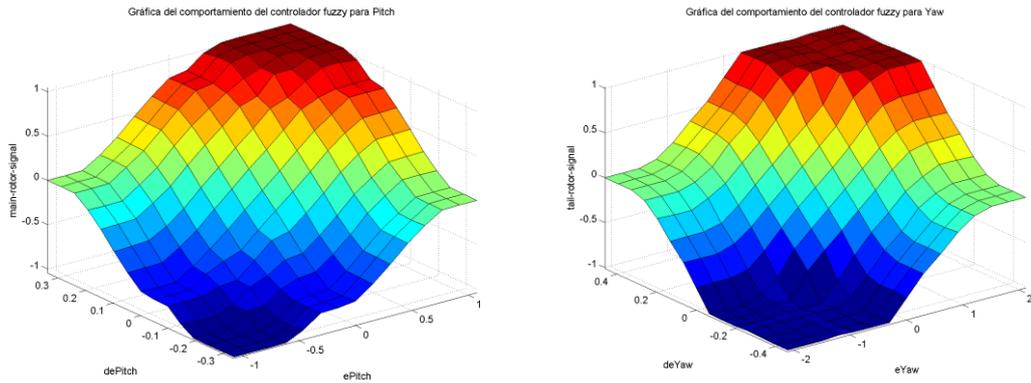


Figura 55 Gráfica de la superficie obtenida para el ángulo Pitch y para el ángulo Yaw.

Al obtener el resultado de la implementación del control fuzzy clásico, véase Figura 56, se observó que es necesaria añadir un control PID en el sistema, por lo que se procede a diseñar con base a la propuesta en la Figura 11, de lo cual se obtuvo los valores que previamente se exponen en (37) y (38).

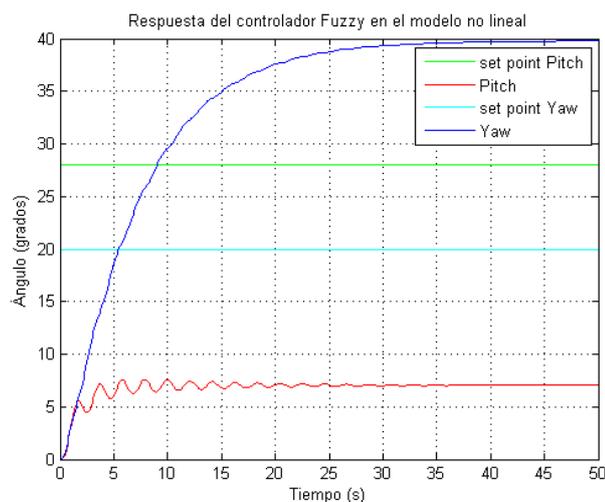


Figura 56 Respuesta del controlador fuzzy clásico en el modelo no lineal

4.2 Controlador autoajustable PID

Los algoritmos para el diseño del controlador se sustentan en los conocimientos anteriormente expuestos en la unidad 2 donde con base la Figura 12, se utiliza el método de mínimos cuadrados recursivos con olvido direccional adaptativo tomando los valores iniciales para el pitch y el yaw de (39) y (41) respectivamente, dando prioridad a estos, desarrollados en la librería utilizada, véase Anexo 6.

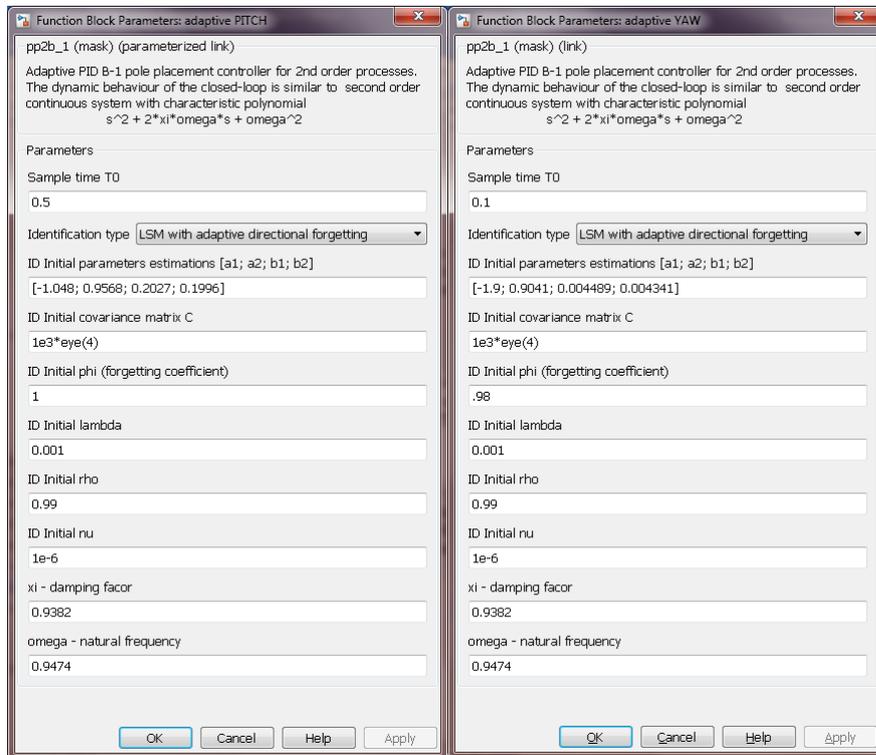


Figura 57 Parámetros utilizados en el controlador PID STR para pitch y yaw

Mientras que para el diseño del controlador se implementó la asignación de polos, véase Figura 13, el cual resulta desde los valores obtenidos de (40) y (42) implementado el algoritmo con la librería utilizada, véase Anexo 7, para así diseñar el control mediante un controlador PID con filtro en la parte derivativa, implementado con ayuda de la librería, véase Anexo 8, los parámetros utilizados se pueden apreciar en la Figura 57.

Los resultados en la evaluación y comparación de los resultados en el modelo real se muestran siguiente unidad.

5 ANÁLISIS DE RESULTADOS

Para mostrar el análisis de los resultados comenzamos con la validación del modelo lineal del sistema el cual permite la comparación del modelo real con el modelo lineal simulado y facilita en el diseño de los controladores.

5.1 Validación de la identificación experimental y del modelo matemático del TRMS

Para la comprobar el modelo lineal elaborado en Ident se procede a realizar la comparación de la respuesta en lazo abierto del sistema real con la simulación del diseño creado en Simulink, para diversas señales de entrada. A continuación, se muestran la respuesta al paso, Figura 58, la respuesta a un tren de pulso, Figura 59, y a una entrada sinusoidal, Figura 60.

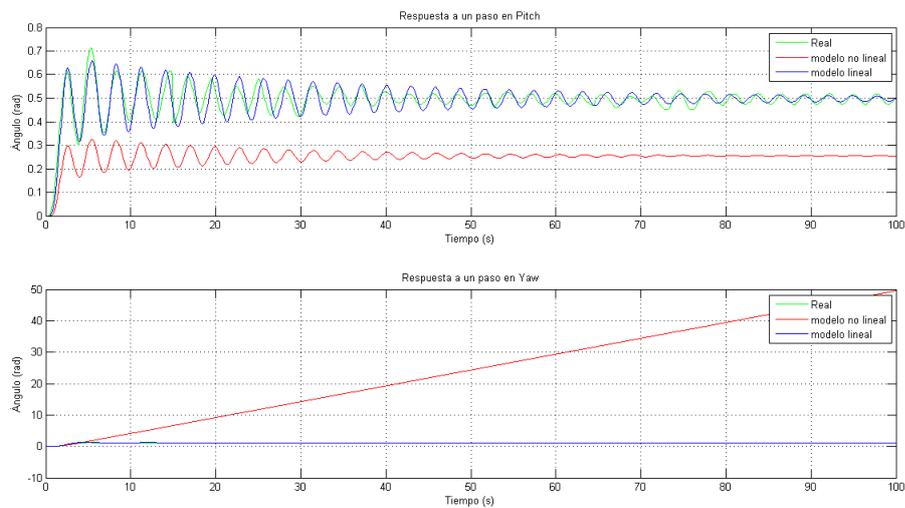


Figura 58 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada paso.

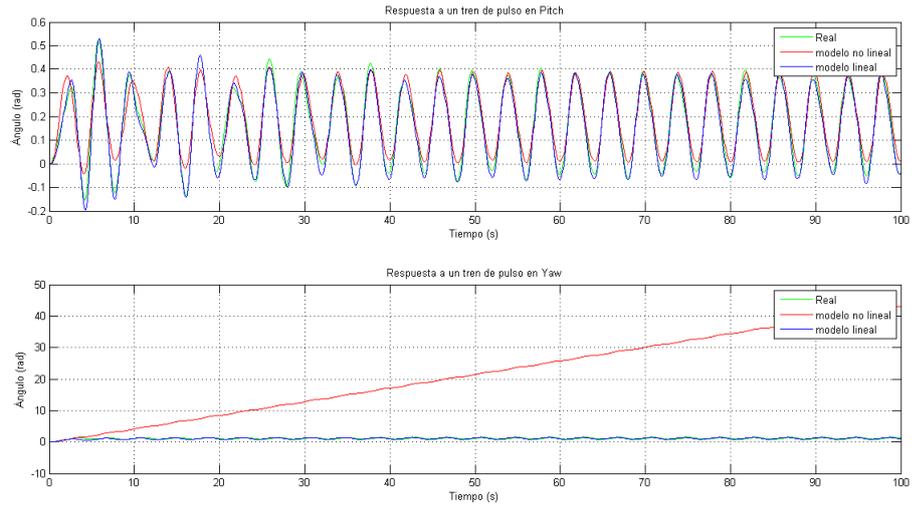


Figura 59 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada Pulso.

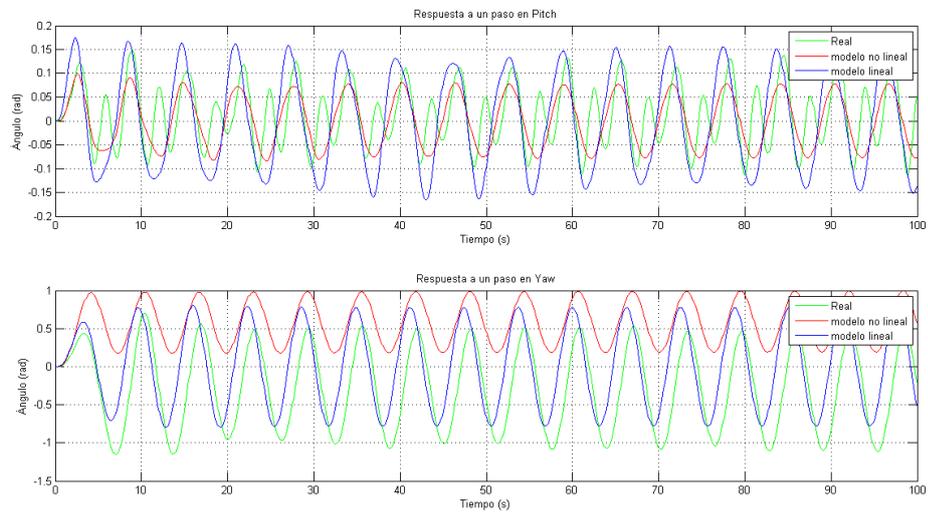


Figura 60 Respuesta obtenida de los ángulos Pitch y Yaw para una entrada sinusoidal.

Con base a los datos obtenidos se analiza con el error RMS propuesto en la (17) con la siguiente tabla.

Tabla 9 Error relativo RMS de el modelado lineal y el no lineal.

Entrada		% RMSE lineal	% RMSE no lineal
Paso	Pitch	6.3992	49.8126
	Yaw	4.1434	2.4181e+03
Tren de pulso	Pitch	13.1340	25.0528
	Yaw	14.0099	2.1179e+03
Sinusoidal	Pitch	675.085	443.8328
	Yaw	190.3952	-333.6983

De los errores calculados se puede observar que el modelo obtenido da una buena aproximación con una entrada paso, mientras que con el tren de pulso y la entrada sinusoidal nuestro modelo obtenido no representa al TRMS, esto debido a que este no es un sistema LTI, mientras que el modelo obtenido sí lo es. Mientras que el modelo no lineal tiene más problemas que el lineal en asemejarse al sistema real con mayor énfasis en el movimiento de Yaw que no representa al sistema.

5.2 Pruebas con el controlador Fuzzy PD+I

Se muestran la respuesta del sistema con el controlador fuzzy descentralizado diseñado para el ángulo Pitch y el ángulo Yaw y la gráfica de la señal de control en la Figura 61.

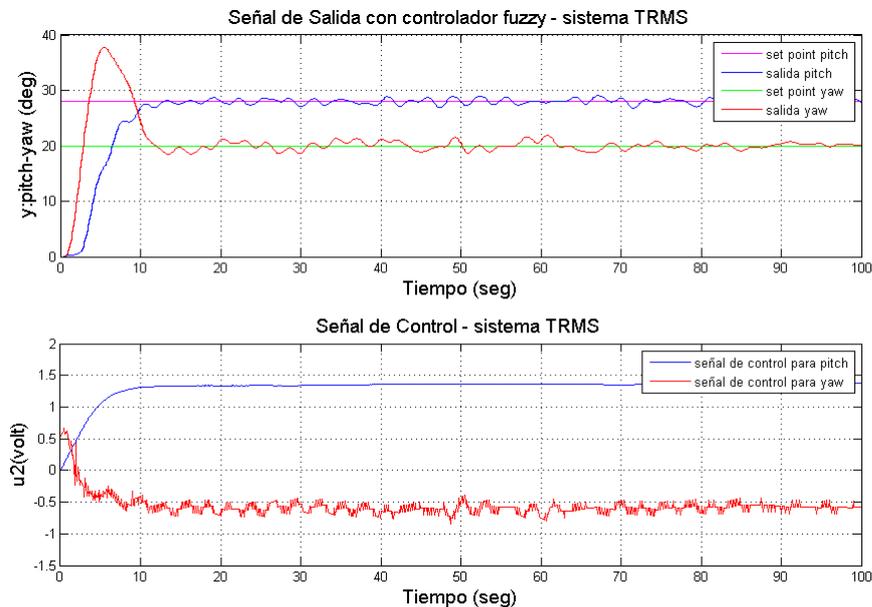


Figura 61 Respuesta al paso con el control Fuzzy PD+I con la señal de control.

Se puede observar que el tiempo de establecimiento aproximado es de 13 segundos para el Pitch y de 18 segundos para el Yaw.

5.3 Pruebas con el controlador PID STR

Después de la elaboración del diseño de los parámetros que influyen en el PID del sistema TRMS, se procede a realizar su implementación para poder verificar el correcto funcionamiento del control.

Se muestran las gráficas de la respuesta del sistema PID STR diseñado para el ángulo Pitch y el ángulo Yaw y la gráfica de la señal de control en la Figura 62.

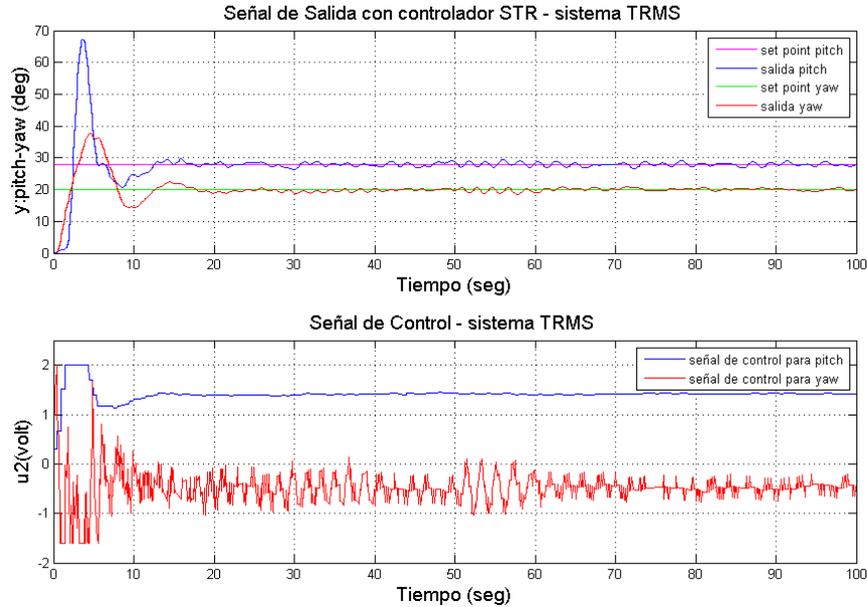


Figura 62 Respuesta al paso con el control PID STR con la señal de control.

Se puede observar que el tiempo de establecimiento aproximado es de 12 s para el Pitch y de 16s para el Yaw.

Tabla 10 Promedio de voltaje de los controladores con una entrada paso.

Ángulo	Voltaje empleado	
	Fuzzy PD+I	PID STR
Pitch	1.3439	1.4231
Yaw	-0.5664	-0.4852

El voltaje en promedio empleado en Pitch es menor para el controlador Fuzzy PD+I, mientras que en el Yaw promedio más bajo de voltaje es el del PID STR.

5.4 Pruebas con entrada escalera

Se muestran las gráficas del Pitch en la Figura 63.

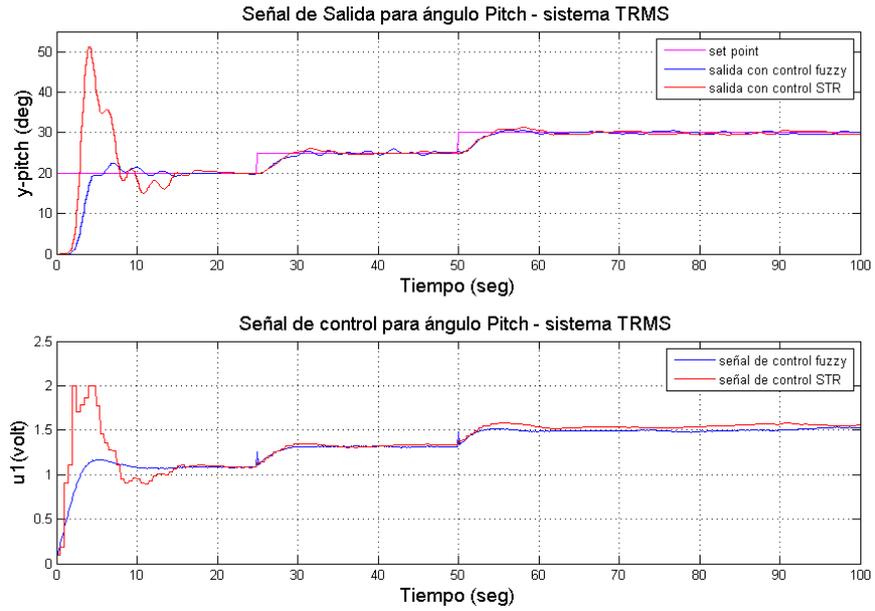


Figura 63 Respuesta a señal escalera del ángulo Pitch con la señal de control.

Se muestran las gráficas del Pitch en la Figura 64.

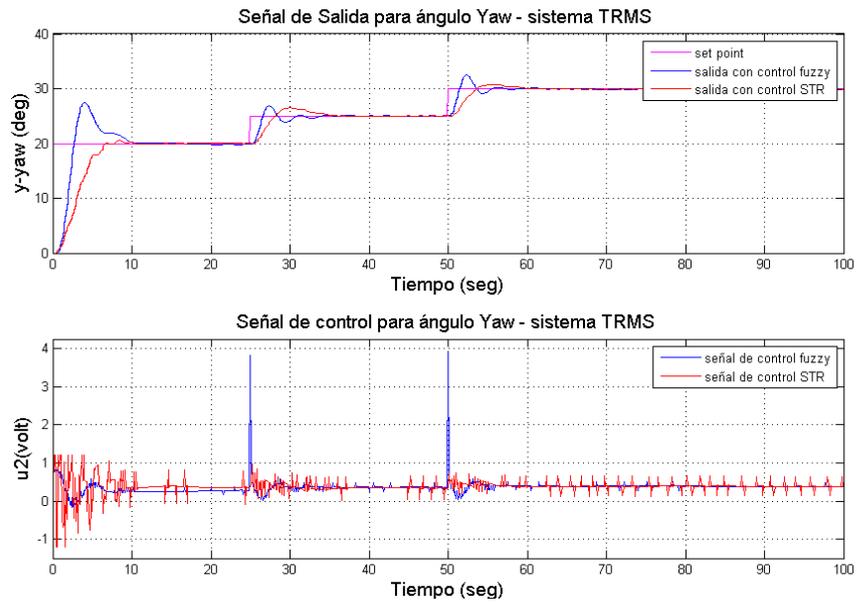


Figura 64 Respuesta a señal escalera del ángulo Yaw con la señal de control.

Tabla 11 Promedio de voltaje de los controladores con una entrada escalera.

Ángulo	Voltaje empleado	
	Fuzzy PD+I	PID STR
Pitch	1.3238	1.3854
Yaw	0.3620	0.3835

El voltaje en promedio empleado en Pitch y para Yaw es menor para el controlador Fuzzy PD+I en este caso.

5.5 Pruebas con perturbaciones

Para comparar la eficiencia de los algoritmos de control propuestos se procede a realizar pruebas con perturbaciones cada 35 segundos en un tiempo de simulación de 100 segundos. La prueba es realizada en ambos ejes.

Se muestran las gráficas obtenidas del ángulo Pitch en la Figura 65.

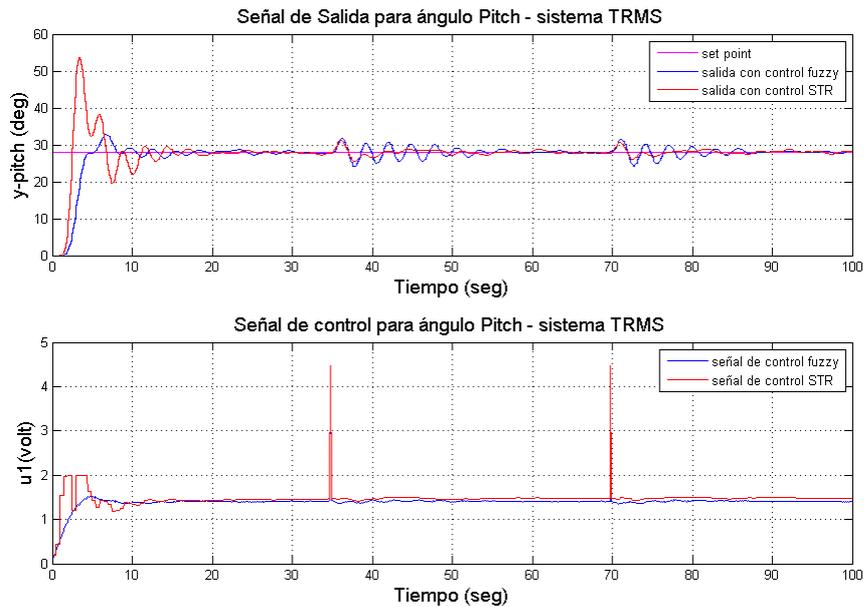


Figura 65 Respuesta con perturbaciones del ángulo Pitch con la señal de control.

Se muestran las gráficas obtenidas del ángulo Yaw en la Figura 66.

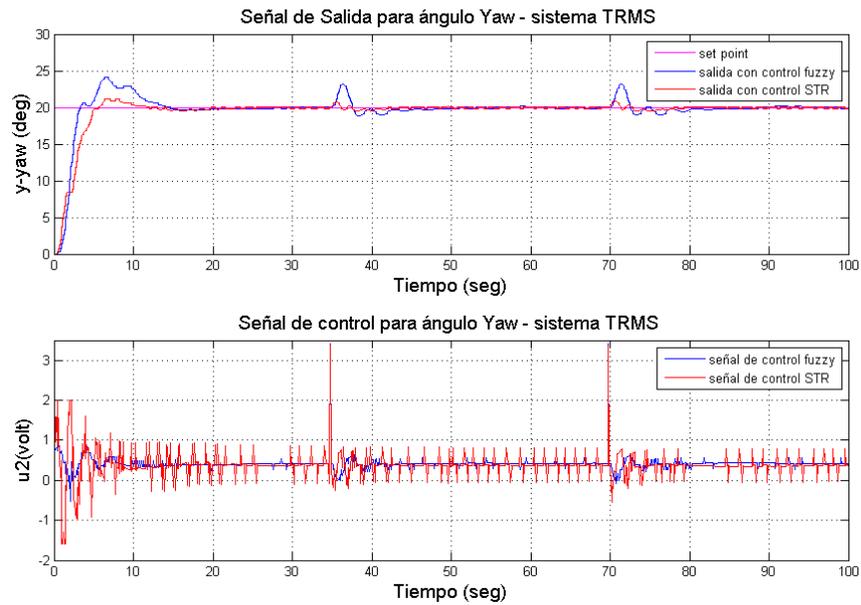


Figura 66 Respuesta con perturbaciones del ángulo Yaw con la señal de control.

Como se puede apreciar en las pruebas realizadas los algoritmos de control diseñados para el PID adaptativo STR son más eficientes debido que el sistema es más estable y más robusto en cuanto a perturbaciones.

Tabla 12 Promedio de voltaje de los controladores con perturbación

Ángulo	Voltaje empleado	
	Fuzzy PD+I	PID STR
Pitch	1.3876	1.4661
Yaw	0.4173	0.3759

El voltaje en promedio empleado en Pitch es menor para el controlador Fuzzy PD+I, mientras que en el Yaw promedio más bajo de voltaje es el del PID STR.

CONCLUSIONES.

Entre las conclusiones se encontró que existe similitud del modelo lineal creado en el presente documento y el modelo no lineal dado por el fabricante en los controladores difusos y adaptativos implementados en los dos ángulos para entradas tipo paso.

Para el diseño del universo discurso del controlador fuzzy se tiene que tener en claro el rango de trabajo que se va a implementar, para cada una de las variables, en este caso el error, la derivada del error y la salida del eje a controlar.

Al observar la respuesta del controlador fuzzy clásico se concluye que para este tipo de sistemas es necesaria la aplicación del PID en el controlador fuzzy, ya que con el fuzzy clásico no es capaz de satisfacer las exigencias que demanda este tipo de procesos.

Las constantes para diseño del PID encontradas pueden ser ajustadas, de acuerdo a la dinámica del proceso, con el fin de obtener una mejor respuesta del sistema.

Para el controlador PID autoajutable es necesario el conocimiento del sistema que se pretende controlar, para la mayor efectividad y rapidez en la estabilización del sistema y la acción del mismo.

El método de asignación de polos en conjunción con el control discreto que se implementa y la dinámica del sistema, resulta en una diferencia significativa en la respuesta del sistema con respecto a los polos diseñados.

Para el funcionamiento en el sistema real se deben aplicar pruebas con rangos pequeños para evitar que el sistema se convierta en inestable. El seguimiento de la referencia de ambos controladores es plausible con las señales de prueba paso como se puede apreciar en sus gráficas de salida.

El controlador Fuzzy, consume menos energía en la acción de control, que el STR, basándose en la utilización de voltaje en la salida en ambos controladores para el seguimiento de referencia.

El controlador PID adaptativo STR es más estable que el controlador Fuzzy a cambios bruscos o perturbaciones externas ya que se va adaptando a la dinámica del sistema.

RECOMENDACIONES.

La identificación experimental del modelo del TRMS puede ser obtenido con diferentes señales de excitación, entre ellas se recomienda para futuros estudios utilizar una PRBS con amplitud variable (APRBS) ya que esta tiene un mejor resultado en sistemas no lineales.

En este proyecto se usa control descentralizado lo que hace que la acción cruzada del sistema TRMS sea despreciada, una opción es el diseño de control centralizado, con esto la acción cruzada sería mejor regulada.

El controlador fuzzy puede ser utilizado acoplado con otro tipo de controlador diferente al PID, entre ellos los algoritmos genéticos puede ser un tema para tomar en investigaciones posteriores.

El control PID para fuzzy en este caso es diseñado mediante sisotool, sin salir del software que se implementa en este proyecto, una alternativa puede ser PID Tuner en Simulink.

Al controlador PID – STR es diseñado mediante el método de asignación de polos, se recomienda investigar con las diversas formas de diseño que este tiene entre ellos el método de Ziegler–Nichols.

Entre las mejoras que se pueden implementar está el diseño de un desacoplador ya que como se mencionó anteriormente la acción cruzada, aunque es poca, perturba el funcionamiento correcto de los controladores diseñados.

En la evaluación de resultados se recomienda que se realicen también en el sistema real ya que los modelados son una aproximación y pueden variar. Con mayor énfasis en sistemas propensos a perturbaciones como el de este proyecto.

Al comparar los controladores se recomienda el diseño e implementación de pruebas que simulen las circunstancias que exigen un esfuerzo a los controladores diseñados.

REFERENCIAS BIBLIOGRÁFICAS.

- [1] Feedback, Manual Twin Rotor MIMO System Control Experiment 33-949S, 2006.
- [2] R. Cajo, Evaluación de los algoritmos de control PID lineal y no lineal, aplicados, Guayaquil: Escuela Superior Politécnica del, 2015.
- [3] K. Ogata, Ingeniería de control moderna, Madrid: Prentice Hall, 2010.
- [4] L. Ljung, System indentificación, New Jersey: Prentice Hall, 1987.
- [5] M. Colmenares, «Sistemas lineales invariantes en el tiempo,» 2 12 2015. [En línea]. Available: https://www.slideshare.net/maricolmenares2/sistemas-lineales-invariantes-en-el-tiempo?nest_slideshow=1. [Último acceso: 16 07 2018].
- [6] I. D. Landau y G. Zito, Digital Control Systems Design, Identification and, London: Springer, 2006.
- [7] O. Nelles, Nonlinear System Identification From Classical Approaches to Neural, Berlin: Springer, 2001.
- [8] MathWork, «What Are Polynomial Models? - MATLAB & Simulink - MathWorks America Latina,» [En línea]. Available: <https://la.mathworks.com/help/ident/ug/what-are-polynomial-models.html>. [Último acceso: 17 07 2018].
- [9] V. Bobál, J. Böhlm, J. Fessl y J. Macháček, Digital Self-tuning Controllers Algorithms, Implementation and Applications, London: Springer, 2005.
- [10] J. Martínez, «Prácticas de Identificación de Sistemas,» Universidad Miguel Hernández de Elche, Alicante, 2005.
- [11] D. Rodríguez y C. Bordóns, «Apuntes de Ingeniería de Control,» 2005.
- [12] V. Bobal y P. Chalupa, Self-tuning controllers simulink library, República Checa: Tomas Bata University in Zlin, 2008.
- [13] T. Chai y R. R. Draxler, «Root mean square error (RMSE) or mean absoluteerror (MAE)? – Arguments against avoiding RMSE in the literature,» *Geosci. Model Dev.*, vol. VII, n° 3, pp. 1247-1250, 2014.
- [14] M. Goerndt, V. J. Monleon y H. Temesgen, «Relating Forest Attributes with Area- and Tree-Based Light Detection and Ranging Metrics for Western Oregon,» *Western Journal of Applied Forestry*, vol. XX, n° 3, pp. 105-111, 2010.
- [15] P. Ponce, Inteligencia Artificial con Aplicaciones a la Ingeniería, Mexico: Alfaomega Grupo Editor, S.A., 2010.
- [16] MathWorks, «Foundations of Fuzzy Logic - MATLAB & Simulink - MathWorks America Latina,» [En línea]. Available: <https://la.mathworks.com/help/fuzzy/foundations-of-fuzzy-logic.html>. [Último acceso: 01 08 2018].
- [17] J. Jantzen, «Tuning Of Fuzzy PID Controllers,» Lyngby, 1998.
- [18] F. Rodriguez y M. López, Control Adaptativo y Robusto, España: Universidad de Sevilla, 1996.

- [19] Tomas Bata University in Zlin, «Self-Tuning Controllers Simulink Library Web link,» [En línea]. Available: <http://www.fai.utb.cz/people/chalupa/stctool/>. [Último acceso: 23 07 2018].
- [20] B. Pratap y S. Purwar, «Sliding mode state observer for 2-DOF twin rotor MIMO system,» de *2010 International Conference on Power, Control and Embedded Systems*, 2010.
- [21] H. Sira-Ramírez, Control de sistemas no lineales: linealización aproximada, extendida, exacta, Madrid: Prentice Hall, 2005.

ANEXOS

HEDS-9000/9100

Two Channel Optical Incremental Encoder Modules



Data Sheet



Description

The HEDS-9000 and the HEDS-9100 series are high performance, low cost, optical incremental encoder modules. When used with a codewheel, these modules detect rotary position. The modules consist of a lensed (LED) source and a detector IC enclosed in a small C-shaped plastic package. Due to a highly collimated light source and unique photodetector array, these modules are extremely tolerant to mounting misalignment.

The two channel digital outputs and the single 5 V supply input are accessed through five 0.025 inch square pins located on 0.1 inch centers.

Standard resolutions for the HEDS-9000 are 500 CPR and 1000 CPR for use with a HEDS-6100 codewheel or equivalent.

For the HEDS-9100, standard resolutions between 96 CPR and 512 CPR are available for use with a HEDS-5120 codewheel or equivalent.

Features

- High performance
- High resolution
- Low cost
- Easy to mount
- No signal adjustment required
- Small size
- -40°C to 100 °C operating temperature
- Two channel quadrature output
- TTL compatible
- Single 5 V supply

Applications

The HEDS-9000 and 9100 provide sophisticated motion detection at a low cost, making them ideal for high volume applications. Typical applications include printers, plotters, tape drives, and factory automation equipment.

Note: Avago Technologies encoders are not recommended for use in safety critical applications. Eg. ABS braking systems, power steering, life support systems and critical care medical equipment. Please contact sales representative if more clarification is needed.

Anexo 1 Encoder HEDS-9000/9100.

Theory of Operation

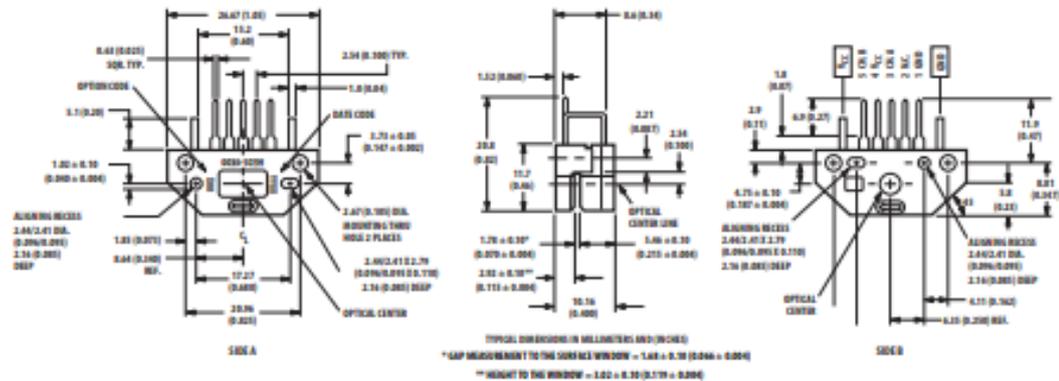
The HEDS-9000 and 9100 are C-shaped emitter/detector modules. Coupled with a codewheel, they translate the rotary motion of a shaft into a two-channel digital output.

As seen in the block diagram, each module contains a single Light Emitting Diode (LED) as its light source. The light is collimated into a parallel beam by means of a single polycarbonate lens located directly over the LED. Opposite the emitter is the integrated detector circuit. This IC consists of multiple sets of photodetectors and the signal processing circuitry necessary to produce the digital waveforms.

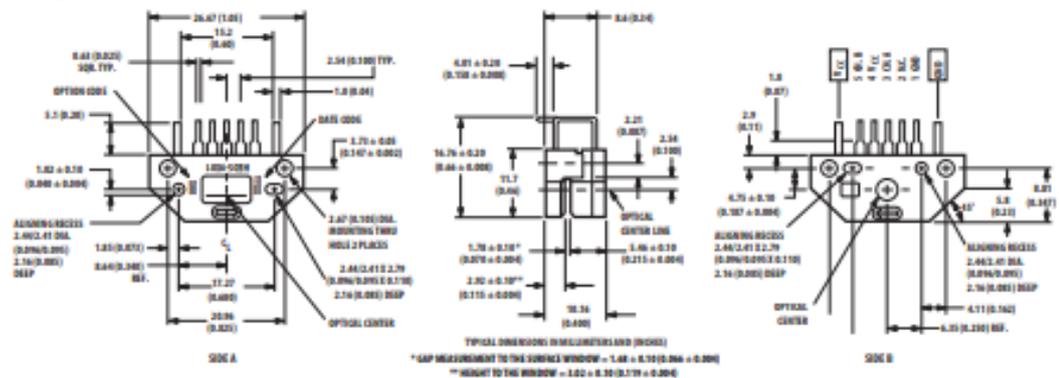
The codewheel rotates between the emitter and detector, causing the light beam to be interrupted by the pattern of spaces and bars on the codewheel. The photodiodes which detect these interruptions are arranged in a pattern that corresponds to the radius and design of the codewheel. These detectors are also spaced such that a light period on one pair of detectors corresponds to a dark period on the adjacent pair of detectors. The photodiode outputs are then fed through the signal processing circuitry resulting in A, \bar{A} , B, and \bar{B} . Two comparators receive these signals and produce the final outputs for channels \bar{A} and \bar{B} . Due to this integrated phasing technique, the digital output of channel A is in quadrature with that of channel B (90 degrees out of phase).

Package Dimensions

HEDS-9x00



HEDS-9x01



Anexo 1 Encoder HEDS-9000/9100.

```

function prbs = create_prbs(ValUinit, ValAmpli, ValDecal,
ValLgReg, ValDivi, Nsamp, Tappli)
%
%
% CREATE_PRBS is used for the generation of a PRBS signal
%
% prbs = create_prbs(ValUinit, ValAmpli, ValDecal,
ValLgReg, ValDivi, Nsamp, Tappli)
%
% "Entry parameters" are :
% ValUinit : Initial steady state
% ValAmpli : Magnitude
% ValDecal : Add-on DC component
% ValLgReg : Register length
% ValDivi : Frequency divider
% Nsamp : Number of samples
% Tappli : Application instant
%
% the initialization is performed

k1 = ValLgReg - 1;
k2 = ValLgReg;

switch(ValLgReg)
case 5
k1 = 3;
case 7
k1 = 4;
case 9
k1 = 5;
case 10
k1 = 7;
case 11
k1 = 9;
end

for i=1:11
sbpa(i) = 1;
end

% After init-phase PRBS algo is running

% Output set to init-value until the PRBS application instant

for i=1:Tappli
prbs(i) = ValUinit;
end

```

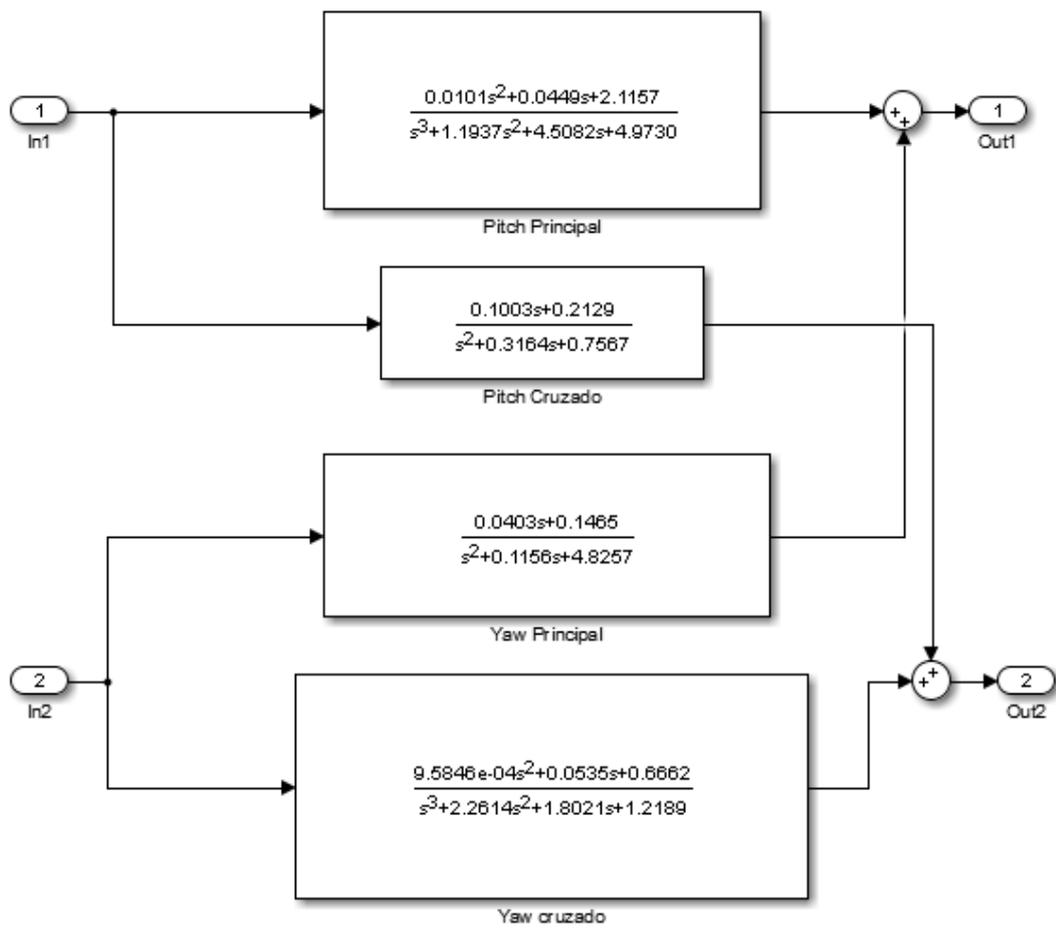
Anexo 2 Función create_prbs.m.

```

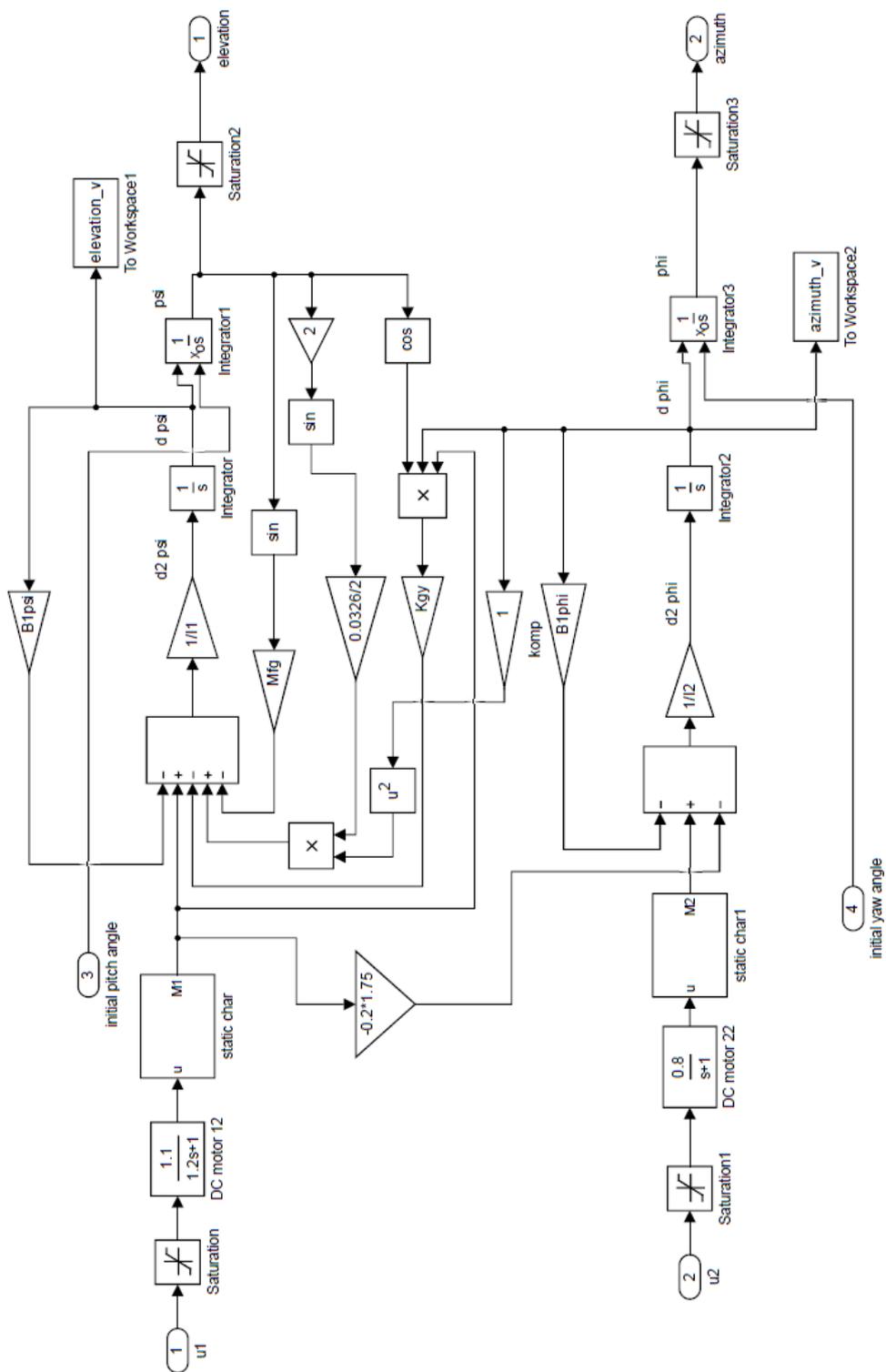
% PRBS sequence generation
i=Tappli+1;
while (i<=Nsamp)
    uiu = -sbpa(k1)*sbpa(k2);
    if (ValLgReg == 8)
        uiu = -sbpa(2)*sbpa(3)*sbpa(5)*sbpa(8);
    end
    j=1;
    while (j<=ValDivi)
        prbs(i) = uiu * ValAmpli + ValDecal;
        i=i+1;
        j=j+1;
    end
    for j=ValLgReg:-1:2
        sbpa(j) = sbpa(j-1);
    end
    sbpa(1) = uiu;
end
end

```

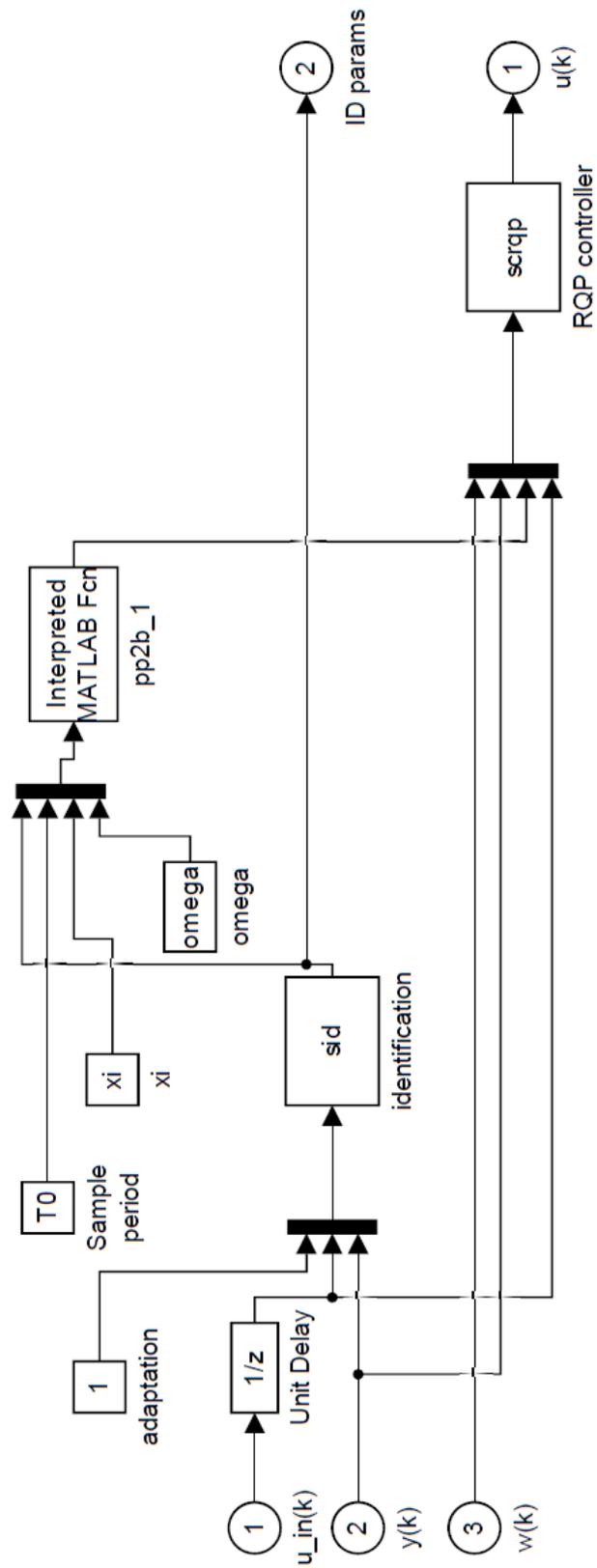
Anexo 2 Función create_prbs.m.



Anexo 3 Moledo lineal del sistema TRMS



Anexo 4 Modelo no lineal del sistema TRMS



Anexo 5 Bloque pp2b_1.mdl del controlador adaptativo.

```

function
[sys,x0,str,ts]=sid(t,x,u,flag,T0,num_order,denum_order,dead_time
,Theta0,ID_type,C0,phi0,lambda0,nu0,rho)
% Discrete model identification function.
% The transfer function of the model is of following type:
%
%
%          Y(z^-1)          b1*z^-1 + ... + bm*z^-m
% Gs(z^-1) = ----- = ----- * z^-d
%          U(z^-1)          1 + p1*z^-1 + ... + pn*z^-n
%
%
% where: m ... order of numerator
%        n ... order of denominator
%        d ... dead time (in sample periods) d>=0
%
% Function identifies system using recursive least square method
with exponential or
% adaptive directional forgetting.
%
%
[sys,x0,str,ts]=sid(t,x,u,flag,T0,num_order,denum_order,dead_time
,Theta0,ID_type,C0,phi0,lambda0,nu0,rho)
% States:
%   x(1:n+m) ..... n+m values [a1,
a2, ... an, b1, b2, ... bm]
%
%                                     parameter estimation
%   x(m+n+1:n+m+n) ..... n values [-y(k-1), -
y(k-2), ... -y(k-n)]
%
%                                     previous system
outputs
%   x(2*n+m+1:2*n+m+m+d-1) ..... m+d-1 values [u(k-2),
u(k-3), ... u(k-m-d)]
%   x(2*(n+m)+d-1, 2*(n+m)+d-1+(m+n)^2) ... (m+n)^2 values -
identification matrix C
%   x(2*(n+m)+d-1+(m+n)^2+1) ..... identificatin
variable lambda
%   x(2*(n+m)+d-1+(m+n)^2+2) ..... identificatin
variable nu
%   x(2*(n+m)+d-1+(m+n)^2+3) ..... identificatin
variable phi
% Inputs:
%   u(1) ... controls the identification process
%           < 1 ..... don't perform identification (just remember
inputs and outputs)
%           >= 1 ... perform identification
%   u(2) ... u(k-1) input to the identified process in previous
step
%   u(3) ... y(k) current output of the identified process
%   T0 ..... sample time
%   num_order ..... (m) order of the polynomial in the numerator
of transfer function
%   denum_order ... (n) order of the polynomial in the
denominator of transfer function

```

Anexo 6 Función sid.m.

```

%   dead_time ..... (d) dead time of the process (in sample times
T0)
%   Theta0 ..... initial parameter estimations [a1, a2, ...
an, b1, b2, ... bm]
%   ID_type ..... type of recursive identification
%               1 ... pure least squares method (LSM)
%               2 ... LSM with exponential forgetting
%               3 ... LSM with adaptive directional
forgetting
%   C0 ..... initial value for identification matrix C
(C0=1e3*eye(m+n))
%   phi0 ..... initial value for identification variable phi
(phi0=1)
%   lambda0 ..... initial value for identification variable
lambda (lambda=1e-3)
%   nu0 ..... initial value for identification variable nu
(nu0=1e-6)
%   rho ..... identification constant rho (rho=0.99)
% Outputs:
%   sys = [a1, b1, a2, b2, a3, b3, ...]

%234567890 234567890 234567890 234567890 234567890 234567890
234567890 234567890 234567890***** (max 90 cahrs)**

m = num_order;
n = denum_order;
d = dead_time;

if flag == 0    %initialization
    x0(1:n+m) = Theta0;                %[a1, a2, ...
an, b1, b2, ... bm]
    x0(m+n+1:n+m+n) = zeros(1,n);     %[-y(k-1), -
y(k-2), ... -y(k-n)]
    x0(2*n+m+1:2*n+m+m+d-1) = zeros(1,m+d-1);    %[u(k-2),
u(k-3), ... u(k-m-d)]
    x0(2*(n+m)+d:2*(n+m)+d-1+(m+n)^2) = C0(:);
    x0(2*(n+m)+d-1+(m+n)^2+1) = lambda0;
    x0(2*(n+m)+d-1+(m+n)^2+2) = nu0;
    x0(2*(n+m)+d-1+(m+n)^2+3) = phi0;
    %information about this function
    sys(1) = 0;                        %Number of continuous states.
    sys(2) = 2*(n+m)+d-1+(m+n)^2+3;    %Number of discrete states.
    sys(3) = m+n;                      %Number of outputs.
    sys(4) = 3;                        %Number of inputs.
    sys(5) = 0;                        %Reserved for root finding. Must be zero.
    sys(6) = 1;                        %Direct feedthrough flag (1=yes, 0=no)
    sys(7) = 1;                        %Number of sample times. This is the number of
rows in TS.
    ts = [T0 0];
    str = [];

```

Anexo 6 Función sid.m.

```

elseif (flag == 2 | flag==3)
    %ID types:
    LSM = 1;           %pure least squares method
    LSM_ef = 2;       %LSM with exponential forgetting
    LSM_adf = 3;      %LSM with adaptive directional forgetting

    uk1 = u(2);           %u(k-1)
    yk = u(3);           %y(k)
    Theta = x(1:n+m);     %[a1, a2, ... an, b1, b2, ... bm]
    %[-y(k-1), -y(k-2), ... -y(k-n), u(k-1), u(k-2), u(k-3), ...
u(k-m-d)]
    yu = [x(m+n+1:n+m+n); uk1; x(2*n+m+1:2*n+m+m+d-1)];
    %[-y(k-1), -y(k-2), ... -y(k-n), u(k-d-1) u(k-d-2), ... u(k-
m-d)]
    PHI = [yu(1:n); yu(n+1+d:n+d+m)];
    C = zeros(m+n);
    C(:) = x(2*(n+m)+d:2*(n+m)+d-1+(m+n)^2);
    lambda = x(2*(n+m)+d-1+(m+n)^2+1);
    nu = x(2*(n+m)+d-1+(m+n)^2+2);
    phi = x(2*(n+m)+d-1+(m+n)^2+3);

    if (u(1) >= 1) %perform identification
        xi = PHI' * C * PHI;
        e = yk - Theta' * PHI;
        if (ID_type == LSM) | (ID_type == LSM_adf) % | (ID_type ==
LSM_ef)
            Theta = Theta + C*PHI/(1+xi) * e;
        elseif (ID_type == LSM_ef)
            Theta = Theta + C*PHI/(phi+xi) * e;
        end;
    end;

    if (flag==2) %update states
        if (u(1) >= 1) %perform identification
            if (xi > 0)
                if (ID_type == LSM)
                    C = C - C*PHI*PHI'*C/(1+xi);
                elseif (ID_type == LSM_ef)
                    C = (C - C*PHI*PHI'*C/(phi+xi))/phi;
                elseif (ID_type == LSM_adf) %| (ID_type == LSM_ef)
                    %Kulhavy, dizertace, vztah 5.56
                    eta = e*e/lambda;

                    %Kulhavy, dizertace, vztah 5.92
                    phi =
1/(1+(1+rho)*(log(1+xi)+((nu+1)*eta/(1+xi+eta)-1)*xi/(1+xi)));
                    %Kulhavy, dizertace, vztah 5.96
                    %phi =
1/(1+(1+rho)*(nu+1)*eta/(1+xi+eta)*xi/(1+xi));
                    %drive pouzivane (asi spatne):
                    %phi =
1/(1+(1+rho)*(log(1+xi)+((nu+1)*eta/(1+xi+eta)-1)*xi/(1+xi)));

```

Anexo 6 Función sid.m.

```

        %Kulhavy, dizertace, vztah 5.49
        epsilon = phi-(1-phi)/xi;

        %Kulhavy, dizertace, vztah 5.48
        C = C - C*PHI*PHI'*C/(inv(epsilon)+xi);

        %Kulhavy, dizertace, vztah 5.52
        lambda = phi*(lambda + e*e/(1+xi));

        %Kulhavy, dizertace, vztah 5.53
        nu = phi*(nu+1);
    end
end
end

    %values for next loop
    x(1:n+m) = Theta;
a2, ... an, b1, b2, ... bm]
    x(m+n+1:n+m+n) = [-yk; PHI(1:n-1)];
-y(k-2), ... -y(k-n)]
    x(2*n+m+1:2*n+m+m+d-1) = yu(n+1:n+d+m-1);
2), ... u(k-m-d)]
    x(2*(n+m)+d:2*(n+m)+d-1+(m+n)^2) = C(:);
    x(2*(n+m)+d-1+(m+n)^2+1) = lambda;
    x(2*(n+m)+d-1+(m+n)^2+2) = nu;
    x(2*(n+m)+d-1+(m+n)^2+3) = phi;

    sys = x;

else    %flag==3 compute output
    %Theta = [a1,a2,...an, b1,b2,...bm]
    %sys = [a1, b1, a2, b2, a3, b3, ...]

    sys = zeros(1,m+n);
    sys(1:2:2*n-1) = Theta(1:n);
    sys(2:2:2*m) = Theta(n+1:n+m);

end

else
    sys=[];
end

```

Anexo 6 Función sid.m.


```

% solve diofant equation A(z^-1)*P(z^-1) + B(z^-1)*(Q(z^-1)+r0)
= D(z^-1)
% P(z^-1) = (1-z^-1) * (1+gama*z^-1)
% Q(z^-1) = (1-z^-1) * (q0m - q2m*z^-1)
% D(z^-1) = 1 + d1*z^-1 + d2*z^-2
s1=a2*(b2*(a1*(b1+b2)+b1*(d2-a2)-b2*(d1+1))-a2*b1^2);
r1=(b1+b2)*(a1*b1*b2-a2*b1^2-b2^2);

q2m = s1/r1;
gama = q2m*b2/a2;
q0m = q2m*(b1/b2-a1/a2)-a2/b2;

r0 = (d1+1-a1-gama-b1*q0m)/b1;
q0 = q0m + r0;
q1 = -(q0m + q2m);
q2 = q2m;
p1 = gama-1;
p2 = -gama;

param=[r0; q0; q1; q2; p1; p2];

```

Anexo 7 Función pp2b_1.m.


```

q2 = u(6);
p1 = u(7);
p2 = u(8);
uk1 = u(9); %u(k-1)

u = r0*wk - q0*yk - q1*yk1 - q2*yk2 - p1*uk1 - p2*uk2;

if (flag==2) %update states
    sys = [yk yk1 uk1]; %[y(k-1) y(k-2) u(k-2)]

else %flag == 3 compute output
    sys = u;

end

else

    sys=[];
end

```

Anexo 8 Función scrqp.m.

```

function [omega_0,dmp]=omega_dmp(tm_req,M_req);
%function [omega_0,dmp]=omega_dmp(tm_req,M_req);
%computes natural frequency omega_0 and damping dmp of a
continues
%2nd order system with required properties: maximum overshoot
M_req
%and rise time tm_req. The continues system has a standard form:
%
%      omega_0
%  G2(s)= -----
%      s^2 + 2.dmp.omega_0.s + omega_0^2

%inputs:
%tm_req ... required rise time in seconds
%M_req ... required maximum overshoot
%outputs:
%omega_0 ... natural frequency of the continues system
%dmp ... damping of the continues system
%
%written by: H. Prochazka, I.D. Landau
%7th june 2002

%damping computing
precision=0.1;%precision of 0.1%
omega_0=6;%initial value 6 rad/s to compute damping for overshoot
dmp_min=0;
dmp_max=1;
error=precision*10;%initial setting
while error>precision,
    dmp_act=dmp_min+ (dmp_max-dmp_min)/2;
    sys=tf([omega_0^2],[1 2*dmp_act*omega_0 omega_0^2]);
    [resp,tim]=step(sys);
    M_act=(max(resp)-1)*100;
    if M_act<M_req,
        dmp_max=dmp_act;
    else
        dmp_min=dmp_act;
    end;
    error=abs(M_act-M_req);
end;
dmp=dmp_act;

%natural frequency computing
k=1;while resp(k)<0.1, k=k+1;end;
t01=tim(k-1);
k=1;while resp(k)<0.9, k=k+1;end;
t09=tim(k);
tm_act=t09-t01;
omega_0=tm_act*omega_0/tm_req;

```

Anexo 9 Función omega_dmp.m