Trabajo Fin de Grado
Grado en Ingeniería de Tecnologías Industriales

# Algoritmos para la Reconstrucción de Datos de Demanda

Autor: Agustín Robles Olmo
Tutor: Teodoro Álamo Cantarero

**Dep. Ingeniería de Sistemas y Automática**
**Escuela Técnica Superior de Ingeniería**
**Universidad de Sevilla**

Sevilla, 2018

Trabajo Fin de Grado
Grado en Ingeniería en Tecnologías Industriales

# Algoritmos para la Reconstrucción de Datos de Demanda

Autor:
Agustín Robles Olmo

Tutor:
Teodoro Álamo Cantarero
Catedrático de la Universidad de Sevilla

Dep. de Ingeniería de Sistemas y Automática
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla
Sevilla, 2018

Trabajo Fin de Grado: Algoritmos para la Reconstrucción de Datos de Demanda

Autor: Agustín Robles Olmo

Tutor: Teodoro Álamo Cantarero

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

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2018

El Secretario del Tribunal

*A todos los que me han apoyado durante este tiempo.*

# Agradecimientos

En primer lugar, dar las gracias a mis padres por creer en mí desde que era niño, por apoyarme, por tener paciencia conmigo durante el grado y buscar siempre lo mejor para mí, por permitirme hacer lo que he creído conveniente en cada momento, ya que algunas de esas decisiones me han hecho crecer como persona, y porque sin ellos no hubiera sido posible llegar hasta aquí.

Agradecer a Teodoro Álamo, tutor de este proyecto, ya que gracias a sus clases en el grado me entró interés en profundizar en Machine Learning. Además, siempre ha estado disponible cuando he necesitado su ayuda y su orientación ha sido vital para la consecución de este proyecto.

A mis hermanos y amigos, por ser una válvula de escape, por hacerme disfrutar en esos viajes que hacemos cada verano y por estar ahí siempre que los he necesitado.

Y en especial a Alejandra, por aguantarme en mis malos momentos y hacer especiales los buenos.

*Agustín Robles Olmo*

*Sevilla, 2018*

# Resumen

Actualmente, las empresas de abastecimiento están implementando "contadores inteligentes" con el objetivo de obtener información de mayor calidad y poder adaptar mejor la distribución de agua, además de para proporcionar al consumidor una mejor compresión del gasto que genera.

Existen situaciones en las que los datos medidos no son recibidos, por ejemplo, debido a problemas en las comunicaciones o fallo en el contador, por lo que el objetivo de este proyecto es, disponiendo de los datos de volumen obtenidos por los contadores, obtener una aproximación del consumo de las medidas que faltan y se pueda utilizar dicha aproximación para poder detectar fugas, deterioro de infraestructuras y fraudes, la identificación de las condiciones de riesgo, alertas automáticas, etc., lo que redundará en una mejora de los procesos de mantenimiento y en un adelanto en la resolución de incidencias. Así mismo, también podrá ofrecerse al ciudadano una información completa y personalizada sobre sus hábitos de consumo y potenciar el consumo responsable.

Además de lo mencionado anteriormente, es necesario tener una alta fiabilidad de los datos obtenidos para pasar al siguiente paso, la predicción del consumo de los usuarios.

# Abstract

Nowadays, supply companies are implementing “Smart meters” in order to obtain higher quality information and adapt the water supply in a better way, as well as to provide the consumer with a better understanding of the water used.

There are situations in which the measured data aren’t received, for example due to communication problems or failure in the meter, so the objective of this project is, from the collected data by the meters, to obtain an approximation of the measures that are missing. The approach can be used to detect leaks, deterioration of infrastructures and fraud, the identification of risk conditions, automatic alerts, etc., which will result in an improvement of the maintenance processes and in an advance in the resolution of incidents. Likewise, the citizen can also be offered complete and personalized information about their consumption habits and promote responsible consumption.

In addition to the aforementioned, it is necessary to have a high reliability of the data obtained to move to the next step, the prediction of user consumption.

# Índice

# ÍNDICE DE TABLAS

# ÍNDICE DE FIGURAS

# 1 INTRODUCCIÓN

En 1972, Theodore George Paraskevakos, mientras trabajaba para Boeing en Huntsville, Alabama, desarrolló un sistema de monitorización por sensores que utilizaba la transmisión digital para mejorar la seguridad y con capacidades de lectura de contadores para todos los servicios públicos. En 1974, el Sr. Paraskevakos obtuvo una patente en EE.UU. para esta tecnología y en 1977 fundó la compañía Metretek Inc., que desarrolló y produjo el primer sistema de gestión de carga y de lectura de contadores totalmente automatizado y comercialmente disponible.

En España, desde 1992 se fabrican los que se denominan Contadores inteligentes de agua, una de las grandes ventajas de estos equipos es su capacidad de comunicación, que permite acceder a toda la información estadística de hábitos de consumo a través de cualquier soporte (Radio, GSM/GPRS, Cable...). En agosto de 2011 había más de 1.500.000 equipos operativos en España y se prevé que en el futuro todos los equipos operativos sean de este tipo.

La tecnología de medición sigue siendo la misma que en los contadores de lectura manual y por lo tanto, los problemas asociados (como los errores de medición, desgaste de piezas móviles, ritmo de deterioro del error del contador, etc.) son los mismos. La diferencia es la transmisión de datos, para proporcionar la información en tiempo real y la discriminación horaria se define una nueva forma de medir denominada "Smart Metering" o medición inteligente, se cuantifica y transmite instantáneamente la información necesaria mediante un emisor de pulsos y tele lectura (recogida de datos a distancia).

En el caso de las empresas de abastecimiento, por ejemplo Emasesa, esta tecnología permite adquirir de forma automática, y con varias lecturas al día, los consumos de particulares y empresas. Además, permite una autonomía de los contadores próxima a los diez años por lo que se reduce significativamente las necesidades de mantenimiento. La tele medida permite una más eficaz y temprana detección de fugas, deterioro de infraestructuras y fraudes, la identificación de condiciones de riesgo, alertas automáticas, etc., lo que redundará en una mejora de los procesos de mantenimiento y en un adelanto en la resolución de incidencias. Así mismo, también permite ofrecer al ciudadano una información completa y personalizada sobre sus hábitos de consumo y potenciar el consumo responsable.

Hay que tener en cuenta que durante el proceso de transmisión de datos existe una cantidad de los mismos que no llegan a su destino, ya sea por fallo en el envío, por problemas en la recepción o simplemente porque el contador ha dejado de funcionar (por ejemplo se le ha acabado la batería). En este proyecto se van a aplicar distintos algoritmos para la reconstrucción de dichos datos.

Se ha realizado en la plataforma Matlab debido a que dicho software está orientado al uso de matrices y es de fácil acceso ya que, aunque no sea de libre acceso, en la Escuela Técnica Superior de Ingeniería de Sevilla hay muchos ordenadores desde los que se puede acceder.

Se han elegido tres métodos de aproximación aumentando el nivel de complejidad progresivamente. El primero de ellos se trata de aproximar el consumo que falta cogiendo datos previos, el segundo es a través de Mínimos Cuadrados, se aproximará usando distintos regresores (teniendo en cuenta sólo datos del contrato a aproximar y otro teniendo en cuenta también un agregado de contratos) y el tercer algoritmo elegido se trata de Procesos Gaussianos, este último es el algoritmo más potente y

complejo por lo tanto se espera obtener unos mejores resultados.

La memoria está dividida en los siguientes capítulos restantes:

- Capítulo 2: Tratamiento Previo.

  Este capítulo se divide en 3 subapartados, en el 2.1 se lleva a cabo una introducción a los datos utilizados y se explica cómo se han tenido que modificar para disponer de ellos de una forma rápida y sencilla. En el 2.2 se explican las funciones que realizan dicha modificación y cómo la llevan a cabo. Por último, en el subapartado 2.3, aparece cómo se ha calculado el consumo y una representación del mismo para un contrato aleatorio.

- Capítulo 3: Primera Aproximación.

  El capítulo 3 se divide en 2 subapartados, en el primero de ellos (3.1) se explica el primer algoritmo elegido, el método seleccionado para realizar la aproximación y para calcular cuánto error se comete a lo largo del proyecto, el tipo de error utilizado, ya que como hay casos en los que el consumo es 0, si se usa por ejemplo el error relativo, con poco error absoluto que se cometa al dividir entre 0 el error relativo se va a infinito, por lo que hay que usar otro tipo de indicador. También se explica la metodología a seguir en todos los algoritmos para intentar obtener unos resultados representativos, de forma que, al usar siempre la misma metodología, se pueden comparar los resultados y se puede calcular si existe una mejora cuando se realiza la aproximación con distintos algoritmos.

  En el subapartado 3.2 se lleva a cabo una modificación del algoritmo anterior y se calcula si se produce mejoría. Esta modificación se corresponde con la introducción de un factor de corrección ($\alpha$) que tiene en cuenta el consumo total del intervalo a aproximar.

- Capítulo 4: Mínimos Cuadrados.

  Consta de 3 subapartados. El 4.1 se corresponde con la teoría necesaria para realizar el algoritmo, en el 4.2 se sigue la metodología de aproximación usando el primer regresor, se ha elegido usar un regresor que tenga en cuenta los datos del mes anterior del contrato que se está aproximando. Una vez obtenido el error cometido, se mejora dicho regresor introduciendo el mismo coeficiente "$\alpha$" que en el capítulo 3. Por último, en el subapartado 4.3 se le añade complejidad al regresor utilizado, en este apartado se tienen en cuenta tanto los datos del contrato actual como un agregado de otros contratos, una vez obtenida la aproximación se intenta mejorarla introduciendo el coeficiente "$\alpha$" y utilizando un factor de olvido.

- Capítulo 5: Procesos Gaussianos.

  Está dividido en 5 subapartados. El 5.1 es una introducción, se exponen los conceptos básicos que se van a desarrollar en los siguientes puntos. En el 5.2, se expone la teoría necesaria para poder desarrollar el algoritmo de los Procesos Gaussianos, se centra en la distribución normal y alguna de sus propiedades y el 5.3 se centra en cómo se maneja la incertidumbre en altas dimensiones.

  En el 5.4, se aproxima el consumo de tres formas distintas, la primera es: eligiendo la media como la media aritmética de todos los elementos dentro del set de puntos elegido. En la

segunda forma, se realiza una mejora y se consigue mejorar los resultados obtenidos, aquí se decide usar 4 medias distintas, es decir, como se toman cuatro medidas a lo largo del día se realizan 4 medias, cada una de ellas para cada tramo horario. Para realizar la tercera mejora se parte de la segunda, en este caso se tiene en cuenta el consumo total del intervalo en el cual faltan los puntos, con esto se consigue una gran mejoría y se obtienen unos resultados que son los mejores de todo el proyecto.

Por último, en el subapartado 5.5 se muestra (como conclusión del proyecto) una tabla comparativa de los errores cometidos para todos los algoritmos usados.

- Capítulo 6: Código Matlab.

Para evitar que este apartado fuera demasiado extenso, se ha decidido incluir el código que es realmente necesario para correcta ejecución del proyecto. Muchas pruebas realizadas se han dejado fuera, al igual que el código de comprobación de errores. De los tres algoritmos utilizados en el proyecto, sólo para los procesos gaussianos aparece el código integro ya que el "esqueleto" de los tres es muy parecido y se ha omitido tanto para la aproximación inicial como para los mínimos cuadrados.

# 2 TRATAMIENTO PREVIO

Antes de empezar con la implementación de los algoritmos para la reconstrucción de datos, es necesario familiarizarse con los mismos y desarrollar ciertos programas para tratarlos y dejarlos de una forma que sea fácil usarlos. Para ello, se llevará a cabo un tratamiento previo al conjunto de los datos y se guardarán en varias variables cada una con unas características que facilite el tratamiento de la información consignada en ellas.

## 2.1 Datos

### 2.1.1 Variables

Al cargar los datos proporcionados en Matlab aparecen dos variables:

- matTelem: Matriz de dimensiones 743980x8.
- tabIDinfo: Estructura que contiene información sobre matTelem.

La variable matTelem está compuesta por:

Tabla 2-1. Contenido matTelem.

| Nº de columna | Contenido | Definición |
|---|---|---|
| 1ª | ID | Número del contrato. |
| 2ª | T_lectura | Fecha y hora del momento de lectura del contador. |
| 3ª | T_adquisición | Fecha y hora del momento en que se recibe el dato. |
| 4ª | Volumen | Volumen medido. |
| 5ª | Sector | Identificador del Sector. |
| 6ª | CNAE | Clasificación Nacional de Actividades Económicas. |
| 7ª | Viviendas | Número de viviendas que se tienen en cuenta en el contrato. |
| 8ª | Habitantes | Número de personas que viven en la vivienda según padrón municipal. |

Los campos necesarios para este proyecto son: la primera, la segunda y la cuarta columna, los demás no se usarán debido a que en muchos casos faltan o no están completos. En caso de que estuvieran completos sería una buena idea tenerlos en cuenta ya que se podría buscar cierta semejanza entre los contratos, puede ser un área de trabajo futuro.

El campo ID toma valores enteros que van desde 1 hasta 1431 por lo que se dispone de información

de 1431 contratos, el T_lectura es un decimal, para transformarlo a formato horario se ha optado por usar el comando "datevec(decimal)", este comando devuelve un vector fila con la fecha y hora de la medida de la siguiente forma:

Año Mes Día Hora

Cada día se toman cuatro medidas, a las 3-5-9-14 horas, las cuales (campo Volumen) vienen dadas en litros.

La variable tabIDinfo tiene los siguientes campos:

Tabla 2-2. Contenido tabIDinfo.

| Campo | Definición |
|---|---|
| ID | Número del contrato. |
| Tipo | Tipo de medida, en este caso volumen. |
| Unidad | Unidad del dato adquirido, en este caso litros. |
| Contador | ID del contador. |
| Contrato | ID del contrato de cada cliente. |
| Sector | Identificador del Sector. |
| Uso | Qué uso tiene el contador, si es comercial, doméstico, etc. |
| CNAE | Clasificación Nacional de Actividades Económicas. |
| Viviendas | Número de viviendas que se tienen en cuenta en el contrato. |
| Habitantes | Número de personas que viven en la vivienda según padrón municipal. |
| Calibre | Diámetro de la tubería. |
| DateAplication | Fecha en la que la universidad hizo los datos. |

No se va a usar por lo que no se va a entrar en detalles, aunque muchos de los campos son similares a los de matTelem, al igual que en el caso anterior, no se utiliza porque muchos de los contratos no tienen completos todos los campos, por lo que no se pueden manejar con total fiabilidad.

### 2.1.2 Contratos

Cada contrato es independiente al anterior, tienen fecha inicial, final y número de medidas distintos por lo que hay que buscar contratos que se adecuen a ciertos requisitos. Hay que tener en cuenta que los contratos no vienen con todos los datos, es decir, hay medidas duplicadas y otras que faltan, por ello, lo primero que se ha realizado ha sido una función que quite las medidas duplicadas y otra que inserte una fila de NaN cuando falte un dato, es decir, cuando se detecte que falta una medida de volumen entre la actual y la siguiente, se inserta una fila entre ambas y se rellenan con NaN todos los campos de la matriz (recordar que la matriz que se está utilizando es una matriz 743980x8). Se ha elegido insertar una fila de NaN porque es más fácil de localizar, hay comandos que son muy útiles como "isnan(var)" que devuelve 1 si var = NaN o 0 si no lo es. Una vez introducidas las filas de NaN, hay que ponerles la fecha y el ID del contrato al que pertenecen para lo cual se ha realizado otra función. Por último, se ha escrito otra función que se encarga de almacenar en una variable el

número de medidas existente en cada contrato.

Estas funciones para el tratamiento previo son:

- "Quitarpuntosrepetidos.m"
- "Ponernan.m"
- "Fechayhora.m"
- "Indices.m"

Hay que destacar que para el correcto funcionamiento de "Quitarpuntosrepetidos.m" hay que ejecutar previamente la función "indices.m". Otro problema que existe es que los contratos tengan demasiadas medidas con consumo 0, es decir, que el volumen no cambie a lo largo del tiempo. Para evitarlo, se ha realizado un filtro que permite que haya un cierto porcentaje de consumo 0 y si se supera, el contrato no será válido. El nombre del programa es: "Quitarcontratosconsumo0.m". Los contratos válidos se guardan en un archivo ".mat" cuyo nombre es:

"datosconfecha%" en % se sustituye el porcentaje permitidos de consumo cero, por ejemplo, si el nombre es "datosconfecha70" significa que se ha permitido que el 70% de las medidas tenga consumo 0. Para la ejecución de los algoritmos descritos más adelante se ha permitido que hubiera un porcentaje de consumo=0 del 50% ya que un consumo 0 mayor ocasiona problemas a la hora de realizar ciertas aproximaciones.

## 2.2 Funciones y programas.

En este apartado se va a explicar brevemente las funciones anteriores estando el código en su totalidad en el capítulo 6.

### 2.2.1 Índices

Se va recorriendo desde el primer dato hasta el último, mientras que se esté en el mismo contrato se incrementa un contador, en el momento en que se cambia de contrato, el valor del contador se pasa a la primera componente del vector "nmu" (número de muestras de cada usuario), se reinicia contador y la próxima vez que se cambie de contrato se pasa el valor del contador a la siguiente componente de "nmu". Cuando se llegue al final, se tendrán en las componentes de "nmu" el número de medidas que tiene cada contrato.

### 2.2.2 Quitar puntos repetidos

Primero determina el punto inicial y el final del contrato que se haya elegido (gracias a que se dispone del número de puntos que tiene cada contrato obtenido con la función descrita anteriormente), a continuación, se recorre dicho contrato y se resta la segunda columna de la matriz (que contiene la fecha) "matTelem(i, 2) - matTelem(i-1, 2)" de forma que si es 0 se trata de dos medidas que se obtuvieron en la misma fecha y hora por lo que está duplicada, en este caso se elimina ("matTelem(i, :)") de los datos del contrato.

### 2.2.3 Poner NaN

Una vez se tienen los datos sin puntos repetidos hay que buscar los que faltan y añadir las filas de NaN correspondientes. Lo primero es usar el comando "datevec" con la segunda columna de los datos para obtener la fecha en que han sido obtenidos. Como se toman cuatro medidas al día y siempre son a la misma hora, si se resta la hora de dos medidas consecutivas tiene que dar unos valores determinados, en caso de no ser así significa que falta alguna medida, aunque esto no es siempre así ya que hay que tener en cuenta que, si entre una medida y la siguiente se cambia de día, aunque el valor de la resta sea el adecuado, faltan datos. En la siguiente tabla quedan reflejadas todas las posibilidades:

Tabla 2-3. Posibilidades diferencia horario.

| Hora inicial | Hora final | Diferencia | Datos que faltan[1] |
|---|---|---|---|
| 3 | 3 | 0 | 4*n1+3 |
| | 5 | 2* | 4*n |
| | 9 | 6 | 4*n+1 |
| | 14 | 11 | 4*n+2 |
| 5 | 3 | -2 | 4*n1+2 |
| | 5 | 0 | 4*n1+3 |
| | 9 | 4* | 4*n |
| | 14 | 9 | 4*n+1 |
| 9 | 3 | -6 | 4*n1+1 |
| | 5 | -4 | 4*n1+2 |
| | 9 | 0 | 4*n1+3 |
| | 14 | 5* | 4*n |
| 14 | 3 | -11 | 4*n1 |
| | 5 | -9 | 4*n1+1 |
| | 9 | -5 | 4*n1+2 |
| | 14 | 0 | 4*n1+3 |

*Si son medidas del mismo día son correctos, no hay que añadir ninguna fila de Nan.

Otra cosa a tener en cuenta es, en caso de que se produzca cambio de mes, se hace de forma similar al cambio de día.

La función va recorriendo el contrato y va mirando la diferencia entre la hora a la que se obtiene el

[1] n= dias(i+1,3)-dias(i,3); es decir, la diferencia entre los días en los que se ha tomado la medida "i" y la siguiente.

La variable n1 = n-1. Se cogen distintas N porque tienen en cuenta el hecho de que en algunos casos la diferencia es negativa y en otros positiva, cuando la diferencia es negativa se usa n1 ya que, aunque haya cambio de día pueden faltar menos de 4 medidas, por eso n1 empieza en 0, mientras que, si la diferencia es positiva, cuando haya cambio de día siempre faltarán 4 o más medidas.

dato actual y el siguiente, en caso de que sea uno de los valores correctos no añade nada, pero si cualquier otro de los valores tabulados, dependiendo de si se cambia de día, mes, o no, se añade un número determinado de filas de NaN. En caso de que el valor obtenido en la resta no sea correcto y no esté tabulado hay un error en la obtención de los datos.

### 2.2.4 Fecha y hora

Esta función se encarga de poner el ID y la fecha a las filas de NaN introducidas anteriormente, para ello se ha observado que al calcular la diferencia entre la componente "i+1" y la "i" de la segunda columna de los datos (sin estar transformada a fecha) se van repitiendo los valores, de forma que cuando haya un NaN sólo hay que mirar a qué hora se realizó la medida anterior y sumar una determinada cantidad que viene tabulada en la tabla 2-4.

Tabla 2-4. Incremento Horario.

| Hora Inicial | Hora final | Diferencia |
|---|---|---|
| 3 | 5 | 0.083333333372138 |
| 5 | 9 | 0.166666666627862 |
| 9 | 14 | 0.208333333372138 |
| 14 | 3 | 0.541666666627862 |

Es decir, para poner la fecha en una fila que sea NaN, se mira la medida anterior, si por ejemplo la hora de medición fue las 3, le sumamos 0.083333333372138 y así con cada NaN.

La función lo que hace es transformar la segunda columna de los datos en fecha. A continuación, se recorre el contrato y cuando encuentra un NaN mira la hora de la medida anterior, dependiendo de cuál sea, le suma una cantidad u otra. Además, se pone el ID en la primera columna de los datos. En caso de que por alguna circunstancia muy particular no se haya podido poner la fecha en el dato anterior y este sea también NaN, se imprimirá un mensaje por pantalla diciendo que ha habido un error y no se ha podido poner la fecha en la medida "i".

## 2.3 Consumo.

Lo que se busca aproximar es el consumo, hasta ahora en la 4ª columna de los datos se tiene el volumen medido por lo que hay que tenerlo en cuenta a la hora de realizar cualquier algoritmo ya que esto ha sido una fuente de error. Para obtener el consumo, se ha realizado la función "calculoconsumo.m", en ella lo que se hace es recorrer el contrato e ir calculando el incremento de volumen. Además, para cuando se quiera calcular el consumo de varios contratos a la vez, se va guardando para cada consumo el ID del contrato al que pertenece. Cuando se llega a un final de contrato, se introduce en el ID y en el consumo un "-1" para que sea más fácil detectarlo.

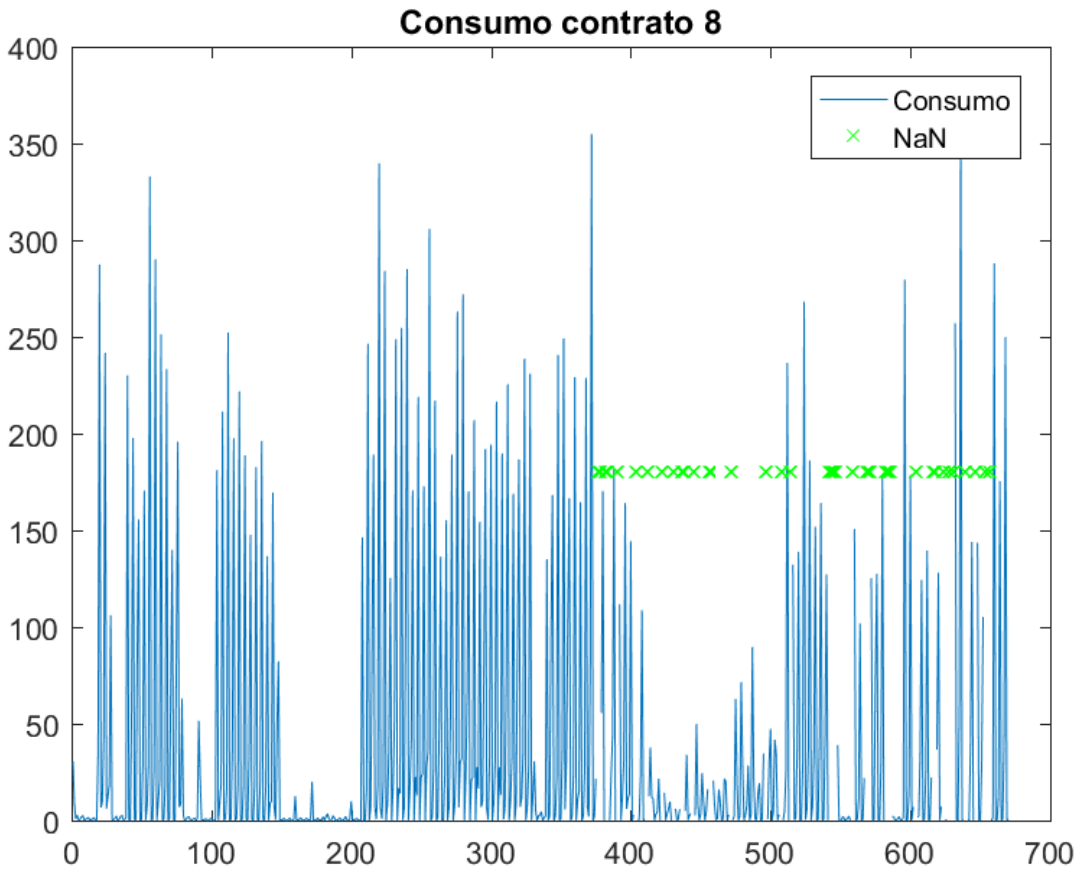



Figura 2-1. Consumo contrato 8.

En la figura anterior se puede observar que el consumo (trazo azul) suele ser más o menos periódico. Esto se debe a que, por ejemplo, si se trata de un contrato de uso doméstico, las personas que habitan la vivienda tienen ciertas rutinas y es muy difícil que las cambien, de forma que muchas veces observando las medidas anteriores se puede obtener una buena aproximación.

En el caso anterior se ve que aproximadamente a partir de la medida 380 empiezan a faltar datos (representado por la "x" verde), la altura a la que se han representado las "x" es totalmente arbitraria. Del análisis del consumo se pueden deducir muchos problemas como, por ejemplo, una fuga de agua. Si un cliente consume ciertos días a la semana muy poco y a partir de cierto momento se nota un aumento constante en todo el consumo puede venir debido a una fuga.

# 3 PRIMERA APROXIMACIÓN

## 3.1 Aproximación inicial.

Como consecuencia de que el consumo en muchos casos es casi periódico se ha decidido hacer un primer algoritmo básico, que se usará como aproximación inicial. Cuando falte un intervalo de puntos, se mira el consumo que se produjo la semana anterior para dicho intervalo y lo toma como aproximación. En caso de que no sea posible coger el dato de la semana anterior, intenta ir otra semana hacia atrás, en caso de que no se pueda coger ninguno de los datos anteriores pasa a buscar una medida válida en los datos futuros de forma análoga. Se podrá ver parte del código en el capítulo 6.

Pretendiendo obtener resultados que sean comparables durante todo el proyecto, se va a seguir una metodología sistemática que va a ser idéntica en todos los algoritmos utilizados:

1. Se introduce un intervalo de puntos (en este caso: 100-300).

2. Se busca un contrato que tenga todo ese intervalo de forma que se puedan quitar determinadas zonas para realizar la aproximación.

3. Se compara el valor real de consumo con el aproximado. En caso de que se usen varios contratos a la vez hay que tener en cuenta que las medidas de las que se disponen no tienen por qué haber empezado en la misma fecha en todos los casos, por lo que si en el contrato 1 se coge el intervalo [100,300], en otro distinto para coger el mismo intervalo no tienen por qué ser los mismos puntos, habrá que anotar la fecha inicial en la que empieza el intervalo en el primer contrato y buscar esa misma fecha en los demás contratos.

Para demostrar que normalmente el consumo de un agregado de contratos tiende a ser más uniforme que los consumos individuales se han elegido de forma aleatoria 10 contratos y se han representado gráficamente tanto sus consumos individuales como el agregado.

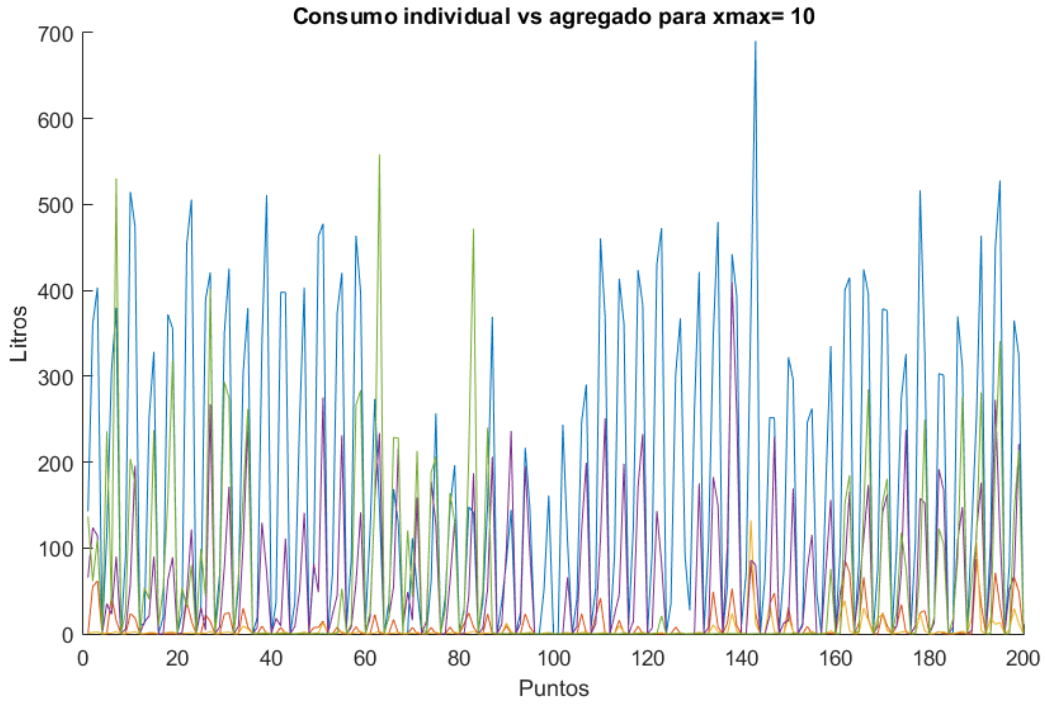



Figura 3-1. Consumo de 5 contratos tenidos en cuenta.

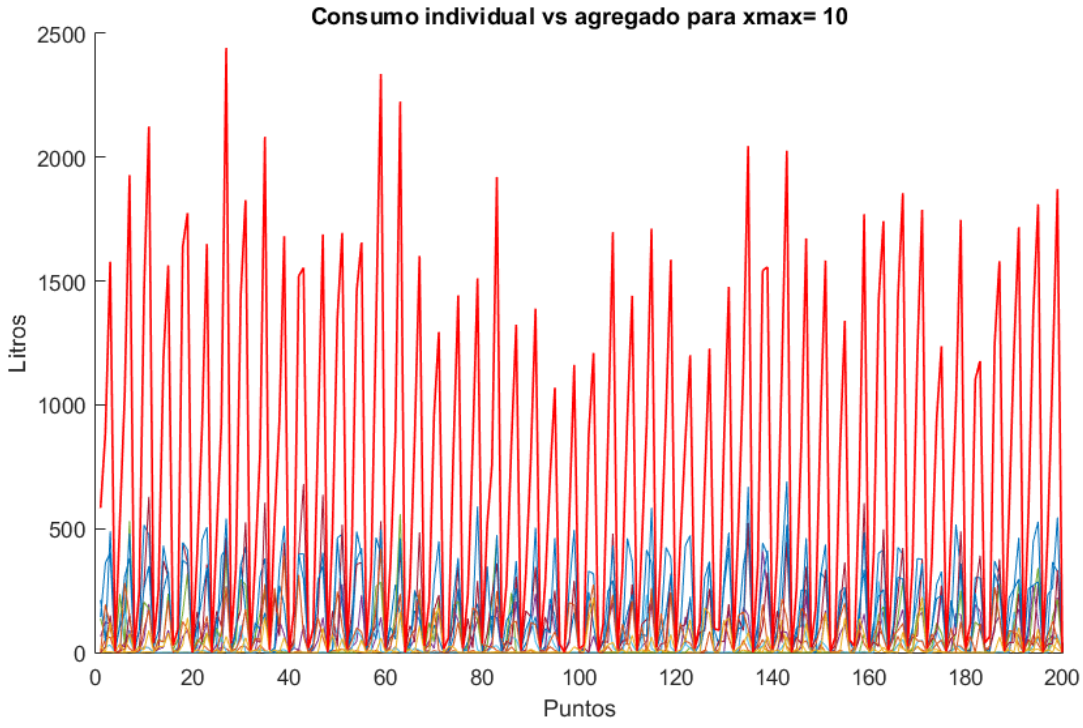



Figura 3-2. Consumo de los 10 contratos individuales y en rojo la suma de todos ellos.

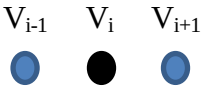
Como se puede observar, los picos de la gráfica en rojo no varían tanto como los que se muestran en la figura 3-1.

Debido a que existe la posibilidad de que en un intervalo el consumo sea "0", se ha decidido no usar ni el error absoluto ni el relativo sino que se usará el error absoluto dividido por la media. Por tanto, cada vez que en tablas o gráficas se hable de error se hace referencia a este.

$$Error = \frac{|Consumo\ Real - Consumo\ Aprox\ |}{\mu}$$

Siendo $\mu$ la media del consumo de los datos disponibles en el contrato que se está aproximando.

En este algoritmo, se hará referencia a dos variables que son X max y P. La primera determina el número de contratos que se tienen en cuenta a la hora de realizar la aproximación (por ejemplo, si X max = 2, se realiza el agregado de 2 contratos válidos y luego se aproxima) y la segunda es el número de puntos consecutivos que faltan. Hay que destacar que P determina el número de puntos que faltan en el volumen, por lo que a la hora de aproximar el consumo, si P es 1 (falta un dato en el volumen) faltarán dos datos en el consumo:




El punto negro se trata de un dato que falta, al calcular el incremento, como hay que restar $V_{i+1}$-$V_i$ y cuando falta un dato el valor es NaN el valor del consumo será NaN (falta), de forma análoga pasa para $V_i$-$V_{i-1}$.

A continuación, se van a realizar varias representaciones gráficas de la evolución del error para los casos X max = 1 y P = 3, X max = 5 y P = 6, X max = 10 y P = 10.

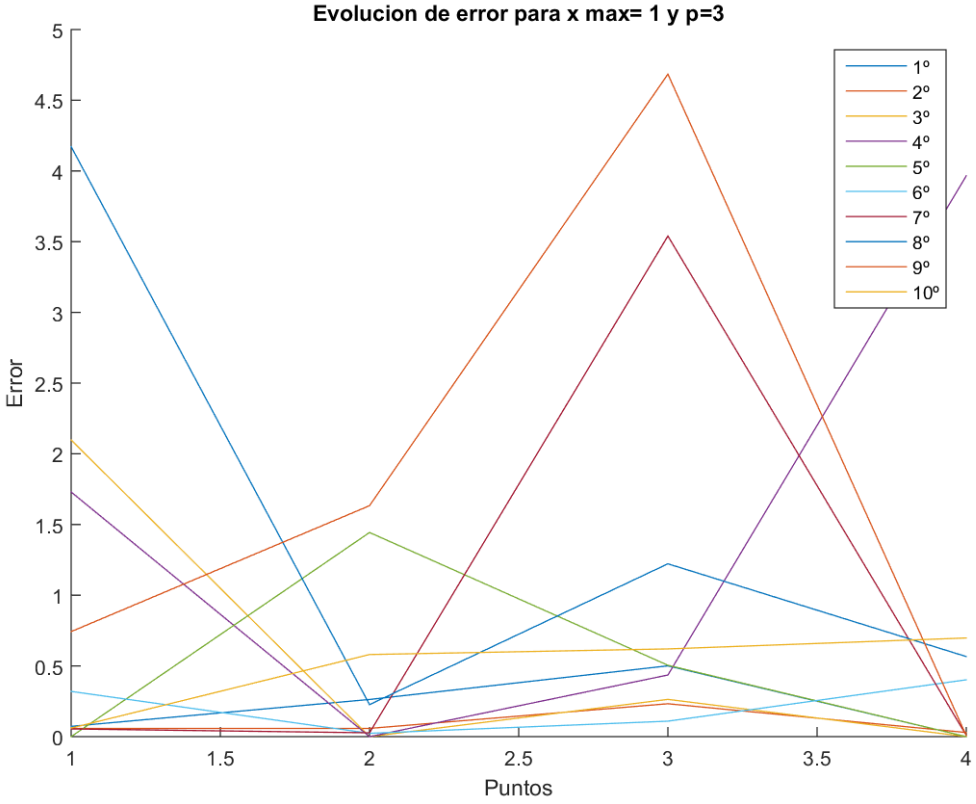



Figura 3-3. Evolución del error para x_max=1 y p=3.

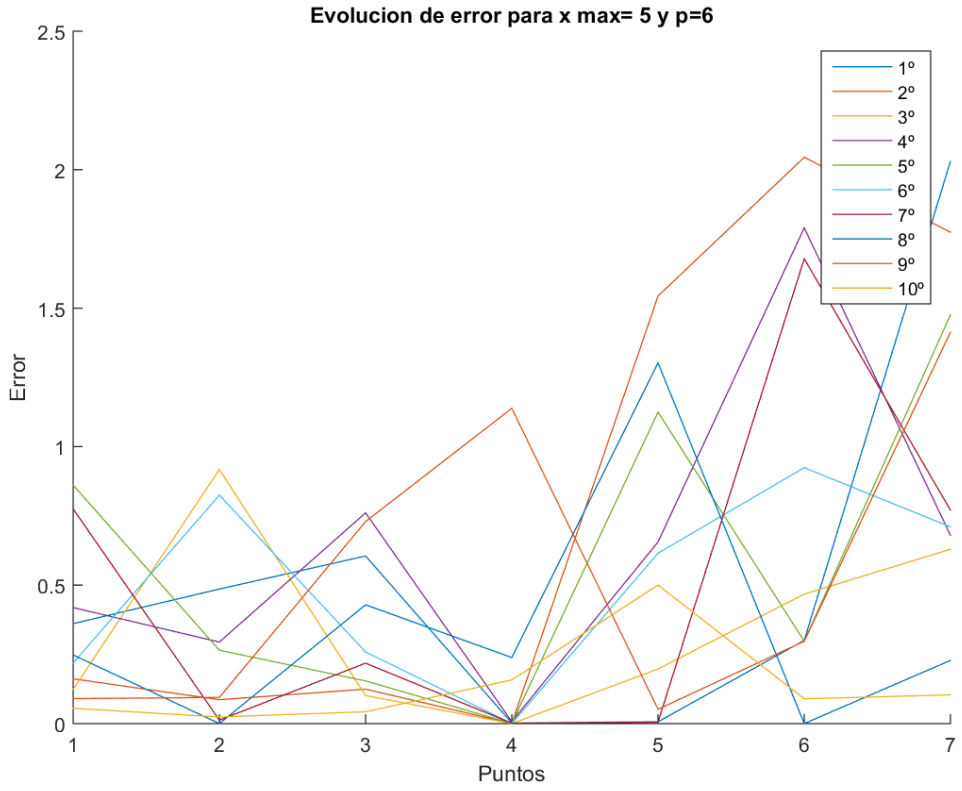



Figura 3-4. Evolución del error para x_max=5 y p=6.

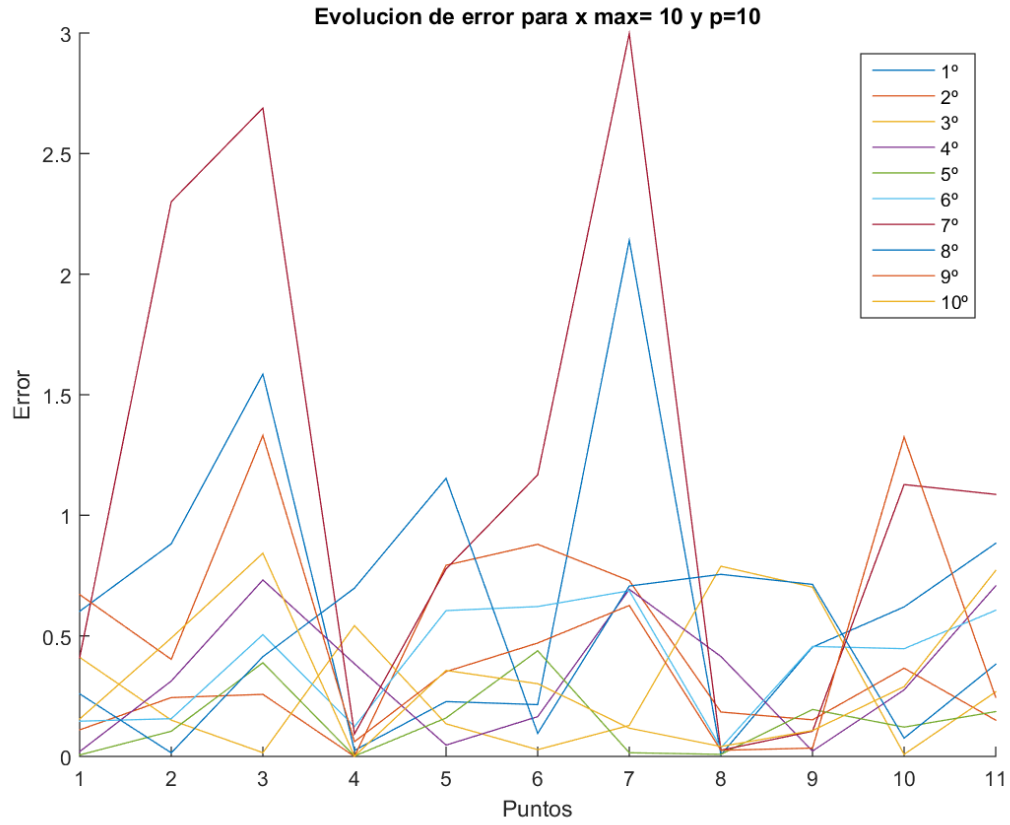



Figura 3-5. Evolución de error para x_max=10 y p=10.

Como se observa en las figuras anteriores, tanto el error como el consumo varían mucho de un contrato a otro y dependen de cuántos puntos se aproximen (a medida que P↑ el error es mayor) y de cuántos contratos se tengan en cuenta (cuando Xmax↑ el error suele disminuir). Debido a esto, se ha decidido hacer una tabla que englobe el error teniendo en cuenta, desde un contrato, hasta diez (X max = 1:10) y desde que falte un dato, hasta que falten 10 puntos consecutivos (P = 1:10). Como se ha dicho anteriormente, el error varía mucho de un contrato a otro, por lo tanto, con el objetivo de obtener unos datos más representativos, se ha decidido realizar cada aproximación 10 veces y realizar la media del error cometido (Error Medio). Un pseudocódigo de la metodología a realizar es:

```
For (x_max=1:10)

    For (p=1:10)

        For (contador=1:10)

            Elección de los puntos a aproximar.
            Suma de volúmenes.
            Cálculo consumo.
            Aproximación de los puntos.
            Cálculo del error.

        End
        Cálculo de error medio.
        Actualización de tabla error.

    End

End
```

Se ha elegido arbitrariamente el caso X max = 10 y P = 10 para representar el consumo real y el consumo aproximado de forma que se vea gráficamente el intervalo elegido:

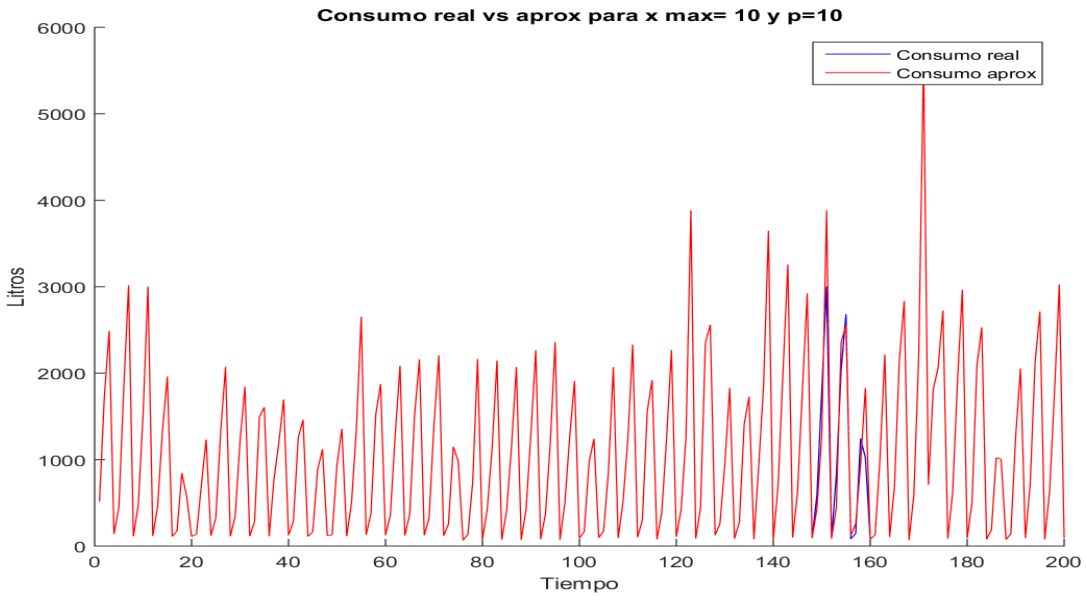



Figura 3-6. Consumo real vs aproximado para el caso de X max=10 y P=10.

Este es sólo un ejemplo aislado y no es representativo. Con el objetivo de obtener unos resultados más representativos, se ha realizado la tabla 3-1, en la que en cada celda aparece el error medio obtenido después de repetir cada aproximación 10 veces con contratos diferentes (es decir, por ejemplo, el caso X max=1 y P=1 se ha realizado para 10 contratos distintos y luego se ha realizado la media del error cometido en cada aproximación).

Tabla 3-1. Evolución del error Medio, aproximación inicial.

| X_max | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.6737 | 0.8561 | 0.7838 | 0.7596 | 0.7389 | 0.7199 | 0.7586 | 0.7822 | 0.7315 | 0.7251 |
| 2 | | 0.6927 | 0.7640 | 0.6492 | 0.5813 | 0.6397 | 0.6890 | 0.6232 | 0.6627 | 0.6467 | 0.6561 |
| 3 | | 0.6455 | 0.6756 | 0.5645 | 0.5616 | 0.6961 | 0.7176 | 0.6375 | 0.6404 | 0.6390 | 0.6528 |
| 4 | | 0.3743 | 0.4055 | 0.3326 | 0.4003 | 0.4795 | 0.5239 | 0.5018 | 0.4936 | 0.4963 | 0.5111 |
| 5 | | 0.3119 | 0.3228 | 0.2796 | 0.3441 | 0.4175 | 0.5000 | 0.4702 | 0.4614 | 0.4687 | 0.4857 |
| 6 | | 0.4488 | 0.5184 | 0.4466 | 0.4645 | 0.4522 | 0.4693 | 0.4252 | 0.4268 | 0.4195 | 0.4194 |
| 7 | | 0.4988 | 0.5338 | 0.4457 | 0.4933 | 0.4783 | 0.5015 | 0.4706 | 0.4515 | 0.4395 | 0.4552 |
| 8 | | 0.3123 | 0.4040 | 0.3340 | 0.3311 | 0.3426 | 0.3847 | 0.3600 | 0.3519 | 0.3496 | 0.3540 |
| 9 | | 0.3888 | 0.4914 | 0.4339 | 0.4346 | 0.4181 | 0.4458 | 0.4139 | 0.4007 | 0.3923 | 0.3936 |
| 10 | | 0.3847 | 0.5484 | 0.4577 | 0.4578 | 0.4545 | 0.5192 | 0.4813 | 0.4602 | 0.4603 | 0.4685 |

Se observa que a medida que van aumentando los contratos tenidos en cuenta (X max), el error va disminuyendo, mientras que cuando van aumentando los datos que faltan (P) el error va aumentando. Para ver mejor la evolución de la tabla se ha representado gráficamente en la figura 3-6, en ella, se ve claramente la tendencia del error.

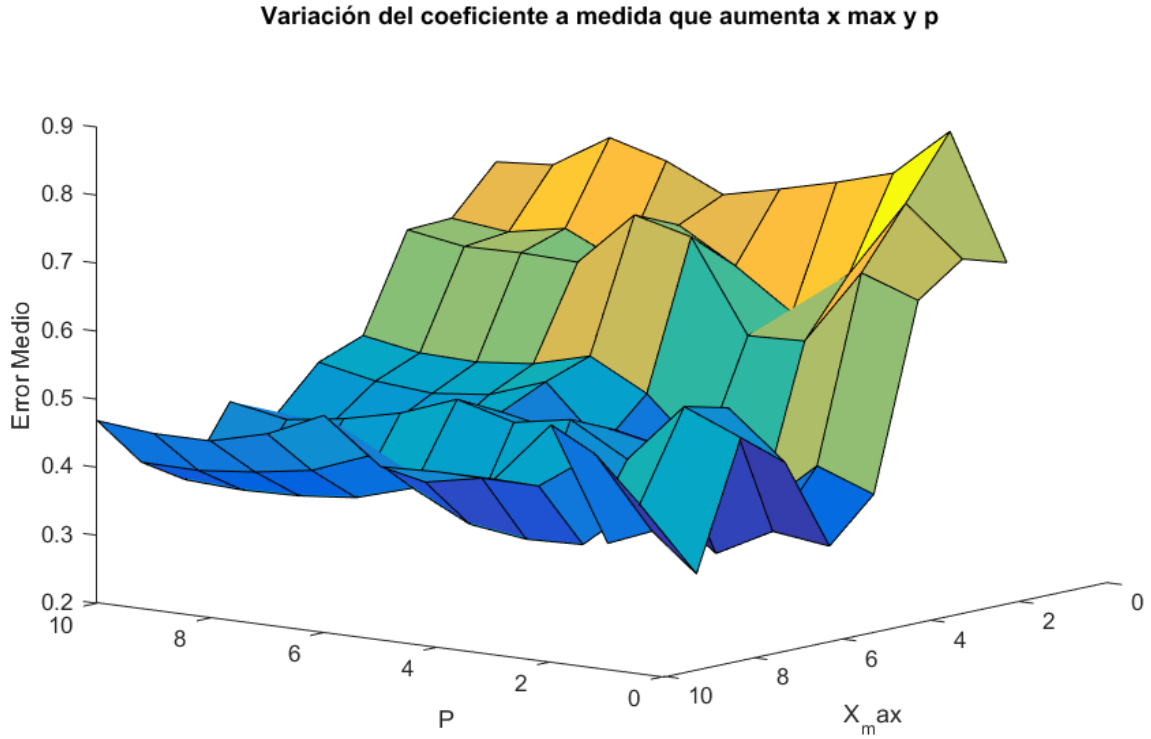



Figura 3-7. Representación gráfica del error medio, aproximación inicial.

De la tabla y de la representación gráfica se puede deducir que el error medio disminuye a medida que se tienen en cuenta más contratos. También se puede observar que en "P=2" hay un error más grande que en el resto de la tabla y a medida que se aumenta "P" el error tiende a disminuir un poco y luego a mantenerse más o menos constante.

El error es muy grande por lo que en el siguiente apartado a costa de aumentar un poco la complejidad se va a realizar una mejora al algoritmo para obtener unos resultados mejores.

## 3.2 Corrección utilizando consumo total registrado.

En este apartado, se ha implementado una mejora al código "aproximacióninicial.m", se ha introducido un coeficiente α de forma que a posteriori de la aproximación se pueda corregir la misma para minimizar el error. Se parte de la base de que aunque no se conozca un intervalo de medidas siempre hay un volumen inicial y uno final que se conocen, por lo tanto, el consumo desde el inicio al final del intervalo es conocido:


$$V_{i+n} - V_i = \alpha * \sum_{j=i}^{i+n} C_j$$

Figura 3-8. Coeficiente Alfa.

Siendo V el volumen y $C_j$ los consumos aproximados. De esta forma, se garantiza que el consumo total del intervalo sea correcto aunque los consumos individuales no lo sean. En las siguientes figuras se ve el efecto que tiene sobre los resultados anteriores:

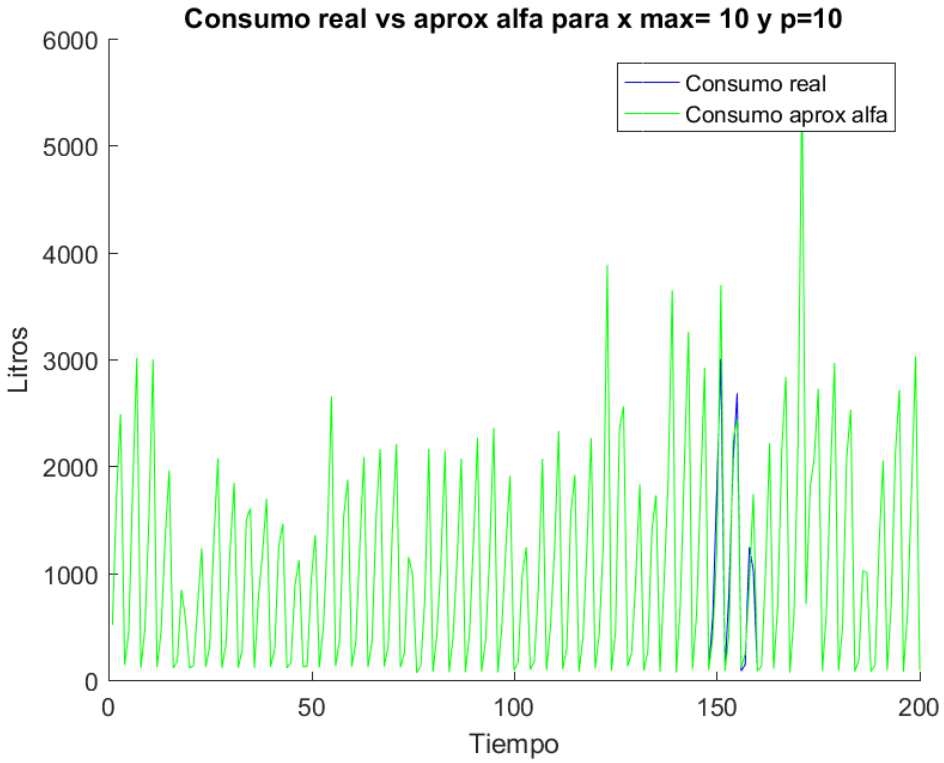



Figura 3-9. Consumo real vs Consumo aproximado con α.

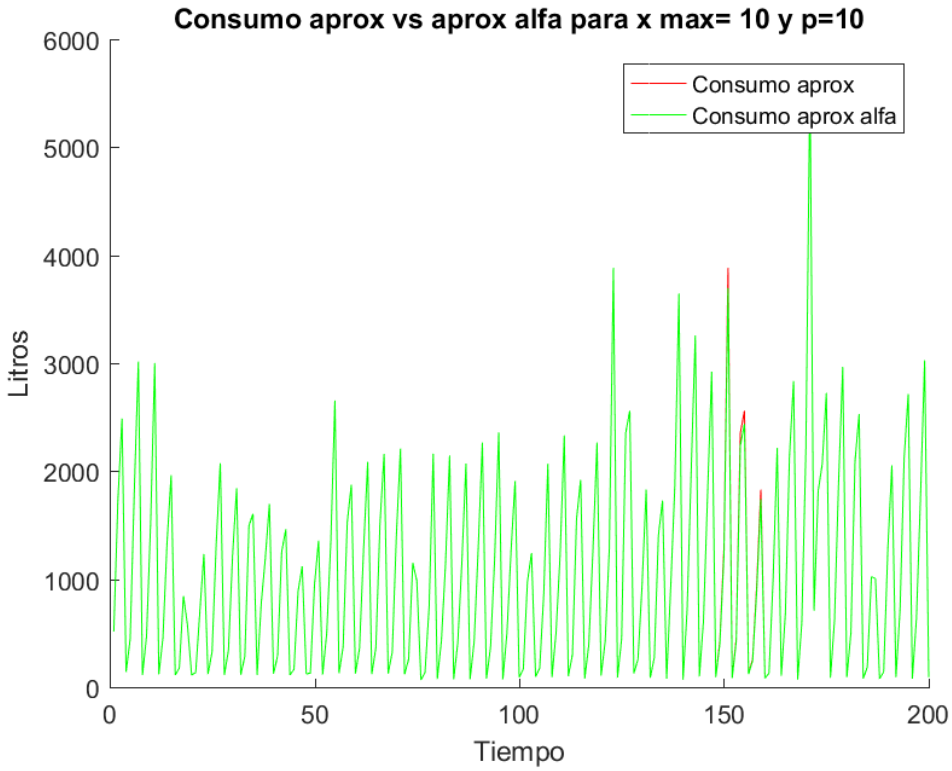



Figura 3-10. Consumo aproximado vs Consumo aproximado con α.

Como se observa en la figura 3-10, el "Consumo aprox α" es muy similar al "Consumo aprox" del caso anterior para "X max=10 y P=10", pero si comparamos las tablas 3-1 y 3-2 se ve que con la modificación del programa dicho error, en general, ha disminuido. Se va a representar gráficamente para que se vea de forma clara.

Tabla 3-2. Error Medio, aproximación inicial con coeficiente α.

| X max | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.0808 | 0.4163 | 0.6253 | 0.6040 | 0.6787 | 0.6746 | 0.7576 | 0.8428 | 0.7948 | 0.7972 |
| 2 | | 0.2486 | 0.5210 | 0.6940 | 0.6002 | 0.6625 | 0.6241 | 0.5679 | 0.6603 | 0.6710 | 0.6890 |
| 3 | | 0.5406 | 0.5474 | 0.4736 | 0.4737 | 0.6830 | 0.6648 | 0.5873 | 0.5636 | 0.5852 | 0.5627 |
| 4 | | 0.2424 | 0.3290 | 0.2954 | 0.3688 | 0.4748 | 0.5103 | 0.4996 | 0.4791 | 0.4858 | 0.4803 |
| 5 | | 0.2645 | 0.3075 | 0.2864 | 0.3711 | 0.4419 | 0.4983 | 0.4643 | 0.4539 | 0.4615 | 0.4591 |
| 6 | | 0.1575 | 0.2700 | 0.2698 | 0.3270 | 0.3557 | 0.3626 | 0.3230 | 0.3190 | 0.3242 | 0.3231 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 7 | 0.1323 | 0.2598 | 0.2531 | 0.2990 | 0.3115 | 0.3357 | 0.3233 | 0.3297 | 0.3257 | 0.3235 |
| 8 | 0.1380 | 0.2497 | 0.2219 | 0.2205 | 0.2548 | 0.2825 | 0.2664 | 0.2594 | 0.2738 | 0.2761 |
| 9 | 0.0823 | 0.2444 | 0.2479 | 0.2657 | 0.2931 | 0.3071 | 0.2909 | 0.2786 | 0.2820 | 0.2786 |
| 10 | 0.1256 | 0.2104 | 0.2118 | 0.2363 | 0.2804 | 0.3006 | 0.2868 | 0.2747 | 0.2918 | 0.3004 |

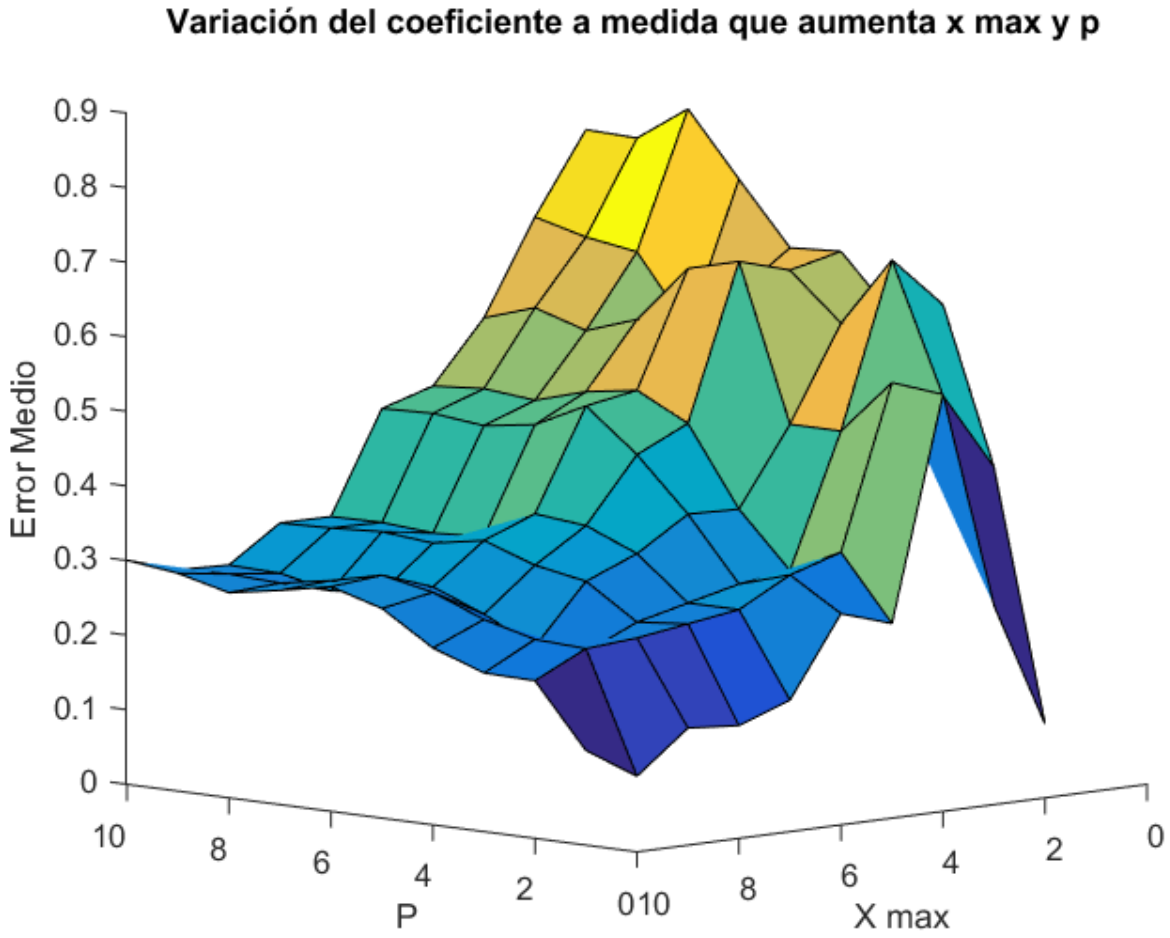



Figura 3-11. Error medio aproximación inicial con α.

Si se compara la figura 3-11 con la 3-7, se puede observar que ahora el error es más suave. A medida que aumenta X max el error disminuye y a su vez, a medida que aumenta P, el error aumenta. Para ver visualmente la mejora producida se van a representar ambas tablas de error en una sola figura:

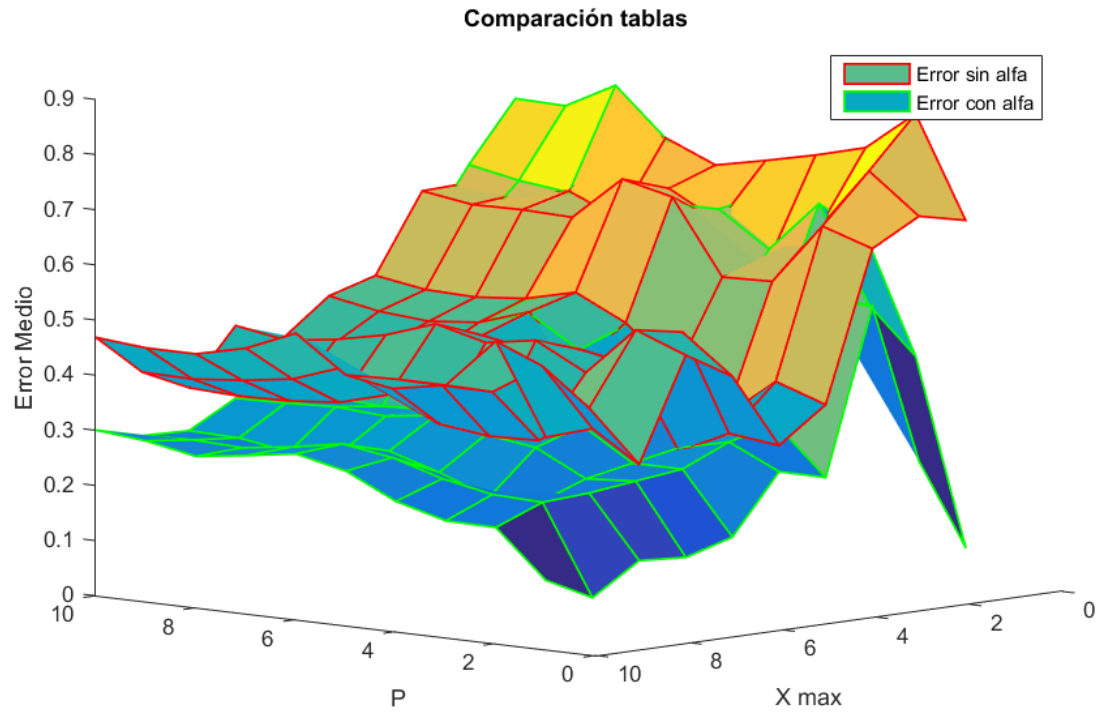



Figura 3-12. 1ª Comparación gráfica de las tablas 3-1 y 3-2.

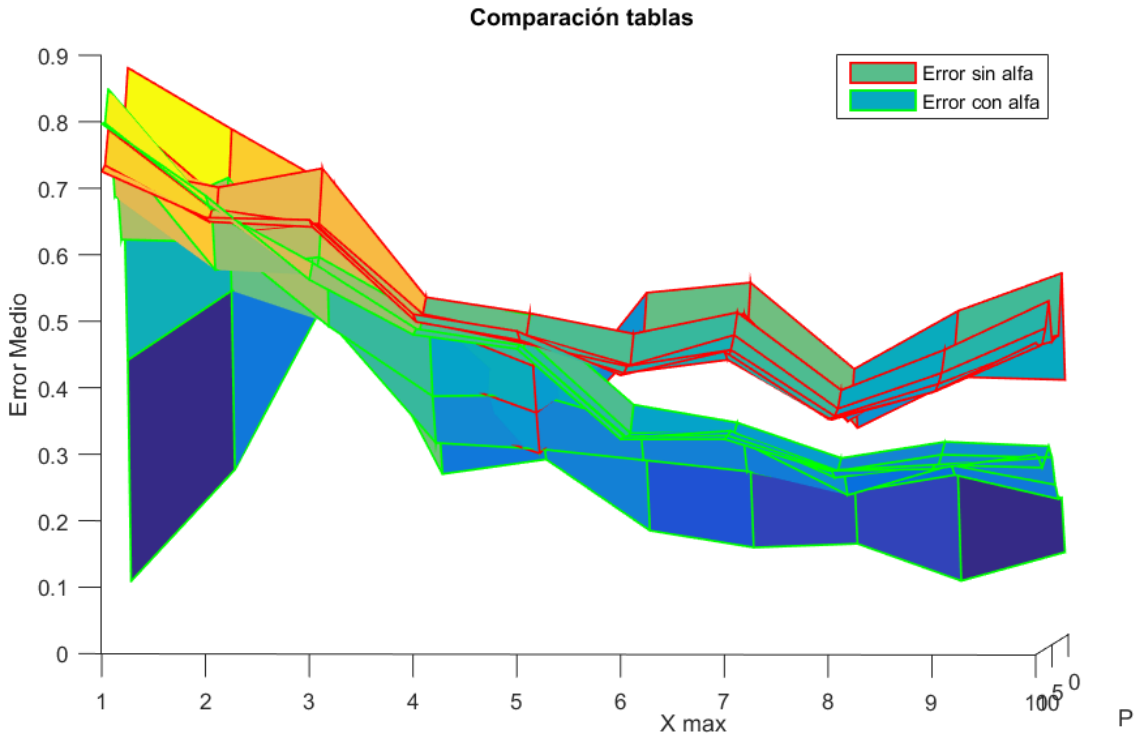



Figura 3-13. 2ª Comparación gráfica de las tablas 3-1 y 3-2.

Se ha conseguido el objetivo de disminuir el error medio ya que, en la representación gráfica, se observa que generalmente el plano del error con el coeficiente α está por debajo del error sin el coeficiente, aunque hay ciertos puntos (por ejemplo X max=1 y P=10) que es al contrario. Para

cuantificar la mejora producida, en la siguiente tabla aparece el porcentaje en que se ha reducido el error medio entre ambas aproximaciones:

Tabla 3-3. Mejora en % comparando aproximación inicial sin y con α.

| X max | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 88.01 | 51.37 | 20.22 | 20.49 | 8.14 | 6.29 | 0.13 | -7.74 | -8.65 | -9.93 |
| 2 | | 64.11 | 31.81 | -6.90 | -3.25 | -3.57 | 9.42 | 8.87 | 0.36 | -3.75 | -5.01 |
| 3 | | 16.25 | 19.00 | 16.11 | 15.65 | 1.88 | 7.36 | 7.87 | 12.00 | 8.42 | 13.80 |
| 4 | | 35.26 | 18.87 | 11.16 | 7.88 | 0.97 | 2.61 | 0.45 | 2.95 | 2.12 | 6.03 |
| 5 | | 15.19 | 4.77 | -2.43 | -7.85 | -5.84 | 0.33 | 1.25 | 1.62 | 1.53 | 5.47 |
| 6 | | 65.00 | 47.92 | 39.57 | 29.61 | 21.35 | 22.73 | 24.03 | 25.26 | 22.71 | 22.96 |
| 7 | | 73.48 | 51.33 | 43.21 | 39.40 | 34.87 | 33.06 | 31.30 | 26.98 | 25.90 | 28.94 |
| 8 | | 55.81 | 38.18 | 33.55 | 33.40 | 25.61 | 26.56 | 26.00 | 26.30 | 21.68 | 22.01 |
| 9 | | 78.82 | 50.25 | 42.86 | 38.88 | 29.89 | 31.11 | 29.72 | 30.47 | 28.13 | 29.21 |
| 10 | | 67.36 | 61.63 | 53.72 | 47.37 | 38.30 | 42.10 | 40.42 | 40.32 | 36.61 | 35.90 |

Se ha mejorado bastante el resultado obtenido en algunas zonas y en las que se ha empeorado ha sido muy poco. Lo máximo que ha empeorado es para "X max=1 y P=10" un 10%. Como el global es muy positivo el error "base", con el cual se va a comparar el siguiente algoritmo va a ser el que se ha obtenido después de realizar la mejora, es decir, el de la tabla 3-2.

La siguiente tabla se ha realizado teniendo en cuenta 100 aproximaciones, por lo que es más representativa y, al tratarse de una tabla más pequeña, se ve más clara la evolución del error, es más intuitiva:

Tabla 3-4. Comparación de errores para 100 aproximaciones, aproximación inicial.

| N Contratos / Error(%) | 1 | 3 | 5 | 10 |
|---|---|---|---|---|
| Aproximación Inicial | 72.2 | 52.5 | 41.9 | 30.0 |
| Aproximación Inicial a | 41.7 | 34.2 | 25.1 | 17.4 |

Este error todavía no es válido ya que es demasiado alto (en algunas zonas de la tabla se puede ver que el error supera el 70%), por lo que en el siguiente apartado se va a intentar mejorarlo mediante otro algoritmo más complejo, el elegido es el de los mínimos cuadrados.

# 4 MÍNIMOS CUADRADOS

El 1 de enero de 1801 el astrónomo italiano Joseph Piazzi descubrió el planetoide Ceres. Los astrónomos fueron capaces de observarlo durante sólo 41 días ya que lo perdieron tras el Sol. Las observaciones fueron de sólo 9 grados de su órbita y fueron publicadas en Junio de 1801. Con el objetivo de detectar Ceres de nuevo, muchos astrónomos intentaron predecir la órbita de Ceres con los datos obtenidos por Piazzi, era un problema difícil debido al déficit de las medidas disponibles. De hecho, Laplace pensó que el problema no se podía resolver usando los métodos numéricos disponibles en la época.

En Septiembre de 1801, se publicaron diferentes aproximaciones de la órbita, entre todas ellas fueron las de Carl Friedrich Gauss las que destacaron, tenía tan sólo 24 años. Cuando Ceres fue redescubierto el 7 de Diciembre de 1801 estaba casi exactamente donde Gauss predijo. Para calcular la órbita, Gauss estimó los parámetros que la describía usando su método de los mínimos cuadrados. Esta metodología descubierta por Gauss fue publicada en 1809 como "Theoria motus corporum coelestium in sectionibus conicus solem ambientum".

Anteriormente en 1805, de una forma independiente, Adrien-Marie Legendre publicó "Nouvelles Méthodes pour la Détermination des Orbites des Comètes" en donde se presentaba una metodología similar. La principal contribución de Gauss con respecto a Legendre fue que Gauss introdujo la noción de la distribución normal para modelar la observación de los errores, demostrando que el método propuesto daba una estimación óptima bajo la suposición de que los errores empíricos seguían esta distribución normal (también conocida como distribución de Gauss).

## 4.1 Teoría.

### 4.1.1 Método de los mínimos cuadrados.

Se supone que los escalares $y_k$, k=0,1,… representan la salida (muestreada) de un sistema dinámico. Se asume asimismo que el valor de la salida $y_k$ puede ser aproximado por una expresión del tipo: $y_k \approx m_k\theta$ donde $m_k \in \mathbb{R}^{1 \times n}{}_m$ es un vector fila, denominado regresor, que está constituido por valores pasados de las entradas y salidas del sistema. Por otro lado, θ es un vector columna donde se concentran los distintos parámetros que determinan la relación entre las entradas y salidas del sistema.

Para un sistema como el que sigue con una salida "y" y una entrada "u" el regresor y el vector de parámetros son:

$$y(k)+a_1y(k-1)+...+a_ny(k-n) = b_1u(k-1)+ ... +b_nu(k-n)$$
$$\downarrow$$
$$y(k)=m(K)\theta$$

Siendo:

$$m(k)=[\ -y(k-1)\ ...\ y(k-n)\ u(k-1)\ ...\ u(k-n)]$$

$$\theta=[\ a_1...a_n\ b_1...b_n\ ]^T$$

El error de predicción se define como:

$$e(k,\hat{\theta}) = y(k)-\hat{y}(k) = y(k)-m(k)\hat{\theta}$$

Partiendo de N pares de (y(k),m(k)) se plantea:

$$E(N,\theta)=Y(N)-M(N)\theta$$
$$E(N,\theta)=[e(n,\theta)...e(N,\theta)]^T$$
$$Y(N,\theta)=[y(n)...y(N)]^T$$
$$M(N)=\begin{bmatrix} m(n) \\ \vdots \\ m(N) \end{bmatrix}$$

Se trata de un sistema de ecuaciones sobre determinado incompatible, por ello, se buscará la pseudosolución ($\theta^*$) óptima del sistema en el sentido de los mínimos cuadrados, es decir minimizando:

$$J(\theta)=\|E(N,\theta)\|^2=\sum_{k=n}^{N} e^2(k,\theta)$$

El índice "J" se puede reescribir como:

$$J(\theta)=(Y(N)-M(N)\theta)^T(Y(N)-M(N)\theta)$$

El mínimo será el valor de "θ" que hace la derivada 0, por tanto:

$$\frac{dJ(\theta)}{d\theta}=0 \rightarrow 2\big(M(N)\theta-Y(N)\big)^T M(N)=0 \rightarrow \boxed{\theta^*=[M(N)^TM(N)]^{-1}M(N)^TY(N)}$$

$\theta^*$ se conoce como "Estimador de Mínimos Cuadrados" y se deduce que tiene que existir la inversa de $M^T(N)M(N)$ para aplicar este método.

### 4.1.2 Mínimos Cuadrados Ponderados.

Los Mínimos Cuadrados ponderados permiten darle más peso a ciertas observaciones (normalmente las más nuevas) que a otras para el cálculo de los parámetros, de forma que es mejor para procesos en los que la dinámica varía a lo largo del tiempo.

Se parte de "J(θ)" añadiéndole el vector de ponderación "w(n)":

$$J(\theta)=\sum_{k=n}^{N} w(k)e^2(k,\theta)=\|E(N,\theta)^T W(N)E(N,\theta)\|^2$$

$$W(N)=\begin{bmatrix} w(n) & & \\ & \ddots & \\ & & w(N) \end{bmatrix}$$

La solución es:

$$\boxed{\theta^*=[M(N)^T W(N)M(N)]^{-1}M(N)^T W(N)Y(N)}$$

Usualmente se aplica un esquema de olvido exponencial:

$$w(k)=\lambda^{N-K} \qquad \lambda \in (0,1)$$

Un λ próximo a 1 "recuerda" muchas medidas por lo que se adapta más lentamente al producirse un cambio, mientras que un λ menor "recuerda" menos por lo que se adapta más rápidamente.

## 4.2 Primer Regresor.

El código en Matlab de los algoritmos de mínimos cuadrados es muy parecido al del algoritmo anterior, en el capítulo 6 se ha incluido la parte que cambia. En este apartado, se ha optado por utilizar un regresor que tenga en cuenta los datos anteriores del contrato que se esté aproximando, en Matlab queda de la siguiente forma:

```
  medidas=120;
   z=5;
  for j=(i-medidas):(i-1)       % El bucle va hasta i-1 porque la última
medida que se quiere tener en cuenta es la anterior a la actual.
      M(z-4,1)=ind(j-1,2);
      M(z-4,2)=ind(j-2,2);
      M(z-4,3)=ind(j-3,2);
      M(z-4,4)=ind(j-4,2);
      Y(z-4)=consumo_x_aisl(j,2);
      z=z+1;
  end
  teta=M'*M\M'*Y';
  Y_final=[ind(i-1,2),ind(i-2,2),ind(i-3,2),ind(i-4,2)]*teta;
  consumo_x_aisl(i,2)=Y_final;
  ind(i,2)=Y_final;     %Se actualizan los consumos para que los regresores se
                  encuentren siempre disponibles.
```

No se tiene en cuenta el consumo agregado, solo se tienen en cuenta los consumos anteriores. El bucle va desde "i-medidas" a "i-1" porque se ha querido tener en cuenta los datos del mes anterior, es decir, como se toman 4 medidas diarias por 30 días de un mes (suponiendo que todos los meses tienen 30 días) son un total de 120 datos.

Hay que destacar que el regresor siempre está disponible para la aproximación, es decir, se realiza una reconstrucción en línea de los datos que faltan, de forma que para aproximar un dato, siempre se tengan los anteriores.De esta forma se garantiza que no hay problemas con el regresor. En caso de que por cualquier circunstancia no se pueda obtener el regresor (por ejemplo se intenta aproximar el dato número 3 y no hay suficientes datos anteriores) no se puede aproximar, para solucionar esto se podrían rellenar los datos "peligrosos" con una aproximación inicial y luego ir mejorándola en bucle.

Para decidir cuántas medidas anteriores tener en cuenta a la hora de construir el regresor "M" se han representado gráficamente 3 tablas de error, teniendo en cuenta las 4, 8 y 12 medidas anteriores, es decir, teniendo en cuenta las medidas del día anterior, de hace 2 y de hace 3 días.

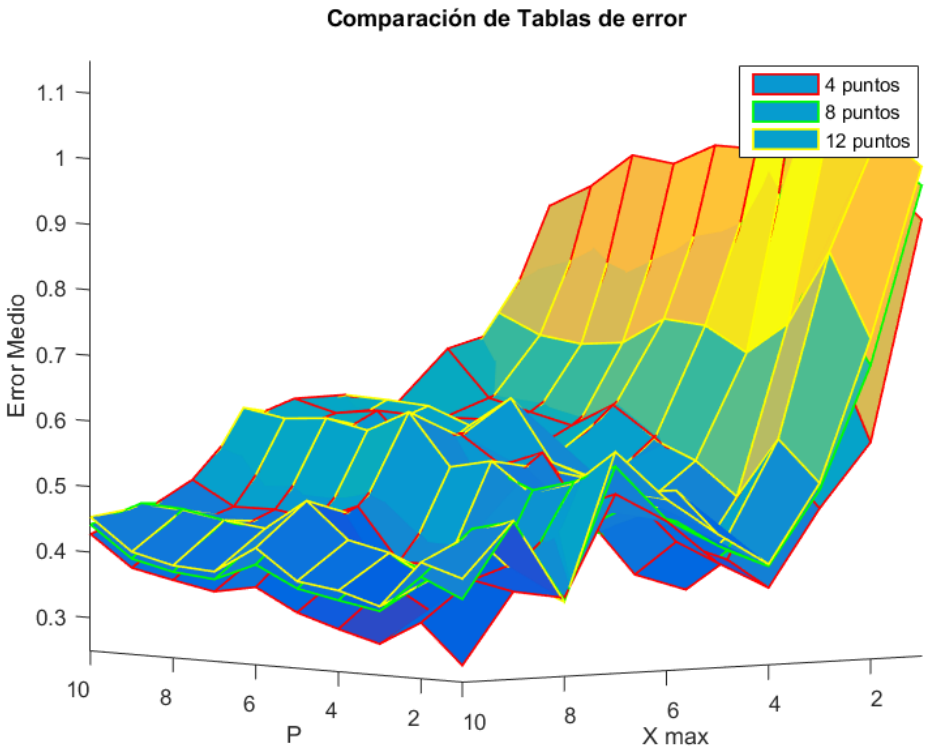



Figura 4-1. Comparación de tablas errores, mínimos cuadrados primer regresor.

Se puede observar en el gráfico que dependiendo de los valores que toman X max y P, existen zonas en las que se aproxima mejor teniendo en cuenta 4 puntos y otras en las que se obtiene un menor error teniendo en cuenta 8 u 12. Sabiendo esto, se ha decidido realizar el regresor teniendo en cuenta 4 medidas anteriores debido a que hay muchas zonas en las que se obtiene una mejor aproximación y en donde aproxima peor que el resto tampoco lo hace con mucha diferencia, además, al tener en cuenta 4 medidas es un algoritmo más versátil ya que si por ejemplo se tuvieran en cuenta 12 medidas, se fuerza a que exista una zona disponible de puntos más grande para realizar la aproximación.

Por lo que el regresor "M" está compuesto por:

Consumo individual (i-1) … Consumo individual (i-4)

La tabla del error para dicho regresor queda:

Tabla 4-1. Evolución del error, mínimos cuadrados primer regresor.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.9164 | 0.9752 | 1.0437 | 1.1242 | 1.0026 | 1.0026 | 0.9691 | 0.9767 | 0.9242 | 0.8887 |
| 2 | | 0.5811 | 0.7283 | 0.6346 | 0.6270 | 0.6464 | 0.6772 | 0.6408 | 0.6494 | 0.6441 | 0.7015 |
| 3 | | 0.4846 | 0.5556 | 0.4843 | 0.5234 | 0.5719 | 0.6155 | 0.5787 | 0.5940 | 0.6099 | 0.6796 |
| 4 | | 0.3680 | 0.4057 | 0.3543 | 0.4673 | 0.5033 | 0.5422 | 0.5352 | 0.5874 | 0.5805 | 0.6002 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 5 | 0.4105 | 0.4372 | 0.3817 | 0.4585 | 0.5241 | 0.5838 | 0.5610 | 0.5955 | 0.6021 | 0.6180 |
| 6 | 0.4787 | 0.5211 | 0.4784 | 0.5669 | 0.5422 | 0.5852 | 0.5467 | 0.6092 | 0.6000 | 0.6166 |
| 7 | 0.5240 | 0.4992 | 0.4728 | 0.5062 | 0.5088 | 0.5704 | 0.5475 | 0.5678 | 0.5741 | 0.5877 |
| 8 | 0.3703 | 0.4228 | 0.3858 | 0.4041 | 0.4036 | 0.4830 | 0.4576 | 0.4672 | 0.4659 | 0.5017 |
| 9 | 0.3841 | 0.3965 | 0.3432 | 0.3807 | 0.3961 | 0.4433 | 0.4164 | 0.4327 | 0.4379 | 0.4555 |
| 10 | 0.2759 | 0.3359 | 0.2979 | 0.3150 | 0.3349 | 0.3683 | 0.3560 | 0.3680 | 0.3814 | 0.4277 |

Si se compara esta tabla con la 3-2, se ve que los valores obtenidos están muy distantes, por lo que se va a realizar la corrección teniendo en cuenta el consumo total del intervalo (introduciendo el mismo coeficiente α que en el caso anterior) con el objetivo de mejorar el error. Se realiza de forma análoga al apartado anterior y la tabla de error obtenida es:

Tabla 4-2. Evolución del error, mínimos cuadrados primer regresor utilizando coeficiente α.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.3713 | 0.5578 | 0.5704 | 0.7326 | 0.7116 | 0.6716 | 0.7019 | 0.7503 | 0.6971 | 0.6737 |
| 2 | | 0.2945 | 0.5261 | 0.4273 | 0.4306 | 0.5308 | 0.5444 | 0.5297 | 0.5375 | 0.5429 | 0.5854 |
| 3 | | 0.3574 | 0.3572 | 0.3390 | 0.4021 | 0.4891 | 0.5121 | 0.4906 | 0.4933 | 0.5154 | 0.5638 |
| 4 | | 0.2958 | 0.2872 | 0.2686 | 0.4099 | 0.4583 | 0.4847 | 0.4788 | 0.5318 | 0.5243 | 0.5228 |
| 5 | | 0.2611 | 0.2541 | 0.2519 | 0.3468 | 0.4577 | 0.4995 | 0.4893 | 0.5207 | 0.5201 | 0.5223 |
| 6 | | 0.2118 | 0.3198 | 0.3078 | 0.3893 | 0.4193 | 0.4481 | 0.4417 | 0.4793 | 0.4742 | 0.4902 |
| 7 | | 0.2250 | 0.2330 | 0.2763 | 0.2941 | 0.3314 | 0.3770 | 0.3896 | 0.3935 | 0.3979 | 0.4085 |
| 8 | | 0.2295 | 0.2969 | 0.2987 | 0.3220 | 0.3294 | 0.3802 | 0.3650 | 0.3741 | 0.3656 | 0.3830 |
| 9 | | 0.1753 | 0.2816 | 0.2797 | 0.3142 | 0.3553 | 0.3920 | 0.3754 | 0.3787 | 0.3782 | 0.3909 |
| 10 | | 0.1596 | 0.2535 | 0.2485 | 0.2720 | 0.3081 | 0.3232 | 0.3161 | 0.3281 | 0.3424 | 0.3899 |

Se ha conseguido reducir el error obtenido en la tabla 4-1 de forma significativa, aunque todavía hay zonas donde la aproximación inicial es mejor, comparando la tabla 4-2 con la tabla 3-2:

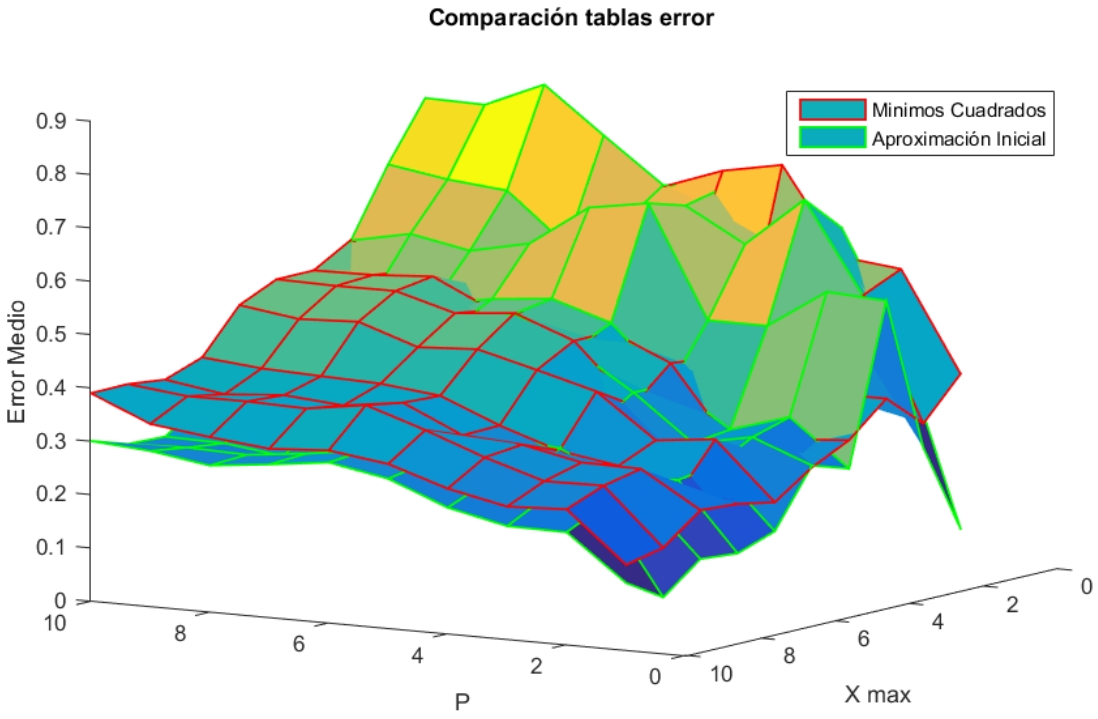


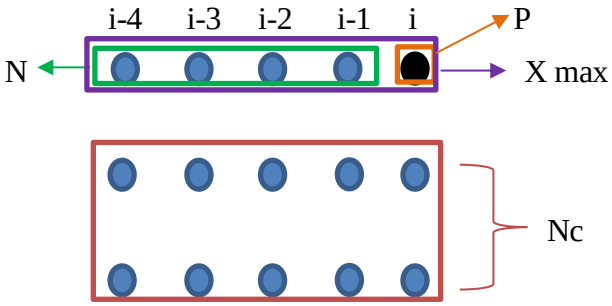

Figura 4-2. Comparación de errores de tablas 3-2 y la 4-2.

De la comparación de ambas tablas se deduce que hasta X max=4 en la mayoría de los puntos el método de los mínimos cuadrados aproxima mejor. Pero a partir de ahí, se invierte esta tendencia y es la aproximación inicial la que obtiene un error menor, aunque es verdad que la diferencia entre ambos errores no es muy grande. En el siguiente subapartado, con el objetivo de mejorar el error obtenido con el algoritmo de los Mínimos Cuadrados se va a aumentar la complejidad del regresor y se va a tener en cuenta un agregado del consumo de varios contratos, la corrección realizada teniendo en cuenta el consumo total del intervalo y un factor de olvido.

## 4.3 Segundo regresor

Para intentar mejorar los resultados obtenidos previamente, en este regresor (además de las medidas anteriores del contrato que se pretende aproximar) se han tenido en cuenta los consumos que se producen durante "N" medidas anteriores para "Nc" contratos distintos, al igual que en los algoritmos anteriores se puede aproximar un solo contrato o un agregado de contratos. Por ejemplo:

Si X max = 1, P=1, N = 4 y Nc = 2 quedaría de la siguiente forma:




Siendo X max el número de contratos para los que se quiere aproximar el punto i-ésimo, Nc el número de contratos que se tienen en cuenta para calcular el agregado, N el número de medidas anteriores del contrato actual que se tienen en cuenta y P el número de puntos que se quieren aproximar. La idea es apoyarse en el consumo agregado de otros contratos para el periodo que se quiere aproximar, en este apartado se cogen contratos al azar que contengan la zona que se necesita pero como mejora futura se pueden ordenar los contratos con algún tipo de coeficiente de forma que para calcular el agregado se tengan en cuenta los contratos que sean similares.

El código de Matlab que implementa este regresor es el siguiente:

```
z=5;
for j=(i-medidas):(i-1)        % El bucle va hasta i-1 porque la última
medida que se quiere tener en cuenta es la anterior a la actual.
      M(z-4,1)=ind(j-1,2);   %M es el regresor.
      M(z-4,2)=ind(j-2,2);
      M(z-4,3)=ind(j-3,2);
      M(z-4,4)=ind(j-4,2);
      M(z-4,5)=ag(j,2);
      M(z-4,6)=ag(j-1,2);
      M(z-4,7)=ag(j-2,2);
      M(z-4,8)=ag(j-3,2);
      M(z-4,9)=ag(j-4,2);
      Y(z-4)=consumo_x_aisl(j,2);    %Hay que modificar el segundo
indice.
       z=z+1;
end
teta=M'*M\M'*Y';                %Vector de parámetros.
Y_final=[ind(i-1,2),ind(i-2,2),ind(i-3,2),ind(i-4,2),ag(i,2),
ag(i-1,2),ag(i-2,2),ag(i-3,2),ag(i-4,2)]*teta;
```

El bucle va desde "X" medidas anteriores hasta la actual y va guardando todas las medidas necesarias (individuales y agregadas) en el regresor "M", este está compuesto por:

| | |
|---|---|
| Consumo individual (i-1) | Consumo de los contratos agregados (i) |
| . | . |
| . | . |
| . | . |
| Consumo individual (i-4) | Consumo de los contratos agregados (i-4) |

### 4.3.1 Aproximación para un contrato cambiando agregado.

Para ver el efecto que tiene aumentar el número de contratos agregados, se ha decidido realizar una primera aproximación para un solo contrato (X max = 1), teniendo en cuenta las 120 medidas anteriores (4 medidas diarias 30 días al mes, son los datos del mes anterior, suponiendo que se trate de un mes de 30 días) y variando tanto el número de puntos a aproximar (P = 1:10) como el número de contratos que se tienen en cuenta para realizar el agregado (Nc = 2:10), se realizará una tabla similar a la de los algoritmos anteriores para ver la evolución del error.

Tabla 4-3. Evolución del error, Nc cambiante y X max = 1.

| Nc | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | - | - | - | - | - | - | - | - | - | - |
| 2 | | 0.8145 | 0.8817 | 1.0083 | 0.9111 | 0.8101 | 0.8147 | 0.7530 | 0.7326 | 0.6874 | 0.6713 |
| 3 | | 0.7769 | 0.8502 | 0.9634 | 0.9489 | 0.8884 | 0.8826 | 0.8836 | 0.8789 | 0.8286 | 0.8119 |
| 4 | | 0.7216 | 0.8190 | 0.9186 | 0.9059 | 0.8327 | 0.8210 | 0.8192 | 0.8166 | 0.7581 | 0.7378 |
| 5 | | 0.7005 | 0.7743 | 0.8911 | 0.8709 | 0.8199 | 0.8039 | 0.7774 | 0.7718 | 0.7369 | 0.7341 |
| 6 | | 0.7081 | 0.7707 | 0.9029 | 0.8772 | 0.8018 | 0.7866 | 0.7474 | 0.7408 | 0.6972 | 0.6895 |
| 7 | | 0.6778 | 0.7664 | 0.8871 | 0.8681 | 0.7843 | 0.7781 | 0.7400 | 0.7346 | 0.6889 | 0.6773 |
| 8 | | 0.6833 | 0.7866 | 0.9018 | 0.8789 | 0.7915 | 0.7876 | 0.7539 | 0.7463 | 0.6977 | 0.6882 |
| 9 | | 0.6549 | 0.7677 | 0.9009 | 0.8652 | 0.7867 | 0.7827 | 0.7480 | 0.7403 | 0.6948 | 0.6824 |
| 10 | | 0.6548 | 0.7664 | 0.9018 | 0.8630 | 0.7898 | 0.7839 | 0.7505 | 0.7418 | 0.6965 | 0.6846 |

La primera fila es - debido a que para hacer el agregado como mínimo se necesitan 2 contratos. Se aprecia que desde Nc = 2 a Nc = 10 el error tiene un máximo en P=3, a partir de P=3 empieza a descender progresivamente. Aunque no mejora demasiado, se puede decir que a mayor Nc menor

error medio por lo que se justifica realizar un agregado de contratos. Se va a representar gráficamente la tabla anterior y el consumo real vs el aproximado para que se vea visualmente si aproxima bien, la figura 4-6 está realizada para Nc=10.

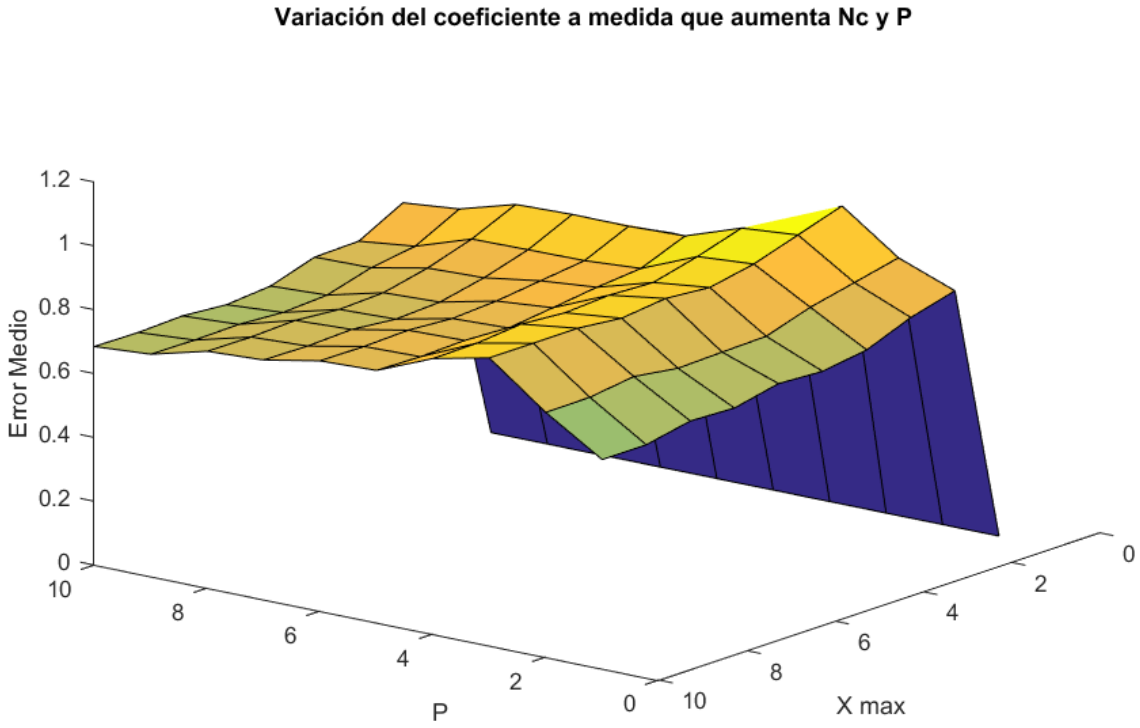



Figura 4-3. Representación Gráfica del error para Nc y P variables, mínimos cuadrados segundo regresor.

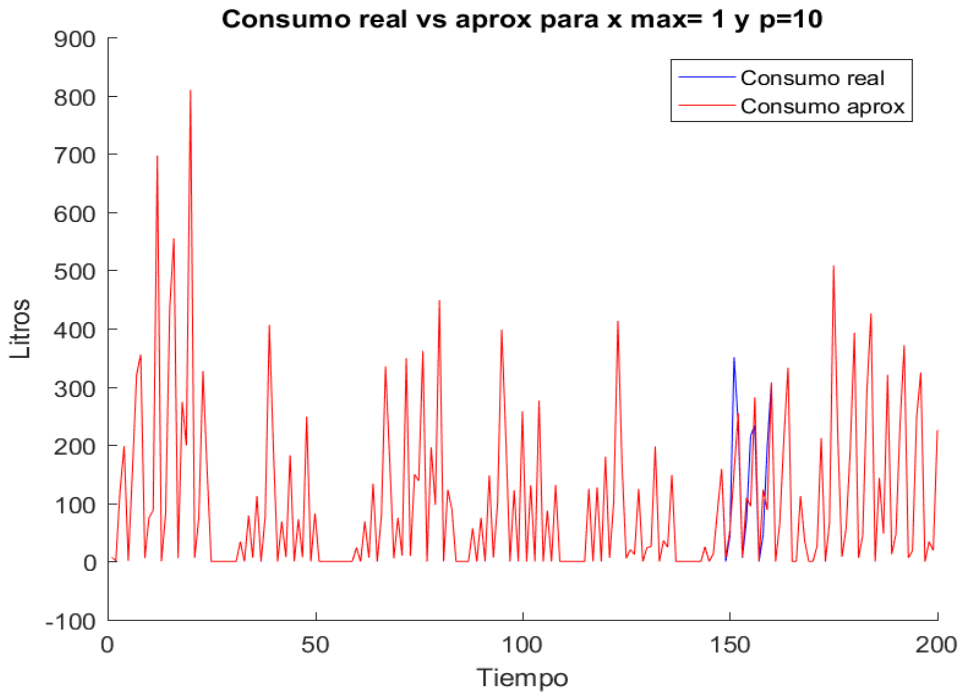



Figura 4-4. Comparación entre consumo real y consumo aproximado para Nc=10, X max=1 y P=10, mínimos cuadrados segundo regresor.

Los resultados obtenidos con esta aproximación no se pueden comparar con los obtenidos en el capítulo 3 ya que en este caso se ha variado Nc y se ha dejado fijo X max, mientras que en la primera aproximación lo que se variaba era X max por lo que se tratan de aproximaciones distintas. En el siguiente apartado se variará también X max por lo que el error obtenido sí se podrá comparar con el error "base" obtenido en el capítulo anterior.

### 4.3.2 Aproximación para un Nc fijo variando X max.

La siguiente aproximación se va a realizar dejando fijo Nc y variando X max y P. Para intentar obtener resultados representativos se han realizado las mismas aproximaciones para tres casos distintos, para Nc=2, para Nc=5 y para Nc=10. Al igual que en todos los experimentos anteriores, N=120, X max y P van de 1 a 10.

El objetivo de este experimento es ver como varía el error en cada caso y ver si Nc tiene un efecto directo en el error cuando se varía X max.

- Resultados obtenidos para Nc = 2.

Tabla 4-4. Evolución del error, mínimos cuadrados segundo regresor Nc=2.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.8145 | 0.8817 | 1.0083 | 0.9111 | 0.8101 | 0.8147 | 0.7530 | 0.7326 | 0.6874 | 0.6713 |
| 2 | | 0.4335 | 0.5970 | 0.5070 | 0.4747 | 0.5183 | 0.5300 | 0.4854 | 0.4927 | 0.5081 | 0.5260 |
| 3 | | 0.4485 | 0.4740 | 0.3967 | 0.4452 | 0.4953 | 0.4912 | 0.4504 | 0.4632 | 0.4726 | 0.5162 |
| 4 | | 0.3871 | 0.4342 | 0.3821 | 0.4227 | 0.4675 | 0.4823 | 0.4631 | 0.4815 | 0.4751 | 0.4908 |
| 5 | | 0.4100 | 0.4090 | 0.3619 | 0.3846 | 0.4453 | 0.4876 | 0.4586 | 0.4633 | 0.4653 | 0.4786 |
| 6 | | 0.4148 | 0.4297 | 0.4151 | 0.4463 | 0.4454 | 0.4756 | 0.4441 | 0.4730 | 0.4719 | 0.4786 |
| 7 | | 0.4348 | 0.4140 | 0.3971 | 0.4043 | 0.4030 | 0.4584 | 0.4427 | 0.4545 | 0.4528 | 0.4604 |
| 8 | | 0.3141 | 0.3270 | 0.3028 | 0.3040 | 0.3132 | 0.3716 | 0.3523 | 0.3567 | 0.3523 | 0.3729 |
| 9 | | 0.3423 | 0.3569 | 0.3025 | 0.3252 | 0.3406 | 0.3774 | 0.3629 | 0.3669 | 0.3728 | 0.3803 |
| 10 | | 0.2775 | 0.3579 | 0.3027 | 0.3130 | 0.3227 | 0.3648 | 0.3499 | 0.3532 | 0.3621 | 0.3918 |

Comparando las celdas de X max =1 y X max= 10 en la tabla 4-4, se observa una reducción del error de más del 50%. A continuación, se representará el consumo real y el consumo aproximado para el caso de X max= 10 y P= 10.

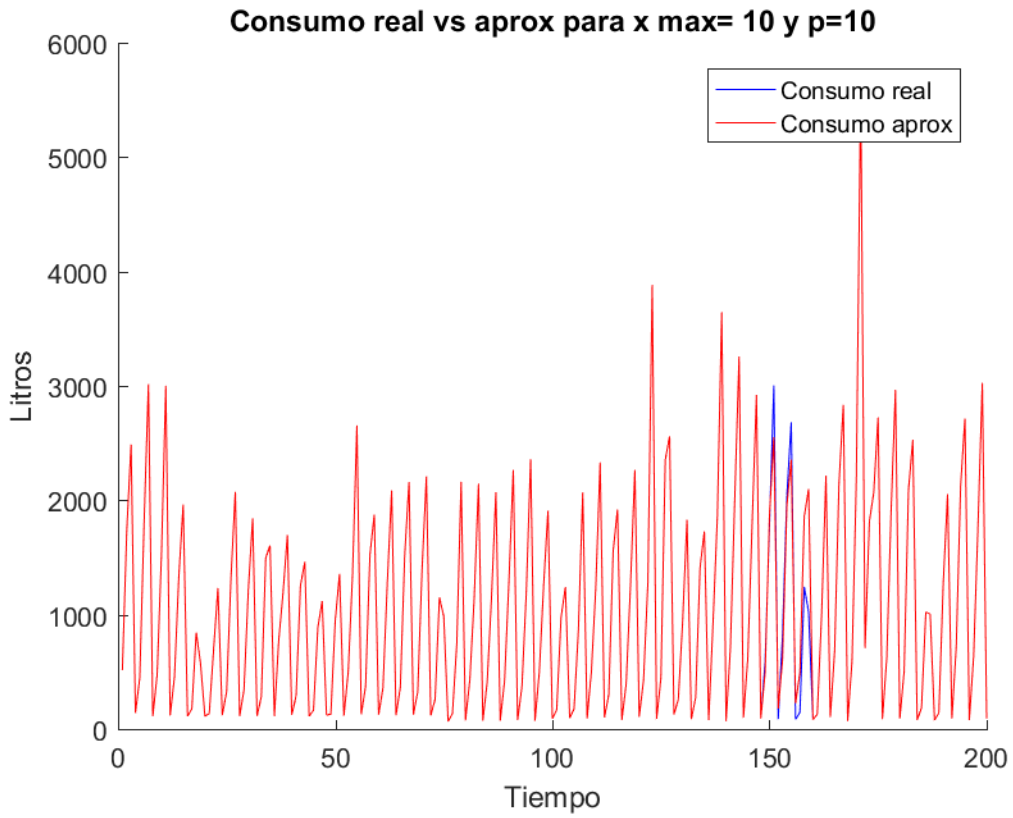



Figura 4-5. Comparación entre consumo real y consumo aproximado para Nc=2, X max=10 y P=10, mínimos cuadrados segundo regresor.

- Resultados para Nc= 5:

Tabla 4-5. Evolución del error, mínimos cuadrados segundo regresor Nc=5.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.7005 | 0.7743 | 0.8911 | 0.8709 | 0.8199 | 0.8039 | 0.7774 | 0.7718 | 0.7369 | 0.7341 |
| 2 | | 0.5139 | 0.6901 | 0.5836 | 0.5673 | 0.6182 | 0.6590 | 0.6112 | 0.6189 | 0.6303 | 0.6499 |
| 3 | | 0.5402 | 0.5358 | 0.4644 | 0.5215 | 0.5849 | 0.5598 | 0.5118 | 0.5201 | 0.5311 | 0.5497 |
| 4 | | 0.4936 | 0.5124 | 0.4624 | 0.4846 | 0.5275 | 0.5119 | 0.4783 | 0.4786 | 0.4805 | 0.4952 |
| 5 | | 0.4290 | 0.4202 | 0.3949 | 0.4108 | 0.4709 | 0.4873 | 0.4484 | 0.4507 | 0.4455 | 0.4510 |
| 6 | | 0.4388 | 0.4524 | 0.4200 | 0.4437 | 0.4318 | 0.4709 | 0.4361 | 0.4555 | 0.4520 | 0.4541 |
| 7 | | 0.4327 | 0.4359 | 0.4205 | 0.4392 | 0.4233 | 0.4592 | 0.4416 | 0.4425 | 0.4358 | 0.4423 |
| 8 | | 0.2559 | 0.2859 | 0.2733 | 0.2988 | 0.2918 | 0.3528 | 0.3330 | 0.3331 | 0.3282 | 0.3386 |
| 9 | | 0.3540 | 0.3517 | 0.3004 | 0.3323 | 0.3389 | 0.3792 | 0.3636 | 0.3637 | 0.3707 | 0.3680 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 10 | 0.3118 | 0.3644 | 0.3145 | 0.3343 | 0.3351 | 0.3650 | 0.3467 | 0.3581 | 0.3615 | 0.3748 |

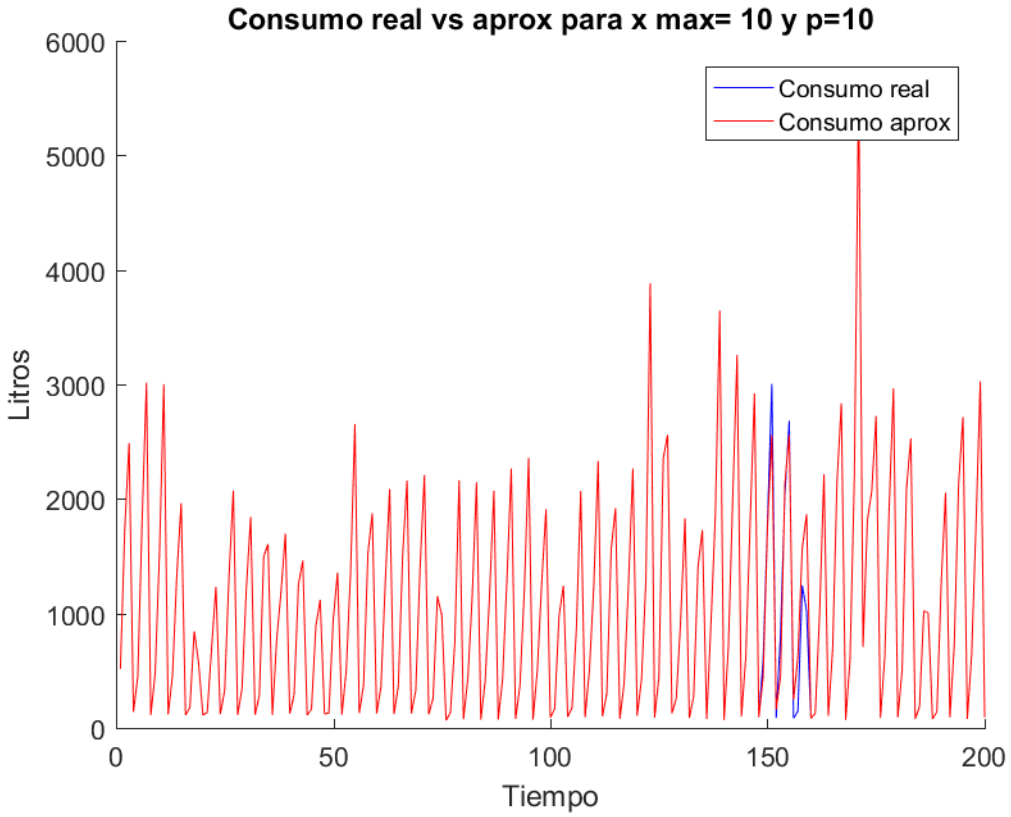



Figura 4-6. Comparación entre consumo real y consumo aproximado para Nc=5, X max=10 y P=10, mínimos cuadrados segundo regresor.

- Resultados para Nc=10:

Tabla 4-6. Evolución del error, mínimos cuadrados segundo regresor Nc=10.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.6631 | 0.7701 | 0.9050 | 0.8657 | 0.7963 | 0.7891 | 0.7589 | 0.7499 | 0.7044 | 0.6935 |
| 2 | | 0.4742 | 0.7077 | 0.5755 | 0.5520 | 0.6055 | 0.6599 | 0.6108 | 0.6023 | 0.6018 | 0.6406 |
| 3 | | 0.4361 | 0.4831 | 0.4750 | 0.4459 | 0.5237 | 0.5132 | 0.4698 | 0.4638 | 0.4865 | 0.4850 |
| 4 | | 0.4370 | 0.4614 | 0.4322 | 0.4511 | 0.4970 | 0.4813 | 0.4540 | 0.4316 | 0.4264 | 0.4286 |
| 5 | | 0.4298 | 0.4407 | 0.4246 | 0.4377 | 0.4970 | 0.5173 | 0.4847 | 0.4701 | 0.4665 | 0.4676 |
| 6 | | 0.4416 | 0.4692 | 0.4453 | 0.4444 | 0.4184 | 0.4387 | 0.4172 | 0.4345 | 0.4226 | 0.4213 |
| 7 | | 0.3732 | 0.3844 | 0.3908 | 0.3964 | 0.3784 | 0.4071 | 0.3992 | 0.3900 | 0.3833 | 0.3777 |
| 8 | | 0.2759 | 0.3124 | 0.2804 | 0.2944 | 0.2842 | 0.3309 | 0.3081 | 0.3059 | 0.3024 | 0.3063 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 0.2776 | 0.2996 | 0.2566 | 0.2870 | 0.2873 | 0.3215 | 0.3149 | 0.3129 | 0.3117 | 0.3155 |
| 10 | 0.3038 | 0.3578 | 0.3125 | 0.3260 | 0.3192 | 0.3569 | 0.3370 | 0.3428 | 0.3352 | 0.3619 |

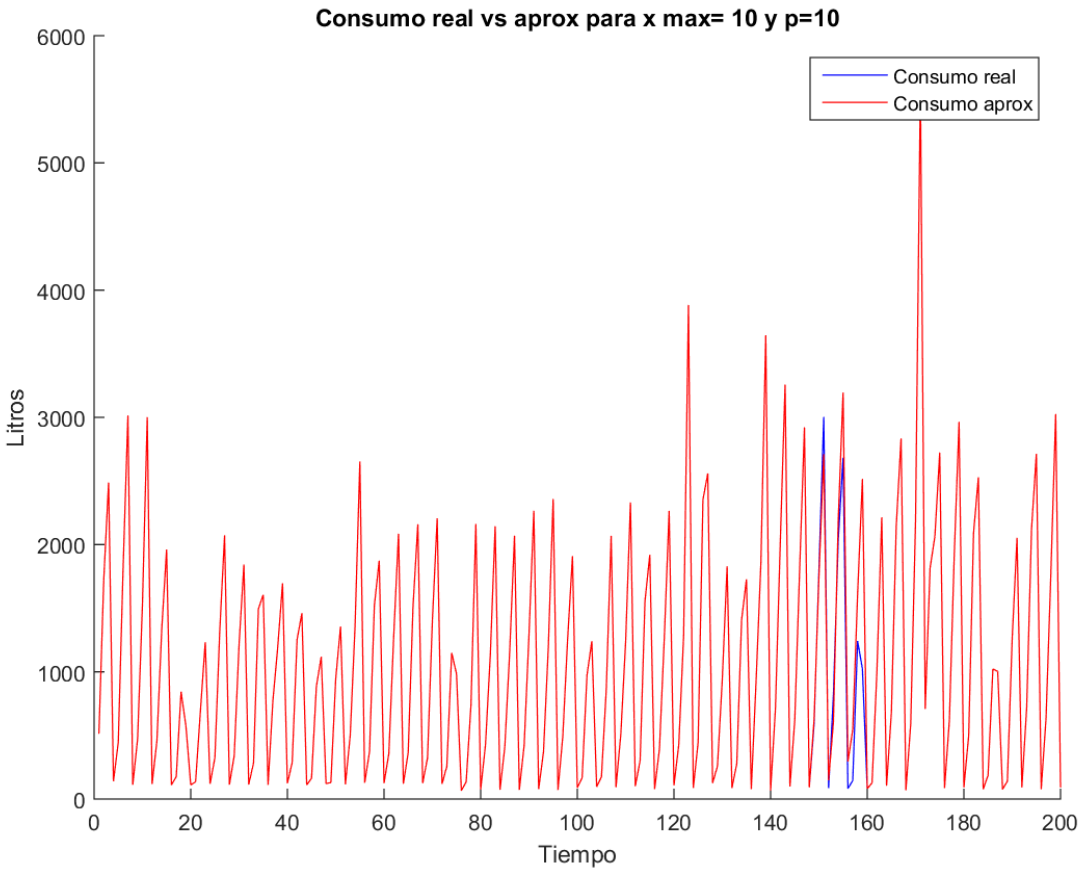



Figura 4-7. Comparación entre consumo real y consumo aproximado para Nc=10, X max=10 y P=10, mínimos cuadrados segundo regresor.

Al ser las tablas tan similares, para ver mejor cada uno de los casos, se han representado las tres en una misma imagen (figura 4-8). Se puede observar que de forma general, a medida que X max aumenta, el error decrece rápidamente. Los casos de Nc=2 y Nc=5 no son tan buenos como el caso de Nc=10 ya que en la gráfica se observa que en muchas zonas está por encima el color rojo (Nc=2) y el verde (Nc=5) lo que significan que cometen un mayor error. Por lo tanto, el caso más favorable es el de Nc=10 ya que sólo se produce el error más grande para X max=5.

Si se comparan los valores obtenidos en la tabla de Nc=10 (tabla 4-6) con los obtenidos en la tabla 3-2, se observa que ya se empieza a acercar, aunque todavía no se mejora. Por este motivo, en el siguiente apartado se va a introducir la corrección teniendo en cuenta el consumo total del intervalo.

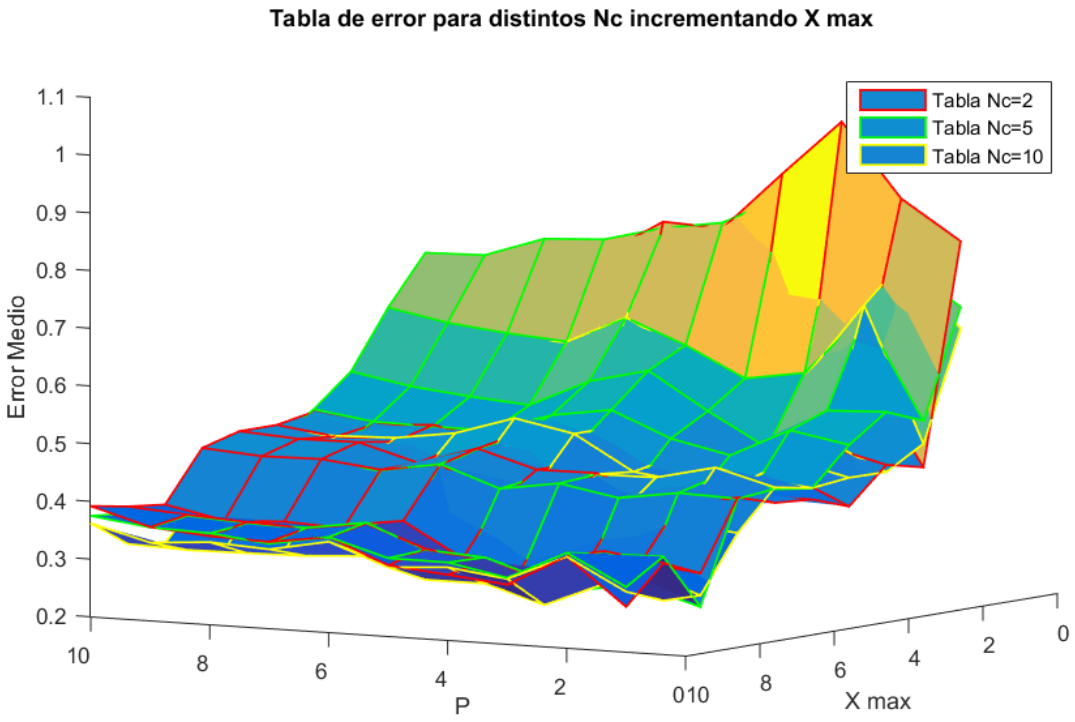



Figura 4-8. Evolución del error, mínimos cuadrados segundo regresor con Nc fija y X max variable.

### 4.3.3 Aproximación introduciendo consumo total del intervalo para Nc fijo y variando X max.

Como el error más bajo se produce para Nc=10, este apartado se centrará en dicho caso. El coeficiente α se obtiene de la misma forma que en el apartado 3, esta es la tabla de errores que se obtiene:

Tabla 4-7. Evolución del error, mínimos cuadrados segundo regresor variando X max y P, con coeficiente α.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.3622 | 0.4236 | 0.5622 | 0.6393 | 0.6831 | 0.6694 | 0.7132 | 0.6933 | 0.6580 | 0.6374 |
| 2 | | 0.2950 | 0.4505 | 0.4029 | 0.4190 | 0.5054 | 0.4974 | 0.4753 | 0.4746 | 0.4634 | 0.4730 |
| 3 | | 0.4226 | 0.4034 | 0.3520 | 0.4300 | 0.4877 | 0.4858 | 0.4437 | 0.4437 | 0.4682 | 0.4765 |
| 4 | | 0.2861 | 0.3213 | 0.2939 | 0.3905 | 0.4214 | 0.4360 | 0.4079 | 0.4119 | 0.4005 | 0.4120 |
| 5 | | 0.2430 | 0.2986 | 0.2731 | 0.3516 | 0.4171 | 0.4686 | 0.4320 | 0.4310 | 0.4190 | 0.4287 |
| 6 | | 0.1940 | 0.3357 | 0.3015 | 0.3870 | 0.3585 | 0.3816 | 0.3663 | 0.3682 | 0.3484 | 0.3707 |
| 7 | | 0.2373 | 0.2605 | 0.2849 | 0.3234 | 0.3006 | 0.3226 | 0.3293 | 0.3410 | 0.3252 | 0.3294 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 8 | 0.1862 | 0.2874 | 0.2516 | 0.2544 | 0.2441 | 0.2866 | 0.2637 | 0.2556 | 0.2508 | 0.2630 |
| 9 | 0.2056 | 0.2102 | 0.2205 | 0.2764 | 0.2783 | 0.2985 | 0.2919 | 0.2906 | 0.2780 | 0.2827 |
| 10 | 0.2159 | 0.2422 | 0.2209 | 0.2406 | 0.2548 | 0.2684 | 0.2655 | 0.2680 | 0.2625 | 0.2950 |

Realizando una comparación gráfica con el caso sin corrección:

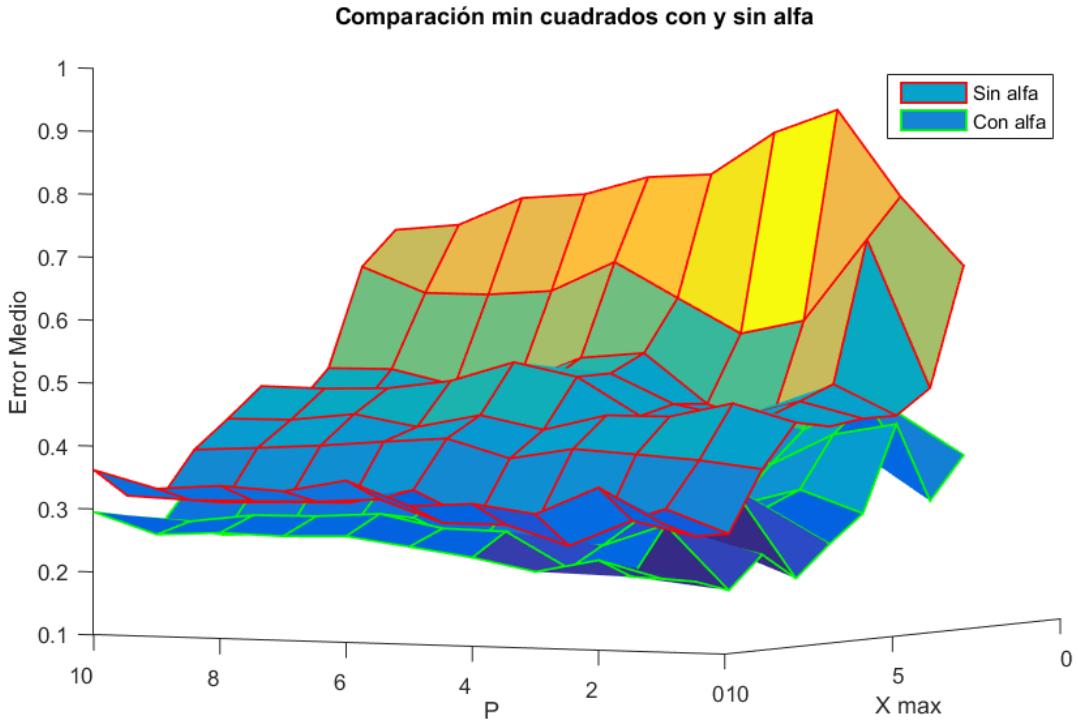



Figura 4-9. Comparación de errores, mínimos cuadrados segundo regresor para Nc=10 con y sin coeficiente α.

Se observa que se ha conseguido una mejoría muy grande respecto al caso anterior ya que el plano del coeficiente está por debajo del sin coeficiente, la última mejora que se va a realizar a este algoritmo es la introducción de un factor de olvido.

### 4.3.4 Aproximación introduciendo alfa y factor de olvido para Nc fijo y variando X max.

Para meter el factor de olvido hay que usar la teoría del apartado 4.1.2. Después de hacer pruebas para elegir un λ apropiado se va a elegir λ=0.9969 ya que cuando λ toma valores alejados de 0.9 el regresor aproxima peor porque olvida demasiados datos. En Matlab, el regresor con el factor de olvido incluido queda:

```
z=5;
M=zeros(length(medidas),9);
Y=zeros(1,length(medidas));
for j=(i-medidas):(i-1)        % El bucle va hasta i-1 porque la última
medida que se quiere tener en cuenta es la anterior a la actual.
      M(z-4,1)=ind(j-1,2);
      M(z-4,2)=ind(j-2,2);
      M(z-4,3)=ind(j-3,2);
      M(z-4,4)=ind(j-4,2);
      M(z-4,5)=ag(j,2);
      M(z-4,6)=ag(j-1,2);
      M(z-4,7)=ag(j-2,2);
      M(z-4,8)=ag(j-3,2);
      M(z-4,9)=ag(j-4,2);
      Y(z-4)=consumo_x_aisl(j,2);%Hay que modificar el segundo indice.
      z=z+1;
end

% Se calcula el factor de olvido:

W=zeros(length(M(1,:)));
landa=0.9969;
N=length(M(:,1));
K=length(M(:,1))-1;

for j=1:N
      W(j,j)=landa^(N-K);
      K=K-1;
end

teta = (M'*W*M)\(M'*W*Y');

Y_final=[ind(i-1,2),ind(i-2,2),ind(i-3,2),ind(i-4,2),ag(i,2),
ag(i-1,2), ag(i-2,2),ag(i-3,2),ag(i-4,2)]*teta;
```

Con esta mejora la tabla del error queda:

Tabla 4-8. Evolución del error, minimos cuadrados segundo regresor variando X max y P, con coeficiente α y factor de olvido.

| Xmax | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.3264 | 0.4791 | 0.5556 | 0.6323 | 0.6809 | 0.6758 | 0.7102 | 0.6817 | 0.6452 | 0.6269 |
| 2 | | 0.3022 | 0.4972 | 0.4130 | 0.4368 | 0.5199 | 0.4996 | 0.4735 | 0.4796 | 0.4569 | 0.4732 |
| 3 | | 0.4200 | 0.4279 | 0.3681 | 0.4516 | 0.4943 | 0.4945 | 0.4447 | 0.4577 | 0.4742 | 0.4892 |
| 4 | | 0.3116 | 0.3465 | 0.3174 | 0.4057 | 0.4260 | 0.4410 | 0.4122 | 0.4267 | 0.4106 | 0.4229 |
| 5 | | 0.2458 | 0.3022 | 0.2837 | 0.3484 | 0.4103 | 0.4629 | 0.4250 | 0.4310 | 0.4168 | 0.4275 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 6 | 0.2272 | 0.3688 | 0.3066 | 0.3929 | 0.3673 | 0.3897 | 0.3760 | 0.3835 | 0.3636 | 0.3916 |
| 7 | 0.2431 | 0.2664 | 0.2793 | 0.3234 | 0.3000 | 0.3170 | 0.3237 | 0.3289 | 0.3159 | 0.3264 |
| 8 | 0.1897 | 0.3074 | 0.2659 | 0.2637 | 0.2494 | 0.2911 | 0.2684 | 0.2596 | 0.2557 | 0.2694 |
| 9 | 0.1927 | 0.1956 | 0.2089 | 0.2673 | 0.2674 | 0.2874 | 0.2862 | 0.2870 | 0.2782 | 0.2849 |
| 10 | 0.1705 | 0.2232 | 0.2005 | 0.2320 | 0.2612 | 0.2668 | 0.2696 | 0.2642 | 0.2653 | 0.2987 |

Igual que en el apartado anterior se va a realizar una gráfica comparativa entre esta tabla y los resultados obtenidos para el caso en que sólo se tenía en cuenta el coeficiente α:

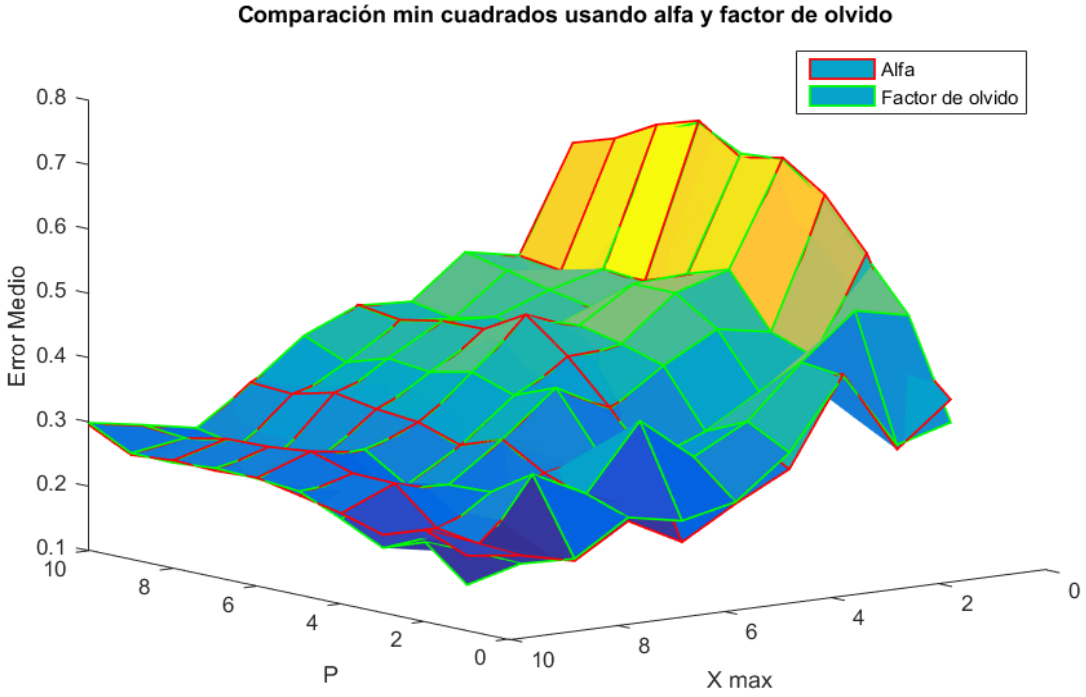



Figura 4-10. Comparación de errores, mínimos cuadrados segundo regresor para caso Nc=10 con alfa y con factor de olvido + alfa.

Con este factor de olvido determinado, hay zonas en que se aproxima mejor y otras en que se hace peor. Además, si se produce mejora no es significativa. Se podría buscar el factor de olvido óptimo para obtener el mejor resultado posible, pero el programa requeriría un tiempo de ejecución elevado por lo que se descarta esa idea. Como no se produce una mejoría clara, se va a tomar la tabla 4-7 como el error final para este algoritmo y por tanto, se va a comparar con los mínimos cuadrados usando sólo los datos anteriores (tabla 4-2) y con la aproximación inicial (tabla 3-2).

Al comparar los resultados de este algoritmo con los del primer regresor se obtiene:

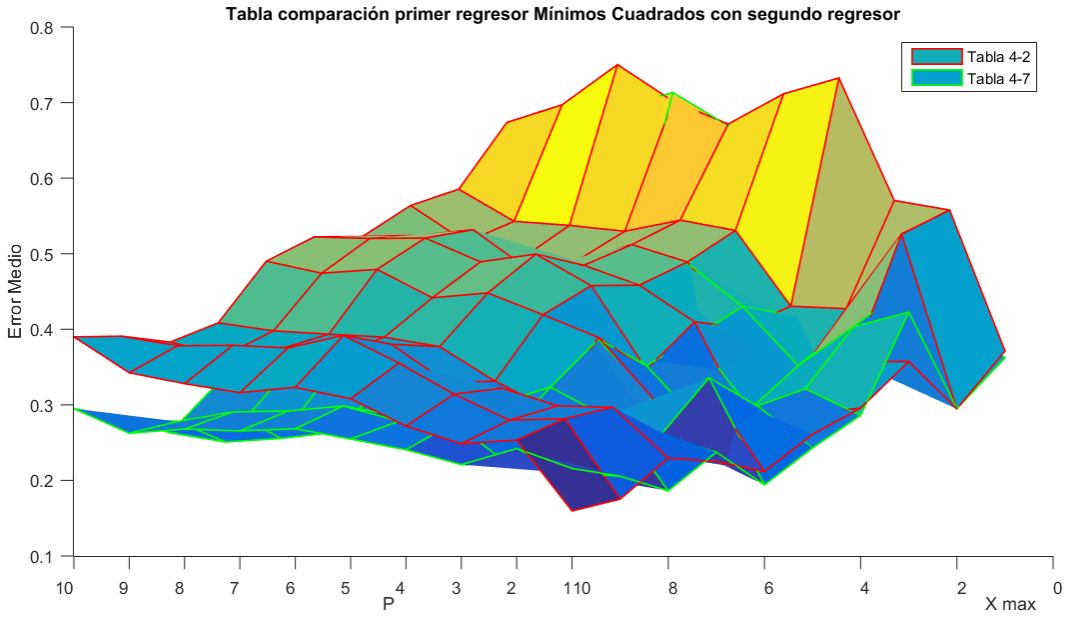



Figura 4-11. 1ª Comparación de resultados obtenidos para los dos regresores de mínimos cuadrados.

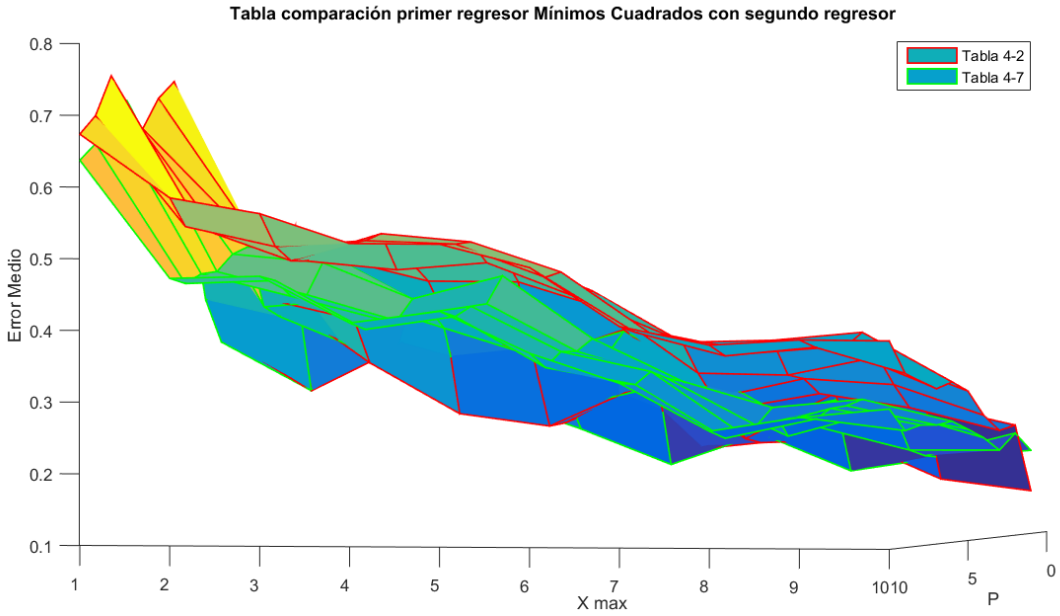



Figura 4-12. 2ª Comparación de resultados obtenidos para los dos regresores de mínimos cuadrados.

Como se observa se ha conseguido mejorar la tabla del error obtenido. El segundo regresor funciona mucho mejor que el primero cuando P va aumentando ya que tiene en cuenta los contratos agregados en el mismo intervalo (dispone de más información) y aunque existan algunas zonas donde el primer regresor aproxima mejor, en el cómputo global sale como ganador el segundo regresor.

Ahora hay que comparar los resultados del segundo regresor con los mejores resultados obtenidos hasta ahora, los de la tabla 3-2. Si se representan tanto la tabla 3-2 como la tabla 4-7 en la misma

gráfica se obtiene:

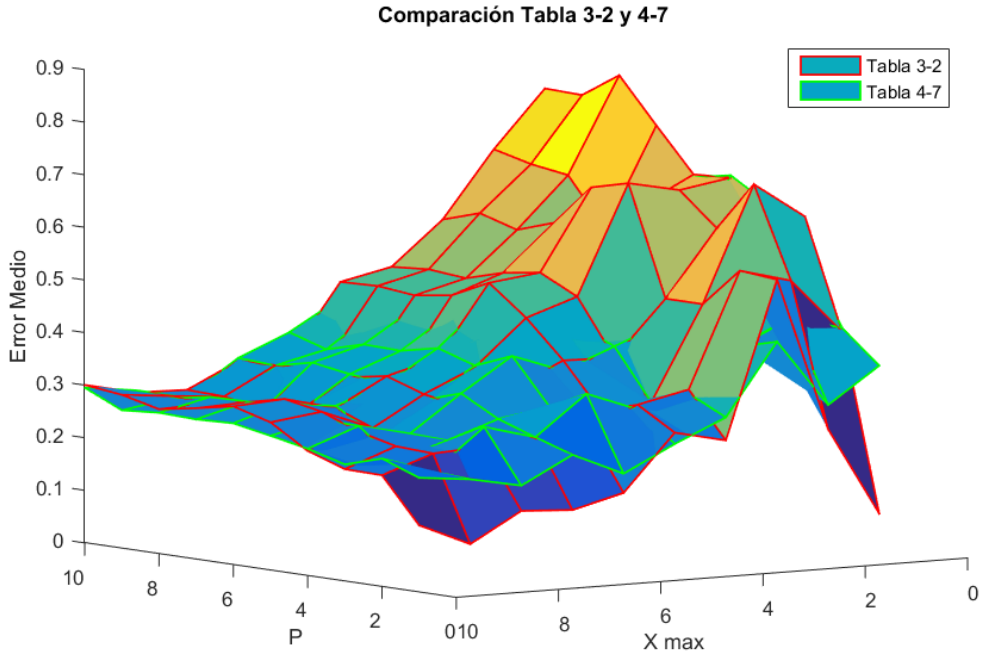



Figura 4-13. 1ª Comparación entre el error obtenido para aproximación inicial con coeficiente alfa (tabla 3-2) y mínimos cuadrados segundo regresor con coeficiente alfa (tabla 4-7).

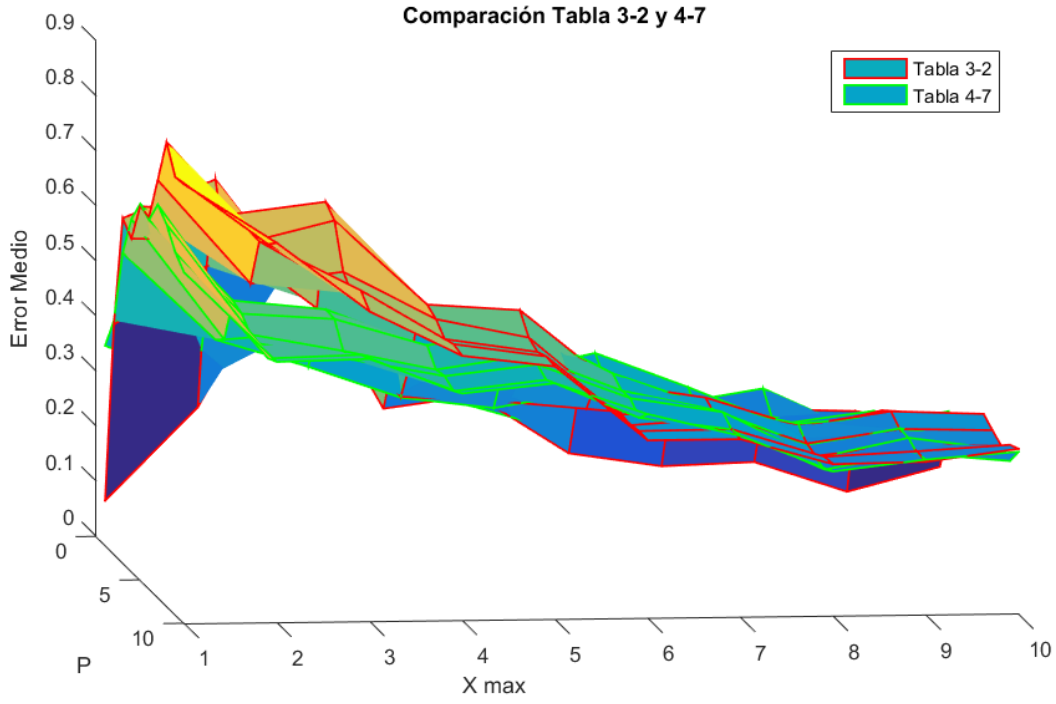



Figura 4-14. 2ª Comparación entre el error obtenido para aproximación inicial con coeficiente alfa (tabla 3-2) y mínimos cuadrados segundo regresor con coeficiente alfa (tabla 4-7).

Se puede apreciar que para el algoritmo de los mínimos cuadrados se comete un error menor en la mayoría de los casos (ya que la gráfica roja es la que está por encima) pero cuando P es 1-2 se obtiene en casi todo el tramo mejor resultado con la aproximación inicial aunque, como se puede observar en la figura 4-14, a medida que P aumenta se invierten la situación. Sacrificando esa mejor aproximación para P bajos, se ha decidido elegir como mejor error obtenido el de los mínimos cuadrados ya que en general aproxima mejor.

La siguiente tabla se ha realizado teniendo en cuenta 100 aproximaciones, en ella se muestra la evolución del error para los algoritmos realizados hasta el momento:

Tabla 4-9. Comparación de errores para 100 aproximaciones, Mínimos Cuadrados.

| N Contratos / Error (%) | 1 | 3 | 5 | 10 |
|---|---|---|---|---|
| Aproximación Inicial a | 41.7 | 34.2 | 25.1 | 17.4 |
| Mínimos Cuadrados | 49.2 | 40.1 | 32.8 | 24.9 |
| Mínimos Cuadrados a | 45.3 | 33.3 | 23.2 | 19.2 |

En el siguiente apartado se va a utilizar otro algoritmo más complejo para intentar mejorar los resultados obtenidos, dicho algoritmo es el de los Métodos Gaussianos.

# 5 MÉTODOS GAUSSIANOS

## 5.1 Introducción.

Con el objetivo de hacer frente a la incertidumbre, se recurre al uso de variables aleatorias. Se usan distribuciones y funciones de densidad para caracterizar, al menos desde un punto de vista probabilístico, las variables aleatorias que aparecen en el contexto de inferencia estadística. Para seguir el algoritmo que se va a usar en este apartado hacen falta ciertos conocimientos previos que se van a exponer a continuación.

Dada una variable aleatoria *w*, se dice que $F_w$: $\mathbb{R} \rightarrow [0,1]$ es su función de distribución si y solo si:

$$\Pr\{w \leq a\} = F_w(a), \forall a \in \mathbb{R}$$

Además, se dice que una variable aleatoria *w* es continua si su función de distribución es continua. La función de densidad de probabilidad de *w* es la función $f_w : \mathbb{R} \rightarrow [0, \infty)$ que satisface

$$\Pr\{a \leq w \leq b\} = \int_a^b f_w(w)dw, \qquad \forall a, \forall b.$$

La función de densidad de probabilidad sirve para definir la media $\mathrm{E}\{w\}$

$$E\{w\} = \int_{-\infty}^{\infty} w f_w(w)dw.$$

También permite obtener la media de una función de una variable aleatoria *w*. Esto es, si se denota $y = g(w)$, es obvio que *y* es una variable aleatoria. Se tiene que

$$E\{y\} = E\{g(w)\} = \int_{-\infty}^{\infty} g(w) f_w(w)dw.$$

Por otro lado, se supone que $\mu_w = \mathrm{E}\{w\}$. Entonces, se le llama varianza $\sigma_w^2$ de *w:*

$$\sigma_w^2 = E\{(w - \mu_w)^2\} = \int_{-\infty}^{\infty} (w - \mu_w)^2 f_w(w)dw.$$

A la raíz cuadrada se le llama desviación estándar de *w*.

$$\sigma_w = \sqrt{E\{(w - \mu_w)^2\}}$$

En el contexto de sistemas de control, es común el caso de que la variable aleatoria es dada como una serie temporal, la cual es una serie de puntos de datos indexados en el tiempo.

## 5.2 Distribución Normal.

La Distribución Normal (Gaussiana), es una de las funciones de densidad de probabilidad que se usan más frecuentemente en los problemas de ingeniería. Se dice que la variable aleatoria *w* tiene una función de densidad de probabilidad normal si:

$$f_w(w) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right)$$

Como se puede observar, la distribución normal es una exponencial multiplicada por un factor normalizador $\frac{1}{\sigma\sqrt{2\pi}}$. Como se muestra en la siguiente fórmula, dicho factor normalizador es elegido para forzar

$$\Pr\{-\infty < w < \infty\} = \int_{-\infty}^{\infty} f_w(w)dw = 1.$$

Se va a denotar

$$Z = \int_{-\infty}^{\infty} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right) dw.$$

Como la integración va desde -∞ hasta ∞, si se hace el cambio de variable $\widehat{w} = w - \mu$, $dw = d\widehat{w}$ se demuestra que la integral no depende del parámetro $\mu$. Esto es,

$$Z = \int_{-\infty}^{\infty} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right) dw = \int_{-\infty}^{\infty} \exp\left(\frac{-\widehat{w}^2}{2\sigma^2}\right) d\widehat{w}\,, \forall \mu \in \mathbb{R}$$

En vez de tratar directamente con el valor de Z, es más conveniente usar el valor de $Z^2$. Esto es lo que se hace en el siguiente desarrollo

$$Z^2 = \left(\int_{-\infty}^{\infty} \exp\left(\frac{-\widehat{w}^2}{2\sigma^2}\right) d\widehat{w}\right)^2 = \left(\int_{-\infty}^{\infty} \exp\left(\frac{-x^2}{2\sigma^2}\right) dx\right)\left(\int_{-\infty}^{\infty} \exp\left(\frac{-y^2}{2\sigma^2}\right) dy\right)$$
$$= \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} \exp\left(\frac{-x^2-y^2}{2\sigma^2}\right) dxdy.$$

$Z^2$ puede obtenerse por medio de una doble integral en un sistema rectangular de coordenadas 2D. La integral se va a resolver usando un sistema de coordenadas polar. Esto es, $x = r\cos(\theta)$, $y = r\sin(\theta)$. En dicho sistema de coordenadas, el área infinitesimal dxdy se escribe $rdrd\theta$. Además,

$$Z^2 = \int_0^{2\pi}\int_0^{\infty} \exp\left(\frac{-r^2}{2\sigma^2}\right) rdrd\theta$$

Resolviendo se llega a que $Z^2 = 2\pi\sigma^2 \rightarrow Z = \sigma\sqrt{2\pi}$. Además, el factor normalizador requerido para la función de densidad de probabilidad es, de hecho, $\frac{1}{Z} = \frac{1}{\sigma\sqrt{2\pi}}$.

Suponiendo que *w* tenga una distribución de densidad de probabilidad normal de parámetros $\mu$ y $\sigma$. Se va a demostrar que $\mu$ y $\sigma^2$ son iguales a la media y la varianza de *w* respectivamente. Debido a la simetría de la función normal de densidad de probabilidad respecto a $\mu$ se infiere que el valor de la media es dado por $\mu$. Por otro lado, si partimos del valor de Z y diferenciamos respecto $\sigma$:

$$Z = \sigma\sqrt{2\pi} = \int_{-\infty}^{\infty} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right) dw \rightarrow \sqrt{2\pi} = \int_{-\infty}^{\infty} \frac{(w-\mu)^2}{\sigma^3} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right) dw$$

Multiplicando ambos términos por $\frac{\sigma^2}{\sqrt{2\pi}}$ y aislando $\sigma^2$ en el término de la izquierda:

$$\sigma^2 = \int_{-\infty}^{\infty} \frac{(w-\mu)^2}{\sigma\sqrt{2\pi}} \exp\left(\frac{-(w-\mu)^2}{2\sigma^2}\right) dw = \int_{-\infty}^{\infty} (w-\mu)^2 f_w(w) dw = E\{(w-\mu)^2\}$$

Se llega a la conclusión de que $\sigma^2$ es igual a la varianza de la variable aleatoria normal *w*. Si *w* es generada por una función de densidad de probabilidad gaussiana de media $\mu$ y varianza $\sigma^2$, se puede obtener mediante integración numérica:

$$Pr\{\mu - \sigma \leq w \leq \mu + \sigma\} \approx 0.6827$$
$$Pr\{\mu - 2\sigma \leq w \leq \mu + 2\sigma\} \approx 0.9545$$
$$Pr\{\mu - 3\sigma \leq w \leq \mu + 3\sigma\} \approx 0.9973$$

Por lo tanto, existe una alta probabilidad de que *w* pertenezca al intervalo $[\mu - 3\sigma \leq w \leq \mu + 3\sigma]$. Este resultado permite introducir el concepto de intervalo de confianza. En la siguiente figura, se muestran distintos intervalos de confianza para una función de densidad de probabilidad gaussiana.

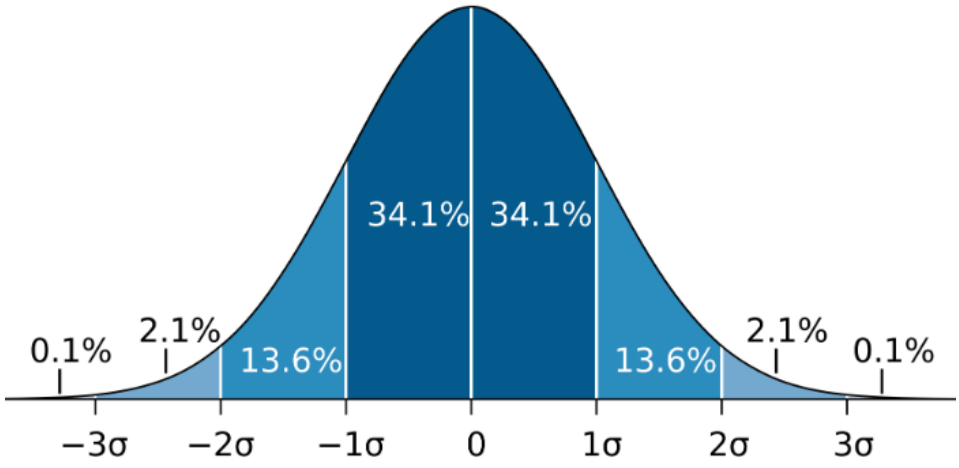



Figura 5-1. Intervalos de confianza para una distribución Normal (o Gaussiana).

# 5.3 Modelado de la incertidumbre en mayores dimensiones.

## 5.3.1 Introducción.

En este subapartado se va a considerar el caso en el que *w* es un vector aleatorio de *n* componentes.

$$w = [w_1 \quad w_2 \quad \dots \quad w_n]^T$$

Se asume que *w* tiene una función de densidad de probabilidad $f_w(\cdot)$ que nos permita, dado cualquier set $D \subseteq \mathbb{R}^n$, determinar cual es la probabilidad $Pr\{w \in D\}$.

$$Pr\{w \in D\} = \int_D f_w(w)dw$$

En general, realizar esta integral es difícil en altas dimensiones, hacen falta técnicas específicas como la integración numérica Montecarlo. Un caso particular relevante es cuando los diferentes componentes de *w* son mutuamente estadísticamente independientes, esto es, cuando la función de densidad de probabilidad puede ser reescrita como

$$f_w(w) = \left(\begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}\right) = \prod_{i=1}^{n} f_{w_i}(w_i)$$

Asumiendo que son independientes, y si el set D es un híper-rectángulo ($D = \{w : w_i^- \leq w_i \leq w_i^+, i = 1, \dots, n\}$) se tiene

$$Pr\{w \in D\} = \prod_{i=1}^{n} Pr\{w_i^- \leq w_i \leq w_i^+\} = \prod_{i=1}^{n} \int_{w_i^-}^{w_i^+} f_{w_i}(w_i)dw_i$$

Lo que puede ser calculado con integración numérica de *n* integrales mono dimensionales.

## 5.3.2 Media y Covarianza de un vector aleatorio.

Las nociones de media y varianza para el caso unidimensional para variables aleatorias tienen una generalización directa para el caso multidimensional. Dado el vector aleatorio *w*, con la función de densidad de probabilidad $f_w(\cdot)$ se define la media $\mu_w$ y la covarianza $\sum_w$ como:

$$\mu_w = E\{w\} = \int w f_w(w)dw$$

$$\Sigma_w = E\{(w - \mu_w)(w - \mu_w)^T\} = \int (w - \mu_w)(w - \mu_w)^T f_w(w)dw$$

Hay que destacar que la media $\mu_w$ es un vector de dimensión *n* dado por el valor de la media de cada componente de *w*. Esto es,

$$\mu_w = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix} = \begin{bmatrix} E\{w_1\} \\ E\{w_2\} \\ \vdots \\ E\{w_n\} \end{bmatrix}$$

Por otro lado, la matriz de covarianza $\Sigma_w$ es, por construcción, simétrica semidefinida positiva con *n* columnas, donde el (i,j) componente $\Sigma_{i,j}$ de $\Sigma_w$ viene dado por

$$\Sigma_{i,j} = E\{(w_i - \mu_i)(w_j - \mu_j)\}, \qquad i = 1, \dots, n, j = 1, \dots, n.$$

Al escalar $\Sigma_{i,j}$ se le llama, correlación o covarianza de $w_i$ y $w_j$. Se dice que $w_i$ y $w_j$ no están correlados cuando su covarianza es cero.

Ahora se supone que *w* tiene una media $\mu_w$ y covarianza $\Sigma_w$. Se considera el vector aleatorio A*w*, donde A es una matriz dada de dimensiones apropiadas. La media $\mu_{Aw}$ y covarianza $\Sigma_{Aw}$ de quedan:

$$\mu_{Aw} = E\{Aw\} = AE\{w\} = A\mu_w$$

$$\Sigma_{Aw} = E\{(Aw - \mu_{Aw})(Aw - \mu_{Aw})^T\} = E\{A(w - \mu_w)(A(w - \mu_w))^T\}$$
$$= A\left(E\{(w - \mu_w)\left((w - \mu_w)\right)^T\}\right)A^T = A\Sigma_w A^T$$

También se supone que el vector aleatorio v tiene media $\mu_v$ y covarianza $\Sigma_v$ y que está correlado con *w* y tiene las mismas dimensiones que *w*. Entonces la media y la covarianza del vector aleatorio z = w + v vienen dadas por $\mu_w + \mu_v$ y $\Sigma_w + \Sigma_v$ respectivamente.

### 5.3.3 Coeficiente de correlación.

La cantidad

$$\rho_{i,j} = \frac{\Sigma_{i,j}}{\sqrt{\Sigma_{i,i}\Sigma_{j,j}}} = \frac{E\{(w_i - \mu_i)(w_j - \mu_j)\}}{\sqrt{E\{(w_i - \mu_i)^2\}E\left\{\left(w_j - \mu_j\right)^2\right\}}}$$

es llamada coeficiente de correlación entre $w_i$ y $w_j$ y sirve como medida de cómo de fácil es predecir $w_j$ a partir del valor de $w_i$ asumiendo que las medias $\mu_i$ y $\mu_j$ son conocidos.

Si dicho coeficiente vale 1, se produce una correlación perfecta, en cambio, si está entre 0 y 1 hay correlación positiva y si vale 0 no existe ningún tipo de correlación entre ambas variables.

### 5.3.4 Multivariable Gaussiana.

Se supone ahora que cada componente del vector aleatorio

$$w = [w_1 \quad w_2 \quad \dots \quad w_n]^T$$

es independiente de los otros y que la función de densidad de probabilidad de cada componente es una gaussiana con media $\mu_i$ y varianza $\sigma_i^2$. Entonces,

$$f_w(w) = \prod_{i=1}^{n} f_{w_i}(w_i) = \prod_{i=1}^{n} \frac{1}{\sigma_i\sqrt{2\pi}} exp\left(\frac{-(w_i - \mu_i)^2}{2\sigma_i^2}\right)$$

Se va a denotar

$$\mu_w = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_n \end{bmatrix}, \Sigma_{\mathrm{w}} = \begin{bmatrix} \sigma_1^2 & & & \\ & \sigma_2^2 & & \\ & & \ddots & \\ & & & \sigma_n^2 \end{bmatrix}$$

Con esta nomenclatura se tiene:

$$f_w(w) = \frac{1}{(2\pi)^{\frac{n}{2}}\sqrt{\det(\Sigma_{\mathrm{n}})}} exp\left(-\frac{1}{2}(w-\mu_w)^T\Sigma_w^{-1}(w-\mu_w)\right)$$

Una función de densidad de probabilidad multivariable con la estructura dada por la expresión anterior es llamada multivariable gaussiana (también conocida como multivariable normal). En este ejemplo particular, $\Sigma_{\mathrm{w}}$ ha sido elegida diagonal, pero para un caso genérico, $\Sigma_{\mathrm{w}}$ es una matriz simétrica definida positiva. Se puede demostrar que dado $\Sigma_{\mathrm{w}} > 0$ y $\mu_w$, la expresión anterior satisface

(1) Es una función de densidad de probabilidad (su integral sobre $\mathbb{R}^n$ es igual a 1).

(2) Los parámetros $\mu_w$ y $\Sigma_{\mathrm{w}}$ son igual a la media y a la matriz de covarianza del vector aleatorio descrito por dicha función de densidad de probabilidad.

Desde un punto de vista de nomenclatura, se denota como $N(\mu_w, \Sigma_{\mathrm{w}})$ la función de densidad de probabilidad gaussiana multivariable definida por los parámetros $\mu_w$ y $\Sigma_{\mathrm{w}}$.

Suponiendo que *w* tiene una función de densidad de probabilidad dada por $N(\mu_w, \Sigma_{\mathrm{w}})$, se puede demostrar que, para una matriz A, el vector aleatorio A*w* también tiene una multivariable gaussiana como función de densidad de probabilidad. Los parámetros de media y covarianza para este nuevo vector aleatorio pueden obtenerse de los resultados de la sección 5.3.2. Dada la matriz A, A*w* tiene media $A\mu_w$ y covarianza $A\Sigma_{\mathrm{w}}A^{\mathrm{T}}$. Esto significa que la función de densidad de probabilidad gaussiana para A*w* viene dada por $N(A\mu_w, A\Sigma_{\mathrm{w}}A^T)$.

## 5.4 Métodos gaussianos.

Dado un conjunto de puntos ($y_i$, $x_i$), i=1, …, N se quiere obtener una predicción de la salida *y* para *x*. En este caso se podría considerar que el vector $[y \quad y_1 \quad y_2 \quad \cdots \quad y_N]^T$ viene dado por una distribución gaussiana. En el contexto de los procesos gaussianos, se asume que:

$$\begin{bmatrix} y \\ y_D \end{bmatrix} \in N\left(\hat{\mu}\begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix}, \begin{bmatrix} \Sigma_{\mathrm{T,T}} & \Sigma_{D,T}^T \\ \Sigma_{\mathrm{D,T}} & \Sigma_{\mathrm{D,D}} \end{bmatrix}\right)$$

Donde $y_D = [y_1 \quad y_2 \quad \cdots \quad y_N]^T$. En esta configuración, $\hat{\mu}$ es un escalar que modela la media de la variable *y*. Si la media de *y* no es conocida, se puede realizar la siguiente estimación:

$$\hat{\mu} = \frac{1}{N}\sum_{i=1}^{N} y_i$$

Por otro lado, la matriz que modela la covarianza de la función de distribución gaussiana consiste en un término diagonal (que corresponde con la covarianza del ruido en las medidas) y un término simétrico que es dado por una función kernel. Esto es:

$$\begin{bmatrix} \Sigma_{\mathrm{T,T}} & \Sigma_{D,T}^T \\ \Sigma_{\mathrm{D,T}} & \Sigma_{\mathrm{D,D}} \end{bmatrix} = \sigma_v^2 I + \begin{bmatrix} \mathrm{K_{T,T}} & \mathrm{K}_{D,T}^T \\ \mathrm{K_{D,T}} & \mathrm{K_{D,D}} \end{bmatrix},$$

Donde

$$\mathrm{K_{T,T}} = k(x,x)$$

$$\mathrm{K}_{D,T}^T = [k(x,x_1) \quad k(x,x_2) \quad \ldots \quad k(x,x_N)]$$

$$\mathrm{K_{D,D}} = \begin{bmatrix} k(x_1,x_1) & k(x_1,x_2) & \ldots & k(x_1,x_N) \\ k(x_2,x_1) & k(x_2,x_2) & \ldots & k(x_2,x_N) \\ \vdots & \vdots & \ddots & \vdots \\ k(x_N,x_1) & k(x_N,x_2) & \ldots & k(x_N,x_N) \end{bmatrix}$$

Los resultados de las secciones anteriores permiten escribir que

$$y \in N\,(\mu_y, \Sigma_{\mathrm{y}})$$

Donde

$$\mu_y = \hat{\mu} + \Sigma_{D,T}^T \Sigma_{D,D}^{-1} \left( y_D - \hat{\mu} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right) = \hat{\mu} + \mathrm{K}_{D,T}^T \left(\sigma_v^2 I + \mathrm{K_{D,D}}\right)^{-1} \left( y_D - \hat{\mu} \begin{bmatrix} 1 \\ 1 \\ \vdots \\ 1 \end{bmatrix} \right),$$

Y

$$\Sigma_{\mathrm{y}} = \Sigma_{\mathrm{T,T}} - \Sigma_{D,T}^T \Sigma_{D,D}^{-1} \Sigma_{\mathrm{D,T}} = \sigma_v^2 + k(x,x) - \mathrm{K}_{D,T}^T \left(\sigma_v^2 I + \mathrm{K_{D,D}}\right)^{-1} \mathrm{K_{D,T}}.$$

Elegir un kernel apropiado es muy importante para obtener unos resultados satisfactorios. Uno de los kernels que se usan más frecuentemente es la función de base radial:

$$k(x_a, x_b) = \sigma_a \sigma_b \exp\left( -\frac{\|x_a - x_b\|^2}{2\sigma_x^2} \right).$$

Hay que puntualizar que $\sigma_a$ y $\sigma_b$ son las asunciones iniciales de las desviaciones estándar para $x_a$ y $x_b$. Por lo tanto, $\exp\left(-\frac{\|x_a - x_b\|^2}{2\sigma_x^2}\right)$ es el factor de correlación entre $x_a$ y $x_b$.

### 5.4.1 Gaussianos una media.

En este subapartado se va a implementar el algoritmo de los Métodos Gaussianos. Hay que definir los parámetros $\hat{\mu}$ y $\sigma$. El primero, se va a escoger (arbitrariamente) como la media de los puntos disponibles del intervalo, es decir, como existirán medidas que serán NaN, dichas medidas no se tienen en cuenta. $\sigma$, como se ha visto en los apartados anteriores, es la desviación típica en el intervalo. Al igual que en la media, las medidas NaN no se tendrán en cuenta. Para ilustrar lo anterior, se va a representar la media y desviación típica de un contrato aleatorio:

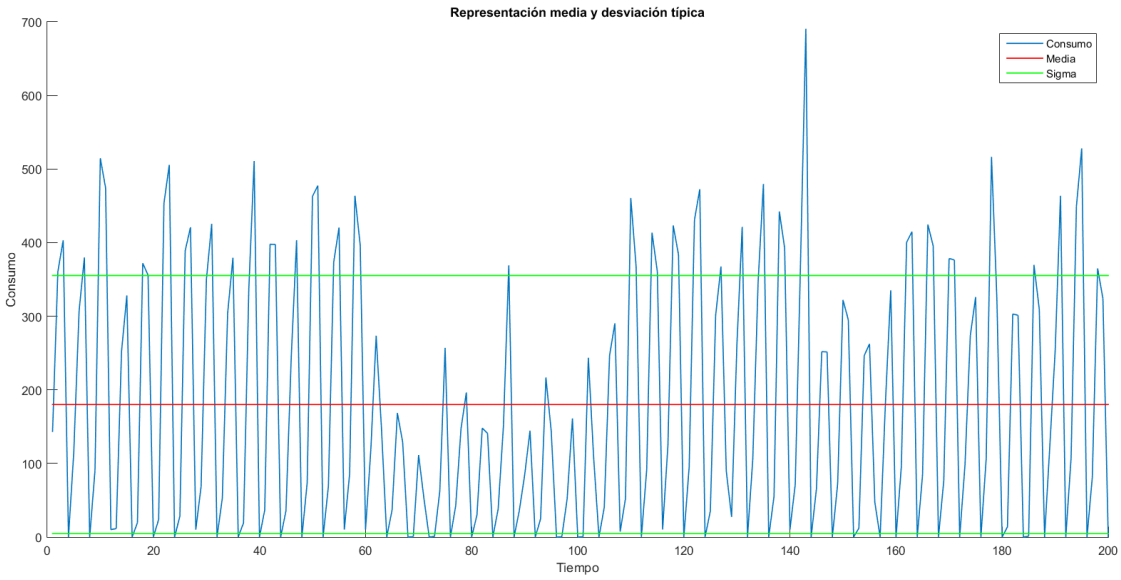



Figura 5-2. Representación media y varianza para un contrato aislado, procesos gaussianos teniendo en cuenta 1 media.

A continuación, se realiza la aproximación del consumo para el caso en que en dicho intervalo falten 10 puntos aleatorios. En la figura 5.3, se han representado los puntos que faltan con un asterisco rojo, la aproximación con un asterisco azul, y la banda de confianza de un sigma con círculos negros. Todos los demás puntos verdes son puntos conocidos y como tal, su banda de incertidumbre es mínima, solo viene dada por el ruido producido en la medición.

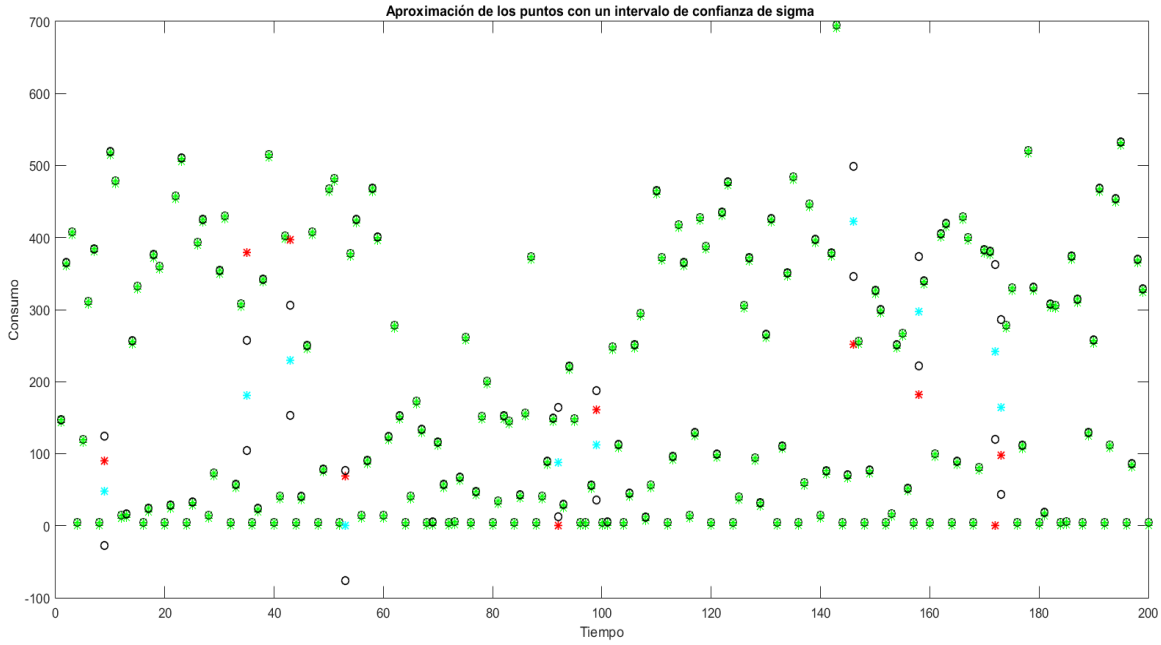



Figura 5-3. Aproximación contrato individual, procesos gaussianos teniendo en cuenta 1 media.

Para que se aprecie mejor la aproximación, se va a aislar los 10 puntos aproximados y se van a representar tres bandas de confianza distintas, para 1 sigma, para 2 sigmas y para 3 sigmas:

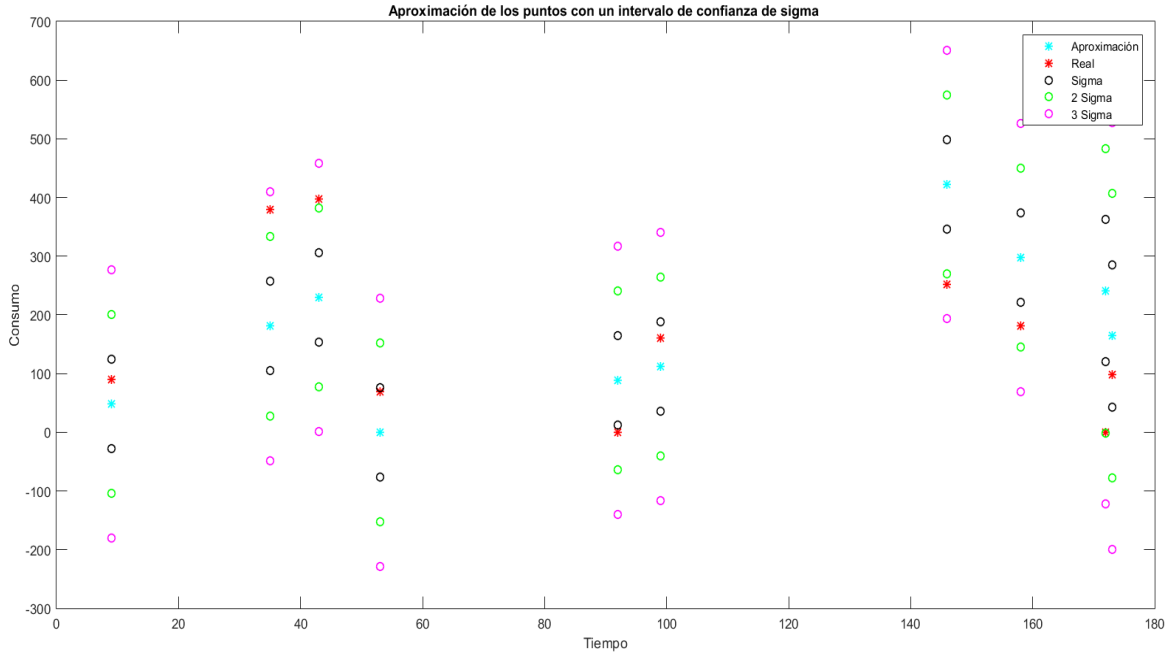



Figura 5-4. Puntos aproximados aislados, procesos gaussianos teniendo en cuenta 1 media.

Como se puede observar hay medidas reales que no están dentro del intervalo de confianza de un

sigma, pero sí lo están para 2 sigmas y 3 sigmas, por tanto, en este algoritmo habría que tener en cuenta si es útil en vez de coger un intervalo de confianza de un sigma, coger 2 o incluso 3 sigmas ya que quizás sea una representación más fiable, sin embargo, se ha decidido coger el intervalo de confianza de un sigma aún sabiendo que se pueden quedar puntos fuera.

Por último, se va a representar el error cometido durante todo el intervalo:

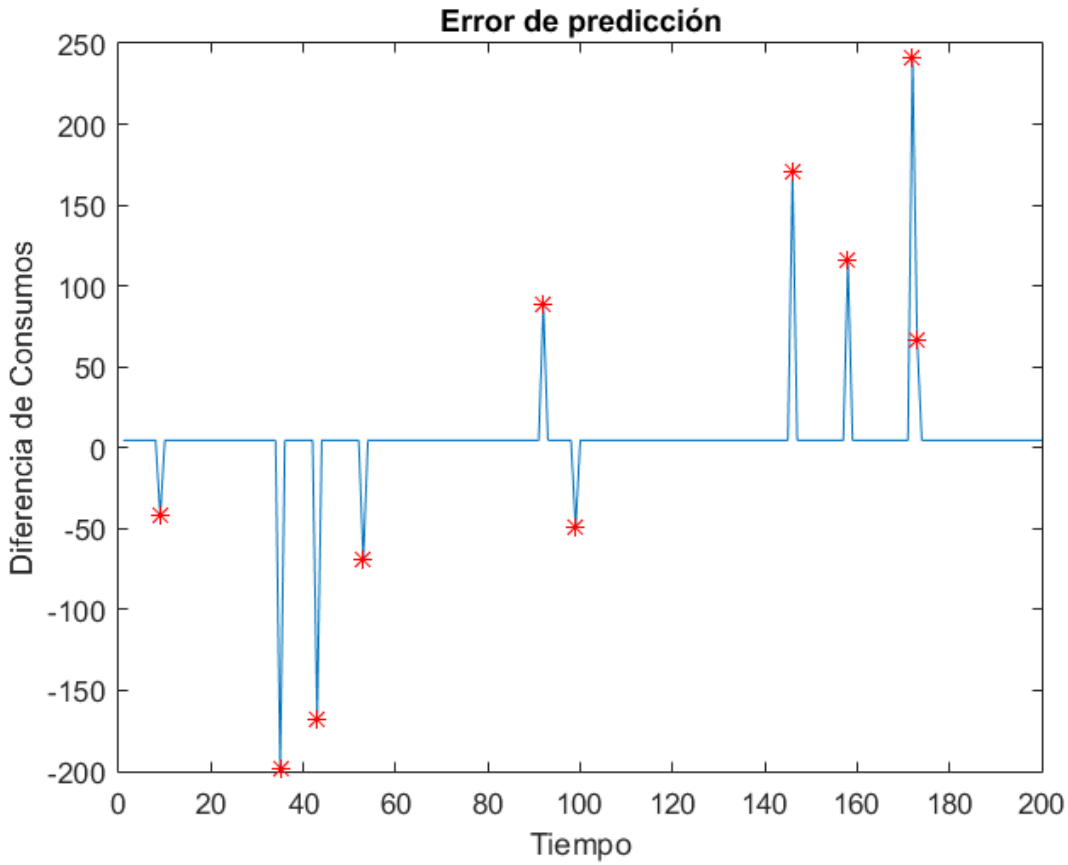



Figura 5-5. Error cometido, procesos gaussianos teniendo en cuenta 1 media.

En todos los puntos existe cierto error que viene dado por el ruido en la medición, por eso, aunque los datos sean conocidos el error está siempre por encima de 0.

Lo siguiente que se va a llevar a cabo es el método de aproximación realizado para todos los algoritmos anteriores, aunque en este caso existe una diferencia. En este algoritmo, se puede modificar tanto el parámetro $\hat{\mu}$ como el parámetro $\sigma_x$, el segundo parámetro es difícil saber a priori si funcionará mejor con un valor bajo o con uno alto, por tanto, lo que se ha realizado es una aproximación "óptima". Recordando el método de aproximación seguido en los apartados anteriores:

Para cada X_max y para cada P, se realiza 10 veces la aproximación y para cada aproximación se ha cogido el $\sigma_x$ óptimo, por ejemplo, para el caso de X_max=10 la tabla de las $\sigma_x$ óptimas queda:

Tabla 5-1. $\sigma_x$ óptimas para X_max = 10, procesos gaussianos teniendo en cuenta 1 media.

| Repet | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.5 | 0.2 | 0.3 | 1 | 1 | 0.6 | 0.5 | 1 | 1 | 0.2 |
| 2 | | 0.6 | 0.2 | 0.3 | 1 | 1 | 0.6 | 0.5 | 1 | 1 | 0.2 |
| 3 | | 0.8 | 1 | 1 | 0.8 | 0.9 | 1 | 1 | 0.4 | 0.3 | 1 |
| 4 | | 0.1 | 0.2 | 1 | 1 | 0.5 | 0.6 | 1 | 1 | 0.3 | 0.2 |
| 5 | | 0.6 | 0.2 | 0.3 | 1 | 1 | 0.6 | 0.5 | 1 | 1 | 0.9 |
| 6 | | 0.1 | 0.2 | 0.3 | 1 | 0.5 | 1 | 0.5 | 1 | 0.3 | 0.9 |
| 7 | | 0.5 | 0.2 | 0.3 | 1 | 0.5 | 0.6 | 0.5 | 1 | 1 | 0.2 |
| 8 | | 0.6 | 1 | 1 | 0.7 | 0.7 | 1 | 0.6 | 0.6 | 0.6 | 1 |
| 9 | | 0.1 | 0.9 | 0.3 | 1 | 1 | 1 | 0.5 | 1 | 1 | 0.9 |
| 10 | | 0.6 | 0.2 | 0.3 | 1 | 1 | 0.6 | 0.5 | 1 | 1 | 0.2 |

Siendo "Repet" el número de veces que se ha repetido la aproximación con contratos distintos para un mismo X_max y una misma P. Como se observa, no se puede coger una $\sigma_x$ que, a priori, aproxime de forma óptima en todos los casos. La tabla del error para este caso óptimo es:

Tabla 5-2. Evolución del error óptimo, procesos gaussianos teniendo en cuenta 1 media.

| X_max | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 1.0115 | 1.1037 | 1.2568 | 1.3046 | 1.1792 | 1.1694 | 1.2186 | 1.2055 | 1.1739 | 1.1260 |
| 2 | | 0.6 | 0.8034 | 0.8403 | 0.8384 | 0.8103 | 0.8920 | 0.9169 | 0.8891 | 0.8812 | 0.9048 |
| 3 | | 0.4445 | 0.7306 | 0.6774 | 0.6623 | 0.6904 | 0.8042 | 0.8053 | 0.7676 | 0.7592 | 0.8625 |
| 4 | | 0.3508 | 0.6297 | 0.6554 | 0.6672 | 0.6691 | 0.7551 | 0.7518 | 0.7719 | 0.7743 | 0.8327 |
| 5 | | 0.3449 | 0.6591 | 0.6737 | 0.6622 | 0.6676 | 0.7841 | 0.7781 | 0.7679 | 0.7797 | 0.8283 |
| 6 | | 0.5108 | 0.8095 | 0.7873 | 0.8516 | 0.8271 | 0.8007 | 0.8103 | 0.8463 | 0.8525 | 0.8657 |
| 7 | | 0.5350 | 0.8383 | 0.7899 | 0.8195 | 0.7816 | 0.8024 | 0.8010 | 0.8127 | 0.8036 | 0.8400 |
| 8 | | 0.4110 | 0.7234 | 0.6797 | 0.6859 | 0.7098 | 0.7350 | 0.7210 | 0.7257 | 0.7467 | 0.7852 |

| | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|
| 9 | 0.5342 | 0.7633 | 0.7910 | 0.8005 | 0.7605 | 0.7747 | 0.7802 | 0.7776 | 0.7773 | 0.7998 |
| 10 | 0.3396 | 0.8119 | 0.8310 | 0.7586 | 0.7197 | 0.8439 | 0.8246 | 0.8232 | 0.8035 | 0.8263 |

Si se compara esta tabla con la 4-6 es claro que este algoritmo aproxima mucho peor, por lo que en el siguiente apartado a consta de aumentar la dificultad se va mejorar para obtener unos mejores resultados.

### 5.4.2 Gaussianos distintas medias y distintas desviaciones típicas.

Este subapartado es muy similar al caso anterior, lo único que cambia es la elección de los parámetros $\hat{\mu}$ y $\sigma$. Como se realizan 4 medidas diarias, a las 3-5-9-14 horas, se van a tener en cuenta cuatro $\hat{\mu}$ distintas, una para cada tramo horario. Se procede de forma análoga con $\sigma$. Representando gráficamente las cuatro medias con sus respectivas desviaciones típicas se obtiene:

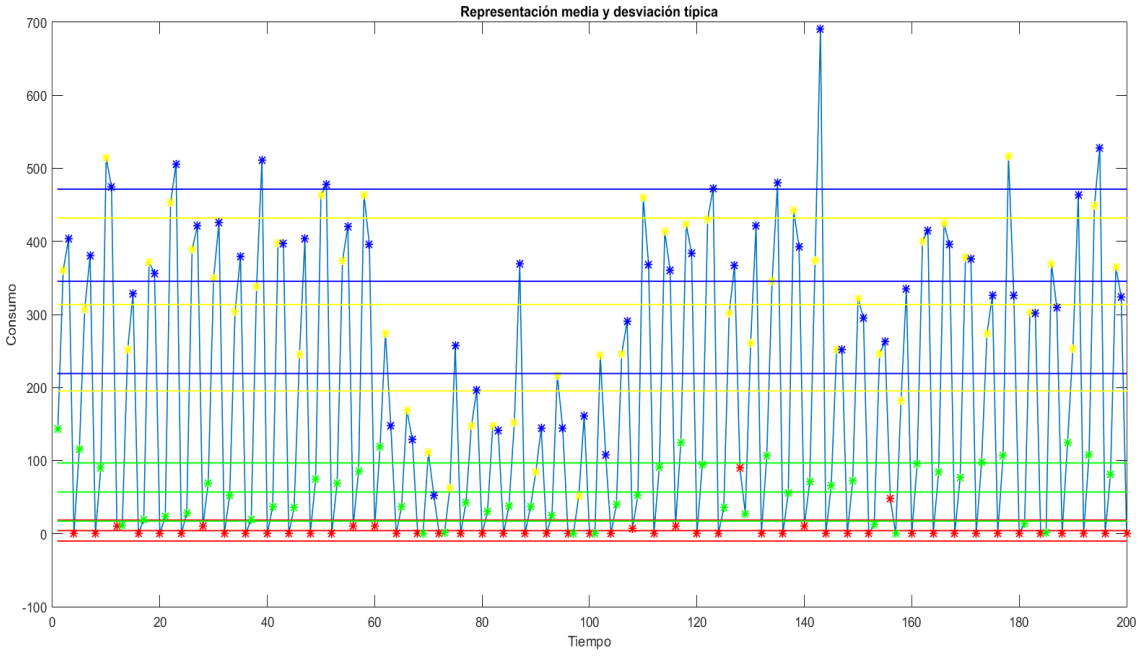



Figura 5-6. Representación media y varianza para un contrato aislado, procesos gaussianos teniendo en cuenta varias medias.

Cada color se corresponde con la media y la varianza para cada tramo horario, se observa que el tramo horario que está representado en rojo tiene una media muy homogénea, mientras que el tramo azul es el más heterogéneo, esto influirá a la hora de aproximar el consumo en el tramo.

A continuación, se quitan 10 puntos aleatorios en el intervalo y se realiza la aproximación. La figura 5-7 es similar a la 5-3, se ha representado lo mismo para dos casos distintos:

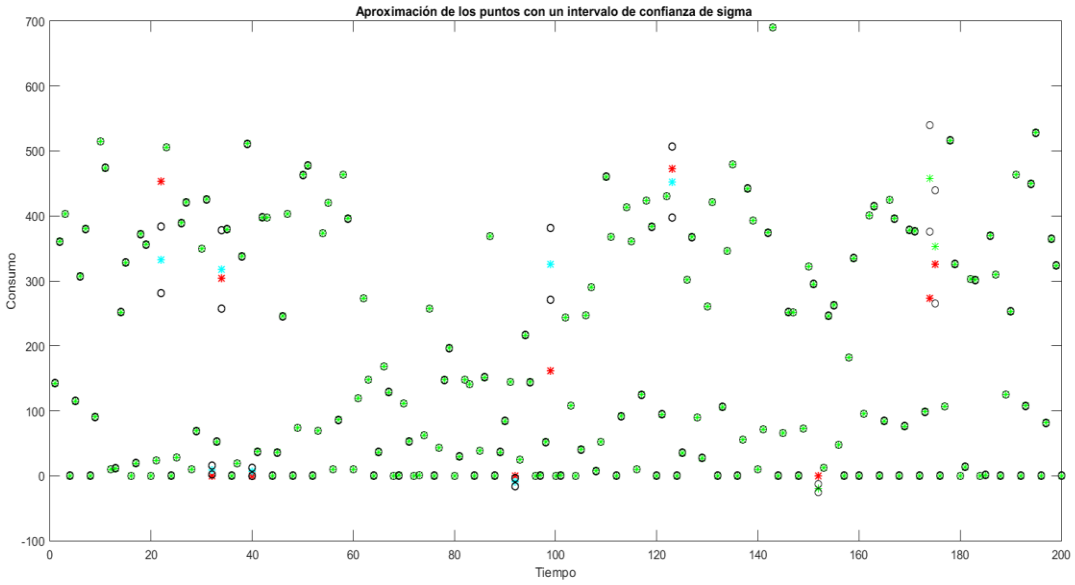



Figura 5-7. Aproximación contrato individual, procesos gaussianos teniendo en cuenta varias medias.

Para que se aprecie mejor la aproximación se va a aislar los 10 puntos aproximados y se van a representar tres bandas de confianza distintas, para 1 sigma, para 2 sigmas y para 3 sigmas:

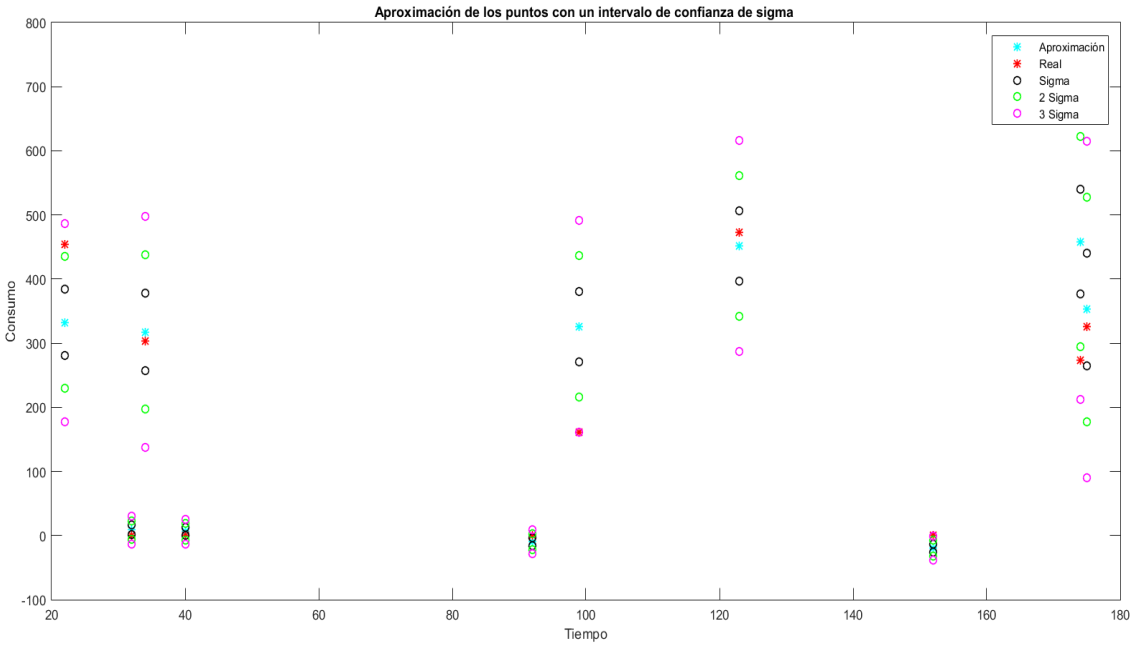



Figura 5-8. Puntos aproximados aislados, procesos gaussianos teniendo en cuenta varias medias.

Al igual que en el algoritmo anterior existen puntos reales que salen fuera del intervalo de confianza $\sigma$, pero están dentro del $2\sigma$ o $3\sigma$. Se puede apreciar que para los puntos que están en el inferior de la gráfica, la media y los intervalos de confianza son bastante pequeños por lo que esos puntos están bastante bien aproximados.

El error cometido para este contrato en concreto es:

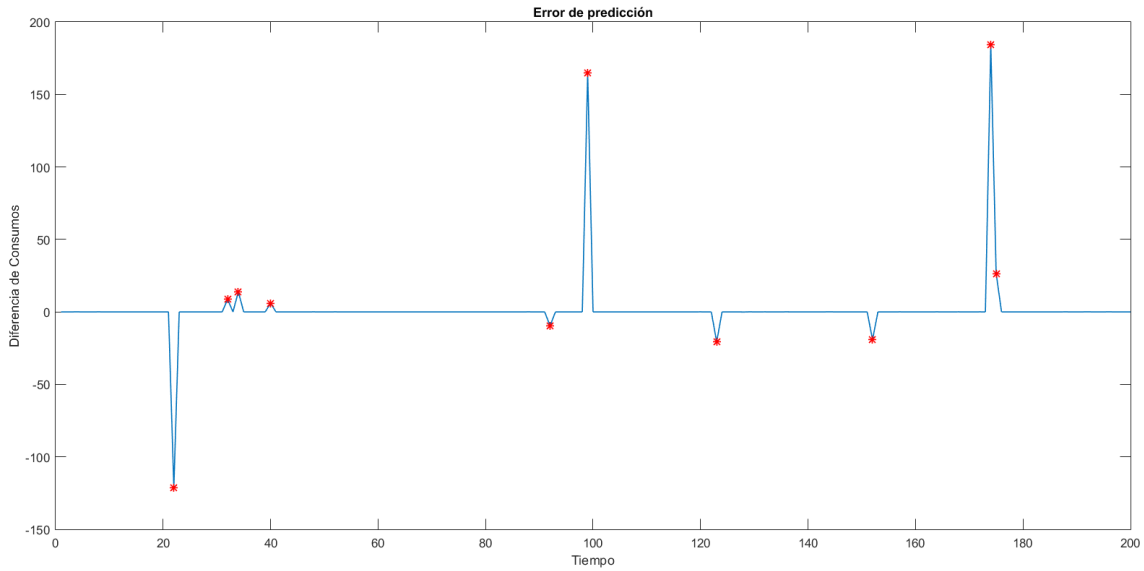



Figura 5-9. Error cometido, procesos gaussianos teniendo en cuenta varias medias.

Si se compara la figura 5-9 con la 5-5 se observa que en la 5-9 se obtienen unos errores menores y en algunos puntos, al ser de la zona donde las medidas tienen poca desviación típica se aproximan muy bien.

Se repite lo mismo que para el algoritmo de una media, la tabla de $\sigma_x$ óptimo para x_max=10 es:

Tabla 5-3. $\sigma_x$ óptimas para X_max = 10, procesos gaussianos teniendo en cuenta varias medias.

| Repet | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.7 | 1 | 1 | 1 | 1 |
| 2 | | 0.1 | 0.2 | 0.3 | 1 | 0.5 | 1 | 1 | 0.4 | 1 | 1 |
| 3 | | 0.1 | 1 | 1 | 1 | 1 | 1 | 0.5 | 0.4 | 0.3 | 0.2 |
| 4 | | 0.6 | 1 | 1 | 1 | 1 | 1 | 1 | 1 | 0.3 | 1 |
| 5 | | 0.5 | 1 | 0.6 | 1 | 0.6 | 0.7 | 0.5 | 0.4 | 0.3 | 1 |
| 6 | | 1 | 0.7 | 1 | 1 | 1 | 0.6 | 1 | 1 | 1 | 0.2 |
| 7 | | 0.6 | 0.6 | 0.6 | 0.7 | 0.7 | 0.7 | 0.5 | 0.4 | 1 | 0.7 |
| 8 | | 0.1 | 0.2 | 0.3 | 0.4 | 0.5 | 0.7 | 0.7 | 0.7 | 0.7 | 1 |
| 9 | | 0.1 | 1 | 0.3 | 0.4 | 0.6 | 1 | 0.5 | 1 | 0.5 | 1 |
| 10 | | 0.1 | 0.2 | 0.3 | 1 | 0.5 | 0.7 | 1 | 0.7 | 0.7 | 0.7 |

La tabla del error para este caso óptimo es:

Tabla 5-4. Evolución del error óptimo, procesos gaussianos teniendo en cuenta varias medias.

| X_max \ P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 0.5089 | 0.6885 | 0.7582 | 0.8069 | 0.7537 | 0.7030 | 0.7363 | 0.6835 | 0.6994 | 0.6491 |
| 2 | 0.2833 | 0.4759 | 0.3931 | 0.4299 | 0.4486 | 0.4474 | 0.4434 | 0.4306 | 0.4296 | 0.4288 |
| 3 | 0.2584 | 0.3248 | 0.3128 | 0.3839 | 0.4030 | 0.4387 | 0.4180 | 0.4332 | 0.4360 | 0.4430 |
| 4 | 0.2170 | 0.2975 | 0.2884 | 0.3530 | 0.3602 | 0.4215 | 0.4013 | 0.4312 | 0.4133 | 0.4121 |
| 5 | 0.2233 | 0.3387 | 0.2931 | 0.3445 | 0.3828 | 0.4433 | 0.4167 | 0.4203 | 0.4203 | 0.4260 |
| 6 | 0.2684 | 0.3745 | 0.3052 | 0.4159 | 0.3362 | 0.3941 | 0.3646 | 0.4027 | 0.3820 | 0.3908 |
| 7 | 0.1964 | 0.3747 | 0.2905 | 0.3955 | 0.3333 | 0.4110 | 0.3732 | 0.3775 | 0.3807 | 0.3905 |
| 8 | 0.2265 | 0.3087 | 0.2421 | 0.3095 | 0.2735 | 0.3299 | 0.2915 | 0.3256 | 0.3103 | 0.3202 |
| 9 | 0.2623 | 0.3098 | 0.2543 | 0.3393 | 0.3030 | 0.3529 | 0.3146 | 0.3349 | 0.3313 | 0.3262 |
| 10 | 0.2296 | 0.3591 | 0.2734 | 0.3493 | 0.3272 | 0.3676 | 0.3492 | 0.3581 | 0.3433 | 0.3528 |

Representando gráficamente la tabla anterior:

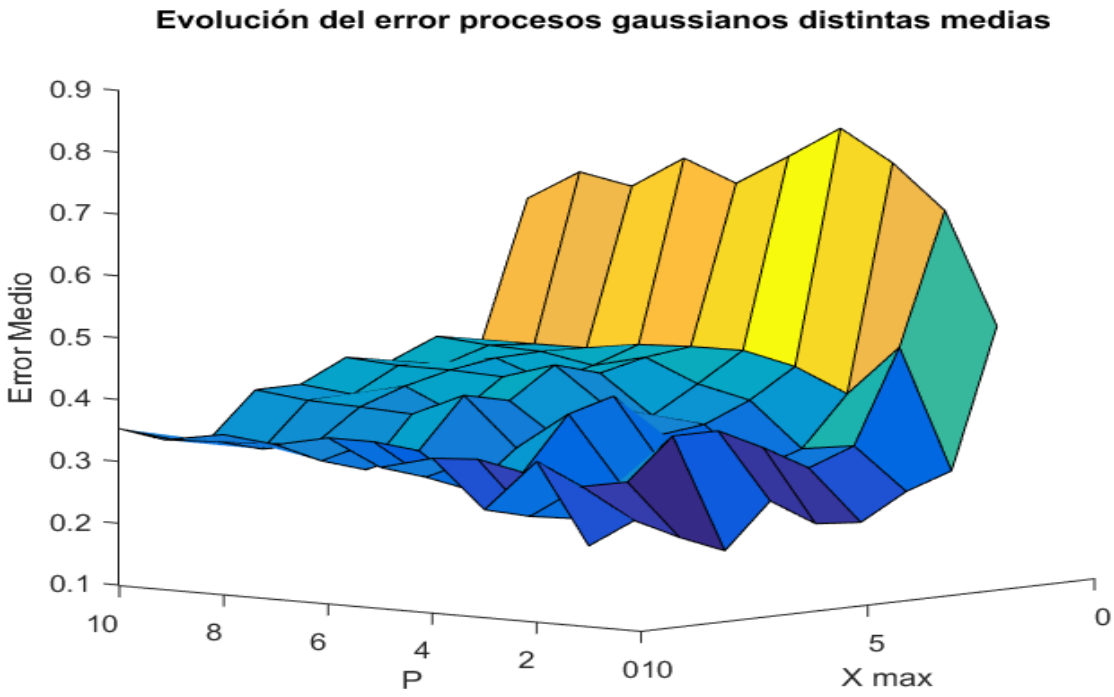



Figura 5-10. Evolución del error, procesos gaussianos teniendo en cuenta varias medias.

Al comparar los resultados obtenidos con los que aparecen en la tabla 4-7, se aprecia que todavía no se consigue mejorar el error obtenido por los mínimos cuadrados al completo, aunque en algunos tramos sí se consigue. Se va a representar gráficamente ambas tablas para que se vea de un modo más claro.

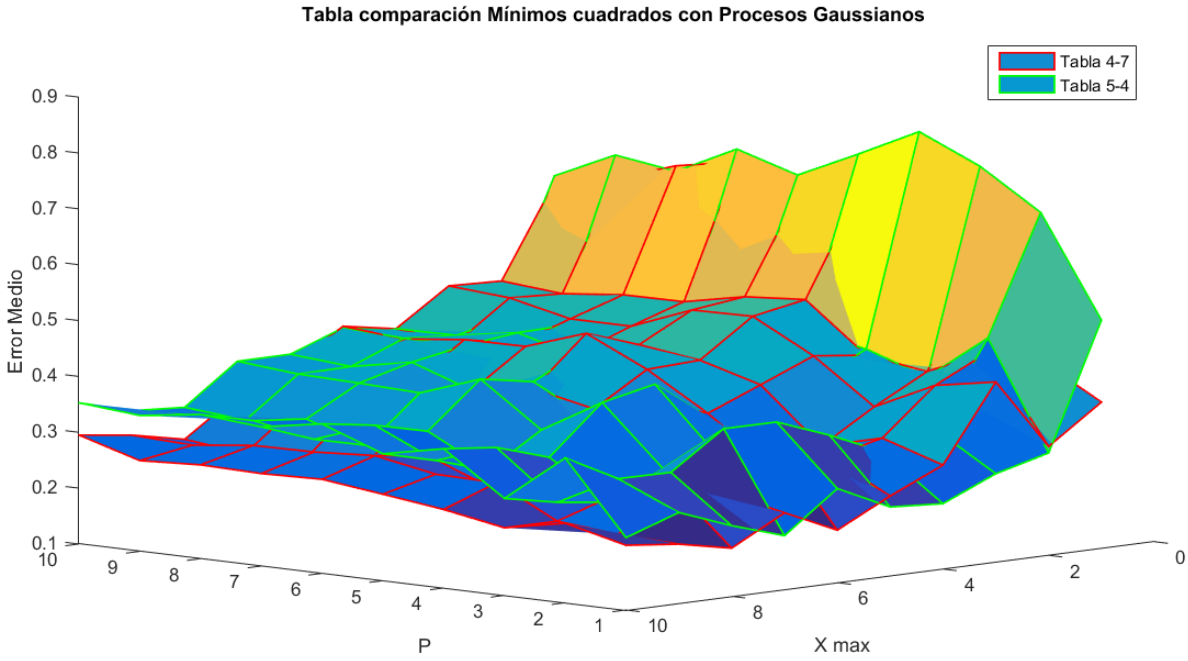



Figura 5-11. Comparación del error obtenido por mínimos cuadrados con el de procesos gaussianos.

Por último, se va a realizar la mejora hecha en los algoritmos anteriores, se va a tener en cuenta el consumo total del intervalo mediante un coeficiente α. Con esto se espera que se mejore el resultado obtenido con el algoritmo de los mínimos cuadrados.

### 5.4.3 Gaussianos distintas medias y distintas desviaciones típicas teniendo en cuenta el consumo total del intervalo.

El algoritmo no cambia sustancialmente por lo que se va a pasar directamente a ver los resultados obtenidos, al igual que para el caso anterior se ha realizado el proceso de aproximación eligiendo $\sigma_x$ óptimo, y esta es la tabla de error obtenido una vez introducido el coeficiente α:

Tabla 5-5. Evolución del error óptimo, procesos gaussianos teniendo en cuenta varias medias y coeficiente α.

| X_max | P | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | | 0.0960 | 0.2312 | 0.3494 | 0.4097 | 0.4888 | 0.4808 | 0.4603 | 0.4685 | 0.4775 | 0.4762 |
| 2 | | 0.1185 | 0.4042 | 0.3173 | 0.2890 | 0.3314 | 0.3134 | 0.3152 | 0.2973 | 0.2945 | 0.2990 |
| 3 | | 0.2560 | 0.2524 | 0.2307 | 0.2905 | 0.2911 | 0.2992 | 0.3130 | 0.3015 | 0.3221 | 0.3285 |
| 4 | | 0.1961 | 0.1992 | 0.2008 | 0.2675 | 0.2741 | 0.3359 | 0.3128 | 0.3323 | 0.3284 | 0.3149 |
| 5 | | 0.2229 | 0.1933 | 0.2217 | 0.2676 | 0.2971 | 0.3724 | 0.3410 | 0.3349 | 0.3440 | 0.3394 |
| 6 | | 0.1155 | 0.1545 | 0.1828 | 0.2380 | 0.2313 | 0.2889 | 0.2926 | 0.3058 | 0.2868 | 0.2937 |
| 7 | | 0.0965 | 0.1003 | 0.1770 | 0.1705 | 0.1695 | 0.2424 | 0.2450 | 0.2465 | 0.2509 | 0.2540 |
| 8 | | 0.1579 | 0.1398 | 0.1492 | 0.1623 | 0.1489 | 0.2369 | 0.2290 | 0.2294 | 0.2330 | 0.2345 |
| 9 | | 0.1045 | 0.0957 | 0.1250 | 0.1541 | 0.1657 | 0.2422 | 0.2270 | 0.2351 | 0.2494 | 0.2422 |
| 10 | | 0.1076 | 0.1403 | 0.1425 | 0.1635 | 0.1664 | 0.2082 | 0.1891 | 0.2119 | 0.2156 | 0.2509 |

Representándola gráficamente se obtiene:

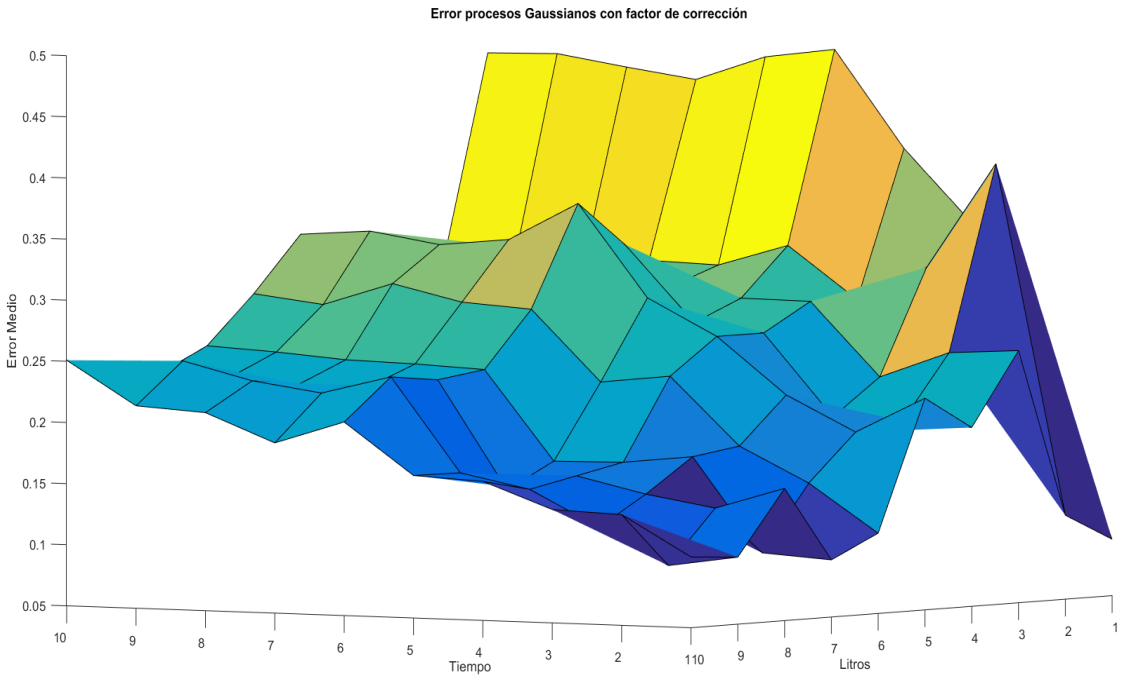



Figura 5-12. Evolución del error, procesos gaussianos teniendo en cuenta varias medias y coeficiente alfa.

Para ver si se ha producido mejora se van a comparar gráficamente las dos tablas de error junto con una tabla en la que se muestra de forma clara cómo evoluciona el error, se han tenido en cuenta 100 aproximaciones para cada celda de la misma:

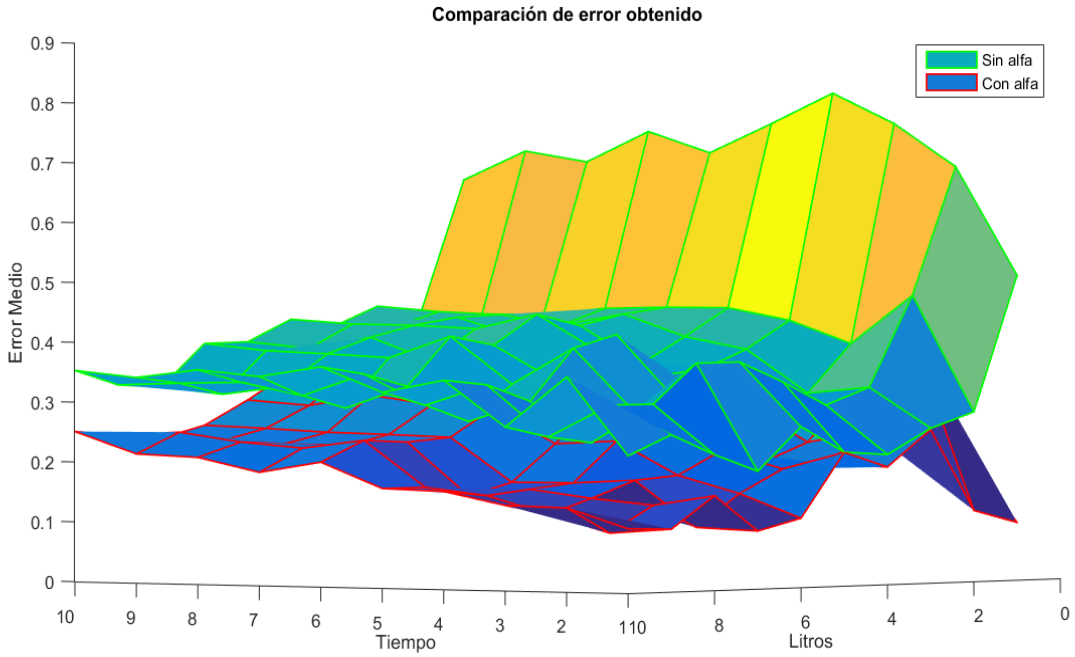



Figura 5-13. 1ª Comparación del error para procesos gaussianos con y sin coeficiente alfa.

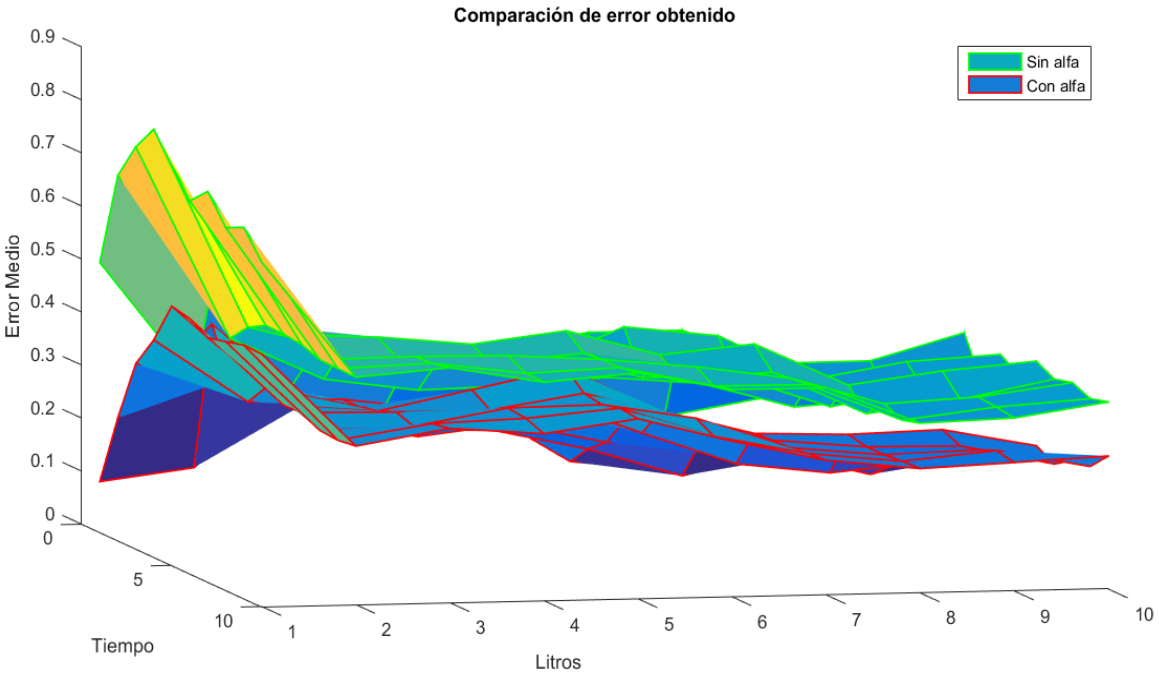



Figura 5-14. 2ª Comparación del error para procesos gaussianos con y sin coeficiente alfa.

Tabla 5-6. Comparación de errores para 100 aproximaciones, Procesos Gausianos.

| Error(%) \ N Contratos | 1 | 3 | 5 | 10 |
|---|---|---|---|---|
| Gaussianos con cuatro medias | 53.5 | 34.6 | 28.3 | 23.0 |
| Gaussianos con cuatro medias a | 24.8 | 20.2 | 16.1 | 13.2 |

Teniendo en cuenta el consumo del intervalo se ha disminuido considerablemente el error cometido en la aproximación. Lo que queda por hacer es comparar los resultados de la tabla 5-5 con los obtenidos en la tabla 4-7 y en la tabla 3-2:

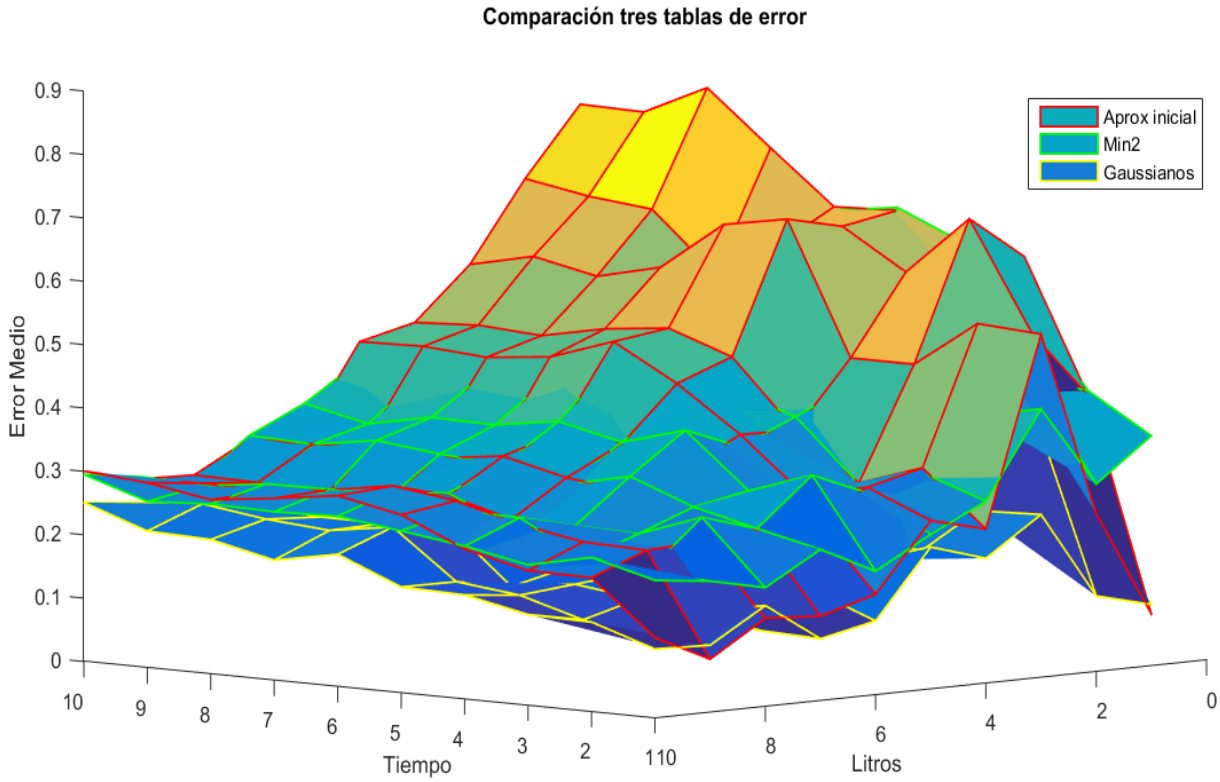



Figura 5-15. 1ª Comparación del error para los tres algoritmos usados.

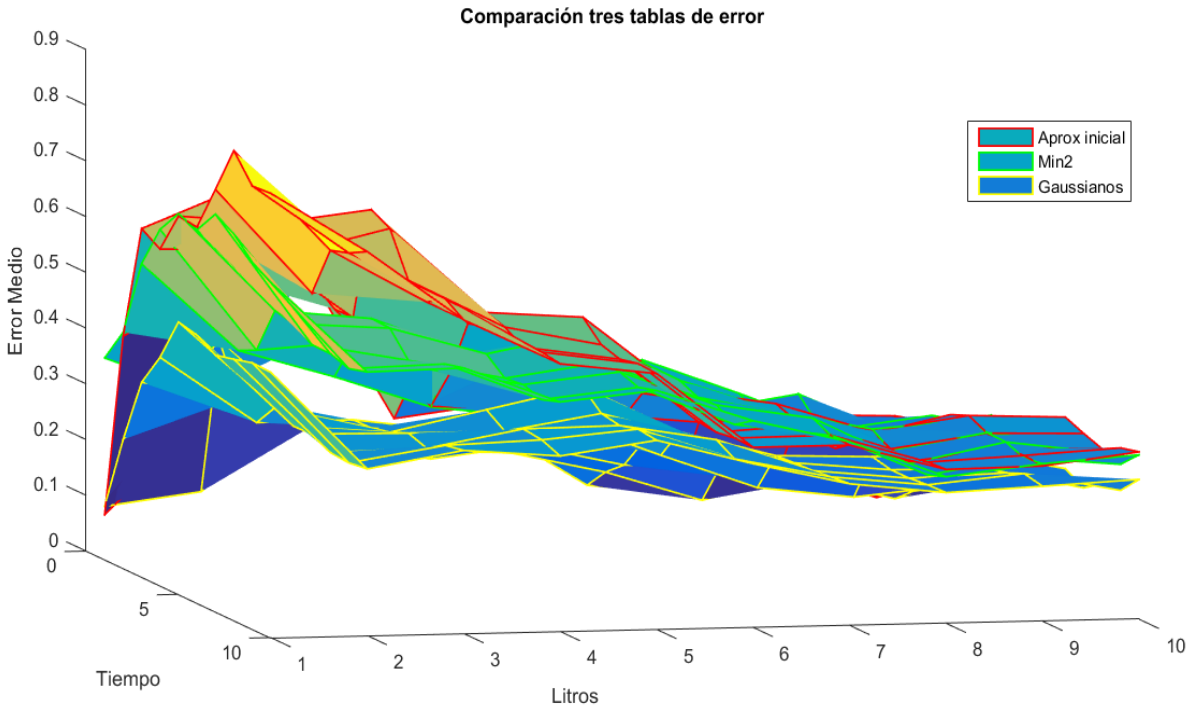



Figura 5-16. 2ª Comparación del error para los 3 algoritmos usados.

A modo conclusión, se va a realizar una tabla teniendo en cuenta 100 aproximaciones de la misma forma que en algoritmos anteriores:

Tabla 5-7. Comparación de errores para 100 aproximaciones, Conclusión.

| N Contratos / Error(%) | 1 | 3 | 5 | 10 |
|---|---|---|---|---|
| Aproximación Inicial a | 41.7 | 34.2 | 25.1 | 17.4 |
| Mínimos Cuadrados a | 45.3 | 33.3 | 23.2 | 19.2 |
| Procesos Gaussianos a | 24.8 | 20.2 | 16.1 | 13.2 |

El mejor error se obtiene para el caso de los Procesos Gaussianos, tal y como se ve en la tabla anterior, se mejora los resultados obtenidos por los otros algoritmos de forma clara. A pesar de que tienen una dificultad mayor con respecto a los otros dos algoritmos es conveniente usar los métodos gaussianos porque sin hacer muchas modificaciones se obtiene una mejora significativa. Como trabajo futuro se podría seguir mejorando los procesos gaussianos (cambiando el Kernel, usando algún tipo de coeficiente entre contratos para que los más parecidos ponderen más, etc).

# 6 CÓDIGO MATLAB

En este capítulo se va a desarrollar todo el código necesario para la realización del proyecto, con el objetivo de que se vea como se han implementado en Matlab los algoritmos anteriormente descritos. Hay muchas funciones y muchas comprobaciones que no se han introducido debido a que se ha pretendido que este capítulo sea lo más escueto posible.

## 6.1 Tratamiento previo.

### 6.1.1 Quitar puntos repetidos.

```
%% Quita las medidas duplicadas.

function[filtro,eliminados]=Quitarpuntosrepetidos(x,nmu,matTelem)

% x = contratos en los que se quiere quitar los datos duplicados.
% nmu = número de medidas cada usuario.

filtro=[];                  %Aquí es donde se guardan los datos filtrados.
z=1;
eliminados=[];              %Se almacenan los puntos repetidos.
for n=1:length(x)
    if(x(n)==1)
        idatos=1;
        fdatos=nmu(x(1));
    else
        idatos=sum(nmu(1:(x(n)-1)))+1;    %inicio de datos.
        fdatos=sum(nmu(1:x(n)));             %fin de datos;
    end
    for i=idatos:fdatos
        if(i==1)
            filtro=[filtro;matTelem(1,:)];              %concatenación
vertical (tiene ;), de forma que lo que se tiene en cuenta es que tengan el
mismo numero de columnas.
        else if(matTelem(i,2)-matTelem(i-1,2)==0)
                eliminados(z)=i;
                z=z+1;
            else
                filtro=[filtro;matTelem(i,:)];
            end
        end
    end
end
end
```

### 6.1.2 Poner NaN.

Este programa se encargará de escanear los datos y donde falte uno, añadirá una fila de NaN. Se parte de los datos filtrados.

```
function[dfinales]=Ponernan(diniciales)

%   diniciales = datos iniciales
%   dfinales   = datos finales con NaN.

dias=datevec(diniciales(:,2));  %se transforma la columna 2 en una fecha.
dfinales=[];
n=0;
n1=0;
j=1;

for i=1:(length(diniciales)-1)                  %Como se evalua el if en i+1
tiene que ser hasta length(diniciales)-1
    if(diniciales(i,1)==diniciales(i+1,1))      %Si está dentro del mismo
contrato.
        if(dias(i,3)==dias(i+1,3))              %Si está en el mismo día.
            if(dias(i+1,4)-dias(i,4)==2 || dias(i+1,4)-dias(i,4)==4 ||
dias(i+1,4)-dias(i,4)==5)    %se meten los casos correctos.
                dfinales(j,:)=diniciales(i,:);
                j=j+1;
            else                                %en caso de que el resultado
no sea ninguno de los anteriores es que faltan datos.
                switch  dias(i+1,4)-dias(i,4)   %dependiendo del resultado
faltan una cantidad de datos distinta, se añaden las filas de NaN
correspondientes.
                    case 6
                        dfinales(j,:)=diniciales(i,:);
                        dfinales=[dfinales;NaN(1,8)];
                        j=j+2;
                    case 9
                        dfinales(j,:)=diniciales(i,:);
                        dfinales=[dfinales;NaN(1,8)];
                        j=j+2;

                    case 11
                        dfinales(j,:)=diniciales(i,:);
                        dfinales=[dfinales;NaN(2,8)];
                        j=j+3;
                    otherwise
                        fprintf('error1')
                end
            end
        else                                %En caso de que se cambie de día.
            if(dias(i+1,3)-dias(i,3)<0)                     %Esto sólo ocurre
al cambiar de mes, por tanto, según el mes en el está se realiza una cosa u
otra para ver cuantos datos faltan.
                switch dias(i,2)

                    case 1
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 2
                        n=28+(dias(i+1,3)-dias(i,3))-1;
```

```
                        n1=n-1;
                    case 3
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 4
                        n=30+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 5
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 6
                        n=30+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 7
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 8
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 9
                        n=30+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 10
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 11
                        n=30+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;
                    case 12
                        n=31+(dias(i+1,3)-dias(i,3))-1;
                        n1=n-1;

                    otherwise
                        disp(i)
                        fprintf('error2')
                end
            end
            if(dias(i+1,3)-dias(i,3)>0)                    %En caso de que
la diferencia no sea menor que 0.
                n1=dias(i+1,3)-dias(i,3)-1;
                n=dias(i+1,3)-dias(i,3);
            end
            switch dias(i+1,4)-dias(i,4)                %se mira cual es el
resultado de restar la hora i y la siguiente, y según el resultado
                                                        %se sabe cuantos
datos faltan y se introducen las filas de NaN correspondientes.
                case 0
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+3,8)];
                    j=j+n1*4+3+1;
                case 2
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4,8)];
                    j=j+n*4+1;
                case 4
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4,8)];
                    j=j+n*4+1;
                case 5
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4,8)];
                    j=j+n*4+1;
```

```
                case 6
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4+1,8)];
                    j=j+n*4+1+1;
                case 9
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4+1,8)];
                    j=j+n*4+1+1;
                case 11
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n*4+2,8)];
                    j=j+n*4+2+1;
                case -2
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+2,8)];
                    j=j+n1*4+2+1;
                case -4
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+2,8)];
                    j=j+n1*4+2+1;
                case -5
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+2,8)];
                    j=j+n1*4+2+1;
                case -6
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+1,8)];
                    j=j+n1*4+1+1;
                case -9
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4+1,8)];
                    j=j+n1*4+1+1;
                case -11
                    dfinales(j,:)=diniciales(i,:);
                    dfinales=[dfinales;NaN(n1*4,8)];
                    j=j+n1*3+1;
                otherwise
                    fprintf('error')
            end

        end
        if(i==(length(diniciales)-1))                           %Por último
lo que se realiza aquí es tener en cuenta que como se llega hasta
lentgh(diniciales)-1
            dfinales=[dfinales;diniciales(i+1,:)];              %no se llega
a diniciales(i+1) y por tanto hay que meterlo en dfinales.
        end
    end
end
```

### 6.1.3 Fecha y Hora.

Este programa se va a encargar de poner fecha y hora a los datos con las filas de NaN.

```
function [dconfecha]=Fechayhora(diniciales_Nan)

%diniciales_Nan = datos con las filas de NaN.
%dconfecha      = datos con fecha.

c1=0;           %Indice del contrato que se está recorriendo.
fecha=datevec(diniciales_Nan(:,2));
for i=2:length(diniciales_Nan-1)
    if(c1~=diniciales_Nan(i,1) && isnan(diniciales_Nan(i,1))==0)    %Si el
dato actual no es NaN y es un nuevo contrato.
        c1=diniciales_Nan(i,1); %Se actualiza c1.
    end
    if(isnan(diniciales_Nan(i,2))==1)    %Si falta la fecha.
        if(isnan(diniciales_Nan(i-1,2))==0)         %coge el dato anterior.
            if(fecha(i-1,4)==5)                     %Se aplica la casuística
para poner la fecha.
                diniciales_Nan(i,2)=diniciales_Nan(i-1,2)+0.166666666627862;
                diniciales_Nan(i,1)=c1;
            end
            if(fecha(i-1,4)==9)
                diniciales_Nan(i,2)=diniciales_Nan(i-1,2)+0.208333333372138;
                diniciales_Nan(i,1)=c1;
            end
            if(fecha(i-1,4)==14)
                diniciales_Nan(i,2)=diniciales_Nan(i-1,2)+0.541666666627862;
                diniciales_Nan(i,1)=c1;
            end
            if(fecha(i-1,4)==3)
                diniciales_Nan(i,2)=diniciales_Nan(i-1,2)+0.083333333372138;
                diniciales_Nan(i,1)=c1;
            end
        else
            fprintf('ERROR, NO SE HA PODIDO PONER LA FECHA EN LA MEDIDA
%d\n',i)
        end
    end
end
  dconfecha=diniciales_Nan;
end
```

### 6.1.4 Índices.

Se guarda en el vector "nmu" el número de medidas que tiene cada contrato.

```
function[nmu]=indices(matTelem)

%nmu = numero de medidas que disponemos de cada usuario.
%matTelem = datos sin tratar.

nmu=zeros(1,max(matTelem(:,1)));
j=matTelem(1,1);
cont=0;

for i=1:length(matTelem(:,1))
        if(matTelem(i,1)==j)
            cont=cont+1;
        end
        if(matTelem(i,1)~=j && matTelem(i,1)~=-1)
            nmu(j)=cont;              % se asigna a nmu el numero de datos
            while(matTelem(i,1)~=j)
            j=j+1;                    %se incrementa j para que pase al
siguiente contrato y se pone cont a 1 para tener en cuenta la muestra actual
que es del contrato siguiente.
            end
            cont=1;
        end
end
nmu(j)=cont;

end
```

### 6.1.5 Cálculo Consumo.

Esta función va destinada al calculo de consumo para cuando se utiliza agregados, ya que para un sólo contrato con una simple operación (consumo= volumen(2:end)-volumen(1:end-1)) se obtiene el consumo.

```
%% Programa para ver el gasto después del filtrado.

function[consumo]=calculoconsumo(dfinales)

y=1;
for(i=1:(length(dfinales(:,1))-1))
    if(dfinales(i+1,1)==dfinales(i,1))
%Si se está en el mismo contrato.
        consumo(i,1)=y;
        consumo(i,2)=dfinales(i+1,4)-dfinales(i,4);
%Cálculo del consumo.
    else
        consumo(i,1)=-1;
        consumo(i,2)=-1;
%Al cambio de contrato se introduce -1 en ambos campos para fácil
identificación.
        y=y+1;
    end
end
consumo(i+1,1)=-1;      %Por dimensiones.
consumo(i+1,2)=-1;      %Para compatibilizar las divisiones en otras
funciones.

end
```

### 6.1.6 Procesos Gaussianos.

Código íntegro del algoritmo de los procesos Gaussianos.

```
clear all
clc
load('datosconfecha50');
primera=0;
porcentaje=50;
puntosb=[100,300];
paprox=250;
agregado=1;
x_inicial=10:1200;
for x_max=1:10
    paprox=250;
    for p=1:10
        paprox(p)=paprox(1)+p-1;
        inicio=1;
        coeficiente=zeros(10,length(paprox)+1);
        coeficiente_provisional=zeros(1,length(paprox)+1);
        coeficiente2=zeros(10,length(paprox)+1);
        for contador1=1:10
            x=x_inicial(inicio:end);
            tabla=[];
            [nmu]=indices(datos_con_fecha);
            c_x=escaneoprograma(x,x_max,datos_con_fecha,puntosb,porcentaje);
            inicio=c_x(end,1)+1;
            c_x_aisl=aislarzonacontratos_mod(c_x,datos_con_fecha);
            if(agregado==1)
                volumen_agregado=sumavolumen(c_x_aisl,c_x);
                c_x_aisl=volumen_agregado;

c_x_spuntos=Quitarpuntosaproximar(agregado,c_x,c_x_aisl,paprox,puntosb);
%sin puntos que se quieren aproximar.
            else

c_x_spuntos=Quitarpuntosaproximar(agregado,c_x,c_x_aisl,paprox,puntosb);
            end
            consumo_x=calculoconsumo(c_x_spuntos);
            consumo_x_real=calculoconsumo(c_x_aisl);
            i=1;
            cont=0;
            while(cont<=length(consumo_x))                              %Esta
parte es para quitar el -1 de las x.
                if(consumo_x(i,1)==-1)
                    consumo x(i,:)=[];
                    consumo_x_real(i,:)=[];
                else
                    i=i+1;
                end
                cont=cont+1;
            end
            if(consumo_x(end,:)==-1)
                consumo_x(end,:)=[];
                consumo_x_real(end,:)=[];
            end
            ind=consumo_x;
            tramo=0;
             for i=1:length(consumo_x)
                if(isnan(consumo_x(i,2)))
```

```
                    if(tramo==0)
                                tramo=1;
                                v_inicial=c_x_spuntos(i,4); %Luego se usara
para calcular alfa.
                                c_inicial=i;
                    end
                    if(tramo==1 && (isnan(consumo_x(i+1,2))==0 ||
consumo_x(i,1)~=consumo_x(i+1,1)))
                                v_final=c_x_spuntos(i+1,4); %Luego se usara
para calcular alfa.
                                c_final=i;
                    end
                end
             end
            %%%%%%%% SE ENTRA EN EL ALGORITMO GAUSSIANO %%%%%%%%%%%%%%
            %% Calcular media y varianza:
            Fechas=c_x_aisl(1:end-1,2);
            Horas=round(24*(Fechas-floor(Fechas)));
            Intervalo=zeros(length(Horas),1);
            I=Horas==3; Intervalo(I)=1;
            I=Horas==5; Intervalo(I)=2;
            I=Horas==9; Intervalo(I)=3;
            I=Horas==14; Intervalo(I)=4;
            media_p=zeros(4,1);
            sigma_p=zeros(4,1);
            for kk=1:4
                I=Intervalo==kk;
                media_p(kk)=mean(consumo_x_real(I,2));
                sigma_p(kk)=std(consumo_x_real(I,2));
            end
            plantilla=isnan(ind(:,2))==0;  %% En esta todo vale 1 menos NaN
            plantilla2=isnan(ind(:,2))==1; %% En esta es al contrario.
            media_error=mean(ind(plantilla,2));

            % Se eligen los demás parámetros necesarios para el algoritmo:

            primera=1;
            sigma_v=0.01;                       %Se supone que existe muy poco
ruido en la medición.
            yD=0;
            j=1;
            %Se almacena donde están las posiciones con NaN:
            tiempo=1:length(ind(:,2));
            I=tiempo(plantilla2);

            Consumo_dato=ind(plantilla,2);



            Data_Vol=consumo_x_real(:,2);
            Data_Int=Intervalo;
            Data_Fec=tiempo';
            Data_Vol(I)=[];     %Puntos Conocidos
            Data_Int(I)=[];     %Quita los puntos con NaN de Data_Int,
Data_Fec y Data_Vol
            Data_Fec(I)=[];

            Test_consumo=consumo_x_real(plantilla2,2);
            Test_t=tiempo(plantilla2);
            Test_Int=Intervalo(I);

            for sigma_w=1:-0.1:0.1
```

```
                nD=length(Data_Fec);
                K_DD=zeros(nD);
                Data_Delta_Vol=0;
                for ii=1:nD
                    Fec_ii=Data_Fec(ii);
                    Int_ii=Data_Int(ii);
                    sigma_ii=sigma_p(Int_ii);
                    mean_ii=media_p(Int_ii);
                    Data_Delta_Vol(ii,1)=Data_Vol(ii)-mean_ii;

                    for jj=1:nD
                        Fec_jj=Data_Fec(jj);
                        Int_jj=Data_Int(jj);
                        sigma_jj=sigma_p(Int_jj);
                        K_DD(ii,jj)=sigma_ii*sigma_jj*exp(-
(0.5/sigma_w^2)*abs(Fec_ii-Fec_jj)^2);
                    end
                end

                for kk=1:length(Fechas)
                    Fec_kk=tiempo(kk);
                    Int_kk=Intervalo(kk);
                    sigma_kk=sigma_p(Int_kk);
                    mean_kk=media_p(Int_kk);
                    K_TD=zeros(1,nD);
                    for jj=1:nD
                        Fec_jj=Data_Fec(jj);
                        Int_jj=Data_Int(jj);
                        sigma_jj=sigma_p(Int_jj);
                        K_TD(jj)=sigma_kk*sigma_jj*exp(-
(0.5/sigma_w^2)*abs(Fec_kk-Fec_jj)^2);

                    end

                    y=mean_kk+K_TD*((sigma_v^2*eye(nD)+K_DD)\Data_Delta_Vol);
                    if(y<0)
                        Y_final(kk)=0;
                    else
                        Y_final(kk)=y;
                    end
                    Sigma_y=sigma_v^2+sigma_kk^2-
K_TD*((sigma_v^2*eye(nD)+K_DD)\K_TD');
                end

                alfa=(v_final-v_inicial)/sum(Y_final(c_inicial:c_final));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                Y_final(plantilla2)=Y_final(plantilla2).*alfa;
                tabla=[consumo_x_real(plantilla2,2) Y_final(plantilla2)'];
                for ii=1:length(tabla(:,1))
                    coeficiente_provisional(ii)=abs(tabla(ii,1)-
tabla(ii,2))/media_error;
                end
                if(primera==1)
                    primera=0;
                    c_x;
                    coeficiente(contador1,:)=coeficiente_provisional;
                    coeficiente2(contador1,:)=1;
                else
```

```
plantilla_c=coeficiente_provisional<coeficiente(contador1,:);

                    for ll=1:length(plantilla_c)
                        if(plantilla_c(ll)==1)

coeficiente(contador1,ll)=coeficiente_provisional(ll); %Se intercambian los
indices de sitio porque uno de ellos es vector columna y otro vector fila.
                            coeficiente2(contador1,ll)=sigma_w;
                        end
                    end
                end
            end
        end
        tamano=size(coeficiente);
        coeficiente_medio=sum(sum(coeficiente))/(tamano(1,1)*tamano(1,2));
        tabla_(x_max,p)=coeficiente_medio;  %Tabla representada en las
gráficas con los errores medios para cada caso.
    end
end
```

### 6.1.7 Mínimos Cuadrados.

Al ser este código muy similar al anterior, sólo se va a incluir la parte que cambia, a partir de donde pone en el caso anterior "%%%%%%%% SE ENTRA EN EL ALGORITMO GAUSSIANO %%%%%%%%%%%%%%".

```
medidas=120;            %Indica el numero de datos anteriores que se quieren
tener en cuenta para M (regresor).
n=100; %Contratos que se quieren mirar para hacer el agregado, si es un
intervalo se miran todos y si no hay suficientes
% se para (en caso de que haya algún contrato válido) o se sigue escaneando
desde fuera del intervalo hasta que haya nc contratos válidos.
%En caso de que no sea un vector y sea solo un número, será el punto de
inicio de escaneo hasta que se tengan "nc" contratos validos.
nc=11;       %Número de contratos que se quieren tener en cuenta como máximo
para el agregado.
agregado=1;
c1=1;

%%%%%%%%% SE ENTRA EN MINIMOS CUADRADOS %%%%%%%%%%%%%%%

%El regresor M está formado por por Y(k-1) Y(k-2) Y(k-3)
%Y(k-4) ag(k) ag(k-1) ag(k-2) ag(k-3) ag(k-4)
z=5;
M=zeros(length(medidas),9);
Y=zeros(1,length(medidas));
for j=(i-medidas):(i-1)       % El bucle va hasta i-1 porque la última medida
que se quiere tener en cuenta es la anterior a la actual.
    M(z-4,1)=ind(j-1,2);
    M(z-4,2)=ind(j-2,2);
    M(z-4,3)=ind(j-3,2);
    M(z-4,4)=ind(j-4,2);
    M(z-4,5)=ag(j,2);
    M(z-4,6)=ag(j-1,2);
    M(z-4,7)=ag(j-2,2);
    M(z-4,8)=ag(j-3,2);
    M(z-4,9)=ag(j-4,2);
    Y(z-4)=consumo_x_aisl(j,2);   %Hay que modificar el segundo indice.
    z=z+1;
end

% Se calcula el factor de olvido:
W=zeros(length(M(1,:)));
%descomentar para obtener landa optimo
%% for(landa=0.8:0.0001:1)
landa=0.98;  %  Elegido de forma arbitraria.
N=length(M(:,1));
K=length(M(:,1))-1;
cont=0;
for j=1:N
    cont=cont+1;
    W(j,j)=landa^(N-K);
    K=K-1;
end
```

```
teta = (M'*W*M)\(M'*W*Y');
Y_final=[ind(i-1,2),ind(i-2,2),ind(i-3,2),ind(i-4,2),ag(i,2),ag(i-1,2),ag(i-
2,2),ag(i-3,2),ag(i-4,2)]*teta;

consumo_x_aisl(i,2)=Y_final; %Se reconstruyen los datos para poder seguir
ind(i,2)=Y_final;            %disponiendo del regresor.

%%%%%%%%% descomentar para obtener landa óptimo%%%%%%%%%%%%%%%%%

%                 error_abs=abs(consumo_x_real(i,2)-Y_final);
%                 error_rel=abs(consumo_x_real(i,2)-
Y_final)/consumo_x_real(i,2);
%                                      if(error_min_abs(k,c2)>error_abs &&
i==(paprox(c2)-puntosb(1)))
%                                         error_min_abs(k,c2)=error_abs;
%                                         landa_min(k,c2)=landa;
%                                     end
%                                     if(error_min_rel(k,c2)>error_rel &&
i==(paprox(c2)-puntosb(1)))
%                                         error_min_rel(k,c2)=error_rel;
%                                         landa_min(k,c2)=landa;
%                                     end
%                                     if(i>paprox(c2)-puntosb(1))
%                                         c2=c2+1;
%                                     end
%                                 end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%

if(tramo==1 && isnan(consumo_x_aisl(i+1,2))==0) % Esto ocurre cuando el
siguiente dato ya no es NaN.
    alfa=(v_final-v_inicial)/sum(ind(c_inicial:c_final,2));
end
```

### 6.1.8 Aproximación Inicial.

En este código no se ha metido la modificación del consumo con el coeficiente alfa pero sólo hay que multiplicar cada consumo aproximado por alfa, es similar al caso de Procesos Gaussianos.

```
%%%%%%%%%%%% SE ENTRA EN ALGORITMO APROXIMACIÓN INICIAL %%%%%%%%%%%%%%%%%%

c1=1;
n=1;
while(isnan(consumo_x(i,2))==1 && i-7*4*n>0 && consumo_x(i,1)==consumo_x(i-
7*4*n,1) )
    if  (isnan(consumo_x(i-7*4*n,2))==0)
        consumo aprox=consumo x(i-7*4*n,2);
        consumo_x(i,2)=consumo_x(i-7*4*n,2);
 %Para ir una semana antes, se pone que el dato se aproxime por el tomado.
    end
    n=n+1;

end
n=1;
while(isnan(consumo_x(i,2))==1 && i+7*4*n<length(consumo_x) &&
consumo_x(i+1,1)==consumo_x(i+7*4*n,1))
    if  (isnan(consumo_x(i+7*4*n,2))==0)
        consumo_aprox=consumo_x(i+7*4*n,2);
        consumo_x(i,2)=consumo_x(i+7*4*n,2);
    end
    n=n+1;
end
if(tramo==1 && isnan(consumo_x_aisl(i+1,2))==0)
% Esto ocurre cuando el siguiente dato ya no es NaN.
    alfa=(v final-v inicial)/sum(ind(c inicial:c final,2));
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
```

### 6.1.9 Escaneo programa.

Este código es el que se encarga de encontrar los contratos que son válidos para el intervalo de puntos dado, es decir, si se quieren aislar 300 puntos es el que se encarga de verificar en qué contratos están esos 300 puntos, sin que falte ninguno, para después poder aislarlos y seguir haciendo la aproximación tal como se ha explicado a lo largo del proyecto.

```
function
[contratos_e]=escaneoprograma(x,x_max,dconfecha_Nan,puntosb,porcentaje)

% contratos_e = contratos válidos.
% x = contratos en los que se quieren buscar los puntos.
% dconfecha_Nan = datos con la fecha y con los NaN.
% puntosb = Intervalo de puntos para que un contrato se considere válido.
% porcentaje = % de consumo=0 que no debe superarse.

contratos_e=zeros(x_max,3);                 %Se inicializan los contratos.
candidato=0;
c1=0;                                  %Identificador del contrato que se está
buscando actualmente.
c2=1;
for i=1:length(dconfecha_Nan)           %Se hace un bucle desde i hasta el
máximo de los datos.
    if(dconfecha_Nan(i,1)>max(x) || c2>x_max)                       %Cuando
se llegue a un contrato que no se quiere escanear.
        fprintf('He llegado al máximo que quiero escanear\n')
        return
    end
    if(dconfecha_Nan(i,1)~=c1 && x(1)<=dconfecha_Nan(i,1))
% En caso de que esté en el contrato que se quiere escanear
        c1=dconfecha_Nan(i,1);
        cont=0;
        if(contratos_e(1,1)==0)          %Si no hay ningún contrato elegido.
            for y=i+puntosb(1)-1:i+puntosb(end)
% Bucle desde el punto inicial al final de la zona que se quiere escanear.
                if(y<length(dconfecha_Nan) && dconfecha_Nan(y,1)==c1)
% Se mira que "y" no sea mayor que el numero final de datos y que se siga en
el contrato actual
                    candidato=candidato+isnan(dconfecha_Nan(y,4));
% Si algún punto del intervalo es NaN candidato vale distinto de 0
                    if(dconfecha_Nan(y+1,4)-dconfecha_Nan(y,4)==0)
                        cont=cont+1;
%Cuenta los consumos=0
                    end
                    if(candidato==0)
                        inicio=i+puntosb(1)-1;
                        final=i+puntosb(end)-1;
                    end
                else
                    candidato=1;
%Si no se cumple lo anterior pongo candidato a uno y salgo del for con break.
                    contratos_e(c2)=0;
                    break
                end
            end

        else
```

```
            while(fecha_inicial-dconfecha_Nan(i,2)>0 &&
c1==dconfecha_Nan(i,1)) %Lo que hace este bucle while es encontrar la fecha
inicial una vez
                i=i+1;
% Contrato_e(1,1) es distinto de 0.
            end
            if(c1~=dconfecha_Nan(i,1))
                i=i-1;                          %Es i-1 porque del while se
sale cuando
                candidato=1;                    %se produce el cambio de
contrato, por tanto si se va una medida atrás se sigue
            end                                 %en el contrato anterior para
que en la siguiente iteración entre en el contrato nuevo.
            if(fecha_inicial-dconfecha_Nan(i,2)==0 && candidato==0)    %Si la
medida "i" se tomó en la fecha inicial del intervalo.

                for y=i:i+puntosb(end)-puntosb(1) %-1
                    if(y<length(dconfecha_Nan) && dconfecha_Nan(y,1)==c1)
                        candidato=candidato+isnan(dconfecha_Nan(y,4));
                        if(dconfecha_Nan(y+1,4)-dconfecha_Nan(y,4)==0)
                            cont=cont+1;
                        end
                    else
                        candidato=1;
                        contratos_e(c2,1)=0;
                        break
                    end
                end
                if(candidato==0)
                inicio=i;
                final=i+puntosb(end)-puntosb(1);
                end
            else
                candidato=1;
            end
        end
        if candidato==0
%En caso de que candidato valga 0 significa que no hay ningun NaN
            if(cont<(puntosb(end)-puntosb(1))*porcentaje/100)
%Si hay menos de "%" consumo=0.
                if(contratos_e(1,1)==0)
                    fecha_inicial=dconfecha_Nan(inicio,2);      %Si es el 1er
contrato que se elige, se guarda la fecha inicial en la que empieza el
intervalo para ese contrato.
                end
                contratos_e(c2,1)=c1;
                contratos_e(c2,2)=inicio;   %Si es válido se guarda en la
columna 2 de "c_x" el punto inicial de la zona válida
                contratos_e(c2,3)=final; %y en la tercera columna se guarda
el punto final.
                c2=c2+1;                                                %se
incrementa la componente del vector contratos_e
            end
        end
    end

    candidato=0;                                    %Se resetea candidato.
end
end
```

# REFERENCIAS

[1] Teodoro Alamo, Data Driven Systems Vector Uncertainty, 2017.

[2] Christopher M. Bishop, Pattern Recognition and Machine Learning, p. 78-88, 2006.

[3] Bárnabas Póczos, Introduction to Gaussian Processes, 2009. [En Línea], Disponible: https://www.cs.cmu.edu/~bapoczos/other_presentations/Introduction_GP_20_10_2009.pdf.

[4] Eduardo Morales, Procesos Gaussianos. [En Línea], Disponible: https://ccc.inaoep.mx/~emorales/Cursos/Aprendizaje2/Acetatos/gaussianprocesses.pdf.

[5] Daniel Rodríguez Ramírez y Teodoro Alamo Cantarero, Apuntes Ingeniería de Control: Identificación mediante el método de los mínimos cuadrados.

[6] Francisco Casellas, Guillermo Velasco, Francesc Guinjoan y Robert Piqué, El concepto de Smart Metering en el nuevo escenario de distribución eléctrica, p. 1. [En Línea], Disponible: https://upcommons.upc.edu/bitstream/handle/2117/9066/5025.pdf