

**PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ**

**ESCUELA DE POSGRADO**



**MODELADO DE UN MOTOR DIÉSEL MARINO Y  
DESARROLLO DE SU SISTEMA DE CONTROL CON  
GANANCIAS PROGRAMADAS**

Tesis para optar el grado académico de

**Magíster en Ingeniería de Control y Automatización**

**Autor:** Fidel Burga Gheresi

**Asesor:** Dr. Juan Javier Sotomayor Moriano

Lima – Perú

2021

## RESUMEN

El sistema de control de los motores diésel marinos juega un rol importante en el desempeño de estos en el campo de la propulsión marina. En la actualidad en muchos casos, la velocidad angular de estos motores es controlada por técnicas convencionales, las cuales resultan ineficientes al no tomar en cuenta el comportamiento no-lineal de esta clase de motores. En el presente trabajo se propuso el modelado de un motor diésel sometido a un ambiente marino utilizando redes neuronales y el desarrollo del sistema de control de sus RPM empleando un PID de ganancias programadas, teniendo como propósito la mejora de su desempeño respecto a un PID convencional. Para alcanzar este objetivo se estudió el comportamiento dinámico de un motor diésel, lo cual permitió identificar las variables de entrada y salida más relevantes. Luego, empleando la información medida de un motor diésel real se realizó el modelado basado en redes neuronales no lineales autorregresivas con entrada exógena (NARX), logrando un buen desempeño en validación con data que no fue usada durante el entrenamiento. El diseño del controlador avanzado con ganancias programadas se realizó, encontrando los puntos de equilibrio para los puntos de operación del motor diésel, linealizando el motor en estos puntos y considerando ciertas restricciones impuestas en un esquema de optimización. Lo anterior permitió encontrar las mejores ganancias programadas del algoritmo PID para los puntos de operación encontrados. Asimismo, para mostrar las mejoras en el desempeño, el controlador PID con ganancias programadas diseñado se comparó con un PID convencional. Los resultados evidenciaron que el uso del controlador PID con ganancias programadas supera en todas las pruebas al PID estándar, con mejor desempeño en el seguimiento de la referencia, ahorro de combustible y estabilidad ante perturbaciones. Finalmente se propuso la implementación práctica del sistema de control desarrollado, seleccionando la instrumentación necesaria y utilizando una unidad de control de motor de código abierto basado en un microcontrolador comercial.

## ÍNDICE

RESUMEN .....	2
INTRODUCCIÓN .....	5
CAPÍTULO 1. ESTADO DEL ARTE DE LOS SISTEMAS DE CONTROL DE MOTORES DIÉSEL MARINOS.....	8
1.1 Introducción .....	8
1.2 Control de motores diésel .....	10
1.3 Control PID .....	22
1.4 Control PID con ganancias programadas.....	23
1.5 Objetivos de la tesis.....	26
CAPÍTULO 2. MODELADO MATEMÁTICO DEL MOTOR DIÉSEL MARINO.....	27
2.1 Introducción .....	27
2.2 Estado del arte del modelado matemático .....	27
2.3 Modelos basados en redes neuronales.....	32
2.4 Modelado matemático del motor diésel.....	36
2.5 Identificación por redes neuronales .....	42
2.6 Conclusiones .....	59
CAPÍTULO 3. DESARROLLO DEL SISTEMA DE CONTROL.....	60
3.1 Introducción .....	60
3.2 Diseño de un controlador PI estándar .....	60
3.3 Diseño de un controlador PI con ganancias programadas .....	64
3.4 Análisis de estabilidad de ambos controladores.....	76
3.5 Conclusiones .....	80
CAPÍTULO 4. PROPUESTA DE IMPLEMENTACIÓN DEL SISTEMA DE CONTROL .....	81

4.1	Introducción .....	81
4.2	Selección de instrumentos .....	81
4.3	Conclusiones .....	88
CONCLUSIONES GENERALES.....		89
RECOMENDACIONES .....		91
REFERENCIAS BIBLIOGRÁFICAS.....		92



## INTRODUCCIÓN

Los sistemas de propulsión marinos están basados principalmente en motores diésel, por su elevada confiabilidad, larga vida útil y amplia gama de potencias. Tan igual que en el campo automotriz e industrial, los motores diésel debido también a su elevada eficiencia han demostrado que siguen siendo los predilectos en muchos campos de la actividad humana, por lo que se continuará con su desarrollo.

Parte esencial de los motores diésel la conforma su sistema de control, el cual tiene como tarea principal, efectuar el gobierno de la potencia suministrada de acuerdo a los requerimientos del operador (que normalmente viene indicado en revoluciones), tratando de mantener un mínimo nivel de consumo de combustible y procurando una apropiada combustión, a fin de minimizar la emisión de contaminantes.

La tecnología actual nos permite aplicar las diferentes técnicas o estrategias de control avanzado en los diversos campos de desarrollo humano, como es el caso del control de los motores diésel marinos, que por lo general en la actualidad presentan sistemas de regulación mecánicos-hidráulicos y electrónicos. En el caso de los sistemas mecánicos-hidráulicos tenemos que estos son controladores proporcionales o proporcionales integrativos, y en el segundo caso son PID (proporcional integral derivativo) convencionales. La simplicidad en la sintonización del PID ha logrado que este se use en la gran mayoría de aplicaciones de sistemas de control, pero en algunos casos su desempeño deja espacio para mejoras.

La sintonización del PID generalmente se hace de forma manual, pero involucra tiempo y experiencia del operador. Una forma de agilizar las aplicaciones del PID es a través de un modelo matemático del proceso. Para motores diésel esto puede llevar cierta dificultad, dado la complejidad de su comportamiento dinámico, además muchas veces en el caso de motores diésel marinos no se tienen las especificaciones técnicas, ni los sensores adecuados para medir todos

los parámetros involucrados en un modelado teórico. Otra forma de realizar el modelado, es lo que se denomina un modelado de caja negra donde solo importan las entradas y salidas del proceso, lo cual es conveniente en el sentido que se requieren menos sensores y menos información acerca de la construcción interna (lo que generalmente es secreto comercial del fabricante). Actualmente uno de los métodos más prometedores para realizar modelado de caja negra es usando redes neuronales, este trabajo se enfocará en este tipo de modelado.

En cuanto al control, es necesario considerar que en el caso de motores diésel marinos como en cualquier motor diésel en general tiene un comportamiento no lineal. Esto quiere decir que el controlador PID convencional podría ajustarse para un punto de operación, pero perdería su eficacia a medida que uno trabaja en otros puntos de operación. Existen técnicas adaptativas para paliar este problema, entre las cuales una que se ha aplicado exitosamente es la de ganancias programadas.

Este trabajo tiene como objetivo general, el desarrollo del modelado de un motor diésel sometido a un ambiente marino utilizando redes neuronales y desarrollo de su sistema de control de las RPM empleando un PID de ganancias programadas, con el fin de mejorar su desempeño respecto a un PID convencional. Para lograr esto se tienen los siguientes objetivos específicos:

- Estudio del comportamiento dinámico y caracterización de las variables que gestionan su desempeño.
- Obtener el modelo matemático del motor diésel marino utilizando redes neuronales.
- Diseñar un controlador PID con ganancias programadas y un PID convencional con el fin de comparar su desempeño de seguimiento de las RPM, eficiencia de uso de combustible y estabilidad ante perturbaciones.
- Desarrollar una propuesta de implementación del controlador avanzado diseñado en un controlador comercial.

Para alcanzar los objetivos planteados, la presente tesis consiste de 4 capítulos que se detallan a continuación:

En el capítulo 1 se presenta el estado del arte de los sistemas de control de motores diésel, donde se describe las variables y estrategias de control más relevantes en la actualidad. Además, también se describe el funcionamiento básico de un motor diésel.

En el capítulo 2 se describe de manera general el modelado matemático y en particular los modelos basados en redes neuronales. Luego se presentan las relaciones que existen entre las distintas variables involucradas en el comportamiento de un motor diésel y en la carga que soporta un buque al trasladarse. Finalmente, se realiza el modelado de un motor diésel con redes neuronales no lineales autorregresivas con entrada exógena (NARX) y se valida el modelo obtenido con una data de prueba distinta a la que se utilizó en el entrenamiento, lográndose un buen desempeño.

En el capítulo 3 se desarrolla el diseño del controlador PID convencional y del controlador PID con ganancias programadas. El diseño de este último consistió en la obtención de los puntos de operación, la linealización y la selección de las ganancias programadas en la búsqueda de un comportamiento deseado. Finalmente se hace una comparación de la respuesta en el tiempo de ambos controladores, de la estabilidad y de la eficiencia en el consumo de combustible.

En el capítulo 4 se propone la implementación del sistema de control donde se elige la instrumentación, una unidad de control del motor (ECU) de código abierto y se genera el código en C para el microcontrolador de la ECU.

## CAPÍTULO 1. ESTADO DEL ARTE DE LOS SISTEMAS DE CONTROL DE MOTORES DIÉSEL MARINOS

### 1.1 Introducción

Actualmente la estrategia de control más utilizada para motores diésel marinos sigue siendo el PID o PI (Wang et al., 2018) en un lazo de control. Estos controladores son calibrados por prueba y error, asimismo, si estos operan lejos de sus puntos de operación nominales no presentarán un buen desempeño (Weigang Pan et al., 2010). El proceso de la calibración suele requerir una serie de experimentos que toman tiempo, por lo que sería una ventaja aplicar una metodología que permita realizar rápidamente la simulación del motor para la calibración de los controladores. Si bien existen controladores electrónicos, es muy común en buques el uso de otro tipo de controladores los cuales no son digitales, sino que son un sistema mecánico que consiste en un par de esferas conectadas al eje del motor y que por fuerza centrífuga mueve un husillo que abre o cierra el paso del flujo de combustible (Fig. 1.1). Este tipo de gobernador mecánico puede clasificarse en moderadores o gobernadores genuinos (Mayr, 1971). Los moderadores funcionan como un controlador proporcional y los gobernadores genuinos como un controlador proporcional integrativo. Actualmente la Marina de Guerra del Perú tiene en su escuadra buques que trabajan con gobernadores mecánicos, siendo parte de la motivación de esta tesis el mejoramiento de la eficiencia del consumo de combustible y el seguimiento de la referencia de estos buques.

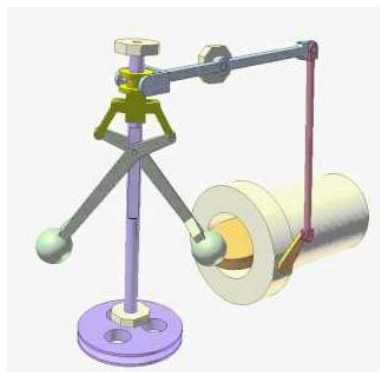


Fig. 1.1. Gobernador mecánico

## **Antecedentes**

El año 2003, el tesista presentó en el curso de Proyecto de Tesis I, un trabajo que es el antecedente de esta tesis, en el cual se llegó a implementar un sistema de control en un motor diésel monocilindro de pruebas (que permitía cambios de carga manual). Se planteó que el sistema de control del motor diésel marino debía mejorar el desempeño, dando la idea de cambio en el control de las revoluciones por la de controlar la potencia entregada (potencia al freno). Lo anterior, logra reducir las compensaciones del actuador para mantener las revoluciones del motor ante los continuos cambios de carga producto de las perturbaciones producidas por las variaciones de las diferentes resistencias de la propulsión naval, las cuales originan constantes transiciones en el proceso de combustión, haciéndolo ineficiente. Manteniéndose la operación del motor en diferentes puntos de similar potencia entregada con rangos permisibles de variación de las revoluciones del motor, se conseguiría mayor estabilidad del proceso de combustión, menor cantidad de transiciones del proceso de combustión, mejora de la eficiencia en el consumo de combustible, reducir los contaminantes productos de la combustión y reducir el desgaste de las piezas del motor.

Esto se efectuó mediante un control por brecha que se implementó por medio del uso de un actuador (motor paso a paso), un sensor de revoluciones, un sensor de seguridad que era una termocupla y una tarjeta electrónica con un microcontrolador microchip 16F877A.

Este trabajo de investigación a pesar de haberse logrado implementar no logro identificar los modelos del motor en los diversos puntos de operación del motor, logrando controlarlo en un solo punto de operación, sin embargo, si nos demostró que era factible la implementación del control electrónico en los motores diésel marinos, así como la necesidad de implementar un tipo de control más eficiente como el de ganancias programadas, lo cual se propone en esta tesis.

## 1.2 Control de motores diésel

Para comprender la dificultad de un motor diésel se explicará primero el funcionamiento de este tipo de motor. Los componentes principales de un motor de combustión diésel se muestran en la Fig. 1.2. El funcionamiento consiste en llevar el aire dentro de la cámara de combustión con una presión ligeramente menor que la presión atmosférica; si no hay turbocargador, el torque se controla con la masa inyectada de combustible. Sin embargo, actualmente es común el uso de turbocargadores para mejorar el torque y en consecuencia la potencia. La carga de aire depende de la presión de carga, que es controlada por su controlador respectivo. A baja carga, el motor diésel opera con exceso de aire (el exceso de aire se representa por el factor  $\lambda$ ). Cuando se tienen grandes cargas,  $\lambda$  se aproxima a 1. Si este valor es muy bajo, el motor empezará a expulsar humo negro. El sistema electrónico que se encarga de la gestión del control de un motor de combustión interna se le conoce como ECU (Engine Control Unit por sus siglas en inglés). La ECU controla la masa inyectada de combustible dentro de la cámara de combustión según el requerimiento del usuario. La cual es muy importante para definir el consumo de combustible, NOx y la formación de hollín. El flujo másico de aire en la admisión esta indirectamente controlado por la válvula de EGR (Recirculación de gases de escape). El lazo del control de flujo másico de aire toma el dato del sensor de flujo de aire y su valor de referencia es calculado con el factor de exceso de aire deseado. Otro lazo de control es el de la presión de carga, que manipula los alabes del turbocargador con geometría variable. Ambos lazos presentan alto acoplamiento. En cuanto a la velocidad del motor, es importante el controlador de velocidad que limita la velocidad reduciendo el flujo de combustible. Por último, existe un control de tratamiento de gases que tiene por objetivo reducir los productos no deseados de la combustión, como el HC, CO, NOx y otras partículas.

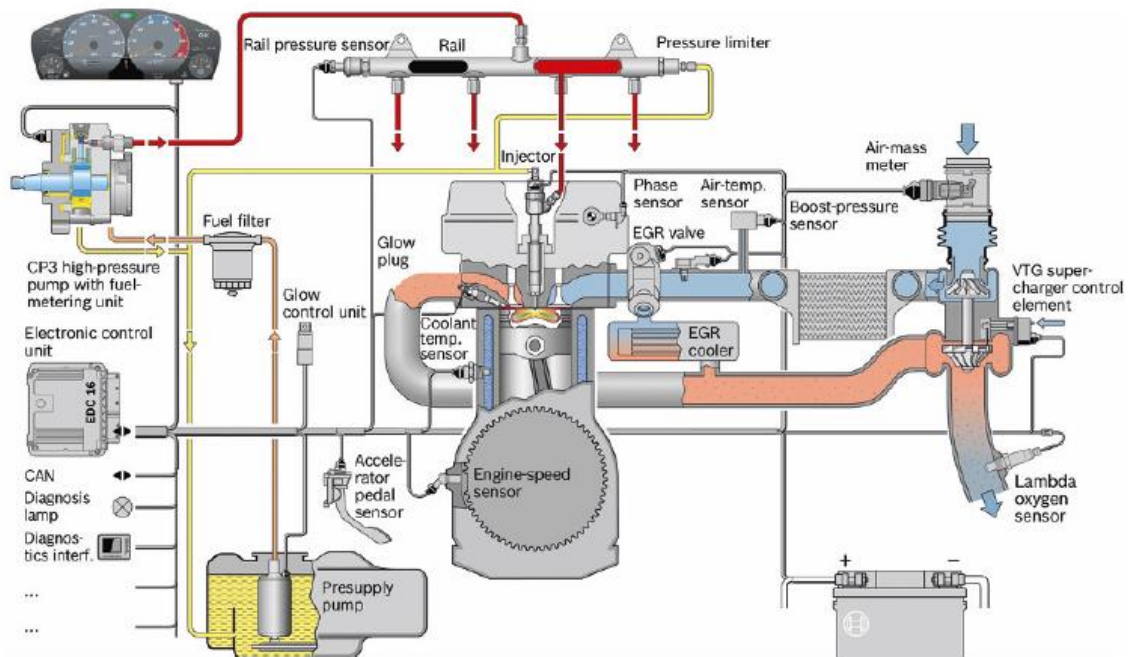


Fig. 1.2. Motor diesel (Isermann, 2014)

El control en un motor de combustión se puede agrupar en los subsistemas o módulos para el control de torque, control de inyección y combustible, control de carga de aire y EGR, y control de emisiones. Las variables manipuladas más utilizadas son el flujo de aire, el ángulo y masa inyectada, presión en el rail, flujo de EGR, fase del balancín y turbocargador. El control en lazo cerrado usualmente se realiza para el flujo de aire, presión en el rail común y presión de carga de aire. Además de los principales lazos de control, existen equipos secundarios que requieren de un controlador como el control de la posición del acelerador, la bomba de inyección, el eje de levas, control de presión para el combustible y el aceite de lubricación. Por otro lado, la ECU también puede controlar por método feedforward, este se implementa como tablas de búsqueda tridimensionales o mapas 3D. Esto es así por el comportamiento no lineal del motor.

El desarrollo del control electrónico de motores diésel no tiene muchos años, pues recién desde el año 1989 se tienen registros de este tipo de control cuando aparecieron las primeras bombas de distribución controladas por microprocesadores. A partir de ese año el desarrollo en el control de motores se extendió más y aparecieron en 1992 el sistema de recirculación de gases de escape (EGR), catalizadores de oxidación y turbocargadores con geometría variables. Hoy en día los motores diésel para automóviles se caracterizan por una inyección common rail de alta presión (2000 bar), piezo inyectores, turbocargadores mellizos o cargadores VGT, altos ratios de EGR, catalizadores DeNox, filtros de partículas con regeneración y sistemas de reducción catalítica selectiva (SCR) (Isermann, 2014)

La computadora del motor tiene la tarea de controlar, optimizar y supervisar todas las funciones que hacen posible el control actual de los motores. Los objetivos principales de este sistema son:

- Generación de torque según la demanda del usuario
- Bajo consumo de combustible
- Bajas emisiones durante el escape y menor ruido
- Buena maniobrabilidad

Los principales sistemas del motor a controlar generalmente se consideran los siguientes:

- Sistema de admisión de aire
- Sistema de inyección de combustible
- Suministro de combustible
- Sistema de lubricación
- Sistema de gases de escape
- Sistema de enfriamiento

La optimización de estos sistemas ha derivado en un creciente número de actuadores y sensores que requieren el manejo de varias funciones dentro de la

ECU. El desarrollo de estas funciones en el caso de motores diésel produjo una mayor optimización del sistema de inyección múltiple, la sincronización de las válvulas, la carga del turbocompresor, el EGR y el filtro de partículas. De acuerdo al trabajo de (Isermann, 2014) los sistemas de control para motores diésel se pueden subdividir en:

- Control de torque
  - o Control de la inyección
    - Control de flujo másico de combustible
    - Control del ángulo de inyección
    - Control de presión de la inyección
    - Control de inyección múltiple
  - o Control de masa de aire
  - o Control de presión de carga
- Control de emisiones
  - o Control de recirculación de escape
  - o Control de filtro de partículas
- Control de limitación de humos
- Control de velocidad en ralentí
- Control de temperatura del refrigerante

Debido a la amplia cantidad de variables en cada sistema de control, esta tesis se enfocará en modelar un motor combustión interna como base para diseñar el control, por lo que las reseñas acá expuestas se centrarán en esta área. El modelado de un motor de combustión es una tarea compleja pues el número de variables a controlar en un motor de combustión puede variar entre 8 y 20 dependiendo de la configuración (Cook et al., 2008). Un motor de combustión común puede tener la configuración y variables mostradas en la Fig. 1.3.

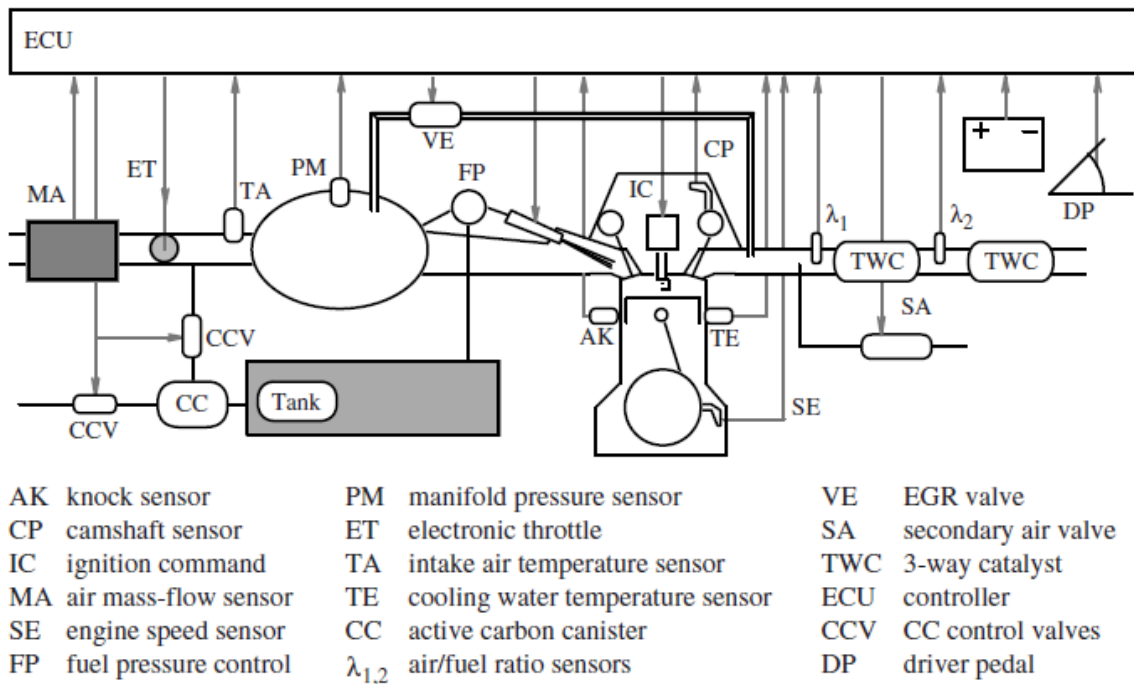


Fig. 1.3. Variables de un motor de combustión interna (Guzzella & Onder, 2010)

Los lazos de control de estas variables se muestran en la Fig. 1.4 donde se puede encontrar la ruta para el combustible  $P_\phi$  y la ruta para el aire  $P_\alpha$  que definen la relación aire combustible que entra a la cámara de combustión. Se muestra también el bloque de combustión  $P_\chi$  que determina la cantidad de torque producido por el motor. Otras señales incluidas en el diagrama son la señal de estallido  $y_\zeta$  medida por el sensor de estallido  $P_\zeta$  y la relación aire combustible de salida  $y_\lambda$  medida por el sensor  $P_\lambda$  ubicado a la salida de las válvulas de escape. La velocidad del motor  $\omega_e$  es la salida del bloque  $P_\theta$ , tomando en cuenta la inercia rotacional del motor, cuyas entradas son el torque del motor  $T_e$  y el torque de la carga  $T_l$ .

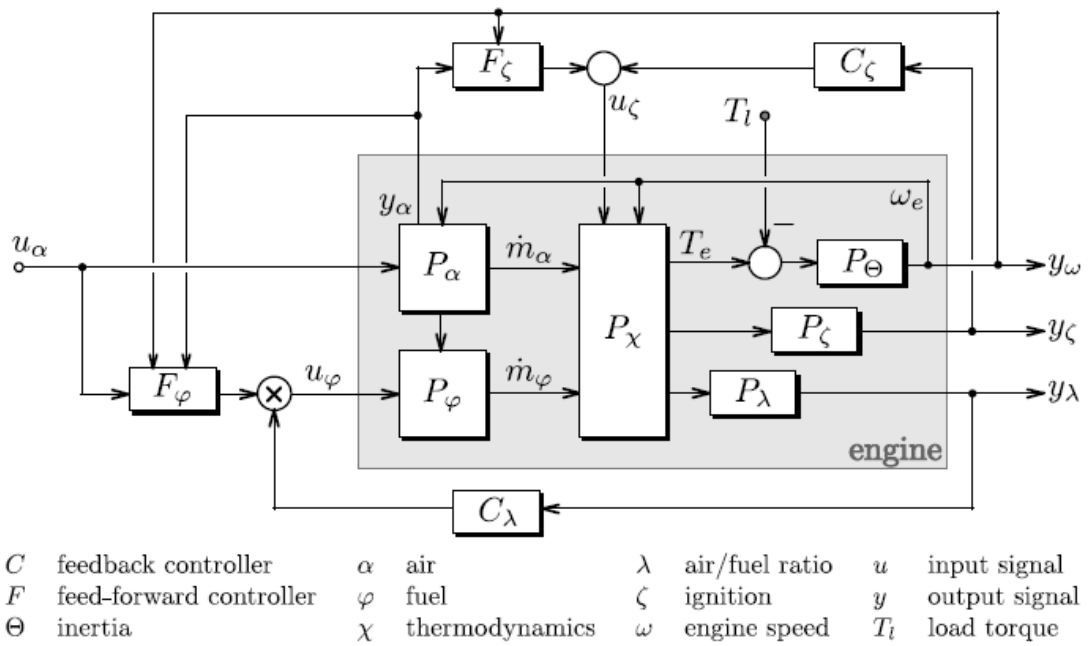


Fig. 1.4. Lazos de control de un motor de combustión interna (Guzzella & Onder, 2010)

De la figura anterior se pueden identificar los lazos principales de control:

- El lazo feedforward de la inyección de combustible.
- El lazo feedback de relación aire / combustible.
- El lazo feedforward del ángulo de ignición
- El lazo feedback del estallido (knock)

En el caso de un motor diésel típico, este tiene la configuración mostrada en la Fig. 1.5.

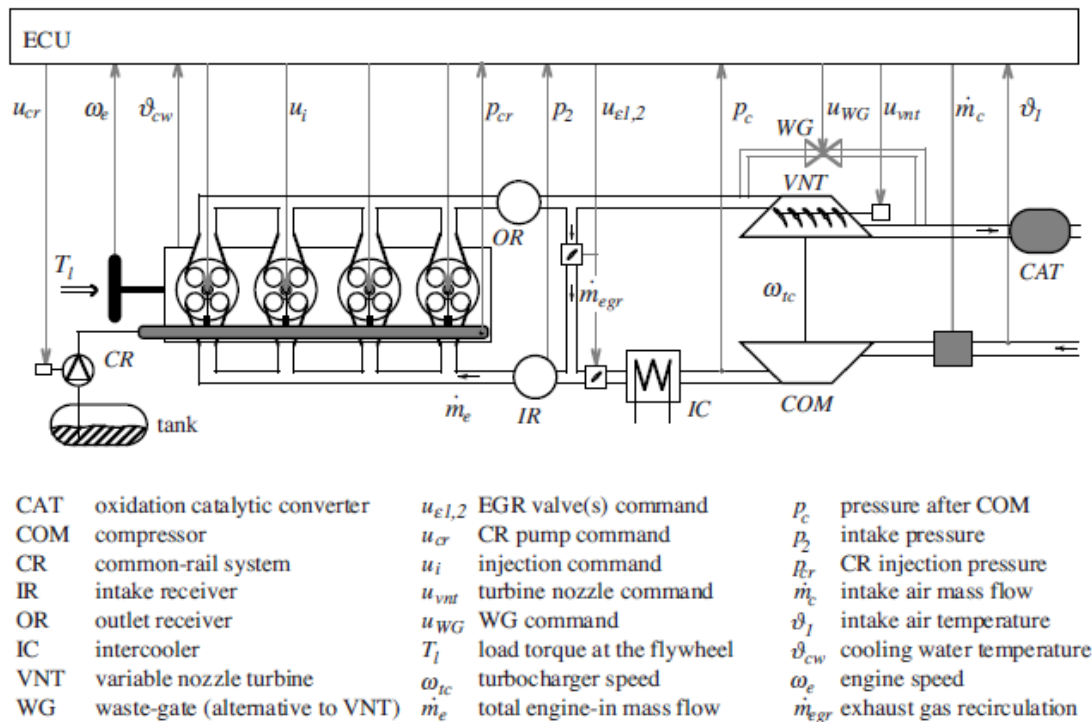


Fig. 1.5. Estructura básica de un motor diésel (Guzzella & Onder, 2010)

Para un motor diésel en particular existen 3 rutas importantes a ser consideradas (ver Fig. 1.6): la ruta del combustible, del aire y del EGR. Un diagrama simplificado de este proceso se muestra en la figura siguiente donde se presenta un esquema de control para un motor diésel. De la figura se observa que la ruta del combustible tiene como salidas el torque, la velocidad y las emisiones de gases, mientras que, sus entradas las obtiene del controlador de inyección. Las entradas del controlador de inyección son el instante de la inyección, la duración de la inyección y la presión de inyección (Guzzella & Onder, 2010). La masa de combustible inyectado se controla mediante un controlador de velocidad, además mediante un medidor de flujo de aire dentro del motor se controla la máxima cantidad de combustible inyectado bajo la base de una relación de aire/combustible predeterminada, previniendo que esta no permanezca debajo de cierto límite o que varíe según las condiciones de operación. Respecto a la ruta del aire, esta se encuentra controlada por el turbocargador, el cual es controlado tomando en cuenta la presión del turbo.

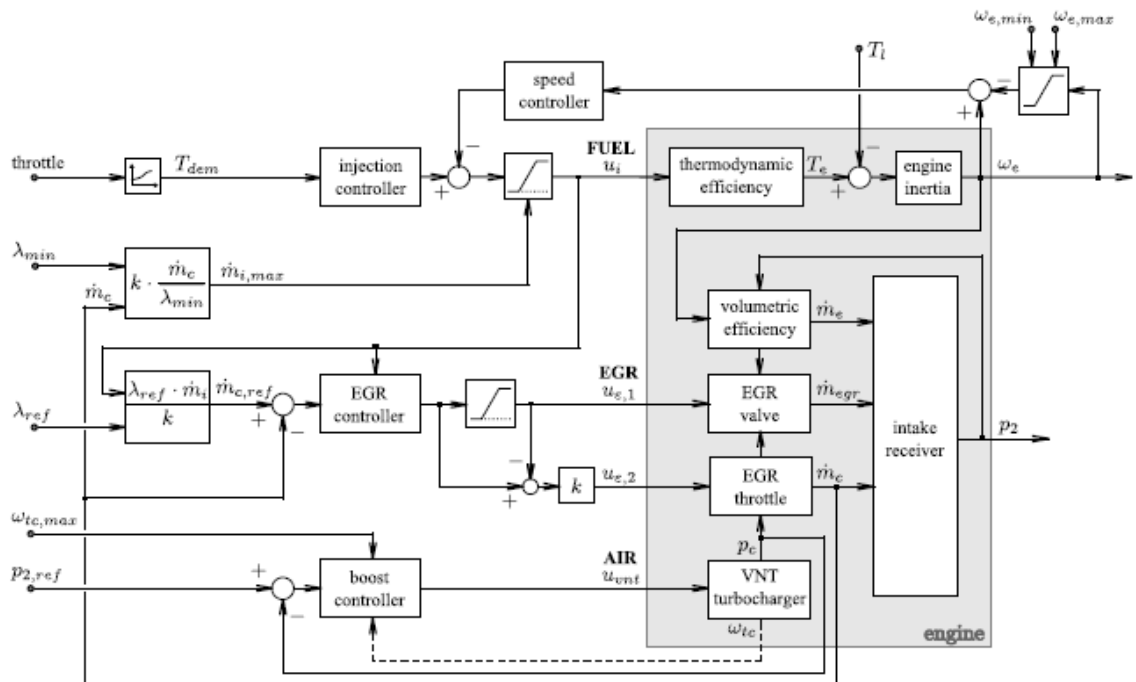


Fig. 1.6. Lazos de control en un motor diésel (Guzzella & Onder, 2010)

Para el control de la inyección de combustible en un motor de automóvil se necesita el estado actual del motor además de las condiciones ambientales, para ello los datos de entrada con los que trabaja una ECU son la posición del pedal de aceleración, el ángulo de rotación del cigüeñal, la velocidad del eje de levas, la presión de carga de aire, la temperatura de aire de admisión, refrigerante y combustible, la velocidad del vehículo, etc. (Bosch, 2003). Sin embargo, es conocido que cualquier estado del motor puede ser descrito principalmente por la presión en el múltiple de admisión (carga) y la velocidad del motor (Hata et al., 1989; Majors et al., 1994).

Las funciones más importantes de la unidad central de control de un motor (ECU) son el control de la relación aire combustible, la regulación anticipada de la chispa de combustión para el control de torque y el control del flujo másico de aire en condiciones de ralentí (Scattolini et al., 1997). Actualmente las ECU usan

dos PI independientes para la ruta del aire, uno para regular la presión en el múltiple de admisión por medio del cambio en la apertura del VGT, mientras que el regulador controla el aire mediante la apertura de la válvula EGR (García-Nieto et al., 2008). Es común el uso de controladores feedforward en los motores ya que tienen la ventaja de no tener problemas de estabilidad, sin embargo, estos controladores necesitan modelos y sensores precisos para las variables principales. Todos los controles dependen fuertemente de la carga y la velocidad en el punto de operación como en el encendido, operación normal y ralentí. En la Fig. 1.7 muestra las variables involucradas en los lazos feedback y feedforward en un motor de combustión.

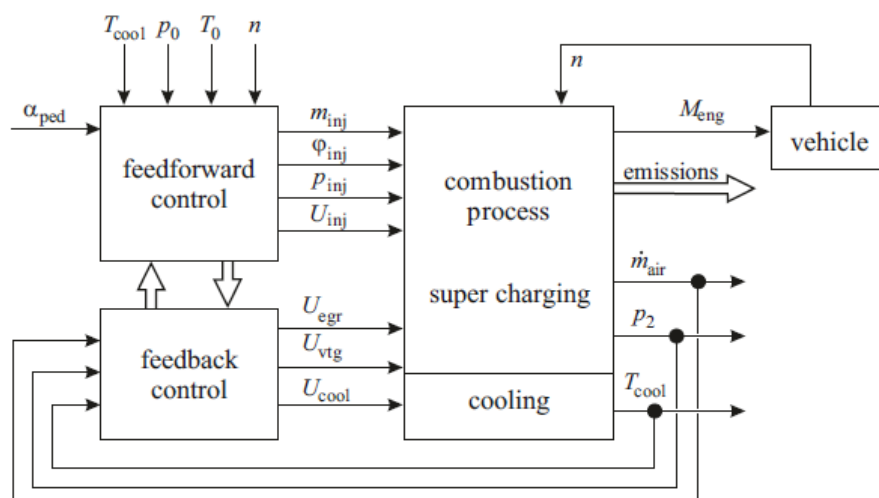


Fig. 1.7. Lazos de control de un motor de combustión (Isermann, 2014)

Estos lazos requieren de la instrumentación necesaria para manejar las variables físicas, esta instrumentación se puede ver en la Fig. 1.8 donde se muestra los sensores y actuadores para un motor diésel común.

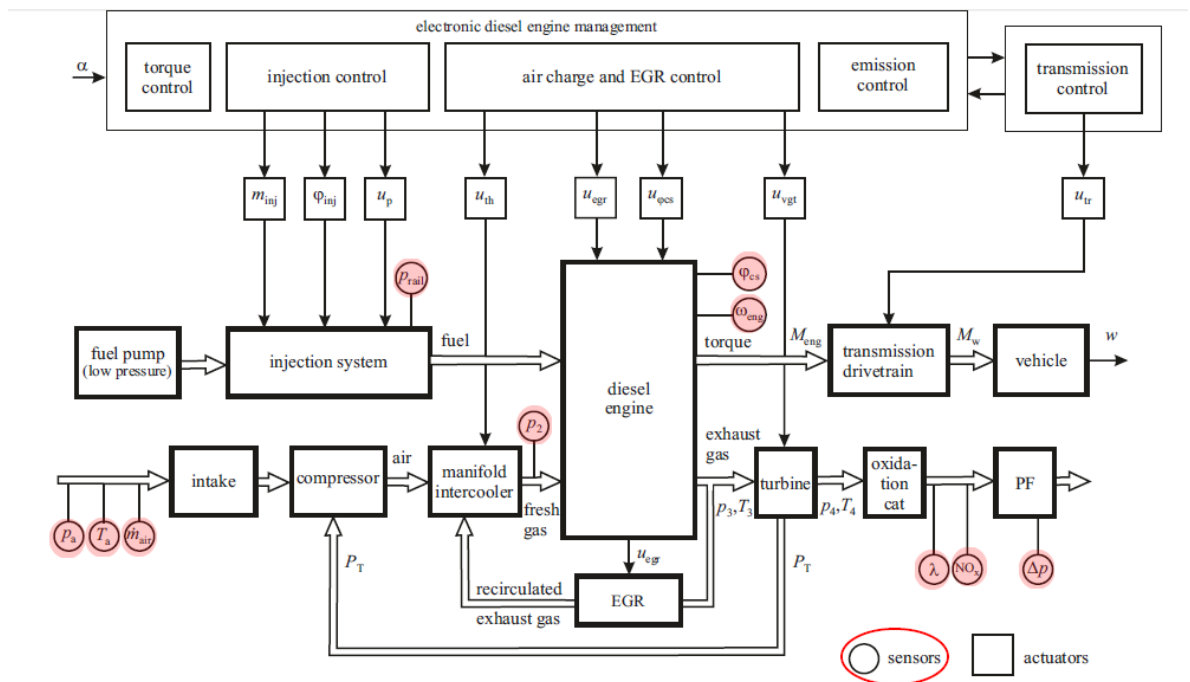


Fig. 1.8. Diagrama de bloques de un motor diésel común (Isermann, 2014)

Con la información expuesta y otras fuentes se elaboró la Tabla 1 donde se resumen los parámetros más importantes a tener en cuenta para el control de un motor diésel.

Tabla 1. Resumen de variables manipuladas y controladas en sistemas de control de motores diésel

Variables\ Referencia	Guzzella 2010	Isermann 2014	Karlsson 2010	Ouladsine 2005	Zhao 2014	Arredondo 2013
<b>Manipuladas</b>						
<b>Ángulo de inyección</b>		X	X	X	X	
<b>EGR</b>	X	X	X		X	
<b>Estallido</b>	X					
<b>Flujo másico de aire</b>	X	X				

<b>Flujo másico de combustible</b>	X	X				X
<b>Posición del pedal</b>		X				
<b>Presión de carga de aire</b>		X				
<b>Presión de inyección</b>		X				
<b>Refrigerante</b>		X				
<b>Señal para el inyector</b>		X				
<b>Torque de carga</b>	X					X
<b>VGT</b>		X	X		X	
<b>Presión de combustible</b>					X	
<b>Controladas</b>						
<b>Aire combustible</b>	X	X				
<b>Ángulo del cigüeñal</b>		X	X			
<b>Estallido</b>	X					
<b>Flujo másico de aire</b>		X			X	
<b>NOx</b>		X	X		X	
<b>Presión de carga de aire</b>	X	X	X	X	X	
<b>Temperaturas</b>		X			X	
<b>Torque</b>		X				
<b>Velocidad</b>	X	X		X	X	X
<b>Torque indicado</b>						X

En cuanto a los lazos identificados en las publicaciones expuestas estas se resumen en la Tabla 2.

*Tabla 2. Lazos más relevantes para el control de motores diésel*

Lazo	Referencia	Variable de entrada	Variable controlada
Control de velocidad	Posición del acelerador	Flujo de combustible Ángulo de inyección Presión de inyección	Velocidad Torque o torque indicado
Control de presión de carga para el turbo	Presión de carga	VGT	Presión de carga de aire
Control de la relación aire-combustible	Relación aire-combustible	EGR Flujo de combustible	Relación aire-combustible

La reseña anterior fue realizada en base a literatura especializada en motores diésel en general. La literatura revisada sobre control electrónico de motores marinos en específico es escasa, sin embargo, las tecnologías de control son básicamente las mismas (Woodyard, 2009). A partir de estas, los principales fabricantes de motores marinos (MAN Diésel, Mitsubishi y Wärtsilä) han desarrollado sus propias tecnologías de actuadores, control y sensores con el fin de reducir el consumo de combustible, reducir las emisiones y monitorear el estado del motor. La configuración del sistema de control de los tres fabricantes es básicamente la misma pues consiste de 2 ECU's, una principal y otra de respaldo en caliente; estas desempeñan el control y supervisión de cada cilindro usando sendos controladores. Respecto a las diferencias de un motor diésel marino con otros tipos, se señala que esta reside principalmente en el aspecto constructivo del motor dado que el régimen de trabajo de un motor marino exige

una alta carga al motor en forma continua por lo que este necesita ser más robusto. (Taylor, 1996). Una diferencia relevante con respecto a motores diésel terrestres es la carga que soporta el motor. Normalmente la carga suele ser casi constante y sostenida por largos periodos de tiempo. Esta carga puede ser modelada teniendo en cuenta el propulsor de la embarcación (Gerr, 2001).

### 1.3 Control PID

Ya que se verificó que los motores diésel trabajan normalmente con controladores PID se hará una breve reseña del estado del arte de este tipo de controlador y una versión mejorada adaptada a sistemas no lineales. El controlador PID (proporcional integral derivativo) es uno de los algoritmos más usados en la industria llegando a encontrarse entre 90 a 95% de los lazos de control (Borase et al., 2020). Este algoritmo es la suma de una acción proporcional al error (diferencia entre la referencia y el valor medido del proceso) más una acción integrativa que se va acumulando en el tiempo corrigiendo la desviación permanente que puede presentar la salida más una acción derivativa que penaliza grandes cambios en el error. La fórmula del algoritmo en su variante paralela es la siguiente:

$$u(t) = K_c \left( e(t) + \frac{1}{T_i} \int e(t) dt + T_d \frac{de(t)}{dt} \right)$$

Donde:

Kc: Ganancia proporcional

Ti: Constante de tiempo integral

Ki=Kc/Ti: Ganancia integral

Td: Constante de tiempo derivativa

Kd=KcTd: Ganancia derivativa

En algunas aplicaciones es conveniente introducir un filtro a la acción derivativa pues esta tiende a tener un desempeño pobre si la señal presenta ruido. Un

esquema del PID en un lazo de control se puede observar en la Fig. 1.9. La llamada sintonización del PID se le llama al proceso de encontrar las ganancias requeridas para controlar un proceso de una forma deseada. Esta sintonización suele hacerse por prueba y error, sin embargo existen otros métodos de sintonización entre los que destacan el método de Ziegler-Nichols , técnicas de algoritmos genéticos, algoritmos basados en relés, etc. (Borase et al., 2020).

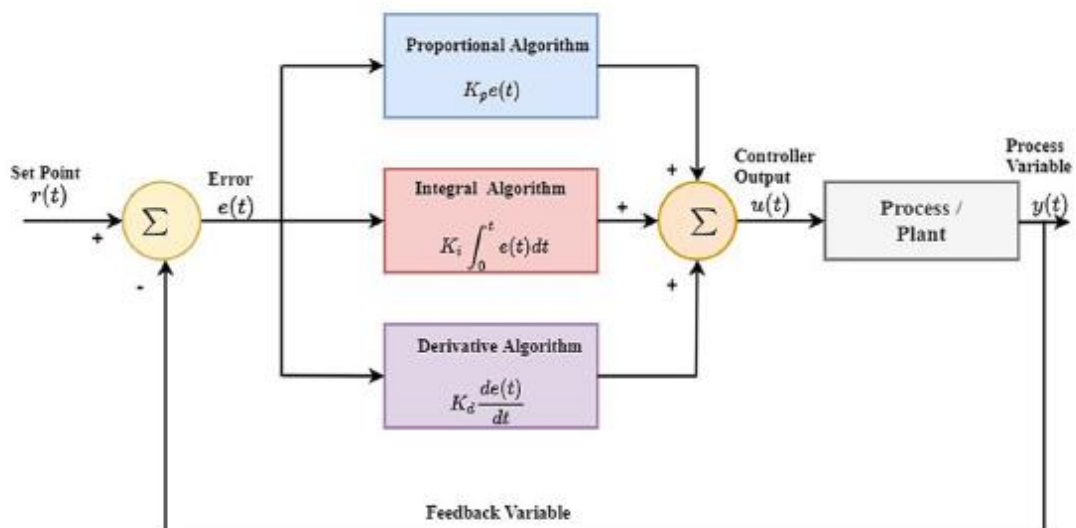


Fig. 1.9. Sistema de control con PID

#### 1.4 Control PID con ganancias programadas

Una de las desventajas del PID convencional es que no puede adaptarse si los parámetros de una planta cambian. Esto no sería problema si el comportamiento de la planta es lineal e invariante en el tiempo, sin embargo, para procesos que requieren pasar por regiones no lineales, el PID estándar puede tener un mal desempeño. Una solución para este problema es trabajar con ganancias programadas. Esta técnica consiste en cambiar las ganancias del PID dependiendo del punto de operación; esto puede ser una tarea ardua si se hace de forma experimental por prueba y error (Åström et al., 1993). La modelación matemática de procesos, sin embargo, junto con algoritmos de sintonización y el uso de computadoras pueden hacer que este proceso sea mucho más rápido.

El proceso para realizar la programación de ganancias es linealizar la planta para varios puntos de operación que se suponen lineales e invariantes en el tiempo y luego diseñar un controlador para cada planta linealizada. Entonces el controlador con ganancias programadas se adapta basado en una tabla de búsqueda o por interpolación. El buen desempeño de esta técnica se basa en la asunción de una variación lenta en el tiempo de los parámetros del planta, pero la desventaja es que esta técnica de diseñar por separado un controlador para cada punto de operación de la planta no garantizaba la estabilidad en los rangos existentes entre los puntos de operación (Rotondo, 2018). Sin embargo, en 1991 se introdujo por primera vez el paradigma de los sistemas con variación lineal de parámetros (LPV) con el trabajo de (Shamma & Athans, 1991). Los LPV están descritos por ecuaciones de la forma:

$$\dot{x}(t) = A(\theta(t))x(t) + B(\theta(t))w(t)$$

$$z(t) = C(\theta(t))x(t) + D(\theta(t))w(t)$$

Donde A, B, C y D son funciones conocidas continuas de un vector  $\theta(t)$  de parámetros que varían en el tiempo. Para una trayectoria  $\theta(t)$  el modelo LPV se convierte en un modelo lineal variante en el tiempo (LTV), de hecho, si  $\theta$  es constante entonces esto deviene en un sistema lineal en el tiempo (LTI). Es así que el proceso de linealización de la planta para puntos de operación puede ser representando por un LPV que en última instancia está representando una familia de LTI parametrizados por la variable de programación  $\theta$ . Debido a esta parametrización, es posible formular un procedimiento de diseño que resulte directamente en un controlador por ganancias programadas. Esta tarea de diseño puede incluso ser formulada como un problema de optimización convexa (Bianchi et al., 2007). Este enfoque de optimización tiene la ventaja que puede ser resuelto por métodos numéricos mediante un software como Matlab. El cual tiene funciones para la linealización y la sintonización del PID, ya que esta última es un problema de optimización que requerirá el establecimiento de objetivos

para minimizar las superficies de sintonización, la cuales contienen los parámetros sintonizables, la variable independiente de estos y la función de ajuste. Un esquema de la implementación de un sistema de control por ganancias programadas se ve en la Fig. 1.10, donde se pueden observar las curvas de sintonización, el PID como un bloque y la planta.

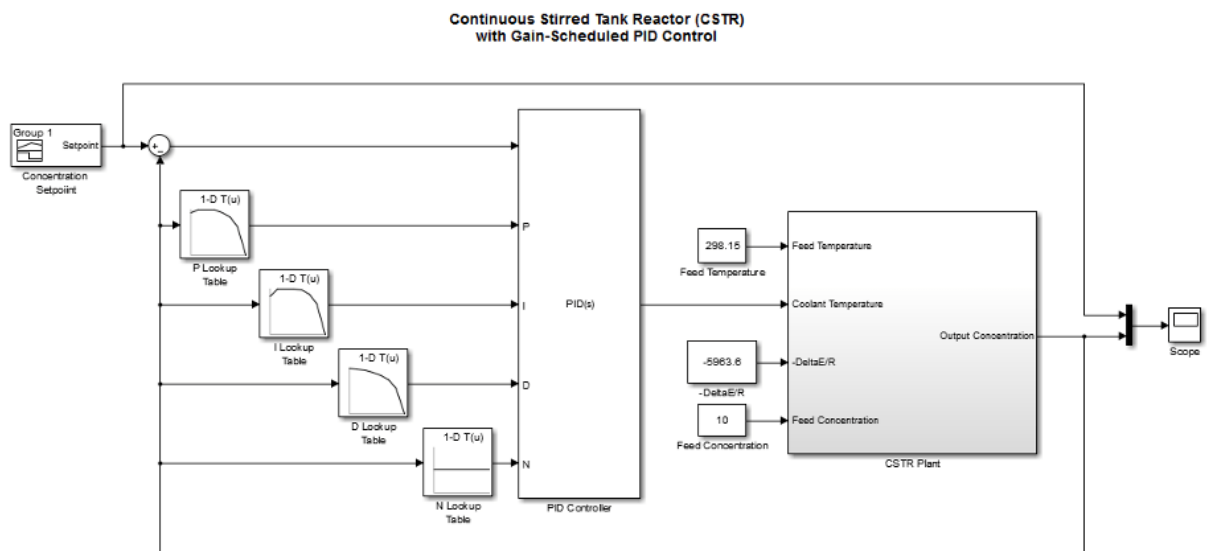


Fig. 1.10. Sistema con PID con ganancia programada (MathWorks, Inc)

El nuevo paradigma basado en LPV ha sido aplicado exitosamente en las últimas 2 décadas en un gran número de aplicaciones como aviones, bioreactores, canales, reproductores de CDs, motores de combustión, motores eléctricos, robots, drones, actuadores electromecánicos, suspensión de vehículos, turbinas de viento, servidores web, etc. (Rotondo, 2018).

## 1.5 Objetivos de la tesis

El presente trabajo tiene por objetivo general el desarrollo del modelado de un motor diésel sometido a un ambiente marino utilizando redes neuronales y desarrollo de su sistema de control de las RPM empleando un PID de ganancias programadas, con el fin de mejorar su desempeño respecto a un PID convencional. Como objetivos específicos se tienen los siguientes:

- Estudio del comportamiento dinámico y caracterización de las variables que gobiernan su desempeño.
- Obtener el modelo matemático del motor diésel marino utilizando redes neuronales.
- Diseñar un controlador PID con ganancias programadas y un PID convencional con el fin de comparar su desempeño de seguimiento de las RPM, eficiencia de uso de combustible y estabilidad ante perturbaciones.
- Desarrollar una propuesta de implementación del controlador avanzado diseñado en un controlador comercial.

## **CAPÍTULO 2. MODELADO MATEMÁTICO DEL MOTOR DIÉSEL MARINO**

### **2.1 Introducción**

En este capítulo se realizará un estudio de las técnicas de modelado matemático con un enfoque al modelado matemático de motores de combustión interna con el fin de determinar el método de identificación más conveniente para un motor diésel. A partir de esto se hará un análisis para escoger las variables a registrar con el fin de aplicar el método de identificación determinado, conseguir una serie de datos y aplicar el método para finalmente tener el modelo matemático que será la base para las pruebas de simulación y diseño del controlador.

### **2.2 Estado del arte del modelado matemático**

El modelado de sistemas es una amplia área en la ingeniería de control la cual se suele subdividir en dos tipos de modelado: el modelado teórico y el modelado experimental. Los principios del modelado teórico para motores de combustión interna siguen una metodología básica la cual está plasmada en los trabajos de (Gawthrop & Smith, 1996; Isermann, 2006; Karnopp et al., 2012). El modelado teórico se fundamenta en el uso de ecuaciones termodinámicas y de dinámica de fluidos para modelar la respuesta del motor. Las ecuaciones fundamentales que se usan para este fin suelen ser:

- Ecuación de balance de masas, energías y momentos
- Ecuaciones fenomenológicas cuando se dan procesos de naturaleza irreversible
- Ecuaciones de balance de entropía si están involucrados varios procesos irreversibles
- Ecuaciones de conexión.

El modelado teórico requiere el conocimiento detallado de la naturaleza del proceso a estudiar por lo cual este tipo de estudio toma más tiempo que si se realiza un estudio de identificación del tipo caja negra. Esto, sin embargo, tiene la ventaja de que cuando se conocen bien las reglas físicas que gobiernan un fenómeno, el modelo resultante es más confiable que el caso de un estudio tipo caja negra. Cuando esto es así, los modelos resultantes se les llaman modelos tipo caja blanca.

El modelado experimental por otro lado, entrega un modelo matemático basado solamente en mediciones de entrada y salida tomadas del proceso a estudiar, esto implica que no es necesario conocer su naturaleza. El modelado de este tipo se conoce como modelo caja negra. Este método suele usarse cuando la dinámica del proceso es muy compleja para ser modelada por conceptos teóricos, cuando se conoce poco acerca de la naturaleza del mismo o cuando los cálculos requieren mucho poder computacional para resolverse. Por ejemplo, el modelo teórico de la concentración de gases de escape en motores de combustión es muy complejo y comprende una alta labor computacional debido a que varios efectos físicos como reacciones químicas, turbulencias, estrangulación de gases, entre otros deben tenerse en cuenta. Es así, que para este caso se usan modelos experimentales (Hafner & Schu, 2000).

En la actualidad se utilizan varios tipos de modelos experimentales los cuales varían según la cantidad de datos teóricos que se incluyen en las ecuaciones, es decir, si bien existen modelos puramente experimentales o denominados modelos tipo caja negra; también se utilizan modelos que en parte son teóricos basado en la física del proceso y en parte son aproximaciones estadísticas del proceso basado en una estructura asumida o también llamados modelos tipo caja gris.

Los procesos estudiados para su identificación pueden tener comportamiento lineal o no lineal. Aquellos procesos con comportamiento lineal son más fáciles de identificar, sin embargo, en el mundo muy pocos procesos tienen este comportamiento, a lo más se suele hacer una identificación lineal dentro de un

rango de operación con este comportamiento. En el caso que sea necesario ampliar el rango de operación del modelo, se debe realizar una identificación no lineal.

Dentro de la identificación no lineal se suele clasificar los métodos en aquellos del tipo estático o dinámico. Los tipos estáticos se encargan de modelar el proceso cuando se encuentra cerca de un punto de operación mientras que los tipos dinámicos tratan de modelar los periodos transitorios del proceso. Dentro de los métodos de identificación estática se encuentran los métodos basados en tablas de búsqueda, la estimación de parámetros y la identificación por redes neuronales. Para procesos dinámicos se tienen las funciones de correlación y estimación de parámetros por el método de mínimos cuadrados. Un esquema de los distintos enfoques para el modelado se puede ver en Fig. 2.1.

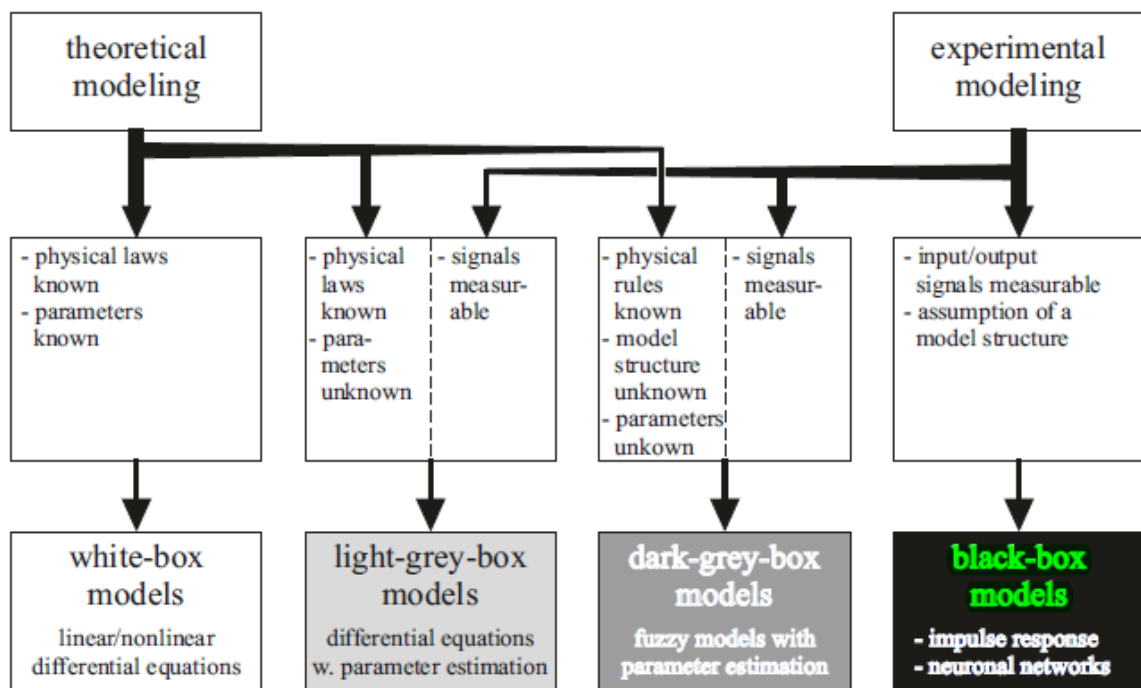


Fig. 2.1. Enfoques para el modelado de sistemas dinámicos (Isermann, 2014)

En la identificación de sistemas no lineales las tablas de búsqueda son ampliamente aceptadas ya que proveen una representación transparente y flexible del sistema (Isermann 2014). En los autos es frecuente el uso de estas tablas que suelen ser alrededor de 100, la naturaleza multivariable del modelo resulta en tablas de más de 2 dimensiones por lo que este modelado tiene el inconveniente de englobar una gran cantidad de información que limita su uso por la falta de espacio en la memoria de la unidad controladora. Sin embargo, en algunos casos, la alta no-linealidad de los motores de combustión interna obliga a utilizar tablas de búsqueda 3D para el funcionamiento del control feedforward. Esta alta no linealidad quedo confirmada al realizarse un estudio previo de identificación a un motor de prueba con un modelo lineal tipo ARMAX el cual no emulaba correctamente el comportamiento del motor.

Un método más adecuado que las tablas de búsqueda cuando se tienen muchas variables en juego es el método por redes neuronales. Este método de identificación no paramétrica tiene distintas variantes, el método se basa en un modelo de neuronas que mapean las entradas con las salidas y a través de varias iteraciones se entrena o aprende a medida que procesa más datos. Los parámetros del modelo entonces se sintonizan y el modelo resultante puede predecir con fiabilidad las salidas para las entradas ingresadas. Este método es muy utilizado en el caso de sistemas que tienen naturaleza no lineal.

Sobre la aplicación de esta área de la ingeniería de control en el campo de motores de combustión interna, se menciona el estudio realizado por (Isermann et al., 1999) donde se hace un breve repaso del modelado actual de motores. Primero el modelado teórico el cual resulta difícil dada la complejidad de los procesos involucrados en la combustión y el otro es el modelado experimental o por identificación donde se indica que, para sistemas no lineales existe diferentes enfoques; los cuales, se infiere, no se encuentra en una etapa madura para el tiempo en que fue publicado el estudio. Un trabajo más actual es el realizado por Dejong, Cunjia et al. (2014) donde indican que no existe un modelo matemático apropiado para la ruta del combustible en un motor, refiriéndose a los modelos teóricos. A pesar de que existen sensores altamente sofisticados para motores

diésel, a nivel del sistema de control de la ruta del combustible se utiliza un sistema de control del tipo “feedforward”, los cuales trabajan con un modelo experimental basado en tablas de búsqueda. A partir de esto se observa la dificultad de modelar por métodos teóricos el comportamiento de un motor de combustión por lo que resulta conveniente el uso de redes neuronales para el modelado.

Otros métodos de identificación no lineal como los modelos NARX han sido usados también en modelado y control de motores (De Nicolao, Rossi et al. 1999), sin embargo actualmente las investigaciones en identificación de motores de combustión interna se centran en la identificación con redes neuronales (Kalogirou 2003), incluso en investigaciones más recientes se aboga por la identificación con redes neuronales pues requiere menores costos y tiempos de desarrollo para modelos altamente no lineales como un motor de combustión (Manikandan Janakiraman, Nguyen et al. 2013).

Sobre la aplicabilidad de la inteligencia artificial en motores, se menciona el artículo realizado por (Kalogirou, 2003) donde se detalla el estado del arte de la inteligencia artificial aplicada al control de motores de combustión interna. El autor reseña varios trabajos aplicativos de inteligencia artificial donde se detallan las 5 técnicas más utilizadas: sistemas expertos, redes neuronales, algoritmos genéticos, lógica difusa y redes neuro-difusas. De las técnicas aplicadas a motores de combustión interna, la más utilizada por gran diferencia es la técnica de redes neuronales. De las 14 referencias reseñadas por el autor se destacan los trabajos de (Korb et al., 1999; Muller et al., 1997) referentes al control y modelización de motores de combustión respectivamente. En el artículo de Muller, Hemberger et al. (1997) se utiliza nuevos sensores para medir la presión en la cámara de combustión y analizar esta señal con redes neuronales con el propósito de controlar el momento de la ignición. En cuanto al artículo de Korb, Jorgl et al. (1999) se presenta un sistema de redes neuronales radiales para obtener el modelo de un motor a gas de 2 entradas y 2 salidas con el fin de ser usado en el diseño de un sistema de control de velocidad.

La aplicación de las funciones para control de motores está basada principalmente en sistemas expertos, pero debido al mayor número de variables, el valor óptimo de operación rara vez puede determinarse manualmente (Hafner et al., 2002). En comparación con las tablas de búsqueda basadas en mapas tridimensionales, las redes neuronales tienen la ventaja de manejar con mayor facilidad sistemas multidimensionales en una forma compacta. Finalmente como una observación adicional se encontró que la aplicación de modelos lineales como ARMAX no son tan efectivos como las redes neuronales según se rescata de la tesis realizada por (Romero Ugalde, 2008) donde se hace una comparación del método de identificación ARMAX y un método de identificación lineal y no lineal de redes neuronales para varios tipos de planta, encontrándose que la identificación no lineal con redes neuronales es la que posee mayor verosimilitud con la respuesta de la planta real.

En base a lo anterior se aplicará el método de identificación por redes neuronales no lineal para obtener el modelo matemático del motor de combustión interna; para ello se estudiará el estado del arte de esta técnica de identificación de sistemas.

### **2.3 Modelos basados en redes neuronales**

El método de identificación no lineal por redes neuronales que se usa regularmente es el de perceptrón multicapa (MLP) y la de funciones básicas radiales (RBS). Estas arrojan una buena aproximación del comportamiento no lineal, sin embargo cuando se tiene muchas variables la estructura puede volverse demasiado compleja con muchos parámetros ajustables (Isermann 2014). Una alternativa a estos métodos es la identificación con modelos locales lineales que varían sus parámetros con el punto de operación o con las amplitudes de las señales de entrada.

Para el caso de sistemas dinámicos, la información a modelar son series de tiempo, las cuales son más apropiadas de modelarse a través de redes neuronales recurrentes. Una arquitectura poderosa de redes neuronales

recurrentes está basada en los modelos no lineales autoregresivos con entrada exógena (NARX). Las ventajas de las redes neuronales NARX usando algoritmo de descenso de gradiente es la más efectiva en el aprendizaje y en la generalización que otras redes neuronales (Diaconescu, 2008). Además se ha observado que este tipo de red se utilizó con éxito para modelar motores y sus índices de desempeño (Janakiraman et al., 2013; Raptodimos & Lazakis, 2020; Yu et al., 2018).

Una red neuronal NARX se puede representar como la siguiente función:

$$y(k) = f[u(k-1), \dots, u(k-n_u), y(k-1), \dots, y(k-n_y)]$$

Donde:

$u(k) \in \mathbb{R}^{u_d}$  y  $y(k) \in \mathbb{R}^{y_d}$  representan las entradas y salidas del sistema respectivamente

$k$  es el índice del tiempo discretizado

$f(\cdot)$  es la correspondencia no lineal

$n_u$  y  $n_y$  es el número de entradas y salidas pasadas que sirven para la predicción

$u_d$  e  $y_d$  son las dimensiones de la entradas y salidas.

Si tomamos  $x$  como un vector de entrada aumentado donde se consideran las mediciones de entrada y salida del sistema como:

$$x = [u(k-1), \dots, u(k-n_u), y(k-1), \dots, y(k-n_y)]^T$$

La secuencia puede ser tomada como data de entrenamiento apropiada para la red neuronal:

$$\{(x_1, y_1), \dots, (x_n, y_n)\} \in (X, Y)$$

El objetivo de la red neuronal será minimizar una función de costo la cual se puede tomar como la del error cuadrático promedio (MSE):

$$R(w) = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i(x, w))^2$$

Donde  $w$  representa los parámetros de la red y  $\hat{y}(x, w)$  representa las predicciones de la red. Como los elementos pasados son entradas de la red neuronal, esta aprende la dinámica del sistema. En base a estas ecuaciones la red puede tener 2 arquitecturas, una arquitectura serie paralelo que puede realizar predicciones de un paso y una arquitectura que puede realizar predicciones a futuro de varios pasos. Ambas arquitecturas se muestran en la Fig. 2.2. En general el entrenamiento se realiza usando la arquitectura serie paralelo ya que la salida real realimentada se tiene en la data de entrenamiento (Narendra & Parthasarathy, 1991). Esto tiene la ventaja de crear un modelo más preciso, luego del entrenamiento se puede realizar la retroalimentación con la salida estimada obteniéndose un modelo feedforward que se alimenta solo de la entrada exógena.

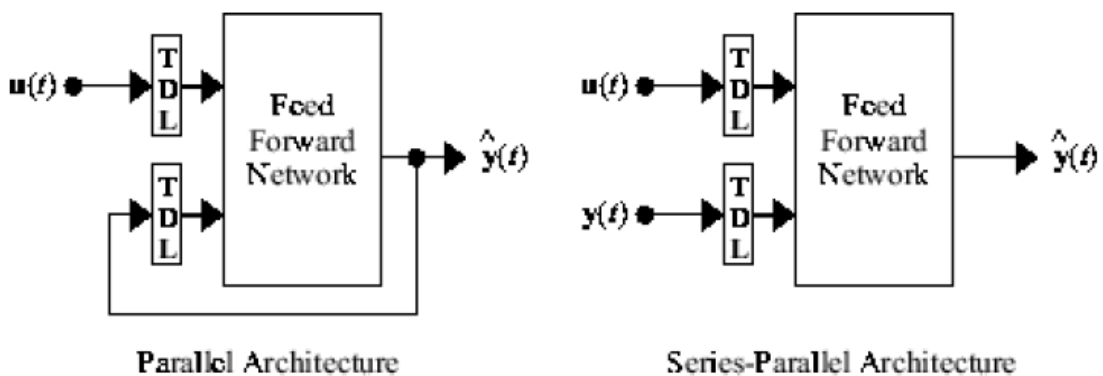


Fig. 2.2. a) Arquitectura Paralela b) Arquitectura serie-paralela (Beale et al., 2020)

La red neuronal bajo el modelo del perceptrón multicapa se construye usando uno o varias capas ocultas donde los nodos conectan directamente entradas con salidas. Para una red con una capa de entrada, una capa de salida y una capa oculta, la salida estimada se puede calcular con:

$$\hat{y} = f_2(b_2 + W_2^T f_1\{b_1 + W_1^T(x)\})$$

Donde  $f_1$  y  $f_2$  son las funciones de activación de la capa oculta y de la capa de salida respectivamente. Los bloques que dicen TDL (Tapped Delay Line) representan la serie de valores pasados que entran a la red neuronal, estos también se llaman retardos.

Para el entrenamiento de la red neuronal existen varios algoritmos basados en el algoritmo de retropropagación de errores, entre los cuales destacan el algoritmo Levenberg-Marquardt (LM) (Hagan & Menhaj, 1994) que es especialmente rápido en la minimización del error cuadrático y una variación de este que aparte de minimizar el error también busca minimizar los parámetros o pesos de la red llamado retropropagación con regularización Bayesiana (Dan Foresee & Hagan, 1997).

La configuración de la red neuronal entonces consistirá del número de neuronas de la capa oculta, los retrasos de la entrada exógena y de las salidas estimadas, las funciones de activación y el algoritmo de optimización. Un esquema de ejemplo de una red NARX su arquitectura paralela, lista para su uso como modelo se muestra en Fig. 2.3.

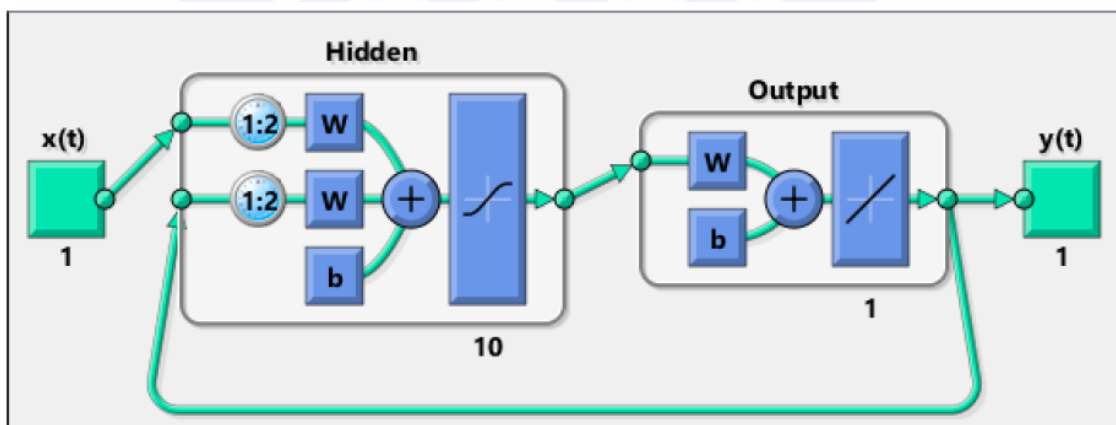


Fig. 2.3. Ejemplo de una red neuronal NARX (Beale et al., 2020)

En el ejemplo de la Fig. 2.3 pueden notarse los retrasos 1 al 2 tanto en la línea de la entrada exógena  $x(t)$  y de las salidas realimentadas.  $W$  y  $b$  representan los pesos o parámetros de la red neuronal. El bloque con una función sigmoidea representa función de activación de la capa oculta la cual tiene 10 neuronas. Más

a la derecha se observa que la capa de salida tiene una función de activación lineal con 1 neurona. El entrenamiento se realiza por épocas, en cada época se espera una reducción de la función de costo hasta lograrse una convergencia. Entrenar por demasiadas épocas puede ser perjudicial porque puede sobreentrenar la red neuronal haciéndola susceptible a seguir el ruido en lugar de la verdadera señal. Para contrarrestar se puede detener el entrenamiento previamente comparando con la data de validación y contando el número de épocas que las estimaciones empiezan a divergir.

#### **2.4 Modelado matemático del motor diésel**

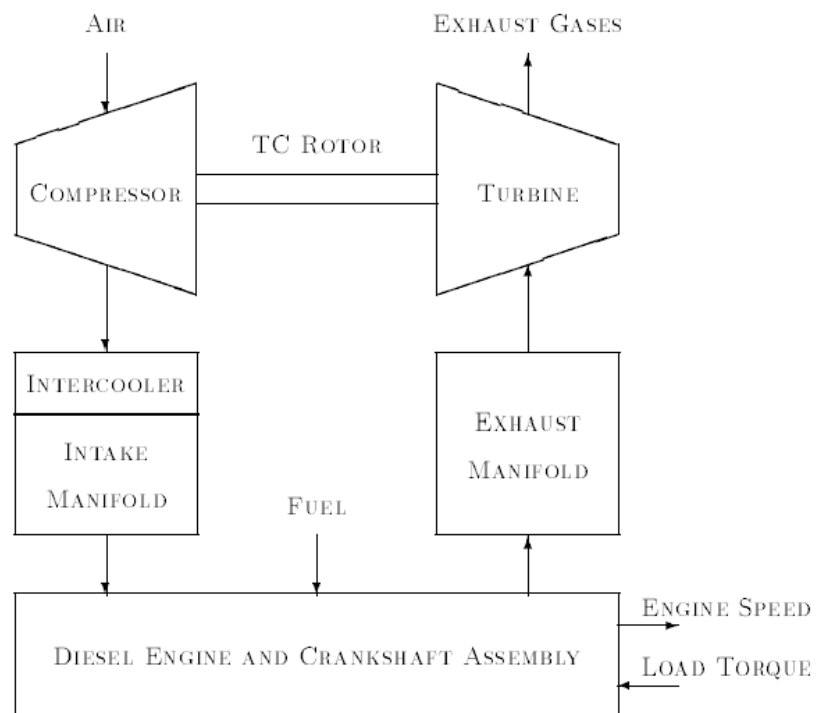
Para realizar el modelado matemático de un motor diésel es necesario estudiar y analizar la naturaleza de esta máquina. Esto es una tarea compleja que puede ser simplificada a una serie de ecuaciones. La literatura acerca del modelado matemático de motores diésel marinos es escasa; sin embargo, las diferencias entre un motor diésel marino y uno convencional realmente son de aspecto constructivo y no desde el punto de vista funcional, por lo que podemos apoyarnos en la literatura de modelado para motores diésel en general. La metodología del modelado será explorar las variables más relevantes para el control de velocidad del motor diésel y luego hacer un modelado por redes neuronales usando las variables.

En la literatura se encontraron desde modelos bien básicos (Farouk et al., 2012) hasta modelos más complejos para sistemas marinos (Alegret et al., 2015; Altosole et al., 2017, p. 20, 2019; Llamas & Eriksson, 2019; Nahim et al., 2015; Tadros et al., 2015) y motores diésel en general (Arredondo Díaz, 2013; Hellstrom, 2005; Jensen et al., 1991; Kao & Moskwa, 1995; Omran et al., 2009, p. 200; Yanakiev & Kanellakopoulos, 1995; Yin et al., 2017). En base a este sondeo se presentarán las ecuaciones que gobiernan la naturaleza de un motor diésel enfocándonos en explorar las variables que afectan a las RPM del motor.

El motor diésel sea marino (Altosole et al., 2019) como general (Yanakiev & Kanellakopoulos, 1995) puede agruparse en los siguientes subsistemas:

1. Cilindros
2. Colector de admisión
3. Colector de escape
4. Compresor
5. Turbina
6. Dinámica del turbocompresor
7. Torque del motor

Esquemáticamente esto se puede observar en las Fig. 2.4 y Fig. 2.5.



*Fig. 2.4 Esquema de un motor diésel (Yanakiev & Kanellakopoulos, 1995)*

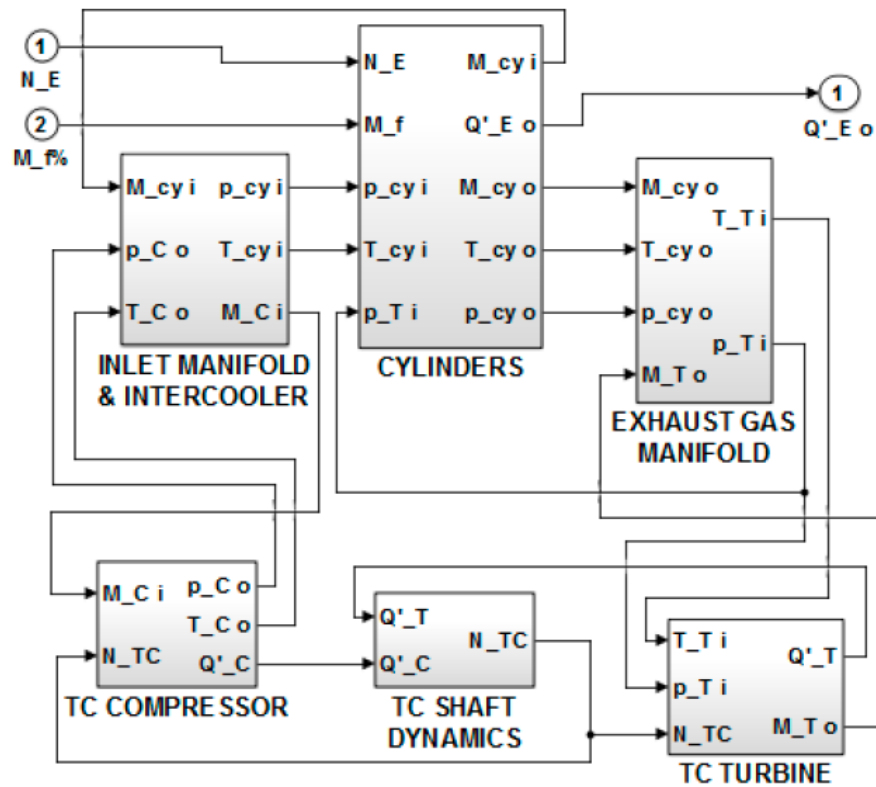


Fig. 2.5. Diagrama de bloques de un motor diésel marino (Altosole et al., 2019)

De la Fig. 2.5 puede notarse que las variables más relevantes para nuestro modelo se encuentran en el grupo de los cilindros (Ne: RPMs del motor, Mf: porcentaje de flujo de combustible, Qe: torque indicado); sin embargo, en este modelado las RPM no figuran como salida, solo figura como salida el torque en el motor. De (Omran et al., 2009; Tadros et al., 2015) podemos sacar expresiones que involucran el torque y las rpm. En el caso del torque producido por el motor tenemos:

$$T_E = T_{ind} - T_{fric}$$

Donde:

Te: Torque efectivo o torque al freno

Tind: Torque indicado o torque debido a los gases de combustión

Tfric: Torque de fricción

Las RPM producidas por el motor puede hallarse a través de la 2da ley de Newton (Tong et al., 2019; Yanakiev & Kanellakopoulos, 1995):

$$\sum T = J_e \dot{\omega}_e$$

Donde la sumatoria de torques vendría estar dado por:

$$\sum T = T_E - T_{carga}$$

Donde:

$T_{carga}$  = Torque de la carga

$J_e$  = Momento de inercia

$\omega_e$ : Velocidad angular

De (Nahim et al., 2015; Omran et al., 2009; Tadros et al., 2015; Yanakiev & Kanellakopoulos, 1995; Yin et al., 2017) se tienen formulas parecidas relativas al torque indicado, de las que se puede concluir que el torque indicado es una función del flujo de combustible, su poder calorífico y la eficiencia de combustión, esta última puede ser simplificada a un polinomio de segundo orden dependiente de las RPM por lo tanto considerando que el poder calorífico es una constante, entonces:

$$T_{ind} = f(\dot{m}_f, RPM)$$

En el caso del torque de fricción, las mismas fuentes coinciden en indicar que este torque depende de las RPM a través de un polinomio de segundo orden, por lo tanto:

$$T_{fric} = f(RPM)$$

En el caso del torque de carga ya que el objetivo es simular la operación de un motor diésel marino, la carga sería el arrastre generado en el propulsor. De (Gerr, 2001) la demanda de potencia del propulsor para un bote promedio está dada por:

$$PHP = C * RPM^{2.7}$$

Donde C es una constante. De esta ecuación podemos tener el torque al dividir entre las RPM. Las otras variables involucradas en el comportamiento de un motor diésel como las mostradas en la Fig. 2.5, son variables internas. Ya que el modelado se realizará con redes neuronales, la dinámica interna será identificada a través del proceso de aprendizaje de la red neuronal.

Estas variables internas que en la literatura se agrupan en las llamadas ruta del aire y la ruta del combustible (García-Nieto et al., 2008; D. Zhao et al., 2014), pueden ser en algunos casos controladas; sin embargo estos están más orientados a controlar los gases contaminantes producidos por el motor.

Para esta tesis nos enfocaremos en el control de velocidad del motor diésel por lo que podemos obviar para el modelado, los lazos relativos al control de gases contaminantes. Con esto podemos escoger finalmente las siguientes variables que debe incluir nuestro modelo matemático.

#### Variables de entrada

Flujo másico de combustible

Velocidad angular del motor

#### Variables de salida

Torque al freno

Velocidad angular del motor

Estas variables también fueron utilizadas para realizar modelado de motores con éxito tal como aparece en (Yu et al., 2018; J. Zhao, Hu, et al., 2019).

#### Datos experimentales del motor diésel

Dado que las pruebas experimentales requieren una serie de sensores y bancos de pruebas a los que no se tiene acceso, se optó en esta tesis por la búsqueda de un conjunto de datos de libre acceso. En el trabajo de (J. Zhao, Zhou, et al.,

2019) se tiene acceso a un conjunto de datos de un motor diésel del fabricante Changchun FAW Sihuan Engine Manufacture. Las características de la adquisición de datos es la siguiente:

- Motor diésel de 4 cilindros y 2.7L de cilindrada.
- La adquisición de señales se realizó usando un sistema analógico digital de alta velocidad durante 1800 segundos bajo 3 condiciones de aceleración: bajas, medias y altas.
- Periodo de muestreo de 0.1s

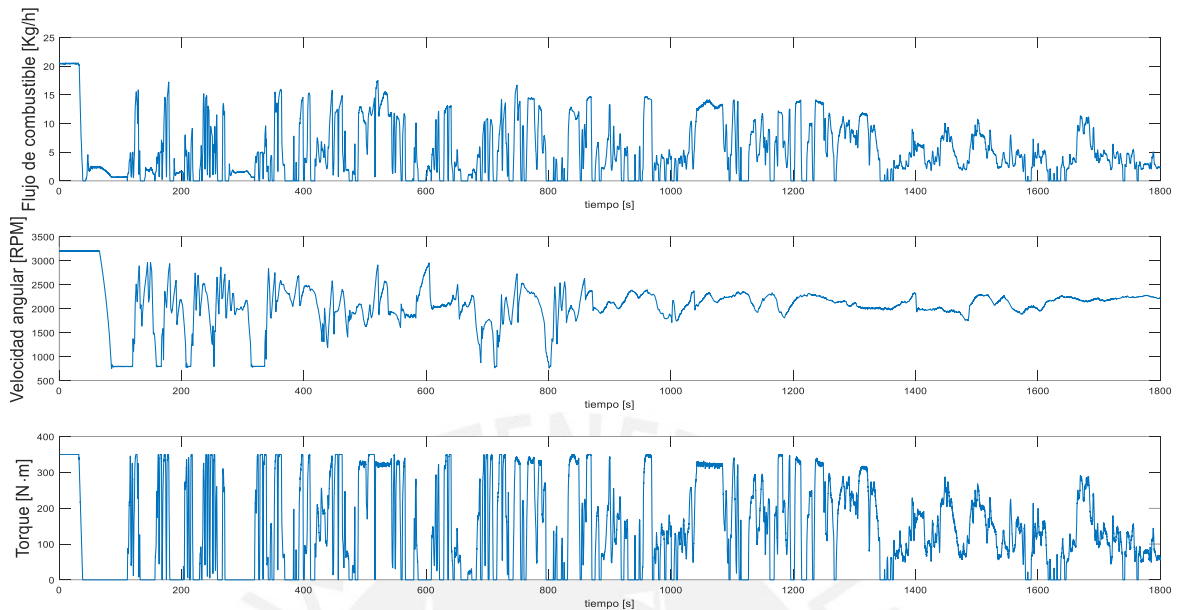
Los datos adquiridos son de 7 tipos de variables los cuales se describen a continuación:

- Flujo másico de aire [mg/ciclo] abreviado con las siglas “AIM”
- Posición del acelerador [%] abreviado con las siglas “APS”
- Masa inyectada de combustible [mg/ciclo] abreviado con las siglas “FIR”
- Presión del riel de combustible [hPA] abreviado con las siglas “RP”
- Inicio de la inyección de combustible [°] abreviado con las siglas “SOI”
- Velocidad [rpm] con la variable de nombre “speed”
- Torque [Nm] con la variable de nombre “torque”

El flujo másico de combustible en Kg/h puede obtenerse con la siguiente formula:

$$\dot{m}_e = FIR * speed * \frac{1 \text{ ciclo}}{2 \text{ rev}} * \frac{60 \text{ min}}{1 \text{ h}} * \frac{1 \text{ Kg}}{10^6 \text{ mg}}$$

La data para las 3 condiciones de carga se tomó en una sola corrida y se muestra en la Fig. 2.6.



*Fig. 2.6. Serie de tiempo de las variables para el modelado*

Esta data se usará para hacer la identificación de sistemas, el torque de la fuente es el torque al freno por lo que no se tiene el torque de la carga; sin embargo, la información disponible servirá para hacer un modelado por redes neuronales tomando como entradas el flujo de combustible y la velocidad angular y como salida el torque al freno; luego, estimando el torque de carga de forma teórica, dividiendo entre el momento de inercia e integrando, podemos obtener la velocidad angular de salida del motor que servirá como realimentación para la red neuronal.

## 2.5 Identificación por redes neuronales

Ya que los datos son series de tiempo se requiere una red neuronal recursiva, en ese sentido una red neuronal no lineal autoregresiva con entrada exógena (NARX) sería apropiada tal como se expuso en el estado del arte del modelado. Para la selección de parámetros de la red neuronal se hicieron varias pruebas con el número de neuronas, los retardos y las funciones de activación. Para predecir sistemas dinámicos se trabaja generalmente con una sola capa oculta y una función de activación en la capa de salida. Para la función de costo se usó

el error medio cuadrático (MSE). Para el desarrollo nos apoyaremos del software Matlab y sus complementos Simulink, Control System Toolbox y Deep Learning Toolbox.

Como se mencionó anteriormente en la sección 2.4 sobre los datos experimentales del motor diésel, la data se realizó para 3 tipos de condiciones por lo que se tienen 3 series de tiempo de 600 segundos. La primera serie de tiempo tiene al inicio datos que se consideraron no representativos de la naturaleza del motor pues parecen dar valores saturados por lo que se utilizaron solo los últimos 500 segundos. Estas 3 series de tiempo fueron repartidas en un conjunto de entrenamiento, uno de validación y uno de prueba. La red neuronal será entrenada con el primer conjunto, la data de validación nos permitirá detener el entrenamiento a tiempo para evitar ajustar demasiado la red, esto introducirá un pequeño sesgo a nuestro modelo; finalmente el modelo será comparado con el conjunto de prueba, el cual no tiene sesgo alguno pues no se hacen ajustes a la red entrenada con este conjunto. Los segmentos para repartir las 3 series de tiempo son los siguientes:

Datos de entrenamiento: 80%

Datos de validación: 10%

Datos de prueba: 10%

A continuación, se muestra el script para preprocesar la data:

```
u = [FR speed]; % Entradas: flujo de combustible y velocidad angular
z = torque; % Salida: torque al freno

scatter3(u(1001:18000,1),u(1001:18000,2),z(1001:18000,1))

% Entradas
U1 = num2cell(u(1001:6000,:),'',1); % Tomamos 500 seg de la 1ra serie de tiempo
U2 = num2cell(u(6001:12000,:),'',1); % 600 seg. a carga media
U3 = num2cell(u(12001:18000,:),'',1); % 600 seg. a carga ligera
```

```

UU = catsamples(U2,U3,U1,'pad'); % Concaquetamos las series de tiempo

% Salidas
Z1 = num2cell(z(1001:6000,:)); % Tomamos 500 seg de la 1ra serie de tiempo
Z2 = num2cell(z(6001:12000,:)); % 600 seg. a carga media
Z3 = num2cell(z(12001:18000,:)); % 600 seg. a carga ligera

ZZ = catsamples(Z2,Z3,Z1,'pad'); % Concaquetamos las series de tiempo

% Configuramos la data para que pueda ser usada por las funciones
X = tonndata(UU,true,true); % UU
T = tonndata(ZZ,true,true); % ZZ

```

Como se expuso en el estado del arte del modelado lo más recomendable es realizar el entrenamiento en la arquitectura serie-paralelo también llamada en lazo abierto. Por lo tanto, el procedimiento de las pruebas será el siguiente:

1. Configurar los parámetros de la red neuronal (algoritmo de retropropagación, número de neuronas, función de activación, retardos)
2. Realizar el entrenamiento en lazo abierto hasta que se cumplan algunas de las condiciones que evita el sobre entrenamiento (número de fallas en la data de validación, número máximo de épocas)
3. Cerrar el lazo y reentrenar la red.
4. Medir el desempeño a través de la función de costo.
5. Analizar el resultado, si no es satisfactorio volvemos al primer paso.

Para la primera prueba se comenzó con los siguientes parámetros:

Algoritmo: Levenberg-Marquardt

Número de neuronas de la capa oculta: 10

Función de activación de la capa oculta: sigmoidea

Retardos a la entrada: 1

Retardos a la salida: 1

Esquemáticamente esto se muestra en la Fig. 2.7.

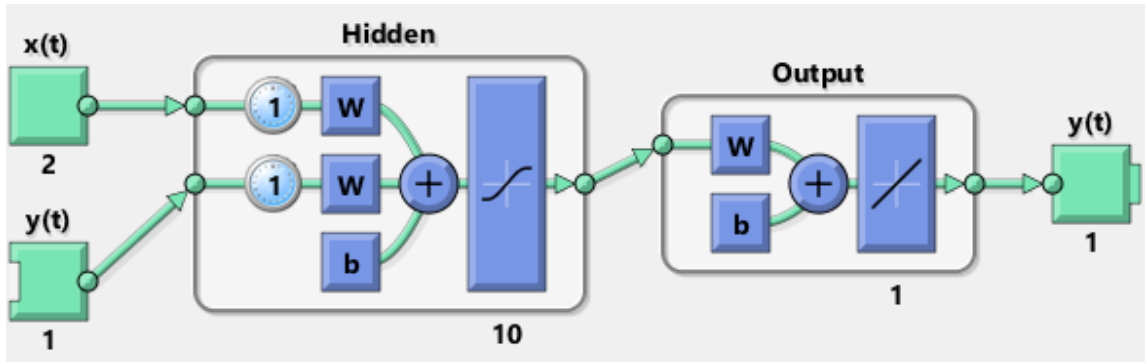


Fig. 2.7. Esquema de la primera prueba

Esto se configuró con el siguiente script basado en código de configuración del Deep Learning Toolbox de Matlab para redes NARX:

```
%% Configuración
trainFcn = 'trainlm'; % Algoritmo Levenberg-Marquardt
inputDelays = 1; % retrasos para la entrada
feedbackDelays = 1; % retrasos para la salida retroalimentada
hiddenLayerSize = 10; % neuronas para la capa oculta

net = narxnet(inputDelays,feedbackDelays,hiddenLayerSize,'open',trainFcn);

net.sampleTime = 0.1;
net.divideFcn = 'divideblock'; % Divide la data en bloques
net.divideMode = 'time'; % Divide la data por tiempo
net.divideParam.trainRatio = 80/100; % 80% entrenamiento
net.divideParam.valRatio = 10/100; % 10% validación
net.divideParam.testRatio = 10/100; % 10% pruebas
net.performFcn = 'mse'; % Funcion de costo MSE

net.plotFcns = {'plotperform','plottrainstate','ploterrhist', ...
               'plotregression','plotresponse','ploterrcorr','plotinerrcorr'};

net.trainParam.max_fail = 10; % Máximo número de fallas de validaciones
```

```

net.trainParam.epochs = 6000; % Número máximo de épocas

net.layers{1}.transferFcn = "tansig"; % Función de costo sigmoidea
net.layers{2}.transferFcn = "purelin"; % Función de costo lineal

```

En tanto para el entrenamiento en lazo abierto y en lazo cerrado se continuó configurando el código adaptándolo al problema del motor diésel:

```

%% Entrenamiento en lazo abierto
% Introducimos los retraso en la data y se realiza el entrenamiento
[x,xi,ai,t] = preparets(net,X,{},T);
[net,tr] = train(net,x,t,xi,ai);

%% Medimos el desempeño en lazo abierto
[x,xi,ai,t] = preparets(net,X,{},T);
y = net(x,xi,ai);
trainTargets = gmultiply(t,tr.trainMask);
valTargets = gmultiply(t,tr.valMask);
testTargets = gmultiply(t,tr.testMask);
trainPerformance = perform(net,trainTargets,y);
valPerformance = perform(net,valTargets,y);
testPerformance = perform(net,testTargets,y)

%% Cerramos el lazo
[y,xf,af] = net(x,xi,ai);
netc = closeloop(net,xf,af);
netc.name = [net.name ' - Lazo cerrado'];
%view(netc)
[xc,xic,aic,tc] = preparets(netc,X,{},T);
yc = netc(xc,xic,aic);
closedLoopPerformance1 = perform(netc,tc,yc)

%% Reentrenamos la red en lazo cerrado
[netc,tr] = train(netc,xc,tc,xic,aic);
yc = netc(xc,xic,aic);
closedLoopPerformance2 = perform(netc,tc,yc)

```

En el entrenamiento del lazo abierto puede observarse la disminución de la función de costo en la Fig. 2.8.

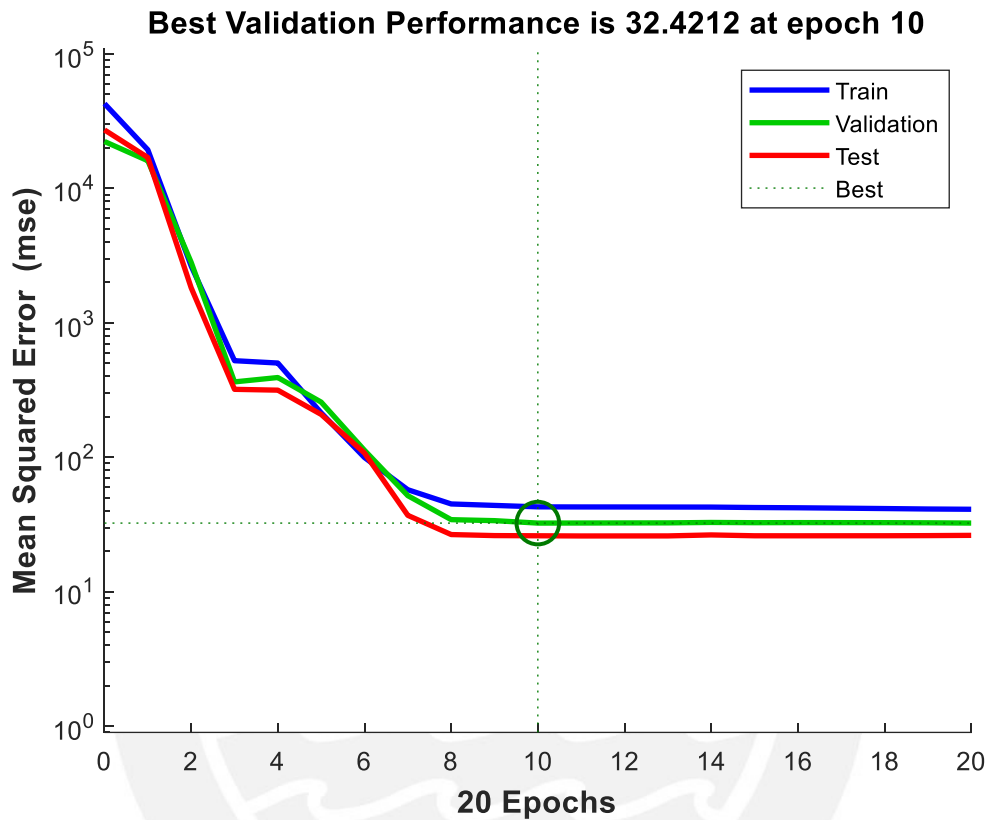


Fig. 2.8. Función de costo vs Épocas

La convergencia se logra aproximadamente a las 10 épocas, pero luego empieza a diverger en relación a la data de validación. En este caso el MSE para la validación resulta ser 32.4212. Si analizamos la relación entre las salidas estimadas y las reales para cada conjunto podemos observar que en general se estima con relativa precisión, el indicador estadístico  $R^2$  es casi 1 para los 3 conjuntos y para la data en su totalidad como se ve en la Fig. 2.9.

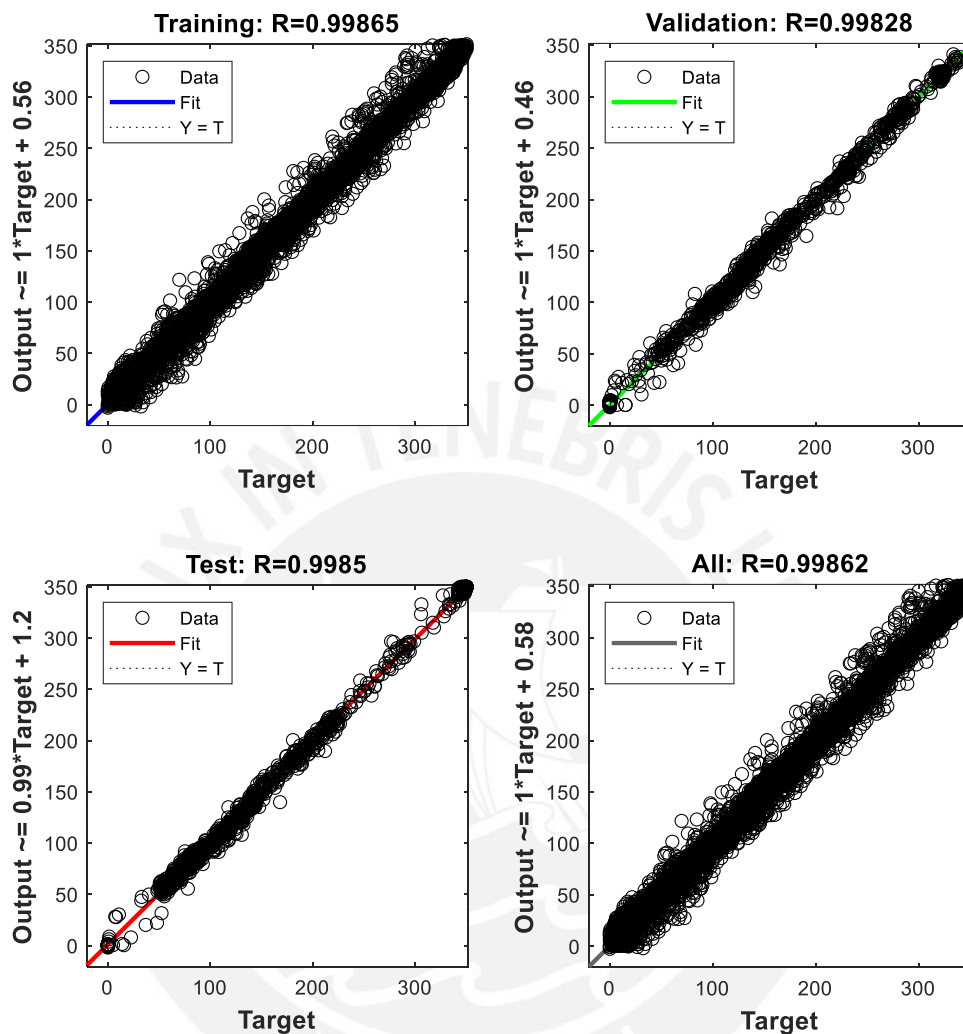


Fig. 2.9. Análisis de regresión de las salidas estimadas

En la Fig. 2.10 se observa una de las series de tiempo medidas y estimadas, coloreadas según el grupo a la que pertenece (entrenamiento, validación, prueba), observándose algunos errores dispersos.

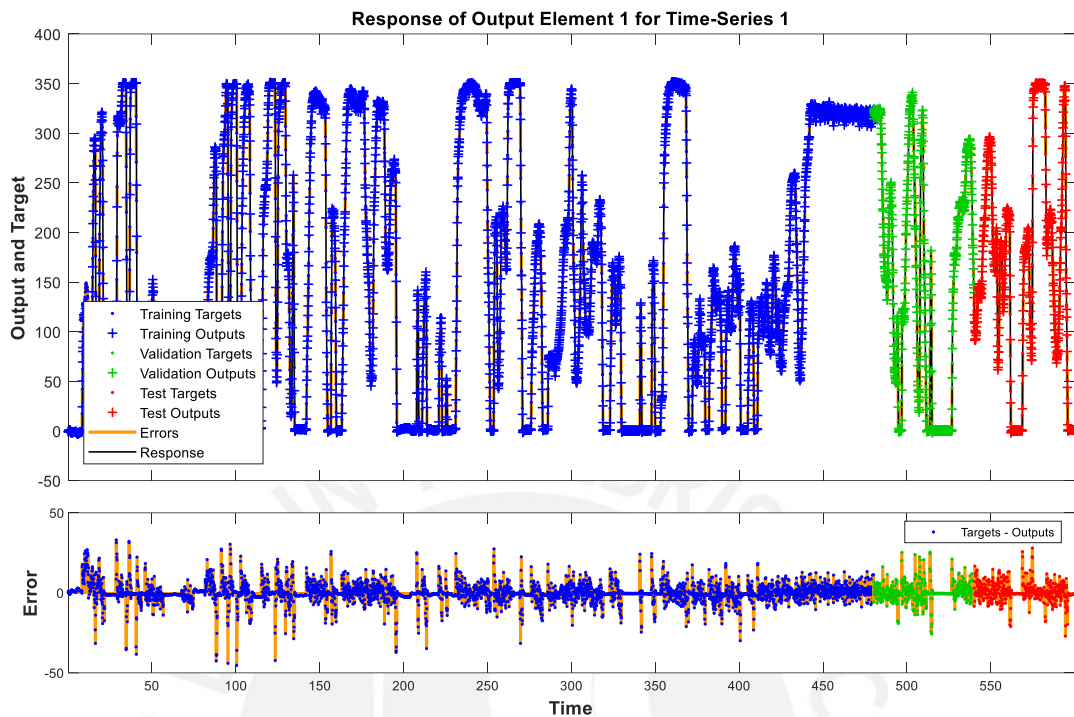
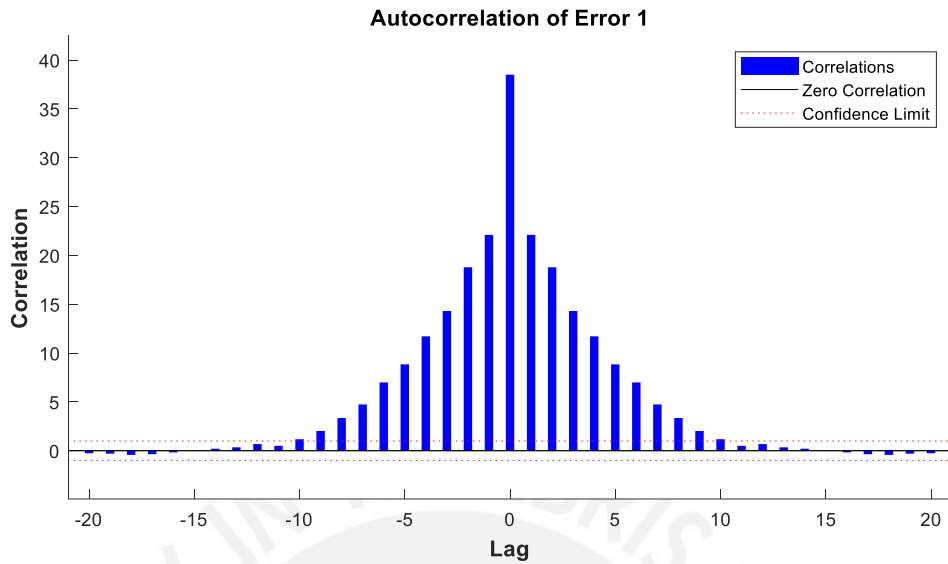
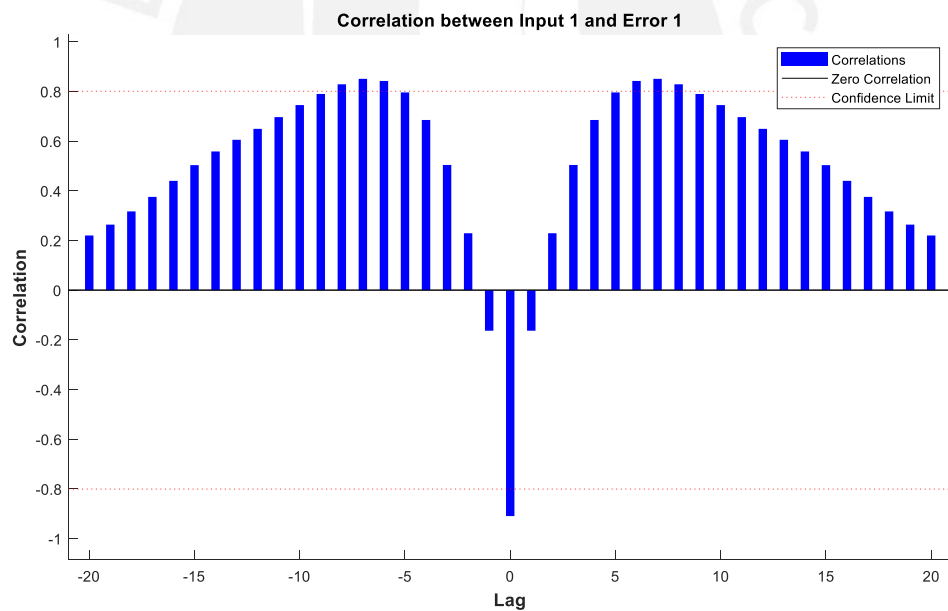


Fig. 2.10. Salida real y estimada en el tiempo

Estos resultados a lazo abierto podrían indicar un aparente buen desempeño, pero hay que recordar que en cada instante de tiempo se realimenta la salida real para favorecer un mejor entrenamiento, al momento de realizar el lazo cerrado el desempeño puede degradarse bastante, como se verá más adelante. Para conseguir un buen modelado es importante observar que los errores resultantes luego del modelado idealmente deberían ser producto del ruido en las mediciones y por lo tanto no debería tener correlación en el tiempo. La misma observación debe tenerse en cuenta en el caso de que exista correlación entre la señal de entrada y el error de la salida (Hagan et al., 2014). Para ello se muestra en la Fig. 2.11 la autocorrelación del error y en la Fig. 2.12 la correlación cruzada entre la entrada y el error a la salida.



*Fig. 2.11. Autocorrelación del error*



*Fig. 2.12. Correlación cruzada entre entrada y error*

Obsérvese que en ambos casos existe correlación para retardos distintos de cero lo que nos indica que el modelo es mejorable, ya que existe espacio para predecir ese comportamiento. Para ello en la etapa de mejora podemos manipular la cantidad de retardos que se alimentaran al sistema para generar

una mejor predicción y obtener un error no correlacionado. Ahora bien, si cerramos el lazo y reentrenamos la red para mejorar obtenemos un seguimiento muy malo como se observa en la Fig. 2.13

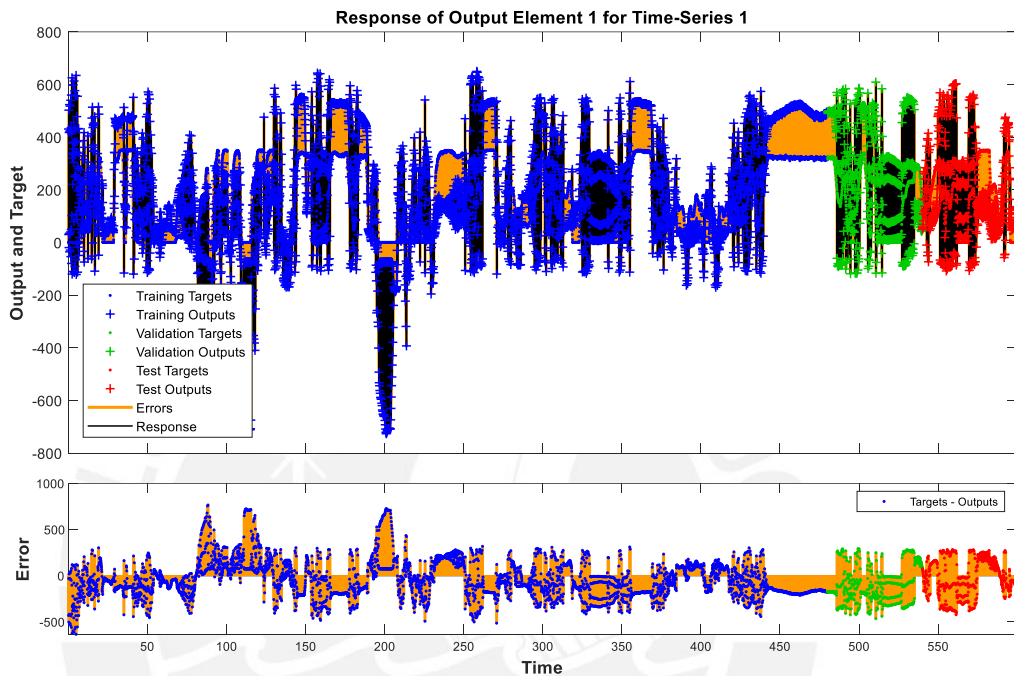


Fig. 2.13. Salida real y estimada en el tiempo en lazo cerrado

Como se observa, los pequeños errores en lazo abierto tienen una gran influencia a medida que se intenta predecir un horizonte de tiempo largo pues el error se propaga. Como se muestra en la Tabla 3 el MSE calculado se dispara, por lo que se requiere una mejora considerable.

Tabla 3. MSE para la prueba inicial

	Lazo abierto	Lazo cerrado
Entrenamiento	42.79	4.34e+4
Validación	32.49	4.61e+4
Prueba	26.08	4.05e+4
Total	40.76	4.34e+4

Para la mejora de estos indicadores será necesario aumentar los retardos para capturar la dinámica que se evidenció en la correlación del error, en base a esto podemos hacer pruebas con retardos entre 10 a 20 segundos. Luego de capturar este espacio de mejora se podría mejorar la generalización escogiendo un mayor o menor número de neuronas. En el caso del movimiento exagerado de la respuesta estimada de la Fig. 2.13 es posible que se deba realizar el proceso de regularización para evitar que los pesos de la red neuronal crezcan demasiado, en primera instancia los datos se normalizan para evitar esto pero puede buscarse penalizar pesos grandes a través de la elección del algoritmo, en ese sentido la opción a seguir sería escoger el algoritmo de retro propagación con regularización Bayesiana. En la Tabla 4 puede observarse los resultados más destacables durante la búsqueda de parámetros.

Tabla 4. Pruebas de la red neuronal

Parámetros	Comentario del modelo
Algoritmo: Levenberg-Marquardt Neuronas de la capa oculta: 10 Función de activación: sigmoidea Retardos a la entrada: 1:10 Retardos a la salida: 1	Autocorrelación error: considerable Correlación cruzada entrada – error: buena MSE en lazo cerrado: 8.34e3 El error es demasiado
Algoritmo: Levenberg-Marquardt Neuronas de la capa oculta: 10 Función de activación: sigmoidea Retardos a la entrada: 1:15 Retardos a la salida: 1:2	Autocorrelación error: aceptable Correlación cruzada entrada – error: buena MSE en lazo cerrado: 6.35e3 Las predicciones varían demasiado, posible sobre generalización
Algoritmo: Levenberg-Marquardt Neuronas de la capa oculta: 2 Función de activación: sigmoidea	Autocorrelación error: aceptable Correlación cruzada entrada – error: buena

Retardos a la entrada: 1:15 Retardos a la salida: 1:2	MSE en lazo cerrado: 749.32 Se disminuyó el número de neuronas para evitar la sobre generalización. Mejora sustancial, pero hay zonas que se desvía mucho del objetivo
Algoritmo: Levenberg-Marquardt Neuronas de la capa oculta: 2 Función de activación: sigmoidea Retardos a la entrada: 1:18 Retardos a la salida: 1:2	Autocorrelación error: aceptable Correlación cruzada entrada – error: buena MSE en lazo cerrado: 378.03 Se intentó aumentar neuronas, pero sobre generalizaba.
Algoritmo: Regularización Bayesiana Neuronas de la capa oculta: 2 Función de activación: sigmoidea Retardos a la entrada: 1:15 Retardos a la salida: 1:10	Autocorrelación error: buena Correlación cruzada entrada – error: buena MSE en lazo cerrado: 152.09 MSE para la data de prueba: 57.29 Para mejorar cambiamos el algoritmo. Nos quedamos con este modelo

Presentamos los resultados de las pruebas empezando con la configuración de la red neuronal en la Fig. 2.14 y el resultado de la evolución de la función de costo en la Fig. 2.15; donde se observa la convergencia de la función de costo luego de 294 épocas. Tanto la data de validación como la de prueba tienen aproximadamente el mismo MSE, la data de entrenamiento tiene un MSE más alto que podemos explicar si tenemos en cuenta que al inicio la data empieza con un torque al freno de cero; además, ya que la data original se tomó de una sola vez desde una carga mayor a menor, el transitorio del cambio de carga coincidiría con las etapas de validación y pruebas, lo que explicaría su mejor desempeño. Una ventaja de las redes neuronales es que se puede amoldar la información marginal que es incongruente.

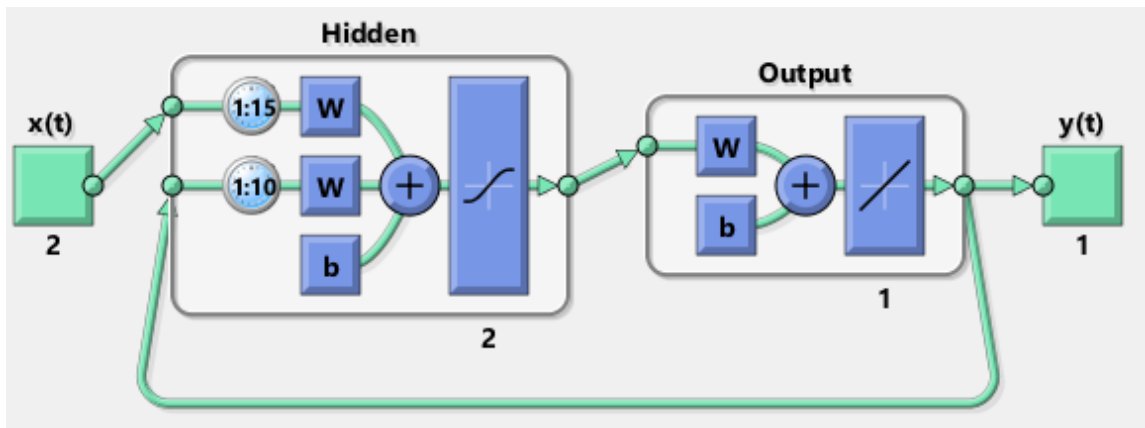


Fig. 2.14. Configuración final de la red neuronal NARX

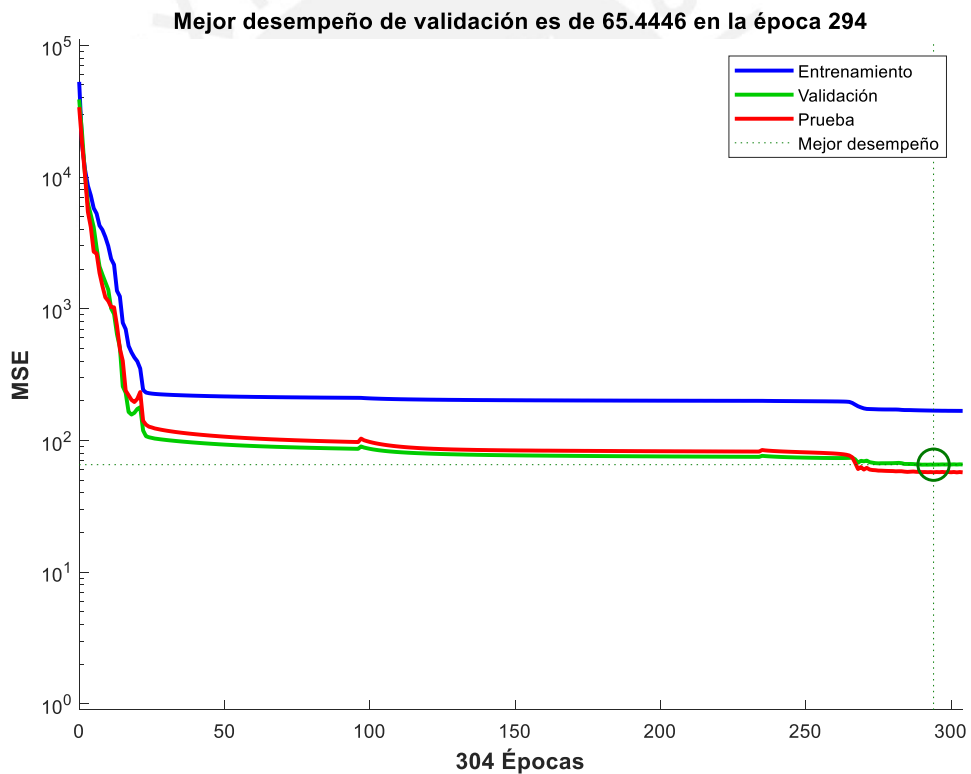


Fig. 2.15. Evolución del índice de desempeño

En cuanto a la correlación cruzada entrada – error (Fig. 2.16) y la autocorrelación del error (Fig. 2.17) las consideramos buenas.

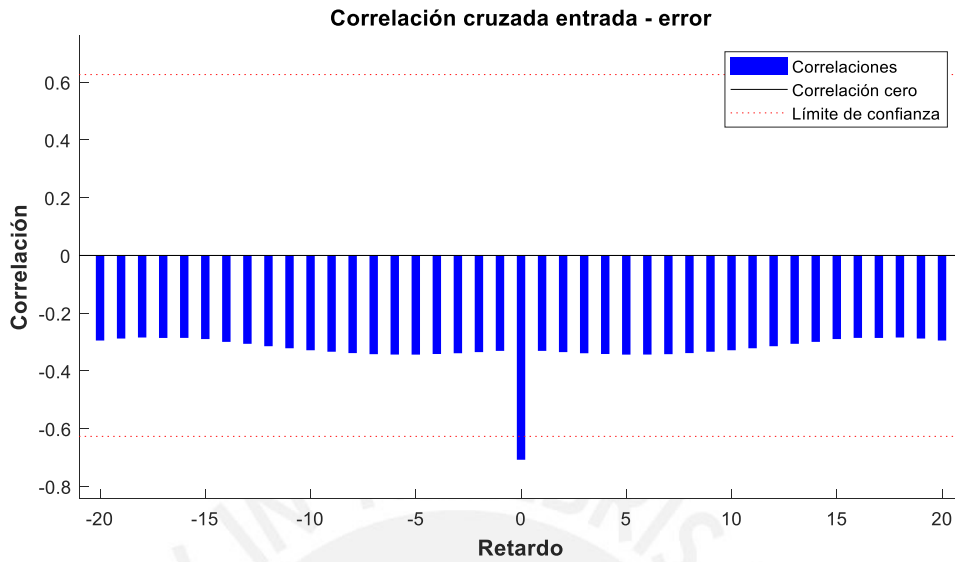


Fig. 2.16. Correlación cruzada entrada - error

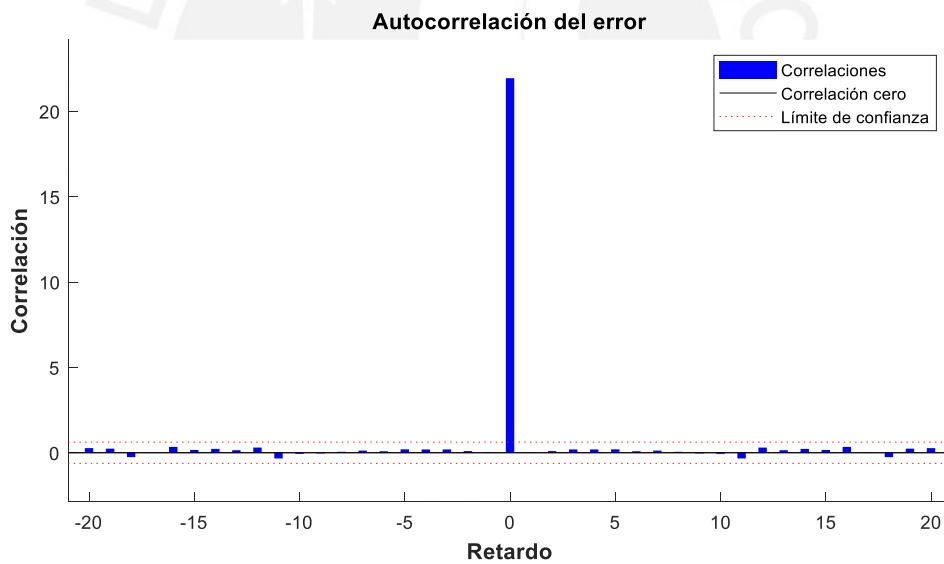


Fig. 2.17. Autocorrelación del error

La respuesta en el tiempo que se observa en la Fig. 2.18 y la comparación de las salidas reales y estimadas se muestra en la Fig. 2.19. En estas figuras es importante recalcar que se tiene el análisis de regresión con la red neuronal en lazo cerrado, es decir solo se recibe las entradas (flujo combustible, RPM); las salidas estimadas se realimentan a la red.

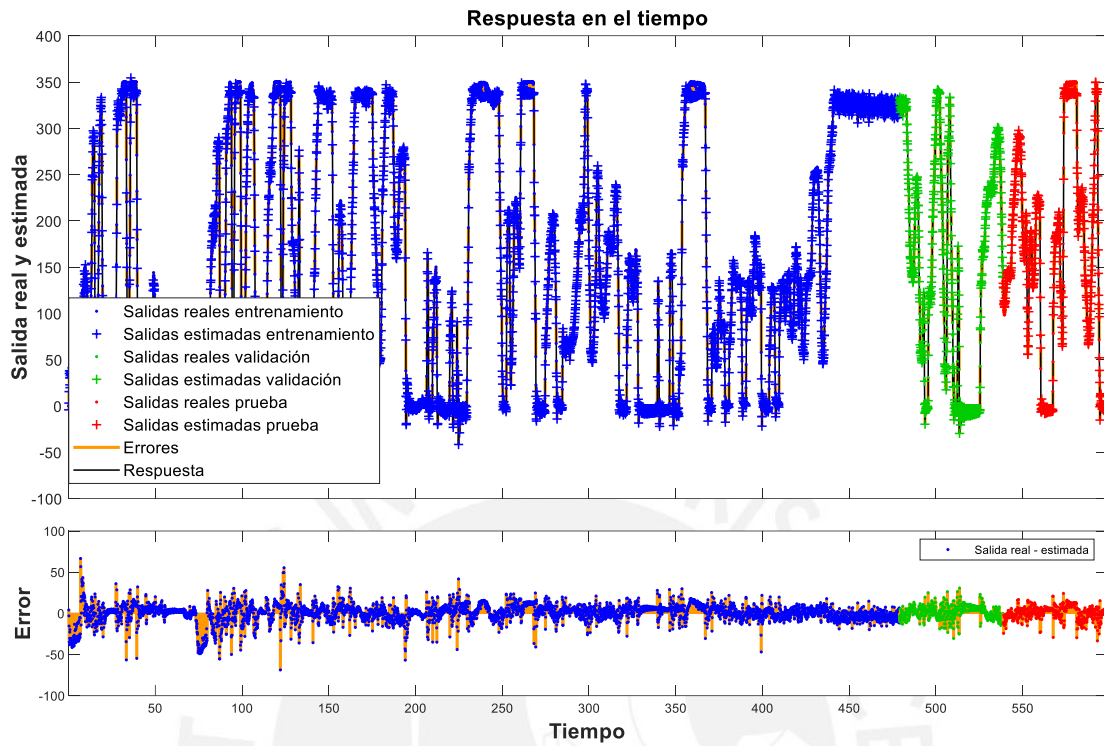


Fig. 2.18. Respuesta en el tiempo lazo cerrado

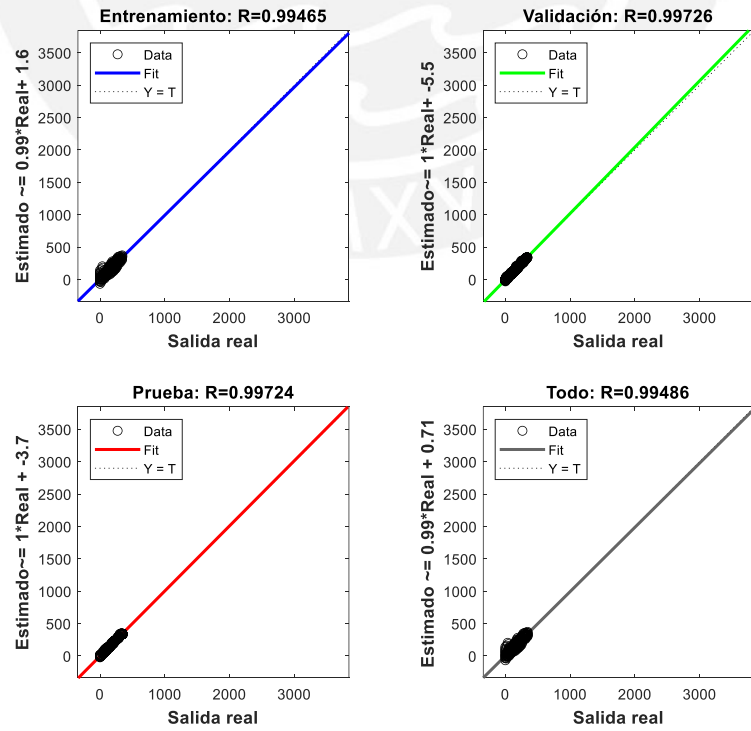


Fig. 2.19. Regresión Estimación - Real

Es conveniente para la simulación y diseño del controlador construir la red neuronal en Simulink, para ello se armó el esquema de la Fig. 2.20 con el detalle de las capas ocultas y de salida en la Fig. 2.21 y Fig. 2.22 respectivamente.

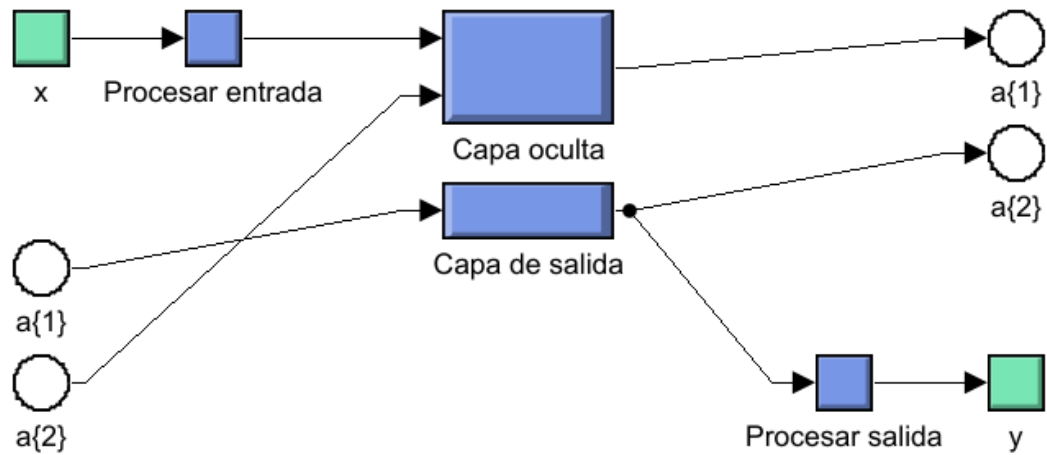


Fig. 2.20. Esquema de la red neuronal en Simulink

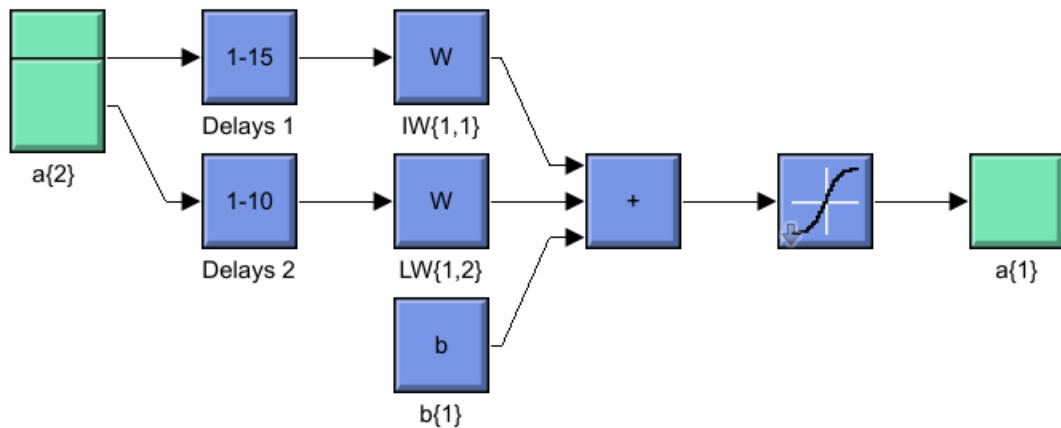


Fig. 2.21. Detalle de la capa oculta con función de activación sigmoidea

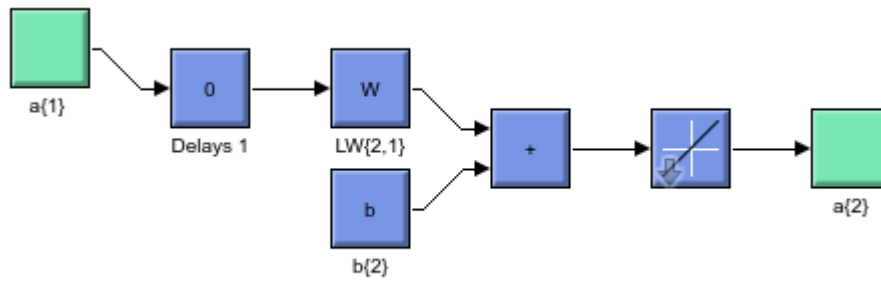


Fig. 2.22. Capa de salida con función de activación lineal

Para validar el modelo en Simulink introducimos los datos de flujo de combustible y RPM para los 3 tipos de condiciones y comparamos con los datos de torque obteniéndose el resultado de la Fig. 2.23 donde se puede observar un buen seguimiento del torque real.

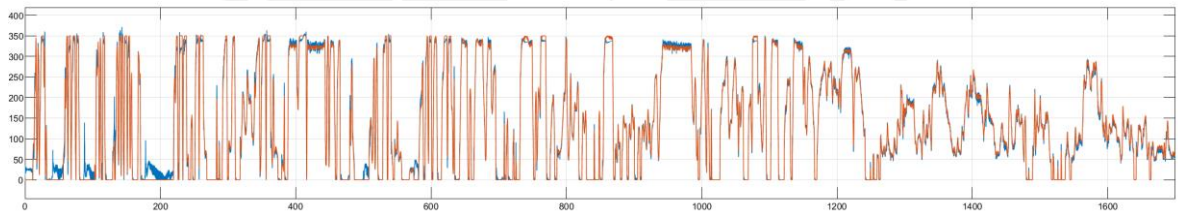


Fig. 2.23. Torque real (azul) y torque estimado (naranja)

Finalmente colocamos el modelo con redes neuronales del motor y añadimos el torque de carga y el cálculo de las RPM como se muestra en la Fig. 2.24. Estos componentes formarán el subsistema llamado “motor RPM”.

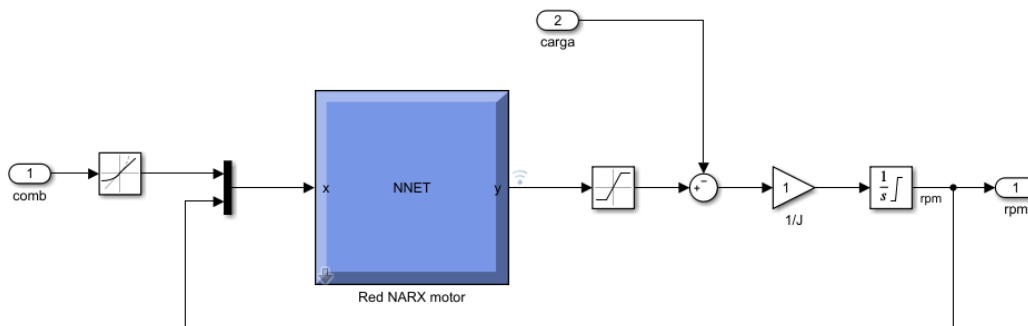


Fig. 2.24. Subsistema del motor con el cálculo de RPM

## 2.6 Conclusiones

- Con el estudio del estado del arte se pudo hallar las variables más relevantes para el control del motor.
- Se determinó que el torque al freno depende de las RPM y del flujo de combustible.
- Las RPM del motor dependen del torque de carga y puede ser hallada por la 2da ley de Newton.
- La identificación por redes neuronales es una buena opción para modelar motores y en este caso se pudo realizar el modelo con una red neuronal NARX obteniéndose un desempeño aceptable con un MSE de 57.29 y un  $R^2 = 0.99$  para la data de prueba.
- La simulación en Simulink estima correctamente el torque al freno real.

## **CAPÍTULO 3. DESARROLLO DEL SISTEMA DE CONTROL**

### **3.1 Introducción**

En este capítulo se desarrollará la estrategia de control de velocidad del motor diésel. Como este trabajo está enfocado al control de motores marinos se tomará el modelo teórico de la demanda del propulsor para generar carga. Como se explicó, los controladores de velocidad más utilizados son los PI, en el caso de los motores marinos en la Marina de Guerra del Perú, estos tienen gobernadores mecánicos y la idea es actualizar sus controladores. En este trabajo se diseñará un controlador estándar PID, pero también se diseñará un controlador PID con ganancias programadas en base al modelo matemático desarrollado, esto se sustenta en que el modelo del motor es un sistema no lineal que puede tener varios puntos de operación. Luego de esto se realizarán las comparaciones y posibles mejoras de eficiencia en el control de la velocidad del motor.

### **3.2 Diseño de un controlador PI estándar**

Para el diseño del controlador PI se utilizará la herramienta PID tuner de Matlab, armando el sistema de control como se muestra en la Fig. 3.1. Como se observa el torque de carga es calculada con la formula vista para la demanda de un propulsor marino.

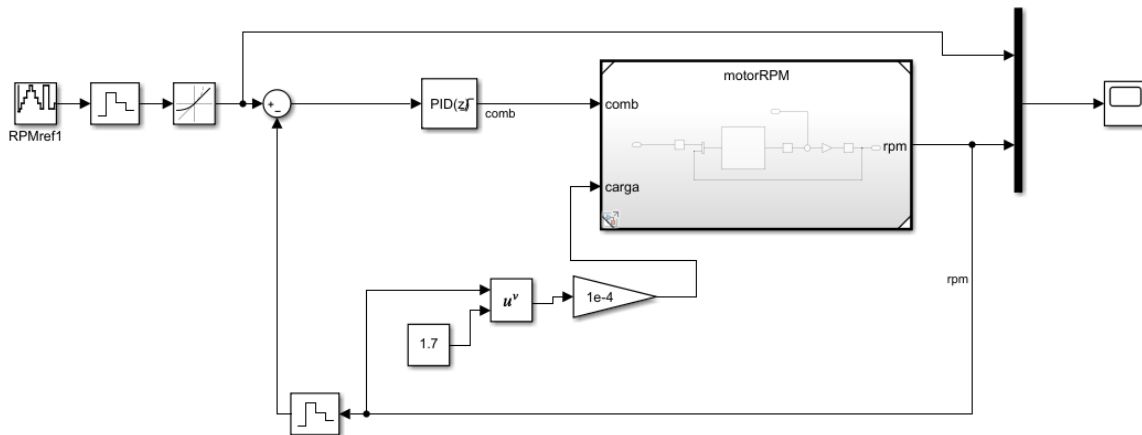


Fig. 3.1. Diagrama de bloques del sistema de control

La herramienta PID tuner realiza una linealización del sistema para definir una respuesta determinada en función de la robustez y la rapidez. Para el control de velocidad de motores se busca robustez pues una sobrevelocidad puede generar el embalamiento del motor, adicionalmente se busca reacciones rápidas en el caso de buques de guerra. Tomando en cuenta esto se buscará la mayor robustez y la mayor velocidad tratando de generar el menor sobreimpulso posible.

Para las pruebas se introdujo una señal de entrada en el sumador que ubique las RPM en distintos puntos de operación tal como se muestra en la Fig. 3.2. Ya que el sistema es no lineal se espera distintos comportamientos para la señal en los puntos de operación 1000, 1500, 2000, 2500, 3000 RPM; tanto subiendo como bajando. Además, se introduce una referencia que cambia rápidamente de 1000 a 3000 RPM en menos de 4 segundos en subida y bajada para probar la robustez del sistema.

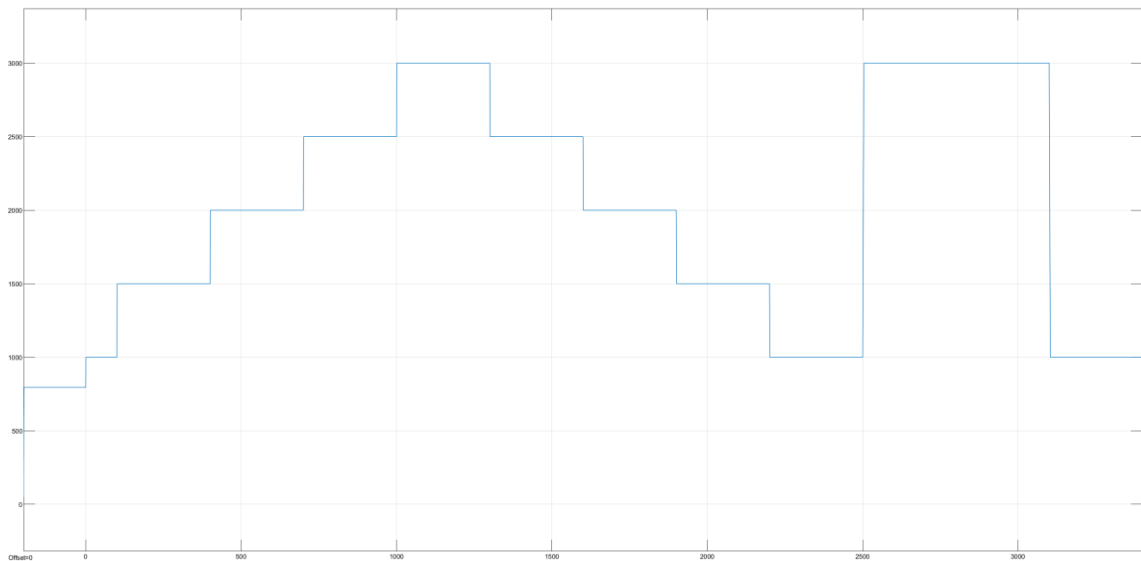
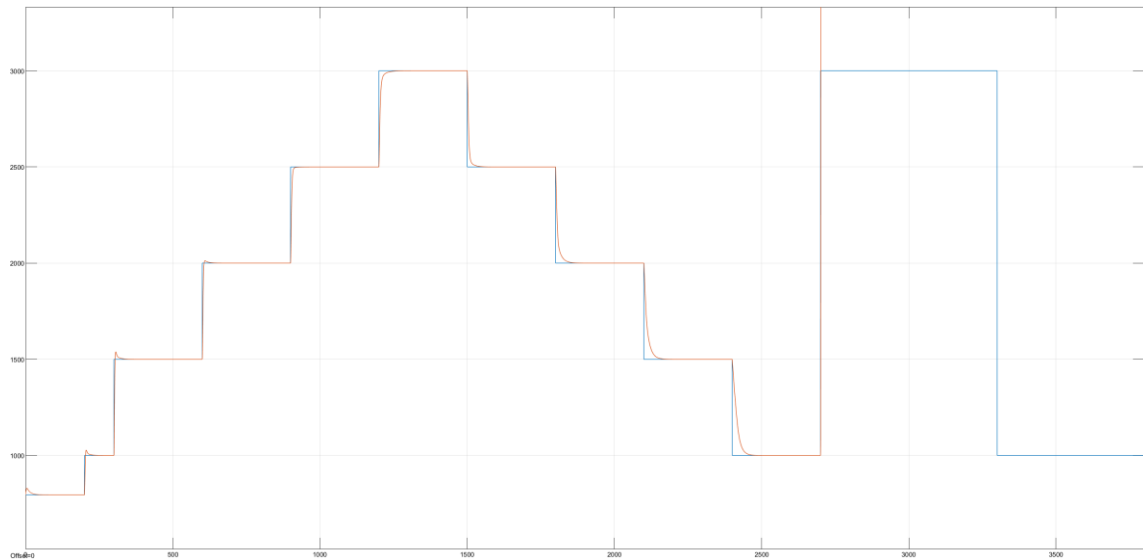


Fig. 3.2. Referencia de RPM

Si buscamos rapidez en torno a un tiempo de establecimiento de aproximadamente 20 segundos, tendremos la respuesta de la Fig. 3.3 usando los parámetros de la Tabla 5. Para cambios no muy grandes de la referencia el controlador sigue correctamente con ligeros sobreimpulsos a bajas RPM; sin embargo, un cambio grande de referencia produce el embalamiento del motor por lo que en este caso el controlador requeriría realizar el cambio de referencia con una menor pendiente.

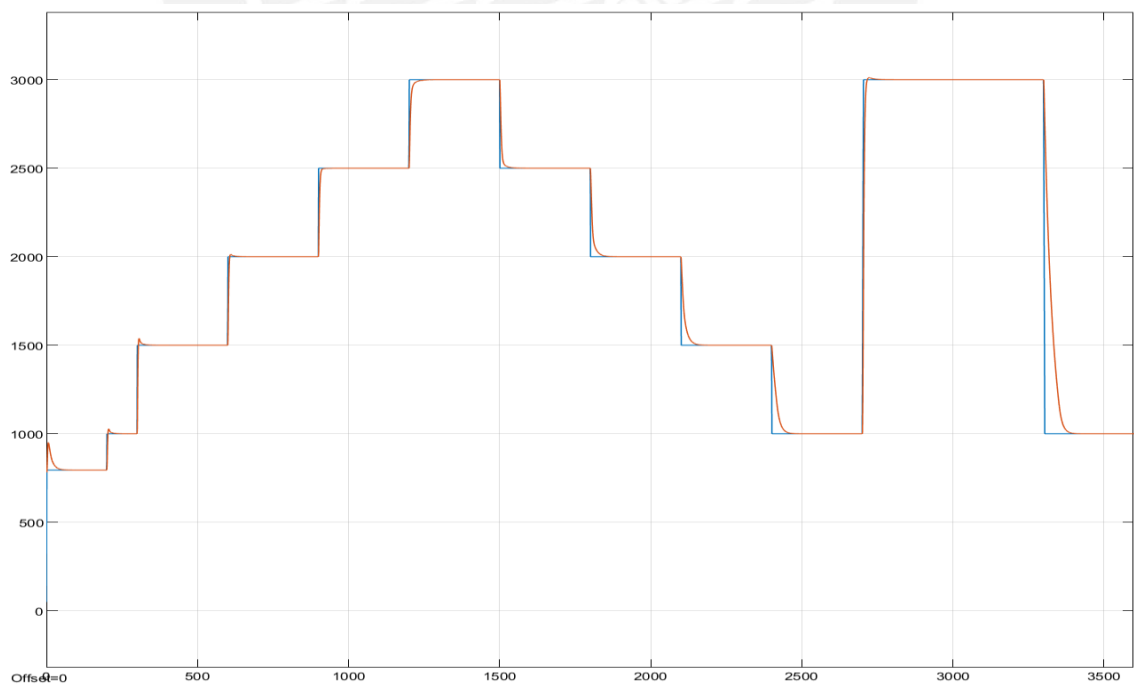
Tabla 5. Parámetros PID estándar

Parámetros	Magnitud
$K_p$	0.00964572406365397
$K_I$	0.000767335996910868
$K_D$	-0.00874122459921082
N	1.04183277486834



*Fig. 3.3. Respuesta rápida, pero falla cuando hay un cambio de 2000 RPM en 2 segundos*

Para poder comparar el desempeño de este controlador con el de ganancias programadas, que sí fue capaz de seguir este cambio de referencia, usaremos una pendiente máxima de 2000 rpm en 4 segundos obteniéndose la Fig. 3.4.



*Fig. 3.4. Respuesta del sistema con un cambio de 2000 RPM en 4 segundos*

Para los primeros 2600 segundos, donde se analizará el cambio entre los distintos puntos de operación, se midió el error entre la referencia y la salida real; y también el consumo de combustible obteniéndose:

Consumo de combustible: 1.302 Kg

MSE: 7.15e+6

Para los primeros 3600 segundos, donde se tiene el cambio grande de referencia, los índices de desempeño son los siguientes:

Consumo de combustible: 2.158 Kg

MSE: 5.327e+7

### 3.3 Diseño de un controlador PI con ganancias programadas

Para esta estrategia es necesario establecer puntos de operación, linealizar la planta en estos puntos y sintonizar los parámetros del PID para cada punto de operación. La transición de los parámetros se puede hacer con una tabla de búsqueda donde si se tienen condiciones intermedias se pueden aplicar interpolación lineal, pero para hacer la transición más suave ajustaremos un polinomio de 2do orden al cambio de ganancias.

Tomaremos como puntos de operación 1000, 1500, 2000, 2500 y 3000 RPM pero calcularemos también para 3500 y 4000 RPM para tener espacio para manejar sobreimpulsos. En cada punto se debe buscar el consumo de combustible en equilibrio, para ello ejecutamos el siguiente script de (MathWorks, 2020a) adaptado al motor diésel:

```
mdl_LA = 'motorRPMcarga'; % Se elaboró un modelo del motor incluyendo la carga
open_system(mdl_LA)

mdl_LC = "NN3";
```

```

open_system mdl_LC

rpm = (1000:500:4000)'; % puntos de operación
for k=1:length(rpm)
    opspec = operspec mdl_LA; % Especificación de punto de operación
                                % de Simulink
    opspec.Inputs(1).u = 1; % Entrada tentativa para empezar la iteración
    opspec.Outputs(1).y = rpm(k); % Establecemos los puntos de operación
    opspec.Outputs(1).Known = true; % Los puntos de operación son conocidos

    % Encontramos los puntos de operación con la función findop
    [op(k),report(k)] = findop mdl_LA,opspec,findopOptions('DisplayReport','off');

end

% guardamos la información
for k=1:length(rpm)
    RPM(k) = report(k).Outputs.y;
    COMB(k) = op(k).Inputs.u;
End

```

Los resultados se muestran en la Tabla 6. Estos puntos fueron validados en Simulink donde en todos los casos se observa una respuesta amortiguada con un tiempo de establecimiento entre 40 a 200 segundos.

Tabla 6. Puntos de operación previo a la linealización

Puntos de operación	
Flujo de combustible [Kg/h]	Velocidad angular [RPM]
0.3243	1000
0.7409	1500
1.5366	2000
2.8436	2500
4.8755	3000
7.9601	3500
12.5269	4000

Con esta data se puede construir la curva estática entre el flujo de combustible y las RPM como se ve en la Fig. 3.5. Como se observa la cantidad de puntos asemeja la curva real y por lo tanto podemos considerar lineal el comportamiento en la vecindad de los puntos de operación.

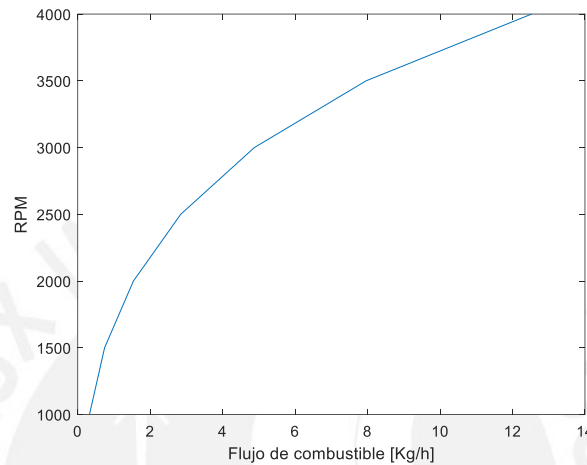


Fig. 3.5 Curva estática Flujo de combustible - RPM

En Simulink se armó un subsistema “Controlador” (Fig. 3.6) donde ubicamos las tablas de búsqueda que almacenarán los valores de  $K_p$ ,  $K_i$ ,  $K_d$  y  $N$  del PID según las RPM. La ubicación de este subsistema se observa en la Fig. 3.7.

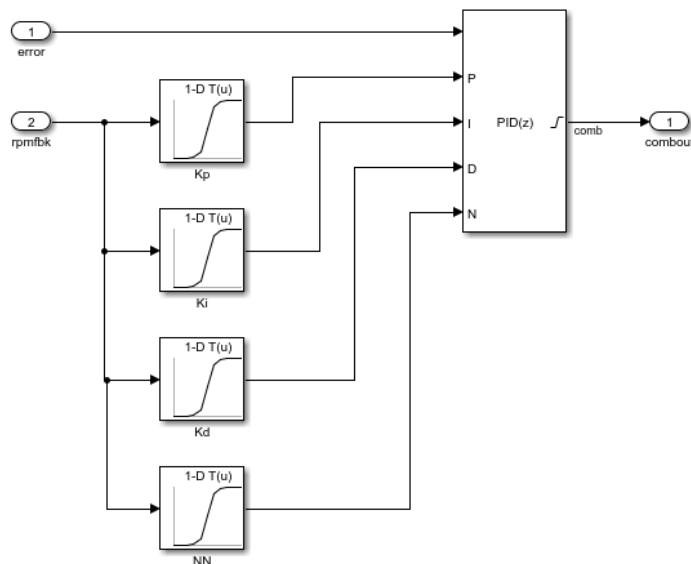


Fig. 3.6. PID con ganancia programada

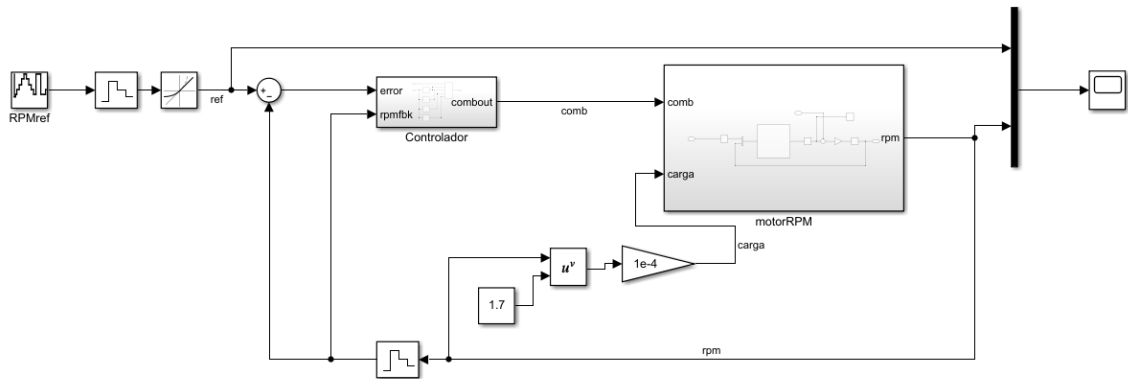


Fig. 3.7. Diagrama de bloques del sistema con PID con ganancia programada

Luego linealizamos, preparamos la superficie de sintonización y asignamos los parámetros a las tablas de búsqueda que se van a sintonizar ejecutando el siguiente script adaptado de (MathWorks, 2020a):

```
%% Linealización y exploración
G = linearize mdl_LA, 'motorRPMcarga/Red NARX motor', op); % Linealización
plot(COMB, RPM)
bode(G)

%% Preparando la superficie de sintonización

TuningGrid = struct('rpm', rpm); % Armamos una estructura que guarda las rpm
ShapeFcn = @(rpm) [rpm, rpm^2]; % Usaremos un polinomio de 2do orden
% para modelar el cambio de parámetros

% Establecemos las superficies de sintonización de los parámetros
Kp = tunableSurface('Kp', 0, TuningGrid, ShapeFcn);
Ki = tunableSurface('Ki', -2, TuningGrid, ShapeFcn);
Kt = tunableSurface('Kd', 0, TuningGrid, ShapeFcn);
NN = tunableSurface('NN', 0, TuningGrid, ShapeFcn);

% Cambiamos el modelo NARX del motor por el linealizado en los puntos de
% operación.
```

```

BlockSubs = struct('Name','NN3/Motor RPM/Red NARX motor','Value',G);
ST0 = sITuner('NN3',{ 'Kp','Ki','Kd','NN'},BlockSubs);
ST0.Ts = 0.1; % Periodo de muestreo para la sintonización

% Registramos puntos de interés para la sintonización
ST0.addPoint({'rpm','comb','RPMref'})

% Especificamos los parámetros sintonizables en el modelo
ST0.setBlockParam('Kp',Kp);
ST0.setBlockParam('Ki',Ki);
ST0.setBlockParam('Kd',Kt);
ST0.setBlockParam('NN',NN);

```

La linealización proporciona una familia de sistemas en espacio estado. Si realizamos un diagrama de Bode de los sistemas obtendremos la Fig. 3.8, la cual muestra la entrada combustible en la primera columna, la entrada RPM en la segunda columna versus la salida. Como se ve existe un gran cambio del diagrama de fase con el cambio de las RPM, lo cual indica un comportamiento no lineal, esto concuerda con los transitorios distintos vistos para cada punto de operación cuando se hizo el diseño del PID estándar. Otra observación es un pico de ganancia en el torque debió a las RPM a una frecuencia de 20 rad/s, sin embargo, a la salida el torque al freno con el torque de carga se integra lo cual producirá un filtrado de este pico.

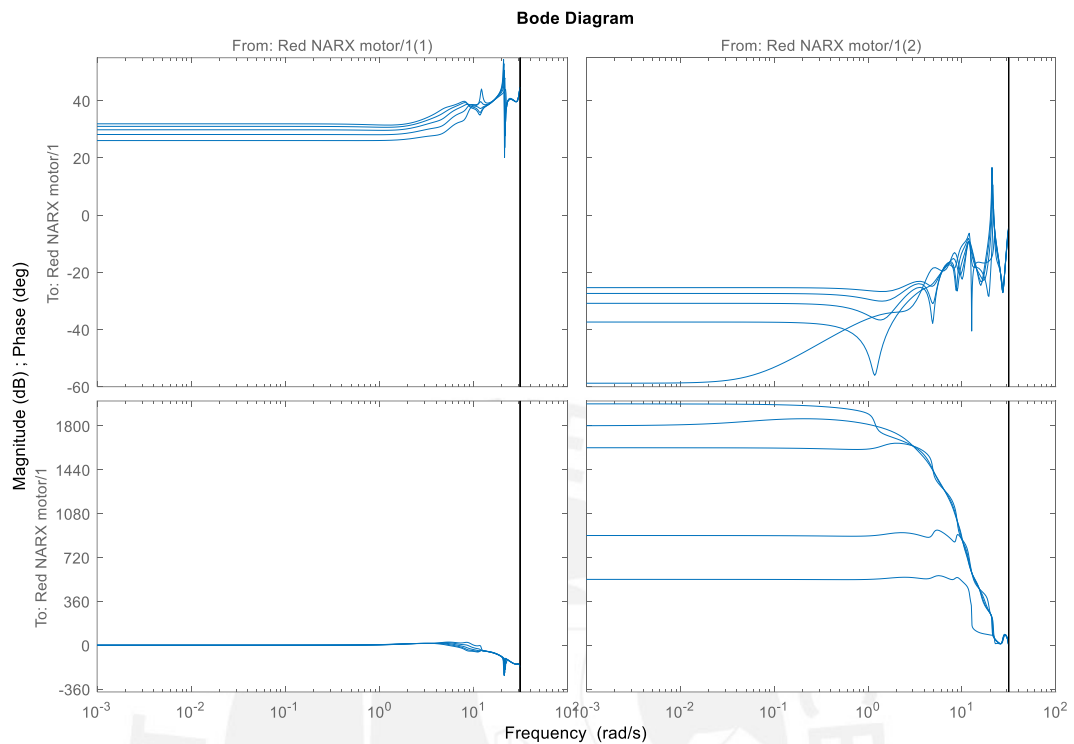


Fig. 3.8. Diagrama de Bode de la familia de sistemas de espacio estado

Con el apoyo de Matlab podemos automatizar la búsqueda de la sintonización, indicándole al software los objetivos de la sintonización, en este caso podemos resumir los objetivos deseados como:

- Seguimiento de la referencia con tiempo de respuesta corto y máximo 10% de sobreimpulso.
- La ganancia de la referencia al flujo inyectado de combustible no debe superar el límite máximo de inyección.
- La ganancia de la referencia a las RPM del motor no debe generar que el motor se embale.
- Sistema robusto con margen de ganancia de 7db y margen de fase de 45°.

Para introducir estos objetivos, ejecutamos el siguiente script donde se establecen estos requerimientos y se realiza la sintonización con comandos de Control System Toolbox de Matlab:

```
%% Objetivos de la sintonización

% Tiempo de respuesta deseado: 4s, Error máximo: 0%
R1 = TuningGoal.Tracking("RPMref","rpm",4,0);
viewGoal(R1,ST)

% Mínimo Mg: 7dB / Mínimo Mf: 45°
R2 = TuningGoal.Margins("rpm",7,45);
viewGoal(R2,ST)

% Ganancia máxima entre la referencia y el flujo de combustible: 10/3000
perfilGanancia = frd([0.02 0.005 0.0025 0.0025],[0 1 10 100]);
R3 = TuningGoal.Gain("RPMref","comb",perfilGanancia);
viewGoal(R4,ST)

% Sobreimpulso máximo: 10%
R4 = TuningGoal.Overshoot("RPMref","rpm",10);

% Buscamos atenuar la ganancia de baja frecuencia
perfilGanancia2 = frd([0.04 0.0091 1. 1. 1.],[0 0.004 2 15 150]);
R5 = TuningGoal.Gain("RPMref","rpm",perfilGanancia2);
viewGoal(R5,ST)

% Sintonización (R1, R3 y R4 restricciones fuertes)
ST = systune(ST0,[R2 R5 ],[R1 R3 R4]);
writeBlockValue(ST) % actualizamos los parámetros del modelo
```

Algunos comentarios acerca de la elección de los parámetros de los objetivos se resumen a continuación:

- Para mantener controlado la inyección de combustible y que el motor no se embale inicialmente se trabajó con una ganancia fija para todo el espectro, sin embargo, si bien los resultados fueron buenos el sistema no controlaba mejor que el PID estándar. Un análisis de la inyección de combustible en el tiempo (Fig. 3.9) mostró que se estaba limitando el combustible para los cambios cortos de referencia por lo que se trabajó entonces con un perfil de ganancias. El perfil se definió observando el diagrama de Bode desde la referencia hasta la inyección de combustible (Fig. 3.10). Para frecuencias altas cuando existe grandes cambios de referencias la atenuación debe ser mayor que para frecuencias bajas, por lo tanto, el perfil sigue ese comportamiento; para las magnitudes del perfil de ganancias se consideró que en las RPM más altas (3000 RPM) no se generen más de 10 Kg/h de combustible y por simple división se estableció la ganancia en altas frecuencias, de la misma forma se trabajó con las ganancias de baja frecuencia y a partir de ahí se hicieron algunos ajustes.
- Durante las pruebas se logró un buen desempeño al subir las RPM hasta los puntos de operación, pero para en el caso de bajar las RPM los tiempos de establecimientos era muy largos. Para evitar se elaboró un perfil de ganancias deseado para atenuar la ganancia en bajas frecuencias desde la referencia hasta la RPM según se observa en Fig. 3.11.

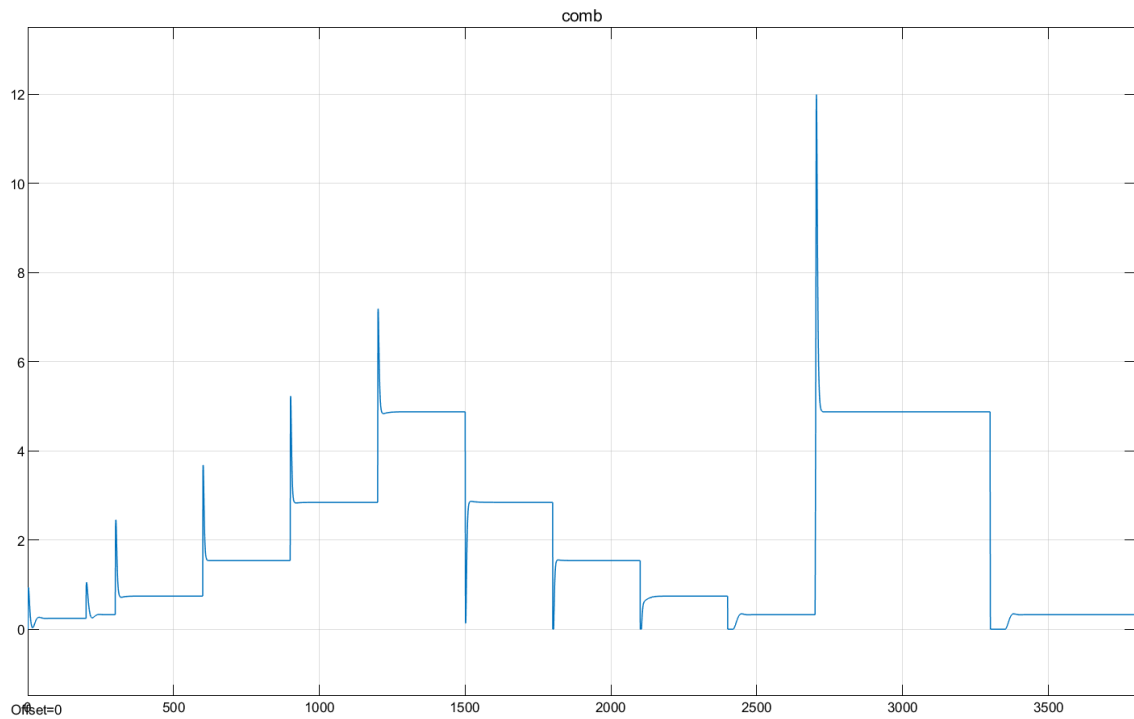


Fig. 3.9. Flujo de combustible en el tiempo

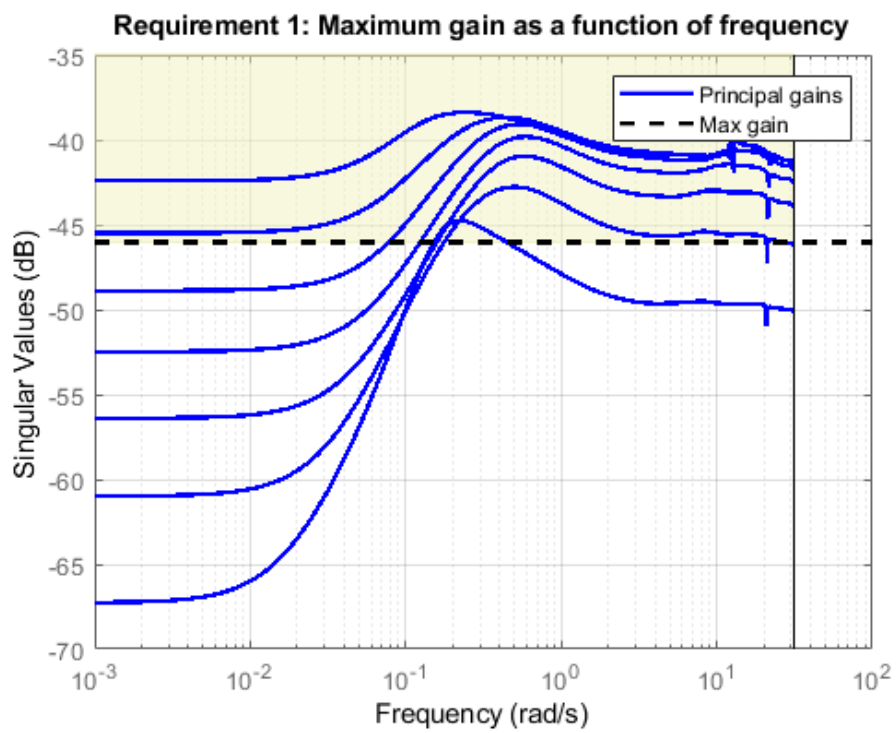


Fig. 3.10. Ganancias máximas desde la referencia hasta el flujo de combustible

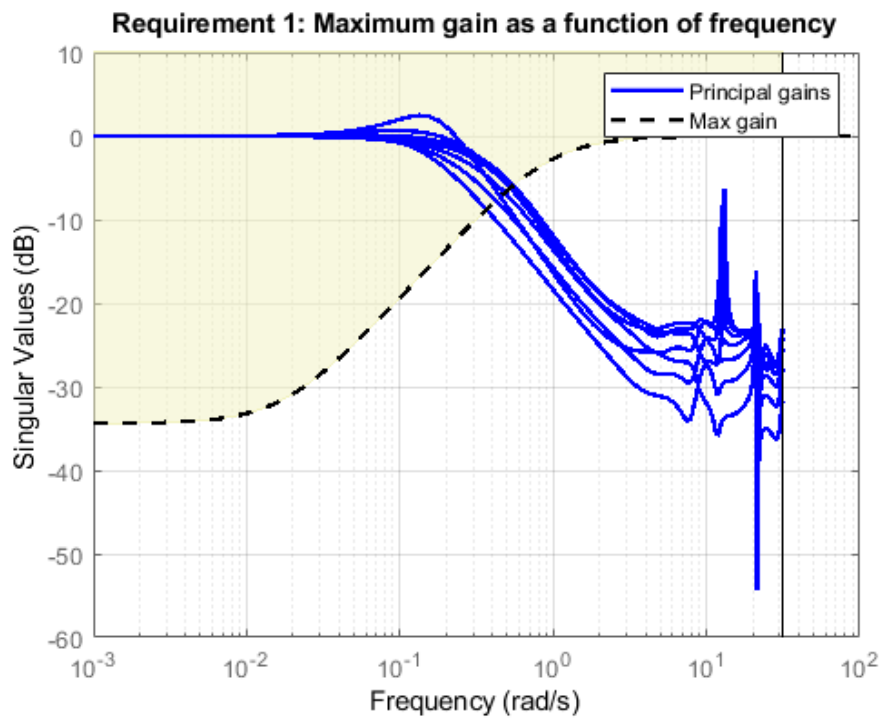


Fig. 3.11. Ganancias máximas desde la referencia hasta las RPM del motor

La sintonización se realiza usando las restricciones suaves R2 y R5 y las restricciones fuertes R1, R3, R4. Los resultados de la sintonización de Matlab se representan con un puntaje, siendo los puntajes menores a 1 cuando los objetivos se satisfacen. En este caso para las restricciones suaves el puntaje fue de 52.4 y para las fuertes fue de 2.4. Esto se logró en 159 iteraciones. Las restricciones fuertes no fueron completamente satisfechas, pero se logró una cercanía aceptable. Para las restricciones suaves el puntaje es alto, en particular nos interesa que el sistema tenga un buen desempeño y que se cumpla la restricción R2 (márgenes de ganancia y fase deseados). En la Fig. 3.12 se observa que los márgenes se logran satisfactoriamente por lo que consideramos estos resultados buenos. Se debe notar que esta figura se logró con el comando *viewgoal* del Control System Toolbox de Matlab y está basada en un análisis de estabilidad usando márgenes de disco, que es un método que considera cambios de ganancia y fase simultáneos, además de incertidumbre (Seiler et al., 2020). Este método representa una mejor herramienta comparada con el análisis clásico de márgenes, donde se varía solo uno a la vez.

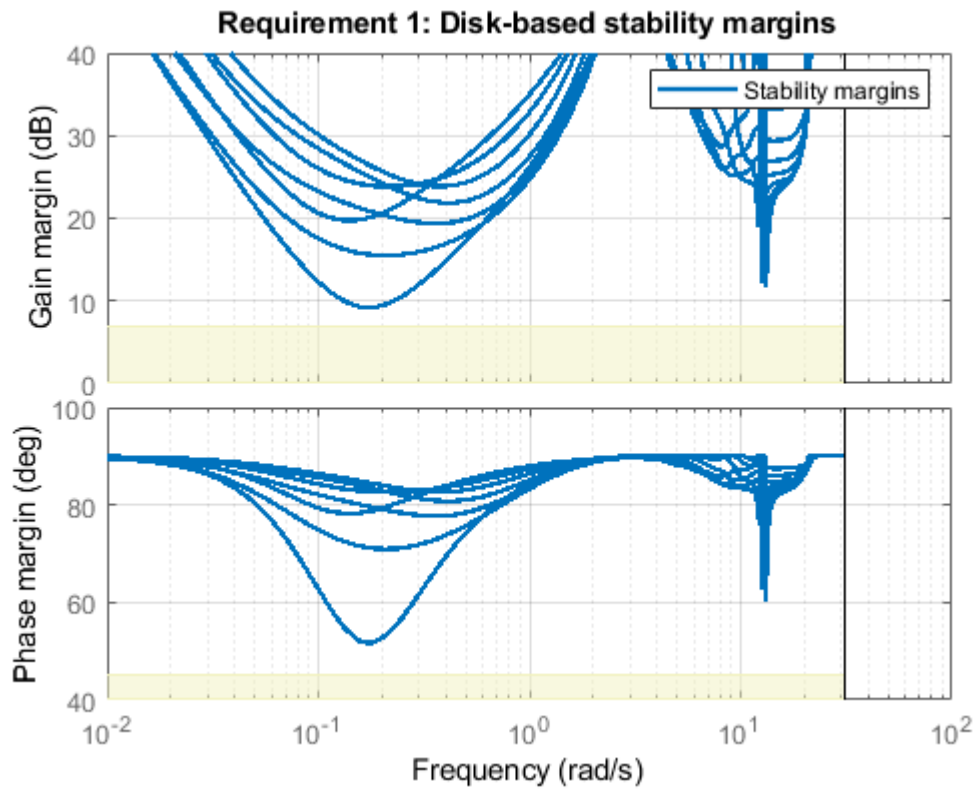


Fig. 3.12. Los márgenes de ganancia y fase se encuentran por encima de lo mínimo deseado (zona amarilla)

Luego de la sintonización actualizamos los valores de las tablas de búsqueda. Los resultados de esto se muestran en las curvas de sintonización de la Fig. 3.13.

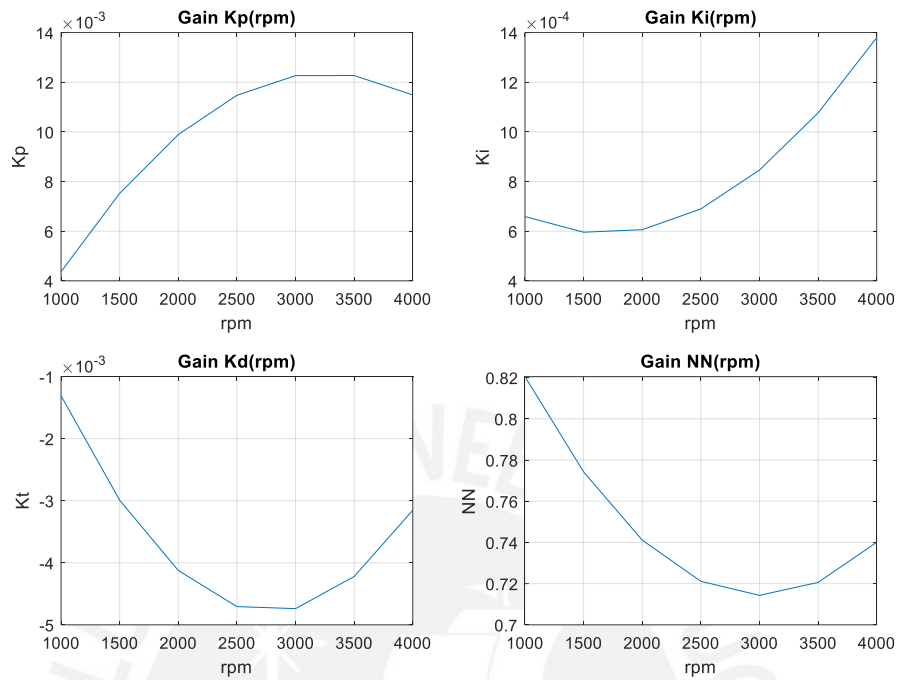


Fig. 3.13. Curvas de sintonización de los parámetros  $K_p$ ,  $K_i$ ,  $K_d$ ,  $NN$

Finalmente realizamos la simulación en Simulink obteniéndose la respuesta de la Fig. 3.14.

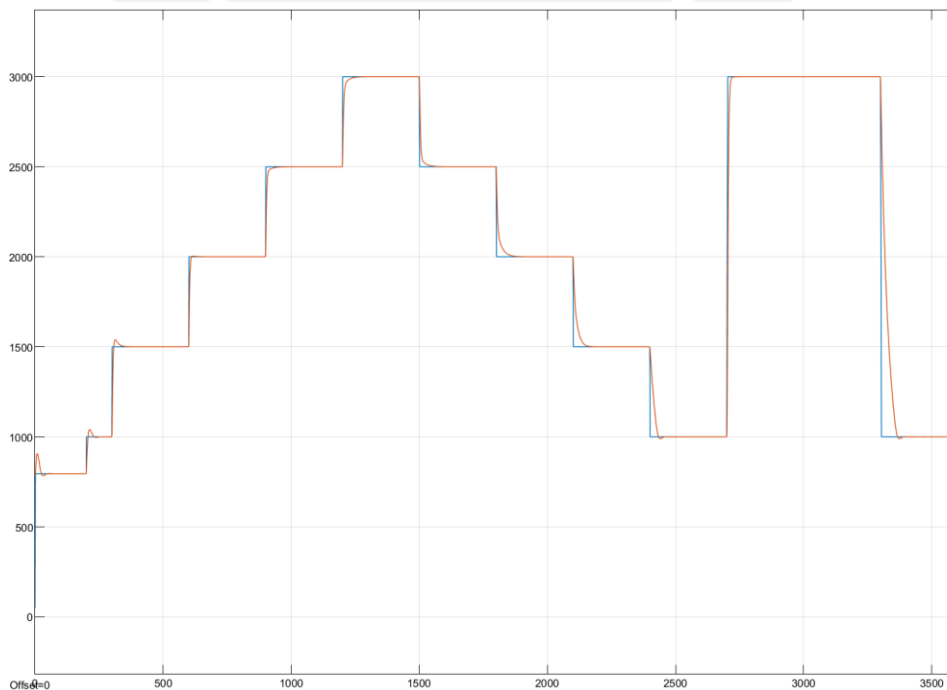


Fig. 3.14. Respuesta con PID con ganancia programada

Analizamos el desempeño primero observando los primeros 2600 segundos:

Consumo de combustible: 1.302 Kg

MSE: 6.841+6

Para los primeros 3600 segundos, donde se tiene el cambio grande de referencia, los índices de desempeño son los siguientes:

Consumo de combustible: 2.156 Kg

MSE: 5.178e+7

En ambos casos los resultados muestran una mejora de hasta 4.3% en el seguimiento y 0.09% en el consumo de combustible en favor del PID con ganancias programadas.

### 3.4 Análisis de estabilidad de ambos controladores

Es conveniente analizar la estabilidad para medir la robustez del sistema; para ello podemos linealizar el sistema en lazo abierto y medir los márgenes para determinar la estabilidad en lazo cerrado. Ejecutamos el siguiente script con comandos de Control System Toolbox:

```
%% Análisis estabilidad
```

```
% Sistema con PID estándar
```

```
ioInput = linio("NN2/Controlador",1,'looptransfer');
```

```
ioOutput = linio("NN2/motorRPM",1,'looptransfer');
```

```
Li = linearize("NN2",ioInput,COMB);
```

```
Lo = linearize("NN2",ioOutput,COMB);
```

```
Si = allmargin(-Li)
```

```
So = allmargin(-Lo)
```

```
% Sistema con PID con ganancia programada
```

```
ioInput2 = linio("NN3/Controlador",1,'looptransfer');
```

```
ioOutput2 = linio("NN3/motorRPM",1,'looptransfer');
```

```
Li2 = linearize("NN3",ioInput2,COMB);
Lo2 = linearize("NN3",ioOutput2,COMB);
```

```
Si2 = allmargin(-Li2)
So2 = allmargin(-Lo2)
```

En el script se tomaron 2 puntos para medir la estabilidad:

- Abriendo el lazo desde la entrada de combustible para medir la estabilidad del controlador dentro del sistema (Li y Li2)
- Abriendo el lazo desde la salida de RPM para medir la estabilidad ante perturbaciones (Lo y Lo2)

Los resultados para la estabilidad del controlador se muestran en la

Tabla 7. Se observa que en todos los casos de márgenes de ganancia y márgenes de retardo (máximo retardo de tiempo que puede soportar el sistema) del PID con ganancia son superiores a los del PID estándar. En el caso de los márgenes de fase el PID con ganancia programada es superior en la mayoría de los casos. Estos resultados aseguran la estabilidad antes perturbaciones y ruido.

*Tabla 7. Estabilidad desde la entrada de combustible*

	Punto de operación	Márgenes de ganancia mínimo[db]	Márgenes de fase [°]	Márgenes de retardo [s]
PID estándar	1000	32.02	53.96	30.06
	1500	30.95	52.63	28.43
	2000	32.53	62.19	30.96
	2500	23.30	61.71	19.77
	3000	26.28	59.55	23.31
	3500	32.83	57.01	32.38
	4000	29.91	53.22	26.80
PID con	1000	33.57	60.90	47.90
	1500	33.61	60.47	43.39

	2000	33.89	61.07	47.75
	2500	33.76	59.44	42.00
	3000	33.78	60.48	45.93
	3500	33.70	63.10	47.52
	4000	33.48	59.14	42.21

En el caso de la estabilidad desde la salida de RPM, la cual mide la estabilidad antes perturbaciones de carga, se obtuvo la Tabla 8. En este caso se presenta un caso similar que el caso anterior: el PID con ganancia programada es superior al PID estándar en márgenes de ganancia y márgenes de retardo. Los márgenes de fase son parecidos.

Tabla 8. Estabilidad desde la salida de RPM

	Punto de operación	Márgenes de ganancia mínimo [db]	Márgenes de fase [°]	Márgenes de retardo [s]
PID estándar	1000	-33.85	59.14	28.09
	1500	-35.97	58.76	26.64
	2000	-32.84	59.29	28.89
	2500	23.36	57.97	18.79
	3000	26.64	58.72	21.89
	3500	33.45	60.97	29.85
	4000	-29.79	57.36	25.04
PID con ganancia programada	1000	-37.46	53.08	44.56
	1500	-24.36	51.77	40.57
	2000	33.44	61.04	44.23
	2500	33.53	60.49	39.06
	3000	33.38	58.38	42.46
	3500	33.26	55.92	43.85
	4000	-23.57	52.25	39.27

Es importante recalcar que en todos los casos tenemos sistemas estables, si bien existen márgenes de ganancia negativos, esto no necesariamente quiere decir que los sistemas son inestables pues pueden tener comportamiento de fase

no mínima. Esto también se puede corroborar al analizar el diagrama de Nyquist o en la misma información que ofrece Matlab en el diagrama de Bode (Fig. 3.15).

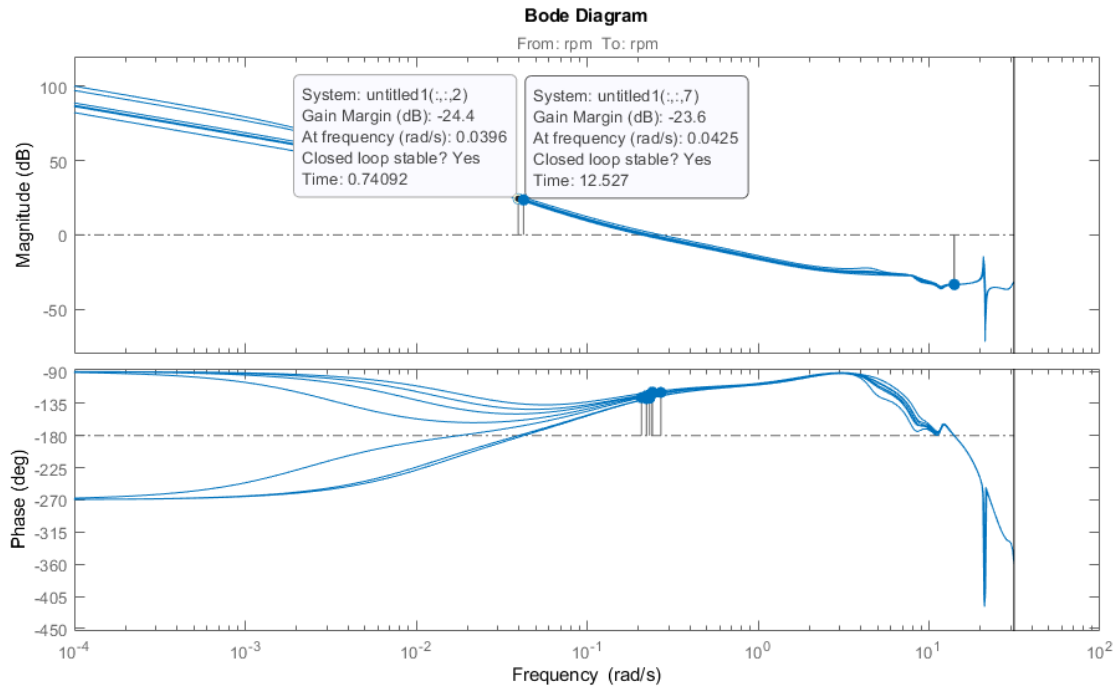


Fig. 3.15. Diagrama de Bode y márgenes desde la salida de RPM

Finalmente se resumen los resultados en la Tabla 9.

Tabla 9. Resumen resultados

Índice de desempeño	PID estándar	PID con ganancia programada	Diferencia
Consumo de combustible (2600s)	1.302 Kg	1.302 Kg	0%
MSE (2600s)	7.15e6	6.841e6	-4.32%
Consumo de combustible (3600s)	2.158 Kg	2.156 Kg	-0.09%
MSE (3600s)	5.327e7	5.178e7	-2.79%

<b>Promedio márgenes de ganancia</b>	29.69 dB	33.68 dB	+13.43%
<b>Promedio márgenes de fase</b>	57.18°	60.66°	+6.08%
<b>Promedio márgenes de retardo</b>	27.39s	45.24s	+65.17%

### 3.5 Conclusiones

- Se realizó el diseño de un sistema controlado por PID obteniéndose un MSE para los primeros 3600s de  $5.327e7$  con un consumo de 2.158 Kg.
- Se realizó el diseño de un sistema controlado por PID con ganancia programada con un MSE para los primeros 3600s de  $5.178e7$  con un consumo de 2.156 Kg, lo cual proporciona una mejora de 2.79% para el seguimiento y de 0.09% para el combustible.
- Se analizó la estabilidad de ambos sistemas encontrándose mejoras significativas en márgenes de ganancia (+13.43%) y márgenes de retardo (+65.17%)
- El PID con ganancia programada demuestra ser una mejor estrategia de control que un PID estándar debido al mejor seguimiento medido por el error cuadrático medio, menor consumo de combustible y mucho mayor robustez demostrado por sus márgenes de ganancia y retardo.

## **CAPÍTULO 4. PROPUESTA DE IMPLEMENTACIÓN DEL SISTEMA DE CONTROL**

### **4.1 Introducción**

Para la propuesta de implementación se necesitan seleccionar los instrumentos para medir y manipular las variables relevantes de sistema, por ello se requiere:

- Sensor de RPM
- Inyector de combustible
- Potenciómetro para la referencia
- Sistema embebido (microcontrolador)

### **4.2 Selección de instrumentos**

#### Sensor de RPM

Para medir la velocidad angular instantánea se puede usar el sensor de posición angular del cigüeñal (Ribbens, 2017). Generalmente se usan sensores magnéticos u ópticos, pero es preferible un sensor magnético porque no le afecta el aceite o polvo u otros contaminantes. Sin embargo, una posible desventaja de un sensor magnético es la ausencia de señal de salida cuando el motor no se mueve. Un tipo de sensor que no tiene este problema es el sensor de posición basado en el efecto Hall (Fig. 4.1). Estos sensores envían una onda cuadrada con una amplitud de 5V, lo cual es apto para el uso con un microcontrolador.



*Fig. 4.1. Sensor de posición por efecto Hall (Bosch)*

#### Inyector de combustible

Un inyector de combustible es realmente una válvula solenoide. La válvula permite o bloque el flujo de combustible al motor (Ribbens, 2017). Para motores diésel estos inyectores pueden soportar presiones entre 1600 a 2500 bar. Se necesita un inyector por cada cilindro del motor. El manejo del flujo de combustible del inyector se puede hacer con un driver conectado al microcontrolador. La Fig. 4.2 muestra un inyector con válvula solenoide para motor diésel.



*Fig. 4.2. Inyector con válvula solenoide (Bosch)*

## Potenciómetro

La referencia para el motor se puede dar a través de un potenciómetro. En el caso de automóviles el potenciómetro va unido al pedal del auto. En el caso de un buque, la referencia se da a través de una palanca (Fig. 4.3). La señal enviada puede ser adaptada para que este en el rango de entrada de un microcontrolador.



Fig. 4.3. Palanca para la referencia de RPM (Norisys 4 LT4)

## Sistema embebido

Para el sistema embebido, hasta donde se investigó los fabricantes de motores no suelen brindar mucha información de sus unidades controladoras de motor (ECU). Sin embargo, existen soluciones de código libre para sistemas de gestión de motores que incluyen planos electrónicos y código fuente. Entre las soluciones investigadas son destacables las siguientes:

- SECU-3: ECU desarrollada con un microcontrolador Atmel ATmega644 de 8bit, 20MHz, 4KB RAM. Tiene la capacidad de controlar inyectores, estimar el flujo de aire, ralentí en lazo abierto, entre otros pero la documentación está en ruso (SECU-3, 2020).
- rusEFI: ECU desarrollada con microcontroladores STM32f4 y STM32f7 con 160MHz y 200KB de RAM. El proyecto tiene varias presentaciones como la placa microRusEfi que es capaz de controlar motores de 4 inyectores, 1 sensor hall y 10 entradas analógicas (rusEFI, 2020).

- Speeduino: ECU desarrollada con una plataforma Arduino Mega 2560. Este arduino tiene un microcontrolador Atmel ATmega2560 de 8bit, 16MHz, 8KB RAM. Tiene la capacidad para ingreso de mapas de 16x16, control de motores de hasta 8 cilindros, calibración de sensores, compatibilidad con decoders, control de ralentí en lazo abierto y cerrado, etc. (Speeduino, 2020).

Para este trabajo nos basaremos en la ECU Speeduino porque se consideró que tiene un mayor desarrollo respecto a las otras y la información es más accesible. En la Fig. 4.4 podemos observar la placa de la ECU (Fig. 4.5) y el microcontrolador (Fig. 4.6) del Speeduino junto con los sensores y actuadores permitidos. En este caso la ECU está enfocada a motores de encendido por chispa, pero se puede adaptar para el control del motor diésel del presente trabajo. Como se observa la placa puede admitir sensores Hall y hasta 4 inyectores. En el caso del potenciómetro puede conectarse a la entrada del TPS (Throttle position sensor por sus siglas en inglés).

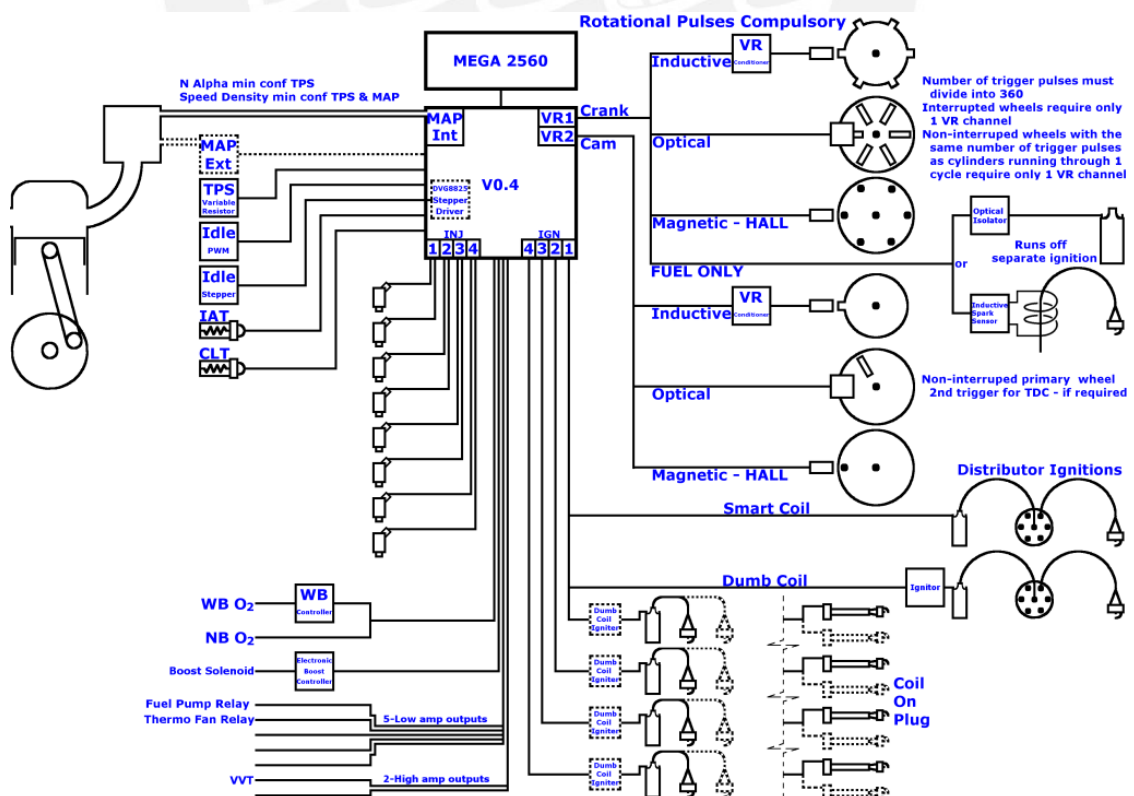
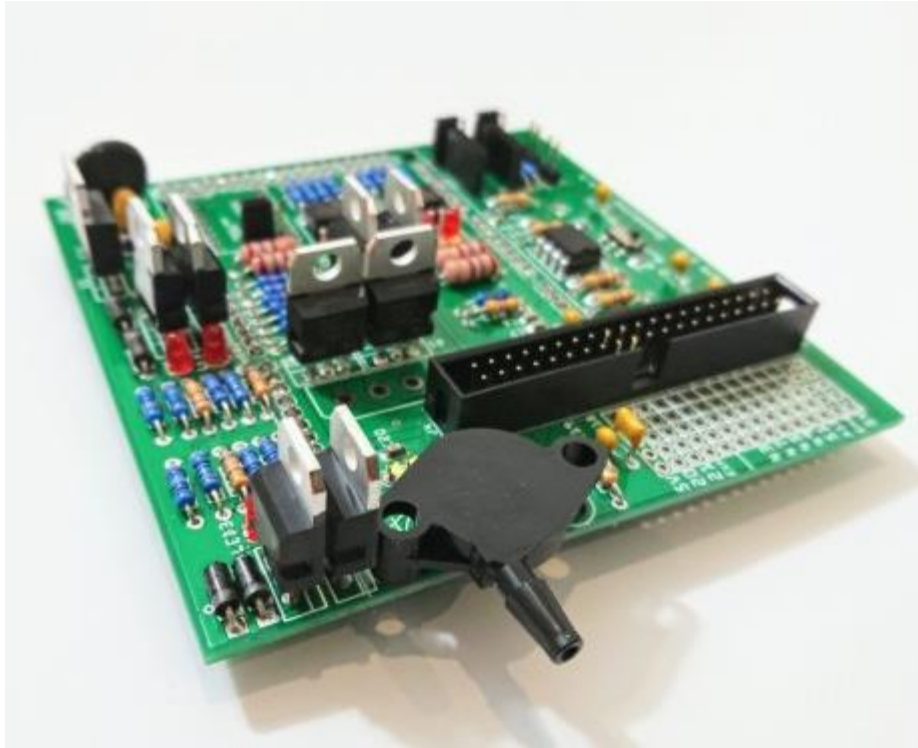
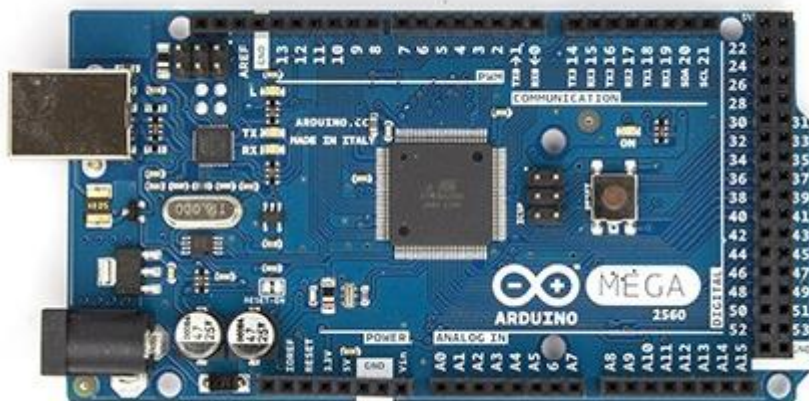


Fig. 4.4. Diagrama de las posibles conexiones del Speeduino



*Fig. 4.5. PCB de la ECU*



*Fig. 4.6. Plataforma Arduino con microcontrolador ATmega2560*

El código fuente debe ser compilado en una IDE como Arduino IDE o PlatformIO IDE y luego puede cargarse en el microcontrolador. Este código fuente debe ser modificado para realizar el algoritmo del PID con ganancias programadas.

## Generación de código fuente

El código fuente del controlador realizado con Simulink puede ser generado con la herramienta Simulink Coder (MathWorks, 2020b). Dado que los microcontroladores ATmega pueden ser programados en lenguaje C, realizaremos la siguiente configuración en la ventana de configuración de parámetros (Fig. 4.7):

- Fijamos un paso fijo de 0.1 segundos con un tiempo de parada infinito.
- En implementación de hardware escogemos como proveedor Atmel y tipo de dispositivo AVR (8-bit)

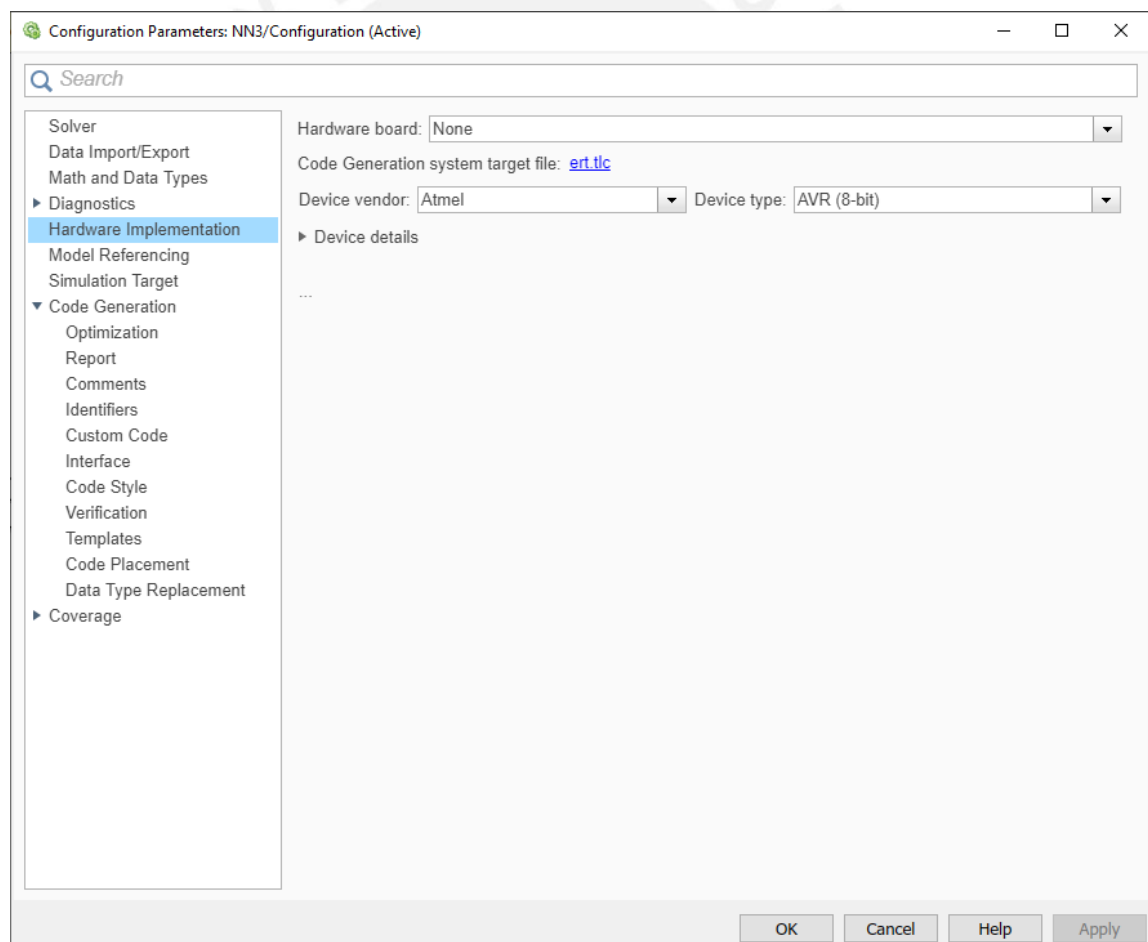


Fig. 4.7. Configuración para generar código C para microcontroladores ATmel

Luego le damos click derecho al subsistema “Controlador”, luego escogemos “C/C++ code” y luego “Embedded Coder Quick Start” y seguimos las opciones por defecto. Simulink Coder generará un archivo Controlador.c y sus cabeceras que podemos incluir como función en el código fuente del Speeduino.

Para las pruebas se pueden hacer la denominada prueba “Hardware in the loop” o HIL por sus siglas en inglés donde la computadora simulará la planta y la ECU real se conectará a esta a través de una tarjeta de adquisición de datos para realizar el control como se observa en la Fig. 4.8.



Fig. 4.8. Esquema para realizar pruebas HIL con la ECU

Finalmente se presenta en la Fig. 4.9 el esquema de la implementación del sistema de control.

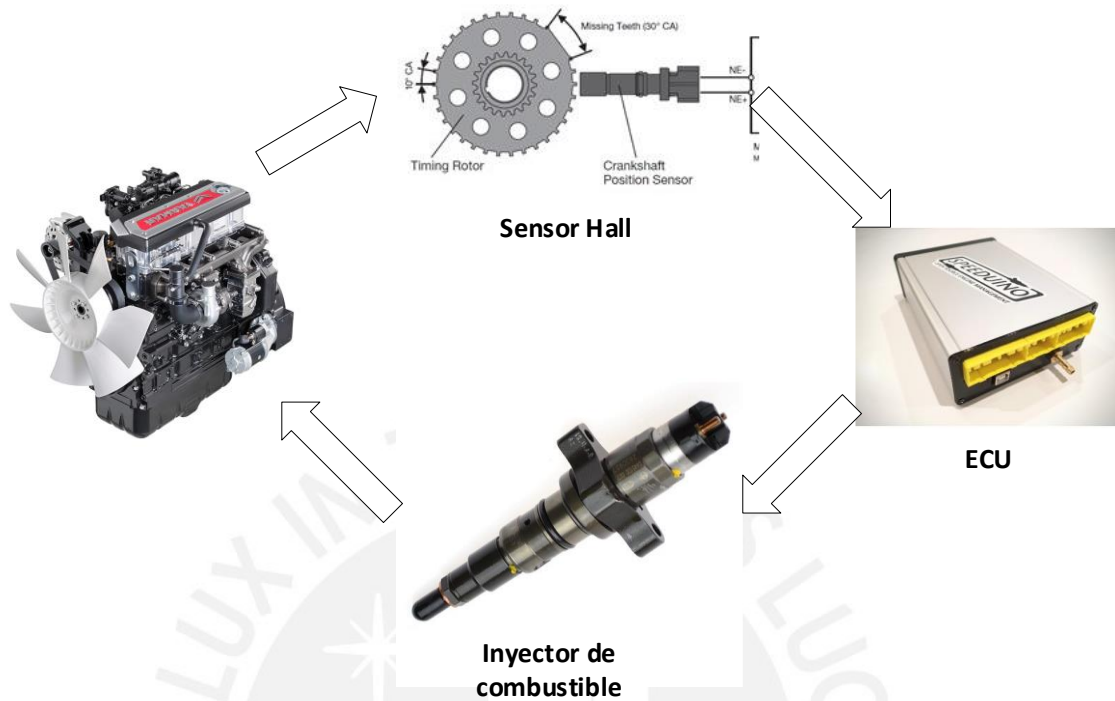


Fig. 4.9. Diagrama de la implementación propuesta para el sistema de control

### 4.3 Conclusiones

- Se realizó una propuesta de implementación del PID con ganancia programada sobre una plataforma Arduino con un microcontrolador ATmega2360 y una plataforma ECU de código libre llamada Speeduino.
- Como instrumentación se propuso un sensor de posición por efecto Hall para medir las RPM, un inyector con válvula solenoide para la inyección de combustible y un potenciómetro para introducir la referencia al sistema.
- El código fuente en lenguaje C fue generado por Simulink Coder para ser introducido al código fuente de libre acceso de la plataforma Speeduino.

## CONCLUSIONES GENERALES

- Se obtuvo el modelo de un motor diésel sometido a un ambiente marino utilizando redes neuronales y se desarrolló sistema de control de sus RPM empleando un PID de ganancias programadas, con lo cual se mejoró su desempeño respecto a un PID convencional.
- Se realizó un estudio sobre el comportamiento dinámico de los motores diésel encontrándose que las variables que influyen las RPM del motor son el torque de carga, el flujo de combustible y la misma rotación del motor.
- Se desarrolló un modelo matemático basado en redes neuronales no lineales autoregresivas con entrada exógena (NARX) con datos de libre acceso de un motor diésel. Los resultados arrojaron un  $R^2=0.99$  entre la data de prueba (información que fue usada para el entrenamiento) y la data real. La carga fue modelada con la ecuación de demanda de un propulsor marino.
- Dado el comportamiento no lineal del motor se diseñó un controlador PID con ganancias programadas y con fines de comparación un PID estándar. Los resultados favorecen el uso del controlador PID con ganancias programadas, el cual supera en todas las pruebas el desempeño al PID estándar, con hasta 4.32% mejor desempeño en el seguimiento de la referencia y un 0.09% de ahorro de combustible. Asimismo, se demostró que el controlador con ganancia programada es más robusto en los puntos de operación del motor cuando se analizaron los márgenes de ganancia, fase y de retardo.
- Se propuso la implementación del controlador PID con ganancias programadas usando un sensor de efecto Hall para medir las RPM, un

inyector de combustible con válvula solenoide para el flujo de combustible y una placa Speeduino de código abierto manejada con un microcontrolador ATmega2360.



## RECOMENDACIONES

- La condición inicial de los pesos de la red neuronal se toma de forma aleatoria y luego a través del proceso de aprendizaje converge. Esto puede generar distintos modelos para los mismos parámetros sintonizables por lo que se recomienda ejecutar desde cero más de un entrenamiento a la red con los mismos parámetros en la etapa de pruebas pues es posible tener un mejor desempeño con condiciones iniciales más favorables.
- Es importante luego del entrenamiento de la red neuronal en lazo abierto, cerrar el lazo y reentrenar la red. Esto genera una mejora adicional en el desempeño de la red.
- La elección de los parámetros para optimizar en el proceso de sintonización puede ser un proceso iterativo largo, se recomienda automatizar el proceso para que Matlab realice estas pruebas y guarde los valores de cada iteración para su posterior análisis.

## REFERENCIAS BIBLIOGRÁFICAS

- Alegret, G., Llamas, X., Vejlggaard-Laursen, M., & Eriksson, L. (2015). Modeling of a Large Marine Two-Stroke Diesel Engine with Cylinder Bypass Valve and EGR System. *IFAC-PapersOnLine*, 48(16), 273-278.  
<https://doi.org/10.1016/j.ifacol.2015.10.292>
- Altosole, M., Benvenuto, G., Campora, U., Laviola, M., & Zaccone, R. (2017). Simulation and performance comparison between diesel and natural gas engines for marine applications. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 231(2), 690-704.  
<https://doi.org/10.1177/1475090217690964>
- Altosole, M., Campora, U., Figari, M., Laviola, M., & Martelli, M. (2019). A Diesel Engine Modelling Approach for Ship Propulsion Real-Time Simulators. *Journal of Marine Science and Engineering*, 7(5), 138.  
<https://doi.org/10.3390/jmse7050138>
- Arredondo Díaz, M. (2013). *Estimación del torque indicado en un motor diesel*.
- Åström, K. J., Hägglund, T., Hang, C. C., & Ho, W. K. (1993). Automatic tuning and adaptation for PID controllers—A survey. *Control Engineering Practice*, 1(4), 699-714. [https://doi.org/10.1016/0967-0661\(93\)91394-C](https://doi.org/10.1016/0967-0661(93)91394-C)
- Beale, M., Hagan, M., & Demuth, H. (2020). *MATLAB Deep Learning Toolbox User's Guide*.

- Bianchi, F. D., Battista, H. de, & Mantz, R. J. (2007). *Wind Turbine Control Systems: Principles, Modelling and Gain Scheduling Design*. Springer-Verlag. <https://doi.org/10.1007/1-84628-493-7>
- Borase, R. P., Maghade, D. K., Sondkar, S. Y., & Pawar, S. N. (2020). A review of PID control, tuning methods and applications. *International Journal of Dynamics and Control*. <https://doi.org/10.1007/s40435-020-00665-4>
- Bosch. (2003). *Diesel-engine management: An overview*. Bosch.
- Cook, J. A., Sun, J., Buckland, J. H., Kolmanovsky, I. V., Peng, H., & Grizzle, J. W. (2008). Automotive Powertrain Control—A Survey. *Asian Journal of Control*, 8(3), 237-260. <https://doi.org/10.1111/j.1934-6093.2006.tb00275.x>
- Dan Foresee, F., & Hagan, M. T. (1997). Gauss-Newton approximation to Bayesian learning. *Proceedings of International Conference on Neural Networks (ICNN'97)*, 3, 1930-1935. <https://doi.org/10.1109/ICNN.1997.614194>
- Diaconescu, E. (2008). *The use of NARX Neural Networks to predict Chaotic Time Series*.
- Farouk, N., Sheng, L., & Said, L. (2012). *Speed Control System on Marine Diesel Engine Based on a Self-Tuning Fuzzy PID Controller*. 6.
- García-Nieto, S., Martínez, M., Blasco, X., & Sanchis, J. (2008). Nonlinear predictive control based on local model networks for air management in diesel engines. *Control Engineering Practice*, 16(12), 1399-1413. <https://doi.org/10.1016/j.conengprac.2008.03.010>

- Gawthrop, P., & Smith, L. (1996). *Metamodelling: Bond Graphs and Dynamic Systems*.
- Gerr, D. (2001). *The Propeller Handbook: The Complete Reference for Choosing, Installing, and Understanding Boat Propellers*.
- Guzzella, L., & Onder, C. H. (2010). *Introduction to modeling and control of internal combustion engine systems* (2. ed). Springer.
- Hafner, M., Jost, O., & Isermann, R. (2002). *Mechatronic design approach for engine management systems*. 12.
- Hafner, M., & Schu, M. (2000). Fast neural networks for diesel engine control design. *Control Engineering Practice*, 11.
- Hagan, M. T., Demuth, H., & Beale, M. (2014). *Neural Network Design* (2.<sup>a</sup> ed.). Martin Hagan. <https://hagan.okstate.edu/NNDesign.pdf>
- Hagan, M. T., & Menhaj, M. B. (1994). Training feedforward networks with the Marquardt algorithm. *IEEE Transactions on Neural Networks*, 5(6), 989-993. <https://doi.org/10.1109/72.329697>
- Hata, Y., Ikeura, K., Morita, T., & Abo, T. (1989). Engine control system using a cylinder pressure sensor. *IEE Proceedings D Control Theory and Applications*, 136(2), 84. <https://doi.org/10.1049/ip-d.1989.0013>
- Hellstrom, M. (2005). *Engine Speed Based Estimation of the Indicated Engine Torque*.
- Isermann, R. (2006). *Mechatronic Systems: Fundamentals*.
- Isermann, R. (2014). *Engine modeling and control: Modeling and electronic management of internal combustion engines*. Springer.

- Isermann, R., Schaffnit, J., & Sinsel, S. (1999). Hardware-in-the-loop simulation for the design and testing of engine-control systems. *Control Engineering Practice*, 11.
- Janakiraman, V. M., Nguyen, X., & Assanis, D. (2013). Nonlinear identification of a gasoline HCCI engine using neural networks coupled with principal component analysis. *Applied Soft Computing*, 13(5), 2375-2389.  
<https://doi.org/10.1016/j.asoc.2013.01.006>
- Jensen, J.-P., Kristensen, A. F., Sorenson, S. C., Houbak, N., & Hendricks, E. (1991). *Mean Value Modeling of a Small Turbocharged Diesel Engine*. 910070. <https://doi.org/10.4271/910070>
- Kalogirou, S. A. (2003). Artificial intelligence for the modeling and control of combustion processes: A review. *Progress in Energy and Combustion Science*, 29(6), 515-566. [https://doi.org/10.1016/S0360-1285\(03\)00058-3](https://doi.org/10.1016/S0360-1285(03)00058-3)
- Kao, M., & Moskwa, J. J. (1995). Turbocharged Diesel Engine Modeling for Nonlinear Engine Control and State Estimation. *Journal of Dynamic Systems, Measurement, and Control*, 117(1), 20-30.  
<https://doi.org/10.1115/1.2798519>
- Karnopp, D. C., Margolis, D. L., & Rosenberg, R. C. (2012). *System Dynamics: Modeling, Simulation, and Control of Mechatronic Systems*. John Wiley & Sons.
- Korb, R., Jörgl, H. P., & Lutz, B. (1999). Nonlinear Dynamic Modelling of a Gas Engine Using an RBF-Network. *Mathematical and Computer Modelling of Dynamical Systems*, 5(2), 133-151.  
<https://doi.org/10.1076/mcmd.5.2.133.6171>

- Llamas, X., & Eriksson, L. (2019). Control-oriented modeling of two-stroke diesel engines with exhaust gas recirculation for marine applications. *Proceedings of the Institution of Mechanical Engineers, Part M: Journal of Engineering for the Maritime Environment*, 233(2), 551-574.  
<https://doi.org/10.1177/1475090218768992>
- Majors, M., Stori, J., & Cho, D. (1994). Neural network control of automotive fuel-injection systems. *IEEE Control Systems*, 14(3), 31-36.  
<https://doi.org/10.1109/37.291459>
- MathWorks. (2020a). *Control System Toolbox User's Guide*.
- MathWorks. (2020b). *Simulink Coder User's Guide*.
- Mayr, O. (1971). Maxwell and the Origins of Cybernetics. *Isis*, 62(4), 424-444.
- Muller, R., Hemberger, H.-H., & Baier, K. (1997). *Engine Control using Neural Networks: A New Method in Engine Management Systems*. 8.
- Nahim, H. M., Younes, R., Nohra, C., & Ouladsine, M. (2015). Complete modeling for systems of a marine diesel engine. *Journal of Marine Science and Application*, 14(1), 93-104. <https://doi.org/10.1007/s11804-015-1285-y>
- Narendra, K. S., & Parthasarathy, K. (1991). Learning automata approach to hierarchical multiobjective analysis. *IEEE Transactions on Systems, Man, and Cybernetics*, 21(1), 263-272. <https://doi.org/10.1109/21.101158>
- Omran, R., Younes, R., & Champoussin, J.-C. (2009). Optimal Control of a Variable Geometry Turbocharged Diesel Engine Using Neural Networks: Applications on the ETC Test Cycle. *IEEE Transactions on Control*

*Systems Technology*, 17(2), 380-393.

<https://doi.org/10.1109/TCST.2008.2001049>

Raptodimos, Y., & Lazakis, I. (2020). Application of NARX neural network for predicting marine engine performance parameters. *Ships and Offshore Structures*, 15(4), 443-452.

<https://doi.org/10.1080/17445302.2019.1661619>

Ribbens, W. (2017). *Understanding Automotive Electronics: An Engineering Perspective*.

Romero Ugalde, H. M. (2008). *Identificación de sistemas utilizando redes neuronales*.

Rotondo, D. (2018). *Advances in Gain-Scheduling and Fault Tolerant Control Techniques*. Springer International Publishing.

<https://doi.org/10.1007/978-3-319-62902-5>

rusEFI. (2020). *RusEFI wiki*. GitHub. <https://github.com/rusefi/rusefi>

Scattolini, R., Siviero, C., Mazzucco, M., Ricci, S., Poggio, L., & Rossi, C. (1997). Modeling and identification of an electromechanical internal combustion engine throttle body. *Control Engineering Practice*, 5(9), 1253-1259. [https://doi.org/10.1016/S0967-0661\(97\)84364-X](https://doi.org/10.1016/S0967-0661(97)84364-X)

SECU-3. (2020). *Características del SECU-3*. МПС3 SECU-3 / Ignition and fuel injection system. <https://secu-3.org/RU/features/>

Seiler, P., Packard, A., & Gahinet, P. (2020). An Introduction to Disk Margins. *arXiv:2003.04771 [cs, eess, math]*. <http://arxiv.org/abs/2003.04771>

- Shamma, J. S., & Athans, M. (1991). Guaranteed properties of gain scheduled control for linear parameter-varying plants. *Automatica*, 27(3), 559-564.  
[https://doi.org/10.1016/0005-1098\(91\)90116-J](https://doi.org/10.1016/0005-1098(91)90116-J)
- Speeduino. (2020). *Speeduino Manual*. Speeduino Manual.  
<http://wiki.speeduino.com/en/home>
- Tadros, M., Ventura, M., & Soares, C. (2015). Numerical simulation of a two-stroke marine diesel engine. En C. Soares, R. Dejhalla, & D. Pavletic (Eds.), *Towards Green Marine Technology and Transport* (pp. 609-617). CRC Press. <https://doi.org/10.1201/b18855-80>
- Taylor, D. A. (1996). *Introduction to marine engineering* (2nd ed., rev). Butterworth Heinemann.
- Tong, Q., Xie, H., Song, K., & Zou, D. (2019). A Control-Oriented Engine Torque Online Estimation Approach for Gasoline Engines Based on In-Cycle Crankshaft Speed Dynamics. *Energies*, 12(24), 4683.  
<https://doi.org/10.3390/en12244683>
- Wang, R., Li, X., Ahmed, Q., Liu, Y., & Ma, X. (2018). Speed Control of a Marine Engine using Predictive Functional Control based PID Controller. *2018 Annual American Control Conference (ACC)*, 3908-3914.  
<https://doi.org/10.23919/ACC.2018.8431287>
- Weigang Pan, Hairong Xiao, Yaozhen Han, Changshun Wang, & Guiyong Yang. (2010). Nonlinear active disturbance rejection controller research of main engine for ship. *2010 8th World Congress on Intelligent Control and Automation*, 4978-4981.  
<https://doi.org/10.1109/WCICA.2010.5554754>

- Woodyard, D. F. (Ed.). (2009). *Pounder's marine diesel engines and gas turbines* (9. ed). Elsevier.
- Yanakiev, D., & Kanellakopoulos, I. (1995). *Engine and Transmission Modeling for Heavy-Duty Vehicles*. 62.
- Yin, J., Su, T., Guan, Z., Chu, Q., Meng, C., Jia, L., Wang, J., & Zhang, Y. (2017). Modeling and Validation of a Diesel Engine with Turbocharger for Hardware-in-the-Loop Applications. *Energies*, 10(5), 685.  
<https://doi.org/10.3390/en10050685>
- Yu, M., Tang, X., Lin, Y., & Wang, X. (2018). Diesel engine modeling based on recurrent neural networks for a hardware-in-the-loop simulation system of diesel generator sets. *Neurocomputing*, 283, 9-19.  
<https://doi.org/10.1016/j.neucom.2017.12.054>
- Zhao, D., Liu, C., Stobart, R., Deng, J., Winward, E., & Dong, G. (2014). An Explicit Model Predictive Control Framework for Turbocharged Diesel Engines. *IEEE Transactions on Industrial Electronics*, 61(7), 3540-3552.  
<https://doi.org/10.1109/TIE.2013.2279353>
- Zhao, J., Hu, Y., & Gao, B. (2019). Sequential Optimization of Eco-Driving Taking Into Account Fuel Economy and Emissions. *IEEE Access*, 7, 130841-130853. <https://doi.org/10.1109/ACCESS.2019.2939837>
- Zhao, J., Zhou, S., Hu, Y., Ju, M., Ren, R., & Chen, H. (2019). Open-source dataset for control-oriented modelling in diesel engines. *Science China Information Sciences*, 62(7), 77201, s11432-018-9783-x.  
<https://doi.org/10.1007/s11432-018-9783-x>