

Proyecto Fin de Máster  
Máster universitario en Automática, Robótica y  
Telemática

Control predictivo aplicado al modelo macroscópico  
del tráfico en una ciudad

Autor: Juan Alfonso Serrano Viseas

Tutor: Eduardo Fernández Camacho

Departamento de Ingeniería de Sistemas y  
Automática

Escuela Técnica Superior de Ingeniería

Sevilla, 2017





Proyecto Fin de Máster  
Máster Universitario en Automática, Robótica y Telemática

# **Control predictivo aplicado al modelo macroscópico del tráfico en una ciudad**

Autor:

Juan Alfonso Serrano Viseas

Tutor:

Eduardo Fernández Camacho

Catedrático de la Universidad de Sevilla

Departamento de Ingeniería de Sistemas y Automática

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2017



Proyecto Fin de Máster: Control predictivo aplicado al modelo macroscópico del tráfico en  
una ciudad

Autor: Juan Alfonso Serrano Viseas

Tutor: Eduardo Fernández Camacho

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2017

El Secretario del Tribunal



*A mi familia*

*A mis maestros*







# Agradecimientos

---

Antes de hablar de nadie debería agradecer a mi familia y a mi pareja el apoyo recibido durante los años del máster que, de alguna forma, se ven reflejado en este trabajo. Han sido duros, y quizás más para vosotros que habéis tenido que aguantarme día a día. Por todo esto os doy las gracias, ya que no habría llegado hasta aquí si no fuera así.

Tras esto, agradecer a mi tutor, D. Eduardo Fernández Camacho no solo por la oportunidad que me dio al aceptar este proyecto de forma improvisada, si no por su inestimable ayuda y preocupación en el desarrollo del mismo.

A Filibeto Fele por su ayudan en todo el desarrollo del proyecto, en especial en la segunda parte de este. Sin su ayuda no habría sido posible.

Y no puedo acabar sin agradecer a mis compañeros de clase con los que he caminado para llegar al punto en el que me encuentro ahora.

Gracias a todos,  
Juan Alfonso Serrano Viseas



# Resumen

---

El trabajo consiste en la estimación de un modelo macroscópico del comportamiento del tráfico urbano (en especial de vehículos eléctricos), y en la implementación en simulación de un control predictivo que maximice los beneficios de las estaciones de recarga.









# Índice

---

<b>Agradecimientos</b>	<b>xi</b>
<b>Resumen</b>	<b>i</b>
<b>Índice</b>	<b>i</b>
<b>Índice de Tablas</b>	<b>i</b>
<b>Índice de Figuras</b>	<b>i</b>
<b>1 Introducción</b>	<b>1</b>
<b>2 Descripción del sistema</b>	<b>3</b>
<b>3 Simulador</b>	<b>7</b>
3.1 <i>Configuración del simulador</i>	9
<b>4 Modelado del sistema</b>	<b>15</b>
4.1 <i>Simulación base</i>	15
4.2 <i>Niveles de precio y número de simulaciones</i>	16
4.3 <i>Modelo y ajuste</i>	17
4.3.1 <i>Mínimos cuadrados</i>	18

4.3.2	Red neuronal	19
4.4	<i>Modelo obtenido y resultados</i>	20
5	<b>Control predictivo</b>	<b>25</b>
5.1	<i>Juego de coaliciones aplicado al sistema</i>	26
5.2	<i>Función objetivo</i>	29
6	<b>Resultados</b>	<b>33</b>
6.1	<i>Diferencia entre Gran Coalición y no coalición</i>	34
6.2	<i>Coaliciones y efecto de los costes de mantenimiento</i>	36
7	<b>Conclusiones</b>	<b>41</b>
[8]	<b>Referencias</b>	<b>45</b>

# ÍNDICE DE TABLAS

---

Tabla 3-1 Parámetros del simulador	11
Tabla 3-2 Archivos necesarios para la simulación	12
Tabla 4-1 Porcentaje de ajuste de los modelos	22



# ÍNDICE DE FIGURAS

---

Figura 2-1 Diagrama del sistema	4
Figura 2-2 Distribución de CSs en Sevilla	6
Figura 3-1 Diagrama del simulador	9
Figura 4-1 Parte del diagrama de la red neuronal	20
Figura 4-2 Error relativo de los modelos	21
Figura 4-3 Incrementos de energía en simulaciones aleatorias	23
Figura 6-1 Diferencia entre el beneficio en una <i>GC</i> antes y después del reparto	34
Figura 6-2 Comparativa entre <i>GC</i> y sin coaliciones	35
Figura 6-3 Resultados de las coaliciones CS1-CS4 y CS2-CS3	37
Figura 6-4 Comparativa costes de mantenimiento	38









# 1 INTRODUCCIÓN

---

En los últimos años, ha habido muchísimos avances en todos los campos y de gran importancia (en el campo de las comunicaciones, en el ámbito energético, en la electrónica,...). Por ellos muchos investigadores han centrado sus esfuerzos en encontrar y perfeccionar métodos para cambiar la fuente de energía usada para el desplazamiento de los vehículos. Estos avances, han estado centrados en su mayor parte en diseñar y crear vehículos cuya fuente de energía sea la electricidad, la cual puede obtenerse de forma que no perjudique al planeta. Aunque existen desde hace décadas los motores eléctricos, hasta hace poco no se ha conseguido que tengan las prestaciones necesarias para incluirlos en un vehículo que requiere unas prestaciones exigentes, como pueden ser autonomía, rendimiento, vida útil,...

Poco a poco estas limitaciones van siendo superadas, y la cantidad de vehículos eléctricos (EVs) en las calles aumenta año tras año. Por ello es necesario la construcción de estaciones de recarga (CS) para EVs. Hay tres tipos de infraestructuras para la recarga: nivel 1 (edificios residenciales y comerciales), nivel 2 (instalaciones de carga para requerimientos especiales), y nivel 3 (carga rápida). Por otro lado, durante el día muchos EVs necesitarán realizar una carga rápida, ya que no dispondrán del tiempo para regresar a sus domicilios ni para realizar una recarga lenta. Por ello, este trabajo se centra en la gestión de CSs de carga rápida. Estas deben proporcionar información, usando tecnologías de la información aplicadas a la industria del transporte (ITS), a cada uno de los EVs del estado de sus postes de recarga, del precio de

su estación y de las colas en tiempo real. El EV decidirá, teniendo en cuenta que debe recargar una batería de 10 kWh (estación de recarga de nivel 3 demora 15 minutos en recargar esta batería), el estado de las CS y el estado del tráfico, a que estación debe ir a recargar para minimizar su gasto de tiempo y dinero.

Gracias al trabajo [1], el cual se realizó junto a AYESA usando los datos proporcionados por el proyecto Zem2All [2], para esto proyecto se tiene un simulador de tráfico para cualquier ciudad. Este simulador permite la realización del presente trabajo sin la necesidad de tener que realizar pruebas en el sistema real, ya que esto sería muy costoso por tiempo y dinero, y actualmente es prácticamente imposible.

El principal objetivo de este trabajo es maximizar los beneficios de cada una de las CSs presentes en el escenario de estudio. Para ello se usará un control predictivo, teniendo en cuenta que se conocen todos los datos necesarios para ello. Además se tendrá en cuenta que las CSs pueden cooperar unas con otras para establecer una estrategia de precios que les beneficie a todas las colaboradoras [3].

## 2 DESCRIPCIÓN DEL SISTEMA

---

En este capítulo se va a describir el sistema objeto de estudio, incluyendo número de entradas y salidas, tiempo de muestreo, restricciones y los diversos factores que puedan influir en él. El sistema objeto de estudio es el comportamiento a nivel macroscópico del tráfico en una población y su influencia en el consumo de electricidad en las Estaciones de Carga Rápida (CS) presentes en dicha población. Se trata de un sistema compuesto por un número de entradas igual al número de salidas, ya que se considera que existe una entrada y una salida por cada CS. La cantidad de kWh recargados en una estación durante el tiempo de muestreo corresponde una salida. Estas salidas se verán influidas por el número de Vehículos Eléctricos (EV) que se detengan a recargar en cada CS. Cada EV decidirá ir a recargar a una estación u otra dependiendo de varios parámetros, como son la lejanía de cada CS desde su posición actual, el precio de la energía en cada CS,... Este último parámetro, el precio de la energía en cada CS (€/kWh), es el que tomaremos como entrada del sistema, ya que es posible actuar sobre él en tiempo real y variar la demanda de energía en cada CS en función de los intereses del vendedor.

El sistema actualizará sus entradas y salidas una vez al día, por lo que como tiempo de muestreo se tomarán 24 horas. Los precios de cada CS pueden ser actualizados al principio de cada día. La cantidad de energía consumida a lo largo del día se contabiliza al finalizar este.

Aunque se considera que el tiempo de muestreo es de 24 horas, la energía contabilizada es a lo largo de 12 horas durante el día (se ignora la noche, ya que no se considera relevante). Conociendo todo esto, se considera que en el día  $k$  como entradas tenemos los precios fijados al inicio de ese día, y las salidas son las sumas de los consumos en cada CS durante todo ese día.

El interés real de este sistema se basa en establecer los cambios en la demanda de energía en función de los cambios que se apliquen en las entradas cada día. Por ello, se establecerá un punto de operación base, con la energía que se demanda para un precio determinado. A partir de este punto de operación, se consideran como entradas del sistema los incrementos de precio respecto al precio base en cada CS, y como salidas los incrementos de la demanda de energía respecto a la demanda de energía base en cada CS.

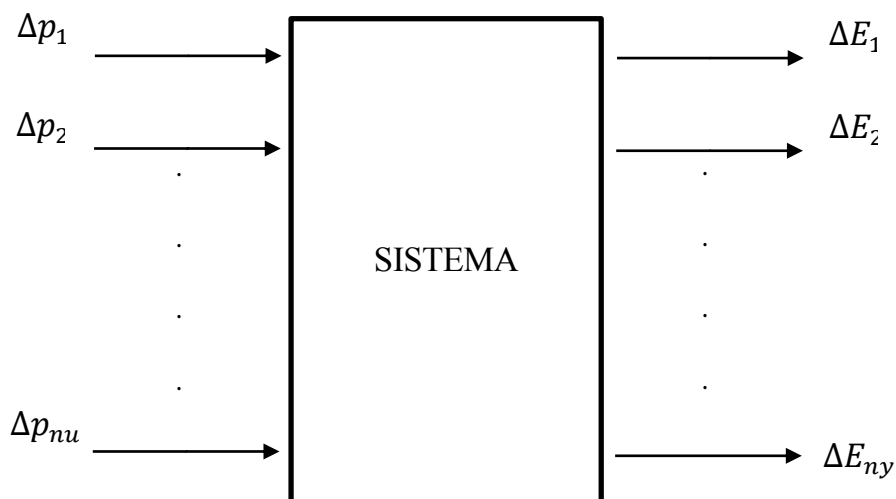


Figura 2-1 Diagrama del sistema

El principal objetivo de este trabajo es el de conseguir aplicar un control predictivo al sistema para que se maximicen los gastos y la energía demandada. Se definirá una función de coste la cual habrá que maximizar en cada uno de los tiempos de muestreo, para ir estableciendo al principio de cada día el precio en cada estación en función del objetivo marcado por la función de coste. Además, se podría establecer cooperaciones entre las distintas estaciones para maximizar sus beneficios. Todo esto podría aplicarse aplicando control predictivo al sistema.

El sistema real está compuesto por las CSs, EVs y demás vehículos en la ciudad de Sevilla. Se trata de la relación entre los precios de 4 CSs distribuidas en el mapa de Sevilla y la energía consumida en estas por 500 EVs. Estos EVs circularan por la ciudad sin interferencia de ningún otro tipo de vehículo (el número de RVs será 0). El mapa de Sevilla junto con la distribución de las CSs puede verse en la Figura 2-2.



Figura 2-2 Distribución de CSs en Sevilla

## 3 SIMULADOR

---

Como se puede intuir, no es posible hacer pruebas en el sistema real por tiempo, coste y disponibilidad de EVs y CSs. Por ello es necesario hablar de la herramienta de software libre Simulation of Urban Mobility (SUMO), un simulador a nivel microscópico que permite simular cómo se comporta un determinado volumen de tráfico (en vehículos individuales) a través de una red de carreteras dada al inicio de la simulación. La simulación permite abordar un gran número situaciones en el tráfico. Es puramente microscópico: cada vehículo es modelado explícitamente, tiene una ruta propia, y se mueve individualmente a través de la red. Las simulaciones son determinísticas por defecto pero hay varias opciones para introducir la aleatoriedad.

SUMO es una herramienta muy potente, pero por si sola no es útil para el propósito de este proyecto. Por ello se recurre al simulador macroscópico Simulador SUMO-MATLAB [1]. El simulador está formado por dos capas independiente que se comunican entre sí. La primera parte se basa en SUMO (Simulation of Urban MObility) [4], el simulador de tráfico desde un punto de vista microscópico ya descrito anteriormente. Las tareas que SUMO desempeña son la creación del vehículo en el mapa, el movimiento de los vehículos de acuerdo a las condiciones del mapa desde un arco a otro y la eliminación del vehículo del mapa cuando este llega a su arco de destino. Para SUMO no existe diferencia entre os vehículos eléctricos y los

que no lo son (RV). La segunda capa se desarrolla en MATLAB y es la responsable de gestionar la parte relativa a la energía de los EVs y la supervisión de la simulación. Las tareas que se realizan en MATLAB son el incremento de la batería durante las recargas, el decremento de la batería durante el movimiento, la toma de decisiones sobre donde, cuando y cuanto recargar, etc. Gracias al uso de Matlab como gestor de la herramienta, es el uso de un software conocido, potente, sencillo de usar, con mucha información a nuestra disposición y muy usado a nivel educacional.

La interfaz entre las dos capas se denomina TraCI4MATLAB [5] y está formado por un conjunto de funciones desarrolladas en MATLAB que usan a su vez métodos de Java para abrir un canal de comunicaciones entre las dos capas usando sockets. Este tipo de comunicación está soportado por TraCI, un módulo de SUMO. Cada vez que la simulación en MATLAB requiere de la colaboración de SUMO, se envía una petición a través de esta capa de comunicación que devuelve una respuesta si procede. Por ello SUMO es usado como un ejecutable externo sin necesidad de modificar o ni siquiera compilar ninguna parte de su código fuente.

SUMO recibe información tanto de MATLAB como desde una serie de archivos XML donde es posible establecer la geometría del grafo del mapa, las velocidades en cada uno de los arcos, el número de carriles y otros parámetros relacionados con la simulación. MATLAB también necesita leer la información contenida en archivos externos, en este caso archivos .mat, que contienen información sobre los niveles de batería inicial, los niveles de batería durante la



recarga, la demografía del mapa, etc.

Esta estructura provee al simulador de una gran flexibilidad en cuanto al cambio de rutas, número de coches, edición de EVs y CSs, además de un desacoplo entre la gestión eléctrica (Matlab) y la gestión del tráfico (SUMO).

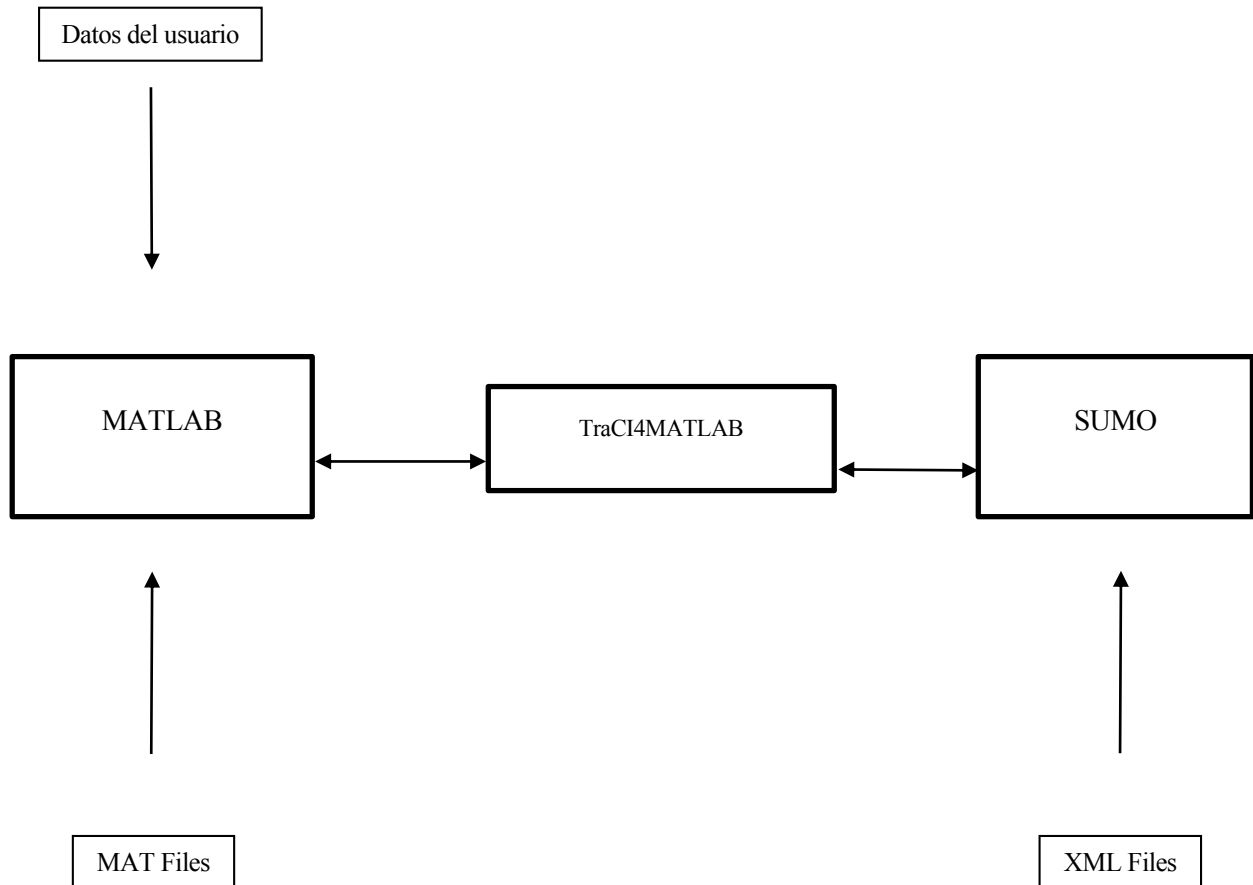


Figura 3-1 Diagrama del simulador

### 3.1 Configuración del simulador

Este simulador, como ya se ha indicado, es configurable de forma rápida desde Matlab por el usuario. A continuación se detallan en la

Tabla 3-1 los parámetros que configuran la simulación a su inicio.

<b>Parámetros de configuración</b>	
<b>Parámetro</b>	<b>Descripción</b>
SimulationID	Identificador de la simulación
Know Experience	Activa/Desactiva el uso de una experiencia previa
Experience File Name	Archivo con la experiencia previa (si es necesario)
PresetPrice	Vector de precios de las estaciones [€/kWh]
PresetPlugPower	Potencias de los enchufes de las CSs [kW]
PresetStantionPower	Potencia de las CSs [kW]
<b>Parámetros de la simulación</b>	
<b>Parámetro</b>	<b>Descripción</b>
SUMO Step	Paso de simulación en SUMO [s]
MATLAB Step	Paso de simulación en MATLAB [s]
Final Time	Tiempo final de Simulación [h]
Map Root Name	Nombre raíz del mapa
Graphic Mode	Activa/Desactiva el modo gráfico
Know Experience	Activa/Desactiva el uso de una experiencia previa
<b>Parámetros de las estaciones</b>	
<b>Parámetro</b>	<b>Descripción</b>
Random-CS Number	Activa estaciones aleatorias
From a file	Activa estaciones cargadas desde un archivo
<b>Parámetros de los vehículos</b>	
<b>Parámetro</b>	<b>Descripción</b>

EV-Number	Numero de Evs a simular
EV-Up to	Tiempo final de simulación para EVs [h]
EV-Color	Color de los EVs en el modo gráfico [RGB]
RV-Number	Numero de Rvs a simular
RV-Up to	Tiempo final de simulación para RVs [h]
RV-Color	Color de los RVs en el modo gráfico [RGB]
Initial Battery Distribution	Archivo con la información sobre la distribución del nivel inicial de batería
Battery to Charge Distribution	Archivo con la información sobre la distribución de los niveles de bateria en las recargas

#### Parámetros de rutas

Parámetro	Descripción
Mean	Longitud media de las rutas [m]
Deviation	Desviación estándar de longitud de las rutas [m]
Minimun	Longitud mínima de las rutas [m]
Static-Time in the day	Activa modo estático. Instante del día en que se simula [h]
Dynamic-Initial time in the day	Activa modo dinámico. Tiempo inicial de la simulación [h]

Tabla 3-1 Parámetros del simulador

Por otro lado, como ya se ha indicado antes, son necesarios una serie de archivos para el funcionamiento del simulador. Estos archivos deben ser proporcionado a Matlab y a SUMO, y es posible ver la lista completa en la Tabla 3-2.

<b>Archivos necesarios para SUMO</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>ROOT.sumoconfig</i>	Configuración de SUMO (la simulación lo crea automáticamente)
<i>ROOT.net.xml</i>	Mapa en un formato adecuado para SUMO
<i>ROOT.rou.xml</i>	Rutas de los vehículos no creados desde Matlab
<b>Archivos necesarios para Matlab</b>	
<b>Nombre</b>	<b>Descripción</b>
<i>ROOT.edgeinfo.mat</i>	Arcos del mapa: ID y posición
<i>ROOT_demography.mat</i>	Información demográfica para la creación de rutas
<i>ROOT_graph_SUMO.mat</i>	Mapa en un formato adecuado para Matlab
<i>ROOT.stationtest.mat</i>	Archivo necesario con las CSs ya definidas. Necesario cuando se selecciona el uso de las estaciones desde un archivo.
<i>InitialInfo.mat</i>	Información sobre los niveles iniciales de batería
<i>BatteryInformation.mat</i>	Información sobre los niveles de batería en el proceso de recarga

Tabla 3-2 Archivos necesarios para la simulación

En esta sección es necesario destacar, que el propio simulador tiene ya implementada una función muy útil. Se trata de la posibilidad de ejecutar varias simulaciones seguidas indicándole al programa únicamente los vectores de precio, potencia de los enchufes y potencia para cada CS, además del vector de IDs para las simulaciones. Esta función es muy útil y necesaria, ya que a la hora de ajustar un modelo que tenga un comportamiento similar al del simulador (considerado como sistema real en este caso) serán necesarias muchas simulaciones para distintas combinaciones de precios.

Además de las funciones y características comentadas anteriormente, este simulador tiene algunas más que, como no han sido usadas y no es el objetivo de este trabajo explicar al detalle el simulador, no serán mencionadas ni explicadas.



# 4 MODELADO DEL SISTEMA

---

El modelado del sistema es una parte esencial para poder aplicar un control y maximizar los beneficios. Para que el modelo sea fiable y tenga un comportamiento similar al sistema cerca del punto operación sobre el que se trabaja, se debe elegir con cuidado el tipo de modelo a utilizar y el método de ajuste.

## 4.1 Simulación base

Para empezar es necesario definir el punto de trabajo, para ello se fijan los precios base,  $p_b$ , de los que ya se ha hablado en otro capítulo de este trabajo, y se realiza una simulación para esos precios. La energía base,  $E_b$ , es la suma de toda la energía vendida en cada una de las 4 estaciones de sevilla durante el tiempo de muestreo (12 horas). El punto de trabajo (precio y energía base) son los indicados en las ecuaciones (4.1) y (4.2).

$$p_b = [0,3 \quad 0,3 \quad 0,3 \quad 0,3] \text{ €/kWh} \quad (4.1)$$

$$E_b = [497,5 \quad 737,08 \quad 530 \quad 260] \text{ kWh} \quad (4.2)$$

La simulación base ha sido realizada con el número de vehículos que se detalla en el capítulo 0.

A partir de esta simulación base se realizarán todas las demás simulaciones necesarias para la obtención del modelo, es decir, las decisiones aleatorias que toma el simulador cada vez que se le hace funcionar, serán las mismas en la simulación base, el único cambio será el precio de la energía en cada estación.

## 4.2 Niveles de precio y número de simulaciones

Para obtener el modelo es necesario tener el máximo número de simulaciones posibles para el rango de precios en el que se va a ajustar el modelo, teniendo en cuenta que se necesita un conjunto de datos para el ajuste del modelo y otro conjunto para validar el modelo. El rango de precios para el ajuste del modelo es de  $\pm 0,025 \text{ €/kWh}$ . Para obtener un modelo fiable, se debe hacer una simulación de 12 horas por cada combinación posible. Se fijan 5 niveles de precios [0,275 0,2875 0,3 0,3125 0,325]. Para realizar el mínimo de simulaciones posibles, se va a simular todas las combinaciones posibles entre los 5 niveles de precios.

$$n_{simulaciones} = n_{niveles}^{nCS} = 625 \text{ simulaciones} \quad (4.3)$$

Donde  $n_{simulaciones}$  es el número de simulaciones a realizar,  $nCS$  es el número de CSs y



$n_{niveles}$  es el número de niveles de precio a combinar. De esta forma se obtienen 625 datos para el ajuste del modelo.

Por otra parte el conjunto de simulaciones para validar el modelo serán realizadas con valores de precios aleatorios para las 4 CSs, de forma que el número de simulaciones para la validación sea el 30% del total de la suma de estas y el número de simulaciones para el ajuste del modelo (268 simulaciones para la validación). Los valores de precio aleatorio para estas simulaciones estarán siempre en el rango de  $\pm 0,03 \text{ €/kWh}$ .

### 4.3 Modelo y ajuste

El modelo no depende del estado anterior de los precios o la energía, se trata de un modelo que ajusta cada una de las salidas dependiendo únicamente del incremento actual del precio respecto del precio base, es decir, este modelo puede ser, en principio, un conjunto de 4 funciones, una por cada CS, que devuelve un incremento de energía vendida en cada estación en función del incremento de precio respecto del precio base en cada CS. Teniendo en cuenta (4.4) y (4.5) el modelo buscado es el que aparece en la ecuación (4.6).

$$\Delta E(k) = E(k) - E_b \quad (4.4)$$

$$\Delta p(k) = p(k) - p_b \quad (4.5)$$

$$\Delta E(k) = B \cdot \Delta p(k) \quad (4.6)$$

Donde  $\Delta E(k)$  y  $\Delta p(k)$  son respectivamente las variaciones de energía y de precio respecto a

la simulación base. Se decide que  $B$  puede ser estimada de dos formas, y por lo tanto, puede ser dos tipos de modelos: una matriz cuadrada ajustada por mínimos cuadrados o una red neuronal.

#### 4.3.1 Mínimos cuadrados

$B$  es una matriz cuadrada de dimensión  $n$ , siendo  $n$  el número de CSs presentes en la ciudad. Para estimarla se usa la función *armax* de Matlab [6], la cual es capaz de ajustar un modelo ARMAX en función de una serie de datos que dependen del tiempo. En este caso no es necesaria ni deseada esta dependencia temporal que puede estimar la función, por ello, primero se debe tener en cuenta que el modelo ARMAX tiene la forma de la ecuación (4.7).

$$A(q) \cdot y(t) = B(q) \cdot u(t - n_k) + C(q) \cdot e(t) \quad (4.7)$$

Donde,

$$A(q) = 1 + a_1 \cdot q^{-1} + \dots + a_{n_a} \cdot q^{-n_a} \quad (4.8)$$

$$B(q) = b_1 + b_2 \cdot q^{-1} + \dots + b_{n_b} \cdot q^{-n_b+1} \quad (4.9)$$

$$C(q) = 1 + c_1 \cdot q^{-1} + \dots + c_{n_c} \cdot q^{-n_c} \quad (4.10)$$

Siendo  $n_a$  el número de polos,  $n_b$  el número de ceros,  $n_k$  el retardo del sistema y  $n_c$  el número

de coeficientes de  $C$ .

Para que el modelo ARMAX tenga la forma del modelo buscado, se debe indicar a la función *armax* que las matrices  $A$ ,  $B$  y  $C$  deben tener la forma de (4.11), (4.12) y (4.13).

$$A(q) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (4.11)$$

$$B(q) = \begin{bmatrix} b_{11} & b_{12} & b_{13} & b_{14} \\ b_{21} & b_{22} & b_{23} & b_{24} \\ b_{31} & b_{32} & b_{33} & b_{34} \\ b_{41} & b_{42} & b_{43} & b_{44} \end{bmatrix} \quad (4.12)$$

$$C(q) = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad (4.13)$$

#### 4.3.2 Red neuronal

$B$  puede también tomar la forma de una red neuronal. Para su ajuste es posible usar la toolbox de Matlab *Neural Net Fitting* [7], la cual permite crear y entrenar una red neuronal con un conjunto de datos. Ese conjunto de datos son los definidos en la sección 4.2, de los cuales los datos para el ajuste del modelo (70%) se usan para entrenar la red neuronal, y del otro 30%, se usa una mitad para validar la red y la otra mitad para testear la red neuronal.

Entonces el primer paso es definir el conjunto de datos a usar, lo cual ya está hecho. Tras ello es necesario crear la red neuronal y elegir sus características. La red neuronal creada tiene 15 capas ocultas (con un tamaño de 15 nodos cada una) cuya función de transferencia es

sigmooidal y una capas de salida (con un tamaño de 4 nodos) con función de transferencia sigmooidal (Figura 4-1).

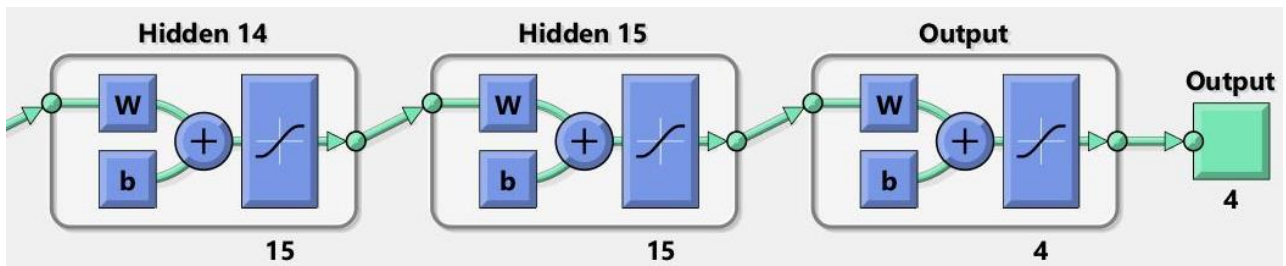


Figura 4-1 Parte del diagrama de la red neuronal

Matlab entrena la red creada para que el error cuadrático medio sea el menor, y así conseguir el mejor ajuste posible.

#### 4.4 Modelo obtenido y resultados

Tras realizar  $n_{simulaciones}$ , se obtiene un vector de precios y un vector de energía vendida por cada simulación, a los cuales se les resta el precio base y la energía vendida base respectivamente para obtener  $\Delta p(k)$  y  $\Delta E(k)$ . A partir de estos datos, y configurando la función *armax* de forma correcta, es posible obtener el modelo usando Matlab. En la ecuación (4.14) se muestra el modelo obtenido con Matlab y la función *armax*.

$$B = \begin{bmatrix} -19838 & 6745 & 5688 & 6919 \\ 7049 & -20030 & 5546 & 8276 \\ 5594 & 5775 & -21636 & 10901 \\ 7081 & 7503 & 10433 & -26228 \end{bmatrix} \quad (4.14)$$

Por otro lado también se obtiene una red neuronal, que junto con el modelo obtenido con mínimos cuadrados, deberán ser validados.

Como segundo paso tras el ajuste, el comportamiento del sistema real y los modelos deben ser comparados para verificar su funcionamiento y que se comportan de forma similar al sistema real en torno al punto de funcionamiento base. Para ello se genera 10 vectores de precios aleatorios (con un rango de  $\pm 0.025 \text{ €/kWh}$ ) para realizar 10 simulaciones y estimar el error y la validez de cada modelo.

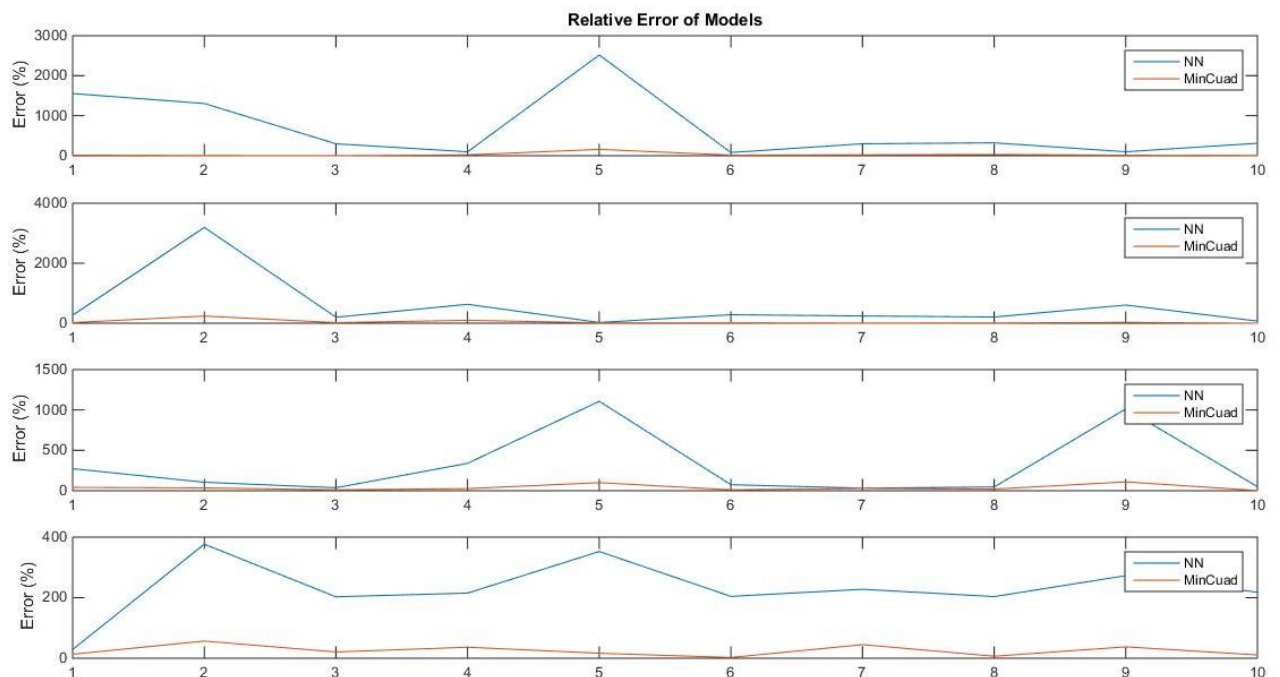


Figura 4-2 Error relativo de los modelos

La Figura 2-1 muestra el error relativo de ambos modelos para cada una de las 10 simulaciones respecto de los resultados del simulador. El modelo de Red Neuronal es mucho peor que el modelo de Mínimos Cuadrados. Esta conclusión también puede observarse en el porcentaje de ajuste, en la Tabla 4-1.

	CS1	CS2	CS3	CS4
Min Cuad	60,86%	40,27%	89,3%	43,6%
Red Neuronal	-570%	-21,25%	13,8%	40%

Tabla 4-1 Porcentaje de ajuste de los modelos

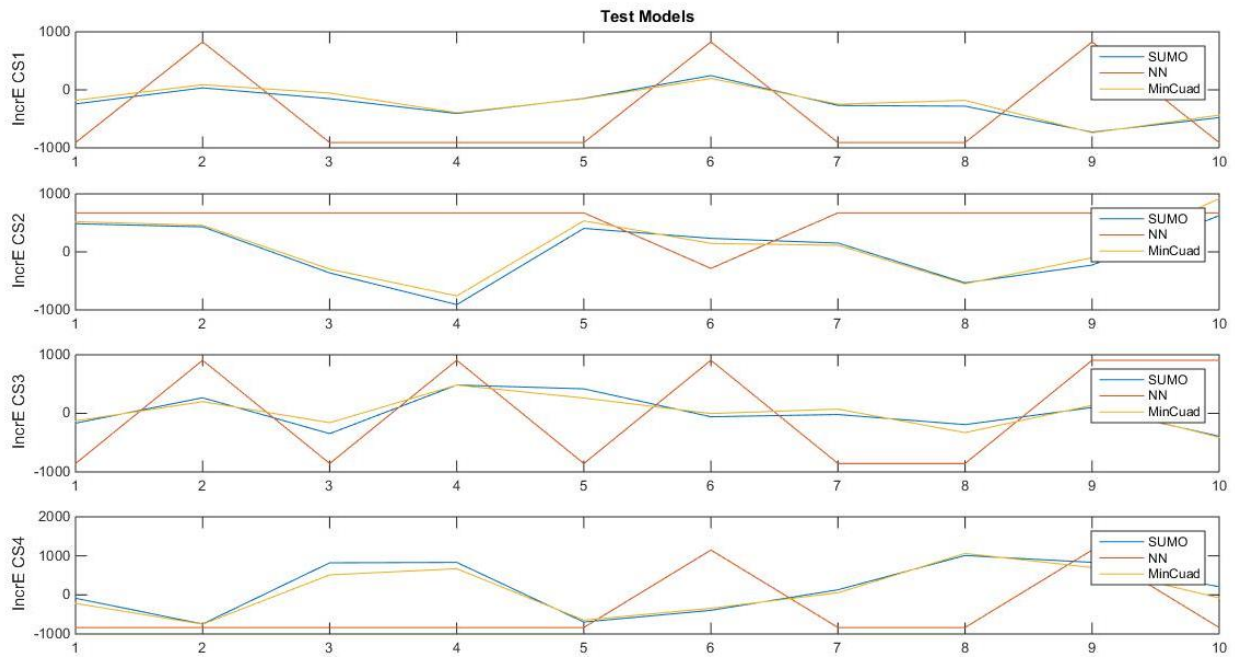


Figura 4-3 Incrementos de energía en simulaciones aleatorias

Como último paso de verificación, se debe tener en cuenta las diferencias entre el incremento de energía vendida respecto al precio base que resulta en SUMO y las que se obtienen de ambos modelos. La Figura 2-1 muestra esta comparación, y a simple vista la conclusión es la misma que la obtenida con el error relativo y con el porcentaje de ajuste. Por lo tanto, el modelo que se usará para el diseño del control es el modelo obtenido por mínimos cuadrados.





## 5 CONTROL PREDICTIVO

---

El objetivo de aplicar control predictivo al sistema es el de conocer los posibles beneficios de cada CS en el horizonte de predicción definido. Hay muchas posibilidades en cuanto a los posibles beneficios de cada CS, ya que se pueden formar coaliciones entre las distintas CSs para mejorar sus beneficios. Por ello, el control debe tener en cuenta las posibles coaliciones así como el comportamiento de las estaciones que no formen coaliciones. Debe ser capaz de elegir el precio que, teniendo en cuenta todo lo anterior, maximice sus beneficios. Hay varios factores que influyen en este control: los gastos de las CS, los precios de las demás CSs, el comportamiento de los EVs cuando se modifica el precio de la energía,... Por ello es necesario definir una función de coste válida que englobe de alguna forma todos estos factores de forma real, para poder predecir y anticipar el comportamiento de los EVs y maximizar los beneficios de las CSs.

Como puede intuirse, el coste computacional de este control es alto, ya que necesita realizar una gran cantidad de operaciones. Por suerte, si en algún momento fuese posible implementar este control, en un principio esto no sería un problema. Como el tiempo de muestreo definido es de un día, se pueden realizar todos los cálculos necesarios durante la noche, tras obtener los datos de la energía vendida ese día, y estimar el precio de la energía que se debe fijar para el día siguiente en función de todos los factores ya mencionados anteriormente. Por otro lado, si en algún momento se deseara implementar un control con un tiempo de muestreo menor, que

modificara los precios durante el día, se deberían estudiar el tiempo de muestreo mínimo para que sea posible llevar a cabo todas las operaciones.

### 5.1 Juego de coaliciones aplicado al sistema

Cada uno de los dueños de las CSs tiene la posibilidad de decidir si quieren asociarse con otros propietarios para obtener más beneficios. Cada una de las posibles combinaciones de colaboración entre las distintas CSs puede definirse como una coalición (C) [8]. Una coalición queda definida únicamente por  $(\mathcal{N}, \nu)$ . El parametro  $\mathcal{N}$  define los CSs que participan en la coalición, y  $\nu$  es el valor de la coalición C.

El valor de cada una de las posibles coaliciones (C) se puede dividir entre cada uno de los miembros de la coalición. La distribución del valor de la coalición  $\nu(C)$  se divide en función de reparto o asignación definidos previamente, cuando se crea la coalición, y es un vector definido como  $\phi_i$ , donde  $i \in C$ , es decir, el beneficio de la coalición C viene definido por  $\nu(C)$  y se distribuye entre cada uno de sus participantes en función de  $\phi_i$ .

En función de  $\mathcal{N}$ , existen un conjunto de posibles coaliciones  $\mathcal{C}$  definidas en (5.1).

$$\nu : 2^{\mathcal{N}} \rightarrow \mathbb{R} \quad (5.1)$$

Cuando existen un conjunto de las posibles coaliciones  $\mathcal{C} = \{C_1, \dots, C_i\}$  el valor una coalición  $C_i \in \mathcal{C}$  viene definida por  $(C_i, \mathcal{C})$  [3].

La función usada para repartir los beneficios de la coalición es la aproximación *minmax*. Esta función asigna a cada participante un valor en función de la mínima cantidad que puede recibir teniendo en cuenta a los demás jugadores. En otras palabras, es el valor que este participante está seguro de recibir conociendo las acciones de los demás participantes. La función *minmax* viene definida según (5.2).

$$\bar{v}_i = \min_{CS_i} \max_{CS_{-i}} v_i(CS_i, CS_{-i}) \quad (5.2)$$

Donde  $CS_i$  define la estaicon a la cual se debe asignar su valor de particion, y  $CS_{-i}$  son las demas estaciones.

Se puede definir el valor de *Shapley* como un método para distribuir el beneficio de una determinada coalición (o juego)  $(\mathcal{N}, v)$  [9]. El valor de *Shapley*  $\phi(v)$  cumple con los siguientes axiomas:

1) *Eficiencia*:

$$\sum_{i \in \mathcal{N}} \phi_i(v) = v(\mathcal{N})$$

2) *Simetría*: Si  $i$  y  $j$  son dos participantes equivalentes en el sentido de que :

$$v(\mathcal{C} \cup \{i\}) = v(\mathcal{C} \cup \{j\}), \forall \mathcal{C} | \mathcal{C} \cup \{i, j\} = \emptyset$$

para cada subconjunto  $\mathcal{C}$  de  $\mathcal{N}$  i ni j, entonces  $\phi_i(v) = \phi_j(v)$ .

3) *Jugador nulo*: Si para un jugador  $i$ , ocurre que:

$$v(\mathcal{C}) = v(\mathcal{C} \cup \{i\}), \forall \mathcal{C} | \mathcal{C} \cup \{i\} = \emptyset$$

entonces  $\phi_i(v) = \emptyset$ .

El axioma de simetría asigna la misma recompensa a los participantes que tiene una aportación similar a la coalición. Por otro lado el axioma *Jugador nulo* establece que un participante de la coalición que no aporta nada a esta, no recibirá recompensa. La formulación del reparto a cada participante de la coalición  $i$ , de acuerdo con el valor de *Shapley*, viene definido por (5.3).

$$\phi_i(v) = \sum_{\mathcal{C} \subseteq \mathcal{N} \setminus \{i\}} \frac{|\mathcal{C}|! (|\mathcal{N}| - |\mathcal{C}| - 1)!}{|\mathcal{N}|!} [v(\mathcal{C} \cup \{i\}) - v(\mathcal{C})] \quad (5.3)$$

La ecuación (5.3) presenta un coste computacional importante, sobre todo al aumentar el número de participantes en las coaliciones. Además durante el control, se debe realizar una optimización de la función objetivo para cada caso posible. Por una parte, existen  $2^{n_{CS}} - 1$  posibles coaliciones. Además, para cada una de las coaliciones posibles, se debe hacer una optimización para cada precio posible que puedan fijar las CSs que no se encuentren dentro de la coalición. Por todo ello, parece obvio que el coste computacional será muy alto, pero esto tiene fácil (en este caso) solución, elegir un tiempo de muestreo alto. En este caso el sistema tiene un tiempo de muestreo de 12 horas, que es más que suficiente para realizar todas las operaciones.

## 5.2 Función objetivo

El control predictivo en este proyecto, tomando como referencia [10], tiene como objetivo optimizar los precios de los componentes de cada una de las posibles coaliciones para maximizar su beneficio, teniendo en cuenta los posibles precios que puedan elegir los competidores fuera de su coalición. Los competidores, a priori, pueden optar por cualquier precio, en función de la valoración haga cada CS fuera de la coalición del estado actual del mercado. Por ello, para poder optimizar los precios de las coaliciones, se establecen unos precios mínimo y máximo para los competidores, es decir, se establece un rango de precio para los competidores. Dentro de este rango, se elige el número niveles de precios posibles de los competidores para “discretizar” el rango de precios elegido y así poder optimizar la función objetivo para cada una de las posibles combinaciones de precios de las CSs que no estén presentes en la coalición. Este proceso debe repetirse para cada una de las coaliciones posibles.

$$0,275 \geq p_i \geq 0,325 \quad (5.4)$$

El rango de precios de los competidores definidos en (5.4), se divide en dos niveles de precio. Se realiza la operación de optimización para cada combinación posible de precios.

$$nComb = nNiveles^{(N_p \cdot nCM_{noCoal})} \quad (5.5)$$

En (5.5) aparece la fórmula para el cálculo del número de combinaciones posibles de precios para el número de niveles de precio elegido  $nNiveles$  en el horizonte de predicción para las CSs que no participan en la coalición  $nCM_{noCoal}$  para cada coalición posible. La optimización debe realizarse para cada coalición posible, cada una de las cuales tendrá un número de

combinaciones posibles distinta. Esto, como ya se mencionó en la sección anterior, presenta un coste computacional importante, pero asequible para el tiempo de muestreo elegido.

Como se ha indicado anteriormente, el objetivo del control es maximizar el beneficio de cada CS. Este, viene definido por  $E_i p_i$ , es decir, el producto de la energía consumida en la estación  $i$  en  $kWh$  por el precio del  $kWh$ . La energía que se espera vender, puede ser obtenido a partir de  $p_i$  y usando el modelo ajustado anteriormente. De esta forma la función objetivo viene definida por la ecuación (5.6):

$$J_i(k) = \sum_{t=k}^{k+N_p} E_i(t) p_i(t) - c_i E_i^2(t) \quad (5.6)$$

siendo la energía es estimada según (5.7).

$$E(t+1) = E_b + B \cdot (p(t+1) - p_b) - k_{alt} \frac{\bar{p}(t+1) - p_{alt}}{p_{alt}} E_b \quad (5.7)$$

Donde  $E$  es la energía global que se va a consumir en todas las CSs estimada usando el modelo,  $c_i$  es una constante de mantenimiento de cada CS y  $k_{alt}$  es una constante que penaliza el cambio de precio durante el horizonte de predicción, siendo  $p_{alt}$  el precio alternativo que se toma como el precio base. Entonces solucionando el problema *minmax* de la ecuación (5.8) se puede obtener el beneficio estimado de cada CS dentro de cada posible coalición.

$$\max_{p_i} \min_{P-i} J_i \quad (5.8)$$

Como ya se ha comentado anteriormente, se debe resolver el problema *minmax* para cada una de las posibles combinaciones de precios en cada posible coalición. Tras ello, se elegirá la coalición que más beneficios aporte a sus miembros. También se debe estimar y demostrar los cambios y beneficios (si lo hubiera) que aporta una coalición (ya sea de todos las CSs o solo algunos) teniendo en cuenta los posibles costes de mantenimiento que puedan tener las CSs.





## 6 RESULTADOS

---

En este capítulo se van a exponer los resultados teniendo en cuenta las posibles variaciones en los costes de mantenimiento de cada CS. Estos costes, por si solos o sumados a las penalizaciones por cambios respecto al precio alternativo, pueden suponer la diferencia en cuanto a la decisión de formar una determinada coalición o continuar en solitario por parte de cada CS. Todos los resultados son testados primero usando como sistema el modelo, y posteriormente como resultados reales se toman los resultados simulando con SUMO.

En primer lugar se va a mostrar las diferencias, cuando se establece una Gran Coalición (*GC*, todas las CSs en la misma coalición), entre el beneficio que obtiene cada CS por separado y el beneficio que obtienen tras realizar el reparto en función del aporte de cada CS. Para este ejemplo se toma  $k_{alt} = 0,15$  y  $c = [120 \quad 90 \quad 105 \quad 100]$ .

La Figura 6-1 muestra los resultados de una *GC*. Se observa que para algunas es beneficioso el reparto debido a la coalición y para otras no depende básicamente de la energía que aportan a la coalición, el precio estimado para maximizar su beneficio, y sus costes de mantenimiento.

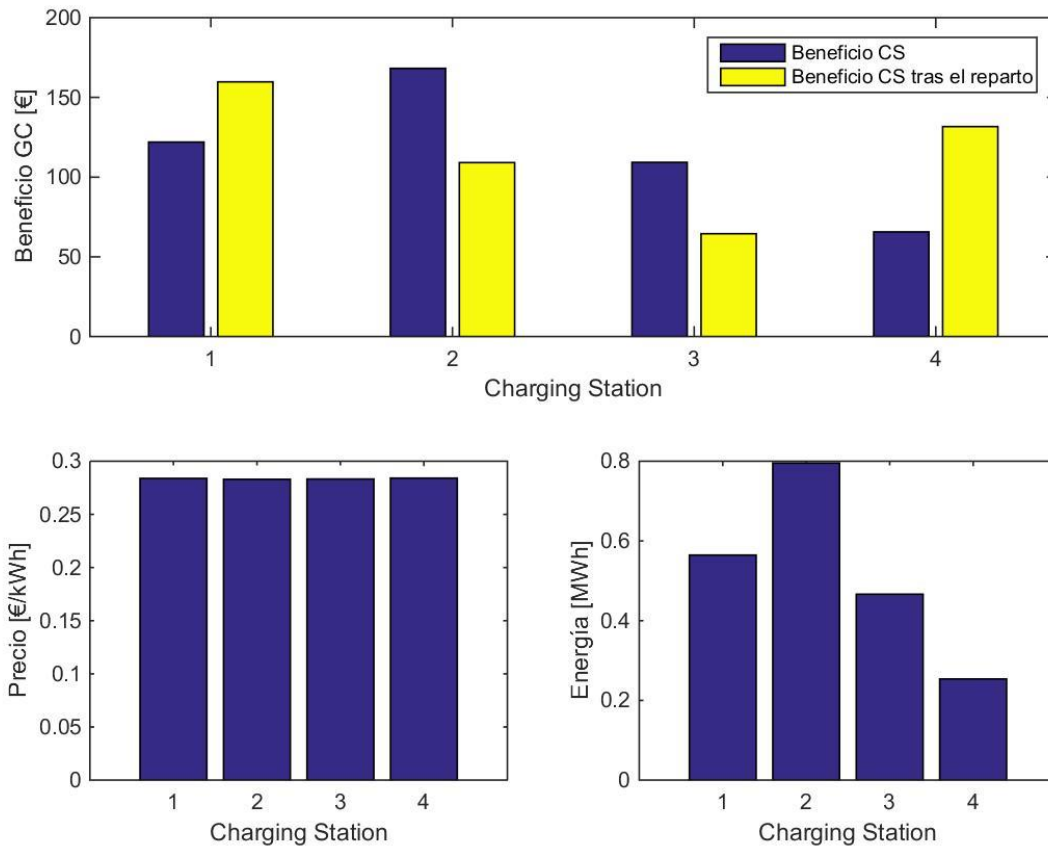


Figura 6-1 Diferencia entre el beneficio en una *GC* antes y después del reparto

En las siguientes subsecciones de este capítulo se van a exponer distintos escenarios variando los costes de mantenimiento y forzando ciertas coaliciones, para realizar una comparativa del funcionamiento del control.

## 6.1 Diferencia entre *Gran Coalición* y no coalición

Es distinto el beneficio que puede obtener una CS si forma se forman coaliciones, por ello la primera comparativa debe ser entre una *GC* forzada y el escenario donde no existen

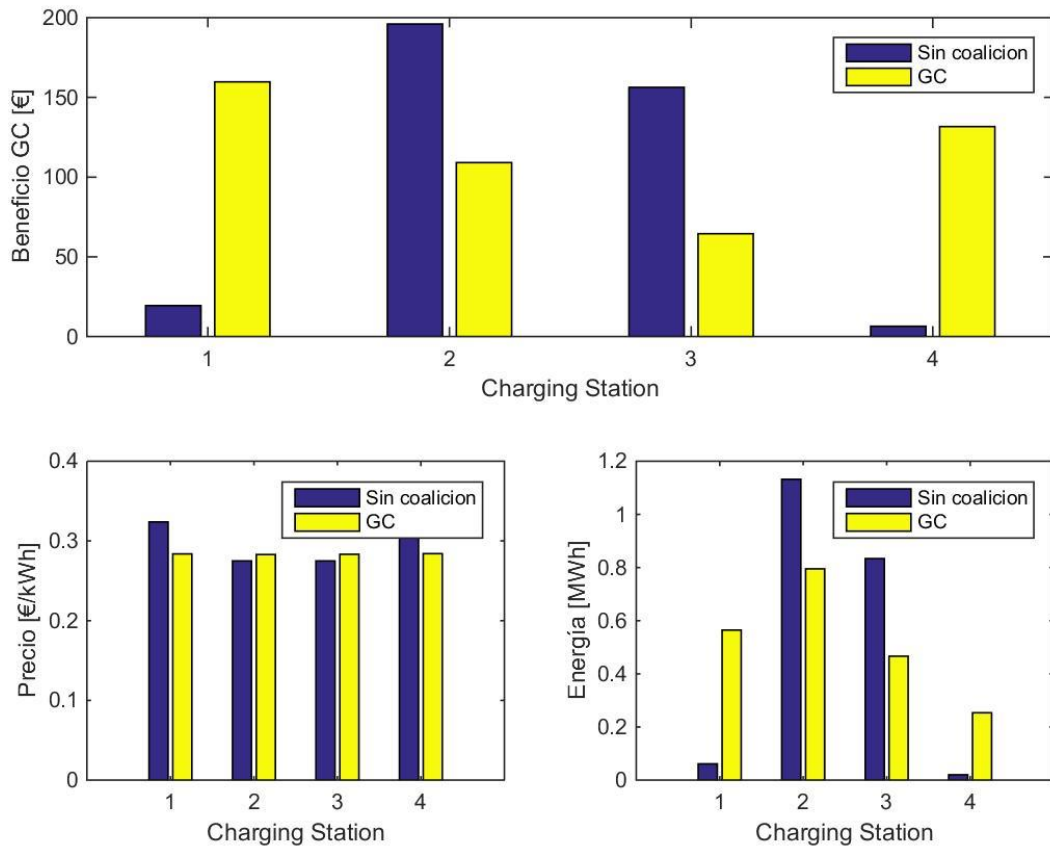


Figura 6-2 Comparativa entre *GC* y sin coaliciones

coaliciones. En FIG se presenta dicha comparativa, fijando los parámetros como  $k_{alt} = 0,15$  y  $c = [120 \ 90 \ 105 \ 100]$ . Se observa que el beneficio tras el reparto en una *GC* es enorme, en especial para algunas CSs, mientras que cuando no existe ninguna coalición hay CSs que tiene beneficios muy bajos en comparación a otras CSs (véase la diferencia en el caso sin coaliciones entre CS1 y CS2). Por lo tanto, mientras que para algunas CSs su pone un gran aumento en los beneficios, para otras puede suponer una disminución tan considerable. Por lo

tanto, como ya se ha explicado, es necesario utilizar la función de coste para estimar cuál de las posibles situaciones es más beneficiosa para cada CSs.

## 6.2 Coaliciones y efecto de los costes de mantenimiento

A continuación, para cada posible situación que pueda darse (posibles coaliciones y precios de las demás CSs) se debe resolver el problema *minmax* para la función objetivo planteada en el capítulo anterior. De esta forma se obtendrán predicciones de los beneficios obtenidos para cada posible caso y de esta forma se podrá escoger los precios y coaliciones deseadas para obtener el mayor benéfico en las CSs.

De igual manera que en la subsección anterior, se va a realizar la primera prueba con los parámetros fijados a  $k_{alt} = 0,25$  y  $c = [120 \ 90 \ 105 \ 100]$ . En este caso se modifica  $k_{alt}$  ya que se obtienen resultados más coherentes que con el valor fijado anteriormente.

En la figura Figura 6-3 muestra los resultados de aplicar el control predictivo para maximizar los beneficios y elegir la mejor coalición. La primera coalición formada es entre CS1 y CS4. Estas dos CSs forman una coalición con la que, estableciendo precios bajos, para la penalización ante el cambio respecto al precio alternativo, consiguen las máximas ventas de energía y los máximos beneficios. Por otro lado, CS2 y CS3 forman la otra coalición, la cual ante un volumen menor de energía vendida, establecen unos precios para intentar aumentar sus beneficios.

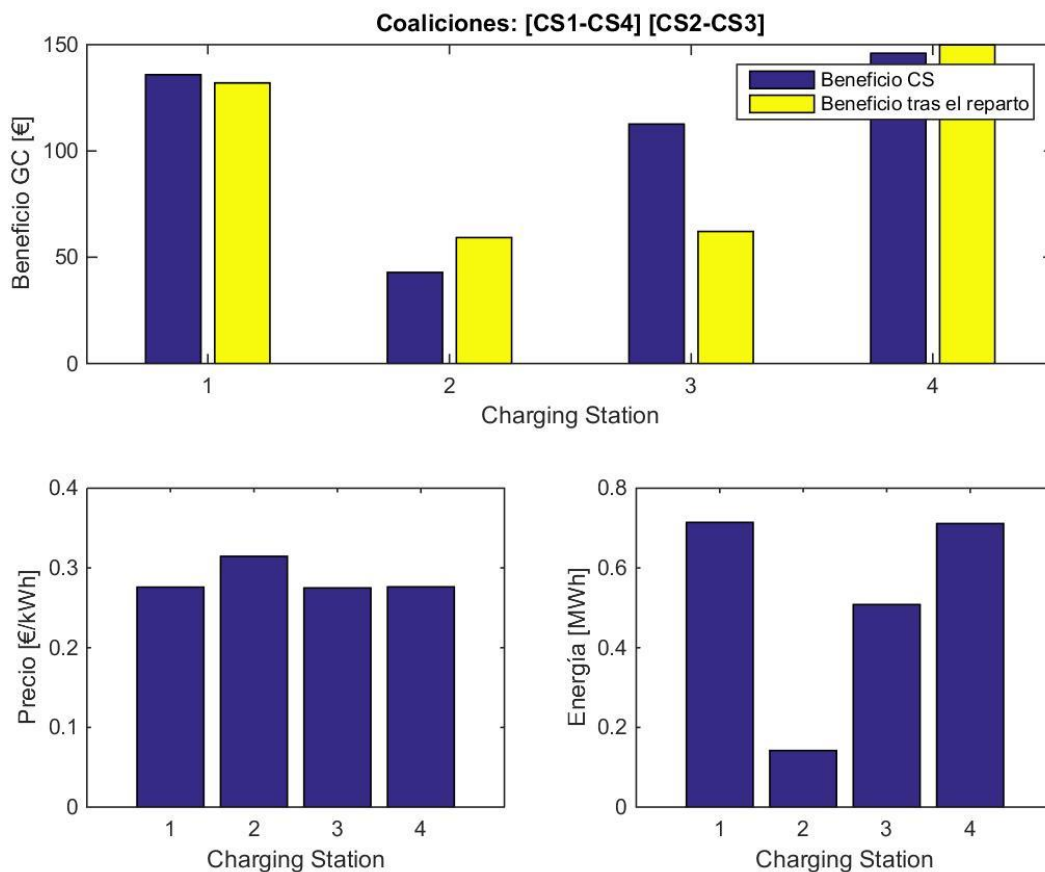


Figura 6-3 Resultados de las coaliciones CS1-CS4 y CS2-CS3

En este punto se van a mostrar los efectos de los costes de mantenimiento de las CS, los cuales van a influir en las coaliciones que se forman en el escenario, en la energía vendida por cada CSs y en los precios.

Como puede verse en la Figura 6-4, se va a realizar una comparativa entre 3 posibles valores que pueda tomar el vector  $c$ . La distribución de coaliciones 1 (en la figura *Coal 1*) corresponde al vector  $c = [120 \ 90 \ 105 \ 100]$ , es decir, el usado hasta el momento. La distribución de coaliciones 2 (en la figura *Coal 2*) corresponde al vector  $c = [120 \ 90 \ 105 \ 125]$ , es decir, solo vaían los costes de CS4 respecto a *Coal 1*. La distribución de coaliciones 3 (en la

figura *Coal 3*) corresponde al vector  $c = [140 \ 90 \ 105 \ 100]$ , es decir, solo vaían los costes de CS2 respecto a *Coal 1*.

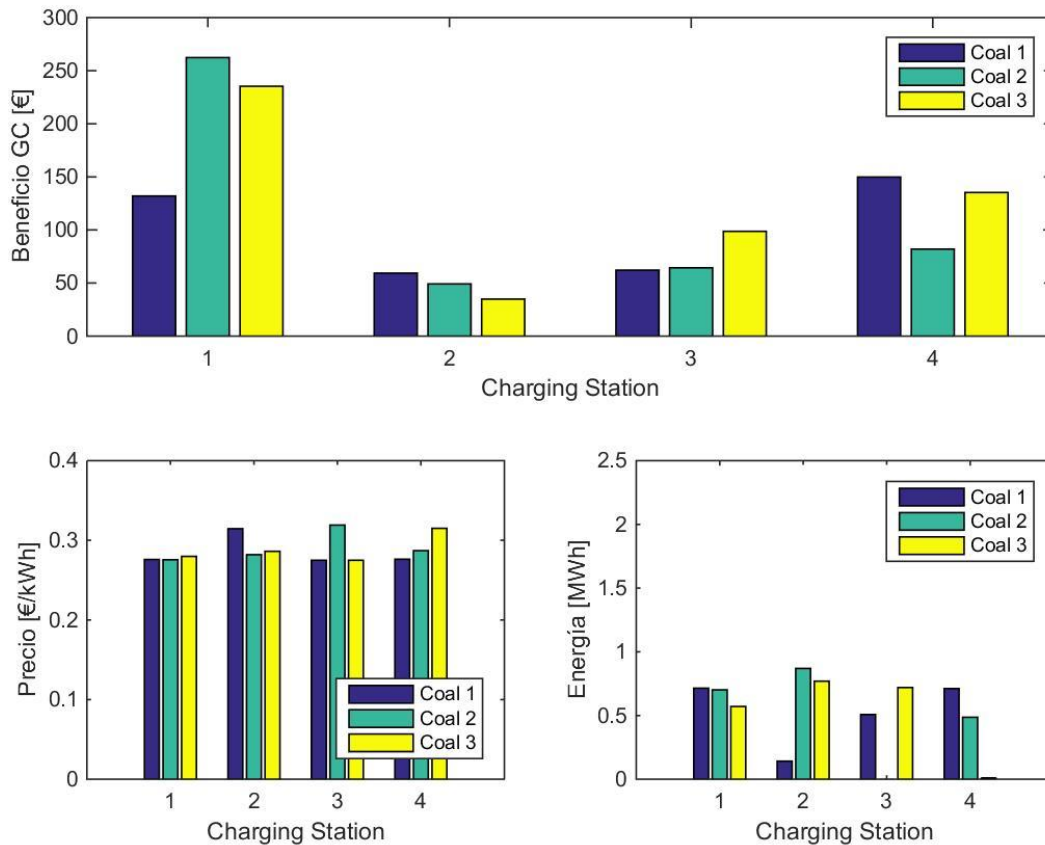


Figura 6-4 Comparativa costes de mantenimiento

En *Coal 1* se tienen las siguientes coaliciones:

- CS1 con CS4
- CS2 con CS3

En este caso no se entra a explicar esta configuración ya que es la explicada con la Figura 6-3.

En *Coal 2* se tienen las siguientes coaliciones:

- CS1 con CS2: CS1 pasa a formar coalición con CS2 ya que CS4 tiene más costes de mantenimiento y no es la coalición más adecuada en este caso. Disminuyendo el precio en CS2, se consigue que, manteniendo el mismo casi el mismo precio y energía vendida en CS1, aumente considerablemente la energía vendida en CS2. CS2 mantendrá prácticamente el mismo beneficio respecto al caso 1, pero CS1 formando coalición con CS2 y teniendo en cuenta los beneficios perdidos por CS4 y la energía no vendida por CS3, consigue que su beneficio aumente considerablemente.
- CS3 con CS4: CS3 provoca que no venda energía aumentando su precio. Esto sumado a su posición en el mapa con respecto a CS4 (muy cerca) provoca que los EVs presentes en esa zona vayan a recargar a CS4.

En *Coal 3* se tienen las siguientes coaliciones:

- CS1 con CS3: Respecto al caso 1, CS1 baja su producción de energía, lo cual, teniendo en cuenta que han subido sus costes de mantenimiento, hace que no afecte mucho la penalización que se le impone por energía vendida. Además, tiene menor penalización del término de cambio de precio respecto al precio alternativo ya que el precio en este caso está más cerca al precio alternativo que en el caso 1. Por otro lado, CS3 aumenta considerablemente su energía vendida. Teniendo en cuenta que tiene los precios más bajos del escenario, aunque CS1 haya aumentado sus costes de mantenimiento respecto al caso 1, ambas estaciones aumentan considerablemente su beneficio tras el reparto.
- CS2 con CS4: Como CS4 establece un precio muy superior a CS3, pierde casi todas

sus ventas. CS2 por su parte aumenta considerablemente sus ventas de energía (gracias a que CS1 las disminuye).



# 7 CONCLUSIONES

---

Tras realizar todas estas pruebas usando el simulador de tráfico SUMO junto a Matlab, se llega a varias conclusiones:

- En este momento es imposible realizar este trabajo con experimentos reales, ya que no existen las infraestructuras, la cantidad de EVs ni las comunicaciones necesarias para ella. En este momento hay pocos EVs respecto al número de vehículos que son necesarios para realizar las pruebas realizadas en SUMO. Además no existe el tipo de telecomunicaciones entre EVs y CSs que necesitaría el control para acceder a todos los datos que usa.
- Gracias a AYESA y a los datos del proyecto *Zem2All*, fue posible la creación de un simulador fiable, a partir del cual se puede extraer un modelo de comportamiento del sistema. Este modelo, obtenido para el punto de trabajo considerado el correcto, permite que se pueda crear, probar y aplicar un control predictivo (MPC) al sistema, sin la necesidad de hacer todas estas pruebas sobre el simulador SUMO (es demasiado lento para realizar la cantidad de pruebas que han sido necesarias para llevar a cabo este trabajo).

Este modelo puede haber sido estimado usando varios métodos y tener varias estructuras posibles. El modelo que se ha usado es un modelo estimado por mínimos cuadrados en el que cada entrada es sumada a las demás aplicándoles una ganancia en función de la salida. Es un modelo muy simple, y se intuye que es posible obtener

modelos más complejos que modelen mejor el comportamiento del sistema. Por ello se queda como trabajo pendiente el estudio y la búsqueda de modelos alternativos que aporten mejores resultados que el actual, y que además sea posible modificar teniendo en cuenta el tipo de EV, tiempos de recarga, cantidad de EVs y CSs,...

- La importancia de algunos términos en la función de coste es muy alta, ya que permiten que el control tenga en cuenta aspectos que son de vital importancia para que en el sistema real (en un futuro) funcione de forma correcta. Por ello, han sido necesarias multitud de pruebas de ensayo-error para ajustar estas constantes (costes de mantenimiento, sensibilidad al cambio de precio,...). Aun así, aún quedan muchos factores que no se han tenido en cuenta, y que pueden tener efecto en las decisiones de los conductores, el funcionamiento de los EVs y de las CSs,... Todo ello en un futuro, si se decidiera llevar a la práctica este trabajo, se debería tener en cuenta y realizar un modelado mucho más exhaustivo.
- En general, las coaliciones son útiles para las CSs, pero depende en gran medida de los costes de mantenimiento de las estaciones, y de su posición en el mapa, es decir, de los beneficios que puedan obtener si todas las CSs imponen el mismo precio y no existen costes de mantenimiento (véase la energía de la simulación base,  $E_b$ ). Para unas CSs un aumento similar en los costes de mantenimiento supone una pérdida mucho mayor que para otras (algunas incluso mejoran al considerar que deben realizar una coalición distinta).

- Se observa en los resultados que el control predictivo tiene en cuenta las pérdidas a la sensibilidad al cambio respecto al precio alternativo y a los costes de mantenimiento, y que actúa en consideración, subiendo o bajando los precios, con el objetivo de maximizar los beneficios de la coalición elegida. Hay que recordar, que en realidad este control, tiene en cuenta todas las posibles coaliciones, para los posibles precios que puedan establecer los competidores, y elige los precios teniendo en cuenta el caso que más beneficios devuelve a la coalición (o a la CS si decidiera actuar sola).



## [8] REFERENCIAS

---

- [1] D. C. Tejero, «Simulador de vehículos eléctricos en MATLAB,» de *XXXVI Jornadas de Automática*, Bilbao, 2015.
- [2] R. d. Río y H. Tamura, «ZEM 2 All Project (Zero Emission Mobility to All),» de *International Conference on Intelligent Transportation Systems (ITSC)*, Las Palmas, Spain, 2015.
- [3] E. E. M. W. Georgios Chalkiadakis, *Computational Aspects of Cooperative Game Theory*, USA: Morgan & Claypool, 2012.
- [4] «Simulation of Urban MObility,» [http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931\\_read-41000/](http://www.dlr.de/ts/en/desktopdefault.aspx/tabid-9883/16931_read-41000/).
- [5] A. Acosta, «TraCI4Matlab,» [En línea]. Available: <https://es.mathworks.com/matlabcentral/fileexchange/44805-traci4matlab>.
- [6] «Armax,» <https://es.mathworks.com/help/ident/ref/armax.html>.
- [7] «Neural Net Fitting,» <https://es.mathworks.com/help/nnet/index.html>.

- [8] F. Fele, J. M. Maestre y E. F. Camacho, «Coalitional Control: Cooperative Game Theory and Control,» de *IEEE CSM*, 2017.
- [9] L. Shapley, «The Shapley value,» *Annals of Math*, vol. 28, pp. 307-317, 1953.
- [10] E. F. Camacho y C. Bordons, «Model Predictive Control,» de *Springer Verlag*, 2014.