Università
degli Studi
di Padova

Dipartimento
di Ingegneria
dell'Informazione

DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE

**CORSO DI LAUREA MAGISTRALE IN
CONTROL SYSTEMS ENGINEERING**

Implementation of a Black-box Learning-based Nonlinear Model Predictive Control for a Clinker Production Plant

Relatore: Prof. Mattia Bruschetta
Correlatori: Dott. Lorenzo Orlietti
Ing. Enrico Picotti

Laureando: Davide Scola

ANNO ACCADEMICO 2021 - 2022
Data di laurea 13 dicembre 2022

Abstract

Cement production plays a significant role in global CO_2 emissions. Advanced control algorithms could reduce its environmental impact by improving the efficiency of the process. Nonlinear Model Predictive Control (NMPC) is a technique particularly fit for this role, since it minimizes a cost function while satisfying a set of constraints. A key element required by NMPC is an accurate mathematical model of the controlled system. However, its derivation could be challenging, especially for complex systems such as cement production plants. As an alternative, learning-based approaches are being investigated. They leverage historical data to design the entire controller or part of its components. In this work, Gaussian Processes regression is used to obtain a black-box model of key system variables of a clinker production plant. A CV-informed GP-NOE model was trained on historical data and compared to a MIMO transfer function model. The results show slight improvements in long multi-step-ahead predictions. The developed model has the potential to be implemented within a learning-based NMPC framework to control the clinker production plant.

Sommario

La produzione di cemento svolge un ruolo significativo nelle emissioni globali di CO_2. Algoritmi di controllo avanzati potrebbero ridurne l'impatto ambientale tramite un miglioramento dell'efficienza del processo. Il controllo predittivo non lineare (NMPC) è una tecnica particolarmente adatta a questo ruolo, poiché minimizza una funzione di costo soddisfacendo una serie di vincoli. Un elemento chiave richiesto dal NMPC è un modello matematico accurato del sistema controllato. Tuttavia, la sua derivazione può risultare onerosa, soprattutto per sistemi complessi come gli impianti di produzione del cemento. Una possibile alternativa sono gli approcci learning-based, attualmente in fase di studio. Essi sfruttano i dati storici per lo sviluppo dell'intero controllore o di alcuni suoi componenti. In questo elaborato, tecniche di regressione gaussiana vengono utilizzate per ottenere un modello black-box delle variabili chiave di un impianto di produzione di clinker. Un modello CV-informed GP-NOE è stato addestrato su dati storici e confrontato con una funzione di trasferimento MIMO. I risultati mostrano leggeri miglioramenti nelle predizioni multipasso a lungo termine. Il modello sviluppato ha il potenziale per essere implementato all'interno di un approccio learning-based NMPC per il controllo dell'impianto di produzione di clinker.

Contents

List of Figures

List of Tables

List of Algorithms

1 Introduction

Cement is one of the most important building materials in the world. It is a fundamental component of concrete, essential for building construction and infrastructure development [12]. The estimated global cement production in 2021 was 4.40 billion metric tons, up 4.8% compared to the previous year [29]. The cement industry is energy-intensive and is one of the largest emitters of CO_2 in the world. In particular, clinker production is the process that contributes the most in terms of both energy demand and pollution. The intensity of direct emissions from cement production reached 0.59 metric tons of CO_2 per ton of cement in 2021, with an increase of approximately 1.5% per year since 2015. To get on track with the Net Zero Emissions by 2050 scenario (NZE), 3% annual reductions to 2030 are necessary [13]. In order to reduce CO_2 emissions while meeting demand, several innovative techniques are being developed and implemented. This process has been incentivized by regulatory action that mandates emission reductions and funds research and development. Larger kiln sizes, waste heat recovery, the use of alternative fuels and raw materials, improvements in grinding efficiency, and carbon capture and storage (CCS) technologies are among the main approaches currently investigated [25]. However, they can present a challenge and significant costs, as they require hardware modifications of the plant.

A different approach that has been developed in parallel to the aforementioned techniques and requires minimal changes in the equipment is energy efficient production automation [33]. In recent decades, the cement industry has seen an increasing adoption of automated control systems. The initial PID controllers were quickly replaced by more complex expert systems based on fuzzy logic. Nevertheless, after the initial success their adoption dwindled in favor of more advanced linear model predictive controllers (MPC). This approach has enjoyed widespread use in recent years due to its ability to handle constrained multi-input multi-output (MIMO) systems [6]. While early automatic control solutions were designed to aid human operators in their day-to-day management of the plant, the advanced capabilities offered by linear MPCs promoted the development of controllers aimed at optimizing production efficiency. One of the main challenges of using MPCs is their reliance on accurate models of the systems that they control. Mismatches can be handled at the cost of control performance. Cement factories are complex MIMO, time-varying, nonlinear systems characterized by multiple interacting chemical and physical processes [24]. This poses significant challenges to the modeling approach, which result in an unavoidable plant-model mismatch exacerbated by the need for linearization. A possible solution to this problem is the use of data-driven techniques to derive nonlinear gray-box or black-box models of the system. They can then be combined with nonlinear model predictive control,

to better suit the intrinsic structure of the clinker production plant [6]. This approach is called learning-based NMPC [10] and will be investigated in this work.

1.1 State of the Art

The modeling techniques of the clinker production systems found in the literature can be divided into two main categories: empirical models and predictive models [12]. The former rely on in situ data collection to derive simplified input-output relationships of the plant variables. Examples of this methodology can be found in [3], [30], and [36]. The latter use first principles derived from either physics, chemistry, or both to provide a description of the system dynamics. Physics-based approaches can vary considerably in complexity but are all based on energy and mass balance equations. Simpler methods rely on a "sinks and sources" description of the plant [28], while more refined models tend to compartmentalize the system [27]. The use of differential equations describing multiple phenomena, combined with detailed knowledge of multiple plant parameters, produces the most complex models found in the literature [8, 18]. Chemistry-based approaches rely on chemical equilibria to describe the dynamics of the system [12].

The main control technique applied to clinker production plants is linear model predictive control. As described previously, MPCs are often designed to minimize energy consumption while meeting production demand and quality requirements. Further constraints on the combustion process can be imposed to satisfy emission limits. The technique is quite well established, thus multiple examples of its application can be found in the literature [3, 18, 21, 27, 30]. In contrast, learning-based MPC is a very new active area of research, gathering the interest of the control community only in more recent years. An extensive literature review of the topic is provided in [10]. In it, the author groups the different implementations that are being actively researched within the LbMPC framework into three distinct classes:

- learning the system dynamics: this approach leverages machine learning techniques to automatically adjust the model representation of the system dynamics. It can either improve a gray-box model based on a priori knowledge by using data to compensate for unmodeled dynamics, or it can directly derive a black-box model of the system;
- learning the controller design: this technique uses data to shape the whole closed-loop MPC behavior by acting on its cost function, constraints, or terminal components;
- model predictive control for safe learning: this solution decouples the optimization of the cost function, which is achieved by the learning-based component, from the satisfaction of system constraints, which is satisfied by MPC techniques.

Since LbMPC techniques are in the early stages of research, limited applications to the

cement industry can be found in the literature. A single paper was found [33]. In it, the author used an identified NARMAX model combined with an NMPC for the control of a rotary kiln. Closed-loop simulation showed that the control goal could be met by the designed LbNMPC architecture.

1.2 Thesis Contributions

The goal of this work was to investigate learning-based techniques for modeling key system variables of a clinker production plant. In particular, Gaussian Processes Regression was applied to historical data collected at a cement factory to derive a black-box model capable of estimating the dynamics of the kiln motor current. The GP-NARX and GP-NOE models described in [15] were used as a base and expanded upon. Since the work was developed within the learning dynamics approach of LbNMPC, the GP model was evaluated on its capability of multi-step-ahead prediction. Further knowledge and characterization of the plant was provided by Alperia Green Future, which developed the current linear MPC implementation that manages the plant. Their model was also used as a benchmark to evaluate the performance of the GP-NOE produced in this work.

1.3 Thesis Outline

This work is organized as follows: background information on the manufacturing of cement is provided in chapter 2. A detailed process description is provided, together with an in-depth review of the state of the art in modeling and control of the clinker production phase. Lastly, the plant investigated in this work is described. chapter 3 offers an overview of the learning-based techniques used for system identification, including a short introduction to the LbMATMPC toolbox [20]. The results obtained are presented in chapter 4, where comparisons between the designed models and the current implementation are described. Lastly, in chapter 5 concluding remarks and future implications are discussed, together with a critical reflection on the limitations of the results obtained and possible solutions.

2 Cement Production

2.1 Concrete Composition

Concrete is a man-made composite material widely used in construction. In its simplest definition, it is made of a filler and a binder. The filler, also known as aggregate, is composed of coarse granular material. Different types of filler exist; they are the inert component of conventional concrete and make up approximately three-quarters of its total volume. The remaining volume is constituted by the binder. Two main categories of binders can be identified on the basis of their composition: organic and inorganic. Nowadays, the most popular inorganic binder is Portland cement; this is due to the wide availability of the raw materials used in its production and to its ease of use. Portland cement is made of 95-97% clinker and 3-5% gypsum or calcium sulfate. Clinker itself is obtained by sintering the basic raw materials: calcium carbonate (limestone), aluminum silicate (argillaceous materials), and corrective materials (silicious, ferruginous, or aluminous) inside a rotary kiln as part of the cement manufacturing process. [17]

2.2 Manufacturing of Portland Cement

Portland cement manufacturing involves a number of steps and advanced operations that are summarized in Figure 2.1. In particular, the so-called dry-process will be described because of its almost exclusive use in modern plants. The procedure starts with the procurement of raw materials, usually from local quarries. Of these, the most important is limestone, followed by clay minerals and, when required, corrective materials, such as sand, iron ore, and bauxite. After being sourced, the different minerals are crushed and milled to obtain fine powders of the desired particle size distribution. Various crushing and milling machines are used in the industry, including hammer and ball mills. The raw mix design is then performed. This is a process in which the proportions of the different raw materials are calculated using equations derived from chemistry. The resulting mixture, called raw meal, has the desired composition and burning behavior. These characteristics ensure the quality of the cement produced while minimizing overall energy consumption and manufacturing costs. [4]

The raw meal then enters the pyroprocessing stage. This is regarded as the main phase of the cement manufacturing process since the most important chemical reactions take place here. Through them, the raw meal is transformed into clinker, thus this stage is also called clinkering process. To perform pyroprocessing, modern plants employ a five-stage preheater-precalciner kiln system, illustrated in Figure 2.2. Inside it, the following chemical

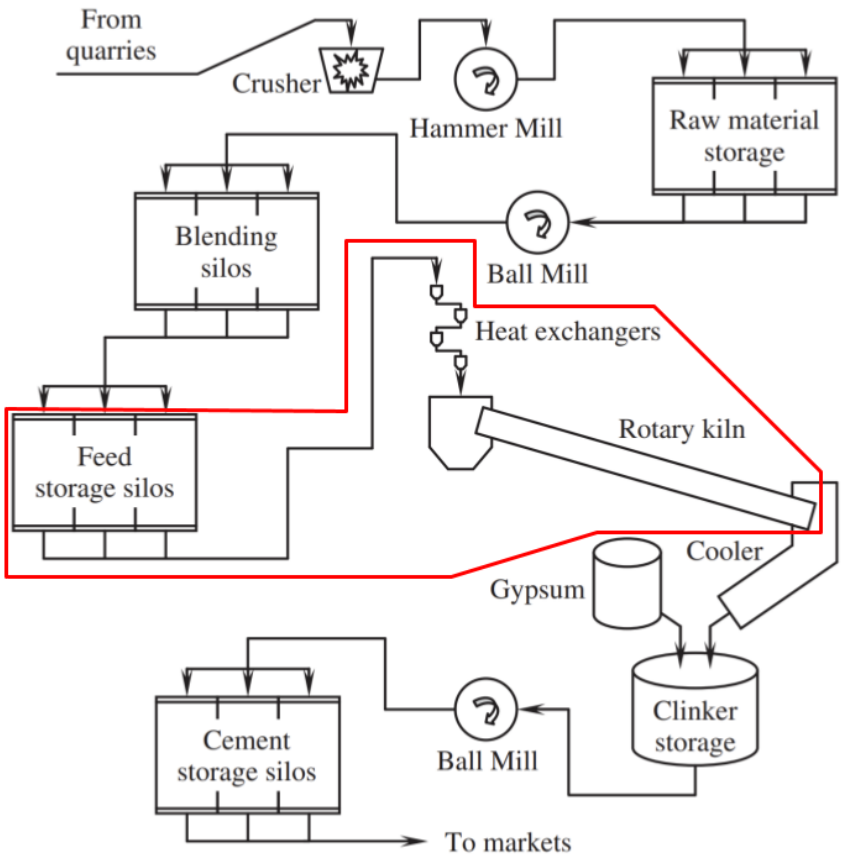

Figure 2.1: Simplified diagram of a Portland cement manufacturing plant. The clinker production phase is highlighted in red. Source: own edit of [17].

processes take place:

- dehydration and dehydroxylation: removal of H_2O and OH respectively, occurring between 27°C and 600°C;
- decarbonation of limestone, also called calcination: removal of CO_2, occurring between 550°C and 1000°C;
- solid-state reactions of lime and other oxides, occurring between 550°C and 1280°C;
- liquid-phase sintering, occurring between 1280°C and 1450°C;
- consolidation of the clinker microstructure through cooling, occurring between 1000°C and 1300°C. [5]

The clinkering process begins with the raw meal being fed into the preheating system, which is composed of four stages of cyclones, also known as suspension preheaters. Here, the mixture, pulled down by gravity, encounters an upward stream of hot gases that exit from the rotary kiln and are drawn by an induced draft fan (ID fan). The temperature of the raw meal increases from ambient conditions to approximately 800°C, leading to complete dehydration and dehydroxylation. Furthermore, the limestone is calcinated to a degree of about 30%.

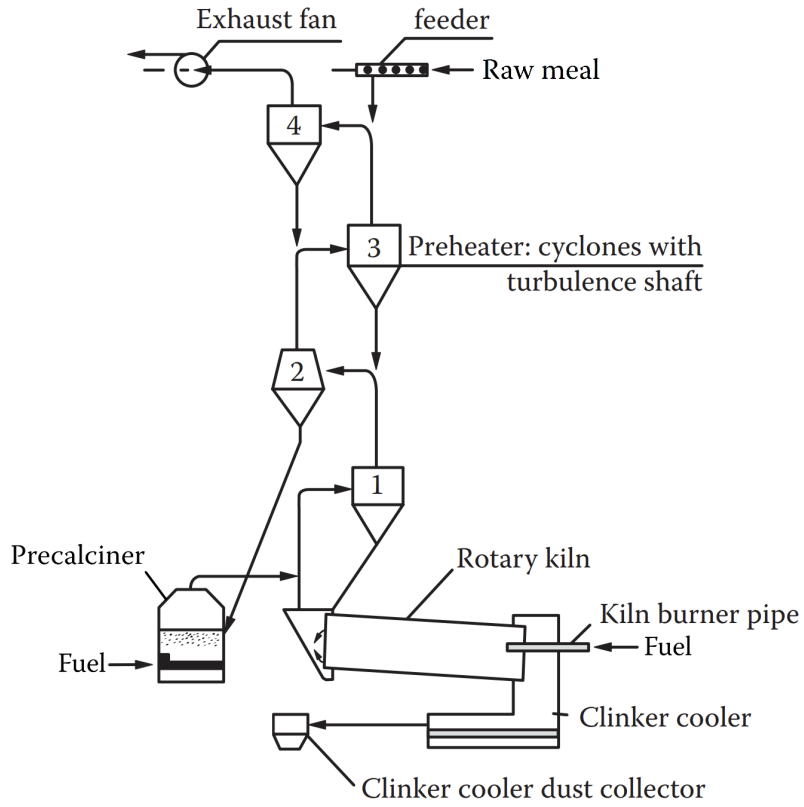

Figure 2.2: Schematic diagram of a five-stage preheater-precalciner kiln system. Source: own edit of [4].

Next in line is the precalciner, a specialized combustion chamber in which the raw meal is entered before reaching the rotary kiln. Its role is to further increase the extent of limestone decarbonation while avoiding excessively high temperatures and oxygen levels. This reduces the thermal load of the kiln, allowing more of its total length to be reserved for solid-state reactions and liquid-phase sintering. Various precalciner designs have been employed in the industry due to the multiplicity of secondary goals to be fulfilled, these include:

- allowing the use of alternative fuels, thus reducing the energy and environmental costs of cement production. Rates of up to 100% alternative fuels have been achieved for decades and their use is now spreading to the kiln burner as well, although slowly due to their impact on the operation of the kiln and the quality of the clinker [25];
- reducing NO_X emissions, to comply with regulatory norms on pollution;
- achieving a controllable degree of limestone decarbonation that ranges between 85% and 95%.

The rotary kiln is a slanted steel cylinder that rotates about its longitudinal axis. The mix exiting the preheater-precalciner stage is fed into the kiln through its upper end. As the cylinder rotates, the material slowly moves downward toward the lower end, where a multiple-channel burner is placed. It utilizes a combination of conventional fuels, such as petroleum coke (petcoke), air, and optionally alternative fuels, to produce a stable

flame. Since the heat source is located at one end of the kiln, a temperature gradient develops inside of it. This allows the completion of calcination, solid-state reactions, and liquid-phase sintering to occur in succession. After these chemical processes take place, the raw meal exits the cylinder as clinker.

The last step of the pyroprocessing stage is the cooling of the clinker. This serves multiple functions:

- ensuring the consolidation of the clinker microstructure;
- reducing the temperature of the clinker for further conveying, storing and processing;
- recovering a significant amount of the clinker heat for reuse in the previous steps.

The first portion of the cooling occurs inside the kiln, after the material has passed the burning zone. Here, the temperature drops to approximately 1200-1250°C. After being discharged, the clinker enters a cooler. Although a wide variety of models are used in the industry, their goal is the same: to quickly cool the clinker down to about 100°C. This slow-fast cooling schedule has been shown to produce the best clinker quality.

The final step in Portland cement manufacturing consists in grinding the clinker together with 3-5% gypsum and optionally other additives. The role of gypsum is to prevent flash setting, i.e. the quick stiffening of cement following the addition of water. Other additives are used to control the cement's pH value; this is of particular interest when concrete is reinforced with steel rebar. In the industry, a variety of grinding equipment, including ball mills, is used to obtain cement of the desired particle size distribution. After this step, the cement is stored and prepared for dispatch to markets. [17]

2.3 Challenges in Modeling and Control of Cement Production

Cement production plants come in many different configurations. Variability is present in both the plant design and in the individual machinery used at each step [4], as previously described. Furthermore, due to their decades-long lifespans, plants generally experience a number of changes to their process structure. For these reasons, it is convenient to divide plant models and control algorithms into separate generic compartments [27]. These can then be tuned to match different plants, resulting in a reduction of engineering and commissioning costs. Modeling and control techniques for cement mills [23], precalciners [28], rotary kilns [27], preheater-precalciner and kiln systems [3], and clinker coolers [31] have been presented in the literature. All these compartments exhibit challenges due to their nature; in fact, they are multivariable, strongly-coupled, time-varying, nonlinear, large time-delay systems [24].

This work focuses on the modeling and control of the pyroprocessing stage. In particular,

only the preheating, precalciner and kiln systems are of concern, thus excluding the cooling phase. For this reason, further descriptions of techniques proposed in the literature are now presented.

2.3.1 Modeling of Preheater-Precalciner Kiln Systems

The main challenge faced by engineers when modeling preheater-precalciner kiln systems is due to their complexity. In fact, the clinkering process consists of multiple chemical and physical subprocesses, which are distributed in time and space across the production plant [12]. The approaches presented in the literature can be categorized according to their characteristics. Empirical models use data collected in situ to derive simplified relationships between input and output variables. In particular, open-loop step response experiments are performed to estimate the parameters of a multi-input multi-output (MIMO) first-order transfer function with pure time delay as in Equation 2.1:

$$G(s) = \begin{bmatrix} g_{11} & \cdots & g_{1n} \\ \vdots & \ddots & \vdots \\ g_{m1} & \cdots & g_{mn} \end{bmatrix} \qquad \text{with} \qquad g_{ij}(s) = e^{-\theta_{ij}s}\frac{K_{ij}}{\tau_{ij}\, s + 1} \tag{2.1}$$

The gain K_{ij}, the time delay θ_{ij}, and the time constant τ_{ij} of each transfer function $g_{ij}(s)$, linking the j-th input to the i-th output, are evaluated. The main downside of this approach is its low robustness against process parameter variations. Examples of this methodology can be found in [3] and [30].

In contrast, predictive models rely on first principles to provide a description of the plant's dynamics. These can be based either on physical or chemical models [12]. The former utilize mainly approaches deriving from energy and mass balance equations. Simpler methods, such as the one presented in [28], leverage a "sinks and sources" representation to obtain a discrete model of the heat energy present in the system, given in Equation 2.2.

$$Q[k+1] = Q[k] + \sum_i q_i[k] - \sum_j q_j[k] \tag{2.2}$$

In it, $Q[k]$ represents the heat of the system at time k, q_i are the heat inflows ("sources") and q_j are the heat outflows ("sinks"). An adaptive bias term is included to handle possible variations in the model parameters. The mass balance is represented by a transport model consisting of a series of time delays. They account for the different transport modes of the various mass inputs to the system (raw meal, fuel, and air).

More complex models, such as the one illustrated in [27], segment the pyroprocessing stage into compartments. The thermodynamic relations and mass transport dynamics are then described separately for each section. In contrast to the previous approach, the raw meal and gas temperature dynamics are modeled individually. Equation 2.3 describes the

thermodynamics of the raw meal in a given section.

$$c^{rm} m \dot{T}^{rm} = \overbrace{\frac{c^{rm} u^{in}}{L}(u^T - T^{rm})}^{\Delta \mathrm{E}^{\text{raw meal}}} + \overbrace{k^t (T^g - T^{rm})}^{\mathrm{E}^{\text{transfer}}} + \overbrace{\frac{u^v m k^c}{L}}^{\mathrm{E}^{\text{chem}}} - \overbrace{\frac{k^l T^{rm}}{L}}^{\mathrm{E}^{\text{loss}}} \tag{2.3}$$

The parameters listed in Table 2.1 are used to derive a more advanced sinks and sources representation that takes into account the energy transfers due to mass flow, gas and raw meal interactions, chemical reactions, and environmental losses. An equivalent equation

Table 2.1: Description of the parameters of Equation 2.3. Adapted from [27].

Symbol	Description
c^{rm}	specific heat capacity of raw meal
k^c	energy source (+1) or sink (-1) due to chemical reaction
k^l	energy loss in the compartment
k^t	heat transfer coefficient between gas and raw meal
L	compartment length
m	average mass density
T^g	gas temperature
T^{rm}	raw meal temperature
u^{in}	raw meal input flow
u^T	raw meal input temperature
u^v	raw meal transport velocity, proportional to kiln rotational speed

is used to describe the gas thermodynamics. The dynamics of mass transport for the raw meal assume homogeneous mass distribution in each section and are described by Equation 2.4.

$$\dot{m} = \frac{u^{in} - m \cdot u^v}{L} \tag{2.4}$$

This methodology allows to obtain a sufficiently accurate model that can satisfy the existing computational limitations.

The most advanced methods, such as the one presented in [18], extend the energy and mass balance approach by taking into account the geometry of the different components of the plant. The primary focus is on the kiln and on the distribution of the raw meal inside of it. The clinker bed behavior is modeled, its angle of repose inside the kiln is necessary to accurately predict its interaction with the gas phase. The air mass balance is also taken into account. Furthermore, the thermodynamic role of the walls of the kiln is included in the model. The final result is a set of partial differential equations that are then discretized in space to convert them into ordinary differential equations. Overall, approximately 100 parameters are used to describe the system, thus requiring high levels of expertise to apply this approach to new plants. Physical models based on finite element methods are also present in the literature, e.g. [1].

Chemical models, such as the one illustrated in [12], rely on chemical equilibria to represent

the dynamics of the preheater-precalciner kiln system. The main chemical compounds that make up the raw meal are: calcium carbonate $CaCO_3$, silicon dioxide SiO_2, aluminum oxide Al_2O_3, and iron oxide Fe_2O_3. The first comes primarily from limestone, while the other three are found mostly in clay and sand. Often in the literature on cement chemistry, the chemical formulae for these and other compounds are replaced by their abbreviations; a list of the most common ones can be found in Table 2.2. The dominant reactions that take place during the pyroprocessing stage are as follows:

- decarbonation of limestone $CaCO_3 \longrightarrow CaO + CO_2(g)$;
- formation of belite $2\,CaO + SiO_2 \longrightarrow (CaO)_2 \cdot SiO_2$;
- formation of alite $3\,CaO + SiO_2 \longrightarrow (CaO)_3 \cdot SiO_2$;
- formation of aluminate $3\,CaO + Al_2O_3 \longrightarrow (CaO)_3 \cdot Al_2O_3$;
- formation of ferrite $4\,CaO + Al_2O_3 + Fe_2O_3 \longrightarrow (CaO)_4 \cdot Al_2O_3 \cdot Fe_2O_3$. [11]

Table 2.2: Common abbreviations of chemical compounds used in cement literature, also known as cement chemist notation.

Abbreviation	Chemical formula
C	CaO
A	Al_2O_3
S	SiO_2
F	Fe_2O_3
T	TiO_2
M	MgO
K	K_2O
N	Na_2O

Figure 2.3 illustrates how the different chemical species and their relative proportions change throughout the conversion from raw meal to clinker. As the temperature increases, the different reactions listed above take place. The work presented in [12] uses them to model the behavior of the plant. The system is first separated into seven sections and then for each one chemical equilibria are used to predict the dynamics of the process. The parameters required for the different calculations are: standard enthalpy of formation, absolute standard entropy, coefficients of heat capacity, thermodynamic data on the different chemical compounds, their amounts, pressures, and temperatures. This approach has the advantage of relying on fixed parameters derived from scientific principles. The physics-based methodologies described above take into account the chemistry of the process to a very limited extent. In [28] and [27] a simple model of oxygen dynamics is included, while [18] uses a more comprehensive description of the combustion process and of the behavior of oxygen concentration. None of the works takes into account chemical equilibria.

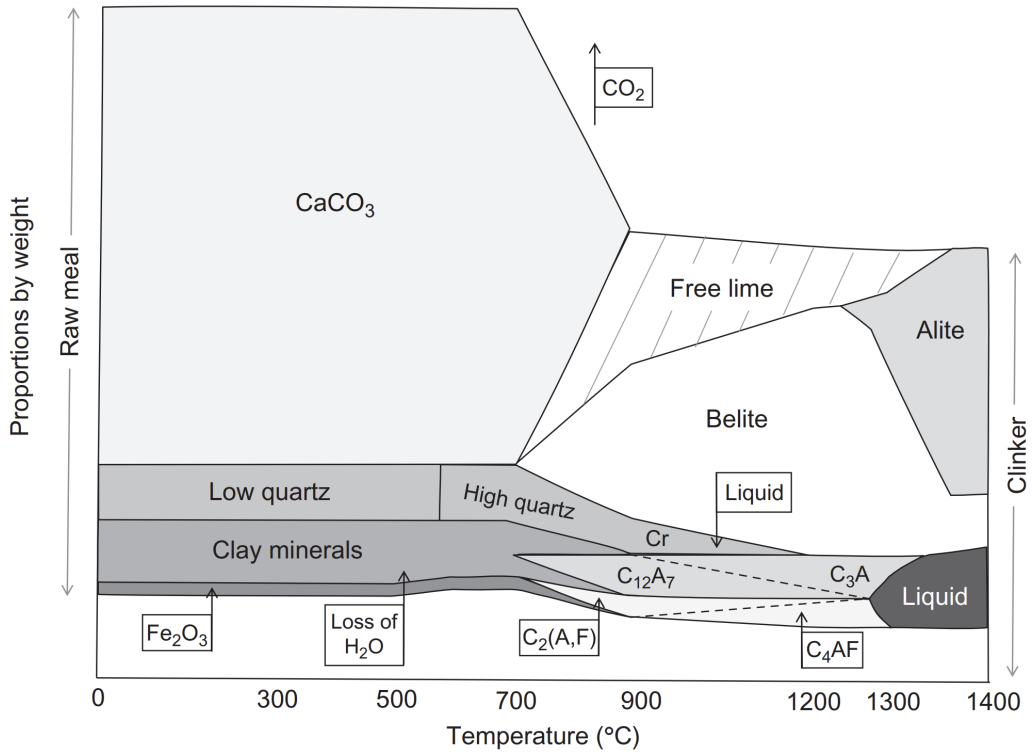

Figure 2.3: Proportions of different chemical species throughout the conversion of raw meal to clinker. Cr is an abbreviation for cristobalite. Source: [11].

2.3.2 Control of Preheater-Precalciner Kiln Systems

As highlighted above, the pyroprocessing stage is a multi-input multi-output, strongly-coupled, nonlinear process. Large time delays and time-varying relationships further increase the complexity of the system. Due to these characteristics, control of precalciner-preheater kiln systems has historically been difficult. The first approaches relied on proportional-integral-derivative (PID) controllers. Equation 2.5 shows the PID control function.

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \tag{2.5}$$

The control action at a given time instant $u(t)$ depends on the error, which is defined as the difference between the reference value for the system output and its measured value $e(t) = r(t) - y(t)$. The proportional K_p, integral K_i and derivative K_d terms are three tunable parameters that can be adjusted to obtain the desired control characteristics. Later, more advanced strategies began to be employed in the field. Expert systems were developed during the 1980s and 1990s. They relied on rules-based fuzzy logic to achieve the desired plant behavior. Figure 2.4 illustrates the working principle of the control architecture. The main component of a fuzzy logic controller (FLC) is the knowledge base. It comprises a data base containing the definitions of the membership functions and a rule base listing the "if-then" statements used for decision making. In the first step, the error $e(t)$ is fed to the FLC as input and undergoes fuzzification. This procedure uses membership functions to assign the value of $e(t)$ to one or more fuzzy sets. Next,

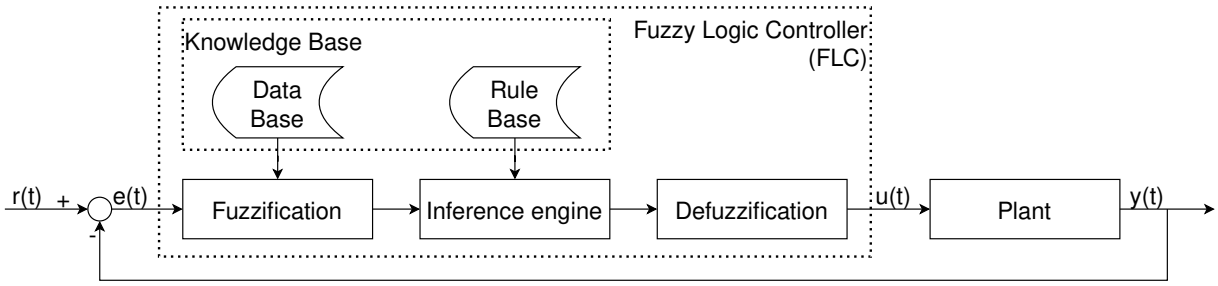

Figure 2.4: Diagram of a fuzzy logic control scheme applied to a generic plant.

the inference engine computes the actions to take according to the rule base. Lastly, the output $u(t)$ is converted into numerical values through the process of defuzzification and it is applied to the plant. A feedback loop allows the controller to assess whether the applied control action yielded the desired plant behavior. After their initial success, the adoption of expert systems dwindled as they failed to achieve the required performance. This was due to the poor robustness against disturbances, the lack of sufficient knowledge bases, the inadequacy of human expertise, and the challenging adaptation and maintenance of the rule bases to different operating conditions. [6]

In recent years, linear model predictive control (MPC) has been widely applied in the cement production industry. Its success stems mainly from its inherent capability to handle constrained MIMO processes with time delays. One of the defining features of MPC is the use of a model to predict the future behavior of the system given a set of inputs. This is used to simulate the forced evolution of the system throughout a given control horizon H_c, after which the final control action is kept constant over the prediction horizon H_p. The optimal input set $\mathcal{U} = [u_0, \cdots, u_{H_c-1}]$ is obtained by minimizing the following cost function:

$$J(k) = \sum_{i=1}^{H_p} \|y(k+i) - r_{mpc}(k+i)\|^2_{Q(i)} + \sum_{i=0}^{H_c-1} \|\Delta u(k+i)\|^2_{R(i)} \tag{2.6}$$

This defines the control goal of reaching the desired output reference $r_{mpc}(k)$ applying the minimal control effort required $\Delta u(k)$. Q and R are weight matrices used to adjust the importance of each goal. After the optimal control sequence is determined, only the first control output $u^*(k)$ is applied to the plant. Its effect is measured, and the optimization is computed again starting from the new initial conditions in a technique called receding horizon control. As presented in subsection 2.3.1 a variety of approaches are available to obtain a model capable of describing the pyroprocessing dynamics. However, their accuracy is the key factor that influences the control performance of the MPC. Having considered this, it must be noted that the intrinsic structure of preheater-precalciner kiln systems demands the use of nonlinear model predictive controllers. This technique is described in detail in chapter 3. [6, 16]

2.4 Investigated Clinker Production Plant Description

The plant investigated in this thesis is a five-stage preheater-precalciner kiln system that uses a dry-process to produce clinker. A schematic of the system is provided in Figure 2.5. A total of four manipulated variables (MVs), one disturbance variable (DV), and 18

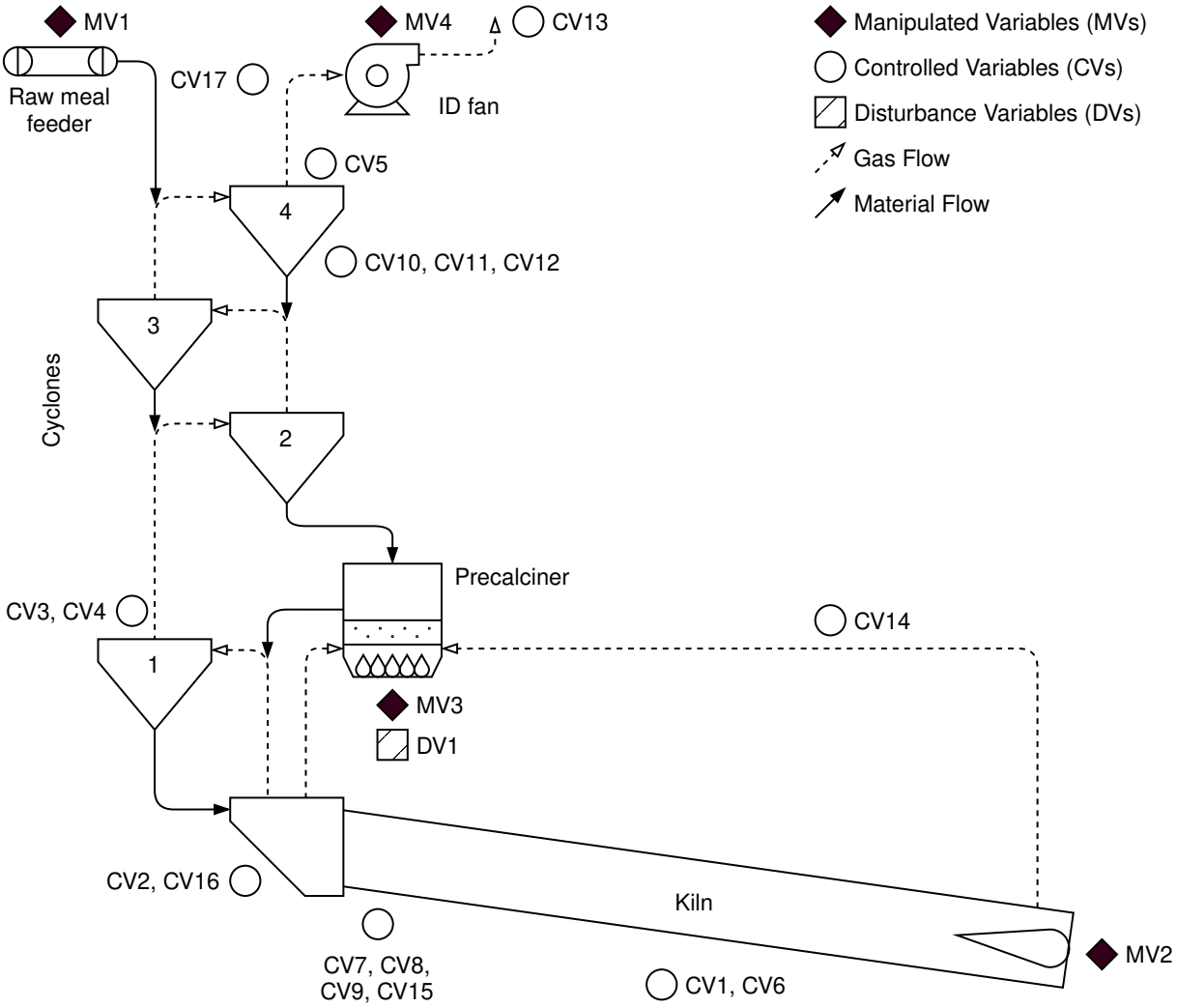

Figure 2.5: Schematic diagram of the investigated clinker production plant, highlighting the different manipulated, controlled and disturbance variables.

controlled variables (CVs) are monitored. Table 2.3 summarizes their names, descriptions, and units of measure. The MVs are the inputs to the system, they include the flow rates for raw meal and fuel, and the induced draft fan speed. DV1 is the setpoint for the flow rate of alternative fuels that burn inside the precalciner. This is manually controlled by the plant operators; thus, although it is an input to the plant, it is treated as a disturbance, since it is not directly regulated by an automated process. The CVs are the outputs of the plant. They were selected by plant managers because they can be used to evaluate production performance. [36] They include the current drawn by the kiln motor, which indirectly measures the overall load of the system, various temperatures distributed throughout

Table 2.3: List of all manipulated, disturbance and control variables of the investigated clinker production plant.

Variable	Description	Units
MV1	Raw meal input flow rate	t/h
MV2	Kiln fuel input flow rate	t/h
MV3	Precalciner fuel input flow rate	t/h
MV4	Induced draft fan speed	rpm
DV1	Precalciner alternative fuels input flow rate (manual setpoint)	t/h
CV1	Kiln motor current	A
CV2	Kiln duct temperature	°C
CV3	First stage right cyclone exit temperature	°C
CV4	First stage left cyclone exit temperature	°C
CV5	Fifth stage cyclone exit temperature	°C
CV6	Kiln combustion chamber temperature	°C
CV7	Kiln inlet oxygen O_2 concentration	%
CV8	Kiln inlet carbon monoxide CO concentration	%
CV9	Kiln inlet nitrogen oxides NOx concentration	ppm
CV10	Fourth stage cyclone oxygen O_2 concentration	%
CV11	Fourth stage cyclone carbon monoxide CO concentration	%
CV12	Fourth stage cyclone nitrogen oxides NOx concentration	ppm
CV13	Exhaust outlet oxygen O_2 concentration	%
CV14	Precalciner tertiary air temperature	°C
CV15	Kiln inlet depression	mmH_2O
CV16	Kiln duct depression	mmH_2O
CV17	Exhaust outlet temperature	°C
CV18	Calculated specific heat consumption	kcal/kg

the plant, redundant oxygen, carbon monoxide, and nitrogen oxide gas concentrations for improved reliability, and depressions in key sections of the system. Lastly, CV18 is a calculated variable expressed as the ratio of total calorific input E_{tot} to raw meal input m_{rm}:

$$\text{CV18} = \frac{E_{tot}}{m_{rm}} = \frac{\Delta H_{c,k} \cdot m_{c,k} + \Delta H_{c,p} \cdot m_{c,p} + \Delta H_{af,p} \cdot m_{af,p}}{m_{rm}} \tag{2.7}$$

where $\Delta H_{.,.}$, $m_{.,.}$ represent the heat of combustion and the mass of coal $c, .$ or alternative fuels $af, .$ fed to the kiln $., k$ or to the precalciner $., p$. Measurements of all variables are collected and stored with a sampling time of 30 seconds.

The current control system has been developed by Alperia Green Future. It is a linear model predictive controller aimed at increasing plant productivity and efficiency while ensuring product quality and limiting pollution and fuel consumption. A detailed description of the methodology used can be found in [34, 35, 36]. After defining the control inputs, outputs, and disturbances as illustrated above, a black-box modeling approach was employed to derive a model of the system. A series of step tests were performed on the manipulated variables and on the disturbance variable, while the controlled variables were monitored. The collected data was then used to obtain a MIMO first-order transfer function with

pure time delay that relates the MVs to the CVs, as shown in Equation 2.1. Some of the entries have zero value, as it was determined that there is no direct relation between some MV-CV combinations. The resulting control matrix is similar to the one presented in [3]. First-order filters for various controlled variables were also designed to reduce measurement noise before feeding the data to the MPC. The controller has been developed with a 60 second sample time, a control horizon of 10 steps (600 seconds), and a prediction horizon of 60 steps (3600 seconds). The cost function minimized by the linear MPC at a generic time instant k is the following:

$$V_{ss}(k) = c_{MV}^T \cdot \Delta MV_{ss} + c_{CV}^T \cdot \Delta CV_{ss} + \rho_{ss\_min}^T \cdot \varepsilon_{ss\_min} + \rho_{ss\_max}^T \cdot \varepsilon_{ss\_max} \tag{2.8}$$

In it, c_{MV}^T and c_{CV}^T are the costs associated with the control action ΔMV_{ss} and the plant output ΔCV_{ss}, while ε_{ss_min} and ε_{ss_max} are the slack variables for CV constraint relaxations with their respective weights $\rho_{ss\_min}^T$ and $\rho_{ss\_max}^T$. The MPC's objective is to minimize fuel consumption (MV2 and MV3) while ensuring that the constraints for all CVs are met. This is achieved by setting a positive, non-zero cost only for the relevant manipulated variables. The upper and lower bounds of each CV can be manually adjusted by plant operators to account for changes in operating conditions. Additionally, a decoupling selector is available, this allows to exclude any controlled variable from the MPC formulation, which is useful for redundant sensors or in the case of defective ones. The current control formulation significantly improved performance compared to the PID controllers installed previously, however it is incapable of fully capturing the process nonlinearities, leading to sub-optimal performance.

3 GP-based LbNMPC

In recent years, model predictive control has experienced widespread applications in a variety of fields. Its ability to handle constrained MIMO systems makes it particularly suitable for process control, which includes the cement production industry, as described in chapter 2. A sufficiently accurate model capable of describing the plant's behavior is at the core of the MPC. If this condition is not met, the reliability and performance of the controller can deteriorate significantly. As illustrated previously, obtaining a descriptive model of a cement plant, and in particular of the pyroprocessing stage, is a daunting task that requires high levels of expertise and possibly a multidisciplinary approach. Recent developments in computational power, sensing, and data collection capabilities, combined with advances in the field of machine learning, have sparked renewed interest in automated controller design techniques. Learning-based model predictive control (LbMPC), as the technique is known formally, is quickly gathering the attention of the control community. This approach combines machine learning techniques with advanced control strategies; it has the potential to improve performance while reducing the burden of model design and controller tuning. Different implementations are being actively researched within the LbMPC framework, they are listed in section 1.1. This work uses the learning dynamics approach described in [10] to derive a black-box model of the clinker production plant described in section 2.4. This choice was motivated by several factors. First, the inability to access the real plant or a high-fidelity reproduction of it prevented the evaluation of the closed-loop MPC performance necessary for a learning design approach. Second, the lack of prior knowledge of the system precluded the use of a gray-box model. Last but not least, LbMATMPC, an open-source toolbox, was recently developed and tested [20], and is described below.

3.1 Gaussian Processes Regression

A Gaussian Process is defined as *"a collection of random variables, any finite number of which have (consistent) joint Gaussian distributions."* [22]. A GP is fully characterized by its mean function $m(\mathbf{x})$ and its covariance function $k(\mathbf{x}, \mathbf{x}')$, also known as kernel, and it defines a distribution in the form

$$f(\mathbf{x}) \sim \mathcal{GP}(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}') \tag{3.1}$$

In the framework of system identification by means of GP, the mean function $m(\mathbf{x})$ can be used to encode a priori knowledge of the system. However, if the identification and validation data are preprocessed to ensure zero mean, $m(\mathbf{x}) = 0$ can be assumed without

loss of generality. One of the most commonly used covariance functions $k(\mathbf{x}, \mathbf{x}')$ when assuming smooth model behavior is the Squared Exponential (SE); its equation is the following:

$$k(\mathbf{x}, \mathbf{x}' \mid \theta) = \sigma_y^2 \exp\left[-\frac{1}{2}\sum_{d=1}^{D}\frac{(\mathbf{x}_d - \mathbf{x}'_d)^2}{l_d^2}\right] \tag{3.2}$$

where $\theta = \{\sigma_y, l_d\};\ d = 1, ..., D$ is the vector of kernel parameters to be optimized during training, σ_y^2 is the signal variance, also called vertical scaling factor, that represents the possible variations of the function, D is the number of regressors or training inputs, and l_d is the set of horizontal length scales. The case in which different length scales are used for different regressors is called Automatic Relevance Determination (ARD). This is an embedded method for regression selection, meaning that it is carried out during the model optimization procedure. The resulting values of l_d can be used to identify which inputs to the GP contribute more to the prediction of the outputs. Different techniques can be used to train Gaussian Processes models; this work uses log marginal likelihood maximization within the Statistics and Machine Learning Toolbox available in MATLAB [19]. After optimizing the model parameters, it is possible to calculate the predictive distribution of a new test point x_*, which is Gaussian. Its maximum a posteriori estimator corresponds to the posterior mean given by the following equation:

$$\bar{f}(x_*) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*) \tag{3.3}$$

where $k(\mathbf{x}_i, \mathbf{x}_*)$ is the vector of covariances between the new test point and the n training points, and

$$\alpha = \left(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \mathbb{I}\right)^{-1} \mathbf{y} \tag{3.4}$$

is a vector of kernel functions, each one centered on one of the training points. It is realistic to assume that the observations used for fitting the GP present white Gaussian noise $\nu \sim \mathcal{N}(0, \sigma_n^2)$ with zero mean and σ_n^2 variance. In Equation 3.4 $\mathbb{I} \in \mathbb{R}^{n \times n}$ is the identity matrix. In the case where the mean $m(\mathbf{x})$ of the Gaussian Process in Equation 3.1 is different from zero, it can be defined by a linear basis function $\mathbf{h} = [1, \mathbf{x}]$. The equations of the posterior mean and of the vector of kernel functions then become:

$$\bar{f}(x_*) = \mathbf{h}(\mathbf{x}_*)^T \beta + \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}_*) \tag{3.5}$$

$$\alpha = \left(k(\mathbf{x}, \mathbf{x}') + \sigma_n^2 \mathbb{I}\right)^{-1} (\mathbf{y} - \mathbf{h}\beta) \tag{3.6}$$

where the vector of parameters β is optimized during the training of the GP model, together with the kernel parameters and the Gaussian noise covariance σ_n^2. [14, 22, 32]

One of the challenges of Gaussian process prediction is that it scales as $\mathcal{O}(n^3)$. Therefore, large-scale datasets ($n > 10,000$), like the one considered in this work, require the use of

approximations. A straightforward but effective method is the Subset of Data (SoD or SD) technique. It consists of selecting an active set of size $m < n$ from the whole training dataset, thus reducing the complexity to $\mathcal{O}(m^3)$. The selection of the active set is made so that the approximate GP achieves sufficiently accurate predictions. Different methods exist, as this is an active area of research. After the kernel parameters are optimized, the resulting GP is a nonlinear model of the system that generated the training data. [14, 32]

3.1.1 GP-NARX and GP-NOE Models for System Identification

Consider historical input-output data collected from a nonlinear system. Denote by y the measured plant output, by u the plant inputs, and by ν the measurement noise. The goal of system identification methods is to derive a model capable of estimating the output $\hat{y}$ from a set of regressors $\mathbf{x}$ as:

$$\hat{y} = f(\mathbf{x}) + \nu \tag{3.7}$$

Gaussian Processes can be applied to this problem. Two common structures used in this context are the Nonlinear AutoRegressive model with eXogenous input (NARX), and the Nonlinear Output Error model (NOE) [14]. They play different roles, as the former is used for prediction, while the latter is applied to simulation problems. Their characteristics are now described in more detail.

The NARX structure, also known as equation-error or series-parallel model, uses the past m input values $u(k-m)$ and n output values $y(k-n)$ as regressors for the GP. The value to be estimated by the model is the output at the next step. This procedure is called prediction and can then be formulated as follows:

$$\begin{aligned} \hat{y}(k) &= f(\mathbf{x}(k-1)) + \nu \\ &= f\left(y(k-1), y(k-2), ..., y(k-n), u(k-1), u(k-2), ..., u(k-m)\right) + \nu \end{aligned} \tag{3.8}$$

Figure 3.1 shows a schematic representation of the GP-NARX model.

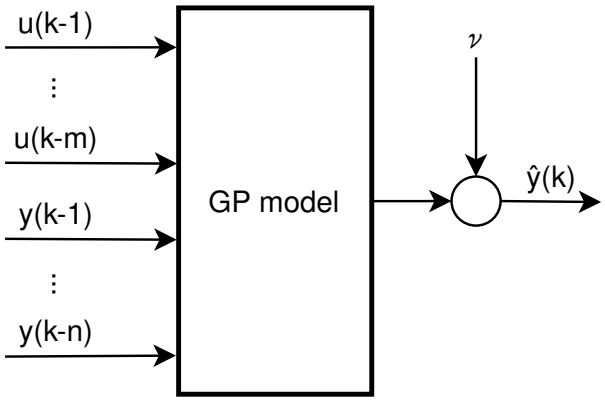

Figure 3.1: GP-NARX model, where the output prediction $\hat{y}(k)$ is a function of m previous inputs $u(k-m)$ and n previous outputs $y(k-n)$; ν is Gaussian noise. Source: [14]

The NOE structure, also known as output-error or parallel model, is used to perform simulation. The goal of this procedure is the multi-step-ahead prediction of the output over a desired horizon of length p. The GP-NOE methodology achieves this by using past predictions to obtain future ones. In other words, the prediction $\hat{y}(k)$ is a function of the past m inputs $u(k-m)$ combined with the past n predicted outputs $\hat{y}(k-n)$.

$$\begin{aligned}\hat{y}(k) &= f(\mathbf{x}(k-1)) + \nu \\ &= f\left(\hat{y}(k-1), \hat{y}(k-2), ..., \hat{y}(k-n), u(k-1), u(k-2), ..., u(k-m)\right) + \nu\end{aligned} \tag{3.9}$$

The new output $\hat{y}(k)$ is then used to predict $\hat{y}(k+1)$. This procedure is applied iteratively until the desired horizon $\hat{y}(k+p)$ is reached. Figure 3.2 displays a schematic representation of this approach.

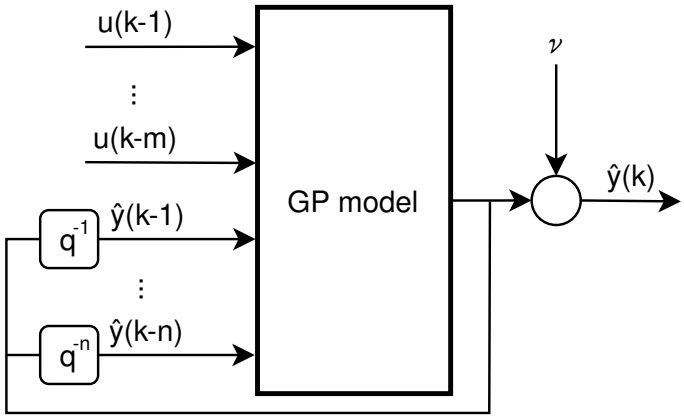

Figure 3.2: GP-NOE model, where the output prediction $\hat{y}(k)$ is a function of m previous inputs $u(k-m)$ and n previous estimated outputs $\hat{y}(k-n)$; q^{-1} is the backshift operator and ν is Gaussian noise. Source: [14]

3.2 LbMATMPC Toolbox

LbMATMPC is an off-the-shelves instrument for the implementation of learning-based nonlinear MPC. It uses Gaussian Processes Regression (GPR) for gray-box and black-box modeling, combined with MATMPC, a MATLAB-based fast nonlinear MPC solver [7]. Given the strong nonlinearities that characterize the investigated preheater-precalciner kiln system, the use of NMPC should result in better control performance compared to the linear MPC implementation described in section 2.4. Furthermore, Gaussian Processes are particularly suitable for capturing complex nonlinear relationships [14, 26]. This property is beneficial as manual nonlinear model design can prove to be particularly challenging. The black-box GP model can be trained from input-output data collected by interaction with the system.

3.2.1 Nonlinear Model Predictive Control

NMPC is an extension of model predictive control that uses nonlinear system models to perform prediction. This makes its combination with system identification by means of GP straightforward. MATMPC offers a number of algorithmic modules for implementing NMPC strategies. Its functioning is now described. Similarly to linear MPC the goal is to solve an Optimal Control Problem (OCP) by minimizing a cost function, which in this case is nonlinear. The first step is to apply direct multiple shooting to the OCP over the desired prediction horizon. This returns a Non-Linear Programming (NLP) problem, which can be solved by means of Sequential Quadratic Programming (SQP). This iterative procedure reformulates the NLP into a series of approximate QP subproblems. MATMPC offers a number of QP solvers for added flexibility. The procedure is repeated at every time step k of the control task. [7]

4 Results

The purpose of this work was to investigate the application of learning-based nonlinear model predictive control techniques to the clinker production phase of a cement factory. In particular, real historical data collected at a cement plant was used to train a Gaussian Processes regression model with the goal of later integrating it with an NMPC. This approach was pursued within the learning dynamics framework described in chapter 3. It leverages machine learning techniques, in this case GP, to obtain a black-box model of the system dynamics, which can then be used by a nonlinear model predictive controller to compute the optimal control action needed to reach the desired plant behavior. In this chapter, the results obtained with the different methodologies selected are described and their interpretation is provided.

4.1 Data Processing

The raw dataset was provided by Alperia Green Future. It contained 146,881 samples of the four manipulated variables, one disturbance variable, and 18 controlled variables described in Table 2.3. The measurements were collected every 30 seconds between the 7^{th} of January 2020 and the 26^{th} of February 2020. After a first inspection of the data, it was noticed that the plant was not operational for a significant portion of the time, spanning from January 13^{th} to February 4^{th}. Furthermore, faulty sensor readings from some CVs were observed immediately after the startup until February 10^{th}. During the remainder of the time, the plant was operational and its dynamics were controlled by the linear model predictive controller described in section 2.4. For these reasons, the original dataset was divided into:

- training dataset containing 46,080 samples collected between February 11^{th} and 26^{th};
- test dataset containing 11,520 samples collected from January 8^{th} to 11^{th}.

Before being used to optimize the GP and test its performance, the two datasets were preprocessed. First, they were subsampled with a sampling time of 60 seconds, matching that of the linear MPC. This ensures that the predictions are compatible with the NMPC formulation. Next, a first-order filter, in the form

$$H(s) = \frac{1}{\tau s + 1} \tag{4.1}$$

was applied to all CVs. The time constant τ is different for each one of them and was determined by Alperia Green Future during the design of the linear MPC. This procedure removes some of the measurement noise and improves the controller's performance. A

comparison between unfiltered and filtered measurements of CV1 is shown in Figure 4.1. In this case, the original data are particularly noisy; thus, the time constant was set to

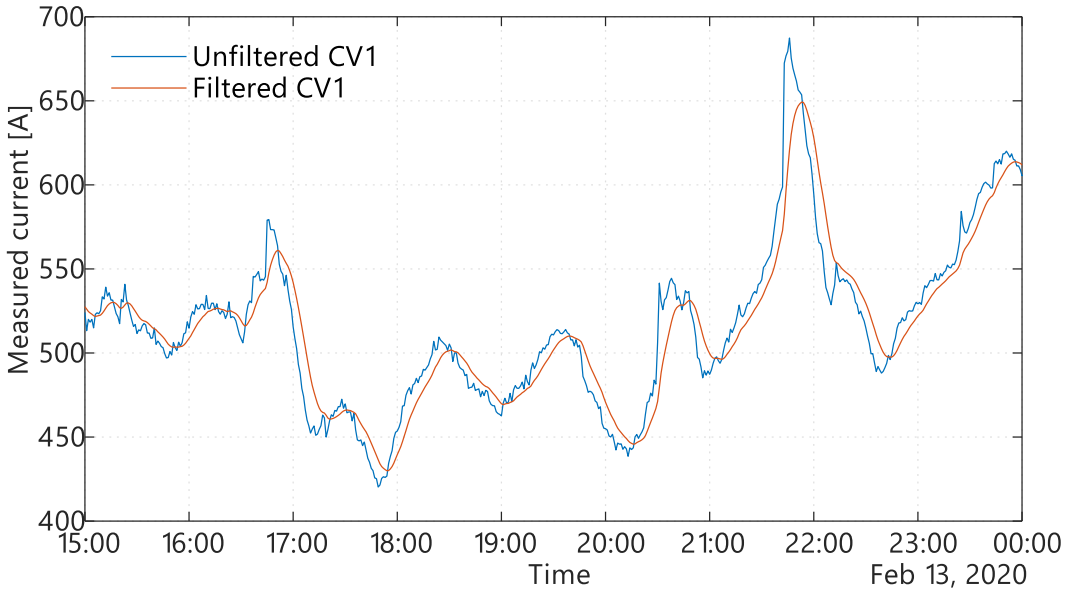

Figure 4.1: Plot comparing the unfiltered and filtered measurements of CV1 collected during a 9 hour window on February 13$^{\text{th}}$ 2020.

$\tau = 360s$ and is the largest among all CVs. It can be observed that the application of the first-order filter reduces noise at the expense of introducing a delay. The last stage of the preprocessing procedure was the manual removal of faulty sensor reads. Only a few controlled variables required this step, namely CV7, CV8, CV10, CV11, and CV18. The oxygen sensors (CV7 and CV10) presented negative readings and sudden out-of-scale peaks. The carbon monoxide sensors (CV8 and CV11) showed only instances of negative measurements. Lastly, the calculated variable CV18 presented extremely high peaks, due to the denominator term of Equation 2.7 reaching values close to zero. After the faulty data points were identified, they were replaced by applying a zero-order hold to the last value recorded before the error.

4.2 GP Models for the Prediction of CV1

Together with the engineers at Alperia Green Future, controlled variable 1 was identified as the most relevant for the clinker production process. CV1 records the current fed to the kiln motor required to power its rotation. It can also be interpreted as an indirect measure of the plant's activity, since it is directly proportional to the amount of material being processed inside the rotary oven. For these reasons a black-box Gaussian Processes model capable of predicting the one-step-ahead value of CV1 was developed. The different methodologies applied are now detailed.

4.2.1 Simple GP-NARX Model

The first implementation investigated was the Nonlinear AutoregRessive model with eXogenous input (NARX) structure described in subsection 3.1.1. In this approach, the past m input values $u(k-m)$ and n output values $y(k-n)$ are used as the GP regressors. The control matrix designed by Alperia Green Future was used to aid in the selection of the relevant plant inputs. According to their work [36] the behavior of CV1 is directly impacted by the raw meal flow rate and the fuel flow rates, thus excluding the ID fan speed. For this reason, the inputs selected as regressors of the GP-NARX model are the following:

$$u(k) = [\mathrm{MV1}(k), \mathrm{MV2}(k), \mathrm{MV3}(k), \mathrm{DV1}(k)] \tag{4.2}$$

Different values of maximum input lag m were tested, namely 2, 4, and 6 minutes. Regarding the choice of output delay $n = 3$ was selected, as it showed better performance compared to other values. The squared exponential covariance function with ARD, reported in Equation 3.2, was chosen. This embedded method for regressor selection automatically adjusts the weights of the input features, tuning out less important ones. A linear basis function $\mathbf{h} = [1, \mathbf{x}]$ was considered in order to better capture the variations of the output. The prediction was then computed as reported in Equation 3.5. Lastly, a subset of data approximation was used; an active set of size $m_{as} = 1500$ was selected because it achieved a good compromise between accuracy and computational execution time.

The results of the one-step-ahead prediction on training and test datasets for the case $m = 4$ are reported in Figure 4.2. It can be seen that the accuracy of the estimation is comparable between the two datasets. A substantial difference between the two, with poorer performance in the test dataset, would be an indication of overfitting [32]. Another observation is that the model performance decreases when the measured data experience positive and negative peaks. Trend inversions appear to be slightly more challenging to predict for the GP-NARX. The metric used to evaluate the quality of the one-step-ahead prediction is the coefficient of determination R^2 [9], defined as:

$$R^2 = 1 - \frac{\sum_{i=1}^{p}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{p}(y_i - \bar{y})^2} \tag{4.3}$$

where p is the size of the dataset, y are the measured values of CV1, $\bar{y}$ is their mean value, and $\hat{y}$ are the corresponding estimated values generated by the GP-NARX model. The coefficient of determination is a goodness-of-fit measure. Values close to 1 indicate that the model predictions accurately capture the variance present in the true data. As R^2 decreases, so does the quality of the estimates provided by the model being evaluated. Negative values can arise in the case of nonlinear regression and indicate extremely poor performance.

Since the GP model obtained has to be implemented within the LbNMPC framework,

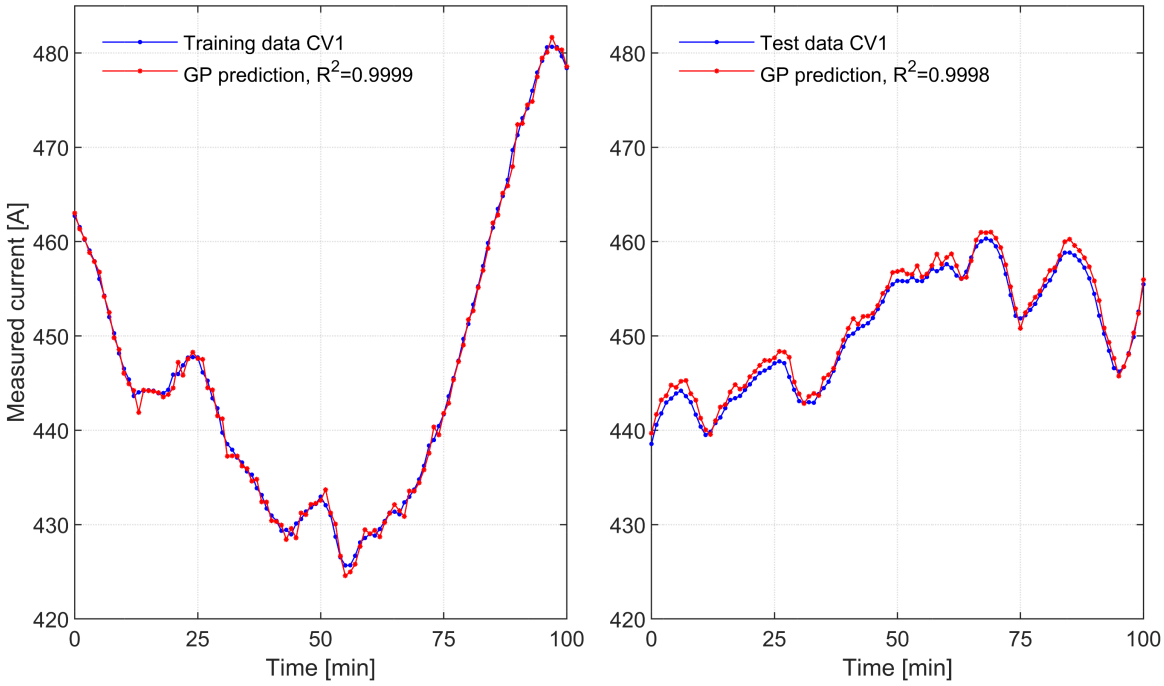

Figure 4.2: Plot of a 100 minute long detailed view of the one-step-ahead predictions (in red) on training (left) and test (right) datasets obtained using the GP-NARX model with input lag $m = 4$.

its purpose is the multi-step-ahead prediction over a desired horizon. This technique, also known as simulation, can be achieved through the use of a NOE model as described in subsection 3.1.1. In this context, a more representative metric of the GP model's performance is the average R^2 value over multiple prediction horizons:

$$ {}^{p}\bar{R}^2 = \frac{1}{d}\sum_{i=1}^{d} R_i^2 \tag{4.4} $$

where ${}^{p}\bar{R}^2$ is the average coefficient of determination for the prediction horizon of length p and $d = s_t - p$ is the number of simulated horizons defined by the size of the test dataset s_t. Throughout this work, five different horizons, with a duration p of 5, 10, 20, 30, and 60 minutes, were used to evaluate the performance of each model. The three GP-NARX models described previously were converted to GP-NOE models and used to perform simulations. The five horizons of length $p = [5, 10, 20, 30, 60]$ were simulated starting at each of the points contained in the test dataset; the simulation results $\hat{y}$ were then compared to the true output y by means of $\bar{R}^2$. The results for the three models, with input lag m of 2, 4, and 6 minutes, are reported in Table 4.1. It can be observed that all three models present large negative values of ${}^{p}\bar{R}^2$ for every prediction horizon investigated. This is indicative of poor simulation performance, thus making these models unsuitable for control purposes. Furthermore, a trend is present in the results. Increasing the input lag, which corresponds to using a higher number of regressors, reduces the simulation performance of the GP model. This could be due to the fact that the extra regressors,

Table 4.1: ${}^{p}\bar{R}^2$ values for three GP-NOE models with input lag $m = [2, 4, 6]$. The results were obtained by averaging all simulations of a given horizon over the whole test dataset. * indicates the best model.

Model input lag	Prediction horizon p [min]				
	5	10	20	30	60
$m = 2^*$	-31.12	-33.72	-40.70	-44.48	-28.14
$m = 4$	-34.55	-37.82	-47.71	-49.58	-28.63
$m = 6$	-32.22	-36.01	-52.94	-60.82	-59.84

being highly correlated, do not provide additional information regarding the system's dynamics. This could lead to worse results as the optimization of the model becomes more complex. [14]

4.2.2 GP-NARX Model with Extended Input Delays

The approach presented in the previous subsection was extended to include prior knowledge about the system. In fact, Alperia Green Future provided a matrix of input/output delays relating all MVs and CVs. These values were estimated during the design of the linear model predictive controller described in [36]. The identified delays between the plant inputs and controlled variable 1 are reported in Table 4.2. This information was used to

Table 4.2: Delays between plant inputs and controlled variable 1 expressed in seconds and in time steps. MV4 is not directly related to CV1 thus no delay was provided.

	MV1	MV2	MV3	MV4	DV1
CV1 [seconds]	1200	900	900	n/a	3900
CV1 [steps]	20	15	15	n/a	65

augment the set of regressors used for prediction by the GP-NARX model. Rather than using m consecutive past values of the input u, a larger time window m_d, determined by the set of delays, was selected. Due to the large number of regressors resulting from this approach, not every value was used; instead, the inputs were subsampled every 5 steps (300 seconds). Furthermore, the time window was extended by one step. This was done to account for the additional delay introduced by the first-order filters during the preprocessing step and for possible inaccuracies in the delay estimation. The resulting inputs are the following:

$$
\begin{aligned}
u_{\mathrm{MV1}}(k) &= [MV1(k-25), MV1(k-20), ..., MV1(k-5), MV1(k)] \\
u_{\mathrm{MV2}}(k) &= [MV2(k-20), MV2(k-15), MV2(k-10), MV2(k-5), MV2(k)] \\
u_{\mathrm{MV3}}(k) &= [MV3(k-20), MV3(k-15), MV3(k-10), MV3(k-5), MV3(k)] \\
u_{\mathrm{DV1}}(k) &= [DV1(k-70), DV1(k-65), ..., DV1(k-5), DV1(k)] \\
u(k) &= [u_{\mathrm{MV1}}(k), u_{\mathrm{MV2}}(k), u_{\mathrm{MV3}}(k), u_{\mathrm{DV1}}(k)]
\end{aligned}
\tag{4.5}
$$

The output delay was set to $n = 3$ as described in the previous subsection. The same selection of linear basis function, SE covariance function, and SoD approximation was made. A GP-NARX model was trained using the aforementioned regressors. The results of the one-step-ahead prediction on training and test datasets are comparable to those obtained in subsection 4.2.1.

The obtained GP model was then used within the NOE framework to perform simulation. Five horizons of length $p = [5, 10, 20, 30, 60]$ minutes were simulated, and the resulting ${}^{p}\bar{R}^2$ values are reported in Table 4.3. Compared to the three models tested previously,

Table 4.3: ${}^{p}\bar{R}^2$ values for the GP-NOE model with extended input delay. The results were obtained by averaging all simulations of a given horizon over the whole test dataset.

Model	Prediction horizon p [min]				
	5	10	20	30	60
Extended input delay	-24.67	-25.92	-36.08	-46.44	-45.60

this implementation achieves better performance over horizons shorter than 20 minutes. However, the quality of the prediction degenerates for longer horizons: the 30 minute one is comparable with that of the best model presented previously, and the 60 minute one is noticeably worse. Figure 4.3 is a plot of two simulations obtained with both the $m = 2$ GP-NOE model of subsection 4.2.1 (in red) and the GP-NOE model with extended input delays (in black). From it, it is possible to observe the behavior emerging in the ${}^{p}\bar{R}^2$ values. Compared to the measured values of CV1 (in blue), the GP-NOE with extended input delays shows qualitatively good prediction capabilities at the beginning of the simulation

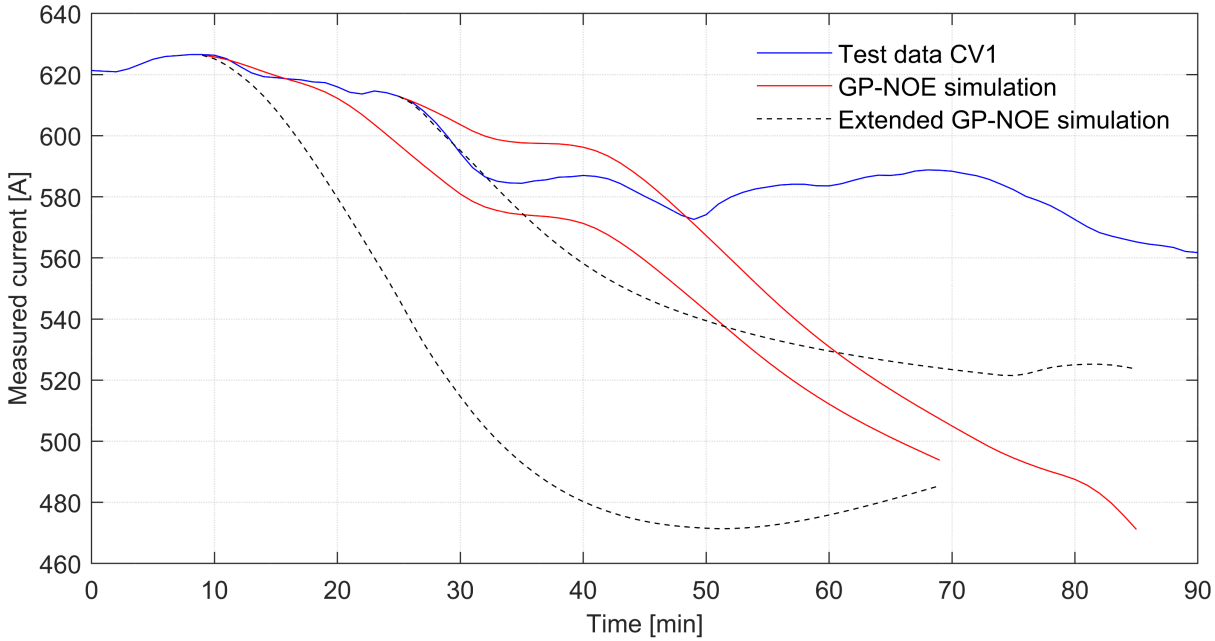

Figure 4.3: Comparison of four simulated 60 minute horizons obtained with the $m = 2$ GP-NOE model introduced in subsection 4.2.1 (in red) and with the GP-NOE model with extended input delays (in black). The blue line represents the ground truth.

horizon; however, it accumulates a large error as time goes on. On the other hand, the simple GP-NOE can perform less adequately at the beginning of the horizon, but over time, its prediction stays closer to the ground truth. Figure 4.3 has mostly qualitative value, as it represents a small fraction of the data, nevertheless it can be explicative for the overall trend reported in Table 4.3. Having considered this, it is clear that the large negative ${}^{p}\bar{R}^2$ values are indicative of poor simulation performance and that the GP-NARX model with extended input delays is not suitable for control applications.

4.2.3 CV-informed GP-NARX Model

The performance of the models presented in the previous subsections cannot be considered satisfactory. For this reason, a further extension of the regressor set was performed. Since the provided dataset contains measurements of all controlled variables, they were used to increase the number of inputs for the Gaussian Processes regression. This GP formulation, augmented by the knowledge of other controlled variables, was denominated "CV-informed" GP. The same output delay $n = 3$ used for CV1 was applied to all other CVs. A squared exponential kernel with automatic relevance determination was used to train a GP model with input $u = [MVs, DV, CVs]$. The lengthscales obtained through the optimization were used to choose the most relevant regressors needed for the one-step-ahead prediction of CV1. Furthermore, Alperia Green Future provided their expert knowledge of the plant to aid in the selection. This step was performed with the goal of using only the necessary information needed by the model; this is important to avoid excess complexity [14]. The final set of controlled variables selected as regressors is the following:

$$\begin{aligned} u_{\mathrm{CVi}} &= [CVi(k-3), CVi(k-2), CVi(k-1), CVi(k)] \\ u_{\mathrm{CVs}} &= [u_{\mathrm{CV2}}, u_{\mathrm{CV3}}, u_{\mathrm{CV4}}, u_{\mathrm{CV6}}, u_{\mathrm{CV7}}, u_{\mathrm{CV8}}, u_{\mathrm{CV10}}, u_{\mathrm{CV11}}] \end{aligned} \tag{4.6}$$

The MVs and DV with extended input delays and CV1 with output delay $n = 3$ were also used to train the CV-informed GP-NARX model as described in subsection 4.2.2. The same choices of covariance function, basis function, and approximation were made. Once again, the one-step-ahead prediction on training and test datasets showed results comparable to those described previously.

In the CV-informed GP-NOE formulation, the estimation at any step of the prediction horizon $\hat{y}(k + t_p)$ is a function not only of the plant inputs, but also of the controlled variables $u_{\mathrm{CVs}}(k + t_p - 1)$. For $t_p > 1$ the GP-NOE requires knowledge of future CV measurements to estimate the plant output. Since these values cannot be directly accessed, two approaches were considered to solve this problem. In the first one, called "updating-CVs", it was assumed that accurate predictions of the other CVs were accessible. This assumption is reasonable, as GP-NOE models capable of simulating the relevant CVs can be obtained by applying the same approach used for CV1. In this case, real measured data was used as other models were not available. The second approach, called "fixed-CVs",

does not assume knowledge of the future values of the CVs. Instead, it keeps the last measured value constant throughout the simulation horizon. This is valid assuming that over sufficiently short time frames a variable is subject to small changes. Figure 4.4 is a schematic representation of this method. Compared to Figure 3.2, it can be observed that

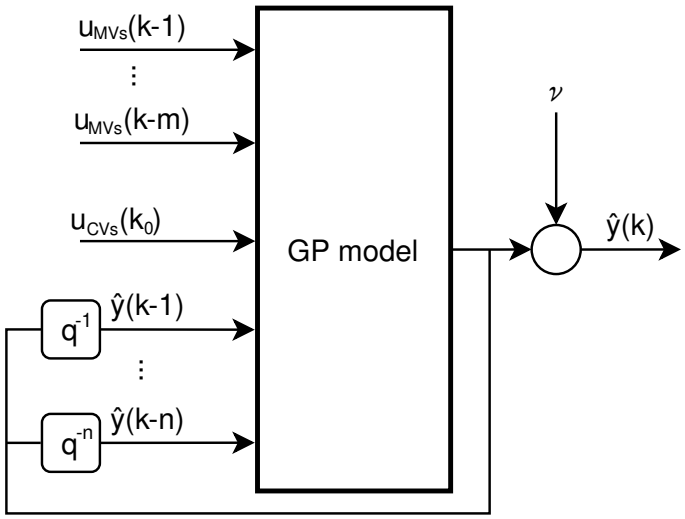

Figure 4.4: CV-informed GP-NOE model, where the output prediction $\hat{y}(k)$ is a function of m previous inputs $u(k-m)$, the measured CVs at time k_0, and n previous estimated outputs $\hat{y}(k-n)$; q^{-1} is the backshift operator and ν is Gaussian noise. Source: own edit of [14]

the GP model uses an additional input $u_{\text{CVs}}(k_0)$ for its estimates. This is the measured value of the CVs used as regressors at the beginning of the simulation horizon, and is kept constant for every prediction step k. This corresponds to the following prediction equation:

$$\hat{y}(k) = f\left(\hat{y}(k-1), ..., \hat{y}(k-n), u(k-1), u_{\text{CVs}}(k_0)\right) + \nu \tag{4.7}$$

with $u(k-1)$ defined as $u(k)$ in Equation 4.5 backshifted by one step.

The simulation results for both the updating-CVs and fixed-CVs approaches are reported in Table 4.4. As with previous models, prediction horizons $p = [5, 10, 20, 30, 60]$ were calculated throughout the test dataset. The updating-CVs model achieves a simulation performance similar to that of the best GP-NARX model of subsection 4.2.1, with marginally degraded results for the longest prediction horizon $p = 60min$. All ${}^{p}\bar{R}^2$ values are negative and indicate an extremely poor ability to capture the variance present in the true output data. Considering the large amount of additional inputs used to train

Table 4.4: ${}^{p}\bar{R}^2$ values for the updating-CVs and fixed-CVs GP-NOE models. The results were obtained by averaging all simulations of a given horizon over the whole test dataset.

Model	Prediction horizon p [min]				
	5	10	20	30	60
Updating-CVs	-30.39	-29.65	-38.75	-47.68	-33.13
Fixed-CVs	-37.98	-34.94	-56.22	-72.81	-52.94

the Gaussian Processes regression, the optimization appears to be unable to use them effectively to extract additional information about the process dynamics. This could be the result of different phenomena. The additional regressors could have a low prediction value or the large size of the training dataset could negatively impact the optimization process. The fixed-CVs model, on the other hand, shows significantly worse performance, particularly for longer simulation horizons. This result is expected because the assumption of small changes in the variables can be applied only to short time frames. The use of old data, inadequate to describe the current state of the plant, causes the CV-informed GP-NOE model to produce the worst predictions presented so far. Therefore, an accurate simulation of the dynamics of each controlled variable is required. Since a complete implementation of the LbNMPC architecture introduced in chapter 3 would be based on the design of a GP-NOE model for all CVs, the simulation capability described should be readily available.

4.2.4 GP-NARX Model of Rate of Change

The simulation capabilities obtained by training a GP-NARX model to perform one-step-ahead prediction of the value of CV1 cannot be considered sufficient. Extending the set of regressors resulted in sporadic marginal improvements at the cost of computational effort. The feedback action provided by a nonlinear model predictive controller can partially compensate for the large model-plant mismatch; however, this results in a loss of control performance [2]. In order to reduce the simulation error, a different training objective was selected. A GP model was optimized for the prediction of the rate of change (RoC) of CV1. This approach is comparable to the one described in subsection 4.2.1, and its prediction equation is:

$$\Delta\hat{y}(k) = f\left(y(k-1), y(k-2), ..., y(k-n), u(k-1), u(k-2), ..., u(k-m)\right) + \nu \quad (4.8)$$

The ground truth was obtained from the measurements of CV1 as $\Delta y(k) = y(k+1) - y(k)$. This technique was selected because it should be able to better capture the variations of the controlled variable. It can be seen as a simplification of the GP-NARX approach with linear basis function reported in Equation 3.5. Since $\mathbf{h}$ is the mean function of the Gaussian process, it should be able to approximate the overall trend of the data while the covariance function describes the remaining complexity. By considering Δy as the prediction goal, the variations of the measurements can be captured directly while setting $\mathbf{h} = 0$. This has the added benefit of reducing the number of parameters that must be optimized during model training.

Three simple GP-NARX models for the one-step-ahead prediction of the rate of change of CV1 were trained. They used input lag values $m = [2, 4, 6]$ respectively. No additional regressors were selected. An SE kernel with ARD was chosen, together with an SoD approximation of active set size $m_{as} = 1500$. The results of the training and test datasets

for the case $m = 4$ are shown in Figure 4.5. The first behavior that can be observed

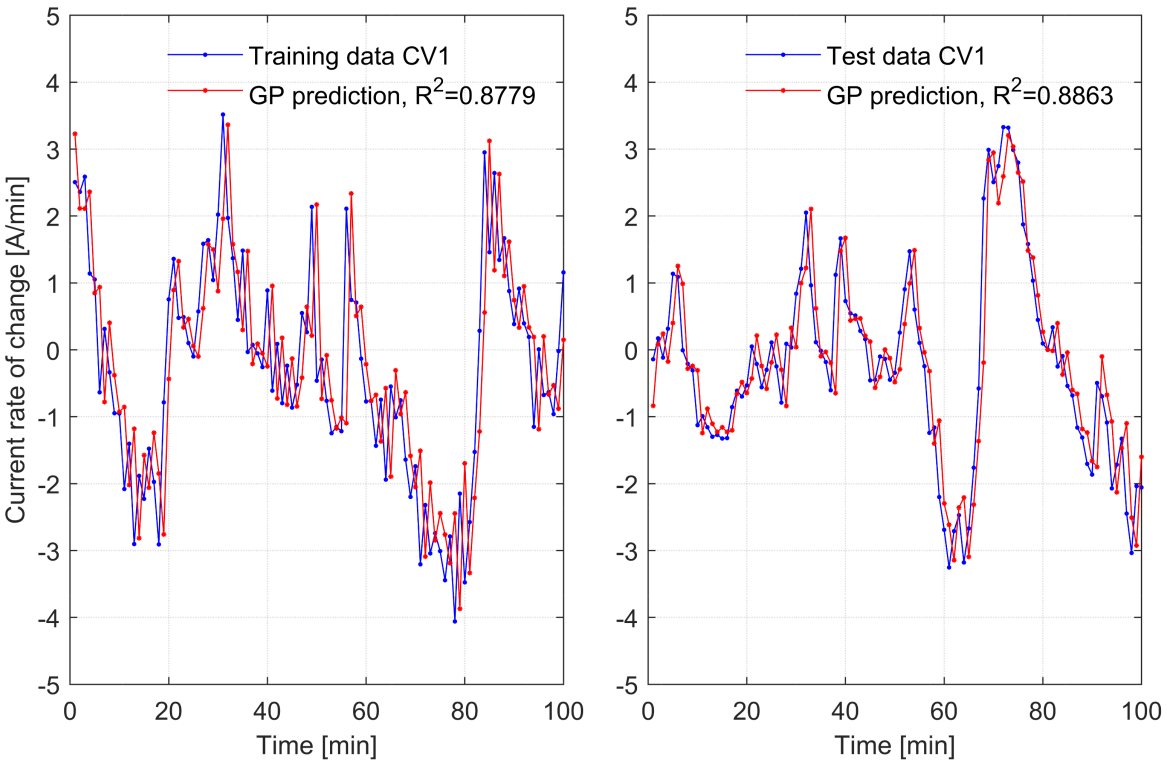

Figure 4.5: Plot of a detailed view of the one-step-ahead predictions (in red) of the RoC of CV1 on training (left) and test (right) datasets obtained using the GP-NARX model with input lag $m = 4$.

in both plots is a one-step delay between the ground truth and the model predictions. This indicates that the trained GP-NARX model has the tendency to act as a zero-order hold. The value of the RoC at the previous time step can be calculated by the model as $\Delta y(k-1) = y(k) - y(k-1)$ since it has access to both required values. In this case, the value of R^2 is not necessarily indicative of the prediction capabilities of the model and can be misleading. The same plots were produced for the GP-NARX model with $m = 4$ trained to directly predict the value of CV1, presented in Figure 4.2. The RoC was obtained by applying its definition to the prediction results $\Delta\hat{y}(k) = \hat{y}(k+1) - \hat{y}(k)$. The results are shown in Figure 4.6. Although significantly noisier, the predictions for both training and test datasets display a one-step delay similar to the one observed in Figure 4.5. This behavior is undesirable, as it produces seemingly good results for the one-step-ahead prediction which do not translate into satisfying simulation performance.

As in previous subsections, the trained GP-NARX model was used within the NOE framework to perform simulation. The ${}^{p}\bar{R}^2$ of the three different models for prediction horizons of length $p = [5, 10, 20, 30, 60]$ minutes are reported in Table 4.5. As expected, poor performance can be observed across all tests. This confirms that the models underestimate the variance of the true data. This error, while overlooked in the one-step-ahead predictions, is amplified by the iterative process applied during simulation. Large negative values

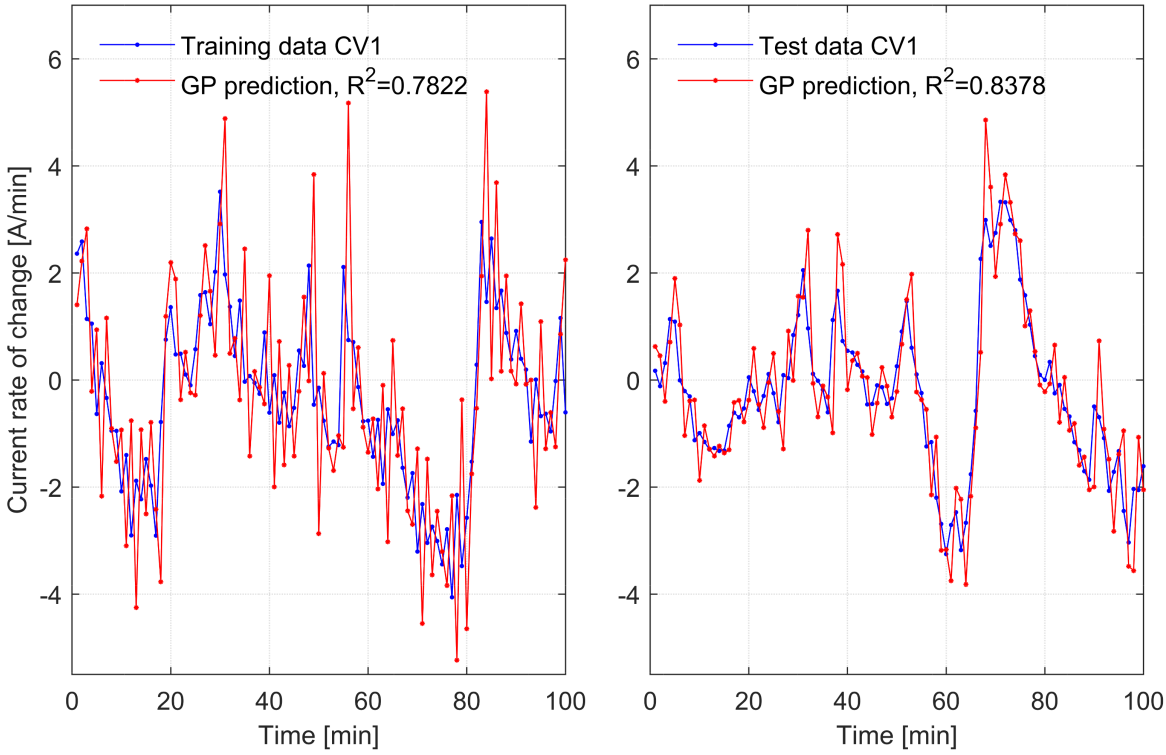

Figure 4.6: Plot of a detailed view of the RoC of CV1 obtained from the one-step-ahead predictions (in red) of the GP-NARX model with input lag $m = 4$ introduced in subsection 4.2.1.

Table 4.5: ${}^{p}\bar{R}^2$ values for three GP-NOE models with input lag $m = [2, 4, 6]$. The results were obtained by averaging all simulations of a given horizon over the whole test dataset.

Model input lag	Prediction horizon p [min]				
	5	10	20	30	60
$m = 2$	-18.84	-16.66	-19.36	-21.06	-23.12
$m = 4$	-19.75	-17.40	-26.46	-31.18	-18.59
$m = 6$	-18.53	-16.64	-21.32	-26.22	-29.29

characterize even the shortest prediction horizon, with generally worse performance as p increases. Although not acceptable, the performance obtained by estimating the rate of change of CV1 is superior to that resulting from the direct estimation of CV1's value, reported in Table 4.1.

4.2.5 CV-informed GP-NARX Model of RoC

The RoC implementation presented in the previous subsection yielded a measurable improvement over all models that directly predict the value of CV1. For this reason, a CV-informed GP-NARX model, comparable to the one described in subsection 4.2.3, was designed for the one-step-ahead prediction of the rate of change of CV1. Several adjustments to the architecture introduced previously were made in order to achieve better simulation performance. First, extended input delays were applied; however, a finer

subsampling interval of 4 steps (240 seconds) was selected. The time window m_d was also increased to account for the additional delay introduced by the first-order filter during the preprocessing of the data and for possible inaccuracies in the delay estimation. The resulting inputs are the following:

$$\begin{aligned} u_{\mathrm{MV1}}(k) &= [MV1(k-23), MV1(k-19), ..., MV1(k-3)] \\ u_{\mathrm{MV2}}(k) &= [MV2(k-18), MV2(k-14), ..., MV2(k-2)] \\ u_{\mathrm{MV3}}(k) &= [MV3(k-18), MV3(k-14), ..., MV3(k-2)] \\ u_{\mathrm{DV1}}(k) &= [DV1(k-68), DV1(k-64), ..., DV1(k)] \\ u_{\mathrm{MVs}}(k) &= [u_{\mathrm{MV1}}(k), u_{\mathrm{MV2}}(k), u_{\mathrm{MV3}}(k), u_{\mathrm{DV1}}(k)] \end{aligned} \tag{4.9}$$

A second adjustment was applied during the selection of past CV1 values fed as regressors to the GP-NARX. In order to prevent the model from having direct access to the RoC of CV1 at the previous time step $\Delta y(k-1)$, a subsampling of 4 steps was applied in combination with an output delay of $n = 16$. This resulted in the following values being used as regressors:

$$u_{\mathrm{CV1}}(k) = [CV1(k-16), CV1(k-12), CV1(k-8), CV1(k-4), CV1(k)] \tag{4.10}$$

The same criterion was applied to all other CVs used to augment the knowledge of the GP-NARX model. The variables selected as regressors were the same as those listed in subsection 4.2.3.

A CV-informed GP-NARX model for the one-step-ahead prediction of the rate of change of CV1 was trained. The characteristics of the model were consistent with those detailed in previous subsections; a squared exponential kernel with ARD was selected, the basis function was set to zero, and a subset of data approximation of active set size $m_{as} = 1500$ was applied. The prediction results on the training and test datasets are shown in Figure 4.7. It can be clearly observed how the delay, first encountered in Figure 4.5, is still present in both plots. The choice of subsampling the past CV1 values did not reduce this behavior; instead, it increased the delay from one to three steps. This could explain the lower R^2 values, as the true data and the predictions are less similar.

The trained GP-NARX model was used within the NOE framework to perform simulation. Algorithm 1 describes this procedure, combined with the calculation of ${}^{p}\bar{R}^2$, using pseudocode. In particular, lines 2 to 10 detail the steps required for the simulation of a desired horizon of length p using the CV-informed GP-NOE model developed to predict the rate of change of CV1. At first, an empty vector of predictions $\hat{\mathbf{y}}$ and one containing the n past measurements of CV1 are initialized. The first value of $\hat{\mathbf{y}}$ is set to the current value of CV1 and then the algorithm starts iterating through all p steps of the horizon. As seen in subsection 4.2.3, two approaches can be used to perform simulation: updating-CVs and fixed-CVs. The former is implemented in line 6, while the latter is implemented

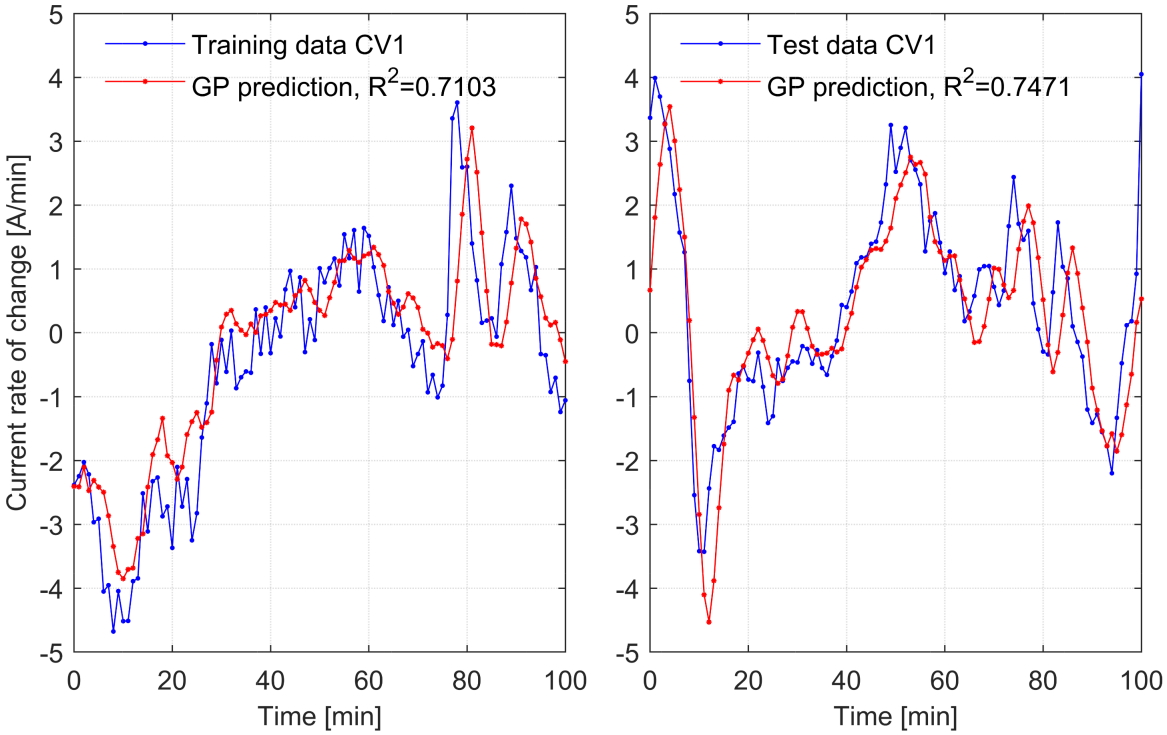

Figure 4.7: Plot of a detailed view of the one-step-ahead predictions (in red) of the RoC of CV1 on training (left) and test (right) datasets obtained using the CV-informed GP-NARX model with extended input delays.

in line 7. After the rate of change $\Delta\hat{y}$ is estimated, it is used to calculate the value of CV1 at the next step of the horizon. Lastly, the vector of past measurements is updated by removing the first (and oldest) entry and adding the new estimate $\hat{y}(j+1)$ and the loop is repeated. This procedure was applied to the first $l-p$ steps of the test dataset, in order to simulate multiple horizons and calculate their individual R^2 values. The average coefficient of determination for the horizon lengths $p = [5, 10, 20, 30, 60]$ minutes is reported in Table 4.6, both updating-CVs and fixed-CVs approaches have been evaluated. It can be observed that the CV-informed GP-NOE Model of RoC methodology yielded

Table 4.6: ${}^{p}\bar{R}^2$ values for the updating-CVs and fixed-CVs GP-NOE models of RoC. The results were obtained by applying Algorithm 1.

Model	Prediction horizon p [min]				
	5	10	20	30	60
Updating-CVs	-15.46	-7.37	-4.42	-3.57	-2.46
Fixed-CVs	-18.34	-10.62	-7.12	-6.69	-5.91

the best results when compared to all previous techniques presented. Long prediction horizons show better performance when compared to shorter ones. This could be due to a transitory behavior of the estimation, which initially is not able to match the true data but over time reaches a value closer to it. Another explanation could be that short horizons are characterized by little variance, which is overestimated by the GP-NOE

Algorithm 1 Computation of the average coefficient of determination over multiple horizons. Lines 2 to 10 implement the simulation of a desired horizon using a CV-informed GP-NOE model of RoC.

Initialize: Gaussian Processes model $f(\mathbf{x})$; test dataset D of length l containing process inputs u_{MVs}, process outputs u_{CVs}, and ground truth y_D; horizon length p; vector of $l-p$ coefficients of determination ${}^{p}\bar{\mathbf{R}}^2$

1: **for** $k \leftarrow 0$ to $(l-p)$ horizons **do**
2: **Initialize:** vector of p predictions $\hat{\mathbf{y}} = \mathbf{0}$; vector of n past measured values $\mathbf{x}_p$
3: $\mathbf{x}_p \leftarrow [y_D(k-n), ..., y_D(k-1)]$
4: $\hat{y}(0) \leftarrow y_D(k)$
5: **for** $j \leftarrow 0$ to p time steps **do**
6: $\Delta\hat{y} \leftarrow f(u_{\text{CVs}}(k+j),\ u_{\text{MVs}}(k+j),\ \mathbf{x}_p,\ \hat{y}(j))$ ▷ Approach: updating-CVs
7: $\Delta\hat{y} \leftarrow f(u_{\text{CVs}}(k),\ u_{\text{MVs}}(k+j),\ \mathbf{x}_p,\ \hat{y}(j))$ ▷ Approach: fixed-CVs
8: $\hat{y}(j+1) \leftarrow \hat{y}(j) + \Delta\hat{y}$
9: $\mathbf{x}_p \leftarrow [x_p(1), ..., x_p(n), \hat{y}(j+1)]$
10: **end for**
11: ${}^{p}\bar{R}^2(k) \leftarrow R^2$ of horizon k ▷ As defined in Equation 4.3
12: **end for**
13: **output:** ${}^{p}\bar{R}^2 \leftarrow \text{mean}({}^{p}\bar{\mathbf{R}}^2)$ ▷ As defined in Equation 4.4

model. As expected, the fixed-CVs approach performs worse than the updating-CVs one. However, it outperforms all previous model in horizons with $p \geq 10$. Overall, the inclusion of additional information about the plant greatly improved the estimation capabilities, compared to the simpler implementation of subsection 4.2.4.

In order to better understand the large differences in ${}^{p}\bar{R}^2$ values between short and long horizons, three 60 minute simulations are plotted in Figure 4.8. Line (a) shows the behavior that could explain the large negative values corresponding to shorter horizons. For the

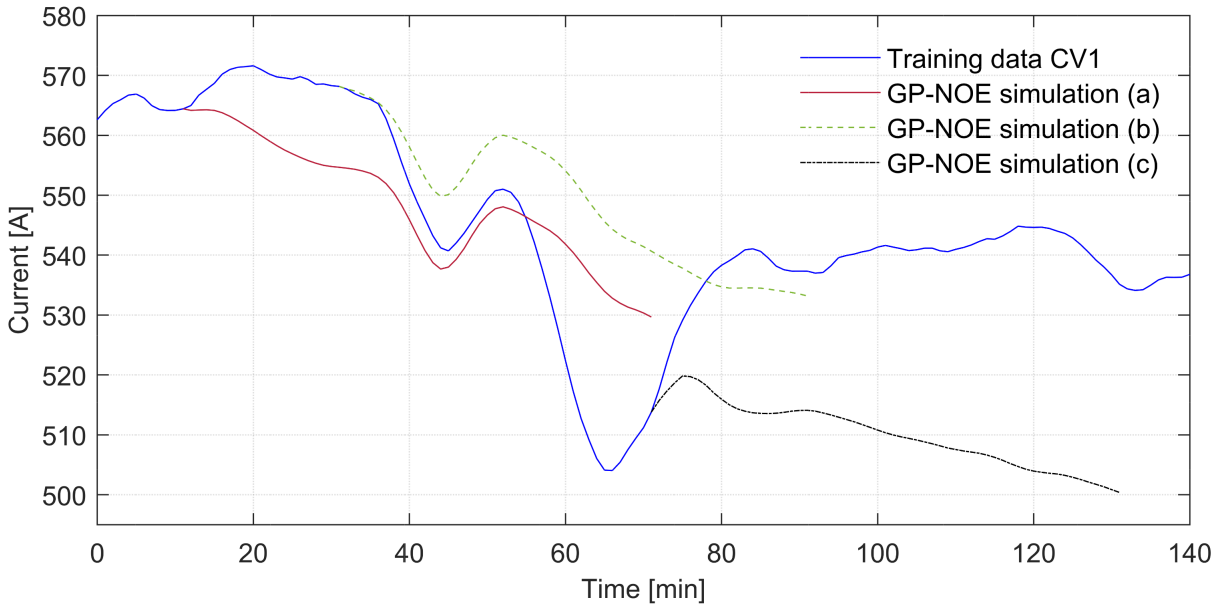

Figure 4.8: Comparison of three simulated 60 minute horizons obtained with the CV-informed GP-NOE model of RoC and the updating-Cvs approach.

first half of the simulation there is a significant error between estimation and true data; however, towards the tail end the two plots match. On the other hand, line (b) shows the opposite, with particularly good estimations at the beginning of the horizon, which accumulate error as time goes on. Lastly, line (c) is an example of the inability of the GP-NOE model to simulate the correct trajectory of the measured data. This plot has mostly illustrative value and cannot be used to quantitatively describe the performance of the trained model. Compared to Figure 4.3, the CV-informed approach for the estimation of the RoC appears to better capture the complexity of the system.

To gain more insight into the CV-informed GP-NOE model for the simulation of the RoC of CV1, two tests were carried out. The first one was aimed at investigating the importance of the past values of CV1 on the estimation accuracy. The four past regressors CV1(k-4), CV1(k-8), CV1(k-12), CV1(k-16) were randomized in pairs and used to retrain the GP-NARX model. This approach is based on the assumption that if a regressor contributes to the determination of the estimate randomizing it would result in a loss of performance. Vice versa, regressors that do not significantly impact the estimation process can be randomized without occurring in larger errors. Furthermore, it can be expected that performance degradation is directly proportional to the importance of the randomized regressor. The results of the simulations performed with three randomized GP-NOE models are reported in Table 4.7. Comparing them with those obtained by the

Table 4.7: ${}^{p}\bar{R}^2$ values for different updating-CVs GP-NOE models of RoC. The best model is compared with models trained on randomized regressors (RR) and a model trained on the unfiltered dataset.

Model	Prediction horizon p [min]				
	5	10	20	30	60
Best CV-informed GP-NOE of RoC	-15.46	-7.37	-4.42	-3.57	-2.46
RR: CV1(k-4), CV1(k-8)	-70.04	-46.52	-47.67	-48.43	-34.38
RR: CV1(k-8), CV1(k-12)	-18.33	-9.42	-6.15	-5.35	-3.18
RR: CV1(k-12), CV1(k-16)	-22.45	-14.13	-12.14	-12.51	-12.42
Unfiltered dataset	-78.41	-115.01	-92.90	-63.01	-36.67

best (non randomized) model it can be observed how CV(k-4) contributes the most to the estimation. Randomizing it and CV(k-8) causes a significant deterioration of ${}^{p}\bar{R}^2$ values across all horizons. CV(k-8) and CV(k-12) contribute the least, showing only a slight loss of performance. Lastly CV(k-16) appears to play a more important role, but not as significant as CV(k-4). These results confirm the hypothesis that the trained GP model bases its estimations heavily on recent values of CV1. The second test was performed to evaluate the impact that data preprocessing could have on the learning process. The GP-NARX model was retrained on the original unfiltered dataset. It was then implemented within the NOE scheme and used for simulation. The results are also reported in Table 4.7. A significant loss of performance can be observed. This is probably due to the large amount

of noise, amplified by the use of the rate of change as the prediction goal. In this condition the GP struggles to distinguish real information from noise, thus achieving unsatisfactory estimation capabilities.

To conclude the evaluation of the CV-informed GP-NOE model, its performance was compared with that of the MIMO first-order transfer function with time delay designed by Alperia Green Future and described in section 2.4. The ${}^{p}\bar{R}^2$ value of both implementations was calculated for the horizons with $p = [5, 10, 20, 30, 60]$ minutes, simulated throughout the test dataset. The results are reported in Table 4.8. It can be observed that the

Table 4.8: ${}^{p}\bar{R}^2$ values for the best updating-CVs GP-NOE models of RoC and for the MIMO transfer function with time delays developed by Alperia Green Future.

Model	Prediction horizon p [min]				
	5	10	20	30	60
Best CV-informed GP-NOE of RoC	-15.46	-7.37	-4.42	-3.57	-2.46
MIMO transfer function	-5.82	-4.02	-3.23	-3.17	-4.10

performance of the two methodologies is similar. The MIMO transfer function achieves better results in shorter horizons; however, the GP-NOE model outperforms it slightly in the longest horizon $p = 60$ minutes. This is the length of the prediction horizon used in the linear MPC implementation designed by Alperia Green Future, thus the most significant one. In lieu of the tests and comparisons performed, the quality of the best CV-informed GP-NOE model obtained can be considered sufficient. Although the ${}^{p}\bar{R}^2$ values obtained via simulation are negative, its capabilities should be adequate for a LbNMPC implementation.

5 Conclusion

The application of Gaussian Processes regression for the estimation of key process variables of a cement plant was investigated successfully. Real historical data was used to train a variety of GP models with the goal of finding the best architecture capable of simulating the behavior of a variable over a desired prediction horizon. The GP-NOE technique, introduced in [15], was extended through an iterative investigation approach. The methodology devised combines information provided by manipulated, disturbance, and controlled variables to produce multi-step ahead predictions of the selected output with sufficient accuracy. The results obtained were validated against those achieved with the current identified model of the plant. Although short prediction horizons were simulated with less accuracy, the developed technique was able to outperform the previous implementation over longer ones. This result is particularly significant considering that the work was carried out within the learning dynamics framework of learning-based nonlinear model predictive control described in [10]. This application combines machine learning techniques, used for system identification, with NMPC. Therefore, accurate simulation of long horizons is fundamental to achieve good control performance.

The results obtained are promising; however, they present important limitations. The use of real data collected in situ compounded the difficulty in identifying a complex system such as the clinker production phase of a cement factory. The time-varying, nonlinear, large time-delay nature of the plant, combined with noisy measurements, resulted in a trained black-box model capable of less than optimal predictions. In order to improve the control performance of a future LbNMPC implementation, it is paramount to reduce the estimation error of the GP model; therefore, more work is needed. The study presents a second limitation: only one of the many process variables was investigated. The design of a complete LbNMPC requires a model for each controlled variable used in the cost function of the NMPC. This procedure should require minimal tuning once the GP model architecture is defined. The last issue that was not considered was the timing requirements of the simulation step. Since the control action is updated every 60 seconds, the controller must be able to simulate the horizon and solve the optimal control problem within that time frame. In a complete implementation, where the behavior of multiple controlled variables must be predicted, care must be taken to satisfy this generous but strict requirement.

The GP model presented in this work was designed to be easily incorporated into the LbMATMPC architecture presented in [20]. In the future, a GP model could be trained for each of the remaining controlled variables to obtain a complete simulation of the clinker production plant. Further refinement of the architecture could be required in order to carry out this process. The black-box models could then be imported into the LbMATMPC

toolbox, where a nonlinear model predictive controller could be designed. Ultimately, the new architecture could be thoroughly tested and compared to the current control implementation. The results obtained in this work show promising potential that could possibly lead to improved plant performance if an LbNMPC is implemented.

References

[1] Susana Arad, Victor Arad, and Bogdan Bobora. *Advanced control schemes for cement fabrication processes*. Intech, 2008.

[2] Abhijit S. Badwe et al. "Detection of model-plant mismatch in MPC applications". In: *Journal of Process Control* 19.8 (2009), pp. 1305–1313.

[3] Eugene Boe et al. "Predictive control and optimization applications in a modern cement plant". In: *Conference Record Cement Industry Technical Conference, 2005.* IEEE. 2005, pp. 1–10.

[4] Anjan Kumar Chatterjee. *Cement production technology: Principles and practice*. CRC Press, 2018, pp. 1–63, 141–173.

[5] Anjan Kumar Chatterjee. "Chemico-Mineralogical characteristics of raw materials". In: *Advances in Cement Technology*. Elsevier, 1983, pp. 39–68.

[6] Anjan Kumar Chatterjee. *Intelligent and Sustainable Cement Production: Transforming to Industry 4.0 Standards*. CRC Press, 2021.

[7] Yutao Chen et al. "MATMPC-a MATLAB based toolbox for real-time nonlinear model predictive control". In: *2019 18th European Control Conference (ECC)*. IEEE. 2019, pp. 3365–3370.

[8] Christopher Csernyei and Anthony G. Straatman. "Numerical modeling of a rotary cement kiln with improvements to shell cooling". In: *International Journal of Heat and Mass Transfer* 102 (2016), pp. 610–621.

[9] Alessandro Di Bucchianico. "Coefficient of determination (R 2)". In: *Encyclopedia of statistics in quality and reliability* 1 (2008).

[10] Lukas Hewing et al. "Learning-based model predictive control: Toward safe learning in control". In: *Annual Review of Control, Robotics, and Autonomous Systems* 3 (2020), pp. 269–296.

[11] Peter Hewlett and Martin Liska. *Lea's chemistry of cement and concrete*. Butterworth-Heinemann, 2019.

[12] Bodil Hökfors. "Phase chemistry in process models for cement clinker and lime production". PhD thesis. Umeå Universitet, 2014.

[13] IEA. *Cement*. Tech. rep. IEA, Paris, 2022. URL: `https://www.iea.org/reports/cement`.

[14] Juš Kocijan. *Modelling and control of dynamic systems using Gaussian process models*. Springer, 2016.

[15] Juš Kocijan et al. "Dynamic systems identification with Gaussian processes". In: *Mathematical and Computer Modelling of Dynamical Systems* 11.4 (2005), pp. 411–424.

[16] Björn Lautenschlager. "Data-Driven Learning and Model Predictive Control for Heating Systems". PhD thesis. HafenCity Universität Hamburg, 2019.

[17] Zongjin Li et al. *Advanced concrete technology*. John Wiley & Sons, 2022, pp. 1–44.

[18] Derek Machalek and Kody M Powell. "Model predictive control of a rotary kiln for fast electric demand response". In: *Minerals Engineering* 144 (2019), p. 106021.

[19] Matlab MathWorks et al. *Statistics and machine learning Toolbox user's guide*. 2022.

[20] Enrico Picotti et al. "LbMATMPC: an open-source toolbox for Gaussian Process modeling within Learning-based Nonlinear Model Predictive Control". In: *2022 European Control Conference (ECC)*. IEEE. 2022, pp. 736–742.

[21] Guru Prasath et al. "Application of soft constrained MPC to a cement mill circuit". In: *IFAC Proceedings Volumes* 43.5 (2010), pp. 302–307.

[22] Carl Edward Rasmussen. "Gaussian processes in machine learning". In: *Summer school on machine learning*. Springer. 2003, pp. 63–71.

[23] Gerasimos Rigatos et al. "Nonlinear H-infinity control for optimizing cement production". In: *2018 UKACC 12th international conference on control (CONTROL)*. IEEE. 2018, pp. 248–253.

[24] Rajesh Sahasrabudhe et al. "Control and optimization in cement plants". In: *IEEE Control Systems Magazine* 26.6 (2006), pp. 56–63.

[25] Martin Schneider et al. "Sustainable cement production—present and future". In: *Cement and concrete research* 41.7 (2011), pp. 642–650.

[26] Harsh A Shukla et al. "Convergence certificate for stochastic derivative-free trust-region methods based on Gaussian processes". In: *arXiv preprint arXiv:2010.01120* (2020).

[27] Konrad S. Stadler, Jan Poland, and Eduardo Gallestey. "Model predictive control of a rotary cement kiln". In: *Control Engineering Practice* 19.1 (2011), pp. 1–9.

[28] Konrad S. Stadler, Burkhard Wolf, and Eduardo Gallestey. "Precalciner control in the cement production process using MPC". In: *IFAC Proceedings Volumes* 40.11 (2007), pp. 201–206.

[29] U.S. Geological Survey. *Mineral Commodity Summaries 2022*. Tech. rep. US Geological Survey: Reston, VA, 2022. DOI: https://doi.org/10.3133/mcs2022.

[30] Ravi Teja, P. Sridhar, and M. Guruprasath. "Control and optimization of a triple string rotary cement kiln using model predictive control". In: *IFAC-PapersOnLine* 49.1 (2016), pp. 748–753.

[31] Dimitris Tsamatsoulis. "Optimizing the Control System of Clinker Cooling: Process Modeling and Controller Tuning". In: *ChemEngineering* 5.3 (2021).

[32] Christopher K. I. Williams and Carl Edward Rasmussen. *Gaussian processes for machine learning*. Vol. 2. 3. MIT press Cambridge, MA, 2006.

[33] A Wurzinger et al. "Data driven modeling and nonlinear model predictive control design for a rotary cement kiln". In: *IFAC-PapersOnLine* 52.16 (2019), pp. 759–764.

[34] Silvia Maria Zanoli, Francesco Cocchioni, and Crescenzo Pepe. "Energy efficiency technologies in cement and steel industry". In: *IOP Conference Series: Earth and Environmental Science*. Vol. 121. 5. IOP Publishing. 2018, p. 052083.

[35] Silvia Maria Zanoli, Crescenzo Pepe, and Matteo Rocchi. “Improving performances of a cement rotary kiln: a model predictive control solution”. In: *Journal of Automation and Control Engineering Vol* 4.4 (2016).

[36] Silvia Maria Zanoli et al. “Optimization of the Clinker Production Phase in a Cement Plant”. In: *Portuguese Conference on Automatic Control*. Springer. 2020, pp. 263–273.