

SolarTherm: A New Modelica Library and Simulation Platform for Concentrating Solar Thermal Power Systems

Alberto de la Calle^{1*}, Jim Hinkley¹, Paul Scott², John Pye²

¹CSIRO Energy, Newcastle (NSW), Australia; *alberto.delacallealonso@csiro.au

²Australian National University, Canberra (ACT), Australia

SNE 28(3), 2018, 101-103, DOI: 10.11128/sne.28.sn.10427
 Received: Sept. 15, 2016 (Selected EUROSIM Congress 2016 Postconf. Publ.), Revised Sept. 15, Accepted: Sept. 20, 2018
 SNE - Simulation Notes Europe, ARGESIM Publisher Vienna, ISSN Print 2305-9974, Online 2306-0271, www.sne-journal.org

Abstract. SolarTherm is a new CST technology simulation platform. Based on the Modelica language, it offers an easy and high-flexible alternative to the conventional CST simulation tools. SolarTherm is composed by a Modelica library of components and systems full-editable and runnable with the main Modelica environments and Python-based tools and scripts that provides a simulation framework.

Introduction

According to the International Energy Agency (IEA), around the 11% of the global electricity production in 2050 would be generated by Concentrating Solar Thermal (CST) power systems [1]. With this scenario, CST technology has a high potential to become one of the key technologies in a decarbonised energy future.

To make this a reality, researchers and industry must strive to reduce the investment, operating and maintenance CST cost until this technology become competitive with other energy sources. Although solar thermal offers a number of benefits over other renewable energy technologies, its cost level is yet far from others technologies like the wind, hydro or photovoltaic power stations.

The Australia Solar Thermal Research Initiative (ASTRI) [2] is working to develop disruptive technologies that dramatically reduce CST costs. In order to achieve this goal, it focussed its effort on four strategic objectives:

- Increasing the capacity factor - extending operating hours to sell more electricity to the grid.
- Reducing capital expenditure for CST power plants.
- Improving efficiency of CST components and overall systems.
- Adding product value: through (a) solar fuels and (b) improving O&M technologies yield and reduce cost.

It is necessary to collect and test all proposed ideas through the modelling and simulation. There are several simulating tools for carrying out system performance and financial simulations for CST power system, e.g. SAM [3] or Greenius [4]. Although these tools are suitable for modelling and simulation of conventional systems, it may be excessively complicated to simulate novel designs like some of the proposed systems in ASTRI project.

SolarTherm, presented here, aims to meet this need. It provides a simulation platform on which users can easily modify partial previous-designed systems and/or develop new ones.

1 SolarTherm

SolarTherm was created under the ASTRI program with the aim to provide solar thermal community with an easy tool to simulate and optimise CST power plants. The focus of this tool is the annual performance and the economic assessment of novel designs of solar thermal plants.

It consists of a CST component and system library along with a simulation framework. As other simulation tools, it allows a high-level process modelling through

the connection of previous defined components where users can simulate systems configuring only few parameters such as the size of the storage or the receiver area. However the attraction of this tool is its high flexibility, where it is easy to replace, develop or customise new components. This makes SolarTherm a suitable tool to explore of new technologies related with CST power systems including thermochemical systems (i.e. solar fuels).

SolarTherm is a tool based on the object-oriented modelling Modelica language [5] for its component models and simulation. This language provides class encapsulation allowing a modular and hierarchical modelling with replaceable and customized components. Each of them can be formulated acausally, i.e. the translation of the physical equations into the formulation language is direct, making it very well suited for representing the physical structure of systems. Also, Modelica can be interfaced with external tools allowing a stochastic statistical analysis of the plant performance.

SolarTherm provides a range of Python-based tools and scripts that automate the process of testing, simulating, optimising and visualising the results [6]. Besides, if the user prefers to manage flow-sheet diagrams, Modelica library can be simulated with user-friendly graphical Modelica environments such as OMEdit [7] or Dymola [8].

One of the strengths of SolarTherm is that is a fully open source project. All the code is fully accessible and hackable including component models as well as the simulation framework on a GitHub server (<https://github.com/solartherm/solartherm>). Being fully open source and fully compatible with the free-open source Modelica language simulator OpenModelica [7], it aims to become an attractive tool where researchers joint their efforts working on the future of the CST technology.

1.1 Description of the library

The component and system library has been implemented with the last available version of the the Modelica language (3.2.1) [5]. Modelica Fluid and Modelica Thermal connectors have been used in order to define the relationships between components. The thermodynamic properties have been calculated with medium models that extend the Modelica Media interface. These decisions guarantee that SolarTherm is compatible with any component from the Modelica Standard Library or another component that are imple-

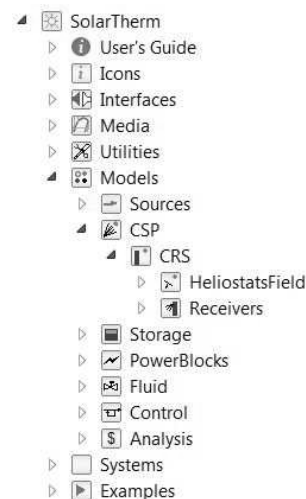


Figure 1: The SolarTherm library.

mented with the same interfaces. Also, all the components are locally balanced helping to ensure robust modelling and debugging [9].

Figure 1 shows the current main packages that composes the SolarTherm library. For the moment, the library is focused on Central Receiver Systems (CRS) with storage for electricity production. Some simple models for all these domains have been implemented including some fluid or control models to guarantee the stability of the systems.

An example of a CRS model implemented with SolarTherm library is depicted in Figure 2. A data file provides the input weather data necessary to run the system. The sun model provides the sun position and the Direct Normal Irradiance (DNI) to the heliostats field which calculates the total concentrated power into the receiver. Part of this energy is transferred to the working fluid (molten salts) which flows from the cold tank to the hot tank. The fluid returns to the cold tank after crossing the power block where the electricity produced by the plant is calculated. Two pumps with each automatic control systems drives the fluid from one tank to the other.

2 Conclusion

In this work we presented the work-in-progress to develop SolarTherm, a new CST technology simulation platform. It is designed to offer user an easy tool suitable for the modelling and simulation of CST power

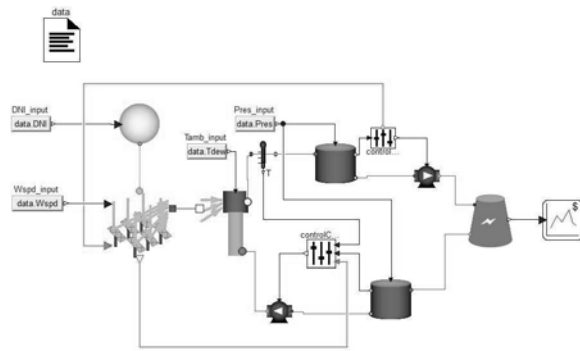


Figure 2: Flow-sheet diagram example.

systems including unconventional ones. SolarTherm is composed by a Modelica library of components and systems and Python-based tools and scripts that provides a simulation framework. It is also compatible with the user-friendly Modelica environments OMEdit and Dymola and it is developed with an open source philosophy, providing all the code in a GitHub server.

Acknowledgement

This work was performed as part of the ASTRI, a project supported by the Australian Government, through the Australian Renewable Energy Agency (ARENA).

References

- [1] IEA. World Energy Outlook 2014. *Tech. rep.*, International Energy Agency, Paris, France. 2014. URL <http://www.worldenergyoutlook.org/publications/weo-2014/>
- [2] ASTRI. Australian Solar Thermal Research Initiative. 2016. URL <http://www.astri.org.au/>
- [3] SAM. Solar Advisor Model. 2016. URL <https://sam.nrel.gov/>
- [4] Greenius. *The green energy system analysis tool*. DLR, version 4. ed. 2016. URL <http://freegreenius.dlr.de>
- [5] Modelica Association. Modelica Specification 3.3. 2016. URL www.modelica.org/documents
- [6] Scott P, de la Calle A, Hinkley JT, Pye J. SolarTherm : A flexible Modelica-based simulator for CSP systems. In: *SolarPaces Conference*. Abu Dhabi, United Arab Emirates: Upcoming. 2016; .
- [7] Open Source Modelica Consortium. OpenModelica. 2016. URL <https://openmodelica.org/>
- [8] Dassault Systemes. Dymola 2015 - Dynamic Modeling Laboratory. 2016. URL www.3ds.com
- [9] Olsson H, Otter M, Mattsson SE, Elmqvist H. Balanced Models in Modelica 3.0 for Increased Model Quality. In: *Proc. 6th International Modelica Conference*. Bielefeld, Germany. 2008; pp. 21–33.