University of Cologne
Faculty of Mathematics and Natural Sciences
Mathematical Institute

Online Optimization - Cleaning strategies for CSP plants

BACHELOR'S THESIS

10th May 2021

Author
Simone Horstmann

Supervisor
Dr. Kevin Schewior

Contents

Abstract

Summary of the thesis

Due to the high solar irradiation Concentrated Solar Power plants are predominately located in arid areas, where the problem of soiling is often encountered. Dust and sand deposits on the mirror surfaces reduce the reflectivity and thus mirrors must be cleaned frequently in order to maintain the efficiency of the CSP plant. As excessive water consumption is a problem in arid areas, this work investigates a method to reduce consumption by scheduling cleaning in an optimal way.

The thesis outlines a new methodology to derive an online algorithm that computes a water-saving cleaning schedule for CSP plants. The optimization algorithm is based on a Markov decision process, which evaluates the sequence of various cleaning options and derives a cleaning policy based on the forecasted soiling rate for a time horizon of up to ten days. A new aspect implemented within this algorithm are space resolved cleanliness values that are used to assess the need for cleaning certain parts of the solar field and thus maximize the energy yield of the whole plant.

A technical overview of the plant is presented first, as well as an assessment of characteristics and requirements to the algorithm. Also the definition of a Markov decision process is given and the fundamental algorithm is described. The second part of this thesis is devoted to the formulation of the problem as a Markov decision process and the developed algorithm. Finally this algorithm is evaluated, by applying a quantitative analysis of multiple simulations with a variation of the parameters.

The results of this study indicate that a Markov decision process is an appropriate mathematical formulation of the problem at hand. Moreover, the algorithm is able to achieve the following two improvements, compared to a constant cleaning schedule which is currently the common practice. Either a reduced water consumption by up to 20% could be achieved, while maintaining the energy output, or a stronger water reduction of 40% to 70% could be achieved with a minor energy yield reduction of 1% to 2%. Furthermore the use of a single parameter was found to be suitable for regulating the extent to which the optimizer reduces the number of cleaning actions and thus the consumption of water.

Acknowledgements

This work was created within the scope of the SOLWARIS project; funded by the European Union within the Horizon 2020 research program grant agreement nr. 792103. Further content of this work was created with support of the EnSysInt project sponsored by HGF.

I want to thank my fellow colleagues Nathalie Hanrieder and Fabian Wolfertstetter for providing me with insight on many meteorological topics and supporting me with measurement data, which enabled me to formulate research based theories that were not within my expertise.

Furthermore I also want to thank my colleagues at the German Aerospace Center, Matthias Loevenich and Sebastian Müller for always supporting me with their technical understanding of the CSP systems.

A thanks to Dr. Philipp Ross and Cosima Gräfin von Hohenthal for proofreading this thesis and their helpful comments.

I want to express my sincere and special thanks to my advisor Lisa Kaborn, at the German Aerospace Center for her dedicated support, for sharing her profound knowledge of the CSP systems with me and guiding me throughout this project.

Finally I want to thank my advisor Dr. Kevin Schewior, at the Mathematical Institute of the University of Cologne, for having such an interest in this research and constantly supporting me throughout this project.

Glossary

A	Action space
S	State space
α	Alpha value
β	Transition probability
η_{clean}	Cleanliness efficiency
η_i	Cleanliness efficiency value for sector i
$cleanRes$	Cleanliness resolution
fa	Forecast accuracy
fd	Forecast deviation
h	Optimization horizon
$p(s,a,s')$	Transition probability for tuple
$Q(s,a)$	Value of taking a specific action a while being in state s
$r_s()$	One-stage reward
$V(s)$	Value of being in state s
$V_0(s)$	Terminal reward function for state s
CSP	Concetrated Solar Power
DNI	Direct normal irradiation
MDP	Markov decision process
NCR	Natural cleaning rate
SF	Solar Field
SR	Soiling rate

1 Introduction

Concentrated solar power is a clean and efficient way to use the energy provided by the sun. Concentrated Solar Power (CSP) plants are particularly productive in areas with high irradiation, often being arid areas. Many plant locations face the problem of continuous soiling, i.e. the deposition of dust and sand on the mirror surfaces, which leads to a lower efficiency of the CSP plant. Therefore, mirrors are frequently cleaned to maintain the yield. However, the cleaning process consumes water, which might be rare in the CSP plants location. To remain a way of sustainable energy production, only a minimum amount of water should be consumed to ensure the required energy output. This can only be achieved if cleaning is performed only when the net effective yield increase is strongest, i.e. when the cleaning process is most efficient. This arises the question how the objective of both water reduction and energy output can be formulated and optimized. Within this thesis a method is presented that both describes the objective function and converts it into a mathematical model that can be optimized. Based on that, an online algorithm is developed to determine an optimized cleaning schedule for CSP plants. This will be done by means of mathematical optimization, in particular using an approach called Markov decision process.

In recent years the quest for solutions to the problem of soiling has become a topic of discussion within the field of CSP research. A short review is presented over the next paragraphs: Early studies such as [BF81] have evaluated the complex influence of soiling on the yield of a CSP plant. They addressed the interaction between meteorological phenomena such as rain, wind and dust composition of the atmosphere and the soiling process. An approach to solve the problem of soiling using a constant cleaning schedule was introduced. The interval of this constant cleaning schedule is optimized specifically to solve the tradeoff between energy production and cleaning costs.

A threshold based cleaning schedule has been introduced in [WWD+18] and was compared to a constant cleaning schedule. This strategy initiates a cleaning action whenever the average reflectance of the mirrors within the solar filed is below a certain value. Within this study, the different strategies have been compared in terms of their financial profit increase and a possible yield increase of 1.71 % for the threshold based schedule was found. For this, the loss induced by a lower reflectance of the mirrors and therefore a lower energy production was set against the cost of cleaning, regarding both fixed and variable cleaning costs. The author further discussed the effect of dumping on the efficiency of a cleaning strategy. Dumping is the event in which energy cannot be used by the power plant due to the temperature limits and storage capacity. An optimized schedule should consider these events in its planning in order to save water.

Within [TWW+19] the cleaning schedule was optimized using an Artificial Neural Network (ANN) algorithm, which was trained based on a 25 years meteorological dataset and 5 years soiling dataset. The soiling dataset was artificially extended using the correlation of rain and soiling, which was found in the existing datasets. Within the algorithm the cleaning decision was made, considering a soiling forecast with different time horizons. The ANN was trained based on the input datasets and the relative profit increase was used as a reward to the algorithm. It was shown that the algorithm can increase the relative profit by $1.28 - 1.36\%$.

The authors of [TBCW+17] and [TBCP+20] first introduced the concept of Condition-based cleaning to this problem. Therefore, the cleaning decision is made based on the comparison of the current reflectivity to a time-varying threshold. This time-varying threshold does consider the stochastic nature of the soiling process and its dependency on seasonal variation. As a methodology to derive the cleaning policy a Markov decision process has been formulated. The author extended this model in [TBCP+20] to derive a reflectance-based model that furthermore accounts for the space-varying soiling of different solar field parts. Both publications have been essential for the formulation of the idea as well as the development of the online algorithm within this thesis.

Besides the research on cleaning strategies, different tools are currently in development to facilitate the measurement of soiling and the analysis of its effect on the pant's performance, or enable the forecasting. One such tool, developed in the SOLWARIS project, consists of multiple soiling sensors distributed within the solar field, that enable the spacially resolved measurement of mirror-reflectance [CdGV] and [VZS+20]. This allows to apply cleaning only to parts of the solar field where it is most needed. Furthermore, a soiling forecast is being developed that can facilitate planning ahead, to include future events in the current cleaning decision. The availability of both tools for future use has led to the principles of the optimizer developed in this thesis, which are the integration of space-resolved reflectance values as well as soiling forecasts.

The goal of this thesis is to minimize the water consumption of the cleaning process, while maintaining or, if possible, maximizing the energy output. Instead of a predefined schedule, the optimizer developed in this thesis makes a cleaning decision individually based on the current reflectance of the mirrors as well as the forecasts for soiling and meteorological data made available. Provided with this input data the optimizer analyses different scenarios of which some are more favorable than others. The optimization algorithm is used to chose between alternative cleaning options while targeting to reduce the water consumption.

Compared to a fixed cleaning schedule, the optimizer was able to reduce the water consumption, whereas no significant increase of the energy output could be achieved. The performance of the optimization is dependent on different parameters such as the forecast horizon or the resolution of the input parameter.

The field of mathematical optimization offers a wide variety of techniques to solve technical problems. In order to find the right technique, it is crucial to understand the

practical problem first. Based on that the mathematical model can be formulated and an algorithm can be developed. Therefore, the thesis has been structured in a way that describes this process: In the first chapter an overview of the technical structure of a CSP plant is presented, while focusing on the solar field. The phenomena of soiling and how it affects the plant's performance is discussed. Furthermore the simulation, used to calculate the energy production of a CSP plant as well as the model to describe the effect of soiling on the plants performance is presented. In the third Chapter the principles of a Markov decision process are outlined as well as the available algorithms. Based on the theoretical foundation presented in these two chapters, Chapter 4 introduces the formulation of the problem and the algorithm to approach it. The evaluation of the optimization algorithm is performed in Chapter 5. The thesis is concluded with a presentation of the results of this work and certain aspects which need to be further examined.

2 CSP Plants - Description and modeling

In this chapter a brief overview of Concentrated Solar Power (CSP) technology will be given to provide the reader with information on the technical background of the optimization problem. To convey a better understanding of the dimensions of a CSP plant, the structure and functionality is explained by the example of the Andasol III Power plant shown in Figure 2.1. This CSP plant is located in southern Spain and has a power output of 50 MW, which is a common value for this kind of power plant. Therefore, it is used as reference for this thesis [And08] .

Figure 2.1: CSP plant Andasol III [Source: Marquesado Solar]

2.1 Concentrated Solar Power - The plant and its subsystems

A CSP power plant uses the energy provided by the sun through radiation. Therefore, three main parts are used to convert the energy into electricity: the solar field, the thermal storage and the power block, as can be seen in Figure 2.2

Different types of CSP plants can be distinguished by their focusing systems, being line-focusing or point-focusing systems [PP08]. The latter are for example solar tower

plants that use heliostats installed in a circle around a tower, onto which the solar irradiation is focused. In contrast, within line-focusing systems the sun is focused on an absorber tube, installed at the focal line of the mirrors. One common type of such a collector system are parabolic through collectors (PTC), which will be considered in this thesis.

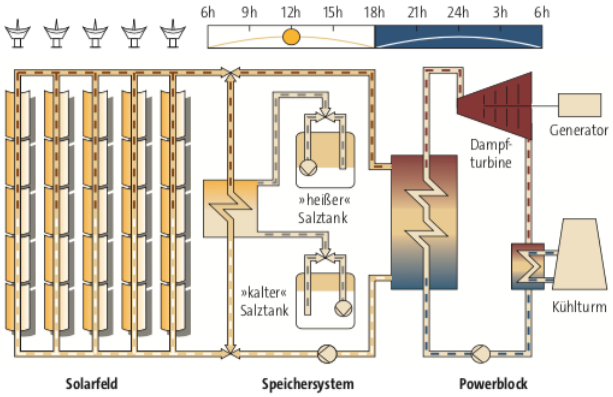

Figure 2.2: Structure of the Andasol III CSP plant [And08]

Within the solar field, collectors concentrate the direct normal irradiation of the sun onto an absorber tube. A heat transfer fluid is led through the absorbers, that reaches a temperature of around 400 degrees Celsius and is then routed into the attached power block. Within the power block the hot fluid is fed into the heat exchanger, where water is heated to power a conventional steam turbine which drives a generator that produces the electricity. One advantage of this technology compared to other forms of renewable energy production such as Wind or PV, is the possibility of storing the heat generated by the solar field in a thermal storage. This allows the operation of the power block at any time electricity is demanded.

The main focus will be placed on the solar field, further referred to as SF, as it is the part primarily influenced by soiling. The structure and the functions of the solar field will therefore be described in detail in the following section.

2.1.1 Solar Field

Depending on the scale of the CSP plant, the solar field has a combined mirror surface, also called aperture area of around 500.000 square meters or more. The solar field is built up in rows of parabolic through collectors, which are oriented on a North-South axis to enable the adjustment of the mirror orientation based on the position of the sun. Within each row the PTC collector units are installed successively and each unit can thereby individually measure and follow the position of the sun. Within a collector unit the mirrors are assembled to form a parabolic shaped surface as can be seen in Figure 2.3. The sun rays that hit this surface are reflected onto the receiver centered at the focal line due to

Figure 2.3: Parabolic Trough Collector at the Plataforma Solar de Almería (PSA). Owner and operator of PSA is the Research Center CIEMAT. [Source: DLR/Markus-Steuer.de.]

its parabolic shape. The receiver consists of an absorber tube, through which a thermal heat fluid (HTF) is led. The absorber tube has special characteristics and insulation to enable the radiation to pass its surface and keep the collected heat inside.

To enable steady heating of the thermal heat fluid, a piping system is used as shown in Figure 2.4. The heat fluid circulates through the SF in 'cold' pipes illustrated in blue and 'hot' pipes illustrated in red. After it is cooled down within the power block, the fluid enters the solar field and is distributed through the cold pipes. It is then separately fed into the so called loops, each consisting of two collector rows and a header pipe at the end. While passing through the loop, the fluid is heated and the separate hot streams are then merged together within the hot pipes. These are eventually routed to the thermal storage or directly to the power block.

The power block operates most efficiently at a fluid inlet temperature of around 400 degrees Celsius depending on its specification. This temperature has to be provided by the solar field. When there is less heat produced, for example due to a lower Direct Normal Irradiation (DNI) caused by cloudy weather, the heat fluid has to circulate with a reduced flow rate to reach the required temperature. Thus the reduced mass flow within the plant and accordingly within the power block, results in a lower electricity production. It follows that the amount of electricity produced is mainly regulated by the adjustment of the mass flow.

This is a simplification of the processes within a CSP plant, as there are various optical

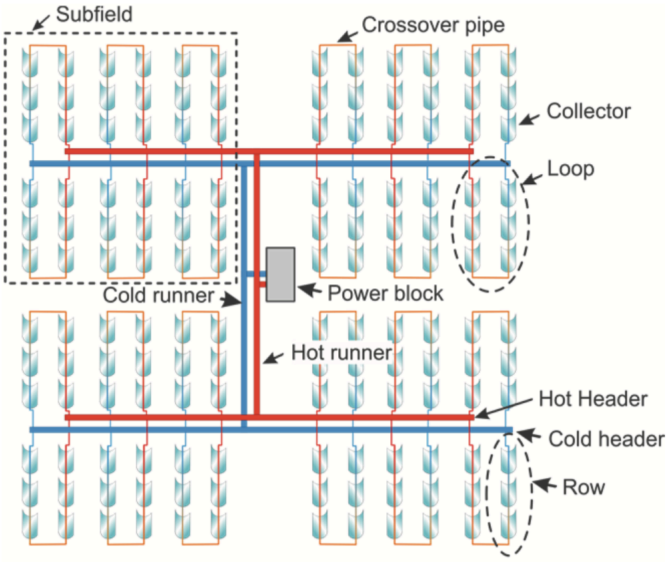

Figure 2.4: Structure of the solar field [HDF+17]

or thermal phenomena that influence the plants efficiency. In this thesis the attention is restricted to the reflectivity loss due to soiling, whereas other effects are not further examined. Still they are modeled in the previously existing simulation tool, which is used to calculate the plants electricity production. For further information see SolarPaces Guideline [HDF+17] .

2.1.2 Soiling

Dust and sand particles that deposit on the mirror surface lead to scattering and absorption of the sunlight and thus reduce the reflectance of the mirrors. As fewer sun rays hit the receiver, less heat can be produced inside the absorber tube. The profound analysis of how soiling affects the reflectance of the mirror surface is not within the scope of this thesis. The same applies for the process of soiling on the mirror surfaces, as its complex physical phenomenon is influenced by multiple factors. Meteorological data such as relative humidity, precipitation, wind, temperature and the natural and artificial dust concentration all have an impact on the amount of soiling [BF81]. Furthermore the position of the mirrors within the SF as well as the location of the power plant itself determine the extent to which the mirrors will soil.

2.1.3 Cleaning

To restore the reflectivity of the collector units, they have to be cleared of all particles that have deposited on the mirror surfaces. In general this is done by cleaning trucks with

an attached cleaning system, using pressurized water techniques or contact scrubbing. In [KVH12] both techniques are compared by their effectiveness with regard to different deposition types. Both types also differ in the amount of cleaning water used. [Kai11] has evaluated different truck systems and shown that, depending on the technique between 0.3 and 1 l of water is consumed per square meter. Other parameters that have been evaluated are the extent to which the reflectance is restored, the fuel consumption of the trucks and the cleaning speed.

2.2 CSP - Modeling for the optimization algorithm

The Optimizer aims to reduce the water consumption while maximizing the energy output at the same time. Therefore, the need for cleaning in every time step is analyzed. As each cleaning decision may lead to different scenarios, these need to be valuated first, in terms of water consumption and energy produced. The effect of soiling on the plant's performance is calculated using a simulation model that represents the plant by its technical and physical processes, described in the following section.

Each of the CSP subsystems, introduced in the previous section, need to be modeled separately [HDF+17] . For the optimizer an existing simulation tool, developed in the research group of the Institute for Solar Research within the German Aerospace Center, has been extended by a more detailed model of the solar field. This will be described below, whereas the power block and the thermal storage are subsystems that are not described in this thesis and therefore are further considered here as black boxes. To evaluate the optimizer it is key to understand the interaction between the different subsystems. They can be formulated as balance equations, for which the following interface variables are used, also depicted in Figure 2.5.

- mass flow $\dot{m}_{SF}^{t}$
- inlet temperature $T_{in,SF}$
- outlet temperature $T_{out,SF}$
- inlet and outlet pressure $p_{in,SF}$ and $p_{out,SF}$

The *mass flow* through the SF describes the amount of thermal heat fluid per time unit that circulates trough the pipes in the SF. It can be regulated by the main pump located at the power block outlet and SF inlet as well as by smaller valves at the loop inlets. Within the simulation the mass flow rate is determined in the SF model and then inserted as an input variable for the power block or thermal storage.

The *SF inlet and outlet temperatures* are the temperatures measured at the solar field exit and inlet. In the SF simulation model they are considered as given parameters, as they are both determined by the power block. As described above, the power block operates most efficiently at a certain temperature and therefore the aim is to extract exactly this temperature from the SF. The SF inlet temperature is determined by the cooling sys-

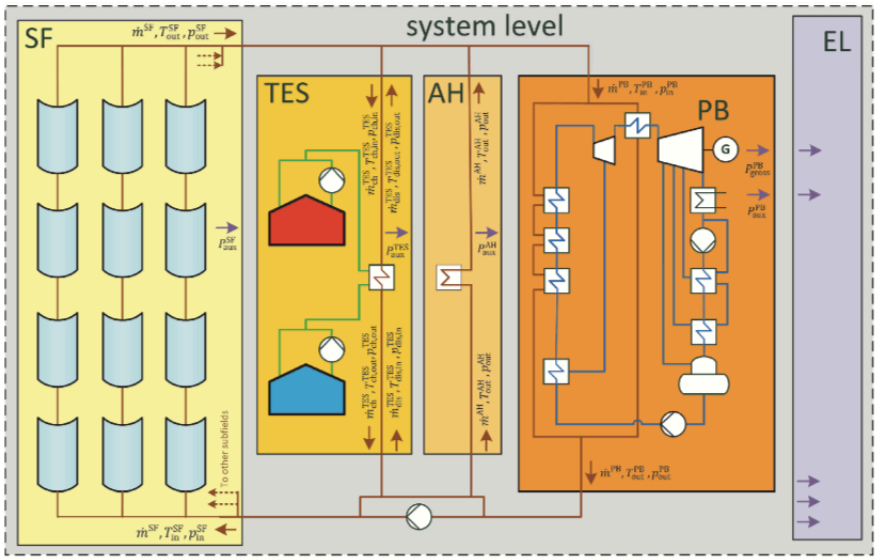

Figure 2.5: CSP subsystems and interface variables - [HDF+17]

tem within the power block.

As described in the previous section, the mass flow m of each loop i is the adjustable factor in the solar field model. It can be calculated using the following set of equations that are dependent on the direct normal irradiation and the actual cleanliness of the solar field:

$$0 = \dot{Q}^t_{abs,i} - \dot{Q}^t_{loss,i} - \dot{m}^t_i \cdot cp \cdot (T^t_{out,i} - T^t_{in,i}) \quad \text{for all loops i} \in \{1, ..., n\}, \tag{2.1}$$

$$0 = \sum_{i=1}^{n} \dot{m}^t_i \cdot cp \cdot (T^t_{out,i} - T^t_{in,i}) - \dot{m}^t_{SF} \cdot cp \cdot (T^t_{out,SF} - T^t_{in,SF}) \quad \text{for SF field,} \tag{2.2}$$

for all timesteps t :

$\dot{Q}^t_{abs}$	Absorbed power of loop i
$\dot{Q}^t_{loss}$	Thermal power loss of loop i
$\dot{m}^t_i$	Mass flow of loop i
$T^t_{in,i}$	Loop inlet temperature
$T^t_{out,i}$	Loop outlet temperature
$T^t_{in,SF}$	Solar field inlet temperature
$T^t_{out,SF}$	Solar field outlet temperature
cp	Thermal heat capacity

For each loop the balance equation 2.1 is set up at any time step, as well as the balance

equation for the whole field given in 2.2. By solving this system of equations at any time step we derive the mass flow m. Depending on the operational state at that time step, the mass flow is then passed on to the power block or thermal storage 'black box' as an input factor. By repeating this, the energy yield can be calculated for the whole day. Thereby one aims to influence the mass flow in such a way, that the energy yield is maximized. [1]

Within equation 2.3 the absorbed thermal power $\dot{Q}_{abs}$ stated in kW is the fraction of the available thermal power provided by the sun that can effectively be captured. Effects that reduce the heat production are modeled within equation 2.3 as efficiency values η.

$$\dot{Q}_{abs} = G_{bn} \cdot A_{nom} \cdot \dot{Q}_{avail} \cdot \eta_{opt} \cdot \eta_{avail} \cdot \eta_{clean} \cdot f_{foc,A} \tag{2.3}$$

where :

η_{clean}	Cleanliness efficiency
$\dot{Q}_{avail}$	Available thermal power by the sun
η_{opt}	Optical efficiency
$f_{foc,A}$	Focus factor
A_{nom}	Nominal aperture area

Within equation 2.1 the term $\dot{Q}_{loss}$ describes the heat losses, that occur within the pipes of the loops, as well as in the header pipes . They are functions of the mean temperature in the respective pipe section.

All variables have to be within their limits. For example on days with very high irradiation, too much heat is produced and the mass flow rate reaches its maximum value. If the energy cannot be used by the power block or stored in the thermal storage the overheating of the system has to be prevented. In that case collector units have to be defocused to enable a safe operation of the power block and therefore available energy is not used, which is known as dumping. In the simulation the focus factor f_{foc} is set to a value below one.

When calculating the absorbed power in a collector unit, there are several efficiency values that have to be considered. They can be determined by evaluating the processes within the collector unit. Taking a closer look, one can see that there are several sources of errors that can affect the efficiency of the collector, such as the position of the absorber tube, a wrong angle between the collector position and the incoming radiation or particles on the mirror surface summarized as the optical efficiency. The last effect is due to soiling and will be addressed in the following section.

[1] The interface variable for the pressure is not considered here. Furthermore the effects of cool-down and heat-up of the CSP plant in the mornings or evenings have to be considered, but are not discussed in this thesis.

2.2.1 Reduced reflectance due to soiling

In the available literature on cleaning optimization of CSP plants, the effect of soiling has widely been modeled as a reduction of the DNI value. To describe the effect soiling has on the mirror performance, the term cleanliness is defined. It describes the ratio of the actual reflectance of the mirror to the reflectance in a perfectly clean state:

$$\eta_{clean}(t) = \frac{\rho(t)}{\rho_{clean}}$$

Within the simulation the cleanliness value is modeled as an efficiency value, the *cleanliness efficiency* $\eta_{clean} \in [0,1]$. In Equation 2.3 the direct influence of the cleanliness on the energy that is absorbed by the receiver can be seen.

Furthermore, instead of one average value for the whole field, the cleanliness will be determined for different parts of the solar field $\eta_{clean,i}(t)$. This approach is chosen due to the spatial dependency of soiling. Also the influence of a higher resolution of soiling on the plants performance has been evaluated in [RAF+19]. It was shown that, using locally resolved cleanliness values, a lower solar field mixed outlet temperature and mass flow rate was computed while all other parameters remained the same, which was indicated by the author to be a more realistic calculation.

Within the model presented in this thesis, the cleanliness value is assumed to be constant for a whole day. [HSSN20] evaluated the impact of the incidence angle on the effect of soiling on the plants performance. A noticeable decrease of specular reflectance was found for an increase of the angle of incidence. This effect can be included in the cleaning optimizer by adjusting the cleanliness efficiency value accordingly throughout the day, though this is not considered within the scope of this thesis.

The Cleanliness efficiency value η_{clean} can be determined in different ways. The data on soiling used in this thesis has been collected with a TraCS which is a reflectometer that is installed within the plant location and measures the reflectance of a sample mirror which is comparable to the mirror surface of the solar field [WWD+18].

As part of the SOLWARIS project Soiling sensors are developed that can be installed throughout the field to deliver automatic and space-resolved soiling data from the SF [CdGV]. As these are installed directly within the mirror surface, an accurate reflectance value can be measured. Hence the space-resolved reflectance data enables the optimizer to select parts of the SF for cleaning showing lowest cleanliness value.

Furthermore, the cleanliness values can be forecasted for a short time horizon. For this we introduce the term Soiling Rate (SR), which describes the rate of change of the cleanliness value:

$$SR(t) = \frac{\Delta\eta_{clean,i}(t)}{\Delta t} \tag{2.4}$$

$$\eta_{clean}(t+\Delta t) = \eta_{clean}(t) + SR \cdot \Delta t$$

During some rain events, with a precipitation that is strong enough to clean the mirror surface, we speak of a natural cleaning event. Therefore, the effect of rain can modeled as the natural cleaning rate NCR and applied correspondingly.

The logic of the optimization uses a time horizon of ten days, which is the time required to clean the entire solar field. This horizon has been chosen to enable the optimizer to account for significant events in the near future that need planning ahead, such as rainfall with a cleaning effect. Other effects that could occur not related to soiling are high or low DNI values or other effects that impact the plants performance, making cleaning either more favorable or even unnecessary.

To derive possible cleanliness values for a time horizon of ten days, a forecast for the SR has to be provided to the optimizer. In general, the creation of a forecast of the SR is very difficult due to the complex relation of meteorological phenomena and soiling. In [WWT+19] a physical model has been developed to better understand the process of soiling and derive a model that determines the soiling rate based on weather parameters. Until now, a valid forecast is only available for the upcoming three days. Beyond that, a forecast cannot be derived from the same detailed physical model. This is due to the meteorological input data needed, that cannot be accurately determined for a period longer than three days. In [TBCP+20] the problem has been addressed by using a physical model with stochastically generated weather data as input data. Therefore, a detailed physical model can be used to derive a stochastical SR.

Within this thesis it is assumed that the optimizer is provided with a valid forecast for the first three days based on the above mentioned physical model. The forecasts for day four to ten will be developed based on the analysis of a soiling dataset consisting of soiling measurements for six years provided by [WWD+17] .

2.2.2 Modeling the cleaning action

The cleaning action is modeled as the option provided to the optimizer, to reset certain cleanliness efficiency values to 100%. For this purpose, the cleaning speed is of a high interest, as it determines the subdivision of the solar field into sets of loops. The number of loops and therefore the size of the subdivision equals the number of loops that can be cleaned within one cleaning action. The cleaning speed depends on the layout of the field and the accessibility of the loops, which may be obstructed by the piping system or uneven surface areas. Cleaning the whole solar field takes between seven to fifteen days, for some techniques even longer. For the optimizer we assume that one cleaning truck is able to clean the entire field in ten days and hence 10% of the field within one shift. In reality, the solar field is commonly cleaned during nighttime, as it is not in operation and no mirrors have to be defocused. Therefore, the assumption was made that cleaning actions will happen during the night only.

Each cleaning action consumes a certain amount of water and causes variable cleaning costs. For that reason the decision to clean must be linked to a certain penalty which is described within the model as the *alpha value* α. The α-value has to be defined as a parameter within the optimization algorithm and can be adjusted according to the importance of saving water. In regions with a shortage of water this value can be set very high to avoid cleaning as much as possible.

3 Markov decision process

While formulating the problem as a mathematical model, we encountered the following specifications of the problem, presenting certain difficulties for the optimization: First of all, the impact of soiling on the energy production can not be described by a simple function, as it is a complex system of processes, that can only be computed using a simulation model. Secondly, cleaning the entire solar field and thus restoring the clean condition of all mirrors takes at least a week or longer. Therefore, the optimizer needs to consider different time steps simultaneously. Thirdly, the decisions of the optimizer influence each other and therefore the algorithm needs to optimize a sequence of decisions instead of a single one. Another aspect to be considered is the fact that we do not know from the beginning what will happen throughout the year. For this reason the concept of an online algorithm was chosen, which receives new information in each time step and hereupon adapts its decision process based on the new input.

Both in [TBCW+17] and [TBCP+20] the problem of optimizing the cleaning decision is formulated as a Markov decision process. A Markov decision process describes the evolution of a system over a discrete time horizon. The system is influenced by a decision maker that can choose from different actions in each time step and thus initiate a change of the system. The word state can be interpreted as the state of the system, summarizing certain properties. The change of the system can then be described by the transition from different states to one another, and therefore changing the respective properties of the system. An action taken in a specific state does not necessarily lead to the next state that is predefined for the pairing, but does also depend on stochastic transition probabilities.

The Markov decision process fits to two essential requirements of the cleaning optimization problem very well:

1. The aim to find an optimal value of water consumption and energy output over a long term. An algorithm that focuses on a short term only, neglecting upcoming decisions, might not be able to find this optimal value.
2. Furthermore the stochastic nature of the soiling process and the meteorological data such as irradiation can be considered .

In order to apply a Markov decision process, further denoted as MDP, to the cleaning optimization problem, the mathematical fundamentals as well as a definition of the MDP will be given in the following chapter. Furthermore the basic algorithms suitable to solve the cleaning optimization problem will be discussed.

3.1 Fundamentals of Markov decision processes

Markov decision processes were first introduced in the field of statistical sequential analysis and then became a topic of research within the area of dynamic programming. Besides others, the work of Bellman [Bel66] and Howard [How60] contributed to the development of MDP research [HRS16]. Today Markov decision processes can be found within the fields of Mathematics and Computer science. It is used to describe Reinforced Learning Algorithms within the field of Machine Learning [SB]. The theory of a Markov decision process is based on the concept of a Markov chain. As Markov chains are defined within the field of Stochastics, some basic definitions will be given first, based on [Gal13]. Throughout this chapter, basic stochastical principles are assumed to be known to the reader.

A probability model describes the mathematical representation of a random phenomenon. Given the sample space Ω of a probability model, all possible subsets of Ω are denoted as *events* $A \subset \Omega$ with a corresponding probability $\Pr(A) \geq 0$.

Within the theory of Markov decision processes, one often encounters the conditional probabilities of an event A given the event B, which is defined by:

$$\Pr(A|B) = \frac{\Pr(A \cap B)}{\Pr(B)} \tag{3.1}$$

A random variable is a function that maps every outcome from the sample space $\omega \in \Omega$ to an unique real number $X : \Omega \to \mathbb{R}$, $\omega \to x$. Within this thesis only discrete random variables will be covered, each of which is associated with a probability distribution, that maps a probability to each value of the random variable. The expected value $\mathbb{E}[X]$ of a discrete random variable is given by:

$$\mathbb{E}[X] = \sum_{x} x \cdot \Pr(x) \tag{3.2}$$

Using the concept of a random variable one can describe the stochastic development of a system over time. A **stochastic process** $(X_t)_{t\in\mathbb{N}} = (X_0, X_1, ...)$ is a finite or infinite sequence of random variables defined on a common probability model. The index t can be regarded as time [Gal13], such that in each time step the random variable indicates the state of the system. The set of states is further denoted as the *state space S*, whereas the random variable represents a state by its index. The definition of a state includes certain properties of the system. The transitions from each state to one another are defined by transition probabilities, such that only states can be reached, for which the transition probability is greater than zero. This development is called a stochastic process.

Stochastic processes can be distinguished by their time values and sample values, both being either discrete or continuous. An example of a stochastic process that is defined on a continuous time space is a Poisson process. Within this thesis only discrete time stochastic processes will be discussed, as they fit best to the time discretization used here due to fixed nighttime cleaning cycles of the problem at hand.

3.1.1 Markov Chains

A Markov chain has a special characteristic, further denoted as the Markov property, which implies that the transition from a state $s_{i_{t-1}}$ to state s_{i_t} depends on the current state only and further information about the past do not influence this probability. A Markov chain is defined as follows [Gal13]:

Definition 3.1.1. A **Markov Chain** is a stochastic process with countable state space S that meets the following condition: The state at time t, denoted as the sample value of the random variable X_t at time t is dependent on the most recent random variable X_{t-1} only, which can be described by the conditional probability:

$$\Pr(X_t = i_t | X_{t-1} = i_{t-1}, X_{t-2} = i_{t-2}, ..., X_0 = i_0) = \Pr(X_t = j | X_{t-1} = i) \tag{3.3}$$

for all conditioning events $X_{t-1} = i_{t-1}, ..., X_0 = i_0$ with positive probability. The probability can then be defined by $\Pr(X_t = i_t | X_{t-1} = i_{t-1}) = P_{i_{t-1} i_t}$ further referred to as the **transition probability** from state i_{t-1} to state i_t:

In case that the state space S is finite we speak of a finite-state Markov chain, which is the case for the model described in this thesis. Furthermore, we regard Markov chains only, for which the transition probability does not dependent on the time, known as Homogeneous Markov Chains.

Example *Imagine a passenger taking the tram to pass through a city. At each station the tram either stays at its current stop due to technical problems, or drives to the next station based on certain probabilities. The probability of arriving at the next station is only based on the current station and thus does not depend on all previously visited stations.*

For a graphical representations of Markov Chains a directed graph can be used, in which each node represents one state and a directed arc is drawn for each non-zero transition probability. Furthermore a Markov Chain can be described by the $n \times n$-matrix P with elements $P_{i_{t-1} i_t}$, whereas n is the size of the state space S.

$$P = \begin{bmatrix} P_{11} & P_{12} & \cdots & P_{1n} \\ P_{21} & P_{22} & \cdots & P_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ P_{n1} & P_{n2} & \cdots & P_{nn} \end{bmatrix} \tag{3.4}$$

Thus, if the initial probability distribution for X_0 and the transition probabilities are given by P, the probability distribution for all stages can be calculated by iteratively multiplying P with the probability distribution vector of the previous stage.

Markov Chains with rewards

Before introducing Markov decision processes, a Markov chain with rewards can be looked at. Assuming that within each state that is entered a reward can be received, one can calculate the expected reward over all time steps. To do so the transition probability distri-

butions of all time steps are multiplied with the rewards, that are earned when entering a state. This concept is simpler and shows the idea of increasing the value over a series of states.

Example *The tram problem can be extended by the assumption that at each station the passenger loses a certain number of minutes of his time, representing the, in this case negative reward for being in a state. The overall expected time consumption over a certain number of stages can be calculated, based on the probabilities for all states of moving forward or staying at the station*

3.2 Definition of Markov decision processes

By adding a decision maker to this theory of a Markov chain with rewards one receives a Markov decision process. The decision maker can choose among various options of rewards r_s and the corresponding sets of transition probabilities. Within [HRS16] the formal definition of a Markov or Markovian decision process is given:

Definition 3.2.1. A Markov decision process with finite horizon and finite state space is defined as a tuple $(S, A, p, r_S, V_0, \beta)$ whereas

- S is the state space.
- A is the action space.
- p is the transition probability $p : S \times A \times S \to [0,1]$, the probability for reaching state s', when starting in state s and choosing action a, such that all transition probabilities from one state s sum up to one: $\sum_{s' \in S} p(s,a,s') = 1$.
- r_s is the one-stage reward function $r_s : S \times A \times S \to \mathbb{R}$, the reward received when being in state s and taking action a, whereas r_s can also be dependent on the successor state s' or not.
- V_0 terminal reward function $V_0 : S \to \mathbb{R}$, that is obtained when the system reaches an end state, i.e. a certain state for which the process terminates.
- $\beta \in \mathbb{R}^+$ is the discount factor.

Example *Again the example can be regarded: at each station the passenger can choose from either taking the tram or walking to the next station. At this time the negative rewards are associated with the state-action tuple such that taking the tram to the next station takes 1 minute and walking takes 5 minutes. With a certain probability the tram fails and the passenger remains at the current station, whereas he reaches the next station certainly by walking. The question arises at which station the passenger should walk, and at which stations he should take the tram in order to arrive at a certain destination with a minimum time consumption.*

A policy is defined as a function $\pi : S \to A$, used by the decision maker to select an alternative in each time stage depending on the current state. The sequence of states

that are entered due to a certain policy π, when starting in s_0 can be viewed as a Markov chain on some probability space $(\Omega, \mathbb{P}_{\pi,s_0})$, with transition probabilities for each state-action tuple $(s_{i_t}, \pi(s_{i_t}))$ to reach state $s_{i_{t+1}}$. Hence the MDP satisfies the Markov Chain property defined in Definition 3.1.1: Starting in an initial state s_0 the system evolves over the time horizon, whereas for each state the policy decides on an action to be taken. Thus each transition from a state s_{i_t} to its successor $s_{i_{t+1}}$ depends on the most recent state only and is therefore independent of all before visited states $s_0, s_{i_1}, ..., s_{i_{t-1}}$.

For a simplification the states in each time step t are denoted as s for the starting state and s' for the successor state in the remaining chapter. Also we regard $n \in \{N, ..., 0\} \subset \mathbb{N}$ as the number of stages the process will pass until an end state is reached.

The **expected N-stage reward** for the intitial state s_0 and a policy π is the expected aggregated reward over the entire time horizon $N \in \mathbb{N}$ [HRS16]. The aim is to find the maximum expected N-stage reward over all policies:

$$V_N(s_0) := \sup_{\pi \in \text{set of policies}} V_{N,\pi}(s_0) \tag{3.5}$$

The question arises how to determine a policy that maximizes V_N. To do so, $Q(s,a)$ is introduced, calculating the value and thus the benefit of taking a specific action when being in state s for each stage n:

$$Q_n(s,a) := \sum_{s' \in S} p(s,a,s') \cdot r_s(s,a,s') + \beta \cdot \mathbb{E}[V_{n-1}(s')], \quad n \geq 1, a \in A \tag{3.6}$$

The first part of $Q_n(s,a)$ is calculated by evaluating the immediate reward of all possible successor states. The second part of the sum calculates the expected aggregate rewards when finishing the Markov process from the successor state on and the remaining time steps are reduced by one stage: V_{n-1}

A decision policy π_n is called a *maximizer for stage n* if it selects the optimal action at stage n: $a = \arg\max_{a \in A} Q_n(s,a)$

Based on this, the following principles can be stated, based on the theory of Bellman [HRS16] :

1. **Optimality Criterion (OC)** :
   If π_n is a maximizer at stage n for $1 \leq n \leq N$, the N-stage policy $(\pi_1, ..., \pi_N)$ is optimal for $N \in \mathbb{N}$
2. **Value iteration (VI)**

$$V_n(s) = \max_{a \in A} \left[\sum_{s' \in S} p(s,a,s') \cdot \left[r_s(s,a,s') + \beta \cdot V_{n-1}(s') \right] \right] \tag{3.7}$$

Based on the Optimality Criterion as well as the Value Iteration the algorithm stated in the following section can be formulated .

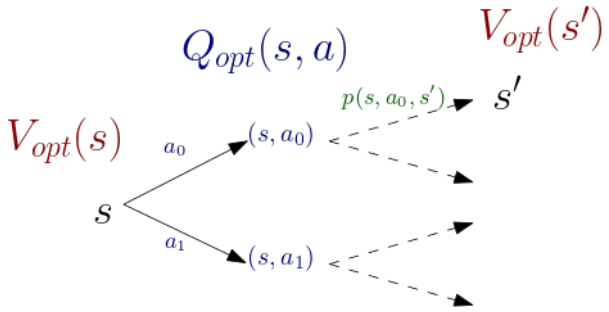

Figure 3.1: Optimality criterion applied

3.3 Algorithm

To solve a MDP two algorithms are commonly proposed, that are Policy Evaluation and Value Iteration, whereas the latter will be described below. Policy evaluation simply calculates the values of all states for a given policy. This is done repeatedly until all states converge to a stationary value [LS].

Within Value Iteration the above stated optimality criterion is used as follows [LS] :

$$Q_{\text{opt}}(s,a) = \sum_{s'} p(s,a,s') \cdot [r_s(s,a) + \beta V_{\text{opt}}(s')] \tag{3.8}$$

and

$$V_{\text{opt}}(s) = \begin{cases} 0, & , s \text{ is end state} \\ \max_{a \in A} Q_{\text{opt}}(s,a), & \text{else} \end{cases} . \tag{3.9}$$

The value $V(s)$ of a state describes the value of being in that state. If a state has a higher value than others, this implies that it is more favorable to be in that state. Here all end states have value zero, as theoretically all upcoming states provide zero reward. Furthermore the variable $Q(s,a)$ is used, which represents the expected value obtained by taking a specific action. As a specific action can lead to multiple new states s', the calculation of Q(s,a) requires each immediate reward $r(s,a,s')$ and V(s') to be weighted by p(s,a,s'), the probability to enter that specific state s'.

Recursively choosing the action that returns Q_{opt} and thus receiving the optimal policy due to the Optimality Criterion, implies the following dynamic programming algorithm.

Algorithm 1 Value Iteration

Input: Markov decision process $(S, A, p, r_S, V_0, \beta)$
Return: Decision policy $\pi(s) \in A$ for all states s

1: **for all** states **do**
2: $V_{opt}^0 \leftarrow 0$
3: **for** iteration i = 1, ..., I_{VI} **do**
4: **for all** states **do**
5: $V_{opt}^i(s) = \max_{a \in A} \left[\sum_{s' \in S} p(s,a,s') \cdot (r_s(s,a,s') + \beta \cdot V_{opt}^{i-1}(s')) \right]$

This returns the computation of the optimal policy as well:

$$\pi_{\text{opt}}(s) = \arg\max_{a \in A} Q_{\text{opt}}(s,a) \tag{3.10}$$

Within each iteration step $V_{opt}(s)$ is updated by applying the optimality criterion. The respective action is chosen that maximizes $Q_{opt}(s,a)$. The algorithm converges toward the optimal value after a certain number of iterations. The number of iterations is defined by the stopping criterion for some $\epsilon > 0$, such that $|V^{i+1} - V^i| < \epsilon$. A sufficient condition under which V converges, is either that β is strictly less than one or the MDP graph is acyclic [LS] . Due to the specific structure of the state space, this algorithm can be used and furthermore simplified to solve the MDP model developed in this thesis.

3.3.1 Algorithm for a MDP with characteristic state space

The state space of the MDP modeled in this thesis can be represented as a directed graph. Each state is depicted as a node and the transition from a state to each of its possible successors as a directed arc. Thus for each combination of an action and a corresponding transition probability greater than zero, an arc is drawn. As the state space is defined such that no state can be visited more than once, the directed graph does not consist of any cycles and can therefore be described as a directed tree. We further regard the depth of this tree as the *period* of a state and in each time step one state from each period can be reached. The process starts with the initial state, the root of the tree and all states in the final period are end states, represented as the leaves of this tree.

The optimality criterion can be applied as follows: After setting the value $V(s_{end})$ to zero for all end states, its is now possible to calculate $Q(s,a)$ for all states in the penultimate period. To determine the value of a specific state s the maximum over all associated $Q(s,a)$ is chosen. Likewise the policy $\pi(s)$ for each state is the $\arg\max_{a \in A} Q(s,a)$,being the action that corresponds to the highest $Q(s,a)$ value. This process is repeated for all periods, until we reach the initial state in period 0. The action that should be chosen in the initial state is then saved in $\pi(s_0)$. Thus the algorithm does only need one iteration to compute the optimal values.

The value of each state can be computed by going backwards from the last period to period 0 and iteratively updating V for each state in the following manner:

Algorithm 2

Input: Markov decision process $(S, A, p, r_S, V_0, \beta)$ with the initial state s_0 and transition probability matrix P

Return: Decision policy $\pi(s) \in A$ for all states s

for all end states s_{end} **do**
    $V(s_{end}) = 0$
for period $k = K$ to 0 **do**
    for all states s with period k **do**
        for all actions $a \in A$ **do**
            compute Q for each state: $Q(s,a) = \sum_{s'} p(s,a,s') \cdot [r_s(s,a,s') + V(s')]$
        compute V : $V(s) = \max_{a \in A} Q(s,a)$
        compute π : $\pi(s) = \text{argmax}_{a \in A} Q(s,a)$
return $\pi(s_0)$

4 Optimization

In this chapter the mathematical model to the problem presented in Chapter 2 will be introduced. Therefore, the basic idea and application of the optimizer is briefly stated. Secondly, a description is given on how the MDP is applied to the problem and the variables used within the model are introduced.

As described in Chapter 2, the efficiency of the CSP plant is reduced through the soiling of mirror surfaces and therefore cleaning is necessary to restore the reflectance of the mirrors. At the same time each cleaning action consumes a certain amount of water. In some locations water is only available in limited quantities or very expensive, and therefore reducing the water consumption is critical. If we want to find a solution to this tradeoff in the longterm, the aim of the optimizer is to make a sequence of decisions whether to clean the solar field or not.

The sequence of decisions is described as follows: At each day a cleaning decision is made by running the algorithm. As input the algorithm obtains the current measured cleanliness values of all sectors, as well as the soiling rate and meteorological forecast for the upcoming ten days. By using these inputs, a cleaning decision for the current time step is computed.

4.1 MDP applied to the problem

The algorithm applied at every decision, is based on a Markov decision process, which can be represented by the tuple (S, A, p, r_s, V_0). For the optimizer we define this tuple as following:

- In the finite state space S each state s is represented by the cleaning configuration of the solar field and by the optimization period: $((\eta_1, ..., \eta_{10}), k)$
- The action space A consists of three different actions 'no cleaning', 'cleaning one set of loops' or 'cleaning two sets of loops'
- p is the transition probability $p : S \times A \times S \rightarrow [0, 1]$, such that all transition probabilities starting from one state s sum up to one: $\sum_{s' \in S} p(s, a, s') = 1$
  Starting with a certain cleaning configuration and choosing a specific cleaning action we receive a new cleaning configuration in the next period based on the SR and NCR probabilities.
- V_0 terminal reward function $V_0 : S \rightarrow \mathbb{R}$, that is obtained when the system reaches

an end state, which is set to 0 for the model.

- r_s is the one-stage reward function $r_s : S \times A \times S \to \mathbb{R}$, that represents the energy produced in one period based on the cleaning configuration minus the *alpha value* α representing the hypothetical cost value of a cleaning action.

The β value introduced in Chapter 3 is set to 1 for the model, as no discount factor is applied.

In Figure 4.1 the model is depicted as a directed graph. Starting from an initial state, the optimizer analyzes the three cleaning options within the action space $A = \{a_0, a_1, a_2\}$ depicted as blue arrows. The basic idea is that, based on the current state and the chosen cleaning action, the system can reach several new states due to the various SR or NCR that occur with different probabilities depicted as green arrows. The probability $p(s_0, a_0, s_1)$ corresponds to the probability that certain SRs and NCRs are given for that time period. Both are then applied, in order to receive the new cleanliness values for period 1.

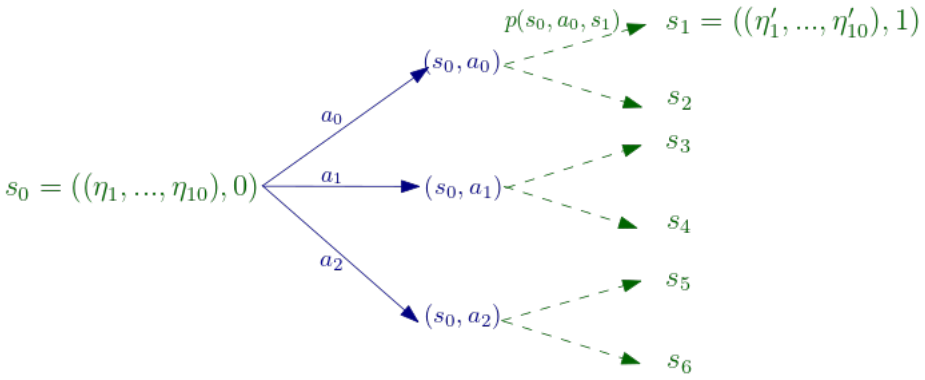

Figure 4.1: Representation MDP

Time horizon

When the algorithm is called, it receives a time step as an input that defines the starting point of the optimization. The algorithm itself has a time horizon of ten days modeled in the following as discrete-time values further titled as k periods, whereas $k \in \{1, 10\}$.

State space: cleanliness configuration and period

The solar field is divided into n sets of loops, further denoted as sections, which are determined using the geometry of the solar field and practical properties of the cleaning event. The number of sets equals the amount of days required to clean the entire solar field using one cleaning unit. Therefore, it is assumed that all sets consist of an equal number of loops and that they are cleaned within one time unit, being one cleaning shift. Combining loops to a section will influence the output of the solar field compared to a model in which a cleanliness value is used for each loop respectively. This will not significantly influence the result of the optimizer due to the fact, that in reality loops are

not cleaned randomly throughout the field, but in a consecutive order. Therefore, after one cleaning event, all loops within one section have the same cleanliness.

The cleanliness values do have a certain resolution further denoted as *cleanRes*. We assume that low cleanliness values will only appear rarely as they cause high losses in electricity generation and therefore are not chosen by the optimizer, as the main purpose of a power plant is to produce electricity. In the following the cleanliness values for all sectors in one period will be entitled as the *cleanliness configuration* of the SF. It is important to note, that the same cleanliness configuration of the SF in two different periods will lead to a different electricity production. This is due to the difference in DNI values and other time-specific parameters that affect the plant's performance. Therefore, they have to be handled as two different states and we define each *state s* by its cleanliness configuration as well as its period $s = ((\eta_1, ..., \eta_{10}), k)$.

Action space

The action space consists of three actions $A = \{a_0, a_1, a_2\}$ that can be chosen from:

- a_0 : no cleaning of the field
- a_1 : one set of loops will be cleaned
- a_2 : two sets of loops will be cleaned

When a_1 is applied, the set that has the lowest cleanliness value is selected for cleaning. Subsequently its cleanliness value is set to 1 in the cleanliness configuration. For a_2 the two sets with the lowest cleanliness values are set to 1 respectively. The available actions are the same for any state.

Transition probability

The transition probability is the probability of starting with a certain cleanliness configuration $(\eta_1, ..., \eta_{10})$ in period k while choosing action a (tuple $(s, a) \in S \times A$) to get the cleanliness configuration $(\eta'_1, ..., \eta'_{10})$ in period $k+1$. This new cleanliness configuration is mainly influenced by the soiling rate and natural cleaning rate that is forecasted for period $k+1$.

One-stage reward function - optimizer calls simulation model

The one stage reward function $r_s : (S, A, S) \rightarrow \mathbb{R}$ maps an one-stage reward to each new stage that can be reached in period k. The reward includes the energy yield minus the water consumption in period k. Thereby the reward function is dependent on the new state s' as well as the action a that has led to this state, as it is defined as follows:

$$r_s(s, a, s') = \text{Energy output corresponding to s'} - \alpha \cdot \text{number of cleaning actions} \quad (4.1)$$

To compute the reward corresponding to a state, the CSP simulation model described in Section 2 is used. The optimizer calls a simulation for one day including the respective cleanliness configuration $(\eta'_1, ..., \eta'_{10})$ and the simulation input for the corresponding time step to calculate the energy output for that specific day. The one-stage reward r_s is then determined by the energy output calculated by the CSP simulation for one day and

the water consumption caused by the chosen cleaning action. Therefore, the α-value is used to determine the cost value of a cleaning action. The α-value is a hypothetical value that represents the costs of water consumption and can be set depending on the costs of water itself or the variable cleaning costs. In case of drought periods the α-value can be set very high to avoid cleaning as much as possible.

4.2 Reducing the size of the state space

In a first step the *state space S* is computed. To do so, the following methodology is used that starts with the initial cleanliness configuration and iterates over all periods. In each period all available cleaning actions are applied causing a new cleanliness configuration in the next stage. For all resulting cleanliness configurations the cleanliness values are reduced by the various SR or NCR values, forecasted for the respective period. As this process would lead to a number of possible states that is exponential in the size of the optimization horizon, a modification is applied after a predefined period:

The first assumption is that two cleanliness configurations can be viewed as equivalent if their sorted vector of cleanliness values is identical.

Secondly, to reduce the space of possible states in each period, the number of sections can be reduced from ten to five by combining two states respectively. Starting from the two sections with the lowest values up to the two sections with the highest values, their cleanliness is set to the average value of both.

In a third step the cleanliness resolution of the average cleanliness values is set to the parameter *cleanRes*. Rounding the cleanliness values does have an impact on the calculated energy output of the CSP plant. The resulting influence of the cleanliness resolution on the performance of the optimizer has been evaluated in section 5.2.3.

4.3 Algorithm applied to the specific problem

The algorithm presented in Chapter 3 can be used to compute an optimal decision for the initial state. In a first step Algorithm 3 is applied to compute the state space and the one-stage reward is determined using the CSP simulation tool. Within the state space each state does only occur once due to the structure of the state space described above. Though the same state can be reached from multiple predecessors.

Algorithm 3

Input: Initial state $s_0 = ((\eta_1, ..., \eta_{10}), 0)$ and forecast for SR, NCR and meteorological Data
Return: State space S and *Energy output*(s) for all $s \in S$

```
for period k = 0 to 9 do
    for all states with period k do
        for all actions a ∈ A do
            Compute possible passage state (s,a)
            for all SR and NCR in forecast set do
                Apply all possible SR and NCR to the passage state (s,a) to receive s'
                if State s' ∉ S then
                    Calculate Electrical output for one day
```

Based on this newly computed state space Algorithm 4 can be applied. The algorithm gets as input the above described MDP model, with the probabilities according to the forecast data for the SR and NCR. Starting from the last period the algorithm iteratively updates all V and Q values. Both values are computed by a recursion and thus the algorithm is solved via dynamic programming. The proof of an optimal value stated in Chapter 3 can be transferred to this application of Algorithm 2.

Algorithm 4

Input: $MDP_{\text{cleaning optimization}}$ with the initial state $s_0 = ((\eta_1, ..., \eta_{10}), 0)$
Return: Cleaning decision $\pi(s_0) \in A$ for state s_0

```
V, Q ← ∅
for all states s with period k = 10 do
    V(s) = 0
for period k = K to 0 do
    for all states s with period k do
        for all actions a ∈ A do
            compute r(s,a,s') = Energy output(s') − α · number of cleaning actions
            compute Q for each state: Q(s,a) = Σ_{s'} p(s,a,s') · [r_s(s,a,s') + V(s')]
        compute V : V(s) = max_{a∈A} Q(s,a)
        compute π : π(s) = argmax_{a∈A} Q(s,a)
return π(s_0)
```

5 Evaluation

In order to improve the output of the optimizer, an evaluation of various parameters has been performed. To do so, the optimizer was tested on a test bench with a time period of either six months or one year. The performance of the optimizer was evaluated with respect to five different aspects:

- Performance loss due to a reduced optimization horizon $h \in \{5, 10\}$
- Sensitivity to the SR forecast dataset
- Parameter study for α-value
- Sensitivity to the cleanliness resolution
- Sensitivity to the accuracy of the SR forecast

The performance gain was measured in terms of reduced water consumption and increased energy output. Secondly, the optimizer is compared to a constant cleaning schedule.

In the following section the test bench as well as all parameters and inputs sets will be described. Thereafter, the results will be presented and discussed.

5.1 General condition of the evaluation

During each test simulation, the optimization algorithm is executed at 12am at every night to decide on the possible cleaning actions based on the current cleanliness configuration of the solar field. All cleanliness values are adjusted accordingly to the cleaning action chosen by the optimizer and the SR for the next day. This newly calculated cleanliness configuration is then returned to the test bench simulation, which calculates the energy yield and all characteristic figures for the overall time period.

The following assumptions have been made that apply for all simulations:

- The same meteorological input data was used.
- At the beginning of the test simulation the initial cleanliness configuration was set to 0.9 for all sectors.
- The SR forecast for the first three days is assumed to occur with a probability of 1, which means that we assume to predict these with a very high certainty. After the third day, the probability of the SR is defined by discrete distributions.

- The cleanliness values have a precise resolution in the first three periods and after the third period the cleanliness values are rounded as described in section 4.2.

Furthermore the algorithm was compared to a constant cleaning schedule which performs one cleaning action each night, cleaning the sections in a fixed order. After all sectors are cleaned once, the process starts with the first sector again. The output of the constant cleaning schedule was calculated with the same input data as described above. For all simulations performed on the different SR forecast datasets, the results have been compared to and identified as a percentage of the results of the constant cleaning schedule.

5.1.1 Input

The meteorological data used for all simulations is applicable for a plant located at the Plataforma Solar de Almeria in Spain. The meteorological forecast data, required by the optimizer to evaluate future states was assumed to be forecasted with one hundred percent accuracy.

In order to evaluate the optimizer based on various years having different soiling occurrences, artificially SR forecast datasets, further denoted as *SR Input i*, were created in a first step. The methodology used to create these datasets was based on the evaluation of the SR for a plant located in Spain [Q11,SolaPacesPoster2017]. A mean SR of 0.0051 with a standard deviation of 0.006 was found, meaning that each day the cleanliness and therefore mirror reflectivity reduces on average by 0.5%. Furthermore it could be shown that the measurements of the SR could be described by an exponential distribution.

Therefore, six SR input datasets were randomly generated. For these the SR was drawn from an exponential distribution. To represent the inter-annual differences in the SR, the mean of this exponential distribution was itself drawn from a normal distribution with a mean of 0.0051 and a standard deviation of 0.002. Within this dataset the SR was consistently positive, which neglects the effects of natural cleaning.

The mean SR for the six datasets are listed in the following table:

SR forecast	mean
SR Input 1	0.0050
SR Input 2	0.0066
SR Input 3	0.0042
SR Input 4	0.0092
SR Input 5	0.0036
SR Input 6	0.0084

For the evaluation the term *forecast accuracy (fa)* is used to describe the conditions of the SR forecast. In case that the measured values of the SR are used as a forecast we denote this as a *deterministic* SR. As these will occur with a probability of 1, the optimizer is assumed to have an accurate forecast of the future SR. Additional simulations were run, to evaluate the effects of an inaccurate soiling rate forecast. Therefore, the optimizer receives two SRs as input for each

period following the third period. One of these SRs is correctly forecasted, whereas the other is assumed to differ by a factor of fd, labelled *forecast deviation (fd)*.

The optimization algorithm used in this thesis simulates the instability of long term weather forecasts by defining different levels of accuracy within the time horizon of the SR forecast. During the first three time periods, the transition to the succeeding state is deterministic, as the SR is assumed to apply with a probability of 1. From the fourth period on, the optimizer can still choose from three actions, but after an action is performed the cleanliness configuration does not necessarily end in the next state, but is determined by the stochastical SR for that time step.

5.2 Results

The evaluation is divided into four steps, each of which was performed with regard to one of the following parameters of the optimization algorithm:

- Alpha value of $\alpha \in \{1, 1.5, 2, 5, 10\}$
- Forecast accuracy (fa) of either 90% or 50% and forecast deviation (fd) of 0.75, 1.25 or randomly selected from a normal distribution with mean of 1 and a standard deviation of 0.002.
- Cleanliness resolution $cleanRes \in \{0.0025, 0.005\}$
- Optimization horizon $h \in \{5, 10\}$

In a first step the α-value was adjusted based on the output computed on six independent SR Input sets. As a second step the use of various stochastic SR Input sets is implemented. The question arises to what extend the optimizer is influenced by possible inaccurate forecasts. Another point influencing the performance of the optimizer is the use of a higher cleanliness resolution. In order to reduce the state space, the cleanliness resolution was adjusted and as a result, different states were possibly considered equal. As this has an effect on the one-stage reward, its influence on the output of the optimizer must be analyzed. The last aspect to be evaluated is the horizon of the optimization, as it influences the size of the state space and therefore the runtime of the algorithm. Hence the possibility to implement the algorithm with a shorter time horizon is evaluated.

5.2.1 Alpha value

The application of the hypothetical α-value to penalize a cleaning action, instead of a detailed model for cleaning costs or water consumption, requires an adjustment of this value in a first step. To enable a consistent performance of the optimizer with regard to the various SR inputs, the influence of the α-value on the output data has been evaluated for the α-values 1, 1.5, 2, 5, and 10, as described above.

When valuating one cleaning action and its corresponding water consumption with an α-value of 1.5 or less, no significant increase in energy production can been achieved. The increase is in a range of +0.35% to +0.01%, whereas the changes in the number of cleaning actions are in a range of -18% and +2% depending strongly on the SR Input .

A stronger effect can be seen when the alpha value is defined higher. Figure 5.1 shows the output values of different simulations as a percentage of the output values generated applying

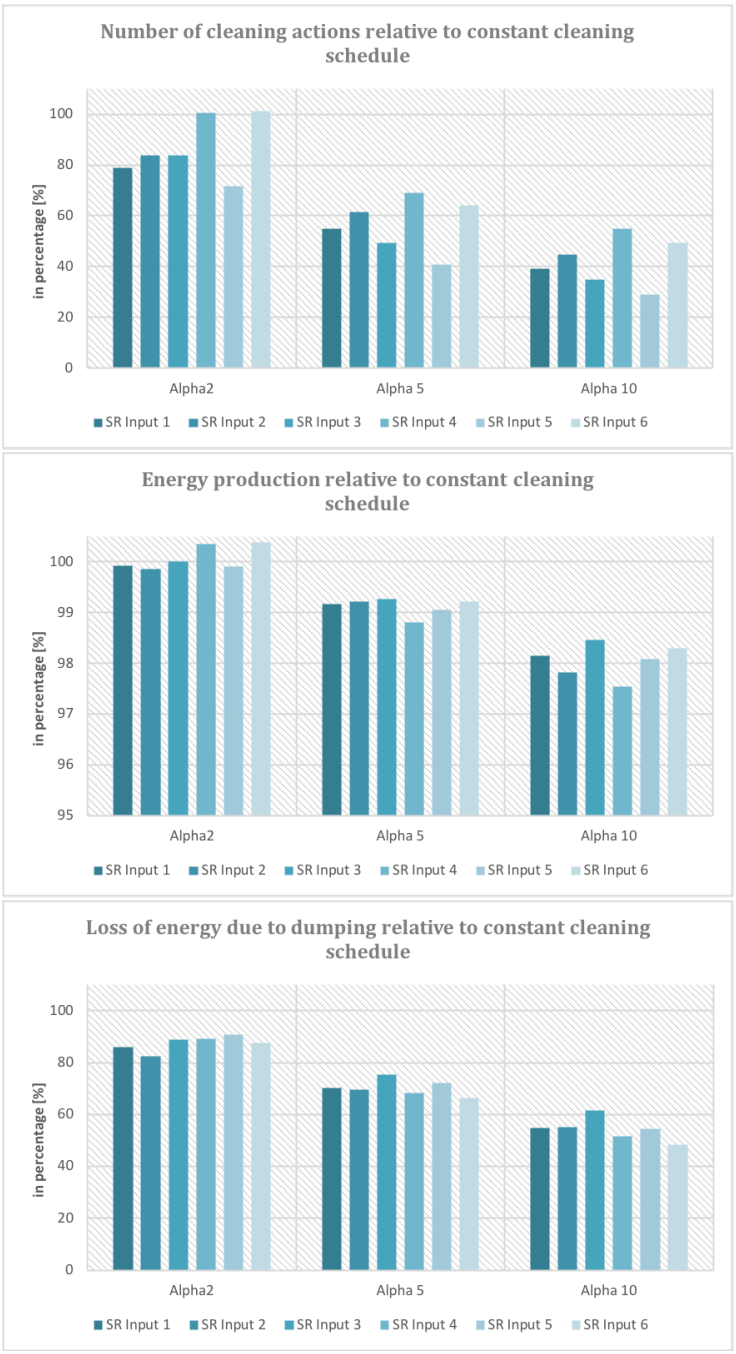

Figure 5.1: Simulation output for the alpha values 2, 5 and 10 compared to the output resulting from a constant cleaning schedule for the corresponding SR Input

a constant cleaning schedule. In the first diagram the number of cleaning actions is depicted.

It can be seen that a higher α-value leads to an increased reduction of cleaning actions. The highest reduction was obtained for a α-value of 10 and the SR input 5, being more than 71% which translates to 53 cleaning actions (184 for the constant cleaning schedule). At the same time an energy production only 2% lower than in case of a constant cleaning schedule was produced. How is this performance achieved? As shown in the bottom graph of Figure 5.2 the energy loss due to reaching the maximum power limit, i.e. dumping is significantly reduced for higher alpha values. For the most water-saving scenario ($\alpha = 10$), the energy dumping can hence be reduced by around 50%. The conclusion can be drawn, that the optimizer is able to forecast events of dumping and allows for a lower reflectance of the mirrors, in case that the need for defocusing would possibly occur.

In Figure 5.2 the tradeoff between the energy production and water consumption is depicted as a two-dimensional plott. Two conclusions can be drwan from this data:

- Water savings of up to 20% are achievable without reducing the energy output.
- While aiming to reduce the water consumption by around 50% the optimizer can still produce an energy output of 98% – 99%.

Within Figure 5.2 the correlation of the number of cleaning actions and the energy output can be seen. Less cleaning leads to a lower energy production, denoting a tradeoff line.

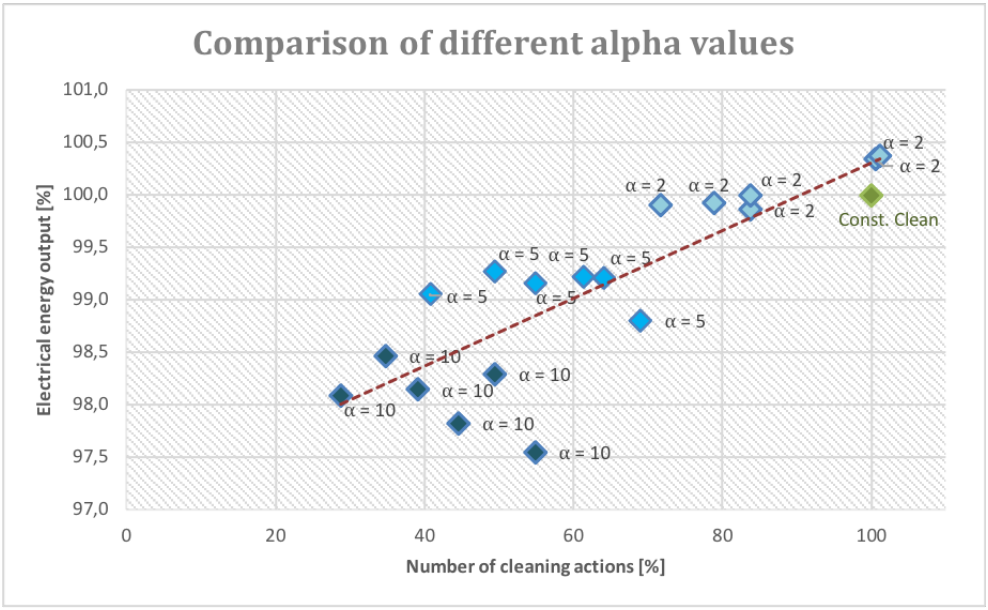

Figure 5.2: Simulation output for six SR Input sets and α-values 2, 5 and 10 compared to the output resulting from a constant cleaning schedule for the corresponding SR Input

As the evaluation has shown, a reduction of water consumption can be achieved by the optimizer. On the other hand, at the same water consumption of 100%, the energy output could only be increased slightly by about 0.4% (see Figure 5.2) compared to the constant cleaning schedule. One reason for that is the high frequency of cleaning within the constant cleaning schedule, which lead to an average reflectivity range of $[0.95, 0.98]$ for all SR Inputs. Hence, the constant

cleaning schedule already produced a high level of electricity and therefore further improvements were difficult to achieve.

5.2.2 Forecast accuracy

For the above presented evaluation of the optimization deterministic inputs were used, which implies that the optimization algorithm had knowledge of the SR occurring in the upcoming periods. Therefore, all transitions probabilities were set to 1. In reality this is not the case and

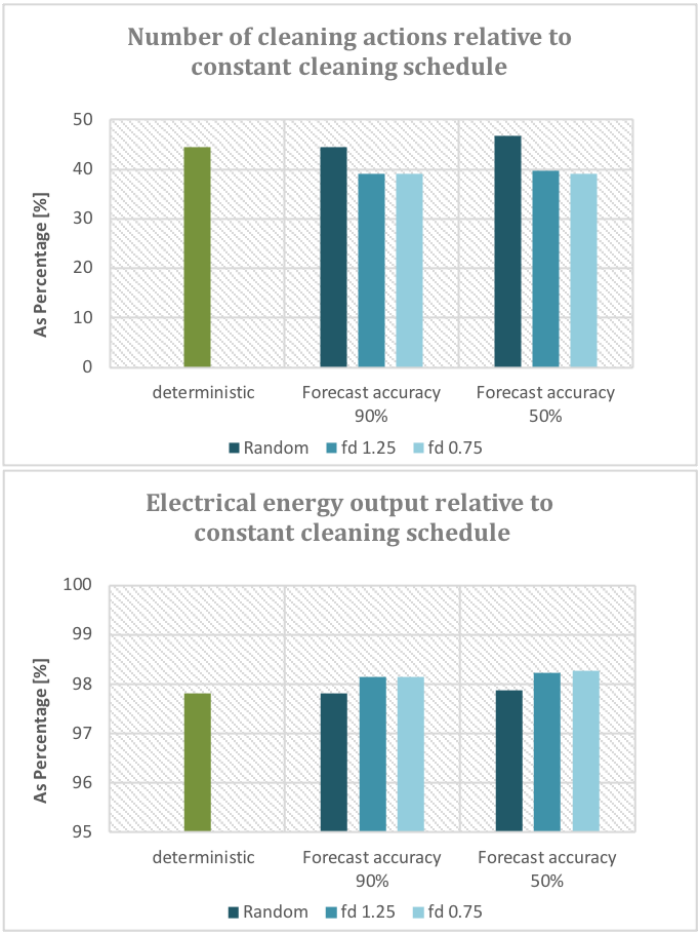

Figure 5.3: Output for SR Input 3 and α-value = 10, relative to the output for the constant cleaning schedule with same SR Input .

thus, in a second step, the influence of a reduced forecast accuracy was examined. Therefore, the optimizer was adapted in the following way: Within the first test it was assumed that with a probability of 0.9 the SR after period three will be correctly forecasted. With the remaining probability of 0.1 the optimizer overestimated the forecast by 25%. Within the second test, the optimizer underestimated the SR by 25%. As in reality both options would occur, another test was run with a random forecast deviation. Again the optimizer forecasted the correct SR by 0.9 chance, though

with a probability of 0.1 the SR would deviate by fd, whereas fd was drawn randomly from a normal distribution with a mean of 1 and a standard deviation of 0.002. The ratio 0.9 to 0.1 is very optimistic and therefore the same evaluation as described above was conducted with a forecast accuracy of only 0.5, explicitely the correct SR as well as the forecast deviation both occurred with a probability of 0.5.

The results can be seen in Figure 5.3. The output for the optimization based on a deterministic SR is almost identical to the output based on a forecast with random deviation. For the latter a small improvement can be seen for a forecast accuracy of 90% compared to an accuracy of 50%. The results based on an fd of either 1.25 or 0.75 are the same both for the reduction of cleaning actions as well as the energy output.

A focus has to be placed on the greater reduction of cleaning actions the optimizer can achieve with a fd of 1.25 or 0.75 compared to a random fd or the deterministic SR. Moreover a higher energy output can also be obtained. The optimizer designed with a deterministic SR was assumed to outperform the other variations due to its knowledge of future events. This is not the case, neither for a constant over estimation ($fd = 1.25$) nor for a constant under estimation ($fd = 0.75$). A complete understanding of these apparently contradicting results could not be reached within the frame of this work. Still, the potential further gain that these results hint towards, might be a sign of higher performance potential with a further optimization of the algorithm.

5.2.3 Influence of cleanliness resolution

Due to the necessity to reduce the state space, the cleanliness resolution is reduced after period three to a value of either 0.005 or 0.0025. The effect of this reduction has been evaluated using three different SR Input sets. The results are illustrated in Figure 5.4. The performance of the optimization algorithm using a *cleanRes*-value of 0.0025 compared to a *cleanRes*-value of 0.005, is better for all SR Inputs. This improvement is significant for SR Input 3, as this dataset has a mean SR of 0.0042 compared to a mean SR of 0.066 for SR Input 2. Therefore, its loss of precision relative to the mean SR is higher. In that case the optimizer tends to misinterpret the calculated rewards and is not able to compute the optimal decision. At the same time, the energy output does not change more than 0.2% for all input sets. In summary it can be seen that the cleanliness resolution has a strong impact on the performance of the optimizer and the number of cleaning actions that the optimizer can save. Hence, it is preferred to keep the *cleanRes*-value high, as long as runtime allows.

5.2.4 Optimization horizon

Finally, the performance of the optimizer can be evaluated with regard to the optimization horizon. For all simulations performed with a variation of the parameters evaluated in the previous sections, the optimizer defined with a horizon of 5 days achieves a greater cleaning reduction. At the same time, the energy output is lower for a time horizon of 5 than that produced for a time horizon of 10. Again a tradeoff line could can be imagined, that implies the almost linear relationship of water reduction and energy output. An interesting phenomenon can be seen for the cluster of data points, which are both the outputs for the horizon of 5 and 10. The optimizer performs as good for both, whereas the alpha value for the time horizon of 10 is set to 5 and for the time horizon of 5 it is set to 10. As all results tend to lie on the same assumed tradeoff line, the conclusion can be drawn that the optimizer does not necessarily perform better for a time

horizon greater than 5. Further simulations are required to evaluate the transferability of this implication to other SR input data.

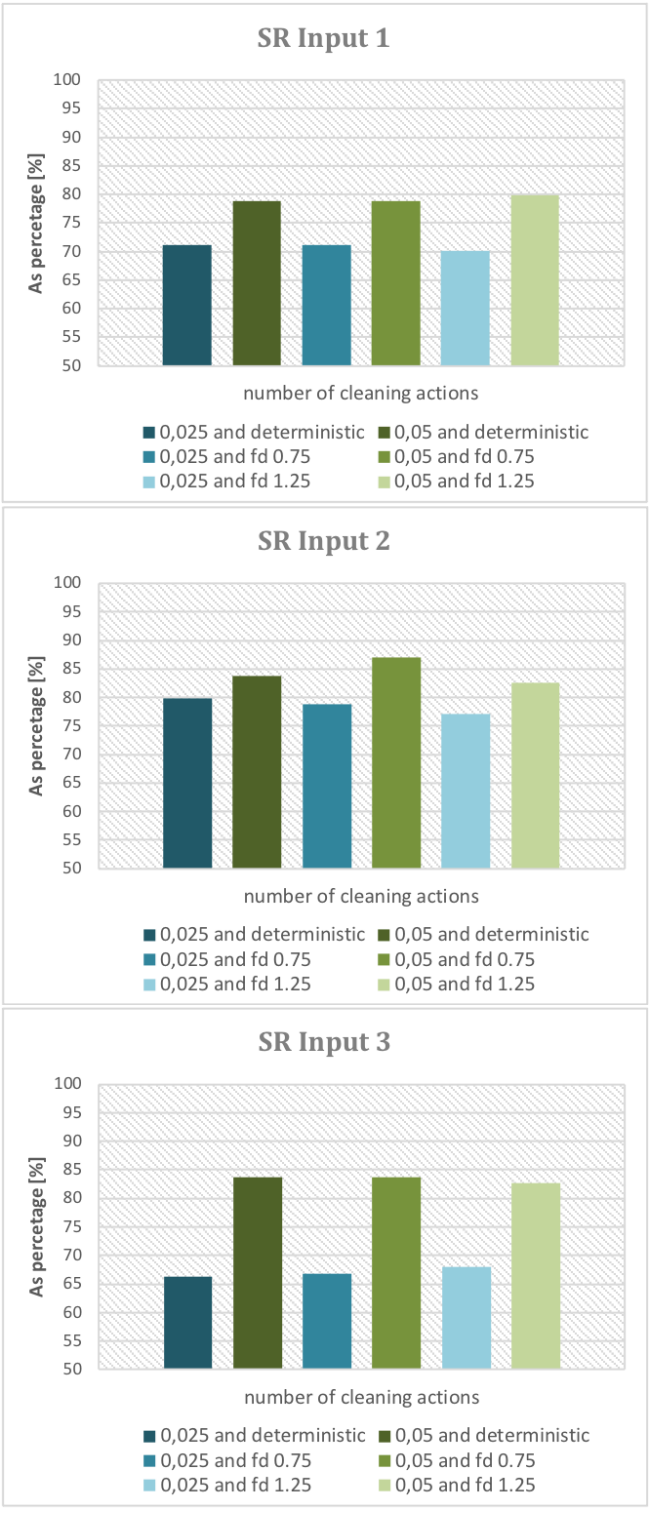

Figure 5.4: Simulation outputs for $cleanRes$ 0.0025 and 0.005 relative to the output of a constant cleaning schedule for the same SR Input and an α-value of 2

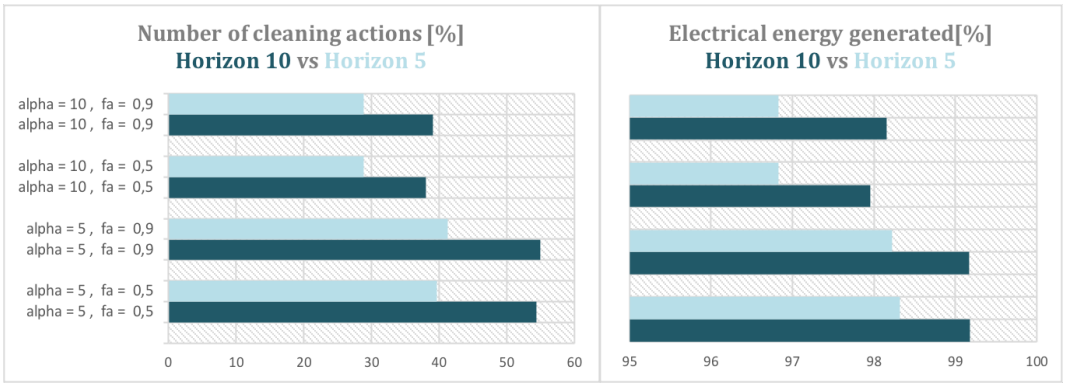

Figure 5.5: Simulation outputs for optimization horizon 5 or 10 compared to the output resulting from a constant cleaning schedule for the same SR Input

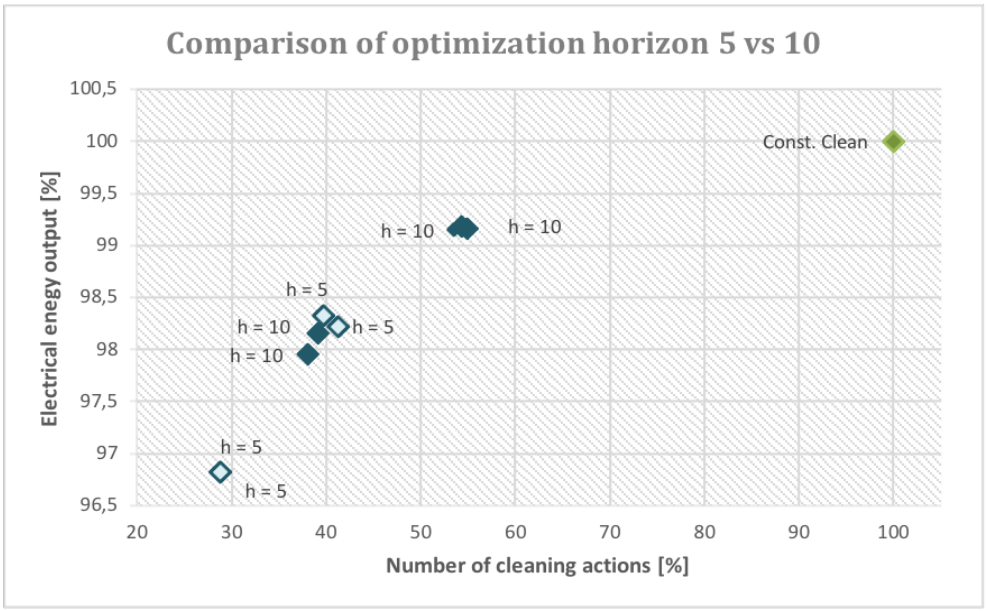

Figure 5.6: Simulation outputs for optimization horizon 5 or 10 compared to the output resulting from a constant cleaning schedule for the same SR Input and an α-value of 2

6 Conclusion

The tradeoff between water consumption and energy output of a Concentrated Solar Power plant has been approached within this thesis. As a first step, the technical properties of the problem have been evaluated to derive certain requirements to the algorithm. Based on these requirements, the theory of a Markov decision process was found to describe the problem very well and thus could be applied. The algorithm used to implement the optimizer was based on the idea of Value Iteration.

The primary aim of the optimizer was the reduction of cleaning actions and thus the possibility to save water. As the evaluation confirmed, the optimizer was able to achieve significant water savings for all six SR input sets. The extent of this saving can be regulated by a single parameter, the α-value. Compared to a constant cleaning schedule, water savings of up to 20% were achieved, without significantly reducing the energy output. The water consumption can be reduced by 40% and for some SR input sets even by 70%, when accepting a small loss in energy yield in the range of 1 - 2 %. It can be concluded that the optimization algorithm developed within this thesis is able to determine a cleaning schedule that saves water while maintaining a high electrical output. In addition, the application as an online algorithm makes its use more feasible for real-life application.

While evaluating the performance of the optimizer, the following aspects have been found to be a subject of future research:

While the study showed significant improvement compared to a constant cleaning schedule with a relatively high frequency of cleaning actions, the question arises how large the improvement is quantitatively when comparing to a fixed cleaning schedule with a reduced cleaning frequency.

Due to the focus of improving multiple parameters within this thesis, an evaluation using a greater dataset with multiple years could not be conducted and is left to further research on this topic. The dependency of some parameters on the SR input set suggests the possibility of further improvement of the algorithm by conducting a parameter study. The parameters that were found to have an influence on the performance of the optimizer are the time horizon, the cleanliness resolution and the alpha value. The latter was found to be very suitable as an adjusting factor to weighting water reduction against energy output.

The precise mechanism leading to the contradictory finding of a better performance of the optimizer with an inaccurate forecast, compared to the deterministic forecast remains to be investigated. This implies further research on the dependency of the performance on the accuracy of the soiling rate forecast, that could also include different levels of accuracy of the meteorological data.

A greater focus on the correlation of different parameters could result in interesting findings that can contribute to improving the robustness of the optimizer. In order to investigate these correlations in more detail, a lager data sample would be needed.

A further study could assess the performance of the optimizer using input sets that comprise

real SR data and corresponding meteorological data in order to draw further conclusions on its performance in real life application. Moreover, the extension of the optimizer to account for natural cleaning events as well would be a great benefit to the assessment.

List of Figures

Bibliography

[And08] Solar Millenium AG . Die Parabolrinnen-Kraftwerke Andasol 1 bis 3. Available at http://large.stanford.edu/publications/coal/references/docs/Andasol1-3deutsch.pdf, 2008. Retrieved March 25, 2021.

[Bel66] R. Bellman. Dynamic programming. *Science*, 153(3731):34–37, 1966.

[BF81] K. D. Bergeron and J. M. Freese. Cleaning strategies for parabolic-trough solar-collector fields; guidelines for decisions. Available at https://www.osti.gov/biblio/6376410, 1981. Retrieved March 1, 2021.

[CdGV] R. Calvo, E. F. de Gorostiza1, and C. Villasante. Scattering sensor embedded in mirrors for continuous soiling monitoring of CSP plants - Poster for SolarPACES Conference 2020. Available at https://solwaris.eu/wp-content/uploads/2020/11/Poster_Scattering_Sensor_SolarPACES_2020.pdf. Retrieved March 1, 2021.

[Gal13] R. G. Gallager. *Stochastic Processes: Theory for Applications*. Cambridge University Press, 2013.

[HSSN20] A. Heimsath, C. Sutardhio, P. Schöttl, and P. Nitz. Soiling of solar mirrors-impact of incidence angles on csp plant performance. In *AIP Conference Proceedings*, volume 2303, page 210003. AIP Publishing LLC, 2020.

[HRS16] K. Hinderer, U. Rieder, and M. Stieglitz. *Dynamic optimization*. Springer, 2016.

[HDF+17] T. Hirsch, J. Dersch, T. Fluri, J. Garcia-Barberena, S. Giuliano, F. Hustig-Diethelm, R. Meyer, N. Schmidt, M. Seitz, and M. Eck. Solarpaces guideline for bankable ste yield assessment. IEA Technology Collaboration Programme SolarPACES, 2017.

[How60] R. A. Howard. Dynamic programming and markov processes. 1960.

[Kai11] A. Kaiser. Reflektivitätsmessungen und Waschfahrzeugauswahl in einem Parabolrinnen-Kraftwerk (reflectance measurements and selection of cleaning vehicles in a parabolic trough power plant), 2011. Hochschule München, Germany.

[KVH12] K. Kattke and L. Vant-Hull. Optimum target reflectivity for heliostat washing. In *SolarPACES Conference, Marrakech, Morocco, Sept*, pages 11–14, 2012.

[LS] P. Liang and D. Sadigh. Lecture notes to CS221: Artificial Intelligence: Principles and Techniques. Available at https://stanford-cs221.github.io/autumn2019-extra/lectures/mdp1-6pp.pdf. Retrieved March 4, 2021.

[PP08] R. Pitz-Paal. Concentrating solar power. volume 1 of *Future Energy*, pages 171–192. Elsevier Ltd., 2008.

[RAF+19] S. Rohani, N. Abdelnabi, T. Fluri, A. Heimsath, C. Wittwer, and J. G. P. Ainsua. Optimized mirror cleaning strategies in ptc plants reducing the water consumption and the levelized cost of cleaning. In *AIP Conference Proceedings*, volume 2126, page 220004. AIP Publishing LLC, 2019.

Bibliography

[SB] R. S. Sutton and A. G. Barto. Reinforcement learning: An introduction. Available at https://muse.jhu.edu/chapter/2175552#info_wrap. Retrieved March 4, 2021.

[TWW+19] F. Terhag, F. Wolfertstetter, S. Wilbert, T. Hirsch, and O. Schaudt. Optimization of cleaning strategies based on ann algorithms assessing the benefit of soiling rate forecasts. In *AIP Conference Proceedings*, volume 2126, page 220005. AIP Publishing LLC, 2019.

[TBCW+17] H. Truong-Ba, M. Cholette, R. Wang, P. Borghesani, L. Ma, and T. Steinberg. Optimal condition-based cleaning of solar power collectors. *Solar Energy*, 157:762–777, 11 2017.

[TBCP+20] H. Truong-Ba, M. E. Cholette, G. Picotti, T. A. Steinberg, and G. Manzolini. Sectorial reflectance-based cleaning policy of heliostats for solar tower power plants. *Renewable Energy*, 166:176–189, 2020.

[VZS+20] C. Villasante, C. Zuazo, M. Sanchez, J. Babarena, G. Perez, and G. Ubach. SMARTMIRROR: Making CSP plants smarter and more profitable - Poster for SolarPACES 2020. Available at https://solwaris.eu/wp-content/uploads/2020/11/Poster_SMARTMIRROR_SolarPACES_2020.pdf, 2020. Retrieved March 1, 2021.

[WWD+18] F. Wolfertstetter, S. Wilbert, J. Dersch, S. Dieckmann, R. Pitz-Paal, and A. Ghennioui. Integration of soiling-rate measurements and cleaning strategies in yield analysis of parabolic trough plants. *Journal of Solar Energy Engineering*, 140(4), 2018.

[WWD+17] F. Wolfertstetter, S. Wilbert, S. Dieckmann, J. Dersch, R. Affolter, P. Bellmann, and A. Ghennioui. Comparison of Soiling Rate Data from two Sites and ist Application to Yield Analysis. 2017.

[WWT+19] F. Wolfertstetter, S. Wilbert, F. Terhag, N. Hanrieder, A. Fernandez-García, C. Sansom, P. King, L. Zarzalejo, and A. Ghennioui. Modelling the soiling rate: Dependencies on meteorological parameters. In *AIP conference proceedings*, volume 2126, page 190018. AIP Publishing LLC, 2019.

Hiermit versichere ich, Simone Horstmann, dass ich die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, dass alle Stellen der Arbeit, die wörtlich oder sinngemäß aus anderen Quellen übernommen wurden, als solche kenntlich gemacht worden sind, und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt wurde.

Köln, den 10. Mai 2021

Simone Horstmann