



GRADO EN MATEMÁTICA COMPUTACIONAL

ESTANCIA EN PRÁCTICAS Y PROYECTO FINAL DE GRADO

Aproximación constructiva de funciones de varias variables mediante redes neuronales

Autor:
Alba CORREAS PASCUAL

Supervisor:
Francisco FALOMIR ARIAS
Tutores académicos:
Juan José FONT FERRANDIS
Marina MURILLO ARCILA

Curso académico 2018/2019

Resumen

En este escrito se detalla tanto la estancia en prácticas realizada en UBE CORPORATION EUROPE, como la capacidad de las redes neuronales para aproximar funciones continuas de varias variables definidas sobre un conjunto compacto.

En la primera parte se describe la estancia en prácticas. El propósito en la empresa fue desarrollar un programa de evaluación de datos almacenados en la base de datos de UBE para localizar los controladores que están actuando de manera incorrecta.

En la segunda parte del trabajo, correspondiente a la memoria del Proyecto Final de Grado se prueba la capacidad de aproximar funciones continuas de varias variables en \mathbb{R}^2 definidas sobre un compacto mediante redes neuronales activadas por funciones sigmoideas acotadas. Además, se presenta un ejemplo para ilustrar la utilidad de estos resultados.

Palabras clave

Red neuronal, función sigmoideal, función acotada, aproximación de funciones continuas.

Keywords

Neural network, sigmoidal map, bounded map, approximation of continuous maps.

Índice general

1. Estancia en prácticas	7
1.1. Introducción	7
1.2. Objetivos del proyecto formativo	7
1.3. Explicación detallada del proyecto realizado en la empresa	8
1.3.1. Conceptos previos	8
1.3.2. Tecnologías empleadas	9
1.3.3. Lenguajes de programación utilizados	10
1.3.4. Objetivo de la estancia	10
1.3.5. Etapas para el desarrollo de los algoritmos KPIs	12
1.3.5.1. Extracción y manipulación de datos	12
1.3.5.2. Creación de una vista	13
1.3.6. Desarrollo en R	15
1.3.6.1. Indicadores de rendimiento desarrollados	15
1.3.7. Planificación temporal de las tareas	25
1.3.7.1. Primera quincena	25

1.3.7.2.	Segunda quincena	25
1.3.7.3.	Tercera quincena	25
1.3.7.4.	Cuarta quincena	26
1.3.7.5.	Quinta quincena	26
1.3.7.6.	Sexta quincena	27
1.3.8.	Grado de consecución de los objetivos propuestos	27
2.	Memoria TFG	29
2.1.	Introducción	29
2.1.1.	Conceptos previos	32
2.2.	Aproximación de funciones de varias variables mediante redes neuronales	33
3.	Conclusiones	45
4.	Manual de uso de CPM	49
5.	Código CPM.exe	69

Capítulo 1

Estancia en prácticas

1.1. Introducción

En esta sección se detalla la memoria técnica de la estancia en prácticas realizada en UBE CORPORATION EUROPE, un grupo empresarial internacional que dispone de un complejo industrial próximo al Puerto de Castellón. El responsable de la supervisión de este proyecto fue Francisco Falomir Areas, jefe del departamento de Sistemas de Control.

1.2. Objetivos del proyecto formativo

En UBE CORPORATION EUROPE hay un complejo industrial de $300.000m^2$. En él se encuentra un cuantioso número de tanques en los que se introduce materia prima para crear productos. Los procesos de producción se dirigen mediante sistemas de control de lazo cerrado (PID), los cuales evalúan la diferencia entre el valor deseado y el valor dado. Uno de los componentes de los controladores son los elementos de control final (dispositivos que manipulan la temperatura, la presión, etc, como por ejemplo, válvulas de control, variadores de frecuencia, relés...).

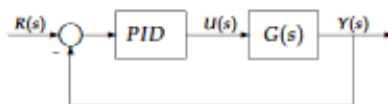


Figura 1.1: PID

Es casi inviable saber cuándo el rendimiento de un controlador está disminuyendo hasta que es demasiado tarde, ya que la productividad cambia tan levemente que pasa inadvertida.

El proyecto realizado en las prácticas tiene como finalidad localizar cambios en el rendimiento del controlador y evaluar el origen de los mismos, para así saber qué elemento de control final (válvulas de control, variadores de frecuencia, motores eléctricos, etc) está funcionando de manera incorrecta. Para ello, se calculan una serie de indicadores de rendimiento (KPI) cuyo análisis permite determinar si existe algún tipo de mal funcionamiento en alguno de los lazos de control.

Los mecanismos de control facilitan a los administradores todo lo necesario para averiguar, localizar y solventar problemas que afecten de forma nociva al rendimiento de cualquier lazo de control antes de que estos impliquen un periodo de parada disparatado. Estas herramientas pueden examinar varios controladores de una misma planta o, por el contrario, pueden ser concretas para un PID en particular.

Los KPIs programados han sido:

- AAE.
- MV Histogram.
- Noise Band.
- MV Reversals.
- Mode Changes.
- Set Point Changes.
- Percent Time In Abnormal.
- Stiction.
- Process Data.

1.3. Explicación detallada del proyecto realizado en la empresa

1.3.1. Conceptos previos

Varias ideas que hay que tener en cuenta para poder comprender el proyecto realizado en las prácticas son:

- **Controlador PID:** Un controlador PID (Controlador Proporcional, Integral y Derivativo) es un mecanismo de control simultáneo por realimentación ampliamente usado en sistemas de control industrial. Este calcula la desviación o error entre un valor medido y un valor deseado.
- **KPI:** Un KPI (key performance indicator), conocido también como indicador clave o medidor de desempeño o indicador clave de rendimiento, es una medida del nivel del rendimiento de un proceso. El valor del indicador está directamente relacionado con un objetivo fijado previamente y normalmente se expresa en valores porcentuales.
- **Elemento de control final:** Un elemento final de control es un mecanismo que altera el valor de una variable manipulada en respuesta a una señal de salida desde el dispositivo de control automático: típicamente recibe una señal del controlador y manipula un flujo de material o energía para el proceso. El elemento final de control puede ser una válvula de control, variadores de frecuencia y motores eléctricos, un relé, un amortiguador, etc.

1.3.2. Tecnologías empleadas

Para la implementación de este proyecto se han utilizado las siguientes tecnologías:

- **Base de datos:** Es un sistema que permite acumular grandes cantidades de datos de cualquier tipo en tablas de forma estructurada con diferentes propósitos y usos a los que luego podremos acceder fácilmente.
- **Exaquantum:** Es un sistema de gestión de bases de datos diseñado para plantas industriales (PIMS) disponible en las industrias de procesos. Es capaz de obtener datos de todas las partes de un proceso y convertirlos en información fácilmente utilizable. Además, estos registros son distribuidos sencillamente en toda la empresa y se puede acceder a ellos desde múltiples sistemas de control de procesos.
- **Microsoft .NET:** Es una plataforma para el desarrollo del software que consta de un conjunto de mecanismos y técnicas que posibilitan la construcción de aplicaciones. Estas herramientas simplifican las tareas más frecuentes del programador. Una de las ventajas más significativas de Microsoft .NET es que es multi-lenguaje y el desarrollador no necesita aprender un lenguaje específico para poder codificar aplicaciones.
- **Microsoft Excel:** Es un software que facilita el desarrollo de tareas contables y financieras. Existe la posibilidad de crear tablas, gráficos, operaciones matemáticas, funciones matemáticas, etc.

1.3.3. Lenguajes de programación utilizados

- ***R Project for Statistical Computing***: R es un software libre pensado para la computación estadística y la creación de gráficos, por lo que se trabaja con conjuntos de datos que se transforman en matrices para poder hacer analíticas sobre ellos. En el entorno de programación R también figuran multitud de librerías y paquetes. En ellos hay colecciones de funciones, datos y código R que implementan la funcionalidad base de este lenguaje. Fue creado por Ross Ihaka y Robert Gentleman en la universidad de Oakland, Nueva Zelanda.
- ***SQL***: Es un sistema para la gestión de bases de datos producido por Microsoft. Es un lenguaje de programación estándar para manejar información desde una base de datos relacional (distintas entidades o tablas). Este entorno nos permite realizar un mantenimiento de la base de datos, es decir, crear, insertar, modificar, eliminar y consultar registros. Para poder manipular el lenguaje SQL se emplea un sistema gestor de base de datos (MySQL, SQL Server, ORACLE, etc).

1.3.4. Objetivo de la estancia

Una planta industrial es una fábrica donde se genera materia prima para transformarla en producto mediante maquinaria industrial. Algunos de los artefactos que se utilizan en el sector de la industria son tanques, en los cuales se produce parte de la modificación de la materia prima para luego elaborar el producto.

Aunque se haya hecho un buena elaboración de la maquinaria de la planta, cabe la posibilidad de que surjan perturbaciones o errores que afecten a la elaboración de los productos. Por ejemplo, se pretende que un líquido que se ha introducido en un tanque a una temperatura x salga a otra temperatura y , pero, por el exceso de perturbaciones, no se consigue el objetivo.

Para poder regular determinados procesos, disminuir las posibilidades de error y obtener los propósitos deseados, por ejemplo, que el líquido alcance la temperatura y , se implantan los sistemas de control de lazo cerrado (PID). Estos mantienen una relación entre una variable de salida y otra de entrada, es decir, la señal de salida tiene efecto sobre la acción de control. Como se trata de un sistema con realimentación la salida dada por el proceso se utiliza como una de las entradas del sistema, dando lugar a una señal de salida diferente, que actúa de la misma manera que la salida anterior.

A continuación se muestra un ejemplo del sistema de control de lazo cerrado y una explicación de cada una de sus componentes:

- **Señal de referencia**: Valor que se desea obtener al final del proceso.

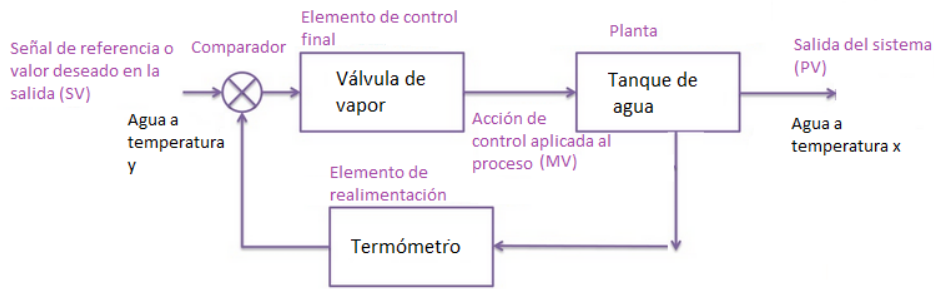


Figura 1.2: *Sistema de control de lazo cerrado*

- **Comparador o controlador:** Instrumento encargado de comparar el valor deseado con el valor de realimentación, es decir, el valor dado la salida del sistema.
- **Elemento final de control:** Instrumentos que reciben la señal del controlador y ejercen cambios en el proceso transformando los parámetros hacia el valor deseado (MV).
- **Planta:** Lugar donde se produce la acción de control.
- **Elemento de realimentación:** Herramientas que permiten el control automático de un sistema.
- **Salida:** Es el verdadero valor que toma la señal que queremos controlar en cada instante.

Cada tanque lleva insertado un lazo de control de bucle cerrado. Estos pueden actuar de tres maneras diferentes, en modo automático, cascada o manual. Los dos primeros son automáticos y operan continuamente, sin interrupción y sin intervención externa, mientras que en el último modo se necesita un operador que esté introduciendo manualmente la variable de retroalimentación del sistema.

Cada lazo de control PID guarda registros por segundos en una base de datos. Estos registros son los siguientes:

- **SV (Setpoint Value):** Se trata de la señal de referencia, el valor que deseamos obtener al final del proceso, por ejemplo, el agua a temperatura y .
- **MV (Manipulated Value):** Señal que resulta de la transformación efectuada por el elemento de control final para que los parámetros se aproximen al valor deseado.
- **PV (Process Value):** valor medido por el sensor, en nuestro caso, el termómetro y salida dada por el sistema. Por ejemplo, el agua a temperatura x .

- **PVL:** Valor máximo aproximado que puede alcanzar la salida del sistema.
- **PVH:** Valor mínimo aproximado que puede alcanzar la salida del sistema.
- **MVL:** Valor máximo aproximado que puede alcanzar la salida dada por el controlador.
- **MVH:** Valor mínimo aproximado que puede alcanzar la salida dada por el controlador.
- **Mode:** Modo en que se encuentra el lazo de control. Los PIDs que funcionen de manera correcta se deben encontrar en modo automático o cascada (sin intervención externa).

En una planta industrial hay un número muy elevado de tanques, por lo tanto, de PIDs y en consecuencia, de registros. Esto dificulta el poder detectar cuándo alguno de estos sistemas está fallando. Por ello, existen indicadores de rendimiento de los procesos (KPIs). Mediante estos algoritmos podemos detectar a tiempo qué sistema de control está actuando de manera equívoca.

1.3.5. Etapas para el desarrollo de los algoritmos KPIs

1.3.5.1. Extracción y manipulación de datos

Para cada lazo de control extraemos los datos de un día con frecuencia de 1 minuto. Por lo tanto, por cada lazo y por cada día tenemos 1440 valores de cada registro (SV, MV, PV, MVL, MVH, ...). Se realizan los cálculos para cada lazo de control por día, es decir, se seleccionan todos los datos de un día, se manipulan y se guardan los resultados y así se consigue tener un registro por días, meses o años, a elección del usuario. Los resultados de estas herramientas quedan plasmados en gráficas y tablas. Esto facilita la visualización de los problemas de mal funcionamiento que hacen que el rendimiento del controlador PID baje.

- **Almacenamiento de datos en UBE:** Exaquantum es un sistema de gestión de base de datos donde se encuentra toda la información de toda la planta de UBE. Los lazos de control están organizados por unidades y áreas. Se almacenan datos con frecuencia de un minuto, es decir, en un día hay 1440 minutos, y en total hay 1226 lazos de control, por lo que tendríamos un total de 1.765.440 registros por cada día, es decir, 1440 registros por cada lazo de control.
- **Base de datos donde se almacenan los resultados:** Se crea una base de datos llamada CPM, con dos tablas, RAW Y RESULTS. En la primera, se almacenan todos los registros necesarios para calcular los KPIs correspondientes; estos se extraerán de Exaquantum. En la segunda, se almacenarán los resultados de los KPIs programados.

- Almacenamiento de datos en CPM:** Se desarrolla una aplicación de consola .NET llamada UBECPM.exe. Por cada ejecución de ésta, se accede a Exaquantum y se vuelcan por día todos los registros necesarios de los lazos de control organizados por unidades en RAW, es decir, un total de 1.765.440 datos por cada ejecución de la aplicación de consola. Una vez RAW tiene toda la información de un día, la aplicación de consola ejecuta un programa desarrollado en R, que accede a la tabla RAW, extrae la información de ésta, programa los algoritmos KPIs y vuelca los resultados en la tabla RESULTS. Una vez obtenidos los resultados de un día, UBECPM.exe le pregunta al usuario si desea calcular los resultados de los lazos de control del día siguiente. Si es así, la aplicación de consola mira la última fecha almacenada en RESULTS, calcula la siguiente fecha, vacía la tabla RAW para que no se sature de información e introduce los 1.765.440 registros del siguiente día en RAW.

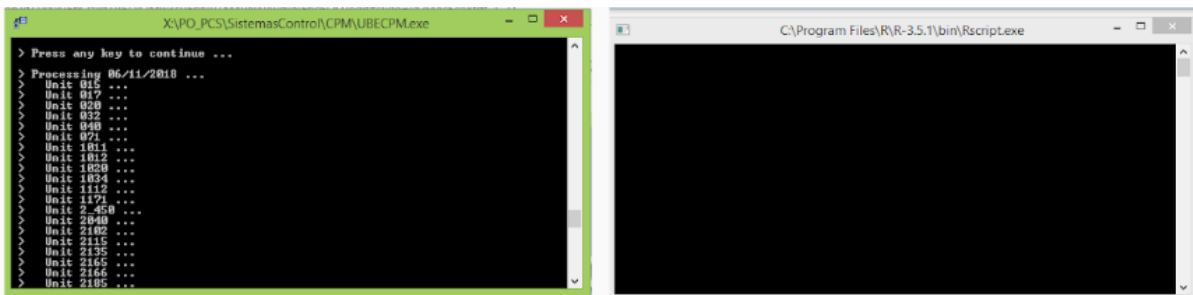


Figura 1.3: *UBECPM.exe*

1.3.5.2. Creación de una vista

Se crea una vista de la tabla RESULTS en una hoja Excel, en la que se pueden visualizar todos los resultados obtenidos en forma de tablas y gráficas de cada uno de los algoritmos programados.

- Hipervínculos**

En esta parte se da la opción de acceder a la sección deseada (resumen de todo, cualquier KPI o todos ellos juntos).

- Segmentación de datos**

Para cada indicador de rendimiento se da la opción de visualizar los datos por controlador, área o unidad. Para seleccionar cualquiera de las tres opciones, solo se tiene que escoger el

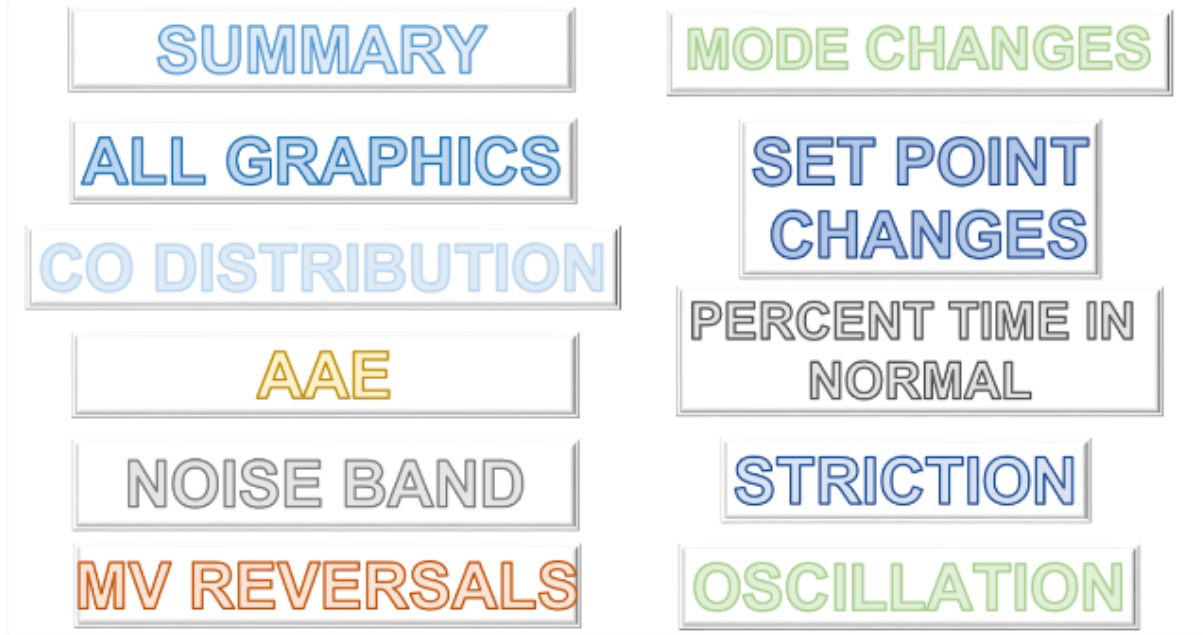


Figura 1.4: *Hipervínculos*

tag (lazo de control), el área o la unidad deseada y pulsar sobre ella. También es posible seleccionar varios controladores, áreas o unidades. Por ejemplo, en la imagen siguiente se puede ver cómo se han seleccionado cinco lazos distintos.

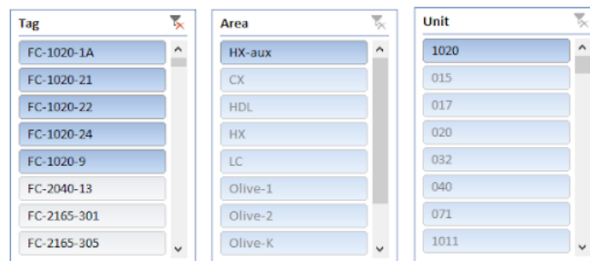


Figura 1.5: *Segmentación de datos*

- **Escala de tiempo**

La escala de tiempo permite escoger la etapa de la se desea visualizar los resultados; para ello basta con escoger el periodo de tiempo (años, trimestres, meses o días).



Figura 1.6: *Escala de tiempo*

■ Resumen

En este apartado, se proporciona un resumen por unidades o áreas de los KPI's empleados en el rango de fechas deseado.

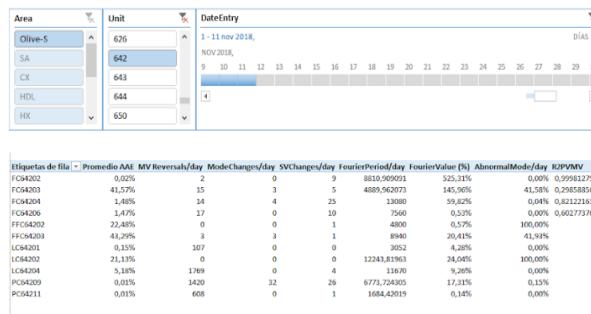


Figura 1.7: *Resumen*

1.3.6. Desarrollo en R

En primer lugar, RScript.exe se conecta a la base de datos CPM, accede a la tabla RAW, se descarga los registros almacenados en ésta y los guarda en un objeto de R llamado *dataframe* (hoja de datos). Esta información se manipula para desarrollar los indicadores de rendimientos y una vez calculados los resultados, se introducen en la tabla RESULTS.

1.3.6.1. Indicadores de rendimiento desarrollados

- **MV HISTOGRAM:** Este KPI muestra la distribución estadística de la salida del controlador (MV), es decir, determina el rango en el que el elemento de control final está actuando, y por tanto permite detectar si está saturado. Esto ocurre cuando opera habitualmente en el mismo rango, o cuando se desplaza excesivamente. La representación de los datos de esta herramienta se ha realizado mediante un histograma de la salida del

lazo de control, en el que quedan reflejadas las posiciones del elemento de control y la frecuencia en cada uno de los intervalos.

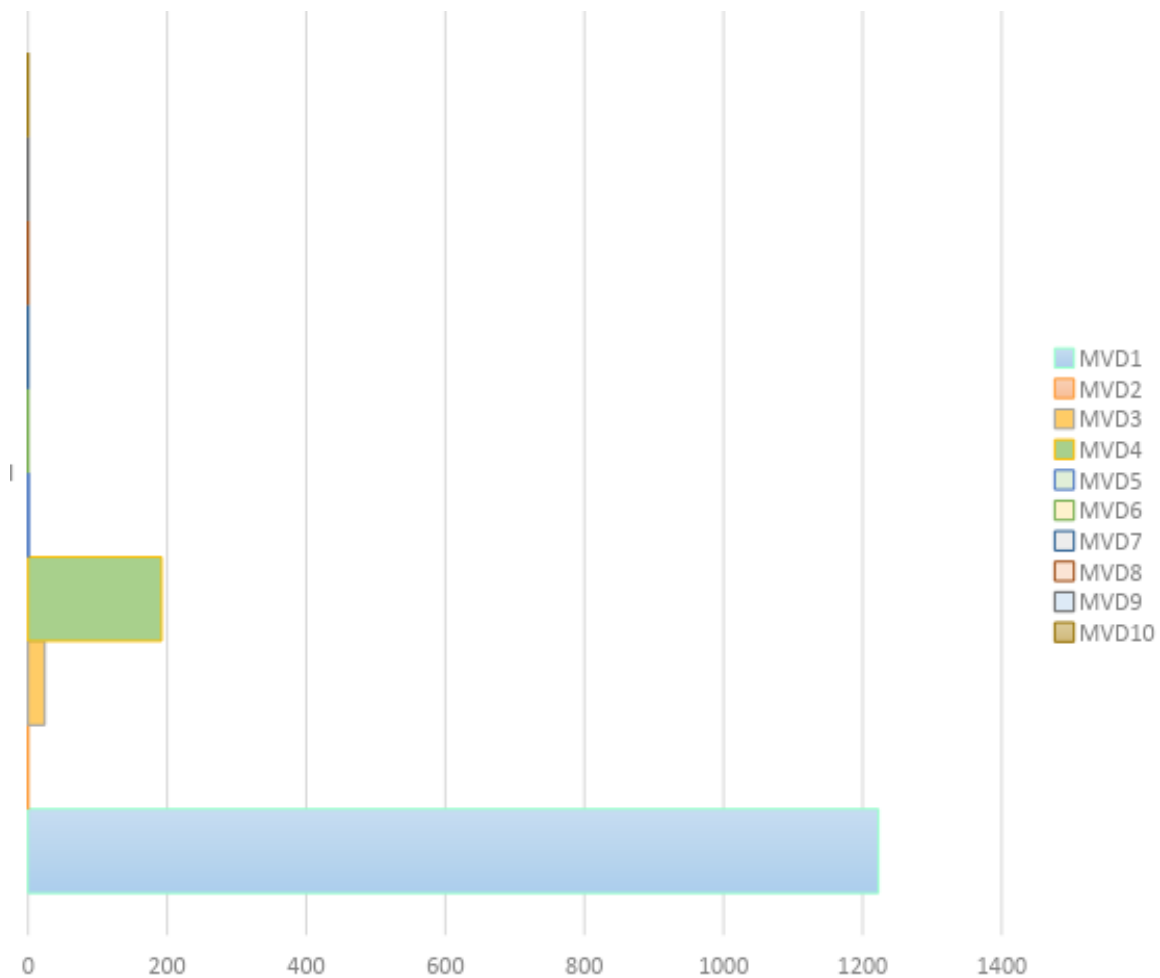


Figura 1.8: De los 1440 valores del controlador TC-433-14 registrados el día 1 de Noviembre, tenemos 1210 oscilaciones entre los valores 0-10, una media de 190 datos, oscilando entre los valores 30-40 y unos 40 valores entre 20-30.

- **AAE (Absolute Average Error):** Determina la diferencia entre el valor deseado en la salida (SV) y el valor real del sistema (PV). Realiza un seguimiento de la capacidad del controlador para mantener el punto de consigna (valor de la variable controlada al que se le ajusta un dispositivo de control). Cuanto más se aproximen los valores resultantes a cero, menos distarán los valores SV y PV entre sí, por lo que mejor mantendrá el controlador el punto de consigna.

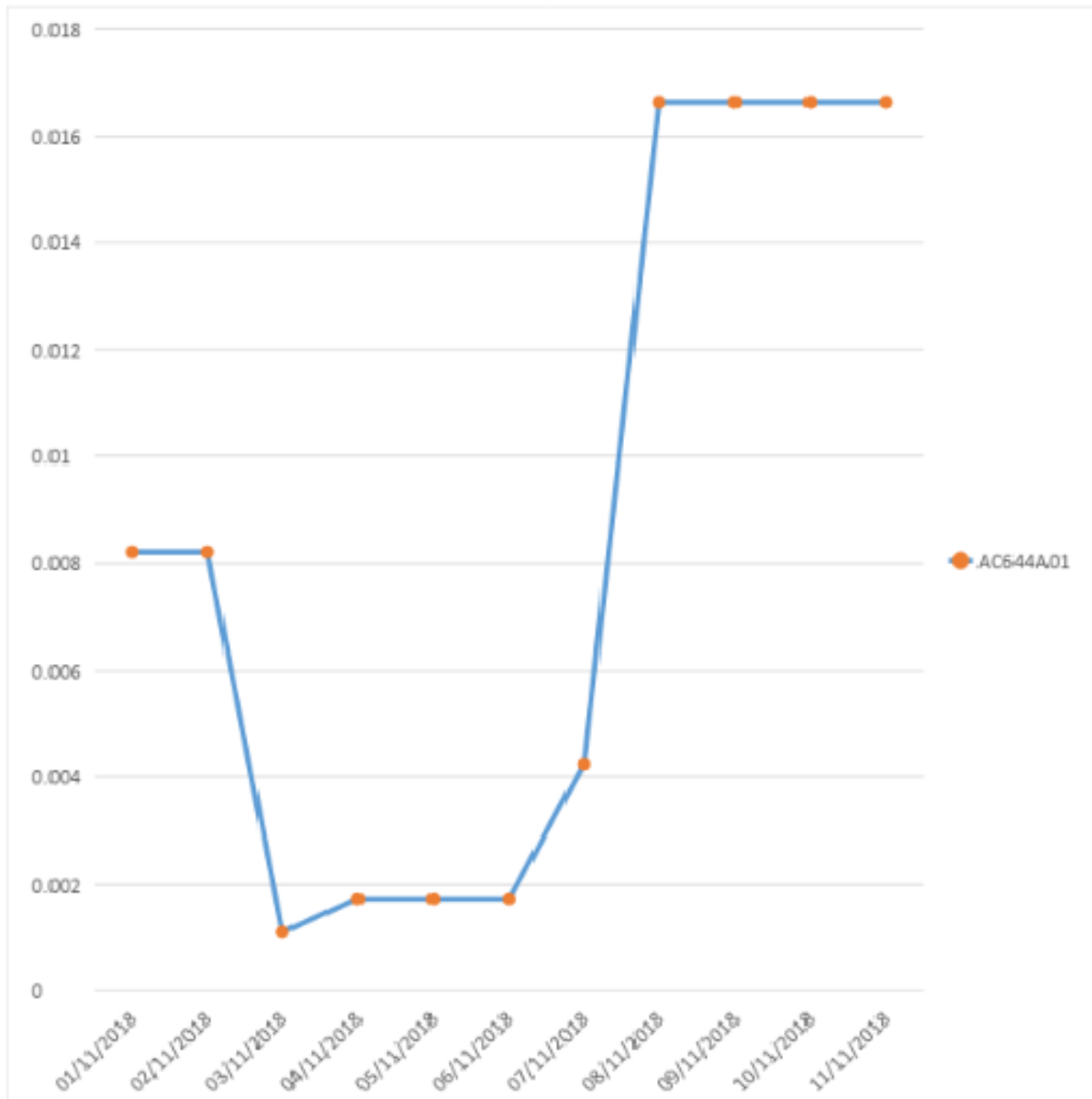


Figura 1.9: Se puede observar una buena actuación del lazo de control AC644A01 del día 1 al 11 de Noviembre, ya que el AAE se aproxima a 0, aunque los últimos días está actuando de manera más irregular.

- **Noise Band:** Este KPI se basa en el cálculo del ruido de alta frecuencia, ya que un ruido elevado podría indicar un fallo en el sensor. En la gráfica 2.10 se puede ver el valor promedio del PV y la desviación típica relativa a este promedio. Si la banda es ancha, se podría deducir un ruido elevado del sistema.

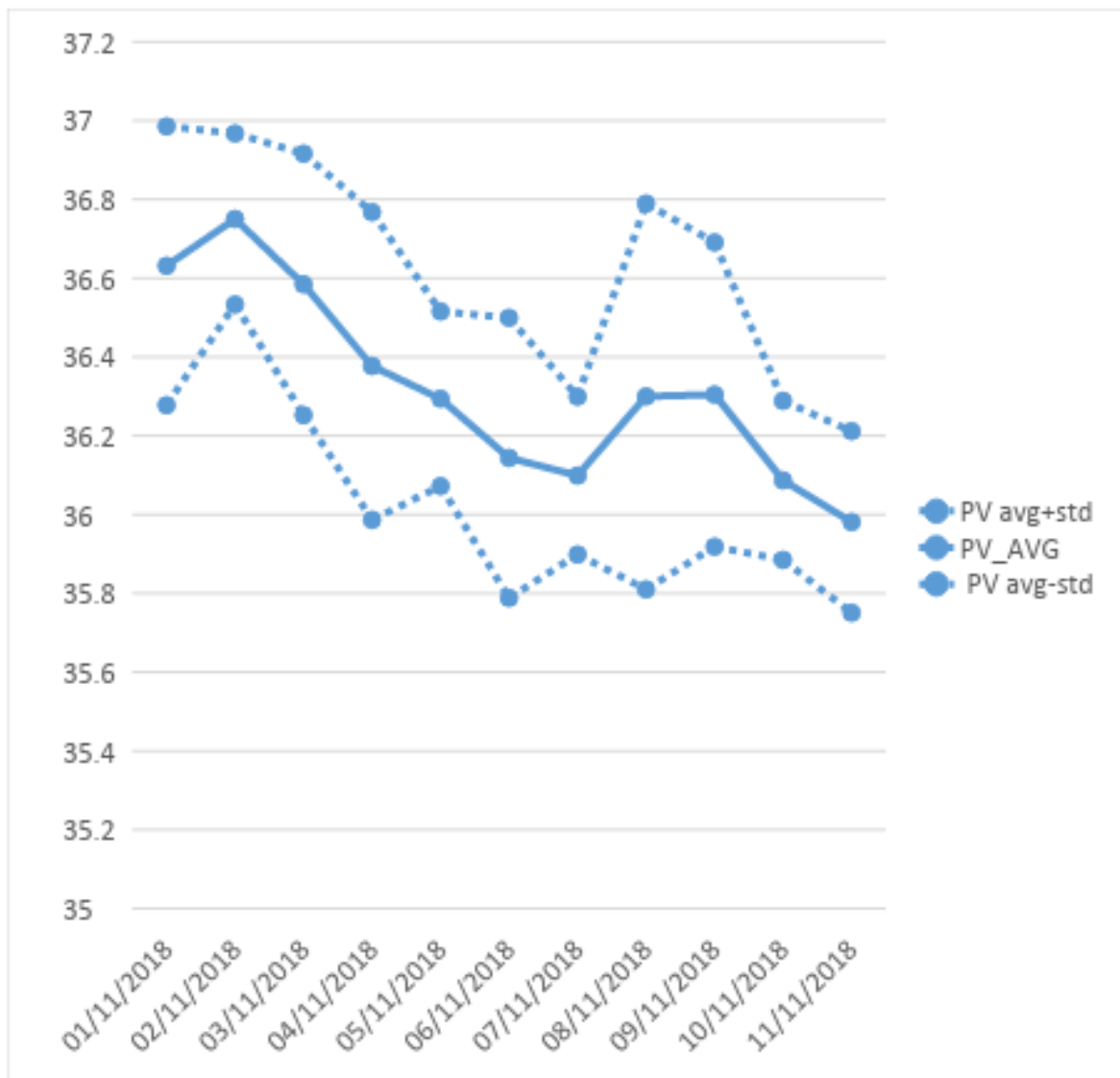


Figura 1.10: Como la franja no es muy ancha, no hay un ruido elevado.

- MV Reversals:** Este KPI enumera la cantidad de veces que la salida del lazo de control (MV) cambia de dirección (ascendente o descendente). Un gran número de inversiones indica que el lazo de control realiza cambios abruptos en la salida o bien amplifica mucho el ruido. La representación de los datos resultantes de este KPI se muestra a través de una gráfica en la que se plasma la cantidad de inversiones que hay en la fecha deseada.

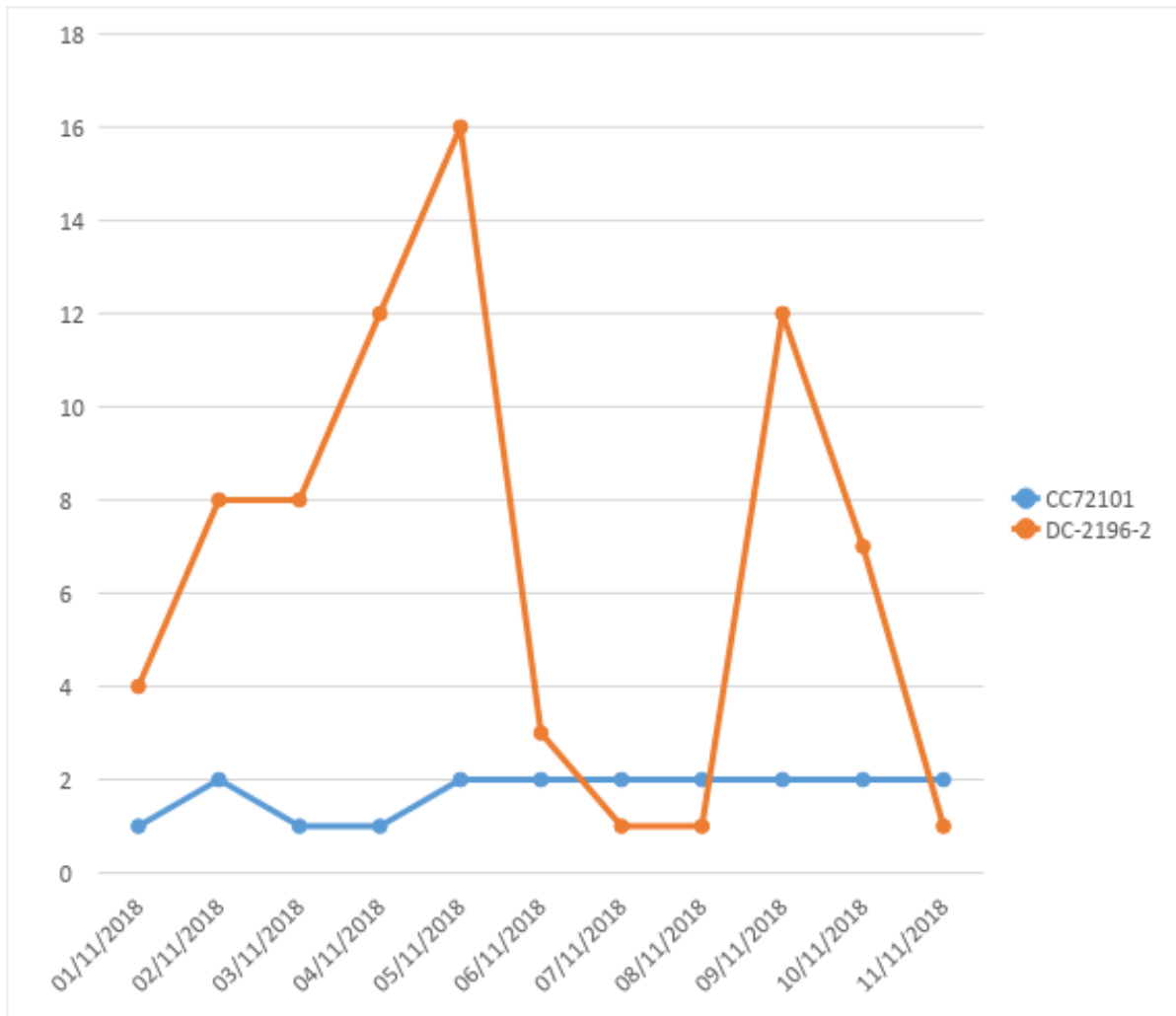


Figura 1.11: Podemos ver que el lazo de control CC72101 actúa de manera más adecuada que el lazo DC-2196-2.

- Mode Changes:** Los lazos de control PID tienen asignado un modo de actuación (automático, cascada, manual ...). La alteración de éste de manera continuada equivaldría a problemas potenciales. Mode changes cuenta las veces que cambia el modo en un rango de fechas deseado. Un número elevado de cambios indica que el lazo no funciona correctamente en modo automático de forma continuada.

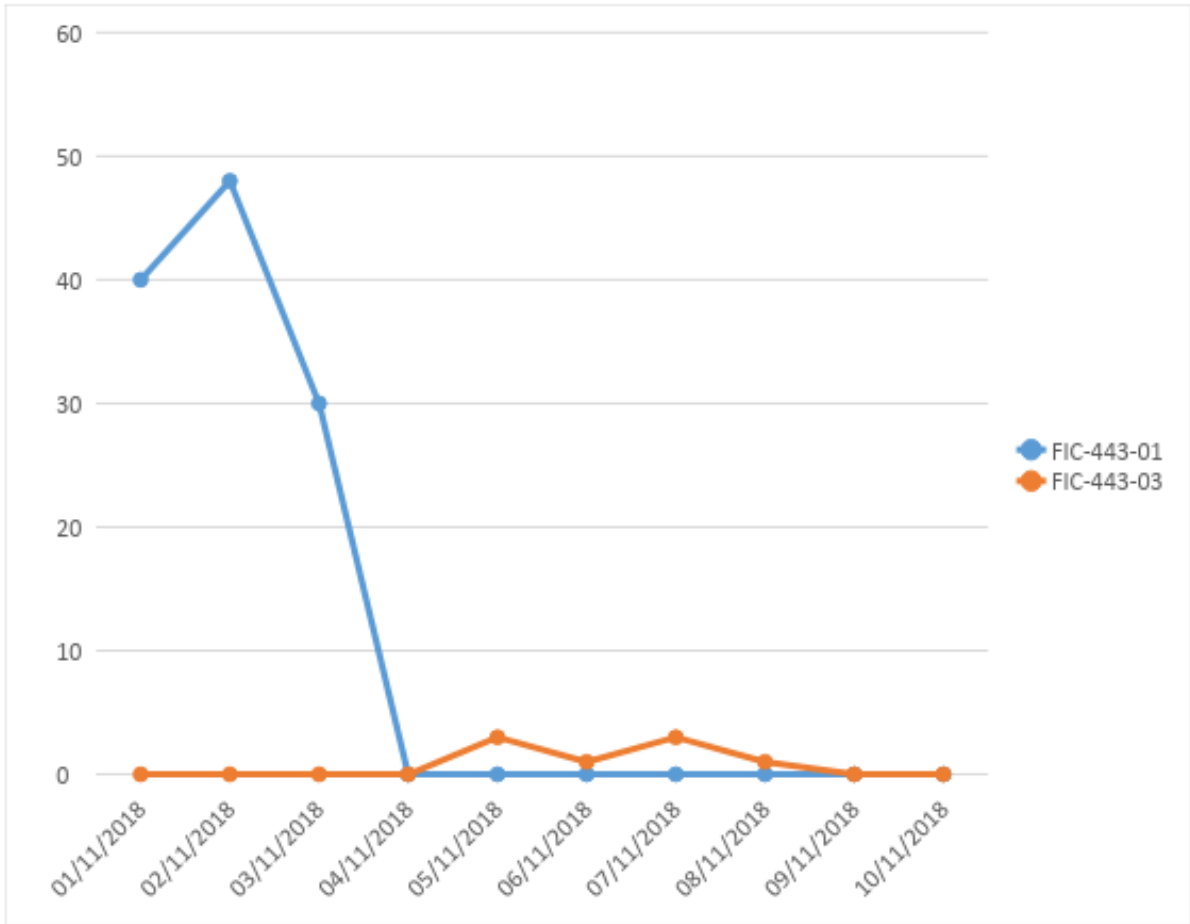


Figura 1.12: Como se puede observar en los primeros días el lazo FIC-443-03 funcionaba mejor que el lazo FIC-443-01.

- **Set Point Changes:** Una gran cantidad de cambios en el valor de la señal de referencia indicaría una intervención abundante del operador en el proceso de control. Set Point Changes se encarga de contar el número de veces por día que el punto de consigna varía.

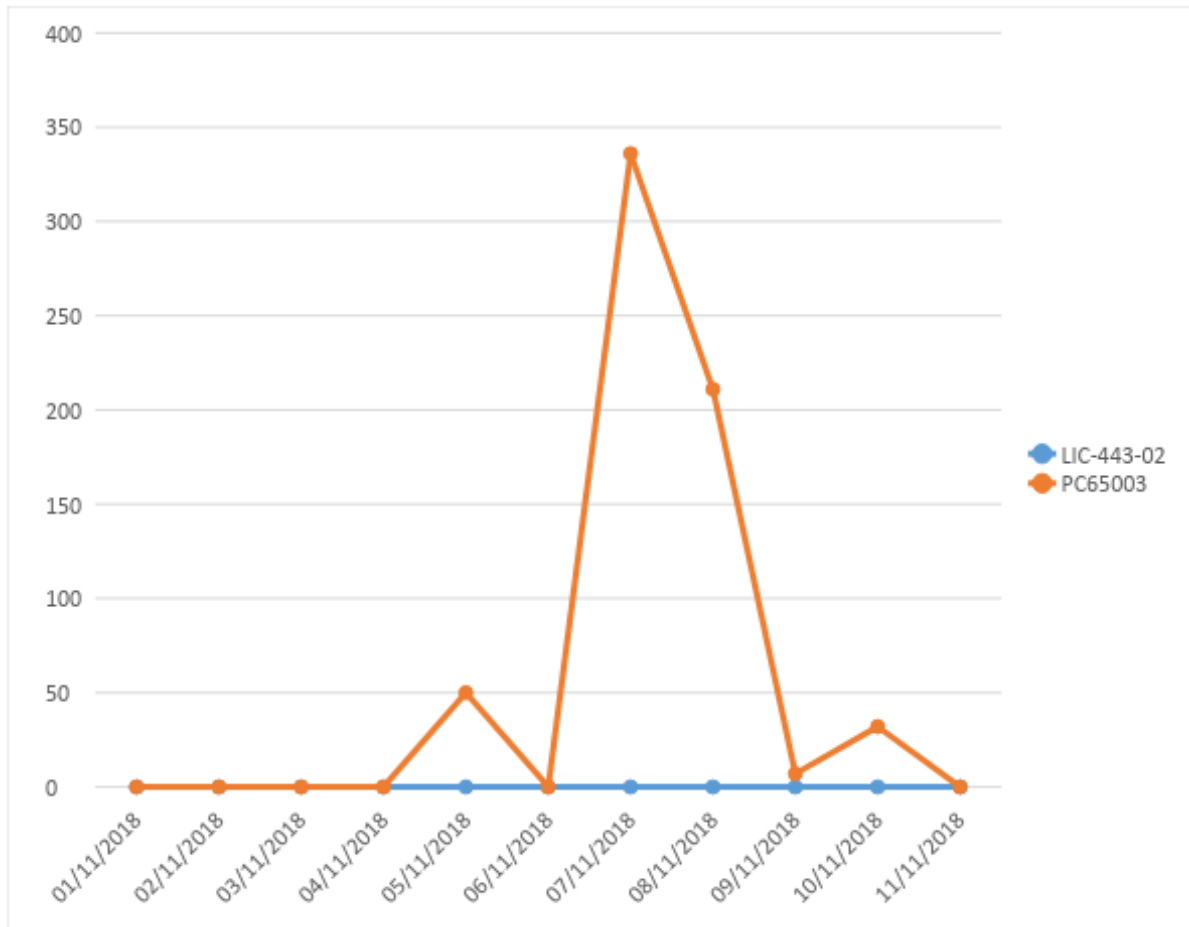


Figura 1.13: Los días 7 y 8 de Noviembre el operador intervino de manera excesiva en el control del proceso PC65003.

- Percent Time In Abnormal:** Un lazo de control PID está diseñado para funcionar en los modos automático o cascada. Percent Time In Abnormal evalúa el porcentaje de tiempo en el que el controlador está en modo distinto de cascada o automático.

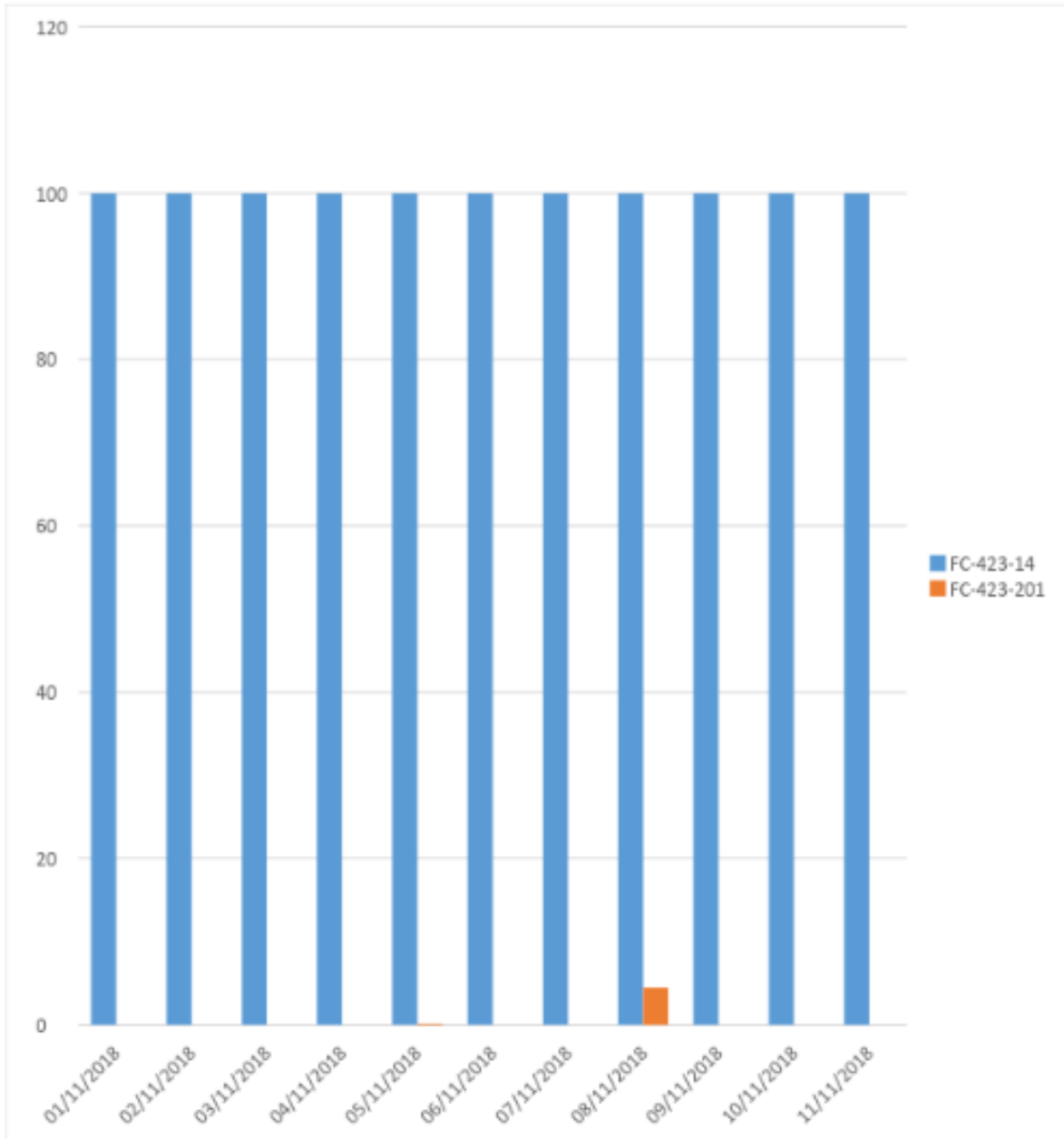


Figura 1.14: El lazo FC-423-14 pasa el 100 % de su tiempo en un modo distinto de cascada o automático, por lo que aparenta tener problemas potenciales.

- **Stiction:** Los componentes de los elementos de control finales pueden desgastarse con el tiempo, generando problemas en el movimiento de las válvulas. Este indicador evalúa este desgaste. Se calcula sólo para los lazos de control de caudal ya que generalmente existe

una correlación lineal entre la posición de la válvula y el caudal trasegado. Este indicador calcula el coeficiente R^2 de la correlación de modo que si se aprecia un decremento a lo largo del tiempo, se puede intuir una pérdida de linealidad y por tanto un deterioro del elemento.

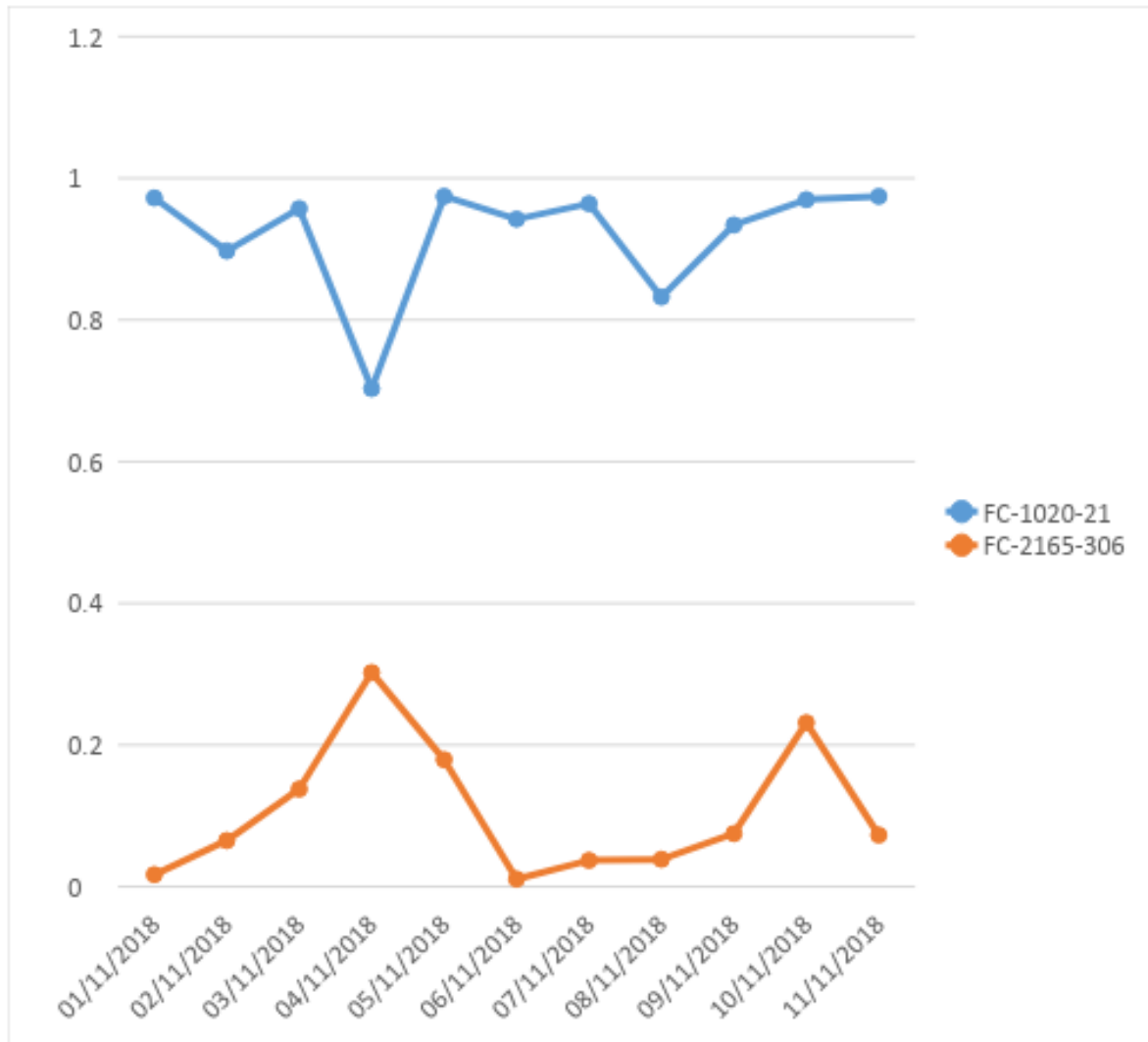


Figura 1.15: En la figura se aprecia una baja correlación en el caso del lazo de control FC-2165-306 ya que se aproxima más a cero.

■ **Process Data:**

En este apartado podemos observar el promedio, la desviación típica y los límites de la señal de referencia (SV) y de la salida del sistema (PV), es decir, nos proporciona una distribución estadística dentro del rango.

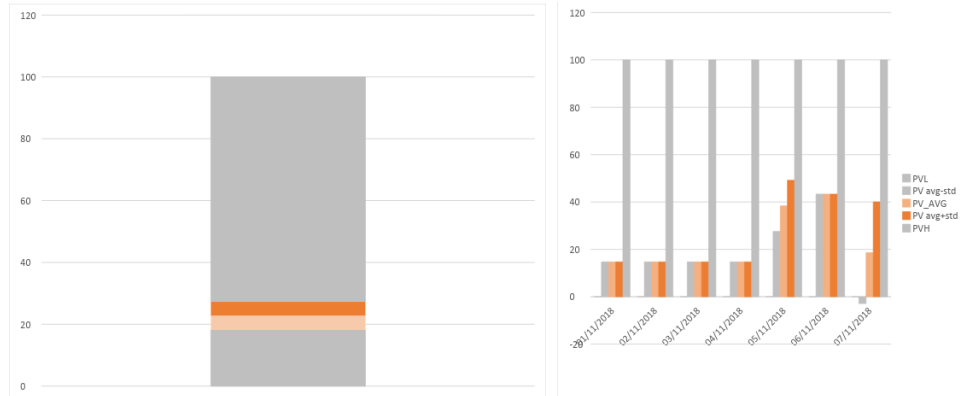


Figura 1.16: *Salida del sistema (PV)*

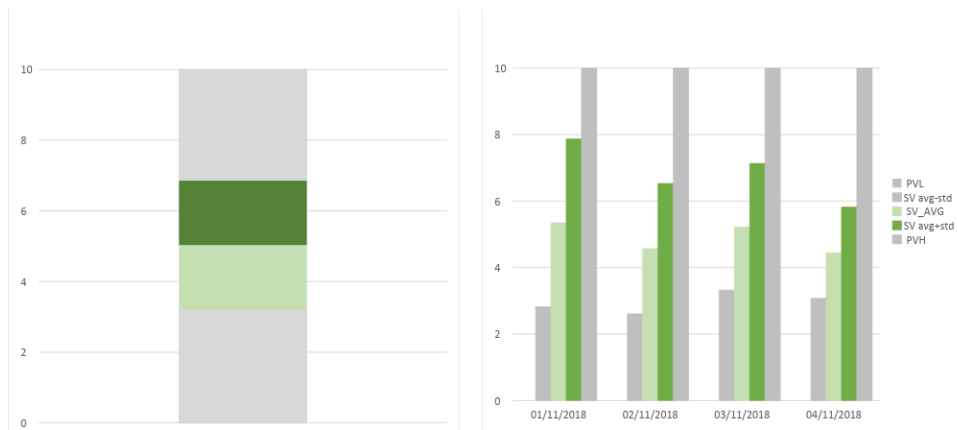


Figura 1.17: *Mediante los gráficos podemos ver si tanto el PV como el SV varían demasiado. Para un rango de fechas, en el primer gráfico se muestra la media de todos los datos durante todo el rango deseado, mientras que en el segundo, se muestra una media de los datos por cada día.*

1.3.7. Planificación temporal de las tareas

1.3.7.1. Primera quincena

Durante la primera quincena se ha mostrado y explicado el proyecto a llevar a cabo durante la estancia en prácticas. Para ello, se procede a leer apuntes sobre una asignatura del grado en Ingeniería Industrial llamada "Sistemas Automáticos", que tratan sobre un conjunto de componentes físicos conectados o relacionados entre sí, de manera que regulan o dirigen su actuación por sí mismos mediante reguladores de control (P, PI o PID). Una vez se han leído y entendido los apuntes de dicha asignatura y para poder acceder a la base de datos en la que se encuentra la información necesaria, se realiza una conexión SQL en el lenguaje de programación R.

1.3.7.2. Segunda quincena

Una vez extraída de la base de datos toda la información necesaria, se han programado los siguientes KPIs:

- **AAE (Absolute Average Error):** se trata de la resta, en valor absoluto, de la salida dada y la entrada deseada, para así saber cuanto dista el valor que queremos conseguir con el valor que realmente tenemos.
- **MV Histogram:** este algoritmo mide la frecuencia con la que aparece cada valor en la salida del controlador PID. Con esta información podemos saber si el elemento de control final está actuando de manera errónea y por lo tanto, detectar si está averiado.
- **Mode Changes:** cuantifica el número de veces en las que un lazo de control cambia de modo. Si este varía demasiado en un periodo de tiempo corto es porque el lazo está dañado.
- **Noise Band:** se basa en el cálculo de la desviación típica de la salida del sistema (PV) por minuto. Si ésta es elevada, indica la presencia de ruido de alta frecuencia y por lo tanto, un fallo en el elemento de realimentación del sistema, en nuestro ejemplo el termómetro (figura 2.2).

1.3.7.3. Tercera quincena

Durante la tercera quincena se han acabado de programar todos los algoritmos necesarios para localizar cambios en el rendimiento del controlador.

El elemento de control final transforma los parámetros para que éstos se aproximen lo máximo posible a los valores que deseamos. Los resultados de esta transformación tendrían que ser aproximadamente constantes, por lo que uno de ellos cuenta el número de veces que el elemento de control final invierte el resultado de las transformaciones. Si el valor es elevado indicaría un fallo en el lazo de control.

La señal de referencia (SV) es la que se introduce en el lazo de control. Como los PID's son automáticos, el valor del SV siempre es aproximadamente el mismo, pero en el momento en que cambia es porque un operador lo está cambiando manualmente y una intervención elevada de un operador en un lazo de control equivaldría a un error en el mismo, por lo que otro algoritmo se encarga de contar la veces por día que el punto de consigna varía.

Otra manera de saber si el operador interviene de manera excesiva es contar cuántas veces al día está el lazo de control en modo manual y no en modo automático o cascada.

Para saber si el elemento de control final, como por ejemplo una válvula, está actuando de manera equívoca, se calcula el coeficiente R^2 entre la salida del sistema y la salida dada por el elemento de control final, ya que existe una correlación lineal entre la posición de la válvula y el caudal trasegado. Si este coeficiente se aproxima a cero puede que el elemento de control final se esté deteriorando.

1.3.7.4. Cuarta quincena

Una vez programados todos los métodos anteriores, se ha creado una base de datos llamada CPM en la que se encuentran dos tablas denominadas RAW y RESULTS. En la primera se han almacenado los registros necesarios para el desarrollo de los algoritmos en los distintos días que se requiera, para así no tener que acceder a la base de datos Exaquantum. En la segunda se han recogido todos los resultados de los mismos. Además se ha convertido todo el código R en un ejecutable (.exe) para poder calcular el resultado de los distintos algoritmos cuando se desee y con los datos del día que se requiera. Después de esto, se han realizado todo tipo de pruebas para poder localizar errores en el desarrollo de los KPIs y solventarlos.

1.3.7.5. Quinta quincena

En la quinta quincena, se ha creado una vista de la tabla RESULTS en una página Excel con varias pestañas, una para cada método programado, además de un resumen de todos los datos. En cada una de éstas se han elaborado una tabla, una gráfica dinámica, una escala de tiempo y una segmentación de datos por lazo de control para poder filtrar y visualizar la información que el usuario desee. Una vez acabada la página Excel con el resultado de todos los algoritmos, se

han redactado dos manuales, uno en español adjuntado al final de la memoria de las prácticas y otro en inglés en los que se expone de manera detallada la función de cada pestaña y la interpretación de toda la información que ésta refleja.

1.3.7.6. Sexta quincena

En las últimas dos semanas se ha desarrollado una aplicación de consola .NET que relaciona la base de datos Exaquantum con las dos tablas y el código R. Así pues, la aplicación permite ejecutar el programa desarrollado en R en el intervalo de fechas deseado y después interpretar los resultados en la vista de la página Excel. Después de esto se ha puesto en práctica todo lo realizado en prácticas, analizando cada lazo de control y los resultados de los algoritmos.

1.3.8. Grado de consecución de los objetivos propuestos

La finalidad de las prácticas ha sido desarrollar un programa ejecutable en el lenguaje de programación R, que permite, mediante multitud de registros e información procedente de controladores instalados en los bidones que contienen materia prima para crear producto en la planta de UBE, averiguar cuál de ellos está dañado y actúa de forma errónea.

Para ello, se han programado una serie de indicadores que muestran cambios en el rendimiento de los lazos de control cerrados. Para realizar esta parte del proyecto se empleó una tercera parte del tiempo, ya que, una vez desarrollados, se tuvieron que realizar pruebas y solventar fallos notables.

Una vez acabado el programa en R con todos los KPIs programados, se desarrolló el ejecutable R, y se creó la aplicación de consola .NET. De esta forma, el usuario podría acceder a los resultados de todos los algoritmos en la fecha deseada.

Después de todo, se creó una vista en una página Excel para poder interpretar los resultados, averiguar y localizar los problemas que afectan al rendimiento de cualquier lazo de control y se redactó un manual para el usuario, tanto en inglés como en español, para poder utilizar y comprender todos los resultados dados por las gráficas y las tablas de la vista.

En definitiva, se cumplieron todos los objetivos planeados, y aunque ya había programado en el lenguaje R en la carrera, aprendí nuevas aplicaciones y funciones, además de reforzar mi conocimiento en el desarrollo mediante SQL. La estancia en prácticas también me ha servido para aprender la disciplina de trabajo en una empresa y ampliar mis conocimientos sobre gestión de bases de datos.

Capítulo 2

Memoria TFG

2.1. Introducción

Las redes neuronales, también llamadas “redes de neuronas artificiales”, son modelos bastante simplificados de las redes de neuronas que forman el cerebro, y, al igual que éste, intentan “aprender” a partir de los datos que se le suministran.

Las redes neuronales se caracterizan porque, por una parte, pueden ser entrenadas para efectuar alguna labor en concreto y, por tanto no es necesario realizar ningún estudio previo para poder desarrollar un programa informático. Además si existe alguna nueva necesidad en la tarea que han de realizar, no hace falta reescribir el código, ya que pueden volver a entrenarse para adaptarse al nuevo requisito. Por otra parte, al actuar como el cerebro humano, una vez acabado el entrenamiento, su velocidad de respuesta es muy alta, son capaces de identificar objetos, palabras, etc. sin demora. Por último, el conocimiento se divide por toda la red, así, si se estropea una parte la red neuronal, se siguen dando respuestas correctas.

Desde la década de los 40, gracias a McCullough y Pitts, los cuales describieron el cálculo lógico de las redes neuronales y perfeccionaron el primer módulo formal de una neurona elemental, el modelo neuronal ha ido acompañando al desarrollo de la Informática.

En la década de los 50, Minsky comenzó a construir la primera neurocomputadora basada en modelos de redes neuronales que imitan al cerebro.

A finales de los 50 también se publicó un libro, escrito por Minsky y Pappert (autoridades de la inteligencia artificial (IA) clásica), donde se hablaba de las limitaciones de los redes neuronales de una capa. Esto hizo que se perdiera interés en el campo de las redes neuronales hasta la

década de los 80, en que el estudio de nuevas arquitecturas de redes y la mayor potencia de los ordenadores permiten el diseño de redes muy eficientes en tareas en las que otros procedimientos de tipo simbólico encuentran dificultades.

En 1962, Frank Rosenblatt presenta los resultados de una máquina a la que denominó "Perceptrón", la cual reproducía una estructura neuronal muy simplificada, capaz de aprender a reconocer y clasificar determinadas figuras.

En 1982, J. Hopfield describe las "redes de Hopfield" en las que se utilizan funciones de energía para entender las redes dinámicas. En 1983, Cohen y Grossberg desarrollaron el principio de la memoria direccional. En 1986, Rumulhart, Hunton y Williams redescubrieron el algoritmo de "back-propagation" para el aprendizaje de redes neuronales y, por otro lado, se empezaron a construir redes neuronales implementadas en silicio gracias a las nuevas tecnologías de fabricación de microchips.

Entre las aplicaciones más importantes de las redes neuronales, destacan el análisis y reconocimiento de voz, de imágenes, de formas, el diagnóstico clínico, etc. Por otro lado también destaca su uso en el análisis de series temporales así como en los modelos meteorológicos y en robótica.

En general, el esquema de una red neuronal (artificial) es el siguiente:

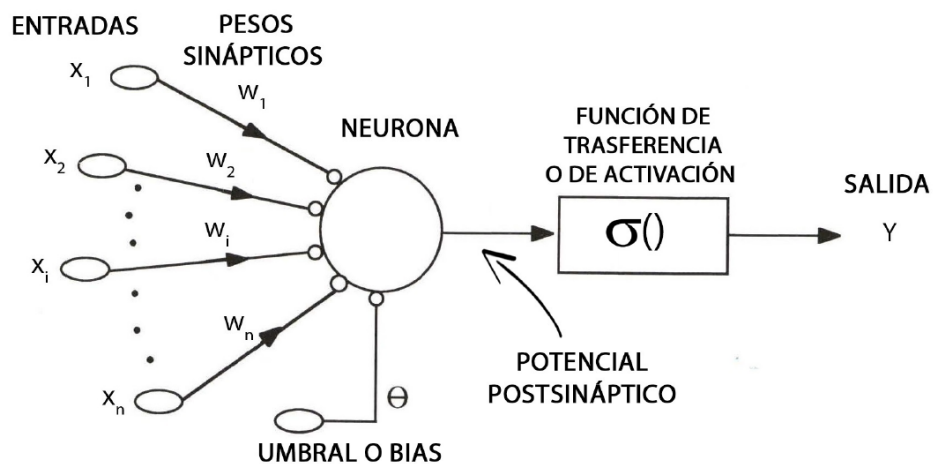


Figura 2.1: Neurona Artificial

En el contexto que nos ocupa, el tipo de red neuronal más utilizado es el que consta de tres capas (con una capa oculta). Se define como una función, H , que transforma un elemento $x \in \mathbb{R}^n$ en un número real de la siguiente forma:

$$H(x) = \sum_{j=1}^s w_j \cdot \sigma(x \cdot a_j + \theta_j)$$

donde los coeficientes w_j y θ_j son números reales, los pesos $a_j \in \mathbb{R}^n$ y la función $\sigma : \mathbb{R} \rightarrow \mathbb{R}$ es la denominada *función de activación*.

Como puede observarse, una red neuronal tiene una estructura bastante compleja, con una gran variedad de parámetros y variables, lo que las convierte en herramientas muy flexibles que se ajustan a, prácticamente, cualquier función. De hecho, se ha demostrado que, para varios tipos de funciones de activación, cualquier función continua definida sobre un compacto $K \subset \mathbb{R}^n$ que toma valores reales (es decir, cualquier función en $C(K)$) puede ser aproximada arbitrariamente por redes neuronales de tres capas; en otras palabras, la siguiente envoltura lineal

$$\mathcal{H}(\sigma) := LIN\{\sigma(x \cdot a_j + \theta_j) : \theta_j \in \mathbb{R}, a_j \in \mathbb{R}^n\}$$

es densa en el espacio $C(\mathbb{R}^n)$ con la topología de la convergencia uniforme sobre los compactos de \mathbb{R}^n . El resultado más general en este contexto se debe a Leshno, Lin, Pinkus y Schocken [1]: una condición necesaria y suficiente para que $\mathcal{H}(\sigma)$ sea denso en $C(\mathbb{R}^n)$ (i.e., un aproximador universal para $C(\mathbb{R}^n)$) es que la función de activación no sea un polinomio.

En este trabajo nos centraremos en las redes neuronales de tres capas con las funciones de activación más utilizadas: las funciones sigmoidales.

Definición 1 *Una función sigmoideal es una aplicación*

$$\sigma : \mathbb{R} \longrightarrow \mathbb{R}$$

$$\text{tal que } \lim_{t \rightarrow +\infty} \sigma(t) = 1 \text{ y } \lim_{t \rightarrow -\infty} \sigma(t) = 0$$

Las funciones de activación sigmoidales más habituales son las siguientes:

$$\text{La función de Heaviside o escalón: } \sigma(x) = \begin{cases} 1 & \text{si } x > 0 \\ 0 & \text{si } x \leq 0 \end{cases}$$

La función lineal a trozos o rampa unidad:
$$\sigma(x) = \begin{cases} 1 & \text{si } x \geq 1 \\ x & \text{si } 0 \leq x < 1 \\ 0 & \text{si } x < 0 \end{cases}$$

La función logística:
$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

En particular, la función logística se ha venido utilizando en muy diversos ámbitos, como en Biología, Física, Estadística, Demografía, etc. De hecho, en el siglo XIX ya se utilizaron para modelar el crecimiento de poblaciones.

Los primeros resultados sobre aproximación de funciones usando redes neuronales activadas mediante funciones sigmoideas se deben a Cybenko [2]. En este artículo y en algunos posteriores, se obtienen pruebas no constructivas de dicha propiedad aproximativa en el que juegan un papel importante varios resultados clásicos del Análisis Funcional, como el Teorema de Representación de Kolmogorov [3] y, principalmente, el Teorema de Stone-Weierstrass [4] que, como es bien sabido, trata sobre la densidad de ciertas subálgebras o subretículos de funciones continuas. Sin embargo, el aspecto no constructivo de estas pruebas dificulta las aplicaciones prácticas de estos resultados.

Nuestro principal objetivo será presentar, con todo detalle, una prueba constructiva de la capacidad aproximativa de las redes neuronales activadas mediante una función sigmoideal acotada, en el contexto de funciones de varias variables.

2.1.1. Conceptos previos

Definición 2 *Un espacio vectorial topológico es un espacio vectorial $(X, +, \cdot)$ sobre un cuerpo \mathbb{K} dotado de una topología τ donde los puntos son cerrados y de tal manera que las aplicaciones:*

$$\begin{array}{l} + : X \times X \longrightarrow X \\ (a, b) \mapsto a + b \end{array} \quad y \quad \begin{array}{l} \cdot : \mathbb{K} \times X \longrightarrow X \\ (\lambda, b) \mapsto \lambda b \end{array}$$

son continuas (usando en los productos cartesianos las respectivas topologías producto) respecto de la topología τ .

Definición 3 Una función $f : \mathbb{R} \rightarrow \mathbb{R}$ se dice que es **continua** en un punto $a \in \mathbb{R}$ si, para cada $\epsilon > 0$, se puede encontrar un $\delta > 0$ tal que para todo $x \in \mathbb{R}$ con $|x - a| < \delta$ se verifica que $|f(x) - f(a)| < \epsilon$.

Tendremos **continuidad uniforme** cuando δ no dependa de a .

Definición 4 Una función $f : X \rightarrow \mathbb{R}$ se dice que está **acotada** si, para todo $x \in X$, existe un número real $M > 0$ tal que $|f(x)| \leq M$.

Definición 5 Sea V un espacio vectorial sobre un cuerpo K . Una función $\|\cdot\| : V \rightarrow K$ es una norma en V si satisface las siguientes propiedades:

$$\begin{aligned} (i) \quad & x \neq 0 \Leftrightarrow \|x\| > 0. \\ (ii) \quad & \|\alpha x\| = |\alpha| \|x\|, \text{ para todo } \alpha \in K. \\ (iii) \quad & \|x + y\| \leq \|x\| + \|y\|, \text{ para todo } x, y \in V. \end{aligned} \tag{2.1}$$

En concreto, si $x = (x_1, \dots, x_n) \in \mathbb{R}^n$, podemos definir la norma euclídea de x como el número real no negativo $\sqrt{x_1^2 + \dots + x_n^2}$ que denotaremos por el símbolo $\|x\|_2$, es decir,

$$\|x\|_2 = \sqrt{x_1^2 + \dots + x_n^2}.$$

2.2. Aproximación de funciones de varias variables mediante redes neuronales

En esta sección mostraremos la capacidad de las redes neuronales para aproximar funciones continuas de varias variables en un compacto. En primer lugar enunciamos el siguiente lema que será fundamental para probar nuestro teorema principal y que se deduce directamente de la definición de función sigmoideal.

Lema 1 Sean $(x_0, y_0), (x_1, y_1), \dots, (x_N, y_N) \in \mathbb{R}^2$ para algún $N \in \mathbb{N}^+$ fijo y sea σ una función sigmoideal. Para todo $\epsilon > 0$ y $h > 0$, existe $\bar{w} := \bar{w}(\epsilon, h) > 0$ tal que si $w \geq \bar{w}$ y $k = 0, 1, \dots, N$, entonces se cumple

1. $|\sigma(w \| (x, y) - (x_k, y_k) \|_2) - 1| < \epsilon;$

2. $|\sigma(-w \| (x, y) - (x_k, y_k) \|_2)| < \epsilon,$

para todo $(x, y) \in \mathbb{R}^2$ tal que $\| (x, y) - (x_k, y_k) \|_2 \geq h$.

A continuación, mostramos el teorema principal donde se prueba la aproximación de funciones de dos variables sobre un compacto de \mathbb{R}^2 a partir de redes neuronales activadas por funciones sigmoideas acotadas. Este teorema fue demostrado por D. Costarelli y R. Spigler en [5].

Teorema 1 Sea σ una función sigmoidea acotada, y sea $f \in C(Q)$, donde $Q := [a, b] \times [c, d] \subset \mathbb{R}^2$ con $b - a = d - c = l$ fijo. Para todo $\epsilon > 0$ existen $N \in \mathbb{N}^+$ y $w > 0$ (dependiente de N) tales que si se define

$$\begin{aligned} (\tilde{G}_N f)(x, y) := & \sum_{i=1}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})] \sigma(w \chi_{ij}(x, y) \| (x, y) - (t_{x_i}, t_{y_j}) \|_2) \\ & + \sum_{i=1}^N f(x_i, y_0) \sigma(w \chi_{i0}(x, y) \| (x, y) - (t_{x_i}, t_{y_0}) \|_2), \end{aligned} \quad (2.2)$$

donde $(x, y) \in Q$, $h := \frac{l}{N}$, $x_i := a + hi$, $y_j := c + hj$, para $i, j = -1, 0, 1, \dots, N$ y $t_{x_i} := \frac{x_{i-1} + x_i}{2}$, $t_{y_j} := \frac{y_{j-1} + y_j}{2}$ para $i, j = 0, 1, \dots, N$ y además

$$\chi_{ij}(x, y) := \begin{cases} +1 & \text{si } x \in (x_{i-1}, x_i] \text{ e } y \geq y_j \\ -1 & \text{en otro caso} \end{cases}$$

para $i = 2, \dots, N$ y $j = 0, \dots, N$, y

$$\chi_{1j}(x, y) := \begin{cases} +1 & \text{si } x \in [x_0, x_1] \text{ e } y \geq y_j \\ -1 & \text{en otro caso} \end{cases}$$

para $j = 0, \dots, N$, entonces

$$\| \tilde{G}_N f - f \|_\infty = \sup_{(x,y) \in Q} | \tilde{G}_N f(x, y) - f(x, y) | < \epsilon.$$

Demostración.

Fijemos $\epsilon > 0$. Puesto que, por hipótesis, f es continua sobre el subconjunto Q compacto, será uniformemente continua. Por tanto, dado

$$\eta := \frac{\epsilon}{(\|f\|_\infty + \|\sigma\|_\infty + 2)}$$

existe $\delta > 0$ tal que $|f(x, y) - f(z, t)| < \eta$ para cada $(x, y), (z, t) \in Q$ que cumplen $\|(x, y) - (z, t)\|_2 < \delta$.

Ahora, se elige $N \in \mathbb{N}^+$ tal que $h := l/N < \delta \sqrt{2}$ y $1/N < \eta$. Además, se fija

$$w \geq \bar{w}(1/N^2, h/2) = \bar{w}(1/N^2) > 0$$

donde $\bar{w}(1/N^2)$ se obtiene del lema anterior para los puntos $(t_{x_i}, t_{y_j}), i, j = 0, 1, \dots, N$.

Se considera la función $\tilde{G}_N f$ definida en (2.2) para w , y se fija $(x, y) \in Q$. Así pues, existen $k, \mu = 1, \dots, N$ tales que $(x, y) \in (x_{k-1}, x_k] \times [y_{\mu-1}, y_\mu]$ siempre que $k \geq 2$ ó $(x, y) \in [x_0, x_1] \times [y_{\mu-1}, y_\mu]$ en otro caso. Definimos

$$\mathcal{L}_{k\mu}(x, y) := f(x_k, y_0) + [f(x_k, y_1) - f(x_k, y_0)]\sigma(w\chi_{k\mu}(x, y) \|(x, y) - (t_{x_k}, t_{y_1})\|_2),$$

si $\mu = 1$, y

$$\begin{aligned} \mathcal{L}_{k\mu}(x, y) := & \sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] + f(x_k, y_0) \\ & + [f(x_k, y_\mu) - f(x_k, y_{\mu-1})]\sigma(w\chi_{k\mu}(x, y) \|(x, y) - (t_{x_k}, t_{y_\mu})\|_2), \end{aligned} \quad (2.3)$$

si $\mu \geq 2$. En ambos casos, escribiremos

$$\begin{aligned} & \left| (\tilde{G}_N f)(x, y) - f(x, y) \right| \\ & \leq \left| (\tilde{G}_N f)(x, y) - \mathcal{L}_{k\mu}(x, y) \right| + \left| \mathcal{L}_{k\mu}(x, y) - f(x, y) \right| := H_1 + H_2. \end{aligned}$$

Primero acotaremos H_1 cuando $\mu \geq 2$. Así pues,

$$\begin{aligned} H_1 = & \left| \sum_{i=1}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})]\sigma(w\chi_{ij}(x, y) \|(x, y) - (t_{x_i}, t_{y_j})\|_2) \right. \\ & + \sum_{i=1}^N f(x_i, y_0)\sigma(w\chi_{i0}(x, y) \|(x, y) - (t_{x_i}, t_{y_0})\|_2) \\ & - \left[\sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] + f(x_k, y_0) \right. \\ & \left. \left. + [f(x_k, y_\mu) - f(x_k, y_{\mu-1})]\sigma(w\chi_{k\mu}(x, y) \|(x, y) - (t_{x_k}, t_{y_\mu})\|_2) \right] \right|. \end{aligned} \quad (2.4)$$

Si ahora reagrupamos los términos, obtenemos que:

$$\begin{aligned}
H_1 = & \left| \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})] \sigma(w\chi_{ij}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_j}) \parallel_2) \right. \\
& + \sum_{j=1}^N [f(x_k, y_j) - f(x_k, y_{j-1})] \sigma(w\chi_{kj}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_j}) \parallel_2) \\
& + \sum_{\substack{i=1 \\ i \neq k}}^N f(x_i, y_0) \sigma(w\chi_{i0}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_0}) \parallel_2) \\
& + f(x_k, y_0) \sigma(w\chi_{k0}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_0}) \parallel_2) \\
& - \sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] - f(x_k, y_0) \\
& \left. - [f(x_k, y_\mu) - f(x_k, y_{\mu-1})] \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_\mu}) \parallel_2) \right|. \tag{2.5}
\end{aligned}$$

Por tanto se tiene que:

$$\begin{aligned}
H_1 \leq & \left| \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})] \sigma(w\chi_{ij}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_j}) \parallel_2) \right. \\
& + \sum_{\substack{i=1 \\ i \neq k}}^N f(x_i, y_0) \sigma(w\chi_{i0}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_0}) \parallel_2) \\
& + \sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] \left[\sigma(w\chi_{kj}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_j}) \parallel_2) - 1 \right] \\
& + \sum_{j=\mu+1}^N [f(x_k, y_j) - f(x_k, y_{j-1})] \sigma(w\chi_{kj}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_j}) \parallel_2) \\
& + [f(x_k, y_\mu) - f(x_k, y_{\mu-1})] \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_\mu}) \parallel_2) \\
& - [f(x_k, y_\mu) - f(x_k, y_{\mu-1})] \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_\mu}) \parallel_2) \\
& \left. + f(x_k, y_0) \sigma(w\chi_{k0}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_0}) \parallel_2) - f(x_k, y_0) \right|. \tag{2.6}
\end{aligned}$$

Usando la desigualdad triangular obtenemos que:

$$\begin{aligned}
H_1 &\leq \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N |f(x_i, y_j) - f(x_i, y_{j-1})| \left| \sigma(w\chi_{ij}(x, y)) \left\| (x, y) - (t_{x_i}, t_{y_j}) \right\|_2 \right| \\
&\quad + \sum_{\substack{i=1 \\ i \neq k}}^N |f(x_i, y_0)| \left| \sigma(w\chi_{i0}(x, y)) \left\| (x, y) - (t_{x_i}, t_{y_0}) \right\|_2 \right| \\
&\quad + \sum_{j=1}^{\mu-1} |f(x_k, y_j) - f(x_k, y_{j-1})| \left| \sigma(w\chi_{kj}(x, y)) \left\| (x, y) - (t_{x_k}, t_{y_j}) \right\|_2 \right| - 1 \Big| \\
&\quad + \sum_{j=\mu+1}^N |f(x_k, y_j) - f(x_k, y_{j-1})| \left| \sigma(w\chi_{kj}(x, y)) \left\| (x, y) - (t_{x_k}, t_{y_j}) \right\|_2 \right| \\
&\quad + |f(x_k, y_0)| \left| \sigma(w\chi_{k0}(x, y)) \left\| (x, y) - (t_{x_k}, t_{y_0}) \right\|_2 \right| - 1 \Big|.
\end{aligned} \tag{2.7}$$

Como $\| (x_i, y_j) - (x_i, y_{j-1}) \|_2 < \delta$ para cada $i, j = 1, \dots, N$, $\| (x, y) - (t_{x_i}, t_{y_j}) \|_2 \geq h/2$ para cada $(t_{x_i}, t_{y_j}) \neq (t_{x_k}, t_{y_\mu})$, y por las condiciones 1 y 2 del lema anterior y la definición de χ_{ij} , obtenemos:

$$\begin{aligned}
H_1 &< \frac{1}{N^2} \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N \eta + \frac{1}{N^2} \sum_{\substack{i=1 \\ i \neq k}}^N \|f\|_\infty + \frac{1}{N^2} \sum_{j=1}^{\mu-1} \eta + \frac{1}{N^2} \sum_{j=\mu+1}^N \eta + \frac{1}{N^2} \|f\|_\infty \\
&= \frac{1}{N^2} \eta(N-1)N + \frac{1}{N^2} \|f\|_\infty (N-1) + \frac{1}{N^2} \eta(N-1) + \frac{1}{N^2} \|f\|_\infty \\
&= \frac{1}{N^2} \eta \left[(N^2 - 1) \right] + \frac{1}{N^2} \|f\|_\infty N \\
&\leq \eta + \frac{1}{N} \|f\|_\infty < (1 + \|f\|_\infty) \eta.
\end{aligned} \tag{2.8}$$

Vamos ahora a acotar H_1 para $\mu = 1$.

$$\begin{aligned}
H_1 = & \left| \sum_{i=1}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})] \sigma(w\chi_{ij}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_j}) \parallel_2) \right. \\
& + \sum_{i=1}^N f(x_i, y_0) \sigma(w\chi_{i0}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_0}) \parallel_2) \\
& \left. - \left[f(x_k, y_0) + [f(x_k, y_1) - f(x_k, y_0)] \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_\mu}) \parallel_2) \right] \right|.
\end{aligned} \tag{2.9}$$

De nuevo, de forma similar al caso anterior reagrupamos términos y obtenemos:

$$\begin{aligned}
H_1 = & \left| \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N [f(x_i, y_j) - f(x_i, y_{j-1})] \sigma(w\chi_{ij}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_j}) \parallel_2) \right. \\
& + \sum_{j=1}^N [f(x_k, y_j) - f(x_k, y_{j-1})] \sigma(w\chi_{kj}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_j}) \parallel_2) \\
& + \sum_{\substack{i=1 \\ i \neq k}}^N f(x_i, y_0) \sigma(w\chi_{i0}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_0}) \parallel_2) \\
& + f(x_k, y_0) \sigma(w\chi_{k0}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_0}) \parallel_2) \\
& \left. - f(x_k, y_0) - [f(x_k, y_1) - f(x_k, y_0)] \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_1}) \parallel_2) \right|.
\end{aligned} \tag{2.10}$$

Y finalmente se tiene que:

$$\begin{aligned}
H_1 \leq & \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N |f(x_i, y_j) - f(x_i, y_{j-1})| \left| \sigma(w\chi_{ij}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_j}) \parallel_2) \right| \\
& + \sum_{\substack{i=1 \\ i \neq k}}^N |f(x_i, y_0)| \left| \sigma(w\chi_{i0}(x, y) \parallel (x, y) - (t_{x_i}, t_{y_0}) \parallel_2) \right| \\
& + \sum_{j=2}^N |f(x_k, y_j) - f(x_k, y_{j-1})| \left| \sigma(w\chi_{kj}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_j}) \parallel_2) \right| \\
& + |f(x_k, y_0)| \left| (\sigma(w\chi_{k0}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_0}) \parallel_2) - 1) \right|.
\end{aligned} \tag{2.11}$$

De forma análoga al caso $\mu \geq 2$ acotamos cada término obteniendo:

$$\begin{aligned}
H_1 &< \frac{1}{N^2} \sum_{\substack{i=1 \\ i \neq k}}^N \sum_{j=1}^N \eta + \frac{1}{N^2} \sum_{\substack{i=1 \\ i \neq k}}^N \|f\|_\infty + \frac{1}{N^2} \sum_{j=2}^N \eta + \frac{1}{N^2} \|f\|_\infty \\
&= \frac{1}{N^2} \eta(N-1)(N) + \frac{1}{N^2} \|f\|_\infty (N-1) + \frac{1}{N^2} \eta(N-1) + \frac{1}{N^2} \|f\|_\infty \\
&= \frac{N^2-1}{N^2} \eta + \frac{1}{N} \|f\|_\infty \leq \eta + \frac{1}{N} \|f\|_\infty < (1 + \|f\|_\infty) \eta.
\end{aligned} \tag{2.12}$$

Finalmente, teniendo en cuenta que:

$$\begin{aligned}
&\sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] + f(x_k, y_0) = f(x_k, y_1) - f(x_k, y_0) + f(x_k, y_0) \\
&+ f(x_k, y_2) - f(x_k, y_1) + f(x_k, y_3) - f(x_k, y_2) + \dots + f(x_k, y_{\mu-2}) \\
&- f(x_k, y_{\mu-3}) + f(x_k, y_{\mu-1}) - f(x_k, y_{\mu-2}) = f(x_k, y_{\mu-1})
\end{aligned} \tag{2.13}$$

y que $\|(x, y) - (x_k, y_{\mu-1})\|_2 \leq \sqrt{2}h < \delta$, obtenemos:

$$\begin{aligned}
H_2 &= \left| \sum_{j=1}^{\mu-1} [f(x_k, y_j) - f(x_k, y_{j-1})] + f(x_k, y_0) \right. \\
&\quad \left. + [f(x_k, y_\mu) - f(x_k, y_{\mu-1})] \sigma(w\chi_{k\mu}(x, y) \|(x, y) - (t_{x_k}, t_{y_\mu})\|_2) - f(x, y) \right|.
\end{aligned} \tag{2.14}$$

Por tanto, reagrupando términos obtenemos que:

$$H_2 = \left| f(x_k, y_{\mu-1}) + [f(x_k, y_\mu) - f(x_k, y_{\mu-1})] \sigma(w\chi_{k\mu}(x, y) \|(x, y) - (t_{x_k}, t_{y_\mu})\|_2) - f(x, y) \right|. \tag{2.15}$$

Si aplicamos la desigualdad triangular se tiene:

$$\begin{aligned}
H_2 &\leq \left| f(x_k, y_{\mu-1}) - f(x, y) \right| + \left| f(x_k, y_\mu) - f(x_k, y_{\mu-1}) \right| \left| \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_\mu}) \parallel_2) \right| \\
&< \left| f(x_k, y_{\mu-1}) - f(x, y) \right| + \eta \|\sigma\|_\infty < \eta + \eta \|\sigma\|_\infty = (\|\sigma\|_\infty + 1)\eta.
\end{aligned} \tag{2.16}$$

A continuación acotamos H_2 para el caso $\mu = 1$:

$$H_2 := \left| f(x_k, y_0) + [f(x_k, y_1) - f(x_k, y_0)]\sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_1}) \parallel_2) - f(x, y) \right|.$$

Por lo tanto se tiene que:

$$\begin{aligned}
H_2 &\leq \left| f(x_k, y_0) - f(x, y) \right| + \left| f(x_k, y_1) - f(x_k, y_0) \right| \left| \sigma(w\chi_{k\mu}(x, y) \parallel (x, y) - (t_{x_k}, t_{y_1}) \parallel_2) \right| \\
&< \eta + \eta \|\sigma\|_\infty = \eta(1 + \|\sigma\|_\infty).
\end{aligned} \tag{2.17}$$

A partir de (2.12) y (2.17), se tiene que:

$$\begin{aligned}
\left| (\tilde{G}_N f)(x, y) - f(x, y) \right| &\leq H_1 + H_2 < (1 + \|f\|_\infty)\eta + (1 + \|\sigma\|_\infty)\eta \\
&= \eta(2 + \|f\|_\infty + \|\sigma\|_\infty) = \epsilon.
\end{aligned} \tag{2.18}$$

Finalmente, podemos concluir que :

$$\left\| \tilde{G}_N f - f \right\|_\infty < \epsilon.$$

Para ilustrar el teorema anterior presentamos una función de dos variables y la red neuronal activada por la función logística que la aproxima:

Ejemplo 1 Consideremos la función de dos variables

$$h(x, y) := (y^4 - 2y)\sin x - xy^3 + \frac{x}{3} + \frac{1}{x^{16}/30 + e^{-3|y|}/50 + 1/100}$$

en el intervalo $[-4,4]$. La aproximaremos mediante la red neuronal definida en (2.2) para $N=25$ y $N=50$ nodos. Además tomaremos $w = N^2/8$ y $\sigma(x) = (1 + e^{-x})^{-1}$.

El código de programación realizado en Matlab para obtener la red neuronal y su representación gráfica se muestra a continuación:

```
function [q] = xi(X,Y,i,j)
    a=-1;
    if i==1
        if X(1)<= X(i,j) <= X(2) & Y(i,j)>=Y(j)
            a=1;
        else
            a=-1;
        end
    else
        if X(i-1)<X(i,j)<X(i) & Y(i,j)>=Y(j)
            a=1;
        else
            a=-1;
        end
    end
    q=a;
end
```

```
function [z] = f(x,y)
z=(y.^4 - 2.*y) .* sin(x) - x.*(y.^3) + x./3 + 1./ ((x.^16 ./30)
+ (exp(-3.*abs(y)) ./50) + (1./100));
end
```

```
clc
clear
close all
```

```

a=;
b=;
N=;
l=b-a;
w=N.^2 ./ l;
h=l ./ N;
[X,Y]=meshgrid(a:h:b);
xp=a+h.*( -1);
yp=a+h.*( -1);
at=(xp+X(1))./2;
bt=(X(length(X)-1,length(X)-1)+X(length(X),length(X)))./2;
[X,Y]=meshgrid(a:h:b);
[TX,TY]=meshgrid(at:h:bt);
r=0;
m=1;
for i = 2:N+1
    for j=2:N+1
        r= r+ ((f(X,Y)-f(X,Y(m))).*(1+exp(-(w.*xi(X,Y,i,j)
        .*sqrt((X-TX).^2 + (Y-TY).^2))))).^(-1));
        m=m+1;
    end
end
for m= 2:N+1
    r=r+f(X,Y(1)).* ((1+exp(-(w.*xi(X,Y,m,1)
    .*sqrt((X-TX).^2 + (Y-TY(1)).^2))))).^(-1));
end
subplot(1,2,1)
surf(X,Y,f(X,Y))
subplot(1,2,2)
surf(X,Y,r)

```

En la gráfica siguiente se muestra la función $h(x,y)$.

A continuación se muestran las aproximaciones de la función $h(x,y)$ mediante la red neuronal para $N=25$ y para $N=50$.

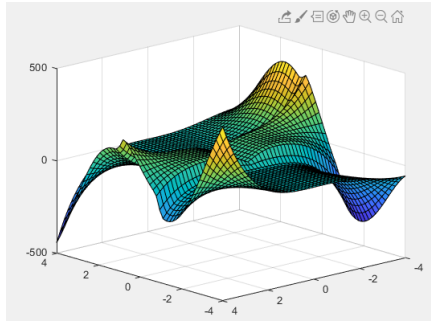


Figura 2.2: Gráfica de la función $h(x,y)$

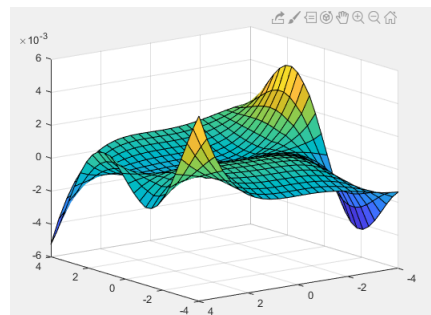


Figura 2.3: Aproximación de la función h para $N=25$

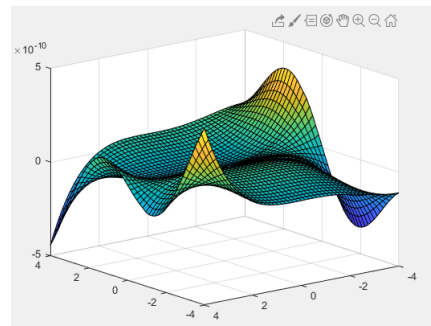


Figura 2.4: Aproximación de la función h para $N=50$

Podemos observar que a medida que se aumenta el valor de la N , vemos mas semejanza respecto a la función original.

Capítulo 3

Conclusiones

En el periodo de prácticas se desarrolló e implementó un programa mediante el lenguaje R para el grupo empresarial UBE CORPORATION EUROPE con el objetivo de controlar el rendimiento de las válvulas que regulan la materia prima que hay en los tanques.

Al inicio de las prácticas tuve que aprender diversos contenidos sobre la asignatura “Sistemas Automáticos” del grado en Ingeniería Industrial, ya que necesitaba entender cómo funcionaban los tanques con materia prima, los PIDs, etc. Gracias a estas prácticas he aprendido cómo se regulan elementos para luego poder elaborar productos. Además de esto, como tenía que procesar miles de datos por segundos y programar mediante sentencias SQL, tuve que poner en práctica lo aprendido en las asignaturas “Fundamentos Estadísticos en la Minería de Datos” y “Bases de Datos” estudiadas en el Grado. En definitiva la estancia en prácticas fue satisfactoria ya que me permitió ampliar mis conocimientos de programación, en particular en el lenguaje R.

Por otra parte, en lo que respecta al Trabajo Final de Grado, se ha llevado a cabo el estudio de la aproximación constructiva multivariante mediante redes neuronales.

En este documento, se ha demostrado que cualquier función continua definida sobre un compacto $Q \subset \mathbb{R}^2$ que toma valores reales puede ser aproximada arbitrariamente por redes neuronales activadas mediante funciones sigmoideas acotadas. Para ello, hemos desarrollado con todo detalle una prueba constructiva de esta aproximación. Finalmente proporcionamos un ejemplo de aproximación a una función concreta, la cual me ha permitido mejorar mi conocimiento sobre Matlab.

Bibliografía

- [1] LESHNO M., LIN V.Y., PINKUS A. AND SCHOCKEN S., *Multilayer Feedforward Networks with a Nonpolynomial Activation Function Can Approximate any Function*, Neural Networks, **6**, (1993), 861-867.
- [2] CYBENKO, G., *Approximation by Superpositions of a Sigmoidal Function*, Mathematics of Control, Signals, and Systems, **2**, (1989), 303-314.
- [3] TORRES L.G., *Redes Neuronales y Aproximación de Funciones*. Boletín de Matemáticas, [S.l.], **1**(2), (1994), 35-58.
- [4] COTTER N.E., *The Stone-Weierstrass Theorem and Its Application to Neural Network*. IEEE Transactions on Neural Networks, **1**(4), (1990), 290-295.
- [5] DANILO COSTARELLI AND RENATO SPIGLER., *Constructive Approximation by Superposition of Sigmoidal Functions*, Anal. Theory Appl, **29** (2013), 169-196.
- [6] CHEN T., CHEN H. AND LIU R.W., *Approximation Capability in $C(\mathbb{R}^n)$ by Multilayer Feedforward Networks and Related Problems*, IEEE Trans Neural Netw. **6**(1), (1995), 25-30.
- [7] HUANG G.B. AND BABRI H.A., *Comments on "Approximation Capability in $C(\mathbb{R}^n)$ by Multilayer Feedforward Networks and Related Problems"*. IEEE Transactions on Neural Networks, **9**(4), (1998), 714-715.
- [8] LLANAS B. Y SÁINZ F.J., *Aproximación eficiente de una función afín definida sobre un politopo, por medio de redes neuronales*. II Congreso Internacional de Matemáticas en la Ingeniería y la Arquitectura, (2008), 303-309.

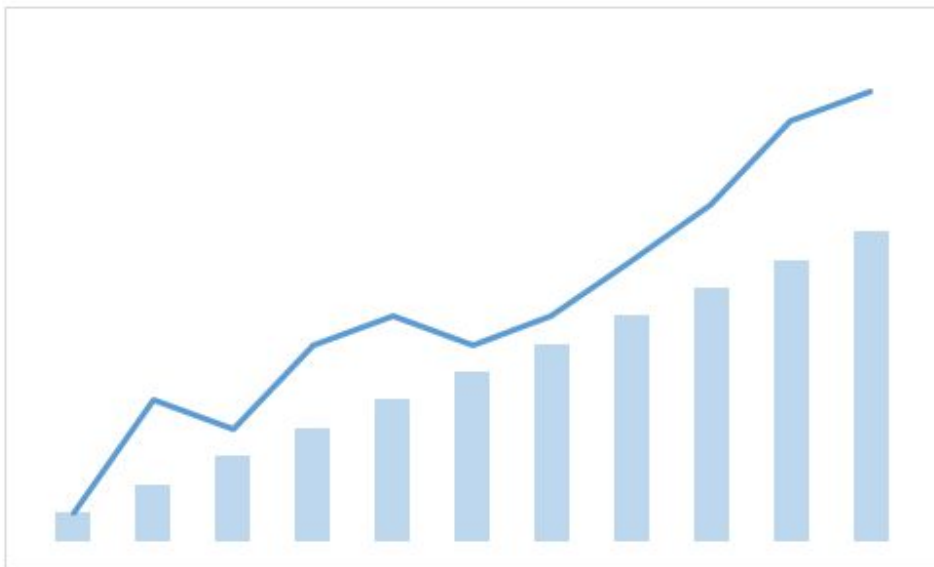
Capítulo 4

Manual de uso de CPM

CPM

Control Performance Monitoring

Manual de uso



Enero del 2019.

PRESENTACIÓN

Es casi inviable saber cuándo el rendimiento de un controlador está disminuyendo hasta que es demasiado tarde, esto es porque la productividad cambia tan levemente que pasa inadvertida.

El Control Performance Monitoring (CPM) tiene como finalidad localizar cambios en el rendimiento del controlador y evaluar el origen de los mismos. Para ello, se calculan una serie de indicadores de rendimiento (KPI) cuyo análisis permite determinar si existe algún tipo de mal funcionamiento en alguno de los lazos de control.

Los mecanismos de control facilitan a los administradores todo lo necesario para averiguar, localizar y solventar problemas que afecten de forma nociva al rendimiento de cualquier lazo de control antes de que estos impliquen un periodo de parada disparatado. Estas herramientas pueden examinar varios controladores de una misma planta o, por el contrario, pueden ser concretas para un PID en particular.

Durante el manual se hace referencia al “elemento de control final”. Éste es un dispositivo que manipula el valor de una variable como respuesta a una señal de salida desde el dispositivo de control automático. Los elementos de control final que se pueden encontrar, entre otros, son válvulas de control, relés, amortiguadores, variadores de frecuencia, etc.

A continuación, se explicarán todas las herramientas empleadas para monitorizar el correcto funcionamiento de los distintos PIDs, el modo en el que se han extraídos los datos, cómo se han manipulado y cómo se han interpretado. Además se aclarará con detalle como manipular la hoja Excel para poder visualizar todos los mecanismos de control empleados.

CONTENIDO

[EXTRACCIÓN Y MANIPULACIÓN DE DATOS](#)

[USO DE LA SEGMENTACIÓN DE DATOS](#)

[ESCALA DE TIEMPO](#)

[HIPERVÍNCULOS](#)

[RESUMEN](#)

[INDICADORES DE RENDIMIENTO \(KPI'S\)](#)

[MV HISTOGRAM](#)

[AAE](#)

[MV REVERSALS](#)

[MODE CHANGES](#)

[SET POINT CHANGES](#)

[PERCEPT TIME IN NORMAL](#)

[STRICTION](#)

[NOISE BAND](#)

[PROCESS DATA](#)

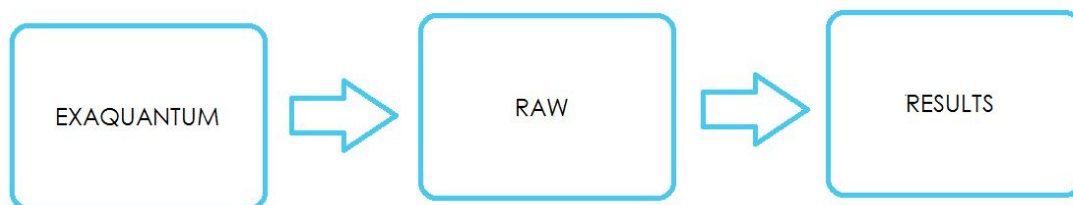
EXTRACCIÓN Y MANIPULACIÓN DE DATOS

Para cada lazo de control se obtienen datos de un día con frecuencia de 1 minuto. Se obtienen datos del SV ("Set point Value", señal de referencia, valor deseado en la salida), MV ("Manipulated Value", acción de control aplicada al proceso, salida proporcionada por el controlador) y PV ("Process Value", valor medido por el sensor, salida dada por el sistema). Es decir, por cada lazo y por cada día tenemos 1440 valores de SV, MV y PV.

Se realizan los cálculos para cada lazo de control por día, es decir, se seleccionan todos los datos de un día, se manipulan y se guardan los resultados, así pues, se consigue tener un registro por días, meses o años, a elección del usuario.

Los resultados de estas herramientas quedan plasmados en gráficas y tablas. Esto facilita la visualización de los problemas de mal funcionamiento que hacen que el rendimiento del controlador PID baje.

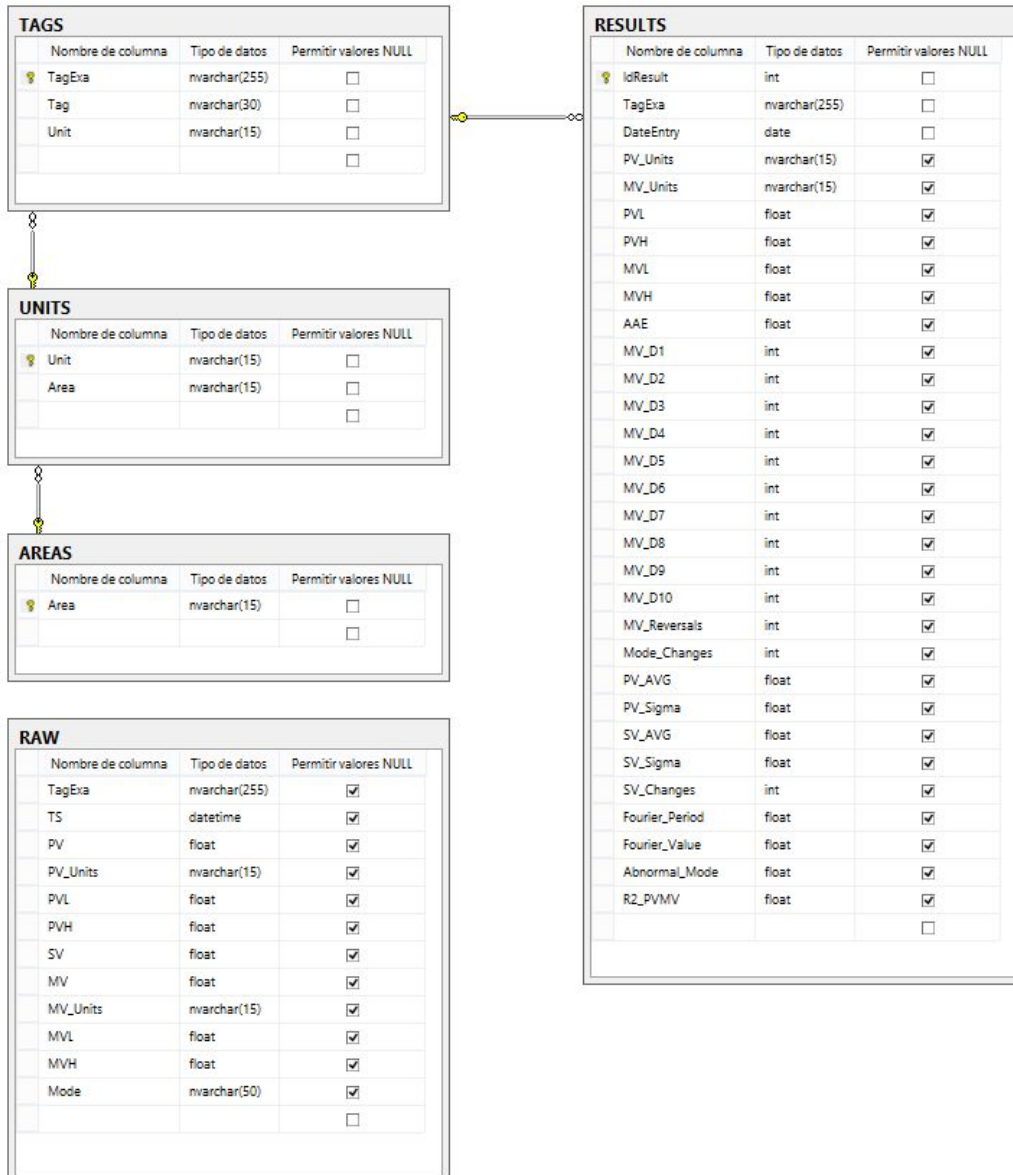
FLUJO DE DATOS



- Acceso a Exaquantum y almacenamiento de datos en la tabla RAW

Mediante la aplicación de consola .Net se descargan los datos requeridos de Exaquantum y se almacenan, por días, en la tabla RAW. Éste proceso se basa en coger la última fecha registrada en RESULTS (tabla donde se almacenarán todos los resultados de los KPIs) y calcular la siguiente, después de esto, se vacía la tabla RAW y se almacenan los datos de la fecha calculada. Por cada día y tag habrá 1440 datos (uno por minuto), así pues, en RAW solo almacenará los datos de un día.

En la siguiente imagen se puede ver el esquema empleado para extraer la información deseada.



- Cálculo de los KPIs en R

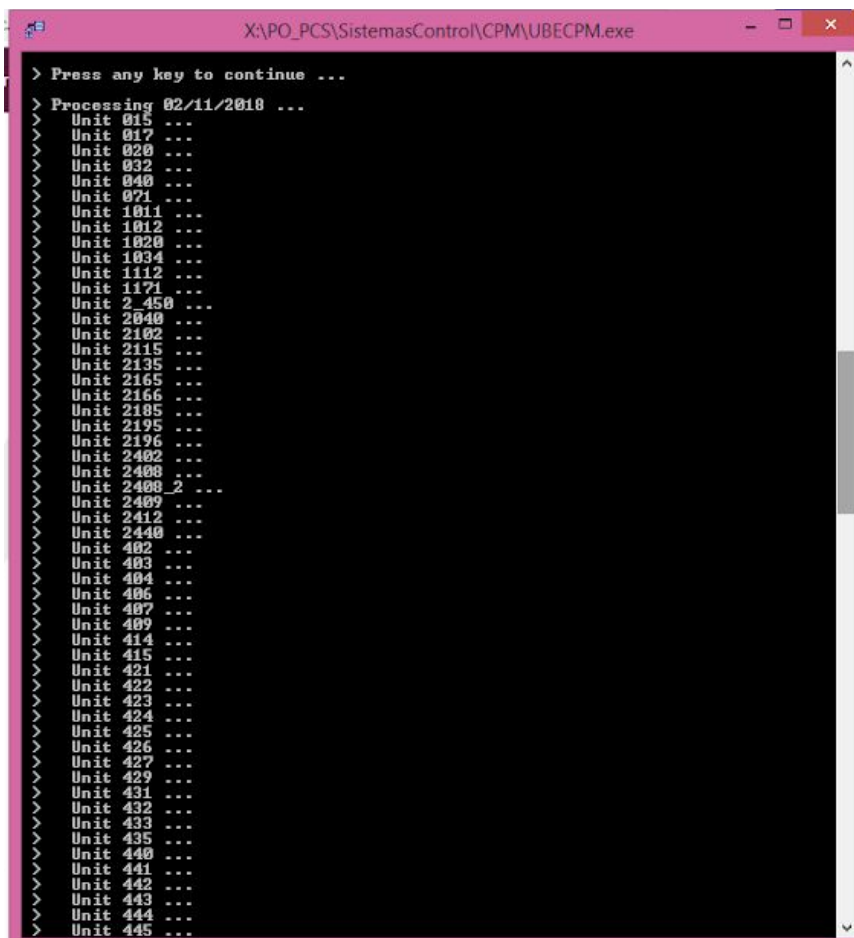
El código en R se conecta a la tabla RAW, en la que, se encuentran los datos de la figura anterior almacenados, mediante algoritmos se calculan todos los KPIs y se guardan en un objeto de R llamado “dataframe” (hoja de datos).

- Grabación en la tabla

Al final de la ejecución de todo el programa R, el dataframe contiene todos los resultados del KPI, y mediante un INSERT, se introducen en la tabla RESULTS.

- UBECPM.exe, nombre y ubicación de la base de datos y tablas empleadas.

Por cada ejecución de éste programa, se extraen los datos por día y unidad de Exaquantum, cuando se acaban de almacenar en RAW todos los datos, UBECPM.exe ejecuta el programa .R introduciendo en RESULTS los resultados de los KPIs, una vez acabado éste proceso, se vacía la tabla RAW, y si el usuario quiere almacenar los datos del día siguiente, pulsaría cualquier tecla.



```
X:\PO_PCS\SistemasControl\CPM\UBECPM.exe
> Press any key to continue ...
> Processing 02/11/2018 ...
> Unit 015 ...
> Unit 017 ...
> Unit 020 ...
> Unit 032 ...
> Unit 040 ...
> Unit 071 ...
> Unit 1011 ...
> Unit 1012 ...
> Unit 1020 ...
> Unit 1034 ...
> Unit 1112 ...
> Unit 1171 ...
> Unit 2_450 ...
> Unit 2040 ...
> Unit 2102 ...
> Unit 2115 ...
> Unit 2135 ...
> Unit 2165 ...
> Unit 2166 ...
> Unit 2185 ...
> Unit 2195 ...
> Unit 2196 ...
> Unit 2402 ...
> Unit 2408 ...
> Unit 2408 2 ...
> Unit 2409 ...
> Unit 2412 ...
> Unit 2440 ...
> Unit 402 ...
> Unit 403 ...
> Unit 404 ...
> Unit 406 ...
> Unit 407 ...
> Unit 409 ...
> Unit 414 ...
> Unit 415 ...
> Unit 421 ...
> Unit 422 ...
> Unit 423 ...
> Unit 424 ...
> Unit 425 ...
> Unit 426 ...
> Unit 427 ...
> Unit 429 ...
> Unit 431 ...
> Unit 432 ...
> Unit 433 ...
> Unit 435 ...
> Unit 440 ...
> Unit 441 ...
> Unit 442 ...
> Unit 443 ...
> Unit 444 ...
> Unit 445 ...
```

El nombre de la base de datos es CPM y se encuentra en VPSQLS001\UUBEINST01. Las tablas empleadas son RAW, RESULTS, AREAS, UNITS Y TAGS.

- Visualizar los datos en una hoja de cálculo Excel

Después de almacenar en RESULTS los datos necesarios, mediante una vista, los volcamos en una hoja Excel para poder graficar y entender el funcionamiento de los distintos controladores.

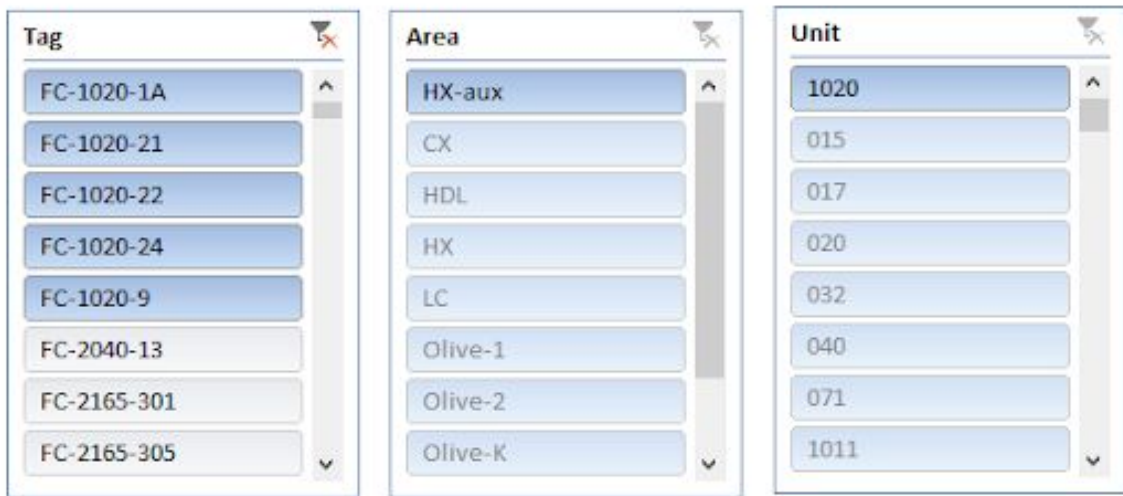
USO DE LA SEGMENTACIÓN DE DATOS

En todos los métodos empleados, se da la opción de visualizar los datos por controlador, área o unidad.



Para seleccionar cualquiera de las tres opciones, solo tendríamos que escoger el tag (lazo de control), el área o la unidad deseada y pulsar sobre ella. También es posible seleccionar varios controladores, áreas o unidades.

Por ejemplo, en la imagen siguiente se puede ver como se han seleccionado cinco lazos distintos.



Para saber cuál de las tres alternativas está seleccionada, hay que observar la marca situada en la parte superior derecha de las opciones, llamada “borrar filtro”.



Para deseleccionar cualquiera de las opciones basta con pulsar dicha marca o teclear Alt + C.

ESCALA DE TIEMPO

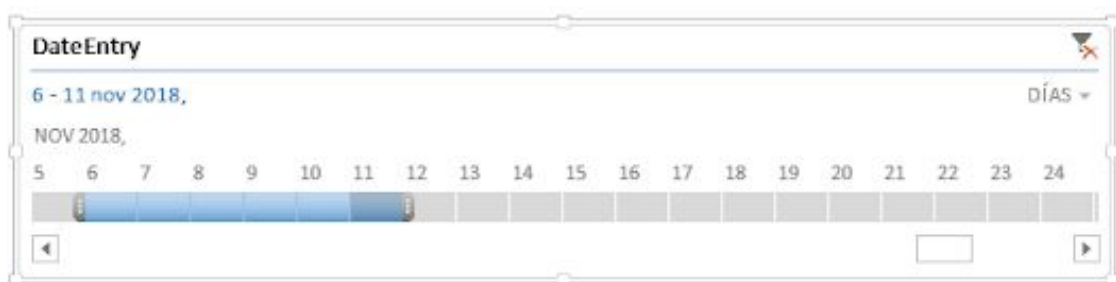
La escala de tiempo nos permite filtrar los resultados por días, meses o años.



Para escoger el periodo de tiempo basta con pulsar sobre la palabra MESES y seleccionar las fechas deseadas.



Además, puedes establecer el rango de años, trimestres, meses y días presionando sobre la franja azul.



Así pues en el ejemplo de la figura anterior estarían seleccionados los datos del 6 al 11 de noviembre del 2018.

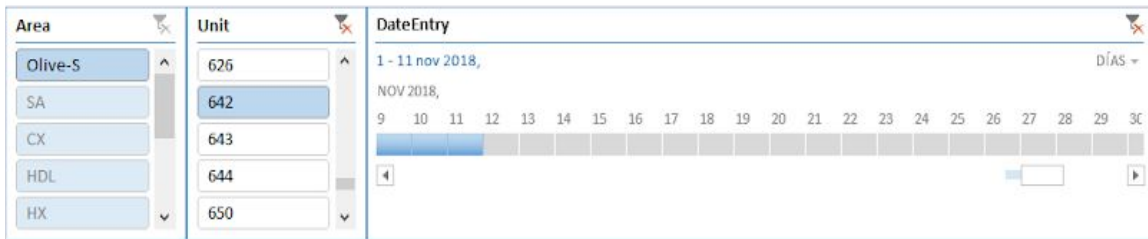
HIPERVÍNCULOS

En ésta parte puedes seleccionar cuál de los indicadores KPI se desea visualizar. Se trata de vínculos que te llevan a las páginas de detalle.

RESUMEN

En este apartado, se da un resumen por unidades o áreas de los KPI's empleados en el rango de fechas deseado.

Por ejemplo, para la unidad 642, tenemos un resumen de la información de todos sus lazos del 1 al 11 de noviembre de 2018.



Etiquetas de fila	Promedio AAE	MV Reversals/day	ModeChanges/day	SVChanges/day	FourierPeriod/day	FourierValue (%)	AbnormalMode/day	R2PVMV
FC64202	0,02%	2	0	9	8810,909091	525,31%	0,00%	0,999812795
FC64203	41,57%	15	3	5	4889,962073	145,96%	41,58%	0,298588565
FC64204	1,48%	14	4	25	13080	59,82%	0,04%	0,821221615
FC64206	1,47%	17	0	10	7560	0,53%	0,00%	0,602773763
FFC64202	22,48%	0	0	1	4800	0,57%	100,00%	
FFC64203	43,29%	3	3	1	8940	20,41%	41,93%	
LC64201	0,15%	107	0	0	3052	4,28%	0,00%	
LC64202	21,13%	0	0	0	12243,81963	24,04%	100,00%	
LC64204	5,18%	1769	0	4	11670	9,26%	0,00%	
PC64209	0,01%	1420	32	26	6773,724305	17,31%	0,15%	
PC64211	0,01%	608	0	1	1684,42019	0,14%	0,00%	

GRÁFICAS CONJUNTAS

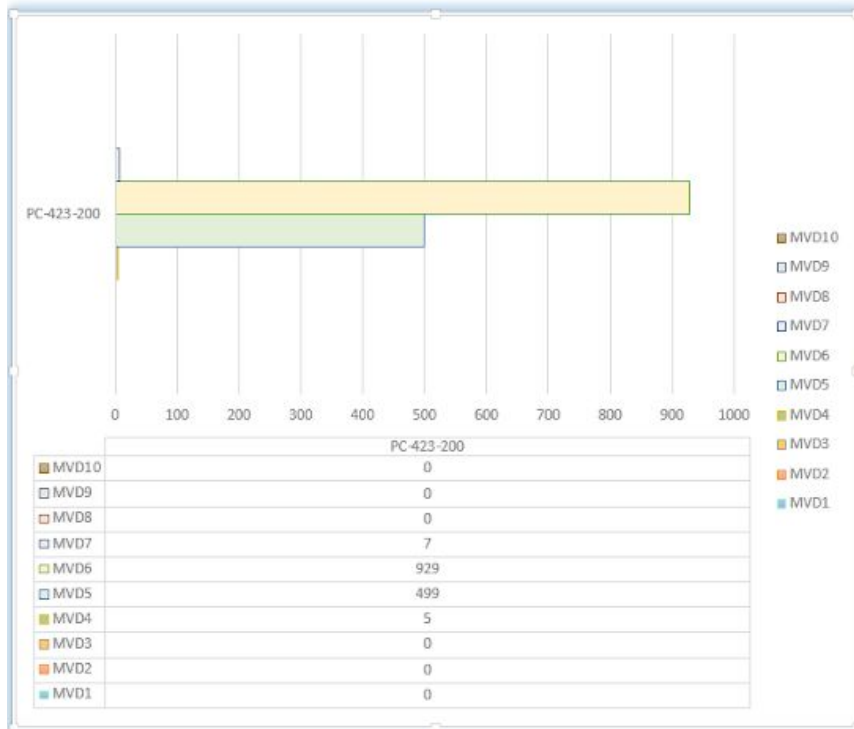
Aquí, puedes valorar todos juntos los KPI's explicados en este manual que indican el desempeño de cada lazo.

INDICADORES DE RENDIMIENTO (KPI'S)

MV HISTOGRAM

Muestra la distribución estadística de salida del controlador (MV). Determina el rango en el que el elemento de control final está actuando. Permite detectar si está saturado, es decir, si opera habitualmente en los límites del rango, o si se desplaza excesivamente.

La representación de los datos de esta herramienta se ha realizado mediante un histograma de la salida del lazo de control, en el que quedan reflejadas las posiciones del elemento de control y la frecuencia en cada uno de los intervalos.



El MVD1 recoge los valores del mv que estén comprendidos entre el 0 y el 10 del rango del actuador, el MVD2, los que estén comprendidos entre el 10 y el 20, así, hasta el MVD10 que recoge aquellos comprendidos entre el 90 y el 100. De este modo se divide el rango del instrumento en 10 intervalos.

Por ejemplo, de los 1440 valores del controlador PC-423-200 registrados en un día, 5 de ellas toman valores en el intervalo [30, 40] %, 499 en el intervalo [40, 50] %, 929 en el intervalo [50, 60] % y 7 en el intervalo [60, 70] %.

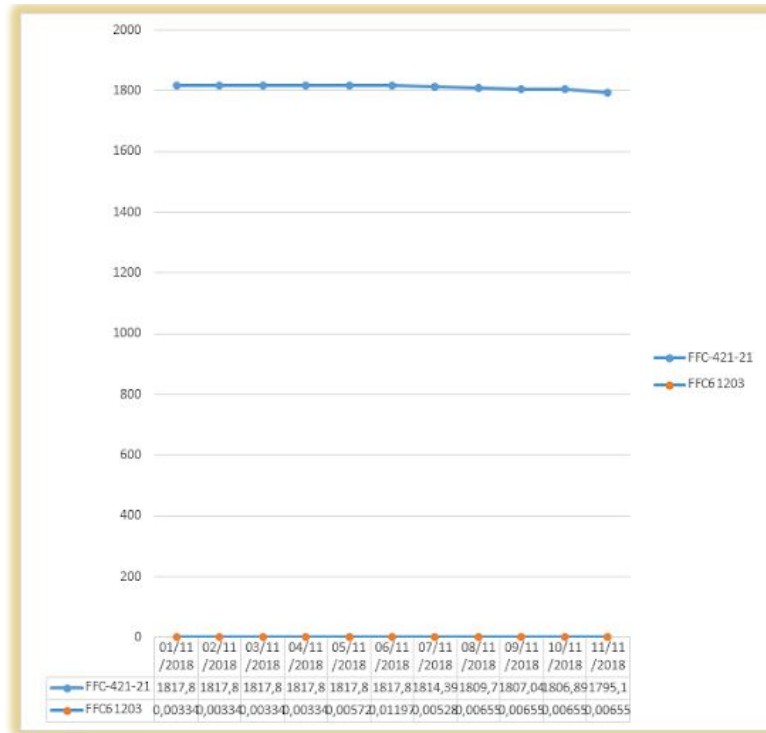
AAE

El AAE (Absolute Average Error, error absoluto medio) es en encargado de determinar la diferencia entre el valor deseado en la salida, y la salida real del sistema. Realiza un seguimiento de la capacidad del controlador para mantener su punto de consigna.

Para éste KPI, se ha calculado la diferencia en valor absoluto de la salida del sistema y la señal de referencia, y éste resultado, se ha fraccionado entre la media del valor deseado en la salida

Se visualizan los datos mediante una gráfica en la que se representa en un rango de fechas deseado los valores que toma el AAE.

Cuanto más alejados estén los valores de cero, peor mantendrá el controlador su punto de consigna.

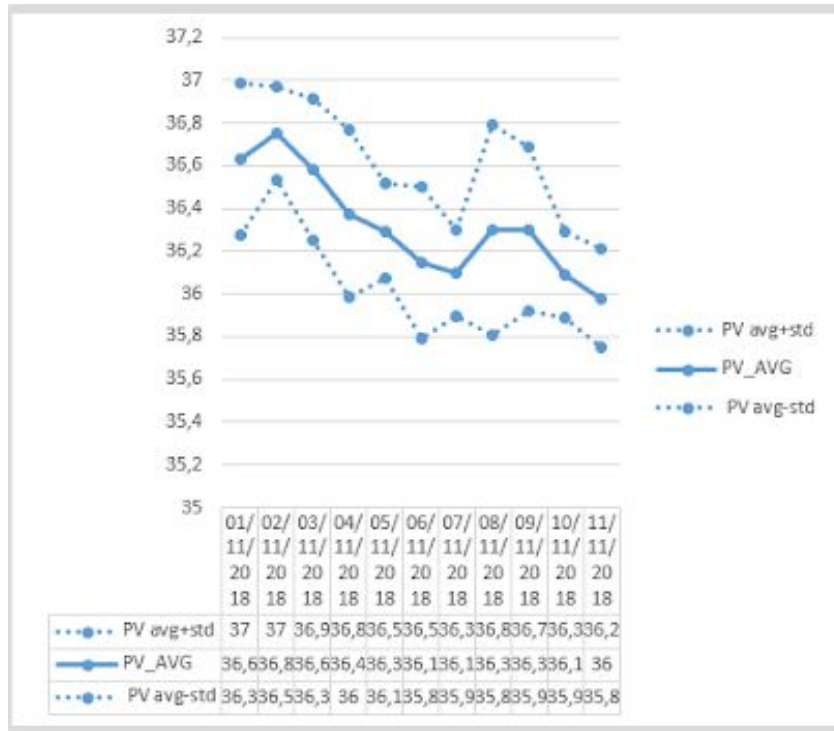


NOISE BAND

Este KPI se basa en el cálculo del ruido de alta frecuencia. Un ruido elevado podría indicar un fallo en el sensor.

En la siguiente gráfica podemos el valor promedio del PV y la desviación típica relativa a este promedio. Si la banda es ancha se podría deducir un ruido elevado del sistema.

En la siguiente gráfica, se observa un lazo de control que funciona correctamente.

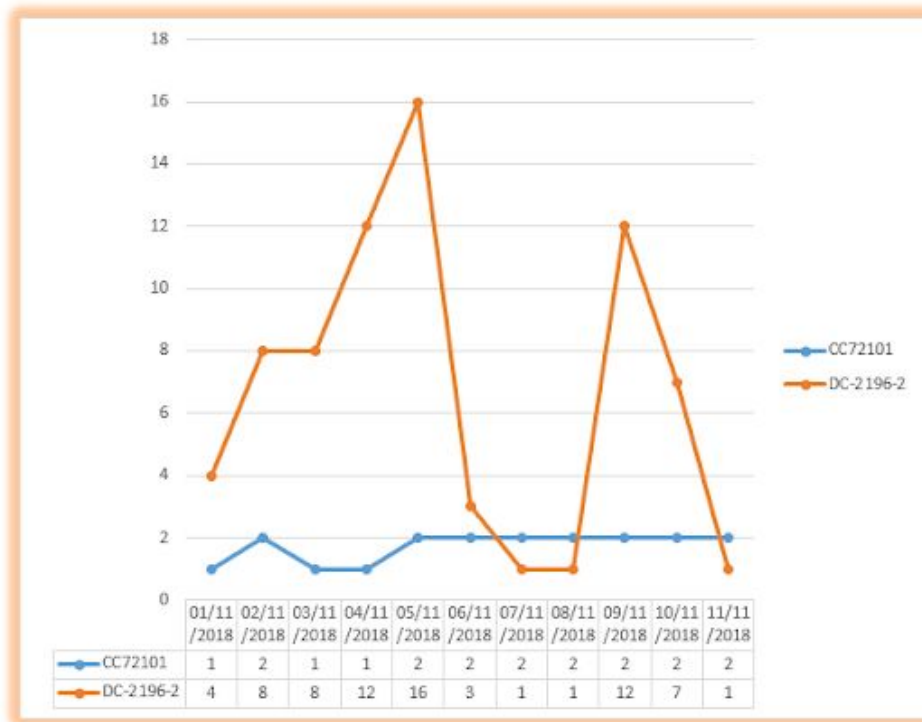


MV REVERSALS

Este KPI enumera la cantidad de veces que la salida del lazo de control (MV) cambia de dirección (ascendente o descendente). Un gran número de inversiones indica que el lazo de control realiza cambios abruptos en la salida o bien amplifica mucho el ruido.

La representación de los datos resultantes de este KPI se muestra a través de una gráfica en la que se plasma la cantidad de inversiones que hay en la fecha deseada.

En la siguiente gráfica podemos ver que el lazo CC72101 actúa de manera adecuada.

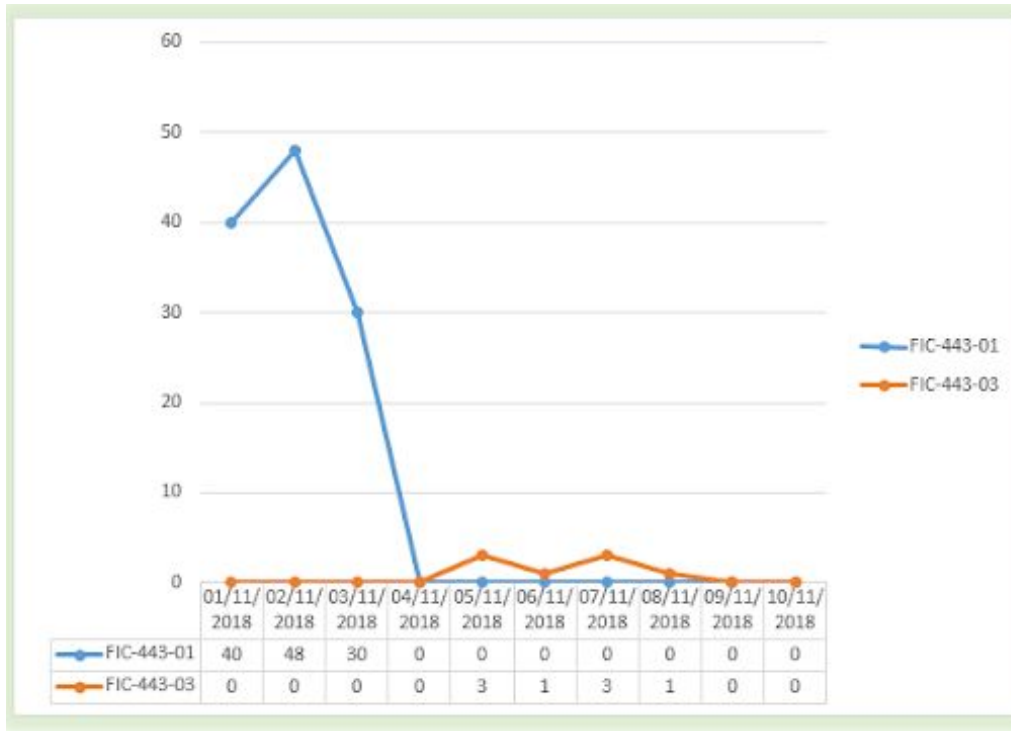


MODE CHANGES

Los lazos de control PID tienen asignado un modo de actuación. La alteración de éste de manera continuada equivaldría a problemas potenciales.

“Mode changes” cuenta las veces que cambia el modo en un rango de fechas deseado. Un número elevado de cambios indica que el lazo no funciona correctamente en modo automático de forma continuada.

Por ejemplo, se observa en la gráfica que el día 02/11/2018 el lazo FIC-443-01 cambió de modo 48 veces.

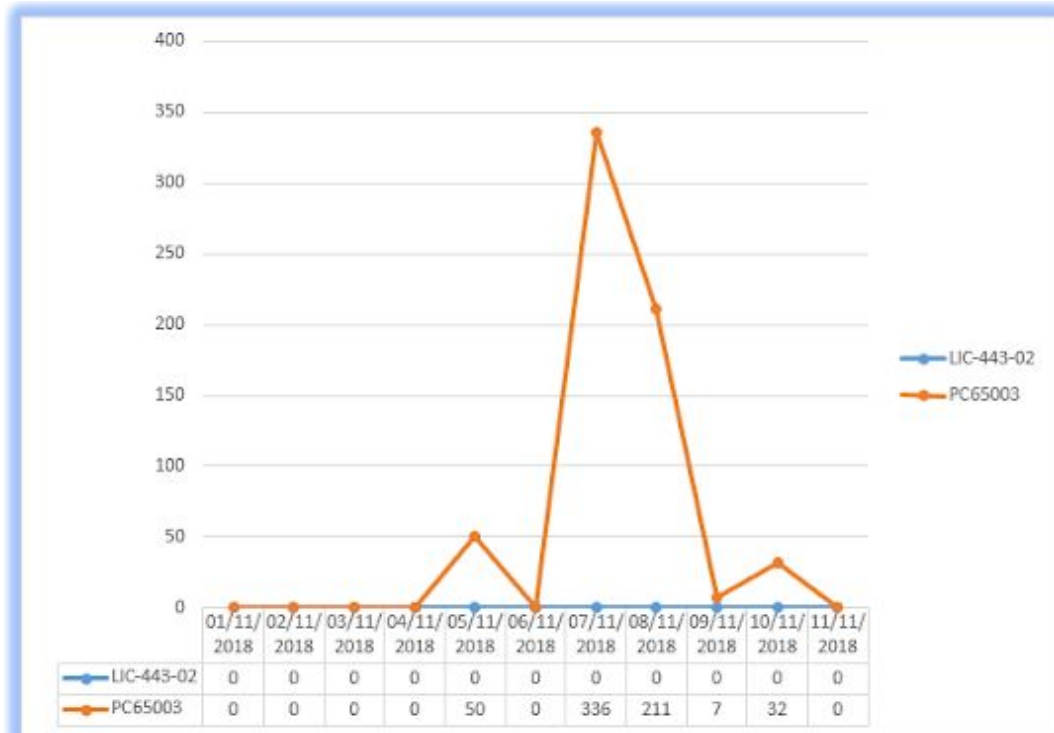


SET POINT CHANGES

Una gran cantidad de cambios en el valor de la señal de referencia indicaría una intervención abundante del operador en el proceso de control. "Setpoint Changes" se encarga de contar el número de veces por día que el punto de consigna varía.

Si el sistema se encuentra en modos cascada, éste cálculo no se realizará ya que cuando éste modo se encarga de modificar constantemente el sv.

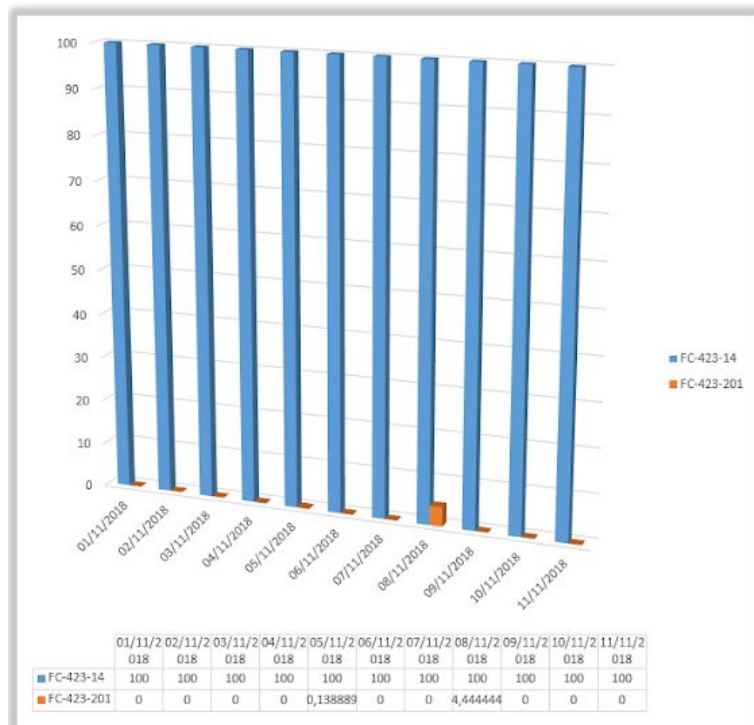
Los días 7 y 8 de Noviembre el operador intervino de manera excesiva en el control del proceso PC65003.



PERCENT TIME IN ABNORMAL

Un lazo de control PID está diseñado para funcionar en los modos automático o cascada. Percent Time In Abnormal evalúa el porcentaje de tiempo en el que el controlador está en modo distinto de cascada o automático.

El lazo FC-423-14 pasa el 100% de su tiempo en un modo distinto de cascada o automático, por lo que aparenta tener problemas potenciales.

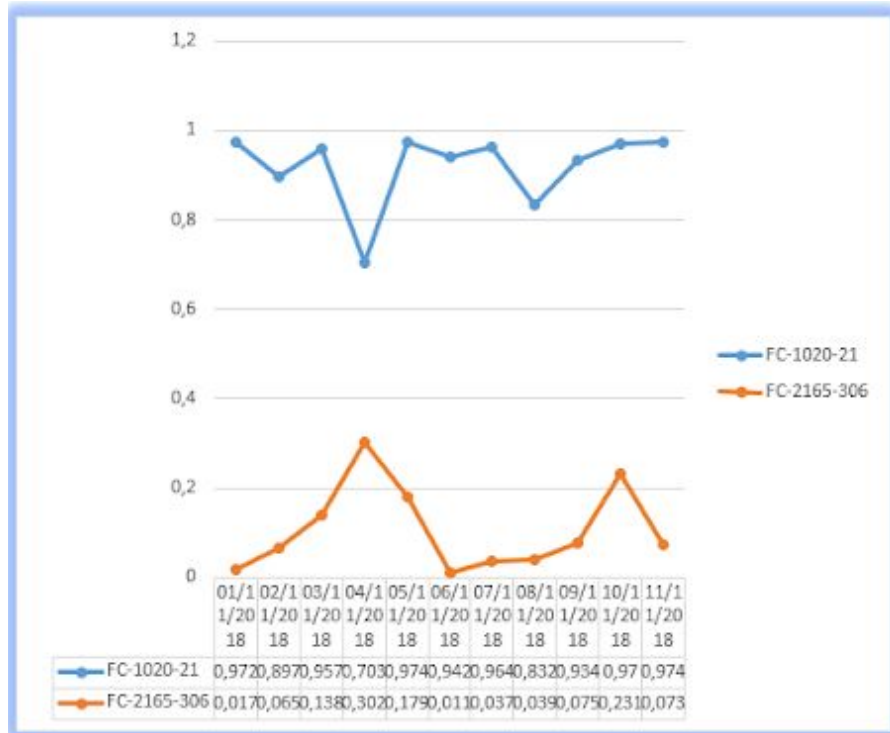


STICTION

Los componentes de los elementos de control finales pueden desgastarse con el tiempo, generando problemas en el movimiento de las válvulas. Este indicador evalúa éste desgaste.

Se calcula sólo para los lazos de control de caudal ya que generalmente existe una correlación lineal entre la posición de la válvula y el caudal trasegado. Este indicador calcula el coeficiente R^2 de la correlación de modo que si se aprecia un decremento a lo largo del tiempo, se puede intuir una pérdida de linealidad y por tanto un deterioro del elemento.

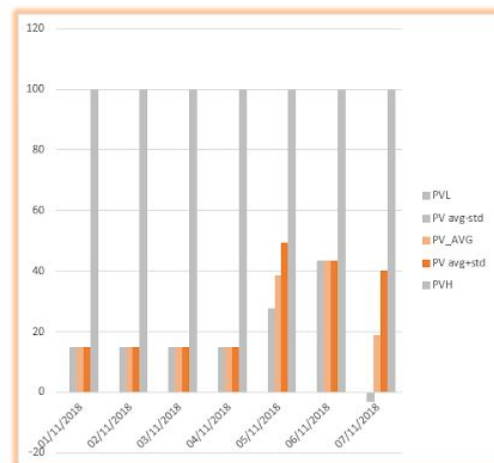
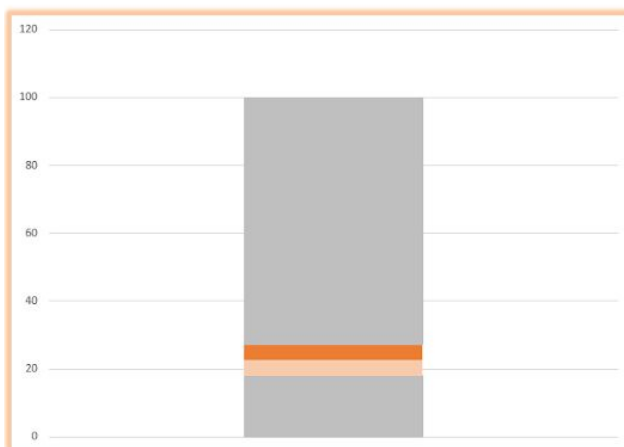
En la figura se aprecia una baja correlación en el caso del lazo de control FC-2165-306.



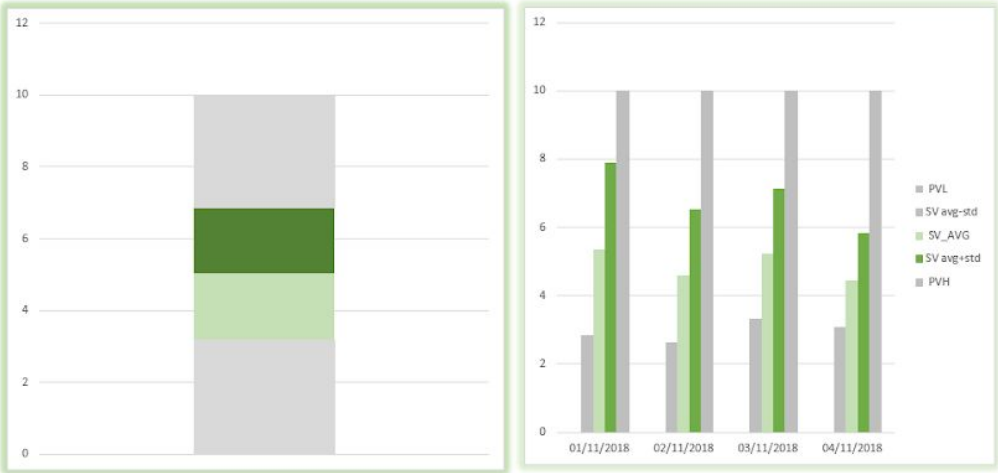
PROCESS DATA

En este apartado podemos observar el promedio, la desviación típica y los límites de la señal de referencia y de la salida del sistema, es decir, nos da una distribución estadística dentro del rango.

Mediante los gráficos podemos ver si tanto el PV como el SV varían demasiado.



Para un rango de fechas, el primer grafico te muestra la media de todos los datos durante todo el rango deseado, mientras que el segundo, te muestra una media de los datos por cada día.



Capítulo 5

Código CPM.exe

```
cat("\n > Processing raw data...")

library(RODBC)

dbhandleRAW <- odbcDriverConnect('driver={SQL Server};
server=UBEAPCENG\\SQLEXPRESS;database=UBECPM;trusted_connection=true')

dbhandleRESULTS <- odbcDriverConnect('driver={SQL Server};
server=CTOPRO\\SQLEXPRESS;database=UBECPM;uid=ube_cpm;pwd=ube_cpm')

raw <- sqlQuery(dbhandleRAW, 'SELECT * FROM raw order by 1,2')

numcasos = nrow(raw)/1440

introducir=matrix(nrow=numcasos, ncol=28)

introducir<-data.frame(introducir)

a=""
for(i in 1:numcasos){
  rangoi=1440*i+1-1440
  rangof=1440*i
  sv<- raw[rangoi:rangof,7]
  pv<- raw[rangoi:rangof,3]
```

```

mode<-raw[rangoi:rangof,12]
mv<-raw[rangoi:rangof,8]
mvl<-raw[rangoi:rangof,10]
mvh<-raw[rangoi:rangof,11]
introducir$X1[i]=paste("'",raw[rangoi,1],"',sep="")
introducir$X2[i]=paste("'",
substring(raw[rangoi,2],1,10),"',sep="")
introducir$X3[i]=paste("'",raw[rangoi,4],"',sep="")
introducir$X4[i]=paste("'",raw[rangoi,9],"',sep="")
introducir$X5[i]=raw[rangoi,5]
introducir$X6[i]=raw[rangoi,6]
introducir$X7[i]=mvl[1]
introducir$X8[i]=mvh[1]
anomalo=0
k=1
vectorcambios=c()
contadorcambiosmv=0
contmodo=0
cambiosvmodo=0
erroraae=0
valoresaae=0
for(j in 1:(length(mv)-1)){
  pos=0
  n=mv[j+1]-mv[j]
  if(n>0){ pos=1} else if(n<0){pos=-1}
  if (pos==1 || pos== -1){
    vectorcambios[k]=pos
    k=k+1
  }
  if (mode[j]=="CAS" || mode[j]=="RCAS" ||
mode[j]=="AUT"){
    erroraae=erroraae+abs(pv[j]-sv[j])
    valoresaae=valoresaae+1
  }
  if (mode[j]!='CAS' && mode[j]!='AUT'
&& mode[j]!='RCAS'){
    anomalo=anomalo+1
  }
  if (mode[j+1]!= mode[j] ){
    contmodo=contmodo+1
  }
  if (mode[j+1]=="CAS" && mode[j+1]=="RCAS"

```

```

        && mode[j+1]!="PRD" && sv[j+1]!=sv[j]){
            cambiosvmodo=cambiosvmodo+1
        }
    }
    if (mode[j+1]!='CAS' && mode[j+1]!='AUT'
    && mode[j+1]!='RCAS'){
        anomalo=anomalo+1
    }
    if (length(vectorcambios) == 1){contadorcambiosmv=1}
    else if (length(vectorcambios >1)){
        for (t in 1:(length(vectorcambios)-1)){
            if (vectorcambios[t]!=vectorcambios[t+1]){
                contadorcambiosmv=contadorcambiosmv+1
            }
        }
    }
}
aae="NULL"
if (mean(sv)!=0 && valoresaae >0) aae = (erroraae/valoresaae)
/mean(sv)
introducir$X9[i]=aae
introducir$X27[i]=(anomalo/1440)
introducir$X20[i]=contadorcambiosmv
introducir$X21[i]=contmodo
introducir$X26[i]=cambiosvmodo
introducir$X22[i]=mean(pv)
introducir$X23[i]=sd(pv)
introducir$X24[i]=mean(sv)
introducir$X25[i]=sd(sv)
if ( grepl (".FC",raw[rangoi,1],fixed=TRUE) && sd(mv)!=0
&& sd(pv)!=0 ) {
    coeficiente=cor(mv,pv)^2 }
else{
    coeficiente= " NULL " }
introducir$X28[i]=coeficiente
maxi=mvh[1]
mini=mvl[1]
if (max(mv)>mvh[1]) {
    maxi =max(mv)
}
if (min(mv)<mvl[1]){
    mini=min(mv)
}
}

```

```

rango=maxi-mini
vector=c()
vector[1]=mini
for(k in 2:11){
    vector[k]=((k-1)*rango)/10
}
frecuencia=hist(mv,breaks=vector,plot=FALSE)$counts
introducir[10:19][i,]=frecuencia
concatenar=paste(introducir[i,],collapse=",")
concatenar=paste("select ",concatenar," union all",
sep="")
a=paste(a,concatenar,collage=" ")
}
a=substring(a,1,nchar(a)-11)
consulta="INSERT into results "
con<-paste(consulta,a,sep=" ")
sqlQuery(dbhandleRESULTS, con)

```