

INTERFAZ PARA UN OPTIMIZADOR QUE GESTIONA LA PRODUCCIÓN DE ELECTRICIDAD EN UNA FÁBRICA AZUCARERA

Sergio Hernández¹, Cristian Pablos^{1,2}, L.Felipe Acebes¹

¹Departamento de Ingeniería de Sistemas y Automática, Universidad de Valladolid, 47011 Valladolid, España

² Instituto de Procesos Sostenibles, Universidad de Valladolid, 47011 Valladolid, España
sergio.hernandez.sanchez@alumnos.uva.es, cristian.pablos@uva.es, felipe.acebes@eii.uva.es

Resumen

La optimización de la operación de procesos industriales, se erige como una base fundamental de la nueva revolución industrial. Sin embargo, su implementación en el día a día de las empresas, muchas veces es lastrada debido a la necesidad de conocimientos de programación por parte del usuario final, y lo poco intuitivos que son los entornos desde los cuales se puede ejecutar. En este artículo, se presenta una propuesta para la interacción y manejo de un algoritmo desarrollado en Python, que trata de optimizar la operación de una planta azucarera simulada que participa en el mercado diario eléctrico español. Para ello, se ha desarrollado una Interfaz Hombre Máquina, que actúa como enlace entre optimización y usuario, y que ha sido implementada mediante el software industrial Wonderware y el estándar de comunicaciones OPC UA. Como resultado, el operario es capaz de lanzar optimizaciones con facilidad, e interpretar sus resultados sin necesitar conocimiento alguno de programación.

Palabras clave: SCADA, Comunicaciones Industriales, Optimización, Simulación, Cogeneración.

1 Introducción

Con la llegada de la Industria 4.0, la optimización de procesos industriales se ha convertido en una herramienta imprescindible para mejorar la eficiencia de los procesos productivos, aumentando a su vez el respeto por el medio ambiente, e incrementando los beneficios de su explotación.

La llamada revolución de los datos, está permitiendo tener una mayor información de lo que ocurre en los procesos de producción, pudiendo realizar herramientas más precisas, que además gracias a técnicas como el *Machine Learning* [1] permiten obtener modelos matemáticos de manera sencilla. Python como lenguaje principal de esta revolución, se está erigiendo como actor principal en el

desarrollo de muchas de estas herramientas. Esto es debido a la facilidad de programación que presenta al usuario, la gran cantidad de librerías disponibles, y al hecho de que se trata de un lenguaje abierto y gratuito.

Sin embargo, en muchas ocasiones el exceso de complejidad de las herramientas desarrolladas lastra su implementación en el día a día de las empresas. Entre otros problemas, esto es debido a la falta de formación de los operarios en materias como la programación o la optimización, impidiendo aprovechar al máximo los beneficios que podrían aportar estas aplicaciones.

En este artículo se parte de un caso de estudio en el que se tiene una aplicación programada en Python, que trata de minimizar los costes de operación de una factoría azucarera, manipulando el ritmo de producción y la cantidad de energía eléctrica que se genera en un sistema de cogeneración asociado. De esta forma, se pretende aprovechar la posibilidad de vender excedentes de electricidad generados, en el mercado diario eléctrico español. Cabe destacar que dicha optimización encierra una alta complejidad, y por ello el objetivo de este trabajo es crear una interfaz amigable que sirva para que cualquier usuario final pueda interactuar con ella con independencia de su formación.

Para probar y demostrar el funcionamiento de la interfaz, se ha creado una arquitectura basada en simulación. De esta forma, la planta real es sustituida por una simulación desarrollada con el software EcosimPro[2]. Además, esto servirá para que los futuros usuarios de la herramienta puedan ser entrenados en su uso sin necesidad de poner en peligro el funcionamiento de la planta real.

Por otro lado, debido a las posibilidades que ofrece, y a su amplia presencia en la industria en entornos de supervisión y control de procesos, como software para crear la interfaz propuesta se ha utilizado el programa Wonderware[3]. Para la comunicación de la interfaz, simulación y optimización se ha utilizado el estándar de comunicaciones OPC UA[4]. Dicho

estándar, ha sido elegido por permitir intercambiar grandes cantidades de datos de varias clases a grandes velocidades, evitando así que influya en el comportamiento final de la herramienta.

El resto del artículo se organiza de la siguiente manera: en el segundo apartado se proporciona una explicación del caso de estudio considerado, en el tercer apartado se muestra y explica la arquitectura de comunicaciones desarrollada, en el cuarto apartado se da visión del funcionamiento de la interfaz y, por último, se finaliza con unas conclusiones.

2 Caso de estudio

Debido a la globalización del mercado del azúcar [5], se pretende aumentar la competitividad de una fábrica azucarera, a través de la reducción de sus costes de operación y el aprovechamiento de una planta de cogeneración asociada para explotar las posibilidades que ofrece el mercado eléctrico español.

En la Figura 1 se muestra un esquema del sistema estudiado. A modo de resumen, el proceso de obtención de azúcar considerado se trata de un proceso continuo en el que el azúcar es extraído de la remolacha, obteniendo un jugo que va siendo purificado tras pasar por una serie de subprocesos, y concentrado en una etapa de evaporación. Finalmente, dicho jugo concentrado se cristaliza obteniendo los cristales de azúcar listos para su venta. Todas y cada una de estas etapas requieren de energía eléctrica y calorífica, en forma de vapor, para ser llevadas a cabo. En el caso de estudio propuesto, dicha energía es obtenida de una planta de cogeneración que dispone de tres turbinas de contrapresión capaces de generar un total de 11 MW. Una explicación más profunda del proceso de obtención del azúcar puede encontrarse en [6] y [7].

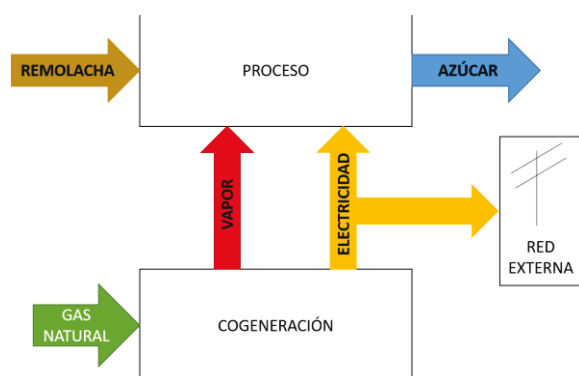


Figura 1. Esquema del sistema considerado

Las dificultades que presentan las diferentes posibilidades en torno a la compraventa de energía eléctrica, la legislación actual aplicada en materia de

cogeneración, y la incertidumbre inherente a los procesos industriales, hacen que muchas empresas que cuentan con tecnología de cogeneración, prefieran no vender su excedente de electricidad, renunciando a una parte importante de ingresos potenciales.

Las opciones de compraventa de energía eléctrica incluyen diferentes tipos de contratos bilaterales que son negociados directamente entre empresas y comercializadoras de servicios eléctricos. Algunos ejemplos pueden ser el de carga base, donde se mantiene un precio fijo para cualquier rango horario y este es mantenido durante un periodo amplio de tiempo, o el contrato por tramos, en el que el precio va cambiando para un grupo determinado de horas. Por otro lado, también se puede participar en diversos mercados eléctricos, donde el precio de la electricidad es mucho más inestable, pero las opciones de obtener mejores beneficios son mayores (mercado diario, mercado intradiario, mercado de futuros...)[8].

Si se escoge participar en el mercado diario, cualquier interacción con el mercado, ya sea de compra o venta de energía eléctrica, debe comunicarse al operador del mercado (OMIE), antes de las 12:00 p.m. del día anterior al negociado [9]. A esta hora, se decide el precio horario que el día siguiente tendrá la electricidad en España. Esta característica hace que las empresas deban predecir la cantidad de electricidad que van a consumir con un día vista, sin saber con certeza cuál va a ser el precio final de la electricidad, lo que hace que alrededor de esta decisión exista una gran incertidumbre. Además, debe tenerse en cuenta que si no se cumple el compromiso de compraventa pactado, la empresa deberá pagar una serie de penalizaciones proporcionales.

Para facilitar la interacción entre la fábrica azucarera estudiada y el operador del mercado eléctrico, se ha desarrollado una herramienta de optimización que permite calcular la cantidad de electricidad que la empresa podría comprometerse a vender en la subasta para el día siguiente, en función de los dos elementos que más incertidumbre provocan en la toma de decisiones: el precio de la electricidad, y la llegada de remolacha a la fábrica el día siguiente. Por lo tanto, además de la cantidad de energía eléctrica comprometida para cada hora, la herramienta deberá determinar la política de producción de azúcar para obtener los mínimos costes posibles.

Así, la herramienta busca modificar a conveniencia el consumo energético de la planta a la vez que decide cuanta electricidad negociar. Debido a las características del proceso de producción de azúcar, la única forma de cambiar significativamente el

consumo energético de la planta es modificar el ritmo de producción [10]. Por lo tanto, para poder generar un mayor o menor excedente de electricidad, sin incumplir con la legislación aplicable, debe modificarse la cantidad de remolacha que se trata en la fábrica. Esto implica que la velocidad de los cambios en la cantidad de electricidad que se quiera vender a la red, está limitada por el tiempo que tarda el proceso de producción en moverse de un punto de operación a otro. Dado que este tiempo es cercano al tiempo que tarda la electricidad en cambiar de precio (una hora), los transitorios no pueden ser despreciados, y el problema tiene que resolverse mediante técnicas de optimización dinámica[11].

La herramienta de optimización ha sido desarrollada utilizando Python como lenguaje de programación. Se ha escogido dicho lenguaje debido a las posibilidades de implementación que ofrece al ser un lenguaje abierto. Además, al ser muy utilizado en la actualidad, posee una gran cantidad de posibilidades de comunicación con otras herramientas, ya sea mediante OPC UA u otros protocolos. Dentro de Python, se ha utilizado el módulo de Pyomo para llevar a cabo la optimización. Este módulo posee características interesantes de cara a la optimización, tales como una sintaxis amigable, discretización y diferenciación automática, y compatibilidad con una gran variedad de solvers tales como IPOPT, CPLEX, BONMIN, GUROBI, entre otros.

Un problema siempre presente con este tipo de herramientas, es que para manejar la optimización, se deben poseer conocimientos de programación. Desde el punto de vista de la empresa, esto puede suponer una limitación importante que puede hacer que la optimización quede en el olvido. Para impulsar la utilización de la aplicación desarrollada se ha creado una interfaz, utilizando un software de control y supervisión de procesos (SCADA) comercial, y que permite interactuar con la aplicación de una forma sencilla. Como se ha mencionado en el apartado anterior, para probar su correcto funcionamiento y entrenar a futuros operarios se ha sustituido el proceso real, por una planta simulada. Los detalles de cada uno de los módulos utilizados en dicha arquitectura se muestran a continuación.

3. Arquitectura de la herramienta

De acuerdo con lo descrito en el punto 2, la arquitectura software de todo el sistema desarrollado viene conformada por tres módulos: una interfaz de visualización, un módulo de optimización y el simulador del proceso. Adicionalmente se ha usado Excel como un módulo intermedio para facilitar la visualización de ciertos resultados imposibles de mostrar en la interfaz gráfica desarrollada.

Las herramientas usadas en el desarrollo del sistema han sido: Wonderware, para el sistema de visualización, Python para el módulo de optimización y EcosimPro como herramienta de modelado y simulación del proceso. Las comunicaciones entre ellos están basadas en el protocolo OPC UA, ya que de una forma u otra las herramientas usadas para el desarrollo de cada módulo soportan este estándar de comunicación (Figura 2).

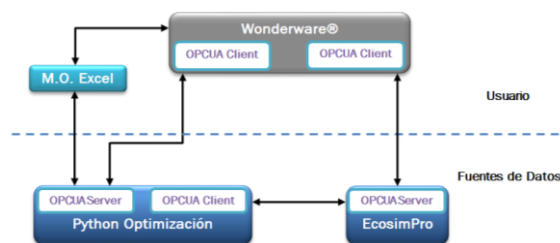


Figura 2. Arquitectura del sistema y comunicaciones entre módulos

La simulación de la planta industrial se ha desarrollado usando el entorno de modelado y simulación de sistemas EcosimPro/PROOSIS. Para la descripción de modelos y experimentos, este software utiliza un lenguaje de programación propio, orientado a objetos: EL, EcosimPro Language. Además, permite el modelado acausal incorporando mecanismos para la manipulación automática de los modelos. Dispone de una potente Interfaz Hombre Máquina (HMI) que facilita la creación, y reutilización, de librerías de modelos, soportando el modelado jerárquico que permite la construcción de sistemas complejos de gran tamaño. Incorpora también una HMI para el diseño y ejecución de experimentos sobre los modelos desarrollados.

Las simulaciones desarrolladas con EcosimPro también pueden ejecutarse de forma autónoma. En concreto, el entorno de desarrollo de EcosimPro permite que un modelo de simulación sea una aplicación independiente que pueda comunicarse a través del protocolo OPC UA, actuando éste como servidor.

Así, el módulo de simulación se ha implementado como un servidor OPC UA al que le llega una determinada ley de control, en nuestro caso calculada por el optimizador que actúa como cliente OPC, y calcula la respuesta del sistema a partir de ella. La respuesta será visualizada usando la interfaz desarrollada, que también actúa como cliente OPC. El servidor OPC de la simulación, contiene diversos métodos que se encargan de manejar su ejecución. Entre otras cosas, son capaces de iniciar el proceso de simulación, establecer un paso de integración para el cálculo y de reiniciar los valores de las variables de la simulación.

Para programar la optimización, se ha utilizado el lenguaje de programación Python. La optimización tiene que ser capaz de generar sus resultados en base a un escenario de precios y llegada de materia prima introducidos por el operario de control a través de la interfaz desarrollada. Estos resultados serán comunicados a la simulación, en función de la decisión del operario, que podrá elegir entre establecer las acciones de control manualmente o delegar en la optimización para una evolución automática del proceso.

Ante la necesidad de comunicar la optimización con la simulación y la interfaz de visualización, se ha introducido una capa servidor/cliente al código desarrollado con Python, que se gestiona internamente. Para ello, se ha utilizado la librería freeOPCUA [12] que cuenta con licencia LGPL de libre acceso. Esta librería proporciona diferentes métodos que facilitan la gestión y configuración del cliente y del servidor. También dispone de varios ejemplos, que permiten al usuario conocer la utilización de los métodos necesarios para una correcta implementación.

Para la realización del diseño gráfico y de la gestión de las comunicaciones anteriormente descritas, se ha utilizado Wonderware. Éste es un software dedicado a la gestión de infraestructuras y operación de datos en tiempo real de entornos industriales. El programa cuenta con motor gráfico de alta calidad, que permite diseñar aplicaciones HMI y SCADAs atractivas para el usuario.

La versión que se ha utilizado en este caso es Wonderware System Platform 2014 R2 SP1, que utiliza un entorno de desarrollo bajo la arquitectura Archestra; ésta actúa como elemento de interconexión entre los diferentes módulos que lo componen. Estos módulos se utilizan principalmente para diseñar gráficamente la aplicación, establecer un protocolo de comunicaciones, gestionar una base de datos históricos y para comunicarse con el paquete de Microsoft Office de Windows.

El programa permite la creación de uno o varios clientes OPC UA que puedan comunicarse con diferentes servidores simultáneamente. Una vez establecida la comunicación con éstos, Wonderware gestionará internamente la comunicación con la aplicación gráfica que se haya diseñado. Para establecer la comunicación con los servidores de la simulación y el optimizador, se han creado dos clientes OPC UA que se conectarán cada uno al servidor correspondiente. La conexión se realiza utilizando un puerto específico, por el cual comenzará la transmisión, utilizando el protocolo OPC UA que se basa en la comunicación TCP/IP.

Esta conexión es gestionada y administrada por un gateway propio del software. Este gateway actúa como traductor de protocolos de comunicación. Por un lado, se encarga de gestionar y almacenar las variables, métodos y objetos enviados por los servidores, y por otro lado, una vez almacenados, se encarga de utilizar los protocolos internos necesarios para establecer una comunicación con la interfaz gráfica y otras herramientas internas que puedan ser necesarias.

Debido a las limitaciones de Wonderware para la manipulación de gráficas, se han utilizado dos formas de representar los resultados. Por un lado, los datos que se mostrarán en la interfaz corresponderán a la información proporcionada por el servidor de la simulación, éstos se corresponden con las variables más relevantes del proceso. Allí se representará su valor de forma visual y en forma de gráficos de tendencia en tiempo real. Por otro lado, para mostrar los resultados proporcionados por el optimizador, se utilizará el programa Excel. Estos resultados permitirán la realización de varios gráficos para el análisis y estudio de los resultados aportados por el optimizador.

La Figura 3, muestra el flujo de datos de cada aplicación, aclarando las situaciones descritas anteriormente. Puede sorprender el hecho de que los precios de la electricidad y la llegada de remolacha se envíen a la optimización por medio de Excel. Esto es debido a que estos datos son tratados mediante vectores, y la forma que tienen de tratarlos Wonderware y Python no es compatible, por lo que se utiliza Excel como traductor.

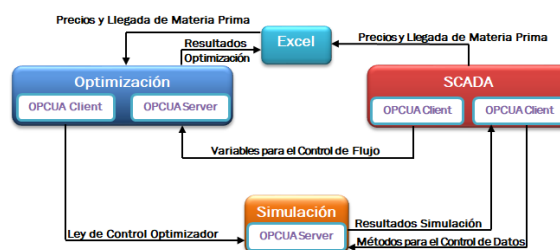


Figura 3. Flujo de comunicación

4. Funcionamiento

El diseño final de la aplicación desarrollada se muestra en la Figura 4. En éste se ha representado un resumen del proceso estudiado, y el acceso a las tareas anteriormente descritas. Además, se ha implementado un botón de información, que permitirá al usuario conocer los pasos a seguir para utilizar la herramienta.

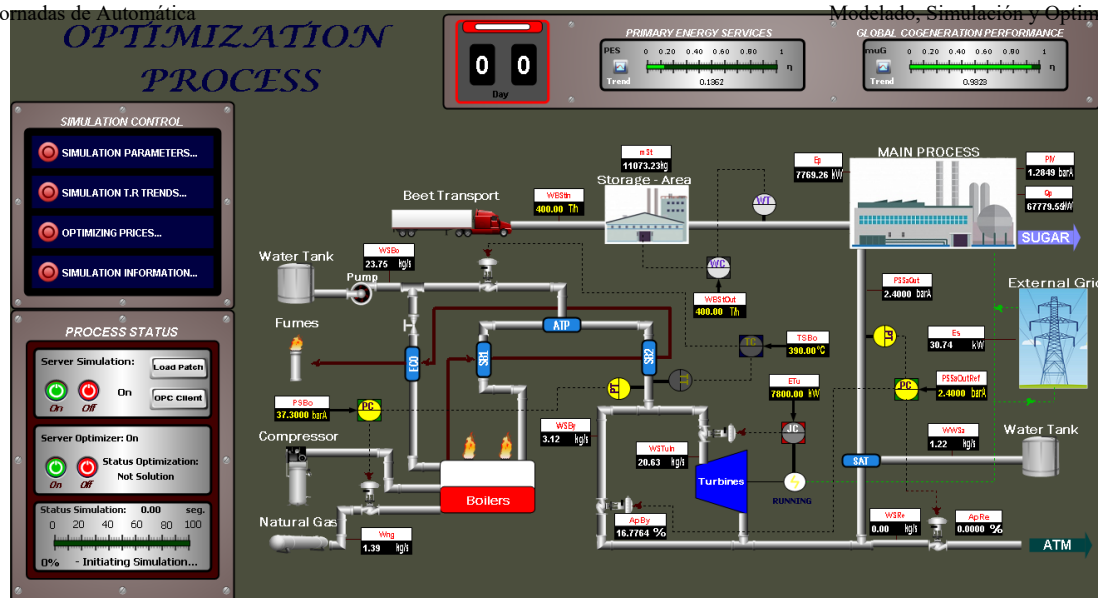


Figura 4. Diseño de la aplicación desarrollada

Inicialmente el usuario mediante la aplicación establecerá en una hoja Excel, un escenario de precios de la electricidad para el día siguiente, y la previsión de remolacha que se va a recibir. Estos datos son los que utilizará el optimizador para realizar los cálculos necesarios y ofrecer al sistema las consignas de operación del proceso y de la cogeneración para el día siguiente. De esta manera, si el operario da permiso, el proceso evolucionará de acuerdo a los resultados aportados por el optimizador automáticamente. Si en algún momento los resultados enviados al proceso no son adecuados, el operario podrá cancelar el control del optimizador y retomar el control manual del proceso.

A continuación, se han agrupado las tres funcionalidades principales de la aplicación desarrollada, para explicar con detalle cada paso a seguir por el operario de control.

- *Control y Supervisión de la Simulación*
- *Datos de Entrada para el Optimizador*
- *Control y Supervisión mediante el Optimizador*

Primeramente, para realizar el control y la supervisión de la simulación es necesario establecer comunicación con el servidor de la simulación, para ello en el menú que se muestra en la Figura 5, se ha implementado un panel de control.



Figura 5. Panel de control del servidor de la simulación.

Este panel permite establecer la ruta de acceso en la que se encuentre el servidor, para que después se pueda activar la comunicación pulsando en el botón verde “On”. Si la ruta se ha introducido correctamente, con la activación de este botón, en la interfaz se comenzarán a monitorizar los datos que le proporciona el servidor. Para verificar que la comunicación se ha realizado correctamente entre el servidor de la simulación y la interfaz, se puede activar el botón “OPC Client”, que proporcionará información sobre el estado de la comunicación.

Cuando la comunicación se haya establecido, se podrán modificar los parámetros necesarios para iniciar la simulación del proceso. En el menú que se muestra en la Figura 6, existe un botón que permite el acceso a los parámetros de la simulación, “SIMULATION PARAMETERS”, donde éstos se podrán configurar.

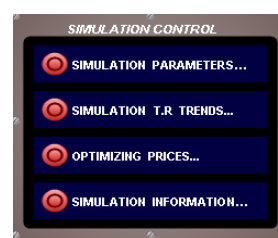


Figura 6. Panel de control de la aplicación.

Dichos parámetros se corresponden con el tiempo de finalización en horas, y el factor de aceleración para establecer la velocidad de simulación del proceso. La ventana que permite su configuración se muestra en la Figura 7.

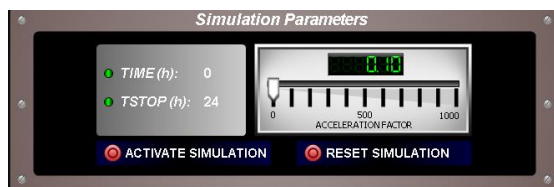


Figura 7. Parámetros de la simulación.

Si los parámetros son adecuados, podrá pulsarse el botón “Activate Simulation” y el servidor comenzará a proporcionar resultados del proceso. Estos resultados se estarán monitorizando y visualizando en el sinóptico del proceso. En éste se tiene la oportunidad de mostrar gráficas de la evolución de las variables en tiempo real, y establecer valores de consigna adecuados en los controladores del proceso. Un ejemplo de gráfica en tiempo real se muestra en la Figura 8.

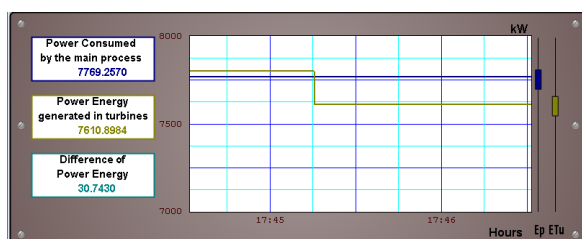


Figura 8. Ejemplo de gráfica en tiempo real.

Éstas, están diseñadas para que su uso sea fácil e intuitivo para el operario. Se han incorporado características que faciliten la mayor cantidad de información posible con un solo vistazo y utilizando colores sin brillo para que no acaben dañando la vista del operario de control.

Para introducir los valores necesarios relativos a los controladores del proceso, se ha creado la ventana que se muestra en la Figura 9. En estas ventanas se pueden establecer los valores de control del algoritmo implementado, y los valores de consigna necesarios en las variables que involucren al control.

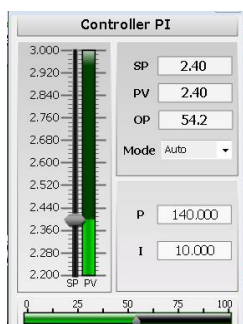


Figura 9. Ejemplo de interfaz para un controlador PI.

Para que el optimizador pueda comenzar a calcular los resultados, es necesario introducir unos datos de

entrada. La manera de ingresar los precios del mercado eléctrico y de la cantidad de remolacha que se va a recibir al día siguiente al simulado, se muestra en la Figura 10.

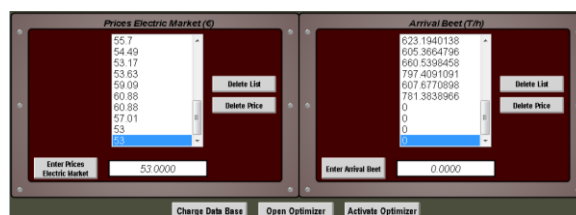


Figura 10. Interfaz para introducir los precios de la electricidad y la llegada de remolacha.

Con estos datos, el optimizador es capaz de generar los resultados con los que se podrá realizar el control del proceso. Para que el optimizador reciba los datos, el operario deberá pulsar en el botón “Activate Optimizer”. Si todo se ha realizado correctamente en el panel de control “PROCESS STATUS”, en la ventana relativa al servidor de la optimización, se activará un mensaje de aviso que mostrará “Control Optimizador: Disponible”, como se puede ver en la figura 11. Si se acepta la ley de control calculada por el optimizador, este tomará el control del proceso a partir de las 12:00 a.m. del día siguiente.



Figura 11. Panel de control del servidor de la optimización

Si el usuario desea volver a tomar el control del proceso, en el panel de control anteriormente mencionado para el control del Servidor de la optimización, se podrá pulsar el botón “Off” que terminará la comunicación con éste, devolviendo inmediatamente el control del proceso al operario. En caso de situaciones anómalas, como el hecho de que la optimización se retrase y no se pueda obtener resultados antes de las 12:00 pm, no se podrá participar al día siguiente en el mercado, y por lo tanto no se podrá ceder el control al optimizador.

5. Conclusiones

En este artículo se ha desarrollado una interfaz hombre máquina, que permite acercar la optimización de la operación de un proceso azucarero en base al mercado eléctrico a cualquier operario, aunque carezca de conocimientos de programación. De esta forma, con el fin de participar en el mercado eléctrico español, se consigue que se puedan evaluar fácilmente, distintos escenarios de llegada de materia prima y precios de la electricidad. Así, este trabajo trata de dar una solución a la implementación de

herramientas de optimización en la industria. Un problema que muchas veces es olvidado en favor del desarrollo de la propia herramienta.

Para ello, se ha empleado software ampliamente utilizado en la industria, como es Wonderware, y el standard de comunicaciones OPC UA. De esta forma se pretende, que el usuario pueda tener incrustado el manejo de la optimización, junto con el control general del proceso, y que pueda acceder a él sin necesidad de abandonar la aplicación SCADA que se esté utilizando para realizar el control básico del proceso.

Agradecimientos

Los autores agradecen el apoyo del Ministerio de Ciencia, Innovación y Universidades (MICINN) a través del proyecto PGC2018-099312-B-C31. Además C. Pablos también agradece la financiación por parte de la beca FPI BES-2016-079388, y el gobierno regional de Castilla y León junto con la EU-FEDER, CLU 2017-09 y UIC 233.

English summary

HMI for the optimal management of the electricity production in a sugar factory

Abstract

The optimization of the operation of industrial processes, stands as a fundamental basis of the new industrial revolution. However, its implementation in the industry daily work is often limited due to the operator's lack of programming knowledge, and the counter-intuitive environments where the optimization is programmed and executed. In this article, we present a proposal for the interaction and management of an algorithm developed in Python, which seeks to optimize the operation of a simulated sugar plant that participates in the Spanish electricity daily market. For this purpose, a Human Machine Interface has been developed, which works as a link between the optimization and the final user. It has been implemented using the industrial software Wonderware and the communication standard OPC UA. As a result, the operator is able to run optimizations easily, and interpret their results without the need of any programming knowledge. This is key in order to push the use of optimization in the industry.

Keywords: SCADA, Industrial communications, Optimization, Simulation, Cogeneration.

Referencias

- [1] J. H. Lee, J. Shin, and M. J. Realff, "Machine learning: Overview of the recent progresses and implications for the process systems engineering field," *Comput. Chem. Eng.*, vol. 114, pp. 111–121, 2018.
- [2] Empresarios Agrupados, "EcosimPro | PROOSIS Modelling and Simulation Software." [Online]. Available: <https://www.ecosimpro.com/>. [Accessed: 19-Jun-2019].
- [3] AVEVA Group Plc, "Wonderware Software - Powering the Industrial World." [Online]. Available: <https://www.wonderware.com/>. [Accessed: 19-Jun-2019].
- [4] OPC Foundation, "OPC UA." [Online]. Available: <https://opcfoundation.org/>. [Accessed: 19-Jun-2019].
- [5] M. Maitah and L. Smutka, "The Development of World Sugar Prices," *Sugar Tech*, vol. 21, no. 1, pp. 1–8, Feb. 2019.
- [6] M. Asadi, *Beet-Sugar Handbook*. John Wiley & Sons, 2005.
- [7] P. W. der Poel, "Sugar technology. Beet and cane sugar manufacture/PW van der Poel, H. Schiweck, T. Schwartz," *Berlin Verlag Dr. Albert Vartens KG*, 1998.
- [8] P. Bennerstedt and J. Grelsson, "Spain's electricity market design -A case study," 2012.
- [9] "Conoce nuestro mercado | OMIE." [Online]. Available: <http://www.omie.es/inicio/mercados-y-productos/conoces-nuestro-mercado-de-electricidad>. [Accessed: 19-Jun-2019].
- [10] C. Pablos, A. Merino, and L. F. Acebes, "Modeling On-Site Combined Heat and Power Systems Coupled to Main Process Operation," *Processes*, vol. 7, no. 4, p. 218, 2019.
- [11] L. T. Biegler, *Nonlinear programming: concepts, algorithms, and applications to chemical processes*, vol. 10. Siam, 2010.
- [12] GitHub Pages, "Freeopcua.github.io | FreeOpcUa: Open Source C++ and Python OPC-UA Server and Client Libraries and Tools." [Online]. Available: <http://freeopcua.github.io/>. [Accessed: 19-Jun-2019].



© 2019 by the authors.
Submitted for possible
open access publication

under the terms and conditions of the Creative Commons Attribution CC BY-NC-SA 4.0 license (<https://creativecommons.org/licenses/by-nc-sa/4.0/deed.es>)