**ÍNDICE DE CONTENIDOS**

1. ***Introducción 5***
2. ***Descripción del Sistema 7***
   1. Presentación del sistema 7
   2. Análisis del sistema 7
3. ***Identificación y Definición del Sistema 10***
   1. Introducción 10
   2. Descripción de las técnicas abordadas para la identificación del sistema 10
   3. System Identification Toolbox™ 11
      1. Black – Box Modeling 13
         1. Adquisición de datos 14
         2. Tratamiento de datod 15
         3. Aplicación de estructuras tipo Black – Box 18
      2. ODE Parameter Estimation. Grey – Box Modeling 23
         1. Definición y ajuste del sistema simplificado 24
         2. Generación de Inputs y Outputs 26
         3. Aplicación de ODE Parameter Estimation 27
   4. Optimización de Parámetros 33
      1. ¿En qué consiste la Optimización de Parámetros? 34
      2. Aplicación al sistema real 35
      3. Modelo Analítico No Lineal 37
      4. Descripción de los módulos del modelo 39
      5. Elección de los parámetros a optimizar 45
      6. Resultados tras optimización 46
      7. Estudio de sensibilidad del modelo resultante 47
      8. Valoración del modelo resultante 64
4. ***Herramientas para la identificación de fallos en el sistema 65***
   1. Introducción 65
   2. Técnica basada en la Optimización de Parámetros 66
      1. Estructura de funcionamiento 66
      2. Valoración 67
   3. Redes Neuronales Artificiales (RNA) 67
      1. ¿Qué son las RNA? ¿Cómo funcionan? 68
      2. Arquitectura de una RNA 69
      3. Entrenamiento y aprendizaje de una RNA 75
      4. Diseño de RNA para patrones de presión en cámara fijos 77
      5. Diseño de RNA para patrones de presión en cámara variables 84
5. ***Conclusiones y Resultados 92***
   1. Introducción 92
   2. Identificación y Definición del Sistema 93
   3. Técnicas para Diagnosis Predictiva de Fallos 93
   4. Efectos de de grietas en cigüeñal y fallos de inyección sobre la dinámica 96
6. ***Desarrollo Futuro 99***
   1. Introducción 99
   2. Líneas de Investigación 99
   3. Implementación 100
7. ***Bibliografía 102***

## Agradecimientos

A mis padres, Pedro y Virtudes, por recibir incondicionalmente su apoyo y calor durante el período de desarrollo del proyecto y durante toda mi etapa académica.

A mi tutor, Dr. Francisco J. Jiménez Espadafor, por compartir toda su experiencia, sus consejos y su apoyo durante este tiempo en el que hemos trabajado conjuntamente para desarrollar este proyecto. Sin él, no hubiera sido posible.

A la Escuela Técnica Superior de Ingeniería en general, por despertar en mí la pasión que siento por la ingeniería y por demostrarme que *“querer es poder”*.

A mis amigos y enemigos, porque gracias a ellos soy quien soy.

*Gracias.*

## Introducción

*1.*

Son muchos los fallos que se pueden dar en un motor de combustión interna alternativo: mecánicos, eléctricos, de naturaleza termodinámica, etc. En función del tamaño y aplicación a la que esté destinada el motor, las consecuencias del fallo o fallos tendrán un mayor o menor impacto: no es lo mismo, en lo que a consecuencias se refiere, un fallo de combustión en un motor destinado a la automoción que el mismo tipo de fallo en un motor estacionario destinado al abastecimiento eléctrico de una población.

El proyecto fin de carrera que se desarrolla en este texto, tiene como objeto la búsqueda, diseño y análisis de una herramienta informática capaz de identificar y predecir la existencia de fallos de naturaleza mecánica en un motor de combustión interna alternativo estacionario destinado a la generación de energía eléctrica. El interés que suscita el hecho de disponer de una herramienta que proporcione información en “tiempo real” del estado del motor es tal que, con el desarrollo de esta herramienta se conseguirían situaciones como:

* Programación y planificación de las paradas de mantenimiento preventivo, con el aumento de eficiencia en términos de gestión de la operación del motor.
* Incremento de la vida útil del motor, con los beneficios económicos que esto conlleva.
* Aplicación de medidas de mantenimiento tales que eviten fallos mayores como roturas en cigüeñal u otros elementos, que acarrearían la parada del motor y por tanto el desabastecimiento energético para el que está proyectado.

Gracias a la información que es posible adquirir del motor en la actualidad (magnitudes físicas como par motor, velocidad instantánea, presión en cámara de combustión, etc.) y al trabajo desarrollado previamente en el campo de la diagnosis predictiva de fallos, se inicia con este proyecto una labor de investigación y desarrollo enfocada a:

* Identificación del sistema y análisis de la dinámica del mismo.
* Desarrollo de técnicas para la diagnosis y mantenimiento predictivo del sistema.

El esquema que sigue el proyecto, a grandes rasgos, sobre cada una de sus etapas es el de:

* Documentación.
* Desarrollo, análisis y valoración de resultados de la etapa en cuestión.

Posteriormente al desarrollo, análisis y valoración de cada una de las etapas, se muestra una sección resumen dónde se presentan esquemáticamente todos los resultados obtenidos en este proyecto.

Una vez sea puesto de manifiesto los resultados que ofrece este proyecto fin de carrera, se plantean eventuales líneas de investigación enfocadas a la ampliación de los resultados aquí obtenidos en base a otras líneas de trabajo y con objeto de complementar estos resultados.

## Descripción del Sistema

\_\_ *2.*

## 2.1 Presentación del sistema.

El sistema objeto de estudio consiste en el motor de combustión interna alternativo estacionario alojado en una central de producción de energía eléctrica, cuya ubicación se sitúa en la ciudad de Mahón, al este de la isla de Menorca (Islas Baleares, España).

El motor desarrolla su potencia según un ciclo Diesel turboalimentado de 2 tiempos. Para la transformación de potencia mecánica en potencia eléctrica, se hace uso de un alternador alojado en uno de los extremos del cigüeñal.

## 2.2 Análisis del sistema.

Con objeto de conocer de forma más detallada el sistema real con el que se va a trabajar, se procede a continuación a hacer una descripción más detallada del mismo.

El sistema, tal y como previamente se ha esbozado, consiste en un motor a combustión interna alternativo. Dicho motor trabaja según un ciclo operativo sobrealimentado de dos tiempos diesel. Desarrolla potencia capaz de cubrir un rango operativo desde los 8 MW eléctricos hasta los 14 MW eléctricos. Realmente el motor, en condiciones de operación, entrega una potencia ligeramente superior tanto a los 8 MW como a los 14 MW anteriormente indicados, ya que se ha de tener en consideración las pérdidas energéticas que se dan en el alternador a la hora de acometer la transformación de energía mecánica en energía eléctrica.

Sus especificaciones técnicas principales son:



1. Tabla de especificaciones técnicas del motor con el que se trabaja.

Para la generación de dicha potencia eléctrica el motor trabaja a régimen constante. Por tanto, el parámetro que regula la potencia desarrollada es el par motor como consecuencia de la presión interna desarrollada en la cámara de combustión de los cilindros.

La presión interna se regula, siendo un ciclo Diesel, de forma cualitativa según la cantidad de combustible inyectado en las cámaras de combustión de los cilindros. Dada la demanda energética a la que se enfrenta el sistema a la hora de producir potencia, es posible observar 4 regímenes operativos claramente diferenciables. Esto no excluye el hecho de que el sistema produzca potencia a regímenes intermedios, pero por ser esos 4 regímenes operativos aquéllos en los que el sistema trabaja con mayor frecuencia, se va a centrar el estudio en torno a esas 4 condiciones operativas, las cuáles se definen como:

* 8MW eléctricos producidos a 125rpm con una presión interna máxima de 77bar.
* 10MW eléctricos producidos a 125rpm con una presión interna máxima de 88bar.
* 12MW eléctricos producidos a 125rpm con una presión interna máxima de 102bar.
* 14MW eléctricos producidos a 125rpm con una presión interna máxima de 105bar.

Dado que el motor tiene alojado en uno de los extremos de su eje motor un medidor de par y captadores de presión en el interior de las cámaras de combustión de los cilindros, es posible tener conocimiento de:

* La energía introducida en el motor, es decir, la presión desarrollada por cada uno de los cilindros. Por tanto, se toma como *INPUT* del sistema las curvas de presión a lo largo de un ciclo completo de cada uno de los 10 cilindros del motor.
* La energía desarrollada por el conjunto completo de los 10 cilindros a la salida del cigüeñal y antes del alternador, en términos de par. Por tanto, se establece como *OUPUT* del sistema dicho par a la salida del cigüeñal.

Por tanto, al tener conocimiento de la presión desarrollada en el interior de los cilindros y del par de salida del mismo, el sistema se esboza como un modelo mecánico del cigüeñal, el cual, es sometido a la acción de los pares de cada cilindro y al resistivo ofrecido por el alternador. Solo queda definirlo al completo.

# Identificación y Definición del Sistema

## 3.

## 3.1 Introducción.

Con objeto de tener completamente definido el sistema y por tanto poder trabajar con un modelo analítico que represente fielmente la realidad operativa del sistema, se va a llevar a cabo una labor de identificación y definición del mismo.

El interés que presenta el hecho de disponer de un modelo analítico de un sistema, el cual, es objeto de estudio, reside en la posibilidad de observar, estudiar y analizar la dinámica del sistema y ver qué parámetros intervienen y, con qué peso, en la misma.

Actualmente existen muchas técnicas para la definición o análisis del modelo analítico de un sistema, apoyadas en teorías o ciencias muy dispares. A continuación, se exponen las técnicas que han sido analizadas y utilizadas para la identificación.

## 3.2 Descripción de las técnicas abordadas para la identificación del sistema.

Para la identificación y por tanto definición del sistema real, se han abordado en este proyecto dos técnicas diferentes, las cuales proporcionan resultados diferentes:

* System Identification Toolbox™.
* Optimización de Parámetros.

Existen muchas técnicas para la identificación de sistemas y de sus parámetros. Sin embargo, en este proyecto se focaliza la atención en estas 2 herramientas diferentes debido a que tras un proceso de documentación y asesoramiento, se configuran como las más adecuadas para el objetivo final que se persigue.

La primera de las técnicas abordadas, se apoya en un paquete operativo incluido en el software de cálculo numérico Matlab ®. Dicho paquete (toolbox a partir de ahora) recibe el nombre de *System Identification Toolbox.* Este paquete se presenta como una herramienta con la que abordar la identificación de sistemas lineales o no lineales a partir de magnitudes medidas del sistema, por lo que en principio se aventura como herramienta eficaz para las necesidades que presentan. Posteriormente, se desarrollará tanto la estructura interna de este toolbox así como los resultados que arroja y las conclusiones que se pueden tomar en base a dichos resultados.

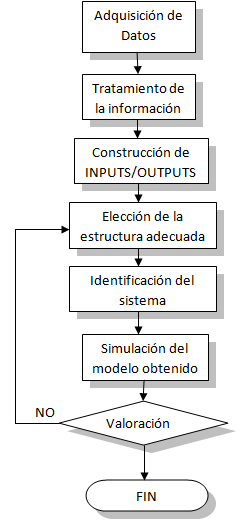
La segunda de las técnicas abordadas se apoya también en el software de cálculo numérico Matlab®, pero esta vez no se hace uso de ningún toolbox adicional, sino que se hace uso de la capacidad de programación y cálculo que dicho software ofrece. Concretamente en este caso, la forma de proceder es la siguiente: se construye un modelo analítico del sistema, en el que se dejan ciertos parámetros libres para así aplicar una técnica de optimización según ciertas restricciones impuestas al comportamiento del modelo. Con esto, se obtiene un modelo analítico coherente y con una fiel representación del comportamiento real del sistema. Al igual que con System Identification Toolbox, posteriormente se desarrollará toda la estructura, resultados y conclusiones de esta segunda técnica, la cual, recibe el nombre de *Optimización de Parámetros*.

## 3.3 System Identification Toolbox™.

Tal y como se ha comentado previamente, donde se ha hecho una ligera presentación de esta herramienta, System Identification Toolbox ofrece la posibilidad de construir un modelo matemático, previa adquisición de datos, que sea capaz de representar tanto un sistema con dinámica lineal como con dinámica no lineal. Por tanto, las utilidades que se adivinan con esta herramienta son la de analizar el comportamiento dinámico del sistema, simulación a partir de INPUTS dados y/o predicción del comportamiento del mismo.

Para poder construir y por tanto identificar el sistema objeto de estudio, es necesario disponer de una base de datos, que se divide en INPUTS y OUTPUTS. Una vez se disponga de dicha base de datos, que posteriormente se explicará cómo ha de ser construida, se ha de elegir la estructura que más se ajuste al comportamiento y/o casuística del sistema real. Llegado a ese punto, se procede a usar el toolbox en calidad de usuario y analizar los resultados que éste arroja.

Por tanto, la forma de proceder para usar esta herramienta de forma eficiente se ilustra según el siguiente diagrama:



1. Diagrama de flujo del proceso de identificación con estructuras Black – Box.

Este paquete operativo ofertado por el software Matlab®, ofrece la posibilidad de estimar tanto modelos lineales como no lineales de sistemas dinámicos a partir de información adquirida del propio sistema real. Dicha información, puede se proporcionada tanto en el dominio del tiempo como en el dominio de la frecuencia. Para la estimación de un modelo para el sistema real, es necesario definir en primer lugar el tipo de estructura interna que poseerá el modelo y si es de tipo lineal o no lineal. Esta última consideración, va de la mano de la elección de la estructura del modelo.

Todas las estructuras que se pueden elegir para la estimación de un modelo, se pueden englobar en tres grandes grupos:

* Modelos lineales y no lineales de tipo caja negra (Black- Box Modeling).
* Modelos lineales y no lineales basados en estimación de parámetros ODE (ODE Grey – Box Parameter Estimation).
* *Time – Series model identification.* Esta estructura sólo precisa de uno o más outputs y ningún input para la estimación del modelo.

Puesto que la dinámica del sistema que se trata de identificar es fuertemente no lineal y que como objetivo se tiene la obtención de un modelo que represente de manera fiel al sistema real, las estructuras que se utilizarán para probar la potencia de esta herramienta son dos:

* Identificación de modelos no lineales de tipo caja negra.
* Identificación de modelos no lineales basados en estructuras ODE.

A continuación, se abordarán ambas y se pondrá en conocimiento sus bases así como los resultados que arroja cada una.

## 3.3.1 Black – Box Modeling.

### Bases.

Los modelos tipo black - box se basan en complejas estructuras polinómicas, las cuales, dependen del orden y de los coeficientes, como en cualquier polinomio. El orden de estas estructuras ha de ser elegido a priori antes de comenzar con el proceso de identificación del sistema, siendo los coeficientes del polinomio los parámetros a ajustar por el toolbox.

Cabe mencionar aquí, que existen varias estructuras de modelos tipo black - box. Estructuras lineales o no lineales y definidas en el dominio del tiempo o en el dominio de la frecuencia.

En este caso, se va a identificar el sistema en base a modelos tipo black – box lineales y no lineales, definidos en el dominio del tiempo.

### Limitaciones de los modelos tipo Black – Box.

El principal problema de estos modelos es que, aunque den buen resultado (cosa que se comprobará a lo largo del texto), no ofrecen un significado físico, es decir, del modelo polinómico no se puede extraer una relación con los parámetros físicos del sistema real.

La única información de la que se dispondrá una vez que el sistema haya sido identificado por la herramienta en base a esta tipología de modelos, serán los coeficientes del polinomio. Con dichos coeficientes no se podrá encontrar una interpretación de los parámetros físicos que caracterizan el sistema real. Este hecho es la principal causa que haga desechar esta opción de identificación.

En cualquier caso, se va a hacer un análisis de la potencia que los modelos tipo black – box ofrecen. Por lo tanto, independientemente de la estructura que se utilice para identificar el sistema y con objeto de ver cómo se comporta esta herramienta, se va a hacer uso de un sistema simplificado respecto al sistema real.

Los inputs y outputs para este sistema simplificado son:

* 10 INPUTS que corresponden a la presión interna de la cámara de combustión de cada uno de los 10 cilindros. Dichas presiones se transforman en par, para que tanto input como output sean coherentes en unidades.
* 1 OUTPUT correspondiente al par medido a la salida del cigüeñal.

Los niveles de carga en los que trabaja el sistema simplificado son: 10MW, 12MW y 14MW.

Se considera como un sistema simplificado porque simplemente se están considerando las presiones internas de cada cilindro como inputs y el par a la salida del cigüeñal como output, obviando toda la complejidad interna que el sistema real presenta.

A continuación se detalla el proceso de adquisición de datos y tratamiento previo, que es común para todas las estructuras de modelos tipo black - box que se utilizarán para tratar de identificar el sistema.

## 3.3.1.1 Adquisición de Datos.

Anteriormente se ha comentado que el sistema real tiene alojado ciertos dispositivos para la adquisición de datos referentes a:

* Presión interna de cada uno de los 10 cilindros.
* Par motor a la salida del cigüeñal.

Con dichos dispositivos en operación, se inicia el proceso de adquisición de datos, el cual, no consiste en más que adquirir información para cada nivel de carga (10MW, 12MW y 14MW) y en 9 ocasiones distintas, por lo que se llega a disponer de:

* 9 registros de presión interna por cilindro y por cada nivel de carga.
* 9 registros de par a la salida del cigüeñal por cada nivel de carga.

Una vez que se dispone de toda el paquete de datos descrito anteriormente, se ha de proceder a un tratamiento previo de dicha información en base a depuración y transformación (a términos de par) de la misma.

El hecho de adquirir información en 9 ocasiones distintas es motivado por la intención de disponer de una vasta base de datos a la que posteriormente aplicar el proceso de depuración y que reste por tanto una base de datos representativa de la casuística del sistema real.

Por último y antes de comenzar con la descripción detallada del proceso de depuración y transformación de la información adquirida por los dispositivos alojados en el sistema real, cabe mencionar que como cualquier dispositivo de adquisición de datos, la señal que finalmente proporciona al ordenador contiene un nivel de ruido que dificulta el post tratamiento de la misma, por lo que se acomete una operación de filtrado y suavizado de la señal adquirida, mediante una función de librería de Matlab ® (csaps)

## 3.3.1.2 Tratamiento de la información adquirida.

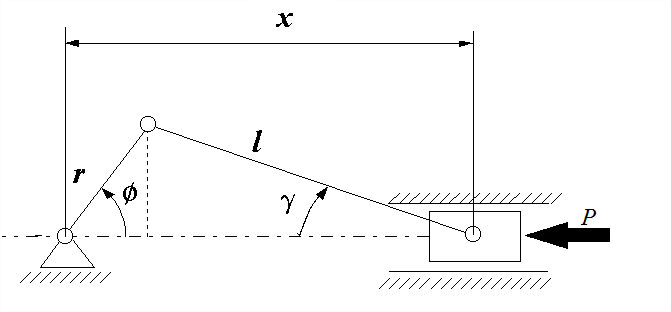
### Depuración de los datos.

El proceso de depuración de los datos surge de la intención de disponer de una información que represente de la manera más fiel posible el comportamiento real del sistema. Por ello, aunque el motor este trabajando a un nivel de carga de por ejemplo 10 MW, nunca va a ofrecer 10 MW justos. Es por ello que se hacen varias mediciones (9 en este caso) con las que poder disponer de un rango completo para cada nivel de carga.

Para aplicar este proceso, se toma un criterio en base a una tolerancia referida a la potencia media entregada en cada nivel de carga. La potencia media entregada ha de estar muy próxima al valor nominal, es decir, el nivel de carga. El registro que no cumpla la tolerancia impuesta, se elimina.

Para poder aplicar este proceso, es necesario transformar la información adquirida referente a la presión interna de los cilindros a información en términos de potencia, previo paso por términos de par.

Para hacer una transformación de presión a par, es necesario hacer uso de las ecuaciones de lazo de un mecanismo biela – manivela referido en este caso al sistema real. Dichas ecuaciones de lazo son las que se muestran a continuación:



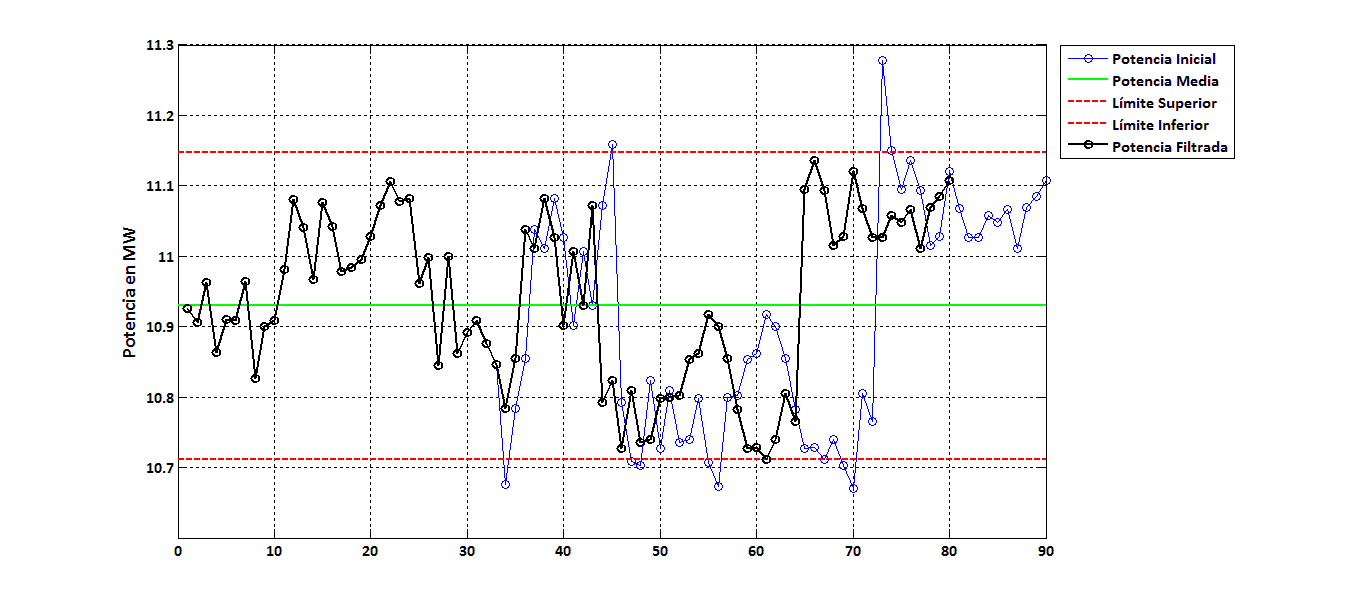
1. Representación del mecanismo biela – manivela, con los parámetros básicos.



Una vez que se ha transformado la información a términos de par, la conversión a términos de energía mecánica es inmediata.

Disponiendo de los datos en términos de energía mecánica para todos y cada uno de los 10 cilindros en los 9 ensayos realizados, se está en condiciones de aplicar el proceso de depuración.

A continuación se muestra una representación donde se puede apreciar la potencia media ofrecida por cada uno de los cilindros en cada uno de los 9 ensayos de adquisición de datos. Aplicando una tolerancia del 2% sobre la potencia media de todos los cilindros para todos los ensayos, el registro que no esté dentro de la tolerancia, se elimina. La representación mostrada a continuación corresponde a la ejecución del depurado sobre los datos adquiridos para 10MW, siendo similar el caso para el resto de niveles de carga (12MW y 14MW).



1. Representación del proceso de filtrado de la información adquirida, 10 MW en este caso.

En la figura 3 se aprecia un vector inicial (curva azul) de 90 componentes. Dichas componentes hacen referencia a la potencia media de cada uno de los 10 cilindros para los 9 ensayos. Los registros que no cumplen la tolerancia impuesta, es decir, los registros que se encuentran o por encima o por debajo de las horizontales de color rojo, son eliminados y por tanto el vector resultante (color negro) contiene un número inferior a 90 componentes, 80 en el ejemplo mostrado. Cabe mencionar que en este punto, cuando se habla de registros, es lo mismo que hablar de ensayos.

Este proceso de depurado de la información adquirida se realiza para todos los niveles de carga en los que trabaja el motor.

### Reestructuración de la información.

Aplicado el proceso de depuración y en conocimiento de los registros que son susceptibles de ser eliminados, se procede construir los Inputs y Outputs.

Los Inputs finales (en términos de información disponible) con los que se va a trabajar son 10 vectores de par correspondientes a los 10 cilindros. Puesto que antes del proceso de filtrado se disponía de 10 matrices para cada cilindro, ya que cada matriz contenía los 9 ensayos por cilindro, y posteriormente al proceso de filtrado se dispone de 10 matrices para cada cilindro (esta vez con un número diferente de columnas = ensayos), lo que se hace es pasar para cada cilindro de una matriz a un vector, haciendo para ello, la media todos los ensayos que han sido considerados como válidos. Véase a continuación un esquema ilustrativo del proceso.



1. Esquema orientativo del proceso de filtrado y reestructuración de la información adquirida.

Habiéndose identificado los ensayos que se consideran válidos, se hace un tratamiento similar con las matrices correspondientes al par a la salida del cigüeñal, es decir, con las matrices que sirven para construir el Output.

Por último, se normaliza la información ya depurada con objeto de que el orden de magnitud con el que se trabaje sea la unidad. La normalización beneficiará tanto los tiempos de cálculo como la precisión obtenida.

Hecho todo lo anterior, se dispone de toda la información necesaria para poder comenzar a utilizar System Identification Toolbox.

### Generación de Inputs / Outputs.

Llegado el punto en el que se dispone de toda la información ya tratada y lista para ser utilizada, se han de crear los paquetes de datos con los que System Identification Toolbox puede trabajar. Estos paquetes de datos son nombrados por la bibliografía como iddata (identification data).

Para generar un iddata es necesario proporcionar al menos 1 Input y al menos 1 Output, pudiendo ascender dichas cantidades al antojo del usuario en función de las necesidades. En este caso se va a generar un iddata multiexperimental que contemple toda la información posible, es decir, que contenga los datos adquiridos para los 3 niveles de carga distintos. Por lo tanto, se dispondrá de un paquete de datos que contendrá:

* 10 Inputs y 1 Output para cada uno de los 3 niveles de carga o experimentos distintos, por lo que se tiene un total de 30 Inputs y 3 Outputs.

## 3.3.1.3 Aplicación de las estructuras de modelos tipo Black – Box.

Dentro de este apartado, se puede hacer una clara distinción entre la aplicación de estructuras para modelos lineales y para modelos no lineales.

Se analizarán ambos grupos de estructuras con objeto de ver qué grupo ofrece un mejor ajuste y comportamiento.

### Linear Model Identification.

De forma general, los modelos lineales siguen la siguiente configuración:

Linear Poly eq.bmp

* ,,, y  son los distintos polinomios que configuran la estructura de identificación, que dependen del operador temporal .
* representa el/los Outputs.
* representa el/los Inputs, donde  hace referencia al retraso que existe entre los Inputs. En este caso, el retraso es 0.
* hace referencia al ruido.

En función de los polinomios que tomen partido en la estructura final de identificación, se configuran las distintas estructuras lineales:

* ARX (toman partido A, B y k).
* ARMAX (toman partido A, B, C y k).
* Output – Error (toman partido B, F y k).
* Box – Jenkins (toman partido B, C, D, F y k).

El primer paso para la identificación es elegir una de las estructuras lineales anteriores. Posteriormente se estima el orden de los polinomios (según qué estructura, el toolbox proporciona un estimador orientativo) y por último se da comienzo al proceso de identificación (ajuste de los coeficientes de los polinomios). Puesto que a priori no se conoce qué estructura ofrecerá un mejor ajuste, se van a valorar las 4.

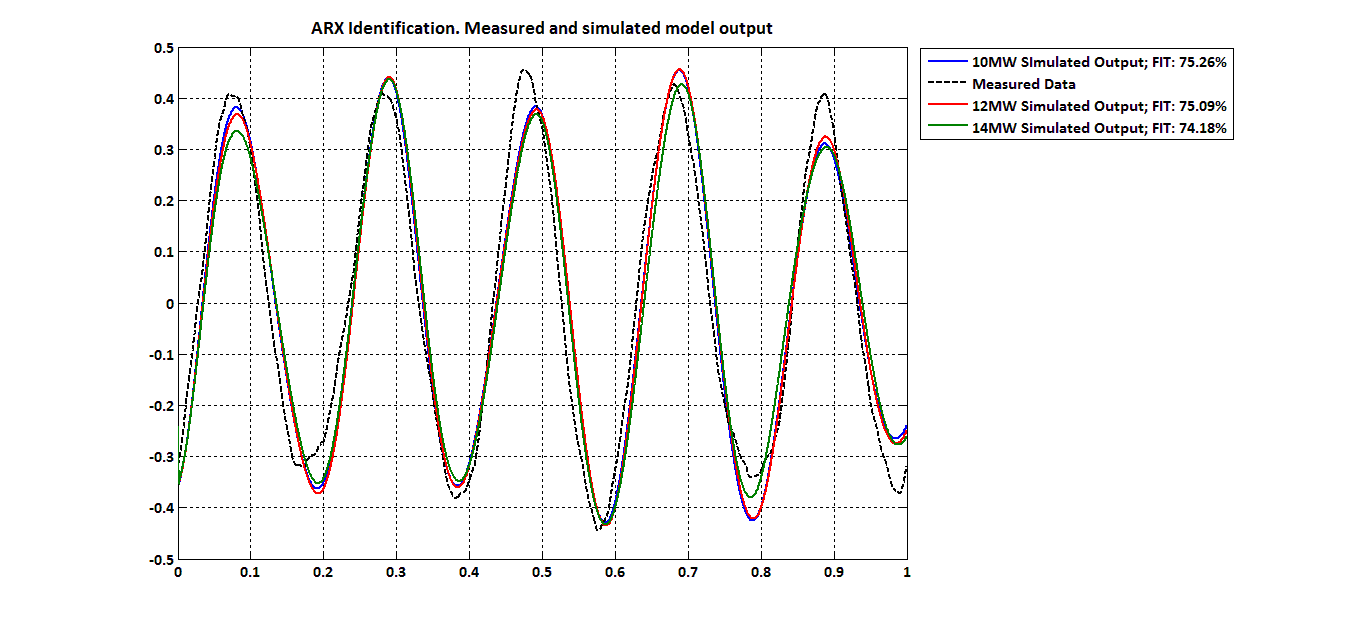
La forma de proceder en la identificación del sistema por parte de las diferentes estructuras lineales, es la siguiente:

* Identificación del sistema a partir del iddata multiexperimental.
* Simulación del modelo obtenido para todos los niveles de carga.
* Valoración de los ajustes obtenidos.

a.1) Estructura ARX.

Para este tipo de estructuras, System Identification Toolbox ofrece la posibilidad de estimar los órdenes de los polinomios, en vez de tener que elegirlos manualmente, en base al mejor ajuste encontrado.

Se realiza por tanto la estimación automática de los órdenes y se da comienzo a la identificación del sistema.



1. Representación del ajuste ofrecido por estructura lineal ARX510 (na=5, nb=1, nk=0).

Puesto que para las estructuras tipo ARX, la selección de los órdenes para los polinomios se hacen de forma automática por parte del toolbox, no se hacen más pruebas, siendo el modelo resultante el siguiente:

* ARX con na=5 (orden del polinomio A), nb=1 (orden del polinomio B) y retraso nulo.

Véase que, aunque el ajuste del modelo no es del todo malo, no supera apenas el 75% de ajuste respecto a los datos adquiridos.

Cabe mencionar en este punto, que el cálculo del ajuste por parte del toolbox, sigue la siguiente expresión:

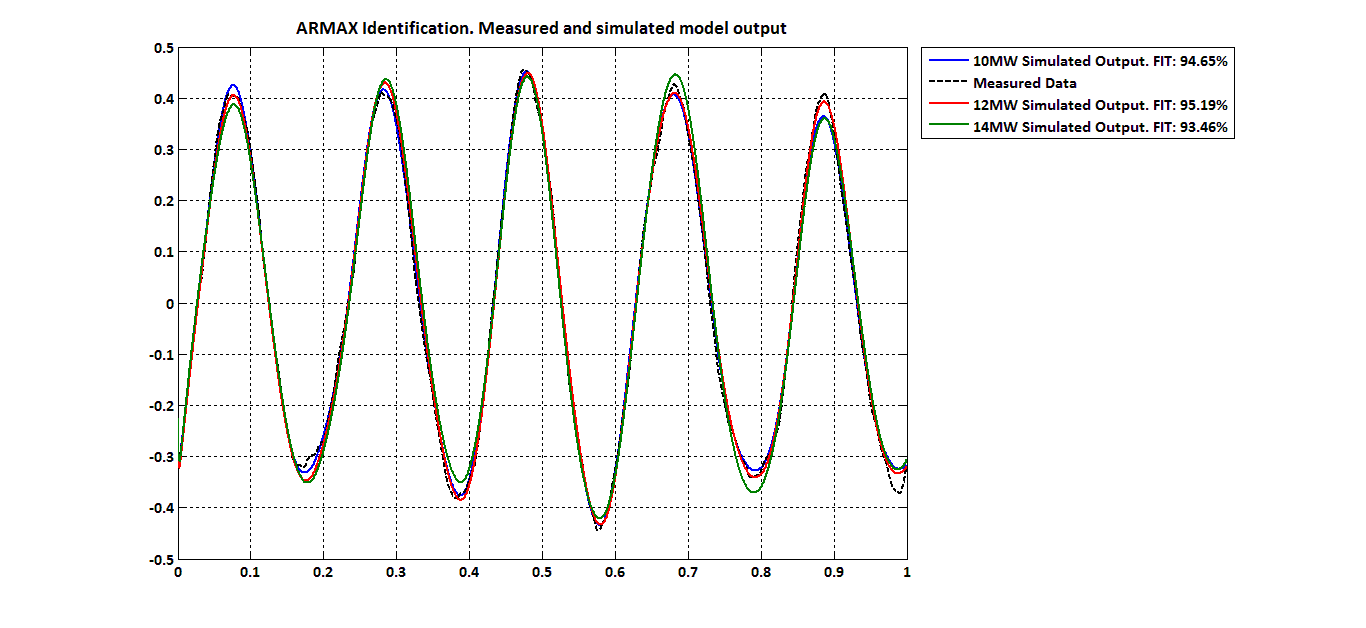


a.2) Estructuras ARMAX.

En este caso, la elección de los órdenes para los polinomios ha de ser acometida por el usuario, por lo que es en este caso donde ha de hacerse una labor de prueba y error hasta conseguir el máximo ajuste posible.

Puesto que lo que se está haciendo es una labor de ajuste, es coherente pensar en que cuanto mayor sean los órdenes de los polinomios que se encargan la identificación, mayor será el grado de ajuste. Por tanto, ha de llegarse a un compromiso entre los órdenes de los polinomios y el efecto indeseable de overfitting, muy común en las labores de identificación de sistemas y entrenamiento de redes.

Tras varias estimaciones, se llega a un modelo tipo ARMAX3300 (na=3, nb=3, nc=0 y retraso nulo) que ofrece los siguientes ajustes:



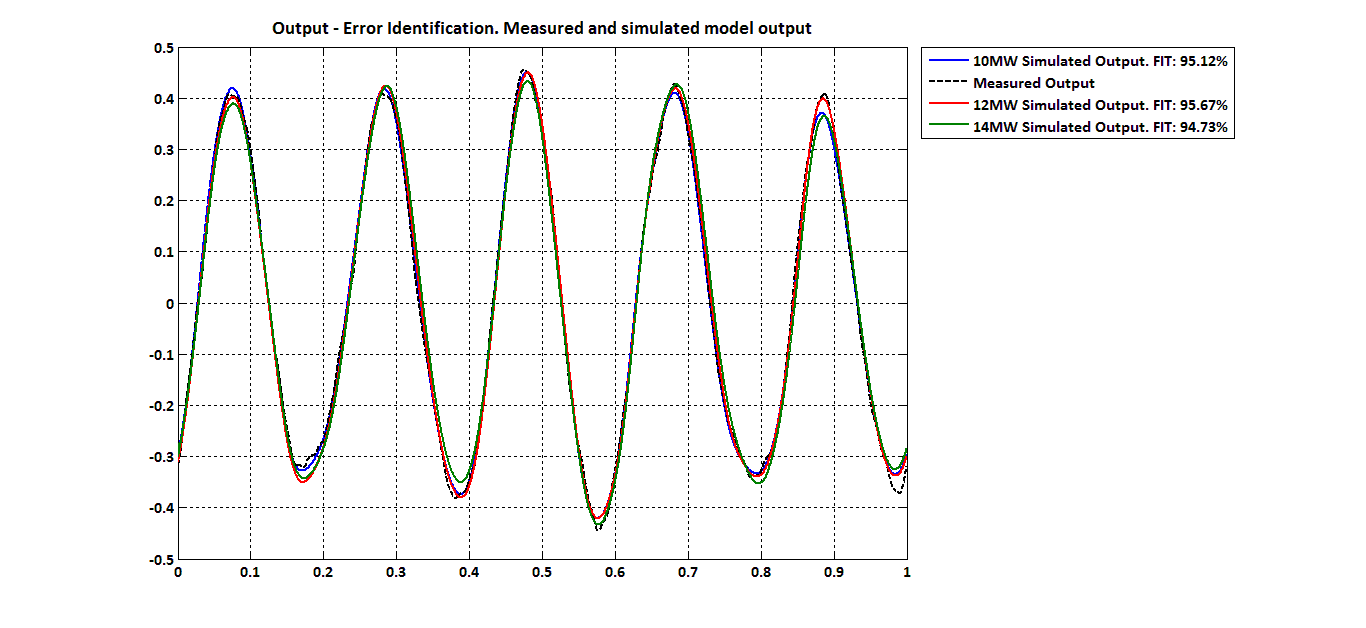
1. Representación del ajuste ofrecido por la estructura lineal ARMAX3300 (na=3, nb=3, nc=0, nk=0).

Se fijan los órdenes de los polinomios haciendo varias pruebas y viendo el comportamiento del ajuste, es decir, viendo en qué momento el ajuste comienza a descender.

En este caso el ajuste ofrecido es muy superior al que la estructura tipo ARX ofrecía (75%).

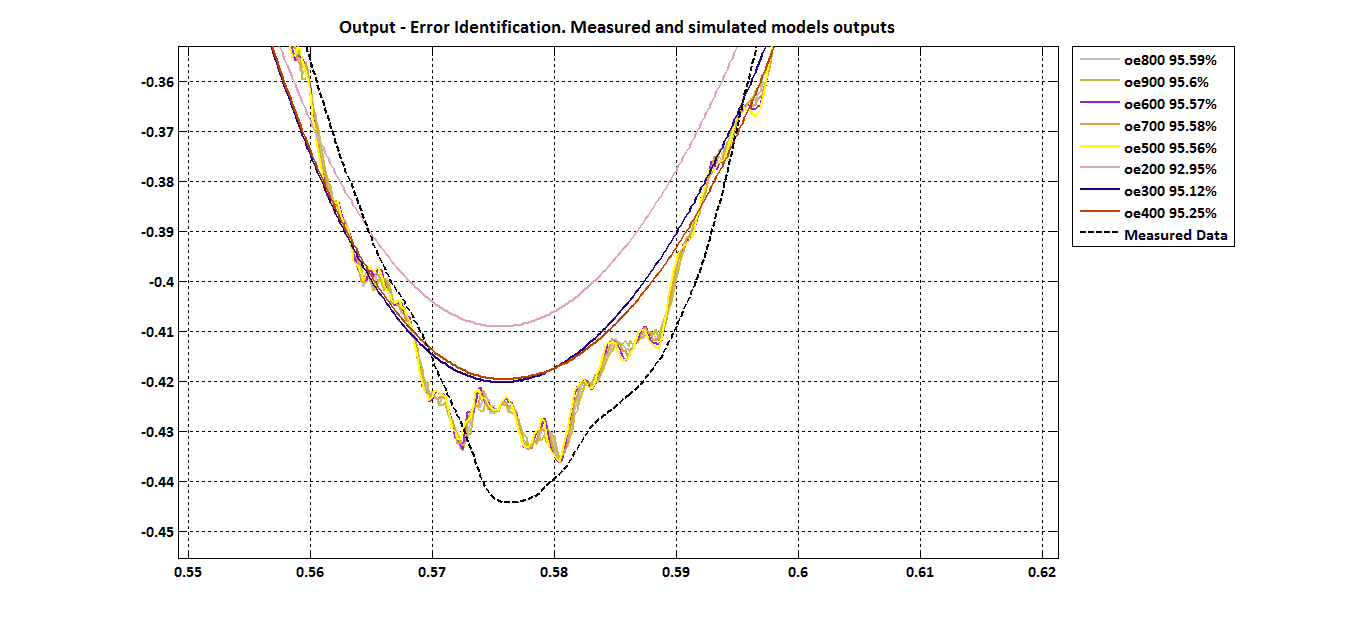
a.3) Estructuras Output – Error.

Para este tipo de estructuras se procede de la misma manera que en el caso anterior, obteniendo un modelo OE300 (nb=3, nf=0 y retraso nulo).



1. Representación del ajuste ofrecido por la estructura OE300 (nb=3, nf=0, nk=0).

En este tipo de estructuras, se puede encontrar un ajuste aún más elevado pero a costa de introducir mucho ruido en la señal. Véase el detalle en la siguiente figura, en la que se representan modelos con ajuste más elevado:



1. Representación del ruido generado en la señal simulada debido a efectos de overfitting.

Por tanto, de dar esta estructura como válida, se ha de elegir el modelo OE300 ya que un aumento de los órdenes implica una generación de ruido no admisible.

a.4) Estructuras Box – Jenkins.

Por requerimientos operativos de la aplicación, la configuración de los órdenes para los polinomios que intervienen en esta estructura lineal de identificación es tal, que el resultado es una estructura tipo ARMAX, por lo tanto no se utiliza esta estructura para la identificación del sistema.

### Non Linear Model Identification.

Estas estructuras de identificación son muy similares a las estructuras lineales que anteriormente se han desarrollado, solo que ofrecen mayor flexibilidad a la hora de tener en cuenta las no linealidades que presenta el sistema real.

Según la bibliografía, es recomendable utilizar estas estructuras de identificación cuando se da alguna de las siguientes situaciones:

* Las estructuras lineales de identificación ofrecen un ajuste muy pobre.
* Existen no linealidades en el proceso de adquisición.
* El sistema es fuertemente no lineal.

En este caso, exceptuando la condición de que el sistema sea fuertemente no lineal, no se dan ninguna de las otras dos situaciones, por lo tanto no se va a entrar en el estudio de las estructuras no lineales ya que al haber encontrado estructuras lineales que ofrecen un buen ajuste, es redundante buscar modelos no lineales dadas las limitaciones que presentan los modelos Black – Box.

## 3.3.2 ODE Parameter Estimation. Grey – Box Modeling.

De las herramientas ofertadas por *System Identification Toolbox* que ofrecen una potencial aplicación al caso presente, se ha visto que los modelos tipo Black – Box no dan la oportunidad de obtener un modelo analítico del que se pueda extraer información de los parámetros y características físicas del sistema real, puesto que los modelos resultantes presentan estructuras polinómicas cuyos coeficientes no tienen significado físico.

Sin embargo, la estimación de parámetros ODE si oferta la posibilidad de obtener un modelo analítico cuyos parámetros tengan un significado físico.

### Bases.

Los parámetros ODE (Ordinary Differential Equation), tal y como su nombre indica, son los coeficientes de las ecuaciones de un sistema de ecuaciones diferencial y ordinario. Esta herramienta ofrece la posibilidad de hacer una estimación de dichos coeficientes, previa aportación de un paquete de datos adquiridos como en los casos anteriores. Además, es necesaria la definición por parte del usuario de la estructura de la matriz del sistema de ecuaciones diferenciales ordinarias, en la que están presentes los coeficientes a estimar.

### Limitaciones.

Las limitaciones de esta herramienta vienen dadas por las propias bases en las que se apoya, es decir, es necesario conocer la dinámica completa del sistema si se quiere obtener un modelo analítico completo que represente fielmente el comportamiento real del sistema. Esto es una limitación desde el punto de vista de tener que conocer toda la dinámica además de, si se quiere una fiel representación, tener que incluir no linealidades.

En cualquier caso, aunque las limitaciones de esta herramienta hagan difícil su aplicación, se realiza un análisis y estudio de la misma, como en el caso anterior (Black – Box Modeling).

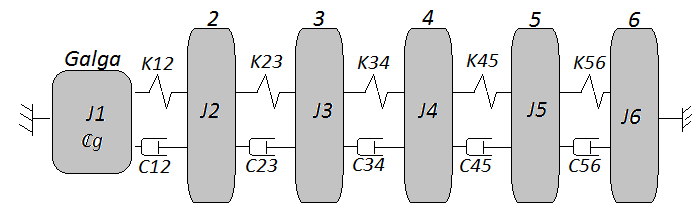
Al igual que en el caso anterior, Black – Box Modeling, para probar la potencia de esta herramienta se hará uso de un sistema simplificado y definido por el usuario para así evitar la limitación que ofrece el hecho de desconocer la dinámica completa del sistema y, por lo tanto, poder ver la funcionalidad de la herramienta. Se procede a identificar un sistema que ya es conocido y completamente definido.

Además, es con dicho sistema simplificado con el que se van a generar los outputs que, junto con el par de cada uno de los 5 cilindros más el par del grado de libertad denominado como galga a modo de inputs, van a alimentar la herramienta de identificación.

## 3.3.2.1 Definición y ajuste del sistema simplificado.

En este caso se define un sistema de 6 grados de libertad, basado en un modelo mecánico de torsión del cigüeñal, sobre el que actúan los torsores generados por las presiones internas de desarrolladas en las cámaras de combustión de los cilindros y el par resistivo que presenta el alternador o galga, para un grado de carga en este caso.

Un esquema orientativo de cómo se configura el sistema simplificado es el siguiente:



1. Esquema representativo del sistema simplificado de 6 g.d.l.

Véase que en el grado de libertad nombrado como Galga, existe un parámetro que servirá para el ajuste energético del sistema, ya que para que el sistema sea coherente es necesario que la energía introducida en el sistema por parte de los cilindros sea igual a la energía disipada por el amortiguamiento estructural de las muñequillas más la energía eléctrica de salida producida por el alternador.

El sistema de ecuaciones de movimiento del sistema simplificado es el siguiente:



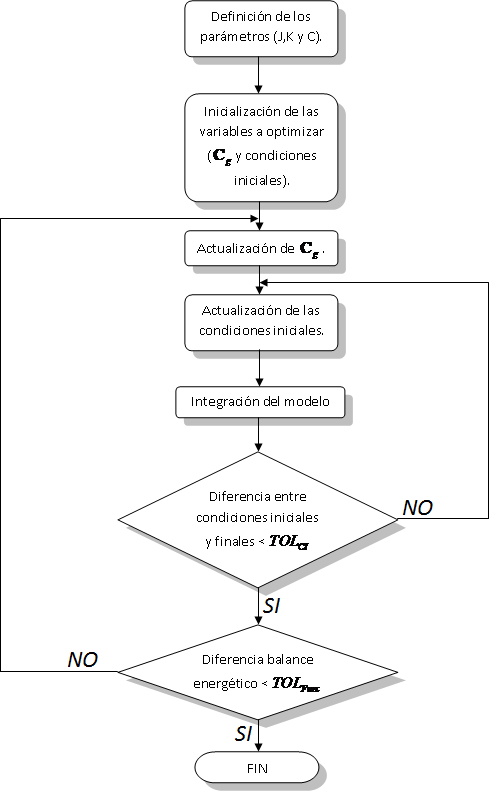
En la definición del sistema simplificado se toman valores para los parámetros muy representativos y con los mismos órdenes de magnitud que el sistema real. Es decir, se toman los siguientes valores para los parámetros físicos de rigideces, amortiguamientos e inercias:



Definidos los parámetros del sistema, solo resta ajustar el valor de de forma que el sistema sea congruente energéticamente. Para ello, se va a utilizar un proceso iterativo de optimización de parámetro en base a minimizar la diferencia en el balance energético. Además de exigir que la diferencia en el balance energético sea menor que una tolerancia impuesta (), se impone una condición adicional:

* La diferencia entre las condiciones iniciales de las variables de integración y el valor final de las variables de integración en un ciclo, ha de ser menor que una tolerancia impuesta (), ya que dicha diferencia influye en el balance energético.

Por tanto, ya que la bondad del balance energético depende fundamentalmente de dos condiciones, el proceso para encontrar el valor de la constante es el siguiente:



1. Diagrama de flujo del proceso de ajuste del sistema simplificado de 6 g.d.l.

## 3.3.2.2 Generación de Inputs / Outputs.

Una vez que se concluye el proceso de ajuste del modelo simplificado, se generan los paquetes de datos (a modo de datos adquiridos) con los que alimentar la herramienta. Dichos paquetes de datos consisten en:

* INPUTS: torsor generado por cada uno de los 5 cilindros y torsor resistivo del alternador o torsor galga.
* OUTPUTS: evolución temporal del grado de libertad i, es decir, y/o de su derivada primera .

Cabe comentar en este punto que la única información que es generada a partir del modelo simplificado ya ajustado, es la evolución temporal de todos los grados de libertad y sus derivadas. La información referente a los Inputs proviene de la adquisición de datos que se hizo con anterioridad para el uso de estructuras de identificación tipo Black – Box Modeling, por lo que no se va a reiterar la exposición del proceso de adquisición y tratamiento de la información. Véanse las secciones 3.3.1.1 y 3.3.1.2.

## 3.3.2.3 Aplicación de ODE parameter estimation.

La aplicación de la estimación de parámetros ODE (Grey – Box Modeling) consiste en construir el sistema de ecuaciones diferenciales que gobiernan el comportamiento del sistema, en función de los parámetros que quieren ser estimados. Así pues, en este caso se construyen las ecuaciones diferenciales en función de la inercia (), amortiguamiento () y rigidez () de cada grado de libertad, por lo que la cantidad de parámetros a estimar es de 16. Sin embargo, se introducen las siguientes simplificaciones:

* .
* .
* .
* .
* .
* 

Por lo tanto, con las simplificaciones anteriores, el número de parámetros libres es de 6.

Una vez construido el sistema como función de los parámetros anteriores, lo que sigue es proceder a iniciar el proceso de estimación, partiendo de unos valores iniciales para los parámetros. Se toman como valores iniciales, valores que entren dentro del orden de magnitud de cada uno de los parámetros.

La primera estimación, con  y como INPUTS y  como OUTPUT, da los siguientes resultados:



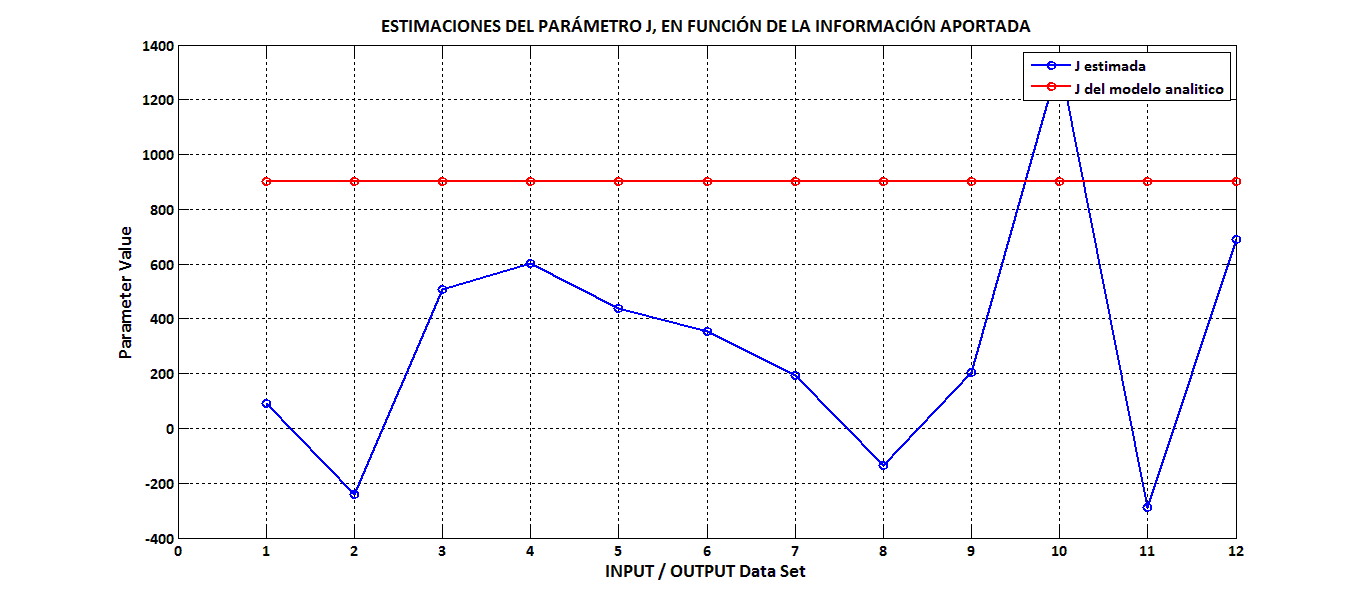
Se puede observar que los resultados distan mucho de lo esperado, por lo que se prueba a aportar una base de datos (Inputs / Outputs) más amplia con el objetivo de proporcionar más información a la herramienta y ver si de esa forma los resultados obtenidos se ajustan más a lo esperado.

Dado que con el modelo analítico se tiene la posibilidad de obtener las evoluciones temporales de todos los grados de libertad, así como de sus derivadas, se procede a la identificación del sistema con los paquetes de datos que a continuación se detallan, teniendo en cuenta que los INPUTS son siempre los mismos con independencia del paquete de datos aportado, siendo los OUTPUTS los que cambian:

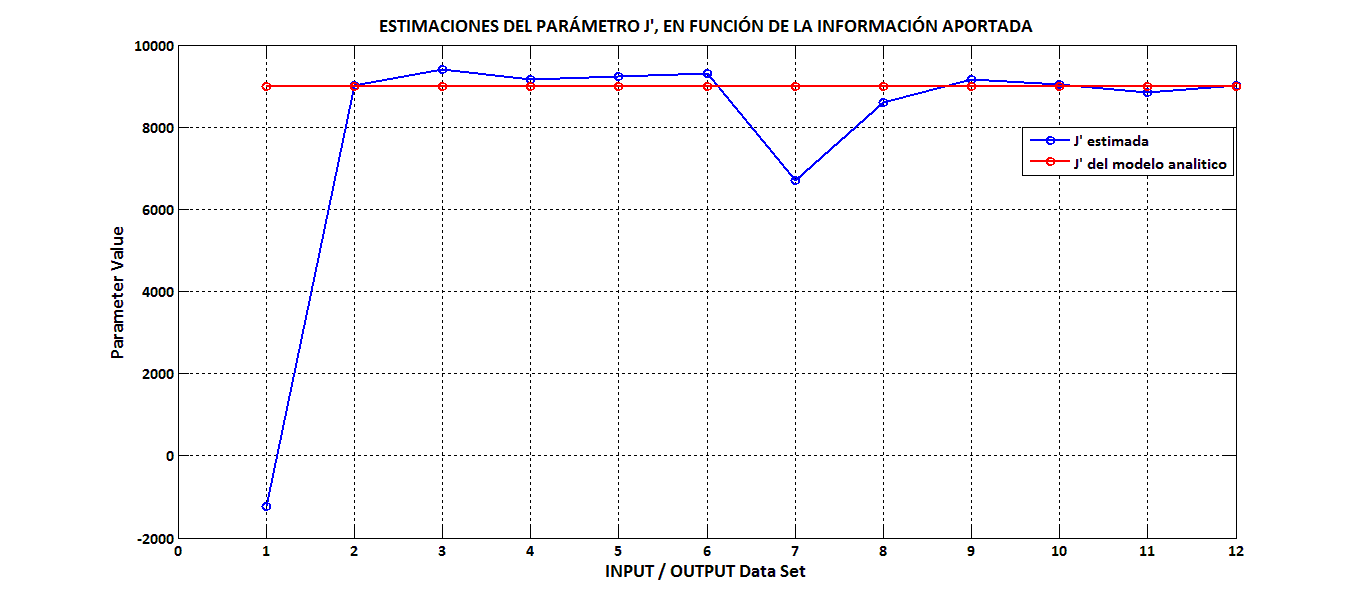
* Paquete de datos 1: .
* Paquete de datos 2:  y .
* Paquete de datos 3: ,  y .
* Paquete de datos 4: , , y .
* Paquete de datos 5: , ,,  y .
* Paquete de datos 6: , ,, , y .
* Paquete de datos 7:  y .
* Paquete de datos 8: y .
* Paquete de datos 9: .
* Paquete de datos 10: .
* Paquete de datos 11: .
* Paquete de datos 12: .

Véase que el paquete de datos 12 aporta las evoluciones temporales de todos los grados de libertad y de sus derivadas primeras, es decir, aporta la máxima información posible.

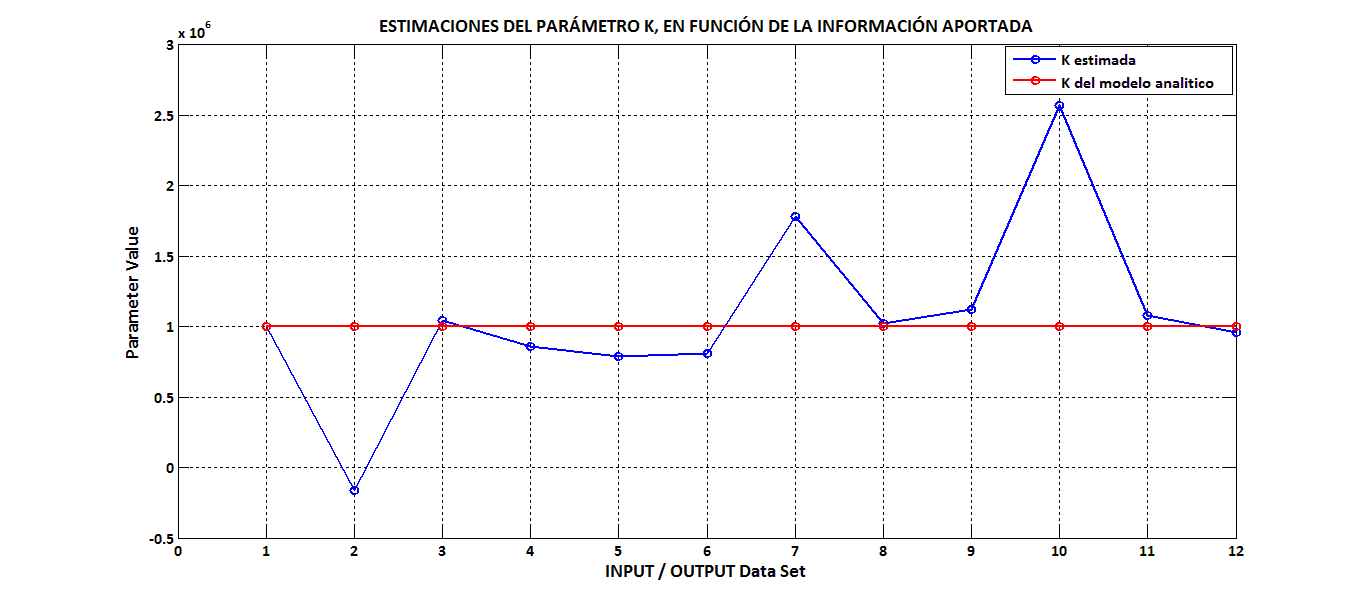
A continuación se muestran los resultados obtenidos, de forma gráfica, para cada uno de los parámetros:



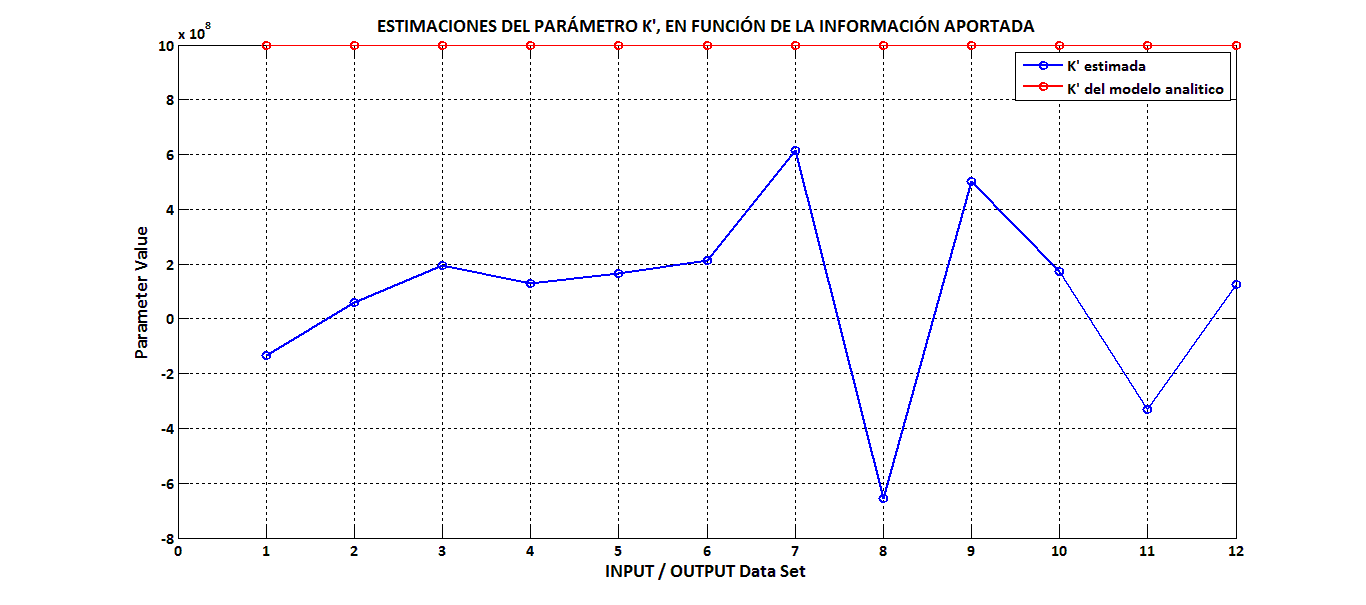
1. Representación de las estimaciones del parámetro J, en función del paquete de datos.



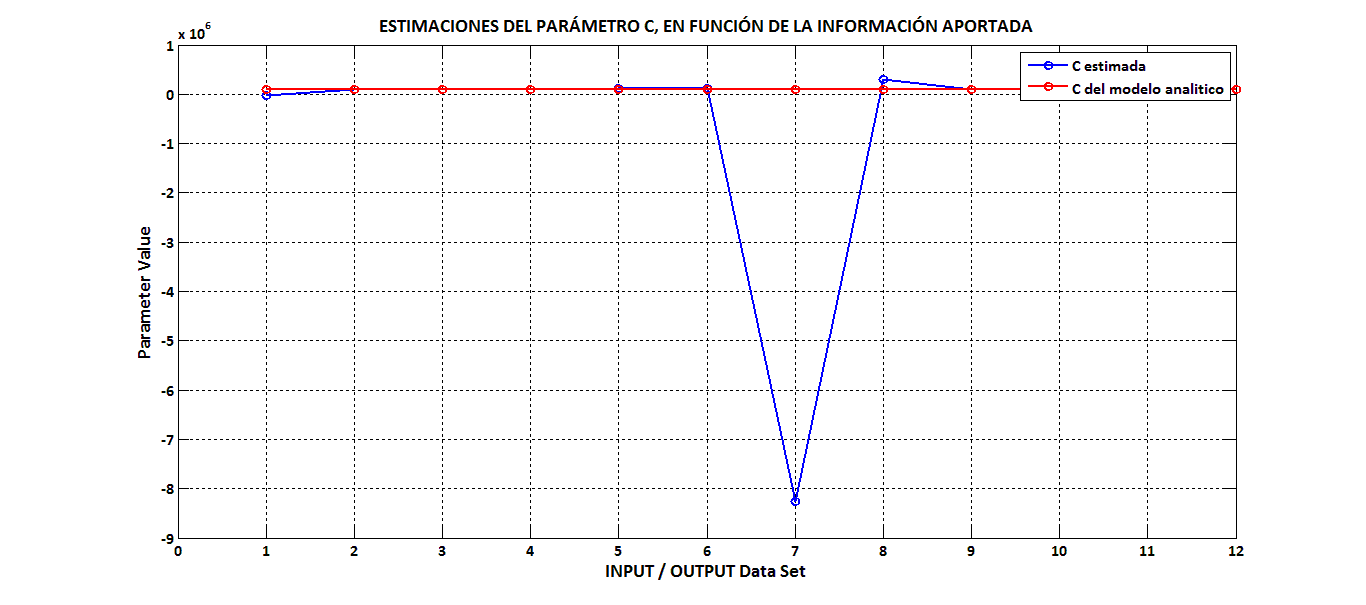
1. Representación de las estimaciones del parámetro J’, en función del paquete de datos.



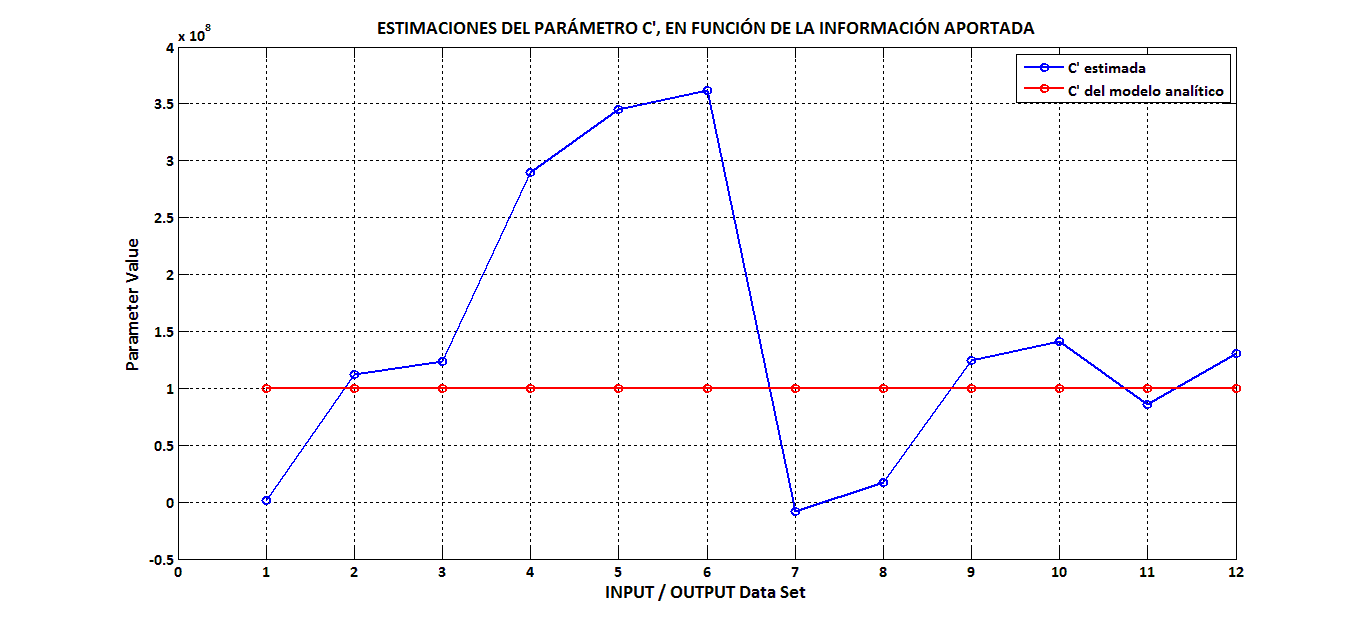
1. Representación de las estimaciones del parámetro K, en función del paquete de datos.



1. Representación de las estimaciones del parámetro K’, en función del paquete de datos.



1. Representación de las estimaciones del parámetro C, en función del paquete de datos.



1. Representación de las estimaciones del parámetro C’, en función del paquete de datos.

A la vista de los resultados que se han obtenido, se concluye que la herramienta de estimación de parámetros ODE (ODE Parameter Estimation, Grey – Box Modeling) no es capaz de realizar una estimación fiel de los valores de los parámetros físicos, por lo que no se puede utilizar dicha herramienta para la identificación del sistema real que es objeto de estudio.

En cualquier caso, antes de cerrar esta sección, se muestra a continuación una tabla donde se pueden ver los errores relativos de los parámetros estimados respecto a sus valores originales, con objeto de reafirmar la conclusión anterior. En dicha tabla se pueden encontrar los errores relativos de cada parámetro según el paquete de datos utilizado a la hora de alimentar la herramienta de estimación. Además, sombreado en verde, se han marcado los errores medios relativos más bajos y en la esquina inferior derecha (sombreado en azul) se muestra el error medio global de todos los experimentos:



1. Tabla de errores relativos obtenidos en las estimaciones de parámetros mediante Grey – Box Modeling.

Por tanto, vistos los órdenes de magnitud de los errores obtenidos, se ha de descartar esta herramienta para la identificación del sistema.

## 3.4 Optimización de parámetros.

La optimización de parámetros es la segunda técnica de identificación de sistemas que ha sido abordada en este proyecto con el fin de disponer de un modelo analítico que represente el comportamiento real del sistema con un buen grado de ajuste.

Anteriormente se han visto varios módulos de identificación de sistemas proporcionados por el paquete operativo *System Identification Toolbox* incluido en Matlab®, los cuales, no han proporcionado resultados que se puedan considerar como satisfactorios en el marco de lo que se está buscando, ya que, aunque los modelos proporcionados por la estimación lineal de Black – Box Modeling arrojaban ajustes muy buenos, no proporcionaban modelos cuyos parámetros tengan un significado físico, que es lo que se está buscando.

Por ello, se inicia el análisis y la aplicación de las técnicas de optimización de parámetros, cuyos resultados, como se mostrará en adelante, son muy satisfactorios en el marco de lo que se está buscando.

En primer lugar se hará una breve descripción de dichas técnicas de optimización, sus bases y limitaciones. Posteriormente se va a mostrar cómo se aplicará la técnica de optimización de parámetros escogida a la identificación del sistema real que está siendo sometido a estudio. Para ello, se va a describir el modelo analítico no lineal que se construye sobre el que se aplicará la optimización de parámetros y qué parámetros se deciden estimar y por qué. Por último se mostrarán los resultados y un estudio de la sensibilidad del modelo obtenido.

## 3.4.1 ¿En qué consiste la optimización de parámetros?

Las técnicas de optimización abordan un campo de aplicación muy amplio, ya que son muchos los problemas que surgen en la ingeniería (y fuera de ella) dónde es necesario o es de gran ayuda el uso de la optimización de parámetros.

### Bases.

Casi cualquier situación ingenieril donde se tenga que abordar un problema, existe la posibilidad de plantear dicho problema como una *función objetivo*, la cual, proporciona el resultado de dicho problema. Para una definición correcta y que exprese con detalle la casuística del problema, la *función objetivo* debe ser función de todas las variables y factores que afecten en la resolución del problema, y es la combinación más eficiente de dichas variables y factores la que proporciona el mejor resultado de la *función objetivo.*

Si existe la posibilidad de definir la *función objetivo* del problema sujeto a estudio y se imponen las restricciones o condiciones que han de cumplir las variables de del problema, se llega a un problema clásico de optimización, el cual, sigue la siguiente estructura de definición:



Donde la matriz representa las relaciones que han de guardar las variables, , para que se cumplan las restricciones impuestas por el vector . Por otro lado,  y  representan los límites inferior y superior entre los que han de estar contenidos los valores que tomen las variables. Por último  es la función objetivo que se pretende minimizar o maximizar su valor, según el caso.

Por tanto, *la optimización de parámetros se basa en la búsqueda de la mejor combinación de las variables de un problema de forma que se cumplan las restricciones del mismo así como que dicha combinación esté dentro del intervalo de definición de las variables y, además, proporcione un valor de la función objetivo lo más óptimo posible.*

### Limitaciones.

Las limitaciones de las técnicas de optimización radican principalmente en la propia definición de la técnica, es decir, es necesario hacer un proceso de definición exhaustivo tanto de la función objetivo como de las restricciones del problema, si se quiere llegar a un resultado de calidad y que sea representativo.

Es necesario que en las restricciones del problema se contemple toda la dinámica del sistema, tanto lineal como no lineal, así como los efectos causados por la adquisición de los datos a partir de los cuales se hará la optimización. Además, hay que prestar especial atención a la hora de decidir qué parámetros van a ser objeto de la optimización y qué parámetros no.

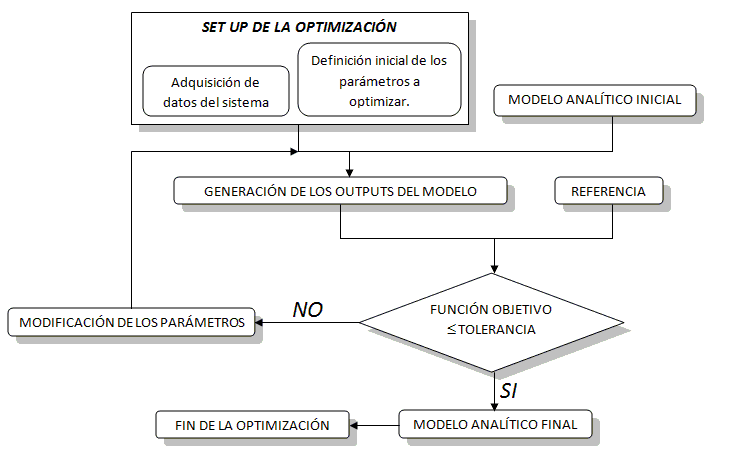
Por otra parte, las técnicas de optimización aportan una limitación adicional, que es la potencia de cálculo necesaria para poder abordar una optimización de los parámetros de un problema magnitudes considerables (no es lo mismo optimizar 5 que 50 parámetros).

Sin embargo, a pesar de todas las limitaciones que se presentan, las técnicas de optimización son una opción muy utilizada debido a los buenos resultados que ofrecen.

## 3.4.2 Aplicación al sistema real objeto de estudio.

A continuación se va a detallar la estructura de la optimización aplicada al sistema real, además de poner en conocimiento cuál es la función de optimización óptima para el caso estudiado.

Una vez que se defina el modelo analítico y sus restricciones (definición que se detallará posteriormente), se está en condiciones de lanzar la optimización, la cual, sigue el siguiente esquema de decisión:



1. Diagrama de flujo del esquema de decisión del proceso iterativo de optimización de parámetros.

Tal y como se aprecia en la figura 17, el proceso iterativo de optimización de los parámetros del sistema contiene ciertos bloques básicos. De dichos bloques, los más importantes son:

* SET UP DE LA OPTIMIZACIÓN: en este bloque se realiza un proceso de adquisición de datos que servirán para la alimentación del modelo analítico y, por tanto, para la generación de los outputs del modelo, los cuales, se utilizarán para el bloque de decisión.
* MODIFICACIÓN DE LOS PARÁMETROS: en este bloque, al no cumplirse las restricciones (tolerancia impuesta), se modifican los parámetros para volver a calcular unos nuevos outputs con los que iniciar una nueva iteración.

La adquisición de datos en esta ocasión se realiza de forma similar a las ocasiones anteriores, en las que se utilizaba la otra técnica para la identificación y definición del sistema. Sin embargo, el proceso de post – tratado de la información es distinto que en las ocasiones anteriores, puesto que sólo se aplica un proceso de filtrado de la señal adquirida con objeto de reducir el ruido que acompaña la señal, estando las transformaciones pertinentes a cargo de uno de los módulos del modelo analítico. Dicho módulo, se expondrá en detalle en la sección 3.4.3.

Por otro lado, el proceso tanto de comprobación del cumplimiento de la función objetivo como de la modificación de los parámetros corre a cuenta de la función utilizada para la optimización. Dicha función atiende al nombre de *lsqnonlin (Non Linear Least – Squares)*, que es una función de librería de Matlab®.

Como su propio nombre indica, es una función enmarcada en la optimización matemática en el sentido de los mínimos cuadrados. Se elige esta función por su potencia y capacidad para el ajuste de funciones no lineales. Su definición sigue la siguiente estructura:



Donde la función , tal y como se observa en la definición, puede estar definida en forma vectorial. Dicha característica tiene una utilidad potencial, ya que será utilizada para optimizar los parámetros del modelo de forma global, es decir, para 4 grados de carga simultáneos.

El hecho de que se optimicen los parámetros del modelo de forma global (para los 4 grados de carga simultáneamente) ofrece el resultado de disponer de un modelo que representa la dinámica del sistema real con un grado de fidelidad aún más elevado, puesto que el sistema real tiene un rango de funcionamiento que va desde los 8MW eléctricos hasta los 14MW eléctricos.

Al igual que se comentó con anterioridad durante el desarrollo de la sección 2, aunque el sistema real tenga un rango de funcionamiento comprendido en el intervalo [8MW 14MW], se observan 4 niveles de carga en los que el sistema real trabaja con más frecuencia: 8MW, 10MW, 12MW y 14MW.

Resta por tanto definir dicha función objetivo  que será objeto de minimización y optimización por parte de la función *lsqnonlin.* Para ello, hay que hacer una definición de todo el modelo analítico en el que se va a basar la optimización de parámetros.

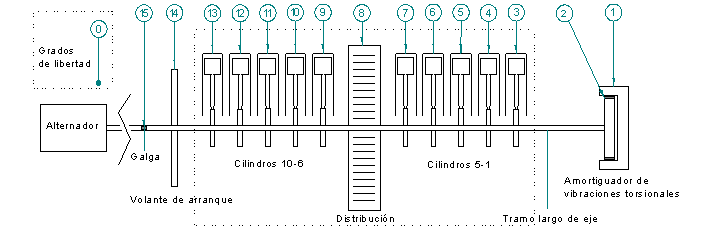
## 3.4.3 Modelo analítico

Una vez definido el esquema de decisión que seguirá la rutina de optimización, resta definir el modelo analítico que representará el sistema y todos los módulos que lo configuran.

El modelo analítico se basa en un modelo mecánico a torsión de 15 grados de libertad, libre en uno de sus extremos y empotrado en el otro. Este modelo representa la torsión de las siguientes partes del sistema:

* Tramo que va desde el amortiguador de vibraciones torsionales hasta el inicio de la primera muñequilla.
* Muñequillas de los 10 cilindros.
* Tramo central, donde está alojado el sistema de distribución del motor.
* Tramo que va desde el cigüeñal hasta el volante de inercia. Este tramo se corresponde con el grado de libertad denominado como galga, que es donde están alojados los dispositivos de adquisición de datos.
* Tramo que va desde el volante de inercia hasta el alternador. Dicho tramo se considera que termina en un empotramiento, para imponer que las oscilaciones en el alternador son nulas. Esta consideración es válida debido a la gran inercia que presenta el volante respecto al resto de elementos que toman partido en la dinámica del modelo.

A continuación se muestra un esquema de la configuración que toman los grados de libertad:



1. Esquema equivalencia grados de libertad del modelo analítico respecto al motor.

El cigüeñal es excitado por los torsores consecuencia de la presión de combustión generada en el interior de cada cilindro y por los torsores resistivos del alternador, de fricción y de inercia alternativa. El torsor resistivo originado por la fricción hace referencia a las pérdidas mecánicas del modelo.

Por otra parte, el modelo es alimentado por la siguiente información, para cada nivel de carga:

* Registros de la evolución temporal de la presión interna desarrollada en la cámara de combustión de todos y cada uno de los 10 cilindros, para generar los torsores de combustión.
* Registro de la evolución temporal del par desarrollado por el sistema, medido en el grado de libertad denominado como *galga,* a modo de referencia para la comprobación de la función objetivo*.*
* Registro de la evolución temporal de la velocidad angular del grado de libertado denominado como *galga.*
* Factores correctores de los parámetros que son sujetos a optimización.
* Número de armónicos considerados para el cálculo de los coeficientes de Fourier con los que se resolverá el problema.

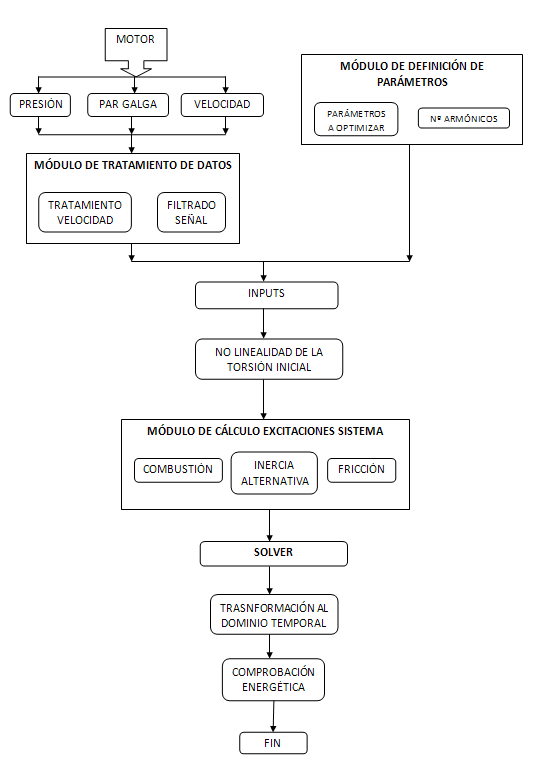
Una vez proporcionados los inputs, el modelo calcula la solución del sistema de ecuaciones del modelo a torsión. Con esta solución, se tiene la posibilidad de calcular cualquier información ulterior necesaria para el proceso de optimización o para el posterior análisis del sistema. Por tanto, no existe un paquete de outputs claramente definidos, pudiendo ser elegidos por el usuario.

El modelo analítico se resuelve en el dominio de la frecuencia, tal y como se ha comentado con anterioridad, ya que haciendo esto se aprovecha el carácter lineal que adopta el problema. Que el problema adopte un carácter lineal no quiere decir, como se verá más adelante, que no se tengan en consideración las no linealidades presentes en la dinámica del modelo.

Por tanto, haciendo uso de la expresión en términos de series de Fourier, se consigue la siguiente transformación:, donde es la matriz física del sistema y el vector de excitaciones del sistema.

## 3.4.4 Descripción de los módulos del modelo

En esta sección se describen todos los módulos que configuran el modelo completo y que sirven para el tratamiento de la información adquirida, cálculo de excitaciones, resolución del problema y comprobación de la solución obtenida. Un esquema orientativo de la estructura interna del modelo, es el siguiente:

****

1. Esquema ilustrativo de la estructura interna del modelo analítico del sistema.

### Definición de Parámetros.

En este módulo se realiza la definición y actualización de los parámetros físicos del modelo, así como de los factores necesarios para la resolución del problema.

Los parámetros físicos son los siguientes:

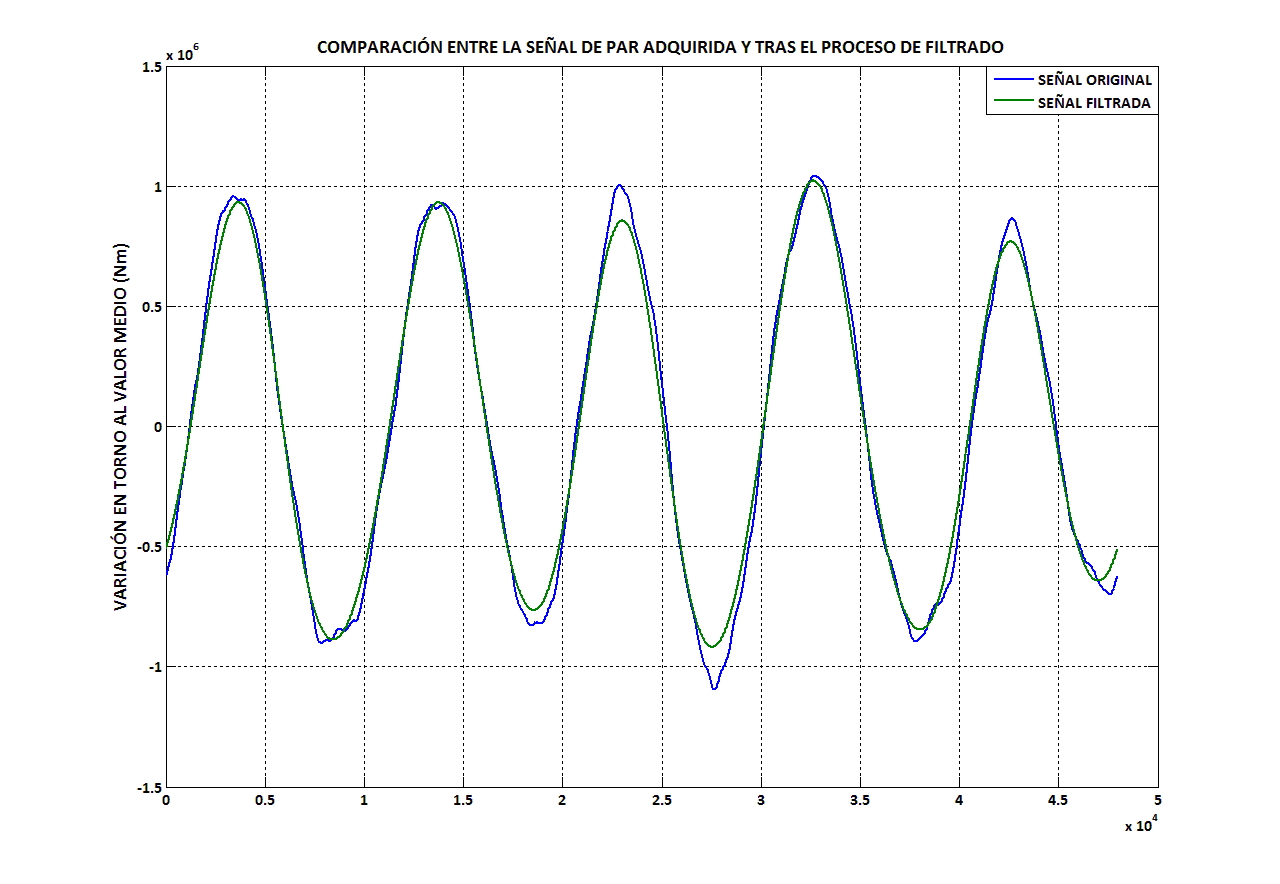
* Inercia de cada uno de los 15 grados de libertad. Vector columna 15x1.
* Rigidez de cada uno de los 15 grados de libertad. Vector columna 15x1.
* Amortiguamiento estructural histerético de cada uno de los 15 grados de libertad. Vector columna 15x1.
* Amortiguamiento a tierra de cada uno de los 15 grados de libertad. Vector columna de 15x1.
* Amortiguamiento viscoso del dispositivo amortiguador de vibraciones torsionales. Es una constante.

Por otra parte, se define un factor que no tiene un carácter físico como los anteriormente expuestos, pero que sin embargo tiene una importancia de gran magnitud de cara a la resolución del problema. Se trata de un factor corrector que tiene en consideración el desfase existente entre el inicio de la adquisición de datos y la posición real del Punto Muerto Superior del cilindro 9. Es decir, se define este factor para en tener en cuenta que, aunque se trate de calibrar el dispositivo de adquisición de datos de la forma más precisa posible, el primer registro de los datos adquiridos no coincide justamente con el PMS del cilindro que se toma como referencia (cilindro 9) y debido a este hecho se incurre en un incremento/defecto de energía eléctrica generada por el modelo que origina diferencias entre la modelada y la real que van desde los 0.8MW hasta los 8MW.

### Tratamiento de la información adquirida.

En este módulo, tal y como indica el título, se trata la información adquirida. Para ello, se toma la información referente a la velocidad angular del grado de libertad 15 (galga) y se le aplica un proceso de filtrado (eliminación de ruido) y una serie de operaciones que proporcionan la velocidad media real y el número de registros que serán relevantes para la resolución del problema. Es decir, que es el registro de velocidad angular el que regula la cantidad de información adquirida que será utilizada para la resolución del problema.

Además se realiza un proceso de filtrado sobre la señal de par adquirida con objeto de eliminar la incertidumbre que ofrece el dispositivo de adquisición. Véase la siguiente gráfica, dónde se aprecia el efecto del filtrado:



1. Señal de par antes (azul) y después (verde) del proceso de filtrado.

### Actualización de la torsión inicial para convergencia de los efectos no lineales.

En este módulo se tiene en cuenta la torsión inicial de cada uno de los grados de libertad, para la convergencia del modelo no lineal.

Este módulo surge del hecho de tener que asignar unos valores a la torsión inicial de los grados de libertad para el cálculo de los armónicos de las excitaciones. Sin embargo, estas torsiones iniciales se alejan bastante de la realidad, por lo que se han de actualizar iterativamente hasta encontrar una convergencia.

La forma de actualizar dichos valores, se hace en base a la torsión media de cada grado de libertad, que a su vez, es consecuencia de la resolución del problema. Es decir, hay que resolver el problema tantas veces como sea necesario hasta que este módulo detecte que se ha llegado a una convergencia.

### Cálculo de las excitaciones del modelo.

Tal y como se indica en el título, en este módulo se calculan todas las excitaciones que son aplicadas sobre el cigüeñal. Las excitaciones se calculan en términos de series de Fourier, tal y como se ha comentado previamente, ya que haciendo esto se consigue que el problema adopte un carácter lineal y por tanto se pueda resolver con operaciones algebraicas. Se calcula por tanto el vector del sistema .

d.1) Par de combustión.

Este sub – módulo precisa de la presión interna de la cámara de combustión de cada cilindro, además del número de armónicos que se desean tener en cuenta para la resolución del problema.

Con los datos de presión en cámara y la geometría interna del motor, se consigue la transformación a términos de par. Posteriormente, se aplican las técnicas de Fourier para la obtención de los coeficientes de la serie del par de combustión de cada cilindro.

d.2) Inercia alternativa.

En este sub – módulo se obtiene el par de inercia no lineal que presenta el mecanismo completo.

Dado que el mecanismo biela manivela en el motor está en posicionado en dirección vertical (como es común) y la aceleración de los pistones y biela no es constante, aunque si lo sea la velocidad media (realmente no es 100% constante, existen variaciones a lo largo de un ciclo), el par de inercia transmitido a cada muñequilla no es constante.

Debido a este hecho, se necesita calcular el par de inercia instantáneo en función de la posición y aceleración del pistón.

d.3) Fricción.

Se calcula el par medio de fricción, que supondrá las pérdidas mecánicas del motor. Se basa en un modelo proporcional a la carga a la que trabaje el motor.

Aunque posteriormente se comentará, cabe decir en este punto que dicho par medio de fricción está afectado por un factor que se incluye en el conjunto de los parámetros sujetos a optimización.

### Solver.

En este bloque se calcula la matriz del sistema tipo . Este módulo necesita los parámetros físicos de inercias, amortiguamientos y rigideces de todos los grados de libertad, así como los coeficientes de las series de Fourier de las excitaciones (contenidos en el vector ).

Como resultado, proporciona los coeficientes de las series de Fourier de las evoluciones temporales de cada grado de libertad, es decir, proporciona una matriz  con los coeficientes de todos los grados de libertad.

Como se indicó en puntos anteriores, gracias a la transformación a términos de series de Fourier, el problema se puede resolver con operaciones algebraicas. Concretamente, se hace uso de la inversión de la matriz del sistema: 

### Transformación de la solución al dominio temporal.

Para poder realizar las comprobaciones pertinentes, cálculo de potencias y actualización de las torsiones iniciales para la convergencia de los efectos no lineales, es necesario hacer una transformación de la solución del problema al dominio temporal. Para ello, se hace uso de la expresión de una variable temporal en términos de series de Fourier, es decir:



Donde:

* , 
* , , 

Una vez que se realiza la transformación arriba detallada, se dispone de las evoluciones temporales de todos los grados de libertad del sistema, por lo que se puede acceder a cualquier información referente a la dinámica del mismo.

### Módulo de comprobación.

Por último, en este módulo se realizan una serie de comprobaciones en términos energéticos y de par, para comprobar que la respuesta que arroja el modelo es correcta y ver el grado de ajuste con la realidad. Concretamente se realizan las siguientes operaciones:

* Se obtienen las potencias de combustión, disipación por fricción, disipación por amortiguamiento, de inercia y por último la potencia total del modelo.
* Se compara la potencia total del modelo con la potencia real adquirida mediante los dispositivos alojados en el grado de libertad 15 (galga).

De la comparación de las potencias modelo y real, se obtiene un error porcentual que refleja el grado de ajuste del modelo con respecto al sistema real. Dicho error porcentual se calcula según la siguiente expresión:



Sin embargo, aunque se puede utilizar el error porcentual arriba reflejado como *función objetivo* de cara a la optimización del modelo, se decide en su lugar utilizar el error absoluto del ajuste en potencia:



## 3.4.5 Elección de los parámetros a optimizar

Una vez mostrada la estructura interna del modelo y el esquema de decisión que sigue la rutina de optimización, falta definir qué parámetros van a ser optimizados para encontrar un ajuste entre el modelo y el sistema real lo más satisfactorio posible.

Tras varios procesos iniciales de optimización y análisis de los resultados, se encuentra un paquete de 13 parámetros que son indispensables optimizar. Dichos parámetros son los siguientes:

* Inercia de todos los grados de libertad, a excepción de la inercia del dispositivo amortiguador de vibraciones torsionales. Se suponen 5 parámetros distintos, ya que se considera que las inercias de los 10 cilindros son iguales.
* Rigidez de todos los grados de libertad, a excepción de las rigideces del dispositivo amortiguador de vibraciones torsionales y del tramo del mismo hasta la primera muñequilla. Se toma esta consideración debido a que las rigideces de estos dos tramos están perfectamente definidas. Se suponen por tanto 5 parámetros distintos, ya que al igual que en el caso de las inercias, se consideran iguales las rigideces de las muñequillas de los 10 cilindros.
* Amortiguamiento a tierra de los cojinetes del cigüeñal. Al igual que en los casos anteriores, se suponen iguales los amortiguamientos que presentan los 10 cojinetes, por lo que se añade un parámetro.
* Factor regulador del inicio del PMS. Esto supone un parámetro adicional.
* Factor regulador de las pérdidas mecánicas por fricción. El modelo de pérdidas mecánicas por fricción, tal y como se detalló con anterioridad, se supone proporcional a la carga del sistema y por ello se puede ver regulado por un factor de proporcionalidad. Esto supone el último del paquete de 13 parámetros sujetos a optimización.

## 3.4.6 Resultados

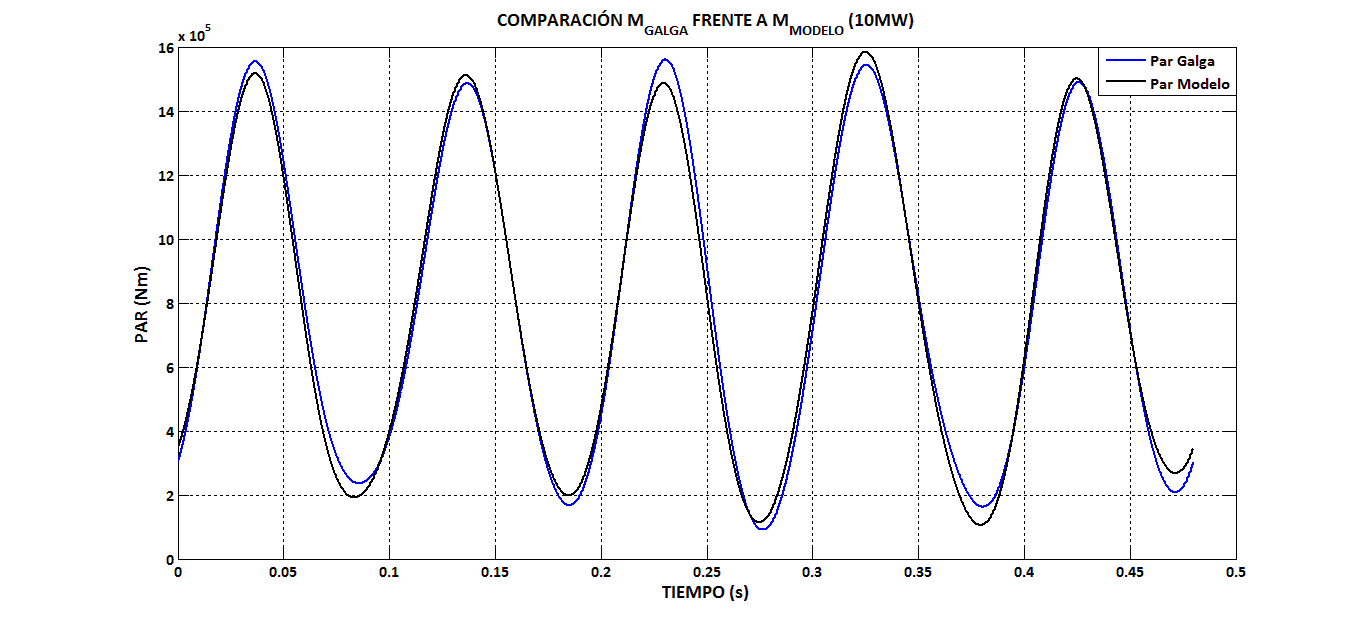
Tras el proceso de optimización de los parámetros detallados en la sección anterior, se obtienen los siguientes valores:



El parámetro de fricción es adimensional, dado que el modelo de fricción se ha considerado proporcional a la carga.

Cabe comentar en este punto que, la definición de los parámetros del sistema en el modelo, es de la forma: . De esta forma, se optimiza solo el factor proporcional, consiguiendo así que los valores se muevan en unos órdenes de magnitud inferiores, característica que influye positivamente en el proceso de optimización.

Véase a continuación el ajuste entre el par medido en el g.d.l. galga y el proporcionado por el modelo:



1. Representación del par total ofrecido por el sistema, medido (azul) y generado por el modelo (negro).

Los errores porcentuales del ajuste entre la potencia medida y la generada por el modelo, según la carga, son:



## 3.4.7 Estudio de sensibilidad del modelo

Una vez que se ha encontrado el paquete de factores optimizados que consiguen que el modelo represente el comportamiento real del sistema con un grado de ajuste dentro de los límites establecidos, se procede a realizar un estudio de sensibilidad del mismo.

El objetivo del estudio de sensibilidad es poner de manifiesto la capacidad que tiene el modelo desarrollado ante cambios en los parámetros físicos del mismo. Es decir, ver si cuando, por ejemplo, alguna muñequilla presenta una rigidez inferior a la considerada como normal, el modelo responde ofreciendo valores desviados del comportamiento normal y en qué magnitud.

La forma de proceder en este caso es:

* Aplicar pequeñas amplificaciones/reducciones a los parámetros.
* Analizar el comportamiento del par total ofrecido por el modelo, en términos de:
* Armónico cero, ya que es este el coeficiente que controla el valor medio de la potencia, en comparación con el mismo coeficiente cuando no se ha aplicado amplificación/reducción sobre el parámetro.
* Error cuadrático medio del ajuste de la curva de par generada tras la amplificación/reducción del parámetro respecto a la misma curva generada con los parámetros sin modificar.

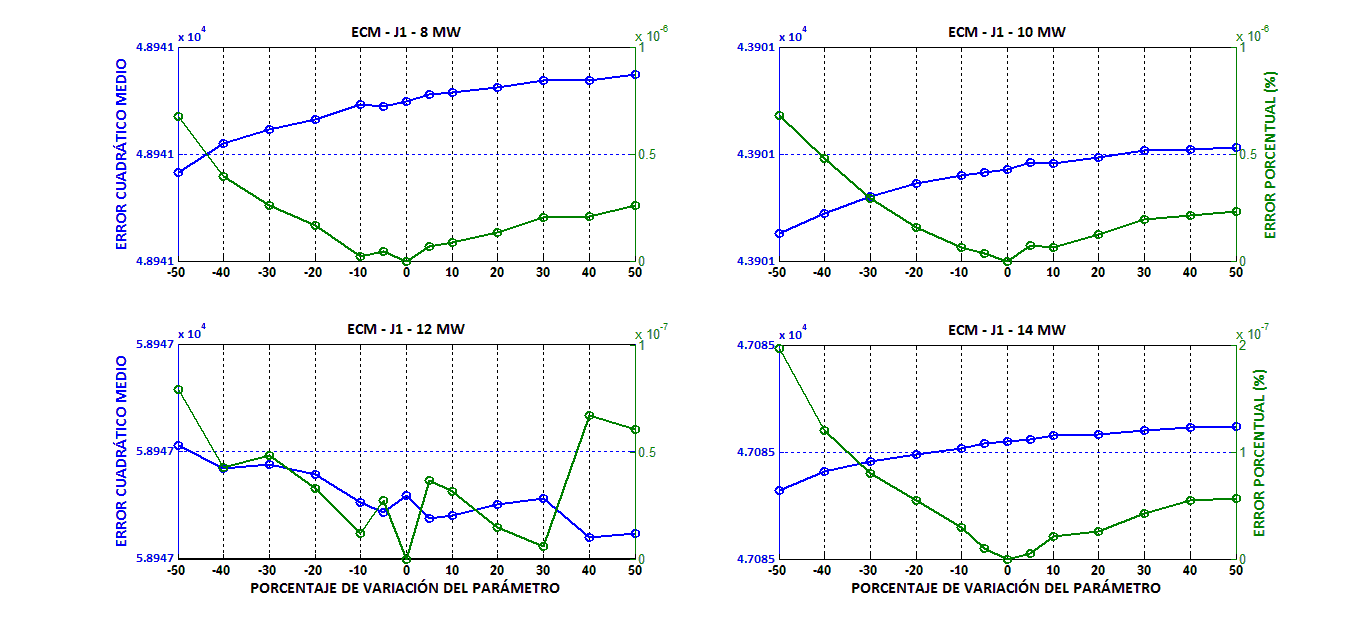
Se considera la variación de sólo un parámetro manteniendo el resto en su valor original, no analizándose la casuística de variaciones de varios parámetros de forma simultánea, dada la gran casuística que ello representa.

Se observa, tras el estudio, que los parámetros que más influencia tienen sobre la dinámica del sistema y por tanto hacen que el modelo se comporte de forma más sensible ante variaciones sobre los mismos, son:

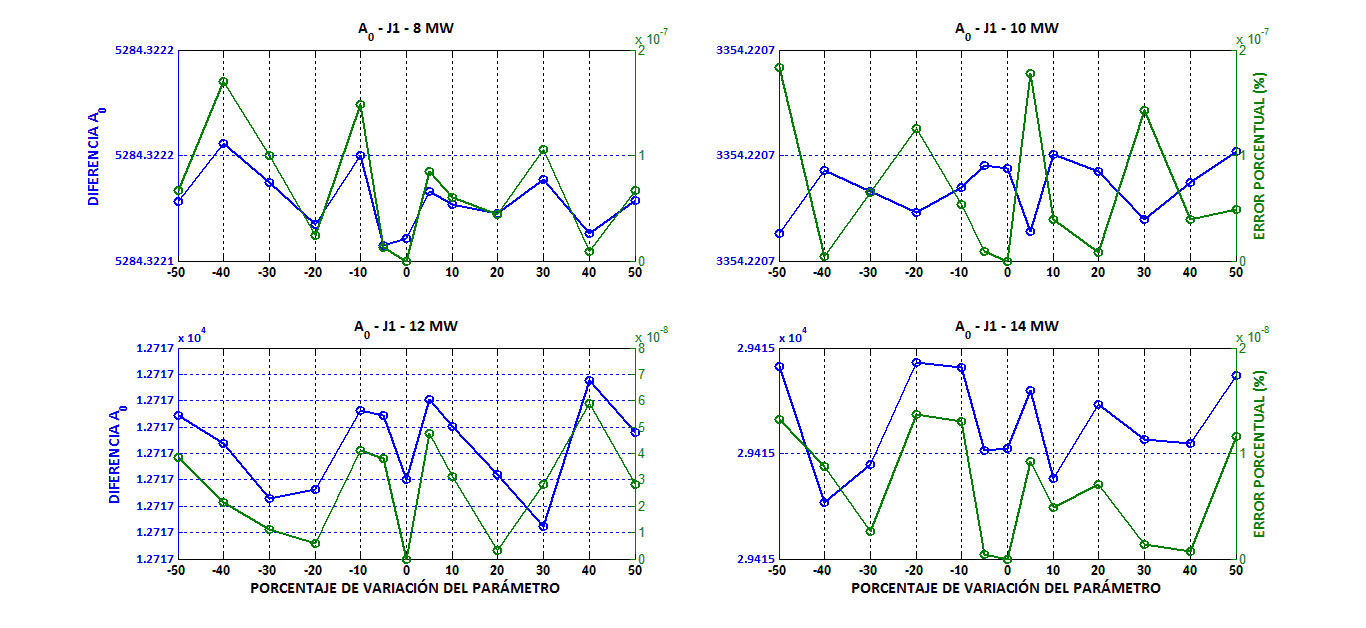
* ***Factor corrector del PMS.***
* ***Rigideces de las muñequillas.***
* ***Rigidez del tramo galga.***

### Inercia del amortiguador de vibraciones torsionales (J1):

El modelo no es muy sensible ante cambios en este parámetro, puesto que dicho parámetro no tiene una alta influencia en la dinámica del sistema. Véase en la siguiente representación que, aunque se aprecien variaciones en las curvas, los valores de los ejes son muy próximos, lo que indica que la variación es prácticamente nula:



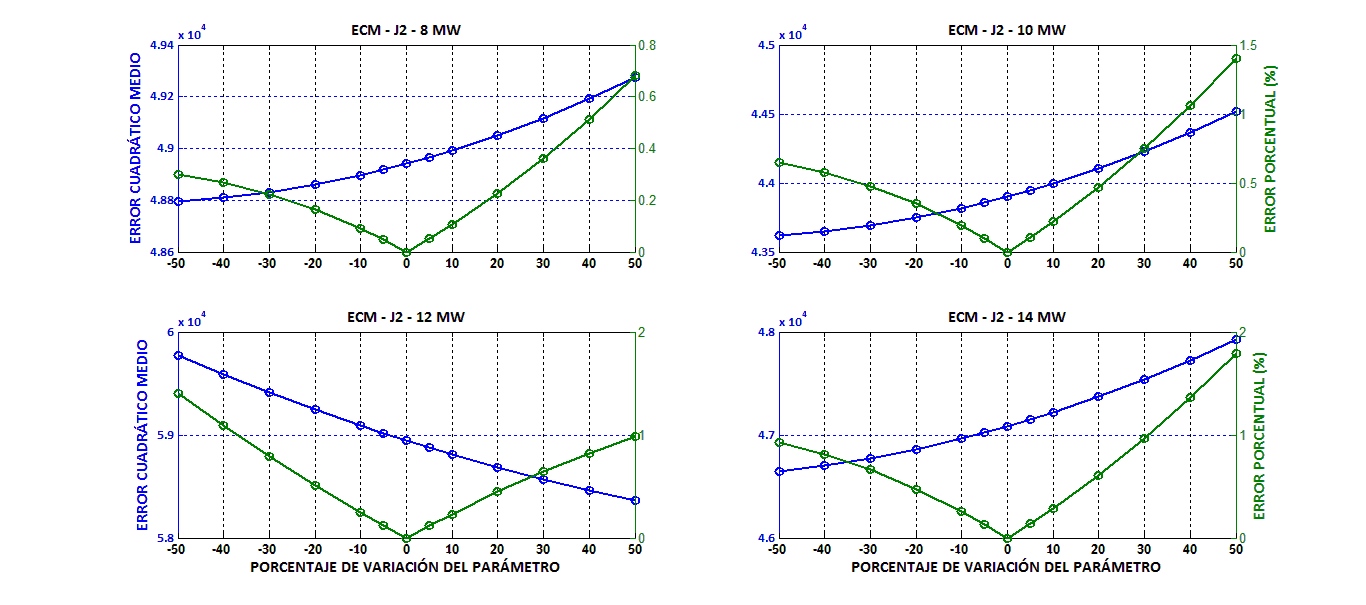
1. Sensibilidad del modelo ante cambios en J1, en términos del ECM. En verde, en términos porcentuales.



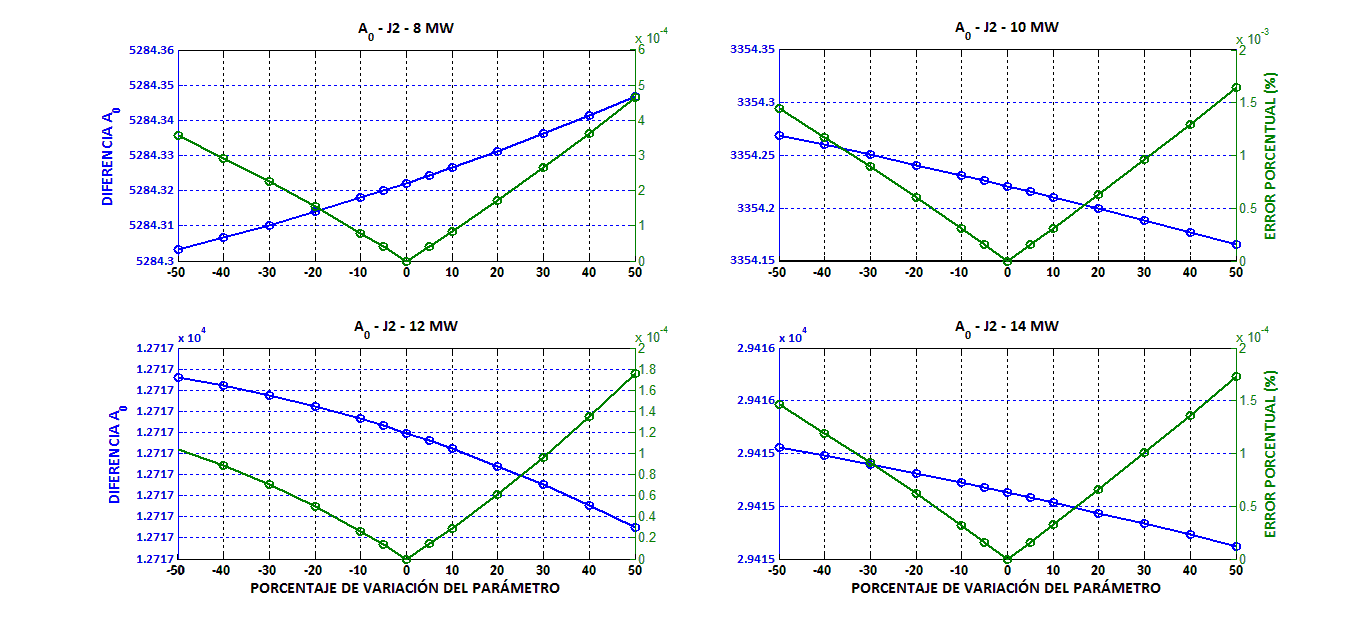
1. Sensibilidad del modelo ante cambios en J1, en términos de Ao. En verde, en términos porcentuales.

### Inercia del tramo entre el amortiguador torsional y la primera muñequilla (J2):

Este parámetro ya empieza a tener mayor influencia en la dinámica del sistema real y por ello, se aprecia una mayor sensibilidad del modelo ante cambios en dicho parámetro. Sin embargo, la mayor desviación apenas llega a un 2% en términos de error cuadrático medio:



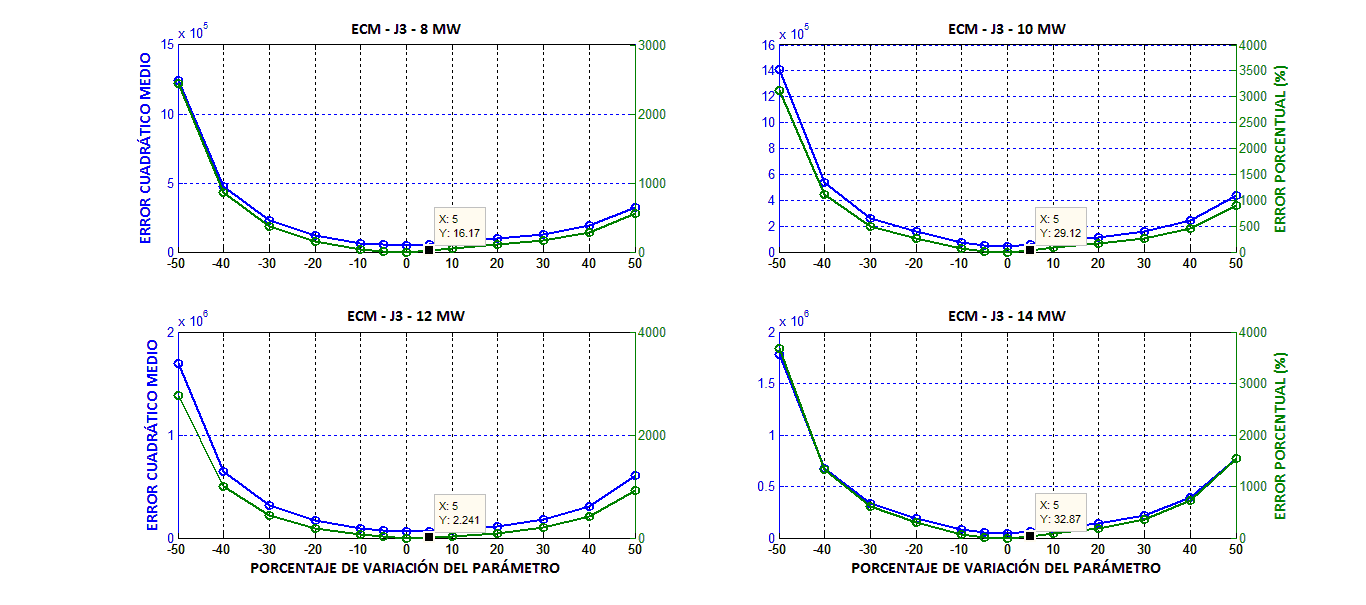
1. Sensibilidad del modelo ante cambios en J2, en términos de ECM. En verde, en términos porcentuales.



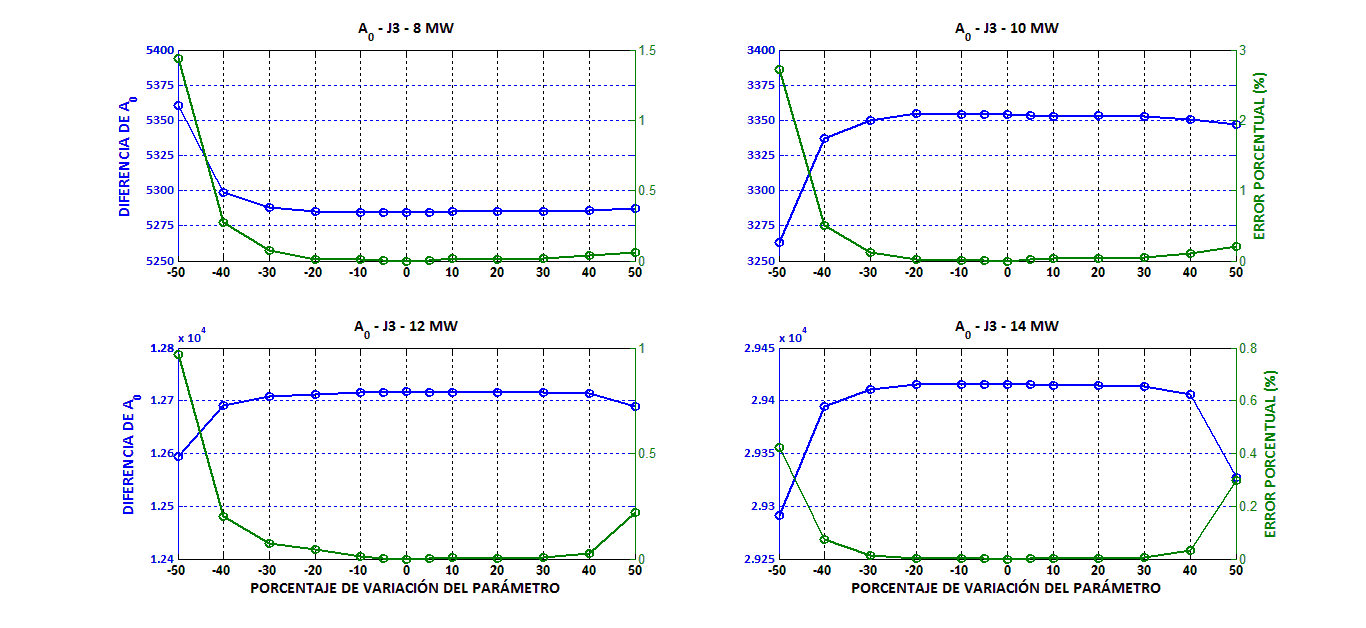
1. Sensibilidad del modelo ante cambios en J2, en términos de Ao. En verde, en términos porcentuales.

### Inercia de las muñequillas (J3):

Este parámetro si tiene bastante influencia en la dinámica del sistema, por lo que si se aprecia una alta sensibilidad del modelo ante variaciones del mismo. Sin embargo, cabe remarcar que dicha sensibilidad se aprecia en términos del ECM, ya que es la forma de la curva de par la que cambia, sin embargo en términos de potencia total no afecta tanto:



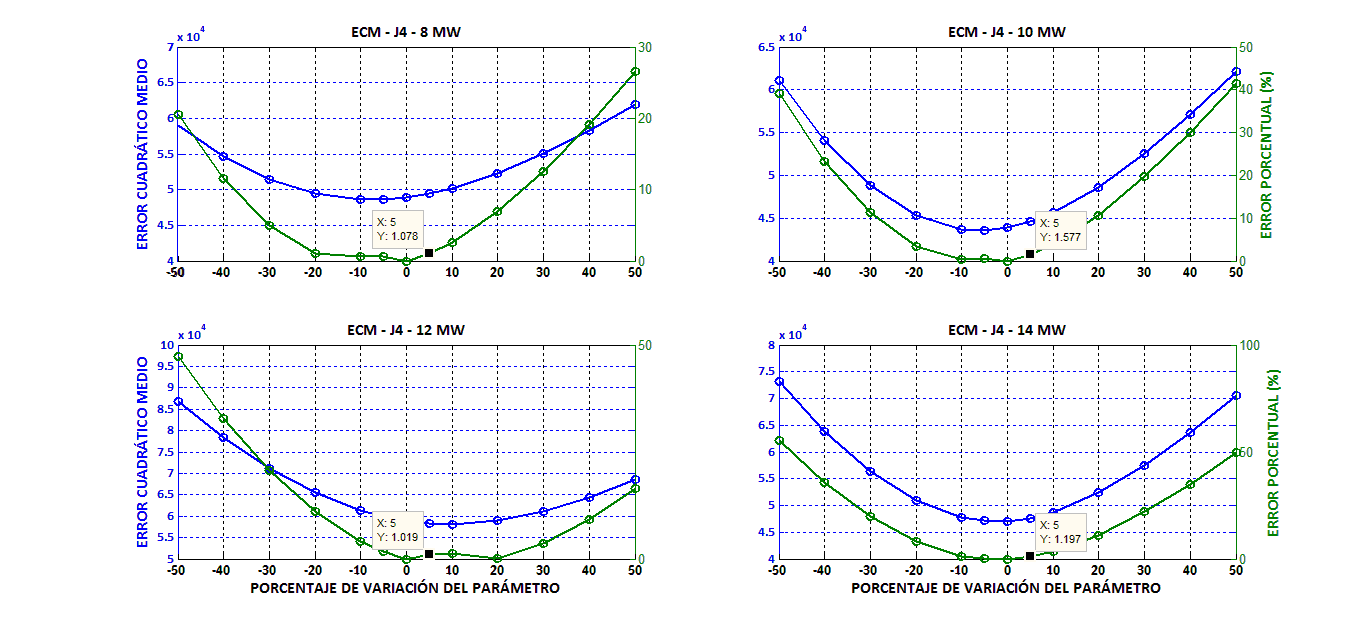
1. Sensibilidad del modelo ante cambios en J3, en términos de ECM. En verde, en términos porcentuales.



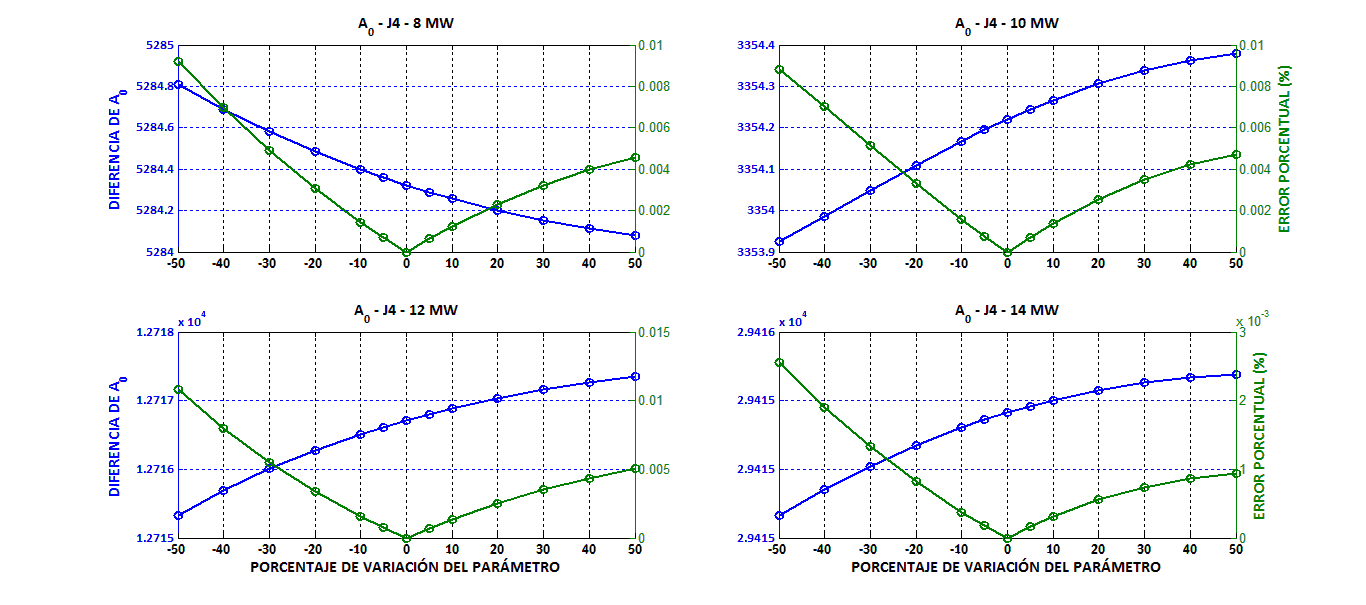
1. Sensibilidad del modelo ante cambios en J3, en términos de Ao. En verde, en términos porcentuales.

### Inercia del sistema de distribución (J4):

La influencia de este parámetro sobre la dinámica del sistema es similar al caso de la inercia de las muñequillas, tal y como se aprecia en las siguientes representaciones:



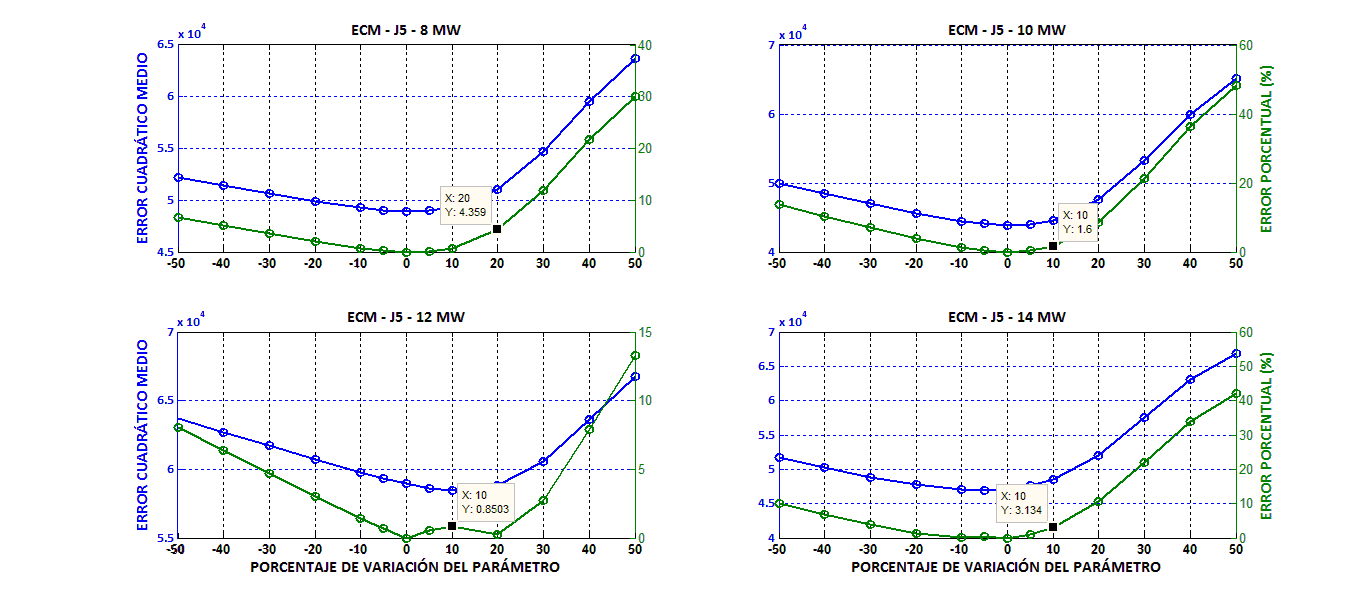
1. Sensibilidad del modelo ante cambios en J4, en términos de ECM. En verde, en términos porcentuales.



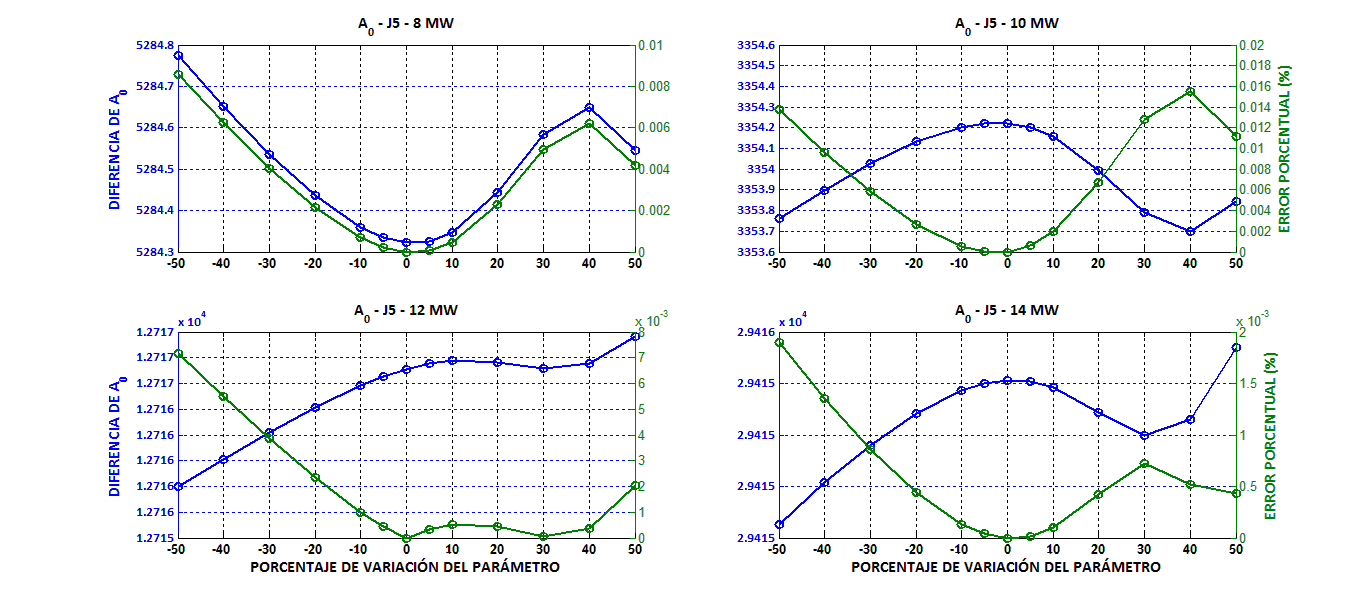
1. Sensibilidad del modelo ante cambios en J4, en términos de Ao. En verde, en términos porcentuales.

### Inercia del tramo galga (J5):

Este tramo, al presentar un valor de inercia elevado, también tiene una influencia importante en la dinámica del sistema.



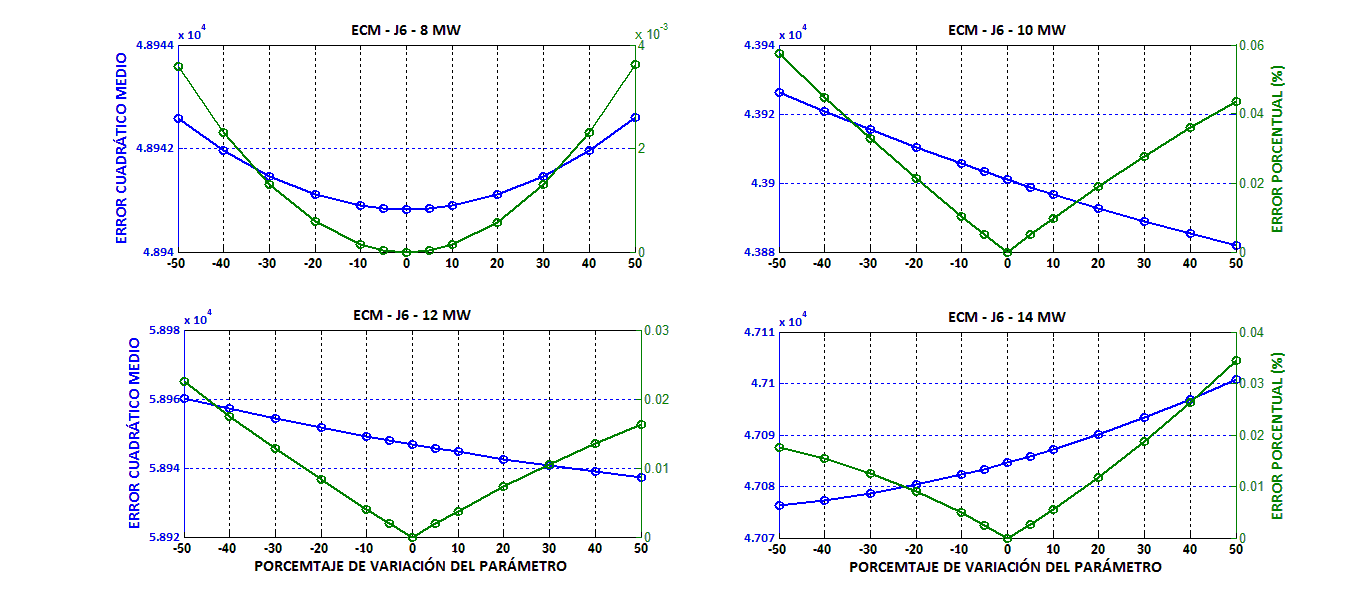
1. Sensibilidad del modelo ante cambios en J5, en términos de ECM. En verde, en términos porcentuales.



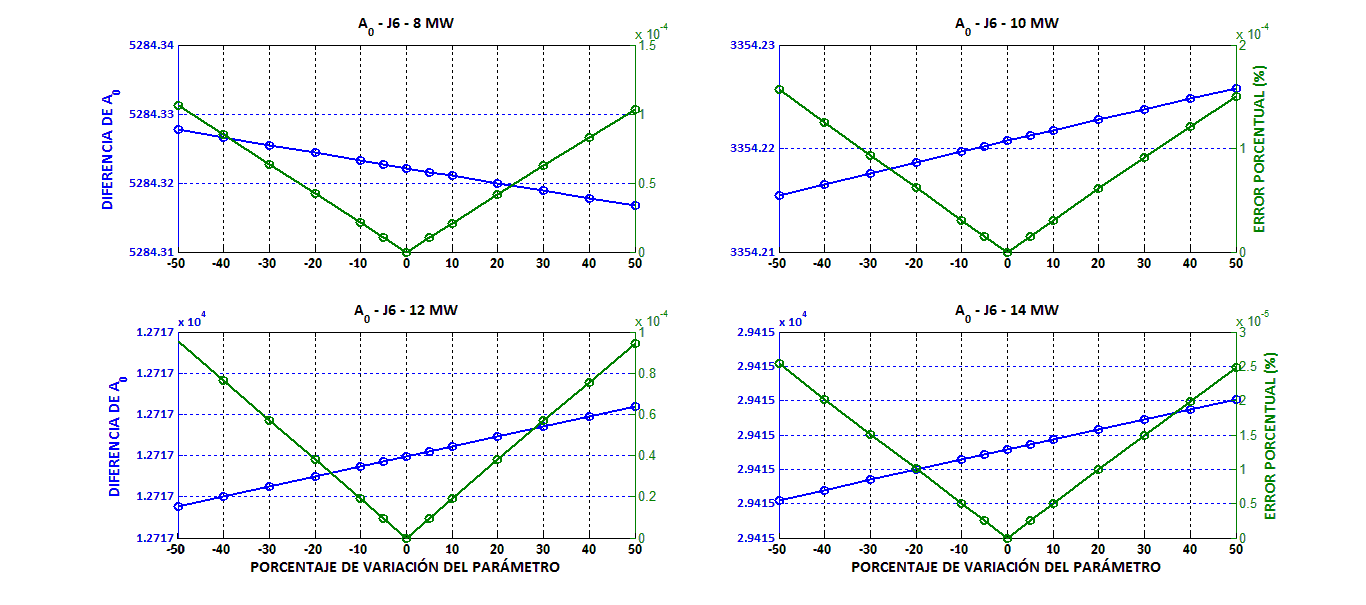
1. Sensibilidad del modelo ante cambios en J5, en términos de Ao. En verde, en términos porcentuales.

### Inercia alternador (J6):

Este parámetro no presenta una influencia clara en la sensibilidad del modelo, tal y como se aprecia en las representaciones mostradas a continuación.



1. Sensibilidad del modelo ante cambios en J6, en términos de ECM. En verde, en términos porcentuales.

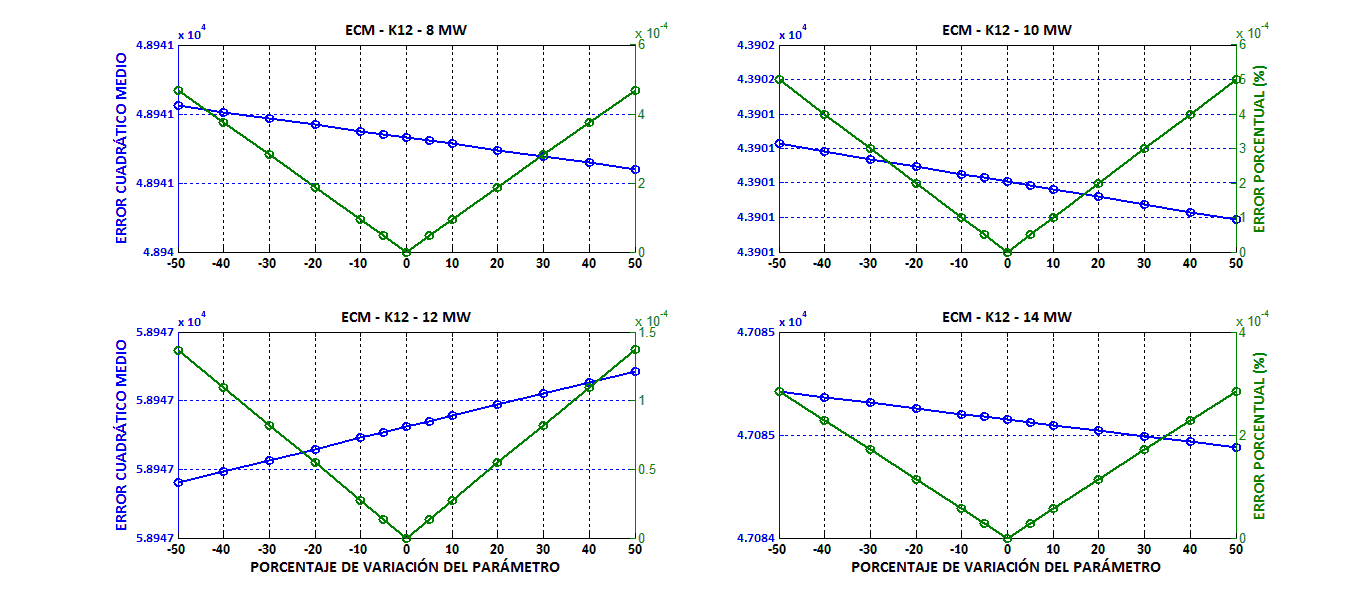


1. Sensibilidad del modelo ante cambios en J6, en términos de Ao. En verde, en términos porcentuales.

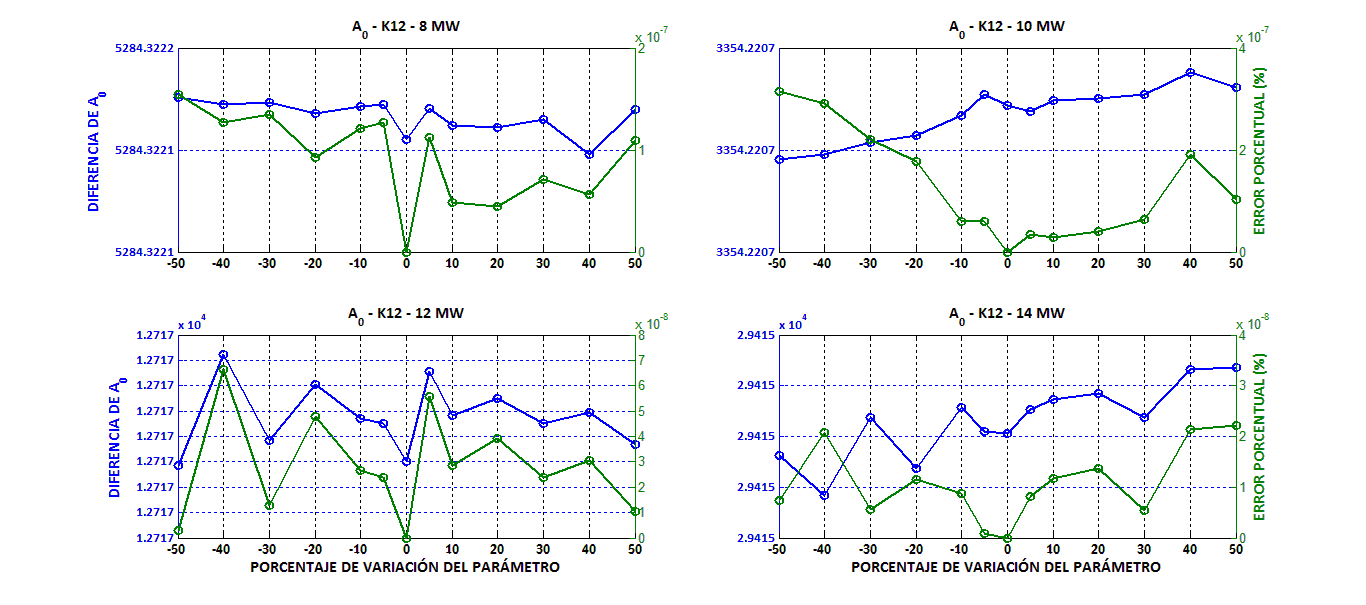
A continuación, el estudio de sensibilidad del modelo continúa con los parámetros de rigidez, que son los que realmente ofrecen una potencial utilidad de cara al mantenimiento predictivo.

### Rigidez del tramo amortiguador – 1ª muñequilla (K12):

Este parámetro no afecta prácticamente a la dinámica del sistema, tal y como se aprecia:



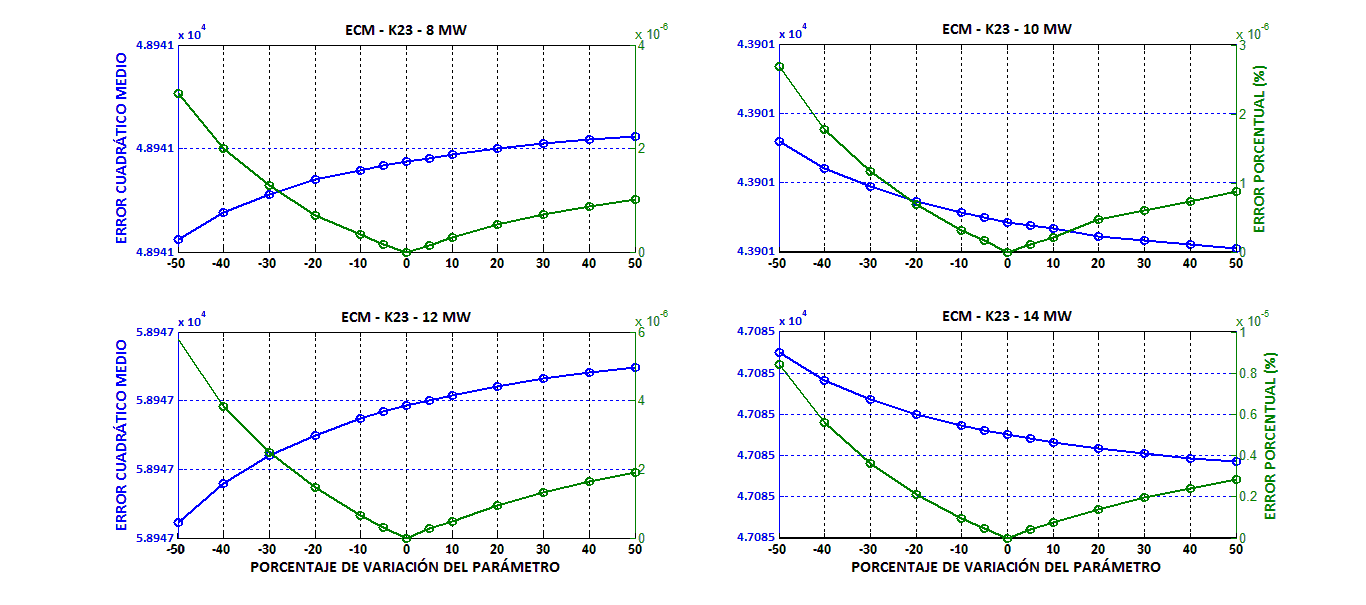
1. Sensibilidad del modelo ante cambios en K12 en términos de ECM. En verde, en términos porcentuales.



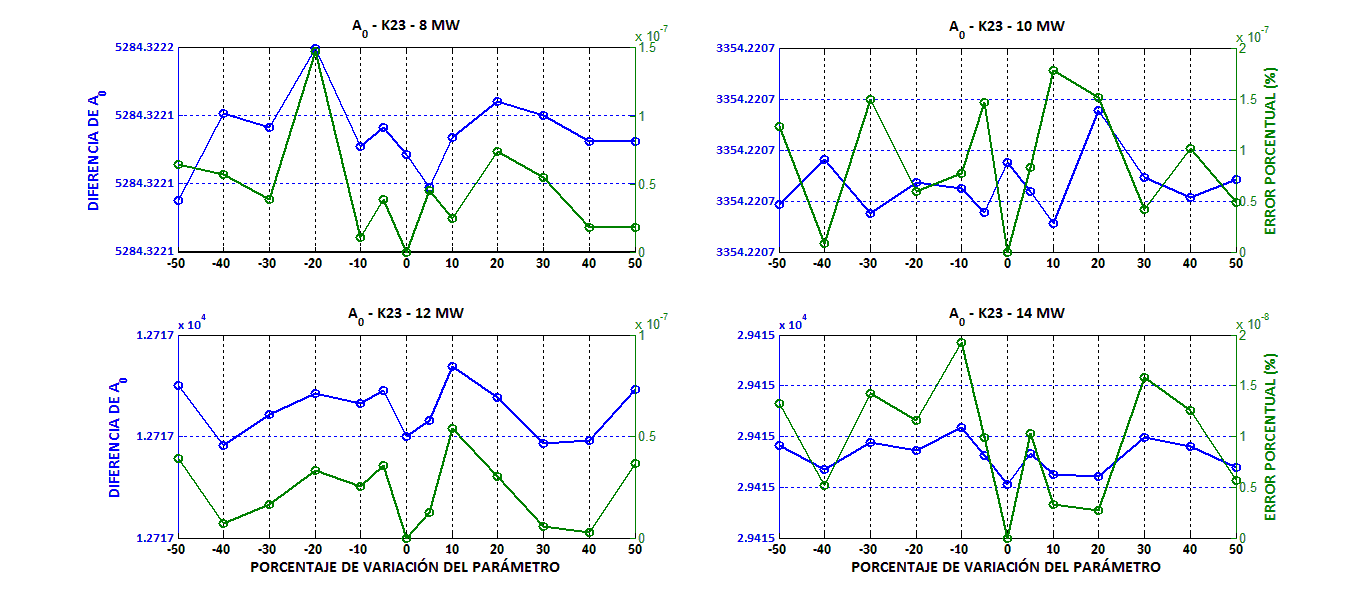
1. Sensibilidad del modelo ante cambios en K12 en términos de Ao. En verde, en términos porcentuales.

### Rigidez de la 1ª muñequilla (K23):

Este parámetro no muestra una influencia elevada en la sensibilidad del modelo.



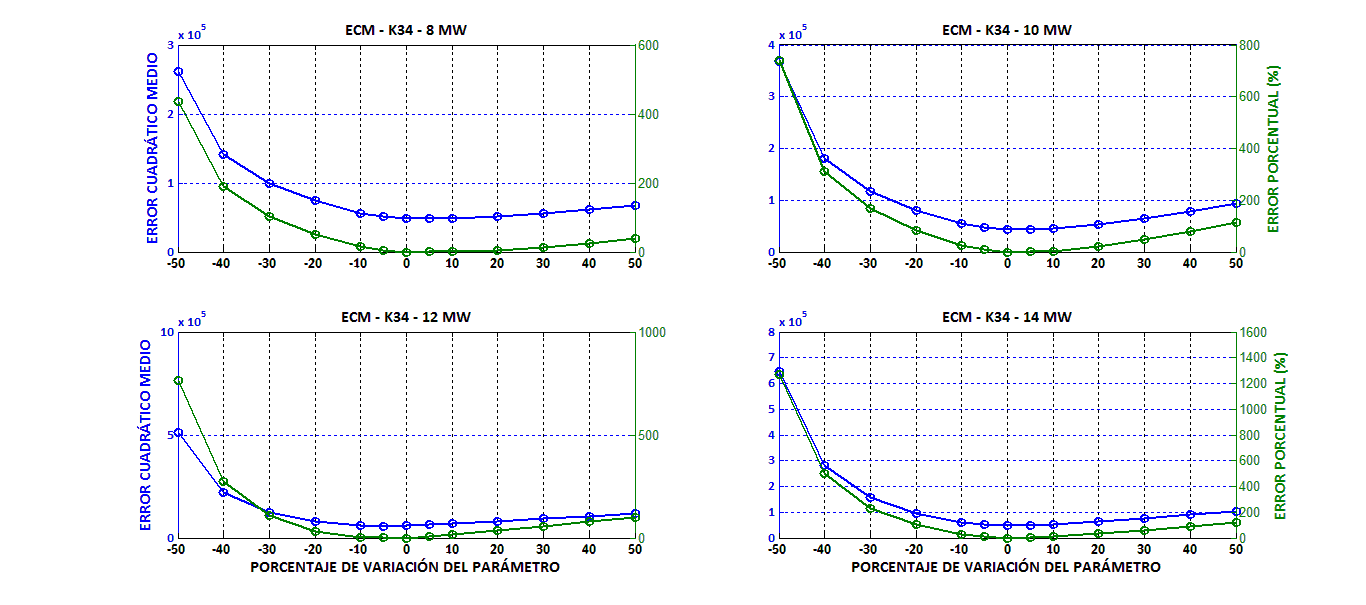
1. Sensibilidad del modelo ante cambios en K23 en términos de ECM. En verde, en términos porcentuales.



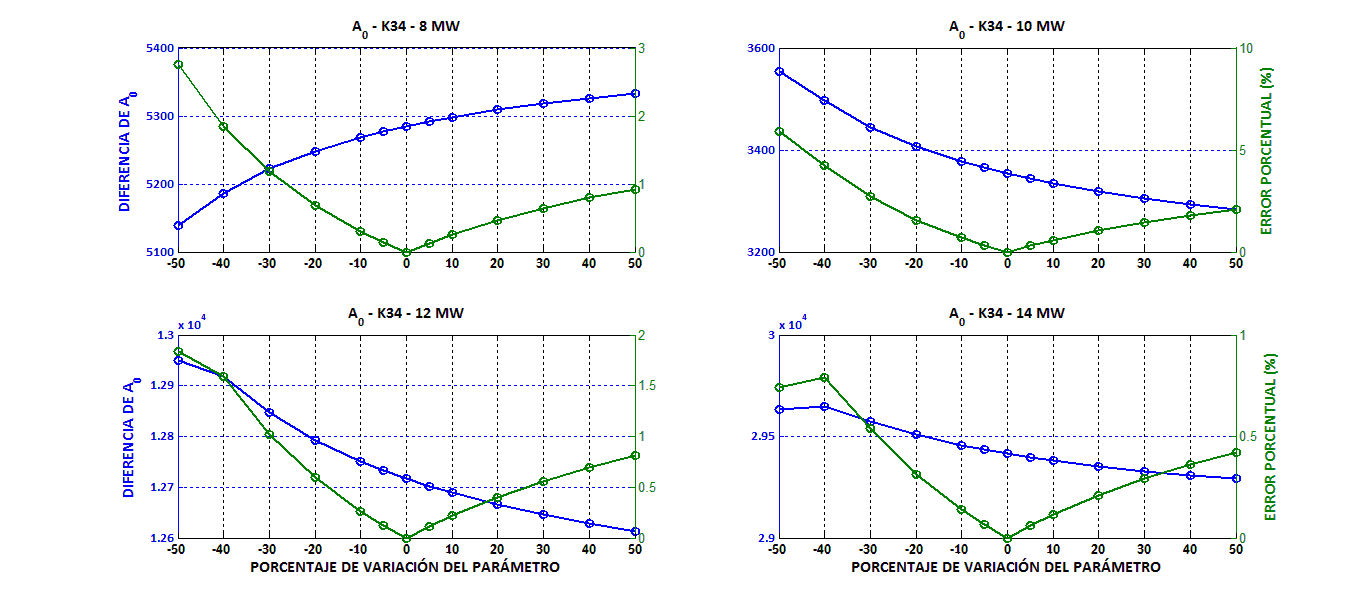
1. Sensibilidad del modelo ante cambios en K23 en términos de Ao. En verde, en términos porcentuales.

### Rigidez de las muñequillas (K34):

Este parámetro ya si es importante de cara a la sensibilidad del modelo, como era de esperar, ya que afecta directamente sobre la dinámica del sistema.



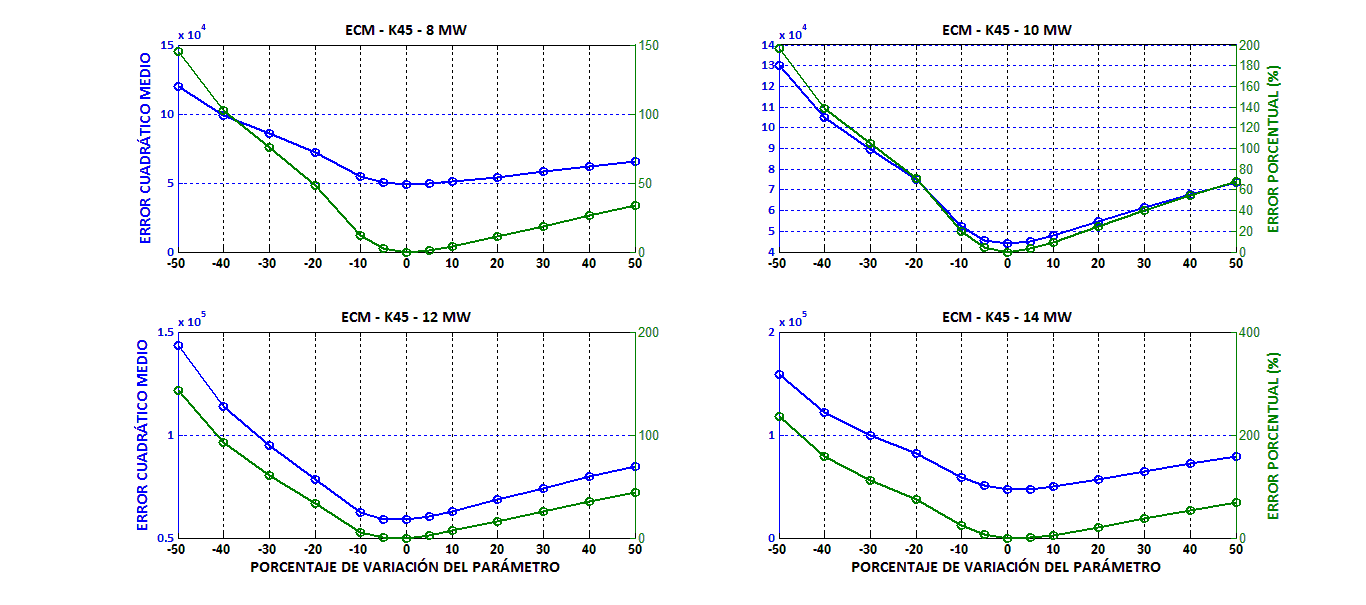
1. Sensibilidad del modelo ante cambios en K34 en términos de ECM. En verde, en términos porcentuales.



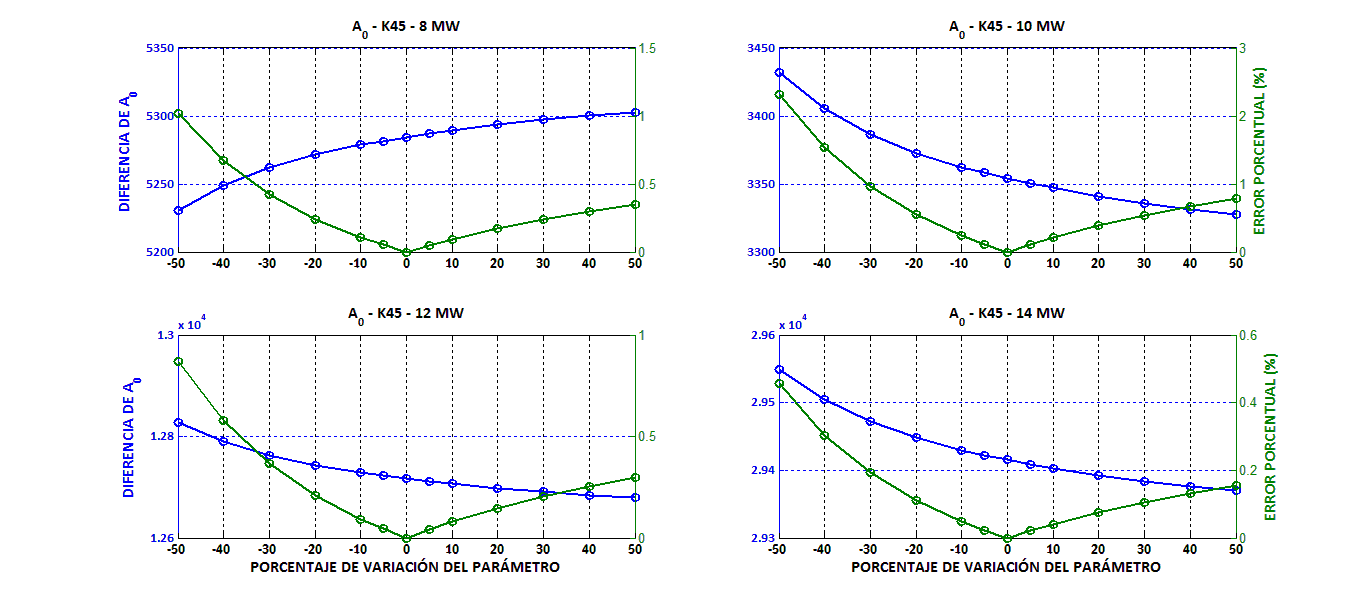
1. Sensibilidad del modelo ante cambios en K34 en términos de Ao. En verde, en términos porcentuales.

### Rigidez del tramo de distribución (K45):

Este tramo de eje, al igual que ocurre con las muñequillas del cigüeñal, tiene una influencia notable en la dinámica del sistema y por ello, el modelo es bastante sensible a cambios en su valor.



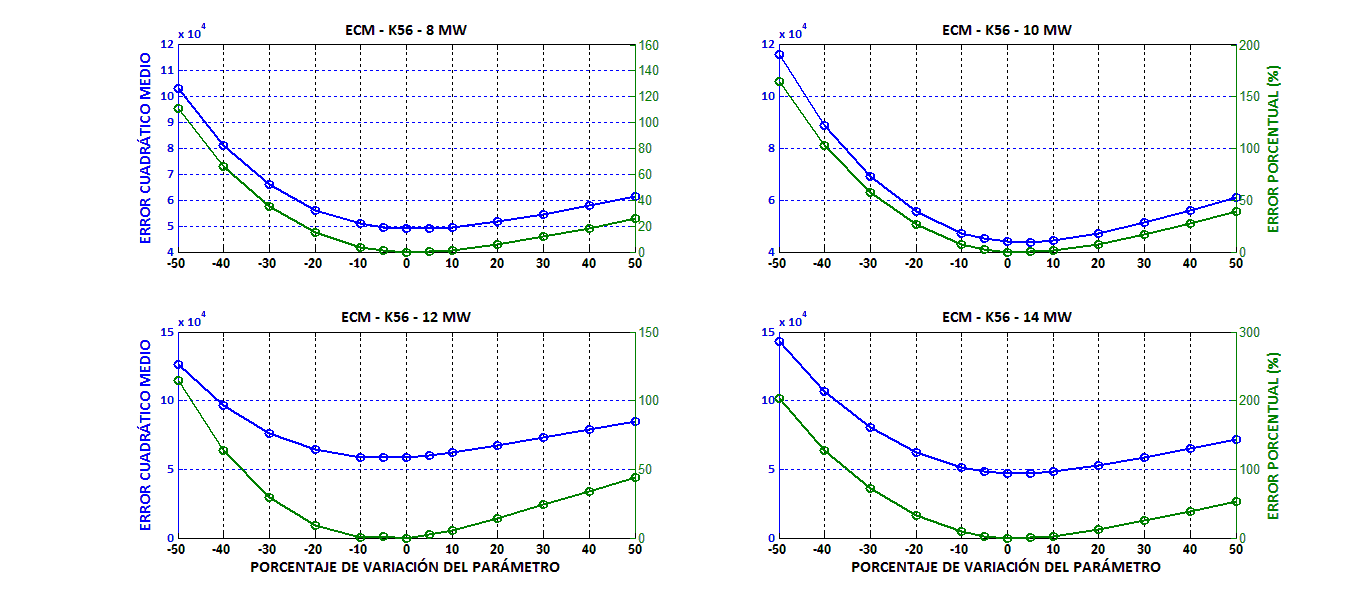
1. Sensibilidad del modelo ante cambios en K45 en términos de ECM. En verde, en términos porcentuales.



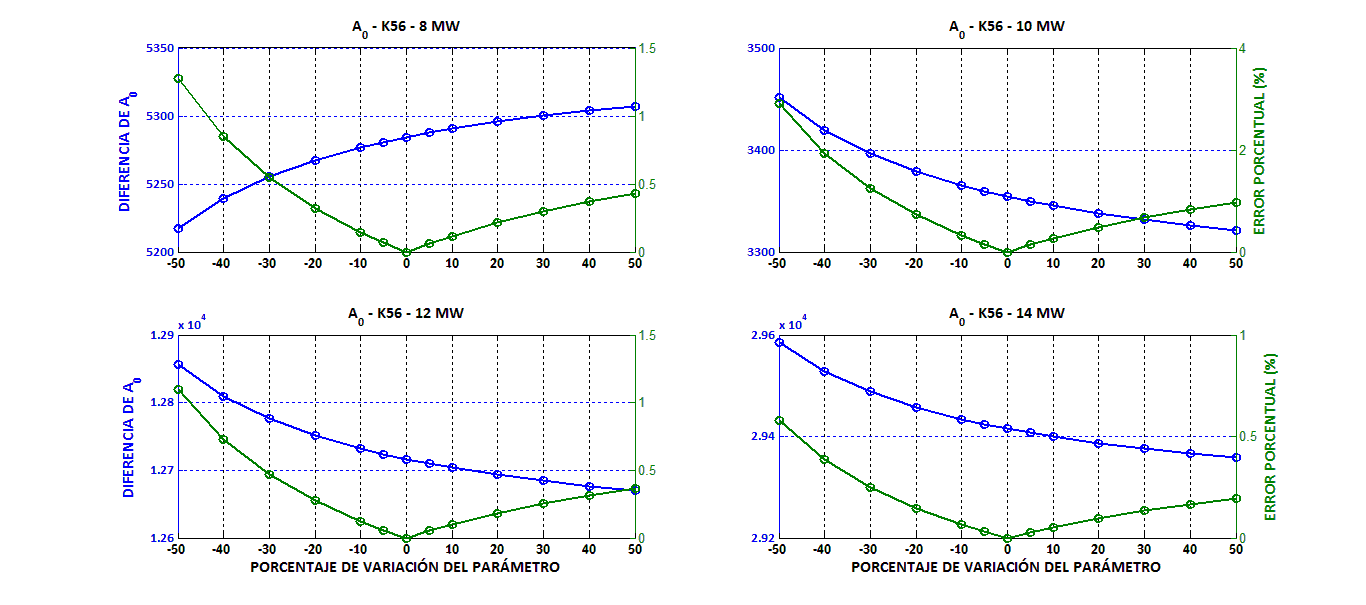
1. Sensibilidad del modelo ante cambios en K45 en términos de Ao. En verde, en términos porcentuales.

### Rigidez de la última muñequilla (K56):

La experiencia en el campo de dinámica de motores a combustión interna alternativos pone de manifiesto que la principal zona de aparición de grietas en un motor es la última muñequilla del cigüeñal, ya que dicha zona “ve” todo el par motor:



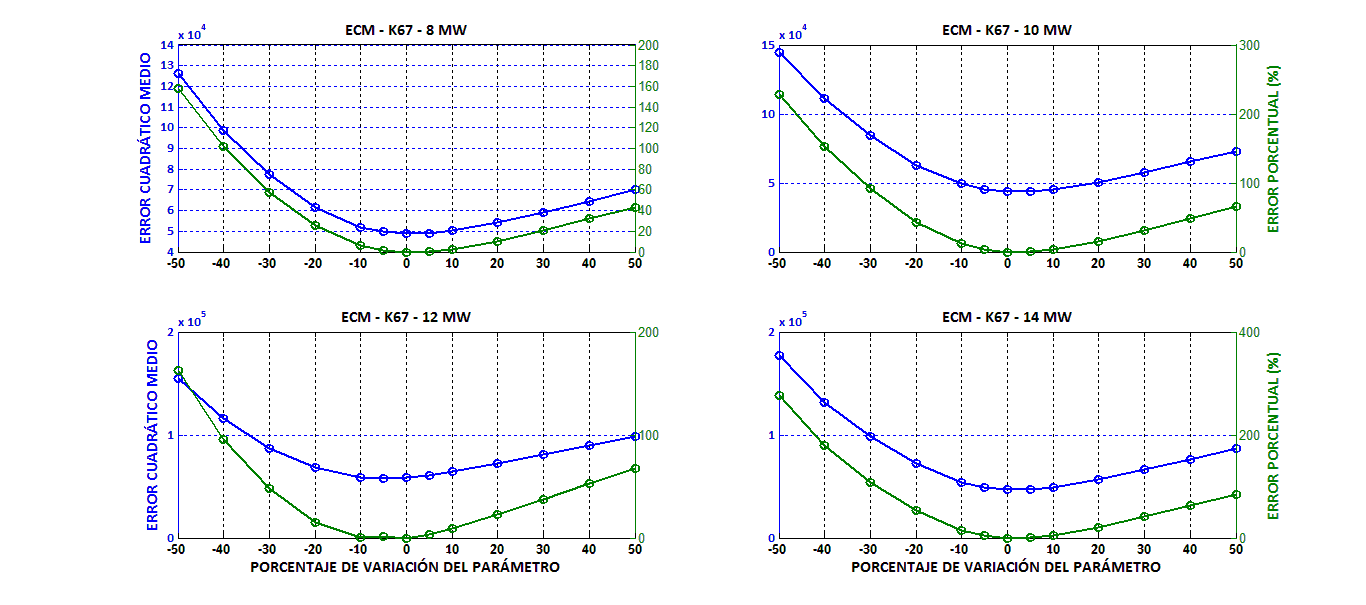
1. Sensibilidad del modelo ante cambios en K56 en términos de ECM. En verde, en términos porcentuales.



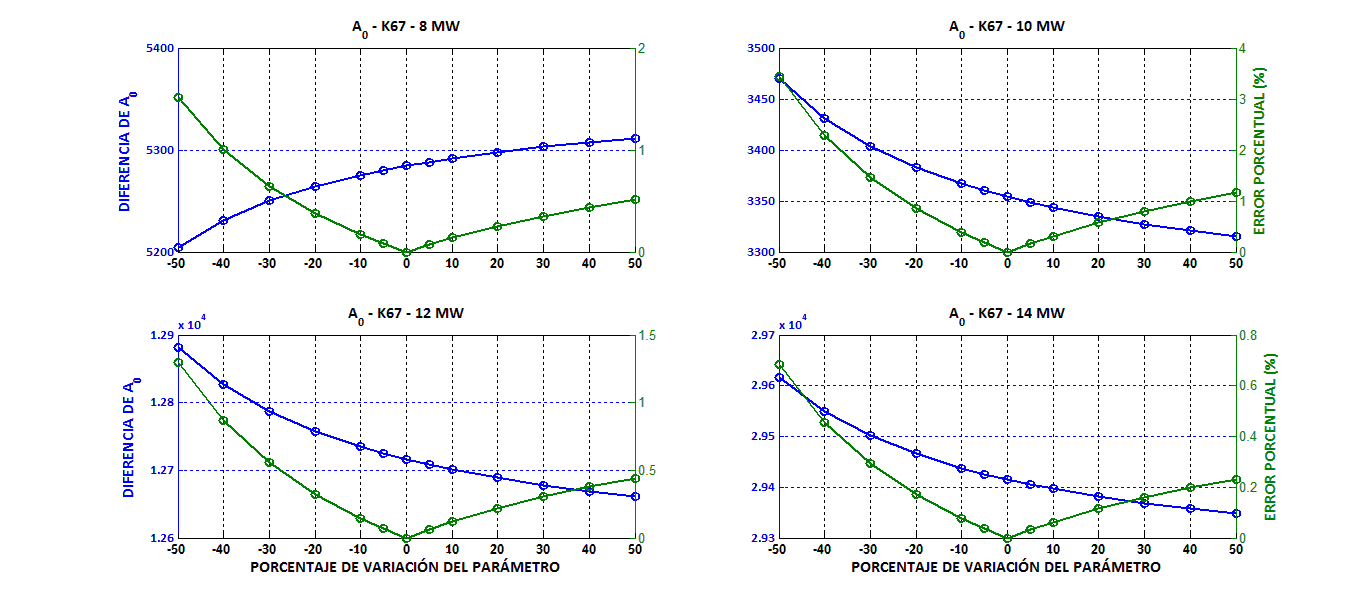
1. Sensibilidad del modelo ante cambios en K56 en términos de Ao. En verde, en términos porcentuales.

### Rigidez del tramo última muñequilla – galga (K67):

Este tramo también es muy crítico puesto que es donde se comienza a concentrar los torsores más altos.



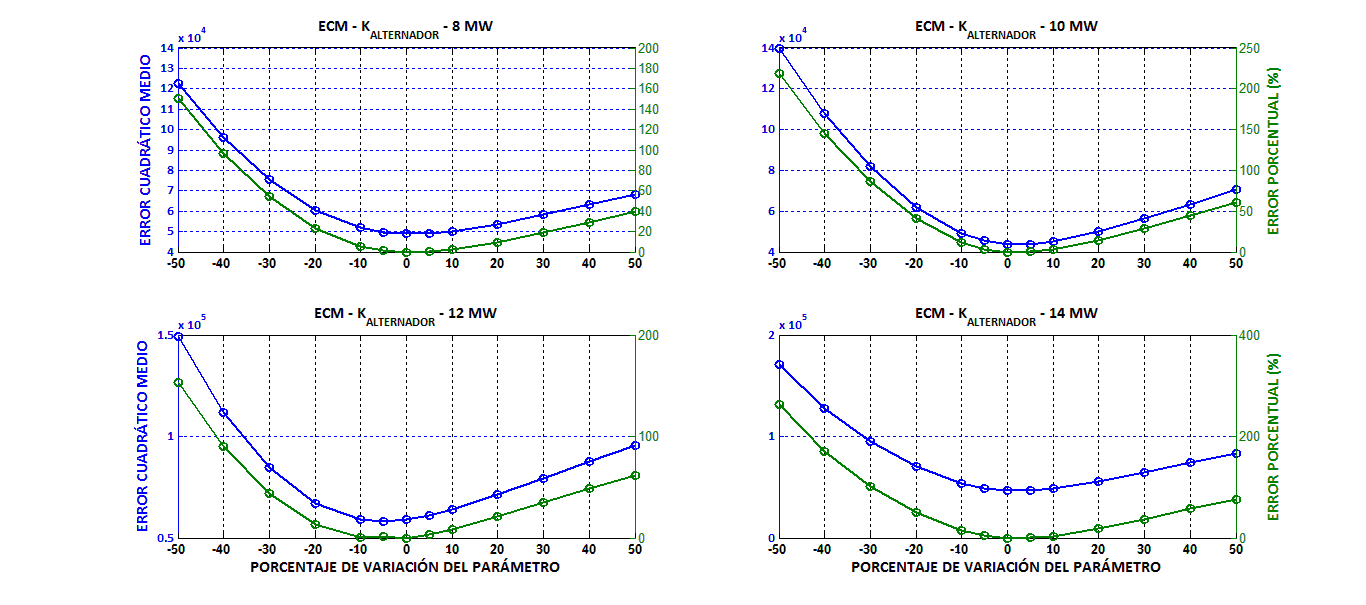
1. Sensibilidad del modelo ante cambios en K67 en términos de ECM. En verde, en términos porcentuales.



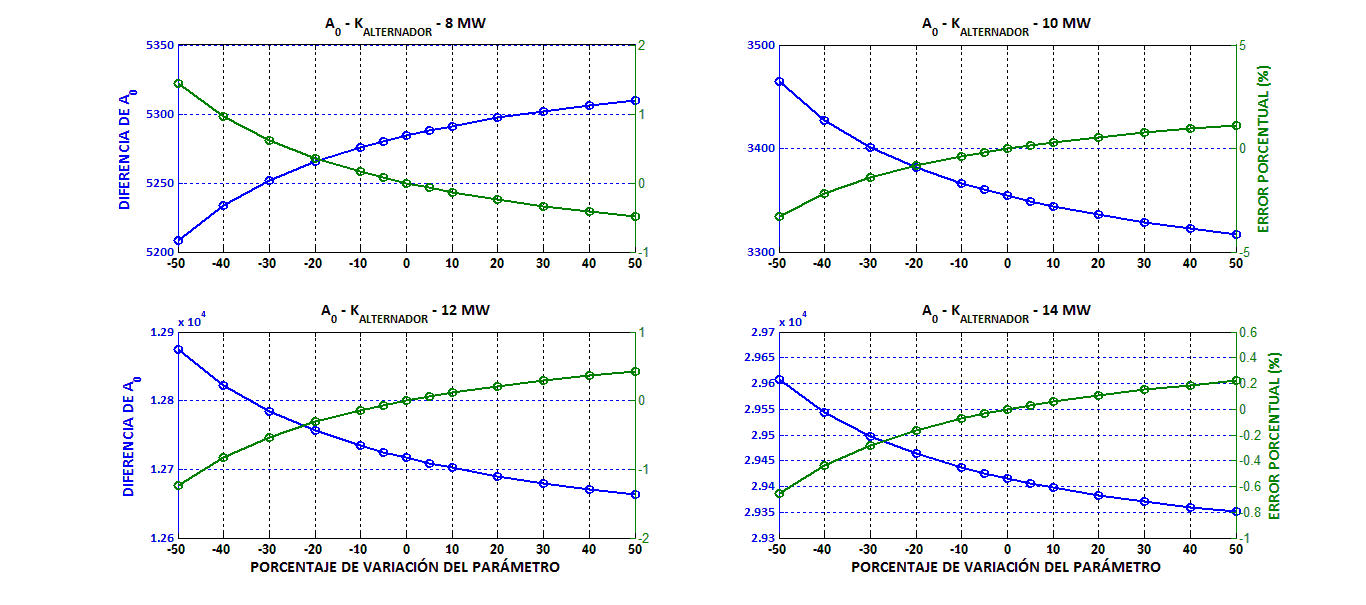
1. Sensibilidad del modelo ante cambios en K67 en términos de Ao. En verde, en términos porcentuales.

### Rigidez del tramo galga – alternador (Kalternador):

En este tramo, igual que en el anterior, se concentran torsores muy altos y por ello, influye de forma crítica en la dinámica del sistema.



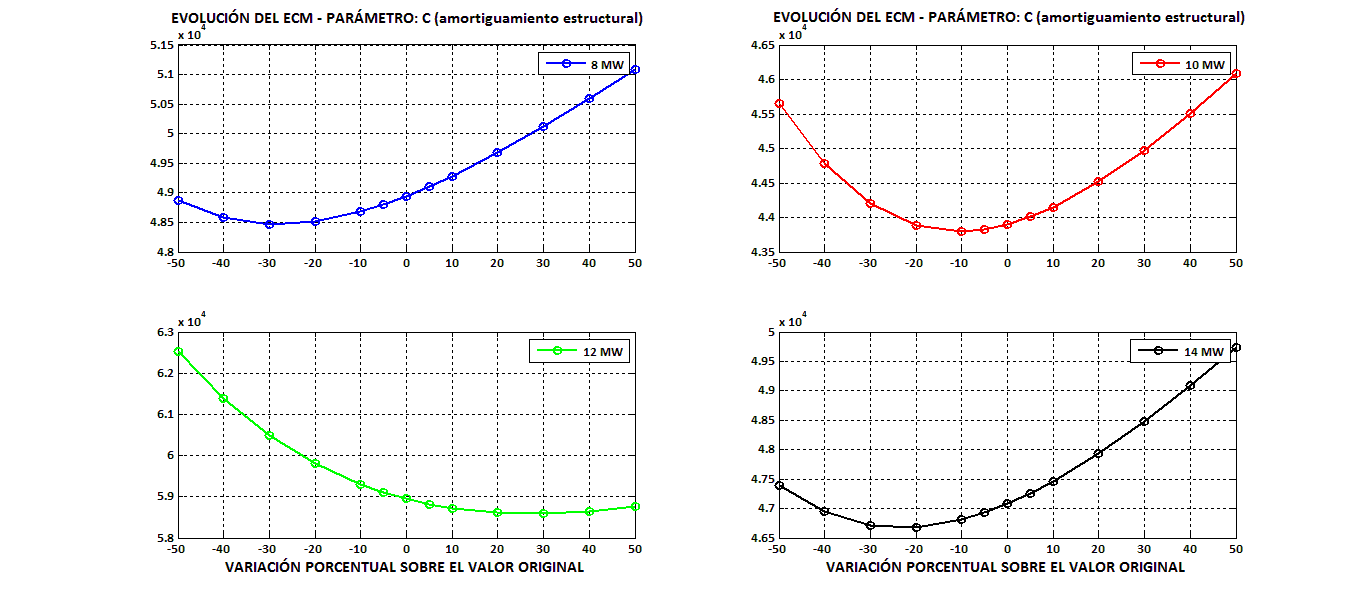
1. Sensibilidad del modelo ante cambios en Kalternador en términos de ECM. En verde, en términos porcentuales.



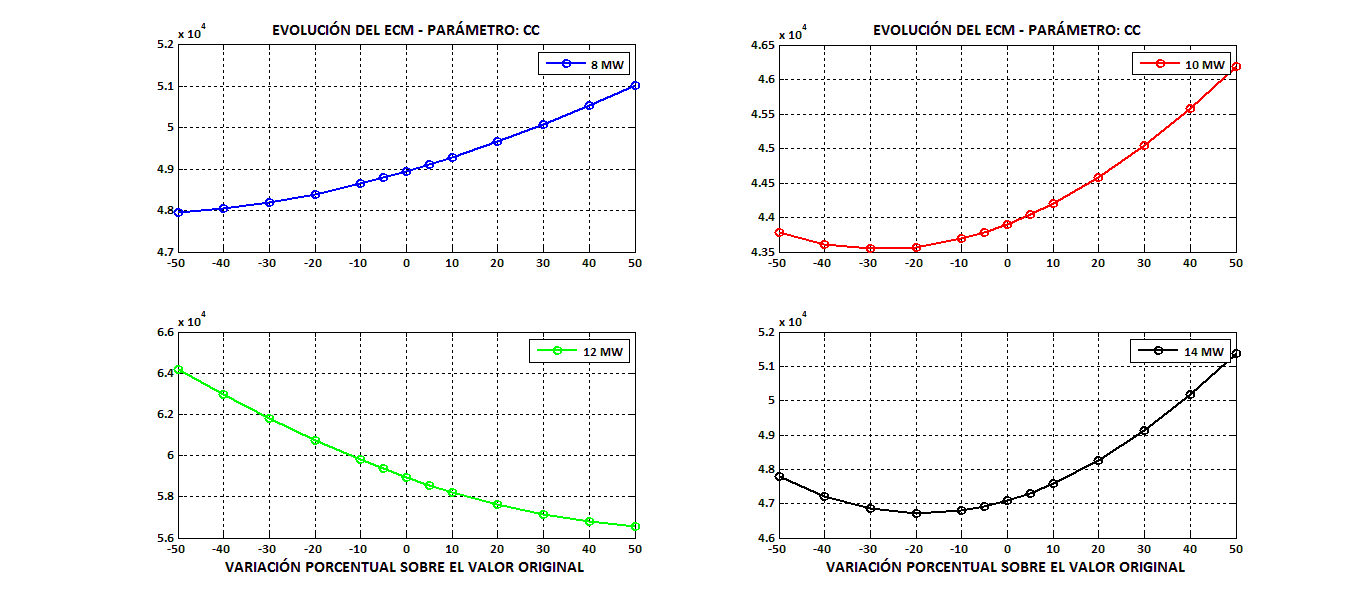
1. Sensibilidad del modelo ante cambios en Kalternador en términos de Ao. En verde, en términos porcentuales.

### Parámetros de amortiguamiento (estructural, a tierra y torsional):

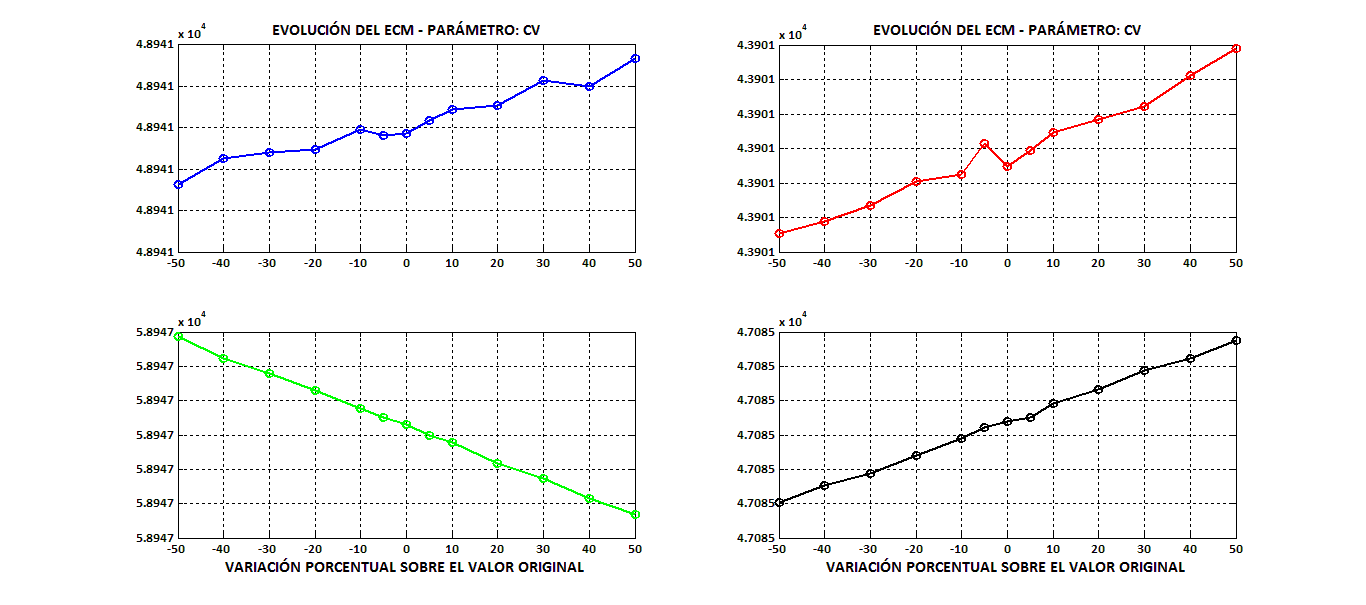
En las figuras mostradas a continuación se puede observar claramente que los parámetros de amortiguamiento apenas tienen influencia sobre la dinámica del sistema, haciendo por tanto que la sensibilidad del modelo sea moderada.



1. Sensibilidad del modelo ante cambios en C en términos de ECM.



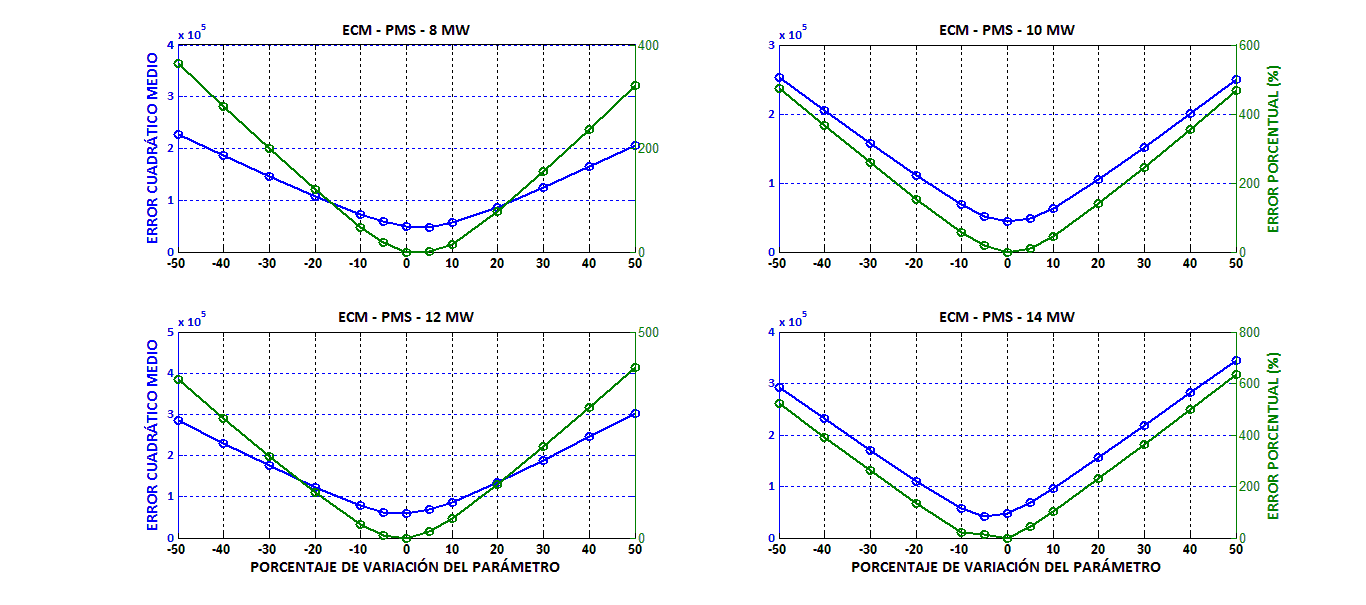
1. Sensibilidad del modelo ante cambios en CC en términos de ECM.



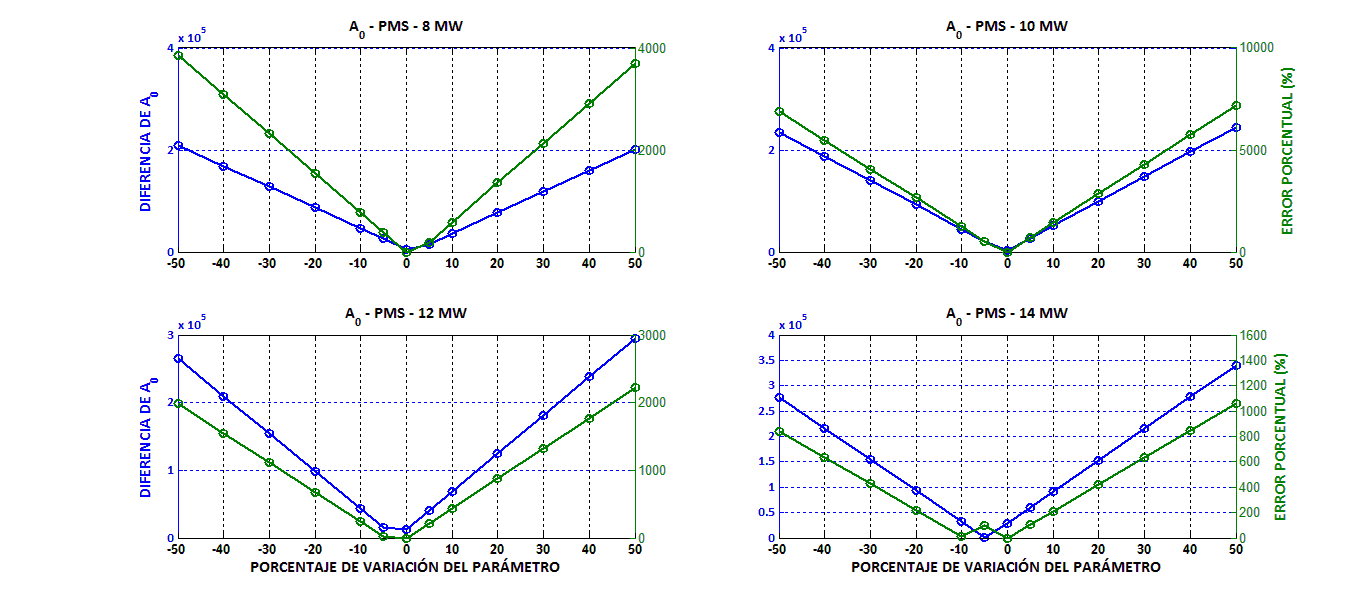
1. Sensibilidad del modelo ante cambios en CV en términos de ECM.

### Parámetro corrector del PMS:

Tal y como se comentó cuando se puso de manifiesto la definición de este parámetro, la influencia sobre la dinámica del sistema real es de carácter muy importante. Se puede apreciar en las figuras que, para variaciones muy pequeñas sobre el valor nominal del parámetro, el sistema responde con un cambio muy elevado.



1. Sensibilidad del modelo ante cambios en PMS en términos de ECM. En verde, en términos porcentuales.



1. Sensibilidad del modelo ante cambios en PMS en términos de Ao. En verde, en términos porcentuales.

## 3.4.8 Valoración del modelo obtenido

Tras los procesos de definición, ajuste, optimización y análisis del modelo aquí expuesto, se llega a un punto en el que se dispone de un modelo analítico que representa el comportamiento del sistema real al que modela, con un error máximo en términos energéticos del 2.66% (que suponen 0.3MW) y que tiene una sensibilidad muy alta ante cambios en parámetros físicos como son la rigidez e inercia de sus elementos. Por tanto, a la vista de los resultados obtenidos, ***el modelo queda validado***.

El hecho de que el modelo sea sensible ante variaciones en la rigidez de ciertas partes del mismo, hace que disponer de este modelo sea muy interesante de cara al mantenimiento predictivo del sistema real, puesto que resolviendo el problema de forma inversa, se pueden obtener los valores de los parámetros y compararlos con los nominales ya obtenidos y valorar así el estado del sistema real.

El interés que presenta el hecho de hacer una valoración del sistema real mediante un simple análisis de los valores de los parámetros físicos que se obtengan radica en la anticipación ante posibles situaciones de fallo del sistema que originarían paradas del mismo y por tanto la no generación de energía eléctrica.

# Identificación de Fallos y Mantenimiento Predictivo del Motor

*4.*

## 4.1 Introducción

Una vez que se ha definido e identificado el sistema real y, por lo tanto, se dispone de un modelo analítico validado del mismo, se está en condiciones de plantear las técnicas y estrategias a seguir para realizar un mantenimiento predictivo del sistema en base a una diagnosis predictiva de los fallos que el mismo presente.

Teniendo el modelo analítico del sistema, se pueden simular fallos y ver el comportamiento del mismo. Con esto se consigue ver qué parámetros tienen mayor incidencia sobre la dinámica del sistema. Esto es análogo al estudio de sensibilidad que se realiza en la sección anterior. Esta situación, es vital y necesaria para plantear las técnicas de mantenimiento, ya que el modelo hará las veces de sistema real y por lo tanto, se tiene la posibilidad de analizar qué técnica de mantenimiento es más efectiva y precisa.

Son muchos los fallos que un sistema como un motor de combustión interna alternativo se pueden presentar (combustión, turbo alimentación, proceso de renovación de la carga, mecánicos, etc.). Sin embargo, este proyecto tiene el objetivo de desarrollar técnicas de mantenimiento predictivo enfocadas a la detección de fallos mecánicos. Concretamente, se pretende disponer de una herramienta eficaz en la identificación de muñequillas dañadas y en qué grado lo están, entendiéndose como muñequilla dañada aquélla que presente grietas superficiales y/o internas de forma que la rigidez equivalente asociada se vea mermada respecto a su valor original.

A continuación se desarrollan detalladamente dos técnicas de mantenimiento predictivo enfocadas a la identificación de grietas en el cigüeñal del motor.

## 4.2 Técnica basada en la Optimización de Parámetros

No se va a realizar en este punto una descripción detallada de las bases de funcionamiento de la optimización de parámetros, puesto que ya se ha llevado a cabo dicha descripción en el punto 3.4 del presente texto.

Por tanto, a continuación se hará una descripción de cómo se aplica la técnica de optimización de parámetros en el marco de la diagnosis de fallos en el cigüeñal.

## 4.2.1 Estructura de funcionamiento

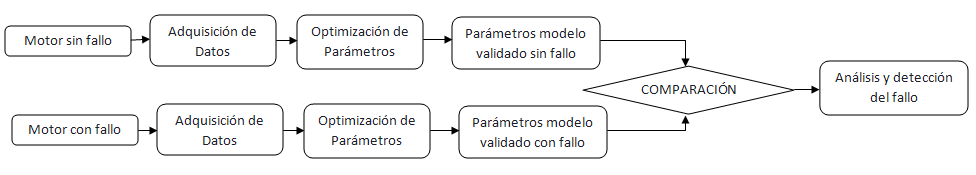
La optimización de parámetros se basa en la variación de los parámetros que se dejan libres dentro de la función objetivo de forma que, éstos se van ajustando hasta que la función objetivo se desvía de la referencia tan sólo la tolerancia permitida, tal y como se expuso en puntos anteriores. Por lo tanto, si el sistema presenta una desviación o fallo, esta situación se verá reflejada en los parámetros resultantes.

Una vez que el modelo ha sido validado, se disponen de los valores de los parámetros en condiciones óptimas de funcionamiento, es decir, sin fallo. Éstos se utilizan como referencia para la comparación de los resultados obtenidos de las optimizaciones de monitorización.

Sirva la siguiente situación a modo de ejemplo:

*Supóngase que el motor presenta una grieta localizada en la muñequilla del cilindro 7. Esta grieta provoca una disminución de la rigidez del grado de libertad asociado a dicho tramo del cigüeñal. Estando el motor en dichas condiciones se realiza un proceso de adquisición de datos del par en el eje. Con estos datos de par, se realiza un proceso de optimización de los parámetros libres del modelo analítico, tal y como se describe en la sección 3.4. Los resultados de esta optimización han de ser iguales a los parámetros de referencia, a excepción del parámetro asociado a la rigidez del grado de libertad asociado al tramo de cigüeñal donde está alojada la muñequilla del cilindro 7.*

*El esquema del proceso es el siguiente:*



1. Esquema funcionamiento técnica de mantenimiento basada en optimización de parámetros.

Tal y como se observa en el esquema anterior, es necesario disponer de una batería de parámetros del sistema en la situación original o sin fallos, con los que realizar una comparación de los parámetros en la situación con fallo. El análisis de dicha comparación será el que dé la oportunidad de detectar el fallo.

## 4.2.2 Valoración

El uso de la optimización de parámetros como técnica para el mantenimiento predictivo del sistema no es del todo acertado, puesto que presenta los siguientes inconvenientes o desventajas:

* Tiempo de cálculo elevado (en torno a 24h por cada proceso de optimización).
* Alta probabilidad de finalización del proceso de optimización en un mínimo local en vez de en el mínimo global.
* Tal y como está estructurado el proceso de optimización, no da la posibilidad de identificar la posición del cilindro dañado.
* Posibilidad de encontrar varias combinaciones de los parámetros que den resultados similares.

Debido a estas desventajas, no se recomienda el uso de la optimización de parámetros como técnica de mantenimiento predictivo.

## 4.3 Redes Neuronales Artificiales

Puesto que la valoración de la optimización de parámetros como técnica para el mantenimiento predictivo del sistema no ha sido positiva, se decide investigar en otra dirección, pero con la misma filosofía. Es decir, se pretende diseñar una herramienta capaz de identificar tanto la posición como el grado de daño. Para ello en este punto se investigan las denominadas redes neuronales artificiales, las cuáles, suelen dar muy buen resultado en este tipo de aplicaciones. El cálculo, definición y análisis de la RNA se hará en calidad de usuario, utilizando para ello el Neural Network Toolbox proporcionado por el software Matlab®.

Antes de comenzar con la descripción del proceso de obtención de la red neuronal, se va a hacer una introducción a las mismas.

## 4.3.1 ¿Qué son las RNA? ¿Cómo funcionan?

Las RNA son el resultado de la interconexión entre las denominadas unidades fundamentales de cálculo o neuronas, de forma que se crea un entramado o red de neuronas inspirado en los sistemas nerviosos biológicos. Son las conexiones y/o relaciones entre neuronas las que determinan el funcionamiento de la red, tal y como ocurre en un sistema nervioso biológico.

Por tanto una RNA es una representación matemática de un sistema nervioso biológico natural. Tanto más avanzada y compleja sea la RNA, más se acercará al sistema nervioso biológico más desarrollado y avanzado, es decir, el cerebro humano.

Al margen de la similitud entre una RNA y el cerebro humano, ambos sistemas comparten varias características:

* ***Aprendizaje:*** en el caso biológico, se define como la capacidad de adquirir el conocimiento de una cosa por medio del estudio, ejercicio o experiencia. Las RNA pueden cambiar su comportamiento en función del entorno. Se les muestra un conjunto de entradas y ellas mismas se ajustan para producir unas salidas consistentes.
* ***Generalizar:*** en el caso biológico, se define como la capacidad de extender o ampliar una cosa. Las RNA generalizan automáticamente debido a su propia estructura y naturaleza. Estas redes pueden ofrecer, dentro de un margen, respuestas correctas a entradas que presentan pequeñas variaciones debido a los efectos de ruido o distorsión.
* ***Abstraer:*** en el caso biológico, se define como la capacidad de aislar mentalmente o considerar por separado las cualidades de un objeto. Algunas RNA son capaces de abstraer la esencia de un conjunto de entradas que aparentemente no presentan aspectos comunes o relativos.

En los sistemas nerviosos biológicos, existen infinidad de neuronas que se encargan de transmitir la información (en forma de impulsos eléctricos) de una neurona a otra. Esta información es percibida por los diferentes receptores (ojos, piel, sistema auditivo, etc.), transmitida por neuronas a la vez que es procesada hasta que las diferentes regiones del cerebro se encargan de interpretar dichos estímulos. A groso modo, los receptores transmiten el estímulo o INPUT que es procesado por las neuronas hasta generar un OUTPUT, que es el resultado de la interpretación de la información por parte de las regiones del cerebro.

En el caso de las RNA, se disponen de tres elementos básicos:

* Neuronas.
* Funciones de transferencia de información entre neuronas.
* Capas.

Tal y como se ha comentado al inicio de esta sección, es la organización o distribución de las neuronas en distintas capas junto con las funciones de transferencia para cada neurona, la que dotará a la red de una capacidad u otra (ajuste de funciones, reconocimiento de patrones de comportamiento, predicciones de estados futuros, etc.). Existen muchas aplicaciones para las RNA, pero el caso que es abordado por el presente texto es el de *reconocimiento de patrones*.

En primer lugar, ¿qué es un patrón de comportamiento? No es más que un conjunto de información que representa la relación *causa – efecto* de un sistema. Por otra parte, para que una RNA sea capaz de reconocer un patrón de comportamiento, es necesario definir una arquitectura interna destinada a tal efecto y posteriormente, entrenar la red con una cantidad tal de patrones que le dé a la red la generalidad suficiente para ser validada.

Por tanto, en el proceso de construcción de una ANN existen dos etapas básicas y de mucha relevancia sobre el resultado final:

* Elección de la arquitectura interna de la RNA.
* Proceso de entrenamiento o aprendizaje de la RNA.

En lo que sigue, se detallarán ambas etapas para el caso que ocupa este proyecto. Sin embargo, y dado lo amplio que es el campo de las RNA, se va a focalizar el proceso descriptivo a efectos de reconocimiento de patrones de comportamiento.

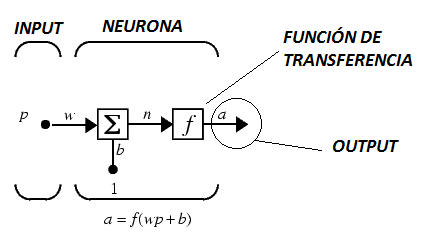
## 4.3.2 Arquitectura de una RNA

Tal y como se ha dejado ver con anterioridad, existen infinidad de combinaciones entre los elementos básicos de una RNA, las cuales, dan resultados muy diferentes. A una combinación de estos elementos básicos se le denomina arquitectura de la red.

Para definir una arquitectura es necesario:

* Definir el número de capas.
* Definir el número de neuronas de cada capa.
* Definir la función de transferencia de las neuronas de cada capa.

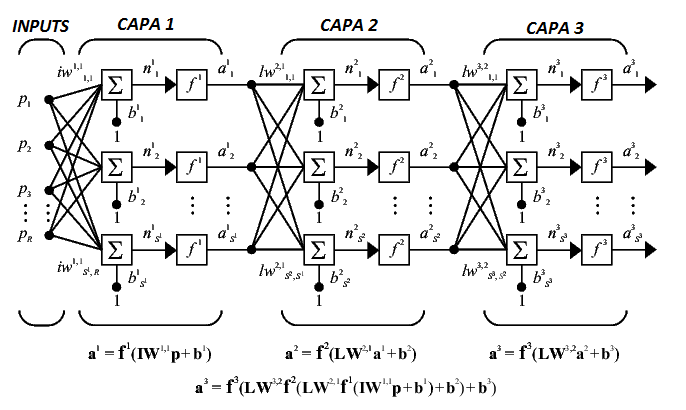
El modelo básico y general de una neurona se puede entender como expresa la siguiente imagen:



1. Esquema general de una neurona.

Tal y como se puede observar, el input llega a la neurona dónde es afectado por un peso (w) y un desfase (b). El resultado de esto (pw +b) es el input de la función de transferencia (f), la cual, actúa generando el output (a). El caso más simple de una RNA es el representado en la figura 53, es decir, una capa con una neurona.

Conforme se aumenta el número de capas y el número de neuronas por capa, se llega a una situación como la representada a continuación:



1. Esquema general de una RNA multicapa.

En el caso más general, como el que se muestra en la figura 54, todos y cada uno de los elementos o variables que conforman el input del patrón de comportamiento son distribuidos a todas las neuronas de la primera capa. Una vez que éstas realizan sus respectivas operaciones, los outputs de cada una de ellas son distribuidos a su vez a todas las neuronas de la capa siguiente y así sucesivamente hasta llegar a la capa de salida, en la que forzosamente el número de neuronas ha de ser estrictamente igual al número de variables o elementos que conforman el output del patrón de comportamiento. Esto es lo que se denomina propagación hacia delante. Sin embargo, se da el caso en el que cierta información se propaga hacia atrás, tal y como se verá más adelante.

Mostrado el funcionamiento interno general de una RNA, toca elegir una arquitectura básica sobre la que construir la misma. Para ello, se realiza una labor de documentación en base a toda la información proporcionada por el software Matlab®. En ella se puede analizar los tipos de arquitecturas existentes y las propiedades que ofrecen cada una de ellas.

La que más se ajusta al propósito de este proyecto es la arquitectura denominada como *Feed Forward Back Propagation Neural Network,* la cual, tiene muy buenas propiedades como reconocimiento de patrones de comportamiento ya que bien entrenada es capaz de dar muy buenos resultados ante inputs que no hayan sido proporcionados durante el entrenamiento y por tanto, que la red nunca haya “visto”, es decir, puede ser dotada de una amplia generalidad.

El entrenamiento de este tipo de RNA precisa de un conjunto de patrones formados por parejas INPUT/OUTPUT con el que realizar la etapa de aprendizaje, la cual, consiste en un ciclo de propagación – adaptación de 2 fases:

* Fase 1 – Propagación hacia delante: en esta fase se estimula la primera capa con el INPUT de un patrón. Este estímulo se propaga por las demás capas hasta generar una salida u OUTPUT. Se compara este OUTPUT con la salida que se desea obtener y se calcula un error para cada neurona de la capa de salida.
* Fase 2 – Propagación hacia atrás: en esta fase, los errores calculados en la fase anterior se transmiten hacia atrás desde la capa de salida hacia todas las neuronas de las capas intermedias. Cada una de estas neuronas recibe un porcentaje de error en función del nivel de participación sobre la salida generada.

Con estas dos fases se consigue un proceso de auto - adaptación de los coeficientes de las neuronas, por eso se dice que el entrenamiento o etapa de aprendizaje es del tipo supervisado y adaptativo.

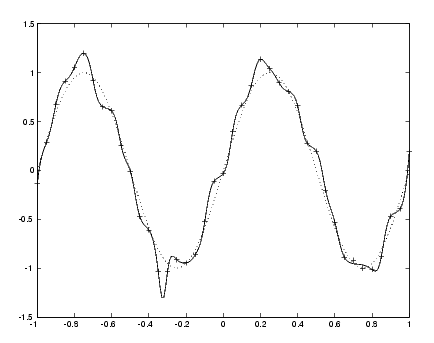
Este tipo de redes, bien entrenadas, son capaces de reproducir con muy buenos resultados situaciones en las que los datos de entrada:

* Estén incompletos.
* Presenten ruido.
* Sean parecidos, pero presenten variaciones.

Estas tres características representan la línea que se va a seguir para la generación de los patrones de entrenamiento para el aprendizaje de la RNA.

La elección del número de capas y del número de neuronas no viene fijado por una regla básica, es decir, no es una ciencia exacta. Además no se pueden tomar cantidades elevadas de neuronas y capas, puesto que existe el riesgo de incurrir en efectos de *overfitting.* Este efecto negativo que se da en las RNA es muy común y es necesario que el proceso de definición de la arquitectura de la red se haga con suma cautela.

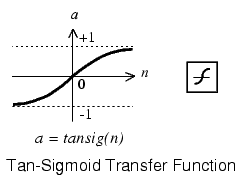
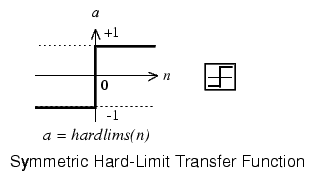
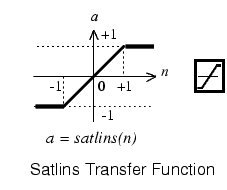
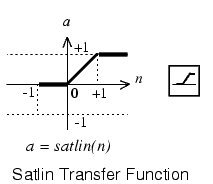
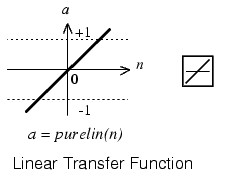
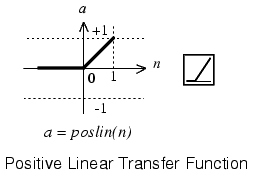
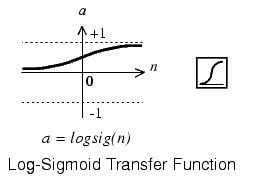
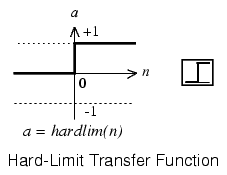
Cuando se presenta *overfitting,* lo que ocurre es que, al tener tantas neuronas, la red memoriza los datos que son presentados durante el entrenamiento y por lo tanto pierde la capacidad de generalización, es decir, no es capaz de predecir outputs a partir de inputs que no hayan sido presentados durante el entrenamiento. Véase la siguiente imagen a modo de ejemplo.



1. Ejemplo de Overfitting. La arquitectura de la red es de 22 neuronas en 3 capas (1-20-1).

Elegir el número de capas y neuronas por capa es un proceso dinámico y retroalimentado por los resultados de las simulaciones fruto del proceso de entrenamiento, es decir, se parte de una base y se van variando tanto número de capas por un lado, como número de neuronas por capa, como ambas cosas simultáneamente. Por ello, la definición completa de la arquitectura de la red se detallará en el apartado de entrenamiento.

El último apartado que resta en la definición de la arquitectura de una RNA, es la elección de las funciones de transferencia de las neuronas. Tal y como se dejó ver antes, la función de transferencia no es más que el elemento de transformación de la información que le llega a la neurona, es decir, transforma el input local de la neurona en un output local de la misma que a su vez será el input local de otras tantas neuronas (en función de la configuración de la red). La forma que tiene la función de transferencia de transformar la información queda definida según su tipología. Algunos ejemplos de funciones de transferencia son:



1. Ejemplos de funciones de transferencia para las neuronas de una RNA.

Se utilizan funciones de transferencia de tipo “sigmoide”, concretamente la función *tansig*, por la agilidad en los tiempos de cálculo y la buena resolución que ofrecen. Esto se debe a que este tipo de funciones trabaja con valores comprendidos en el intervalo [-1 1]. Para poder trabajar con estas funciones de transferencia, es necesario que la información destinada al entrenamiento y simulación sea tratada de forma que los valores estén comprendidos en el mismo intervalo con el que trabaja la función de transferencia. En el punto siguiente se detalla el proceso de pre y post tratado de la información para poder utilizar las funciones de transferencia tipo “sigmoide”

## 4.3.3 Entrenamiento o aprendizaje de la red

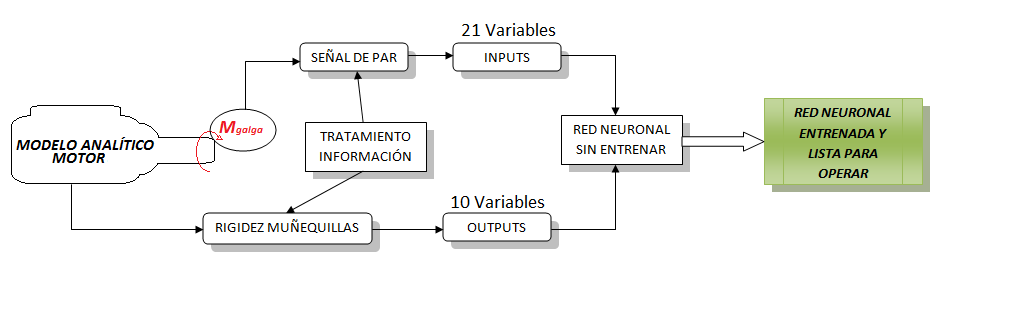
Una vez que decidida la arquitectura de la red, es necesario definir qué información de entrada estimulará la red y, por consiguiente, qué información se requiere que sea generada por la red (cabe recordar que la elección de cantidades de capas y neuronas se hace en un proceso iterativo durante la fase de aprendizaje. En este punto se detallará la generación de patrones).

Tal y como se ha comentado en la introducción a las técnicas para la diagnosis de fallos, son muchos los fallos que se pueden encontrar en un motor a combustión interna alternativo. Sin embargo, este proyecto se centra en la detección de grietas en el cigüeñal del mismo.

Parece coherente por tanto, que la información generada por la red muestre el estado de las muñequillas del cigüeñal en términos de rigidez, puesto que la presencia de grietas en un sólido está intrínsecamente relacionada con la característica de rigidez que presenta el mismo, concretamente con la pérdida de rigidez. Dado que el motor está formado por 10 cilindros, serán las rigideces asociadas las muñequillas de todos los cilindros las variables de salida de la red u OUTPUTS.

Por otro lado y de cara a una futura implantación en red de la herramienta, puesto que se disponen de sensores alojados en el sistema que permiten la adquisición de la señal de par en el eje, se decide alimentar la red con la señal de par. Esta señal de par sufre primero un proceso de transformación al dominio de la frecuencia, vía Fourier. Se usan 10 armónicos en el proceso de transformación de la señal, por lo que se disponen de 21 coeficientes (A0, A1,…, A10, B1,…, B10). Se toman los 21 coeficientes como INPUTS para la red, ya que usando todos ellos, se dispone de más información (valor medio, módulo y fase).

Una representación esquemática del flujo de información en la red es la siguiente:



1. Diagrama esquemático representativo del flujo que toma la información hasta llegar a la red.

Decidida la información que va a tratar la red, es necesario comenzar el proceso de entrenamiento de la misma. Para ello, hay que alimentar la misma con un conjunto de patrones que cubran la casuística más amplia posible. Tanto más amplio sea el conjunto de patrones de entrenamiento y las situaciones contempladas, tanto más eficaz será la red en la labor de detección y caracterización de fallos en el motor. Por ello, inicialmente se genera un conjunto de patrones de entrenamiento destinado a ver la respuesta de la red en una dirección y tras los resultados obtenidos, se decide generar otro conjunto de patrones de entrenamiento focalizado en una casuística más amplia. Los dos casos trabajados son:

1. Generación de patrones para detección de grietas (5 niveles diferentes) en 10 cilindros y para 13 niveles de carga distintos, con patrón de curva de presión en cámara fijo 🡪 650 patrones.
2. Generación de patrones para detección de grietas (3 niveles diferentes) en 1 cilindro y para 2 niveles de carga, pero con patrones de curva de presión en cámara variable 🡪 354.294 patrones.

El hecho de generar 2 conjuntos diferentes de patrones de entrenamiento radica que en la ambición de dotar a la red de una generalidad más amplia con el fin de poder implantar la red en el sistema real. En el primer caso estudiado, se analiza si la red es capaz de diferenciar entre cilindros no dañados y dañados (y en qué nivel de daño se encuentran) para un conjunto de 5 niveles de daño diferentes y para 13 niveles de carga diferentes. Como se verá más adelante, los resultados son muy buenos y debido a ello, se decide trabajar en la generación de patrones de entrenamiento que representen condiciones más realistas.

En el segundo caso trabajado se considera que, para cada nivel de carga considerado, la curva de presión en cámara para cada cilindro no es fija sino que puede sufrir variaciones en su morfología. Esta situación reflejaría la casuística real que se presenta en la cámara de combustión, en la que las curvas de presión no siempre son iguales. Además, se combina esta casuística con 3 niveles de grieta para la muñequilla existente entre el cilindro 9 y el cilindro 10 (esta muñequilla es la que recibe la totalidad del par motor y por tanto, la más susceptible a fallar. Además, el modelo es muy sensible ante cambios en la rigidez asociada a dicha muñequilla). En el apartado de generación de patrones para este segundo caso, se detallará la forma de construir los patrones de presión y los criterios adoptados.

Como se ha dejado adivinar, la generación de patrones de entrenamiento se hará con la ayuda del modelo analítico obtenido en el proceso de identificación del sistema. No es posible ni coherente usar el sistema real para generar estos patrones de entrenamiento, puesto que por la casuística que representan, sería necesario incurrir en acciones destructivas sobre el sistema real para la adquisición de los patrones.

## 4.3.4 RNA para patrones de presión en cámara fijos

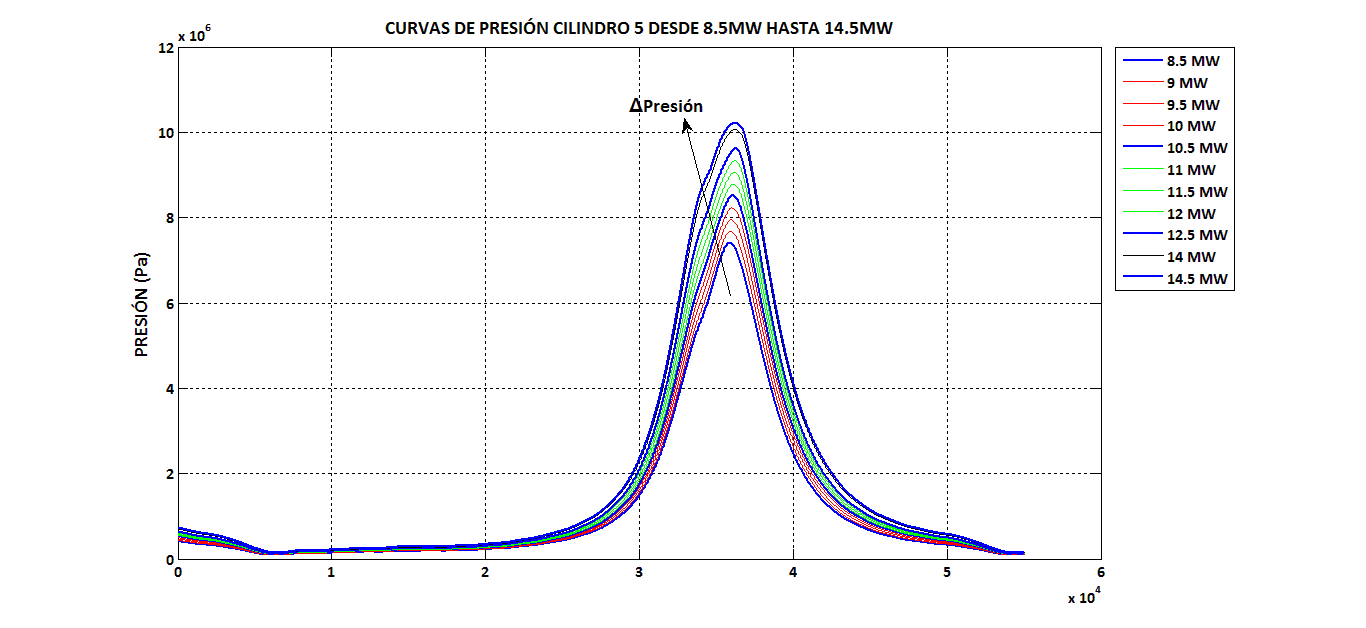
### Generación de patrones de entrenamiento

La forma de construir los patrones de entrenamiento en este caso, se detalla a continuación:

* Se afecta 1 muñequilla con un factor proporcional con valores entre 0 y 1 que representa el nivel de daño. Se consideran 5 niveles de daño diferentes: 0%, 5%, 10%, 15% y 20%. Estos porcentajes de daño sobre la rigidez de las muñequillas son aplicados sobre todas las muñequillas, de forma no simultánea.
* Se ejecuta el modelo analítico con esta rigidez afectada.
* A modo de adquisición de datos, se toma la señal de par en el eje generada por el modelo.
* Se repite este proceso para todas las muñequillas y todos los niveles de carga posibles. Comentar en este punto que, inicialmente se disponen de datos de presión adquiridos para 4 niveles de cargas diferentes. Mediante un proceso de interpolación que se detallará a continuación, se generan en total 13 niveles de carga diferentes.

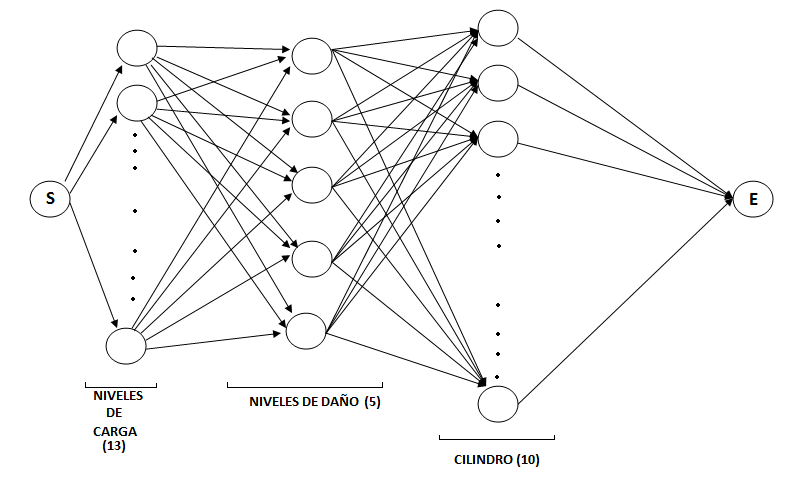
Puesto que se disponen de 13 niveles de carga, combinados con 5 niveles de daño diferentes para cada uno de los 10 cilindros del motor, se disponen en total de 650 patrones de entrenamiento.

El proceso de generación de los 9 niveles de carga adicionales a partir de los 4 niveles de carga de los que se disponen gracias al proceso de adquisición, se realiza mediante una interpolación lineal. Se imponen niveles de carga intermedios a los disponibles y se construyen las curvas. Véase la siguiente imagen, en la que se observan las curvas de presión para un cilindro concreto (cilindro 5), a modo de ejemplo, para los diferentes niveles de carga:



1. Representación de las curvas de presión interpoladas para la generación de los 13 niveles de carga.

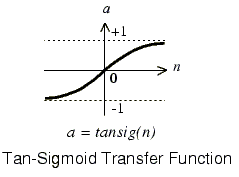
Por último, se muestra a continuación un diagrama de flujo en el que se muestran todas las posibilidades que se pueden tomar para generar patrones:



1. Grafo de decisión para la generación de patrones de entrenamiento de la RNA para patrones de presión fijos.

### Tratamiento de la información de entrenamiento

Como ya se comentó con anterioridad, las funciones de transferencia que se han elegido para las neuronas que conforman la red, son de tipo “sigmoide” y concretamente la función *tansig*. Ésta, tiene la particularidad de que los valores que arroja están comprendidos entre -1 y 1, por lo que para un entrenamiento coherente y eficaz de la red, la información de entrada y salida ha de estar comprendida entre dichos valores.



1. Representación de la función de transferencia tipo “sigmoide” que utilizan las neuronas de la RNA.

Para conseguir que la información tanto de entrada (INPUT) como de salida (OUTPUT) esté comprendida en el intervalo [-1 1], se utiliza una función de librería proporcionada por el software Matlab®. Dicha función atiende al nombre de *mapminmax* y realiza un mapeo de la información de tal forma que:

* Al máximo valor de la información de partida le asigna el valor 1.
* Al mínimo valor de la información de partida le asigna el valor -1.
* El resto de valores intermedios de la información de partida son reescalados según los factores de conversión que hayan sido necesarios para las transformaciones de los valores máximos y mínimos.

Cuando se aplica esta función tanto a INPUT como a OUTPUT, la función genera una estructura para cada caso, que contienen la siguiente información:

* Para el caso del tratado de INPUT de entrenamiento, se genera una estructura que contiene la información necesaria para mapear futuros INPUTS que se deseen introducir en la RNA en busca de los correspondientes OTPUTS simulados por la misma.
* En el caso del tratado de OUTPUT de entrenamiento, se genera una estructura que contiene la información necesaria para revertir el mapeo de OUTPUTS simulados y poder así comparar los resultados de la RNA en términos de rigidez.

Teniendo ya la información tanto de entrada como de salida ya mapeada y tratada, solo resta un pequeño detalle antes de alimentar la red con los patrones de entrenamiento. Este detalle consiste en el “orden” en el que se construye la matriz de entrenamiento, es decir:

* Cuando se ejecuta el entrenamiento, la aplicación precisa de una matriz de entrada y otra de salida, donde estén contemplados todos los patrones de entrenamiento. En el caso de los INPUTS, hay que generar una matriz de 21x650. El orden de las 650 columnas (patrones) importa en el sentido de que, el software divide automáticamente el training dataset en 60% entrenamiento, 20% simulación y 20% validación de forma correlativa, es decir, conforme “va leyendo los patrones (cada columna corresponde a un patrón)”. Para asegurar que en cada división existen patrones correspondientes a todos los niveles de carga, se crea la matriz de patrones con un orden aleatorio.

### Set – up de la red y entrenamiento

Una vez que se ha preparado la información de entrenamiento, según la forma de proceder anteriormente comentada, sólo queda realizar el set – up de la red y entrenarla.

El set – up no es más que decidir qué funciones de entrenamiento y cálculo de error cometido se utilizarán durante el entrenamiento, así como el valor inicial de los pesos y desfases de cada una de las neuronas.

En el caso de la función de entrenamiento, existe una gran variedad dónde elegir. Cada una de ellas presenta unas cualidades diferentes (algoritmos internos, velocidad, consumo de recursos, etc.), que derivan en un mejor o peor comportamiento en función del uso que se le quiera dar a la red (reconocimiento de patrones, tratamiento de señales, predicción, etc.).

En este primer caso, se utiliza la función de entrenamiento *Levenberg-Marquardt,* la cual, presenta muy buenas cualidades ante ajuste de funciones con una velocidad aceptable. Además según la bibliografía, se presenta como la más potente y capaz de dar buenos resultados ante casi cualquier problema presentado.

La función de cálculo de error sin embargo, no presenta tantas opciones dónde elegir. Sólo existen 2 funciones para el cálculo de error:

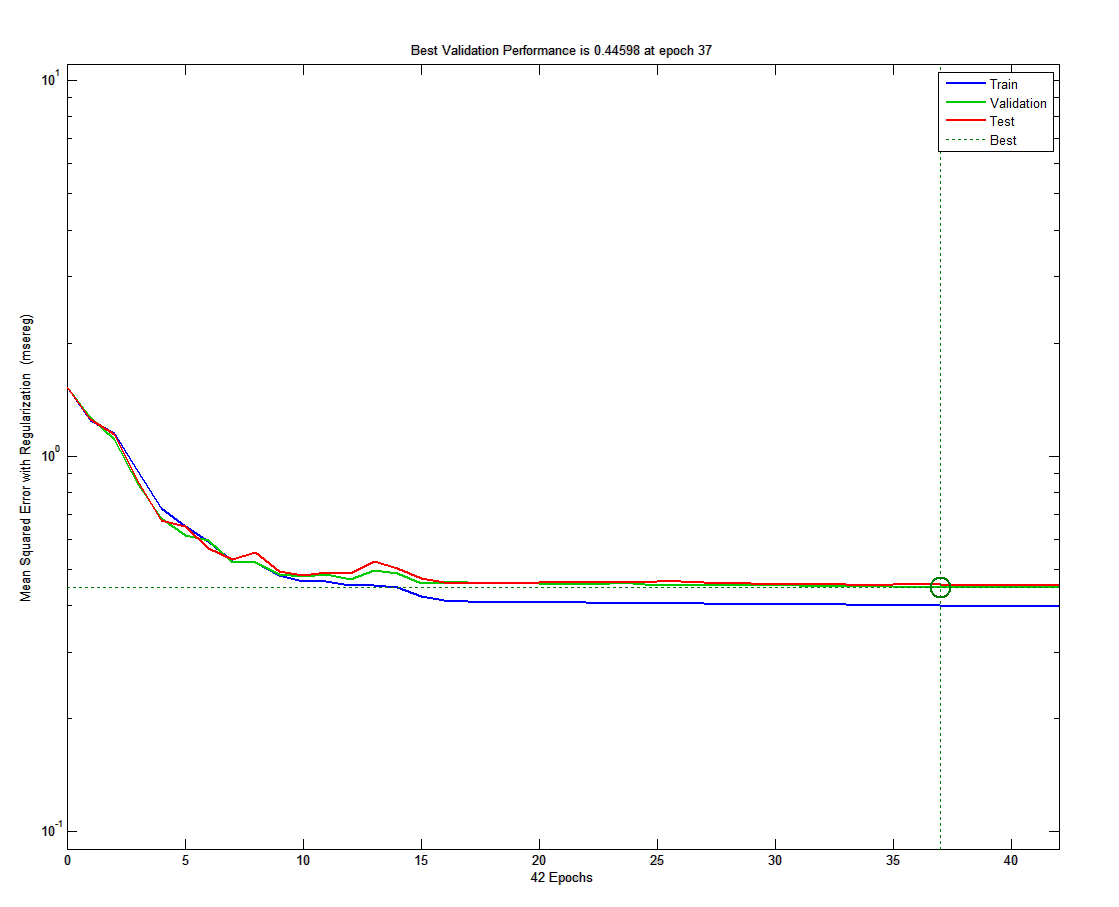
* *Mean sum of Square Error (MSE):* esta función calcula el error en el sentido de los mínimos cuadrados.
* *Mean sum of Square Error Regresion (MSEREG):* es una modificación de la función anterior que deriva en una mejora de cara a la generalidad que adquiere a la red sin pérdida de ajuste. La ley que controla esta función de cálculo de error es la siguiente:
* , donde:
  + g es el performance ratio.
  + MSW es el error en el sentido de los mínimos cuadrados de los pesos y desfases de la red.

Se toma por tanto, la función MSEREG como función para el cálculo de errores.

Una vez que se ha finalizado el proceso de set – up de la red, se da inicio al proceso de entrenamiento con los patrones generados anteriormente. Teniendo en cuenta que la cantidad de neuronas de la capa de salida está fijada por el número de variables que componen el OUTPUT, la arquitectura de la red queda definida tras el proceso de entrenamiento como se detalla a continuación:

* 4 capas, de las cuales 1 es de entrada, 1 de salida y 2 ocultas.
* 10 neuronas en la capa de salida (fijado por OUTPUT).
* 15 neuronas en la capa de entrada.
* 20 neuronas en la primera capa oculta.
* 15 neuronas en la segunda capa oculta.
* Todas las funciones de transferencia de las neuronas son *tansig*, las cuales, dan muy buenos resultados para este tipo de aplicaciones.

Una vez se obtiene la red, se simula con patrones de simulación que consisten en patrones que no han sido contemplados en el conjunto de entrenamiento. Pero antes de pasar al apartado de simulación y resultados, véase la siguiente imagen dónde se puede apreciar el performance obtenido por la red:



1. Curva de aprendizaje de la red 15x20x15x10 para patrones de presión fijos.

### Simulación y resultados

Se generan patrones de simulación de forma que la casuística no haya sido contemplada en la generación de patrones de entrenamiento, para ver cómo responde la red ante situaciones que no han sido proporcionadas durante el entrenamiento y probar así la generalidad que ha adquirido la misma.

Los patrones de simulación consisten en:

* 5 niveles de grieta diferentes: 0%, 6%, 9%, 14% y 19%.
* 13 niveles de carga: desde 8.5MW hasta 14.5MW, con incrementos de 0.5MW.
* 10 cilindros.

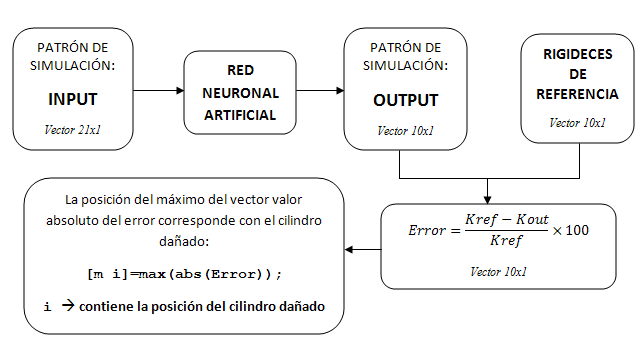
Lo importante de cara a los resultados que se obtengan de la simulación no es tanto que de el valor exacto de daño de la muñequilla sino la detección de la muñequilla dañada.

Tras la simulación de los 650 patrones calculados al efecto, una vez que se obtienen los valores de rigidez proporcionados por la red, se evalúa la bondad de la misma en base al porcentaje de equivocación en la detección del cilindro dañado.

Para identificar el cilindro dañado, se utiliza el siguiente método, en base a los resultados de la simulación:

* Se toma el valor de rigidez de referencia (es decir, sin daño) de cada muñequilla.
* Se calcula el error porcentual de la rigidez simulada frente a la rigidez de referencia, para todos los cilindros y para cada caso.
* El máximo valor absoluto del error relativo calculado, corresponde al cilindro dañado.

Véase el siguiente esquema ilustrativo del proceso de identificación del cilindro dañado:



1. Esquema ilustrativo del proceso análisis de la información generada por la RNA para la detección del cilindro con presencia de grietas.

Los resultados son los siguientes:

* La red no presenta sensibilidad ante daños en el cilindro 1.
* La red presenta alta sensibilidad ante daños en cilindros 2 a 10.
* Detecta los fallos en cilindros con una bondad del: ***99.231%***
* En los 4 casos que la red ha fallado en la detección de daño, siempre ha sido en las situaciones en las que estaba dañada la muñequilla del cilindro 2 y con un nivel de daño del 6%, es decir, las condiciones ante las que el modelo presenta menor sensibilidad.
* Una vez detectado el cilindro dañado, la desviación máxima respecto al valor de consigna es del ***5.65%.***

## 4.3.5 RNA para patrones de presión en cámara variables

Para el caso en el que los patrones que siguen las curvas de presión en el interior de las cámaras de combustión de los cilindros son variables, se han desarrollado 3 RNA iguales, en lo que a arquitectura se refiere, pero con distintos conjuntos de patrones de entrenamiento. Siendo la misma arquitectura, los patrones de entrenamiento atienden a:

* Nivel de carga de 8MW, 3 niveles de grieta y 59049 patrones de presión.
* Nivel de carga de 14MW, 3 niveles de grieta y 59049 patrones de presión.
* Niveles de carga de 8MW y 14MW, 3 niveles de carga y 59049 patrones de presión.

Con la tercera RNA entrenada se tratará de analizar el comportamiento de la misma ante cargas intermedias no contempladas en el entrenamiento, mientras que con las otras 2 RNA restantes se analizará tanto el error en detección de fallo como en nivel de fallo. Además se analizará si existe alguna tendencia con la carga.

### Generación de patrones de entrenamiento

En este segundo caso, la generación de los patrones para la etapa de entrenamiento cambia, puesto que ahora se pretende tener en consideración la variabilidad que presentan las condiciones de combustión en el interior de la cámara de combustión de cada uno de los 10 cilindros, conjuntamente con la existencia de grietas en muñequilla.

Se decide investigar según esta línea puesto que la red obtenida en el caso anterior es entrenada con patrones de entrenamiento donde las curvas de presión siguen patrones que son fijos para cada cilindro (si bien cambian en magnitud, pues se entrena para diferentes niveles de carga). En el caso real existe variabilidad en las condiciones que se dan en la cámara de combustión y por tanto en el mismo proceso de combustión desarrollado en el interior de los cilindros que afectan no a la magnitud de la curva de presión sino a la forma de ésta, es decir, que la curva de presión en cámara siga otro patrón. Se pretende comprobar si el efecto de la variabilidad en las curvas de presión sobre la curva de par produce resultados equivalentes al efecto que produce la existencia de grietas en el cigüeñal sobre la misma curva de par resultante, ya que de ser así, la herramienta de identificación podría detectar un falso fallo. Debido a esto, el output de las RNA que se desarrollen en este segundo caso ha de contemplar las 10 rigideces, por si existiera la posibilidad de que un patrón de presión generara un falso fallo en cualquier otro cilindro.

Para evitar la posibilidad de falsas detecciones de fallos en el cigüeñal por parte de la RNA, se decide construir un conjunto de patrones de entrenamiento que tiene las siguientes características:

* 3 patrones de presión diferentes para cada uno de los 10 cilindros, con la posibilidad que se den simultáneamente. Esto supone 310=59049 patrones de presión, según la ley de permutaciones con repetición.
* Para cada uno de los 59049 patrones de presión, la posibilidad de que se den 3 factores de daño sobre la muñequilla del cilindro 10. Esto supone 177.147 patrones de entrenamiento, por cada nivel de carga.
* Se consideran 2 niveles de carga diferentes: 8.5MW y 14.5MW. Nótese que al ser los valores extremos del intervalo de potencia ofrecida por el motor, en el caso del entrenamiento para ambos niveles de carga conjuntamente, se podrá comprobar el comportamiento de la red resultante ante los niveles de carga intermedios. Al considerar 2 niveles de carga diferentes, el conjunto completo de patrones de entrenamiento asciende a la cantidad de 354.294 patrones, mientras que para cada nivel de carga por separado asciende a 177.147 patrones.

Nótese que sólo se está considerando, en este segundo caso, una muñequilla y 2 niveles de carga. Esto se debe al gran volumen de cálculo en el que se incurriría a la hora de tener en consideración los 10 cilindros y los 13 niveles de carga, es decir, 23.029.110 patrones de entrenamiento, considerando sólo 3 niveles de daño sobre las muñequillas. Como es un número muy amplio que conlleva un tiempo de cálculo muy elevado, se decide analizar la respuesta ante este caso, que se dibuja más sencillo pero que ofrece una visión global.

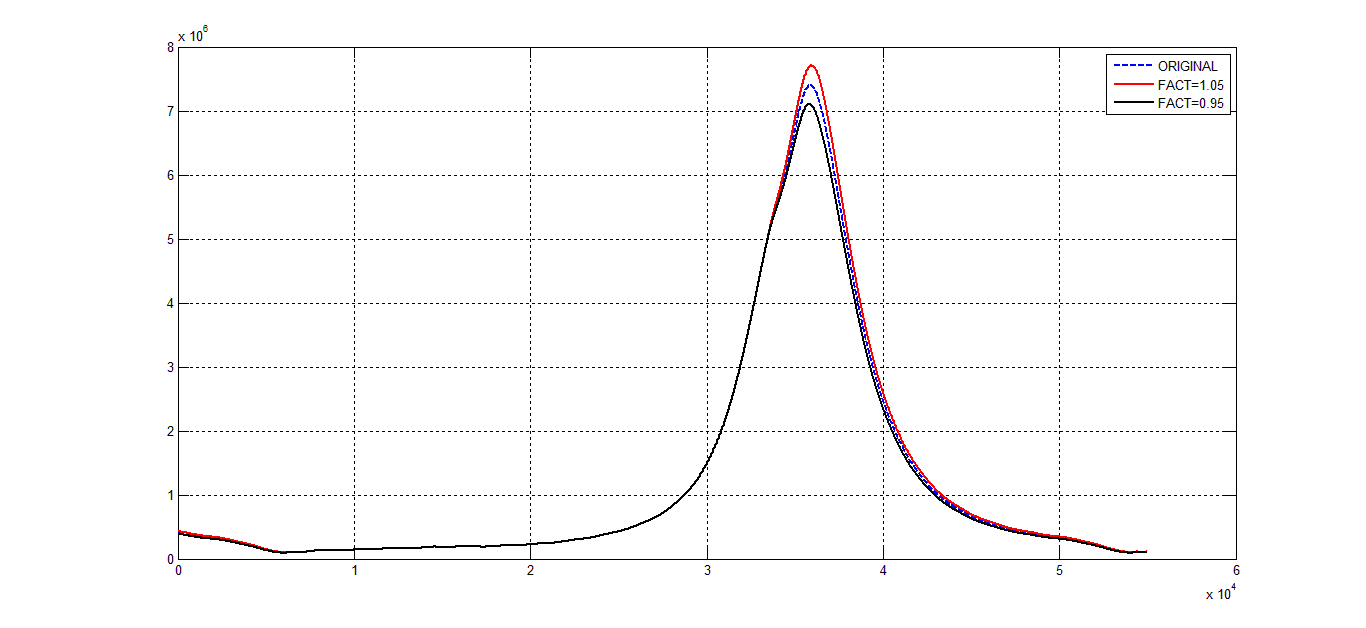
a.1) Cálculo de los patrones de presión en cámara

En primer lugar, comentar que esta etapa es independiente del nivel de carga considerado en lo que a forma de proceder se refiere, si bien es verdad que las curvas de presión a modificar son distintas en un nivel de carga y en otro, como es lógico.

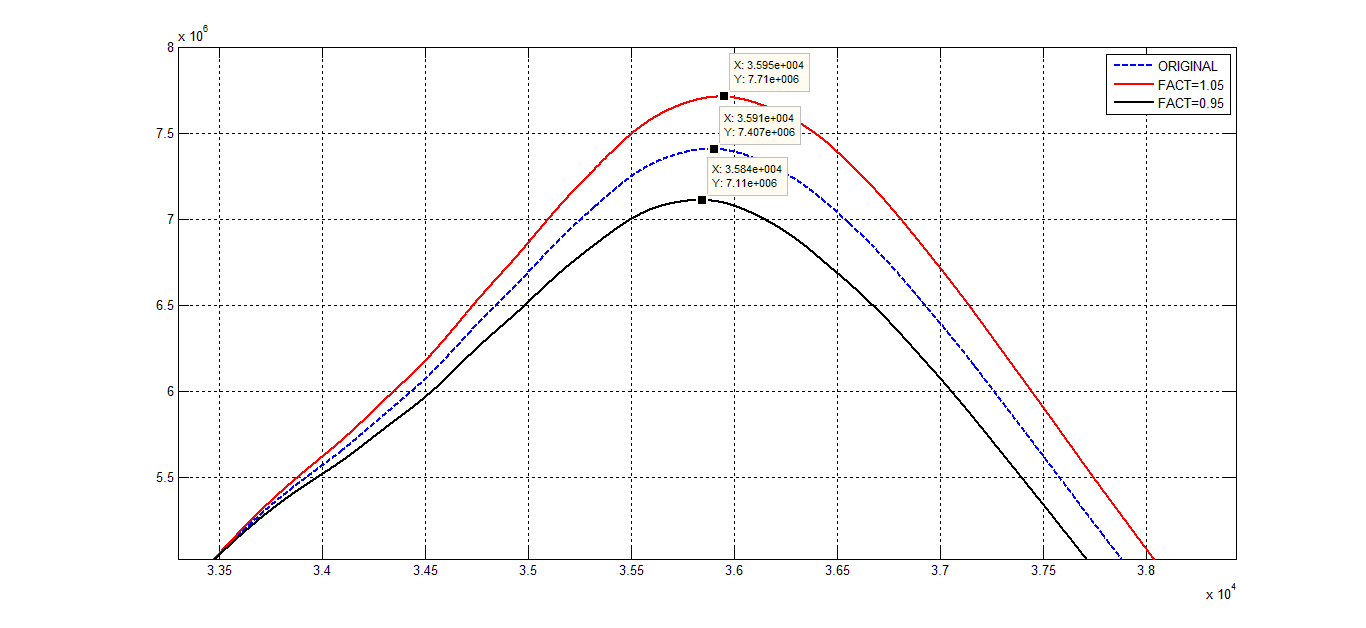
Para la generación de los patrones de presión en cámara, se decide afectar la curva de presión de cada cilindro según los siguientes criterios:

* *La zona de la curva correspondiente a la compresión ha de permanecer inalterada, pues en el caso real, esta zona de la curva presenta esta característica. Sólo serán afectadas por tanto, las zonas de combustión, expansión y escape.*
* *Los factores que se utilizarán para modificar las curvas de presión de cada cilindro serán elegidos de forma que, la potencia media indicada de cada cilindro varíe en un ±10% como máximo sobre la potencia media indicada de cada cilindro en la situación original.*

Con estas dos consideraciones, se buscan los dos factores (1.05 y 0.95) y se construyen las curvas de presión modificadas. Véase en la siguiente imagen, para un cilindro cualquiera, las curvas de presión modificadas respecto a la curva original, en el caso de 8MW:



1. Representación de las curvas de presión modificadas respecto a la curva original, para 8MW.



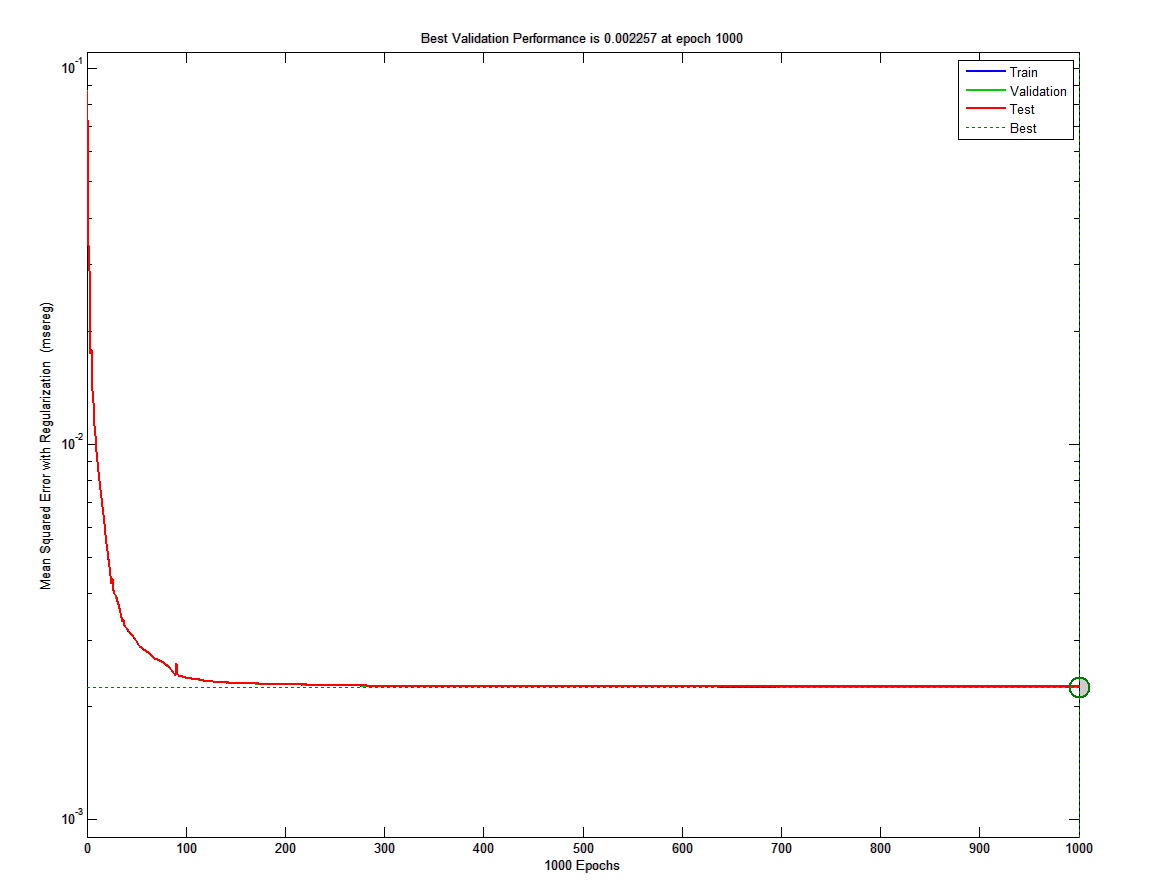
1. Detalle de la zona de máxima presión, en la que se puede apreciar la diferencia entre cada curva, para 8MW.

Una vez calculados los 59049 patrones de presión en cámara, sólo resta generar los 354.294 patrones de entrenamiento y proceder al entrenamiento de la misma.

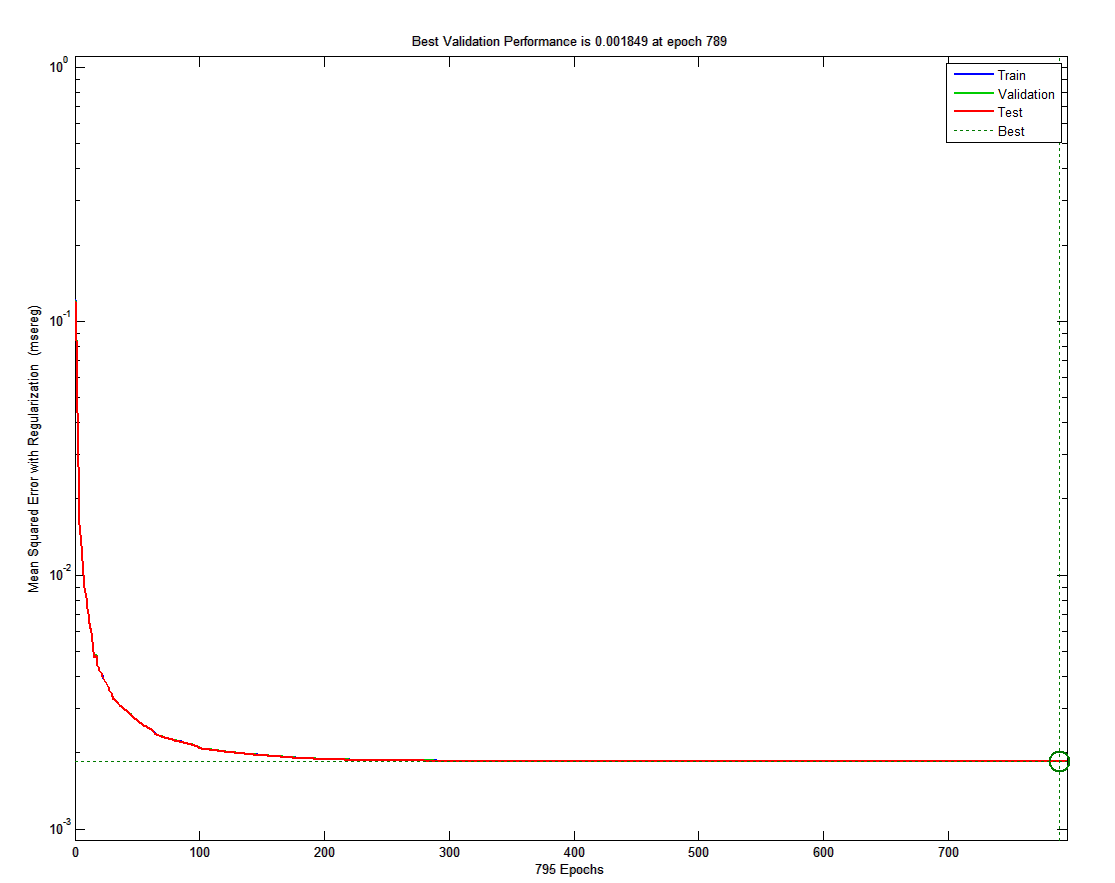
Cabe mencionar llegado este momento, que el proceso de tratamiento de los patrones de entrenamiento, set – up de la red, simulación y proceso de detección de fallos es análogo que en el caso anterior (sección 4.2.4), con la siguiente excepción:

* La función de entrenamiento elegida para este caso es diferente a la del caso anterior. En este caso se ha elegido la función *Resilient Backpropagation* por tener las cualidades de consumir muchos menos recursos que la función anterior *(Levenberg-Marquardt)* además de ofrecer unos resultados muy buenos en los casos de reconocimiento de patrones. Se elige esta función de entrenamiento, principalmente por el gran volumen de información a tratar por la red durante el entrenamiento.

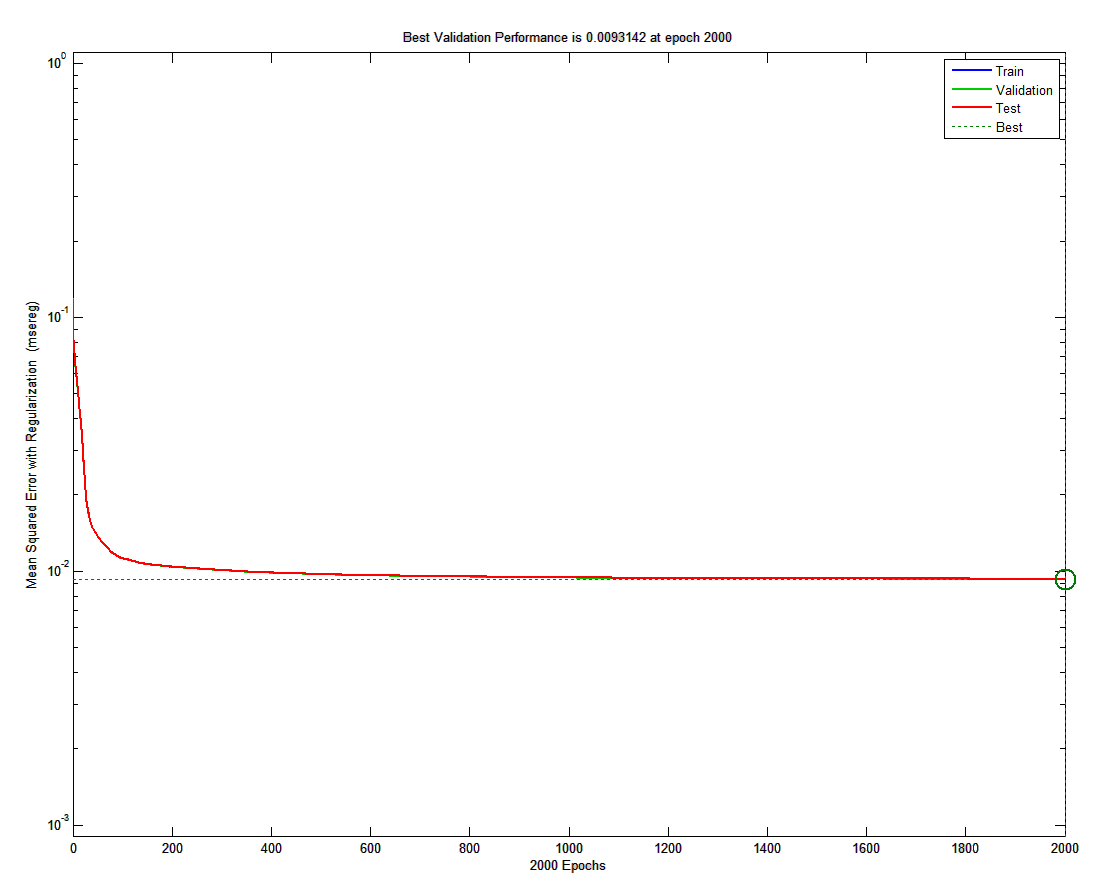
Al igual que en el caso anterior, se procede al entrenamiento con los patrones construidos a tal efecto. Una vez entrenadas las diferentes RNA, se simulan para analizar los resultados. Antes de pasar al apartado de análisis de los resultados de simulación, véanse las siguientes imágenes dónde se puede apreciar el performance de cada una de las 3 RNA obtenidas:



1. Curva de aprendizaje de la RNA 15x10 para patrones de presión variables y nivel de carga 8MW.



1. Curva de aprendizaje de la RNA 15x10 para patrones de presión variables y nivel de carga 14MW.



1. Curva de aprendizaje de la RNA 15x10 para patrones de presión variables y niveles de carga 8MW y 14MW.

### Simulación y resultados

Al igual que en el caso de patrones de presión en cámara fijos, en este caso los patrones de simulación son generados de tal forma que la casuística que representan no hay sido contemplada durante el entrenamiento de la red.

En el caso de la RNA para 8MW, el conjunto de patrones de entrenamiento representa la siguiente casuística:

* 5 niveles de grieta diferentes: 0%, 5%, 7%, 15% y 18%.
* 30 patrones de presión diferentes para cada nivel de grieta, escogidos aleatoriamente, pero que al menos contenga el caso en el que no existe modificación sobre la curva de presión de ningún cilindro.
* Este conjunto asciende a 150 patrones de simulación.

En el caso de la RNA para 14MW, el conjunto de patrones de entrenamiento representa la siguiente casuística:

* 5 niveles de grieta diferentes: 0%, 5%, 7%, 15% y 18%.
* 30 patrones de presión diferentes para cada nivel de grieta, escogidos aleatoriamente, pero que al menos contenga el caso en el que no existe modificación sobre la curva de presión de ningún cilindro.
* Este conjunto asciende a 150 patrones de simalción.

Por último, para el caso de la RNA para 8MW y 14MW conjuntamente, el conjunto de patrones de entrenamiento no es más que la combinación de los 2 conjuntos anteriores, por lo que este último conjunto de patrones de entrenamiento asciende a 300 patrones de simulación.

La forma de evaluar la bondad de cada una de las RNA obtenidas en este segundo caso se hará en base a los siguientes criterios:

* Número de aciertos/equivocaciones en la detección del cilindro dañado frente al número de casos simulados.
* Desviación del nivel de grieta detectado respecto al valor de consigna.

RNA para 8MW

* Puesto que la red ha sido entrenada para la detección de grietas en una sola muñequilla, no presenta sensibilidad ante el resto de muñequillas.
* Si la red es alimentada con patrones donde están dañados otros cilindros, no presenta falsos errores sobre el cilindro para la que está entrenada.
* La red detecta fallos en la muñequilla con una bondad del: ***100%.***
* Una vez detectado el cilindro dañado, la máxima desviación frente al valor de consigna es del: ***2.367%.***

RNA para 14MW

* Puesto que la red ha sido entrenada para la detección de grietas en una sola muñequilla, no presenta sensibilidad ante el resto de muñequillas.
* Si la red es alimentada con patrones donde están dañados otros cilindros, no presenta falsos errores sobre el cilindro para la que está entrenada.
* La red detecta fallos en la muñequilla con una bondad del: ***100%.***
* Una vez detectado el cilindro dañado, la máxima desviación frente al valor de consigna es del: ***2.53%.***

RNA para 8MW y 14MW

* Puesto que la red ha sido entrenada para la detección de grietas en una sola muñequilla, no presenta sensibilidad ante el resto de muñequillas.
* Si la red es alimentada con patrones donde están dañados otros cilindros, no presenta falsos errores sobre el cilindro para la que está entrenada.
* La red detecta fallos en la muñequilla, cuando realmente existen, con una bondad del: ***100%.*** Sin embargo, en el caso en el que la muñequilla no está dañada, la red presenta desviación respecto al valor de consigna (1 cuando la muñequilla no está dañada). Esta desviación induce a pensar que existe grieta cuando realmente no existe y en algunos casos muestra que existe grietas del ***4.3%***, siendo la media de la desviación es del ***2%.***
* Una vez detectado el cilindro dañado, la máxima desviación frente al valor de consigna es del: ***2.367%.***

**Conclusiones y Resultados**

*5.*

## 5.1 Introducción

El desarrollo de este texto se ha centrado principalmente en dos cuestiones básicas:

* Identificación del sistema.
* Desarrollo y análisis de técnicas para la diagnosis predictiva de fallos en el sistema.

La labor de identificación del sistema se ha llevado a cabo durante la sección 3 del presente texto, en la que se han desarrollado las vías de trabajo:

* Módulo de Matlab®, *System Identification Toolbox™.*
* Técnica de *Optimización de Parámetros.*

Por otra parte, en la sección 4 se desarrollan e investigan las técnicas de diagnosis predictiva de fallos en el sistema. Concretamente, se ha trabajado en las siguientes técnicas:

* Herramienta de Identificación vía *Optimización de Parámetros.*
* Herramienta de Identificación vía *Redes Neuronales Artificiales (Neural Network Toolbox™).*

Además del desarrollo de la identificación del sistema y de la búsqueda de la técnica de mantenimiento predictivo óptima para el sistema, se han desarrollado cuestiones como:

* Descripción detallada del sistema, en la sección 2.
* Descripción detallada de los fundamentos de cada vía de trabajo tanto en la labor de identificación del sistema como en el desarrollo de las técnicas de mantenimiento predictivo del sistema.

A continuación, se hará un breve repaso a los resultados obtenidos en cada caso y a las valoraciones de dichos resultados. Además, al final de este capítulo, se muestran los efectos que se pueden observar sobre la dinámica del sistema cuando se afecta el mismo con patrones de presión en cámara variables y grietas en el cigüeñal.

## 5.2 Identificación del Sistema

En esta etapa del proyecto, se investigan dos vías de trabajo con las que conseguir la definición completa del sistema real:

* Por una parte se trabaja con el módulo ofertado por Matlab®, *System Identification Toolbox,* el cual, no ofrece un resultado óptimo en el marco de los objetivos que persigue este proyecto, pues si bien ofrece buena representación del sistema real (en términos de ajuste y errores en los resultados), los modelos obtenidos con este módulo de trabajo carecen de sentido físico y por tanto, no dan la oportunidad de realizar un análisis detallado de la dinámica del sistema real. Debido ello, se descarta como herramienta identificadora del sistema real.
* Por otra parte, se hace uso de las funciones de librería de Matlab® de la mano de la programación en código Matlab®, con el fin de obtener un modelo analítico del sistema real. A tal efecto, se diseña un modelo analítico no lineal de 15 grados de libertad en el que se dejan ciertos parámetros libres y se tiene en consideración toda la dinámica tanto lineal como no lineal del sistema. Una vez construido, se lleva a cabo la resolución de un problema de optimización restringido. Con ello, se consigue disponer de un modelo analítico cuyos parámetros representan la dinámica del sistema real y el cual, ofrece unos grados de ajuste y representación muy reales, por lo que se valida y acepta el modelo analítico no lineal como representación del sistema real.

## 5.3 Herramienta para la identificación de fallos en el sistema

En esta otra etapa del proyecto, se busca diseñar una red RNA que consiga realizar una labor de identificación de daños en el cigüeñal a partir de los coeficientes de Fourier de la señal de par adquirida en el sistema real.

En la labor de diseño y definición de una red, existen dos etapas importantes a las que hay que prestar especial atención:

* Definición de una arquitectura óptima.
* Proceso de entrenamiento de la red 🡪 generación de patrones.

Tal y como se expone durante el desarrollo de la sección 4.3 del texto, se define en primer lugar una RNA con una arquitectura y patrones de entrenamiento específicos diseñada para un objetivo concreto y, en base a los buenos resultados obtenidos, se decide realizar el diseño de otra RNA con un objetivo más ambicioso, la cual, presenta una arquitectura diferente y es entrenada con 3 conjuntos de patrones de entrenamiento diferentes. Estas redes consisten en:

RNA para patrones de presión en cámara fijos

* 4 capas, de las cuales 1 es de entrada, 1 de salida y 2 ocultas.
* 10 neuronas en la capa de salida (fijado por OUTPUT).
* 15 neuronas en la capa de entrada.
* 20 neuronas en la primera capa oculta.
* 15 neuronas en la segunda capa oculta.
* Funciones de transferencia de tipo “sigmoide”.
* Función de entrenamiento: *Levenberg-Marquardt.*
* Función de cálculo de error: *Mean sum of Square Error Regresion (MSEREG).*
* Pre y post tratamiento de la información, mediante un mapeo que convierte los valores de la información al intervalo [-1 1].
* Patrones de entrenamiento generados a partir de 5 niveles de daño para cada uno de los 10 cilindros para 13 niveles de carga diferentes. Esto supone 650 patrones de entrenamiento.
* No se considera la existencia de grietas de forma simultánea. Es decir, se considera que sólo falla un cilindro a la vez.
* Los patrones de presión en cámara para cada cilindro son fijos, por lo que no se considera el fallo en inyección de combustible de manera simultánea a la existencia de grietas en el cigüeñal.
* La red está diseñada para identificar qué muñequilla está dañada y en qué grado, con la consigna de no identificar falsos fallos en el resto de cilindros. Esta identificación se hace en base al máximo valor absoluto del error porcentual entre la rigidez simulada y la de referencia.

RNA para patrones de presión en cámara variables

* 2 capas, de las cuales 1 es de entrada y la otra es de salida.
* 10 neuronas en la capa de salida (fijado por OUTPUT).
* 15 neuronas en la capa de entrada.
* Funciones de transferencia de tipo “sigmoide”.
* Función de entrenamiento: *Resilient Backpropagation.*
* Función de cálculo de error: *Mean sum of Square Error Regresion (MSEREG).*
* Pre y post tratamiento de la información, mediante un mapeo que convierte los valores de la información al intervalo [-1 1].
* 3 conjuntos de patrones de entrenamiento diferentes, generados a partir de 59049 patrones de presión en cámara combinados con 3 niveles de daño para la muñequilla existente entre el cilindro 9 y el cilindro 10 para 2 niveles de carga diferentes. Esto supone 1 conjunto de 354.294 patrones de entrenamiento y dos conjuntos de 177.147 patrones de entrenamiento cada uno.
* Los diferentes patrones de presión en cámara tienen en consideración la existencia de defectos de inyección de combustible.
* La red está diseñada para la identificación de fallo en la muñequilla existente entre el cilindro 9 y el cilindro 10, con la consigna de no identificar falsos fallos en el resto de cilindros, considerando la existencia de fallos de inyección de combustible.

La RNA para patrones de presión en cámara fijosofrece uno resultados excepcionales en el objetivo para el que está diseñada. Sin embargo, de cara una implementación real de la herramienta de mantenimiento predictivo, el hecho de no considerar los fallos de inyección que puedan inducir a falsas detecciones de fallos, lo que hace que no sea una opción que se pueda llegar a implementar.

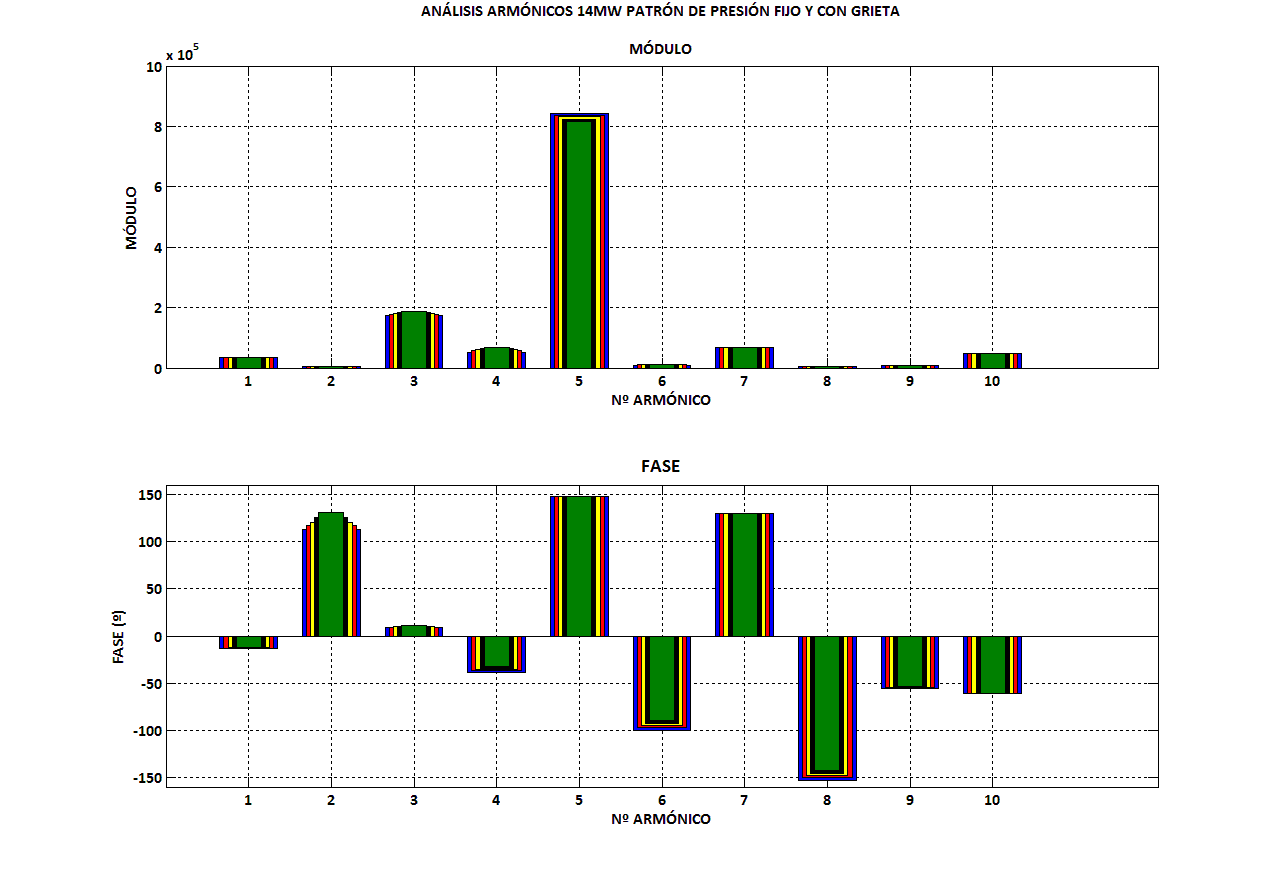
Por ello, se decide trabajar sobre un RNA para patrones de presión en cámara variables. Los resultados de esta RNA entrenada con los diferentes conjuntos de patrones de entrenamiento conducen a que en principio es necesario disponer de una RNA para cada nivel de carga diferenciado (8MW, 10MW, 12MW y 14MW) puesto que los resultados de la RNA entrenada con el conjunto de patrones de entrenamiento en el que se consideran los 2 niveles de carga distintos, si bien no ofrece unos resultados pobres, se alejan del alto nivel conseguido cuando se entrena para un nivel de carga en concreto, además de no tener una alta sensibilidad cuando es simulada con patrones de entrenamiento referentes a niveles de carga intermedios (10MW y 12MW).

## 5.4 Efectos de la existencia de grietas y fallos de inyección sobre la dinámica del sistema

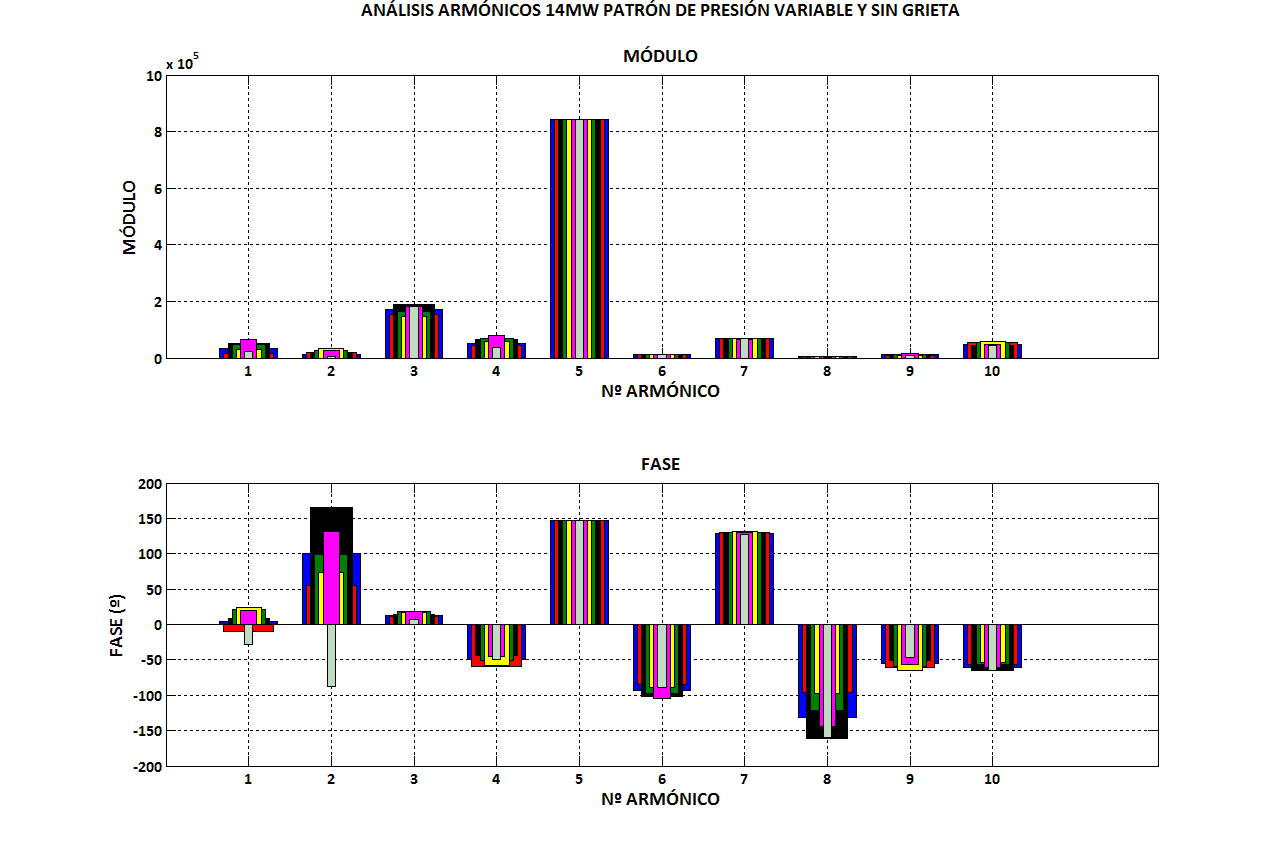
Este último apartado de la sección surge en consecuencia del continuo análisis de los resultados ofrecidos durante las etapas de generación de patrones y simulación de las RNA desarrolladas.

Durante dichas etapas se ha podido apreciar en los resultados que, la dinámica del sistema responde de una forma distinta si el sistema es afectado con grietas en el cigüeñal y patrones de presión fijos que cuando el sistema es afectado con patrones de presión variables y no existiendo grietas sobre el cigüeñal.

Concretamente, se observa mediante el análisis armónico de la señal de par generada por el sistema, que la fase de dicha señal de par viene principalmente controlada por la morfología de la curva de presión en cámara, es decir, por el patrón de presión que se desarrolle, mientras que el módulo de la señal viene controlado en mayor medida por la existencia de grietas en el cigüeñal, aunque no de forma tan concluyente como en el caso de la fase. En una primera aproximación, este resultado conduce a que ambos efectos están desacoplados en la dinámica resultante del sistema. A continuación se pueden ver unas representaciones del análisis armónico que se ha realizado para llegar a esta conclusión:



1. Análisis armónico de la señal de par con patrones de presión fijos y grietas en cilindro 9.



1. Análisis armónico de la señal de par con patrones de presión variables y sin grietas en cigüeñal.

### Módulo

En ambas figuras se puede apreciar que el módulo de cada uno de los armónicos cambia para cada caso considerado (cada color representa un caso distinto), sin embargo, en el caso en el que el sistema es afectado sólo por grietas en el cigüeñal (figura 68) se aprecia cómo varía el módulo del armónico 5, que es el que mayor peso tiene sobre la dinámica del sistema. En el caso en el que el sistema es afectado sólo por patrones de presión variables, el módulo del armónico 5 no sufre variación alguna (aunque si existe variación sobre el módulo de otros armónicos menos determinantes sobre la dinámica del sistema).

### Fase

Ocurre algo que comparte tendencia con el caso del módulo, es decir, todos los armónicos varían su fase para cada caso considerado. Sin embargo, en la fase si se aprecia que la variación de la misma está más controlada por la existencia de patrones de presión variables que por la existencia de grietas en el cigüeñal. Este resultado conduce a que, si se realiza un análisis armónico de la señal de par generada por el sistema y se atiende a la fase de la misma, se puede observar la existencia de defectos de inyección/combustión.

Evidentemente, estos resultados expuestos han de ser mucho más profundizados y desarrollados más exhaustivamente para confirmar si realmente ambos efectos están desacoplados o qué nivel de acoplamiento presentan. Sin embargo, un primer abordaje del tema como el que se ha mostrado aquí, da una idea inicial de por dónde se ha de trabajar en esta nueva línea de investigación abierta.

**Desarrollo Futuro**

*6.*

## 6.1 Introducción

Esta sección tiene el objetivo de poner de manifiesto las líneas de investigación que se han quedado abiertas tras la ejecución de este proyecto así como la futura y posible implementación que esta técnica desarrollada tiene.

La intención de poner de manifiesto las líneas de investigación que quedan abiertas no es otra que, una vez captado el interés en el desarrollo de este tema, se muestren las direcciones en las que han de ir dirigidos los esfuerzos de mejora de los resultados obtenidos en el período durante el que se ha desarrollado este proyecto.

Como futuro resultado de ese proceso de mejora y de refinamiento de los resultados obtenidos, se muestra la posible implementación que estos resultados tendrían sobre el sistema real.

## 6.2 Líneas de investigación

Una vez vistos los resultados iniciales obtenidos en la utilización de las RNA como herramienta de mantenimiento y diagnóstico predictivo de sistemas, resta realizar una labor de profundización en la elaboración de patrones de entrenamiento, puesto que se ha visto que los patrones de entrenamiento son el elemento clave a la hora de desarrollar una RNA.

En la búsqueda de un conjunto de patrones de entrenamiento que represente una casuística lo más completa y real posible, se han de desarrollar los siguientes modelos:

* Modelo analítico de inyección de combustible en cámara de combustión.
* Modelo analítico de combustión.

Con estos modelos se tendrá la capacidad de provocar fallos de inyección/combustión que afecten a las curvas de presión desarrolladas en las cámaras de combustión de los cilindros y consecuentemente, a la curva de par generada por el motor.

Por tanto, el trabajo en paralelo de estos dos modelos analíticos daría como resultado las curvas de presión de cada uno de los 10 cilindros que alimentarían al modelo analítico no lineal de la dinámica del sistema ya desarrollado, que junto a la consideración de grietas en las muñequillas del cigüeñal hará que se disponga de una base de datos de patrones de entrenamiento realista con la que entrenar la RNA para diagnosis predictiva de fallos en el cigüeñal del motor.

Por último y con todos los modelos preparados para la generación de patrones de entrenamiento realistas, se ha de entrenar una RNA para cada nivel de carga diferenciado, pues planteando la RNA global como un conjunto de 4 RNA diferentes, se conseguiría un resultado mucho más preciso además de un análisis más exhaustivo de la información adquirida del motor.

Además, tal y como se ha comentado en la sección anterior, gracias a los análisis que se han realizado durante las etapas de entrenamiento y simulación de las redes, se ha observado tanto el efecto que tiene los fallos de inyección/combustión como el efecto de la existencia de grietas sobre el cigüeñal del motor sobre el par generado por el sistema y por tanto, sobre la dinámica del mismo. Estos efectos apreciados conducen a que, en una primera aproximación, ambos efectos están desacoplados. Por tanto, he aquí una nueva línea de investigación: ¿están realmente desacoplados dichos efectos en la dinámica resultante?

## 6.3 Implementación

Una vez desarrollada una RNA que haya sido entrenada con una base de datos que contemple fallos en cigüeñal conjuntamente con fallos en inyección/combustión, se procedería a la implementación de la red en el sistema real. Este proceso es sencillo, puesto que sólo es necesario instalar en el equipo de monitorización del motor el software Matlab® e introducir tanto la RNA como la información necesaria para su uso.

Una vez volcada la RNA desarrollada en el equipo de adquisición, la señal de par que es adquirida por los dispositivos instalados a tal efecto, tras el proceso de filtrado y transformación, alimentaría la RNA, la cual, acto seguido mostraría el estado de las muñequillas del cigüeñal del motor. Con esto se conseguiría un hardware capaz de mostrar en tiempo real (realmente, según el tiempo empleado en la adquisición y filtrado de la señal) el estado de las muñequillas del cigüeñal.

La posibilidad de disponer de dicha información con un elevado grado de seguridad, resulta ser una situación muy beneficiosa, puesto que permitiría programar una parada del motor y realizar las reparaciones oportunas, las cuales, evitarían una rotura del motor. Una rotura en un motor de estas magnitudes y en las aplicaciones a las que suelen estar destinados este tipo de motores, puede considerarse como no admisible por el gran impacto económico que conllevaría, por lo que la herramienta de diagnosis predictiva de fallos en cigüeñal se hace aún más interesante.

**Bibliografía**

*7.*

1. “System Identification: Theory for the User”. Second Edition, by Lennart Ljung, Prentice Hall PTR, 1999.
2. “Internal Combustion Engines Fundamentals”. J.B. Heywood. Mc Graw Hill Book Company Inc., 1988.
3. “Motores de Combustión Interna Alternativos”. Tomás Sánchez Lencero, Antonio Muñoz Blanco y Francisco J. Jiménez Espadafor. Publicaciones Universidad de Sevilla (Escuela Técnica Superior de Ingeniería), 2009.
4. “Neural Network Toolbox™ User’s Guide”. Mark Hudson Beale, Martin T. Hagan y Howard B. Demuth. The Math Works Inc. 1992-2012.