

ADAPTACIÓN DINÁMICA DE FRECUENCIA DE INTERCAMBIO DE DATOS EN UNA SIMULACIÓN DISTRIBUIDA

Fernando Gutiérrez

Grupo de Control y Supervisión de Procesos. Universidad de Valladolid (UVa). Institute of Sustainable Processes. C/ Dr. Mergelina s/n, 47011 Valladolid, España, fgutrod@autom.uva.es

Rogelio Mazaeda

Grupo de Control y Supervisión de Procesos. Universidad de Valladolid (UVa). Institute of Sustainable Processes. C/ Dr. Mergelina s/n, 47011 Valladolid, España, rogelio@cta.uva.es

Jesús M. Zamarreño

Grupo de Control y Supervisión de Procesos. Universidad de Valladolid (UVa). Institute of Sustainable Processes C/ Dr. Mergelina s/n, 47011 Valladolid, España, jesusm@autom.uva.es

Resumen

Este artículo describe un administrador de simulación dinámica distribuida, usando programación en PYTHON, que utiliza el estándar de interoperabilidad OPC UA para comunicarse con los diferentes servidores que constituyen una simulación distribuida, actuando como cliente configurable. Los servidores OPC UA se implementan utilizando una herramienta de modelado de procesos y simulación externa. Para que dicha herramienta tenga un funcionamiento acorde a las necesidades de la simulación, debe disponer de alguna forma de control de la frecuencia de intercambio de datos a medida que el proceso evoluciona.

En la presente contribución se describe el desarrollo de una estrategia de adaptación de la frecuencia de intercambio de información entre los diferentes servidores mediante comparación con una simulación de referencia desarrollada en paralelo o por la obtención de una referencia anterior. Mediante esta estrategia, se lleva a cabo la configuración de las condiciones de simulación, la medición del error y la adaptación del tiempo de intercambio, de manera análoga a los algoritmos de control de paso empleados tradicionalmente por algunos algoritmos de integración dinámica.

Palabras clave: simulación, OPC UA, intercambio, adaptación, error.

1 INTRODUCCIÓN

En la actualidad, los métodos de modelado y simulación de procesos dinámicos se encuentran entre las actividades fundamentales de cualquier proyecto científico o técnico. Las etapas de desarrollo de una solución técnica requieren del

empleo de modelos lo más fidedignos posibles al proceso real. Dentro de este contexto cabe señalar las contribuciones realizadas en [1] y [2]. Se llega a un compromiso entre la exactitud de los resultados proporcionados por la simulación y la carga computacional requerida. Existiendo diferentes alternativas para lidiar con este compromiso, como puede ser el uso de modelos jerárquicos, o la adoptada en este trabajo, de dividir la simulación original y distribuirla en varios procesos en el ordenador, que puedan ser incluso desplegadas en varios ordenadores conectados red. Ejemplos de estas simulaciones pueden consultarse en [5] y [7].

Para una estrategia de simulación distribuida, se dispone de varias simulaciones numéricas al mismo nivel, interrelacionadas entre sí, de tal forma que unas dependen de las otras y necesitan intercambiar información con una frecuencia determinada. Esta frecuencia podría ser fija, tanto en el tiempo como a lo largo de toda la simulación, y tendría que ser la menor posible para garantizar el grado de exactitud requerido, de acuerdo al propósito de la simulación. En este punto hay que mencionar que podría ser muy conveniente relajar esta imposición. En ocasiones, cada partición puede tener requerimientos diferentes, pero una situación más frecuente, es que estos requerimientos pueden modificarse a lo largo del tiempo de simulación. En aquellos momentos en los que el proceso se encuentre en una situación con una dinámica cuasi-estacionaria, o con variaciones lentas, el tiempo de intercambio puede ser mayor, pero en otros intervalos donde existan cambios bruscos, se precisa de una frecuencia de intercambio inferior.

En esta contribución se comenta un algoritmo de variación de la frecuencia de intercambio, teniendo en cuenta la magnitud del error que se comete, de manera análoga a como se realiza en los algoritmos de integración de paso variable (como puede ser RK45). Aquí el error se mide por comparación con

una simulación monolítica de mismo proceso, con el objetivo, aunque se pueden obtener estimaciones suficientemente buenas de ese error, apelando a la propia simulación distribuida como se indica más adelante. Se aprovechan además las posibilidades que ofrece el estándar OPC UA [6]. Una forma de acceso a datos mediante OPC UA puede verse en [8].

2 MATERIALES Y MÉTODOS

2.1 HERRAMIENTAS DE SOFTWARE EMPLEADAS

La estrategia de variación del tiempo de intercambio programada forma parte de la librería de control de la herramienta de gestión distribuida DSM-UA. Esta herramienta está constituida por una serie de librerías, con misiones determinadas cuyo alcance queda fuera del presente artículo. La estructura de esta herramienta queda reflejada en la figura 1.

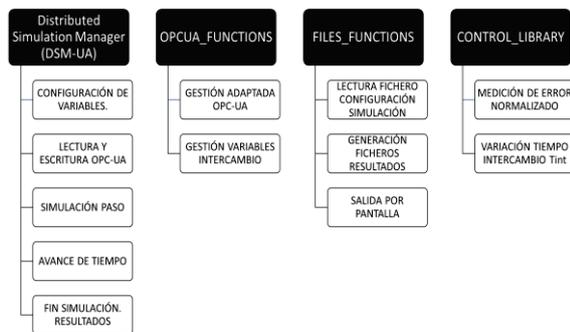


Figura 1: Librerías y atribuciones de la herramienta de gestión distribuida empleada.

La filosofía seguida para el desarrollo de la estrategia de variación de la frecuencia de intercambio aprovecha la experiencia adquirida en el diseño de la herramienta DSM-UA, mediante el empleo del lenguaje de programación PYTHON. Aprovechando las ventajas que ofrece este lenguaje de programación, y el amplio soporte a multitud de plataformas, se continúa con el manejo de las estructuras de datos adecuadas, de tal forma que la información sea accesible de forma intuitiva sin más que conocer los nombres de las variables, así como sus correspondientes valores simulados. Las funciones constituyentes de la librería de control interactúan con los servidores OPC UA constituyentes de la simulación distribuida. La obtención de estos servidores puede consultarse en [9].

2.2 MODELO EMPLEADO

Para verificar el funcionamiento de la estrategia de control de la frecuencia de intercambio, se emplea un modelo de referencia, que haga el papel de test. Estos modelos se diseñan en ECOSIMPRO [3]. Para el modelo test, se dispone de la simulación de un proceso sencillo, que consiste en la unión en serie de un número determinado de depósitos, mediante un parámetro de construcción en la etapa de diseño del modelo. El hacerlo de esta forma, permite modificar de manera fácil, la carga computacional que supone la simulación y evaluar, en futuros trabajos, los posibles beneficios de realizar la distribución frente a la simulación monolítica. En el presente trabajo, se configura el modelo original con 50 depósitos, mientras que la versión distribuida con una partición en dos submodelos conectados de $N_1=30$ depósitos y $N_2=20$ depósitos respectivamente. Estos depósitos intercambian un flujo másico de agua entre sí, de forma reversible entre cada uno de ellos. Como punto de partida de la simulación, todos los depósitos se encuentran inicialmente vacíos. En cuanto a la partición de la simulación distribuida, ambas partes intercambian flujo y presión, de tal forma que el servidor OPC UA 1 escribe el valor de la presión en el servidor OPC UA 2, mientras que el servidor OPC UA 2 escribe el valor del caudal sobre el servidor OPC UA 1. El intercambio de flujo entre ambos servidores también es reversible. Como variable de entrada para la simulación, se define el nivel de entrada en el primer depósito del primer servidor OPC UA. Sobre esta variable, es posible llevar a cabo la introducción de perturbaciones, con el objetivo de verificar la influencia en el error que tienen las variaciones, y eliminar posibles estados estacionarios que reduzcan el error. Como variables de salida, se toman los niveles de cada uno de los depósitos y los caudales de salida. El modelo de simulación distribuida empleado puede observarse en la figura 2.

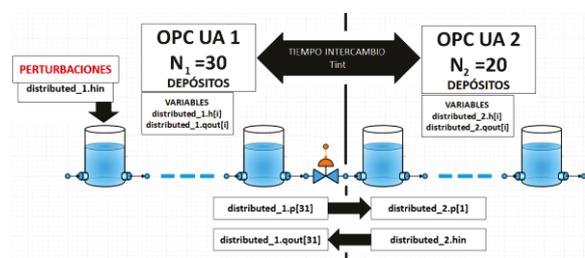


Figura 2: Modelo físico. Simulación distribuida y variables de intercambio.

La simulación monolítica se ejecuta de forma paralela a la distribuida [4]. El esquema de esta simulación puede verse en la figura 3.

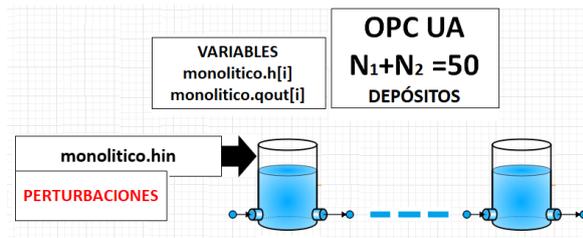


Figura 3: Modelo físico. Simulación monolítica

En este punto hay que señalar que la simulación de referencia no podrá ser usada en el caso real en el que se lleve a cabo la distribución de las simulaciones. No obstante, existen alternativas al empleo de una simulación monolítica. Por ejemplo, es posible tomar como error la diferencia entre las presiones y los flujos máscicos intercambiados, según muestran las ecuaciones 1 y 2. A falta de pruebas más exhaustivas, se observa una buena correlación entre el error así calculado y el error real.

$$E_1 = |P_2 - P_1| \quad (1)$$

$$E_2 = |Q_1 - Q_2| \quad (2)$$

Se toma como error crítico de decisión el máximo de ellos en el momento del intercambio de valores, según muestra la ecuación 3.

$$E_{CRITICO} = \max(E_1, E_2) \quad (3)$$

Cuando el error es cero, y las simulaciones distribuidas están perfectamente coordinadas, las presiones y caudales que se intercambian han de ser idénticas puesto que hacen referencia a la misma variable física. De igual manera, una diferencia mayor entre estas variables, es síntoma de que ha ocurrido un cambio dinámico importante, que provoca una falta de sincronía grande que debe ser corregida y corresponderá a un error también grande entre la simulación distribuida y la monolítica.

En el presente trabajo se calcula el error real utilizando la simulación monolítica, aunque se reconoce que en la práctica habrá que utilizar la alternativa descrita, que se calcula de forma autosuficiente, a partir de las variables intercambiadas en la simulación distribuida.

2.3 DESCRIPCIÓN DEL ALGORITMO PROGRAMADO

2.3.1 Objetivos

El principal objetivo desarrollado mediante el presente trabajo es la dotación a la herramienta de gestión de simulaciones distribuidas de una estrategia para la variación de la frecuencia de intercambio de

valores mediante comparación con una simulación de referencia.

Al disponer de una simulación de referencia dividida en una simulación distribuida, surge un error de simulación. Este error es directamente proporcional a la frecuencia de intercambio de valores en la frontera entre los servidores OPC UA que contienen la simulación distribuida, de tal forma que a menor frecuencia de intercambio el error aumenta. Por otra parte, a mayor frecuencia de intercambio, menor será la velocidad ejecución de la simulación completa. De esta forma, queda patente que existe un compromiso entre la exactitud lograble y la carga computacional a través de la elección de esa frecuencia de intercambio.

Mediante el empleo de la librería de control, es posible llevar a cabo la medición del error, por comparación entre el valor de referencia y el valor obtenido mediante el avance de la simulación. Partiendo de este error, se continúa con la estrategia de variación del tiempo de intercambio de manera similar a como se hace en los algoritmos de control de paso empleados habitualmente en simulación numérica. Como objetivo complementario, se precisa la facilidad para llevar a cabo la configuración y su interacción a través de los ficheros de configuración o el propio usuario.

2.3.2 Implementación

Para el desarrollo de la estrategia de adaptación de la frecuencia de intercambio de datos, se parte de la herramienta de gestión DSM-UA. Esta herramienta está constituida por varios módulos. Aparte de las librerías necesarias para la gestión de clientes OPC UA, se dispone de una librería para la gestión de datos, ficheros de configuración y resultados, denominada FILES_FUNCTIONS. Mediante los datos obtenidos, se realiza la medición del error y la adaptación de la frecuencia de intercambio de datos en la librería CONTROL_LIBRARY. El esquema del funcionamiento queda reflejado en la figura 4.

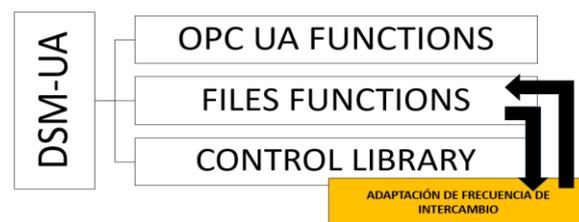


Figura 4: Implementación de la variación de la frecuencia de intercambio.

2.3.3 Funcionamiento

La primera etapa de para llevar a cabo la variación de la frecuencia de intercambio consiste en la configuración de las variables de intercambio. En esta etapa se puede definir el tiempo de intercambio, y si éste va a ser variable mediante la lectura de un fichero de configuración, o por interacción con el usuario.

Una vez realizada la configuración, comienza el avance de la simulación, de tal forma que, a medida que la simulación avanza en el tiempo, se va comparando de forma dinámica el valor simulado con el valor equivalente en la simulación distribuida. El avance de la simulación de referencia permite ver las posibles dinámicas de planta, que tienen una gran influencia en el error. El error, medido para la iteración k-ésima, corresponde al error normalizado por comparación del valor de referencia y el valor obtenido mediante la simulación, según muestra la expresión 4.

$$(4) \text{ERROR}_k = \frac{|Y_{\text{REFERENCIA}} - Y_{\text{DISTRIBUIDO}}|}{Y_{\text{REFERENCIA}}}$$

Para la modificación del tiempo de intercambio, en función del error de la iteración k-ésima, se sigue una secuencia de variación del tiempo de intercambio según recoge la expresión 5.

$$(5) \begin{cases} \text{Si } \text{ERROR}_k < \text{TOL}_{\min} \Rightarrow \text{Tint}_{k+1} = 2 \text{Tint}_k \\ \text{Si } \text{TOL}_{\min} \leq \text{ERROR}_k \leq \text{TOL}_{\max} \Rightarrow \text{Tint}_{k+1} = \text{Tint}_k \\ \text{Si } \text{ERROR}_k > \text{TOL}_{\max} \Rightarrow \text{Tint}_{k+1} = \frac{1}{2} \text{Tint}_k \end{cases}$$

Cuando se dispone del valor del error cometido para cada una de las variables de la simulación distribuida, se procede a la comparación con unos valores de tolerancia previamente establecidos. De esta forma, si el error se encuentra por debajo de la tolerancia mínima, se incrementa el valor de la frecuencia de intercambio, duplicando dicho valor. Si el valor del error es superior a la tolerancia máxima, se disminuye la frecuencia de intercambio a la mitad para el siguiente paso. En caso de que el error se encuentre dentro del rango de tolerancias establecido, la frecuencia de intercambio se mantiene para el siguiente paso. En todo caso, los valores obtenidos en el paso k se emplean en la siguiente iteración. El valor máximo de la frecuencia de intercambio se encuentra limitado, con el objetivo de no perder información ante un cambio brusco en las condiciones de la simulación.

Una posibilidad de variar esta estrategia sería la de añadir una variable de decisión a la hora de realizar la siguiente iteración, asumiendo como no aceptable

el error si se encuentra fuera de unos márgenes de tolerancia, de tal forma que se recalcula el paso actual hasta que este error sea aceptable. La ventaja de implementar esta posibilidad hace que la exactitud de la simulación aumente, a costa de un compromiso con la velocidad de ejecución. En la actualidad, esta opción para la gestión del error está en fase de desarrollo, basándose en la posibilidad de configuración que ofrece DSM-UA. Otra opción es asumir el error y esperar que se reduzca en promedio con el avance de la simulación. El algoritmo de ejecución puede verse en la figura 5.

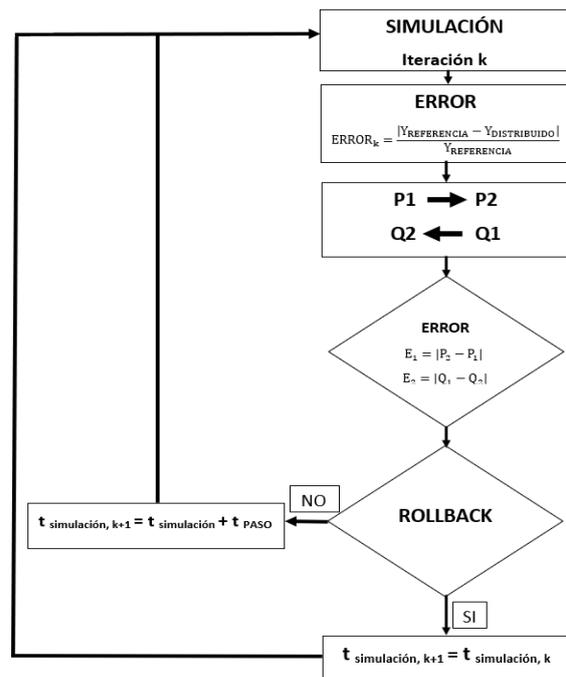


Figura 5: Algoritmo de simulación.

3 RESULTADOS

Para verificar la importancia de disponer de alguna forma de variar la frecuencia de intercambio de datos, se llevaron a cabo una serie de simulaciones del proceso para poder comparar los diferentes resultados obtenidos. El diagrama de bloques de la simulación empleada recoge la posibilidad de introducir perturbaciones en cada una de las variables de entrada, y su influencia en el cálculo del error. Este diagrama puede verse en la figura 6.

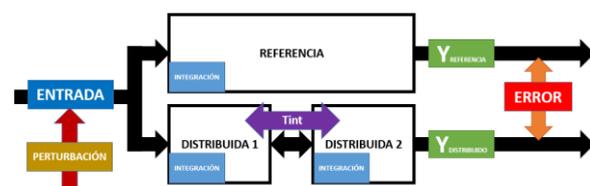


Figura 6: Diagrama de bloques del lazo de control empleado en la obtención de los resultados.

La simulación seguida corresponde a un tiempo de 600 unidades de tiempo, en el cual se introdujeron una serie de perturbaciones a la entrada, con el objetivo de estudiar el comportamiento del error ante diferentes regímenes dinámicos. Las perturbaciones introducidas para la variable de entrada del primer servidor distribuido, así como los valores numéricos de la variable de entrada quedan reflejadas en la tabla 1. Fuera de estos intervalos, el valor de entrada se encuentra prefijado por defecto a 0.7 m.

Tabla 1: Perturbaciones introducidas.

Intervalo	Valor (m) (%)
50-100	0.35 m (50%)
200-250	0.35 m (50%)
400-420	0.07 m (10%)
500-510	0.07 m (10%)

Las perturbaciones elegidas, así como el tiempo de simulación se escogieron de tal forma que pudiesen ser visibles diferentes dinámicas de planta.

Una vez que se establecieron las premisas del ejemplo de simulación que se iba a llevar a cabo, se estableció la configuración previa de las variables de la simulación distribuida mediante ficheros de texto.

Para el estudio del comportamiento del sistema simulado, y puesto que en el modelo se dispone de una simulación distribuida formada por dos servidores, se obtuvieron las representaciones gráficas del comportamiento dinámico de una variable del primer servidor y de una variable del segundo servidor. También se obtuvieron las representaciones gráficas de la evolución de la variación de la frecuencia de intercambio de datos en función del tiempo.

3.1 Comparación de simulación distribuida con simulación monolítica. Sin adaptación del tiempo de intercambio.

En la figura 7 se obtuvo la comparación de la simulación realizada para una variable de la simulación monolítica de referencia y su equivalente en la simulación distribuida.

La variable representada corresponde con el nivel del primer depósito del segundo servidor de la simulación distribuida, y su variable equivalente en la simulación monolítica. Para realizar esta simulación, no se realizó adaptación alguna del tiempo de intercambio. En esta simulación, el tiempo de intercambio se prefijó a 5 segundos.

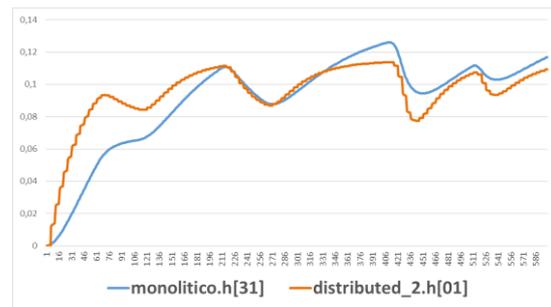


Figura 7: Comparación del nivel del primer depósito para el segundo servidor de la simulación distribuida y equivalente en simulación monolítica.

Para ilustrar el comportamiento del flujo intercambiado, en el mismo depósito de la simulación distribuida y su equivalente en la monolítica, se obtuvo la representación de la figura 8.

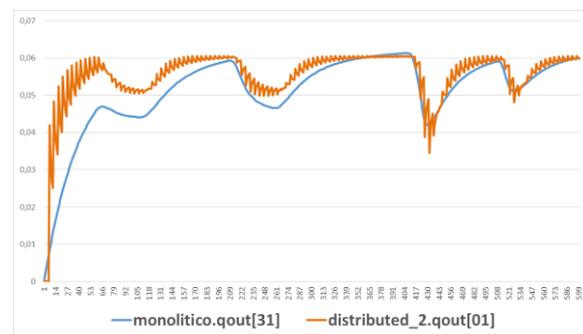


Figura 8: Variación del flujo másico intercambiado, primer depósito del segundo servidor de la simulación distribuida y equivalente monolítico.

3.2 Simulaciones con frecuencia de intercambio fija.

Para ilustrar la influencia del tiempo de intercambio en el ajuste de la simulación, se realizaron tres simulaciones diferentes, para Tint fijos de 2, 3 y 10 segundos, respectivamente. Se obtuvieron las representaciones del nivel del primer depósito del segundo servidor de la simulación distribuida, como se muestra en la figura 9.

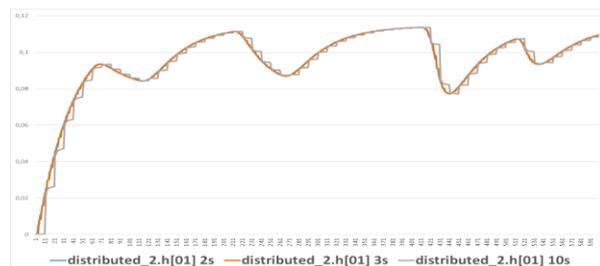


Figura 9: Nivel primer depósito del segundo servidor de la simulación distribuida, para Tint diferentes.

3.3 Simulación con tiempo de intercambio variable. Variables del segundo servidor.

Para mostrar la influencia de la frecuencia de intercambio en variables de entrada del segundo servidor, se obtuvieron representaciones de una misma variable, el nivel del primer depósito del segundo servidor de la simulación distribuida, y dos tiempos de intercambio inicial diferentes. Estos tiempos de intercambio iniciales se van modificando a medida que avanza la simulación. Para un tiempo de intercambio inicial de 2 segundos, se obtuvo la gráfica de la figura 10, y para un tiempo de intercambio inicial de 3 segundos, la gráfica reflejada en la figura 11.

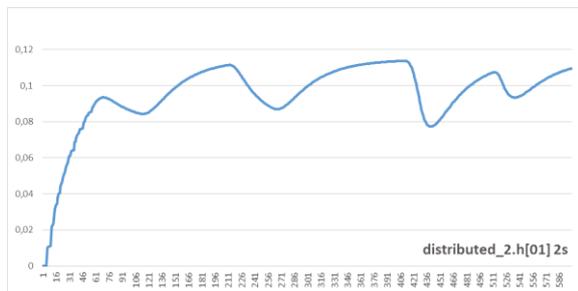


Figura 10: Comportamiento del nivel del primer depósito del segundo servidor para T_{int} inicial de 2 segundos.

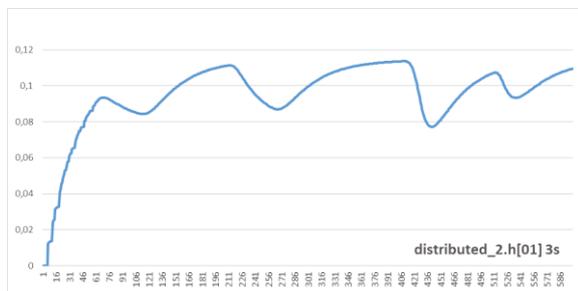


Figura 11: Comportamiento del nivel del primer depósito del segundo servidor para T_{int} inicial de 3 segundos.

Para ilustrar la influencia de la variación del tiempo de intercambio en la salida del segundo servidor, se eligió como variable a representar el caudal de salida del primer depósito del segundo servidor de la simulación distribuida, y se obtuvieron las representaciones gráficas de la figura 12, para un tiempo de intercambio inicial de 2 segundos y la de la figura 13, para un tiempo de intercambio inicial de 3 segundos, que se modifican en función del error a medida que avanza la simulación.

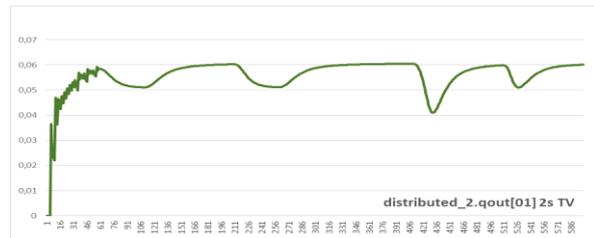


Figura 12: Comportamiento del caudal del primer depósito del segundo servidor distribuido para un tiempo de intercambio inicial de 2 segundos.

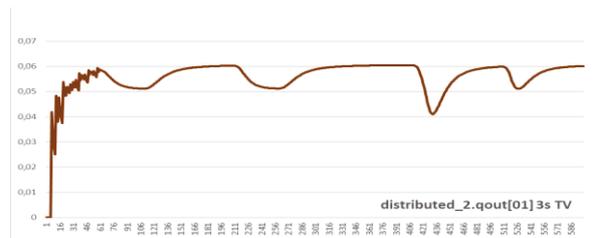


Figura 13: Comportamiento del caudal del primer depósito del segundo servidor distribuido para un tiempo de intercambio inicial de 3 segundos.

3.4 Comparativa de errores

A continuación, se muestra la representación gráfica del error unitario, medido según la expresión 4, por comparación entre los valores de la simulación monolítica y sus equivalentes de la distribuida. En la figura 14 se muestra el error cuando se emplea el algoritmo de variación, para tiempos de intercambio iniciales de 2 y 3 segundos, mientras que en la figura 15 se muestra el error sin empleo del algoritmo.

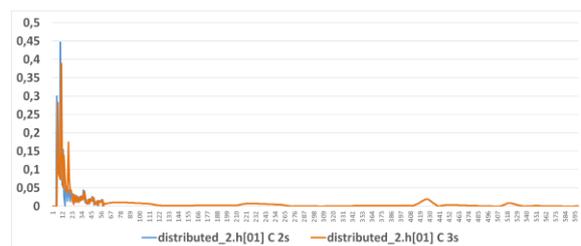


Figura 14: Error normalizado empleando librería de control.

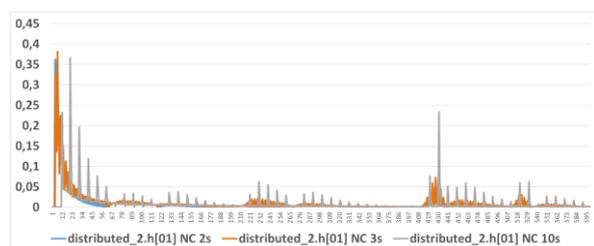


Figura 15: Error normalizado sin empleo de la librería de control.

4 DISCUSIÓN Y CONCLUSIONES

Hasta ahora se dispone de una herramienta de gestión para una simulación distribuida desarrollada íntegramente en librerías de PYTHON. Mediante esta herramienta, se interactúa con una simulación encapsulada en servidores OPC UA. La novedad radica en la existencia de una estrategia para la variación del tiempo de intercambio entre los diferentes servidores constituyentes de la simulación distribuida. Es posible llevar a cabo esta adaptación mediante comparación con una simulación monolítica en paralelo o utilizando una estrategia que solo dependa del valor de las variables intercambiadas en la simulación distribuida. Esta estrategia queda integrada dentro de la librería correspondiente dentro de la herramienta de gestión distribuida DSM-UA. Análogamente al desarrollo de DSM-UA, la implementación de la estrategia de adaptación se ha llevado a cabo mediante el empleo de PYTHON.

En cuanto al comportamiento dinámico de las variables representadas en la sección de resultados, puede observarse que para el primer servidor el comportamiento sólo se ve alterado por la introducción de las perturbaciones. Para las variables del segundo servidor, y mediante variación del tiempo de intercambio, se aprecia un comportamiento dinámico similar para ambos intervalos iniciales, si bien se observa que un aumento del tiempo de intercambio inicial produce un aumento del error en los primeros instantes de la simulación. La adaptación del tiempo de intercambio provoca una reducción del error, manteniendo un tiempo de intercambio pequeño durante la introducción de las perturbaciones, independientemente del tiempo de intercambio inicial seleccionado. El efecto se agudiza si se observan las representaciones gráficas de la variable de salida del segundo servidor.

Para el valor del error normalizado, puede observarse cómo el error es generalmente mayor cuando se comienza con un tiempo de intercambio superior respecto a un valor más pequeño. El comportamiento del error se va estabilizando a medida que avanza la simulación, observándose pequeños aumentos cuando se introducen nuevas perturbaciones. Es posible observar también cómo el error es superior cuando se emplea un tiempo de intercambio fijo. En caso de emplear un tiempo de intercambio de tamaño fijo, se observa la misma tendencia en la evolución temporal del error.

Para futuras líneas de investigación y trabajo, existen múltiples posibilidades de mejora. En primer lugar, la investigación sobre cómo proceder a una eliminación de la simulación monolítica, adoptando una solución mucho más realista a nivel industrial. El empleo de la

simulación monolítica para el desarrollo del presente trabajo se basa en la importancia de disponer de un modelo que sirva de referencia para realizar pruebas sobre el programa desarrollado, y prescindir de su uso en futuros trabajos.

Otras líneas de investigación radican en la adopción de diferentes posibilidades de configuración sobre los procedimientos a seguir cuando el error es mayor que el admitido. Un procedimiento posible sería reducir la frecuencia de intercambio, pero continuar con la simulación en el instante actual, esperando que el error se reduzca en el siguiente paso de simulación (asumiendo un mayor error puntual, pero manteniendo el promedio del error a lo largo de toda la simulación dentro de unos límites aceptables). La otra alternativa, más exigente en cuanto a la precisión, sería llevar a cabo un retroceso en el tiempo simulado, y relanzar la simulación con una frecuencia de intercambio inferior.

Agradecimientos

Este trabajo ha sido financiado con ayuda del Ministerio de Ciencia e Innovación, con ref. PGC2018-099312-B-C31 y cofinanciado por la Unión Europea a través del FEDER. El segundo y tercer autor agradecen el apoyo de la Junta de Castilla y León y EU/FEDER (CLU 2017-09, CL-EI-2021-07).

English summary

DYNAMIC ADAPTATION OF DATA EXCHANGE FREQUENCY IN A DISTRIBUTED SIMULATION

Abstract

This article describes a distributed dynamic simulation manager, using PYTHON programming, that uses the OPC UA interoperability standard to communicate with the different servers that make up a distributed simulation, acting as a configurable client. OPC UA servers are implemented using an external simulation and process modeling tool. For this tool to function according to the simulation needs, it must have some form of control over the frequency of data exchange as the process evolves.

This contribution describes the development of a strategy for adapting the information exchange frequency between the different servers by comparing it with a reference simulation developed in parallel or by obtaining a previous reference. Through this strategy, the configuration of the simulation conditions, the measurement of the error and the

adaptation of the exchange time are carried out, in a similar way to the step control algorithms traditionally used by some dynamic integration algorithms.

Keywords: simulation, OPC UA, exchange, adaptation, error.

Referencias

- [1] Alves R., De Prada C., Normey-Rico J., Merino A. y Acebes L.F. (2005) "OPC based distributed real time simulation of complex continuous processes". *Simulation Modelling Practice and Theory* 13. pp. 525–549.
- [2] Cellier, F.E., Kofman, E., (2006). "Continuous system simulation". <https://doi.org/10.1007/0-387-30260-3>
- [3] E.A. Empresarios Agrupados, EcosimPro, <http://www.ecosimpro.com> (acceso 3 de mayo de 2022).
- [4] Fujimoto R. M. (2001). "Parallel and Distributed Simulation Systems", *Proceeding of the 2001 Winter Simulation Conference*.
- [5] Merino, A.; Acebes, L. F.; Mazaeda, R.; De Prada, C. (2009) "Modelado y Simulación del proceso de producción del azúcar". *Revista Iberoamericana de Automática e Informática Industrial*. pp. 21-31.
- [6] OPC Foundation, Home Page, <http://www.opcfoundation.org/> (acceso 4 de mayo de 2022).
- [7] 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.
- [8] Zamarreño, J.M. (2010) "Acceso a datos mediante OPC". Editorial Andavira SL, Santiago de Compostela.
- [9] Zamarreño J. M., Mazaeda R., Caminero J. A., Rivero A. J., Arroyo J. C. (2014) "A new plug-in for the creation of OPC servers based on EcosimPro© simulation software". *Simulation Modelling Practice and Theory*. pp. 86-94



© 2022 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-ncsa/4.0/deed.es>).