MODEL PREDICTIVE CONTROL OF A SOLAR-THERMAL REACTOR

BY

María Elizabeth Saade Saade

B.S., Monterrey Institute of Technology, 2006

M.S., University of Colorado, 2009

A Thesis Submitted to the

Faculty of the Graduate School of the

University of Colorado in Partial Fulfillment

of the Requirement for the Degree of

Doctor of Philosophy

Department of Chemical and Biological Engineering

2013

This thesis entitled:

Model Predictive Control of a Solar-Thermal Reactor
written by María Elizabeth Saade Saade

has been approved for the Department of Chemical and Biological Engineering

____
Alan W. Weimer

____
David E. Clough

*Date*_______________

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

ABSTRACT

Saade Saade, María Elizabeth (PhD, Chemical and Biological Engineering)

Model Predictive Control of a Solar-Thermal Reactor

Thesis directed by Professor Alan W. Weimer and Professor David E. Clough

Solar-thermal reactors represent a promising alternative to fossil fuels because they can harvest solar energy and transform it into storable and transportable fuels. The operation of solar-thermal reactors is restricted by the available sunlight and its inherently transient behavior, which affects the performance of the reactors and limits their efficiency. Before solar-thermal reactors can become commercially viable, they need to be able to maintain a continuous high-performance operation, even in the presence of passing clouds. A well-designed control system can preserve product quality and maintain stable product compositions, resulting in a more efficient and cost-effective operation, which can ultimately lead to scale-up and commercialization of solar thermochemical technologies.

In this work, we propose a model predictive control (MPC) system for a solar-thermal reactor for the steam-gasification of biomass. The proposed controller aims at rejecting the disturbances in solar irradiation caused by the presence of clouds. A first-principles dynamic model of the process was developed. The model was used to study the dynamic responses of the process variables and to identify a linear time-invariant model used in the MPC algorithm. To provide an estimation of the disturbances for the control algorithm, a one-minute-ahead direct normal irradiance (DNI) predictor was developed. The proposed predictor utilizes information obtained through the analysis of sky images, in combination with current atmospheric measurements, to produce the DNI forecast.

In the end, a robust controller was designed capable of rejecting disturbances within the operating region. Extensive simulation experiments showed that the controller outperforms a finely-tuned multi-loop feedback control strategy. The results obtained suggest that our controller is suitable for practical implementation.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER ONE

INTRODUCTION

1.1 Background and Motivation

The increasing energy consumption and the inevitable fact that fossil fuels are a limited resource have led humankind to the search for an alternative energy resource. Solar energy is the most abundant source of energy, but its availability is not uniform, with higher direct normal irradiation in equatorial regions of the Earth [1]. As a result, many efforts have been dedicated to the development of technologies that can transform solar energy into chemical energy, i.e., into a fuel that can be stored and transported to places outside the earth's sunbelt [2]. Solar-thermal reactors are one example of such technologies.

Solar-thermal reactors use concentrated solar energy as their main energy source. Solar energy is typically concentrated using parabolic mirrors and directed towards a receiver (or reactor), where the heat is used to drive an endothermic, high-temperature reaction. These reactors can be used to produce storable fuels [2], to synthesize commodities or to process toxic waste [3].

The use of solar-thermal reactors to produce storable, transportable fuels has spiked some interest in the recent decades [4]. Some examples include the production of hydrogen through thermochemical water-splitting cycles [5][6], the decarbonization of fossil fuels to produce hydrogen (solar reforming) [7], and the steam gasification of carbonaceous materials to produce synthesis gas [8].

Another example of such processes is the solar-thermal conversion of biomass into synthesis gas (H_2 and CO), which can be catalytically reformed to fungible liquid fuels and used throughout the world [9].

Biomass gasification is conventionally carried out in fluidized bed, fixed bed or downdraft gasifiers, which operate at temperatures below 1000 °C [10]. Tars, which can foul the catalysts used to convert the synthesis gas into liquid fuels, form at temperatures below 1000 °C [11]. The use of concentrated solar energy allows biomass gasification to be carried out at higher temperatures, reducing the amount of tar produced and resulting in a higher-quality synthesis gas [12]. In addition, conventional processes burn 25 to 30% of their feedstock in order to generate enough energy to drive the reaction [10]. The use of solar-thermal energy will increase efficiency, by eliminating the need to consume some of the feedstock to produce heat. Solar-thermal biomass gasification has been studied since the ‘80s [13] and continues to be studied in current times [14], [15].

One of the most important challenges that solar thermochemical processes face is dealing with the intermittent nature of solar irradiation. In these processes, the main source of energy cannot be manipulated, thus it acts as a disturbance from the control point of view. The incoming energy to the system depends not only on seasonal and daily variations, but also on atmospheric conditions such as cloud cover, humidity and air transparency [16]. Changes in solar irradiation directly affect the performance of solar-thermal reactors, causing intermittent shutdowns and start-ups, complications in the purification processes downstream of the reactor, and damage to the reactor materials due to thermal shock. Being able to maintain a continuous high performance operation, even in the presence of transients in the solar irradiation input, is one of the main concerns of the feasibility of solar-thermal processes.

Previous work by Petrasch et al. [17] has shown that implementing a control system in a solar-thermal reactor makes its operation more efficient, improving the quality and stability of the product composition throughout transients in the solar irradiation input. These improvements also lead to an increase in the solar-to-chemical energy conversion efficiency, which translates into a more efficient operation. In other words, a robust control system will provide an important competitive advantage to any solar thermochemical process.

This study proposes the following hypotheses: (1) a model-based control system that manipulates the flow rates into the reactor will have a better performance than a traditional feedback control system; (2) using images of the sky and other atmospheric measurements to anticipate the incoming solar irradiation will allow the controller to operate in a predictive manner, improving the performance of the reactor even further.

The fact that model-based control systems have a better performance than conventional controllers has been recognized by Camacho et al. in their work on control systems for solar power plants [18]. Previous work that has succeeded using a model–based predictive technique for solar power plants [19] shows promise that this technique will also be successful for solar-thermal reactors, since they have similar characteristics (highly nonlinear systems with controllers designed to reject disturbances in the solar irradiation). An overview of previous work on the control of solar-thermal reactors and solar power plants will be presented in the next section.

1.2 Current State of the Art

The only known previous attempt to design a control system for a solar-thermal reactor was by Petrasch et al. [17]. In their work, they developed a Linear Quadratic Gaussian with Loop Transfer Recovery (LQG/LTR) controller [20] that manipulated the reactant flows according to changes in the measured variables (product composition and reaction temperature). The controlled system showed an improved performance over the uncontrolled system [17]. However, we propose that the performance can be improved even further by implementing a predictive control scheme that takes into account measured disturbances.

Even though there has not been much research on the control of solar-thermal reactors, a broad spectrum of work has been developed in terms of control of solar power plants, which are also highly nonlinear systems that use solar energy as their main source of heat. Camacho et al. performed an extensive survey on the control techniques applied to solar power plants throughout the years [16], [19]. A technique that has been successful in the control of solar power plants is Model Predictive Control (MPC) [21]–[24]. The term MPC includes a wide range of control strategies that make an explicit use of a model of the process in order to obtain the control signal by minimizing an objective function [25].

A large number of the control systems developed for solar power plants were designed for distributed collector systems, which consist of a series of parabolic mirrors that reflect solar irradiation into a pipe containing a heating fluid used to produce steam for electricity generation [26]. Typically, solar-thermal reactors operate within a central receiver system, in which a large number of flat mirrors, or heliostats, track the incident sunrays and concentrate them towards a solar receiver, usually located on the top of a tower [27]. In this type of system, little has been done in terms of manipulating the inlet flow rates to control outlet variables such as temperature and pressure. One of the few control techniques developed [28] focuses on manipulating the water inlet flow rate to control the outlet steam conditions. The controller adapts to two different models: one used in the presence of clouds and one for clear sky. However, manual interventions were necessary to adjust the control scheme since it does not incorporate an adaptive algorithm.

Cloud tracking has been used in the past to mitigate disturbances in central receiver systems [29]. This technique detects the presence of clouds that cover part of the heliostat field and will interfere with the solar irradiation hitting the receiver. When a cloud disturbance is detected, a master control points the heliostats gradually away, so that the receiver temperature can decrease before the cloud covers the sun, reducing its risk of damage due to thermal shock. A similar system for cloud detection will be applied to the control system developed in this dissertation; however, instead of manipulating the direction of the heliostats, this project will attempt to predict the resulting incoming radiation and use it as an input for an MPC system. The inlet flow rates into the reactor will be manipulated so that the product composition can be maintained constant throughout transients.

1.3 Research Goals and Approach

The main goal of this study is to design and implement a model-based control system that will maximize reactor productivity given the inevitable variations in the solar energy flux. The steps to follow in order to achieve this include (1) the development of a simplified dynamic model of the reactor system and the validation of this model with experimental data, (2) the development of an algorithm that predicts solar irradiation using digital images of the sky in combination with atmospheric measurements, (3) the design and implementation of the model-based predictive control algorithm, and (4) the integration of the vision-based solar irradiation predictor with the model-based predictive control algorithm, to be tested through simulations. The final stage of the project (5) consists in the implementation of a prototype control system at the High Flux Solar Furnace (HFSF) at the National Renewable Energy Laboratory (NREL). However, the scope of this project is limited to goals (1) to (4), leaving the real system implementation as future work.

A model derived from first principles was chosen because it can provide more insight into the dynamics of the reactor than an empirical model based solely on experimental data. The system is highly nonlinear, so a single linear "black box" model might present inaccuracies that will affect the performance of the controller. It is possible to get a separate linear model for different operating points and to have the controller choose the most suitable one, but obtaining these models from experimental data would be an expensive process with a high degree of plant intrusion. In addition, a fundamental model can be applied to a wide range of systems with similar characteristics and to different operating conditions by just changing the values of the physical properties and other parameters.

The control system proposed in this study is an MPC system, which makes use of a model of the process to estimate future values for the outputs. The control signal is calculated by minimizing the error between the future outputs and a reference trajectory. This control strategy has been shown to be successful in dealing with multiple input–multiple output (MIMO) systems and with complex processes [25]. Furthermore, this technique has been applied extensively to control solar power plants [19], showing that this strategy is successful when dealing with measured disturbances, as in the case of solar irradiation.

A diagram of the proposed control system is presented in Figure 1. A digital charge-coupled device (CCD) will capture images of the sky at a sampling interval of one minute. These pictures will be analyzed using on-line pattern recognition techniques. The information retrieved from the images will be used, in combination with atmospheric measurements, to anticipate the incoming solar irradiation at future instances. The predictions will be made through classification systems trained using the data available from the Solar Radiation Research Laboratory (SRRL) from NREL [30]. The predicted solar irradiation will serve as an input for the MPC, which will ensure steady product composition through the manipulation of the flow rates into the reactor, despite transients in solar irradiation. In addition, the system will include an attenuator control, which will eliminate fast transients that could cause damage to the reactor materials. The system will be coupled with feedback control for the reactor temperature and product compositions.

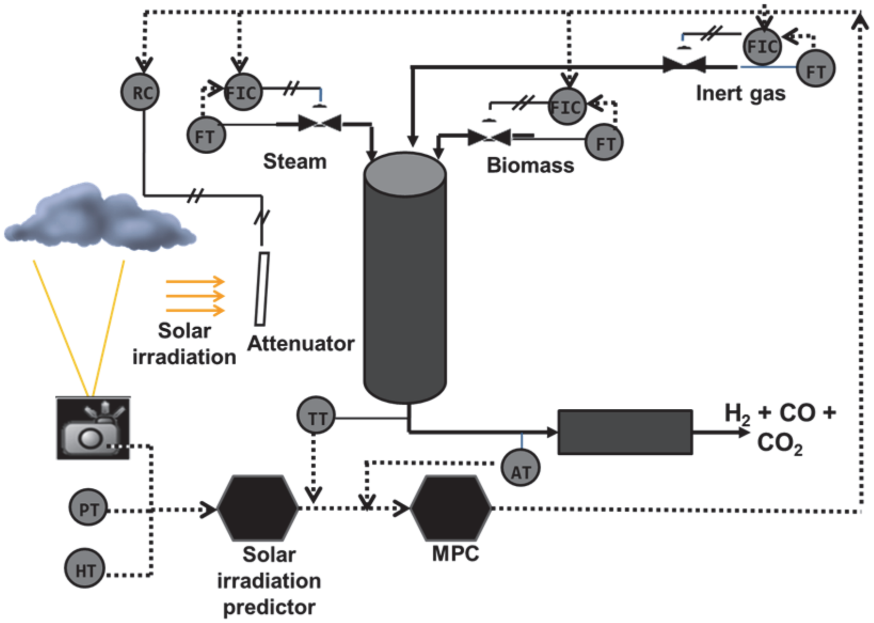


Figure 1.Diagram of proposed control system

1.4 Significance

This study is the first attempt of predictive control of a solar-thermal reactor using digital images of the sky. The control system that will result from this project will be applicable not only to the gasification of biomass, but also to any solar-thermal process with similar characteristics. The control system will enable the continuous operation of the reactor in the presence of uncontrollable disturbances in the incoming solar irradiation, eliminating unnecessary shutdowns and startups and thus making the process more feasible. In addition, it will permit a smooth transition in the case of required shutdowns, preventing thermal shock that can cause expensive material damages to the reactor. A well-designed control scheme can be the determining factor for the feasibility of a process of this type, and operating it in a way that maximizes productivity is crucial to make it profitable.

1.5 Scope of Thesis

After this introductory chapter, Chapter Two covers the derivation and validation of a simplified dynamic model of the process, based on unsteady mass and energy balances. Chapter Three describes the development of a one-minute-ahead solar irradiation predictor, based on images of the sky. Chapter Four presents the development of the model predictive controller, along with an evaluation of its performance. The final chapter summarizes the conclusions resulting from this work and provides recommendations for further work.

CHAPTER TWO

DYNAMIC MODEL OF A SOLAR-THERMAL TRANSPORT-TUBE REACTOR

2.1 Introduction

A transport-tube reactor uses concentrated solar irradiation in order to drive the endothermic biomass gasification reaction for the production of synthesis gas [9]. As in any solar-thermal process, the intermittent nature of solar irradiation presents a challenge. Changes in solar irradiation can directly affect the performance of solar-thermal reactors, possibly causing unnecessary shutdowns and start-ups, complications in the purification processes downstream of the reactor, and damage to the reactor materials due to thermal shock. A robust process control system can mitigate these problems and allow for continuous high performance operation of the reactor, making it feasible and more profitable. Despite the transient nature of solar irradiation, the controller will ensure steady product composition through the adjustment of the flow rates into the reactor. In addition, the control system will manipulate an attenuator to moderate solar power, which will attenuate upward transients that could cause damage to the reactor materials.

As explained in Chapter One, the system proposed in this study is a model predictive control (MPC) system, which makes use of a model of the process to obtain the control signal by minimizing the error between the predicted process trajectory and a reference trajectory [25]. Since the control algorithm makes explicit use of a model of the plant, the first step is to describe the dynamic behavior of the process and analyze its transient response to variations in cloud cover. A model derived from first principles was chosen because it can provide more insight about the dynamics of the reactor than an empirical model based solely on experimental data. The plant is highly nonlinear, so the use of a single linear “black box” model would be inappropriate. It is possible to get a separate linear model for different operating points and to have the controller choose the most suitable one, but obtaining these models from experimental data would be an expensive process with a high degree of plant intrusion. In addition, a fundamental model can be applied to a wide range of systems with similar characteristics and to different operating conditions by changing the values of the physical properties and other parameters.

Several models have been developed by others for reacting particles flowing through tubular reactors at high temperature. Some investigators have studied fundamental aspects of gasification of biomass, gasification of charcoal or pyrolysis [17][31]. Others have modeled solar-thermal reactors for the dissociation of metal oxides [32] or the decarbonization of methane [33]. However, these models were developed with the goal of providing a better understanding of the system, so they include many details that cause their computational times to be extremely long. Therefore, they cannot be solved in real time. We attempt to overcome this limitation by developing a simple but fairly accurate model that can be used to develop a control algorithm.

This chapter presents a detailed description of a general simplified dynamic model that was developed based on unsteady mass and energy balances. The model can be applied to a wide range of solar-thermal reactors. It was validated experimentally using data for inert gases obtained at the High Flux Solar Furnace (HFSF) at the National Renewable Energy Laboratory (NREL). In addition, the model was applied to a solar reactor for the gasification of carbon and validated with more experimentation carried out at the HFSF at NREL. The model proved to be computationally efficient, but still capable of describing the system within 5% accuracy.

2.2 Simplified Dynamic Model of a Solar-Thermal Reactor

Mathematical models of physical processes can include different levels of detail, depending on their intended use. The model described in this chapter is to be used to develop a control algorithm; therefore, it needs to be simple enough to be solved in real time, but at the same time it needs to describe the main physical characteristics of the system accurately.

The model was developed and validated for a reactor with a reflective cavity, which allows for faster collection of the experimental data. The following sections describe the development of this model, as well as the model validation using experimental data obtained at NREL.

2.2.1 Domain

The reactor consists of a 35-cm long transport tube with an internal diameter of 20 mm. (O.D. = 25 mm) enclosed by a reflective cavity. It receives solar radiation through a quartz window placed directly in front of a concentrator. Particles entrained in an inert gas (argon in this case) flow inside the tube and are indirectly heated to high temperatures (1000-1500 K) by the incoming solar irradiation. The base model geometry is shown in Figure 2. The dimensions shown in the figure were chosen to match those of the reactor used at NREL for model validation. The tube is made from Hexoloy® SA Silicon Carbide which has an emissivity of 90% [34]. The inputs are the power level, P_{solar}, and the inlet flow rates of gas and particles. The measured variables, or outputs, are the temperatures, T, and the product molar flow rates, $\dot{n}_{i,out}$.

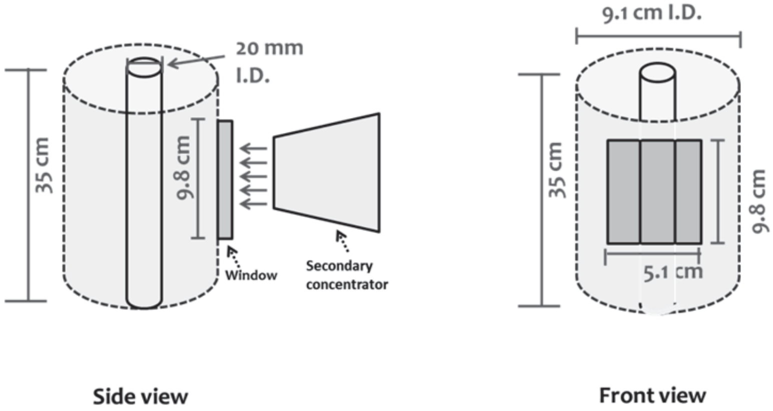


Figure 2. Base model geometry and dimensions

2.2.2 Model Strategy and Assumptions

The main assumptions employed in developing the model are: (1) changes in the radial dimension are negligible, (2) gases are transparent to radiation, (3) radial and angular components of the gas velocity are negligible, (4) the particles are small enough to be entrained by the gas, (5) the particles are isothermal, (6) the tube surfaces and the particles are treated as black bodies for the radiation modeling, (7) the system is at constant pressure, (8) all the gases are ideal gases, and (9) the back of the tube does not receive significant direct solar irradiation. Justifications for these assumptions are shown in Table 1. The physical properties were taken from [34]–[40]. The particles used in the simulations were acetylene black with the properties shown in Table 2.

Table 1. Main assumptions and justification

Assumption	Justification	Reference
(1)	Similar models have shown a maximum temperature variation in the radial direction of about 10 K.	[32]
(2)	Particles have high emissivity and high surface area for heat transfer. Most of the radiation is transferred to the gas by convection.	[35], [36]
(3)	There is no external force that would cause radial or angular velocities. The fluids flow into the reactor axially.	
(4)	Stokes number (St) <<1 ($1 \times 10^{-7} – 2 \times 10^{-7}$)	[41]
(5)	Time for internal conduction (tic) << time for radiation (tr) t_{ic}=2 x 10^{-10} s; t_r=1 x 10^{-4} s	[42], [43]
(6)	Tescari et al. applied this assumption to their work and showed that for surfaces with an emissivity higher than 0.8 this assumption is valid.	[44]
(7)	All the gases leave the reactor as they are produced.	
(8)	This assumption is valid at low pressures. Operation occurs at atmospheric pressure.	
(9)	A simple simulation of the geometry showed that almost all the radiation hitting the reflective cavity exits the reactor without reaching the back of the tube.	

Table 2. Properties of the acetylene black particles used in the simulations

Property [units]	Value	Reference
Particle diameter [nm]	40	[36]
Density [kg/m^3]	1750	[36]
Specific surface area [m^2/g]	70	[36]
Heat capacity [J/kg K]	$1355\left(1-\exp\left(-0.00232\left(T_p-298.15\right)\right)\right)+712.2$	[37]
Emissivity	0.97	[35]

In order to simplify the model for computation, the tube has been divided into a series of small disks that are coupled with each other. As a consequence of assumption (1), all the physical properties and outputs are constant inside each disk. Figure 3 shows schemes of the simplification approach.

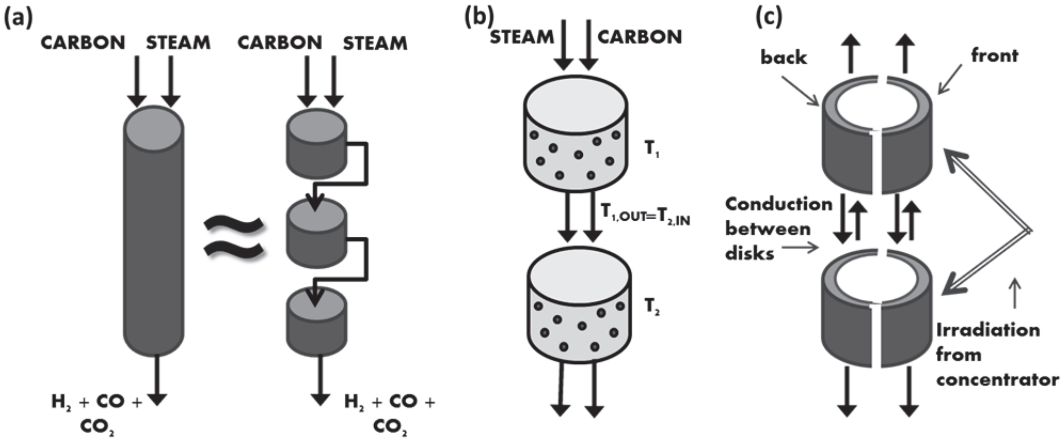


Figure 3. Model simplification schemes: a) The tube was divided into a series of disks. b) Particles and gases were modeled as series of CSTR's. c) The tube wall was divided in front and back

The gas and the particle phases were modeled as a series of continuous-flow stirred-tank reactors (CSTR's), i.e., the outlet conditions of each disk are the inlet conditions of the following disk. On the other hand, to model the tube walls, each disk was divided into a front and a back section because of the significant temperature difference between them (a consequence of assumption (9)).

Figure 3.c shows that conduction between disks is used to couple the energy balances for the tube wall between one disk and the next.

A surface exchange model was used to describe the radiation between the tube walls and the particle phase. Given the small size of the particles, volumetric exchange theory would be a more appropriate way to model this. Figure 4 shows a comparison of the response to a step change in power level for a model that uses surface exchange (dashed line) and a model that uses volumetric exchange theory to model radiation (solid line). The relative difference remains below 9% for all the temperatures and below the experimental error for carbon conversion. Since the implementation of the surface exchange model is less complex, this method was used to develop the equations presented in section 2.2.3. However, this approach might not be appropriate for particles with different optical properties and size distributions. For those cases, volumetric exchange theory should be used unless a comparison of both approaches shows that the results are not affected. For a description on how to implement volumetric exchange theory in a simplified model, see section 2.2.4.2.

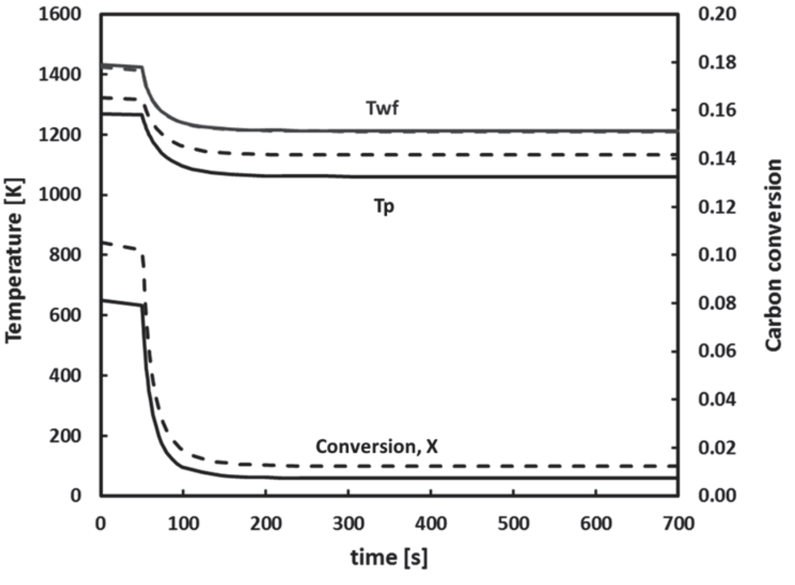


Figure 4. Front wall temperature, particle temperature and conversion response to a step change in power level for a model that uses surface exchange (dashed line) and volumetric exchange theory to model radiation (solid line)

2.2.3 Governing Equations

Each disk can be modeled with four energy balances: two for the tube wall front and back segments, one for the particle phase, and one for the gas phase.

2.2.3.1. Energy Balance for the Tube Wall

The energy balance for the tube wall, after expressing the internal energy in terms of temperature and assuming constant specific heat and constant density, yields:

$$\rho_w C_{p_w} \frac{dT_w}{dt} = \frac{k_w(T_{w_{in}} - T_w)}{\Delta z^2} - \frac{k_w(T_w - T_{w_{next}})}{\Delta z^2} + \hat{Q}_{source} \qquad \textbf{(2-1)}$$

Eqn. (2-1) applies for both the front and back sections of the tube. The first two terms on the right hand side account for heat conduction from the previous disk into the control volume and from the control volume into the following disk, respectively. The control volume is illustrated in Figure 5 for both sections of the tube walls. The term $\hat{Q}_{source}$ represents the volumetric heat source for the tube walls. This term is different for the front and the back wall: the front is heated by direct radiation from the concentrator; while the back is heated mainly by conduction and radiation from the front wall.

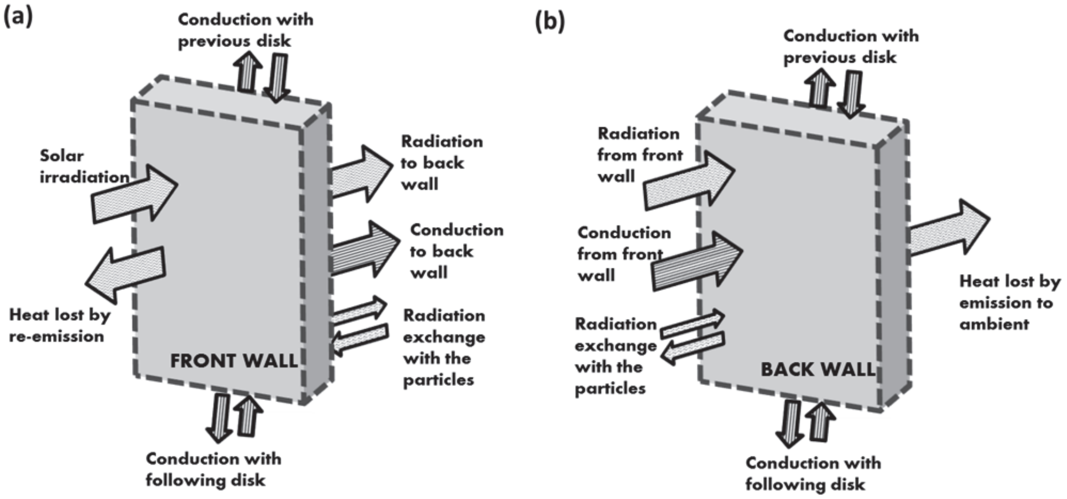

(a)
Conduction with previous disk
Solar irradiation
Heat lost by re-emission
Radiation to back wall
Conduction to back wall
Radiation exchange with the particles
FRONT WALL
Conduction with following disk
(b)
Conduction with previous disk
Radiation from front wall
Conduction from front wall
Radiation exchange with the particles
BACK WALL
Heat lost by emission to ambient
Conduction with following disk

Figure 5. Control volumes for the energy balances of the tube wall. a) Front wall. b) Back wall.

The volumetric heat source for the front section of the tube wall can be represented by:

$$\hat{Q}_{source,f} = \frac{1}{V_w}\left[Q_{solar} - f_{fb}\sigma A_i\left(T_{w,f}^4 - T_{w,b}^4\right) - f_{fp}\sigma A_i\left(T_{w,f}^4 - T_p^4\right) - h_{lam}A_i\left(T_{w,f} - T_g\right) - \sigma A_o T_{w,f}^4\right] - \frac{8k_w}{(\pi r_o)^2}\left(T_{w,f} - T_{w,b}\right) \quad \textbf{(2-2)}$$

The first term on the right hand side of Eqn. (2-2) accounts for the solar radiation incoming to the cavity from the secondary concentrator; the second term accounts for radiation from the front to the back wall; the third term represents the heat transmitted by radiation from the front wall to the particle phase; the fourth term describes the convection from the tube wall to the gas phase; the fifth term represents the heat losses by radiation re-emitted by the tube and lost thorough the cavity window, and it is modeled as a radiation sink, equivalent to the radiation exchange with a blackbody at 0 K. The last term represents the heat transferred to the back wall by conduction. Note that the convection from the wall to the gas was still accounted for, even though it is not as significant as the other modes of heat transfer present in the system.

The volumetric heat source for the back section of the tube wall can be represented by

$$\hat{Q}_{source,b} = \frac{1}{V_w}\left[f_{fb}\sigma A_i\left(T_{w,f}^4 - T_{w,b}^4\right) - f_{bp}\sigma A_i\left(T_{w,b}^4 - T_p^4\right) - h_{lam}A_i\left(T_{w,b} - T_g\right) - \sigma A_o(T_{w,b}^4 - T_{amb}^4)\right] + \frac{8k_w}{(\pi r_o)^2}\left(T_{w,f} - T_{w,b}\right) \quad \textbf{(2-3)}$$

The first three terms in Eqn. (2-3) are also present in Eqn. (2-2). The fourth term corresponds to the heat losses by re-radiation. Unlike the analogous term in Eqn. (2-2), this term represents a radiation exchange between the back wall and the cavity walls, which are assumed to be at ambient temperature. The last term, also present in Eqn. (2-2), represents the heat transferred from the front by conduction.

2.2.3.2 Energy Balance for the Particle Phase

The energy balance for the particle phase, after assuming constant specific heat and constant density inside each disk, is given by

$$\rho_p C_{p_p} \frac{dT_p}{dt} = \frac{\rho_p C_{pp} v_{in}\left(T_{p_{in}} - T_{ref}\right)}{\Delta z} - \frac{\rho_p C_{pp} v_{out}\left(Tp_{out} - T_{ref}\right)}{\Delta z} - \hat{Q}_{conv_{pg}} + \hat{Q}_{rad} + \hat{Q}_{rxn} \tag{2-4}$$

The first two terms on the right hand side of Eqn. (2-4) account for the change in enthalpy due to the particle flow. The rest of the terms represent the volumetric heat sources for convection to the gas phase, the radiation received from the tube wall, and the energy required to drive the chemical reaction.

The convection from the particle phase to the gas phase is given by

$$\hat{Q}_{conv_{pg}} = \frac{h_{conv}(T_p - T_g) A_p n_p}{V_{disk}} \tag{2-5}$$

where h_{conv} is the heat transfer coefficient and was calculated using the Ranz-Marshall correlation [45].

The radiation from the tube walls to the particle phase is represented by:

$$\hat{Q}_{rad} = -\frac{f_{wp} \sigma A_p n_p}{V_{disk}} \left[\left(T_p^4 - T_{w,f}^4\right) + \left(T_p^4 - T_{w,b}^4\right)\right] \tag{2-6}$$

and the heat of reaction is calculated as

$$\hat{Q}_{rxn} = -\Delta H_{rxn} r_{carbon} \tag{2-7}$$

For the case of particles flowing in an inert gas, no reaction takes place and $\hat{Q}_{rxn} = 0$. This term will become important for the case of chemical reaction presented in section 2.4.

2.2.3.3 Energy balance for the Gas Phase

The energy balance for the gas phase, after assuming constant specific heat and constant density inside each disk, is given by

$$\rho_g C_{p_g} \frac{dT_g}{dt} = \frac{\rho_g C_{p_g} v_{in}\left(T_{g_{in}} - T_{ref}\right)}{\Delta z} - \frac{\rho_g C_{p_g} v_{out}\left(Tg_{out} - T_{ref}\right)}{\Delta z} + \hat{Q}_{conv_{pg}} + \hat{Q}_{conv_{wg}} \tag{2-8}$$

where the first two terms on the right hand side account for the enthalpy change due to gas flow, and the last two terms account for convection from the particles and from the wall, respectively.

The convection from the walls to the gas is calculated as

$$\hat{Q}_{conv_{wg}} = \frac{h_{lam}A_i}{V_{disk}}\left[T_{w,b} + T_{w,f} - 2T_g\right] \quad \textbf{(2-9)}$$

where h_{lam} is the heat transfer coefficient for laminar flow and was obtained from the formula presented in [46].

2.2.4 Radiative Heat Transfer

2.2.4.1 Incoming Solar Radiation

As mentioned in the previous section, the front section of the tube receives radiation directly from the secondary concentrator through the quartz window. Figure 6 shows the profile of the incoming heat flux through the window, obtained from ray tracing simulations performed by Lichty et al. using the program SOLTRACE [47]. Figure 6 shows that the heat flux depends mainly on the height (*y*), and it reaches a maximum at the center of the window (*y*=0). In order to account for this dependency, a correction factor was derived by fitting the average heat flux along the horizontal position (*x*) as a function of the vertical position (*y*). This correction factor resembles the form of a cosine function, with a maximum value of 1 at the center of the tube (z_{avg}=*L/2*). The correction factor has the following form:

$$\beta = \frac{D}{\sqrt{({}^{L}/_{2} - z_{avg})^2 + D^2}} \quad \textbf{(2-10)}$$

in which *D* is the coefficient that was fitted in order to minimize the sum squared of the error between the correction factor and the SOLTRACE simulation results, and was found to have a value of 0.0182 m. The average values of the heat flux and the fitted function are shown in Figure 7.

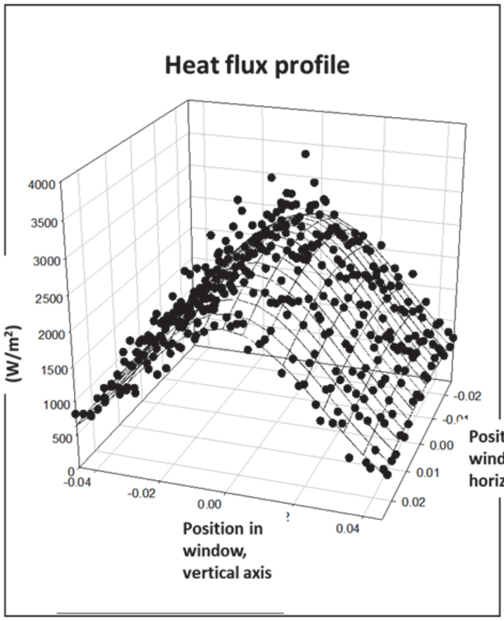


Figure 6. Heat flux profile incident in the reactor's window. Obtained from SOLTRACE simulations [47].

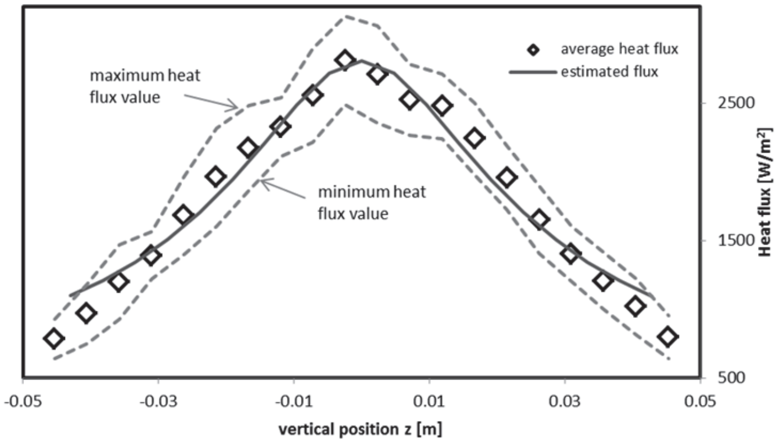


Figure 7. Average heat flux as a function of vertical position in the window

An average value was used for the solar power level input at each height. In order to calculate this average power level, the tube wall was divided into segments of constant arc length, and the angle at which the concentrated solar rays hit the tube was calculated for each segment. The projected area for each of the segments was calculated, and it was integrated with respect to the angle of the segment. The result of the integral was divided by the total surface area, in order to obtain the fraction of the maximum power level that equals the average power level. This calculation is summarized with the following formula:

$$\langle Q \rangle = \frac{\int_{-\pi/_2}^{\pi/_2} q_{solmax} Cos\theta \Delta zrd\theta}{\int_{-\pi/_2}^{\pi/_2} \Delta zrd\theta} = 0.6346\, q_{solmax} \qquad \textbf{(2-11)}$$

where $q_{solmax} = \frac{P_{solar}}{A_{win}}$

As a result, the term for the incoming solar irradiation was calculated using the following expression:

$$Q_{solar} = 0.885\beta\tau_{win}\langle Q \rangle A_o \qquad \textbf{(2-12)}$$

where 0.885 is a correction factor that accounts for the heat losses of the secondary concentrator.

The rays that do not hit the front wall of the tube in the first pass are reflected in the cavity and leave the cavity through the window without hitting the tube on their way out. This was demonstrated by simple simulations of the geometry. Thus, the back of the tube does not receive any direct solar radiation, only the radiation that it exchanges with the front wall and the particles.

2.2.4.2 Implementation of Volumetric Exchange Theory in a Simplified Model

The carbon black particles used in this study are very small in relation to the wavelength ($\xi = \pi d_p/\lambda_m$ < 0.3); therefore Rayleigh theory can be used to calculate the absorption and scattering coefficients [48]. For larger particle sizes (0.3 < ξ < 5) Mie theory should be used to calculate these coefficients. For particles with ξ > *5* the surface exchange model can be used [49]. Scattering can be neglected because the albedo is much smaller than 1 [49]. To implement a volumetric exchange approach using Rayleigh, the reactor was modeled as an enclosure with an absorbing and emitting medium in between, using an absorption coefficient for the medium. The enclosure is shown in Figure 8.

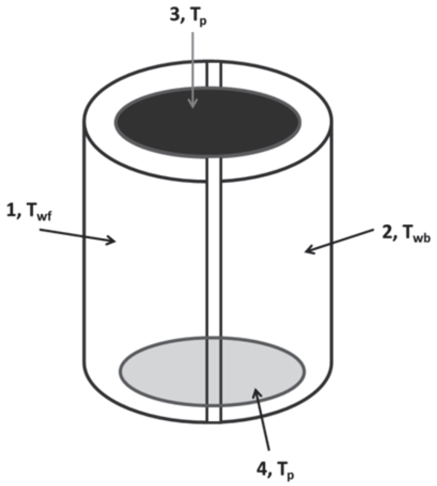


Figure 8. Diagram of the enclosure used to model radiation using a volumetric exchange approach.

Surfaces 1 and 2 (in Figure 8) represent the front and back walls, respectively. Surfaces 3 and 4 are assumed to have the same temperature as the particles in that disk (*T_p*); therefore, they do not participate in any radiation exchange with the particles. The energy transferred by radiation for surfaces 1 and 2 was calculated using the following equation for an enclosure [49]:

$$\sum_{j=1}^{N}\left(\frac{\delta_{kj}}{\varepsilon_{\lambda,j}}-F_{k-j}\frac{1-\varepsilon_{\lambda,j}}{\varepsilon_{\lambda,j}}\tau_{\lambda,k-j}\right)q_{\lambda,j}=\sum_{j=1}^{N}\left[\left(\delta_{kj}-F_{k-j}\tau_{\lambda,k-j}\right)e_{\lambda b,j}-F_{k-j}\alpha_{\lambda,k-j}e_{\lambda b,g}\right] \qquad \textbf{(2-13)}$$

where δ_{kj} is the Kronecker delta and $\delta_{kj}=1$ when k=j and $\delta_{kj}=0$ when k≠j; $\alpha_{\lambda,k\text{-}j}$ is the spectral geometric-mean absorption factor; and $\tau_{\lambda,k-j}$ is the spectral geometric transmission factor and $\tau_{\lambda,k-j}=1-\alpha_{\lambda,k-j}$.

All the surfaces are assumed to be black, so $\varepsilon_{\lambda,j}=1$, for all j.

Integrating over all wavelengths (using an average α_{k-j}, which is only a function of temperature and particle volumetric fraction) and writing the equations for surfaces 1 and 2 gives:

$$\hat{Q}_1=\frac{A_{w,i}\sigma}{V_w}\left[\left(1-f_{1-1}+\alpha_{Twf}f_{1-1}\right)T_{wf}^4-\alpha_{Tp}f_{1-1}T_p^4-f_{1-2}T_{wb}^4+\alpha_{Twb}f_{1-2}T_{wb}^4-\alpha_{Tp}f_{1-2}T_p^4-2f_{1-3}\left(T_3^4-2\alpha_{T3}T_3^4+2\alpha_{Tp}T_p^4\right)\right] \qquad \textbf{(2-14)}$$

$$\hat{Q}_2 = \frac{A_{w,i}\sigma}{V_w}\left[-f_{1-2}T_{wf}^4 + \alpha_{Twf}f_{1-2}T_{wf}^4 - \alpha_{Tp}f_{1-2}T_p^4 + (1 - f_{2-2} + \alpha_{Twb}f_{2-2})T_{wb}^4 - \alpha_{Tp}f_{2-2}T_p^4 - 2f_{2-3}(T_3^4 - 2\alpha_{T3}T_3^4 + 2\alpha_{Tp}T_p^4)\right] \quad \textbf{(2-15)}$$

Eqns. (2-14) and (2-15) include the radiation exchange between both walls and the radiation exchange between the particles and the walls. Eqn. (2-14) was used to replace the second and third term on the right hand side of Eqn. (2-2). In the same way, Eqn. (2-15) replaced the first two terms on the right hand side of Eqn. (2-3).

The radiation that the particles receive was calculated as:

$$\hat{Q}_{rad} = -\frac{V_w}{V_{disk}}\left(\hat{Q}_1 + \hat{Q}_2\right) \quad \textbf{(2-16)}$$

Eqn. (2-16) substituted Eqn. (2-6) in the energy balance for the particle phase.

The spectral geometric-mean absorption factor (*α_λ*) was calculated using the geometric-mean beam length approach. The mean beam length (*L_e*) for a cylinder of infinite height radiating to its concave bounding surface and corrected for finite optical thickness was considered to be *0.95d* [38]. Using this mean beam length, *α_λ* was calculated as:

$$\alpha_\lambda = 1 - \exp(-a_\lambda L_e) \quad \textbf{(2-17)}$$

where *a_λ* is the spectral absorption coefficient and was calculated using the optical constants for acetylene soot found in [50].

α_λ was used to calculate the average *α*, through the following expression:

$$\alpha = \frac{\int_\lambda \alpha_\lambda e_{\lambda,b}}{\sigma T^4} \quad \textbf{(2-18)}$$

where *$e_{\lambda,b}$* represents the blackbody spectral emission and was calculated using Planck's spectral distribution of emissive power [49].

Average values of *α* were obtained for different temperatures and particle volumetric fractions and were used to fit an expression that gives *α* as a function of temperature (*T*) and particle volumetric fraction (*X_{out}*). This expression was obtained through a stepwise linear regression and was found to be:

$$\alpha = -0.009352 + 20.68TX_{out} - 147.9T^2X_{out}^2 - 2506X_{out} + 0.00003T \quad \textbf{(2-19)}$$

These changes were implemented in the base model presented in section 2.3. Although a volumetric exchange radiation model represents a more complicated implementation, the model is computationally efficient and appropriate to be used to study the dynamics of the system.

2.2.5 Numerical Solution and Mesh Verification

The model consists of four simultaneous ordinary differential equations (ODE's) for each disk. These equations were solved numerically using the *ode23s* solver function in MATLAB® [51]. This solver was chosen because it can manage the stiffness of the system.

Although a lumped approach (single lump) would reduce the computational expense, the use of this method would be invalid. According to Lee et al. [52], the dispersive Peclet number (Pe) can be used as a measurement for axial dispersion, with a Pe approaching zero indicating that the system can be modeled by a CSTR (lumped model) and a Pe approaching infinity indicating that the system would be best represented by a plug-flow reactor (PFR). The Pe for this system ranges from 10 to 150, indicating that it is incorrect to model the system as a single CSTR.

The number of disks that would approach a continuous tube has been determined by performing simulations with an increasingly higher number of disks until the variations in the temperature profile became negligible. This occurs at approximately 65 disks, so this number of disks was selected for all simulations, requiring the simultaneous solution of 260 ODE's.

2.2.6 Results for Inert Particles

The dynamics of the system are not expected to change significantly with the addition of chemical reaction to the model. Even with a model that does not include chemical reaction, simulations can provide relevant information regarding the response time of the system to changes in the solar radiation input.

Results for the simulations at different power level inputs are presented in Figure 9. This figure shows that the dynamic response of the temperature changes with the power level (the temperature reaches steady state faster at 6 kW than at 2 kW), which implies that the system is nonlinear. This is an important characteristic to take into account when designing a control system, because the classic approach of fitting a linear model to experimental data might result in unstable closed-loop responses. Instead, a linearized scheduled strategy could be employed. In this strategy, a linear model is fitted for each operating point and the control algorithm chooses model parameters depending on the operating point. The linearized scheduled approach has been successfully implemented in the past for diverse nonlinear applications [53].

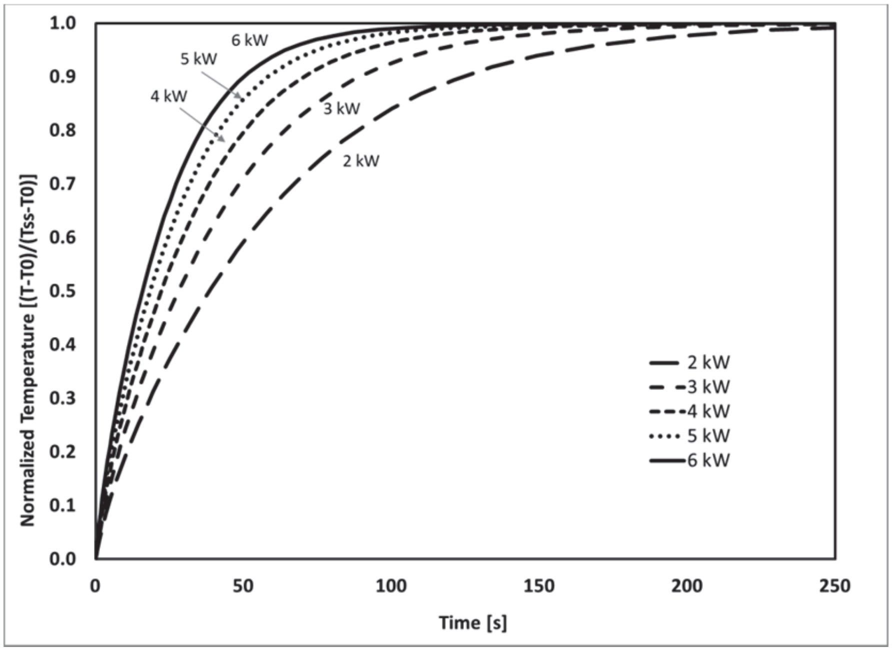


Figure 9. Normalized dynamic responses for the front wall temperature for different power levels.

2.3 Model Validation

Several experimental runs were performed at the HFSF at NREL [54]. The results were compared with model simulations in order to validate the model. At the 10-kW HFSF facility at NREL, sunlight is reflected by a heliostat that directs the light into a primary concentrator which delivers a beam into a secondary concentrator and then into the reactor. The power level is controlled by a shutter that opens a certain percentage depending on the readings of Normal Incidence Pyrheliometer (NIP), which is a measurement of the solar radiation at normal incidence, and on the power level setpoint that the user defines. A more detailed description of the experimental apparatus is presented in the following section.

2.3.1 Experimental Apparatus

The experimental unit consists of a single tube made of Hexoloy® SA Silicon Carbide 35 cm in length and with an internal diameter of 20 mm. This material was chosen for being a ceramic that can withstand high temperatures but that also has a high thermal conductivity, an important characteristic to diminish thermal shock [38]. The tube is enclosed in a reflective aluminum cavity that receives solar irradiation from a secondary concentrator through a 94 mm x 51 mm quartz window. A reflective cavity was preferred over an absorbing cavity, because the latter results in long start up times. The use of a reflective cavity allows for faster experimental runs.

A mixture of propylene glycol and water is used as cooling water for three different cooling zones in the reactor: the top, the bottom and the cavity walls. Argon is fed into the tubes at 1.0 SLPM. An automated shutter controls the amount of sunlight admitted to the secondary concentrator and is manipulated to apply ramps in the power level. The power levels that were tested were 2, 3, 4, 5 and 6 kW. The transient response of temperature was recorded at different points of the tube, using type B and type K thermocouples, placed as shown in Figure 10.

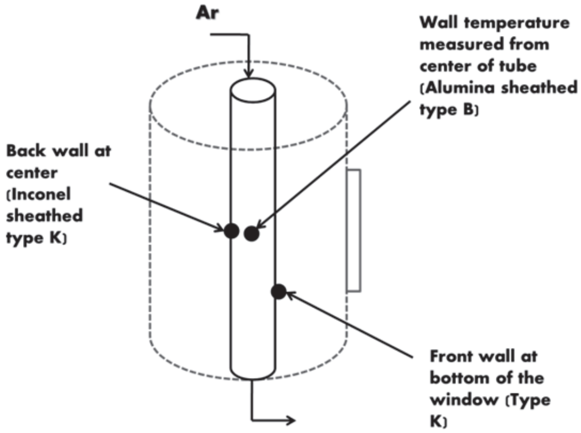

Ar
Wall temperature measured from center of tube (Alumina sheathed type B)
Back wall at center (Inconel sheathed type K)
Front wall at bottom of the window (Type K)

Figure 10. Position of the thermocouples during the experimental runs.

2.3.2 Comparison with the Model

The recorded transients were analyzed and compared to the results of simulations for the same power levels. Figure 11 shows a comparison between the experimental and simulation results at steady state for different power levels. This figure shows that the simulation results are in close agreement with the experimental values, especially at high power levels.

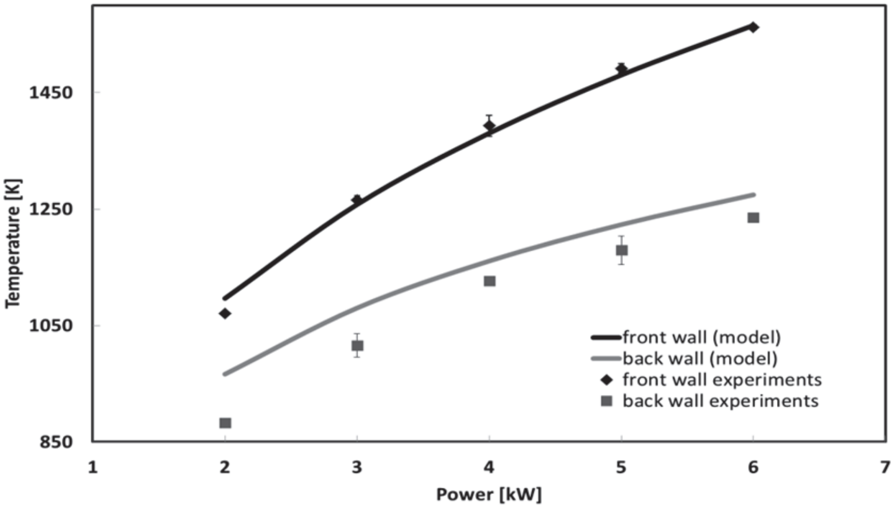


Figure 11. Comparison of experimental and simulation results for different power levels at steady state.

The comparison between the numerically-simulated and the experimentally-measured transient responses in temperature are shown in Figure 12. A transient response for the front wall temperature at the center of the tube after a 3-kW ramp was applied is shown in Figure 12.a. The steady state values show a good agreement. On the other hand, the dynamic responses show some differences. This disagreement can be explained by the location of the temperature measurement inside of the tube instead of directly at the tube wall. The thermocouple has to be heated by radiation from the tube wall until it reaches the temperature of the tube wall. This adds a mechanism of heat transfer (not included in the simulated results) to the measurements, causing a delay in the response of the system. In order to correct for this difference in the dynamics, a first-order transfer function was fitted between the values measured by the thermocouple and the temperature values predicted by the model. The first-order time constant was estimated to be 85 seconds.

In the case of the back wall temperature at the center of the tube (shown in Figure 12.b), the dynamic response shows a better agreement with the experimental data than that of the front wall temperature. In general, the difference obtained was about 50 K, which represents an error of less than 5%. This was judged to be an acceptable error for a model with low computational times.

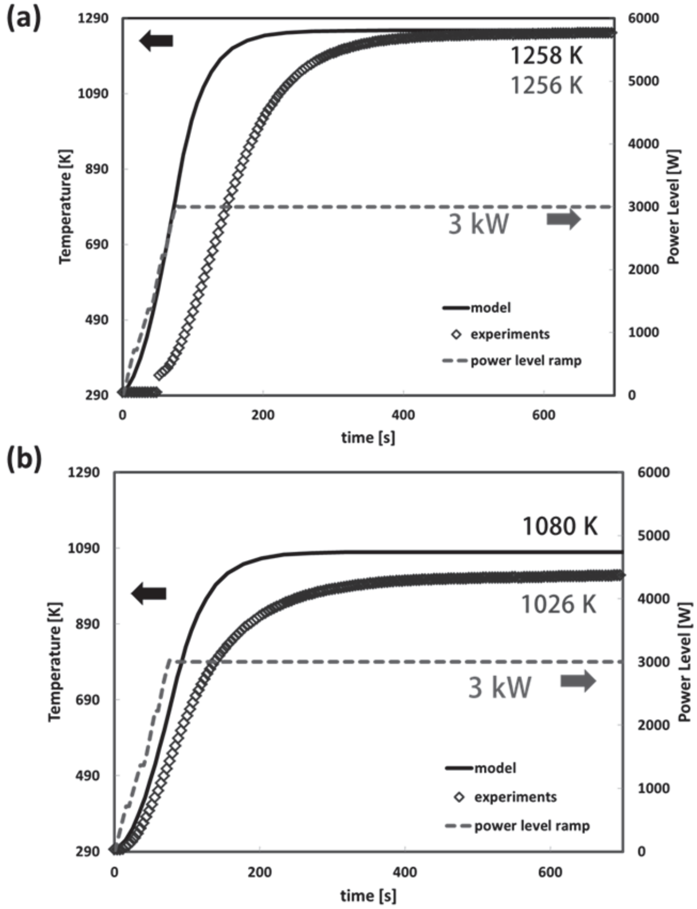


Figure 12. Comparison of transient responses between experimental data and results for simulations after a 3 kW ramp was applied. a) Front wall temperature at the center of the tube. b) Back wall temperature at the center of the tube

2.4 Simplified Dynamic Model for the Carbon Gasification Reaction

The mathematical model was applied to a system for the carbon steam-gasification reaction. Carbon gasification was chosen because of the wide availability of information on reaction kinetics in the literature. Carbon steam-gasification is represented by the following endothermic reaction:

$C_{(g)} + H_2O \rightarrow CO + H_2$

with $\Delta H_{rxn} = 131$ MJ/kmol [11]

Depending on the reaction conditions, water-gas shift can also occur:

$CO + H_2O \leftrightarrow CO_2 + H_2$

The reaction was modeled in each disk using a molar balance for each of the species,

$$\frac{dN_i}{dt} = \dot{n}_{i,in} - \frac{N_i v_{out}}{V_{disk}} + r_i V_{disk} \qquad \textbf{(2-20)}$$

Since all the carbon that reacts is converted into CO and CO_2, the rate of reaction for carbon can be represented by

$$r_c = -(r_{co} + r_{co2}) \qquad \textbf{(2-21)}$$

The rates of reaction for water, CO, CO_2 and H_2 were obtained from the literature and will be described in the following section.

In addition to the differential equations representing the molar balances for each of the species, an algebraic equation describing the volumetric flow rate is added to the model:

$$v_{out} = \dot{n}_g \frac{RT_g}{P} \qquad \textbf{(2-22)}$$

The model is then transformed into a differential-algebraic equations system (DAE), consisting of 9**n* ODE's and *n* algebraic equations, where *n* represents the number of disks used in the model. This system can no longer be solved by the MATLAB® function *ode23s*, so the function *ode15s* is used instead. Conversion is calculated from the results of the simulations as follows:

$$X = \frac{\dot{n}_{CO}(65) + \dot{n}_{CO_2}(65)}{\dot{n}_{C,in}} \qquad \textbf{(2-23)}$$

2.4.1 Experimental Set Up

Experiments that included carbon gasification were carried out at the HFSF at NREL. The results of these experimental runs were used to choose the set of kinetic parameters that best fit the data. A diagram of the experimental unit is shown in Figure 13. During the experimental runs, a brush feeder developed by Woodruff et al. [55] was used to feed Chevron Phillips SHAWINIGAN Black ® Acetylene Black (carbon black), with an average particle diameter of 40 nm. A flow of 1.0 SLPM of argon was used to entrain the particles. Steam was fed using a syringe pump at a rate of 135 μL/min. A non-dispersive infrared sensor (NDIR) was used to quantify the amount of CO and CO_2 produced by the reaction.

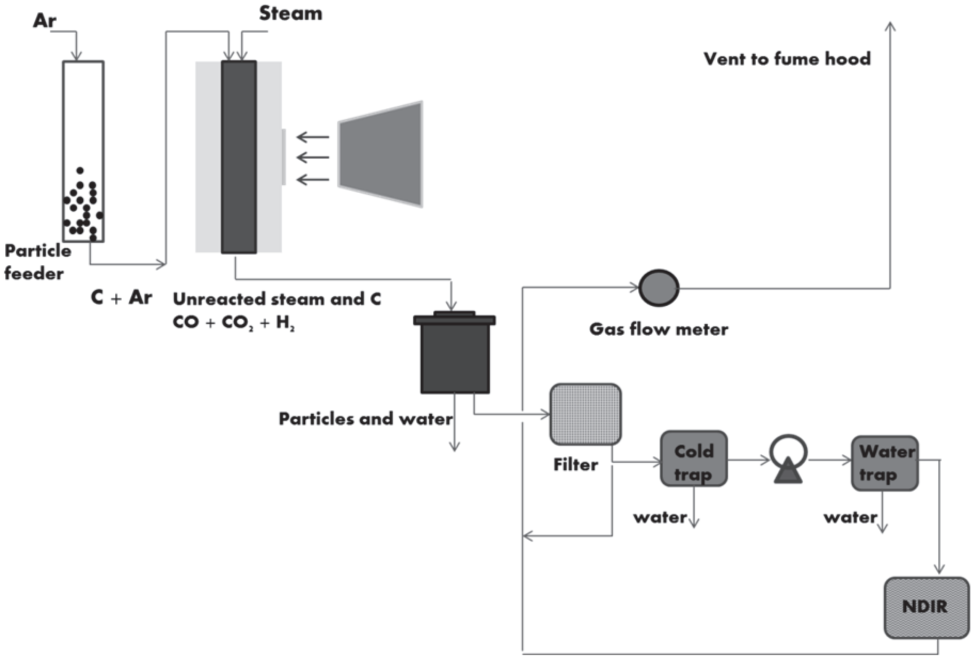


Figure 13. Diagram of the experimental set up for gasification runs at NREL.

Results from a typical experimental run are shown in Figure 14. The reactor tube was heated to the reaction temperature by opening the solar attenuator until the desired power level was reached. Once the temperature approximated steady state, steam flow was started in order to gasify any remaining carbon on the tube walls. When all the carbon black on the tube walls was gasified, particles were fed into the reactor at rates that ranged from 300 to 750 μg/s. The particles were fed into the reactor at a constant rate until the amount of CO and CO_2 measured by the NDIR reached a steady state.

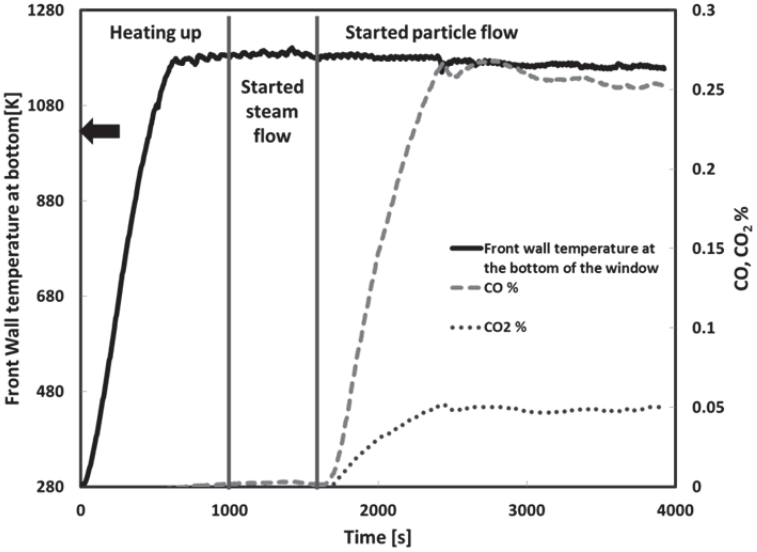


Figure 14. Typical gasification run at NREL.

From these data, the carbon conversion was calculated as

$$X = \frac{\dot{n}_{CO}+\dot{n}_{CO_2}}{\dot{n}_{C,in}} \tag{2-24}$$

Table 3 shows a summary of the experimental runs and the conditions at which they were carried out, as well as the conversion obtained for each of the runs. Run #3 resulted in a carbon conversion that was lower than expected for a power level of 5.5 kW. The same conditions were repeated in run #4, which provided completely different results. This suggests that either the experimental conditions were not the expected ones or there were significant errors in the measurement system. Hence, run #3 will not be considered for further analysis. Run #10 shows a conversion that is too high for this power level. Since there was some concern about the accuracy of the NDIR during this run, the NDIR calibration was tested using calibration gas after the run, and the measured values were higher than the actual concentrations of the calibration gas. Because of this, this run will also be neglected for further analysis.

Table 3. Description of the conditions for the experimental runs at NREL

Run #	Power level [kW]	Carbon in [moles/s]	Steam in [moles/s]	Ar flow [SLPM]	Conversion
1	5	5.53E-05	1.25E-04	1.5	0.075
2	4	6.25E-05	1.25E-04	1.5	0.050
3	5.5	6.65E-05	1.25E-04	1.5	0.003
4	5.5	4.72E-05	1.25E-04	1.5	0.083
5	3	4.81E-05	1.25E-04	1.5	0.001
6	5	4.97E-05	1.25E-04	1.5	0.065
7	5	3.25E-05	1.25E-04	1.5	0.092
8	5.5	2.57E-05	1.25E-04	1.5	0.158
9	5	6.16E-05	1.25E-04	1	0.186
10	4	4.61E-05	1.25E-04	1.5	0.123
11	4	3.28E-05	1.25E-04	1.5	0.070

The experimental conditions in Table 3 are the same conditions used in the simulations presented in the following section.

2.4.2 Kinetic Parameters

Various authors have studied the kinetics of carbon-steam gasification [56]–[62] . The kinetics obtained by some of these authors were tested in the model and compared to the experimental data. This comparison is shown in Figure 15. Linear regressions with 95%-confidence intervals showed that the set of kinetics obtained by Trommer et al. [56] represented the best fit for the experimental data.

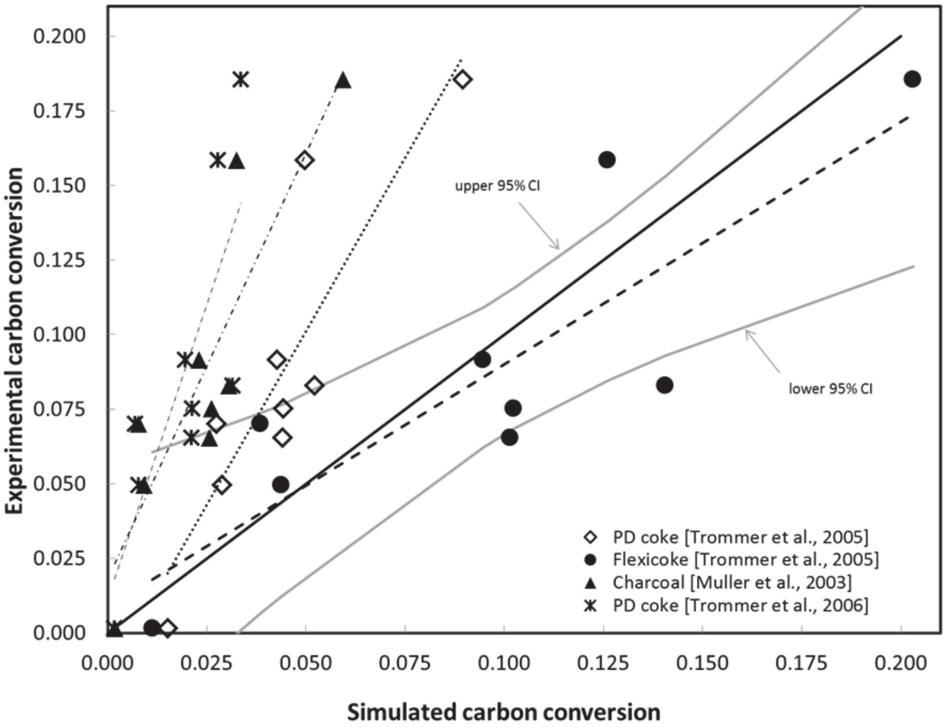


Figure 15. Comparison between the experimental and simulated results, using different sets of kinetics from literature.

Since the rate laws derived by Trommer et al. [56] were proportional to the mass of carbon in the mixture, the rate laws for all the species were multiplied by the carbon concentration in the disk, *Cc*, which is defined as the mass of carbon divided by the volume of disk. The rate laws are formulated as follows:

$$r_{h2o} = (-k_1 P_{H2O})C_c \quad \textbf{(2-25)}$$

$$r_{h2} = (k_1 P_{H2O})C_c \quad \textbf{(2-26)}$$

$$r_{co} = (k_1 P_{H2O} + 2k_2 P_{CO2} - 2k_1 k_3 P_{H2O} P_{CO})C_c \quad \textbf{(2-27)}$$

$$r_{co2} = (-k_2 P_{CO2} + k_1 k_3 P_{H2O} P_{CO})C_c \quad \textbf{(2-28)}$$

The partial pressures are calculated assuming ideal gases:

$$P_i = P_{tot} \frac{N_i}{N_{tot} - N_c} \quad \textbf{(2-29)}$$

The temperature dependence of each rate constant k_i is determined by the Arrhenius equation:

$$k_i(T) = k_0 exp\left(\frac{-E_A}{RT}\right) \quad \textbf{(2-30)}$$

The values for the rate constants and activation energies were corrected to account for the differences in surface area between the particles used by Trommer et al. and the particles used in the experimental runs, using the formula for the mass specific surface area presented in [56]:

$$a = 1\frac{m^2}{g}(^{300\ \mu m}/_{d_p}) \quad \textbf{(2-31)}$$

The values obtained for the kinetic parameters are shown in Table 4.

Table 4. Rate constant parameters used in simulations

Reaction constant [units]	EA [kJ/mol]	k0
k1 [mol/(g s Pa)]	158	3.56 x 10^{1}
k2 [mol/(g s Pa)]	66.5	7.65 x 10^{-3}
k3 [1/Pa]	-200	8.18 x 10^{-9}

2.4.3 Results and Discussion

Dynamic results for carbon conversion at different power levels are presented in Figure 16. This figure shows that conversion varies dramatically with the power level, reaching a conversion of 19% for a run at 6 kW, but staying below 1% for a run at 2 kW at the same flow rates and residence time. Similar to the results for the temperatures, the dynamics of the conversion response change with the power level, reaching steady state faster at higher power levels. Dynamic experimental data for conversions are not available due to limitations in the equipment used (the NDIR used does not deliver instantaneous readings of the compositions). Nevertheless, the steady state values were compared. The straight line in Figure 16 represents the experimental conversion obtained at the same conditions as the simulations at 5 kW. It is important to note that the low conversions obtained are a result of the very short residence times (~1 s) in the hot zone of the reactor. As the reactor is scaled-up to an industrial size and lengthened considerably, the residence time will be substantially increased and higher conversions will be achieved.

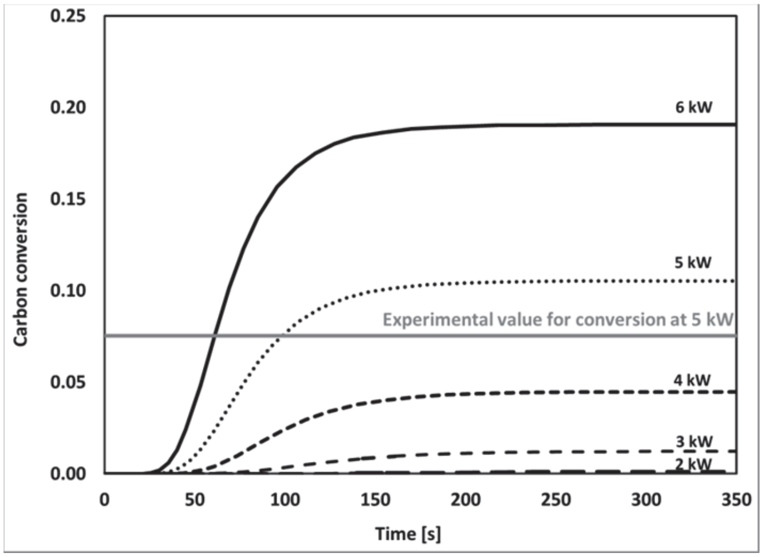


Figure 16. Dynamic results for conversion at different power levels.

The model was used to simulate conditions typical of a partly cloudy day. In this particular simulation, a step change in power level was applied to go from 5 kW to 2 kW, simulating the sudden presence of a cloud. Figure 17.a shows that the front wall temperature drops from 1420 K to 1070 K in about 150 seconds. The back wall temperature drops from 1260 K to 970 K also in a similar time period. On the other hand, the carbon conversion drops from 20% to 4% almost instantaneously (first order time constant of 12 seconds). These results emphasize the importance of having a control system that is capable of regulating the flow rates rapidly in response to the changes in solar irradiation. Without a control system, the drop in conversion will increase the concentration of the unreacted carbon and steam at the exit of the reactor, complicating the purification processes.

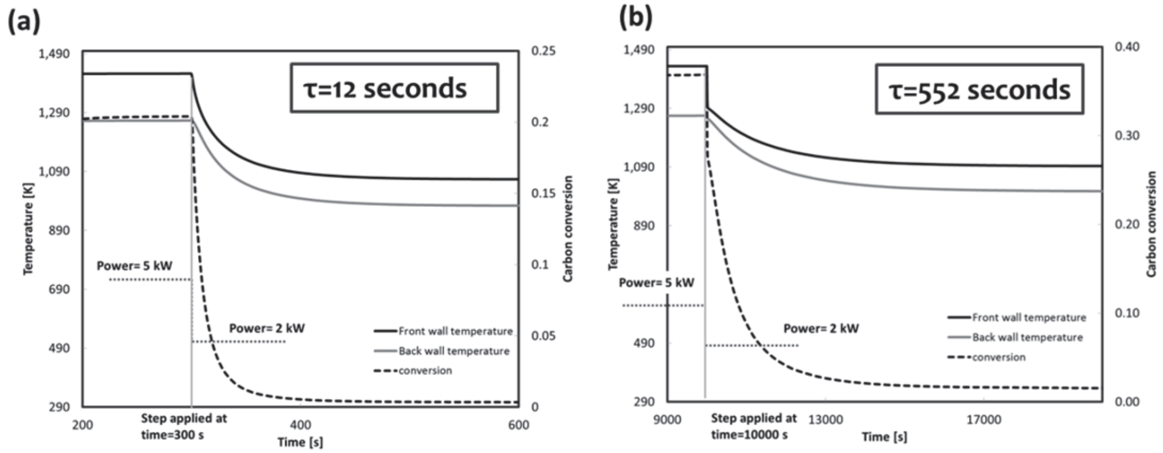


Figure 17. Simulation of a cloudy day. a) Reflective cavity, a step change in power level was applied at time=300 s. b) Absorbing cavity, a step change in power level was applied at time=10000 s.

A simulation with the same conditions was performed for a reactor with an absorbing cavity. This type of receiver is closer to what would be present in an industrial setting. Figure 17.b shows that the dynamics for an absorbing cavity are much slower than the dynamic response seen in the reflective cavity simulation. Note that the time scales in Figure 17.a and Figure 17.b have different orders of magnitude. The first-order time constant for the conversion in an absorbing cavity was estimated to be 550 seconds, which is 50 times slower than the response for a reflective cavity. Despite the slower dynamics, the temperatures and the conversion have a sharp drop when the step change is applied. By having a predictive control system that regulates the flow rates into the reactor in anticipation of the changes in solar irradiation, the drop in conversion would be mitigated. This will keep the productivity of the reactor as high as possible and will avoid problems in the separation processes downstream of the reactor.

2.5 Conclusions

A simplified dynamic model was developed for a solar-thermal transport-tube reactor. It was applied to a reactor for carbon gasification and validated with experimental data obtained at the HFSF at NREL. Simulations of this model provided results that are within 50 K of the temperatures obtained experimentally, which means that the error is less than 5% in terms of the steady state temperatures. In addition, the model presented is computationally efficient, allowing it to be used as a central element of a control algorithm that would make the operation of a solar-thermal reactor more feasible. Results from model simulations were shown both for a reactor with a reflective cavity and a reactor with an absorbing cavity. These results emphasize the need for a well-designed control system, which can be the determining factor for the feasibility of a solar-thermal reactor.

2.6 Nomenclature

a	Particle mass specific surface area (m^2/g)
a_λ	Spectral absorption coefficient for participating medium (m^{-1})
A	Area (m^2)
C_c	Carbon concentration in each disk (g/m^3)
C_p	Heat capacity (J/kg K)
D	Constant in correction factor for radiation hitting the tube (m)
d	Tube diameter (m)
d_p	Particle diameter (μm)
E_A	Activation energy (KJ/mol)
$e_{\lambda b}$	Blackbody spectral emission (W/sr m^2 μm)
f	Radiation view factor
h	Convection heat transfer coefficient (W/m^2 K)
k	Thermal conductivity (W/m K)
k_0	Kinetic frequency factor
k_j	Rate constant for reaction *j*

L	Length of tube in the hot zone (m)
L_e	Mean beam length (m)
$\dot{n}_i$	Molar flow rates of species *i* (mol/s)
N_i	Number of moles of species *i* (mol)
P	Partial pressure (Pa)
P_{tot}	Total pressure in the system (Pa)
P_{solar}	Solar power level after secondary concentrator (W)
q_{solmax}	Maximum solar heat flux coming through window (W/m^2)
$<Q>$	Average solar heat flux (W/m^2)
$\hat{Q}_{conv_{pg}}$	Volumetric heat source term, convection from particles to gas (W/m^3)
$\hat{Q}_{conv_{wg}}$	Volumetric heat source term, convection from tube walls to gas (W/m^3)
$\hat{Q}_{rad}$	Volumetric heat source term, radiation from tube walls to particles (W/m^3)
$\hat{Q}_{rxn}$	Volumetric heat source term, heat used for chemical reaction (W/m^3)
$\hat{Q}_{source}$	Volumetric heat source for tube walls (W/m^3)
r	Tube radius
r_i	Volumetric molar rate of production/consumption of species *i* (mol/m^3 s)
R	Ideal gas constant (J/mol K)
t	Time (s)
T	Temperature (K)
v	Gas velocity (m/s)
V	Volume (m^3)
X	Carbon conversion
X_{out}	Particle volumetric fraction
z_{avg}	Axial position of the center of each disk (m)

α	Geometric-mean absorption factor
β	Correction factor to account for dependency of heat flux on z.
δ	Kronecker delta
ΔH_{rxn}	Heat of reaction (MJ/kmol)
Δz	Length of each disk (m)
ε	Emissivity
λ	Wavelength
ξ	Size parameter
ρ	Density (kg/m^3)
υ	Volumetric gas flow rate (m^3/s)
σ	Stefan-Boltzmann constant (W/m^2 K^4)
τ	Transmittance
θ	Angle at which solar rays hit the tube wall (rad)

Subscripts:

amb	Ambient
b	Back
C	Carbon
conv	Convection
disk	Disk
f	Front
g	Gas
i	Counter for species
k	Counter for surfaces
j	Counter for surfaces
in	Inlet

lam	laminar flow
m	Medium
next	Following disk
p	Particles
o	Outer surface
out	Outlet
ref	Reference temperature
w	Tube wall
win	Window
λ	Spectral property

CHAPTER THREE

VISION-BASED SOLAR IRRADIATION PREDICTION

3.1 Introduction

The efforts to harvest solar energy as a renewable alternative to fossil fuels have increased considerably in the past decade [63]. Photovoltaics (PV) installations totaled 742 MW in the second quarter of 2012, four times more than the installations made in the first quarter of 2010 [64]. After the first half of 2012, there was a total of 546 MW of concentrating solar power (CSP) capacity operating in the United States [64], a 57% increase in the total CSP capacity in comparison to 2006 [63].

A major challenge that solar energy faces is its transient nature. The ability to forecast changes in solar radiation will bring economic benefits to any process that uses the sun as a main source of energy [65]. For instance, in order to integrate the electricity produced through PV to the smart grid successfully, accurate hourly and sub-hourly generation forecasts are required [66]. Solar radiation forecasts can also be crucial for the optimal operation of thermal storage systems of CSP plants [67]. As explained in Chapter One, the efficient operation of solar-thermal reactors requires a robust control system that can act in response to changes in solar radiation. One of the main hypotheses of this dissertation is that having a control system that performs the control actions before the changes in solar radiation happen will outperform a control system that mainly reacts to these changes. In order to test this hypothesis, short-term (one-to-five minutes ahead) solar radiation predictions are required.

Depending on the application, relevant time horizons for the solar radiation predictions can range from minutes to several hours. There have been multiple successful efforts to forecast solar radiation one hour ahead. Marquez and Coimbra developed a method that uses predicted meteorological variables from the US National Weather Service (NWS) as inputs to an

Artificial Neural Network model [68]. Mellit et al. [69] and Sfestos et al. [70] have also developed solar irradiance forecasts based on Artificial Neural Networks. Other authors use time series [71] or information from satellite images [72] to forecast solar radiation. Satellites have also been used to predict solar radiation 30 minutes ahead [73], but satellite forecasts are currently inadequate for shorter time horizons [74]. To be useful for control applications, solar forecasts need to be developed for time horizons shorter than 10 minutes.

Lopez-Martinez et al. developed a method that uses images of the sky to determine what fraction of a heliostat field will be covered by clouds and used this information to manipulate the position of the heliostats to protect the solar receiver from thermal shock [29]. Although they did not attempt to predict solar irradiation, they showed that images of the sky can be very useful for obtaining information on the available sunlight at very short time horizons.

Recently, several efforts have been developed to use Total Sky Imagers (TSI) in solar radiation forecasts. Marquez et al. used sky cover measurements from a TSI, in combination with current Global Horizontal Irradiance (GHI) measurements and measurements from infrared radiometers, to forecast GHI using Artificial Neural Networks [75]. Although their predictions were for the one-hour-ahead horizon, their results show that the use of TSI images can improve the forecasts' performance for intra-hour predictions.

Another study [74] analyzed images from a TSI to make a binary prediction (covered or uncovered) and used this prediction to make a simple estimation of the GHI, assuming that for the covered case the GHI is 40% of the clear sky GHI. They concluded that sky imagers are useful for forecasts that have a time horizon shorter than 25 min. They also found some limitations with the sky imager used and proposed combining the image analysis methods they developed with satellite- and numerically-based forecasts.

The most recent study was carried out by Marquez et al. [76]. In their study, they develop a methodology to process images from a TSI and predict the cloud fractions at horizons that ranged from 3 to 15 minutes. The cloud fraction predictions were used to forecast the

Direct Normal Irradiance (DNI) as a fraction of the maximum DNI for a clear day. DNI forecasts are more relevant for concentrated solar-thermal applications than GHI forecasts; consequently, the work by Marquez et al. is the most relevant to our study. They found that their method performs best for the five-minute-ahead horizon and anticipated that improving the image-based cloud classification algorithms and incorporating stochastic learning techniques into their forecasts will improve the prediction accuracy at short term horizons.

In this work, we propose a one-minute-ahead DNI forecast that combines information obtained using images of the sky with other atmospheric measurements and uses these data as inputs for a numerical classifier that outputs the forecasted DNI. This predictor will be integrated with a control algorithm; therefore, simplicity is a preferred characteristic.

3.2 Strategy

The general strategy for the proposed DNI forecast is depicted in Figure 17. Images of the sky are captured every minute using a TSI-880, developed by Yankee Environmental Systems [77]. The TSI has been part of the instrumentation of the Solar Radiation Research Laboratory (SRRL) since 2004 [30] and it consists of a charged coupled device (CCD) camera that points downwards to a heated hemispherical mirror that reflects the sky. The mirror is equipped with a sun-tracking shadow band, which protects the camera from overexposure to the sunlight. The TSI includes a built-in algorithm that pre-processes the images and determines the total cloud cover and opaque cloud cover percentages. Figure 18 shows an example of the raw images obtained with the TSI, and the output images after they have been analyzed with the TSI built-in algorithm. The images are obtained as *jpeg* files from NREL’s intranet, which is updated every minute. The images are further analyzed using a MATLAB® algorithm that determines the current sky scenario. After this, other relevant attributes - such as the speed and direction of the clouds - are determined using our own algorithms, described in detail in section 3.3.

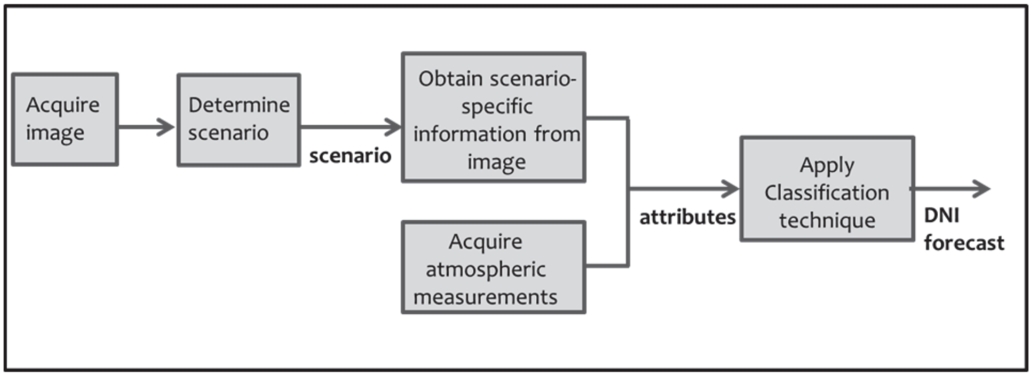

Acquire image
Determine scenario
scenario
Obtain scenario-specific information from image
Acquire atmospheric measurements
attributes
Apply Classification technique
DNI forecast

Figure 18. Proposed strategy for DNI forecasting

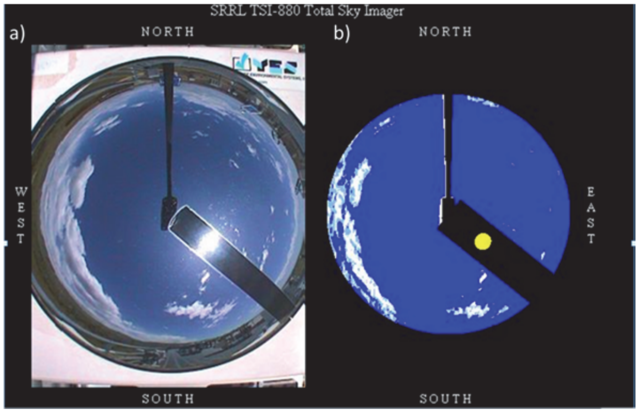


Figure 19. Example of images obtained with a YES® TSI-880: a) Raw image. b) Output image after built-in algorithm.

The information acquired from the images is combined with a set of atmospheric measurements, obtained from the SRRL, located at the National Renewable Energy Laboratory (NREL) in Golden, CO (Latitude 39.74° North, Longitude 105.18 ° West, Elevation 1829 m). The High Flux Solar Furnace (HFSF), where the control system will be implemented in the future, is located a few meters away from the SRRL, so the measurements taken there are directly applicable to the HFSF. A list of the selected atmospheric variables and the instrumentation used to measure them is shown in Table 5. These measurements are sampled every minute directly from the SRRL's [http://www.nrel.gov/midc/apps/display.pl?site=BMS], using a MATLAB® algorithm that was developed for this purpose.

Table 5. Selected atmospheric variables and instrumentation used.

Variable Name	Units	Instrument
Direct Normal Irradiance (DNI)	W/m^2	Eppley Normal Incidence Pyrheliometer [78]
DNI Derivative	W/m^2/min	Calculated using current and past DNI
Dry bulb temperature	°C	HMP45C-L Temperature and Relative Humidity Probe [79]
Wet bulb temperature	°C	Calculated using dry bulb temperature, dew point and station pressure
Dew point temperature	°C	Calculated using dry bulb temperature and relative humidity
Relative humidity	%	HMP45C-L Temperature and Relative Humidity Probe [79]
Total cloud cover	%	YES® TSI-880 [77]
Opaque cloud cover	%	YES® TSI-880 [77]
Avg. wind speed at 22 ft.	m/s	Measured by two NRG #40C anemometers [80]
Avg. wind speed at 42 ft.	m/s	Measured by two NRG #40C anemometers [80]
Peak wind speed at 22 ft.	m/s	Max. 2-sec anemometer reading during 1 min-interval [30]
Peak wind speed at 42 ft.	m/s	Max. 2-sec anemometer reading during 1 min-interval [30]
Avg. wind direction at 22 ft.	° from N	NRG #200P wind direction vane [81]
Avg. wind direction at 42 ft.	° from N	NRG #200P wind direction vane [81]
Station pressure	mBar	Vaisala Pressure transmitter PTB101B [82]
Zenith angle	°	Calculated using Solar Position Algorithm [83]
Azimuth angle	°	Calculated using Solar Position Algorithm [83]

The image-based and the atmospheric attributes are used as inputs to a classification algorithm, that was developed using the open source software WEKA [84]. The development and selection of the classification scheme will be described in more detail in section 3.4.

3.3 Scenario-Based Image Processing

The first step in the development of the image-based DNI forecast was to develop the procedures to obtain relevant information from the images of the sky. Current images are sampled every minute and analyzed according to the current sky scenario. Past images (one minute before) are also analyzed at each time step and used to determine some of the relevant attributes. The following sections describe the main components of the image-processing algorithm that was developed using MATLAB®'s Image Processing Toolbox [85].

3.3.1 Blue Pixels Fraction and Scenario Selection

The fraction of blue pixels in the area of interest in the image will be used to determine the current sky scenario. The image contains a protective shadow band and other objects that are not part of the sky; therefore, the first step is to transform the image into a useful binary representation of the area of interest. This is done by decomposing the image into the red, green and blue (RGB) planes. These planes are used to generate three masks, which are matrices that assign true or false values to the pixels that correspond to locations on the image. True values are assigned to the locations that fulfill certain threshold condition (e.g. value > 150), while false values are assigned to the rest of the locations.

Mask 1 represents all the pixels that are not blue, including the pixels corresponding to the clouds and the pixels corresponding to the shadow band and other objects. This mask was generated by applying a threshold in a way such that pixels with high values of red are assigned a true value and pixels with values below the threshold are false (high values of red mean no blue in original image). *Mask 2* represents all the objects that are not part of the sky, such as the shadow band region and the area outside of the TSI's dome. This mask was generated by applying an inverted threshold to the blue plane image, such that pixels with low values of blue are assigned a true value and pixels with values above the threshold are false. *Mask 3* was generated by the addition of *Mask 1* and *Mask 2*, and the false values in *Mask 3* represent the blue pixels in the area of interest. Figure 20 shows a representation of the three masks, where the white areas represent the true pixels and the black areas represent the false pixels.

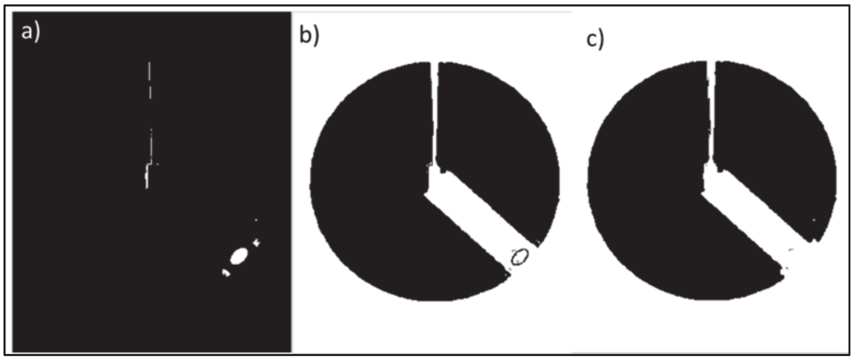


Figure 20. Output images after applying masks. a) Mask 1, b) Mask 2 , c) Mask 3

The fraction of blue pixels in the area of interest (R_b) was calculated using the following expression:

$$R_b = \frac{\#\ of\ false\ pixels\ in\ Mask\ 3}{\#\ of\ false\ pixels\ in\ Mask\ 2} \quad \textbf{(3-1)}$$

where R_b can take values between 0 and 1, with values approaching one representing clear sky and values closer to zero representing overcast conditions.

The value of R_b was used to determine the current sky scenario. Figure 21 depicts the four scenarios used in this classification and their main characteristics. The thresholds for sunny and cloudy classifications were determined by inspection of a subset of images. The blue pixels fraction was also used to identify errors (or images that could not be found) by setting an error threshold equal to the value of R_b for the cases when the image could not be found. Table 6 shows the values for the thresholds used in this classification.

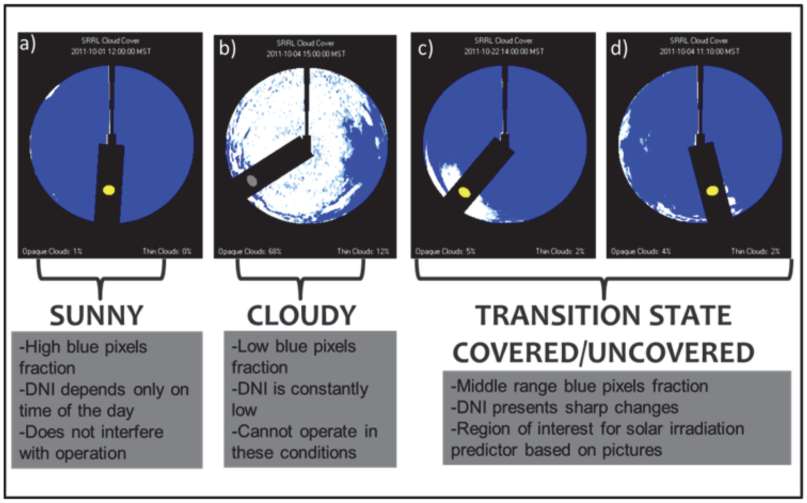


Figure 21. Four sky scenarios and main characteristics. a) Sunny scenario. b) Cloudy scenario. c) Transition scenario, covered case. d) Transition scenario, uncovered case.

Table 6. Thresholds used in scenario classification

Threshold	Blue pixels fraction
Sunny scenario	0.95
Cloudy scenario	0.27
Image not found	0.036145

3.3.2 Sunny Scenario

When the scenario is sunny, operation can continue without requiring any manipulation to the flow rates into the reactor except for normal feedback control via MPC. Without the presence of clouds, the DNI depends only on the time of the day and the month of the year, so it can be estimated as a function of the zenith and azimuth angles. Figure 22 shows the typical DNI profile of a sunny day. As this figure shows, short-term solar forecasts for perfectly sunny days will give accurate predictions by assuming that the future DNI will be equal to the current DNI.

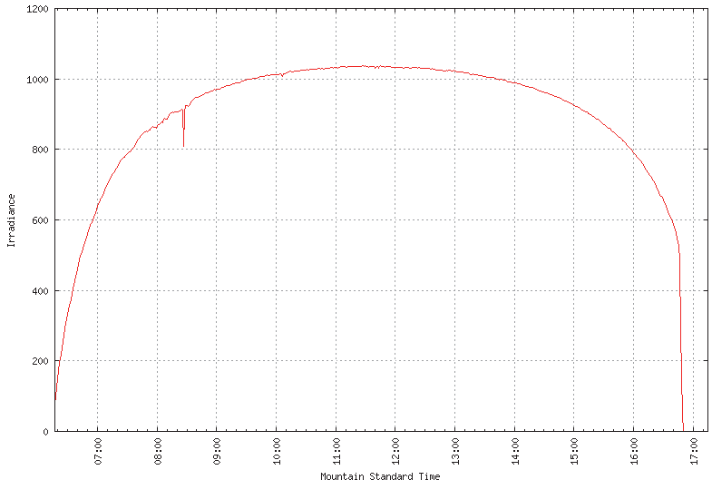


Figure 22. Typical DNI profile for a sunny day.

For the sunny scenario, the only relevant image-based attributes are the current and past blue pixels fraction (*$R_b(t)$* and *$R_b(t-1)$)*. The rest of the attributes used for the solar radiation forecast are the atmospheric variables listed in Table 5.

3.3.3 Cloudy Scenario

When the blue pixels fraction in the current image is below the threshold for cloudy conditions, the DNI is too low to operate the reactor. In these cases, the reactor would need to go into an idle state or transition into a shut-down state.

In addition to the image-based attributes used for the sunny scenario, the binary variable *C(t-1)* is included in the cloudy scenario data sets. This variable is assigned a value of one if the sun was covered in the previous image (one time step before) and a value of zero if it was not covered. The procedure to obtain the value of *C(t-1)* will be described in more detail in section 3.3.4.1.

3.3.4 Transition Scenario

When the blue pixels fraction falls between the thresholds for cloudy and sunny, the scenario is defined as a transition state. This scenario is the most relevant for control purposes, but it is also the one that presents more challenges for the solar radiation forecasts. In the transition state, the sun may be already covered by a small cloud, and this cloud might stay there or move away during the time scale of interest. In other cases, the sun might be uncovered, but clouds can be approaching and cover it in future time steps. Therefore, the transition scenario is further divided into two cases: covered and uncovered. Different image-based attributes are determined based on the current case. However, the initial processing steps are the same for both cases. These procedures will be described in section 3.3.4.1.

The calculation of the transition attributes was based on current and past images. The initial processing steps were performed in both images and the locations of the objects in the pictures were compared from one image to the next to determine the characteristics of the cloud movement. The methods used to determine these characteristics are described in sections 3.3.4.2 and 3.3.4.3, for the uncovered and covered cases, respectively.

3.3.4.1 Processing Images

After the images are acquired, the masks described in section 3.3.1 are applied. *Mask 1* was used for cloud identification, because the true pixels in this mask directly represent cloud pixels. Noise and small objects are removed from the image, to make the cloud identification more accurate. Figure 23 shows an example of an image after applying Mask 1(23.b) and after noise removal (23c).

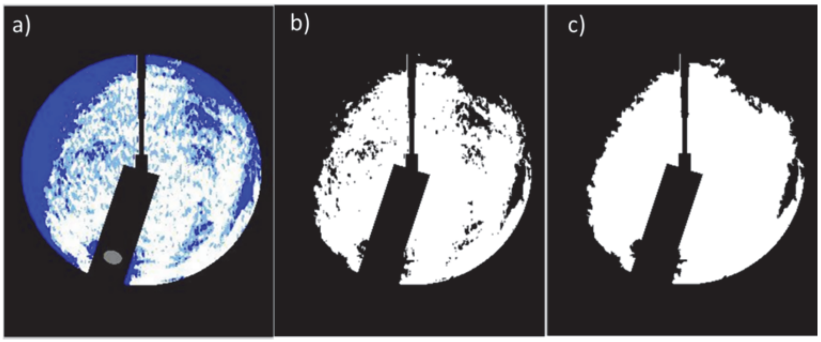


Figure 23. Output images after initial processing. a) Original image, b) Mask 1, c) Image after noise removal.

The sun position in the image is calculated as a function of the theoretical zenith (*α*) and azimuth angles:

$$x_{sun} = x_c + d * Sin(\beta) \tag{3-2}$$

$$y_{sun} = y_c - d * Cos(\beta) \tag{3-3}$$

where x_c and y_c are the coordinates of the center of the image, β is the azimuth angle in radians and *d* is the horizontal plane projection of the sun's position in the TSI's dome and was calculated through the following expression:

$$d = h * Cos(\theta) \tag{3-4}$$

where θ is the elevation angle in radians ($^{\pi}/_{2} - \alpha$), and *h* is the radius of the dome in the image in pixels.

The zenith and azimuth angles are a function of the time of the day and year and the latitude of the location. These angles were determined following the algorithm developed by the Earth System Research Lab (ESRL) at the National Oceanic and Atmospheric Administration (NOAA) [86], based on the Astronomical Algorithms by Jean Meeus [87].

Clouds were identified in the image by using the MATLAB® function *bwboundaries* to detect the boundaries of all the objects in the image. Once the objects are identified, certain properties of the objects, such as their area and centroid location, can be determined using specific functions included in MATLAB®'s Image Processing Toolbox.

The sun position can also be determined from the image. The ability to detect the sun as one of the objects in the image is an indication that the sun might be uncovered. When the sun is covered by a cloud it appears as grey in the original picture and its value is below the threshold when *Mask 1* is applied, assigning false pixel values to its location in the binary image. On the contrary, when the sun is uncovered it appears as yellow in the original image, assigning true pixel values to the binary image after *Mask 1* is applied. To identify the sun in the image, a metric similar to the one proposed in [88] is used. The metric used in this study can be calculated as:

$$Metric = \frac{2\pi A}{P^2} \qquad \textbf{(3-5)}$$

When an object's metric falls between 0.45 and 0.55, it can be safely assumed that that object corresponds to the sun and, the centroid of that object is assumed to be the sun's centroid, an important variable for the calculations that will be described in sections 3.3.4.2 and 3.3.4.3. If the algorithm cannot find any object that fulfills the metric, the sun is assumed to be covered, and the variable *C(t)* is assigned a value of one. If the algorithm is able to find the sun in the image, the measured value of the DNI is used to determine if the sun is covered or not, assuming that the sun is covered every time the DNI is below 900 W/m^2. When the DNI is equal or above 900 W/m^2, the sun is uncovered, and *C(t)* is assigned a value of zero. The choice of the DNI threshold for covered and uncovered cases was determined by inspection. The same procedure was followed in the image from a minute before to determine *C(t-1).*

3.3.4.2 Calculating Attributes for the Uncovered Case

For the uncovered case, relevant image-based attributes were the speed, the direction and the distance-to-the-sun of the closest approaching cloud, and the time that it will take for the closest approaching cloud to cover the sun. Note that the calculations are specific to clouds that are approaching the sun, so first it is necessary to determine if a cloud is approaching the sun or not, i.e. if the cloud will cover the sun in the future if it keeps moving in its current direction.

The first step is to locate the cloud centroid that is closest to the sun. To do this, the distance from the centroid of each object to the centroid of the sun is calculated for all of the objects in the image, using the equation for the distance between two points:

$$D_{O-S,i} = \sqrt{(x_{O_i} - x_S)^2 + (y_{O_i} - y_S)^2} \qquad \textbf{(3-6)}$$

The minimum distance and its corresponding index are located using a minimum search. The index is used to access the closest centroid from the object's array. If more than two objects are detected in the image, the second and third closest centroids are also identified.

Once the closest centroid is identified, the distance from the sun to each of the points in the boundary of the corresponding object is calculated using Eqn. (3-6). The minimum distance and the index corresponding to the minimum distance are determined through a minimum search. This process is repeated for the objects corresponding to the second and third closest centroids, if applicable.

Depending on the size of the cloud, the closest border might not belong to the object with the closest centroid. Since the most relevant parameter is the distance from the sun to the closest border, the object with the closest border is defined as the closest cloud. The objects are ranked in ascending order with respect to the distance from their closest border to the sun.

The speed and direction of movement of the clouds were determined by frame to frame comparison from the previous image (*J*) to the current image (*I*). For this comparison to be possible, it is necessary to locate the centroid in image *J* that corresponds to the closest object to the sun in image *I*. This is done by finding the minimum distance from each of the centroids in image *J* to the centroid corresponding to the closest cloud in image *I*. This process is repeated for the second and third closest clouds in image *I*.

The distance from the sun to each of the points in the border of the closest centroid in *J* is calculated using Eqn. (3-6). The minimum distance corresponds to the closest border in image *J*. The speed and direction of the cloud are calculated by comparing the displacement of the closest border from its location in image *J* to its location in image *I*.

The following expression was used to calculate the cloud speed (*S*) in pixels/min:

$$S = \frac{\sqrt{(x_{cb_I} - x_{cb_J})^2 + (y_{cb_I} - y_{cb_J})^2}}{\Delta t} \tag{3-7}$$

The cloud direction (in radians) was determined using:

$$Direction = Tan^{-1}\left(\frac{y_{cb_I} - y_{cb_J}}{(x_{cb_I} - x_{cb_J}}\right) \tag{3-8}$$

After the speed and direction of the closest cloud are calculated, it is necessary to determine if the cloud will cover the sun at future time steps. The technique developed to do this is described in section 3.3.4.2.1. If it is determined that the cloud will not cover the sun, the speed and direction of the second closest cloud are calculated using Eqns. (3-7) and (3-8). The direction of the second closest cloud is used to determine if this cloud will cover the sun. If it is determined that it will not cover the sun, the same procedure is repeated for the third closest cloud. If it is determined that the third closest cloud will not cover the sun, it is assumed that the sun will not be covered by clouds in the relevant time horizon. In this case, the image-based attributes are assigned artificial values that will be identified by the prediction algorithm. The speed is assigned a value of 0.5 pixels/min, which is an arbitrary value that will not produce errors when is used to calculate the time for cover, and the direction is assigned a value of zero. The distance from the closest approaching cloud to the sun is given a value of 255 pixels, which corresponds to the diameter of the circle in the image. The decision algorithm explained in this paragraph is represented in Figure 24.

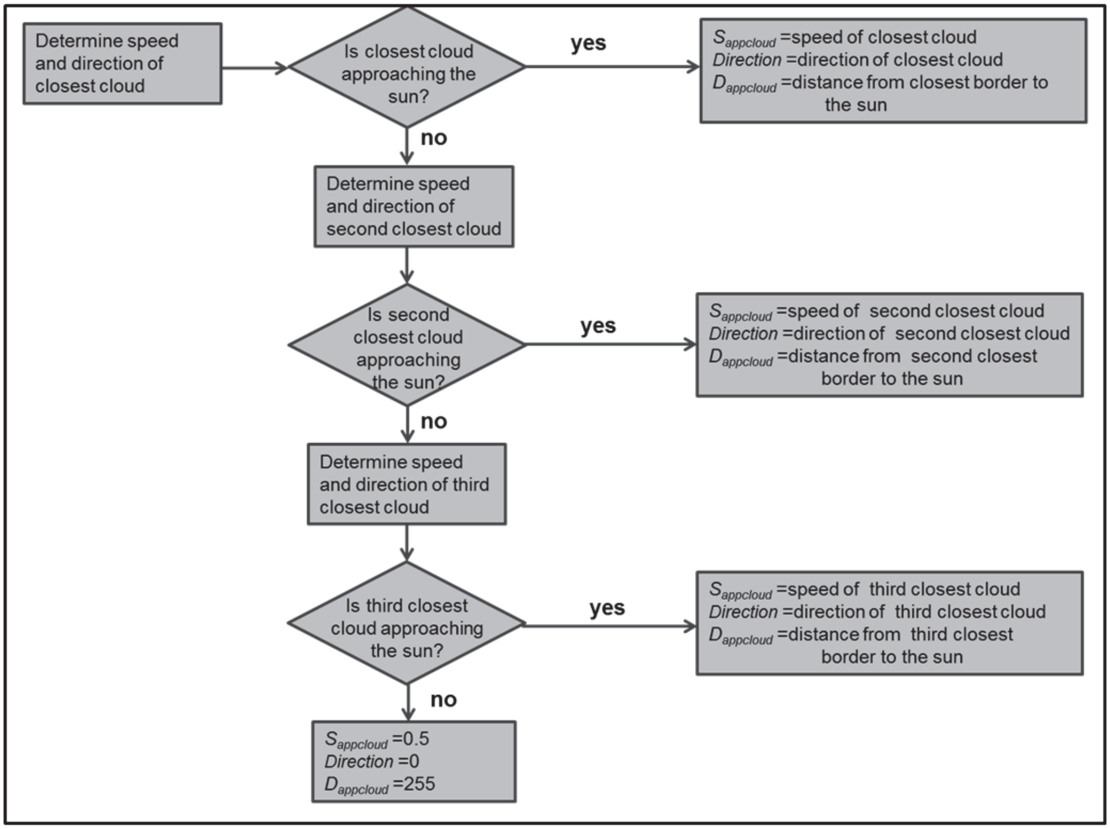

Determine speed and direction of closest cloud
Is closest cloud approaching the sun?
yes
$S_{appcloud}$ =speed of closest cloud
Direction =direction of closest cloud
$D_{appcloud}$ =distance from closest border to the sun
no
Determine speed and direction of second closest cloud
Is second closest cloud approaching the sun?
yes
$S_{appcloud}$ =speed of second closest cloud
Direction =direction of second closest cloud
$D_{appcloud}$ =distance from second closest border to the sun
no
Determine speed and direction of third closest cloud
Is third closest cloud approaching the sun?
yes
$S_{appcloud}$ =speed of third closest cloud
Direction =direction of third closest cloud
$D_{appcloud}$ =distance from third closest border to the sun
no
$S_{appcloud}$ =0.5
Direction =0
$D_{appcloud}$ =255

Figure 24. Decision algorithm to obtain attributes of closest approaching cloud.

The time (in minutes) that it will take for the closest approaching cloud to cover the sun is calculated as:

$$t_{cover} = \frac{D_{app\ cloud}}{S_{app\ cloud}} \tag{3-9}$$

where $D_{app\ cloud}$ is the distance from the sun to the closest approaching border. $D_{appcloud}$ is defined as the minimum sun-to-border distance that fulfills the decision algorithm presented in Figure 24.

If the cloud speed is estimated to be equal to zero (the closest border has not changed location from image *J* to *I*), the time for cover will go to infinity. In order to avoid this, the values of the cloud speed and direction at time *t-1* are used every time that $S_{appcloud}(t)=0$.

The calculations presented in this section are only valid for the cases where the closest border of the cloud is not located inside of the shadow band. For the instances in which the closest borders are inside or at the border of the shadow band, see section 3.3.4.2.2.

3.3.4.2.1 Determining if a cloud will cover the sun

The approach used to determine if a cloud will cover the sun is based on the assumption that all the points in a cloud object follow the same motion vector and that the direction of this vector remains constant at future time steps. This might seem like a big assumption, but it follows the same methodology that an observer of the clouds would take. To determine if a cloud will cover the sun, an observer would look at the image and follow what he considers to be the cloud path, automatically assuming that the entire cloud keeps moving in the same direction. Then, the observer would try to determine if the cloud's path is crossing the sun's path in the foreseeable future. We developed an algorithm that imitates this process and programmatically determines if a cloud will cover the sun.

First, straight line projections in the direction of movement are calculated for each of the border points in the closest cloud. These lines are extended towards the y-axis and their intercepts are determined, as shown in Figure 25. The maximum ($y_{a,0}$) and minimum ($y_{b,0}$) intercepts correspond to the edges of the cloud. If the straight lines formed by the maximum and minimum intercept (lines *a* and *b*, respectively) are extended towards the sun, it is possible to determine if the cloud's path will encounter the sun. If the sun is located between lines *a* and *b*, we can predict that the cloud will cover the sun. If the sun is located outside the area formed between the two lines, it can be assumed that the cloud is not approaching the sun. As Figure 25 shows, this can be easily determined by observing the image. The challenge is to develop an algorithm that can programmatically determine if the sun is located inside or outside the cloud's path.

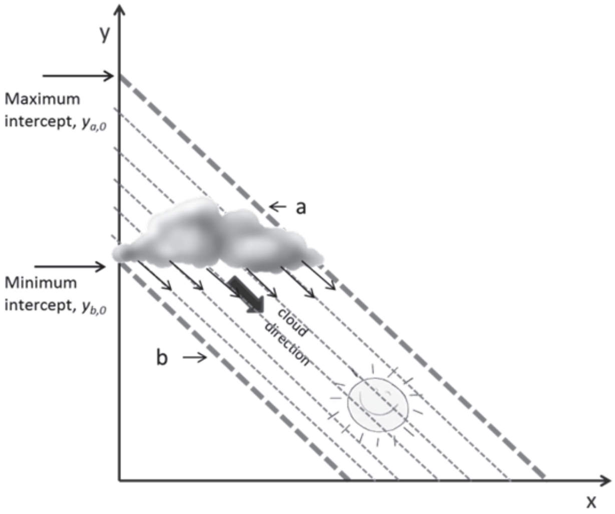


Figure 25. Straight line projections in the direction of movement of the cloud, starting at each of the points at the border of the cloud. The projections were used to find the maximum (ya,0) and minimum y-axis intercepts (yb,0).

Since all the lines have the same slope, the perpendicular distance between each pair of lines remains constant at any location. The perpendicular distance between lines *a* and *b* (D_{ab}) can be calculated using the following expression:

$$D_{ab} = \frac{|y_{a,0} - y_{b,0}|}{\sqrt{m^2+1}} \quad \textbf{(3-10)}$$

where *m* is the slope of the lines and it was determined using the cloud direction angle:

$$m = Tan(Direction) \quad \textbf{(3-11)}$$

The distances from the sun centroid to line *a* (D_a) and from the sun centroid to line *b* (D_b) were determined using Equations 3-12 and 3-13:

$$D_a = \frac{|y_s - m * x_s - y_{a,0}|}{\sqrt{m^2+1}} \quad \textbf{(3-12)}$$

$$D_b = \frac{|y_s - m * x_s - y_{b,0}|}{\sqrt{m^2+1}} \quad \textbf{(3-13)}$$

As Figure 26.a shows, if the sun is between lines *a* and *b,* the sum of D_a and D_b will be equal or less than D_{ab}. On the other hand, if the sun is outside the cloud's path, the magnitude of the sum of D_a and D_b will be larger than D_{ab}. This is shown in Figure 26.b. This criterion provides a programmatic methodology to determine if the clouds are approaching the sun.

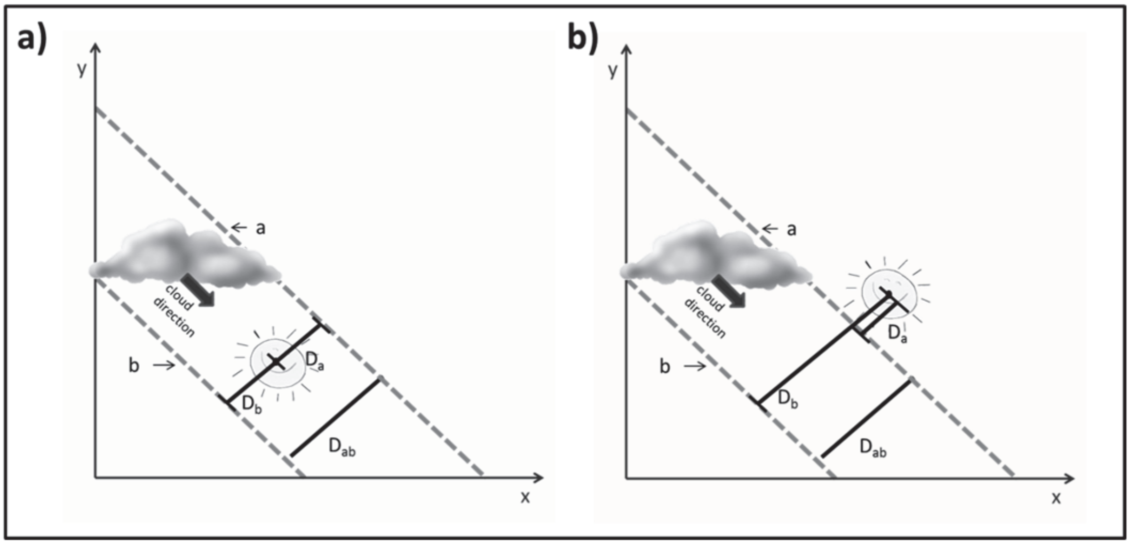


Figure 26. Illustration of the procedure used to determine if the cloud is approaching the sun. a) The cloud is approaching the sun. b) The cloud is not approaching the sun.

3.3.4.2.2 Dealing with the shadow band

The location of the shadow band presents a challenge for short term DNI forecasts, because it obstructs the region surrounding the sun and it makes it difficult to determine if there are clouds present in that region. For the uncovered case, the moment when the clouds will cover the sun becomes difficult to predict. These challenges have also been encountered by other authors [76], who in response focused on forecasts with horizons larger than three minutes. Besides complicating the analysis of the clouds that are approaching the sun, the false pixels in the shadow band region will divide any large cloud covering the sun into two different objects, complicating the calculations of the attributes for the covered case.

The shadow band will lead to inaccurate predictions if no actions are taken to deal with its presence. If the actual closest border of a cloud is located in the region inside the shadow band, the methodology described in section 3.3.4.2.1 will determine that the closest border is located at the edge of the shadow band. At the next time step, the location of the closest border (according to the image analysis algorithm) will remain the same, even if the actual closest border may have moved in the direction of the sun. The image processing algorithm will conclude that the cloud is not moving and, therefore, not approaching the sun.

To avoid erroneous predictions related to the shadow band region, the image processing algorithm uses a different set of equations to determine the attributes when operating in this region. A binary variable, $I(t)$, was assigned a value of one for the cases when the closest border of the cloud is inside the shadow band region, and a value of zero when the cloud is outside that region. A set of rules has been developed to determine if the closest border is inside or outside the shadow band region. The following cases assign a value of one to $I(t)$:

1. The sun is covered by a cloud ($C(t)=1$).
2. The distance from the closest border to the sun is less or equal than the distance from the edge of the shadow band to the sun.
3. The speed of the closest approaching cloud was estimated to be equal to zero.
4. $I(t)$ or $I(t-1)$ have a value of one and the previous value of the time for cover ($t_{cover}(t-1)$) is less than one minute. This represents a case when the sun becomes covered.
5. $C(t-1)=1$ and the previous time for uncover ($t_{uncover}(t-1)$) is larger than 200 minutes. In this case the sun remains covered.

Every time a cloud enters or leaves the shadow band region, i.e. $I(t) \neq I(t-1)$, the speed and direction of the cloud are recorded. A time stamp of this moment is also recorded and is used in further calculations as it will be shown below. Instead of using Equations (3-7) and (3-8) to determine the speed and direction of the cloud, these variables are assumed to remain constant while the cloud is inside the shadow band region. The distance of the closest border to the sun is estimated using the following expression:

$$D_{app\ cloud} = D_{SB-S} - \Delta t_{SB} S_{app\ cloud} \qquad \textbf{(3-14)}$$

where $D_{SB\text{-}S}$ is the distance from the edge of the shadow band to the sun, and, after image inspections, it was determined to be equal to 21 pixels. Δt_{SB} is the time that has passed since the border of the cloud entered the shadow band region, in minutes. $S_{appcloud}$ is the speed of the closest approaching cloud, which is equal to the speed that the cloud had when it entered the shadow band region.

The use of Eqn. (3-14) to estimate the distance of the closest approaching border to the sun provides a more accurate estimate of the time for cover in Eqn. (3-9). However, due to the constant speed assumption, Eqn. (3-14) might estimate a negative value for $D_{app\ cloud}$, assuming that the border of the cloud should have covered the sun already, when in reality it has not. In these cases, $D_{app\ cloud}$ is assigned a value of zero.

3.3.4.3 Calculating Attributes for the Covered Case

Just as in the uncovered case, the cloud speed and direction are relevant attributes for the covered case. However, the methods used to estimate the values of these variables when the sun is covered are slightly different than the ones used in the uncovered case. Instead of using the border of the cloud that is closest to the sun, the calculations for the covered case are based on the farthest border of the closest cloud. As an observer would conclude, we can assume that the sun will remain covered until the farthest border of the cloud has passed through. The calculation of the time that it will take for the sun to be uncovered presents more challenges than the calculation of the time for cover.

The location of the shadow band also presents some challenges for the covered case. As it can be seen in Figure 27a, the shadow band splits a single cloud that is covering the sun into two objects, complicating the calculations of the time for the sun to be uncovered. This is not always the case, as a cloud can be covering the sun without being on both sides of the shadow band, as shown in Figure 27.b. Although the case shown in Figure 27.a. occurs more often than the case shown in Figure 27.b, it is important to determine if this shadow-band-related problem is occurring in the image that is being analyzed. To do this, we developed the algorithm described in section 3.3.4.3.1. This algorithm also corrects for errors in the closest-object definitions, ensuring that the clouds defined as "closest objects" are actually the ones that are closer to the sun. After the closest-object definitions have been corrected, the methods described in section 3.3.4.3.2 are used to calculate the speed and direction of the cloud. Finally, the methodology used to determine the time for the sun to be uncovered is explained in section 3.3.4.3.3.

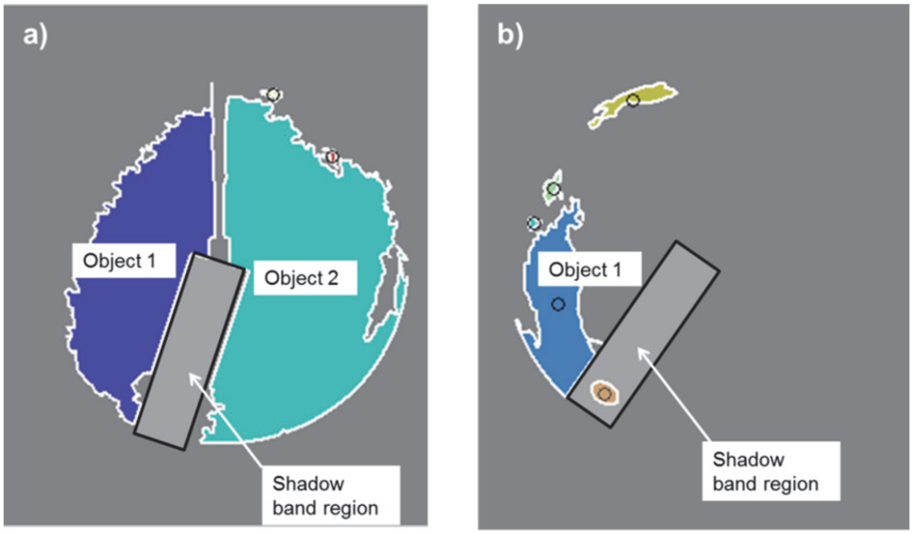


Figure 27. Example of problems caused by shadow band for covered case. a) Shadow band splits cloud into two objects. b) Cloud is one side of the shadow band, so shadow band does not split cloud into two objects.

3.3.4.3.1 Determining if a cloud is divided by the shadow band

To identify if the shadow band is dividing a single cloud into two objects, the first step is to determine if the two closest borders are located on the same side of the sun. If a cloud is split by the shadow band, the two closest borders would be located on different sides of the sun, more specifically at the edges of the shadow band. This is shown in Figure 28. If we trace an imaginary line, *c*, between the center of the image and the centroid of the sun, we can use Eqn. (3-15) to determine if both points are on the same side of the line:

$$Det_1 = (x_S - x_C)\left(y_{CB_1} - y_C\right) - (y_S - y_C)\left(x_{CB_1} - x_C\right) \tag{3-15}$$

where Det_1 is the value of a determinant that can be used to identify if both points are on the same side of line *c*. x_s, y_s are the coordinates of the sun and x_c, y_c are the coordinates at the center of the image. The subscript *1* refers to the closest border. Eqn. (3-15) was also applied to calculate Det_2, the value of the determinant for the second closest border (CB_2).

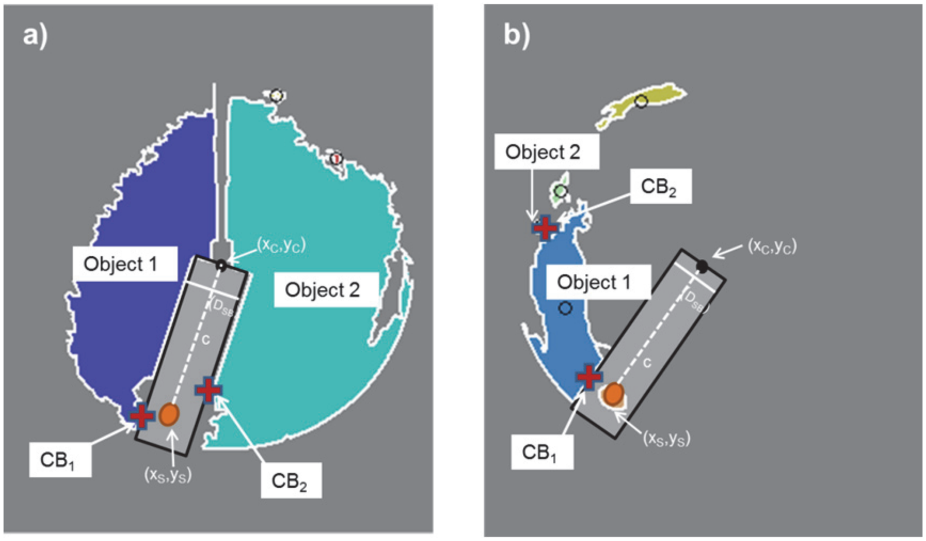


Figure 28. Determining if a cloud is divided by the shadow band. a) The shadow band is dividing the cloud into two objects. b) The shadow band is not diving the cloud.

If both points (CB_1 and CB_2) are located on the same side of the line, the value of Det_1 and Det_2 would have the same sign. In other words, for CB_1 and CB_2 to be located on different sides of *c* the product Det_1*Det_2 has to be negative.

Once it is established that the closest borders are located on different sides of the sun, the distance between them is calculated using the formula for the distance between two points. If the shadow band was dividing a cloud into two objects, CB_1 and CB_2 would be located at the edges of the shadow band. Therefore, we can use the distance between CB_1 and CB_2 to detect if these points are located on the edges of the shadow band. If the distance between CB_1 and CB_2 is equal to the distance between the two edges of the shadow band, it can be concluded that the shadow band is splitting the cloud into two objects.

In some cases, the distance between CB_1 and CB_2 can be larger than the distance between the two edges, but the cloud can still be affected by the shadow band (Figure 29.a). In other cases, CB_1 and CB_2 are determined to be located on the same side of line c, when in reality there is a cloud being divided by the shadow band (Figure 29.b). These problems occur when large clouds are present in the image. If a cloud object has a large area, its centroid may be farther from the sun than the centroids of other smaller objects, even if its borders are closer to the sun. Since the cloud identification algorithm searches for the three closest centroids to the sun before looking for the closest borders, borders that are close to the sun but belong to large clouds can be completely missed in the process. To correct for these errors, if it has been concluded that the shadow band is not dividing a cloud in two, the algorithm performs a "double-check" and searches for the closest border to the sun in the object with the biggest area. Then, it uses a determinant to identify if the closest border of the large cloud (CB_L) and CB_1 or CB_2 are located on different sides of line *c.* If it is found that CB_L is located on the other side of the shadow band from either CB_1 or CB_2, then CB_L becomes now the new closest border (or second closest, depending on which border is on the other side of the shadow band). This automatically defines the large cloud as the closest object for all the subsequent calculations. The algorithm also concludes that the shadow band is dividing a cloud into two objects, a decision that will determine the method to be used in the succeeding calculations.

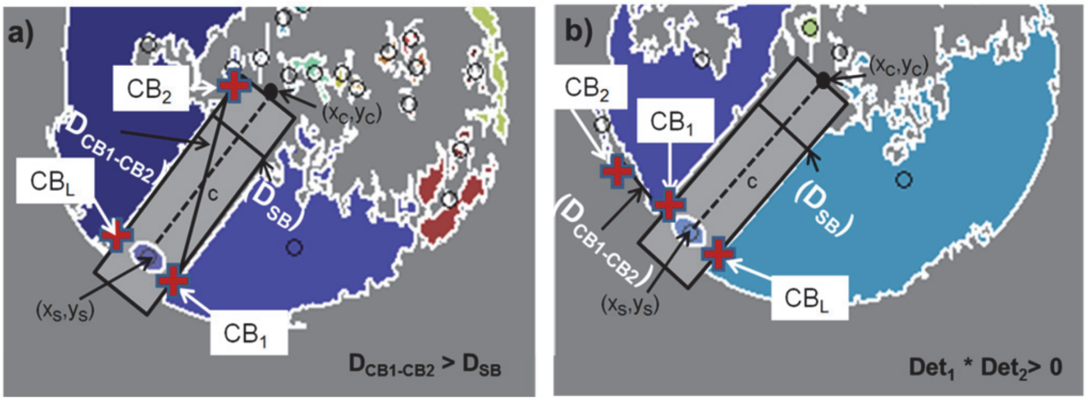


Figure 29. Example of problems caused by shadow band for covered case. a) Shadow band splits cloud into two objects. b) Cloud is one side of the shadow band, so shadow band does not split cloud into two objects.

3.3.4.3.2 Calculating the speed and direction a cloud covering the sun

The methods used to determine the speed and direction of the cloud are similar to the ones presented in section 3.3.4.2. The distances from the sun to each of the points on the border of the closest object are calculated using Eqn. (3-6). The main difference with respect to the uncovered case is that, in the covered case, the algorithm looks for the maximum distance, because the algorithm is looking for the farthest border in the closest cloud. Once the farthest border of the closest cloud is identified in images *I* and *J*, the displacement (*s*) of the farthest point from image *J* to image *I* is calculated using the following expression:

$$s = \sqrt{(x_{fb_I} - x_{fb_J})^2 + (y_{fb_I} - y_{fb_J})^2} \quad \textbf{(3-16)}$$

Sometimes, the object identified as closest object in image *J* is not the same object as the one identified as closest object in image *I*. Using farthest borders that correspond to different objects will cause Eqn. (3-16) to output unreasonable displacement values. Therefore, if the value for the displacement is considered unreasonable (*s* > *50* is a good threshold, since the clouds do not move at 50 pixels/minute), the algorithm looks for the displacement with respect to the farthest border of the second closest object in *J*. If the resulting displacement is considered unreasonable again, the algorithm uses the farthest border of the third closest object in *J*. In the unlikely case that the third displacement does not make physical sense, the algorithm picks the smallest displacement of the three and uses the farthest border of the corresponding object in *J* for the calculations.

After the most accurate displacement has been determined, the speed of the cloud (*S*, in pixels/minute) is calculated as:

$$S = \frac{s}{\Delta t} \tag{3-17}$$

The cloud direction (in radians) was determined using Eqn. (3-18):

$$Direction = Tan^{-1}\left(\frac{y_{fb_I} - y_{fb_J}}{(x_{fb_I} - x_{fb_J}}\right) \tag{3-18}$$

3.3.4.3.3 Estimating the time for the sun to be uncovered

For most cases, the estimation of the time for the sun to become uncovered ($t_{uncover}$) is more complicated than a simple division of the distance between the sun and the farthest border over the cloud speed. For the case shown in Figure 30.a, a good estimate of $t_{uncover}$ can be obtained by dividing the distance to the farthest border over the cloud speed. Problems arise when dealing with a situation like the one shown in Figure 30.b. The difference between Figures 30.a and 30.b is the direction of movement of the cloud. For the covered case, we are only interested in the portion of the cloud that is moving towards the sun. As long as a section of the cloud is moving towards the sun, the sun will remain covered. The sun will become uncovered only when the farthest border moving towards it reaches the sun and passes through. At that point, the cloud will be *moving away* from the sun and not covering it anymore.

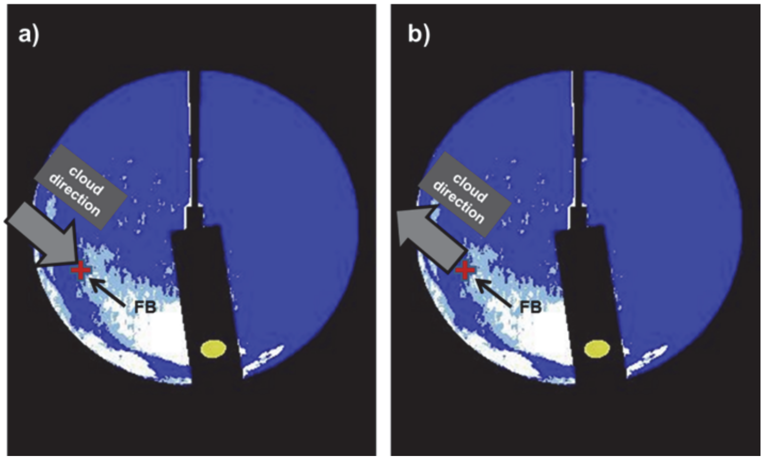


Figure 30.The estimation of the time for the sun to be uncovered depends on the direction of movement of the cloud. a) The cloud is moving towards the sun. b) The cloud is moving away from the sun

In section 3.3.4.3, two different cases were presented: the case in which the cloud is only on one side of the shadow band (Fig. 27.a) and the case in which the cloud is on both sides of the shadow band (Fig. 27.b). The estimation of $t_{uncover}$ follows a different methodology for each of these two cases. The procedure followed to estimate $t_{uncover}$ for the first case is described in section 3.3.4.3.3.1 and the used in the second case is detailed in section 3.3.4.3.3.2.

3.3.4.3.3.1 Case 1: The cloud covers the sun from one side of the shadow band

Depending on the direction of movement and the location of the cloud with respect to the sun, there are four cases that can be presented when a cloud is covering the sun from one side of the shadow band. These cases are shown in Figure 31. The main challenge is to identify which of the four cases is occurring in the current image. To achieve this identification, a criterion was developed based on the direction of movement of the cloud and its location with respect to the sun.

First, the value of *Det*

$$Det_{far} = (x_S - x_C)(y_{FB} - y_C) - (y_S - y_C)(x_{FB} - x_C) \tag{3-19}$$

where the subscript *FB* refers to the farthest border in the closest cloud.

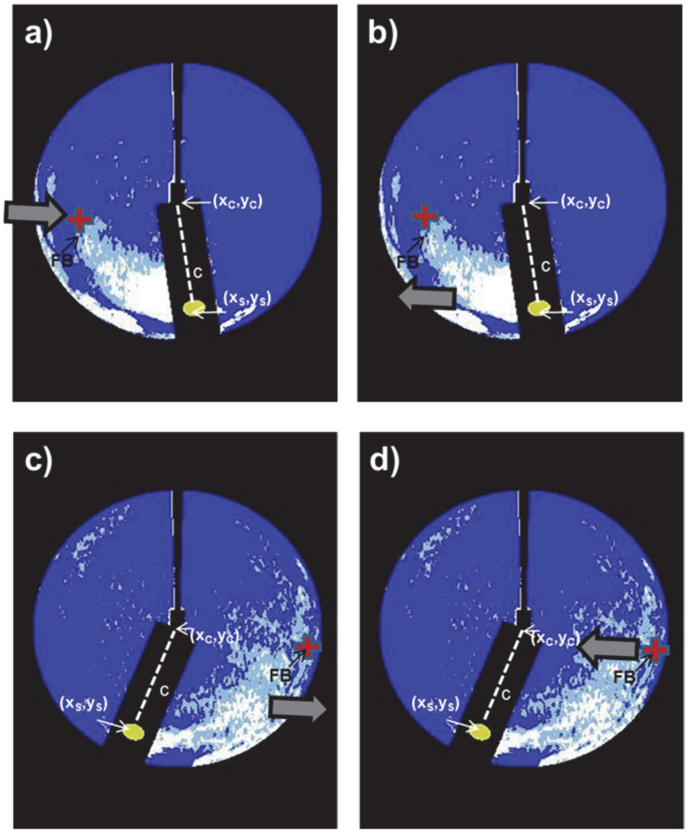


Figure 31. Four cases that can be presented when the cloud is located on one side of the shadow band. a) Cloud to the left of the sun, moving towards the sun. b) Cloud to the left of the sun, moving away from the sun. c) Cloud to the right of the sun, moving away from the sun. d) Cloud to the right of the sun, moving towards the sun.

The sign of Det_{far} is then compared to the sign of the displacement of *FB* in the x-direction, from image *J* to image *I*. This displacement (s_x) is defined as:

$$s_x = x_{fb_I} - x_{fb_J} \tag{3-20}$$

If *Det*$_{far}$ and s_x have the same sign, *FB* is moving towards the sun (cases 1a and 1d, from Figure 31). If the product of *Det*$_{far}$* s_x is negative, *FB* is moving away from the sun (cases 1b and 1c from Figure 31).

In cases 1a and 1d the farthest border of the cloud is moving towards the sun, but in some cases *FB*'s path never crosses the sun's path. As Figure 32 shows, in these cases, the distance of interest for the calculations of the time for uncover is not the distance between *FB* and the sun ($D_{FB\text{-}S}$), but the distance between the sun and the farthest border in the direction of movement of the cloud ($D_{FL\text{-}S}$). If we trace a straight line (*p*) parallel to the direction of the cloud and we extend this line from the sun's centroid towards the cloud's border, the point where the line *p* and the cloud border intercept can be considered the farthest border in the direction of movement of the cloud (*FL*).

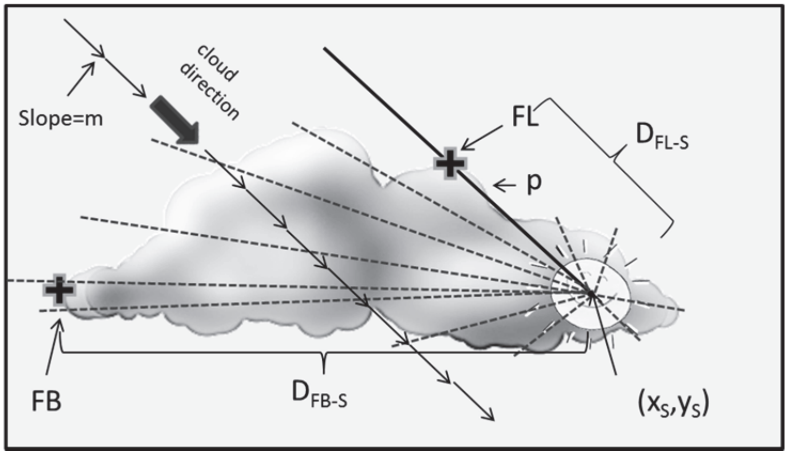


Figure 32. Calculation of the distance between the sun and the farthest border of the cloud in the direction of movement of the cloud

To find *FL* programmatically, the algorithm first calculates the slopes of the straight lines formed between the sun and each of the points in the cloud's border, using the following expression:

$$m_{CO_i} = \frac{y_{CO_i} - y_S}{x_{CO_i} - x_S} \qquad \textbf{(3-21)}$$

where the subscript *CO* refers to the closest object to the sun.

The slopes computed using Eqn. (3-21) are compared to the slope that corresponds to the cloud's direction of movement (*m*). The point of the cloud's border that generates the slope that is closer to *m* is identified as *FL*.

Then, $t_{uncover}$ is calculated using the following expression:

$$t_{uncover} = \frac{\sqrt{(x_{FL}-x_S)^2+(y_{FL}-y_S)^2}}{S} \qquad \textbf{(3-22)}$$

To understand how to deal with cases 1b and 1c, it is important to go back to the observer's example. An observer would look first at the cloud's direction. If the cloud is moving away from the sun, he would look at the farthest border on the opposite side of the cloud to try to predict the time for uncover, as shown in Figure 31.b and 31.c. This opposite border is moving *towards* the sun. For the specific case in which the cloud is only on one side of the shadow band, the farthest opposite border is located in the shadow band region, making it impossible to identify its location from the image. To overcome this limitation, we assumed that the distance that the cloud has to travel to uncover the sun (D_{OB-S}) is equal to the distance from the sun to the edge of the shadow band (D_{SB-S}). Therefore, $t_{uncover}$ will have a value of *21/S* for cases 1b and 1c. This is a significant assumption, but the instances in which the opposite border of the cloud lies inside the shadow band region are few. In addition, the classification techniques that will be used for solar radiation forecast will be able to "learn" this assumption, reducing the mispredictions that can result from it.

3.3.4.3.3.2 Case 2: The cloud covers the sun from both sides of the shadow band

Figure 33 shows the four cases that can be presented when a cloud is covering the sun from both sides of the shadow band. As in the cases described in section 3.3.4.3.3.1, the difference between these four cases is the direction of movement and the location of the cloud with respect to the sun.

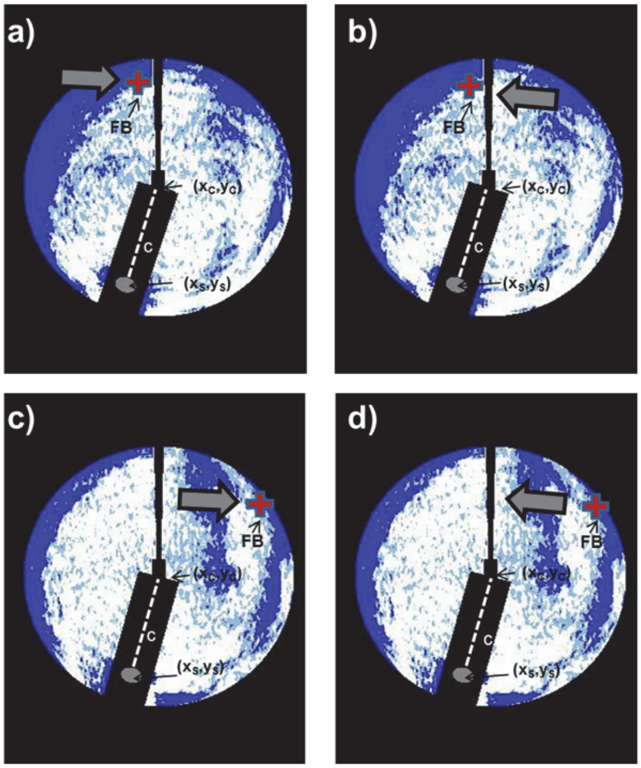


Figure 33. Four cases that can be presented when the cloud is covering the sun from both sides of the shadow band. a) Farthest border located to the left of the sun, border moving towards the sun. b) Farthest border located to the left of the sun, border moving away from the sun. c) Farthest border located to the right of the sun, border moving away from the sun. d) Farthest border located to the right of the sun, border moving towards the sun.

The method described in section 3.3.4.3.3.1 was employed to differentiate between cases 2a an 2d and cases 2b and 2c. The determinant used to identify where the farthest border lies with respect to line *c* (Det_{far}) was estimated using Eqn. (3-19). Then, the sign of Det_{far} was compared to the sign of the displacement of *FB* from image *J* to image *I,* in the x-direction (s_x).

If the product $Det_{far}* s_x$ is positive, the farthest border of the cloud (*FB*) is moving towards the sun (cases 2a and 2d). In these situations, $t_{uncover}$ is calculated following steps described in section 3.3.4.3.3.1 for cases 1a and 1d (Eqns. (3-21) and (3-22)).

For the cases in which the product $Det_{far}* s_x$ is negative (cases 2b and 2c), we refer once again to the observer's example described in section 3.3.4.3.3.1. In contrast to cases 1b and 1c, cases 2b and 2c present a cloud located on both sides of the shadow band. This cloud is divided by the shadow band into two objects. What this means is that the opposite side of the cloud is actually a border in the *second closest* object. As Figure 34 shows, to find *FL* programmatically the algorithm uses Eqn. (3-21) to estimate the slopes of the straight lines formed between the sun and each of the points at the border of the second closest object. Then, the point at the border that forms a line with the same slope as the direction of movement of the cloud (*m*) is identified as *FL*. Once *FL* is found, $t_{uncover}$ is estimated using Eqn. (3-22).

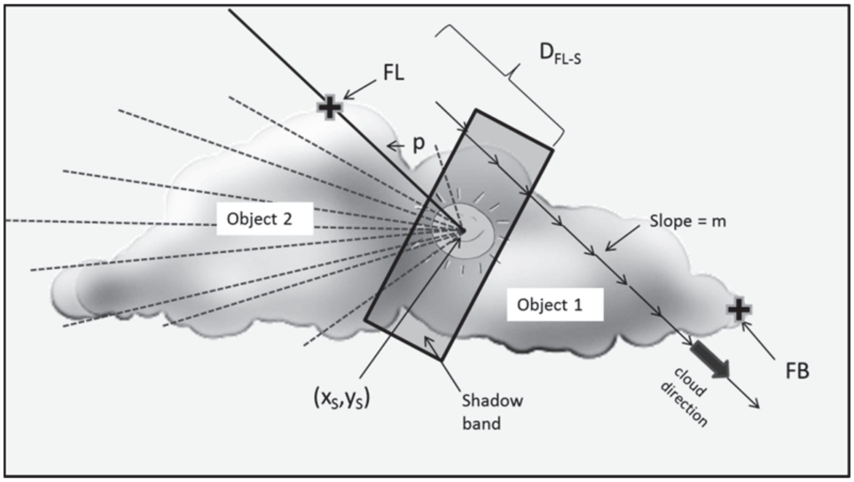


Figure 34. Determination of *FL* for cases 2a and 2b

3.4 Solar irradiation Prediction Using Machine Learning Techniques

3.4.1 Introduction

The strategy presented in section 3.2 consists of using machine learning tools to predict the incoming solar irradiation one minute ahead. Machine learning is the process of taking raw data and inferring the structure that lies underneath it [89]. Given a set of objects –each belonging to a group or class- a machine learning technique can be used to predict the group to which a new object belongs, based on the characteristics or attributes of the object. In this work, the open-source software WEKA [84] was used to apply different learning techniques, called classifiers, in order to select the best performing one for the prediction of solar irradiation.

WEKA is a workbench for machine learning that contains a set of state-of-the-art machine learning algorithms [89]. The workbench was developed at the University of Waikato in New Zealand and the name stands for *Waikato Environment for Knowledge Analysis*.

In order to be used for prediction, the classifier has to be previously “trained” with a set of data before it can classify any new objects. A set of data taken from the SRRL database was used to train the classifiers. The trained classifier was then validated, using a different set of data, corresponding to different dates throughout the year.

Since most of the learning schemes tested in this work perform some type of regression on the data, the root mean squared error (RMSE) was used to determine the best performing classifier for each scenario. Once a classification technique is chosen and trained, the classifier will be able to classify new instances of data, i.e., it will be able to predict the solar irradiation for new sets of data.

The steps followed in the development of the solar irradiation predictor based on learning techniques are described in the following sections. Section 3.4.2 details the steps followed to generate the training data sets. Section 3.4.3 describes the classifiers that were tested for all the scenarios. The performance of the classifiers tested for each scenario is compared in section 3.4.4.

3.4.2 Creating Training Sets

Before a classifier can be used for prediction, it needs to be *trained* using a training data set. The training set is a collection of instances, where each instance is an individual and independent example of the concept to be learned [89], and it contains the attributes and actual values of the variable (or class) that we want to predict. Through these examples, the classifier algorithm can *learn* if a correlation exists between the attributes and the outcome of the class of interest. The features that the classifier learns can then be applied to predict the outcome of other instances. As it can be expected, the quality of the prediction depends heavily on the quality of the training set provided. Therefore, it is important to use a representative data set that incorporates all the different instances that can be presented in the outcome of interest.

WEKA -the software used for machine learning- requires the user to provide data sets in the form of ARFF files, which is a standard way of representing datasets consisting of independent and unordered instances [89]. Any comma–separated-values file (CSV) can be converted into an ARFF file. The first step in the training process is the generation of these CSV files that contain representative instances of the system.

A series of images were sampled every minute from the YES TSI-880 at the SRRL. A MATLAB® script generated for this purpose analyzed each of the images in the series and, according to the scenario presented, calculated a set of image-based attributes. The image-based attributes were combined with a selection of atmospheric attributes obtained from the SRRL database [30]. This database provides an extensive amount of recorded measurements, such as direct normal irradiance, wind speed, azimuth and zenith angles, relative humidity, barometric pressure, etc. One-minute data for the dates of interest were directly downloaded from the SRRL database into a text file. The MATLAB® script read this file and matched the image-based attributes with the corresponding atmospheric attributes, for each picture. The image-based and atmospheric attributes for each picture were stored into a new text file, according to their scenario. The actual value of the one-minute-ahead DNI was also stored in the same file. Table 7 shows a list of all the attributes used for all scenarios. Attribute 1 is the minutes after midnight (UTC), attributes 2 -11 are the image based attributes and attributes 12-28 are the atmospheric attributes. Table 8 specifies which of the attributes in Table 7 were used for each of the five possible scenarios (sunny, cloudy, covered, uncovered, error).

Table 7. List of attributes used for all scenarios

Attribute index	Attribute name	Units	Data type
1	Minutes after midnight, $time_{mid}$	minutes	numeric
2	Blue pixels fraction, $R_b(t)$		numeric
3	Old blue pixels fraction, $R_b(t-1)$		numeric
4	Cover status, $C(t)$		nominal
5	Old cover status, $C(t-1)$		nominal
6	Distance from closest border of closest approaching cloud to the sun, $D_{appcloud}$	pixels	numeric
7	Distance from farthest border of the cloud to the sun, D_{FL-S}	pixels	numeric
8	Speed of the cloud (speed of approaching cloud if uncovered and speed of covering cloud if covered), S	pixels/min	numeric
9	Direction of the cloud (of approaching cloud if uncovered and of covering cloud if covered), *Direction*	radians	numeric
10	Time for cover, t_{cover}	minutes	
11	Time for uncover, tuncover	minutes	numeric
12	Current DNI, $DNI(t)$	W/m^2	numeric
13	Derivative of DNI, ΔDNI	$W/m^2\ min^{-1}$	numeric
14	Dry bulb temperature, T	°C	numeric
15	Wet bulb temperature, T_{WB}	°C	numeric
16	Dew point temperature, T_D	°C	numeric
17	Relative humidity , RH	%	numeric
18	Total cloud cover , CC_{tot}	%	numeric
19	Opaque cloud cover , CC_{opaque}	%	numeric
20	Average wind speed at 22 ft, $S_{w,22}$	m/s	numeric
21	Average wind speed at 42 ft, $S_{w,42}$	m/s	numeric
22	Peak wind speed at 22 ft, $S_{pw,22}$	m/s	numeric
23	Peak wind speed at 42 ft, $S_{pw,42}$	m/s	numeric
24	Wind direction at 22 ft, $D_{w,22}$	° from N	numeric
25	Wind direction at 42 ft, $D_{w,42}$	° from N	numeric
26	Pressure, P	mbar	numeric
27	Zenith angle, α	°	numeric
28	Azimuth angle, β	°	numeric

Table 8. Attributes used in the training sets for each scenario

Scenario	Attributes used
Sunny	1, 2, 3, 12-28
Cloudy	1, 2, 3, 4, 5, 12-28
Covered	1, 2, 3, 5, 6, 8, 9, 10, 12-28
Uncovered	1, 2, 3, 5, 7, 8, 9, 11, 12-28
Error (missing picture)	1, 12-28

,

A separate data set was generated for each scenario, containing its corresponding image-based attributes, atmospheric attributes and one-minute-ahead DNI. The data were preprocessed in order to eliminate infinite values or *NaN* values, which were replaced by a *'?'*, a symbol that WEKA interprets as a missing value.

In order to encompass as many situations as possible, the training sets were generated using instances throughout the year. This ensures that season-to-season variations are well represented. Due to problems with the imager, very few images could be obtained during the summer months, causing the proportion of instances to lean towards the winter months. Despite these problems, some data for July could be collected and were included in the training sets. The training sets included data from January 26th, January 27th, January 29th, January 30th, February 19th– 22nd , July 25th -26th , November 15th-16th and December 2nd, all in 2012. To calculate performance indicators for the classifiers, such as the mean squared error (MSE), a validation data set was required for each scenario. The validation sets were generated using data from January 28th, February 17th and December 3rd of 2012.

The ARFF files deal with two data types: nominal and numeric. Most of the attributes used for the DNI predictor are numeric, which means that they can take any real value. On the other hand, attributes such as *C(t)* (which indicates if the sun is currently covered by a cloud or not) can only take values of 0 or 1. These attributes are nominal. When a CSV file is opened in

WEKA, the software automatically assumes that all the attributes with numeric values are numeric; therefore, it is required to apply a filter that converts the nominal attributes from numeric to nominal. The last column on Table 7 indicates the data type of each attribute. All the nominal attributes should be converted into the correct data type before training the classifier. Once this conversion is done, the files can be saved directly in the ARFF format and used for classification.

3.4.3 Classifier Selection

WEKA has an extensive collection of algorithms designed to work best with numeric attributes. In the development of the one-minute-ahead DNI predictor, several of these algorithms were tested for each of the scenarios and their performance was compared. However, before any of the algorithms were tested, the attributes that have more influence for each case were identified. The use of irrelevant attributes in a dataset can often degrade the performance of a machine learning system, because the learner gets confused by them [89]. Reducing the number of attributes required for the prediction can also reduce the computational time of the process. Thus, attribute selection is an important step in the classification process.

Instead of selecting the most relevant attributes for each scenario independently from the classifier used, we used a *metalearner* that automatically selects the most relevant attributes during the learning process. Metalearning algorithms can improve the learning of any classifier [89]. The metalearning algorithm used in this case was the *Attribute Selected Classifier*, which automatically performs an attribute selection before applying the classifier or prediction scheme.

Since the Attribute Selected Classifier is a metalearner, it needs to be applied in combination of other classifiers or learning algorithms. The one-minute-ahead DNI is a numeric class, and most of the attributes used are also numeric, hence numeric prediction algorithms, such as different types of regression, must be used to generate the prediction. Different numeric prediction algorithms were tested and their performance compared, for each of the scenarios.

The criterion used to evaluate the performance was the root-mean-square error (RMSE), which is defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(p_i - a_i)^2}{n}} \quad \textbf{(3-23)}$$

where *p* represents the predicted values and *a* represents the actual values.

Another metric used for the evaluation of the algorithms' performance was the correlation coefficient. The correlation coefficient (CC) is defined as:

$$CC = \frac{S_{PA}}{\sqrt{S_P S_A}} \quad \textbf{(3-24)}$$

where $S_{PA} = \frac{\sum_i (p_i - \bar{p})(a_i - \bar{a})}{n-1}$, $S_P = \frac{\sum_i (p_i - \bar{p})^2}{n-1}$ and $S_A = \frac{\sum_i (a_i - \bar{a})^2}{n-1}$. The accent ¯ represents the average value of the corresponding variable.

The prediction algorithm that produced the lowest RMSE was selected for each case. The results of this evaluation are presented in section 3.4.4.

Some of the machine learning algorithms tested in this work are described below in the following sections.

3.4.3.1 Linear Regression

When the desired outcome and the most relevant attributes are numeric, linear regression is a technique that comes naturally into consideration. Linear regression is the simplest numeric prediction technique, but sometimes it can perform as well or better than other more complex techniques, so it is always a good start. Linear models can also serve as a base for the development of more complex models.

The predicted class or desired outcome is expressed as a linear combination of attributes, with predetermined weights or coefficients. In general, a linear regression can be defined as:

$$y = w_0 + w_1 x_1 + w_2 x_2 + \cdots + w_k x_k \quad \textbf{(3-25)}$$

where *y* is the predicted class, $x_1, x_2 \ldots x_k$ are the attribute values and w_0, w_1, $w_2 \ldots w_k$ are the weights. The weights are calculated from the training data, by finding the values that minimize the sum squared of the error over all the training instances.

3.4.3.2 Least Median Squared Linear Regression

Since least squares (LS) linear regression works by minimizing the sum squared of all the errors, large outliers can really affect the results obtained. In other words, LS linear regression is not really robust to the presence of large outliers. Besides outliers in the outputs, another problem with LS regression is leverage, which is caused by the presence of observations that are very different from the range of interest (e.g. very large values of the independent variables).

A more robust regression method is the Least Median Squared (LMS) linear regression. Instead of minimizing the sum squared of the error like in LS linear regression, in the LMS regression, the weights are calculated by finding the values that minimize the median of the squared error [90]. This means that the criterion for the LMS regression is:

$$\min_{\boldsymbol{w}}\left[med\left(r_i^2\right)\right] \qquad \textbf{(3-26)}$$

where r_i represents the residual or prediction error at each instance and $\boldsymbol{w}$ is the *1 x n* vector of linear coefficients.

3.4.3.3 Pace Regression

Pace regression improves the classical LS linear regression by evaluating the effect of each variable and using a clustering analysis to improve the statistical basis for estimating their contribution to the overall regressions [91]. This method, developed by Wang and Witten [92], estimates the distribution of the effects of variables from the data and uses them to improve modeling. Pace regression is especially useful when dealing with a large number of coefficients, since it automatically reduces the dimensionality of the data set [93]. Pace regression outperforms LS linear regression, but the WEKA algorithm cannot handle missing values in the data sets; therefore, it could not be tested in all the scenarios.

3.4.3.4 Gaussian Processes

This method uses Gaussian Distributions to reproduce properties of neural network methods [94]. A Gaussian Process is a stochastic process that consists of random values that can be defined by a Gaussian Distribution. Just as a Gaussian Distribution is fully specified by its mean and its covariance matrix, a Gaussian Process is specified by a mean and a covariance function, which are estimated from the training data set.

3.4.3.5 Multilayer Perceptron

Multilayer Perceptron is a feedforward neural network technique that trains using backpropagation [89]. A perceptron represents a *hyperplane* in instance space. A hyperplane partitions the space into two subspaces and it is defined as:

$$w_0a_0 + w_1a_1 + w_2a_2 + \cdots + w_ka_k = 0 \tag{3-27}$$

where a_0, a_1, a_2 … a_k are the attribute values and w_0, w_1, w_2…w_k are the weights that define the hyperplane.

A multilayer perceptron consists of many layers of perceptrons in which the weights of the connections between each of the layers are fitted using the training set. The classifier has to learn two things: the structure of the network and the connection weights. Multilayer perceptrons are usually trained by minimizing the squared error of the network's output.

3.4.3.6 Support Vector Machine for Regression

Support vector machines are learning algorithms based on an algorithm that finds the maximum margin hyperplane [89]. The maximum margin hyperplane is the hyperplane that results in the largest separation between two classes. The instances that are closest to the maximum margin hyperplane are called support vectors. The equation defining the maximum margin hyperplane can be written in terms of the support vectors:

$$x = b + \sum_i \alpha_i y_i \boldsymbol{a}(i) * \boldsymbol{a} \tag{3-28}$$

where y_i is the class value of training instance *a(i)* and *b* and α_i are numeric parameters that are determined by the learning algorithm. The term **a(i)*a** represents the dot product of the test instance with one of the support vectors. The determination of the parameters *b* and α_i requires the solution of a constrained quadratic optimization problem. The algorithm that WEKA uses implements the sequential minimal optimization algorithm for regression problems proposed by Smola and Schölkopf [95].

3.4.4 Best Performing Classifiers for Each Case

Several learning algorithms, including the ones described in section 3.4.3, were tested for each of the five scenarios. The performance of the classifiers was evaluated using the root-mean-squared error (RMSE) defined in equation 3-23. The classifier with the smallest RMSE was selected as the best performing for each case. The following section describes the results obtained for each scenario.

3.4.4.1 Sunny Scenario

Table 9 shows a comparison of the learning algorithms evaluated for the sunny scenario. The training set contained 939 instances and the validation set was composed of 365 instances.

The two learning algorithms with the lowest RMSE were Linear Regression and Pace Regression. The attributes selected to model the DNI and the coefficients obtained were similar for both methods. Equation 3-29 shows the one-minute-ahead DNI for the sunny case, using the coefficients obtained through linear regression.

$$DNI(t+1) = -0.047 time_{mid} + 0.961 DNI(t) - 1.08 T_{WB} + 91.5 \tag{3-29}$$

Table 9 . Comparison of the learning algorithms evaluated for the sunny scenario

Learning Algorithm	Correlation coefficient	RMSE
Isotonic Regression	0.76	51.7
Least Median Squared Linear Regression	0.76	52.6
Linear Regression	0.76	50.3
Multilayer Perceptron	0.78	53.2
Pace Regression	0.76	50.3
Radial Basis Function Network	0.30	77.7
Support Vector Machine for Regression	0.76	52.0
Regression Tree	0.69	60.0

3.4.4.2 Cloudy Scenario

Table 10 shows a comparison of the learning algorithms evaluated for the cloudy scenario. The training set consisted of 1344 instances and the validation set was composed of 313 instances.

Table 10. Comparison of the learning algorithms evaluated for the cloudy scenario.

Learning Algorithm	Correlation coefficient	RMSE
Gaussian Process	0.88	69.8
Least Median Squared Linear Regression	0.94	49.1
Linear Regression	0.67	139.6
Multilayer Perceptron	0.88	71.4
Pace Regression	0.67	139.2
Radial Basis Function Network	-0.07	147.2
Support Vector Machine for Regression	0.94	50.0
Regression Tree	0.93	55.2

The learning algorithm with the lowest RMSE was the Least Median Squared Linear Regression, which also had the highest correlation coefficient. The resulting function for the one-minute-ahead DNI prediction is shown in Equation 3-30.

$$DNI(t+1) = 2.75R_b(t) + 5.72[C(t) = 0] + 0.995DNI(t) + 0.015\Delta DNI + 0.016P - 13.6 \qquad \textbf{(3-30)}$$

3.4.4.3 Covered Scenario

Table 11 shows a comparison of the learning algorithms evaluated for the covered scenario. The training set contained 1350 instances and the validation set was composed of 366 instances.

Table 11. Comparison of the learning algorithms evaluated for the covered scenario

Learning Algorithm	Correlation coefficient	RMSE
Gaussian Process	0.91	121.3
Least Median Squared Linear Regression	0.91	121.5
Linear Regression	0.91	117.0
Multilayer Perceptron	0.89	135.2
Radial Basis Function Network	0.56	236.4
Support Vector Machine for Regression	0.91	120.1
Regression Tree	0.89	129.8

The performance for most of the learning algorithms was very similar, but the lowest RMSE was obtained using Linear Regression. The one-minute-ahead DNI equation for the covered scenario is described by Equation 3-31:

$$DNI(t+1) = 175R_b(t) + 0.861DNI(t) + 0.583CC_{tot} - 2.39S_{w,_{42}} - 64.1 \qquad \textbf{(3-31)}$$

3.4.4.4 Uncovered Scenario

Table 12 shows a comparison of the learning algorithms evaluated for the uncovered scenario. The training set contained 261 instances and the validation set consisted of 56 instances.

Table 12. Comparison of the learning algorithms evaluated for the uncovered scenario

Learning Algorithm	Correlation coefficient	RMSE
Gaussian Process	0.79	44.7
Least Median Squared Linear Regression	0.94	17.8
Linear Regression	0.75	50.3
Multilayer Perceptron	0.56	54.0
Pace Regression	0.89	37.5
Radial Basis Function Network	0.89	43.8
Support Vector Machine for Regression	0.94	17.1
Regression Tree	0.84	39.8

Least media squared linear regression and support vector machine for regression produced the best results. Since the performance of both of these learning algorithms is very similar (RMSE 17.8 and 17.1 for LMS regression and support vector machine, respectively) and LMS linear regression is less complex to implement, the linear function resulting from the LMS learner was selected to be used in the DNI prediction. The one-minute-ahead DNI prediction for the uncovered case is:

$$DNI(t+1) = -8.83C(t-1) + 0.984DNI(t) + 0.012RH + 0.062CC_{opaque} + 22.8 \tag{3-32}$$

3.4.4.5 Error Scenario

A one-minute-ahead DNI prediction function was also generated for the cases in which no image is available. These cases are referred as *error scenarios*. Because the images are not available for these cases, the attributes used for this scenario are all atmospheric variables (attributes 12-28 from Table 7).

The training set for this scenario was composed of 531 instances, while the validation set consisted of 149 instances. Table 13 shows a comparison of the performance of the different learning algorithms tested for this scenario.

Table 13. Comparison of learning algorithms evaluated for the error scenario

Learning Algorithm	Correlation coefficient	RMSE
Gaussian Process	0.98	82.4
Least Median Squared Linear Regression	0.98	75.2
Linear Regression	0.98	74.1
Multilayer Perceptron	0.98	71.7
Support Vector Machine for Regression	0.98	75.0
Regression Tree	0.98	85.6

The learning algorithm that produced the lowest RMSE was linear regression. Using this algorithm, the one-minute-ahead DNI for the error scenario can be described by:

$$DNI(t+1) = 0.955DNI(t) + 0.802T - 0.554CC_{opaque} + 26.3 \quad \textbf{(3-33)}$$

3.5 Results and Discussion

Table 14 summarizes the relevant attributes for each of the scenarios.

Table 14. Relevant attributes for each of the scenarios

Scenario	Relevant attributes	Relevant attributes index
Sunny	minutes after midnight, current DNI, wet bulb temperature	1, 12,15
Cloudy	blue pixels ratio, cover status, current DNI, DNI derivative, pressure	2, 4, 12, 13, 26
Covered	blue pixels ratio, current DNI, total cloud cover %, average wind speed at 42 ft.	2, 12, 18, 21
Uncovered	old cover status, current DNI, relative humidity, opaque cloud cover %	5, 12, 17,19
Error	current DNI, dry bulb temperature, opaque cloud cover %	12, 14, 19

Image-based attributes, such as the speed of closest approaching cloud and the distance from the cloud to the sun, were expected to improve the performance of the prediction for the cases in which the DNI changes drastically, more specifically for the covered and uncovered scenarios. However, as Table 14 shows, only the blue pixels ratio was used from the image-based attributes. The rest of the attributes used presented in Table 14 were sampled directly from the SRRL database.

One explanation for the fact that the image-based attributes were not relevant in the one-minute-ahead DNI predictions is errors in the attributes estimation. For example, for the uncovered case, most of the instances showed cases where the clouds were not approaching the sun. As explained in section 3.3.4.2, when the clouds were not approaching the sun an artificial value for the speed (0.5 pixels/minute) and an artificial value for the distance (255 pixels, equivalent to the diameter of the circle in the image) were assigned to those attributes. Since most of the cases in the training data set show those values, regardless of the actual one-minute-ahead DNI, the learning algorithms assume that they must not be relevant for the prediction.

Inaccuracies in the time for cover or time for uncover calculations might also be causing these attributes not to be relevant. Most of the inaccuracies are caused by the presence of the shadow band, despite the efforts to avoid mispredictions by implementing the strategy described in section 3.3.4.2.2. Other authors [76] have also had troubles caused by the shadow band and, in response, focused on prediction horizons larger than three minutes ahead.

Another source of error is the coordinates of the system. The total sky imager takes a picture of a hemispherical surface, which means that the pixels in the image should be considered to be in spherical coordinates and transformed to rectangular coordinates before calculating the distances between two points. Clouds at the edges of the circle would appear to move faster if this transformation is not done. Marquez and Coimbra [76] also found that making the transformation to a rectangular grid was necessary before analyzing the movement of the clouds.

Since the image-based attributes were not used, a new classifier was trained using the data from all the scenarios together, containing only the atmospheric attributes. As in the individual scenarios, different learning algorithms were tested and the one with the lowest RMSE was selected. For the combined-scenarios case, the best performing learning algorithm was linear regression. The one-minute-ahead DNI for the combined scenarios can be calculated by:

$$DNI(t+1) = 0.923DNI(t) - 0.661CC_{tot} + 64.6 \quad \textbf{(3-34)}$$

The one-minute-ahead DNI calculated using Eqn. (3-34) will be used as a baseline for the evaluation of the performance of the proposed strategy. The proposed strategy needs to outperform the combined scenarios prediction in order to be considered useful.

Another baseline for comparison is the persistence model, in which we assume that the one-minute-ahead DNI will be equal to the current DNI. This strategy has been shown to be successful for really short horizons. The persistence model can be summarized as:

$$DNI(t+1) = DNI(t) \tag{3-35}$$

Table 15 presents a comparison of the performance of the three DNI predictors, for each scenario. As this table shows, the advantage of using the proposed strategy over predictions that do not use information from the images or a persistence model is marginal.

Table 15. Comparison of the performance of the three DNI predictors, for each scenario

Scenario	Proposed Strategy, RMSE	No image prediction, RMSE	Persistence Model, RMSE
Sunny	50	52	52
Cloudy	49	50	49
Covered	117	119	120
Uncovered	18	20	17

Figure 35 shows a comparison of the predicted one-minute-ahead DNI with the measured DNI, for two different dates (February 27th and November 14th). It also shows the comparison with a persistence model (dashed line). Both the proposed strategy and the persistence model perform really well for cases in which the DNI is not varying sharply. Figure 36 shows a zoomed in version of the predictions, where it is clear that the proposed strategy only recognizes the drops in DNI just after they happen. As it can be seen in this figure, the proposed strategy did not represent an improvement over the persistence model. The main reason for this is the high number of instances for which the one-minute-ahead DNI is equal to the current DNI. The cases in which the DNI change was higher than +/-50 W/m^2 from one minute to the next accounted for only 10% of the data. Since the learning algorithms aim at minimizing the overall error, they focus on obtaining an accurate prediction that satisfies the majority of instances, neglecting big errors in the cases with drastic changes, because these errors are fewer.

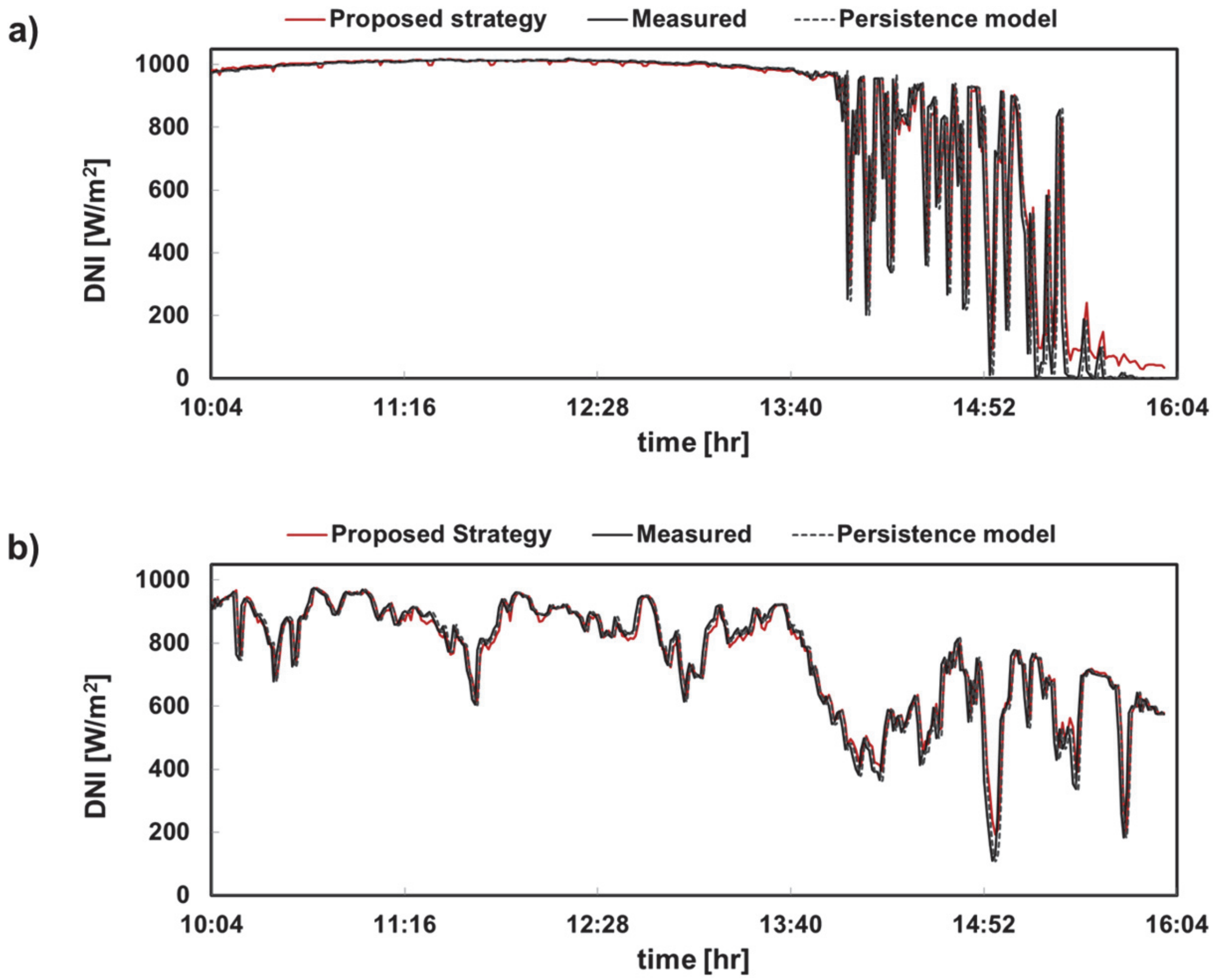


Figure 35. Comparison between the predicted one-minute-ahead DNI and the measured DNI, for two different dates. a) February 27th, 2012. b) November 14th, 2012.

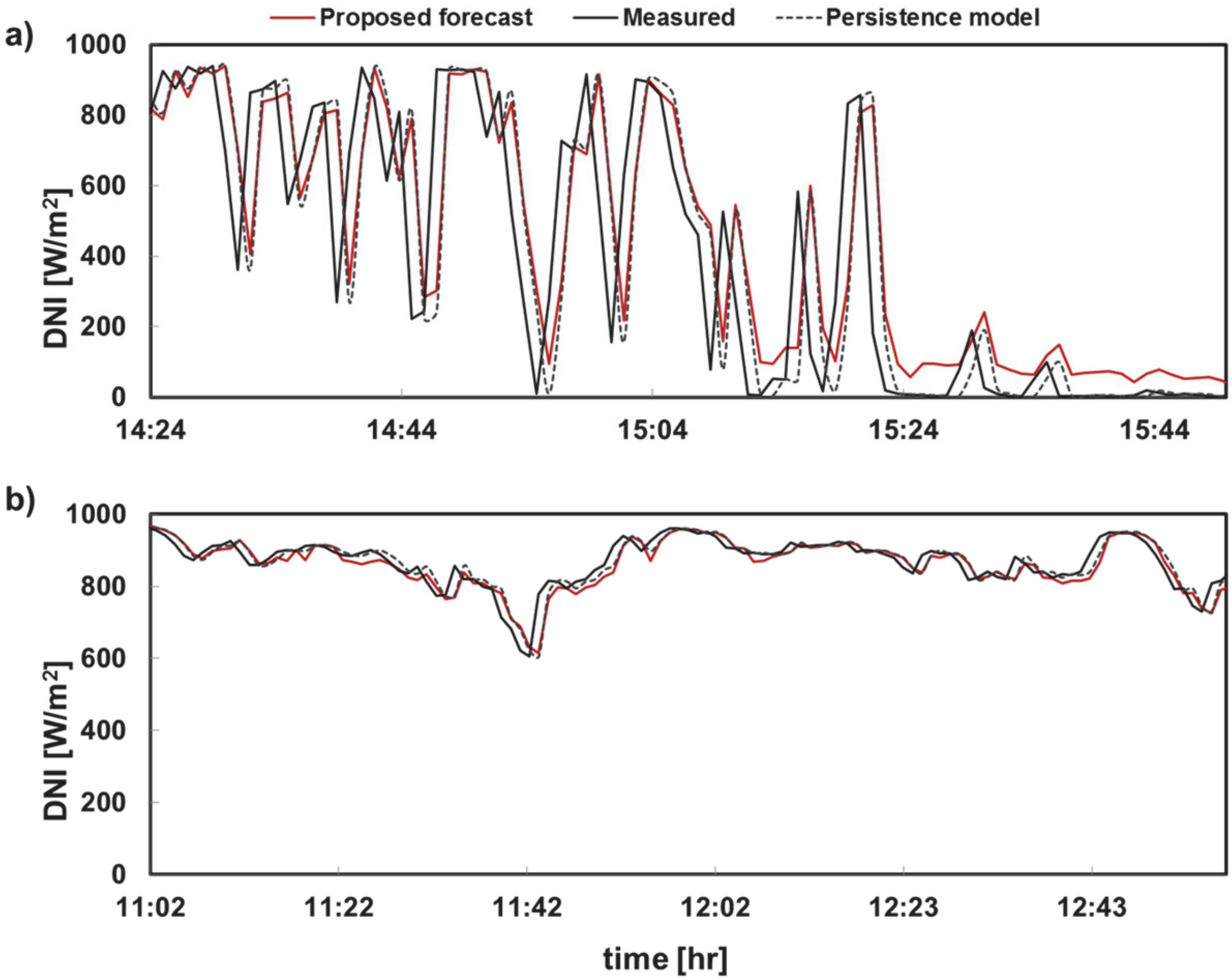
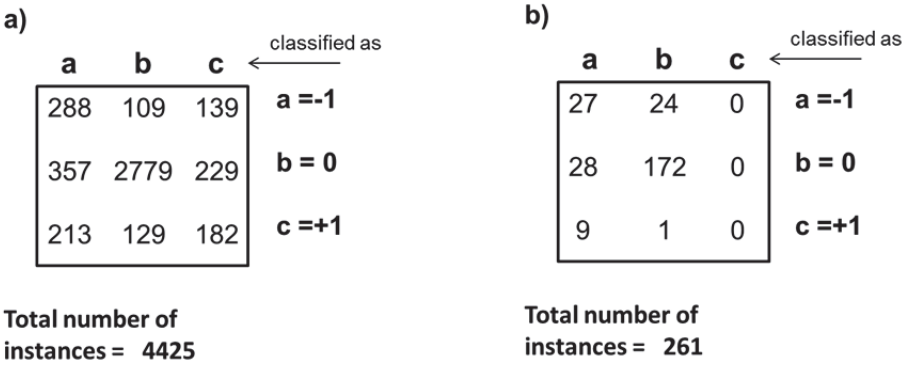


Figure 36. Comparison between predicted and measured DNI, enlarged scale. a) February 27th, 2012. b) November 14th, 2012.

For control purposes, these hard-to-predict DNI changes are the cases of interest, as big changes in DNI would require some action on the control part. Since these predictions are so important, other strategies were tested with the aim of improving the performance of the predictor for the cases with drastic changes. One technique that can be applied consists in increasing the number of instances that present sudden changes in the DNI (sampling a higher number of unusual instances). The problem with the implementation of this technique is that the learning algorithm will fail to predict the cases in which the DNI actually stays constant. This would cause false positives in the predictions, that is, the predictor would forecast drastic changes in DNI for instances in which the DNI does not change.

Another strategy that could provide a better prediction was to, instead of trying to predict the numeric class, classify the instances into three nominal classes: drastic increase in DNI, drastic decrease in DNI, and no drastic change in DNI. Then, depending on the class, an average change constant would be added to the current DNI in order to get the predicted value for the one-minute-ahead DNI. This strategy was tested using all the instances for the combined scenarios and also just using the instances for the uncovered case. The uncovered case was tested in order to determine if the strategy proposed of dividing the data into different scenarios would improve the performance. The threshold applied to classify an instance as a drastic change (in the training set) was a change in DNI of 35 W/m^2 or more from one minute to the next. The learning algorithm that performed best was *BayesNet*, which learns Bayesian networks using nominal attributes (numerical attributes are discretized) [89]. Figure 37.a shows the confusion matrix for the classification in three classes according to the change in DNI for the combined-scenarios case and Figure 37.b shows the equivalent results for the uncovered scenario. The confusion matrix shows the number of instances that were predicted to be in each class in relation to the actual class they belong to. Class *a* represents a drastic decrease in DNI, class *b* represents no drastic change in DNI and class *c* represents a drastic increase in DNI. The diagonal values of the confusion matrix represent the number of instances that were predicted accurately, which equal 73% for the combined scenario case and 76% for the uncovered case. The second column represents the instances classified as *b*, with the top row representing the instances with drastic decreases that were predicted to have no change and the bottom row representing the instances with drastic increases that were predicted to have no change. The combination of these mispredictions is considered a false negative, which accounted for 5% of the instances for the combined-scenarios case and for 10% of the instances for the uncovered case. The false positive cases, i.e., cases that were predicted as drastic changes with no drastic changes occurring in reality, are represented by the first and third columns in the second row. The combined-scenario case predicted 13% of the instances as false positives in comparison to 10% for the uncovered case. The numbers in the corners of the matrix represent the instances that were predicted to have a drastic increase, but, in reality showed a drastic decrease (row 1, column 3) or the instances that were predicted to have a drastic decrease, but, in reality increased (row 3, column 1). The percentage of these extreme mispredictions was 8% for the combined-scenarios case and 3% for the uncovered case.



a)

a	b	c	← classified as
288	109	139	a =-1
357	2779	229	b = 0
213	129	182	c =+1

Total number of instances = 4425

b)

a	b	c	← classified as
27	24	0	a =-1
28	172	0	b = 0
9	1	0	c =+1

Total number of instances = 261

Figure 37.Confusion matrix for the classification in three classes, according to the changes in DNI. a) Combined scenarios. b) Uncovered case.

After this classification was performed, the forecast of the one-minute-ahead DNI was calculated using:

$$DNI(t+1) = DNI + \overline{\Delta DNI_c} \qquad \textbf{(3-36)}$$

where $\overline{\Delta DNI_c}$ represents the average change constant for each class, which was calculated by taking the average change in DNI for all the instances in each class. Table 16 shows the values of the average change constants for each class, for the combined-scenarios case and the uncovered case.

Table 16. Average change constants for each class

Class	$\overline{\Delta DNI_c}$ for combined scenarios case	$\overline{\Delta DNI_c}$ for uncovered case
a=sudden decrease	-142	-162
b=no sudden change	3	-3
C=sudden increase	122	32

The performance of the new prediction strategy was compared to the performances of the original proposed strategy (using a numeric classifier such as linear regression) and the persistence model. Table 17 shows the RMSE values for all the strategies being evaluated. As Table 17 shows, the performance of all the models is the same for the uncovered case. In the combined-scenarios case, the numeric classifier strategy performs best.

Table 17.Comparison of different DNI forecasts

Model used	RMSE for combined scenarios case	RMSE for uncovered case
Nominal classification	558	115
Persistence model	564	115
Linear regression	556	115

As it was mentioned before, accuracy in the predictions for the instances with drastic DNI changes is more relevant from the control point of view than overall accuracy. Thus, an RMSE value that only included the predictions for these instances was calculated, for all the strategies proposed. These RMSE values are shown in Table 18. Even though the performances of the nominal classification model and the linear regression model are very similar for the combined-scenarios case, the nominal classification strategy performs best for the uncovered case. This suggests that a combined strategy that uses a vision-based algorithm to determine the scenario and to estimate image-based attributes that can be later used in a nominal classification algorithm, might improve the overall performance of the DNI forecast. A scheme that illustrates the suggested strategy is shown in Figure 38.

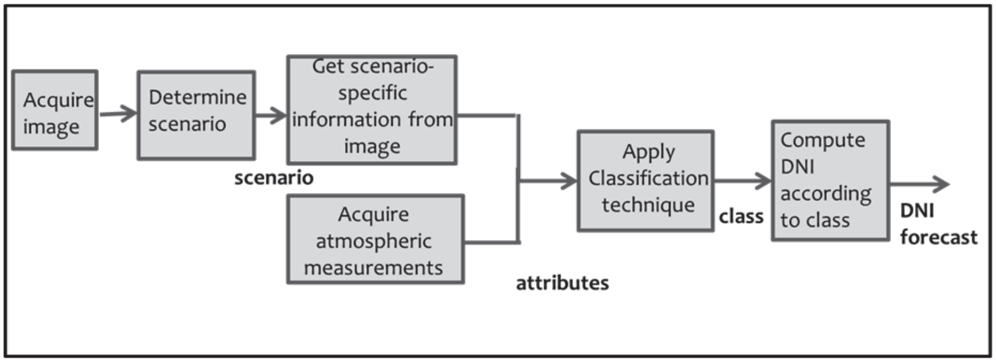

Acquire image
Determine scenario
Get scenario-specific information from image
scenario
Acquire atmospheric measurements
attributes
Apply Classification technique
class
Compute DNI according to class
DNI forecast

Figure 38. Proposed strategy for DNI forecasts that combines image-based scenario determination with a nominal classification algorithm

Another strategy that could improve the performance of the one-minute-ahead DNI forecast is the use of autoregressive models. Instead of treating the data as independent instances, the forecasts may benefit from a longer history for some of the attributes.

3.6 Conclusions

An image-based strategy to predict the one-minute-ahead DNI was proposed and implemented. The strategy consisted on acquiring images using a total sky imager and estimating image-based attributes from them. Before estimating the image-based attributes, the images were classified into a sky scenario (sunny, cloudy, covered, uncovered and error). The image-based attributes were combined with atmospheric attributes and used to train a machine learning algorithm. After training, the most relevant attributes for each scenario could be used directly to predict the one-minute-ahead DNI.

For all the scenarios tested, the image-based attributes were not selected to be used in the final prediction. The DNI predictions were calculated through linear regressions that were strongly correlated with the current DNI. The performance of the proposed strategy showed no improvement over the performance of a persistence model, in which the one-minute-ahead DNI is assumed to be equal to the current DNI. This condition is true for most instances, especially at short time horizons. The instances in which drastic DNI changes occur represent a small percentage of the total number of instances (approximately 10%) and, since classifier training is carried out by minimizing the overall error for all the instances, the classifier ignores the errors in the predictions for the instances with drastic DNI changes and focuses on predicting more accurately the instances with no change. It was expected that the image-based attributes would help recognize the instances with drastic DNI changes, but large errors caused these attributes to become useless for the DNI predictions. It is expected that improvements in the image-based attributes will improve the overall performance of the proposed DNI forecasting strategy. The strategy presented in this work did not show an improvement over simpler DNI forecasts; however, we presented a framework that is expected to work with more accurate image-based attributes.

When the reactor is enclosed in a reflective cavity, the system responds fast to changes in solar irradiation; thus, the horizon for the DNI forecast need to be of the order of one minute. In an industrial application, where an absorbing cavity would be the norm, the dynamics of the system are about 50 times slower, allowing for longer prediction horizons. The application of the proposed strategy over longer prediction horizons (three to five minutes ahead) is expected to show a significant improvement over the persistence model, because the presence of the shadow-band in the picture will be less of a problem at longer horizons.

In conclusion, even though the proposed strategy did not represent an improvement over a persistence model for one-minute horizons, the algorithms developed for the acquisition of image-based attributes and for the DNI forecast still represent contributions that can be applied to predictions over longer horizons.

CHAPTER FOUR

DESIGN AND SIMULATION OF A MODEL PREDICTIVE CONTROL SYSTEM FOR A SOLAR-THERMAL REACTOR

4.1 Introduction

One of the biggest challenges that solar thermochemical processes face is dealing with the intermittent nature of solar irradiation. Changes in solar irradiation directly affect the performance of solar-thermal reactors causing unnecessary shutdowns and start-ups, complications in the purification processes downstream of the reactor, and damage to the reactor materials due to thermal shock. Being able to maintain a continuous high performance operation, even in the presence of passing clouds, is one of the main concerns of the feasibility of solar-thermal processes.

A robust control system can help minimize the problems associated with the solar irradiation intermittency and allow for a more efficient operation. Previous studies have shown that techniques such as dynamic optimization [67], conventional feedback control [96], feedforward control [26], fuzzy logic control [97] or more complex control systems [98]–[100] can really improve the performance of solar-thermal power plants. Camacho et al. performed an extensive survey on the control techniques applied to solar power plants throughout the years [16], [19]. A large number of the control systems for solar power plants were developed for distributed collector systems. The distributed collector systems consist of a series of parabolic mirrors that reflect solar irradiation into a pipe that contains a heating fluid used to produce steam for electricity generation [26]. Typically, solar-thermal reactors operate within a central receiver system, in which a large number of mirrors, or heliostats, track the incident sunrays and concentrate them towards a solar receiver, usually located on the top of a tower [27]. In this type of system, little has been done in terms of manipulating the inlet flow rates to control outlet variables such as temperature and pressure. One of the few control techniques developed [28] focuses on manipulating the water flow rate to control the outlet steam conditions. The controller adapts to two different models: one used in the presence of clouds and one for clear sky. However, manual interventions were necessary to adjust the control scheme since it does not incorporate an adaptive algorithm.

Despite the extensive studies regarding the control of solar power plants, the only known attempt to design a control system for a solar-thermal reactor was developed by Petrash et al. [17]. In their work, they developed a Linear Quadatric Gaussian with Loop Transfer Recovery (LQG/LTR) controller [20] that manipulated the reactant flows according to changes in the measured variables (product composition and reaction temperature). The controlled system showed an improved performance over the uncontrolled system [17]. However, we propose that the performance can be improved even further by implementing a predictive control scheme that takes into account measured disturbances.

A technique that has been successful in the control of solar power plants is Model Predictive Control (MPC) [98], [21]–[24]. The term MPC includes a wide range of control strategies that make an explicit use of a model of the process in order to obtain the control signal by minimizing an objective function [25].

In this work, we developed an MPC system for a solar-thermal reactor in which the flow rates of reactants are manipulated as a response to changes in solar irradiation. The proposed control system uses the Direct Normal Irradiance (DNI) forecasts obtained through the strategies presented in Chapter Three, and then uses the model of the process to predict the response of the process outputs to these changes. With this strategy, the control algorithm can determine the required manipulation that will minimize the error and act accordingly before changes in DNI start happening. The performance of the MPC controller was compared to the performance of more conventional control strategies, such as Proportional-Integral-Derivative

(PID) control, showing that an MPC system can handle the solar irradiation intermittencies better.

4.2 Model Predictive Control

Model predictive control (MPC) is a term used to describe a set of advanced control techniques that make an explicit use of a model of the process to obtain the control signal by minimizing an objective function [25]. MPC algorithms were first developed in the 1970s by two industrial research groups: Shell, who developed the Dynamic Matrix Control (DMC) algorithm [101], and ADERSA, who developed the Model Predictive Heuristic Control (MPHC) [102] ,later known as Model Algorithmic Control (MAC). Both algorithms used a dynamic model of the system (step response in the first one and an impulse response for the second one) in order to estimate the future values of the output in response to future control actions. These predictions were used to compute the future errors with respect to a reference trajectory and to determine the optimal control signal that would minimize those errors [25]. Since the time they were first developed, MPC techniques have had a major impact in industry and have been a relevant research subject both in industry and academia.

One of the major advantages of MPC is that it can manage difficult multivariable control problems. The MPC algorithms can be directly applied to Multiple Input-Multiple Output Systems (MIMO), even if the plant is non-square, i.e., if the plant has a different number of outputs and variables that can be manipulated [103]. In addition, MPC intrinsically introduces feedforward control to compensate for measurable disturbances, and it can automatically compensate for process delays (dead times) [25]. It is also very useful when dealing with constrained systems, as it can prevent violations of input and output constraints before sending the control signal [104].

The biggest drawback of MPC is that it requires an accurate model of the process. Since the MPC explicitly uses the model to predict the future errors and calculate the required control signal, the performance of the controller will be strongly affected by discrepancies between the real plant and the model.

4.2.1 MPC Strategy

MPC calculations are performed out to a moving horizon, which is based on the current sampling instant [105]. Figure 39 shows a graphic representation of the MPC controller's strategy. At any sampling instant, *k*, the current and past values of the measured outputs, y_k, y_{k-1} …, and the past values of the manipulated inputs, u_{k-1}, …, are known. The future outputs, y_{k+1} … y_{k+P}, can be predicted over a determined horizon *P* (called the prediction horizon) using the model of the process. The set of future control actions, $u_k \ldots u_{k+M}$, are calculated over a control horizon, *M,* by minimizing an objective function that contains terms that account for the predicted future errors. The current control signal, u_k, is sent to the process and the future control signals, $u_{k+1} \ldots . u_{k+M}$, are erased. The controller then calculates a new sequence of control moves at the next sampling instant, after new measurements become available, but it only sends the first input move to the process. This approach is known as receding horizon [104]. It is important to note that, once the measurements become available at each sampling instant, the controller computes the difference between the predicted output and the measured output and uses this information to improve the predictions. This strategy is known as output feedback [104].

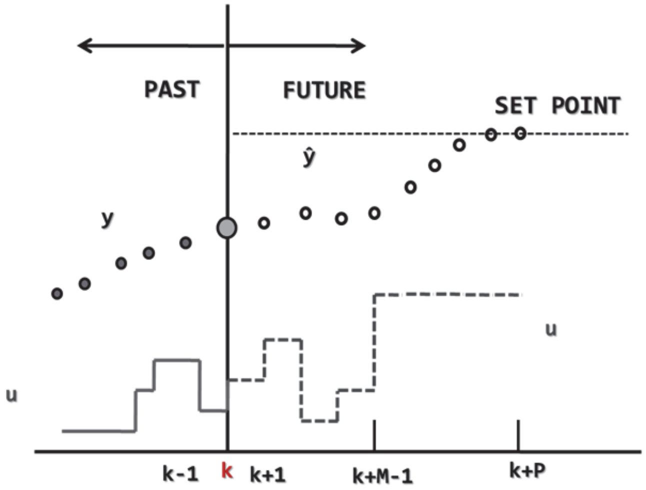


Figure 39. MPC strategy (adapted from [104])

For the cases in which the disturbances are measured, the controller can include these values in the predictions and compensate for the effect of the disturbances on the output immediately, instead of waiting until the effect of the disturbances appear on the measurement. This means that the MPC algorithm provides feedforward compensation for measured disturbances automatically. When the disturbances are measured, the controller can assume a zero-order hold for the output predictions, i.e., the controller can assume that the disturbances maintain the measured value at the sampling instant *k* over the prediction horizon. If predictions for the disturbances are available, disturbance rejection can be further improved by using the predicted values of the disturbances in the calculation of the predicted outputs. This strategy is known as *look-ahead*.

4.2.2 Elements of a Model Predictive Controller

The elements described in this section are common to any MPC strategy. For some of these elements, such as the predictive model structure and the objective function formulation, the different options to be chosen is what gives rise to different algorithms [25]. For others, such as the horizons and weights, different values affect the performance of the controller. In any case, is important to understand the role of each of these elements and its effect on the performance of the MPC controller.

4.2.2.1 Design Parameters

The parameters described in this section must be specified in order to design an MPC. Some of these key parameters can be used to tune the MPC. The effect of each of those parameters on the MPC will be briefly described.

4.2.2.1.1 Control interval or sampling period

The MPC algorithm is typically implemented as a discrete-time controller, which means that it only takes action at regularly spaced, discrete-time instants. The sampling instants are the times at which the controller acts and the control interval is the interval separating the successive sampling instants. Typically, the control interval *(*Δt) is chosen so that the plant's open-loop settling time, t_s*,* is approximately 20-30 control intervals. An equivalent criterion is to make the control interval approximately one fifth of the dominant time constant [103].

4.2.2.1.2 Prediction horizon

The prediction horizon, *P*, is the number of control intervals after the sampling instant *k* for which a prediction of the output is calculated using the process model. A typical rule of thumb is to choose the prediction horizon to be equal to the number of sampling periods used to determine the control interval (20-30, as presented in section 4.2.2.1.1) [103]. Decreasing the value of *P* tends to generate a more aggressive controller [104].

4.2.2.1.3 Control horizon

The control horizon, *M*, is the number of control intervals after the sampling instant *k* for which the control algorithm computes the optimal control moves. As the control horizon increases, the controller tends to become more aggressive and the required computational effort increases. Relatively small values of *M* are generally used in practice, and a typical rule of thumb is selecting a value of *M* between 5 and 20 [104]. For cases with time delay (*D*), it is important to select the values of *P* and *M*, such that *P* >> *D* and *M* << *(P – D)* [103].

4.2.2.2 Process Model

MPC algorithms make an explicit use of the model of the process in order to calculate the output predictions over the prediction horizon. Thus, the model is a key element in the MPC design. An inaccurate model can cause the MPC to perform poorly. Therefore, it is important to dedicate enough effort so that the best possible model can be obtained.

Different strategies of MPC were developed using different models. In practice, any type of model formulation can be used. Linear, time-invariant models are usually preferred over nonlinear models, because they simplify the calculations during the optimization step.

Some of the commonly used models are described in the following subsections. With the exception of the state-space model, the models described here are written in a single input–single output (SISO) form, but they can be extended to accommodate for more input/output combinations. Measured disturbances can also be added to the models, as they can be treated as another input to the process.

4.2.2.2.1 Impulse response model

In an impulse response model, the output, *y*, can be described as a function of the input, u, by the following equation:

$$y(t) = \sum_{i=1}^{N} h_i u(t-i) \qquad \textbf{(4-1)}$$

where *N* is the model horizon and h_i is the sampled output when the process is perturbed by a unitary impulse.

The predictions, using an impulse model, will be given by

$$\hat{y}(t+k|t) = \sum_{i=1}^{N} h_i u(t+k-i|t) \qquad \textbf{(4-2)}$$

Eqn. (4-2) can also be written in terms of the backward shift operator (z^{-1}), as:

$$\hat{y}(t+k|t) = H(z^{-1})u(t+k|t) \qquad \textbf{(4-3)}$$

where *$H(z^{-1})$* is the polynomial function *$H(z^{-1})=h_1 z^{-1}+h_2 z^{-2}+\ldots h_N z^{-N}$*.

Impulse models are typically used in Model Algorithmic Control (MAC) systems. This model is very intuitive and reflects the effects of each manipulated variable on the output very clearly. A disadvantage of this model is that large number of parameters are usually required, as *N* is usually in the order of 40-50 [25].

4.2.2.2.2 Step response model

The step response model has a similar structure to the impulse response model described in section 4.2.2.2.1, except that a step change is applied to the process in order to obtain the parameters for the model.

The predictions, using a step response model, are given by

$$\hat{y}(t+k|t) = \sum_{i=1}^{N} g_i \Delta u(t+k-i|t) \qquad \textbf{(4-4)}$$

where *g_i* are the sampled output values in response to the step input and $\Delta u=u(t)-u(t-1)$.

In terms of the backward shift operator, Eqn. (4-4) can be written as:

$$\hat{y}(t+k|t) = G(z^{-1})(1-z^{-1})u(t+k|t) \qquad \textbf{(4-5)}$$

where *$G(z^{-1})$* is the polynomial defined as *$G(z^{-1})=g_1 z^{-1}+g_2 z^{-2}+\ldots .g_N z^{-N}$*

Step response models also have the disadvantage of requiring a large number of parameters. They are typically used in DMC applications.

4.2.2.2.3 Transfer function models

Discrete transfer function models use the concept of a transfer function to define the output, making it a function of two polynomials, such that:

$$A(z^{-1})y(t) = B(z^{-1})u(t) \qquad \textbf{(4-6)}$$

with $A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} + \cdots + a_{na} z^{-na}$ **(4-7)**

and $B(z^{-1}) = b_1 z^{-1} + b_2 z^{-2} + \cdots + b_{nb} z^{-nb}$ **(4-8)**

where *na* represent the number of poles in the system and *nb* represent the number of zeros.

Using this type of models, the prediction is given by [25]:

$$\hat{y}(t+k|t) = \frac{B(z^{-1})}{A(z^{-1})} u(t+k|t) \tag{4-9}$$

The advantage of this model over impulse and step response models is that it requires a fewer number of parameters (depending on *na* and *nb*). One disadvantage is that prior knowledge of the dynamics of the process is required, especially when specifying the order of the polynomials.

4.2.2.2.4 State-Space models

A state-space model is a compact way of representing dynamic systems by using state variables, which are the defined as the smallest subset of variables that can represent the entire state of the system at any given time.

The general representation for linear state-space models is given by:

$$\dot{\boldsymbol{x}} = \boldsymbol{Ax} + \boldsymbol{Bu} + \boldsymbol{Ed} \tag{4-10}$$

$$\boldsymbol{y} = \boldsymbol{Cx} + \boldsymbol{Du} \tag{4-11}$$

where the bold notation is used to represent vectors and matrices. The vector ***x*** represents the state vector, which contains all the *n* state variables; ***u*** is the input vector of *m* manipulated variables and ***d*** is the vector of *v* measured disturbances. The time derivative of ***x*** (dx/dt) is denoted by $\dot{\boldsymbol{x}}$, which is also a vector. ***A*** is the *n x n* state matrix, ***B*** is the *n x m* input matrix, ***E*** is the *n x v* disturbances matrix, ***C*** is the *n x q* output matrix and ***D*** is the *q x m* feedthrough matrix, which is usually equal to zero. Matrices ***A, B, C, D***, and ***E*** are constant matrices. The vectors in Eqn. (4-10) and Eqn. (4-11) are usually written in terms of deviation variables [104]. The advantage of state-space models is that their implementation for multivariable problems is straightforward [25].

The discrete representation of the state-space models is given by:

$$\boldsymbol{x}(\boldsymbol{k}+\mathbf{1}) = \boldsymbol{Ax}(\boldsymbol{k}) + \boldsymbol{Bu}(\boldsymbol{k}) + \boldsymbol{Ed}(\boldsymbol{k}) \tag{4-12}$$

$$\boldsymbol{y}(\boldsymbol{k}) = \boldsymbol{Cx}(\boldsymbol{k}) + \boldsymbol{Du}(\boldsymbol{k}) \tag{4-13}$$

where *k* represents the current sampling instant.

4.2.2.2.5 Other models

In some cases, nonlinear models can be used to determine the predicted outputs. The use of nonlinear models results in a more complicated optimization problem; however, some processes show a high degree of nonlinearity that cannot be modeled successfully through a linear model. Other techniques that can help avoid the use of nonlinear models exist, such as linearized scheduled MPC, in which a set of linear models are fitted around different operating points. The control algorithm then selects the model to be used in the predictions, depending on the current state of the plant.

Some studies have used other forms to represent the process, such as Neural Networks or Fuzzy Logic [25].

4.2.2.3 Objective Function

The MPC calculates the required control signal by minimizing an objective function. This objective function includes terms that account for the future errors, but it also contains terms that can force the controller to make smaller moves (move suppression). In addition, it contains terms that can keep input variables that are not being used for control at a certain nominal value. The degree to which each of these terms affect the control signal calculation depends on the corresponding weights that they have in the cost equation.

For the future error terms, the controller predicts how much each output will deviate from its setpoint over the prediction horizon. It multiplies each deviation (or error) by the output's weight and calculates the weighted sum squared of the errors, $S_y(k)$, using the following equation [103]:

$$S_y(k) = \sum_{i=1}^{P} \sum_{j=1}^{n_y} \left[w_j^y \left[r_j(k+i) - y_j(k+i) \right] \right]^2 \qquad \textbf{(4-14)}$$

where n_y is the number of outputs of the plant, w^y_j is the weight for output j and r_j is the reference trajectory. As it will be discussed in section 4.2.2.4, if an output is not to be controlled the weight of the corresponding output will be set equal to zero.

Minimizing the value of $S_y(k)$ only can result in an overactive controller. The controller might choose large manipulated-variable adjustments, which could be impossible to achieve or could cause damage to the actuators. In the worst case, an overactive controller can lead to an unstable control system. A way to overcome this is to add a move suppression term to Eqn. (4-14). This term will penalize large changes in the manipulated variables, forcing the controller to select smaller moves. The move suppression term is the weighted sum of controller adjustments, $S\Delta u(k),$ described as [103]:

$$S_{\Delta u}(k) = \sum_{i=1}^{M} \sum_{j=1}^{n_u} [w_j^{\Delta u} \Delta u_j(k+i-1)]^2 \tag{4-15}$$

where n_u is the number of manipulated variables of the plant, $w^{\Delta u}{}_j$ is the weight for each manipulated variable j and Δu_j is the control move, $u_j(k)$-$u_j(k$-$1)$.

In most applications, the manipulated variables should be allowed to move freely within an operating region, to allow the controller to track setpoint changes and to reject disturbances. In some cases, however, plants can have a larger number of manipulated variables than controlled variables. In those cases, a common approach is to hold the extra manipulated variables to a nominal value or setpoint. This can be achieved by adding by adding a setpoint tracking term for the manipulated variables, written as follows [103]:

$$S_u(k) = \sum_{i=1}^{M} \sum_{j=1}^{n_u} [w_j^u [\overline{u_J} - u_j(k+i-1)]]^2 \tag{4-16}$$

where $w^u{}_j$ is the weight for each manipulated variable j and $\overline{u_J}$ is the nominal value of the manipulated variable. If no manipulated variables need to be hold to any setpoint, as it is usually the case, the weight, $w^u{}_j,$ can be made equal to zero.

The standard form of the cost function that the controller algorithm needs to minimize in order to obtain the control signal will be the sum of the terms $S_y(k)$, $S_{\Delta u}(k)$ and $S_u(k)$. For an unconstrained system, the cost function can be written as:

$$J = \sum_{i=0}^{P-1}\left(\sum_{j=1}^{n_y}\left[w_{i+1,j}^{y}[y_j(k+i+1|k) - r_j(k+i+1)]\right]^2 + \sum_{j=1}^{n_u}\left[w_{i,j}^{\Delta u}\Delta u_j(k+i|k)\right]^2 + \sum_{j=1}^{n_u}\left[w_{i,j}^{u}[u_j(k+i|k) - \overline{u}_j(k+i)]\right]^2\right) \tag{4-17}$$

The weights in Eqn. (4-17) have a subscript, *i*, that represents the step in the prediction horizon. This means that the weights for each output or manipulated variable can change over the prediction horizon.

4.2.2.4 Weights

Choosing the weights w^y_j, $w^{\Delta u}_j$ and w^u_j in Eqn. (4-17) is a critical step in the MPC controller design. Adjusting the values of the weights can change the performance of the controller; therefore, it is important to understand the effect that each of the weights has in the controlled-system's response. The tuning of an MPC controller usually involves varying the weights until the desired behavior is achieved.

In the case of the output weights, w^y_j, a larger weight means that minimizing the error for that specific output *j* is more relevant than minimizing the error for outputs with lower weights. In other words, *if* $w^y_j >> w^y_{i\neq j}$ the controller tries to track r_j, sacrificing r_i tracking if necessary [103]. As it was mentioned in section 4.2.3.3, if the weight for an output is zero, the controller completely ignores the deviations from the setpoint for that output. Increasing the values of the output weights, in relation to the move suppression weights, will typically generate a more aggressive controller, because holding the setpoints becomes more important in the objective function than suppressing the moves.

For the move suppression weights, $w^{\Delta u}_j$, increasing the values of the weights will generate a more cautious, conservative controller. If the weights are too large, the controller's setpoint tracking could degrade and the response could become sluggish. On the other hand, a larger value for the move suppression weights can help the controller become more robust, making it less sensitive to prediction inaccuracies.

The manipulated variable weights, w^u_j, are usually set to zero, except for the cases in which there are extra manipulated variables. Increasing these weights for the non-extra manipulated variables would degrade the setpoint tracking of the controller.

As it was briefly mentioned in section 4.2.3.3, the values of the weights can change over the prediction horizon. For example, the output weights that correspond to predicted errors early in the prediction horizon can be given a larger value than the ones that correspond to predicted errors at the end of the horizon, or vice versa [104]. Also, if one of the controlled variables has a faster response than the rest, the output weights for this variable can be given smaller values early in the prediction horizon, to allow the controller to focus the efforts on the slower controlled variables first.

The final weight values selected for each system depend on the process and on the desired performance (i.e., the main objective of the control system). These weights are usually found by analyzing the system's response to different values, using simulations.

4.2.2.5 Constraints

One of the advantages of MPC is that it can easily deal with constraints, both in the manipulated variables and the output variables. Since the MPC algorithm uses a prediction of the output variables before calculating the control signal, it can automatically detect if one of the outputs will violate a constraint, given a certain control signal. In the same way, if the algorithm knows that the manipulated variables are subject to constraints, it would restrict the computed control signals to values that do not violate those constraints.

There are two types of constraints: hard constraints and soft constraints. Hard constraints are the constraints that cannot be violated, and they are seen as a fixed limit that cannot be crossed. The constraints used in the manipulated variables are typically hard constraints. Hard constraints in the manipulated variables can be applied as a minimum and maximum limit for each manipulated variable, but they can also be applied in the rate at which the manipulated variables change. This is especially useful when dealing with actuators that are limited in the rate at which they can respond, for example, a valve that cannot open faster than a given percentage in one time step.

Hard constraints for the manipulated variables and the manipulated variables rates can be expressed as [104]:

$$u^{-}(k) \leq u(k+i|k) \leq u^{+}(k) \quad \textbf{(4-18)}$$

$$\Delta u^{-}(k) \leq \Delta u(k+i|k) \leq \Delta u^{+}(k) \quad \textbf{(4-19)}$$

for *i=0,1, 2…M-1*

where the superscripts – and + represent the lower and upper limits, respectively.

Soft constraints are constraints that could be violated, but the degree of violation is mathematically penalized. This is achieved through a modification in the objective function, which allows small constraint violations for short periods of time. Usually, the constraints in the output variables are soft. Using hard output constraints can result in solutions that are not feasible, because for some conditions (e.g., a large disturbance) constraint violations might be unavoidable. The controller must allow these violations, so the hard constraints are transformed into soft constraints by adding slack variables, ε. The softened output constraints can be expressed as [103]:

$$y^{-}(k+i+1) - \varepsilon V_j^y \leq y(k+i+1|k) \leq y^{+}(k+i+1) + \varepsilon V_j^y \quad \textbf{(4-20)}$$

For i=0,1,2….P-1

where V^y_j represents the Equal Concern for the Relaxation (ECR) vector, which quantifies the concern for relaxing the corresponding constraint: a value of V^y_j equal to zero means that no violations are allowed and the constraint is hard, whereas a value of V^y_j equal to one means that the constraint is soft.

The numerical values of the slack variables can be determined during the optimization, if a term that penalizes the slack variables is added to Eqn. (4-17). The modified cost function becomes:

$$J = \sum_{i=0}^{P-1}\left(\sum_{j=1}^{n_y}\left[w_{i+1,j}^{y}[y_j(k+i+1|k)-r_j(k+i+1)]\right]^2+\sum_{j=1}^{n_u}\left[w_{i,j}^{\Delta u}\Delta u_j(k+i|k)\right]^2+\sum_{j=1}^{n_u}\left[w_{i,j}^{u}[u_j(k+i|k)-\overline{u}_j(k+i)]\right]^2\right)+\rho_\varepsilon\varepsilon^2 \quad \textbf{(4-21)}$$

where ρ_ε is the weight associated with the slack variables.

4.2.2.6 Input Blocking

In Figure 39, the controller has a prediction horizon *P=9* and a control horizon *M=4*. The controller computes the sequence of the optimal *M* moves and then assumes that the input remains constant for the following *P-M* moves.

An alternative is to block the controller moves in a different way. A block can be defined as a group of sampling periods in which the manipulated variable remains constant, i.e. $\Delta u=0$. The duration of the block is the number of sampling periods in each block. When all the unblocked moves happen at the beginning of the prediction horizon, they tend to be larger because they only last one sampling period. In contrast, when blocking is used, the control moves become smaller: because the moves last longer than one sampling period, there is more time for each move to affect the predicted output and, therefore, there are fewer fluctuations. Thus, the use of blocking can lead to smoother responses [103]. In addition, input blocking can reduce the required computational effort for the cases with large control horizons.

4.2.3 MPC Algorithm

The MPC implementation algorithm can be summarized as follows [106]:

The first step is to initialize the controller with the current state of the system. Typically the system starts at steady state and the predictor uses deviation variables, so often the initial state of the system is equal to zero. The second step is to use the model of the process to predict the state of the system over the prediction horizon. After the predicted outputs are obtained, the controller computes the sequence of control moves, $\Delta u(k|k)$, $\Delta u(k+1|k)$, … $\Delta u(M-1+k|k)$, that minimize the objective function presented in Eqn. (4-17). If the system is subject to soft constraints in the outputs, the numerical values for the slack variables ε need to be computed, in addition to the sequence of *M* moves, by minimizing the cost function presented in Eqn. (4-19). Once the sequence of moves is computed, the first move is used to determine the control signal, $u(k)$, and this control signal is sent to the plant. After the control move is implemented, the actual plant outputs are measured and the measurements are used to update the predictions (output feedback). At this point, the sampling instant $k+1$ becomes the current sampling instant (k) and the process is repeated again for the new current sampling instant.

For the unconstrained MPC, an analytical expression for the optimal sequence of moves can be derived offline and used to compute the control moves at every sampling instant, reducing the computational effort. In the case of the constrained MPC, the optimization problem needs to be solved at each control instant. For linear models, the optimization problem can be solved using Quadratic Programming (QP) techniques.

4.2.4 Development of an MPC Controller

Figure 40 shows a block diagram with the steps that were taken towards the development of the MPC controller. The first step is the initial controller design, in which the controlled, manipulated and disturbance variables are selected. The main control objectives were also stated in this phase. The selections made during the initial controller design phase can be modified or revised after dynamic testing.

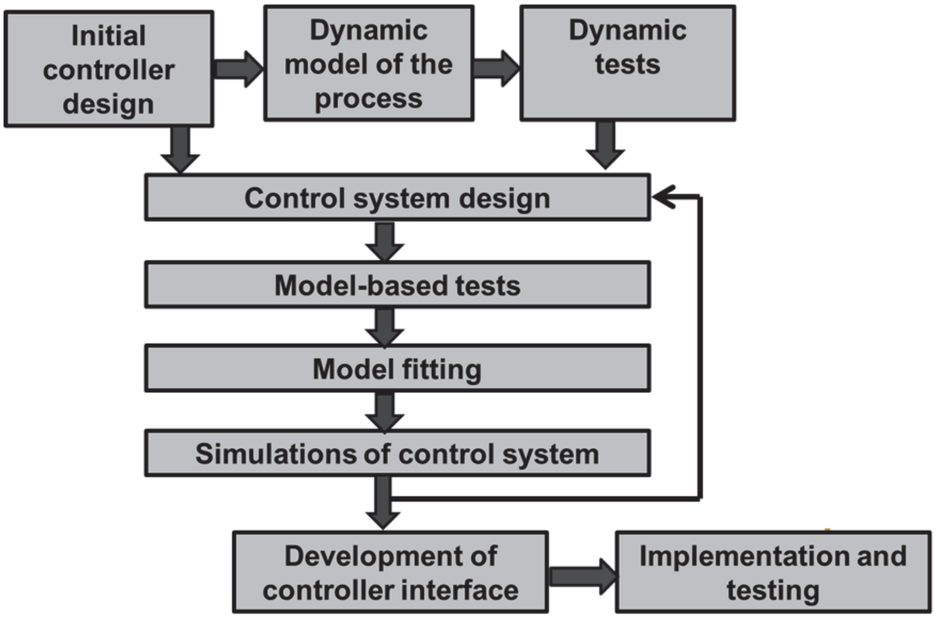

Initial controller design
Dynamic model of the process
Dynamic tests
Control system design
Model-based tests
Model fitting
Simulations of control system
Development of controller interface
Implementation and testing

Figure 40. Steps taken in the development of a MPC controller

With the initial controller design in place, a dynamic model that represents the system was developed. The derivation and validation of the dynamic model for the process being studied was presented in Chapter Two. This dynamic model was used to perform the dynamic analysis of the system, which corresponds to the next step in the block diagram from Figure 40. During this phase, the open-loop responses of the outputs to step changes in all input variables are evaluated, in order to get an idea of the steady-state gains and approximate settling times for each input-output pair. These tests could be performed directly on the plant, but for our system plant tests are expensive and time consuming (not to mention very hard to plan, due to the transient nature of solar irradiation). For these reasons, the dynamic model described in Chapter Two was used in all the plant tests, including the tests for system identification.

After the preliminary testing is done, the design of the control system can be updated and completed. With the information obtained from the dynamic testing, the nominal values and desired operating ranges of the process can be established. The initial controller design from the first step is evaluated and modified if needed. The initial parameters of the MPC, including sampling periods, weighting factors, control and prediction horizons and process constraints, can be selected during this stage. The process of designing a control system is iterative, and during this stage necessary modifications to the control system can be identified and carried out.

After the design of the control system was completed, system identification tests were performed. In this case, the dynamic model of the process was used to perturb the system in order to obtain quality data, used to fit a linear model of the process. Different model structures were evaluated, and the ones that provided the best fits to the validation data were selected and used in the MPC algorithm. After an accurate model was developed, simulations of the MPC were performed, varying key parameters, such as the prediction and model horizons and the output and manipulated variables weights. The parameters that provided the response that best matched the control objectives were selected.

When the final control parameters were selected, the next steps were to develop the controller interface and to implement the controller in the plant and test it. Due to limitations with the available equipment, the control system could not be implemented at the High Flux Solar Furnace (HFSF) as was originally planned. Therefore, the final controller was tested through simulations of the plant, using the dynamic model described in Chapter Two to simulate the plant responses.

The steps followed in the development of the MPC are described in detail in the following sections. Section 4.3 presents the initial controller design and dynamic analysis of the system. Section 4.4 describes the model-based system identification process. The MPC design and simulations are presented in section 4.5. The performance of the MPC was compared to the performance of conventional PID controllers. The design and tuning of the PID controllers is described in section 4.6 and the comparison and results obtained after the MPC implementation are shown in section 4.7.

4.3 Initial Controller Design and Dynamic Analysis of the System

As section 4.2.4 explains, the first step in the development of a controller is to define which of the plant outputs are going to be controlled and which of the inputs are going to be manipulated. It is also important to identify the disturbances in the process and determine if they are measured or not. In addition, the main control objectives need to be spelled out. In order to achieve the goals of the first design stage, it is important to know the system and have a general idea of how the inputs affect the outputs. In order to familiarize the reader with the main characteristics of the system being studied, a description is included in section 4.3.1. Afterwards, the initial controller design is defined in section 4.3.2.

The dynamic model of the process (developed in Chapter Two) was used to perform a more detailed analysis of the system. First, the steady state behavior of the system was studied, in order to select a reasonable nominal operating point and to define the operating ranges. Afterwards, the dynamic behavior of the process within the operating range was explored. The steady state and dynamic analyses are presented in sections 4.3.3 and 4.3.4, respectively.

The information learned through the steady-state and dynamic analyses was used to update the initial controller design and to propose a realizable controller, in line with the dynamic response of the process.

4.3.1 The System

Figure 41 shows a representation of the actual system, implemented at the HFSF. The reactor consists of a 35-cm-long transport tube enclosed by a reflective cavity, which receives concentrated solar irradiation directly through a quartz window. A brush feeder (developed by Woodruff et al. [55]) is used to feed carbon black particles, with an average diameter of 40 nm. The particle flow can be manipulated through a pulse width modulator (PWM), which makes it possible to adjust the speed of rotation of the brush from LabVIEW [107]. Argon gas is used to entrain the particles, and its flow is regulated using mass flow controllers (MFC). Water is fed to the system using a syringe pump and converted to steam inside the reactor. Concentrated solar irradiation is used to drive the endothermic reaction between the particles and steam, to produce synthesis gas (H_2, CO and CO_2). The product streams go through several separation steps, to eliminate the remaining carbon particles and water from the stream before it reaches the analytical equipment. A non-dispersive infrared sensor (NDIR) was used to quantify the amount of CO and CO_2 produced by the reaction. Besides measuring the product composition, the temperature of the reactor is measured though a thermocouple placed in the back wall of the tube. All the instruments are connected to the main Data Acquisition System in LabVIEW, mostly through serial communications. The communication with the mass flow controllers is especially slow, which adds an 8 second delay from the time that the input is changed in the computer to the time that the MFC actually makes the change in gas flow. There is also a 280-second transport delay for the steam flow, from the time the water leaves the syringe pump and reaches the reactor. This transport delay is caused by the low flow of water (on the order of magnitude of 100 µL/min) and the relatively long distance that the water needs to travel before it enters the system (about 0.5 m). There is also a delay of approximately 300 s from the time that the products leave the reactor and their composition is analyzed by the NDIR and registered in the computer. The presence of these time delays (not present in an industrial process) can complicate the controller design and implementation. As was mentioned in section 4.2.4, problems with the equipment at the HFSF barred the implementation and testing of the control system. As a result, and to simplify the controller design, the time delays in the inputs were removed from the process, resembling what it is regularly seen in industrial practice.

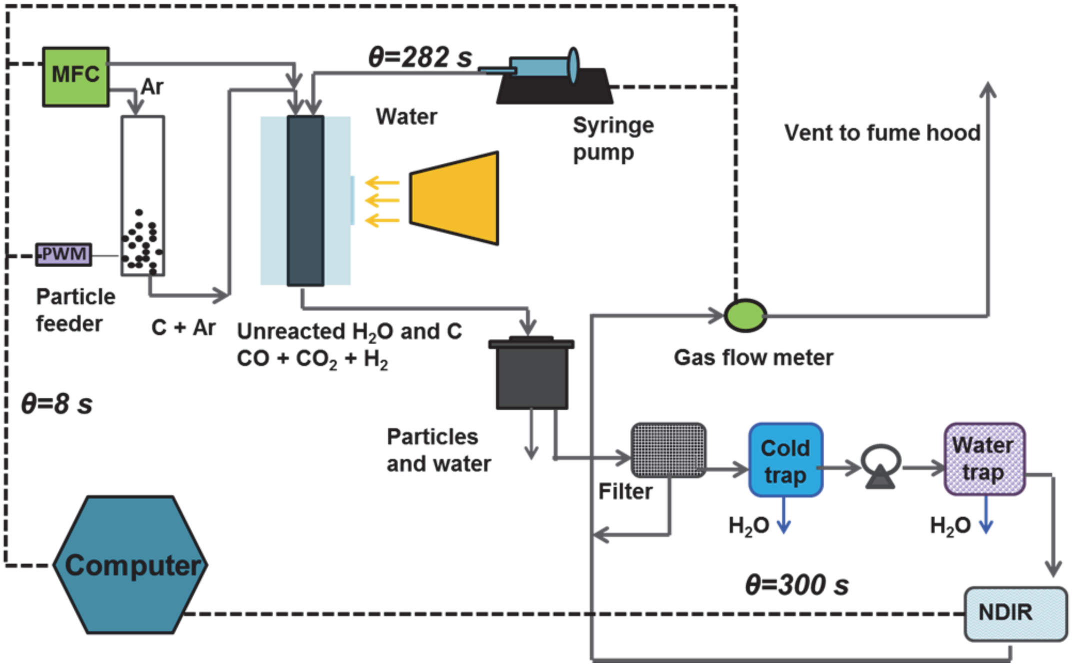


Figure 41. Diagram of the system at NREL

Besides the large dead times in the inputs, this system presented another issue not seen in industrial practice. More than the individual quantities of CO and CO_2 produced by the reaction, an important output to consider is the ratio between the CO and CO_2 in the product stream. This output determines the quality of the synthesis gas: only CO and H_2 are used in the production of liquid fuels through Fischer-Tropsch [108], thus CO_2 is usually considered a byproduct of the reaction and its production should be minimized. The CO_2 byproduct is produced from CO through the water-gas shift reaction [11]:

$$CO + H_2O \leftrightarrow CO_2 + H_2$$

At temperatures above 1200 K, the kinetics of the water-gas shift reaction are slow in comparison to the kinetics of the steam-gasification reaction. As a result, the amount of CO_2 produced is insignificant. An equilibrium study by Müller et al. [62] showed that at atmospheric pressure and temperatures above 1200 K, the gasification reaction goes to completion, producing an equimolar mixture of H_2 and CO. As the temperature decreases, the molar fraction of CO_2 in the product increases, reaching a maximum of approximately 25% at a temperature between 700 and 750 K. Below 700 K, $C_{(s)}$, CH_4, CO_2 and H_2O are the thermodynamically stable species and there is almost no CO and H_2 in the product stream.

We performed a steady-state study of the system, in order to analyze the temperature profiles and the respective CO:CO_2 ratios in the reactor. Figure 42 shows some of the results obtained, for different inlet H_2O:C ratios and reactor residence times. Figure 42.a. shows the steady-state particle temperature profile, for different runs with an inlet particle flow of 80 mg/min and a DNI of 950 W/m^2. As the figure shows, the temperature profile does not vary significantly with the H_2O:C ratio or the gas flow. The reactor reaches its hottest point around the center of the tube, but the region that remains above 1200 K is very short (about 18 disks, or 5 cm). After this hot zone, the temperature drops significantly, in most of the cases below 700 K. This temperature drop causes the CO produced in the hot zone of the reactor to be converted to CO_2. Figure 42.b. shows that decreasing the H_2O:C ratio or increasing the gas flow at the inlet will produce a higher CO:CO_2 ratio in the hot zone. However, independently of how much CO was produced in the hot zone, all the CO is converted to CO_2 as the gas passes through the cooler zones of the tube. The exit CO:CO_2 ratio is below 2, even for the best case scenario.

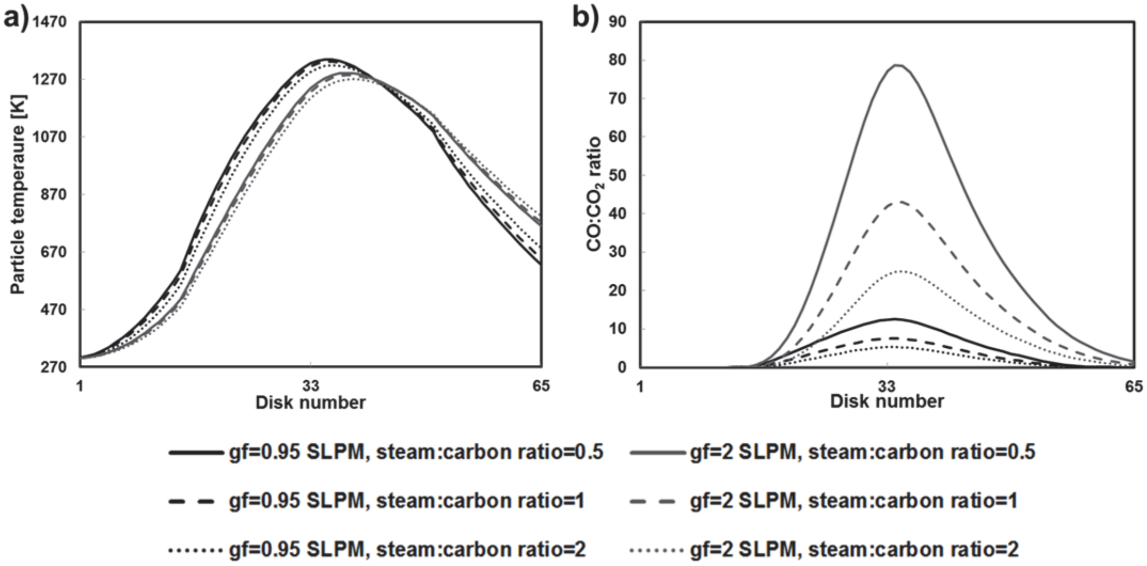


Figure 42. Effect of gas flow and H_2O:C ratio in process outputs, at steady state. a) Particle temperature profile. b) CO:CO_2 ratio profile

In industry, the product gases leave the reactor at high temperatures, and the synthesis gas goes through a quenching stage after it leaves the reactor. The gases are cooled down so quickly that the water-gas shift reaction does not have enough time to occur. It would be impractical to design a control system for a reactor that produces a low quality synthesis gas; therefore, the dynamic model developed in Chapter Two was modified to represent a system with a more uniform radiation distribution, which is more representative of an industrial system. This was done by modifying the correction factor, *β*, in Eqn. (2-10). A value of 1 was given to *β,* for all the disks that were downstream of the center of the tube. Figure 43 shows the particle temperature and $CO:CO_2$ ratio profiles that resulted after implementing this change. As this figure shows, a more uniform radiation profile results in a longer hot-zone in the reactor tube (Figure 43.a.) and thus a smaller fraction of the produced CO gets converted into CO_2, resulting in a higher-quality synthesis gas. The $CO:CO_2$ ratios at the exit of the reactor ranged from 5 to 50, depending on the inlet gas flow and H_2O:C ratio.

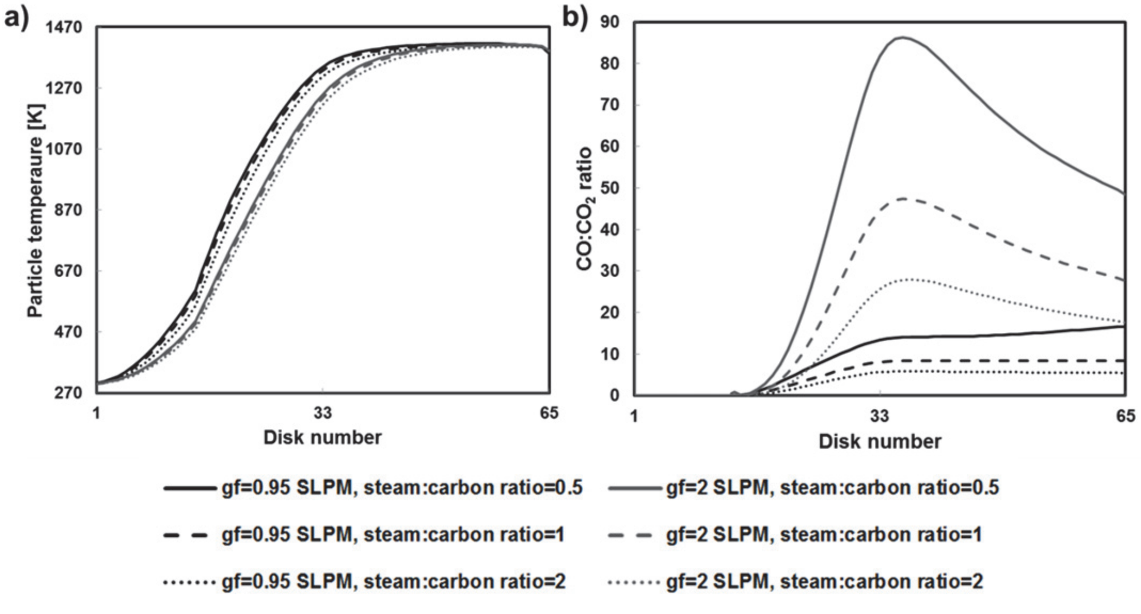


Figure 43. Steady state profiles for a system with uniform radiation distribution, for different gas flows and H_2O:C ratios. a) Particle temperature profile. b) $CO:CO_2$ ratio profile

4.3.2 Initial Controller Design

The system described in section 4.3.1 has three measured outputs: the reactor temperature (measured at center of the back wall of the tube) and the concentrations of CO and CO_2 in the product streams. Combining these measurements with known inputs, it is possible to calculate more relevant outputs for the process, such as the carbon conversion and the fraction of useful synthesis gas in the product. The ratio of CO to CO_2 in the product stream is another output that provides more information than the concentrations by themselves. With this information at hand, it was decided that the output variables of interest were the temperature of the reactor, the fraction of useful synthesis gas (defined as the fraction of H_2 + CO in the product gas), the ratio of CO over CO_2 in the product stream and the carbon conversion.

The inputs of the system are the flow rates into the reactor (inert gas, steam and particle flows), the attenuator position, and the amount of solar irradiation available. From these inputs, the solar irradiation cannot be manipulated and is considered a measured disturbance for the process. During the model validation phase, it was observed that the range over which the particle flow can be modified is too small to affect the outputs. For this reason, the particle flow rate will not be manipulated. The attenuator position will be adjusted through a separate feedback loop, in order to ensure that the reactor does not overheat when sudden increases in DNI take place (e.g., when a cloud moves away from the sun). The measurement of the reactor temperature will be the controlled output in this feedback loop. For more details on this strategy, see section 4.3.5.

The design decisions mentioned above leave only two inputs as potential manipulated variables: the gas flow and the steam flow. In order to achieve control, the number of controlled variables cannot exceed the number of available manipulated variables [104]. Hence, only two outputs can be controlled through the manipulation of the gas flow and the steam flow. Preserving the quality and composition of the product stream can reduce complications in the separation processes downstream of the reactor and was therefore considered more important than maintaining a stable carbon conversion. This means that the synthesis gas fraction and the $CO:CO_2$ ratio were chosen as the controlled variables. Even though the conversion will not be held to a setpoint, it is common practice in MPC to apply range control to the outputs that are not being controlled [104]. In this case, the MPC controller will aim at keeping the conversion above 20%.

After the controlled and manipulated variables have been defined, the control objectives need to be established. The MPC control system should:

(i) Maintain the synthesis gas fraction and $CO:CO_2$ ratio in the product stream as close as possible to their nominal operating values (good set-point tracking).

(ii) Reject the disturbances in solar irradiation faster than a well-tuned feedback strategy.

(iii) Ensure that the conversion does not fall below its lower limit (20%).

(iv) Prevent excessive movement in the manipulated variables and keep the manipulated variables within a physically realizable operating range.

4.3.3 Steady-State Analysis

The dynamic model of the process was used to analyze the system at steady state, with the goal of identifying the nominal values of inputs and outputs and the process operating range.

The nominal value of the DNI was chosen to be equal to 950 W/m^2 to match the average DNI observed during the model validation runs (see Chapter Two for more information). The particle flow was chosen to be 80 mg/min, to match the physical capabilities of the brush feeder in the experimental apparatus. In section 4.3.1, we discussed that increasing the inlet gas flows and decreasing the inlet $H_2O:C$ ratios results in larger $CO:CO_2$ ratios, which would suggest that we want to operate at a high inlet gas flow and a low inlet $H_2O:C$ ratio. However, the effect of the manipulated variables in the other two outputs should also be taken into account. Figure 44 shows the steady-state values for the synthesis gas fraction (Figure 44.a), the $CO:CO_2$ ratio (Figure 44.b) and the carbon conversion (Figure 44.c), for different inlet gas flow and $H_2O:C$ ratios. The large gas flow and low $H_2O:C$ ratios that favor a large $CO:CO_2$ ratio have the opposite effect on the synthesis gas fraction, where large $H_2O:C$ ratios and low gas flow rates result in larger synthesis gas fractions in the product stream. Favorable inputs for the $CO:CO_2$ are not favorable for the carbon conversion either: large gas flow rates mean shorter residence times in the reactor, which result in a lower carbon conversion. In addition, the inlet $H_2O:C$ ratio limits the conversion: regardless of the residence time, at least 50% of the carbon will remain unreacted for an inlet $H_2O:C$ ratio of 0.5, because, after the first 50% of the carbon reacts, there will be no more steam left to carry out the reaction.

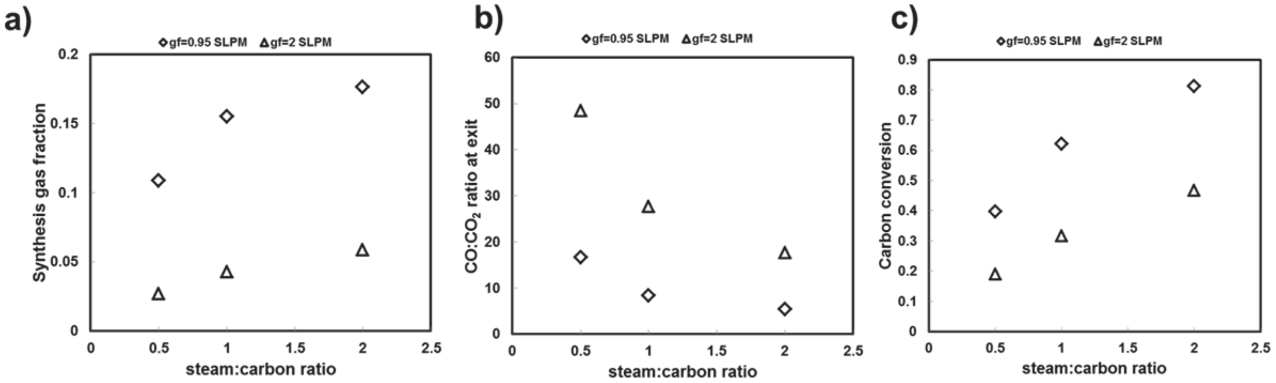


Figure 44. Steady state simulation for different inlet gas flow rates and H2O:C ratios. a) Synthesis gas fraction. b) $CO:CO_2$ ratio at the exit of the reactor. c) Carbon conversion.

Because the manipulated variables have the opposite effect in each of the controlled outputs, a compromise has to be made, between a high synthesis gas fraction and a high $CO:CO_2$ ratio. A balance can be obtained with low gas flow rates and low $H_2O:C$ ratios. Nevertheless, the nominal values cannot be too low; otherwise the controller will not be able to maintain the setpoint at lower DNI values, because it will be operating at the limits of its capacity already. The nominal values for the output and input variables are shown in Table 18.

Table 18.Nominal values for the input and output variables.

Variable	Nominal value
Synthesis gas fraction in product stream	0.1088
CO:CO2 ratio at exit of reactor	16.62
Carbon conversion	0.3963
Gas flow rate	0.95 SLPM
Steam flow	5.97 x 10^{-5} L/min
DNI	950 W/m2

After the nominal values for the inputs and outputs were established, the gas flow rates and steam flow rates required to maintain the setpoints at different DNI values were determined through simulations. The results are shown in Figure 45. As this figure shows, to maintain the setpoints at 700 W/m^2, a gas flow rate of 0.5 SLPM is required. Lower DNI values would require even lower gas flow rates. The system is physically limited to a gas flow rate of 0.5 SLPM, below which the nominal particle feeding rates cannot be achieved. Therefore, it was determined that the system will only operate at DNI values above 700 W/m^2. When the DNI falls below that limit, the system will go into a “stand-by” state, in which the particle flow will be shut off. During the “stand-by” state, the gas and steam flows will be kept at a low value, to allow a quick restart once the DNI goes above 700 W/m^2. In Figure 45, the nonlinearity of the plant can be observed, since the required gas flow and steam flow values for different DNI values cannot be described by a straight line.

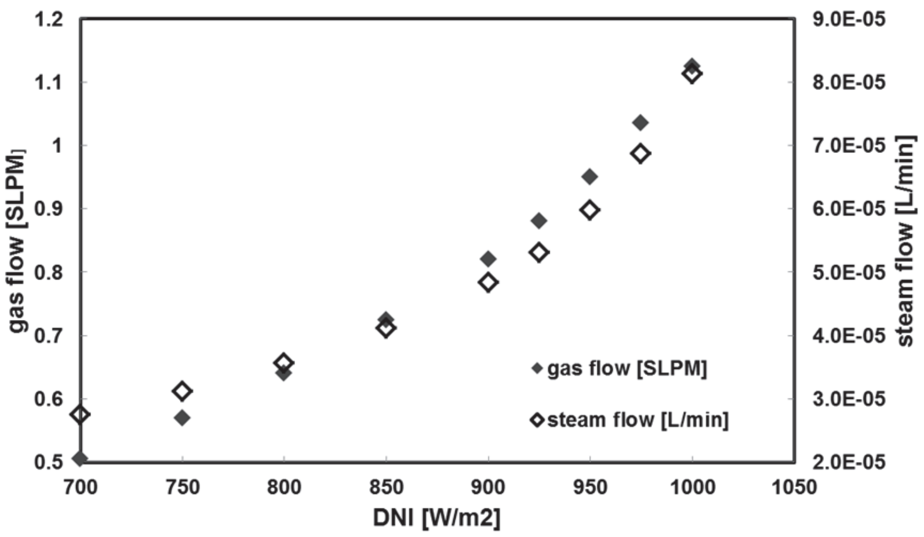


Figure 45. Required gas and steam flows to maintain controlled variables at nominal value, for different DNI values

4.3.4 Dynamic Analysis of the System

The dynamic model was used to study the individual step-responses for all the input-output combinations. The goal is to characterize the steady-state gains and settling times for all input-output pairs. This information will be used to plan the system identification tests.

Figure 46 shows the set of step-responses for the input-output pairs of interest. The first thing to note from the figure is that the outputs respond faster to changes in the gas flow and steam flow than to changes in the DNI (Note the different time scales for the responses to a step change in DNI.). The outputs settle almost immediately at the new steady state when step changes in the gas and steam flow are applied; whereas the outputs take approximately 100 s to reach the new steady state after a step change in DNI is applied. This means that the system will respond faster to changes in the manipulated variables than to disturbances, which is a desired characteristic to achieve good control. The next thing to note are the inverse responses that the CO:CO_2 ratio and the conversion show to a step change in the gas flow. In addition, as it was observed in the steady-state analysis, the effects that the manipulated variables have in the synthesis gas are the opposite than the effect they have in the CO:CO_2 ratio: when the gas flow decreases, the synthesis gas fraction increases, but the CO:CO_2 ratio decreases. In the same way, when the steam flow decreases, the synthesis gas decreases and the CO:CO_2 ratio increases.

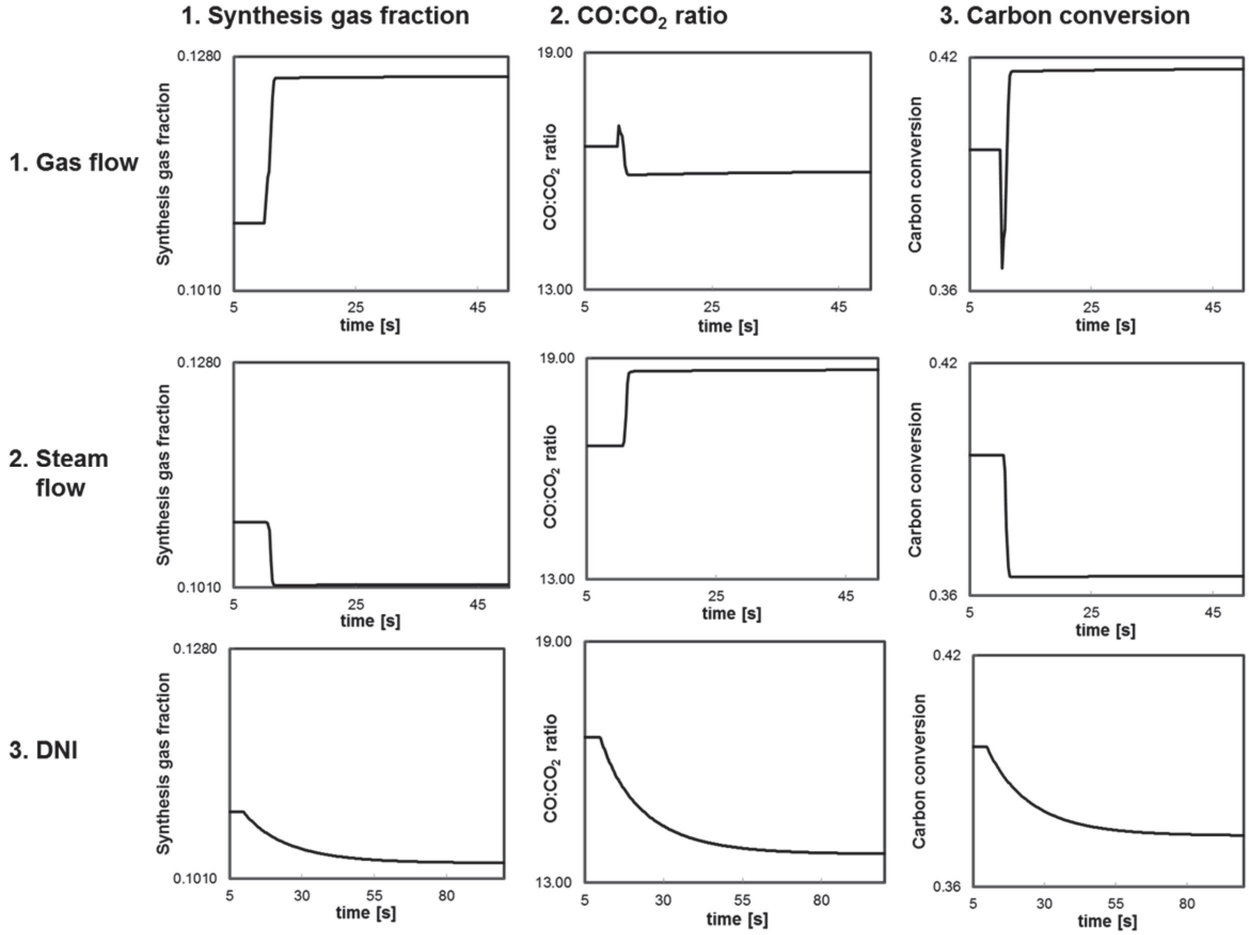


Figure 46. Step responses for each input-output pair

The magnitudes of the step changes in the inputs are shown in Figure 47. The gas flow rate was reduced from 0.95 to 0.85 SLPM, the steam flow was reduced from 5.97x10^{-5} to 5.37 x10^{-5} L/min and the DNI was changed from 950 to 900 W/m^2. All the step changes were applied at time=10 s.

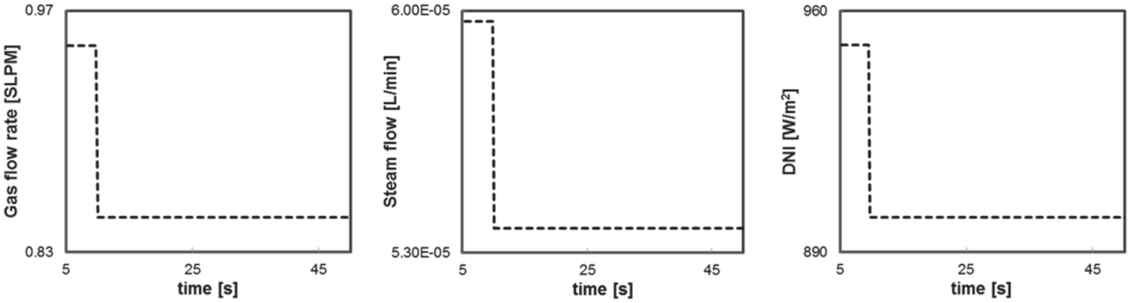


Figure 47. Magnitude of step changes

First-order-plus-dead-time transfer functions were fitted for each input-output pair. The transfer function parameters will aid in the design of the tests for system identification. The matrix of transfer functions for each input-output pair is:

$$G_p(s) = \begin{bmatrix} \frac{-0.1694e^{-0.22s}}{0.76s+1} & \frac{6.42e^{-1.5s}}{0.05s+1} & \frac{-0.2073e^{-1.5s}}{0.74s+1} \\ \frac{1248e^{-0.93s}}{0.10s+1} & \frac{-3.441x10^{5}e^{-0.52s}}{0.83s+1} & \frac{5235e^{-0.93s}}{0.08s+1} \\ \frac{1.230x10^{-4}}{15.4s+1} & \frac{0.0595}{13.8s+1} & \frac{4.721x10^{-4}}{15.4s+1} \end{bmatrix} \tag{4-22}$$

where the first column represents the synthesis gas fraction, the second column represents the CO:CO$_2$ ratio and the third column represents the carbon conversion. The rows of the matrix represent the inputs: gas flow, steam flow and DNI, in that order. From the transfer function matrix we can see that the output responses to changes in the gas and steam flow occur in a different time scale than the responses to changes in DNI because the time constants have different orders of magnitude. We can also recognize that the CO:CO$_2$ ratio responds extremely fast (about 0.25 s before reaching steady state) to changes in the gas flow. The synthesis gas fraction takes about 4 s to reach steady state after a step change in the gas flow rate is applied, while it only takes about 0.5 s when a change in steam flow is applied. This information will be used to design the system identification tests, which will have to occur at frequencies that capture the dynamics in two scales, one for the manipulated variables and the other one for the disturbances.

4.3.5 Proposed Controller Design

The proposed control system is presented in Figure 48. As proposed in chapter one, the control system will use information acquired through the analysis of digitized images of the sky. The development of a one-minute-ahead DNI prediction scheme based on sky images was the topic of Chapter Three. The DNI predictions will be used in the controller at various levels, the first of which is a feedforward component for the attenuator control: if it is predicted that the available solar irradiation will increase suddenly from one minute to the next (i.e. a cloud moves away from the sun), the attenuator position will be manipulated to avoid sudden spikes in the reactor temperature. This feedforward component will be complemented by a feedback loop that controls the reactor temperature. The second level in which the DNI predictions will be used is in a feedforward ON/OFF control system for the particle flow, in which the particle flow will be shut off when the predicted solar irradiation falls below the 700 W/m^2 operating range limit. The third level is as an input for the MPC algorithm, in which the DNI predictions will be used to implement the look-ahead on disturbances that will allow the controller to start acting before the disturbances actually happen. In this work, we will focus solely on the design and development of the third control level, assuming that all the other levels are already in place.

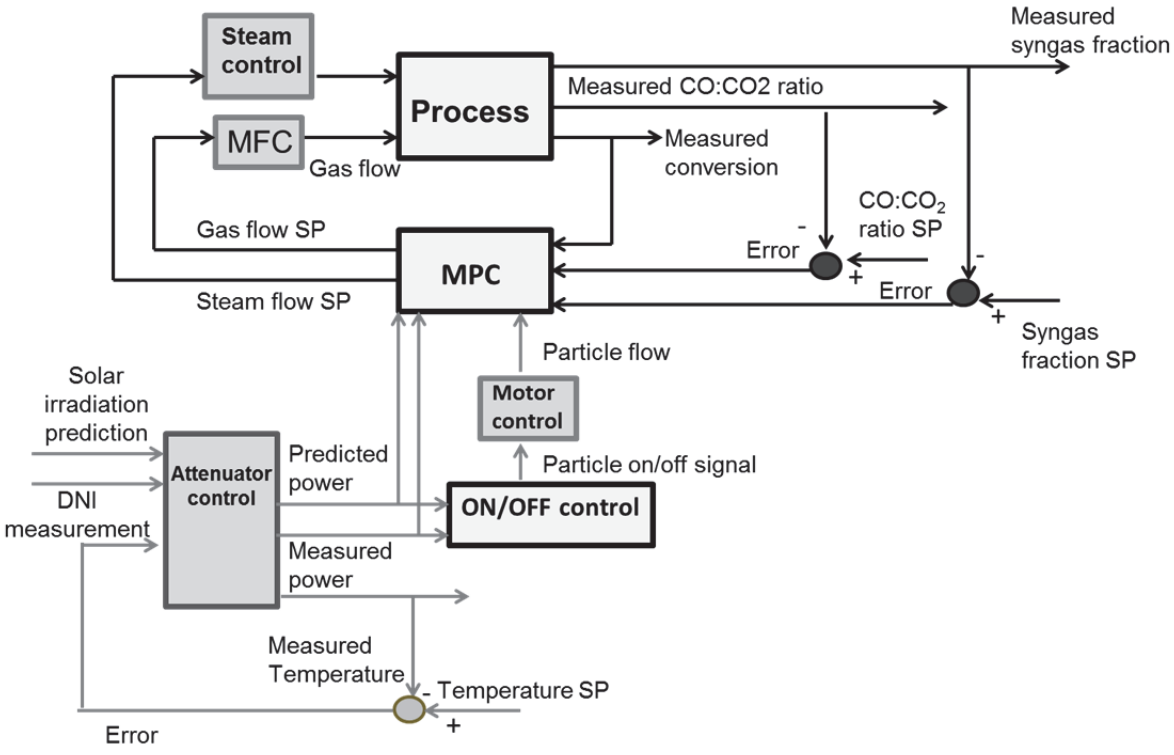


Figure 48. Proposed control strategy

The MPC algorithm will use the DNI predictions as an estimation of future disturbances. The predictions will be compared with actual DNI measurements to check for inaccuracies. The information on the current and future disturbances, along with the current measurements of the outputs, will be used in the MPC calculations. The controller will determine the required manipulation in the gas and steam flow rates and will send the control signals as setpoints for the MFC and the steam controller. After the control moves are implemented in the process, all the controlled outputs will be measured and compared with the model predictions, to correct for model/plant mismatch.

Figure 49 presents a block diagram of the MPC controller. Given a system's current state and measured and future disturbance values, the model of the process is used to estimate the future states of the outputs over the prediction horizon. The predicted outputs are compared with the setpoints to estimate the future errors. The future errors are included in the cost function (presented in Eqn. (4-21)), and an optimization algorithm finds the sequence of manipulations in the gas and steam flow that will minimize the cost function, subject to constraints in the conversion (lower limit of 20%) and the manipulated variables (corresponding to physically realizable limits). The first move in the sequence is used to calculate the control signals, which are sent as setpoints for a mass flow controller that regulates the gas flow into the reactor and a steam flow controller.

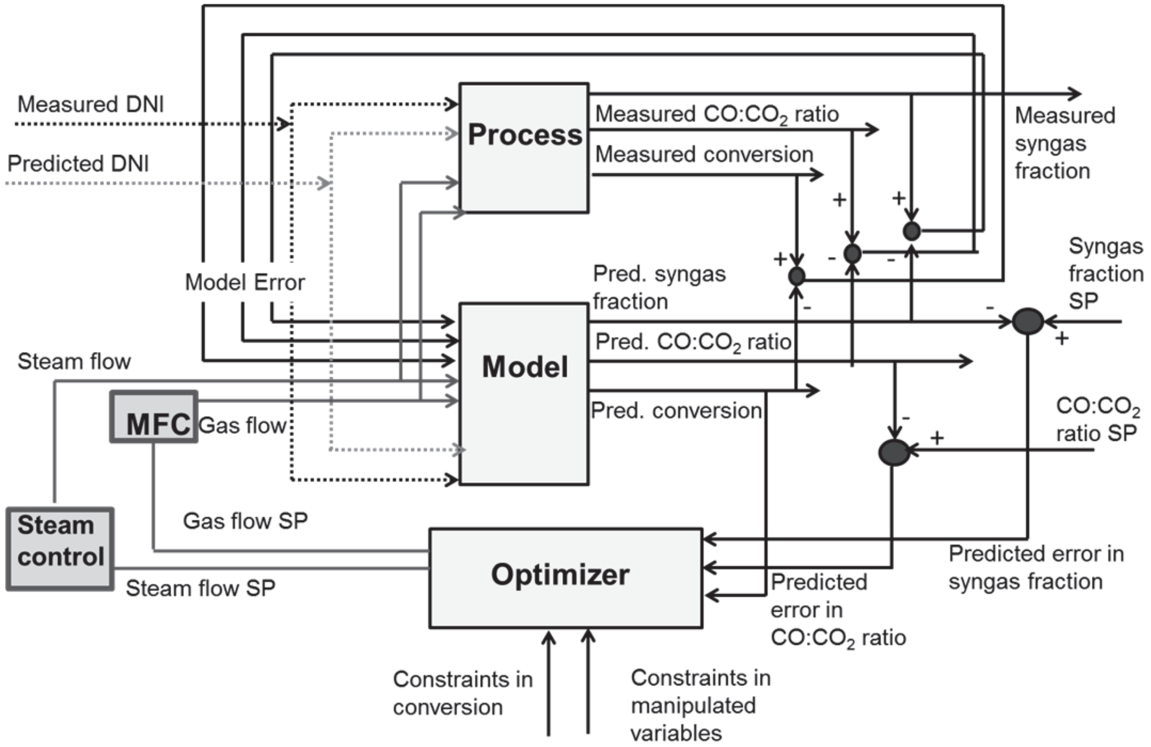


Figure 49. Block diagram for the MPC controller

4.4 System Identification

As section 4.2.2 stated, a model of the process is one of the main components of any MPC strategy. The use of a linear model simplifies the optimization procedure by making it a quadratic programing problem, for which many solution algorithms have been developed. The model derived in Chapter Two is too complex to be used directly in the MPC algorithm, so a linear model needs to be identified.

System identification is a term used to describe the development of empirical steady-state and dynamic models [104]. It is typically carried out using experimental data, acquired through plant perturbations. In our work, however, we will use the dynamic model of the process developed in Chapter Two to simulate plant perturbations and their respective output responses. The design of the plant tests for system identification is described in section 4.4.1, while the development of the linear model is explained in section 4.4.2.

4.4.1 System Perturbation

Two types of signals were used to excite the process to collect data for system identification: a pseudorandom-binary sequence (PRBS) and a sequence of doublet pulses. The system perturbation for model identification was performed around the nominal operating point, with a +/-5% variation in the DNI. The upper and lower limits for the gas and steam flow matched the values required to maintain the controlled variables at their setpoints, at the corresponding upper and lower DNI values. Increasing the magnitudes of the input changes would result in nonlinear behavior that cannot be modeled using a linear model.

A PRBS is a two-level, periodic signal in which a sequence of binary random signals of specified length is used to vary the input between a high and low value. The sequence can be repeated a multiple number of times, and the length of the signal is determined by the dynamics of the process. A PRBS sequence is characterized by two parameters [104]: the length of the switching sequence, N_s, and the switching time, T_s, which determines the minimum length (as a multiple of the sampling time) of the signal before it changes level. The signal repeats itself after N_sT_s units of time. N_s and T_s can be estimated using dynamic process information, through the following formulas [109]:

$$T_s \leq \frac{2.78\tau_{dom}^{L}}{\alpha_s} \quad \textbf{(4-23)}$$

$$N_s \geq \frac{2\pi\beta_s\tau_{dom}^{H}}{T_s} \quad \textbf{(4-24)}$$

where τ_{dom}^{L} and τ_{dom}^{H} represent the low and high estimates of the dominant time constant, β_s is an integer corresponding to the settling time of the process (e.g. for 95% of steady state response, β_s=3) and α_s represents the desired closed-loop response of the system with respect to the open-loop response speed, typically set as α_s=2.

Table 19 shows the parameters used in the PRBS test applied to the system. Two sequences were run: one to obtain data for estimation and the other one to obtain data for model validation. The total length for each sequence was 458 seconds.

Table 19. Parameters used in the PRBS test.

Input	Nominal value	Low value	High value	Ts (s)	Ns
Gas flow (SLPM)	0.95	0.82	1.125	1.5	25
Steam flow (ml/min)	0.0597	0.0484	0.0812	1.5	25
DNI (W/m^2)	950	900	1000	18	25

Figure 50 shows a subset of the input excitation applied to the system. Figure 51 shows the output responses to this excitation. Both sequences start at time=300 s.

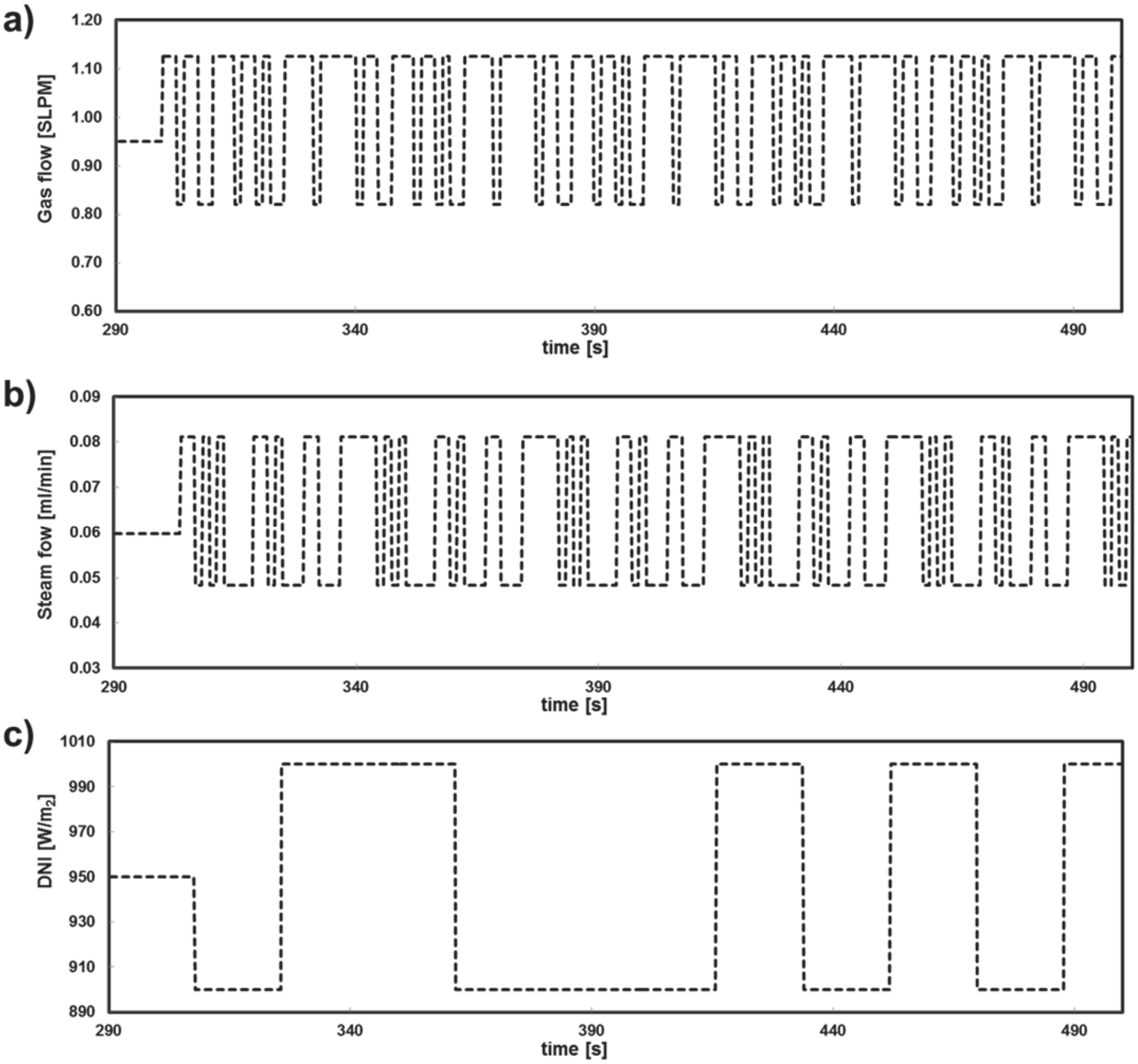


Figure 50. PRBS inputs. a) Gas flow. b) Steam flow. c) DNI

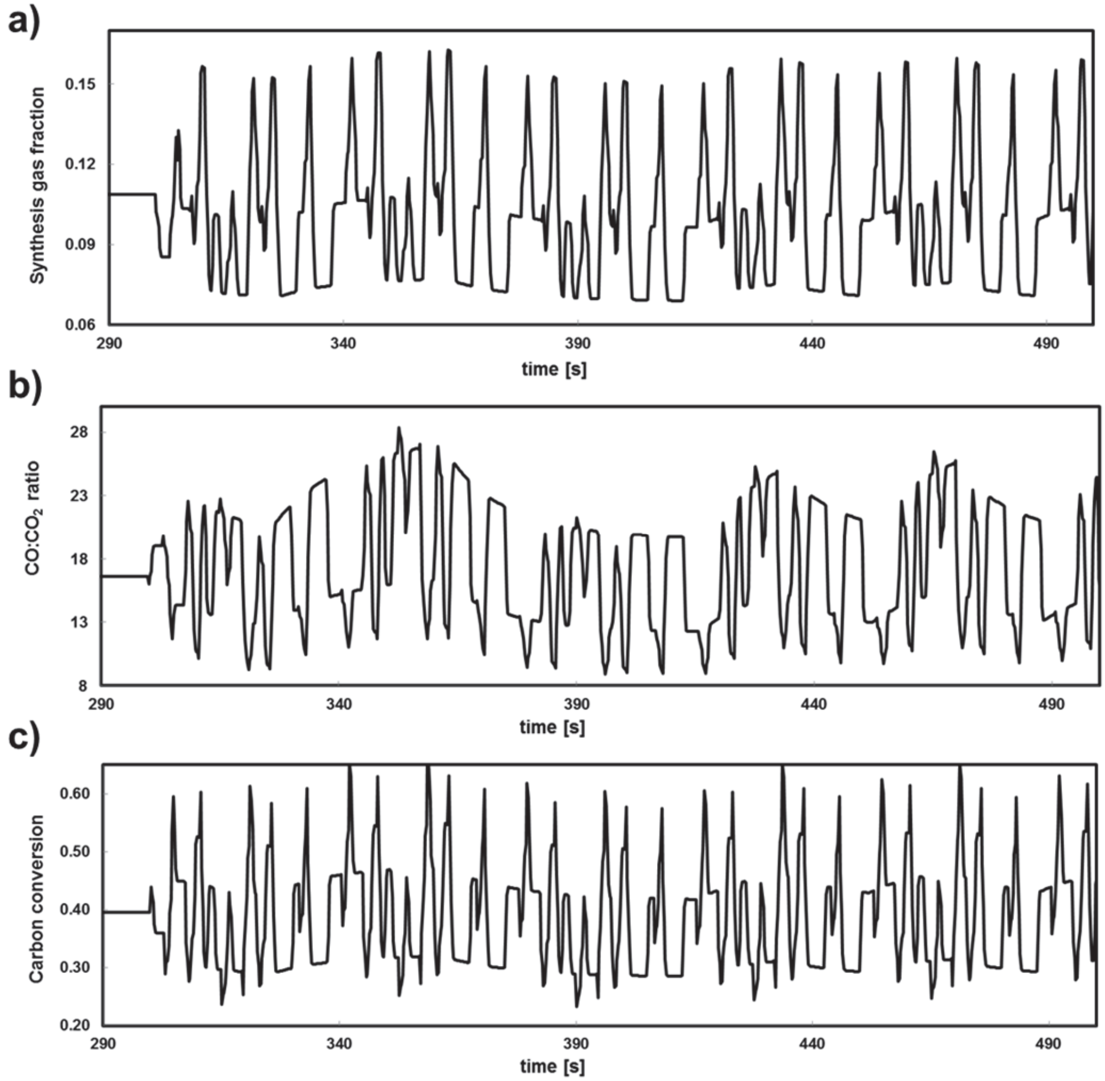


Figure 51. Output response to PRBS inputs. a) Synthesis gas fraction. b) CO:CO$_2$ ratio. c) Carbon conversion

The second form of input excitation was a doublet pulse test, in which a set of pulses are applied to each of the inputs independently. In addition, a simultaneous pulse is applied to subsets of the inputs, to identify interactions between them. Each pulse consists of an upward step change, followed by a downward step change of double magnitude of the first and another upward step to return to the original value. The magnitudes of the step changes were set using a criterion similar to the one used in the PRBS test design. Two different pulse sequences were run (each with steps of different magnitudes); the second sequence will be used for model validation.

Figure 52 presents the pulse sequence applied to the inputs. Figure 53 presents the output responses to this sequence. The data acquired through this sequence will be used for model estimation.

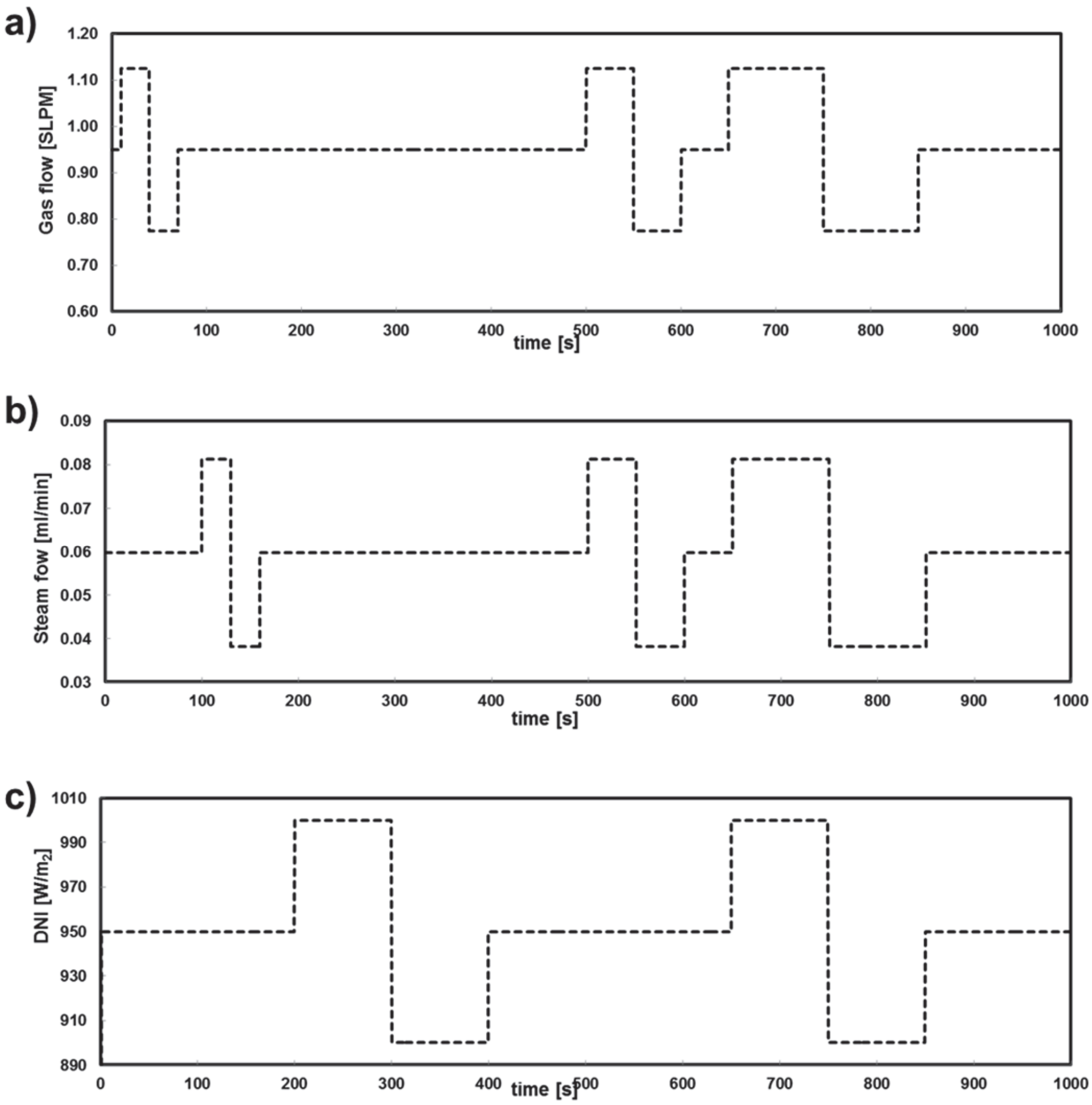


Figure 52. Input pulse sequences. a) Gas flow. b) Steam flow. c) DNI

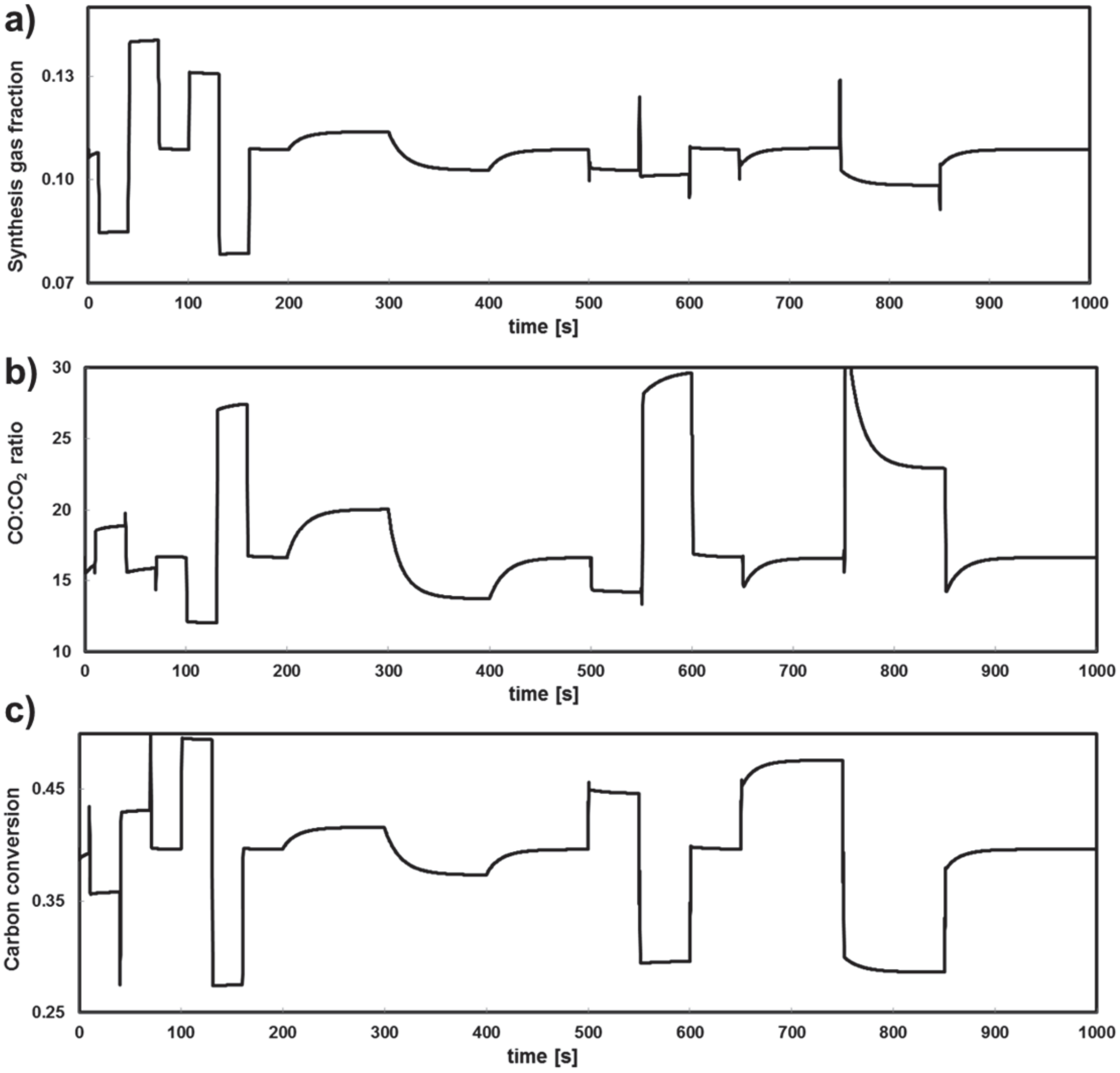


Figure 53. Output response to pulse sequence. a) Synthesis gas fraction. b) CO:CO_2 ratio. c) Carbon conversion

4.4.2 Estimation of Model Parameters

The data acquired through the input excitation tests was used to fit discrete-time, linear models. The first step was to normalize the variables and to convert them into deviation variables, to make the selection of the MPC parameters more straight-forward. The input and output variables were normalized using the following equation:

$$\bar{x} = \frac{x - x_{ss}}{x_{ss}} \qquad \textbf{(4-25)}$$

where $\bar{x}$ represents the input or output variable and the subscript *ss* refers to the nominal steady-state value of that variable.

The models were fitted using MATLAB®'s System Identification Toolbox™ (SIT) [110]. With this toolbox, the estimation of model parameters can be carried out through a graphical user interface (GUI), after providing a data object that contains the data acquired from the process excitation. The graphical user's interface makes it easier to compare different model structures and orders, to select the best performing one.

Beside a plot of the model predictions and the validation data, the SIT provides a fit value, for numerical comparison. The fit (in percentage) is defined as [110]:

$$fit = 100\left(1 - \frac{|y-\hat{y}|}{|y-\bar{y}|}\right) \tag{4-26}$$

where $\hat{y}$ represents the model predictions and $\bar{y}$ represents the mean of *y*.

The preferred model structure was an ARX (Autoregressive with Exogenous Variables), which is a polynomial model defined by the equation:

$$A(z^{-1})y(t) = B(z^{-1})u(t-n_k) + e(t) \tag{4-27}$$

where n_k represents the dead time (in number of sampling instants), *e(t)* is the random error and z^{-1} is the delay operator. $A(z^{-1})$ and $B(z^{-1})$ are polynomials of the form:

$$A(z^{-1}) = 1 + a_1 z^{-1} + \cdots + a_{na} z^{-na} \tag{4-28}$$

$$B(z^{-1}) = b_1 + b_2 z^{-1} + \cdots + b_{nb} z^{-nb+1} \tag{4-29}$$

The orders of the model were determined through trial and error, testing different combinations and selecting the ones that best fitted the validation data. In the end, three models were selected: the best fitting model using the pulse data, the best fitting model using the PRBS data and a model that has the same order as model 1, but uses the data obtained through PRBS to estimate the parameters. The orders and fits for each of the models are presented in Table 20. Model 1 is the one that fitted best the validation data, for all the outputs.

Table 20. **ARX models comparison**

Model	**na**	**nb**	**nk**	**fit for synthesis gas fraction**	**fit for CO:CO$_2$ ratio**	**fit for conversion**

1	[7 0 0;
	0 6 0;
	0 0 3]
		[8 6 2;
		5 5 4;
		5 5 1]
			[1 1 1;
			1 1 1;
			1 1 1]
				87.65	61.85	83.11

2	[7 0 0;
	0 6 0;
	0 0 3]
		[8 6 2;
		5 5 4;
		5 5 1]
			[1 1 1;
			1 1 1;
			1 1 1]
				78.63	49.21	77.75

3	[5 1 1;
	1 4 1;
	1 1 4]
		[4 4 2;
		4 4 2;
		4 4 2]
			[1 1 1;
			1 1 1;
			1 1 1]
				80.88	58.85	78.19

Figure 54 shows a comparison of model predictions and validation data, using model 1. The model provided a good fit to the validation data for case of the synthesis gas fraction and carbon conversion (88% and 83 %, respectively). However, the best fit that could be achieved for the CO:CO$_2$ ratio was 62%.

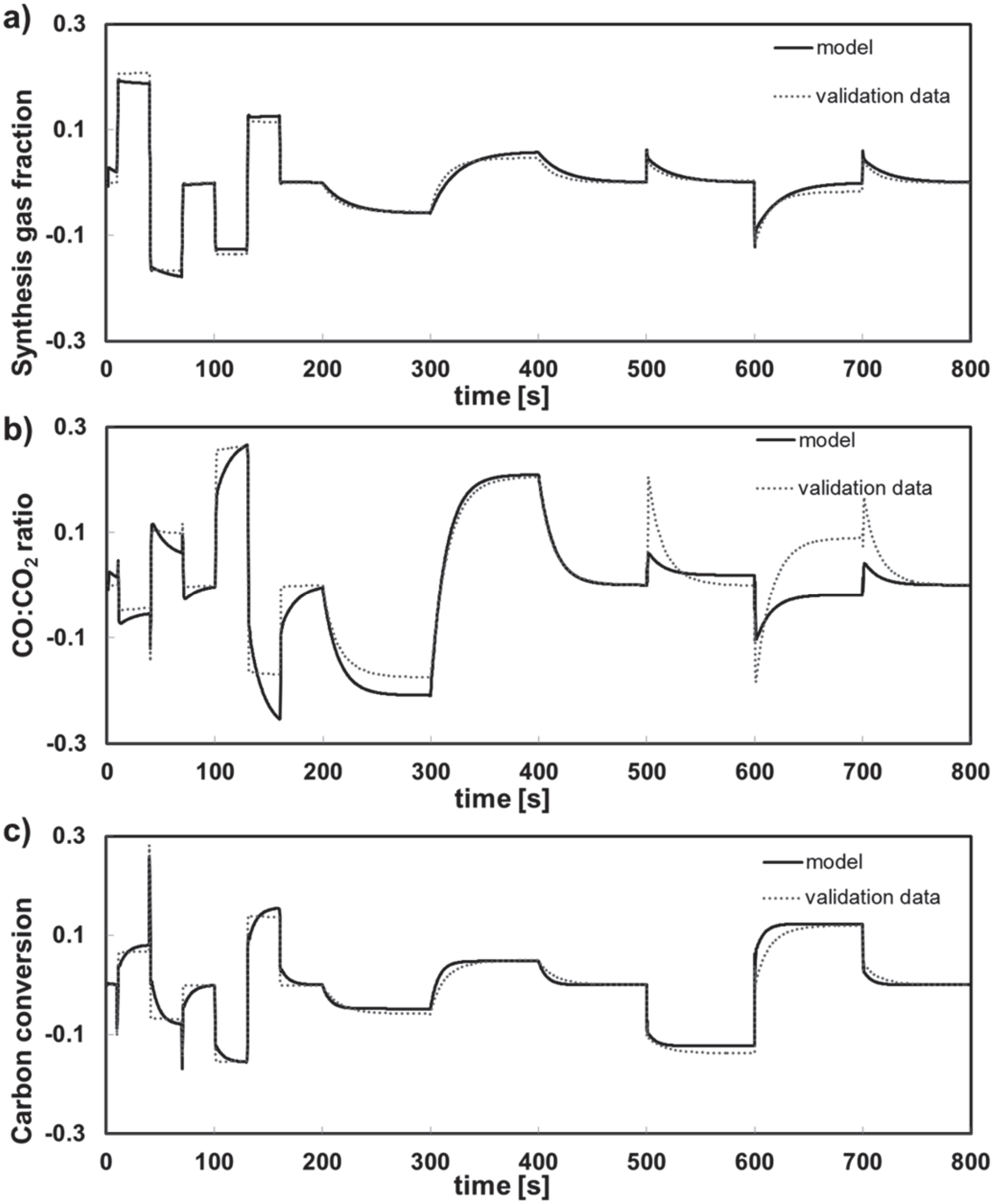


Figure 54. Comparison of model output and validation data for Model 1. a) Synthesis gas fraction. b) CO:CO$_2$ ratio. c) Carbon conversion

Figure 55 shows the predictions of model 2 in comparison with the validation data. Model 2 uses PRBS data for estimation, so data obtain through the second PRBS sequence was used to validate the model. As in model 1, the predictions for the CO:CO_2 ratio are the ones that show the highest amount of error. Figure 56 presents a similar comparison for model 3.

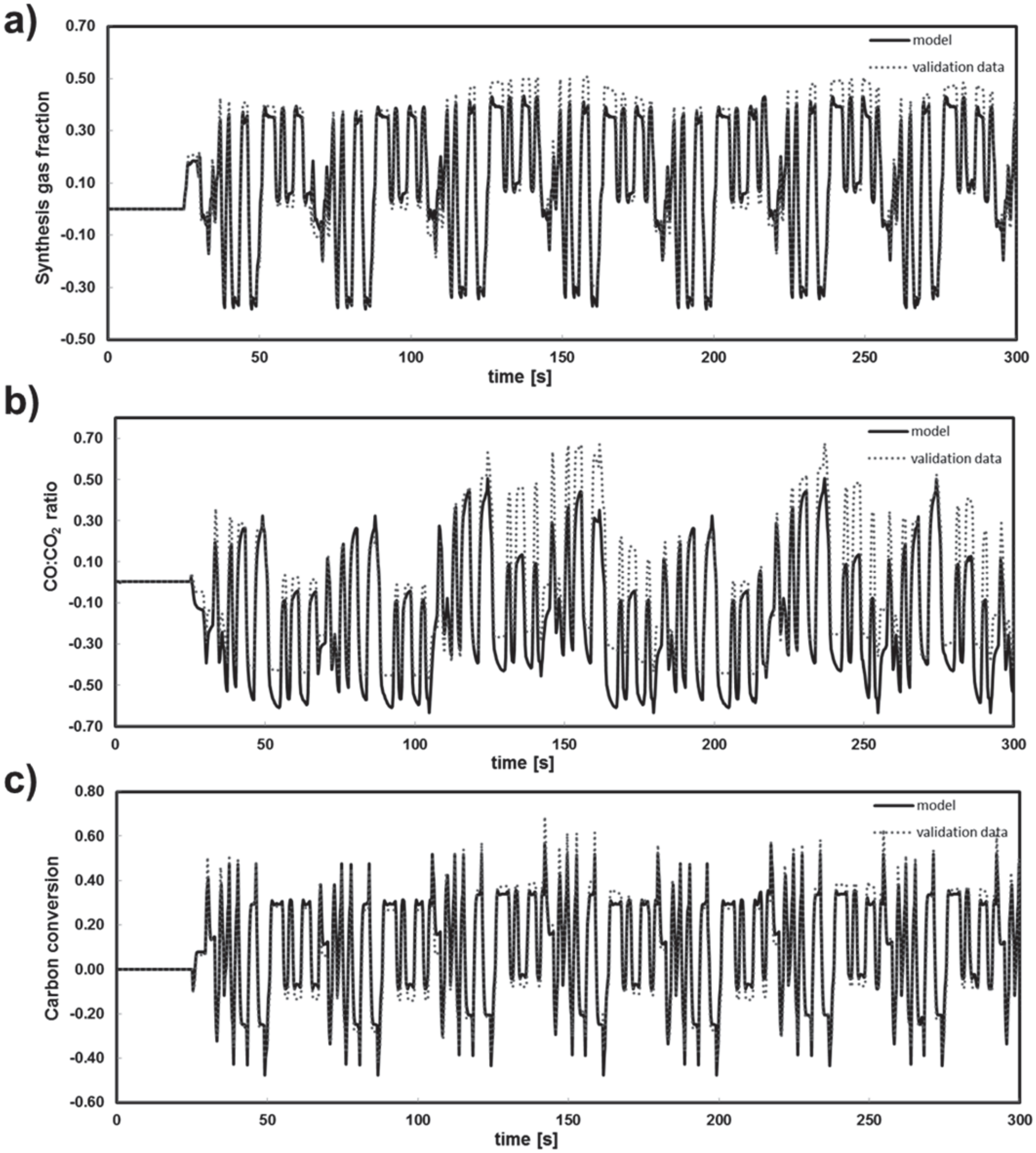


Figure 55. Comparison of model output and validation data for Model 2. a) Synthesis gas fraction. b) CO:CO_2 ratio. c) Carbon conversion

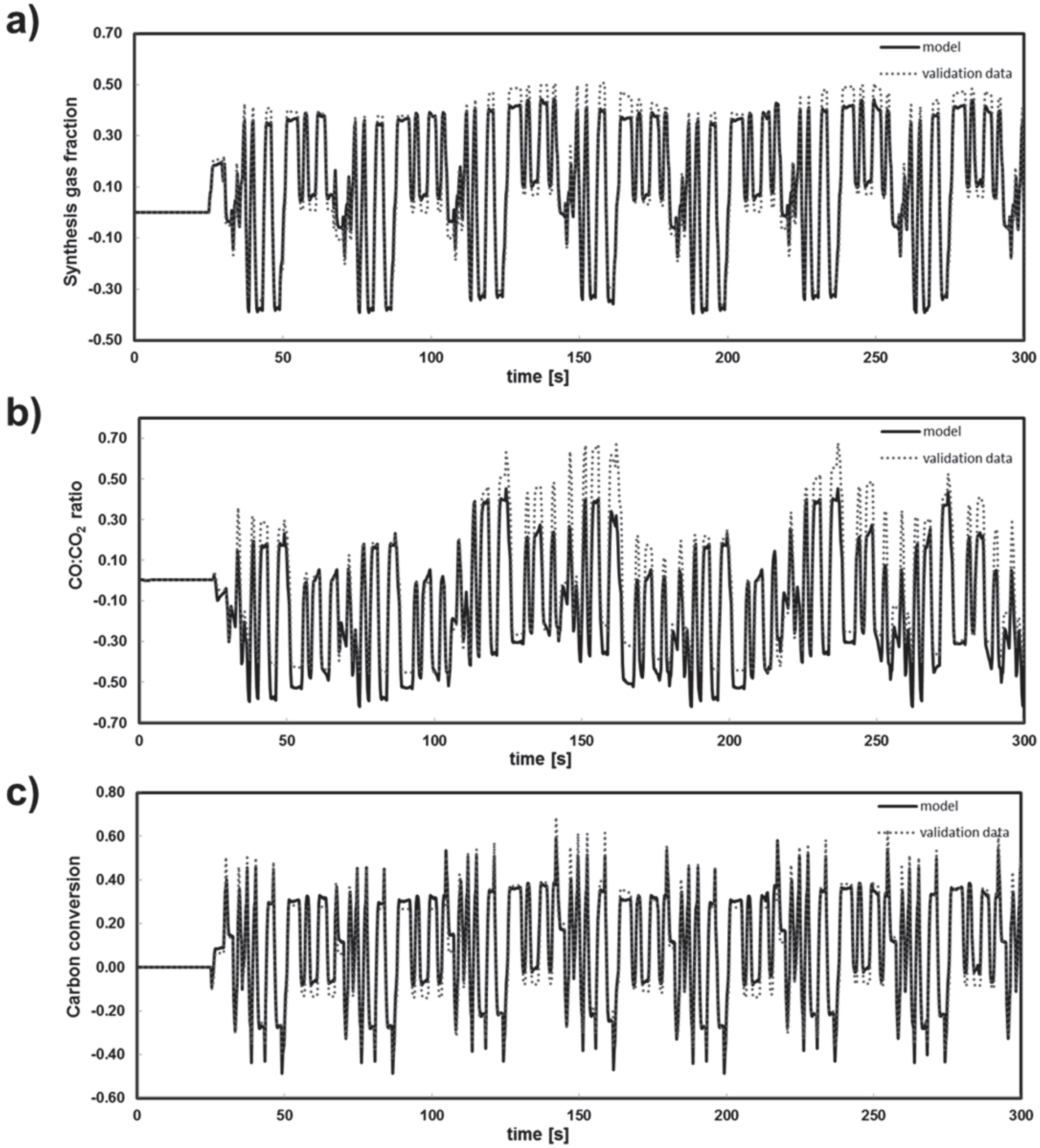


Figure 56. Comparison of model output and validation data for Model 3. a) Synthesis gas fraction. b) CO:CO2 ratio. c) Carbon conversion.

The residuals between the model estimations and the validation data were also taken into account when determining the goodness of the model. Figure 57 shows a normal probability plot of residuals for the outputs, for each of the three models. In most of the cases the residuals follow an approximate straight line, which is the condition to expect for residuals in a normal distribution. For the case of the synthesis gas fraction in model 1, it is clear that the residuals are not normal, and, therefore, the model might not be valid. However, a better fit could not be obtained, so the model selection remains the same. The performance of an MPC controller designed using each of the three models will be studied in section 4.5.1.The model used in the design of the best performing MPC controller will be selected for use in further calculations.

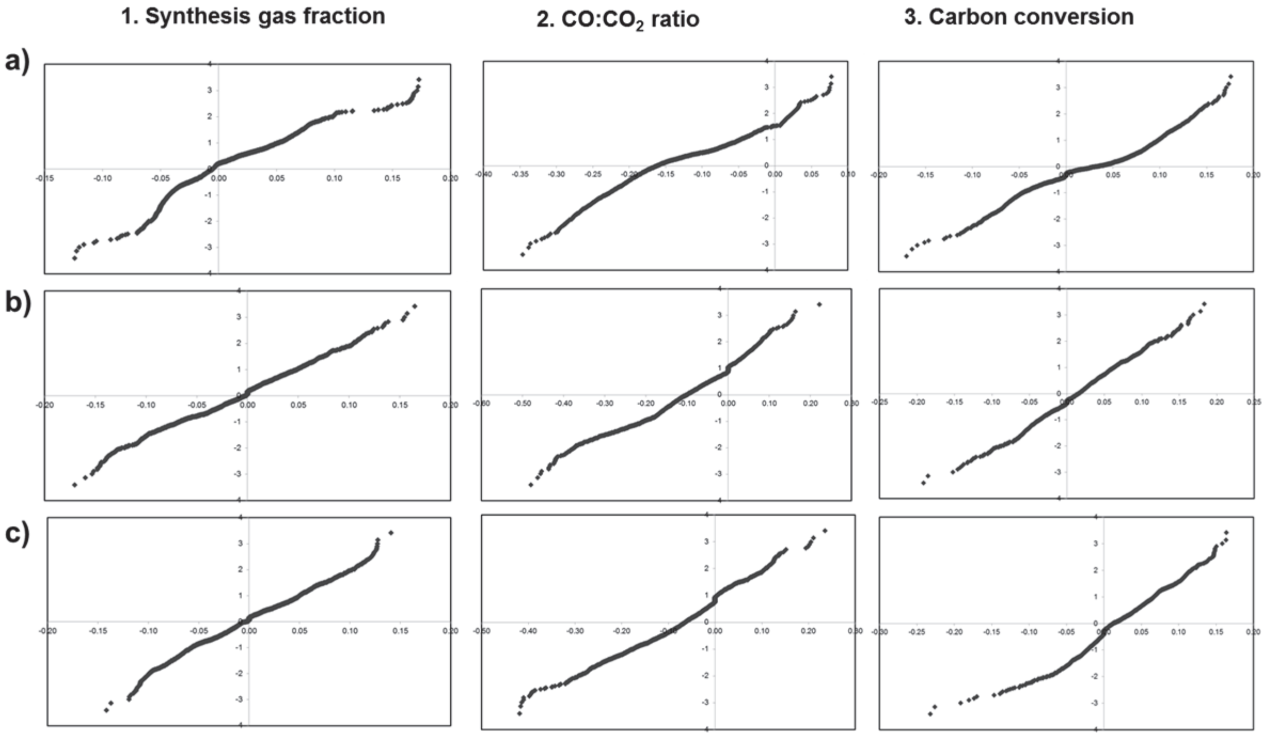


Figure 57.Normal probability plot of residuals for each of the three outputs. a) Model 1. b) Model 2. c) Model 3.

4.5. Model Predictive Controller Design

After a set of linear models were developed, the next step was to implement an MPC controller via simulations and to test its performance in response to changes in DNI. MATLAB®'s Model Predictive Control Toolbox [103] was used to compute the MPC moves. The control signals were sent as arguments for a simulation of the process, consisting of the Differential-Algebraic Equations (DAE) system, described in Chapter Two. The outputs of the simulation are used as measured values for the MPC outputs, used to update the model predictions. The process is repeated again at the next sampling instant.

The Model Predictive Control Toolbox uses a state-space model for state estimation. The models identified in section 4.4 were ARX models, but these models can be transformed into state-space models using the Toolbox. So before the MPC was generated, the models were transformed into the state-space form.

Before the MPC can make any computations, it needs to be initialized. After the initial state of the system is given, the MPC automatically updates the state after each control move is sent to the plant (or to the simulation of the DAE system, in our case).

Because of the fast responses of the system, the control interval was selected to be 0.25 seconds. The initial controller parameters were determined through quick exploratory simulations, using the MPC Toolbox's graphical interface. The selected initial parameters are shown in Table 21.

Table 21. Initial MPC parameters

P	M	Wy	w∆u
15	10	[80 4 0]	[2.5 5]

The output weight on conversion is set to zero because the conversion will not be controlled. The rest of the parameters were varied in order to find the best performing set. The parameter tuning is described in section 4.5.1.

4.5.1 Parameter Tuning

A set of 25 runs with different parameters were evaluated, in order to study the effect that each of the parameters has in the MPC performance. A step change in DNI from 950 to 900 W/m^2 was applied at a time of 20 seconds. The performance of each of the MPC controllers was compared, using the integral sum of the squared error (ISE) as a performance indicator. Table 22 presents a summary of the runs carried out for parameter tuning.

Table 22. Summary of parameter tuning runs.

Run	Model	P	M	Wy	w∆u	Disturbance lookahead?	ISE syngas fraction	ISE CO:CO2 ratio
1	model 1	15	10	[80 4 0]	[2.5 5]	No	8.11E-07	1.17
2	model 2	15	10	[80 4 0]	[2.5 5]	No	4.49E-05	11.66
3	model 3	15	10	[80 4 0]	[2.5 5]	No	2.07E-05	7.64
4	model 1	15	10	[80 4 0]	[2.5 5]	Yes	7.54E-07	1.11
5	model 2	15	10	[80 4 0]	[2.5 5]	Yes	3.87E-05	11.18
6	model 3	15	10	[80 4 0]	[2.5 5]	Yes	1.90E-05	7.28
7	model 1	30	10	[80 4 0]	[2.5 5]	Yes	6.84E-07	1.03
8	model 1	10	5	[80 4 0]	[2.5 5]	Yes	7.56E-07	1.46
9	model 1	15	5	[80 4 0]	[2.5 5]	Yes	6.99E-07	2.47
10	model 1	15	3	[80 4 0]	[2.5 5]	Yes	6.49E-07	2.88
11	model 1	20	10	[80 4 0]	[2.5 5]	Yes	7.29E-07	1.10
12	model 1	40	10	[80 4 0]	[2.5 5]	Yes	6.39E-07	1.17
13	model 1	30	10	[60 4 0]	[2.5 5]	Yes	7.23E-07	1.11
14	model 1	30	10	[100 4 0]	[2.5 5]	Yes	6.59E-07	1.04
15	model 1	30	10	[80 3 0]	[2.5 5]	yes	6.72E-07	1.21
16	model 1	30	10	[80 5 0]	[2.5 5]	Yes	6.95E-07	0.97
17	model 1	30	10	[80 4 0]	[5 5]	Yes	7.27E-07	1.12
18	model 1	30	10	[80 4 0]	[1.25 5]	Yes	6.70E-07	1.02
19	model 1	30	10	[80 4 0]	[2.5 2.5]	Yes	6.77E-07	0.88
20	model 1	30	10	[80 4 0]	[2.5 8]	Yes	7.08E-07	1.65
21	model 1	40	10	[80 4 0]	[2.5 2.5]	Yes	6.27E-07	1.34
22	model 1	30	10	[100 4 0]	[2.5 2.5]	Yes	6.60E-07	0.97
23	model 1	30	10	[100 5 0]	[2.5 2.5]	Yes	6.80E-07	0.84
24	model 1	40	10	[100 5 0]	[2.5 2.5]	Yes	6.33E-07	1.22
25	model 1	30	10	[80 4 0]	[2.5 2.5]	No	7.13E-07	0.87

The first three runs evaluate the effect that using different models has on the MPC performance. Figure 58 shows a comparison of the closed-loop response for MPC controllers designed using models 1, 2 and 3. All three MPC showed smaller deviations from the setpoints than the uncontrolled case. The MPC designed using model 1 (run 1) showed smaller deviations in the outputs when the controller starts to take action. In addition, the controller in run 1 had a lower ISE value than runs 2 and 3, for both outputs. This performance was expected, since model 1 fitted the validation better than models 2 and 3. Therefore, model 1 was selected to be used in subsequent runs.

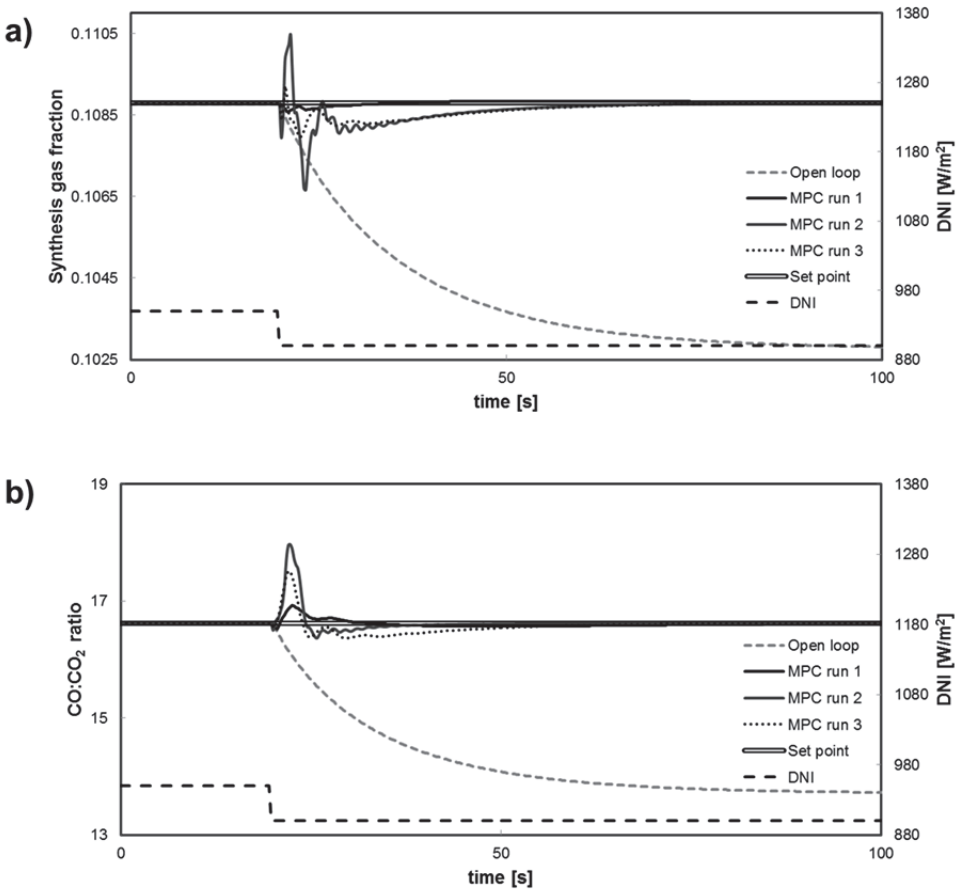


Figure 58. Effect of the model used in the closed-loop response of the MPC. a) Synthesis gas fraction. b)CO:CO_2 ratio

The next step was to compare the effect of adding look-ahead on disturbances. Figure 59 shows this comparison. It was expected that the addition of look-ahead would improve the performance of the controller, since it can take action before the disturbances happen. The controller with look-ahead (run 4) performed better than the controller with no look-ahead, but the improvement was minimal. Look-ahead on disturbances was used in all of the subsequent simulations.

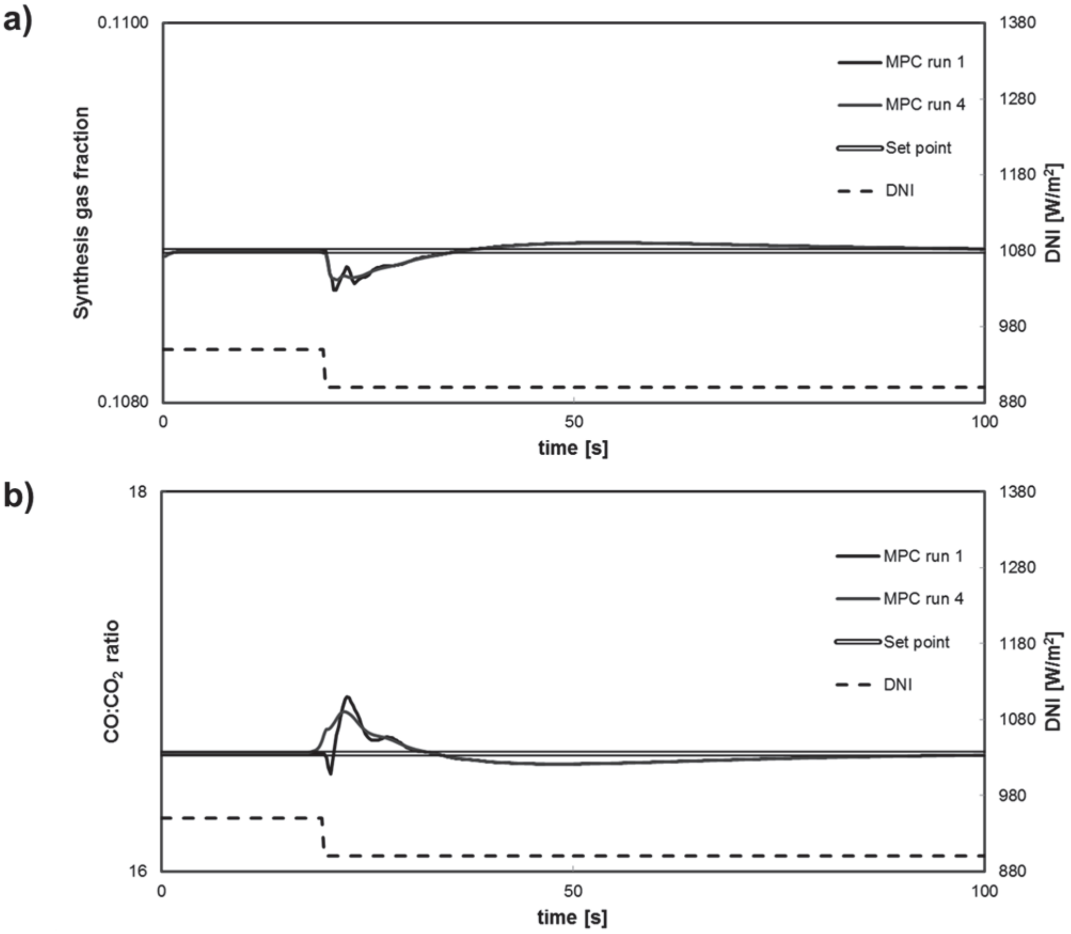


Figure 59. Effect of using look-ahead for disturbances in the closed-loop response of the MPC. a) Synthesis gas fraction. b) CO:CO$_2$ ratio

Figure 60 shows a comparison of MPC controllers with different prediction horizons. This figure shows that changing the prediction horizon does not affect the performance of the setpoint tracking for the synthesis gas fraction. However, as the prediction horizon increases, the closed-loop response for the CO:CO$_2$ ratio becomes slightly worse. This was expected: the CO:CO$_2$ ratio had the worst model fit, which will produce inaccurate predictions. If the predictions are inaccurate, predicting the outputs for a longer horizon will just make matters worse, degrading the controller's performance.

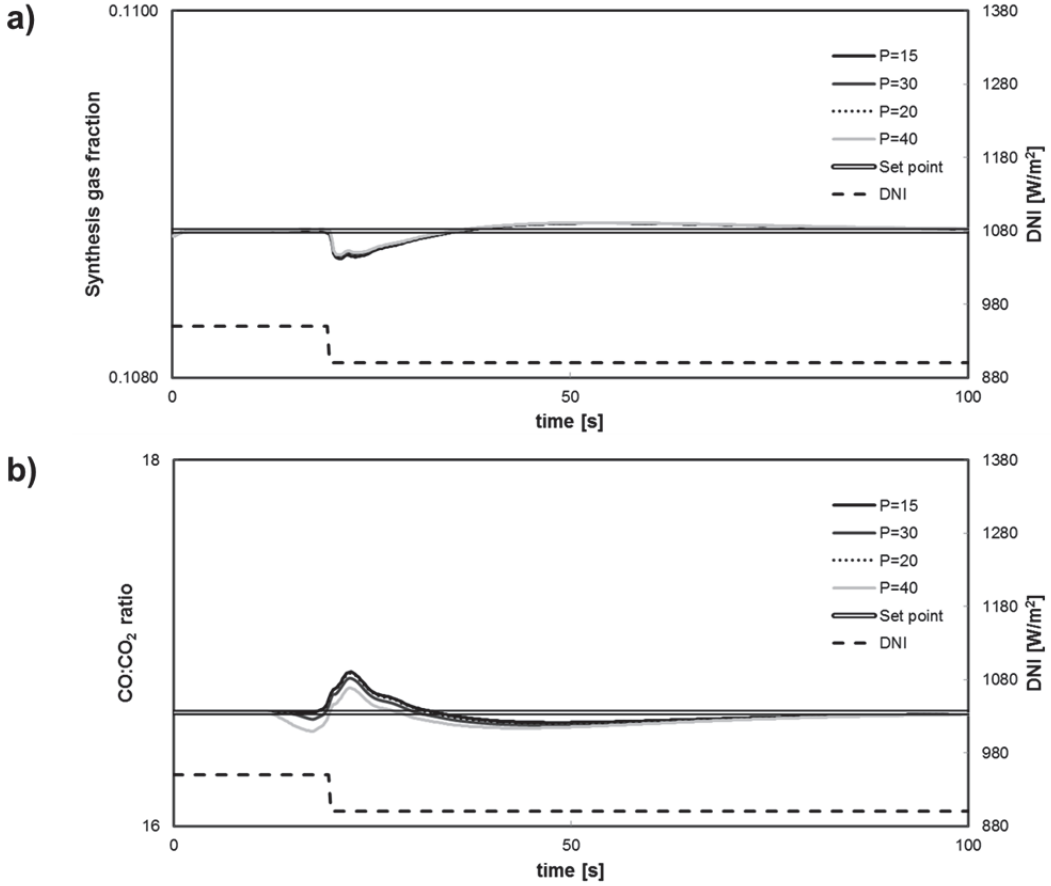


Figure 60. Effect of varying the prediction horizon in the closed-loop response of the MPC. a) Synthesis gas fraction. b) CO:CO$_2$ ratio.

Figure 61 compares the performance of the MPC controller using different control horizons. As in the case of the prediction horizon, the control horizon does not affect the closed-loop response for the synthesis gas; however, a larger control horizon improves the control of the CO:CO$_2$ ratio.

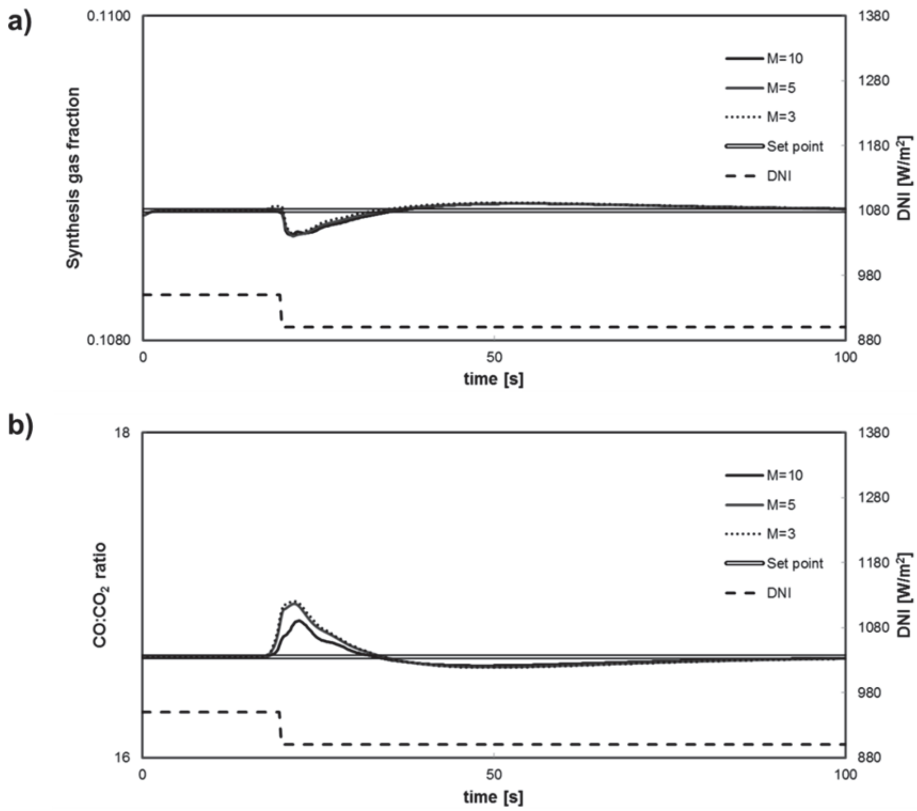


Figure 61. Effect of varying the control horizon in the closed-loop response of the MPC. a) Synthesis gas fraction. b) CO:CO$_2$ ratio

The effect of varying the output weights was also explored. Typically, increasing the output weights generates a more aggressive controller. The effect of varying the weight of the synthesis gas setpoint tracking is shown in Figures 62.a and 62.b, while the effect of varying the weight of the CO:CO$_2$ setpoint tracking is shown in Figures 62.c and 62.d. Figure 62 shows that varying the output weights has no effect in the closed-loop responses.

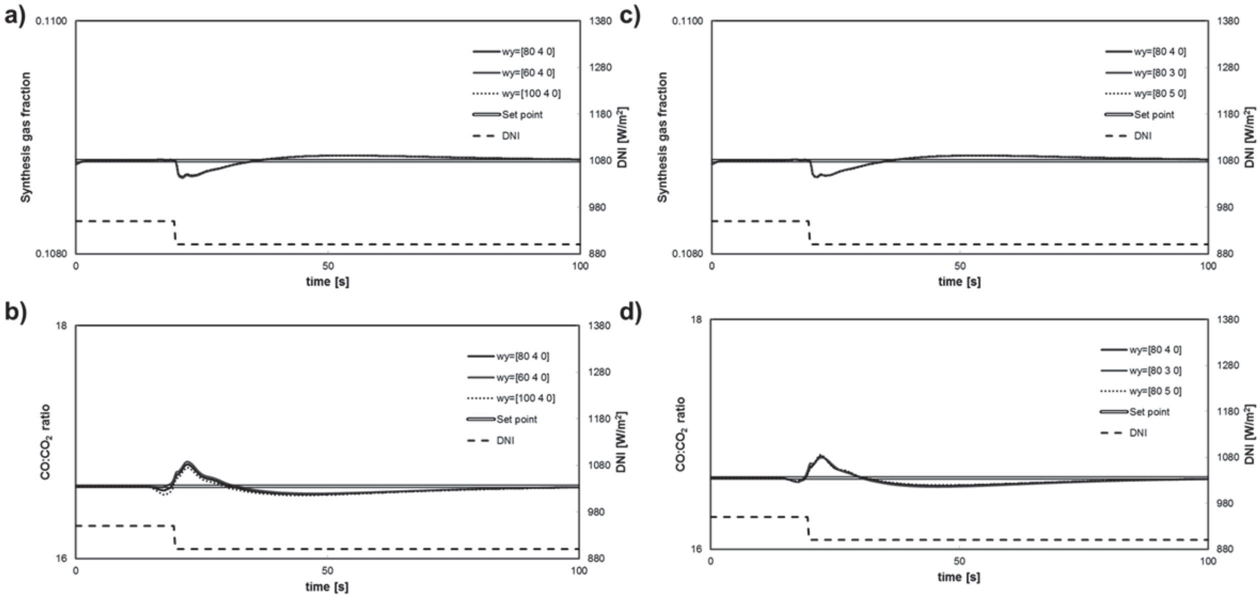


Figure 62. Effect of varying the output weights in the closed-loop response of the MPC. a) Synthesis gas fraction, for changes in synthesis gas fraction weight. b) CO:CO_2 ratio, for changes in the synthesis gas fraction weight. c) Synthesis gas fraction, for changes in the CO:CO2 ratio weight. d) CO:CO_2 ratio, for changes in the CO:CO_2 ratio weight

A similar analysis is performed for the weights of the manipulated variable rates. Figures 63.a and 63.b compare the performance of MPC controllers with different values of the weight for the rate of change of the gas flow. Figures 63.c and 63.d show a similar comparison between controllers with different values for the weights corresponding to the rate of change of steam. Changing the weights in the manipulated variable rates did not have an effect in the closed-loop responses of the controllers, except for the case shown in Figure 63.d, where decreasing the value of the weight on the rate MPC for steam flow improves the performance of the controlled CO:CO_2 ratio response.

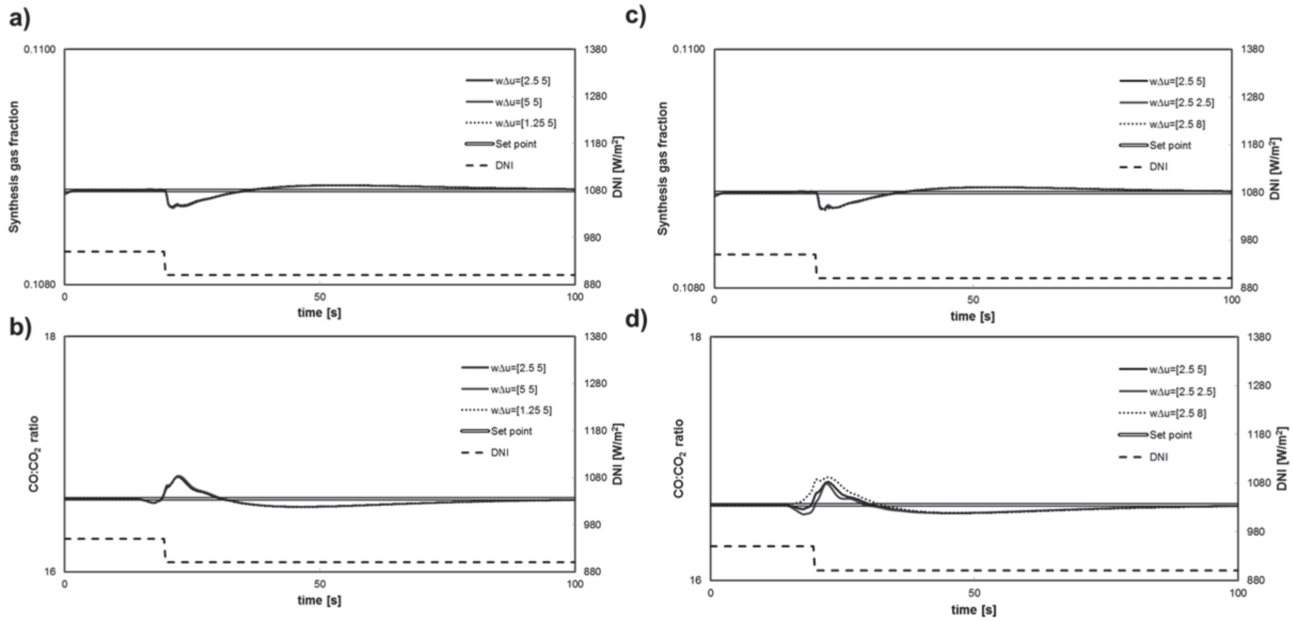


Figure 63. Effect of varying the weights of the manipulated variable rates in the closed-loop response of the MPC. a) Synthesis gas fraction, for changes in the weight corresponding to the rate of change of gas. b) CO:CO_2 ratio, for changes in the weight corresponding to the rate of change of gas c) Synthesis gas fraction, for changes in the weight corresponding to the rate of change of steam. d) CO:CO_2 ratio, for changes in the weight corresponding to the rate of change of steam.

The best overall performance for the tuning runs was obtained with the controller in run 19, which has the parameters presented in Table 23.

Table 23. MPC parameters after tuning runs

P	M	Wy	wΔu
30	10	[80 4 0]	[2.5 2.5]

The performance of the tuned MPC was compared to the open-loop response. As Figure 64 shows, for a step change in DNI from 950 to 900 W/m^2, the MPC controller provides good setpoint tracking for both controlled outputs, to the point that deviations from the setpoint are barely visible in the plots.

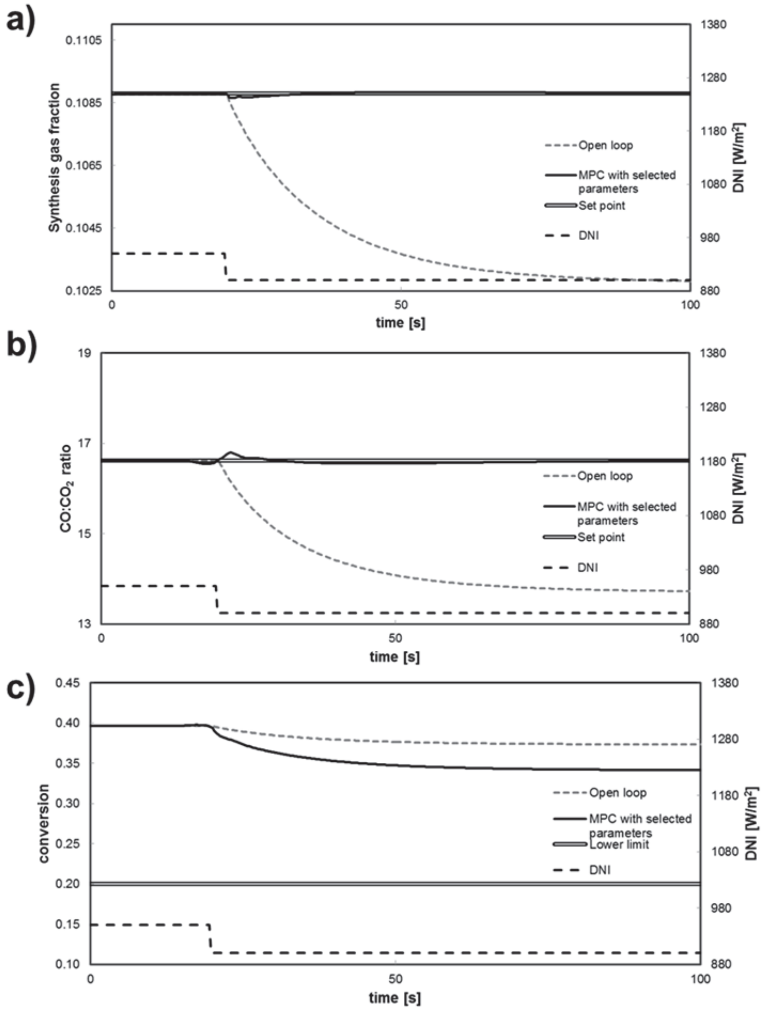


Figure 64. Comparison between tuned MPC controller and uncontrolled system. a) Synthesis gas fraction. b) CO:CO2 ratio. c) Conversion

4.5.2 MPC Robustness

A desirable characteristic of a controller is to show a good disturbance rejection, regardless of the size of the DNI disturbance, as long as the disturbance remains within the operating region. The plant is highly nonlinear; thus, it is expected that a single linear controller designed around the nominal operating point will have some problems. Therefore, the tuned MPC controller was tested through a sequence of DNI step changes that covered the operating range. Figure 65 shows the controlled output responses obtained. As apparent in the figure, the controller tracks the setpoints well for DNI changes around the nominal value of 950 W/m^2.

However, the closed-loop response becomes oscillatory when the DNI reaches values of 850 W/m^2 or below. When the DNI falls to 700 W/m^2, the controller shows sustained oscillations and only seems to be able to return to the setpoints after the DNI goes back up to 950 W/m^2. This behavior must be corrected, since it can lead to instabilities that can compromise the safe operation of the system or to actuator wear.

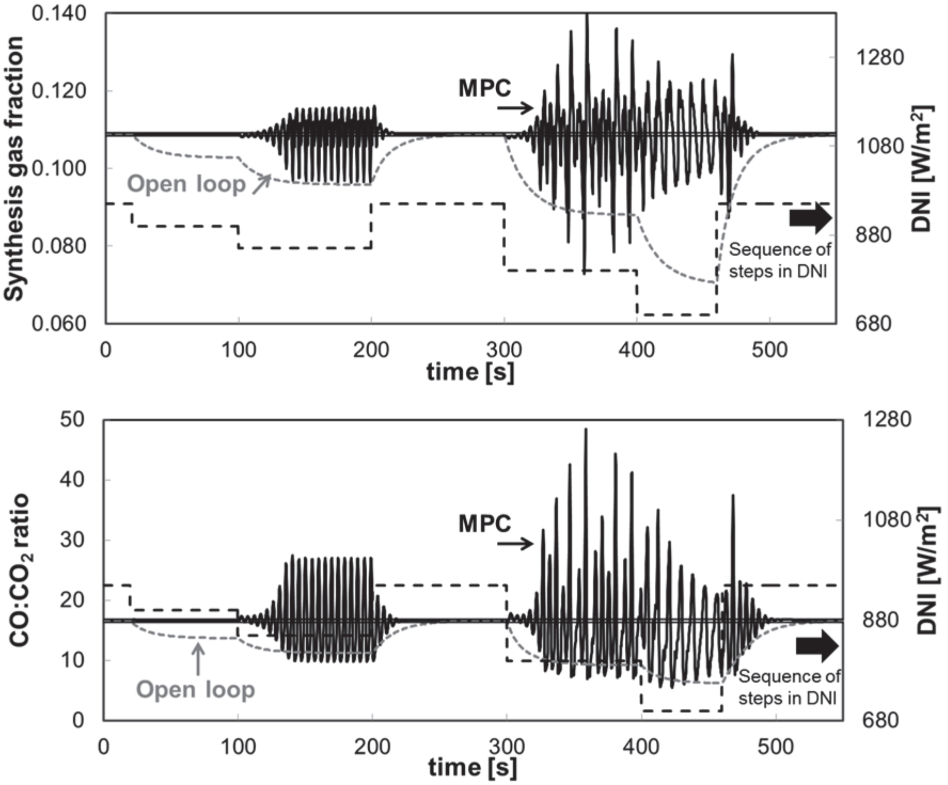


Figure 65. Controlled responses to sequence of step changes in DNI, using tuned MPC. a) Synthesis gas fraction. b) CO:CO$_2$ ratio

. These oscillations were dramatically reduced by re-tuning the MPC controller and testing if more conservative parameters produce an acceptable performance. An alternative and more complex approach to achieve the same goal would be the implementation of a linearized scheduled MPC algorithm, which selects the most appropriate linear model among several models fitted around different operating points.

Following the first approach, the controller was made more conservative by increasing the weights of the manipulated variables rates until the response became less oscillatory. The conservative best-performing MPC was implemented using the parameters presented in Table 24.

Table 24. Parameters of robust MPC

P	M	wy	w∆u
30	10	[80 4 0]	[65 80]

Figure 66 shows the closed-loop responses achieved with the re-tuned controller, for the same sequence of step changes in the DNI as in Figure 65. The deviations from the setpoints are minimal when compared to the open-loop response, even if the DNI falls below the range for which the model was fitted. Thus, this controller can be considered robust and can be used throughout the operating range without resulting in an unstable system.

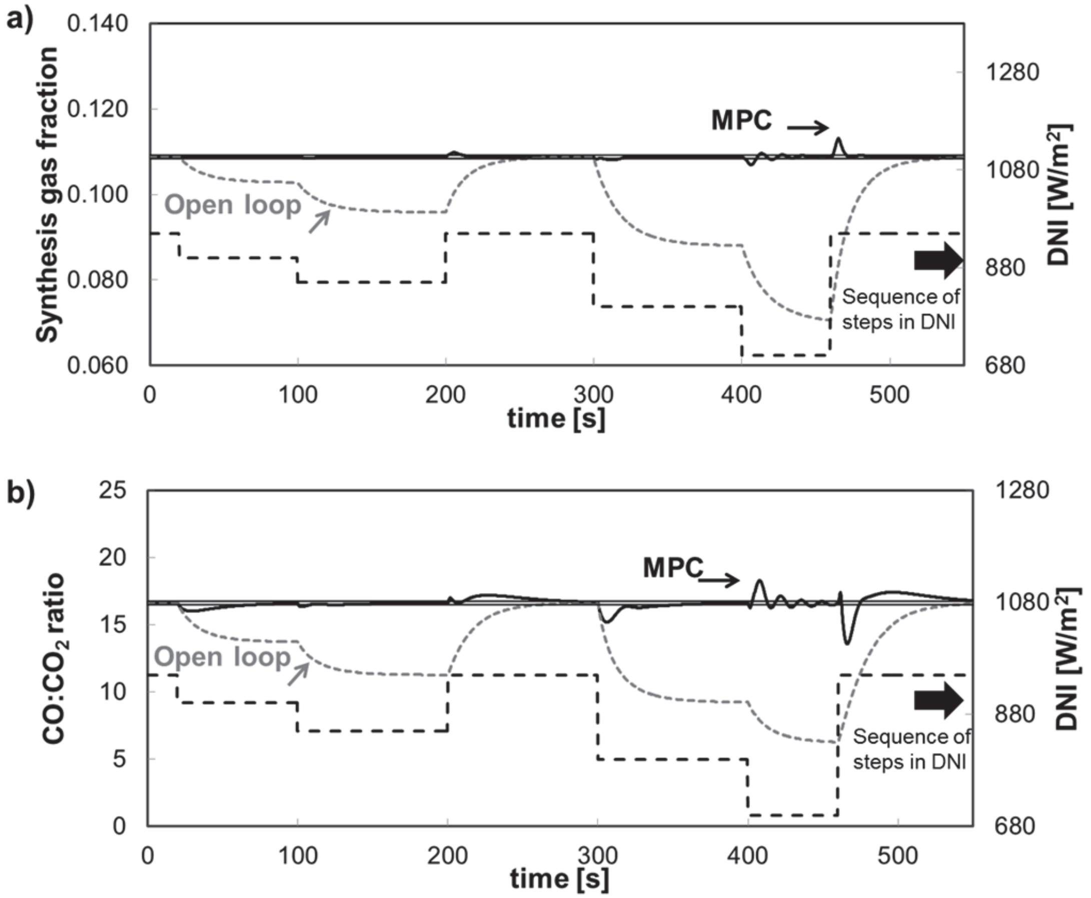


Figure 66. Controlled responses to sequence of step changes in DNI, using re-tuned MPC. a) Synthesis gas fraction. b) CO:CO$_2$ ratio

4.5.3 Integration with Vision-Based Solar Irradiation Predictor for Disturbance Estimation

The image-based solar irradiation predictor developed in Chapter Three was incorporated into the MPC algorithm, using the one-minute-ahead DNI predictions as estimations of the future disturbances. Instead of using the current measured DNI value for the output estimation (zero-order hold), the MPC algorithm uses the estimations of the future disturbances, predicted as described in Chapter Three. It is expected that an MPC algorithm that uses estimated values for the measured disturbances will outperform a controller that uses zero-order hold for disturbances in the output estimation. However, this strongly depends on the quality of the predictions, since erroneous disturbance predictions will result in the controller computing incorrect control signals. The solar irradiation predictor developed in Chapter Three was unable to recognize sharp changes in DNI before they happened; therefore, it is important to compare the performance of this controller to the performance of other MPC controllers that use perfect predictions, to better understand the effect of prediction errors. In addition, we studied the performance of a controller that uses only the current measured disturbance (no look-ahead). The comparison of all these MPC strategies is the topic of section 4.5.4.

The block diagram in Figure 67 presents the strategy followed in the implementation of the look-ahead for disturbances in the MPC controller. Images of the sky are acquired using a total sky imager (TSI-880). The images are analyzed using a MATLAB® script developed for this purpose, and the characteristics of the image are combined with current measurements of the system, acquired from the Solar Radiation Research Laboratory (SRRL) database. This information is sent to the solar irradiation predictor, which estimates the values for the one-minute-ahead DNI. The output of the solar irradiation prediction serves as future disturbances estimation for the MPC block. The MPC algorithm then uses the predictions, in combination with current measurements of the outputs and DNI, to compute the control signals that will best reject the disturbances. The manipulations are sent to the process using a mass flow controller to regulate the gas flow and a steam flow controller to regulate the flow of steam into the process. Measurements of the outputs are sent to the MPC algorithm after the manipulations are applied. The MPC is updated and the calculation algorithm starts again.

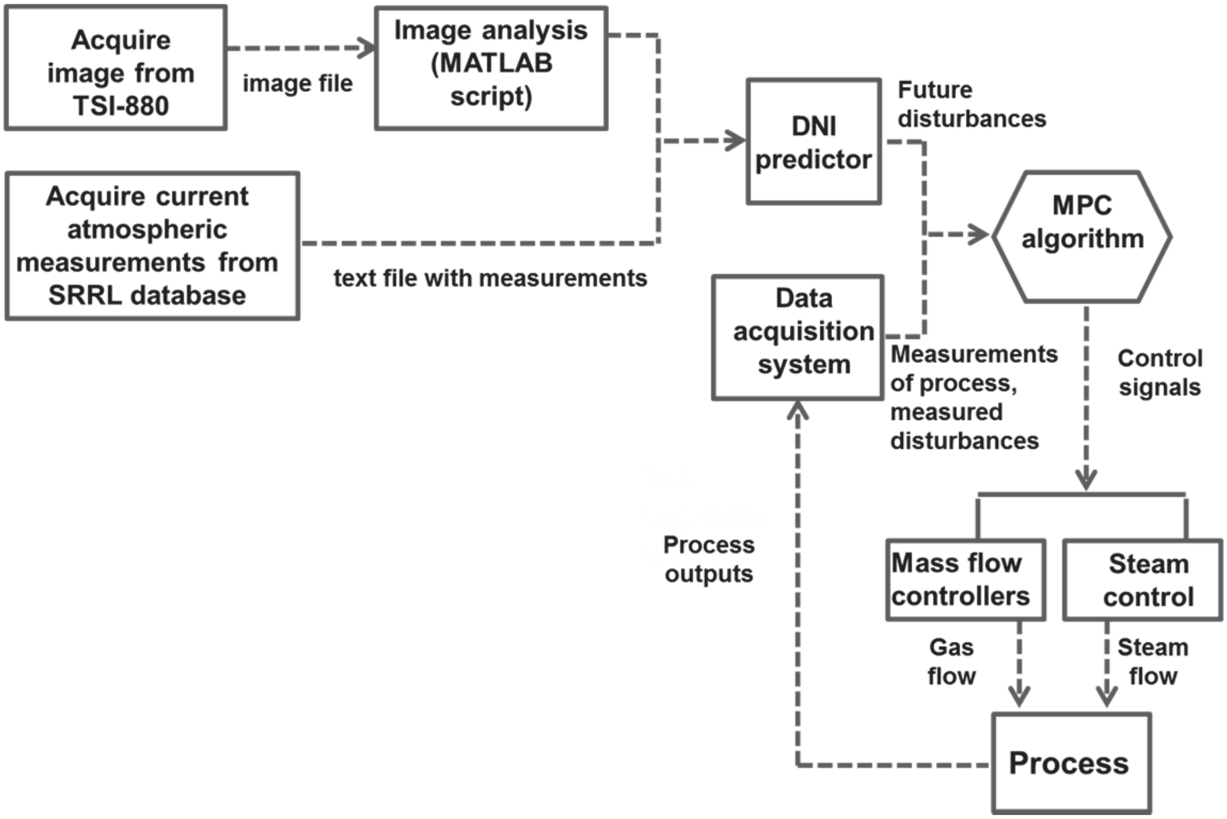

Acquire image from TSI-880
image file
Image analysis (MATLAB script)
Acquire current atmospheric measurements from SRRL database
text file with measurements
DNI predictor
Future disturbances
MPC algorithm
Data acquisition system
Measurements of process, measured disturbances
Control signals
Process outputs
Mass flow controllers
Steam control
Gas flow
Steam flow
Process

Figure 67. Use of image-based DNI predictions in a MPC algorithm with look-ahead for disturbances

4.5.4 Comparison of Different MPC Strategies

A DNI sequence corresponding to a real day (January 25th, 2012) was used to test the control strategies described in the previous section. The simulation started at 11:30 am MST (18:30 UTC) and stopped four hours later. The irradiation profile is representative of a typical day in which the reactor would operate: clear sky conditions at the beginning of the operation with the appearance of transient clouds later on. To remain within the operating region for which the controller was designed, DNI values were truncated at 700 W/m^2. All the controllers were implemented using the conservative parameters presented in Table 24.

The controlled responses for the three outputs are shown in Figure 68. From this figure it can be seen that the three controllers are able to minimize the effect of the solar irradiation changes in the product composition. Table 25 shows the average and maximum deviations for each of the controllers. For the synthesis gas fraction, the largest deviation from the setpoint was about 2-3% of the nominal value. For the CO:CO_2 ratio the largest deviations were more significant than for the synthesis gas case: they ranged from 11 to 15% of the nominal value. These deviations are larger than for the case of the synthesis gas fraction because the model used for the CO:CO_2 was less accurate than the model used to estimate the future values of the synthesis gas fraction. For all cases, the average deviation from the setpoint is less than 1%, which means that all the control strategies implemented had a satisfactory performance.

Table 25. Deviations from setpoint for different control strategies.

	Synthesis gas fraction		CO:CO_2 ratio	
Control strategy	**largest deviation from SP**	**average deviation from SP**	**largest deviation from SP**	**average deviation from SP**
MPC with image-based DNI predictions	2%	0.01%	11%	0.77%
MPC with DNI perfect predictions	2%	0%	15%	0.01%
MPC without DNI predictions	3%	0%	14%	0.01%

To make a better comparison of the different MPC strategies, the integral of the squared-error (ISE) was calculated for each run. Table 26 summarizes the ISE values for each controlled variable, obtained under the different MPC configurations. As expected, the lowest overall ISE values were obtained using the MPC controller that uses perfect predictions in the estimation of the future disturbances. For the synthesis gas case, there was no difference in performance between the controller that uses perfect DNI predictions and the controller that uses the image-based predictions. However, for this controlled variable, the use of predictions improved the performance of the controller by 50%. In the case of the CO:CO_2 ratio, the addition of DNI predictions to the control scheme only improved the performance by 3%, reducing the

ISE value from 3819 for the case with no DNI predictions to 3720 for the case with perfect DNI predictions. It is important to note that, for the CO:CO_2 ratio, the inaccuracy of the DNI predictor caused the controller that uses the image-based forecast to perform worse than the controller without any DNI predictions, resulting in a 100% increase in the ISE value. This reinforces the need for good DNI predictions, especially for a process with fast dynamics, where a good performance can be achieved with an MPC controller without disturbance predictions.

Table 26. ISE values for different MPC strategies.

MPC strategy	ISE for synthesis gas fraction	ISE for CO:CO2 ratio
MPC with image-based DNI predictions	0.0013	7868
MPC with perfect predictions	0.0014	3720
MPC with no look-ahead for disturbances	0.0026	3819

The use of predictions, regardless of their accuracy, improved the performance of the controllers for the synthesis gas fraction. Nonetheless, the implementation of the look-ahead with perfect predictions did not result in a large improvement for the CO:CO_2 ratio, in comparison with the case with no look ahead. The reason for this is that the model for synthesis gas fraction represented a better fit to the validation data than the model for the CO:CO_2 ratio; so even if the DNI predictions are perfect, the estimated values for CO:CO_2 ratio could still be inaccurate, reducing the improvement that can be obtained through the implementation of look-ahead for disturbances.

In Figure 68.c we can observe that the carbon conversion remained above the lower limit for all cases.

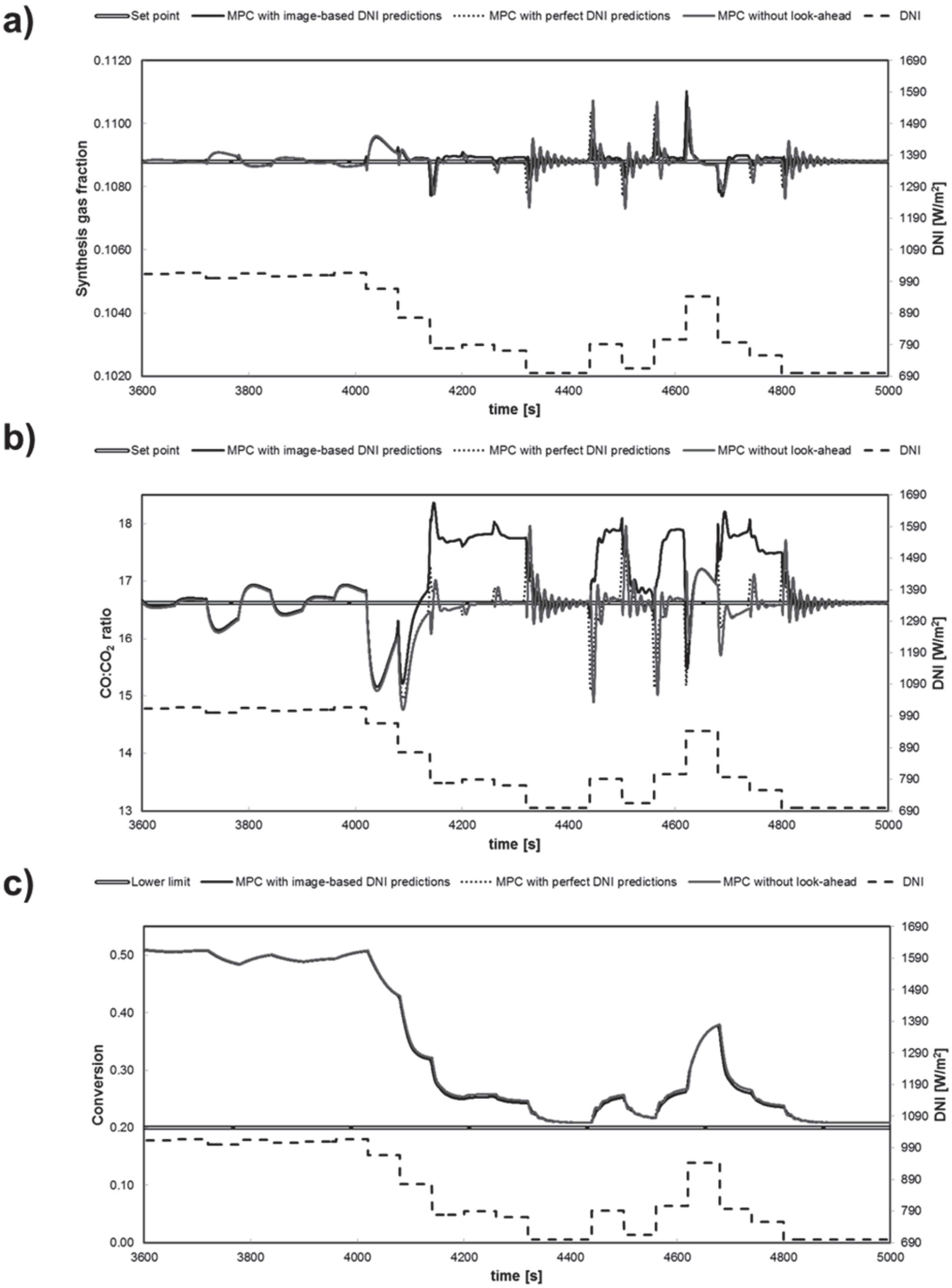


Figure 68. Comparison of MPC strategies. a) Synthesis gas fraction. b)CO:CO2 ratio. c) Conversion.

Figure 69 shows the manipulated variables for the different control strategies. For all strategies, the inputs do not present any sudden changes that can cause damages to the actuators. Also, the inputs remained within the desired limits. Therefore, the controllers show good disturbance rejection without being too aggressive on the manipulated variables.

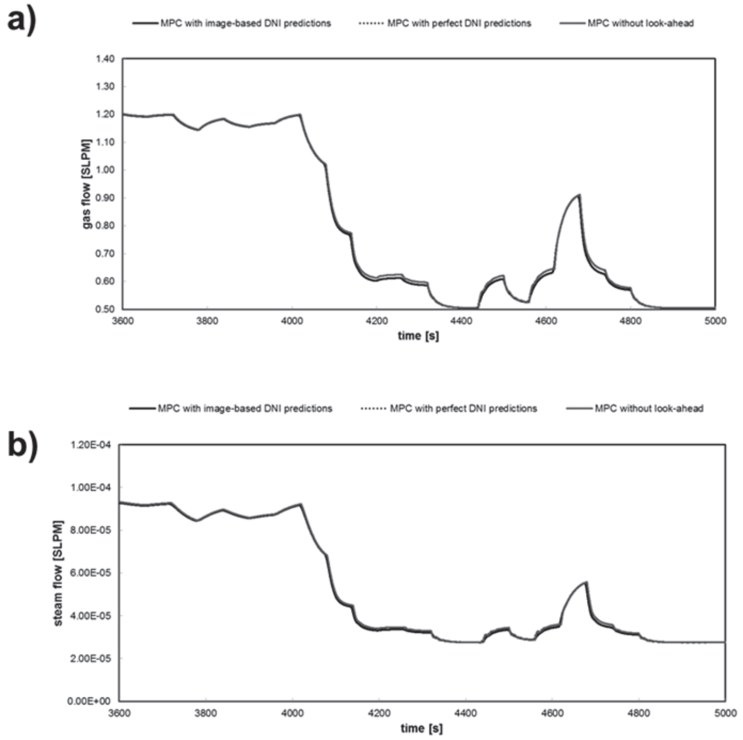


Figure 69. Manipulated variables using different MPC strategies. a) Gas flow rate. b) Steam flow.

4.6 Development of a Feedback Controller for Comparison

One of the objectives of the proposed control system is to outperform a well-tuned conventional feedback control system. The MPC controller is expected to perform better, because it automatically takes into account process interactions. Process interactions are one of the biggest concerns when designing a control strategy for MIMO systems, and they are considered to be present when each manipulated variable affects both controlled variables. This is the case for the solar-thermal reactor system, in which the gas flow and steam flow affect both the synthesis gas fraction and the CO:CO_2 ratio response. When these interactions are present, achieving the most effective control configuration is not straightforward. In general, each manipulated variable is paired with a controlled variable, but the difficulty in the design consists in defining the appropriate manipulated-controlled variable pairs. In addition, MPC inherently compensates for the effects of measured disturbances (such as the changes in DNI), providing predictive control action before the effect of the disturbances is presented in the controlled variables.

Section 4.6.1 describes the methodology applied to carry out the input-output pairings. Section 4.6.2 summarizes the controller tuning procedure and section 4.6.3 touches on the robustness of the feedback controller.

4.6.1 Feedback Control Strategy

The feedback strategy selected to control the solar-thermal reactor system consists of two parallel proportional-integral-derivative (PID) controllers working together. Each PID controller will use one of the manipulated variables to control one of the controlled variables.

In order to determine the most appropriate input-output pairs, Bristol's relative gain array method was employed [111]. In this method, an array containing the relative gains for all input-output pairs is calculated using the steady-state for the independent input-output pairs. The relative gain λ_{ij} between a controlled variable y_i and a manipulated variable u_j is defined as the dimensionless ratio of the open-loop gain over the closed-loop gain [104]:

$$\lambda_{ij} = \frac{\left(\frac{\partial y_i}{\partial u_j}\right)_u}{\left(\frac{\partial y_i}{\partial u_j}\right)_y} \tag{4-30}$$

For a system with two inputs and two outputs, an analytical expression for the relative gain can be obtained. The relative gain for the first input-output combination can be calculated by [104]:

$$\lambda = \frac{1}{1 - \frac{K_{12}K_{21}}{K_{11}K_{22}}} \tag{4-31}$$

where K_{ij} denotes the steady-state gain between y_i and u_j.

The relative gain array, Λ, can be computed as:

$$\Lambda = \begin{bmatrix} \lambda & 1-\lambda \\ 1-\lambda & \lambda \end{bmatrix} \tag{4-32}$$

Bristol [111] recommends that the controlled and manipulated variables be paired, so that the corresponding relative gains are positive and as close to one as possible.

The relative gain array for the solar-thermal reactor system was computed using the information from the transfer function matrix, presented in Eqn. 4-22. The resulting relative gain array was:

$$\Lambda = \begin{bmatrix} 1.16 & -0.16 \\ -0.16 & 1.16 \end{bmatrix} \tag{4-33}$$

where the columns correspond to the synthesis gas fraction and the CO:CO_2 ratio, and the rows represent the gas flow and the steam flow, respectively. When a relative gain for an input-output pair is negative, it means that the open-loop and the closed-loop gains between y_i and u_j have different signs, which can cause oscillations. Therefore, inputs and outputs that have a negative relative gain should not be paired. That leaves us with only one possible input-output combination: the gas flow will be used to control the synthesis gas fraction and the steam flow will be used to control the CO:CO_2 ratio.

4.6.2 PID Tuning

There are several methods that can be used to tune multi-loop PID control systems. In this work, we tuned the PID controllers using the sequential tuning method [104], in which the controller for a selected input-output pair is tuned first. After the first controller is tuned, the loop is closed and the second controller can be tuned. In this method, fastest loops are usually tuned first.

Looking at the open-loop transfer functions presented in Eqn. (4-22), it can be observed that the fastest loop corresponds to the synthesis gas fraction and gas flow pairing. Thus, this controller will be tuned first. When the first PID is tuned, the PID that uses steam flow to control for CO:CO_2 ratio can be tuned.

As an initial guess, the PID parameters were estimated using the formulas suggested by Madhuranthakam et al.[112]:

$$Kc = \frac{0.5249}{K_p}\left(\frac{\theta}{\theta+\tau}\right) \quad \textbf{(4-34)}$$

$$\tau_I = \tau\left(-1.9167\left(\frac{\theta}{\theta+\tau}\right) + 2.1356\right) \quad \textbf{(4-35)}$$

$$\tau_D = \tau\left(1.1321\left(\frac{\theta}{\theta+\tau}\right)^2 + 0.1788\left(\frac{\theta}{\theta+\tau}\right)\right) \quad \textbf{(4-36)}$$

where K_c, τ_I and τ_D represent the controller's gain, the integral time and the derivative time, respectively, and K_p, τ and θ represent the parameters of the first-order transfer function for the input-output pair being evaluated.

The use of the tuning relations presented above resulted in overactive controllers that became unstable in the presence of disturbances. The tuning parameters obtained through Eqns. 4-34 to 4-36 were used as initial guesses, but the controller parameters were iteratively changed and tested through simulations until the overall multi-loop control system provided a satisfactory response. One thing to note is that to make the controller less aggressive, the derivative term was removed (τ_D=0), which resulted in a Proportional-Integral (PI) controller, instead of a PID.

The controller parameters that produced the best response to a step change in DNI from 950 to 900 W/m^2 are presented in Table 27.

Table 27. Tuned controller parameters

Parameter	PI for synthesis gas fraction	PI for $CO:CO_2$ ratio
Kc	-6	$-0.6x10^{-6}$
τI	0.8	0.8

The performance of the controllers with the set of parameters proposed in Table 27 was evaluated, applying a step change in DNI from 950 to 900 W/m^2. Figure 70 compares the controlled and the open-loop responses for the synthesis gas fraction and the $CO:CO_2$ ratio. The feedback control strategy was able to maintain the setpoints without incurring in oscillations or large deviations. The response for the conversion is also shown in Figure 70.c, where we can see that the carbon conversion is sacrificed to maintain the setpoints in the synthesis gas fraction and the $CO:CO_2$ ratio.

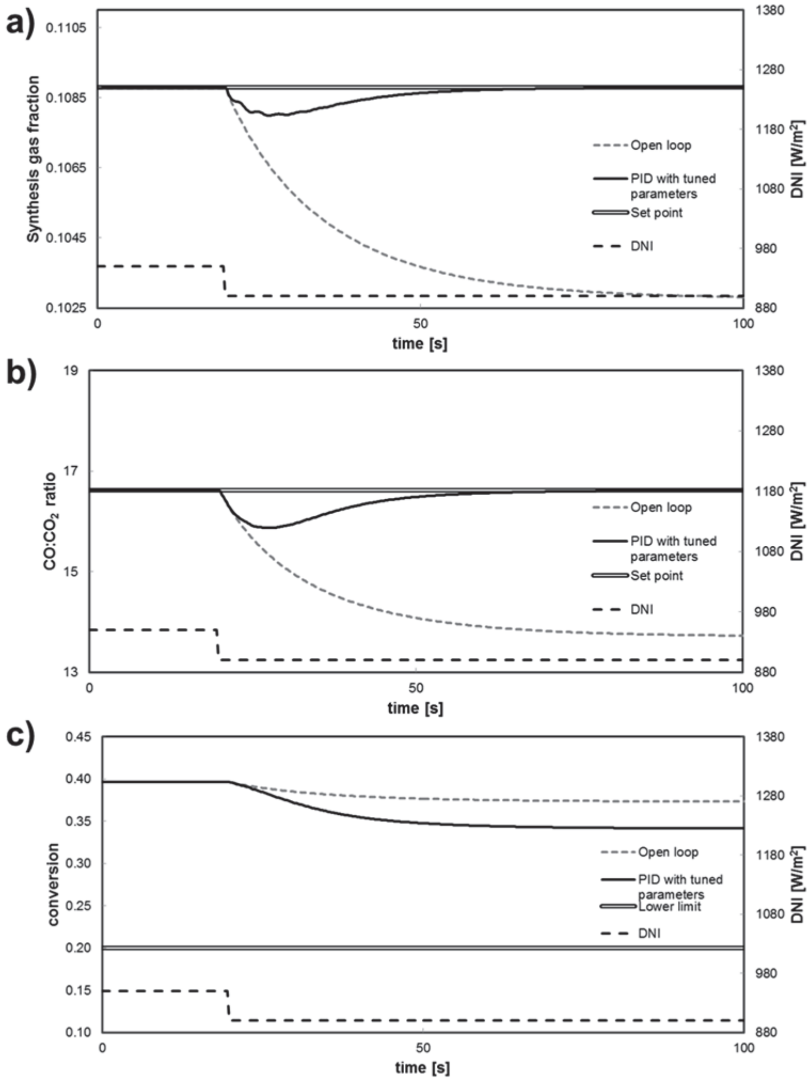


Figure 70. Closed-loop response to a step change in DNI, using PI controllers. a) Synthesis gas fraction. b)CO:CO2 ratio c) conversion

4.6.3 Robustness of Feedback Strategy

The proposed feedback control strategy responded well to step changes in the DNI around the nominal operating point. However, to have a robust control system, the control strategy must respond well to changes in the DNI throughout the entire operating region. The tuned PI controllers were tested using the same sequence of DNI changes used in section 4.5.2. The PI controllers became unstable when the DNI dropped below 900 W/m^2, which signaled that the controllers had to be re-tuned to ensure robust control. The controllers were made more conservative by increasing the value of the integral time (τ_I) and decreasing the value of the controller's gain (K_c). The PI parameters were changed iteratively, until no oscillations were observed for DNI values within the operating region. The final PI parameters are presented in Table 28.

Table 28. Parameters of robust PI controllers

Parameter	PI for synthesis gas fraction	PI for CO:CO$_2$ ratio
Kc	-4.5	-0.45×10^{-6}
Ti	4.2	3.6

Figure 71 shows the controlled response using the PI controllers with the parameters from Table 28. These controllers are slower and show larger deviations from the setpoint than the MPC controllers, but the deviations that they present are still smaller than those obtained without using any control strategy.

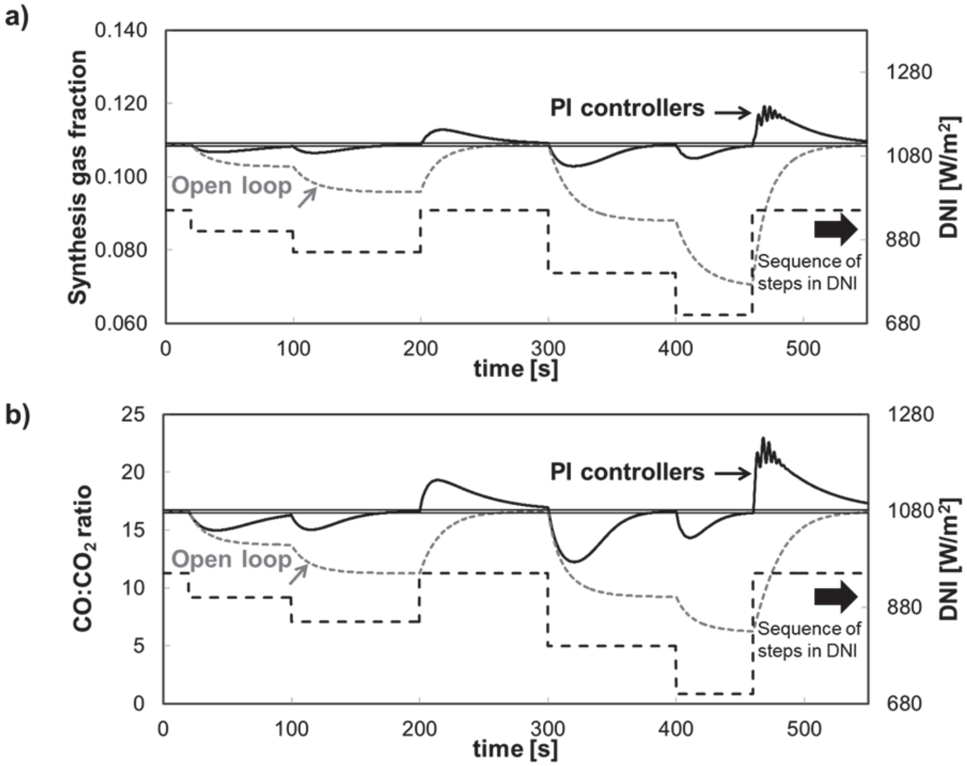


Figure 71. Closed-loop response to a sequence of steps in DNI using the robust PI controllers. a) Synthesis gas fraction. b) CO:CO$_2$ ratio

4.7 Comparison of Control Strategies

The proposed MPC controller was compared to the feedback control strategy described in section 4.6 through a simulation experiment, using the DNI sequence described in section 4.5.4. The controllers' response was also compared to the response of the uncontrolled process and to the response of an MPC controller with perfect DNI predictions. Figure 72 shows a comparison of the output responses. In general, it can be observed that, regardless of the control strategy implemented, the controlled response shows remarkably fewer deviations than the uncontrolled response. This means that the quality of the synthesis gas produced can be significantly improved by implementing a control system. Figure 72.a. shows a comparison of the responses for the synthesis gas fraction, under different control strategies. As it can be seen in this figure, both MPC strategies outperform the PI controllers, showing smaller deviations from the setpoints. As in section 4.5.4, the inaccuracies in the solar irradiation predictions do not affect the performance of the controlled response for the synthesis gas.

For the case of the CO:CO$_2$ ratio, shown in Figure 72.b., it can be observed that both the proposed MPC and the feedback control strategy are unable to maintain the setpoint as closely as the MPC controller with perfect DNI predictions. Since the response for the PI controllers is slower, the deviations from the setpoint last longer than the deviations from the setpoint for the proposed MPC controller. Therefore, even if the inaccuracies in the DNI prediction degrade the performance of the MPC, the proposed control system still outperforms a feedback strategy.
Figure 72.c. shows that the conversion did not fall below its limit for any of the strategies tested.
Table 29 compares the ISE for synthesis gas fraction and CO:CO$_2$ ratio obtained with the different controllers tested. The results suggest that implementing a feedback control strategy reduces the ISE by two or three orders of magnitude. The performance is improved even further when a MPC strategy is implemented: the ISE for the proposed MPC controller is 65 times less than that of the feedback controllers in the case of the synthesis gas fraction, and five times less in the CO:CO$_2$ ratio case. For the CO:CO$_2$ ratio, the ISE can be reduced even further with a

controller that uses perfect predictions, which results in an ISE value ten times smaller than the ISE for the feedback controllers.

The results show that, even when the DNI predictions are not perfect, the proposed controller rejects the disturbances in solar irradiation better than a feedback controller, fulfilling the control objectives established in section 4.3.2.

Table 29. Comparison of ISE for different control strategies

Control strategy	ISE for synthesis gas fraction	ISE for CO:CO_2 ratio
MPC with image-based DNI predictions	0.0013	7868
MPC with perfect predictions	0.0014	3720
PI controllers	0.0835	37376
No controller	59.08	4606700

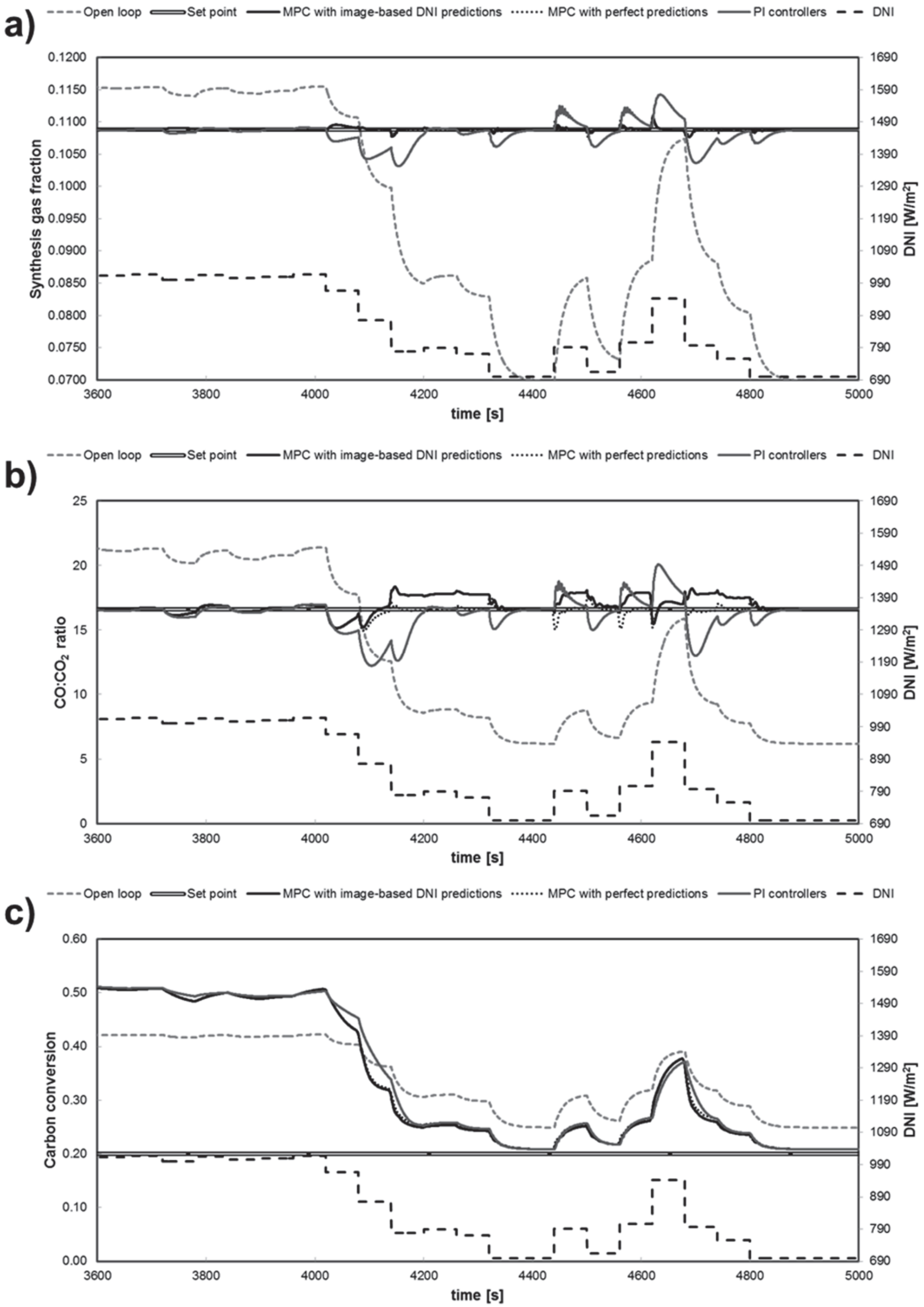


Figure 72. Comparison of different control strategies. a) Synthesis gas fraction. b) $CO:CO_2$ ratio. c) Carbon conversion

4.8 Conclusions

A model predictive controller was developed for a solar-thermal reactor, aiming at minimizing the effect of variations in the solar irradiation in the product composition. The goal of the controller was to maintain a stable synthesis gas fraction and the $CO:CO_2$ ratio in the product stream, through the manipulation of the inlet gas and steam flows. The controller uses DNI predictions to estimate the future outputs and, therefore, it can start acting before changes in the DNI happen. The proposed controller uses the image-based DNI predictions, developed in Chapter Three.

The proposed controller was compared to a control system that uses perfect DNI predictions for the disturbances and to a controller that does not use any predictions for the DNI. It was found that the use of DNI predictions in the control algorithm did not present a significant improvement over a controller that does not use disturbance predictions. Moreover, the use of inaccurate DNI predictions was shown to affect the performance of the controller negatively, in the case of the $CO:CO_2$ ratio.

The proposed controller was compared to a multi-loop feedback control strategy. Through simulations of both control systems, it was shown that the MPC controller outperforms the feedback control strategy, even when it uses inaccurate DNI predictions.

Even though solar-thermal reactors are highly nonlinear systems, robust control could be achieved by tuning the MPC with more conservative parameters, more specifically, by using larger weights for the move suppression term. This is an important result because the implementation of a more complex strategy, such as linearized scheduled MPC or nonlinear MPC, could be avoided, resulting in faster computations of the control moves.

The results obtained suggest that our controller is suitable for a real-world implementation.

CHAPTER FIVE

CONCLUSIONS

5.1 Summary

In this work, a model predictive control (MPC) system for a solar-thermal reactor was developed. The controller aimed at minimizing the effect of variations in solar irradiation through the manipulation of gas and steam flow rates into the reactor.

First, a simplified dynamic model of the process was derived from unsteady mass and energy balances. The model was validated with experimental data obtained at the High Flux Solar Furnace (HFSF at National Renewable Energy Laboratory (NREL). Simulations showed that the error of the model was less than 5% when compared to the experimental reactor temperatures at steady-state. In addition to the development of the model, a simplified approach for the implementation of volumetric exchange theory was derived and used to model the radiative heat transfer between the tube walls and the carbon particles. Originally, the use of a surface exchange model was proposed, as a computationally inexpensive way to model the radiation from the tube walls to the particles. Nonetheless, the simplicity of our approach made it possible to adopt volumetric exchange theory to model radiation without increasing the computational time for the simulations. In our case, there was no significant difference between the particle temperatures obtained through both models, but we recognize that the use of surface exchange theory might not be appropriate for particles with different optical properties and size distributions.

The model of the process was used to perform a steady-state analysis of the system, which led to the selection of the nominal operating point and the specification of operating regions. Next, the dynamics of the process were studied, and a control system was consequently proposed. The control system objectives were:

(i) To maintain the synthesis gas fraction and CO:CO$_2$ ratio in the product stream as close as possible to their nominal operating values;

(ii) To minimize the effect of disturbances in solar irradiation better than a well-tuned feedback control strategy;

(iii) To ensure that the conversion does not fall below 20%;

(iv) To prevent excessive movement in the manipulated variables and to keep them within a physically realizable operating range.

Then, the dynamic model of the process was used as a basis for obtaining data for the identification of a linear model. The use of a linear model in the MPC algorithm converts the optimization problem into rapidly-solved quadratic programming problem.

Next, a linear MPC controller was implemented using MATLAB®'s MPC Toolbox. The first-principles dynamic model was used to simulate the plant's response. The controller was tuned through simulations until its performance throughout the operating range was considered satisfactory.

Through the dynamic simulations carried out, the plant was shown to be highly nonlinear, and, initially, the implementation of a linearized scheduled MPC approach was proposed. Results showed that a conservative MPC controller was robust enough to reject the disturbances in Direct Normal Irradiance (DNI) across the entire operating region. This suggests that satisfactory performance can be reached by tuning the linear MPC for robustness, rather than by implementing a more complex strategy such as linearized scheduled MPC or nonlinear MPC.

In the introductory chapter, we proposed a control system that uses DNI predictions in the estimation of the future outputs of the process. In order to forecast the DNI, a one-minute ahead solar irradiation predictor was developed. The predictor uses images of the sky and atmospheric measurements to forecast DNI. In particular, images are processed by an algorithm which analyzes them, classifies them in a scenario (sunny, cloudy, covered and uncovered) and provides an attribute representation of them. A set of images that corresponded to representative days throughout the year were analyzed, and the image-based attributes obtained through this analysis were combined with the corresponding atmospheric measurements for that date and time. The attribute sets were used to train a machine learning algorithm for each scenario. Different learning algorithms were tested using the software WEKA, and the best performing algorithms for each scenario were selected and implemented in the predictor. It is important to note that the most predictive attributes were not the image-based attributes, but some of the atmospheric attributes and, above all, the current DNI. We attribute this to the presence of a shadow-band which, by covering the sun to protect the camera, makes it impossible to see when clouds are about to cover (or uncover) the sun. The presence of this band negatively affects the quality of the attributes of the image, which in turn negatively affects the quality of the prediction.

The DNI forecasts obtained through the proposed strategy were compared to a persistence model, in which the one-minute-ahead DNI is predicted to be equal to the current DNI. The proposed DNI predictor did not show an improvement over the persistence model, because they both failed to recognize sharp changes in the DNI.

We compared the performance obtained by alternatively feeding the controller with the predictions given by our predictor and by feeding it with perfect predictions. For the case of the synthesis gas fraction, both controllers were able to reject the disturbances equally well. However, inaccurate DNI predictions caused the $CO:CO_2$ ratio to have larger deviations from the setpoint than those obtained with perfect predictions.

Finally the proposed control system was shown to outperform a feedback control strategy, even if the DNI predictions were not accurate. In conclusion, the proposed controller was able to achieve all goals and was considered suitable for a real-world implementation.

5.2 Conclusions

The first hypothesis of this thesis was that a model-based predictive controller applied to a solar-thermal reactor process performs better than a conventional feedback control strategy. This hypothesis was proven through dynamic simulations of both control systems, which showed that, when controlled with the proposed MPC controller, both controlled outputs had smaller deviations from their setpoints than when controlled with a set of fine-tuned PI controllers. After a four-hour simulation under real conditions, the MPC controller had ISE (integral of the squared-error) values of 0.0013 for the synthesis gas fraction and 7868 for the $CO:CO_2$ ratio. These values were one order of magnitude smaller than the values obtained with the PI controllers for the same simulation, which were 0.0835 and 37376, respectively.

Second, we had hypothesized that anticipating the disturbances was going to improve the performance of the controller even further. After a four-hour simulation with real conditions, the MPC controller that used perfect predictions for the disturbances had ISE values of 0.0014 and 3720. In comparison, the MPC controller without any disturbance estimation (no look-ahead for disturbances) had ISE values of 0.0026 and 3819. These numbers represent a 50% decrease in the ISE for the synthesis gas fraction when the disturbance predictions were added, but only a 3% decrease in the ISE for the $CO:CO_2$ ratio. Installing a system that anticipates the disturbances might be expensive, so such installation must result in a significant improvement in the performance of the controller to be justified. However, we expect that for a system with an absorbing cavity, the addition of disturbance estimations will result in a more significant improvement in performance than for the system with a reflective cavity. Systems with absorbing cavities have inherently slower dynamic responses; therefore, if the changes in DNI are anticipated, the control can start acting before these changes happen, resulting in faster disturbance rejection.

Third, since the plant being studied is highly nonlinear, we had anticipated the use of a linearized scheduled MPC strategy. However, by re-tuning the MPC to make it more conservative, robust control was achieved using only one linear MPC.

Finally, we recognize that the solar irradiation predictor developed in Chapter Two needs to be able to anticipate sharp changes in the DNI, since these changes are the most interesting ones from a control point of view. As explained above, the presence of the shadow-band in the pictures negatively impacts the performance of the DNI predictor, at the short horizons required by our system. Nevertheless, we expect that the quality of image-based attributes at longer horizons will be higher than the quality of the attributes at short horizons, resulting in more accurate DNI predictions.

5.3 Recommendations

The goal of this study was to develop a control system that could be applied to the existing prototype reactor at NREL. To this end, the controller was designed based on the dynamic characteristics of the reactor. Limitations of the equipment available at NREL hindered the field implementation at the time of the study, but future implementation is still a possibility. Nevertheless, some modifications to the proposed system would be required. The most important of these modifications stems from the differences in time delays between the simulated system and the real one. For the simulations, the gas flow and the steam flow were both assumed to have a delay of 0.25 s; in the real system, the gas flow had a delay of 8 s and the steam flow had a delay of 300 s. The presence of these large time delays, especially in the case of the steam flow, will cause the MPC to require longer prediction horizons than for the case without large time delays (P must be larger than the largest time delay). At longer time horizons, inaccuracies in the model will have a higher impact on the performance of the controller, suggesting the need for more accurate models. To improve the accuracy of the linear models, the range for linearization has to be reduced, resulting in a narrower region in which the model would be applicable. As a consequence, the approach proposed in this study is likely to become invalid and a single linear MPC controller will not be able to operate in the whole region without causing instability. To summarize, the implementation at NREL requires the development of a controller of higher complexity, and, since the modified controller will not have any practical relevance for industry, we do not recommend it.

On the other hand, we strongly recommend the adaptation of the control system to a scaled-up process with an absorbing cavity. Absorbing cavities represent a better option for scale-up and commercialization of solar thermochemical technologies, because, in comparison with reflective cavities, they can achieve higher efficiencies and more uniform temperature profiles. In addition, absorbing cavities have shown slower dynamic responses, thus a predictive control system that anticipates changes in solar radiation using images of the sky, like the one proposed in this study, would be more beneficial for systems with absorbing cavities than for systems with reflective cavities. The adaptation of the MPC to an absorbing cavity system would also entail the development of a DNI predictor at horizons longer than one minute ahead, which is expected to be less affected by the presence of the shadow band and, therefore, expected to provide more accurate results.

REFERENCES

[1] World Resources Institute, “Global Direct Normal Solar Irradiation Map.” May-2009.

[2] A. Steinfeld and R. Palumbo, “Solar Thermochemical Process Technology,” *Encyclopedia of Physical Science and Technology*, vol. 15. Academic Press, pp. 237–256, 2001.

[3] R. Palumbo, M. Keunecke, S. Möller, and A. Steinfeld, “Reflections on the design of solar thermal chemical reactors: thoughts in transformation,” *Energy*, vol. 29, no. 5–6, pp. 727–744, Apr. 2004.

[4] E. A. Fletcher, “Solarthermal Processing: A Review,” *Journal of Solar Energy Engineering*, vol. 123, no. 2, p. 63, 2001.

[5] C. Perkins, “Likely near-term solar-thermal water splitting technologies,” *International Journal of Hydrogen Energy*, vol. 29, no. 15, pp. 1587–1599, Dec. 2004.

[6] M. Roeb, C. Sattler, R. Klüser, N. Monnerie, L. de Oliveira, A. G. Konstandopoulos, C. Agrafiotis, V. T. Zaspalis, L. Nalbandian, A. Steele, and P. Stobbe, “Solar Hydrogen Production by a Two-Step Cycle Based on Mixed Iron Oxides,” *Journal of Solar Energy Engineering*, vol. 128, no. 2, p. 125, 2006.

[7] J. Dahl, “Solar-thermal dissociation of methane in a fluid-wall aerosol flow reactor,” *International Journal of Hydrogen Energy*, vol. 29, no. 7, pp. 725–736, Jul. 2004.

[8] A. Zgraggen, P. Haueter, D. Trommer, M. Romero, J. Dejesus, and A. Steinfeld, “Hydrogen production by steam-gasification of petroleum coke using concentrated solar power—II Reactor design, testing, and modeling,” *International Journal of Hydrogen Energy*, vol. 31, no. 6, pp. 797–811, May 2006.

[9] C. Perkins, B. Woodruff, L. Andrews, P. Lichty, B. Lancaster, C. Bingham, and A. Weimer, “Synthesis gas production by rapid solar thermal gasification of corn stover,” presented at the 14th Biennial CSP SolarPACES Symposium, Las Vegas, Nevada, 2008.

[10] P. McKendry, “Energy production from biomass (part 3): gasification technologies,” *Bioresource technology*, vol. 83, no. 1, pp. 55–63, 2002.

[11] C. Higman and M. van der Burgt, *Gasification*. USA: Elsevier, Inc, 2003.

[12] P. Lichty, C. Perkins, B. Woodruff, C. Bingham, and A. Weimer, “Rapid High Temperature Solar Thermal Biomass Gasification in a Prototype Cavity Reactor,” *J. Sol. Energy Eng.*, vol. 132, no. 1, pp. 011012–011012, Jan. 2010.

[13] D. W. Gregg, R. W. Taylor, J. H. Campbell, J. R. Taylor, and A. Cotton, “Solar gasification of coal, activated carbon, coke and coal and biomass mixtures,” *Solar Energy*, vol. 25, no. 4, pp. 353–364, 1980.

[14] J. Lédé, “SOLAR THERMOCHEMICAL CONVERSION OF BIOMASS,” *Solar Energy*, vol. 65, no. 1, pp. 3–13, Jan. 1999.

[15] C. Perkins and A. W. Weimer, “Solar-thermal production of renewable hydrogen,” *AIChE Journal*, vol. 55, no. 2, pp. 286–293, Feb. 2009.

[16] E. F. Camacho, F. R. Rubio, M. Berenguel, and L. Valenzuela, “A survey on control schemes for distributed solar collector fields. Part I: Modeling and basic control approaches,” *Solar Energy*, vol. 81, no. 10, pp. 1240–1251, Oct. 2007.

[17] J. Petrasch, P. Osch, and A. Steinfeld, “Dynamics and control of solar thermochemical reactors,” *Chemical Engineering Journal*, vol. 145, no. 3, pp. 362–370, Jan. 2009.

[18] E. F. Camacho, *Advanced control of solar plants*. Springer Verlag, 1997.

[19] E. F. Camacho, F. R. Rubio, M. Berenguel, and L. Valenzuela, “A survey on control schemes for distributed solar collector fields. Part II: Advanced control approaches,” *Solar Energy*, vol. 81, no. 10, pp. 1252–1272, Oct. 2007.

[20] S. Skogestad and I. Postlethwaite, *Multivariable Feedback Control*.

[21] E. F. Camacho and M. Berenguel, “Application of Generalized Predictive Control to a Solar Power Plant,” presented at the 3rd IEEE Conference on Control Applications, 1994, vol. 3, pp. 1252–1272.

[22] E. F. Camacho and M. Berenguel, “Robust adaptive model predictive control of a solar plant with bounded uncertainties,” *International Journal of Adaptive Control and Signal Processing*, vol. 11, pp. 311–325, 1997.

[23] F. Coito, J. M. Lemos, R. M. Silva, and E. Mosca, “Adaptive control of a solar energy plant exploiting accesible disturbances,” *International Journal of Adaptive Control and Signal Processing*, vol. 11, pp. 327–342, 1997.

[24] A. Meaburn and F. M. Hughes, “A simple predictive controller for use on large scale arrays of parabolic trough collectors,” *Solar Energy*, vol. 56, no. 6, pp. 583–595, 1996.

[25] E. F. Camacho and C. Bordons, *Model Predictive Control*. London: Springer Verlag, 1999.

[26] E. F. Camacho, F. R. Rubio, and F. M. Hughes, “Self-tuning control of a solar power plant with a distributed collector field,” *IEEE Control Systems Magazine*, pp. 72–78, 1992.

[27] M. Romero, R. Buck, and J. E. Pacheco, “An Update on Solar Central Receiver Systems, Projects, and Technologies,” *Journal of Solar Energy Engineering*, vol. 124, no. 2, p. 98, 2002.

[28] D. D. Sworder and R. Rogers, “An LQ-Solution to a control problem associated with a solar thermal central receiver,” *IEEE Transactions on Automatic Control*, vol. AC-28, no. 10, pp. 971–978, 1983.

[29] M. López-Martínez, M. Vargas, and F. Rubio, “Vision-Based System for the Safe Operation of a Solar Power Tower Plan,” *Advances in Artificial Intelligence—IBERAMIA 2002*, pp. 943–952, 2002.

[30] National Renewable Energy Laboratory, “Solar Radiation Research Laboratory.” [Online]. Available: http://www.nrel.gov/midc/srrl_bms/.

[31] M. Simone, E. Biagini, C. Galletti, and L. Tognotti, “Evaluation of global biomass devolatilization kinetics in a drop tube reactor with CFD aided experiments,” *Fuel*, vol. 88, no. 10, pp. 1818–1827, Oct. 2009.

[32] C. Perkins and A. Weimer, “Computational Fluid Dynamics Simulation of a Tubular Aerosol Reactor for Solar Thermal ZnO Decomposition,” *Journal of Solar Energy Engineering*, vol. 129, no. 4, p. 391, 2007.

[33] J. Wyss, J. Martinek, M. Kerins, J. K. Dahl, A. Weimer, A. Lewandowski, and C. Bingham, “Rapid solar-thermal decarbonization of methane in a fluid-wall aerosol flow reactor - fundamentals and application,” *International Journal of Chemical Reactor Engineering*, vol. 5, 2007.

[34] St. Gobain Advance Ceramics, “Normal Spectral Emissivity of Heloxoy(R) SA Silicon Carbide.” Technical Data Sheet, 2003.

[35] Omega, “Emmisivity of Common Materials.” [Online]. Available: http://www.omega.com/literature/transactions/volume1/emissivityb.html.

[36] “Chevron Phillips Schawinigan Black (R) Acetylene Black, Material property data.” [Online]. Available: http://www.matweb.com/search/datasheetext.aspx?matid=3101.

[37] I. Barin, *Thermochemical Data of Pure Substances, Part 1*. New York: VCH, 1989.

[38] St. Gobain Advance Ceramics, “Hexoloy(R) SA Silicon Carbide, Technical Data Sheet.” 2003.

[39] Engineers Edge, “Thermal Conductivity of Gases.” [Online]. Available: http://www.engineersedge.com/heat_transfer/thermal-conductivity-gases-htm.

[40] P.E. Khizhnyak, A.V. Chechetkin, and A.P. Glybin, “Thermal conductivity of carbon black,” *translated from Inzhenerno-Fizicheskii Zhurnal*, vol. 37, no. 3, pp. 475–478, 1979.

[41] W. Hinds, *Aerosol Technology: Properties, behavior, and measurement of airborne particles*, 2nd ed. USA: John Wiley and Sons, Inc., 1999.

[42] A. W. Weimer and D. E. Clough, “Modeling of char particle size/conversion distributions in a fluidized bed gasifier:non-isothermal effects,” *Powder Technology*, vol. 27, pp. 85–103, 1980.

[43] C. Dupont, G. Boissonnet, J. M. Seiler, P. Gauthier, and D. Schweich, “Study about the kinetic processes of biomass steam gasification,” *Fuel*, vol. 86, pp. 32–40, 2007.

[44] S. Tescari, P. Neveu, and N. Mazet, “Thermochemical Solar Reactor: Simplified Method for the Geometrical Optimization at a Given Incident Flux,” *International Journal of Chemical Reactor Engineering*, vol. 8, no. 1, 2010.

[45] W. E. Ranz and W. R. Marshall, “Evaporation from drops, Part I,” *Chemical Engineering Progress*, vol. 48, no. 3, pp. 141–146, 1952.

[46] R. B. Bird, W. E. Stewart, and E. N. Lightfoot, *Transport Phenomena*, 2nd ed. USA: John Wiley and Sons, Inc., 2002.

[47] P. Lichty, C. Perkins, B. Woodruff, and A. W. Weimer, “Rapid high temperature solar thermal biomas gasification in a prototype cavity reactor,” *Journal of Solar Energy Engineering*, vol. 132, 2010.

[48] M. Kerker, *The scattering of light and other electromagnetic radiation*. USA: Academic Press, 1969.

[49] R. Siegel and J. Howell, *Thermal Radiation Heat Transfer*, 4th ed. USA: Taylor & Francis, 2002.

[50] W. H. Dalzell and A. F. Sarofim, “Optical constants of soot and their application to heat-flux calculations,” *Journal of Heat Transfer*, vol. 91, no. 1, pp. 100–104, 1969.

[51] *MATLAB®*. The MathWorks, Inc.

[52] C. K. Lee, M. Morbidelli, and A. Varma, “Steady state multiplicity behavior of an isothermal axial dispersion fixed-bed reactor with nonuniformly active catalyst,” *Chemical engineering science*, vol. 42, no. 7, pp. 1595–1608, 1987.

[53] A. A. Kumar and K. A. Stol, “Scheduled model predictive control of a wind turbine,” 2009.

[54] A. Lewandowski, C. Bingham, J. O’Gallagher, R. Winston, and D. Sagie, “Performance characterization of the SERI high-flux solar furnace,” *Solar Energy Materials*, vol. 24, no. 1, pp. 550–563, 1991.

[55] R. B. Woodruff, P. Kreider, and A. W. Weimer, “A novel brush feeder for the pneumatic delivery of dispersed small particles at steady feed rates,” *Powder Technology*, vol. 229, pp. 45–50, Oct. 2012.

[56] D. Trommer, F. Noembrini, M. Fasciana, D. Rodriguez, A. Morales, M. Romero, and A. Steinfeld, “Hydrogen production by steam-gasification of petroleum coke using concentrated solar power I. Thermodynamic and kinetic analyses,” *International Journal of Hydrogen Energy*, vol. 30, no. 6, pp. 605–618, May 2005.

[57] A. Z’Graggen and A. Steinfeld, “A two-phase reactor model for the steam-gasification of carbonaceous materials under concentrated thermal radiation,” *Chemical Engineering and Processing: Process Intensification*, vol. 47, no. 4, pp. 655–662, Apr. 2008.

[58] J. Fermoso, B. Arias, C. Pevida, M. G. Plaza, F. Rubiera, and J. J. Pis, “Kinetic models comparison for steam gasification of different nature fuel chars,” *Journal of Thermal Analysis and Calorimetry*, vol. 91, no. 3, pp. 779–786, 2008.

[59] D. Trommer and A. Steinfeld, “Kinetic modeling for the combined pyrolysis and steam gasification of petroleum coke and experimental determination of the rate constants by dynamic thermogravimetry in the 500-1520 K range,” *Energy and Fuels*, vol. 20, pp. 1250–1258, 2006.

[60] M. D. Mann, R. Z. Knutson, J. Erjavec, and J. P. Jacobsen, “Modeling reaction kinetics of steam gasification for a transport gasifier,” *Fuel*, vol. 83, no. 11–12, pp. 1643–1650, Aug. 2004.

[61] R. C. Everson, H. W. J. P. Neomagus, H. Kasaini, and D. Njapha, “Reaction kinetics of pulverized coal-chars derived from inertinite-rich coal discards: Gasification with carbon dioxide and steam,” *Fuel*, vol. 85, no. 7–8, pp. 1076–1082, May 2006.

[62] R. Müller, P. v. Zedtwitz, A. Wokaun, and A. Steinfeld, “Kinetic investigation on steam gasification of charcoal under direct high-flux irradiation,” *Chemical Engineering Science*, vol. 58, no. 22, pp. 5111–5119, Nov. 2003.

[63] L. Sherwood, “U.S. Solar Market Trends 2010,” Interstate Renewable Energy Council, Jun. 2011.

[64] Solar Energy Industries Association and GTM Research, “U.S. Solar Market Insight Report Q2 2012,” Executive Summary, 2012.

[65] GE Energy, “Western Wind and Solar Integration Study: Executive Summary,” National Renewable Energy Laboratory, 2010.

[66] A. Ipakchi and F. Albuyeh, “Grid of the future,” *IEEE Power and Energy Magazine*, vol. 7, no. 2, pp. 52–62, Mar. 2009.

[67] K. M. Powell, J. D. Hedengren, and T. F. Edgar, “Dynamic Optimization of a Solar Thermal Energy Storage System over a 24 Hour Period using Weather Forecasts.”

[68] R. Marquez and C. F. M. Coimbra, “Forecasting of global and direct solar irradiance using stochastic learning methods, ground experiments and the NWS database,” *Solar Energy*, vol. 85, no. 5, pp. 746–756, May 2011.

[69] A. Mellit and A. M. Pavan, “A 24-h forecast of solar irradiance using artificial neural network: Application for performance prediction of a grid-connected PV plant at Trieste, Italy,” *Solar Energy*, vol. 84, no. 5, pp. 807–821, May 2010.

[70] A. Sfetsos and A. H. Coonick, “Univariate and multivariate forecasting of hourly solar radiation with artificial intelligence techniques,” *Solar Energy*, vol. 68, no. 2, pp. 169–178, 2000.

[71] L. Martín, L. F. Zarzalejo, J. Polo, A. Navarro, R. Marchante, and M. Cony, “Prediction of global solar irradiance based on time series analysis: Application to solar thermal power plants energy production planning,” *Solar Energy*, vol. 84, no. 10, pp. 1772–1781, Oct. 2010.

[72] R. Perez, P. Ineichen, K. Moore, M. Kmiecik, C. Chain, R. George, and F. Vignola, “A new operational model for satellite-derived irradiances: description and validation,” *Solar Energy*, vol. 73, no. 5, pp. 307–317, 2002.

[73] A. Hammer, D. Heinemann, E. Lorenz, and B. Lückehe, “Short-term forecasting of solar radiation: a statistical approach using satellite data,” *Solar Energy*, vol. 67, no. 1, pp. 139–150, 1999.

[74] C. W. Chow, B. Urquhart, M. Lave, A. Dominguez, J. Kleissl, J. Shields, and B. Washom, “Intra-hour forecasting with a total sky imager at the UC San Diego solar energy testbed,” *Solar Energy*, vol. 85, no. 11, pp. 2881–2893, Nov. 2011.

[75] R. Marquez, V. G. Gueorguiev, and C. F. M. Coimbra, “Forecasting of Global Horizontal Irradiance Using Sky Cover Indices,” presented at the ASME 2011 5th International Conference on Energy Sustainability, Washington, DC, 2011.

[76] R. Marquez and C. F. M. Coimbra, “Intra-hour DNI forecasting based on cloud tracking image analysis,” *Solar Energy*, Oct. 2012.

[77] Yankee Environmental Systems, Inc., “TSI-880 Autmatic Total Sky Imager.” 2003.

[78] Eppley Lab, “Normal Incidence Pyrheliometer.” [Online]. Available: http://www.eppleylab.com/PrdNormIncPyrhelmtr.htm.

[79] Campbell Scientific, Inc, “Campbell Scientific,” 2012. [Online]. Available: http://www.campbellsci.com/hmp45c-l.

[80] NRG Systems, Inc, “NRG Systems, NRG #40C,” 2008. [Online]. Available: http://www.nrgsystems.com/sitecore/content/Products/1900.aspx.

[81] NRG Systems, Inc, “NRG Systems, NRG #200P,” 2008. [Online]. Available: http://www.nrgsystems.com/sitecore/content/Products/1904.aspx.

[82] Campbell Scientific, Inc, “CS105/CS105MD Barometric Pressure Sensor, Instruction Manual.” 2007.

[83] I. Reda and A. Andreas, “Solar position algorithm for solar radiation applications,” *Solar Energy*, vol. 76, no. 5, pp. 577–589, Jan. 2004.

[84] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA data mining software: an update,” *ACM SIGKDD Explorations Newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[85] The Mathworks, Inc, “Image Processing Toolbox for MATLAB.” 2012.

[86] NOAA Earth System Research Laboratory, “NOAA Solar Calculator.” [Online]. Available: http://www.esrl.noaa.gov/gmd/grad/solcalc/.

[87] Jean Meeus, *Astronomical Algorithms*, 1st ed. Willmann-Bell, 1991.

[88] The Mathworks, Inc, “Identifying Round Objects, MATLAB’s Image Processing Toolbox Examples.” 2012.

[89] Ian H. Witten and Eibe Frank, *Data Mining: Practical Machine Learning Tools and Techniques*, 2nd ed. USA: Elsevier, Inc, 2005.

[90] Peter J. Rousseeuw, “Least Median of Squares Regression,” *Journal of the American Statistical Association*, vol. 79, no. 388, Dec. 1984.

[91] Yong Wang and Ian H. Witten, “Pace Regression,” *working paper*, 1999.

[92] Yong Wang, “A new approach to fitting linear models in high dimensional spaces.” PhD Thesis, 2000.

[93] Yong Wang and Ian H. Witten, “Modeling for optimal probability prediction,” presented at the Nineteenth International Conference in Machine Learning, Sydney, Australia, 2002.

[94] David J.C. Mackay, “Introduction to Gaussian Processes.” 1998.

[95] A. J. Smola and B. Schölkopf, “A tutorial on support vector regression,” *Statistics and computing*, vol. 14, no. 3, pp. 199–222, 2004.

[96] C. Maffezzoni and F. Parigi, “Dynamic analysis and control of a solar power plant—II. Control system design and simulation,” *Solar Energy*, vol. 28, no. 2, pp. 117–128, 1982.

[97] F. R. Rubio, M. Berenguel, and E. F. Camacho, “Fuzzy logic control of a solar power plant,” *IEEE Transactions on Fuzzy Systems*, vol. 3, no. 4, pp. 459–468, 1995.

[98] A. J. Gallego and E. F. Camacho, “Adaptative state-space model predictive control of a parabolic-trough field,” *Control Engineering Practice*, vol. 20, no. 9, pp. 904–911, Sep. 2012.

[99] R. Pickhardt, “Nonlinea modelling and adaptive predictive control of a solar power plant,” *Control Engineering Practice*, vol. 8, pp. 937–947, 2000.

[100] M. R. Arahal, M. Berenguel, and E. F. Camacho, “Neural identification applied to predictive control of a solar plant,” *Control Engineering Practice*, vol. 6, no. 3, pp. 333–344, 1998.

[101] C. R. Cuttler and B. L. Ramaker, “Dynamic Matrix Control-A Computer Control Algorithm,” presented at the Joint Automatic Control Conference, San Francisco, 1980.

[102] J. Richalet, A. Rault, J. L. Testud, and J. Papon, “Model predictive heuristic control: Applications to industrial processes,” *Automatica*, vol. 14, no. 5, pp. 413–428, 1978.

[103] A. Bemporad, M. Morari, and N. L. Ricker, “Model Predictive Control Toolbox,” *User’s Guide, Version*, vol. 2, 2004.

[104] D. E. Seborg, T. F. Edgar, D. A. Mellichamp, and F. J. Doyle, *Process Dynamics and Control*, 3rd Edition. USA: John Wiley and Sons, Inc., 2011.

[105] P. Richmond and D. Chen, “A model predictive control package for undergraduate education,” *Education for Chemical Engineers*, vol. 7, no. 2, pp. e43–e50, May 2012.

[106] L. Wang, “A Tutorial on Model Predictive Control: Using a Linear Velocity-Form Model,” *Developments in Chemical Engineering and Mineral Processing*, vol. 12, no. 5–6, pp. 573–614, 2004.

[107] *LabVIEW*. National Instruments.

[108] M. Röper, “Fischer-Tropsch Synthesis,” in *Catalysis in C1 Chemistry*, Netherlands: Springer Verlag, 1983, pp. 41–88.

[109] D. E. Rivera and K. S. Jun, “An integrated identification and control design methodology for multivariable process system applications,” *IEEE Control Systems*, vol. 20, no. 3, pp. 25–37, Jun.

[110] L. Ljung, “System Identification Toolbox^{TM} 7,” *User’s Guide*, 1988.

[111] E. H. Bristol, “On a new measure of interaction for multivariable process control,” 1966.

[112] C. R. Madhuranthakam, A. Elkamel, and H. Budman, “Optimal tuning of PID controllers for FOPTD, SOPTD and SOPTD with lead processes,” *Chemical Engineering and Processing: Process Intensification*, vol. 47, no. 2, pp. 251–264, Feb. 2008.