The present work was submitted to the Institute of Automatic Control of RWTH Aachen University

Development of Control Strategies for Heliostat Aiming in Solar Power Tower Plants

Master Thesis
of
David Zanger, B. Sc.
Matriculation Number 333845

Field of study:	Automation Engineering
Duration of this work:	22 Weeks
Submitted on:	04.11.2020
Number:	S2108
Supervisor:	Eugen Nuss, M. Sc.
External supervisor:	Laurin Oberkirsch, M. Sc.

Ich versichere, diese Arbeit im Rahmen der am Institut üblichen Betreuung selbständig angefertigt und keine anderen als die angegebenen Quellen verwendet zu haben.

(David Zanger)

Aachen, 04.11.2020

Abstract

In this work, two closed-loop controllers are proposed, which control the flux density distribution on the receiver of a solar power tower plant. The first controller, called Hot Cold Control, is based on the idea from GarcÍa-MartÍn et al. [1] and modified to work with the flux density measurement as input. Secondly, a novel control approach is presented, called Static Optimal Control, which is based upon the ant colony optimization algorithm. In a simulation, the controllers are tested under three test cases. The first two test cases evaluate the controllers capacity to compensate for disturbances like mirror errors and to react to overflux conditions. The last test case investigates the controller's reaction to dynamic disturbances like clouds. The Static Optimal Control shows the most promising results. Under the influence of a mirror error it raises the relative power from 88.9 % to 92.2 %. This is close to the calculated maximum of 93.7 % and 1.67 % higher than the Hot Cold controller, which yields a relative power of 90.53 %. The overflux condition in test case 2 is eliminated after one control step by the Static Optimal Control and results in a relative power of 92.02 %. The Hot Cold control requires 61 steps and yields a relative power of 90.53 %. Only the Static Optimal Control is tested for test case 3, where it increases the power while the heliostat field is partly shaded and compensates overflux conditions after the cloud passes. Finally, the Static Optimal Control is tested on the solar tower in Jülich. Here, the controller also shows promising results. However, for a commercial plant the elimination of an overflux condition is not fast enough. To guarantee a robust control behavior, further testing and development has to be done.

Contents

List of Figures

List of Tables

List of Symbols

Latin Symbols

$\mathbf{A}$	System matrix
$A^{(1)}$	Area of one bin
$A^{(2)}$	Area of an absorber module
A_H	Mirror area of a heliostat
$\mathbf{B}$	Input matrix
b	Improvement factor
$\mathbf{C}$	Output matrix
$c_{\mathrm{p,air}}$	Heat capacity of air
$c_{\mathrm{p,module}}$	Heat capacity of an absorber module
$\mathbf{D}$	Feedthrough matrix
d	Horizontal distance between tower and heliostat
d_{y}	Distance covered in y-direction
e	Error
efr	Exceeded weighted flux densities
F	Flux
F_{max}	Maximum allowed flux
f_{at}	Air attenuation factor
f_{bs}	Blocking and shading factor
$f_{\mathrm{intercept}}$	Intercept factor
h	Index heliostat
$h_{\mathrm{heliostat}}$	Heliostat height
h_{receiver}	Receiver height
h_{tower}	Tower height
I_D	Direct normal sun irradiation
J	Objective function
$\mathbf{K}$	State feedback matrix
K_{d}	Derivative factor
K_{i}	Integrating factor
K_{p}	Proportional factor
k_{noc}	Number of steps after no overflux occurs
$\dot{m}_{\mathrm{air}}$	Mass flow of air
m_{module}	Mass of an absorber module
e_{mse}	Mean squared error

n	Natural number
P_H	Power reflected by a heliostat
$P_{\mathrm{heliostats}}$	Reflected power by the heliostats
P_{receiver}	Power on the receiver
P_{rel}	Relative power
$p^{(1)}$	Proximity
$p^{(2)}$	Probability
$p^{(3)}$	Penalty factor
$\dot{Q}_{\mathrm{air,in}}$	Incoming heat flow by air
$\dot{Q}_{\mathrm{air,out}}$	Outgoing heat flow by air
$\dot{Q}_{\mathrm{loss}}$	Heat loss through irradiation
$\dot{Q}_{\mathrm{rad}}$	Heat flow through irradiation
$\dot{Q}_{\mathrm{total}}$	Total heat flow
q	Random number
q_0	Decision parameter
$s^{(1)}$	Slant range (distance between the heliostat and the receiver center)
$s^{(2)}$	Edge
$s^{(3)}$	Scaled value
T_{a}	Ambient temperature
$T_{\mathrm{air,out}}$	Air outlet temperature
T_{max}	Maximum material temperature
T_{module}	Temperature of an absorber module
$\overline{T}_{\mathrm{air,in}}$	Mean air inlet temperature
t	Time
t_{max}	Critical time before the receiver is damaged
$\mathbf{u}$	Input vector
w	Weight
w_{receiver}	Receiver width
$\mathbf{x}$	State vector
x_{a}	x-coordinate of an aim point
$\mathbf{y}$	Output vector
y_{a}	y-coordinate of an aim point
y_{critical}	Critical flux density
y_h	Flux density for one heliostat
y_{m}	Measured flux density
y_{max}	Allowed flux density
y_{weighted}	Weighted flux density
$\hat{y}$	Simulated flux density

Greek Symbols

$\alpha^{(1)}$	Angular position of a heliostat
$\alpha^{(2)}$	Absorptivity of the absorber material
$\beta^{(1)}$	Angle between the normal vector of the heliostat and the vector of the sun rays
$\beta^{(2)}$	Weighting factor
ϵ	Emissivity of an absorber module
η	Visibility value
μ_{cloud}	Mean value last steps before cloud passes
μ_{f10}	Mean value first 10 steps
μ_{l20}	Mean value last 20 steps
ω	Angular velocity
ω_p	Angle between the normal vector of the receiver and the heliostat
ϕ	Flux density
$\rho^{(1)}$	Mirror reflectivity
$\rho^{(2)}$	Evaporation ratio
σ	Stefan-Boltzmann constant
σ_{HF}	Total effective deviation
σ_{slp}	Deviation caused by the mirror error
σ_{sun}	Deviation caused by the sun shape
σ_t	Deviation caused by the tracking error
τ	Pheromone value
ξ	Heliostat impact

List of Abbreviations

ADRC	Active disturbance rejection control
BIBO	Bounded-input bounded-output
DNI	Direct normal irradiation
LQR	Linear-quadratic regulator
MILP	Mixed-integer linear programming
MIMO	Multiple-input multiple-output
MPC	Model predictive control
PID	Proportional Integral Derivative
SISO	Single-input single-output

1 Introduction

1.1 Motivation

The expansion of renewable energies is constantly increasing, especially due to climate change and the resulting need to achieve climate targets. For example, the EU's 2030 Climate and Energy Policy Framework demands to increase the share of renewable energy to 32 % [2]. In 2018, the share of renewable energy in the EU was at 18.9 % and thus, needs to be expanded more than 50 % over the next 10 years [3].

In the field of solar energy a variety of different technologies to produce electricity such as photovoltaic, parabolic trough, Fresnel reflector, dish Stirling as well as solar towers are used. Among these technologies, solar power tower plants are a promising option. Compared to other solar technologies, they can achieve a higher theoretical overall efficiency, can be upscaled easily and cost-effectively and offer a way of storing energy to compensate load fluctuations [4]. One example for the use of stored energy are hybrid plants, which combine the more economical energy production of photovoltaic with the more economical energy storage of solar towers [5].

In a solar tower power plant, numerous mirrors focus the sunlight onto a receiver mounted on top of a tower. The concentrated radiation is absorbed by a heat transfer medium (e.g. molten salt) in the receiver. The energy of the medium can then be supplied to a connected energy conversion process (e.g. steam power process) for electricity production.

Nowadays, around 20 solar towers such as the Ashalim Power Station in Israel with a maximum capacity of 121 MW or the Khi Solar One in South Africa with a maximum capacity of 50 MW are operated [6]. Another four solar towers are under construction such as the Atacama-1 in Chile with a maximum capacity of 110 MW [7].

To make the expansion of solar tower power plants more economically attractive, an increase in efficiency is of great importance. One possibility to increase the efficiency is by optimizing the irradiation distribution over the receiver module. Current operated solar towers have a solar to electric conversion efficiency usually between 15 % to 18 % averaged over a year [8]. The objective is to minimize irradiation losses, while limiting the radiation per area to prevent damage to the receiver. Furthermore, the solar power tower plant is subject to disturbances such as clouds or mirror errors. Disturbances can lead to an increase or decrease of the irradiation. Thus, they either reduce the efficiency or may lead to critical damage to the receiver. Due to dynamic changes in the sun's position and the limit of the radiation during operation as well as the disturbances, an open- or closed-loop controller is required, which can react to these changes. Some models for calculating the optimal irradiation distribution and thus, for controlling the heliostats with an open-loop control are already available in the literature. However, there are only a few approaches to control the radiation distribution via a closed-loop control.

1.2 Objective

The objective of this thesis is to develop suitable control approaches for a solar power tower plant. The control approaches shall increase the efficiency of the solar power tower plant by increasing the power on the receiver. However, damage to the receiver shall be prevented and therefore, the maximum flux density must be restricted. Furthermore, the controllers shall reject disturbances such as tracking or mirror errors. Rejection of highly dynamic disturbances such as clouds is desirable. Additionally, the controller should be able to work with different receiver types (e.g. cylindrical or rectangular receivers).

Based on the literature review, it can be concluded that an open-loop control is not sufficient to face the control objectives. The open-loop controllers can model the disturbances but an inaccurate modeling of these disturbances may lead to a non-optimal control behavior. Furthermore, the controller should not depend on a precise estimation of the disturbances, which would imply an accurate measurement of them, which is often economically unreasonable. Thus, the open-loop controller is regarded as not capable to fulfill the requirement of rejecting disturbances.

In order to compensate for disturbances a closed-loop controller needs to be designed. So far, none of the controllers in the literature fulfills all the requirements. For example, one controller lacks the ability to work with rectangular receivers. One possibility to design a new controller would be to adapt the existing controller to work also with rectangular receivers. However, due to the fact that only a few closed-loop control approaches were developed so far, there may be other more suitable solutions to face the control objectives. For this purpose, new control strategies are developed in this work by considering existing control approaches used for solar power tower plants as well as control approaches from the control theory literature.

To develop a control strategy which fulfills the requirements, this work will follow the process of control system design as stated by Skogestad and Postlethwaite [9]. The process of control system design, which is slightly adapted, considers the following steps:

1. Study the system (plant) to be controlled and obtain initial information about the control objectives.
2. Model the system and simplify the model, if necessary.
3. Analyze the resulting model; determine its properties.
4. Decide which variables are to be controlled (controlled outputs).

5. Decide on the measurements and manipulated variables; what sensors and actuators will be used and where will they be placed.
6. Decide on the type of controller to be used.
7. Design a controller.
8. Analyze the resulting controlled system.
9. Simulate the resulting controlled system either on a computer or a pilot plant.

1.3 Structure of the Thesis

The work consists of six parts. With respect to the process of control system design, this chapter stated the control objectives (step 1). Chapter 2 gives an introduction to the solar power tower plant and explains basics regarding the control theory. Then, it presents solutions from literature and explains the ant colony optimization algorithm. Subsequently, chapter 3 models and analyzes the plant system (step 2 and 3) and then defines the inputs and outputs of the controller (step 4 and 5). Furthermore, possible closed-loop control strategies for the respective system are presented. From the presented controller, three controllers are selected (step 6). Two of them, are novel control approaches developed in this thesis and the last is taken from literature as a benchmark. The selected controllers are elaborated in more detail (step 7) and finally, a short description of the simulation software is given. The resulting control system is analyzed and optimized for one controller in chapter 4 (step 8). Finally, in chapter 5 the controllers are simulated on a computer and the most promising control is also tested on a real plant (step 9). Lastly, chapter 6 summarizes the thesis and gives an outlook for future work.

2 Fundamentals and Literature Review

In this work a closed-loop control for a solar power tower plant is developed. This chapter describes the structure of a solar tower system and the parts essential to this work. Subsequently, the chapter introduces the basics of control theory relevant for this thesis and presents solutions from the current literature. Finally, the ant colony optimization is described because knowledge about it is needed for the Static Optimal Control proposed in section 3.6.

2.1 Solar Power Tower Plant

A solar power tower plant converts solar radiation to electricity via an energy conversion process or uses it to support thermochemical processes. It consists of a multitude of mirrors called heliostats which focus the solar radiation onto a receiver which is mounted on top of a tower. The concentrated radiation is absorbed by a heat transfer medium in the receiver. The medium transfers the absorbed energy to an attached energy conversion process (e.g. a Rankine cycle to produce electricity). Instead of being transferred to the energy conversion process the medium can also be directed to an energy storage module for later use. [10]

Based on the preceding description a solar power tower plant can be divided into a series of functional components: the heliostat field, the receiver, the energy conversion system and the storage module which are depicted in Fig. 2.1. Within the context of this work the heliostats, the receiver as well as the influence of the sun are essential for further understanding. These parts are explained in more detail in the next three subsections.

2.1.1 Heliostats

The heliostats focus the solar radiation onto the receiver. Their aim points determine how much and where the radiation is concentrated on the receiver. Furthermore, they are affected by disturbances which lessen or divert the radiation reflected by the heliostats. The composition of the heliostat, the arrangement of them (*field layout*) as well as its deficiencies and physical limitations (*optical losses*) are described in the following paragraphs.

A heliostat is an optical aperture, which is used to always direct the solar radiation to the same fixed point, regardless of the change of the sun's position [10]. The solar radiation is directed by a reflecting surface, which is supported by a structure. A controller tracks the Composition

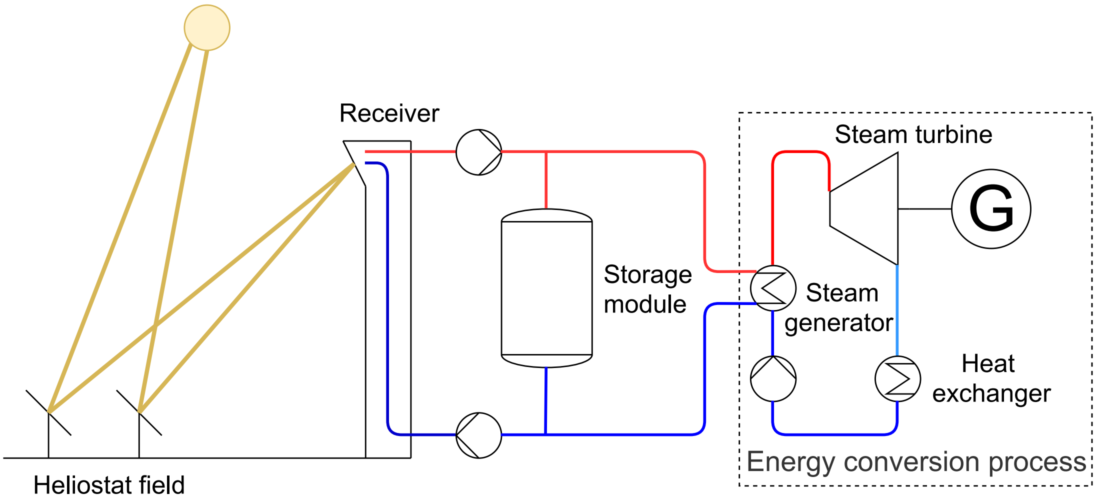


Figure 2.1: Simplified schematic representation of a solar power tower plant.

sun's position and moves the reflecting surface to keep the radiation directed to the same fixed point. Thus, the heliostat consists of a reflecting surface, a supporting structure and a tracking mechanism. Depending on the heliostat type the reflecting surface is made up of one or more mirrors called facets. The facets are usually arranged in the same plane but may be canted to achieve a certain focal length. From one type to another the facets can differ in size, form and amount. For example, the form may be rectangular but bent towards the inside to focus the radiation. Usually, the heliostats have a rectangular shape and its facets are supported by a T-structure due to economic reasons [11] as shown in Fig. 2.2. The tracking mechanism guarantees that the heliostat always aims at the same fixed point on the receiver, the so-called aim point. To enable this, the heliostat can usually be moved bidirectional in the direction of the elevation and the azimuth axis, which are illustrated in Fig. 2.3. For example big heliostats are moved by worm drives while smaller heliostats use push rod motors [10].

Field Layout

The aggregate of heliostats, which are ordered around the receiver of a solar power tower plant, is called the heliostat field. Heliostat fields consist of a multitude of usually identical heliostats, which are arranged on concentric circles or in rows. Heliostat fields can be divided into the categories all-round, north or south field as shown in Fig. 2.4. The category is determined by the geographical location. All-round fields located near the poles have an increasing lower field efficiency on one side of the field due to the steep incident angle of the radiation of this respective side. Thus, when moving towards the poles an all-round field becomes less economical. Other influential factors than the geographical position are stated in [12]. These factors concern for example the spacing between the heliostats.

Regarding this thesis the only important aspect is the differentiation between an all-round and a north or south field, as this implies the receiver type (see subsection 2.1.3) which directly influences the manipulated variables of the controller (see subsection 2.3.1).

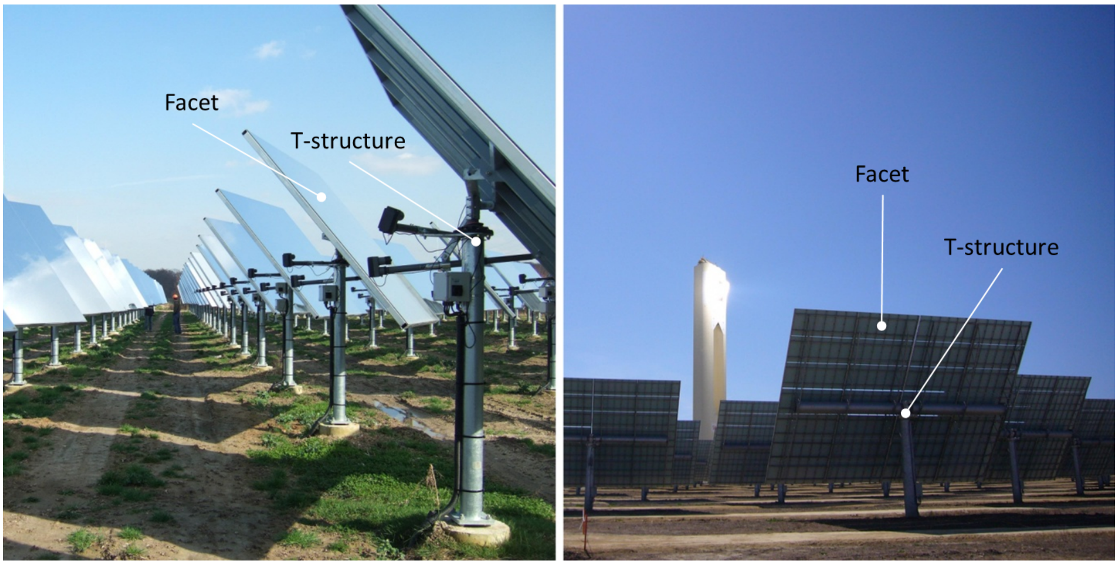


Figure 2.2: Ausra heliostat of the solar tower power plant Jülich with 8.2 m^2 mirror surface (left) and Sanlúcar-120- Heliostat of the PS10 tower power plant (Sanlúcar la Mayor) with 120 m^2 mirror surface (right) [10]

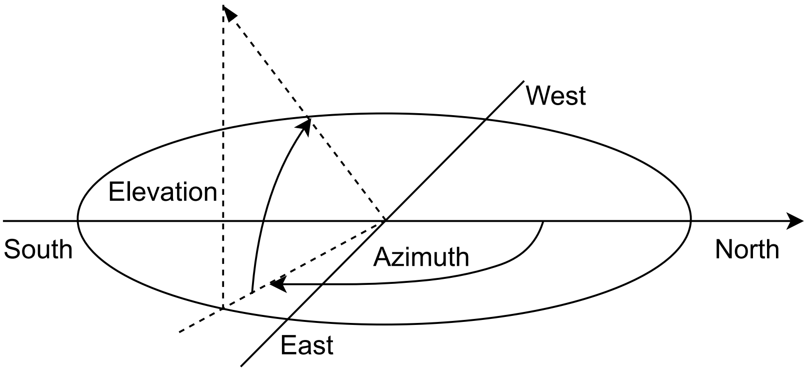


Figure 2.3: Illustration of the elevation and azimuth axis.

2.1.2 Optical Losses

The heliostats as well as their arrangement are suspect to physical limitations and deficiencies, which lower the absorbed energy by the receiver. These limitations and deficiencies, summarized by the word optical losses, influence the shape, aim point and radiation intensity of the projected image onto the receiver. The optical losses include cosine loss, mirror reflectivity, blocking, shading, mirror errors, spillage, tracking error and air attenuation. Each optical loss is briefly explained within the next paragraphs.

Cosine Losses

The solar radiation hits the mirror surface under a certain angle depending on the sun position

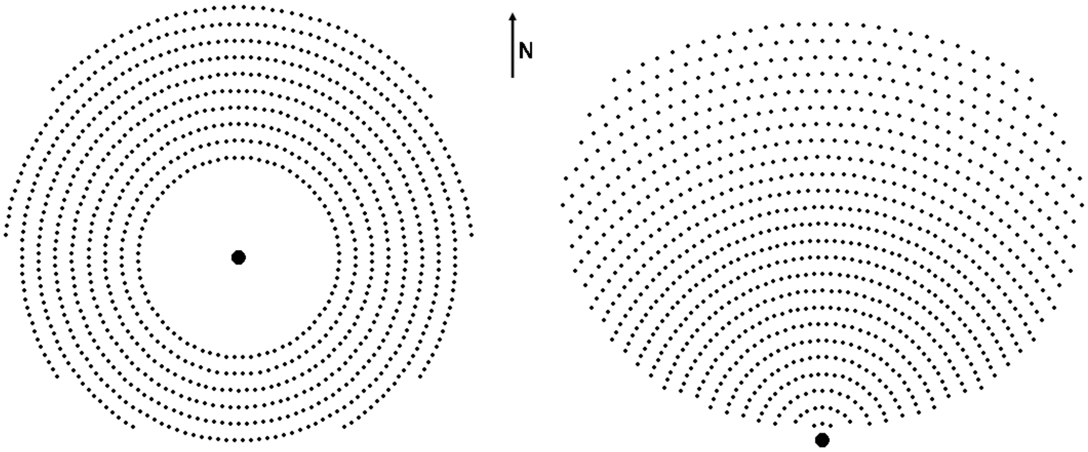


Figure 2.4: Different heliostat field layouts in the Northern Hemisphere: all-round field (left), northern field (right) [10]

and the inclination of the mirror. This angle determines the effective reflection surface. The effective reflection surface is the mirror surface projected perpendicular to the direction of arrival of the solar radiation. The ratio between the effective reflection surface to the mirror surface of the heliostat corresponds to the cosine of the angle between the direction of arrival and of the main normal of the mirror surface as illustrated in Fig. 2.5. Hence, the effective reflection surface decreases with an increasing β and therefore the cosine efficiency which is represented by the term $\cos(\beta)$. Thus, the cosine efficiency is directly related to the solar power reflected by the heliostat. For example, the Solar One Tower in Barstow, California has an annual average cosine efficiency from around 0.65 up to 0.9 depending on the heliostat position [13].

Reflection Losses

Part of the solar radiation is absorbed by the facets. This decreases the reflectivity of the mirror surface, which can be further reduced through environmental influences like rain and

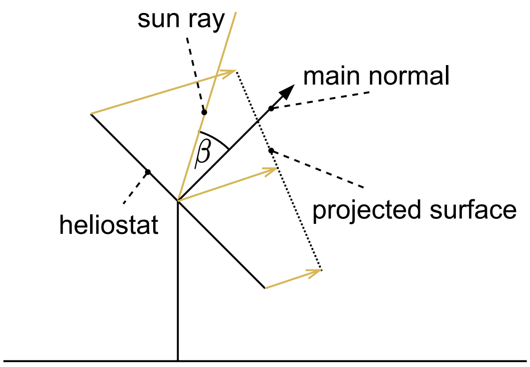


Figure 2.5: Side view of a heliostat illustrating the angle β in the context of cosine loss.

dust. Typical reflection coefficients for cleaned glass facets lie between 0.87 and 0.94 and can drop to 0.8 through environmental influences [10]. The change in reflectivity has a direct influence on the power of the solar tower.

Blocking and Shading

Depending on the sun position, field layout and shadowing objects like the tower, heliostats may be shadowed or block other heliostats as illustrated in Fig. 2.6. In the case of shadowing, an object like a heliostat or the tower disturbs the optical path between the sun and the heliostat so that the solar radiation does not reach the entire heliostat. In the case of blocking, the reflected radiation of one heliostat is partly blocked by another heliostat, which lies within the optical path of the reflection. Shadowing occurs mainly during low sun positions while blocking prevails more in the rear heliostat rows. These losses can be decreased by a proper field design [14].

Mirror Error

The form deviation of a mirror, which leads to slope errors in the mirror, is called mirror error. The mirror error is often noticed by waviness of the mirror material and leads to a wider reflection image of a non-point-shaped sun compared to an ideal mirror. Form deviations may result from manufacturing defects or from stresses within the heliostat structure which also affect the facets. These stresses may be caused by thermal, gravitational or wind deformations. The local mirror error corresponds to the angle between the local normal vector of the mirror and the ideal local normal vector. Usually, a standard deviation of the local mirror error is given and lies typically between 1.5 and 2.5 mrad. [10]

Spillage

Spillage denominates the part of the reflected radiation, which does not hit the receiver and therefore is energy that cannot be fed to the energy conversion process. Spillage is caused for example by the mirror error or due to the reason that the heliostat distance to the receiver is so high that the reflected image onto the receiver exceeds the size of the receiver.

Tracking Error

The tracking error is a deviation in the tracking of a heliostat, which results in an offset of the reflected image onto the receiver. Tracking errors may be caused for example by inaccuracies of the motor aligning the heliostats, dirt or wear at the joints of the tracking axes or imperfections in the construction of the heliostat [15]. Typically, the tracking error differs from 0.5 to 2 mrad [10] and is distributed in a non-Gaussian way [15].

Air Attenuation

The reflected radiation is weakened while it covers the distance between the heliostat and the receiver due to absorption and scattering processes in the air. The weakening depends in particular on the distance between heliostat and receiver but also on the height of the heliostat field above sea level, the local humidity and the vertical distance covered. Under unfavorable conditions the attenuation at a distance of 1000 m between receiver and heliostat can be up to 1.2 %. [10]

2.1.3 Receiver

The receiver absorbs the reflected radiation by the heliostats and conducts the stored energy within the receiver material to a heat transfer medium via convection. The heat transfer medium is then fed to an energy conversion process. Additionally, due to its physical

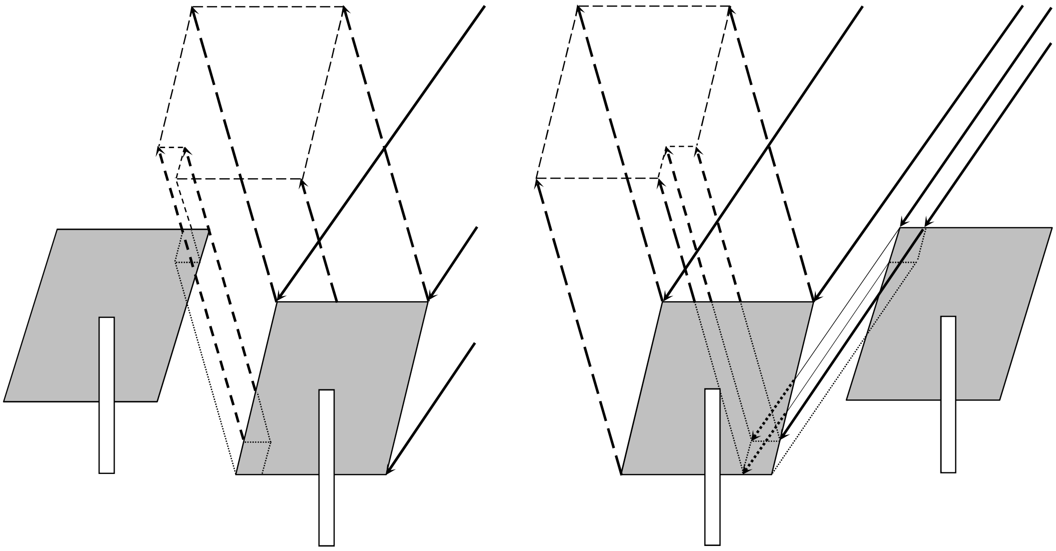


Figure 2.6: Blocking (left) and shading (right) of a heliostat. [10]

limitations it imposes restrictions on the intensity of the radiation. The next two paragraphs describe the composition of the receiver as well as its physical limitations, which result in an allowed flux density.

Composition

The receiver is located in the top part of the tower. It has a rectangular or a cylindrical shape depending on the field layout. Rectangular receivers are used in north or south fields while circular receivers are used in all-round fields. A modification of a rectangular receiver is a receiver that is curved inwards to protect it from wind. The purpose of the receiver is to absorb the solar radiation reflected by the heliostats, conduct the heat to a heat transfer medium and transport it to an energy conversion process. The heat transfer medium may be air, molten salt, water, liquid metals or particles [16]. Depending on the transfer medium, the receiver surface consists of different materials such as metallic tubes for molten salt or porous ceramics for air [17]. In thermal receivers, the energy conversion process may represent a conventional power plant process (e.g. Rankine cycle), while in thermochemical receivers, the heat will be used for a reaction (e.g. hydrogen production).

Radiation Physical Quantities

The radiant flux F is defined as the radiant energy Q emitted or absorbed per unit time:

$$F = \frac{\mathrm{d}Q}{\mathrm{d}t} \tag{2.1}$$

The flux density ϕ is the radiant flux Φ per area $\mathrm{d}A$ and can be expressed as

$$\phi = \frac{\mathrm{d}\Phi}{\mathrm{d}A} \tag{2.2}$$

In order to determine the gross power that is fed to the energy conversion process the flux density is measured on each point on the receiver aperture. The power P can then be calculated by integrating the flux density ϕ over the receiver surface A:

$$P = \int_A \phi \, \mathrm{d}A \tag{2.3}$$

Since the measurement has no infinite fidelity, the flux density is usually given with respect to a certain area called bin size. A sample flux density distribution is shown in Fig. 2.7.

Allowed Flux Density

If characteristics, such as the corrosion limit as well as the maximum allowed thermal stresses of the receiver are exceeded, the receiver can be damaged [18]. For example, for a molten salt tube receiver these characteristics are related to the local salt temperature and velocity as well as the wind velocity [19]. However, the local salt temperature is influenced by the flux density on the receiver.

Damage to the receiver material must be prevented. Therefore, a maximum allowed flux density for the receiver aperture is set, which can be calculated based on the receiver characteristics. For example, for tube receivers that are cooled with liquid sodium, maximum radiant flux densities of 2.5 MW/m^2 are allowed, while with air-cooled tube receivers only 200 kW/m^2 [10] can be achieved. There may also exist a minimum allowed flux density for example due to the problem of freezing in molten salt receivers as well as restrictions to the flux gradient, which will not be considered in this thesis. Therefore, the allowed flux density always denotes the maximum allowed flux density within this work.

While the thermal site of the receiver is a major aspect in solar power tower plants and a research area on its own, it is only implicitly important to this thesis because it yields the allowed flux density. Within this work, it is assumed that the thermal site remains static during the control process, which means that the allowed flux density is considered static, too.

2.1.4 Sun Shape

The sun shape has an indirect role in the control system. Equivalent to the mirror error (see Fig. 2.1.2) it influences the shape of the flux density profile as well as its intensity.

Due to the fact that the sun is not a point source and does not emit parallel rays of light, the angle between a sun ray and the facet of a heliostat may differ from sun ray to sun ray [20]. An observer on earth will note a decrease of the intensity of the sun rays when looking further away from the center of the sun. This intensity profile which is dependent on the sun angle α between the observed direction and the direction to the center of the sun is called sun shape. The angle is illustrated in Fig. 2.8. A sun ray may be scattered on particles in the atmosphere. If it is scattered multiple times, it still reaches the observer, but from a different angle than that from which it was originally emitted. Thus, an observer can recognize sun rays even when not looking directly into the sun. Up to an angle of 4.6 mrad the observer is looking directly in the sun [10]. The radiation, which can be recognized above this angle is called circumsolar radiation. The ratio between circumsolar radiation $E_{aureole}$ and the sum of

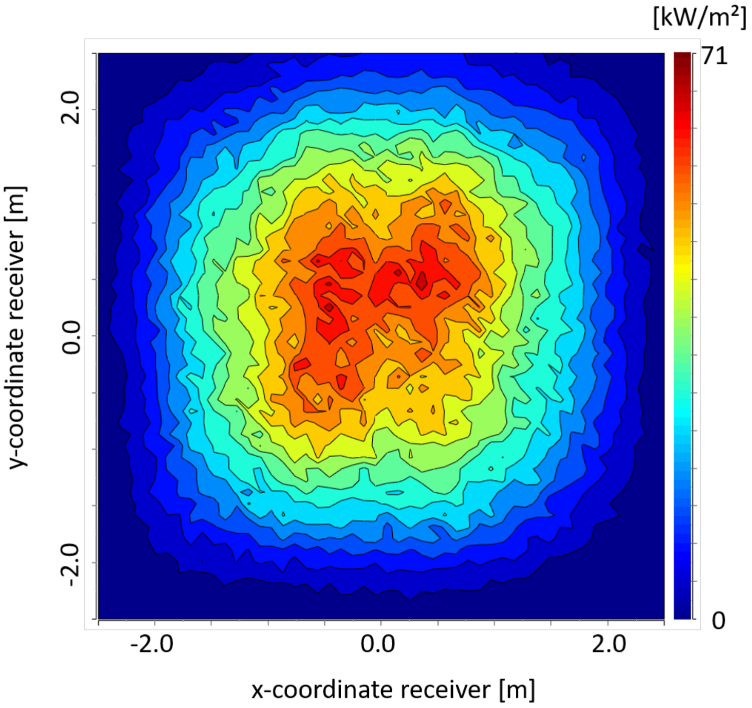


Figure 2.7: Flux density distribution of a simulated plant with 100 heliostats with 8 m^2 mirror surface and a mirror error of 3 mrad.

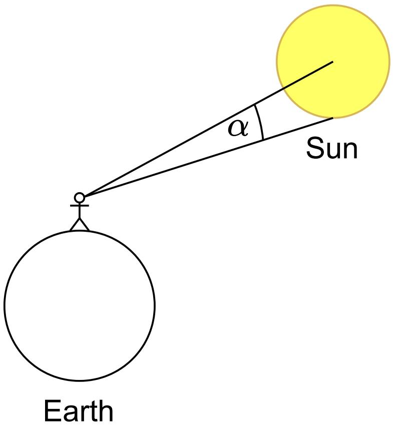


Figure 2.8: Illustration of the sun angle α in context of the sun shape.

the circumsolar radiation and the radiation of the solar disk E_{disk} is called circumsolar ratio CSR, which can be expressed by

$$CSR = \frac{E_{aureole}}{E_{aureole} + E_{disk}} \tag{2.4}$$

Neumann et al. [21] classified different measured sun shape distribution into six groups of different CSR-ranges. Averaged distributions were then calculated from the sun shape distributions of the respective groups, which are depicted in Fig. 2.9.

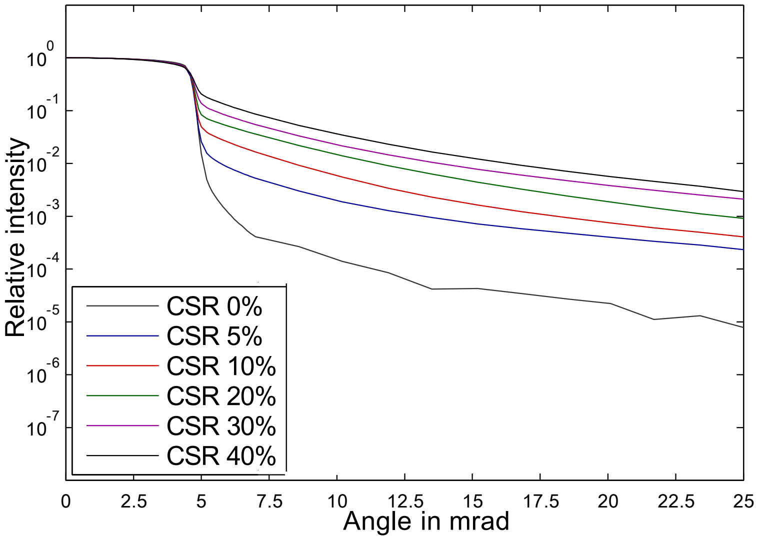


Figure 2.9: Sun shape distributions with different circumsolar components [21].

These representative sun shape distributions often serve as the basis for the simulation of solar radiation in simulation programs for concentrating solar systems [10].

2.2 Control Theory

The objective of a controller is generally to adjust certain variables, usually output variables of technical processes, so that they are equal to predefined reference variables [22]. The variables to be controlled should follow changes in the reference variables as closely as possible and should be influenced as little as possible by disturbances affecting the process.

Controllers can be distinguished between closed- and open-loop controllers. In a closed-loop control system, the controlled variables $\mathbf{y}$ are measured by a sensor and the measurement results $\mathbf{y_m}$ are compared with the reference variables $\mathbf{y}_{\mathrm{ref}}$. From the differences $\mathbf{e}$ between the reference and controlled variables, control actions $\mathbf{u}$ (manipulated variable) are derived, which are suitable for reducing the differences. This feedback of output variables defines the closed-loop control and is illustrated in Fig. 2.10a. In the context of closed-loop control, a distinction is made between a force tracking a disturbance rejection control. The former tries to follow a given reference trajectory as fast as possible, while the latter attempts to maintain a reference value under the influence of disturbances.

The open-loop control is defined by the absence of the feedback. The input variables to the system are only dependent of the reference variables and the control algorithm as shown in Fig. 2.10b. In contrary to the closed-loop control, the controlled variable is not measured. Disturbances or errors in the system model can not be compensated by this type of control. However, if the disturbances can be measured or estimated, the open-loop control can be extended to a feed forward control as illustrated in Fig. 2.10b. A feed forward control accounts for disturbances, and adjusts the manipulated variable so that deviations in the controlled variable from the reference point are minimized. Thus, the controller can reject disturbances before they affect the controlled variable.

In order to design a controller, knowledge about the system and its characteristics such as stability is usually essential. To derive these characteristics, the system needs to be described mathematically.

As will be shown in subsection 3.1.4, the system treated in this thesis is a non-linear time-invariant multiple-input multiple-output (MIMO) system. Nonlinear MIMO-systems can generally be described by a nonlinear state space representation consisting of a vector differential equation:

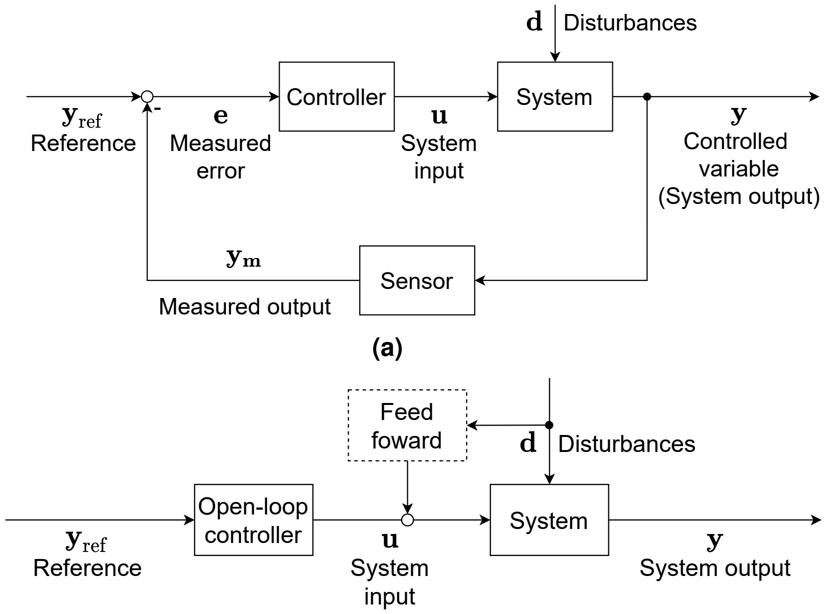
$$\dot{\boldsymbol{x}} = \boldsymbol{f}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.5}$$

and an output equation:

$$\boldsymbol{y} = \boldsymbol{g}(\boldsymbol{x}, \boldsymbol{u}) \tag{2.6}$$

Here, $\mathbf{x}$ is the n-dimensional state vector, $\mathbf{u}$ the m-dimensional control or input vector, $\mathbf{f}$ the n-dimensional vector function of the system, $\mathbf{y}$ the r-dimensional output vector and $\mathbf{g}$ the r-dimensional output vector function.

A non-linear system imposes difficulties on the design of the controller because an overall stability can usually not be guaranteed but only a stability within an environment of an equilibrium point. An equilibrium point is defined as the state $\mathbf{x}$ where $\dot{\mathbf{x}} = 0$. In contrast to a linear system, there may exist more than one equilibrium point. The equilibrium point of a non-linear system is defined as stable when all trajectories $\mathbf{x}(t)$, which start within an environment of the equilibrium point, run in this point with increasing time and stay there [23]. A MIMO-system imposes additional difficulties because the inputs and outputs may be coupled, which means that one input can influence several outputs. Thus, either several inputs need to be adjusted or it is impossible to change one output while keeping the others constant. Furthermore, in systems theory most of the analysis possibilities are limited to linear single-input single-output (SISO) systems, since the theory in this area is much simpler and more advanced than that of nonlinear MIMO-systems.


$\mathbf{d}$ Disturbances
Controller
System
$\mathbf{y}_{\text{ref}}$ Reference
-
$\mathbf{e}$ Measured error
$\mathbf{u}$ System input
$\mathbf{y}$ Controlled variable (System output)
$\mathbf{y_m}$
Measured output
Sensor
Feed foward
$\mathbf{d}$ Disturbances
Open-loop controller
System
$\mathbf{y}_{\text{ref}}$ Reference
$\mathbf{u}$ System input
$\mathbf{y}$ System output

(a)

(b)

Figure 2.10: Closed-loop (a) and open-loop or feed forward control (b) scheme.

2.3 Existing Heliostat Aiming Control Strategies

One goal when optimizing solar power tower plants is to increase the efficiency in order to make the plant economically more attractive. The efficiency can be raised in many ways such as optimizing the field layout, the receiver design or the aim point coordinates of each heliostat [19]. The latter called *heliostat aiming* is subject to this thesis.

As stated in chapter 1, the objective of the controller is to minimize the spillage, while not violating the allowed flux density and also be robust to disturbances. After a comprehensive literature review, it was found that most developed controllers are open-loop controllers, which face the first two requirements of minimizing the spillage and not violating the constraints. However, only a few closed-loop controllers were developed, which cover also the rejection of disturbances.

Most of the open-loop controllers, try to solve the heliostat aim point optimization problem, which is explained in the next subsection. The understanding of this problem, is also crucial to understand the controllers, which are developed later in this thesis. The proposed controllers in literature are categorized and presented in the last two subsections.

2.3.1 Heliostat Aim Point Optimization

Optimal heliostat aiming is achieved when the flux density at each bin on the receiver is equal to the allowed flux density at the respective bin. However, one heliostat correlates with many bins on the receiver. The flux density on one bin can not be adjusted without changing the flux density on other bins. Thus, it may be impossible to find a solution, where each bin value is equal to the allowed flux density. Therefore, the objective when optimizing the heliostat aiming is usually to maximize the intercepted power P_{receiver} of the receiver and thus reduce the spillage while not violating the restrictions (allowed flux density) at each measurement bin. This can be formulated as an optimization problem as follows

$$\begin{aligned} &\text{maximize} \quad P_{\text{receiver}} \\ &\text{subject to} \\ &\qquad \phi_i \leq \phi_{\max}, \quad \forall i = 1, \ldots, n_{\text{bins}}, \\ &(x_{\text{a},h}, y_{\text{a},h}) \in \mathbb{R}^2, \qquad \forall h = 1, \ldots, n_{\text{heliostats}} \end{aligned} \tag{2.7}$$

where ϕ_i is the flux density at bin i, $\phi_{\max,i}$ the respective allowed flux density and $(x_{\text{a},h}, y_{\text{a},h})$ the aim point coordinates for heliostat h. The power P_{receiver} can be calculated by multiplying the flux density (Eq. 2.2) with the bin area and sum them up for each bin:

$$P_{\text{receiver}} = \sum_i \phi_i \cdot \mathrm{d}A \tag{2.8}$$

Because the bin area is constant and usually equal for every bin, the formulation in Eq. 2.7 is equivalent to

$$\begin{aligned} &\text{maximize} \quad \sum_i \phi_i \\ &\text{subject to} \\ &\qquad \phi_i \leq \phi_{\max}, \quad \forall i = 1, \ldots, n_{\text{bins}}, \\ &(x_{\text{a},h}, y_{\text{a},h}) \in \mathbb{R}^2, \qquad \forall h = 1, \ldots, n_{\text{heliostats}} \end{aligned} \tag{2.9}$$

The optimization problem can be formulated continuous or discrete. While the continuous formulation (Eq. 2.9) allows all possible aim point coordinates ($(x_{\text{a},h}, y_{\text{a},h}) \in \mathbb{R}^2$), the discrete problem specifies a subset A of possible aim point coordinates usually arranged in a grid. Thus, the discrete optimization problem can be mathematically expressed as

$$\begin{aligned} &\text{maximize} \quad \sum_i \phi_i \\ &\text{subject to} \\ &\qquad \phi_i \leq \phi_{\max}, \quad \forall i = 1, \ldots, n_{\text{bins}}, \\ &(x_{\text{a},h}, y_{\text{a},h}) \in A, \qquad \forall h = 1, \ldots, n_{\text{heliostats}} \end{aligned} \tag{2.10}$$

Although the discrete problem can reduce the solution space vastly, the problem can probably not be solved in polynomial time and thus belongs to the class of the so-called NP-hard problems [16]. NP stands for non-deterministic polynomial-time. A NP-hard problem is at least as “hard” as all NP problems. This means that an algorithm that solves an NP-hard problem can be used with the help of a reduction to solve all NP problems. It is unlikely that such an algorithm exist. In order to lower the spillage the heliostats aim preferably at the center (or centerline for a cylindrical receiver) but pointing all at the center usually exceeds the allowed flux density. Thus, the heliostats have to be moved while keeping the spillage low. Which heliostat to move and in which direction depends on the location of the heliostat (assuming all heliostats are identical) as well as the type of the receiver. Heliostats, which have a smaller distance to the receiver, spread the solar radiation over a smaller area on the receiver. Since they reflect almost the same power but on a smaller area as heliostats that are further away, closer heliostats achieve higher maximum flux densities. Hence, when moving closer heliostats to the edge of the receiver the spillage is increased less due to the smaller projected image onto the receiver. Additionally, the flux density on one bin on the receiver can be reduced more.

It should be noted that an optimal solution to the formulated problem does not represent the optimal solution with respect to the real system because the plant is subject to disturbances. The optical losses described in subsection 2.1.2 and clouds are considered as disturbances. They alter the flux density distribution and therefore, can increase and/or decrease the intercepted power. For example, a tracking error may result in a violation of the allowed flux density when it induces a shift to a bin with a high flux density. Clouds reduce the reflected radiation, which offers the possibility to shift the aim points of the heliostats more to the center. When the cloud passes, the radiation increases, which then can lead to a violation of the allowed flux density in the center.

2.3.2 Open-Loop Controllers

Solving the heliostat aim point optimization problem requires high computational effort due to its many inputs (aim point coordinates) and outputs (measurement bins on the receiver) as well as its strong correlation between them. Therefore, most controllers in the literature are concerned with the problem of finding aim point coordinates for each heliostat which yield a high intercept power while not exceeding the allowed flux density.

Heuristic Techniques

Vant-Hull [19] proposes an algorithm for cylindrical receivers, which allocates an assumed beam radius for the reflected image to each heliostat. Half of the heliostats are placed so that the beam radius corresponds to the distance between the upper rim of the receiver and the aim point. The other half is placed with respect to the lower rim. The beam radius is deduced from a calculated cone angle of each heliostat. This cone angle can be multiplied by a constant factor (the *aiming strategy parameter*) to center the reflected image more. An optimal aiming strategy parameter, which decreases the spillage and does not violate the restrictions, has to be found. This algorithm was further modified by Flesch et al. [24] and Sánchez-González et al. [25] as well as by Collado and Guallar [26]. Flesch et al. [24] added a product to the factor for the cone angle. The product consists of a new parameter and the distance between heliostat and center of the receiver called slant range. The new parameter scales the punishment for the slant range. Sánchez-González et al. [25] divide the all-round field for a cylindrical receiver into different sectors in circumferential direction and propose an algorithm which finds suitable aiming strategy parameters for each sector. A similar approach is implemented by Collado and Guallar [26] who divides the all-round field in three zones in radial direction and allocates to zone one and two one aiming strategy parameter and to zone three another. Thus, the proposed algorithm fits two parameters.

Astolfi et al. [4] implemented different optimization approaches for a cylindrical receiver. All approaches are based on branching the optimization problem in simpler sub-problems.

Local Search

Maldonado et al. [27] present an iterative algorithm, which they call local search. The controller examines in each step the neighborhood of the aim point for a single heliostat. In each examination step, the algorithm evaluates the shifts to the neighbouring aim points with respect to the overall receiver performance. Then, the heliostat aim point assignment is changed to the shift with the highest increase in the receiver performance. If no shift increases the performance, the initial solution is maintained.

Metaheuristic Techniques

Metaheuristic techniques to solve the discrete optimization problem are suggested by Salomé et al. [28] using the TABU algorithm, Besarati et al. [29] applying a genetic algorithm as well as Belhomme et al. [30] utilizing an ant colonization algorithm. The ant colonization algorithm is explained in section 2.4 in more detail.

MILP

By reformulating the discrete optimization problem stated in Eq. 2.10 the optimization problem can be solved by mixed-integer linear programming (MILP). This was done by Ashley et al. [31], who implemented a MILP algorithm to find an optimal aim point distribution. The algorithm was also successfully tested under the influence of clouds. However, to consider clouds a prediction of these is necessary for the control.

Richter et al. [15] present a MILP formulation based on Γ-robustness. The aiming strategy directly considers the uncertainty of the tracking error to improve the efficiency while maintaining a high degree of safety. In contrary to other methods, it also constrains the flux gradients.

Summary

All proposed controllers rely on approximations of the heliostat errors and have rather high numerical costs (except the Vant Hull and modified Vant Hull algorithm). If the approximation is not accurate enough or the numerical costs lead to a high computational time, the controller is not suitable for real world applications. At least without other systems compensating these problems. Only the algorithm by Richter et al. [15] may compensate inexact modeling of the errors but still has rather high numerical costs. Tt needs around 60 s to find a good solution for a plant of 624 heliostats [32]. However, commercial plants consist of usually 10^4 heliostats and require a sample time of only a few seconds to react fast enough to critical situations like overflux conditions. Furthermore, only some controllers consider clouds (although all can be modified to work with clouds). However, getting nearly optimal solutions would rely on a very precise prediction of the clouds, which is currently not the state of the art. To consider inaccurate approximations or disturbances like clouds, a closed-loop control would be a solution.

2.3.3 Closed-Loop Controllers

While there exist many open-loop controllers, the choice of closed-loop controllers is rather small so far.

Heuristic Controllers

Vant-Hull [19] proposes a simple control to prevent overflux conditions. When overflux is measured at one bin, the heliostat producing the greatest flux on the respective bin is identified and pointed to an aim point outside the receiver surface.

GarcÍa-MartÍn et al. [1] present an algorithm based on measuring the temperature at different points on a rectangular receiver surface and shifting the aim points of the heliostat to colder zones of the receiver when the maximum allowed temperature is exceeded.

PID

A controller based on decoupling the system into single-input single-output (SISO) subsystems and then applying a Proportional Integral Derivative (PID) controller to these subsystems is implemented by García et al. [33]. Each subsystem consists of a group of heliostats with two manipulated variables. As manipulated variables they introduce the shifting of the assigned aim points of the groups as well as the dispersion and concentrating of aim points within a group. Six PID controllers are needed for the plant system because they divided the field into three groups with two manipulated variables each. The control is designed for a cylindrical receiver.

MPC

García et al. [34] use a multivariable model predictive control (MPC) approach for cylindrical receivers. Instead of controlling single heliostats they control groups of heliostats, using the same manipulated variables (shifting and dispersion) as presented in the previous paragraph. In [35] the same authors modify the MPC to also work with clouds. To overcome the problem of an overshoot of the flux density after the cloud passes, a readjustment approach is implemented, which is supported by a PI controller.

Summary

Except the heuristic controllers, the controllers are only designed for cylindrical receivers so far. Although, the heuristic controllers can handle both receiver types (cylindrical and rectangular), they can only handle overflux conditions or base on the measurement of temperatures instead of flux densities.

2.4 Ant Colony Optimization

In this section the ant colony optimization is explained, which is used by the Static Optimal Control described in section 3.6. The ant colony optimization is an algorithm, which imitates the behavior of ant colonies during foraging. The objective is to find the optimal path along a route with different junctions. In every iteration, several ants try to find the optimal path, which is represented by an aim point configuration. The possible paths an ant can take is defined by the number of heliostats n_h and aim points n_a. Because the numbers n_h and n_a are finite, the algorithm solves a combinatorial problem i.e. it solves the discrete optimization stated in Eq. 2.10. An example, for a path is shown in Fig. 2.11, in which the ant chooses an aim point configuration where heliostat 1 and heliostat 5 aim at aim point 2, heliostat 2 and 3 at aim point 3 and heliostat 4 at aim point 1. The ant starts with choosing an aim point for heliostat 1 and then, can choose at every node an aim point j for the next heliostat i, which is equal to choosing one of the nodes of the next column.

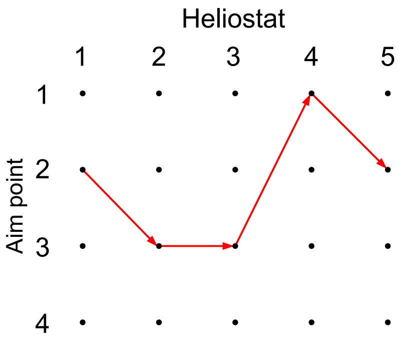


Figure 2.11: Sample aim point configuration for a plant with five heliostats and four aim points.

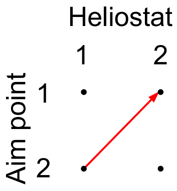
Transition Rule

The decision, which edge s_{ij} the ant takes, can be expressed through a so-called transition rule, which is given by

$$s_{ij} = \begin{cases} \max\left\{\tau_{ij}(t) \cdot \eta_{ij}^{\beta}\right\} & \text{if } q \leq q_0 \\ \text{Monte-Carlo choice} & \text{else} \end{cases} \tag{2.11}$$

Here, τ_{ij} is the so-called pheromone value, η_{ij} the so-called visibility value, β the weighting factor, q a random number and q_0 a decision parameter.

The edge s_{ij} which is chosen, is determined by the equally distributed random number $q \in [0, 1]$. If the number q is greater than or equal to the parameter q_0, the edge is determined according to Eq. 2.11. Hence, the edge is chosen which has a high pheromone amount and at the same time a high visibility value weighted by the parameter β. For example, in Fig. 2.12 the ant would choose the edge $s_{2,1}$ due to the higher pheromone value on this node.



$$\tau_{2,1} = 0.8 \; \eta_{2,1} = 0.5 \Rightarrow s_{2,1} = 0.8 \cdot 0.5^1 = 0.4 \; p_{2,1} = 0.8$$

$$\tau_{2,2} = 0.2 \; \eta_{2,2} = 0.5 \Rightarrow s_{2,2} = 0.2 \cdot 0.5^1 = 0.1 \; p_{2,2} = 0.2$$

Figure 2.12: Illustration of the transition rule with $\beta = 1$.

The pheromone values for the heliostat aim point combinations are updated in each iteration by the equation for the *pheromone update*, which is explained later in this subsection.

The visibility value η_{ij} is a constant and expressed by the intercept factor of heliostat i aiming at aim point j. The intercept factor $f_{\text{intercept}}$ is defined by the power intercepted by the receiver P_{receiver} with respect to the reflected power of the heliostats $P_{\text{heliostats}}$ and ranges from 0 to 1:

$$f_{\text{intercept}} = \frac{P_{\text{receiver}}}{P_{\text{heliostats}}} \tag{2.12}$$

If the random number q is less than the parameter q_0, the edge is selected by a Monte Carlo selection, where the probabilities p_{ij} of the edges are determined according to Eq. 2.13.

$$p_{ij}(t) = \frac{\tau_{ij}(t) \cdot \eta_{ij}^{\beta}}{\sum_l \tau_{il}(t) \cdot \eta_{il}^{\beta}} \tag{2.13}$$

For example, in Fig. 2.12, there is a higher chance that the edge $s_{2,1}$ is chosen due to the higher pheromone value at this node.

To be able to reasonably adjust the weighting factor β, the pheromone value should be of the same magnitude as the weighting factor i.e. around one. Furthermore, both values must be greater or equal to zero. Otherwise, the calculation of the probabilities is illogical.

Choosing a relatively high value for q_0 leads to a search with a more local and greedy character due to the lesser occurrence of randomly chosen nodes. In contrary, a smaller q_0 intensifies the global search for solutions so that local optima are left again and the possibility of stagnation is reduced.

Pheromone Update

In addition to q_0, the pheromone update also intensifies the local as well as the global search for an optimal solution. Consequently, it is divided into a global and a local update. Both updates are based on Eq. 2.14, but there are differences in the added pheromone amount $\Delta\tau_{ij,k}$.

$$\tau_{ij}(t+1) = (1-\rho) \cdot \tau_{ij}(t) + \rho \cdot \Delta\tau_{ij,k}(t) \tag{2.14}$$

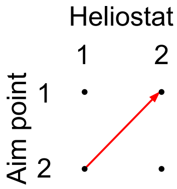
The constant ρ quantifies the relative amount of pheromone that evaporates i.e. vanishes.

The global update is influenced only by the ant k which has the highest objective value in the current iteration. The amount of the secreted pheromone $\Delta\tau_{ij,k}$ corresponds to the value of the objective function J_k for the best ant:

$$\Delta\tau_{ij,k}(t) = J_k \tag{2.15}$$

The global update increases the pheromone value on every node passed by the best ant. This increases the chance that these nodes are chosen by other ants in the next iteration. Thus, the global update pushes the virtual ants to search in the environment of the current, global best path. An example for a global update is given in Fig. 2.13.

J_k must be greater than zero, because the best ant should lead to an increase in the pheromone value. Furthermore, it is desired that the search starts from the current aim point configuration. Therefore, the pheromone values of the nodes of the current aim point



Global update for best ant 1 with $J_1 = 1$

$$\tau_{1,2}(t) = 0.8 \Rightarrow \tau_{1,2}(t+1) = (1-0.5)\cdot 0.8 + 0.5 \cdot 1 = 0.9$$

$$\tau_{2,1}(t) = 0.6 \Rightarrow \tau_{2,1}(t+1) = (1-0.5)\cdot 0.6 + 0.5 \cdot 1 = 0.8$$

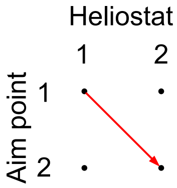
Figure 2.13: Global pheromone update with $\rho = 0.5$.

configuration are updated using the global update before the first iteration. If the objective value of the current aim point configuration would be less or equal to zero, the pheromone values would be decreased or left unchanged. Both cases would not lead to a search in the direction of the current aim point configuration. Additionally, to keep the pheromone value in the magnitude of one, the objective value should be in the same magnitude i.e. around one.

In contrary to the global update, the local update updates the pheromone values of the passed nodes for all ants from the current iteration. The value is updated as described in Eq. 2.14 with $\Delta\tau_{ij,k}$ for ant k given by

$$\Delta\tau_{ij,k} = \tau_0, \tag{2.16}$$

where τ_0 denotes the pheromone start value of the optimization. An example for a local update is given in Fig. 2.14.



Local update for ant 2

$$\tau_{1,1}(t) = 0.8 \Rightarrow \tau_{1,1}(t+1) = (1-0.5)\cdot 0.8 + 0.5 \cdot 0.2 = 0.5$$

$$\tau_{2,2}(t) = 0.2 \Rightarrow \tau_{2,2}(t+1) = (1-0.5)\cdot 0.2 + 0.5 \cdot 0.2 = 0.2$$

Figure 2.14: Local pheromone update with $\rho = 0.5$ and $\tau_0 = 0.2$.

Unlike the natural model, the local update leads to a reduction of the pheromone amount on the respective node. Thus, the attractiveness of used nodes is reduced, which forces the search in different directions to lower the risk of stagnation. It is worth noting that the update never reduces the pheromone amount to a value less than the pheromone start value. Thus, there is always a possibility that a node can randomly be chosen under the condition that the respective visibility value for this node is not zero.

Overall, the pheromone traces act as a collective memory. Thus, during the optimization process, not only the previous best solution is stored, but also other good solutions, in whose environment the search is intensified.

3 Modelling of the System and the Controllers

The first section of this chapter establishes a mathematical model for the flux density distribution on the receiver of the solar power tower plant and investigate the dynamics of the heliostats and the receiver and other properties of the system. Then, the properties of the controller such as the in- and outputs as well as the sample time of the flux density measurement are determined. Subsequently, the possible controllers for the solar power tower plant are presented and suitable control strategies are chosen. The three chosen controllers are elaborated in the following sections. Lastly, a short presentation of the used simulation software as well as its enhancements are given.

3.1 System Modelling and Analysis

In order to design a suitable controller, knowledge about the plant system is essential for most controllers. This section describes an analytical model from literature for the flux density computation on the receiver and investigates the dynamics of the heliostat movement as well as the dynamics of the heating of the receiver. Finally, the properties are analyzed that are relevant for the controller design.

3.1.1 Analytical Model

To calculate the flux density distribution on the receiver surface from one heliostat reflecting the solar radiation on a given aim point, the HFLCAL model is used in this section. It is an analytical model developed by Kiera [36] and further modified by Schwarzbözl et al. [37]. The HFLCAL model is chosen because it is commonly used as in [15, 24, 26, 29, 34].

The flux calculation of the HFLCAL model is based on a bivariate normal distribution and is calculated at a given point (x, y) on the receiver for a heliostat h aiming at an aim point (x_a, y_a) as follows:

$$F_{x,y}^{(h)} = \frac{P_H \cos \omega_p}{2\pi\sigma_{HF}^2} e^{-\frac{(x-x_a)^2+(y-y_a)^2}{2\pi\sigma_{HF}^2}}, \tag{3.1}$$

where P_H denotes the power reflected by the heliostat, σ_{HF} is the total effective deviation and ω_p represents the angle between the normal vector of the receiver and the heliostat.

The reflected power P_H can be expressed through

$$P_H = I_D \cdot A_H \cdot \cos\beta \cdot f_{at} \cdot \rho \cdot f_{bs}, \tag{3.2}$$

where I_D is the direct normal sun irradiation, A_H the mirror area, β denotes the angle between the normal vector of the heliostat and the vector of the sun rays radiating onto the heliostat (see Fig. 2.5 in subsection 2.1.2), f_{at} represents the air attenuation factor, ρ the mirror reflectivity and f_{bs} the blocking and shading factor (see subsection 2.1.2).

The total effective deviation σ_{HF} is derived from the convolution of three Gaussian distributions [35].

$$\sigma_{HF}^2 = s^2 \cdot [\sigma_{sun}^2 + 2 \cdot (1 + \cos\beta) \cdot \sigma_{slp}^2 + \sigma_t^2], \tag{3.3}$$

where σ_{sun} represents the deviation caused by the sun shape (see subsection 2.1.4), σ_{slp} and σ_t take the mirror and the tracking error (see subsection 2.1.2) into account. s denotes the distance between the heliostat and the receiver center, also called slant range.

Based on this model a plant system with multiple heliostats and measurement points can be mathematically formulated. If the velocity of the aim point movement $(\dot{x}_a, \dot{y}_a)$ is considered as inputs instead of the aim point coordinates (x_a, y_a), a state space representation for a simple plant with two heliostats and two measurement points can be formulated. The state space representation allows for a standard description of the system and thus makes it easier to deduce characteristics such as the stability and eigendynamic of a system.

In order to represent Eq. 3.1 in a more compact way, it can be reformulated as

$$F_{x,y}^{(h)} = a^{(h)} e^{-\frac{(x-x_a)^2+(y-y_a)^2}{b^{(h)}}} \tag{3.4}$$

with

$$a^{(h)} = \frac{P_H \cos\omega_p}{2\pi\sigma_{HF}^2} \tag{3.5}$$

$$b^{(h)} = 2\pi\sigma_{HF}^2 \tag{3.6}$$

For reasons of simplicity the angle ω_p is considered static. Then the derivative of the flux F can be expressed as follows

$$\dot{F}_{x,y}^{(h)} = -2a^{(h)}/b^{(h)} \cdot [(x - x_a)\dot{x}_a + (y - y_a)\dot{y}_a] e^{-\frac{(x-x_a)^2+(y-y_a)^2}{b^{(h)}}} \tag{3.7}$$

which can be further simplified with Eq. 3.4 to

$$\dot{F}_{x,y}^{(h)} = -2/b^{(h)} \cdot [(x - x_a)\dot{x}_a + (y - y_a)\dot{y}_a] F_{x,y}^{(h)} \tag{3.8}$$

Based on the preceding equations, a system of equations for a solar power tower plant with two heliostats and two measurement points can be expressed by

$$\begin{aligned}
&\dot{x}_1 = u_1 && \dot{x}_a \text{ for } h_1 \text{ (heliostat 1).}\\
&\dot{x}_2 = u_2 && \dot{y}_a \text{ for } h_1.\\
&\dot{x}_3 = u_3 && \dot{x}_a \text{ for } h_2.\\
&\dot{x}_4 = u_4 && \dot{y}_a \text{ for } h_2.\\
&\dot{x}_5 = -\frac{2}{b_5} \cdot [(c_1 - x_1)u_1 + (r_1 - x_2)u_2]x_5 && \text{Flux density change } h_1 \text{ point } (c_1, r_1).\\
&\dot{x}_6 = -\frac{2}{b_6} \cdot [(c_2 - x_1)u_1 + (r_2 - x_2)u_2]x_6 && \text{Flux density change } h_1 \text{ point } (c_2, r_2).\\
&\dot{x}_7 = -\frac{2}{b_7} \cdot [(c_1 - x_3)u_3 + (r_1 - x_4)u_4]x_7 && \text{Flux density change } h_2 \text{ point } (c_1, r_1).\\
&\dot{x}_8 = -\frac{2}{b_8} \cdot [(c_2 - x_3)u_3 + (r_2 - x_4)u_4]x_8 && \text{Flux density change } h_2 \text{ point } (c_2, r_2).\\
\\
&y_1 = x_5 + x_7 && \text{Total flux density point } (c_1, r_1).\\
&y_2 = x_6 + x_8 && \text{Total flux density point } (c_2, r_2).
\end{aligned} \tag{3.9}$$

These equations can be restructured to yield the state space representation:

$$\dot{\mathbf{x}}(t) = \mathbf{A}\,\mathbf{x}(t) + \mathbf{B}(\mathbf{x})\,\mathbf{u}(t) \tag{3.10}$$

$$\mathbf{y}(t) = \mathbf{C}\,\mathbf{x}(t) + \mathbf{D}\,\mathbf{u}(t) \tag{3.11}$$

with

$$\begin{aligned}
\mathbf{A} &= \mathbf{0}\\
\mathbf{B} &= \begin{bmatrix}
1 & 0 & 0 & 0\\
0 & 1 & 0 & 0\\
0 & 0 & 1 & 0\\
0 & 0 & 0 & 1\\
-2/b_5(c_1 - x_1)x_5 & -2/b_5(r_1 - x_2)x_5 & 0 & 0\\
-2/b_6(c_2 - x_1)x_6 & -2/b_6(r_2 - x_2)x_6 & 0 & 0\\
0 & 0 & -2/b_7(c_1 - x_3)x_7 & -2/b_7(r_1 - x_4)x_7\\
0 & 0 & -2/b_8(c_2 - x_3)x_8 & -2/b_8(r_2 - x_4)x_8
\end{bmatrix} \cdot \mathbf{u}\\
\mathbf{C} &= \begin{bmatrix}
0 & 0 & 0 & 0 & 1 & 0 & 1 & 0\\
0 & 0 & 0 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}\\
\mathbf{D} &= \mathbf{0}
\end{aligned} \tag{3.12}$$

3.1.2 Dynamics of the Heliostats

For the analysis of the dynamics of the heliostats, the time a heliostat needs to move its aim point over the maximum distance on the receiver is determined. Assuming a rectangular receiver, the maximum distance is the distance from the lower left to the upper right corner. As stated in subsection 2.1.1, the heliostats can move in longitudinal and elevation direction i.e. one motor moves the aim point from left to right and the other from down to up. Because the motors may not move simultaneously (which is the case at the solar tower in Jülich), the time needed to cover the maximum distance is the sum of the times needed to move from the left to the right edge and from the lower to the upper edge.

The distance d_{y} covered in y-direction dependent of the angular position α of the heliostat is expressed by

$$d_{\mathrm{y}} = s \cdot \tan\alpha \tag{3.13}$$

with s being the slant range calculated by

$$s = \sqrt{d^2 + (h_{\mathrm{tower}} - h_{\mathrm{heliostat}})^2} \tag{3.14}$$

The angle α is depicted with other relevant parameters in Fig. 3.1. For reasons of simplicity the receiver is assumed to be oriented so that its surface is perpendicular to the vector from the heliostat center to the receiver center. The distances between receiver and heliostat are usually so high that the tilt angle of the receiver is negligible. Therefore, this simplification is appropriate. α depends on the angular velocity ω. For reasons of simplicity the angular velocity ω is considered constant.

$$\alpha = \omega \cdot t \tag{3.15}$$

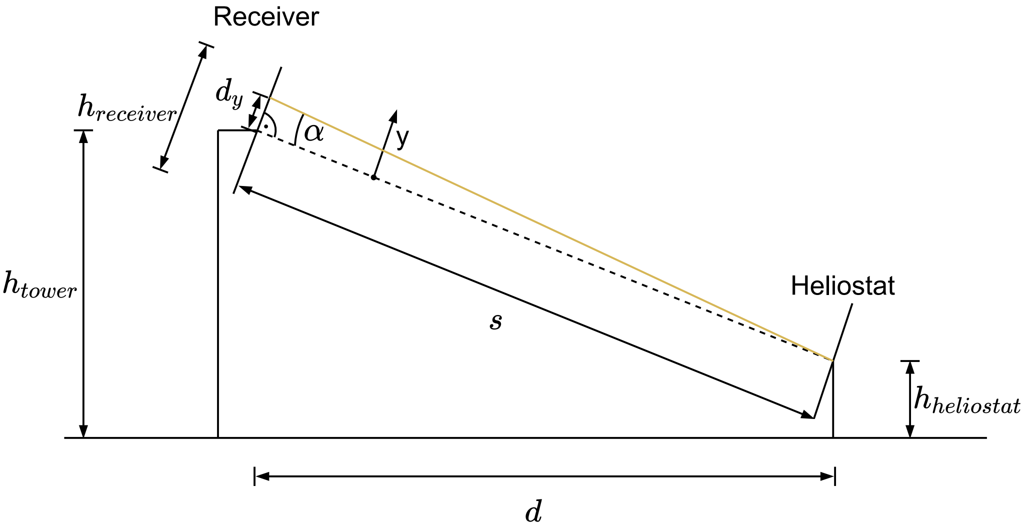


Figure 3.1: Longitudinal movement of the aim point.

Using Eq. 3.15, Eq. 3.13 can be expressed as

$$d_\mathrm{y} = s \cdot \tan(\omega \cdot t) \tag{3.16}$$

Therefore, the time to cover $d_\mathrm{y} = h_\mathrm{receiver}$ can be calculated by

$$t = \frac{\arctan(\frac{h_\mathrm{receiver}}{s})}{\omega} \tag{3.17}$$

The time to cover the distance w_receiver from the left to the right edge can be calculated in the same way.

For example, the tower in Jülich has a height of 60 m. For a close heliostat ($d = 30\,\mathrm{m}$) and the heliostat height of 1.5 m the slant range is equal to 65.7 m. The angular speed of the heliostats is estimated to 0.05 rad/s. This value is calculated by measuring the time a heliostat needs to aim from a 30 m distant off-receiver point to the receiver center at the solar tower in Jülich. With the receiver height of 5.2 m and the width of 4.4 m the heliostat presented above needs 1.51 s and 1.28 s to cover the respective distance. This would result in a time of 2.79 s to cover the maximum distance on the receiver.

3.1.3 Dynamics of the Receiver

For the analysis of the dynamics of the receiver, the critical time t_max in which an overflux condition damages the receiver is calculated. A ceramic receiver with air as heat transfer medium is considered because the controller will be tested on the solar tower in Jülich, which operates on a ceramic receiver. A ceramic receiver consists of several ceramic absorber structures also called modules. At a stationary operating point, a module has a module temperature T_module and a maximum material temperature T_max. Furthermore, it absorbs the irradiation with the flux density ϕ. The heat transfer medium flows through the module with a mass flow $\dot{m}_\mathrm{air}$ and exits it with the outlet temperature $T_\mathrm{air,out}$. Usually, the hot outlet air is returned after it was fed to the heat exchanger and mixed with the incoming air from the environment. For reasons of simplicity, this mixture of air streams going into the absorber is considered as one air stream with a mean air inlet temperature $\overline{T}_\mathrm{air,in}$. All parameters are depicted in Fig. 3.2. As stated in subsection 2.1.3, the allowed flux density depends on the characteristics of the used material. The maximum temperature is the restricting characteristic for the ceramic receiver.

In this work, the time t_max is defined as the time an overflux of 5 % requires to overcome a safety gap ΔT_module to the maximal allowed temperature of 10 K.

t_max can be expressed by an energy balance as follows:

$$\Delta T_\mathrm{module} = \int_{t_\mathrm{max}} \frac{\dot{Q}_\mathrm{total}}{m_\mathrm{module} \cdot c_\mathrm{p,module}} \mathrm{d}t, \tag{3.18}$$

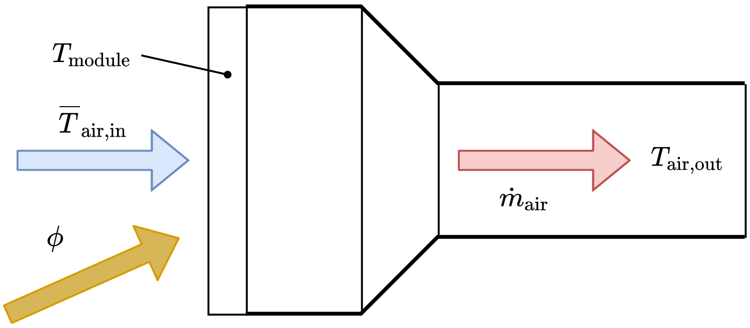


Figure 3.2: Parameters of an absorber module.

where m_{module} is the mass and $c_{\mathrm{p,module}}$ the heat capacity of the module. $\dot{Q}_{\mathrm{total}}$ represents the total heat flow, which can be expressed by

$$\dot{Q}_{\mathrm{total}} = \dot{Q}_{\mathrm{air,in}} - \dot{Q}_{\mathrm{air,out}} + \dot{Q}_{\mathrm{rad}} - \dot{Q}_{\mathrm{loss}}, \tag{3.19}$$

where $\dot{Q}_{\mathrm{air,in}}$ and $\dot{Q}_{\mathrm{air,out}}$ denote the incoming and outgoing heat flow of the air respectively, $\dot{Q}_{\mathrm{rad}}$ the incoming radiation of the heliostats and $\dot{Q}_{\mathrm{loss}}$ the loss of heat due to the fact that the absorber radiates itself. The four heat flows can be calculated by

$$\dot{Q}_{\mathrm{air,in}} = \dot{m}_{\mathrm{air}} \cdot c_{\mathrm{p,air,in}} \cdot \overline{T}_{\mathrm{air,in}} \tag{3.20}$$

$$\dot{Q}_{\mathrm{air,out}} = \dot{m}_{\mathrm{air}} \cdot c_{\mathrm{p,air,out}} \cdot T_{\mathrm{air,out}} \tag{3.21}$$

$$\dot{Q}_{\mathrm{rad}} = \alpha \cdot A \cdot \phi \tag{3.22}$$

$$\dot{Q}_{\mathrm{loss}} = \epsilon \cdot \sigma \cdot A \cdot (T_{\mathrm{module}}^4 - T_{\mathrm{a}}^4) \tag{3.23}$$

with α representing the absorptivity of the absorber material, $c_{\mathrm{p,air}}$ the heat capacity of the air, A being the area of the absorber, which receives the radiation, ϵ the emissivity of the module, σ denoting the Stefan-Boltzmann constant and T_{a} the ambient temperature.

It is assumed that $T_{\mathrm{air,out}}$ is equal to T_{module}. Then, t_{max} can be calculated using the values for a sample operating point at the solar power tower plant in Jülich, given in Tab. 3.1. A simulation done in MATLAB with the ode45 solver to solve a reformulated form of Eq. 3.18 yields an approximate time t_{max} of 10.0 s.

3.1.4 System Properties

The essential characteristics to design a controller can be derived from the small example presented in the previous subsection because the general form of the state space representation does not change with a larger system. Firstly, the small system has four inputs (aim point velocity) and two outputs (flux density). Thus, the system belongs to the class of multiple-input multiple-output (MIMO) systems. A solar power tower plant usually represents a large MIMO-system. For example, the Solar Two tower has 1926 heliostats (3852 inputs,

Table 3.1: Parameters for the thermal receiver model.

Parameter	Value	Unit
ϵ	0.92	-
σ	$5.67 \cdot 10^{-8}$	W/m^2 K^4
α	0.92	-
$c_{\text{p,air,in}}$	1200	J/kg K
$c_{\text{p,air,out}}$	1200	J/kg K
A	0.0225	m^2
m_{module}	1.2	kg
$c_{\text{p,module}}$	1000	J/kg K
$\dot{m}_{\text{air}}$	0.015	kg/s
T_{a}	298	K
$\overline{T}_{\text{air,in}}$	425	K
$T_{\text{air,out}}$	1373	K
T_{max}	1373	K

considering two aim point coordinates) and measures a flux image of the size 24 $\times$ 21 (equivalent to 504 outputs). Secondly, the plant system is a non-linear system because the matrix $\mathbf{B}$ depends on the system states. Furthermore, the system is stable in every operating point, when setting $\mathbf{u} = 0$. Then, $\dot{\mathbf{x}} = 0$ for every $\mathbf{x} \in \mathbb{R}^2$, which follows from Eq. 3.10 and 3.12. Thus, the system states do not change when no input is given and therefore the system is stable with respect to the definition given in section 2.2. The system is also stable in the sense that a bounded input yields a bounded output of the system, because the output i.e. a flux density can not exceed a certain limit due to the non-linearity of the system. The output is limited to the value, when all heliostats aim at one bin. While the outputs are bounded for every input, the system states are not. States like the aim point coordinates can rise till infinity for a constant input of the aim point velocity.

A crucial assumption is that the system can be considered static, when the sample time is higher than the time the heliostats need to move. After the heliostats have moved, ($\mathbf{u} = 0$) the system states will not change due to the fact that $\mathbf{A} = 0$. This is untypical for plant systems in control theory in which stabilizing a dynamic plant system is one main objective of the controller.

3.2 Controller Properties

After determining the properties of the plant system, the next step is to define the controlled variables (outputs of the plant system) and the manipulated variables (inputs to the plant system) [9]. This is done in the next subsection. Subsequently, a sample time is derived in order to be able to consider the plant system as static system and to derive a maximum sample time to prevent damage to the receiver.

3.2.1 Input and Output Selection

Within the stated example (see subsection 3.1.1) the aim point velocities are the inputs and the flux density distribution (consisting of several flux density measurements) is the output of the system. The flux density distribution is chosen as output because the control of this variable has a direct influence on the intercepted power of the receiver and thus the efficiency of the plant, which coincides with the objective of the thesis. Furthermore, by choosing the flux density distribution as output, the simulation does not need to model the thermal side, which simplifies the simulation.

In literature, the common choice for the inputs are the aim point coordinates, which is adopted in this work. Using other inputs, which influence the outputs, such as the voltage of the heliostat motors imposes more disturbances to consider and requires an extended simulation model. Furthermore, the dynamics of the motors is much higher than the sample time of the flux density measurement. Therefore, the heliostat movement can be implemented in a secondary control. Due to the fact, that the plant system is considered static, the aim point velocities are not chosen as inputs because this would insert a dynamic into the system.

3.2.2 Sample Time

The sample time is limited by a lower and an upper boundary. The lower bound is determined by the maximal time the heliostats need to move from one aim point to another plus the computational time of the control algorithm. If the sample time would be lower than the lower bound, the system can not be considered static. In this work, the minimum sample time (excluding the computational time of the control algorithm) is set to 2.79 s, which corresponds to the time to cover the maximum distance on the receiver at the solar tower in Jülich (see subsection 3.1.2).

The upper bound is influenced by the time after which a slight overflux damages the receiver. An overflux does not directly imply damage to the receiver but the increased temperatures and thermal stresses, which follow from an overflux condition [18]. Thus, the sample time needs to be smaller than the time an overflux causes critical temperatures or critical thermal stresses. In the worst case scenario, the overflux condition occurs right after the measurement. Then, the controller measures the critical condition after one sample time and has then another sample time to react. Consequently, in the worst case the controller needs two times the sample times to react. Therefore, the maximum sample time should be half of the critical time $t_{\max}$ in which an overflux condition damages the receiver. As stated in subsection 3.1.3, the time $t_{\max}$ at the solar tower in Jülich is estimated to 10.0 s. Therefore, the upper limit for the sample time is set to 5.0 s.

Usually, the sample time also depends on the system dynamics (e.g. Shannon theorem [22]) but as stated in subsection 3.1.4 the plant system is regarded as static. Thus, only the lower and upper bound are crucial.

In conclusion, a sample time between 2.8 s (plus the computational time) and 5.0 s would be required to be able to safely control the solar power tower plant in Jülich. The sample time range may change for other heliostat or receiver types as well as a more accurate approximation.

3.3 Considered Controller Architectures

To control the plant system, a controller, which is capable to control a static non-linear multi-input multi-output (MIMO) system, has to be designed. Within literature exist several control algorithms to address this kind of system. The next paragraphs present various control algorithms and evaluated them with respect to the given plant system. Existing control strategies for solar power tower plants like the MPC and the PID control are reevaluated as well.

H_∞ Loop Shaping

The H_∞ loop shaping tries to shape the frequency response of a closed-loop system. The frequency response is a measure of gain and phase of the output as a function of frequency given a sinusoidal input. Usually, a gain around 1 is preferred for a low frequency so that the system follows the reference signal with low deviation, while for high frequencies a gain towards zero is favored to suppress noises and other uncertainties. The control system designer shapes the closed-loop transfer function by weighting the plant transfer function in the frequency domain and obtaining a controller through optimization using an objective function including the ∞-norm. The H_∞ controller can handle multiple-input multiple-output (MIMO) systems but needs a linearized system [9]. As shown in Eq. 3.1, the plant system resembles a behavior similar to a Gaussian bell and is thus highly non-linear. Therefore, a linearization may not be appropriate.

Neural Nets

Neural nets are modeled loosely after the human brain. They try to learn a certain behavior based on given input and output data. A neural net could be used directly as a controller i.e. it calculates the manipulated variables (e.g. aim point coordinates) from the error signal (e.g. reference flux density distribution minus the measured flux density distribution). To obtain a neural net, training data is required, which would require measurements beforehand [38]. However, it is desired that the controller works without measurement data from the real plant.

Decoupling and PID Controller

A MIMO system can be decoupled into many single-input single-output (SISO) systems. Then, common approaches like a PID controller can be used to control the respective SISO systems. This approach is only reasonable, if the coupling between the inputs and outputs of the system is rather loose. However, a heliostat can influence several output variables. Thus, as also stated in [34], an approach with a decoupled system may not handle these interactions appropriately.

MPC

Model predictive control (MPC) is based on predicting the future states of a time-discrete dynamic model and calculating the next plant system input by minimizing an optimization problem. The optimization problem may consider a reference trajectory, the future states and outputs as well as state constraints. This control is not reasonable for the given plant system because the plant system behaves statically, as shown in subsection 3.1.4. Thus, a prediction of the future states is not reasonable because the states do not alter after one time step.

Extremum-Seeking Control with Sine Perturbation

In contrary to the other controllers, the extremum-seeking control does not follow a given reference signal but tries to find a minimum of an objective function during control. The classical approach is to perturb the input signal with a sine wave [39]. The resulting output signal is evaluated with respect to the objective function. The input signal is then changed in the direction of the lowest objective value. The controller reacts after the system was perturbed with a full sine wave. Due to the rather high sampling time, the period of the sine perturbation needs to be high in order to measure sufficient points of the sine wave. Hence, this approach is not suitable because the controller may react to slow.

ADRC

Active disturbance rejection control (ADRC) redesigns the PID controller to a non-linear feedback controller. It is a robust control method that extends the system model by a virtual state, which represents the sum of the disturbances and model errors, the so-called *total disturbance* [40]. The virtual state is estimated online by an adapted state observer. The algorithm is designed to reject the total disturbance. Thus, it can handle vast uncertainties, which would be helpful with regard to uncertainties like clouds. In contrast to the other control strategies the ADRC is a relatively new control approach. However, the ADRC contains many adjustable parameters in the magnitude of the number of states [41]. The number of states is proportional to the number of heliostats and the number of bins and rises usually above the magnitude of 10^7. Hence, fitting all the parameters may be a cumbersome and time-consuming process.

Hot Cold Controller

As explained in subsection 2.3.3, GarcÍa-MartÍn et al. [1] proposed a heuristic controller algorithm, which shifts the aim points from “hotter” zones to “colder” zones. While the algorithm is based on the measurement of the temperature, the idea could be easily adapted to work with the measurement of flux densities.

Static Optimal Control

A novel approach would be to adapt an existing open-loop controller for solar power tower plants to a closed-loop controller by integrating a feedback. The controller may be able to reject disturbances through the feedback of the measured signal. An option would be to extend an open-loop controller using an optimization algorithm. A feedback could be implemented by extending the objective function of the optimization algorithms with weights, which are modified depending on the measured error. The new controller would inherit the advantages of the optimization algorithm, which is the general formulation with respect to the receiver type and the disturbances. In doing so, it would also inherit the drawbacks of the optimization algorithm, which is mainly the computational time.

Comparison of Controllers

Most of the presented controllers are unattractive as a choice for a controller because of their drawbacks such as requiring a linearized or observable system. A summary of the drawbacks is given in Tab. 3.2. The most promising control strategies seem to be the heuristic Hot Cold controller and the novel Static Optimal Control approach. These controllers will be elaborated, implemented and tested within this work. For comparison i.e. as benchmark, the existing Vant-Hull controller is implemented and tested as well. The structure of the Vant-Hull controller, the Hot Cold controller and the Static Optimal Control are explained in more detail within the next sections.

3.4 Vant-Hull Control

The Vant-Hull control detects the highest measured overflux and then removes the heliostat from the receiver with the highest influence on that measurement point [12]. This step is repeated until the overflux is dispersed.

The first step after measuring the flux density distribution on the receiver is to calculate the critical flux distribution by subtracting from each measured flux density $y_{i,\mathrm{m}}$ the respective allowed flux density $y_{i,\max}$ at that bin and weighting it by the inverse of $y_{i,\max}$.

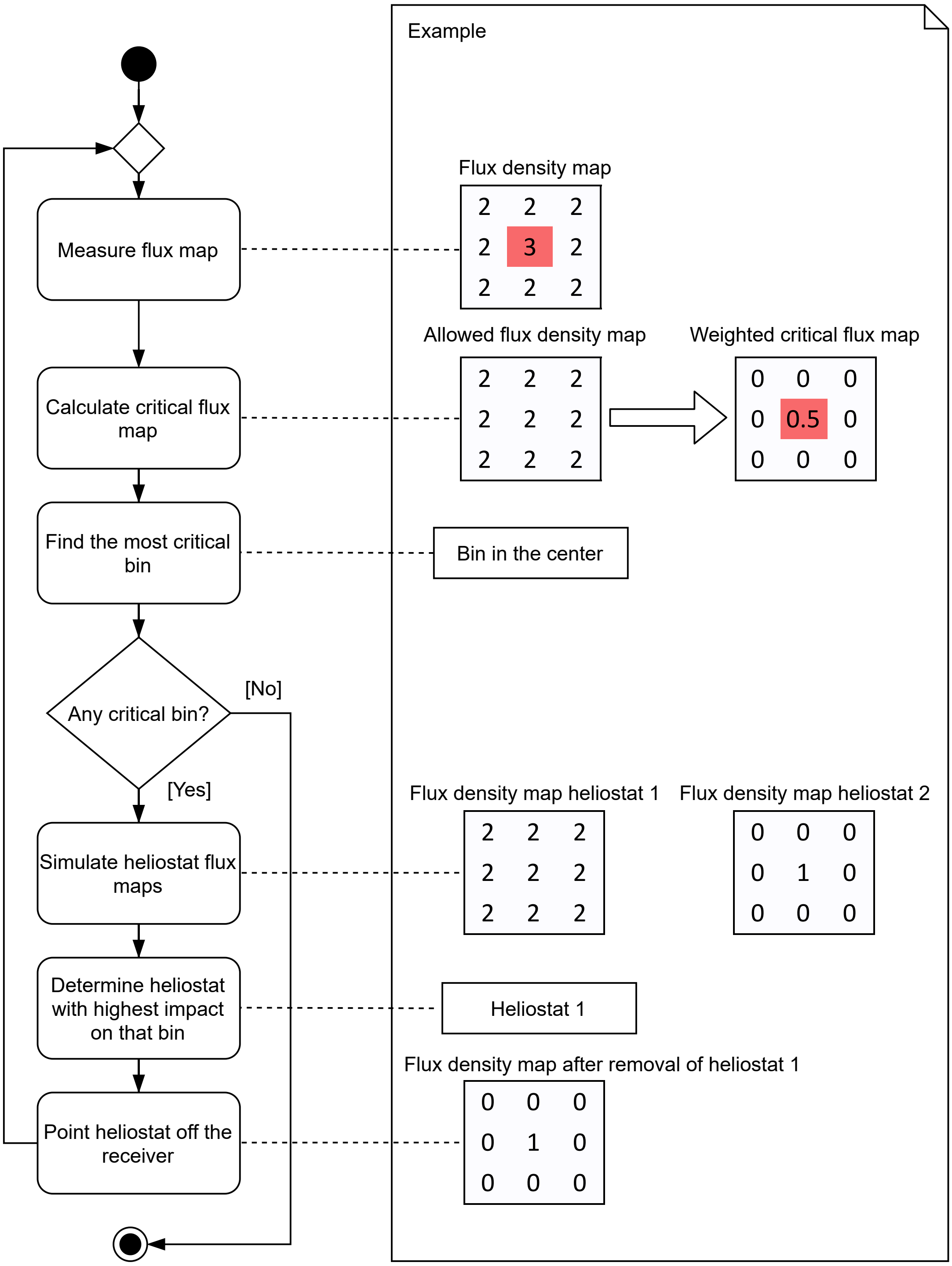
$$y_{i,\mathrm{weighted}} = \frac{y_{i,\mathrm{m}} - y_{i,\max,i}}{y_{i,\max}} \; \forall i = 1, \ldots, n_{\mathrm{bins}} \tag{3.24}$$

If all weighted flux densities are less than zero, no overflux condition is present. Thus, no bin is considered critical and the algorithm can stop. Otherwise, the most critical bin is the bin with the highest weighted flux density $y_{i,\mathrm{weighted}}$. The heliostat has to be determined, which increases the flux density at that bin the most. To determine the flux density impact of each heliostat, the flux density distribution of each heliostat is simulated. For the simulation, the software STRAL is used (see section 3.7). As the last step, the heliostat with the highest impact on the critical bin, is shifted to an off-receiver aim point, where it has no influence on the receiver. The steps of the algorithm as well as an example are shown in Fig. 3.3.

The Vant-Hull controller can only handle overflux conditions and thus requires a good initial aim point configuration. If the initial aim point configuration produces only low flux densities or disturbances reduce the flux density, the controller does not react. Thus, an additional controller, which calculates an initial aim point configuration or reacts to disturbances, would be necessary. Furthermore, the algorithm depends on a good simulation of the flux density distributions of the heliostats. The controller may shift a heliostat away, which has not the highest impact on the most critical bin, when the simulated flux density distribution is inaccurate. For example, when the impact of a heliostat is actually less than calculated due to an aim point offset caused by a tracking error. Lastly, the algorithm may decrease the flux density too much. For example, another heliostat may reduce the flux density sufficiently but decreasing the intercepted power less. This case is shown in the example of Fig. 3.3, where shifting heliostat 2 is a better solution. Another heliostat may also decrease more critical bins simultaneously and therefore its shift to an off point would be more reasonable.

Table 3.2: Comparison of the drawbacks of the considered controllers.

Controller	Drawbacks
H_∞ Loop Shaping	• Requires a linear system
Neural Nets	• Training data required
Decoupling and PID Controller	• Not reasonable for strongly coupled systems
MPC	• Not reasonable for static systems
Extremum-Seeking Control	• Not reasonable for a high sample time
ADRC	• Relatively new and not widely tested • Many adjustable parameters
Hot Cold Controller	• No optimal solution
Static Optimal Control	• Maybe high computation time


Measure flux map
Calculate critical flux map
Find the most critical bin
Any critical bin?
[No]
[Yes]
Simulate heliostat flux maps
Determine heliostat with highest impact on that bin
Point heliostat off the receiver
Example
Bin in the center
Heliostat 1

Flux density map

2	2	2
2	3	2
2	2	2

Allowed flux density map

2	2	2
2	2	2
2	2	2

Weighted critical flux map

0	0	0
0	0.5	0
0	0	0

Flux density map heliostat 1

2	2	2
2	2	2
2	2	2

Flux density map heliostat 2

0	0	0
0	1	0
0	0	0

Flux density map after removal of heliostat 1

0	0	0
0	1	0
0	0	0

Figure 3.3: Activity diagram of the Vant-Hull algorithm with a small example.

3.5 Hot Cold Control

The Hot Cold controller shifts the heliostat aim points from *hot* to *cold* areas, if the allowed flux density is exceeded. Otherwise, it shifts the aim points from colder to hotter areas. Hot areas denotes areas with relatively high flux densities while cold areas have rather low flux densities.

After measuring the current flux density distribution, the critical bins are determined like in the Vant-Hull algorithm (see section 3.4) by subtracting the allowed flux density from the measured flux density and weighting it by the inverse of the allowed flux density.

$$y_{i,\mathrm{weighted}} = \frac{y_i - y_{i,\mathrm{max}}}{y_{i,\mathrm{max}}} \; \forall i = 1, \dots, n_{\mathrm{bins}} \tag{3.25}$$

$$y_{i,\mathrm{critical}} = \max\left(y_{i,\mathrm{weighted}}, 0\right) \tag{3.26}$$

In the second step, the value $y_{i,\mathrm{critical}}$ of each bin is allocated to the closest aim point. The aim point with the highest sum of the allocated critical fluxes is considered most critical. Then, the most critical heliostat is determined by calculating the impact for each heliostat pointing to the most critical aim point and choosing the one with the highest impact. This is done by determining the overall *proximity* to the allowed flux density. This proximity p is defined by

$$p(\mathbf{y}) = \|\mathbf{y} - \mathbf{y}_{max}\|_1 \tag{3.27}$$

The heliostat impact ξ is then defined as the difference of the proximity of the measured flux density distribution $\mathbf{y}_{\mathrm{m}}$ with and without the heliostat flux density distribution $\mathbf{y}_h$.

$$\xi = p(\mathbf{y}_{\mathrm{m}}) - p(\mathbf{y}_{\mathrm{m}} - \mathbf{y}_h) \tag{3.28}$$

If no heliostat aims at the most critical aim point, the step is repeated for the next most critical aim point until an aim point is found at that heliostats aim. Subsequently, the most critical heliostat is shifted virtually to the upper, lower, left and right aim point with respect to the current aim point. A virtual shift consists of subtracting the simulated flux density distribution of the heliostat $\mathbf{y}_{\mathrm{h}}$ from the measured flux density distribution and adding the shifted simulated flux density distribution $\mathbf{y}_{\mathrm{h}}^{(\mathrm{shifted})}$. The shift with the highest impact which simultaneously reduces the sum of the exceeded weighted flux densities efr is chosen.

$$\xi_{\mathrm{h}}^{(\mathrm{shifted})} = p(\mathbf{y}_{\mathrm{m}}) - p(\mathbf{y}_{\mathrm{m}} - \mathbf{y}_{\mathrm{h}} + \mathbf{y}_{\mathrm{h}}^{(\mathrm{shifted})}) \tag{3.29}$$

$$efr = \sum_i \left(\max(y_{i,\mathrm{weighted}}, 0) - \max(y_{i,\mathrm{weighted}}^{(\mathrm{new})}, 0)\right) \tag{3.30}$$

with

$$y_{i,\mathrm{weighted}}^{(\mathrm{new})} = \frac{y_{i,\mathrm{m}} - y_{i,\mathrm{h}} + y_{i,\mathrm{h}}^{(\mathrm{shifted})} - y_{i,\mathrm{max}}}{y_{i,\mathrm{max}}} \; \forall i = 1, \dots, n_{\mathrm{bins}} \tag{3.31}$$

For $efr > 0$ the overflux condition is reduced. For $efr \leq 0$ the overflux does not decrease or increases. If $efr \leq 0$ for all shifts, the heliostat is shifted further (two steps up, down, left and right) and the impact is evaluated again. These steps are repeated until a valid shift is found. If no valid shift exists, the heliostat is shifted to an off-receiver aim point. The procedure is repeated until the overflux condition is resolved. The algorithm is illustrated with a small example in Fig. 3.4.

If no critical bin exists ($y_{i,weighted} \leq 0 \; \forall i$), a heliostat is moved from a cold spot to a hotter spot. To determine the coldest spot all weighted flux densities $y_{i,weighted}$ are allocated to the closest aim points. The coldest aim point is the one with the lowest sum. After that the procedure is the same as when shifting from hot to cold with the difference that only shifts are allowed which do not lead to an overflux condition ($efr \leq 0$). Moreover, if no valid aim point shift is found after a continuous increase of the neighborhood i.e. the considered aim points, the heliostat is not removed from the receiver but remains at its position. The same steps can be used because the impact is a measurement of how close the current flux density distribution is to the allowed flux density distribution. Thus, both when moving from hot to cold and from cold to hot, a shift, which yields a higher impact, is beneficial for the receiver.

The advantage of this algorithm is that it can react to all sorts of disturbances inter alia clouds. The drawback is that an inaccurate simulation of the flux density distribution of a heliostat may lead to a non-optimal shift. For example, if due to an offset of the heliostat, which was not simulated, the shift causes an overflux condition. Another drawback is that the algorithm only shifts one heliostat per step and also in a usually small neighborhood. Therefore, the algorithm may need many iterations to find good aim points for all the heliostats. The latter disadvantage can be addressed by grouping the heliostats or by allowing more heliostats to shift within one control step. Only the final result was considered to evaluate the controller within this work and thus, the latter suggestion was not implemented.

3.6 Static Optimal Control

The Static Optimal Control algorithm is based on an optimization algorithm. The controller maximizes an optimization problem, which is defined as follows

$$\begin{aligned} &\underset{\mathbf{u}}{\text{maximize}} \quad J(\hat{\mathbf{y}}, \mathbf{w}) \\ &\text{subject to} \\ &\hat{\mathbf{y}} \leq \mathbf{y}_{\max} \end{aligned} \tag{3.32}$$

where J denotes the objective function, $\hat{\mathbf{y}}$ is the flux density distribution calculated by a simulation given the aim point configuration $\mathbf{u}$. Furthermore, $\mathbf{y}_{\max}$ represents the allowed flux density and $\mathbf{w}$ the weights. All vectors have a size of $n_{\text{bins}} \times 1$ except for $\mathbf{u}$, whose length is equal to the number of aim points $n_{\text{aimpoints}} \times 1$.

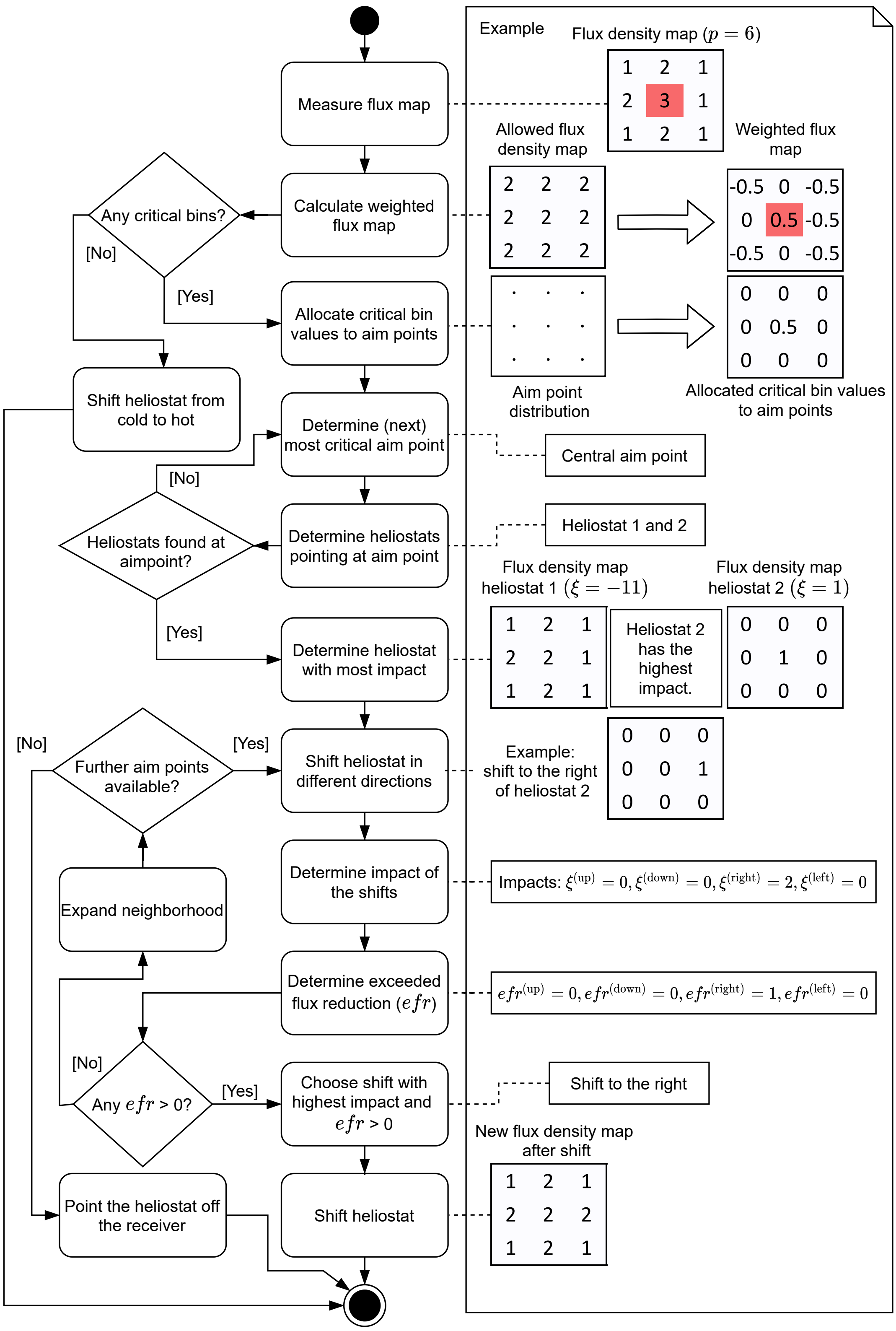

Measure flux map
Calculate weighted flux map
Any critical bins?
[No]
[Yes]
Allocate critical bin values to aim points
Shift heliostat from cold to hot
Determine (next) most critical aim point
[No]
Heliostats found at aimpoint?
Determine heliostats pointing at aim point
[Yes]
Determine heliostat with most impact
[No]
Further aim points available?
[Yes]
Shift heliostat in different directions
Expand neighborhood
Determine impact of the shifts
Determine exceeded flux reduction (efr)
[No]
Any efr > 0?
[Yes]
Choose shift with highest impact and efr > 0
Point the heliostat off the receiver
Shift heliostat
Example
Flux density map ($p = 6$)
1 2 1
2 3 1
1 2 1
Allowed flux density map
2 2 2
2 2 2
2 2 2
Weighted flux map
-0.5 0 -0.5
0 0.5 -0.5
-0.5 0 -0.5
Aim point distribution
0 0 0
0 0.5 0
0 0 0
Allocated critical bin values to aim points
Central aim point
Heliostat 1 and 2
Flux density map heliostat 1 ($\xi = -11$)
1 2 1
2 2 1
1 2 1
Heliostat 2 has the highest impact.
Flux density map heliostat 2 ($\xi = 1$)
0 0 0
0 1 0
0 0 0
Example: shift to the right of heliostat 2
0 0 0
0 0 1
0 0 0
Impacts: $\xi^{(\text{up})} = 0, \xi^{(\text{down})} = 0, \xi^{(\text{right})} = 2, \xi^{(\text{left})} = 0$
$efr^{(\text{up})} = 0, efr^{(\text{down})} = 0, efr^{(\text{right})} = 1, efr^{(\text{left})} = 0$
Shift to the right
New flux density map after shift
1 2 1
2 2 2
1 2 1

Figure 3.4: Activity diagram of a Hot Cold control step with a small example.

In every control step, the weights of the objective function are calculated based on the measured error e. The weights are estimated by a weight calculator in a way, that the algorithm yields a solution, which reduces the error. An activity diagram of one control step plus a small example of how the controller could possibly compensate a tracking error is given in Fig. 3.5.

The controller can be divided into the following functional components: the optimization algorithm (including the objective function), the error signal calculation and the weight calculator, which are shown in Fig. 3.6. These components are explained in more detail in the next subsections.

3.6.1 Optimization Algorithm

The optimization algorithm solves the stated problem in Eq. 2.9 (see subsection 2.3.1). Within this work, the ant colony optimization is chosen, which was adapted for the heliostat aiming problem by Belhomme [10] and is explained in section 2.4. The algorithm showed to find near optimal solutions in an adequate time [30]. In addition, the implementation of the algorithm is property of the DLR and thus, the source code was open to the author.

3.6.2 Objective Function

The objective function of the optimization algorithm depends on the simulated flux density distribution $\hat{\mathbf{y}}$, the allowed flux density $\mathbf{y}_{\max}$ and the weights $\mathbf{w}$, which are estimated by the weight calculator (see subsection 3.6.4). It could also depend on other variables, such as the covered distance of the aim point shift to restrict the heliostat movement. However, this is not within the scope of this work. Within this work the following four objective functions are investigated:

$$J = \mathbf{w}^T\hat{\mathbf{y}} - p\sum_i \max\left(\hat{y}_i - y_{i,\max}, 0\right), \tag{3.33}$$

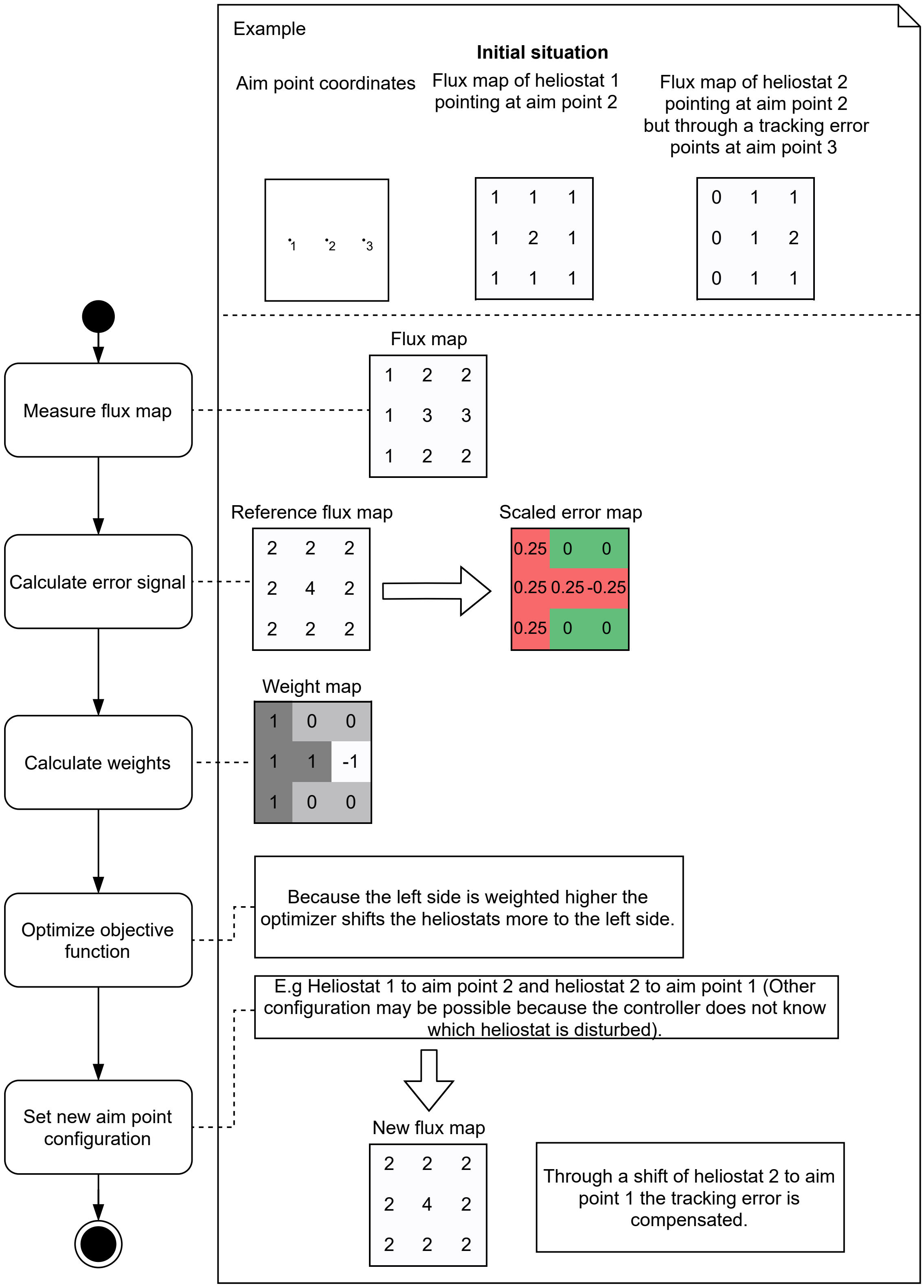
$$J = \mathbf{w}^T\hat{\mathbf{y}} - p\sum_i \max\left(w_i(\hat{y}_i - y_{i,\max}), 0\right), \tag{3.34}$$

$$J = \hat{\mathbf{y}} - p\sum_i \max\left(\hat{y}_i - w_i y_{i,\max}, 0\right), \tag{3.35}$$

$$J = -\|\mathbf{y}_{\mathrm{ref}} - \hat{\mathbf{y}} + \mathbf{w}\|_2^2 - p\sum_i \max\left(\hat{y}_i - y_{i,\max}, 0\right), \tag{3.36}$$

where p is the penalty factor.

The objective value is increased with an increasing simulated flux density, which can be influenced by the chosen aim points. An increase in the simulated flux density corresponds to an increase in the simulated efficiency of the plant. However, if the simulated flux density exceeds the allowed flux density, the objective value is reduced by the overflux multiplied with



Flux map of heliostat 1 pointing at aim point 2

1	1	1
1	2	1
1	1	1

Flux map of heliostat 2 pointing at aim point 2 but through a tracking error points at aim point 3

0	1	1
0	1	2
0	1	1

Flux map

1	2	2
1	3	3
1	2	2

Reference flux map

2	2	2
2	4	2
2	2	2

Scaled error map

0.25	0	0
0.25	0.25	-0.25
0.25	0	0

Weight map

1	0	0
1	1	-1
1	0	0

Because the left side is weighted higher the optimizer shifts the heliostats more to the left side.

E.g Heliostat 1 to aim point 2 and heliostat 2 to aim point 1 (Other configuration may be possible because the controller does not know which heliostat is disturbed).

New flux map

2	2	2
2	4	2
2	2	2

Through a shift of heliostat 2 to aim point 1 the tracking error is compensated.

Figure 3.5: Activity diagram of a control step of the Static Optimal Controller with a small example.

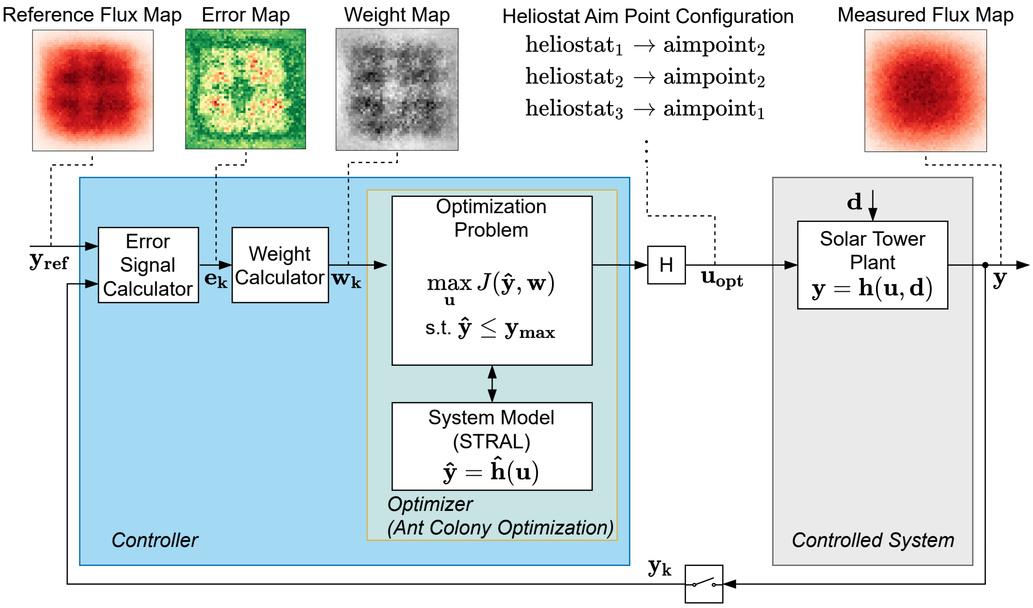

Reference Flux Map
Error Map
Weight Map
Heliostat Aim Point Configuration
$\text{heliostat}_1 \rightarrow \text{aimpoint}_2$
$\text{heliostat}_2 \rightarrow \text{aimpoint}_2$
$\text{heliostat}_3 \rightarrow \text{aimpoint}_1$
Measured Flux Map
$\mathbf{y_{ref}}$
Error Signal Calculator
$\mathbf{e_k}$
Weight Calculator
$\mathbf{w_k}$
Optimization Problem
$\max_{\mathbf{u}} J(\hat{\mathbf{y}}, \mathbf{w})$
s.t. $\hat{\mathbf{y}} \leq \mathbf{y_{max}}$
System Model (STRAL)
$\hat{\mathbf{y}} = \hat{\mathbf{h}}(\mathbf{u})$
Optimizer (Ant Colony Optimization)
Controller
H
$\mathbf{u_{opt}}$
$\mathbf{d}$
Solar Tower Plant
$\mathbf{y} = \mathbf{h}(\mathbf{u}, \mathbf{d})$
$\mathbf{y}$
Controlled System
$\mathbf{y_k}$

Figure 3.6: Static Optimal Control scheme. Above every signal, a sample signal is depicted.

the penalty factor. In this way, the constraint, mentioned in subsection 2.1.3, is imposed as a soft constraint into the optimization. In doing so, solutions that slightly violate the constraints are still valid. The four objective functions differ mostly in the placement of the weights. The objective of the weights is to direct the aim points of the heliostats to certain bins in order to correct uncertainties in the simulation like unknown disturbances or modeling errors.

Weighted Flux

The first function weighs the flux density at every bin. Thus, the optimizer preferably aims the heliostats closer to bins where the weights are higher relative to other weights.

Weighted Flux and Exceeded Flux

In contrast to the first function, the second function additionally weighs the overflux $(y_i - y_{i,\max})$. Thus, the weights also act like a second penalty factor, which may dampen the control action. For example, this may happen in the case of clouds shading the field. Clouds reduce the measured flux density but not the simulated flux density. Thus, the measured flux density is significantly lower than the simulated flux density. Therefore, the optimization algorithm needs to exceed the allowed flux density within the simulation in order to bring the measured flux density closer to the reference flux density. However, the optimization algorithm may not allow to exceed the allowed flux density in the simulation because the weights also act like a penalty. To counteract this behavior the penalty factor p should be smaller than or close to one in order to reduce the penalization by the weights.

Weighted Allowed Flux

The third function weighs the allowed flux density. Therefore, higher weights allow the optimizer to exceed the actual allowed flux density and thus allow the algorithm to concentrate the flux density more.

Quadratic

The last function is a quadratic function which also considers the reference flux density y_{ref}. To deduce more characteristics the function can be reformulated to:

$$J = -\sum_i (y_{i,\text{ref}} - y_i + w_i)^2 - p\sum_i \max\left(y_i - y_{i,\max}, 0\right) \tag{3.37}$$

Eliminating the brackets gives

$$\begin{aligned} J = &-\sum_i (w_i^2 + 2w_i y_{i,\text{ref}} + y_{i,\text{ref}}^2 - 2w_i y_i - 2y_{i,\text{ref}} y_i + y_i^2) \\ &- p\sum_i \max\left(y_i - y_{i,\max}, 0\right) \end{aligned} \tag{3.38}$$

Erasing the constant terms w_i^2, $2w_i y_{i,\text{ref}}$ and $y_{i,\text{ref}}^2$, which have no influence on the optimal solution, yields

$$J = \sum_i (2w_i y_i + 2y_{i,\text{ref}} y_i - y_i^2) - p\sum_i \max\left(y_i - y_{i,\max}, 0\right) \tag{3.39}$$

This function is similar to the first function as they both contain the term $\mathbf{w}^T\mathbf{y}$ and thus, in both functions the weights influence the flux density linearly. The quadratic function additionally considers the flux density in a quadratic term and the reference flux density. The reference flux density $\mathbf{y}_{\text{ref}}$ acts like another weight where bins with a higher reference flux density are promoted more.

Scaling

As stated in section 2.4, the objective value of the best ant is not allowed to be less or equal than zero and should be in the magnitude of one. Thus, the objective value has to be scaled, due to the fact that the proposed objective function can yield negative values or values greater than one. Within this work a simple linear scaling function is used, which can be expressed by

$$s = \frac{s_{\max} - s_{\min}}{x_{\max} - x_{\min}} \cdot (x - x_{\min}) + s_{\min} \tag{3.40}$$

$x_{\min}$ is set to the objective value using the simulated flux density from the current aim point configuration. Thus, the objective value of the best ant can not be less than zero as required because the best objective value needs to be greater than the one from the current aim point configuration, which is equal to the set minimum. Therefore, the simulated flux density distribution from the current configuration is a reasonable choice. $x_{\max}$ is calculated by

$$x_{\max} = x_{\min} + b \cdot |x_{\min}| \tag{3.41}$$

Here b denotes the improvement factor, which has to be fitted. b is chosen, such that the objective function does not exceed or only slightly exceeds $x_{\max}$ Then, the scaled objective value is within the magnitude of $s_{\max}$, which is set to the maximal visibility value, an intercept factor of unity. $s_{\min}$ is a parameter, which needs to be fitted to work with the ant colony optimization. The parameter must be greater than zero to fulfill the requirement of a positive objective value for the best ant.

3.6.3 Error Signal

The error is fed to the weight calculator. To calculate the error from the measured flux density distribution $\mathbf{y}$ and the reference flux density distribution $\mathbf{y}_{\mathrm{ref}}$ the following two formulations are proposed:

$$\mathbf{e} = \frac{\mathbf{y}_{\mathrm{ref}} - \mathbf{y}}{\|\mathbf{y}_{\mathrm{ref}}\|_{\infty}} \tag{3.42}$$

$$\mathbf{e} = \mathbf{y}_{\mathrm{ref}} - \mathbf{y} \tag{3.43}$$

The first formulation scales the error to the maximum flux density of the reference signal. Therefore, the error is around one and thus, the magnitude of the weights is more determined by the parameters of the weight calculator than by the error signal. It is desired that the magnitude of the weights is determined by the weight calculator. Additionally, through the scaling the reference flux density distributions can be exchanged without changing the parameters of the weight calculators. Scaled

The second formulation is an error signal as normally used in control loops. This error signal is only used for the quadratic objective function, because it requires weights around the magnitude of the measured signal. The magnitude of the weights for the quadratic objective function is different because the weights are not multiplied with but added to the measured output. Normal

3.6.4 Weight Calculator

The weight calculator calculates the weights used in the objective function and thus, impels the optimization algorithm to focus the radiation more onto certain bins than other. Generally, the weight calculator increases the weights if the error is positive i.e. if the measured flux density is lower than the reference flux density. Because the algorithm tends to focus on bins with higher weights, the increased weights reduce the error. If the error is negative, the weight calculator decreases the weights in order to defocus the heliostats from that point and thus, reduces the error. Within this work, the following two weight calculators are proposed:

$$\mathbf{w}_{k+1} = \mathbf{w}_k + K_{\mathrm{i}} \cdot \mathbf{e}_k, \tag{3.44}$$

$$\mathbf{w}_{k+1} = K_{\mathrm{p}} \cdot \mathbf{e}_k + K_{\mathrm{i}} \cdot \mathbf{e}_{k,\mathrm{integrated}} + K_{\mathrm{d}} \cdot (\mathbf{e}_k - \mathbf{e}_{k-1}) \tag{3.45}$$

with

$$\mathbf{e}_{k,\mathrm{integrated}} = \mathbf{e}_{k-1,\mathrm{integrated}} + \mathbf{e}_k \tag{3.46}$$

The first calculator is an integrator, which increases the current weight by the current error multiplied with the factor K_{i}. An integrator in control theory has the advantage that it does Integrator not permit a permanent control deviation but it has the disadvantage that it reacts rather slow [22].

The second estimator is adapted from an PID controller. Through the factors K_p the controller can react immediately on changes in the output. The factor K_d dampens the update of the weights and so reduces overshoot. The factor K_i acts like in the previous described weight calculator. PID

3.7 Simulation Software

There exist many models to simulate the flux of one heliostat on a specific point of the receiver such as the presented HFLCAL model in subsection 3.1.1. However, in this thesis, the solar tower ray tracing laboratory (STRAL) proposed by Belhomme [10] is used because it is more accurate than the HFLCAL model. Moreover, the simulation software was available to the author. STRAL is a Monte Carlo ray tracing tool, which can simulate the flux density distribution with a very high accuracy in a small amount of time [42]. For example, for a field with 100 heliostats it needs less than a second to calculate the flux density distribution. Furthermore, it can simulate all optical losses mentioned in subsection 2.1.2. However, the Vant-Hull and the Hot-Cold are implemented in STRAL 2 in contrary to the Static Optimal Control. Thus, only the Static Optimal Control can be tested with clouds.

In order to speed up the simulation of the flux density distribution precalculated flux density distributions for every heliostat aim point combination are used. Then, the simulation can sum up the respective flux density distributions of the heliostats to gain the final flux density distribution instead of tracing every ray. By using a GPU for the summation of the flux density distributions, the calculation is further accelerated.

The simulation software is not only required to test the algorithm in a simulation environment, but the precalculated flux density distributions are also used by the proposed controllers in this work. Thus, a precise simulation enhances the control quality.

4 Optimization of the Static Optimal Control

This chapter optimizes the parametrization of the Static Optimal Control in order to compare the controller against the other two controllers, which have no need for parametrization. The Static Optimal Controller is tested and optimized for three test cases, which are described in the first subsection. The algorithm possesses many adjustable parameters, which are contained within the ant colony optimization, objective function and weight calculator. The reference flux density distribution can be interpreted as a parameter as well. However, this work does not intend to prove the capability of the ant colony optimization for the heliostat aiming problem, which was already done by Belhomme [10]. Therefore, fixed parameters for the ant colony optimization were explored beforehand and used for all three test cases. Furthermore, the allowed flux density is considered static as stated in subsection 2.1.3 and therefore also the reference flux density distribution is set static. The fixed parameters and the reference flux density distribution are stated in the second subsection. For the comparison of different objective functions and different weight calculators, up to three criteria are chosen for each test case, which are described in the third subsection. Regarding the Static Optimal Control, this work only investigates the influence of different objective functions and different weight calculators. These are evaluated in the last two subsections.

4.1 Test Plant

The simulated plant is a small plant with a rectangular receiver and 100 heliostats arranged in a north-field layout as shown in Fig. 4.1a. Relevant parameters for receiver, heliostats and environment are stated in Tab. 4.1, Tab. 4.2 and Tab. 4.3 respectively. Furthermore, the allowed flux density is set to 50 kW/m^2. The allowed flux density is kept small in comparison to the maximum possible flux density in order to force the controller to find a solution which differs from the trivial solution of pointing all heliostats at the center. The aim points are distributed on a 5×5 grid as illustrated in Fig. 4.1b. Optical losses such as cosine loss, mirror reflectivity and air attenuation are considered.

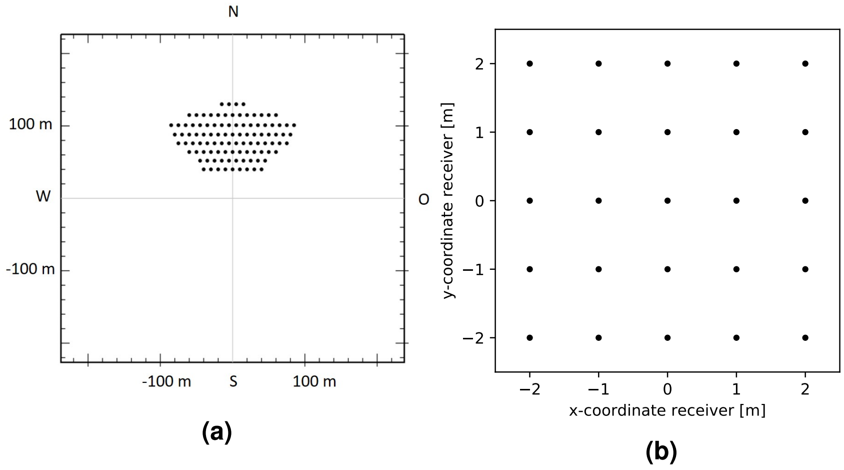


Figure 4.1: Field layout of the small reference plant (a) and aim point coordinates (b).

Within this work, three cases are tested to examine the control quality of the proposed controllers. In order to compare the flux density distributions of different controllers and test cases the *relative power* P_{rel} is introduced. The relative power is defined by the ratio of the current power P_{receiver} to the power induced by the reference flux density distribution P_{ref}:

$$P_{\mathrm{rel}} = \frac{P_{\mathrm{receiver}}}{P_{\mathrm{ref}}} \tag{4.1}$$

Here, the flux density distribution with a power of 587 kW shown in Fig. 4.2b is chosen as reference flux density distribution. It was determined by using the ant colony optimization algorithm to find an optimal aim point configuration with respect to the allowed flux density of 50 kW/m^2 without a mirror error.

Test Case 1

In the first test case, a mirror error of 3 mrad is simulated to test the behavior of the controllers with respect to optical losses. The initial flux density distribution is determined by applying the mirror error to the reference flux density distribution. The resulting flux density distribution is shown in Fig. 4.2a. Through the mirror error, the relative power on the receiver is decreased by 11.1 % to 88.90 %. The controllers task is to increase the power on the receiver. The best relative power is assumed to be around 92.74 %. This value was found by using the ant colony optimization to find an optimal aim point configuration, which considered the mirror error during optimization in contrary to the controller. Therefore, starting from the initial flux density distribution a maximal improvement of 3.84 % is possible.

Test Case 2

The second case tests the controller's reaction to overflux conditions. Therefore, the initial flux density distribution has flux densities going up to 75 kW/m^2 i.e. 25 kW/m^2 above the allowed flux density. The initial flux density distribution is illustrated in Fig. 4.3. Furthermore, a mirror error of 3 mrad is simulated to enforce a deviation between the simulated plant and the plant system used by the controllers. Without a deviation between these two plant

Table 4.1: Receiver parameters of the test plant.

Parameter	Value	Unit
Tower height to receiver center	60	m
Height	5	m
Width	5	m
Measurement bins horizontal	50	-
Measurement bins vertical	50	-

Table 4.2: Heliostat parameters of the test plant.

Parameter	Value	Unit
Total refelective area	8	m^2
Facets vertical direction	2	-
Facets horizonal direction	2	-
Facet width	1.25	m
Facet height	1.6	m
Gaps between facets	None	
Pedestal height	2	m
Focal length	0	m
Canting point	(0,0,100)	m
Tracking error	0	mrad
Total reflectivity	0.92	-
Axis offset	0	m

Table 4.3: Environment parameters of the test plant.

Parameter	Value	Unit
Latitude location	37.095	°
Longitude location	-2.360	°
Ambient temperature	298.15	K
Metres above sea level	500	m
DNI	850	W/m^2
CSR sun distribution	5	%
Azimuth sun position	0	°
Elevation sun position	45	°

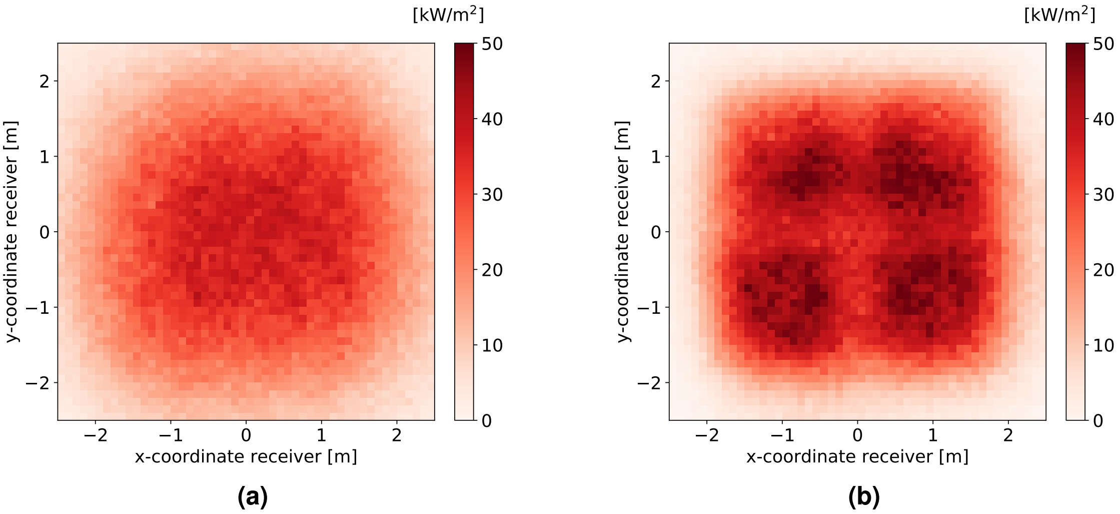


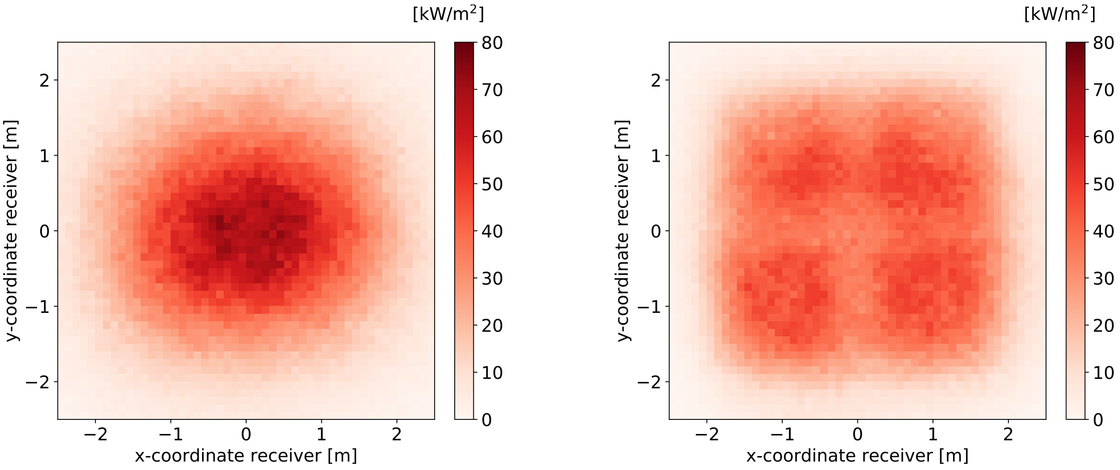
Figure 4.2: Initial flux density distribution test case 1 (a) and reference flux density distribution (b).

systems an open-loop controller would be sufficient and the test of a closed-loop controller unnecessary.

Test Case 3

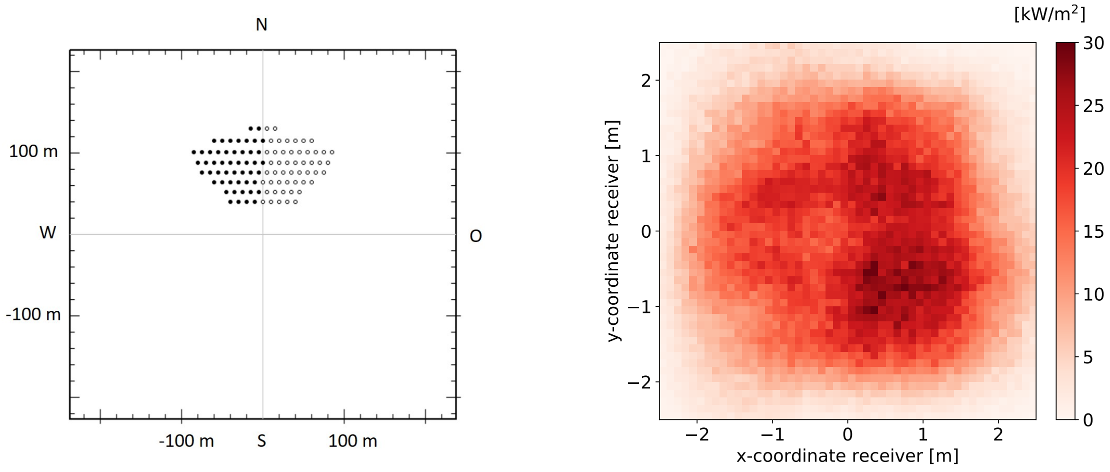
The third case tests the behavior of the controller with regard to dynamic disturbances such as clouds. The cloud simulation is achieved by setting the DNI to zero for half the heliostats after 30 control steps. This is equivalent to the case that clouds shade half of the field. The shaded heliostats are shown in Fig. 4.4a. For example, the flux density distribution in Fig. 4.2a, would look like the flux density distribution shown in Fig. 4.4b, when half the field is shaded. The relative power drops from 88.90 % to 45.19 % due to the clouds. After another 30 control steps, at step 60, the clouds pass i.e. the DNI is set to the initial intensity again. Furthermore, a mirror error of 3 mrad is simulated during the whole test for the same reason as described in the previous paragraph.

The presented test cases cover most of the situations the controller is confronted with in a real life application. The mirror error reduces and widens the flux density distribution. Other errors (see subsection 2.1.2) have a similar influence. They usually either decrease and/or widen the flux density distribution. Therefore, the simulation of the mirror error is a good choice to represent different optical losses. The second case tests the critical overflux condition. The test gives a tendency of how overflux condition are handled. In contrary to the first two test cases, the last case tests the reaction with respect to dynamic changes. By means of the mirror error of 3 mrad, the first two test cases only simulate a slight deviation between the controlled plant system and the plant system used within the controller. However, by shading half of the field in test case 3, a vast deviation is simulated. Therefore, the controller's ability to work with high modeling errors is tested. Furthermore, a cloud usually leads to an overflux as described in subsection 2.3.1, which usually causes a higher overflux than other disturbances. Therefore, the test case 3 additionally tests the reaction to an overflux



(a) (b)

Figure 4.3: Initial flux density distribution test case 2 (a) compared to the reference flux density distribution (b).



(a) (b)

Figure 4.4: Shaded heliostats marked as black points (a) and resulting flux density distribution (b).

condition. Because test case 3 starts from the same initial flux density distribution as test case 1, and also usually includes an overflux condition, it tests implicitly the same situations presented in test case 1 and 2. However, the first two test cases are still tested to evaluate each situation separately.

4.2 Fixed Parameters

The set parameters for the ant colony optimization are listed in Tab. 4.4.

The number of control steps is set to 100 because the algorithm usually converges during this amount of steps. The weights are always initialized with 1 except for the quadratic objective function, whose weights are initialized with 0.

If the Static Optimal Control has no feedback, i.e. the weight calculator does not update the weights, this corresponds to an open-loop control that uses only the ant colony optimization algorithm. The objective is to show that test case 1 can be initialized in such a way that the performance improvement can only be attributed to the closed-loop and not to the open-loop control. Thus, the effectiveness of the Static Optimal Controller can be proven. For this purpose, the initial aim point configuration is set in a way that it corresponds to an optimal aim point configuration for the ant colony optimization algorithm. This means that the open-loop controller alone cannot improve the performance. If the Static Optimal Controller is used, the performance improvement can then be fully attributed to the closed-loop control. Based on the initial aim point configuration, a flux density distribution without mirror errors is simulated. This simulated flux density distribution is set as reference flux density distribution for all three test cases. This reference flux density distribution corresponds to the reference flux density distribution in Fig. 4.2b.

Table 4.4: Parameters of the ant colony optimization algorithm.

Parameter	Value
Number of runs	50
Number of ants	50
β	2.0
Pheromone start value	0.01
ρ	0.15
q_0	0.98
$s_{\min}$ and $s_{\max}$ for scaling	[0.12, 1.0]

4.3 Evaluation Criteria

The error of the relative power P_{rel} is defined as the difference of the current relative power and the relative power of one. In order to evaluate the parameter combinations for test case 1 and 2 the mean squared error e_{mse} of the relative power for all 100 control steps is taken into account.

$$e_{\mathrm{mse}} = \frac{1}{100} \sum_{i=1}^{100} (P_{\mathrm{rel,i}} - 1)^2 \tag{4.2}$$

The mean squared error is chosen because it is commonly used in the literature. A lower e_{mse} is desired because it represents a higher final convergence value and/or a faster convergence. However, because the algorithm converges within less the half of the control steps, the e_{mse} represents rather a higher convergence value than a fast convergence. In order to separate these influences, the mean relative power of the first ten μ_{f10} and of the last twenty μ_{l20} control steps is also considered.

For μ_{f10}, the range is set to ten, as this seemed to be an approximate number of control steps the algorithm needs to settle at the beginning. Thus, this criterion describes the convergence speed rather than a high convergence value.

On the one hand, μ_{l20} considers the last twenty steps and not more to ensure that the algorithm already converged. On the other hand, it considers not less than 20 steps to give a representative mean value, which compensates fluctuations in the relative power of the last control steps. The mean relative power is considered instead of the mean squared error in order to compare these criteria to the initial relative power.

For test case 3, two criteria are considered for the evaluation. The first criterion k_{noc} (noc = no overflux condition) states how many control steps the controller needs to eliminate the overflux condition, which may arise after the clouds pass as described in subsection 2.3.1. The other criterion describes the final convergence value of the relative power during the cloud phase. This criterion is expressed by the mean relative power μ_{cloud} of step 49 to 59, the final phase before the cloud passes at step 60. A low value for k_{noc} and high value for μ_{cloud} are desired. However, a higher μ_{cloud} value usually leads to a higher overflux after the clouds passes because the controller concentrated the flux more in the center. To eliminate a higher overflux, the controller usually needs more steps. Thus, a higher μ_{cloud} usually leads to a higher k_{noc}.

The criteria differ for runs with the same parameter combination due to the fact that the ant colony optimization is non-deterministic. To account for fluctuations in the criteria, the value for each criterion takes the mean of five runs of the same parameter configuration into account.

4.4 Evaluation of Different Objective Functions

The objective functions are evaluated for all three test cases in combination with the integrating weight calculator (see subsection 3.6.4). The integrating weight calculator is chosen because it has the fewest parameters to adjust. Each objective function is influenced mainly by the penalty factor and the weights. To compare the objective functions comprehensively, each one should be tested with a wide range of different weights and penalty factors. A wide range of weights is tested indirectly by varying the integrating factor and through the amount of control steps. Thus, for each objective function, two parameters are adjusted: the penalty factor p and the integrating factor K_i of the weight calculator.

Each objective function is tested with a different set of parameters listed in Tab. 4.5. The parameters were chosen in a way that they yielded solutions which were close to the maximum of 92.74 % (see section 4.1). The following subsections describe the results for each test case.

Table 4.5: Parameter ranges for the test of the objective functions used for all test cases (WF = Weighted Flux, WFEF = Weighted Flux and Exceeded Flux, WMF = Weighted Maximum Flux, Q = Quadratic).

Objective function	p	K_i
WF	{1000, 100, 10, 1, 0}	{1000, 100, 50, 10, 5, 3, 1}
WFEF	{2, 1, 0.5, 0.3, 0.1}	{1000, 100, 50, 10, 5, 1}
WMF	{1000, 100, 10, 1, 0}	{10, 5, 3, 1, 0.1}
Q	{1000, 100, 10, 1, 0}	{10, 5, 3, 1, 0.5, 0.1}

4.4.1 Results Test Case 1

To ensure a safe plant operation, it is important that no overflux condition is caused by the controller. Therefore, for test case 1 only parameter combinations are regarded valid, which do not cause an overflux condition during the 100 control steps. Regarding this condition, the weighted maximum flux objective function showed the best results with respect to all criteria. The best value for each criterion of each objective function is stated in Tab. 4.6. If an objective function is better in one criterion, it is also for the other criteria. Overall, it can be recognized that a high overall mean squared error relates to a fast progress as well as a high final relative power, although there are slight differences.

An overview of criteria such as the mean error over all bins, the sum of the overflux, the relative power as well as certain flux density distributions for the parameter configuration with the best e_mse is shown in Fig. 4.5. By comparing the initial and the last flux density distribution, it can be recognized that the controller centers the aim points more during the control. However, the last flux density distribution still does not resemble the reference flux density distribution as the mirror error does not allow such a distinct concentration. Furthermore, the relative power as well as the error of the weighted maximum flux objective function rises quickly in the first step, but then it drops again, before it converges. An explanation for this behavior could not be found, but this is not a desired behavior. Therefore, the quadratic objective function, which is the second best objective function, may be a better choice. This objective function rises more steadily to the final value without big drops as can be seen in Fig. 4.6.

Table 4.6: Best results and their parameters for test case 1 (WF = Weighted Flux, WFEF = Weighted Flux and Exceeded Flux, WMF = Weighted Maximum Flux, Q = Quadratic).

Objective Function	e_{mse}	p	K_{i}	μ_{f10}	p	K_{i}	μ_{l20}	p	K_i
WMF	0.635 %	1	1	91.72 %	1	0.1	92.16 %	1	1
Q	0.684 %	100	3	91.66 %	100	3	91.78 %	10	0.5
WF	0.690 %	1	1	91.39 %	1	1	91.77 %	10	10
WFEF	0.741 %	1	100	91.36 %	1	100	91.46 %	1	0.1

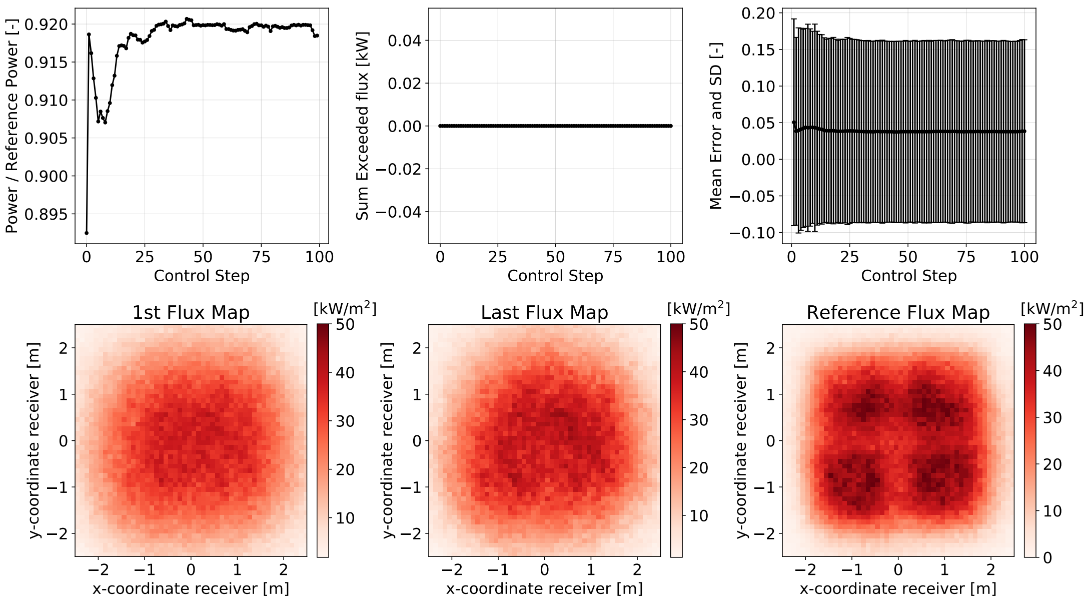


Figure 4.5: Plots of the weighted maximum flux objective function with the best e_{mse} for test case 1.

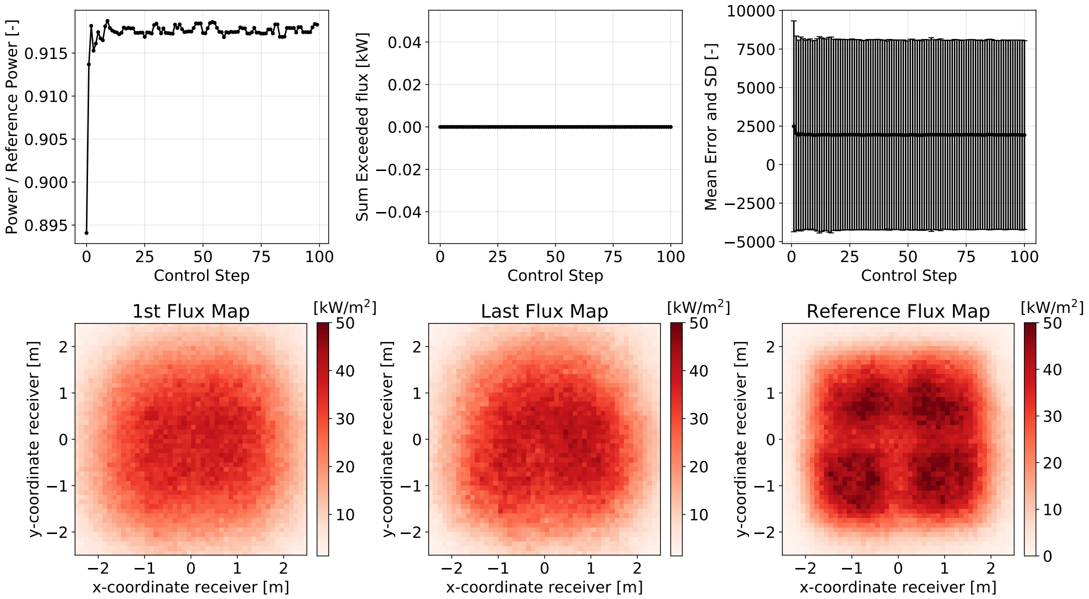


Figure 4.6: Plots of the quadratic objective function with the best e_{mse} for test case 1.

4.4.2 Results Test Case 2

In test case 2, there exist, a parameter combination for each objective function that leads to an immediate reduction of the flux density so that the overflux condition is not present after one step. However, in all these combinations, the relative power drops significantly at first. It decreases to a level of about 85 % and then raises again to its convergence value. The fast reaction to an overflux condition may be due to the almost precise simulation of the flux density within the controller, which only differs through the mirror error from the simulated flux density distribution. Therefore, the vast overflux condition also represents an overflux condition within the simulation of the controller. An overflux condition decreases the objective value due to the penalty term. Thus, the optimizer defocuses the aim points from the center to decrease the overflux and increase the objective value. Additionally, the initial overflux causes lower weights in the center and higher weights far from the center. This also impels the optimization to defocus the heliostats more from the center. The combination of both reasons, probably causes the initial big drop of the relative power.

There are also combinations needing more than one control step to eliminate the overflux and have better values with respect to the criteria. Nonetheless, due to safety reasons a fast elimination of the overflux is more important than a high value in any criteria. Consequently, only parameter combination eliminating the overflux condition after one step and producing no overflux condition thereafter are considered valid. In doing so, the e_{mse} and the other criteria are comparable. The best value for each criterion of each objective function in test case 2 is stated in Tab. 4.7. The weighted maximum flux objective function again yields the best results. An overview for the weighted maximum flux objective function with the best e_{mse} is depicted in Fig. 4.7. Here, the drop of the relative power after the first step can be noticed.

Table 4.7: Best results and their parameters for test case 2 (WF = Weighted Flux, WFEF = Weighted Flux and Exceeded Flux, WMF = Weighted Maximum Flux, Q = Quadratic).

Objective Function	e_{mse}	p	K_{i}	μ_{f10}	p	K_{i}	μ_{l20}	p	K_i
WMF	0.687 %	1	1	90.99 %	1	0.1	92.02 %	1	0.5
WF	0.690 %	1	1	90.20 %	1	1	91.88 %	10	5
Q	0.774 %	100	3	89.46 %	100	3	91.39 %	100	3
WFEF	0.781 %	1	1	90.25 %	1	0.1	91.28 %	1	1

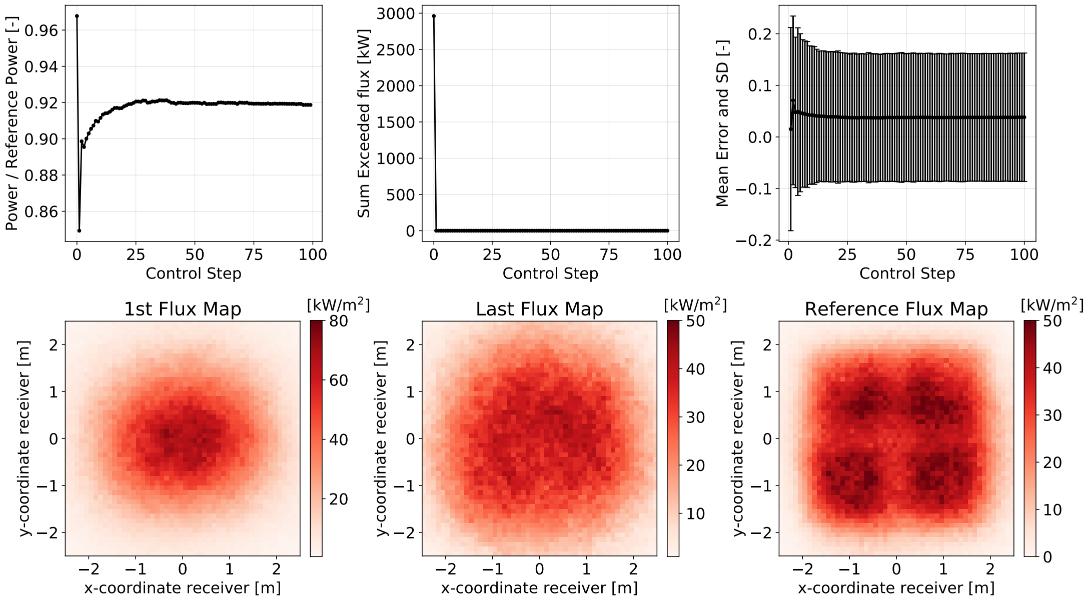


Figure 4.7: Plots of the weighted maximum flux objective function with the best e_{mse} for test case 2.

4.4.3 Results Test Case 3

In test case 3, most parameter combinations caused an overflux condition after the cloud passed. Especially, the combinations which yielded a high μ_{cloud} value. In contrary, to test case 2 the parameter combinations do not eliminate the overflux condition within one control step. This is probably due to the reason that the weights are already adjusted to the cloud situation, in contrary to test case 2 where they start from the initial weights. In order to readjust the weights, the controller needs more than one control step. This situation is more realistic than the situation in test case 2. There are also parameter combinations, which did not cause an overflux condition after the cloud passed. However, these combinations usually performed worse with respect to the μ_{cloud} and μ_{120} value.

In Tab. 4.8, the best value for the criteria k_{noc}, μ_{cloud} and μ_{120} of each objective function in test case 3 is stated. With respect to the μ_{cloud} and μ_{120} values the weighted maximum flux objective functions yields the best results. However, it needs almost double as many steps to eliminate the overflux than the worst k_{noc} from the other objective functions. The quadratic objective has a lower k_{noc} value and is the second best objective function with respect to the other criteria. For the weighted flux objective function and the weighted flux and exceeded flux objective function exist parameter combinations, which yield a k_{noc} equal to zero. However, these combinations result in worse values with respect to the other criteria. An overview of the results for the quadratic objective function tested under the conditions of test case 3 is depicted in Fig. 4.8. The overflux is eliminated after 14 steps. However, for the last 4 steps before the elimination of the overflux, the flux density does not exceed more than 5 % of the allowed flux density. Still, this would not be sufficient for a commercial plant. Furthermore, the figure reveals that the relative power still drops even when the overflux condition is eliminated. This is not desired and probably due to a non-optimal reference flux density distribution. The reference flux density distribution has its maximum not in the center but around the center in 4 particular areas. Therefore, the controller may defocus the heliostats from the center to these areas, which leads to a reduction in the relative power.

Table 4.8: Best results and their parameters for test case 3 (WF = Weighted Flux, WFEF = Weighted Flux and Exceeded Flux, WMF = Weighted Maximum Flux, Q = Quadratic).

Objective Function	e_{mse}	k_{noc}	μ_{cloud}	μ_{120}	p	K_i
WF	9.612 %	0	46.11 %	91.97 %	100	5
WF	9.282 %	14	46.86 %	92.13 %	100	100
WF	9.312 %	12	46.81 %	92.15 %	10	5
WFEF	9.945 %	0	45.32 %	90.40 %	5	0.1
WFEF	9.500 %	0	46.21 %	91.82 %	1	10
WFEF	9.500 %	0	46.21 %	91.82 %	1	10
WMF	9.300 %	29	47.00 %	91.47 %	5	0.1
WMF	8.950 %	37	48.04 %	92.50 %	1	1
WMF	8.950 %	37	48.04 %	92.50 %	1	1
Q	9.284 %	12	46.84 %	92.14 %	100	1
Q	9.263 %	17	46.87 %	92.14 %	100	5
Q	9.303 %	13	46.78 %	92.18 %	1	0.5

A better parameter combination would be the one for the weighted flux objective function shown in Fig. 4.9. This combination does not cause an overflux condition. It still improves the relative power although, even though it is less than the combination from the quadratic objective function. However, it cannot be guaranteed that this combination does not lead to an overflux condition in any cloud situation.

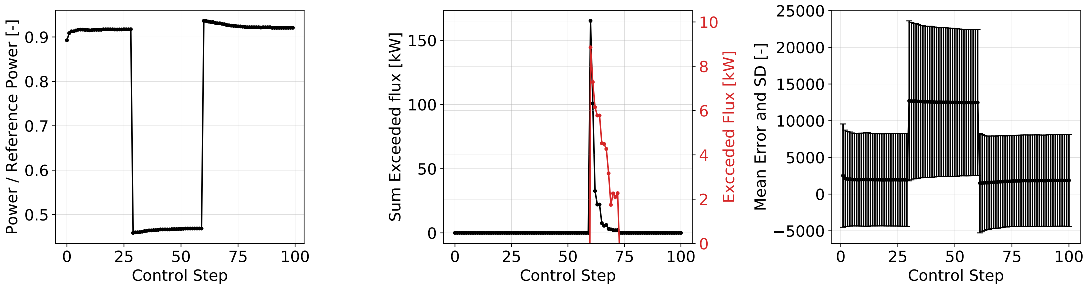


Figure 4.8: Plots of the quadratic objective function for test case 3.

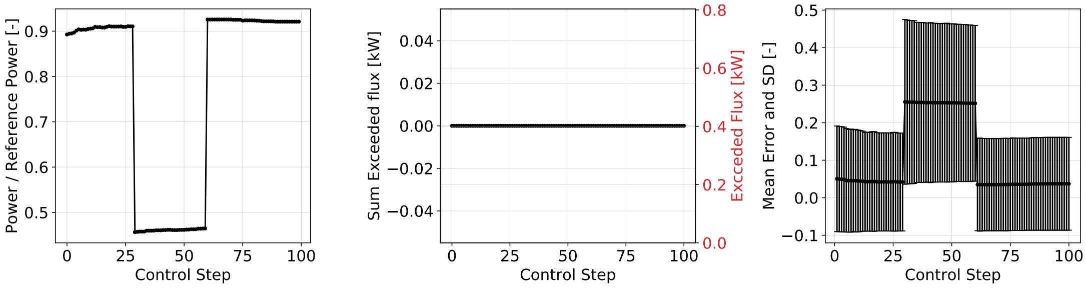


Figure 4.9: Plots of the weighted flux objective function for test case 3.

4.4.4 Insights Parametrization

Based on the test cases, some additional insights regarding the adjustable parameters can be concluded for each objective function.

For the weighted flux objective function, the results yielded that an integrating factor should be around the same magnitude as the penalty factor. On the one hand, with a high integrating factor, the weights rise quickly above the magnitude of the penalty factor. The influence of the penalty term decreases and a solution with an overflux condition may yield a higher objective value. Thus, a too high integrating factor relative to the penalty factor leads to an overflux condition during control. On the other hand, yielding a solution close to the reference flux density distribution, may require to exceed the allowed flux density within the internal simulation of the controller. If the weights are too small, they can not counteract the penalty term and a solution with an internal overflux does not yield a higher objective value. Hence, a too low integrating factor relative to the penalty factor leads to a slow rise in the relative power as shown in Fig. 4.10.

The quadratic objective function yields the best solutions with an integrating factor around 3. This is reasonable because the weights should be around the magnitude of the flux density as described in subsection 3.6.3, which is enforced by an integrating factor around the magnitude of one. In test case 1, the penalty factor has no influence on the objective function, as can be seen in Tab. 4.9. This means that the allowed flux density is not exceeded during the simulation.

Table 4.9: Results for different penalty factors with the quadratic objective function in test case 1 (Q = Quadratic).

Test Case	Objective Function	e_{mse}	μ_{f10}	μ_{l20}	p	K_{i}
1	Q	0.710 %	91.52 %	91.61 %	1000	3
1	Q	0.710 %	91.53 %	91.60 %	100	3
1	Q	0.710 %	91.54 %	91.61 %	10	3
1	Q	0.710 %	91.52 %	91.60 %	1	3
1	Q	0.710 %	91.51 %	91.61 %	0.1	3

With respect to the weighted flux and exceeded flux objective function, it could be determined that a lower penalty factor yields better solutions regarding the criteria. A tendency for the magnitude of the factor K_{i} could not be found.

If the integrating factor is above 3 for the weighted flux objective function, the allowed flux density is usually exceeded. A small penalty factor usually gives better solutions.

4.4.5 Conclusions

In conclusion, all objective functions raise the relative power during control. The weighted maximum flux objective function showed the best results with respect to a high relative power. For a save operation, it is more important that overflux conditions are eliminated quickly than yielding a high relative power. Here, the weighted flux objective function shows its limits. It is the slowest in eliminating the overflux. Therefore, other objective functions like the weighted flux objective function may be more suitable.

4.5 Evaluation of Different Weight Calculators

The weight calculators are tested with the weighted flux objective function. The weighted flux objective function has no quadratic term and also, its weights do not directly correlate with the penalty factor as it is the case for the other proposed objective functions.

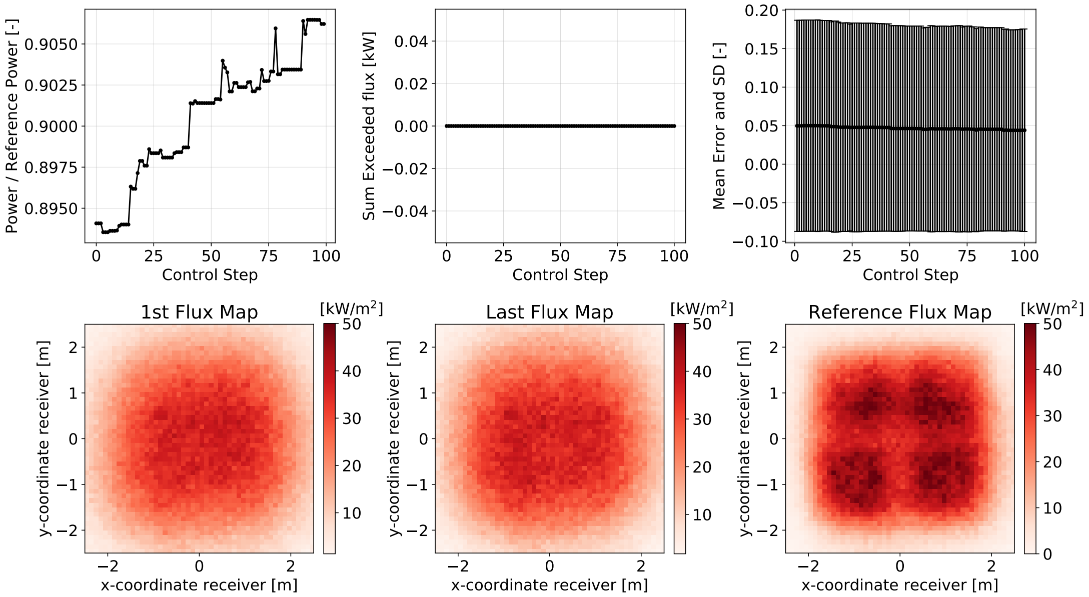


Figure 4.10: Plots of the weighted flux objective function with $p = 100$ and $K_\mathrm{i} = 1$.

Within this work, an integrating and a PID weight calculator are tested. The objective of using an integrating weight calculator is not to react quickly to changes in the flux density distribution but to yield a high final convergence value over time. Therefore, by extending the integrating weight calculator to an PID weight calculator, the objective is not to increase the convergence value but to be able to react faster to changes in the flux density distribution while not decreasing the convergence value. Due to safety reasons, a fast reaction is especially important when overflux conditions occur like in test case 2 and 3. However, for test case 2 the integrating weight calculator can eliminate the overflux within one control step as shown in the previous subsection. Because a faster reaction is not possible in test case 2, the weight calculators are only evaluated with respect to test case 3.

In context of overflux condition the criterion k_noc is of special interest because it gives an evidence of how fast the controller reacts to an overflux. Consequently, the different parameter combinations are evaluated with respect to this criterion.

4.5.1 Results Test Case 3

An evaluation beforehand yielded that for a PID weight calculator no parameters could be found, which gave similar convergence values as the integrating weight calculator. This does not imply that there is no parameter combination which yields better solutions. However, one test run needs about 2.2 min and there are no rules of thumb to adjust the parameters. Therefore, only a few combinations could be tested in the infinite space of possible combinations. In the range of untested parameter combinations, there may be more suitable combinations.

For that reason, the influence of the P and the D term are tested separately in combination with the I term. The integrating parameter is set constant to reduce the parameter space. The integrating parameter and the penalty factor is set to 1 because this yielded good solutions in combination with the weighted flux objective function.

The PI weight calculator was tested in a range from 0 to 6 with a step size of 0.01 for the K_p value. Values above 2 lead to an overflux condition caused by the controller before the occurrence of clouds and are not appropriate. Values between 0 to 2 had almost no influence on the μ_l20 and the μ_cloud value as required. These values deviated not more than 0.01 % and 0.03 % respectively compared to the parameter combination with $K_\mathrm{p} = 0$. However, the P-term can reduce the k_noc value up to 2.6 steps in average from 74.8 to 72.2 compared to the integrating weight calculator.

The D-term of a PID controller has a dampening effect for $K_\mathrm{d} > 0$. However, in case of a sudden overflux condition a dampening effect is not desired because it would prolong the time until the overflux condition is eliminated. Therefore, the effect is reversed by choosing $K_\mathrm{d} < 0$. The PD weight calculator was tested in a range from 0 to −6 with a step size of 0.01 for the K_d value. An overflux condition before the occurrence of clouds was caused by values below −2 and thus, these values are not appropriate. The μ_l20 and the μ_cloud values were almost not influenced by values between 0 to −2 for K_d. They deviated not more than 0.02 % and 0.04 % respectively compared to the parameter combination with $K_\mathrm{d} = 0$. The D-term can reduce the k_noc value more than the P-Term. The best k_noc value is 69.2. Thus, 5.6 steps faster than the integrating weight calculator. The results for the best k_noc value and other criteria are listed in Tab. 4.10.

Table 4.10: Comparison of different weight calculators.

Weight calculator	k_noc	μ_cloud	μ_l20	K_p	K_i	K_d	p
I	74.8	46.83 %	92.15 %	0	0	1	1
PI	72.2	46.83 %	92.14 %	0.3	0	1	1
ID	69.2	46.84 %	92.15 %	0	-0.3	1	1

4.5.2 Conclusions

In conclusion, the PID weight calculator can enhance the reaction with respect to overflux conditions without impeding the convergence value. However, finding a suitable parameter combination can only be done by trial and error and therefore, may be cumbersome and time-consuming.

5 Controller Evaluation and Performance Analysis

In this chapter, the proposed control algorithms are tested and compared to each other in a simulation environment. To test and compare the controllers, the small reference plant and the test cases that were used to optimize the static optimal control (see section 4.1) are used. The first four subsections present the results for each controller and compare them to each other. Subsequently, the Static Optimal Control is investigated with respect to its computational performance. Lastly, test results of the solar tower in Jülich are presented and an overall conclusion is given.

5.1 Results Vant-Hull

The Vant-Hull control is only tested with test case 2 because the controller can only react to overflux conditions. The controller has no parameters to adjust and is therefore solely tested once. The overflux of 25 kW/m^2 is eliminated within 35 control steps and resulted in a final relative power of 61.78 %. In every control step one heliostat is removed from the receiver. Thus, 65 heliostats focus on the receiver when the overflux condition is removed.

The development of the relative power and the sum of the overflux as well as the first and last flux density distribution is shown in Fig. 5.1. The plot of the relative power is almost a straight line because the reflected power of the heliostats only differs slightly. Each removed heliostat decreases the overflux more in earlier than in later steps due to the fact that fewer bins have an overflux at the end.

5.2 Results Hot Cold

The Hot Cold algorithm is validated with test case 1 and 2. Test case 3 is not tested because the algorithm was implemented for the use with STRAL 2, which is not capable of simulating clouds. Moreover, the Hot Cold algorithm has no parameters to adjust.

In test case 1, the controller stops after 17 iterations. The final relative power is 90.56 %, which is 2.11 % higher than the initial relative power. The development of the relative power and the sum of the overflux as well as first and last flux density distribution is shown in Fig. 5.2. As can be seen, the power increases almost steadily without exceeding the allowed

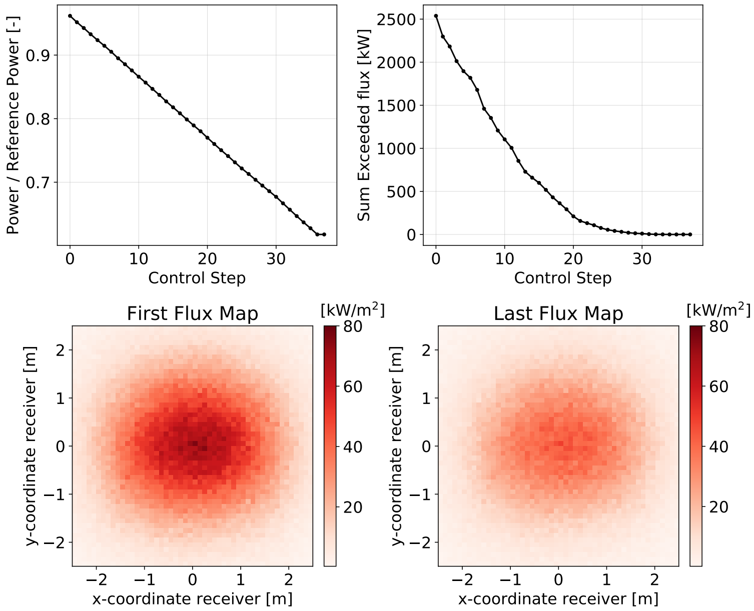


Figure 5.1: Plots of the Vant Hull control for test case 2.

flux density. As a result of the increasing power, the last flux density distribution is more concentrated than the first one.

The overflux condition in test case 2 is eliminated after 61 iterations and the final flux density distribution has a relative power of 90.53 %. An overview of the results is displayed in Fig. 5.3. The overflux decreases at step 41 the first time slightly below 0. Then the algorithm tries to move the heliostats from cold to hot spots to increase the power. Due to the fact that the flux density simulation of the controller differs from the one of the plant system, the controller executes shifts, which lead to a slight overflux. The algorithm changes then to shifting from hot to cold areas again. This leads to a behavior where the algorithm oscillates around an overflux value of 0. The algorithm finally does not find a valid shift from a cold to a hot spot at step 61 and comes to a hold. However, from step 41 on, the controller decreases the relative power around 1.25 %, from 91.78 % to 90.53 %.

5.3 Results Static Optimal Control

The results for the Static Optimal Control are taken from the optimization of the controller in chapter 4. The best results for the criteria are listed in Tab. 5.1. With μ_{120} values of 92.02 % to 92.50 % the controller achieves results close to the maximum relative power of 92.74 % (see section 4.1). The e_{mse} value of test case 3 can not be compared to the other test cases due to the fact that they yield a different interpretation. It has to be considered that these values

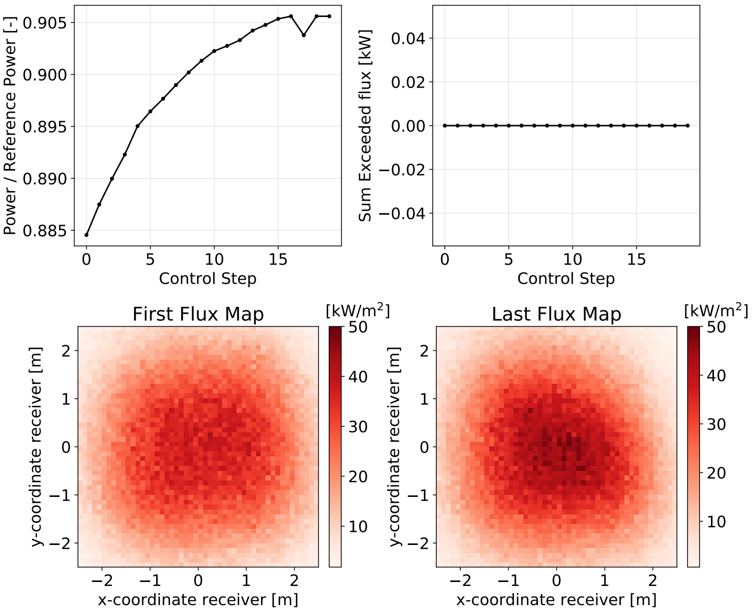


Figure 5.2: Plots of the Hot Cold control for test case 1.

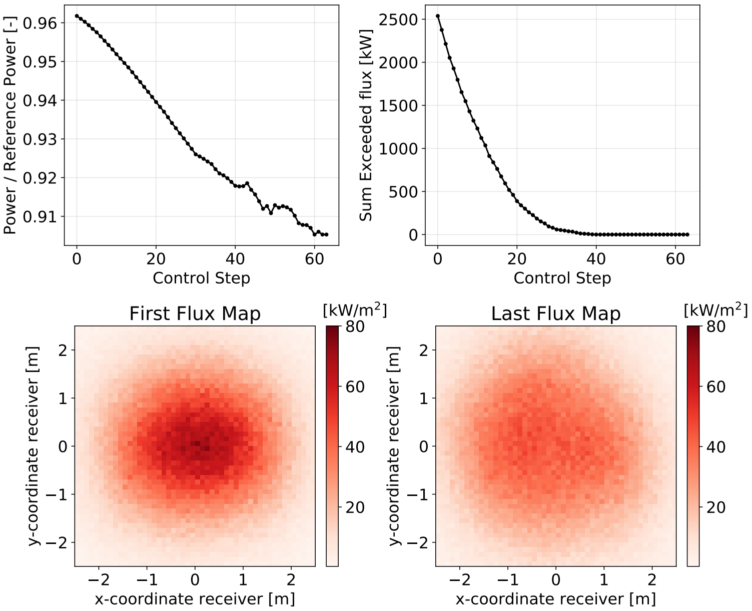


Figure 5.3: Plots of the Hot Cold control for test case 2.

Table 5.1: Results of the Static Optimal Control for all test cases.

Test Case	e_{mse}	μ_{f10}	μ_{l20}	k_{noc}	μ_{cloud}
1	0.635 %	91.72 %	92.16 %	-	-
2	0.687 %	90.99 %	92.02 %	-	-
3	8.950 %	-	92.50 %	0	48.04 %

are subject to different parametrization. The control designer must adapt the parametrization with regard to the criteria that are important to meet the requirements of the controller.

5.4 Comparison of the Controllers

Only the Static Optimal Controller is tested under test case 3. Therefore, the controller can only be compared for test case 1 and 2, except for the Vant Hull control, which was tested solely for case 2. The controllers are compared with respect to their final relative power. The performance i.e. the settling time is not compared. To compare these, the Vant-Hull and the Hot Cold algorithm need to be further enhanced with respect to their performance.

An overview of the final relative powers for test case 1 and 2 is given in Tab. 5.2. For the Static Optimal Control, the μ_{l20} of the best weighted maximum flux objective function is considered as the final relative power. As can be seen, the Static Optimal control yields the best results for all test cases. The Vant-Hull algorithm performances worst with a gap of almost 30 % to the other two controllers. The Hot Cold controller comes close to the results of the Static Optimal Control, only with a difference of 1.60 % and 3.57 % with respect to test case 1 and 2. For test case 2, the Hot Cold controller yields a first valid solution (without overflux) at step 41 with a relative power of 91.78 % as stated in section 5.2. If this value is considered as final relative power, the gap to the Static Optimal Control is reduced to 0.24 %.

Table 5.2: Comparison of the final relative powers for the different control strategies.

Test Case	Vant Hull	Hot Cold	Static Optimal Control
1	-	90.56 %	92.16 %
2	61.78 %	90.53 %	92.02 %

5.5 Performance of the Static Optimal Control

The performance of the algorithm i.e. the computational time is tested on a Windows 7 (64 bit) PC with an Intel(R) Xeon(R) E5-2687W 0 @ 3.10 GHz CPU using 4 cores and the

NVIDIA GeForce RTX 2080 Ti GPU. The algorithm as well as the used STRAL software is implemented in C++. The GPU code enables to calculate the paths of multiple ants at the same time on thousands of cores. The most time-consuming part during the calculation of the path for an ant is the calculation of the objective value. To calculate the objective value, the total flux density distribution is needed, which is calculated by summing up the precalculated flux density distributions of the respective heliostats. In addition to distributing the path calculation for different ants, this summation is also distributed to different cores for each bin. Thus, the computational time is further decreased.

The performance is evaluated for the weighted flux objective function and the integrating weight calculator for test case 1. The controller executed a control step within an average time of 0.48 s. Thus, the performance of the controller is real time capable because the computational time plus the maximum time to move the heliostats of 2.79 s is less than the upper limit of 5.0 s for the sample time (see subsection 3.2.2).

As can be seen in Tab. 5.3, using a GPU for 50 ants per iteration yields only a small improvement of factor 2.4 with respect to a CPU. If the number of ants is small, the GPU uses only a part of its cores. Therefore, when increasing the number of ants, the computational time on the GPU rises only slightly due to fact that many cores were unused before and thus, most tasks can still be done simultaneously. For example, by increasing the number of ants by a factor of 100 from 100 to 10000 ants, the computational time doubles. In contrary, the computational time on the CPU rises proportionally with the number of ants. Thus, the benefit of using a GPU unfolds with a high number of ants. For example, using 10000 ants, the GPU is about 215 times faster than the CPU.

Using a higher number of ants can lead to a better solution. Or the number of iterations can be reduced when raising the number of ants while the quality of the solution does not change. However, due to a not found error in the implementation, the GPU results differ from the CPU results. Thus, these statements can not be validated. However, the algorithm for the GPU is still representative to evaluate the performance of the algorithm.

Furthermore, it is possible to decrease the computational time by grouping the heliostats i.e. allocating one aim point to a group of heliostats. Then, the GPU should allow to simulate a commercial plant with tens of thousands of heliostats without significantly increasing the computational time. However, this was not tested within the scope of this work.

Table 5.3: Comparison of the computational time using a CPU (4 cores) and the CPU (4 cores) plus a GPU.

No. of iterations	No. of ants	Mean time CPU [s]	Mean time GPU [s]	Factor CPU/GPU
50	50	1.14	0.48	2.41
50	100	2.23	0.50	4.49
50	1000	22.17	0.57	38.91
50	10000	221.55	1.03	215.44

5.6 Tests at the Solar Tower in Jülich

The Static Control algorithm is tested at the solar tower in Jülich to analyze the capability of the controller under real conditions. The receiver is a rectangular receiver, which is bend outwards and works with air as heat transfer medium. In total, the receiver has a height of 5.22 m and a width of 4.35 m. It consists of ceramic absorber structures, which are aligned in a grid of 30 horizontal and 36 vertical structures. The heliostats concentrate the solar radiation onto these structures to heat them. Each structure aspires air and transfers the absorbed energy to the air. The heated air can be used for an energy conversion process.

The heliostats have a rectangular shape and consist of 4 facets each. They move bidirectionally in elevation and azimuth axis. Compared to other heliostats, they have a relatively small reflective area of 8 m^2 [8]. Due to reconstruction work, replacement of drives and controllers, only 39 heliostats of the 2153 heliostats [8] were available for the tests.

A camera measures the light intensity on each absorber structure. Based on the light intensities, the current flux density distribution is calculated by an algorithm. During the test, the allowed flux density distribution is changed to simulate conditions with a too high and a too low flux density as well as dynamic changes. When the allowed flux density changes, the reference flux density distribution is scaled in a way that its maximum is equal to the allowed flux density. As an example, the used reference flux density distribution for an allowed flux density of 5 kW/m^2 is shown in Fig. 5.4a.

The controller was tested with the weighted flux objective function ($p = 100$) and the integrating weight calculator ($K_\mathrm{i} = 20$) as well as the ant colony optimization parameters listed in Tab. 4.4 (see section 4.2).

5.6.1 Controller Adaptions

The controller had to be adapted in order to work under real conditions. Depending on the sun position, the flux density of the simulation within the controller was almost twice as high as the measured flux density. Due to the high flux densities and the relative small allowed flux density, the optimization algorithm could not find a valid solution without an overflux in the simulation. If then the heliostats aim more on one side, the weights are adapted so that they are higher on the other side. Thus, in the next step the controller focus the heliostats more on the other side. This happens in a loop and it comes to an oscillating behavior. Without strong peak areas, it does not come to an oscillating behavior. High peak areas, which exceed the allowed flux density, are usually distributed due to the penalty term. However, because the overflux was present at almost every bin, the penalty term had no significant effect. Therefore, an oscillating behavior was recognized due to the deviation of the measured and the simulated flux density distribution.

To reduce the deviation of the measured and the simulated flux density distribution and thus the oscillating behavior, a correction coefficient was introduced. For the calculation of the correction coefficient, a mean value of the flux densities is calculated for each 3×3 grid on the flux density map. Then, the maximum mean value of the measured flux density distribution μ_m is divided by the simulated maximum mean value μ_sim to yield the correction coefficient c:

$$c = \frac{\mu_\mathrm{m}}{\mu_\mathrm{sim}} \tag{5.1}$$

The coefficient is multiplied with the flux densities of the simulation so that the deviation is reduced. The deviation was dependent on the time of day. In the morning and in the afternoon, the correction coefficient decreased to a value of 0.5 while the deviation at noon was relatively small with a coefficient of about 0.95. After the tests, it was noticed that the simulation was based on a heliostat field with 48 instead of 39 heliostats. Additionally, the direct normal irradiation (DNI) was set higher in the simulation than the measured DNI. Both these factors account for a part of the deviation. There may be other reasons, which could not be found until the end of this work.

A second problem was the movement of the heliostats. They moved slower than expected and often readjusted themselves, which led to a constantly changing flux density. The readjustment is necessary because the heliostats often move to far or not far enough. Furthermore, the sun position changes during the control, which changes the incident angle of the solar radiation. The change in the incident angle shifts the aim point of the heliostat. Hence, the heliostats also readjust to keep the aim point at the same position. Additionally, the 39 heliostats are connected to two bus systems (around 20 heliostats are connected to one bus system). The cycle time is 100 ms. Thus, the twentieth heliostats receives the signal after 2 s. The combination of the readjustment and the delay through the bus system led to a steady heliostat movement. However, these reasons seemed not to explain fully the slow heliostat movement, but further tests to investigate this issue could not be conducted until now. To compensate the steady movement partly, the sampling time was set to 20 s to give them sufficient time to readjust. However, this sample time is four times higher than the calculated upper sampling time in subsection 3.2.2.

5.6.2 Test Results

The results for one test run are shown in Fig. 5.4b. The plots are divided into 5 parts. The change to another part is marked by the change of the allowed flux density (red line). In the left plot, the relative power as well as the allowed flux density is shown over the amount of control steps. Here, the reference power is set in a way that the maximum relative power is 100 % during the test run. On the right side, the sum of the overflux and the allowed flux density is plotted. The first part is the initial phase. At first the heliostats aim at a point offside of the receiver. The allowed flux density is set to 30 kW/m^2. This flux density cannot be exceeded, even when all heliostats aim at the center. Therefore, the controller aims all heliostats to the center to achieve the maximum power. Then, the actual test starts. After 5 steps, the allowed flux density is reduced to 5 kW/m^2. The sum of the overflux jumps due to a lower allowed flux density but is almost eliminated after 5 steps at step 10. Nonetheless,

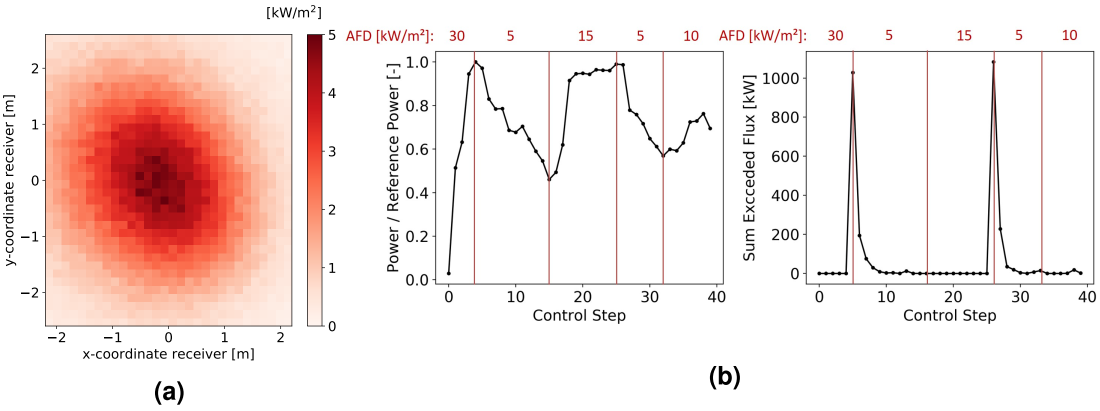


Figure 5.4: Reference flux density distribution in Jülich with respect to an allowed flux density (AFD) of 5 kW/m^2 (a) and evaluation plots of the test run in Jülich (b).

after step 10 the relative power decreases further. This is probably due to the reason that the overflux is not completely eliminated and the controller tries to find a better solution. However, the found solutions decrease the relative power more than desired. After 15 steps, the allowed flux density is raised to 15 kW/m^2. The controller increases the relative power, while not exceeding the allowed flux density as desired. Then, at step 25 the allowed flux density is again decreased to 5 kW/m^2. The behavior of the controller is similar to the previous decrease to 5 kW/m^2. At last, the allowed flux density is increased to 10 kW/m^2. The relative power increases as well, but at one control step a slight overflux is caused.

5.7 Conclusions

The results regarding the final relative power of the Hot Cold and Static Optimal Control were compared for two test cases. For test case 2, the controllers were also compared to the existing Vant-Hull control to validate their functionality. In both test cases, the Static Optimal Control yielded the best results. The Hot Cold controller achieved only slightly worse results with a maximal gap of 3.57 % to the final relative power of the Static Optimal Control. In test case 2, the controllers could eliminate the overflux condition but yielded a final relative power, which was almost 1.5 times better than the Vant-Hull control.

Furthermore, the computational performance of the Static Optimal Control was tested on a GPU showing that the controller is real-time capable.

Because the Static Optimal Control achieved the best results, the controller was also tested at the solar tower in Jülich. The Static Optimal Control increased the relative power, when the allowed flux density was increased almost without exceeding the allowed flux density. Furthermore, after a decrease of the allowed flux density, the controller reduced most of the overflux within one control step. Thereafter, it further reduced the overflux but not always completely. Overall, this is a promising tendency. However, for a commercial operation overflux condition need to be eliminated within one control step as stated in subsection 3.2.2. This may be realized with better fitted parameters of the controller and different objective functions or weight calculators. Furthermore, there must also be found a solution for the readjusting heliostats in order to reduce the sample time.

With respect to the objectives of this thesis, it can be concluded that both controllers (Hot Cold and Static Optimal Control) increase the efficiency of the solar power tower plant, while restricting the flux density. Furthermore, both controllers can compensate for mirror errors. However, only the Static Optimal Control was tested under the influence of clouds, which the controller compensated for successfully. Within this work it could not be tested, if the controllers also work for different receiver types. They were only tested for a rectangular receiver.

6 Summary and Outlook

In this work, two novel closed-loop control strategies for heliostat aiming of solar power tower plants were presented. The first control algorithm, called Hot Cold control, is based on an existing control approach, which shifts the heliostats from hotter to colder zones and vice versa based on measurements of the receiver temperature. This strategy was adapted to work with the measurements of flux densities. The second approach, which is called Static Optimal Control, modifies the existing open-loop ant colony optimization algorithm for heliostat aiming to work as a closed-loop control. Both approaches were tested with up to three test cases and compared to the existing Vant-Hull control to validate their functionality. The first two test cases evaluated the controllers capacity to compensate disturbances like mirror errors and to react to overflux conditions. The last test case investigated the controller's reaction to dynamic disturbances like clouds. The best results yielded the Static Optimal Control. However, the Hot Cold algorithm was only slightly worse. Both controllers achieved a relative power, which was almost 1.5 times better than the Vant-Hull control.

The Static Optimal Control requires a high computational effort due to its internal ant colony optimization. To test if the controller is real-time capable, the algorithm was implemented on a GPU. An evaluation yielded that the computational time is sufficiently small and in the boundary of the calculated lower and upper sample time for the solar tower in Jülich.

The Static Optimal Control was also tested at the solar tower in Jülich. Here, the run consisted of changing the allowed flux density in order to test the controller's behavior with respect to overflux conditions and its capability to raise the power on the receiver. The controller showed a promising tendency but its reaction to overflux conditions needs to be further enhanced to work in a commercial plant.

Overall, the developed controllers could fulfill the objectives of this thesis, namely increasing the efficiency, restricting the flux density and compensating for disturbances such as mirror errors. However, only the Static Optimal Control was tested successfully under clouds and both controllers were not tested with different receiver types.

As mentioned in section 3.5, the Hot Cold controller can be further enhanced by shifting several heliostats in one control step. Moreover, the algorithm should also be tested for more cases like under the influences of clouds.

Based on this work, further work can be done in the direction of enhancing the robustness of the Static Optimal Control as well as extending its application areas. Two ideas to enhance the robustness of the controller are presented in the following paragraphs.

Increased Robustness

The controller in this work was only tested for some test cases and therefore, may react badly in other situations. Therefore, the controller should be tested under more test cases e.g. the impact of a tracking error or the simulation of a slowly passing cloud. However, a more comprehensive testing does not guarantee a robust behavior of the controller i.e. that no critical overflux condition can occur. This is due to the non-deterministic optimization algorithm, which makes a prediction of the behavior of the controller impossible. To enable a robust control, a switching control mode can be introduced. The controller would switch to a second control algorithm if an overflux condition occurs, which can not be handled fast enough by the Static Optimal Controller. The second controller would be a robust controller (e.g. the Vant-Hull controller), which would bring the system back to a safe state. When the system is in a safe state, the controller would switch back to the Static Optimal Control. This idea is inspired by the Simplex architecture [43].

Cloud Prediction

There exist methods to predict upcoming clouds and their influence on the heliostats e.g. by detecting the clouds with the Nowcasting System. These predictions could be included in the plant system model of the controller to generate more precise and robust predictions. However, the weights try to compensate also modeling errors. If the plant system is changed, the weights may have to be adjusted accordingly.

In contrary to the previous presented ideas, the ideas in the following paragraphs regard the application areas of the controller. So far, the Static Optimal Control was only tested on a rectangular receiver simulating the optical site of the plant system. However, the controller can be tested with other plant types and with different plant system models.

Commercial Plants

The controller was only tested on a small plant with 100 heliostats and a rectangular receiver. However, commercial plants usually have tens of thousands of heliostats and a cylindrical receiver. The practicability of the controller with respect to the control quality and the computational time could be tested with these plants, while using a grouping strategy and a GPU.

Extended System Model

The current system model only includes the optical site. However, in real application the thermal site dictates the allowed flux density, which also is not static as assumed in this work. Thus, the controller should be tested in connection with a thermal model and a changing allowed flux density. The model may be directly included in the plant system model used in the controller or be connected externally by just setting the allowed flux density from outside by a thermal model.

Energy Conversion Process

Different energy conversion processes such as for hydrogen production have different receiver types and different demands regarding the flux density distribution. It could be investigated, if the controller works also for these receiver types. A more general formulation of the controller and the control problem may be necessary to cope with different receiver types. However, an adaption of the current controller may also be sufficient.

In conclusion, the Static Optimal Control seems to be promising, but there exists still a lot of possible future work to enhance and extend the functionality of the controller.

A Implementation

The Vant-Hull and the Hot Cold algorithm are implemented in Python because Python is easy to learn and use. However, the Static Optimal Control is implemented in C++ because it is based on the ant colony optimization, which was programmed in C++.

Care was taken to ensure that the code is programmed in a way, which is comprehensible and reusable. To achieve this, the programming followed certain principles such as DRY (Don't repeat yourself!), information hiding and strong coupling (module based coding), which are further explained in [44]. Moreover, the code is written in a self commenting way. Functions and classes are kept small to attain a better readability and reusability of the code. The readability as well as the programming style is further enhanced by taking the Google style guides for Python and C++ [45, 46] into account. The modular structure is achieved by using design patterns such as Dependency Injection, Factory and Adapter (further explanation can be found in [44]). To ensure the correct functionality of the code and to build a safety net as mentioned by Roth [44], unit tests were implemented by using Pytest for Python and Google Test for C++.

Although it was attempted to consider all these principles, guidelines and patterns, the code is certainly not perfect. Due to the limited time, the code is not well commented other than through its self-commenting style. Moreover, the tests in C++ are not very extensive. Furthermore, the programming style has developed and was adopted during the process of implementation. Thus, some parts of the code are written more comprehensive and reusable than others depending on the time when they were coded.

To be able to easier reproduce the algorithms in Python and C++, a short explanation of the structure of the implementation is given in the next two sections.

A.1 Structure of the Vant-Hull and Hot Cold Control

A class diagram of the Vant-Hull and Hot Cold algorithm is shown in Fig. A.1. The Controller class consists of a SolarTowerData, a SolarTowerSystem and a Tracker class as well as some internal attributes and functions, which are not listed in the diagram. The SolarTowerData class saves information such as the aim point coordinates and configuration as well as the bin position. It can perform various tasks on these data, which are helpful for the controller. The SolarTowerSystem is a class, which can communicate with the solar tower. For example, it can query the current flux density distribution as well as set and get the current aim point

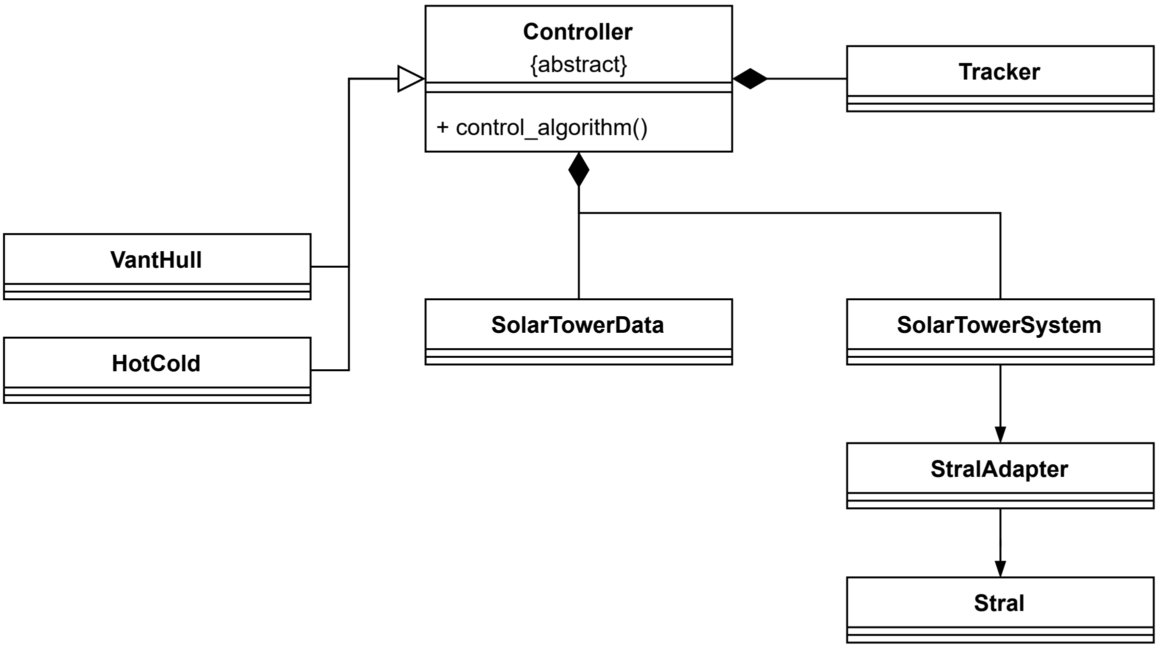


Figure A.1: Class diagram of the Vant-Hull and Hot Cold algorithm.

configuration. The Tracker class saves data like the current power, intercept power and current flux density distribution during control.

The Controller is an abstract class with the abstract method control_algorithm(). This method is overwritten by the derived VantHull and HotCold classes in order to implement the functionality of these algorithms.

A.2 Structure of the Static Optimal Control

In Fig. A.2, the class diagram of the Static Optimal Control is depicted. The controller consists of the components ObjectiveFunction, WeightEstimator, ErrorSignal, Optimizer, SolarTowerSystem and DataTracker. The ObjectiveFunction, WeightEstimator and ErrorSignal are abstract classes, which contain certain attributes and their respective getter and setter methods. From these classes, child classes are derived, which implement the functionality described in section 3.6. The child class Quadratic also needs the ReferenceOutputGetter interface in contrast to the other ObjectiveFunctions. Each of these abstract classes has its own factory class.

The Optimzer is an interface, which provides functionality such as setting the objective function as well as starting the optimization. The Optimizer in this work is the ant colony optimization as explained in section 3.6. Therefore, the existing ant colony optimization by Belhomme [10] is connected via an adapter to this interface.

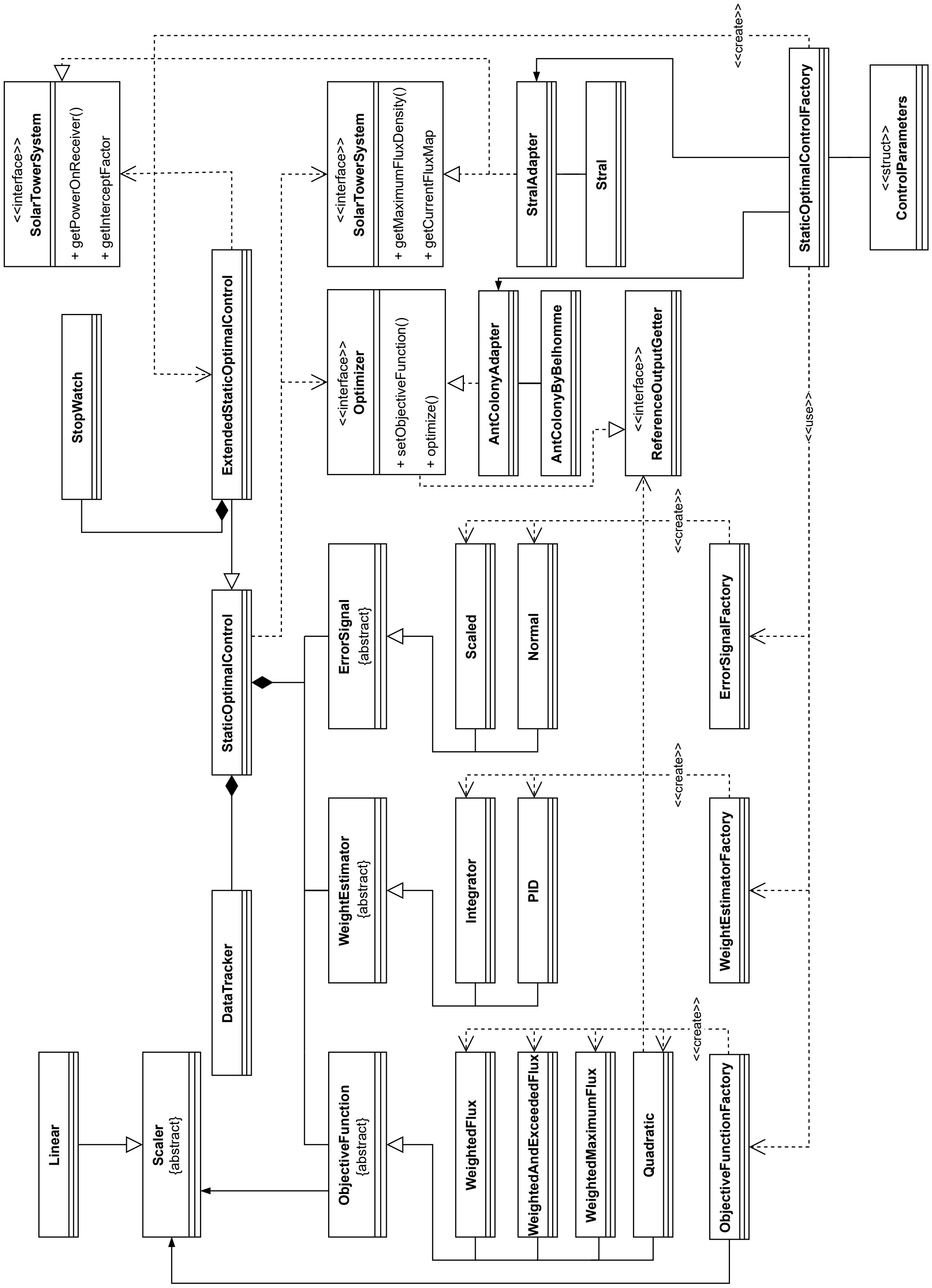


Figure A.2: Class diagram of the Static Optimal Control.

The SolarTowerSystem is an interface as well, which connects the controller to the solar tower by defining functions such as retrieving the allowed flux density, the current flux density distribution or setting the aim point configuration (the latter is not shown in the diagram). Within the simulation, this interface is connected to STRAL via an adapter.

The DataTracker saves data such as the current flux density distribution. This data can be processed by another class or program.

From the StaticOptimalController the ExtendedStaticOptimalController is derived. This class implements some extra functionality for tracking data. It can measure the execution time of one control step through the StopWatch class. Furthermore, it needs the SolarTowerSystem interface, which provides functions to retrieve the current power and the intercept factor.

The ExtendedStaticOptimalController is created by the StaticOptimalControlFactory. This factory class uses the existing factory classes of the components. As inputs, it needs an adapter for the Optimizer, the SolarTowerSystem, the SolarTowerData and the ControlParameters struct. The ControlParameters struct defines all parameters, which can be set within the controller.

The presented classes are the most relevant in context of the Static Optimal Control. However, many more classes were coded to create a simulation environment and provide other functionality.

Bibliography

[1] F. J. GarcÍa-MartÍn et al. “Heuristic knowledge-based heliostat field control for the optimization of the temperature distribution in a volumetric receiver”. In: *Solar Energy* 66.5 (1999), pp. 355–369. ISSN: 0038092X. DOI: 10.1016/S0038-092X(99)00024-9.

[2] Sebastian Oberthür. “Hard or Soft Governance? The EU’s Climate and Energy Policy Framework for 2030”. In: *Politics and Governance* 7.1 (2019), pp. 17–27. ISSN: 2183-2463. DOI: 10.17645/pag.v7i1.1796. URL: https://www.cogitatiopress.com/politicsandgovernance/article/view/1796.

[3] *Statistics | Eurostat*. 2020-12-10. URL: https://ec.europa.eu/eurostat/databrowser/view/nrg_ind_ren/default/table?lang=en (visited on 12/10/2020).

[4] Marco Astolfi et al. “Heliostat aiming point optimization for external tower receiver”. In: *Solar Energy* 157 (2017), pp. 1114–1129. ISSN: 0038092X. DOI: 10.1016/j.solener.2016.03.042.

[5] Adriana Zurita et al. “Assessment of time resolution impact on the modeling of a hybrid CSP-PV plant: A case of study in Chile: Solar Energy, 202, 553-570”. In: *Solar Energy* 202 (2020), pp. 553–570. ISSN: 0038092X. DOI: 10.1016/J.SOLENER.2020.03.100.

[6] *Concentrating Solar Power Projects*. 2020. URL: https://solarpaces.nrel.gov/by-status/operational (visited on 09/24/2020).

[7] *Concentrating Solar Power Projects*. 2020. URL: https://solarpaces.nrel.gov/by-status/under-construction (visited on 09/24/2020).

[8] Gopalakrishnan Srilakshmi et al. “Challenges and opportunities for Solar Tower technology in India”. In: *Renewable and Sustainable Energy Reviews* 45 (2015), pp. 698–709. ISSN: 1364-0321. DOI: 10.1016/j.rser.2015.02.016. URL: http://www.sciencedirect.com/science/article/pii/S1364032115001045.

[9] Sigurd Skogestad and Ian Postlethwaite. *Multivariable feedback control: Analysis and design*. 2. ed., reprint. Chichester: Wiley, 2009. ISBN: 9780470011683.

[10] Boris Belhomme. *Bewertung und Optimierung von Zielpunktstrategien für solare Turmkraftwerke: Zugl.: Aachen, Techn. Hochsch., Diss., 2010*. Berichte aus der Energietechnik. Aachen: Shaker, 2011. ISBN: 9783844002591.

[11] Scott A. Jones et al. *Heliostat cost reduction study*. 2007. DOI: 10.2172/912923. URL: https://www.researchgate.net/publication/255017439_Heliostat_Cost_Reduction_Study.

[12] F. W. Lipps and L. L. Vant-Hull. “A cellwise method for the optimization of large central receiver systems”. In: *Solar Energy* 20.6 (1978), pp. 505–516. ISSN: 0038092X. DOI: 10.1016/0038-092X(78)90067-1.

[13] R. J. Holl. *Definition of two small central receiver systems*. 1978. URL: https://www.researchgate.net/publication/236562393_Definition_of_two_small_central_receiver_systems.

[14] Xiudong Wei et al. “A new code for the design and analysis of the heliostat field layout for power tower system”. In: *Solar Energy* 84.4 (2010), pp. 685–690. ISSN: 0038092X. DOI: 10.1016/j.solener.2010.01.020.

[15] Pascal Richter et al. “Optimization of robust aiming strategies in solar tower power plants”. In: *SOLARPACES 2018: International Conference on Concentrating Solar Power and Chemical Energy Systems*. AIP Conference Proceedings. AIP Publishing, 2019, p. 030045. DOI: 10.1063/1.5117557.

[16] Daniel Maldonado et al. “Evaluation of aim point optimization methods”. In: AIP Conference Proceedings. Author(s), 2018, p. 040025. DOI: 10.1063/1.5067061.

[17] David Barlev, Ruxandra Vidu, and Pieter Stroeve. “Innovation in concentrated solar power”. In: *Solar Energy Materials and Solar Cells* 95.10 (2011), pp. 2703–2725. ISSN: 09270248. DOI: 10.1016/j.solmat.2011.05.020.

[18] Alberto Sánchez-González, María Reyes Rodríguez-Sánchez, and Domingo Santana. “Allowable solar flux densities for molten-salt receivers: Input to the aiming strategy”. In: *Results in Engineering* 5 (2020), p. 100074. ISSN: 25901230. DOI: 10.1016/j.rineng.2019.100074.

[19] Lorin L. Vant-Hull. “The Role of “Allowable Flux Density” in the Design and Operation of Molten-Salt Solar Central Receivers”. In: *Journal of Solar Energy Engineering* 124.2 (2002), pp. 165–169. ISSN: 0199-6231. DOI: 10.1115/1.1464124{\#}.

[20] Ari Rabl. *Active solar collectors and their applications*. New York: Oxford University Press, 1985. ISBN: 9781280523007. URL: http://search.ebscohost.com/login.aspx?direct=true&scope=site&db=nlebk&db=nlabk&AN=168984.

[21] Andreas Neumann et al. “Representative Terrestrial Solar Brightness Profiles”. In: *Journal of Solar Energy Engineering* 124.2 (2002), pp. 198–204. ISSN: 0199-6231. DOI: 10.1115/1.1464880.

[22] Dirk Abel. *Regelungstechnik*. 35th ed. Verlagsgruppe Mainz GmbH Aachen, 2011.

[23] Jürgen Adamy. *Nichtlineare Systeme und Regelungen*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2018. ISBN: 978-3-662-55684-9. DOI: 10.1007/978-3-662-55685-6.

[24] Robert Flesch et al. “Towards an optimal aiming for molten salt power towers”. In: *Solar Energy* 155 (2017), pp. 1273–1281. ISSN: 0038092X. DOI: 10.1016/j.solener.2017.07.067.

[25] Alberto Sánchez-González, María Reyes Rodríguez-Sánchez, and Domingo Santana. “Aiming strategy model based on allowable flux densities for molten salt central receivers”. In: *Solar Energy* 157 (2017), pp. 1130–1144. ISSN: 0038092X. DOI: 10.1016/j.solener.2015.12.055.

[26] Francisco J. Collado and Jesus Guallar. “A two-parameter aiming strategy to reduce and flatten the flux map in solar power tower plants”. In: *Solar Energy* 188 (2019), pp. 185–189. ISSN: 0038092X. DOI: 10.1016/j.solener.2019.06.001.

[27] Daniel Maldonado, Robert Flesch, and Peter Schwarzbözl. “Hybridization of Aim Point Optimization Methods for Solar Tower Power Plants”. In: *MATHMOD 2018 extended abstract volume*. Ed. by Felix Breitenecker et al. Argesim Report. Vienna: ARGESIM Publisher, 2018, pp. 39–40. ISBN: 9783901608919. DOI: 10.11128/arep.55.a55230.

[28] Adrien Salomé et al. “Control of the flux distribution on a solar tower receiver using an optimized aiming point strategy: Application to THEMIS solar tower”. In: *Solar Energy* 94 (2013), pp. 352–366. ISSN: 0038092X. DOI: 10.1016/j.solener.2013.02.025.

[29] Saeb M. Besarati, D. Yogi Goswami, and Elias K. Stefanakos. “Optimal heliostat aiming strategy for uniform distribution of heat flux on the receiver of a solar power tower plant”. In: *Energy Conversion and Management* 84 (2014), pp. 234–243. ISSN: 01968904. DOI: 10.1016/j.enconman.2014.04.030.

[30] Boris Belhomme, Robert Pitz-Paal, and Peter Schwarzbözl. “Optimization of Heliostat Aim Point Selection for Central Receiver Systems Based on the Ant Colony Optimization Metaheuristic”. In: *Journal of Solar Energy Engineering* 136.1 (2014). ISSN: 0199-6231. DOI: 10.1115/1.4024738.

[31] Thomas Ashley, Emilio Carrizosa, and Enrique Fernández-Cara. “Optimisation of aiming strategies in Solar Power Tower plants”. In: *Energy* 137 (2017), pp. 285–291. ISSN: 03605442. DOI: 10.1016/j.energy.2017.06.163.

[32] Sascha Kuhnke et al. “Robust optimal aiming strategies in central receiver systems”. In: *Renewable Energy* 152 (2020), pp. 198–207. ISSN: 09601481. DOI: 10.1016/j.renene.2019.11.118.

[33] Jesús García et al. “Heat Flux Distribution Over a Solar Central Receiver Using an Aiming Strategy Based on a Conventional Closed Control Loop”. In: *2017 proceedings of the ASME 11th International Conference on Energy Sustainability/15th Fuel Cell Science, Engineering, and Technology Conference*. New York, N.Y.: The American Society of Mechanical Engineers, 2017. ISBN: 978-0-7918-5759-5. DOI: 10.1115/ES2017-3615.

[34] Jesús García et al. “Multivariable Closed Control Loop Methodology for Heliostat Aiming Manipulation in Solar Central Receiver Systems”. In: *Journal of Solar Energy Engineering* 140.3 (2018). ISSN: 0199-6231. DOI: 10.1115/1.4039255.

[35] Jesús García et al. “Dynamic performance of an aiming control methodology for solar central receivers due to cloud disturbances”. In: *Renewable Energy* 121 (2018), pp. 355–367. ISSN: 09601481. DOI: 10.1016/j.renene.2018.01.019.

[36] M. Kiera. “Heliostat Field: Computer Codes, Requirements, Comparison of Methods”. In: *undefined* (1989). URL: `https://www.semanticscholar.org/paper/Heliostat-Field%3A-Computer-Codes%2C-Requirements%2C-of-Kiera/e1678d349a2cff799e542968b39dc8da29759c6d`.

[37] Peter Schwarzbözl, Robert Pitz-Paal, and Mark Schmitz. “Visual HFLCAL - A Software Tool for Layout and Optimisation of Heliostat Fields”. In: *Proceedings*. Ed. by Thomas Mancini and Robert Pitz-Paal. 2009. ISBN: 9783000287558. URL: `https://elib.dlr.de/60308/`.

[38] Jürgen Adamy. *Fuzzy Logik, neuronale Netze und evolutionäre Algorithmen*. 5. überarbeitete Auflage. Berichte aus der Steuerungs- und Regelungstechnik. Düren: Shaker Verlag, 2019. ISBN: 9783844069297.

[39] Y. Tan et al. “Extremum seeking from 1922 to 2010”. In: *Proceedings of the 29th Chinese Control Conference*. 2010, pp. 14–26. ISBN: 2161-2927.

[40] Jingqing Han. “From PID to Active Disturbance Rejection Control”. In: *IEEE Transactions on Industrial Electronics* 56.3 (2009), pp. 900–906. ISSN: 1557-9948. DOI: `10.1109/TIE.2008.2011621`.

[41] Futao Zhao and Yash P. Gupta. “A simplified predictive control algorithm for disturbance rejection”. In: *ISA Transactions* 44.2 (2005), pp. 187–198. ISSN: 00190578. DOI: `10.1016/S0019-0578(07)60177-3`.

[42] B. Belhomme et al. “A New Fast Ray Tracing Tool for High-Precision Simulation of Heliostat Fields”. In: *Journal of Solar Energy Engineering* 131.3 (2009). ISSN: 0199-6231. DOI: `10.1115/1.3139139`.

[43] D. Seto et al. “The Simplex architecture for safe online control system upgrades”. In: *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No.98CH36207)*. 1998, 3504–3508 vol.6. ISBN: 0743-1619. DOI: `10.1109/ACC.1998.703255`.

[44] Stephan Roth. *Clean C++: Sustainable Software Development Patterns and Best Practices with C++ 17*. Apress, 2017. ISBN: 9781484227923.

[45] *Google C++ Style Guide*. 2020-09-25. URL: `https://google.github.io/styleguide/cppguide.html` (visited on 10/26/2020).

[46] styleguide. *styleguide*. 2020-09-25. URL: `https://google.github.io/styleguide/pyguide.html` (visited on 10/26/2020).