A Universal Heliostat Control System

Fabian Gross¹,a), Mark Geiger¹ and Reiner Buck¹

¹Institute for Solar Research, German Aerospace Center DLR, Wankelstr. 5, 70563 Stuttgart, Germany

a)Corresponding author: Fabian.Gross@dlr.de

Abstract. This paper describes the development of a universal heliostat control system as part of the AutoR project [1]. The system can control multiple receivers and heliostat types in a single application. The system offers support for multiple operators on different machines and is designed to be as adaptive as possible. Thus, the system can be used for different heliostat field setups with only minor adaptations of the system’s source code. This is achieved by extensive usage of modern programming techniques like reflection and dependency injection. Furthermore, the system features co-simulation of a ray tracer, a reference PID-controller implementation for open volumetric receivers and methods for heliostat calibration and monitoring.

INTRODUCTION

The heliostat field collects, reflects and concentrates solar radiation and irradiates the receiver of a solar power plant. The task of controlling the irradiation is challenging because of several competing targets: maximizing the intercepted radiative power whilst respecting flux limitations and temperature gradients to enhance the life span of the receiver. A heliostat control system has to ensure safe operation, reduce spillage and satisfy the receiver requirements.

The paper is organized as follows: The first section describes the system architecture and its modular design principle. Then, we present the core task of system control and the achieved performance in terms of temperature control and update rate. Next follow the important support tasks of heliostat calibration and monitoring. The presentation of the graphical user interface (GUI) shows the key features of the implementation. Finally, we summarize and provide an outlook.

MODULAR DESIGN PRINCIPLE

The control system is written in Java using the Maven build tool. Its architecture can be seen in Figure 1. The system follows the principles of client/server-based system architecture. The core system can be deployed on any computer that runs a tomcat webserver (server). The main system offers an open web based REST protocol to communicate with the clients. Therefore, the system is independent from its GUI (client) and supports multi-operator access with a suitable hierarchy of user rights. The heliostat command handler translates general commands to specific protocols.
The core system is divided into several Maven submodules to clearly organize dependencies. The design paradigms of dependency injection and reflection allow the code to adapt to new heliostat/receiver types or to configurable setups. Several layers of abstract classes and interfaces ensure that new heliostat/receiver types can be added to the system by creating new child classes of the general abstract implementations. Extensive usage of the observer design pattern allows loose communication across the different submodules. This ensures that each submodule is mostly independent from other modules. This increases the flexibility and maintainability of the system.

**SYSTEM CONTROL**

As safe operation is the key requirement to system control, the control loop first ensures that critical parameters (e.g. temperatures, wind speed, pressure, mass flow) are within the allowed range. Otherwise the controls triggers an emergency and a sends a defocus command. An independent watchdog thread makes sure that the main loop is working or suppresses the “alive”-heartbeat. With each new aim point, the control system communicates an escape route to the heliostats which they follow in case of defocus in order to avoid high flux densities in vulnerable areas.

The control system has a dedicated implementation that corresponds to the autonomous rim-drive heliostat with its local intelligence by Trinamic Motion Control [3] and wireless communication/safety system by the Institute of Telematics/TUHH [4] which is tested at the solar tower Jülich.

The solar plant’s administrator can dynamically assign heliostats to one of the receivers due to the receiver’s representation as individual state machines with flexible update mechanisms. Once a receiver owns the heliostat, the mirror is subject to the receiver’s control logic and is employed for its control and optimization duty. Other modes like calibration and maintenance are integrated into the ownership model.

The reference implementation of the receiver control uses a PID-controller to control the receiver outlet temperature by setting the input power. With individual heliostat efficiencies from ray tracing and DNI measurement the controller converts the power into the necessary number of heliostats. The PID-parameters follow from “process reaction curve”-method [5], hence adapt to different receiver thermal capacities, power ratings and load factors as well as dead times until heliostats respond. D-part is averaged over 100 frames (ca. 1.6 s) to smooth out possible noise in the DNI and temperature data.

The control adds the standby-heliostat with the highest ray traced efficiency if more irradiation is necessary and removes the weakest tracking heliostat if power has to be reduced. This choice can be combined with geometrical considerations to achieve a desirable flux density distribution.

**PERFORMANCE**

Figure 2 shows the success of the control for the simulated open volumetric receiver and simulated 2153 heliostats: the actual output temperature follows closely the set value through ramp up, operation and ramp down although the synthetic DNI shows sudden changes from “clouds”. Temperature deviations happen when the available power is insufficient to meet the demand – either because of initial movement from stow to track or lack of DNI – or the ramp down rate is faster than the thermal losses and cooling mass flow even with full defocus. An active control of the receiver mass flow will be implemented next.
When simulating the Jülich field with its 2153 heliostats, the control system achieves the targeted 60 frames/s on a standard processor (tested with an Intel® Core™ i7 870 @ 2.93GHz CPU). The system aims to support fields up to 50000 heliostats. To keep up with the high amount of needed calculations in larger heliostat fields, it is planned to transfer calculations that have to be done for each heliostat to the GPU.

![Graphs showing temperature and DNI perturbation over time](image)

**FIGURE 2.** Simulation of a receiver heating up (2.5 °C/s) and cooling down. The upper graph shows the “is” \(T_{\text{is}}\) and “set” \(T_{\text{set}}\) temperature and their difference, the middle diagram shows the DNI perturbation and the lower diagram shows the number of commands to the heliostat field sent by the control system.

**CALIBRATION**

The most important support processes are heliostat calibration and monitoring: In the current implementation, heliostats aim at an external target plane. An industry-grade camera observes the focal spot. Image acquisition and processing [2] extracts the relevant flux density distribution and characterizes position and shape. Finally, a fit of a kinematic model to the data improves the tracking accuracy. In the reference implementation, the model has 8 degrees of freedom to account for the orientation of the axes and gear ratios. The offset between axes and mirror are taken into account. The analysis is precise enough to reveal a 0.1 mrad polygon effect in the chain drive, see Figure 3. The motor angle \(\tau_{\text{M}}\) is transferred to the mirror rotation by an alternating gear ratio as the 9-toothed gear wheel forms a polygon rather than a perfect circle. By a modification of the calibration module, new types of measurements or procedures can be incorporated into the control system as they become available.
The choice of heliostat to be calibrated next depends on a heuristic that takes the individual need for measurement and tracking improvement into account.

The heliostat monitoring compares the expected and measured flux density distribution to highlight deviations that might need the attention of the maintenance team. The ray tracer (SPRAY) uses a detailed heliostat surface model based on deflectometry data, reflectivity and a simultaneous DNI measurement. A calibrated Baumer-camera quantifies the flux density of the empty background and the focal spot on the target plane. The camera exposure is set such that the empty target creates a background intensity of ca 80/255 and the brightest focal spot is below saturation at 255/255, therefore the dynamic range is properly used and the intensity measurement per pixel is accurate to a few percent. The spatial resolution of the image is 3 cm/pixel. In Figure 4, a facet with 30 m focal length aims at a calibration target ca. 90 m away. The comparison shows that the flux density agrees well for most of the reflector surface. On the one hand there are regions with reduced irradiance (dark red) where a mirror fault (here a PV panel has been installed on the mirror after deflectometry measurement) happened. On the other hand excess radiation (blue) from another heliostat can perturb the calibration and leads to a rejection of this measurement.

**FIGURE 3.** shows the slight oscillations in azimuth of the normal vector due to the chain drive’s polygon effect.
FIGURE 4. shows flux densities for a prototype heliostat whose focal length is about a third of the distance to the calibration target. Left: predicted flux density based on deflectometry data; center: measured flux density; right: difference between ray traced prediction and measurement. The x/y coordinates are pixels in the orthoimages whilst the flux density is shown as kW/m².

The raw data and results of each measurement are stored in a database. The fit routine uses the data modulated with a statistical weight to account for reduced reliability in bad weather (high wind, low DNI). The calibration can incorporate a different kinematic model for each heliostat type.

GRAPHICAL USER INTERFACE

The graphical user interface allows the operator to load and adapt configuration files, change the receiver mode (ramp up, normal operation, ramp down, emergency), check the performance and critical parameters of the plant, observe the calibration process of heliostats. The administrator can dynamically allocate heliostats to different receivers and their respective operators.

Figure 5 shows a 2-receiver operation with heliostats allocated to left or right receiver or none. The upper right panel allows the configuration of parameters during runtime. The lower right panel plots set/is-temperature curves for the two receivers. The panels can be adapted in size and position to give the operator the relevant information for the task at hand and allow maximum control.
FIGURE 5. Exemplary implementation of GUI shows heliostat field tracking on two receivers in parallel. The visualization of heliostats shows corresponding receiver, current orientation and efficiency based on ray tracing.

SUMMARY

We show the philosophy of a universal heliostat control system and present the modular implementation. Heliostat calibration and monitoring allow for minimal tracking errors and targeted maintenance work. The test at the Solar Tower Jülich shows the functionality of both heliostat and control system. The current implementation is already strong enough to handle more than 2000 heliostats with 60 Hz update rate.

The control system is proposed for a solar tower research project in Brazil and available for licensing.

ACKNOWLEDGMENTS

The authors would like to acknowledge the funding by the German Ministry for the Economy and Energy (BMWi AutoR-Project 0325629C). We thank our colleagues Christoph Prahl and Timo Effertz for their support in measurement, image processing and calibration. Thanks to the work of Trinamic Motion Control and Institute of Telematics/TUHH, we could set up a full system from heliostat via mesh network to control system.

REFERENCES