



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Optimización Inteligente del Consumo de Cal para el
Control de Emisiones en Procesos Industriales

Autor: Carlos Larriba Rius

Director: Javier Peiro Balaguer

Madrid

Junio de 2025

Declaro, bajo mi responsabilidad, que el Proyecto presentado con el título
Optimización Inteligente del Consumo de Cal para el Control de Emisiones en Procesos
Industriales

en la ETS de Ingeniería - ICAI de la Universidad Pontificia Comillas en el

curso académico 2024/25 es de mi autoría, original e inédito y

no ha sido presentado con anterioridad a otros efectos.

El Proyecto no es plagio de otro, ni total ni parcialmente y la información que ha sido

tomada de otros documentos está debidamente referenciada.



Fdo.: Carlos Larriba Rius

Fecha: 18/ 07/ 2025

Autorizada la entrega del proyecto

EL DIRECTOR DEL PROYECTO



Fdo.: Javier Peiro Belaguer

Fecha: 18/ 07/ 2025



MÁSTER UNIVERSITARIO EN INGENIERÍA INDUSTRIAL

TRABAJO FIN DE MÁSTER

Optimización Inteligente del Consumo de Cal para el
Control de Emisiones en Procesos Industriales

Autor: Carlos Larriba Rius

Director: Javier Peiro Balaguer

Madrid

Junio de 2025

OPTIMIZACIÓN INTELIGENTE DEL CONSUMO DE CAL PARA EL CONTROL DE EMISIONES EN PROCESOS INDUSTRIALES

Autor: Larriba Rius, Carlos.

Director: Peiro Belaguer, Javier.

Entidad Colaboradora: FCC Medioambiente

RESUMEN DEL PROYECTO

En este proyecto se presenta el diseño, desarrollo e implementación de un sistema de control automático tipo PID para la dosificación de cal en el proceso de eliminación de emisiones. Todo el trabajo se ha llevado a cabo en la planta de valorización energética a partir de residuos (*Energy from Waste*, EfW) ubicada en Lincoln, Reino Unido. El control implementado supone un ahorro anual estimado de £115,000.

Palabras clave: PID, Setpoint, Energy from Waste (EfW), DCS, Machine Learning (ML), XGBoost, Correction Factor (CF), Upstream, Downstream, Lime (cal), K-Folding Cross Validation, Dataset, Feature, R^2

1. Introducción

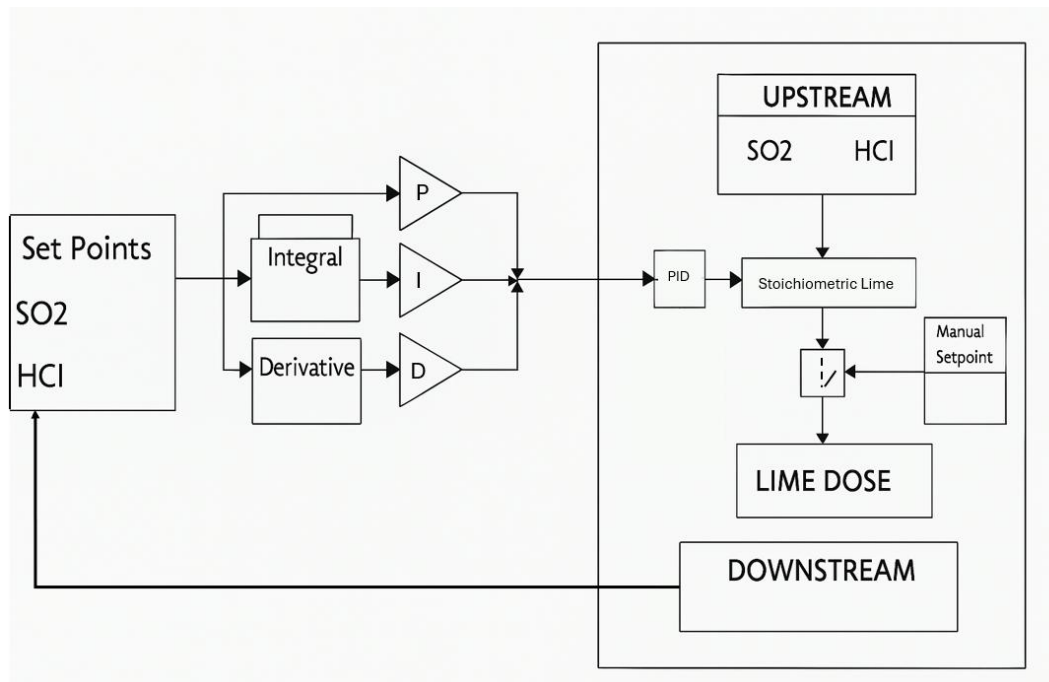
La operación de plantas de valorización energética a partir de residuos conlleva el desafío de controlar contaminantes altamente volátiles generados durante la combustión, como el ácido clorhídrico (HCl) y el dióxido de azufre (SO₂). Este Trabajo de Fin de Máster se centra en el diseño e implementación de un sistema de control automático de la dosificación de cal, el activo que elimina estos contaminantes, con el objetivo de cumplir los límites regulatorios establecidos por la agencia medioambiental británica. El sistema ha sido desarrollado en el entorno Valmet de una planta EfW ubicada en Lincoln, Reino Unido.

2. Definición del proyecto

El proyecto consiste en el diseño e implantación de un controlador PID para regular la inyección de cal como agente neutralizante de emisiones ácidas. Dado que la planta está sujeta a reportar promedios de emisiones cada 30 minutos y diariamente, el objetivo ha sido optimizar el control en esas escalas temporales, asumiendo imprecisiones aceptables a nivel minuto a minuto. Se ha buscado minimizar el riesgo de superar los límites normativos con márgenes conservadores que permiten garantizar el cumplimiento y reducir el consumo de cal.

3. Descripción del sistema de control

El sistema se basa en un PID en el que se calcula una base estequiométrica de cal cada segundo, basada en las lecturas aguas arriba de HCl y SO₂, y se multiplica por un factor de 2 para compensar la baja eficiencia de reacción química (30–40%). A esta base se le suma la acción PID. El controlador actúa sobre la cal total a inyectar, utilizando como señal de error el máximo entre los dos errores individuales (HCl y SO₂) de las lecturas aguas abajo frente a los puntos de operación fijados, lo que garantiza que ambos contaminantes se mantengan por debajo de sus límites simultáneamente.



Esquema lógico del sistema PID

4. Resultados

Durante varios días de operación continua, el nuevo controlador PID ha mantenido las emisiones de HCl y SO₂ por debajo de los límites regulatorios, cumpliendo con las medias exigidas de 30 minutos y 24 horas. Los valores finales aguas abajo se han ajustado muy cerca de los valores objetivo, especialmente en el caso del HCl, cuyo seguimiento preciso es el principal indicador de la eficacia del control. La mayor precisión frente al sistema anterior ha permitido reducir el consumo de cal medio entorno en un 29 %.

5. Conclusiones

La optimización del control ha eliminado el sobretratamiento previo, consiguiendo una reducción significativa en el consumo de cal y un ahorro económico sustancial (£115,000 anuales estimadas). Esto confirma la capacidad de un PID bien ajustado para optimizar procesos industriales complejos cumpliendo estrictas exigencias medioambientales.

SMART OPTIMIZATION OF LIME CONSUMPTION FOR EMISSION CONTROL IN INDUSTRIAL PROCESSES

Author: Larriba Rius, Carlos.

Director: Peiro Belaguer, Javier.

Collaborating Entity: FCC Environment

PROJECT SUMMARY

This project presents the design, development, and implementation of an automatic PID-based control system for lime dosing in the emission abatement process. All the work has been carried out at the Energy from Waste (EfW) plant located in Lincoln, United Kingdom. The implemented control system represents an estimated annual saving of £115,000.

Keywords: PID, Setpoint, Energy from Waste (EfW), DCS, Machine Learning (ML), XGBoost, Correction Factor (CF), Upstream, Downstream, Lime, K-Folding Cross Validation, Dataset, Feature, R^2

1. Introduction

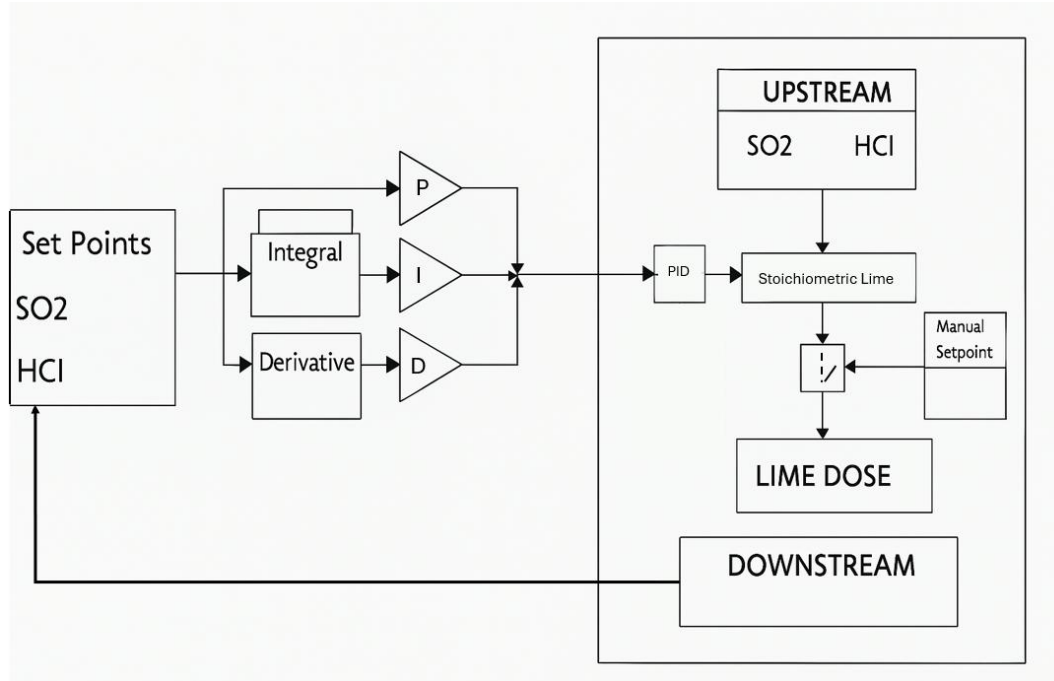
Operating energy-from-waste (EfW) plants involves the challenge of controlling highly volatile pollutants generated during combustion, such as hydrogen chloride (HCl) and sulfur dioxide (SO₂). This Master's Thesis focuses on the design and implementation of an automatic control system for lime dosing—the reagent used to eliminate these pollutants—with the objective of meeting the regulatory limits set by the UK environmental agency. The system has been developed within the Valmet environment at an EfW plant located in Lincoln, United Kingdom.

2. Project Definition

The project consists of the design and implementation of a PID controller to regulate lime injection as a neutralizing agent for acidic emissions. Since the plant is required to report emission averages every 30 minutes and daily, the objective has been to optimize control over these time scales, accepting minute-by-minute deviations as long as they remain within reasonable bounds. The aim has been to minimize the risk of exceeding regulatory limits by applying conservative safety margins that ensure compliance while reducing lime consumption.

3. Control System Description

The system is based on a PID controller that calculates a stoichiometric lime baseline every second, using upstream readings of HCl and SO₂. This baseline is multiplied by a factor of 2 to compensate for the low chemical reaction efficiency (30–40%). The PID action is then added to this baseline. The controller regulates the total lime to be injected, using as the error signal the maximum of the two individual errors (HCl and SO₂) between downstream readings and their respective setpoints. This ensures that both pollutants remain below their regulatory limits simultaneously.



Logical Diagram of the PID System

4. Results

Over several days of continuous operation, the new PID controller successfully maintained HCl and SO₂ emissions below regulatory limits, meeting both the 30-minute and 24-hour average requirements. The downstream concentrations were adjusted very close to the target values, especially for HCl, which serves as the main indicator of control effectiveness. The improved accuracy compared to the previous system enabled an average reduction in lime consumption of approximately 29 %.

5. Conclusions

The optimized control strategy has eliminated the previous overtreatment, achieving a significant reduction in lime consumption and an estimated annual cost saving of £115,000. This confirms the ability of a well-tuned PID controller to optimize complex industrial processes while meeting strict environmental requirements.

Índice de la memoria

Capítulo 1. Introducción	6
1.1 Motivación del proyecto.....	6
1.2 Contexto del problema	6
1.3 Objetivo del TFM.....	7
1.4 Rol personal y contexto del proyecto	7
Capítulo 2. Descripción de las Tecnologías.....	8
2.1 Sistema de Control Distribuido Valmet (DCS).....	8
2.2 Controlador PID	9
2.3 Modelo Predictivo	11
2.4 Variables de Control y Medición	12
Capítulo 3. Estado de la Cuestión	14
3.1 Control convencional de emisiones.....	14
3.2 Estudios relacionados y casos industriales similares.....	15
3.3 Instalación objetivo del proyecto	16
3.3.1 Funcionamiento general de la planta.....	17
3.3.2 Tratamiento de los gases de combustión.....	17
3.3.3 Control de cal previo en la planta.....	19
3.3.4 Evaluación del rendimiento.....	20
3.4 Alternativas de control avanzado	22
3.5 Oportunidad de mejora.....	23
Capítulo 4. Definición del Trabajo	24
4.1 Justificación.....	24
4.2 Objetivos	24
4.3 Metodología.....	25
4.3.1 Simulación inicial del PID en Python	25
4.3.2 Diseño y entrenamiento de un Modelo Predictivo	25
4.3.3 Implementación del PID en planta.....	25
4.3.4 Validación en planta real.....	26
4.3.5 Análisis de resultados y propuestas futuras	26

4.4	Planificación y Estimación Económica	26
4.4.1	<i>Estimación económica</i>	26
4.4.2	<i>Planificación temporal</i>	27
4.5	Análisis de riesgos	28
4.5.1	<i>Riesgos asociados al uso del PID</i>	28
4.5.2	<i>Riesgos asociados al Modelo Predictivo</i>	29
Capítulo 5.	<i>Desarrollo del sistema de control</i>	31
5.1	Análisis del Sistema previo	31
5.2	Diseño y simulación del PID	32
5.2.1	<i>Lógica del nuevo control</i>	32
5.2.2	<i>Resultados y evaluación del control simulado</i>	35
5.3	Desarrollo del modelo predictivo	38
5.3.1	<i>Identificación del problema</i>	38
5.3.2	<i>Exploratory Data Analysis (EDA)</i>	39
5.3.3	<i>Arquitectura del modelo</i>	41
5.3.4	<i>Funcionamiento del algoritmo</i>	41
5.3.5	<i>Resultados del modelo predictivo</i>	43
5.3.6	<i>Valoración técnica y abandono del enfoque predictivo</i>	44
5.4	Implementación del PID en Valmet y pruebas en lazo abierto	45
5.4.1	<i>Descripción general de la implementación</i>	45
5.4.2	<i>Arquitectura funcional y descripción de módulos</i>	46
5.4.3	<i>Resultados de las pruebas en lazo abierto</i>	48
5.4.4	<i>Conclusiones preliminares</i>	48
5.5	Diseño de la interfaz de control	49
5.6	Puesta en marcha en lazo cerrado	50
5.6.1	<i>Control de los Downstream instantáneos</i>	50
5.6.2	<i>Control de los Downstream medios en 30 minutos</i>	54
Capítulo 6.	<i>Análisis de Resultados</i>	57
6.1	Resultados operativos	57
6.2	Resultados económicos	59
Capítulo 7.	<i>Conclusiones y Trabajos Futuros</i>	61

<i>Capítulo 8. Bibliografía.....</i>	<i>63</i>
<i>ANEXO I - HAZOP.....</i>	<i>65</i>
<i>ANEXO II – Código PID.....</i>	<i>66</i>
<i>ANEXO III – Códigos Modelo Predictivo</i>	<i>69</i>
<i>ANEXO IV – Diagrama de bloques lógico</i>	<i>79</i>
<i>ANEXO V – Alineamiento con los Objetivos de Desarrollo Sostenible (ODS).....</i>	<i>81</i>

Índice de figuras

Figura 1. Planta Energy from Waste FCC, Lincoln (resource.co)	16
Figura 2. Esquema lógico del controlador previo (DCS de la planta).....	20
Figura 3. Evolución de la media diaria de HCl a lo largo de un año (DCS de la planta)....	21
Figura 4. Modulo I/O Modbus TCP ADAM – 6260 (Advantech.eu)	27
Figura 5. Diagrama de Gantt de la planificación temporal del trabajo.....	28
Figura 6. Resultados de la simulación de 10 steps del controlador en Python.....	36
Figura 7. Resultados de la simulación de 100 steps del controlador en Python.....	37
Figura 8. Mapa de calor de correlaciones entre variables candidatas a predecir el CF	40
Figura 9. Resultados de la precisión de las predicciones instantáneas frente a los valores reales medidos	43
Figura 10. Resultados de la precisión de las predicciones frente a los valores reales medidos con el horizonte temporal de 2 minutos.....	44
Figura 11. Interfaz del sistema de control (DCS de la planta)	50
Figura 12. Evolución de la dosis de cal frente a los downstreams de HCl las primeras 24 horas de funcionamiento en bucle cerrado. (DCS de la planta)	51
Figura 13. Evolución de la dosis de cal frente a los downstreams de HCl tras modificar las ganancias. (DCS de la planta).....	52
Figura 14. Evolución de la dosis de cal antes y después de la introducción del factor de confianza. (DCS de la planta).....	53
Figura 15. Evolución de la dosis de cal frente a los downstreams de HCl tras implementar el control de 30 minutos. (DCS de la planta)	55

Índice de tablas

Tabla 1. Comparativa de los resultados de las distintas configuraciones.....	53
Tabla 2. Resultados del control PI definitivo implementado	57
Tabla 3. Resultados del control previo en condiciones similares.....	58
Tabla 4. Comparativa entre los controles nuevo y antiguo	58
Tabla 5. Impacto económico de la implementación del nuevo control	59

Capítulo 1. INTRODUCCIÓN

1.1 MOTIVACIÓN DEL PROYECTO

El control eficiente de las emisiones en plantas de valorización energética es fundamental para proteger el medio ambiente y optimizar los costes operativos. En este contexto, el uso de cal como agente neutralizante desempeña un papel clave para reducir los contaminantes emitidos. Mejorar la precisión en su dosificación no solo garantiza el cumplimiento de normativas ambientales estrictas, sino que también permite obtener ahorros significativos y promueve un proceso industrial más sostenible. Este proyecto aborda ese desafío mediante soluciones prácticas con alto impacto social, económico y ambiental.

1.2 CONTEXTO DEL PROBLEMA

La planta en la que se desarrolla este proyecto se dedica a la combustión de residuos para la generación eléctrica, un proceso que genera gases residuales con presencia de compuestos contaminantes como el HCl y el SO₂. Para cumplir con la normativa ambiental vigente, las emisiones deben mantenerse por debajo de unos límites marcados por la agencia medioambiental, la cual exige reportes periódicos basados en medias móviles de 30 minutos y 24 horas.

Actualmente, la estrategia de control implementada se basa en una lógica semiautomática que aplica un ajuste manual por parte del operario y responde a las desviaciones de las medias respecto a unos *setpoints* (punto de operación) fijo. Aunque este enfoque ha permitido mantenerse dentro de los límites legales, lo ha hecho a costa de una gran imprecisión y una alta volatilidad en los resultados. Las medias reportadas presentan oscilaciones frecuentes, a veces muy por debajo del objetivo, y otras peligrosamente cerca del límite, lo que refleja una incapacidad del sistema actual para estabilizar las emisiones en torno a un valor deseado.

Esta falta de control efectivo no solo compromete la eficiencia del proceso, sino que genera un sobreconsumo de cal por sobredosificación, debido a que se opta por una estrategia excesivamente conservadora ante la falta de fiabilidad. El sistema actual no es capaz de eliminar el error de precisión de forma sistemática, y eso representa tanto un sobrecoste operativo como un riesgo de incumplimiento ambiental en escenarios menos favorables.

1.3 OBJETIVO DEL TFM

El objetivo principal de este trabajo es diseñar, desarrollar e implementar un sistema de control automático PID (aunque el controlador implementado es de tipo PI, en este documento se utilizará el término PID de forma genérica por conveniencia) para la dosificación de cal, que mejore la precisión en la reducción de emisiones de HCl y SO₂, especialmente en los intervalos auditados por la agencia medioambiental (medias diarias y medias de 30 minutos). Además, se busca optimizar el consumo de cal, logrando un ahorro económico significativo sin comprometer el cumplimiento normativo. Como complemento, se ha explorado el desarrollo de un modelo predictivo que anticipe la evolución de las emisiones para reforzar la eficacia del controlador automático.

1.4 ROL PERSONAL Y CONTEXTO DEL PROYECTO

Este proyecto se ha llevado a cabo en el marco de mi trabajo en FCC Environment EfW, Lincoln, Reino Unido, durante las prácticas universitarias realizadas desde enero hasta julio del año 2025. Este documento presenta un desarrollo independiente que introduce un controlador PID basado en el error máximo entre las emisiones de HCl y SO₂. Este enfoque ha permitido mejorar la precisión en las mediciones relevantes para el reporte ambiental y reducir notablemente el consumo de cal.

Capítulo 2. DESCRIPCIÓN DE LAS TECNOLOGÍAS

En este capítulo se describen las tecnologías, herramientas y protocolos fundamentales empleados en el desarrollo e implementación del sistema.

2.1 SISTEMA DE CONTROL DISTRIBUIDO VALMET (DCS)

El sistema Valmet DCS [1] es la plataforma central utilizada para la gestión y control del proceso industrial en la planta. Este sistema permite integrar y supervisar múltiples variables de proceso, ejecutar algoritmos de control complejos y facilitar la interacción con los operadores a través de interfaces gráficas. Está formado por:

- **Controladores distribuidos**, que ejecutan en tiempo real la lógica de regulación (bloques PID, alarmas, seguridad) próximos a sensores y actuadores.
- **Red de comunicaciones**, basada en protocolos industriales (OPC UA, Modbus, PROFIBUS), que conecta los controladores entre sí y con la sala de control.
- **Estación de operador (HMI/SCADA)**, que muestra tendencias, alarmas y parámetros de proceso, y permite la modificación de *setpoints* y la puesta a punto de los lazos de control.
- **Historiador de datos**, encargado de almacenar mediciones y eventos para posterior análisis.

En este proyecto, Valmet DCS se utilizó para implementar el controlador PID, así como para configurar la adquisición y procesamiento de señales *upstream* (aguas arriba) y *downstream* (aguas abajo) y el mando de la señal cal (o *lime*, en inglés) a inyectar.

2.2 CONTROLADOR PID

Un controlador PID (Proporcional–Integral–Derivativo) [2] es un algoritmo de regulación de procesos que ajusta una señal de control en función de tres componentes:

- **Proporcional (P):** corrige el error actual entre la variable de proceso (PV) y el *setpoint* (SP).
- **Integral (I):** corrige el error acumulado a lo largo del tiempo, eliminando desviaciones sostenidas.
- **Derivativo (D):** anticipa la tendencia del error, mejorando la estabilidad frente a cambios rápidos.

En este proyecto se implementó únicamente la parte **PI** ($K_D=0$), por su equilibrio entre sencillez y efectividad.

El controlador PI tiene los siguientes componentes:

1. Acción Proporcional (P)

Genera una corrección instantánea proporcional al **error**, definido en este proyecto como:

$$e(t) = PV(t) - SP$$

de modo que si la concentración real *downstream* (PV) supera el punto de operación (SP), el error y por tanto, la corrección, aumentan para dosificar más cal.

La contribución proporcional se expresa como:

$$u_p(t) = K_p \cdot e(t)$$

donde K_p es la ganancia proporcional, que ajusta la intensidad de la respuesta inmediata.

2. Acción Integral (I)

Se encarga de corregir los errores acumulados a lo largo del tiempo, evitando desviaciones sostenidas respecto al *setpoint*.

Su aporte viene dado por:

$$u_I(t) = K_I \cdot \sum_{i=0}^t e(i)$$

donde K_I es la ganancia integral, que determina la rapidez con la que se corrigen los errores persistentes.

La señal de control total se obtiene sumando ambas contribuciones:

$$PI = u_P(t) + u_I(t) = K_P \cdot e(t) + K_I \cdot \sum_{i=0}^t e(i)$$

En este proyecto, **PI representa el factor de corrección**, o **CF**, aplicado a la dosis de cal estequiométrica, es decir, es el inverso de la eficiencia de la reacción química. Indica qué proporción, expresada en tanto por uno, de cal es necesaria respecto a la dosis estequiométrica. La dosis de cal empleada se calcula como el producto entre la cal estequiométrica y el factor (2+CF).

Al ajustar un controlador PID, se modifican sistemáticamente las ganancias proporcionales (K_P), integrales (K_I) y derivativas (K_D , no utilizada) para adaptar la respuesta del sistema a sus características dinámicas.

La ganancia proporcional determina la rapidez de reacción al error inmediato, la integral corrige desviaciones acumuladas y la derivativa atenúa oscilaciones anticipando la tendencia del error.

2.3 *MODELO PREDICTIVO*

Un modelo predictivo [3] es un algoritmo de aprendizaje supervisado que, a partir de un conjunto de datos históricos debidamente preparados (*features*), aprende a estimar el valor futuro de una variable de interés (*target*). El proceso general consiste en:

1. Recolección y limpieza de datos

Se agrupan y depuran registros históricos de las variables relevantes, eliminando valores erróneos o inconsistentes y normalizando unidades y escalas. También denominado EDA (*Exploratory Data Analysis*) [4].

2. Selección de características (*features*)

Se identifican las variables de entrada que mejor explican el comportamiento del *target*, descartando aquellas que introducen ruido o redundancia.

3. Entrenamiento del modelo

El algoritmo ajusta sus parámetros internos para minimizar un error de predicción sobre un subconjunto de los datos (conjunto de entrenamiento).

4. Validación y ajuste de hiperparámetros

Mediante técnicas como validación cruzada K-Fold¹, se evalúa la capacidad de generalización del modelo y se ajustan sus hiperparámetros (por ejemplo, la profundidad de los árboles en un modelo de *boosting*) [5].

¹ La validación cruzada K-Fold consiste en dividir el conjunto de datos en K particiones (*folds*), entrenar el modelo K veces usando cada vez un *fold* distinto como conjunto de prueba y los restantes K-1 como entrenamiento, y promediar el rendimiento para evaluar su capacidad de generalización.

5. Inferencia

El modelo entrenado se utiliza para predecir el valor del *target* en datos nuevos, proporcionando una estimación anticipada.

En particular, en este trabajo se empleó un modelo de **XGBoost Regressor**, un algoritmo de *gradient boosting*² basado en árboles de decisión, para predecir el CF con un horizonte de varios minutos y así anticipar la dosificación de cal en el sistema de control.

2.4 VARIABLES DE CONTROL Y MEDICIÓN

El desarrollo y la validación del controlador PI y del modelo predictivo se llevaron a cabo en **Visual Studio Code** [8], un entorno de programación moderno, ligero y extensible, que ofrece:

- Editor de código con resaltado sintáctico y autocompletado, facilitando la escritura y revisión de scripts Python.
- Integración con el terminal y control de versiones (Git), para gestionar el código fuente y ejecutar comandos desde el mismo entorno.
- Extensiones específicas (Python, Pylance, Jupyter) que permiten depurar, ejecutar notebooks y analizar resultados de forma interactiva.

La lógica de simulación y el entrenamiento del modelo predictivo se implementaron en **Python**, utilizando las siguientes bibliotecas:

- Pandas: estructuras *DataFrame*³ para carga, limpieza, transformación y agregación de datos temporales.

² El *gradient boosting* es una técnica de ensamblado que construye un modelo fuerte combinando múltiples árboles de decisión secuenciales, donde cada nuevo árbol corrige los errores residuales de la suma de los anteriores, optimizando una función de pérdida mediante descenso por gradiente.

³ Un *DataFrame* es una estructura de datos bidimensional etiquetada, provista por la biblioteca Pandas de Python, que permite almacenar, manipular y analizar datos tabulares de forma eficiente.

- NumPy: operaciones vectoriales y matriciales de alto rendimiento.
- Scikit-learn: herramientas de preprocesamiento (escalado, codificación), particiones de datos y métricas de evaluación.
- XGBoost: implementación de *gradient boosting* optimizada para entrenar modelos de regresión con gran eficiencia y control de sobreajuste.
- Matplotlib y Seaborn: generación de visualizaciones estáticas para analizar tendencias y distribuciones.

Este conjunto de herramientas permitió diseñar, probar y refinar algoritmos de control y predicción antes de su transición al entorno de Valmet.

Capítulo 3. ESTADO DE LA CUESTIÓN

3.1 CONTROL CONVENCIONAL DE EMISIONES

El control de emisiones en plantas de valorización energética basadas en la incineración de residuos es un aspecto crítico debido a la liberación de compuestos ácidos como el cloruro de hidrógeno (HCl) y el dióxido de azufre (SO₂). Las regulaciones medioambientales obligan a mantener dichas emisiones por debajo de ciertos umbrales, lo que impone la necesidad de sistemas de control robustos y fiables.

Estas autoridades medioambientales —como la *Environment Agency* en el Reino Unido— exigen a estas plantas reportes periódicos de las emisiones registradas, tanto en medias diarias como en periodos de 30 minutos, bajo unas condiciones normalizadas de presión, temperatura y contenido de oxígeno (normalmente referidas a 11 % O₂ en gases secos) [9]. Además, los reportes de SO₂ y HCl son corregidos individualmente con factores que permiten a las plantas excederse en sus medidas, ya que estos reducen las cantidades reportadas. De todas formas, la exigencia normativa condiciona fuertemente el diseño de los sistemas de control y penaliza cualquier comportamiento errático o excesivamente variable en las emisiones.

El tratamiento de gases ácidos generados durante la combustión suele llevarse a cabo mediante tecnologías de inyección seca de reactivos (como cal o bicarbonato), combinadas con sistemas de filtrado como filtros de mangas (*baghouse*) o precipitadores electrostáticos. Para garantizar el cumplimiento de los límites legales de emisión, muchas instalaciones optan por sistemas de control automático, siendo los lazos PID una de las soluciones más extendidas en aquellos entornos donde existe capacidad de supervisión mediante analizadores en continuo aguas abajo.

No obstante, el uso de controladores PID no está completamente generalizado en todas las plantas. En muchas instalaciones, especialmente aquellas con sistemas heredados o

limitaciones técnicas, el control de la dosificación sigue realizándose de forma manual o semiautomática, ajustado por operarios experimentados a partir de observaciones visuales o tendencias de los analizadores.

3.2 ESTUDIOS RELACIONADOS Y CASOS INDUSTRIALES SIMILARES

Diversos trabajos han abordado la optimización de la inyección de reactivos alcalinos para la reducción de emisiones ácidas en gases de combustión, especialmente en sectores como la valorización energética de residuos, la incineración o el tratamiento de lodos.

Uno de los estudios más relevantes es el de **Mendes et al. (2021)** [10], donde se implementó un sistema de control basado en lógica difusa⁴ para regular la dosificación de cal en una planta de incineración de residuos sólidos urbanos. Los resultados mostraron una mejora del 12 % en la eficiencia del uso de reactivo frente al control proporcional tradicional.

Asimismo, **Chen et al. (2019)** [11] propusieron un sistema predictivo basado en modelos MPC⁵ para una planta de tratamiento térmico de lodos industriales. El modelo tuvo en cuenta la carga variable de contaminantes en el *upstream*, mejorando la estabilidad de las emisiones de HCl aguas abajo, con un ahorro de reactivo cercano al 18 %.

En el entorno europeo, diversas plantas integradas en redes de valorización energética han optado por enfoques híbridos. Por ejemplo, en la planta de **Malmö (Suecia)**, se ha

⁴ La lógica difusa es una extensión de la lógica clásica que permite trabajar con valores de verdad intermedios entre el 0 (falso) y el 1 (verdadero). Esta aproximación resulta especialmente útil en sistemas de control donde las condiciones no son estrictamente binarias y se requiere una representación gradual de la incertidumbre. En el contexto industrial, se emplea para modelar decisiones similares a las humanas, especialmente en procesos donde intervienen múltiples variables no lineales y difícilmente modelables de forma exacta.

⁵ El control predictivo basado en modelo (*Model Predictive Control*, MPC) es una estrategia avanzada de control que utiliza un modelo dinámico del proceso para predecir su comportamiento futuro y optimizar la acción de control en función de un criterio de rendimiento. MPC permite anticiparse a perturbaciones, manejar restricciones operativas y mejorar la estabilidad y eficiencia del sistema, lo que lo convierte en una opción especialmente valiosa en procesos con retardos significativos o dinámicas complejas.

documentado el uso de control adaptativo vinculado a sensores en continuo, con una arquitectura descentralizada de toma de decisiones que permite anticipar picos de emisión.

En general, estos estudios coinciden en que la implementación de estrategias de control avanzadas permite no solo mantener las emisiones dentro de los límites legales, sino también minimizar el coste del reactivo, uno de los principales factores operativos.

El presente trabajo se sitúa en esta línea, incorporando una lógica de control optimizada basada en el error máximo de emisión, diseñada para una planta industrial real con condiciones variables y exigencias medioambientales similares a los casos mencionados.

3.3 INSTALACIÓN OBJETIVO DEL PROYECTO

La planta en la que se ha desarrollado este proyecto es la instalación de valorización energética de residuos de Lincoln (Reino Unido), operada por FCC Environment. Esta instalación se dedica a la valorización de residuos sólidos urbanos no reciclables mediante incineración controlada, con el fin de recuperar su contenido energético. La energía térmica generada se convierte en electricidad que se inyecta a la red, contribuyendo a la producción renovable en el sistema eléctrico británico.



Figura 1. Planta Energy from Waste FCC, Lincoln ([resource.co](https://www.resource.co))

El funcionamiento de la planta se basa en un ciclo termodinámico cerrado, similar al de una central térmica convencional, pero adaptado a las particularidades de este tipo de combustible heterogéneo, con un poder calorífico variable y una importante carga contaminante.

3.3.1 FUNCIONAMIENTO GENERAL DE LA PLANTA

El proceso comienza con la alimentación del horno, donde los residuos se queman en una parrilla móvil a altas temperaturas. Esta combustión genera energía térmica que se transfiere a un circuito de agua. El agua se calienta hasta transformarse en vapor de alta presión, que se canaliza hacia una turbina de vapor conectada a un alternador. La expansión del vapor en la turbina convierte la energía térmica en energía mecánica, y finalmente el alternador transforma esta energía en electricidad que se inyecta a la red.

Este ciclo se complementa con sistemas de recuperación de calor adicionales para maximizar el rendimiento, así como con procesos auxiliares de tratamiento de gases y residuos.

3.3.2 TRATAMIENTO DE LOS GASES DE COMBUSTIÓN

La incineración de residuos genera una corriente de gases de combustión que contiene diversos contaminantes atmosféricos, entre ellos dióxido de azufre (SO_2), ácido clorhídrico (HCl), óxidos de nitrógeno (NO_x), partículas, metales pesados y compuestos orgánicos volátiles. Para cumplir con la normativa medioambiental, estos gases deben ser tratados antes de su liberación a la atmósfera.

El tratamiento de gases se realiza en varias etapas:

1. Tras el aprovechamiento de calor los intercambiadores primarios (donde se transfiere la máxima energía posible al agua en el evaporador), los gases pasan por intercambiadores de calor secundarios (economizadores), donde ceden parte de su energía térmica a otros fluidos del sistema, contribuyendo a mejorar la eficiencia global del ciclo.

2. A continuación, atraviesan el punto de medición *upstream*, el *Labloop*, donde se analiza la concentración de contaminantes antes del tratamiento. Aquí se monitorizan especialmente los niveles de SO₂ y HCl, dos de los compuestos más críticos desde el punto de vista regulatorio.
3. Justo después se produce la inyección de reactivos, donde se añaden dos componentes principales:
 - **Carbón activado (*activated carbon*)**: se utiliza para adsorber metales pesados y compuestos orgánicos.
 - **Cal (*lime*)**: reacciona químicamente con el SO₂ y el HCl para formar sales inertes no volátiles.
4. Los gases tratados se dirigen entonces a la *baghouse*, un gran filtro de mangas dividido en celdas. Aquí tiene lugar tanto la reacción química entre los contaminantes y los reactivos como el filtrado físico de partículas sólidas.
5. Tras esta etapa, los gases pasan por un analizador *downstream* que mide las concentraciones residuales de contaminantes antes de su salida definitiva por la chimenea.

Los valores medidos por el analizador *downstream* se consideran emisiones crudas o *raw*. Estos datos iniciales reflejan las concentraciones reales de contaminantes antes de cualquier corrección o normalización.

Para el control automático, las mediciones crudas se convierten a condiciones estándar de 11 % de O₂ en gas seco. A partir de los valores en crudo y de las lecturas simultáneas de oxígeno y humedad, se calculan dos medias móviles: una de 30 minutos y otra de 24 horas (media diaria). Estas medias normalizadas se retroalimentan directamente al sistema semiautomático de dosificación de cal, garantizando que la lógica de control trabaje con datos ajustados a las condiciones de referencia.

En los informes oficiales enviados a la agencia medioambiental, las mismas medias normalizadas se someten además a factores de confianza estadísticos: para el HCl se aplica una reducción adicional del 40 % y, para el SO₂, del 20 %. Estos ajustes finales permiten

cumplir con el margen de confianza exigido por la normativa, constituyendo las cifras definitivas que se reportan como cumplimiento de los límites legales. De este modo, el control de planta opera sobre las medias normalizadas SIN considerar los factores de confianza, mientras que los reportes oficiales incorporan estas reducciones para ajustarse a los criterios regulatorios.

3.3.3 CONTROL DE CAL PREVIO EN LA PLANTA

Previo al desarrollo del presente proyecto, el control de la dosificación de cal en la planta se realizaba mediante un sistema semiautomático fundamentado en controlar dos tipos de medida extraídas a partir de los valores *downstream* corregidos únicamente a condiciones de referencia de 11 % de oxígeno seco:

- **Media móvil de 24 horas:** permitía valorar la tendencia acumulada de las emisiones.
- **Valor instantáneo de los *downstreams*:** actuaba sobre las variaciones instantáneas de las emisiones.

Ambas medidas eran calculadas por separado para los dos principales contaminantes regulados: el ácido clorhídrico (HCl) y el dióxido de azufre (SO₂). La lógica de control establecía unos *setpoints* internos definidos por la planta que coinciden con los límites ambientales impuestos por la agencia medioambiental:

- Para HCl: 8 mg/Nm³ en la media diaria y 60 mg/Nm³ en la media de 30 minutos.
- Para SO₂: 40 mg/Nm³ en la media diaria y 200 mg/Nm³ en la de 30 minutos.

Las desviaciones entre las medias diarias y sus setpoints y los valores instantáneos respecto al setpoint de 30 minutos, determinaban la aplicación de unos factores multiplicativos de corrección (K_i), que modificaban la dosis estequiométrica base de cal. Esta estrategia buscaba adaptar la inyección de reactivo a las necesidades reales del sistema, incrementándola o reduciéndola en función del grado de cumplimiento ambiental observado.

Adicionalmente, el operador tenía la capacidad de introducir manualmente un coeficiente adicional (K_o), ajustable desde la sala de control, que multiplicaba el resultado del cálculo

automático. Este parámetro ofrecía un margen de intervención directa para refinar la dosificación en función de la experiencia operativa y de condiciones particulares no captadas por el sistema de medición.

El esquema lógico de la planta era el siguiente:

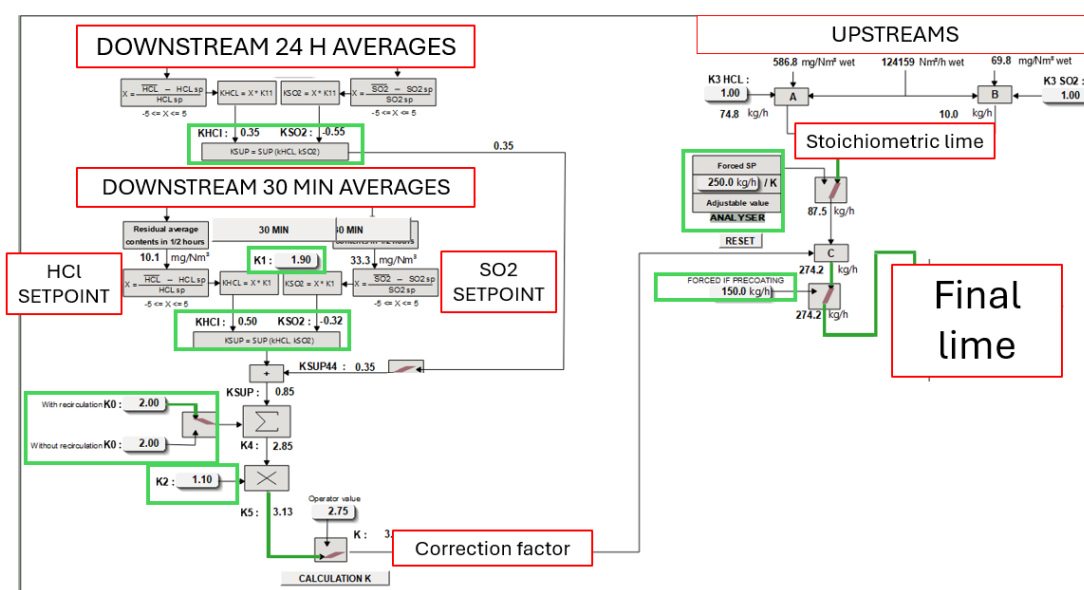


Figura 2. Esquema lógico del controlador previo (DCS de la planta)

La parte superior derecha calcula la cal estequiométrica necesaria en función de los upstream. Esta se multiplica por el CF calculado en la parte izquierda a partir de las distancias de las medias diaria y de 30 minutos de sus límites ambientales. Los recuadros verdes muestran la cantidad de factores multiplicativos de corrección (K_i) que se utilizan en este control para determinar el CF final aplicado a la cal estequiométrica. De dicha multiplicación se obtiene finalmente la dosis de cal. Resulta evidente que su estructura es desordenada y de difícil comprensión.

3.3.4 EVALUACIÓN DEL RENDIMIENTO

Tal como se ha descrito en los apartados previos, el sistema de control de cal operaba en base a medias móviles y cantidades instantáneas de contaminantes obtenidas a partir de datos corregidos a condiciones de referencia, pero sin aplicar aún los factores de confianza

exigidos por la agencia medioambiental. Esta estrategia, aunque conservadora, introducía una desviación significativa entre el valor real de las emisiones y el objetivo operativo empleado en la lógica de control.

El uso de *setpoints* internos sin considerar el factor de corrección aplicable a cada contaminante implicaba un sobredimensionamiento sistemático en la dosificación. Una vez aplicadas las reducciones del 40 % para el HCl y del 20 % para el SO₂, las emisiones reales reportadas estaban, de facto, muy por debajo de los límites establecidos, lo que evidenciaba un margen de actuación subutilizado y un consumo innecesariamente elevado de reactivo.

El análisis estadístico de los resultados muestra que, en el caso del HCl, la media diaria corregida tras aplicar los factores de confianza se situaba habitualmente en torno a los 4,8 mg/Nm³ (8 x 0,6), con oscilaciones frecuentes entre 3 y 6,5 mg/Nm³, lo que representa una desviación típica de 2,8 mg/Nm³. Esto supone un coeficiente de variabilidad del 27%, o lo que es lo mismo, una **precisión media del 69%**.

Esta variabilidad, unida a la distancia existente con respecto al umbral legal real permitido, pone de manifiesto un desempeño ineficiente tanto en términos de estabilidad como de consumo de cal.

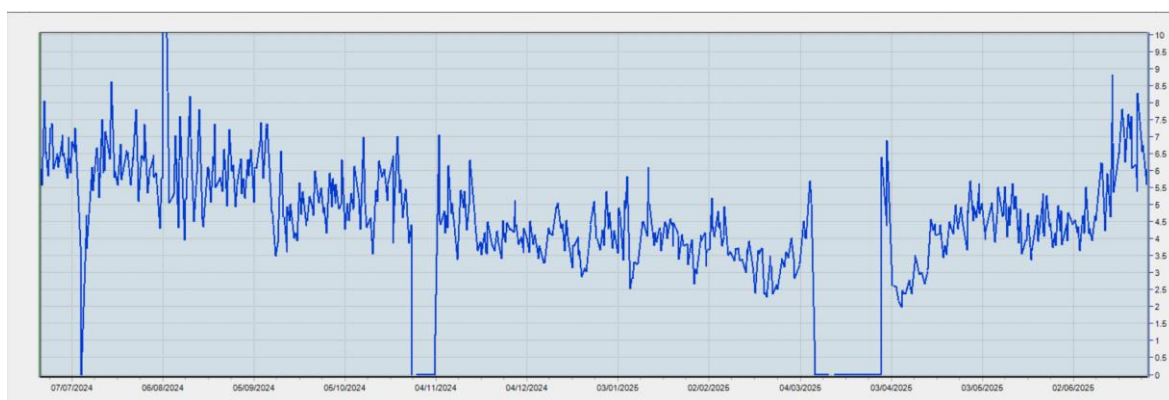


Figura 3. Evolución de la media diaria de HCl a lo largo de un año (DCS de la planta)

Como puede observarse, la media diaria de emisiones presenta una elevada variabilidad alrededor de su *setpoint* real (4,8 mg/Nm³). Excluyendo los valores nulos debidos a paradas

de planta, resulta evidente que el control actual no consigue estabilizar las emisiones, mostrando una volatilidad e imprecisión significativas.

En definitiva, el sistema previo, aunque cumplía con la normativa vigente, no lo hacía de forma optimizada. La combinación de un margen de seguridad excesivo, una alta dispersión en los resultados y una lógica de ajuste limitada motivó la necesidad de desarrollar una solución de control más precisa y adaptativa, que permitiera reducir el consumo de cal sin comprometer la garantía de cumplimiento ambiental.

3.4 ALTERNATIVAS DE CONTROL AVANZADO

El desarrollo de técnicas de control avanzado ha permitido, en ciertos sectores industriales, la incorporación de enfoques más sofisticados que los tradicionales lazos PID. Entre ellos, el Control Predictivo Basado en Modelo (MPC) destaca por su capacidad para anticipar la evolución del proceso y ajustar la acción de control en consecuencia. Este tipo de soluciones ha demostrado ser eficaz en industrias como la petroquímica, el refinado o las plantas termoeléctricas convencionales, donde los procesos son complejos, multivariables y con restricciones operativas estrictas.

Sin embargo, su aplicación práctica en plantas de incineración o valorización energética de residuos es aún limitada. La alta variabilidad de los residuos, la dificultad de construir modelos robustos y la necesidad de inversión en infraestructura adicional (tanto hardware como software) han frenado en gran medida su adopción. Del mismo modo, enfoques basados en aprendizaje automático, *Machine Learning*, o control adaptativo han mostrado potencial en estudios experimentales o en simulación (este trabajo, uno de ellos), pero aún no se han consolidado como soluciones de referencia en este tipo de instalaciones.

Por tanto, aunque existen fundamentos técnicos para considerar alternativas predictivas o inteligentes en el control de emisiones, su uso sigue siendo experimental o incipiente en este sector, y su implementación debe evaluarse cuidadosamente en función de las características específicas de cada planta.

3.5 OPORTUNIDAD DE MEJORA

Tal como se ha analizado en el apartado anterior, el sistema de control vigente cumplía con los límites regulatorios, pero presentaba una clara ineficiencia en el uso de reactivos y una falta de consistencia en los resultados. Esta situación, lejos de constituir un fallo crítico, representa una oportunidad clara para optimizar el proceso.

En este contexto, se plantea la implementación de un nuevo enfoque de control basado en un sistema PID avanzado, diseñado específicamente para esta planta. El objetivo es mantener las emisiones dentro de los márgenes regulatorios con mayor precisión y estabilidad, reduciendo al mismo tiempo el consumo de cal. La estrategia se centra en eliminar la intervención manual y adaptar el control dinámicamente a las condiciones reales del proceso, aprovechando así todo el margen operativo permitido por la normativa, es decir, utilizando los factores de confianza. Asimismo, se ha explorado el desarrollo de un modelo predictivo que anticipe la evolución de las emisiones para reforzar y complementar la acción del PID.

Capítulo 4. DEFINICIÓN DEL TRABAJO

4.1 JUSTIFICACIÓN

Como se ha visto en el capítulo anterior, el sistema de control previo a este trabajo cumplía con la normativa, pero a costa de una dosificación de cal sobredimensionada y sin precisión. Se trataba de un esquema semiautomático, basado en reglas poco refinadas y con una clara dependencia del criterio del operador. Esta forma de operar, aunque funcional, no respondía a los principios actuales de eficiencia energética, automatización o sostenibilidad económica.

Ante este contexto, el proyecto tiene pleno sentido técnico y de negocio. La posibilidad de sustituir un sistema rudimentario por un controlador PID autónomo, capaz de mantener las emisiones cerca del límite legal sin incurrir en excesos, representa una mejora significativa tanto en términos de robustez como de ahorro económico.

Por ello, se plantea una alternativa innovadora frente al típico control basado en *setpoints* conservadores: utilizar directamente el margen legal (factor de confianza), asumiendo el riesgo de forma calculada y con la precisión suficiente como para evitar incumplimientos incluso con el factor de confianza aplicado.

4.2 OBJETIVOS

- Sustituir el sistema actual de dosificación por un controlador automático, autónomo y de alto rendimiento.
- Diseñar un PID específico para este proceso, en el que la variable de error sea el máximo entre HCl y SO₂, dando prioridad al contaminante más cercano o superior al límite en cada momento.

- Implementar dicho PID en el entorno Valmet, asegurando compatibilidad con el sistema DCS actual.
- Validar el rendimiento del nuevo controlador en planta real, demostrando:
 1. Reducción en el consumo de cal.
 2. Mayor precisión y estabilidad en el seguimiento de *setpoints*.
 3. Cumplimiento continuo de los valores límite sin sobredosis sistemática.
- Explorar el potencial de integración futura con un modelo predictivo basado en *Machine Learning* como asistencia al PID.

4.3 METODOLOGÍA

La ejecución del proyecto se ha dividido en varias etapas:

4.3.1 SIMULACIÓN INICIAL DEL PID EN PYTHON

Creación de un controlador que utiliza como error la máxima desviación (sin valor absoluto) entre las emisiones y sus respectivos *setpoints*. Pruebas sobre datos simulados para entender el comportamiento dinámico de las emisiones.

4.3.2 DISEÑO Y ENTRENAMIENTO DE UN MODELO PREDICTIVO

Desarrollo de un modelo predictivo para anticipar el CF aplicado a la dosis de cal, de modo que el controlador pueda compensar los retardos del proceso y optimizar la sincronización entre la acción de dosificación y su efecto real en las emisiones.

A pesar de obtener buenos resultados de predicción, se concluyó que el modelo no era aplicable directamente como sustituto del controlador en tiempo real, lo que llevó a retomar el enfoque PID.

4.3.3 IMPLEMENTACIÓN DEL PID EN PLANTA

Traducción del diseño del PID original al entorno real de la planta. Configuración de bloques funcionales, interconexión de señales y puesta a punto del sistema en coordinación con el

equipo de control. Diseño de una interfaz entendible y funcional para los operadores de la planta.

4.3.4 VALIDACIÓN EN PLANTA REAL

Período de prueba prolongado (varias semanas), durante el cual se monitorizó el comportamiento del nuevo sistema en comparación con el anterior. Se analizaron medias móviles de 30 minutos y 24 horas, consumo de cal y cumplimiento de límites y se realizaron los ajustes necesarios para determinar el modelo PID definitivo, principalmente, modificando las ganancias proporcional (K_P) e integral (K_I), además de algunos cambios en la lógica general.

4.3.5 ANÁLISIS DE RESULTADOS Y PROPUESTAS FUTURAS

Definición del PID definitivo que controlará en sistema de cal en la planta y evaluación cuantitativa del rendimiento y planteamiento de futuras líneas de mejora, como la combinación del modelo predictivo con el PID implementado.

4.4 PLANIFICACIÓN Y ESTIMACIÓN ECONÓMICA

4.4.1 ESTIMACIÓN ECONÓMICA

El proyecto no ha supuesto ningún coste económico directo, ya que todas las herramientas empleadas han sido de uso libre o ya disponibles en la planta. Las simulaciones se han desarrollado en entornos de programación gratuitos como Python, utilizando bibliotecas como pandas, xgboost o scikit-learn, y la implementación final del PID se ha realizado directamente sobre el DCS de Valmet ya presente en la planta. No ha sido necesaria la adquisición de hardware, licencias adicionales ni servicios externos.

En caso de haberse implementado, o querer implementarse el Modelo Predictivo, sí habría sido necesaria la inversión en un equipo capaz de establecer una comunicación tipo Modbus⁶ entre el DCS y el ordenador externo donde se ejecutase en modelo. Este dispositivo fue identificado por uno de los operadores de la planta y el modelo propuesto es un *ADAM – 6260 Modbus TCP Module*, valorado en unos 210 £.



Figura 4. Módulo I/O Modbus TCP ADAM – 6260 (Advantech.eu)

4.4.2 PLANIFICACIÓN TEMPORAL

La planificación del proyecto se extendió desde finales de enero hasta finales de junio de 2025. A lo largo de ese periodo se llevaron a cabo distintas fases: desde la comprensión del sistema actual y simulación en Python, hasta el desarrollo de un control PID y un modelo predictivo y, finalmente, la implementación del PID definitivo.

⁶ *Modbus* es un protocolo de comunicación serial estándar, utilizado principalmente en sistemas industriales para transmitir datos entre dispositivos electrónicos como controladores programables (PLC), sensores y sistemas de supervisión

En la siguiente figura se presenta un diagrama de Gantt con la distribución temporal de las actividades principales del proyecto:

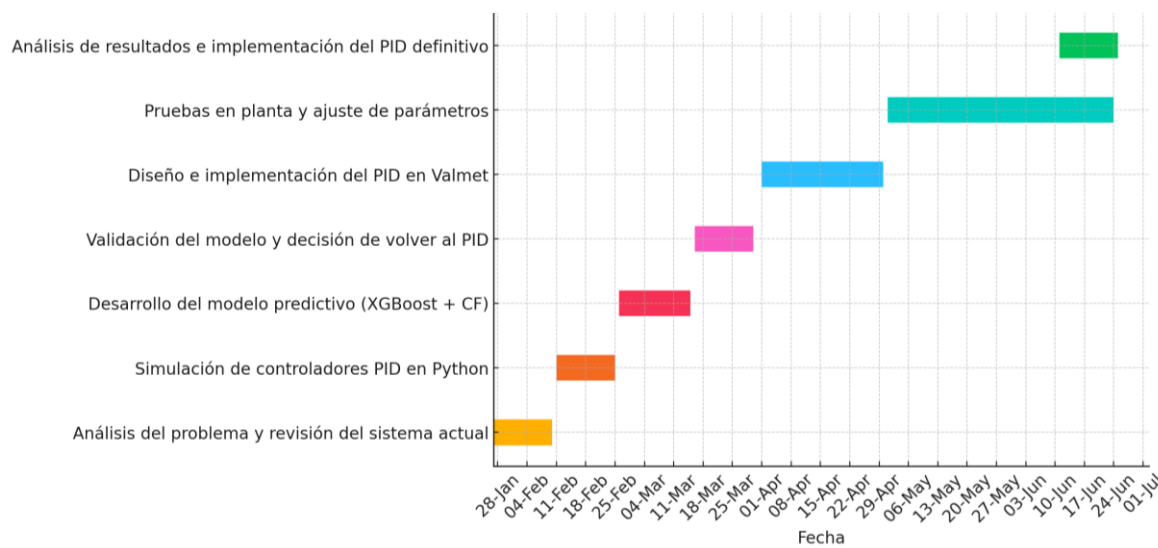


Figura 5. Diagrama de Gantt de la planificación temporal del trabajo

Esta planificación permitió un avance ordenado, asegurando la validación de cada fase antes de pasar a la siguiente, y culminando con la implementación exitosa del nuevo controlador PID en planta.

4.5 ANÁLISIS DE RIESGOS

Antes de implementar cualquier cambio en el sistema de dosificación de cal en la planta, se realizó un análisis exhaustivo de los posibles riesgos asociados a las modificaciones propuestas. Este análisis se divide en dos bloques principales, correspondientes a las dos alternativas de control planteadas durante el proyecto:

4.5.1 RIESGOS ASOCIADOS AL USO DEL PID

- **Impacto sobre el sistema de dosificación:** Al tratarse de un cambio en la lógica de cálculo del dosificado de cal dentro del DCS, el riesgo principal es que una mala

configuración o fallo en el controlador provoque una sobredosificación o infradosificación.

- **Sobredosificación:** Un exceso de cal podría provocar bloqueos o acumulación en la zona de filtrado de gases, la *baghouse*, afectando su correcto funcionamiento y generando posibles paradas o fallos en la planta. Además, aumentaría los costes operativos y el riesgo de generar residuos excesivos.
- **Infradosificación:** Riesgo de superar los límites de emisión de HCl y SO₂, con implicaciones legales y medioambientales.
- **Fallo en el sistema de control:** Se considera bajo gracias a la robustez del DCS y los controles de seguridad existentes, pero se han planificado pruebas en lazo abierto para validar el comportamiento antes de cerrar el control.

4.5.2 RIESGOS ASOCIADOS AL MODELO PREDICTIVO

Este enfoque, además de compartir los mismos riesgos que el PID al implicar un cambio en la lógica de inyección, presenta riesgos adicionales derivados de tratarse de una tecnología completamente distinta:

- **Complejidad tecnológica:** La introducción del modelo predictivo implica una arquitectura más compleja que requiere monitoreo continuo y mantenimiento especializado, aumentando la probabilidad de errores operativos.
- **Dependencia de datos y precisión:** Si el modelo predictivo falla o presenta desviaciones significativas, el sistema puede reaccionar incorrectamente a condiciones reales, provocando dosificaciones erróneas.
- **Tiempo de respuesta:** La latencia del modelo y la necesidad de cálculos avanzados pueden dificultar la respuesta rápida ante variaciones de la planta, afectando la estabilidad del control.
- **Gestión del cambio:** Implementar un paradigma nuevo en la planta puede generar resistencia operativa y requiere formación adicional, incrementando el riesgo de errores humanos.

Estos riesgos se han recogido en un documento interno llamado HAZOP (*HAZard and OPerability Study*) que se encuentra en el ANEXO I al final del proyecto.

Capítulo 5. DESARROLLO DEL SISTEMA DE CONTROL

Este capítulo describe el desarrollo completo del sistema de control propuesto, desde la caracterización inicial del problema hasta su implementación en el entorno real de planta. A lo largo de este proceso, se han abordado distintas fases: análisis del sistema actual, diseño de una primera solución PID en entorno Python, exploración de un modelo predictivo como alternativa, retorno al enfoque del PID con su diseño en el DCS, y finalmente, su implementación, validación y ajuste fino en el sistema real de la planta.

Cada etapa se ha diseñado en respuesta directa a los problemas observados en el control anterior: inestabilidad en las medias de emisión reportadas, sobredosificación de reactivo y ausencia de un sistema autónomo y preciso. A partir de estas necesidades, se ha ido refinando la solución con un enfoque iterativo y práctico.

A continuación, se describen de forma ordenada todas las fases de este desarrollo.

5.1 ANÁLISIS DEL SISTEMA PREVIO

Antes de abordar el diseño e implementación de un nuevo controlador, se realizó un estudio exhaustivo del sistema de tratamiento de gases y del esquema de dosificación de cal existente tal y como se describe en el capítulo 3.3 de este documento.

El resumen es que:

La planta de valorización energética de Lincoln quema residuos sólidos urbanos para generar vapor y electricidad y, en el control de las emisiones, emplea un sistema semiautomático de dosificación de cal basado en medias móviles de 30 min y 24 h, ajustadas manualmente por el operario. Este enfoque, aunque garantizaba el cumplimiento de los límites diarios y de

media hora (gracias a márgenes conservadores y al factor de confianza aplicado en los informes), presentaba una elevada volatilidad en las emisiones reportadas, con frecuentes fases de infra- y sobredosificación, y no compensaba adecuadamente los retardos de varios minutos entre la inyección de cal y su efecto medido en los analizadores *downstream*.

A partir de este diagnóstico —que puso de manifiesto la necesidad de un control automático más preciso, estable y con menor dependencia del operario— se establecieron los requisitos técnicos para el nuevo sistema. Entre ellos figuraron:

1. Gestionar las latencias temporales de hasta 5 min en el lazo de realimentación.
2. Orientar el control a las medias oficiales (30 min y 24 h) en lugar de al valor instantáneo.
3. Garantizar una dosificación optimizada para minimizar el consumo de reactivo sin comprometer el cumplimiento ambiental.

Con estos criterios, la primera etapa de desarrollo consistió en la implementación de un controlador PID discreto en Python, simulando su comportamiento sobre datos históricos de la planta para evaluar su capacidad de corrección dinámica y sentar las bases del modelo que se detallará a continuación.

5.2 DISEÑO Y SIMULACIÓN DEL PID

5.2.1 LÓGICA DEL NUEVO CONTROL

Como primer paso en el desarrollo del nuevo sistema de control, se diseñó un PID en Python con el objetivo de evaluar su respuesta ante datos reales y comprender cómo reaccionaría una lógica automática sencilla en un entorno simulado.

La lógica de este controlador se basa en el **cálculo de un factor de corrección (CF)** a partir del error máximo entre las medidas de emisión (*downstream*) de HCl y SO₂ instantáneas y sus respectivos *setpoints*. Este factor se aplica sobre la dosis estequiométrica de cal, que se

calcula previamente a partir de la concentración de HCl y SO₂ (*upstream*) y el caudal de gases. La ecuación general empleada para obtener la dosis real de cal en fue:

$$Dosis_{real} \left[\frac{kg}{h} \right] = Dosis_{estequiométrica} \times (1 + CF)$$

La dosis estequiométrica de cal se calcula, al igual que en el control original, a partir de las concentraciones de HCl y SO₂ medidas en los gases y el conocimiento de las reacciones químicas que tienen lugar. La ecuación utilizada fue:

$$Dosis_{estequiométrica} \left[\frac{kg}{h} \right] = \left(\frac{Q_{HCl}}{(36.45 \cdot 2)} + \frac{Q_{SO2}}{64} \right) \cdot 74 \cdot \frac{Caudal}{10^6}$$

Donde:

- Q_{HCl} y Q_{SO2} son las concentraciones *upstream* y están en mg/Nm³.
- $(36.45 \cdot 2)$, 64 y 74 representan las masas molares de los respectivos compuestos que reaccionan estequiométricamente.
- Caudal de gases está en Nm³/h.
- El divisor 10⁶ convierte de mg a kg.

El CF se calculaba mediante la formulación PID, previamente descrita. Así, el controlador implementado fue:

$$CF = PI = K_p \cdot e(t) + K_I \cdot \sum_{i=0}^t e(i)$$

donde:

- $e(t)$ es el **error máximo** entre los valores *downstream* de HCl y el SO₂, respecto a sus respectivos *setpoints*, en el instante t .
- K_p , K_I son las ganancias proporcional e integral, respectivamente.
- El operador $\sum_{i=0}^t e(i)$ representa una suma acumulada de errores anteriores.

Este planteamiento garantiza que, si las emisiones están por encima del límite deseado (es decir, $PV > SP$), el error será positivo, lo que incrementará el *correction factor* CF y, en consecuencia, aumentará la dosis de cal para contrarrestar el exceso de contaminantes.

La razón de multiplicar por $(1+CF)$ en vez de solo CF radica en la asunción de que la reacción nunca será 100% eficiente y por ende siempre requerirá algo más. Se espera por tanto un CF positivo.

Los segmentos más relevantes del código utilizado en Python son los siguientes:

```
def calculate(self, downstream_so2, downstream_hcl):
    error = max(((downstream_so2 - self.setpoint_so2)/self.setpoint_so2),
                ((downstream_hcl - self.setpoint_hcl)/self.setpoint_hcl))
    P = self.Kp * error

    self.integral += error
    I = self.Ki * self.integral

    D = self.Kd * (error - self.prev_error)

    self.prev_error = error

    output = P + I + D
    print(f'PID: {output:.2f}, Error: {error:.2f}, Integral:
{self.integral:.2f}')
    return output
```

Cálculo del CF en Python

```
def lime_dosing_control(upstream_SO2, upstream_HCl, setpoint_SO2, setpoint_HCl,
fg_flow, cf):
    lime_needed_SO2 = (((upstream_SO2 - setpoint_SO2) * fg_flow/10**6) / 64) * 74
    lime_needed_HCL = (((upstream_HCl - setpoint_HCl) * fg_flow/10**6) / (36.45 *
2)) * 74
    stoich_lime = lime_needed_HCL + lime_needed_SO2
    lime_needed = stoich_lime * (1+cf)
    print(f'Stoich lime: {stoich_lime:.2f}')
    print(f'Lime used:{lime_needed:.2f}')
    return lime_needed
```

Cálculo de dosis de cal total en Python

Al tratarse de un sistema simulado, también es necesario simular los valores *downstream*. Para ello, se han tenido en cuenta unas eficiencias individuales estimadas de HCl y SO₂ del

25% y 5% respectivamente, calculadas a partir de los datos históricos de estos compuestos y también contrastadas con los ingenieros de la planta.

El cálculo se realiza mediante la diferencia entre las concentraciones upstream y las correspondientes a los compuestos neutralizados por la dosis de cal del ciclo anterior. Para ello, se emplea la reacción estequiométrica inversa: a partir de la cantidad de cal aplicada, se determina el equivalente de contaminante neutralizado y se multiplica por las eficiencias antes descritas. Por seguridad y coherencia física, se limitaron estos valores a números positivos. En el código, se representa de la siguiente forma:

```
efficiency_so2 = 0.05
efficiency_hcl = 0.25

neutralized_SO2 = max(((lime_dosing * 64) / 74) * efficiency_so2, 0)
neutralized_HCL = max(((lime_dosing * (36.45 * 2)) / 74) * efficiency_hcl,
0)

downstream_SO2_kg_h = max(0, upstream_SO2_kg_h - neutralized_SO2)
downstream_HCL_kg_h = max(0, upstream_HCL_kg_h - neutralized_HCL)

downstream_SO2 = downstream_SO2_kg_h / (fg_flow / 10**6)
downstream_HCL = downstream_HCL_kg_h / (fg_flow / 10**6)
```

Simulación de los downstream en Python

El código completo se encuentra en el [ANEXO II](#).

5.2.2 RESULTADOS Y EVALUACIÓN DEL CONTROL SIMULADO

Una vez implementado el controlador PID en entorno Python, se procedió a realizar diversas simulaciones para validar su comportamiento. Estas pruebas revelaron que el sistema respondía de forma precisa en la corrección del error, ajustando adecuadamente el *correction factor* (positivo) y, en consecuencia, la dosificación de cal. Las ganancias que demostraron el mejor comportamiento en las simulaciones fueron:

- $K_p = 0.01$
- $K_I = 0.04$

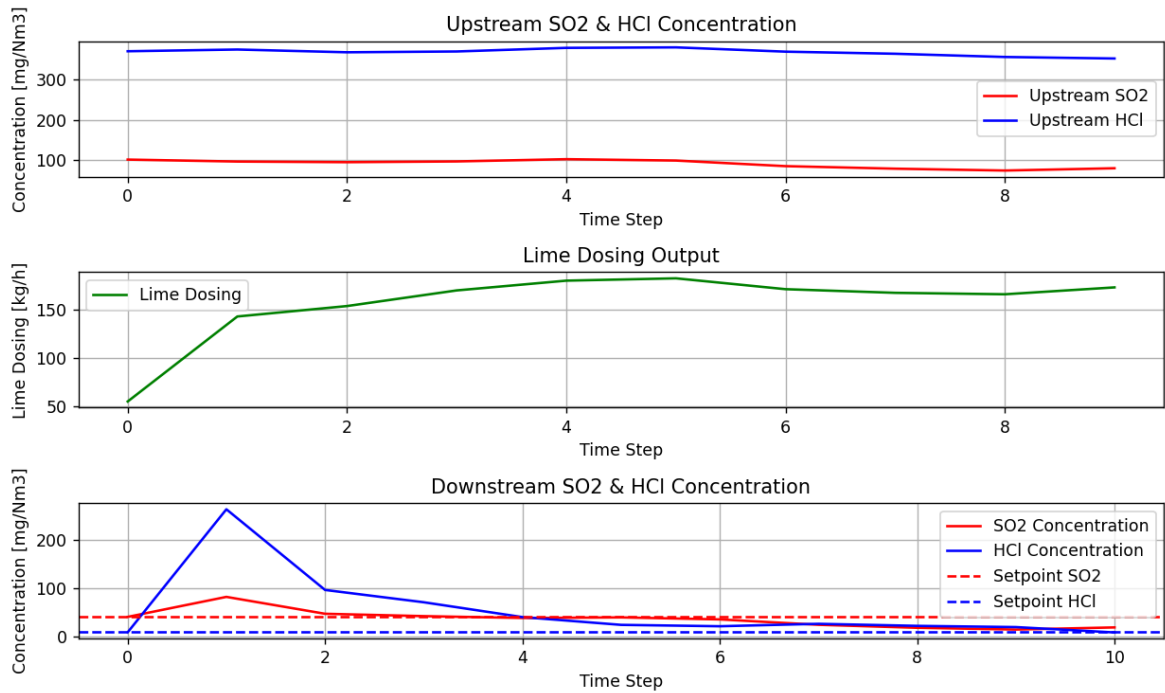


Figura 6. Resultados de la simulación de 10 steps del controlador en Python

La respuesta del sistema en los primeros *steps* es correcta y demuestra que se alcanza el *setpoint* en un tiempo aceptable (unos 10 *steps*).

Sin embargo, la estabilidad del sistema no era óptima a largo plazo, en gran parte debido a que los valores de emisiones *downstream* empleados en la simulación no reflejaban de forma realista los retardos y la dinámica de la planta. Al no contar con un *feedback* real que reprodujera fielmente el comportamiento físico del sistema, las simulaciones no eran suficientes para garantizar la robustez del controlador ante perturbaciones reales.

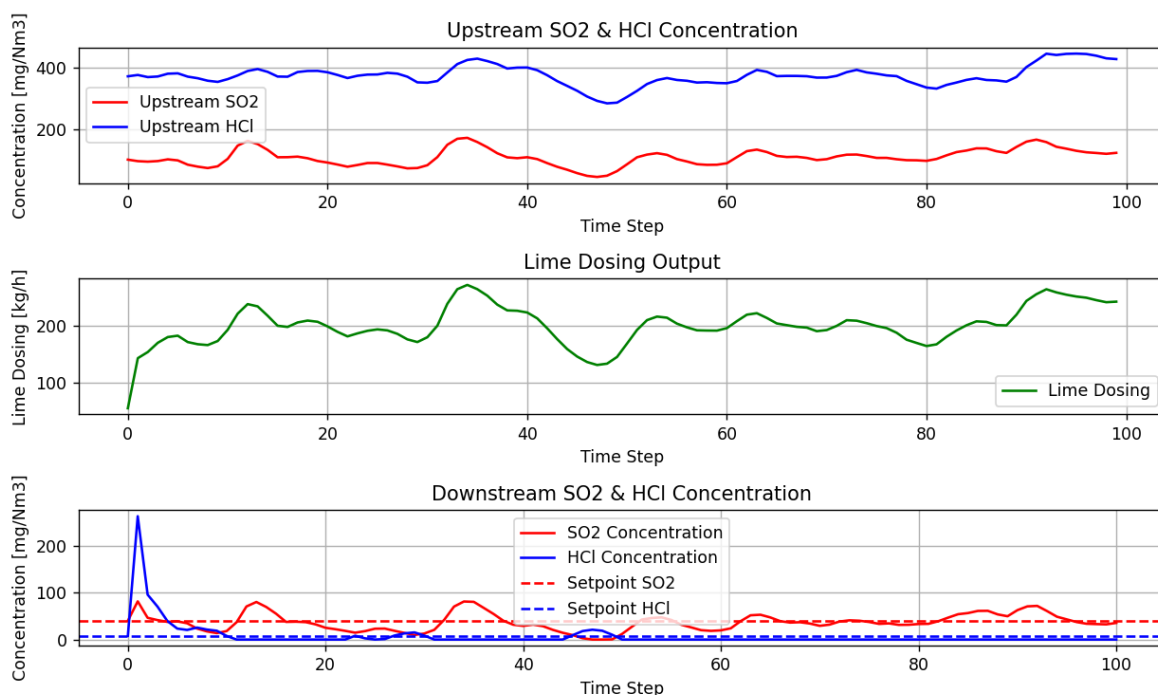


Figura 7. Resultados de la simulación de 100 steps del controlador en Python

Los resultados mostraron que el control no es capaz de mantener ambos compuestos por debajo de los límites de forma constante. Las reacciones del control no eran del todo coherentes (crecimientos del *downstream* poco justificados según los *upstream*). Todo parecía indicar que las simulaciones del *feedback* no reflejaban el comportamiento real de la planta.

A pesar de ello, los resultados preliminares indicaban que el enfoque PID era potencialmente viable y que su rendimiento potencial podía ser alto. Pero quedó claro que las pruebas en tiempo real serían determinantes para validar la consistencia del algoritmo y ajustar K_P y K_I para obtener una respuesta adecuada.

Sin embargo, durante esta etapa, surgió una reflexión clave: la ausencia de una simulación precisa de los retardos del sistema planteaba dudas razonables sobre la capacidad de un PID clásico para anticiparse a los efectos reales de la dosificación. Esto llevó a considerar la posibilidad de utilizar un modelo predictivo que permitiera anticipar la evolución de las

emisiones y compensar las latencias, lo cual daba pie a explorar un nuevo enfoque para el control.

5.3 *DESARROLLO DEL MODELO PREDICTIVO*

5.3.1 IDENTIFICACIÓN DEL PROBLEMA

Una vez diseñado y validado el comportamiento del controlador PID en el entorno de simulación en Python, surgió un nuevo desafío al plantear su implementación en el sistema real: la existencia de retardos significativos en la respuesta del sistema que no habían sido considerados inicialmente en la etapa de modelado.

En primer lugar, se identificó un retardo físico en el sistema de transporte de cal. Desde que se emite la orden de dosificación hasta que la cal llega efectivamente al punto de inyección transcurre un tiempo estimado de aproximadamente dos minutos, debido a la distancia y dinámica del transportador desde el silo hasta el conducto de gases.

A este retardo físico se suma un segundo componente: el retardo químico asociado a la reacción entre la cal inyectada y los contaminantes presentes en el flujo gaseoso. El análisis de los datos históricos reveló que los efectos de un cambio en la concentración de contaminantes medido *upstream* no se reflejan en la medición *downstream* hasta transcurridos al menos otros dos minutos adicionales. Esto implica que desde el momento en que se inyecta la cal hasta que se puede observar su efecto real en las mediciones *downstream* transcurren en total unos **cuatro minutos de retardo**.

Este comportamiento supone una limitación crítica para la aplicación directa de un controlador PID clásico. Dado que el sistema trabaja con un tiempo de muestreo de 30 segundos, el controlador estaría tomando decisiones basadas en información desfasada más de ocho ciclos de control, lo que comprometería seriamente su capacidad de respuesta y ajuste.

Ante esta situación, se planteó una solución alternativa basada en técnicas de *machine learning*: el diseño de un modelo predictivo capaz de estimar el valor futuro del *Correction Factor*, es decir, el valor que debería adoptar el PID no en el momento actual, sino dentro de cuatro minutos, cuando la cal que se dosifique en el presente tenga efecto medible en los sensores *downstream*. De este modo, sería posible calcular la cantidad de cal estequiométrica a inyectar y multiplicarla por un CF anticipado, mejorando así la sincronización entre causa (dosificación) y efecto (reducción de emisiones).

5.3.2 EXPLORATORY DATA ANALYSIS (EDA)

Se partió de históricos de datos reales exportados desde el sistema de adquisición de planta, con miles de registros que incluían variables como:

- Concentraciones de HCl y SO₂ *upstream* y *downstream*
- Dosis de cal inyectada
- Temperaturas, presiones y más variables entorno

El primer paso fue una limpieza y análisis detallado del *dataset*⁷. Se eliminaron aquellas variables que no aportaban valor al modelo, tenían elevada correlación entre sí, o que introducían ruido, y se seleccionaron como *features* aquellas que demostraban correlación con la evolución del CF.

⁷ Un *dataset* es un archivo de texto digital en el que se almacenan valores históricos de diferentes variables a lo largo de un periodo de tiempo determinado.

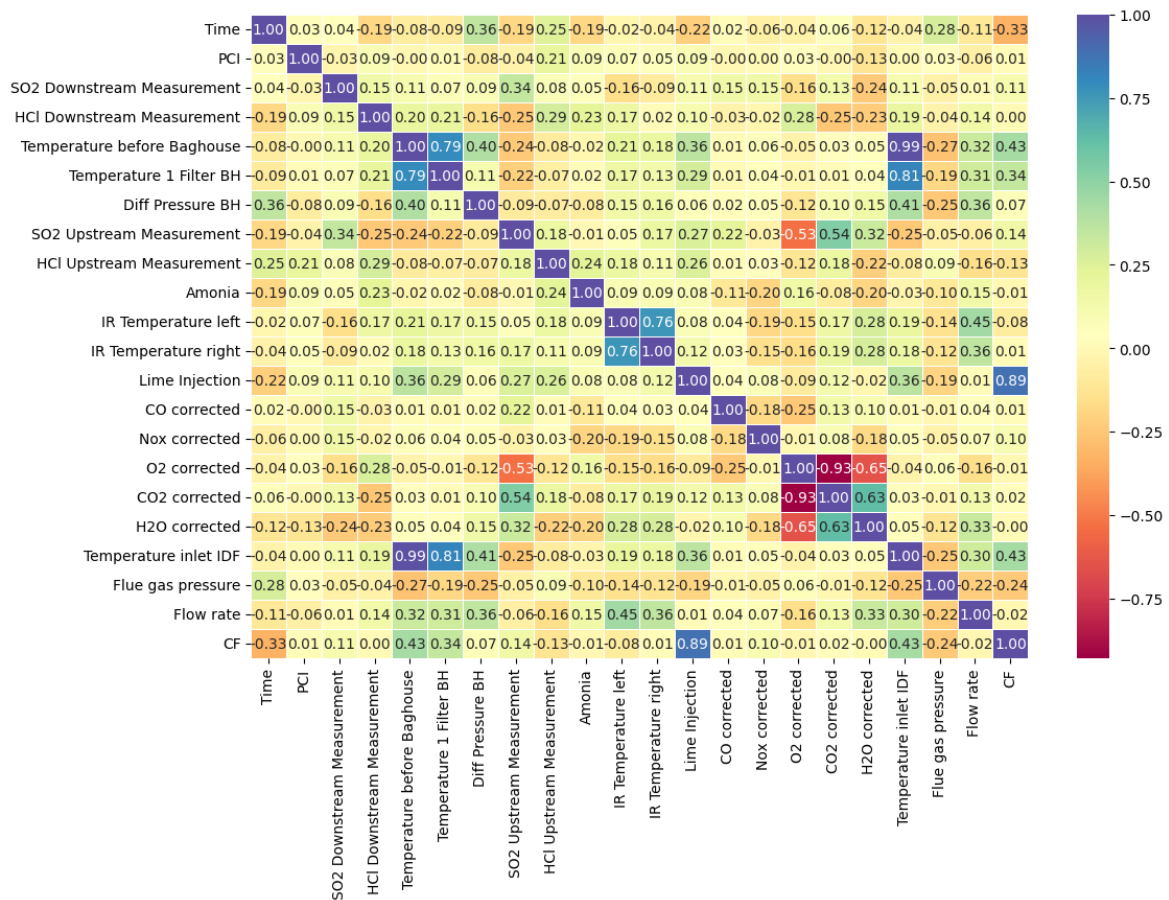


Figura 8. Mapa de calor de correlaciones entre variables candidatas a predecir el CF

Tras un análisis de correlación basado en al mapa de calor anterior, se determinaron las variables que mejor explicaban el comportamiento del CF y se excluyeron aquellas que no aportaban información relevante.

```
excluded_columns = [
    "IR Temperature right", "CF", "SO2 Downstream Measurement", "CO2 corrected",
    "HCl Downstream Measurement", "Lime injection", "Time",
    "Temperature inlet IDF", "Temperature 1 Filter BH"
]
```

Variables eliminadas del modelo predictivo

Asimismo, para optimizar el rendimiento del algoritmo predictivo, se truncaron todas las variables a cuatro decimales y se descartaron las filas con lecturas erróneas o fuera de los rangos operativos de la planta.

Una vez limpiados los datos a utilizar se procedió a diseñar el algoritmo del modelo.

5.3.3 ARQUITECTURA DEL MODELO

Inicialmente, se planteó un enfoque directo: para cada fila de datos (es decir, para un instante concreto de operación), predecir el CF real que se estaba aplicando en ese momento, en base a los valores de todas las otras variables. Tras un correcto funcionamiento del predictor, se explorará la predicción a futuro.

El modelo elegido fue **XGBoost Regressor**, una técnica de *gradient boosting* eficiente y bien adaptada a *datasets* estructurados como este. Para validar su comportamiento, se empleó **validación cruzada (k-fold)** sobre distintos tramos temporales del *dataset*. La métrica de evaluación utilizada fue el R^2 sobre las predicciones de CF.

El coeficiente de determinación R^2 cuantifica la proporción de la variabilidad de los datos que un modelo de regresión es capaz de explicar. Su valor va de 0 a 1; un R^2 próximo a 1 indica que las predicciones del modelo se ajustan muy bien a los datos observados, mientras que un R^2 cercano a 0 revela que el modelo apenas capta la tendencia de los datos.

5.3.4 FUNCIONAMIENTO DEL ALGORITMO

En una primera fase se estableció como objetivo la predicción instantánea del *Correction Factor* correspondiente a cada registro. Posteriormente, se redefinió el objetivo para predecir el CF con un horizonte de dos minutos, manteniendo la misma selección de características.

Para ambas fases, los datos históricos se dividieron en orden cronológico, reservando el 80% inicial para el entrenamiento y el 20% final para la prueba. Esta estrategia garantizó que el modelo aprendiera a partir de datos pasados y se evaluara sobre sucesos posteriores, sin contaminación temporal.

Para optimizar el rendimiento del *XGBoost Regressor* fue necesario ajustar parámetros como la profundidad de los árboles, la tasa de aprendizaje y el grado de regularización. En lugar de probar manualmente cada combinación posible, se recurrió a un método automatizado

llamado *RandomizedSearchCV*, que explora de forma aleatoria un amplio intervalo de valores para cada parámetro y evalúa rápidamente su desempeño.

La evaluación interna de cada combinación se hizo mediante *TimeSeriesSplit*, una variante de validación cruzada diseñada para datos temporales. En lugar de mezclar aleatoriamente los casos, *TimeSeriesSplit* divide la serie cronológica en “bloques” consecutivos, de modo que en cada iteración el modelo se entrena con datos antiguos y se prueba con datos inmediatamente posteriores. Este procedimiento respeta el orden temporal real y evita que el modelo “vea” el futuro durante su entrenamiento.

Con *RandomizedSearchCV* y *TimeSeriesSplit* se configuraron cinco particiones temporales (*folds*). Para cada combinación de hiperparámetros, el algoritmo:

1. Toma el primer 80 % de los datos de entrenamiento en un bloque y entrena el modelo.
2. Valida su rendimiento en el siguiente 20 % de ese bloque.
3. Repite este proceso desplazando el bloque de entrenamiento hacia adelante, hasta completar las cinco particiones.
4. Calcula el promedio de la métrica de interés (R^2) a lo largo de las cinco validaciones.

La combinación de parámetros que produjo el menor error medio en estas pruebas se seleccionó como óptima. Este enfoque garantiza que los valores elegidos funcionen de manera consistente en distintos períodos de operación, no solo en un tramo concreto.

Una vez entrenado con los mejores parámetros, el modelo final se probó en el 20% de datos reservado al inicio (que nunca se había utilizado ni en el entrenamiento ni en la búsqueda de hiperparámetros). Allí se calculó el coeficiente de determinación R^2 que cuantifica cuán cerca quedaron las predicciones del valor real del *Correction Factor*.

El código completo se encuentra en el ANEXO III.

5.3.5 RESULTADOS DEL MODELO PREDICTIVO

La primera aproximación funcionó bien: el modelo era capaz de reconstruir con bastante precisión el CF a partir de los *inputs* del sistema, demostrando que existía una relación clara entre estado de operación y corrección requerida. El R^2 fue de **0,972**.

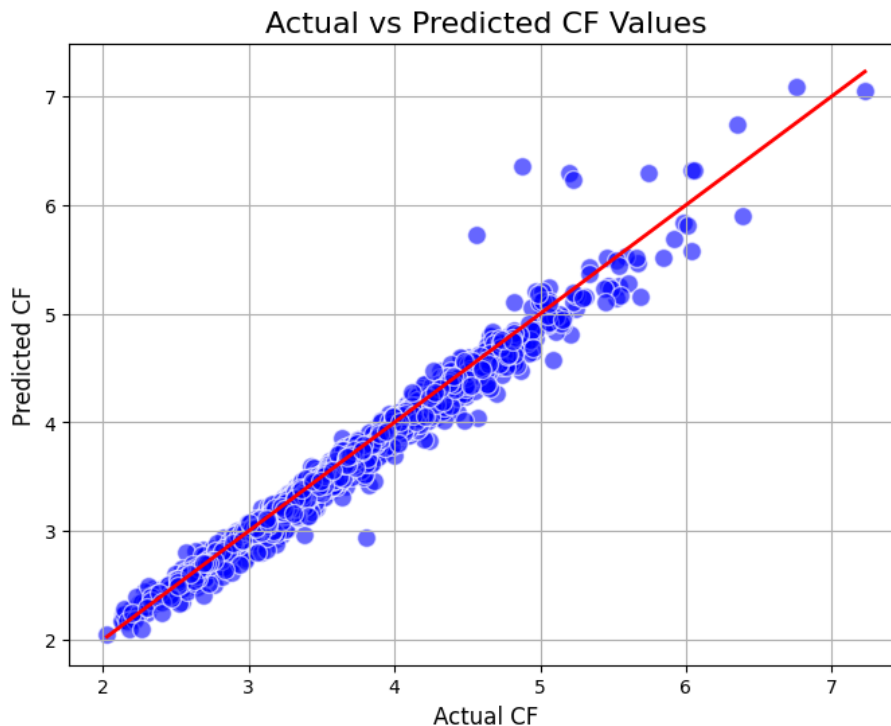


Figura 9. Resultados de la precisión de las predicciones instantáneas frente a los valores reales medidos

El verdadero reto vino al adaptar el modelo para que predijera el CF que sería necesario dentro de 4 minutos, que es el retardo estimado del sistema entre la inyección y la medición efectiva *downstream*. Esta predicción temporal exigía al modelo aprender no sólo las relaciones estáticas entre variables, sino también su evolución a futuro. El primer acercamiento tuvo como objetivo la predicción del CF 2 minutos en el futuro.

En esta configuración, la precisión del modelo se redujo sensiblemente: aunque seguía dando resultados bastante buenos, ya no era lo suficientemente fiable como para predecir con un objetivo temporal aún mayor y sustituir al control tradicional. El R^2 fue de **0,897**.

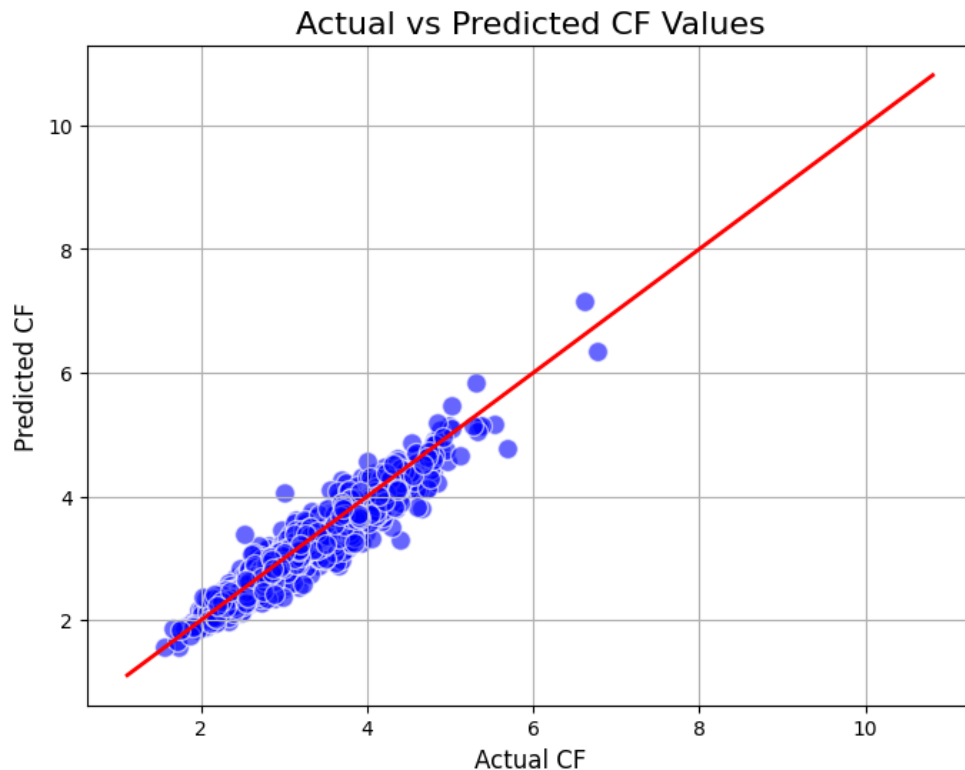


Figura 10. Resultados de la precisión de las predicciones frente a los valores reales medidos con el horizonte temporal de 2 minutos

5.3.6 VALORACIÓN TÉCNICA Y ABANDONO DEL ENFOQUE PREDICTIVO

Pese a que el modelo tenía potencial, se identificaron varias barreras clave para su implementación real en planta:

- **Falta de *feedback* en tiempo real:** al no poder comparar constantemente las predicciones del modelo con el resultado real a futuro, no era posible ajustar ni corregir automáticamente su comportamiento.
- **Complejidad técnica y operativa:** se requería una infraestructura adicional que incluyese un servidor de inferencia, conexión en tiempo real con el DCS a través de protocolos como Modbus, y un sistema de monitorización continua del modelo.

- **Disminución de precisión con el horizonte temporal:** al alejar la predicción varios minutos, el modelo perdía fiabilidad, especialmente en condiciones cambiantes de operación.
- **Falta de predicción de los *upstream*.** Sin ellos, la cal inyectada estaba determinada por un CF, que sí trataba de estar alineado con los retardos, y por una componente estequiométrica que aún se calculaba en función de las lecturas instantáneas de los valores *upstream*, en vez de una predicción de estos.

Por tanto, se decidió abandonar esta línea de trabajo. La conclusión fue que, si bien técnicamente viable, el modelo predictivo no ofrecía una ventaja clara respecto a un controlador PID bien planteado, sobre todo si este actuaba no sobre los valores instantáneos, sino directamente sobre las medias a reportar. Esta constatación condujo a retomar la lógica PID.

5.4 IMPLEMENTACIÓN DEL PID EN VALMET Y PRUEBAS EN LAZO ABIERTO

Tras validar exhaustivamente la lógica del controlador PI en Python, se trasladó su diseño al sistema de control distribuido (DCS) de Valmet, donde se integró de forma modular y segura con la infraestructura existente. A continuación, se describen en detalle tanto la puesta en marcha como la arquitectura funcional y los resultados de las primeras pruebas en lazo abierto.

5.4.1 DESCRIPCIÓN GENERAL DE LA IMPLEMENTACIÓN

Migración de la lógica

La implementación se realizó mediante bloques funcionales (*Function Blocks*) en el entorno FBCad de Valmet DNA. Cada bloque encapsula una parte concreta de la lógica original en Python, garantizando trazabilidad y facilidad de mantenimiento.

Referencias de diseño

El ANEXO IV – Diagrama de bloques lógico muestra el diagrama de bloques completo en dos páginas. En la primera página se agrupan las entradas, parámetros y límites; en la segunda, la lógica auxiliar de habilitación del PID. En el siguiente punto se explica con mayor detalle el funcionamiento de la arquitectura.

Cálculo de dosis y ejecución del controlador

- Dosis estequiométrica: calculada cada segundo a partir de las lecturas *upstream* (HCl, SO₂ y *flue gas flow*).
- Ejecución del PI: desencadenada únicamente al detectar nuevas lecturas *downstream* (aprox. cada 30 s), para optimizar recursos y evitar cálculos innecesarios.
- Ganancias iniciales: $K_P = 0,01$; $K_I = 0,04$; $K_D = 0$ (implementación funcional como PI).

Entorno de pruebas en lazo abierto

Se habilitó un modo de simulación dentro del DCS para comparar la consigna propuesta por el nuevo algoritmo con la consigna real del sistema previo, sin accionar físicamente la válvula de cal.

5.4.2 ARQUITECTURA FUNCIONAL Y DESCRIPCIÓN DE MÓDULOS

La lógica de control se organizó en módulos independientes, comunicados mediante señales internas y “tags” del DCS siguiendo el diagrama de bloques mencionado:

5.4.2.1 Módulo de entradas y parámetros

- Entradas fijas: masas molares, *setpoints* (HCl, SO₂), límites de dosificación (130–1.380 kg/h), factores de confianza y ganancias PID.
- Entradas variables: lecturas *upstream* (conversión a mg/Nm³), lecturas *downstream* y flujo de gases.
- Control manual/automático vía un botón de modo.

5.4.2.2 Módulo de cálculo de error y acción proporcional

- Aplica factores de confianza a las emisiones *downstream*.
- Calcula el error normalizado $\frac{PV-SP}{SP}$ para HCl y SO₂.
- Selecciona el error máximo entre ambos contaminantes y genera un “tag error”.
- Multiplica este error por K_P para obtener la acción proporcional.

5.4.2.3 Módulo de acción integral

- Mantiene un acumulador del “tag error” sólo si la variable auxiliar (módulo auxiliar) habilita el PID.
- Multiplica el acumulado por K_I para producir la acción integral.

5.4.2.4 Módulo de acción derivativa

- Calcula la diferencia entre el error actual y el error previo.
- Multiplica esta diferencia por K_D para obtener la acción derivativa.
- Aunque no se utilice actualmente, se ha dejado desarrollada la lógica de su funcionamiento por si en el futuro se quisiera añadir esta acción.

5.4.2.5 Módulo de cálculo de dosis de cal

- Suma P + I + D para generar el *Correction Factor*.
- Aplica CF sobre la dosis estequiométrica para emitir la consigna de cal final.

5.4.2.6 Módulo auxiliar de habilitación del PID

- Impide que el PID se ejecute si la dosis está en su límite inferior o superior.
- Solo permite actualizar P, I y D cuando cambian las lecturas *downstream* activando la variable auxiliar.
- Garantiza estabilidad evitando recálculos redundantes y respetando las capacidades físicas del actuador.

5.4.3 RESULTADOS DE LAS PRUEBAS EN LAZO ABIERTO

Los resultados mostraron que la señal de cal calculada por el nuevo sistema seguía de forma razonablemente cercana a la del controlador anterior, aunque con un leve retardo atribuible a la estrategia implementada, en la cual la actualización del PID se condiciona a la aparición de cambios en las emisiones *downstream*. Esta similitud en el comportamiento general indicaba que el nuevo controlador era capaz de interpretar de forma efectiva las condiciones del proceso y responder con un nivel de corrección de cal comparable al ya validado previamente en operación real. Además, se verificó que el sistema reaccionaba adecuadamente ante los incrementos en las emisiones de HCl o SO₂, aumentando el CF en proporción a la magnitud del error y sin generar una sobredosis de reactivo. De igual manera, cuando las emisiones descendían, el CF se reducía progresivamente, demostrando una respuesta simétrica y controlada.

Por otro lado, al tratarse de un controlador que opera corrigiendo una base estequiométrica dependiente de variables *upstream*, también se comprobó que la consigna final de cal se mantenía dentro del rango operativo habitual del sistema anterior. Esto aseguraba que el nuevo enfoque no proponía consignas incompatibles con la capacidad operativa del sistema ni comprometía la contención de emisiones.

Durante esta fase de testeo interno, se verificaron:

1. Estabilidad del cálculo frente a variaciones bruscas en emisiones y caudal.
2. Coherencia funcional entre la simulación en Python y la implementación en Valmet.

5.4.4 CONCLUSIONES PRELIMINARES

Los ensayos realizados durante la fase de pruebas en lazo abierto permitieron validar la solidez del nuevo sistema de control desde el punto de vista funcional. Se constató que la lógica implementada era capaz de generar señales de cal consistentes con las del sistema anterior, respetando los rangos operativos habituales y respondiendo adecuadamente a las variaciones reales del proceso. Esta similitud de comportamiento, combinada con una estructura de cálculo más transparente y modular, avaló la validez del diseño propuesto.

A partir de estos resultados, se consideró técnicamente justificado avanzar hacia la siguiente fase del proyecto: la implementación en lazo cerrado. El nuevo controlador demostró tener la capacidad de interpretar correctamente los cambios en las condiciones del proceso y ajustar la dosificación de cal de forma proporcional y controlada, lo que ofrecía garantías suficientes para iniciar el ajuste dinámico de ganancias en tiempo real. Este paso suponía el comienzo de una etapa clave del proyecto, centrada en la optimización fina del sistema y su integración completa en la operación diaria de la planta.

5.5 DISEÑO DE LA INTERFAZ DE CONTROL

Una vez validada la lógica del controlador en lazo abierto, se procedió al diseño de una interfaz gráfica que facilitara tanto la operación como el seguimiento del sistema dentro del entorno Valmet. Esta interfaz tenía que cumplir dos funciones principales: visualizar el comportamiento del controlador de forma clara e intuitiva y permitir la supervisión en tiempo real de las variables clave del proceso.

Para ello, se creó una nueva pantalla en el DCS con los siguientes elementos:

- Indicadores de las emisiones de HCl y SO₂ (*upstream* y *downstream*).
- Estado de las medias diarias y de los últimos 30 minutos para ambos compuestos.
- Representación de los valores de consigna (*setpoints*) y el error máximo.
- Visualización del *Correction Factor* y de la dosis real de cal inyectada.
- Estado del controlador (activo/inactivo) y variables auxiliares, como el PID output o las constantes K_P y K_I.
- Esquema del cálculo del PID.

El diseño se cuidó especialmente para asegurar accesibilidad y rapidez de interpretación, priorizando la información relevante para los operadores y reduciendo la necesidad de intervención manual. Además, se incluyó una funcionalidad para forzar valores o pasar a modo manual, en caso de requerirse ajustes operativos puntuales.

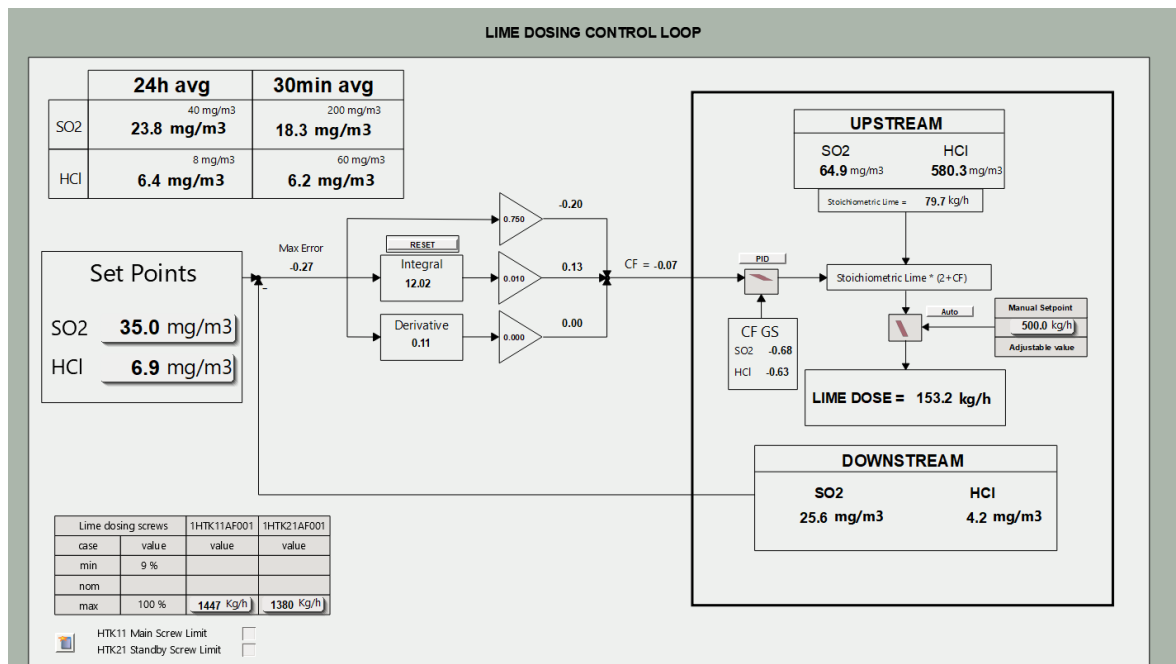


Figura 11. Interfaz del sistema de control (DCS de la planta)

5.6 PUESTA EN MARCHA EN LAZO CERRADO

5.6.1 CONTROL DE LOS DOWNSTREAM INSTANTÁNEOS

Durante la primera fase de puesta en marcha en lazo cerrado, el principal objetivo fue asegurar la precisión del nuevo controlador, posponiendo la consideración del factor de confianza aplicado por la agencia reguladora. Esta decisión respondía a la necesidad de validar primero la capacidad del sistema para mantener las emisiones dentro de los márgenes exigidos, sin comprometer la estabilidad del proceso ni provocar sobreinyecciones descontroladas.

Los parámetros iniciales fueron los mismos que en las pruebas en lazo abierto, que corresponden a unas ganancias de:

- $K_P = 0.01$
- $K_I = 0.04$

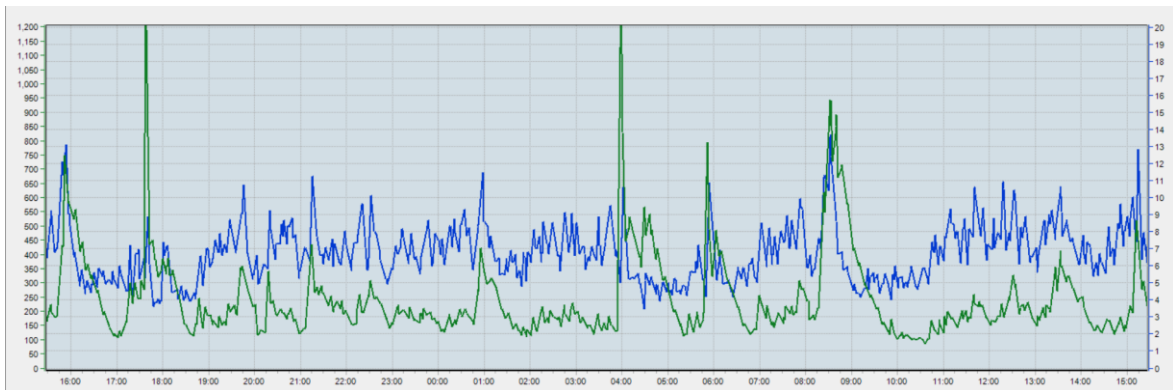


Figura 12. Evolución de la dosis de cal frente a los downstreams de HCl las primeras 24 horas de funcionamiento en bucle cerrado. (DCS de la planta)

En este gráfico, la **línea verde** representa la **dosis de cal**, con el eje izquierdo como ordenada, y la **línea azul** las emisiones **HCl downstream** sin aplicar el factor de confianza, con el eje derecho como ordenada.

En estas primeras pruebas se observó un comportamiento característico del nuevo algoritmo: al estar basado en el error máximo entre las emisiones de HCl y SO₂, el controlador tendía a reaccionar con mayor facilidad ante aumentos de emisiones que a reducir la dosis cuando estas bajaban. Esta asimetría provocaba que la señal de mando aumentara con más frecuencia que descendía, generando una tendencia acumulativa en la dosificación.

Tras analizar esta dinámica, se decidió ajustar drásticamente los parámetros del controlador para mitigar este efecto. En concreto, se definieron nuevas ganancias:

- $K_P = 1.0$
- $K_I = 0.01$

Esta configuración respondía a una lógica clara: dar un peso mucho mayor a la parte proporcional, para conservar la reactividad en casos de superación del *setpoint*, y al mismo tiempo ralentizar el crecimiento de la integral, permitiendo que su efecto fuera más estable y menos susceptible a oscilaciones de corto plazo. En otras palabras, se aumentó la inercia del control, confiando en la acción proporcional como herramienta de respuesta rápida, pero dejando que la integral actúe como corrector lento y sostenido.

Esta estrategia también se apoyó en una observación clave extraída del análisis de datos: **el HCl muestra un comportamiento mucho más estable y predecible que el SO₂**, cuyas lecturas presentan una gran variabilidad natural alrededor del *setpoint*. Este hecho justificaba el refuerzo del término proporcional, dado que el objetivo era reaccionar ante picos reales sin sobrerresponder ante ruido o fluctuaciones menores, especialmente en el canal de SO₂.

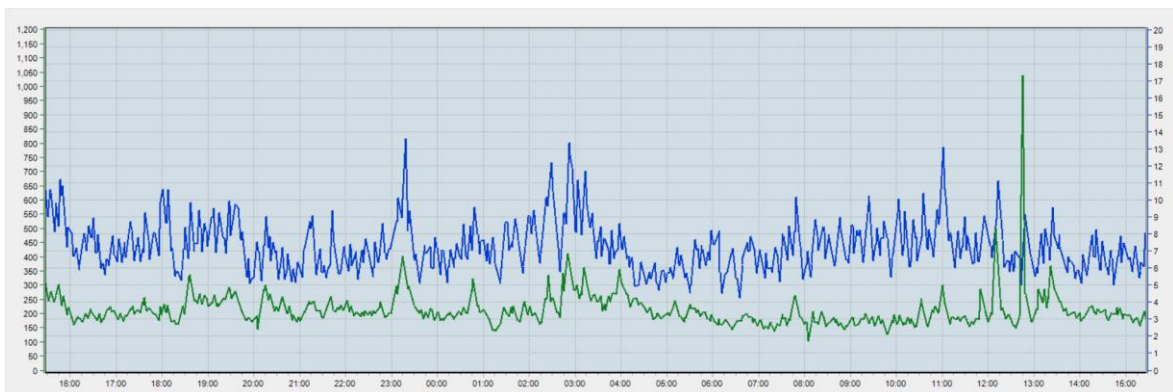


Figura 13. Evolución de la dosis de cal frente a los *downstreams* de HCl tras modificar las ganancias. (DSC de la planta)

Este nuevo comportamiento mostró una mayor estabilidad tanto en la dosificación de cal como en la variación de las emisiones *downstream*, y equilibró la agresividad del control, reaccionando de manera uniforme tanto ante incrementos como disminuciones de la dosis.

Una vez estabilizado el sistema con este conjunto de ganancias, **se introdujo el factor de confianza**, permitiendo que el controlador ajustara la dosis según el nivel de exigencia real impuesta por la agencia medioambiental. Esto debía garantizar un menor consumo de cal. Este cambio se realizó el día 11 de junio.

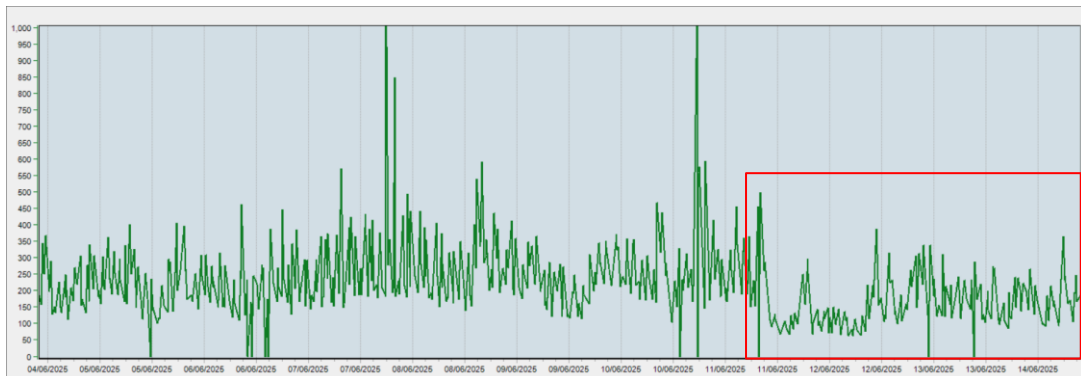


Figura 14. Evolución de la dosis de cal antes y después de la introducción del factor de confianza. (DCS de la planta)

En la figura anterior se puede apreciar como claramente, a partir del día 11 de junio, la dosis de cal oscila en torno a un valor medio menor a los días previos. El ahorro de cal empieza a ser tangible.

A partir de ese momento, se inició un proceso sistemático de *fine tuning* (ajuste fino), en el que se probaron varios conjuntos alternativos de parámetros K_P y K_I , evaluando su impacto sobre la precisión del control, la estabilidad de la señal de cal y el cumplimiento continuo de los límites legales.

Durante esta etapa se registraron diferentes medidas resultado del control que sirven para comparar el desempeño de las distintas configuraciones probadas.

Controller comparison					
Controller	Date	Kp	Ki	Lime dose	Daily HCl AVG
1	21-Jun	0.80	0.0120	249.35	6.12
	22-Jun	0.80	0.0120	262.44	5.95
2	24-Jun	1.00	0.0105	234.00	6.66
	25-Jun	1.00	0.0105	227.07	6.47
3	27-Jun	0.75	0.0105	204.42	6.39
	28-Jun	0.75	0.0105	192.50	6.16
	29-Jun	0.75	0.0105	215.64	6.45

Tabla 1. Comparativa de los resultados de las distintas configuraciones

A partir de los resultados anteriores, se determinó que el **Controlador 2** ofrece el mejor desempeño, ya que logró mantener una buena precisión en el *downstream* obtenido, apuntando consistentemente a un *setpoint* de 6,9 mg/Nm³ para el HCl, establecido por motivos de seguridad, que ha sido una media de **6,57**.

Durante las sucesivas iteraciones de ajuste fino en lazo cerrado, se corroboró que el uso del error instantáneo —tomando el máximo entre HCl y SO₂ en cada muestreo— inducía un sesgo no deseado: cada vez que el SO₂ experimentaba un pico momentáneo (cada 5–10 min), éste dominaba el cálculo del error y disparaba la corrección de cal, provocando un sobredimensionamiento temporal que, a pesar de su buena capacidad de reacción, llevaba la media diaria de HCl por debajo del *setpoint* de seguridad (6,9 mg/Nm³). En los días más críticos, esta dinámica se traducía en medias de HCl de 6,16 o incluso 5,95 mg/Nm³, a expensas de responder a oscilaciones en el SO₂ de muy corta duración.

5.6.2 CONTROL DE LOS DOWNSTREAM MEDIOS EN 30 MINUTOS

Para mitigar la influencia de las oscilaciones breves del SO₂ y evitar que picos momentáneos desvíen la acción de control, se redefinió la señal de error de entrada del PID pasando de valores instantáneos a **medias móviles de 30 minutos**. En este nuevo esquema, cada valor de HCl y SO₂ se pondera a lo largo de un intervalo de media hora, calculando para cada contaminante su media móvil con ventana deslizante. A continuación, el controlador toma el error máximo entre esas dos medias (media – *setpoint*) como señal de entrada.

Este enfoque ofrece dos ventajas clave:

1. Suavizado de picos transitorios: los aumentos bruscos de corta duración en el SO₂ apenas alteran su media de 30 min, de modo que el error relativo del SO₂ solo supera al de HCl cuando existen desviaciones sostenidas.
2. Prioridad al compuesto más restrictivo: dado que el HCl presenta una dinámica más estable y normalmente domina el error de las medias, el controlador vuelve a centrarse en mantener su media diaria cercana al *setpoint* de 6,9 mg/Nm³, salvo cuando el SO₂ experimente una tendencia prolongada al alza.

Al aplicar esta lógica, la corrección de cal se ajusta con mayor coherencia a las condiciones reales de reporte, evitando reacciones excesivas a perturbaciones fugaces y reduciendo la variabilidad en la dosificación. En las pruebas iniciales el conjunto de ganancias utilizadas fue:

- $K_P = 0.70$
- $K_I = 0.007$

Se han reducido las ganancias del controlador PID para adaptarse a la dinámica más lenta y suavizada del proceso al utilizar una media móvil de 30 minutos como variable de control. Mantener las ganancias originales, diseñadas para errores instantáneos más rápidos, generaría respuestas excesivamente agresivas que provocarían oscilaciones e inestabilidad en el sistema. Al disminuir las ganancias, se busca lograr un control más estable y gradual, evitando sobresaltos en la dosificación de cal y permitiendo que el sistema reaccione acorde con la verdadera dinámica del proceso y el retardo inherente a la medición promedio.

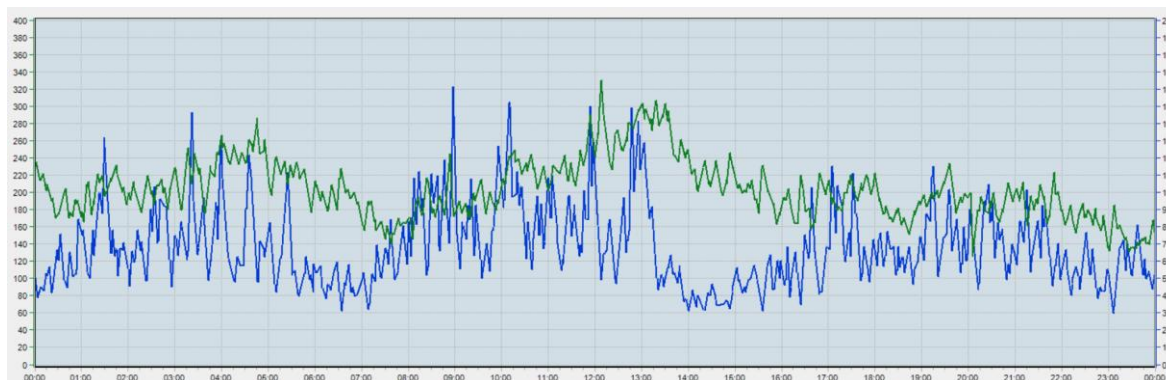


Figura 15. Evolución de la dosis de cal frente a los downstreams de HCl tras implementar el control de 30 minutos. (DCS de la planta)

Como se observa en la figura, el nuevo sistema de control basado en medias móviles de 30 minutos permite una dosificación de cal mucho más estable y gradual. A diferencia del esquema anterior, en el que la dosis respondía de forma directa a las lecturas instantáneas de HCl y SO₂, provocando rampas bruscas ante cualquier pico momentáneo, el nuevo enfoque suaviza significativamente la respuesta del controlador.

Aunque puedan producirse valores puntualmente altos de contaminantes *downstream*, su impacto sobre la media móvil es limitado, lo que evita reacciones exageradas en la dosificación. En consecuencia, la cal inyectada mantiene una evolución más progresiva y menos volátil, lo que no solo mejora la eficiencia del control, sino que también reduce el riesgo operativo asociado a inyecciones puntualmente elevadas.

En particular, esta mayor estabilidad contribuye a minimizar la probabilidad de *baghouse blockages*, un problema recurrente cuando se producen acumulaciones de cal excesivas en cortos periodos de tiempo. Al evitar estas dosificaciones agresivas e innecesarias, el sistema mejora tanto la seguridad operativa como la sostenibilidad del proceso a largo plazo.

Capítulo 6. ANÁLISIS DE RESULTADOS

Una vez implementada la nueva lógica y finalizado el proceso de ajuste fino y estabilizado el sistema con el conjunto de ganancias más eficiente, se evaluó el desempeño del nuevo controlador PI en operación real, comparándolo con el sistema anterior en términos de precisión, eficiencia en el uso de cal e impacto económico.

6.1 RESULTADOS OPERATIVOS

Para evaluar el impacto del nuevo esquema de control basado en medias móviles de 30 minutos, se ha realizado una comparación directa entre días operativos equivalentes —es decir, con requerimientos similares de cal estequiométrica— bajo el control anterior y el nuevo control PID. En la tabla siguiente se presentan, para varios días seleccionados, las medias diarias de consumo total de cal, las concentraciones diarias medias de HCl aguas abajo (la métrica regulatoria principal) y la base estequiométrica calculada para ese mismo día. También se incluye la desviación típica de las concentraciones de HCl *downstream*, así como el coeficiente de variación, para cuantificar la estabilidad del control.

Controller Performance								
Date	Kp	Ki	Stoich. Lime needed	Lime dose	MAX Lime	Daily HCl AVG	MAX HCl	Precision
12-Jul	0.70	0.0070	95.03 kg/h	202.22 kg/h	343.87 kg/h	6.86 mg/m3	24.49 mg/m3	99.45%
13-Jul	0.70	0.0070	95.99 kg/h	204.22 kg/h	326.59 kg/h	6.76 mg/m3	16.07 mg/m3	97.95%
14-Jul	0.70	0.0070	87.13 kg/h	194.93 kg/h	296.25 kg/h	7.19 mg/m3	21.89 mg/m3	95.77%
15-Jul	0.70	0.0070	89.73 kg/h	194.09 kg/h	347.92 kg/h	6.62 mg/m3	22.64 mg/m3	95.91%
16-Jul	0.70	0.0070	88.67 kg/h	192.52 kg/h	428.44 kg/h	6.89 mg/m3	36.16 mg/m3	99.89%
17-Jul	0.70	0.0070	87.42 kg/h	179.71 kg/h	279.17 kg/h	6.80 mg/m3	17.03 mg/m3	98.62%

Avg	-	194.62 kg/h	337.04 kg/h	6.85 mg/m3	23.05 mg/m3	97.93%
CV	-	4.07%	14.15%	2.55%	28.59%	1.63%

Tabla 2. Resultados del control PI definitivo implementado

Old controller (days with similar stoichiometric needs)					
Stoich. Lime needed	Lime dose	MAX Lime	Daily HCl AVG	MAX HCl	Precision
90-100 kg/h	273.04 kg/h	300.00 kg/h	5.78 mg/m3	11.00 mg/m3	72%
90-100 kg/h	273.04 kg/h	300.00 kg/h	5.78 mg/m3	11.00 mg/m3	72%
80-90 kg/h	272.30 kg/h	368.73 kg/h	5.43 mg/m3	12.19 mg/m3	68%
80-90 kg/h	272.30 kg/h	368.73 kg/h	5.43 mg/m3	12.19 mg/m3	68%
80-90 kg/h	272.30 kg/h	368.73 kg/h	5.43 mg/m3	12.19 mg/m3	68%
80-90 kg/h	272.30 kg/h	368.73 kg/h	5.43 mg/m3	12.19 mg/m3	68%
Avg	272.55 kg/h	345.82 kg/h	5.55 mg/m3	11.79 mg/m3	69%
CV	0.13%	9.37%	2.93%	4.76%	2.93%

Tabla 3. Resultados del control previo en condiciones similares

Summary			
	New Controller	Old Controller	Difference in %
Lime usage	194.62 kg/h	272.55 kg/h	-29%
Downstream Precision	97.93%	69%	+41%
Daily HCl Coefficient of Variation (sample vs all 2024)	2.55%	26.77%	-90%

Tabla 4. Comparativa entre los controles nuevo y antiguo

Los resultados muestran una diferencia significativa en la precisión de ambos sistemas. Bajo el nuevo control, el coeficiente de variación del HCl *downstream* es muy bajo (2,55 %), en contraste con los valores considerablemente más altos observados con el control anterior (26,77 % en todo el año pasado). Esta baja variabilidad confirma que el nuevo sistema consigue mantener el contaminante bajo control de forma mucho más estable, lo que a su vez permite operar más cerca del límite medioambiental sin riesgo de incumplimiento. Además, la precisión del nuevo control resultante es muy elevada (97,93%) respecto a la precisión del control anterior (69%), lo que destaca la capacidad del nuevo control de obtener los resultados a los que apunta.

Además, al comparar días con la misma base estequiométrica requerida, se observa una diferencia notable en la cantidad de cal realmente inyectada. Mientras que el sistema anterior tiende a sobredosificar para cubrir picos transitorios, el nuevo PID ajustado a medias móviles consigue cumplir el objetivo regulatorio con un menor exceso. Esta diferencia media se ha

cuantificado en torno a una **reducción del 29 %**, lo que confirma que el nuevo enfoque permite un control más ajustado y eficiente, adaptado a las dinámicas reales del proceso.

6.2 RESULTADOS ECONÓMICOS

A partir de la diferencia relativa de cal dosificada entre el antiguo control y el nuevo (29 %), se ha estimado el ahorro económico anual. Tomando como referencia el consumo histórico de aproximadamente 2.000 toneladas de cal al año, el nuevo sistema permitiría una reducción proyectada equivalente a unas 2.000 toneladas anuales.

Dado un coste aproximado de 200 €/tonelada de cal, este ahorro material se traduce en una reducción de costes próxima a **115.000 € anuales**. Esta estimación, basada en datos reales de operación bajo condiciones comparables, demuestra que la implementación del nuevo controlador PID no solo mejora la precisión del proceso, sino que además tiene un impacto económico tangible y significativo, con retorno inmediato y sostenido en el tiempo.

	SAVINGS		
	Lime usage diff	Porcentaje of Lime savings	Annual estimated savings
Day 1	70.83 kg/h	26%	£ 103,760.70
Day 2	68.82 kg/h	25%	£ 100,816.40
Day 3	77.37 kg/h	28%	£ 113,652.94
Day 4	78.21 kg/h	29%	£ 114,886.28
Day 5	79.78 kg/h	29%	£ 117,190.41
Day 6	92.60 kg/h	34%	£ 136,019.21
Average Annual estimated savings	77.93 kg/h	29%	£ 114,387.66

Tabla 5. Impacto económico de la implementación del nuevo control

Además del ahorro directo asociado a la reducción del consumo de cal, cabe destacar un beneficio económico adicional derivado de este menor uso. Al inyectarse menos cantidad de cal, el volumen de residuos generados en el sistema de filtrado (APCR, *Air Pollution Control Residues*) también se reduce. Esto implica una disminución en la cantidad de toneladas de residuo que deben ser gestionadas y eliminadas, lo que se traduce en un menor coste de

tratamiento y disposición final. Aunque este ahorro asociado a la gestión de residuos no ha podido ser cuantificado con precisión en esta fase del proyecto, representa una fuente adicional de eficiencia económica que refuerza la conveniencia del nuevo sistema de control.

Los ahorros totales potenciales para el grupo FCC Medioambiente serían aún mayores si se considera que la compañía opera 13 plantas de valorización energética, en todas las cuales podría implementarse esta tecnología.

Capítulo 7. CONCLUSIONES Y TRABAJOS FUTUROS

Este trabajo ha tenido como objetivo principal el rediseño y mejora del sistema de control de inyección de cal en un proceso industrial de reducción de emisiones ácidas, centrándose en optimizar tanto la precisión del cumplimiento ambiental como la eficiencia en el consumo de reactivo. A lo largo del desarrollo se ha demostrado que, con un enfoque estructurado basado en simulación, análisis de datos reales y validación progresiva en planta, es posible mejorar sustancialmente el comportamiento de un sistema ya operativo.

La implementación de un controlador PI adaptado al comportamiento de los compuestos regulados ha permitido **reducir el consumo de cal en torno a un 29 %**, manteniendo a su vez las emisiones de **HCl y SO₂ por debajo de los límites legales**. Esta mejora no solo implica un beneficio medioambiental, al minimizar el exceso de reactivo, sino también **un ahorro económico directo estimado en más de 100.000 € anuales**. Estos resultados validan la elección de un controlador sencillo, pero bien ajustado, con capacidad de adaptación a las características de un proceso real y con márgenes de inestabilidad no despreciables.

Asimismo, se ha desarrollado un modelo predictivo supervisado con técnicas de *machine learning*, entrenado sobre históricos reales del sistema. Aunque este modelo no fue implementado en planta, su desarrollo ha permitido explorar una línea de trabajo con un alto potencial futuro, especialmente en procesos continuos donde los retardos, las perturbaciones externas o la alta variabilidad suponen retos difíciles de afrontar mediante controladores tradicionales.

De cara a trabajos futuros, se plantea como posibilidad la integración progresiva de modelos predictivos como el desarrollado en este trabajo dentro de la lógica de control. Este modelo, basado en aprendizaje automático, ha demostrado su capacidad para anticipar el comportamiento del sistema a corto plazo, concretamente estimando el factor de corrección dos minutos hacia adelante. A partir de esta base, se abre la opción de emplearlo como

sistema de respaldo, herramienta de ajuste dinámico o incluso como componente central en esquemas de control más avanzados, como un Control Predictivo por Modelo (MPC). La incorporación de estas soluciones permitiría anticipar desviaciones, probar respuestas simuladas ante distintos escenarios y mejorar aún más la eficiencia operativa, especialmente en contextos donde la disponibilidad de datos en tiempo real es elevada.

En conclusión, este trabajo demuestra que el rediseño de sistemas de control industrial, incluso a partir de controladores clásicos como el PID, puede beneficiarse significativamente del uso de técnicas modernas como el aprendizaje automático. La implementación de un modelo predictivo funcional y su validación sobre datos reales sienta las bases para una futura evolución hacia sistemas de control más inteligentes, capaces de combinar robustez, flexibilidad y capacidad de anticipación.

Capítulo 8. BIBLIOGRAFÍA

- [1] Valmet Corporation. "Valmet DNA Distributed Control System – System Overview and Architecture". Valmet Technical Documentation, 2020. <https://www.valmet.com/dna-system-overview>
- [2] National Instruments. *The PID Controller & Theory Explained*. National Instruments, 2025. <https://www.ni.com/en/shop/labview/pid-theory-explained.html>
- [3] Analytics Vidhya. *XGBoost Tutorial: A Complete Guide*. Analytics Vidhya, 2024. <https://www.analyticsvidhya.com/blog/2024/01/xgboost-tutorial-a-complete-guide/>
- [4] A. Malanis, "EDA and Modeling with RandomForestRegressor on Real World Vehicle Dataset Prediction," *Medium*, 2023. <https://medium.com/@ashmalanis08/eda-and-modeling-with-randomforestregressor-on-real-world-vehicle-dataset-prediction-bbaf1e8e10c2>
- [5] M. Sadeghi, "How to Split Your Data for Machine Learning," *Medium*, 2023. <https://medium.com/@masadeghi6/how-to-split-your-data-for-machine-learning-eae893a8799c>
- [6] N. Öztürk, "Nested Cross Validation Against Overfitting," *Medium*, 2022. <https://medium.com/@nlztrk/nested-cross-validation-against-overfitting-b2e33fc47060>
- [7] J. Brownlee, "Nested Cross-Validation for Machine Learning with Python," *Machine Learning Mastery*, 2020. <https://machinelearningmastery.com/nested-cross-validation-for-machine-learning-with-python/>
- [8] Microsoft. "Using Python in Visual Studio Code." Visual Studio Code Documentation, 2024. <https://code.visualstudio.com/docs/python/python-tutorial>
- [9] Environment Agency. "Emissions Monitoring and Reporting Requirements for Industrial Plants". UK Environment Agency, 2020. <https://www.gov.uk/guidance/environmental-monitoring-emissions-reporting>
- [10] Mendes, J., Silva, C., & Carvalho, L. (2021). *Fuzzy logic control applied to dry sorbent injection systems in waste incineration plants*. Journal of Environmental Chemical Engineering, 9(5), 105894. <https://doi.org/10.1016/j.jece.2021.105894>

- [11] Chen, X., Liu, H., & Wang, Y. (2019). *Model predictive control for flue gas desulfurization under variable waste sludge incineration loads*. Chemical Engineering Journal, 375, 121997. <https://doi.org/10.1016/j.cej.2019.121997>

ANEXO I - HAZOP

Proposed Change	Documents being Reviewed			
	Doc Name	Description	Area	Notes
		-	-	

Guide Words	Deviation	Possible Cause	Consequences	Protection Means	Action Required	RAMS
Emissions	Overemission	Lime underdosing	Breach	The DCS interface includes a selector switch that allows the operator to alternate between automatic and manual control modes	Hourly inspections will be performed by operators to ensure pollutant concentrations stay within permitted limits	-
				HCl Alarm	In response to the HCl Level 1HTA4QC001Z04 alarm, operators will switch to manual control to restore emissions to acceptable levels	-
				SO2 Alarm	In response to the SO2 Level 1HTA4QC001Z04 alarm, operators will switch to manual control to restore emissions to acceptable levels	-
Lime	Overdosing	Excessive system inertia and limited responsiveness when reducing lime dosage after concentration spikes	Baghouse Block, Increase in material costs	If pressure readings in the baghouse exceed the high-high (HH) threshold, the Differential Pressure Alarm (1HTEDCP001X002) will be triggered	Operators will be required to open an access door in the baghouse and manually dislodge the accumulated lime deposits	LN-02-01-OP-OPS8H-004-01-MS LN-02-01-OP-OPS8H-001-01-MS
Maintenance	Incorrect downstream response profiles following control actions	Controller tuning	Overdosing / Underdosing	The new controller was implemented in the DCS using its native programming language, FB Cad, ensuring readability and maintainability for personnel familiar with the system	Gain adjustments should be made by an authorized individual with DCS expertise, directly within the loop defined in the 1HTK10DF04 file	Lime_Dosing_System_Operations_Manual.pdf
	Integral Windup	Controller in open loop for an extended period of time	Inaccurate lime dose	Integral Reset button.	Press the button to reset the integral back to 0	Lime_Dosing_System_Operations_Manual.pdf
Operations	-	Operators trained to understand the lime dosing control in the DCS	Overdosing / underdosing / breach	-	Operators to read the training document.	Lime_Dosing_System_Operations_Manual.pdf

ANEXO II – CÓDIGO PID

01/07/2025, 08:26

MAX_error_PID.py

~\OneDrive - FCC\Desktop\Stuff\Lime dosing\PID\Juergen\MAX_error_PID.py

```

1 import numpy as np
2 import matplotlib.pyplot as plt
3 import math
4 import pandas as pd
5
6 # Constants
7 MAX_SCREW = 1380 * 1 # [kg/h]
8 MIN_SCREW = 1380 * 0.09 # [kg/h]
9
10 class PIDController:
11     def __init__(self, Kp, Ki, Kd, setpoint_so2, setpoint_hcl):
12         self.Kp = Kp
13         self.Ki = Ki
14         self.Kd = Kd
15         self.setpoint_so2 = setpoint_so2
16         self.setpoint_hcl = setpoint_hcl
17         self.prev_error = 0
18         self.integral = 0
19
20     def calculate(self, value_so2, value_hcl):
21         error = max(((value_so2 - self.setpoint_so2)/self.setpoint_so2), ((value_hcl -
self.setpoint_hcl)/self.setpoint_hcl))
22         P = self.Kp * error
23
24         self.integral += error
25         I = self.Ki * self.integral
26
27         D = self.Kd * (error - self.prev_error)
28
29         self.prev_error = error
30
31         output = P + I + D
32         print(f'PID: {output:.2f}, Error: {error:.2f}, Integral: {self.integral:.2f}')
33         return output
34
35 # Lime dosing calculation with Feedforward and PID
36 def lime_dosing_control(upstream_SO2, upstream_HCl, setpoint_SO2, setpoint_HCl, fg_flow, cf):
37     lime_needed_SO2 = (((upstream_SO2 - setpoint_SO2) * fg_flow/10**6) / 64) * 74
38     lime_needed_HCl = (((upstream_HCl - setpoint_HCl) * fg_flow/10**6) / (36.45 * 2)) * 74
39     stoich_lime = lime_needed_HCl + lime_needed_SO2
40     lime_needed = stoich_lime * (1+cf)
41     print(f'Stoich lime: {stoich_lime:.2f}')
42     print(f'Lime used: {lime_needed:.2f}')
43     return lime_needed
44
45 # Simulation function
46 def simulate_pid_controller(steps, setpoint_SO2, setpoint_HCl, initial_SO2, initial_HCl):
47     # PID controllers
48     Kp, Ki, Kd = 0.01, 0.04, 0.0
49     pid = PIDController(Kp, Ki, Kd, setpoint_SO2, setpoint_HCl)
50     downstream_SO2 = 40
51     downstream_HCl = 8
52
53     # Leer datos desde CSV
54     data = pd.read_csv("Dataset2024_1min.csv")
55     if len(data) < steps:
56         raise ValueError("El archivo tiene menos filas que pasos de simulación.")
57

```

localhost:62694/7fac5ac4-274a-41e9-9ec3-cb1ee4424dc/

1/3

```
01/07/2025, 08:26 MAX_error_PID.py
58 upstream_SO2_series = data["SO2 Upstream Measurement"].values
59 upstream_HCL_series = data["HCL Upstream Measurement"].values
60 fg_flow_series = data["Flow rate"].values
61
62 # Eficiencias
63 efficiency_so2 = 0.05
64 efficiency_hcl = 0.25
65
66 # Plotted variables
67 upstream_so2_values = []
68 upstream_hcl_values = []
69 so2_values = [initial_SO2]
70 hcl_values = [initial_HCL]
71 lime_doses = []
72 cf = 0
73
74 for step in range(steps):
75     print(f'\n\nStep {step}')
76     print('-----')
77
78     upstream_SO2 = upstream_SO2_series[step]
79     upstream_HCL = upstream_HCL_series[step]
80     fg_flow = fg_flow_series[step]
81
82     #upstream_SO2 = 120
83     #upstream_HCL = 400
84     #fg_flow = 120000
85
86     upstream_HCL_kg_h = upstream_HCL * fg_flow / 10**6
87     upstream_SO2_kg_h = upstream_SO2 * fg_flow / 10**6
88
89     print(f'SO2 Upstream: {upstream_SO2:.2f}')
90     print(f'HCL Upstream: {upstream_HCL:.2f}')
91
92     cf = pid.calculate(downstream_SO2, downstream_HCL)
93
94     lime_dosing = lime_dosing_control(upstream_SO2, upstream_HCL, setpoint_SO2, setpoint_HCL,
95                                     fg_flow, cf)
96     lime_doses.append(lime_dosing)
97
98     neutralized_SO2 = max(((lime_dosing * 64) / 74) * efficiency_so2, 0)
99     neutralized_HCL = max(((lime_dosing * (36.45 * 2)) / 74) * efficiency_hcl, 0)
100
101     downstream_SO2_kg_h = max(0, upstream_SO2_kg_h - neutralized_SO2)
102     downstream_HCL_kg_h = max(0, upstream_HCL_kg_h - neutralized_HCL)
103
104     downstream_SO2 = downstream_SO2_kg_h / (fg_flow / 10**6)
105     downstream_HCL = downstream_HCL_kg_h / (fg_flow / 10**6)
106
107     print(f'SO2 Downstream: {downstream_SO2:.2f}')
108     print(f'HCL Downstream: {downstream_HCL:.2f}')
109
110     so2_values.append(downstream_SO2)
111     hcl_values.append(downstream_HCL)
112     upstream_so2_values.append(upstream_SO2)
113     upstream_hcl_values.append(upstream_HCL)
114
115     return so2_values, hcl_values, lime_doses, upstream_so2_values, upstream_hcl_values
116
117 # Example usage
```

localhost:62694/7fac5ac4-274a-41e9-9ec3-cb1eef4424dc/

2/3

01/07/2025, 08:26

MAX_error_PID.py

```
118 setpoint_SO2 = 40 # [mg/Nm3]
119 setpoint_HCl = 8 # [mg/Nm3]
120 initial_SO2 = 40 # [mg/Nm3]
121 initial_HCl = 8 # [mg/Nm3]
122 steps = 10
123
124 so2_values, hcl_values, lime_doses, upstream_so2_values, upstream_hcl_values = simulate_pid_control-
    ler(steps, setpoint_SO2, setpoint_HCl, initial_SO2, initial_HCl)
125
126 plt.figure(figsize=(10, 6))
127
128 plt.subplot(3, 1, 1)
129 plt.plot(upstream_so2_values, label="Upstream SO2", color='r')
130 plt.plot(upstream_hcl_values, label="Upstream HCl", color='b')
131 plt.title("Upstream SO2 & HCl Concentration")
132 plt.xlabel("Time Step")
133 plt.ylabel("Concentration [mg/Nm3]")
134 plt.legend()
135 plt.grid(True)
136
137 plt.subplot(3, 1, 3)
138 plt.plot(so2_values, label="SO2 Concentration", color='r')
139 plt.plot(hcl_values, label="HCl Concentration", color='b')
140 plt.axhline(y=setpoint_SO2, color='r', linestyle='--', label="Setpoint SO2")
141 plt.axhline(y=setpoint_HCl, color='b', linestyle='--', label="Setpoint HCl")
142 plt.title("Downstream SO2 & HCl Concentration")
143 plt.xlabel("Time Step")
144 plt.ylabel("Concentration [mg/Nm3]")
145 plt.legend()
146 plt.grid(True)
147
148 plt.subplot(3, 1, 2)
149 plt.plot(lime_doses, label="Lime Dosing", color='green')
150 plt.title("Lime Dosing Output")
151 plt.xlabel("Time Step")
152 plt.ylabel("Lime Dosing [kg/h]")
153 plt.legend()
154 plt.grid(True)
155
156 plt.tight_layout()
157 plt.show()
158
```

ANEXO III – CÓDIGOS MODELO PREDICTIVO

PREDICCIÓN INSTANTÁNEA

01/07/2025, 09:04

Notebook_RFR_1year

1. Librerías

```
In [1]: import pandas as pd
import numpy as np
from sklearn.ensemble import RandomForestRegressor
from sklearn.model_selection import KFold, RandomizedSearchCV
from sklearn.metrics import r2_score
from tqdm import tqdm
from scipy.stats import randint, uniform
import matplotlib.pyplot as plt
import seaborn as sns
```

2. Carga y preprocesado

```
In [2]: # Cargar dataset maestro
maestro_data = pd.read_csv("Dataset2024_30min.csv")
maestro_data['Time'] = pd.to_datetime(maestro_data['Time'], format='%d/%m/%Y %H:%M')
#maestro_data = maestro_data.drop(columns=["Time"])

# Filtrar datos según Los rangos
'''
rangos = {
    'PCI': {'min': 7500, 'max': 11300},
    'SO2 Downstream Measurement': {'min': 4, 'max': 140},
    'HCl Downstream Measurement': {'min': 3.4, 'max': 11},
    'Temperature before Baghouse': {'min': 146, 'max': 160},
    'Temperature 1 Filter BH': {'min': 138, 'max': 155},
    'Diff Pressure BH': {'min': 17.2, 'max': 24},
    'SO2 Upstream Measurement': {'min': 40, 'max': 350},
    'HCl Upstream Measurement': {'min': 260, 'max': 1000},
    'Amonia': {'min': 20, 'max': 84},
    'IR Temperature left': {'min': 700, 'max': 920},
    'IR Temperature right': {'min': 720, 'max': 900},
    'Lime Injection': {'min': 100, 'max': 600},
    'CO corrected': {'min': 1.75, 'max': 12},
    'Nox corrected': {'min': 100, 'max': 340},
    'O2 corrected': {'min': 4, 'max': 10},
    'CO2 corrected': {'min': 7.5, 'max': 11},
    'H2O corrected': {'min': 13.3, 'max': 25.2},
    'Temperature inlet IDF': {'min': 143, 'max': 154},
    'Flue gas pressure': {'min': 935, 'max': 1000},
    'Flow rate': {'min': 106000, 'max': 132000},
    'CF': {'min': 1, 'max': 8.5}
}

# Convertir las columnas relevantes a tipo numérico (en caso de que contengan cadenas)
for col in rangos:
    maestro_data[col] = pd.to_numeric(maestro_data[col], errors='coerce')

# Filtrar según Los rangos numéricos
for col in rangos:
    min_val = rangos[col]['min']
    max_val = rangos[col]['max']
    maestro_data = maestro_data[(maestro_data[col] >= min_val) & (maestro_data[col] <= max_val)]

# Eliminar filas con valores NaN después de la conversión (si hubo errores de conversión)
maestro_data = maestro_data.dropna()
```

file:///C:/Users/LARRIBAC7649/OneDrive - FCC/Desktop/Stuff/TFM/Anexos/Notebook_RFR_1year.html

1/6

01/07/2025, 09:04

Notebook_RFR_1year

```
# Redondear todas las variables numéricas a 2 decimales  
maestro_data = maestro_data.round(4)
```

```
In [ ]: plt.figure(figsize=(12, 8))  
sns.heatmap(maestro_data.corr(), annot=True, cmap="Spectral", fmt=".2f", linewidths=0.5)  
plt.show()
```

3. Definir características y objetivo

```
In [3]: # Coger una muestra del dataset original para pruebas  
#maestro_data = maestro_data.iloc[:30000]  
  
# Definir features y target  
target = "CF"  
excluded_columns = ["Time", "CF", "SO2 Downstream Measurement", "CO2 corrected",  
                    "HCl Downstream Measurement", "Lime injection",  
                    "Temperature inlet IDF", "Temperature before Baghouse"]  
features = maestro_data.columns.difference(excluded_columns)  
  
# Seleccionar muestras para entrenamiento  
#train_data = maestro_data.sample(n=8000)  
train_data = maestro_data.iloc[:8000]  
  
# Datos de entrenamiento  
X_train = train_data[features]  
y_train = train_data[target]  
  
# Datos de test (el resto de los datos)  
test_data = maestro_data.drop(train_data.index)  
X_test = test_data[features]  
y_test = test_data[target]  
  
# Almacenar valores actuales de downstream y lime solo para el conjunto de test  
actual_SO2_downstream = test_data["SO2 Downstream Measurement"]  
actual_HCl_downstream = test_data["HCl Downstream Measurement"]  
actual_lime_injection = test_data["Lime Injection"]  
SO2_upstream = test_data["SO2 Upstream Measurement"]  
HCl_upstream = test_data["HCl Upstream Measurement"]
```

4. Validación cruzada externa

```
In [4]: # Configuración de la validación cruzada externa  
cv_outer = KFold(n_splits=10, shuffle=True)  
  
# Parámetros para RandomizedSearchCV  
param_dist = {  
    'n_estimators': [5, 20, 50, 100, 200],  
    'max_depth': [3, 5, 7, 10, 20],  
    'min_samples_split': [2, 5, 7, 10],  
    'min_samples_leaf': [1, 5, 7, 10],  
    'bootstrap': [True, False]  
}
```

5. Validación cruzada interna RandomizedSearchCV

```
In [5]: # Almacenar resultados  
outer_results = []
```

01/07/2025, 09:04

Notebook_RFR_1year

```
best_models = []
train_r2_scores = []
val_r2_scores = []

# Barra de progreso con tqdm
with tqdm(total=cv_outer.get_n_splits(X_train), desc="Nested CV Progress", ncols=100) as pbar:
    for train_idx, test_idx in cv_outer.split(X_train):
        # Dividir Los datos
        X_train_fold, X_test_fold = X_train.iloc[train_idx], X_train.iloc[test_idx]
        y_train_fold, y_test_fold = y_train.iloc[train_idx], y_train.iloc[test_idx]

        # Validación cruzada interna
        cv_inner = KFold(n_splits=3, shuffle=True)

        # Definir el modelo
        model = RandomForestRegressor()

        # Realizar La búsqueda aleatoria en Los hiperparámetros
        search = RandomizedSearchCV(model, param_distributions=param_dist, n_iter=10, scoring='r2')
        result = search.fit(X_train_fold, y_train_fold)

        # Obtener el mejor modelo de La búsqueda aleatoria
        best_model = result.best_estimator_
        best_models.append(best_model)

        # Evaluar en el conjunto de entrenamiento y prueba
        y_train_pred = best_model.predict(X_train_fold)
        y_test_pred = best_model.predict(X_test_fold)

        train_r2 = r2_score(y_train_fold, y_train_pred)
        val_r2 = result.best_score_
        test_r2 = r2_score(y_test_fold, y_test_pred)

        train_r2_scores.append(train_r2)
        val_r2_scores.append(val_r2)
        outer_results.append(test_r2)

    # Reportar progreso
    print(f"> Train R²: {train_r2:.3f}, Val R²: {val_r2:.3f}, Test R²: {test_r2:.3f}, Best params: {search.best_params_}")

    # Actualizar La barra de progreso
    pbar.update(1)

# Resumen de Los resultados
print(f"\nMean Test R²: {np.mean(outer_results):.3f} ± {np.std(outer_results):.3f}")
print(f"Mean Train R²: {np.mean(train_r2_scores):.3f} ± {np.std(train_r2_scores):.3f}")
print(f"Mean Val R²: {np.mean(val_r2_scores):.3f} ± {np.std(val_r2_scores):.3f}")
```

```
Nested CV Progress: 10%|██████████| 1/10 [00:21<03:11, 2.123s/it]
> Train R²: 0.996, Val R²: 0.988, Test R²: 0.992, Best params: {'n_estimators': 50, 'min_samples_split': 7, 'min_samples_leaf': 7, 'max_depth': 20, 'bootstrap': True}

Nested CV Progress: 20%|██████████| 2/10 [00:43<02:55, 2.192s/it]
> Train R²: 0.999, Val R²: 0.992, Test R²: 0.993, Best params: {'n_estimators': 50, 'min_samples_split': 5, 'min_samples_leaf': 1, 'max_depth': 20, 'bootstrap': True}

Nested CV Progress: 30%|██████████| 3/10 [01:27<03:42, 3.184s/it]
> Train R²: 0.998, Val R²: 0.991, Test R²: 0.993, Best params: {'n_estimators': 200, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': 20, 'bootstrap': True}

Nested CV Progress: 40%|██████████| 4/10 [01:45<02:38, 2.638s/it]
> Train R²: 0.998, Val R²: 0.990, Test R²: 0.994, Best params: {'n_estimators': 50, 'min_samples_split': 10, 'min_samples_leaf': 1, 'max_depth': 20, 'bootstrap': True}

Nested CV Progress: 50%|██████████| 5/10 [01:57<01:46, 2.122s/it]
```

file:///C:/Users/LARRIBAC7649/OneDrive - FCC/Desktop/Stuff/TFM/Anexos/Notebook_RFR_1year.html

3/6

01/07/2025, 09:04

Notebook_RFR_1year

```
> Train R²: 0.995, Val R²: 0.987, Test R²: 0.988, Best params: {'n_estimators': 50, 'min_samples_split': 5, 'min_samples_leaf': 7, 'max_depth': 10, 'bootstrap': True}
Nested CV Progress: 60% | ██████████ | 6/10 [02:17<01:23, 20.78s/it]
> Train R²: 0.997, Val R²: 0.990, Test R²: 0.993, Best params: {'n_estimators': 100, 'min_samples_split': 7, 'min_samples_leaf': 1, 'max_depth': 10, 'bootstrap': True}
Nested CV Progress: 70% | ██████████ | 7/10 [02:37<01:01, 20.60s/it]
> Train R²: 0.999, Val R²: 0.992, Test R²: 0.992, Best params: {'n_estimators': 50, 'min_samples_split': 7, 'min_samples_leaf': 1, 'max_depth': 20, 'bootstrap': True}
Nested CV Progress: 80% | ██████████ | 8/10 [03:17<00:53, 26.83s/it]
> Train R²: 0.997, Val R²: 0.990, Test R²: 0.994, Best params: {'n_estimators': 200, 'min_samples_split': 2, 'min_samples_leaf': 5, 'max_depth': 20, 'bootstrap': True}
Nested CV Progress: 90% | ██████████ | 9/10 [03:25<00:20, 20.80s/it]
> Train R²: 0.993, Val R²: 0.982, Test R²: 0.988, Best params: {'n_estimators': 20, 'min_samples_split': 2, 'min_samples_leaf': 10, 'max_depth': 20, 'bootstrap': True}
Nested CV Progress: 100% | ██████████ | 10/10 [03:51<00:00, 23.11s/it]
> Train R²: 0.995, Val R²: 0.988, Test R²: 0.992, Best params: {'n_estimators': 100, 'min_samples_split': 2, 'min_samples_leaf': 7, 'max_depth': 10, 'bootstrap': True}

Mean Test R²: 0.992 ± 0.002
Mean Train R²: 0.997 ± 0.002
Mean Val R²: 0.989 ± 0.003
```

6. Evaluación

```
In [6]: # Obtener el mejor modelo (el del mejor fold externo)
best_overall_model = best_models[np.argmax(outer_results)]

# Hacer predicciones sobre el conjunto de test (resto de los datos no usados para entrenar)
y_test_pred = best_overall_model.predict(X_test)
test_r2 = r2_score(y_test, y_test_pred)

print(f"\n🔴 Unseen Final Test R² : {test_r2:.3f}")

🔴 Unseen Final Test R² : 0.972
```

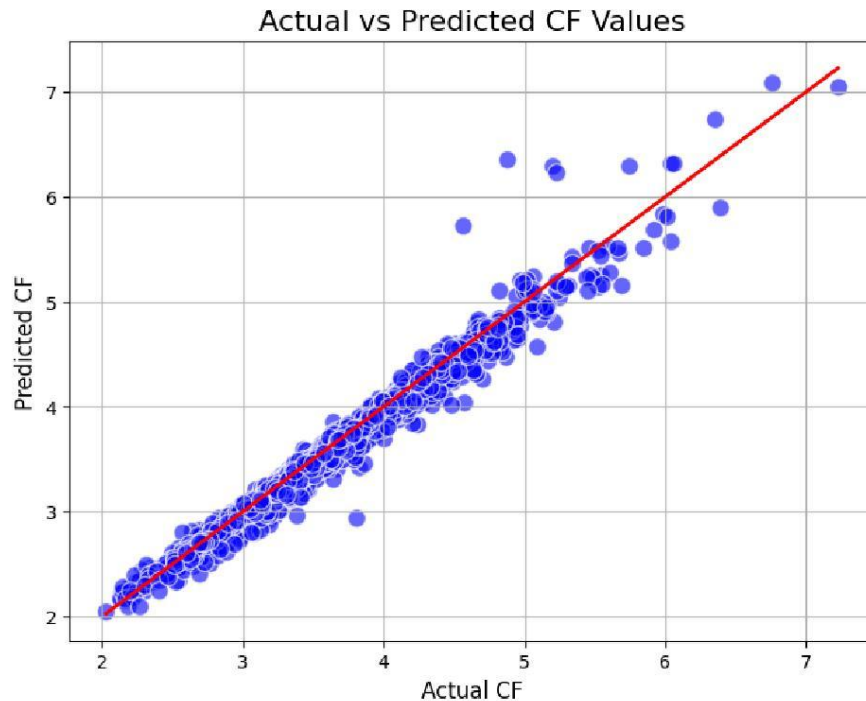
7. CF reales vs. predichos

```
In [7]: # Ploteo de los valores reales vs predichos
plt.figure(figsize=(8, 6))

# Graficar los valores reales vs. los valores predichos
plt.scatter(y_test, y_test_pred, color='blue', alpha=0.6, edgecolors='w', s=100)
plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linewidth=2)
plt.title('Actual vs Predicted CF Values', fontsize=16)
plt.xlabel('Actual CF', fontsize=12)
plt.ylabel('Predicted CF', fontsize=12)
plt.grid(True)
plt.show()
```

01/07/2025, 09:04

Notebook_RFR_1year



8. Cálculo de la inyección de cal

```
In [9]: # 1. Cálculo de la cal necesaria para reducir SO2 y HCl hasta Los setpoints
lime_needed_mg_Nm3 = (((X_test["HCl Upstream Measurement"] - 8) / (36.45 * 2)) +
                      ((X_test["SO2 Upstream Measurement"] - 40) / 64)) * 74

# Evitar valores negativos (si upstream ya es menor que el setpoint, no se necesita cal)
lime_needed_mg_Nm3 = lime_needed_mg_Nm3.clip(lower=0)

# 2. Conversión a kg/h usando el flujo de gases
lime_needed_kg_h = lime_needed_mg_Nm3 * X_test["Flow rate"] * 1e-6 # Conversión a kg/h

# 3. Aplicación del CF predicho para obtener la inyección de cal recomendada
lime_injection_pred = lime_needed_kg_h * y_test_pred

# Imprimir las medias de Lime usada (real vs predicha) y valores Downstream
mean_actual_lime = actual_lime_injection.mean()
mean_predicted_lime = lime_injection_pred.mean()
mean_actual_SO2_downstream = actual_SO2_downstream.mean()
mean_actual_HCl_downstream = actual_HCl_downstream.mean()

print(f"Mean Actual Lime Injection: {mean_actual_lime:.4f} kg/h")
print(f"Mean Predicted Lime Injection: {mean_predicted_lime:.4f} kg/h")
print(f"Mean SO2 Downstream emissions: {mean_actual_SO2_downstream:.4f} mg/m3")
print(f"Mean HCl Downstream emissions: {mean_actual_HCl_downstream:.4f} mg/m3")

# 4. Graficar la inyección de cal predicha vs La real

# Crear un gráfico con dos subgráficos
fig, axs = plt.subplots(2, 1, figsize=(10, 8))

# Subgráfico 1: Upstream Measurements (SO2 and HCl)
```

file:///C:/Users/LARRIBAC7649/OneDrive - FCC/Desktop/Stuff/TFM/Anexos/Notebook_RFR_1year.html

5/6

01/07/2025, 09:04

Notebook_RFR_1year

```

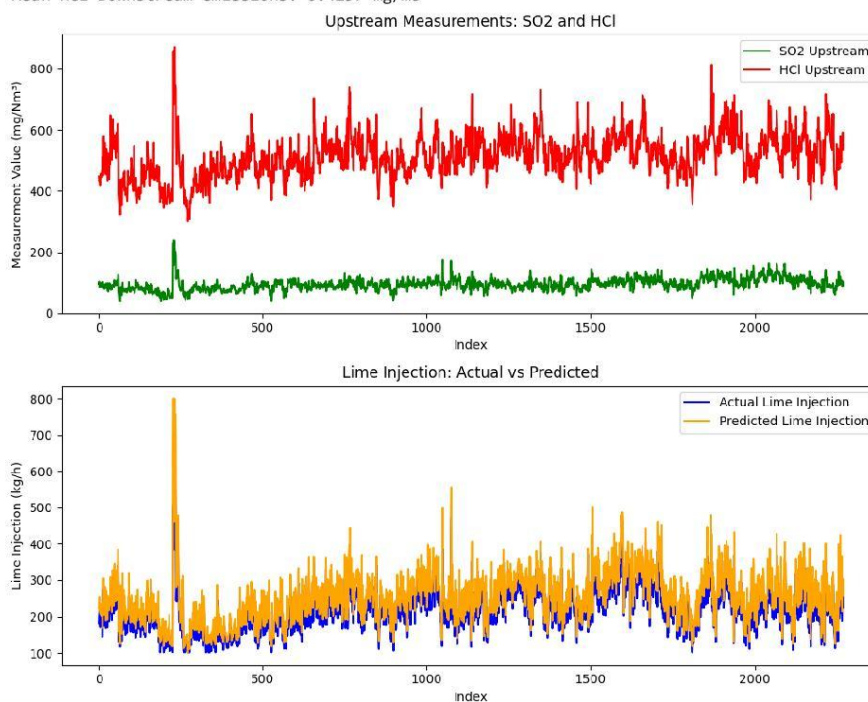
axs[0].plot(SO2_upstream.values, label='SO2 Upstream', color='green', linestyle='--')
axs[0].plot(HCl_upstream.values, label='HCl Upstream', color='red', linestyle='--')
axs[0].set_title("Upstream Measurements: SO2 and HCl")
axs[0].set_xlabel("Index")
axs[0].set_ylabel("Measurement Value (mg/Nm³)")
axs[0].legend()

# Subgráfico 2: Lime Injection (Actual vs Predicted)
axs[1].plot(actual_lime_injection.values, label='Actual Lime Injection', color='blue', line
axs[1].plot(lime_injection_pred.values, label='Predicted Lime Injection', color='orange', l
axs[1].set_title("Lime Injection: Actual vs Predicted")
axs[1].set_xlabel("Index")
axs[1].set_ylabel("Lime Injection (kg/h)")
axs[1].legend()

# Ajustar el espaciado y mostrar el gráfico
plt.tight_layout()
plt.show()

```

Mean Actual Lime Injection: 220.0304 kg/h
Mean Predicted Lime Injection: 253.3314 kg/h
Mean SO2 Downstream emissions: 42.2009 mg/m3
Mean HCl Downstream emissions: 6.4237 mg/m3



PREDICCIÓN 2 MINUTOS EN EL FUTURO

01/07/2025, 09:09

NB_CF

1. Librerías

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import TimeSeriesSplit, RandomizedSearchCV
from sklearn.metrics import r2_score
from xgboost import XGBRegressor
```

2. Carga y preprocesado

```
In [2]: # Cargar dataset maestro
maestro_data = pd.read_csv("Dataset2024_1min.csv")
maestro_data['Time'] = pd.to_datetime(maestro_data['Time'], format='%d/%m/%Y %H:%M')

# Convertir CF a numérico y filtrar rango en una sola pasada
maestro_data['CF'] = pd.to_numeric(maestro_data['CF'], errors='coerce')
maestro_data = maestro_data.query('1 <= CF <= 20').dropna()

# Convertir todas las columnas numéricas a float32 para ahorrar memoria
for col in maestro_data.select_dtypes(include=['float64', 'int64']).columns:
    maestro_data[col] = maestro_data[col].astype('float32')

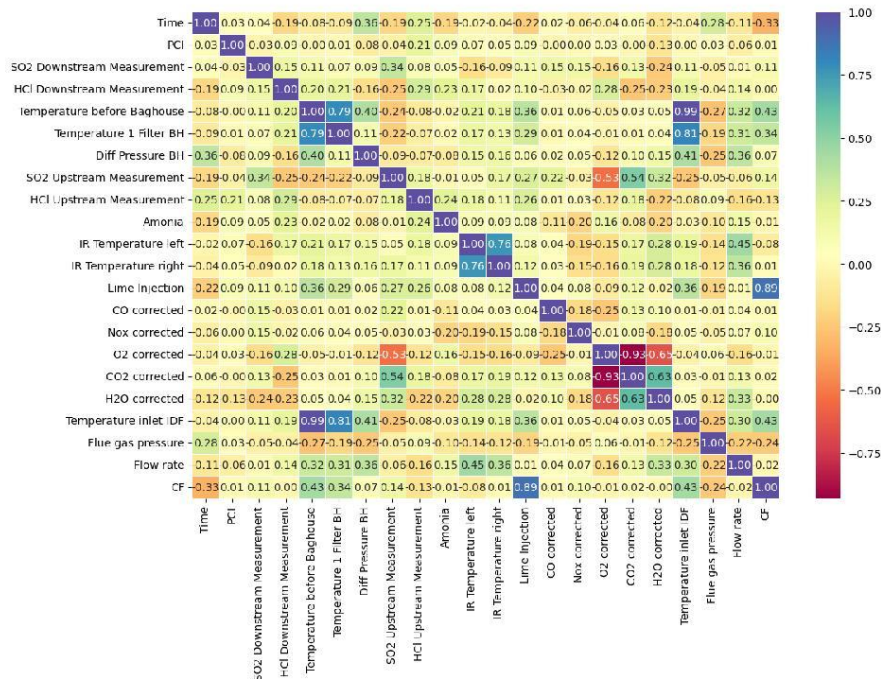
# Redondear todas las variables numéricas a 4 decimales
maestro_data = maestro_data.round(4)

C:\Users\LARRIBAC7649\AppData\Local\Temp\ipykernel_8420\2001674698.py:2: DtypeWarning: Columns (21) have mixed types. Specify dtype option on import or set low_memory=False.
maestro_data = pd.read_csv("Dataset2024_1min.csv")

In [3]: plt.figure(figsize=(12, 8))
sns.heatmap(maestro_data.corr(), annot=True, cmap="Spectral", fmt=".2f", linewidths=0.5)
plt.show()
```

01/07/2025, 09:09

NB_CF



3. Definir características y objetivo

```
In [4]: # Crear Lags y medias móviles para CF y sensores principales
lag_features = ['CF', 'Lime Injection'] # Usa el nombre correcto de la columna
window_sizes = [2, 5]

for feat in lag_features:
    for lag in range(1, 3): # Lags de 1 y 2 minutos
        maestro_data[f'{feat}_lag{lag}'] = maestro_data[feat].shift(lag)
    for w in window_sizes:
        maestro_data[f'{feat}_ma{w}'] = maestro_data[feat].rolling(window=w).mean()

# Eliminar filas con NaN generadas por los lags/medias móviles
maestro_data = maestro_data.dropna().reset_index(drop=True)

# Definir el target como el valor de CF 2 filas hacia adelante (2 minutos)
target = maestro_data["CF"].shift(-2)
maestro_data = maestro_data.iloc[:-2]
target = target.iloc[:-2]

# Excluir columnas no predictivas
excluded_columns = [
    "IR Temperature right", "CF", "SO2 Downstream Measurement", "CO2 corrected",
    "HCl Downstream Measurement", "Lime injection", "Time",
    "Temperature inlet IDF", "Temperature 1 Filter BH"
]

# Excluir también el target y columnas originales de lags/ma si quieres
features = [col for col in maestro_data.columns if col not in excluded_columns]

# Split temporal: 80% train, 20% test (los últimos datos como test)
split_idx = int(len(maestro_data) * 0.8)
X_train = maestro_data.iloc[:split_idx][features]
y_train = target.iloc[:split_idx]
```

01/07/2025, 09:09

NB_CF

```
X_test = maestro_data.iloc[split_idx:][features]
y_test = target.iloc[split_idx:]
```

5. Validación cruzada RandomizedSearchCV

```
In [5]: param_dist = {
        'n_estimators': [100, 200],
        'max_depth': [3, 5, 7],
        'learning_rate': [0.01, 0.05, 0.1],
        'subsample': [0.7, 0.9, 1.0],
        'colsample_bytree': [0.7, 0.9, 1.0],
        'gamma': [0, 0.1, 0.2],
        'min_child_weight': [1, 3, 5]
    }

    cv = TimeSeriesSplit(n_splits=10)
    model = XGBRegressor(random_state=42, n_jobs=-1, tree_method='hist')

    search = RandomizedSearchCV(
        model, param_distributions=param_dist, n_iter=20, scoring='r2',
        cv=cv, refit=True, n_jobs=-1, random_state=42, verbose=1
    )
    search.fit(X_train, y_train)
    best_model = search.best_estimator_

    print("Mejores hiperparámetros:", search.best_params_)
```

Fitting 10 folds for each of 20 candidates, totalling 200 fits

Mejores hiperparámetros: {'subsample': 1.0, 'n_estimators': 200, 'min_child_weight': 5, 'max_depth': 5, 'learning_rate': 0.05, 'gamma': 0.2, 'colsample_bytree': 0.7}

6. Evaluación

```
In [6]: y_test_pred = best_model.predict(X_test)
        test_r2 = r2_score(y_test, y_test_pred)
        print(f"\n🚀 Unseen Final Test R² : {test_r2:.3f}")
```

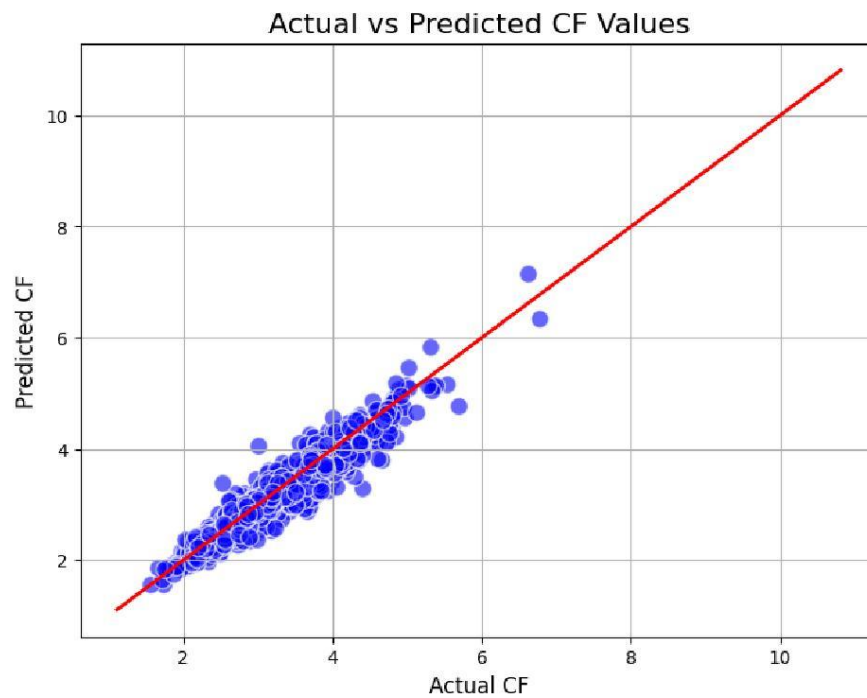
🚀 Unseen Final Test R² : 0.897

7. CF reales vs. predichos

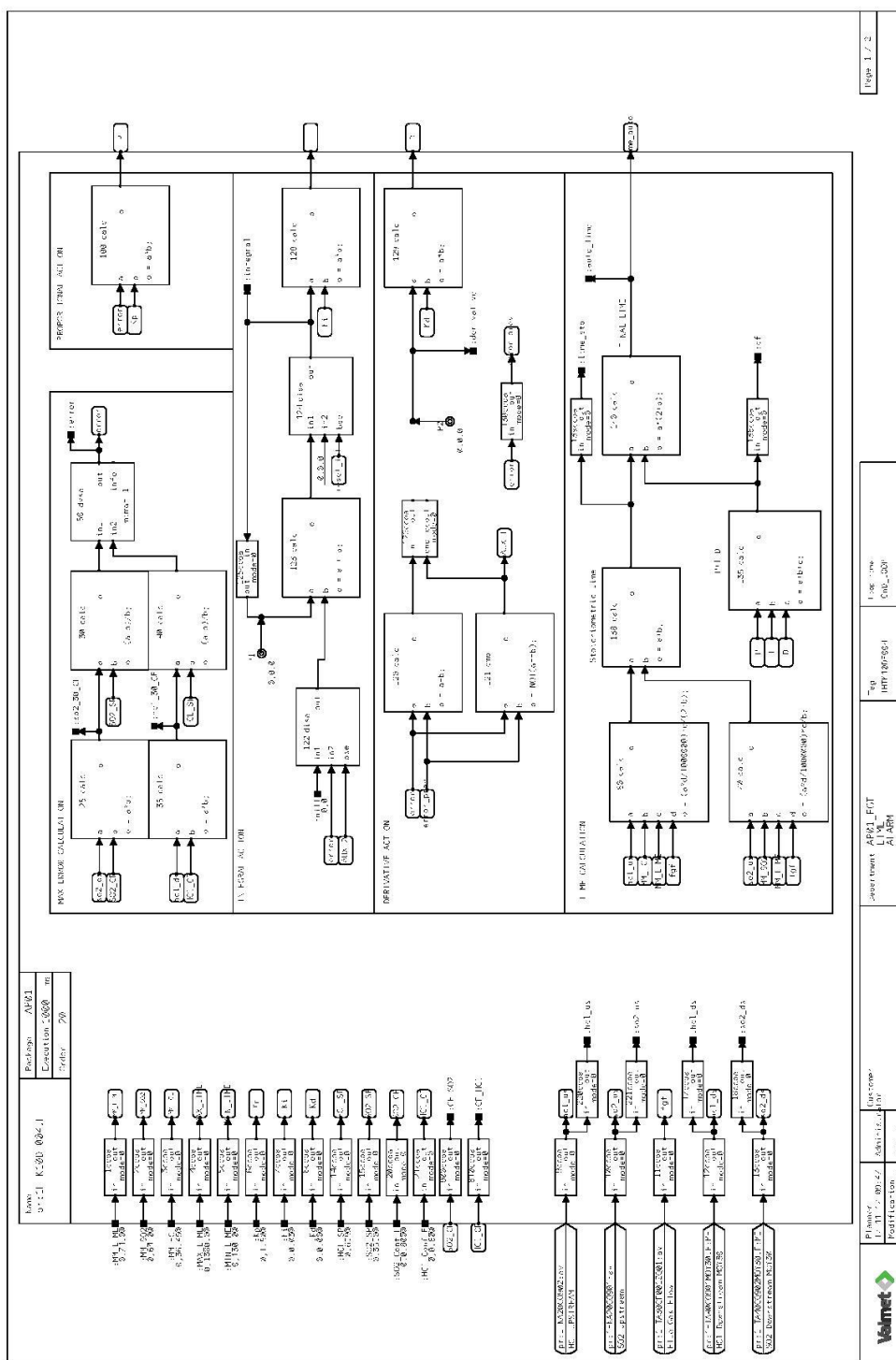
```
In [7]: plt.figure(figsize=(8, 6))
        if len(y_test) > 2000:
            idx = np.random.choice(len(y_test), 2000, replace=False)
            plt.scatter(np.array(y_test)[idx], np.array(y_test_pred)[idx], color='blue', alpha=0.6,
        else:
            plt.scatter(y_test, y_test_pred, color='blue', alpha=0.6, edgecolors='w', s=100)
        plt.plot([min(y_test), max(y_test)], [min(y_test), max(y_test)], color='red', linewidth=2)
        plt.title('Actual vs Predicted CF Values', fontsize=16)
        plt.xlabel('Actual CF', fontsize=12)
        plt.ylabel('Predicted CF', fontsize=12)
        plt.grid(True)
        plt.show()
```

01/07/2025, 09:09

NB_CF



ANEXO IV – DIAGRAMA DE BLOQUES LÓGICO





ANEXO V – ALINEAMIENTO CON LOS OBJETIVOS DE DESARROLLO SOSTENIBLE (ODS)

El proyecto desarrollado guarda una estrecha relación con diversos Objetivos de Desarrollo Sostenible (ODS) establecidos en la Agenda 2030 de Naciones Unidas, al promover prácticas industriales más eficientes, sostenibles y respetuosas con el medio ambiente. A continuación, se expone el alineamiento con los principales ODS implicados:

ODS 3 – Salud y bienestar

La reducción efectiva de contaminantes atmosféricos como el ácido clorhídrico (HCl) y el dióxido de azufre (SO₂) contribuye a una mejora en la calidad del aire, reduciendo riesgos para la salud humana. El sistema de control diseñado permite mantener estos compuestos por debajo de los límites normativos con mayor fiabilidad, minimizando la exposición de la población a emisiones nocivas.

ODS 7 – Energía asequible y no contaminante

El proyecto optimiza la operación de una planta de valorización energética a partir de residuos (EfW), contribuyendo a un modelo de generación energética más sostenible. La eficiencia lograda en el proceso de tratamiento de gases apoya la viabilidad ambiental de este tipo de infraestructuras como parte del mix energético futuro.

ODS 9 – Industria, innovación e infraestructura

La implementación de un sistema de control automático avanzado, incluyendo lógica PID ajustada y desarrollo de modelos predictivos, supone un claro ejemplo de innovación aplicada al sector industrial. Este tipo de soluciones refuerza la digitalización de procesos clave, mejorando su eficiencia, estabilidad y sostenibilidad.

ODS 12 – Producción y consumo responsables

La disminución en el consumo de cal sin comprometer la eficacia en la reducción de emisiones refleja una gestión optimizada de recursos. Esta mejora promueve un uso más racional de materias primas y una reducción del impacto asociado a su extracción, transporte y uso, en línea con los principios de economía circular.

ODS 13 – Acción por el clima

Aunque el objetivo principal del sistema se centra en emisiones ácidas y no directamente en gases de efecto invernadero, la mejora global del proceso de tratamiento de gases y el uso más eficiente de reactivos contribuyen indirectamente a la mitigación del impacto ambiental y al cumplimiento de compromisos climáticos a largo plazo.