

INFRAESTRUCTURA PARA EXPLOTACIÓN DE DATOS DE UN SIMULADOR AZUCARERO

Jesús M. Zamarreño, Cristian Pablos, Alejandro Merino, L. Felipe Acebes, César de Prada
Grupo de Investigación en Control y Supervisión de Procesos, Universidad de Valladolid
jesusm@autom.uva.es, cristian.pablos@uva.es, alejandro@cta.uva.es, felipe.acebes@eii.uva.es,
prada@autom.uva.es

Resumen

En este trabajo mostramos el desarrollo de una posible infraestructura para implementar algoritmos RTO (Real Time Optimization, optimización en tiempo real) en un sistema industrial. Para ello, construiremos el sistema sobre un simulador de una factoría azucarera que nos permita probar los algoritmos en simulación. Como base de datos industrial que permita el almacenamiento y análisis de los datos usaremos el PI System de la empresa Osisoft®. Finalmente, como entorno desde el que probar los algoritmos RTO usaremos Matlab de la empresa Mathworks®.

Palabras Clave: RTO, Simulador, Base de datos, Energía.

1 INTRODUCCIÓN Y OBJETIVOS

El presente trabajo está enmarcado dentro del proyecto “Integración de Optimización y Control en plantas de procesos” (DPI2015-70975-P) financiado por el Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional (FEDER). Dentro de dicho proyecto se plantea un caso de estudio que consiste en un simulador de planta completa de una factoría azucarera orientado al secuenciamiento y operación conjunta de secciones continuas-batch de gran escala, así como el desarrollo de sistemas RTO para la gestión energética (electricidad y vapor) en la factoría, y la coordinación de ambos RTO. Para hacer posible las pruebas de estos sistemas, además del propio simulador de planta completa (ya disponible como resultado de proyectos previos [4]), es necesario un sistema de gestión de datos industrial que permita un registro de los datos en tiempo real del simulador, así como un análisis de los mismos, y un entorno donde implementar y probar los algoritmos RTO.

Los sistemas RTO buscan determinar en tiempo real las mejores condiciones de operación de un proceso, normalmente considerando objetivos económicos, a la vez que se satisfacen una serie de condiciones

impuestas por las características físicas del mismo y los condicionantes de seguridad, calidad, etc., que puedan existir.

Debido a la creciente competencia a nivel mundial, a los costes energéticos, las exigencias de calidad en los productos, la necesidad de cumplir con normativas medioambientales o de seguridad, etc., así como a la complejidad creciente de los procesos productivos, una vez que se ha alcanzado un nivel básico de control y estabilidad en la operación de una planta, cada vez es más importante saber operar los procesos de forma óptima. Para ello, se debe decidir cuanto y cuando hay que producir en cada subproceso, las fuentes de energía más adecuadas, los puntos de operación de las unidades, etc. de modo que, respetando los rangos de operación admisibles de cada subproceso, el conjunto de los elementos de una planta conduzcan a la operación más eficiente y segura posible [3].

Los sistemas RTO están basados en un modelo del proceso y toman sus decisiones resolviendo un problema de optimización del tipo:

$$\begin{aligned} \min_{u,x,\theta} \quad & J(u,x) \\ h(x,u,\theta) &= 0 \\ g(x,u,\theta) &\leq 0 \end{aligned} \quad (1)$$

Donde u son las variables de decisión o grados de libertad del proceso, x otras variables del mismo, J la función de costo económica que se quiere minimizar, h representa el modelo del proceso, con θ indicando sus parámetros, y g las restricciones de operación del mismo.

El modelo suele formularse en estado estacionario, dado que se busca encontrar las mejores condiciones de operación en las que debe mantenerse el proceso, las cuales se pasan a los sistemas de control como valores de referencia que deberán ser alcanzados y mantenidos por el mismo, según el esquema que puede verse en la parte izquierda de la Figura 1, en el que MPC representa el control predictivo multivariable. El problema (1) se resuelve

periódicamente a intervalos regulares para adaptarse al estado del proceso.

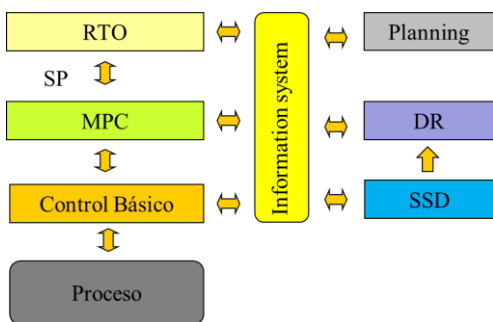


Figura 1: Jerarquía de acciones de decisión y control, sistema de información de la planta e implementación de RTO

Para formular específicamente y resolver el problema RTO (1), se requiere información tanto de los objetivos, precios, etc. impuestos por la planificación general de la producción en la factoría considerada, como información del estado del proceso. Ahora bien, dado que muchas variables del modelo no son medidas con precisión, y que los parámetros del mismo pueden ser inciertos o haber cambiado, lo más frecuente es acompañar al RTO de un problema de reconciliación de datos (DR), que se ejecuta con su misma periodicidad, y que proporciona valores de los parámetros θ del modelo y de las variables del mismo coherentes con el modelo y lo más cercanas posible a los valores medidos. Este problema de reconciliación de datos (DR) se formula también como uno de optimización, con el mismo modelo:

$$\begin{aligned} \min_{u, x, \theta} \quad & \sum [x - x_{med}]^2 + [u - u_{med}]^2 \\ h(x, u, \theta) = \quad & 0 \\ G(x, u, \theta) \leq \quad & 0 \end{aligned} \tag{2}$$

Donde el subíndice *med* indica valores medidos en planta. De este modo, el modelo se adapta a los posibles cambios en el proceso y se dispone de valores de las variables medidas y no medidas fiables. Como los cálculos se efectúan para estado estacionario, junto a la reconciliación de datos suele incluirse un módulo de detección de estado estacionario SSD.

Dado que son muchos los módulos que intercambian información sobre el estado del proceso y las acciones previstas, a nadie se le escapa el papel clave que juega el disponer de información actualizada, correcta y accesible a todos los niveles, para la realización de las funciones del RTO, así como de otras muchas [2]. Esto es, de un sistema de información que actúe como eje vertebral de comunicaciones, de modo que un esquema de

implementación del sistema RTO próximo a la realidad queda reflejado en la Figura 1 en su totalidad.

El artículo se estructura en 6 secciones. En esta primera sección de introducción hemos planteado el problema que vamos a abordar y los objetivos perseguidos. En la sección 2 describiremos someramente el simulador azucarero de planta completa. El sistema de gestión de datos se detalla en la sección 3. Las secciones 4 y 5 muestran el nivel de desarrollo actual del trabajo y el planteamiento futuro del mismo, respectivamente. Por último, terminamos con unas conclusiones generales en la sección 6.

2 EL SIMULADOR

Los simuladores son herramientas muy útiles en la industria de procesos, que permiten sustituir parcial o globalmente una planta industrial con fines muy diversos. Por ejemplo, la prueba y sintonía de sistemas de control, la incorporación o modificación de algunas partes de la planta, o incluso el entrenamiento de operarios de sala de control. La industria azucarera se caracteriza por ser un proceso muy complejo (Figura 2), que presenta gran parte de las operaciones unitarias presentes en la industria de proceso que operan en conjunto con dinámicas muy diferentes haciendo que modelarlo suponga un verdadero reto.

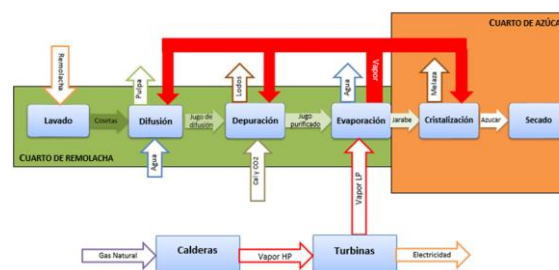


Figura 2: Esquema del proceso

Como se puede observar en la Figura 2, en el proceso azucarero existe una gran interacción energética entre las diferentes partes del proceso, relacionada sobre todo con el consumo del vapor procedente de la sección de evaporación por otros subprocesos. Además, se puede entrever que una de las mayores dificultades del modelado de la fábrica reside en que no solamente se trata de modelar el proceso principal, sino también otros subsistemas asociados, como la generación de vapor en las calderas o de electricidad en las turbinas.

Para el desarrollo del simulador se utilizó como herramienta de modelado y simulación EcosimPro® [5]. Este entorno de modelado y simulación de

carácter general, utiliza EL (EcosimPro Language) como lenguaje de programación perteneciente a la familia de lenguajes basados en ecuaciones y orientados a objetos. El paradigma del modelado orientado a objetos permite reducir drásticamente el tiempo dedicado al modelado de un sistema, gracias a características como modularidad, herencia y agregación.

Respecto a la arquitectura del simulador, cabe destacar tres líneas bien diferenciadas: la simulación, la comunicación entre simulaciones, y la interfaz de usuario del simulador. Respecto a la simulación, en primer lugar se desarrollaron una serie de librerías que contienen los elementos de proceso existentes en una fábrica [7]. Estos componentes abarcan desde intercambiadores de calor, depósitos, tuberías y válvulas, hasta equipos mucho más complejos y específicos como difusores, evaporadores, filtros o cristalizadores. Dichas librerías y la interfaz gráfica de EcosimPro permitieron, mediante la agregación de los componentes de las librerías de proceso, la creación de modelos mucho más complejos que representan fielmente el comportamiento de las distintas secciones en las que se divide el proceso. Se elaboraron un total de diez modelos de secciones o subsecciones diferentes. Estos modelos, además de simular el comportamiento del proceso en condiciones de operación normales, incluyen la posibilidad de provocar fallos en diversos equipos de la planta. Esto último resulta de gran relevancia ya que el objetivo inicial del simulador consiste en el entrenamiento de operarios de sala de control [8].

Debido al gran tamaño del modelo que se obtendría si se unieran los modelos de las secciones individuales en uno único, se hace imposible la ejecución del mismo en un único ordenador. Por ello, el simulador que nos ocupa se ejecuta de manera distribuida utilizando varios ordenadores para simular el comportamiento global de la planta, concretamente cinco.

Tanto para la comunicación de las simulaciones individuales, como para la comunicación entre las simulaciones y la interfaz de usuario, implementada usando un SCADA, se utilizó el estándar OPC [9]. Así, se crearon servidores OPC asociados a las simulaciones individuales, de modo que posteriormente se pudieran comunicar e intercambiar datos entre ellas. Para la creación de los distintos servidores OPC se desarrolló un software propio denominado CreaOPC, obteniéndose un total de diez servidores OPC a partir de los mencionados modelos de secciones completas y subsecciones.

Se desarrolló el software UneSim para intercambiar variables entre los servidores OPC cada intervalo de comunicación, y a su vez sincronizar la simulación

completa permitiendo que las diversas simulaciones avanzaran de forma coordinada en tiempo real o en proporción al mismo [12].

Finalmente, se desarrolló un completo SCADA, denominado EDUSCA [13], de modo que los usuarios pudieran manejar el simulador como si de una sala de control se tratase. Este SCADA es capaz de comunicarse vía OPC con las simulaciones de proceso. En la Figura 3 puede observarse un esquema conjunto del proceso de creación de un componente del simulador completo y de las comunicaciones asociadas.

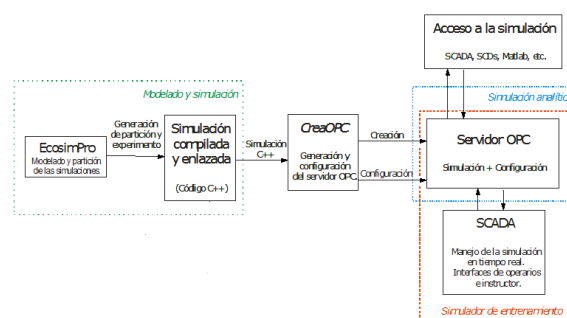


Figura 3: Diseño y comunicaciones del simulador

3 GESTOR DE DATOS

La empresa Osisoft® [11] comercializa un producto llamado PI System con capacidades para recoger, analizar, visualizar y compartir grandes cantidades de datos provenientes de múltiples fuentes con el objetivo de mejorar la inteligencia operacional de las empresas. Facilita la toma de decisiones en tiempo real en base a los datos actuales e históricos. Consta de una base de datos centralizada diseñada para almacenar millones de datos, tanto series de tiempo como datos basados en eventos. Dispone de herramientas que son capaces de analizar tanto la información en tiempo real como los datos pasados para descubrir información de interés sobre el funcionamiento de la planta. Es un producto utilizado por algunas de las empresas más importantes a nivel mundial, sobre todo del campo petroquímico, energético, etc.

El PI System consta de una serie de componentes que se comunican entre sí y facilitan la distribución de tareas y la flexibilidad y ampliación del sistema. Estos componentes y su interconexión se pueden ver en la Figura 4.

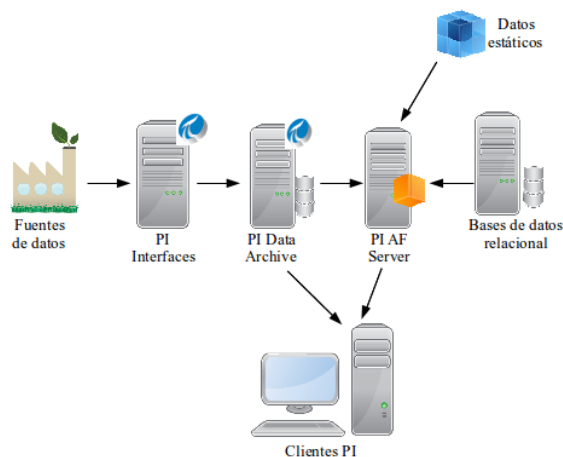


Figura 4: Componentes de un PI System

El PI System dispone de más de 400 interfaces para conectarse a fuentes de datos, siendo una de las más populares la interfaz de conexión a OPC DA [14], que será la que utilizemos en nuestro proyecto. Tal y como se ha comentado en la sección anterior, el simulador azucarero de planta completa está estructurado como un conjunto de servidores OPC, donde cada uno de ellos representa una parte de la factoría, y que son coordinados entre sí por un componente de nivel superior. Esto facilita que el PI System recoja las variables de interés del simulador vía OPC en tiempo real y las almacene internamente.

Una vez que los datos fluyen hacia el PI System, estos son almacenados en un servidor llamado Data Archive. Existe otro servidor llamado AF (Asset Framework) que estructura de forma conveniente tanto los datos dinámicos provenientes del Data Archive como datos estáticos y otras fuentes como tablas de bases de datos relacionales, ficheros de hojas de cálculo, etc.

Una de las ventajas del servidor AF es que, además de estructurar y fusionar la información de múltiples fuentes, es capaz de realizar análisis históricos sobre los datos, por ejemplo, tiempos de operación de unidades o equipos, disponibilidad de los mismos, rendimientos, análisis de calidad, etc, y ofrecer estos resultados como nuevos datos disponibles a los clientes. Además, incluye detección de eventos y notificaciones.

Finalmente, dentro de la estructura general, existirán diversos tipos de clientes que harán uso tanto de los datos en bruto, como los estructurados o los resultados de los análisis. Los más conocidos dentro del ecosistema de Osisoft serían ProcessBook y Coresight (ahora llamado Visio) que permiten construir interfaces gráficas de visualización, y DataLink para enlazar con MS Excel.

Pero si queremos realizar cálculos avanzados sobre los datos, un complemento perfecto sería usar la versatilidad y potencia algorítmica de las diversas librerías construidas sobre Matlab [6]. Las posibilidades de comunicación entre ambos sistemas son variadas [10], cada una con sus ventajas y sus inconvenientes; nosotros hemos optado en principio por usar la interfaz PI Web API que ofrece el PI System, por su carácter estándar y abierto (funciona en todos los sistemas operativos, se comunica vía HTTP y se basa en el formato JSON [1]). De esta forma, una visión simplificada de la comunicación entre los diversos elementos la podemos ver en la Figura 5. Aunque en nuestro caso planteemos Matlab como medio de desarrollo de algoritmos, es posible utilizar otros entornos o lenguajes de programación de propósito general.

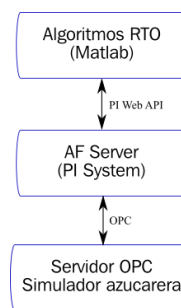


Figura 5: Esquema de comunicación

4 DESARROLLO ACTUAL

Tal y como se ha comentado en la sección 1, el simulador azucarero de planta completa ya fue desarrollado en trabajos anteriores en el entorno EcosimPro® [5] y desplegado como una simulación distribuida encapsulada en varios servidores OPC, cada uno de ellos correspondiente a una sección de la factoría y coordinados entre sí por un supervisor de nivel superior.

Si subimos de nivel, según la Figura 5, el siguiente paso sería llevar los datos del simulador al PI System. Para ello, fue necesario realizar los siguientes pasos:

1. Configurar los servidores OPC correspondientes al simulador para su correcta conexión remota por parte del PI System. Esto involucra configurar adecuadamente los parámetros de seguridad DCOM de los servidores.
2. Configurar una nueva interfaz PI para OPC desde la herramienta Interface Configuration Utility del PI System.
3. Crear los puntos PI en el servidor del PI System. Dado que se trataba de miles de variables, se realizó de forma semi-

automática a través de una combinación de un script en Matlab y el PI DataLink en Excel.

4. Crear la estructura de elementos en el servidor AF, con ayuda de plantillas en aquellos casos donde aparecen elementos del mismo tipo (controladores PID, evaporadores, turbinas, depósitos, etc.).

Actualmente, se ha completado en el servidor AF la configuración de la sección de evaporación, calderas y turbinas, y Tacha I del cuarto de azúcar (Figura 6).

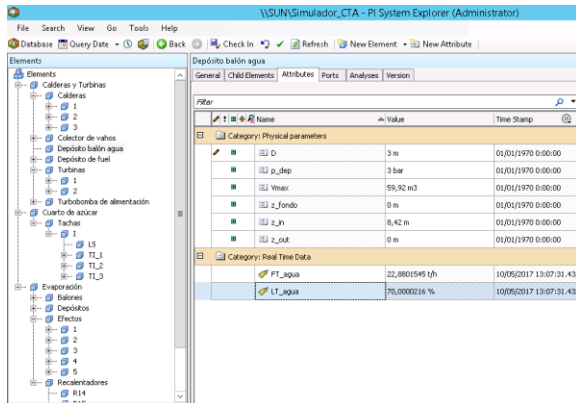


Figura 6: Estructura en el servidor AF

Como se puede ver en dicha figura, la estructura de elementos en forma de árbol permite acceder fácilmente a cada uno de ellos para ver sus atributos, tanto estáticos (por ejemplo, parámetros de construcción) como dinámicos (variables provenientes de los transmisores disponibles en planta).

Una vez que los datos están disponibles en el servidor AF, y siguiendo con el esquema de la Figura 5, nos quedaría establecer la comunicación con el entorno donde estén implementados los algoritmos RTO. Como se comentó en la sección anterior, uno de los métodos más versátiles para ello es el uso de la PI Web API. Para mostrar su facilidad de uso, se muestra en la Figura 7 el esquema básico con la información de entrada requerida por la función webread de Matlab y el resultado como estructura JSON.

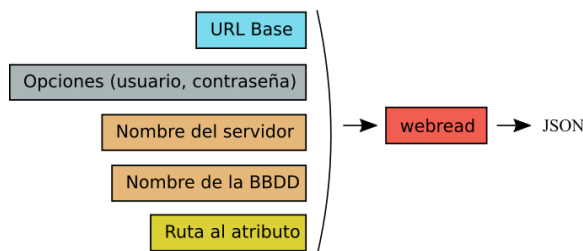


Figura 7: Uso de la función webread de Matlab

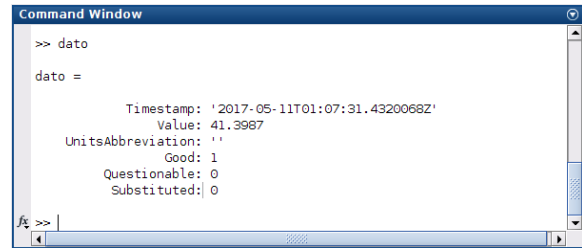


Figura 8: Obtención del dato actual en el PI System desde Matlab

5 PLANTEAMIENTO DE LOS ALGORITMOS RTO

Además de completar la configuración del servidor AF añadiendo el resto de secciones del simulador azucarero: Difusión y Secadero, Depuración, Tachas II, Tachas Tercera, Tacha Continua, Tacha Magma, Centrifugas y Refundidora, se deben desarrollar e implementar diversos algoritmos RTO que permitan mejorar la eficiencia energética de la factoría azucarera.

Se pretende así realizar una herramienta de optimización de la producción de una planta azucarera, que cuenta con un sistema de cogeneración asociado para autoabastecerse energéticamente y con la posibilidad de generar energía eléctrica para el exterior. La planificación de la producción deberá tener en cuenta los costes de la energía, de forma que una función de beneficio económico, sujeta a unas restricciones dinámicas, tenga el mayor valor posible. En la Figura 9 puede observarse un esquema básico del sistema asociado al problema de optimización/planificación.

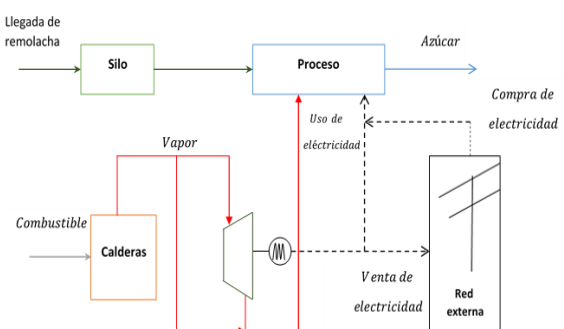


Figura 9: Esquema del proceso

La optimización además deberá cumplir diversas restricciones relacionadas con el proceso y su viabilidad, y otras aplicadas desde la UE a este tipo de procesos, que exigen altos índices de eficiencia energética a la hora de producir energía eléctrica.

Para poder llevar a cabo la optimización en tiempo real teniendo en cuenta la dinámica del proceso, se

necesitarán modelos dinámicos reducidos que relacionen la cantidad de remolacha a tratar en la fábrica con el consumo energético de la misma. Una vez obtenidos los modelos reducidos y el RTO en su conjunto, se procederá a probarlo en el simulador descrito en el apartado 2.

6 CONCLUSIONES

En este trabajo se ha mostrado el esquema de una infraestructura para RTO donde se hace uso de un sistema de gestión de datos de ámbito industrial que amplía el abanico de posibilidades de cara a la explotación y análisis de dichos datos. Debido a su carácter abierto, los algoritmos RTO a desarrollar en entornos de programación más potentes, como por ejemplo Matlab, pueden acceder a dicha información de forma eficiente. Aquí estaríamos hablando no solamente de datos en tiempo real, sino también de los datos de análisis de interés generados por el propio sistema gestor de datos.

Aún queda lo más interesante: probar la efectividad de diversos algoritmos RTO con los que conseguir una gestión energética más eficiente en la factoría azucarera. Pero, por el momento, la infraestructura que haga posible estas pruebas está ya preparada.

Agradecimientos

Los autores agradecen el apoyo del Ministerio de Economía y Competitividad y el Fondo Europeo de Desarrollo Regional (FEDER) a través del proyecto DPI2015-70975-P.

Referencias

- [1] Colaboradores de Wikipedia, "JSON," Wikipedia, La enciclopedia libre, <https://es.wikipedia.org/w/index.php?title=JSON&oldid=98972971> (acceso 11 de mayo de 2017).
- [2] de Prada, C., (2015) "Overview: Control hierarchy of large processing plants.", *Encyclopedia of Systems and Control* · Article ID: 366985 · Chapter ID: 241 Springer Verlag London, Edt. J. Baillieul, T. Samad, 10.1007/978-1-4471-5058-9_241.
- [3] de Prada, C., Gutierrez, G., (2012) "Present and future trends in process control", *Ingeniería Química, edición especial IChE*, ISSN: 0210-2064, nº 505, año XLIV, pp.38-42.
- [4] de Prada, C., Mazaeda, R., Acebes F., Rueda, A., Merino, A., (2012) "Sugar factory simulator for operators training", 7th Symposium on Advances in Control Education ACE2006, IFAC.
- [5] Empresarios Agrupados, EcosimPro, <http://www.ecosimpro.com> (acceso 12 de mayo de 2017).
- [6] Mathworks, Matlab, <http://es.mathworks.com> (acceso 12 de mayo de 2017).
- [7] Merino, A., Acebes, L.F., Mazaeda, R., de Prada, C., (2009) "Modelado y Simulación del Proceso de Producción del Azúcar", *RIAI*, vol. 6, nº 3, pp. 21-31.
- [8] Merino, A., Mazaeda, R., Alves, R., Rueda, A., Acebes, L.F., de Prada, C., (2006) "Sugar factory simulator for operators training", *Advances in Control Education*, pp. 259-264.
- [9] OPC Foundation, Home Page, <http://www.opcfoundation.org/> (acceso 17 de mayo de 2017).
- [10] Osisoft, LLC, (2014) "White Paper - Using PI Data with MATLAB", <https://pisquare.osisoft.com/docs/DOC-1305> (acceso 11 de mayo de 2017).
- [11] Osisoft, PI System, <http://www.osisoft.com> (acceso 12 de mayo de 2017).
- [12] Santos, R.A., Normey-Rico, J.E., Gómez, A.M., Arconada, L.F.A., Moraga, C. de P., (2008) "Distributed continuous process simulation: An industrial case study", *Computers & Chemical Engineering*, 32, pp. 1195–1205. doi:10.1016/j.compchemeng.2007.04.022.
- [13] Santos, R.A., Normey-Rico, J.E., Gómez, A.M. de Prada Moraga, C. (2006), "EDUSCA (EDUCATIONAL SCADA): FEATURES AND APPLICATIONS", *IFAC Proceedings Volumes*, 39, pp 614–619. doi:10.3182/20060621-3-ES-2905.00105.
- [14] Zamarreño, J.M., (2010) "Acceso a datos mediante OPC", Editorial Andavira SL, Santiago de Compostela.