

Grado en Ingeniería Electrónica Industrial y Automática

Curso 2019/2020

CONTROL INTELIGENTE

Reporte 2

Control de un Motor Diesel con un sistema Fuzzy

Emanuel David Nuñez Sardinha

13 de Enero de 2020

Introducción

El objetivo de este proyecto es el ajuste de un controlador basado en lógica Fuzzy para un motor diesel simulado. Nos concentramos en el proceso de diseño del controlador, aumentando progresivamente su complejidad, iniciando con un controlador directo de una variable, modificando para construir uno integral de una entrada, culminando en un controlador integral de dos entradas. Se presenta el proceso de diseño gradual y las decisiones y factores que motivaron cambios y modificaciones. Finalmente comparamos los parámetros de nuestro controlador finalizado con su equivalente PID, y comprobamos sus relaciones teóricas.

Presentación del Problema

De la misma forma que en el proyecto anterior, se ha escogido el modelo de un motor diesel incluido en los ejemplos de Matlab. El modelo corresponde a un sistema cerrado no lineal complejo, que computa la velocidad de un motor frente a una señal respecto al tiempo (Figura 1).

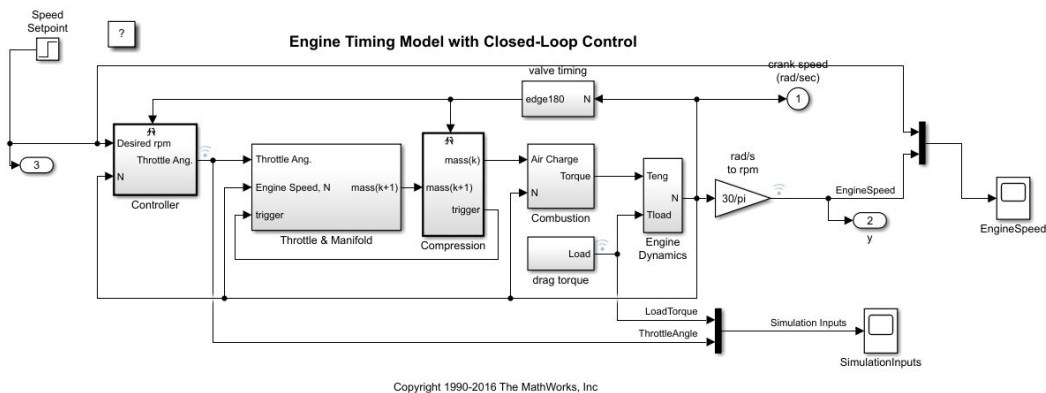


Figura 1: Planta del motor de diesel

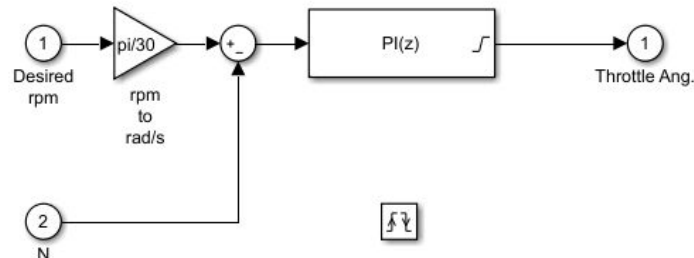


Figura 2: Controlador del sistema

Mas adelante, se implementará un sistema de dos entradas, utilizando la derivada del error. Configuramos nuestro sistema con un intervalo de tiempo fijo de 0.0001sec, garantizando respuesta consistente.

Sistema equivalente al PID

En el caso particular de controladores borrosos, podemos partir de un controlador PID existente. Sin embargo, procederemos desarrollando un controlador iterativamente, y una vez finalizado lo compararemos con su equivalente PID.

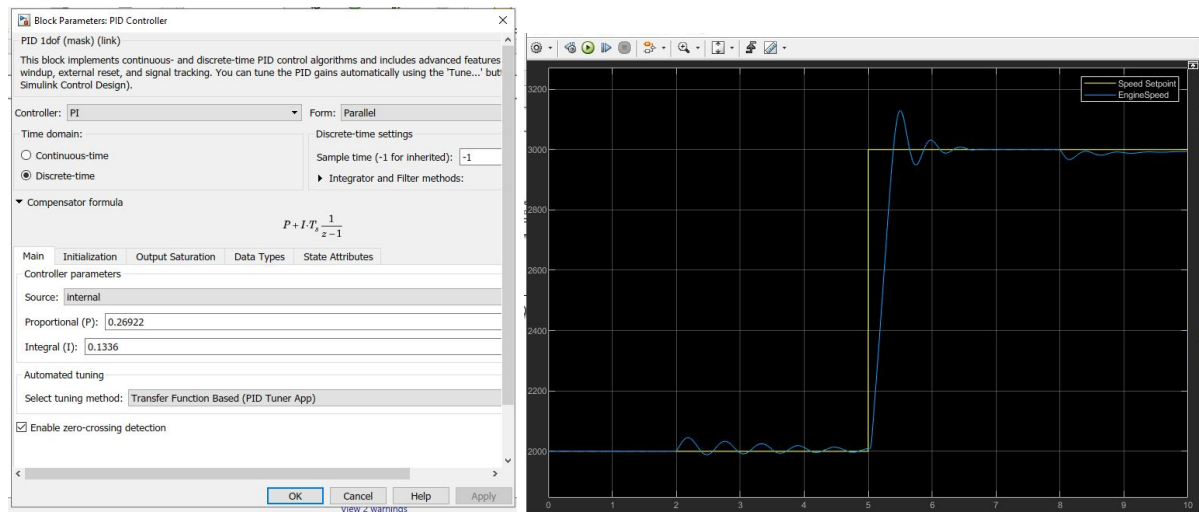


Figura 3: Controlador PI discreto y respuesta, utilizado en el proyecto 1. $K_p=0.26922$, $K_i=0.1336$.

Para el controlador previo, obtenemos la respuesta predeterminada de la figura, que utilizaremos como referencia.

Creación de Controlador

Controlador sencillo

Nuestro controlador es responsable de controlar el ángulo de apertura de la entrada de combustible para el motor, de esta forma controlando su velocidad. El ángulo de apertura puede variar entre [0-90] grados. Podemos desarrollar un set de reglas sencillas para su funcionamiento

Sustituimos nuestro controlador PI por uno borroso dentro del sub-bloque de control (Figura), y procedemos a diseñar nuestro sistema borroso. Podemos observar que en su configuración actual, el sistema cuenta con una sola entrada, diferencia entre la velocidad deseada y la actual (en rad/s), que denominamos *error*.

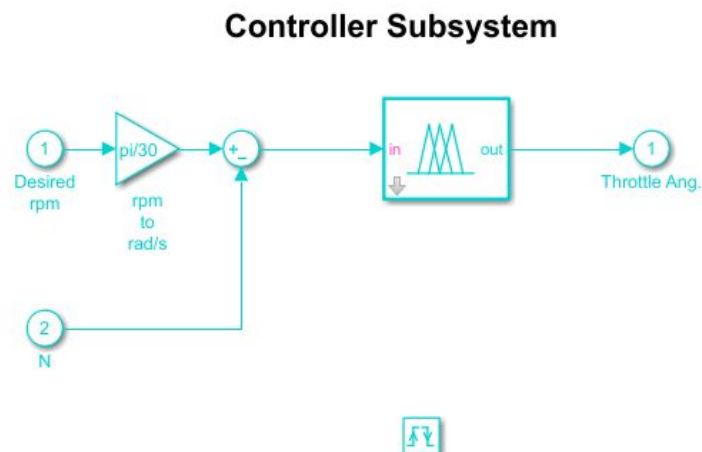


Figura 4: Controlador borroso directo.

Partiendo de un sistema Mamdani, configuramos la fuzzificación de la entrada.

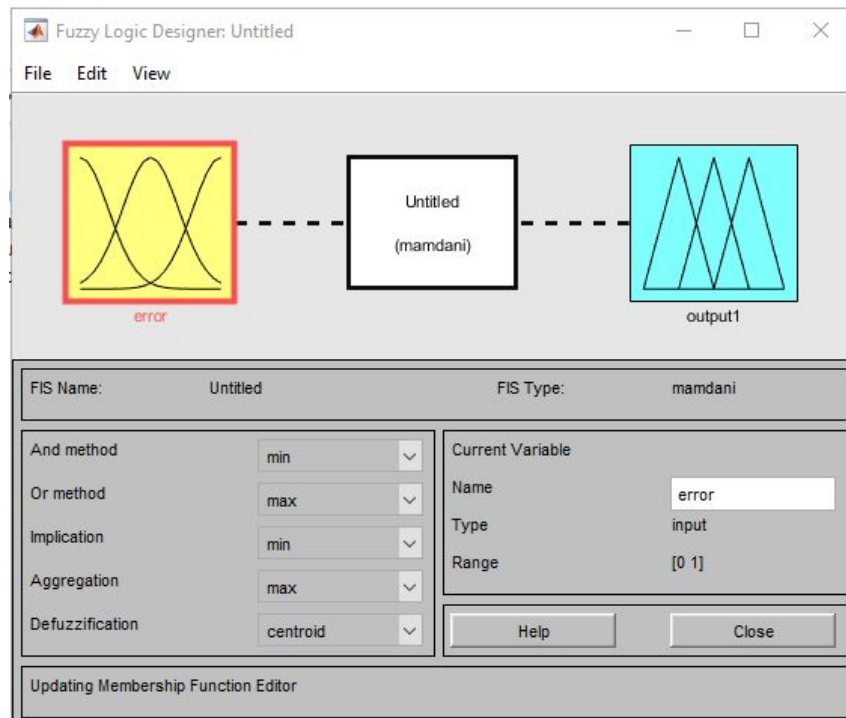


Figura 5: Sistema borroso Mamdani

Para encontrar el rango del error, podríamos calcular los valores teóricos a partir de un rango de revoluciones por minuto y convertir a rad/s, pero esto no tendríamos un rango claro de trabajo del sistema, solo de los extremos. Añadimos un *scope* a el bloque erro y observamos su comportamiento durante la ejecución normal del sistema (Figura). Observamos como en un sistema estable (a), los valores oscilan en un rango aproximado $[5, -5]$. Observamos un pico grande, que corresponde a el cambio abrupto de señal de entrada (salto de 2000 a 3000 rpms).

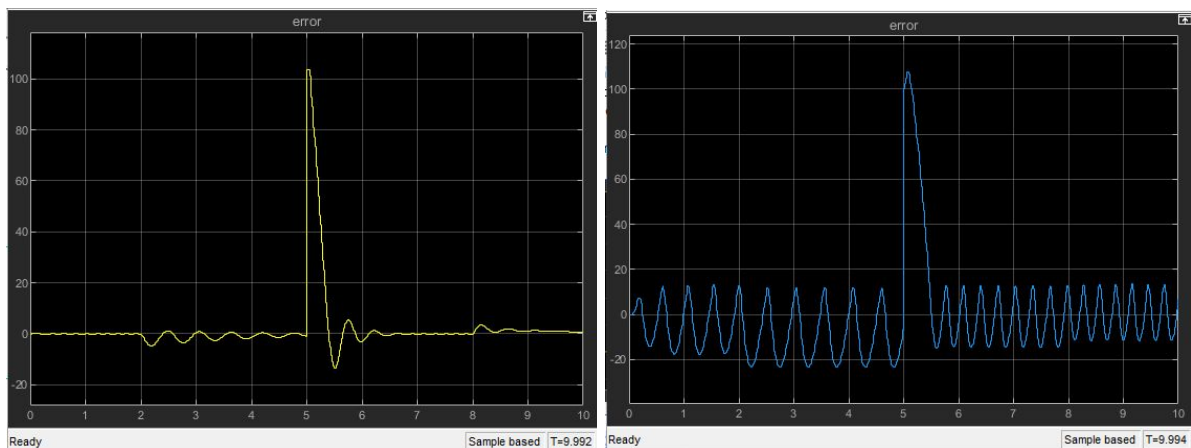


Figura 6: Error con controlador original PI. Error con controlador fuzzy sin configurar

En nuestro sistema borroso, dividimos la entrada del error en 3 categorías generales, z (zero), p (positivo) y n (negativo), para un rango de $[-5, 5]$ (Figura 7). Agregamos un bloque de saturación previo para garantizar una respuesta en este caso.

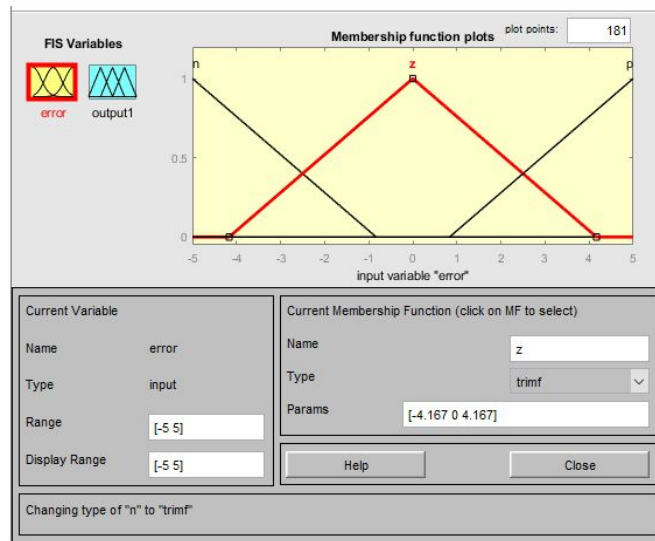


Figura 7: Categorías de entrada

En cuanto a la salida, su valor corresponde a un ángulo entre 0 y 90. Asignamos tres categorías correspondientes: menor, medio y mayor (Figura 8).

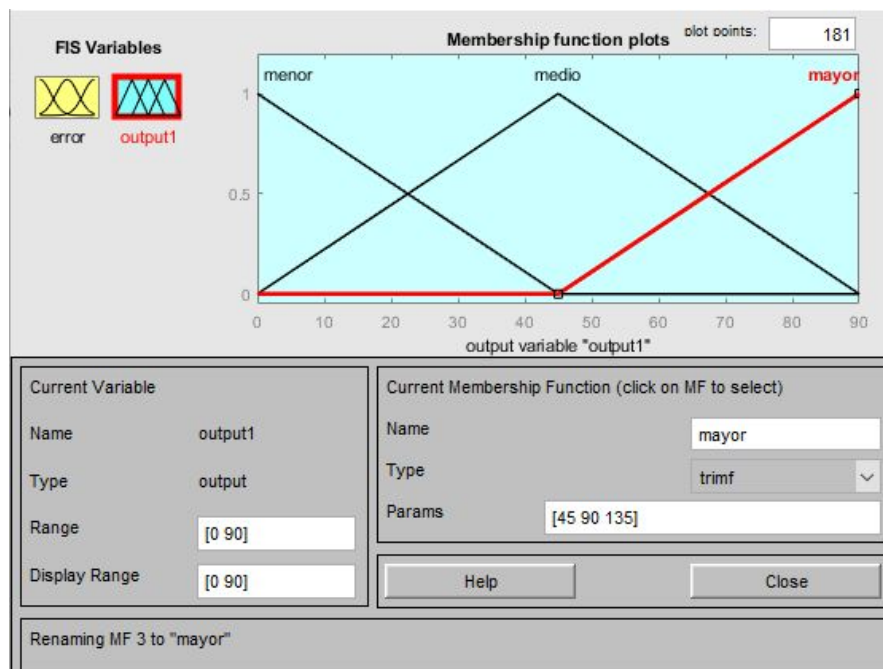


Figura 8: Etiquetas de salida

Finalmente, asignamos las reglas de nuestro sistema. Si la señal de error tiene un valor positivo, el controlador debe aumentar la respuesta, si es negativo debe ocurrir el caso contrario, si es cero debe acercarse al valor central (Figura 9).

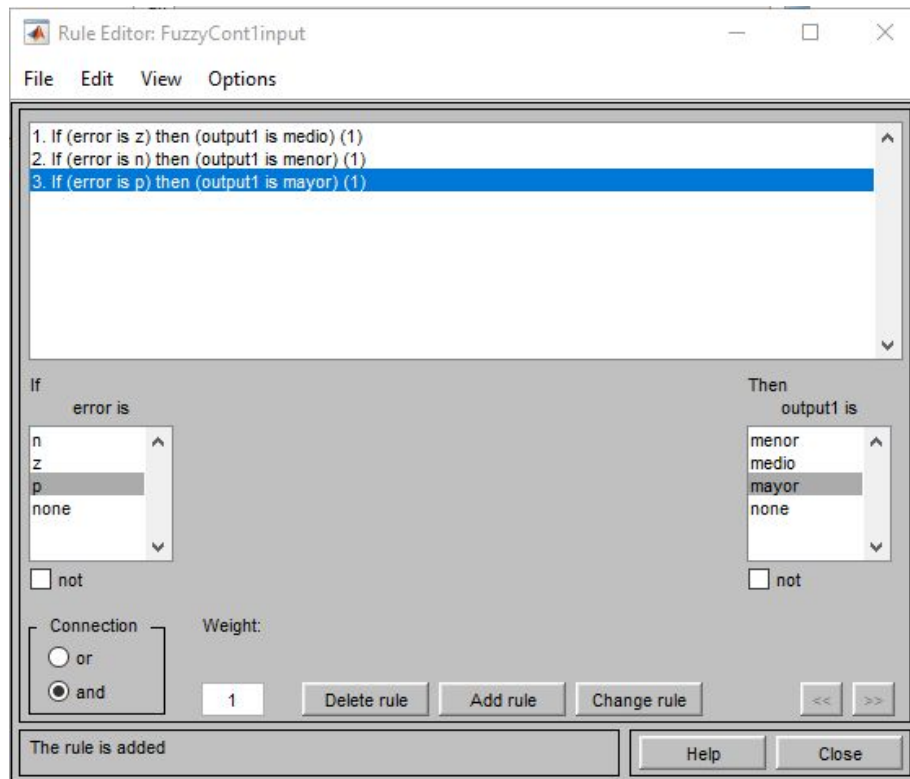


Figura 9: Asignación de reglas

Para la configuración anterior, obtenemos la respuesta de la Figura 10. Guardamos este sistema borroso como "FuzzyCont1input.fis". Una vez cargada en el bloque controlador, obtenemos la respuesta de la figura 10.c

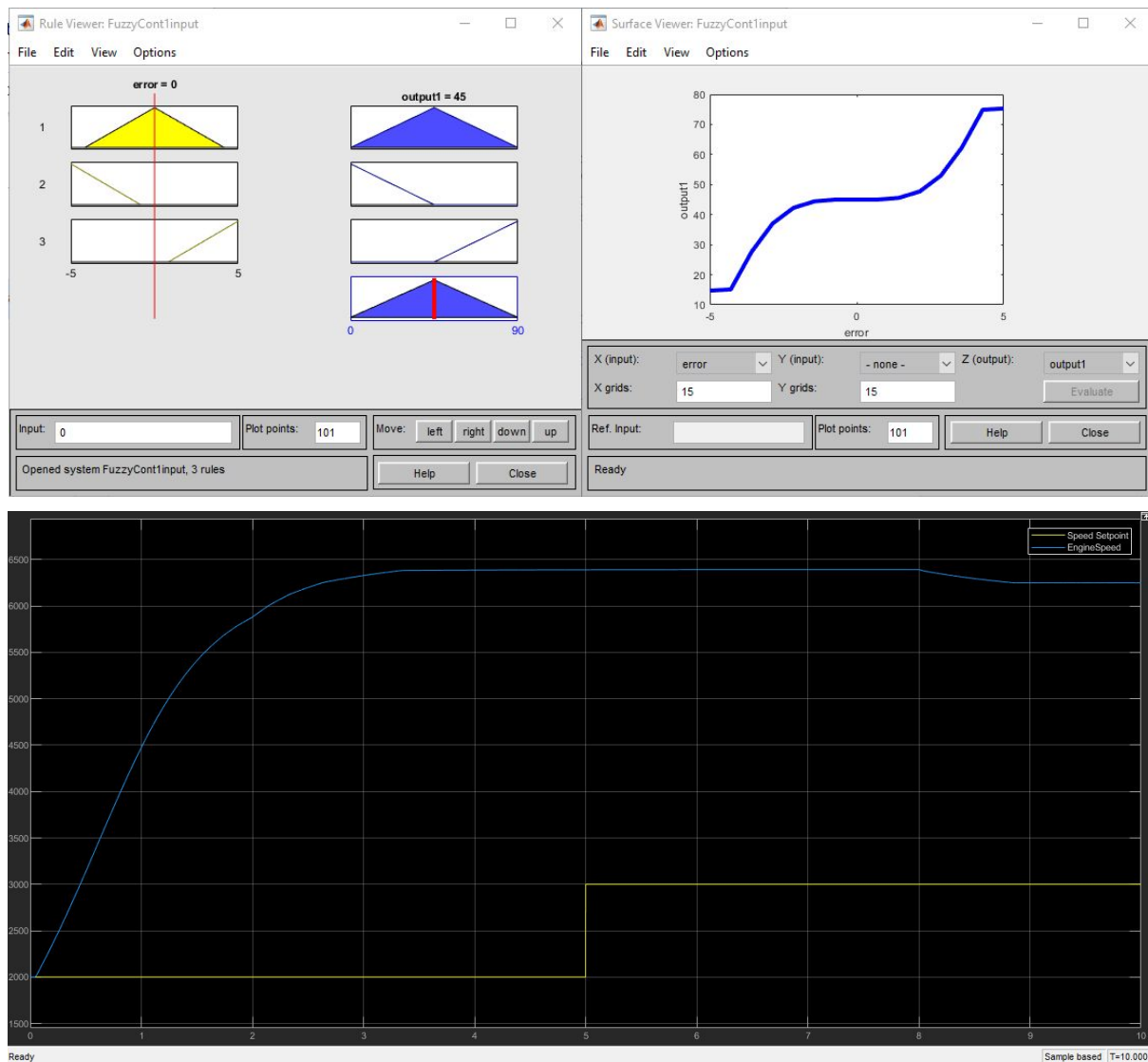


Figura 10 (a) Influencia de reglas (b) Función de transferencia.(c) Respuesta del sistema con el sistema de reglas definido en FuzzyCont1input.fis

Observamos que la señal no es correctamente controlada. Regresando al bloque PI original, podemos observar directamente su comportamiento al agregar un *scope* en sus salidas. Podemos ver que su comportamiento durante operación correcta cubre un rango grande, pero los valores se concentran en un rango más pequeño [7-13], centrado en 10 grados, no en 45 como se asumió previamente.

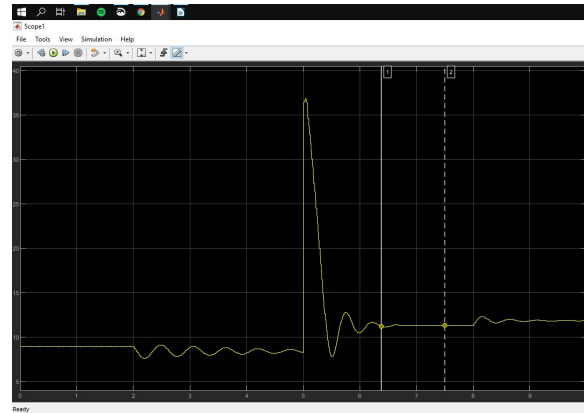
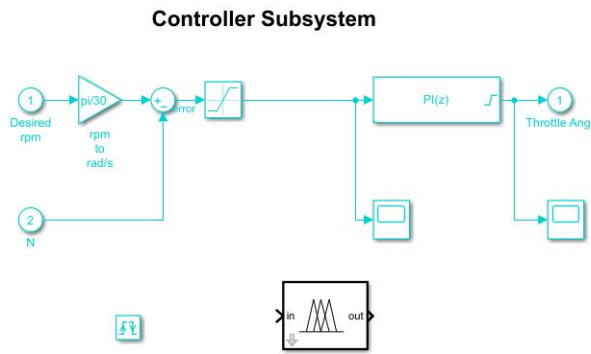


Figura 11: (a) Controlador PI con scopes (b) Respuesta del PI en funcionamiento correcto

Tomando esto en consideración, modificamos la función de salida para que la respuesta de salida esté centrada en 10 (Guardado como *FuzzyCont1inputAlt.fis*). Adicionalmente podemos restringir el rango, el archivo *FuzzyCont1inputShort.fis* trabaja en un rango de [5-15]. Podemos observar las nuevas respuestas

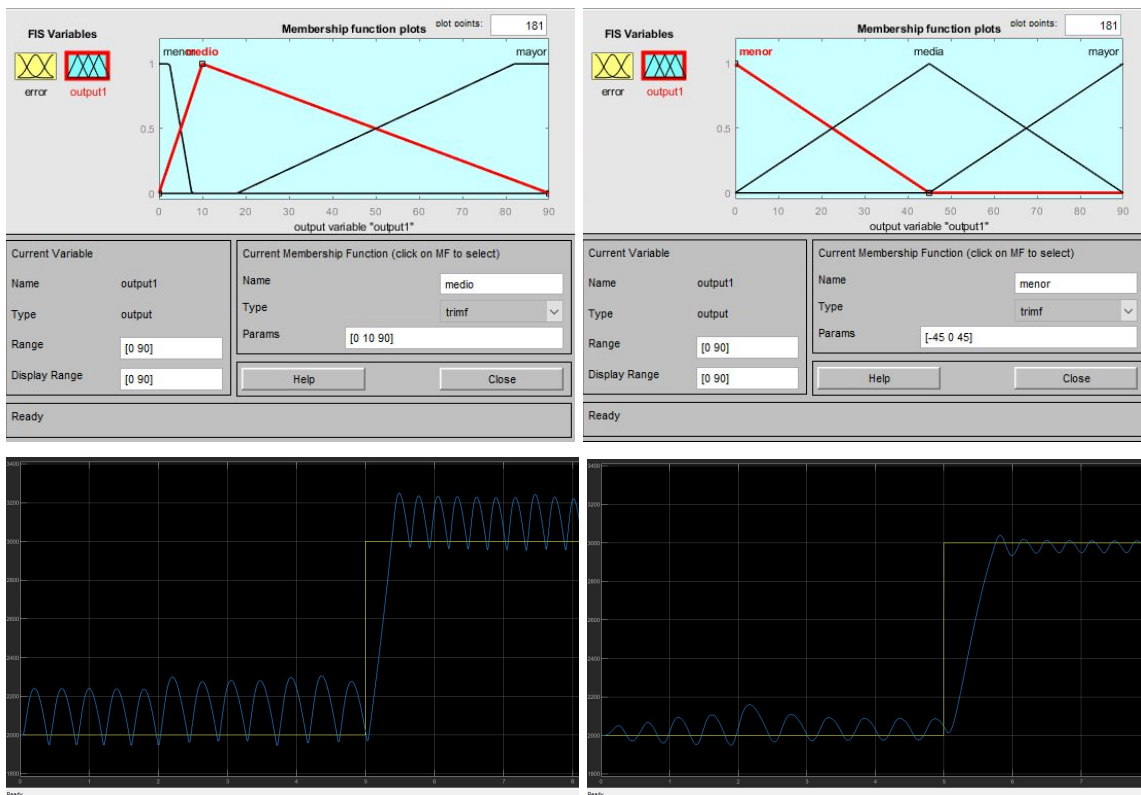


Figura 12: (izquierda) Centro en 10, *FuzzyCont1inputAlt.fis* (derecha) Rango reducido a [5-15], *FuzzyCont1inputShort.fis*

Observamos que obtenemos un controlador lejos de perfecto, pero funcional, con solamente una entrada y 3 reglas sencillas.

Implementación de Control Integral

En la sección anterior desarrollamos un controlador sencillo que funciona específicamente para la señal de entrada indicada: Un pulso que inicia en 2000 rpm y salta a 3000 rpm después de 5 segundos. Sin embargo, ajustar la respuesta a este rango en particular genera problemas. Para una respuesta escalera como la mostrada en la Figura 13, el controlador solo funciona adecuadamente para el rango entre las dos señales utilizadas previamente. Para tener un controlador de uso general, implementamos un sistema de control integral, reemplazando el controlador con el mostrado en la Figura 14. Hemos agregado bloques de saturación para fijar los valores dentro de los límites establecidos.

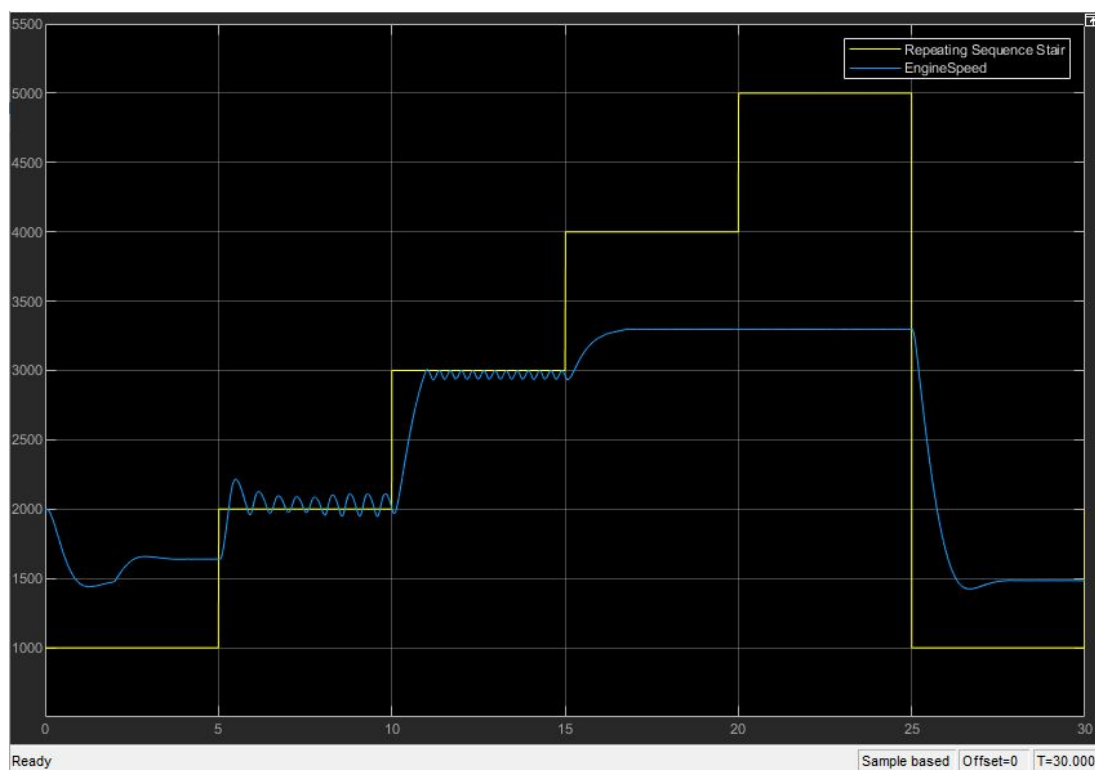


Figura 13: Respuesta fuera del rango preparado

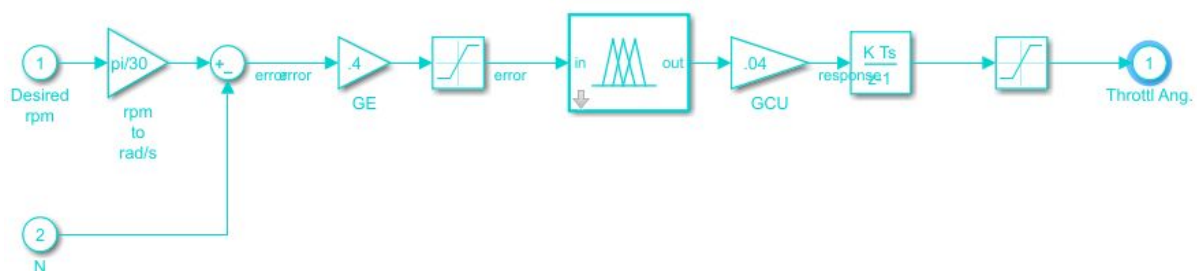


Figura 14: Control integral con integrador discreto

Este sistema cuenta con la ventaja de que no controla la entrada de la planta directamente, sino el cambio de la respuesta, aplicando un bloque integrador en la salida. El sistema borroso dentro del bloque se mantiene fijo durante la optimización, sólo los valores de las constantes (GE y GCU) se modifican. El sistema borroso cuenta con entradas definidas para un rango $[-100,100]$ con 5 etiquetas, salidas entre $[-200,200]$ con 9 etiquetas y reglas mostradas en la Figura 15 (guardado como *Fuzzy1InputSugeno.fis*).

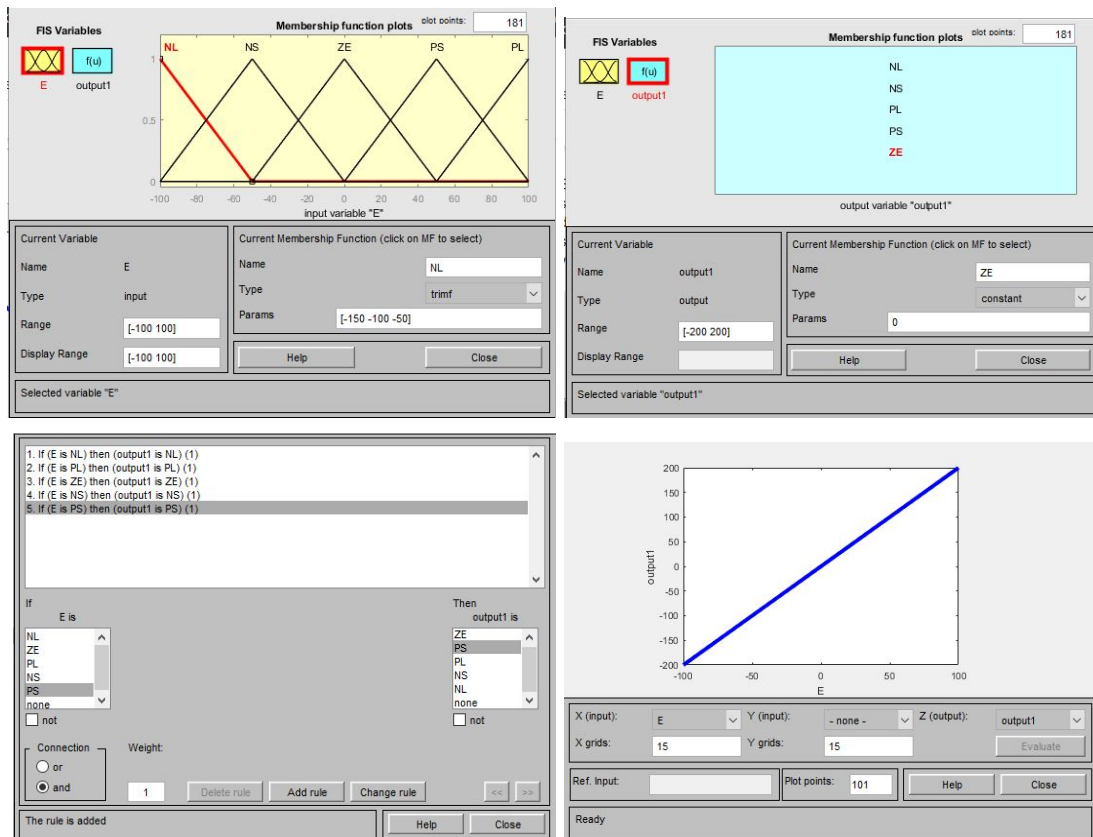


Figura 15: (a) Entradas (b) Salidas (c) Reglas (d) Superficie

Una vez cargado en el sistema, tras ajustar los pesos, obtenemos la respuesta de la planta de la Figura 16. Observamos que tenemos control en todo el espectro con mínimo esfuerzo, pero tenemos una respuesta lenta.

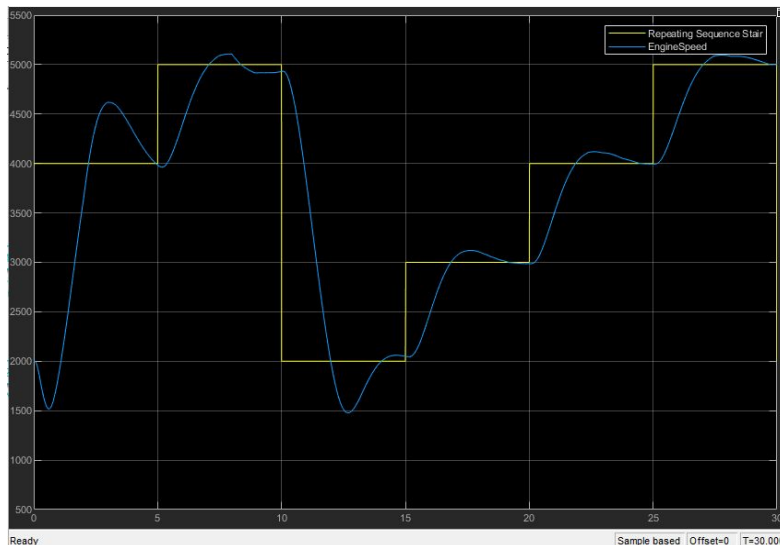


Figura 16: Respuesta para $GE=.3$ $GCU=.06$

Para ajustar los valores de las constantes GE , añadimos un probe a su salida y ajustamos el valor de tal forma que el comportamiento general se encuentre en el rango esperado $[-100.100]$ (Figura 13.a). La constante GCU se ajusta posteriormente para controlar las oscilaciones. Desafortunadamente, con solo estos parámetros no es posible obtener un controlador estable.

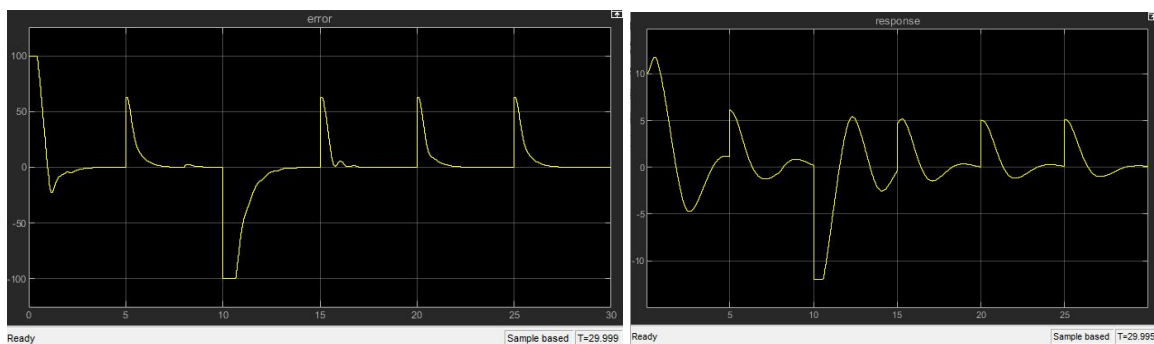


Figura 17: Entrada y Salida con rango ajustado

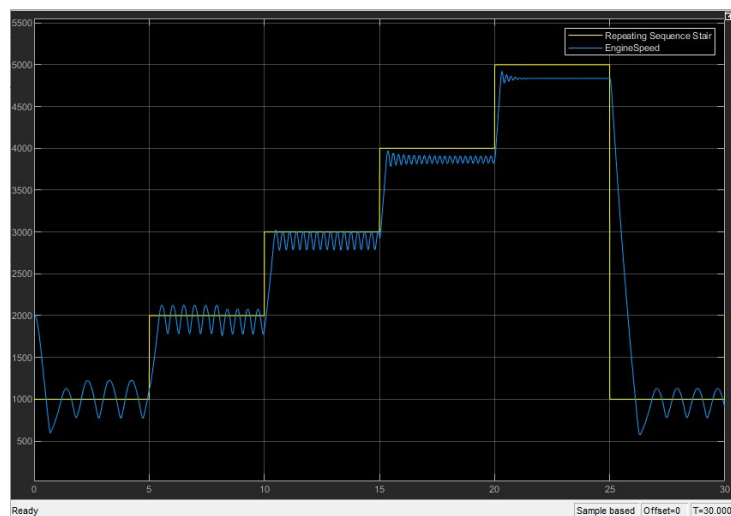


Figura 18: Respuesta después de ajustar constantes.

Control Integral con dos entradas

Podemos implementar una mejora significativa a nuestro controlador: Respuesta no solo al error sino al cambio de error. En otras palabras, reproducir un controlador PI con nuestro sistema borroso.

En el primer reporte determinamos que un bloque diferenciador sobre la señal de entrada produce una señal extremadamente ruidosa, y método experimental determinó que tenían un efecto negativo en la respuesta. Sin embargo, en este caso podemos hacer uso de este, ya que la salida de nuestro sistema pasa por un bloque integrador, teniendo un efecto lineal neto.

La señal contiene ruido, pero este tendrá una influencia mucho menor sobre el sistema final. Como consideración especial, se utiliza un bloque discreto sincronizado con el bloque *trigger* de activación.

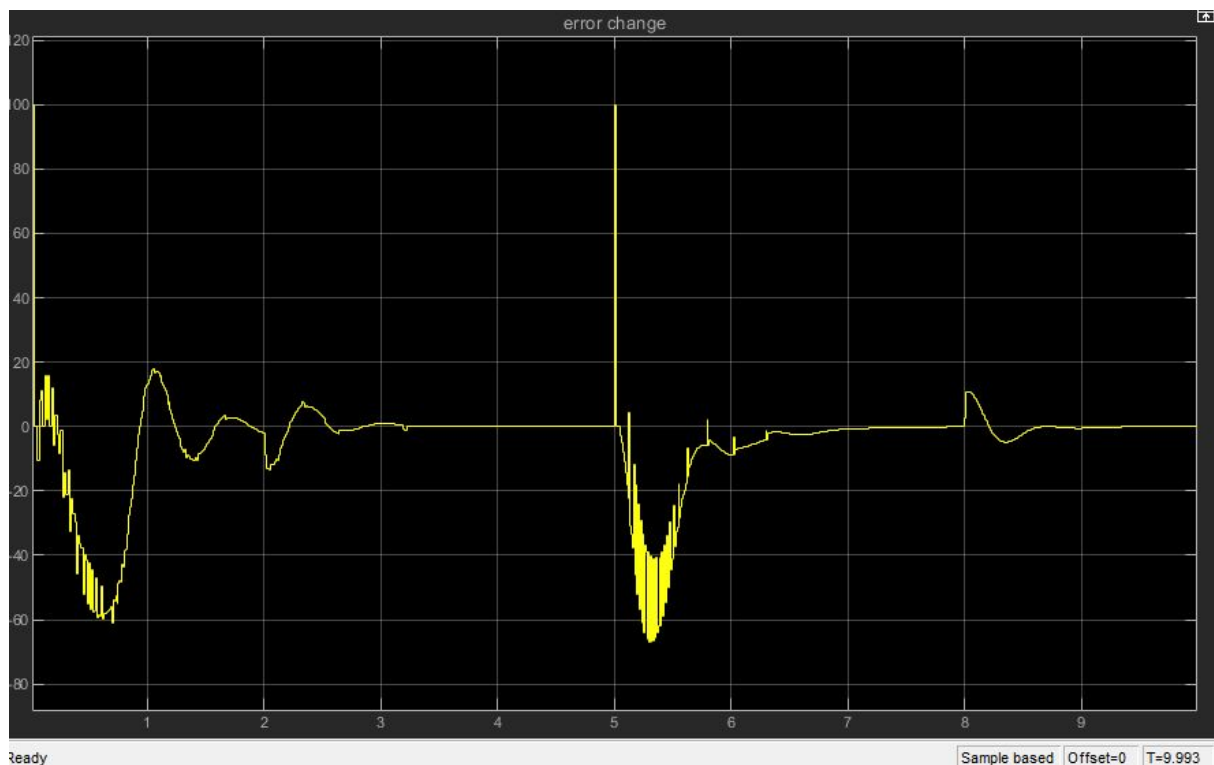


Figura 19: Respuesta derivativa discreta

Partiendo del sistema borroso anterior, agregamos una variable de entrada nueva y configuramos de igual forma, con 5 etiquetas para valores (Muy negativo, Poco negativo, Cero, Poco positivo y Muy Positivo) y nuevas reglas que relacionan ambas variables linealmente (*Fuzzy2InputSugeno.fis*).

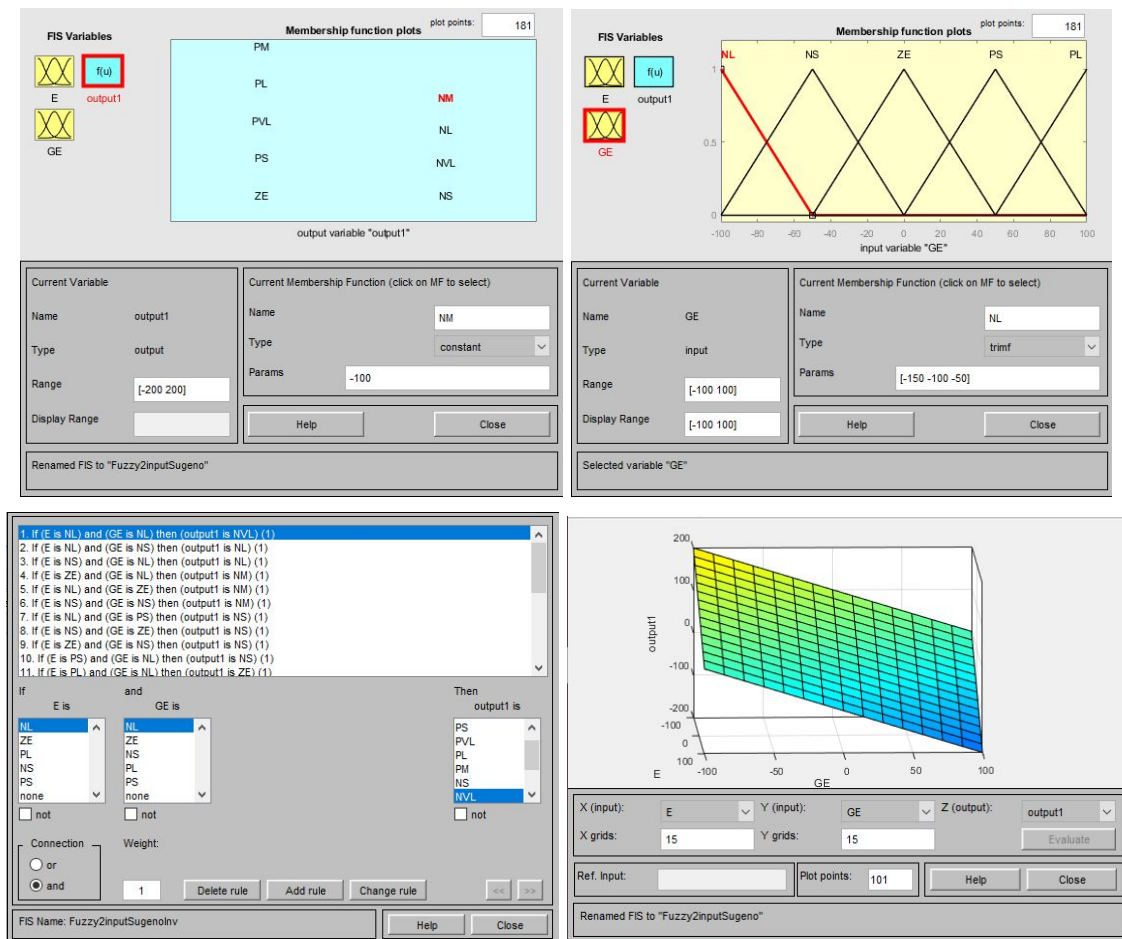


Figura 20: (a) Entradas (b) Salidas (c) Reglas (d) Superficie

Con el sistema borroso completado, hay que realizar las modificaciones necesarias al sistema para soportar dos variables. Se añade un bloque de ganancia, derivador discreto y límite de saturación que alimenta la entrada del bloque del controlador borroso mostrado en la Figura 21 (Guardado como "FuzzyDoubleInputIntegral.slx").

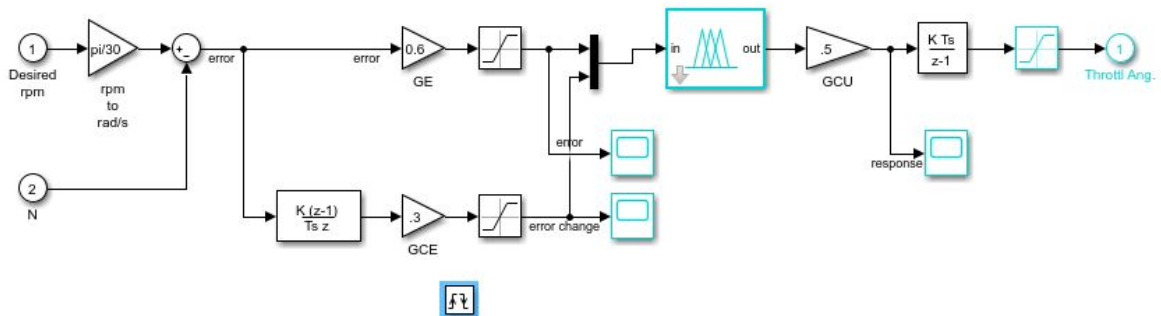


Figura 21: Sistema Borroso PI

Finalmente podemos observar el efecto del componente derivativo en el sistema general (Figura 22), en particular para GCE = 0 (deshabilitado). En la Figura 16 se muestra el comportamiento de la función para valores GE=0.3, GCU=0.5 variando GCE. Podemos observar que el componente derivativo tiene un efecto "suavizante" sobre la respuesta. Esto

nos permite incrementar las constantes GE y GU para obtener una respuesta más agresiva pero contrarrestando oscilaciones.

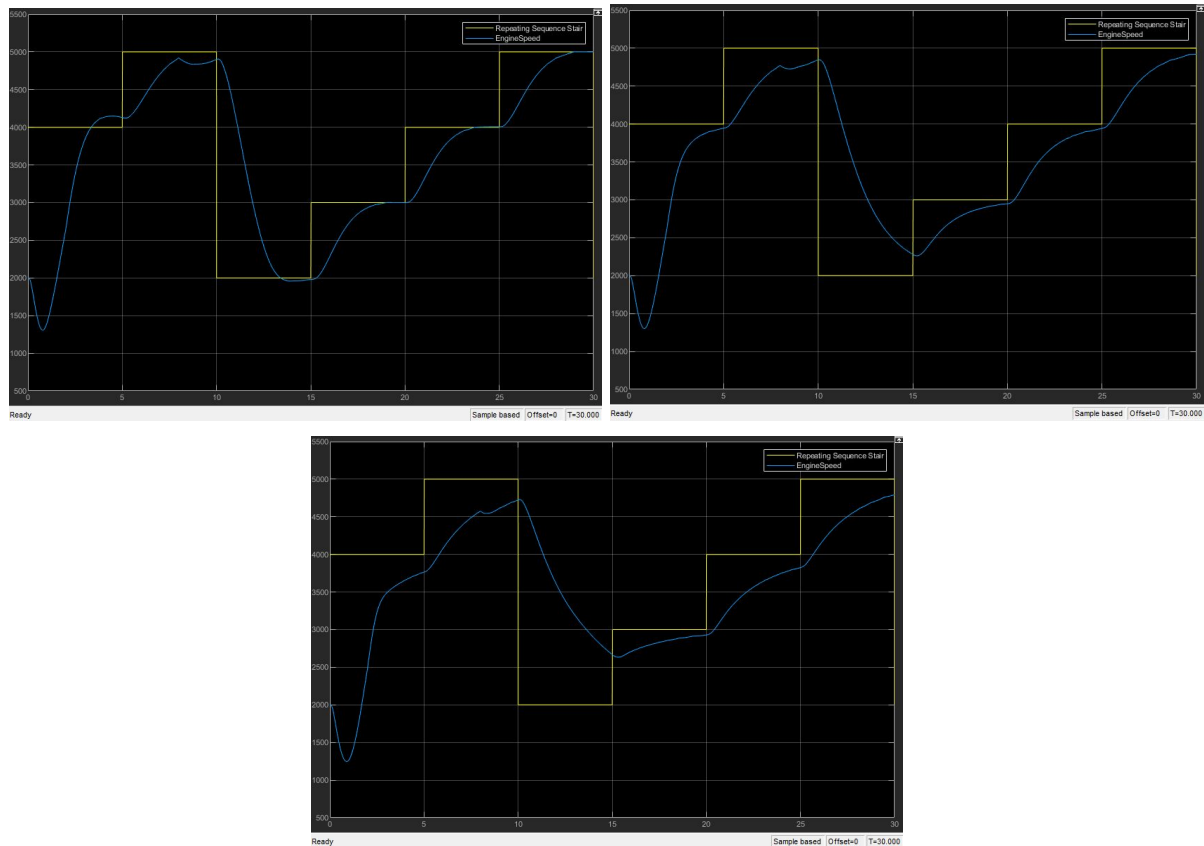


Figura 22: Respuestas para $GE = .3$, variando GCE. a 0 b 0.2 c 0.5

Con el sistema correctamente generado, podemos proceder a ajustar pesos y optimizar nuestra respuesta.

Optimizacion

La planta ha sido ajustada heurísticamente hasta obtener las respuestas mostradas. Se mantiene la restricción explicada anteriormente, Cambiar GE y GCU manteniendo los valores en el rango esperado por el controlador, según el scope de salida. Luego se ajusta el valor de GCE para anular oscilaciones, y se repite el proceso.

La Figura 23 sugiere una característica notable de la planta analizada. Podemos ver que para la configuración mostrada, el controlador oscila considerablemente para bajas rpm, pero se estabiliza rápidamente para valores elevados (Exceptuando distorsiones internas). La discrepancia de oscilaciones se debe a la característica no lineal de la planta, con sensibilidad dependiente al ángulo de apertura y velocidad de giro. Una posible solución a este problema corresponde a alterar la distribución de etiquetas en el bloque borroso, volviendolo un sistema no lineal.

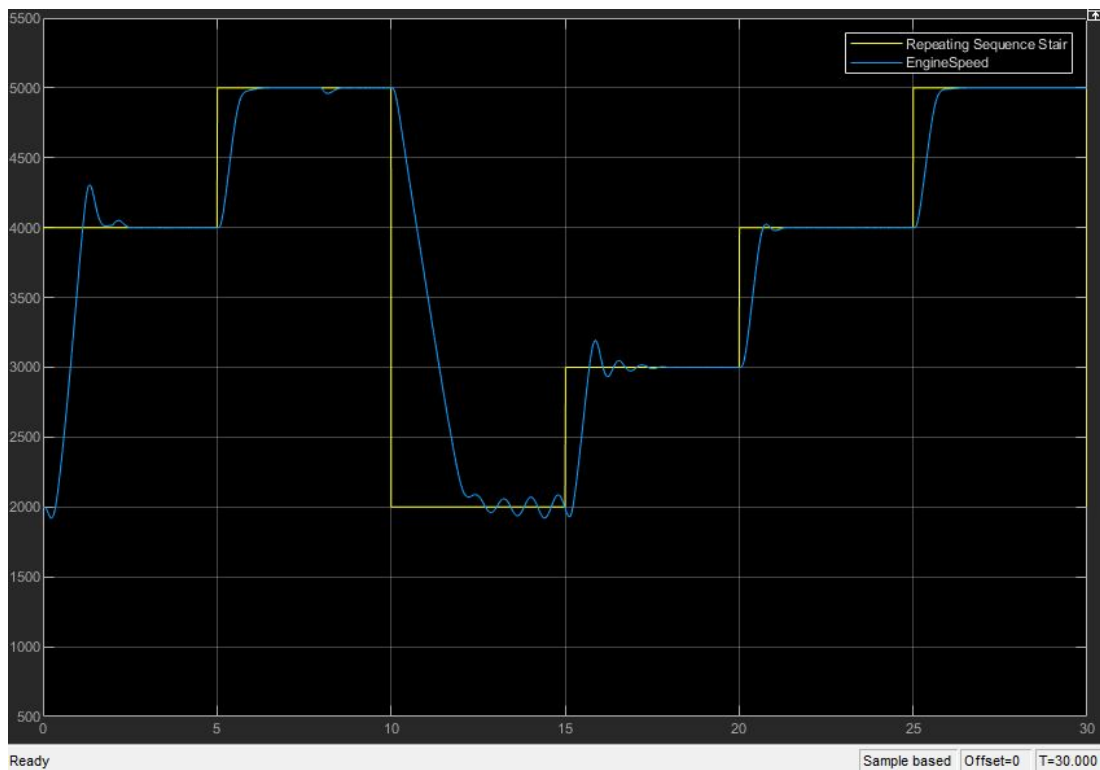


Figura 23: Sistema tras algunas iteraciones. GE = 1.8, GCE = 0.5 GCU = 0.3

Tras algunas iteraciones, obtenemos una respuesta aceptable, con el sistema cambiando entre impulsos rápidamente sin oscilaciones notables.

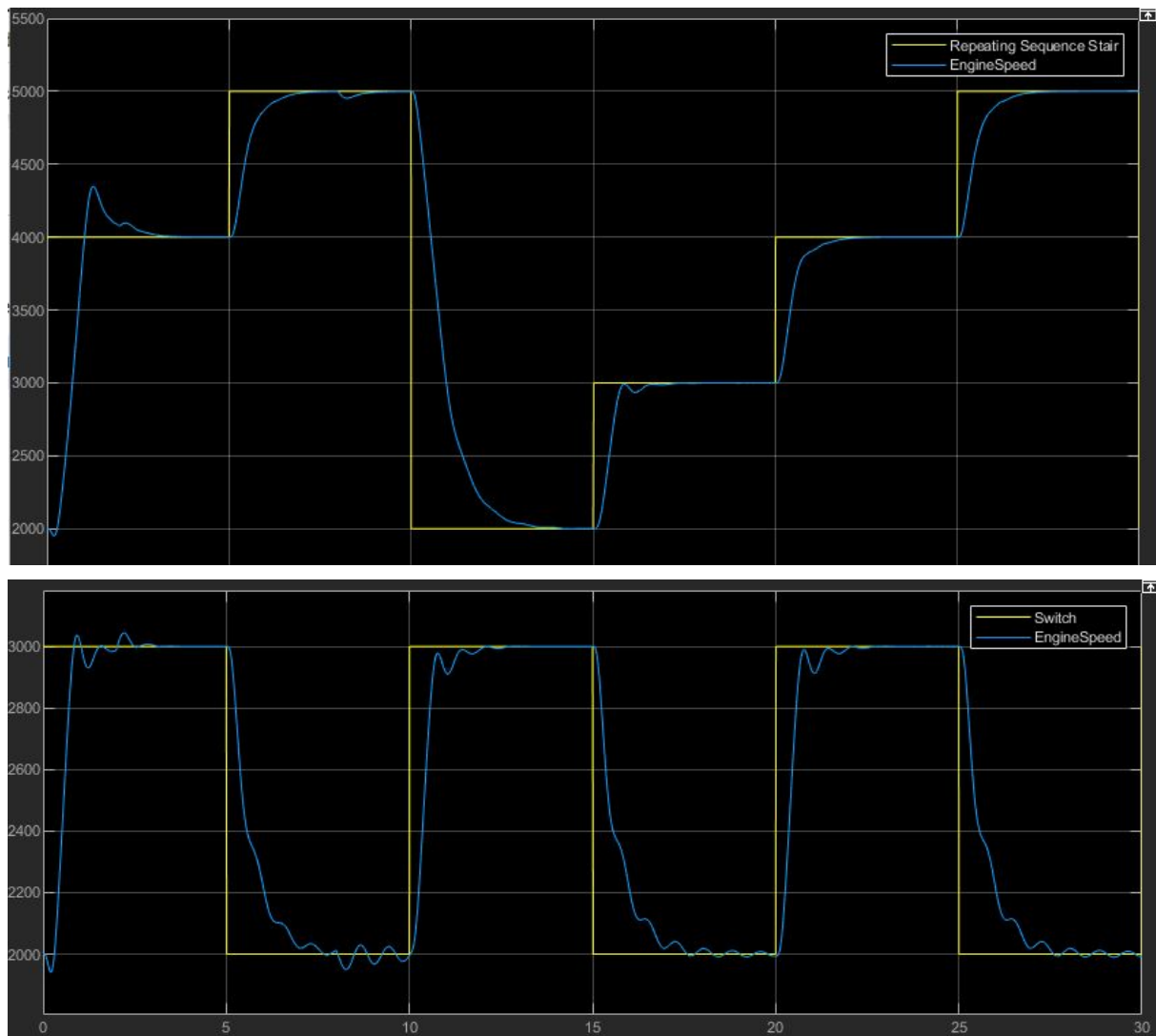


Figura 24: Iteracion final, Respuestas a diferentes pulsos de entrada. GE = 0.6, GCE =0.3
GCU = 0.4

Nuestro controlador final es notable por su relación con el controlador PI mostrado inicialmente. A pesar de que las constantes hayan sido determinadas heurísticamente y de la característica no lineal de la planta, los valores son muy similares a los esperados teóricos dados por:

$$\begin{aligned}
 GE \times GCU &= K_p \\
 K_{p_Real} &= (0.6)(0.4) = 0.24 \\
 K_{p_Teorico} &= 0.27
 \end{aligned}$$

En el caso de T_d el periodo de oscilación

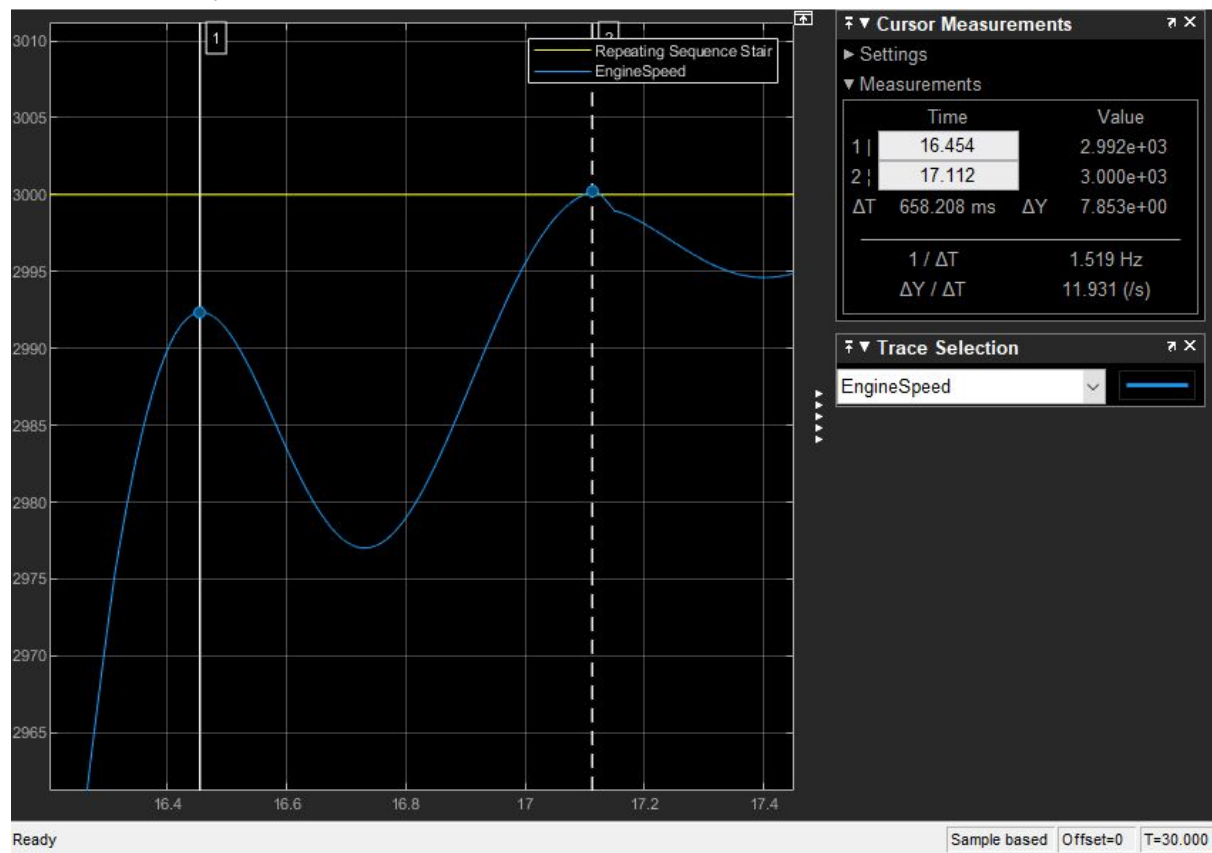


Figura 25: Periodo de la primera oscilación

$$T_{p_Real} = 0.658 \text{ ms}$$
$$T_{d_Teorico} = GCE / GE = (.3)/(0.6) = 0.5 \text{ ms}$$

Conclusion

Durante el desarrollo del sistema logramos diseñar 3 iteraciones de un controlador borroso preciso para un sistema altamente complicado, cada una solucionando problemas fundamentales del sistema anterior.

La ventaja principal de los sistemas borrosos es su gran nivel de abstracción. Diseñar el conjunto de reglas es rápido e intuitivo, requiriendo sólo información básica del sistema. El sistema de lógica fuzzy cumple su cometido original, permitir traducir lenguaje natural en lenguaje matemático fácilmente. Cabe notar que a pesar de desarrollar y optimizar el sistema independientemente del PID previo, los resultados son muy similares a adaptación directa de este.

Sin embargo, como se vio en los resultados, utilizar un sistema fuzzy para controlar sistemas no lineales es particularmente difícil. Aunque matlab cuenta con herramientas poderosas de visualización, ajustar las etiquetas y variables para obtener la respuesta buscada resulta un proceso de prueba y error.

Una posible mejora a los resultados podría ser aplicar algoritmos de optimización del proyecto anterior. Diseñar funciones de costo para los parámetros libres y minimizar para obtener respuestas específicas. Dado que los resultados tienen sistema PID equivalente, se pueden usar las ecuaciones que lo relacionan para obtener ecuaciones con las limitaciones matemáticas necesarias.