

HACIA LA UNIFICACIÓN DE LAS HERRAMIENTAS DE SIMULACIÓN DE APLICACIONES SOLARES PARA PROCESOS INDUSTRIALES

Frasquet M.*, Aramburo-Pasapera J.A.** , Nájera-Trejo M.** , Silva M.***

* SOLATOM, Universidad Politécnica de Valencia, C/ Pedro Duque, 7, 46022 Valencia,
miguel.frasquet@solatom.com

** Centro de Investigación en Materiales Avanzados, Calle CIMAV 110, Arroyo Seco, 34147 (Durango) México

***Ingeniería Energética, Universidad de Sevilla, Camino de los Descubrimientos, s/n, 41092 Sevilla

<https://doi.org/10.34637/cies2020.2.4151>

RESUMEN

La disponibilidad de herramientas que permitan realizar estudios de viabilidad en proyectos de energía solar térmica para procesos industriales (SHIP), de manera rápida, precisa y con bajo coste, es crítica para la aceptación de esta tecnología por el sector industrial. Las herramientas surgidas en respuesta a esta demanda son muy heterogéneas, lo que dificulta la comparación de proyectos. Esta disparidad limita el desarrollo comercial de la tecnología al generar incertidumbre en empresas promotoras y entidades financieras. En este artículo se presenta una metodología para unificar los esfuerzos que se están llevando a cabo para el desarrollo de estas herramientas de simulación. La metodología propuesta combina la utilización de código abierto y privado. De esta forma se garantiza un desarrollo conjunto y continuado por universidades y centros público/privados, además de permitir a empresas privadas su utilización, manteniendo privada cualquier información sensible.

PALABRAS CLAVE:

Energía Solar, Simulador, Código abierto, Procesos Industriales

ABSTRACT

The availability of tools capable of performing pre-feasibility studies of Solar Heat for Industrial Processes (SHIP) projects in a fast, precise, and low cost way has become necessary for the acceptance of this technology by the industrial sector. The tools that have emerged so far are very heterogeneous, making it difficult to benchmark projects. The significant disparity in the tools limits the commercial development of the technology by generating uncertainty in project promoters and financial entities. This work presents a methodology to unify the efforts that are being carried out for the development of these simulation tools. The proposed methodology combines the use of open and proprietary source. In this way, open source development is maintained by universities and public / private centers, while private companies still keep any sensitive information private.

KEYWORDS:

Solar Energy, Simulator, Open Source, Industrial Processes

INTRODUCCIÓN

El elevado potencial de la energía solar térmica para procesos industriales ha sido identificado en numerosos estudios. Sin embargo, sólo el 0.02% de la demanda energética global está cubierta con tecnologías solares (IEA, 2019). La complejidad y variedad de los diferentes procesos industriales, hace que sea necesario la realización de análisis específicos para cada caso. El elevado coste y tiempo necesario para realizar dichos análisis, representa una de las barreras más importantes de la tecnología. Para superar esta barrera, se están desarrollando numerosas herramientas de simulación capaces de hacer estudios de viabilidad de manera rápida y sencilla.

El desarrollo de herramientas de viabilidad suele estar englobado normalmente en proyectos de investigación, o en la estrategia comercial de empresas fabricantes de equipos solares. Cuando la herramienta proviene de un proyecto de investigación, el desarrollo suele ser colaborativo, y coordinado normalmente por universidades y centros de investigación. La principal ventaja de estas herramientas es que suelen estar validadas con resultados reales (llevados a cabo durante el proyecto), y que existe información pública sobre su funcionamiento y base teórica. Estas herramientas tienen una flexibilidad limitada a la hora de seleccionar esquemas de integración en la industria, los datos económicos (de incluirse) suelen tener una incertidumbre elevada, y su mantenimiento está acotado a la duración del proyecto. Algunos ejemplos de estas herramientas son la herramienta InSun (desarrollada en 2015 durante el proyecto FP7 del mismo nombre), o Appsol (desarrollada en 2014 por Aguasol para un proyecto del CORFO chileno).

Las herramientas de viabilidad desarrolladas por fabricantes de colectores permiten realizar simulaciones utilizando únicamente sus equipos. La ventaja de estas herramientas es que disponen de datos reales de colectores, y costes reales de mercado. Además, suelen estar bien mantenidas ya que forman parte de la estrategia comercial de la empresa. La mayor desventaja es que el código de las herramientas no suele ser público, para que la información sensible de la empresa no pueda ser accesible. Ejemplos de este tipo de herramientas son la del fabricante sueco de colectores cilindro parabólicos Absolicon, o la de Oventrop para colectores de tubo de vacío.

El elevado número de herramientas disponibles, y su heterogeneidad, dificulta el benchmarking de la tecnología. También implica que se están destinando recursos a desarrollar herramientas nuevas, las cuales, en mayor o menor medida, cumplen el mismo objetivo y tiene una estructura similar. Actualmente, además de las ya existentes, un elevado número de herramientas de este tipo está siendo desarrollado en paralelo, bajo el marco de proyectos como INSHIP, SHIP2FAIR, Solar-Payback o SECASOL.

METODOLOGÍA

La propuesta que aquí se describe trata de combinar las ventajas del código abierto en lo relativo a desarrollo colaborativo, con la privacidad del código privado. La estructura propuesta utiliza un código común único, abierto y accesible públicamente como motor para las simulaciones, conectado a un conjunto de interfaces de código privado. El motor de simulación de código abierto está alojado en un repositorio público. En este repositorio se gestiona y organiza las contribuciones de los diferentes participantes en el desarrollo. Este tipo de repositorios públicos de código abierto incorporan además las herramientas necesarias para facilitar el desarrollo de software colaborativo (sistema de versiones, resolución de conflictos, etc.). Las interfaces donde reside la información sensible están por el contrario, alojadas en los servidores privados de la empresa encargada de su desarrollo.

Cuando un usuario desea realizar una simulación, accede a la interfaz privada e introduce los datos de entrada de la simulación. La interfaz accede al repositorio público y carga la última versión del motor de simulación en el servidor privado. Al motor de simulación se le alimenta con la información necesaria para realizar la simulación (datos de entrada del usuario, datos de rendimiento del captador, información de coste, etc.). Una vez la simulación termina, el motor genera los resultados y estos se presentan al usuario a través de la interfaz. Cualquier modificación y mejora en el motor de simulación se traslada a las interfaces cuando estas cargan el motor desde el repositorio. De esta manera es posible desarrollar el motor de simulación de manera colaborativa, pública y abierta, manteniendo privada la información sensible necesaria para llevar a cabo las simulaciones. La figura 1 describe la arquitectura propuesta.

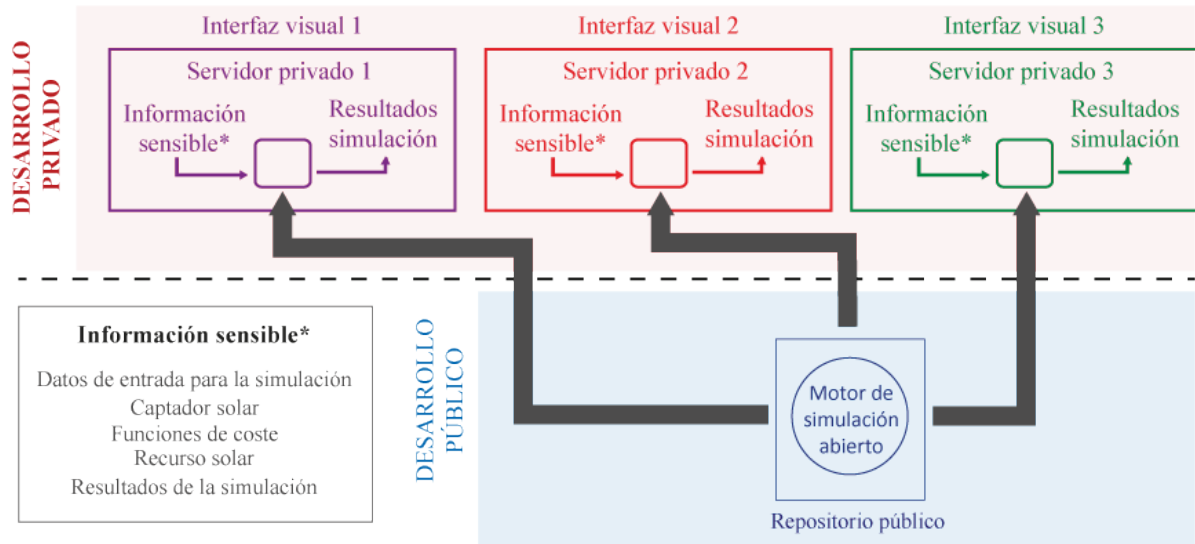


Fig. 1. Esquema de la arquitectura propuesta

Para el correcto funcionamiento de la metodología descrita es fundamental garantizar que el futuro desarrollo del código de simulación no interfiera ni inhabilite el funcionamiento de las interfaces conectadas a él. Para ello es necesario mantener cierto grado de retrocompatibilidad. Un crecimiento significativo de las interfaces conectadas al motor de simulación supondría también un esfuerzo creciente para mantener la retrocompatibilidad con todas ellas. Afortunadamente, en la metodología propuesta la interconexión entre motor de simulación e interfaz es únicamente a través de los inputs y outputs del primero. Garantizar la futura retrocompatibilidad pasa por verificar automáticamente esta interconexión. Estos mecanismos de verificación automática existen desde hace años (Ponomarenko y Rubanov, 2012).

IMPLEMENTACIÓN

La metodología descrita ha sido puesta a prueba utilizando SHIPcal como motor de simulación. SHIPcal es un software de código abierto, para la simulación de aplicaciones solares de concentración en procesos industriales (Frasquet, 2016). La primera interfaz conectada a SHIPcal se denomina ReSSSPI y ha sido desarrollada por la empresa SOLATOM. La interfaz ReSSSPI se conectó al motor SHIPcal en Diciembre 2017. Desde entonces se han llevado a cabo 5079 simulaciones por 243 usuarios de 32 países diferentes.

VALIDACIÓN

La interfaz ReSSSPI, pese a ser la primera interfaz conectada a SHIPcal, no valida la metodología descrita, ya que su desarrollador, SOLATOM, es también el desarrollador principal del motor de simulación SHIPcal. Para validar la metodología es conveniente que las interfaces sean desarrolladas por entidades independientes. La primera validación externa se realizó por el Centro de Investigación en Materiales Avanzados (CIMAV) de Durango (México). En 2019 el CIMAV realizó una interfaz propia adaptada a las características del mercado mexicano.

La interfaz del CIMAV se desarrolló en Python, el mismo lenguaje en el que está escrito SHIPcal. La interfaz utiliza la plataforma de trabajo (framework en inglés) Django. Django es uno de los frameworks de desarrollo web de Python más utilizados. Este framework se estructura por medio de aplicaciones, cada una desempeña una tarea específica dentro de la interfaz (p.ej. modificar un usuario, mostrar los resultados o generar los informes de resultados). La interfaz está alojada en el servidor del CIMAV y es accesible a través de una dirección web (<http://shipcal.cimav.edu.mx/simulate/>). La información de los datos de entrada de las simulaciones, variables financieras y las características de los colectores, está alojada en el servidor del CIMAV, por lo que su acceso está restringido.

Además del desarrollo de la interfaz, el CIMAV agregó funcionalidades y se mejoraron algunas funciones clave del motor de simulación SHIPcal. Estas mejoras se coordinaron a través del repositorio público de SHIPcal.

MEJORAS EN EL MOTOR DE SIMULACIÓN INTRODUCIDAS POR EL CIMAV

Temperatura de agua de red

Algunos de los modelos de integración de SHIPcal son de circuito abierto, es decir, utilizan en el cálculo termodinámico la temperatura del agua suministrada por una red externa, en lugar de utilizar la temperatura del agua de retorno del proceso. En la versión original de SHIPcal, la temperatura del agua proveniente de una red de distribución externa al proceso, se calculaba con valores promedios mensuales de la red de España. Con el objetivo de tener una aproximación más general, se realizó una modificación para que la temperatura del agua proveniente de la red se calcule como función de la información meteorológica específica de la locación seleccionada para la simulación. La temperatura del agua proveniente de la red se obtiene a partir de la Ec. (1) (Klein et al., 2018)

$$T_{grid} = \bar{T}_{amb} + 3 + ratio(T_{max}/2)\sin(-90 + (day - 15 - lag)360/365) \quad (1)$$

$$ratio = 0.22 + 0.0056(\bar{T}_{amb} - 6.67)$$

$$lag = 1.67 - 0.56(\bar{T}_{amb} - 6.67)$$

Donde day es el número de día del año con años de 365 días, T_{max} es la temperatura ambiente máxima del año, y \bar{T}_{amb} es la temperatura ambiente promedio del año. Todas las temperaturas son definidas en grados Celsius.

Simulación de colectores planos

SHIPcal se desarrolló originalmente para la simulación de concentradores solares, por lo que la definición del rendimiento de los captadores seguía la metodología descrita en (Eck et al., 2014). Para permitir la simulación de captadores planos, el código se habilitó para recibir los parámetros de eficiencia máxima del colector (ρ_0) y de pérdidas térmicas (ν_1 y ν_2), de manera que sea posible construir la ecuación cuadrática general de eficiencia Ec. (2) (Duffy y Beckman, 1991).

$$\eta = \rho_0 K_{(\tau\alpha)} - \frac{\nu_1(T_m - T_a)}{G} - \frac{\nu_2(T_m - T_a)^2}{G} \quad (2)$$

El factor $K_{(\tau\alpha)}$ es el modificador de ángulos e incidencia y se calcula en cada hora resultando en una ecuación de eficiencia diferente para cada hora. El valor de $K_{(\tau\alpha)}$ se puede obtener con dos métodos diferentes; con una tabla de valores de $K_{(\tau\alpha)}$ conocidos para ángulos de incidencia previos e interpolando linealmente para los ángulos no contemplados (este es el método pre-existente) o a partir de Ec(3) (Duffy y Beckman, 1991).

$$K_{(\tau\alpha)} = 1 - b \left(\frac{1}{\cos(\xi)} - 1 \right)^n \quad (3)$$

Los parámetros b y n se obtienen a partir de la interpolación lineal de valores conocidos de $K_{(\tau\alpha)}$ en Ec. (4), la cual no es más que Ec. (3) reescrita con logaritmos y ξ es el ángulo de incidencia en el plano.

$$x = \ln \left(\frac{1}{\cos(\xi)} - 1 \right)$$

$$y = \ln(1 - K_{(\tau\alpha)})$$

$$y = \ln(b) + nx \quad (4)$$

Este segundo método ofrece una mayor exactitud en el cálculo de $K_{(\tau\alpha)}$ y menor consumo de memoria pues sólo utiliza dos variables en lugar de buscar e interpolar en un archivo completo para computar el $K_{(\tau\alpha)}$.

Además, SHIPcal es ahora capaz de seleccionar del archivo meteorológico la Irradiancia Normal Directa (DNI) si se simula un colector de concentración, o la Irradiancia Global Horizontal (GHI) si se utiliza un colector de placa plana.

CONCLUSIÓN

Para el desarrollo masivo de proyectos solares en aplicaciones industriales es necesario que los promotores tengan acceso a herramientas que permitan analizar la viabilidad de este tipo de instalaciones. La metodología presentada, que conecta un motor de simulación de código abierto con interfaces privadas, combina de manera efectiva las ventajas de los simuladores existentes de código abierto y los de código privado. La metodología se ha validado

utilizando el motor de simulación SHIPcal. El desarrollo realizado por el CIMAV, ha permitido conectar una nueva interfaz a SHIPcal, especialmente diseñada para el mercado solar de México. A día de hoy, tanto ReSSSPI como la interfaz del CIMAV están conectadas a SHIPcal, y ambas se han beneficiado de las mejoras introducidas en él, probando la viabilidad de desarrollar colaborativamente un único motor de simulación común.

REFERENCIAS

Duffie J.A. and Beckman W.A. (1991) *Solar Engineering of Thermal Processes*, 2nd edn. pp. 289-296. Wiley Interscience, New York.

Eck, M., Hirsch T., Feldhoff, J.F., Kretschmann, D., Dersch J., Gavilan Morales, A., Gonzalez-Martinez, L., Bachelier, C., Platzer, W., Riffelmann, K-f., Wagner, M. (2014). Guidelines for CSP Yield Analysis – Optical Losses of Line Focusing Systems; Definitions, Sensitivity Analysis and Modeling Approaches, *Energy Procedia*, Volume 49, 2014, Pages 1318-1327, ISSN 1876-6102, <https://doi.org/10.1016/j.egypro.2014.03.141>.

Frasquet M., (2016). SHIPcal: Solar Heat for Industrial Processes Online Calculator. *Energy Procedia*, ISSN: 1876-6102, Vol: 91, Page: 611-619

IEA (2019), *Renewables 2019*, IEA, Paris <https://www.iea.org/reports/renewables-2019>

Klein, S.A., et al., (2018) *TRNSYS 18 –Mathematical Reference*, pp. 4–697

Ponomarenko, A., Rubanov, V. (2012) Backward compatibility of software interfaces: Steps towards automatic verification. *Program Comput Soft* 38, 257–267 (2012). <https://doi.org/10.1134/S0361768812050052>