

Universidad de Alcalá
Escuela Politécnica Superior

GRADO EN INGENIERÍA ELECTRÓNICA Y AUTOMÁTICA
INDUSTRIAL



Trabajo Fin de Grado

Diseño y aplicación de un controlador inteligente para el
ajuste de la posición y velocidad de un vehículo a escala



ESCUELA POLITECNICA
SUPERIOR

Autor: Héctor Leceta Cuevas

Tutor/es: Carlota Salinas Maldonado

2022



Escuela Politécnica Superior

**GRADO EN INGENIERÍA EN
ELECTRÓNICA Y AUTOMÁTICA
INDUSTRIAL**

Trabajo Fin de Grado

Diseño y aplicación de un controlador inteligente para el
ajuste de la posición y velocidad de un vehículo a escala

Autor: Héctor Leceta Cuevas

Tutor: Carlota Salinas Maldonado

TRIBUNAL:

Presidente: D. Javier Alonso Ruiz

Vocal 1º: D. Rubén Izquierdo Gonzalo

Vocal 2º: D. Carlota Salinas Maldonado

FECHA: Septiembre 2022

Índice general

Resumen en castellano	2
English summary.....	3
Índice de figuras	4
Índice de tablas.....	6
1. Introducción.....	7
2. Estado del arte	8
2.1. Estudio inicial del Motor DC.....	8
2.2. Pruebas iniciales con el Motor DC	11
2.3. Controlador PID clásico	15
2.4. Controlador PID Fuzzy	17
2.5. Análisis de los modelos de posición y velocidad del motor.....	24
2.5.1. Modelo en Espacio de Estados	24
2.5.2. Modelo Box-Jenkins	26
3. Estudio experimental de los modelos de posición y velocidad del motor	27
3.1. Modelo de posición	27
3.1.1. Modelo de posición Box-Jenkins	30
3.1.2. Modelo de posición en Espacio de Estados	45
3.2. Modelo de velocidad	57
3.2.1. Modelo de velocidad Box-Jenkins	60
3.2.2. Modelo de velocidad en Espacio de Estados	73
3.3. Conclusiones sobre los modelos a elegir	86
4. Pruebas reales en el Motor DC	86
4.1. Motor sin perturbaciones.....	86
4.1.1. Motor sin perturbaciones: Posición.....	86
4.1.2 Motor sin perturbaciones: Velocidad	96
4.2. Modelo con perturbaciones: Motor libre	106
4.2.1. Modelo con perturbaciones: Motor libre: Posición	106
4.2.2. Modelo con perturbaciones: Motor libre: Velocidad.....	111
5. Conclusiones	117
6. Bibliografía	119

Resumen en castellano

El estudio de los controladores PID de un motor DC pasa por comprender, en primera instancia, qué es un motor DC y cuáles son sus limitaciones. También será necesario comprender que el motor DC con el que trabajemos va a tener unas prestaciones reducidas, así que el margen de mejora no es demasiado amplio, y además de todo ello también habremos de interiorizar que, si bien se estudiará el dispositivo y su comportamiento para alcanzar consignas de posición y velocidad está diseñado en última instancia para las segundas, por lo que exigiremos alta precisión en su seguimiento de este tipo de referencias. Posteriormente se expondrán los distintos tipos de Controladores PID empleados en el análisis, siendo estos el Clásico y el Fuzzy (o difuso), y con ello podemos establecer una comparativa para juzgar la conveniencia de uno por encima del otro. Para no sobrecargar al motor partiremos del estudio matemático y simulado de la planta o sistema (nuestro motor) y diseñaremos los controladores a partir de este, una vez hecho esto los aplicaremos en el motor en dos situaciones diferentes: libre sin carga y libre con carga, para poder estudiar los parámetros en régimen transitorio y permanente del mismo, así como la eficacia de sendos controladores. A lo largo de este documento se pondrá a prueba la respuesta de un motor de corriente continua de calidad reducida y se observará cómo su comportamiento puede verse altamente mejorado mediante el empleo de controladores, cerciorándonos además de que la versatilidad y adaptabilidad de los controladores borrosos hacen que sean considerados una alternativa más plausible que los clásicos.

Palabras clave: Controlador PID, Motor DC, Vehículos Autónomos, Lógica borrosa, Modelo, Identificación de Sistemas, Control de Sistemas, Controladores Inteligentes

English summary

The study of PID controllers applied to a DC motor goes through the understanding, in the first place, of what is a DC motor and which are its limitations. It will also be necessary comprehending that the DC motor with which we will be working has a reduced quality, and thus the improvement gap is not too wide, besides we will also have to interiorize that, although the device will be studied for it to reach position and velocity references it is ultimately designed for the latter, so we will impose high precision to it in the act of following these kind of signals. After this we will present the various types of PID Controllers used in the analysis, these being the Classic and the Fuzzy ones, and along with that we will be capable of stablishing a comparison between the two to judge the convenience of one kind over the other. In the spirit of avoiding overloads on the DC motor we will depart from the mathematic and simulated study of the plant or system (our motor) and we will design the controllers based on this, once it is done we will apply them to the real device in two different settings: free without load and free with load, to be capable to understand the parameters of the system in transitory and steady states, along with the efficiency of said controllers. Throughout this End of Degree Assignment, to summarize, the response of a low quality DC motor will be put at test, and it will be shown how its behavior can be highly improved through the utilization of controllers, also proving that the versatility and adaptability of the Fuzzy controllers makes them a better alternative rather than the classic ones.

Keywords: PID Controller, DC Motor, Autonomous Vehicles, Fuzzy logic, Model, System Identification, System Control, Intelligent Controllers

Índice de figuras

Figura 1: Dibujo esquemático de un motor con una sola espira	8
Figura 2: Rotor de motor de corriente continua	10
Figura 3: Sensor de efecto Hall en cuadratura	10
Figura 4: Representación de las entradas del puente en H.....	12
Figura 5: Esquema del conexionado	13
Figura 6: Respuesta del sistema ante señales de velocidad tipo rampa.....	15
Figura 7: Respuesta del sistema ante señales de velocidad tipo escalón	15
Figura 8: Esquema general de un controlador PID Clásico	16
Figura 9: Representación de la agregación de premisas en PID Fuzzy.....	19
Figura 10: Representación del criterio máximo en PID FUzzy	20
Figura 11: Representación del criterio COG en PID FUzzy	21
Figura 12: Representación del criterio de centros ponderados en PID FUzzy.....	21
Figura 13: Representación del funcionamiento de un sistema tipo Takagi-Sugeno.....	22
Figura 14: Ejemplo de desborrosificación	23
Figura 15: Representación del funcionamiento de un sistema tipo Mamdani.....	24
Figura 16: Representación del diagrama de bloques de Espacio de Estados.....	25
Figura 17: Diagrama de bloques del modelo de posición en Simulink.....	1
Figura 18: Datos de posición de entrenamiento.....	29
Figura 19: Datos de posición de validación	30
Figura 20: comportamiento del modelo de posición Box-Jenkins con el real.....	30
Figura 21: representación matemática del modelo Box-Jenkins obtenido.....	31
Figura 22: Modelo Box-Jenkins sin control PID.....	31
Figura 23 a), b) c): Salida de Box-Jenkins ante distintas entradas (posición)	32
Figura 24: Modelo de posición Box-Jenkins con Control PID.....	34
Figura 25 a), b), c): Salida de Box-Jenkins con PID Clásico (posición).....	35
Figura 26: Modelo de posición Box-Jenkins con Controlador PID Fuzzy	37
Figura 27 a), b), c), d), e), f): Controlador PID Fuzzy de posición Box-Jenkins	38
Figura 28 a), b) c): Salida de Box-Jenkins ante distintas entradas con PID Fuzzy (posición).....	41
Figura 29: entrada de posición alimentada al motor real y salida del mismo	43
Figura 30 a), b), c): Box-Jenkins sin controlador, con PID Clásico y PID Fuzzy.....	44
Figura 31: comportamiento del modelo de posición en Espacio de Estados con el real	45
Figura 32: representación matemática del modelo de posición en Espacio de Estados	46
Figura 33 a), b) c): Salida en Espacio de Estados ante distintas entradas (posición)....	47
Figura 34 a), b), c): Salida en Espacio de Estados con PID Clásico (posición).....	49
Figura 35 a), b), c), d), e), f): Controlador PID Fuzzy de posición Espacio de Estados ..	51
Figura 36 a), b), c): Salida en Espacio de Estados con PID Fuzzy (posición)	54
Figura 37 a), b), c): Espacio de Estados sin controlador, con PID Clásico y PID Fuzzy (posición).....	56
Figura 38: estructura del modelo de velocidad montado en Simulink	1
Figura 39: Datos de entrenamiento para el modelo de velocidad.....	59
Figura 40: Datos de validación para el modelo de velocidad.....	60
Figura 41: modelo de velocidad Box-Jenkins comparado con el real	61
Figura 42: representación matemática del modelo Box-Jenkins de velocidad	61
Figura 43: estructura del modelo Box-Jenkins de velocidad en Simulink.....	62

Figura 44 a), b) c): Salida de Box-Jenkins ante distintas entradas (velocidad).....	62
Figura 45 a), b), c): Salida de Box-Jenkins con PID Clásico (velocidad)	64
Figura 46 a), b), c), d), e), f): Controlador PID Fuzzy de velocidad Box-Jenkins.....	66
Figura 47: esquema del modelo Box-Jenkins de velocidad con PID Fuzzy	67
Figura 48 a), b), c): Salida de Box-Jenkins con PID Fuzzy (velocidad)	67
Figura 49 a), b), c): Box-Jenkins sin controlador, con PID Clásico y PID Fuzzy (velocidad)	70
Figura 50: salida de Box-Jenkins comparada con el motor real ambos con PID Fuzzy .72	72
Figura 51: modelo de velocidad en Espacio de Estados comparado con el real.....	73
Figura 52: representación matemática del modelo de velocidad en Espacio de Estados	74
Figura 53 a), b) c): Salida en Espacio de Estados ante distintas entradas (velocidad) ...	75
Figura 54: estructura del modelo de velocidad en Espacio de Estados con PID Clásico	77
Figura 55 a), b), c): Salida en Espacio de Estados con PID Clásico (velocidad)	77
Figura 56: estructura del modelo de velocidad en Espacio de Estados con PID Fuzzy ..	79
Figura 57 a), b), c), d), e), f): PID Fuzzy de velocidad en Espacio de Estados	80
Figura 58 a), b), c): Salida en Espacio de Estados con PID Fuzzy (velocidad)	83
Figura 59 a), b), c): Espacio de Estados sin controlador, con PID Clásico y PID Fuzzy (velocidad).....	85
Figura 60: estructura en Simulink para controlar el motor real en posición con un PID Clásico.....	1
Figura 61: estructura en Simulink para controlar el motor real en posición con un PID Fuzzy.....	1
Figura 62 a), b) ,c), d): PID's Clásicos y Fuzzy, obtenidos con Box-jenkins y Espacio de Estados, aplicados al motor real en posición	89
Figura 63: Motor real con PID Clásico ajustado comparado con PID Clásico de Box- Jenkins en posición	91
Figura 64 a), b) ,c), d), e), f): PID Fuzzy de posición ajustado.....	92
Figura 65: Motor real con PID Fuzzy ajustado comparado con PID Fuzzy de Box-Jenkins en posición.....	94
Figura 66 Respuesta real en posición sin controlador, con PID Clásico y PID Fuzzy ...	94
Figura 67: estructura en Simulink para controlar el motor real en velocidad con un PID Clásico.....	1
Figura 68: estructura en Simulink para controlar el motor real en velocidad con un PID Fuzzy.....	1
Figura 69 a), b) ,c), d): PID's Clásicos y Fuzzy, obtenidos con Box-jenkins y Espacio de Estados, aplicados al motor real en velocidad	99
Figura 70: Motor real con PID Clásico ajustado comparado con PID Clásico de Espacio de Estados en velocidad	101
Figura 71 a), b) ,c), d), e), f): PID Fuzzy de velocidad ajustado.....	102
Figura 72: Respuesta real con PID Fuzzy en posición sin controlador	103
Figura 73 Respuesta del motor en posición sin controlador, con PID Clásico y Fuzzy	104
Figura 74 a), b), c), d), e): motor sin controlador comparado con respuesta en velocidad con PID Clásico y PID Fuzzy, antes y después del ajuste	106
Figura 75 a), b) ,c), d), e), f): PID Fuzzy de posición ajustado ante perturbaciones	109
Figura 76 a), b), c), d), e): motor sin controlador comparado con respuesta en velocidad con PID Clásico y PID Fuzzy, antes y después del ajuste	111
Figura 77 a), b) ,c), d), e), f): PID Fuzzy de velocidad ajustado ante perturbaciones ...	114

Índice de tablas

Tabla 1: Parámetros del sistema Box-Jenkins ante entradas tipo escalón con un controlador PID clásico, en régimen permanente y transitorio	36
Tabla 2: Parámetros del sistema Box-Jenkins ante entradas tipo escalón con un controlador PID Fuzzy, en régimen permanente y transitorio.....	42
Tabla 3: Parámetros en régimen transitorio y permanente de la salida del modelo en Espacio de Estados regulado por un Controlador PID Clásico	50
Tabla 4: Parámetros en régimen permanente y transitorio del sistema en Espacio de Estados con Controlador PID Fuzzy ante distintas estradas.....	55
Tabla 5: Parámetros en régimen transitorio y permanente del modelo Box-Jenkins de velocidad sin regulación	64
Tabla 6: Parámetros en régimen transitorio y permanente del modelo en Espacio de Estados sin controlador.....	76
Tabla 7: Parámetros en régimen permanente y transitorio del modelo de velocidad en Espacio de Estados con Control PID Clásico	79
Tabla 8: Parámetros en régimen permanente y transitorio del modelo de velocidad en Espacio de Estados regulado por un Controlador PID Fuzzy	84
Tabla 9: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy.....	96
Tabla 10: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy.....	105
Tabla 11: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy del sistema con perturbaciones.....	111
Tabla 12: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy del sistema con perturbaciones.....	116

1. Introducción

La inteligencia artificial se halla en la actualidad prácticamente en todas las facetas de la existencia de las personas en países con un desarrollo industrial elevado: ya sea con objetivos bélicos, automovilísticos, o de entretenimiento, el empleo de redes neuronales y estructuras borrosas abre nuevos campos de posibilidades en el reconocimiento de objetos o personas, así como en el control de sistemas. Es precisamente hacia un objetivo automovilístico al que hemos encauzado este Trabajo de Fin de Grado, así como las pruebas y experimentos registrados en este documento, puede llegar a emplearse en la regulación de los motores de los vehículos de un futuro que ya es palpable. Así pues, planteamos como meta de este documento el ser partícipes en el desarrollo de la automoción inteligente del futuro, aunque sea a pequeña escala, aportando ideas que pudiesen servir de inspiración a aquellos expertos que se erigen como pioneros de los vehículos autónomos.

Los sistemas inteligentes comprenden un abanico de posibilidades y aplicaciones tan amplio que en la actualidad estos mismos sistemas están en fase de experimentación enfocados a instaurarse como un componente fundamental de los vehículos autónomos, buscándose la identificación y reconocimiento de obstáculos, peatones e incluso otros automóviles (1). El Trabajo de Fin de Grado aquí presente se centrará en la capacidad de control de sistemas reales, actuando sobre un motor de un vehículo a escala para demostrar como las redes neuronales permiten una regulación fina de esta clase de dispositivos. Aunque el material empleado en las pruebas y experimentos es de bajas prestaciones la mejora obtenida en el mismo al ejercer una acción de control sobre él es cuanto menos notable. El motor a escala, por su simplicidad de manejo y gobierno, resulta ideal para realizar las pruebas aquí expuestas.

En este documento se aglutinan las pruebas y resultados obtenidos a partir de la experimentación sobre nuestro pequeño motor eléctrico siguiendo distintas pautas: en primera instancia se modeló el motor a través de dos técnicas que se enfrentaron para poder ver qué representación se aproximaba de forma más fidedigna a la realidad (siendo estos los modelos en Espacio de Estados y Box-Jenkins), emulando el comportamiento tanto en posición como en velocidad de nuestro dispositivo, ya que, aunque nuestro enfoque va directamente ligado a ser capaces de gobernar la velocidad angular, la posición puede utilizarse para expresar la precisión real del motor para alcanzar una consigna. Una vez realizadas las pruebas en simulación con los modelos de Espacio de Estados se prosiguió con la aplicación de sendos controladores PID Clásicos y PID Inteligentes al motor real, diseñados a partir de aquel modelo que aportase los resultados que mejor se adecúen a la realidad. En primera instancia sin perturbaciones (para poder trabajar en las condiciones más cercanas a la idealidad que sean posibles) y a continuación funcionando libre con perturbaciones (para poder apreciar cómo la inclusión de perturbaciones exige un ajuste de los controladores ya diseñados). Con las modificaciones necesarias se irán ajustando los controladores previamente mencionados y se contrastarán los comportamientos arrojados por el motor, para poder discernir qué acercamiento a la teoría de control, si la clásica o la inteligente, es la más adecuada.

2. Estado del arte

2.1. Estudio inicial del Motor DC

La implementación de controladores sobre el motor de corriente continua pasa por comprender cómo se comporta este último tanto teórica como en la práctica, observando cuales son los principios que lo gobiernan y sus límites de funcionamiento reales. El motor del que disponemos es un Motor DC sencillo de escobillas, con inductor de imanes permanentes en el estator e inducido en el rotor compuesto por espiras enrolladas alrededor de una pieza cilíndrica metálica con incisiones en el extrarradio y un eje que lo atraviesa por el centro.

El funcionamiento del motor es sencillo: consta de un inductor y un inducido, siendo el inductor unos imanes permanentes en la superficie lateral del cilindro que compone el armazón del motor, estos dos imanes constituyen un par polo norte/polo sur que generan un campo magnético dirigido (no disperso) entre ambos. El inducido se halla entre estos dos imanes, y está compuesto por bobinas enrolladas en las hendiduras de una pieza metálica cilíndrica con un eje solidario a esta y que es la que da vueltas al atravesarle la corriente eléctrica y generar una fuerza magnética que produce un par rotativo en el eje que la atraviesa. Las escobillas permiten la circulación de corriente a cada una de las bobinas conforme el inductor gira, entrando en contacto con las mismas en el colector de delgas (2), tal y como ilustra la *Figura 1*:

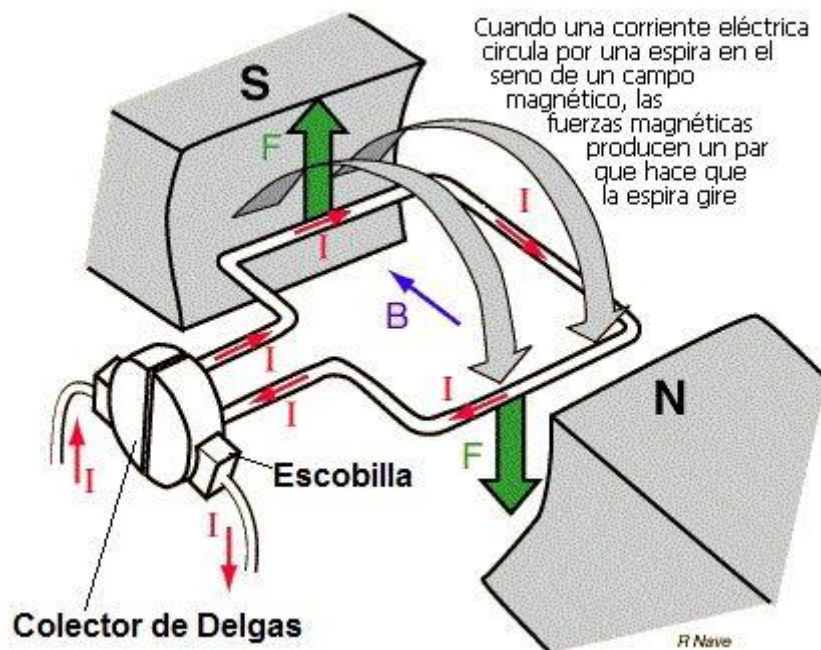


Figura 1: Dibujo esquemático de un motor con una sola espira (3)

Donde:

- \vec{B} es el vector campo magnético
- \vec{F} es el vector fuerza magnética
- \vec{L} es el vector que indica la dirección y sentido de la corriente
- I es la corriente que atraviesa la bobina
- θ es el ángulo formado entre \vec{L} y \vec{B}
- L es la longitud del conductor

De manera que la fuerza magnética responde a la siguiente expresión:

$$\vec{F} = I \cdot \vec{L} \times \vec{B} \quad (1.1)$$

Donde \times representa el producto escalar entre \vec{L} y \vec{B} . El módulo de esta fuerza se representa como:

$$F = B \cdot L \cdot I \cdot \text{sen}(\theta) \quad (1.2)$$

Y sabiendo que el vector par se consigue como:

$$\vec{M} = \vec{r} \times \vec{F} \quad (1.3)$$

Donde \vec{r} es el vector radio (es decir, aquel cuya dirección es perpendicular al eje del motor, su sentido es hacia el exterior, y su módulo es la distancia desde el eje hasta el punto de aplicación de la fuerza), empleando el resultado de la ecuación (1.3), podemos averiguar la velocidad angular con facilidad:

$$P = M \cdot \omega; \quad \omega = \frac{P}{M} \quad (1.4)$$

Donde P es la potencia del motor y M es el módulo del par de rotación. Puesto que la potencia nominal del motor es de 2W (Vatios) y la máxima es de 3,1W (**4**) podemos hallar con facilidad esta magnitud.

Aunque la ilustración solo muestra el ejemplo con una espira en el caso del motor de corriente continua lo único que hemos de considerar es que el proceso se repite con bobinas situadas una al lado de la otra, al portar el rotor un colector de galgas que permite a las escobillas transmitir la corriente a cada una de ellas (*Figura 2*):



Figura 2: Rotor de motor de corriente continua

Por su parte la medición de la posición y velocidad angulares se logra a través de sensores de efecto Hall en cuadratura, separados 90° entre sí, de 341,2 PPR (Pulsos Por Revolución) (4). Estos dos sensores constituyen un encoder que funciona mediante la generación de pulsos cuadrados en cada uno de ellos, dichos pulsos son resultado de la permisión (“1” lógico) o bloqueo (“0” lógico) de un campo magnético orientado hacia cada sensor debido al giro de una rueda con ranuras solidaria al eje del motor (Figura 3):

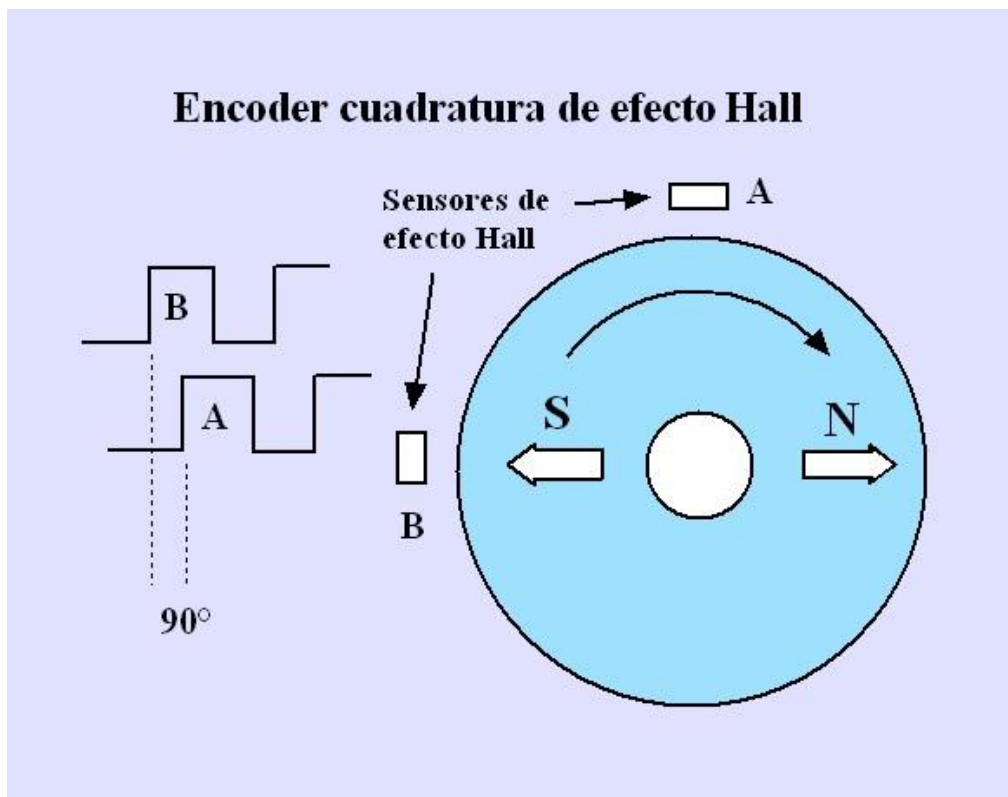


Figura 3: Sensor de efecto Hall en cuadratura (4)

El desfase entre ambos sensores nos permite determinar sentido de giro del motor, tal y como muestra la Figura 3.

2.2. Pruebas iniciales con el Motor DC

En el desempeño de la labor de controlar el motor de corriente continua, que es la piedra angular en que se sustenta este Trabajo de Fin de Grado, primero es necesario entender este dispositivo, y para ello requerimos hacer pruebas físicas con el mismo, de manera que su funcionamiento sea comprendido y sus límites de trabajo establecidos. La primera misión que se desempeñó con el motor de corriente continua era intentar entender su funcionamiento y los rangos de valores con que trabajaba. Para ello se utilizará una infraestructura que permita ejercer una acción de control a partir de una placa Arduino UNO (5) y un puente en H. Para la parte de recogida de datos y análisis del comportamiento del motor, se utilizará Matlab y Simulink (6). En la placa se cargaron programas que gobernaron el puente en H directamente, estableciendo el sentido de giro del motor y su velocidad a través de la modulación de una señal PWM en la entrada de habilitación del puente. En una primera prueba se intentó hallar con qué valores de dicha PWM el motor era capaz de iniciar el movimiento, puesto que se postula que debido al rozamiento estático este no era capaz de iniciar el movimiento hasta quedar por encima de un cierto valor en el ciclo de trabajo de dicha señal.

Resulta conveniente hablar también de cómo se puede configurar la señal PWM (7) generada por la placa Arduino UNO: esta cuenta con dos tipos de salidas, siendo digitales y analógicas. Las analógicas pueden proporcionar cualquier valor entre 0V y 5,5V con una resolución de 255 bits, mientras que las digitales pueden ofrecer solo 0V ó 5,5V, sin gradación alguna entre estos valores extremos. No obstante, se puede realizar dos tipos de escritura en estos pines digitales, que son, valga la redundancia, la digital y la analógica, y mientras que la digital solo permite que dichos pines presenten los valores extremos de tensión (haciendo la escritura de “0” ó “1” lógicos en los mismos, es decir, 0V ó 5,5V respectivamente) la escritura analógica permite cualquier valor comprendido entre 0 y 255, que en los pines que presentan el símbolo “~” dibujado junto a ellos indica que dicho pin permite generar una señal PWM cuadrada con una resolución de 256 bits en su ciclo de trabajo, lo cual significa que un valor de “0” implica un ciclo de trabajo nulo (señal constante de 0V) y uno de “255” un ciclo de trabajo del 100% (señal constante de 5,5V), permitiendo una variación gradual del mismo entre estos extremos. Introduciendo esta señal en la entrada de habilitación del puente en H (que denominaremos “Enable”, “Habilitar” en inglés) se lograba que este dejara pasar la corriente a través de él durante más o menos tiempo, en aumento cuanto más cerca del 100% estuviera el ciclo de trabajo, que modificaba a su vez la velocidad del motor (mayor a más cercanía de un ciclo de trabajo del 100%, al permitir la llegada de corriente durante un mayor tiempo al rotor del mismo).

Existen por otro lado otras entradas en el puente en H que, por simplicidad, denominaremos “In1” e “In2”, que permitían conmutar el sentido de giro del motor, de forma que si en “In1” existía un nivel alto y en “In2” uno bajo el motor giraba en la dirección que denominaremos “hacia adelante”, y si se invierten los valores en ambos (“In1” en nivel bajo e “In2” en alto) iba “hacia atrás”. Si en los dos se presentaran el mismo nivel (alto o bajo indistintamente) el puente en H corta la corriente, deteniendo el motor. Dichas entradas controlan la conmutación de las salidas de solo uno de los

lados del puente en H. El motor por su parte fue alimentado por tres baterías en serie que en conjunto ofrecen unos 12V. El esquema de puente en H se expone en la *Figura 4*:

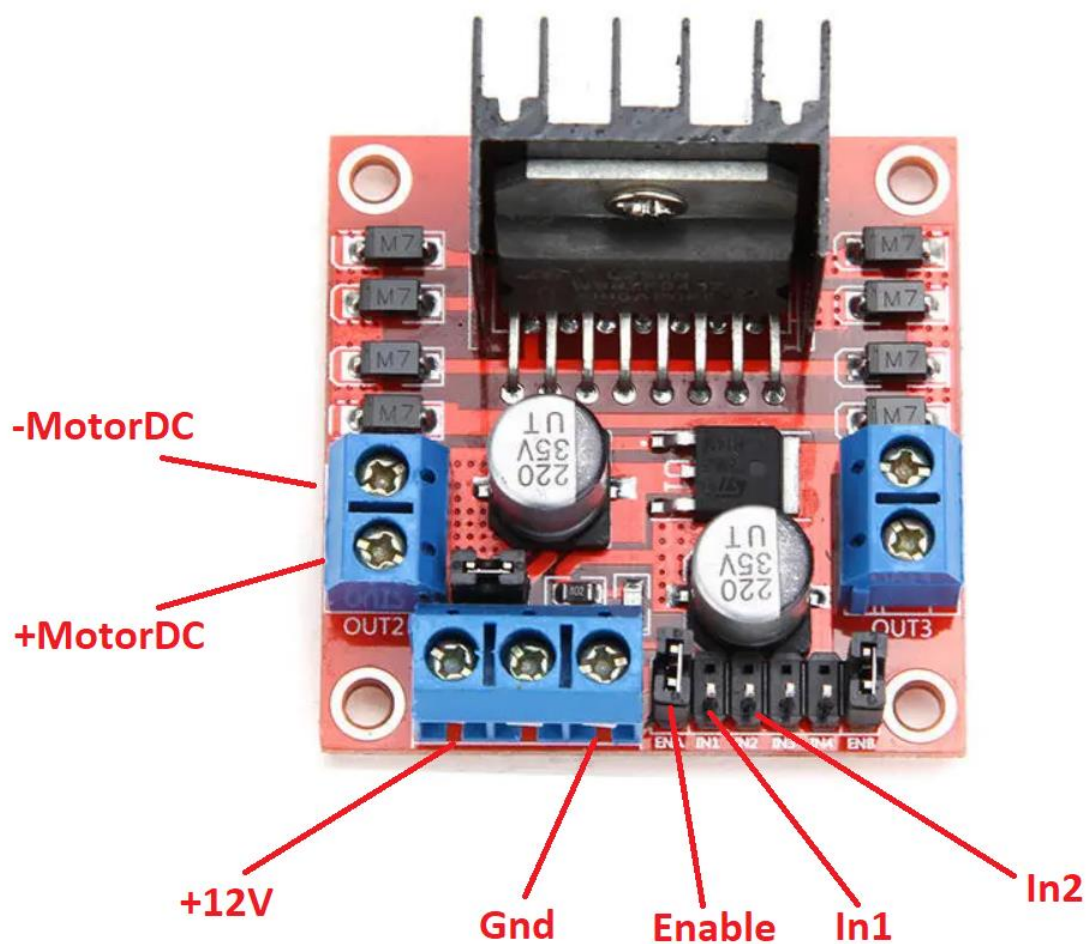


Figura 4: Representación de las entradas del puente en H

A continuación se muestra el esquema completo de conexiones entre la placa Arduino UNO, el puente en H, las baterías, el motor y el ordenador (*Figura 5*):

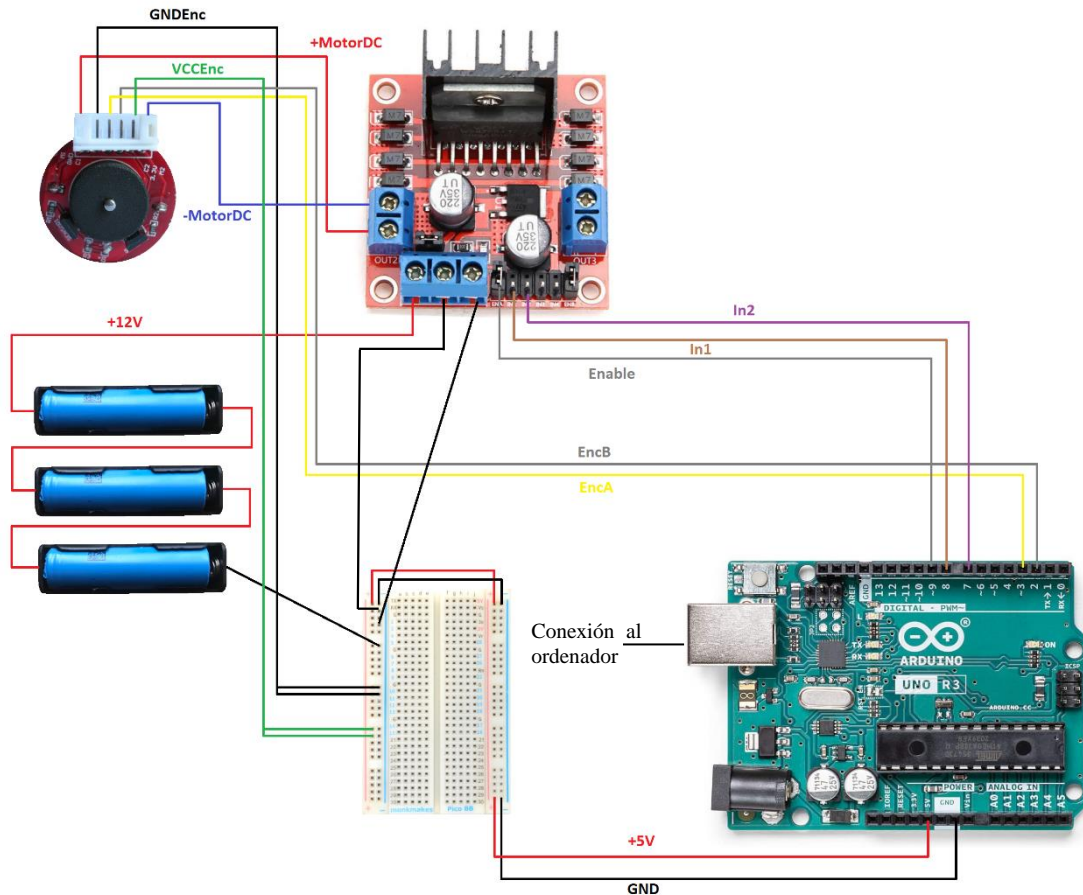


Figura 5: Esquema del conexionado entre la placa Arduino Uno, la protoboard, las baterías, el puente en H, el motor DC y el ordenador

Tras numerosas pruebas con el motor utilizando la infraestructura expuesta anteriormente, se empleó un programa cargado en la placa Arduino Uno para poder medir la velocidad real del motor a través de la lectura del encoder, dicho elemento constituyente del motor de corriente continua tiene una resolución de 748 pulsos por vuelta (4). Para ello se partió de un código que permitía convertir el número de cuentas del dispositivo a grados, y se hizo dar al motor tres vueltas (es decir, cuando la lectura del número de pulsos fuera de 3 multiplicado por 748, equivalente a 3 vueltas, se detuvo su giro) a máxima velocidad (5.5V en la entrada de “Enable” del puente en H). Para asegurar que el motor había alcanzado su pico de velocidad y evitar variaciones en la misma se contó el tiempo entre la primera (748 pulsos) y la segunda vuelta (2 por 748 pulsos), evitando así falseamiento de los datos por el arranque y aceleración, y su posterior frenado. Una vez ejecutado el código el programa mostró el siguiente resultado a través del monitor serie: el tiempo en milisegundos es 236 para realizar un giro completo, lo cual implica que el motor va más rápido de lo que indica su hoja de características, aplicando la siguiente ecuación para calcular las revoluciones por minuto (rpm) teóricas del motor:

$$\omega(rpm) = \frac{60}{t(s)} \quad (2.1)$$

Obtenemos una velocidad de unos 255rpm, muy por encima de los 210rpm otorgados por el fabricante **(8)**. A través de un conjunto de pruebas realizadas posteriormente, se pudieron acotar los márgenes de trabajo y manera de trabajar del motor. Se constató que si está quieto no empieza el movimiento hasta un valor de PWM de 60 (recordemos que este puede estar comprendido entre 0 y 255, lo cual implica que 60 es aproximadamente un ciclo de trabajo del 23,53%) al hacer frente al rozamiento estático, y una vez que empieza a moverse no se detiene hasta estar por debajo de 25 (ciclo de trabajo del 9,8% más o menos), no obstante este fenómeno solo se materializa en la realidad si se llega del valor primero al último de forma progresiva, puesto que si se realiza la transición bruscamente el motor se detiene completamente.

Una vez realizadas las pruebas con la Arduino UNO se procedió a configurar la conexión entre la placa y Simulink, para lo cual se instaló la Toolbox ArduinoIO **(9)**, que permite entablar una conexión con el puerto serie de la tarjeta, así como enviar o recibir señales en los distintos pines de entrada y salida. En primera instancia, y tal y como se relatará en apartados siguientes, se comenzó por introducir escalones aleatorios en el motor como consignas de velocidad y posición, pero posteriormente se experimentó con escalonamientos progresivos y rampas para la modificación de este primer parámetro, gracias a lo cual pudieron conocerse más cosas sobre él, y la fundamental fue que el cambio de la velocidad no es linealmente proporcional al aumento del ciclo de trabajo del PWM introducido en el pin “Enable” del puente en H que lo controla.

Las *Figuras 6 y 7*, muestran ejemplos de las mediciones realizadas. En ambas imágenes, la gráfica inferior representa el valor introducido en la salida de PWM de la placa Arduino Uno, que permite controlar el ciclo de trabajo del mismo (es decir, “255” es un ciclo de trabajo del 100% y velocidad máxima, mientras que “0” implica ciclo de trabajo nulo y detención del motor) y la gráfica superior la velocidad proporcionada por la lectura del “Encoder” al medir el ángulo en la muestra actual y la anterior y dividir por el tiempo de muestreo (definición de derivada discreta). Se aprecia por un lado que hay algún tipo de perturbación presente en el “Encoder” y que, además, la velocidad aumenta menos de lo que la señal de entrada exige para valores elevados de la misma. Demostrándose la inexistencia de una correlación lineal entre entrada y salida. Se ve que hasta un valor de 60 en la salida PWM de la placa nuestro motor no inicia su movimiento si se encontraba completamente detenido, pero que una vez que empieza a girar puede bajar por debajo de este valor, tal y como se expuso previamente. También es perceptible un cierto retardo en la salida respecto a la entrada.

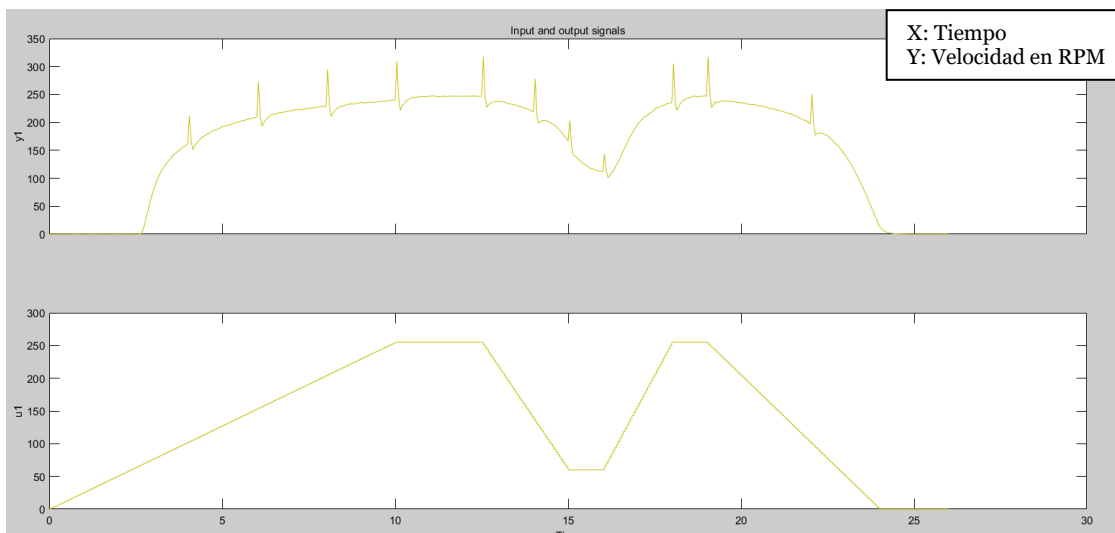


Figura 6: Respuesta del sistema (gráfica superior) ante señales de velocidad tipo rampa (gráfica inferior)

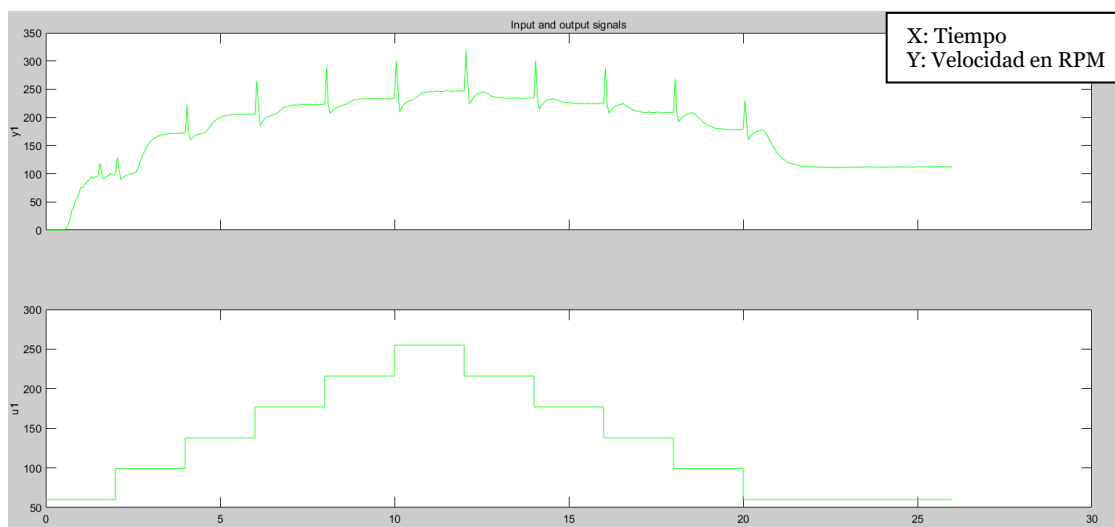


Figura 7: Respuesta del sistema (gráfica superior) ante señales de velocidad tipo escalón (gráfica inferior)

2.3. Controlador PID clásico

Los controladores PID deben sus siglas a las distintas acciones que lo integran: P por Proporcional, I por Integral, D por Derivativo. Se caracteriza este controlador por ser especialmente eficaz con sistemas LTI (Linear Time Invariant), lo que significa que son sistemas ajustados a un comportamiento definido por una función lineal cuyos coeficientes y parámetros no cambian con el tiempo ni con las condiciones iniciales, por lo que se han de diseñar en torno a lo que se conoce como posición de equilibrio, que es un entorno de valores alrededor del cual se cree que el sistema se comportará y dentro del cual la acción del controlador será efectiva **(10)**. Ello implica que, si el sistema se sale de esas especificaciones, la estructura de regulación nos resultará inútil si es clásico (no

es un dispositivo “inteligente”) al no poder actualizar los parámetros ante estas nuevas condiciones. El esquema general de un controlador PID es (Figura 8):

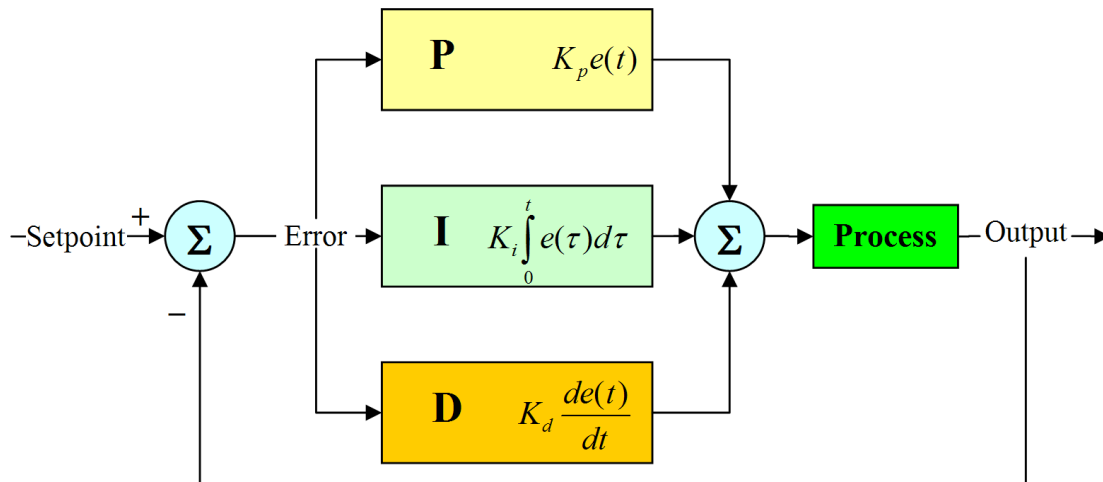


Figura 8: Esquema general de un controlador PID Clásico

Cada uno de los constituyentes del controlador cumple las siguientes funciones:

-Proporcional: se encarga de hacer que el error entre la consigna y la respuesta del sistema en régimen permanente tienda lo máximo posible a cero. Su ajuste puede provocar la aparición de un sobreimpulso excesivo en el sistema, hecho que ha de evitarse.

-Derivativo: ejecuta una labor predictiva de control, intentando hallar valores futuros del error a partir de su tendencia (pendiente o derivada). Su constante regula los parámetros en régimen transitorio del sistema, como pueden ser el tiempo de subida, el tiempo de establecimiento o el sobreimpulso máximo, confiere mayor oscilación y velocidad de respuesta al sistema a mayor valor, aumentado su riesgo de inestabilidad con su aumento.

-Integrador: reduce el error en régimen permanente, fomentando la acción del control proporcional, pero se diferencia de este en que la acción integradora no solo aplica un factor de proporcionalidad al error para actuar sobre él, sino que toma en consideración todos los valores del error de forma acumulativa por su acción integral. Aumenta, en definitiva, la precisión del sistema y su estabilidad a cambio de su velocidad, controlando sus parámetros en régimen permanente.

El controlador PID Clásico tiene la desventaja de no poder cambiar sus constantes proporcional, derivativa o integral para distintos puntos de equilibrio de nuestro sistema, por lo que hay ciertas ocasiones en que puede desestabilizar la planta o no actuar el controlador en absoluto.

2.4. Controlador PID Fuzzy

El concepto de controlador PID Fuzzy pasa por entender qué es un sistema Fuzzy o borroso (**11**): los controladores borrosos se basan en el concepto de conocimiento heurístico, es decir, aquel basado en el saber humano, en la reglas que imponemos de forma inconsciente a las acciones cotidianas (calentar la comida “un poco”, pisar el pedal del embrague “a fondo”, apretar un botón “con fuerza”,...), es decir, descripciones difusas y no basadas en números ni magnitudes, sino en la abstracción y la experiencia. Los controladores borrosos tienen sus fundamentos a su vez en las redes neuronales, aquellas basadas en la conexión de “neuronas” (funciones matemáticas que reciben unas entradas y generan unas salidas) conectadas entre sí por “pesos” o ganancias ajustados por el sistema a partir del entrenamiento con un conjunto de datos y la posterior validación de la estimación. Aunque este concepto pueda resultar confuso solo hemos de pensar en redes neuronales como sistemas que aprenden solos cómo ajustar sus parámetros para ser capaces de hacer que el sistema responda de la forma más óptima posible a las entradas otorgadas en base a reglas a las que nosotros imponemos que tiene que responder.

El concepto de controlador borroso se basa en la lógica borrosa, que sigue una secuencia de pasos que es la siguiente:

1º. Evaluación de premisas: consiste en ver si se cumple o no un conjunto de reglas lógicas a la entrada del controlador del tipo “If-Then”. Estas reglas constituyen un “Conjunto borroso”, que es la agrupación de normas a las que están sujetas las entradas del sistema. Estos condicionales “If” pueden relacionar entradas entre sí mediante la utilización de conectores como “AND” u “OR”. Un ejemplo de una norma que podría utilizarse en el control de un vehículo es:

If speed is slow AND distance_to_next_stretch is long THEN acceleration is high

Los operadores “is” vinculan el grado de certeza de un valor lingüístico a un conjunto determinado, y construyen lo que se llama “función de pertenencia”. En definitiva, estas funciones de pertenencia indican un grado de probabilidad de que, por ejemplo y siguiendo el caso del control del vehículo, la variable identificada por el concepto lingüístico “speed” (velocidad) pertenezca al conjunto borroso o, dicho de otra forma, pueda definirse como, “slow” (lenta). Al ser un concepto estadístico la pertenencia de un concepto a un conjunto borroso oscila entre “0” (no pertenece en absoluto) y “1” (pertenece totalmente). Las funciones de pertenencia, a su vez, pueden ser de distinto tipo, por ejemplo: triangulares, trapezoidales, gaussianas, cuadradas,... entre otras.

2º. Inferencia a la salida: se aplica el operador (“And”, “Or”,...) a las premisas una vez evaluada la pertenencia de cada una al conjunto borroso indicado. Ello consiste en generar un número que relacione la pertenencia de los conceptos en la misma oración a cada conjunto borroso indicado precisamente para cada uno de ellos, evaluándolo como un todo, y posteriormente esto se aplica a la salida. Diferentes operadores implican diferentes acciones:

-**AND**: mínimo o producto

-**OR**: máximo o suma geométrica ($\mu_A + \mu_B - \mu_A \cdot \mu_B$)

Veamos un ejemplo para interiorizar el concepto:

AND como mínimo:

*If speed is 80% slow AND distance_to_next_stretch is 60% long THEN
acceleration is 60% high*

AND como producto:

*If speed is 80% slow AND distance_to_next_stretch is 60% long THEN
acceleration is 48% high*

OR como máximo:

*If speed is 80% slow AND distance_to_next_stretch is 60% long THEN
acceleration is 80% high*

OR como suma geométrica:

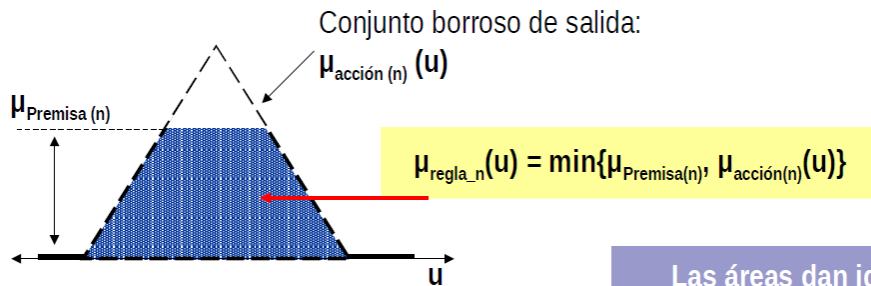
*If speed is 80% slow AND distance_to_next_stretch is 60% long THEN
acceleration is 92% high*

El operador “IS NOT” otorga el complementario de la probabilidad obtenida con “IS”, es decir, si decimos que “speed IS NOT 80% slow” equivaldría a decir que “speed IS 20% slow”. De entre todos estos el operador que más se utiliza es “AND” como mínimo, por su capacidad de estabilizar los sistemas en que se implementa al optar siempre por el menor valor posible.

Una vez determinada la probabilidad conjunta se aplica (multiplica) esta a la salida que forma parte de la conclusión del condicional “IF-THEN”, es decir, en base a las premisas se pondera la salida para todas y cada una de las reglas activas (se considera activa aquella regla en que el resultado de la evaluación conjunta del grado de pertenencia de sus premisas es distinto de cero). Durante este proceso se produce lo que se conoce como borrosificación, es decir, la traducción del lenguaje humano de “IF-THEN” a números asociados a la pertenencia de cada función.

3º. Agregación: una vez obtenida la ponderación de cada una de las salidas a cada uno de los distintos conjuntos borrosos asociados a todas ellas es hora de obtener un número que sirva de “media” o “ponderación” de cada una para poder generar una señal que introducir en el sistema. Este proceso se llama agregación. Para ello suelen seguirse dos métodos fundamentales: truncamiento y escalado, ambos representados en la *Figura 9*:

- **Truncamiento (chop off the top):**



- **Escalado (product):**

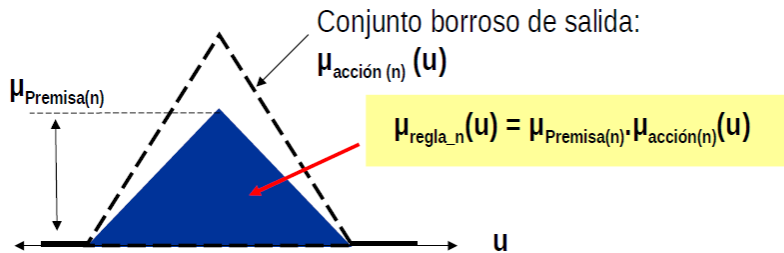


Figura 9: Representación de las dos fundamentales posibilidades de agregación de premisas

4º Desborrosificación: consiste en el proceso a través del cual devolvemos un valor de salida que puede relacionarse directamente con el lenguaje humano, convertimos las conclusiones extraídas del conjunto borroso en un número que simboliza el valor de una magnitud de salida, en definitiva, obtenemos el valor concreto (o “crisp”) resultado de someter a un conjunto de entradas a normas humanas racionales interpretadas por un ordenador. A continuación se muestra el funcionamiento general de: Criterio Máximo (Figura 10), Criterio del Centro de Gravedad COG (Figura 11) y Criterio de los Centros Ponderados (Figura 12):

-Máximo:

■ Criterio máximo

- Se suele utilizar para el caso de **que todas las reglas estén implicadas**

$$u^{crisp} = \operatorname{argsup}_u \{ u_{regla_i} \}$$

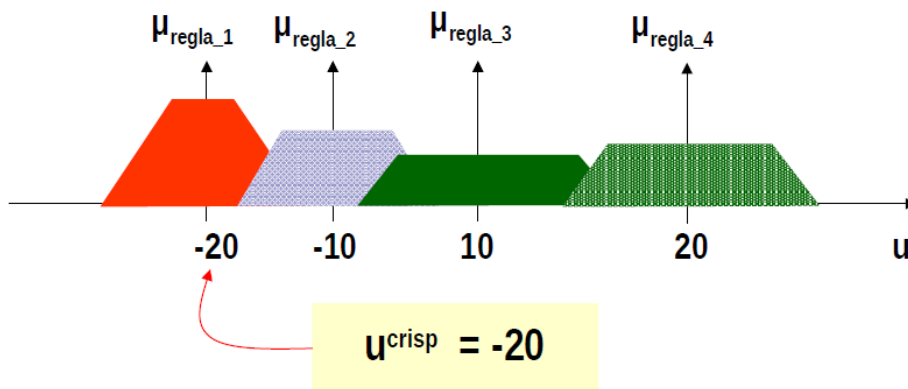


Figura 10: Representación del criterio máximo

-COG o Center Of Gravity:

■ **Ejemplo: COG**

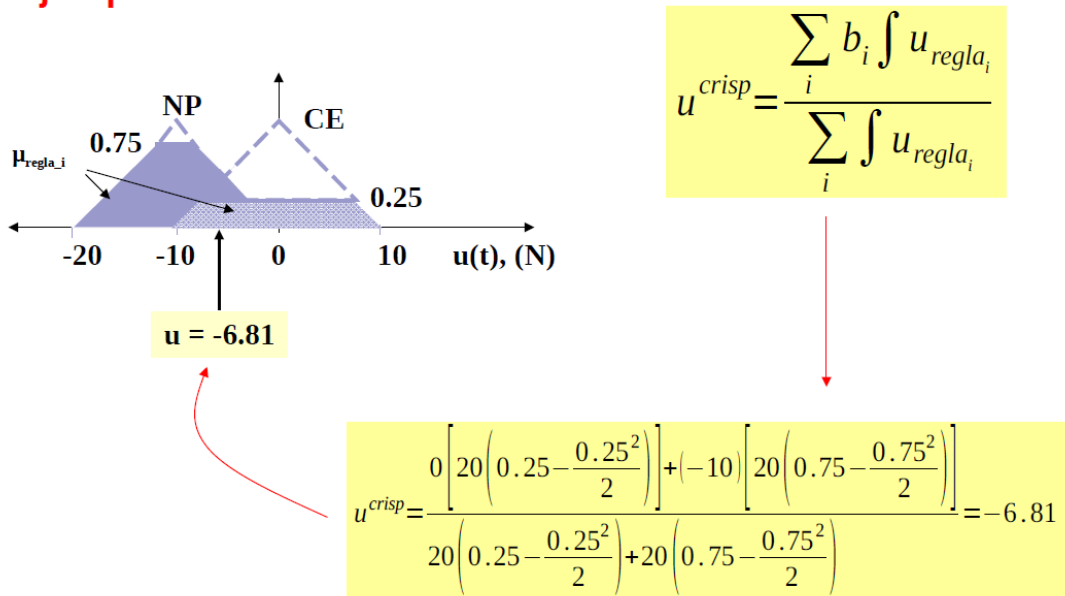


Figura 11: Representación del criterio COG

-Centros ponderados:

■ **Ejemplo: centros ponderados**

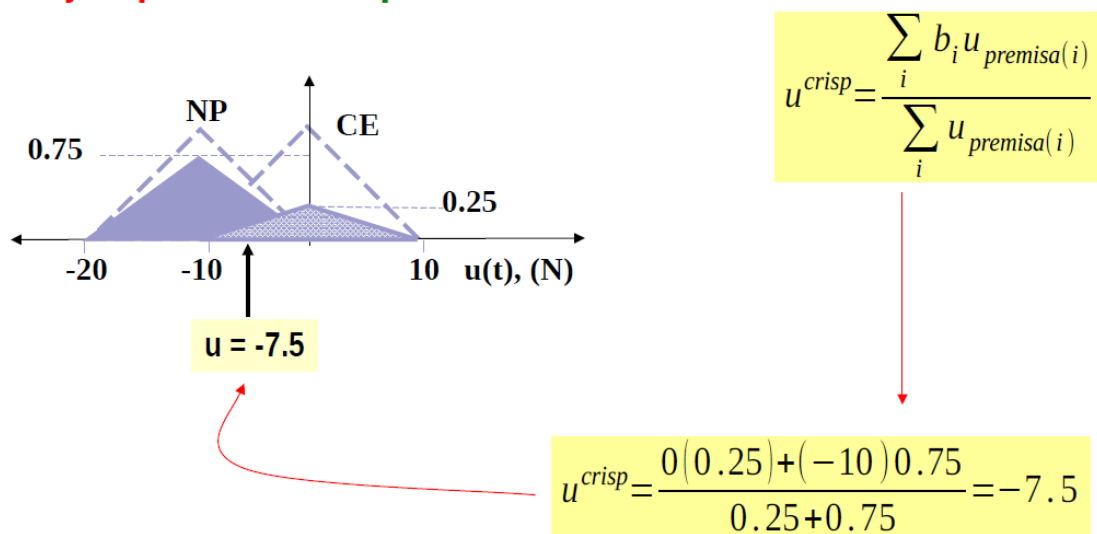


Figura 12: Representación del criterio de Centros ponderados

De entre todos los tipos de sistemas hay dos que se imponen sobre el resto a la hora de obtener u_{crisp} :

Takagi-Sugeno: genera un valor numérico concreto asociado a una función de salida, que es la que se expone a continuación:

$$U_{crisp_j} = \frac{\sum_{i=1}^R f_i \cdot \mu_{premise(i)}}{\sum_{i=1}^R \mu_{premise(i)}}$$

Donde:

f_i = función de pertenencia del conjunto borroso i aplicada la premisa

$\mu_{premise(i)}$ = función de pertenencia de la premisa al conjunto i

R = nº de conjuntos borrosos de entrada

Takagi-Sugeno es poco intuitivo, pero funciona bien con técnicas lineales, de optimización y adaptativas, garantiza continuidad a la salida y se adapta mejor al análisis matemático, como puede verse suele utilizar más el criterio de centros ponderados para realizar la desborrosificación. En la *Figura 13*, se expone a un ejemplo de este sistema.

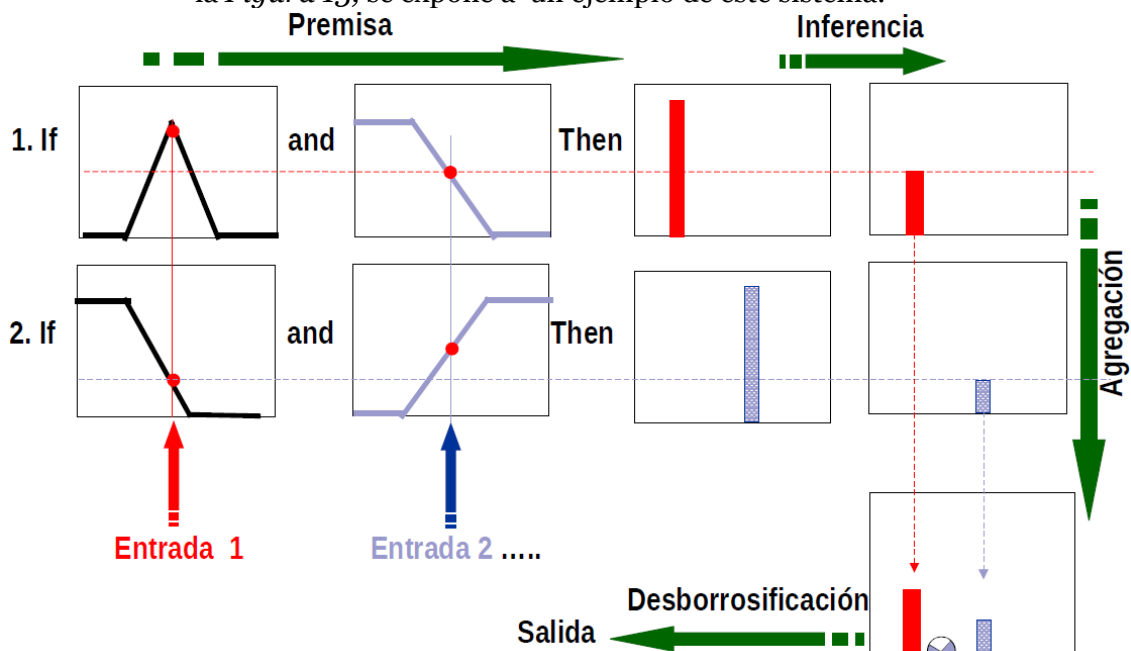


Figura 13: Representación del funcionamiento de un sistema tipo Takagi-Sugeno

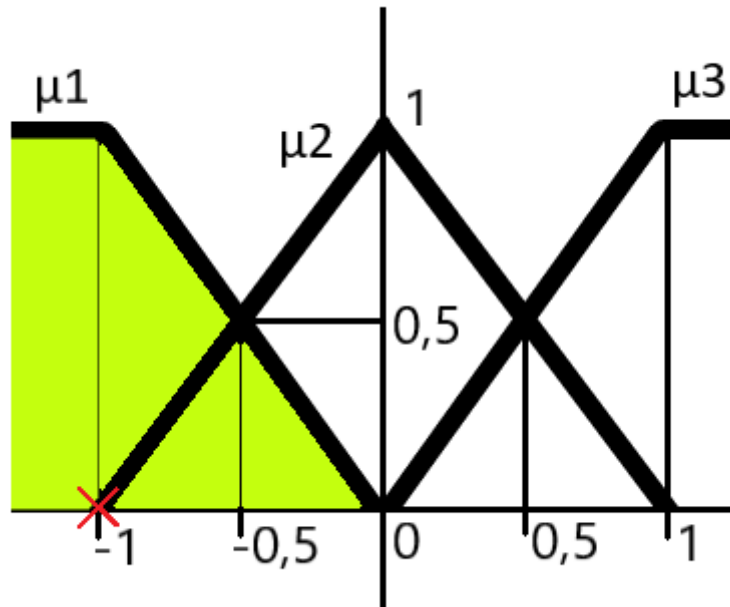


Figura 14: Representación de $\mu_{\text{premisa}(i)}$, con $i=\{1,2,3\}$, con una premisa igual a -1 , donde $\mu_{\text{premisa}(1)}$ y $\mu_{\text{premisa}(3)}$ son trapezoidales, y $\mu_{\text{premisa}(2)}$ es triangular. El área en verde representa la probabilidad con que la premisa pertenece a cada uno de los conjuntos

$$x = -1$$

$$f1 = -x - 2 \rightarrow f1(x) = -1$$

$$f2 = x - 1 \rightarrow f2(x) = -2$$

$$f3 = x + 0,5 \rightarrow f3(x) = 0,5$$

$$U_{\text{crisp}_j} = \frac{\sum_{i=1}^R f_i \cdot \mu_{\text{premisa}(i)}}{\sum_{i=1}^R \mu_{\text{premisa}(i)}} = \frac{u_1 \cdot f_1 + u_2 \cdot f_2 + u_3 \cdot f_3}{u_1 + u_2 + u_3} = \frac{1 \cdot (-1) + 0 \cdot (-2) + 0 \cdot 0,5}{1 + 0 + 0} \quad (3)$$

$$= -1$$

Mamdani: se diferencia de Takagi-Sugeno en que este tipo de agregación es mucho más intuitiva, gráfica y mejor adaptada al lenguaje humano. Suele basarse en la búsqueda de los centros ponderados en base a las áreas de las funciones de pertenencia de salida para generar una constante. Este tipo de sistemas realizan la desborrosificación, habitualmente, a partir del método COG, y es por ello que son más intuitivos que los Takagi-Sugeno. Vemos una representación de este tipo de sistemas en la *Figura 15*:

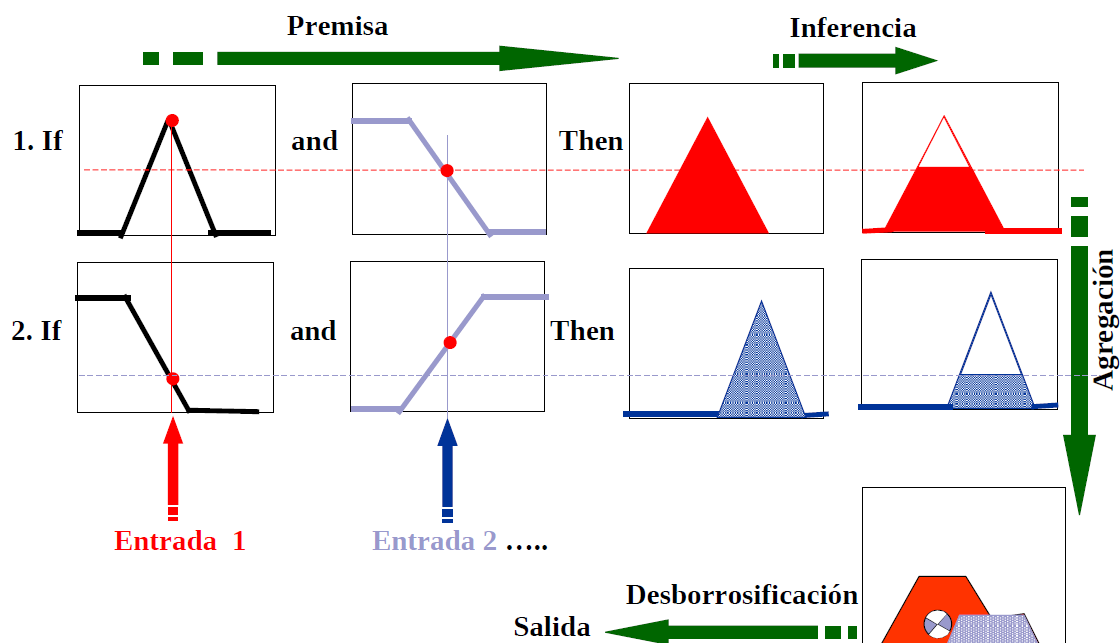


Figura 15: Representación del funcionamiento de un sistema tipo Mamdani

Los controladores PID Fuzzy resultan de la fusión de los conceptos de controlador PID y sistema Fuzzy, es en definitiva un controlador PID cuyos parámetros KP, KI y KD son variables en función del punto de trabajo del sistema, por lo que no están sometidos a las ataduras de trabajar en torno a una posición de equilibrio, sino que pueden ajustarse a distintas situaciones y condiciones iniciales.

2.5. Análisis de los modelos de posición y velocidad del motor

Controlar el motor implica gobernar sobre dos de sus variables físicas principales: la posición y la velocidad angulares. Para poder llevar a cabo esta tarea, primero hemos de ser capaces de digitalizar sendos comportamientos para crear modelos simulados, y así realizar ajustes de los controladores sin necesidad de hacer trabajar al motor para ello. Emplearemos la Toolbox de Matlab de System Identification (12) para hallar los parámetros óptimos de cada modelo y así emular con fidelidad el motor DC. Se expondrán a continuación los tipos de modelos que se acabaron empleando y comparando para modelar las características físicas del motor con la mayor fidelidad posible:

2.5.1. Modelo en Espacio de Estados

El modelado en espacio de estados consiste en la representación de sistemas a partir de ecuaciones diferenciales relacionadas de forma matricial en el dominio del tiempo o de la frecuencia. De esta forma contamos con tres vectores: uno de entradas, uno de salidas, y uno de variables de estado. La representación del modelo en que se apoya el Espacio de Estados (13) es el siguiente (Figura 16):

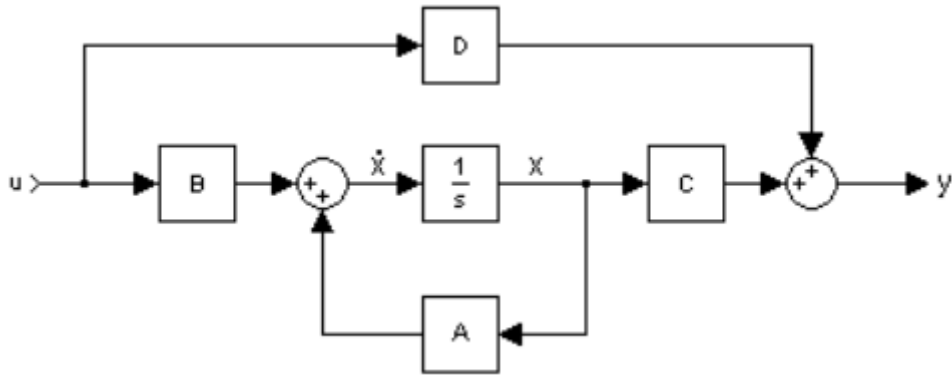


Figura 16: Representación del diagrama de bloques constituyente del modelo en Espacio de Estados (13)

Donde:

- $A(t)$ es la matriz de variables del Espacio de Estados o matriz de estados
- $B(t)$ es la matriz de entrada
- $C(t)$ es la matriz de salida
- $D(t)$ es la matriz de transmisión directa

Y las ecuaciones diferenciales que relacionan las matrices entre sí son:

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.1)$$

$$y(t) = Cx(t) + Du(t) \quad (4.2)$$

Estas pueden expresarse en el dominio de Laplace como:

$$sX(s) = AX(s) + BU(s) \quad (4.3)$$

$$Y(s) = CX(s) + DU(s) \quad (4.4)$$

Pero ya que estamos trabajando con modelos computacionales simulados (y por ende discretos) nos resulta más sensato representarlo en el dominio de la Transformada Z como:

$$zX(z) = AX(z) + BU(z) \quad (4.5)$$

$$Y(z) = CX(z) + DU(z) \quad (4.6)$$

Y despejando vemos que:

$$X(z) = (zI - A)^{-1}BU(z) \quad (4.7)$$

$$Y(z) = (C((zI - A)^{-1}B) + D)U(z) \quad (4.8)$$

Con ello obtenemos la función de transferencia en Espacio de estados como:

$$G(z) = \frac{X(z)}{Y(z)} = C(zI - A)^{-1}B + D \quad (4.9)$$

De manera que el comportamiento del sistema se reduce a averiguar los parámetros que constituyen las matrices del Espacio de Estados. Este tipo de representación es muy robusta y adaptable, puesto que simplemente mediante el cálculo de constantes (siempre y cuando el sistema sea Linear Time-Invariant, que se cumple en nuestro caso) somos capaces de emular el comportamiento de sistemas MIMO (Multiple Input – Multiple Output) de cualquier tipo. En relación con el Motor DC el sistema a modelar será SISO (Single Input – Single Output), por lo que nos resultará incluso más sencillo de obtener su modelo en Espacio de Estados.

2.5.2. Modelo Box-Jenkins

El modelo Box-Jenkins se basa en un sistema autorregresivo de ajuste a una serie temporal determinada **(14)**. Está basado en los modelos ARX y ARMAX. Dicho sistema se utiliza con series numéricas estacionarias (media y varianza constantes en el tiempo) y con tendencias estacionales (patrones repetitivos cada cierto periodo de tiempo y, por ello, predecibles y controlables), por lo que se emplean con asiduidad en el análisis de la tendencia de la bolsa financiera debido a su naturaleza. La metodología Box-Jenkins se implementa en tres pasos:

- **Identificación y selección de modelo regresivo:** una vez comprobado que la serie temporal es estacionaria y estacional (si bien similares en nombre recordemos que estos conceptos aluden a realidades distintas) se selecciona el modelo de regresión, que puede ser Promedio Autorregresivo (AR) o Promedio Móvil (MA).
- **Estimación de parámetros del modelo:** proceso de gran consumo computacional del que puede encargarse una herramienta de simulación como MATLAB. Suelen emplearse dos metodologías: Estimación de Máxima Verosimilitud o Mínimos Cuadrados no Lineales, siendo el primero el preferido habitualmente.
- **Comprobación experimental del modelo:** ver si el ajuste del modelo es fiel a la sucesión que se le ha alimentado, observando si los residuos son reducidos y no hay correlación entre ellos, y que la media y la varianza son constantes en el tiempo (para ello puede emplearse el método de Ljung-Box). En caso de no cumplirse esto se ha de volver a empezar.

Box-Jenkins, en definitiva, estima unos parámetros estadísticos para intentar seguir el comportamiento de una sucesión matemática lo mejor posible, pero por supuesto ello estará sujeto a limitaciones, como la estacionalidad o estacionariedad del modelo.

3. Estudio experimental de los modelos de posición y velocidad del motor

3.1. Modelo de posición

En las numerosas pruebas que se hicieron con la toolbox de System Identification de Matlab se generaron un conjunto de modelos, algunos más satisfactorios que otros, que intentaban emular en la medida de lo posible el comportamiento real del motor a partir de las señales que se expusieron en apartados anteriores. Para generar estas señales se introdujo en el motor consignas aleatorias comprendidas entre 0° y 360° , concediendo al dispositivo un margen de tolerancia de $\pm 5^\circ$ en torno a la misma, para que una exactitud excesiva no produjera una oscilación acuciante en el mismo. Para lograr este objetivo se construyó la estructura en Simulink que muestra la *Figura 17*:

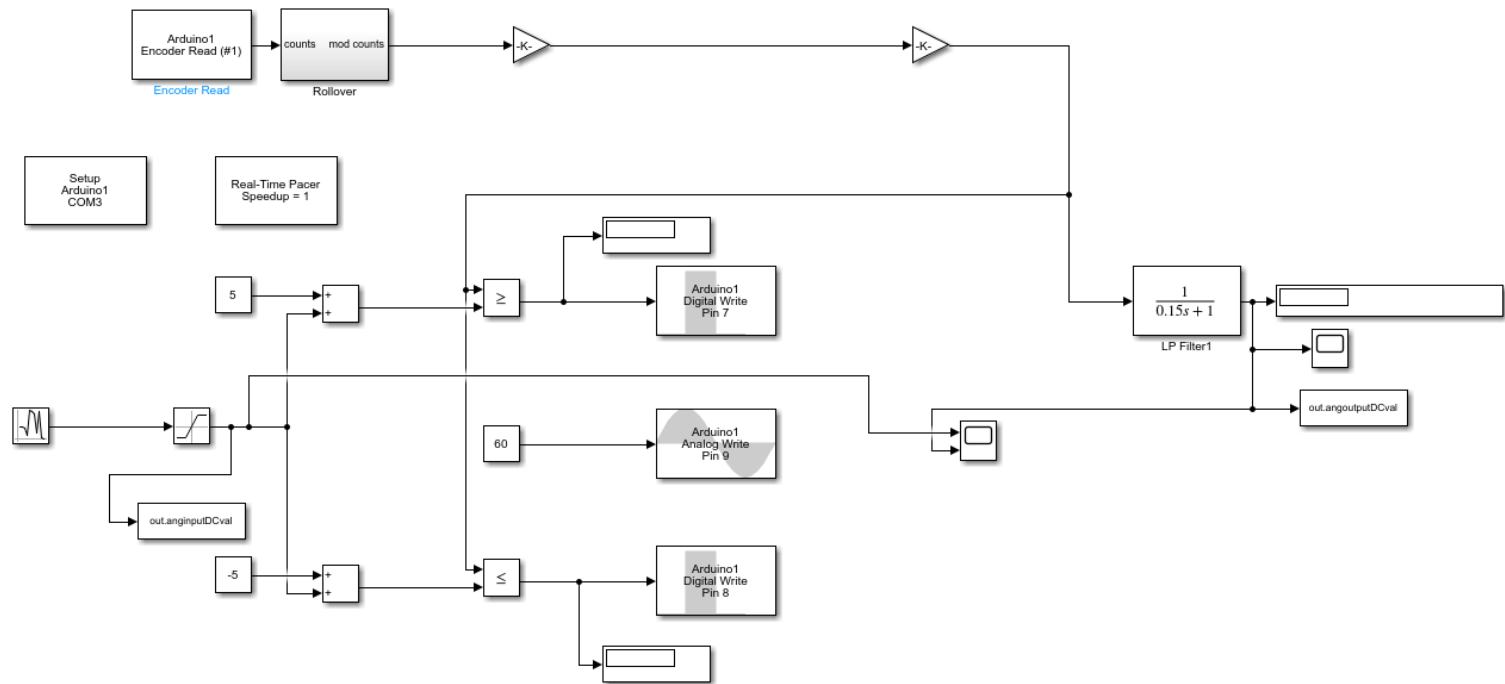


Figura 17: Diagrama de bloques del modelo de posición en Simulink

El funcionamiento de dicho modelo es sencillo: la señal de “Enable” del puente en H (Analog Write Pin 9) está configurado a 60 rpm constantes, la velocidad mínima permitida para su arranque y avance, puesto que no nos interesa controlar la velocidad en este modelo y un menor valor de esta garantizará una mayor precisión. Mientras tanto, se introduce la señal de consigna entre 0° y 360° y se lee la posición del motor a partir de su “Encoder” (Arduino Uno Encoder Read), se traducen los pulsos a cuentas y las cuentas a posición angular en grados. Después se compara este valor con la consigna: si está por encima de esta más 5° el motor gira a derechas (se habilita la entrada “In1” del puente en H a partir de la introducción de un “1” lógico, o sea, 5,5V, a través de la salida Digital Write Pin 7, y un “0” lógico, o sea, 0V, en la entrada “In2” del puente en H a partir del Digital Write Pin 8), y si está por debajo del valor de la consigna menos 5° se gira a izquierdas (“0” en “In1”, “1” en “In2”). Esto provoca un comportamiento oscilatorio del motor alrededor de la referencia. En apartados posteriores se introducirá una solución mejor al problema de la tolerancia en torno a la posición deseada para poder reducir las oscilaciones al mínimo y permitir a su vez que el motor alcance su consigna con la mayor rapidez posible. De esta forma se crearon dos conjuntos, uno de entrenamiento y otro de validación sin correlación alguna, pues ambas señales son completamente aleatorias y distintas (o pseudoaleatorias, tal y como dictamina el comportamiento de los ordenadores), expuestos en las Figuras 18 y 19, respectivamente:

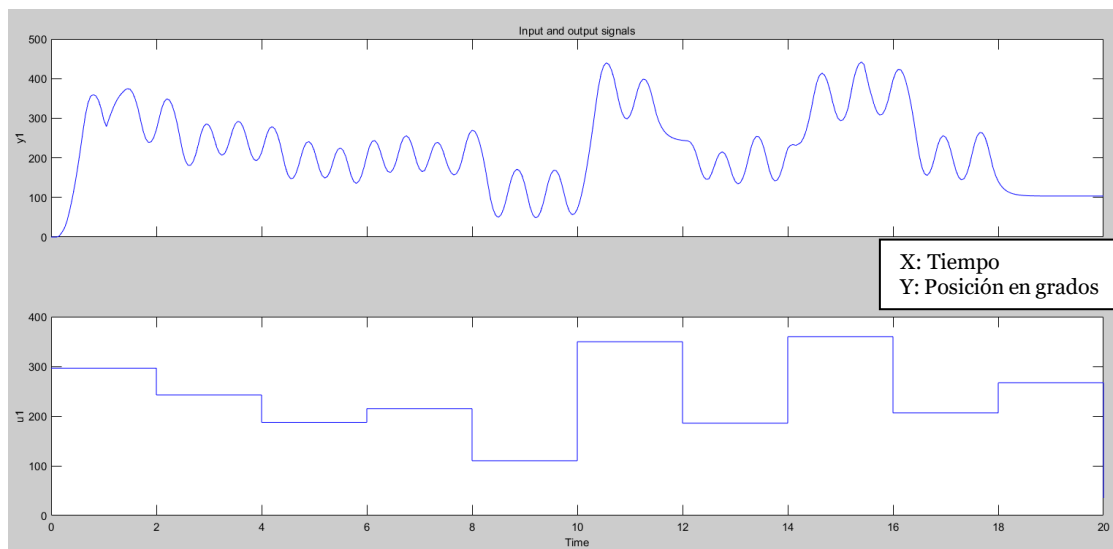


Figura 18: Datos de entrenamiento (gráfica superior: output, gráfica inferior: input)

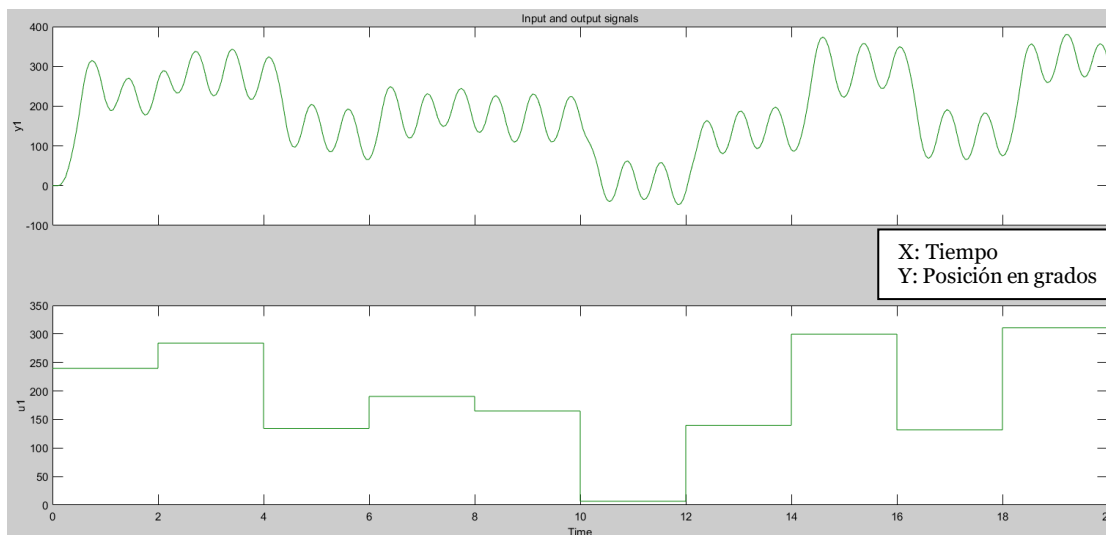


Figura 19: Datos de validación (gráfica superior: output, gráfica inferior: input)

En los siguiente apartados se muestra y explica el funcionamiento de dichos modelos, empleando para ello los resultados observados y los parámetros medidos en cada caso.

3.1.1. Modelo de posición Box-Jenkins

Tras numerosas pruebas y modificaciones de los parámetros de Box-Jenkins se obtuvo un modelo que se ajustaba a la realidad lo suficiente como para poder ser utilizado para simular el comportamiento del motor real. Los resultados obtenidos se muestran en las siguientes figuras (Figura 20 y 21).

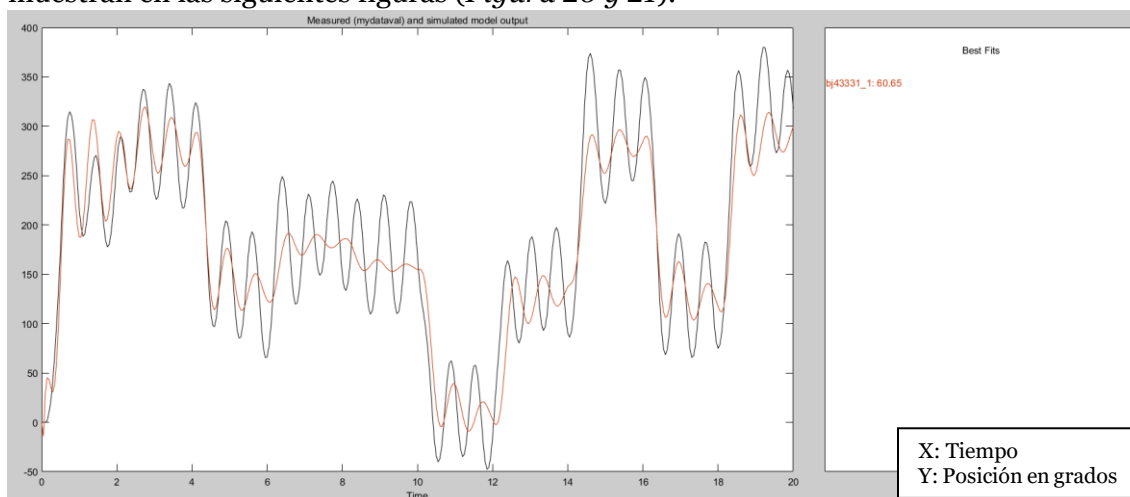


Figura 20: Comportamiento del modelo de posición Box-Jenkins (naranja) en comparación al real (gris)

Discrete-time BJ model: $y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)$
 $B(z) = -0.006795 z^{-1} + 0.02216 z^{-2} - 0.004148 z^{-3} + 0.0116 z^{-4}$
 $C(z) = 1 - 0.2451 z^{-1} - 0.1817 z^{-2} + 0.1551 z^{-3}$
 $D(z) = 1 - 2.727 z^{-1} + 2.651 z^{-2} - 0.9211 z^{-3}$
 $F(z) = 1 - 2.607 z^{-1} + 2.412 z^{-2} - 0.7807 z^{-3}$

Figura 21: Representación matemática del modelo Box-Jenkins obtenido

Con un ajuste del 60,65% el modelo Box-Jenkins no presenta una oscilación tan amplia como el real, pero sus parámetros de media y varianza son lo suficientemente fieles a la realidad como para poder emplearse en el “tuning” de los controladores. En primera instancia se decidió probar el modelo sin ningún tipo de control para medir sus parámetros en régimen transitorio y permanente más significativos a partir del siguiente esquema de Simulink (Figura 22).

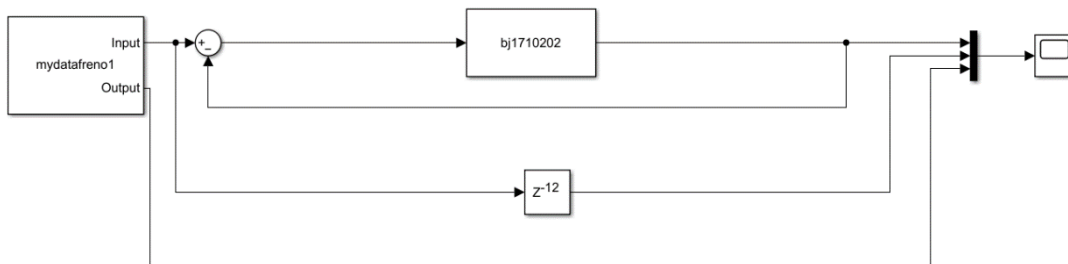


Figura 22: Modelo Box-Jenkins sin control PID

El bloque central “bj1710202” es el modelo Box-Jenkins obtenido de las pruebas con la Toolbox de System Identification, que ha sido posteriormente exportado e implementado en un bloque de Simulink. Por su parte, el retardo se introdujo para realizar una comparación fiel entre la entrada y la salida, puesto que existe un “delay” de unos 0,6s entre ambas señales (dado que el tiempo de muestreo es de 0.05s podemos deducir que 0.6s es 12 veces el tiempo de muestreo, y por ende, Z^{-12} en el dominio discreto). Introduciendo señales tipo escalón de 10° , 160° y 270° , y obteniendo los resultados que se muestran en la Figura 23, en que se expone la salida de posición del modelo ante dichas referencias:

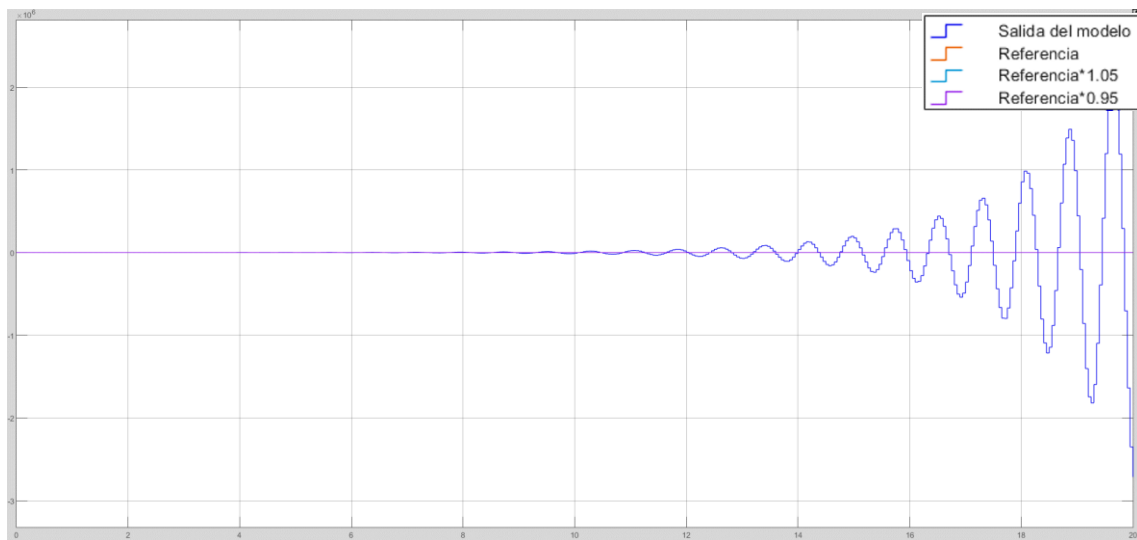


Figura 23 a)

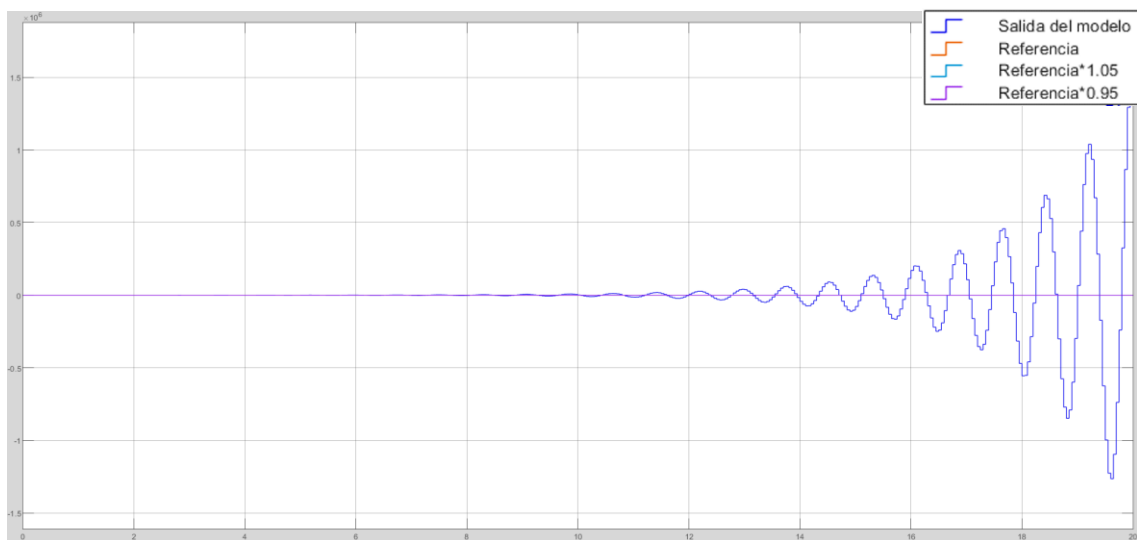


Figura 23 b)

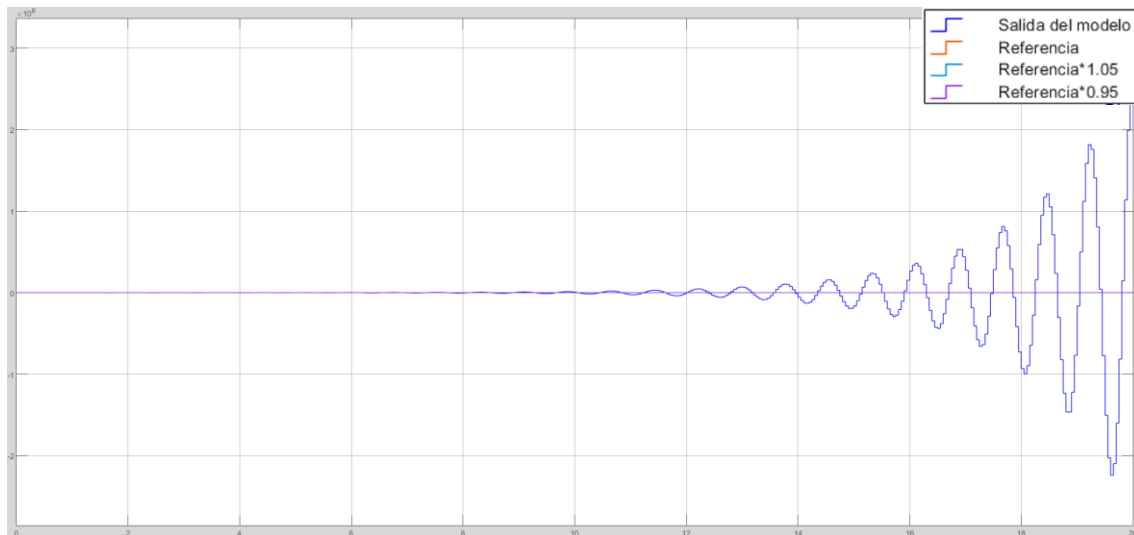


Figura 23 c)

Figura 23: a) Salida del modelo Box-Jenkins con una referencia de 10° y sin controlador, b) Salida del modelo Box-Jenkins con una referencia de 160° y sin controlador, c) Salida del modelo Box-Jenkins con una referencia de 270° y sin controlador

Consideramos que el sistema sufre de una inestabilidad insalvable independientemente de la entrada, por lo que no podemos medir parámetros temporales ni magnitudes del mismo. El motor real no llegaba a perder su estabilidad a pesar de no disponer de ningún tipo de control cuando se realizaron pruebas en el mismo, por lo que se intuye que este descontrol puede ser cosa del modelo o de los parámetros de la simulación. Visto esto concluimos que es necesaria la presencia de algún tipo de controlador para que el sistema, al menos simulado, funcione. Ante este obstáculo, por ende, concluimos que la ruta de actuación óptima que ha de seguirse es la del diseño de dos tipos de controladores PID, uno Clásico y otro Borroso, para poder comparar sus capacidades de control sobre la planta y ser capaces de imponer uno sobre el otro.

A continuación, se llevó a cabo un ajuste manual de los parámetros del controlador PID ante una señal de tipo escalón, empleando la metodología de Ziegler-Nichols **(15)**, llevando a cabo los siguientes pasos:

- Primero se averiguó KP (constante de proporcionalidad) aumentando el valor de este parámetro hasta que el sistema generara una señal oscilante de amplitud aproximadamente constante (o dicho de otra forma justo antes de generar inestabilidad) y se dividió a la mitad.
- Posteriormente se aumentó el valor de KI (constante de integración) hasta que el sistema siguiera a la consigna escalón con suficiente precisión, sin producir un sobreimpulso elevado.

- Finalmente, KD (constante de derivación) se obtuvo dando valores crecientes a la constante, hasta que la oscilación del sistema se redujera lo suficiente y los parámetros en régimen transitorio fueran aceptables, aumentando la velocidad de respuesta del modelo lo máximo posible sin generar descontrol. En definitiva, se intentó reducir al máximo el tiempo de subida del sistema sin producir inestabilidad.

Para poder realizar las pruebas se empleó el siguiente modelo de Simulink (Figura 24):

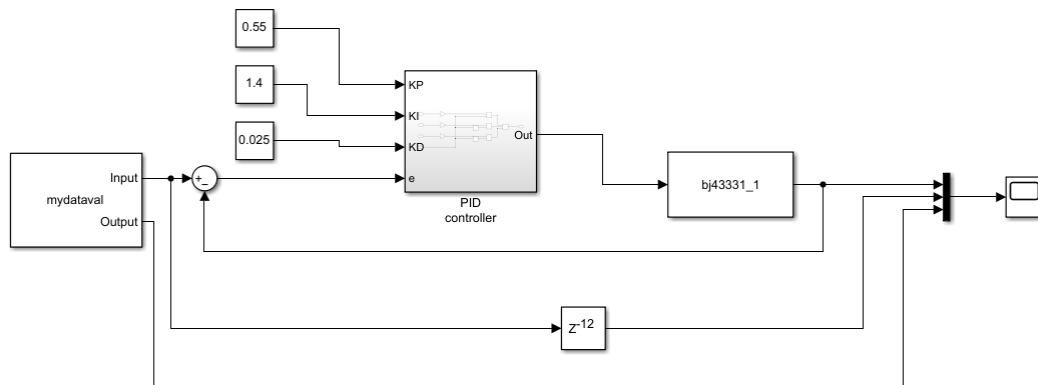


Figura 24: Modelo de posición Box-Jenkins con Control PID (tómese en consideración que, aunque el valor de KP sea el doble del indicado a continuación, en el bloque PID controller hay una ganancia de 0.5 que lo multiplica, dejándolo a la mitad) (15)

Tras numerosos intentos se obtuvieron los valores de los parámetros que se muestran a continuación:

$$KP = 0,225; KI = 1.4; KD = 0.025$$

Una vez configurado se realizaron pruebas con los mismos escalones que se emplearon en el modelo sin controlador, obteniéndose lo siguiente:

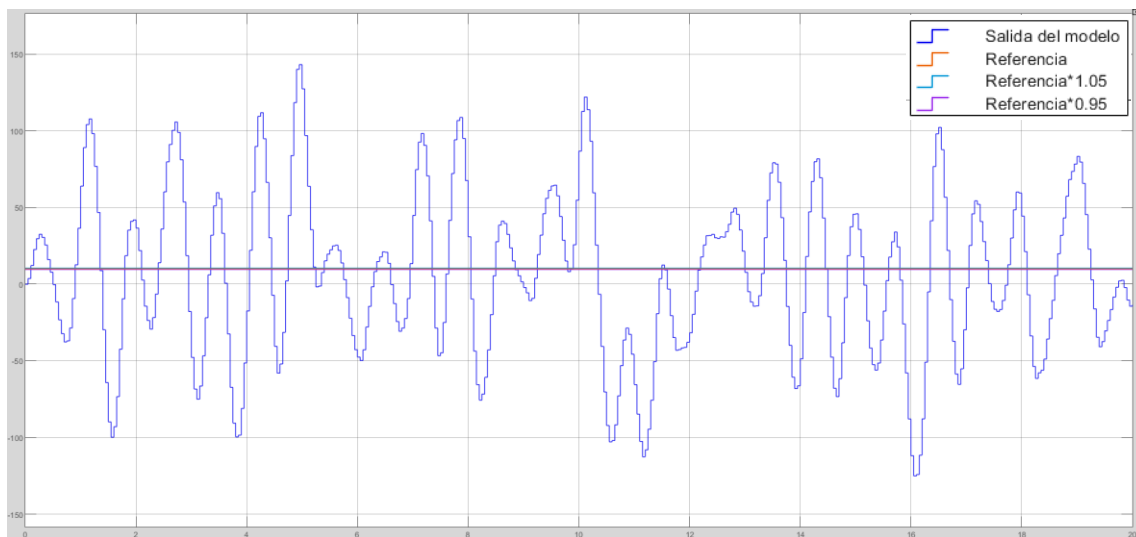


Figura 25 a)

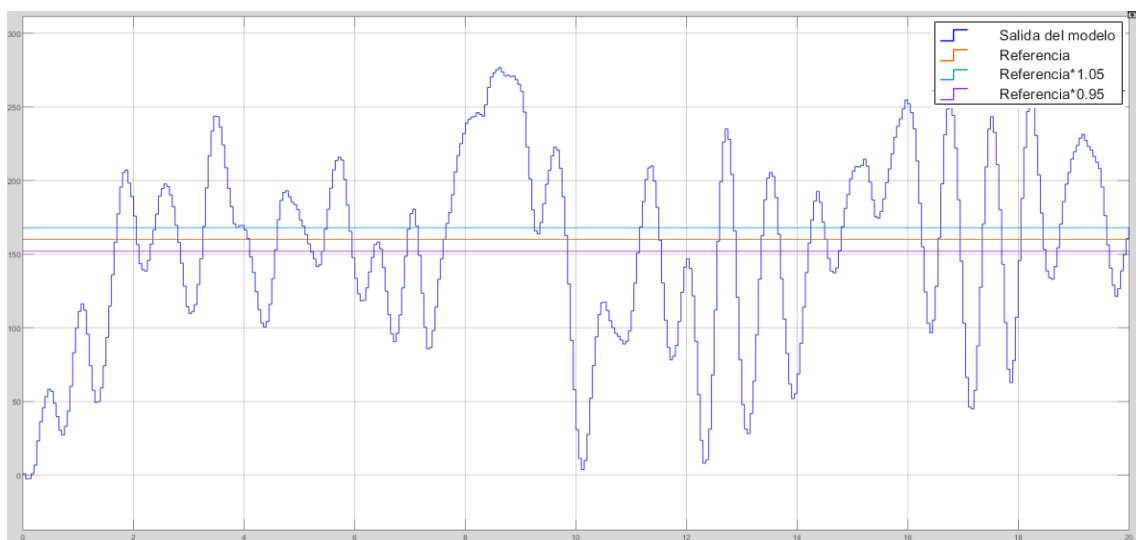


Figura 25: b)

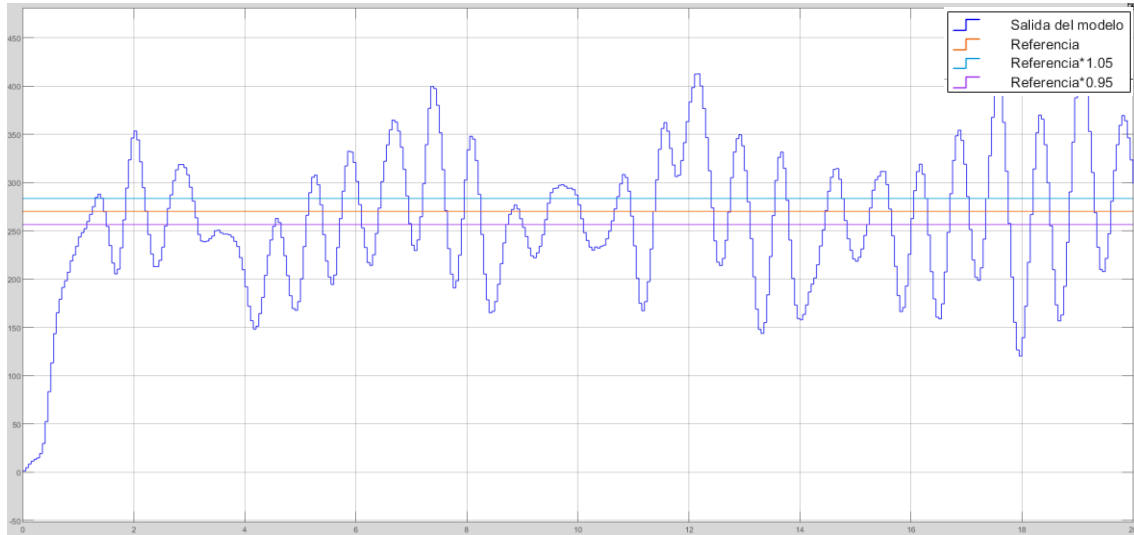


Figura 25: c)

Figura 25: a) Salida del modelo con una referencia de 10° y con PID, b) Salida del modelo con una referencia de 160° y con PID, c) Salida del modelo con una referencia de 270° y con PID

La realización de estas pruebas sobre el modelo nos permite medir sus parámetros característicos en régimen permanente y transitorio. Dada la naturaleza del sistema consideraremos régimen permanente aquel alcanzado cuando el sistema profesa una señal sinusoidal de amplitud aproximadamente constante. Los datos obtenidos se detallan en la *Tabla 1*:

Tabla 1: Parámetros del sistema Box-Jenkins ante entradas tipo escalón con un controlador PID clásico, en régimen permanente y transitorio

Escalón de referencia	de	Tiempo de subida (5%-95%)	Tiempo de establecimiento	de	Sobreimpulso máximo
10°		0.05s	-		1298%
160°		1.417 s	-		72,875%
270°		0.958s	-		58.41%

Una vez conseguido el controlador PID para el sistema, se procede a implementar el controlado PID Fuzzy. Para ello, se diseñaron los conjuntos de entrada y salida del controlador PID Fuzzy, de manera que las constantes que integran el PID clásico sirvieron como punto de partida para determinar los conjuntos de salida del mismo. Se empleó el siguiente modelo de Simulink para realizar las pruebas (*Figura 26*):

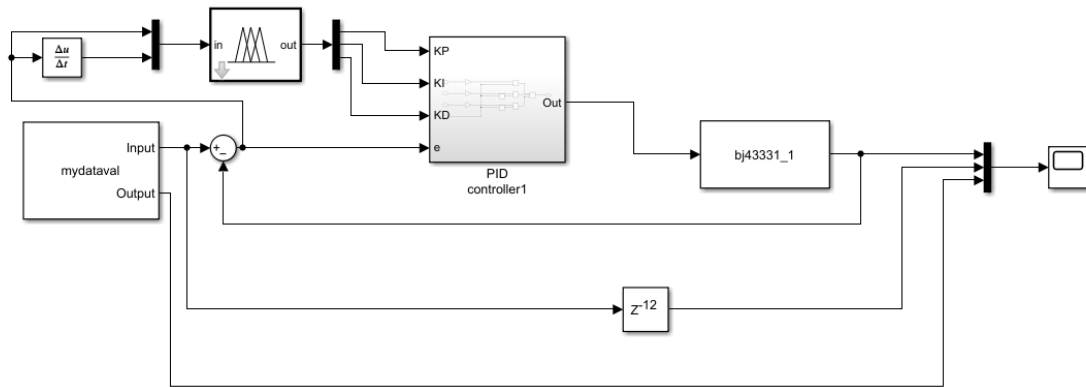


Figura 26: Modelo Box-Jenkins con Controlador PID Fuzzy

Partiendo de los valores de las constantes obtenidos para el Controlador PID Clásico se diseñó el PID Fuzzy (o Borroso) **(15)**, para ello se tomaron como variables de entrada el error (e) generado al comparar la salida con la entrada o referencia y la derivada del error (de). Como salida se generaron las variables que forman parte de cada una de las acciones que realiza el controlador: KP (proporcional), KI (integral) y KD (derivativa), convirtiéndose en parámetros dinámicos del controlador que se ajustan en función de la tendencia de la señal de salida en comparación a la de entrada (error). Las funciones de pertenencia aplicadas a las entradas son triangulares y trapezoidales, y se distribuyen de forma que los valores intermedios de las mismas tengan mayor patencia que los extremos, mientras que los de salida (aun siendo también trapezoidales y triangulares) están repartidas de forma más equilibrada a lo largo de toda la amplitud de valores para cada variable. Todos estos conceptos pueden adquirirse de una forma mucho más visual a partir del conjunto de imágenes comprendidas bajo el nombre *Figura 27*, que están integradas por capturas de pantalla del Controlador Fuzzy diseñado con la ayuda de la Toolbox de Matlab de Fuzzy Logic **(16)**:

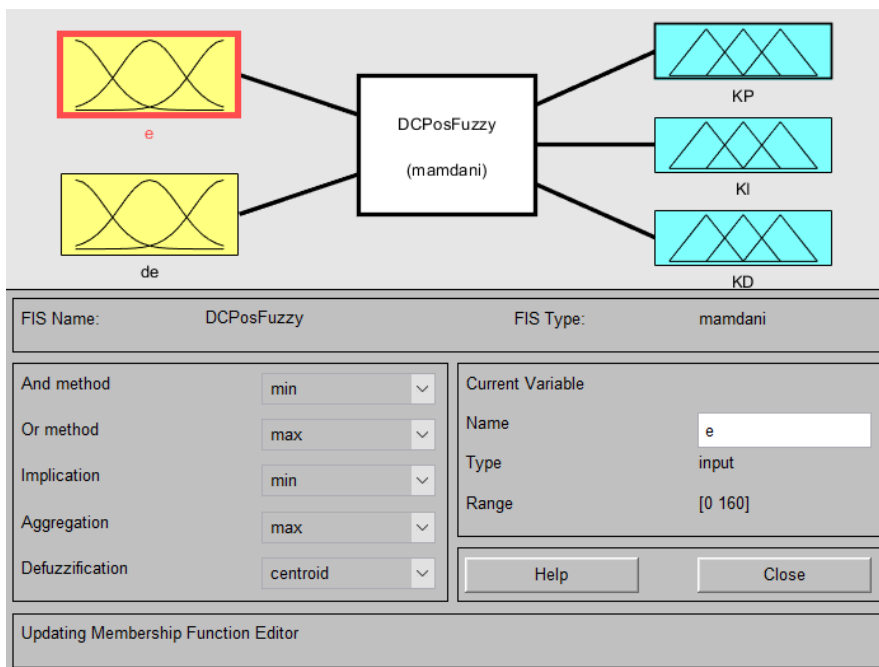


Figura 27 a)

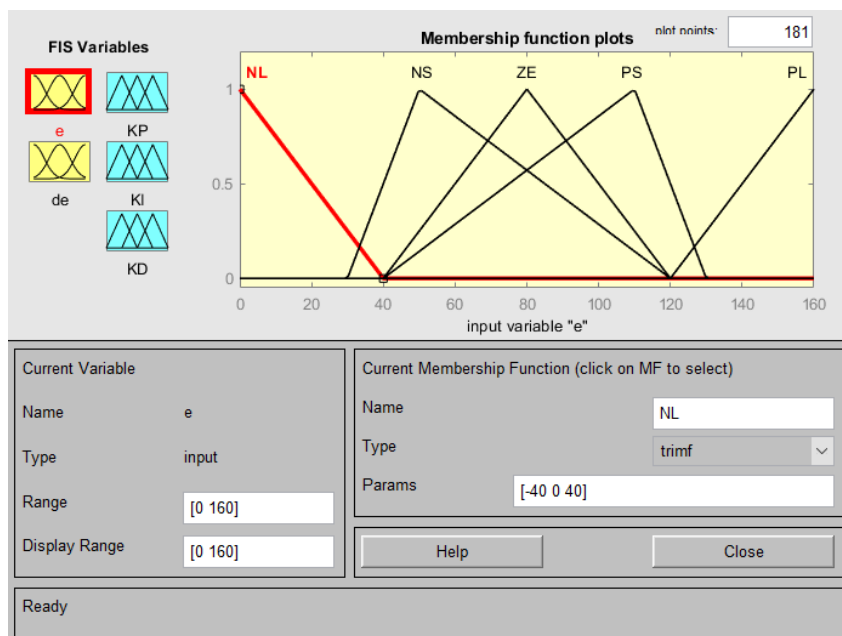


Figura 27 b)

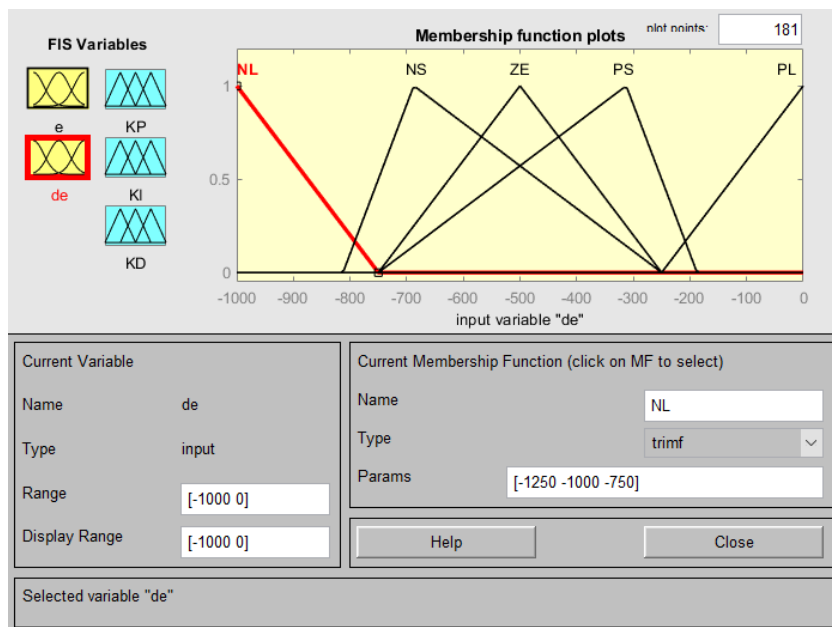


Figura 27 c)

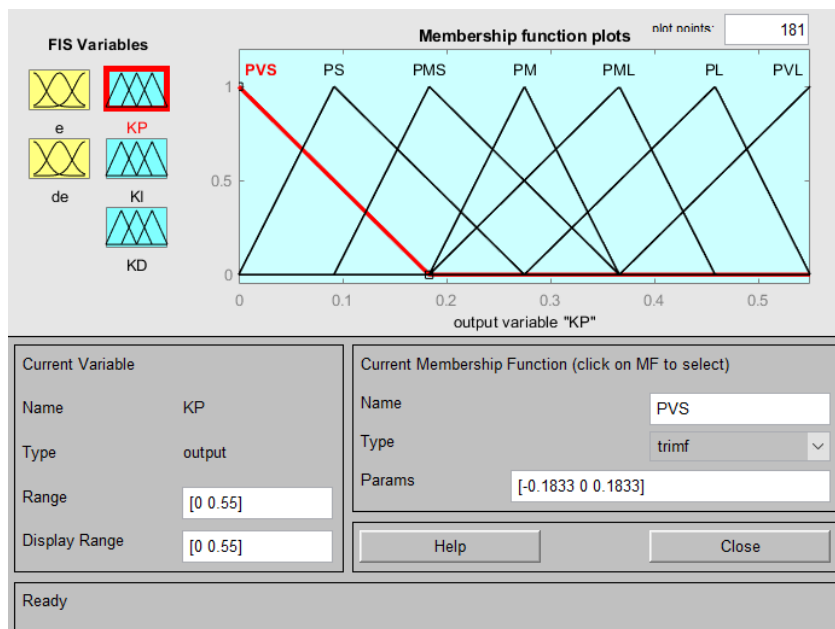


Figura 27 d)

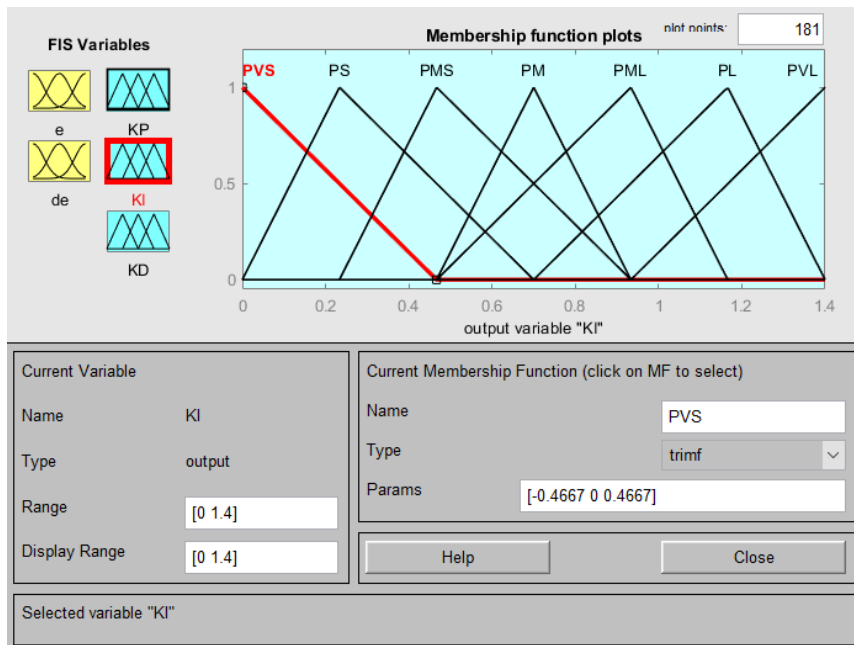


Figura 27 e)

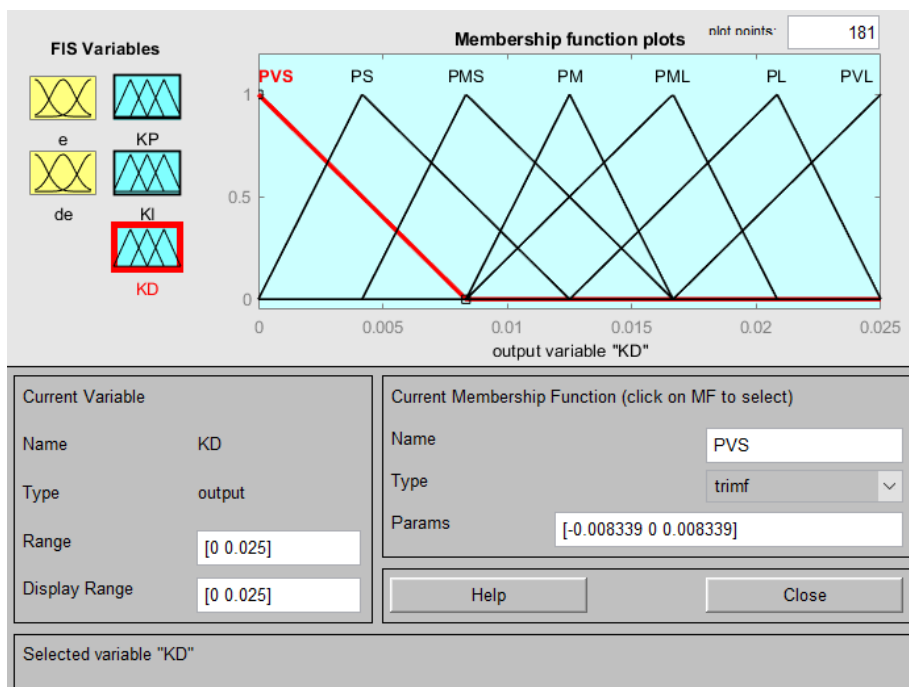


Figura 27 f)

Figura 27: a) Controlador PID Fuzzy para el modelo de posición Box-Jenkins, b) Primera entrada del Controlador PID Fuzzy para el modelo de posición Box-Jenkins: $e(t)$, c) Segunda entrada del Controlador PID Fuzzy para el modelo de posición Box-Jenkins: $de(t)$, d) Primera salida del Controlador PID Fuzzy para el modelo de posición

Box-Jenkins: KP, e) Segunda salida del Controlador PID Fuzzy para el modelo de posición Box-Jenkins: KD, f)
Tercera salida del Controlador PID Fuzzy para el modelo de posición Box-Jenkins: KI

Al implementar este controlador dinámico sobre las mismas consignas se consiguieron los resultados que se muestran en la *Figura 28*:

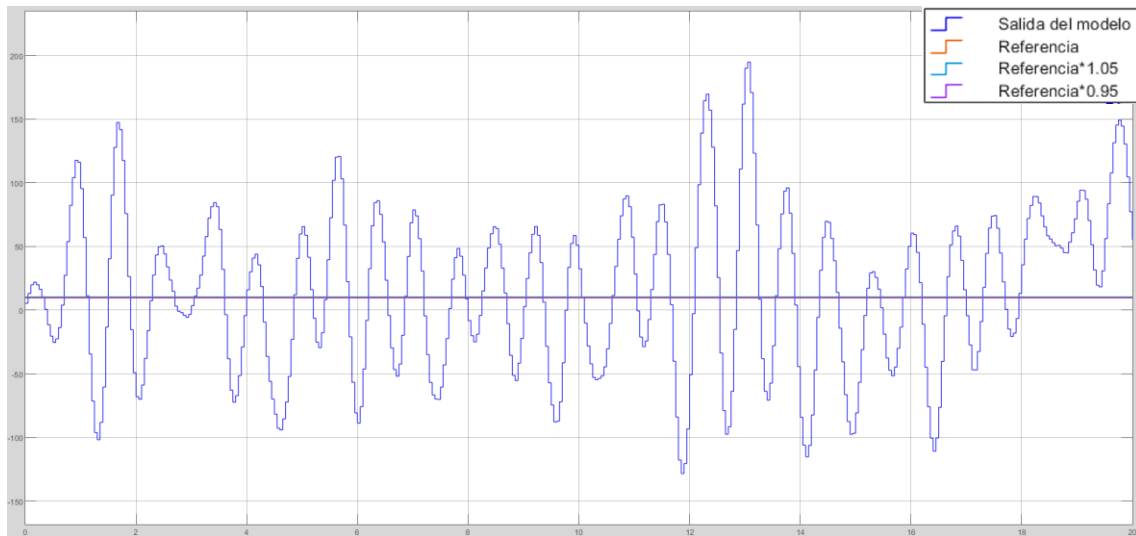


Figura 28 a)

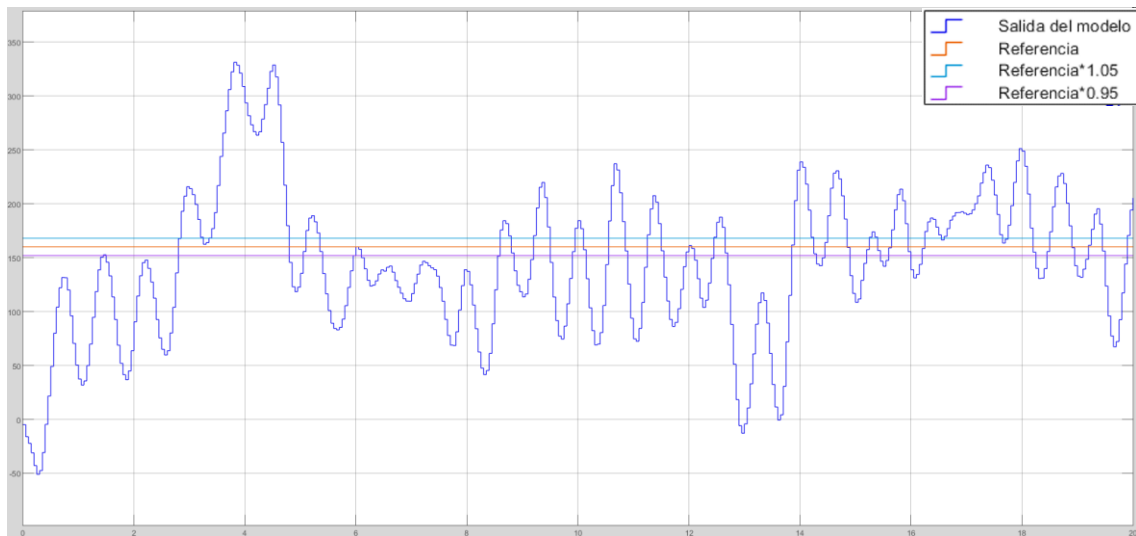


Figura 28 b)

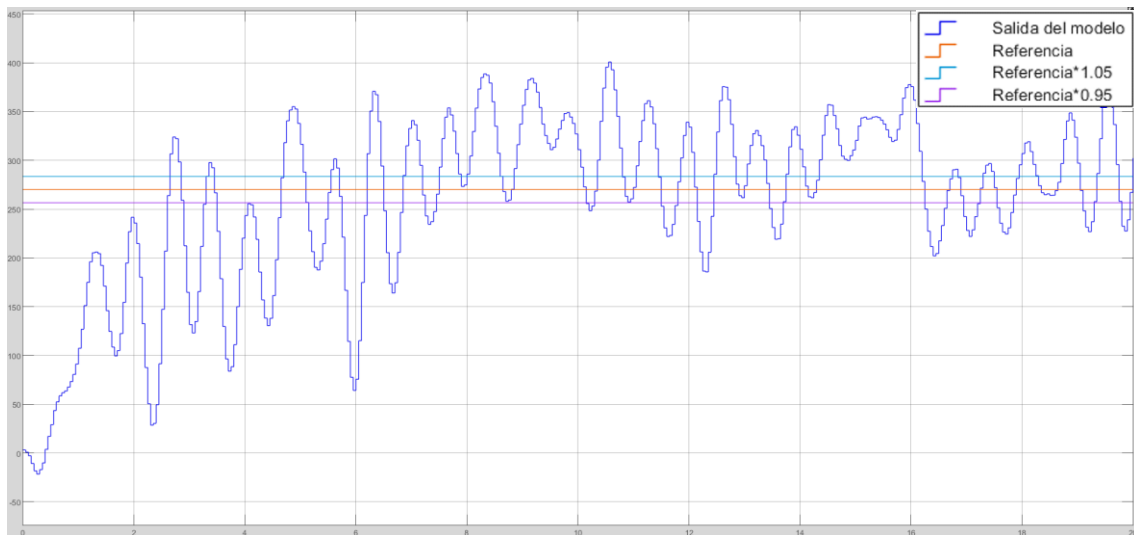


Figura 28 c)

Figura 28: a) Salida del modelo de posición Box-Jenkins con una referencia de 10° y con PID Fuzzy, b) Salida del modelo de posición Box-Jenkins con una referencia de 160° y con PID Fuzzy, c) Salida del modelo de posición Box-Jenkins con una referencia de 270° y con PID Fuzzy

De nuevo la estabilidad permite la medición de parámetros en régimen transitorio, detallados en la *Tabla 2*:

Tabla 2: Parámetros del sistema Box-Jenkins ante entradas tipo escalón con un controlador PID Fuzzy, en régimen permanente y transitorio

Escalón de referencia	de	Tiempo de subida (5%-95%)	Tiempo de establecimiento	de	Sobreimpulso máximo
10°		0.05s	-		1519%
160°		0.997s	-		102%
270°		2.152s	-		48.41%

Al igual que en el caso anterior el sistema sufre un sobreimpulso excesivo cuando la referencia es pequeña. También se aprecia que el controlador PID Fuzzy genera un sobreimpulso mayor ante valores intermedios (ni demasiado grandes ni demasiado pequeños) que el Controlador PID Clásico, aunque el tiempo de subida sea menor. A priori, parece que este conjunto de experimentos otorga la victoria al Controlador PID Clásico sobre el Fuzzy. Como puede apreciarse, no se puede hablar de tiempo de establecimiento, pues la señal no se centra en la banda de tolerancia del 5% en ninguno de los casos, hecho que también se manifestaba con el Controlador PID Clásico.

Para poder ser capaces de discernir cuál de los dos modelos, si el de Box-Jenkins o el de Espacio de Estados, es el mejor para representar el motor, y por tanto, a partir de los resultados de qué modelo debemos partir, someteremos primero al de Box-Jenkins a unas entradas que podemos considerar “prácticas” (es decir, que bien podrían darse en el mundo real), y compararemos las conclusiones arrojadas por esta prueba con las

obtenidas en Espacio de Estados al implementar los mismos pasos que se siguieron con Box-Jenkins. Las señales denominadas “prácticas” se exponen en la *Figura 29*:

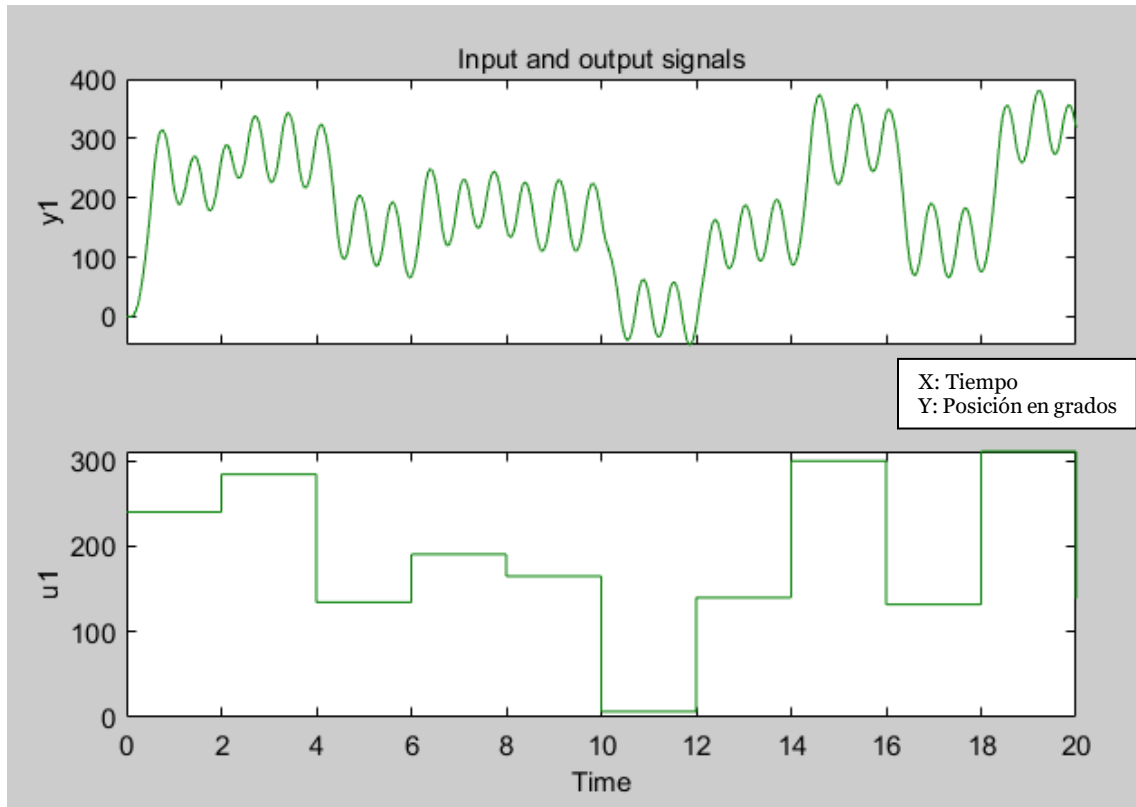


Figura 29: Entrada (gráfico inferior) alimentada al motor real y salida (gráfico superior) otorgada por el mismo, esta señal coincide con los datos de validación del modelo

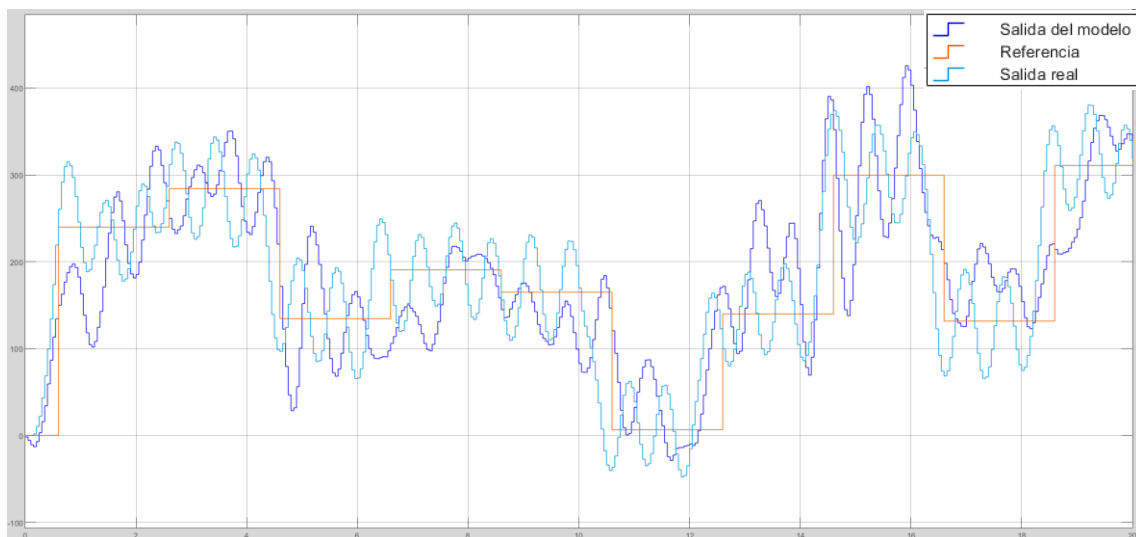


Figura 30 a)

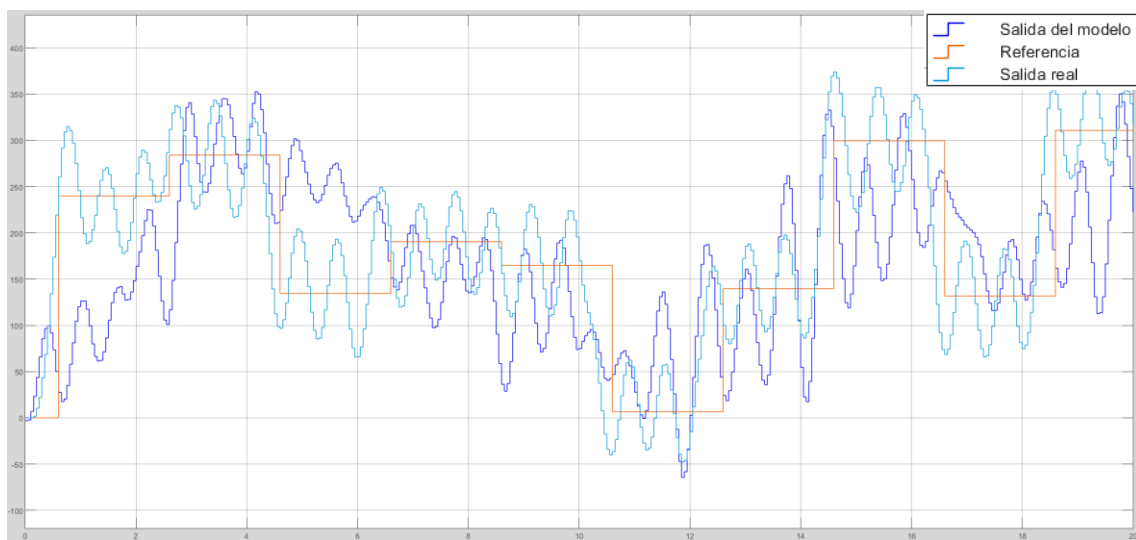


Figura 30 b)

Figura 30: a) Salida del modelo Box-Jenkins con Controlador PID Clásico (azul marino), referencia (naranja) y salida real del motor DC con Controlador PID Clásico (cian), b) Salida del modelo Box-Jenkins con Controlador PID Fuzzy (azul marino), referencia (naranja) y salida real del motor DC con Controlador PID Fuzzy (cian)

Con los datos de los que disponemos es apreciable el hecho de que sin ningún tipo de controlador el sistema es incapaz de seguir las señales introducidas con la precisión deseada, por lo que la implementación del mismo es necesaria. Ahora bien, el Controlador PID Clásico demuestra su eficacia por encima del PID Fuzzy, siendo más rápido, preciso, y estable. El PID Clásico muestra poca oscilación de la señal y regulación rápida de la misma, cosa que, a pesar de su capacidad de manipulación de los parámetros proporcional, integral y derivativo, no es capaz de presentar el PID Fuzzy.

3.1.2. Modelo de posición en Espacio de Estados

El modelo en Espacio de Estados debería ofrecer por su naturaleza oscilaciones menores que el Box-Jenkins, lo cual puede verse simultáneamente como una ventaja, al meter menos ruido en el sistema, o como un impedimento, al no ser capaz de imitar con fidelidad el amortiguamiento del modelo real. De nuevo, primero escogemos los parámetros del modelo hasta lograr que se ajuste lo mejor posible al sistema real a partir de los conjuntos de entrenamiento y validación expuestos previamente (Vaya a las *Figuras 31* y *32* para observar el modelo obtenido, y las *Figuras 18* y *19* para ver los conjuntos de entrenamiento y validación):

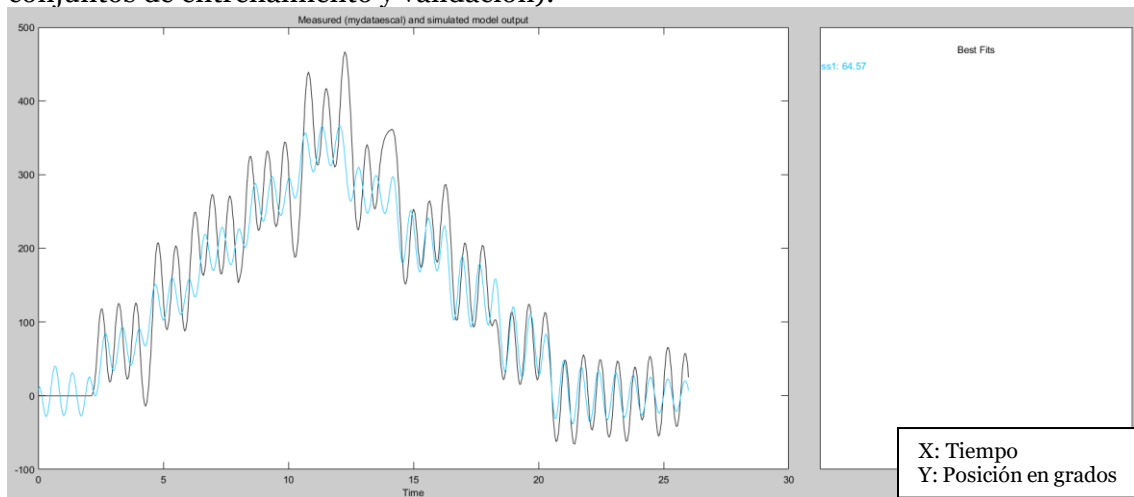


Figura 31: Representación de la salida del modelo en Espacio de Estados (cian) frente a la real del Motor DC (negro)

Continuous-time identified state-space model:

$$\dot{x}(t) = A x(t) + B u(t) + K e(t)$$

$$y(t) = C x(t) + D u(t) + e(t)$$

A =

	x1	x2	x3	x4
x1	-0.5231	4.944	-0.2552	1.15
x2	-5.034	-0.7219	7.585	-4.148
x3	-0.6519	-6.103	-0.7621	6.382
x4	0.1836	-0.789	-2.639	-5.042

B =

	u1
x1	0.0001925
x2	-0.002641
x3	0.003547
x4	6.432e-05

C =

	x1	x2	x3	x4
y1	1479	174.7	33.98	17.69

D =

	u1
y1	0

K =

	y1
x1	0.0111
x2	0.07489
x3	0.05797
x4	0.07638

Figura 32: Representación matemática del modelo de posición en Espacio de Estados

Montamos el mismo esquema que en el caso del BJ pero cambiando el modelo sobre el que actuamos, realizando pruebas otra vez con 10° , 160° y 270° para ver su reacción, obteniendo lo siguiente sin ningún tipo de control (Figura 33):

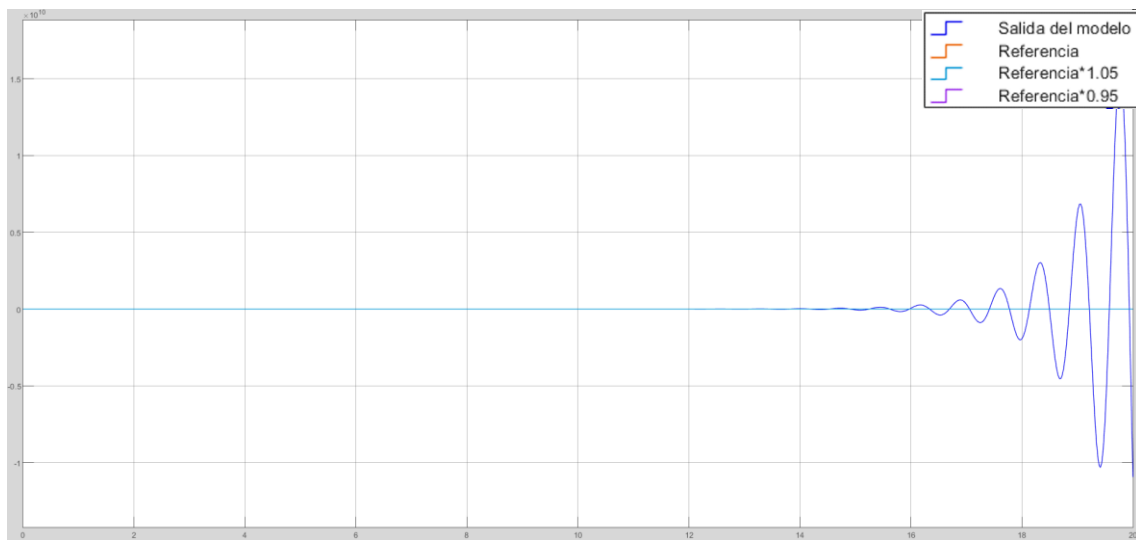


Figura 33 a)

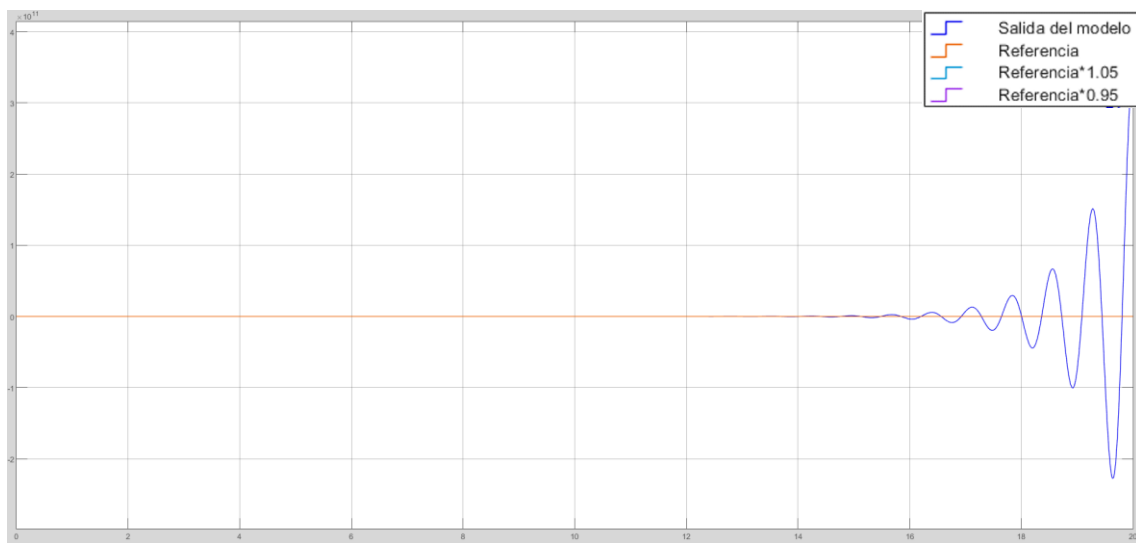


Figura 33 b)

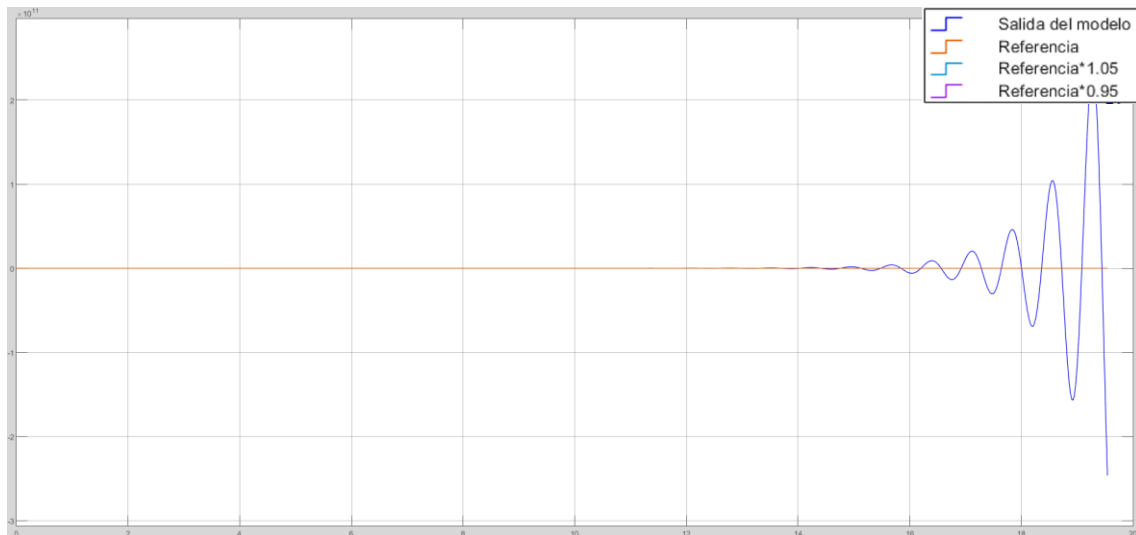


Figura 33 c)

Figura 33: a) Salida del modelo en Espacio de Estados con una referencia de 10° y sin controlador, b) Salida del modelo en Espacio de Estados con una referencia de 160° y sin controlador, c) Salida del modelo en Espacio de Estados con una referencia de 270° y sin controlador

De nuevo el modelo se desestabiliza sin controlador, aunque el sistema real no lo haga. Puesto que este hecho acontece tanto en Box-Jenkins como en Espacio de Estados podemos concluir que es causa de algún parámetro desconocido mal ajustado en Simulink o algún problema propio de los bloques de System Identification, y no algo intrínseco al propio modelo. Procedemos a implementar y ajustar el PID clásico siguiendo el mismo procedimiento que en Box-Jenkins (“Tunning” manual de Ziegler-Nichols) **(15)**, obteniendo las siguientes constantes:

$$K_P = 0.075; K_I = 0,8; K_D = 0.005$$

Y los resultados presentados a continuación (*Figura 34*):

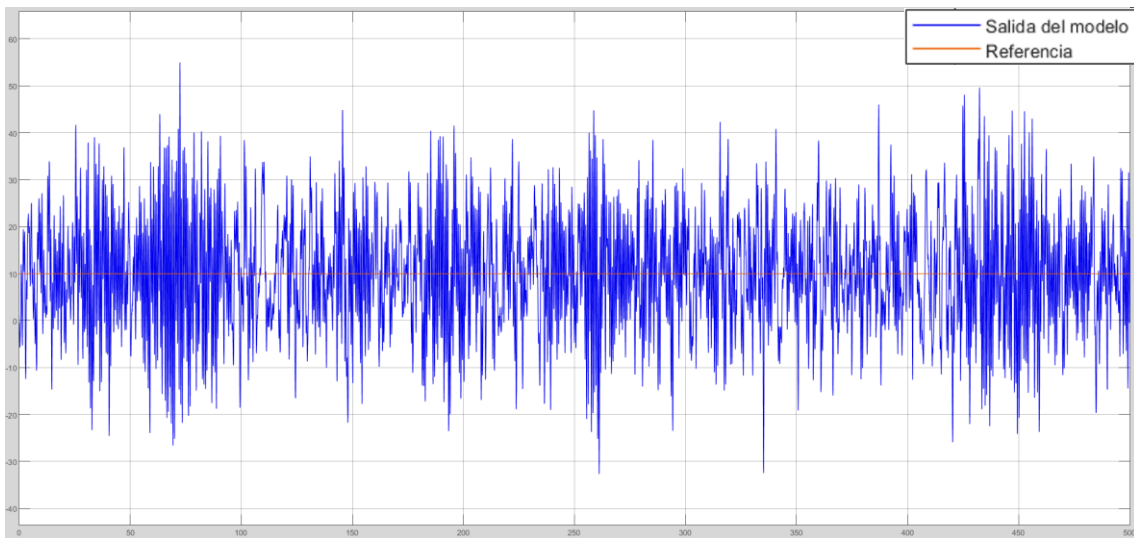


Figura 34 a)

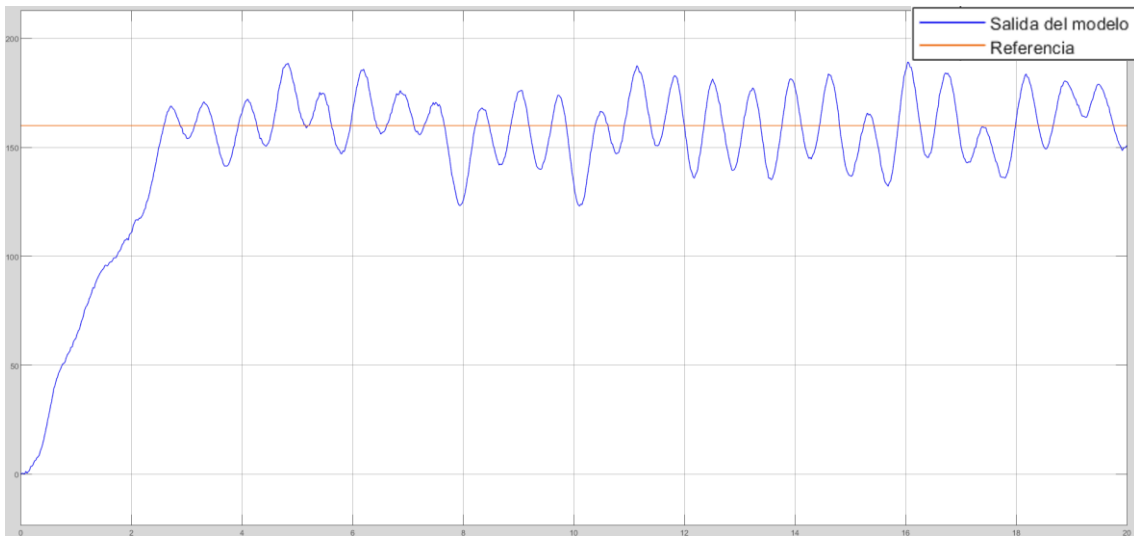


Figura 34 b)

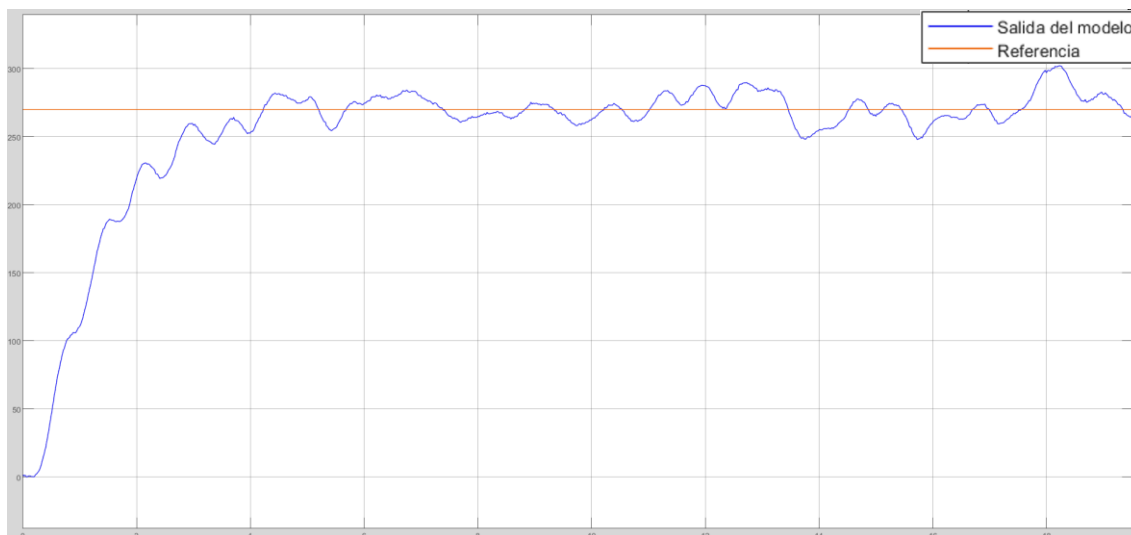


Figura 34 c)

Figura 34: a) Salida del modelo en Espacio de Estados con una referencia de 10° y Controlador PID Clásico, b) Salida del modelo en Espacio de Estados con una referencia de 160° y Controlador PID Clásico, c) Salida del modelo en Espacio de Estados con una referencia de 270° y Controlador PID Clásico

Se aprecia de nuevo que ante consignas pequeñas el modelo no trabaja bien, al igual que con Box-Jenkins, pero ello no nos impide medir parámetros de la respuesta ante otro tipo de entradas (ver *Tabla 3*):

Tabla 3: Parámetros en régimen transitorio y permanente de la salida del modelo en Espacio de Estados regulado por un Controlador PID Clásico

Escalón de referencia	de	Tiempo de subida (5%-95%)	Tiempo de establecimiento	de	Sobreimpulso máximo
10°		-	-		-
160°		2.205s	2.723s		23%
270°		2.725	-		11.851%

Repetimos el “Tuning” del PID Fuzzy de la misma forma que con Box-Jenkins, consiguiendo la configuración siguiente del mismo tras sucesivas pruebas (*Figura 35*):

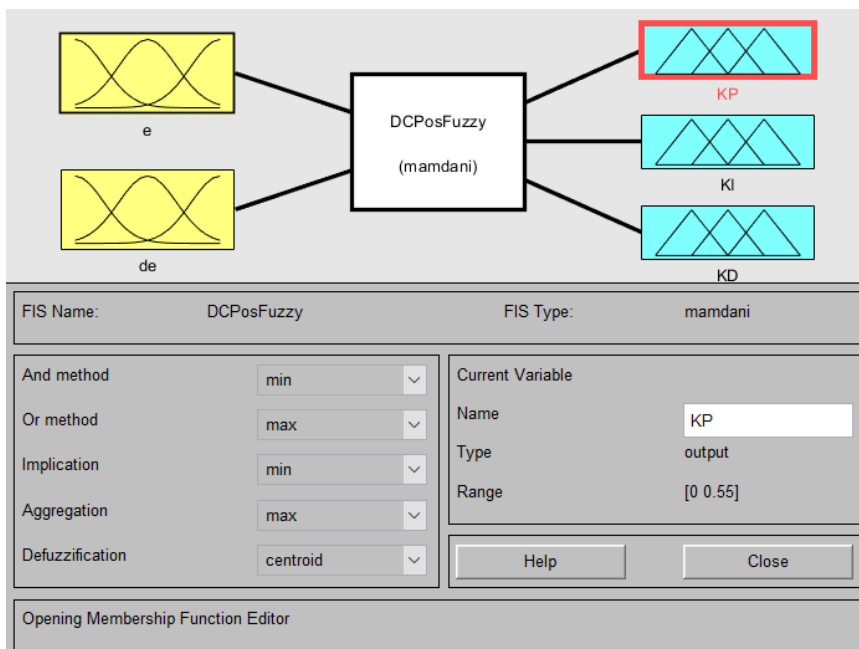


Figura 35 a)

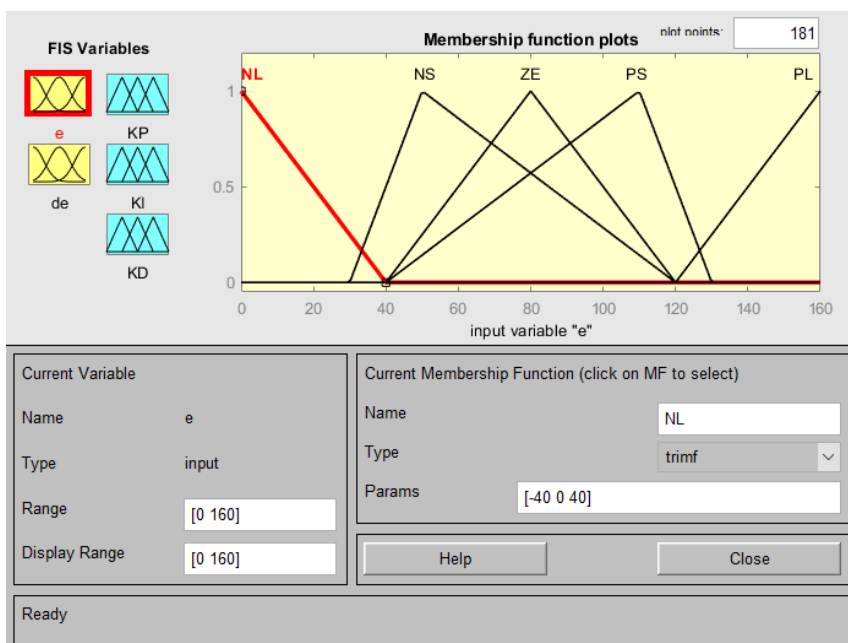


Figura 35 b)

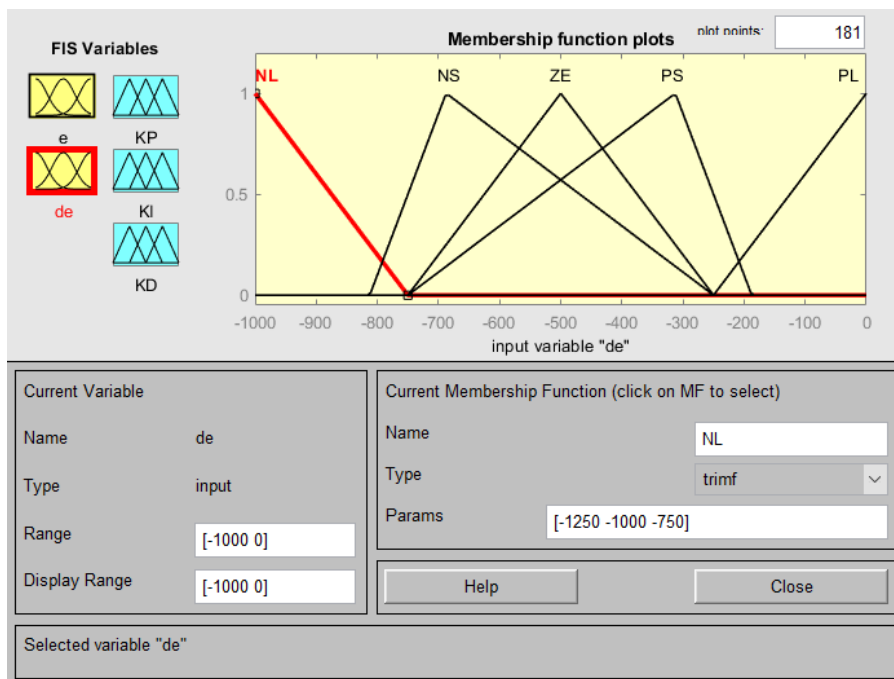


Figura 35 c)

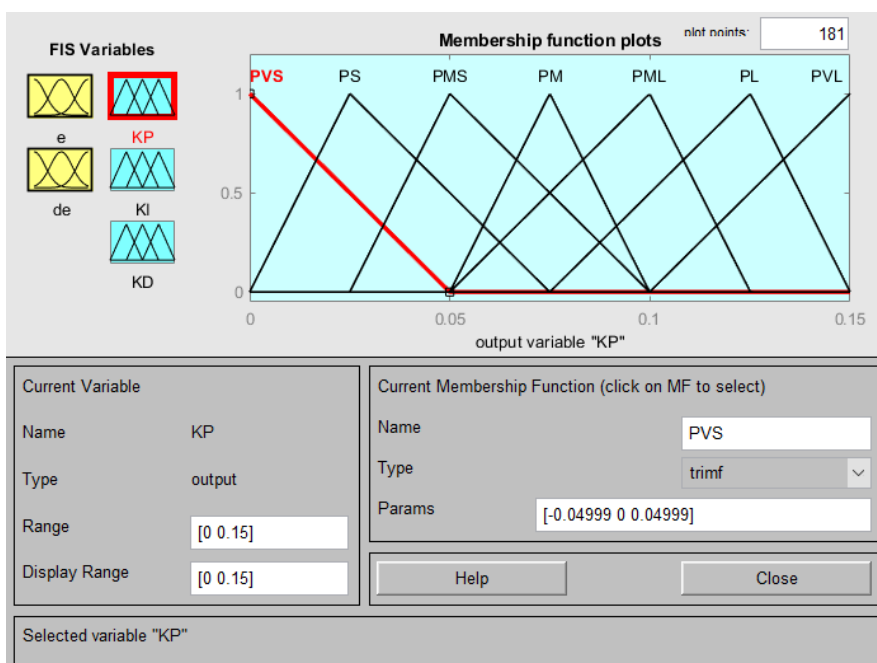


Figura 35 d)

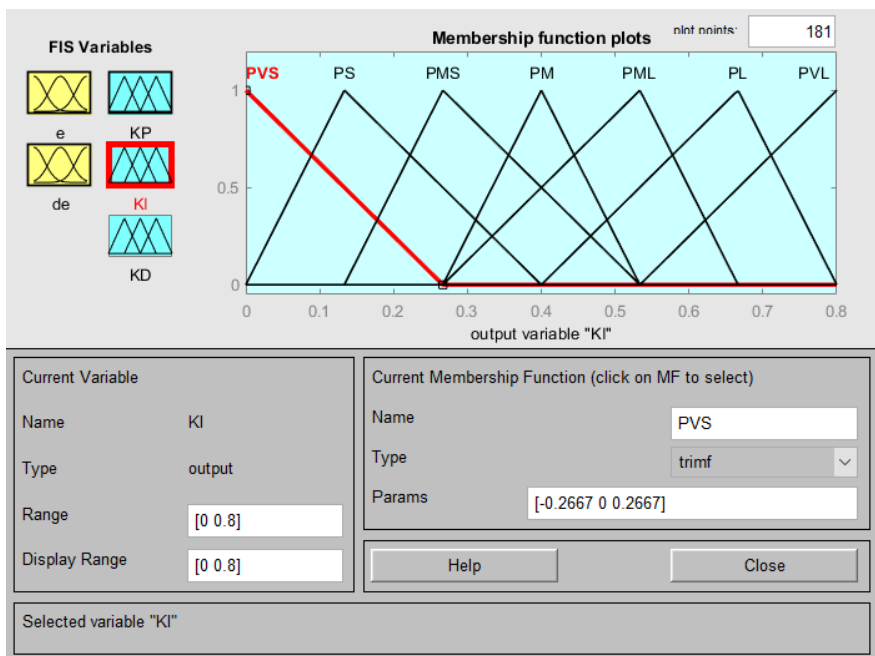


Figura 35 e)

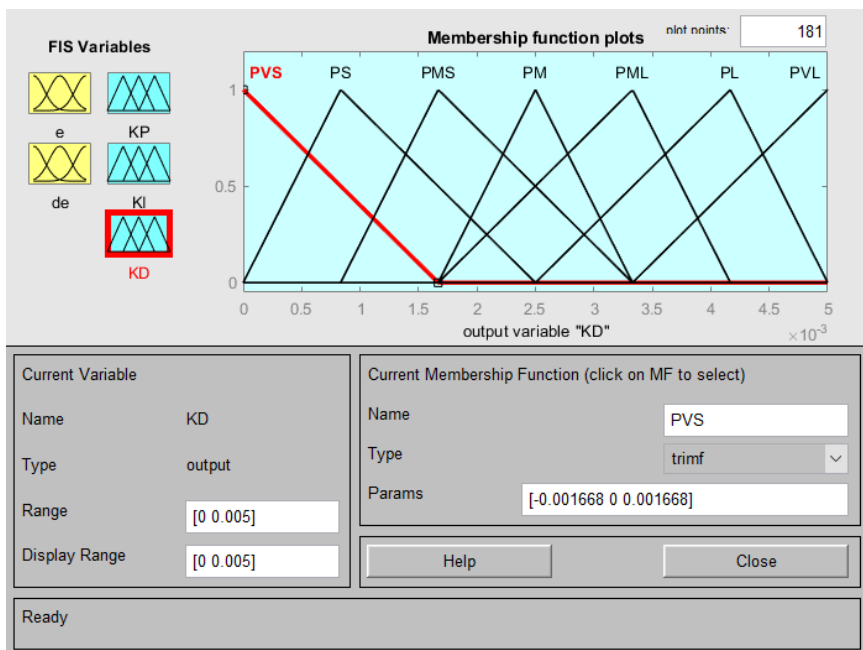


Figura 35 f)

Figura 35: a) Controlador PID Fuzzy creado a partir de los parámetros del PID Clásico aplicado al modelo de Espacio de Estados, b) Primera entrada del Controlador PID Fuzzy aplicado al modelo de Espacio de Estado: $e(t)$, c) Segunda entrada del Controlador PID Fuzzy aplicado al modelo de Espacio de Estado: $de(t)$, d) Primera salida del Controlador PID Fuzzy aplicado al modelo de Espacio de Estado: KP, e) Segunda salida del Controlador PID Fuzzy aplicado al modelo de Espacio de Estado: KI, f) Tercera salida del Controlador PID Fuzzy aplicado al modelo de Espacio de Estado: KD

Los resultados de la implantación de PID Fuzzy fueron los expuestos en la *Figura 36*:

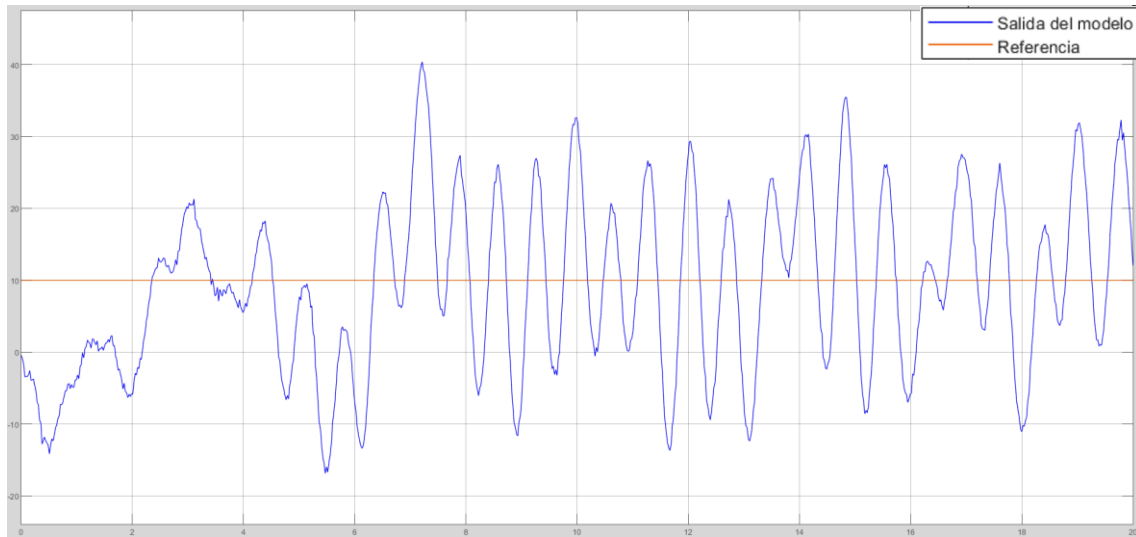


Figura 36 a)

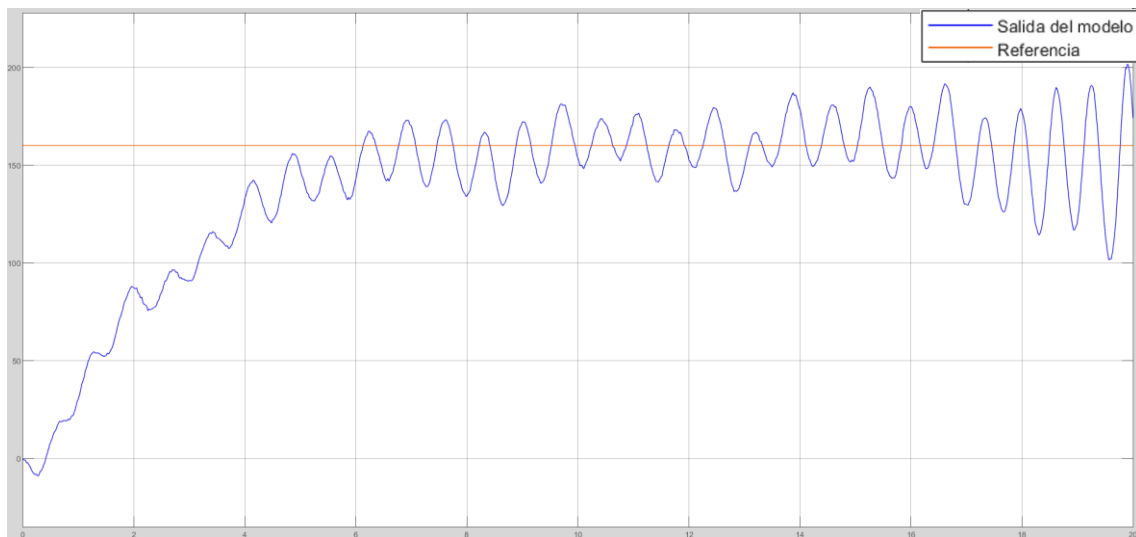


Figura 36 b)

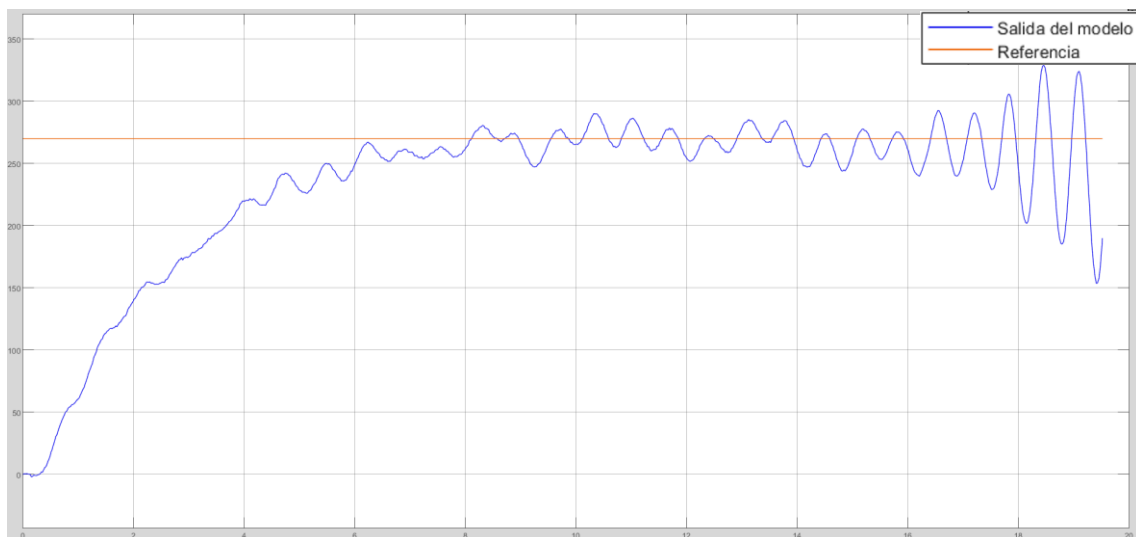


Figura 36 c)

Figura 36: a) Salida del modelo en Espacio de Estados ante una entrada de 10° con controlador PID Fuzzy, b) Salida del modelo en Espacio de Estados ante una entrada de 160° con controlador PID Fuzzy, c) Salida del modelo en Espacio de Estados ante una entrada de 270° con controlador PID Fuzzy

Aunque como es costumbre existe un cierto descontrol ante entradas pequeñas en este caso no es una situación tan acusada como con el PID Clásico. Los parámetros medidos se muestran en la *Tabla 4*:

Tabla 4: Parámetros en régimen permanente y transitorio del sistema en Espacio de Estados con Controlador PID Fuzzy ante distintas estradas

Escalón de referencia	de	Tiempo de subida (5%-95%)	Tiempo de establecimiento	Sobreimpulso máximo
10°	-	-	-	-
160°		4.291s	6.923s	25,93%
270°		5.577s	8.314s	43.037%

Para poder comparar con Box-Jenkins volvemos a someter al modelo a señales “prácticas” (que pueden acontecer en el mundo real), que son iguales a la señal de validación del modelo (*Figura 19*), y comparamos los resultados (*Figura 37*):

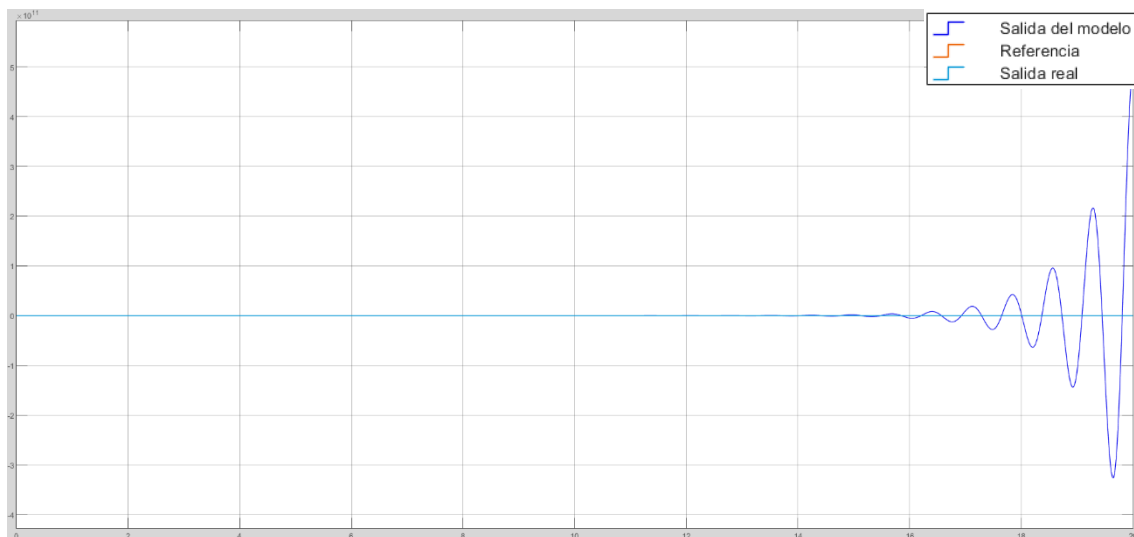


Figura 37 a)

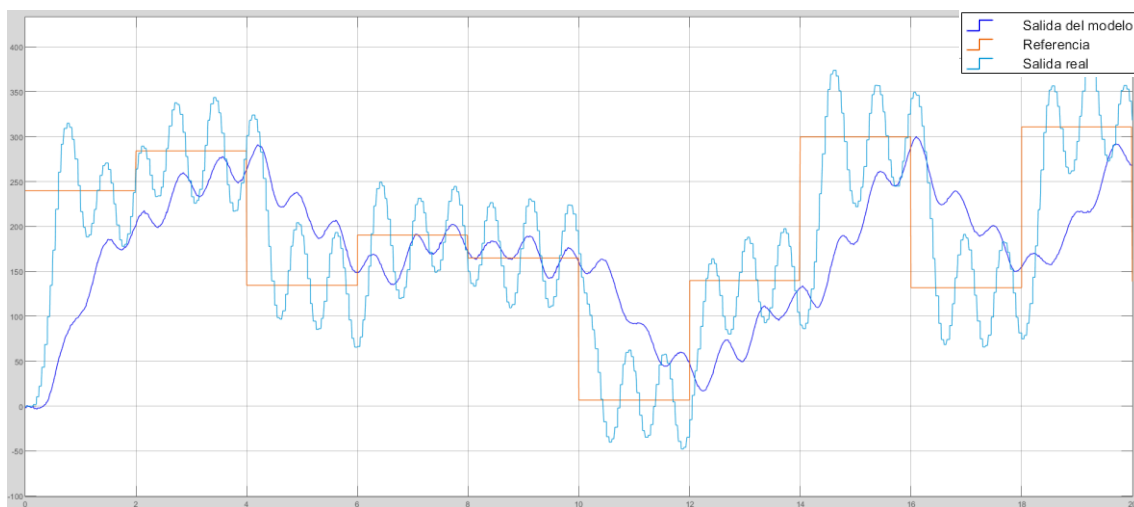


Figura 37 b)

Figura 37: a) Respuesta del modelo en Espacio de estados (azul marino) sin controlador, comparado con la salida real del Motor DC sin regular (cian) al someter a ambos a la misma referencia (naranja), b) Respuesta del modelo Box-Jenkins regulado con un PID Clásico (azul marino) comparado con la respuesta del Motor DC real también sometido a la regulación de un PID Clásico (cian) al someterlos a la misma referencia (naranja)

No ha sido posible obtener respuesta del sistema con PID Fuzzy debido a un error de cálculo en el modelo de Espacio de Estados, por lo que nos es imposible realizar una comparativa. Aun así, podemos extraer la conclusión de que si no utilizamos ningún tipo de control el modelo se desestabiliza completamente, y que es por ello necesario, al menos, el PID Clásico.

3.2. Modelo de velocidad

Para la implementación del modelo de velocidad se introdujeron también señales aleatorias e incorreladas en la arquitectura montada en Simulink que se describe en la *Figura 38*:

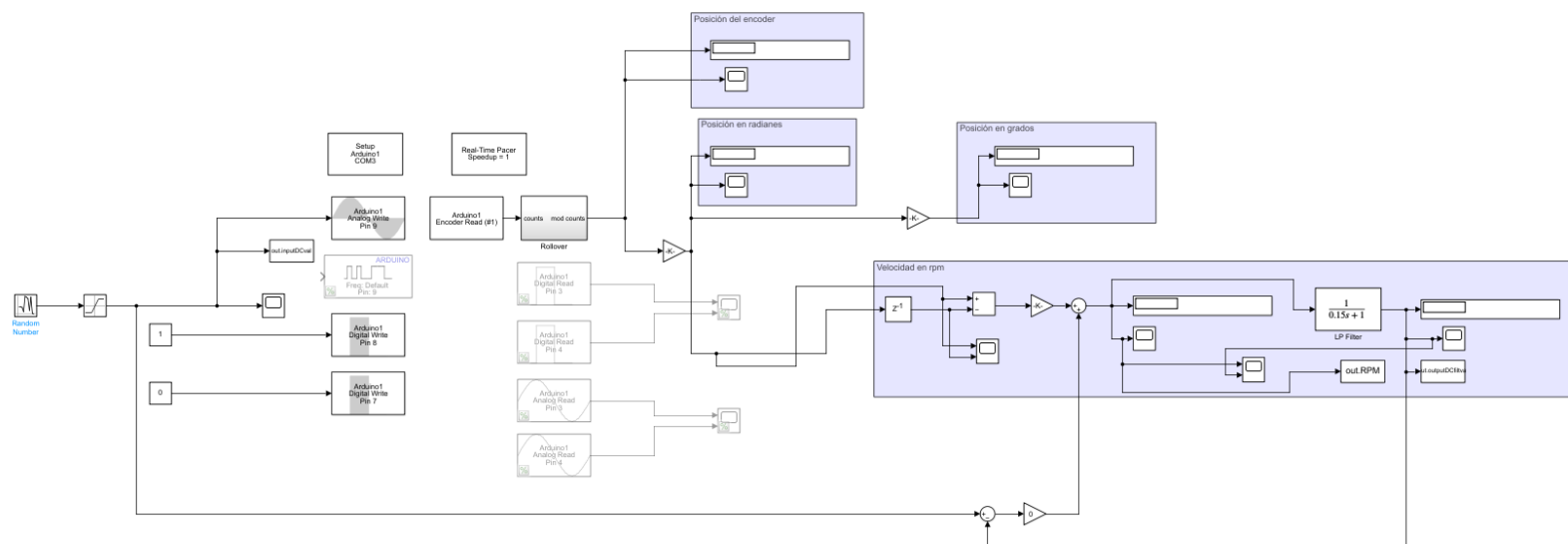


Figura 38: Estructura del modelo de velocidad montado en Simulink

Esta señal de entrada está comprendida entre 0 y 255, que son los valores en código binario que podemos escribir en el pin 9 de nuestra Arduino UNO (Analog Write Pin 9) para variar el ciclo de trabajo de la señal (desde el 0% hasta el 100%, asociados, respectivamente, a un valor de “0” y de “255” en el pin) que va a parar a la entrada de “Enable” del puente en H. Coincidentemente los márgenes máximo y mínimo de los valores que podemos escribir en este pin (“0” a “255”) es igual a las velocidades mínima y máxima del motor en rpm (0 rpm y 255 rpm, respectivamente). De esta forma se hizo correr al motor en una misma dirección todo el tiempo (“1” lógico, ó 5,5V, en Digital Write Pin 8, y “0” lógico, ó 0V, en Digital Write Pin 7) y se midió la salida del sistema para ver si coincidía con la consigna introducida. De esta manera se generaron las siguientes señales de entrenamiento (*Figura 39*) y validación (*Figura 40*):

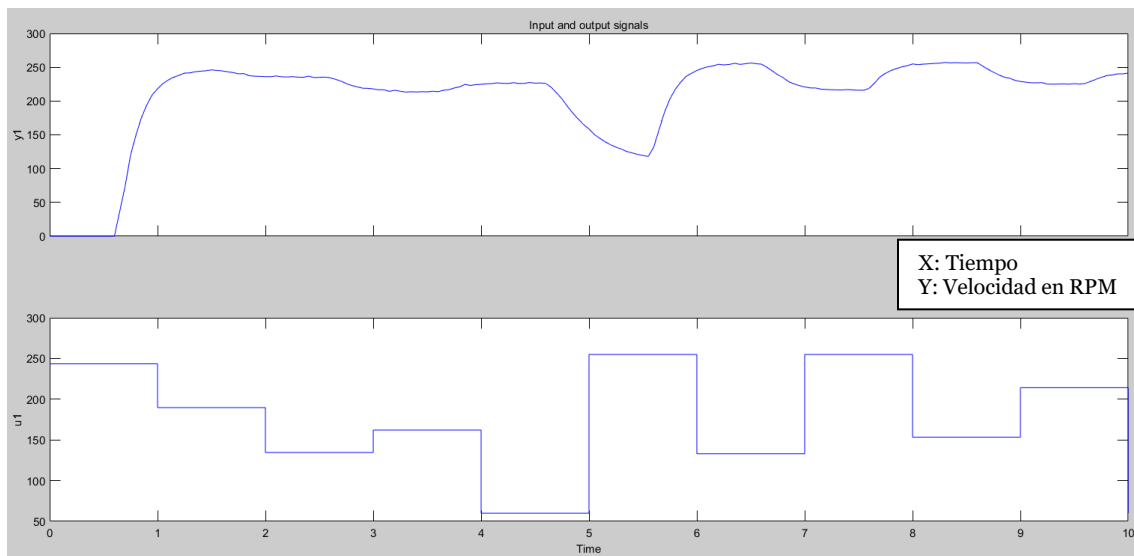


Figura 39: Datos de entrenamiento para el modelo de velocidad (gráfica superior: output, gráfica inferior: input)

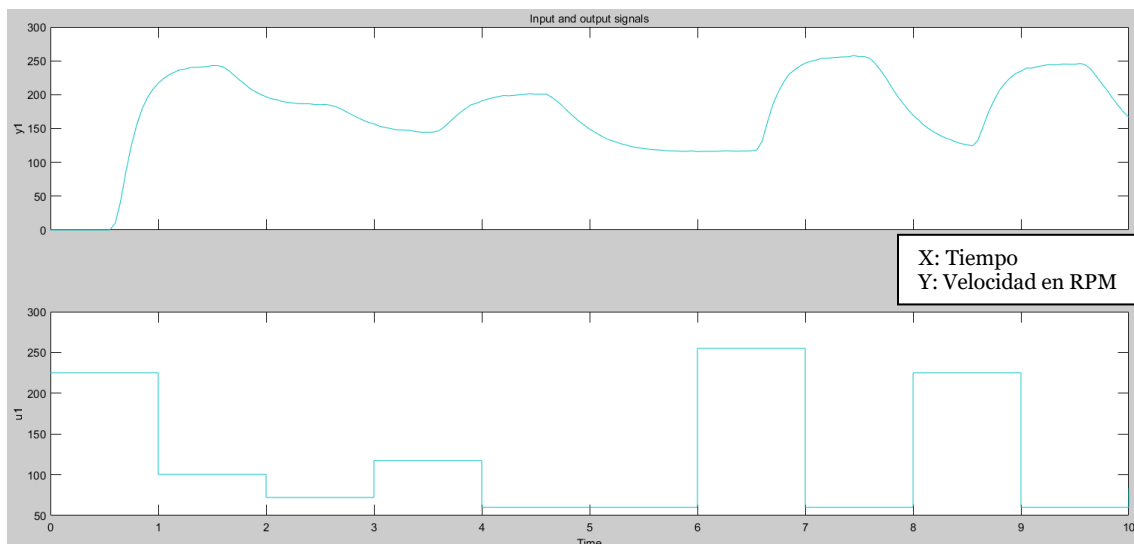


Figura 40: Datos de validación para el modelo de velocidad (gráfica superior: output, gráfica inferior: input)

Estas se emplearon de nuevo para generar modelos basados en Box-Jenkins y Espacio de estados, cuyo desempeño se mostrará en apartados posteriores.

3.2.1. Modelo de velocidad Box-Jenkins

Comenzamos el estudio del comportamiento del Motor DC con respecto a su velocidad creando un modelo de Box-Jenkins para emular su comportamiento. A priori podríamos llegar a deducir que puesto que Box-Jenkins es un sistema estadístico basado en la media y la varianza del sistema, y sujeto a su estacionalidad y estática (14), las oscilaciones que presente nos podrían hacer plantearnos optar por otro modelo, es decir, que es probable que la alternativa en Espacio de Estados sea una opción más sensata de cara a simular la respuesta del motor. Una vez expuestas estas conjeturas procedemos a exponer el modelo creado tras sucesivas pruebas con distintos valores de los parámetros del mismo, hasta obtener un ajuste lo más fiel posible a la realidad. El comportamiento del modelo creado se muestra en la *Figura 41 y 42*:

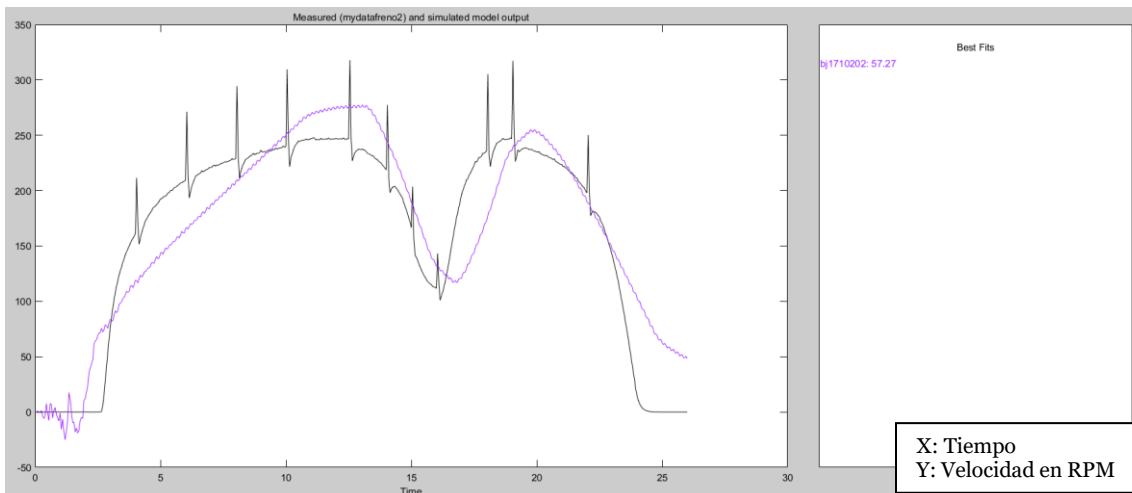


Figura 41: Comparación entre la salida real del motor (negro) y la ofrecida por el modelo de velocidad Box-Jenkins (morado) una vez ajustados sus parámetros para obtener la respuesta más óptima posible

Discrete-time BJ model: $y(t) = [B(z)/F(z)]u(t) + [C(z)/D(z)]e(t)$

$$B(z) = 0.06283 z^{-2}$$

$$C(z) = 1 + 0.1932 z^{-1} - 0.2331 z^{-2} - 0.3214 z^{-3} - 0.2713 z^{-4} - 0.2092 z^{-5} - 0.0965 z^{-6} + 0.002419 z^{-7}$$

$$D(z) = 1 - 1.388 z^{-1} - 0.02434 z^{-2} + 0.2274 z^{-3} + 0.2332 z^{-4} + 0.1038 z^{-5} - 0.279 z^{-6} + 0.2947 z^{-7} - 0.2326 z^{-8} + 0.09856 z^{-9} - 0.03334 z^{-10}$$

$$F(z) = 1 - 1.466 z^{-1} + 0.966 z^{-2} - 0.5806 z^{-3} - 0.0141 z^{-4} + 0.5273 z^{-5} - 0.6094 z^{-6} + 0.416 z^{-7} - 0.1984 z^{-8} + 0.01019 z^{-9} + 0.116 z^{-10} - 0.1036 z^{-11} + 0.03633 z^{-12} + 0.02268 z^{-13} - 0.1163 z^{-14} + 0.3468 z^{-15} - 0.486 z^{-16} + 0.2044 z^{-17} + 0.2728 z^{-18} - 0.4262 z^{-19} + 0.154 z^{-20}$$

Figura 42: Representación matemática del modelo Box-Jenkins de velocidad

De nuevo se realizaron pruebas introduciendo señales escalón en el sistema, de 50rpm, 100rpm y 200rpm, para poder medir sus parámetros en régimen transitorio y posteriormente ajustar los valores de las constantes del Controlador PID Clásico a partir del método manual de Box-Jenkins. Primero se puso a prueba el modelo sin control alguno. El esquema Simulink utilizado para estas pruebas se muestra en la Figura 43:

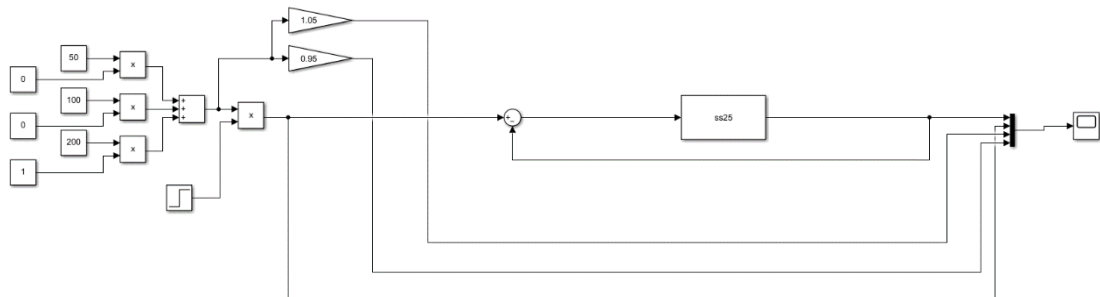


Figura 43: Estructura del modelo Box-Jenkins de velocidad con realimentación negativa (bloque "ss25") en Simulink

Se obtuvieron los siguientes resultados (Figura 44):

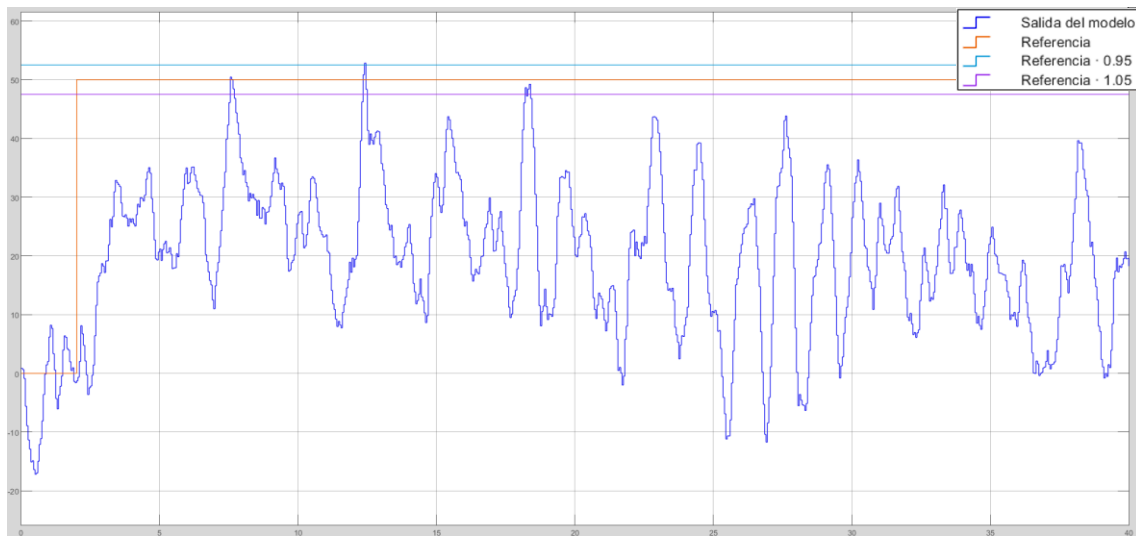


Figura 44 a)

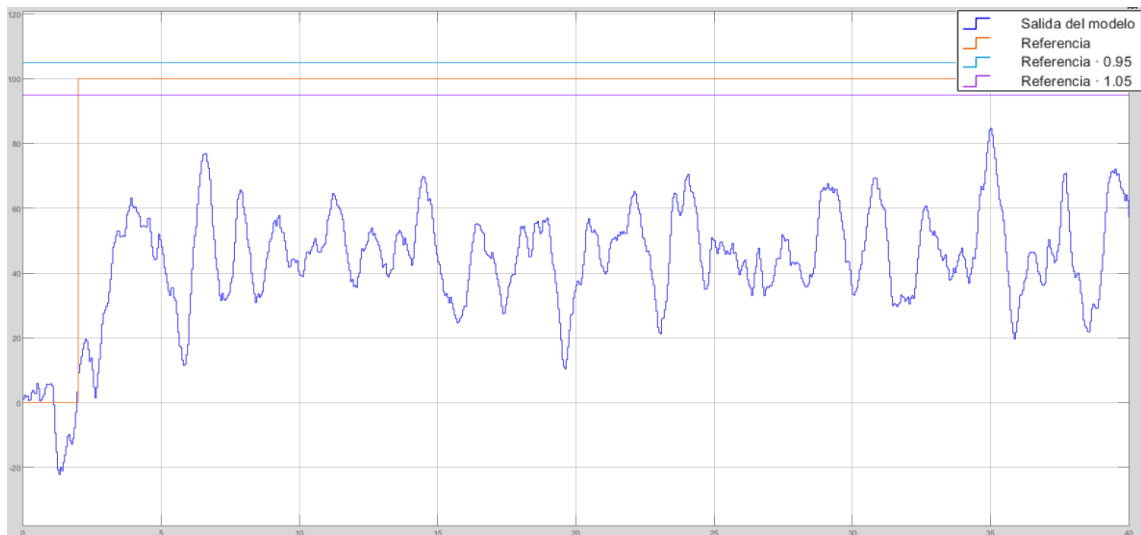


Figura 44 b)

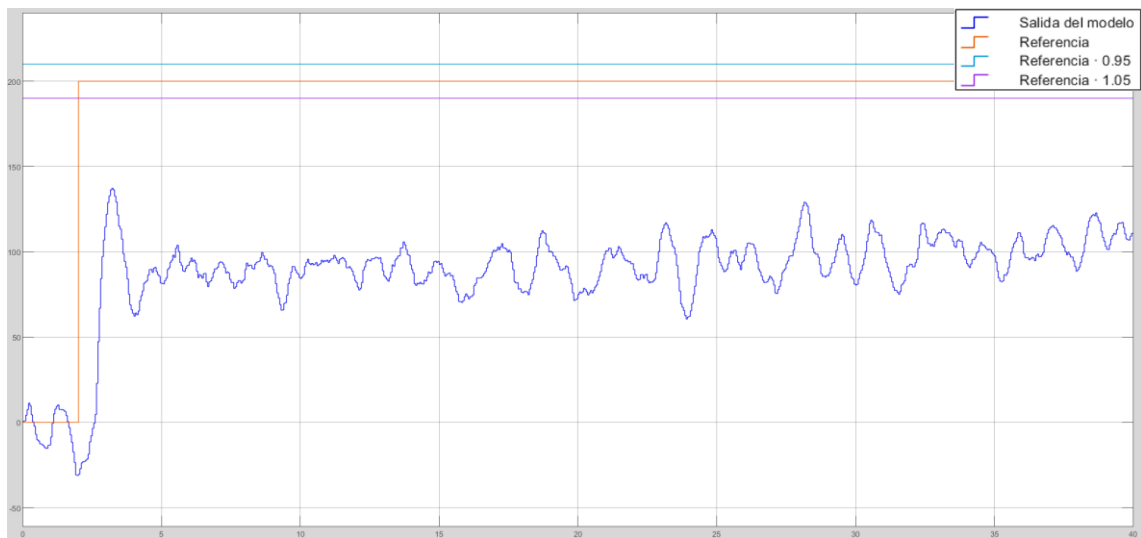


Figura 44 c)

Figura 44: a) Respuesta del modelo Box-Jenkins de velocidad ante un escalón de 50rpm y sin controlador, b) Respuesta del modelo Box-Jenkins de velocidad ante un escalón de 100rpm y sin controlador, c) Respuesta del modelo Box-Jenkins de velocidad ante un escalón de 200rpm y sin controlador

Dado que, a diferencia de los modelos de posición, el Box-Jenkins de velocidad sí es estable a pesar de no contar con un Controlador PID de ningún tipo, en esta ocasión pudieron medirse los parámetros en régimen permanente y transitorio del modelo, adquiriéndose así información valiosa que podrá usarse para ser comparada con la respuesta del sistema al implementar regulación de algún tipo (Ver Tabla 5):

Tabla 5: Parámetros en régimen transitorio y permanente del modelo Box-Jenkins de velocidad sin regulación

Escalón de referencia	Tiempo de subida (5% - 95%)	Tiempo de establecimiento	Sobreimpulso máximo	Error en régimen permanente(ess)
50rpm	2,462s	-	164,1%	≈20rpm
100rpm	2,966s	-	69,32%	≈50rpm
200rpm	0,214s	-	37,2%	≈100rpm

Se aprecia que el sistema sufre un gran error en régimen permanente (Figura 45) (lo cual se palia con el parámetro KI de nuestro PID Clásico, así como KD), y una oscilación acuciante, de manera que no llega a alcanzar un régimen permanente como tal. Esto da buena cuenta de la certeza de que, tal y como se dijo, la presencia de un controlador es necesaria para estabilizar el modelo. Una vez realizadas las pruebas en régimen transitorio del modelo sin controlador procedemos a diseñar el PID Clásico volviendo a emplear el “tunning” manual de Ziegler-Nichols (Figura 45):

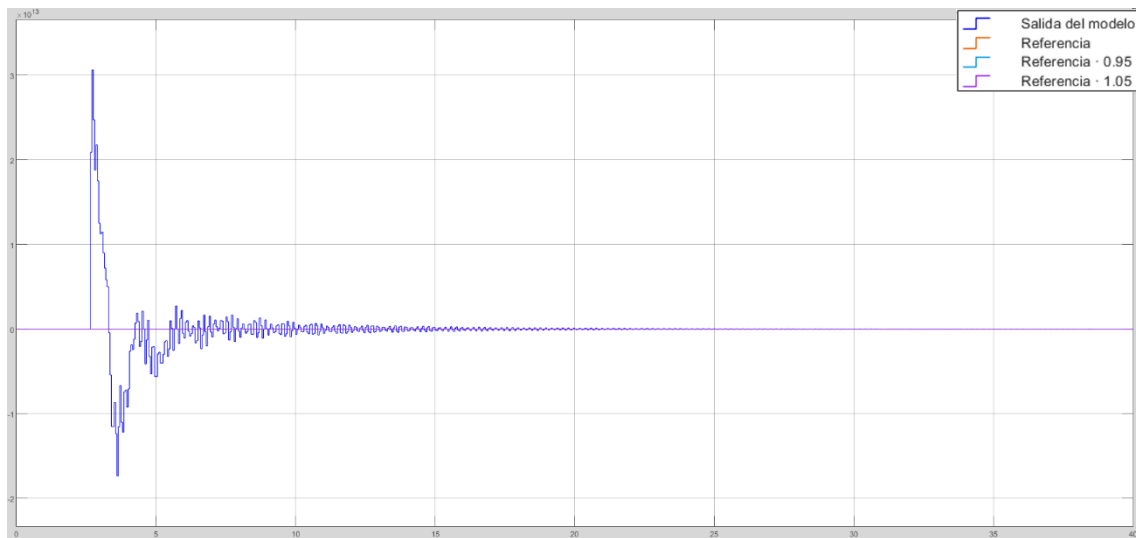


Figura 45 a)

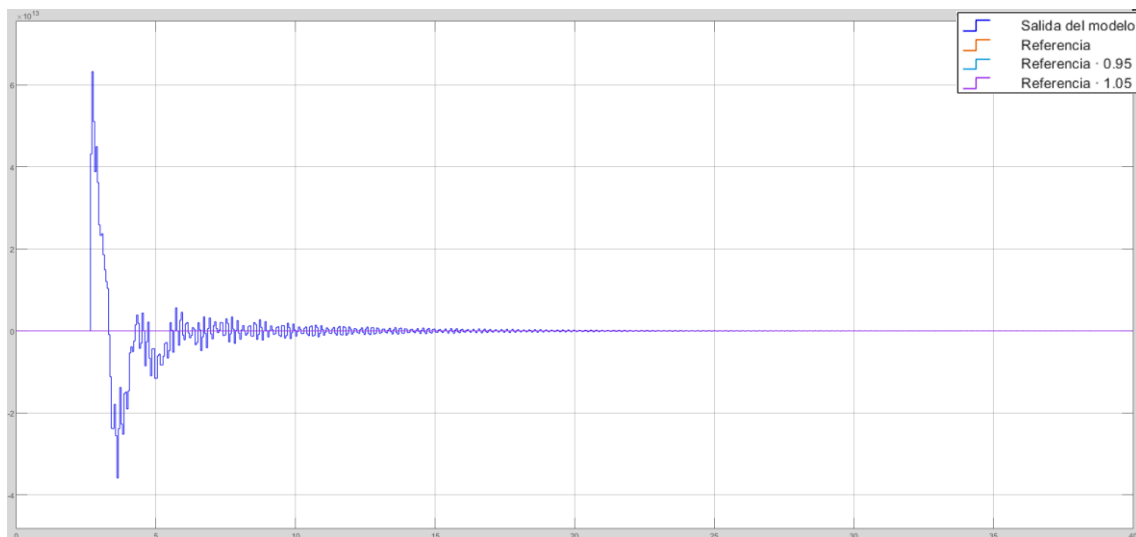


Figura 45 b)

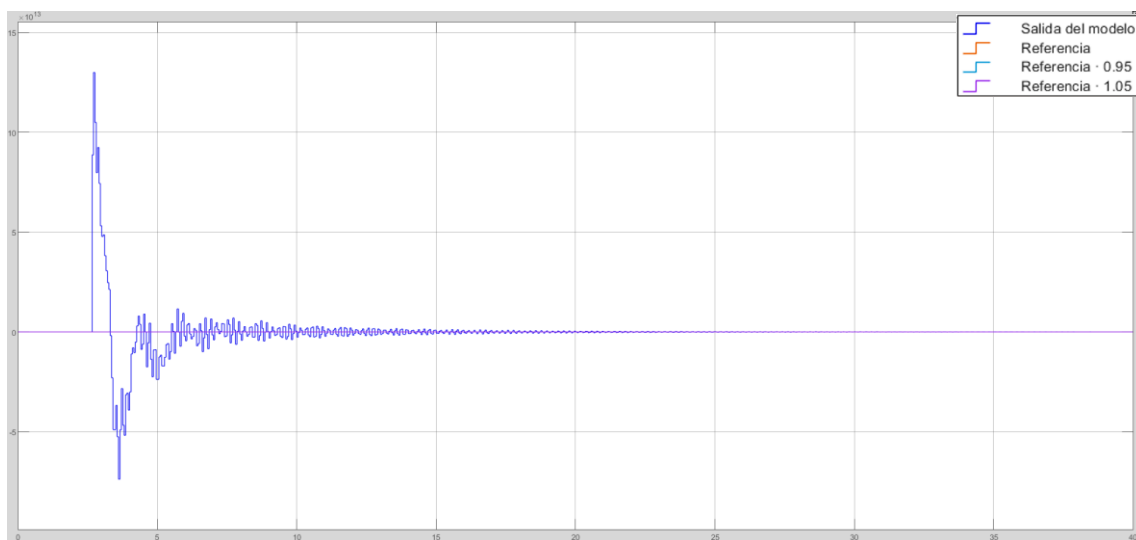


Figura 45 c)

Figura 45: a) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 50rpm con PID Clásico, b) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 100rpm con PID Clásico, c) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 200rpm con PID Clásico

A pesar de las numerosas pruebas de sintonización manual realizadas el modelo en simulación se volvía irremediabilmente inestable ante una señal de entrada tipo escalón. Dado que no se pudo identificar el motivo de este comportamiento se decidió optar por la introducción en el modelo de otro tipo de señal para hacer la selección de parámetros óptimos, consistente en la aceleración progresiva del mismo y el frenazo brusco del motor, una señal que, de nuevo, puede considerarse “práctica” (Figura 46):

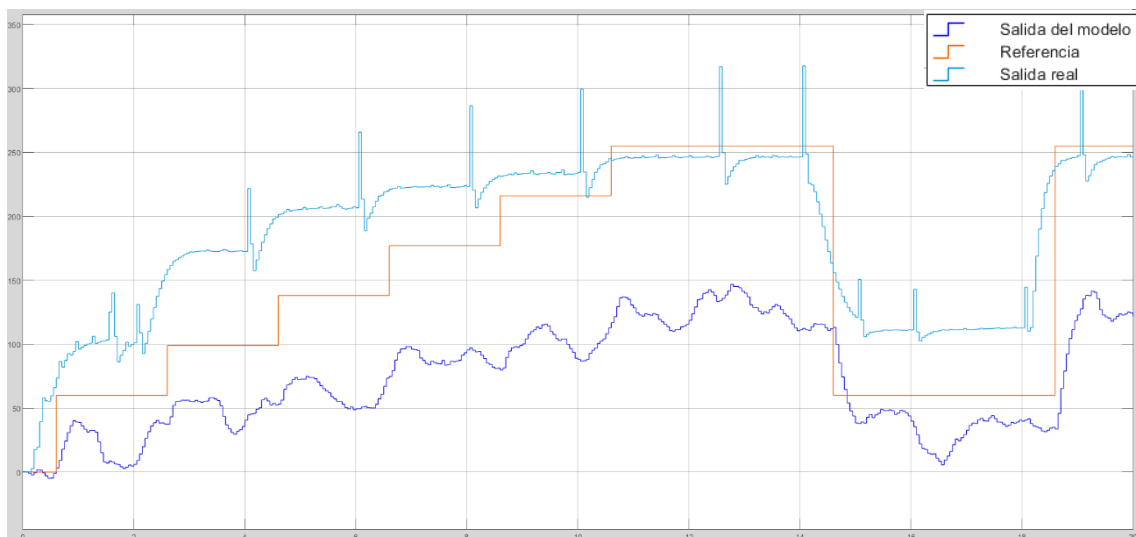


Figura 46 a)

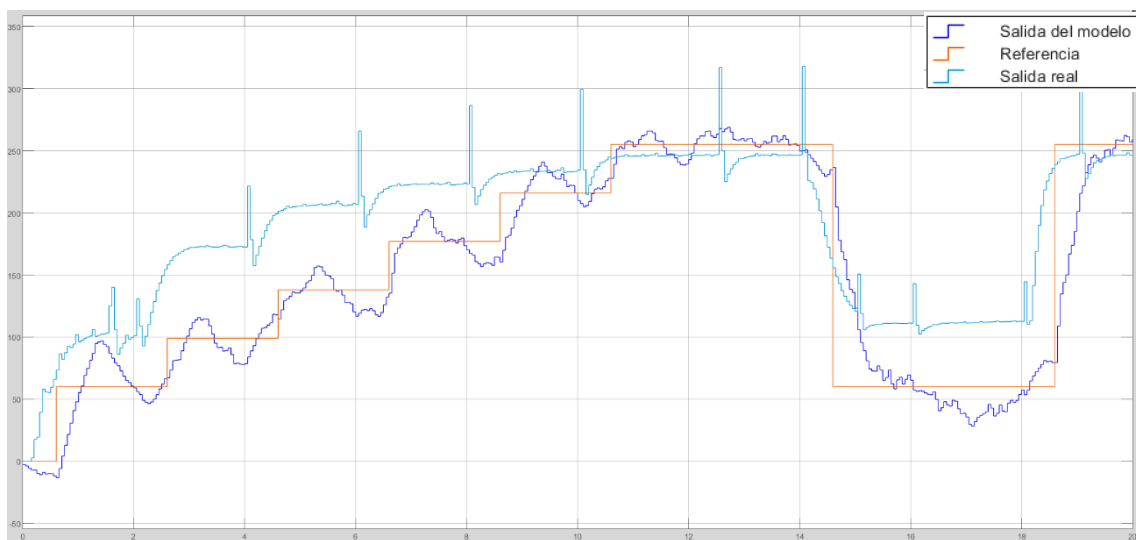


Figura 46 b)

Figura 46: a) Salida del modelo Box-Jenkins de velocidad (azul marino) y del sistema real (cian), ambos sin controlador, ante una misma referencia (naranja), b) Salida del modelo Box-Jenkins de velocidad (azul marino) y del sistema real (cian), ambos con Control PID Clásico, ante una misma referencia (naranja)

Una vez ajustadas las constantes, arrojaron los siguientes valores:

$$KP = 1,155; KI = 1.5; KD = 0.1$$

En base a estos resultados se sintonizó el Controlador PID Fuzzy, modificando de nuevo el rango de las funciones de transferencia de la salida para que sus valores intermedios oscilaran en torno a los de las constantes KP, KI y KD que acaban de

exponerse. Contamos con el esquema del modelo en la *Figura 47* y con el PID Fuzzy diseñado en la *Figura 48*:

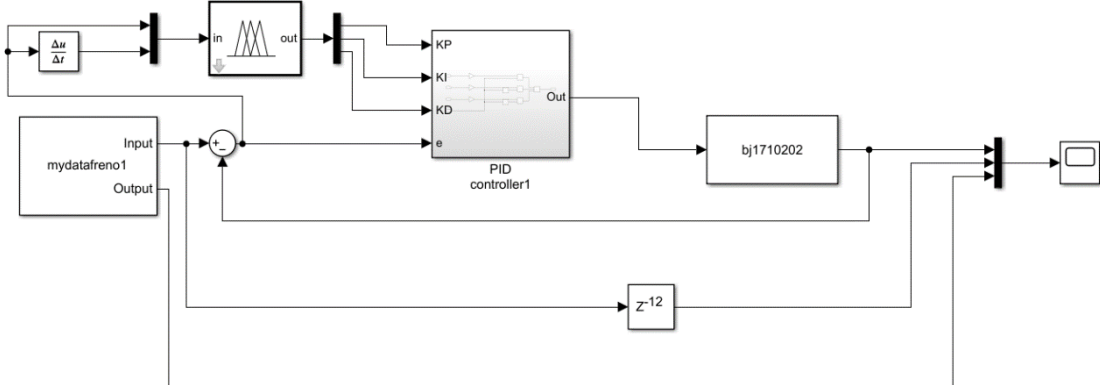


Figura 47: Esquema del modelo Box-Jenkins de velocidad regulado por un Controlador PID Fuzzy

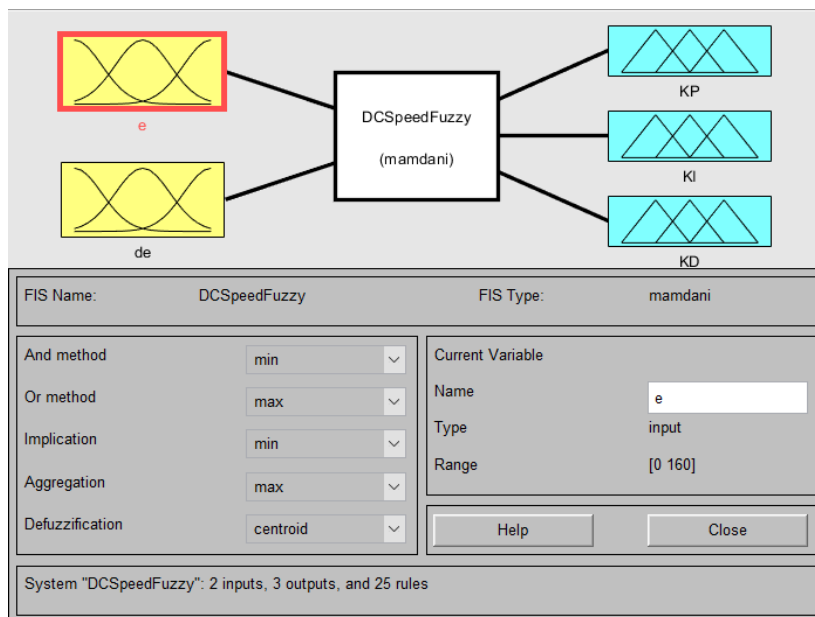


Figura 48 a)

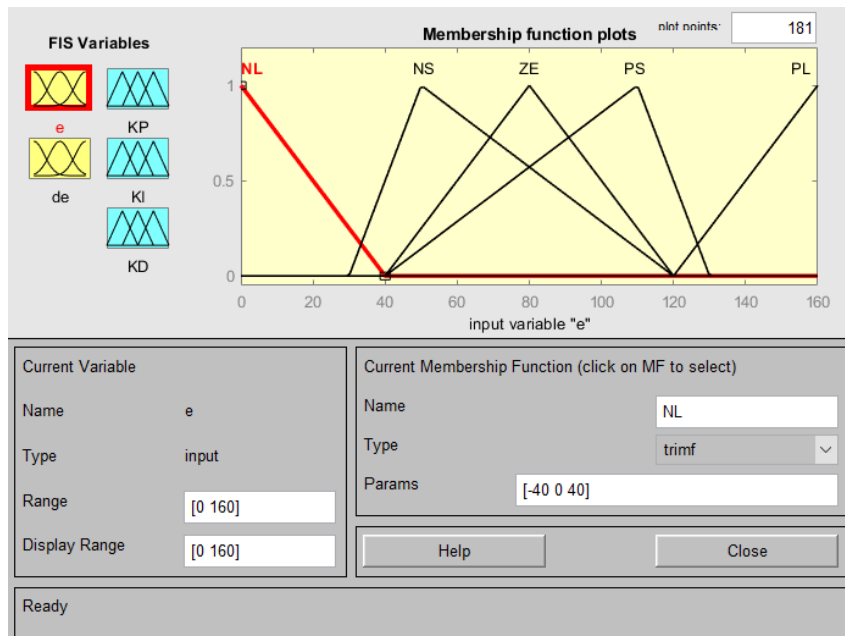


Figura 48 b)

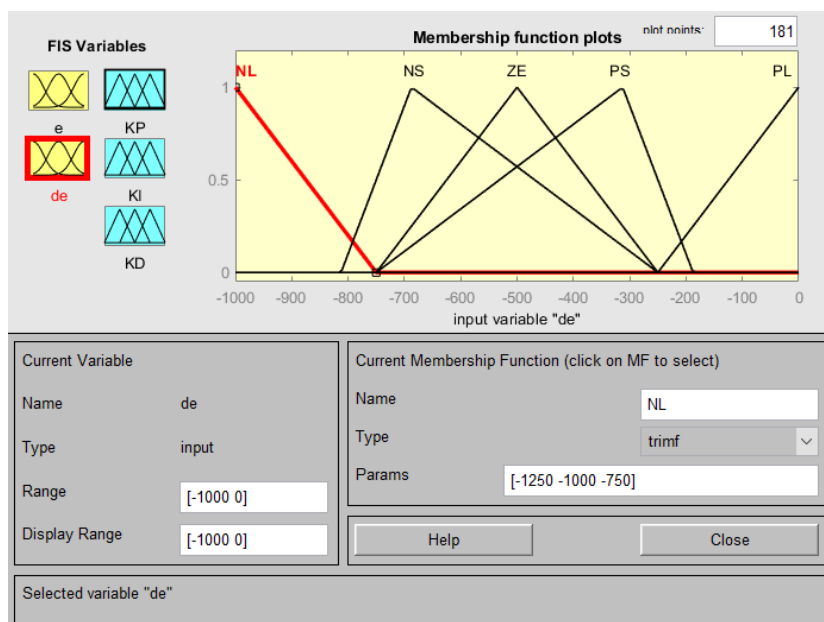


Figura 48 c)

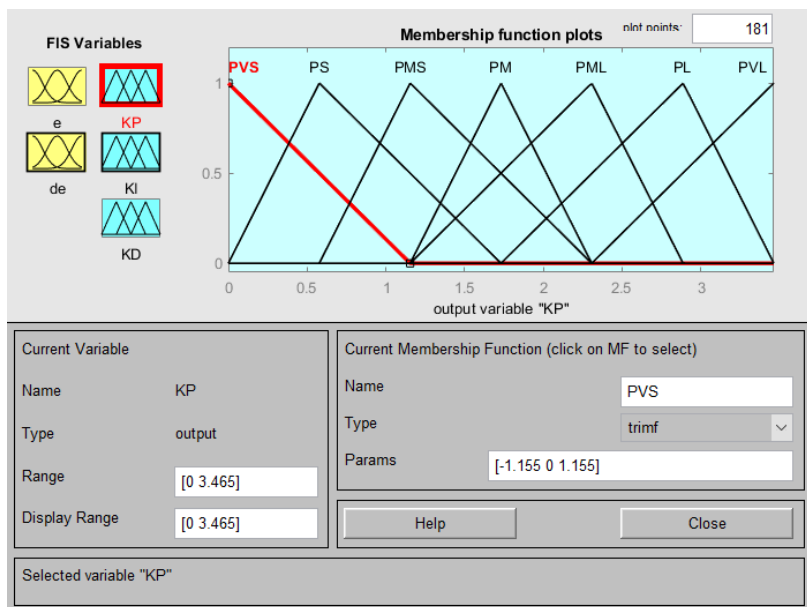


Figura 48 d)

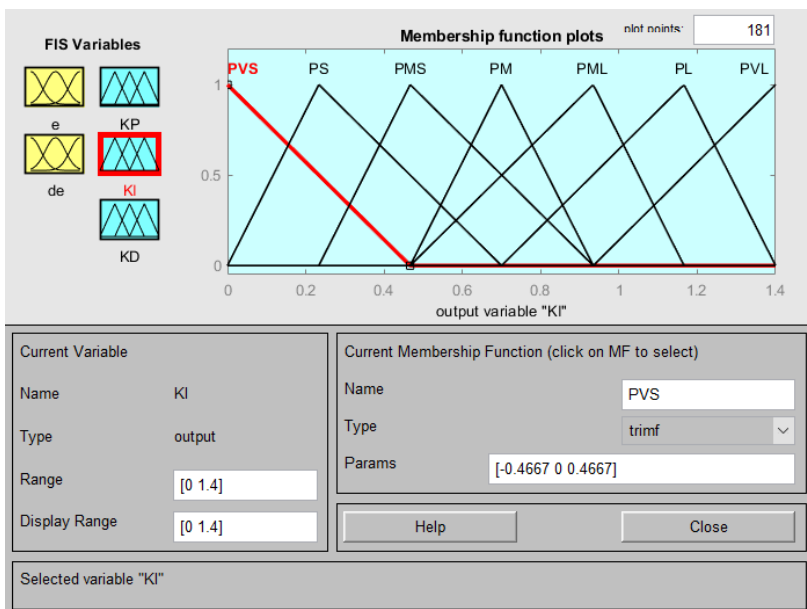


Figura 48 e)

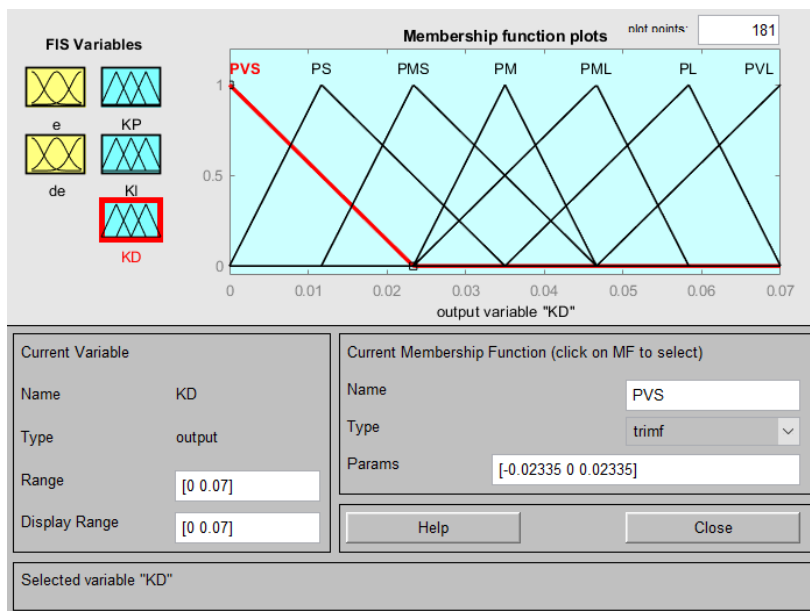


Figura 48 f)

Figura 48: a) Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad, b) Primera entrada del Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad: $e(t)$, c) Segunda entrada del Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad: $de(t)$, d) Primera salida del Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad: KP, e) Segunda salida del Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad: KI, f) Tercera salida del Controlador PID borroso de 2 entradas y 3 salidas diseñado para el modelo Box-Jenkins de velocidad: KD

Se consiguieron lo siguientes resultados ante las sucesivas entradas tipo escalón (Figura 49):

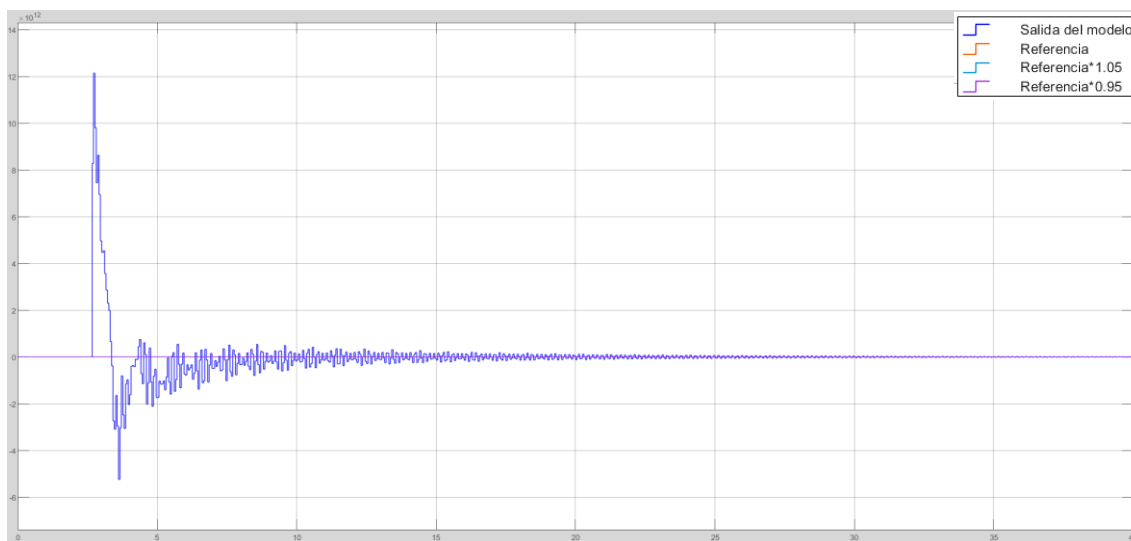


Figura 49 a)

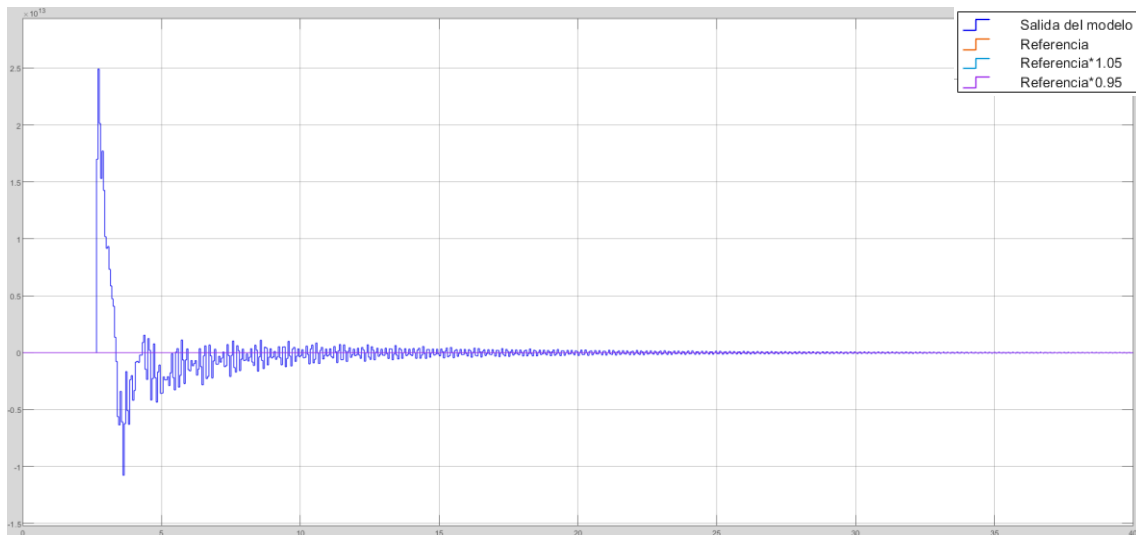


Figura 49 b)

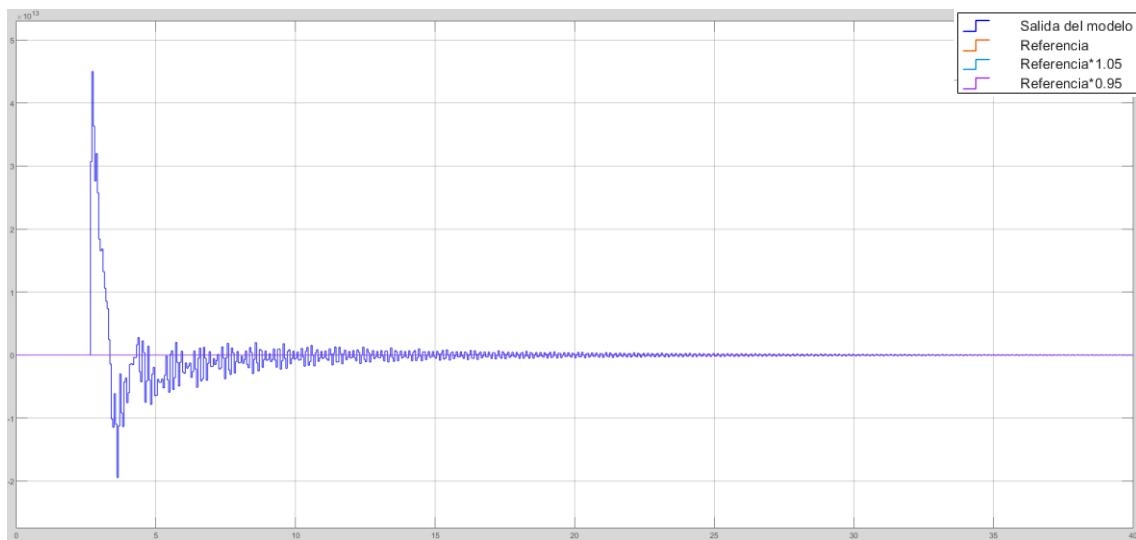


Figura 49 c)

Figura 49: a) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 50rpm con PID Fuzzy, b) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 100rpm con PID Fuzzy, c) Respuesta del modelo Box-Jenkins de velocidad a un escalón de 200rpm con PID Fuzzy

Los resultados vuelven a ser irremediabilmente dispares con la realidad y descontrolados, por lo que optamos de nuevo por introducir la misma señal empleada para el “tuning” del PID Clásico en el modelo regulado con PID Fuzzy, obteniendo (Figura 50):

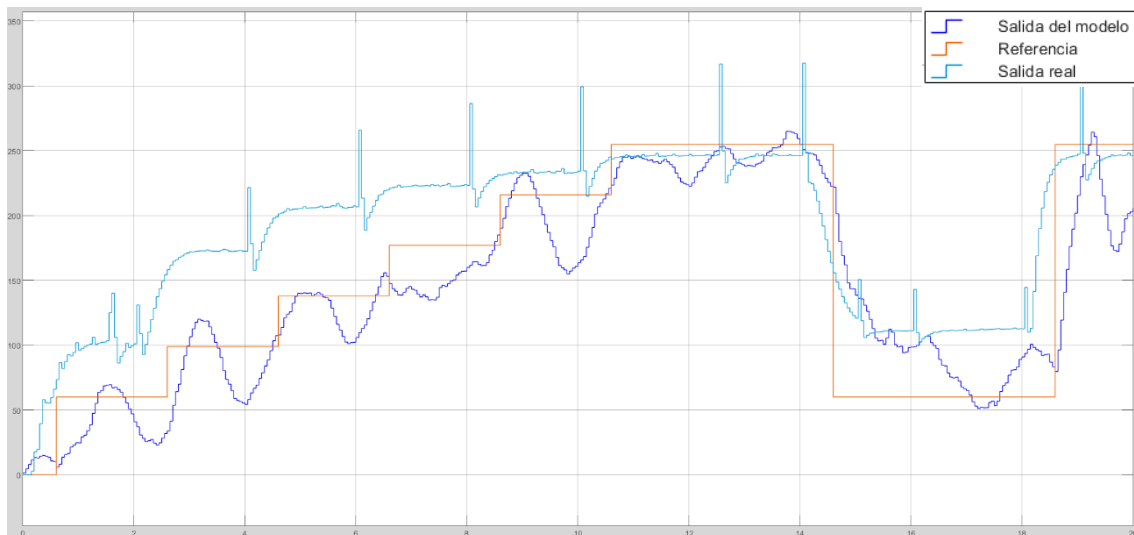


Figura 50: Salida del modelo Box-Jenkins de velocidad (azul marino) y del sistema real (cian), ambos con Control PID Fuzzy, ante una misma referencia (naranja)

Las oscilaciones e inconsistencia de los resultados nos obligan a abogar por otro tipo de modelo de velocidad, que será el de Espacio de Estados, si bien no por ello las conclusiones obtenidas son menos interesantes: resulta curioso ver cómo, con el empleo de un PID Clásico, el modelo tarda menos en alcanzar la consigna, pero la precisión se ve mermada y las oscilaciones más pronunciadas, que con el PID Fuzzy. Podemos deducir, a pesar de las complicaciones asociadas a Box-Jenkins, que la acción dinámica del PID Fuzzy aporta un matiz de precisión y estabilidad que el PID Clásico simplemente no puede ofrecer, por lo que lo imponemos ante este.

3.2.2. Modelo de velocidad en Espacio de Estados

Se creó un modelo en Espacio de Estados que fuera capaz de imitar con mayor fidelidad el Motor DC que el Box-Jenkins, obteniéndose el siguiente resultado (*Figura 51*):

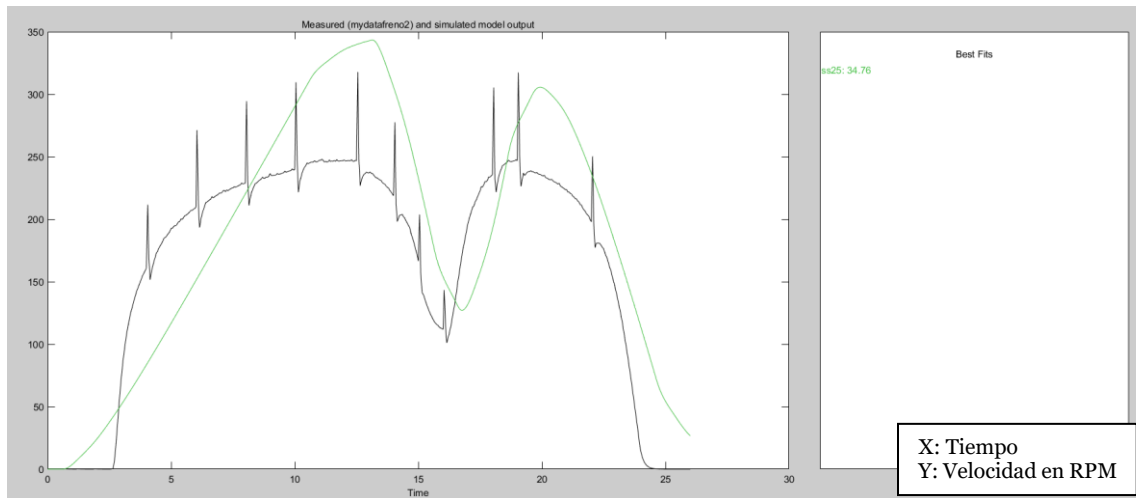


Figura 51: Representación de la salida del modelo de velocidad en Espacio de Estados (verde) frente a la respuesta del motor real (negro) después de ajustar los parámetros del primero para que arrojará la respuesta más óptima posible

```

Continuous-time identified state-space model:
dx/dt = A x(t) + B u(t) + K e(t)
y(t) = C x(t) + D u(t) + e(t)
A =
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10
x1 -8.308   60.32      0      0      0      0      0      0      0      0
x2 -60.32  -8.308      0      0      0      0      0      0      0      0
x3  0        0     -1.441   41.75      0      0      0      0      0      0
x4  0        0     -41.75  -1.441      0      0      0      0      0      0
x5  0        0      0      0      -3.723  15.57      0      0      0      0
x6  0        0      0      0     -15.57  -3.723      0      0      0      0
x7  0        0      0      0      0      0     -9.603      0      0      0
x8  0        0      0      0      0      0      0    -0.7254      0      0
x9  0        0      0      0      0      0      0      0     -1.68    5.101
x10 0        0      0      0      0      0      0      0     -5.101  -1.68
B =
      u1
x1  1.792
x2 -1.32
x3 -0.04864
x4  0.003866
x5  0.07555
x6  0.0448
x7  0.03401
x8 -0.001904
x9 -0.003852
x10 0.02858
C =
      x1      x2      x3      x4      x5      x6      x7      x8      x9      x10
y1 0.9921  0.1925  -3.602  -0.623  -3.173  -8.128  -10  -428  41.72  36.71
D =
      u1
y1 0
K =
      y1
x1  2.755
x2 -0.5091
x3 -0.03825
x4  0.1688
x5  0.02889
x6  0.2078
x7  0.1391
x8 -0.09127
x9 -0.4117
x10 0.1248
Input delays (seconds): 0.55

```

Figura 52: Representación matemática del modelo de velocidad en Espacio de Estados

Si bien el ajuste a la señal real de salida es bastante bajo (34,76%), el modelo es capaz de imitar la tendencia de la salida real con relativa fidelidad y no presenta apenas oscilaciones en la respuesta en régimen permanente, por lo que podemos considerar que es apto. Repetimos las medidas en régimen transitorio sobre este modelo, comenzando con la planta sin controlador (Figura 53):

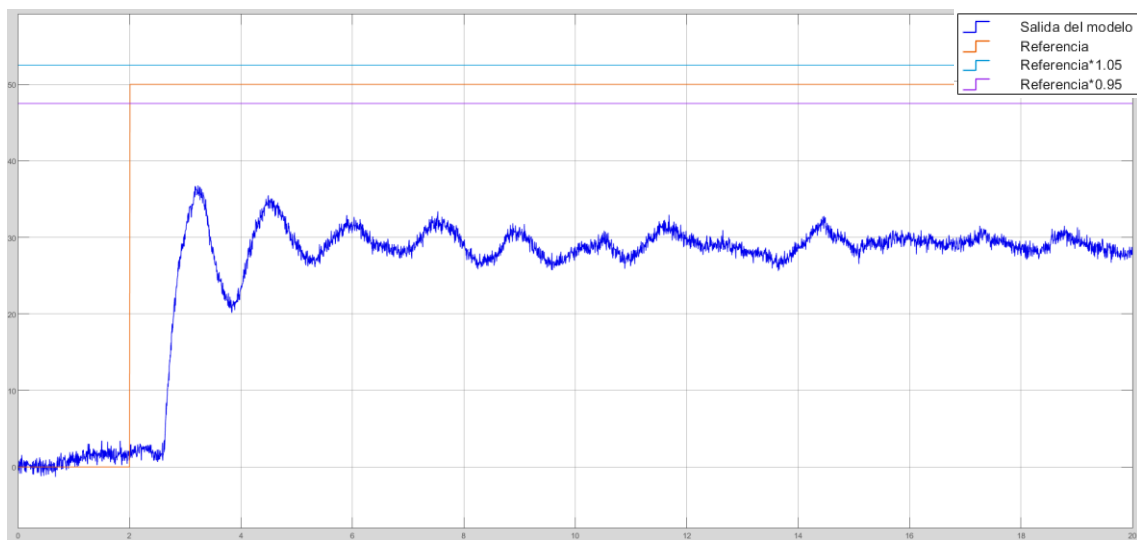


Figura 53 a)

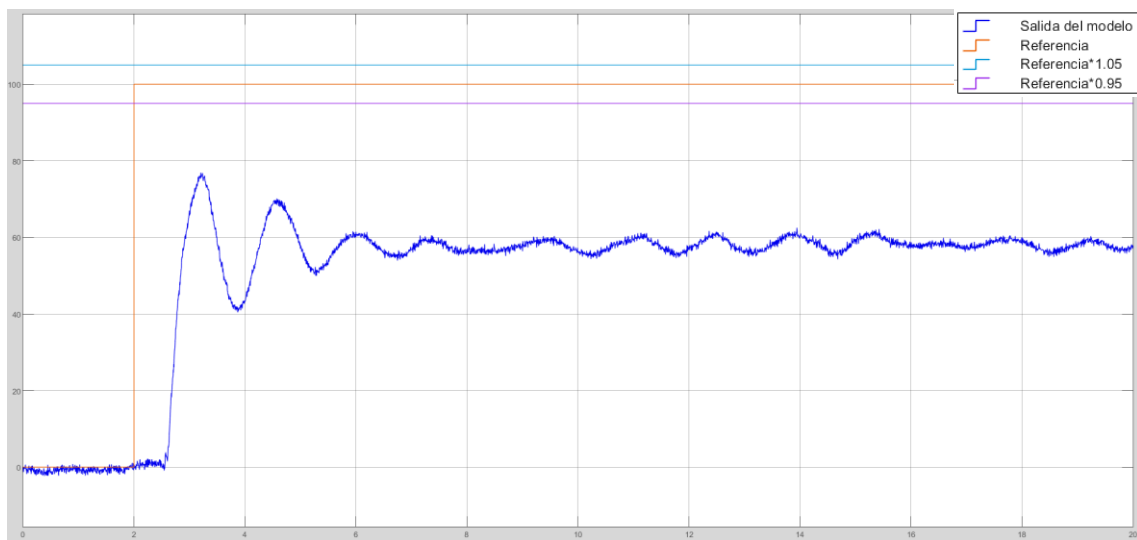


Figura 53 b)

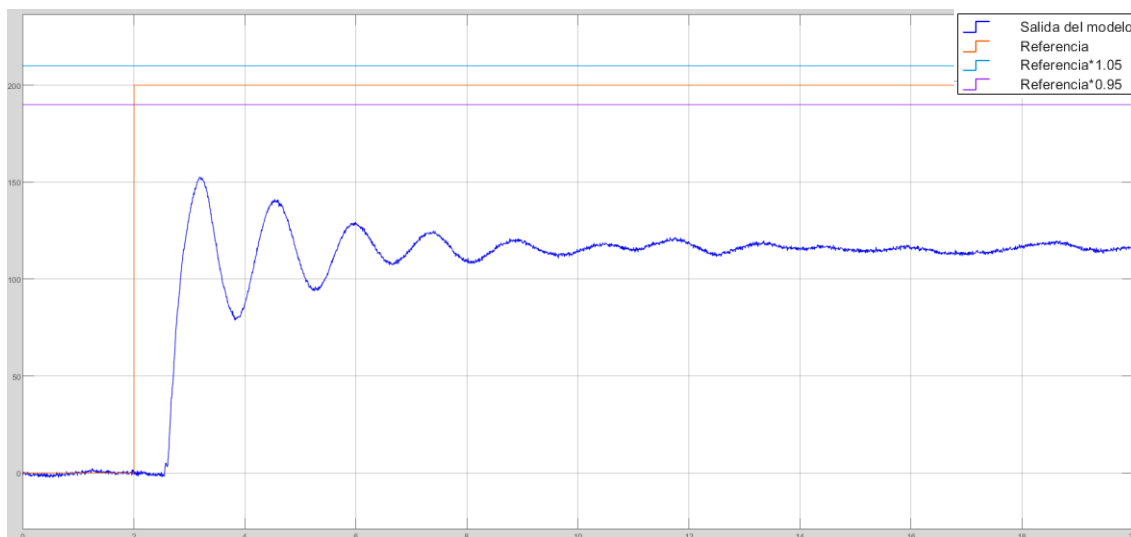


Figura 53 c)

Figura 53: a) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 50rpm (naranja) sin controlador, b) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 100rpm (naranja) sin controlador, c) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 200rpm (naranja) sin controlador

A raíz de los resultados arrojados por la *Figura 53* se pudieron registrar los parámetros en régimen transitorio y permanente de nuestro modelo. Gracias a la estabilidad presente en las salidas se puede medir el error en régimen permanente como la diferencia entre el valor de consigna y el valor medio de la señal de salida (*Tabla 6*):

Tabla 6: Parámetros en régimen transitorio y permanente del modelo en Espacio de Estados sin controlador

Escalón de referencia	Tiempo de subida (5%-95%)	Tiempo de establecimiento	Sobreimpulso máximo	Error en régimen permanente (ess)
50 rpm	0,246s	4,158s	34,4%	29,07rpm
100 rpm	0,249s	4,359s	34,58%	56,91rpm
200 rpm	0,249s	4,751s	33,95%	114,3rpm

Como se puede ver, el modelo no puede seguir a la entrada con precisión sin regulación alguna, pero por fortuna la labor de “tunning” realizada con el modelo Box-Jenkins nos sirve como punto de partida para ajustarlo. Siguiendo el mismo método empezamos por un ajuste manual, obteniendo los siguientes parámetros:

$$KP = 0.8; KI = 0,7; KD = 0.05$$

54:

Se puso a prueba el modelo con PID Clásico utilizando la estructura de la *Figura*

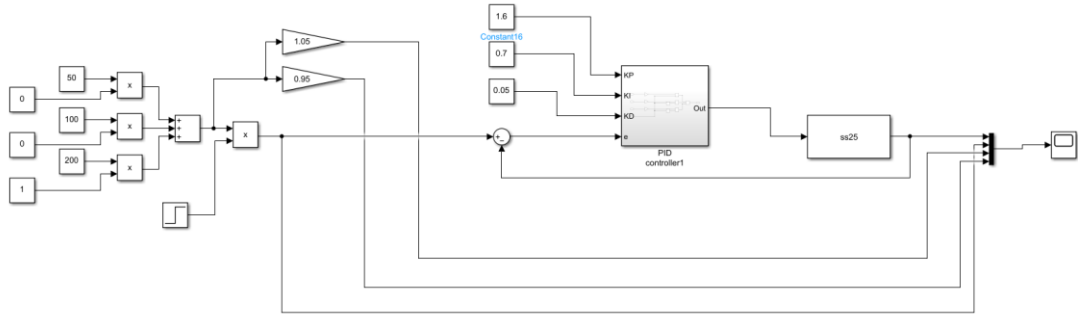


Figura 54: Estructura del modelo de velocidad en Espacio de Estados regulado mediante un Controlador PID Clásico

Con la implementación de estos nuevos parámetros en el esquema en Simulink del controlado PID Fuzzy, se obtienen los resultados mostrados en la *Figura 55* y las mediciones especificadas se detallan en la *Tabla 7*:

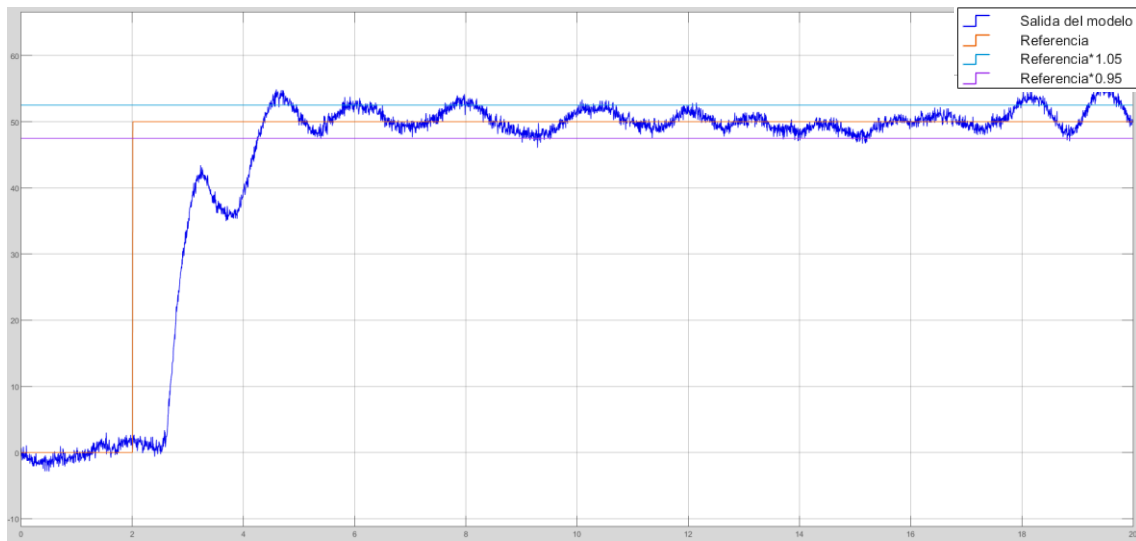


Figura 55 a)

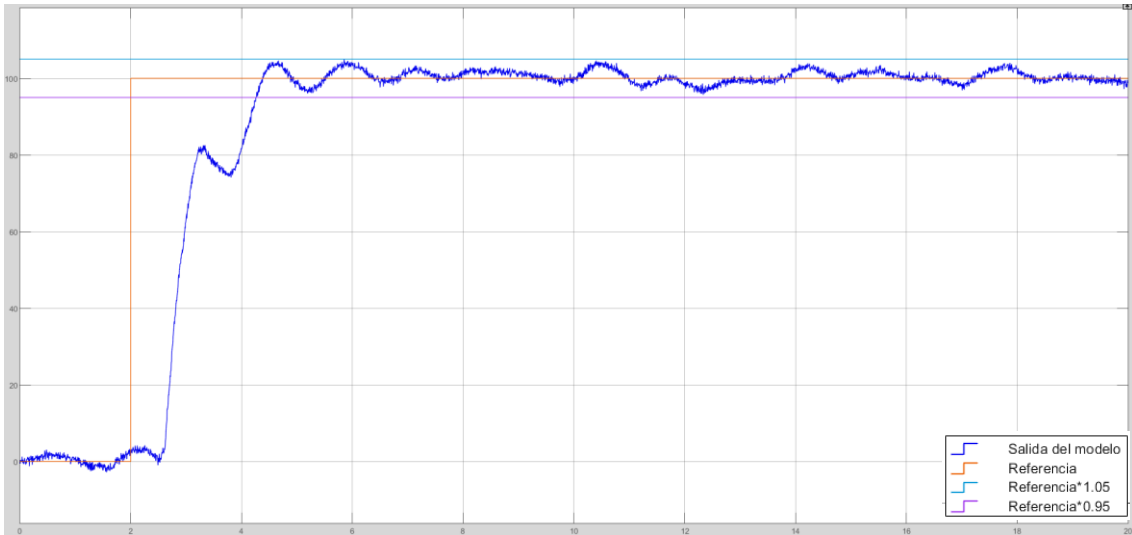


Figura 55 b)

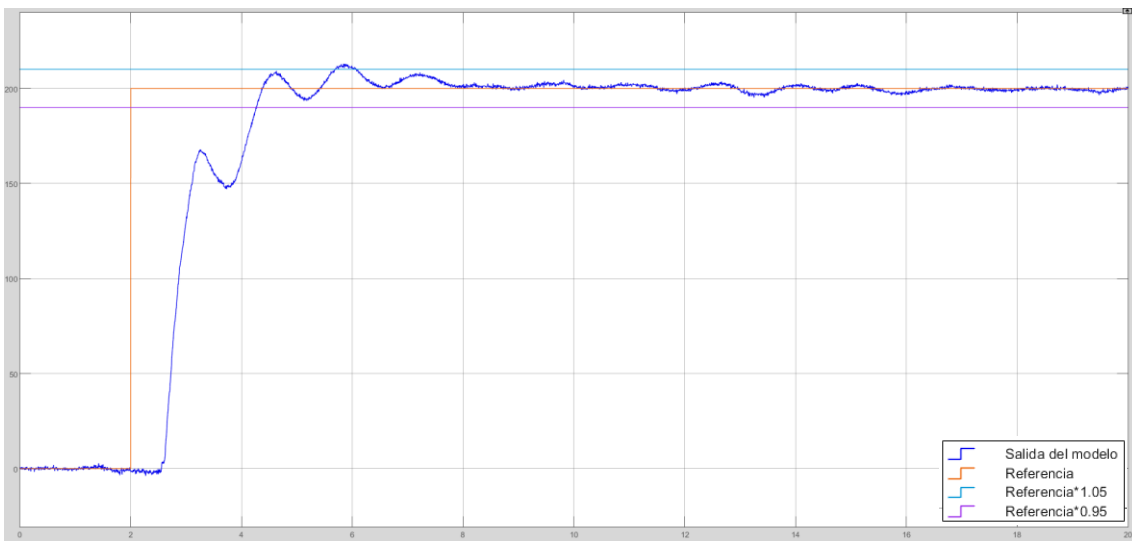


Figura 54 c)

Figura 54: a) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 50rpm (naranja) con Control PID Clásico, b) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 100rpm (naranja) con Control PID Clásico, c) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 200rpm (naranja) con Control PID Clásico

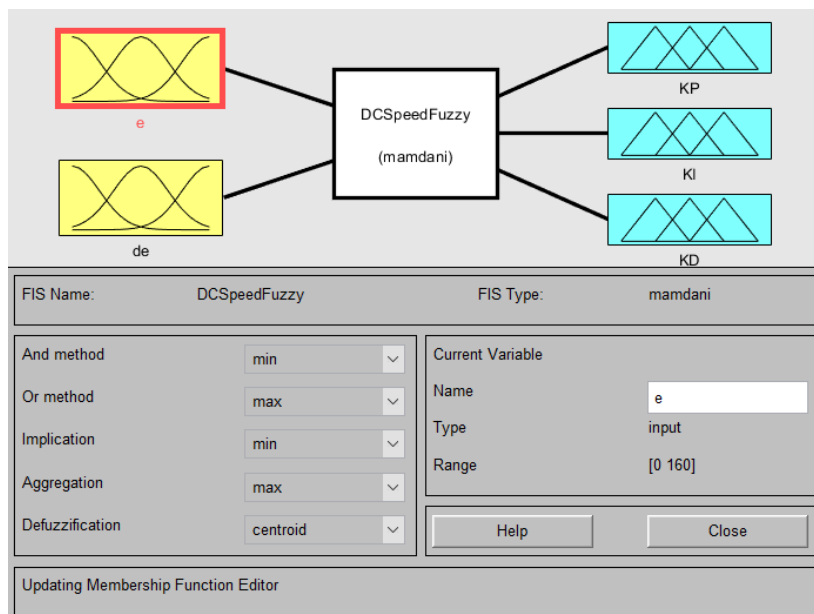


Figura 57 a)

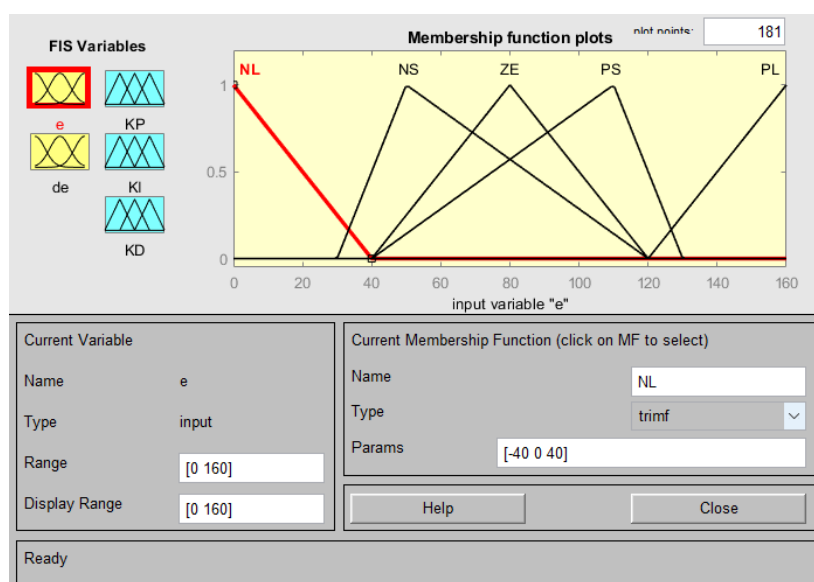


Figura 57 b)

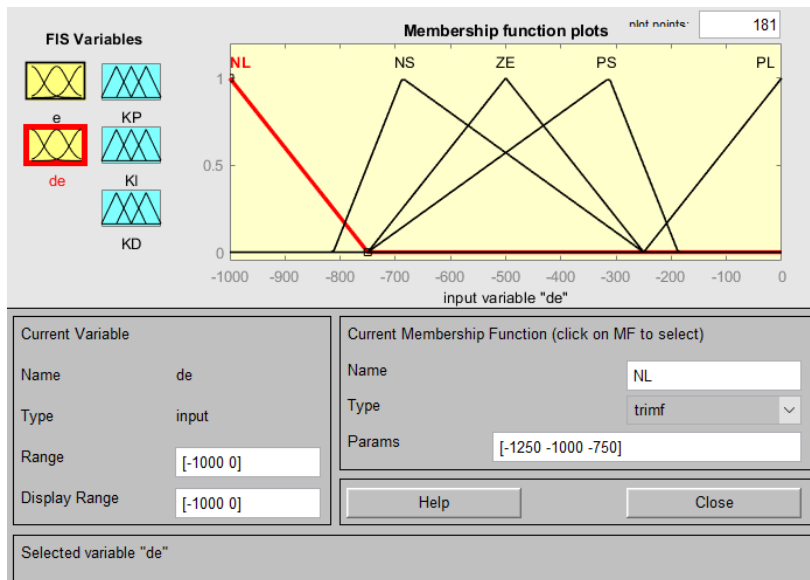


Figura 57 c)

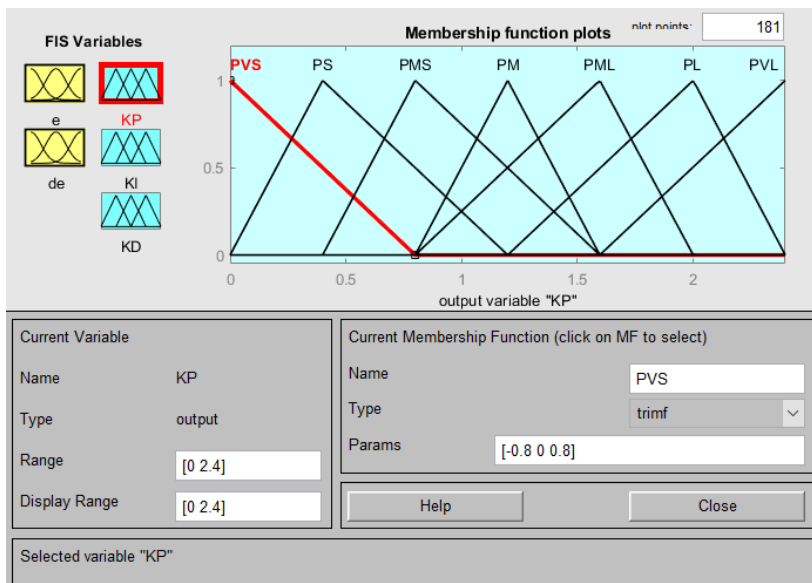


Figura 57 d)

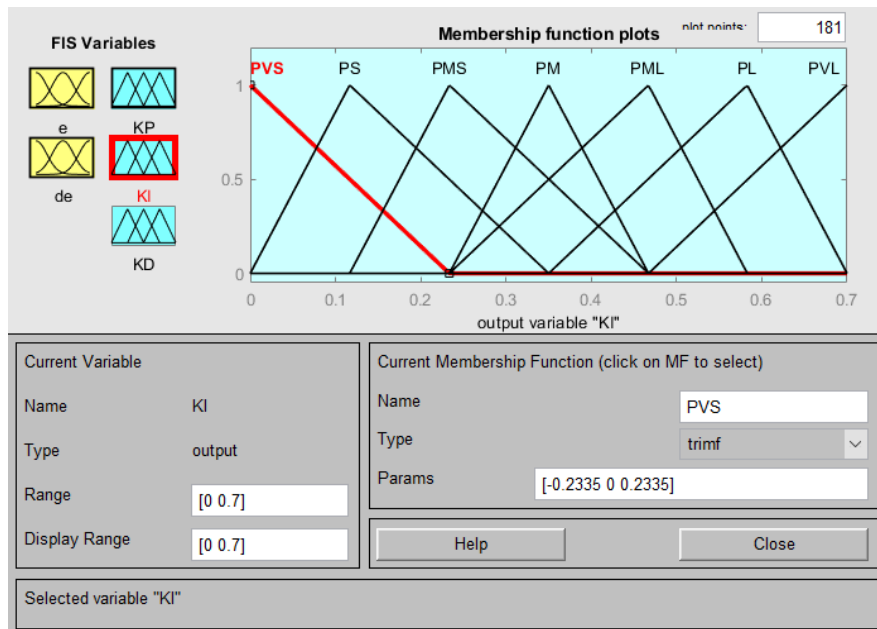


Figura 57 e)

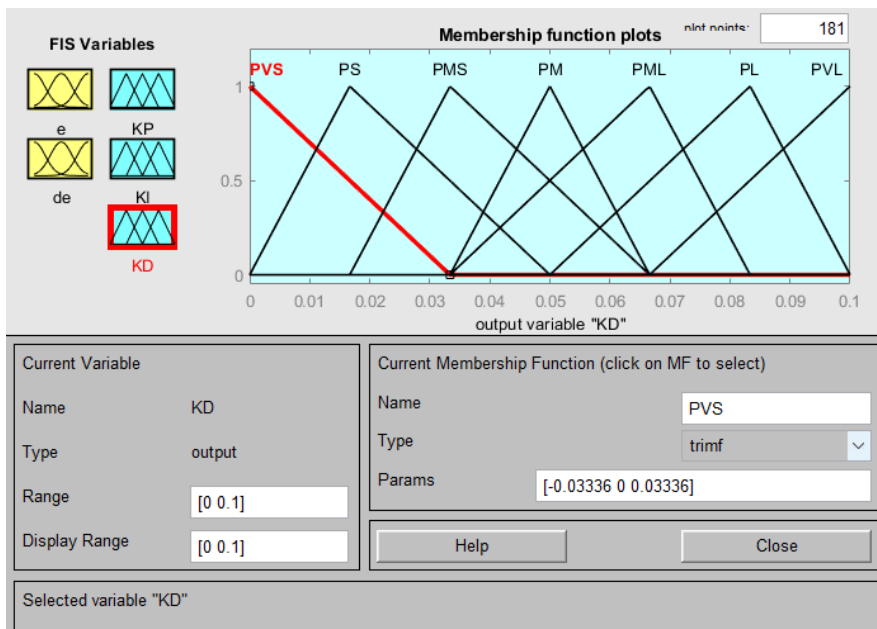


Figura 57 f)

Figura 57: a) Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados, b) Primera entrada del Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados: $e(t)$, c) Segunda entrada del Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados: $de(t)$, d) Primera salida del Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados: KP, e) Segunda salida del Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados: KI, f) Tercera salida del Controlador PID Fuzzy diseñado para regular el modelo de velocidad en Espacio de Estados: KD

Tras haber sintonizado el Controlador PID Fuzzy obtuvimos los siguientes resultados (*Figura 58*):

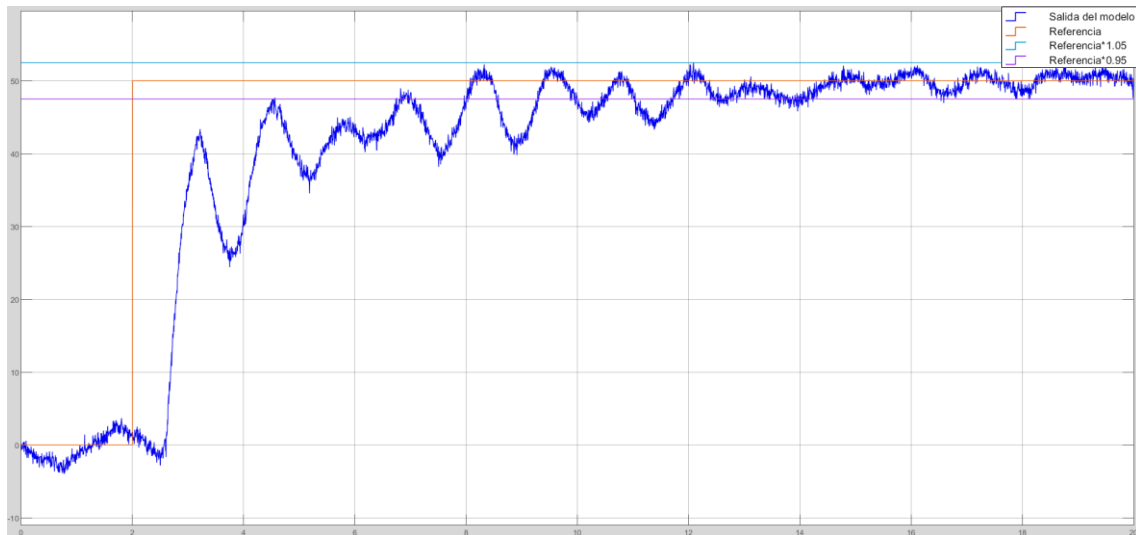


Figura 58 a)

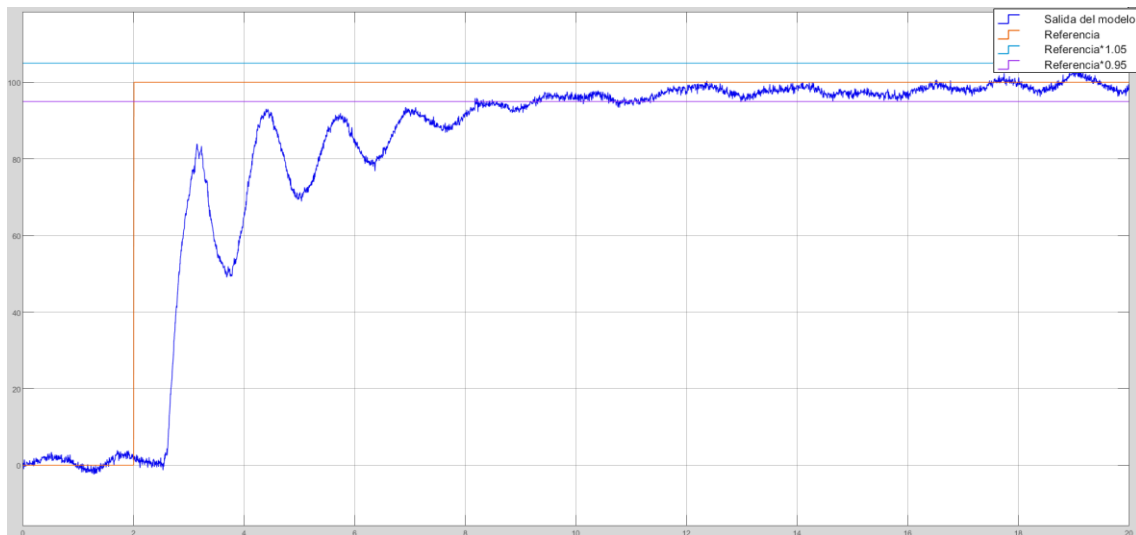


Figura 58 b)

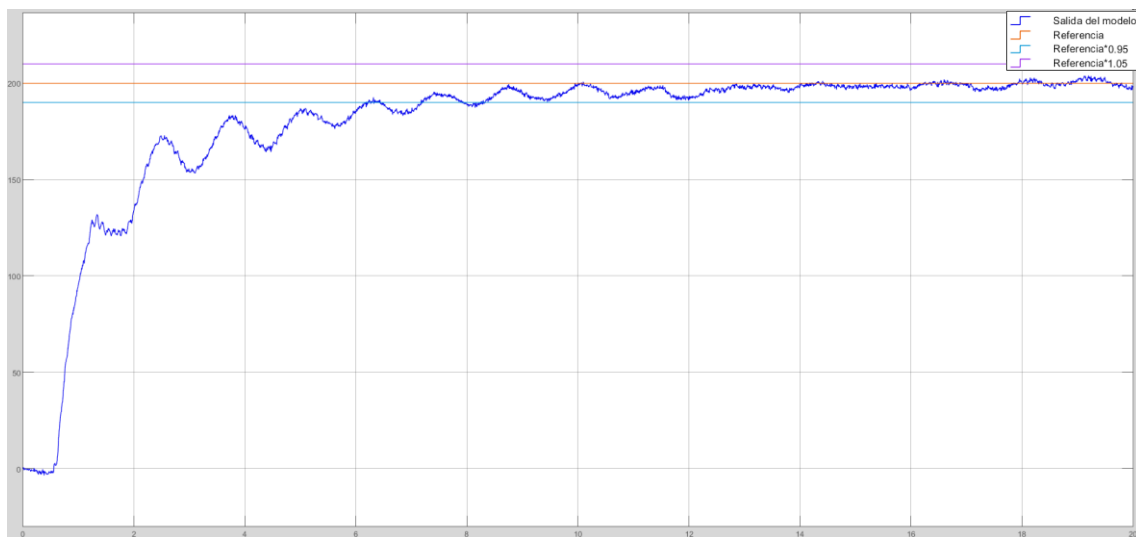


Figura 58 c)

Figura 58: a) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 50rpm (naranja) con Control PID Fuzzy, b) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 100rpm (naranja) con Control PID Fuzzy, c) Respuesta del modelo de velocidad (azul marino) en Espacio de Estados ante un escalón de 200rpm (naranja) con Control PID Fuzzy

Dichos resultados fueron agrupados en la Tabla 8:

Tabla 8: Parámetros en régimen permanente y transitorio del modelo de velocidad en Espacio de Estados regulado por un Controlador PID Fuzzy

Escalón de referencia	de	Tiempo de subida (5%-95%)	Tiempo de establecimiento	de	Sobreimpulso máximo
50 rpm		5.731s	11.729s		4,97%
100 rpm		6.601s	9.232s		2,57%
200 rpm		5.551s	8.182s		1.75%

Se aprecia que el controlador PID Clásico ofrece unos mejores tiempos de subida y de establecimiento que el PID Fuzzy, pero este último ofrece un sobreimpulso máximo mucho menor y mayor estabilidad (menores oscilaciones) en régimen permanente, sobre todo ante entradas pequeñas. Puesto que lo que buscamos es la mayor seguridad de que nuestro motor seguirá la consigna con precisión escogemos el PID Fuzzy en vez del Clásico, desmintiendo aquello obtenido con el modelo Box-Jenkins de velocidad, demostrando que una simulación extremadamente ruidosa puede llegar a falsear nuestros resultados.

A continuación, se expondrá también en este caso la respuesta del sistema ante una señal “práctica” para poder comparar con Box-Jenkins (Figura 59):

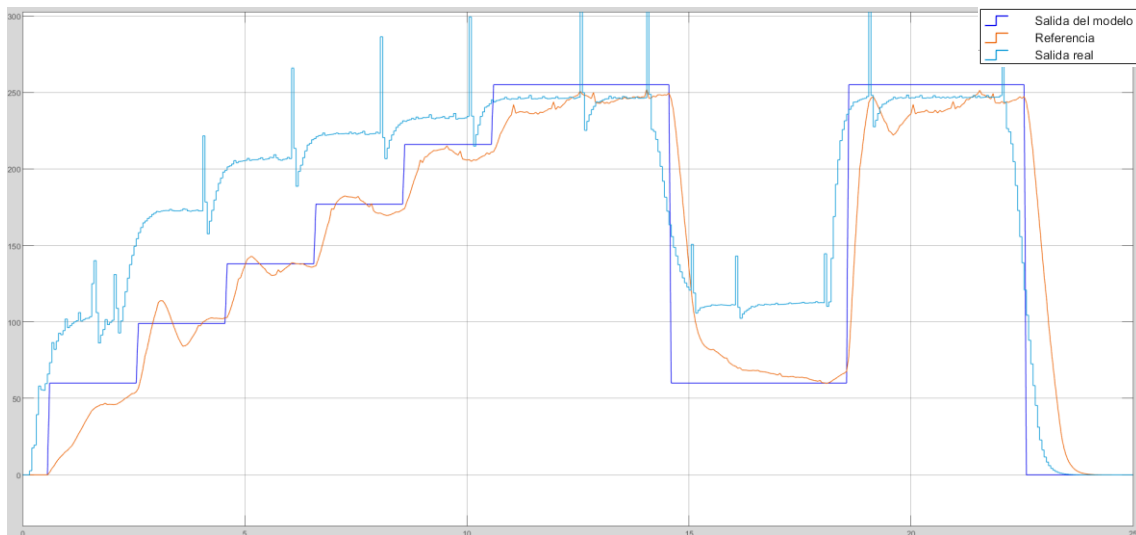


Figura 59 a)

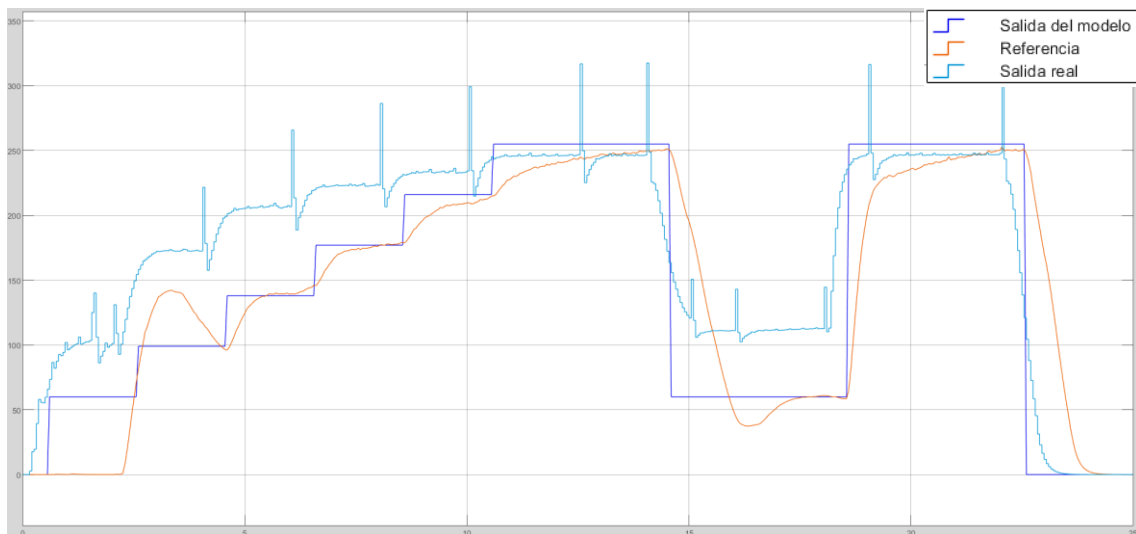


Figura 59 b)

Figura 58: a) Salida del modelo de velocidad en Espacio de Estados (azul marino) y del sistema real (cian), ambos con Control PID Clásico, ante una misma referencia (naranja), b) Salida del modelo de velocidad en Espacio de Estados (azul marino) y del sistema real (cian), ambos con Control PID Fuzzy, ante una misma referencia (naranja)

El amortiguamiento de la salida y la rápida respuesta nos hacen elegir el modelo en Espacio de Estados por encima del Box-Jenkins. Ello, además de la ausencia de problemas de simulación ante entradas tipo escalón, nos inclina a proseguir nuestro estudio a partir de las conclusiones de este último modelo para diseñar los controladores que implementaremos en el motor real.

3.3. Conclusiones sobre los modelos a elegir

Dadas las respuestas arrojadas previamente por sendos modelos de Box-Jenkins y Espacio de Estados, podemos concluir que, en el caso del modelo de posición, Box-Jenkins presenta oscilaciones que nos son de ayuda para emular el comportamiento prácticamente sinusoidal que presenta el motor al intentar seguir una consigna de posición angular, mientras que en el caso del modelo de velocidad optaremos por el Espacio de Estados, pues refleja mejor el comportamiento deseado por nosotros ante una referencia de velocidad angular al presentar amortiguamiento, suavidad y precisión en la salida. Partiremos de las conclusiones de estos modelos para implementar aquellos controladores con que trabajará el motor para cada tipo de entrada (posición y velocidad) y veremos si las premisas que acabamos de obtener a raíz de los resultados arrojados por cada simulación se materializan también en la realidad.

4. Pruebas reales en el Motor DC

Una vez diseñados los modelos de posición y velocidad, y puestos a prueba, es hora de implementar los controladores diseñados con ellos en el motor real, partiendo de Box-Jenkins en posición y de Espacio de Estados en velocidad, como acabamos de mencionar.

4.1. Motor sin perturbaciones

Aunque un sistema sin perturbaciones solo es alcanzable en un mundo puramente matemático e ideal trataremos de aproximarnos a pesar de ello lo más posible a la perfección teórica en nuestra experiencia realista y material. Para ello se experimentó con el motor sin acoplarle las ruedas ni montarlo en el resto del armazón del vehículo para poder registrar la respuesta del dispositivo sin que la acción de otras fuerzas (como el peso de las ruedas, la fricción estática de estas, la reacción del suelo, el rozamiento de los rodamientos,...) afectara a su comportamiento.

4.1.1. Motor sin perturbaciones: Posición

El modelo de Simulink implementado para integrar los controladores PID Clásico y PID Fuzzy en el motor real se expone a continuación (*Figura 60 y 61*):

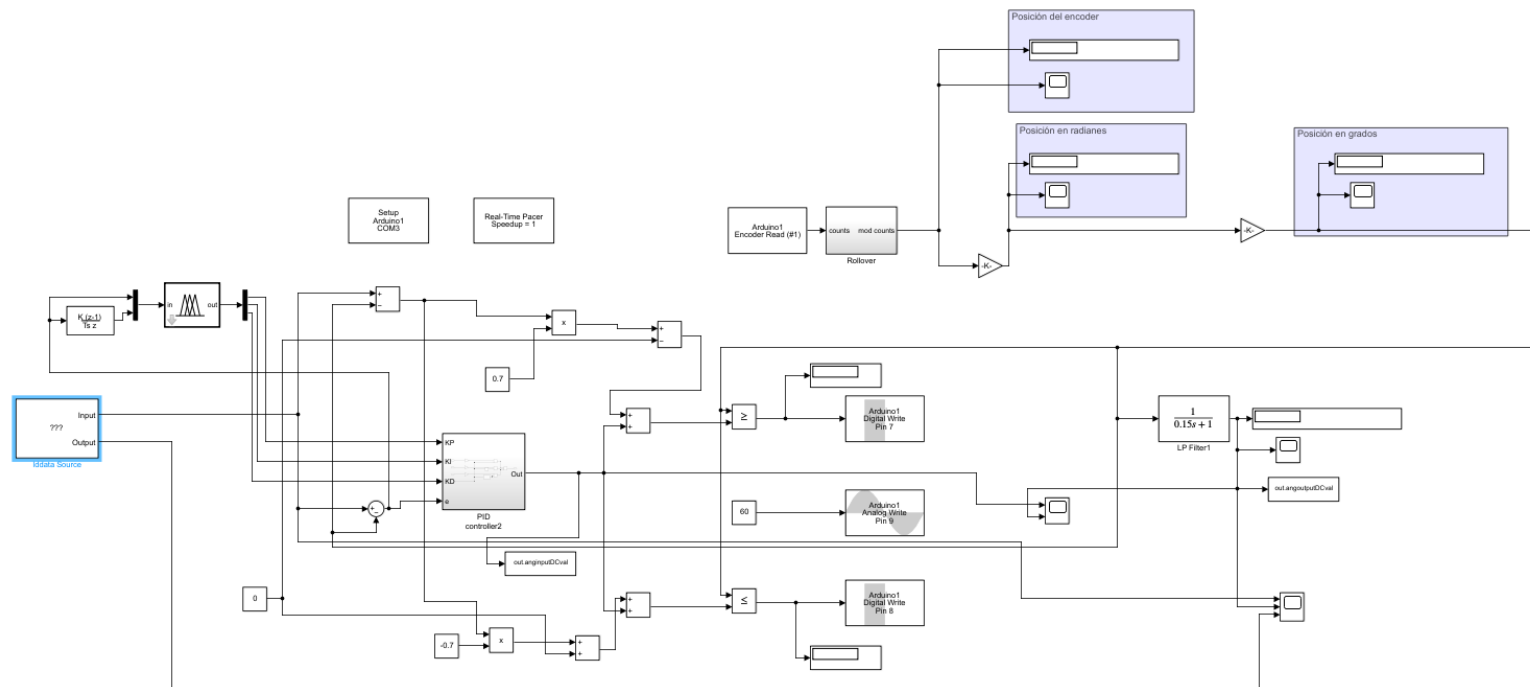


Figura 61: Estructura en Simulink empleada para hacer llegar al motor real la referencia de posición regulada por un Controlador PID Fuzzy

Si recordamos el comportamiento del sistema sin controlador la posición del motor se determinaba comparando la consigna con la orientación real del eje del dispositivo proporcionada por el “Encoder”, trabajando a una velocidad fija y girando en un sentido o en el opuesto en función de si nos encontrábamos por encima o por debajo de la referencia, y fuera de una tolerancia de $\pm 5^\circ$. Ahora en vez de comparar la referencia con la salida se coteja esta última con la salida del controlador, permitiendo una anticipación a la entrada que hace que el motor frene e inicie el giro en el sentido opuesto antes de “pasarse de largo” la referencia, volviéndose así capaz de oscilar lo menos posible en torno a la consigna deseada. Se probaron los controladores derivados tanto del modelo Box-Jenkins como del de Espacio de Estados para corroborar las conclusiones obtenidas en apartados anteriores. Lo resultados gráficos se detallan en la *Figura 62*:

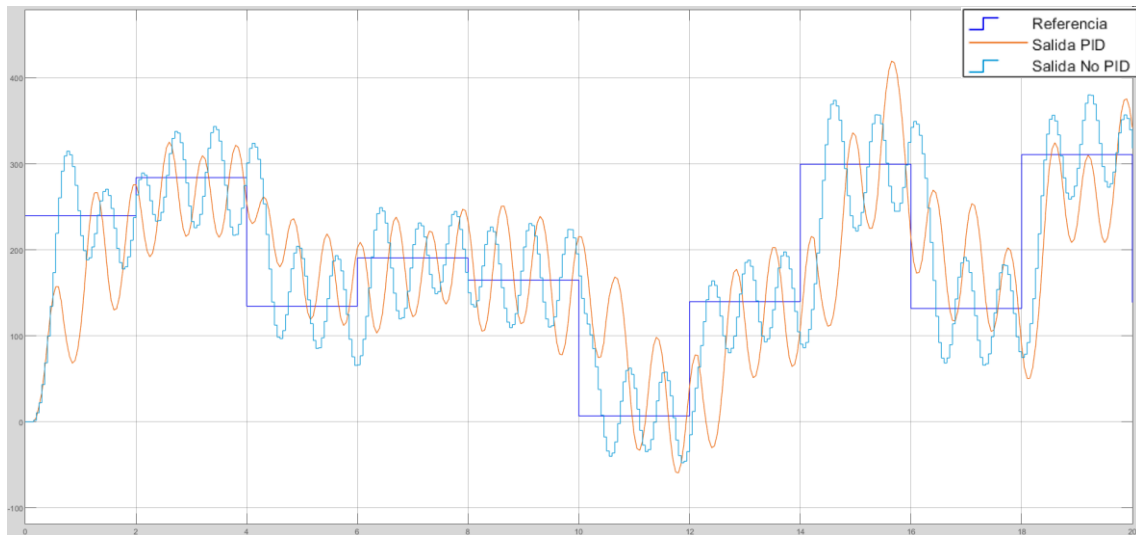


Figura 62 a)

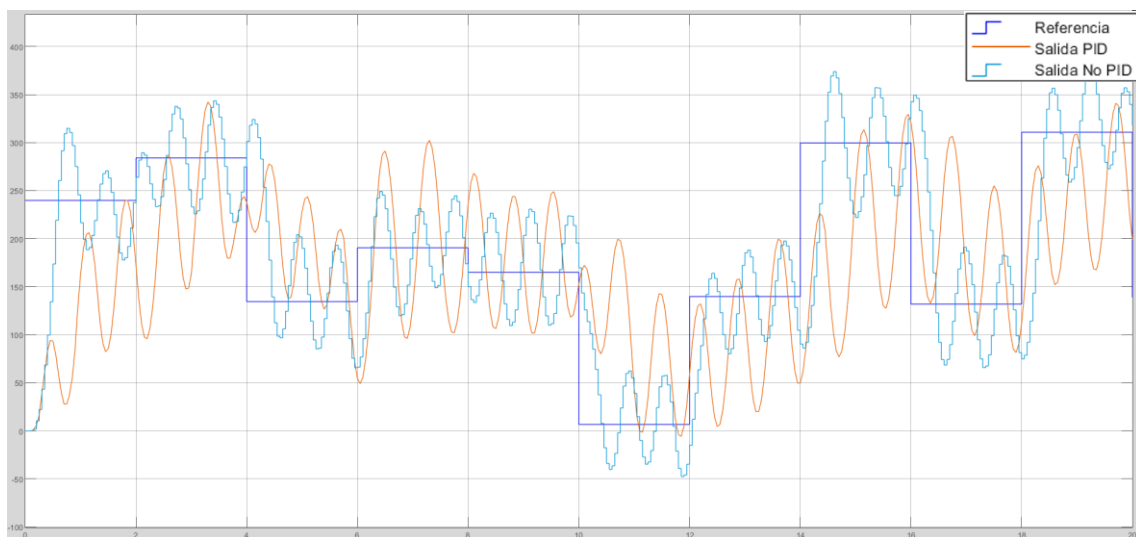


Figura 62 b)

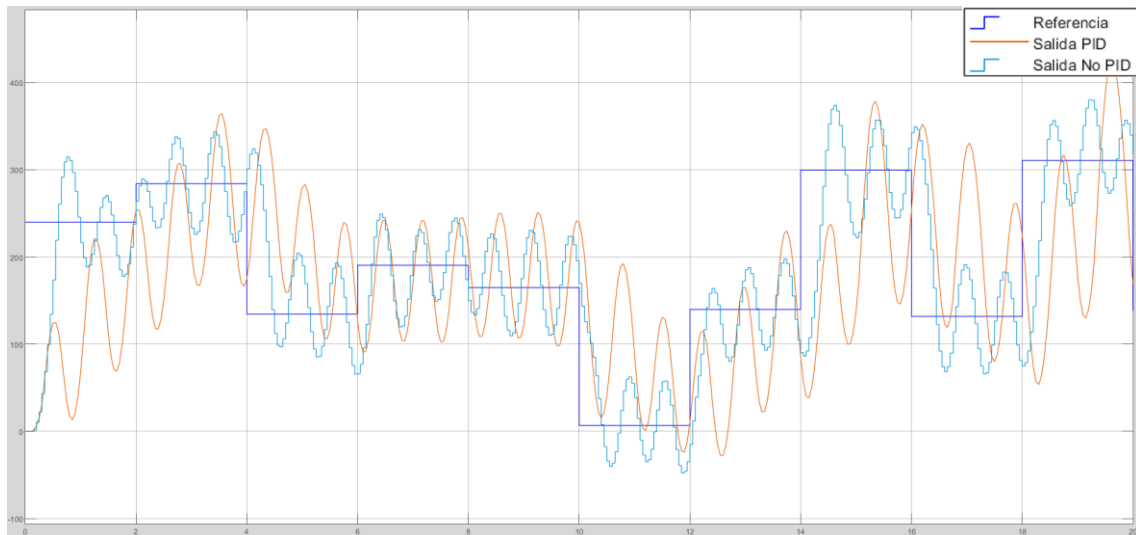


Figura 62 c)

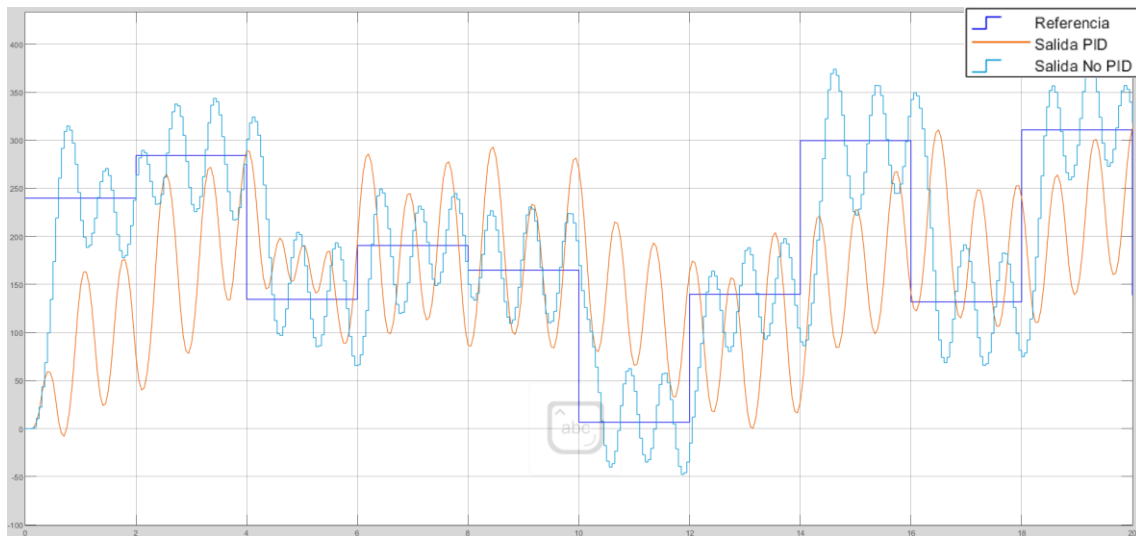


Figura 62 d)

Figura 62: a) Respuesta real del Motor DC regulado con PID Clásico, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de posición de Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), b) Respuesta real del Motor DC regulado con PID Fuzzy, partiendo de los conjuntos de entrada obtenidos a partir de las pruebas con el modelo de posición de Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), c) Respuesta real del Motor DC regulado con PID Clásico, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de posición en Espacio de Estados (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), d) Respuesta real del Motor DC regulado con PID Fuzzy, partiendo de los conjuntos de entrada obtenidos a partir de las pruebas con el modelo de posición en Espacio de Estados (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino)

Observamos que los resultados obtenidos a partir de los controladores derivados del modelo Box-Jenkins se comportan mejor que los generados con Espacio de Estados, al tener menor oscilación, mayor precisión y menor tiempo de subida para alcanzar la consigna deseada, por lo que partiremos de ellos para diseñar el controlador que mejore lo máximo posible el comportamiento de nuestro sistema. Comenzaremos por el PID Clásico y a partir de este iremos a por el PID Fuzzy. Es necesario destacar que el criterio que se utilizó anteriormente con que se imponía un margen de 5° por encima y por debajo de la consigna se ha sustituido por otro en que se aboga por que este margen varíe proporcionalmente a la diferencia entre la posición real del motor y la deseada (a mayor diferencia entre salida y consigna mayor margen alrededor de la consigna), para que pudiera alcanzar con mayor rapidez este valor si se encontraba su ángulo muy alejado del deseado. Así, los valores de las constantes del PID clásico son:

$$K_P = 0.1375, K_I = 1.4, K_D = 0.15$$

Y la respuesta del sistema con Controlador PID Fuzzy (*Figura 63*):

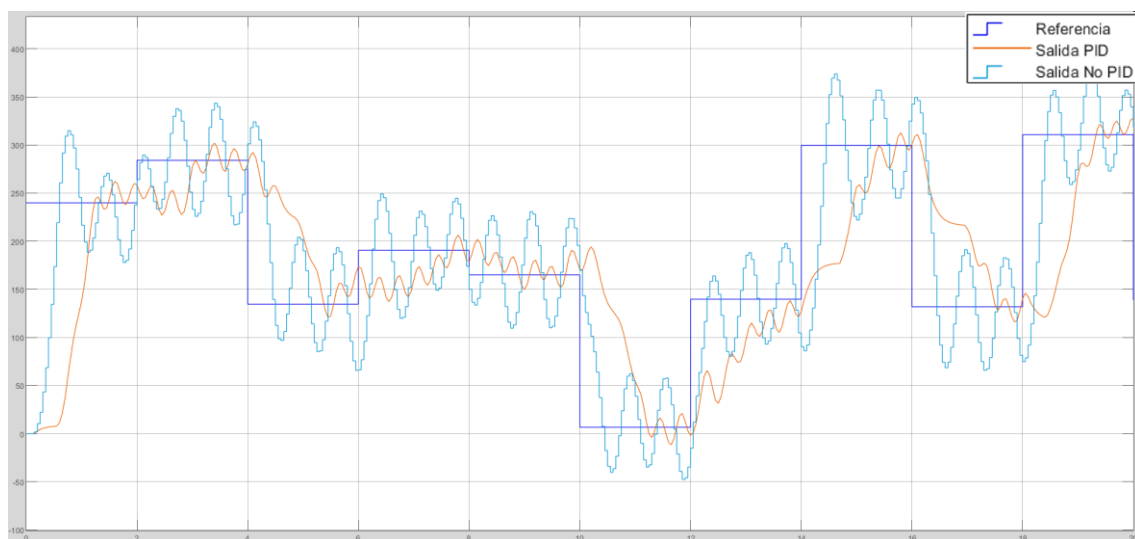


Figura 63: Respuesta real del Motor DC regulado con PID Clásico ajustado para lograr la mejor respuesta posible partiendo de las conclusiones arrojadas por el controlador PID Clásico derivado del modelo de posición Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino)

Al igual que en los modelos simulados, para el diseño del controlador PID Fuzzy se partió del diseño de las constantes del PID Clásico y a raíz de ellas se modificaron los rangos de sus salidas para que los valores medios de estas estuvieran en torno a los valores obtenidos de K_P , K_I y K_D como resultado del “tunning”, mientras que, de nuevo, las entradas se conservaron exactamente con los mismos rangos. En definitiva, el ajuste del controlador PID Fuzzy sigue siendo manual y puramente experimental, siguiendo los mismos pasos que hasta ahora, por lo que las únicas imágenes que se conservarán de los PID Fuzzy que se diseñen de ahora en adelante serán las de las salidas de los mismos, ya que las entradas son las mismas y no se han incluido salidas nuevas (*Figura 64*):

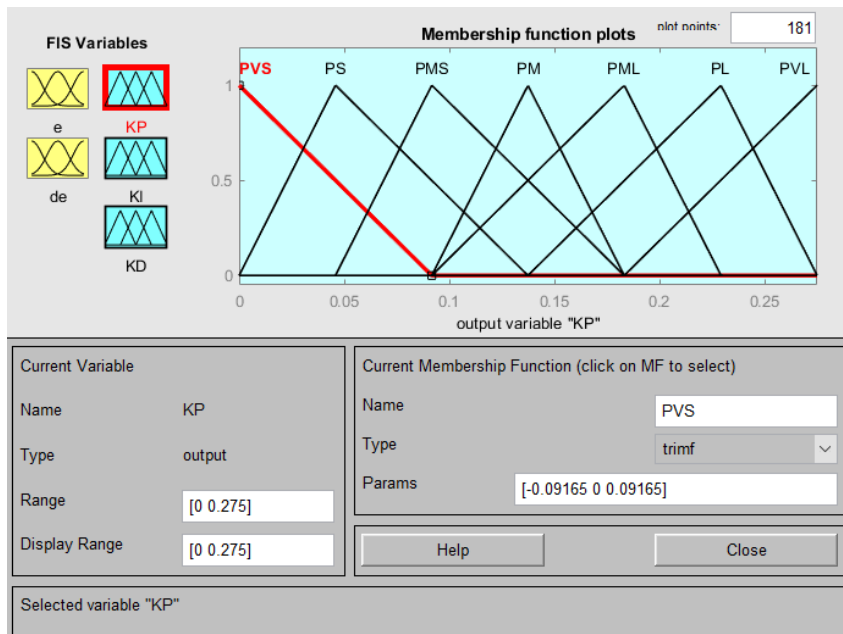


Figura 64 a)

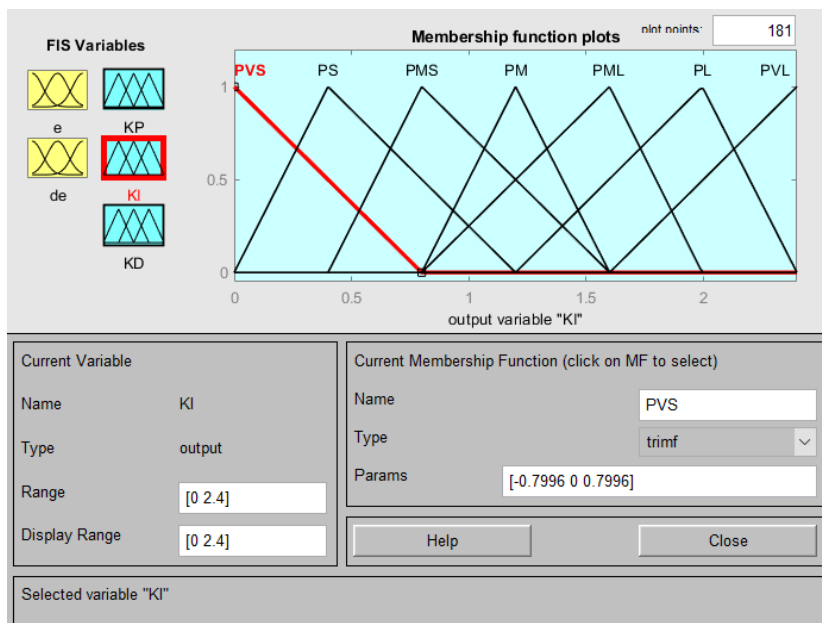


Figura 64 b)

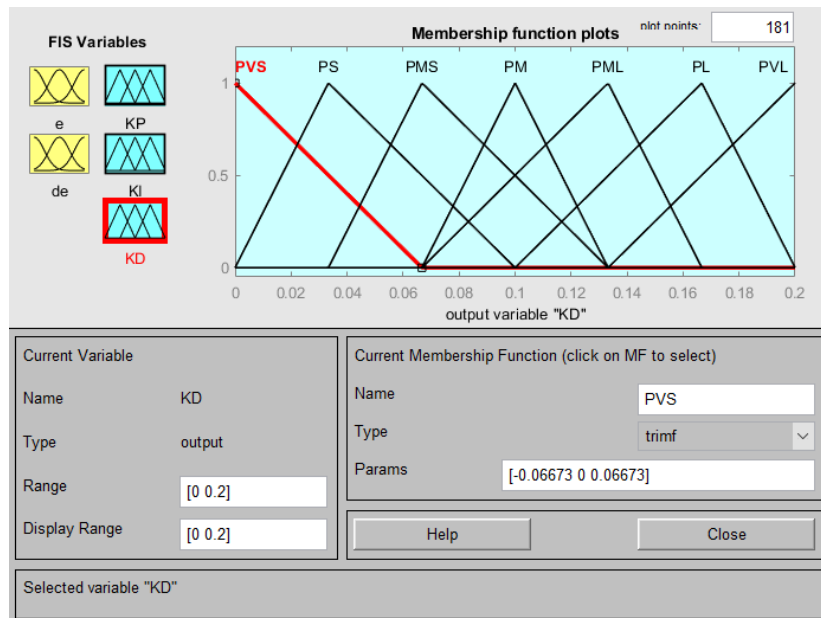


Figura 64 c)

Figura 64: a) Primera salida del Controlador PID Fuzzy de posición diseñado a partir de las constantes del PID Clásico que acaban de exponerse: KP, b) Segunda salida del Controlador PID Fuzzy de posición diseñado a partir de las constantes del PID Clásico que acaban de exponerse: KI, c) Tercera salida del Controlador PID Fuzzy de posición diseñado a partir de las constantes del PID Clásico que acaban de exponerse: KD

La Figura 65 pone de nuevo en relieve que la adaptabilidad del Controlador PID Fuzzy permite una mayor velocidad de respuesta y menor oscilación, viendo así su superioridad sobre el PID Clásico en el modelo de posición. Es destacable el hecho de que el valor de la constante KP se ha tenido que reducir sobremanera para conseguir que el motor se centrara más en torno a la consigna deseada, puesto que un valor excesivo de esta generaba oscilaciones excesivas en el sistema real. Con ello se ha conseguido reducir la desviación típica desde un valor de unos 100° hasta aproximadamente 20° , conservando bajos tiempos transitorios y alta precisión:

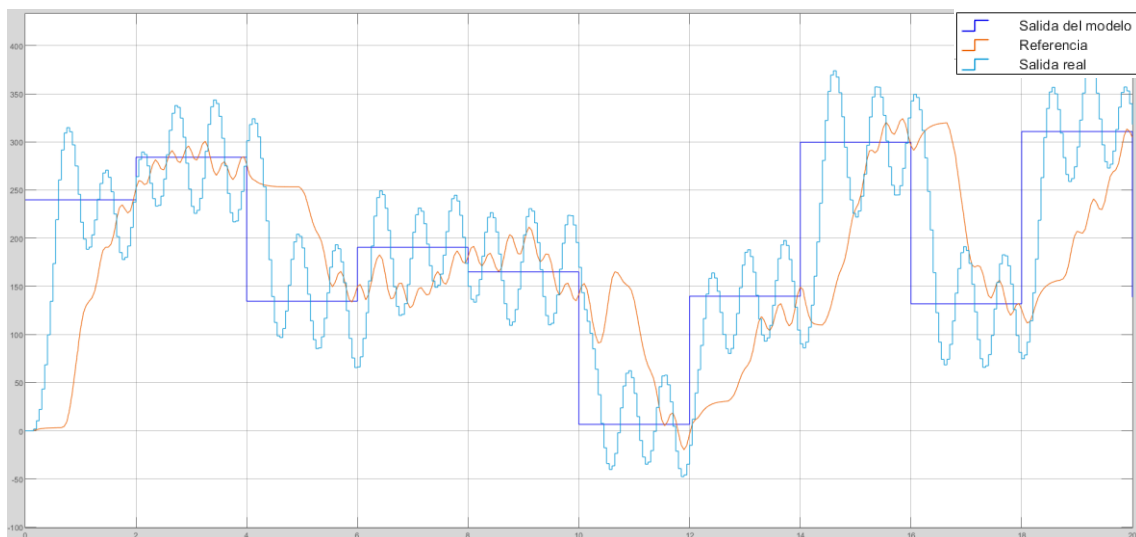


Figura 65: Respuesta real del Motor DC regulado con PID Fuzzy ajustado para lograr la mejor respuesta posible partiendo de las conclusiones arrojadas por el controlador PID Fuzzy derivado del modelo de posición Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino)

Se midió la respuesta del sistema ante un escalón de 270° para comprobar su comportamiento en régimen transitorio, obteniendo los resultados que muestran las gráficas de las Figura 66:

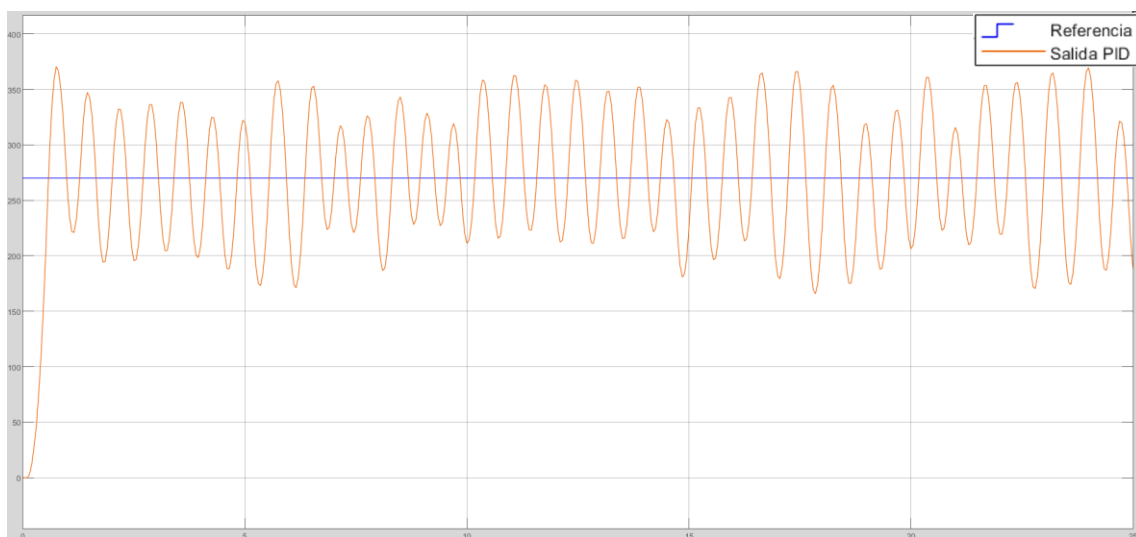


Figura 66 a)

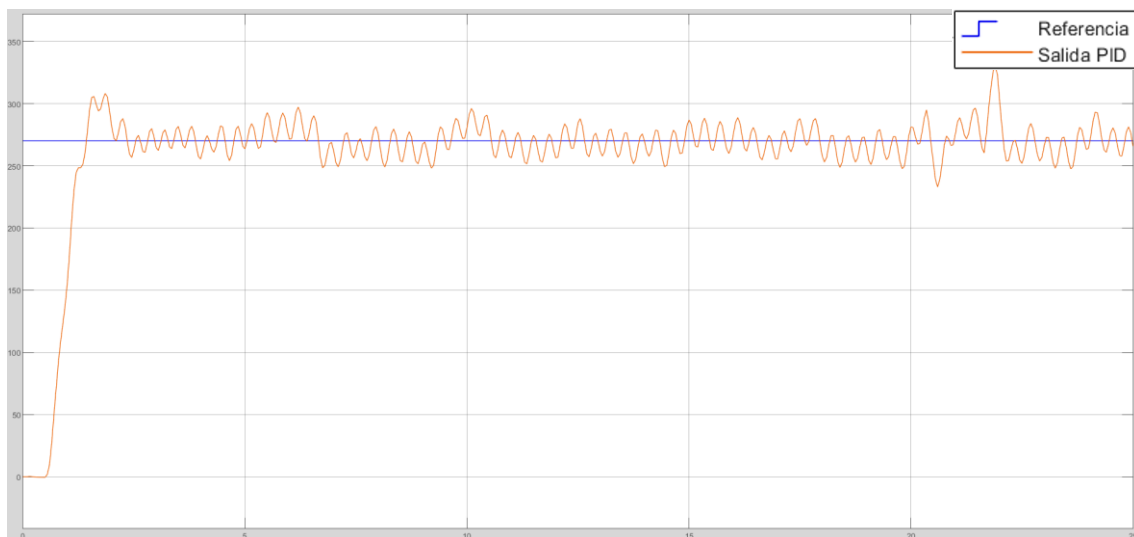


Figura 66 b)

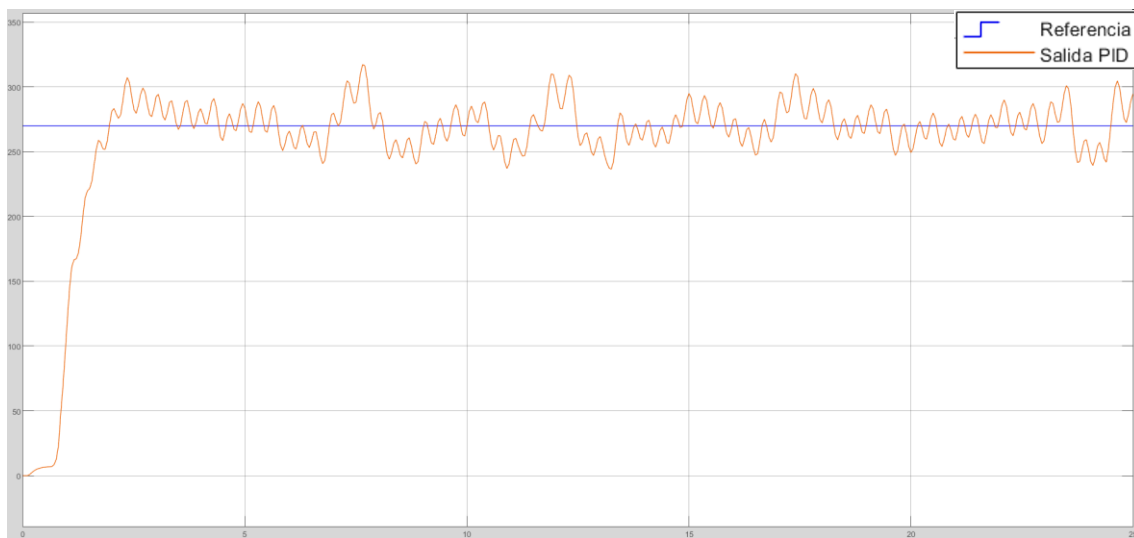


Figura 66 c)

Figura 66: a) Respuesta real de posición del motor sin controlador (naranja) ante una referencia constante de 270° (azul marino), b) Respuesta real de posición del motor con Controlador PID Clásico (naranja) ante una referencia constante de 270° (azul marino), c) Respuesta real de posición del motor con Controlador PID Fuzzy (naranja) ante una referencia constante de 270° (azul marino)

Para cuantificar el comportamiento real del sistema, se repitieron los experimentos de medición del comportamiento en régimen permanente y transitorio que se aplicaron sobre los modelos que emulaban el comportamiento del motor sobre el sistema real. Consideraremos en este caso que el régimen permanente se alcanza cuando se llega a una señal prácticamente sinusoidal de amplitud aproximadamente constante después del primer pico de amplitud. Gracias a las gráficas arrojadas en la Figura 66 fuimos capaces de cuantificar el tiempo de subida (t_s), el de establecimiento (t_e), el

sobreimpulso máximo (Mp), y el error en régimen permanente (ess), y se agrupó todo ello en la *Tabla 9*:

Tabla 9: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy

Sistema	ts (5%-95%)	te	Mp (%)	ess
Sin controlador	0,361s	0,746s	36,852%	$\approx 0^\circ$
PID	0,771s	1,583s	22,222%	$\approx 0^\circ$
PID Fuzzy	0,935s	2,354s	17,481%	$\approx 0^\circ$

Nuevamente se observa que el sobreimpulso sin controlador es excesivo, aunque los tiempos en régimen transitorio sean mejores. Por su parte, los tiempos transitorios del PID Clásico son mejores que los del PID Fuzzy, pero el sobreimpulso máximo es mayor en el primer caso, además de sufrir de amortiguamiento y precisión menores. Es por ello que podemos asegurar apoyándonos en estas conclusiones que el Controlador PID Fuzzy arroja unos resultados mejores que el PID Clásico.

4.1.2 Motor sin perturbaciones: Velocidad

Se realizó el mismo proceso que en la determinación del controlador de posición, partiendo del controlador PID Clásico derivado del modelo en Espacio de Estados, de manera que los reguladores se integraron con el resto de la arquitectura de Simulink de la siguiente manera (*Figuras 67 y 68*):

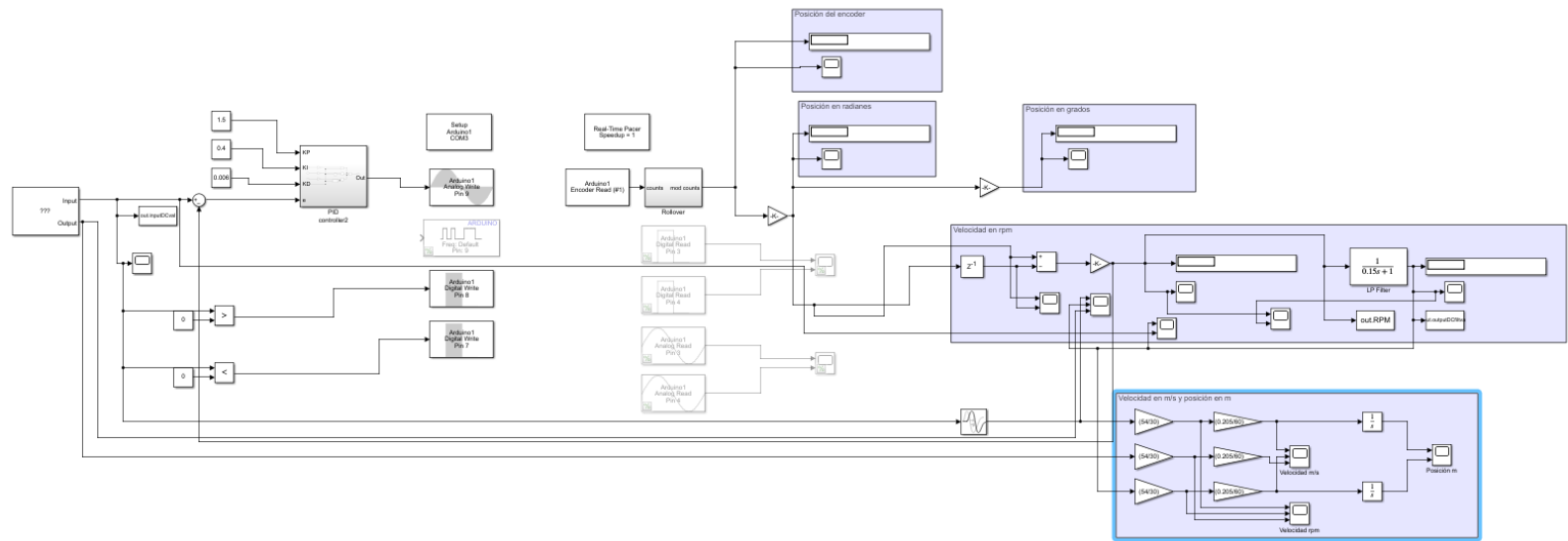


Figura 67: Estructura en Simulink empleada para hacer llegar al motor real la referencia de velocidad regulada por un Controlador PID Clásico

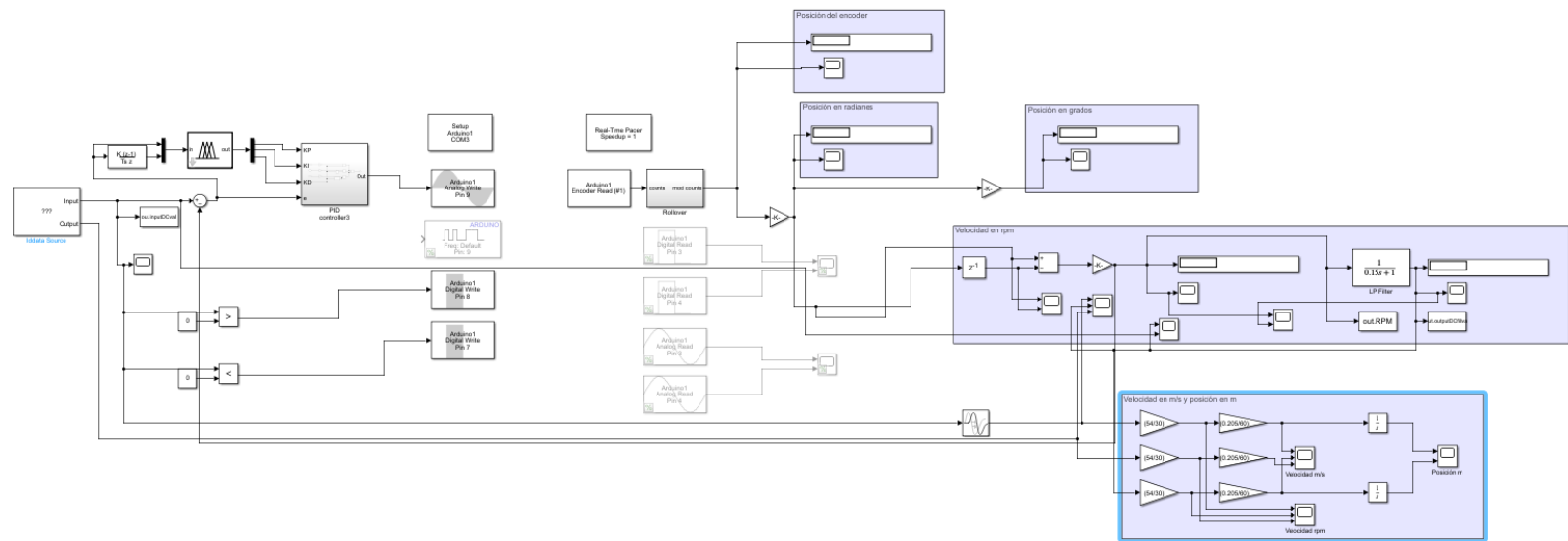


Figura 68: Estructura en Simulink empleada para hacer llegar al motor real la referencia de velocidad regulada por un Controlador PID Fuzzy

La implementación de sendos controladores es mucho más fácil que en el caso del de posición, pues basta con hacer llegar la salida del controlador a la entrada de habilitación (“Enable”) del puente en H, sin necesidad de comparar con la salida real del motor. Se introdujo, no obstante, una modificación ligera a la estructura con respecto al original para lograr que, si la consigna de velocidad era negativa se permitiera girar al motor en sentido contrario, habilitando así la rotación en dos sentidos. Se probaron en el motor los controladores diseñados con los modelos Box-Jenkins y Espacio de Estados, para corroborar una vez más que este último proporciona resultados más robustos, estables y precisos que el primero, apreciándose los siguientes resultados (*Figura 69*):

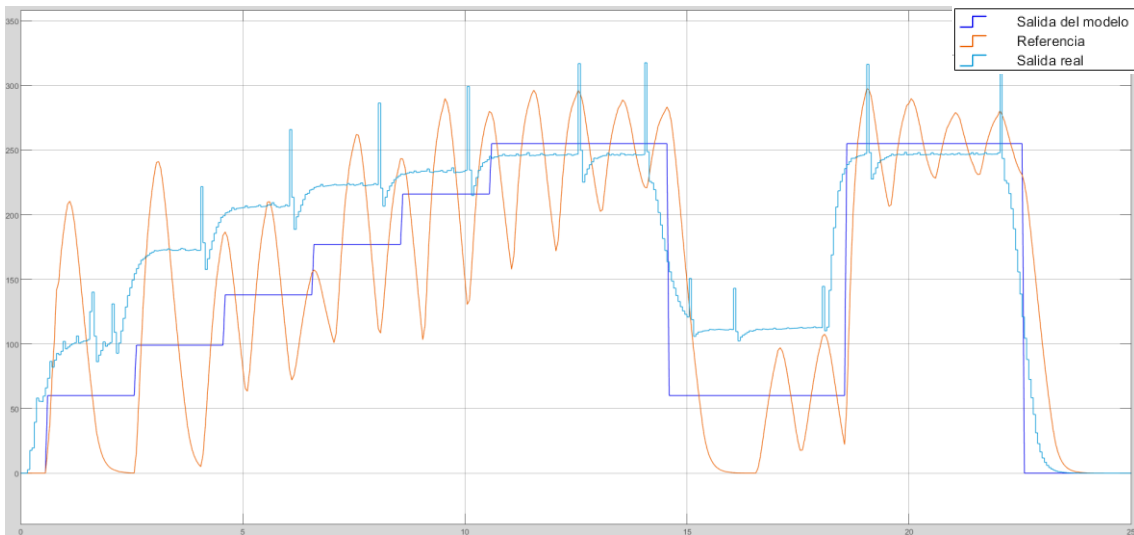


Figura 69 a)

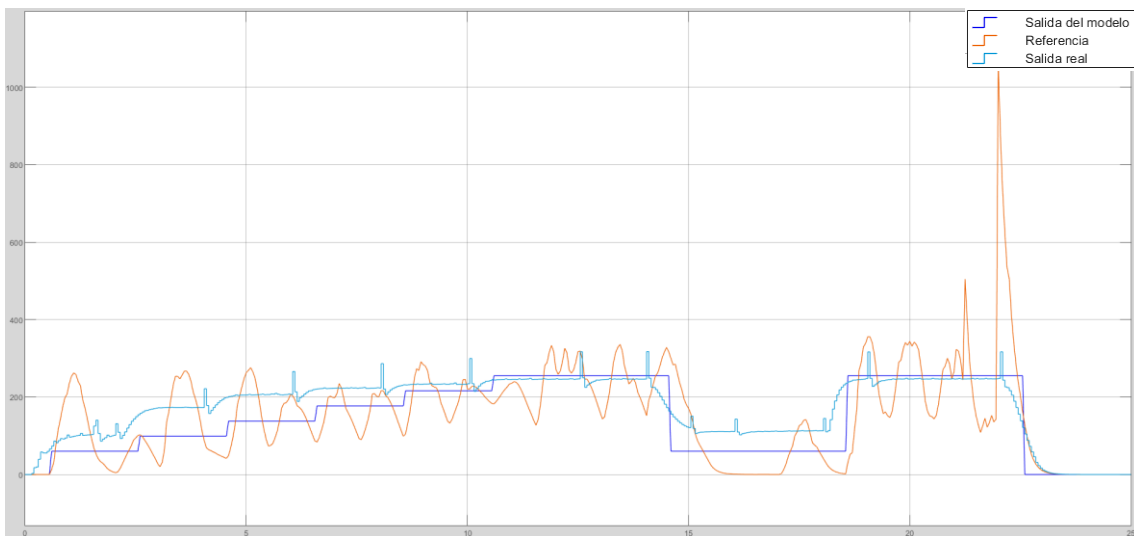


Figura 69 b)

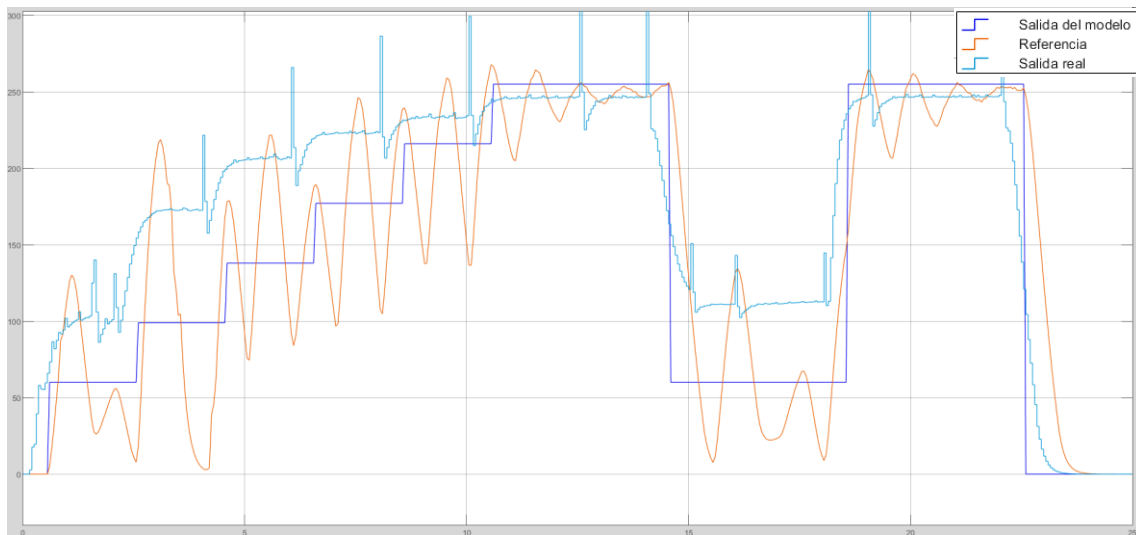


Figura 69 c)

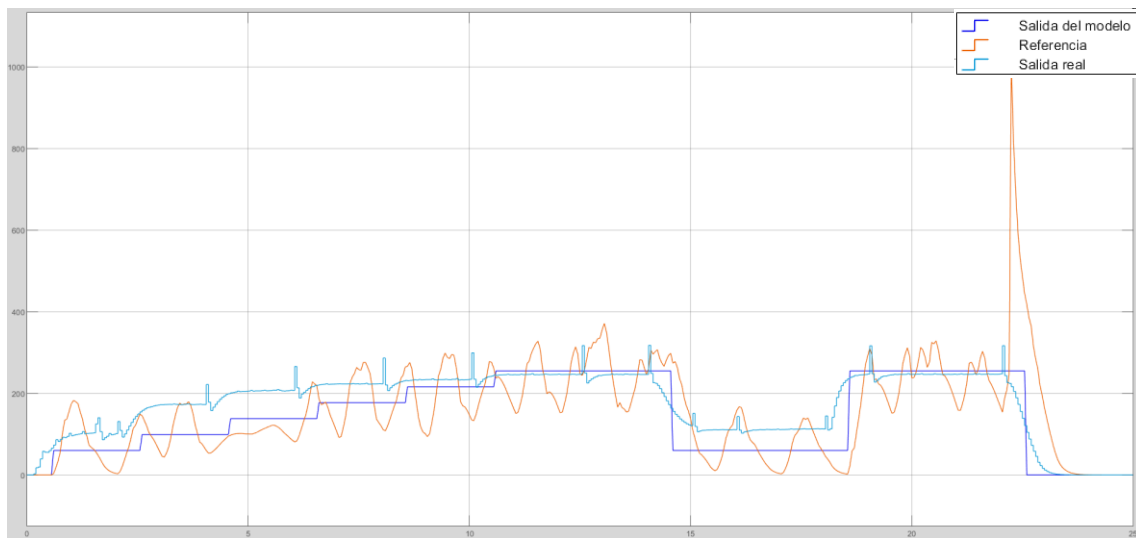


Figura 69 d)

Figura 69: a) Respuesta real del Motor DC regulado con PID Clásico, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de velocidad de Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), b) Respuesta real del Motor DC regulado con PID Fuzzy, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de velocidad de Box-Jenkins (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), c) Respuesta real del Motor DC regulado con PID Clásico, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de velocidad en Espacio de Estados (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino), d) Respuesta real del Motor DC regulado con PID Fuzzy, partiendo de las constantes K_P , K_I y K_D obtenidas a partir de las pruebas con el modelo de velocidad en Espacio de Estados (naranja). Esta respuesta se compara a aquella del motor sin regulación (cian) sometido en ambos casos a la misma entrada (azul marino)

Se observa que los modelos no logran mejorar la respuesta del sistema de manera significativa, ya fueran creados con el modelo Box-Jenkins o Espacio de Estados. Sin embargo, los controladores basados en Espacio de Estados se comportan ligeramente mejor. Por lo tanto, se realizará un “tuning” fino a partir del PID Clásico en Espacio de Estados y posteriormente lo usaremos para ajustar el PID Fuzzy. Una vez realizadas las pruebas necesarias se ajustaron los parámetros de los controladores para lograr un seguimiento lo más preciso posible de la consigna. En primera instancia se sometió al motor a señales escalonadas y con frenazos para poder hacer un “tuning” fino de los controladores, adquiriéndose las siguientes respuestas (*Figura 70*):

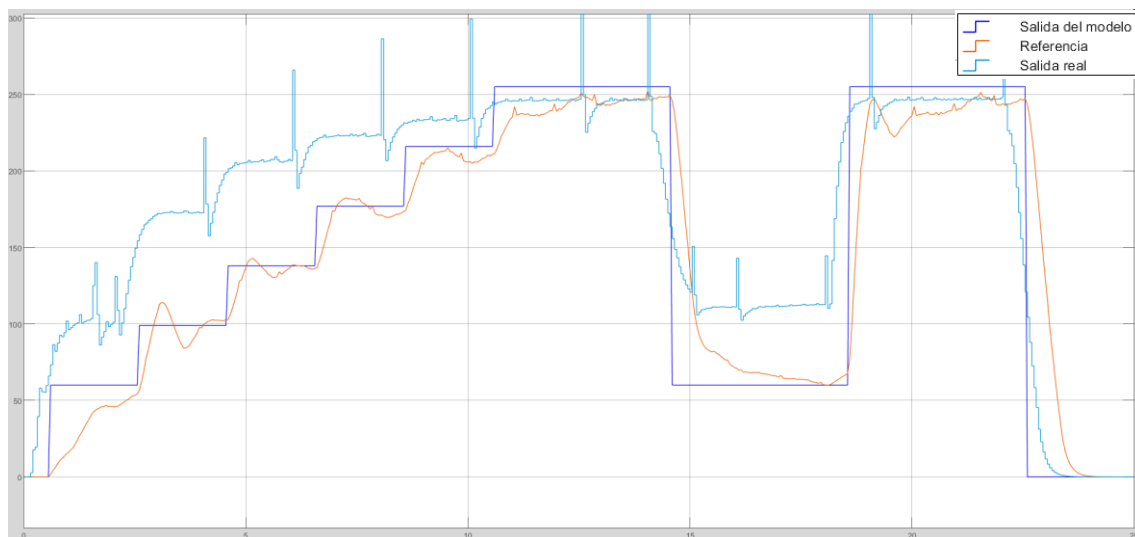


Figura 70: Respuesta de velocidad del motor regulado con Controlador PID Fuzzy (naranja) comparado con la salida ofrecida por el mismo sin regulación alguna (cian) sometido en ambas ocasiones a la misma consigna (azul marino) ajustando los parámetros del controlador a partir del creado para actuar sobre el modelo en Espacio de Estados

Se ve una grandísima mejora entre el comportamiento del sistema sin controlador y aquella con él, aunque se aprecia que no es capaz de alcanzar el valor máximo de velocidad con precisión. Analizando los experimentos realizados, lo más probable es que las baterías estuvieran a media carga, lo que no permitió alcanzar la corriente requerida. Sin embargo, lo verdaderamente importante es que el sobreimpulso máximo no es muy acusado, con lo que nos aseguramos de que el motor no irá a trompicones cuando le ordenemos seguir una consigna de velocidad determinada. Los valores de las constantes del controlador, una vez ajustadas, son:

$$K_P = 0.3, K_I = 0.4, K_D = 0.001$$

En base a ello se creó un controlador PID Fuzzy siguiendo la misma casuística que en el modelo de posición (Figura 71):

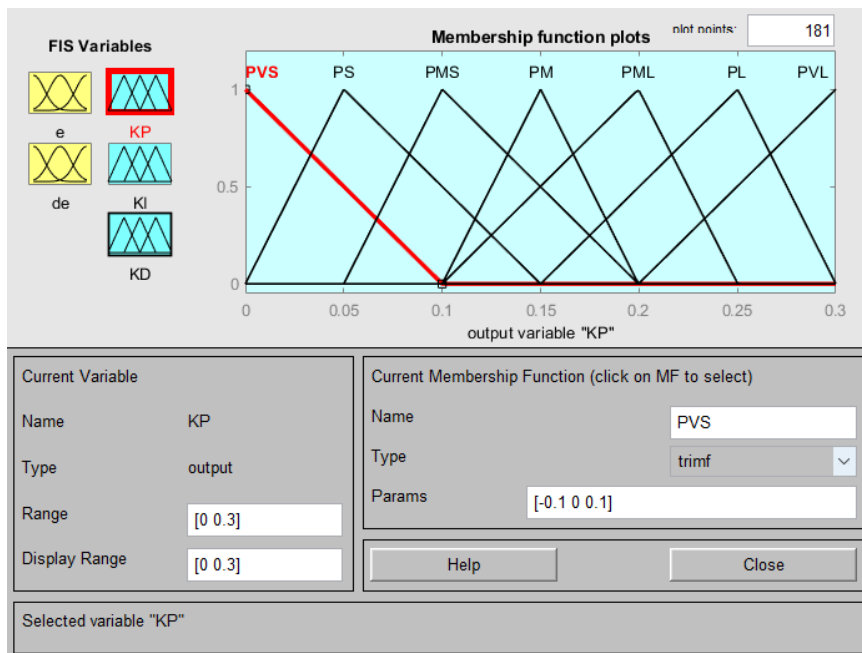


Figura 71 a)

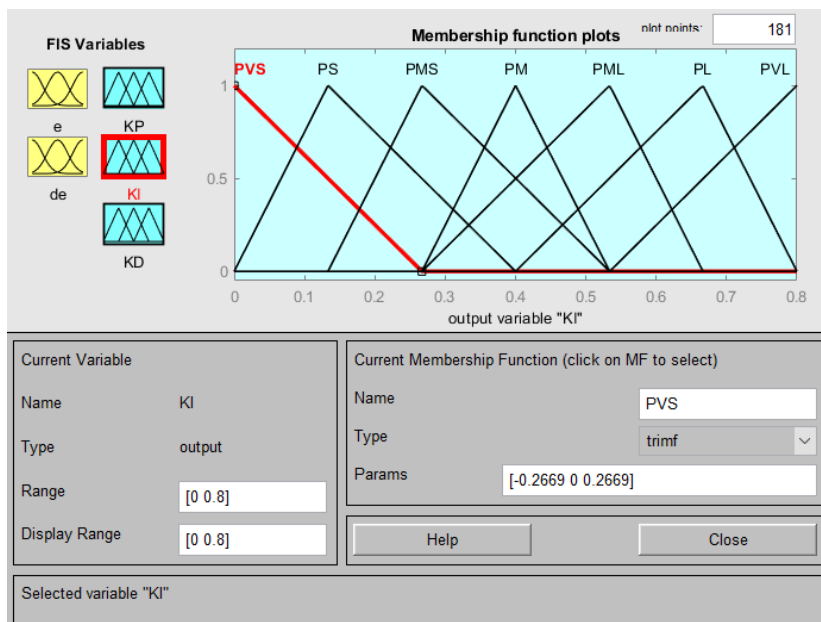


Figura 71 b)

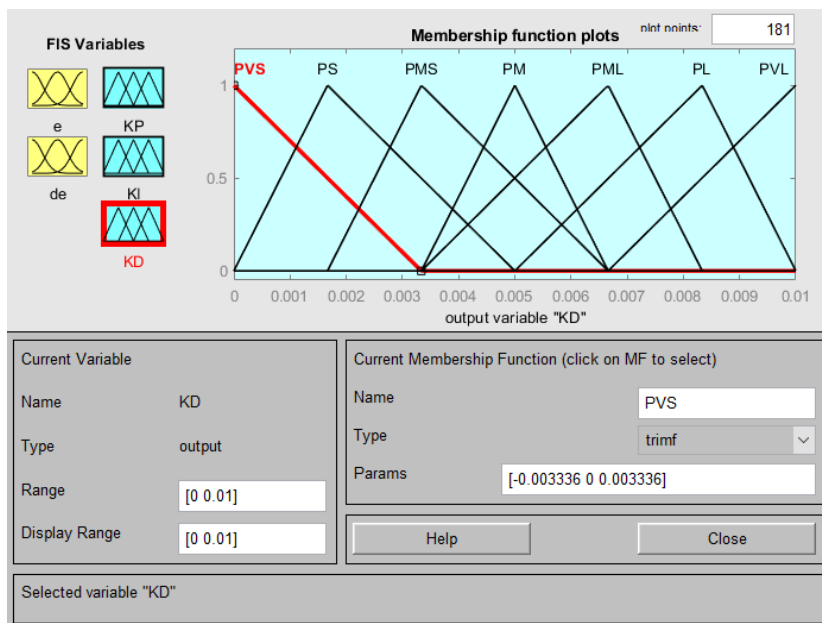


Figura 71 c)

Figura 71: a) Primera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KP, b) Segunda salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KI, c) Tercera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KD

Sometiendo de nuevo al sistema real a una señal práctica pudo observarse la siguiente respuesta (Figura 72):

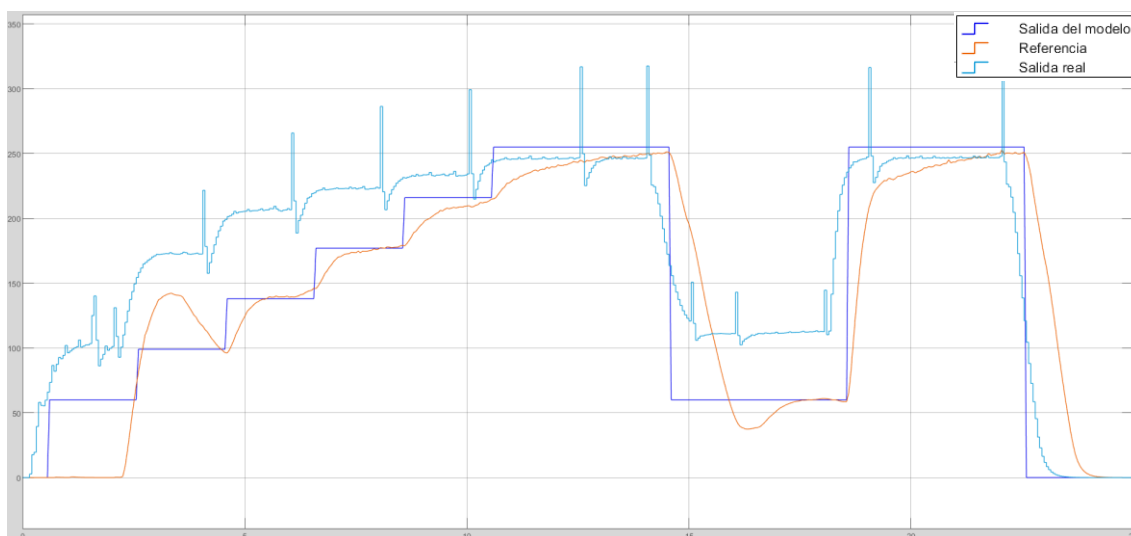


Figura 72: Salida del motor DC regulado a través del Controlador PID Fuzzy previamente expuesto (naranja) en comparación con la respuesta sin controlador (cian) sometido en ambas ocasiones a la misma consigna (azul marino)

Se ve la superioridad del controlador PID Fuzzy, al otorgar una salida más estable y precisa que el PID Clásico, gracias a su adaptabilidad ante las distintas consignas y situaciones que se puedan presentar. Una vez ajustado el modelo se midieron sus parámetros en régimen transitorio y el error en régimen permanente para comprobar que, en efecto, el afinado de los parámetros de los controladores ha acabado resultando satisfactorio y comparar, con ello, el PID Clásico con el Fuzzy. Los resultados arrojados por esta prueba se pueden observar en la *Figura 73* y en la *Tabla 10* (de ahora en adelante se utilizará la misma nomenclatura para designar los parámetros en régimen permanente y transitorio de las tablas de resultados, con “ts” para tiempo de subida, “te” para tiempo de establecimiento, “Mp” para sobreimpulso máximo, y “ess” para error en régimen permanente). Los resultados se muestran en la *Figura 73*:

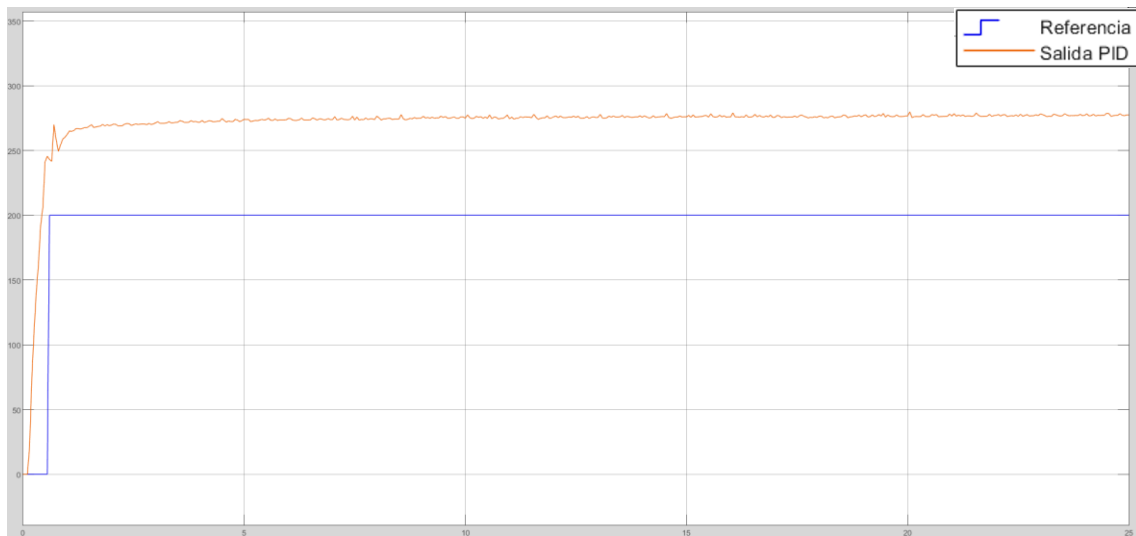


Figura 73 a)

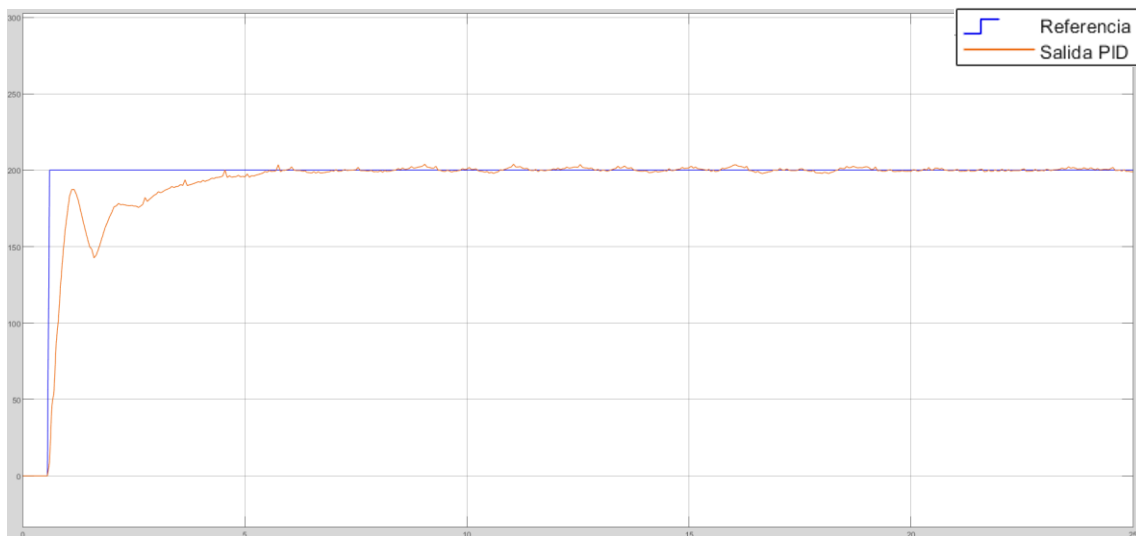


Figura 73 b)

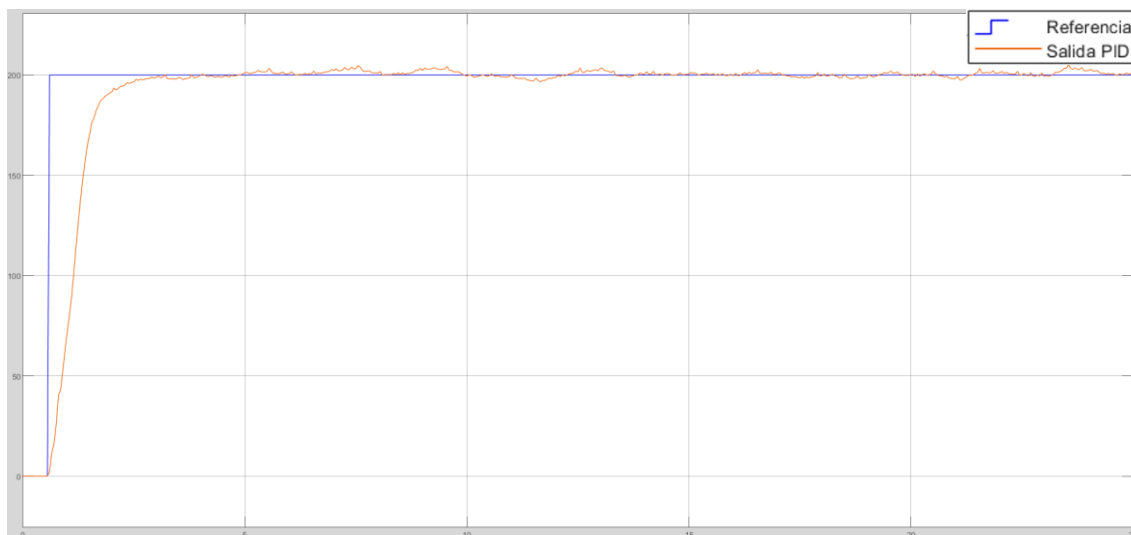


Figura 73 c)

Figura 73: a) Respuesta de velocidad del motor sin controlador (naranja) ante una entrada escalón (azul marino), b) Respuesta de velocidad del motor con Controlador PID Clásico (naranja) ante una entrada escalón (azul marino), c) Respuesta de velocidad del motor con Controlador PID Fuzzy (naranja) ante una entrada escalón (azul marino)

Tabla 10: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy

Sistema	ts (5%-95%)	te($\pm 5\%$ en ss)	Mp (%)	Ess
Sin controlador	0,558s	0,697s	1,051%	76rpm
PID	2,92s	3,535s	1,95%	≈ 0 rpm
PID Fuzzy	1,247s	1,895s	2,3%	≈ 0 rpm

Se pueden extraer varias conclusiones con estos resultados: primero, que la velocidad del motor máxima no es de 255rpm en estas condiciones de trabajo, sino que probablemente la carga de las baterías solo permitía llegar a ese valor cuando se realizaron las pruebas iniciales, de manera que es capaz de alcanzar, al menos justo después de recargar los elementos de descarga, unas 303rpm como máximo; segundo, afirmamos que el motor necesita algún tipo de controlador para reducir el error en régimen permanente, que es lo que buscamos (que siga una consigna con precisión) y tercero, que el controlador PID Fuzzy tiene en la realidad unos tiempos transitorios menores a los del PID clásico, un sobreimpulso mayor y menor oscilación, por lo que en cómputo global es preferible emplear un método de control borroso sobre uno clásico.

4.2. Modelo con perturbaciones: Motor libre

Este apartado se enfoca en el estudio del funcionamiento del motor con perturbaciones, que se incluyeron acoplando las ruedas del vehículo a escala a una barra de metal, cuyo eje de revolución coincide con el de un engranaje que es solidario al del motor, siendo este último el responsable de su giro. En primera instancia se hizo girar al dispositivo en libre, es decir, suspendido, sin la acción ni reactiva ni de rozamiento del suelo, solo el peso de las ruedas, el eje y su engranaje, y el rozamiento estático y dinámico de los rodamientos que permiten su giro. Debido a la introducción de tantas variables dinámicas fue necesario de nuevo ajustar los controladores para lograr una regulación óptima del conjunto, partiendo de los empleados con el motor en libre y sin perturbaciones:

4.2.1. Modelo con perturbaciones: Motor libre: Posición

Se muestran a continuación los resultados del motor ante una consigna de posición, regulado con sendos Controladores PID Clásico y Fuzzy, antes y después de ajustarlos para que su funcionamiento fuese el más óptimo posible (*Figura 74*):

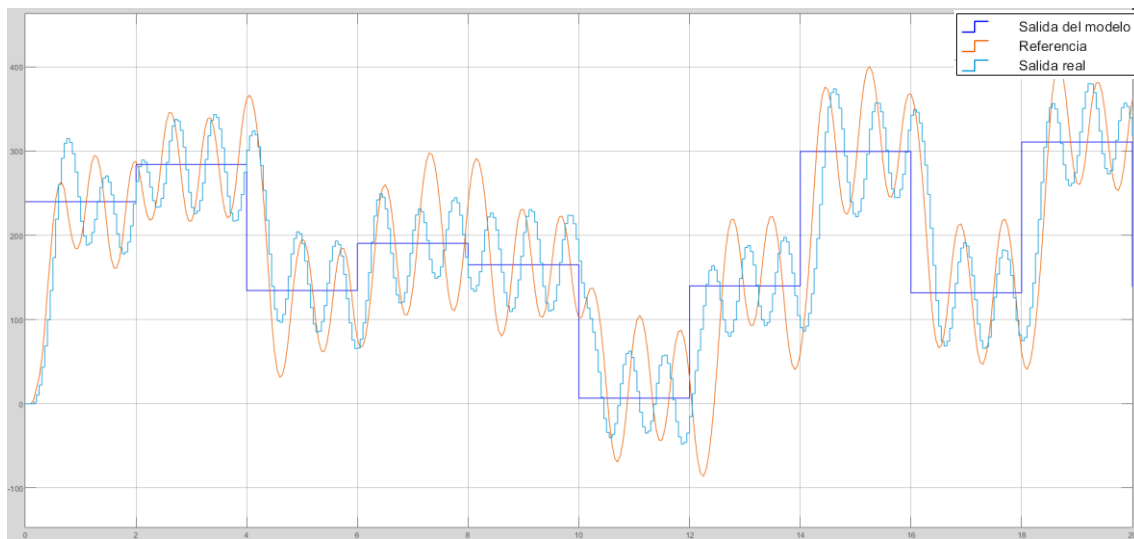


Figura 74 a)

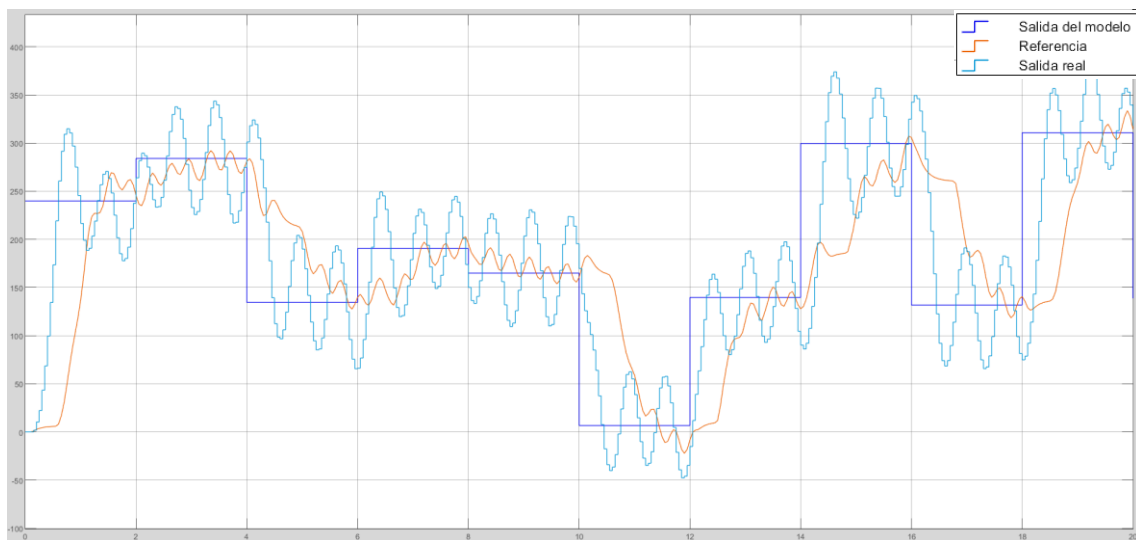


Figura 74 b)

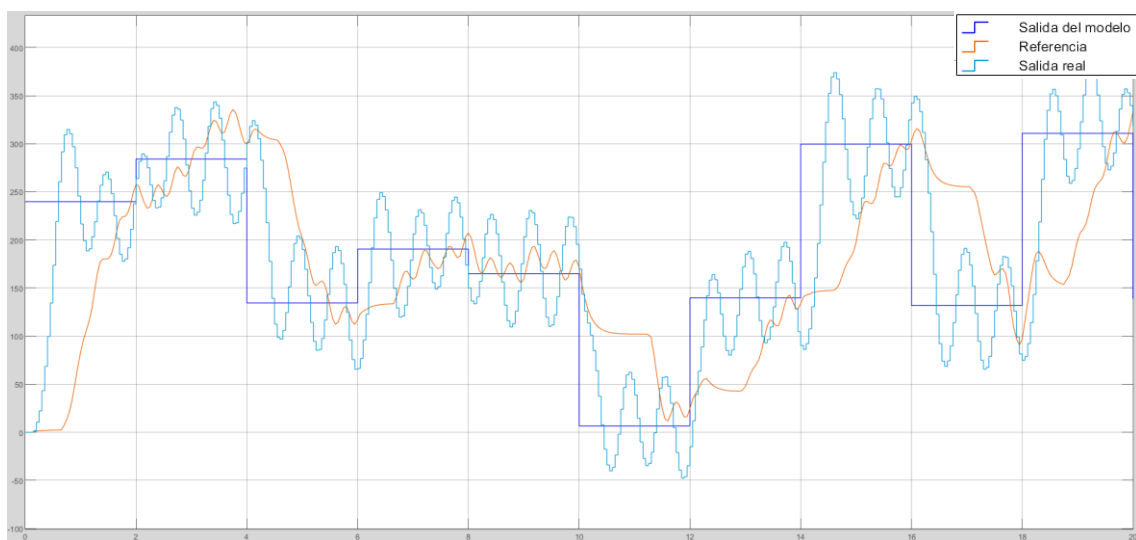


Figura 74 c)

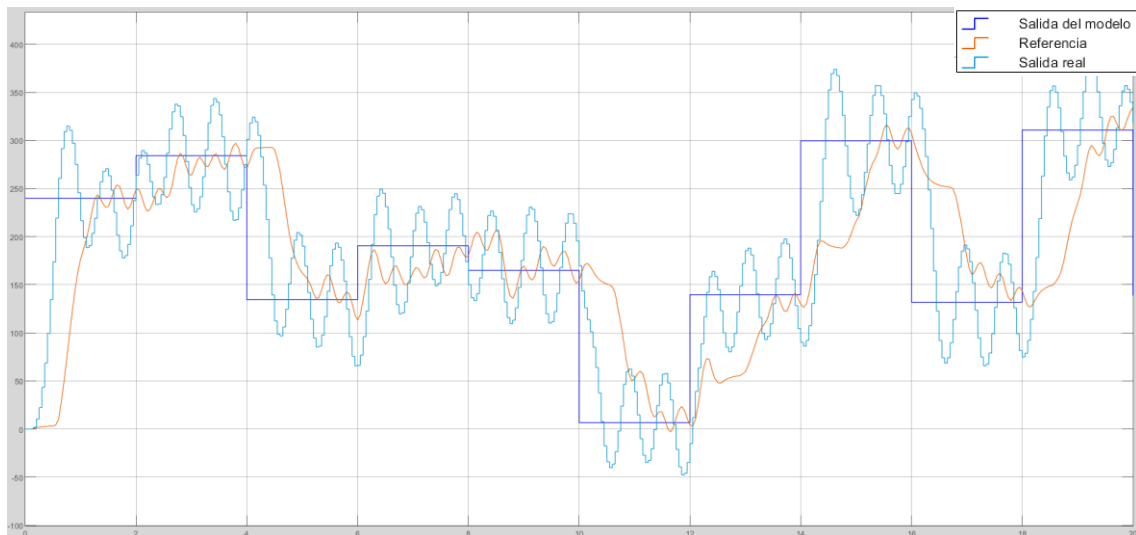


Figura 74 d)

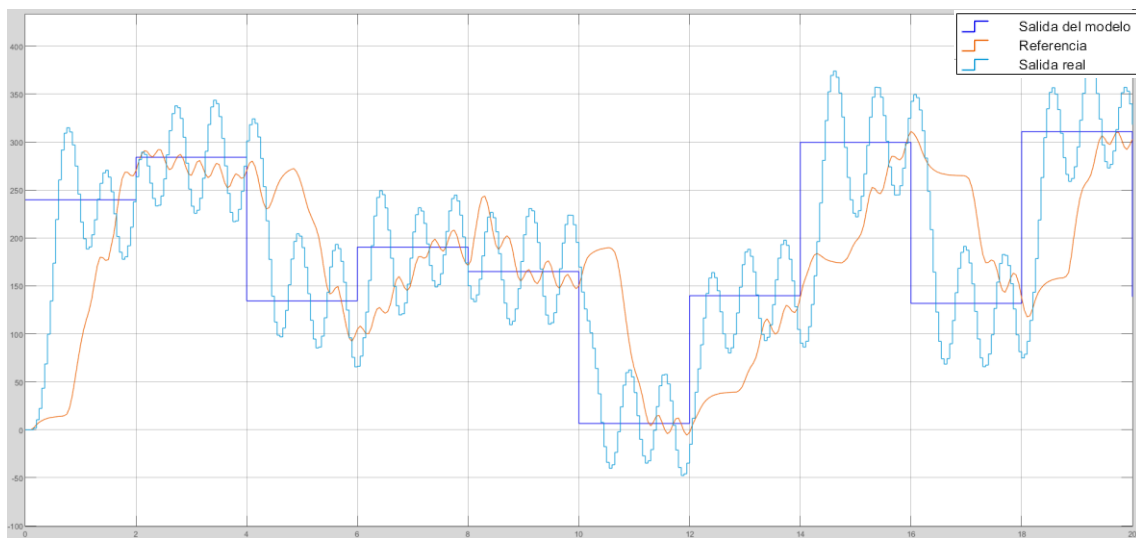


Figura 74 e)

Figura 74: a) Respuesta del motor con perturbaciones sin controlador (naranja) comparado con la del mismo sin perturbaciones (cian) ante la misma consigna de posición (azul marino), b) Respuesta del motor con perturbaciones con Controlador PID Clásico (naranja) comparado con la del mismo sin perturbaciones (cian) ante la misma consigna de posición (azul marino) antes de realizar el reajuste del controlador, c) Respuesta del motor con perturbaciones con Controlador PID Fuzzy (naranja) comparado con la del mismo sin perturbaciones (cian) ante la misma consigna de posición (azul marino) antes de realizar el reajuste del controlador, d) Respuesta del motor con perturbaciones con Controlador PID Clásico (naranja) comparado con la del mismo sin perturbaciones (cian) ante la misma consigna de posición (azul marino) después de realizar el reajuste del controlador, e) Respuesta del motor con perturbaciones con Controlador PID Fuzzy (naranja) comparado con la del mismo sin perturbaciones (cian) ante la misma consigna de posición (azul marino) después de realizar el reajuste del controlador

Se ve una notable mejoría en el comportamiento del motor tras volver a ajustar las constantes de los controladores. Las pruebas en el motor se realizaron con la implementación de los mismos modelos en Simulink que en apartados anteriores. Se aprecia que la instauración del margen dinámico de tolerancia (proporcional a la diferencia entre consigna y valor real de posición angular) en vez de los $\pm 5^\circ$ fijos asegura que el motor responderá más rápidamente ante cambios grandes de la señal de referencia. Los parámetros elegidos para el controlador PID clásico fueron:

$$K_P = 0,1375; K_I = 1,4; K_D = 0,14$$

Y el controlador PID Fuzzy ideado a partir de este fue (Figura 75):

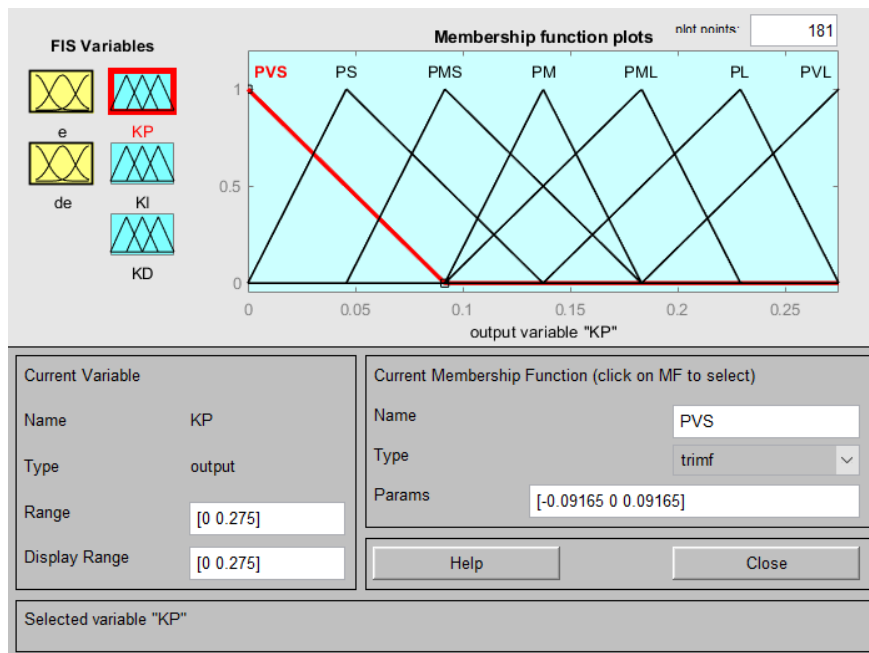


Figura 75 a)

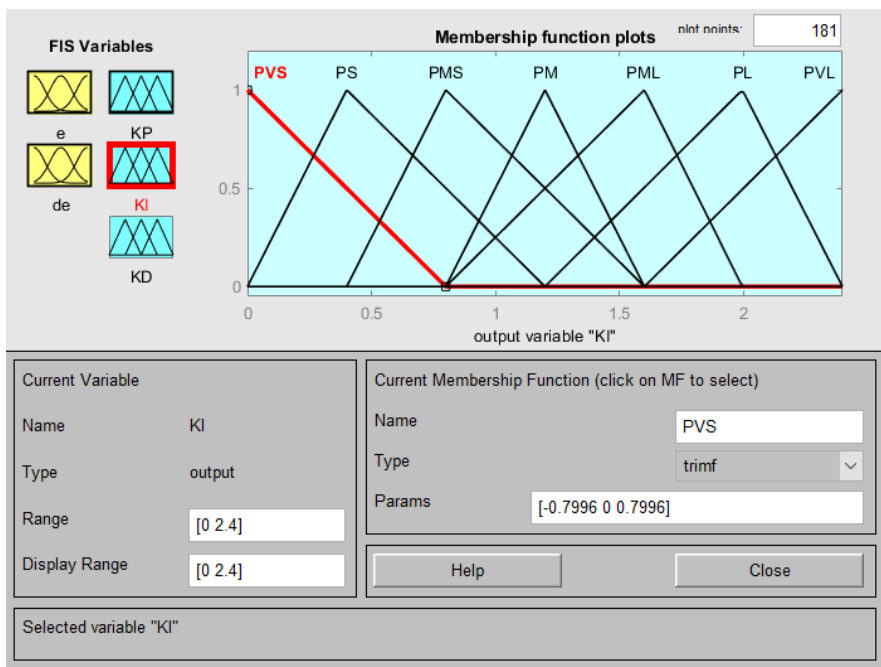


Figura 75 b)

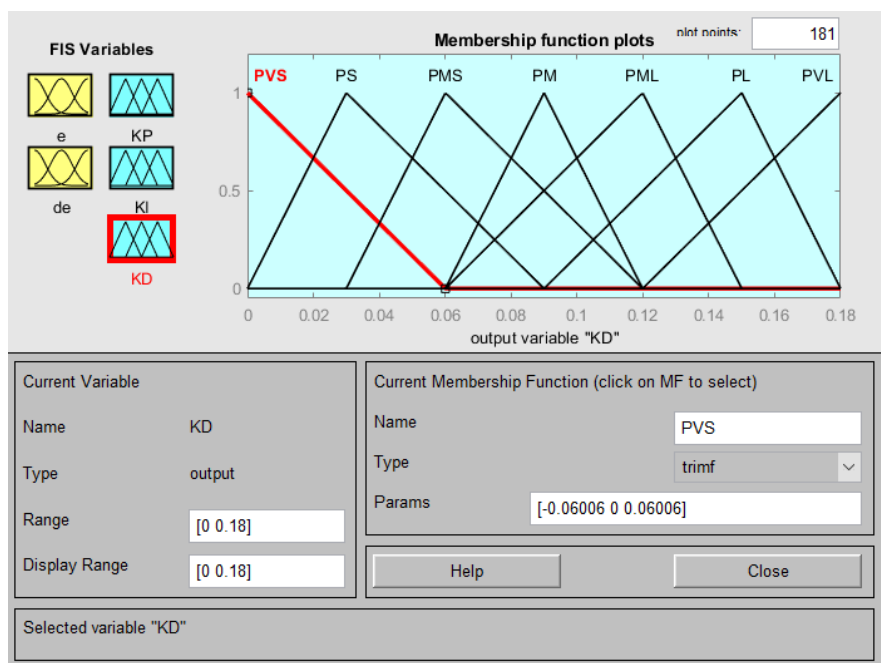


Figura 75 c)

Figura 75: a) Primera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KP, b) Segunda salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KI, c) Tercera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KD

Volvió a someterse al sistema a un escalón de 270° una vez más, y se obtuvieron los siguientes parámetros (*Tabla 11*):

Tabla 11: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy del sistema con perturbaciones

Sistema	ts (5%-95%)	te	Mp (%)	ess
Sin controlador	0,361s	0,746s	36,852%	≈0rpm
PID	0,853s	4,609s	18,222%	≈0rpm
PID Fuzzy	0,889s	5,214s	14,815%	≈0rpm

Atendiendo a la *Tabla 9* podemos realizar una comparativa entre los resultados arrojados por el motor sin perturbaciones y estos cuando las tiene. Se ve que debido a la presencia de perturbaciones el motor tiene un tiempo de subida mayor y un sobreimpulso máximo más acusado que en el caso sin perturbaciones. Ello es algo lógico puesto que ahora existen fuerzas de oposición que antes no estaban presentes (o al menos no de forma tan acusada). No obstante, como el motor tiene que contrarrestar pares de fuerzas contrapuestos (el del motor contra el de rozamiento) estos ayudan a que alcance la consigna con mayor precisión al ir más lento de lo habitual.

4.2.2. Modelo con perturbaciones: Motor libre: Velocidad

Repetimos el mismo proceso de análisis con la consigna de velocidad, y vemos los resultados representados en la *Figura 76*:

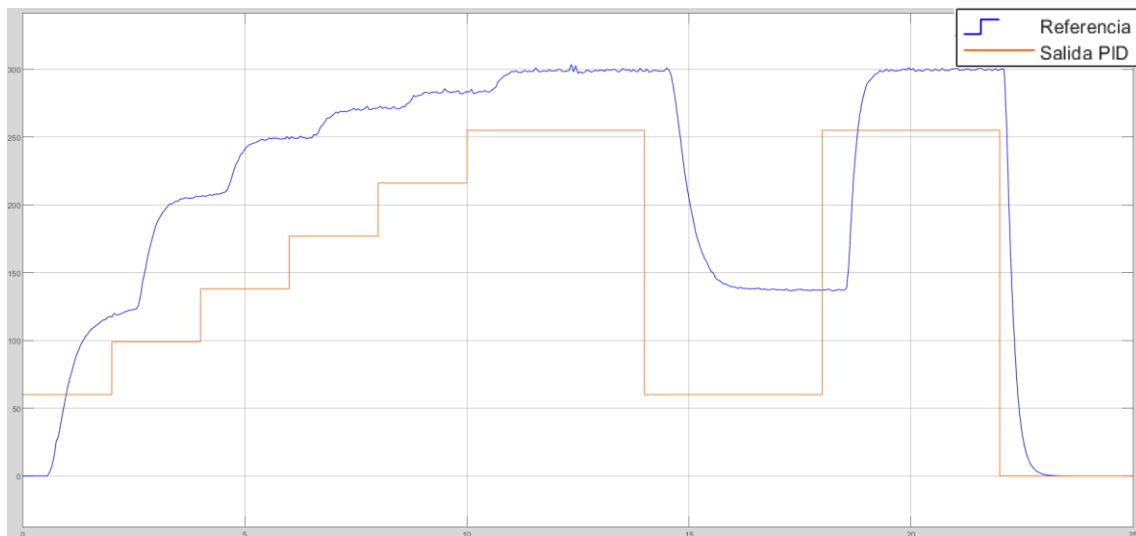


Figura 76 a)

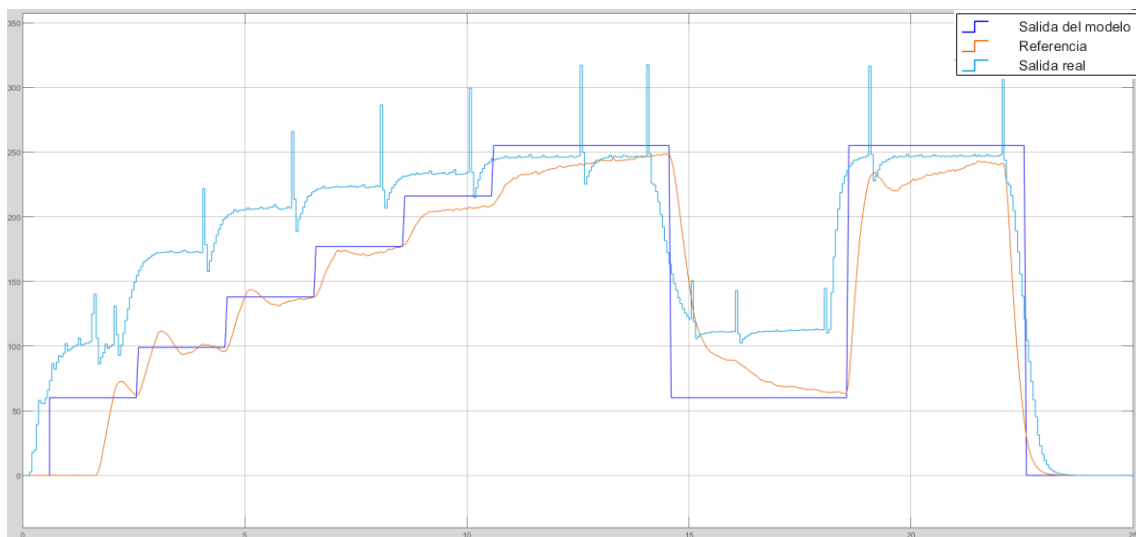


Figura 76 b)

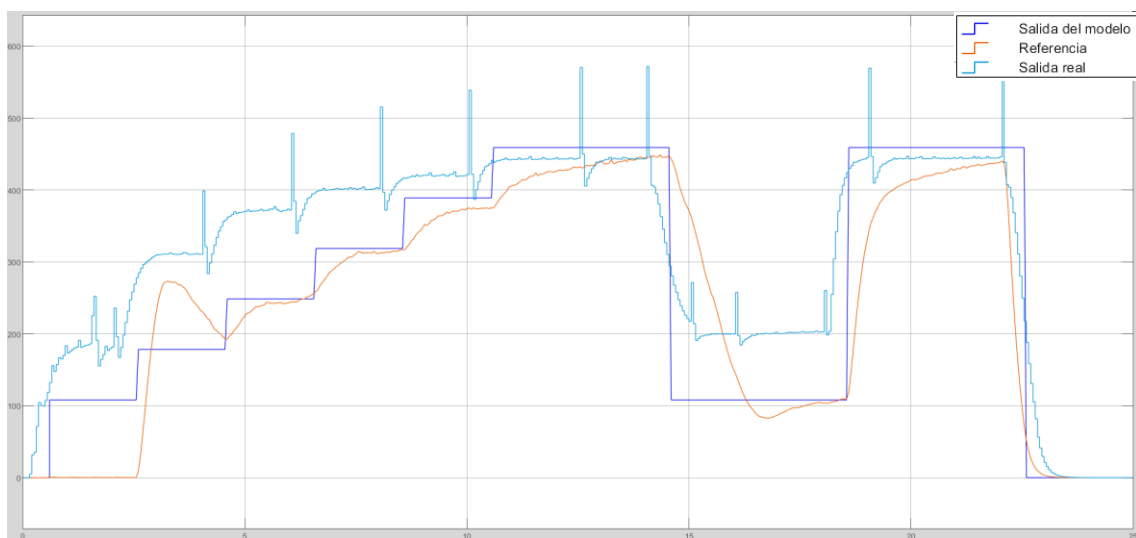


Figura 76 c)

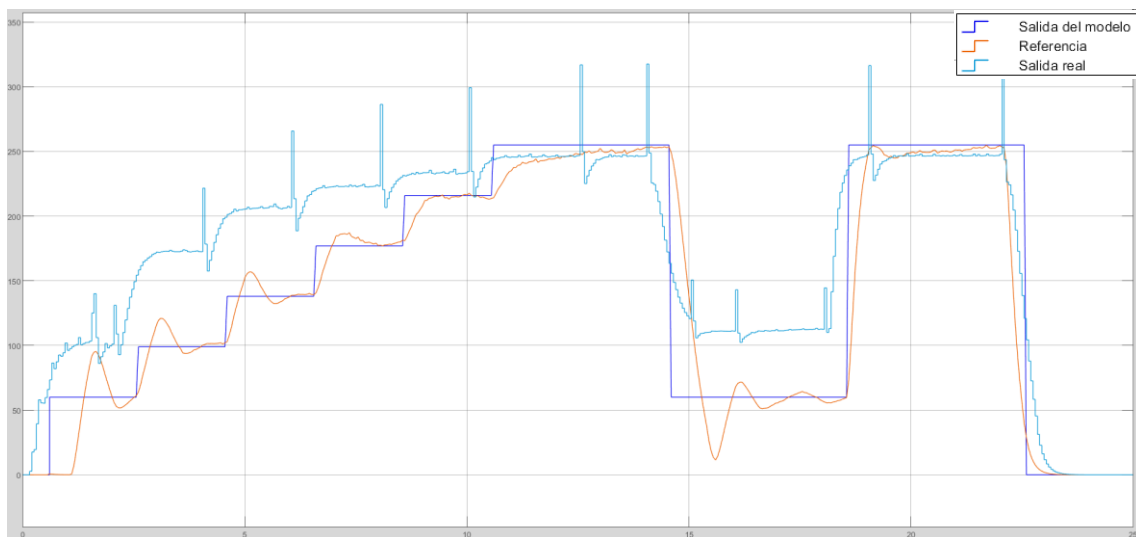


Figura 76 d)

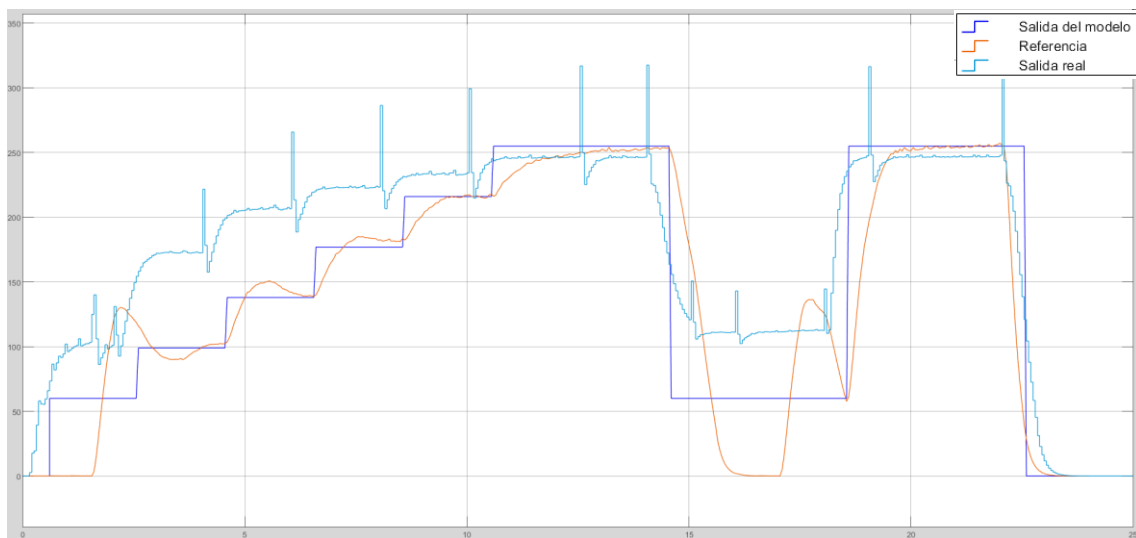


Figura 76 e)

Figura 76: a) Respuesta del motor con perturbaciones sin controlador (azul marino) ante una consigna de velocidad determinada (azul marino), b) Respuesta del motor con perturbaciones con Controlador PID Clásico (naranja) comparado con aquella del motor sin controlador (cian) ante una consigna de velocidad determinada (azul marino) antes del ajuste, c) Respuesta del motor con perturbaciones con Controlador PID Fuzzy (naranja) comparado con aquella del motor sin controlador (cian) ante una consigna de velocidad determinada (azul marino) antes del ajuste, d) Respuesta del motor con perturbaciones con Controlador PID Clásico (naranja) comparado con aquella del motor sin controlador (cian) ante una consigna de velocidad determinada (azul marino) una vez realizado el ajuste, e) Respuesta del motor con perturbaciones con Controlador PID Fuzzy (naranja) comparado con aquella del motor sin controlador (cian) ante una consigna de velocidad determinada (azul marino) una vez realizado el ajuste

De nuevo, se aprecia una notable mejora en el sistema una vez ajustados los parámetros de controlador. Una vez más, se comenzó por un ajuste de las constantes del

PID clásico para poder tener un punto de partida al ajustar los parámetros del diseño del PID Fuzzy, y obtuvimos los siguientes valores de constantes en este primer controlador:

$$K_P = 0,3; K_I = 0,7; K_D = 0,005$$

Se aprecia que estos valores se asemejan a aquellos obtenidos en el diseño del Controlador PID Clásico para el motor real sin perturbaciones, por lo que deducimos que la modificación de estos parámetros no ejerce una influencia tan drástica en el modelo de velocidad como lo hace en el de posición. Partiendo de estas constantes se creó el controlador PID Fuzzy mostrado a continuación, de nuevo modificando solo los rangos de actuación en las salidas KP, KI, y KD (Figura 77):

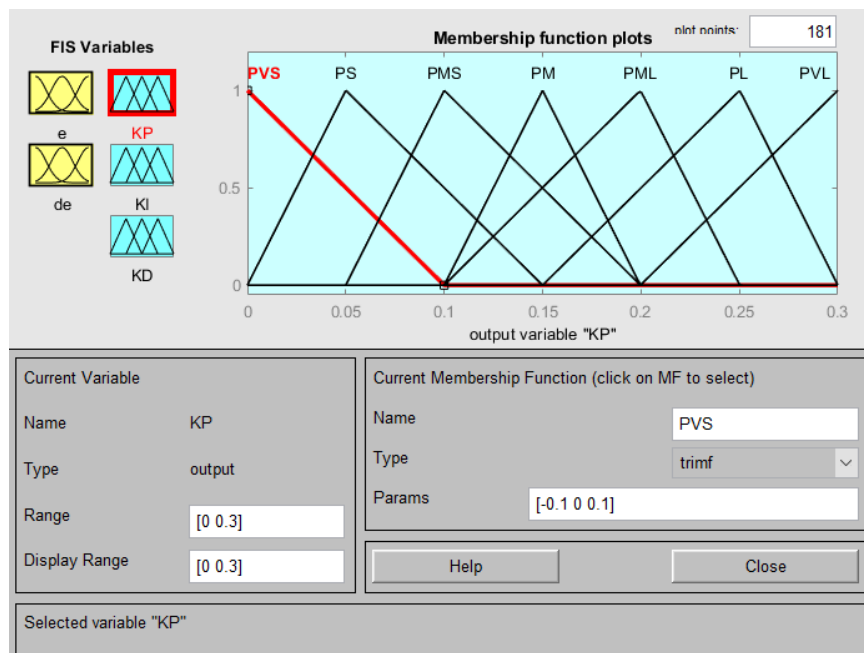


Figura 77 a)

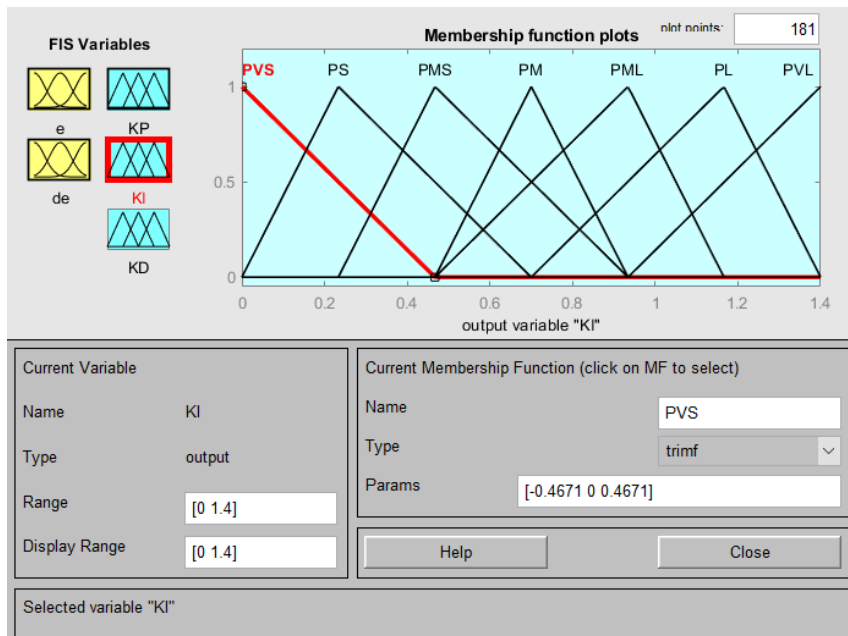


Figura 77 b)

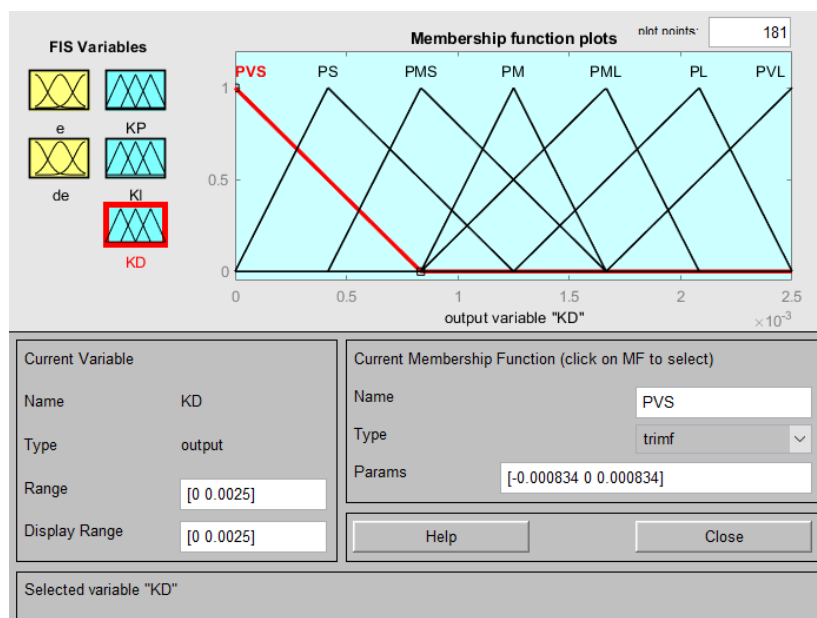


Figura 77 c)

Figura 77: a) Primera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KP, b) Segunda salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KI, c) Tercera salida del Controlador PID Fuzzy de velocidad diseñado a partir de las constantes del PID Clásico que acaba de exponerse: KD

De nuevo se sometió al sistema real a un escalón de 200rpm y se midieron los siguientes valores de los parámetros en régimen transitorio y permanente (*Tabla 12*):

Tabla 12: Parámetros en régimen transitorio y permanente del sistema ante la misma consigna sin controlador, con PID Clásico y con PID Fuzzy del sistema con perturbaciones

Sistema	ts (5%-95%)	te($\pm 5\%$ en ss)	Mp (%)	Ess
Sin controlador	0,558s	0,697s	1,051%	76rpm
PID	2,92s	3,535s	1,95%	≈ 0 rpm
PID Fuzzy	1,247s	1,895s	2,3%	≈ 0 rpm

Realizamos la comparativa con la *Tabla 10*, referente a los parámetros del motor sin perturbaciones: al igual que en el caso de la posición, las perturbaciones frenan al motor, aumentando, como es lógico, los tiempos transitorios y el sobreimpulso máximo, aunque el aumento de este sobreimpulso no es demasiado grave si contrastamos con lo obtenido con el PID Clásico. Es destacable el hecho de que con el controlador PID clásico el tiempo de establecimiento es menor, ya que la menor velocidad seguramente permita un ajuste más fino con parámetros permanentes, pero en el PID Fuzzy la adaptabilidad es mejor, de forma que el tiempo de subida disminuye.

5. Conclusiones y trabajos futuros

En la realización de este documento varios ingredientes fueron necesarios: en primera instancia el afán de descubrimiento, puesto que con cada paso que se daba se averiguaba algo nuevo sobre las limitaciones y características del material con que se trabajaba, ya fuese el hecho de que el motor era capaz de trabajar a una velocidad mayor a la otorgada por el fabricante, el que la imposición de una tolerancia dinámica sobre la consigna de posición permitía una velocidad de respuesta mayor y una reducción de las oscilaciones más drástica sobre la tolerancia estática inicial, o el lograr comparar modelos y ajustar parámetros de los controladores diseñados para conseguir el ajuste más fino posible del dispositivo con o sin perturbaciones. Fue precisamente esta fuerza motriz la que impulsó el avance de las sucesivas pruebas y experimentos realizados, ya que el viaje experimental acabó adquiriendo un matiz emocionante al dar cabida a la implementación de soluciones creativas para abordar problemas prácticos.

El segundo ingrediente fue la paciencia, ya que la obtención de resultados concluyentes no fue algo inmediato: fue necesaria la realización de innumerables pruebas y la convivencia con el fracaso se convirtió en algo habitual pero incentivador, ya que es algo inherente al alma del ingeniero el resolver quebraderos de cabeza mediante la utilización de la imaginación.

El tercer ingrediente fue la visión de una meta que con cada avance se materializaba de forma más y más tangible, y esta es la creencia de que las conclusiones, modelos y experiencias reunidas, corregidas y agrupadas en este trabajo pueden servir de soporte en el futuro a científicos, diseñadores e ingenieros para crear técnicas de control inteligente que poder implementar en el coloquialmente llamado “coche del futuro”.

A lo largo de este trabajo se abordaron numerosos temas relevantes que acabaron confluyendo en el resultado final, que es la creación de un controlador PID Fuzzy lo suficientemente versátil y eficiente como para poder cambiar radicalmente, y para mejor, el comportamiento de un motor de corriente continua de bajas prestaciones. En primera instancia se estudiaron y definieron los límites de funcionamiento del dispositivo para poder averiguar con qué estábamos trabajando, una vez hecho esto se crearon modelos de posición y velocidad de Box-Jenkins y Espacio de Estados, para poder emular el comportamiento del motor con la mayor fidelidad posible, comparando cuál de ellos se aproximaba más a la realidad y tomando los Controladores PID Clásico y Fuzzy de aquel modelo que se acercaba más a aquello con los que trabajamos en el mundo físico, tanto en posición como en velocidad. A continuación se implementaron y ajustaron estos controladores en el motor real sin perturbaciones, y después se reajustaron para aplicarlo en el dispositivo libre con perturbaciones, obteniendo gratos resultados.

Si bien el camino que ha conllevado hacer este documento es largo pero gratificante, hay ciertos objetivos que por falta de tiempo no lograron cumplirse, que son:

-En primera instancia implementar los controladores en un vehículo libre sobre el suelo, sin las constricciones que conlleva el empleo de conectores de cobre, pudiendo ser sustituidos estos por comunicaciones inalámbricas.

-En segundo lugar, y una vez diseñada e implementada la aparatada asociada al control inalámbrico se habría querido ver si los controladores diseñados después del ajuste realizado de los mismos al experimentar con el motor libre con perturbaciones eran óptimas, aplicándolos sobre el motor libre rodando por el suelo.

-En último lugar, desvincular totalmente de Matlab la toma de datos, envío de señales e implementación de controladores, pues este es un software potente y práctico para realizar pruebas y simulaciones, pero no se usa en vehículos autónomos ni en un entorno ingenieril que no sea el de la investigación.

Como cierre para este trabajo podemos llegar a afirmar sin un solo atisbo de duda que los Controladores PID Fuzzy se imponen sobre los Clásicos, debido a su versatilidad, capacidad de reajuste de sus parámetros y reducción del sobreimpulso máximo en la salida, abogando por la precisión por encima de la velocidad de respuesta.

Bibliografía

1. *El País*: "Los ojos de los coches autónomos, mejores que los de los humanos". [En línea] Noviembre de 2019.
https://elpais.com/retina/2019/11/07/innovacion/1573119342_580859.html.
2. Wikipedia: "Motor de corriente continua". [En línea] Septiembre de 2022.
https://es.wikipedia.org/wiki/Motor_de_corriente_continua.
3. Portal Educativo de la Generalidad Valenciana: "Motor de Corriente Continua". [En línea] Diciembre de 2020. <https://portal.edu.gva.es/wp-content/uploads/sites/607/2020/12/2BACH-TEMA4.pdf>.
4. Puntoflotante.com: "Principios de funcionamiento del encoder cuadratura de efecto Hall para medición de velocidad en motores de corriente directa". [En línea] <https://www.puntoflotante.net/FUNCIONAMIENTO-ENCODER-CUADRATURA-EFECTO-HALL.htm>.
5. Página oficial de Arduino. [En línea] <https://www.arduino.cc/>.
6. Página web oficial de Matlab. [En línea] <https://es.mathworks.com/>.
7. Wikipedia: "Modulación por ancho de pulsos". [En línea] Enero de 2022.
https://es.wikipedia.org/wiki/Modulaci%C3%B3n_por_ancho_de_pulsos.
8. Puntoflotante.com: "Motor de CD con motorreductor, caja de engranes metálicos y encoder cuadratura. Torque de 5.2 kg.cm a 110 rpm". [En línea] <https://www.puntoflotante.net/MOTOR-CD-MOTORREDUCTOR-ENCODER.htm>.
9. Página web oficial de Arduino IO Toolbox. [En línea] https://es.mathworks.com/help/supportpkg/arduino/referencelist.html?s_eid=PSM_15028.
10. Ogata, Katsuhiko. *Ingeniería de control moderna*. s.l. : PRENTICE-HALL, 1998. ISBN 970-17-0048-1.
11. Wikipedia: "Lógica difusa". [En línea] Mayo de 2022.
https://es.wikipedia.org/wiki/L%C3%B3gica_difusa.
12. Sitio web oficial de la Toolbox de System Identification de Matlab. [En línea] <https://es.mathworks.com/products/sysid.html>.

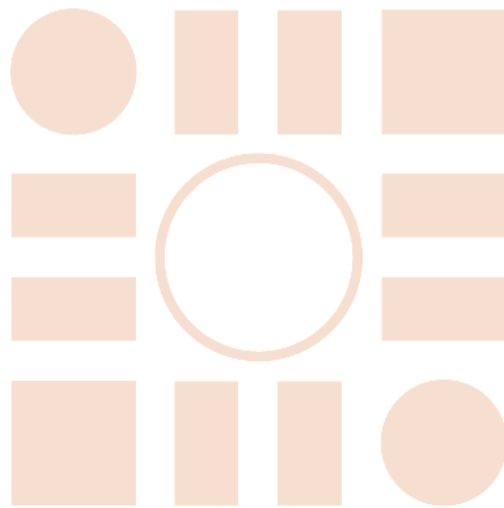
13. Wikipedia: "Espacio de Estados". [En línea] Abril de 2022.
https://es.wikipedia.org/wiki/Espacio_de_estados.

14. Wikipedia: "Box-Jenkins Method". [En línea] Julio de 2022.
https://en.wikipedia.org/wiki/Box%E2%80%93Jenkins_method.

15. Kumar Bansal, Umesh y Narvey , Rakesh. "Speed Control of DC Motor Using Fuzzy PID Controller". [En línea] 2013.
https://www.ripublication.com/aeer/070_pp%20%20%20%20%201209-1220.pdf.

16. Sitio web oficial de la Fuzzy Logic Toolbox de Matlab. [En línea]
<https://es.mathworks.com/products/fuzzy-logic.html>.

Universidad de Alcalá
Escuela Politécnica Superior



ESCUELA POLITECNICA
SUPERIOR



Universidad
de Alcalá