Truls Ødegaard

Simulation and Control of Anaerobic Digestion Process for Biogas Energy Production: A Methodology Comparison

Master's thesis in Robotics and Cybernetics
Supervisor: Tiina Marjatta Komulainen
Co-supervisors: Morten Rostad Haugen, Hilde Johansen, Fabio Polesel
January 2024

Oslo Metropolitan University (OsloMet)
Faculty of Technology Art and Design

ABSTRACT

State-of-the-art mechanically driven model of the anaerobic digestion process was compared to data driven methodology of System Identification and Machine Learning.

Through collaboration with OsloMet, Veas WWTP and DHI was data collected for analysis and to compose a dataset that DHIs West software could use to run their mechanically driven simulation based on the ADM1 model.

Live data from the same plant was also collected to create a dataset where two data driven models were created; one using System Identification methodology and the other using Machine Learning. Both data driven models were attempted for closed-loop integration with controllers.

DHIs model output tracked the actual output, showing similar behaviour to the actual plant. The data driven models showed less stable behaviour, although at times closer tracking to actual output than in West. Closed-loop integration with PI-controller and MPC on the SI-model showed close setpoint tracking with predictive inputs to the MV. Controller implementation on the ML-model showed less stable behaviour and further development of controller tuning is needed.

Keywords: Renewable Energy, Wastewater Treatment, Biogas Production, Anaerobic Digestion, Veas, DHI, West, Simulation, Control, Machine Learning, Recurrent Neural Network, Long-Short Term Memory, System Identification

PREFACE

I would first of all thank my Supervisor, Tiina Marjatta Komulainen. The knowledge and support given has made the journey through this project more pain free. The love for your work really shines through.

I would like to thank our collaboration partners in Veas and DHI, especially Hilde Johansen, Morten Rostad Haugen and Fabio Polesel. You have all been very supportive, so I want to thank you for the time you invested in me.

I have to thank OsloMet for four wonderful years at your institution, to all my past Professors and fellow students, and a special thank to Chi Huu Luong. Meeting you in the hallways were a bright spot in our studies.

And lastly, I want to thank all my friends and family who have supported me in the last four years and who I've not been able to visit for months. So, to Lasse Heiland-Ødegaard, my nephew: I'm sorry, but now I'm finally free and I'll come visit you soon.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

List of all abbreviations in alphabetic order:

- **AD** Anaerobic Digester
- **ADM1** Anaerobic Digester Model no.1
- **COD** Chemical Oxygen Demand
- **CV** Controlled Variable
- **DV** Disturbance Variable
- **FSS/FS** Inorganic Suspended Solids
- **IAE** Integral of Absolute Error
- **ISE** Integral of Squared Error
- **LSTM** Long-Short Term Memory
- **ML** Machine Learning
- **MV** Manipulated Variable
- **PID** Proportional-Integral-Derivative
- **RNN** Recurrent Neural Network
- **SI** System Identification
- **TSS/TS** Total Suspended Solids
- **VSS/VS** Organic Suspended Solids
- **WWTP** WasteWater Treatment Plant

CHAPTER

ONE

INTRODUCTION

According to International Water Association, wastewater treatment plants (WWTP) consumes an estimated 1-3% of global energy output [1]. The European Commission revised the Urban Waste Water Treatment Directive (UWWTD) in 26-October-2022, "adapting it to the newest standards" and aims to "reduce pollution, energy use and greenhouse gas emissions" [2]. Capturing biogas energy from WWTPs potentially supports all these goals.

Through organic processes, our waste can be exploited of its organic energy potential through the natural degradation by microorganisms. Contained in an anaerobic environment the activity the microorganism cultures' degradation of the organic material they get in contact with, can under the right circumstances exhaust biogas that can be captured and used for electricity production, heating or fuel.

At the wastewater treatment plant at Veas in Røyken, Norway, stripped sludge captured from the wastewater purification process contains organic material that gets pumped into large tanks, void of oxygen (anaerobic), containing microorganism cultures that breaks down the organic material and produces biogas. Sludge from the anaerobic tanks (digesters) are also pumped out continuously as new sludge is filled and is a source for fertilizer production for use in agriculture [3].

There are challenges to this technology, however. The different types of bacteria cultures present in an anaerobic digester (AD) breaks down different molecules and creates sub-product that can affect the activity of other bacteria cultures, resulting in a multi-layer balance problem for the resulting biogas output. By avoiding stagnation of the liquid, local concentration differences can be avoided, but will also result in organic material being pumped out of the AD as the liquid sludge approaches uniform content distribution. Any non-digested organic material pumped out of the AD increases the difficulty of avoiding this being digested outside of the controlled volume of the AD. If not controlled properly, any organic activity outside of the controlled volumes can result in greenhouse gasses being leaked into our atmosphere. Increasing the hold-up time, the time which the bacteria has to digest all organic material in the AD, will reduce the concentration of non-digested organic material exiting the AD, but will also reduce the production

rate of biogas.

Other challenges of the AD-process are the observation and monitoring of the tank states. The contained volume of the AD is highly corrosive and proposes a health hazard to operators and sensors mounted inside would need regular maintenance due to clogging, bio fouling and degradation. Monitoring of the inlet sludge can be used to estimate the states of the AD, but the first-principles models developed are extensive and may require too many values to analyze continuously for practical purposes.

1.1 Motivation

The aim of this thesis was to advance the exploration of methods in modelling for the collaboration project between OsloMet and Veas, MaxBiogas. By analyzing collected plant data and composing it to usable datasets, different modelling approaches were tried and compare to find possible limitations of the simulation and control landscape.

1.2 Research questions

- How does modern Machine Learning methods and traditional System Identification methods compare to a state-of-the-art mechanically driven model?
- Can controllers be implemented to regulate biogas production of a simulated AD model within the operation limits of the plant specification?

1.3 Project description

In this project three different models were created using three different methods:
- West by DHI and the included AD model
- System Identification
- Recurrent Neural Network

Live data from Veas was extracted and in combination with lab analysis of daily sludge samples resulted in the composition of two datasets. One dataset, containing data from the lab analysis was created to fill the requirements for running simulation of the AD model in West. The next dataset contained plant data and was used to test how accurate data driven models can become using live data measurements.

CHAPTER

TWO

THEORY

2.1 Anaerobic Digestion Process

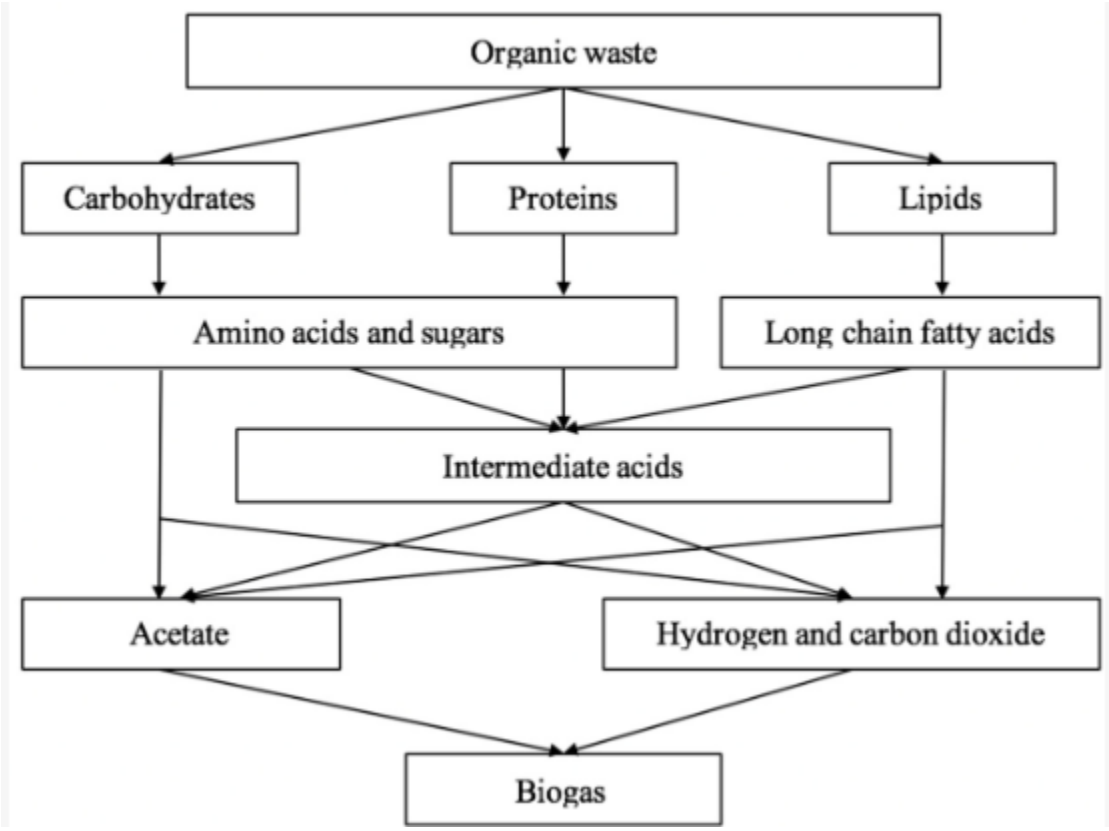

Figure 2.1.1: Simplified scheme of interaction between subprocesses in anaerobic digestion [4]

The anaerobic digestion process can be categorised as the sum of four subprocesses: hydrolysis, acidogenesis, acetogenesis, and methanogenesis [4]. Illustrated in fig. 2.1.1, hydrolysis is the first stage in the digestion process where carbohydrates, proteins and lipids are converted to amino acids, sugars and long chain fatty acids. Acidogenesis is the second stage and can output intermediate volatile fatty acids (VFA) including acetate, while acetogenesis, the third stage converts

the remaining products of hydrolysis, combined with intermediate acids from the acidogenesis process and produces hydrogen and more acetate. The final stage is methanogenesis where intermediate products like acetate and hydrogen is converted to methane.
There is a balance between these subprocesses that need to be held at an optimal equalibrium to maximize methane production. For instance, Acidogenesis is concidered a faster working process [5], that can result in a buildup of acids where VFA acidification has been reported being the cause of digester failure [6].

To maintain a stable balance between the subprocesses of AD, some considerations can be made when designing and operating an AD plant:
- Loading rate, the amount of organics fed to a digester per day in continuous digesters, can accelerate acidogenesis and lead to acidification and digester failure if kept too high [7].
- Hydraulic Retention Time, the mean length of time that liquids remain in a digester, is linked to Loading rate, but is found to have an optimal value of 15-30 days for mesophilic digestion [8].
- Temperature. Mesophilic (operation around 35°C) is found to digest slower and have a lower production yield, while thermophilic (operation around 55°C) have a higher heating cost. However, increased production rate leads to increased loading rates while avoiding acidification [9].

2.2 ADM1 and West by DHI

The ADM1 (Anaerobic Digestion Model No. 1) is a mathematical model for simulating the anaerobic digestion process. The ADM1 model was developed to describe and predict the complex biochemical reactions that occur during anaerobic digestion. It considers various substrates, microorganisms, and intermediate products involved in the process. The model accounts for factors such as pH, temperature, and the concentrations of different compounds to simulate the dynamics of anaerobic digestion accurately.

ADM1 has been applied in various fields, including wastewater treatment, biogas production from organic waste, and environmental engineering. It helps researchers, engineers, and policymakers understand and optimize anaerobic digestion systems, leading to improved efficiency, energy recovery, and waste management strategies [10].

West, a software developed by DHI is a simulation tool for wastewater treatment and includes an in-house developed AD model based on ADM1. The mechanically driven model simulates activity in ADM1 and calculates the states of the substrates, microorganisms and intermediate products in the AD and can directly estimate biogas production output by feeding the model with data of the influent. This method can be classified as a white-box model as the topology of the real process is known and mimiced as done in First Principles Modelling.

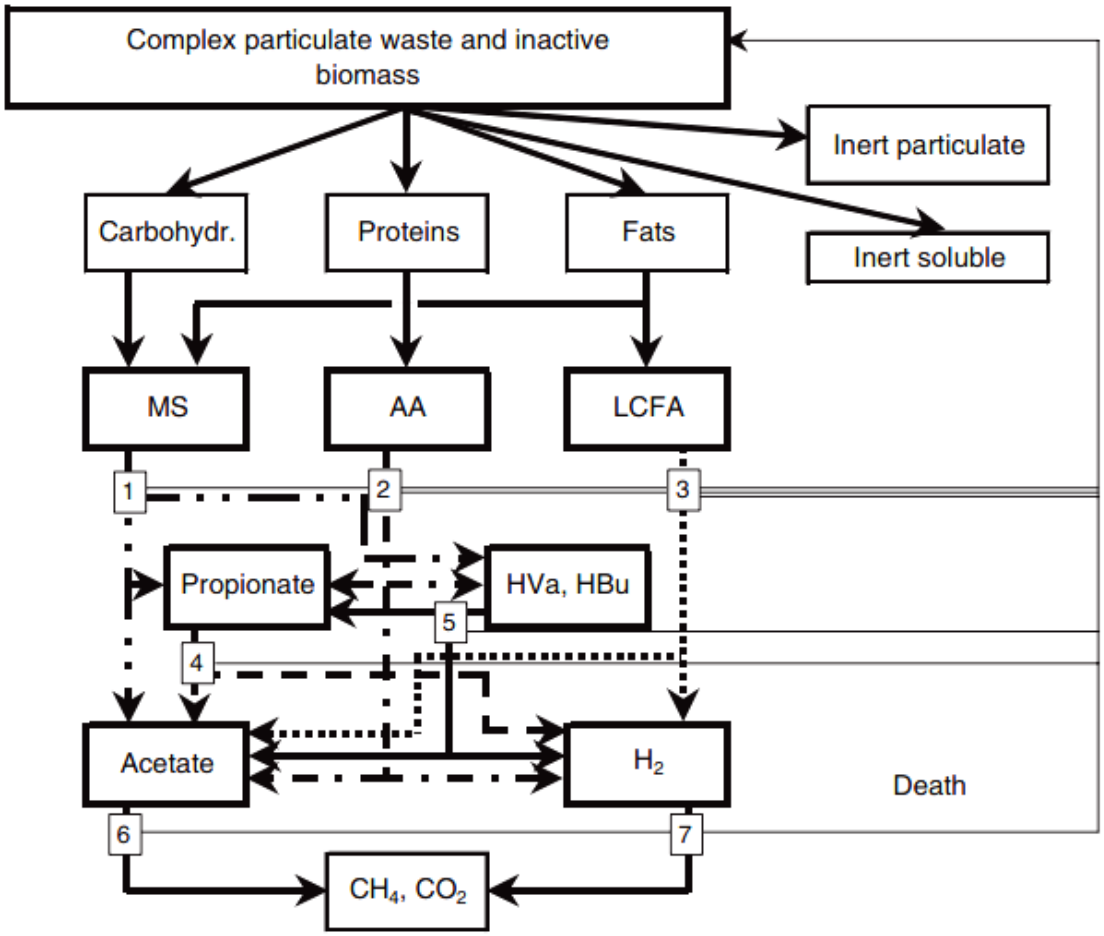

Figure 2.2.1: Conceptual drawing of ADM1 [10]

2.3 System Identification

Unlike for the AD model in West, System Identification does not use the white-box method of First Principles Modelling by using differential equations to estimate the response of the system. By using a data driven approach of using data from inputs and outputs of the estimated process, response function based on ordinary differential equations are chosen. To estimate the likeness of the resulting response to the actual process output, the constants of that response function is adjusted, using error minimization methods and performance metrics are saved after reaching optimum. Choice of response function can be based on trial and error or experience, so any understanding of the nature of the process to replicate by the proposed model would be useful and makes the method more like a grey-box approach.

2.4 Long-Short Term Memory

A Long-Short Term Memory node (LSTM) is a node used in recurrent neural networks. Unlike in simple neural network, like Multi-Layer Perceptron (MLP), these nodes has multiple tuneable parameters. Through an internal layout illustrated in figure 2.4.1, the LSTM has two "memory channels" moving horisontally through

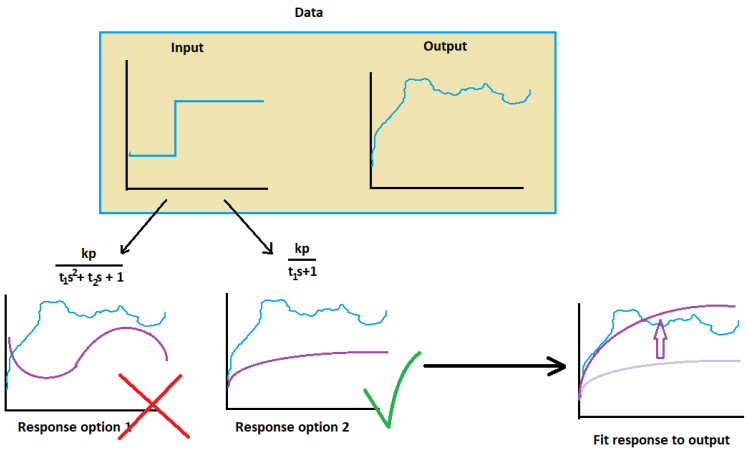

Figure 2.3.1: System Identification method

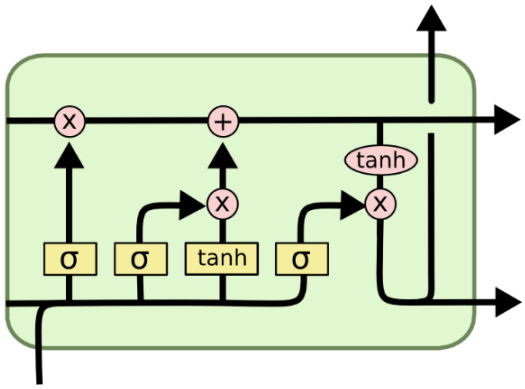

Figure 2.4.1: An LSTM node with layer height in vertical direction and layer stack in horisontal direction.

the node. The values in these channels gets altered as a sample is passed through the network, but is kept for the next sample. That way, unlike normal MLPs the output of an input sample depends on the samples passed through the LSTM beforehand. The properties and benefit of the top (long-term) and the bottom (short-term) memory channels makes LSTM suitable for time series data, where data is organized in a perticular order. Each path in the LSTM-node is scaled in array dimension according to input layer height and layer height in the hidden layers.

Creating a Recurrent Neural Network of LSTM-nodes to fit to input and output of a process does not require any prior information of any differential equations or responses of the process. The RNN topology, which defines the model design can be created through trial and error and can in most cases be defined as a black-box method.

2.5 Literature review

Previous work on prediction biogas production rates of anaerobic digesters have been tested with many different methods of machine learning. It seems to be

a popular method for the field, as scientific papers for prediction of AD using machine learnig methods are in abundance, although using system identification yielded unsuccessful. [11] got an R^2 of 0.38 with an LSTM model on two years of data, while increased this to 0.68 by incorporating dual-stage attention LSTM to the network (DA-LSTM). They did, however have 16 variables were used as inputs, which are alot of variables to correlate with the process output. This article [12] was also a study on a biogas facility linked to a WWTP and had the lowest amount of input variables, by eight. Although they did not use any neural network that had any way of predicting trends in time series data, they did compare complex machine learning models, like ANFIS-GP, they did try a regular MLP and got results in the range of R^2=<0.87-0.92> for the latter, indicating that normal MLPs can predict well given the right variables.
It is noteable, observing the inputs of the neural networks created in many of the articles that inputs to the models are measured states inside the AD[13][11][12][14]. It is apparent that one or more states from inside the AD volume like VFA values, pH and alcalinity are observable for all projects. This can be the culprit of how they achieved so accurate prediction for their models, but the number of observable input variables could also play a factor.

CHAPTER

THREE

METHODS

3.1 Strategy

The strategy used for this project was to create two different datasets based on data gathered from the biogas facility at Veas. When these were created, three different models were configured, optimized and trained. One model was created using West software, another using Machine Learning and LSTM nodes and the last using System Identification. Lastly, controllers was implemented to regulate the different simulation models. This enabled the possibility of optimizing the simulated operation of the biogas plant based on desireable performance metrics, such as biogas production rate and sludge used or total organic load.

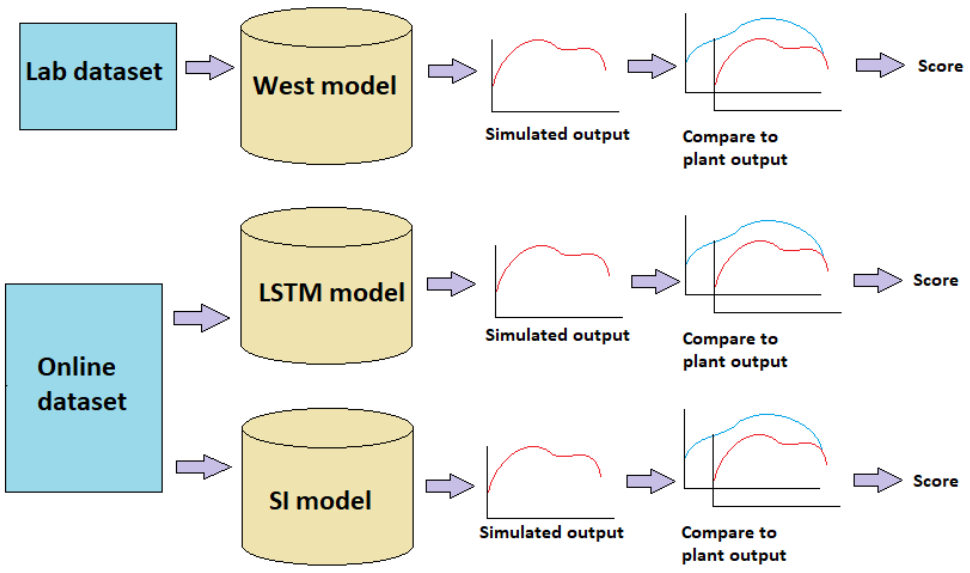

Figure 3.1.1: Overview of the modelling strategies for comparison

The first dataset was created based on data from lab samples collected daily during the month of July 2022. This dataset was created to feed the anaerobic digester model in West with the required data for the model to run the simulation for the collected data period. The second dataset created was solely based on online data captured live during operation. This dataset was created to run with

the models created using System Identification and Machine Learning methods. To compare the different models, the same date range was used to ensure comparable performance metrics.

3.2 Plant Layout

The parts of the biogas plant at Veas relevant for this thesis are two buffer tanks, a third intermediate buffer tank, four anaerobic digester tanks and three heat exchangers.

3.2.1 Inlet

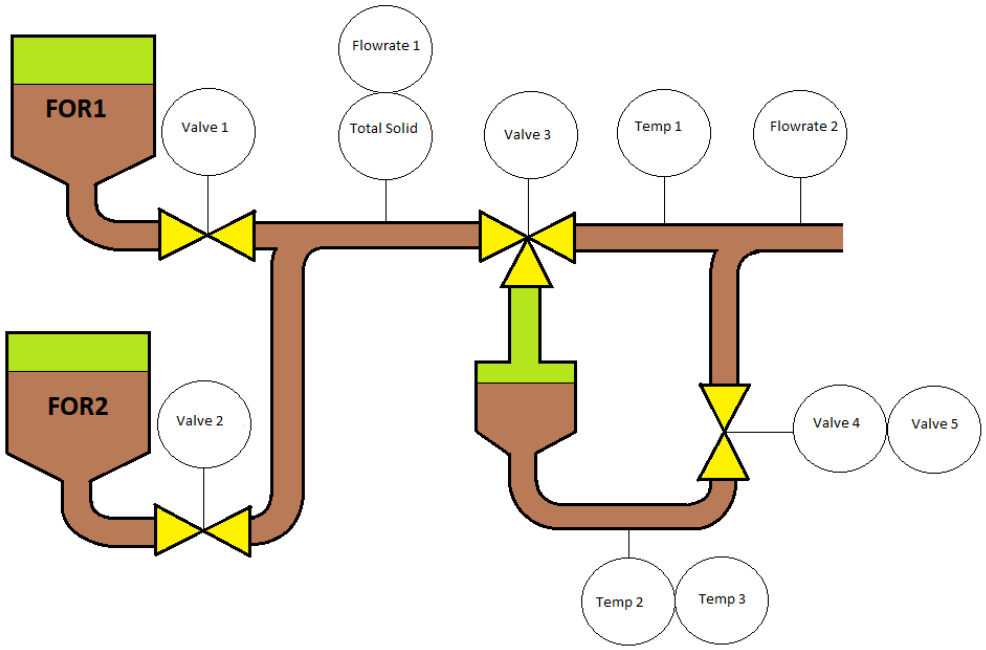

Figure 3.2.1: Simplified illustration of the sludge inlet layout with tags of data parameters used in data collection. Pumps and parallel pipes are excluded in illustration.

From the two primary buffer tanks, FOR1 and FOR2 (ref. fig.3.2.1), sludge are pumped by two pumps in parallel for redundancy, and which tank is being emptied can be determined by the valve position of Valve 1 and Valve 2 for tank FOR1 and FOR2 respectively. Both volumetric flowrate (Flowrate 1) and percentage of total solids (Total Solids) for the sludge inlet are measured after these pumps. To prevent excessive hydro static pressure to achieve adequate flowrate, a intermediate tank with a second pair of pumps are used, but can also be bypassed by a three-way valve. Since temperature readings of the inlet sludge consists of three temperature measurements at the three different pipe paths, the valve positions of valve Valve 3, Valve 4 and Valve 5 will be used to determine which temperature reading (Temp 1-3) will be used for every time stamp in the online data collection. It is necessary to mention, however that Flowrate 1 and Total Solids measurements does not necessarily coencide with the actual flow entering

the anaerobic digesters when the flow gets redirected to the intermediate buffer tank. Flowrate 2 can replace Flowrate 1 as it is located after the merging of the flow paths, but the Total Solids measurement does not have a replacement for this, and will be noted as a source of error or uncertainty.

3.2.2 Anaerobic digesters

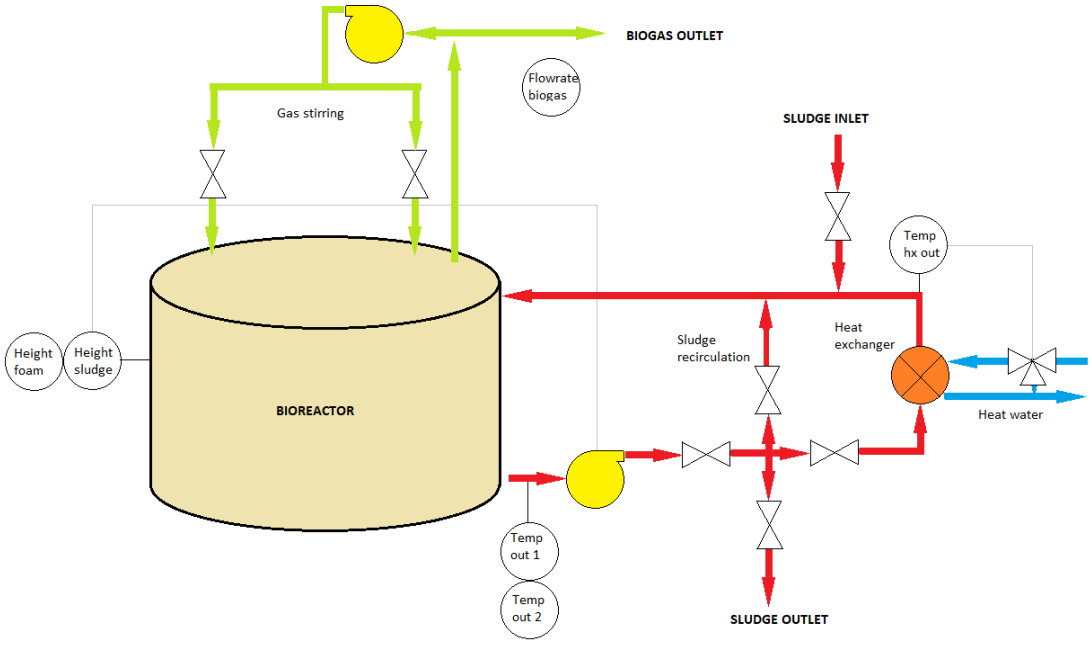

Figure 3.2.2: Simplified illustration of the layout surrounding an anaerobic digester tank. In all, there are four tanks working in parallel sharing three heat exhangers.

Figure 3.2.2 illustrates the pipe layout of a single anaerobic digester with relevant online measurement locations and valves used for identification of plant operation and data collection.

There are four anaerobic digester tanks at Veas, identical in physical properties. Each tank is controlled individually, but they all operate in a cyclic rotation of operational modes to ensure a continuous flow of sludge into the anaerobic digesters in a Round-Robin fashion. Each tank has an internal volume of 6000 m^3 with one inlet pipe and one outlet pipe. The outlet pipe is used for empying of the tank, recirculation of the sludge and heating of the sludge. Two temperature sensors are connected to the outlet pipe and indicates the temperature of the sludge in the respective tank. All four anaerobic digester tanks share the same heat exchanger system consisting of three heat exchangers working in parallel. Heat water is used as a heating medium of the sludge in the heat exchangers, and a bypass valve in the heat water regulates the effect of the heat exchanger by regulating the flowrate of heat water through the heat exchanger. The outlet of the sludge through the heat exchanger converge with the sludge inlet and the piping for sludge recirculation to one single inlet pipe into the digester tank. Inside the tank there are two live measurements. Height level of the sludge in the tank and the height of the foam created by sludge activity. The height level of the

tank is connected to the outlet pump to regulate the sludge height during the emptying phase. The biogas outlet is connected to a gas stirring system between each two tanks where pumps pushes gas through pipes down into the sludge to release any gas bubbles that may be stuck in the sludge and to potentially stir the solid material that sinks to the bottom of the tank. The final useful measurement in this part of the biogas plant is a flowrate signal of the biogas outlet flow.

3.3 Plant Operation

Plant operation of the four anaerobic digester tanks consists of six actions described in table 3.3.1.

Table 3.3.1: Actions during normal operation of the biogas plant

Filling	Pumping of new sludge through the inlet pipe
Emptying	Pumping of sludge from the tank through the outlet pipe
Heating	Pumping of sludge from the tank through the heat exchanger
Recirculation	Pumping of sludge from the outlet pipe to the inlet pipe
Gas stirring left	Pumping of biogas into the left side of the tank
Gas stirring right	Pumping of biogas into the right side of the tank

During normal operation of the biogas plant the tanks are regulated through a scheduled set of phases with static time intervals where each phase includes one or more of the aforementioned plant actions.
The first phase consists of filling and heating. The flowrate of the filling actions are controlled based on the amount of sludge in the buffer tanks in the inlet, while the flowrate of sludge through the heat exchanger is controlled to a fixed static value. These two actions are combined to prevent too low temperature values for sludge pumped into the tank through the pipe inlet that can affect the anaerobic activity. When the first phase is completed, filling and heating stops and recirculation starts. This second phase lasts twice as long as the first and third phase. The last phase consists solely of emptying of the tank.
Gas stirring of the tanks are operated on a schedule independent of the other actions. Tank 1 and Tank 2 as a pair operates on a gas stirring schedule similar to tank pair 3 and 4, but with different time intervals. The schedule operates of alternating of tank and then alternating side of the tank of where gas is being pumped. So, for a tank pair gas is first pumped into the left side pipes of the first tank before it switches to the other side. Then, the pumping of gas for stirring switches to the other tank, starting with the left side, before it switches to the

right side. When this cycle is completes it starts over from the top.

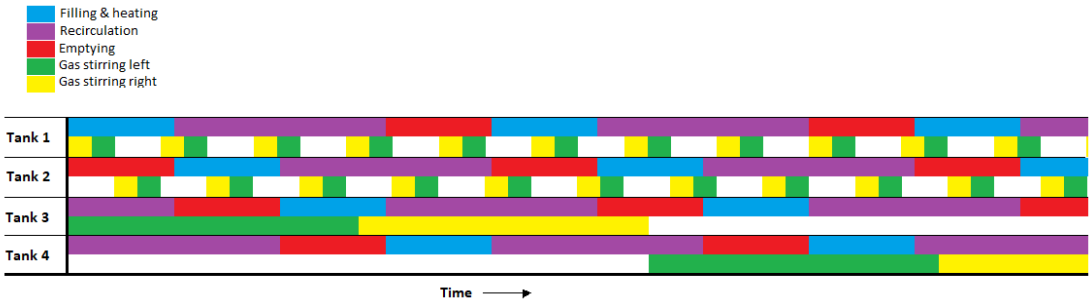

Figure 3.3.1: Plant operation schedule for all anaerobic tanks

Figure 3.3.1 illustrates all actions in a time scheduled plant operation. Note that the filling action is always active in one of the four tanks to ensure a constant flow from the buffer tanks. For each tank one of the actions of heating, recirculation and emptying is active. And since these actions all require sludge passing the temperature measurement in the tank outlet pipe, this temperature reading will at all times measure a moving flow from the tank and therefore can represent the tank temperature.

3.4 Data

Data used in this project came from two different sources. Data used for data driven model creation were extracted using ABB's Edge Insight and Linnea software used by Veas. Combined they provide the user an plant wide interface to observe plant states live and the tools to extract logged data for every measured parameter for different intervals. Data for the West simulation software came from online data from the Linnea software, but also lab analysis of daily sludge samples done to determine the composition of the sludge in the buffer tanks and the bioreactor tanks.
The strategy was to extract all online data relevant to the biogas reactor considered as either disturbance variables, state variables, manipulated variables or output variables. Then this data was to be analyzed to find important changes in operation, observe the nature of noise and outliers for data manipulation and to compare parameters against each other to find the parameters that mostly affect the reactor output.

3.4.1 Data collection

Data was extracted 13:30:00 05 February 2024 and data before 12:50:00 07 July 2021 was discarded as most data is not available up until this date. Online data available for collection can be divided into six sections:
- Recirculation
- Gas stirring
- Heat exchanger

- Inlet
- Outlet
- Tank

3.4.1.1 Recirculation

For each tank there is a valve and a flow meter that determines the flowrate of sludge from the outlet to the inlet for sludge circulation.

3.4.1.2 Gas stirring

In the subsystem of the biogas plant responsible for sludge stirring by pumping gas back into the sludge has two useful parameters available: pump speed and valve position for each side of the tank in each tank. Pump speed could be used to predict gas flow rate into the sludge. This could have been useful if it affected biogas output readings, but since those sensors are positioned after the gas stirring loop, the stirring process should not introduce any measurable disturbances or noise to the biogas output flowrate measurement. In total eight parameters were extracted from the gas stirring subsystem, valve position for each of the eight gas piping channels.

3.4.1.3 Heat exchanger

From the heat exchanger system, temperature values and flowrates were extracted. Four flowrate values were used; three for sludge flow into each heat exchanger and one for total hot water flow rate into the whole heat exchanger system. A total of ten temperature values were available and extracted. Six values for sludge temperature in and out of each of the three heat exchangers, three for hot water temperature out of each heat exchanger and one temperature value for the hot water flow into the whole heat exchanger system.

3.4.1.4 Inlet

Three different parameters related to sludge influent are available from online measurements: Volumetric flowrate, temperature and percentage of total solids.
Temperature are measured in two different places between the thickening tanks and the biogas reactor tanks, on both side of a hold up tank with a bypass pipe which can re-direct the sludge inlet flow past the second set of sensors. The second set of sensors after the hold-up tank are split into two channels in parallel with one pump each for redundancy. Since both flows after the hold-up tank and the pipe channel bypassing these pumps and hold-up tank are connected to valves logged in the online data, the temperature readings in each channel can be linked to the position of the adjacent valve. Using that method, temperature readings for the inlet sludge flow are determined by the temperature reading of the channel with a valve in the open position.
Percentage of TS are measured before the hold-up tank and volumetric flowrate between the thinkening tanks and the bioreactor tanks are measured both before and after the hold-up tank.

3.4.1.5 Outlet

Outlet data are values connected to both the gas and sludge outlet pipes. For the sludge outlet pipes the volumetric flowrate from all tanks are extracted aswell as valve position of each tank outlet to determine which tank the sludge is flowing from. For gas outlet normalized volumetric flowrate from each tank is extracted aswell as a value for all tanks expected to be a summation for each tank and not an independent measurement.

3.4.1.6 Tank

Two parameters are extracted from each tank: Sludge height and foam height and are the only available online measurements to determine bio reactor state measured within the tanks. Temperature measured from the sludge outlet is assumed identical to the sludge in the tank and is the last reactor state parameter. These measurements consists of two sensor readings, believed to be used for redundancy.

3.4.2 Data pre-processing

Collected data can consist of unwanted values due to a variety of sources. Disturbances in the measuring environment can manifest in noisy measurements, and can also lead to faulty measurements due to clogging. Sensors itself has their own specifications that determines how a sensor will react to changes in the measured medium. Resolution defines the step size in the sensor output, sensing range defines the maximum and minimum values it can measure, sensitivity defines how small of a change in the measured medium can be for the sensor to be able to respond and responsiveness can tell how fast the sensor can settle to a steady value after a change in the measured medium. Because of this, a raw measured value is not a direct representation of what the sensor tried to measure.
To improve on the correlation between a raw measured value and the actual unknown value, the nature of the actual unknown value would be useful for determine the quality of the measured data, and to find the optimal strategy for modifying the measured data.
One way of doing this is through the use of a Kalman filter. This uses First Principles Modelling, by knowing the differential equations of the system to predict the measured value as a combination of the actual value and one or several noise values.
Since First Principal Modelling will not be used in this project, there are no differential equations available to create a theoretical model for pre-processing of the data. Any modifications of the data will, for that reason, be based on experience and knowledge of the physical plant from plant operators and engineers.

After extraction of relevant data all signal values were treated equally in terms of filtering and smoothing. Any signals that were analyzed for having outlier values were subjected to removal of these values and replaced using linear interpolation.

Smoothing of the data was done to remove the oscillations in the signal from plant operation. The motivation of this action was that any optimization and

learning algorithm might try to fit their model to these high frequency oscillations. As the aim of the project was to try to find a model that can simulate a biogas plant in the range of hours and days, any oscillations with shorter wave length than this needed to be filtered out. A moving mean window was chosen for this filtering, to reduce these oscillations and applied to all signals collected.
The window size was chosen to not exceed 1/10th of the biggest time constant for the model created using System Identification. As the work of processing data was needed to be done before creation of any model, the time constant to use as reference was set to 1350 minutes based on previous work on the same biogas facility[15].

For both model creation methods of Machine Learning and System Identification, min-maxing was done to all variables. For System Identification modelling, normalization was also done to all variables since System Identification optimization functions using deviation variables. To normalize the variables all signal values are subtracted by the average of the signal value.

For any online data included in the lab dataset for the West model, all values were scaled to the correct unit required in the West software but no min-maxing or normalization was needed for that dataset.

3.4.3 Dataset creation

3.4.3.1 Online dataset

After all pre-processing of online signal values, a decision was needed to choose what variables to include for the models. Volumetric flowrate of biogas is the linked variables and the parameter the project was aimed at controlling. A correlation plot between the controlled variable and other relevant variables in the biogas plant was created to create a tool for determine what parameter variables to choose as the input to the later created models.

3.4.3.2 Lab dataset

Available from the lab analysis were results from sludge samples taken daily from monday through friday from 30 June 2022 through 29 July 2022. Samples were taken from two locations in the biogas plant. The data used for this project was lab analysis from the sludge samples taken from buffer tank 1. This was done in order to calculate what was pumped into the anaerobic digesters. Table B.1 lists all values measured from the daily sludge samples, while table B.2 lists all values needed for the anaerobic digester model in the West software.

The West model required all acids to be converted to theoretical COD equivalents in units $gCOD/m^3$. From [16] equation 3.1 was used to calculate the conversion from pure acid concentration in mg/l to theoretical COD weight for all acids where "n", "a", "c" and "b" represent number of carbon, hydrogen, oxygen and nitrogen atoms respectively in the acid molecule. Converted unit output is unchanged, but the sludge density was assumed to be 1000 kg/m^3, so mg/l was converted to g/m^3 directly.

$$thCODeq = \frac{(2n + 0.5a - 1.5c - b)16}{12n + a + 16b + 14c} \tag{3.1}$$

To create a complete dataset containing all required inputs for the West model, the following strategy was used:

- Sludge flowrate (Q) was imported from online data and converted to the proper unit.
- All acids were converted using equation 3.1 and imported, combining heptanoic and hexanoic acids as a combined parameter and likewise with isovaleric and valeric acid.
- S_su (sugars) were calculated using $S\_su = \frac{sCOD * X_ch}{tCOD - sCOD}$
- S_cat were imported directly from lab data and $S\_an = 14 - S\_cat$
- S_Inert was calculated as the difference between sCOD and the sum of sugars, VFAs and aminoacids: $S\_Inert = sCOD - S\_ac - S\_bu - S\_fa - S\_pro - S\_va - S\_su - S\_aa$
- Carbohydrates (X_ch), X_li (lipids) and X_pr (proteins) were imported from lab data and converted to correct unit
- X_Inert was calculated as $X\_Inert = tCOD - sCOD - X\_ch - X\_li - X\_pr$
- FSS (X_u_ig) was calculated as TS-VS and converted from percentage to g/m^3
- All remaining parameters needed for input into the West anaerobic digester model were set to 0.0001, as these parameters were assumed not present. They were, however not set to zero to avoid any potential division error during simulation.

3.5 Modelling

3.5.1 System Identification

The goal model using System Identification was to create an array of transfer functions between the array of input variables and the biogas output variable (see fig. 3.5.1). Using that model layout a single layer of transfer functions separates inputs from output and simplifies the model compared to the ADM1 model. The this layout with the signal inputs included in the online dataset, the model equation in Laplace space results to:

$$F_{out}(s) = u_1(s) * G_{p1}(s) + u_2(s) * G_{p2}(s) + ... + u_n(s) * G_{pn}(s) \tag{3.2}$$

Each transfer function for every input variable was identified for two different responses; zero-order and 1st-order response, both without time delay.

$$0 - order : G_{pn}(s) = K_p \tag{3.3}$$

$$1^{st} - order : G_{pn}(s) = \frac{K_p}{T_1 s + 1} \tag{3.4}$$

The online data was normalized by subtracting every value by the variable mean, and the dataset was split in two subsets with equal amount of sample data points, where the chronologically first subset was used for training while the last subset was used for validation. This method was used to avoid overfitting by

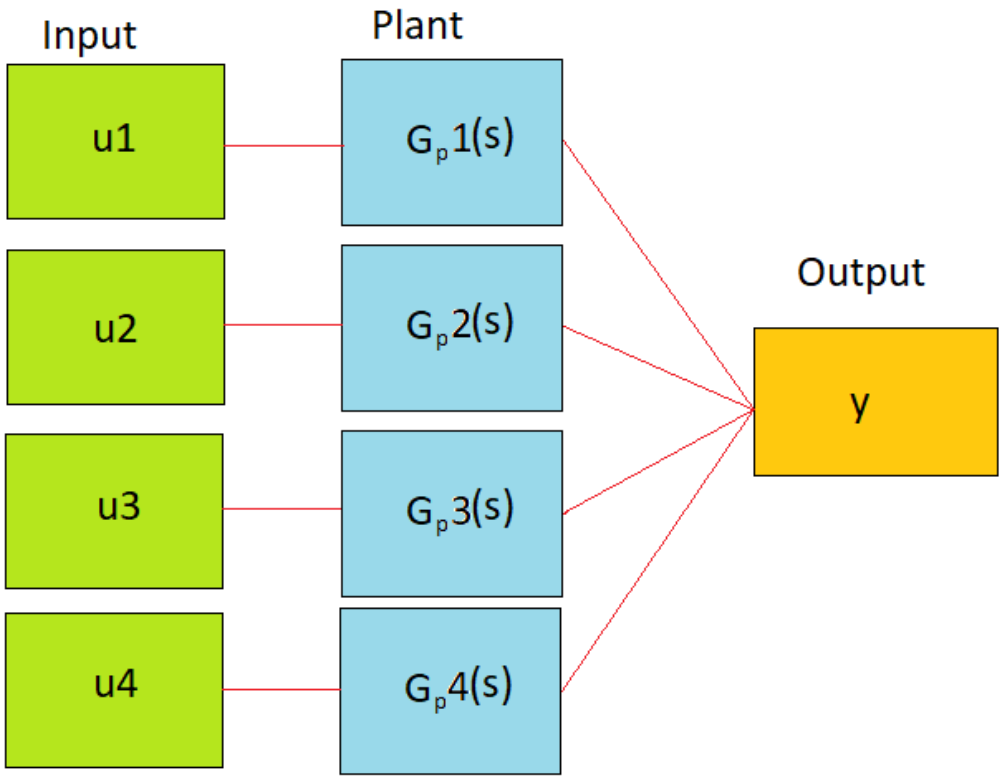

Figure 3.5.1: Flow chart of the model created using System Identification method

testing the model to a part of the dataset it has not been fitted for.
A loop was used to iterate through each combination of response types for all input variables, and estimate all transfer functions to fit to the output data. For every combination of transfer functions for the model, the coefficient of determination (R^2-index) was calculated after training to determine how similar the model output is to the actual plant data. For model output $\widetilde{y_i}$ and actual plant output y_i for i samples, coefficient of determination was calculated as follows:

$$\bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i \tag{3.5}$$

$$SS_{res} = \sum_{i=1}^{n}(y_i - \widetilde{y_i})^2 \tag{3.6}$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{3.7}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \tag{3.8}$$

3.5.2 Machine Learning - RNN

The created model architecture or topology did not have a defined shape for initial training runs, but rather iterations were used to find a topology that would output a stable response. Too small of a network and the network model would not have enough adjustable parameters to respond to back propagation to optimize its response to the resolution of the actual plant data output it tries to replicate. Too many tuneable parameters as a result of too large of a network and the model could output a oscillating response output. The number of hidden layers and the layer height was therefore initiated at small values and increased until oscillations started to become present.

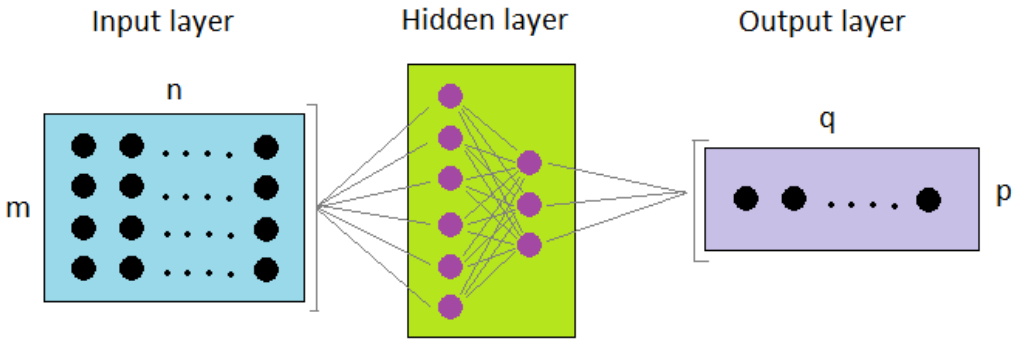

Figure 3.5.2: Topology of a LSTM network

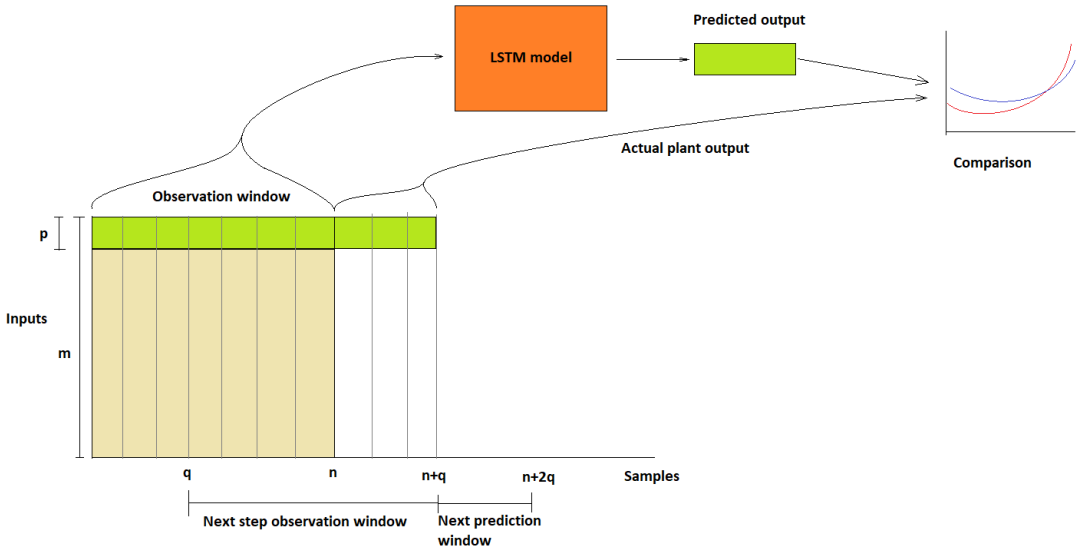

Figure 3.5.3: Sliding prediction windows for prediction of future plant output by LSTM model

Dimensions of the input layer were dependent on the number of samples the network needed to analyze to predict a future output and the number of input and

output variables. The dimension of the output layer is dependent on number of future samples the network is designed to predict for every step and the number of output variables. Where m is the sum of input and output variables, n is the number of samples in the observation horizon, p is the number of output variables and q is the number of predicted future output samples, figure 3.5.2 illustrates the topology of the input and output layers in the constructed network.

Illustrated by figure 3.5.3 shows the construction of the data fed to the RNN-model for prediction of future plant output. Two-dimensional windows of m variables and n samples got stacked after one another where every subsequent window was shifted by the length of the prediction horizon, q. For a dataset of length l, the resulting input data object was a three-dimensional data object of size $m * n * \frac{l}{n}$, while the output data object was a three-dimensional data object of size $p * q * \frac{l}{n}$.

3.5.3 West model

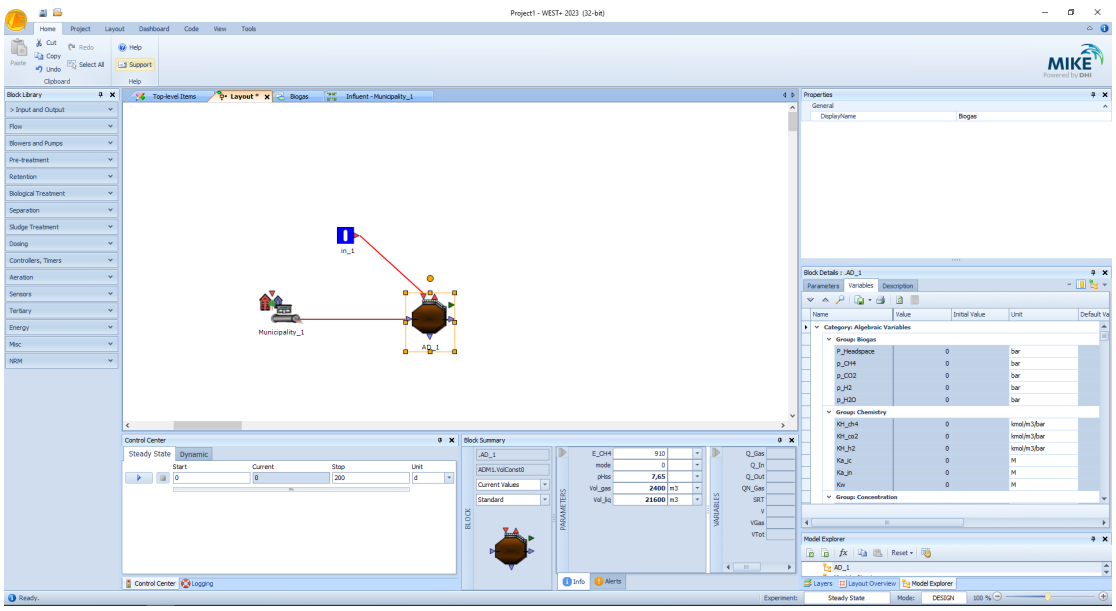

Figure 3.5.4: Layout of the simulation project in West software

The model created in West was set up using specifications directly adopted from the Veas facility. From the Sludge Treatment-pane in the West software, the Anaerobic Digester module were imported and all specifications were kept as default except for parameters listed in table 3.5.1.

All areas listed in table 3.5.1 are calculated from defining the anaerobic digerster tanks at Veas as four cylindrical volumes of 20 meters of height and internal volume of 6000 m^3 each. The adopted surface areas were calculating the resulting area on top, bottom and sides as the sum of the calculated surfaces of all four tanks. Volumes were calculated as the liquid height had a static height value of 18 meters with a two meter headroom occupied by the biogas volume.

Table 3.5.1: List of defined parameters set in Anaerobic Digester module in West software

Parameter name	Value
A_bottom	1200 m^2
A_top	1200 m^2
A_wall	4912 m^2
Vol_gas	2400 m^3
Vol_liq	21600 m^3

Temperature parameter for the Anaerobic Digester-module could be set to a static value, but since this parameter was available from the online dataset, temperature measurements were calculated as follows:
- There were two temperature measurements in the outlet of each of the four tanks. The average of these two values were adopted with 10 minute intervals.
- The average tank temperature value were summed up for all four tanks, weighted by their sludge height. Where T_i is average temperature reading for tank i, h_i is height of sludge in tank i and T_{tank} is overall temperature reading used in West simulation:

$$T_{tank} = \frac{\sum_{i=1}^{4} T_i h_i}{\sum_{i=1}^{4} h_i} \tag{3.9}$$

- All temperature samples were summed up and the mean value for each day to a daily temperature value to achieve the same sample frequency as the rest of the lab dataset.

Flowrate of sludge in the inlet was imported from the online dataset. values were converted from l/s to m^3/day by first converting to $m^3/sample$ (in online dataset one sample equals ten minutes). Then all sample values were summed up for each day to achieve daily flowrate values that, in similar fashion to temperature data, would have the same sample frequency as the lab dataset.

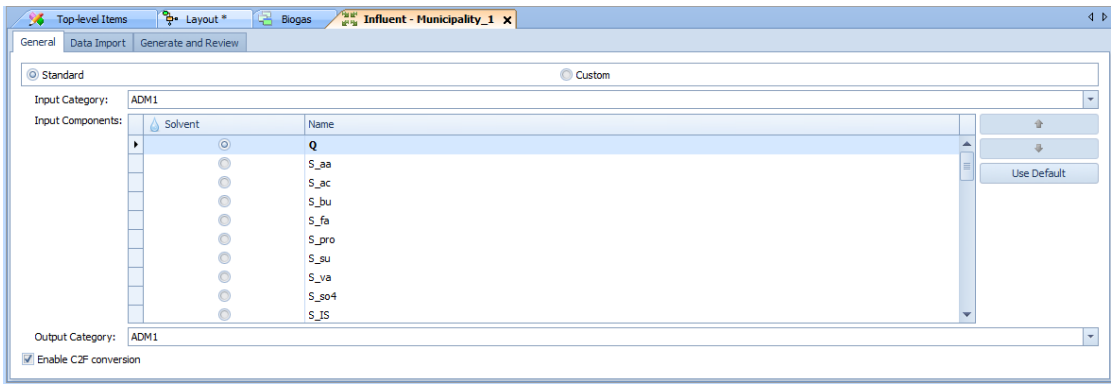

Figure 3.5.5: Setup of the Municipality Wastewater module in West. No fractionation was used as all ADM1 parameters were defined directly in the lab dataset

A Municipality Wastewater module from the Input and Output pane was also

imported, to simulate a flow into the Anaerobic Digester module. This module were set up to import and export ADM1 parameter data directly from the lab dataset (fig 3.5.5) including flowrate of the sludge influent from the online dataset.

3.6 Control

Controller strategies was developed for each of the three created models individually. For the model created using System Identification, both a PI-controller and an MPC-controller was implemented. For the model created using the West software and the LSTM model, PI-controllers were implemented. The aim of the work of creating closed loop systems by implementing controllers to the models was to regulate the simulated biogas production rates.

3.6.1 System Identification model

Using MatLab Simulink the model was loaded using an LTI block. All normalized input and output variable data from the online dataset were imported as a two-dimensional arrays with two columns each. The first column contained the time step iterations and the second column contained the data values.
The flowrate of sludge in the inlet was set as manipulated variable (MV), while the remaining input variables were set as disturbance variables (DV). All DVs were connected to the model, while the MV was replaced by the PI-controller. The argument for the input to the PI-controller was the error between the model output and a desired setpoint. The final diagram of the closed loop system is illustrated in figure 3.6.1.

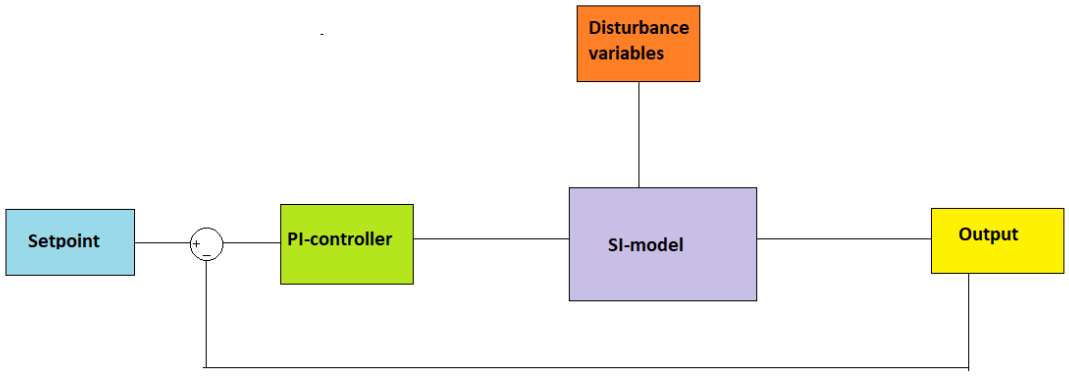

Figure 3.6.1: Simplified diagram of the closed loop system including a PI-controller and the SI-model.

3.6.1.1 PI-controller

The PI controller was tuned by using the method of Skogestad's Internal Model Control (SIMC) [17]:

For a PID controller:

$$c(s) = \frac{K_c}{\tau_I s}(\tau_I \tau_D s^2 + (\tau_I + \tau_D)s + 1) \tag{3.10}$$

Removing the D-term yields:

$$c(s) = K_c(1 + \frac{1}{\tau_I s}) \tag{3.11}$$

Calculating K_c and τ_I was done by observing the transfer function for the MV in the SI-model. For a first order function 3.4:

$$K_c = \frac{\tau_1}{K_p \tau_c}) \tag{3.12}$$

and

$$\tau_I = min\{\tau_1, 4 * \tau_c\} \tag{3.13}$$

where τ_c is a single parameter used to tune the controller to achieve a desired model output and regulation of the MV.

The controller was tested and tuned by using the actual plant output data from the online dataset as setpoint and then observe both simulated model output and the controller regulation of the MV and adjusting τ_c. The goal was to avoid an on-off regulation of the MV, while also get the model output to track the setpoint as close as possible.

3.6.1.2 Model Predictive Controller (MPC)

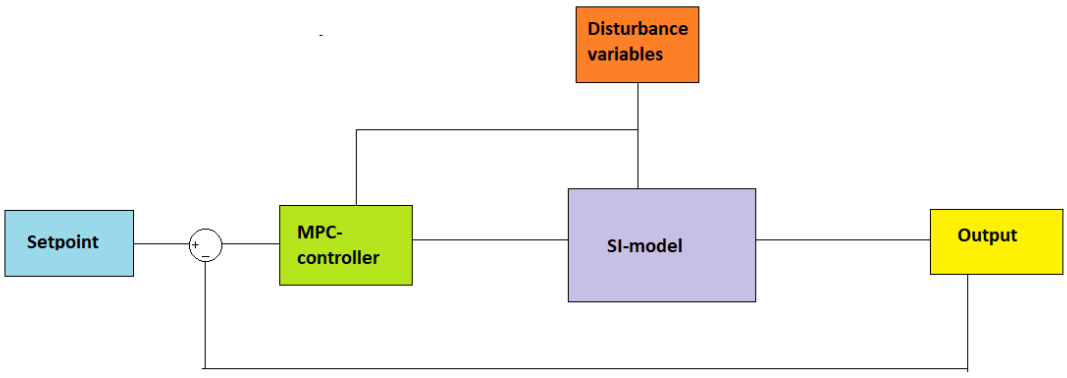

Figure 3.6.2: Simplified diagram of the closed loop system including a MPC-controller and the SI-model.

The MPC was implemented the same way as the PI-controller by loading the SI-model in MatLab Simulink and creating a closed loop system with the model, the MPC and the online data as DVs, MV and controlled variable (CV) (figure 3.6.2). As all the DVs in the closed loop system are measured live, these variables was considered observable, so were included as arguments for the MPC controller

to use.
Tuning of the MPC was done by initially setting the weight for the MV value to 0, and the weight for the error between biogas output rate and the setpoint to 1. Then, weight of rate change of MV was increased until controller instability were reduced and on-off behaviour of the MV was reduced to a movement pattern similar to the actual plant data so as to avoid unnecessary wear on the plant system.

CHAPTER
FOUR

RESULTS

4.1 Data

4.1.1 Online data analysis

4.1.1.1 Sludge height

Observing sludge height measurements it is apparent that the measurements are recording the emptying of Tank 4 in July 2023 (fig.4.1.1). This divides the operation into two chapters: Before and after emptying of Tank 4. There are sudden increases in tank height measurement values in Tank 4 during the emptying phase, but these value changes can be ignored as this was not part of the emptying operation.

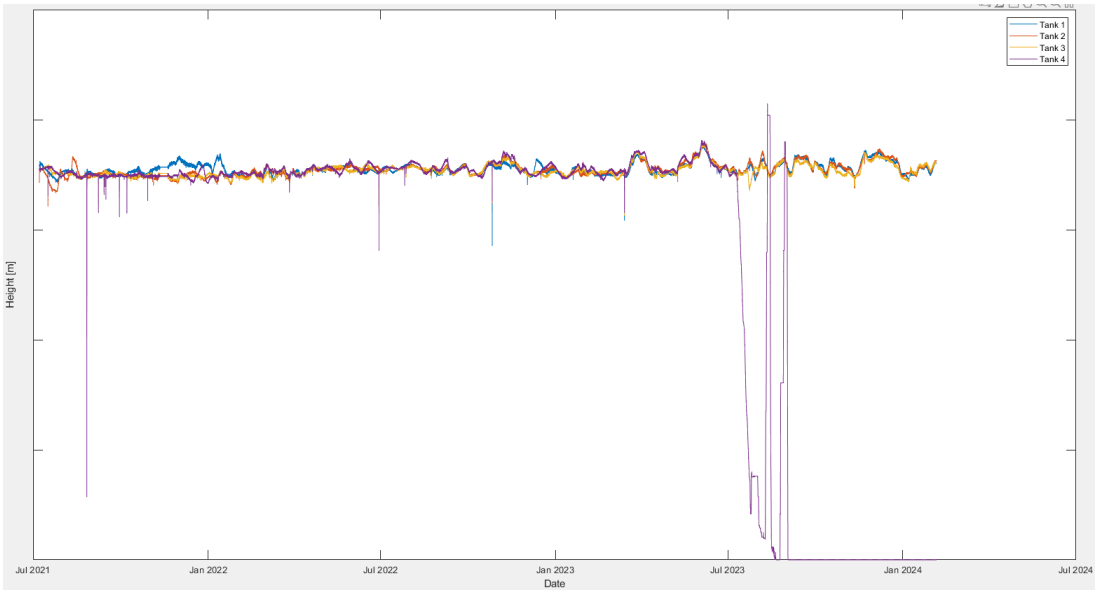

Figure 4.1.1: Sludge height measurements for all tanks

It is also observable that there are outliers and oscillations in the data, and would suggest that there is a need for filtering of these measurements. A last observation is also that the sludge height in all tanks stays fairly consistent and similar for all tanks during operation.

4.1.1.2 Tank temperature

Each tank has two sensors measuring the sludge effluent temperature for redundancy. By plotting all eight values (fig.4.1.2), a similar pattern emerges between sludge height (fig.4.1.1) and tank temperature: Tank 4 has sudden changes for temperature at the same time as as the sludge height is reduced. As the tank is taken out of the circulation through the heat exchangers, the temperature sensors, located at the outlet of the tank measures a declining temperature value. When the emptying phase is completed, the sensors are still in operation, but are measuring the ambient temperature that is decreasing until January 2024.

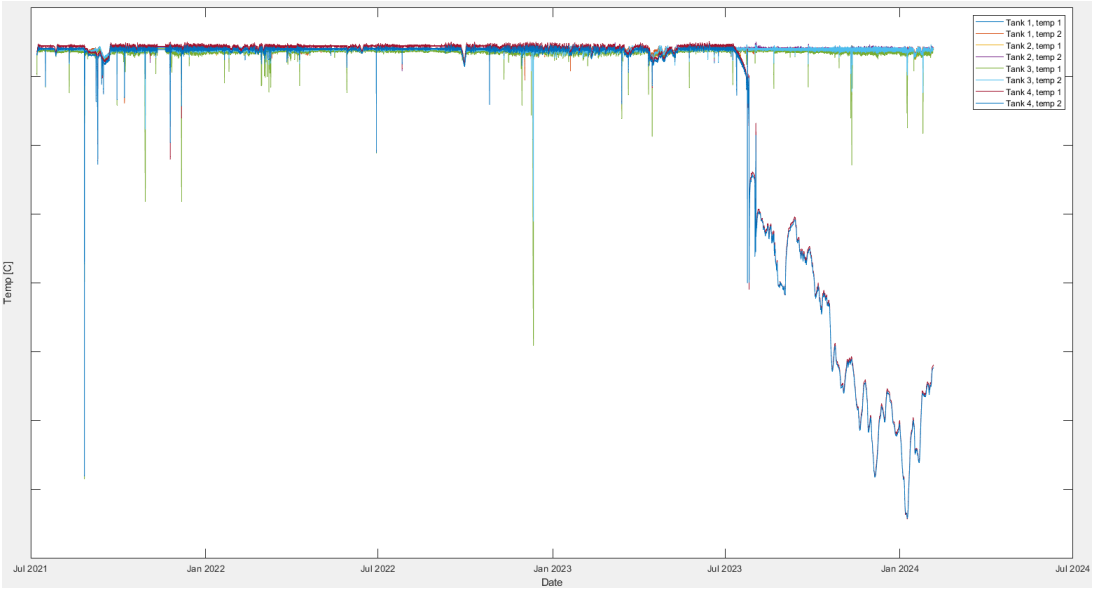

Figure 4.1.2: Temperature measurements for all tanks

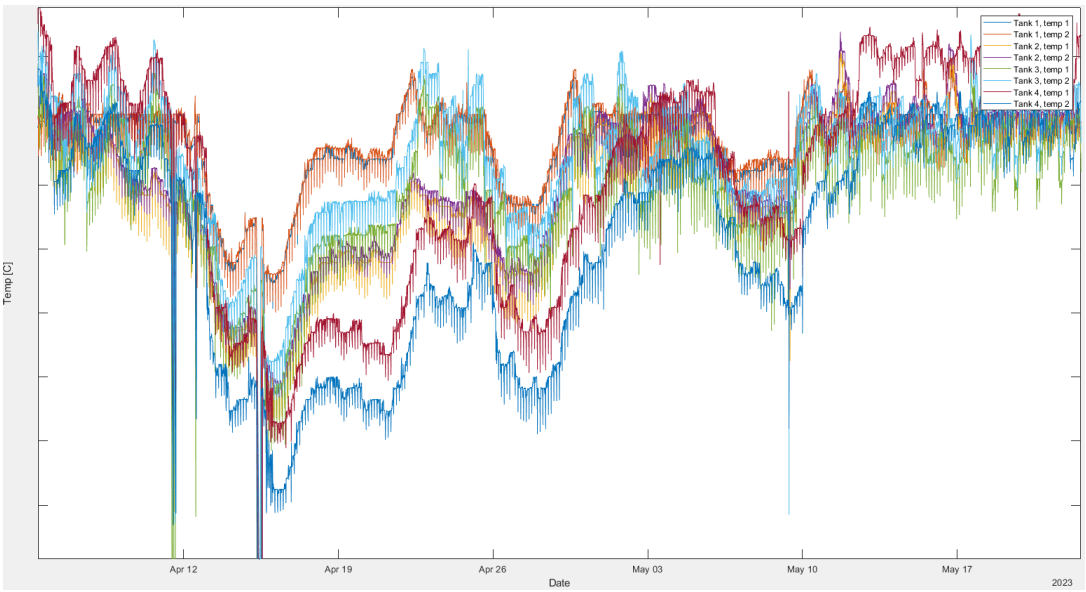

Figure 4.1.3: Sludge effluent temperature measurement oscillations

The wave length for the temperature measurement oscillations is observed to be consistent at approximately three hours before Tank 4 is taken out of operation,

but reduces in length after it is out of operation. These oscillations is a contribution of the Round-Robin operation of the heat exchangers. There are outliers present in these temperature values that is not confirmed by any of the operators at Veas, and will treated as faulty values.

4.1.1.3 Influent temperature

Temperature measurements in the influent (fig.4.1.4) consists of three sensors at three different places in the pipe network between the main hold-up tanks and the biogas reactor tanks. Two of the sensors are connected to two pumps in parallel after a secondary hold-up tank between the main sludge hold-up tanks and the bio reactor tanks, while the last temperature sensor is connected to a bypass valve of the secondary hold-up tank (fig. 3.2.1). The different temperature measurements are following the same trends of increasing and decreasing temperature with the seasons, but they are also oscillating with another shorter wave length. By comparing this high frequency oscillation with what buffer tank is being emptied, a pattern emerged, indicating that the buffer tank temperatures were different. The sludge fed to the two buffer tanks are coming from two different parts of the Veas facility, so this can be a plausible scenario. There are, however some noticeable spikes in the signals that should be treated as faulty measurements.

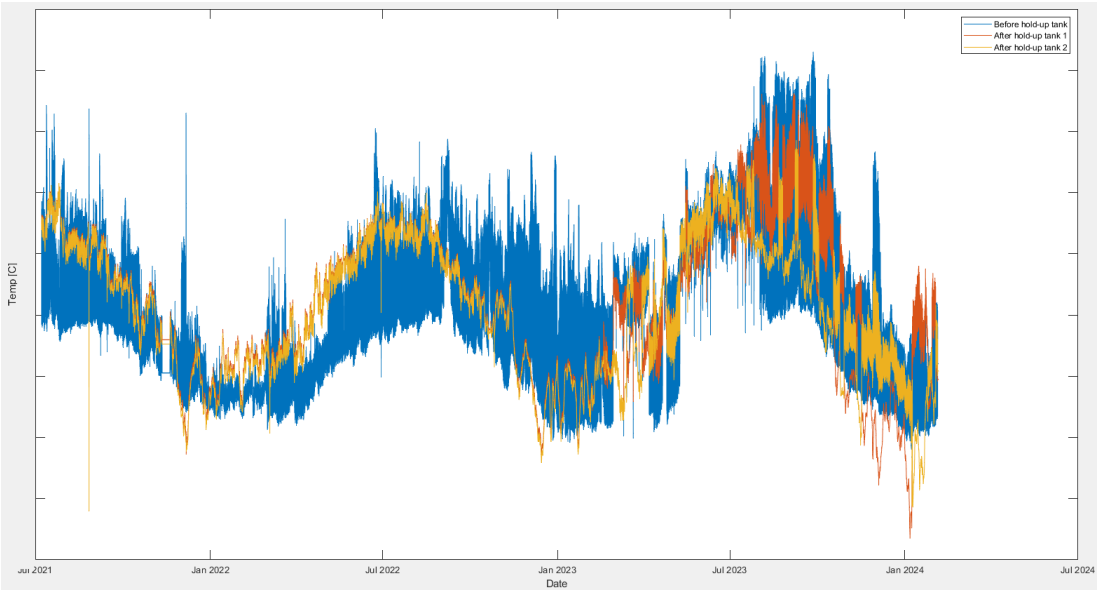

Figure 4.1.4: Sludge Influent temperature measurements

4.1.1.4 Influent flowrate

Studying the data from the flowrate measurements in the influent, there are noticeable "steps" in the data with height difference of 0.5 l/s, suggesting a result of operation control. There are, however, outliers present where values of 0 are logged. Confirming this with Process Engineer at Veas, Morten Rostad Haugen, these are a result of maintenance and other logged event during operation. These zero-values are therefore not considered as faulty measurements and will not be filtered out as missing values.

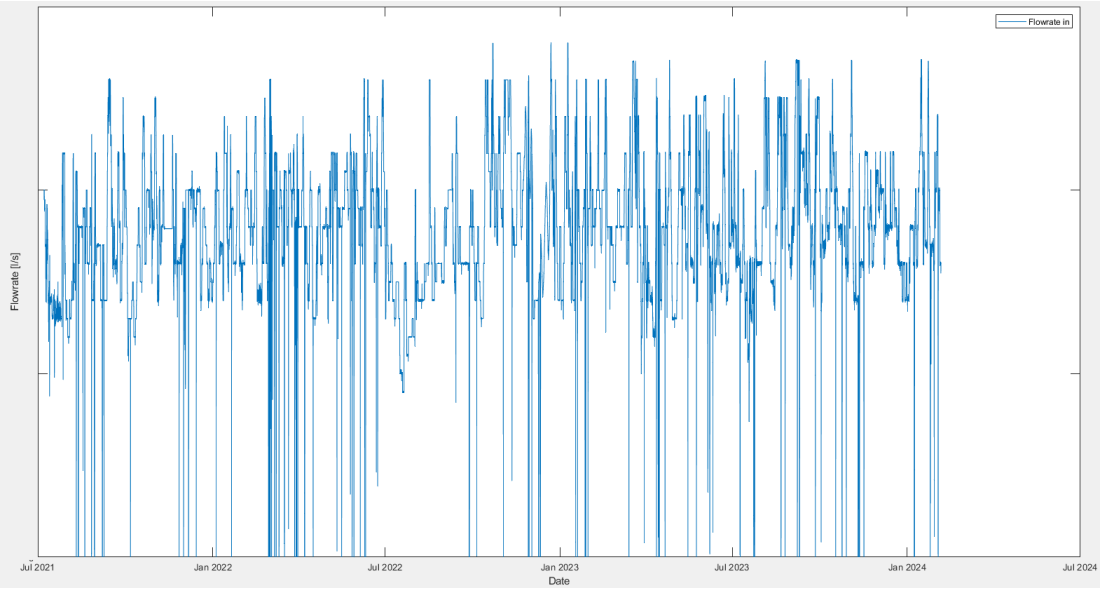

Figure 4.1.5: Sludge Influent flowrate measurements

4.1.1.5 Influent TS

Percentage of TS in the sludge influent stays consistent through the years, but experiences a high frequency oscillation similar to the inlet temperature signals. This, in conjunction with a steady frequency might indicate these oscillations as a result of plant operation and not as a measurement noise.
Confirming with Veas, one of the buffer tanks is receiving its sludge from a return flow from the outlet of the biogas plant. As sludge is being pumped out of the tanks, it enters a thickener tanks where excess water is removed and sent back into the biogas plant. This is the source for the sludge entering buffer tank 2, so for that reason it is reasonable to assume that the concentration of total solids in this tank is lower. As with influent temperature, these high frequency oscillations will therefore not be treated as noisy measurements, but rather as a part of normal plant operation.
There is also a static measurement value for TS in the sludge influent between 11 Nov and 19 Nov 2021, which reduces the data range for the final dataset.

4.1.1.6 Effluent Biogas

Flowrate of produced biogas out of the bioreactor is logged both as a parameter for total biogas production flowrate for the whole plant and as single parameters for each tank. Comparison between the total signal and the sum of all the four individual signals gives an IAE = 66620 and an average difference of 0.4906 per sample. Dividing this average difference with the average value for total biogas flowrate in the data, a relative average difference between the two parameters can be determined to see how similar they are to each other. With the average of total biogas flowrate being 1194.4 [Nm3/h], the relative average difference between the two parameters equals 0.041%, indicating that these values are near identical. The production rate has a trend of reduced values during the summer holidays, there are oscillations with both high and low frequencies in the signal, and outliers observed as spikes in the data. There is also static values in this data from 30 Oct

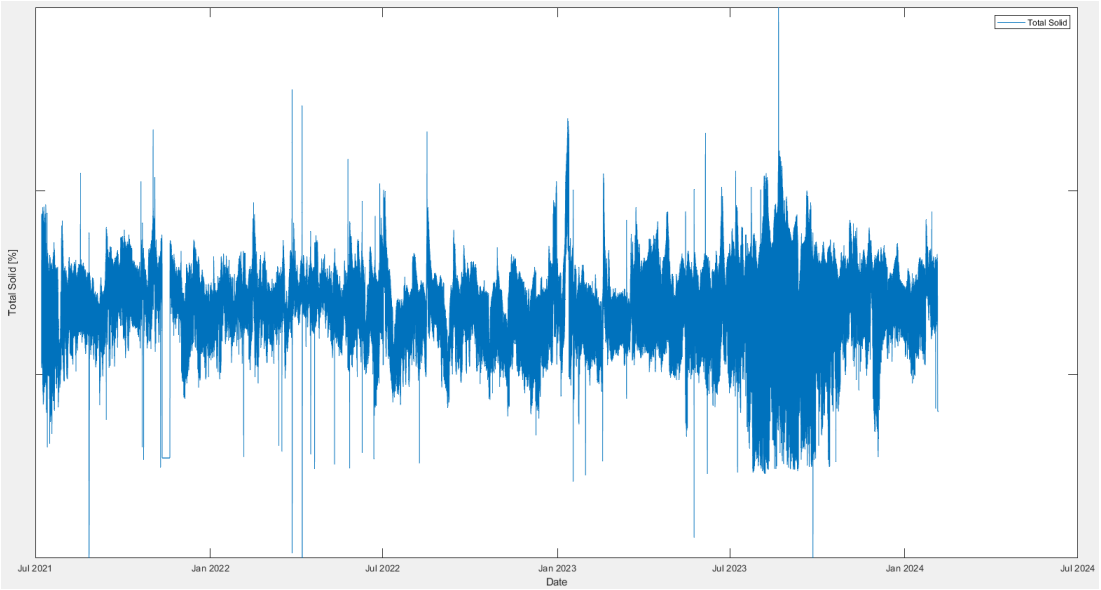

Figure 4.1.6: TS sludge influent

to 10 Nov 2021.

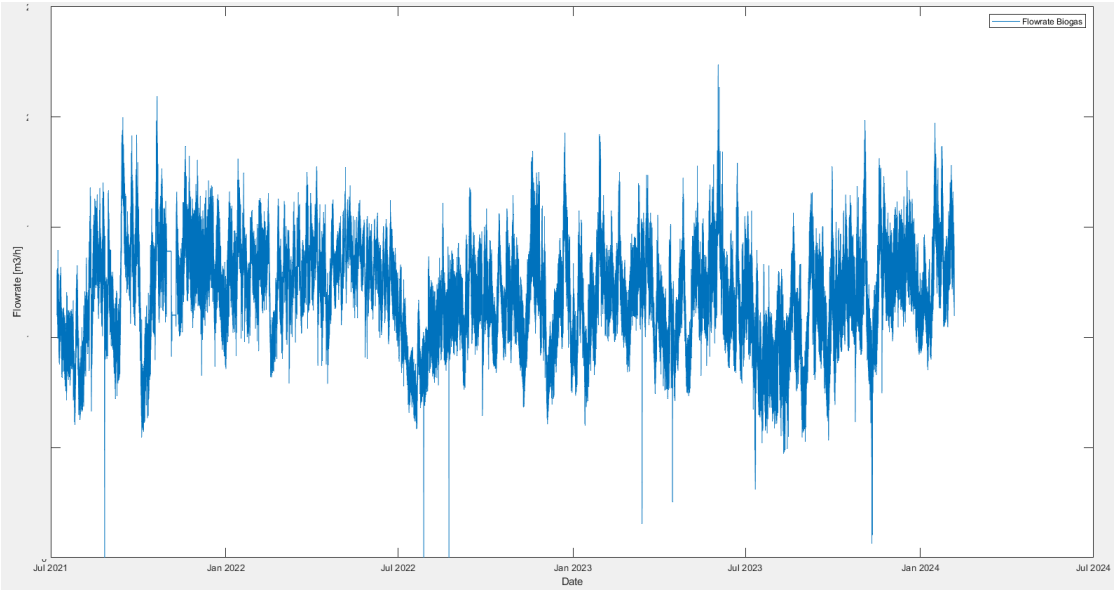

Figure 4.1.7: Biogas production measurements

4.1.1.7 Dataset range

The available data for analysis ranged from 07 July 2021 at 12:50:00 to 05 February 2024 at 13:30:00. However, due to emptying of tank 4, operation of the biogas plant changed during this date range and resulted in an end date for the created dataset at the start of the emptying phase. Noted previously, there are noteable static values in the data for sludge influent temperature and TS%. This can affect the modelling part of the project, so the final dataset date range will be reduced to start at 00:00:00 19 Nov 2021 and end at 10 July 2023 at 09:00:00.

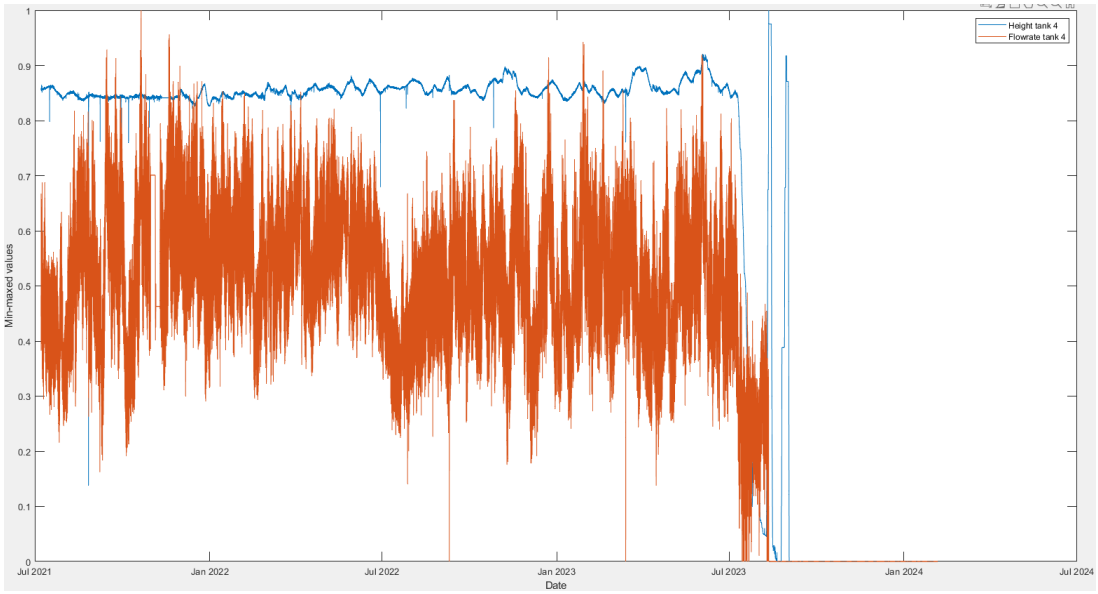

Figure 4.1.8: Sludge height and biogas flowrate in tank 4 (min-maxed values)

4.1.2 Dataset creation

The final online dataset was created to include both the output biogas flowrate and all input parameters fed to the created models. To identify what input parameters to use as inputs, a correlation plot was created. By minmaxing all parameters, the following parameters were included in the analysis:

Input:
- Temperature in the sludge inlet(T in)
- Flowrate of sludge in the inlet(F in)
- Average temperature in the tanks (T tank)
- Total height of sludge in all tanks (H sludge)
- Total foam height in all tanks (H foam)
- Total solids percentage in the inlet sludge (TS in)

Output:
- Volumetric biogas production (F out)

Using the Pandas library available for Python, three methods were available for correlation comparison:
- Pearson
- Spearman
- Kendall

Table 4.1.1 show highest correlation for flowrate and percentage of total solids inlet sludge, and height of foam in the digester tanks. Temperature and height of sludge in the tanks showed the least correlation to production rate, but since these are regulated parameters in normal plant operation, these values shows less

variance and might affect any correlation calculation. Temperature in the sludge inlet showed a negative correlation for all of the three methods, which means that the correlations found expects a reduced production rate for increase in inlet sludge temperature.

Table 4.1.1: Correlation values between selected parameters and biogas production rate using Pearson, Spearman and Kendall method

Input	Pearson	Spearman	Kendall
F in	0.6156	0.6308	0.4633
T in	-0.2287	-0.2612	-0.1703
TS in	0.2494	0.2896	0.1990
T tank	0.0663	-0.0507	-0.0341
H sludge	-0.0449	-0.1159	-0.0766
H foam	0.3121	0.3130	0.2202

Choosing parameters to include in the online dataset ended up being:

Input:
- Sludge flowrate in (F in)
- Temperature in (T in)
- Total Solids in (TS in)
- Height of foam in tank (H foam)

Output:
- Volumetric biogas production (F out)

4.2 Modelling

4.2.1 System Identification

The System Identification algorithm was run using MatLab version R2023a on a desktop computer running Windows 10. The specifications on the computer was an Intel Core i5-6500, 16 GB of DDR4 RAM and an Nvidia GTX1080Ti graphics card.

After testing all combinations of transfer functions for the model, the combination with the best coefficient of determination was with a zero-order function between the $TSin$ and $Fout$ and the rest as first order functions (see table 4.2.1).

The dataset was divided into two equally large subsets, and the later, test dataset was used for performance evaluation.

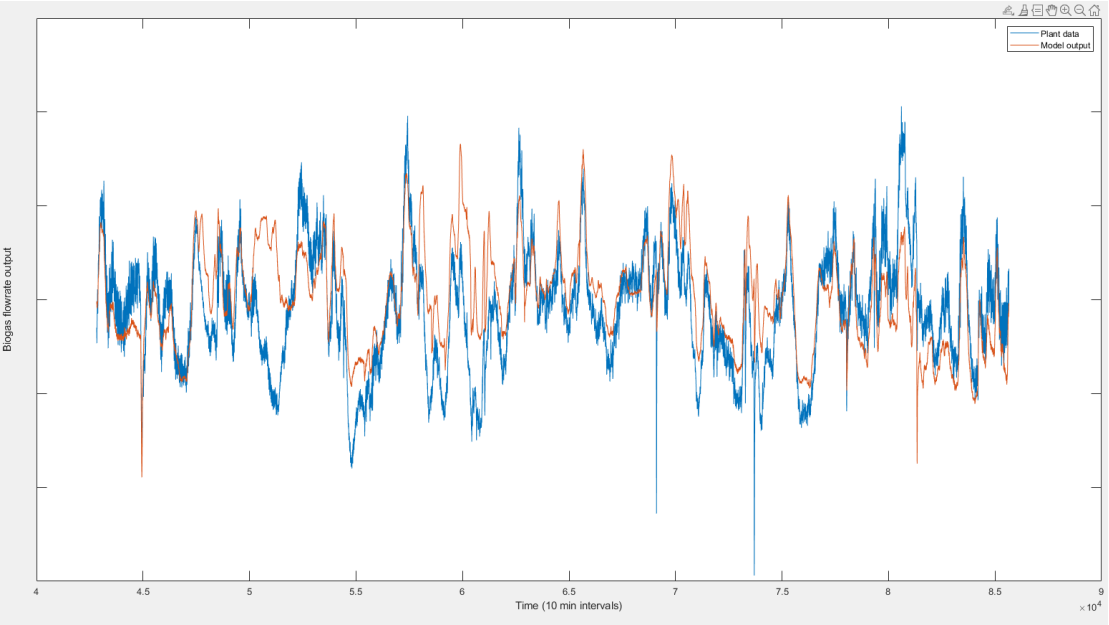

Figure 4.2.1: Biogas output of actual plant data (blue) and System Identification model output (orange) for testing subset of online dataset

Table 4.2.1: Parameters for transfer functions for the fitted model with the highest coefficient of determination

Input variable	K_p	τ_p
F in	0.7599	59.64
T in	-0.3198	3432
TS in	0.4205	0
H foam	0.0748	3.662

Table 4.2.2: Coefficient of determination (R^2-index) between actual plant output and SI-model output for training and testing subset of online dataset

Dataset	R^2
Training	0.6944
Testing	0.3368

4.2.2 Recurrent Neural Network

The models was created using Visual Studio Code running on the same computer as was used on the System Identification-work. Tensorflow and Keras library was used to set up the LSTM-model in Python and Pandas and Numpy was used for handling and construction of the datasets.
The dataset was split in two parts with the same date ranges as for SI-model. The first date range subset was used for training, while the second set was used for testing and calculating performance in terms of accuracy to actual plant data.

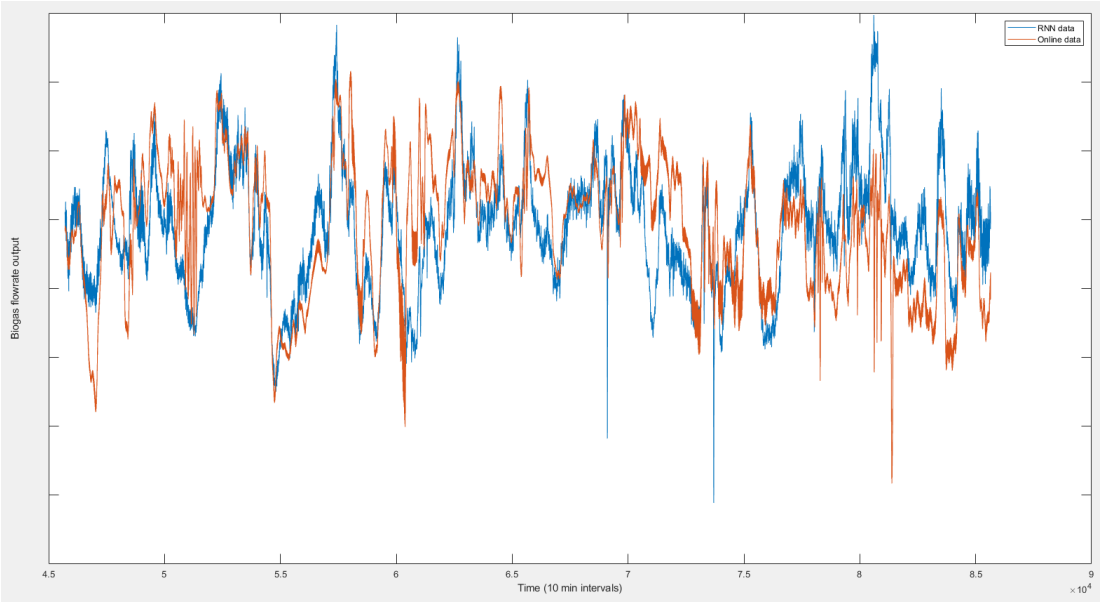

Figure 4.2.2: Biogas output of actual plant (orange) data and RNN model (blue)

Table 4.2.3: Coefficient of determination (R^2-index) between actual plant output and LSTM model output for training and testing subset of online dataset

Dataset	R^2
Training	0.6469
Testing	0.1355

4.2.3 West model

The West simulation outputted values in m^3/day and was converted to $m^3/hour$ to match with the online data and the output of the other models. Although the input in the simulation had sample frequency of one sample per day, the exported data had 144 samples per day, resulting in equal sample frequency as the other models.

To compare performance metrics between the models, both the SI model and the LSTM model outputs were compared for the same lab dataset date range. Neither model was trained or fittted to that part of the online data range.

4.2.4 Model comparison

To compare performance metrics between the models, both the SI model and the LSTM model outputs were compared for the same lab dataset date range. Neither model was trained or fitted to that part of the online data range.

New coefficients were also calculated for the new lab dataset range. Doing this created a performance metric to compare all three models during the same conditions.

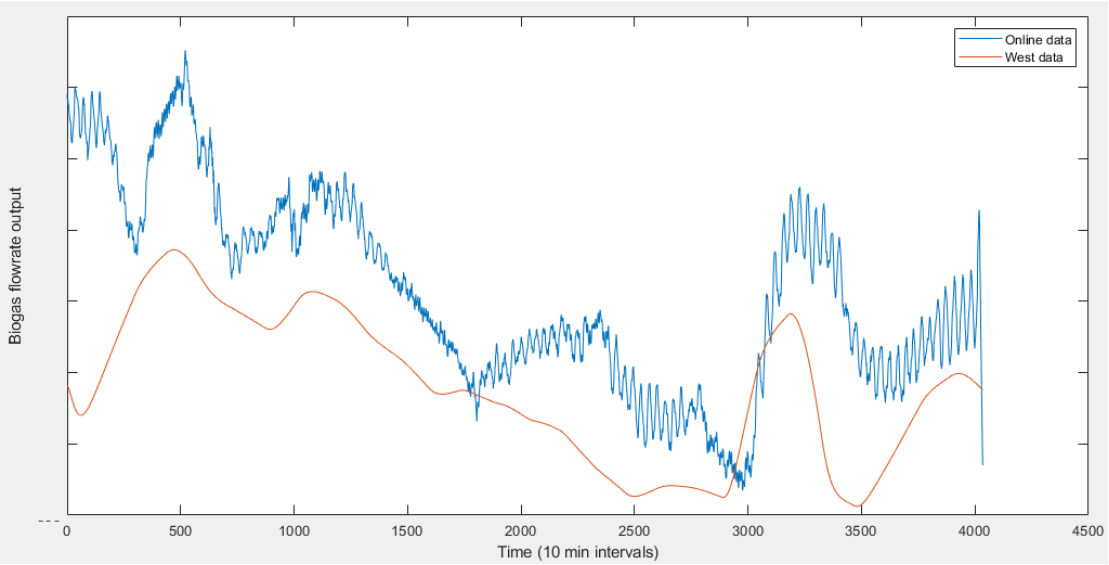

Figure 4.2.3: Biogas output of actual plant data and West model output between 30-June-2022 and 28-July-2022

Table 4.2.4: Coefficient of determination for all three models against the online biogas production rate during the span of the lab data range

Model	R^2
SI	0.5009
LSTM	-3.139
West	-0.3726

4.3 Controllers

4.3.1 System Identification model

4.3.1.1 PI controller

Using flowrate of sludge into the anaerobic digesters as the manipulated variable for the PI-controller to modify so to regulate the controlled variable, biogas flowrate out of the plant, the controller was tuned by the transfer function between them in the SI-model:

$$G_{fin}(s) = \frac{0.7599}{59.64s + 1} \tag{4.1}$$

By tuning the PI-controllers parameter, τ_c, the resulting PI-controller parameters ended at:

The resulting closed loop system was then tested by using the actual plant output from the dataset as the setpoint, testing if the system would use similar sludge flowrate into the system to achieve equal biogas production rates as the actual plant. Figure 4.3.1 and A.2 shows closed loop performance for this setpoint.

Table 4.3.1: Parameters of tuned PI-controller

Parameter	Value
τ_c	100
P	0.7848
I	0.0132

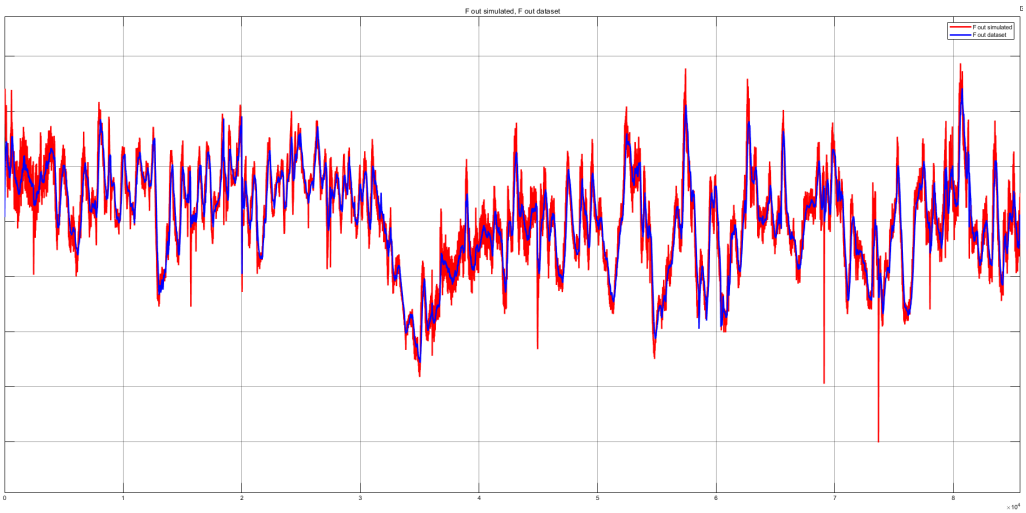

Figure 4.3.1: Biogas production rate of the anaerobic digester. Blue graph represents the actual data values, while red are the flowrate controlled by the PI-controller

4.3.1.2 MPC controller

Setup of the MPC-controller was to use the same input variable as manipulated variable as for the PI-controller (see table 4.3.2). MV rate weight was not used and set to 0 to be able to compare performance with the PI-controller. MV rate weight, penalizing changes in MV reduces the amplitude of the controller behaviour, and was increased until the resulting adjustment the MPC made avoided saturation at the limits or excessive oscillations. MV limits were set to the minimum and maximum values of the sludge flowrate into the anaerobic digester in the dataset.

Table 4.3.2: Parameters of tuned MPC-controller

Parameter	Value
Sampling time	1
Prediction horizon	120
Control horizon	60
Lower MV limit	min(F in)
Upper MV limit	max(F in)
CV weight	1
MV weight	0
MV rate weight	20

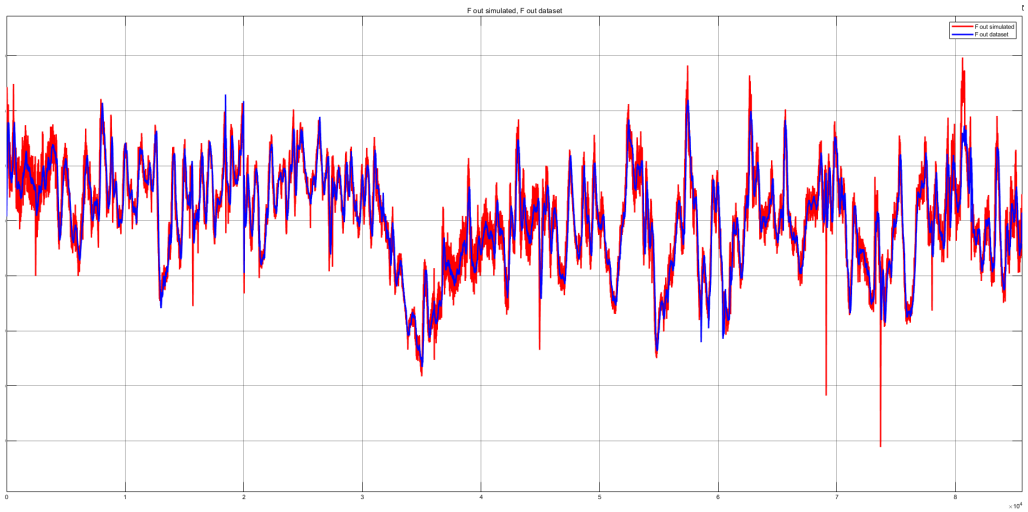

Figure 4.3.2: Biogas production rate of the anaerobic digester. Blue graph represents the actual data values, while red are the flowrate controlled by the MPC

4.3.2 Recurrent Neural Network model

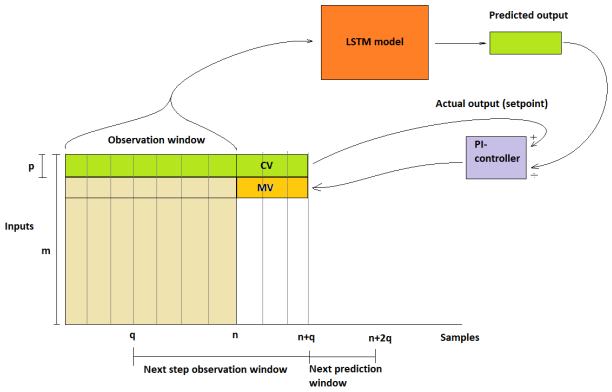

Figure 4.3.3: Layout of closed-loop integration of PI-controller with RNN-model

The tuned PI-controller from the SI-model was adopted directly while keeping the same parameter values. For each prediction window passed into the RNN-model, the output prediction would be compared to the actual plant output for the same time sample window. For both prediction output and actual output, the array of sample values was summed up and the difference in sum between the actual output and the prediction became the error argument which was passed back into the PI-controller. The new error argument would then produce a new output out of the PI-controller that would become the new value for MV (F in) for the next iteration of prediction for the RNN-model.

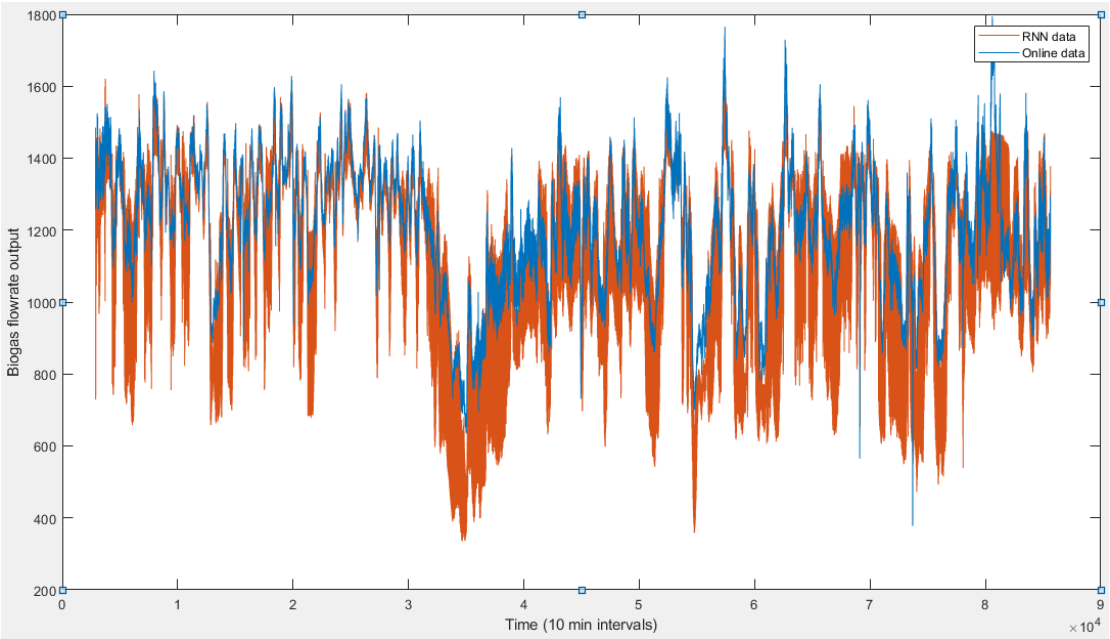

Figure 4.3.4: Flowrate of biogas production. Orange graph represents the output of the closed loop system of RNN model and PI-controller, while blue graph is the setpoint (actual plant data).

CHAPTER

FIVE

DISCUSSION

5.1 Discussion

5.1.1 Datasets

The data available were extensive and the availability of the data resulted in an extensive online dataset covering flow rates, height measurements and valve positions for the entire biogas facility. Comparing the data available to other similar scientific simulation projects revolving the anaerobic digester, data on sludge composition and AD states could result in more accurate simulations if available.
Looking back on the filtering on the dataset, the created online dataset could probably need more filtering, as oscillations as a result of plant operation was still present after pre-processing. This could be the culprit for the oscillations of both the RNN-model and the SI-model for which it was trained.
The lab dataset was a complex task to combine, but the sludge sample analysis and the data it produced to compose this dataset resulted in a simulation run trending in the same direction during the dataset range for the AD model in West.

5.1.2 Modelling

There were clear differences in the result of the different methods in modelling applied in this thesis. System Identification showed promising result although the R^2-index dropped between the subset used for training and the subset used for testing. Observing figure A.1, the model follows the actual output more closely than any of the other two in that sample range, even as that range falls outside the training dataset.
RNN-model based on LSTM-nodes proved to be a complex task in tuning. Oscillation could occurr for small network topologies, down to two hidden layers and five nodes in each. If the batch size was set too small or the learning rate too low, oscillation or unwanted behaviour could easily arise. Studying the methods adopted by others in their research papers, some scientific studies applied a hybrid approach, by combining different types of machine learning network types.
The simulation run done using the AD-module in DHIs West tool showed more stable results than the other two during the date range it was fed. The output

of the model increased and decreased as the actual output did the same, even as there was an offset between the two signals. This can undoubtedly be corrected by further tuning of the AD-module settings in the software

5.1.3 Controller

All controllers implemented in the project were able to regulate the Controlled Variable with varying accuracy. For the SI-model both the PI-controller and the MPC-controller were able to track the setpoint without extensive deviation while also being be tuned to reduce controller actions applied to the MV.
Applying the PI-controller to the RNN-model worked, while still being sub optimal. Oscillations dominated the signal spectrum and should be avoided for all physical systems. Oscillations trained inside the RNN-model could initiate this, but it can also result from undesirable controller tuning. An idea of creating a learning algorithm that could autotune the PI-controller was an idea, but the project time frame would not allow it.

5.1.4 Future work

The MaxBiogas collaboration between Veas and OsloMet will continue and there are challenges that can be handled as the next step in the project. Predicting future influent to the holdup tanks could be beneficial, so the setpoint control can be regulated to make room for potential increases in influent or organic concentration. Trying to replicate or directly adopt more advanced Machine Learning models could maybe result in more stable output, and finding a solution for controller tuning for closed-loop ML-models.

REFERENCES

[1] International Water Association. *Circular Economy: Tapping the Power of Wastewater*. https://iwa-network.org/learn/circular-economy-tapping-the-power-of-wastewater/ [Accessed: (15.05.2024)]. 2024.

[2] European Commission. *Urban wastewater*. https://environment.ec.europa.eu/topics/water/urban-wastewater_en [Accessed: (15.05.2024)]. 2024.

[3] Veas. *Fra bord til jord*. https://veas.nu/verdiene-i-avlop-og-avfall/gjodsel [Accessed: (15.05.2024)]. 2024.

[4] Jay N. Meegoda et al. "A Review of the Processes, Parameters, and Optimization of Anaerobic Digestion". In: *International Journal of Environmental Research and Public Health* 15.10 (2018). ISSN: 1660-4601. DOI: 10.3390/ijerph15102224. URL: https://www.mdpi.com/1660-4601/15/10/2224.

[5] D. Deublein and A. Steinhauser. *Biogas from Waste and Renewable Resources: An Introduction*. John Wiley Sons, 2008.

[6] M. Akuzawa, T. Hori, and S. Haruta. "Distinctive Responses of Metabolically Active Microbiota to Acidification in a Thermophilic Anaerobic Digester". In: *Environmental Microbiology* 61 (2011). DOI: https://doi.org/10.1007/s00248-010-9788-1. URL: https://link.springer.com/article/10.1007/s00248-010-9788-1.

[7] Ingrid H Franke-Whittle et al. "Investigation into the effect of high concentrations of volatile fatty acids in anaerobic digestion on methanogenic communities". In: *Waste management* 34.11 (2014), pp. 2080–2089.

[8] Chunlan Mao et al. "Review on research achievements of biogas from anaerobic digestion". In: *Renewable and sustainable energy reviews* 45 (2015), pp. 540–555.

[9] Veronica Moset et al. "Mesophilic versus thermophilic anaerobic digestion of cattle manure: methane productivity and microbial ecology". In: *Microbial biotechnology* 8.5 (2015), pp. 787–800.

[10] Pratap Rai et al. "Laboratory-scale Biogas Production from Food Waste in Batch Reactor under Mesophilic Condition". In: (Apr. 2016). DOI: 10.13140/RG.2.1.3780.5680.

[11] Kwanho Jeong et al. "Prediction of biogas production in anaerobic co-digestion of organic wastes using deep learning models". In: *Water Research* 205 (2021), p. 117697. ISSN: 0043-1354. DOI: https://doi.org/10.1016/j.watres.2021.117697. URL: https://www.sciencedirect.com/science/article/pii/S0043135421008915.

[12] Mohsen Asadi, Huiqing Guo, and Kerry McPhedran. "Biogas production estimation using data-driven approaches for cold region municipal wastewater anaerobic digestion". In: *Journal of Environmental Management* 253 (2020), p. 109708. ISSN: 0301-4797. DOI: https://doi.org/10.1016/j.jenvman.2019.109708. URL: https://www.sciencedirect.com/science/article/pii/S0301479719314264.

[13] Yan Wang, Tyler Huntington, and Corinne D Scown. "Tree-based automated machine learning to predict biogas production for anaerobic co-digestion of organic waste". In: *ACS Sustainable Chemistry & Engineering* 9.38 (2021), pp. 12990–13000.

[14] Djavan De Clercq et al. "Interpretable machine learning for predicting biomethane production in industrial-scale anaerobic co-digestion". In: *Science of The Total Environment* 712 (2020), p. 134574. ISSN: 0048-9697. DOI: https://doi.org/10.1016/j.scitotenv.2019.134574. URL: https://www.sciencedirect.com/science/article/pii/S0048969719345656.

[15] T. Komulainen et al. "*Modeling and control of WRRF biogas production*". In: *64th International Conference of Scandinavian Simulation Society* (Oct. 2023). DOI: https://doi.org/10.3384/ecp200027.

[16] M. Henze et al. *Biological Wastewater Treatment Principles, Modelling and Design*. IWA Publishing, 2008.

[17] S Skogestad. "Simple analytic rules for model reduction and PID controller tuning". In: *Journal of Process Control* 13 (2003), pp. 291–309.

APPENDICES

A - EXTRA GRAPHS

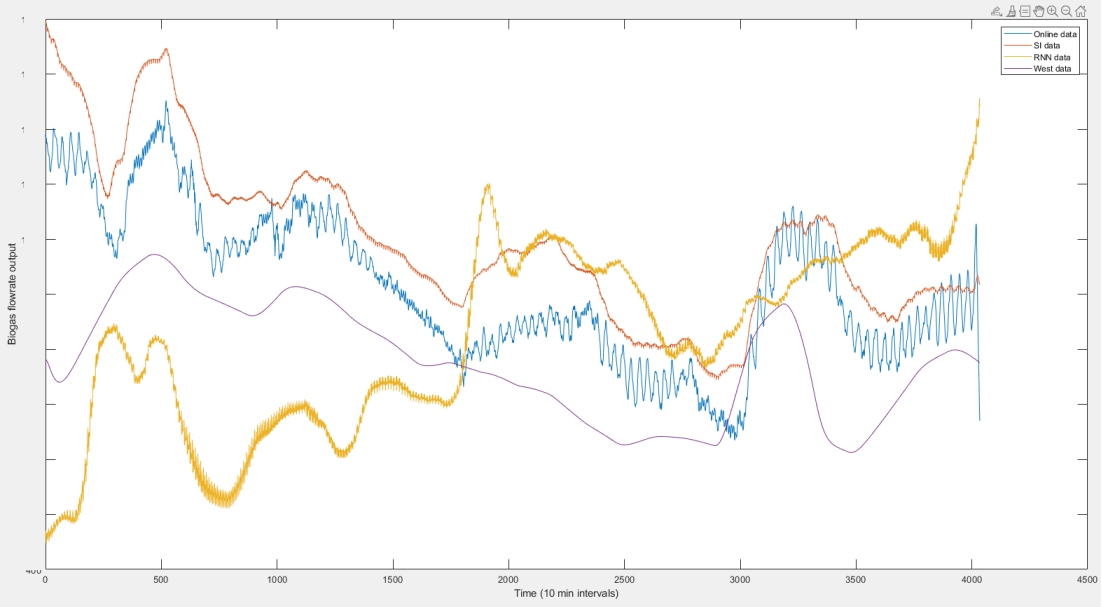

Figure A.1: Comparison of all models and actual AD output within lab dataset range between 30-June-2022 and 28-July-2022

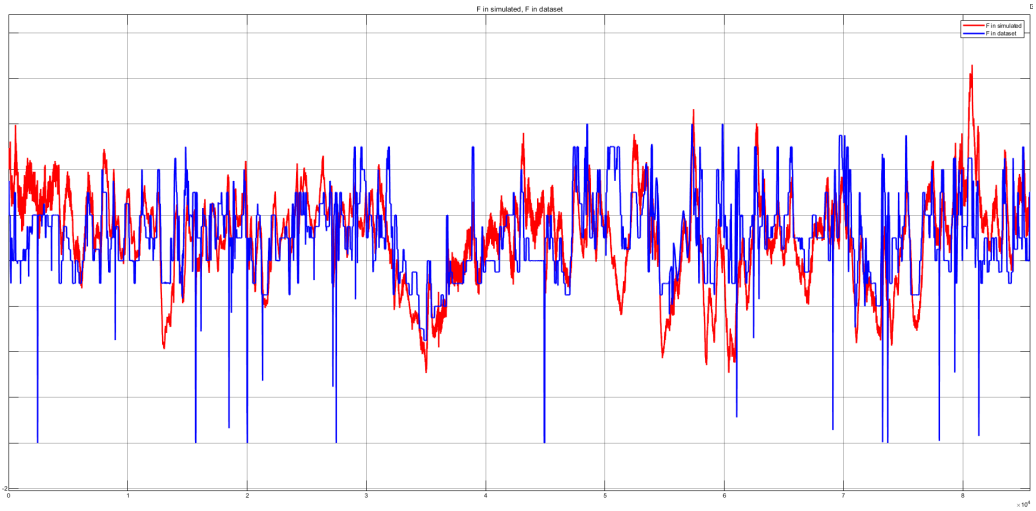

Figure A.2: Flowrate of sludge into the anaerobic digester. Blue graph represents the actual data values, while red are the flowrate controlled by the PI-controller

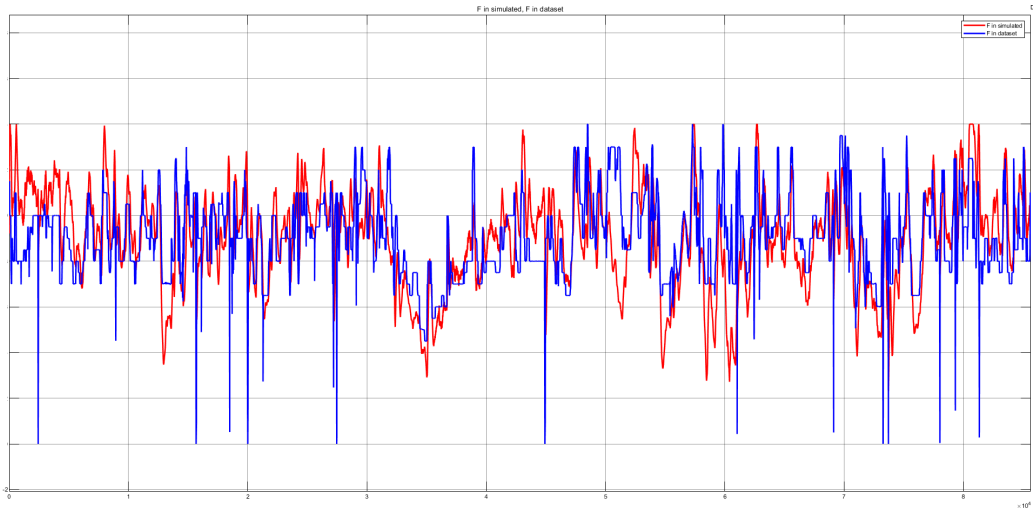

Figure A.3: Flowrate of sludge into the anaerobic digester. Blue graph represents the actual data values, while red are the flowrate controlled by the MPC

B - TABLES

Table B.1: Values extracted from the lab samples

Description	Abbreviation	Unit
Total Carbon Oxygen Demand	tCOD	mg/l
Soluble Carbon Oxygen Demand	sCOD	mg/l
Ammonium	NH4-N	mg/l
Acidity / Alkalinity	pH	1-14
Total Solids mass	TS	%
Organic Solids mass of Total Solids	VS	%
Organic acids		meq/l
Total alkalinity		meq/l
Protein		g/l
Raw fats		g/100g
Carbohydrates		g/l
Acetic acid		mg/l
Propionic acid		mg/l
Butanoic acid		mg/l
Isovaleric acid		mg/l
Valeric acid		mg/l
Hexanoic acid		mg/l
Heptanoic acid		mg/l

Table B.2: Values needed for anaerobic digester model in West

Description	Abbreviation	Unit
Volumetric flowrate	Q	m^3/d
Aminoacids	S_aa	g/m^3
Acetic acid	S_ac	g/m^3
Propionic acid	S_bu	g/m^3
Hexanoic + Heptanoic acid	S_fa	g/m^3
Propionic acid	S_pro	g/m^3
Sugars	S_su	g/m^3
Valeric + Isovaleric acid	S_va	g/m^3
Sulfate	S_so4	g/m^3
Sulfide	S_IS	g/m^3
Methane	S_ch4	g/m^3
Hydrogen	S_h2	g/m^3
Dissolved liquid methane	S_ch4_liq	g/m^3
Dissolved liquid carbon dioxide	S_co2_liq	g/m^3
Dissolved liquid hydrogen	S_h2_liq	g/m^3
Anion	S_an	g/m^3
Cation	S_cat	g/m^3
Inorganic carbon	S_IC	g/m^3
Inorganic nitrogen	S_INN	g/m^3
Soluble inert COD	S_Inert	g/m^3
Phosphate	S_po	g/m^3
Aminoacids degraders	X_aa	g/m^3
Acetic acids degraders	X_ac	g/m^3
Composite organics	X_c	g/m^3
Butanic acid degraders	X_c4	g/m^3
Carbohydrates	X_ch	g/m^3
Long chain volatile fatty acid degraders	X_fa	g/m^3
Hydrogen degraders	X_h2	g/m^3
Lipids	X_li	g/m^3
Protein degraders	X_pr	g/m^3
Propionic acid degraders	X_pro	g/m^3
Sugar degraders	X_su	g/m^3
Bacteria degraders	X_asrb	g/m^3
Bacteria degraders	X_hsrb	g/m^3

Continued on next page

Table B.2: Values needed for anaerobic digester model in West (Continued)

Bacteria degraders	X_c4srb	g/m^3
Bacteria degraders	X_psrb	g/m^3
Non-degradable particulate organic fraction	X_Inert	g/m^3
Metal salt	X_meoh	g/m^3
Metal salt - phosphorous	X_mep	g/m^3
Metal salt - Sulfur	X_mes	g/m^3
Inorganic solids (FSS)	X_u_ig	g/m^3
Total gas flowrate	G_Q	m^3/d
Methane gas flowrate	G_CH4	m^3/d
Carbon dioxide gas flowrate	G_CO2	m^3/d
Hydrogen gas flowrate	G_H2	m^3/d
Hydrogen suflide gas flowrate	G_H2S	m^3/d

B - CODE

Python code to train an RNN model on a loaded dataset

import numpy as np import pandas as pd import tensorflow as tf from tqdm import tqdm from keras.models import Sequential from keras.layers import LSTM, Dense from keras.callbacks import ModelCheckpoint, ReduceLROnPlateau from sklearn.model$_s electionimporttrain_t est_s plit$

path = 'C:/Users/truls/OneDrive - OsloMet/Semester 8/Data/Datasets/Report sets/subset$_r nn_m inmaxed_f iltered.csv'df = pd.read_csv(path)df = df.drop(["Date","Hsludg 1)data_l ength = df.shape[0]input_d im = df.shape[1] - 2$

time$_s tep = 10minuteshour = int(60/time_s tep)day = 24 * hourweek = 7 * day$

batch$_s ize = [5]ahead_p redictions = [2 * hour]retention_t ime = 20days$

def train$_m odel(batch_s izes, ahead_p redictions)$:

for ahead$_p redictioninahead_p redictions$:

for batch$_s izeinbatch_s izes$:

output$_d im = ahead_p rediction$

forecast$_w indow = int(retention_t ime * day)train_w indow = 42826test_w indow = data_l ength - train_w indow - forecast_w indow - ahead_p redictionprint(f"Testdays : int(test_w indow/day)")$

if ahead$_p rediction >= day : model_n ame = f"str(int(ahead_p rediction/day))d"elifahead_p red hour : model_n ame = f"str(int(ahead_p rediction * 10))m"else : model_n ame = f"str(int(ahead_p rediction/hour))h"$

model$_n ame+ = f"_b str(batch_s ize)"$

""" print("Normalizing...")

for col in tqdm(range(df.shape[1])): minimum = min(df.iloc[:,col]) maximum = max(df.iloc[:,col]) for row in range(df.shape[0]): df.iloc[row,col] = (df.iloc[row,col] - minimum)/(maximum-minimum) """

X$_t rain = []y_t rain = []X_t est = []y_t est = []$

print("Constructing...") print(train$_w indow, test_w indow)$

for i in tqdm(range(0, train$_w indow + test_w indow, ahead_p rediction)) : ifi < train_w indow : row = [np.transpose([a])foraindf.to_n umpy()[i : i+forecast_w indow, 1 : input_d im+1]]X_t rain.append(row)row = [np.transpose([a])foraindf.to_n umpy()[i+ forecast_w indow : i+forecast_w indow+ahead_p rediction, df.shape[1]-1]]y_t rain.append(row)else row = [np.transpose([a])foraindf.to_n umpy()[i : i+forecast_w indow, 1 : input_d im+ 1]]X_t est.append(row)row = [np.transpose([a])foraindf.to_n umpy()[i+forecast_w indow : i + forecast_w indow + ahead_p rediction, df.shape[1] - 1]]y_t est.append(row)$

X$_t rain = np.asarray(X_t rain).astype('float32')X_t est = np.asarray(X_t est).astype('float32') np.asarray(y_t rain).astype('float32')y_t est = np.asarray(y_t est).astype('float32')$

print(X$_t rain.shape, X_t est.shape)print(y_t rain.shape, y_t est.shape)$

model = Sequential([LSTM(20, return$_sequences = True, input_shape = (forecast_window, input_dim$
$linear')])

opt = tf.keras.optimizers.Adam(learning$_rate = 0.001)

model$_checkpoint_callback = ModelCheckpoint(filepath = f'C : /Users/truls/OneDrive-$
$OsloMet/Semester8/Models/RNN/Reportmodels/model_name.keras', monitor ='$
$mse', mode =' min', save_best_only = True)$

learning$_rate_reduction = ReduceLROnPlateau(monitor =' mse', patience =$
$2, verbose = 1, factor = 0.5, min_lr = 0.0000001)$

model.compile(optimizer = opt, loss='mean$_squared_error', metrics =' mse')$

model.fit(X$_train, y_train, epochs = 50, batch_size = batch_size, callbacks = [model_checkpoint_callback,$

model.reset$_states()y_train_pred = model.predict(X_train)y_test_pred = model.predict(X_test)$

print(y$_train.shape, y_test_pred.shape)$

results = pd.DataFrame(data='Predicted' : y$_pred,' Actual' : y)$

train$_dict = test_dict =$

for window in range(len(X$_train)) : train_dict[f'Predictedwindowwindow'] =$
$y_train_pred[window, :].flatten()train_dict[f'Actualwindowwindow'] = y_train[window, :$
$, 0].flatten()forwindowinrange(len(X_test)) : test_dict[f'Predictedwindowwindow'] =$
$y_test_pred[window, :].flatten()test_dict[f'Actualwindowwindow'] = y_test[window, :$
$, 0].flatten()$

results$_train = pd.DataFrame(train_dict)results_test = pd.DataFrame(test_dict)$

results = pd.DataFrame(data='Predicted' : y$_pred[144, :].flatten(),' Actual' :$
$y[144, :, 0].flatten())$

results$_train.to_csv(f'C : /Users/truls/OneDrive-OsloMet/Semester8/Data/RNNoutput/Repor$
$False)results_test.to_csv(f'C : /Users/truls/OneDrive-OsloMet/Semester8/Data/RNNoutput/Rep$
$False)$

train$_model(batch_size, ahead_predictions)$

Load a trained RNN model and a dataset to run prediction and output a prediction saved to csv

import tensorflow as tf import keras import numpy as np import pandas as pd from tqdm import tqdm

path = 'C:/Users/truls/OneDrive - OsloMet/Semester 8/Data/Datasets/Report sets/subset$_rnn_minmaxed_filtered.csv'df = pd.read_csv(path)df = df.drop(["Date","Hsludge"], axi$
$1)data_length = df.shape[0]input_dim = df.shape[1]$

time$_step = 10minuteshour = int(60/time_step)day = 24 * hourweek = 7 * day$

batch$_size = 5ahead_prediction = 2 * hourretention_time = 20days$

def predict$_id_mv() :$

forecast$_window = int(retention_time * day)train_window = 42826test_window =$
$data_length-train_window-forecast_window-ahead_predictionprint(f"Testdays :$
$int(test_window/day)")$

if ahead$_prediction < day : model_name = f"str(int(ahead_prediction/hour))h"else :$
$model_name = f"str(int(ahead_prediction/day))d"model_name+ = f"_bstr(batch_size)"$

output$_dict =$

print("Constructing...")

model = keras.saving.load$_model(f'C : /Users/truls/OneDrive-OsloMet/Semester8/Models/RN$

model.reset$_states()$

for i in range(0, train$_window + test_window, ahead_prediction) :$

print(i)

y = [] y$_pred = []$

row = [np.transpose([a]) for a in df.to$_n umpy()[i : i + forecast_w indow, :]]x = np.array(row)x = tf.reshape(x, [−1, 2880, 5])y_w indow = model.predict(x)

for j in range(ahead$_p rediction) : y.append(df.loc[i + forecast$_w$indow + j + 1, "Fout"])y$_p$red.append(y$_w$indow[0, j])df.loc[i+forecast$_w$indow+j+1, "Fout"] = y$_w$indow[0, j]

output$_d ict[f'Predictedwindowint(i/ahead_p rediction)'] = y_p red$

results = pd.DataFrame(output$_d ict)

results.to$_c sv(f'C : /Users/truls/OneDrive−OsloMet/Semester8/Data/RNNoutput/Rep False)$

predict$_p id_m v()$

Load a RNN-model and a dataset, define setpoint and PI-controller tune and run prediction in closed loop state. Output saved to csv

import tensorflow as tf import keras import numpy as np import pandas as pd from tqdm import tqdm

path = 'C:/Users/truls/OneDrive - OsloMet/Semester 8/Data/Datasets/Report sets/subset$_r nn_m inmaxed_f iltered.csv'df = pd.read_c sv(path)df = df.drop(["Date","Hsludg 1)data_l ength = df.shape[0]input_d im = df.shape[1]$

time$_s tep = 10minuteshour = int(60/time_s tep)day = 24 * hourweek = 7 * day$

batch$_s ize = 5ahead_p rediction = 2 * hourretention_t ime = 20days$

class PID:

def $_{init}{}_{(self,pid_v alues):self.p=pid_v alues[0]self.i=pid_v alues[1]self.integrator=0}$

def u(self, e): self.integrator += self.i*e u = self.p*e + self.integrator return u

pid$_c ontroller = PID([0.7848, 0.0132])$

def predict$_p id_m v() :$

set$_p oint = df.loc[:, "Fout"]$

forecast$_w indow = int(retention_t ime * day)train_w indow = 42826test_w indow = data_l ength−train_w indow−forecast_w indow−ahead_p rediction print(f"Testdays : int(test_w indow/day)")$

if ahead$_p rediction < day : model_n ame = f"str(int(ahead_p rediction/hour))h"else : model_n ame = f"str(int(ahead_p rediction/day))d"model_n ame+ = f"_b str(batch_s ize)"$

output$_d ict =$

print("Constructing...")

model = keras.saving.load$_m odel(f'C : /Users/truls/OneDrive−OsloMet/Semester8/Mode$

model.reset$_s tates()$

for i in range(0, train$_w indow + test_w indow, ahead_p rediction) :$

print(i)

y = [] y$_p red = []$

row = [np.transpose([a]) for a in df.to$_n umpy()[i : i + forecast_w indow, :]]x = np.array(row)x = tf.reshape(x, [−1, 2880, 5])y_w indow = model.predict(x)$

for j in range(ahead$_p rediction) : y.append(df.loc[i + forecast_w indow + j + 1, "Fout"])y_p red.append(y_w indow[0, j])df.loc[i+forecast_w indow+j+1, "Fout"] = y_w indow[0, j]$

error = set$_p oint[i+forecast_w indow+1 : i+forecast_w indow+ahead_p rediction+ 1].sum() − np.sum(y_w indow)$

fin = pid$_c ontroller.u(error)$

df.loc[i+forecast$_w indow + 1 : i + forecast_w indow + ahead_p rediction, "Fin"] = [fin] * ahead_p rediction$

output$_d ict[f'Predictedwindowint(i/ahead_p rediction)'] = y_p red$

results = pd.DataFrame(output$_d ict$)

results.to$_c sv(f'C : /Users/truls/OneDrive-OsloMet/Semester8/Data/RNNoutput/Reportoutp$
$False)$

predict$_p id_m v()$

Combine all signal exported from Veas to one single csv-file

import os import pandas as pd from tqdm import tqdm

path = 'C:/Users/truls/OneDrive - OsloMet/Semester 8/Data/Datasets/Five
years' start$_y ear =' 2019' output_f ile = "D : /test.csv"$

columns =[]

combined$_d f = pd.DataFrame(columns = columns)$

shortest = float("nan")

for root, dirs, files in os.walk(path): for file in tqdm(files): if file.endswith(".csv"):
df = pd.read$_c sv(os.path.join(root, file), sep =';', dtype = "string")start_i ndex =$
$0year = df.iloc[start_i ndex, 0].split("")[0].split(" - ")[0]while year! = start_y ear :$
$start_i ndex+ = 1year = df.iloc[start_i ndex, 0].split("")[0].split(" - ")[0]$

if isinstance(shortest, float) or (df.shape[0] - start$_i ndex) < shortest : shortest =$
$int(df.shape[0] - start_i ndex)$

print(columns, shortest)

for root, dirs, files in os.walk(path): for file in tqdm(files): if file.endswith(".csv"):
df = pd.read$_c sv(os.path.join(root, file), sep =';', dtype = "string")start_i ndex =$
$0year = df.iloc[start_i ndex, 0].split("")[0].split(" - ")[0]while year! = start_y ear :$
$start_i ndex+ = 1year = df.iloc[start_i ndex, 0].split("")[0].split(" - ")[0]$

df.rename(columns= df.columns[1]: df.columns[0].split(".")[0], df.columns[0]:
"Date" , inplace = True)

df = df.iloc[start$_i ndex : start_i ndex+shortest, :]df.reset_i ndex(drop = True, inplace =$
$True)$

if combined$_d f.shape[0] == 0 : combined_d f = pd.concat([combined_d f, df], axis =$
$1)else : combined_d f = pd.concat([combined_d f, df.iloc[:, 1]], axis = 1)$

combined$_d f = combined_d f.iloc[132263 :, :]$

combined$_d f.to_c sv(output_f ile, index = False)$

Methods (functions) used for composing online dataset

def get$_r ow_i ndex(df, date_s tart, date_e nd) : for row in range(df.shape[0]) : if df.loc[row, "Date"] ==$
$date_s tart : row_s tart = row eli f df.loc[row, "Date"] == date_e nd : row_e nd = row$

return row$_s tart, row_e nd$

def get$_t in(df, temp_d ict)$:

temp$_l ist = []$

empties = 0 empty$_g ap = 0empty_g ap_m ax = 0$

for row in range(df.shape[0]): found = False multiple = False param$_p rev =$
$[] for key in temp_d ict.keys() : if df.loc[row, temp_d ict[key]] > 0 : param_p rev.append(key) if found ==$
$False : found = True else : multiple = True if found : if multiple : temp_v al =$
$0total = 0 for param in param_p rev : total+ = df.loc[row, temp_d ict[param]]temp_v al+ =$
$df.loc[row, temp_d ict[param]] * df.loc[row, param]temp_v al/ = totaltemp_l ist.append(temp_v al) else :$
$temp_l ist.append(df.loc[row, param_p rev[0]])$

else: temp$_l ist.append(temp_l ist[-1])empties+ = 1empty_g ap+ = 1 if empty_g ap >$
$empty_g ap_m ax : empty_g ap_m ax = empty_g ap$

return temp$_l ist$

def get$_t ank_t emp_h eight(df, tags)$:

temp$_l ist = []height_l ist = []$

for row in range(df.shape[0]):
temp$_row = 0 height_total = 0$
for tank in tags.keys(): temps = [df.loc[row,tag] for tag in tags[tank]["temp"]]
$temp_avg = sum(temps)/len(temps) tank_height = df.loc[row, tags[tank]["height"]] temp_row += temp_avg * tank_height height_total += tank_height$
$temp_row /= height_total$
$temp_list.append(temp_row) height_list.append(height_total)$
return $temp_list, height_list$
def $get_foam(df, foam_tags)$:
$foam_list = []$
for row in range(df.shape[0]): vals = [df.loc[row,tag] for tag in $foam_tags] foam_height = sum(vals) foam_list.append(foam_height)$
return $foam_list$
def $get_biogas(df, biogas_tags)$:
$biogas_list = []$
for row in range(df.shape[0]): vals = [df.loc[row,tag] for tag in $biogas_tags] biogas_flow = sum(vals) biogas_list.append(biogas_flow)$
return $biogas_list$
Composing of the online dataset
import pandas as pd from $combine_online_methodsimport*$
df = $pd.read_csv('C : /Users/s356159/OneDrive-OsloMet/Semester8/Data/Datasets/Fi$
$pd.read_csv('C : /Users/truls/OneDrive-OsloMet/Semester8/Data/Datasets/Fiveyears/co$
$date_year_start = "2021" date_month_start = "11" date_day_start = "19" date_hour_start = "13" date_minute_start = "00" date_second_start = "00" date_start = f"date_year_start-date_month_start - date_day_startdate_hour_start : date_minute_start : date_second_start"$
$date_year_end = "2023" date_month_end = "07" date_day_end = "07" date_hour_end = "09" date_minute_end = "00" date_second_end = "00" date_end = f"date_year_end - date_month_end - date_day_enddate_hour_end : date_minute_end : date_second_end"$
$flow_tag = "FOR_FT02"$
$temp_in_tags = 'RT - RT - TI94' :' RT - RT - KSV132','RT - RT - PU22_TT' :' RT - R$
$tank_tags_temp_height = "tank1" : "temp" : ["RT1_TT06","RT1_TT03"], "height" : "RT1_LT05"$
$foam_tags = ["RT1_LB01","RT2_LB01","RT3_LB01","RT4_LB01"]$
$biogas_tags = ["RT - GSB - FI41","RT - GSB - FI42","RT - GSB - FI43","RT - GSB - FI44"]$
def $create_new_subset()$:
$row_start, row_end = get_row_index(df, date_start, date_end)$
$temp_list, height_list = get_tank_temp_height(df, tank_tags_temp_height)$
$new_df_dict = "Date" : df.loc[:,"Date"], "T_in" : get_tin(df, temp_in_tags), "F_in" : df.loc[:,"FOR_$
length = float("nan") uneven = False for i, range in enumerate($new_df_dict.values()$) :
$if len(range) != length and i > 0 : uneven = True else : length = len(range)$
if uneven == True: print("Uneven length of data points") return pd.DataFrame(columns=[])
else: $new_df = pd.DataFrame(new_df_dict) new_df = new_df.loc[row_start : row_end, :].reset_index(drop = True) return new_df$
$new_df = create_new_subset() print(new_df) print(new_df.loc[0,"Date"]) print(new_df.loc[new_df.$
$1,"Date"]) print(new_df.shape)$
$new_df.to_csv('C : /Users/truls/OneDrive-OsloMet/Semester8/Data/Datasets/Reportse$
$False)$
Extract lab measurements and convert to proper units

import pandas as pd

df = pd.read$_e$xcel('C : /Users/s356159/OneDrive−OsloMet/Semester8/Data/Datasets/Merged
andOnlinedataplusconversionsVeasjuni − juli22.xlsx')

start$_r$ow = 4end$_r$ow = 33

def fill$_c$olumn(df, col, start, end) : prev$_v$al$_r$ow = float("nan")prev$_v$al = float("nan")gap =
False

for row in range(start,end):

if df.iloc[row,col] == df.iloc[row,col]:

if gap == True: end$_v$al = df.iloc[row, col]foriinrange(row − prev$_v$al$_r$ow) :
try : df.iloc[prev$_v$al$_r$ow + i, col] = prev$_v$al + (i ∗ ((end$_v$al − prev$_v$al)/(row −
prev$_v$al$_r$ow)))except : print(prev$_v$al$_r$ow + i, col, end$_v$al, prev$_v$al)gap == False

prev$_v$al$_r$ow = rowprev$_v$al = df.iloc[row, col]

else: gap = True

return df.iloc[start:end, col]

columns = 'tCOD' : 'Rot' : 3, 'FOR' : 6 , 'sCOD' : 'Rot' : 9, 'FOR' : 12 ,
'Protein' : 'Rot' : 53, 'FOR' : 54 , 'Eddiksyre' : 68, 'Butansyre' : 70, 'Heksansyre'
: 73, 'Heptansyre' : 74, 'Propansyre' : 69, 'Isovaleriansyre' : 71, 'Valeriansyre' :
72, 'Karbohydrater' : 'Rot' : 57, 'FOR' : 58 , 'pH' : 'Rot' : 19, 'FOR' : 20 ,
'Rafett' : 'Rot' : 55, 'FOR' : 56 , 'TS' : 'Rot' : 21, 'FOR' : 22 , 'VS 'Rot' : 23,
'FOR' : 24 , 'NH4-N' : 'Rot' : 13, 'FOR' : 14

new$_d$ict =

for key in columns.keys(): if isinstance(columns[key], dict): cols = columns[key].keys()
for col in cols: column = key + " " + col new$_d$ict[column] = columns[key][col]else :
print(key)new$_d$ict[key] = columns[key]

for key in new$_d$ict.keys() : new$_d$ict[key] = fill$_c$olumn(df, new$_d$ict[key], start$_r$ow, end$_r$ow)

new$_d$f = pd.DataFrame(new$_d$ict)

new$_d$f.loc[:,' ProteinRot']∗ = 1000new$_d$f.loc[:,' ProteinFOR']∗ = 1000new$_d$f.loc[:
,' RafettRot']∗ = 10000new$_d$f.loc[:,' RafettFOR']∗ = 10000new$_d$f.loc[:,' KarbohydraterRot']∗ =
1000new$_d$f.loc[:,' KarbohydraterFOR']∗ = 1000new$_d$f.loc[:,' TSRot']∗ = 10000new$_d$f.loc[:
,' TSFOR']∗ = 10000

print(new$_d$f.loc[:,' ProteinFOR'])

new$_d$f.to$_e$xcel('C : /Users/s356159/OneDrive−OsloMet/Semester8/Data/Datasets/test.xlsx', i
False)

Import extracted lab measurements and create the lab dataset for West
simulation

import pandas as pd

df = pd.read$_e$xcel('C : /Users/s356159/OneDrive−OsloMet/Semester8/Data/Datasets/test.xls

s$_c$od = df.loc[:,' sCODFOR']t$_c$od = df.loc[:,' tCODFOR']x$_c$od = t$_c$od − s$_c$od

x$_p$r = df.loc[:,' ProteinFOR']x$_c$h = df.loc[:,' KarbohydraterFOR']

s$_a$a = (s$_c$od∗x$_p$r)/x$_c$ods$_a$c = df.loc[:,' Eddiksyre']s$_b$u = df.loc[:,' Butansyre']s$_f$a =
df.loc[:,' Heksansyre'] + df.loc[:,' Heptansyre']s$_p$ro = df.loc[:,' Propansyre']s$_s$u =
(s$_c$od ∗ x$_c$h)/x$_c$ods$_v$a = df.loc[:,' Isovaleriansyre'] + df.loc[:,' Valeriansyre']s$_a$n =
14−df.loc[:,' pHFOR']s$_c$at = df.loc[:,' pHFOR']s$_i$nn = df.loc[:,' NH4−NFOR']s$_i$nert =
s$_c$od − s$_a$c − s$_b$u − s$_f$a − s$_p$ro − s$_v$a − s$_s$u − s$_a$ax$_l$i = df.loc[:,' RafettFOR']x$_i$nert =
x$_c$od − x$_c$h − x$_l$i − x$_p$r

x$_{ui}$g = df.loc[:,' TSFOR'] ∗ df.loc[:,' VS

new$_d$ict = 'x$_c$h' : x$_c$h,' x$_p$r' : x$_p$r,' s$_a$a' : s$_a$a,' s$_a$c' : s$_a$c,' s$_b$u' : s$_b$u,' s$_f$a' : s$_f$a,' s$_p$ro' : s$_p$ro,' s$_s$u' : s$_s$u,' s

new$_d$f = pd.DataFrame(new$_d$ict)

new$_df.to_excel('C:/Users/s356159/OneDrive-OsloMet/Semester8/Data/Datasets/new_$
$False)$
print(new$_df.columns)$
corr$_data = readtable('C:/Users/truls/OneDrive-OsloMet/Semester8/Data/Datasets/$
date = corr$_data$:,$'Date'$;
corr$_data = removevars(data,'Date')$;
time$_constant = 1350; sample_length = 10; time_constant_samples = time_constant/sample_leng$
moving$_mean_window = round(time_constant_samples/10)$;
for col = 1:width(corr$_data$)
corr$_data$:$, col = movmean(corr_data$:$, col, moving_mean_window)$;
col$_min = min(corr_data$:$, col); col_max = max(corr_data$:$, col)$;
for row = 1:height(corr$_data$)
corr$_datarow, col = (corr_datarow, col - col_min)/(col_max - col_min)$;
end end
corr$_data.("Date") = date$;
writetable(corr$_data,' correlation_data_minmaxed_filtered.csv')$;
raw$_data = readtable('C:/Users/truls/OneDrive-OsloMet/Semester8/Data/Datasets/$
raw$_data = removevars(raw_data,'Date'); raw_data = removevars(raw_data,'Hsludge'); raw_d$
$removevars(raw_data,'Ttank')$;
time$_constant = 1350; sample_length = 10; time_constant_samples = time_constant/sample_leng$
moving$_mean_window = round(time_constant_samples/10)$;
for col = 1:width(raw$_data$)
raw$_data$:$, col = movmean(raw_data$:$, col, moving_mean_window); end$
fin$_max = max(raw_data$:$,"Fin"); fin_min = min(raw_data$:$,"Fin")$;
tsin$_max = max(raw_data$:$,"TSin"); tsin_min = min(raw_data$:$,"TSin")$;
fout$_max = max(raw_data$:$,"Fout"); fout_min = min(raw_data$:$,"Fout")$;
for row = 1:height(raw$_data$)$raw_datarow,"Fin" = (raw_datarow,"Fin"$-$fin_min)/(fin_max-$
$fin_min); raw_datarow,"TSin" = (raw_datarow,"TSin"$-$tsin_min)/(tsin_max$-$tsin_min); raw_datar$
$(raw_datarow,"Fout" - fout_min)/(fout_max - fout_min); end$
fin$_avg = mean(raw_data$:$,"Fin"); tsin_avg = mean(raw_data$:$,"TSin"); fout_avg =$
$mean(raw_data$:$,"Fout")$;
data = readtable('C:/Users/truls/OneDrive - OsloMet/Semester 8/Data/Dataset-
s/Report sets/normalized$_data.csv','VariableNamingRule','preserve')$;
model = load('si$_model.mat').tf$;
t = 0:1:height(data)-1; t$_end = length(t) - 1$;
tin = [t' data:,"T in"]; fin = [t' data:,"F in"]; hfoam = [t' data:,"H foam"];
tsin = [t' data:,"TS in"]; fout = [t' data:,"F out"];
k = model(1,1).Numerator/model(1,1).Denominator(2); tau$_1 = model(1,1).Denominator(1)$
tau$_c = 100; Kc = tau_1/(k * tau_c); tau_I = min(tau_1, 4 * tau_c)$;
P = Kc; I = Kc/tau$_I; D = 0$;
data$_length = height(data); data_split = int32(data_length/2)$;
u1 = [data1:data$_split,"Fin", data1 : data_split,"Tin", data1 : data_split,"TSin", data1 : data_$
$[data1 : data_split,"Fout"]; est_data = iddata(y1, u1, 1)$;
u2 = [datadata$_split+1 : data_length,"Fin", datadata_split + 1 : data_length,"Tin", datadata_sp$
$[datadata_split + 1 : data_length,"Fout"]; val_data = iddata(y2, u2, 1)$;
N1 = max(length(y1)); t1=0:1:N1-1;
y1mean=ones(N1,1)*mean(y1); e1m=sum((y1-y1mean).^{2});

N2=max(length(y2)); t2=N1:1:N1+N2-1; y2mean=ones(N2,1)*mean(y2); e2m=sum((y2-y2mean).^{2});

R2tf$_o ld = -100000000000; remaining = power(2, 4);$

for i = 0:1 for j = 0:1 for k = 0:1 for l = 0:1 new$_o rder = [ijkl]; new_t f = tfest(est_d ata, new_o rder,' InputName', ["Fin","Tin","TSin","Hfoam"],' OutputName',"Fout");$

ytfval = lsim(new$_t f, u2, t2'); etfval = sum((y2 - ytfval).^2); R2tfval = 1 - (etfval/e2m);$

if R2tfval > R2tf$_o lddisp("newbest")order = new_o rdertf = new_t f; ytf = ytfval; etf = etfval; R2tf_o ld = R2tfvalend$

remaining = remaining - 1

end end end end

plant = setmpcsignals(model, 'MD', [2 3 4]);

Ts = 1; P = 120; M = P/2;

mv$_m ax = max(fin(:, 2)); mv_m in = min(fin(:, 2));$

MV1 = struct('Min',mv$_m in,' Max', mv_m ax);$

MV = [MV1];

ov$_m ax = max(fout(:, 2)); ov_m in = min(fout(:, 2));$

OV = struct('Min',ov$_m in,' Max', ov_m ax);$

Q=[1]; Ru=[0]; Rd=[20]; W=struct('ManipulatedVariables',Ru,'ManipulatedVariablesRate',Rd,'Ou

MPCcaseB = mpc(plant,Ts,P,M,W,MV);

xmpcaseB = mpcstate(MPCcaseB);