

## Modelado multiescala de un esterilizador mediante co-simulación

Merino, A.<sup>a,\*</sup>, Vilas, C.<sup>b</sup>, García-Álvarez, D.<sup>c</sup>, Galán-Casado S.<sup>d</sup>, Ruiz-Gonzalez, R.<sup>a</sup>

<sup>a</sup>Departamento de Digitalización, Universidad de Burgos, Avda. Cantabria s/n, 09006, Burgos, España.

<sup>b</sup>Grupo de Biosistemas e Ingeniería de Bioprocesos, IIM-CSIC, C/Eduardo Cabello, 6, 36208, Vigo, España.

<sup>c</sup>Departamento de Informática, Universidad de Valladolid, P.º de Belén, 15, 47011 Valladolid, España.

<sup>d</sup>Departamento de Ingeniería Química, Universidad Politécnica de Madrid, Madrid, España.

**To cite this article:** Merino, A., Vilas, C., García-Álvarez, D., Galán-Casado S. Ruiz-Gonzalez, R. 2025. Multiscale modeling of a sterilizer using co-simulation. Actas del I Simposio CEA de los GT: Ingeniería de Control - Modelado, Simulación y Optimización - Educación en Automática. Sevilla, España.

### Resumen

Este artículo presenta una prueba de concepto sobre el uso de la co-simulación para integrar modelos desarrollados con distintas herramientas y tipos de modelos mediante el estándar FMI (*Functional Mock-up Interface*). El estudio se aplica al modelado de un esterilizador compuesto por tres subsistemas. A nivel macroscópico, el autoclave, sus equipos auxiliares y sistemas de control se han modelado usando primeros principios y un enfoque orientado a objetos. Por su parte, las latas en el interior del autoclave incluyen dos modelos: el líquido, representado con un modelo de Aprendizaje Automático, y el sólido, modelado con ecuaciones diferenciales parciales simplificadas mediante *Proper Orthogonal Decomposition* (POD). El artículo detalla los modelos de los subsistemas y su conexión a través de un algoritmo maestro que gestiona la comunicación y sincronización. El análisis de las principales variables del proceso, muestra un comportamiento adecuado de las mismas, lo que permite validar el esquema propuesto.

**Palabras clave:** Co-simulación, Functional Mock-up Interface (FMI), Modelado multiescala, Descomposición Ortogonal Propia (POD), Redes Neuronales de memoria a largo y corto plazo (LSTM), Modelado Orientado a Objetos

### Multiscale modeling of a sterilizer using co-simulation

#### Abstract

This article presents a proof of concept for the use of co-simulation to integrate models developed with different tools and types of models through the FMI (*Functional Mock-up Interface*) standard. The study focuses on modeling a sterilizer composed of three subsystems. At the macroscopic level, the autoclave, its auxiliary equipment, and control systems were modeled using first principles and an object-oriented approach. Meanwhile, the cans inside the autoclave are represented by two models: the liquid, modeled using a Machine Learning Model, and the solid, described by partial differential equations simplified through Proper Orthogonal Decomposition (POD). The article details the subsystem models and their integration via a master algorithm that manages communication and synchronization. The analysis of the main variables of the process shows an adequate behavior of these, which allows validating the proposed scheme.

**Keywords:** Co-simulation, Functional Mock-up Interface (FMI), multi-scale modelling, Proper Orthogonal Decomposition (POD), Long Short-Term Memory (LSTM) Neural Networks, Object Oriented Modeling

### 1. Introducción

La creciente complejidad de los sistemas de ingeniería y la necesidad de analizar su comportamiento desde un punto de vista multidisciplinar y con distintos niveles de detalle, ha im-

pulsado el desarrollo de nuevas estrategias de simulación. Uno de estos enfoques, la co-simulación (Schweiger et al., 2019) ha emergido como una metodología que permite la integración de múltiples modelos especializados, desarrollados en distin-

\*Autor para correspondencia: alejandromg@ubu.es

tas plataformas y con diferentes paradigmas de modelado. Esta aproximación se basa en la coordinación de simulaciones independientes para trabajar de manera conjunta, superando las limitaciones de los enfoques de simulación monolíticos y facilitando el análisis de sistemas heterogéneos y distribuidos.

El modelado multiescala es una metodología que integra procesos y fenómenos en diferentes escalas temporales o espaciales para analizar sistemas complejos de manera integral (Ruscone et al., 2024). Los modelos multiescala ofrecen varias ventajas significativas, particularmente en la mejora del rendimiento y la eficiencia en diversas aplicaciones. Estos modelos permiten la integración de componentes que operan a diferentes frecuencias, permitiendo por ejemplo, el análisis y diseño de sistemas de control que abarcan desde escalas macroscópicas hasta moleculares, facilitando la comunicación dinámica entre diferentes niveles de modelado (Paulson et al., 2016). Además, la división en subsistemas con pasos de tiempo adaptados, permiten mejoras significativas en la velocidad de simulación y la gestión de recursos, particularmente el caso de modelos grandes, como el que se describe en (Li et al., 2020).

Actualmente se considera que existen dos estándares para co-simulación: HLA (*High Level Architecture*) (IEEE Standards Association, 2010), principalmente utilizado para la co-simulación de sistemas de eventos discretos, y FMI (Modelica Association, 2022), para sistemas continuos que pueden presentar eventos discretos. Para el caso de estudio de este artículo, dado que se trata de un modelo continuo, se trabajará con el estándar FMI.

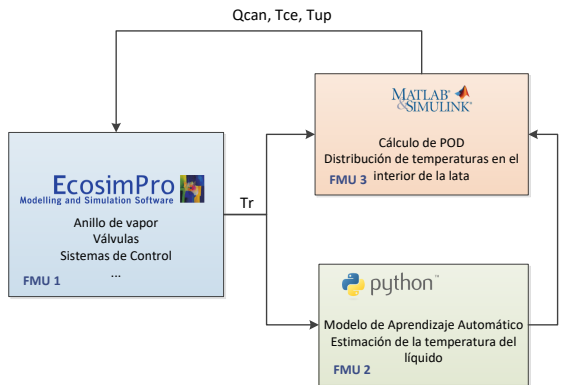


Figura 1: Esquema de co-simulación

En este trabajo se presenta una unidad de esterilización de conservas de atún que utiliza agua sobrecalentada para aportar calor al proceso (Pitarch et al., 2021). Para este sistema, los autores disponen de modelos matemáticos desarrollados en distintas herramientas y con distintos paradigmas de modelado, que representan sistemas en distintas escalas y con necesidades de integración en pasos diferentes, que será necesario integrar para simular el conjunto de manera coordinada. Se trata por tanto de un sistema multiescala en el que cada uno de los modelos disponibles representa un nivel de detalle diferente de las distintas partes del sistema de estudio. En concreto, se ha empleado un modelo de Aprendizaje Automático (ML, *Machine*

*Learning*) basado en redes neuronales para estimar la temperatura del líquido de relleno en las latas a esterilizar, desarrollado en Python (Python Core Team, 2019) con la librería Pytorch (Paszke et al., 2019), complementado con un modelo que calcula la temperatura en el interior de la parte sólida del interior de la lata (pastilla de atún), desarrollado en Matlab®/Simulink® (Inc., 2023), así como con un modelo completo del proceso de esterilización creado en EcosimPro® (EA Internacional, 2024). Para el caso de estudio, se integraron los modelos en formato FMUs (*Functional Mock-Up Units*) y se desarrolló un coordinador (máster) encargado de comunicar y sincronizar los distintos módulos (Figura 1).

## 2. Descripción del caso de estudio

Como se comentó en la introducción el caso de estudio presentado en este artículo es el de una unidad de esterilización de conservas de atún mediante agua sobrecalentada con una estructura similar a la mostrada en la Figura 2. El objetivo de la esterilización de las latas es inactivar los microorganismos potencialmente nocivos mediante calor. En este proceso, el vapor producido en una planta de cogeneración se utiliza para calentar agua en un intercambiador de calor de placas. El agua sobrecalentada se introduce en el esterilizador mediante pulverización, distribuyéndose sobre las latas. El agua se extrae del fondo del esterilizador y se recircula al intercambiador.

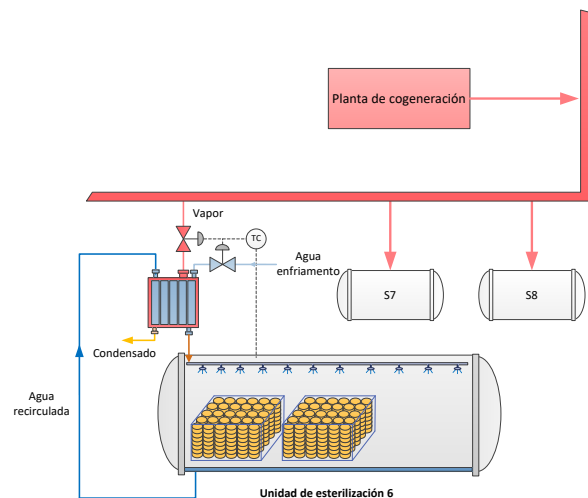


Figura 2: Esquema de un esterilizador

### 2.1. Descripción general del ciclo de esterilización

El proceso comienza con la carga del esterilizador en el que se introducen de manera manual los carros en los que están apiladas las latas de conserva. Una vez introducidos los carros en el esterilizador, este se cierra y comienza el proceso de esterilización que consta de tres etapas: calentamiento, mantenimiento y enfriamiento. En la etapa de calentamiento se aporta calor al sistema a través del agua sobrecalentada pulverizada sobre las latas, de manera que la temperatura en el interior del esterilizador aumenta en forma de rampa hasta alcanzar la temperatura

de mantenimiento. En ese momento se inicia la etapa de mantenimiento en la que se mantiene constante la temperatura en el interior del esterilizador durante un tiempo determinado en el que se asegura la desactivación de los microorganismos. Finalmente, se enfría el esterilizador mediante la pulverización de agua fría. El sistema está controlado mediante un Controlador Lógico Programable (PLC, *Logic Programmable Controller*) que gestiona las transiciones entre etapas y el seguimiento de las diferentes consignas (Rivero-Contreras et al., 2024).

### 3. Modelos matemáticos

#### 3.1. Modelo de la fase sólida de las latas

La transmisión de calor en la parte sólida del interior de la lata (atún) se puede modelar mediante la ecuación de Fourier. Partiendo de la asunción de que la temperatura del autoclave es homogénea, el sistema 3D representado en la Figura 6 se puede reducir, sin pérdida de información, a un sistema 2D rectangular en coordenadas cilíndricas:

$$\frac{\partial T_a}{\partial t} = \alpha_p \left[ \frac{\partial^2 T_a}{\partial z^2} + \frac{1}{r} \frac{\partial}{\partial r} \left( r \frac{\partial T_p}{\partial r} \right) \right] \quad (1)$$

donde  $r \in [0, R]$  es el radio de la lata y  $z \in [0, L]$  es la altura de la pastilla de atún.  $T_a(r, z, t)$  denota la temperatura en todos los puntos del atún. El parámetro  $\alpha_p$  representa la difusividad térmica. En la frontera  $r = 0$  se utiliza una condición de simetría:

$$\left. \frac{\partial T_a}{\partial r} \right|_{r=0} = 0. \quad (2)$$

En las otras tres fronteras ( $r = R$ ,  $z = 0$ , y  $z = L$ ) se utilizan condiciones de Robin:

$$\begin{aligned} \alpha_p \left. \frac{\partial T_a}{\partial z} \right|_{z=L} &= h_t (T_l - T_a|_{z=L}); \\ \alpha_p \left. \frac{\partial T_a}{\partial z} \right|_{z=0} &= h_b (T_R - T_a|_{z=0}); \\ \alpha_p \left. \frac{\partial T_a}{\partial z} \right|_{r=R} &= h_r (T_R - T_a|_{r=R}); \end{aligned} \quad (3)$$

donde,  $T_R$  y  $T_l$  son la temperatura del autoclave y del líquido de relleno, respectivamente. Los parámetros  $\bar{h}_t$ ,  $\bar{h}_b$  y  $\bar{h}_r$  representan los coeficientes de transferencia entre el sólido y el medio en contacto con el mismo. Aún reduciendo el problema 3D a uno 2D, la aplicación de métodos clásicos para resolver ecuaciones diferenciales parciales, como el método de los elementos finitos, resulta en número elevado de ecuaciones, lo que lo convierte en ineficiente para tareas en tiempo real. Por lo tanto, en este trabajo se utiliza el método de descomposición ortogonal propia (POD, por sus siglas en inglés) para obtener un modelo de orden reducido (Sirovich, 1987). En este método, se utilizan series de Fourier para aproximar la solución de la PDE:

$$T_a(r, z, t) = \sum_{i=1}^{\infty} \phi_i(r, z) \mathcal{T}_a(t) \approx \sum_{i=1}^m \phi_i(r, z) \mathcal{T}_a(t) \quad (4)$$

Las autofunciones ( $\phi_i(r, z)$ ) se obtienen resolviendo el siguiente problema de valor propio:

$$\phi_i = \lambda_i \int_{\mathcal{V}} \mathcal{K} \phi_i d\mathcal{V}, \quad \mathcal{K} = \lim_{t \rightarrow \infty} \frac{1}{t} \int_0^t T_a T_a' dt, \quad (5)$$

donde  $T_a$  son soluciones obtenidas experimentalmente o mediante otro método numérico. Al proyectar la ecuación (1) sobre las  $m$  primeras autofunciones se llega a un sistema de unas pocas ecuaciones diferenciales ordinarias (en este caso, 10):

$$\frac{d\mathcal{T}_a}{dt} = \mathcal{A} \mathcal{T}_a + \mathcal{B} f(T_R, T_l) \quad (6)$$

Donde  $\mathcal{A} \in \mathcal{M}^{m \times m}$  y  $\mathcal{B} \in \mathcal{M}^{m \times 1}$  son, respectivamente, las proyecciones del operador laplaciano y de las condiciones frontera sobre las funciones base.

El modelo descrito se ha implementado en Simulink (ver Figura 3), ya que FMI solo está disponible para modelos desarrollados en Simulink.

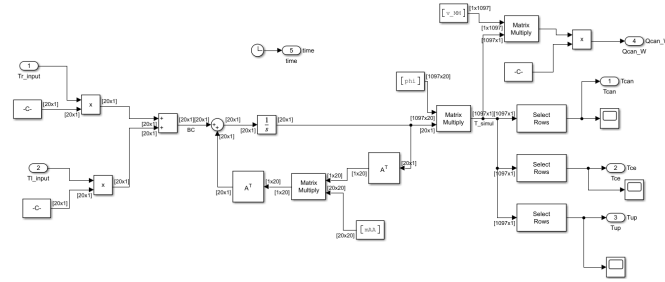


Figura 3: Modelo de POD desarrollado en Simulink

#### 3.2. Modelo de la temperatura en el líquido de relleno

Este modelo trata de representar el comportamiento del líquido contenido dentro de la lata. Se puede desarrollar un modelo basado en primeros principios para este proceso. Sin embargo, dicho modelo no es conocido y resultaría complejo de desarrollar y validar. Por lo tanto, se ha optado por desarrollar un modelo de Aprendizaje Automático basado en datos. Al tratarse de un sistema dinámico en el que los estados actuales dependen de estados anteriores, se eligió usar las redes LSTM (*Long Short-Term Memory*) al haber demostrado su eficacia como buenos regresores de series temporales. Concretamente, se ha modelado mediante una arquitectura *Encoder-Decoder* basada en capas LSTM, desarrollada en Pytorch®.

Para disponer de suficientes datos para el entrenamiento de la red, es necesario registrar valores de la temperatura del líquido  $T_{liq}$  y la temperatura de la retorta  $T_r$  en distintos puntos de operación, para que el modelo capture las distintas dinámicas existentes en el proceso. En esta fase preliminar no se dispone de datos reales de planta, por lo que se ha generado un conjunto de datos sintéticos para la temperatura del líquido mediante un modelo de primer orden, ecuación 7, en distintos puntos de operación en condiciones similares a las que aparecerían en el proceso real.

$$\frac{d(T_{liq})}{dt} = \frac{1}{\tau} T_r \quad (7)$$

Se trata de una solución muy aproximada pero que nos permite validar la metodología presentada en el artículo. Una vez se disponga de los datos reales del proceso, se re-entrenará el modelo con los nuevos datos.

El modelo de Aprendizaje Automático consiste en una configuración *Encoder-Decoder*. El *Encoder* contiene una capa interna formada por 10 celdas LSTM. Éste recibe como entrada la serie temporal de la temperatura de la retorta  $T_r$ . El estado

interno de la capa LSTM del *Encoder*, actúa como entrada del estado interno de la capa LSTM del *Decoder*, también formado por una única capa con 10 celdas LSTM y cuya entrada es el último valor de la serie temporal que se aplicó como entrada al *Encoder*, de esta forma se consigue que la predicción del modelo sea de un único instante de tiempo (ver Figura 4). Los hiperparámetros utilizados en este modelo han sido; el número de celdas de cada capa (10), la tasa de aprendizaje (0.001) (*Learning Rate*) y el tamaño de la secuencia ( $w$ ) (100).

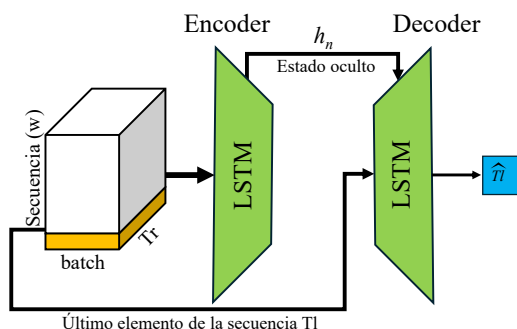


Figura 4: Arquitectura Encoder-Decoder del modelo de Aprendizaje Automático

Partiendo por tanto de los datos sintéticos, se ha entrenado el modelo, incluyendo un tamaño de secuencia de 100 valores pasados de temperatura del líquido. Después de entrenamiento y validación cruzada se obtienen muy buenos resultados presentando un valor del MSE (*Mean Squared Error*) de 0.1072 para la temperatura del líquido. La comparativa entre los valores experimentales y de simulación con el modelo de aprendizaje automático se muestran en la Figura 5, en la que puede apreciarse cómo temperatura predicha y la real se solapan de manera casi perfecta.

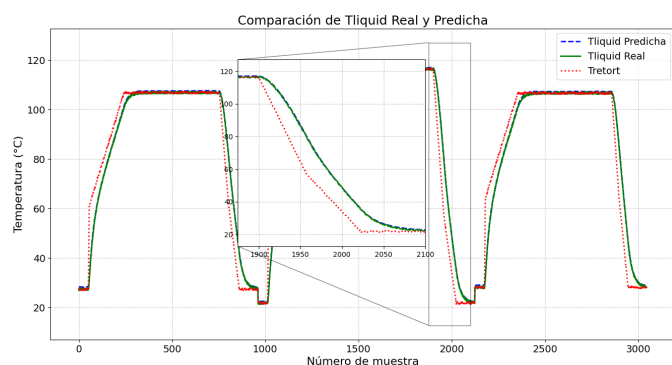


Figura 5: Resultados del ajuste de la temperatura del líquido con el modelo de Aprendizaje Automático

El modelo resultante se ha encapsulado en un módulo FMU utilizando la librería *pythonfmu* (Smith and contributors, 2024), que permite la encapsulación de código Python en FMUs siguiendo los estándares de la versión 2.0 y 3.0 de FMI.

### 3.3. Modelo del esterilizador

Este modelo abarca los componentes macroscópicos del sistema, incluyendo el equipo de esterilización, los equipos auxiliares y el sistema de control. En particular, se han modelado el

esterilizador, el intercambiador de calor, las válvulas de control, la generación de carros y el anillo de vapor. Para este tipo de sistemas resulta conveniente utilizar modelos basados en primeros principios con orientación a objetos, debido a características como la herencia y la agregación de componentes en topologías definidas por el usuario.

Como se mencionó en la introducción, en este caso se ha utilizado EcosimPro® como herramienta de modelado y simulación. En esta herramienta se han desarrollado los siguientes componentes.

- **Generador de carros:** componente que genera los carros con el producto enlatado seleccionado a una velocidad configurable por el usuario.
- **Anillo de vapor:** este componente se encarga de distribuir el vapor a los distintos esterilizadores. Se trata de un modelo globalizado sencillo, que opera como un acumulador recibiendo un determinado caudal de vapor fijo y calculando la presión del sistema mediante la ecuación de estado del gas ideal y la temperatura del mismo suponiendo vapor saturado. En este caso, en el que se ha simulado únicamente un esterilizador, el anillo simplemente se comporta como una fuente de vapor.
- **Válvula de vapor:** Calcula el caudal de vapor que entra en el intercambiador de calor en base a las ecuaciones típicas de flujo de fluidos en válvulas de gas.
- **Intercambiador de calor:** Modelo matemático de un intercambiador formado por 90 placas metálicas dispuestas en forma de canales paralelos en los que se alterna una corriente caliente (vapor) y una corriente fría (líquido). El modelo consiste en dos ecuaciones diferenciales ordinarias que describen la evolución de la temperatura en las placas con vapor y líquido, respectivamente. En la última fase de la esterilización (enfriamiento) se cierra la válvula de vapor y por las placas correspondientes se hace pasar una corriente de agua fría para enfriar rápidamente el autoclave de forma que no se siga degradando la calidad del producto.
- **Autoclave:** Incluye el modelo del autoclave en la que están contenidas las latas que se modelan en las FMUs descritas con anterioridad. Se asume que la temperatura en todo el autoclave es homogénea por lo que el modelo consiste en una ecuación diferencial ordinaria que describe la temperatura dentro del autoclave.

- **Sistema de control:** Se ha incluido un controlador de tipo Proporcional Integral (PI) que se encarga de mantener el valor de la temperatura en la retorta en el valor de consigna en las etapas de calentamiento y mantenimiento. Existen también elementos que permiten la conmutación entre las etapas del proceso y la lógica del proceso.

Dado que EcosimPro® es un lenguaje de programación orientado a objetos, se han definido cada uno de los componentes descritos de manera individual y se han conectado a través de puertos para construir el modelo completo del sistema. La extensión de los componentes descritos someramente con anterioridad es elevada, por lo que no se incluyen en detalle en este

trabajo. El lector interesado en estos modelos puede consultar Pitarch et al. (2021).

### 3.4. Variables intercambiadas

En la Figura 6 se muestra un esbozo de una lata de conserva, indicando las variables que se han intercambiado entre los distintos bloques de co-simulación. Desde el modelo del esterilizador, FMU 1, (se puede ver la identificación de los diferentes FMUs en la Figura 1), se envía la temperatura en el interior del esterilizador, conocida como retorta  $T_r$ , tanto al modelo de la fase sólida, FMU 3, como al modelo de la fase líquida FMU 2. Desde la fase sólida se envían el calor absorbido por las latas  $Q_{can}$ , la temperatura en el centro de la lata  $T_{ce}$  y la temperatura en la interfaz de la lata con la retorta  $T_{up}$ . Desde la fase líquida se envía la temperatura del líquido  $T_l$  hacia la fase sólida.

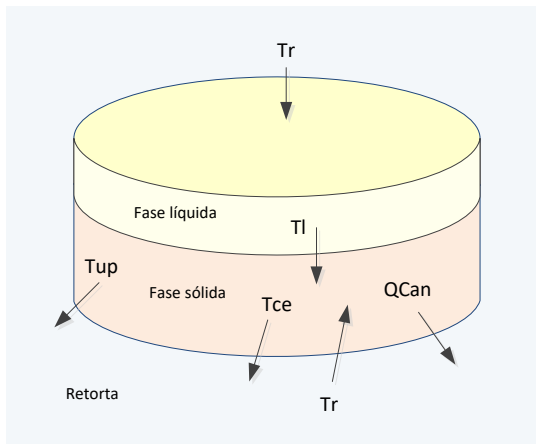


Figura 6: Esquema de una lata con sus fases sólida y líquida

### 3.5. Algoritmo máster

En la estrategia de co-simulación es necesario que exista un coordinador que ejecute, coordine y ordene los intercambios de datos entre los distintos módulos FMU que van a formar parte del modelo conjunto. En este caso, el coordinador se implementado en un programa escrito en Python que dirige la co-simulación haciendo uso de la librería fmpy (Dassault Systèmes, 2024).

En la Figura 7 puede observarse el bucle de co-simulación que se ejecuta en el máster, que sigue un esquema en serie o Gauss-Siedel no iterativo (Gomes et al., 2018). Se comienza ejecutando un paso de integración del modelo del esterilizador “doStep” y la temperatura de la retorta  $T_r$  obtenida se envía al resto de FMUs. A continuación, se ejecuta un paso de integración de la FMU 2 (fase líquida) y se envía la temperatura del líquido calculada  $T_l$  al resto de FMUs. Finalmente se ejecuta un paso de integración del modelo de la FMU 3 (fase sólida) y se envía la temperatura del centro de la lata  $T_{ce}$ , la temperatura en la interfaz de la lata con la retorta  $T_{up}$  y el calor absorbido por las latas  $Q_{can}$  a la FMU 1, esterilizador.

En este caso de estudio, la mayoría de los fenómenos modelados presentan una dinámica lenta. Tras diversas pruebas, se determinó que un macro-paso de 100 s es adecuado para capturar con precisión el comportamiento del proceso global. No obstante, la dinámica del líquido es significativamente más

rápida, lo que requiere un muestreo más frecuente para garantizar la exactitud de las predicciones realizadas por el modelo de Aprendizaje Automático. Por esta razón, se entrenó la red utilizando un intervalo de tiempo de 5 s. Como resultado, el modelo del líquido debe ejecutarse con un macro-paso reducido, aunque los valores intercambiados con las demás FMUs solo se sincronizan al final del macro-paso global de 100 s.

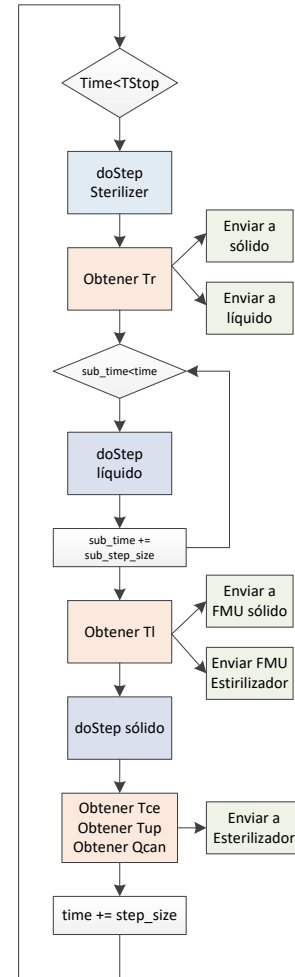


Figura 7: Bucle de co-simulación

## 4. Resultados

Para evaluar el desempeño del sistema se han simulado 7000 segundos que se corresponde aproximadamente con la duración de un ciclo de esterilización completo. En la Figura 8 se muestra el resultado de esta simulación en la que se puede ver cómo el sistema se comporta de manera adecuada, siguiendo el perfil típico de temperaturas propio de los ciclos de esterilización. Los puntos discontinuos azules, rojos y verdes representan, respectivamente, la temperatura del autoclave  $T_r$ , y las temperaturas en la superficie  $T_{up}$  y en el centro  $T_{ce}$  de la pastilla de atún. La línea naranja representa la temperatura del líquido  $T_{liq}$ .

Las temperaturas iniciales en los tres subsistemas se establecen en 20 °C, comenzando con la etapa de calentamiento, que se prolongará hasta alcanzar la temperatura de consigna de

121 °C. Una vez alcanzada, esta temperatura se mantiene durante 2 420 segundos, el tiempo necesario para inactivar los microorganismos y garantizar que la temperatura en el centro de la lata (representada por la línea verde) llegue al nivel requerido. Finalizada esta etapa, se introduce agua fría en el esterilizador, lo que provoca un descenso rápido de la temperatura del sistema.

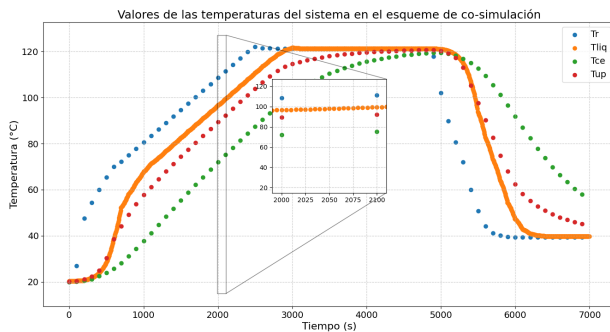


Figura 8: Temperaturas en el interior del esterilizador

El tiempo total de ejecución del sistema es de 21.5 segundos en un equipo portátil Intel® Core™ i7-1195G7 con 16 GB de RAM.

En este caso, no se cuenta con un modelo monolítico equivalente para comparar la precisión del modelo. Sin embargo, los resultados obtenidos muestran coherencia tanto con datos experimentales de sistemas similares, como con los generados por modelos desarrollados previamente por los autores de este trabajo.

## 5. Conclusiones

En este artículo se ha demostrado la viabilidad de la utilización de técnicas de co-simulación para modelar y simular un sistema complejo de esterilización de latas de conserva. Los resultados obtenidos muestran que la co-simulación permite integrar modelos desarrollados en diferentes herramientas y con distintos paradigmas de modelado, logrando una representación adecuada del comportamiento del sistema. El sistema ha sido coordinado mediante un algoritmo maestro en Python ha permitido coordinar y sincronizar los distintos módulos de co-simulación.

Como trabajo futuro, se propone incorporar datos de temperatura del líquido en el interior de las latas, obtenidos mediante sensores, y re-entrenar el modelo de aprendizaje automático. Esto permitirá desarrollar un sistema más realista. Otro objetivo será la optimización de los macro-pasos en los modelos de simulación, ajustando su tamaño para maximizar la velocidad de ejecución sin comprometer la precisión. Esto implicará definir métricas específicas para evaluar los límites aceptables de error en función de los requerimientos del sistema. Finalmente, se propone explorar e incorporar al trabajo algunas de las posibilidades que incorpora FMI 3.0, especialmente su capacidad para gestionar discontinuidades y eventos, aspectos críticos en sistemas dinámicos complejos. Aunque estas funciones aún no están plenamente implementadas en muchas herramientas, se consideran esenciales para mejorar la eficiencia y reducir el error en las simulaciones.

## Agradecimientos

El artículo forma parte de los proyectos: ‘Gestión óptima en tiempo real bajo incertidumbre para gemelos digitales (OptiDit)’, PID2021-123654OB-C33, ‘Modelos basados en datos y actualización de modelos para gemelos digitales’, PID2021-123654OB-C32, y ‘Aprendizaje Avanzado para la Mejora de la Productividad en Factorías Inteligentes’, PID2021-126659OB-I00, financiados por el MCIN y por la Unión Europea ‘FEDER’

## Referencias

- Dassault Systèmes, 2024. FMPy: A Python Library to Simulate Functional Mock-up Units (FMUs). Version 0.3.21.  
URL: <https://github.com/CATIA-Systems/FMPy>
- EA Internacional, 2024. Ecosimpro. Software de simulación y modelado.  
URL: <https://www.ecosimpro.com/>
- Gomes, C., Thule, C., Larsen, P. G., Denil, J., Vangheluwe, H., 2018. Co-simulation of continuous systems: A tutorial.  
URL: <https://arxiv.org/abs/1809.08463>
- IEEE Standards Association, 2010. IEEE Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) – Framework and Rules. No. IEEE Std 1516-2010. Accessed: 2024-12-02.  
URL: <https://standards.ieee.org/standard/1516-2010.html>
- Inc., T. M., 2023. Matlab version: 23.02.0 (r2023b).  
URL: <https://www.mathworks.com>
- Li, Y., Shu, D., Shi, F., Yan, Z., Zhu, Y., Tai, N., 2020. A multi-rate co-simulation of combined phasor-domain and time-domain models for large-scale wind farms. IEEE Transactions on Energy Conversion.  
DOI: 10.1109/TEC.2019.2936574
- Modelica Association, 2022. Functional Mock-up Interface 3.0: New Features Enabling New Applications. Version 3.0.  
URL: <https://fmi-standard.org/docs/3.0/>
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32. Curran Associates, Inc., pp. 8024–8035.
- Paulson, J., Ramadesigan, V., Subramanian, V., Braatz, R., 07 2016. Control systems analysis and design of multiscale simulation models. pp. 3083–3085.  
DOI: 10.1109/ACC.2016.7525388
- Pitarch, J., Vilas, C., de Prada, C., Palacín, C., Alonso, A., 2021. Optimal operation of thermal processing of canned tuna under product variability. Journal of Food Engineering 304, Article ID 110594.  
DOI: 10.1016/j.jfoodeng.2021.110594
- Python Core Team, 2019. Python: A dynamic, open source programming language. Python Software Foundation, python version 3.7.  
URL: <https://www.python.org/>
- Rivero-Contreras, R. E., Merino, A., Zamarreño, J. M., Vilas, C., Tadeo, F., 2024. Un entorno virtual con controladores lógicos programables: aplicación en evaluación de estrategias de control. Revista Iberoamericana de Automática e Informática industrial 22 (1), 69–80.  
DOI: 10.4995/riai.2024.20881
- Ruscione, M., Checchi, A., Heiland, R., Barillot, E., Macklin, P., Calzone, L., Noël, V., 10 2024. Building multiscale models with physiboss, an agent-based modeling tool. Briefings in Bioinformatics 25 (6), bbae509.  
URL: <https://doi.org/10.1093/bib/bbae509>  
DOI: 10.1093/bib/bbae509
- Schweiger, G., Gomes, C., Engel, G., Hafner, I., Schoegg, J., Posch, A., Nouidui, T., 2019. An empirical survey on co-simulation: Promising standards, challenges and research needs. Simulation Modelling Practice and Theory 95, 148–163.  
DOI: 10.1016/J.SIMPAT.2019.05.001
- Sirovich, L., 1987. Turbulence and the dynamics of coherent structures. Part I: Coherent structures. Quarterly of Applied Mathematics 45 (3), 561–571.
- Smith, S., contributors, 2024. PythonFMU3: A lightweight framework for packaging Python 3 code as co-simulation FMUs. Version 0.1.15.  
URL: <https://github.com/stephensmith25/PythonFMU3>