PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

# MODEL PREDICTIVE CONTROL BASED ON MACHINE LEARNING TECHNIQUES FOR PASTE TAILING PRODUCTION

**PABLO DIAZ TITELMAN**

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Advisor:
ALDO CIPRIANO ZAMORANO

Santiago de Chile, December 2018

PONTIFICIA UNIVERSIDAD CATÓLICA DE CHILE
ESCUELA DE INGENIERÍA

# MODEL PREDICTIVE CONTROL BASED ON MACHINE LEARNING TECHNIQUES FOR PASTE TAILING PRODUCTION

**PABLO DIAZ TITELMAN**

Members of the Committee:
ALDO CIPRIANO ZAMORANO
FELIPE NUÑEZ RETAMAL
JUAN CARLOS SALAS MORALES
MIGUEL HERRERA MARCHANT
DANIEL HURTADO SEPÚLVEDA

Thesis submitted to the Office of Research and Graduate Studies
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering

Santiago de Chile, December 2018

*Gratefully to my family and friends*

## ACKNOWLEDGEMENTS

I would like to thank my family for the continuous support and care given to all my academic endeavors from primary school up to this post-graduate degree. In all the countless hours spent studying the different aspects of engineering I was always nurtured by them and therefore I will be eternally grateful. In that same regard, I extend these words to all my friends from wherever I have met them. They have been the balance throughout this entire career and have been crucial in my life.

Secondly, I would like to thank Professor Cipriano, my advisor, for the patience and support during the entire investigation process. I would also like to extend this gratitude to Professor Nuñez for his unconditional and constant support during this research and Mario Torres for his collaboration in this investigation.

I thank the National Comission of Scientific Investigation and Technology for the funding of this research through the project CONICYT/FONDEF *First contest on technological investigation in the mining industry* from the funding granted by FONDEF/CONICYT project IT16M10012 "Tailing Optimal Control and Supervision Through New Technologies". I would like to thank Yamana Gold Inc., company that, through Minera Florida, contributed experience and technical data related to the operation and control of paste tailing production systems.

## TABLE OF CONTENTS

## LIST OF FIGURES

## LIST OF TABLES

## ABSTRACT

Paste tailing production is a relatively new issue in the mining industry. Dealing with high solid concentration levels make thickener operation particularly difficult and challenging to control.

Model Predictive Control is one of the main techniques used in industrial processes. Traditionally, predictive strategies have been based on linear descriptions of the system under study. However, processes like paste production and thickener operation are highly non-linear and subject to strong disturbances from upstream operation.

Machine Learning algorithms have been used for the past decades to address said issues and to generate better descriptions of systems. Random Forests regression has seen significant commercial and experimental success in the past years. However, its use for time series prediction, forecasting and control is scarce.

The present investigation proposes a Random Forest Model Predictive Controller for the paste production process. The main objective is designing, implementing and validating a purely data-driven controller through simulation of a particular thickening process. The main output corresponds to a general purpose toolbox that connects the mentioned Machine Learning algorithm and predictive control.

The proposed strategy is compared to three other benchmark control techniques, one of which is also a predictive controller. The results show that the new controller has better performance regarding disturbance rejection and setpoint tracking. Overall results prove that the strategy chosen could be used for real operation.



**Keywords:** paste production, thickener control, process control, model predictive control, nonlinear systems, machine learning, random forests

# RESUMEN

La producción de relaves en pasta es un tema relativamente nuevo en la industria minera. Lidiar con los altos niveles de concentración de sólidos hace que la operación del espesador sea particularmente difícil y desafiante de controlar.

El Control Predictivo basado en Modelos es una de las principales técnicas utilizadas en procesos industriales. Tradicionalmente, las estrategias predictivas se han basado en modelos lineales del sistema. Sin embargo, procesos como la producción de pasta y la operación de espesadores son altamente no lineales y están sujetos a fuertes perturbaciones.

Los algoritmos de Aprendizaje de Máquinas se han utilizado durante las últimas décadas para abordar estos problemas y generar modelos de mayor fidelidad. La técnica de *Random Forests* ha tenido éxito comercial y experimental significativo en los últimos años. Sin embargo, su uso en series de tiempo para predicción, pronóstico y control es escaso.

La presente investigación propone un Controlador Predictivo basado en *Random Forests* para el proceso de producción de relaves en pasta. El objetivo principal es diseñar, implementar y validar esta estrategia a través de la simulación del proceso de espesamiento. El producto final es una herramienta de *software* de propósito general que conecta dicho algoritmo de Aprendizaje de Máquinas y el control predictivo.

La estrategia propuesta se compara con otras tres técnicas de control referenciales, una de las cuales es también predictiva. Los resultados muestran que el nuevo controlador tiene mejor rendimiento en el rechazo a perturbaciones y seguimiento de referencias. Los resultados generales muestran que la estrategia desarrollada podría ser utilizada con éxito para la operación real de un espesador.

**Palabras Claves:** relaves en pasta, control de espesadores, control de procesos, control predictivo, control predictivo no lineal, aprendizaje de máquinas, random forests

# 1. INTRODUCTION

## 1.1. Context

By the year 2017 the mining industry represented 10.1% of the chilean GDP, making it the fifth most important economic activity. According to the Chilean Copper Comission (COCHILCO) the copper mining industry has produced approximately 5500 million metric tons of material on average each year since 2003 (COCHILCO, 2014). This number rises by approximately 6% yearly; this is also the projected growth for the year 2018.

Another study conducted by the same entity estimates a potential growth of 13.9% for the total volume of production by the year 2028, amounting to 6324.5 million metric tons by the end of that period (COCHILCO, 2015). Even though all gross production numbers appear to be increasing, by 2015 the ore grade was 0.87%, which halves estimations made in 1993. In other words, while production is predicted to increase consistently, mineral extraction becomes more and more difficult. This in turn results in a proportional rise in waste disposal and an increasing amount of mineral tailing to be disposed of in concentration plants.

According to the Chilean Geology and Mining National Service, tailing or mineral slurries are defined as "grinded solid of small particle diameter which is discarded in mining operations" (Sernageomin, 2018). This waste material is unloaded in a final reservoir which varies mainly by the tailing's density and water content. By the year 2018, there are 740 tailing deposits - considering all their different classes - with an annual generation of approximately 700 million tailing tons produced in 2017. Figure 1.1 shows the projected tailing production in metric tons for the years 2014-2026. Potentially, by the year 2026 the amount of mining waste could amount to 1249.7 million tons per year.

Tailing can be disposed of in many ways. Of the 740 total deposits in the country, 600 of them are of a sub-category refered to as tailing dam. Of the active 101 total deposits, 44 correspond to this class (Sernageomin, 2018). One of the main issues concerning

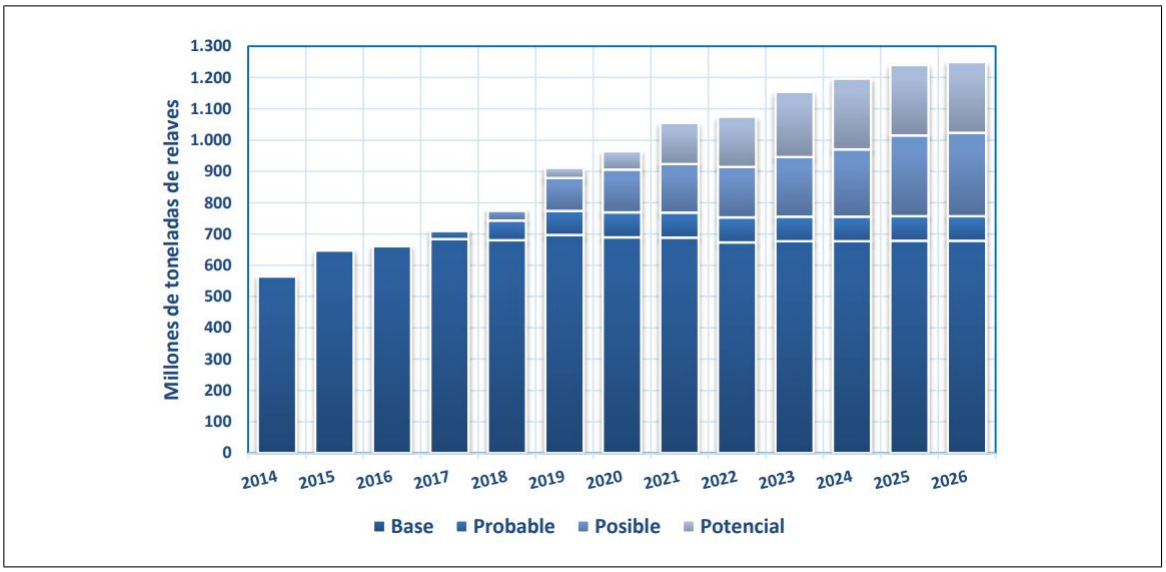


Figure 1.1. Projected tailing production for the years 2014-2026 in metric tons. The graph shows a steady tailing production increase with a potential maximum in 2026 of 1249.7 million tons (COCHILCO, 2015).

traditional tailing dam is the high water content of the slurry in the final reservoir, which is significantly high. This is extremely relevant for two reasons: firstly, water consumption in the mining industry is predicted to rise heavily from 16.7 $m^3/s$ this year to 24.6 $m^3/s$ by the year 2025 and the enforcement and increase in legal and environmental restrictions to guarantee the stability of the dam (COCHILCO, 2014).

Because of this, an important amount of economic and engineering effort has been invested in thickening technology, that is, passing the slurry through an equipment called thickener which favors sedimentation and recovery of the water content. After the thickening process has finished the slurry can be classified as thickened tailing - which is similar in solids content to the standard tailing -, paste or cake. Paste tailings are relatively new in the mining industry and current technologies are dealing with its production and control.

Paste is characterized by a solid content in the range of 65-75% weight concentration. Storing is then much safer with less needs of building sophisticated dams to contain the slurry (Cacciuttolo & Holgado, 2016). This is because of a rheological property - the yield stress - which prevents it from flowing naturally under the force of gravity, much like a

fluid with high viscosity. Even though a high solid content is of great importance to the stability and quality of the tailing in the final dam, it still has to be transported and pumped to that location. Therefore, paste solid content needs to be accurately managed to comply with both objectives.

Additionally, *yield stress* is regarded as an important rheological property in tailing management. It is defined as the minimum shear effort for the tailing suspension to start flowing (Stromberg, 2016). As such, it directly affects paste flow from its production to its deposition. Empirical studies show a nonlinear relationship between this property and underflow solid content; however, its online measurement is still an issue in thickener operation (Cacciuttolo & Holgado, 2016).

Thickener control is therefore an important research topic in the mining industry and academic community. Final solid concentration is usually the controlled variable of greatest interest, since it greatly determines the quality of the paste produced. However, as it is common in industrial process control, there are other variables to consider to ensure an adequate operation of the plant. Other restrictions and constraints arise from measuring, for example, the turbidity of the recovered water or the pressure at the bottom of the equipment.

The main issues regarding thickener control can be summarized as follows (Segovia, Concha, & Sbarbaro, 2011):

- Simulation of the process through first principle models is theoretically and computationally expensive. This limits the understanding of the process dynamics because actual experimental research on the process would require halting regular operations with high economic impacts.
- The process is believed to be highly nonlinear. The relationship between input and output variables is expected to change significantly between different regimes of operation.

- The time constants of the process are extremely large in comparison to other industrial applications. The residence time of the slurry in the thickener tank has a great impact on both the quality of the paste generated and the control of the whole process. Since measurement instrumentation, which is additionally fairly underdeveloped for this process, can only be applied to inputs and outputs, there is a great deal of the internal process which is unknown and has to be either estimated or bypassed.
- The relationship between controlled variables and manipulated variables is widely unkown. One of the main issues identified is the presence of delays in the effect on the controlled variables, which are always a challenging task in controller design.
- The process is heavily influenced by disturbances. Feed rate, feed solid content and feed particle size distribution - which depend mainly on the upstream grinding and flotation process - affect significantly the slurry being fed to the thickener. Therefore, disturbance rejection is an important aspect of the controller design to be considered.

This investigation focuses on thickener control, and therefore, is exposed to all these difficulties in a consistent manner. The first step for stating a research hypothesis which can be tested, measured and validated consists in understanding of current process control techniques and approaches.

## 1.2. Process Control Strategies

Industrial processes become more and more complex as history advances. This is a result of a globalized world which imposes higher standards of quality and competitiveness as time passes. Constraints on quality standards are becoming harder to achieve through manual and expert-only management. Because of this, automation solutions for industrial plants have seen a proliferation since almost three decades ago, with a corresponding increase in instrumentation which is the lowest level of industrial automation (Camacho &

Bordons, 2007). As processes become governed by regulatory loops or classical control feedback relationships, other variables become of interest. These take part in a higher layer of automation, which deals with objectives that concern a complete plant hierarchy. Thus, not only are stability and reference tracking sought, but also economical, environmental and other relatively new considerations play a key role.

A technique that has been implemented with significant success in process control is **Model Predictive Control (MPC)**. This can be attributed to the fact that MPC is, perhaps, the most general way of posing the process control problem in the time domain. Its formulation integrates optimal control, stochastic control, control of processes with dead time, multivariable control and future references when available. Another advantage of this technique is that because of the finite control horizon used, constraints and, in general nonlinear processes which are frequently found in industry, can be handled (Camacho & Bordons, 2007). A detailed historical perspective of MPC can be found in said book and references therein.

A survey conducted by Qin and Bagdwell in 2003 showed that the degree of satisfaction with MPC technology was the highest by 1995. This is explained by control problems dealing with process delays, nonlinearities and disturbances as well as interaction between different stages in industrial process (Qin & Bagdwell, 2003).

A traditional MPC strategy consists of mainly three aspects (Camacho & Bordons, 2007):

- Use of a *model* to predict the process output at future time instants. This is called *prediction horizon*.
- Calculation of a *control sequence* to drive the output to a desired reference. This is done by minimizing an objective function whose decision variables are the control inputs to be determined.

- A receding horizon strategy, which means that only the *first input of that sequence* is applied to the process. On the next time interval, when new process information arrives, this computation is repeated.

The receding horizon strategy is depicted in figure 1.2. At any given sampling instant, the output variable is forecast for a given horizon. Depending on this forecast, the controller decides on the optimal sequence that should be given to the system to drive the output to its reference. This sequence is computed through the minimizing of an objective function which typically penalizes *output error* and *control effort* in the horizon.

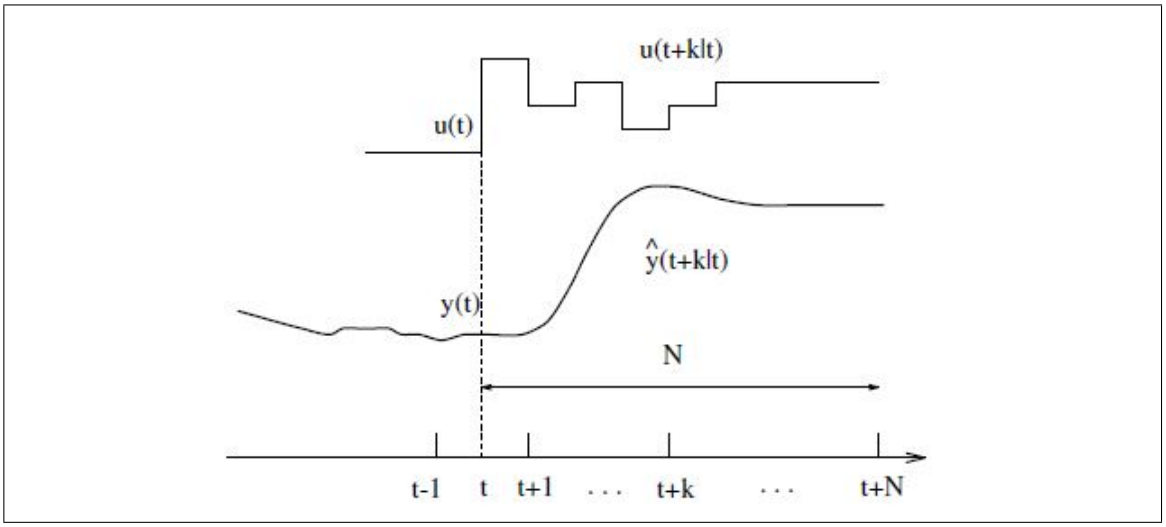


Figure 1.2. Receding horizon strategy. The predicted or forecasted values help to determine the optimal control sequence that would be needed to drive the output to its setpoint. (Camacho & Bordons, 2007).

The controller then obtains the "minimum effort" sequence to achieve - if possible - zero error in the prediction. Once the complete sequence has been computed, only the first value is given as an input to the plant being controlled. From a theory of information point of view, the next instant contains new information that can benefit predictions.

The description above illustrates why MPC has become an important tool in industrial control. Since an optimization problem is solved online, process constraints can be handled naturally. Also, by making the horizon large, the controller can overcome typical

problems regarding delays in input-output relationships and slow time constants. Furthermore, by tuning the weights in the objective function, control engineers can decide systematically how to prioritize different industrial objectives.

As figure 1.2 shows, an integral part of the MPC strategy is output prediction. This implies that the controller is embedded with a *model* - hence the name Model Predictive Control - that represents the system and that can be used to forecast the output beyond the current instant. At the root of the MPC strategy is a high-fidelity model representation of the system being controlled. The more accurate the prediction, the better the performance of the controller can be expected to be.

An important number of commercial MPC schemes are based on input-output error models of the plant (Camacho & Bordons, 2007). Dynamic Matrix Control (DMC) and Generalized Predictive Control (GPC), for example, are based on impulse and step responses of the system under study. By exciting the plant with input signals which generate a comprehensive amount of information of the system, output error models can be fit to those results. These models are usually linear and therefore simplify the optimization problem at hand, but cannot because of their nature, represent nonlinear dynamics. Other MPC schemes have tackled this issue by generating nonlinear descriptors of a system in a technique named **Nonlinear Model Predictive Control (NMPC)**.

Aside from the distinction between a linear and nonlinear predictor, a fundamental issue is that generating a model is fairly difficult. One alternative is to generate first principle models - based on physical, chemical or mathematical laws - that govern the dynamics of the system. Another approach consists on designing and applying specific and very limiting and limited signals as inputs to the process and measuring the process outputs (Hou, Gao, & Lewis, 2017).

This last branch of control and systems theory is known as system identification. A great deal of academic research has been input into this area (Ljung, 1999), but has indeed been unable to overcome the problem of input application. For industrial processes, this is

usually - if at all - done in an early stage of controller design. Running input-output tests during regular operation would imply halting and decreasing production performance for a given amount of time or even damaging equipment and the process as a whole for the sake of model validation (Hou, Gao, & Lewis, 2017).

With the development of information technology and the immense amount of operational data that can be retrieved offline and online, it is of natural interest to use historical records for system identification (Hou, Gao, & Lewis, 2017). In other words, building a model of the process from the collection of input-output information generated without disturbing normal operation.

This is called a *purely data-driven approach*, because it is based solely on input-output data of the process and no additional knowledge or scenarios to generate information. There are very important difficulties associated with this idea, which concern the fact that the plant is functioning usually in a stable operating point and under control and some form of feedback (Ljung, 1999). As such, the models developed under this strategy cannot rely on traditional system identification models.

To overcome this identification issues, Machine Learning developments have been increasingly incorporated in the systems and control academic community. Machine Learning is a branch of computer science and artificial intelligence that deals with the problems of classification, regression, prediction and clustering (Heaton, 2015). By using organized datasets and training algorithms the program is expected to recognize patterns in the data and generate a model representation of that structure. It is clear, even with this brief introduction, that Machine Learning seems an ideal alternative for the generation of a model using operational data of an industrial process. This is in fact one of the main focuses of this investigation.

All the comments made above illustrate why the MPC strategy is ideal for the paste production control. Since the process has slow dynamics and delays incorporated and MPC is designed to overcome these issues, it is only natural to consider this alternative for

thickener control. Furthermore, since paste production is believed to be highly nonlinear, an NMPC scheme should prove to be adequate as well. The challenge lies mainly in using only operational data for model generation and validation which corresponds with the state of the art terminology of **Data-Driven Predictive Control**.

### 1.3. Research Focus and Objectives

This research document is intended to prove if the integration of Model Predictive Control and Machine Learning algorithms using only operational data can control and stabilize the paste production process of a complex dynamical thickener simulator. The general objective of the research development is to develop a technique that integrates MPC and Machine Learning to control the paste tailing production.

Control of a real operating thickener imposes additional challenges on the development of the investigation, such as instrumentation, fault-management and connectivity issues. As such, the strategy is to be tested on a *simulated thickener* built and validated by the research team in a past investigation. The concept here is to generate *pseudo-real* data of the thickening process, that is, using the actual plant input data from a certain operational period to excite the simulator and generate simulated outputs of the process. This will then become the dataset used for model generation and validation and serves as an intermediate step to actual thickener control.

The specific objectives can be summarized as follows:

(i) Offer a revision of the state of the art literature of thickener control techniques.
(ii) Study different Machine Learning techniques suitable for system identification and select one for the development of a prediction model of a thickener.
(iii) Evaluate the predictive performance of the selected Machine Learning algorithm through simulation with real operational data.
(iv) Design and develop a control strategy that integrates MPC and Machine Learning in the form of Data-Driven Predictive Control.

(v) Integrate the developed control system with a complex simulator of the paste tailing thickening process.

(vi) Contrast qualitatively and quantitatively the performance of the designed controller to other control techniques.

(vii) Generate recommendations for the transfer of the designed strategy and controller to a real thickening operation.

This document is divided in seven chapters. In chapter 2 a deeper overview of tailing and paste production is given. Special emphasis is given on thickener and its simulation, which is the strategy chosen to test the controller designed. In chapter 3 an overview of control techniques is provided and a detailed description of MPC is given. State of the art applications to thickener control conclude this section. Chapter 4 deals with Machine Learning concepts and selection of the technique on which the predictive model is based upon. This chapter analyses the results of this identification process and compares it to a benchmark classical output-error model. The main focus of chapter 5 is combination of the MPC strategy with the data-driven predictive model. Specific ideas on how to overcome the issues associated with the online optimization using the model are given. The chapter concludes with the selection of an optimization method suited for this class of problems and the final implementation of the designed controller. Chapter 6 applies all the strategies described so far to the control of the thickening simulator. The constraints and the different performance assessment tests used to validate the controller are specified and contrasted with other controllers. This chapter also specifies the shortcomings of the method. The last chapter summarizes obtained results, examines objective completion and proposes further lines of work in paste production control, Machine Learning and the design of data-driven predictive controllers.

# 2. PASTE TAILING PRODUCTION

This chapter provides an overview of the entire paste production process, from the thickening stage to its final deposition.

In section 2.1 the three fundamental stages of the paste production process are reviewed. Section 2.1.1 explains the most important rheological property regarding tailing deposition in dams and their physical stability. Special emphasis is given on the *thickening stage* and current instrumentation and equipment used in paste production.

In section 2.2 the simulator used throughout this study is introduced and a characterization of it provided. Inputs and outputs are described as well as experiments on them in section 2.2.2. These results are used to explain how the control objectives can be shifted to the thickening stage and provide a comprehensive overview of the assumptions made to support this decision.

Finally, the full control problem to be investigated is stated in section 2.2.3. Control objectives are proposed as well as a complete definition of the dimensions of the system. This chapter is therefore an introduction into thickener production and control and is the foundation of all the research done thereafter.

## 2.1. Tailing Production

Tailing production begins once flotation is finished in concentrating plants. The floating cells increase the concentration of the copper up to levels useful for its smelting and refining stage through agents based on sulfur (Canfield, 2012). This is achieved through the adhesion of the mineral to bubbles that rise in the cells and later overflow. The overflow material (pulp) is collected in gutters and follows its course to the refining stages. Tailing corresponds to the waste material of said process, which contains diverse reactives and low mineral content dissolved in water (Wu, Wang, & Wang, 2015).

Tailing management is a long and multistage process. The final product is accumulated in large deposits called *tailing storage facilities* (TSF). Its final consistency and physical/-chemical properties depend on each stage of the process. Figure 2.1 illustrates the complete production process, from the waste resulting from flotation to its deposition in the final dam.

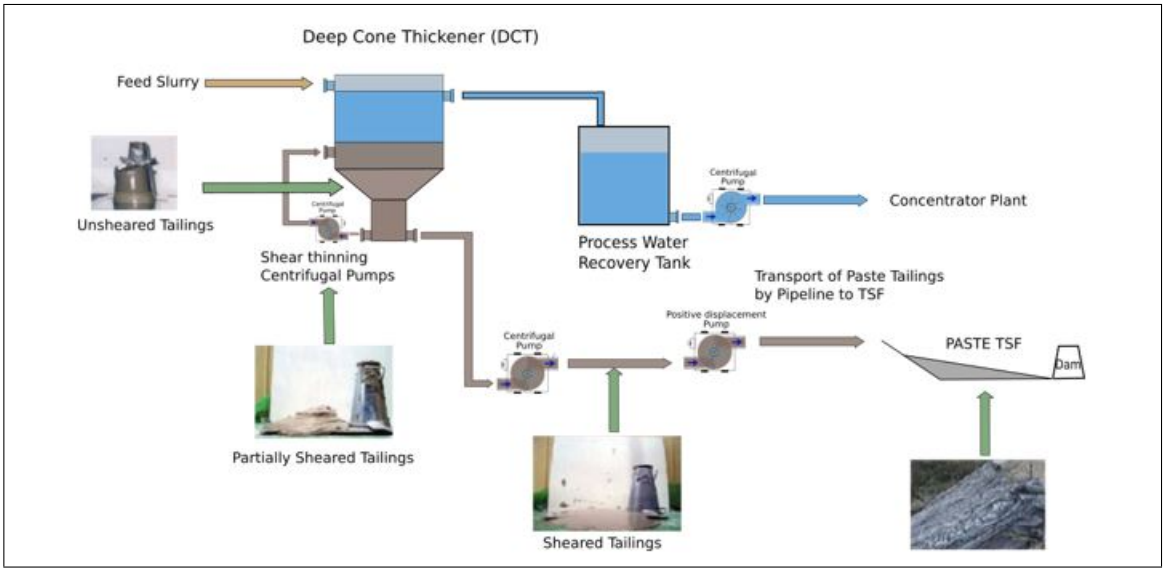


Figure 2.1. Complete process of tailing production (Cacciuttolo & Holgado, 2016)

Figure 2.1 shows all three stages concerning paste tailing production: thickening, transport and deposition in a TSF. Control efforts are focused on the initial stage, since most of the manipulated variables that handle the characteristics of tailing disposal are found in this section of the process.

Paste characterization is based on yield stress criteria. The final objective is ensuring paste quality in the slurry unloaded in the reservoir which implies a large hydric recovery during the thickening process. Thus, it is necessary to understand in some depth the physical meaning of this parameter, as well as the variety of methods and experiments that exist in the literature to quantify it.

### 2.1.1. Yield Stress

The mixture characterization represents a physical problem for fluid mechanics: tailing is not an elastic liquid nor a Newtonian fluid. Fluid properties cannot be reduced to its viscosity and a flow or shear effort versus speed of deformation curve is needed to describe it (Moller, Fall, Chikadi, Didi, & Bonn, 2009).

Paste tailing only flows over a specific shear stress threshold; under that threshold the fluid can be considered quasi-static.

The model commonly used to characterize these fluids is that of Herschel-Bulkley (Moller et al., 2009):

$$\tau = \tau_y + \alpha\dot{\gamma}^n \tag{2.1}$$

where $\tau$ is the shear stress, $\tau_y$ is yield stress, $\dot{\gamma}$ is deformation or velocity gradient and $\alpha$ and $n$ are adjustable parameters.

The objective is to ensure a specific range of yield stress of the tailing unloaded in the dam. *Slurry can be classified as paste when this variable is within in the range of 100 to 800 Pascals* (Cacciuttolo & Holgado, 2016).

Yield stress determination is widely discussed in the literature. Online measurement of this property is difficult, so estimation methods are required. In short, this problem is still an issue in the academic community of rheology.

### 2.1.2. Thickening

Development of the mining industry has generated and accumulated a vast experience in operation and thickener manufacture. The distribution of the particle size, the mineralogy and rheology are fundamental variables that ultimately determine the thickener suitable for a certain type of mining process (Cacciuttolo & Holgado, 2016).

A thickener is a tank of a usually cylindrical or conical shape with blades inside - called a harrow or rake - that concentrate the mixture towards the center. This mixture is the sedimentation of the tail generated in the flotation process. During its residence in the tank different layers of sedimentation are formed. Figure 2.2 shows a diagram of the inside of a thickener.

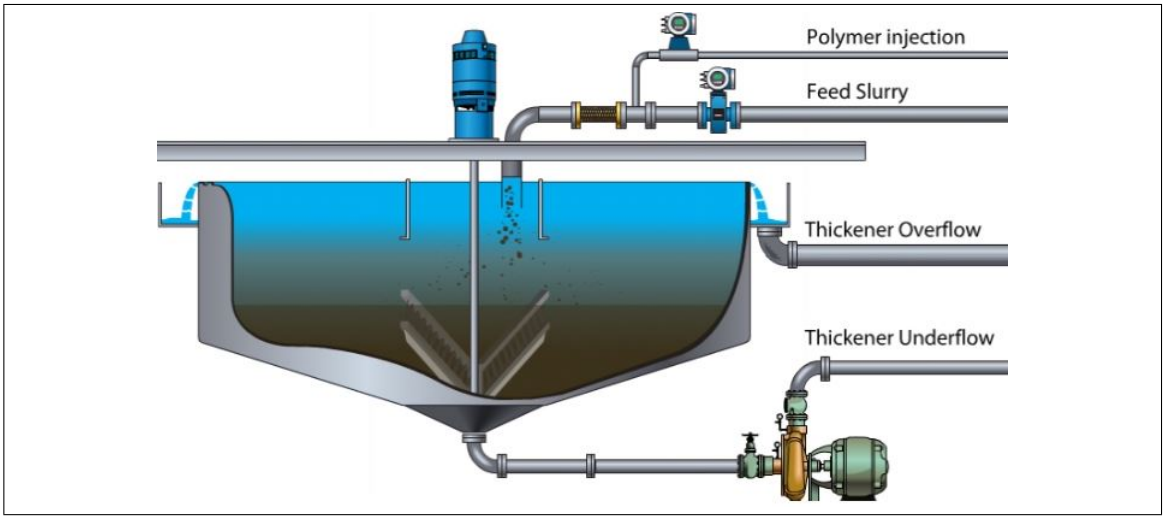


Figure 2.2. General thickener structure and function (Zeroday, 2015)

Almost all thickeners include the components depicted in figure 2.2: slurry feed pipes, a flocculant injector and a tailing scape valve. The rakes that generate the necessary torque for the process to be effective are also shown.

In conventional thickeners the feedwell is at the top of the equipment. Once the tailing is fed, it is diluted by the upward flowing liquid and precipitates at constant velocity to form a sediment at the bottom. In certain thickeners, flocculant is added in this input stream to aid the sedimentation process (Langlois, 2018).

Thickeners generate two distinct mixture zones in their longitudinal direction: an area of clear water (clear zone) and another with much more solid content (compaction zone) (Burger, Karlsen, & Towers, 2005). A detailed review of the different kinds of thickeners and their characteristics can be found in (Cacciuttolo & Holgado, 2016) and in references

therein. Figure 2.3 explains the relationship between different types of thickeners, tailing yield stress and the underflow solid concentration.

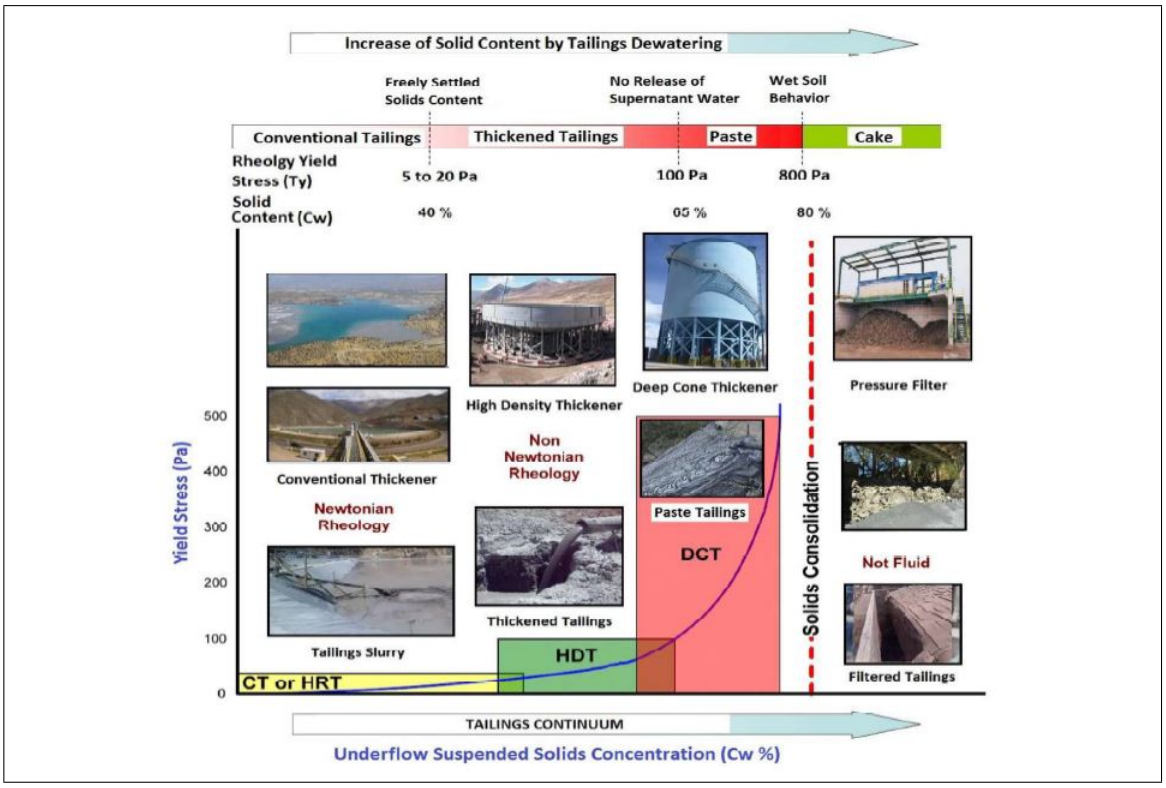


Figure 2.3. Relationship between yield stress, underflow solid concentration and thickeners (Cacciuttolo & Holgado, 2016). The figure shows an explicit non-linear relationship between both variables.

As expected, when removing and recovering water in the thickener, solid concentration ($C_u$ onward) increases and a non-linear relationship with yield stress is obtained. In addition, the graph shows the main type of thickener used for paste production: the Deep Cone Thickener (DCT).

Figure 2.3 suggests how to establish a control strategy based on the solid content in the thickener discharge. Since direct yield stress measurement is difficult, the proposed hypothesis is to *control solids concentration in the discharge* and therefore indirectly control the yield stress in the final dam. The actual ranges of underflow solid concentration

are difficult to establish and vary between different operations. Therefore, additional information needs to be collected regarding paste characterization through $C_u$.

#### 2.1.2.1. Deep Cone Thickener

This type of thickeners are characterized by their large diameter to depth ratio in order to agglutinate particles as much as possible. Likewise, they deal with greatest conical inclination and usually exert greatest torque (Cacciuttolo & Holgado, 2016). DCTs are ideal for paste production because they are designed to handle yield stress up to 300 Pa with an increased water recovery. There are some examples of thickeners of this type in Chile, but their use is limited due to the fact that paste technology is recent and less used than conventional tailing (Espinace, Villavicencio, & Fourie, 2016).

### 2.1.3. Tailing Transport

The thickener outlet is located at the bottom of the compaction zone and therefore contains a suspension with high solids content. Physio-chemical properties are believed change when transported to the final dam (Fitton, 2016). This implies that the paste transport is of significance to guarantee the final condition of the tailing.

Discharge of the slurry from the thickener is generally produced by an initial pumping system, as shown in figure 2.1. However, authors recognize the need of additional equipment to achieve this objective according to the yield stress ate the discharge(Cacciuttolo & Holgado, 2016), (Quelopana, 2016). To estimate this, a manual slump test is applied to a tailing sample. According to the basal diameter of the cone produced, a gross estimation of the tailing characteristics can be obtained. Additional shear thining pumps can be used to guarantee paste quality in the TSF.

The last part of the process is pumping paste into the dam. Siphons or spigots are hoses that are located in the perimeter of the tailing dam. These discharge points play also a crucial role in the tailing slope in final dam (Fitton, 2016).

Effects of pipe and transport on paste are currently under study in the scientific community (Fitton, 2016). However, from a process control point of view, there are not many possibilities to interact with the slurry during this stage. Slurry transport will be ignored under the hypothesis that a well-controlled thickening process guarantees adequate properties downstream. Future research could include the effect of piping transportation and pumping processes in the final product.

### 2.1.4. Tailing Reservoir and Dam

The pumped tailing is finally accumulated in a TSF. This part of the process takes advantage of topographical conditions and generates a dam of considerable size. TSF include a wall that prevents runoff into surrounding areas.

In Chile, the country with the largest projection of tailing deposits thickened by 2025 (Espinace et al., 2016), there is a significant inconsistency between design and operation of the dams. The effect of precipitations and topographic change created by the deposit are underestimated, as well as the drainage of these waters to the soil due to plant failures and other factors (Espinace et al., 2016). This situation shows the need to properly adequate tailing management techniques.

The slope and its geometric shape, as well as slurry properties, play a crucial role in the physical stability of the wall. Slope estimation is of interest in the scientific community because of real time variations due to tailing accumulation. A fairly common approach is to take advantage of certain rheologic, mineral and physical characteristics of the paste or pulp for slope characterization (Quelopana, 2016). This variable affects not only the stability of the retaining wall, but also alters the yield stress of the paste slurry over time.

## 2.2. Thickening Process Simulation

As discussed so far, the paste tailing production process is divided into 3 major stages: thickening, transport and deposition.

Control of the thickening process is generally associated to thickener control. The current technology developed and implemented in Chile is strongly focused on thickeners for conventional tailing, that is, whose yield stress is below 100 Pa.

For this research, the thickening control system will be designed and simulated mainly in Matlab and Simulink lenguage, version 2017a.

A recently completed thesis insert in the same Fondef project generated a Simulink model of a thickener based on its physical and mechanical properties (Langlois, 2018). This model has been calibrated to reproduce production results in the months of August to September 2017 of the operation of Yamana's current DCT. As such, it provides an important opportunity to design, test and validate control alternatives regarding thickener control. Control of the actual thickener is extended beyond the scope of this research, but is the natural next step.

The objective of the next section is to describe the thickener simulation block, specially regarding its input and output relationships. An initial overview of the process variables is provided, as well as a characterization of the simulator as a whole (Langlois, 2018). Finally, in section 2.2.2, the description of the input and output relationships is given based both on available literature on the thickening process and results of the mentioned document.

### 2.2.1. Description and Characterization of Thickening Simulator

Figure 2.4 illustrates conceptually the inside of a thickener and the different regions involved in the process.

This diagram aids in understanding what defines the compression region: a zone in the mixture in which the solids concentration is greater than a critical concentration $\phi_c$. Three distinct regions can also be seen. The geometry of the modeled thickener corresponds to that of the diagram: a constant diameter section from the feedwell down to a cone where the diameter starts to reduce.

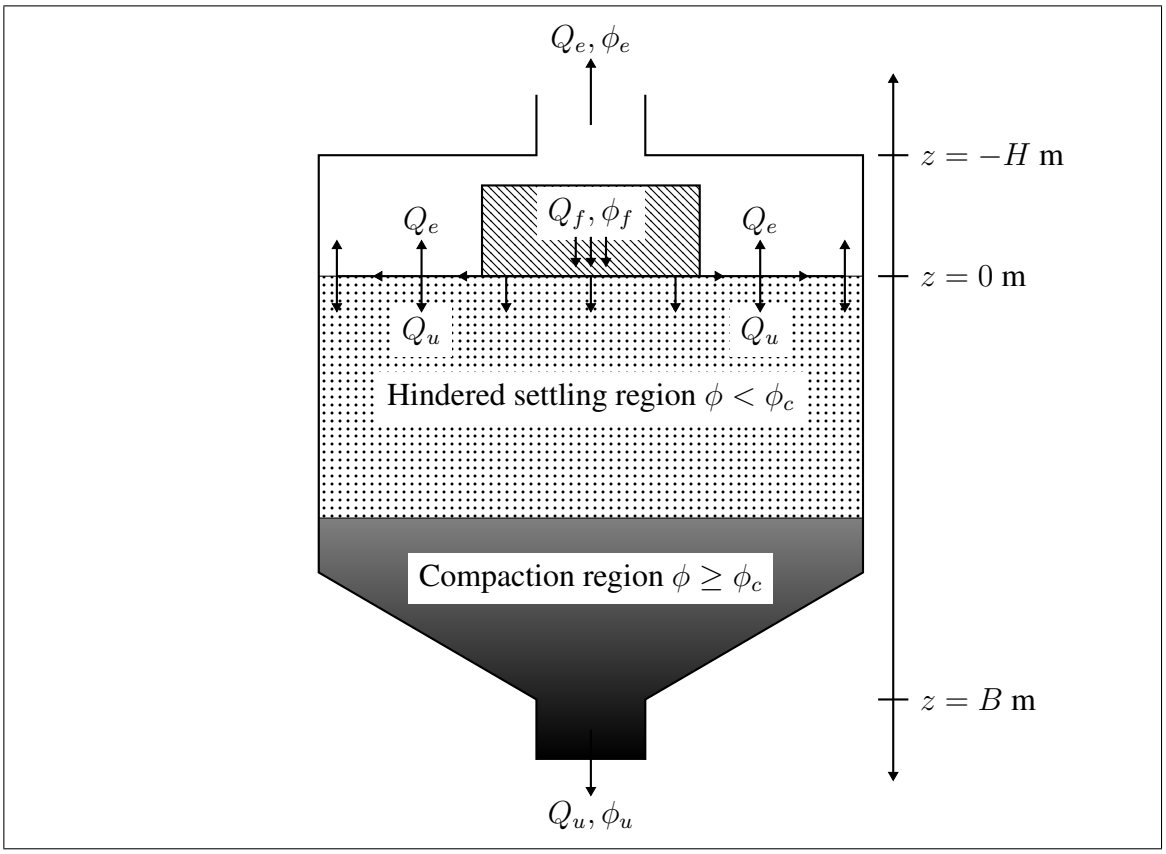


Figure 2.4. Conical thickener diagram (Langlois, 2018)

It is also shown that the distance $z$ is measured as depth into the thickener rather than from the bottom to the surface. At distance $z = 0$ and in the center of the thickener the feedwell mechanism is displayed. This is where the flocculant solution is added to the thickener. By being at its center and in direct contact with the arriving slurry, a better mixture and flocculation process is ensured.

A brief summary of the governing physical and chemical principles in thickening can be found in appendix A. For a full description of the deduction, application and implementation of these principles, the reader is referred to the work of Langlois and references therein (Langlois, 2018).

#### 2.2.1.1. Input Variables

Paste production is dependent on the amount and rate of solid that enters the thickener. However, other variables have important influences on the quality of the paste. These are described below.

**Underflow rate $Q_u(t)$:** The relationship between the feed rate and the underflow rate determines to a large extent the quality of the pulp generated. This is because the residence time in the thickener is fundamental in the sedimentation process and the recovery of water (thickener effluent) (Cacciuttolo & Holgado, 2016). Measured with flowmeters in the thickener outlet.

**Flocculant dosage $F(t)$:** This agent affects *Particle Size Distribution* (*PSD*) as well as the water content of the tailing (Cacciuttolo & Holgado, 2016).

A *feedforward* loop regulates flocculant addition in relationship to solid throughput in the thickener. Given a flocculant dosage setpoint $w_F$, internal regulatory loops handle the concentration and rate of the flocculant solution added to the thickener. This is done by diluting an initial flocculant solution stream $q_F(t)$ of concentration $G_F$ to one tenth. The following relationship illustrates this concept:

$$F(t) = \frac{G_F}{10}\frac{q_F(t)}{Q_s(t)} = \frac{G_F}{10}\frac{q_F(t)}{Q_f(t)\rho(t)C_f(t)} \tag{2.2}$$

**Feed rate $Q_f(t)$:** This variable can be manipulated in theory, but this means reducing plant throughput (Stromberg, 2016). It is measured with flowmeters in the thickener feed pipes.

**Feed solids concentration $C_f(t)$ (F.S.C.):** In general, to keep the thickener in controlled operating ranges, regulatory loops are used for the thickener feedwell in order to soften variations in the concentration of solids at this level. Measured indirectly through feed density $\rho_f(t)$ and is finally expressed as a ratio of the solid mass to the total mass in the feedflow:

$$C_f(t) = \left(\frac{\rho_s}{\rho_f(t)} - \rho_s\right)\frac{100}{\rho_s - 1} \tag{2.3}$$

where $\rho_s$ corresponds to the nominal solid density of the mineral using water as the solvent.

**Feed particle size $\mu(t)$ (F.P.S.):** Corresponds to the fraction in weight of particles with diameter $d$ less than 20 $\mu$m, notated as $P_{20}$. Online measurements of this variable are unavailable. Some models relate particle diameter distribution with the slurry's properties during its residence in the thickener. These in turn impact the paste's yield stress and the underflow solid content (Betancourt, Burger, Diehl, & Faras, 2014).

#### 2.2.1.2. Output Variables

Even though there are many outputs that can be measured in thickener control and paste production, the list below states which will be used in this research.

**Rake torque $T(t)$:** Torque variations are desired to be soft and limited as much as possible. Thickeners have sensors associated with the tank that measure this variable directly. Hard constraints limit the range of this variable since increasing torque usually indicates overloading and can lead to operation halt (Ojeda, Bergh, & Torres, 2014). Usually measured as a percentage between it's minimum and maximum operating limits.

**Underflow solids concentration $C_u(t)$ (U.S.C.):** This variable is critical in determining the yield stress of the paste. However, controlling this output directly can cause big fluctuations or put overwhelming stress on the underflow rate (Stromberg, 2016). Similar to the feed solids concentration, it is expressed as the ratio between solid mass and total mass in the suspension expressed in the following formula:

$$C_u(t) = \left(\frac{\rho_s}{\rho_u(t)} - \rho_s\right)\frac{100}{\rho_s - 1} \tag{2.4}$$

for underflow density $\rho_f(t)$.

**Interface level $h(t)$ (I.L.):** The distance between the clear zone and the beginning of the compaction layer in the thickener is also controlled. It directly affects the flow of

recovered water, as well as the concentration of solids in the outlet of the tank. Sensors allow to find the depth inside the tank where the two phases collide. Measured in meters from the feedwell to the bottom of the tank.

### 2.2.2. Input and Output Relationships

Thickening, particularly paste production, is an active field of research and with many relationships still somewhat unknown.

A useful description of a process can be done in terms of a first order function:

$$g(t) = K_{DC}\left(1 - e^{-\frac{(t-L)}{\tau}}\right) \leftrightarrow G(s) = K_{DC}\frac{e^{-sL}}{s+\tau} \tag{2.5}$$

where $K_{DC}$ is the static gain, $\tau$ is the time constant and $L$ is the transfer function delay. Even though simple, this characterization provides powerful insights on the process under study. The main advantage of this description arises from the fact that by specifying the triplet $(K_{DC}, \tau, L)$ the system is fully described.

The research in (Langlois, 2018) - from which the simulator is produced and validated - provides some insight into how process variables relate to each other. By performing separate input step tests, a description of the system can be obtained. However, since the system is non linear on at least some input-output pairs, *no full characterization by input tests can be done on the system* because different effects can be seen in different operating points.

An experimental study of thickener operation was found in the literature (Aghajani Shahrivar et al., 2013). The effect of feedflow rate $Q_f$, feed solids concentration $C_f$ and feedflow rate $Q_f$ on underflow solids concentration $C_u$ and interface level $h$ were studied for constant underflow rate $Q_u$. Multiple non linear regression models were generated and the best were selected by means of Analysis of Variance (ANOVA) criteria.

Since different operating points can be described by different tuples, a better understanding of the process can be done by a *qualitative* description of the parameters in expression 2.5. Such a description is similar to how an expert on thickener control would describe the operation.

Table 2.1 lists the main conclusions drawn from both of the sources mentioned above. All static gains are mentioned with respect to *positive changes* in the input variables, and time constants and delays are referred to the residence time in the thickener.

Table 2.1. Description of relationship between process variables.

| **Variables** | **Torque** $T(t)$ | **U.S.C.** $C_u(t)$ | **I.L.** $h(t)$ |
|---|---|---|---|
| Underflow Rate $Q_u(t)$ | $K_{DC}$: Negative, small.<br>$\tau$: Medium-fast.<br>$L$: Negligible. | $K_{DC}$: Negative, small.<br>$\tau$: Medium-fast.<br>$L$: Negligible. | $K_{DC}$: Positive, big.<br>$\tau$: Medium-slow.<br>$L$: Small. |
| Flocculant Dosage $F(t)$ | $K_{DC}$: Almost zero.<br>$\tau$: Not detectable.<br>$L$: Not detectable. | $K_{DC}$: Almost zero.<br>$\tau$: Not detectable.<br>$L$: Not detectable. | $K_{DC}$: Positive, small.<br>$\tau$: Very slow.<br>$L$: Large. |
| Feed Rate $Q_f(t)$ | $K_{DC}$: Positive, big.<br>$\tau$: Slow.<br>$L$: Very large.<br>Integrating effect. | $K_{DC}$: Positive, big.<br>$\tau$: Slow.<br>$L$: Very large.<br>Integrating effect. | $K_{DC}$: Negative, big.<br>$\tau$: Slow.<br>$L$: Large.<br>Integrating effect. |
| F.S.C $C_f(t)$ | $K_{DC}$: Positive, big.<br>$\tau$: Slow.<br>$L$: Very large.<br>Integrating effect. | $K_{DC}$: Positive, big.<br>$\tau$: Slow.<br>$L$: Very large.<br>Integrating effect. | $K_{DC}$: Negative, big.<br>$\tau$: Slow.<br>$L$: Large.<br>Integrating effect. |

It is usual for industrial processes to be affected by variables whose effect is a ramp-like behaviour in the outputs. These are denominated *integrating processes* (Camacho & Bordons, 2007). Table 2.1 characterizes feed flow rate and feed solids concentration as such processes.

Flocculant influence on talings is to some extent unknown and currently under research by many authors (Burger, Karlseb, & Towers, 2005), (Betancourt et al., 2014). In fact,

most control strategies for paste thickening do not involve manipulation of the flocculant dosage setpoint (Tan, Bao, & Bickert, 2017).

The work in (Langlois, 2018) suggests that increments in flocculant dosage have only transient effects on underflow solids concentration. This is explained by the fact that flocculant addition does not affect the overall solid content of the thickener. Hence, after a small transient increase in $C_u$ the effect decays yielding no permanent consequence. The coefficient of the $F$ term deducted in the lab experimental study (Aghajani Shahrivar et al., 2013) supports this. Torque level reacts almost identically because of the model proposed (Langlois, 2018).

An increase in flocculant concentration should make sedimentation velocity higher, therefore lowering the solid inventory level (Betancourt et al., 2014). However, this characterization is highly dependent on the operating point of the thickener.

Besides their permanent effect, an important aspect of feed inputs is their rate of change. For example, feedflow rate varies rapidly and therefore the slow dynamics inherent to paste production naturally reject its transient effects. Feed solids concentration, on the other hands, changes at a much slower rate, since its variations come from inadequate plant operation or a different conditions in upstream processes.

Particle size distribution $\mu(t)$ usually lacks instrumentation for online measurement. In the work of (Langlois, 2018) its effect is seen directly on the yield stress of the paste. It is likely that its behaviour is similar to that of feed solids concentration $C_f$ as it distorts the size of flocs and increases the volume of solid relative to the slurry.

### 2.2.3. Control Problem Definition

The field of research in paste tailing production is vastly open. Strategies in many of the mining companies are those of manual-expert type: operators that have acquired considerable knowledge aid in the decision making process (Ojeda et al., 2014).

The main objectives in paste production are (Jewell & Fourie, 2015):

- Ensuring talings classification as paste, that is, maintaining the yield stress of the tailing in the 100 to 800 Pascals range.
- Keeping underflow concentration within a specific range, thereby keeping certain rheological properties of the paste, most importantly the yield stress, from the thickener to the TSF.
- Producing overflow of low solid concentration (close to 0%).
- Significant disturbance rejection, specially regarding solid throughput.

From an automation point of view, these criteria can be expressed as *stabilizing the plant*, *tracking a specified setpoint in output variables*, *process constraints satisfactions* and *disturbance rejection*. The controller designed can include costs in manipulated variable efforts, constraint violations and errors in setpoint tracking.

In the literature reviewed, various experiences in thickener control were found. However, they are directed to conventional thickeners and not DCTs. Thus, the system under study has an added degree of difficulty since it deals with much higher solids content and tighter yield stress constraints.

As stated before, this investigation is centered around *thickener control* under the hypothesis that an adequate, stable and if possible optimal control of this process will ensure paste quality in the TSF.

Figure 2.5 shows the full control loop to be designed. This is the *complete feedback loop considered in this research*. The setpoint vector $\vec{w}$ is supplied from an external source, usually calculated in regards to a steady-state objective. The measured plant outputs $\vec{y}_m$ are compared to their respective reference signals and an error $\vec{e}(t)$ is computed.

The plant considered is a 5-by-3 multiple input-multiple output (**MIMO**) system with only two manipulated variables (*MV*) and 3 disturbances as inputs to the system. Even

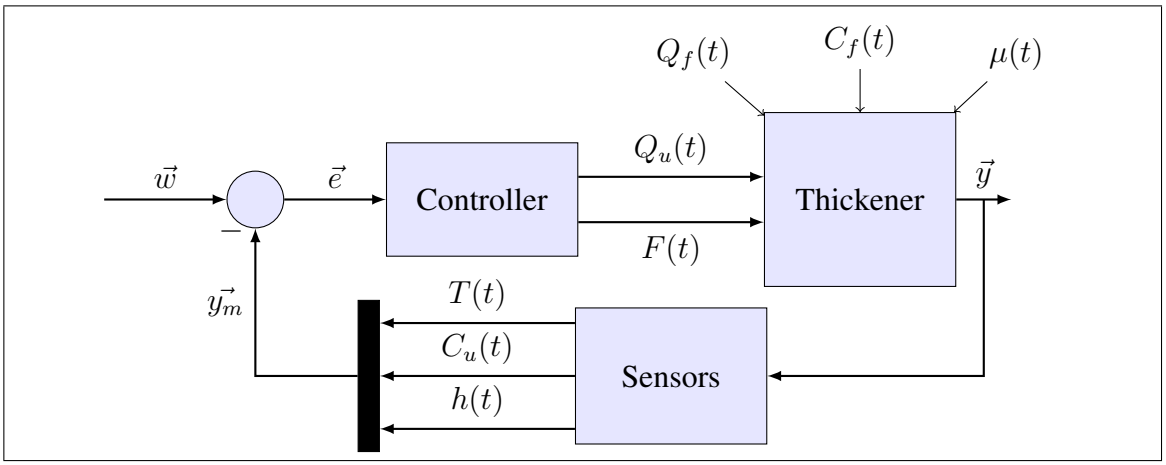


Figure 2.5. Block diagram for thickener operation control. The multivariable nature of the controller design problem is shown.

though many outputs $\vec{y}$ can be studied, for the purpose of this research and the controller design only *three* will be considered as controlled variables (*CV*).

Controlled variables are those through which the final quality of the paste is estimated and predicted. In fact, the control objectives listed above have to be translated to the controlled variables of figure 2.5:

- The yield stress range will be ensured by controlling underflow solids concentration $C_u(t)$. Additionally, constraining torque $T(t)$ to an operation window will be used as a secondary indicator of yield stress.
- Low overflow solids concentration will be achieved by control of the interface level. By keeping it at a reasonable distance from the feedwell, the turbidity of the effluent will be kept to its minimum.

The disturbances (*DV*) which affect the process can be divided in many categories: input or output disturbances and measured or unmeasured disturbances. The first group differentiates between variables which affect go *through* the process and variables which are only added to the outputs.

All the considered disturbances arise from upstream processes of grinding and flotation, as established in section 2.1.2. Feed particle size $\mu(t)$ is the *only unmeasured disturbance* considered in this study.

Table 2.2 summarizes the information regarding all variables considered in this investigation.

Table 2.2. Summary of variable information considered in the control task

| **Variable** | **Type** | **Unit** | **Identifier** | **Symbol** | **Range** |
|---|---|---|---|---|---|
| Torque | Controlled | % | $y_1(t)$ | $T(t)$ | $[0, 100]$ |
| Underflow Solid Concentration | Controlled | % | $y_2(t)$ | $C_p(t)$ | $[0, 100]$ |
| Interface Level | Controlled | % | $y_3(t)$ | $h(t)$ | $[h_{min}, h_{max}]$ |
| Underflow Rate | Manipulated | $\frac{\text{m}^3}{\text{hr}}$ | $u_1(t)$ | $Q_u(t)$ | $[Q_{u_{min}}, Q_{u_{max}}]$ |
| Flocculant Dosage | Manipulated | gpt | $u_2(t)$ | $F(t)$ | $[F_{min}, F_{max}]$ |
| Feedflow Rate | Measured Disturbance | $\frac{\text{m}^3}{\text{hr}}$ | $d_1(t)$ | $Q_f(t)$ | $[250, 450]$ |
| Feed Solid Concentration | Measured Disturbance | % | $d_2(t)$ | $C_f(t)$ | $[0, 100]$ |
| Feed Particle Size | Unmeasured Disturbance | $P_{20}$ % | $d_3(t)$ | $\mu(t)$ | [0,100] |

# 3. CONTROL TECHNIQUES AND APPLICATIONS TO PASTE TAILING PRODUCTION

This section provides a brief overview of the state of the art in paste thickener control. As stated in the introduction, control of the tailing production process is of relatively new interest in the minerals engineering and academic community. This is due to the increasing scarcity of water and tightening of environmental and social restrictions in the industry. Most of these techniques discussed come from the academic environment.

The three main strategies discussed in thickener control literature involve PID regulation, expert or fuzzy systems and some MPC strategies. Therefore, an overview of these techniques is given in sections 3.1 and 3.2.1.

Finally, in section 3.3 the state of the art concerning thickener control is examined to provide key insights in the methods used for thickener control. The conclusions drawn from this analysis will aid in selecting benchmark controllers for performance review and comparison.

## 3.1. Non-Optimal Control Techniques

Traditionally, automation and feedback have been studied from a purely regulatory o stabilizing point of view. In other words, focus is on steering a system's state or output towards a desired reference $w(t)$.

Generally, all these methods are based on the error $e(t)$ in the measured outputs or associated terms, such as its rate of change or cumulative value. They can be grouped under *non-optimal* control strategies since no optimization is done in an offline or online manner.

### 3.1.1. PID Control

PID control is by far one of the most popular controllers in the industry. This is due to its relative simplicity in design, construction and tuning and also because of its adequate setpoint tracking and disturbance rejection characteristics (Camacho & Bordons, 2007).

Usually, to avoid *integrator windup*, another feedback loop with proportional constant $K_b$ can be designed that measures the difference between the calculated controller output and the actual output (Astrom & Murray, 2012).

### 3.1.2. Expert Control

An expert system can be defined as a computer system (hardware and software) that simulates an expert's behaviour in a given area of specialization (Castillo, Guitérrez, & Hadi, 1998). Figure 3.1 shows a simplified version of an expert controller.

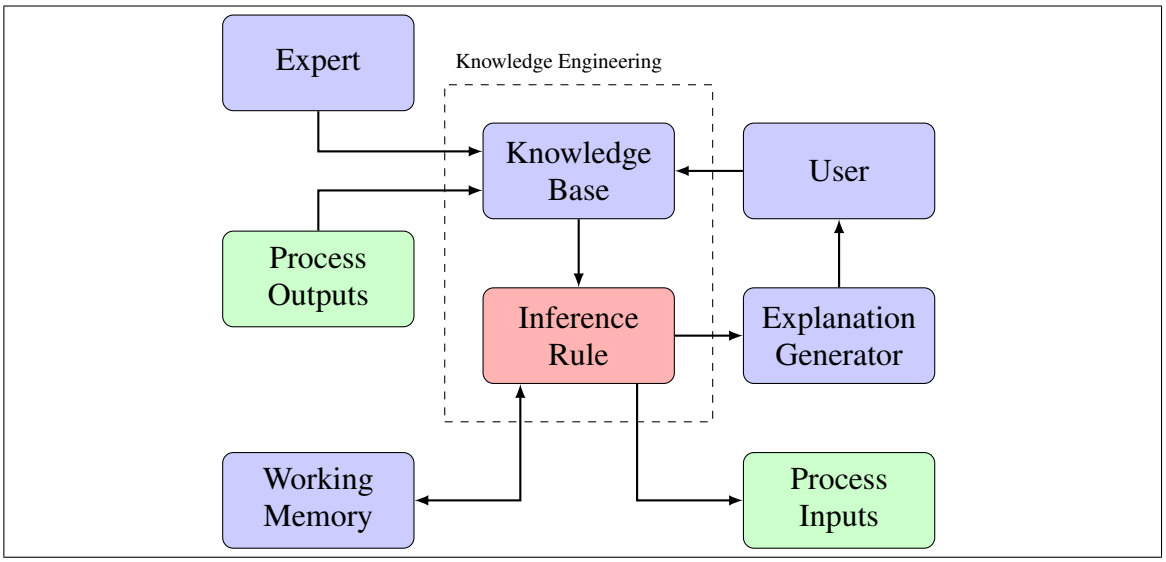


Figure 3.1. Simplified diagram of expert system. A central component of the system is the choice of the inference rule (red). Additional blocks considered for control purposes have been coloured in green.

The two most important blocks in figure 3.1 are the *knowledge base* and the *inference rule*. The first is generated through a rule base which encodes all causal relationships: it

contains premises, consequences and logical rules to literally emulate a logical process (Russell & Norvig, 2015).

Regarding the inference mechanism, many different algorithms can be used. One of the most used rules is *Modus Ponens*:

$$\frac{\alpha \Rightarrow \beta,\ \alpha}{\beta} \tag{3.1}$$

A detailed explanation of this rule and its generalizations can be found in (Russell & Norvig, 2015) and references therein.

In a control and feedback context, rules often consider the error $e(t)$ and associated terms such as:

$$\text{IF}\ \ e(t) \in \mathcal{E}_j \Rightarrow \Delta u(t) \in \mathcal{U}_k \tag{3.2a}$$

$$\text{IF}\ \ \int e(t) \in \mathcal{E'}_n \Rightarrow \Delta u(t) \in \mathcal{U}_p \tag{3.2b}$$

where $\mathcal{E}_j, \mathcal{E'}_n$ represent *linguistic values*: "high", "low", "very positive", "small negative" and so forth. These regions correspond with expert intuition, which explains why this technique became so popular in industrial control. In the case of expression 3.2, the expert controller has integral action and therefore is a *PI Expert control*. After error classification, the variation of the MV $\Delta u$ is assigned another linguistic value $\mathcal{U}_k, \mathcal{U}_p$ or as many as have been designed.

Expert systems are based on boolean logic and therefore each rule can only be true or false. An extension of this can be found in *fuzzy systems* where statements have different *degrees of truth*. The *fuzzyfication* of the rules allows for a smoother description of the system. As a consequence, the process inputs also have a fuzzy quality and have to be *defuzzified* to be sent to the system.

## 3.2. Model Predictive Control

As mentioned in chapter 1, the popularity of predictive controllers stems from their ability to handle MIMO problems with constraints as well as its robustness against disturbances.

This section provides a background of MPC and its evolution, as well as some examples of relatively common controllers. It concludes with an overview of nonlinear MPC and some stability issues. This is closely related to this research and as such must be explored in some depth.

### 3.2.1. Traditional Model Predictive Control Algorithm

The MPC strategy is one of the main tools in the industrial control realm. As industrial instrumentation generates discrete time systems, this method has its origins in the fields of optimal estimation and regulation, more specifically, the Linear Quadratic Regulator (LQR) (Rawlings & Mayne, 2012). For ease of notation, the sampling time $T_s$ will be omitted. Figure 3.2 shows a basic diagram of an MPC architecture.

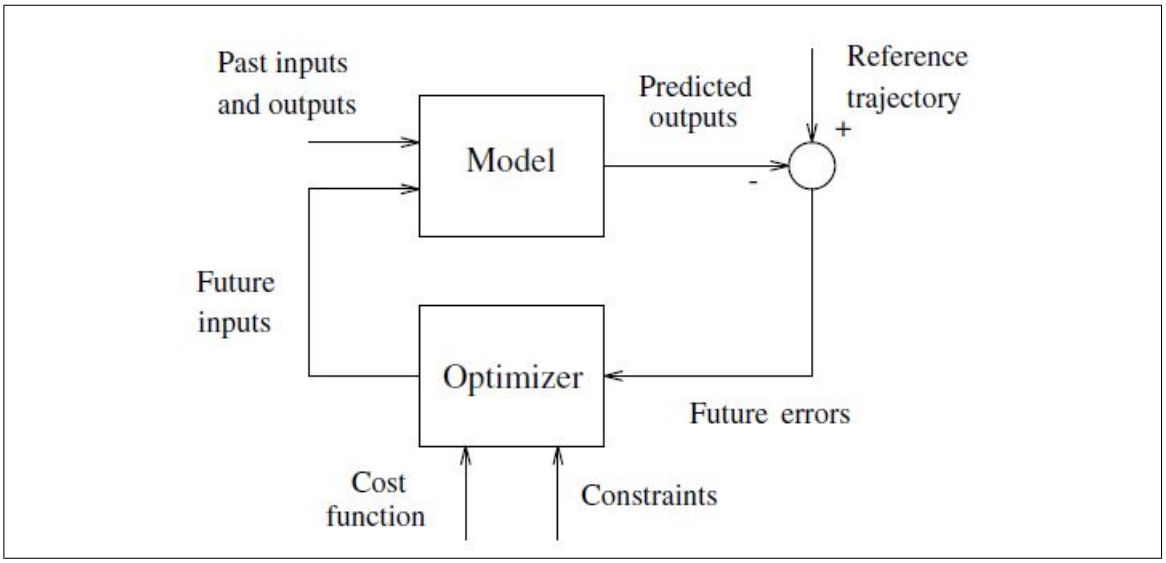


Figure 3.2. Conventional MPC strategy. (Camacho & Bordons, 2007).

The MPC algorithm is composed in its most basic form of three main parts (Camacho & Bordons, 2007):

**Predictive model:** At any given instant $t$, the future outputs for a prediction horizon $N$ must be determined. These outputs, in general, depend on past values of the outputs and inputs up to the current time and *future* values of the inputs (disturbances and manipulated variables).

**Optimal control sequence computation:** Embedded in the predictive controller is an objective function

$$V(y,u) = \sum_{j=0}^{N-1} l(y(t+j), u(t+j)) + l_N(y(t+N)) \tag{3.3}$$

which is to be optimized *online* at every instant. The *stage cost term* $l(y(t+j), u(t+j)$ can, in general, be any kind of function. The term $l_N(y(t+N))$ is called the *terminal cost* and penalizes deviations from a specified reference at the end of the prediction horizon. The decision variable of the optimization problem is the *optimal control sequence* $\{u(t+j)\}_{j=0}^{N-1}$.

As with all optimization problems, the solution is sought within a specified search space. The predictive model is an always present constraint while optionally this solution space can be restricted through *constraints* on variables that can take different forms.

For the special case of a linear or quadratic cost, a linear model and no additional constraints, there is an analytic solution; otherwise an iterative method must be used (Camacho & Bordons, 2007).

**Application of control input:** From the computed sequence, only the first value $u(t)$ is sent to the process, because for the next sampling instant $t+1$ the output can be measured and new information can be gained. The MPC strategy repeats itself in this new sampling instant.

One of the first implementations of MPC that was successfully incorporated in industrial control is Generalized Predictive Control (GPC).

A linear predictor is used for this strategy since linearization of processes around their operating point is a reasonable assumption under feedback (Camacho & Bordons, 2007). This family of predictors is known as the integrated Controlled Auto-Regressive Moving Average (CARIMA) which for the single input-single output (**SISO**) case can be described by:

$$\begin{aligned} A(z^{-1})y(t) &= z^{-d}B(z^{-1})u(t-1) + \frac{C(z^{-1})}{1-z^{-1}}e(t) \\ A(z^{-1}) &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) &= b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c} \end{aligned} \tag{3.4}$$

where $u(t)$ is the input, $y(t)$ is the output, $e(t)$ is a zero-mean white noise and $d$ is the dead time of the system. As such, no *feedthrough* component is used in this formulation and the term $\frac{C(z^{-1})}{1-z^{-1}}e(t)$ is used to model unmeasured disturbances. Measured disturbances $v(t)$ can be readily included by viewing them as uncontrollable inputs with their own transfer function (Camacho & Bordons, 2007).

This family of functions will be analyzed in depth in chapter 4, since they can be thought of as a standard or benchmark predictor function which exhibits good performance in MPC problems.

The objective function for this strategy is based on the LQR mentioned above and, for zero dead time, has the basic form:

$$V(y,u) = \sum_{j=1}^{N_y} q_j \left(\hat{y}(t+j) - w(t+j)\right)^2 + \sum_{j=1}^{N_u} r_j \Delta u(t+j-1)^2 \tag{3.5}$$

A few aspects of these formulation are relevant for this research. First, there is a distinction between a *prediction horizon* $N_y$ and a *control horizon* $N_u$. Generally, $N_u < N_y$ and in such cases the last computed value of $u(t)$ is held for that time difference.

Also, the objective function penalizes *controller effort variations* $\Delta u(t+j-1)$ rather than control inputs themselves. The form is quadratic and therefore, when no other constraints are present, its solution is analytical. Additionally, by tuning and manipulating the relationship between $q_j$ and $r_j$ the controller response can be made more agressive or sluggish. The complete solution and form of the control law can be found in numerous books and articles, such as (Camacho & Bordons, 2007) and references therein.

An input-output formulation, such as the one expressed in equation 3.4, is problematic in some systems. For non minimum phase systems, for example, these zeros become unstable poles. When scaling to a MIMO framework - which in the previous context reduces to estimating transfer matrices - transmission zeros pose an important problem to MPC (Camacho & Bordons, 2007).

As processes become more complex and computational capabilities grew, state space formulations of MPC were introduced. A more general description of a real system to be controlled in the industry corresponds to that depicted in figure 3.3.

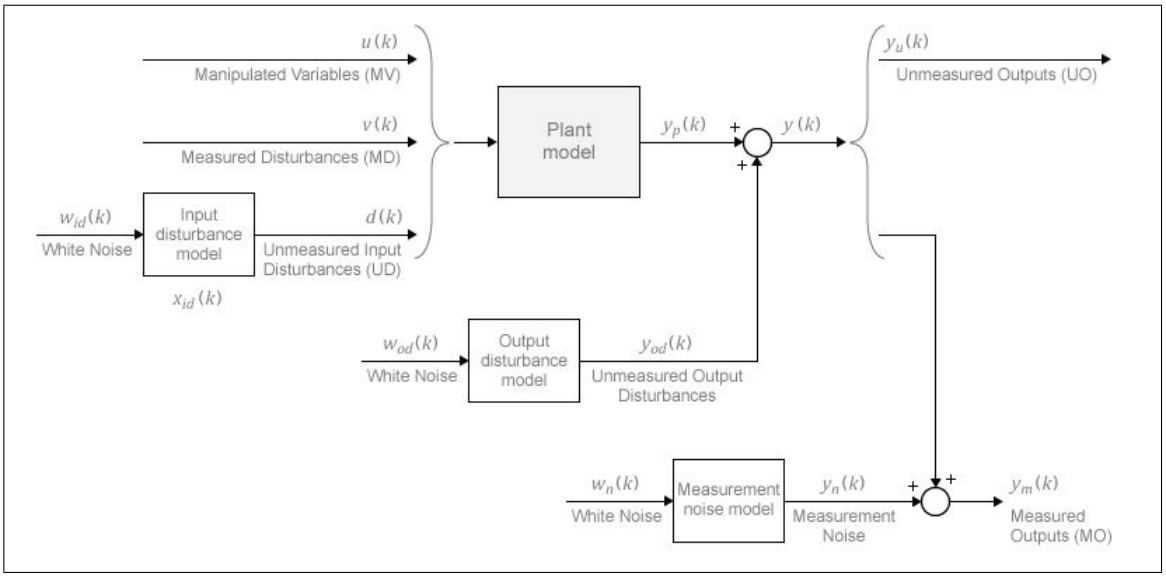


Figure 3.3. A general system for predictive control (The MathWorks, 2018).

The system is subject to input and output measured and unmeasured disturbances, as well as noise. In general, each of these components can be modeled independently. Even though a state space formulation of GPC can be obtained, figure 3.3 can serve as a new perspective on the MPC problem.

For the system above, an augmented state $x_a(t)^T$ vector can be constructed: its entries hold the states of each dynamical system in figure 3.3. As such, the dimension of the sate is larger than that of the system under control and represents every dynamical component.

There are many possible formulations for a state space MPC. A fairly common and natural one comes from using the augmented state mentioned above and generating the prediction equations directly from it. The above system - if LTI - can be represented by the equations:

$$\begin{aligned} x_a(t+1) &= Ax_a(t) + Bu_o(t) \\ y(t) &= Cx_a(t) + Du_o(t) \end{aligned} \tag{3.6}$$

where $u_o(t)^T$ represents the *observer* inputs: similar to $x_c(t)^T$, it is comprised of all the inputs to each system. Naturally, the matrices $(A, B, C, D)$ have some elements set to zero according to the effect of the states on each system.

In real control applications, the state vector is usually partially known, mainly through the system outputs. A state estimation $\hat{x}_a(t|t)$ is required to solve the MPC problem, particularly regarding predicting the outputs for the horizon.

The Kalman Filter (KF) is the optimal state estimator for LTI systems with uncorrelated process and measurement noise and is the basis of most of the linear MPC applications (Rawlings & Mayne, 2012). However, more complex systems demand more accurate state estimator. In that direction, several enhancements exist to the conventional Kalman filter, most importantly the Extended Kalman Filter (EKF) and more recently the Unscented Kalman Filter (UKF) (Rawlings & Mayne, 2012).

By estimating the state, the predictions can be obtained for the entire horizon through the last equation in 3.6. A full deduction of these equations can be found in numerous books (Rawlings & Mayne, 2012).

The last aspect needed to encapsulate a complete formulation of the MPC problem is handling and inclusion of constraints. As mentioned before, for linear LTI systems with no constraints and linear of quadratic objective functions analytical solutions exist. However, constraints on output and input variables such as:

$$\begin{aligned} y &\in \mathbb{Y} \\ u &\in \mathbb{U} \end{aligned} \tag{3.7}$$

restrict the search space and call for other optimization algorithms. Even if the resulting search space is convex and closed, the addition of such restrictions - which arise naturally form process constraints - generally makes the optimization problem difficult.

For the case of a quadratic objective function with linear restrictions on output or manipulated variables and linear systems, is one of the most effective optimization methods.

*Sequential Quadratic Programming* (SQP) has been regarded as one of the most efficient techniques for MPC. The method is similar to Newton's descent, but at each major step an approximation of the Hessian of the Lagrangian is made. The solution to this problem is then used to generate a search direction for a linear solver. A full description of this algorithm can be found in (Hock & Schittkowski, 1983).

Up to this point, the major concern has been *linear* variants of MPC. As mentioned in chapter 1, the machine learning based strategy proposed in this research opens a radically different challenge. Therefore, Non Linear Model Predictive Control (NMPC) must be investigated.

### 3.2.2. Non Linear Model Predictive Control

In its essence, NMPC is another variant of predictive control. The nonlinearity arises from the fact that the state space or output equations are nonlinear:

$$\begin{aligned} x(t+1) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \end{aligned} \tag{3.8}$$

The NMPC algorithm is in fact identical to that mentioned in section 3.2.1. However, as the predictive model is in this case non linear, the amount of issues associated with solving the online optimization problem grows large.

Ensuring *closed-loop stability* is an important issue in NMPC. For linear predictive controllers, the cost function can be used to prove Lyapunov stability for most MPC problems provided an optimal sequence exists, that is, the problem is *feasible* (Rawlings & Mayne, 2012).

In NMPC strategies, additional steps must be taken to guarantee stability results. The initial approach consists on adding *terminal constraints* on the state:

$$x(t+N) \in \mathbb{X}_f \tag{3.9}$$

where the set $\mathbb{X}_f$ is a terminal set. However, terminal constraints can impose feasibility issues in the optimization problem. For this reason, this constraints can be softened by using a terminal cost such as in expression 3.3 (Grune, 2017). If the objective function can be guaranteed to be always decreasing except in this terminal set - regular Lyapunov stability conditions - then NMPC stability can be proven (Rawlings & Mayne, 2012).

Stability results require *global* optimality in the optimization problem. Since the prediction model embedded in NMPC is nonlinear, the constraints that derive from it conform a non-convex search space for the algorithm. For this reason, NMPC is in the general case a *suboptimal predictive controller* since it can ensure local minima (Rawlings & Mayne,

2012). Details on all of these issues can be found in (Rawlings & Mayne, 2012) and (Grune, 2017).

Regarding solution methods for the online optimization problem, their complexity will depend on the nonlinear system at hand. As mentioned in section 3.2.1, the SQP algorithm can be used to solve these sort of optimization problems *provided the nonlinear constraints are twice differentiable* (Grune, 2017). If finite differences are not used, then the NMPC is required to use a *gradient free optimization* strategy (Grune, 2017). The main restrictions associated with these approaches concern computational complexity, time and, of course, the optimality gap generated by the solution.

## 3.3. Recent Developments in Thickener Control

The first important division in thickener control is between simulation and real applications. Academic literature shows a vast number of experiences regarding the first.

### 3.3.1. Simulation Approaches to Thickener Control

The first approaches found in control literature consider strategies as the ones mentioned in section 3.1 (Segovia et al., 2011) (Betancourt et al., 2014) (Xu et al., 2015). These are summarized in table 3.1.

An extensive body of work regarding paste thickener control can be found in the publications of the author Chee Keong Tan and associates. Investigation focuses on MPC control strategies for paste thickeners and therefore is central for this research.

An MPC based on a Kalman predictor is proposed for paste thickener control in a simulation environment (Tan, Setiawan, Bao, & Bickert, 2015). A dynamic model based on the work of (Burger, Karlseb, & Towers, 2005) is adopted and validated using real plant data. Steady-state operating conditions are obtained by linearizing the system around a desired operating point. A linear MPC scheme is developed with MV the underflow rate

Table 3.1. Revision of state of the art thickener control techniques in simulation environments (Segovia et al., 2011) (Betancourt et al., 2014) (Xu et al., 2015).

| | **Controller Structure** | **CV, MV and DV** | **Results, Advantages and Drawbacks** |
|---|---|---|---|
| Segovia et al. | PID expert-fuzzy | CV: U.S.C., bed level<br>MV: Underflow Rate<br>DV: Feed rate | Fuzzy controller gives 2% less squared error. Complex tuning |
| Betancourt et al. | Open loop control through process model | CV: U.S.C., bed level<br>MV: Underflow rate and flocculant dosage<br>DV: Feed rate, F.S.C. | Good overall results. Open-loop expert strategy. |
| Xu et al. | Master-slave: expert governs regulatory slave loops. Physical models. | CV: U.S.C., bed level and torque<br>MV: Underflow rate and flocculant dosage<br>DV: Feed rate, F.S.C. | Better setpoint tracking than PI control. Disturbance rejection is approximately equal to PI control. |

and tracking underflow solids concentration while keeping interface levels constrained. Results show an adequate control of both variables. No unmeasured disturbances are considered and rake torque constraints have not been included in the strategy.

Figure 3.4 shows the results of the one-step ahead predictive model - the blue signal in figure 3.4a - as well as a comparison of the designed MPC strategy (blue) to an existing control system (red) applied on the simulator against a disturbance variable (green) in figure 3.4b. It can be seen that large improvements can be made.

An important improvement, both in controller performance and MPC design strategies, is achieved in a subsequent study (Tan, Tippett, & Bao, 2016). The use of multiple timescales is explored to cope with both fast and slow dynamics.

An MPC with a non-uniformly spaced optimization horizon is developed to address said issue. This controller is compared to the one designed in the previous article since it is essentially the same control problem (Tan et al., 2015). Both MPC schemes are based on linearized first principle models. An important contribution of this article is a stability proof of the MPC strategy developed, which had not been established for non-uniform

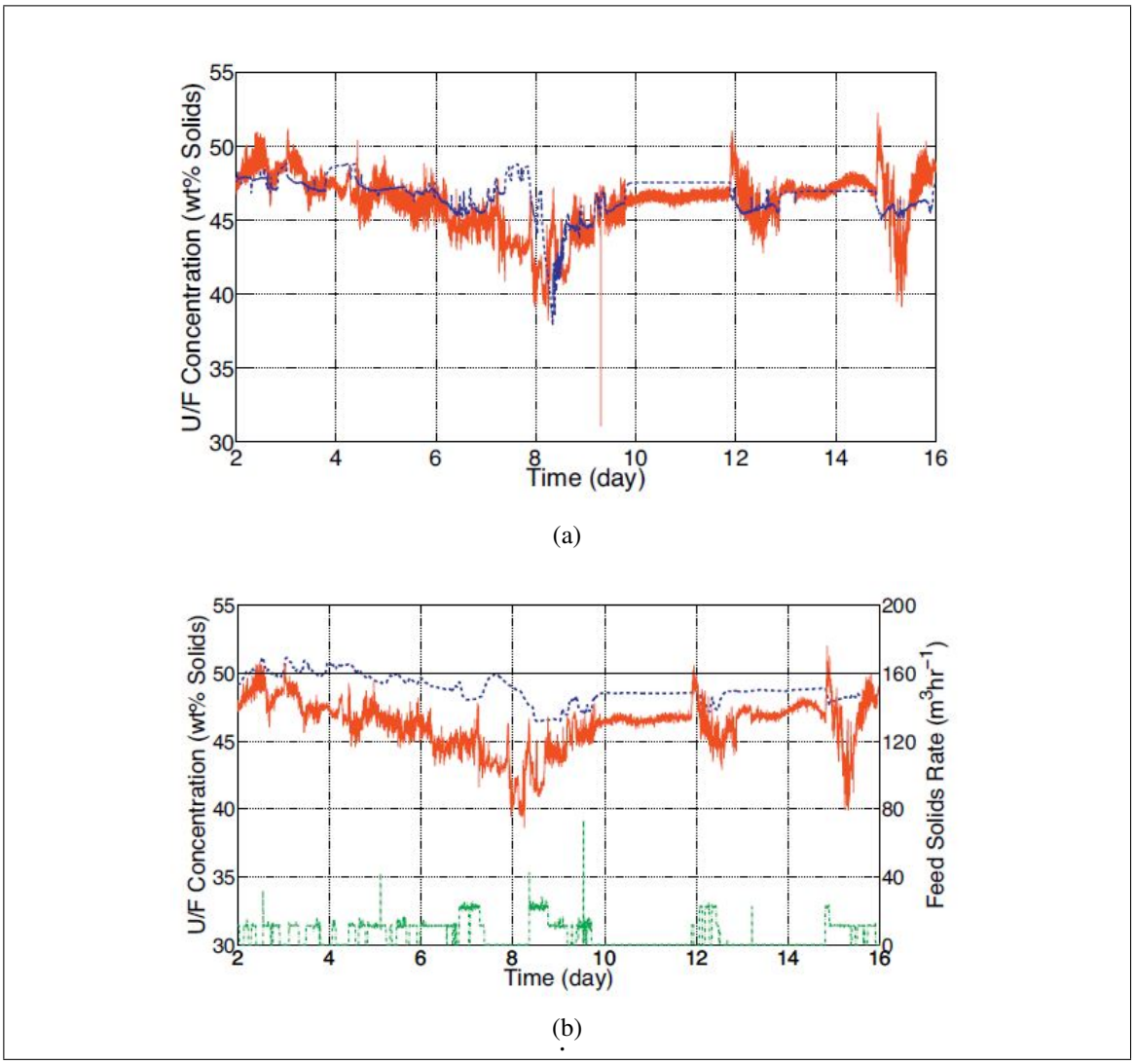


Figure 3.4. Comparison of MPC control with existing control strategy in a simulation environment (Tan et al., 2015). (a) U.S.C. prediction throughout the horizon. (b) Controlled U.S.C. through existing strategy (red) and simulated MPC (blue). Feed solid rate (green is shown throughout the horizon.

spaced optimization horizon. Results show an almost equal controller performance but with 50 times less computational effort.

Finally, the MPC strategy is expanded by adding rake torque constraints (Tan et al., 2017) (Rudman, Paterson, & Simic, 2010). The rake torque model corresponds with the

one used in this research. Extended Kalman Filters (EKF) are used for torque prediction while the MPC uses the non-uniform horizon presented in the previously mentioned article.

As such, this strategy and control design problem is the most similar to the one illustrated in 2.5, with the difference being the underflow rate as the only MV and no consideration of unmeasured disturbances. Results prove again to be encouraging and capable of handling all constraints on inventory level and torque while keeping underflow solid concentration close to its setpoint.

### 3.3.2. Real Experiences in Thickener Control

The work by Ojeda and Bergh deals with expert control of a thickener (Ojeda et al., 2014). The authors indicate that strategies with classical regulatory loops are not usually successful mainly due to tuning problems. The proposed expert controller is actually used on an industrial scale thickener.

The main advantage of the approach mentioned above is that no plant model is needed. The controller makes decisions based on a rule system designed for input and output relationships. These are translated into an expert system of about 25 rules. The strategy is effective in reducing abrupt variations of the controlled variables. However, this strategy is developed without orientation to optimality: it is assumed that expertise entails good performance.

A similar approach is developed by other authors, first by simulation and then in real application (Chai, Jia, Li, & Wang, 2016). However, the strategy proposed is hierarchical: a fuzzy-expert system switches between different possible values for the underflow rate reference based on discharge concentration. Later, an optimal PI control with one-step prediction regulates the complete loop (Chai et al., 2016).

This is an important improvement because of a concrete sense of optimality. The results ensure the stability of the system and MV variations are well handled because of the

cost function (Chai et al., 2016). As in the previous strategy (Xu et al., 2015), the authors establish complex first principle models for MV and CV coupling. However, flocculant dosage remains fixed throughout all the experiment. Predictive controllers require, in addition, the identification of ARX, ARIMAX or CARIMA models (Camacho & Bordons, 2007). In the case of the authors, a low order ARX model is identified to establish this optimal one-step driver.

As encouraging as these results are the problem is of the single-input, single-output type (SISO). No research is carried out regarding disturbance rejection and the complexity of the physio-chemical models used is considerable.

### 3.3.3. Final Remarks on Thickener Control

In summary, the review in thickener control shows that simulation is regarded as a highly effective way to test control approaches. First principle models are the basis of such techniques. Like any model, there are limitations and advantages.

There are cases in which the problem is treated in a MIMO way, but this usually implies an expert or fuzzy-expert control of the plant. Classic regulatory loops prove to be deficient and, in fact, expert controllers are preferred.

Techniques involving predictive strategies require input signals rich in information to be able to identify the system correctly. This is usually something difficult to ensure due to economic restrictions (plant shut down) and physical restrictions (actuator malfunction). All MPC strategies found use either linear or linearized first principle models. Therefore, they cannot be expected to cover different operation regimes.

Most of the strategies reviewed do not manipulate flocculant dosage throughout the operation. A possible explanation to this include the fact that the effect of flocculaton over controlled variables is less known. No control schemes were found which consider unmeasured disturbances.

Since the ultimate purpose of this investigation is to design, test and validate a data driven predictive control strategy, several benchmark controllers will be implemented:

- Classic and regulatory feedback loops. These strategies will include PI or PID controllers, because of their popularity in industrial control and their integration with more advanced techniques.
- Expert control. Its design will be based on the experience accumulated in historical data and operation manuals of the plant, through the access provided by the afore mentioned Fondef project.
- Model predictive control (MPC). This approach is one of the most widely used in the current mining industry and therefore is an important benchmark to this investigation's approach.

All these strategies will help elucidate the benefits and shortcomings of the proposed data-driven predictive controller and therefore are key for an accurate description of its development. The thesis of this investigation states that a machine learning model, which is a purely data-driven method, combined with an MPC receding horizon algorithm can indeed accomplish the objectives established in the beginning of section 2.2.3.

# 4. MACHINE LEARNING FOR PROCESS CONTROL

This chapter deals with Machine Learning and its application to control and automation. The main objective of this chapter is formally introducing *Random Forests* which will be the Machine Learning method used for system identification and subsequent Model Predictive Control.

In section 4.1 a description of Machine Learning will be made focusing on its main concepts as well as mathematical background. Section 4.1.2 will analyze certain aspects of the system identification problem at hand from a theoretical point of view. A state of the art revision regarding Machine Learning and its application to control and automation is given in section 4.1.3.

Random Forests are introduced in section 4.2. This section begins with the implementation of the ARIMAX benchmark predictor and how Random Forests will be used for time series forecasting and system identification. An extensive mathematical background is provided into this technique through the concepts of *ensemble learning* and bootstrapping. This helps to understand how collections of weak learners through a sampling process can build more powerful predictors, as explained in section 4.2.3.

In section 4.3 the training, validation and testing data set is presented. As explained in chapter 1, the task undertaken involves a pseudo-real environment.

Finally, section 4.5 applies the described method to the paste thickener operation dataset. A qualitative and quantitative analysis of the results is provided. Final conclusions about the method and possible improvements are summarized.

## 4.1. Machine Learning Background

As mentioned in chapter 1, Machine Learning stems from artificial intelligence and is therefore a branch of computer science. This research field's origin can be dated back

to 1956 with numerous attempts to understand and emulate rational human behaviour (Russell & Norvig, 2015).

However, its popularity is recently new in history. Due to the constant increase in computational efficiency and memory and the availability of very large datasets, Machine Learning has become a well studied and documented science. A detailed description of its history of these topics as well as a review of its applications can be found in (Russell & Norvig, 2015).

This discipline is very much the same as a branch of probability and statistics known as Statistical Learning. The next section specifies the most relevant concepts from this area that will aid in understanding the process of learning.

### 4.1.1. Statistical Learning and Decision Theory

A function $f(X)$ is sought for predicting values $Y$ given values of $X$. The problem is quantifying how good is the accuracy of an estimate of said function. A *loss function* $L(Y, f(X))$ penalizes errors in prediction. The most common is squared error loss $L(Y, f(X)) = (Y - f(X))^2$.

The expected prediction error is

$$\text{EPE}(f) = \mathbb{E}\left\{(Y - f(X))^2\right\} = \int (y - f(x))^2 P(dx, dy) \tag{4.1}$$

for the joint probability function $P(x, y)$.

The expression above can be transformed by conditioning on $X$ and the following expression for $f(x)$ is obtained:

$$f(x) = \text{argmin}_c \mathbb{E}_{Y|X}\left(\left\{Y - c^2\right\}|X = x\right) = \mathbb{E}\left\{Y|X = x\right\} \tag{4.2a}$$

The last term is known as the *regression function* and states that the best prediction is the conditional mean at any point $X = x$. Since the joint probability function is *unknown* an estimator $\hat{Y} = \hat{f}(X)$ is used in its place (Hastie, Tibshirani, & Friedman, 2009).

The estimator function $\hat{f}(X)$ can have any desired shape. It will seem reasonable to choose linear models if the suspected relationship $Y = f(X)$ is assumed to be linear. In this sense, *ARIMAX or CARIMA models are in fact an application of Machine Learning*.

An important aspect regarding the statistical background is the distinction between *parametric* and *nonparametric methods*. The parametric family of models is comprised of all models which are specified by setting a parameter vector $p$. Examples of this type of models are linear univariate and multivariate regression, neural networks and support vector machines, among others (Hastie et al., 2009).

Nonparametric models, on the other hand, are not characterized by a bounded set of parameters. Instead, they use all gathered information - which can been encoded efficiently - and generate predictions bases on that history. The simplest method is, of course, a lookup table. Other examples involve correlation analysis and nearest neighbours (Russell & Norvig, 2015).

The next section offers a theoretical background to the identification problem. It lists certain key aspects of system identification theory that frame the validity of the proposed methods. The objective is to understand some fundamental limits in model identification that are important in the motivation of this research.

### 4.1.2. Preliminary Considerations on the Identification Problem

In classical system identification, specific input signals - steps, pseudo-random binary sequences (PRBS), sinusoids and others - allow for the identification of models of diverse complexity.

*Black box modeling* allows for studies of the plant's input-output relationships. In a typical system identification framework, step tests are usually conducted separately to inspect and measure the response of each output variable to every input.

Another important aspect of input signal design is guarantying a persistent excitation (p.e.) condition because to *estimate M parameters of a system's impulse response its autocorrelation matrix of order $M$ needs to be invertible*.

In other words, if the greatest invertible matrix $\overline{\mathcal{R}}$ is of order $M$ only $M$ different poles can be identified (Dahleh, 2005). For example, a step input is *p.e.* of order 1, while a PRBS signal of period $M$ is of order $M$. An Auto Regressive Moving Average (ARMA) process is *p.e.* of any order. Classifying operational data in these terms is an interesting question.

Since this data is in *feedback* and the process is under control, it is difficult to examine process nonlinearities. Furthermore, all inputs of the process are active simultaneously for the most part, so tracing the relationship between inputs and outputs becomes an increasingly difficult task. For example, estimating the delay between an input-output pair - which can be done in its most basic form visually - becomes a daunting task.

All the analysis made in this section highlights the value and challenges of the data-driven approach at hand. As the research unravels, some of the questions will be answered within the scope of paste production.

### 4.1.3. Machine Learning Techniques and Model Predictive Control

As Machine Learning is such a vast field of research, the state of the art review will focus on techniques for time series prediction and Model Predictive Control. By doing this, it is possible to filter out alternatives that can prove to be unsuccessful in the long run. There are two main aspects through which Machine Learning has penetrated the field of automation and MPC: prediction model generation - in other words, system identification - and learning of controller structure.

As the complexity of industrial processes increased, more sophisticated model identification algorithms were applied and combined with the predictive strategy. Artificial Neural Networks (ANN) have been used extensively in MPC formulations and as such

constitute an important field of research. Recently, recursive neural networks have gained popularity because of their ability to represent and learn internal states of the system being modeled (Kamesh & Rani, 2017).

Learning a controller structure poses a completely different approach to model predictive control. Hence, it is denominated *Model Free Predictive Control* (MFPC) to highlight the fact that there is no predictive model embedded in the controller (Hou & Jin, 2014). The main research topics include methods such as Iterative Learning, Lazy Learning and Model Free Adaptive Predictive Control (Hou, Liu, & Tian, 2017) (Cao, Zhou, & Hou, 2009) (Jin, Hou, & Chi, 2013).

Use of forms of ANN in MPC strategies is highly exploited. Various articles can be found regarding mechanical and physical problems as well as process control scenarios. However, ANN use in MPC is afflicted the problem of nonlinear and nonconvex optimization. On the other hand, MFPC is specially useful for adaptive control and time-varying systems. Therefore, both of these approaches will be discarded.

A research team from the University of Pennsylvania has done extensive work in the crossover of Regression Trees, Random Forests and MPC (Jain, Smarra, & Mangharam, 2017). The strategy does not use *any* first principle model or input tests for system identification. Its application is building energy management, which is a complex MIMO non linear system. Furthermore, process time constants are slow and the process is exposed to various disturbances, some of which rely on weather forecasts. As such, this experience is viewed as an *ideal frame of reference because of possible similarities with paste production control*. Minimization of MV effort, which is energy consumption in heating, is also considered in the objective function.

*Regression Trees* are a supervised non-parametric model which uses recursive partitioning to learn interactions between variables. At each node in the tree, an optimization problem is solved to choose a *split variable and value*. The seminal algorithm for learning regression trees is CART. Node generation continues down a branch until a minimum leaf

size is reached and the depth of the tree is adjusted (Hastie et al., 2009). Random Forests, on the other hand, are ensemble predictors of trees and will be covered thoroughly in subsequent sections.

Figure 4.1 illustrates the *recursive partition strategy* used by the authors to generate the predictive models (Jain, Smarra, & Mangharam, 2017). For *each prediction horizon step* $h$ a different tree is grown only on disturbances and past values of the outputs $X_d$, which eases computational complexity. In each leaf of the tree, a *linear* model is fit relating the outputs and *only* manipulated variables $X_c$, which is a valid assumption for sufficiently deep trees (Jain, Mangharam, & Behl, 2016). Hence, typical online optimization issues for predictive control can be avoided.

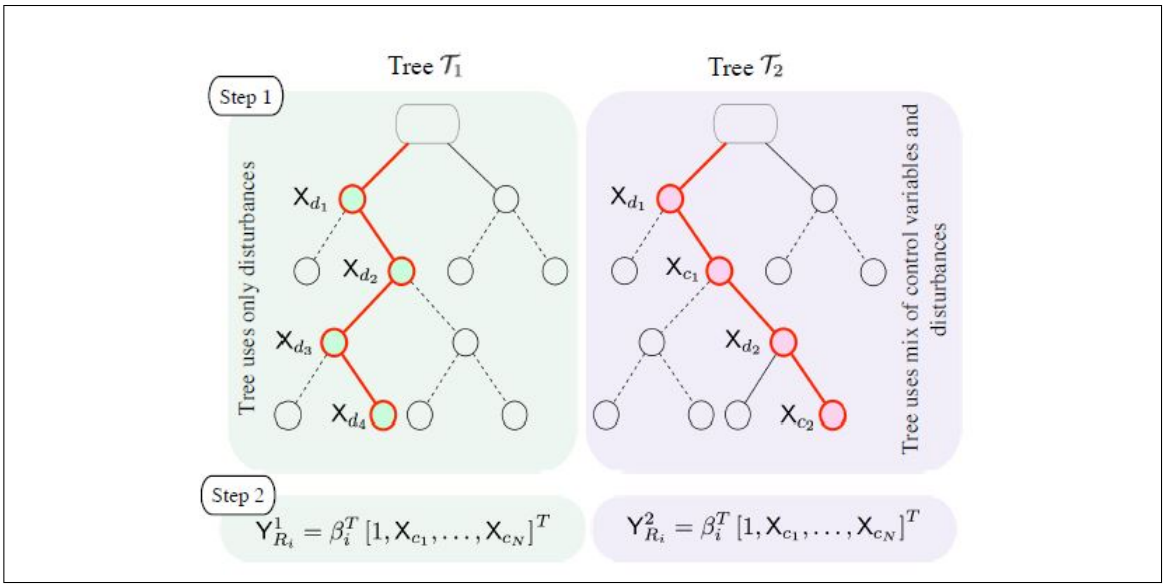


Figure 4.1. State of the art random forest strategy used with MPC. The separation of variables method consists on using random forests on disturbances and linear models for manipulated variables (Jain, Smarra, & Mangharam, 2017).

In (Jain et al., 2016) the Random Tree-MPC (RT-MPC) method is proposed and further developed in (Jain, Behl, & Mangharam, 2017) and (Jain, Smarra, & Mangharam, 2017) to a full Random Forest-MPC (RF-MPC). In the last article, a full comparison of both

methods and traditional MPC - which uses a linearization of the exact plant model - is given, which is summarized in table 4.1.

Table 4.1. Comparison of conventional, regression tree and random forest MPC strategy. (Jain, Smarra, & Mangharam, 2017)

| | **Predictor Structure** | **Prediction MSE** | **Controller Performance** |
|---|---|---|---|
| MPC | Linearized Plant Model | Not studied, presumed best | Mean Total Cost: 22.60<br>Mean Input Cost: 17.16 |
| RT-MPC | One *tree* for each prediction step | 1-step ahead: 0.18<br>6-step ahead: 0.41 | Mean Total Cost: 204.55<br>Mean Input Cost: 16.84 |
| RF-MPC | One *forest* for each prediction step | 1-step ahead: 0.08<br>6-step ahead: 0.14 | Mean Total Cost: 39.26<br>Mean Input Cost: 15.12 |

Table 4.1 shows that the ensemble approach to prediction and control is vastly more effective than that of a single tree. Furthermore, the mean total cost (MPC objective function) of RF-MPC *less than twice* that of the regular MPC while obtaining *less* input cost (MV average).

These results are key to motivate the study of a random forest based predictive controller. As the MPC is based upon the exact mathematical model of the plant being controlled, the fact that the RF-MPC strategy is almost as efficient encourages its exploration.

This research will be based on the last methodology explored, namely, *Random Forest Model Predictive Control* (RF-MPC). This decision can be supported by the following conclusions:

- Time series forecasting with random forests appears to be a less exploited field of research since prediction problems are usually static. Their use in time-series prediction and forecasting can be regarded as an additional contribution of this investigation.
- Research shows that the only experience of RF-MPC has been cited above with promising results (Jain, Smarra, & Mangharam, 2017).

- Random forests are extremely useful for non linear modeling. A hypothesis of this research is that paste thickeners are subject to dramatic nonlinearities as described in chapter 2.
- Building energy management problems are nonlinear MIMO systems subject to strong disturbances (Jain, Smarra, & Mangharam, 2017).

However, substantial differences will be observed with respect to this work. A thorough explanation of the proposed method will be given in 4.5.1, but they can be synthesized in two main aspects:

- No separation of variables will be imposed on the dataset. Predictive Random Forest models will be *dynamic fully nonlinear* estimators. Their construction will depend on lagged samples of outputs, disturbances *and manipulated variables*. This approach will be chosen - despite issues with the online optimization problem - because thickeners are believed to be highly nonlinear.
- In the strategy reviewed, the predictions made are not fed back to the Random Forest for multi-step strategies. The models generated by the authors only use forecasts of *external disturbances* and not those that *Random Forests themselves* produce. This is a rather odd decision, since it is assumed that the predictive power of Random Forests is sufficiently good.

Initially, *only one* random forest predictive model will be built, as opposed to the approach cited above. It will be constructed by minimizing the one-step ahead prediction error, under the hypothesis that its performance will not degrade for longer horizons. A comparison to a bank of random forests - one for each prediction step, as in (Jain, Smarra, & Mangharam, 2017) - will be made.

In the following section, the mathematical background for this technique is analyzed and special focus is given to its application to time-series forecasting.

## 4.2. Random Forests for Time Series Prediction

To fully understand the application of Random Forests in this research, it is necessary to bridge the gap between this method and system theory, which is essential for the controller that is ultimately being designed. To do this, an introductory description of input-output system equations will be discussed.

### 4.2.1. Time Series Forecasting and Prediction

In its most basic form, an input-output model for the SISO case can be expressed as:

$$y(t) = f(y(t-1), y(t-2), ..., y(t-n_a), u(t), u(t-1), ..., u(t-n_b)) \tag{4.3}$$

for $(n_a, n_b)$ the *order* of the auto-regressive and exogenous part, respectively. The properties of $f$ will determine the type of relationship between inputs and outputs.

A linear time invariant **(LTI)** discrete system can be described completely by its responses to *inputs* $u(t)$ and *unmeasured disturbances* $v(t)$. While the *impulse response* $g(t)$ describes the relationship between the output and $u(t)$, $v(t)$ can be expressed as white noise $e(t) \sim N(0, \sigma^2)$ through a filter $h(t)$:

$$y(t) = \sum_{k=0}^{\infty} g(t)u(t-k) + v(t) = \sum_{k=0}^{\infty} g(t)u(t-k) + \sum_{k=0}^{\infty} h(t)e(t-k) \tag{4.4}$$

This way, expression 4.4 can be transformed to its transfer function equivalent:

$$y(t) = G(z^{-1})u(t) + H(z^{-1})e(t) \tag{4.5}$$

Extrapolation of this concept to MIMO systems is, in the linear case, straight-forward. All outputs and inputs become vectors of their respective dimensions and transfer functions become matrices containing each transfer function pair. *Measured disturbances* $d(t)$ can be regarded as additional inputs, with their respective transfer functions.

#### 4.2.1.1. Benchmark Predictor: ARIMAX Model

*Equation error* models transform use rational expressions for $G(z^{-1})$ and $H(z^{-1})$ (Ljung, 1999).

**Definition 4.1.** *Output error models can be defined through the following set of equations*

$$\begin{aligned} A(z^{-1})y(t) &= z^{-L}B(z^{-1})u(t-1) + \frac{C(z^{-1})}{D(z^{-1})}e(t) \\ A(z^{-1}) &= 1 + a_1 z^{-1} + \cdots + a_{n_a} z^{-n_a} \\ B(z^{-1}) &= b_0 + b_1 z^{-1} + \cdots + b_{n_b} z^{-n_b} \\ C(z^{-1}) &= 1 + c_1 z^{-1} + \cdots + c_{n_c} z^{-n_c} \\ D(z^{-1}) &= 1 + d_1 z^{-1} + \cdots + d_{n_d} z^{-n_d} \end{aligned} \tag{4.6}$$

It is important to note that there is *no feedthrough* component from $u(t)$ to $y(t)$ and an input delay $L > 0$ is present $z^{-L}$.

As $D(z^{-1})$ can have in theory any structure, some characterizations are more useful than others. For systems that are subject to slow disturbances, a useful representation is their *integration*, or in other words, applying a low pass filter.

**Definition 4.2.** *The Auto Regressive Integrated Moving Average with Exogenous Input (ARIMAX) structure is the following output error equation*

$$A(z^{-1})y(t) = z^{-L}B(z^{-1})u(t-1) + \frac{C(z^{-1})}{1-z^{-1}}e(t) \tag{4.7}$$

*for polynomials of order $n_a, n_b, n_c$ respectively.*

Extensive literature exists centered around this family of models. One of the most popular and successful implementations of predictive control, GPC, is based on them (Camacho & Bordons, 2007).

Given the structural parameters $(n_a, n_b, n_c)$, the problem comes down to fitting a linear regression over collected input-output data. To do this, a dual equation can be generated for parameter $\theta$ estimation:

$$\begin{gathered} y(t) = \phi^T(t)\theta + e(t), \quad e(t) \sim N(0, \sigma^2) \\ \phi^T(t) = [-y(t-1), ..., -y(t-n_a), u(t-L-1), ..., u(t-L-n_b-1)] \\ \theta^T = [a_1, a_2, ..., a_{n_a}, b_0, b_1, ..., b_{n_b}] \end{gathered} \tag{4.8}$$

Efficient implementations for solving these problems exist in various toolbox and software. A detailed description of these algorithms, their history and implementation issues can be found in (Ljung, 1999) and references therein.

If there are $p$ measured disturbances $d(t)$ present, an explicit inclusion of them can be done in expression 4.8:

$$\begin{gathered} \phi^T(t) = \left[ -\left\{y(t-k)\right\}_{k=1}^{n_a}, \left\{u(t-k-L_u)\right\}_{k=1}^{n_b}, \left\{d(t-k-L_d)\right\}_{k=1}^{n_p} \right] \\ \theta^T = \left[a_1, a_2, ..., a_{n_a}, b_1, ..., b_{n_b}, d_1, ...d_{n_p}\right] \end{gathered} \tag{4.9}$$

where the sequencing operator $\{ \quad \}_a^b$ concatenates values in a left to right manner.

In a MIMO scenario with $n$ outputs, $m$ manipulated variables and $p$ measured disturbances the model description becomes larger. To distinguish between control inputs and measured disturbances, the orders $n_b$ and $n_p$ will be used. There will be, in general $n \times (m+p)$ distinct transfer functions with $n$ noise filters $H(z)$ to determine.

It is important to have a benchmark predictor for comparison of the proposed Machine Learning method, both in pure predictor analysis and in the MPC strategies developed. ARIMAX models will be chosen because of the following reasons:

- According to the analysis made in section 4.1.2, it is natural to use a linear predictor to fit the closed-loop data.
- Most of the industrial MPC strategies are based on linear predictors and models from the equation error family.

- Effective and computationally inexpensive parameter estimation routines exist in various toolboxes. Therefore, a vast range of the parameters $(n_a, n_b, n_c)$ can be explored.
- Disturbances affecting the process, particularly feed solids concentration and particle size distribution, as well as the process themselves, are assumed to have a slow rate of change.

### 4.2.2. Ensemble Predictors

Fundamentally, random forests can be classified as an *ensemble learning* procedure. This process can be broken down into two tasks: developing a population of base learners - *weak learners* - from the training data, and then combining them to form the composite predictor (Hastie et al., 2009).

Any classifier or predictor can be used for ensemble learning. However, since the method chosen for this research regards random forests, regression trees will be introduced in the following section.

#### 4.2.2.1. Regression Trees

Given a dataset $\mathcal{D} = (X, Y)$ comprised of $N$ observations $z_i$ of a process, where $z_i = (x_i, y_i)$, and $r$ variables $x_{i_r}$ for each observation, regression trees *divide the input space* $X$ in $\mathcal{M}$ regions recursively. Therefore, trees are a *nonparametric model*.

For example, an input space $X = x_1$ can be initially divided in $M_1 = \{x_1 | x_1 \leq s_1\}$ and $M_2 = \{x_1 | x_1 > s_i\}$ where $s_i$ are the *split points or values* of each partition. From this original node, two more branches are generated which can be further divided. As such, this procedure generates a binary partition of the space and thus the tree has a binary structure (Hastie et al., 2009). Figure 4.2 illustrates this procedure for two variables and five regions.

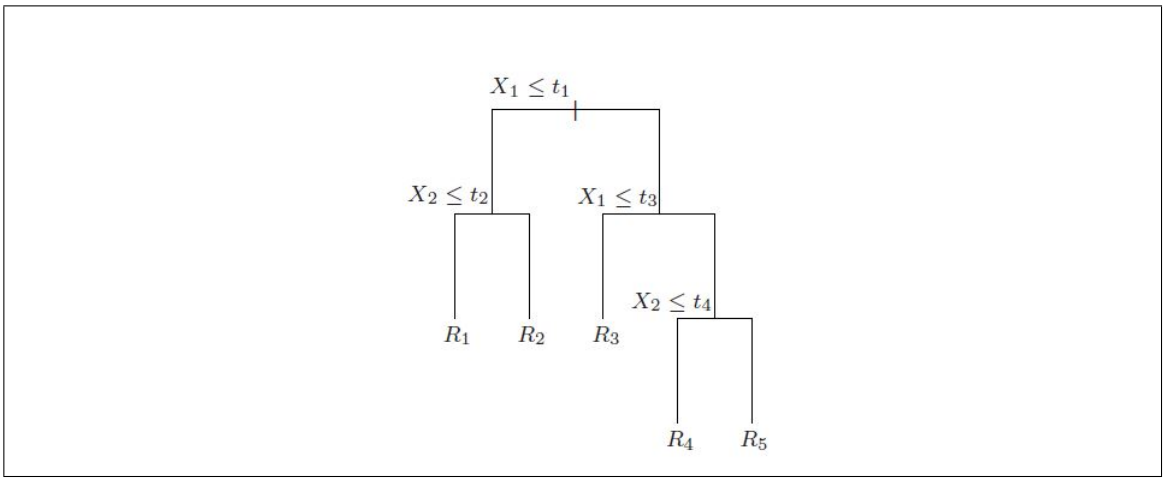


Figure 4.2. Graphical depiction of decision tree. Each node is divided according to one variable $x_i$ and a split value $t_i$ which is chosen automatically (Hastie et al., 2009). In this example, $r = 2$ and $\mathcal{M} = 5$.

For an input space with $p$ variables, the algorithm needs to automatically decide on the splitting variables, split points and also what topology (shape) the tree $T(x)$ should have. The most popular algorithm for regression trees generation is called CART - short for Classification and Regression Trees - and was developed by Breiman in 1984 (Breiman, Freidman, Olshen, Noack, & Stone, 1984). A brief summary can be found in appendix B.

The best estimator for the relationship of input with output data is the mean of the *outputs* associated with the samples in each leaf. In the example shown in figure 4.2, every leaf is dividing the initial input space in smaller regions. The $N$ samples from the dataset fall in one region or another, and therefore the best predictor is the mean of the $y_i$ that "fell" on each region. Trees can capture complex interaction structures in the data, and if grown sufficiently deep, have relatively low bias.

Tree size is a tuning parameter governing the model complexity, and the optimal tree size should be adaptively chosen from the data. Usually, the prefered strategy is to grow a deep tree $T_0$ and stop the splitting once a minimum leaf size is reached. Then the tree is pruned - cut back up to its root - through various algorithms (Hastie et al., 2009).

An important drawback with trees is their high variance because an error in the top split is propagated down to all of the splits below it. As such, regression trees are very susceptible to noise and therefore could prove to be inadequate from an automation point of view (Hastie et al., 2009).

Breiman proposed in 2001 a method for improving tree-based predictors. The proposition was to randomly sample the input space and use different subsets of the training set to grow separate trees (Breiman, 2001). By doing this, the power of the predicting ensemble grew enormously. This method is now called Random Forests and is the center of this research. The following section describes the algorithm thoroughly.

### 4.2.3. Random Forests

Random forests are based on the concept of *bagging*, which is an acronym for *bootstrap aggregating* (Efron, 1979).

*Bootstrapping* consists in drawing $B$ sets of size $N$ from the training set $\mathcal{T}$ with replacement, where a same model $f$ is fit to each bootstrap replica. Given any quantity $S(Z)$ obtained from the model, the bootstrap method allows to estimate any statistic about it by using each replica as a miniature test set. To avoid training and testing sets contamination, for each observation $z_i$ the only predictions considered are those from bootstrap replicas that do not contain said observation.

A predictor $\hat{f}_b(x)$ can be generated from each bootstrap sample. Bagging averages the prediction over a collection of bootstrap replicas, thereby reducing its variance (Hastie et al., 2009).

**Definition 4.3.** *For each bootstrap sample* $Z^b,\ b = 1, 2, \cdots, B$ *with fitted model* $\hat{f}_b(x)$ *the bagging estimate is defined by*

$$\hat{f}_{bag}(x) = \frac{1}{B}\sum_{b=1}^{B} \hat{f}_b(x) \tag{4.10}$$

This approach can dramatically reduce the variance of unstable procedures like trees, specially for squared error loss functions, leading to improved prediction (Hastie et al., 2009). A proof of this can be found in appendix B.

Bagging, as an extension of bootstrapping, generates identically distributed (i.d). predictors because of the resampling process (Hastie et al., 2009). Therefore, *the bias of the bagged ensemble of trees is the same as that of each individual tree*.

An average of $B$ independent and identically distributed (i.i.d) random variables, each with variance $\sigma^2$, has variance $\frac{1}{B}\sigma^2$. For only i.d. random variables, a correlation coefficient $\rho$ exists between them. Applying the bias-variance decomposition, the expectation of the squared error loss of a bagged ensemble of regression trees is:

$$\begin{aligned} \mathbb{E}\left\{T_{bag}\right\} &= \text{Var}\left\{T_{\text{bag}}\right\} + \text{Bias}^2\left\{T_{\text{bag}}\right\} \\ &= \rho\sigma^2 + \frac{1-\rho^2}{B}\sigma^2 + \text{Bias}^2\left\{T_{\text{bag}}\right\} \end{aligned} \tag{4.11}$$

The bias portion of this error cannot be adjusted or manipulated whatsoever, unless a better training set is provided or other learners are trained. However, as $B$ increases, the second term in the variance tends to zero and hence the magnitude $\rho$ of the correlation in the ensemble limits the error.

Random forests deal with this precise issue (Breiman, 2001). By *randomly selecting the input variables at each partition* the correlation coefficient $\rho$ is decreased radically - mimicking an i.i.d distribution - without increasing the overall variance $\sigma^2$ as much.

The number of variables used in each split is adjusted through a parameter $r^*$. Too small a value of this parameter could make the variance $\sigma^2$ of each tree become too large due to underfitting. For regression, studies have shown that the optimal value is $r^* = \frac{r}{3}$ (Hastie et al., 2009).

The Random Forest algorithm is described below (Breiman, 2001):

ALGORITHM 4.1. *To construct a Random Forest predictor follow the procedure*

*1) For* $b = 1$ *to* $B$

*a) Obtain a bootstrap sample* $Z_b$ *of size* $N$ *from the training set* $\mathcal{T}$.

*b) Grow a tree* $T_b$ *of the bootstrapped data, by repeating the following steps for each terminal node of the tree until the minimum leaf size* $l_{min}$ *is reached.*

*i) Select a subset of* $r^*$ *variables of the total* $r$ *predictor variables.*

*ii) Determine the best pair* $(j, s)$ *among the* $r^*$ *variables selected through the CART algorithm.*

*iii) Split the node into two child nodes.*

*2) Store the ensemble of trees* $\{T_b\}_1^B$

*The* Random Forest predictor *is then*

$$\hat{\mathcal{F}}(x) = \hat{f}_{rf}^B(x) = \frac{1}{B}\sum_{b=1}^{B} T_b(x)$$

Some important features are associated with random forests. One of its many advantages is the use of *out of bag samples* (OOB). For each observation $z_i$ in the training set, a record is kept of the trees that were grown using that sample. For these trees, that observation is called an *in bag sample*. Out of bag samples can be used to estimate the prediction error.

Trees allow for a measurement of predictor importance. At each split in each tree, the improvement in the split-criterion is the importance measure attributed to the splitting variable, and is accumulated over all the trees in the forest separately for each variable (Hastie et al., 2009).

However, the most expressive indicator of predictor relevance in random forests is the *OOB Permuted Delta Error*. When $T_b$ is grown, the OOB samples are passed down the tree, and the prediction accuracy is recorded. Then the values for the $j$ variable are randomly permuted in the OOB samples, and the accuracy is again computed. The decrease

in accuracy as a result of this permuting is averaged over all trees, and is used as a measure of the importance of variable $j$ in the Random Forest (Hastie et al., 2009).

Up to this point, random forests have been described mathematically by means of the statistical concepts behind the method. However, the purpose of this research is their integration with a predictive controller. As such, conceptual treatment must be given regarding the use of random forests in time series prediction.

#### 4.2.3.1. Random Forests for Time Series Forecasting

The final objective of the random forests developed is to predict all the process outputs. Therefore, the predictive model needs to be expressed in the same manner as the input-output expression 4.3.

As reviewed in section 4.1.1, Machine Learning algorithms refer to input and output *spaces* rather than systems. The translation to system theory, however, can be done with very few steps.

Since from a control perspective having feedthrough is difficult, a natural first step is defining the predictive model as

$$\hat{y}(t+1) = \hat{\mathcal{F}}(u(t)) \tag{4.12}$$

which corresponds to equation 4.3 with $n_a = 0$, $n_b = 1$. For simplicity, the system sampling time $T_s$ is being omitted. All lags are a multiple of this fundamental rate.

To grow the forest, algorithm 4.1 is followed by defining each observation $z_i$ as the pair

$$(y(t), u(t-1))$$

and training accordingly.

For a MISO system in which $u \in \mathbb{R}^m$ and $d \in \mathbb{R}^p$, equation 4.12 can be written as table with $N$ entries. This is equivalent to *shifting back the output sequence by one sample* and concatenating the output matrix such that:

$$\left[\begin{array}{c|cccccccc} y(1) & u_1(0) & \cdots & u_m(0) & d_1(0) & d_2(0) & \cdots & d_p(0) \\ y(2) & u_1(1) & \cdots & u_m(1) & d_1(1) & d_2(1) & \cdots & d_p(1) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y(t) & u_1(t-1) & \cdots & u_m(t-1) & d_1(t-1) & d_2(t-1) & \cdots & d_p(t-1) \\ y(t+1) & u_1(t) & \cdots & u_m(t) & d_1(t) & d_2(t) & \cdots & d_p(t) \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ y(N) & u_1(N-1) & \cdots & u_m(N-1) & d_1(N-1) & d_2(N-1) & \cdots & d_p(N-1) \end{array}\right] \tag{4.13}$$

The random forest predictor is now:

$$\hat{y}(t+1) = \hat{\mathcal{F}}\left(\vec{u}(t)^T, \vec{d}(t)^T\right) \tag{4.14}$$

For a MIMO system in which $y \in \mathbb{R}^n$ and under the assumption that *outputs are completely uncoupled*, $n$ such predictive functions could be fit:

$$\vec{y}(t+1)^T = \hat{\mathcal{F}}(\varphi_i) \Rightarrow y_i = \hat{\mathcal{F}}^B\left(\vec{u^i}(t)^T, \vec{d^i}(t)^T\right) \tag{4.15}$$

where $\varphi_i$ are the predictor variables on which forest $i$ is grown.

It is rather natural to continue this process to incorporate the orders $(n_b^j, n_p^l)$ for input $j \in 1, ..., m$ and disturbance $l \in 1, ..., p$:

$$\hat{y}_i(t+1) = \hat{\mathcal{F}}_i\left(\left\{\left\{\{u_j(t-k)\}_{k=0}^{n_b^j-1}\right\}_{j=1}^{m}, \{d_l(t-k)\}_{k=1}^{n_p^l-1}\right\}_{l=1}^{p}\right) \tag{4.16}$$

The expression given in 4.16 is analogous to making $A(z) = 1$ in the equation error context. This type of response is a *finite impulse filter* because the system has no memory

of its past values (Ljung, 1999). Now, a full description of the random forest predictor can be given by adding *lagged terms of the output* to the input space.

**Definition 4.4.** *The full Random Forest predictor is expressed as:*

$$\hat{y}_i(t+1) = \hat{\mathcal{F}}_i\left(\{y(t-k)\}_{k=0}^{n_a^i-1}, \left\{\{u_j(t-k)\}_{k=0}^{n_b^j-1}\right\}_{j=1}^{m}, \left\{\{d_l(t-k)\}_{k=0}^{n_p^l-1}\right\}_{l=1}^{p}\right) \tag{4.17}$$

$$\hat{y}_i(t+1) = \hat{\mathcal{F}}_i(t) = \hat{\mathcal{F}}_i(\phi_i(t)) \tag{4.18}$$

*where $\phi_i(t)$ is the* full predictor *for a one-step ahead prediction in the sampling instant $t$.*

To grow the random forests, algorithm 4.1 is applied to a training matrix with a similar structure as 4.13 but which includes lagged terms of its output. As the order $n_a$ grows, the size of predictor $\phi_i$ does too, which is referred to as *state-space explosion* in this context.

This expression is to be compared - for each $i$ output of the $n$ total one $\hat{\mathcal{F}}$ exists - to that in equation 4.9 and the general ARIMAX structure in equation 4.7. It can be seen that expression 4.18 is a *non-linear filter*, as opposed to the ARIMAX structure. Both models, however, rely on past sample data.

Additionally, ARIMAX modeling requires estimating the parameter $\theta$ in expression and therefore is a *parametric model*. Random forests, on the contrary, use the collected data directly and is trained through algorithm 4.1.

Another important difference between the ARIMAX and random forest approach is that the latter *does not account for unmeasured disturbances*. ARIMAX structure, on the other hand, estimates the $C(z)$ polynomial of order $n_c$ and therefore considers this effect.

As forests are trained through the strategy introduced in this section, their focus is *minimizing the one-step ahead prediction error*. A typical problem associated with one-step ahead predictions - specially for MIMO, non-linear systems subject to disturbances and noise - is that a persistence model $\hat{y}(t+1) = y(t)$ can be a good predictor specially if the sampling time is small (Ljung, 1999).

MPC requires a prediction horizon which strictly speaking could be of one sample beyond the current instant. Such a strategy makes sense when a comprehensive window of future data is estimated, which in turn translates to a horizon in the order of the response time of the system (Grune, 2017).

There are two main strategies for multi-step $h = 1, ..., H$ forecasting (Bontempi, Ben Taieb, & Le Borgne, 2013):

**Recursive strategy RF$_r$:** This method trains first a one-step model and then uses it recursively for returning a multi-step prediction. Therefore, forecasts of all inputs for the length of the prediction horizon are needed. In essence, the predictor structure is

$$\hat{y}_i(t+h) = \hat{\mathcal{F}}_i\left(\phi_i(t+h-1)\right), \quad h = 1, ..., H \tag{4.19}$$

for each output $i = 1, ..., n$. Figure 4.3 illustrates this strategy for an example one output, one input and one measured disturbance.

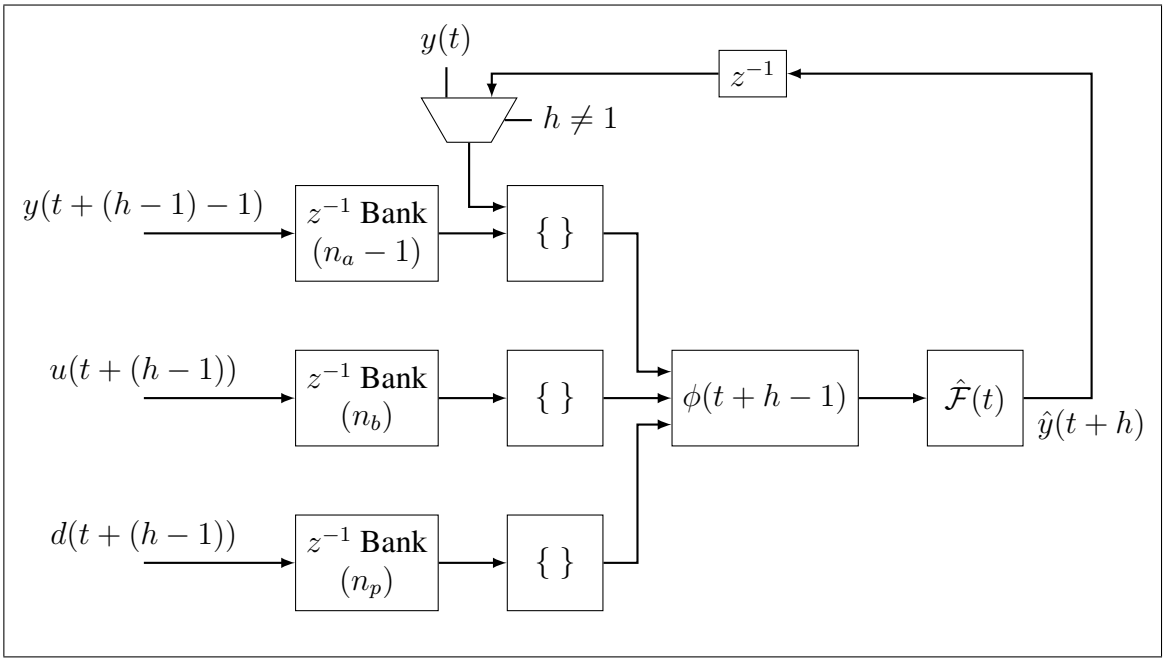


Figure 4.3. Multiple step ahead random forest predictor implementation using the recursive strategy.

At each prediction step $h$, all relevant measurements go through the filter bank and are concatenated as specified in equation 4.18 to generate predictor vector $\phi(t+h-1)$. The prediction $\hat{y}(t+h)$ is then used in $\phi_i$ for the next step which explains the difference in the treatment of $y(t)$ and $u(t), d(t)$ in the diagram. As the objective is to predict the process outputs in an online manner, $\hat{y}(t+h)$ is used as an *estimator of the unknown value* $y(t+h)$.

**Direct or multiple strategy RF$_\mathbf{b}$:** A *bank of predictors* is generated for each instant in the prediction horizon, that is,

$$\hat{y}_i(t+h) = \hat{\mathcal{F}}_{ih}(t)(\phi_i'(t+h-1)),\ h = 1, ..., H$$

$$\phi_i'(t+h-1) = \left\{ \left\{ y_i(t-k) \right\}_{k=0}^{n_a^i-1}, \left\{ \left\{ u_j(t-k+h) \right\}_{k=1}^{n_b^j} \right\}_{j=1}^{m}, \left\{ \left\{ d_l(t-k+h) \right\}_{k=1}^{n_p^l} \right\}_{l=1}^{p} \right\} \tag{4.20}$$

Since the final objective of this strategy is its use in a predictive control design, the $H$ Random Forests will be trained by keeping the lagged output terms static while *forecasting disturbances and inputs*. When testing prediction accuracy and forecasting, this is achieved by feeding the input values to the predictor as $h$ moves along the horizon. Figure 4.4 shows this process for a scaled example.

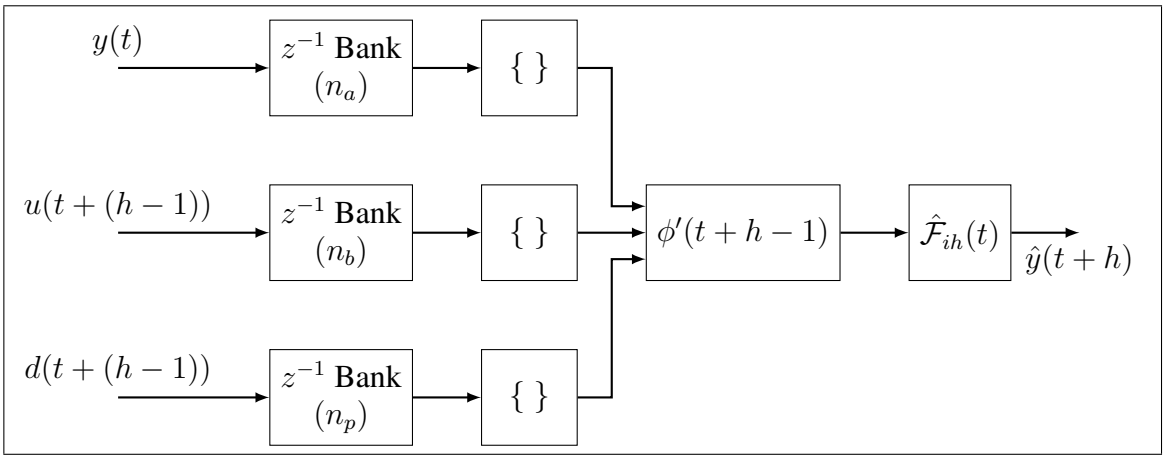


Figure 4.4. Multiple step ahead random forest predictor implementation using the direct or predictor bank strategy.

For $n$ outputs and a prediction horizon of length $H$, and given that outputs are assumed to be uncoupled, $n \times H$ forests will be trained in this framework. It can be seen in figure 4.4 that as $h$ advances, future inputs and disturbances - which have to be forecasted themselves - are used while *only past outputs* form the predictor $\phi'(t+h-1)$.

Since $u(t)$ are the decision variables of the optimization problem, part of the future data is expected to be determined at each time instant. If this strategy proves to be adequate, then building the predictor as stated in equation 4.20 will aid in the implementation of the controller.

For the purpose of this research, disturbances will be forecast with a simple persistence model $\hat{d}(t+h) = d(t)\ \forall t,\ \forall h$. However, filter expressions such as those mentioned in this section could be applied.

### 4.3. Datasets for System Identification

The dataset is split into two categories: the system identification subset and the control subset. The first one corresponds only to the month of August 2017 and is used to fit all models for the purpose of this research. Internally, it is divided into a training portion which hold 85% of August data and the remainder is left for validation and testing. The control subset, on the other hand, corresponds with the first 300 hours of September 2017 operation. A general overview of preprocessing methods and the sequence in which they were applied can be found in appendix C. The dataset used in this research is described below:

**Pseudo-Real Data Set:** Consists of *real operation inputs* applied to the simulator and the *resulting simulated outputs*. This data will be used to train, validate and test the predictive models generated and will ultimately provide the framework in which the controller will be evaluated. Training data is generated without an initial 50 hour transient.

The Distributed Control System (DCS) sampling time is $T_s = 1$s, which corresponds with most of the collected data. However, this sampling rate is much higher than the system's time constants and overall dynamics. Therefore, different sampling times will be considered, which adds a *multi-rate* component to the research.

Table 4.2 specifies these values and the terminology that will be used throughout this document to refer to them.

Table 4.2. Different sampling rates considered in the research.

| **Sampling Rate** | **Symbol** | **Value** | **Unit** |
|---|---|---|---|
| Instrumentation and DCS | $T_s$ | 1 | second |
| Predictive model | $\tau_R$ | 5 | minutes |
| Control action | $\tau_C$ | 5 | minutes |

The following variables are considered from the initial data set:

**Controlled Variables:** Torque $T(t)$, underflow concentration $C_u(t)$ and interface level $h(t)$ were available in historic DCS records. Therefore, they were used accordingly regarding the real operational dataset.

**Disturbance Variables:** Feed rate data $Q_f(t)$ was available in the DCS records. However, feed solid concentration $C_f(t)$ was manually registered, if at all, in an operation spreadsheet. To overcome this issue, hourly data was assumed constant for each hour and missing values were replaced with adjacent data.

**Manipulated Variables:** Underflow rate $Q_u(t)$ was the *only* MV registered directly in the DCS. Flocculant solution and dilution water rate values are needed. This were both included in the original dataset. A feed-forward loop is applied to regulate flocculant addition. This is based on the total solid feed rate on the thickener. Since $C_f(t)$ data had to be assumed constant for an entire hour, flocculant dosage $F(t)$ time series are in result modulated by this artificially slower rate.

### 4.3.1. Thickener Outputs Simulation and Pseudo-Real Dataset

Figure 4.5 shows the input dataset used for system identification. It corresponds with real operational data for the month of August of 2017 and is divided in the training (blue) and validation and testing (red) subsets.

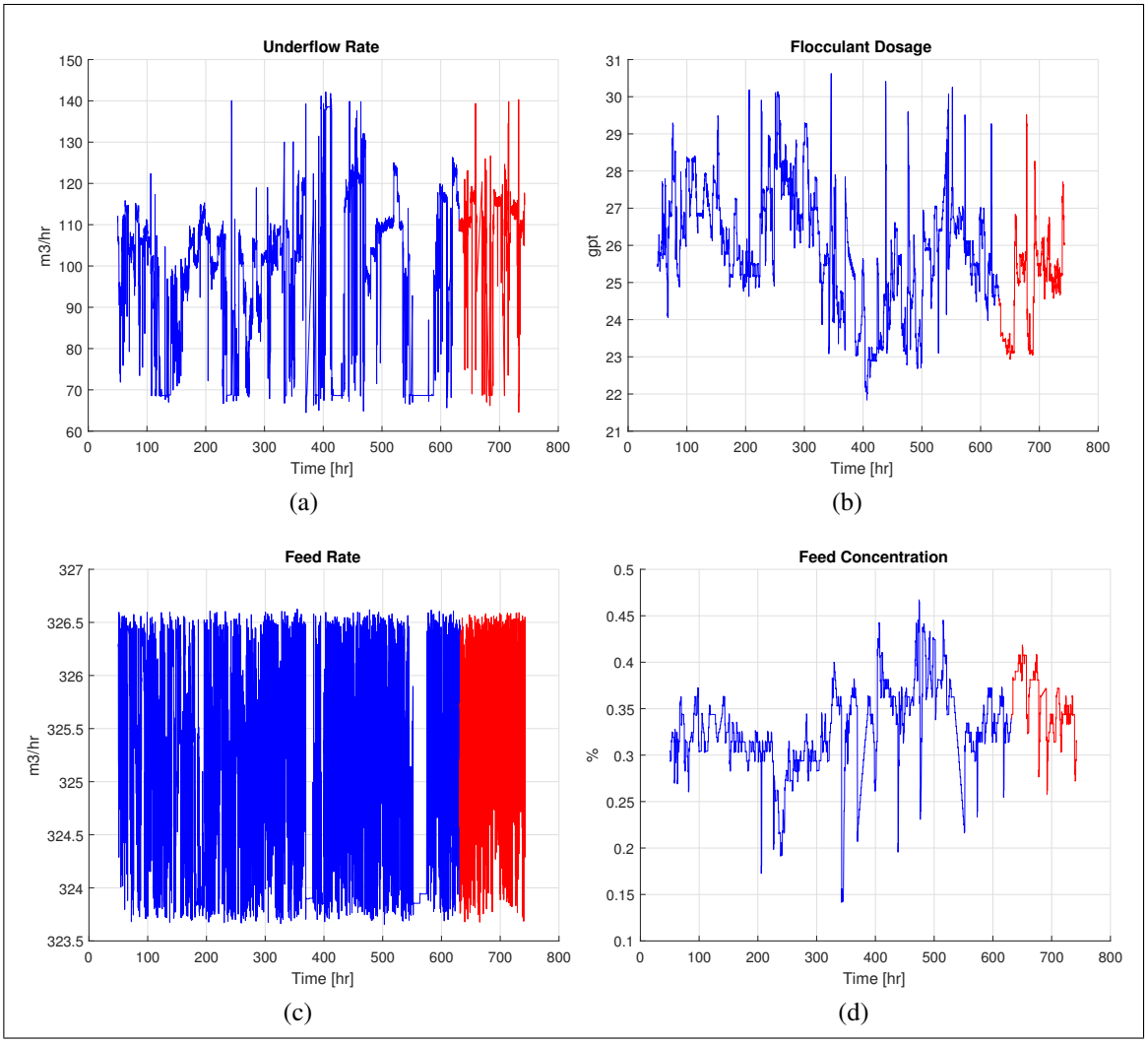


Figure 4.5. Input real dataset used for model training and validation. Corresponds with real operational data for the month of August of 2017. Figures 4.5a and 4.5b correspond to the MVs, while figures 4.5c and 4.5d are the DVs. Training subset is depicted in blue while the section in red is the validation and testing set.

Figure 4.5d shows the slower sampling rate associated with the feed solids concentration provided by manual logging of the variable. Even though feed flow rate can appear to vary rapidly in figure 4.5c, its range is small. This is a result of the noise present in this variable and the data processing used to recover it.

Input data was applied to the initial empty thickener, as specified previously. To emulate a real system identification scenario, noise of 20 dB of signal to noise ratio was added all outputs. Band limited noise was used, since in practice physical systems are never disturbed by white noise. The correlation time specified was 60 seconds. The results of these simulations, which will be used for system identification, are shown in figure 4.6.

Figures 4.6b and 4.6a illustrate an importat relationship between $T(t)$ and $C_u(t)$. They are tightly bound together, which justifies the hypothesis of controlling torque through the underflow solids concentration (Tan et al., 2017). On the other hand, figure 4.6c shows that for August operational data the slurry overflows, which can be identified as the region in which interface level reaches the feedwell.

### 4.4. Benchmark Problem for Algorithm Validation

To test the implementation and algorithms designed for random forest modeling, a smaller and manageable problem was used. The system corresponds to a steam boiler simulator and consitues a benchmark problem in both system identification and controller design (Pellegrinetti & Bentsman, 1996).

As a MIMO nonlinear system, it shares many of the properties that affect thickener modeling. However, as accurate description of the input-output effects exist, the boiler provided an ideal framework to for algorithm design, implementation and debugging. Additionally, it provided key insights to assess if the proposed strategy was to be successful when applied to the thickener problem. The results of these tests can be found in appendix D.1.

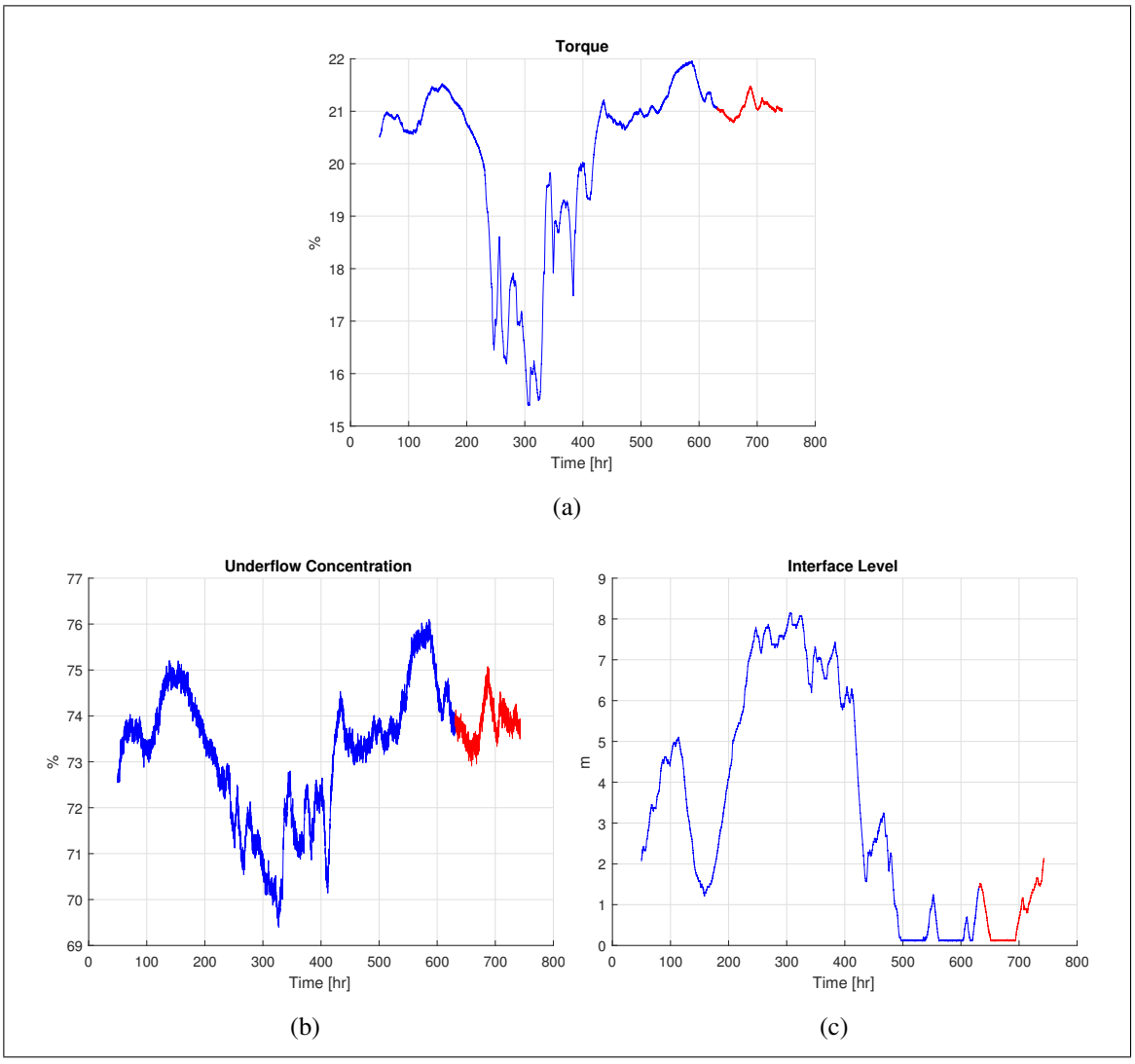


Figure 4.6. Controlled variables used for system identification. Training subset is depicted in blue while the section in red is the validation and testing set. Noise has been added to the simulated outputs to emulate a real system identification scenario.

## 4.5. System Identification Results

Given all the theoretical background for the use of random forests in time series prediction, the next sections deal with its application to the thickener modeling problem, model results and discussion of predictive performance.

### 4.5.1. Random Forests Predictor for Paste Thickener Operation

As explained in previous sections, both the ARIMAX and random forest predictors are fully specified by selecting their structural parameters. Since two variants of random forest predictors are studied, the recursive strategy will be adjusted first. For the multiple random forests strategy, the best forest will be replicated for each prediction step.

The aim will be to test all predictors both in short and long horizons. The maximum forecast into the future that will be examined corresponds to four hours, which add up to 48 prediction steps of $\tau_R = 5$ minutes.

*For each output* of the linear regressor the parameters are $(n_a, n_b, n_p, n_c)$. Additionally, different input delays $L$ can be considered for each transfer function. The initial state for the ARIMAX predictor was backcasted.

For the random forest predictor, a similar situation is attained. By specifying the tuple $(n_a^i, n_b^i, n_p^i)$ for each $i$ output variable the order of the model is specified. Random Forests have three main additional parameters to tune their performance: number of learners in the ensemble $B$, minimum leaf size $l_{\text{min}}$ and number of predictor variables for each split $m$. For each Random Forest model, the total number of predictor variables is, according to equation 4.18,

$$\dim(\phi_i) = n_a^i + \sum_{k=1}^{m} n_b^i(k) + \sum_{k=1}^{w} n_w^i(k) \tag{4.21}$$

The question becomes now specifying the values of these hyperparameters. A simple greedy heuristic will be followed. Certain physical boundaries, inherent to the nature of the process, will be considered.

Residence time in the thickener is of the utmost importance in paste production. However, it is a highly non-linear function, which varies drastically with particle size, rake operation and solid inventory. Nevertheless, its upper bound is estimated at 6 hours (Langlois, 2018). Therefore, all model orders $n_a, n_b, n_w, n_c$ will be inside this range.

An important fact about the predictive model generation proposed is that *no delay coefficient* will be imposed on Random Forests. Throughout experimental phases of this research, convincing evidence was found that this constraint worsened Random Forest performance. Instead, the hypothesis was that by specifying high-order models the Machine Learning algorithm would be able to filter unimportant information.

Some memory and computational restrictions appeared when generating these models These limitations where specially significant for the order of the unmeasured disturbance: as a consequence, a reduced order had to be considered for this coefficient. RAM usage took up *32 GB* when trying to fit larger models, which is probably a problem with the implementation of the identification algorithms.

Table 4.3 specifies the different values tried for all hyperparameters, both for the Random Forest and ARIMAX predictors. The heuristic approach, which is a classical method in both system identification and Machine Learning, will be to test all possible combinations of the parameters specified in 4.3. All order coefficients will be applied to the $n = 3$ outputs.

Table 4.3. Different values for structural parameters tried in model selection. Coefficients are specified as multiples of $\tau_R = 5$ minutes.

| **Parameter** | **Random Forest** | **ARIMAX** |
|---|---|---|
| $n_a$ | $[0, 6, 12, 30, 60, 72]$ | $[0, 6, 12, 30, 60, 72]$ |
| $n_b$ | $[0, 6, 12, 30, 60, 72]$ | $[0, 6, 12, 30, 60, 72]$ |
| $n_p$ | $[0, 6, 12, 30, 60, 72]$ | $[0, 6, 12, 30, 60, 72]$ |
| $n_c$ | - | $[0, 6, 12, 30]$ |
| Delay $L$ | - | $[0, 6, 12, 30, 60, 72]$ |
| $B$ | $[0, 10, 30, 50, 100, 300]$ | - |
| $l_{\text{min}}$ | $[5, 10, 20, 50]$ | - |

As usual, the mean squared error (**MSE**) of the predictions will be accounted for:

$$J_{\text{MSE}} = \frac{1}{N} \sum_{k=1}^{N} \left( y(k) - \hat{y}(k) \right)^2 \tag{4.22}$$

and also the *best fit rate* (**BFR**):

$$J_{\text{BFR}} = (1 - J_{\text{RNMSE}})\,100 = \left(1 - \sqrt{\frac{J_{\text{MSE}}}{\text{Var}\,\{y(k)\}}}\right) 100 \tag{4.23}$$

Model selection, that is, using the different indicators to choose the model that best fits the data, will be done based on $J_{\text{MSE}}$. This is a natural choice because both random forests and ARIMAX models are trained based on minimization of a squared error loss.

### 4.5.2. Analysis of Predictive Models Obtained

After iterating through all combinations specified in table 4.3, the parameters in table 4.4 were found to be the best.

Table 4.4. Best hyperparameter combination for thickener model identification. Coefficients are specified as multiples of $\tau_R = 5$ minutes.

| **Parameter** | **Random Forest** | **ARIMAX** |
|---|---|---|
| $n_a$ | 72 | 60 |
| $n_b$ | 72 | 30 |
| $n_p$ | 72 | 30 |
| $n_c$ | - | 6 |
| Delay $L$ | - | 0 |
| $B$ | 300 | - |
| $l_{\text{min}}$ | 10 | - |

Table 4.4 shows that both strategies identify highly auto-regressive models. Similarly, the order of the inputs is also very high. This amounts to a random forest predictor $\phi_i$ of size 360 according to equation 4.21.

An interesting result shown in table 4.4 is that the best ARIMAX model fit has zero delay. Presumably the size of the coefficient $n_b$ has an impact in this: since a lot of past samples of the input are being considered for the model the delay is absorbed in the $B(z^{-1})$ polynomial. Figure 4.7 shows the pole-zero plot of each pair of transfer functions identified.

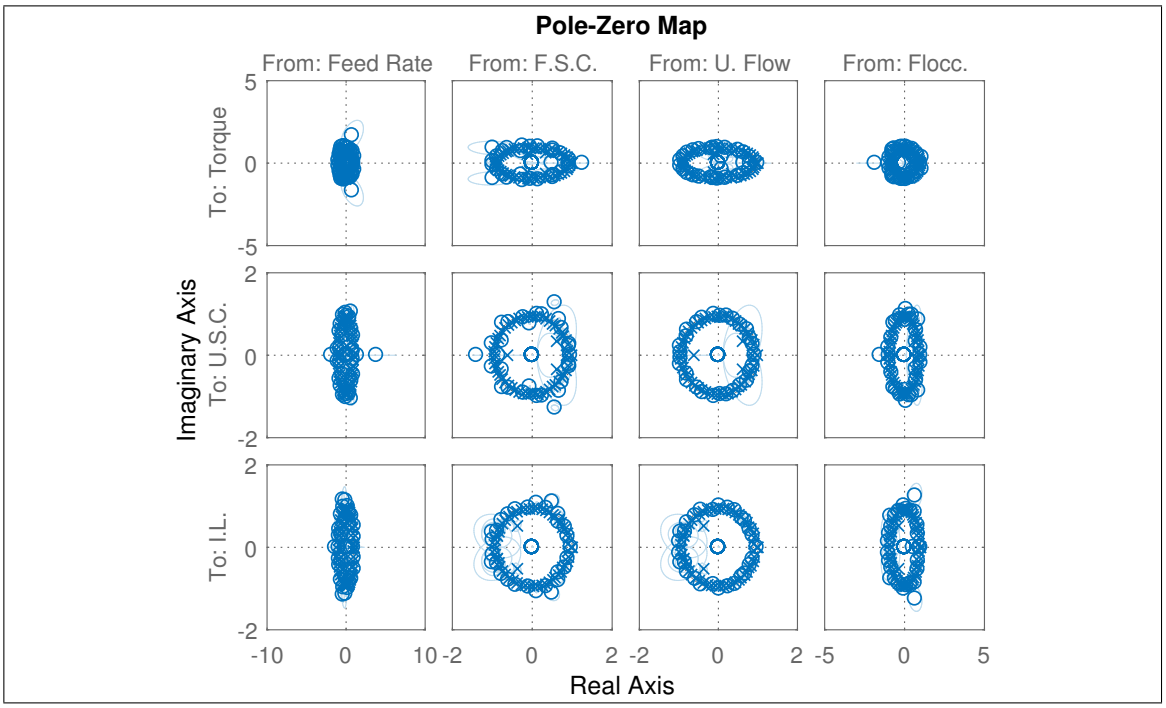


Figure 4.7. Pole-zero map of the identified ARIMAX transfer functions. The grey circles show the confidence interval for the identified pole or zero.

All poles identified fall under the unit circle and thus every transfer function pair is stable, which is a reasonable result given the fact that input-output data is under feedback. However, the confidence region for some poles and zeros is extremely large. This could impact the stability and overall accuracy of the ARIMAX models. Also, some poles and zeros are very close to each other and therefore are virtually canceled. However, the heuristic used to find the model deemed these low complexity models less accurate.

Nevertheless, first and second order transfer functions where identified for the data for the sake of simplicity and its possible use in MPC strategies. Results showed similar performance for low frequency inputs but substantial differences elsewhere. Figures D.5a and D.5b in the appendix show the bode plot of the identified models that support this conclusions.

Increasing the number of learners in random forests *cannot overfit the data* (Hastie et al., 2009). Trees inside the ensemble can if they are too deep. Table 4.4 shows that the

best outcome came from deep trees - given by the leaf minimum size $l_{\min}$ - and a large ensemble of 300 trees.

Deeper analysis of this situation is given in figure 4.8. OOB cumulative error and predictor importance criteria are shown for the underflow solids concentration random forest. The plots for torque and interface level can be found in figure D.6 in the appendix.

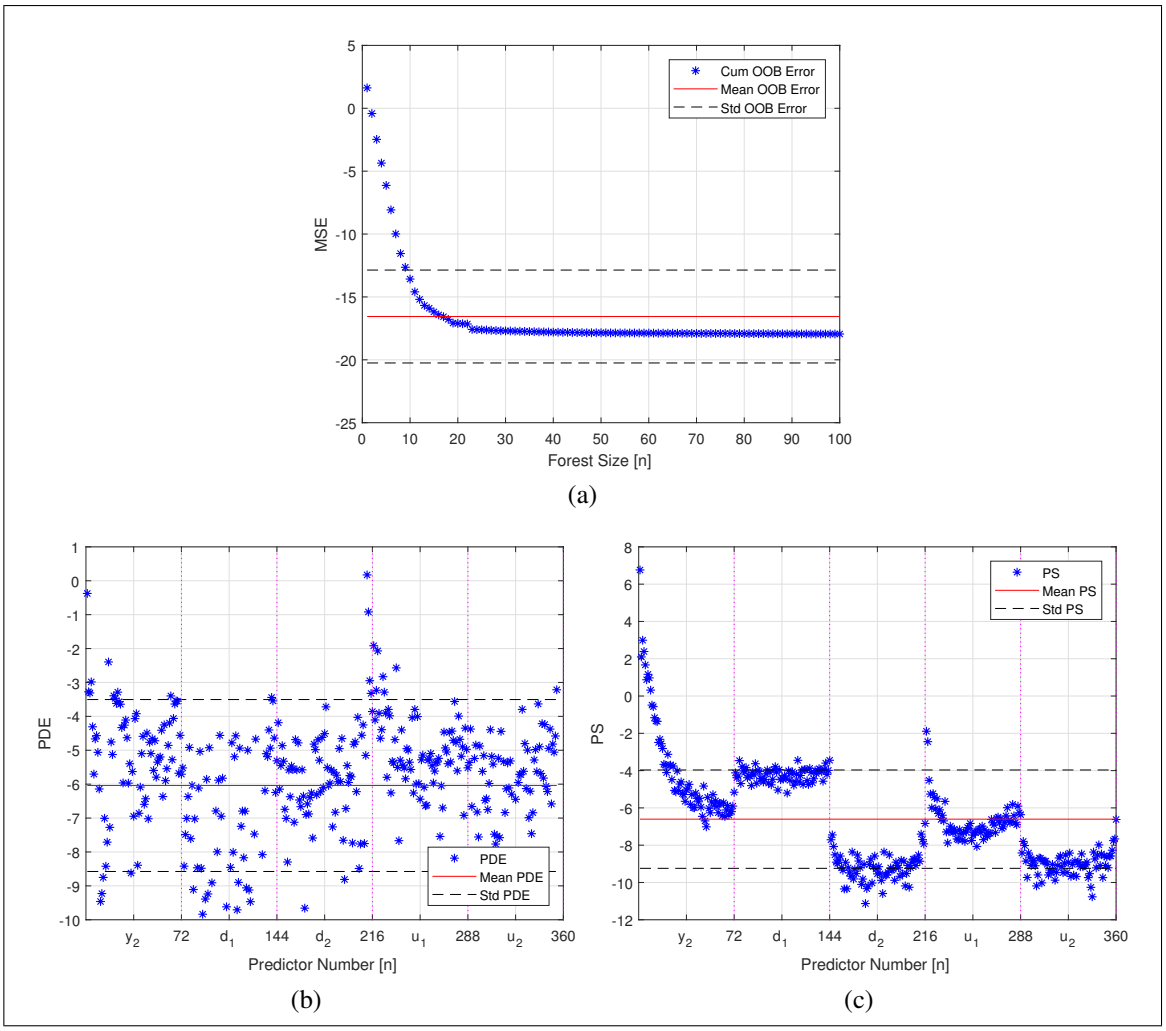


Figure 4.8. OOB error, permuted predictor delta error and number of predictor splits for $C_u(t)$ random forest with regressive strategy. The vertical purple lines group the lagged terms of each variable in the predictor $\phi_i$.

Cumulative MSE in figure 4.8a shows that from 25 to 30 trees the MSE reduction stalls and is well below the mean of the forest. Therefore, *smaller ensembles* can be trained with no significant reduction in performance. For the multiple forest strategy ensembles of 50 trees will be trained while the recursive ensembles will be reduced to 100 learners.

Figures 4.8b and 4.8c show the importance of each predictor in $\phi_i$ used by the ensemble in logarithmic units. The PDE plot gives a measure on how important a predictor is for the whole ensemble; as such, it can be seen that special importance is given to past outputs, specially recent ones, as well as *very old samples of the feed solids concentration*. This justifies using high order coefficients for $n_p$ since it is the only way to capture these dynamics.

Similarly, it can be seen that the forest regards underflow rate as an extremely important variable in predicting underflow solids content. Since the objective of system identification in this context is generating a *controllable* model, this is of the utmost importance. Furthermore, the model also regards feed rate as an unimportant variable in the predictor; this is because the dataset used has an almost constant feed rate as explained in section 4.3.

Contrasting this with this figure 4.8c gives further insight into how the forest fits the data. The plot shows that lagged terms of the output were used as split variables *the most*. Together with the PDE explained previously, this means that a large number of splits in the top of the tree correspond to auto-regressive terms. A similar analysis can be made for underflow rate, as well as the oldest samples of feed concentration.

On the contrary, since feed rate is deemed unimportant and the plot shows a relatively large amount of splits for this variable, it is used at the bottom of the trees. It can be argued that feed rate should be eliminated from the dataset; however, the small variance of this variable in the data set is a result of faulty sensor measurement and its processing. Hence, in theory this variable should be included and therefore it will be used in this research because the forest predictor is able to screen through it.

Finally, table 4.5 compares some forest statistics regarding the ensembles trained for the recursive and multiple strategy, which use the 360 predictors and 50 learners. Since the latter strategy uses 48 different forests, one for each step, the last ensemble will be examined. As each forest is composed of different trees, the results shown average the statistics among all trees in the forest.

Table 4.5. Comparison of random forest structure between the recursive and multiple strategy for each output.

| **Statistic** | **Number of Nodes** | **Number of Branches** | **Node Size Range** |
|---|---|---|---|
| $T(t)$ $RF_r$ | 466.16 | 232.58 | [390.32 , 63.02] |
| $T(t)$ $RF_b$ | 699.08 | 349.04 | [496.54 , 81.00] |
| $C_u(t)$ $RF_r$ | 1168.73 | 583.87 | [698.43 , 22.62] |
| $C_u(t)$ $RF_b$ | 1163.74 | 581.36 | [935.32 , 28.16] |
| $h(t)$ $RF_r$ | 358.6 | 178.80 | [1020.35 , 36.82] |
| $h(t)$ $RF_b$ | 648.28 | 323.64 | [1263.81 , 37.03] |

The difference between nodes and branches is that the latter do not count leafs or terminal nodes. The ratio between the two indicates how wide or tall a forest is; if both numbers are similar then the forest is relatively flat, which translates to a highly nonlinear underlying model.

As forests vary in sizes, a tree node size range will be defined as the sizes of the nodes - number of samples under it - in the first and last 10% of it. For example, for a tree with 100 nodes, its node size range shows the size of the node 10 and 90, from top to bottom. This is averaged across all trees in the ensemble, each with its own tree size.

Some similarities can be appreciated between both strategies. For example, the size of the underflow solids forests is almost identical for both strategies. As such, it is expected that both strategies produce similar results. However, the upper node size range is 33% larger for the multiple ensemble strategy. The impact of this is that its *variance* can be expected to be larger as the prediction steps advance.

Important differences can be seen in the torque and interface level forests. The multiple strategy generates forests that are 55% and 80% larger and whose node size range is also larger. Therefore, it is deciding to make more splits with most of the samples concentrated at the top. This can be interpreted as a signal of a high variance and low bias predictor.

However, the most important insight extracted from table 4.5 is that the models identified with both strategies *are not highly nonlinear*. For all models, the node to branch ratio is close to two, which means that the forest predictor *groups* a lot of samples at the top of the trees rather than making a very granular approach.

To fully confirm all these conclusions, the predictions made by the models will be explored in depth in the following section.

### 4.5.3. Forecasting and Predictive Performance of Predictive Models

Figure 4.9 compares the forecast made by the the three models identified previously for two random time windows of four hours in the validation set. For torque and underflow solids content, the ARIMAX predictor identifies well the shape of the validation data but has important offset errors. The interface level forecast from figure 4.9e, however, is very poorly identified by the linear model.

Both random forest strategies produce opposite results. The recursive strategy shows a smooth but very *flat* curve for the four hour forecast. On the other hand, the multiple forest strategy has high variance - relative to the other strategies - but tends to understand slow dynamics better. For almost all figures, the contour of the $RF_b$ seems to follow the validation data.

However, it is important to look at the scale of the predicted outputs. In the case of torque and underflow solids, no model incurs in more than 0.3% of error. As such, it can be expected that in a *purely predictive* strategy - that is, without a control objective in mind - all models should behave somewhat accurately.


Figure 4.9. Comparison between the different trained predictors in a four hour forecast for two separate time windows. Figures 4.9a, 4.9c and 4.9e to hour 30 of the test dataset, while 4.9b, 4.9d and 4.9f to hour 76.

Figure 4.10 shows the one step ahead prediction for underflow solids and interface level. All models fit the data accurately. Even in the case of figure 4.10a, which is the noisiest output of the three studied in this research, both curves can estimate the value of $C_u(t)$ exactly. The high variance of $RF_b$ identified previously can be seen.

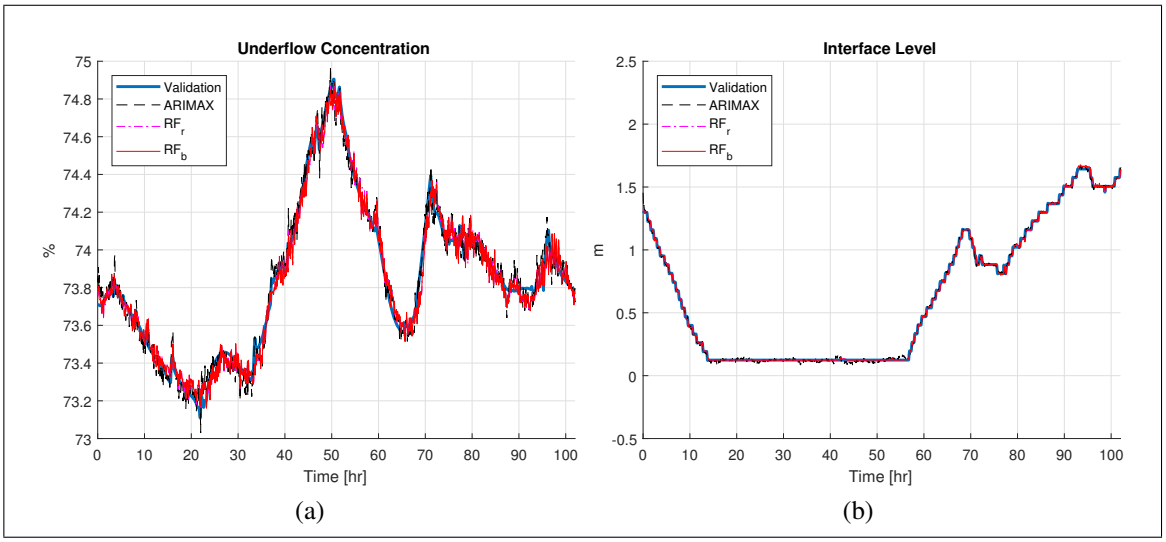


Figure 4.10. Underflow solids concentration and interface level prediction comparison between the different predictors for a five minute ahead prediction.

However, as mentioned in the section 4.5.1, the one step prediction error of a model is not always a good indicator of accuracy (Hastie et al., 2009). No strong conclusion can be drawn from these results without examining a longer prediction horizon.

Figure 4.11 compares $RF_r$ and $RF_b$ results for a two and four hour ahead prediction in underflow solids content. An important issue can be seen in the $RF_r$ strategy: the prediction generated is a *delayed version of the validation data*. This delay corresponds exactly to the prediction horizon specified. If the random forest predictions of figure 4.11 where shifted 23 and 47 steps back in time, one step ahead predictions would be recovered. The results for torque and interface level can be found in the appendix in figure D.7.

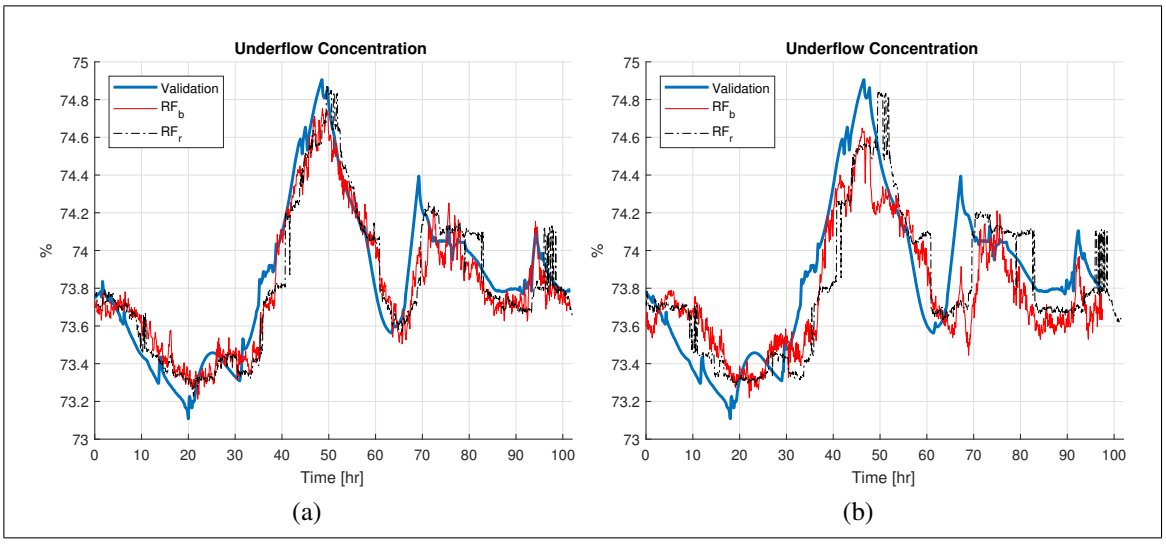


Figure 4.11. Comparison between the recursive and multiple random forest strategies of two (figure 4.11a) and four (figure 4.11b) ahead prediction for $C_u(t)$.

In other words, this predictor is in fact very similar to the persistence model $\hat{y}(t) = y(t-1)$. The prediction model is effectively *holding the last known value throughout the whole prediction horizon*, which explains the flat forecast curves in figure 4.9. In a forecasting situation - such as the online MPC problem - this is a highly undesirable property specially because *the influence of exogenous input is reduced*.

Figure 4.12 compares the results of the linear ARIMAX model to the $\text{RF}_\text{b}$ for a two and four hour ahead prediction for all controlled outputs. In the case of the ARIMAX model, an important increase of variance can be encountered. Such effect can be attributed to the large AR order of the model. A high order auto-regressive model has many poles associated with it. Estimation of these poles becomes more difficult because of their increased sensitivity to noise in the data.

The $\text{RF}_\text{b}$ strategy does not suffer, in general, from the the notorious delay effect of the recursive strategy. Furthermore, it is clear that for such long horizons the variance of the ARIMAX model increases substantially. Random forest predictors also have an increase in variance, specially regarding interface level as shown in figures 4.12e and 4.12f.

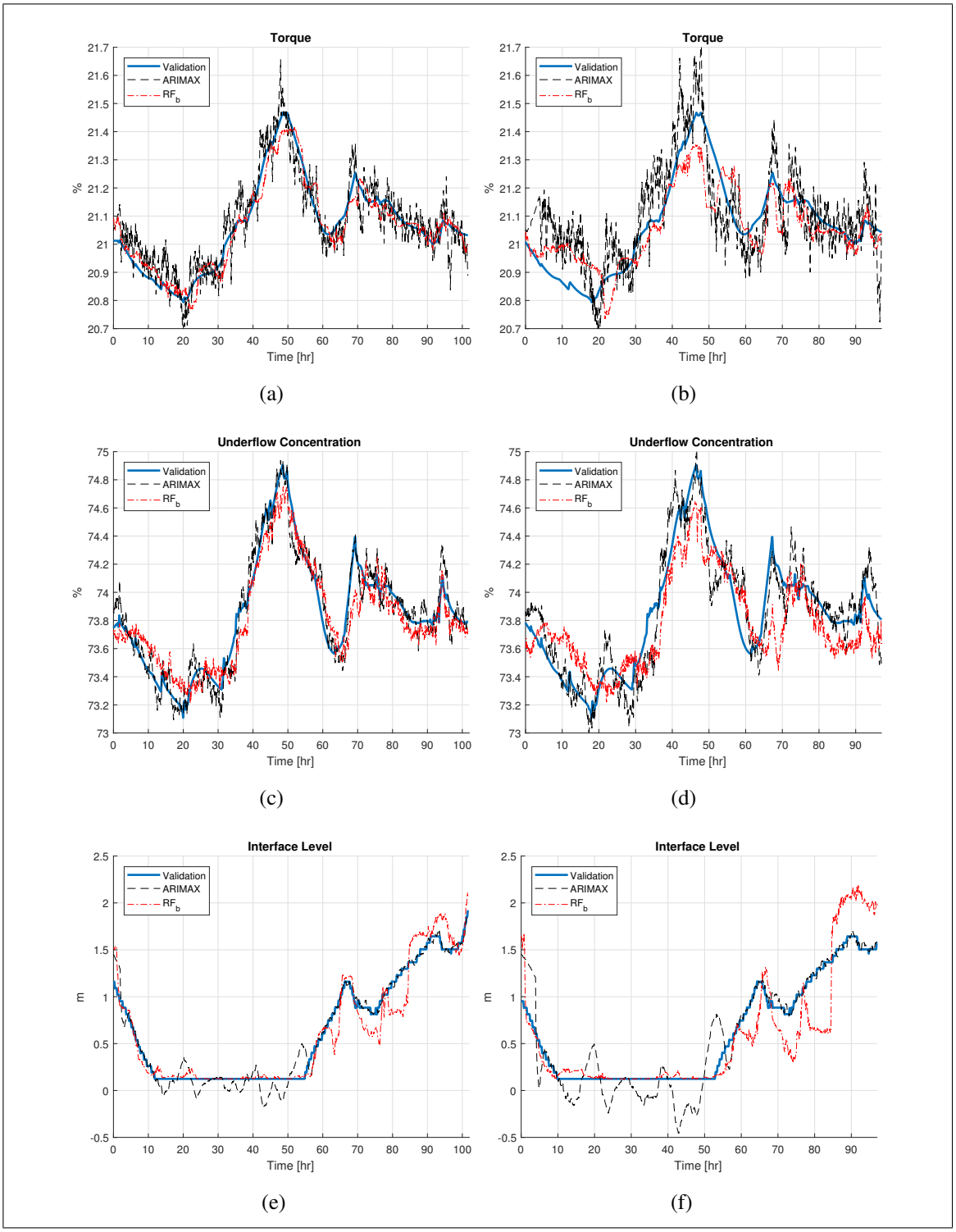


Figure 4.12. Comparison between ARIMAX and multiple random forests in two (figures 4.12a, 4.12c and 4.12e) and four (figures 4.12b, 4.12d and 4.12f) hour ahead predictions for all controlled variables.

An important aspect of the results collected in figures 4.12 is that by far the most difficult variable to predict is underflow solids concentration. For a long forecast, all predictors fail to capture these dynamics accurately. Table 4.6 compares the best fit rate for various prediction horizons and summarizes all the results shown in this section. A table with MSE values can be found in appendix D.

Table 4.6. Fit rate of best random forests and ARIMAX models for all controlled variables.

| **Variable** | **BFR** | | | |
|---|---|---|---|---|
| **Time Ahead [min]** | 5 | 90 | 120 | 240 |
| $T(t)$ ARIMAX | 90.05 | 67.01 | 57.36 | 16.34 |
| $T(t)$ $RF_r$ | 91.03 | 75.57 | 69.59 | 47.86 |
| $T(t)$ $RF_b$ | 90.94 | 78.39 | 74.17 | 44.28 |
| $C_u(t)$ ARIMAX | 68.72 | 62.19 | 60.19 | 50.26 |
| $C_u(t)$ $RF_r$ | 69.10 | 57.84 | 52.89 | 35.46 |
| $C_u(t)$ $RF_b$ | 69.00 | 59.91 | 55.10 | 35.62 |
| $h(t)$ ARIMAX | 96.95 | 86.88 | 82.81 | 66.19 |
| $h(t)$ $RF_r$ | 96.78 | 85.99 | 81.92 | 65.99 |
| $h(t)$ $RF_b$ | 96.84 | 83.53 | 73.78 | 49.40 |

As mentioned above, *all* models fail to characterize underflow solids concentration accurately even for a one step ahead prediction. As the horizon gets larger, the ARIMAX model predicts this variable better while both forests' performance is seriously degraded.

Regarding torque, it is clear that the ARIMAX model identification fails to capture its dynamics accurately, while the random forest approaches do so up to two hours into the horizon. Interface level exhibits a much less definitive outcome, as the ARIMAX model is only slightly better and therefore statistically insignificant.

Nevertheless, the linear model has important difficulties in capturing the effect of rapidly decreasing or low interface levels as shown in figures 4.12e and 4.12f. From hour 0 to around hour 60, the variance of this model is significantly higher than that of both random forest strategies.

This is explained by two main reasons. Firstly, that portion corresponds to a large deviation from the operating point and hence linear approximations become questionable. Most importantly, however, the $n_b, n_p$ coefficients - that translate to the input-output effect - are all of order $10^{-2}$ or smaller, while the auto-regressive coefficients are all larger than that amount. Therefore, the model identified is somewhat *unresponsive* to external inputs.

However, the most important conclusion drawn from table 4.6 is that for a *90 minute prediction horizon* the predictor performance for all models is still at an acceptable level. Comparing the one step and 90 minute ahead predictions, the largest decrease is given by the linear torque model with a 23% decrease in best fit ratio, while forest predictors face 15% and 12% respectively.

### 4.5.4. Final Remarks on Random Forest Predictors

The results from this chapter state clearly that none of the methods chosen can guarantee a completely trustworthy dynamic characterization of the thickener studied. The hypothesis that the simulator represents a highly nonlinear system can now be questioned: either it is not or the input data - which is afflicted by the issues explained in section 4.1.2 - cannot explore a nonlinear range for the plant. The latter explanation is more accurate, as thickener studies and modeling propose a nonlinear physical description.

The main issue in thickener prediction and control is its *highly auto-regressive* behaviour. For any model to fit the data accurately - whether it is linear or nonlinear - various lagged output terms have to be considered. The effect of all inputs are much lower than the auto-regressive part. This is a challenging control scenario for a predictive controller because the online forecast will tend to be flat regardless of the input sequence determined.

All the results examined suggest that the random forest strategy pursued in this investigation is *not suitable* for time series prediction or at least in its application on thickener operation.

The main issue that random forests seem to be unable to tackle is the auto-regressive behaviour. As the predictor vector $\phi_i$ contains samples which are very similar to the current output, it is biased towards them and looses sight of input effects. Forest analysis provided in section 4.5.2 support this claim. The choice of random forests for thickener modeling was inadequate: a highly nonlinear model was chosen for a system which did not reflect such characteristics but exposed the main weakness of the method.

Another aspect of this results that can be questioned is the choice of the prediction step or sampling time $\tau_R$. The results show that it is too short for a good characterization of the thickener and has a negative effect on all predictors. Better results can be obtained for larger prediction steps. However, this constraint is rooted on preliminar analysis of thickener control. Using very large steps - for example, half or whole hours - is believed to largely miss important dynamics of the system.

Nevertheless, since the objective of this research is generating a model for *control* purposes, random forests will still be used for the formulation of a predictive controller.

While the multiple forests take up over *6 GB of memory*, the recursive strategy uses approximately 25 times less disk space. Since for short horizons both predictors behave in an almost identical manner, the *recursive strategy* will be preferred. This decision is also supported by physical evidence of the thickening process, its modeling and all literature regarding the physical principles that govern thickeners.

# 5. DATA-DRIVEN PREDICTIVE CONTROL

The main objective of this research is exploring *data-driven* control. This chapter is centered around the formulation, theoretical background, design and implementation of the Random Forest Model Predictive Controller.

Section 5.1 offers the general formulation of the RF-MPC. It is centered on the mathematical background of the three parts that compose a predictive controller, explained in section 5.1.1. The optimization algorithm used to compute the optimal control input sequence is described in section 5.1.2. Finally, some issues regarding stability results are introduced in section 5.1.3.

The chapter ends with the final implementation of the controller in section 5.2. Special emphasis is given to computational and efficiency issues in section 5.2.1 since the combination of random forests and predictive control poses important difficulties on the predictive controller.

## 5.1. Random Forest Model Predictive Control

The toolbox designed in this research, as well as its mathematical formulation, is built for control problems of any dimension. Even though the task at hand - thickener control - is already defined in terms of dimensions and objectives, as specified in section 2.2.3, the formulation for the RF-MPC will be given in a general manner. Its application to the thickener simulator will be explained in chapter 6.

### 5.1.1. Elements of the Random Forest Model Predictive Controller

As any predictive controller, the proposed RF-MPC follows the same structure specified in section 3.2.1. All the prediction models and objective functions are described in terms of $\tau_R$. For the sake of simplicity, this variable will be dropped unless otherwise specified.

#### 5.1.1.1. Predictive Model

Naturally, the predictive models used are the regressive random forests identified in chapter 4. However, as explained in section 3.2.1, the *predictive model* becomes a constraint of the problem.

For the random forest prediction models trained, the constraints become the nonlinear relationships between MVs and CVs defined through the forest predictor:

$$\hat{y}_i(t+j) = \hat{\mathcal{F}}_i\left(\phi_i(t+j-1)\right), \quad j = 1, ..., N_y, \quad i = 1, ..., n \tag{5.1}$$

which amounts to $n \times N_y$ additional constraints.

It is important to recall, however, that as $j$ advances through the prediction horizon, the predictor $\phi_i$ starts to include previously predicted outputs. For the case of the disturbance forecasting during the prediction horizon, the persistence model $\hat{d}_i(t) = d(t-1)$ was used.

#### 5.1.1.2. Objective Function

The RF-MPC strategy proposed in this research is for the MIMO problem stated in chapter 2. For the purposes of this research the objective function can be constructed as:

$$\min_{y,u} \; V(y,u) = \sum_{i=1}^{n} V_i(y_i, u) \tag{5.2}$$

where each $V_i(y_i, u)$ is:

$$V_i(y_i,u) = \sum_{j=1}^{N_y-1} \frac{\left(\hat{e}_i(t+j)\right)^T Q_i \left(\hat{e}_i(t+j)\right)}{l_i} + \frac{1}{n}\sum_{k=1}^{m}\sum_{j=0}^{N_u-1} \frac{\Delta u_k(t+j)^T R_k \Delta u_k(t+j)}{s_k} + \sum_{j=1}^{N_y} \frac{\epsilon_{ij}^T \Lambda_i \epsilon_{ij}}{l_i} + \beta_i^T \left|\hat{y}_i(t+N_y) - y_{i_{ss}}\right|^2 \tag{5.3}$$

where

- $\hat{e}_i(t+j)$ is the predicted error ($\hat{y}_i(t+j) - w_i(t+j)$).
- $w$ is the reference signal for the horizon $N_y$.
- $y_{i_{ss}}$ is the steady-state target and the relationship $y_{i_{ss}} = w_i(t+N_y)$ holds.
- $|\hat{y}_i(t+N_y) - y_{i_{ss}}|^2$ is the terminal cost.
- $\epsilon_{ij}$ is a binary *on-off* variable. If predictions assert that a constraint will be violated, the value of this variable is one. This is a typical technique for constraint softening, which promotes that the problem remains feasible (Grune, 2017).
- $Q_i, R_k, \beta_i, \Lambda_i \geq 0$ are the weights for the different costs in the objective function. It is worth noting that these coefficients can be chosen to be time dependent, i.e., $Q_{ij}$. However, they were left static for simplicity.
- $l_i$ and $s_k$ are normalization coefficients.
- As the manipulated variable cost appears in each $V_i(y_i, u)$ and these terms are summed in expression 5.2, it is divided by $n$.

The formulation in expression 5.3 uses incremental MV values $\Delta u_k(t)$ which is similar to most predictive control applications (Camacho & Bordons, 2007).

It is important to note that in expression 5.3 no steady-state target value for the MVs is specified, and consequently, no deviation from this setpoint penalized. An important objective of the controller being designed is determining if it is capable of understanding the long term dynamics of the control problem. As such, a steady-state value for the MVs is expected once the error in the outputs stabilizes.

#### 5.1.1.3. Constraints

Aside from the prediction model constraints, both controlled and manipulated variables are subject to multiple restrictions. These involve process and actuator limits as well

as limits on the rate of change of variables. With the help of thickener operation manuals as well as the literature regarding thickener control, the following constraints were included in MPC design:

**Controlled variables constraints:** For each controlled variable, lower and higher limits were established:

$$\underline{Y}_i \leq y_i(t) \leq \overline{Y}_i \ \ \forall t, \ \forall i = 1, ..., n \tag{5.4}$$

However, these constraints will be *softened* through their inclusion in the objective function as expressed in equation 5.3. Hence, hard feasibility issues can be avoided and, for appropriate values of $\Lambda_i$, constraint satisfaction can still be attained.

**Manipulated variables constraints:** Actuator limits were established. Also, rate constraints where considered for the MVs.

$$\begin{aligned} \underline{U}_k \leq u_k(t) \leq \overline{U}_k \ \ \forall k = 1, ..., m \ \forall t \\ |\Delta u_k(t)| \leq \delta_{u_k} \ \ \forall k = 1, ..., m \ \forall t \end{aligned} \tag{5.5}$$

#### 5.1.1.4. Conciliation of Predictive Models and Optimization

An issue that must be addressed is that the predictors $\phi_i\left(t + (j-1)\right)$ contain lagged values of $u\left(t + (j-1)\right)$. Transforming these values to their incremental form through:

$$u_k\left(t + j\right) - u_k\left(t + (j-1)\right) = \Delta u_k\left(t + j\right) \tag{5.6}$$

solves this problem. To comply with actuator limits expressed in the inequalities 5.5, once the optimal MV sequence has been calculated in terms of the incremental values, the controller output will be saturated if necessary.

It is clear that, in general, a random forest is not a linear system in the sense described in section 4.1.2. Even though bagging is a linear method, the weak learners used are

not. Moreover, *the derivative of a regression tree* is not defined. A pseudo-derivative approach can be taken to define a *local* representation *locally*, but it cannot be extended to all the search space. Therefore, constraints in expression 5.1 are both *nonlinear and non-differentiable* and as such represent an important problem to overcome in optimization.

One of two strategies must be followed. The first one involves keeping the model constraints in equation 5.1 so that the objective function is treatable. This approach requires, in general, defining multiple possible trajectories of the system and then searching for the MV values that satisfy those trajectories *and* the prediction constraints.

However, the most used strategy is *including the predicted values as a function of the manipulated variables* in the objective function. This is equivalent to substituting the $n \times N_y$ expressions in equation 5.1 directly in to the objective function and eliminating them from the search space. Hence, only MVs are the decision variables:

$$\min_{y,u} V(y,u) \rightarrow \min_{u} V(u) \tag{5.7}$$

The price to pay for this conversion is now that the objective function is *nonlinear and nondifferentiable*. The parameter space, however, is *convex* since it only includes constraints such as the ones expressed in 5.5. To overcome this issue, *evolutionary algorithms* are one of the most popular and widely studied methods.

### 5.1.2. Evolutionary Algorithms

Evolutionary Algorithms have been used in the scientific community for around three decades. Initially based on biological systems, these methods sample the search space and iteratively converge to the best solution. As such, these methods *do no guarantee convergence* of the optimization method and are *local optimization methods*.

Similarly to hill climbing or simulated annealing algorithms, evolutionary algorithms can get trapped in the basin of attraction of local minimae and therefore may find only

local optimal solutions (Branke, Kaubler, & Schmeck, 2001). Some examples of these algorithms are ant colony optimization, bee algorithm and bat algorithm among others (Marini & Walczak, 2015).

#### 5.1.2.1. Particle Swarm Optimization

In Particle Swarm Optimization (PSO), each candidate solution is called a "particle" and represents a point in a $D$-dimensional space, if $D$ is the number of parameters to be optimized. Accordingly, the position of the particle $i$ may be described by the vector $x_i$ (Marini & Walczak, 2015):

$$x_k = [x_{k1}, x_{k2}, x_{k3}, ..., x_{kD}] \tag{5.8}$$

and the $K$-swarm is created as:

$$X = [x_1, x_2, ..., x_K] \tag{5.9}$$

Figure 5.1 depicts how a particle changes through each iteration according to the fitness in the swarm and its velocity (Kim, Kim, Choi, & Park, 2017).

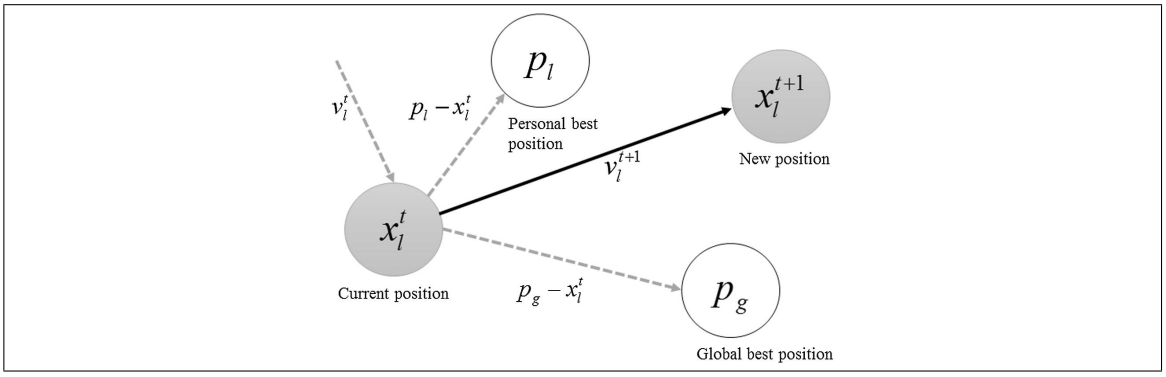


Figure 5.1. Particle Swarm Optimization algorithm (Kim et al., 2017).

At iteration $i$, the objective function is evaluated at each particle - obtaining what is known as the *fitness* value $f(x_k(i))$ - and logged *locally* and *globally*. Each particle has

a best personal position $p_k$ that registers the position for the best objective function value obtained by particle $k$ in all iterations. Likewise, a global best $g$ is logged for the whole swarm.

In searching for the optimal solution of the problem, the particles define trajectories in the parameter space (i.e., iteratively update their positions) based on the following equation of motion:

$$x_k(i+1) = x_k(i) + v_k(i+1) \tag{5.10}$$

where $v_k(i)$ is the vector collecting the velocities in all $D$ dimensions for iteration $i$ of the algorithm.

The velocity vectors govern the way particles move across the search space and can be defined by:

$$v_k(i+1) = v_k(i) + c_1 \left(p_k - x_k(i)\right) R_1 + c_2 \left(g - x_k(i)\right) R_2 \tag{5.11}$$

so that the particle velocity depends on the memory of its own best position (*cognitive component*) and of the whole swarm (*social component*). In some implementations, the social component can be restricted to a local neighborhood of each particle (Marini & Walczak, 2015).

The acceleration constants $c_1$ and $c_2$, which are real-valued and usually in the range $0 \leq c_1, c_2 \leq 4$, are called "cognitive coefficient" and "social coefficient" respectively, and modulate the magnitude of the steps taken by the particle in the direction of its personal best and global best, respectively (Marini & Walczak, 2015). On the other hand, $R_1$ and $R_2$ are two diagonal matrices of random numbers generated from a uniform distribution in $[0, 1]$, so that both the social and the cognitive components have a stochastic influence on the velocity update rule.

PSO can be summarized in the following algorithm.

ALGORITHM 5.1. *The Particle Swarm Optimization algorithm consists of the following steps (Marini & Walczak, 2015):*

*1) Initialize position for particles* $k = 1, ..., K$ $x_k(0)$.

*2) Initialize the best position for each particle* $p_k(0) = x_k(0)$.

*3) Calculate the fitness* $f(x_k(0))$ *of each particle and if* $f(x_k(0)) \leq f(x_j(0))$ *then make* $g = x_k(0)$.

*4) Until a stopping criteria is met, repeat the following steps for all* $K$ *particles:*

*a) Update the particle velocity according to equation 5.11.*

*b) Update the particle position according to equation 5.10*

*c) Evaluate the fitness of the particle* $f\left(x_k(i+1)\right)$.

*d) If* $f\left(x_k(i+1)\right) \leq f\left(p_k\right)$ *update the personal best* $p_k = x_k\left(i+1\right)$

*e) If* $f\left(x_k(i+1)\right) \leq f\left(g\right)$ *update the global best* $g = x_k\left(i+1\right)$

*5) Return* $g$ *and* $f(g)$.

As algorithm 5.1 mentions, the PSO routine ends when certain terminal conditions are met. These criteria, along with the all other parameters, are explained below:

**Maximum iteration number** $I$**:** The number of iterations that the algorithm is allowed to execute. Indirectly controls the execution time (linearly).

**Objective function stall:** Some variants of PSO - as the one used in this research - terminate when the objective function is *stalled* or "stuck" for a certain number of iterations. This constitutes a signal that the algorithm cannot find a better solution and is quite near its optimum. The cost function stall tolerance $\delta_s$ specifies the minimal change in the objective function to continue iterating. Maximum stall iterations number $I_s$ handles the limit of iterations before terminating a stalled PSO.

**Objective function tolerance** $\epsilon$**:** Commonly used in optimization solvers, this quantity states the optimality gap in the cost function.

**Particle distance stall:** Similar to the objective function stall, a particle stall criterion can be used independently of the value of the objective function. As this is non-differentiable and therefore not smooth, there is a possibility that the PSO algorithm gets stuck in a vicinity of points that are very close to each other but have sufficiently different fitness values. Analogous to $\delta_s$ and $I_s$, a particle distance stall tolerance $\epsilon_x$ and iteration limit $\Sigma_x$ can be defined. Since the objective is to compute *the sequence*, the search is shortened by looking past subtle differences in the cost.

### 5.1.3. Stability Issues of Random Forest Model Predictive Control

An interesting question posed by the RF-MPC strategy proposed is ensuring its stability. In section 3.2.2 the stability of predictive controllers was examined and some light was shed on the issue. However, the RF-MPC designed requires further analysis.

The objective function proposed in section 5.1.1.2 is composed of *deviations* from the reference, limits and changes in the manipulated variables. For positive semi-definite weight matrices, this function is minimized when all its components are zero:

$$\begin{aligned}\min_u V(u) &\Rightarrow \hat{e}_i(t+j) = 0 \;\; \forall j = 1, ..., N_y - 1 \;\; \forall i = 1, ..., n \\ &\Rightarrow \Delta u_k(t+j) = 0 \;\; \forall j = 1, ..., N_y \;\; \forall k = 1, ..., m \\ &\Rightarrow \epsilon_{ij} = 0 \;\; \forall j = 1, ..., N_y \;\; \forall i = 1, ..., n \\ &\Rightarrow |\hat{y}_i(t+N_y) - y_{i_{ss}}|^2 = 0 \;\; \forall j = 1, ..., N_y \;\; \forall i = 1, ..., n\end{aligned} \tag{5.12}$$

with its minimum at $V(u^*) = 0$. The cost function contains the origin - in terms of the predicted error and change in manipulated variables - and is continous.

All the conditions mentioned above are necessary to prove the stability of the predictive controller. However, the random forest representation is an *input-output* representation and therefore can be interpreted as a non linear transfer function. This is of the utmost importance, because stability proofs of MPC rely on *state-space* representations of

the system. Particularly, several assumptions on the boundedness and compactness of the state must be stated (Rawlings & Mayne, 2012).

As the RF-MPC deals with output regulation directly and has no state representation, all possible descriptions of the control problem domain must be made in terms of the output domain $\mathbb{Y}$ and the output terminal set $\mathbb{Y}_f$. These sets are compact for the thickener control problem. Additionally, the terminal set $\mathbb{Y}_f$ needs to be control-invariant.

Therefore, a gap needs to be bridged between the output representation of the predictive controller proposed in this research and its underlying state space model. By doing this, not only the RF-MPC stability can be proven, but also many other Machine Learning based predictive controllers.

A last issue that needs to be addressed in stability considerations of the designed controller is optimality. The Lyapunov function used for stability proofs in predictive controllers is the cost function; however, to show that the controller is globally asymptotically stable it is required that both a solution exists (feasibility) and that it is in fact the global optimum.

The optimization problem at hand, as described in section 5.1.2.1, has either a non-convex search space or a non-smooth objective function. As the methodology used to solve the problem is PSO - which is a local optimization method - global optimality cannot be guaranteed. However, *sub-optimality* can be used for the proposed controller for a well constructed PSO routine.

Suboptimal MPC requires a *non-increasing* cost function for successive MPC routine calls. Together with a terminal cost with a sufficiently large weight, suboptimality of the predictive controller can be used to prove local stability of the RF-MPC (Rawlings & Mayne, 2012).

## 5.2. Implementation of Random Forest Predictive Control

The final implementation of the RF-MPC yields a *toolbox* written in Matlab and Simulink programming language. Figure 5.2 shows the different components of the RF-MPC and their interaction.

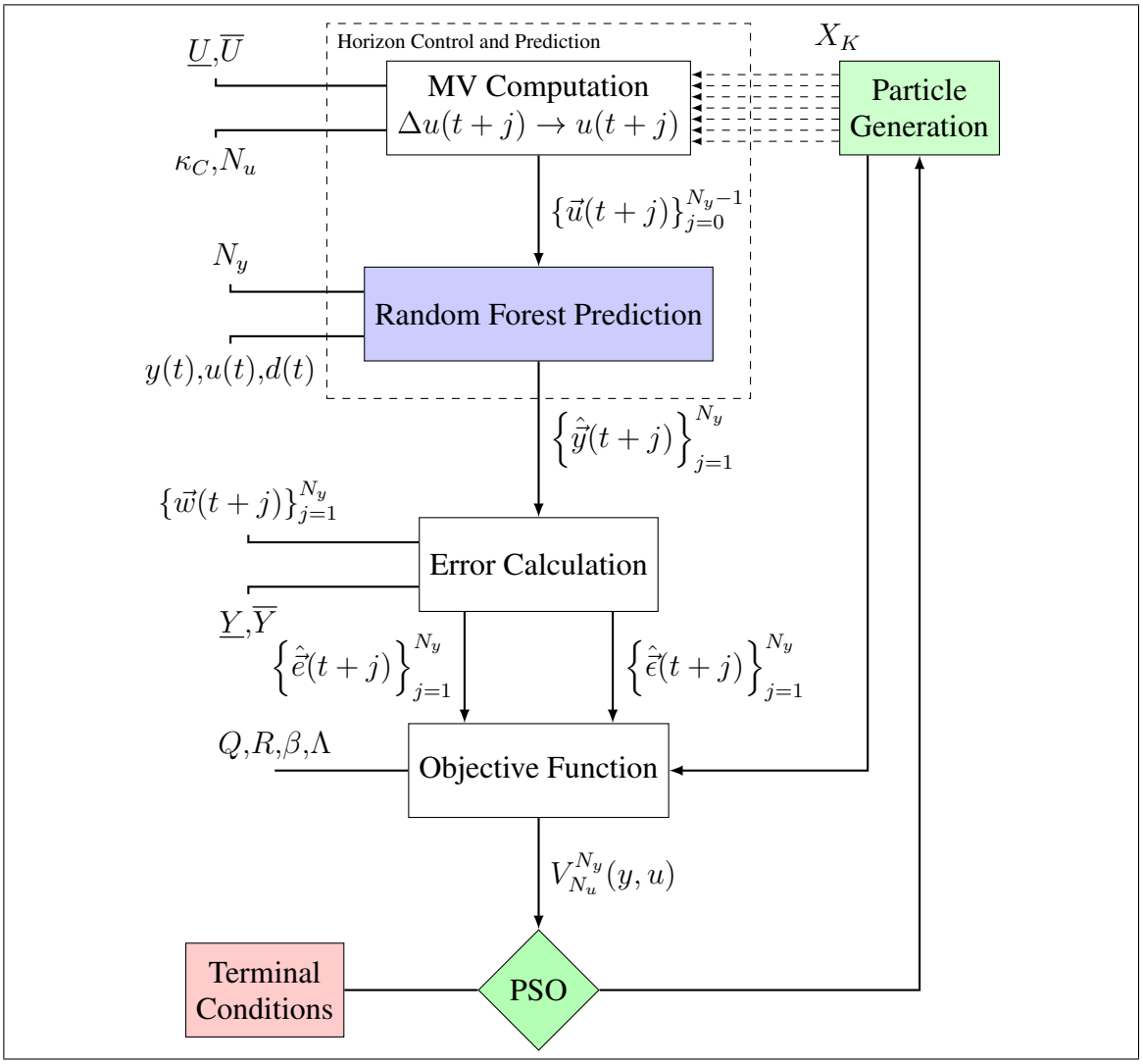


Figure 5.2. Random forest model predictive control algorithm and implementation.

The first important aspect of this implementation is *decission variable encoding* which is done in the particle generation block in figure 5.2. As explained in section 5.1.2.1, each

particle in the PSO strategy is a $D$ dimensional vector and each entry is therefore a decision variable.

The RF-MPC objective function formulation of equation 5.3 uses $\Delta u_k(t+j)$ as its decision variables. Therefore, these must be encoded into a particle for the PSO to iterate over them. The proposed encoding is *unrolling the manipulated variable sequence* into the particle:

$$\begin{gathered} x_k^T = [\{\Delta u_k(t)\}_{k=1}^m \{\Delta u_k(t+1)\}_{k=1}^m ... \{\Delta u_k(t+(N_u-1))\}_{k=1}^m] \\ \Delta u_k(t+j) = 0 \;\; \forall j \geq N_u \;\; \forall k = 1, ..., m \end{gathered} \tag{5.13}$$

which amounts to a particle of dimension $m \times N_u$ where each control instant in the horizon groups all manipulated variables. In the receding horizon strategy, the first $m$ components of the particle are sent to the plant while the others are discarded.

In some MPC strategies, a technique called *move blocking* can be implemented to increase controller performance. It consists on holding values of the MVs for time windows $\kappa_C$ inside the control horizon $N_u$ such that their rate of change is slowed down. This can be expressed as:

$$\begin{gathered} x_k^T = \left[\{\Delta u_k(t)\}_{k=1}^m \{\Delta u_k(t+\kappa_C)\}_{k=1}^m ... \left\{\Delta u_k(t+\left\lfloor \frac{N_y}{N_u} \right\rfloor \kappa_C \right\}_{k=1}^m \right] \\ \Delta u_k(t+s) = 0 \;\; \forall s \neq c\kappa_C \wedge N_u\kappa_C \end{gathered} \tag{5.14}$$

where now the MV sequence is spread through the whole horizon in $N_u$ blocks of $\kappa_C$ time frames. There is evidence that plants with slower dynamics can benefit from these strategy, even though there is a trade-off with the confidence and accuracy in the last computed values.

The particle generation block also handles the MV rate limits from constraints 5.5 by constraining each component in the particle $\delta_{u_k}$ accordingly.

An important aspect of the implementation described in figure 5.2 is its *vectorization*. All particles go through the process *simultaneously*; this means that the output of the particle generation block is a *matrix* whose rows are the particles and columns each particle entry:

$$X_K = [x_1 \; x_2 \; ... \; x_K]^T \tag{5.15}$$

The MV computation block from figure 5.2 transforms the incremental MVs encoded in the particles through equation 5.6 and saturates the control inputs if they surpass the limits. Therefore, the proposed design *handles all MV constraints by construction*.

The next block in the process of figure 5.2 is the Random Forest Prediction block, which executes the recursive predictive strategy exactly as depicted in figure 4.3. It is concerned with building the predictor vector $\phi_i$ for each output and generating predictions for the horizon.

In the error calculation block, all error terms from the objective function 5.3 are computed. In the case of the on-off variables $\epsilon_{ij}$ the predictions are checked to see if limits are violated. Finally, these results are fed to the objective function block which computes the fitness value for all particles.

The last step in figure 5.2 consists on the implementation of the PSO algorithm 5.1. After checking for terminal conditions, the particles' positions are updated and the whole process starts again every $\tau_C$ units of time.

### 5.2.1. Vectorization, Speed and Implementation Issues

As described earlier, the entire RF-MPC algorithm is vectorized. The elements that go through each of the blocks of figure 5.2 are matrices that contain transformations of the particles. As such, each block can be seen as a *transformation* of the particle data set.

For example, the prediction block transforms the particle matrix into a prediction matrix for each separate prediction step and for each separate output. As such, the natural

output of this block is a 3D array with as many rows as particles, the prediction of each output per step - hence the column space is of size $n$ - and height $N_y$. However, before exiting this block, these results are "flattened" to a matrix of dimensions $K \times (n \times N_y)$.

In early developments of this strategy, an unvectorized approach was used. Even though the exact same results were obtained, the execution time for one run was over 45 times more than the vectorized counterpart.

The random forest structures handled by the Matlab language posed an important problem both in memory usage and prediction speed. To cope with the first issue, each tree of each forest was saved in a compact manner, meaning that only the tree *structure* was saved and not the training or out of bag data. In total, the three forests used add up to 7 MB of memory approximately versus their initial 250. Even though for a simulation study such as this research this restriction is not so significant, for a real time implementation - specially in a micro-controller - memory is always a relevant constraint.

Initially, the built-in Matlab functions used for prediction amounted to 30% of the execution time of each RF-MPC iteration. The solution to overcome this issue was to *generate a C-MEX implementation* of the prediction routines.

The Mathworks software provides tools to generate C code directly Matlab routines. However, some functions cannot be translated to this language because they use the Matlab engine directly. In the case of this research, random forest predictions are not supported for C code generation.

Therefore, prediction functions - which load the compact trees and are vectorized for efficiency - where converted to Matlab executable files (`.mex`). This hybrid format supports all built-in Matlab functions but does not suffer from the overhead of calling the Matlab computational engine. The results of this approach dramatically reduced computation of one prediction nearly 100 times. The average prediction time obtained was 3 milli-seconds.

### 5.2.2. Final Remarks on Random Forest Model Predictive Control

The controller designed is the main output of this research. In this chapter, its mathematical formulation was explored complying with the concepts of predictive controllers stated in chapter 2. Its implementation was then specified, specially addressing issues such as speed and code efficiency. The main challenge of this approach is combining all three parts of a predictive controller in an efficient way, at least from the implementation perspective.

The main contribution of this research consists on bridging the gap between data-driven methods and predictive controllers. The algorithms and methods chosen constitute only a small subset of all possible strategies for data-driven predictive controllers. However, the RF-MPC proposed is, to the best of the author's knowledge, the first of its kind.

A few possible improvements can be mentioned at this stage. *Warm start* - that is, starting the solver with a past good MV sequence - can be expected to improve controller performance if needed.

The distance particle tolerance can be specified through a different distance metric. For example, more importance can be given to the first $m$ components which are the values actually sent to the controlled plant.

Because the implementation is believed to be efficient and robust, the next step is to test the controller designed in the control problem on which this investigation is centered. As mentioned in chapter 2, its performance will be quantified and compared to three other existing control strategies. By doing this, a better analysis of the utility of the method proposed can be attained.

# 6. APPLICATION TO PASTE TAILING PRODUCTION

The final implementation of the proposed predictive controller is to be tested on the problem defined in chapter 1 through the thickener simulator. As established, this is a complex control study. Added to this challenge is the use of the models identified in chapter 4 in a control context.

An initial description of the thickener operating point and studied operational limits is given in section 6.1. *Eight* different controller scenarios will be explored in this chapter so that all control objectives of chapter 3 can be addressed.

Section 6.2 specifies the RF-MPC and all three other benchmark controllers final parameters. Tuning was done based on the results in thickener stabilization described at the end of this section.

Finally, in section 6.3 three of the tests are analyzed in depth, both qualitatively and quantitatively. The tests where chosen to represent the three most important objectives in thickener control: underflow solids setpoint tracking, unmeasured disturbance rejection and handling considerable changes in solid throughput. Final conclusions of the overall performance of all controllers is given in section 6.3.4.

## 6.1. Thickening Process Specifications and Limits

As specified in section 4.3.1 the dataset was divided into two main portions: August of 2017 was used for system identification purposes while the first 300 hours of September were destined for controller testing purposes. An initial time window of 200 hours was used to drive the thickener to an adequate steady state, while the latter 100 hours were used for all other tests.

Table 6.1 lists the operating point for all process variables at the beginning of the simulation scenario.

Table 6.1. State of process variables at the beginning of the control experiments.

| | **Controlled Variables** | | | **Manipulated Variables** | | **Disturbance Variables** | | |
|---|---|---|---|---|---|---|---|---|
| | $T(t)$ [%] | $C_u(t)$ [%] | $h(t)$ [m] | $Q_u(t)$ [$m^3$/hr] | $F(t)$ [gpt] | $Q_f(t)$ [$m^3$/hr] | $C_f(t)$ [%] | $\mu(t)$ [%] |
| **Value** | 21.02 | 73.76 | 2.12 | 114.83 | 26.02 | 356.37 | 31.43 | 35.60 |

The underflow solids concentration specified in table 6.1 is well within the window specified in chapter 2. In fact, it is quite high and appears similar to that of real thickener constraints. On the other hand, torque levels are adequate for this operating point, but the interface level is too close to its lower limit and therefore risks reaching the feedwell. For the purposes of these tests, the interface level setpoint will be increased.

As such, the reference $\vec{w}(t)$ used for the control scenarios will be:

$$\vec{w}(t)^T = [21.02, 73.76, 4.12] \tag{6.1}$$

unless specified otherwise.

As mentioned in section 5.1.1, the control problem contemplates various constraints regarding both controlled and manipulated variables. Table 6.2 specifies upper and lower limits for all these variables.

Table 6.2. Process limits considered throughout all scenarios.

| **Limits** | **Torque** $T(t)$ | **Underflow Solids Concentration** $C_u(t)$ | **Interface Level** $h(t)$ | **Underflow Rate** $Q_u(t)$ | **Flocculant Dosage** $F(t)$ |
|---|---|---|---|---|---|
| Higher $\overline{X}$ | 22.07 | 75.97 | 6.25 | 125.00 | 31.00 |
| Lower $\underline{X}$ | 19.97 | 71.55 | 1.91 | 70.00 | 18.00 |
| Rate $\Delta X$ | - | - | - | 15.00 | 5.00 |

These limits come from different operation manuals and equipment specifications collected in the Fondef project that encapsulates this research. Because of the differences

between the real operational data and simulation, some of these had to be rescaled accordingly. However, all the CV constraints comply with the objectives explained in chapter 2.

Table 6.3 summarizes the eight different control scenarios studied in this research.

Table 6.3. Summary of the different control scenarios examined.

| **Test Type** | **Signal Shape** | **Magnitude of Change** | **Test Start Time [hr]** |
|---|---|---|---|
| Stabilization | Fixed Signal | 0 | 0 |
| $h(t)$ Tracking | Step Signal | $+2$m | 10 |
| $C_u(t)$ Tracking | Square Signal | $1\%$ | 10 |
| $Q_f(t)$ Rejection | Step Signal | $-38$ m$^3$/hr | 40 |
| $C_f(t)$ Rejection | Step Signal | $+8\%$ | 60 |
| $\mu(t)$ Rejection | Step Signal | $-4\%$ | 50 |
| Solids Throughput Increase Rejection | 20 hour Pulse Signal | $Q_f(t)$:$+38$ m$^3$/hr $C_f(t)$:$+10$ % | 30 |
| Solids Throughput Increase Rejection | 20 hour Pulse Signal | $Q_f(t)$:$-38$ m$^3$/hr $C_f(t)$:$-10$ % | 30 |

## 6.2. Control Tuning and Testing

Automatic controller tuning is a field of research within itself (Alvarez, Tadeo, & Grimble, 2002). There are many approaches - even some based on Machine Learning algorithms - that can be used to adjust controller parameters (Shridhar & Cooper, 1997).

However, it is often less expensive to manually tune controllers, specially when their parameter vector is large as is the case for MPC controllers. This approach will be followed on the thickener stabilization tests.

Even though this process was largely done by observing the evolution of CVs and MVs when stabilizing the thickener, attention was paid to quantitative measures. To assess control performance, *five* different metrics will be used, of which the first and third - *MSE* and *MAE* - will be more relevant to this study. These are listed below:

**Mean Squared Tracking Error (MSE):** As usual, this criterion is the sum of squares in the error along the control horizon.

$$J_{\text{MSE}} = \frac{1}{N}\sum_{k=1}^{N}\left(w(k) - y(k)\right)^2 \tag{6.2}$$

**Integral Absolute Error (IAE):** This criterion penalizes offset in control variables by cumulatively adding the error in the outputs.

$$J_{\text{IAE}} = \sum_{k=1}^{N}\left|w(k) - y(k)\right| \tag{6.3}$$

Additionally, as this criterion uses the *magnitude* of the error and not its squared value, the units considered correspond to that in which the CVs are measured.

**Maximum Absolute Error (MAE):** A key aspect to quantify in controller performance is how far an output deviates from its setpoint. Large deviations can impact production and overall industrial performance.

$$J_{\text{MAE}} = \max_{k\in\{1,\ldots,N\}}\left|w(k) - y(k)\right| \tag{6.4}$$

**Squared Control Effort (SCE):** By observing this, error tolerance can be balanced with manipulated variable effort. Such a metric consists on the cumulative sum of the squared changes in the control inputs.

$$J_{\text{SCE}} = \sum_{k=1}^{N}\Delta u(k)^2 \tag{6.5}$$

*Both* the conventional MPC and the RF-MPC use this to determine the optimal MV sequence, only scaled by the weight $R$.

**Integral Squared Control Action (ISU):** It can be argued that saturating actuators is not a good control strategy, since the ability of the controller to cope with disturbances or setpoint changes can be affected. Therefore, a criterion that penalizes the controller variance provides insight into how far the control action is driven from a stable average.

$$J_{\text{SCE}} = \sum_{k=1}^{N} \left(\Delta u(k)^2 - \bar{u}\right)^2 \tag{6.6}$$

### 6.2.1. Random Forest Model Predictive Control

The first step in tuning any MPC controller is normally specifying parameters related to time and the predictive horizon. The control action time $\tau_C$ has already been fixed at five minutes. On the other hand, the results of chapter 4 showed the need to use a relatively short horizon 90 minutes for predictive horizon $N_y$. As such, the only parameter left to select is the control horizon $N_u$. Table 6.4 summarizes these values.

Table 6.4. Specification of RF-MPC parameters related to prediction and control horizons.

| **Parameter** | **Value** |
|---|---|
| Control Action $\tau_C$ | 5 min |
| Prediction Horizon $N_y$ | 90 min |
| Control Horizon $N_u$ | 15 min |

Controller move blocking was tried, but performance decreased substantially. On the other hand, values from 5 to 30 minutes were tried for the control horizon finding that a substantial decrease in performance occurred for values over 25 minutes. For short $N_u$, however, results were relatively similar, and as such the largest was chosen. Naturally, all these parameters can change depending on the controlled variable; however, they were kept uniform for simplicity.

Given these constraints, the PSO solver was tuned. As mentioned in section 5.1.2.1, these parameters are critical in the controller performance. Table 6.5 lists the final values for all its parameters.

As control action time was fixed to 5 minutes in section 4.3, the PSO must terminate before a new control action is demanded. Instead of specifying a termination time, this was handled through the maximum iterations number $I$ chosen in table 6.5. Results showed

Table 6.5. Particle Swarm Optimization tuned parameters.

| **Parameter** | **Symbol** | **Value** |
|---|---|---|
| Swarm Size | $K$ | 100 |
| Maximum Iterations Number | $I$ | 30 |
| Cost Function Stall Tolerance | $\delta_s$ | $5 \times 10^{-3}$ |
| Maximum Stall Iterations | $I_s$ | 21 |
| Objective Function Tolerance | $\epsilon$ | $1 \times 10^{-3}$ |
| Particle Distance Tolerance | $\epsilon_x$ | $1 \times 10^{-3}$ |
| Maximum Particle Stall Iterations | $\Sigma_x$ | 21 |

that no further improvement was obtained for more iterations, while solver time per control action was 2.42 seconds in average.

As such, swarm size can be increased heavily. Results in the literature discuss that there should be at least 10 particles per optimization dimension (The MathWorks, 2018) (Marini & Walczak, 2015). As the particle $x_k$ is a 6 dimensional vector, a swarm size of 100 particles can be expected to work properly.

Table 6.6 lists the final parameters regarding the controlled variables in the objective function, which where tuned regarding the controller performance in the thickener stabilization tests.

Table 6.6. Specification of RF-MPC parameters related to controlled variables.

| **Parameter** | **Torque** $T(t)$ | **Underflow Solids Concentration** $C_u(t)$ | **Interface Level** $h(t)$ |
|---|---|---|---|
| Tracking Weights $Q_i$ | 1.00 | 100.00 | 100.00 |
| Terminal Weights $\beta_i$ | 1.00 | 100.00 | 100.00 |
| Constraint Weights $\Lambda_i$ | 10000.00 | 10000.00 | 10000.00 |
| Normalization Coefficient $l_i$ | 3.00 | 3.00 | 5.00 |

As table 6.6 states, more importance was given to the cost in underflow solids concentration and interface level. This is to accomplish the control objectives previously described. The normalization coefficient $l_i$ was chosen with respect to the average variation of the CVs in the training set. Similarly, table 6.7 shows the coefficients that affects MVs in the controller.

Table 6.7. Specification of RF-MPC parameters related to manipulated variables.

| **Parameter** | **Underflow Rate** $Q_u(t)$ | **Flocculant Dosage** $F(t)$ |
|---|---|---|
| Control Effort Weights $R_j$ | 0.05 | 0.50 |
| Normalization Coefficient $s_k$ | $\delta_{u_1}$ | $\delta_{u_2}$ |

Even though the objective of this research is designing a *stabilizing* predictive controller, an *optimal* predictive strategy is a next logical step. In thickener control literature, flocculant usage is regarded as the main variable whose cost is to be reduced (Cacciuttolo & Holgado, 2016). The control effort weights chosen penalize flocculant variation 10 times more, thereby considering a soft notion of optimality.

### 6.2.2. ARIMAX-Based Model Predictive Control

To provide a fair comparison between both predictive techniques used in this research, the parameters used for the conventional MPC controller *are the same as those for the RF-MPC*. This will provide a one-to-one insight of the difference in the predictive models and the optimization algorithm.

As explained in chapter 2, current MPC strategies rely on state-space representations of the system rather than transfer functions such as ARIMAX models. Therefore, the models identified in chapter 4 where transformed to their state-space equivalent through a minimal realization. To include the modeled unmeasured disturbances for each output - recalled from the ARIMAX equation 4.3 - an augmented state space representation was used.

The linear models identified in chapter 4 resulted in *closed-loop unstable* models. Hence, *order reduction* of the models was carried out resulting in a state space representation of order 20. This procedure will in general affect predictive accuracy and therefore the decissions made by the controller.

However, dealing with unstable models in predictive control strategies can be detrimental to their performance (Camacho & Bordons, 2007). Furthermore, the thickening process is believed to be input-output stable; therefore, this result *highlights* another disadvantage of conventional identification methods.

The Matlab Model Predictive Control toolbox offers various algorithms for estimation and optimization. Regarding the first, a Kalman filter will be used based on the reduced state-space model; the process and measurement noise covariances are inherited from the initial identification stage.

Its objective function corresponds to the same as the RF-MPC in equations 5.2 and 5.3. Since the model is linear, the cost function quadratic and the search space convex, gradient based methods can be used to solve the online optimization problem.

Specifically, an SQP algorithm will be used and implemented. The Mathworks MPC toolbox incorporates the KWIK algorihtm (Schmid & Biegler, 1994) for this effect. This algorithm solves the unconstrained MPC problem in a first step and uses that solution as a starting point for the SQP optimization routine.

### 6.2.3. Expert Control

A corrective action is supplied at each instant regarding the error and integral of the error making this a *PI Expert control*. To achieve this, an error range for each CV is considered and divided into regions. Additionally, these regions can be classified as *rough* and *smooth* control regions, the latter being closest to zero error. The first region only considers proportional corrective action to drive the output safely towards the reference; the smooth regions use integral action to achieve zero permanent error.

If the magnitude of the resulting control action is larger than a specified limit - which are the limits for the MVs in table 6.2 divided by the number of error regions - it is then saturated.

The last aspect regarding the expert controller designed is an *attenuation* component. An additional term $\Delta u_D$ is *subtracted* from the control action; this term depends on the sign of the error and the sign of the error change. If both signs are different, then the controller is driving the output *towards* the reference and the MV effort is reduced.

A pseudo-code implementation of the generated routine can be found in E.1. Table 6.8 lists the parameters used by the expert controller described.

Table 6.8. Tuned parameters for expert control.

| **Parameter** | $T(t)$ **Logic** | $C_u(t)$ **Logic** | $h(t)$ **Logic** |
|---|---|---|---|
| Sample Time $\tau_C$ | 5 min | 5 min | 5 min |
| Error Range Considered | [-5.00 , 5.00] | [-10.00 , 10.00] | [-4.00 , 4.00] |
| Number of Error Regions | 21 | 21 | 21 |
| Integral Action Range | [-0.47 , 0.47] | [-0.95 , 0.95] | [-0.38 , 0.38] |
| Proportional Constant $k_p$ | 1.00 | 10.00 | 2.00 |
| Integral Time Constant $T_i$ | 100.00 | 10.00 | 10.00 |
| Control Effort Attenuation | [0.10 , 0.01] | [0.10 , 0.01] | [0.10 , 0.01] |

### 6.2.4. PI Control

Because there are only two control inputs and three outputs, an MV-CV coupling strategy was used. Since literature on thickener control regards underflow rate as the most influential variable on thickener performance, it was reserved to control underflow solids concentration. Indirectly, this controls the torque in the thickener. Flocculant dosage was used to control the depth of the interface. The main difference between the PI controller is that it operates at a rate five times faster than the other controllers. PI controllers are actually implemented at field level, and therefore can operate at a much faster rate.

The PI controller was tuned following the references and guidelines in (Langlois, 2018). Fine-tuning was performed through the thickener stabilization tests. These parameters are listed in table 6.9.

Table 6.9. PI control tuned parameters.

| **Parameter** | $C_u(t)$ **Loop** | $h(t)$ **Loop** |
|---|---|---|
| Control Action $\tau_C$ | 1 min | 1 min |
| $K_p$ | 90.00 $m^3/hr$ | 3.00 gpt |
| $K_i$ | 3.60 $m^3/(hr \times min)$ | $1.50 \times 10^{-4}$ gpt/min |
| Windup Coefficient $K_b$ | 0.01 | 0.02 |

### 6.2.5. Thickener Stabilization

As mentioned, all the parameters chosen above were iteratively selected by analyzing the performance on thickener stabilization to the setpoint 6.1 for the first 200 hours of September 2017.

Such a test is only useful if the free response of the thickener leads to an unstable plant or to a radically different operating point. To prove that this is the case, figure 6.1 shows the evolution of all CVs for the specified time window if control inputs were kept at their initial value. Figure 6.1b plots the measured DVs in that time.

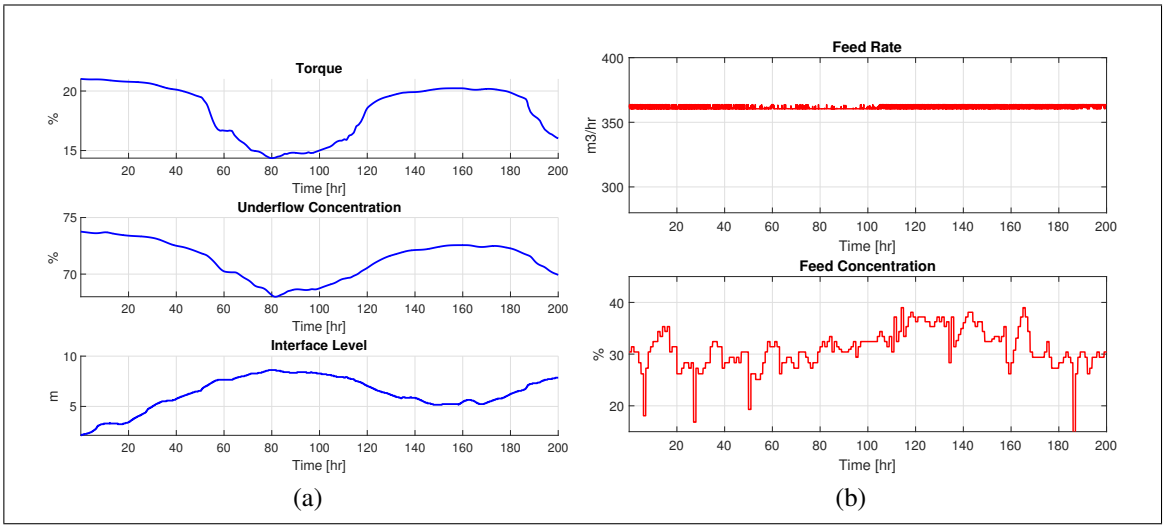


Figure 6.1. Free response of the thickener and disturbance variables for the first 200 hours of September 2017.

It is clear that the operating point cannot be sustained without control. Torque drops substantially, well below the limits of table 6.2. Interface level rises heavily, almost reaching the bottom of the thickener. Even though underflow solid content can guarantee paste quality, the other CVs are well past their limits and as such pose threats to the operation.

Control performance using the final tuning parameters is depicted in figure 6.2. It is important to recall that these tuned controllers achieved *the best* results among all options tried.

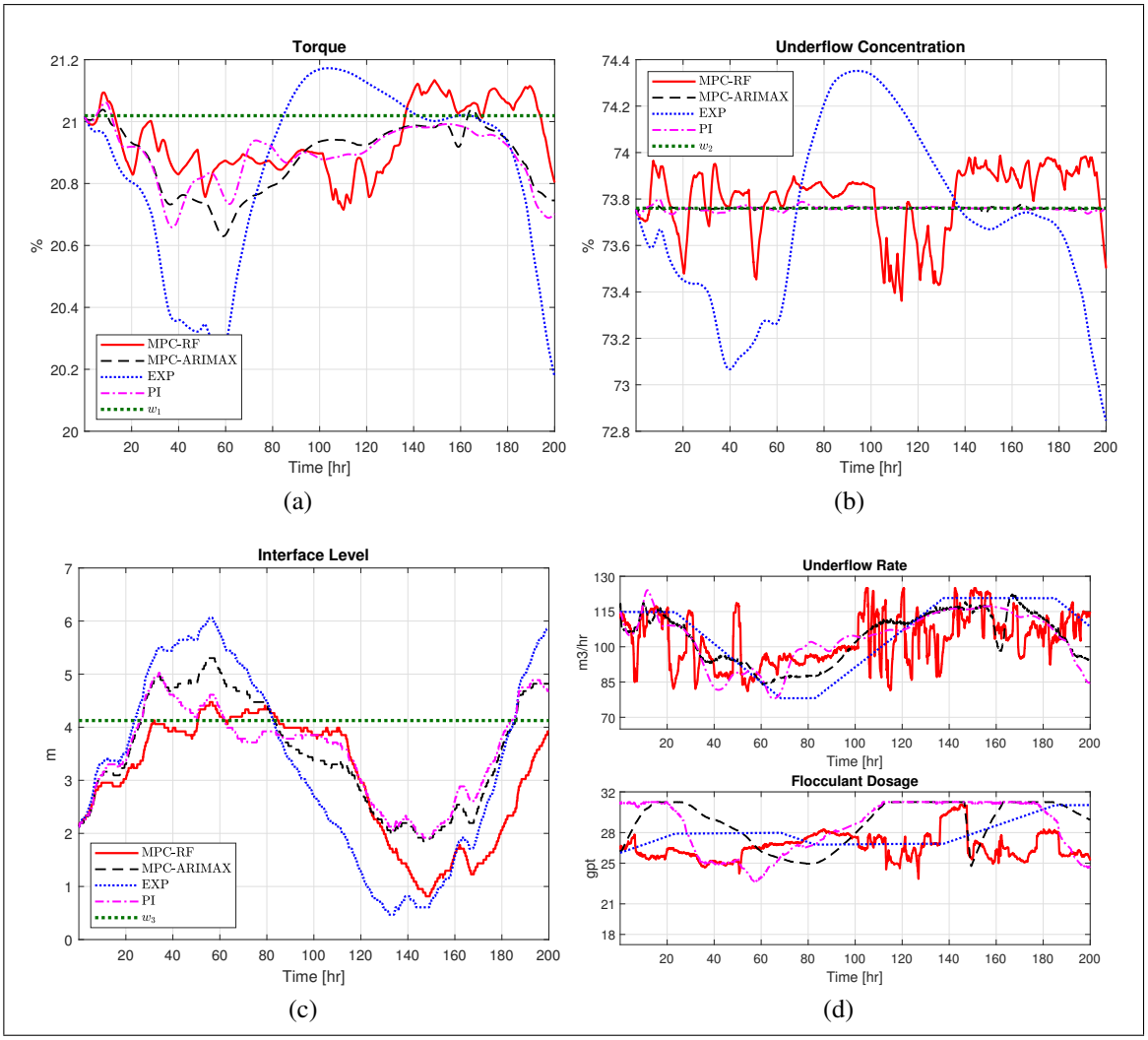


Figure 6.2. Comparison of control performance for thickener stabilization.

Results clearly indicate that the best controllers for this scenario are conventional MPC and PI control. Both of these controllers stabilize underflow solids concentration with close to zero error, with the predictive strategy offering a small advantage. For both torque and underflow solids concentration, none of the limits in table 6.2 are violated by any controller.

The proposed strategy ranks third for this criteria. A closer look at figures 6.1b and 6.2d reveals a possible reason behind this behaviour. Fast and large variations in feed solids concentration make the controller output oscillate significantly. This is reasonable given the identification of chapter 4. However, the proposed strategy *does* stabilize $C_u(t)$ within a band of $\pm 0.03\%$ of the specified setpoint.

It is clear that the expert controller is insufficient to stabilize thickener operation. The worst results are obtained through this strategy for *all* CVs. Even though the expert approach can be perfected continuously by adding and modifying rules in the knowledge base, this process can become complex. Any other strategy appears to be preferable.

Another interesting aspect to observe about these results is interface level control. No controller can keep this variable in its setpoint which is probably due to the fact that the specified operating point is difficult to maintain under the effect of disturbances. In fact, from hours 110 to 150 an important increase of feed concentration can be seen in figure 6.1b. This explains the large deviation from the interface level setpoint in figure 6.2c. Before this time period, the best results were obtained through the RF-MPC strategy which had nearly zero error.

Table 6.10 illustrate this quantitatively through some of the criteria mentioned previously in this section. The full results can be found in appendix E.

As mentioned before, setpoint tracking is of less importance than limit violation for both torque and interface level. The MAE scores of table 6.10a show that the best strategies are again the conventional MPC and PI feedback loops. As a matter of fact, both of

Table 6.10. Comparison of control performance between all techniques for thickener stabilization.

(a) Interface level (CV) criteria.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|
| RF-MPC | 244.73 | 3.31 |
| MPC | 145.05 | 2.28 |
| Expert | 382.52 | 3.66 |
| PI | 113.51 | 2.21 |

(b) Manipulated variable criteria.

| **Criterion** | **SCE** $[1 \times 10^{0}]$ | | **ISU** $[1 \times 10^{6}]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 5744.70 | 40.68 | 82.52 | 1.22 |
| MPC | 390.37 | 2.00 | 83.75 | 3.30 |
| Expert | 6.09 | 0.05 | 159.83 | 1.09 |
| PI | 5.84 | 6.53 | 89.37 | 4.73 |

these stay within the limits of table 6.2. The proposed RF-MPC cannot achieve this and breaks the lower limit.

Apparently the most important drawback of the proposed RF-MPC involves the behaviour of the MVs, specially underflow rate. Table 6.10b quantifies the erratic behaviour depicted in figure 6.2d. The MV effort criteria is almost 1000 times as large as the PI score. This is probably because of two reasons.

It is natural that predictive strategies that operate at $\tau_C$ time units have higher SCE costs than a faster control loop since the latter makes smaller adjustments faster. This explains why the conventional MPC incurs in an SCE cost of nearly 80 times as large. However, the RF-MPC exhibits a much more inefficient behaviour.

Additionally, penalization of MV change is, according to table 6.7, much smaller than the weights $Q_i$ associated with setpoint tracking. This action is deliberate, but can be readily reevaluated. However, these results were the best of all the experiences tried and hence this was admitted for the purposes of this research.

On the other, both MPC strategies achieve the lowest ISU with the RF-MPC. This is clearly depicted in figure 6.2d where the control inputs from both predictive controllers tend to group towards the center of the limits. If a significant disturbance affected the process during the simulation horizon the PI controller could perform unacceptably.

The last interesting aspect to observe from this test is how the PSO operates in this context. Figure 6.3 shows the movement of the swarm for one MPC routine 70 hours into the simulation. Since in this case each particle is a 6 dimensional vector - 2 inputs and 3 control steps - the plots depict the first two components of the particle, which are the ones that are actually applied.

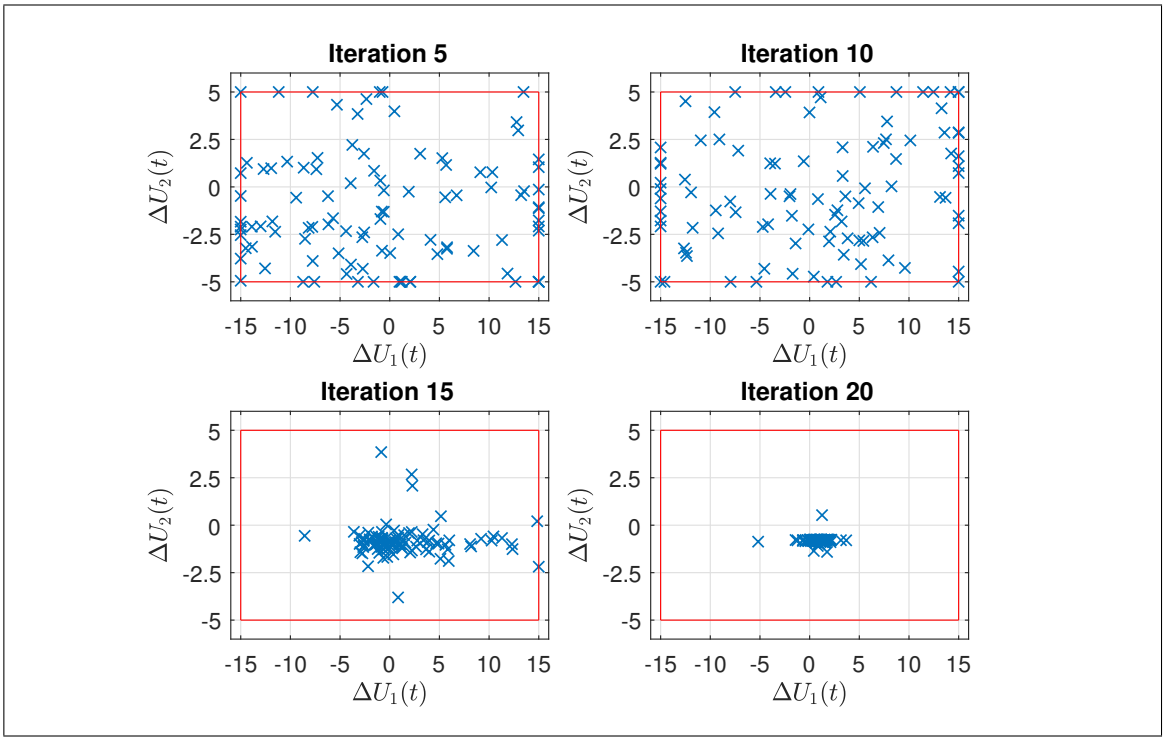


Figure 6.3. Swarm evolution during RF-MPC operation in thickener stabilization.

Initially, the search space - delimited by the rate of changes in the MVs - is sparsely populated; as PSO iterations progress, the swarm tends to group clearly towards a center region. Therefore, the algorithm detects that *at least* a local minimum can be found near the center of the search space.

This can be attributed to the RF-MPC tuned parameters and the random forest prediction. The forecast is probably steady and as such the controller chooses to minimize the effort while admitting some error in the interface level, for example. The final values

applied to the thickener in this iteration (incrementally) are $0.92$ m$^3$/hr and $-0.51$ gpt, where it can be seen that the controller chooses to minimize flocculant addition.

### 6.3. Results of Control Performance

After the 200 initial hours examined in section 6.2.5, the other tests of table 6.3 were conducted for the following 100 hours. For this time frame, figure 6.4 shows the collected DV data.

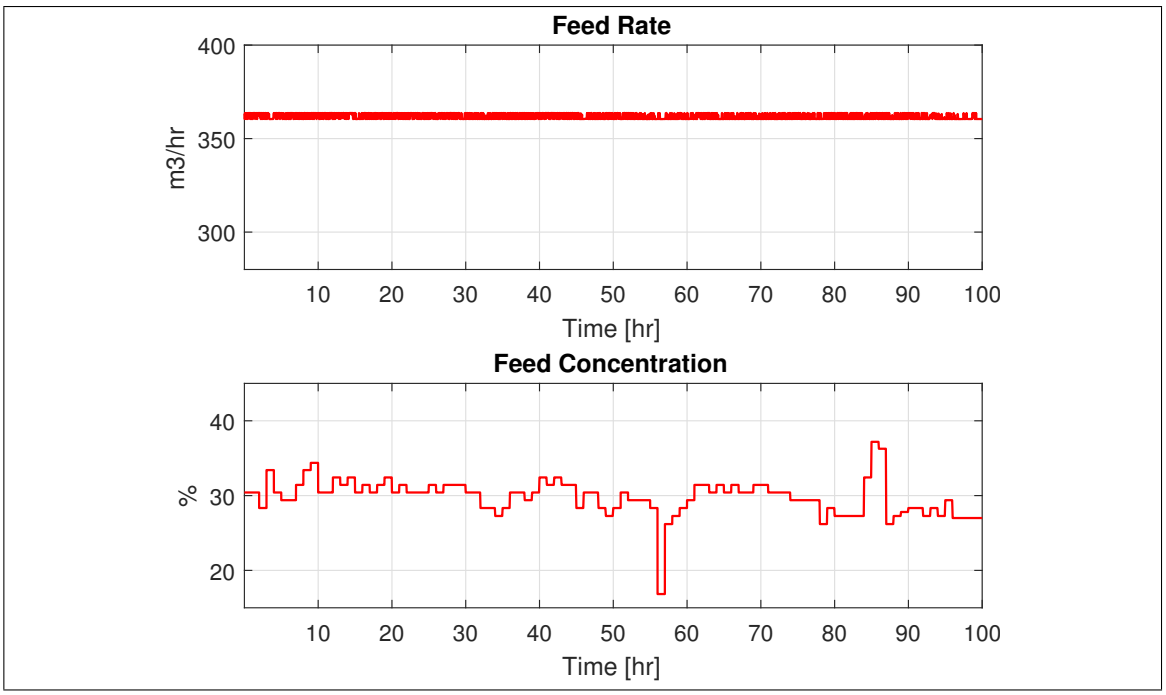


Figure 6.4. Disturbance variable evolution contemplated for control tests.

Perhaps one of the most relevant scenarios of table 6.3 is *reference tracking*. Since ultimately the goal is to take the thickener through different operating points while coping with industrial constraints, reactions to setpoint changes are of significance.

Since feed particle size is an unmeasured disturbance, no actual operational data of this variable exists. Therefore, its effect will be simulated through a *downstream* step change

in feed solids concentration. In other words, the decrease of 4% specified in table 6.3 will be applied to feed solids content *after* it is measured by the predictive controllers.

The last experiments studied will be abrupt changes in solid throughput. Both a simultaneous increase and decrease of 20 hours in feed rate and solid content will be simulated. Such a scenario is a relatively common situation, since change in solid intake is due to abnormalities in grinding and extraction of the mineral.

### 6.3.1. Underflow Solids Concentration Reference Tracking

Figure 6.5 compares how all controllers react to consecutive setpoint changes in underflow solids content. It can be seen that PI control is the most successful for tracking the reference signal, followed somewhat closely by the RF-MPC.

For the first setpoint change, all controllers - except for the expert approach - drive the underflow rate towards its upper limit. In fact, the conventional MPC saturates its output while the RF-MPC produces subtle variations.

However, as shown in figure 6.5b, the conventional MPC stabilizes $C_u(t)$ with no permanent error, while the proposed strategy is unable to do so. The reason behind this is that the interface level $h(t)$ rises towards its limit. Hence, the RF-MPC reduces $Q_u(t)$ beforehand to track both setpoints simultaneously. The PI strategy follows this pattern as well, but because of variable coupling the setpoint can be reached.

The conventional MPC, on the contrary, violates the interface level limit of table 6.2. As $h(t)$ drops rapidly, a situation similar to the results of chapter 4 is encountered. The poorly identified exogenous influence on the CV of the linear model produces a destabilizing response in the conventional MPC strategy. The linear predictor estimates that lower saturation of both MVs is optimal. The RF-MPC strategy, however, offers a radically different solution, attaining stabilization of $C_u(t)$ with no permanent error between hours 40 to 70.

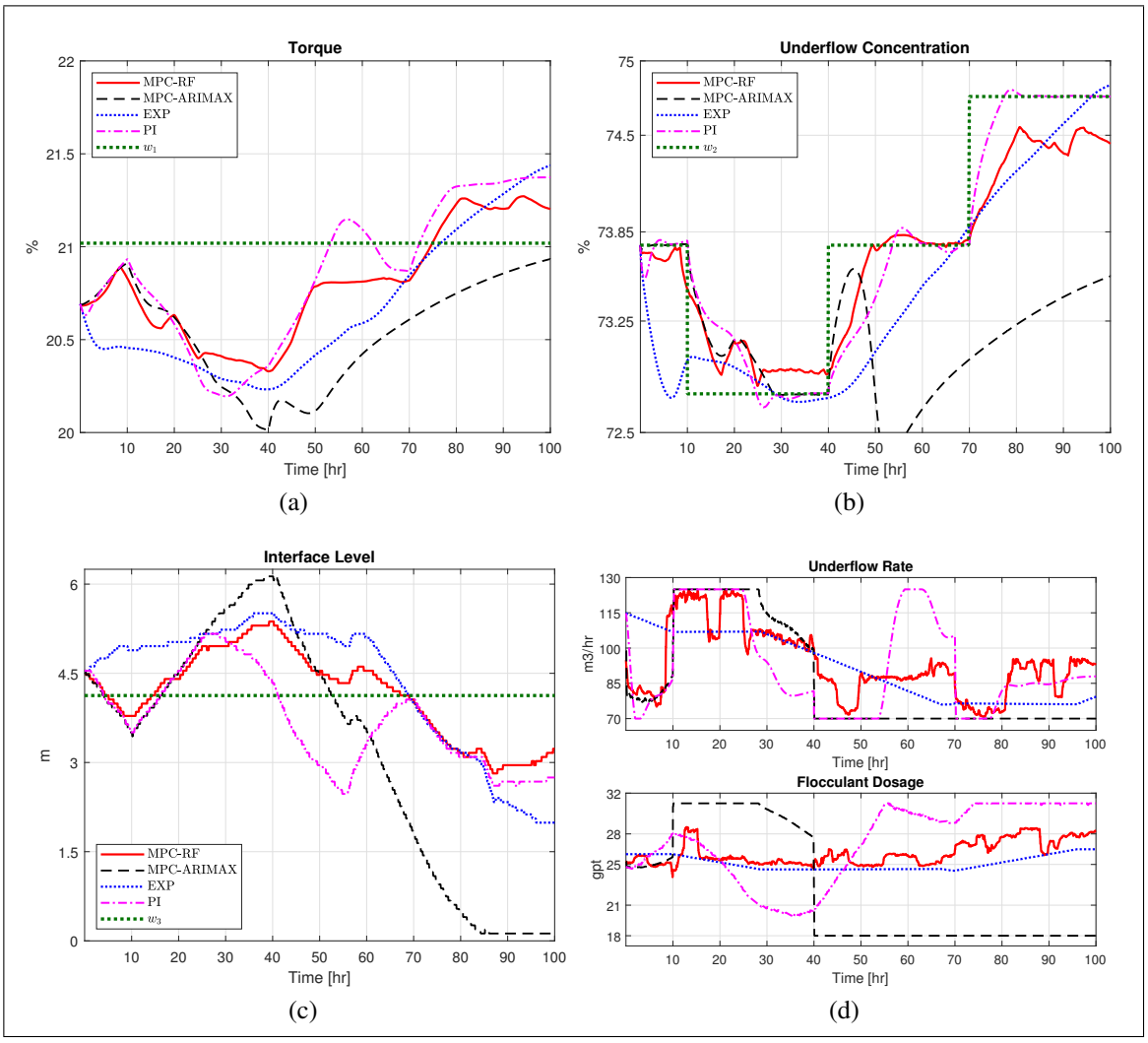


Figure 6.5. Comparison of control performance for underflow solids concentration setpoint tracking.

Flocculant use varies greatly between all control strategies. This is consistent with the less explored effect of this MV on the process explained in chapter 2. PI regulation focuses only on the *error of the interface level* and effectively handles interface variation. Since this strategy deals with only two CVs a different controller output is likely to occur.

The RF-MPC, on the other hand, tends to maintain flocculant levels in the center portion of its limits. This is a consequence of the tuning specified for the controller as well as the predictive model identified. Only at the last portion of the simulation horizon the

flocculant levels increase significantly. At this point the RF-MPC fails to drive U.S.C to its setpoint. The decrease of F.S.C. in the last 20 hours of the horizon explains said behaviour. The RF-MPC increases flocculant added to the system to maintain both $C_u(t)$ and $h(t)$ as close as possible to their setpoints as a response.

A closer inspection shows that the RF-MPC includes a *slower* component than all controllers. The most important changes in both of its MVs happen every 5 to 6 hours. Again, this is a secondary effect of the models trained and used for prediction as well as construction of predictors $\phi_i$. Essentially, the RF-MPC is operating in *residence time batches* and is therefore biased towards this rate of change.

Table 6.11 compares the MSE for all variables for this simulation scenario. The full results can be found in appendix E.

Table 6.11. Comparison of the MSE for all variables between the four controllers designed for underflow solids reference tracking.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | | |
|---|---|---|---|
| **Variable** | Torque $T(t)$ | U.S.C. $C_u(t)$ | Interface Level $h(t)$ |
| RF-MPC | 13.92 | 10.89 | 54.36 |
| MPC | 30.93 | 94.85 | 494.32 |
| Expert | 26.90 | 24.60 | 128.00 |
| PI | 16.77 | 9.50 | 80.32 |

The proposed RF-MPC strategy is the *best alternative* for setpoint tracking when considering *all* controlled variables. PI regulation follows closely but neglects torque control. Conventional MPC and expert control have the worst performance regarding tracking of this output.

Variable coupling is, in this case, helpful in achieving setpoint tracking. Since all other strategies are multi-variable and do not have predefined input-output coupling, such a result is not particularly surprising.

### 6.3.2. Rejection of Feed Particle Size Change

Modeling of feed particle size $\mu(t)$ change will be done as a downstream decrease in feed solids concentration. This decision is justified by the description of both variables provided in section 2.2.2. Figure 6.6 illustrates the results for this simulation on all controlled variables and control inputs.

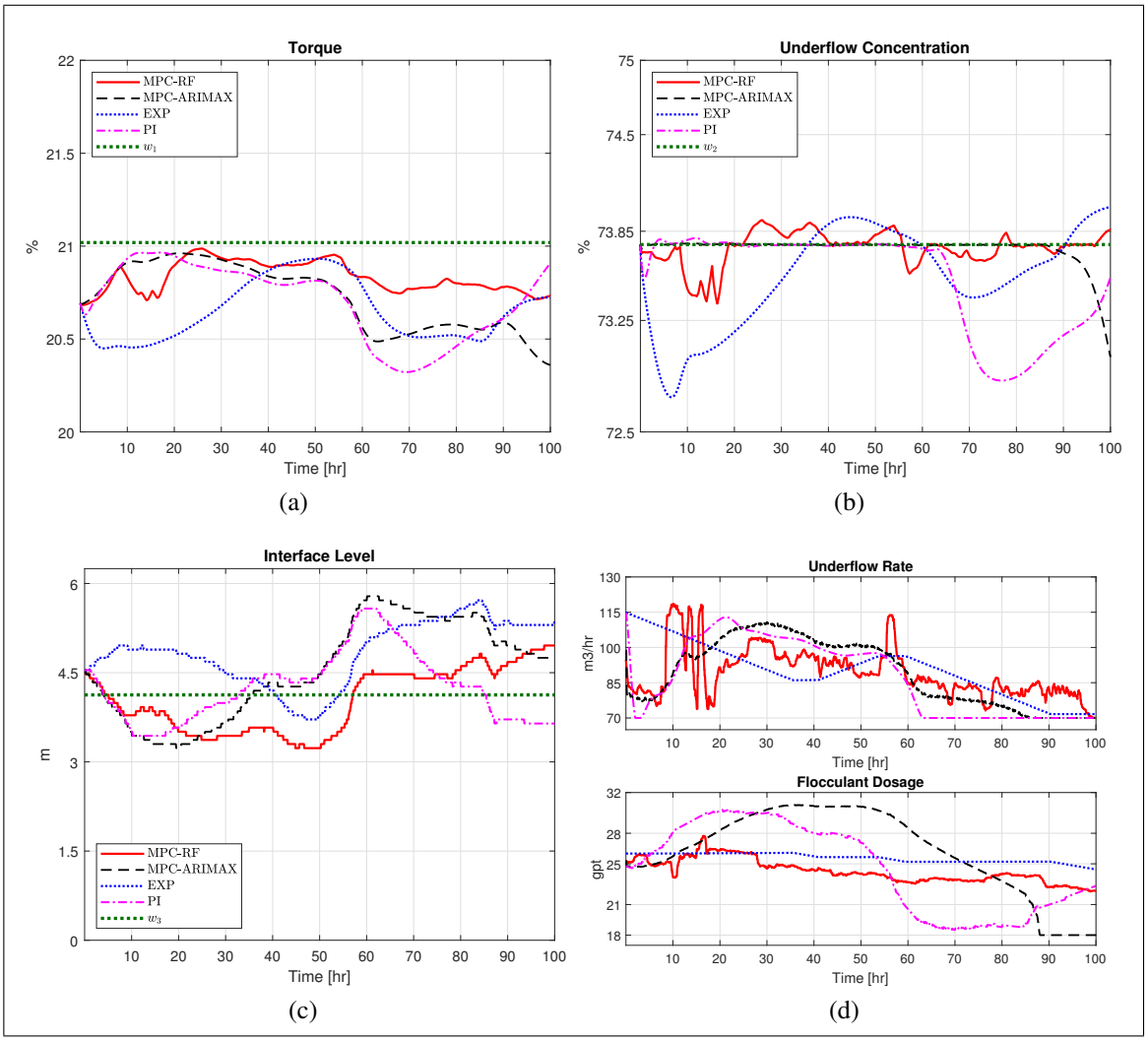


Figure 6.6. Comparison of control performance for feed particle size change.

RF-MPC outperforms every other controller regarding underflow solids concentration. In fact, the PI controller - which up to this point exhibited good behaviour - is heavily

affected by the change in feed particle size. RF-MPC incurs in a maximum error of 0.40% in this CV while the PI strategy almost doubles this amount.

Because the disturbance change is modeled as a *step*, it affects thickener operation from hour 50 to the end of the simulation horizon. As mentioned in chapter 3 this disturbance is believed to behave like an integrator in controlled variables. Controllers are then subject to an important challenge in keeping the plant at its initial operating point. Figure 6.7 provides a closer look into controller performance for a time window centered around the disturbance change.

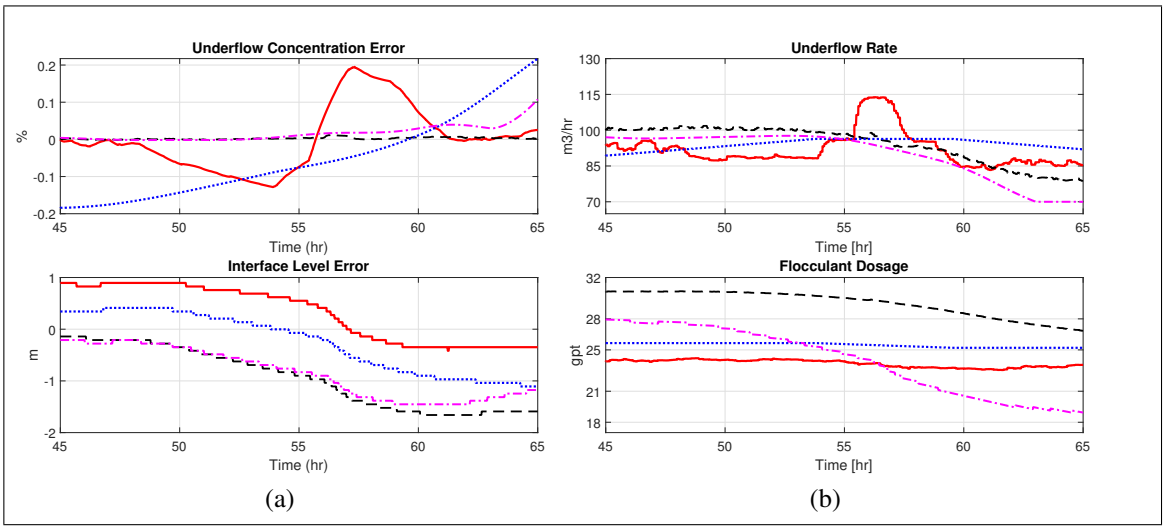


Figure 6.7. Close-up comparison of control performances before and after feed particle size change. Figure 6.7a plots the error, while figure 6.7b displays controller outputs. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

A decrease in feed particle size affects thickening similarly to feed solids concentration. After a delay in the order of the residence time, underflow solids concentration should decrease substantially. In figure 6.7b it can be seen that the RF-MPC overreacts to this change. However, it quickly corrects the outcome and stabilizes underflow concentration *while maintaining the lowest interface error*. Quantitative results can be found in appendix E.

As mentioned in chapter 3, one of the main advantages of predictive control is robustness against unmeasured disturbances. This test serves as an example of the capabilities of such strategy. Results show that the proposed strategy is *the best controller* for such purposes.

Comparison of the RF-MPC and conventional predictive strategy shows that the latter performs better for the first portion of the disturbance effect. However, figure 6.6b depicts that from hour 90 underflow solids content is destabilized by this controller. This is because of flocculant and underflow rate saturation at their lower limits, as shown in figure 6.6d.

### 6.3.3. Rejection of Abrupt Solid Throughput Change

To completely measure the change in solid throughput densimeter data is needed which was not present in the dataset. However, the two measured DVs in this research impact solid throughput in the thickener. By changing them *both* simultaneously the impact on thickener operation is enlarged substantially. Figure 6.8 illustrates both the increase and decreasing scenarios.

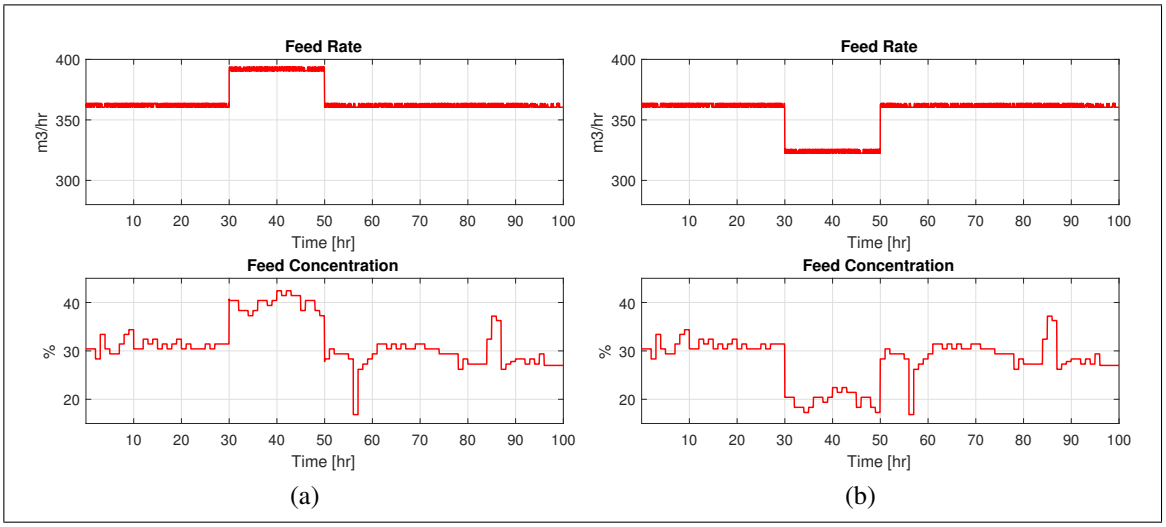


Figure 6.8. Pulse increase (figure 6.8a) and decrease (figure 6.8b) in solid throughput in the thickener.

The first column of figure 6.9 compares all feedback strategies for an increase in solid intake. Both the PI and RF-MPC strategies keep the first two outputs near their setpoints, with the PI loop achieving the best performance regarding underflow solids content. As a matter of fact, the PI controlled thickener appears to be insensitive to this disturbance. RF-MPC also stabilizes this output.

It is clear that the conventional MPC cannot cope with this change in solid throughput. The explanation lies in the linear model on which the control inputs depend. The model training data had a much lower feed rate and solids concentration. Even though offsets and trends where not considered for identification with the ARIMAX strategy, the increase in solid throughput represents a *substantially different operating point*. As such, the predictions cast by the conventional MPC differ from the actual plant output which destabilizes it.

Regarding solid throughput decrease, a different situation can be observed. The best performance is obtained by far by the RF-MPC. *No other controller* can cope adequately with the decrease in solid intake.

Since underflow solids concentration is the variable whose setpoint should be tracked more accurately, RF-MPC results vastly outperform all other strategies. PI and conventional MPC strategies fail to keep this CV undisturbed incurring in more than 0.6% of error. The RF-MPC, on the contrary, has a maximum deviation 10 times smaller while maintaining both torque and interface levels in check.

By observing figure 6.9f the main issue affecting the conventional strategy is repeated. As interface level drops rapidly, the predictive model embedded in the controller recreates the scenario in the underflow solids concentration tracking experiment. This is also the case for solid throughput increase, where now interface level decreases rapidly when the disturbances change, as shown in figure 6.9e.

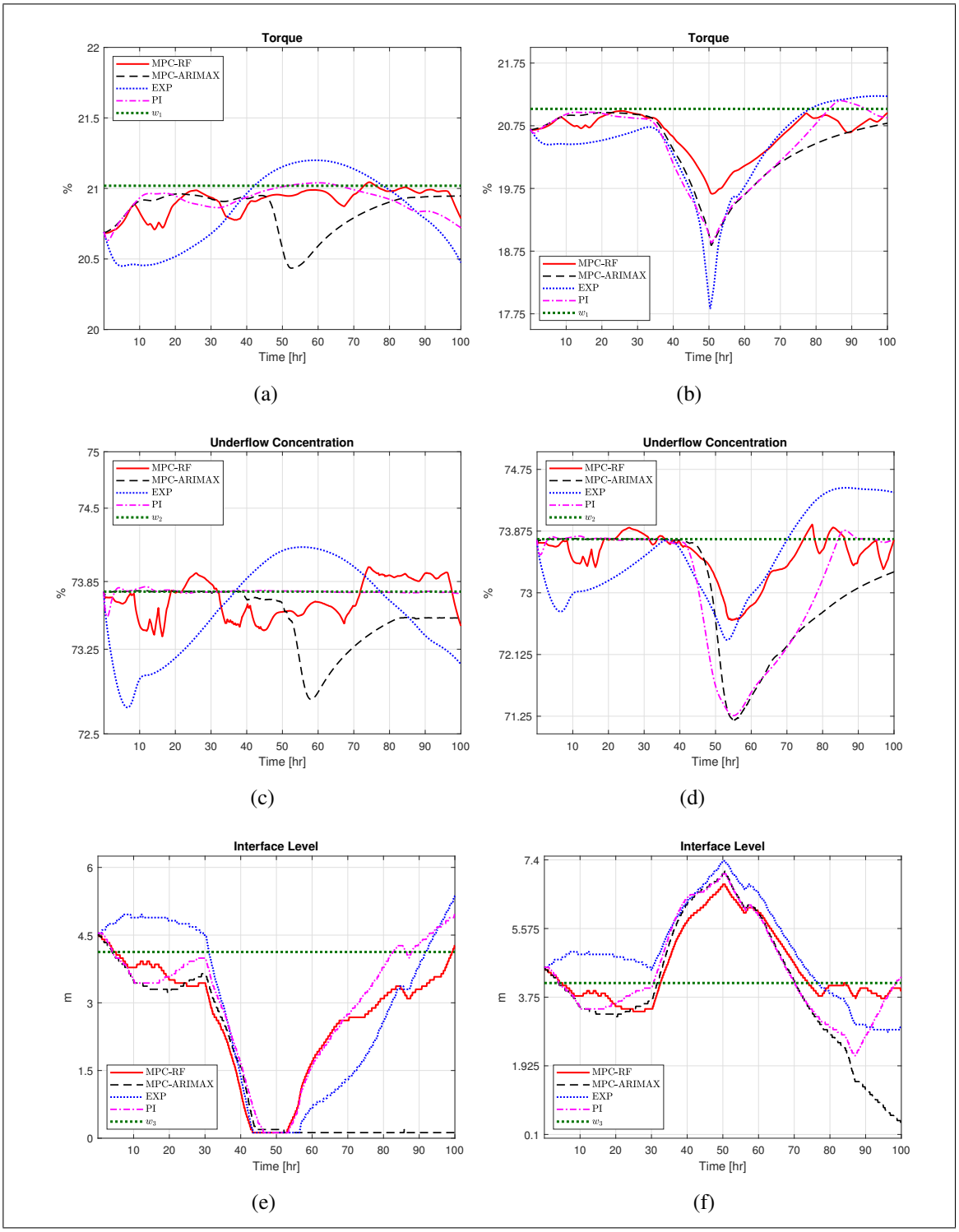


Figure 6.9. Evolution of controlled variables for increase (figures 6.9a, 6.9c, 6.9e) and decrease (figures 6.9b, 6.9d, 6.9f) in thickener solid throughput.

This issue validates the use of random forests instead of linear predictors. Even though the results of chapter 4 showed similar predictive performance, forest models appear to be much more controllable.

Figure 6.10 compares the control inputs for all controllers in both scenarios and illustrates said behaviour in depth.

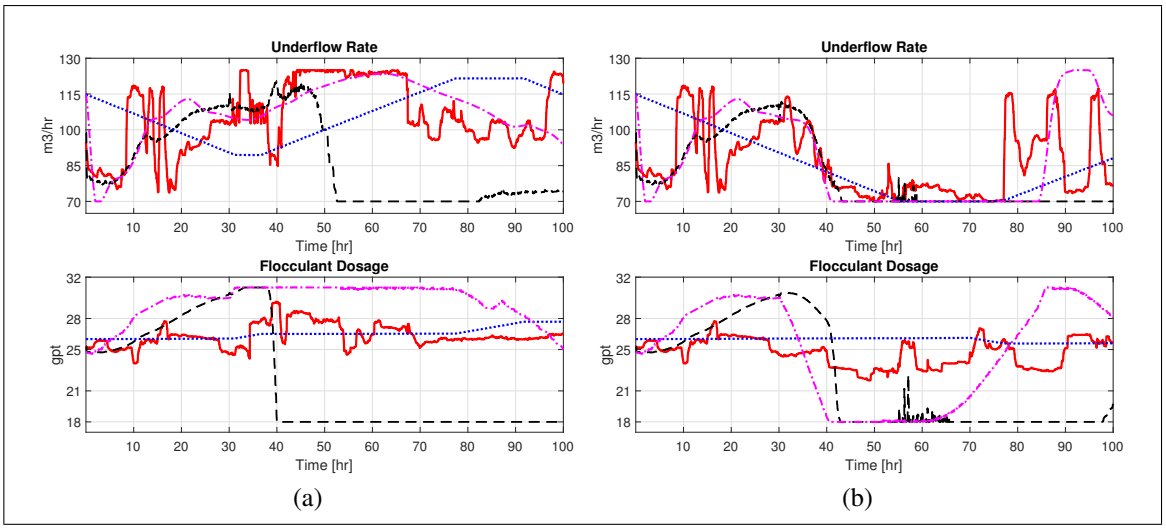


Figure 6.10. Manipulated variables calculated by all controllers for solid intake increase (figure 6.10a) and decrease (figure 6.10b). RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

Recalling that the disturbance change is applied 30 hours into operation, it can be seen that for the solid throughput *decrease* all controllers lower underflow rate to its minimum. As a matter of fact, the PI and predictive strategies take almost exactly the same amount of time to reach this value.

For solid throughput increase the optimal MV sequence is not a saturated control on the underflow rate. However, in both scenarios the conventional MPC controller saturates flocculant dosage and produces an unstable output. Again the benefits of the RF-MPC strategy are highlighted, since even though this controller cannot avoid interface collapse - as do all - it can restore operating conditions after the change in the DVs has passed.

Table 6.12 compares controller actions based on the quantitative criteria described earlier. Both non-optimal controllers have reduced SCE costs, while the predictive strategies are significantly more expensive regarding underflow rate specially. However, as mentioned in chapter 6, this effect can be handled through the costs in the objective function of both strategies.

Table 6.12. Comparison of control performance for decrease in solid throughput regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^0]$ | | **ISU** $[1 \times 10^6]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 3298.24 | 18.96 | 76.80 | 0.64 |
| MPC | 833.59 | 59.25 | 80.65 | 8.72 |
| Expert | 4.18 | 0.00 | 66.89 | 0.02 |
| PI | 36.58 | 6.96 | 138.32 | 8.86 |

The proposed random forest controller has the second lowest ISU score for both MVs. As explained before, it is desirable to keep MVs not saturated at their limits since controller response can be affected negatively. If another disturbance change were to occur or a setpoint changed in the simulation horizon, RF-MPC has more head room to react to it. As expert control is unsuccessful in the main control objectives, it is clear that the proposed approach is the best performing controller for this scenario.

Interface level is arguably the most difficult variable to control in this situations. On both of the solid intake changes, interface level violates the specified limits. Table 6.13 summarizes these results.

For the sudden solid intake scenario of table 6.13a, the MAE shows that all controllers fail to keep the interface far away from the feedwell. This is due to the fact that the disturbance changed when the interface was still close to the top of the thickener. Hence, the necessity of providing *correct setpoints* to paste production is emphasized.

Table 6.13. Interface level control metrics for solid intake changes for all controllers.

(a) Solid throughput increase.

| **Criterion** | **IAE** $[1 \times 10^4]$ | **MAE** $[1 \times 10^0]$ |
|---|---|---|
| RF-MPC | 90.32 | 5.00 |
| MPC | 112.19 | 4.93 |
| Expert | 63.83 | 4.86 |
| PI | 73.25 | 5.00 |

(b) Solid throughput decrease.

| **Criterion** | **IAE** $[1 \times 10^4]$ | **MAE** $[1 \times 10^0]$ |
|---|---|---|
| RF-MPC | 44.16 | 2.65 |
| MPC | 68.87 | 3.28 |
| Expert | 68.64 | 4.94 |
| PI | 47.39 | 2.65 |

For the opposite case, the RF-MPC is the best controller, together with PI feedback. However, no controller can ensure that the high limits of table 6.2 are not violated. The IAE criterion shows an approximate 20% less error in the entire simulation horizon. Even though interface is not a variable of primary interest in paste quality, the fact that the RF-MPC can stabilize underflow solids concentration *and* keep the solid inventory in check is one of its main advantages over the other strategies.

### 6.3.4. Overall Results and Final Remarks on Thickener Control

So far each controller test has been analyzed separately. To draw the final conclusions regarding the proposed predictive controller, an aggregate analysis of the scenarios will be provided.

Table 6.14 compares the MSE for all controllers averaged throughout all scenarios. All additional results and performance criterion can be found in appendix E.

Table 6.14. Average MSE throughout all scenarios for all controlled variables.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | | |
|---|---|---|---|
| **Variable** | Torque $T(t)$ | U.S.C. $C_u(t)$ | Interface Level $h(t)$ |
| RF-MPC | 7.97 | 5.28 | 171.10 |
| MPC | 17.19 | 7.77 | 291.21 |
| Expert | 24.45 | 13.23 | 176.65 |
| PI | 15.10 | 5.29 | 138.63 |

The summary provided leads to the conclusion that *the RF-MPC designed is the best alternative for thickener control*. However, further analysis is required of such a statement.

PI and conventional MPC achieve slightly worse results than the designed RF-MPC regarding torque and underflow solids concentration control. For underflow solids concentration - which impacts the most in paste quality - this means that on average the tracking error (squared root of the MSE) for these methods is of 0.23% and 0.27% respectively. Torque control is approximately two times more effective than the runner-up strategy and more than three times better than expert control.

The RF-MPC scores 0.23% in that metric for underflow solids concentration. This result is better than the linear predictive controller. In fact, all other criteria considered to asses the performance regarding process outputs control points to the same conclusion. It ranks second regarding interface level control and outperforms conventional MPC by almost 60%.

An interesting follow-up question is whether or not the *design and implementation costs* are balanced out by the performance of RF-MPC. PI control achieves very similar results, so it is only logical to doubt the necessity of a much more sophisticated technique.

The proposed method is certainly flawed in some aspects. The issues regarding optimality have been covered, but perhaps the most important shortcoming is that the random forest predictive models cannot fully represent thickener dynamics. As such, it is natural that less complex and more naive strategies produce similar results.

Additionally, this is a *simulation study*; as mentioned in chapter 3, paste production is increasingly difficult and PI-based solutions have not been able to adequately control this industrial process.

Lastly, as there are only 2 control inputs and three outputs, a PI feedback loop *cannot* deal with actively controlling a third output, which in this case corresponds to rake torque. The fact that this output is strongly coupled to underflow solids concentration improves

the performance of this alternative. Such a situation is seldom the case, so a more complex but robust approach is justified.

What is conclusive, however, is that the RF-MPC is *better* than a linear and conventional predictive control approach. Even though the latter deals with a convex optimization which can be solved through gradient based optimization methods, the proposed design is convincingly more adequate for this type of systems. Therefore, the main advantage of the RF-MPC comes from the *predictive model used*. Its combination with the MPC strategy is then viewed as successful, as the issues posed by using such a model - which is far from perfect - are not solved trivially.

Finally, table 6.15 summarizes the performance criteria that focus on the control inputs of the system. As expert control fails all tests in general, it can be disregarded entirely.

Table 6.15. Average control performance throughout all scenarios regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^0]$ | | **ISU** $[1 \times 10^6]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 3221.63 | 17.37 | 58.73 | 0.44 |
| MPC | 570.76 | 18.14 | 73.27 | 5.30 |
| Expert | 3.71 | 0.02 | 40.96 | 0.17 |
| PI | 71.85 | 5.40 | 80.89 | 3.66 |

It is quite clear from the SCE in table 6.15 that the designed strategy strains underflow rate excessively when compared to non optimal techniques. However, *this was deliberately chosen* and designed. If a higher cost had been specified for manipulated variables effort, results would change consistently.

Such is the case for flocculant dosage, in which RF-MPC ranks *second*. As mentioned previously, even though the controller designed had no explicit optimization objectives, the cost specified for flocculant addition is 10 times larger than underflow rate. Some sense of optimality is included in this decision: flocculant addition translates to *operational*

*expenses* since more of this resource must be bought if misused. Pumps do not suffer these consequences in the short run, so preferring its use over flocculant *is* an improvement.

ISU criteria clearly shows that expert control is the better alternative. However, given that this method ranks third or last regarding output control, RF-MPC is deemed as the best alternative for thickener control. By keeping MVs far from their limits as much as possible, the controller is prepared for unexpected setpoint or disturbance changes, such as the ones explored in section 6.3.3.

With these remarks, the research can be concluded. This entire document is a blueprint for data-driven predictive controller design: from system identification through operational data to the final software implementation of the controller. The next chapter revisits the hypotheses stated at the very beginning of this investigation and evaluates the extent to which the objectives specified in chapter 1 have been covered.

# 7. CONCLUSIONS

At the beginning of this research, the importance of paste production control was stated as an ongoing concern in the mining industry. Given this problem, the proposition was to use state of the art data-driven techniques in combination with advanced control strategies for its regulation and possible optimization.

The main research hypothesis was that the integration of Model Predictive Control and Machine Learning algorithms using only operational data can control and stabilize the paste production process of a complex dynamical thickener simulator. The body of work presented in this document confirms this with sufficiently high experimental evidence. This is the first contribution of the work portrayed in this document.

The general objective of this investigation was developing a technique that integrates MPC and Machine Learning to control the paste tailing production. As mentioned previously, the random forest model predictive controller designed and tested in this research has not been found elsewhere and therefore represents a contribution to the scientific community. Hence, this objective has been accomplished.

Furthermore, a general purpose toolbox in the Matlab-Simulink environment has been produced as a result of this investigation. It has been designed to fit any problem regardless of dimensions or complexity. This software is the first of its kind in the scientific community and as such is the most tangible contribution.

The different chapters of this document provided key insights into all the issues involved with a data-driven predictive controller. The first part of this research was concerned with delimiting the paste production process and its control, as well as providing a background for the proposed method and its importance. In this section, a detailed and up to date revision of thickener control techniques was provided, which was another objective of this research.

The middle section of this research explored the random forest technique and proposed its application to time series forecasting and prediction. Its application was examined for the thickener system in a pseudo-real environment and contrasted to a benchmark technique. Furthermore, an overview of Machine Learning and Model Predictive Control state of the art techniques was offered.

By doing all of the above, tangential objectives - specific to the system identification and forecasting task - were covered. Incidentally, the results obtained contributed to a deeper understanding of the difficulties associated with thickener modeling and control, mainly the auto-regressive and inertia effects of paste production.

System identification results showed none of the methods chosen can guarantee a completely trustworthy dynamic characterization of the thickener studied. The main issue in thickener prediction and control is its *highly auto-regressive* behaviour. The random forest strategy should be re-evaluated for time series prediction or at least in its application on thickener operation.

Finally, the last chapters dealt with the combination of all the explored topics in a predictive controller. All issues regarding its implementation were covered, solved and finally tested through various simulation scenarios. A detailed comparison with three other benchmark control methods was examined, one of which is of a similar nature to the controller proposed in this research. The conclusions drawn from this study support the claim that a data-driven control strategy is useful and preferable for the control of sophisticated and complex industrial processes such as paste production.

Specifically, the proposed approach performs better than all other strategies in setpoint tracking and disturbance rejection. The integration of both data-driven approaches - random forests and particle swarm optimization - can adequately control thickener operation in the framework encompassed in this investigation.

## 7.1. Transfer to a Real Thickener Operation

Before moving on to possible areas of future research that stem from this investigation, some recommendations need to be given for the transfer of this strategy to a *real* operation.

A first step would be combining specific mineral and metallurgic knowledge with the data-driven techniques explored in this research. Unnecessary variables can be eliminated from the analysis and hence both computational and engineering efforts can be reduced significantly.

Special attention needs to be paid to the *online optimization* issues that arise from the use of non-linear and non-smooth predictive functions. Almost all state of the art Machine Learning techniques fall under this category and thus require a detailed cost-benefit analysis.

Real time constraints can severely impact the benefits from a data-driven predictive controller such as the one designed. *However*, the results provided were computationally feasible. The optimization problem was solved in *seconds*, which was a deliberate decision.

The design strategy proposed in this investigation has a *modular* quality. As such, any of the blocks designed in chapter 5 can be replaced by better of more efficient implementations. It is clear that perhaps the most benefit can be harnessed from a more accurate *predictive model*.

As mentioned, the random forest strategy was a good and reasonable choice for thickener modeling given the initial hypotheses. Nevertheless, results showed that forests are very sensitive to highly auto-regressive systems. If any model can fit the data more accurately, then it is reasonable to assume that controller performance can only improve.

One final recommendation for real application of this strategy can be given. The prediction step time was fixed in 5 minute throughout this research. It is *highly recommended*

that this decision is reevaluated. By using a smaller forecasting resolution, perhaps better results can be obtained for pure prediction and also control purposes.

Moreover, different time steps can be used for underflow rate and flocculant dosage, since they have radically different time constants.

## 7.2. Future Research and Possible Directions

There are five main research topics that arise from this investigation. The first one is, naturally, *real* thickener control. Even though the simulator represents thickener dynamics, it can only do so to some extent. The next step in this investigation is the transfer of the proposed design to a real and online control scheme.

Secondly, more research should be done in understanding if *any* Machine Learning algorithm can thoroughly represent thickener dynamics. The main challenge, as stated previously, arises from the fact that this system is highly auto-regressive. Moreover, it was seen that the random forest approach was biased by the residence time of the thickener. It is unclear if any form of Statistical Learning procedure can overcome this issue.

Regarding random forests themselves, there are two very interesting research topics that have not been covered in this investigation. The first one is centered on a *pseudo-derivative* of a forest or tree. If such a representation can be obtained, then the choice of an evolutionary algorithm can be contrasted with other optimization methods, such as SQP for example. It would also be interesting to estimate and quantify if controller performance decreases because of this approximation and if so by how much.

The second research direction, and perhaps more interesting from the data science and Machine Learning perspective, is using random forests for *reinforced learning*. This type of approaches have seen a radical increase in popularity because of computational capacities and impressive results over supervised strategies. There is some research into

this topic (Zhou, Guo, & Li, 2016), but for time series prediction there are no apparent efforts in these methods .

Finally, more conventional applications of the designed controller and toolbox can be explored. By doing this, the full potential of the strategy can be studied through its application on other problems. In that same regard, the transfer of the proposed strategy to other nonlinear industrial control problems is also of great research interest.

# REFERENCES


Aghajani Shahrivar, A., Goharrizi Ataallah, S., Ebrahimzadeh, M., Sarafi, A., Mohammad, R., & Hadi, A. (2013, 09). Application of response surface methodology and central composite rotatable design for modeling the influence of some operating variables of the lab scale thickener performance. *International Journal of Mining Science and Technology*, *23*, 717-724.

Alvarez, T., Tadeo, F., & Grimble, M. J. (2002, Sept). Tuning of predictive controller using performance assessment measures: application to ph control. In *Proceedings of the International Conference on Control Applications* (Vol. 1, p. 403-408 vol.1). doi: 10.1109/CCA.2002.1040220

Astrom, K. J., & Murray, R. (2012). Feedback fundamentals. In (p. 293-314). Princeton University Press.

Betancourt, F., Burger, R., Diehl, S., & Faras, S. (2014). Modeling and controlling clarifier-thickeners fed by suspensions with time-dependent properties. *Minerals Engineering*, *62*, 91 - 101. Retrieved from http://www.sciencedirect.com/science/article/pii/S0892687513003968 (SI: Physical Separation) doi: https://doi.org/10.1016/j.mineng.2013.12.011

Bontempi, G., Ben Taieb, S., & Le Borgne, Y.-A. (2013). Machine Learning Strategies for Time Series Forecasting. In M.-A. Aufaure & E. Zimányi (Eds.), *Business Intelligence: Second European Summer School, eBISS 2012, Brussels, Belgium, july 15-21, 2012, tutorial lectures* (pp. 62–77). Berlin, Heidelberg: Springer Berlin Heidelberg. Retrieved from https://doi.org/10.1007/978-3-642-36318-4_3 doi: 10.1007/978-3-642-36318-4_3

Branke, J., Kaubler, T., & Schmeck, H. (2001). Guidance in evolutionary multi-objective optimization. *Advances in Engineering Software*, *32*(6), 499 - 507. Retrieved from http://www.sciencedirect.com/science/article/pii/S0965997800001101 doi: https://doi.org/10.1016/S0965-9978(00)00110-1

Breiman, L. (2001, Oct 01). Random forests. *Machine Learning*, *45*(1), 5–32. Retrieved from https://doi.org/10.1023/A:1010933404324 doi: 10.1023/A:1010933404324

Breiman, L., Freidman, J., Olshen, Noack, R., & Stone, R. (1984). *Classification and Regression Trees*. Chapman-Hall-CRC.

Burger, R., Karlseb, H., & Towers, J. (2005). A model of continous sedimentation of flocculated suspensions in clarifier-thickener units. *Society for Industrial and Applied Mathematics*, *65*, 882-940.

Burger, R., Karlsen, K., & Towers, J. (2005). Mathematical model and numerical simulation of the dynamics of flocculated suspensions in clarifier-thickeners. *Chemical Engineering Journal*, *111*.

Cacciuttolo, C., & Holgado, A. (2016). Management of paste tailings in Chile: A review of practical experience and environmental Acceptance. In *Proceedings of the 19th International Seminar on Paste and Thickened Tailings* (p. 121-136). Santiago.

Camacho, E., & Bordons, C. (2007). *Model predictive control* (2nd ed., Vol. 1). Springer.

Canfield, M. (2012, June). *Mining production stages.*

Cao, A. R., Zhou, B. H., & Hou, C. Z. (2009, Nov). The model-free direct adaptive generalized predictive control approach of permanent magnet linear motor. In *2009 international conference on electrical machines and systems* (p. 1-5). doi: 10.1109/ICEMS.2009.5382669

Castillo, E., Guitérrez, J. M., & Hadi, A. S. (1998). *Expert systems and probability network models* (1st ed.). Spanish Engineering Academy.

Chai, T., Jia, Y., Li, H., & Wang, H. (2016). An intelligent switching control for a mixed separation thickener process. *Control Engineering Practice*, *57*, 61 - 71. Retrieved from http://www.sciencedirect.com/science/article/pii/S096706611630154X doi: https://doi.org/10.1016/j.conengprac.2016.07.007

COCHILCO. (2014). *Water consumption projections in the copper mining industry for 2014-2025.* Retrieved from http://www.mch.cl/wp-content/uploads/sites/4/2015/01/Informe-Proyeccion-consumo-de-agua.pdf

COCHILCO. (2015). *Investment in the chilean mining industry - project catalogue 2015 - 2024.* Retrieved from http://www.mch.cl/wp-content/uploads/sites/4/2015/08/Carpeta-Inversiones-Mineras-2015-2024.pdf

Concha, F. (2014). *Solid-liquid separation in the mining industry*. Springer.

Dahleh, M. (2005, Spring). *6.435 System identification.* Massachusetts Institute of Technology: MIT OpenCourseWare. Retrieved from https://ocw.mit.edu

Efron, B. (1979). Bootstrap methods: another look at the jackknife. *Annals of Statistics*, *7*(1), 1 - 26. Retrieved from http://www.sciencedirect.com/science/article/pii/S002200009791504X doi: https://doi.org/10.1006/jcss.1997.1504

Espinace, R., Villavicencio, G., & Fourie, A. (2016). Thickened tailings disposal technology experiences in chile. In S. Barrera & R. Jewell (Eds.), *Proceedings of the 19th International Seminar on Paste and Thickened Tailings* (p. 527-598). Gecamin Chile.

Fitton, T. (2016). Thickened tailings discharge: Spigot spacing and the merging of streams. In S. Barrera & R. Jewell (Eds.), *Proceedings of the 19th International Seminar on Paste and Thickened Tailings* (p. 343-355). Gecamin Chile.

Grune, J., L.; Pannek. (2017). *Nonlinear model predictive control-theory and algorithms-second edition*. Springer.

Hastie, T., Tibshirani, R., & Friedman, J. (2009). *Elements of statistical learning* (2nd ed.). Springer.

Heaton, J. (2015). *Artificial intelligence for humans, volume 3: Deep learning and neural networks* (1st ed.). CreateSpace Independent Publishing Platform.

Hock, W., & Schittkowski, K. (1983). A comparative performance evaluation of 27 nonlinear programming codes. *Computing*, *30*, 335.

Hou, Z., Gao, H., & Lewis, F. (2017). Data-driven control and learning systems. *IEE Transactions on Industrial Electronics*, *64*(5), 4070-4075.

Hou, Z., & Jin, S. (2014). *Model free adaptative control-theory and applications*. CRC Press-Taylor and Francis Group.

Hou, Z., Liu, S., & Tian, T. (2017, Aug). Lazy-learning-based data-driven model-free adaptive predictive control for a class of discrete-time nonlinear systems. *IEEE*

*Transactions on Neural Networks and Learning Systems*, *28*(8), 1914-1928. doi: 10.1109/TNNLS.2016.2561702

Jain, A., Behl, M., & Mangharam, R. (2017, May). Data predictive control for building energy management. In *2017 American Control Conference (ACC)* (p. 44-49). doi: 10.23919/ACC.2017.7962928

Jain, A., Mangharam, R., & Behl, M. (2016). Data predictive control for peak power reduction. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments* (pp. 109–118). New York, NY, USA: ACM. Retrieved from http://doi.acm.org/10.1145/2993422.2993582 doi: 10.1145/2993422.2993582

Jain, A., Smarra, F., & Mangharam, R. (2017, Dec). Data predictive control using regression trees and ensemble learning. In *56th Annual Conference on Decision and Control (CDC)* (p. 4446-4451). doi: 10.1109/CDC.2017.8264315

Jewell, R., & Fourie, A. (2015). Paste and thickened tailings - a guide. *Australian Center for Geomechanics*, *24*, 1005-1023.

Jin, S., Hou, Z., & Chi, R. (2013, July). Data-driven predictive terminal iterative learning control for a class of discrete-time nonlinear systems. In *Proceedings of the 32nd chinese control conference* (p. 2992-2996).

Kamesh, R., & Rani, K. Y. (2017, Dec). Novel formulation of adaptive mpc as ekf using ann model: Multiproduct semibatch polymerization reactor case study. *IEEE Transactions on Neural Networks and Learning Systems*, *28*(12), 3061-3073. doi: 10.1109/TNNLS.2016.2614878

Kim, H. H., Kim, D. G., Choi, J. Y., & Park, S. C. (2017). Tire mixing process scheduling using particle swarm optimization. *Computers and Industrial Engineering*, *110*, 333 - 343. Retrieved from http://www.sciencedirect.com/science/article/pii/S0360835217302644 doi: https://doi.org/10.1016/j.cie.2017.06.012

Langlois, J. (2018). *Development and Implementation of a Dynamic Simulator for the Paste Tailing Production Process* (Master of Science Thesis). Pontificia Universidad Católica de Chile.

Ljung, L. (1999). *System identification theory for the user* (1st ed., Vol. 2). Prentice Hall.

Marini, F., & Walczak, B. (2015). Particle swarm optimization (pso). a tutorial. *Chemometrics and Intelligent Laboratory Systems*, *149*, 153 - 165. Retrieved from http://www.sciencedirect.com/science/article/pii/S0169743915002117 doi: https://doi.org/10.1016/j.chemolab.2015.08.020

Moller, P., Fall, A., Chikadi, V., Didi, D., & Bonn, D. (2009). An attempt to categorize yield stress fluid behaviour. *Philosphical Transactions of The Royal Society*, *367*.

Ojeda, P., Bergh, L., & Torres, L. (2014, December). Intelligent Control of an Industrial Thickener. In *13th International Conference on Control Automation Robotics and Vision.*

Pellegrinetti, G., & Bentsman, J. (1996, Jan). Nonlinear control oriented boiler modeling-a benchmark problem for controller design. *IEEE Transactions on Control Systems Technology*, *4*(1), 57-64. doi: 10.1109/87.481767

Qin, S. J., & Bagdwell, T. (2003, Sept). A survey of model predictive control technology. *Control Engineering Practice*, *11*(5), 733-764.

Quelopana, H. (2016). Tailings beach slope forecasting: An empirical model. In S. Barrera & R. Jewell (Eds.), *Proceedings of the 19th International Seminar on Paste and Thickened Tailings* (p. 361-378). Gecamin Chile.

Rawlings, J., & Mayne, D. (2012). *Model Predictive Control - Theory and Design* (2nd ed.). Nob Hill Publishing.

Rudman, M., Paterson, D., & Simic, K. (2010). Efficiency of raking in gravity thickeners. *International Journal of Mineral Processing*, *95*(1), 30 - 39. Retrieved from http://www.sciencedirect.com/science/article/pii/S0301751610000438 doi: https://doi.org/10.1016/j.minpro.2010.03.007

Russell, S. J., & Norvig, P. (2015). *Artificial intelligence: A modern approach* (3rd ed.). Pearson.

Schmid, C., & Biegler, L. (1994). Quadratic programming methods for reduced hessian sqp. *Computers and Chemical Engineering*, *18*, 817-832.

Segovia, J. P., Concha, F., & Sbarbaro, D. (2011). On the Control of Sludge Level and Underflow Concentration in Industrial Thickeners. In *Proceedings of the 18th World*

*Congress of The International Federation of Automatic Control.*

Sernageomin. (2018). *Analysis of Tailing Storage Facilities in Chile and Data Structure Guide.* Retrieved from http://www.sernageomin.cl/wp-content/uploads/2018/05/Ana1lisis-de-los-Depositos-de-Relaves-en-Chile.pdf

Shridhar, R., & Cooper, D. J. (1997). A novel tuning strategy for multivariable model predictive control. *ISA Transactions*, *36*(4), 273 - 280. Retrieved from http://www.sciencedirect.com/science/article/pii/S0019057897000360 doi: https://doi.org/10.1016/S0019-0578(97)00036-0

Stromberg, K. (2016). Thickened tailings management, a dynamic process: Understanding and optimizing the thickener operation. In S. Barrera & R. Jewell (Eds.), *Proceedings of the 19th International Seminar on Paste and Thickened Tailings* (p. 31-43). Gecamin Chile.

Tan, C. K., Bao, J., & Bickert, G. (2017). A study on model predictive control in paste thickeners with rake torque constraint. *Minerals Engineering*, *105*, 52 - 62. Retrieved from http://www.sciencedirect.com/science/article/pii/S0892687517300213 doi: https://doi.org/10.1016/j.mineng.2017.01.011

Tan, C. K., Setiawan, R., Bao, J., & Bickert, G. (2015). Studies on parameter estimation and model predictive control of paste thickeners. *Journal of Process Control*, *28*, 1 - 8. Retrieved from http://www.sciencedirect.com/science/article/pii/S0959152415000293 doi: https://doi.org/10.1016/j.jprocont.2015.02.002

Tan, C. K., Tippett, M. J., & Bao, J. (2016). Model predictive control with non-uniformly spaced optimization horizon for multi-timescale processes. *Computers and Chemical Engineering*, *84*, 162 - 170. Retrieved from http://www.sciencedirect.com/science/article/pii/S0098135415002719 doi: https://doi.org/10.1016/j.compchemeng.2015.08.010

The MathWorks, I. (2018). Global optimization toolbox: User's guide [Computer software manual].

Wu, A., Wang, Y., & Wang, H. (2015). Estimation model for yield stress of fresh uncemented thickener tailings: Coupled effects of true solid density, bulk density, and solid concentration. *International Journal of Mineral Processing*, *143*, 117-124.

Xu, N., Wang, X., Zhou, J., Wang, Q., Fang, W., & Peng, X. (2015). An intelligent control strategy for thickening process. *International Journal of Mineral Processing*, *142*, 56 - 62. Retrieved from `http://www.sciencedirect.com/science/article/pii/S0301751615000708` (Mineral Processing in Australia and China) doi: https://doi.org/10.1016/j.minpro.2015.01.007

Zeroday. (2015). *Thickener operation.*

Zhou, M., Guo, Z.-Q., & Li, X. (2016, Sept). Design of model predictive control for time-varying nonlinear system based on gaussian process regression modeling. In *21st International Conference on Emerging Technologies and Factory Automation (ETFA)* (p. 1-6). doi: 10.1109/ETFA.2016.7733737

# APPENDIX

# A. SIMULATION OF THE THICKENING PROCESS

## A.1. Thickening Process

Thickening is the superposition of the phenomenon of sedimentation and consolidation simultaneously. The first refers to the precipitation of solid particles individually by the effect of collisions between them and in the fluid by pressure and friction. For a specific concentration of the suspension, the particles come into contact and give rise to a mesh called sediment (Langlois, 2018).

In compressible particles, such as tailing after a flocculation process, the weight of the sediment compresses the particles of the lower layers and fluid is squeezed through pores in the mesh. This last process corresponds to consolidation.

The main equation governing the thickening process is given by:

$$\frac{\partial \phi}{\partial t} = \frac{\partial}{\partial z}\left(a(\phi)\frac{\partial \phi}{\partial z} - \mathbf{F}(\phi, q)\right) \tag{A.1}$$

where $\phi = \phi(z, t)$ is the concentration of solids in the thickener at depth $z$ and in time $t$, $a(\phi)$ is a diffusion term for solids compression, $q$ is the average velocity in the vertical direction and $\mathbf{F}$ is a convective term for the mixture.

Equation A.1 depicts that the evolution of concentration along the thickener depends on the difference between the longitudinal change of concentration a convection or flow function.

On the other hand, a simple continuity equation relates all entering and exiting flows in the thickener:

$$Q_f = Q_u + Q_e \tag{A.2}$$

for $Q_f$, $Q_e$ and $Q_u$ the feed, effluent and underflow rate respectively.

## A.2. Parameter Specification for the Thickening Simulator

Figure A.1 shows all the dimensional parameters involved in the physical instantiaton of the thickener. Thickening takes place in $z \in [0, B]$ whereas the clear zone corresponds to $z \in [-(H - \Delta_F), 0]$ where $\Delta_F$ is a distance that separates the feeding pipes from the clear region. The thickener's maximum area from its top view is of $2R$. The feedwell mechanism is represented by a rigid cross-section of width $2R_2$ and the discharge pipe is of diameter $2R_1$.

The rake mechanism interacts with the slurry in the region defined between $z_1$ and $z_2$. In this zone, the cross-sectional area of the thickener diminishes constantly through the slope angle $\varphi$.

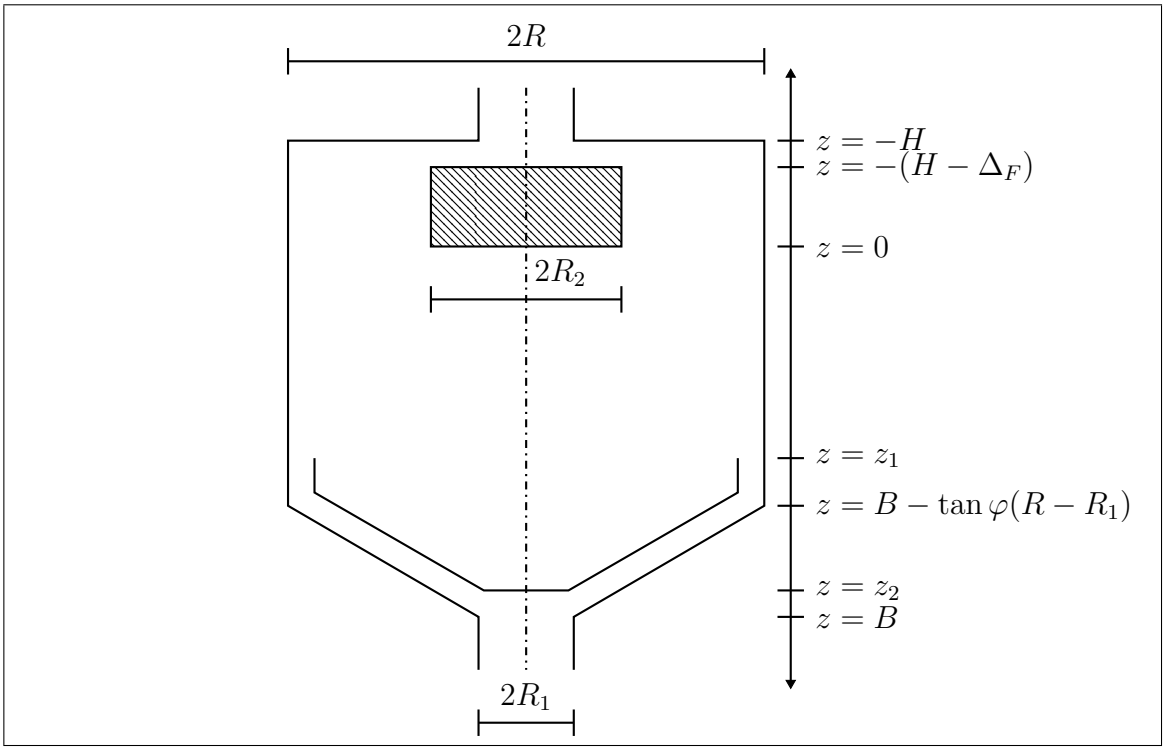


Figure A.1. Diagram of thickener and dimensional parameters (Langlois, 2018)

The geometrical parameters, values and units are specified in table and represent the actual thickener being modeled.

Table A.1. Geometrical parameter specification for thickening simulator

| Parameter | Value | Unit |
|---|---|---|
| $B$ | 12 | m |
| $H$ | 2.5 | m |
| $\Delta_F$ | 0.5 | m |
| $R$ | 8.5 | m |
| $R_1$ | 1.8 | m |
| $R_2$ | 3.6 | m |
| $\varphi$ | 30 | $^\circ$ |
| $z_1$ | 7 | m |
| $z_2$ | 12 | m |

## A.3. Flocculation

The predominant flocculation process in the mining industry is the so-called *bridging floculation* or bridge flocculation. Polymer chains corresponding to the flocculant are absorbed by various particles forming a fractal structure or *floc* that has a larger equivalent diameter as a whole particle.

The effect of the flocculant dosage $c$ (in gpt) on the sedimentation rate is given by a concave and unimodal function $v_{st}(c)$ which can be determined through experimentation (Concha, 2014). This relationship can be established through a dimensionless variable $k(z,t) \in [0,1]$ that impacts the initial precipitation velocity (Langlois, 2018).

This implies that the following function of the $k$ property is standardized with respect to the maximum theoretical sedimentation rate $v_0$:

$$\bar{k}(c) = \frac{v_{st}(c)}{v_0} \tag{A.3}$$

for the range $0 < c < c^*$ where $c^*$ is the optimum flocculant dose. It makes sense to consider this range - where therefore $v_{st}(c)$ is increasing and injective - as it is not convenient to surpass $c^*$.

In the paste production process, flocculant dosage is usually fixed during the whole operation. A setpoint is calculated for steady-state operation considering thickener design

parameters and nominal operation of the tailing treatment plant. This setpoint depends on the initial concentration of the flocculant solution $G_F$ and the total *solid feed rate* $Q_s(t)$. Almost no industrial experiences can be found of varying this value throughout the operation.

## A.4. Rake Torque Model

Empirical evidence shows a linear relationship between torque and yield stress in the underflow. Since this property varies with time and no actual instrumentation exists to measure it, a derivation exists relating torque and underflow solids concentration.

The model considers solids content in the thickener region $[z_1, z_2]$ where the rake operates by defining a compound solids concentration variable $\phi_T(t)$. The used model discretizes the region $[z_1, z_2]$ in cells $\mathbf{Z_T} = [z_1, z_2, \cdots, z_T]$ and therefore calculate this composite property $\phi_T$ as (Rudman et al., 2010):

$$\phi_T = \sum_{j \in \mathbf{Z_T}} C_j \phi_j(t) \tag{A.4}$$

Thus, it is determined that the rake torque is a function of the concentration $\phi_T$:

$$T(\phi_T(t)) = a_1 \left(\phi_T(t)\right)^{a_2} + a_3 = a_1 \left( \sum_{j \in \mathbf{Z_T}} C_j \phi_j(t) \right)^{a_2} + a_3 \tag{A.5}$$

with $a_i$ parameters set for each particular paste thickener.

The rake torque increases monotonically with the composite solids concentration $\phi_T$ which in turn is affected by soldis concentration $\phi(z) \in [z_1, z_2]$. Hence, it is possible hypothetically to maintain torque levels within tight operation windows by controlling underflow solid concentration because of its linear effect on $\phi_T$ (Tan et al., 2017).

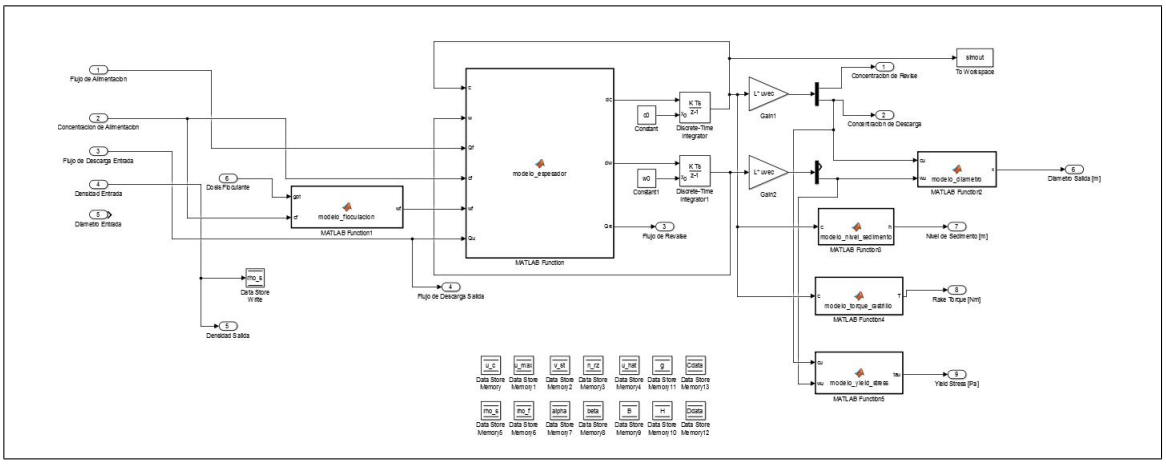


Figure A.2. Simulink thickener block diagram. The different modules used for calculating the process outputs can be seen. The equations mentioned in section A.1 are implemented in the center block.

## A.5. Model Implementation

In (Langlois, 2018) the numerical implementation of the above equations and the specific method used for process simulation is specified. Since thickening is physically a continuous process, computer simulation requires discretization both spatially and temporally by $\Delta_t$ and $\Delta_z$ respectively. The value of this parameters need to satisfy certain conditions to ensure the convergence of the simulation routines. These are specified in the referenced document. Figure A.2 shows the block diagram in Simulink of the thickener simulator.

# B. MATHEMATICAL BACKGROUND FOR RANDOM FORESTS

## B.1. Regression Trees

**Definition B.1.** *A regression tree uses the predictor function:*

$$f(x) = \sum_{m=1}^{\mathcal{M}} c_m \boldsymbol{I}\left(x \in R_m\right) \tag{B.1}$$

*for a constant $c_m$ that represents the region $R_m$ to be determined.*

$I\,(x \in R_m)$ is an indicator function which is equal to 1 if $x_i \in R_m$ and 0 in any other case.

If the criterion to be minimized is the sum of squared errors, then the following statement holds:

PROPERTY B.1. *For loss function*

$$L(y, f(x)) = \sum_{i=1}^{N} (y_i - f(x_i))^2$$

*the $\hat{c}_m$ which minimizes prediction error is*

$$\hat{c}_m = \frac{1}{N_m} \sum_{i=1}^{N_m} y_i I\,(x \in R_m) \tag{B.2}$$

*for $N_m$ the number of samples which satisfy $I\,(x \in R_m) = 1$.*

Considering a splitting variable $j \in X$ and a split point $s$, the half-planes

$$R_1(j, s) = \{X | X_j \leq s\} \ \wedge \ R_2(j, s) = \{X | X_j > s\}$$

are defined.

The pair $(j, s)$ are sought through the minimization of

$$\min_{j,s} \left\{ \min_{c_1} \sum_{x_i \in R_1(j,s)} (y_i - c_i)^2 + \min_{c_2} \sum_{x_i \in R_2(j,s)} (y_i - c_i)^2 \right\} \tag{B.3}$$

For any pair $(j, s)$, the inner minimization is solved by:

$$\hat{c}_1 = \frac{1}{N_1} \sum_{i=1}^{N_1} y_i I\,(x \in R_1(j, s)) \ \wedge \hat{c}_2 = \frac{1}{N_2} \sum_{i=1}^{N_2} y_i I\,(x \in R_2(j, s)) \tag{B.4}$$

### B.2. Bootstrapping and Bagging

Assuming that the training observations are independently drawn from an underlying distribution $\mathcal{P}$, the bagged estimate introduced in definition 4.3 can be interpreted as the expectation of the statistic $\hat{f}(x)$ over the probability distribution $\mathcal{P}$, that is,

$$f_{\text{bag}} = \mathbb{E}_{\mathcal{P}} \left\{ \hat{f}(x) \right\} \tag{B.5}$$

For squared error loss, the expectation analysis yields

$$\begin{aligned} \mathbb{E}_{\mathcal{P}} \left\{ \left( Y - \hat{f}(x) \right)^2 \right\} &= \mathbb{E}_{\mathcal{P}} \left\{ \left( Y - f_{\text{bag}}(x) + f_{\text{bag}}(x) - \hat{f}(x) \right)^2 \right\} \\ &= \mathbb{E}_{\mathcal{P}} \left\{ \left( Y - f_{\text{bag}(x)} \right)^2 \right\} + \mathbb{E}_{\mathcal{P}} \left\{ \left( f_{\text{bag}}(x) - \hat{f}(x) \right)^2 \right\} \\ &\geq \mathbb{E}_{\mathcal{P}} \left\{ \left( Y - f_{\text{bag}(x)} \right)^2 \right\} \end{aligned} \tag{B.6}$$

which proves that true population aggregation never increases mean squared error.

## C. DATA PROCESSING

### C.1. Data Preprocessing

(i) Missing (NaN) values where replaced by extrapolating the last known value of the variable.

(ii) Outlier detection was accomplished through a moving median filter. For each variable, an outlier threshold was specified using deviation statistics of the sample. The window size was 60 samples to encapsulate an entire minute of operation.

(iii) Outlier data was replaced with a linear interpolation between the two adjacent values.

(iv) The first approach to smoothing was done by means of gaussian envelope filtering in a 60 sample window.

(v) A butterworth low order filter was used for a second smoothing stage. The filter's order ranges from 4 to 6 and has a transient phase at the beginning. This first 6 seconds of data were discarded.

Frequency analysis of the output and input variables was conducted to estimate the passband frequency of system. Figure C.1 plots the FFT of both the CV and MV included in the operational data. No DC offset correction has been made, therefore the difference in signal power along the frequency axis is of importance. Since the system's dynamics are slow, almost all signal information is concentrated at low frequencies. Past 0.5 mHz the SNR drops ate least 20 dB from the passband. Frequencies above this threshold were filtered with a low ripple IIR filter.

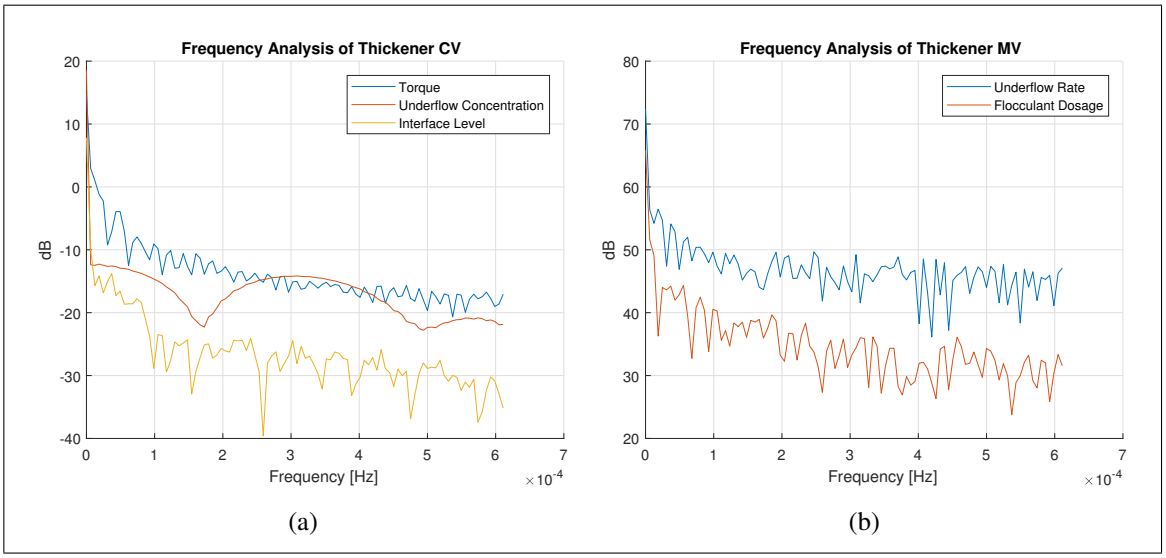


Figure C.1. FFT of thickener operational data. Since the system is over-sampled by $T_s$, an important amount of the frequency spectrum is pure noise. The figures illustrate that above 0.4 mHz there is little signal power left in the spectrum.

The last stage of data preprocessing was decreasing the sampling rate from $T_s$ to $\tau_R$. This was done through an averaging filter with a non-overlapping window of 300 samples (5 minutes). In other words, a new data sample was generated every 300 samples of the original data set.

# D. ADDITIONAL SYSTEM IDENTIFICATION RESULTS

## D.1. Application to Standard System

To verify that the random forest algorithm designed operates successfully, various tests were done on a standardized boiler system used for nonlinear system identification and control (Pellegrinetti & Bentsman, 1996).

The system consists of 3 CVs: oxygen level, water level and steam pressure. There are three MVs, namely supplied fuel, air inflow and water inflow. Finally, a DV is present in the system through a steam demand signal. As such, this MIMO system consists an ideal test plant for all algorithms tested, including early stages of the controller designed.

As this system consists purely of a simulator, a framework similar to the thickener problem was created. To do this, the following procedure was applied:

(i) Input signals were applied to the system in order to collect input-output data. These signals consisted on square, sine, sawtooth and step changes in the various inputs, applied simultaneously and one at a time.

(ii) The collected data was divided into training and testing set. Data produced by the square inputs was left for testing, while step and sinewave data was used for training.

(iii) Different random forest models were fit and the best was selected through a greedy algorithm according to lowest MSE.

Some of the input-output pairs have transmission zeros and therefore not all inputs generate affect all CVs (Pellegrinetti & Bentsman, 1996). For all test scenarios, *only one* input variable (MVs and DVs) was changed while the others remained constant. This allows for accurate assessment of the predictive capability of the random forest.

The following figures depict the results of these tests for a one-step ahead prediction. It is worth mentioning that these results showed encouraging outcomes and therefore provided sufficient proof for the use of random forests in nonlinear time series forecasting.

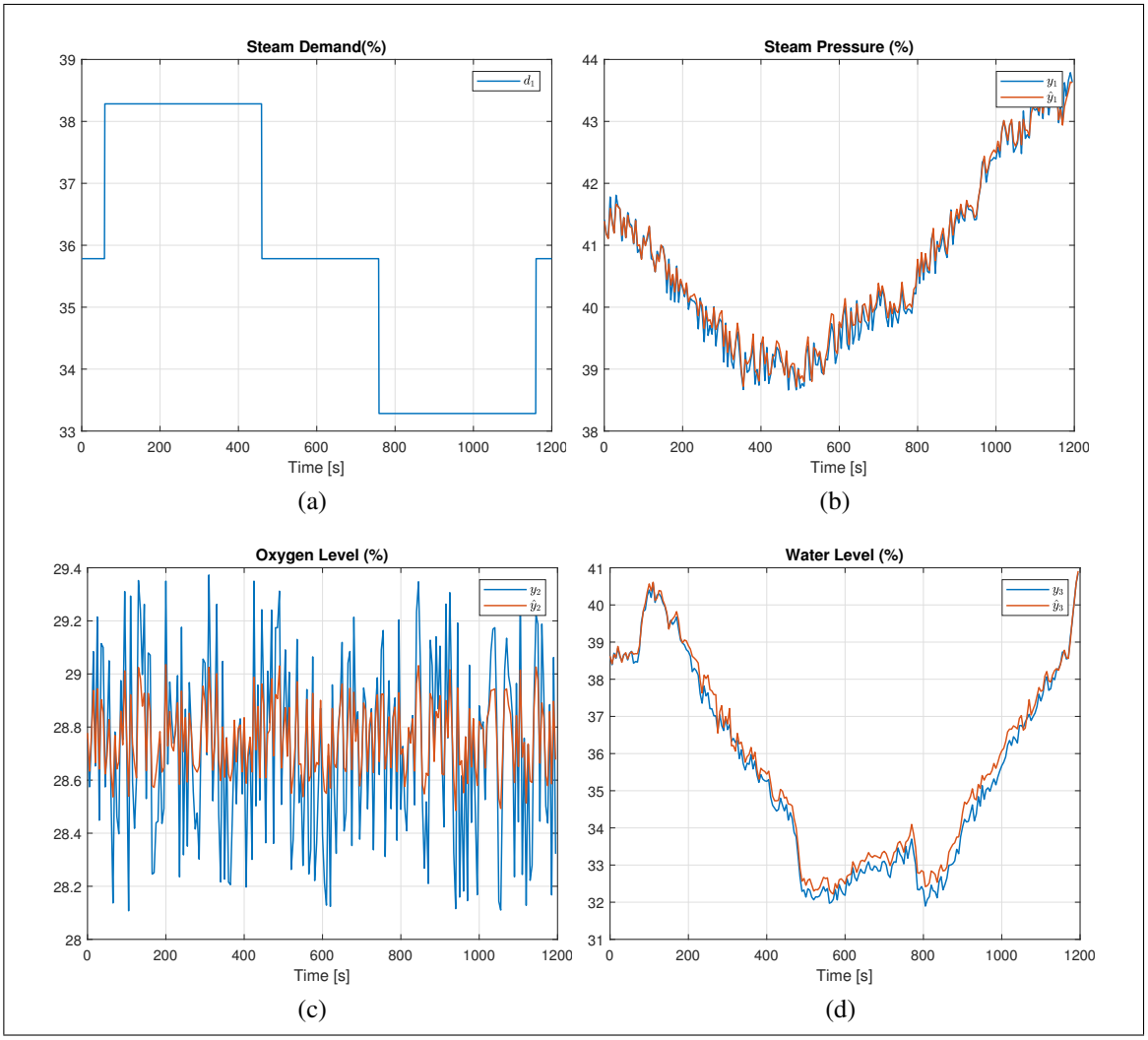


Figure D.1. Random forest one-step ahead prediction of boiler controlled variables for change in steam demand.

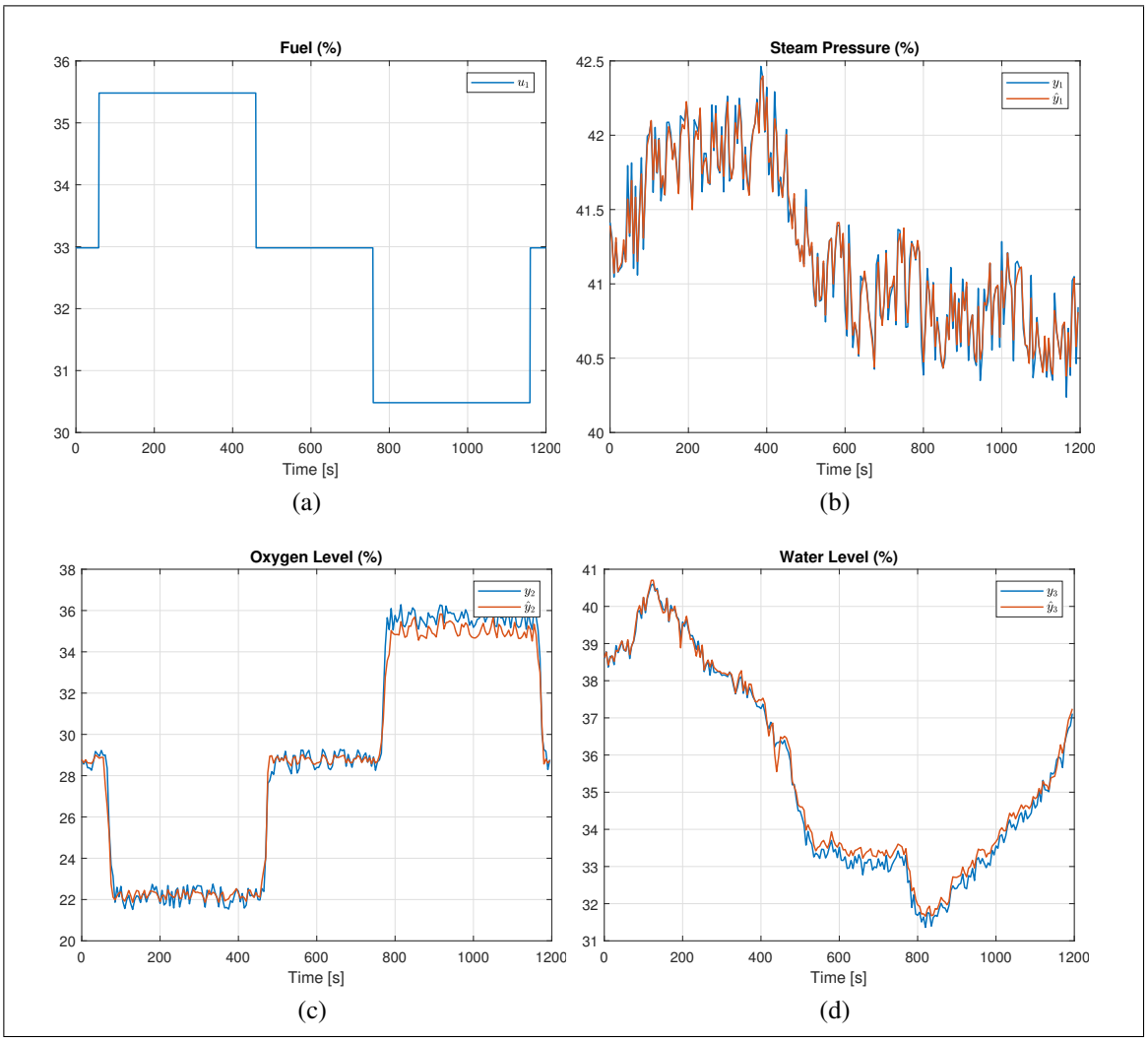


Figure D.2. Random forest one-step ahead prediction of boiler controlled variables for change in fuel.

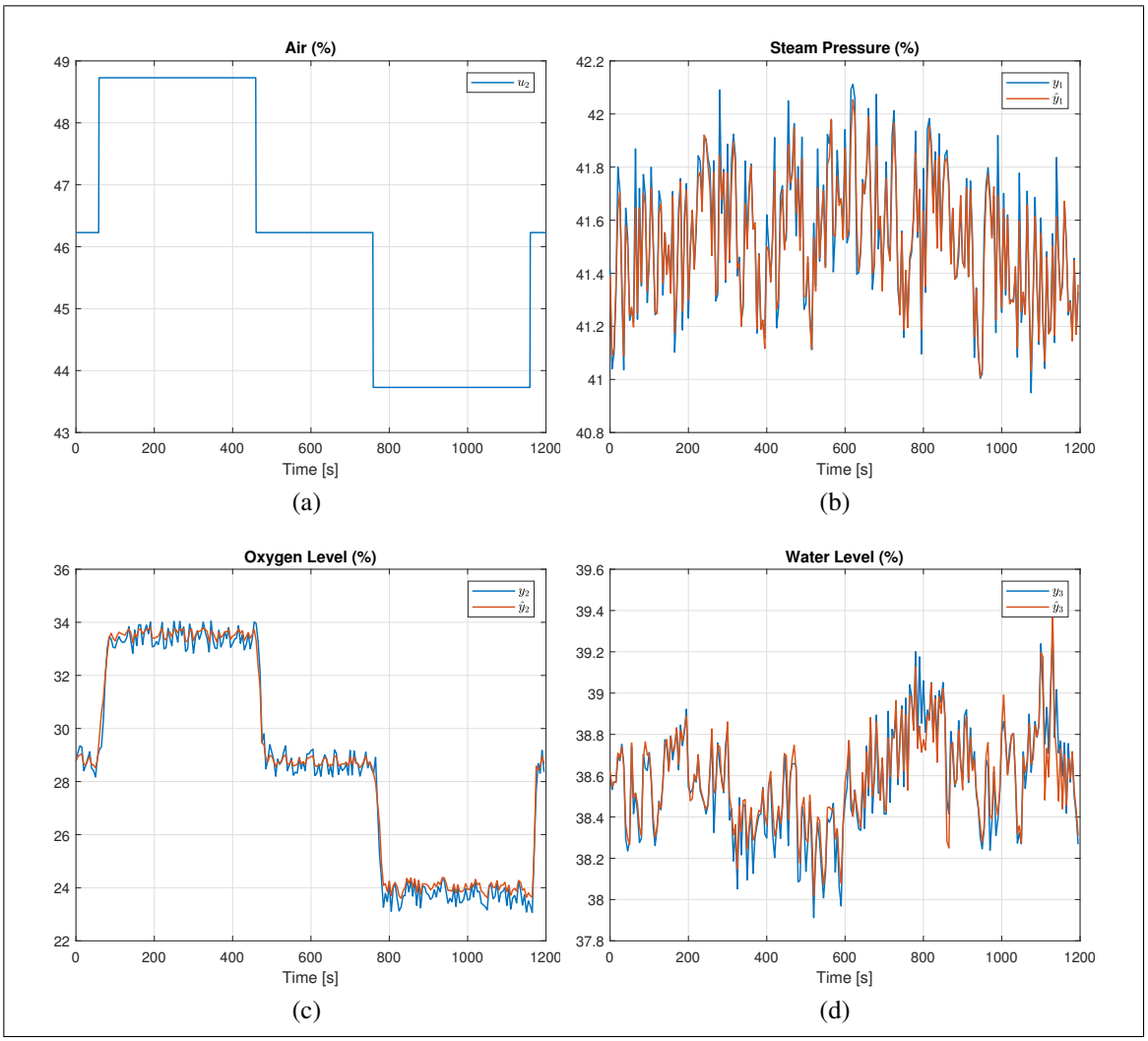


Figure D.3. Random forest one-step ahead prediction of boiler controlled variables for change in air inflow.

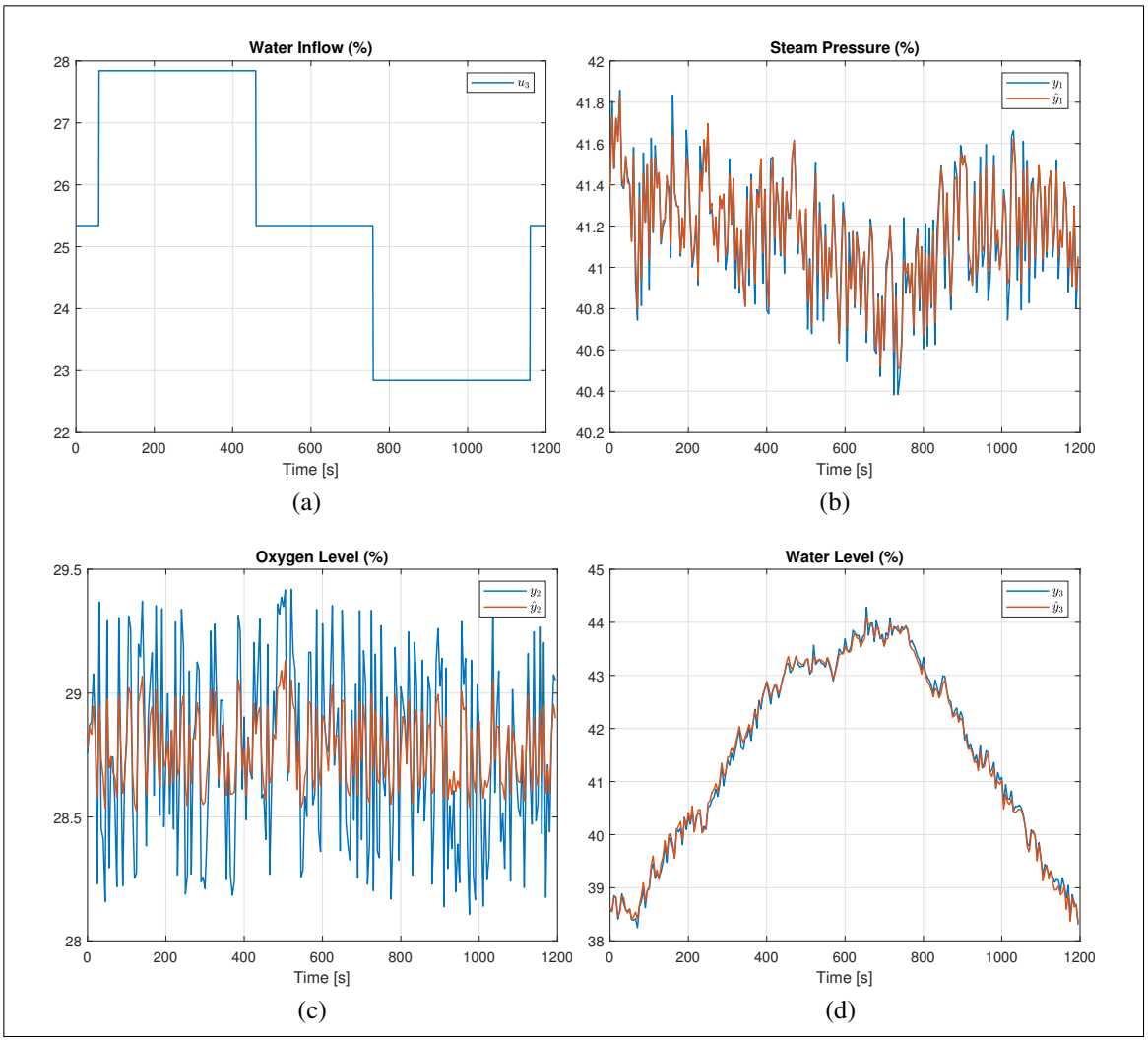


Figure D.4. Random forest one-step ahead prediction of boiler controlled variables for change in water inflow.

## D.2. Identified Models

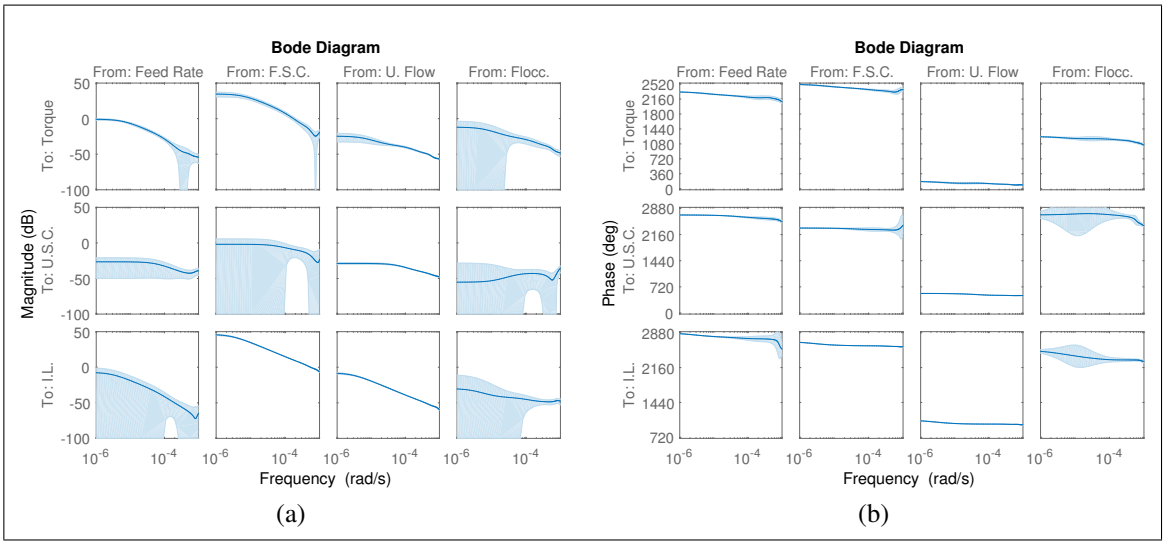


Figure D.5. Magnitude and phase response of transfer function pairs between inputs and outputs. The grey area shows the confidence interval for the identified response.

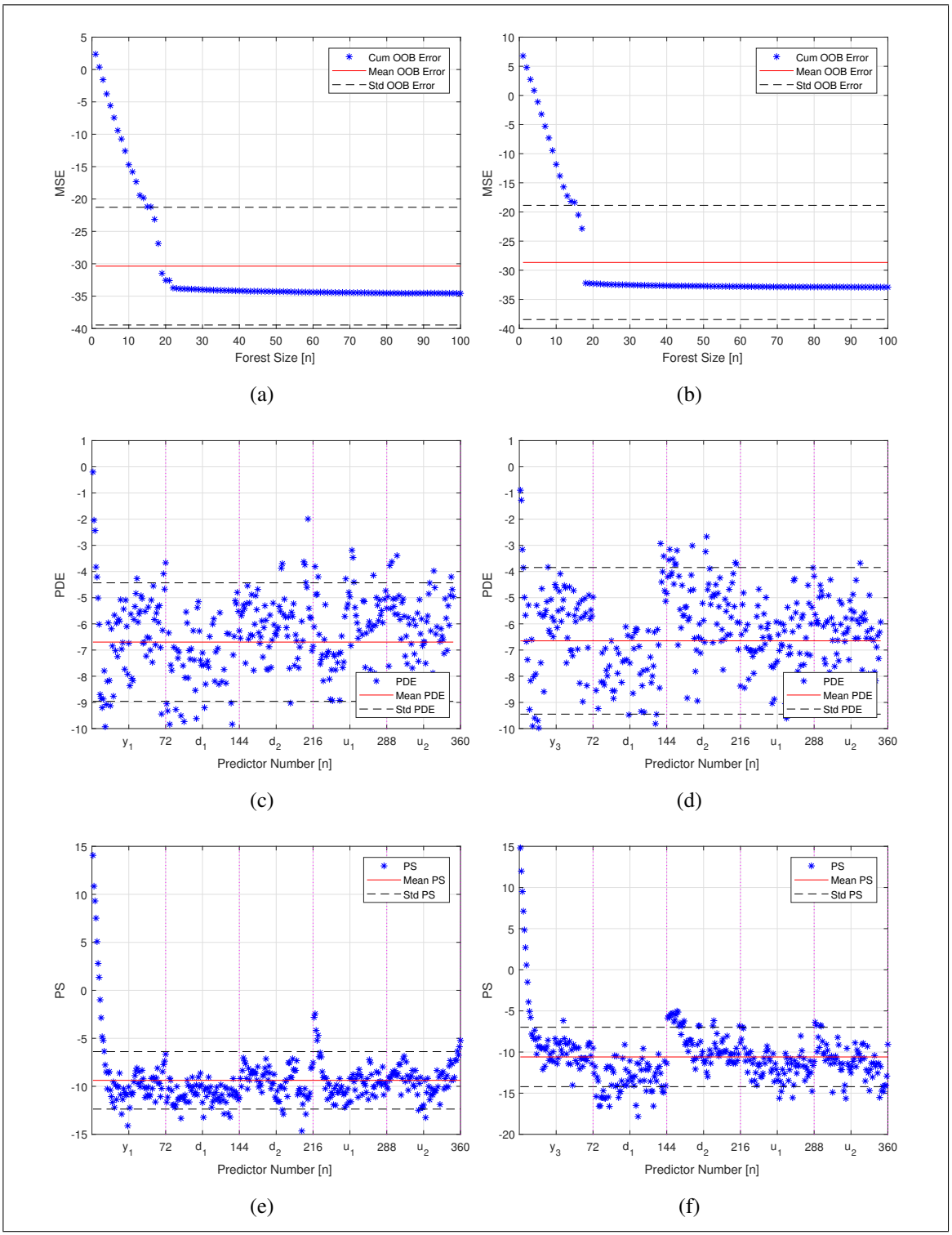


Figure D.6. OOB error, permuted predictor delta error and number of predictor splits for $T(t)$ and $h(t)$ - left and right columns - random forests with regressive strategy.

## D.3. Prediction Results

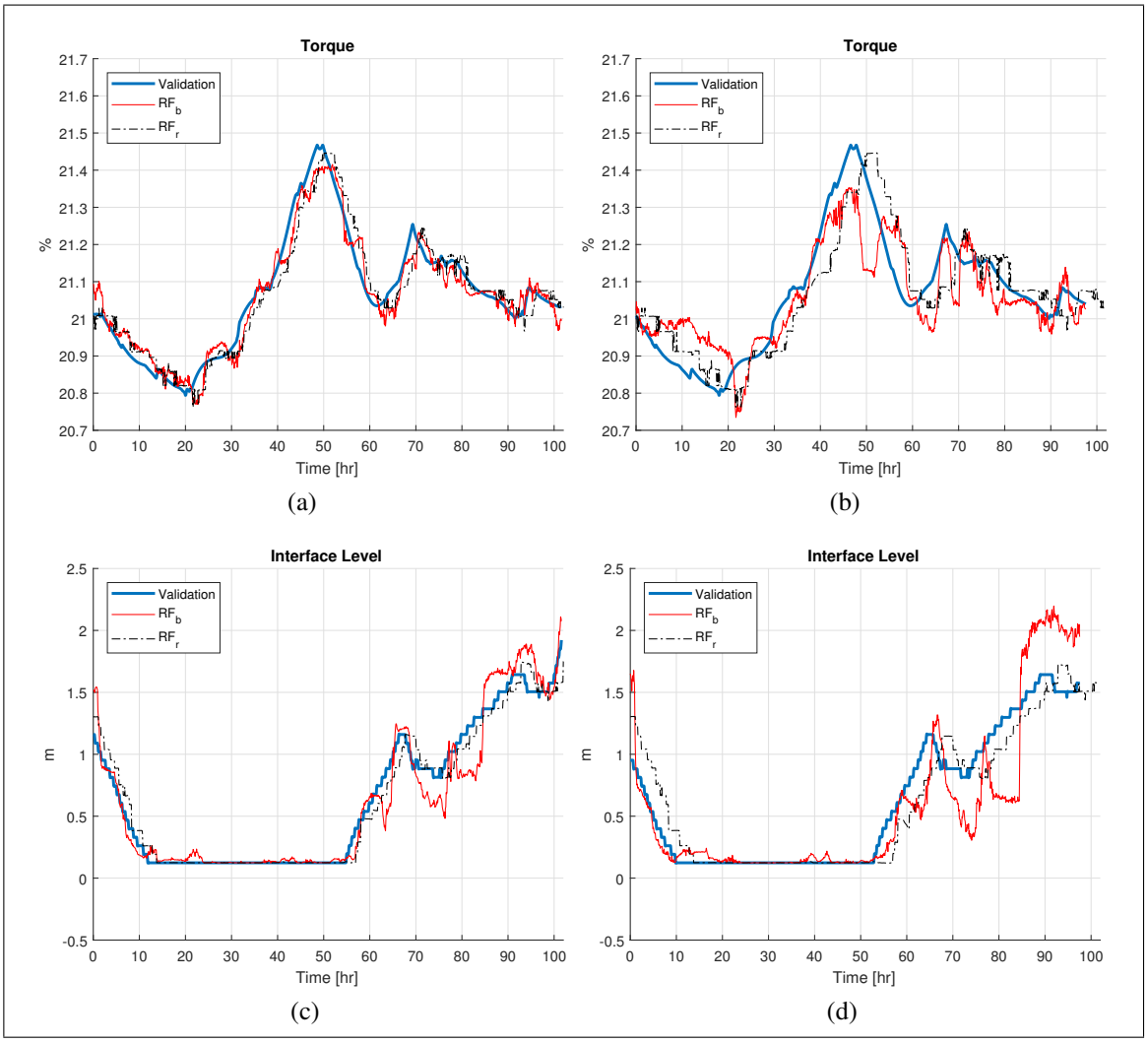


Figure D.7. Comparison between the recursive and multiple random forest strategies of two (figures D.7a and D.7c) and four (figures D.7b and D.7d) ahead prediction for $T(t)$ and $h(t)$.

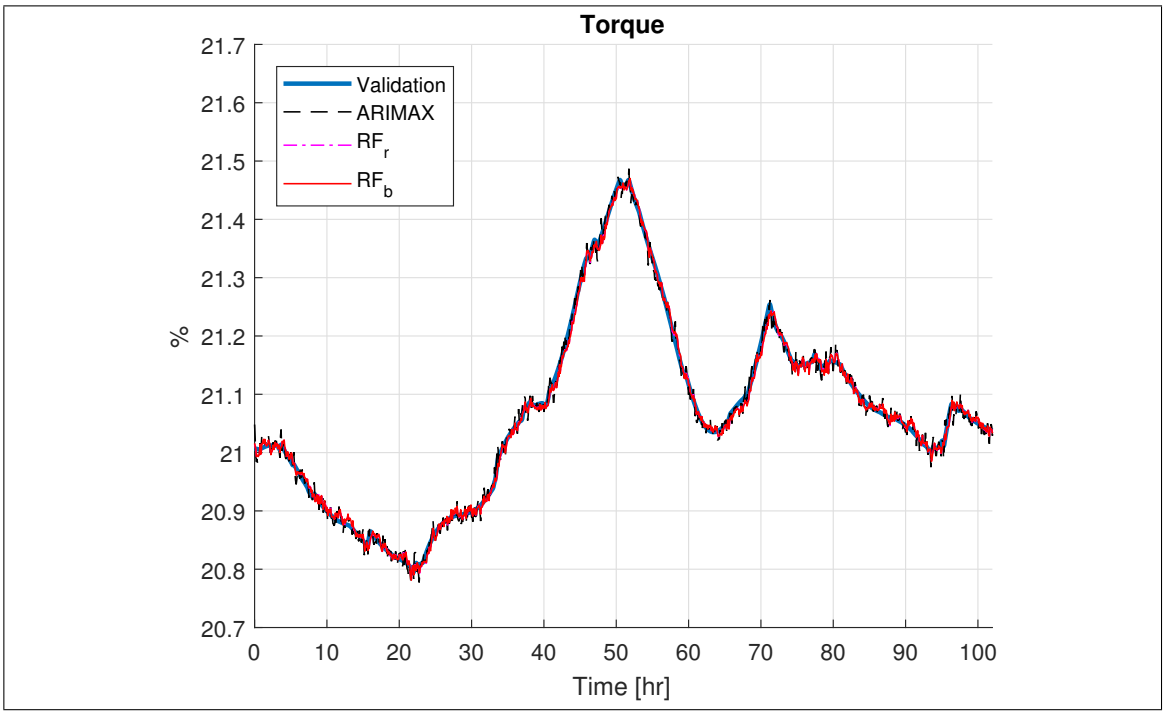


Figure D.8. Torque prediction comparison between the different predictors for a five minute ahead prediction.

Table D.1. Performance indicators of best Random Forests and ARIMAX models fit to all controlled variables.

| **Variable** | **MSE** $[1 \times 10^{-2}]$ | | | | **BFR** | | | |
|---|---|---|---|---|---|---|---|---|
| **Time Ahead [min]** | 5 | 90 | 120 | 240 | 5 | 90 | 120 | 240 |
| $T(t)$ ARIMAX | 0.02 | 0.26 | 0.43 | 1.64 | 90.05 | 67.01 | 57.36 | 16.34 |
| $T(t)$ $RF_r$ | 0.02 | 0.15 | 0.23 | 0.67 | 91.03 | 75.57 | 69.59 | 47.86 |
| $T(t)$ $RF_b$ | 0.02 | 0.12 | 0.16 | 0.77 | 90.94 | 78.39 | 74.17 | 44.28 |
| $C_u(t)$ ARIMAX | 1.55 | 2.27 | 2.51 | 3.92 | 68.72 | 62.19 | 60.19 | 50.26 |
| $C_u(t)$ $RF_r$ | 1.59 | 2.96 | 3.70 | 6.94 | 69.10 | 57.84 | 52.89 | 35.46 |
| $C_u(t)$ $RF_b$ | 1.60 | 2.68 | 3.36 | 6.91 | 69.00 | 59.91 | 55.10 | 35.62 |
| $h(t)$ ARIMAX | 0.03 | 0.59 | 1.01 | 3.92 | 96.95 | 86.88 | 82.81 | 66.19 |
| $h(t)$ $RF_r$ | 0.03 | 0.65 | 1.08 | 3.83 | 96.78 | 85.99 | 81.92 | 65.99 |
| $h(t)$ $RF_b$ | 0.03 | 0.90 | 2.28 | 8.48 | 96.84 | 83.53 | 73.78 | 49.40 |

## E. CONTROL APPLICATIONS TO PASTE PRODUCTION

### E.1. Expert Control Algorithm

ALGORITHM E.1. *The Expert Controller Algorithm Routine is as follows:*

```
for cv = 1:numCV

    %% Determination of Expert membership
    for cv = 1:numCV
        regionMemberCV(cv,:) = whatRegion(error,Regions)
    end
    for cv = 1:numCV
       for mv = 1:numMV
           % Always makes:
           % - Coefficient of MV action
           % - If integral control, verifies finess zone.
           % If derivative verifies sign
           % - Verifies room for MV to change

           % Calculation of proportional action
           propMV(mv,cv) = K_pExpert(cv)*dMV(mv);

            % Calculation of integral action
            intMV(mv,cv) = (dMV(mv)/T_iExpert(cv))*...
                           intError(cv)*...
                           errorCVMembership(cv,:)...
                           *intErrorThr(cv,:)';
```

```
            % Calculation of attenuation
            % works in the opposite direction
            % as proportional and integral control
            attMV(mv,cv) = -1*attenuate*dMV(mv)*...
                           (sign(dError(cv)) ~=...
                            sign(error(cv)));


            mvPerCV(mv,cv) = propMV(mv,cv)+...
                             intMV(mv,cv)+...
                             attMV(mv,cv);
        end
    end
    MV = mvPast+mean(mvPerCV,2);
    % Saturate in limits according
    for mv = 1:numMV
    if MV(mv) > mvRegions(mv,regionMemberCV(2),2);
        MV(mv) = mvRegions(mv,regionMemberCV(2),2);
    elseif MV(mv) < mvRegions(mv,regionMemberCV(2),1)
        MV(mv) = mvRegions(mv,regionMemberCV(2),1);
    end
    end
```

## E.2. Additional Control Results

### E.2.1. Thickener Stabilization

Table E.1. Comparison of control performance for thickener stabilization regarding torque.

| Criterion | MSE $[1 \times 10^{-2}]$ | IAE $[1 \times 10^{4}]$ | MAE $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 1.79 | 8.30 | 0.30 |
| MPC | 2.98 | 9.85 | 0.39 |
| Expert | 11.62 | 16.81 | 0.84 |
| PI | 2.45 | 9.27 | 0.36 |

Table E.2. Comparison of control performance for thickener stabilization regarding underflow solids concentration.

| Criterion | MSE $[1 \times 10^{-2}]$ | IAE $[1 \times 10^{4}]$ | MAE $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 2.23 | 9.24 | 0.40 |
| MPC | 0.00 | 0.18 | 0.03 |
| Expert | 14.95 | 22.68 | 0.91 |
| PI | 0.01 | 0.50 | 0.04 |

Table E.3. Comparison of control performance for thickener stabilization regarding interface level.

| Criterion | MSE $[1 \times 10^{-2}]$ | IAE $[1 \times 10^{4}]$ | MAE $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 244.73 | 82.64 | 3.31 |
| MPC | 145.05 | 74.29 | 2.28 |
| Expert | 382.52 | 120.42 | 3.66 |
| PI | 113.51 | 60.73 | 2.21 |

Table E.4. Comparison of control performance for thickener stabilization regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^0]$ | | **ISU** $[1 \times 10^6]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 5744.70 | 40.68 | 82.52 | 1.22 |
| MPC | 390.37 | 2.00 | 83.75 | 3.30 |
| Expert | 6.09 | 0.05 | 159.83 | 1.09 |
| PI | 5.84 | 6.53 | 89.37 | 4.73 |

### E.2.2. Interface Level Reference Tracking

Table E.5. Comparison of control performance for interface level tracking regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^4]$ | **MAE** $[1 \times 10^0]$ |
|---|---|---|---|
| RF-MPC | 4.40 | 7.23 | 0.34 |
| MPC | 4.78 | 7.34 | 0.33 |
| Expert | 30.83 | 18.87 | 0.77 |
| PI | 5.08 | 7.61 | 0.39 |

Table E.6. Comparison of control performance for interface level tracking regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^4]$ | **MAE** $[1 \times 10^0]$ |
|---|---|---|---|
| RF-MPC | 3.54 | 5.28 | 0.51 |
| MPC | 0.00 | 0.08 | 0.05 |
| Expert | 34.65 | 17.88 | 1.03 |
| PI | 0.08 | 0.39 | 0.22 |

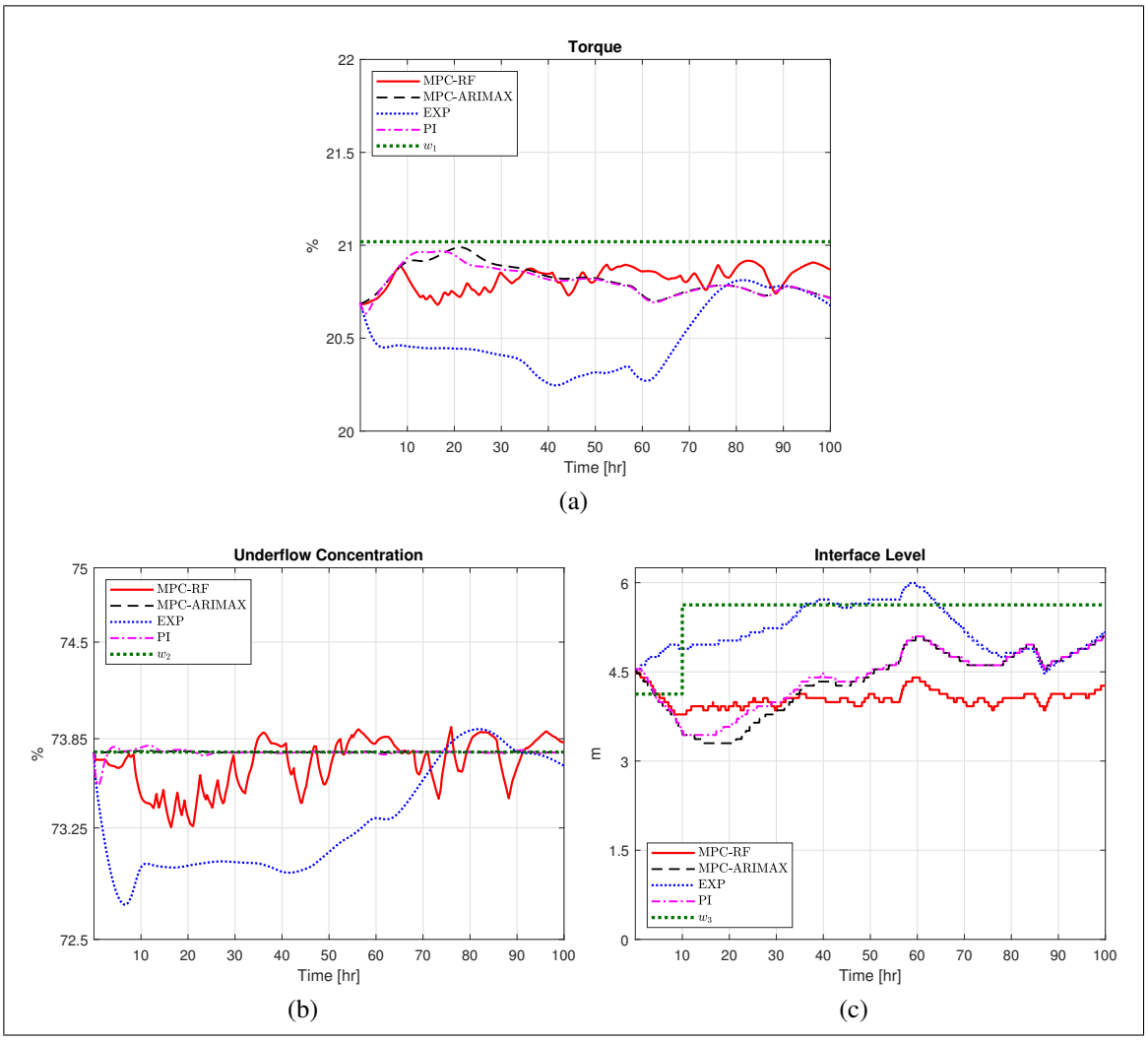


Figure E.1. Evolution of controlled variables for interface level setpoint tracking.

Table E.7. Comparison of control performance for interface level tracking regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 227.60 | 52.07 | 1.84 |
| MPC | 172.98 | 41.96 | 2.33 |
| Expert | 31.82 | 17.01 | 1.15 |
| PI | 151.85 | 39.85 | 2.19 |

Table E.8. Comparison of control performance for interface level tracking regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^0]$ | | **ISU** $[1 \times 10^6]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 7246.36 | 13.60 | 56.94 | 0.10 |
| MPC | 399.24 | 0.50 | 22.66 | 1.33 |
| Expert | 2.88 | 0.03 | 27.26 | 0.47 |
| PI | 21.31 | 11.30 | 24.58 | 0.87 |

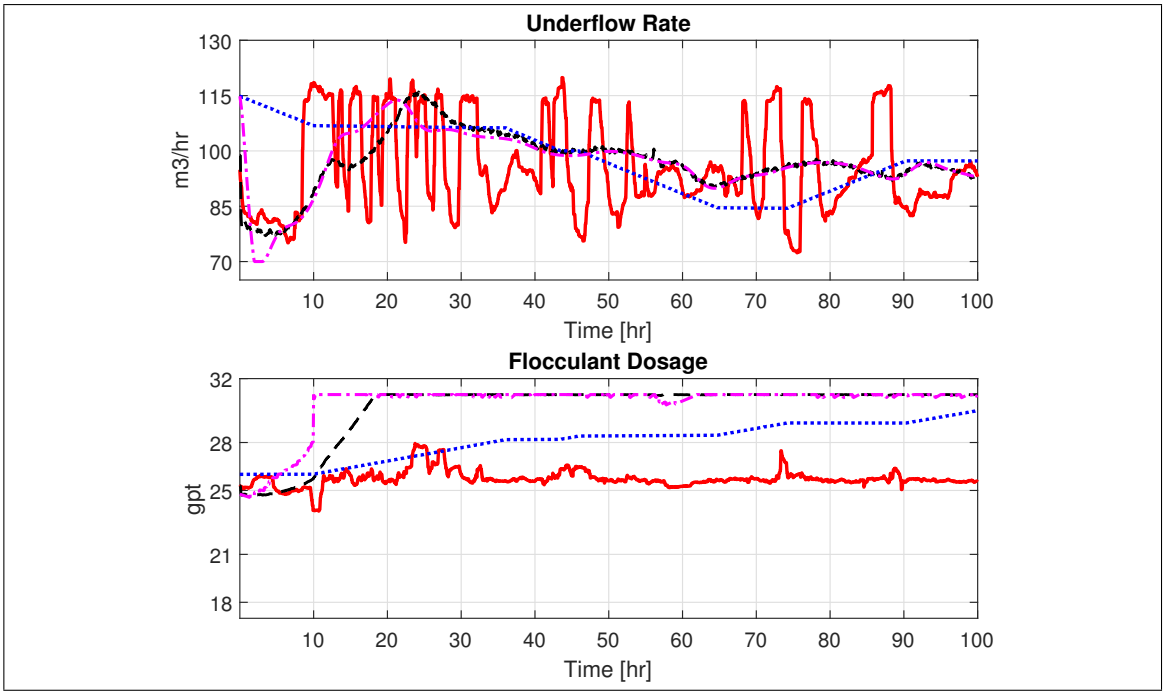


Figure E.2. Manipulated variables calculated by all controllers for interface level setpoint tracking. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

### E.2.3. Underflow Solids Concentration Tracking

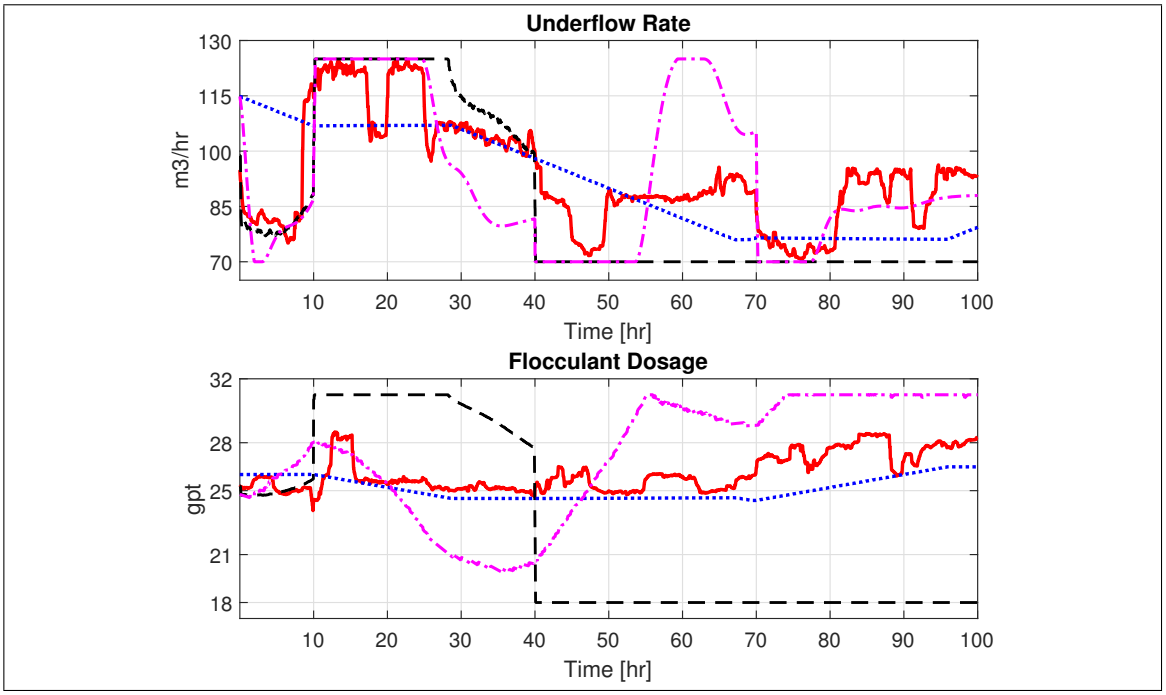


Figure E.3. Manipulated variables calculated by all controllers for underflow solids concentration setpoint tracking. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

Table E.9. Comparison of control performance for underflow solids reference tracking regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 13.92 | 11.72 | 0.69 |
| MPC | 30.93 | 17.32 | 1.01 |
| Expert | 26.90 | 16.81 | 0.79 |
| PI | 16.77 | 12.22 | 0.82 |

Table E.10. Comparison of control performance for underflow solids reference tracking regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 10.89 | 8.95 | 0.96 |
| MPC | 94.85 | 27.46 | 1.77 |
| Expert | 24.60 | 13.51 | 1.03 |
| PI | 9.50 | 6.39 | 1.03 |

Table E.11. Comparison of control performance for underflow solids reference tracking regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 54.36 | 22.34 | 1.31 |
| MPC | 494.32 | 61.82 | 4.00 |
| Expert | 128.00 | 37.13 | 2.14 |
| PI | 80.32 | 27.41 | 1.65 |

Table E.12. Comparison of control performance for underflow solids reference tracking regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^{0}]$ | | **ISU** $[1 \times 10^{6}]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 1368.40 | 14.98 | 77.50 | 0.50 |
| MPC | 1153.66 | 72.29 | 179.96 | 11.37 |
| Expert | 2.87 | 0.03 | 67.80 | 0.17 |
| PI | 398.13 | 4.86 | 151.29 | 5.15 |

### E.2.4. Feed Solids Concentration Change Rejection

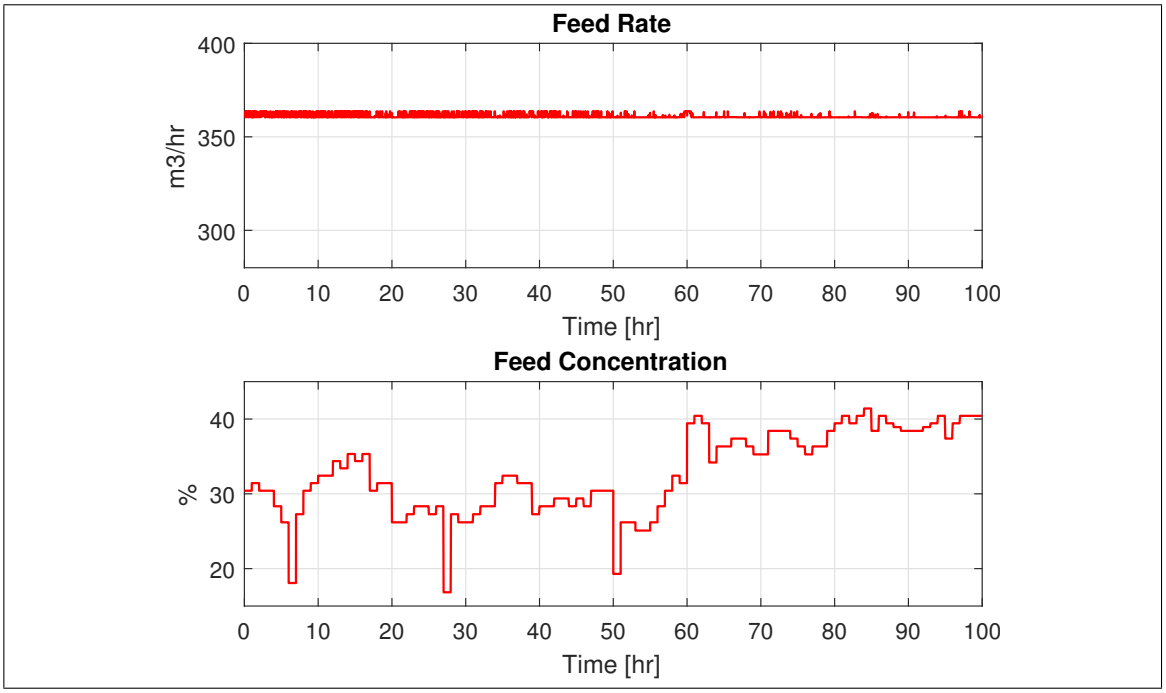


Figure E.4. Disturbance variables for step change in feed solids concentration.

Table E.13. Comparison of control performance for feed solids concentration change regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 1.96 | 4.17 | 0.34 |
| MPC | 2.53 | 4.79 | 0.33 |
| Expert | 8.97 | 8.45 | 0.57 |
| PI | 2.98 | 4.89 | 0.39 |

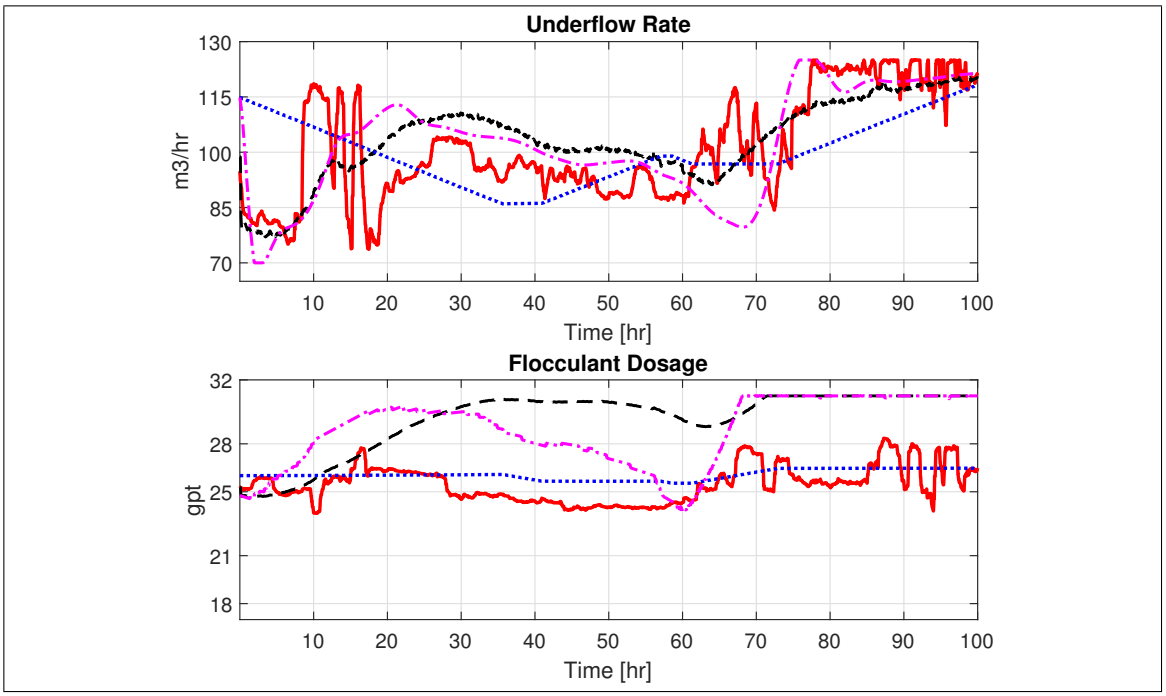


Figure E.5. Manipulated variables calculated by all controllers for feed solids concentration change. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

Table E.14. Comparison of control performance for feed solids concentration change regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 456.03 | 57.80 | 4.00 |
| MPC | 193.09 | 36.98 | 2.90 |
| Expert | 404.34 | 50.85 | 4.00 |
| PI | 272.27 | 41.81 | 3.31 |

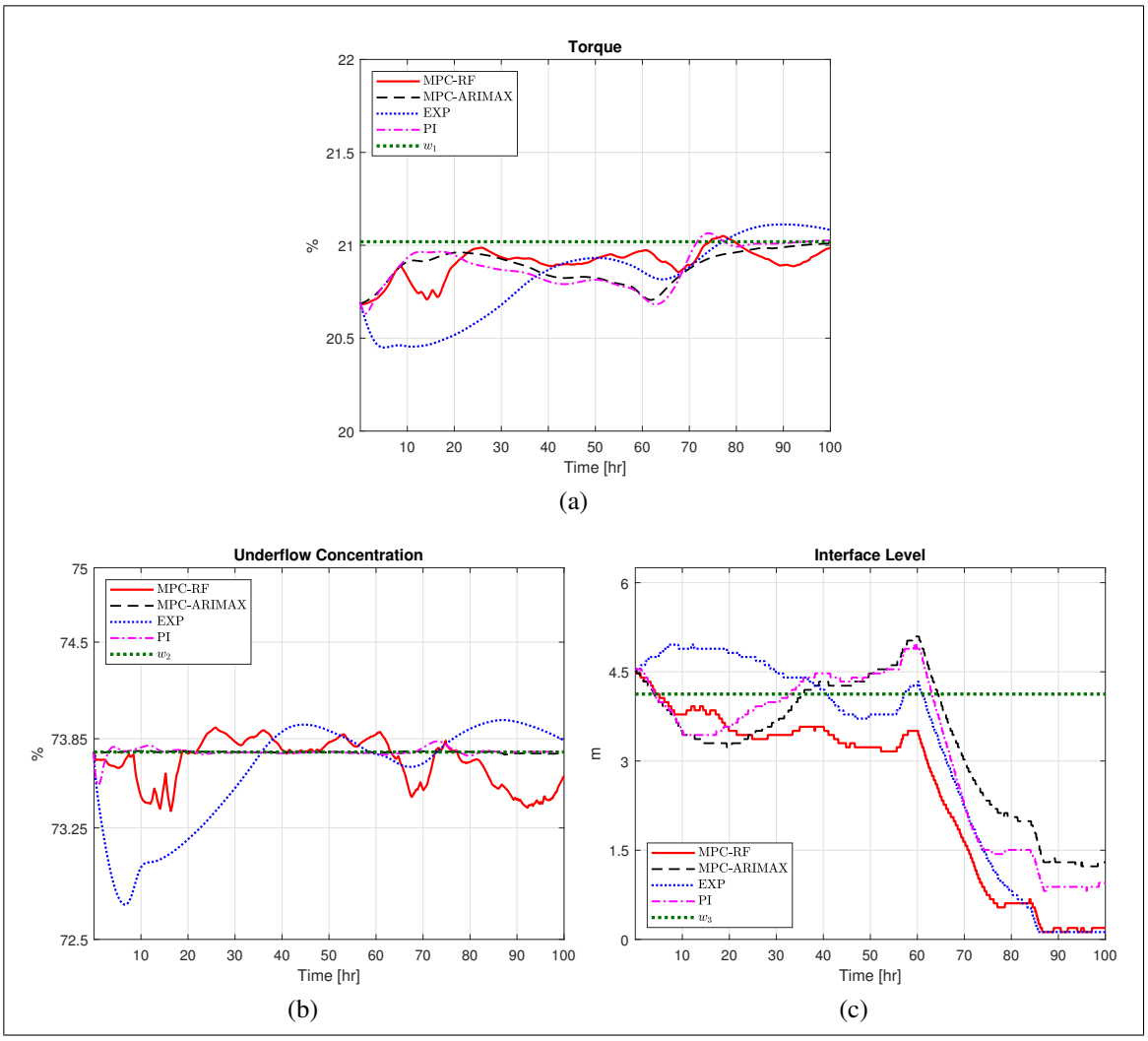


Figure E.6. Evolution of controlled variables for feed solids concentration change.

Table E.15. Comparison of control performance for feed solids concentration change regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 120.54 | 29.57 | 2.00 |
| MPC | 198.80 | 43.65 | 2.07 |
| Expert | 246.69 | 51.85 | 2.14 |
| PI | 142.51 | 32.13 | 2.00 |

Table E.16. Comparison of control performance for feed solids concentration change regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^0]$ | | **ISU** $[1 \times 10^6]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 3296.48 | 29.04 | 74.14 | 0.47 |
| MPC | 381.32 | 0.48 | 42.42 | 1.35 |
| Expert | 4.35 | 0.01 | 26.49 | 0.03 |
| PI | 27.03 | 3.65 | 71.55 | 1.64 |

### E.2.5. Feed Rate Change Rejection

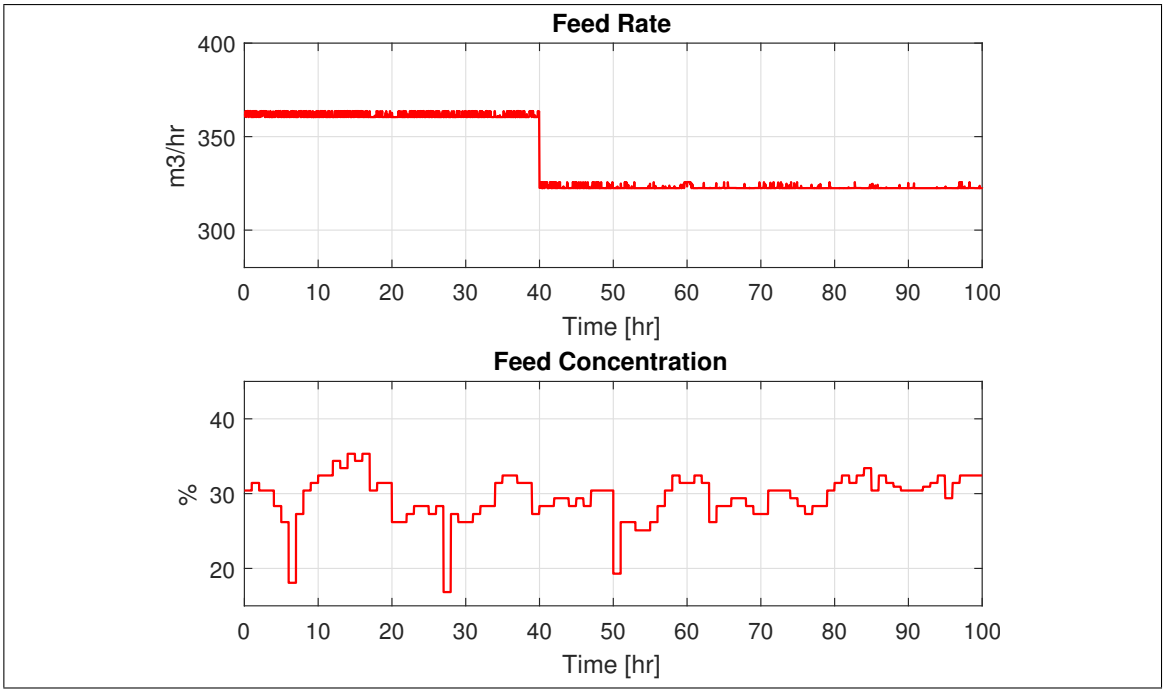


Figure E.7. Disturbance variables for step change in feedflow rate.

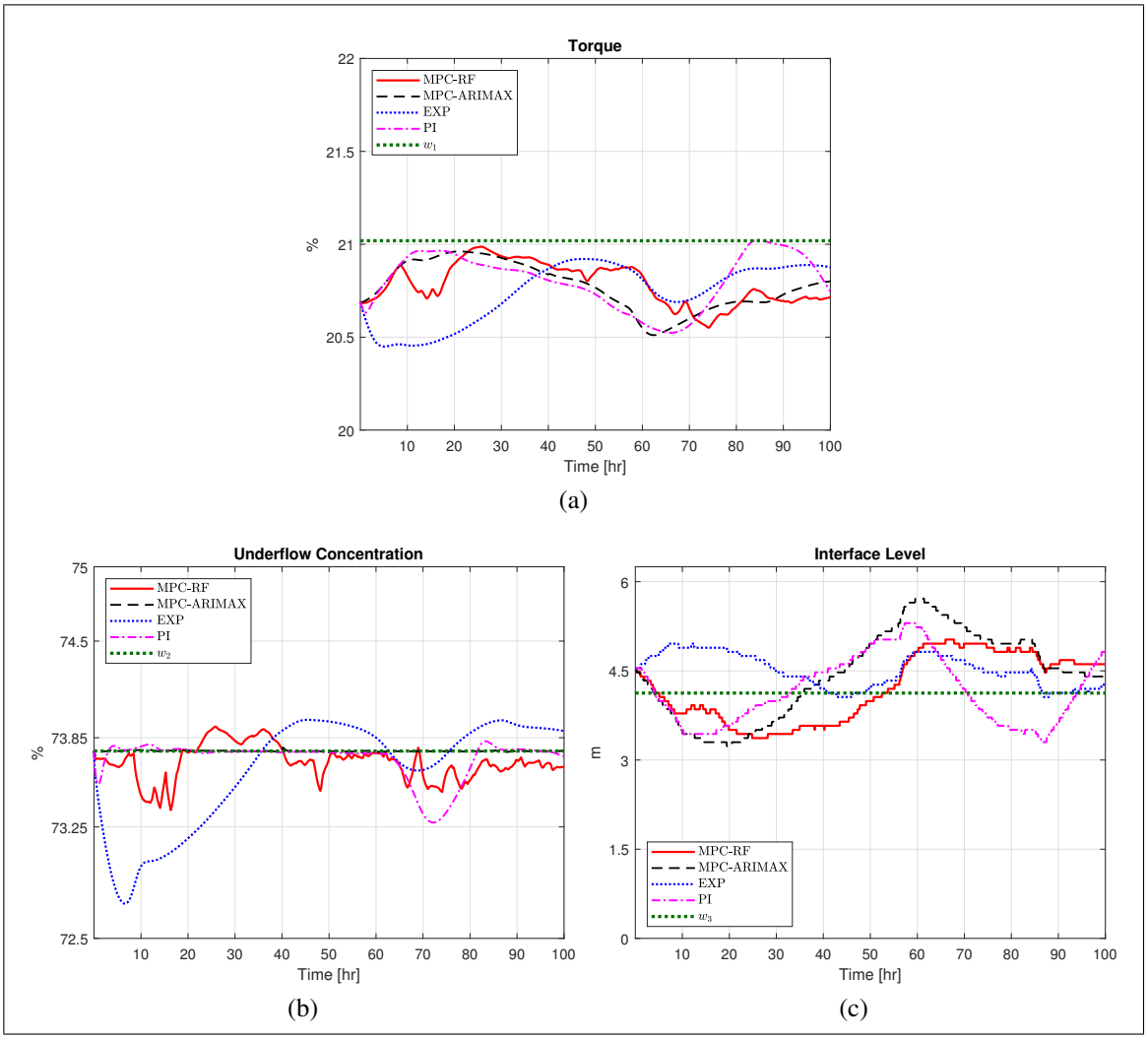


Figure E.8. Evolution of controlled variables for feedflow rate change.

Table E.17. Comparison of control performance for feedflow rate change regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 6.65 | 8.36 | 0.47 |
| MPC | 7.78 | 8.92 | 0.51 |
| Expert | 10.47 | 10.12 | 0.57 |
| PI | 6.65 | 7.66 | 0.50 |

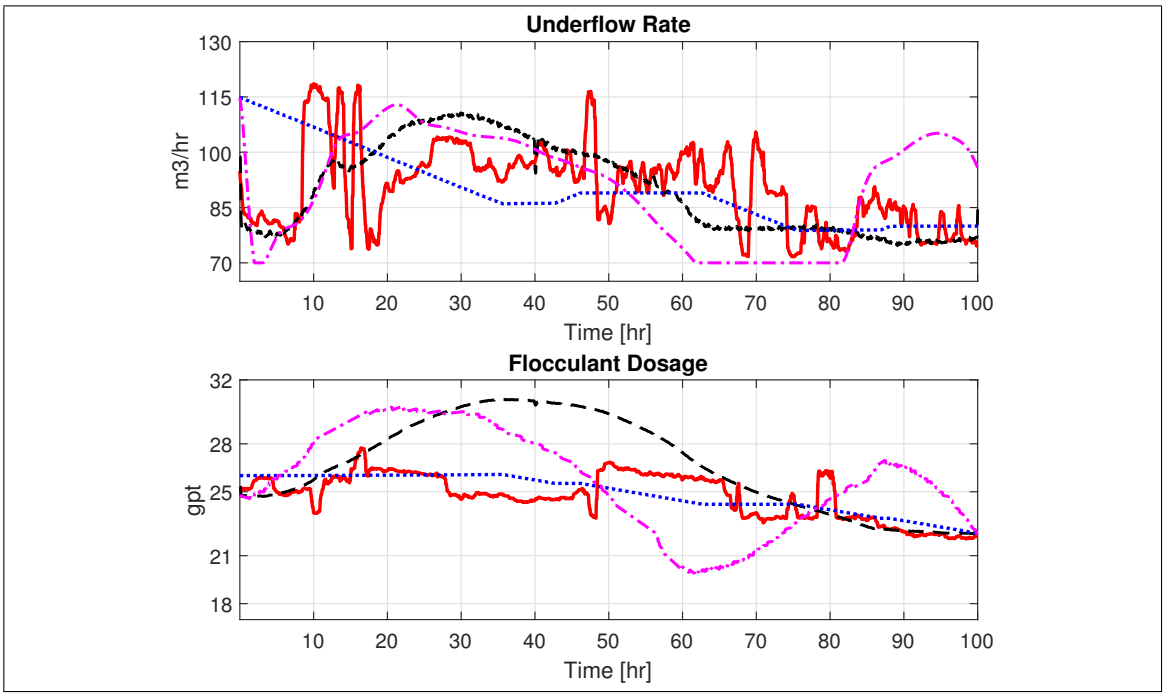


Figure E.9. Manipulated variables calculated by all controllers for feed-flow rate change. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

Table E.18. Comparison of control performance for feedflow rate change regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 1.98 | 4.06 | 0.40 |
| MPC | 0.00 | 0.09 | 0.05 |
| Expert | 14.95 | 10.22 | 1.03 |
| PI | 1.93 | 2.36 | 0.48 |

Table E.19. Comparison of control performance for feedflow rate change regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 32.72 | 18.76 | 0.90 |
| MPC | 62.43 | 24.56 | 1.59 |
| Expert | 22.32 | 14.00 | 0.83 |
| PI | 31.92 | 17.55 | 1.18 |

Table E.20. Comparison of control performance for feedflow rate change regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^{0}]$ | | **ISU** $[1 \times 10^{6}]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 2919.79 | 18.59 | 44.41 | 0.69 |
| MPC | 522.39 | 0.75 | 53.73 | 3.05 |
| Expert | 2.86 | 0.02 | 35.48 | 0.47 |
| PI | 24.94 | 4.23 | 78.02 | 3.57 |

### E.2.6. Feed Particle Size Change Rejection

Table E.21. Comparison of control performance for feed particle size change regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 4.13 | 6.69 | 0.34 |
| MPC | 11.91 | 10.65 | 0.66 |
| Expert | 16.31 | 13.33 | 0.57 |
| PI | 13.93 | 11.29 | 0.70 |

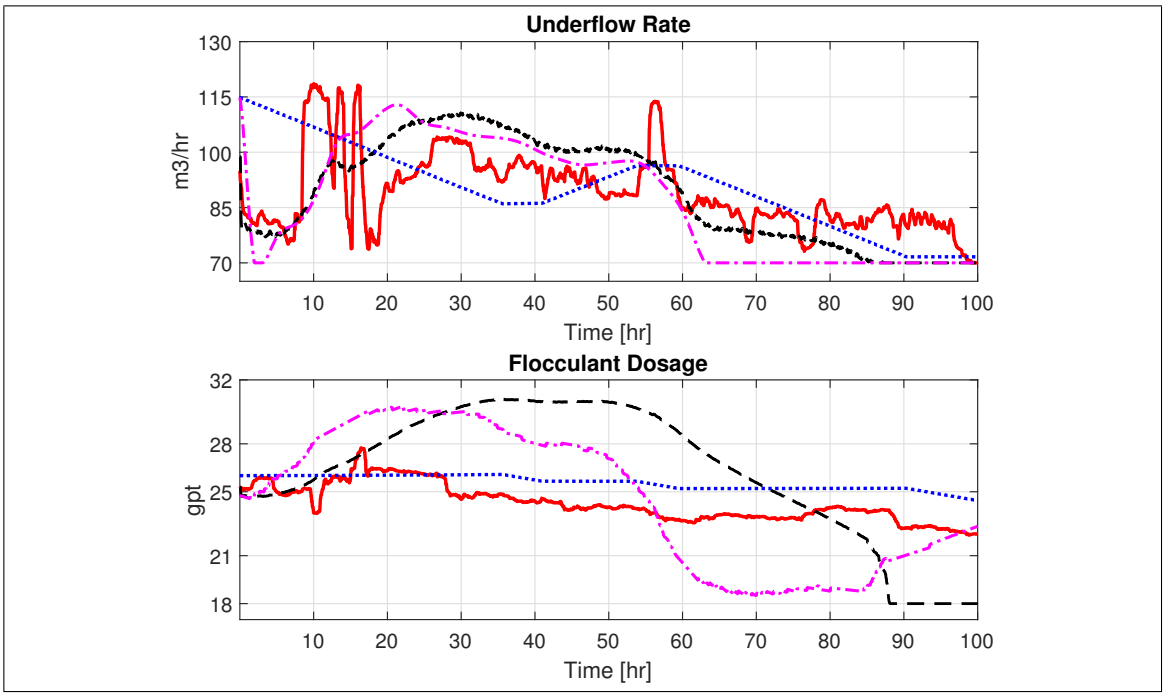


Figure E.10. Manipulated variables calculated by all controllers for feed particle size change. RF-MPC is in solid red, conventional MPC is dashed black, expert is dotted blue and PI is dash-dotted pink.

Table E.22. Comparison of control performance for feed particle size change regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 1.30 | 2.80 | 0.40 |
| MPC | 1.11 | 1.07 | 0.76 |
| Expert | 15.97 | 10.97 | 1.03 |
| PI | 16.56 | 8.63 | 0.91 |

Table E.23. Comparison of control performance for feed particle size change regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 30.23 | 18.13 | 0.90 |
| MPC | 84.10 | 28.09 | 1.66 |
| Expert | 79.02 | 28.07 | 1.59 |
| PI | 35.19 | 17.23 | 1.45 |

Table E.24. Comparison of control performance for feed particle size change regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^{0}]$ | | **ISU** $[1 \times 10^{6}]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 1983.41 | 5.08 | 39.72 | 0.47 |
| MPC | 355.09 | 1.44 | 71.43 | 6.04 |
| Expert | 4.27 | 0.01 | 45.78 | 0.07 |
| PI | 22.29 | 3.72 | 89.77 | 6.74 |

### E.2.7. Solid Throughput Increase Rejection

Table E.25. Comparison of control performance for increase in solid throughput regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 2.01 | 3.92 | 0.34 |
| MPC | 5.39 | 6.49 | 0.58 |
| Expert | 10.35 | 9.51 | 0.57 |
| PI | 1.85 | 3.78 | 0.39 |

Table E.26. Comparison of control performance for increase in solid throughput regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 3.28 | 5.73 | 0.40 |
| MPC | 11.67 | 7.89 | 0.96 |
| Expert | 20.13 | 13.61 | 1.03 |
| PI | 0.08 | 0.42 | 0.22 |

Table E.27. Comparison of control performance for increase in solid throughput regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 380.33 | 53.85 | 4.00 |
| MPC | 973.55 | 96.86 | 4.00 |
| Expert | 496.63 | 63.86 | 4.00 |
| PI | 332.76 | 46.94 | 4.00 |

Table E.28. Comparison of control performance for increase in solid throughput regarding manipulated variables.

| **Criterion** | **SCE** $[1 \times 10^{0}]$ | | **ISU** $[1 \times 10^{6}]$ | |
|---|---|---|---|---|
| **Variable** | $Q_u(t)$ | $F(t)$ | $Q_u(t)$ | $F(t)$ |
| RF-MPC | 3811.01 | 31.59 | 77.71 | 0.43 |
| MPC | 549.61 | 9.95 | 112.66 | 9.23 |
| Expert | 4.30 | 0.01 | 42.34 | 0.09 |
| PI | 21.37 | 4.53 | 53.95 | 1.16 |

### E.2.8. Solid Throughput Decrease Rejection

Table E.29. Comparison of control performance for decrease in solid throughput regarding torque.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 28.84 | 14.57 | 1.35 |
| MPC | 69.59 | 22.17 | 2.18 |
| Expert | 82.11 | 23.49 | 3.19 |
| PI | 69.58 | 20.63 | 2.14 |

Table E.30. Comparison of control performance for decrease in solid throughput regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 16.88 | 10.04 | 1.15 |
| MPC | 114.31 | 25.34 | 2.59 |
| Expert | 46.01 | 21.42 | 1.43 |
| PI | 115.24 | 23.10 | 2.50 |

Table E.31. Comparison of control performance for decrease in solid throughput regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 124.95 | 29.13 | 2.63 |
| MPC | 317.00 | 52.57 | 3.73 |
| Expert | 233.68 | 45.37 | 3.25 |
| PI | 188.34 | 39.28 | 2.90 |

### E.2.9. Overall Results

Table E.32. Average control performance throughout all scenarios regarding underflow solids concentration regarding torque

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 7.97 | 7.57 | 0.52 |
| MPC | 17.19 | 10.61 | 0.74 |
| Expert | 24.45 | 13.76 | 0.95 |
| PI | 15.10 | 9.32 | 0.71 |

Table E.33. Average control performance throughout all scenarios regarding underflow solids concentration.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 5.28 | 5.58 | 0.58 |
| MPC | 27.74 | 7.77 | 0.78 |
| Expert | 23.12 | 13.23 | 1.08 |
| PI | 17.95 | 5.29 | 0.73 |

Table E.34. Average control performance throughout all scenarios regarding interface level.

| **Criterion** | **MSE** $[1 \times 10^{-2}]$ | **IAE** $[1 \times 10^{4}]$ | **MAE** $[1 \times 10^{0}]$ |
|---|---|---|---|
| RF-MPC | 171.10 | 34.83 | 2.10 |
| MPC | 291.21 | 45.15 | 2.65 |
| Expert | 176.65 | 33.54 | 2.22 |
| PI | 138.63 | 30.32 | 2.19 |