



Trabajo Fin de Máster

Máster Universitario en Ingeniería Industrial

Técnicas de optimización Run-to-Run para dispositivos electromecánicos

Pattern search techniques for Run-to-Run optimization of electromechanical devices

Autor

Carlos Campos Martínez

Director

Edgar Ramírez Laboreo

Ponente

Carlos Sagüés Blázquez

Escuela de Ingeniería y Arquitectura
2016



DECLARACIÓN DE
AUTORÍA Y ORIGINALIDAD

(Este documento debe acompañar al Trabajo Fin de Grado (TFG)/Trabajo Fin de Máster (TFM) cuando sea depositado para su evaluación).

D./D^a. Carlos Campos Martinez

con nº de DNI 25196430E en aplicación de lo dispuesto en el art.

14 (Derechos de autor) del Acuerdo de 11 de septiembre de 2014, del Consejo de Gobierno, por el que se aprueba el Reglamento de los TFG y TFM de la Universidad de Zaragoza,

Declaro que el presente Trabajo de Fin de (Grado/Máster) Máster de Ingeniería Industrial _____, (Título del Trabajo)

Técnicas de optimización Run-to-Run para dispositivos electromecánicos

es de mi autoría y es original, no habiéndose utilizado fuente sin ser citada debidamente.

Zaragoza, 22/11/2016

Fdo: Carlos Campos Martinez

Trabajo Fin de Máster

Técnicas de optimización Run-to-Run para dispositivos electromecánicos

Resumen

El presente trabajo ha sido desarrollado dentro del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza. El objetivo principal del mismo ha sido el desarrollo y validación experimental sobre un microcontrolador de técnicas de control Run-to-Run (R2R) para sistemas electromecánicos, más concretamente un relé. Este tipo de control aprovecha el carácter repetitivo de este tipo de sistemas para reducir un determinado coste, en este caso la duración de los rebotes del mecanismo, de manera que se consiga alargar la vida útil del sistema y reducir el ruido generado en cada conmutación.

Para la consecución de los objetivos del trabajo, se ha procedido en un primer momento al estudio de distintos algoritmos de optimización, los cuales son la parte fundamental del control R2R. Más concretamente, debido a las particularidades de este tipo de sistemas, el trabajo se ha centrado en los algoritmos de optimización que no hacen uso de la derivada. Tras este estudio, se ha procedido a una selección de tres algoritmos de optimización. Al mismo tiempo, se han elegido un total de cuatro señales distintas para aplicar las estrategias de control.

Una vez elegidos tanto el algoritmo de optimización como las señales de control, se ha pasado al diseño de las distintas estrategias de control R2R. Estas han sido implementadas sobre un microcontrolador mediante programación en C, haciendo uso de una descripción mediante máquina de estados. Para poder evaluarlas experimentalmente se ha construido un circuito activador, el cual alimenta al relé y le permite llevar a cabo la conmutación. Por otro lado se han generado distintos programas de adquisición y análisis de datos en MatLab.

Tras la realización de las medidas experimentales y su posterior análisis, se ha podido comprobar el buen funcionamiento del control R2R aplicado a los relés, verificando que es posible integrar este tipo de control en un microcontrolador. Además, mediante el uso de nuevos algoritmos de optimización hasta antes no utilizados en este ámbito, se han podido obtener unos resultados que mejoran los obtenidos hasta ahora.

Agradecimientos

En estas pocas líneas me gustaría mostrar mi enorme agradecimiento hacia el Director y el Ponente de este trabajo, Edgar Ramírez Laboreo y Carlos Sagiés Blázquez, tanto por brindarme la oportunidad de realizar este trabajo en el seno del Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, como por toda la ayuda y consejos que he recibido por su parte, haciendo posible la realización de este Trabajo Final de Máster.

Ha sido un placer el poder contar con ellos en todo momento.

ÍNDICE

Capítulo 1. Introducción	7
1.1. Motivación	7
1.2. Estado del arte	9
1.3. Objetivos y tareas a desarrollar	10
1.4. Estructura de la memoria.....	11
Capítulo 2. Control R2R y algoritmos de optimización.....	13
2.1. Formulación del problema	13
2.2. Proceso de optimización.....	15
2.3. Control Run-to-Run.....	17
2.3.1. R2R con Búsqueda Compás	18
2.3.2. R2R con Búsqueda Multidireccional	20
2.3.3. R2R con Nelder-Mead	22
Capítulo 3. Implementación.....	25
3.1. Hardware utilizado	25
3.2. Señal de control.....	26
3.3. Programación microcontrolador.....	28
Capítulo 4. Evaluación experimental y análisis de resultados	31
4.1. Creación de mapa de costes	31
4.2. Procedimiento de medida	32
4.3. Comparativa experimental de los algoritmos de optimización	33
4.4. Comparativa experimental de las señales de control.....	35
4.5. Análisis de los resultados experimentales	36
Capítulo 5. Conclusiones y Líneas futuras.....	38
5.1. Conclusiones	38
5.2. Líneas futuras	39
Referencias.....	41
Índice de figuras	43
Anexo 1. Medidas Experimentales	47
A.1.1. Comparativa algoritmos de optimización.....	47
A.1.1.1 Evolución del coste	48
A.1.1.2 Mejora de coste	49
A.1.1.3 Velocidad de reducción de coste.....	52
A.1.2. Comparativa señales de control.....	53
A.1.2.1 Evolución del coste	54

A.1.2.2	Mejora de coste	56
A.1.2.3	Velocidad de reducción de coste	57
Anexo 2.	Implementación del Microcontrolador	59
A.2.1	Descripción mediante máquina de estados.....	59
A.2.1.1	Máquina de estados para Búsqueda Compás	59
A.2.1.2	Máquina de estados para Búsqueda Multidireccional	60
A.2.1.3	Máquina de estados para Nelder-Mead	61

Capítulo 1. Introducción

1.1. Motivación

Este trabajo nace a partir de la realización de unas prácticas en el Departamento de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, durante el verano de 2016, centradas en la aplicación de estrategias de control Run-to-Run (R2R) para el control de sistemas electromecánicos. El estudio de estos dispositivos es un tema recurrente dentro del departamento, debido a su estrecha relación con la Empresa BSH Electrodomésticos España y el interés mostrado por esta acerca de su control.

Un sistema electromecánico, como su nombre indica, es aquel que cuenta con una parte electromagnética y otra mecánica o móvil. Este conjunto tan amplio engloba desde cualquier dispositivo que cuente con un motor o generador eléctrico, hasta un altavoz o un relé. Este trabajo se centrará en el estudio de estos últimos, los relés.

Un relé se puede definir como un interruptor controlado a través de una bobina, a la que aplicándole niveles relativamente bajos de tensión, controla la apertura o cierre de un circuito secundario, con unos requerimientos mucho mayores de tensión. La bobina al aplicarle una tensión constituye un electroimán, el cual controla el movimiento de la parte mecánica del sistema. Las características de esta señal de tensión determinarán el movimiento de la parte móvil.

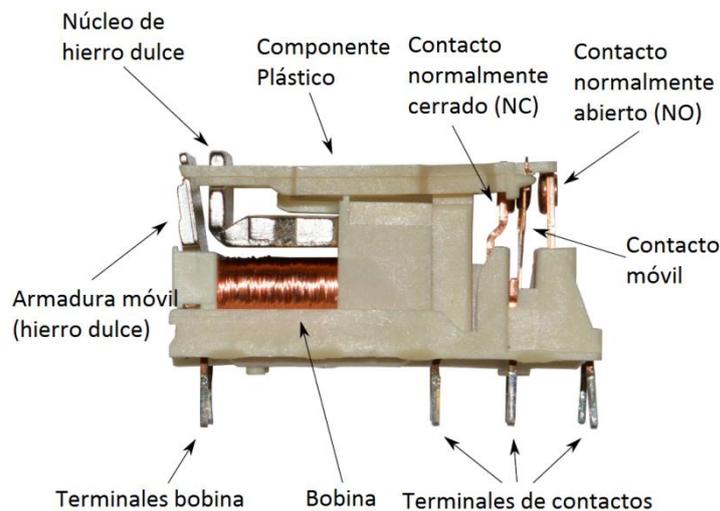


Figura 1. Representación y descripción elemental de un relé electromecánico.

Estos dispositivos cuentan con casi doscientos años de historia, y su estructura y funcionamiento no han variado prácticamente desde sus orígenes. Sin embargo, presentan ciertas ventajas respecto a sus principales rivales, los semiconductores, los cuales les hacen propicios para determinados usos como pueden ser los

electrodomésticos, equipos de potencia o para el control de motores y generadores eléctricos, razones por las cuales siguen siendo ampliamente utilizados:

- La conmutación del relé disipa una cantidad de energía mucho menor que la disipada por los dispositivos semiconductores, no haciendo necesario el uso de elementos radiadores.
- Aísla y separa físicamente los dos terminales del circuito secundario, lo cual confiere una mayor seguridad.
- Son capaces de controlar circuitos que funcionan con grandes tensiones y potencias.

Características	Dispositivos Electromecánicos	Dispositivos Semiconductores
Pérdidas Energéticas	<1W por dispositivo [1]	~40W por dispositivo [1]
Tiempo de conmutación	~1 ms controlado	~100 ns
Dirección Corriente	Bidireccional	Según dispositivo
Vida útil	Limitada por componente mecánicos	No limitada por componentes mecánicos
Frecuencia máxima de conmutación	~20 Hz	~100 kHz
Ruido acústico	Sí	No
Coste	Relé Tyco RZ ~0.4 € + Transistor [2]	IGBT+Driver ~1.2 €

Tabla 1. Comparativa de características de dispositivos electromecánicos y dispositivos semiconductores.

Por otro lado, existen ciertas desventajas de las que hay que ser conscientes:

- El relé presenta una vida limitada debido al desgaste mecánico que se produce en cada conmutación. Existe por lo tanto un número de conmutaciones a partir del cual su comportamiento deja de ser el adecuado. Esta vida útil depende de cómo se hayan llevado a cabo las conmutaciones a lo largo de su periodo útil.
- La conmutación del relé no es instantánea, ya que al contar con partes mecánicas móviles, existen ciertas inercias que ralentizan la acción. Esto hace que la frecuencia de trabajo no sobrepase en ningún caso los 10 Hz, limitando su aplicabilidad.
- Al realizarse la conmutación, las partes móviles se golpean unas con otras, generando cierto ruido, resultando molesto en ciertas aplicaciones.

El motivo principal por el cual se ha llevado a cabo este trabajo ha sido intentar reducir todos estos inconvenientes que presentan los relés frente a los dispositivos

semiconductores, mejorando su comportamiento en las aplicaciones en las que se emplean actualmente.

Para reducir todos estos problemas, hay que centrarse en la causa de cada uno de ellos. En este caso, todos estos problemas vienen causados principalmente por el hecho de que al realizarse la conmutación y no estar el sistema controlado, la parte móvil golpea de manera repetitiva la parte fija del sistema. Estos rebotes son por lo tanto los causantes del desgaste mecánico de los distintos componentes, así como del aumento del tiempo necesario para cada conmutación y del ruido generado por estos dispositivos.

Con el fin de evitar o paliar estos rebotes, se pueden llevar a cabo distintos tipos de control sobre estos sistemas.

1.2. Estado del arte

El afán por reducir este tipo de problemas asociados a los relés, y por lo tanto estudiar su control, no es algo reciente. Como primer paso para el control de estos sistemas, ya en la década de los años 60 se llevaron a cabo distintos trabajos [3] para su modelado, los cuales permitieron predecir la dinámica del sistema. Estos modelos eran relativamente sencillos y estaban basados en análisis energéticos del sistema a través de un modelo muelle-masa. La necesidad de conocer mejor su comportamiento, y de poder utilizar nuevos y mejores métodos de control ha hecho que la búsqueda de mejores modelos haya continuado hasta la actualidad. Ejemplo de ello son [4] donde se introduce el efecto que tiene el arco eléctrico en el comportamiento del sistema, [5] donde se hace uso del Método de Elementos Finitos para modelar el sistema, así como otros trabajos, [6], [7], [8] y [9], en los que para el modelado se hace uso de teoría de circuitos magnéticos, combinada con la descripción mecánica del sistema.

Por otro lado también se ha tratado uno de los mayores problemas a los que se enfrenta el control de este tipo de sistemas, que es la ausencia de sensores de posición [10] con el que poder retroalimentar el sistema.

El estudio de las distintas metodologías de control aplicadas a este tipo de sistemas u otros electromecánicos, como las electroválvulas, se ha realizado de forma paralela al modelado. Así, se han desarrollado técnicas de control óptimo [11], de control robusto [12], o control borroso [13] entre otras.

Sin embargo, todos estos esfuerzos por modelar y controlar este tipo de sistemas, no han obtenido tan buenos resultados como se esperaba. La principal razón radica en la diferencia entre el modelo real y el calculado, debido a la complejidad del sistema, el cual cuenta con fuertes no linealidades. No hay que olvidar que se trata de dos sistemas acoplados, uno electromagnético y otro mecánico, cuyos comportamientos se interfieren entre sí.

Debido a todo lo anteriormente comentado, se ha decidido optar por nuevos tipos de control hasta ahora inexplorados para el caso de los sistemas electromecánicos. En un

reciente artículo [1], se ha propuesto un nuevo método de control Run-to-Run (R2R) en el que se aprovecha el carácter repetitivo de estos sistemas, los cuales ejecutan dos únicas operaciones, el cierre y la apertura, de manera continuada. Si bien este tipo de control no es algo novedoso, sí que lo es su aplicación para sistemas electromagnéticos.

Este tipo de control busca el minimizar una función de coste, que en este caso es la duración de los rebotes, sin hacer uso del modelo del sistema. Como en cualquier otro proceso de optimización, en el control R2R la parte fundamental es el algoritmo de optimización. Este decide cómo se va variando la señal de control para ir reduciendo el coste iteración tras iteración. Mediante la modificación de este algoritmo de optimización se obtendrán distintas estrategias de control Run-to-Run, cuyos comportamientos diferirán entre ellos.

Este trabajo se centra en el estudio y desarrollo de las estrategias de control R2R para dispositivos electromecánicos. El punto de partida de este trabajo es [1], a partir del cual se desarrollan nuevas estrategias de control R2R que serán implementadas para poderlas probar experimentalmente.

1.3. Objetivos y tareas a desarrollar

Antes de iniciar el trabajo, se proponen los siguientes objetivos, por orden decreciente de importancia:

- Demostrar la viabilidad del control R2R para sistemas electromecánicos, mediante su implementación a través de un microcontrolador. Aunque este tipo de control sí que ha sido probado experimentalmente [1] con buenos resultados, no se ha podido probar su viabilidad ya que la tarea de control ha sido realizada mediante una computadora, y la adquisición mediante un osciloscopio. Incluir estos equipamientos en el producto final supondría aumentar tanto su volumen como su coste a niveles inadmisibles. En este trabajo el objetivo principal es hacer uso de únicamente un microcontrolador tanto para la tarea de control como para la de adquisición de datos, lo cual permitiría corroborar la viabilidad del control R2R para sistemas electromecánicos.
- Estudio de nuevos algoritmos de optimización hasta ahora no utilizados en este tipo de control. Diseño y estudio de nuevas estrategias de control R2R basadas en estos nuevos algoritmos de optimización.
- Estudio y comparación de los distintos controles R2R, tanto de los ya existentes como de los desarrollados durante este trabajo.
- Diseño, estudio y comparación tanto de señales de control ya utilizadas [1] como de nuevas señales de control.

Para alcanzar los objetivos citados, se han propuesto las siguientes tareas, en orden cronológico:

- Búsqueda bibliográfica y estudio de los distintos algoritmos de optimización Búsqueda Directa, y más concretamente los denominados Búsqueda de Patrón (*Pattern-Search*).
- Estudio de la viabilidad de la aplicación de algoritmos Búsqueda Directa al control R2R, así como las modificaciones necesarias para adaptarlos a este tipo de control.
- Diseño del circuito de activación del sistema, además de otros circuitos auxiliares para la captación de distintas señales.
- Selección y programación del microcontrolador para la implementación de los distintos algoritmos, la adquisición de datos, así como la comunicación con el PC.
- Adquisición y comparación de los distintos resultados, y su posterior análisis.

1.4. Estructura de la memoria

Esta memoria se va a estructurar en los siguientes capítulos:

- El capítulo 2, donde se expone la formulación matemática del problema de optimización que se está abordando. Presentación de las distintas herramientas para resolver este problema y su integración en el control R2R.
- Un siguiente capítulo, el 3, dedicado a la implementación del control R2R sobre un microcontrolador y a los demás componentes necesarios para la evaluación experimental del control.
- El capítulo 4, donde se presentan las medidas experimentales más relevantes para los distintos controles y el análisis de estos resultados.
- Un último capítulo, el 5, con las conclusiones de todo el trabajo y posibles líneas futuras de desarrollo de investigación.

Además, se han incluido dos anexos:

- El anexo 1, donde se muestran un mayor número de resultados experimentales que los expuestos en el capítulo 4.
- Y un segundo anexo, el 2, en el que se describe el comportamiento de los distintos algoritmos en forma de máquina de estados, necesarios para la programación.

Capítulo 2. Control R2R y algoritmos de optimización

En este capítulo se presenta el problema a resolver, que se formula según un problema de optimización en el que se busca minimizar un coste. Tras esta formulación, se discute qué algoritmos de optimización son los más adecuados para resolver el problema teniendo en cuenta las características del mismo. Por último, una vez escogidos los algoritmos, se trata la integración de estos dentro del esquema de control.

2.1. Formulación del problema

El control de los relés electromecánicos tiene como principal objetivo la reducción del tiempo de conmutación, reducción del ruido generado y la reducción del desgaste del mecanismo. Como ya se ha dicho, estos tres fenómenos vienen asociados a la existencia de rebotes entre la parte móvil y fija del mecanismo, siendo la duración de estos rebotes muy fácil de medir. Una reducción de la duración de estos rebotes implicaría por lo tanto un mejor comportamiento del sistema, teniendo un menor tiempo de conmutación, un menor ruido y aumentando la vida del mismo. Para controlar el comportamiento del sistema únicamente se puede actuar sobre la tensión de alimentación de la bobina del sistema electromecánico. Aplicando una u otra señal, el relé conmutará de una u otra manera. La señal u que se aplique al sistema se parametriza a través de un vector \bar{v} , de valor fijado antes de la conmutación y cuya dimensión n determina los grados de libertad de la señal u .

El problema que se está afrontando en este trabajo es por lo tanto el de minimizar un coste escalar J asociado al tiempo de rebotes, modificando únicamente los parámetros de la señal de entrada.

$$\min_{\bar{v} \in \Omega} J$$

Donde:

$$J = \phi(\bar{y}(t)) = \chi_v(\bar{v})$$

$$\Omega \equiv \{\bar{x} \in \mathbb{R}^n \mid l_i < x_i < h_i \forall i = 1 \dots n\}$$

$$\bar{l}, \bar{h} \in \mathbb{R}^n$$

Sujeto a:

$$\dot{\bar{x}}(t) = F(\bar{x}(t), u(\bar{v})) + \bar{n}(t)$$

$$\bar{y}(t) = H(\bar{x}(t)) + \bar{n}_m(t)$$

Siendo:

- F es la función que define la dinámica del sistema y H es la función de la salida del sistema.
- $\bar{x}(t) \in \mathbb{R}^s$ el vector de estado del sistema, donde s representa el número de estados.
- $\bar{y}(t) \in \mathbb{R}^e$, el vector formado por las e salidas del sistema. Para el sistema que se trata en este trabajo, el relé, estas salidas son las señales NC y NO representadas en Figura 2, así como la corriente que circula por la bobina, y la tensión que se le aplica a la misma, representadas en Figura 3. También se podría medir el ruido para tenerlo en cuenta a la hora de calcular el coste.

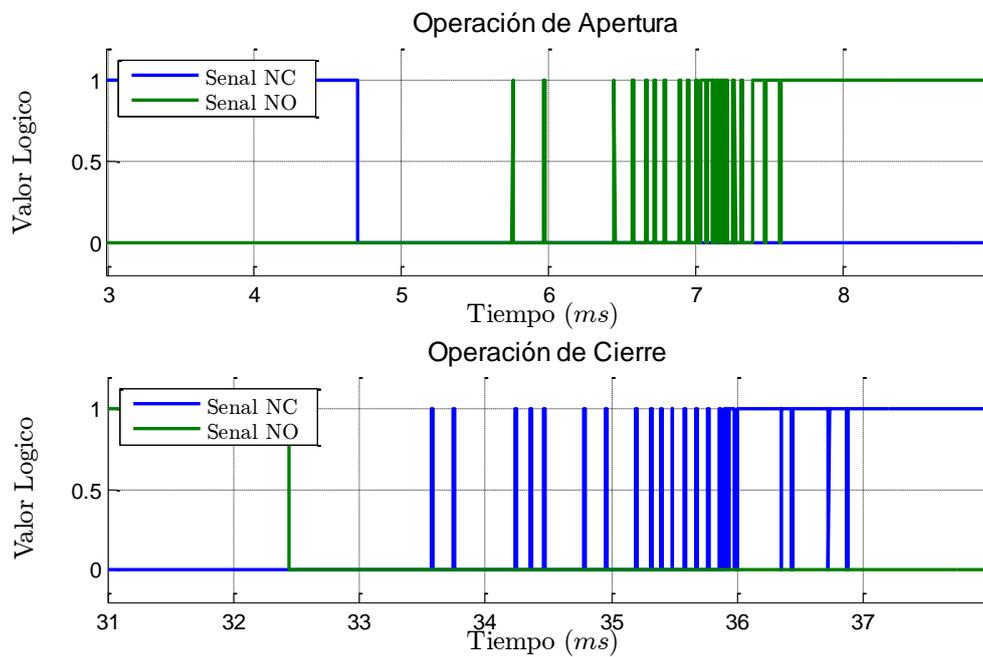


Figura 2. Representación de las señales NC y NO (cuyo valor a 1 indica circuito cerrado en los dos casos) las cuales son las únicas salidas del sistema si no se considera el ruido. Ambas son utilizadas para calcular la duración de los rebotes (tiempo entre el primer y último rebote) tanto en la apertura como en el cierre.

- $\phi: \mathbb{R}^e \rightarrow \mathbb{R}$ es la función que relaciona el coste escalar con la salida del sistema. En este caso será el que relacione
- $\chi_v: \mathbb{R}^n \rightarrow \mathbb{R}$ es la función que relaciona el coste escalar con los parámetros de la señal de entrada. El subíndice v enfatiza que la definición de esta función depende el tipo de señal que se aplique al sistema.
- \bar{l} y \bar{h} son los límites inferior y superior del espacio solución. Estos límites pueden venir fijados por limitaciones físicas de energía, así como por la propia naturaleza de la señal. Con este tipo de restricciones el espacio de búsqueda adopta la forma de un hipercubo.

- $\bar{n}(t)$ y $\bar{n}_m(t)$ las perturbaciones que afectan tanto al sistema como a la medida de la salida.

El problema queda por lo tanto reducido a un problema de optimización, en este caso de minimización con restricciones lineales en el espacio solución.

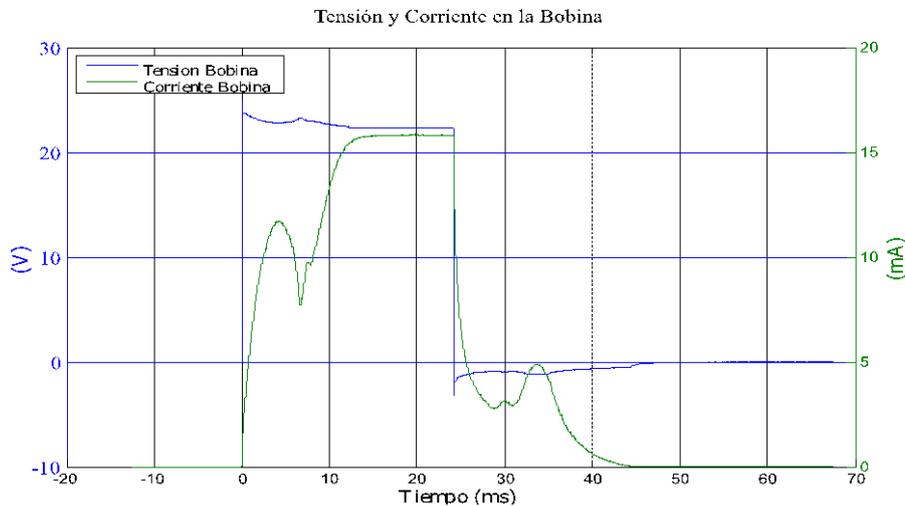


Figura 3. Representación de la única entrada del sistema, tensión de alimentación de la bobina, y de la intensidad medida en la bobina a través de una resistencia. La intensidad se podría utilizar como salida del sistema para calcular el coste, sin embargo aquí únicamente se va a tomar en consideración la duración de los rebotes.

2.2. Proceso de optimización

Una vez que se ha formulado el problema, es necesario encontrar la herramienta que permita resolverlo. Dentro de los distintos métodos de optimización se distinguen dos grandes grupos:

- Métodos iterativos los cuales hacen uso del vector gradiente y de la matriz hessiana, como podrían ser los métodos de Newton, de cuasi-Newton o el de diferencias finitas. Para el cálculo de las derivadas, se hace uso de la expresión analítica de la función de coste o se aproxima aplicando la definición infinitesimal de la derivada.
- Métodos iterativos libres de derivadas, los cuales no hacen uso del gradiente o una aproximación del mismo, sino que van evaluando la función de coste en distintos puntos buscando el mínimo de la función. En este grupo destacan los denominados métodos directos, los cuales a través de una serie de movimientos exploratorios, van convergiendo hacia el mínimo. Estos métodos engloban entre otros el original de Hook y Jeeves [14], Búsqueda Multidireccional [15], Búsqueda Compás, así como el EVOPT [16].

Las características de la función de coste de los sistemas electromecánicos, las cuales no son suaves, así como la presencia de ruido y el parcial o total desconocimiento del

sistema, hacen únicamente viable el uso de métodos libres de derivadas. En el caso de que se usara una estimación de la derivada, esta no sería fiable debido a las discontinuidades de la función de coste y la presencia de ruido.

La principal desventaja por la que los métodos libres de derivada no son tan ampliamente utilizados es la dificultad para demostrar la convergencia así como su velocidad de convergencia. Sin embargo, en los últimos años distintos estudios vienen avalando la convergencia de estos métodos [15] bajo ciertas condiciones, e incluso en el caso de que las evaluaciones de la función no sean exactas [17], como es nuestro caso.

Como se ha dicho, estos algoritmos trabajan de manera iterativa, evaluando la función de coste en distintos puntos, por lo que sería impensable el aplicar estas técnicas de optimización a sistemas que no trabajen de manera repetitiva. Se podría por lo tanto volver a enunciar el problema, teniendo en cuenta que para alcanzar el mínimo se debe proceder de manera iterativa:

$$\min_{\bar{v}_k \in \Omega} \left(\lim_{k \rightarrow \infty} \phi(y_k(t), u(\bar{v}_k)) \right)$$

Por lo tanto, el algoritmo de optimización no busca el minimizar el coste en cada iteración, sino únicamente el minimizarlo en el infinito. Sin embargo, en el caso de aplicar estos algoritmos de optimización al control R2R, también se quiere al mismo tiempo que el coste asociado a la trayectoria de \bar{v}_k sea lo menor posible.

Los algoritmos de optimización de Búsqueda Directa o *Pattern-Search*, de manera genérica, se pueden describir de la siguiente manera:

```

1:   Inicializar  $x_{opt}^0$  y  $\alpha$ ;
2:    $f_{opt}^0 = f(x_{opt}^0)$ ;
3:    $k = 1$ ;
4:   while (no Stop) do {
5:       // Se recorre el patrón en torno al óptimo, y se guarda
6:       // el mejor punto de la malla
7:        $f_{opt}^k = \min_{d \in D(x_k)} f(x_{opt}^{k-1} + \alpha \cdot d)$ ;
8:        $x_{opt}^k = \operatorname{argmin}_{d \in D(x_k)} f(x_{opt}^{k-1} + \alpha \cdot d)$ ;
9:       if ( $f_{opt}^k > f_{opt}^{k-1}$ ) then {
10:          // El mejor punto es  $x_{opt}^{k-1}$ , se considera búsqueda
11:          // fallida y se comprime el patrón
12:           $x_{opt}^k = x_{opt}^{k-1}$ ;
13:           $\alpha = \epsilon \cdot \alpha$ ;
14:       }
15:       else {
16:          // La búsqueda ha sido fructífera, se centra la
17:          // malla en el óptimo  $x_{opt}^k$  (Se puede agrandar la
18:          // malla si se quiere, nunca contraer)
19:           $\alpha = (1/\epsilon) \cdot \alpha$ ;

```

```

20:         }
21:         k++;
22:     }
    
```

Siendo:

- $D(x_k)$ el conjunto que define la malla en la que se evalúa la función, y cuyo valor depende del método de búsqueda que se utilice. Este conjunto se puede expresar en forma de una matriz D de dimensión $n \times m$ donde cada columna es un elemento del conjunto $D(x_k)$. No tiene por qué ser constante en todas las iteraciones y puede ir variando según los movimientos exploratorios de cada algoritmo. Para asegurar la convergencia se ha demostrado en [18] que el conjunto $D(x_k)$ debe ser una base positiva generadora del espacio de búsqueda \mathbb{R}^n .
- α es el tamaño de malla, cuyo valor va disminuyendo conforme nos acercamos al mínimo.
- $\epsilon < 1$ es el parámetro de contracción, el cual fija la velocidad a la que se va disminuyendo o aumentando el tamaño de la malla. Cuanto más cercano a 1 menor será su velocidad, pero más fácil será encontrar el mínimo. Para garantizar la convergencia este debe ser un número racional [18].

2.3. Control Run-to-Run

Como ya se ha dicho, en el caso del relé, el sistema realiza dos operaciones de manera indefinida, que son la apertura y el cierre del circuito secundario. Por lo que en realidad se está llevando a cabo el control de dos procesos similares pero totalmente independientes.

El esquema global de un control Run-to-Run queda representado en la Figura 4, donde ψ_r representa el valor de referencia que queremos alcanzar en la variable de interés. En este caso esta variable sería la duración de los rebotes, que queremos que alcance un valor de referencia nulo. Por otro lado, ψ_i representa el valor actual de esta variable, por lo que el coste se podría definir mediante una norma p como $J_i = \|\psi_r - \psi_i\|_p$. En el bloque R2R, a partir del coste actual y de todos los anteriores, se calcula un vector de decisión v^i haciendo uso de un algoritmo de optimización. En este caso las componentes del vector de v^i serían los distintos parámetros que definen cada señal de control. A partir de estos parámetros, en el bloque V , se calcula la señal temporal $u^i(t)$ que se aplica al sistema, siendo aquí la tensión de alimentación de la bobina. Una vez aplicada esta señal, el estado del sistema $x^i(t)$ va a evolucionar, por lo que la salida $y^i(t)$ que se lee también evolucionará según H . El bloque integrador permite tener en cuenta todos los valores de la salida que se tienen a lo largo de la conmutación. Una vez

finalizada la conmutación ($t = t_f^i$) a través de todas las salidas leídas, se calcula el valor de la variable de interés ψ_i , y se procede de igual modo iterativamente.

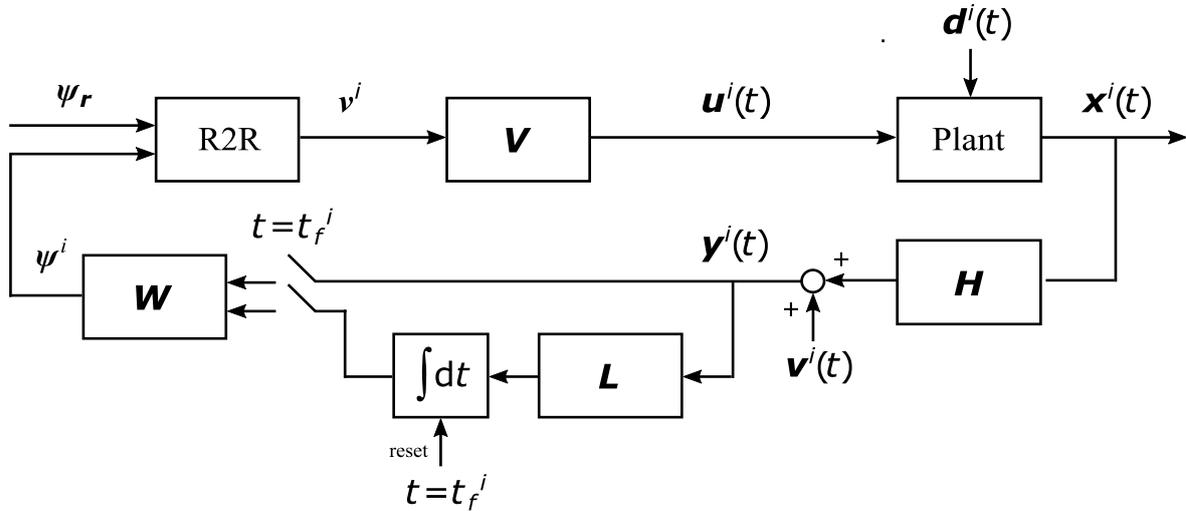


Figura 4. Esquema general de un control Run-to-Run [1].

A diferencia de los problemas de optimización, el objetivo de este control no es únicamente encontrar el mínimo de la función de coste sino también ejecutar la acción en ese punto. Además, en este caso, el coste responde al comportamiento de un sistema real físico, el cual puede tener un comportamiento, y por lo tanto un mínimo, variable con el tiempo. Por todo ello el control R2R diferirá de los algoritmos de optimización clásicos.

En esta sección se van a desarrollar tres métodos de control R2R con los algoritmos de optimización sin cálculo de derivadas: Búsqueda Compás, Búsqueda Multidireccional y Nelder-Mead.

2.3.1. R2R con Búsqueda Compás

Este método ya ha sido implementado [1], ya que supone uno de los métodos de Búsqueda Directa más sencillos. En este algoritmo se utiliza una malla $D(x_k)$, de $2n + 1$ puntos que no varía entre iteraciones. Esta queda definida a través de la matriz D , de dimensión $(n) \times (2n + 1)$, que para el caso de \mathbb{R}^2 es:

$$D = \begin{bmatrix} 0 & +1 & 0 & -1 & 0 \\ 0 & 0 & +1 & 0 & -1 \end{bmatrix}$$

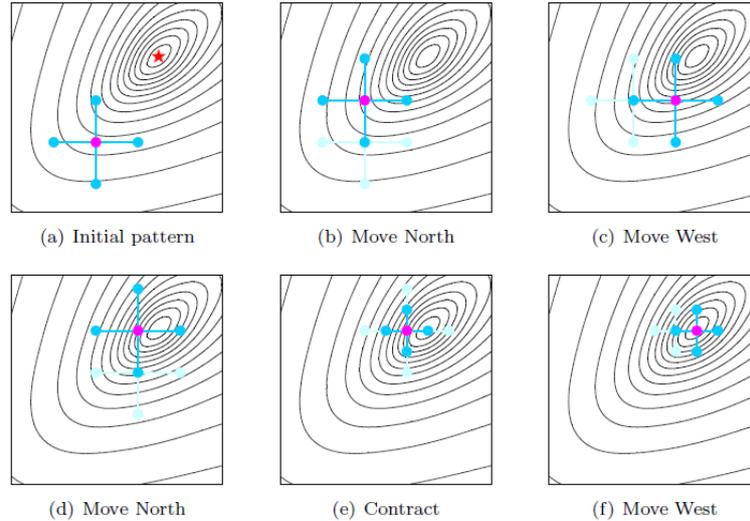


Figura 5. Secuencia de optimización Búsqueda Compás. Figura obtenida de [19].

En este método únicamente se guarda entre cada iteración el mejor punto, aquel que tiene un menor coste. En la siguiente iteración, este punto será el centro del patrón, y se probarán $2n$ nuevos puntos en su entorno. Se utilizará $Col_i(D)$ para referirse a la columna i de la matriz D . El algoritmo de control R2R sigue la siguiente estructura:

```

1:   Inicializar  $x_{opt}^0$  y  $\alpha$ ;
2:    $f_{opt}^0 = f(x_{opt}^0)$ ;
3:    $k = 1$ ;
4:   While(1) {
5:       // Se vuelve a medir el óptimo para asegurarse de que no
6:       // ha cambiado
7:        $f_{opt}^{k-1} = f(x_{opt}^{k-1})$ ;
8:       // Se recorre el patrón en torno al óptimo
9:       for (i=1;i<=2n+1;i++) {
10:           $f_i^k = f(x_{opt}^{k-1} + \alpha \cdot Col_i(D))$ 
11:       }
12:       // El mejor punto de la malla
13:        $f_{opt}^k = \min_{d \in D(x_k)} f(x_{opt}^{k-1} + \alpha \cdot d)$ ;
14:        $x_{opt}^k = \operatorname{argmin}_{d \in D(x_k)} f(x_{opt}^{k-1} + \alpha \cdot d)$ ;
15:       if ( $f_{opt}^k < (f_{opt}^{k-1} \cdot p)$ ) then {
16:           $\alpha = \alpha / \epsilon$ ; //Se agranda el patrón
17:       }
18:       else {
19:           $\alpha = \alpha \cdot \epsilon$ ; // Se empequeñece el patrón
20:          // Se guarda como óptimo el anterior
21:           $f_{opt}^k = f_{opt}^{k-1}$ ;
22:           $x_{opt}^k = x_{opt}^{k-1}$ ;
23:       }
24:        $k++$ ;
    }
```

A la hora de comparar el mejor resultado de una iteración con otra, se ha utilizado un parámetro $p > 0$, que se fija de manera heurística a 0.8, y que tiene como objetivo eliminar el efecto que tiene el ruido sobre las medidas del coste. Con ello se busca aumentar la certeza de que el mínimo que se ha encontrado sea realmente el menor y evitar así la toma de malas decisiones. Dicho en otras palabras, se prefiere permanecer en un punto, cuyo comportamiento se conoce bien, que moverse a otro que parezca mejor pero cuyo comportamiento apenas se conoce.

2.3.2. R2R con Búsqueda Multidireccional

Este método desarrollado en [15] y en el cual también se demuestra su convergencia, pertenece a la familia de los Búsqueda de Patrón, al igual que el de Búsqueda Compás. En este caso, la malla o patrón que se utiliza no permanece constante sino que depende de resultados previos y varía entre iteraciones.

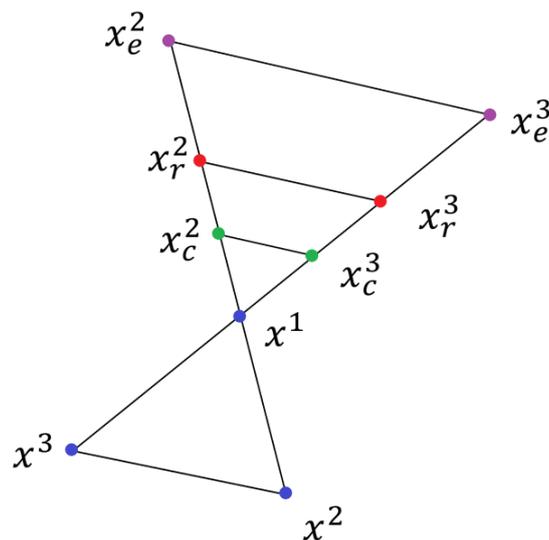


Figura 6. Secuencia de funcionamiento del R2R basado en Búsqueda Multidireccional. En azul se han representado los puntos del simplex original, en rojo, los del simplex rotado en torno al mejor punto, en verde esos mismos puntos contraídos, y en morado expandidos.

Este método se basa en el uso de una estructura geométrica de tipo simplex ($n + 1$ puntos en el espacio de dimensión n), a la cual se le van realizando operaciones de rotación, contracción y expansión tal y como se ve en Figura 6. En este caso, de una iteración a otra solo se conserva el mejor punto.

El algoritmo del control R2R basado en Búsqueda Multidireccional sigue la siguiente estructura:

- 1: Inicializar simplex original ($x^1 \dots x^{n+1}$);
- 2: for ($i=1; i \leq n+1; i++$) {
- 3: $f^i = f(x^i)$; // Evaluamos en cada punto del simplex original

```

4:   }
5:   Ordenar  $f^i$  y  $x^i$  de forma:  $f^1 < f^2 < \dots < f^{n+1}$ ;
6:    $k = 1$ ;
7:   While(1) {
8:       // Se vuelve a medir el óptimo para asegurarse de que no
9:       // ha cambiado
10:       $f^1 = f(x^1)$ ;
11:      // Se rota el simplex en torno al óptimo  $x^1$ 
12:      for (i=2;i<=n+1;i++) {
13:           $x_r^i = x^1 + (x^1 - x^i)$ ;
14:           $f_r^i = f(x_r^i)$ ; // Se evalúa el punto rotado
15:      }
16:      // El mejor punto del simplex rotado
17:       $f_r^{opt} = \min_{i=2\dots n+1} f(x_r^i)$ ;
18:       $x_r^{opt} = \operatorname{argmin}_{i=2\dots n+1} f(x_r^i)$ ;
19:      // Si el rotado es mejor que el original se expande
20:      if ( $f_r^{opt} < (f^1 \cdot p)$ ) then {
21:          for (i=2;i<=n+1;i++) {
22:               $x_e^i = x^1 + \mu(x_r^i - x^1)$  con  $\mu \in [1, \infty)$ ;
23:               $f_e^i = f(x_e^i)$ ; // Se evalúa el punto expandido
24:          }
25:          // El mejor punto del simplex expandido
26:           $f_e^{opt} = \min_{i=2\dots n+1} f(x_e^i)$ ;
27:           $x_e^{opt} = \operatorname{argmin}_{i=2\dots n+1} f(x_e^i)$ ;
28:          if ( $f_e^{opt} < f_r^{opt}$ ) then {
29:              // Se guarda el simplex expandido
30:              for (i=2;i<=n+1;i++)  $x^i = x_e^i$ ;  $f^i = f_e^i$ ;
31:          } else {
32:              // Se guarda el simplex rotado
33:              for (i=2;i<=n+1;i++)  $x^i = x_r^i$ ;  $f^i = f_r^i$ ;
34:          }
35:      }
36:      else {
37:          // Se contrae el simplex y se guarda para la
38:          // siguiente iteración
39:          for (i=2;i<=n+1;i++) {
40:               $x^i = x^1 - \theta(x^1 - x_r^i)$  con  $\theta \in [0,1)$ ;
41:               $f^i = f(x^i)$ ; // Se evalúa el punto contraído
42:          }
43:      }
44:      // Se ordenan los puntos del simplex por coste
45:      Ordenar  $f^i$  y  $x^i$  de forma:  $f^1 < f^2 < \dots < f^{n+1}$ ;
46:       $k ++$ ;
47:   }

```

2.3.3. R2R con Nelder-Mead

Este método fue uno de los primeros dentro de los de Búsqueda Directa. La principal diferencia respecto a los otros dos antes presentados, es que de una iteración a otra, este algoritmo únicamente cambia uno de los puntos de evaluación, salvo que se dé la situación denominada *Shrink*, tal y como se muestra en Figura 7. Por lo tanto, este algoritmo no hace uso de un patrón. Con este algoritmo, si al realizar la medida se obtiene un valor que no es real, ya sea por el ruido, la aleatoriedad del sistema o cualquier otra razón, únicamente se va a cambiar uno de los puntos, por lo que el efecto de esta falsa medida es menor.

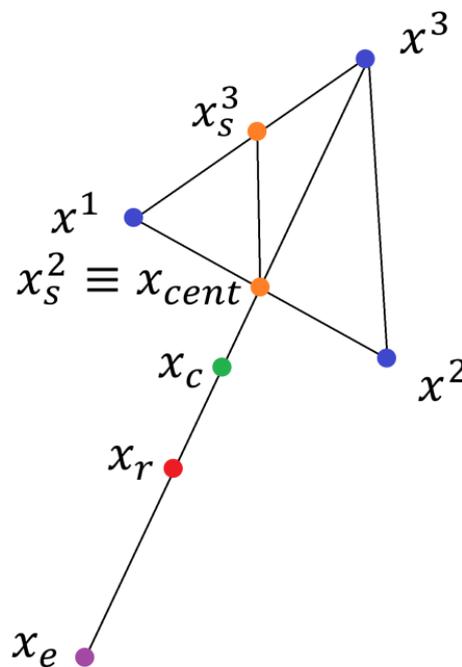


Figura 7. Secuencia de funcionamiento del R2R con Nelder-Mead. En azul se ha representado el simplex original, en rojo el peor punto rotado respecto al punto x_{cent} , en verde el punto rotado contraído, en morado el rotado y expandido, y en naranja, el simplex contraído respecto del mejor punto x_1

Además, al conservar entre iteración e iteración todos los puntos salvo uno, se pueden ir realizando nuevas medidas en estos puntos que se repiten iteración tras iteración, para aumentar la confianza del valor obtenido en estos puntos. Para ello, en este control R2R con Nelder-Mead se va a utilizar un coeficiente de olvido α que sirve para actualizar el valor del coste en los puntos que con anterioridad ya se ha realizado alguna medida.

Con esta repetición de las medidas se está disminuyendo la velocidad de convergencia del método, pero por otro lado se está disminuyendo el riesgo de tomar malas decisiones que puedan acarrear unos malos resultados.

La descripción del algoritmo de Nelder-Mead es la siguiente:

```

1:   Inicializar simplex original ( $x^1 \dots x^{n+1}$ );
2:   for (i=1;i<=n+1;i++) {
3:        $f^i = f(x^i)$ ; // Evaluamos en cada punto del simplex original
4:   }
5:   Ordenar  $f^i$  y  $x^i$  de forma:  $f^1 < f^2 < \dots < f^{n+1}$ ;
6:    $k = 1$ ;
7:   While(1) {
8:       // Se actualizan los costes de los n puntos con menor
9:       // coste, utilizando un factor de olvido  $\alpha$ 
10:      for (i=1;i<=n;i++) {
11:           $f^i = (f^i + \alpha f(x^i)) / (1 + \alpha)$ ;
12:      }
13:      // Se calcula el centroide de los n puntos con menor coste
14:       $x_{cent} = \text{CálculoMedia}(x^1, x^2, \dots, x^n)$ ;
15:      // Se refleja el peor punto respecto al centroide
16:       $x_r = x_{cent} + \eta_r(x_{cent} - x^{n+1})$ ;
17:       $f_r = f(x_r)$ ; // Se evalúa el punto reflejado
18:      if ( $f^1 \leq f_r \leq f^{n+1}$ ) then {
19:          //  $x_r$  no es ni el mejor punto ni el peor
20:           $x^{n+1} = x_r$ ;
21:      }
22:      else if ( $f_r < f^1$ ) then {
23:          //  $x_r$  es el mejor punto, ejecutamos la expansión
24:           $x_e = x_{cent} + \gamma_e(x_r - x_{cent})$ ;
25:           $f_e = f(x_e)$ ; // Se evalúa el punto expandido
26:          if ( $f_e < f_r$ ) then  $x^{n+1} = x_e$ ;
27:          else  $x^{n+1} = x_r$ ;
28:      }
29:      else {
30:          //  $x_r$  es el peor punto, ejecutamos la contracción
31:           $x_c = x_{cent} + \rho_c(x^{n+1} - x_{cent})$ ;
32:           $f_c = f(x_c)$ ; // Se evalúa el punto contraído
33:          if ( $f_c < f^{n+1}$ ) then {
34:               $x^{n+1} = x_c$ ;
35:          }
36:          else {
37:              // Ejecutamos el Shrink
38:              for (i=2;i<=n+1;i++) {
39:                   $x^i = x^1 + \sigma(x^i - x^1)$ ;
40:                   $f^i = f(x^i)$ ;
41:              }
42:          }
43:      }
44:       $k++$ ;
45:  }

```


Capítulo 3. Implementación

Tras el diseño del control, la siguiente etapa de este trabajo ha sido la implementación del mismo sobre un microcontrolador, de manera que se ha podido comprobar su viabilidad a la hora de llevarlo a la práctica. En este capítulo se presentan los distintos componentes de los que consta el sistema, además del cómo trabajan, así como la generación de las distintas señales de control utilizadas y su parametrización.

3.1. Hardware utilizado

Para verificar el buen funcionamiento de las estrategias de control desarrolladas, uno de los objetivos principales de este trabajo, se ha realizado un prototipo experimental para el control de un relé a través de un microcontrolador. Con esto se ha pretendido demostrar la viabilidad y la efectividad del control de sistemas electromecánicos mediante control R2R basados en métodos de optimización de Búsqueda Directa.

El sistema físico consta de cuatro elementos principales:

- **Microcontrolador.** Las funciones principales del microcontrolador son las de ejecutar la tarea de control, generar la señal de control por medio de una PWM y la adquisición de datos (Tiempo de rebotes e intensidad sonora) para calcular un coste para cada conmutación. Se ha optado por un microcontrolador como paso intermedio entre las pruebas experimentales realizadas en otros trabajos [1], en las que se utilizaba un PC y un osciloscopio, y el dispositivo que podría ser utilizado finalmente, por ejemplo un circuito ASIC. Por esta misma razón no se ha pretendido ajustar las especificaciones del dispositivo a las necesidades que planteaba este tipo de control. Como microcontrolador se ha utilizado un ATSAM3X8E, de la marca Atmel, el cual está montado sobre una placa Arduino Due y es programable en C. Este es capaz de trabajar a 84 MHz, más que suficiente para las necesidades de este trabajo.
- **Relé.** Se ha trabajado únicamente con un modelo de relé, el modelo RZH9-001 de la marca TE. Se trata de un relé de propósito general. Cuenta con ocho terminales. Dos de ellos son para la alimentación de la bobina, y los otros seis son los bornes del contacto móvil, el contacto NO y el contacto NC, todos ellos por duplicado. Durante los ensayos, el relé no utiliza para llevar a cabo la conmutación de un circuito secundario, por lo que para medir el tiempo de rebotes se aplica una tensión de 3.3V al contacto móvil, y se mide la tensión de salida tanto del NC como del NO para determinar la duración de estos rebotes. Durante todos los ensayos, el relé ha sido alimentado a 24V, correspondientes a su tensión nominal.

- **Circuito Activador.** Es el circuito encargado de dar la potencia necesaria a la bobina del relé para poder llevar a cabo la conmutación. El esquema de este circuito es el mismo que se venía utilizando hasta ahora, con la salvedad de que se han añadido dos resistencias entre la base y el emisor de los dos BJT con el fin de reducir el tiempo de conmutación. En este caso, como se va a generar las señales a través de una PWM de alta frecuencia, este parámetro pasa a ser crítico.
- **Micrófono.** Con el fin de poder ponderar en la función de coste el ruido que se produce en cada conmutación, se ha utilizado un micrófono con su correspondiente etapa de adaptación. Hay que hacer notar, que si bien se ha dispuesto de este micrófono y el sistema está totalmente preparado para hacer uso del mismo si se desea, durante todo el trabajo se ha tratado con el coste asociado únicamente al tiempo de rebotes, ya que el ruido generado y el tiempo de rebotes están fuertemente relacionados.

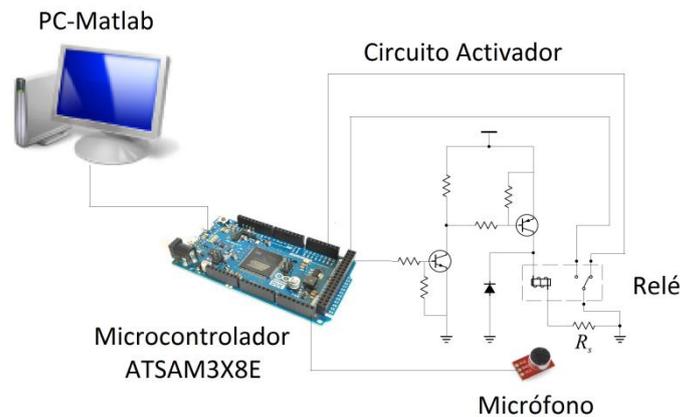


Figura 8. Implementación del sistema de control, con el microcontrolador ATSAM3X8E en una placa Arduino Due, el circuito de activación sobre el que va montado el relé, y el micrófono en el caso de que se quiera utilizar para calcular el coste.

Para el análisis de todos los datos obtenidos a través del microcontrolador se ha implementado una comunicación serie con el PC a través de Matlab. Se han realizado distintos programas en Matlab que permiten llevar a cabo la evaluación y comparación de los distintos algoritmos y señales de control. Estos trabajan de manera casi autónoma, facilitando enormemente la tarea de adquisición de datos.

3.2. Señal de control

Como ya se ha dicho, la señal de control de este sistema es la tensión con la que se alimenta la bobina del relé. Haciendo variar esta tensión se obtienen comportamientos distintos en la conmutación permitiendo obtener un coste, duración de los rebotes, menor.

Es el algoritmo de optimización el encargado de hacer variar esta señal para obtener unos costes menores. Sin embargo, para que el algoritmo de optimización pueda modificarla es necesario previamente definir el tipo de señal y los parámetros que la definen. El tipo se refiere a la forma que tiene la señal, sin especificar sus dimensiones, mientras que los parámetros, agrupados en el vector \bar{v} , son los valores necesarios para que la señal de cierto tipo quede completamente definida. Es por lo tanto el tipo de señal el que fije el número de parámetros a utilizar, y de igual modo, la dimensión del espacio de búsqueda para el algoritmo de optimización. Todo esto se puede entender mejor con la Tabla 2.

Como ya se ha comentado en 2.1 al introducir el problema, al modificar el tipo de la señal de control utilizada, se está modificando al mismo tiempo la característica de la función de coste χ_v . La idoneidad de un tipo de señal se mide tanto en el valor del mínimo de su función de coste, como en la facilidad para encontrarlo. Cuanto mayor es la región convexa más fácil resultará encontrar el mínimo global, y no converger a un mínimo local.

Por todo esto, la señal de control juega un papel crucial en el proceso de optimización del control R2R ya que es la que fija tanto la dimensión del espacio de búsqueda de la solución como la característica de la función de coste.

Por otro lado, hay que hacer notar que un mayor número de parámetros no tiene por qué implicar un mejor comportamiento del algoritmo de optimización. De hecho, un mayor número de parámetros acarrea el riesgo de que no se encuentre el óptimo o de que sean necesarias un mayor número de iteraciones. En el caso extremo, se puede pensar en parametrizar la señal punto por punto, es decir, que cada parámetro determine el valor de la misma en cada instante. Esto conduce a un número infinito de parámetros, haciendo el problema inabordable. Por el contrario, un número demasiado pequeño de parámetros puede conllevar que el mínimo que se alcance no sea lo suficientemente bajo, aunque este sea fácil de encontrar. Hay que buscar por lo tanto un compromiso teniendo en cuenta el punto óptimo, así como la facilidad y la velocidad con la que se converge a ese punto.

Para la generación de las distintas señales, como ya se ha dicho, el microcontrolador utiliza la modulación PWM. La frecuencia de actualización de la PWM se ha fijado a 82 kHz, frecuencia que es suficiente para que sea filtrada por la bobina (Circuito LR equivalente de $L = 10 H$ y $R = 1.5 k\Omega$) y esta se vea afectada únicamente por la componente continua. Al mismo tiempo, esta frecuencia permite tener un *duty* de la PWM que varía entre 0 y 255, siendo la precisión de la señal de 1/256.

Todas las señales utilizadas se muestran en la Tabla 2. Las señales de tipo A y B ya han sido utilizadas en otros trabajos [1], mientras que las señales de tipo C y D han sido probadas por primera vez en este trabajo. La razón por la que en las señales C y D se hace la transición suave entre el nivel alto y bajo no es otra que evitar que una pequeña modificación de los parámetros de tiempo en alto y en bajo conlleve un gran impacto sobre la duración de los rebotes. De este modo, una pequeña modificación en los

tiempos no conllevará un cambio radical entre alto y bajo, sino que se pasará a un valor intermedio, correspondiente a un punto en la transición. Sin embargo no hay que olvidar que esta modificación gradual se da sobre la tensión y no sobre la intensidad, quien en verdad afecta al comportamiento de la bobina.

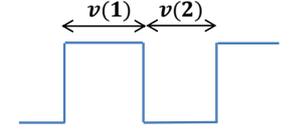
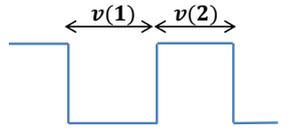
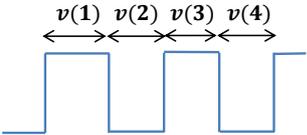
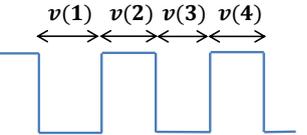
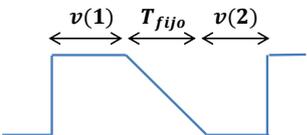
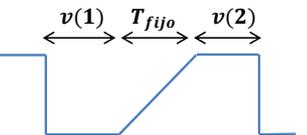
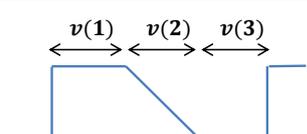
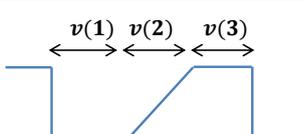
Señal	Apertura	Cierre	Parámetros
Tipo A			$\bar{v} \in \mathbb{R}^2$ $\bar{v} = \begin{pmatrix} v(1) \\ v(2) \end{pmatrix}$
Tipo B			$\bar{v} \in \mathbb{R}^4$ $\bar{v} = \begin{pmatrix} v(1) \\ v(2) \\ v(3) \\ v(4) \end{pmatrix}$
Tipo C			$\bar{v} \in \mathbb{R}^2$ $T_{fijo} \text{ cte}$ $\bar{v} = \begin{pmatrix} v(1) \\ v(2) \end{pmatrix}$
Tipo D			$\bar{v} \in \mathbb{R}^3$ $\bar{v} = \begin{pmatrix} v(1) \\ v(2) \\ v(3) \end{pmatrix}$

Tabla 2. Señales utilizadas, tanto para la apertura como para el cierre, junto con su representación gráfica y parametrización.

3.3. Programación microcontrolador

A la hora de programar el microcontrolador se han distinguido dos tareas principales a ejecutar:

- Cálculo de la señal de control mediante la estrategia de control R2R, la cual está completamente definida ya al comienzo de la iteración. En esta tarea se fijan los valores de los distintos parámetros que caracterizan la señal. Antes del inicio de la siguiente conmutación se envían los datos correspondientes a la conmutación anterior. Estos datos son tanto los puntos de evaluación como los costes asociados a cada uno de ellos. Esta tarea de cálculo de la señal se realiza a la misma frecuencia de conmutación del relé, justo antes de llevar a cabo la conmutación.
- La tarea de generación de la señal de control, la cual a partir de los parámetros característicos previamente calculados, se encarga de aplicar esta señal al

llevar a cabo la lectura de datos, así como el cálculo de costes. Durante todos los experimentos aquí mostrados se ha seleccionado una frecuencia de 200 kHz. Para el cálculo de la duración de los rebotes se han conectado los bornes NO y NC del relé a dos entradas digitales, ya que los valores únicamente pueden ser 0V ó 3.3V. En estos dos pines digitales se han habilitado sendas interrupciones que se activan cuando se produce un cambio en el valor de la entrada.

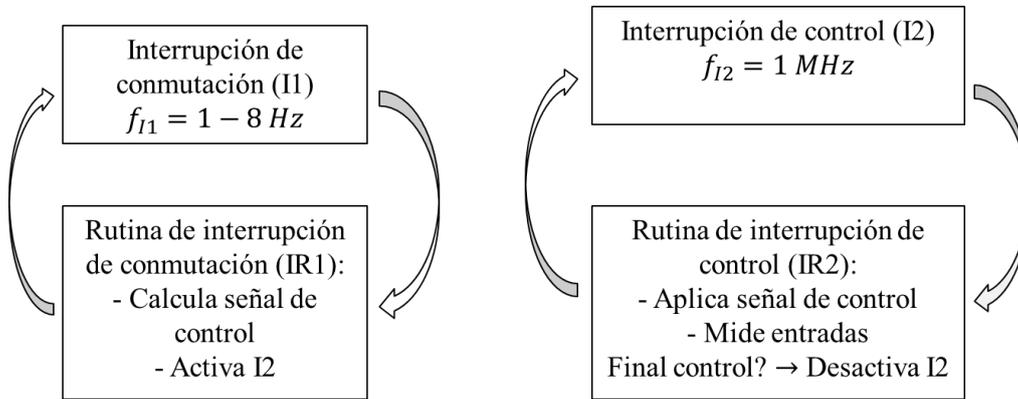


Figura 10. Esquema de funcionamiento del microcontrolador e interacción entre las dos interrupciones.

Una vez que han pasado más de 30 ms (Siendo $t_f = 30 \text{ ms}$ según Figura 9) desde el comienzo de la aplicación de la señal de control, se considera que la conmutación ha terminado, y se desactiva la interrupción 2, quedando desactivada hasta que se dé la siguiente conmutación.

A la hora de implementar los distintos algoritmos de optimización se ha procedido en todos los casos de igual modo, utilizando una descripción mediante máquina de estados. La descripción de las mismas viene recogida en el Anexo 2. Al utilizar siempre una misma estructura de programación se facilita la comprensión de los distintos programas así como la aplicación de nuevos algoritmos de optimización.

Capítulo 4. Evaluación experimental y análisis de resultados

En este capítulo se presentan los resultados experimentales más relevantes. En una primera etapa se ha comparado el comportamiento de los distintos algoritmos de optimización, utilizando una misma señal de control. Fijando la señal de tipo A y variando el algoritmo de optimización, se ha podido analizar cuál es el más adecuado. En una segunda etapa, de manera similar pero para analizar las señales de control, se ha fijado el algoritmo de optimización y se han hecho variar las señales.

4.1. Creación de mapa de costes

Como fase previa a la ejecución de los algoritmos, se ha realizado una etapa de adquisición de datos con el único fin de dibujar un mapa de costes para el caso de la señal de tipo A.

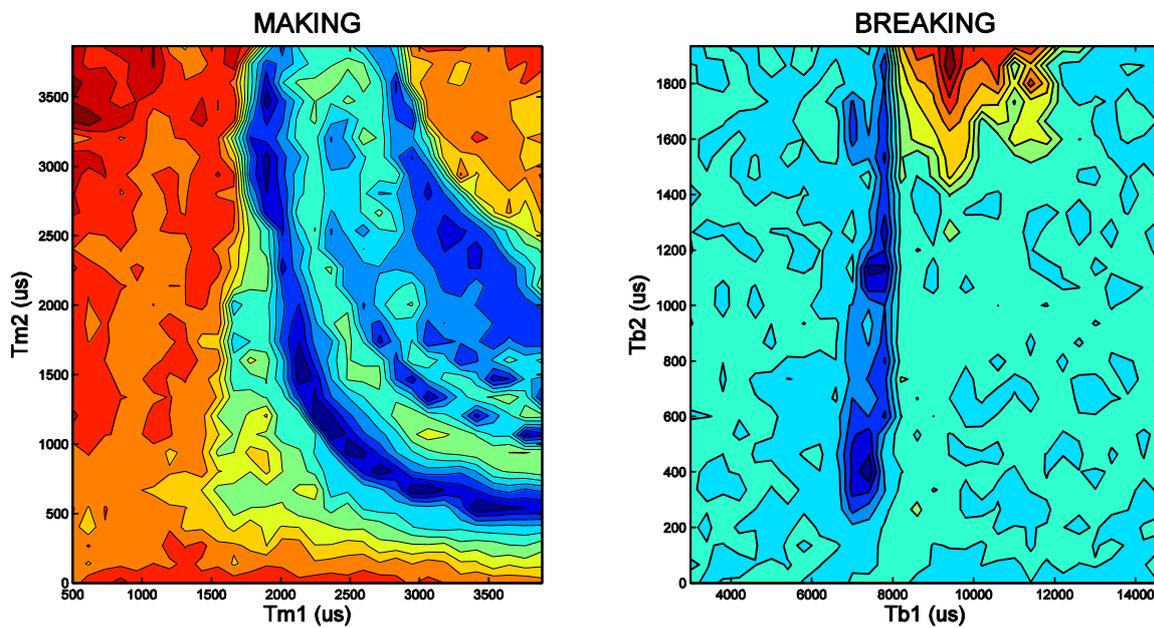


Figura 11. Mapa de costes para la acción de cierre (MAKING) y para la de apertura (BREAKING) para una señal de tipo A. Colores azules oscuros indican coste bajo, colores rojos intensos indican costes altos. Los valores de tiempo en alto y tiempo en bajo, $Tm1$, $Tm2$, $Tb1$ y $Tb2$, corresponden a los valores de $x(1)$ y $x(2)$ según Tabla 2. Los valores límites se han tomado según [1], de modo que $Tm1 \in [500 \mu s, 4000 \mu s]$, $Tm2 \in [0 \mu s, 4000 \mu s]$, $Tb1 \in [3000 \mu s, 15000 \mu s]$ y $Tb2 \in [0 \mu s, 2000 \mu s]$. Para la obtención del coste se han realizado 8 medidas en un total de 900 puntos diferentes.

Con esto se busca estudiar el comportamiento del relé para una señal de tipo A, pudiendo observar en qué región se encuentran los valores mínimos de coste, delimitando así la zona de búsqueda del algoritmo.

Se observa en Figura 11 que las funciones de coste son diferentes para la apertura y para el cierre. En el mapa de coste de Apertura existen amplias zonas con valores mínimos lo que facilita el proceso de búsqueda. Sin embargo, en el cierre, únicamente existe un mínimo, y se encuentra en una región muy localizada. Este a su vez está rodeado por una zona plana, con un coste constante, con gradiente nulo, lo que dificulta en buena medida la búsqueda del mínimo. Para el caso de otras señales donde el número de parámetros es mayor, no es posible representar un mapa de coste en 2D.

Una vez trazados los mapas de costes, se toman como regiones de búsqueda $Tm1 \in [1500 \mu s, 3000 \mu s]$, $Tm2 \in [500 \mu s, 2000 \mu s]$, $Tb1 \in [7000 \mu s, 9000 \mu s]$ y $Tb2 \in [200 \mu s, 1000 \mu s]$.

4.2. Procedimiento de medida

A la hora de realizar las medidas experimentales, hay una multitud de parámetros modificables que dan lugar a un comportamiento u otro del control. Se puede modificar desde el tipo el algoritmo de optimización así como todos los parámetros que definen su comportamiento, hasta el tipo de señal a aplicar o la región de búsqueda y puntos iniciales del control.

Para limitar esta gran cantidad de variables, se van a mantener para todos los experimentos los mismos parámetros del algoritmo de optimización. Por otro lado, al fin de comparar los algoritmos, se ha tomado la señal de control de tipo A, y se ha probado para cada uno de los algoritmos de optimización, tomando distintos puntos iniciales así como distintos tamaños de la zona de búsqueda. Por lo tanto, en una primera fase, cuyo objetivo es obtener una comparativa global de los distintos algoritmos de optimización, con una señal de tipo A y todos los parámetros de optimización fijados, se ha hecho variar:

- El algoritmo de optimización entre Búsqueda Compás, Búsqueda Multidireccional y Nelder-Mead.
- El tamaño de la región de búsqueda, para poder determinar qué algoritmo de optimización es el óptimo dependiendo del tamaño del espacio de búsqueda.
- Los puntos de inicio de control, con el fin de eliminar los posibles efectos propios de cada punto inicial sobre el comportamiento del control.

Estos algoritmos se han comparado con el caso de la conmutación del relé sin control, calculando en qué porcentaje se reduce el coste respecto a este, cuando ya se ha pasado la etapa de transición del algoritmo, y el control ya ha convergido hacia un punto estable. Se ha considerado que el algoritmo ha convergido en cualquier caso cuando ya

se han llevado a cabo 300 conmutaciones. Se ha medido también el coste de cada uno de estos métodos a 50, 100, 200 y 300 conmutaciones para poder así medir la velocidad de convergencia de cada uno de ellos. Además, para cada situación se han llevado a cabo 20 realizaciones, permitiendo medir en qué porcentaje de realizaciones se consigue determinado nivel de reducción.

Una vez realizada esta primera etapa de control, viendo cuál de los algoritmos de optimización tiene un mejor comportamiento, se ha realizado una segunda etapa cuyo fin es comparar el comportamiento para distintas señales haciendo uso del mejor algoritmo antes encontrado. El algoritmo de optimización que ha mostrado un mejor comportamiento en la primera etapa, ha sido evaluado posteriormente para distintas señales de control. En esta segunda etapa, se ha hecho variar:

- El tipo de señal que se aplica entre las señales de Tipo A, Tipo B, Tipo C y Tipo D.
- El tamaño de la región de búsqueda, para poder determinar qué señal de control es la óptima dependiendo del tamaño del espacio de búsqueda.
- Los puntos de inicio de control, con el fin de eliminar los posibles efectos propios de cada punto inicial sobre el comportamiento del control.

Por brevedad a la hora de redactar la memoria, en este capítulo únicamente se ven recogidos los datos que se han considerado más relevantes. Adicionalmente, en el Anexo 1 se recopilan una gran cantidad de datos y medidas complementarias con sus correspondientes análisis y conclusiones.

4.3. Comparativa experimental de los algoritmos de optimización

Una vez realizadas todas las medidas sobre los distintos algoritmos, se obtienen los valores medios guardados en la Tabla 3.

Tamaño dominio Búsqueda	Duración de los rebotes (μs)					
	Pequeño		Mediano		Grande	
	Cierre	Apertura	Cierre	Apertura	Cierre	Apertura
Búsqueda Compás	60	2950	100	3200	100	3150
Búsqueda Multidireccional	60	1020	130	1850	80	1700
Nelder-Mead	30	1000	80	950	70	1000
No control	1500	3900	1500	3900	1500	3900

Tabla 3. Duración media de los rebotes, en la zona de convergencia, para cada uno de los algoritmos y tamaños de región de búsqueda, haciendo uso de una señal de control de Tipo A. El dominio de búsqueda pequeño corresponde a un tamaño de $8,8 \cdot 10^5 (\mu s^2)$, el mediano a uno de $2,4 \cdot 10^6 (\mu s^2)$ y el grande a uno de $4,2 \cdot 10^6 (\mu s^2)$.

Los resultados evidencian un muy buen comportamiento de los tres algoritmos de optimización para el caso del cierre, donde se obtiene en cualquiera de los casos unas reducciones de coste de más del 90 %. Estos resultados son acordes con los obtenidos en [1].

Sin embargo, este comportamiento homogéneo en el cierre no se trasmite a la apertura donde los resultados divergen dependiendo del algoritmo de optimización utilizado. En el caso de la apertura, el algoritmo Búsqueda Compás tiene un comportamiento notablemente peor que el de los otros dos, alcanzando unas reducciones de coste mucho menores. Entre el algoritmo de Búsqueda Multidireccional y el de Nelder-Mead, las diferencias son menores, pero en todos los casos se alcanza un tiempo de rebote menor para el caso de Nelder-Mead.

Este algoritmo no solo muestra un mejor comportamiento de cara al número de rebotes sino también al porcentaje de realizaciones en los que se consiguen grandes reducciones. Con Nelder-Mead se asegura una reducción del 30% de los rebotes en cualquiera de las situaciones, siendo en el 50 % de los casos una reducción de más del 80%.

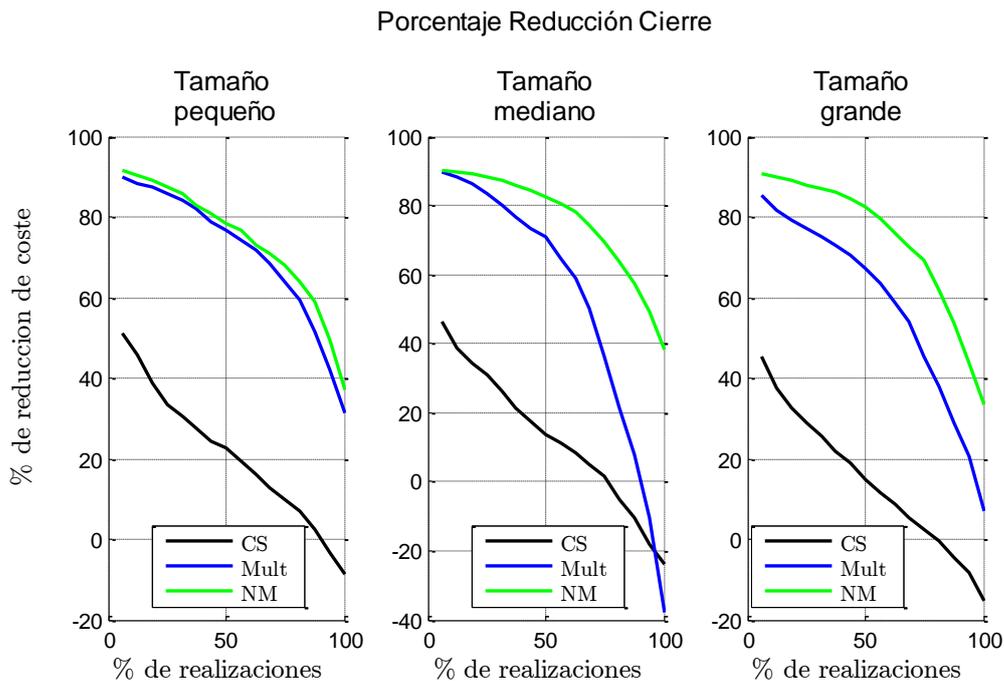


Figura 12. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de la apertura. La situación ideal sería que en todas las realizaciones se consiguiera una reducción del 100%, lo que en esta gráfica equivaldría a una recta al nivel del 100%. Por lo tanto, se puede medir cuánto de buena es una señal según cuánto de cerca se encuentre de esa recta al 100%. Según este criterio, quedaría de mejor a peor los algoritmos: Nelder-Mead, Búsqueda Multidireccional y Búsqueda Compás.

4.4. Comparativa experimental de las señales de control

Una vez realizadas todas las medidas sobre las distintas señales, se obtienen los valores medios representados en la Tabla 4.

A partir de esta tabla, se puede ver que para el caso de la operación de cierre, todas las señales muestran unos buenos resultados, con costes muy cercanos entre ellos. Además, se observa un comportamiento constante aun cuando se aumenta el tamaño de la región de búsqueda. La reducción respecto a la situación de no control varía entre el 87% y el 97%.

Por otro lado, si se considera la operación de apertura, se evidencia un mejor comportamiento de las señales de tipo A y B, en particular de esta segunda. El comportamiento de la señal B es significativamente mejor que el de la señal A para regiones de búsqueda pequeñas, mientras que si esta aumenta, el comportamiento de la señal B va empeorando, acercándose al de la señal A. Aunque a la hora de diseñar las señales C y D se pensó que estas podrían facilitar la convergencia del algoritmo de optimización permitiendo obtener un coste medio menor, para todos los casos se han obtenido un comportamiento mucho peor que el de las de tipo A y B, obteniendo costes mucho mayores para la apertura, tal y como se muestra en Tabla 4.

Tamaño dominio Búsqueda	Duración de los rebotes (μ s)					
	Pequeño		Mediano		Grande	
	Cierre	Apertura	Cierre	Apertura	Cierre	Apertura
Señal Tipo A	100	1050	130	1150	120	1200
Señal Tipo B	50	700	70	850	125	1000
Señal Tipo C	100	2800	120	3000	200	3000
Señal Tipo D	200	2200	200	2600	130	2800
No control	1500	4100	1500	4100	1500	4100

Tabla 4. Duración media de los rebotes, en la zona de convergencia, para cada una de las señales de control y tamaños de región de búsqueda, haciendo uso del algoritmo de optimización Nelder-Mead, que se ha visto que tiene un mejor comportamiento.

En el caso de la apertura, si se observan los porcentajes de reducción de coste para distintos porcentajes de realizaciones, también es la señal B la que muestra un mejor comportamiento.

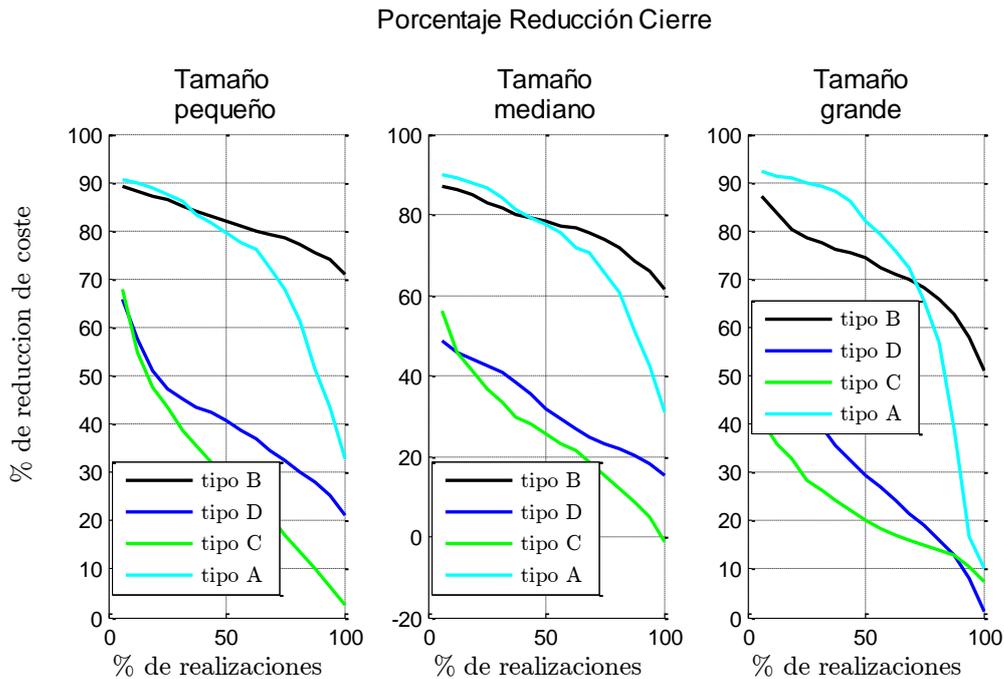


Figura 13. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada tamaño de región de búsqueda y señales. Caso de la apertura.

Para el caso de esta señal, se asegura una reducción del 70%, 60% y 50% para la región de búsqueda pequeña, mediana y grande respectivamente. Comparando la señal B con la señal A en el dominio de búsqueda grande, se comprueba que para un 70% de las realizaciones es la señal A la que obtiene mayores reducciones que la señal B. Por el contrario, también se observa que la de tipo A tiene un comportamiento mucho más variable ya que el porcentaje de reducción de coste varía entre el 10% y el 93%, mientras que en la señal B únicamente varía entre el 52% y el 88%.

4.5. Análisis de los resultados experimentales

Una vez realizados los ensayos para los distintos controles, se obtienen las siguientes conclusiones para el caso de la operación de cierre:

- El uso combinado de la señal de tipo B y el método Nelder-Mead ofrece reducciones de hasta el 97%. Estos resultados quedan por lo tanto difícilmente superables.
- En la operación de cierre se obtienen muy buenos resultados para todos los casos, ya que se tiene un comportamiento poco variable respecto al tipo de la señal de control y el algoritmo de optimización utilizados. Por esta razón se podría optar por la combinación del método de Búsqueda Compás con la señal de tipo A, combinación que es la más fácil de implementar, reduciendo los requerimientos de cálculo del sistema, en este caso el microcontrolador.

Mientras que para el caso de la operación de apertura:

- Tras los ensayos realizados, se ha concluido que es la estrategia de control basada en Nelder-Mead la que permite obtener unos mejores resultados, tanto en la reducción de coste medio como en el porcentaje de realizaciones en los que se consiguen unas mayores reducciones. Se ha comprobado además su buen funcionamiento para espacios de búsqueda de distintas dimensiones (\mathbb{R}^2 para la señal de tipo A y \mathbb{R}^4 para la de tipo B)
- Es el uso combinado de la señal de tipo B y el algoritmo de Nelder-Mead el que obtiene unos mejores resultados de manera global, asegurando para cualquiera de los casos reducciones de coste mayores del 50 %, pudiendo alcanzar incluso el 87%. Aunque para regiones de búsqueda grandes se ha observado que en cierto porcentaje de realizaciones es la combinación de Nelder-Mead con la señal de tipo A la que obtiene unos menores costes, esta combinación presenta una gran variabilidad que no permite asegurar reducciones de coste mayores del 10%, siendo por lo tanto descartada frente a la señal de tipo B.
- Los porcentajes de reducción obtenidos para la operación de apertura son peores que para el caso del cierre. Esto es debido en parte a la forma de las funciones de coste. Para el caso del cierre, la región convexa de la función es mucho mayor que para la apertura, siendo por lo tanto más propicia para la aplicación de algoritmos de optimización.

Globalmente, el control R2R de sistemas electromecánicos, más concretamente de relés, permite reducir de manera considerable la duración de los rebotes de la parte móvil. Sin embargo, no hay que olvidar que esta reducción no se consigue de manera instantánea, sino que hay que proceder a unas medidas previas que hagan converger el algoritmo de búsqueda hacia el mínimo de la función de coste. Por lo tanto, este tipo de control queda condicionado a que los dispositivos electromecánicos trabajen de forma repetitiva, de modo que exista la posibilidad de evaluar la función de coste en distintos puntos. Esta condición no restringe demasiado el ámbito de aplicación del control R2R, ya que estos suelen trabajar de forma repetitiva, como es en el caso de convertidores de tensión, circuitos controladores o etapas de potencia.

Capítulo 5. Conclusiones y Líneas futuras

5.1. Conclusiones

En este Trabajo Fin de Máster se ha analizado de manera teórica y experimental el proceso de conmutación de los dispositivos electromecánicos. Los problemas que aparecen durante este proceso radican en la existencia de rebotes entre la parte fija y móvil del dispositivo. El fin principal de este trabajo ha sido por lo tanto el estudio e implementación de estrategias de control para la reducción de estos rebotes.

Como primer paso, se ha llevado a cabo un estudio del estado del arte referido al control de dispositivos electromecánicos, y más concretamente, de los relés. Se ha observado que la mayoría de los controles existentes hacían uso de un modelo teórico del relé, a partir del cual predecir su comportamiento. Sin embargo, las divergencias entre el comportamiento teórico y el real, hacen que estas estrategias no generen los buenos resultados esperados.

Como alternativa a estos controles basados en el modelo del sistema, se ha propuesto recientemente la aplicación de estrategias de control *Run-to-Run* (R2R), las cuales no necesitan un modelo preciso del sistema. Este tipo de control aprovecha el carácter repetitivo de estos sistemas, los cuales realizan las dos únicas operaciones de apertura y cierre de manera reiterada. Mediante un determinado algoritmo de optimización se consigue reducir un coste, en este caso la duración de los rebotes. Este trabajo se ha centrado exclusivamente en el control R2R.

Si bien este tipo de control ya había sido probado en relés con buenos resultados, quedaban pendientes por estudiar dos puntos muy importantes. Por un lado, probar la viabilidad real de este tipo de control en una implementación real. Con anterioridad se había implementado mediante el uso de costosos y voluminosos equipos como osciloscopios y computadoras. Quedaba por demostrar si este control era factible y se podía llevar a cabo mediante el uso de dispositivos de bajo coste y reducido tamaño, acercándose más a su aplicación final. Y por otro lado, el único algoritmo de optimización que se había utilizado hasta ahora para implementar este tipo de control había sido el de Búsqueda Compás. Pese a ser este uno de los más sencillos se obtenían buenos resultados. Sin embargo, quedaban por probar otros algoritmos para ver si se podían obtener mejores resultados, es decir, una duración de rebotes menor.

Por lo tanto, una vez que se ha decidido utilizar el control R2R se ha pasado a estudiar los distintos algoritmos de optimización con los cuales diseñar la estrategia de control R2R. Entre todos estos algoritmos se han elegido un total de tres: Búsqueda Compás,

Búsqueda Multidireccional y Nelder-Mead, donde se ha incluido Búsqueda Compás de manera que sirva de referencia y se pueda comparar con los otros dos.

Tras la elección de los algoritmos de optimización se han diseñado cada una de las estrategias de control R2R. Para llevar a cabo su implementación, se ha intentado ajustar lo máximo posible las prestaciones del dispositivo encargado del control respecto a las necesidades de potencia de cálculo de cada una de las estrategias. Una vez implementadas estas estrategias de control, se han llevado a cabo una serie de experimentos con el fin de caracterizar a cada una de las estrategias.

Durante la realización de este trabajo se han alcanzado los siguientes objetivos:

- Probar la viabilidad del control R2R para sistemas electromecánicos. Mediante la implementación de este esquema de control sobre un microcontrolador, y la posterior verificación experimental se ha demostrado la viabilidad del control R2R para sistemas electromecánicos que trabajan de forma repetitiva. Se ha pasado por lo tanto de utilizar un osciloscopio y una computadora con un coste alto, al uso de un único microcontrolador, cuyo coste es de decenas de euros.
- Desarrollar y comparar nuevas estrategias de control R2R. Durante este trabajo se han estudiado y utilizado algoritmos de optimización, Búsqueda Multidireccional y Nelder-Mead, hasta ahora no utilizados para el caso del control de este tipo de sistemas mediante técnicas R2R. En particular, para la estrategia basada en el algoritmo de Nelder-Mead, se han obtenido unos resultados muy satisfactorios para la apertura y cierre del relé, que mejoran los obtenidos con el algoritmo de Búsqueda Compás.
- Usar y caracterizar nuevas y distintas señales de control. En total se han probado cuatro señales diferentes. Los mejores resultados han sido obtenidos para la señal de Tipo B, la cual ya había sido utilizada en otros trabajos. Por otro lado, las señales de Tipo C y D se han diseñado y utilizado por primera vez en este trabajo. Aunque para estas dos señales los resultados obtenidos distan mucho de ser los mejores, estas han sido probadas, permitiendo conocer su comportamiento, y descartándolas para futuros trabajos.

5.2. Líneas futuras

Si bien en este Trabajo Final de Máster se avanza en el conocimiento del control R2R de dispositivos electromecánicos, siguen existiendo una gran cantidad de posibles líneas futuras para próximos trabajos. A continuación se pasan a enumerar algunas de estas en orden decreciente de interés:

- Uso simultaneo del modelo aproximado del sistema y de los algoritmos de optimización. Existen técnicas donde se utiliza un modelo local simplificado del sistema de manera combinada con un algoritmo de optimización. Este es el caso por ejemplo del uso de modelos sustitutos (*Surrogates*) [20] [21]. Por otro lado,

otras técnicas como el RSM (Método de Respuesta Superficial) utilizan las medidas realizadas para construir un modelo local del sistema y poder determinar con mayor precisión dónde se puede localizar el mínimo de la función de coste [22]. Existen también otros métodos como el Kriging [23] en el que a partir de las medidas realizadas se construyen modelos estocásticos que, en lugar de determinar el valor en cada punto, determinan qué distribución sigue este valor. El estudio de estos métodos sería la evolución natural de este trabajo.

- Estudio de nuevos algoritmos de optimización. Si bien en este trabajo se han implementado nuevos algoritmos de optimización, existe una gran cantidad de ellos aún sin estudiar. Este trabajo se ha centrado en los métodos de optimización libres de derivada, porque a priori eran los más idóneos. Sin embargo sería interesante estudiar la aplicabilidad de otros métodos basados en el cálculo de la derivada, así como las modificaciones necesarias para hacerlos factibles ante la presencia de ruido.
- Estudio del comportamiento estocástico del sistema. Durante este trabajo se ha constatado un comportamiento aleatorio tanto del sistema como de las medidas obtenidas a través del microcontrolador. Sería interesante estudiar este problema con un enfoque estadístico, mediante el posible uso combinado de estimadores y test estadísticos.
- Uso de los distintos modelos del sistema para inicializar los puntos del control R2R, haciendo que estos se encuentren lo más cerca posible del mínimo de la función de coste, y por lo tanto, facilitando la tarea de minimización. De esta forma se asegura mejorar los resultados obtenidos respecto al caso de usar únicamente el modelo.
- Implementación del control R2R sobre FPGA. Con el fin de acercarse más a la aplicación final sería interesante probar este tipo de control sobre una FPGA, para el posterior diseño de un circuito ASIC.

Referencias

- [1] E. Ramirez-Laboreo, C. Sagues and S. Llorente, “A New Run-to-Run Approach for Reducing Contact Bounce in Electromagnetic Switches,” *IEEE Transactions on Industrial Electronics*, 2016.
- [2] E. R. Laboreo, «Trabajo Final de Master: Modelado y control de dispositivos electromecánicos,» Universidad de Zaragoza, Escuela de Ingeniería y Arquitectura, 2015.
- [3] P. Barkan, “A Study of the Contact Bounce Phenomenon,” *IEEE Transactions on Power Apparatus and Systems*, no. 2, pp. 231-240, 1967.
- [4] J. W. McBride, «Electrical contact bounce in medium-duty contacts,» *IEEE Trans. Compon. Hybrids, Manuf. Technol.*, vol. 12, n° 1, p. 82–90, 1989.
- [5] Z. Guofu, W. Qiya, Y. Wenying, and L. Huimin, «Permanent-magnet equivalent model of calculating relay’s static attractive torque characteristics by finite element method,» *IEEE Trans. Magn.*, vol. 48, n° 9, p. 2467–2471, 2012.
- [6] E. Ramirez-Laboreo, C. Sagues and S. Llorente, “A New Model of Electromechanical Relays for Predicting the Motion and Electromagnetic Dynamics,” *IEEE Transactions on Industrial Electronics*, vol. 52, no. 3, pp. 2545-2553, May/June 2016.
- [7] R.T. Krein, M. Amrhein, «3-d magnetic equivalent circuit framework for modeling electromechanical devices,» *IEEE Trans. Energy Convers.*, vol. 24, n° 2, p. 397–405, 2009.
- [8] Z. Guofo, W. Qiya, and R. Wanbin, «An output space-mapping algorithm to optimize the dimensional parameter of electromagnetic relay,» *IEEE Trans. Magn.*, vol. 47, n° 9, p. 2194–2199, 2011.
- [9] O. Verlinden, D. Wattiaux, «Modelling of the dynamic behaviour of electromechanical relays for the analysis of sensitivity to shocks and vibrations,» *Exp. Mech.*, vol. 51, n° 9, p. 1459–1472, 2011.
- [10] A. G. Espinosa, J.-R. R. Ruiz and X. A. Morera, “A Sensorless Method for Controlling the Closure of a Contactor,” *IEEE Transactions on Magnetics*, vol. 43, no. 10, pp. 3896-3903, October 2007.
- [11] H. Lin, X. Wang, S. Fang, P. Jin and S. L. Ho, “Design, Optimization and Intelligent Control of Permanent Magnet Contactor,” *IEEE Transactions on Industrial Electronics*, vol. 60, no. 11, pp. 5148-5159, November 2013.
- [12] F. Malaguti, «Proportional control of on/off solenoid operated hydraulic valve by nonlinear robust controller,» *Proc. IEEE Int. Symp. Ind. Electron.*, vol. 2, p. 415–419, 2002.

- [13] B. Carse, N. Larsen, H. NourI and T. Davies, “An Approach To The Reduction Of Contact Bounce Using Fuzzy Contro,” *ISIE*, pp. 1025-1029, 1999.
- [14] R. Hooke and T. A. Jeeves, “Direct search solution of numerical and statistical problems,” *J. ACM*, p. 212–229, 1961.
- [15] V. Torczon, «On the convergence of the multidirectional search algorithm,» *SIAM J. Optim.*, vol. 1, p. 123–145, 1991.
- [16] G. E. P. Box, “Evolutionary operation: A method for increasing industrial productivity,” *Appl. Statist.*, p. 81–101, 1957.
- [17] Edward J. Anderson and Michael C. Ferris, “A Direct Search Algorithm for Optimization with Noisy Function Evaluations,” *SIAM Journal on Optimization*, vol. 11(3), p. 837–857, 2001.
- [18] R. M. Lewis and V. Torczon, “Rank Ordering and Positive Bases in Pattern Search Algorithm,” *Technical Report: Institute for Computer Applications in Science and Engineering (ICASE)*, 1996.
- [19] T. G. Kolda, R. M.I Lewis, V. Torczon, “Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods,” *SIAM Review*, vol. 45, no. 3, p. 385–482, 2003.
- [20] Virginia Torczon, Michael W. Trosset, «Using Approximation to Accelerate Engineering Design,» *NASA/CR, ICASE Report*, n° 98-33, 1998.
- [21] A. J. Booker, J. E. Dennis Jr., P. D. Frank, D. B. Serafini, V. Torczon, M. W. Trosset, «A rigorous framework for optimization of expensive functions by surrogates,» *Structural Optimization*, vol. Volume 17, n° 1, p. 1–13, 1999.
- [22] T. Zhang, K. K. Choi, S. Rahman, K. Cho, P. Baker, M. Shakil and D. Heitkamp, “A hybrid surrogate and pattern search optimization method and application to microelectronics,” *Struct Multidisc Optim*, no. 32, pp. 327-345, 2006.
- [23] C. M. Siefert, V. Torczon and M. W. Trosset, “Model-Assisted Pattern Search Methods for Optimizing Expensive Computer Simulations,” *Institute for Computer Applications in Science & Engineering, NASA*, 2000.
- [24] G. Liuzzi, S. Lucidi and F. Rinaldi, “Derivative-free methods for bound constrained,” *Computacional Optimization Application*, vol. 53, p. 505–526, 2012.

Índice de figuras

Figura 1.	Representación y descripción elemental de un relé electromecánico.....	7
Figura 2.	Representación de las señales NC y NO (cuyo valor a 1 indica circuito cerrado en los dos casos) las cuales son las únicas salidas del sistema si no se considera el ruido. Ambas son utilizadas para calcular la duración de los rebotes (tiempo entre el primer y último rebote) tanto en la apertura como en el cierre.....	14
Figura 3.	Representación de la única entrada del sistema, tensión de alimentación de la bobina, y de la intensidad medida en la bobina a través de una resistencia. La intensidad se podría utilizar como salida del sistema para calcular el coste, sin embargo aquí únicamente se va a tomar en consideración la duración de los rebotes.	15
Figura 4.	Esquema general de un control Run-to-Run [1].....	18
Figura 5.	Secuencia de optimización Búsqueda Compás. Figura obtenida de [19].....	19
Figura 6.	Secuencia de funcionamiento del R2R basado en Búsqueda Multidireccional. En azul se han representado los puntos del simplex original, en rojo, los del simplex rotado en torno al mejor punto, en verde esos mismos puntos contraídos, y en morado expandidos.	20
Figura 7.	Secuencia de funcionamiento del R2R con Nelder-Mead. En azul se ha representado el simplex original, en rojo el peor punto rotado respecto al punto x_{cent} , en verde el punto rotado contraído, en morado el rotado y expandido, y en naranja, el simplex contraído respecto del mejor punto x_1	22
Figura 8.	Implementación del sistema de control, con el microcontrolador ATSAM3X8E en una placa Arduino Due, el circuito de activación sobre el que va montado el relé, y el micrófono en el caso de que se quiera utilizar para calcular el coste.	26
Figura 9.	Esquema global de control R2R, con los distintos bloques de programación recuadrados en distintos colores. En rojo, tarea realizada por la interrupción de control. En azul y verde, bloques que se ejecutan dentro de la segunda interrupción. En verde la función encargada de generar la señal de control a partir de los parámetros y a través de una PWM. En azul, la función encargada de calcular la duración de los rebotes.....	29
Figura 10.	Esquema de funcionamiento del microcontrolador e interacción entre las dos interrupciones.	30
Figura 11.	Mapa de costes para la acción de cierre (MAKING) y para la de apertura (BREAKING) para una señal de tipo A. Colores azules oscuros indican coste bajo, colores rojos intensos indican costes altos. Los valores de tiempo en alto y tiempo en bajo, T_{m1} , T_{m2} , T_{b1} y T_{b2} , corresponden a los valores de $x(1)$ y $x(2)$ según Tabla 2. Los valores límites se han tomado según [1], de modo que $T_{m1} \in [500 \mu s, 4000 \mu s]$, $T_{m2} \in [0 \mu s, 4000 \mu s]$, $T_{b1} \in [3000 \mu s, 15000 \mu s]$ y $T_{b2} \in [0 \mu s, 2000 \mu s]$. Para la obtención del coste se han realizado 8 medidas en un total de 900 puntos diferentes.....	31
Figura 12.	Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de la apertura. La situación ideal sería que en todas las realizaciones se consiguiera una reducción del 100%, lo que en esta gráfica equivaldría a una recta al nivel del 100%. Por lo tanto, se puede medir cuánto de buena es una señal según cuánto de cerca se encuentre de esa recta al 100%. Según	

este criterio, quedaría de mejor a peor los algoritmos: Nelder-Mead, Búsqueda Multidireccional y Búsqueda Compás.....	34
Figura 13. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada tamaño de región de búsqueda y señales. Caso de la apertura. 36	
Figura 14. Evolución del coste según las iteraciones para cada uno de los algoritmos y situaciones en el caso de la operación de cierre. (NC=No Control, CS=Búsqueda Compás, Mult=Búsqueda Multidireccional y NM=Nelder-Mead)	48
Figura 15. Evolución del coste según las iteraciones para cada uno de los algoritmos y situaciones en el caso de apertura.	49
Figura 16. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de cierre.	51
Figura 17. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de apertura.	51
Figura 18. Variación de la velocidad de reducción de coste en el cierre, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.....	52
Figura 19. Variación de la velocidad de reducción de coste para la apertura, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.....	52
Figura 20. Evolución del coste según las iteraciones para cada una de las señales de control y situaciones en el caso del cierre.	54
Figura 21. Evolución del coste según las iteraciones para cada una de las señales de control y situaciones en el caso de la apertura.....	55
Figura 22. Evolución del coste de apertura según las iteraciones para la señal B en un total de 500 conmutaciones	55
Figura 23. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de cierre.	56
Figura 24. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de apertura.	57
Figura 25. Variación de la velocidad de reducción de coste para el cierre, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.....	58
Figura 26. Variación de la velocidad de reducción de coste para la apertura, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.....	58
Figura 27. Descripción esquemática mediante máquina de estados del algoritmo de optimización Búsqueda Compás.	59
Figura 28. Descripción esquemática mediante máquina de estados del algoritmo de optimización de Búsqueda Multidireccional.....	60
Figura 29. Descripción esquemática mediante máquina de estados del algoritmo de optimización Nelder-Mead.....	61

ANEXOS

Anexo 1. Medidas Experimentales

A.1.1. Comparativa algoritmos de optimización

En este apartado, para comparar los distintos algoritmos se ha utilizado una única señal de tipo A. Se ha hecho variar tanto el dominio de búsqueda como los puntos iniciales del algoritmo de optimización. Para cada situación:

- En cada realización se ha observado el comportamiento para las 300 primeras conmutaciones tanto para la apertura como para el cierre, de manera que el control tenga el tiempo suficiente para converger.
- Se han llevado a cabo un total de 20 realizaciones para cada situación, de forma que se pueda comprobar si el comportamiento en cada situación es constante, y si no, poder conocer entre qué valores cambia este comportamiento. Por otro lado, haciendo variar localmente los puntos iniciales de búsqueda se consiguen eliminar los posibles efectos asociados a estos puntos, de manera que únicamente aparezcan los fenómenos asociados al tamaño del dominio de búsqueda. Una vez llevadas a cabo estas veinte realizaciones se calcula también una media aritmética para cada número de conmutación.
- Para las operaciones de apertura y de cierre se han tomado tres dominios de búsqueda diferentes, en torno a los puntos óptimos que se han visto con los mapas de coste, y con distintos tamaños. Según la Tabla 2, y para los parámetros dados en microsegundos, se obtiene la Tabla 5. El tamaño de los dominios de búsqueda para la apertura y el cierre no son los mismos. En el caso de la apertura (*breaking*) debido a que la forma de la función de coste es más compleja se ha preferido utilizar un dominio de búsqueda más pequeño que para el cierre (*making*), facilitando la obtención del mínimo.

Dominio Pequeño:	Dominio Mediano:	Dominio Grande:
$v_m^{m\acute{a}x} = (2900,1900);$	$v_m^{m\acute{a}x} = (3267,2333);$	$v_m^{m\acute{a}x} = (3633,2767);$
$v_m^{m\acute{i}n} = (1800,600);$	$v_m^{m\acute{i}n} = (1433,167);$	$v_m^{m\acute{i}n} = (1067,0);$
$v_b^{m\acute{a}x} = (8750,800);$	$v_b^{m\acute{a}x} = (9250,1000);$	$v_b^{m\acute{a}x} = (9750,1200);$
$v_b^{m\acute{i}n} = (7250,200);$	$v_b^{m\acute{i}n} = (6750,0);$	$v_b^{m\acute{i}n} = (6250,0);$
$A_m = 1.43 \cdot 10^6 \mu s^2;$	$A_m = 3.97 \cdot 10^6 \mu s^2;$	$A_m = 7.1 \cdot 10^6 \mu s^2;$
$A_b = 8.72 \cdot 10^5 \mu s^2;$	$A_b = 2.5 \cdot 10^6 \mu s^2;$	$A_b = 4.2 \cdot 10^6 \mu s^2;$

Tabla 5. Dominios de búsqueda tanto para el cierre (*making*) como para la apertura (*breaking*) para una señal de tipo A, así como el área del dominio en μs^2 .

A la hora de seleccionar cuál de los distintos algoritmos de optimización es el mejor, se han comparado las siguientes características:

- Porcentaje de reducción de coste. En este apartado se estudia qué porcentaje de realizaciones para cada una de las situaciones se consigue reducir el coste en un cierto porcentaje respecto al coste del sistema no controlado. La comparación de coste con el sistema no controlado se realiza en la parte estacionaria, cuando el algoritmo ya ha convergido y el coste no varía, esto es de la conmutación 250 en adelante. Con esta comparación se pretende determinar qué algoritmo es más fiable, en el sentido de que consiga una reducción significativa del coste en un porcentaje alto de realizaciones. Se podría decir que no sirve de nada tener un algoritmo que es capaz de minimizar el coste en un gran porcentaje, si este coste se alcanza en un porcentaje muy bajo de realizaciones.
- Velocidad de convergencia. No basta con alcanzar un mínimo muy bajo, sino que además hay que hacerlo en el menor número de iteraciones posible. Se medirá por lo tanto en qué porcentaje reduce el coste para un determinado número de conmutaciones.

A.1.1.1 Evolución del coste

En esta parte se analiza cada método de optimización de manera global, según su perfil temporal, viendo cómo varía el coste en cada conmutación. Para mejorar la visualización de los resultados se representa la convolución de una ventana rectangular con la media de las 20 simulaciones para cada una de las situaciones, de manera que la curva queda filtrada. Por esta misma razón se descartan los primeros y últimos valores.

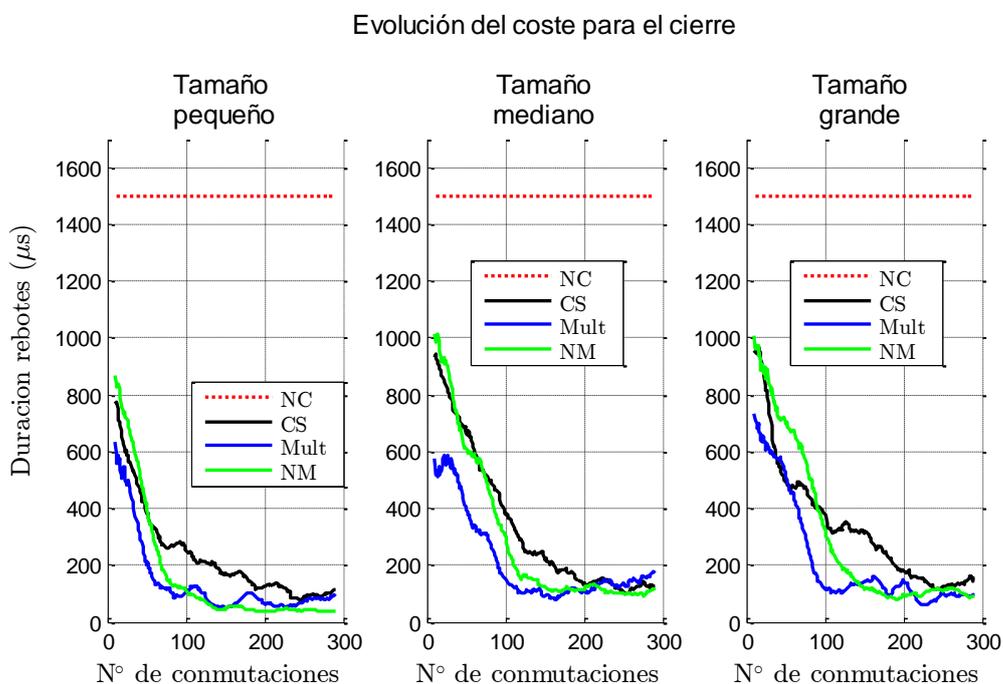


Figura 14. Evolución del coste según las iteraciones para cada uno de los algoritmos y situaciones en el caso de la operación de cierre. (NC=No Control, CS=Búsqueda Compás, Mult=Búsqueda Multidireccional y NM=Nelder-Mead)

Se observa en la Figura 14 que para la operación de cierre los resultados para cada una de las situaciones de los tres algoritmos son bastante similares entre sí. Los tres consiguen reducir el coste a un nivel similar, si bien se observa que el mínimo es alcanzado antes por el método de Búsqueda Multidireccional. La reducción respecto a la situación sin control es en cualquier caso superior al 90%.

Por otro lado, para el caso la apertura, en la Figura 15 se puede ver que la situación es completamente distinta. Mientras que para el método Búsqueda Compás se tiene que el coste no se reduce en ningún caso más allá del 25%, para los métodos de Búsqueda Multidireccional y Nelder-Mead esta reducción puede llegar hasta el 75%. En este caso queda evidenciado el mal funcionamiento del Búsqueda Compás para la operación de apertura.

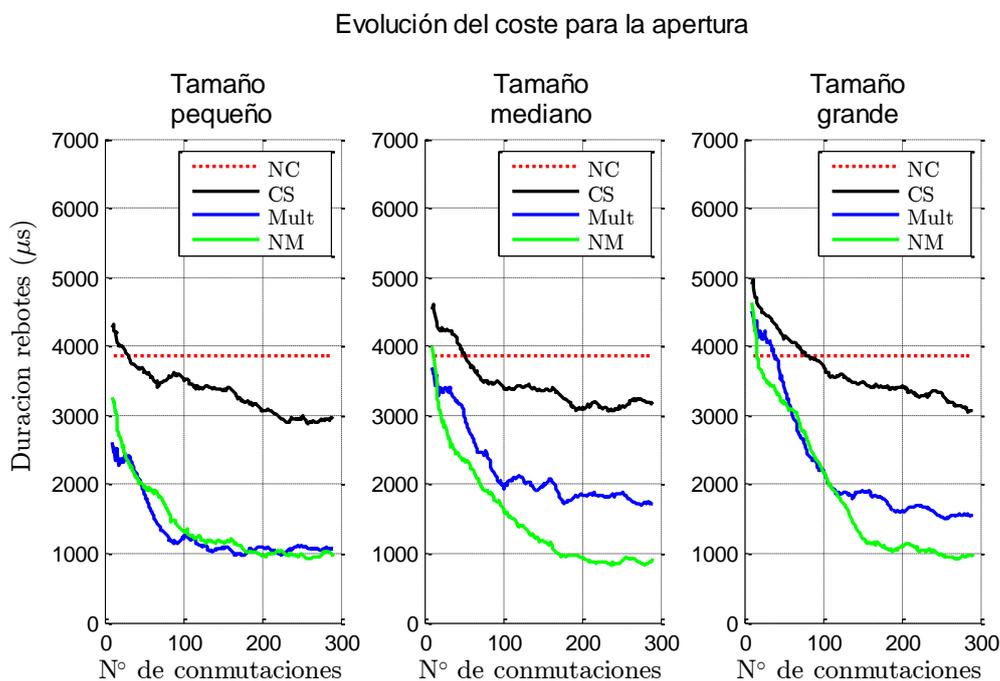


Figura 15. Evolución del coste según las iteraciones para cada uno de los algoritmos y situaciones en el caso de apertura.

Por otro lado, mientras que el algoritmo de Búsqueda Multidireccional empeora su comportamiento conforme el tamaño del dominio de búsqueda aumenta, el método Nelder-Mead sigue alcanzado prácticamente los mismos niveles de coste que en las situaciones más favorables.

A.1.1.2 Mejora de coste

En esta parte se estudiará cuánto se reduce el coste para cada situación. En este caso no se estudiará la media de las simulaciones sino que se verá para qué porcentaje de realizaciones se alcanza cierto nivel de reducción de coste.

Para el caso del cierre, se observa que Nelder-Mead obtiene una reducción de coste mayor para los mismos porcentajes de realizaciones en casi todos los casos. Se puede

destacar que Nelder-Mead garantiza una reducción del 90% en coste para el caso del dominio pequeño, mientras que para el caso del dominio grande la reducción mínima es del 70%. Los algoritmos de Búsqueda Multidireccional y Búsqueda Compás presentan un comportamiento bueno, pero algo peor que el de Nelder-Mead.

Para el caso de la apertura, al igual que en el perfil temporal, se constata el mal comportamiento del algoritmo Búsqueda Compás, el cual tiene unos porcentajes de mejora muy bajos. Es más, en un 20% de las realizaciones, para el dominio de búsqueda grande, se aumenta el coste respecto al que se tiene con el sistema sin controlar. En el extremo contrario está el algoritmo Nelder-Mead que en todos los casos consigue reducciones mayores al 40 % respecto a la situación de no control, y en un 70% de las realizaciones se obtienen reducciones de coste del 75% o más.

% Red	Situación	Algoritmo	Tasa porcentual de mejora		
			Tamaño dominio de Búsqueda (us)		
			Pequeño	Medio	Grande
0%	Cierre	Bús. Compass	100%	100%	100%
		Bús. Multidireccional	100%	100%	100%
		Nelder-Mead	100%	100%	100%
	Apertura	Bús. Compass	78%	71,00%	56%
		Bús. Multidireccional	100%	92%	100%
		Nelder-Mead	100%	100%	100%
25%	Cierre	Bús. Compass	100%	100%	100%
		Bús. Multidireccional	100%	100%	100%
		Nelder-Mead	100%	100%	100%
	Apertura	Bús. Compass	40%	31,00%	31%
		Bús. Multidireccional	100%	80%	88%
		Nelder-Mead	100%	100%	100%
50%	Cierre	Bús. Compass	100%	100%	100%
		Bús. Multidireccional	100%	100%	100%
		Nelder-Mead	100%	100%	100%
	Apertura	Bús. Compass	12%	6,00%	0%
		Bús. Multidireccional	87%	68%	71%
		Nelder-Mead	93%	93%	89%
75%	Cierre	Bús. Compass	100%	93%	89%
		Bús. Multidireccional	97%	85%	91%
		Nelder-Mead	100%	93%	94%
	Apertura	Bús. Compass	0%	0,00%	0%
		Bús. Multidireccional	54%	40%	31%
		Nelder-Mead	69%	69%	69%
80%	Apertura	Bús. Compass	0%	0%	0%
		Bús. Multidireccional	41%	31%	19%
		Nelder-Mead	51%	55%	54%
90%	Cierre	Bús. Compass	75%	75%	69%
		Bús. Multidireccional	83%	69%	82%
		Nelder-Mead	100%	82%	78%
95%	Cierre	Bús. Compass	56%	56%	57%
		Bús. Multidireccional	72%	62%	70%
		Nelder-Mead	89%	75%	57%

Tabla 6. Tabla con valores significativos de reducción. Coloreados en verde las mayores reducciones para cada una de las situaciones de control. La tasa porcentual de mejora representa el porcentaje de simulaciones en las que se obtiene una reducción de coste mayor o igual a la establecida en el % de reducción.

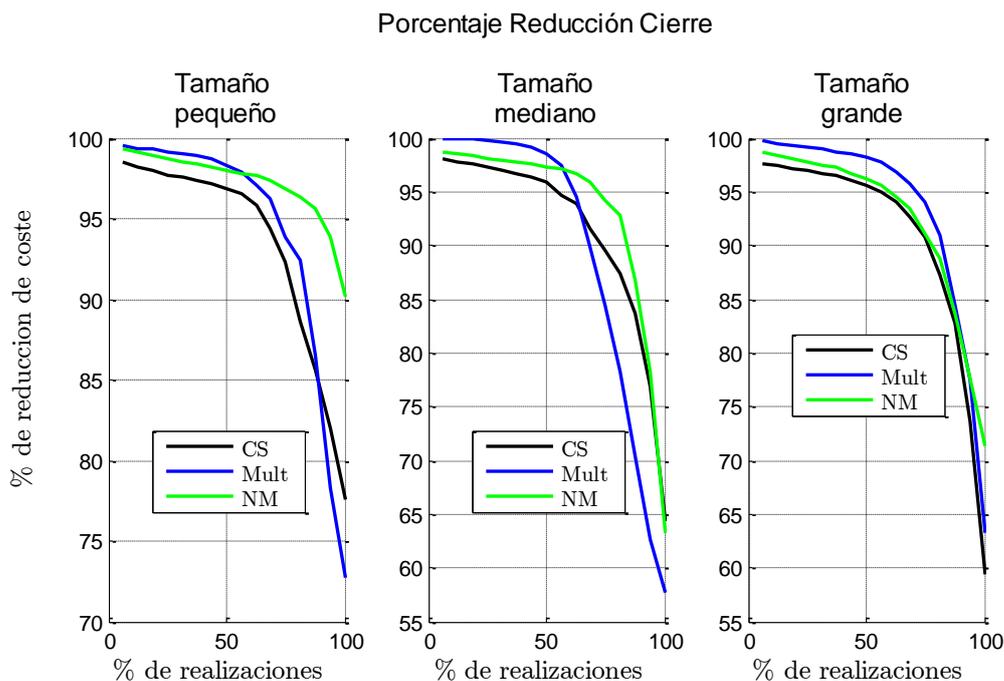


Figura 16. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de cierre.

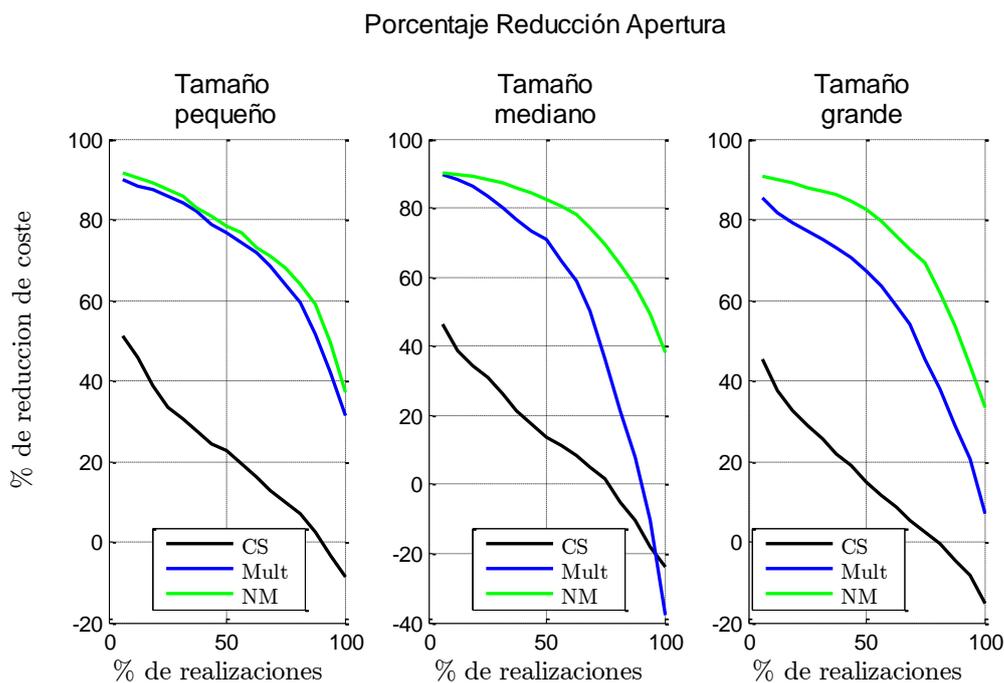


Figura 17. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de apertura.

Fijándose en los datos recogidos en la Tabla 6 se observa que en la mayoría de los casos el algoritmo de optimización que obtiene unos mejores resultados es el Nelder-Mead.

A.1.1.3 Velocidad de reducción de coste

Ahora se pasa a medir la velocidad de convergencia de cada uno de los métodos. Para ello se ha medido la pendiente de la curva de coste en cada una de las situaciones. Para obtener la pendiente se han medido las variaciones de coste que se producen cada 30 iteraciones, y se ha calculado la variación de coste medio por cada iteración.

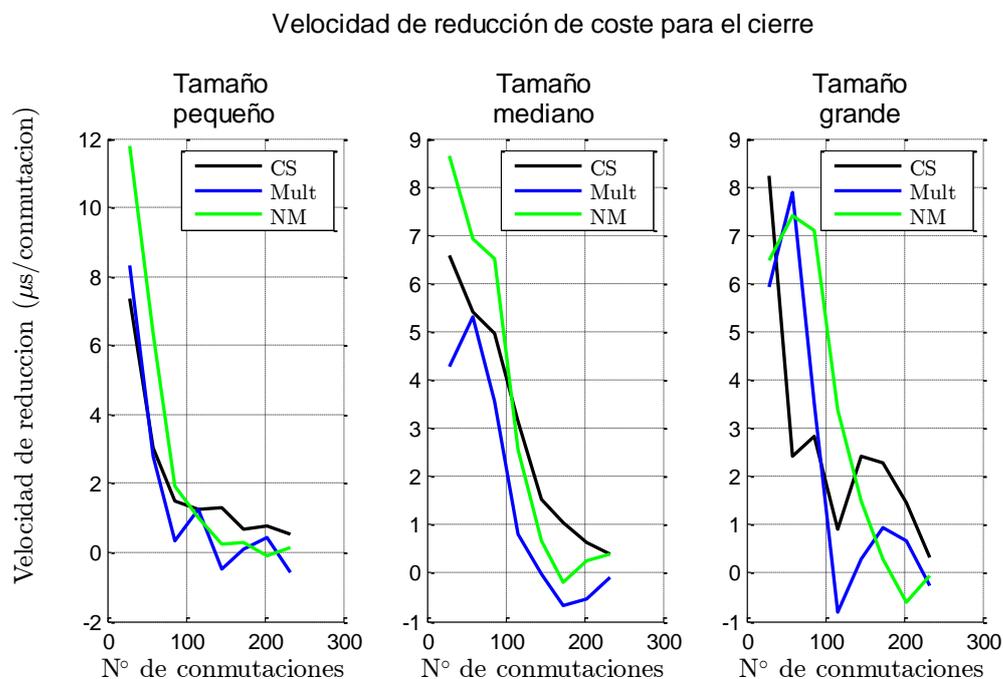


Figura 18. Variación de la velocidad de reducción de coste en el cierre, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.

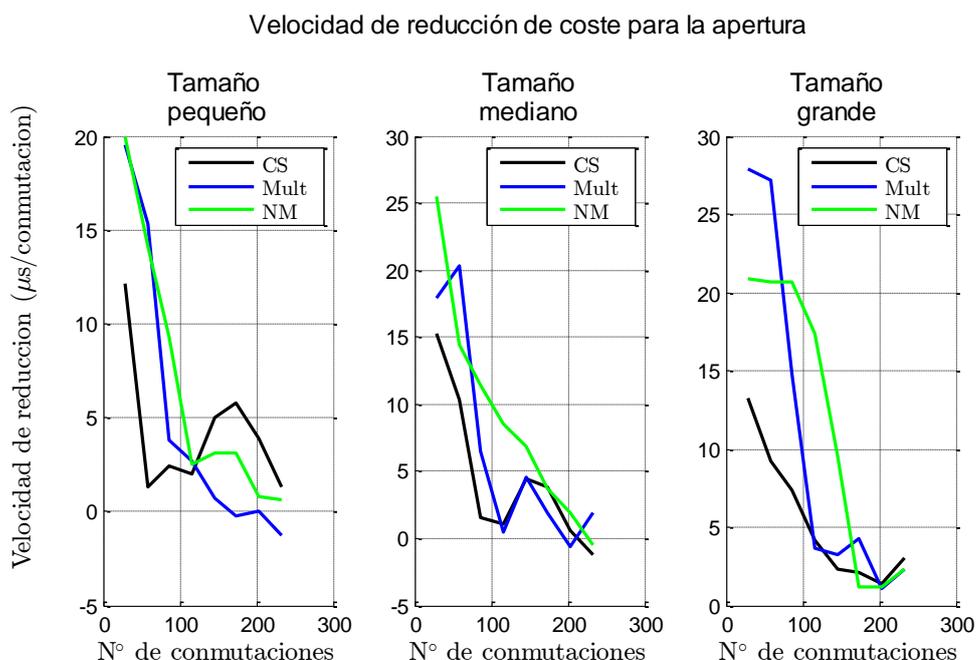


Figura 19. Variación de la velocidad de reducción de coste para la apertura, expresada en $\mu\text{s}/\text{conmutación}$ para cada una de las situaciones.

Para el caso del cierre es de resaltar que las mayores velocidades se alcanzan siempre para el método Nelder-Mead. No sólo es que se alcancen las mayores velocidades de reducción de coste, sino que además estas son mantenidas durante un mayor tiempo. Todo esto explica una convergencia más rápida que los otros dos métodos, a un mínimo menor.

Para el caso de la apertura, también se observa un mejor comportamiento del algoritmo Nelder-Mead, pero con una menor diferencia respecto al método Búsqueda Multidireccional.

Nº de iteraciones	Situación	Algoritmo	Reducción de coste en cada iteración		
			Tamaño dominio de búsqueda (us)		
			Pequeño	Medio	Grande
50 iteraciones	Cierre	Bús. Compás	74,31%	54,81%	66,71%
		Bús. Multidireccional	86,48%	73,10%	64,69%
		Nelder-Mead	71,89%	58,98%	53,06%
	Apertura	Bús. Compás	5,52%	-1,17%	-6,89%
		Bús. Multidireccional	51,06%	21,13%	10,67%
		Nelder-Mead	48,94%	39,12%	16,86%
100 iteraciones	Cierre	Bús. Compás	83,12%	74,58%	74,78%
		Bús. Multidireccional	92,47%	90,05%	92,20%
		Nelder-Mead	96,84%	79,96%	78,28%
	Apertura	Bús. Compás	8,67%	11,96%	3,73%
		Bús. Multidireccional	17,62%	-33,69%	-47,41%
		Nelder-Mead	66,11%	56,91%	43,01%
150 iteraciones	Cierre	Bús. Compás	88,63%	85,07%	78,95%
		Bús. Multidireccional	96,37%	93,81%	90,72%
		Nelder-Mead	92,60%	91,39%	92,20%
	Apertura	Bús. Compás	12,45%	13,13%	10,36%
		Bús. Multidireccional	72,14%	48,37%	50,31%
		Nelder-Mead	69,63%	69,03%	68,51%
200 iteraciones	Cierre	Bús. Compás	91,59%	90,99%	87,90%
		Bús. Multidireccional	95,70%	92,67%	90,11%
		Nelder-Mead	97,58%	92,40%	93,61%
	Apertura	Bús. Compás	12,45%	13,13%	10,36%
		Bús. Multidireccional	71,49%	51,84%	58,57%
		Nelder-Mead	74,91%	75,48%	69,24%

Tabla 7. Tabla de reducción de coste para determinado número de iteraciones. Medida de la velocidad de convergencia de cada método en cada una de las situaciones.

A.1.2. Comparativa señales de control

En este apartado, una vez realizadas las medidas sobre los distintos algoritmos, y que se ha determinado que el método de Nelder-Mead es el mejor, se pasa a testar este algoritmo de optimización con las distintas señales de control. Para comparar las

distintas señales se ha procedido de igual modo que con la comparación de algoritmos. Se han realizado un total de 20 realizaciones con 300 conmutaciones para cada situación. Para cada señal se han probado tres dominios de búsqueda de distintos tamaños, siendo estos diferentes para cada tipo de señal.

Las gráficas y variables de interés van a ser las mismas que en la comparativa de algoritmos, mientras que las conclusiones se extraerán de forma similar.

A.1.2.1 Evolución del coste

En esta parte se analiza cada señal de control según su perfil temporal, viendo cómo varía el coste en cada conmutación. Para mejorar la visualización de los resultados se representa la convolución de una ventana rectangular con la media de las 20 simulaciones para cada una de las situaciones, de manera que la curva queda alisada.

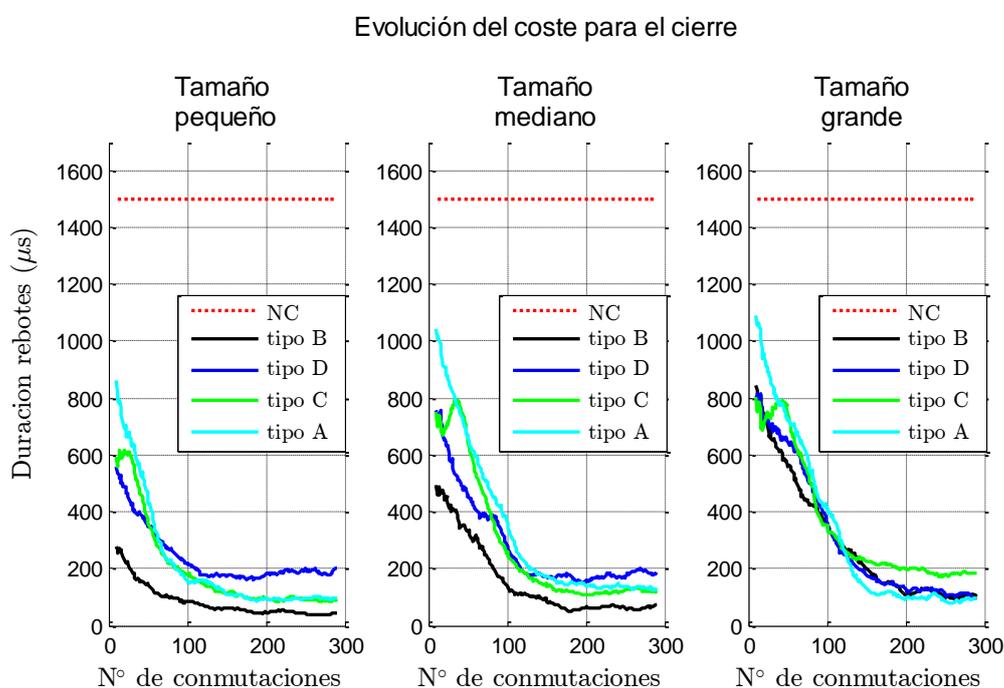


Figura 20. Evolución del coste según las iteraciones para cada una de las señales de control y situaciones en el caso del cierre.

Se observa en la Figura 20 que para la operación de cierre los resultados para cada una de las situaciones de las cuatro señales de control son bastante similares entre sí. Únicamente cabe remarcar que es la señal de tipo B la que consigue una duración de los rebotes algo menor que el resto, si bien es verdad que ya parte de un coste menor que las otras tres señales.

Para el caso de la apertura se vislumbran dos grupos diferentes. Por un lado, las señales C y D con transición suave, con un coste mayor que el segundo grupo, el de las señales A y B. Fijándose únicamente en las señales A y B, se observa que para los dominios de búsqueda pequeños, la señal B presenta un mejor comportamiento que la señal A. Este mejor comportamiento se ve reducido para los dominios de búsqueda de mayor tamaño.

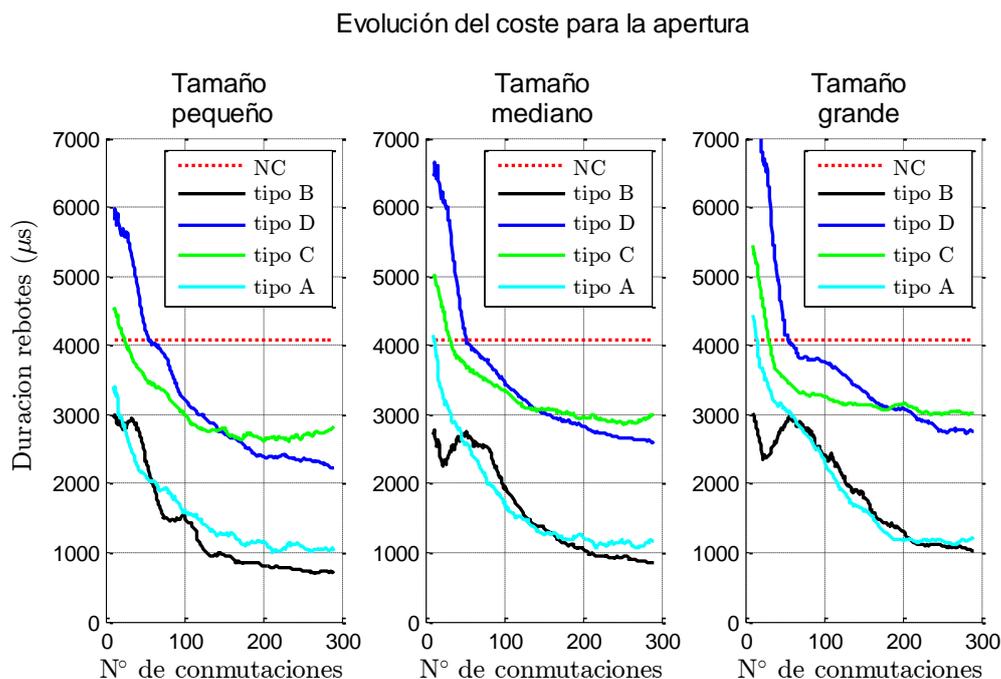


Figura 21. Evolución del coste según las iteraciones para cada una de las señales de control y situaciones en el caso de la apertura.

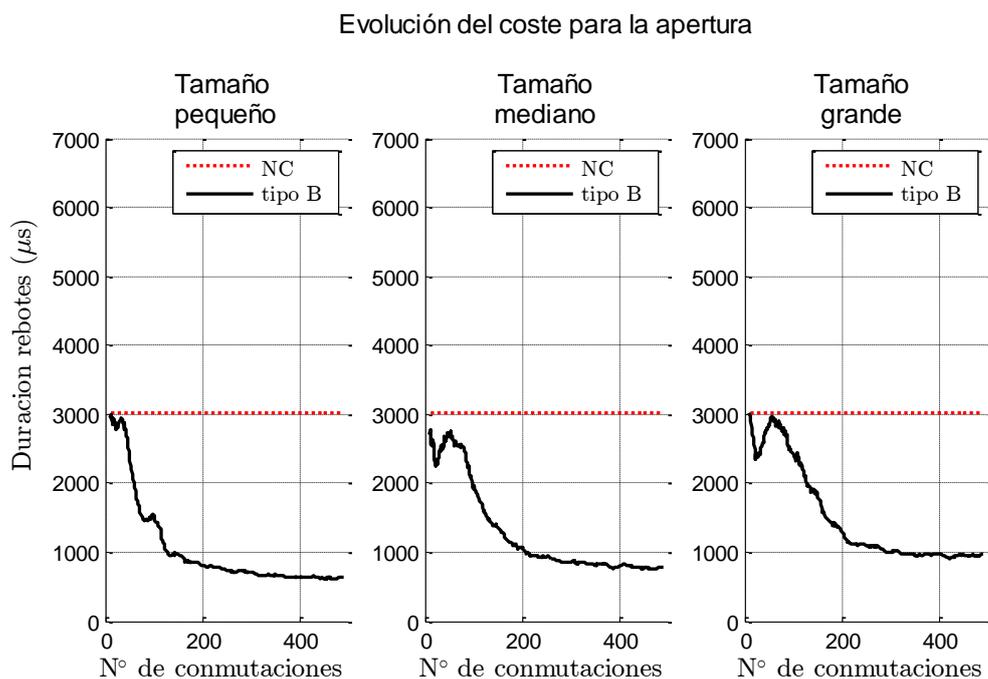


Figura 22. Evolución del coste de apertura según las iteraciones para la señal B en un total de 500 conmutaciones

Al observar la evolución del coste de la apertura para la señal B en la Figura 21 y ver que la pendiente de la curva no es nula en la conmutación 300, cabe preguntarse si se ha alcanzado el estacionario o el coste puede seguir reduciéndose. Por ello, se ha vuelto a realizar el mismo ensayo para la señal de tipo B, pero en este caso para 500

conmutaciones. En la Figura 22 se comprueba que el coste sigue reduciéndose levemente hasta la conmutación 500.

Por otro lado, para la señal de tipo D se ha comprobado que el coste mínimo se obtiene cuando el tiempo de transición es nulo, es decir, cuando la señal corresponde a una de tipo A.

A.1.2.2 Mejora de coste

En esta parte, al igual que en la comparación de los algoritmos de optimización, no se estudiará la media de las simulaciones sino que se verá para qué porcentaje de realizaciones se alcanza cierto nivel de reducción de coste.

Para el caso del cierre, la señal B asegura un 92% de reducción de coste para el dominio de búsqueda pequeño y del 77% para el dominio de mayor tamaño. Es en cualquier caso la señal que presenta un mejor comportamiento. Sin embargo, este mejor comportamiento respecto a las otras señales se ve reducido conforme el dominio de búsqueda aumenta. Por otro lado, se observa un hecho a priori contradictorio que es el mejor comportamiento de la señal D conforme aumenta el tamaño de búsqueda. La explicación a este hecho podría ser que para los dominios de búsqueda pequeño y mediando no se abarque el mínimo de la función de coste, pero sí para el dominio grande.

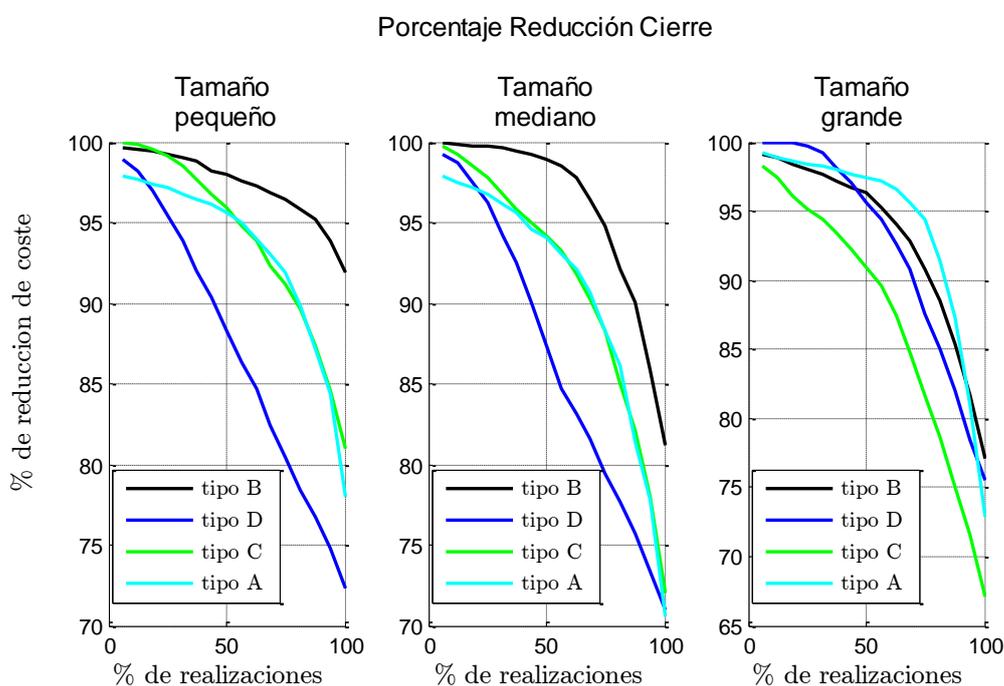


Figura 23. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de cierre.

Para el caso de la apertura también destaca el buen comportamiento de la señal B, la cual asegura una reducción de coste del 70% en el dominio de búsqueda pequeño, y de hasta el 50% para el caso del dominio grande. Al igual que se ha constatado

anteriormente, el mejor comportamiento de la señal B se ve reducido conforme aumenta el tamaño de búsqueda, teniendo la señal A en ciertas ocasiones un mejor comportamiento.

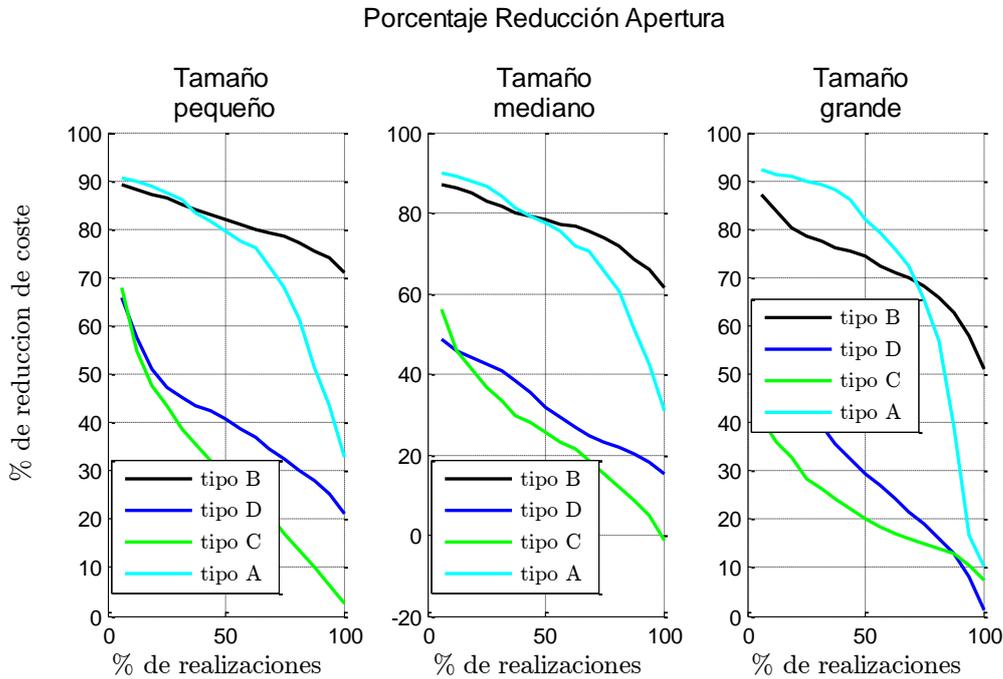


Figura 24. Porcentaje de realizaciones (eje de abscisas) en los que se consigue cierta reducción de coste (eje de ordenadas) para cada una de las situaciones. Caso de apertura.

A.1.2.3 Velocidad de reducción de coste

Considérese ahora la velocidad de convergencia de cada una de las señales de control, de igual modo que se ha hecho para la comparativa de algoritmos.

Para el caso del cierre son las señales A y C, señales de dos únicos parámetros, las que alcanzan unas mayores velocidades de reducción de coste. Es este número reducido de parámetros el que permite que la reducción sea más rápida. En el caso de las señales D y B, tres y cuatro parámetros respectivamente, las velocidades de reducción son menores. Sin embargo, el menor coste de partida de la señal B permite a esta obtener unos mejores resultados en cuanto a duración de los rebotes.

En el caso de la apertura el comportamiento parece opuesto al del cierre, teniendo las señales B y D unas mayores velocidades de reducción. También se constata el hecho de que para los dominios de mayor tamaño, la señal B tiene un tramo inicial en el que se produce un aumento de la velocidad de convergencia, hasta alcanzar un máximo, momento en el cual se produce una desaceleración hasta que la velocidad llega a cero.

Velocidad de reducción de coste para el cierre

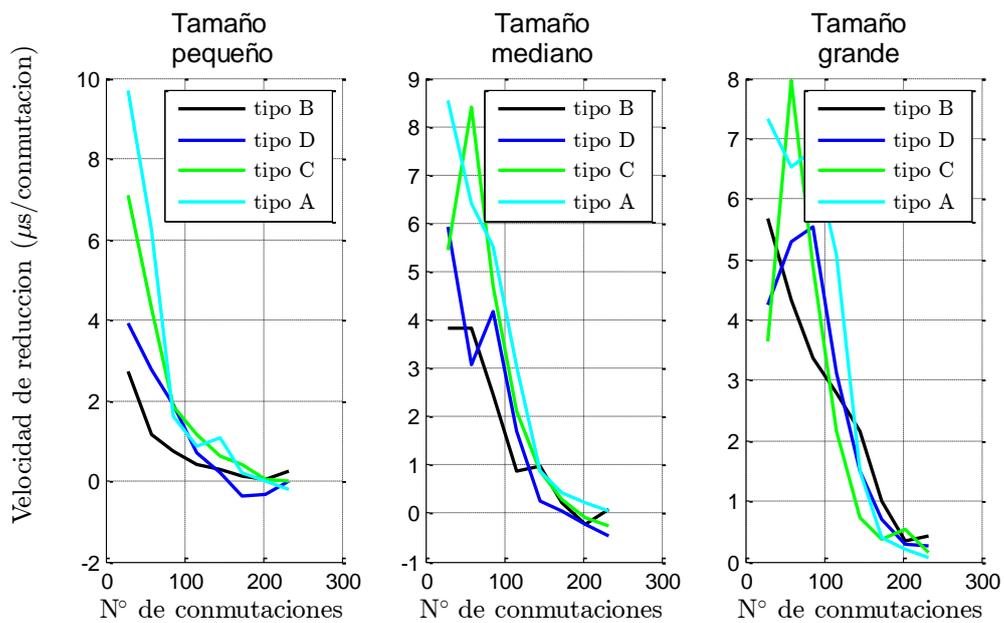


Figura 25. Variación de la velocidad de reducción de coste para el cierre, expresada en μs/commutación para cada una de las situaciones.

Velocidad de reducción de coste para la apertura

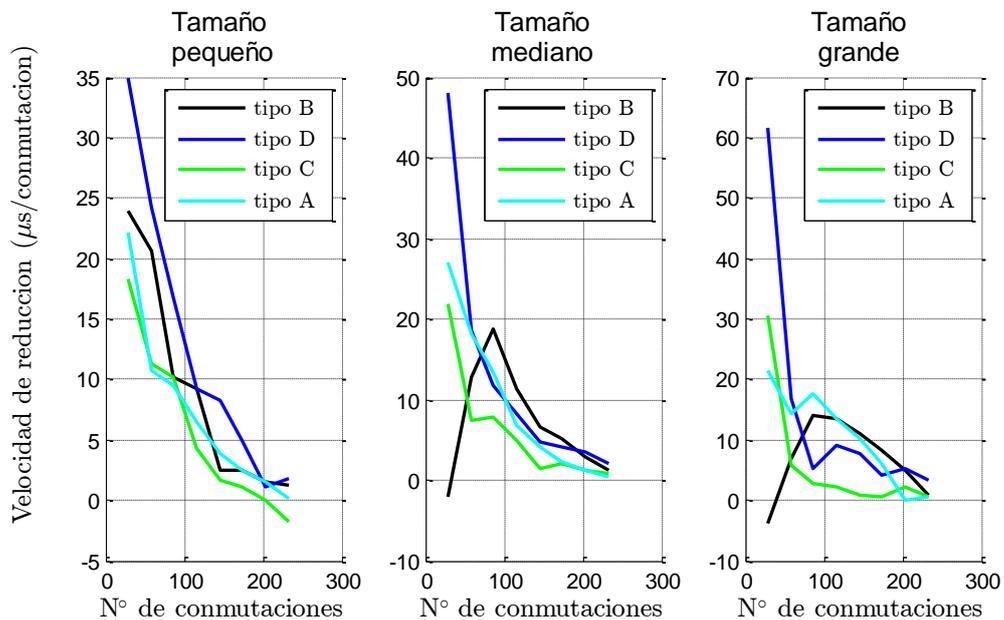


Figura 26. Variación de la velocidad de reducción de coste para la apertura, expresada en μs/commutación para cada una de las situaciones.

Anexo 2. Implementación del Microcontrolador

A.2.1 Descripción mediante máquina de estados

En esta parte se va a mostrar una representación simplificada de las distintas estrategias de control, semejante a la de una máquina de estados, la cual ha sido utilizada para la facilitar la concepción del código de control.

La variable $Estado_m$ jugará el mismo papel que el estado del sistema. Este estado irá variando llevando a cabo distintas acciones para los distintos algoritmos. Se ha representado únicamente el proceso de cierre (*making*), razón por la cual aparecen todos los superíndices y subíndices m . El proceso de apertura (*breaking*) sería completamente semejante.

A.2.1.1 Máquina de estados para Búsqueda Compás

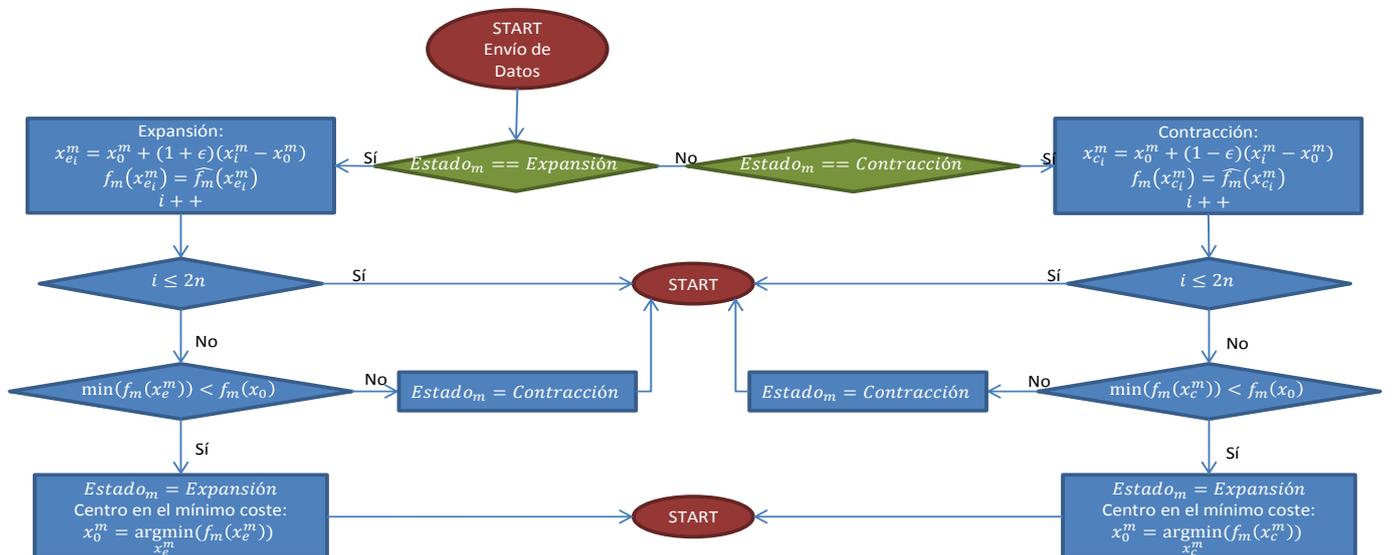


Figura 27. Descripción esquemática mediante máquina de estados del algoritmo de optimización Búsqueda Compás.

A.2.1.2 Máquina de estados para Búsqueda Multidireccional

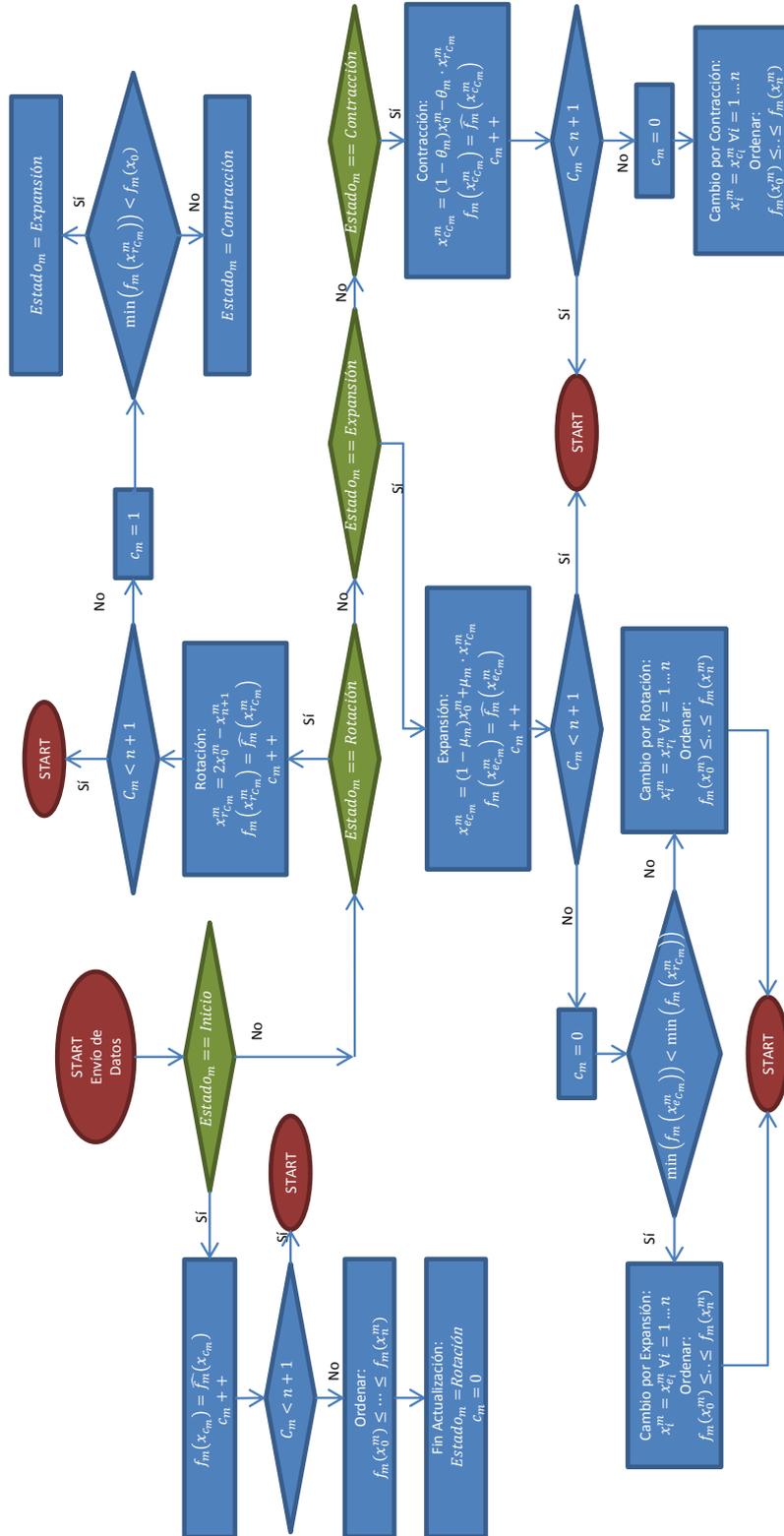


Figura 28. Descripción esquemática mediante máquina de estados del algoritmo de optimización de Búsqueda Multidireccional.

A.2.1.3 Máquina de estados para Nelder-Mead

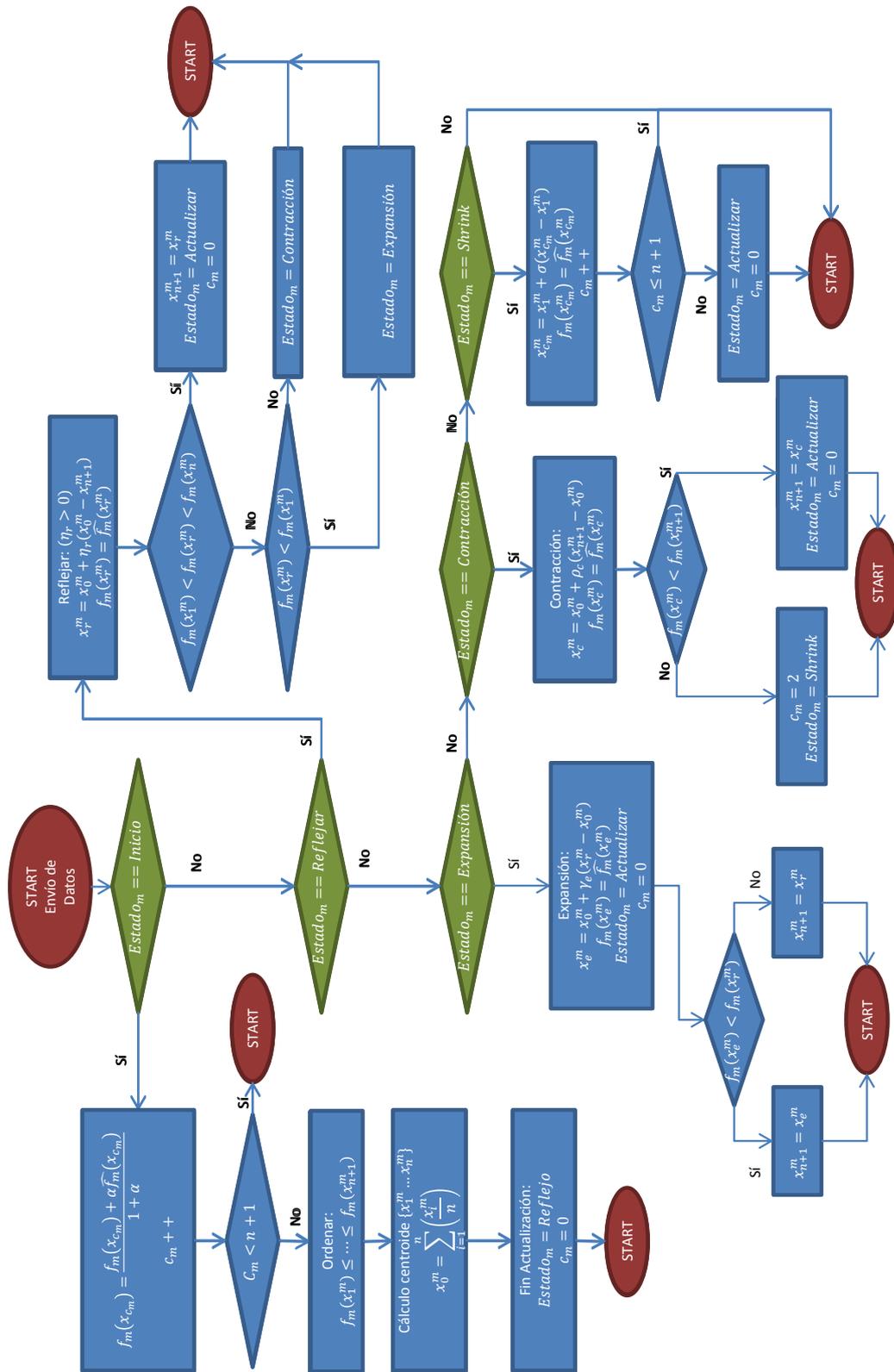


Figura 29. Descripción esquemática mediante máquina de estados del algoritmo de optimización Nelder-Mead.