ACIT5900

MASTER THESIS

in

Applied Computer and Information Technology (ACIT)
May 2023

Robotics and Control

Model predictive controller of phosphorus removal in a wastewater treatment process

Einar Nermo

Department of Mechanical, Electronic and Chemical Engineering (MEK)
Faculty of Technology, Art, and Design

Preface

The research conducted for this project was motivated by my increasing concern for the environment and the challenges that we face. I would like to express my gratitude to everyone involved in this project including various professors and master students for the tremendous help and support you have given me throughout this process. I would especially thank Tiina Komulainen for this opportunity and the endless guidance that she has provided me. I would also like to thank Bilal Mukhtar for the close collaboration that we have had. My family and friends have been of tremendous support, and I wouldn't have been where I am without them. Doing this master's thesis has been a very challenging task to perform. Dealing with sickness when performing this master's thesis made me question my future in this field and whether I was able to deliver a good project to the normed time. However, I pulled through and worked twice as hard to deliver the best possible project. I have gained even more respect for people in this field. Understanding how to preprocess data, getting the correct transfer function by system identification, and then implementing control strategies to control the right nutrient levels have shown to be a very difficult task. However, I do find it interesting, and I'm grateful for this opportunity.

Einar Nermo

Abbreviations

BOD	Biological oxygen demand
CV	Control variable
DO	Dissolved oxygen
DV	Disturbance variable
EBPR	Enhanced biological phosphorus removal
FPE	Final prediction error
Hias	Hamar interkommunale avløpsselskap (Hamar intermunicipal wastewater treatment firm)
IAE	Integral of absolute error
IAMV	Integral of total movement in manipulated variables
IKS	Inter kommunalt selskap (intermunicipal firm)
MBBR	Moving bed biofilm reactor
MV	Manipulated variable
MPC	Model predictive controller
MSE	Mean squared error
PAO	Phosphorus accumulating organisms
PHB	Polyhydroxybutyrate
PID	Proportional-integral-derivative controller
PO4-P	Polyphosphate
sCOD	Soluble chemical oxygen demand
SMPC	Stochastic model predictive controller
TF	Transfer function
VEAS	Vestfjordens Avløpsselskap (Vestfjordens wastewater treatment firm)
VFA	Volatile fatty acids
WRRF	Wastewater treatment and water resource recovery facilities

List of figures

List of tables

Table of contents

Abstract

Historically, poor sanitation has always been a concern as it causes diseases such as cholera, intestinal worm infections, polio, typhoid, and dysentery. Sanitation problems are still a pressing issue for many people around the world. Developing an energy and cost-effective wastewater treatment and water resource recovery facility that could potentially be scaled up for a wider adoption would be a virtues pursuit. The Hias process uses biofilm carriers in anerobic and aerobic basins that absorbs the nutrients that comes into the wastewater treatment facility. This enhanced biological phosphorus removal (EBPR) process is being performed in moving-bed bioreactor (MBBR). It is able to remove 90% of phosphorus or polyphosphate (PO4-P) from the wastewater. Hias IKS Wastewater treatment and water resource recovery facilities (WRRF) is an end user in PACBAL research project lead by Tiina Komulainen, which is the main supervisor for this master's thesis. Control strategies in a wastewater treatment plant is crucial, as it allows for efficient management of the water purification process to safeguard our environment. The focus and goal for this master's thesis is to improve the Hias process energy efficiency by implementing advanced control strategies. Developing the traditional industry standard control strategy, the Proportional-integral-derivative (PID) controller, will be compared against the novel approach of a model predictive controller (MPC) in the Hias process. The MPC designed in this project achieved better results than the PID when it comes to disturbance rejection, set point tracking and energy efficiency. However, the PID utilizes simpler control structures and has lower computational time than the MPC. This is often more desirable for industrial implementation. The development of these control strategies is an important step towards industrial implementation. The control strategies are based on transfer functions derived by system identification of the online data provided for the Hias process. The datasets contained online measurements of soluble chemical oxygen demand (S_{Sins})-, flowrate of wastewater (F_S)- and NO_2/NO_3 (NOX) in the inlet. It also contained flow rate of oxygen (F_O)- and dissolved oxygen (S_O) in the aerobic basins. The datasets also contained polyphosphate that comes out the disc filter (S_{POd}). These variables will be simulated as virtual sensors by utilizing Matlab Simulink. The preprocessing of the datasets achieved adequate correlation between the variables used for the transfer functions. While the dynamic linear models obtained from the system identifications gave sufficient results for control strategy implementation.

The chapters in this thesis will first describe the first part of the project which is the data preprocessing of the online data. Then the system identification to obtain the dynamic linear models represented as transfer functions will be discussed for each chapter. The control strategies developed based on the transfer functions will be the last part of each section.

Keywords—Nutrient removal process, Model predictive controller, Proportional-integral-derivative controller, Hias process, virtual sensors, MBBR, EBPR.

Model predictive controller of phosphorus removal in a wastewater treatment process

Einar Nermo
ACIT master program
Oslo Metropolitan University
Oslo, Norway
s331440@oslomet.no

I. Introduction

Wastewater treatment and water resource recovery facilities (WRRFs) is an expensive- and energy demanding process (Nair et al., 2022). WRRFs are one of the most power consuming parts of the electrical grid in the public domain (European commission, 2021). This is even more emphasized with the rising energy bills economies are facing in the new geopolitical climate. Building a control system that maintains the exact amount of energy needed at any time to maintain the correct nutrient level will use less energy and this will reduce cost. This will reduce taxpayers' money and will make it easier for municipalities to invest in it because of the cost reduction. This will make it easier to build out this much needed infrastructure. The nutrient levels should also not cause any harm to society and the ecological environment. Too many nutrients in an environment can lead to eutrophication problems (Rudi et al., 2019). The Norwegian government requires WRRFs in Norway to remove 90% of phosphorus that comes into the WRRF (*Forskrift Om Begrensning Av Forurensning (Forurensningsforskriften) - Del 4. Avløp - Lovdata*, 2021).

Hias IKS is the company behind the Hias process and is also the company that has provided the data needed for this master's thesis. The Hias process is an enhanced biological phosphorus removal (EBPR) process in a moving bed biofilm reactor MBBR. The Hias process has shown that using biofilms that absorbs the phosphorus and carbon can make significant impact (Rudi et al., 2019). Achieving a 90% removal of phosphorus or polyphosphate (PO4-P) and 66% removal of sCOD, Rudi et al. showcased how advantageous the Hias process can be. The removed PO4-P of the Hias process is being used for the fertilizer struvite, which is very important for the agriculture industry since this is a limited resource (Rudi et al., 2019).

Model predictive controller (MPC) or other advanced control strategies have not yet been implemented to control the phosphorus in the Hias process. Implementing such a system can be a beneficial contribution to reduce the energy need for the air supply. Estimating how much each part of the Hias WRRF that uses the most energy can be challenging. However, Hias IKS has estimated that the air supply does consume the most energy and has estimated it to be around 70%. Controlling the nutrient levels in the most energy efficient way in the Hias process will not only reduce the consumption of energy which will benefit the environment, but also be cost effective and could possibly reduce the need for maintenance. A MPC is a control strategy that predict future behavior of a system and optimize control actions by utilizing mathematical models (Darby & Nikolaou, 2012). The MPC will optimize multiple controlled variables by using information of process influent variables and dynamic models. The MPC will be a replacement of one PID controller and six ratio controllers that controls the flow rate of oxygen in the Hias WRRF.

Developing the traditional and industry standard PID controller to compare it against a novel approach such as MPC will showcase the strengths and weaknesses of the two control strategies. The PID controller is a feedback control algorithm broadly used for industrial applications which can be applied to a variety of processes (Araki, M. 2009). A PID controller has proportional, integral, and derivative elements that serve different objectives. Combinations of these elements can be applied to various purposes such as temperature, speed, and position for instance. The PID controller will control the aeration going through the seven aerobic basins of the Hias process. Using a PID controller at the fifth basin (which is the second aerobic basin) to control the air supply through the valves for each of the seven aerobic basins will be the main objective. The PID controller will control the flow rate of oxygen in basin five (FO5) of the WRRF system. The ratio controllers are a percentage of the PID controller. These ratio controllers will control the flow rate of oxygen for basin 4, 6, 7, 8, 9 and 10 (FO4s, FO6s, FO7s, FO8s, FO9s, FO10s). The first valve has the most impact since the aeration rate is the strongest here and was originally the valve that was going to be controlled by the PID controller. That's why the ratio controller for FO4 has a higher value than for the FO5. The next valves have descending impacts on the PID.

Control strategies can be developed using many different methods. One of which is by collecting online data from the real sensors from the WRRF and use them as virtual sensors in a simulation software. Online data was collected from the Hias WRRF and was used in the simulation program Matlab Simulink. Dynamic linear models in the form of transfer functions can be identified by the software program Matlab System identification. These transfer function will be a mathematical representation of the WRRF and control strategies can simulated using these mathematical models.

A. Research questions

How energy effective would it be to use a MPC to control the phosphorus levels in the Hias process? Are there other control methods such as PID that can reduce the energy consumption better than an MPC? These are the research questions that are going to be explored for this master's thesis.

II. BACKGROUND

The Hias process is an enhanced biological phosphorus removal (EBPR) process in a moving bed biofilm reactor (MBBR). The variables used in this project can be seen in Table 1. This means that microorganisms in the form of biofilm seen in Figure 1 will absorb or "eat" the contaminated nutrients in the environment it is in (Rudi et al., 2019). The environment here being a MBBR process. The phosphorus accumulative organisms (PAO) in the biofilm are the organisms that removes the phosphorus (Xylem YSI Municipal Water, 2021). They can store either polyphosphate (S_{PO}) or poly-B-hydorxybutyrate (PHB), which is illustrated in Figure 2. The PAO process is depended on how much dissolved oxygen is in the wastewater. Volatile fatty acids (VFAs) in the anerobic basins are created in the anaerobic basins by fermentative bacteria using wastewater's carbon content. In the anerobic basins the biofilm will release polyphosphate (S_{PO}) which will increase the S_{PO} levels in the first three anerobic basins of the Hias process. This happens when the PAO releases stored polyphosphate in biomass. The PAO organism will use polyphosphate as energy to get a carbon uptake for VFAs, this will convert the VFAs to poly-B-hydroxybutyrate (PHB) for storage. This can be measured by soluble oxygen demand (sCOD, S_{Sins}) since it is an indirect way of measuring biological oxygen demand (BOD).

When air is applied in the aerobic basins, the biofilm will take up polyphosphate (S_{PO}) and other nutrients which will decrease the levels of the nutrients from basin 4 (B4) until basin 10 (B10), this can be seen in Figure 2. It will have the uptake of S_{PO} since the PAOs is able to generate energy by metabolizing the stored PHB with dissolved oxygen (S_O). Dissolved oxygen and aeration (F_O) are closely correlated since air applied to the system will increase the oxygen levels. That's why the aeration rate (F_O) or the air supply has a crucial role in the process. The conveyer belt moves the biofilm over from basin 10 (B10) to basin 1 (B1), seen in Figure 1. The process will start over, and this will create a continuous loop. The S_{PO} will be separated by the disk filter, where treated wastewater will go out to Gudbrandsdalsågen which leads to the Oslofjord, and S_{PO} will be collected and used for fermentation. The PO4-P analyzer measures S_{PO} that comes after the disc filter. The variable name chosen for this measurement was set to S_{POd}. The biofilm's ability to absorb nutrients is also heavily affected by how much wastewater that comes into the treatment plant. The measurement of the flow rate of wastewater (F_S) at the inlet is therefore crucial for estimation of the polyphosphate levels at the disc filter. Soluble chemical oxygen demand (sCOD, S_{Sins}) at the inlet would be an important measurement since if there are high levels of sCOD it will cause an increase of biological oxygen demand, which effects the biofilm's ability to break down organic matter. Nitrogen dioxide (NO_2) and Nitrate (NO_3) is important nutrients. Organisms are reliant on it for their survival and growth. However, too much of it leads to eutrophication problems.

Other EBPR processes around the world mostly use active sludge-based processes which struggles with efficiency and stability (Rudi et al., 2019). The Hias process has replaced the active sludge-based process with a MBBR process. The benefits with this are that MBBR approaches can maintain low process volume while also being cost effective and continue a stable phosphorus removal process (Helness & Ødegaard, 1999).

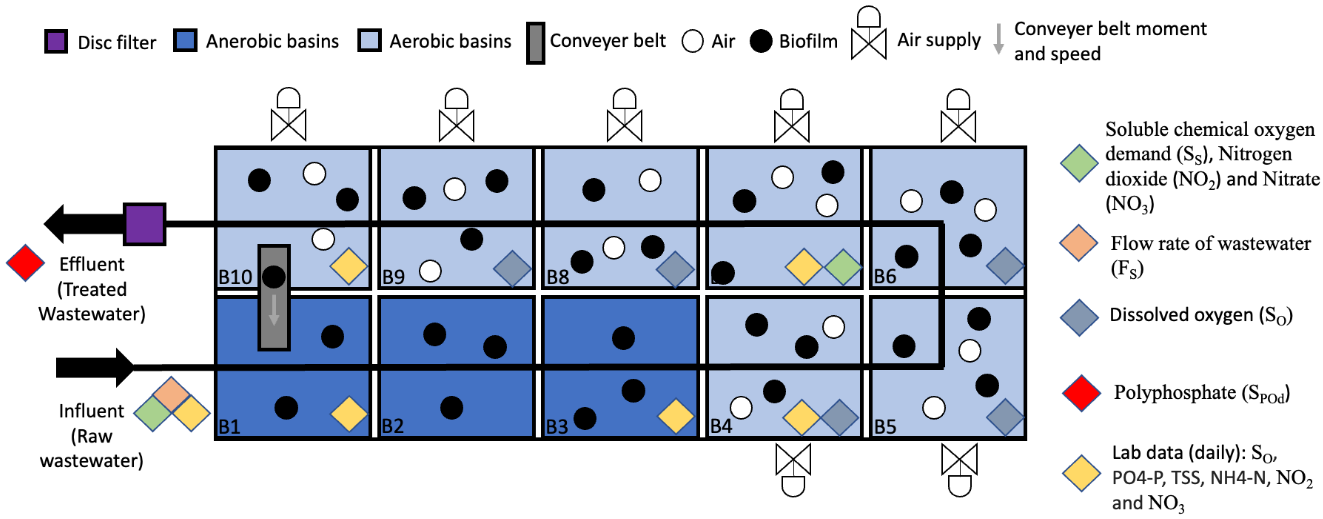

Figure 1 System description of the Hias process

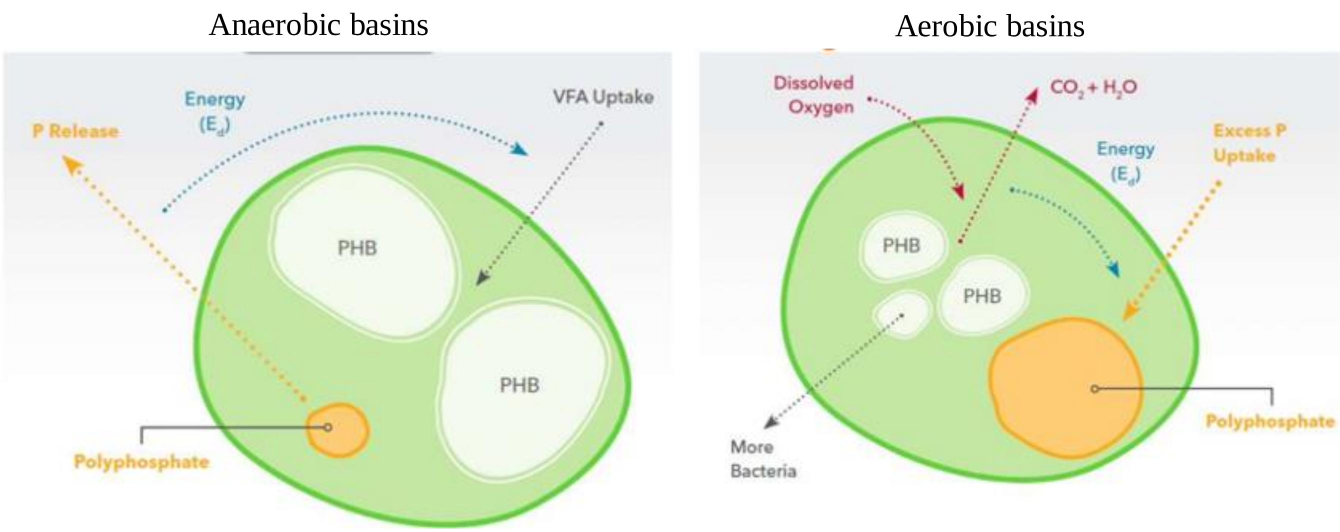

Figure 2 The biofilms behavior in anaerobic and aerobic basins (Phosphorus in wastewater, Analysis Removal Strategies, 2023)

The complete online data for December of 2022 are collected industrial data from the Hias process in Hamar, Norway. The online data has a sampling time of 10 minutes, containing 4399 data points in total. It includes online measurements of wastewater flow (F_S)-, soluble chemical oxygen demand (S_{Sins})- at the influent which is two of the disturbance variables (DVs) seen in Figure 1. Nitrogen dioxide (NO_2) and Nitrate (NO_3) combined (NOX) at the influent is the third disturbance variable for this project. Disturbance variables are variables that creates a lot of noise or instability to a system. The fourth DV are polyphosphate (S_{PO}) at the influent, and this are being estimated by another person on the same project, Bipasha Mukherjee. She's doing this by using regression with S_{Sins} and the F_S as one variable and S_{PO} as another in the influent. Since her work won't be available in time for this master's thesis, the use of transfer function with only S_S and the F_S to estimate the S_{POd} out of the system will be the best solution. This will deem it unnecessary to have an estimation of S_{PO} at the influent. The last disturbance variable is temperature, unfortunately, there are no online measurement of this variable.

The datasets contain online measurements of flow rate of oxygen (F_O) for all the aerobic basins as well. These are the manipulated variables (MVs) in the process. Manipulated variables are the variables in a system that you want to adjust or change to get the most optimal control variables (CVs). The control strategy must be able reject and dampen the DVs as sufficient as possible to contain the set point which is based on the control variable.

The online data for polyphosphate (S_{POd}) that comes out of the disk filter is being collected as well, this is the control variables of the system. A control variable is the desired variable of a system. If the measurements that are being controlled by control strategies can follow CV sufficiently it is a suitable control strategy that has the potential for industrial implementation.

The lab data will not be used for this master's thesis. However, it has been a good instrument for validating assumptions. The industrial dataset also includes online measurements of dissolved oxygen (S_O) in basin 4, 5, 6, 8, and 9. A formula was created for basins 7 and 10, and interpolated values were then used as a substitute. These were originally some of the CVs. However, the control strategies for S_O were not successful since there should have been implemented transfer functions (TFs) between S_O and polyphosphate that comes out of the disk filter (S_{POd}) to get it to work.

The datasets used for the project should have minimal missing values, it should also be a period where the S_{POd} fluctuates as much as possible. Taking this into consideration the period of 20/12/22 until 25/12/22 (week 51) for one of the datasets, and the period of 13/12/22 until 18/12/22 (week 50) was chosen as the datasets used for the master's thesis. Another dataset was originally used as the validation dataset and was the period 06/12/22 until 11/12/22 (week 49). The first time being 07:50 and the last time being 23:40 for all the datasets. The names of the datasets were given as Hias_onlinedata_w49 for the week 49 dataset, Hias_onlinedata_w50 for the week 50 dataset and Hias_onlinedata_w51 for the week 51 dataset.

In Table 1 the variables used for this master's thesis have been provided. The variables without nominal values are just variables needed to explain certain aspect of this paper. The nominal values are the mean of each of the variables during week 51 of 2022. The variables that were scaled has the "s" notation after the variable name to differentiate scaled and unscaled variables, they are all scaled by dividing them by 1000. The scaled variables are used for the whole project. The unscaled variables were only used for the first experiments for system identification. The units will in this case not be accurate for the scaled variables. The units placed in this table are the original units for the variable. It is common to work with dimensionless variables in control engineering. The NOX and F_S variable should have had the "in" notation in its name as well. The variable column describes which type of variable it is, where DV is disturbance variables, MV is manipulated variables, and CV is control variables. The description column explains the variable shortly.

Table 1 Online measurements in the Hias process and other variables

Variable	Description	Nominal value at t=0	Unit	Variable type
S_F	Readily biodegradable substrate		mg COD/ L	
S_A	Volatile fatty acids/acetate (fermentation products)		mg COD/ L	
PP	Stored polyphosphate in biomass		mg P/L	
PHA	Stored PHA in biomass		mg COD/ L	
V	Volume of one basin	215	m^3	
F_S	Flow rate of wastewater that comes in	0.087	L/s	DV
S_S	Soluble chemical oxygen demand (sCOD)		mg COD/L	
S_{Sins}	Soluble chemical oxygen demand (sCOD) in inlet	0.5088	mg COD/L	DV
NO_2	Nitrogen dioxide		mg m^3/L	DV
NO_3	Nitrate		mg/L	DV
NOX	NO_2 and NO_3 combined in inlet	2.6089	mg/L	DV
F_O	Air supply, flow rate of oxygen (aeration)		Nm^3/h	MV
FO4s	Flow rate of oxygen (aeration) in B4	2.9980	Nm^3/h	MV
FO5s	Flow rate of oxygen (aeration) in B5	1.8193	Nm^3/h	MV
FO6s	Flow rate of oxygen (aeration) in B6	1.5715	Nm^3/h	MV
FO7s	Flow rate of oxygen (aeration) in B7	1.0550	Nm^3/h	MV
FO8s	Flow rate of oxygen (aeration) in B8	0.8462	Nm^3/h	MV
FO9s	Flow rate of oxygen (aeration) in B9	0.6239	Nm^3/h	MV
FO10s	Flow rate of oxygen (aeration) in B10	0.5214	Nm^3/h	MV
S_O	Dissolved oxygen O2		mg O2 /L	CV
SO4	Dissolved oxygen O2 in B4	5.3729	mg O2 /L	CV
SO5	Dissolved oxygen O2 in B5	6.0622	mg O2 /L	CV
SO6	Dissolved oxygen O2 in B6	5.8070	mg O2 /L	CV
SO7	Dissolved oxygen O2 in B7	5.4728	mg O2 /L	CV
SO8	Dissolved oxygen O2 in B8	5.1387	mg O2 /L	CV
SO9	Dissolved oxygen O2 in B9	5.0493	mg O2 /L	CV
SO10	Dissolved oxygen O2 in B10	4.9598	mg O2 /L	CV
S_{PO}	Polyphosphate PO4-P		mg P/L	
S_{POd}	Polyphosphate PO4-P after disk filter	0.2325	mg P/L	CV

III. Materials and Methodology

A. Hardware and software

The software was run on a Macbook Air 2020, 16 gig ram and m1 processor. Google Colab with python version 3.10.11 was used for data analysis and preprocessing. The data analysis and preprocessing are implemented in the python code, this can be located in the appendix section IX.C. Microsoft Excel for Mac version 16.71 (23031200) was used for preprocessing. Matlab software package version R2022a was used for the simulations. The simulation method was ode15s with automatic settings for the time step and error tolerance. The model parameters and test procedures are implemented in m-script, seen in the appendix section IX.A. The datasets are imported to Matlab System Identification toolbox, where different models are being tested to get the best parameters and results possible.

B. Simulation models

The simulations models were developed using Simulink, they can be seen in the appendix section IX.B. Where Figure 28 shows the simulation model for the dynamic linear model. Figure 29 describes the simulation model for the linear model. Figure 30 is the simulation model for the Proportional-integral-derivative (PID) controller. Figure 31 shows the simulation model for the model predictive controller (MPC).

C. Collection of data and pre-processing

The operational data was obtained by using the Industrial IoT platform KYB. The platform was developed by Digitread Connect. The online data used for this project was gathered in .csv format. The datasets naturally contained outliers, these where removed. There was also missing values that needed to be filled. This can be done by interpolating, or filling values between two datapoints with data that obtains the same dynamics as the rest of the dataset. This was done with K-nearest-neighbor (kNN) and manually interpolating in Excel with the series function. kNN is a machine learning algorithm that finds the closest value in the dataset and uses it to fill in for the missing values between two points (*Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties on JSTOR*, 2023). Linear interpolation with the "interp1" function in Matlab was also done after importing the datasets. Linear interpolation uses values between two datapoints and will fill them with increasing or decreasing values and not the same value for every missing value which is what kNN does (Meijering, 2002).

D. Sampling time

The time it takes for the wastewater to flow from the inlet to the outlet in the Hias process can be expressed by using time delays that are representative of the process. This delay can be derived by the equation below, eq. (1), where V is the volume each basin, which must be multiplied with 10 basins. $\mathrm{F_{mean}}$ is the mean of the flowrate of wastewater that comes into the system. This variable had to have the $\mathrm{m^3/min}$ unit. The variable for wastewater in Table 1 section II uses the scaled $\mathrm{F_S}$ (0.087) where the unit is for this variable are actually for the unscaled variable ($\mathrm{L/s}$). This means it must be multiplied by 1000 to get L/s (this gives 87 $\mathrm{L/s}$). To get it in minutes it must be multiplied with 60 (5220$\mathrm{L/min}$). Then there is a need to convert liters into cubic meters, this is done by dividing by 1000 (5.22$\mathrm{m^3/min}$):

$$\frac{10 \cdot \mathrm{V}}{\mathrm{F_{mean}}} = \frac{10 \cdot 215\mathrm{m^3}}{5.22\mathrm{m^3/min}} = 411.8774\mathrm{min} \tag{1}$$

The time delays ($\mathrm{T_d}$) can be seen in Table 3. However, the delay for variables in basin 4-10 will vary. For basin 4 for example it would be a sampling time of 164.75min, the next basin would be 205.94min and so on.

E. Dynamic models

The dynamic linearized models were derived by transfer functions that were obtained by the Matlab system identification toolbox. The general principle of system identification can be seen in Figure 3. The experiment design in this case would be the design of the Hias process. The online measurements collected from the Hias process would be the second part. There are many different model sets that can be chosen. The model sets can be state-space models, transfer function (TF), polynomial models, and many others. For this project many different variations of TFs were explored. Zeroth order, first order, second order TFs with and without time delay. A zeroth order TF would not have any time constant, while a first order would, and a second order TF would have two time constants. Many of these models produce adequate results. However, the problem comes in when the criteria are being chosen. Since the transfer functions should represent the system in the best possible way, the parameters are the most important part. For the

Hias wastewater resource and recovery facility (WRRF) the K_p value should be negative for all the aeration (F_O) values and be a small number to remove polyphosphate (S_{POd}). While the K_p value for the disturbance variables (S_{Sins}, F_S, and NOX) should be positive and relatively small to preserve the quality of the wastewater. The time constants and time delays should simulate the time each of the inputs would take in the real process as close as possible. If the chosen model and parameters (criteria) gives bad result (calculated model) it would have to be revised. This will be done over again until the results are adequate to move on. This is decided based on the final prediction error (FPE), mean squared error (MSE), fit to validation- and estimation data. The FPE and MSE values should be as low as possible, while fit to validation- and estimation data should be as close to 100% as possible.

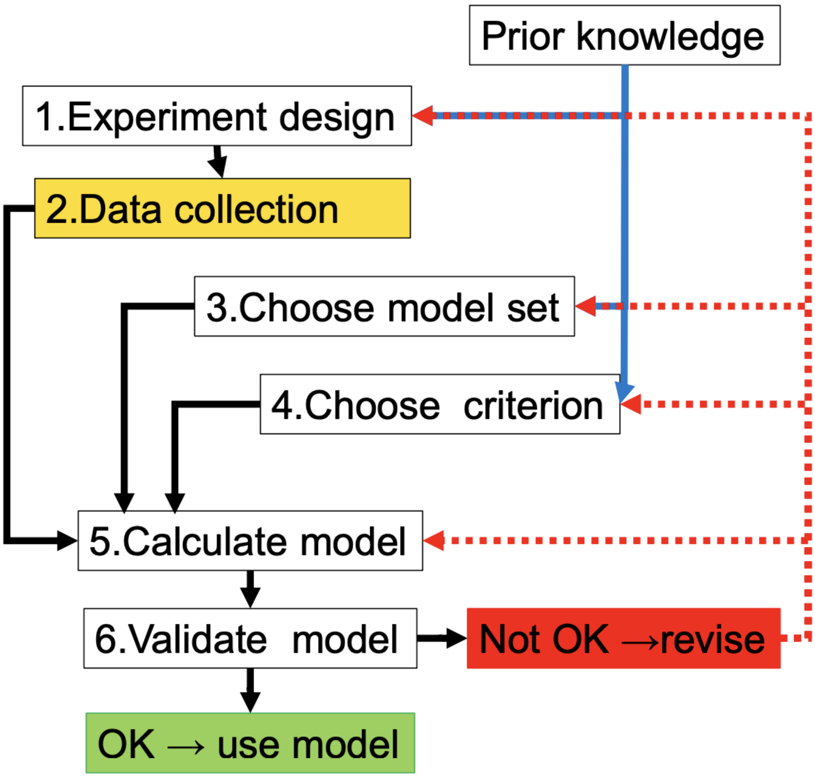

Figure 3 How to perform system identification (Ljung, L. (1999) Chapter 1 figure 1.10 P.15)

The transfer functions for polyphosphate in the disk filter (S_{POd}) can be seen in eq.(2). The equation describes a second order transfer function between the output Y(s) (S_{POd}) and the inputs $U_i(s)$ (S_{Sins}, F_S, NOX, FO4s-FO10s). Where Y(s) contains one gain (K_p), two time constants (T_{p1} and T_{p2}) and one time delay (T_d). The values in the Table 6, Table 7, and Table 8 section 0 are based on equation (2). The same goes for every other table in the appendix chapter IX. The same goes for the tuning of the control strategies. The parameters and transfer functions for S_{POd} can be seen in Table 2. The FO7s_S_{POd} and FO8s_S_{POd} has been replaced with FO5s_S_{POd} since FO7s_S_{POd} and FO8s_S_{POd} originally had positive K_p which is not desirable.

$$\mathrm{TF}(s) = \frac{Y(s)}{U_i(s)} = \frac{K_p}{(1 + T_{p1}s)(1 + T_{p2}s)} e^{-T_d s} \tag{2}$$

Table 2 Transfer functions for S_{POd} (also named tf_ S_{POd} and tf_s51sfnfspo) by using eq. 2

Transfer function name	Input variable	Output variable	K_p	T_{p1}	T_{p2}	T_d
S_{Sins} _S_{POd}	S_{Sins}	S_{POd}	0.43109	60	2.3332	359.72
F_S _S_{POd}	F_S	S_{POd}	1.2924	30	9.2551	79.46
NOX_S_{POd}	NOX	S_{POd}	0.0023115	46.596	0.0026751	147.61
FO4s_S_{POd}	FO4s	S_{POd}	-0.0029016	18.919	24.979	47.75
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	23.394	28.69	81.34
FO6s_S_{POd}	FO6s	S_{POd}	-0.047375	49.545s	6.1161	57.45
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	23.394	28.69	81.34
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	23.394	28.69	81.34
FO9s_S_{POd}	FO9s	S_{POd}	-0.053278	60	13.342	304.8
FO10s_S_{POd}	FO10s	S_{POd}	-0.21394	60	1.3255	315.41

To get the right format for m-script the equation had to be reformulated. The formula for the transfer functions for polyphosphate (S_{POd}) can be seen in eq. (3). This will give different values for K_p, T_{p1}, T_{p2}, which is presented in Table 3

$$TF(s) = \frac{Y(s)}{U_i(s)} = \frac{K_p}{\left(T_{p1}\, s^2 + T_{p2} s + 1\right)} e^{-T_d s} \tag{3}$$

Table 3 Transfer functions for S_{POd} (also named tf_ S_{POd} and tf_s51sfnfspo) using eq. 3

Transfer function name	Input variable	Output variable	K_p	T_{p1}	T_{p2}	T_d
S_{Sins} _S_{POd}	S_{Sins}	S_{POd}	0.43109	139.992	62.332	359.72
F_S _S_{POd}	F_S	S_{POd}	1.2924	277.653	39.2551	79.46
NOX_S_{POd}	NOX	S_{POd}	0.0023115	0.12464	46.598675	147.61
FO4s_S_{POd}	FO4s	S_{POd}	-0.0029016	472.577701	43.898	47.75
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	671.19725	52.085	81.34
FO6s_S_{POd}	FO6s	S_{POd}	-0.047375	303.0221745	55.6611	57.45
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	671.19725	52.085	81.34
FO5s_S_{POd}	FO5s	S_{POd}	-0.029011	671.19725	52.085	81.34
FO9s_S_{POd}	FO9s	S_{POd}	-0.053278	800.52	73.342	304.8
FO10s_S_{POd}	FO10s	S_{POd}	-0.21394	79.53	61.3255	315.41

F. Experimental plan for system identification

Table 4 illustrates the experimental plan for the system identification part of this project. Almost every test conducted for system identification can be seen in this table. For validation data the week 49 dataset has been chosen first and is the validation data for tf_b4_foso, tf_b5_foso, tfb4_ffoso, tf_b5_ffoso, tf_sspo, tf_sfsspo and tf_sfnfspo. The rest of the TFs has week 51 as validation data instead. The week 50 dataset is being used as the estimation data. The inputs and outputs for each of the transfer functions (TFs) can be seen in Table 4. The first TFs being for the control variable dissolved oxygen (S_O), which will not be explored further for this project. While the other TFs are for the main control variable, which is the polyphosphate out of the disc filter (S_{POd}).

Tf_b4_foso and tf_b5_foso were the TFs that was tested first without F, then with F and S_{Sin} and lastly with F_S and S_{Sins}. F being the flow rate of wastewater before scaling it by 1000. S_{Sin} being the Soluble chemical oxygen demand in the inlet before scaling. F_S and S_{Sins} are the same variable just with scaled values. This was done to preprocess and fine tune the model set before doing the same for the rest of the TFs. Some work was done on tf_b4_sffoso, tf_b5_sffoso, tf_b6_sffoso, tf_b7_sffoso, tf_b8_sffoso, tf_b9_sffoso, tf_b10_sffoso before scaling the variables completely in the dataset for tf_s51sfnfspo. Some documentation was made and will be included in the appendix 3. However, it was decided to redo them all with week 51 as validation dataset. The first tests with just aeration rate (F_O) and dissolved oxygen (S_O) was done with week 49 as validation data. The tests with F included was also done with week 49 as validation data. However, all the TFs with F_O, S_O, F_S and S_{Sins} as input and S_O as output was done on the scaled datasets for week 50 and week 51. The variables that were scaled has the "s" notation after the variable name to differentiate scaled and unscaled variables. The scaled variables are used for the whole project. The unscaled variables were only used for the first attempts for system identification seen in Table 4.

The transfer functions for polyphosphate (S_{POd}) were first tested on the week 49 dataset (tf_sspo, tf_sfsspo, tf_sfnfspo). The results here was not adequate. Later in the project the validation dataset was swapped with week 51 dataset (tf_51sfnfspo). This improved the results for S_{POd}. However, the scaling of S_{Sins}, F_{s}, and all the F_{O} improved the result even further. There were also tests done with fewer inputs to check if this would increase the fit to estimation- and validation data (tf_sfnf456spo and tf_sfnf4510spo). This was not the case.

Table 4 Experimental plan for the system identification with their transfer function name

Transfer function name	Input	Output
tf_b4_foso	FO4	SO4
tf_b5_foso	FO5, SO4	SO5
tf_b4_ffoso	FO4, F	SO4
tf_b5_ffoso	FO5, SO4, F	SO5
tf_b4_ffoso	FO4, F, S_{Sin}	SO4
tf_b5_ffoso	FO5, SO4, F, S_{Sin}	SO5
tf_b4_sffoso	FO4s, F_{s}, S_{Sins}	SO4
tf_b5_sffoso	FO5s, SO4, F_{s}, S_{Sins}	SO5
tf_b6_sffoso	FO6s, SO5, F_{s}, S_{Sins}	SO6
tf_b7_sffoso	FO7s, SO6, F_{s}, S_{Sins}	SO7
tf_b8_sffoso	FO8s, SO7, F_{s}, S_{Sins}	SO8
tf_b9_sffoso	FO9s, SO8, F_{s}, S_{Sins}	SO9
tf_b10_sffoso	FO10s, SO9, F_{s}, S_{Sins}	SO10
tf_sspo	SO4, SO5... SO10	S_{POd}

tf_sfsspo	SSin, F, SO4, SO5... SO10	S_{POd}
tf_sfnfspo	SSin, F, NOX, FO4, FO5…FO10	S_{POd}
tf_51sfnfspo	S_{Sin}, F, NOX, FO4s, FO5s…FO10s	S_{POd}
tf_s51sfnfspo	S_{Sins}, F_s, NOX, FO4s, FO5s…FO10s	S_{POd}
tf_sfnf456spo	S_{Sins}, F_s, NOX, FO4s, FO5s, FO6s	S_{POd}
tf_sfnf4510spo	S_{Sins}, F_s, NOX, FO4s, FO5s, FO10s	S_{POd}

G. Simulation of dynamic linear model

Figure 28 in the appendix section IX.B illustrates the dynamic linear model. The simulation model contains the deviation variables that goes into the measurement or transfer function for polyphosphate (tf_S_{POd}). This measurement is compared against the variable of the polyphosphate out of the disc filter (S_{POd}) which is virtual sensor for S_{POd}. A deviation variable is a variable that operates around zero on the y-axis. The transfer function or plant in Matlab only operates with deviation variables. The disturbance and manipulated variables could have been deviation variables if the “remove means” function in system identification was used. However, this function was used at first and then not used again. That’s why it was needed to subtract the disturbance and manipulated variables with the mean of the same variables to get them to operate around zero which will make it a deviation variable. The mean had to be added again after the measurement to get the measurement to operate around right point on the y-axis.

H. Proportional-integral-derivative (PID) controller

The simulation model for PID controller is seen in Figure 30 in the appendix section IX.B. Figure 4 is a simplified representation of the simulation model for the PID. The closed-loop feedback PID control system will have the ability to correct itself when disturbance occurs in the system. The disturbances being soluble chemical oxygen demand (S_{Sins}), flow rate of wastewater (F_S), and NO_2 and NO_3 combined in inlet (NOX) at the inlet of the Hias wastewater treatment and water resource recovery facility (WRRF). The measurement of polyphosphate (S_{POd}_measurement) or the transfer function for S_{POd} will receive the disturbance variables and the manipulated variables as inputs. The manipulated variables being the aeration rate for each basin (FO4s-FO10s). The goal of the measurement is to follow the set point of the polyphosphate (S_{POd}_set point) to the best of its ability. The way it achieves this is by the manipulated variables being adjusted by the PID controller and the ratio controllers. The ratio controllers are only a percentage or gain of the PID controller. A PID controller is only able to have one input and one output. They are controlling the air supply of the valves in the Hias process. Tuning these two types of controllers to follow the set point is the goal. The set point is the most desirable operation point of a process. If the measurement can follow it completely, the system will not use any unnecessary energy. This is the ideal outcome. However, it is unrealistic to be able to achieve this because of disturbances and changes to the system.

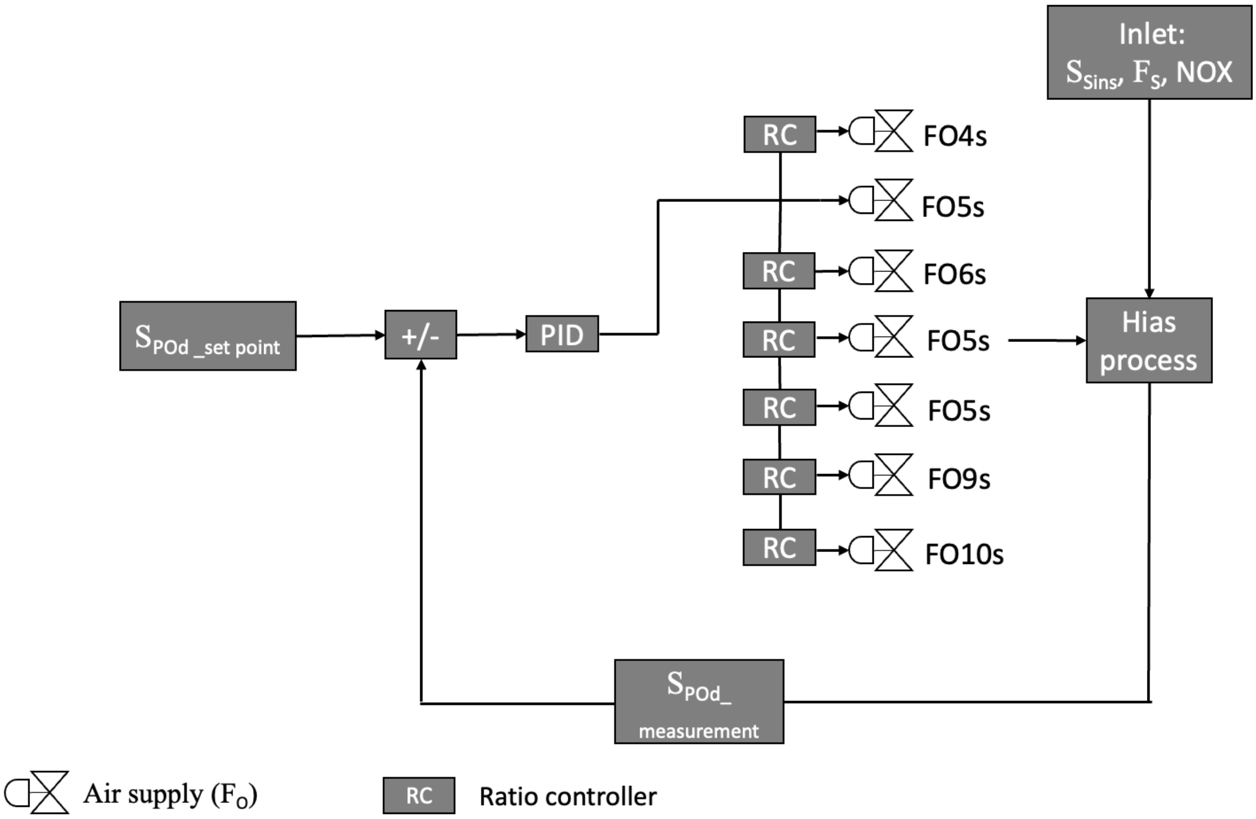

Figure 4 Control design for PID

1) PID controller algorithm

The feedback part of the controller is a set in series instead of parallel since the process is has a significant time delay. The series algorithm for a PID controller can be seen in eq. (4) and is based on (Skogestad, 2003). K_c are the controller gain of the PID, τ_i is the integral time, and τ_d are the derivative time.

$$c(s) = K_c \cdot \left(\frac{\tau_i s + 1}{\tau_i}\right) \cdot (\tau_d s + 1) \tag{4}$$

2) Tuning PID controller based on Skogestad IMC tuning rules

By investigating Skogestad (Skogestad, 2003) IMC tuning rule we can assume that $\theta \approx \tau_c$ from earlier eq. (2). By reformulating eq. (2) to fit the recommended PID controller parameters for a second order process, the gain of the controller (K_c) will be as follows:

$$TF(s) = \frac{K_p}{(1 + \tau_1 s)(1 + \tau_2 s)} e^{-\theta s} \tag{5}$$

$$K_c = \frac{\tau_1}{K_p(\tau_c + \theta)} \tag{6}$$

The integral time (τ_I) would be formulated like this according to Skogestads tuning rules:

$$\tau_i = \min\{\tau_1, 4(\tau_c + \theta)\} \tag{7}$$

The derivative time (τ_d) is only the second time constant (τ_2) according to Skogestad.

I. Model predictive controller (MPC)

The simulation model for MPC is seen in Figure 31 in the appendix section IX.B. Figure 5 illustrates the same process as Figure 4 only with one key difference. Instead of the aeration rate or the manipulated variables being controlled by PID and ratio controllers, it is controlled by a MPC instead. The MPC can have multiple inputs and multiple outputs. The MPC will adjust the manipulated variables to obtain S_{POd}_measurement that follows the S_{POd}_set point hopefully better than that of the PID. When developing control strategies for a system, the objectives, constraints, and test procedures should be the exact same to get a fair comparison between them. However, the algorithm or the way the control strategy manipulates the system can be completely different.

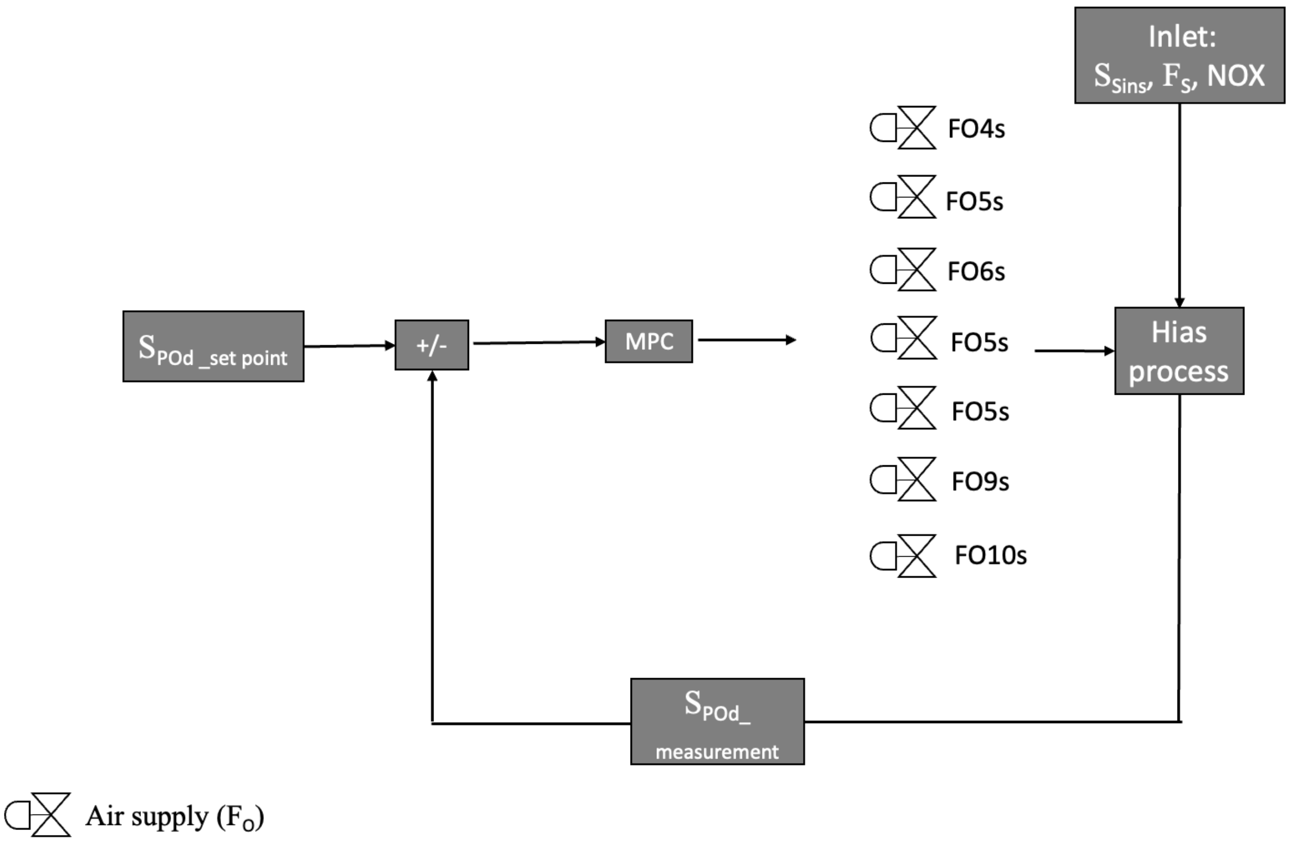

Figure 5 Control design for MPC

1) The MPC algorithm

The MPC algorithm is illustrated in Figure 6. The control horizon, M is the period the MPC algorithm can influence the system. The prediction horizon P is a period where the MPC can predict how the system will behave. The MPC will utilize mathematical models that represents a certain system to find the ideal values for the manipulated variables to reach the set point as effective as possible. The manipulated variables should do this within the control horizon M. It does this to get the control variables of the system to obtain the optimal operation point within the prediction horizon P. The MPC algorithm can be categorized as a horizon algorithm. This means that after each time step new calculations are made. The controller will only execute the next control action based on the most optimal value for the manipulated variable.

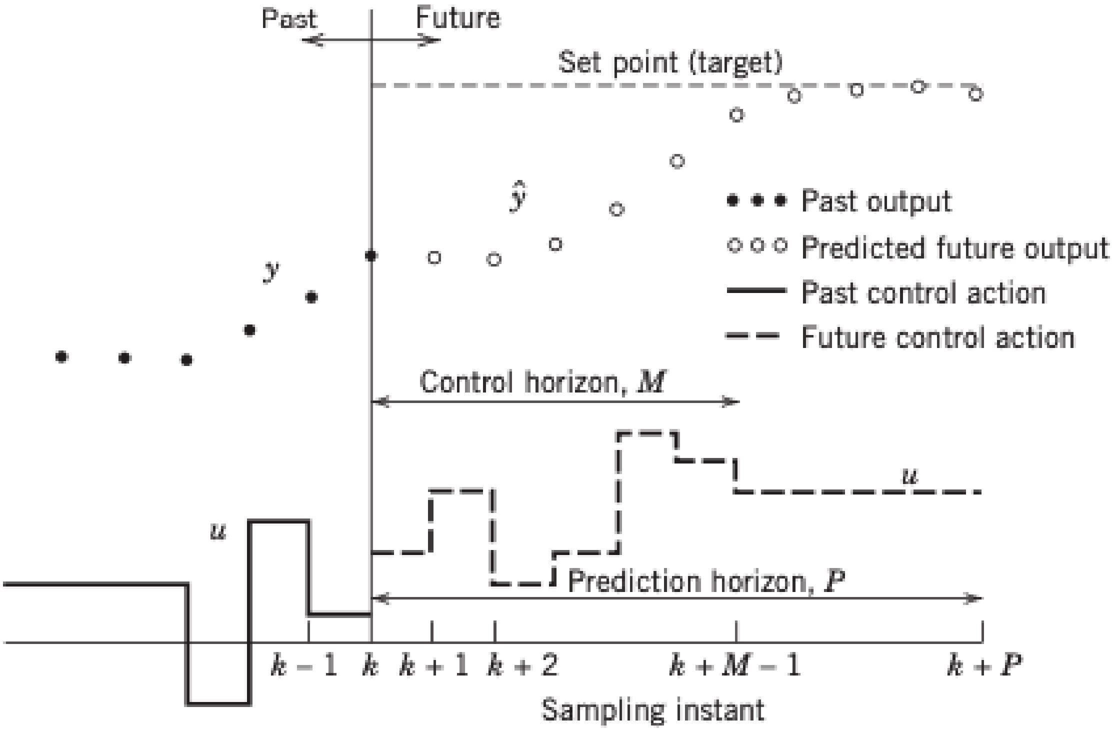

Figure 6 Model predictive controller algorithm (Seborg et al.,2017)

2) Tuning of MPC

Seborg et al. rules for parametrization are a great starting point to obtain initial tuning parameters. These initial parameters are based on the dynamic linear model of a system. The process time constant (τ) for this system is seen in eq. (1). The sampling time for the MPC should be one tenth of process time constant:

$$\mathrm{T}_{\mathrm{sampling}} < \frac{\tau}{10} \tag{8}$$

The settling time according to Seborg should be:

$$\mathrm{T}_{\mathrm{settling}} \approx 4\tau + \theta \tag{9}$$

The modeling horizon N should be a value between 30 and 120. However, Seborg also argues that different modeling horizons can be used as well. A general rule for selecting the model horizon is to subtract the settling time ($\mathrm{T}_{\mathrm{settling}}$) with the sampling time ($\mathrm{T}_{\mathrm{sampling}}$):

$$N = \frac{T_{settling}}{T_{sampling}} \tag{10}$$

The control horizon M should be between $\frac{N}{3}$ and $\frac{N}{2}$:

$$\frac{N}{3} \leq M \leq \frac{N}{2} \tag{11}$$

The prediction horizon P should be the sum of the modeling horizon N and M:

$$P = N + M \tag{12}$$

The weighting matrixes is also important parameters. The weighting matrix Q can have the initial value as 1 and then be tuned accordingly. The weighting matrix R initial value should be low to suppress the movement of the manipulated variables. The starting value could be 0.1.
Appropriate constraints should also be set for all the manipulated variables based on the system.

J. Experiental plan – Test procedure for the control strategies

The experimental plan for seen in Table 5 shows the step changes in the disturbance variables (S_{Sins}, F_s and NOX) and the control variable (S_{POd}) that was used for the control strategies. The step changes are introduced to test the ability of the control strategy to reject the disturbances.

Table 5 Test procedure for control experiment

Time [min]	Action	Collected disturbance variables
t=0	Initial values …	
t=0	Start simulation	
t1=100	+5%	S_{POd}
t2=700	-5%	
t3=2000	+5%	S_{Sins}
t4=2600	-5%	
t5=5000	-5%	F_s
t6=5600	+5%	
t7=7000	-5%	NOX
t8=7600	+5%	
t9	Stop simulation	

K. Experimental plan – Control error indicies for the control strategies

The controllers are compared using the integral of absolute error (IAE) between the controlled variable and its setpoint. The integral of total movement in manipulated variables (IAMV). The IAMV shows how big the amplitude changes are of a signal and its ability to obtain the setpoint. While the IAE shows the difference between the control variable and the setpoint.

The IAE is defined as:

$$\int |e(t)|dt = \int \left|y_{sp}(t) - y(t)\right|dt$$

The IAMV is defined as:

$$\int |u(t) - u(t - T_s)|dt$$

IV. RESULTS AND ANALYSIS

A. Preprocessing the datasets based on results gotten from system identification

To get the correct dataset when importing inputs and output you must switch places the different datasets (week 49, 50, 51). The week 51 dataset replaced week 49 dataset eventually and was used for all the TFs eventually. The variables were there are not a "s" notation behind them in this section shows that they are not scaled.

The soluble chemical oxygen demand in the inlet without scaling (S_{Sin}) variable was not calibrated when the complete dataset arrived. Therefore, the variable came separately with a five-minute sampling time instead of ten-minute sampling time, and in a combined column that needed to be separated. The other variables have a 10 minute sampling times. That's why a method for deleting every other row for the new calibrated variable was necessary. Using zero and ones in a separate column where the ones aligned with the times needed for the variable had to be done. This was done by copying the one number one and one zero that was aligned with correct times and pressing "opt+shift+down" on the cell right under the single one and zero. Then clicking paste for the marked area. There is a filter for every column that has the option to remove every row that includes a zero or a one. Clicking away the zeroes will now result in there only being every other row with the values needed for the rest of the dataset. This was done in Google sheets and then copied and pasted into excel. The columns also needed to be separated into the times and values for S_{Sin}. This was done in excel by using the "Text to Columns" feature and clicking on the semicolon, which is the symbol that separates the times and values. This will generate two separate columns. All of this was then included into the original dataset with the correct time periods.

The datasets were preprocessed by using K nearest neighbor (Knn) for a limited and selected period (week 49 and 50). The code used for performing this can be seen in Appendix IX.C. Knn uses the nearest value between datapoints and inserts the same value in all the missing values to interpolate. Knn did this for most of the values. Removing all outliers such as too high values and NaN values that still occurred was also done by looking at plots in both excel and Matlab. There were not many of them, that's why interpolation in excel between the nearest values was an adequate solution to the problem. This was done for some values in S_{POd} and F_S. Documentation for those values was not made unfortunately. It was also done for values that superseded 4500 in Air_B5 and Air_B6. The values are for 07/12/22 and the times were between 14:00 and 16:00 for week 49 dataset. The table of all the different TFs developed with system identification can be seen in Table 4 Experimental plan for the system identification with their transfer function name. Discovering that new dataset still had too big of a peek for FO5 since a first order transfer function model were at -24.3 for tf_b5_ffoso. This jumped to -10.16 for the same TF after removing the outlier. Therefore, a new dataset was created where the peek was reduced even more by interpolating in excel. The values are still for 07/12/22 and the times were between 13:50 and 20:10 for week 49 dataset instead. Interpolation in excel was also done for FO5 and FO10 for week 50. The outlier values were at 16/12/22 between 07:10 until 08:00. There were also outliers in FO8 for 13:00 and 13:10

on 16/12/22 for week 50. This was done after discovering outliers when looking at the week 49 dataset for tf_b5_ffoso in the system identification. The previous dataset was used for tf_b4_ffoso since there was no outliers in the datasets for the inputs for this transfer function.

There has been done a lot of undocumented testing as well. However, most of the testing will be presented in later in this section. Rescaling the datasets with the remove means function in system identification was done after discovering the problem after a meeting with CEIWA from Tampere University of Applied Science. This was when tfb4_ffoso was under development. Most of the testing, fine tuning and removing of outliers was done on tf_b5_ffoso since this is one of the most representative basins of the system since it's the earliest basin with two initial inputs. Therefore, the argument of showing these results first can be made. However, a chronological order of the results would make more sense. There was also fortunately not a lot of preprocessing needed when dealing with SO4 except for scaling the datasets. That's why the results of the tf_b4_ffoso was not redone when replacing the datasets for tf_b5_ffoso. The final tf_b4_foso and tf_b5_foso was made after tf_b4_ffoso and tf_b5_ffoso.

After discovering that the S_{POd} variable had outliers revolving around almost the same value, were all of them started with 0.23 for all three of the datasets when looking at the recommended diagram in excel. It was easy to spot the values and replace them with interpolated values. They occurred every hour. There was also a big peek at 07/12/22 between 18:20 and 19:30 for the week 49 dataset. There was also too sharp of a peek at 18/12/22 between 03:40 and 04:10. Another peek at 15/12/22 between 08:20 and 11:00 in the week 50 dataset. The week 51 dataset had many 0.23 anomalies, every hour, and some additional ones as well.

NOX also needed some interpolation. For week 49, there were anomalies with the same value, 2,58350974, that had to be removed. This was at 9 out 14 of the values between 9:50 and 12:00 for 07/12/22. It was also the same anomaly for values between 19:30 until 20:00 for the same day. This anomaly occurs the next day at 12:20 until 13:00. And the last one at 08:50 for 09/12/22 for week 49. The same value appeared for the week 50 dataset but only for one day between 19:20 and 00:40. There was also to sharp of a peek at 07:30 on 16/12/22.

Tf_sfnfspo gave very bad Kp values, they were too low which will result in making it hard to do control on them. Therefore, it was decided to scale the F_O just for tf_sfnfspo. This was done by copying all the F_O variables and divide them by 1000 in excel. These variables were given a "s" notation behind them to differentiate them from the original F_O variables. This improved the results significantly. However, there should have been scaling done on F and S_{in} as well. This was implemented later.

There were many difficulties working with the week 49 dataset when working on tf_sfnfspo. Therefore, it was decided that some additional testing would be conducted using week 51 instead of week 49 as validation dataset. One of the reasons why 51 was not picked first was because there are some missing values on 19/12/22 between 22:30 and 23:30 for all the variables. However, these values have been interpolated for using Knn. The new dataset also needed some data preprocessing. Variables that were going to be used for tf_51sfnfspo was the only ones that was checked for preprocessing. FO10s had to sharp of peeks at 24/12/22 between 21:40 and 22:30. Interpolation was done to reduce it somewhat. The 2,58350974 anomaly appears in the NOX variable at 24/12/22 between 01:40 and 02:10, and some more between 02:50 and 04:20 the same day. The last anomaly was at 24/12/22 at 22:00 for NOX. F also had a zero value at 24/12/22 at 21:40. The 0.23 anomaly for S_{POd} appeared for the week 51 dataset as well.

B. All the plots for all the variables for week 50 dataset and week 51 dataset

The plots shown is for the week 50 and 51 datasets after preprocessing them with the sampling time of ten minutes. This will generate 817 datapoints with a sample of the data after 10 minutes. This is the data used for system identification. When importing data for system identification it's possible to set the sampling time. This was set to 10 to get every minute of the process. The variables present dynamical changes that would occur in an industrial process. This was the goal when preprocessing the variables, to not remove too much of the dynamics of the variables.

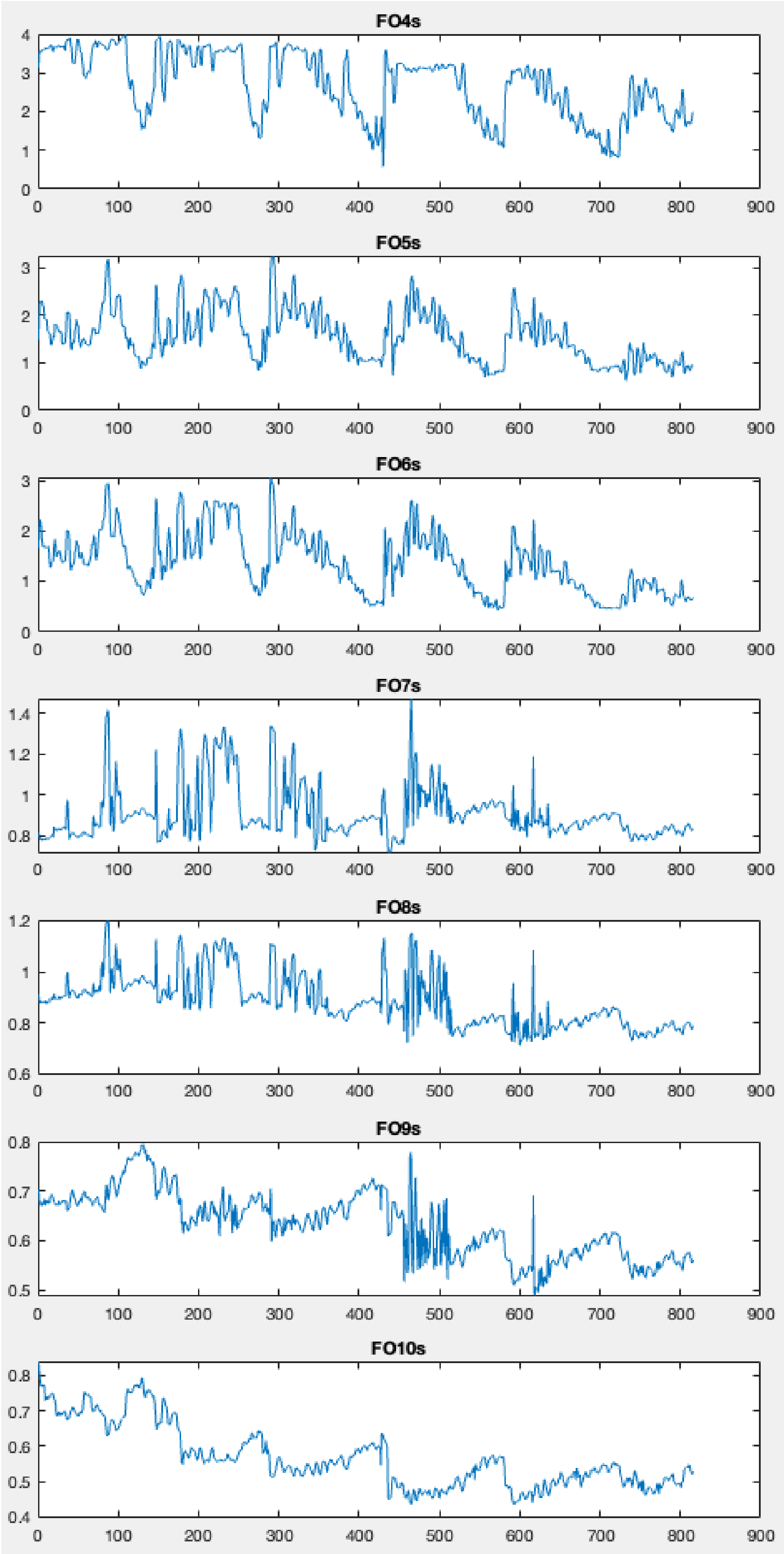

Figure 7 F_O for week 50 dataset

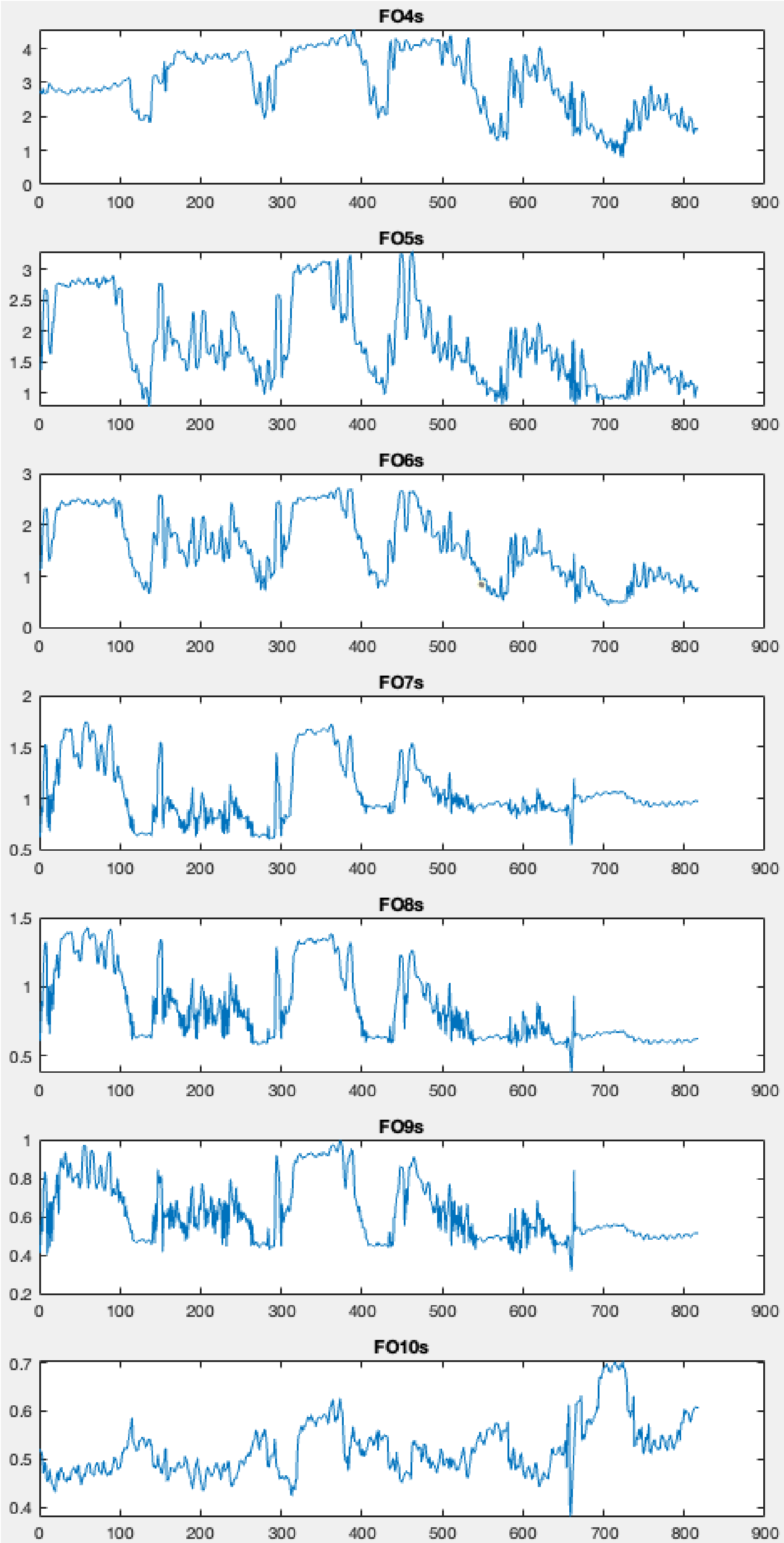

Figure 8 F_O for week 51 dataset

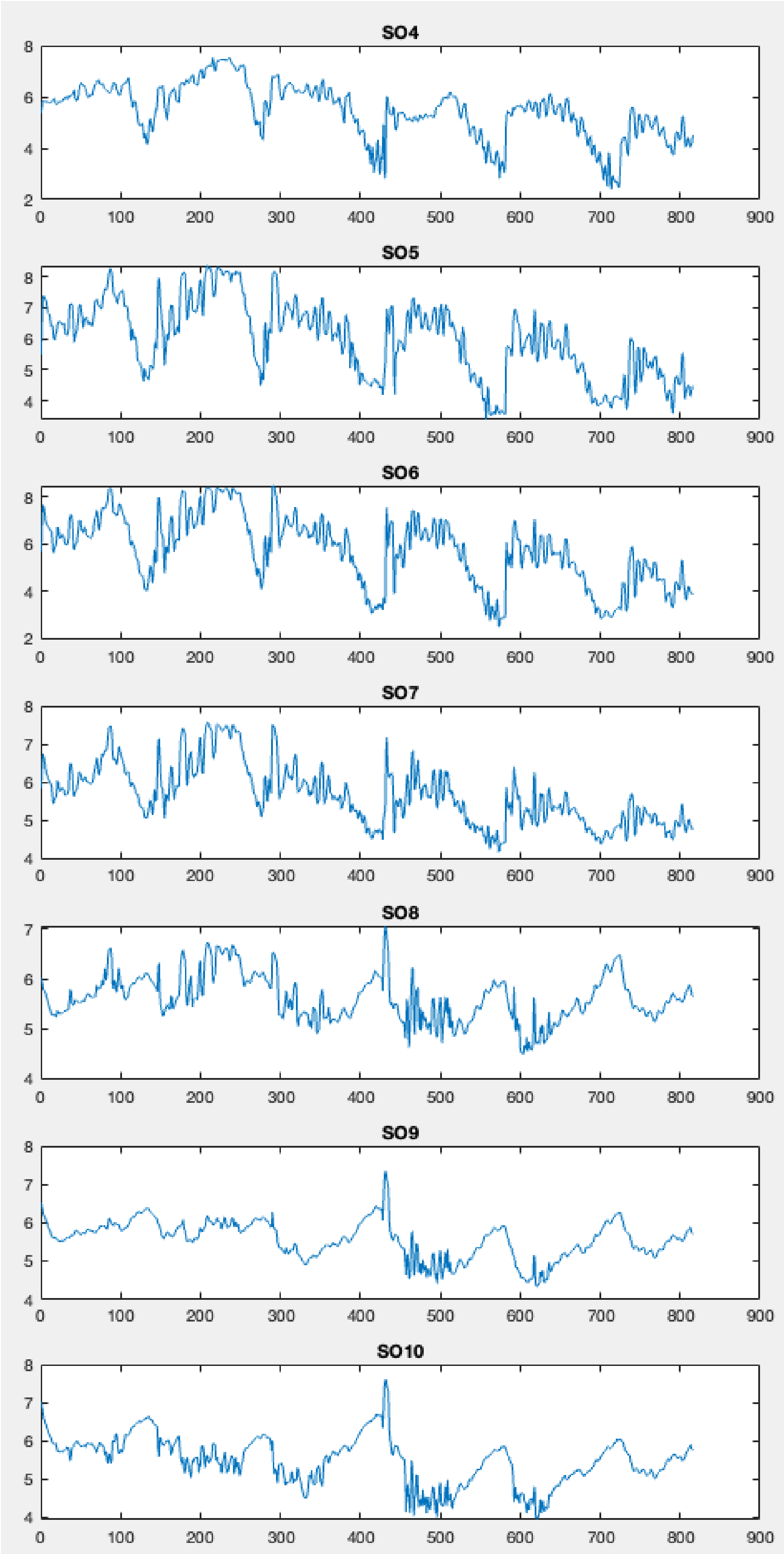

Figure 9 S_O for week 50 dataset

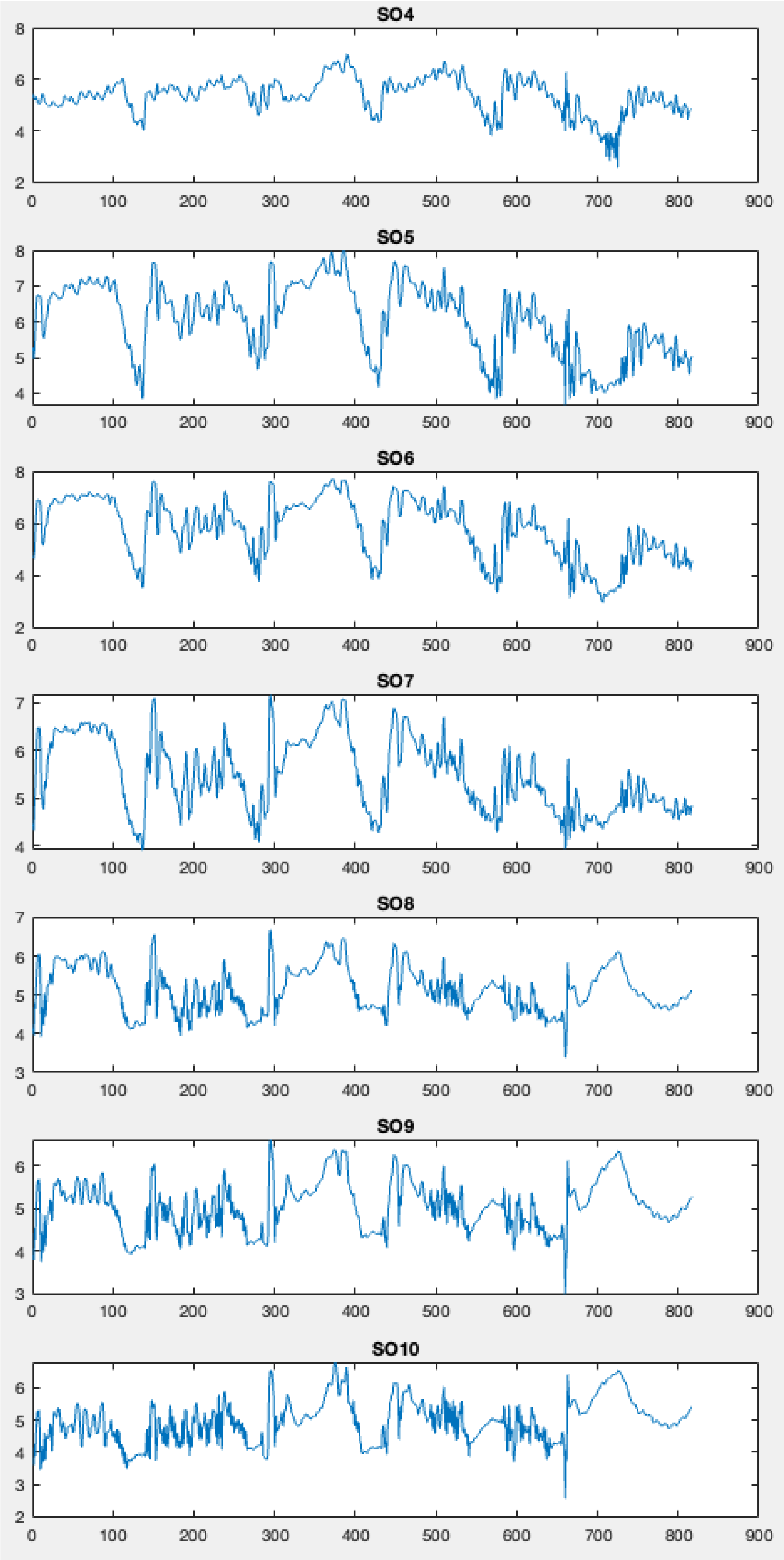

Figure 10 S_O for week 51 dataset

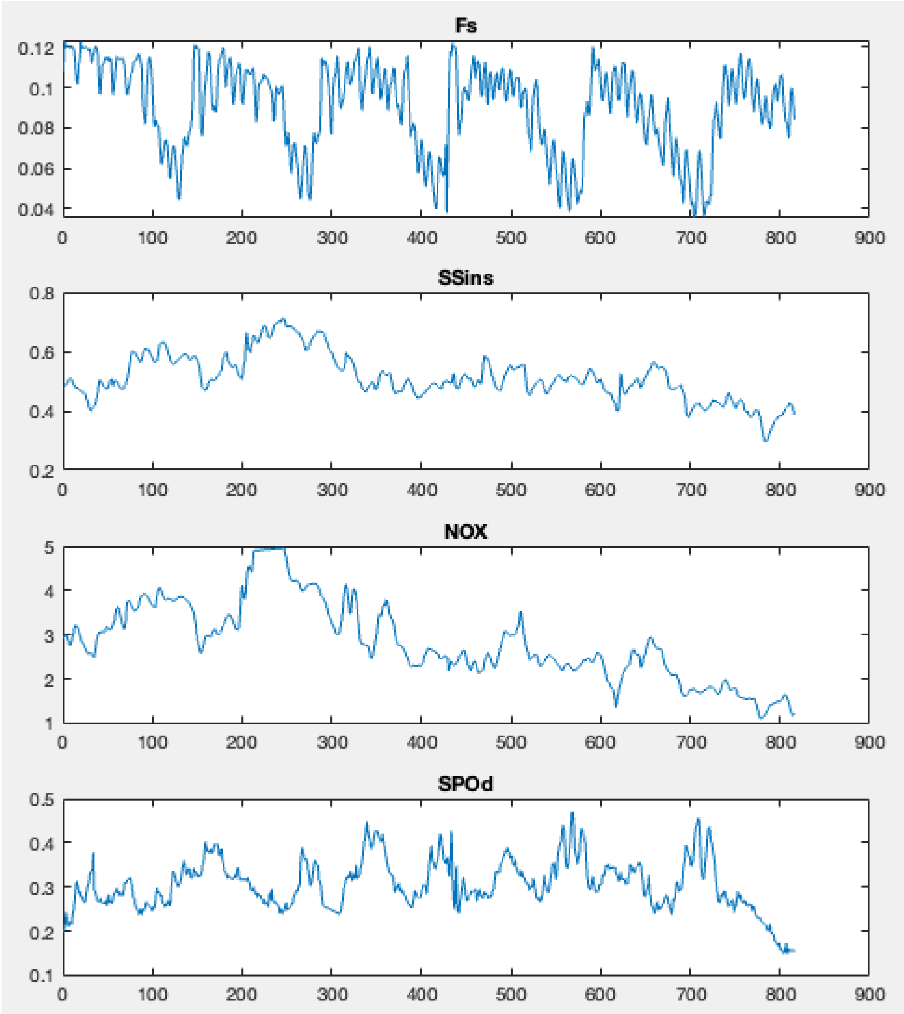

Figure 11 Fs, SSins, NOX and SPOd for week 50

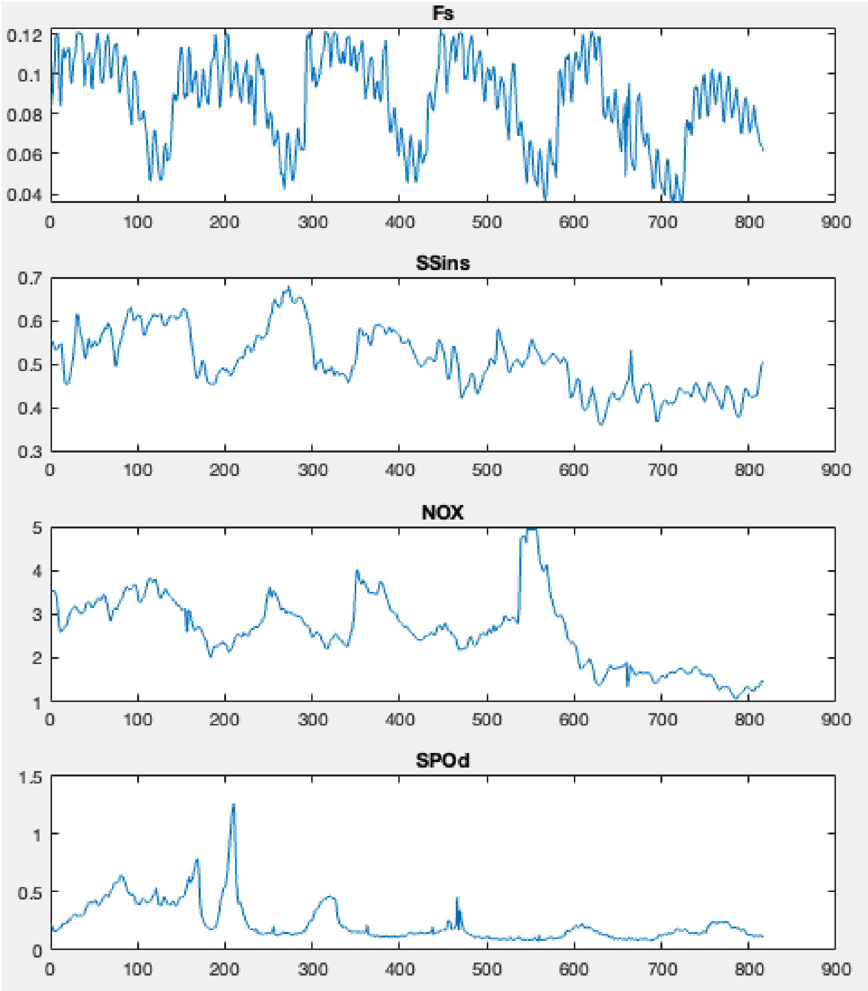

Figure 12 Fs, SSins, NOX and SPOd for week 51

C. All the plots for week 50 dataset after linear interpolating in Matlab

The variables in the m-script had to be in the same time domain as the system identification part. That's why the variable was interpolated again using the "interp1" function in Matlab. This function uses linear interpolation to interpolate. This generated the exact same plots only with the variable being in minutes instead of in every ten minutes (it could have been in hours, days, or other time domains as well). The validation dataset (week 51) was the only dataset used in the m-script for every variable.

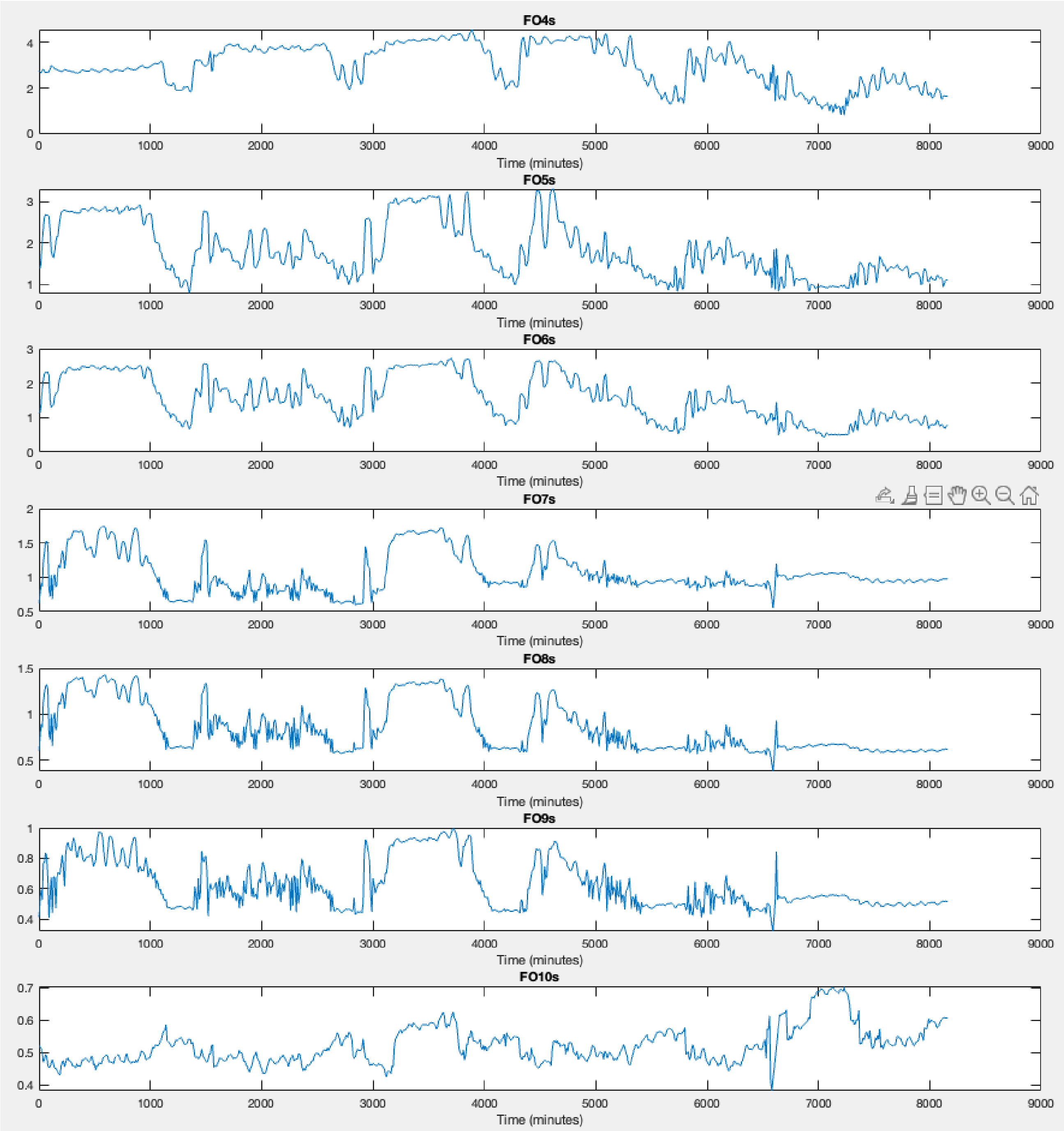

Figure 13 F_O variables for week 51

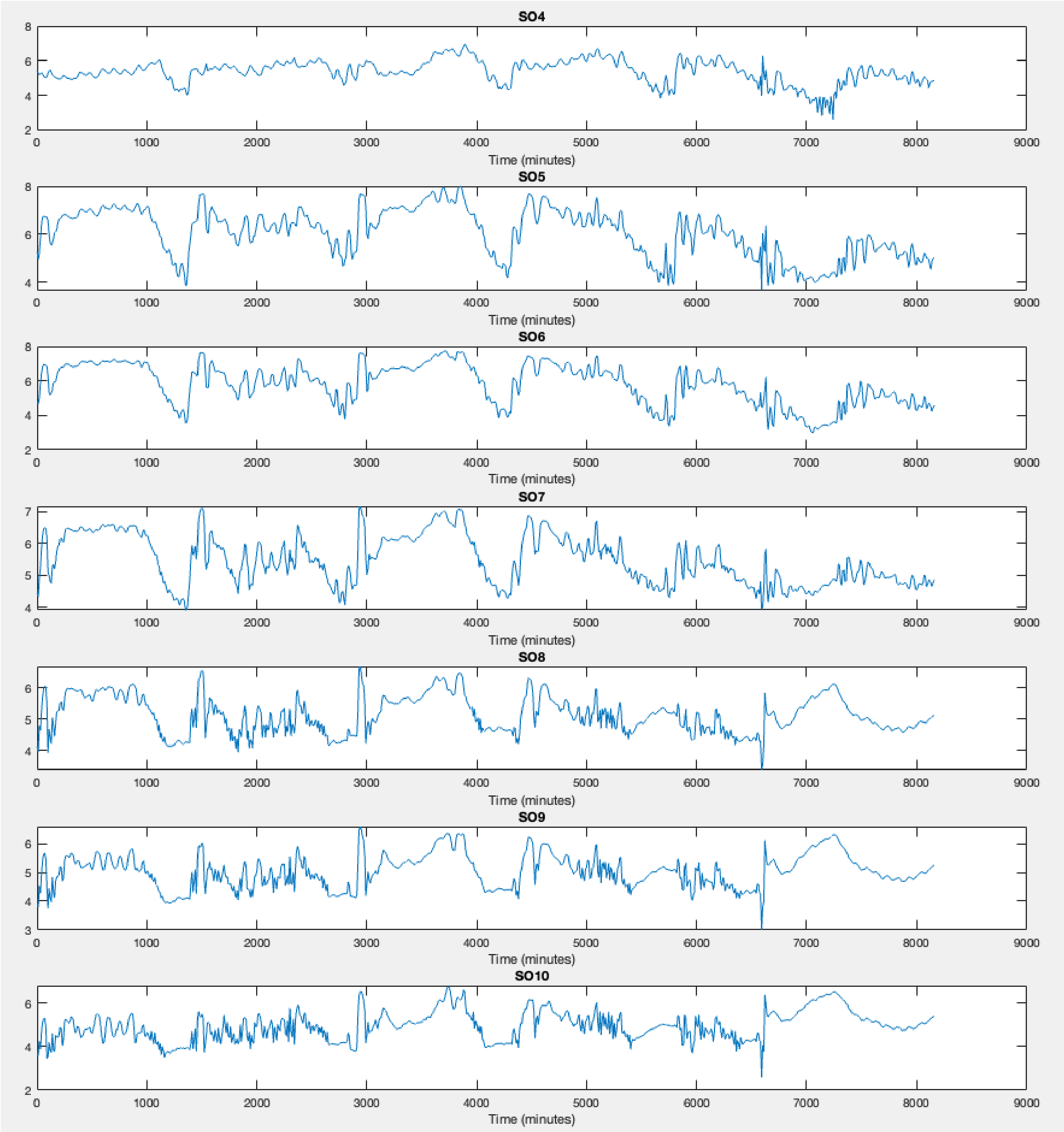

Figure 14 S_O variables for week 51

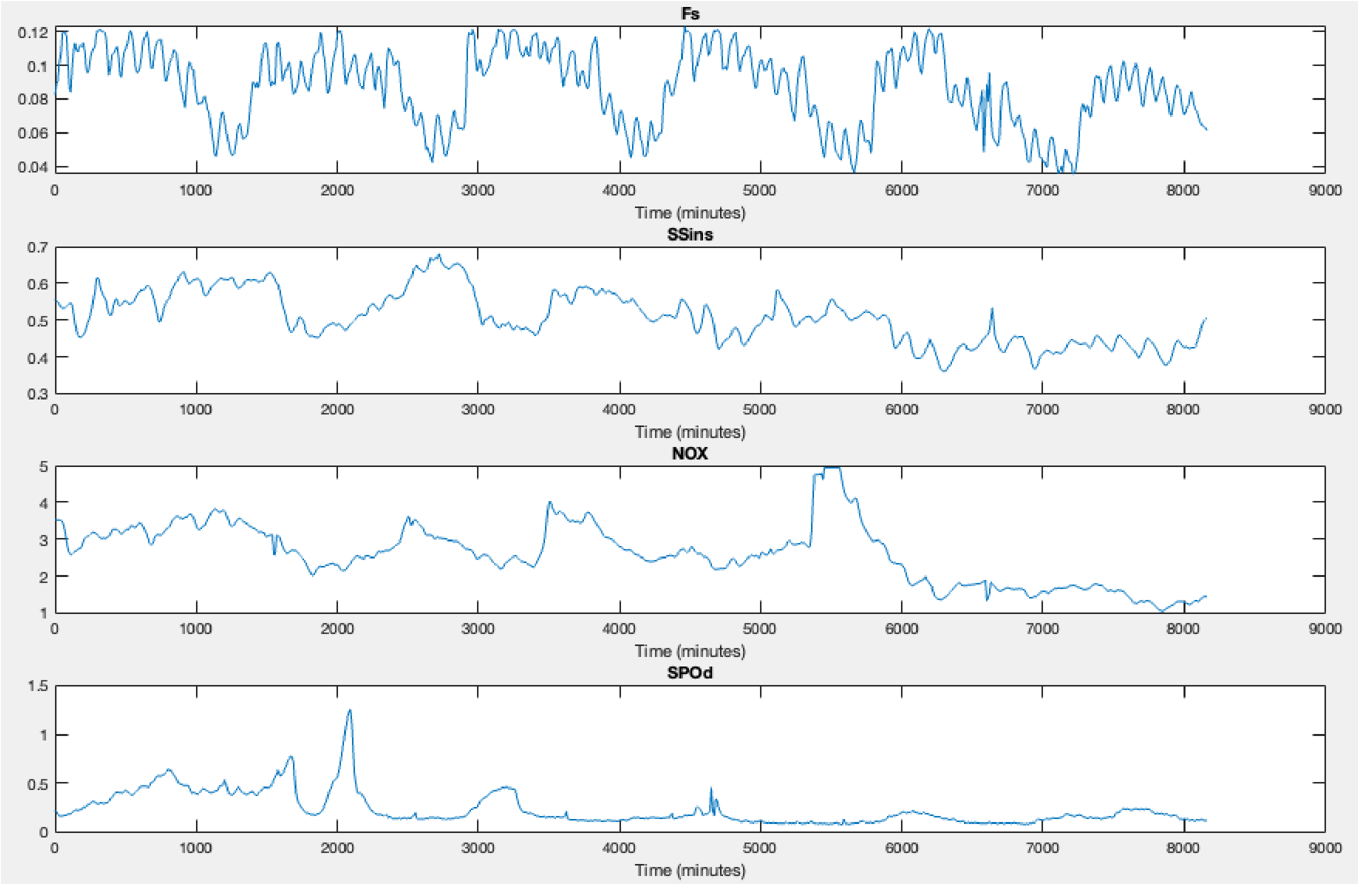

Figure 15 The rest of the variables for week 51

D. Correlation matrix and pair plot

In Figure 16 the correlation matrix for all the variables is being presented. The best correlation between variables is 1, while the worst is -1. Achieving correlation values close to one is the most optimal result. The correlation matrix does not show the diagonal or the variables above the diagonal. The reason for this is because the diagonal is just one because the correlation between the same variables is one which is unnecessary to show. The matrix doesn't show the variables above the diagonal either. The reason for this is because it is the exact same correlations as the ones under the diagonal. By analyzing the correlation matrix, the correlation is the highest for variables that are close to each other. For example, correlation between SO6s and SO5s is at 0.98 which is almost one. The reason for this is because dissolved oxygen (S_O) values in basin six should be close to basin five since it's the closest basin. The aeration rates (F_O) have close relationships with the dissolved oxygen (S_O) in the correlation matrix since applying aeration will have the biggest effect on measurements of dissolve oxygen. That's why it would be a good idea to use close S_O and aeration (F_O) variables to predict other S_O variables in the system identification part. The original plan was to use F_O to S_O transfer functions. However, it was discovered too late that transfer function between S_O and polyphosphate (S_{POd}) was also needed for the control strategies to work. The correlation between S_{POd} and the F_O variables have a high correlation. S_{ins}, F_S and NOX has an adequate correlation with S_{POd}. These variables would be reasonable to use as inputs for estimating phosphate. Almost all the correlations with FO10s are negative. The reason for this is because the conveyer belt that is placed in B10 causes disruptions to the process.

In Figure 17 the pair plot is illustrated between all the variables in the week 51 dataset. The histograms on the diagonal shows that the correlation is one. This is because the same variable will meet on the diagonal. The pair plots above the diagonal are the exact same as the ones under the diagonal. Pair plots where the plots are clustered in clear linear shapes shows that the correlation is very high. The pair plots where the plot is very scattered with no clustering, presents poor correlation between the two variables in question. Investigating plots close to the diagonal illustrates the same findings as the correlation matrix.

That the correlation is high between close by variables. Pair plots is in many ways a great tool to illustrate the correlation matrix visually. There are definitively some outliers between S_{POd} and other variables such as F_O variables and S_{ins}, F_S and NOX variables. However, they're for the most part clustered with somewhat of a linear shape in some of the pair plots. This reflects the correlation found in the correlation matrix.

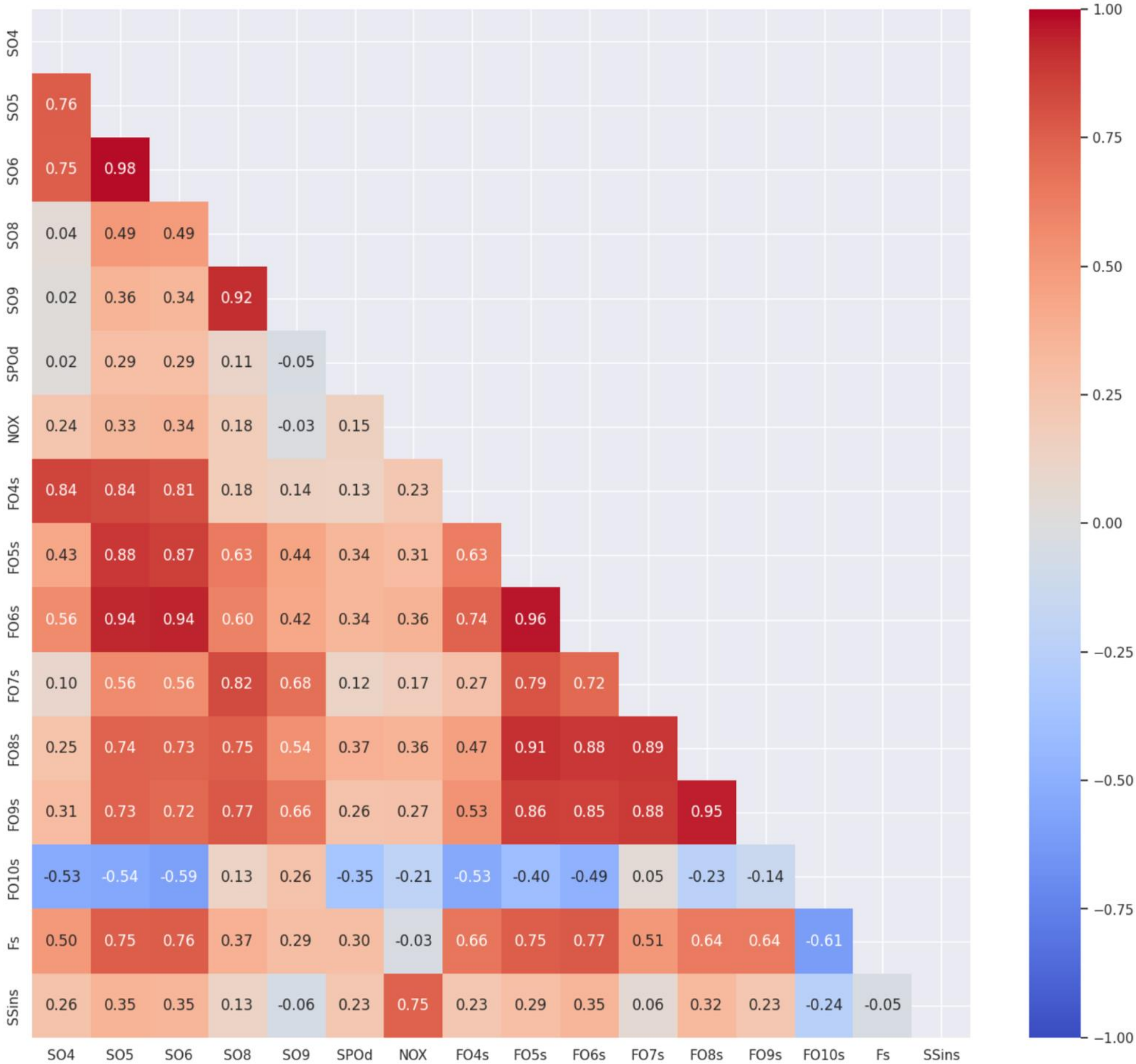

Figure 16 Correlation matrix of all the variables for week 51

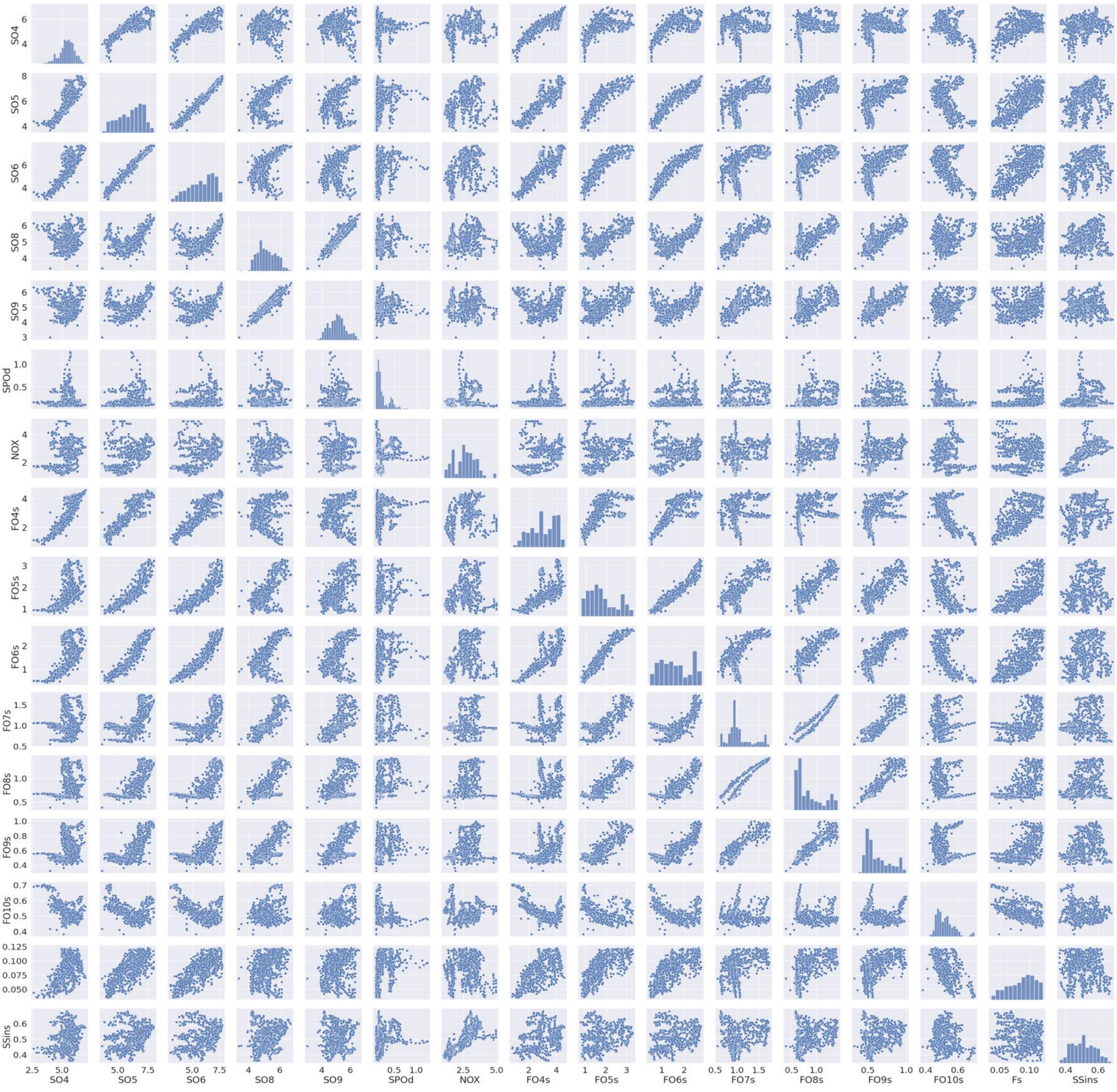

Figure 17 Pair plot of all the variables used, the axis being dependent on which two variables that's being presented

E. System identification results

The system identification results for dissolved oxygen (S_O) can be seen in the appendix section IX.D. Table 6, Table 7, and Table 8 follows the eq. (2) format in section III.E for the transfer functions in the table. The first K_p (K_{p1}) value is for the first input which is in this case soluble chemical oxygen demand in the inlet (S_{Sins}) which is the first input mentioned in the table description, the next being flow rate of wastewater (F_S) for K_{p2} and so on. All the time constants (T_{p1}, T_{p2} and T_d) for the first variable (S_{Sins}) is under the T_{p1} section, the same goes for T_{p2} section (T_{p1}, T_{p2} and T_d) for the second input or variable (which is flowrate of wastewater, F_S) and so on. This can be somewhat confusing but was the best solution that was made. The order of every input is also given in Table 4 section III.E, which is the order for every TF table in this master's thesis, which is for the most part in the appendix section IX.D.

Changing the parameters for many inputs while trying to obtain good enough results was very challenging. The results achieved for model P2 seen in Table 6 showcases that it is possible to obtain promising fit to estimation- and validation data, final prediction error (FPE), mean squared error (MSE) results with second order systems. The reason why second order models achieve better results than other models for tf_s51sfnfspo is because the system is very complex, and a second order function will fluctuate more than a lower order function. However, the parameters that gets generated for the P2 model is not representative of Hias process.

Table 6 tf_s51sfnfspo transfer functions for the P2 model, with S_{Sins}, F_S, NOX, FO4s, FO5s, FO6s, FO7s, FO8s, FO9s, FO10s as input and S_{POd} as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Process Gain Kp5, mg/L	Time constant Tp5, min	Process Gain Kp6, mg/L	Time constant Tp6, min
P2	880.52 +/- $6.1443 \cdot 10^7$	$3.0154 \cdot 10^5$ +/- $2.1056 \cdot 10^{10}$ and 190.84 +/- $1.2081 \cdot 10^4$	-4.9684 +/- 462.09	18.223 +/- 538.59 and 767.99 +/- $7.3503 \cdot 10^4$	-0.13269 +/- 4.5832	473.04 +/- 17324 and 11.8 +/- 478.98	1.3065 +/- 3477.2	88.878 +/- 4411.1 and $1 \cdot 10^4$ +/- $2.6788 \cdot 10^7$	0.79981 +/- $4.6246 \cdot 10^8$	$1.7106 \cdot 10^6$ +/- $9.8931 \cdot 10^{14}$ and 371.68 +/- $5.1668 \cdot 10^7$	1.704 +/- $1.2172 \cdot 10^9$	$1.1932 \cdot 10^4$ +/- $1.3415 \cdot 10^{12}$ and $8.8824 \cdot 10^4$ +/- $7.3432 \cdot 10^{13}$
	Process Gain Kp7, mg/L	Time constant Tp7, min	Process Gain Kp8, mg/L	Time constant Tp8, min	Process Gain Kp9, mg/L	Time constant Tp9, min	Process Gain Kp10, mg/L	Time constant Tp10, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P2	$6.3353 \cdot 10^{-2}$ +/- 11.922	242.48 +/- $4.9773 \cdot 10^4$ and 13.06 +/- 1953.8	$7.0445 \cdot 10^{-2}$ +/- 29.669	294.77 +/- $1.2631 \cdot 10^5$ and 5.877 +/- 2704.1	-99.591 +/- $4.2053 \cdot 10^5$	2.8804 +/- 246.29 and $2.9505 \cdot 10^4$ +/- $1.2456 \cdot 10^8$	-0.21547 +/- 184.06	844.49 +/- $7.3544 \cdot 10^5$ and $1.4417 \cdot 10^{-2}$ +/- 3950.2	$1.567 \cdot 10^{-3}$	$1.386 \cdot 10^{-3}$	34.61	32.54

Table 7 showcases that adding a delay to the P2 model still achieves prominent results. Adding a delay to the system would be beneficial. Since this will simulate the process better. However, this model has no constraint or criterion that would provide the most representative parameters for the system.

Table 7 tf_s51sfnfspo transfer functions for the P2D model, with S_{Sins}, F_S, NOX, FO4s, FO5s, FO6s, FO7s, FO8s, FO9s, FO10s as input and S_{POd} as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Process Gain Kp5, mg/L	Time constant Tp5, min	Process Gain Kp6, mg/L	Time constant Tp6, min
P2D	-0.69341 +/- 1976.2	2.1762 +/- $1.6152 \cdot 10^4$ and $1.0852 \cdot 10^4$ +/- $3.1022 \cdot 10^7$ and Td = 352.58 +/- $1.5870 \cdot 10^4$	$5.2236 \cdot 10^{-3}$ +/- 2.9565	142.48 +/- $8.7993 \cdot 10^4$ and 0.65879 +/- $4.0586 \cdot 10^7$ and Td = 113.04 +/- $4.077 \cdot 10^7$	0.10953 +/- $2.3696 \cdot 10^6$	$4.6536 \cdot 10^4$ +/- $1.1636 \cdot 10^{12}$ and 6394.8 +/- $2.1573 \cdot 10^{10}$ and Td = 324.36 +/- $5.6415 \cdot 10^5$	1.1007 +/- $1.1612 \cdot 10^6$	$6.5756 \cdot 10^5$ +/- $6.937 \cdot 10^{11}$ and 40.185 +/- $2.7947 \cdot 10^4$ and Td = 36.87 +/- $1.6 \cdot 10^4$	3.5075 +/- $3.6715 \cdot 10^6$	10833 +/- $1.134 \cdot 10^{10}$ and 0.22528 +/- $8.0511 \cdot 10^{11}$ and Td = 90.79 +/- $8.0513 \cdot 10^{11}$	-5.9134 +/- 6371.8	14.447 +/- 197.03 and $2.4765 \cdot 10^4$ +/- $2.673 \cdot 10^7$ and Td = 16.59 +/- 114.26
	Process Gain Kp7, mg/L	Time constant Tp7, min	Process Gain Kp8, mg/L	Time constant Tp8, min	Process Gain Kp9, mg/L	Time constant Tp9, min	Process Gain Kp10, mg/L	Time constant Tp10, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P2D	0.33891 +/- 67.901	5.8822 +/- 643.5 and 1550.4 +/- $3.1114 \cdot 10^5$ and Td = 132.41 +/- 470.1	-2.07 +/- 991.46	15.739 +/- 441.17 and 6301.7 +/- $3.008 \cdot 10^6$ and Td = 136.68 +/- 518.36	5.3444 +/- $1.0491 \cdot 10^5$	339 +/- $1.0087 \cdot 10^5$ and $2.9511 \cdot 10^4$ +/- $5.8319 \cdot 10^8$ and Td = 46.33 +/- 7884	-1.4738 +/- $5.5640 \cdot 10^4$	112.64 +/- $2.4175 \cdot 10^4$ and $3.5294 \cdot 10^4$ +/- $1.3337 \cdot 10^9$ and Td = 311.85 +/- 6311.4	$2.168 \cdot 10^{-3}$	$1.871 \cdot 10^{-3}$	24.01	31.31

The P2DL model shown in Table 8 was an attempt on getting the criteria as close as possible to be as representative to the Hias process as possible. The values in this table are generally smaller since it is derived by eq. (2) instead of eq. (3) from section III.E. The values for FO7s and FO8s is also presented in this table. The FPE and MSE are very low which indicates that the model is very accurate at representing the actual data. However, the fit to estimation- and validation data are also low. These values should have been higher. The model seems to be overfitting the data. However, this is the transfer functions that will be utilized for the rest of the project with some adjustments.

Table 8 tf_s51sfnfspo transfer functions for the P2DL model, with S_{Sins}, F_S, NOX, FO4s, FO5s, FO6s, FO7s, FO8s, FO9s, FO10s as input and S_{POd} as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Process Gain Kp5, mg/L	Time constant Tp5, min	Process Gain Kp6, mg/L	Time constant Tp6, min
<u>P2DL</u>	0.43109 +/- 51.65	60 +/- 7026.4 and 2.3332 +/- 799.03 and Td = 359.72 +/- 838.8	1.2924 +/- 42.05	30 +/- 730.98 and 9.2551 +/- 584.05 and Td = 79.46 +/- 287.77	$2.3115 \cdot 10^{-3}$ +/- $8.7999 \cdot 10^{4}$	46.596 +/- $1.6008 \cdot 10^{9}$ and $2.6751 \cdot 10^{-3}$ +/- $2.0291 \cdot 10^{12}$ and Td = 147.61 +/- $2.0292 \cdot 10^{12}$	$-2.9016 \cdot 10^{-3}$ +/- 0.14928	18.919 +/- $1.8742 \cdot 10^{4}$ and 24.979 +/- $1.9438 \cdot 10^{4}$ and Td = 47.75 +/- 2213.6	$-2.9011 \cdot 10^{-2}$ +/- 0.7829	23.394 +/- 6485.3 and 28.691 +/- 8129.6 and Td = 81.34 +/- 415.09	$-4.7375 \cdot 10^{-2}$ +/- 1.0079	49.545 +/- 1137.8 and 6.1161 +/- 429.72 and Td = 57.45 +/- 251.73
	Process Gain Kp7, mg/L	Time constant Tp7, min	Process Gain Kp8, mg/L	Time constant Tp8, min	Process Gain Kp9, mg/L	Time constant Tp9, min	Process Gain Kp10, mg/L	Time constant Tp10, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
<u>P2DL</u>	0.13821 +/- 45.776	60 +/- $2.39 \cdot 10^{4}$ and 30 +/- 2495.5 and Td = 132.74 +/- 392.31	0.12275 +/- 15.364	40.808 +/- $2.2766 \cdot 10^{4}$ and 30 +/- $1.3772 \cdot 10^{4}$ and Td = 118.58 +/- 446.06	$-5.3278 \cdot 10^{-2}$ +/- 1.4885	60 +/- 3913.1 and 13.34 +/- 2525.6 and Td = 304.8 +/- 1425.3	-0.21394 +/- 9.2486	60 +/- 2573.7 and 1.3255 +/- 1620.2 and Td = 315.41 +/- 1503.4	$3.494 \cdot 10^{-3}$	$3.015 \cdot 10^{-3}$	3.543	-2.063

1) Fitness index and plot of tf_s51sfnfspo model outputs

The fitness index and plot each of the model outputs is illustrated in Figure 18. This describes the differences of the models to each other.

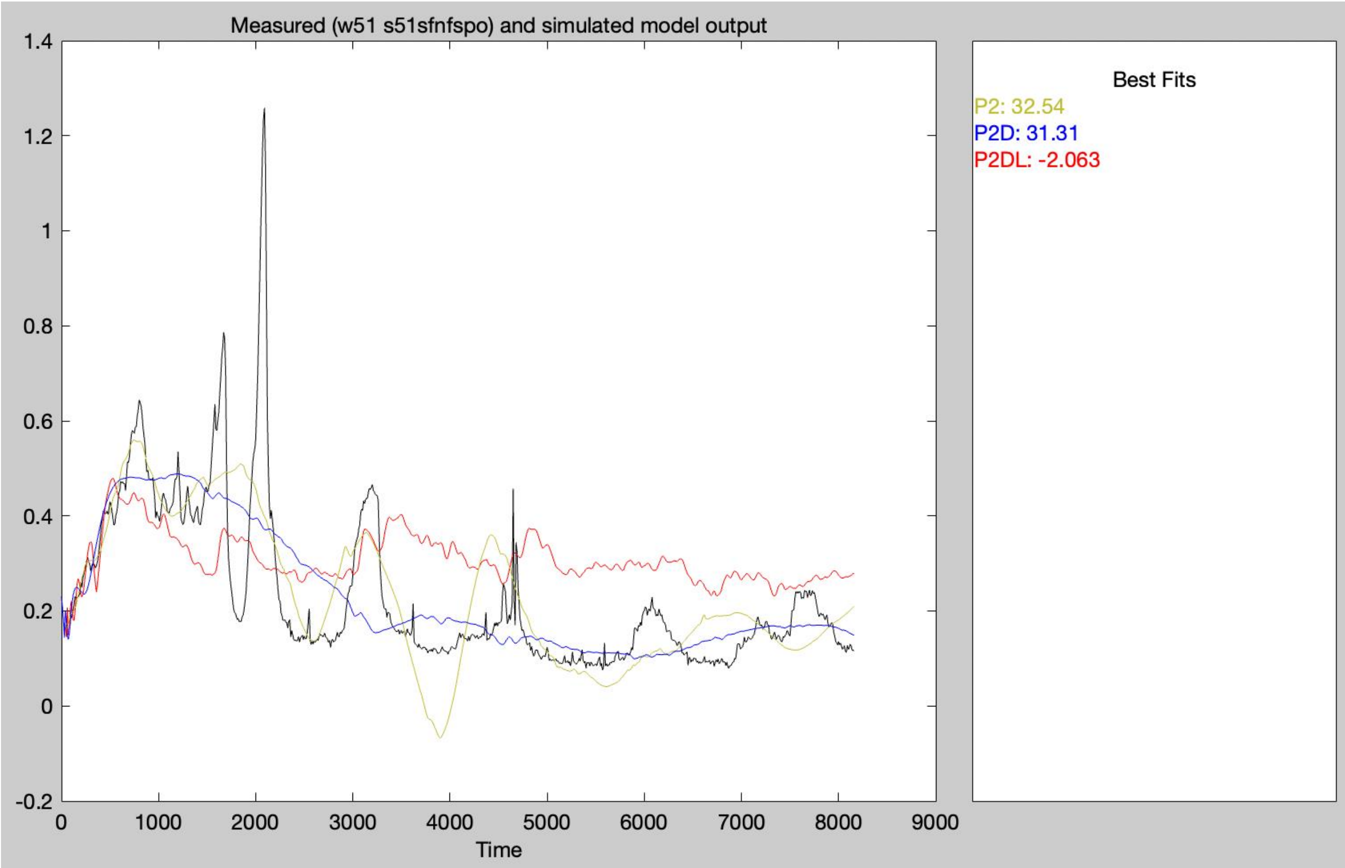

Figure 18 s51sfnfspo fitness index and plot of the best model outputs with S_{Sins}, F_S, NOX, FO4s, FO5s, FO6s, FO7s, FO8s, FO9s, FO10s as input and S_{POd} as output, and week 51 as validating data and week 50 as estimation data. The x-axis being in minutes

F. Simulated dynamic model of S_{POd}

The original dynamic linearized model is shown in Figure 19. This shows the same output as the system identification model. However, because FO7s_S_{POd} and FO8s_S_{POd} transfer functions (TFs) had positive proportional gains (Kp), it made the system unstable. Which is why they were replaced. They were replaced with the TFs between FO8s_S_{POd}. The dynamic linear model used for this project is seen in Figure 20. This will give completely different results for the system identification part. It most likely would give worse system identification results. This is unknown, and there could be ways to improve the results as well. However, there are still some similarities between the original linear model and the linear model used for the control part. The original linear model has an absolute integral index (IAE) of 919.9, while the linear model used for this project has a 1152, between the linear model and the measurement of polyphosphate (S_{POd}). This can be seen in Table 9. The IAE for the linear model indicates that there are some large differences. However, by visual inspection of Figure 20, the linear model does an adequate job of representing the measurement S_{POd}. It does follow some dynamic trends and are not too far off from the measurement of S_{POd}.

All the deviation variables used are completely the same as the subplots for week 51 in IV.B section. Except that the subplots show the variables for every ten minutes which is the sampling time for the dataset. While the deviation variables, the system identification models, all outputs for the control part shows the data for every minute, this is seen in the IV.C section. The purpose of system identification is to get the parameters that represents the system well enough based on the datasets provided. While getting adequate fit to validation data results would be preferable, it is not the most important part.

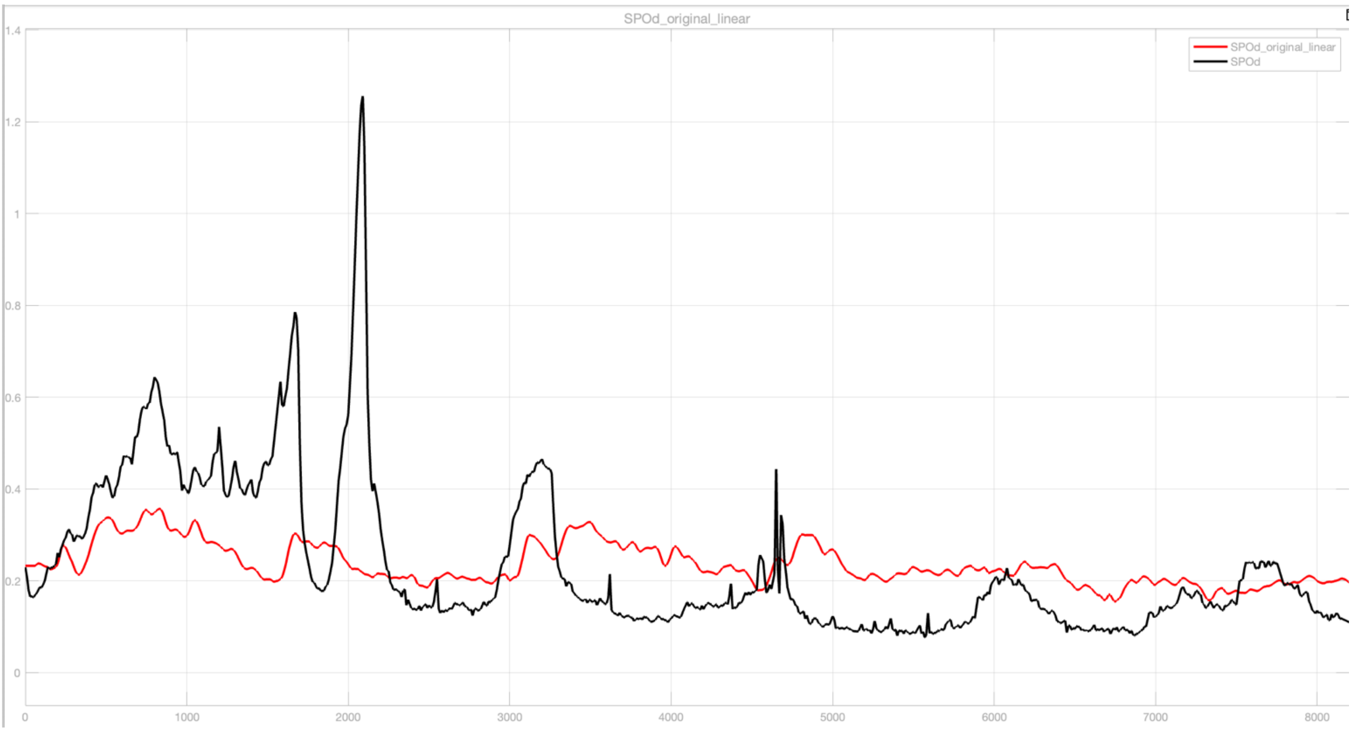

Figure 19 Original dynamic linear model of S_{POd}, y-axis being mg P/L, and x-axis being in minutes

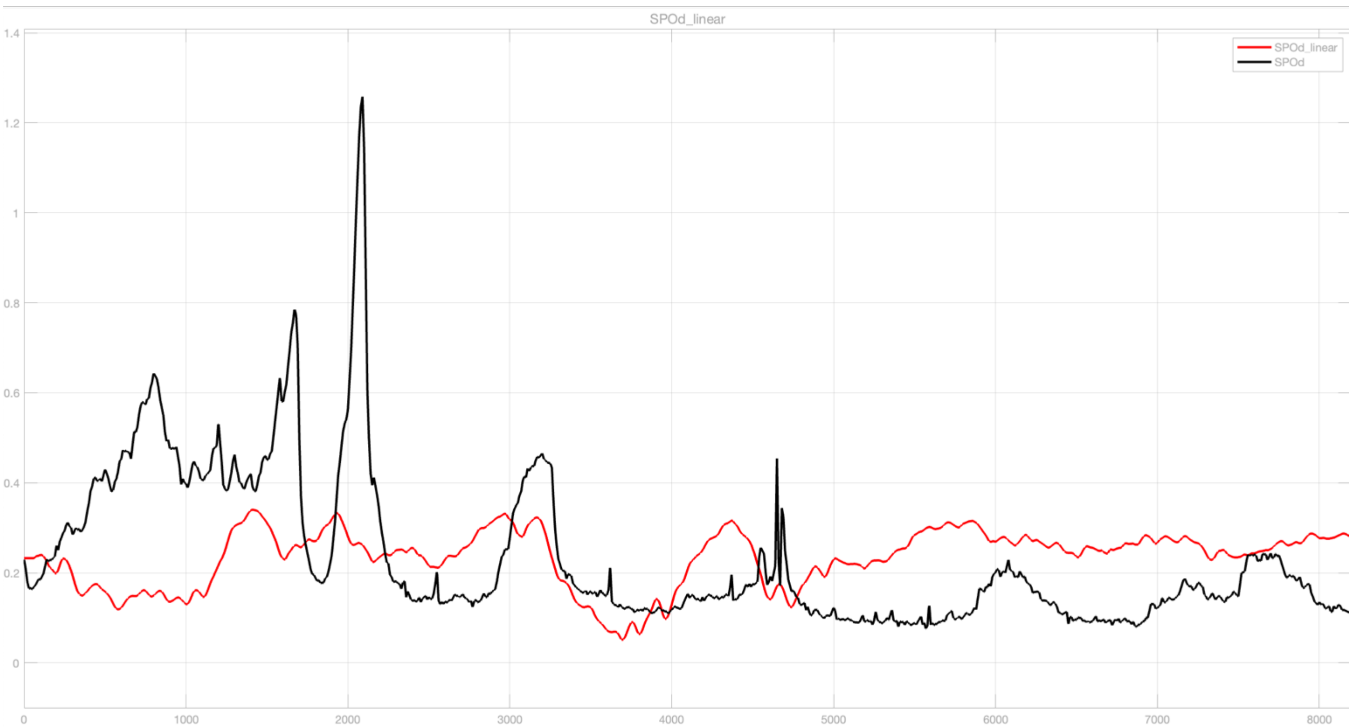

Figure 20 Dynamic linear model of S_{POd}, y-axis being mg P/L, and x-axis being in minutes

Table 9 IAE of the original linear model and the linear model error against measurement of S_{POd}

Linear model	***IAE_S_{POd}***
Original linear model	919.9
Linear model	1152
Difference in % ($\frac{linear\ model}{original\ linear\ model} \cdot$ 100%)	125.23%

G. The results for the control startegies

1) Controller tuning

Table 10 presents the tuning parameters for the Proportional-integral-derivative (PID) controller. The skogestad tuning rules are a great way getting initial parameters (Skogestad, 2003). The parameters in this project are based on those rules. This has been explained in section III.H.2). The transfer function (TF) FO5s_S_{POd} is the TF used for the PID. The TF has the time constants T_{p1} (τ_1) = 28.6533 and T_{p2} (τ_1) =23.4192 which can be seen in Table 2 in chapter III.E. This was the initial values used and they achieved adequate result. However, they were tuned based on trial-and-error in Simulink. The values presented in Table 10 gave better results visually and better Integral of absolute error (IAE) and Integral of total movement in manipulated variables (IAMV). The K_p value for FO5s_S_{POd} were -0.02901, this value was not tuned further. The T_d (which is the same as θ and τ_c) value for FO5s_S_{POd} are 81.34 min. Calculating K_c were done by utilizing eq. (6) from section III.H.2). Calculating τ_i can be done by utilizing eq. (7) in the same section. τ_d parameter is based on T_{p2} for FO5s_S_{POd}. This was originally 23.4192. However, τ_d = 30.5810 min achieved better results. Table 11 shows the gains used for the ratio controller. The initial parameters for the ratio controllers were based on the mean of the aeration rate (F_O) for each of the controllers in relationship to the aeration rate for the fifth basin (FO5). However, they were tuned further to achieve better IAE and IAMV results.

Limits has been set to 0.3 as an upper limit and -0.3 as a lower limit for all the manipulated variables (aeration rate, F_O). The reason for the upper limit having the same value as the lower limit just as a negative one instead is because it's deviation variables and it should fluctuate around zero. These can be set in the settings of the PID. This was done for the model predictive controller (MPC) as well. However, the limits are being set in the m-script instead and can be seen in Table 12.

The parameters for the model predictive controller (MPC) are shown in

Table 12. The equations used to get initial values for the parameters is seen in section III.I.2). The process time constant in equation (1) section III.D is 411.8774 min. However, the process time constant was chosen to be 600 min instead due to prior knowledge about the system. The sampling time ($T_{settling}$) is one tenth of the process time constant, which is 60. The settling time ($T_{settling}$) should have been 2800 min according to Seborgs, this was too high of a settling time. 600 min achieved the best results. The model horizon N is just the $T_{settling}$ divided by $T_{settling}$, which is 10. The control horizon M should have been between $\frac{10}{3}$ and $\frac{10}{2}$. However, this was chosen to be 10 instead. The weighting matrix Q was set to 10. Many different values were tested for the weighting matrix R. However, the best weight was the initial value. All the tuning was done to achieve better IAE and IAMV results and get the best possible visual representation of the system.

Table 10 PID controller parameters after tuning

CV-MV	Controller	τ_c	K_c	τ_i	τ_d
S_{PO}- FO5	PID	81.34	-8.5098	40.1606	30.5810

Table 11 Ratio controller gains for each of the MVs

CV-MV	Ratio Controller	Gain
S_{PO}- FO4	rFO4	1.8
S_{PO}- FO6	rFO6	0.72418
S_{PO}- FO7	rFO7	0.4519
S_{PO}- FO8	rFO8	0.38225
S_{PO}- FO9	rFO9	0.2081
S_{PO}- FO10	rFO10	0.10

Table 12 MPC parameters after tuning

		MPC controller
$T_{sampling}$	Sampling time	60
$T_{settling}$	Settling time	600
N	Model horizon	10
M	Control horizon	10
P	Prediction horizon	610
Q	Weighting CVs	10
R	Weighting MVs	[0.1 0.1 0.1 0.1 0.1 0.1 0.1]
MV	Constraints MVs [FO4 FO5 FO6 FO7 FO8 FO9 FO10]	MV1 = (min, -0.3, max, 0.3) MV2 = (min, -0.3, max, 0.3) MV3 = (min, -0.3, max, 0.3) MV4 = (min, -0.3, max, 0.3) MV5 = (min, -0.3, max, 0.3) MV6 = (min, -0.3, max, 0.3) MV7 = (min, -0.3, max, 0.3)
OV	Constraints CV [S_{PO}]	OV1 = (min, -0.3, max, 0.3)

2) Controller testing

In Figure 21, the test procedure trends for control variable polyphosphate (S_{POd}) and for setpoint of S_{POd} using the Proportional-integral-derivative (PID) controller can be observed. The disturbances can be seen in Figure 25, Figure 26 and Figure 27. The results illustrates that the PID are cable of suppressing the disturbances that occurs. It does an excellent job of tracking the setpoint. The amplitude of the measurement of S_{POd} are not very high either by investigating the y-axis of the output. One of the highest peeks at about 1000 minutes is on about 0.24 mg phosphorus/L, while the set point is on 0.2325 mg phosphorus/L.

Figure 22 however, presents the test procedure trends for control variable S_{POd} and setpoint of S_{POd} using the MPC controller instead. The MPC can be visually seen as achieving impressive results. The settling time after each disturbance occurs are remarkable. The overall changes in the output of the MPC and the PID are very similar in the sense of them following the set point and the disturbances occur at the same time which is expected. The MPC has one of its highest peeks at 3 000 min, between 0.235 mg phosphorus/L and 0.24 mg phosphorus/L.

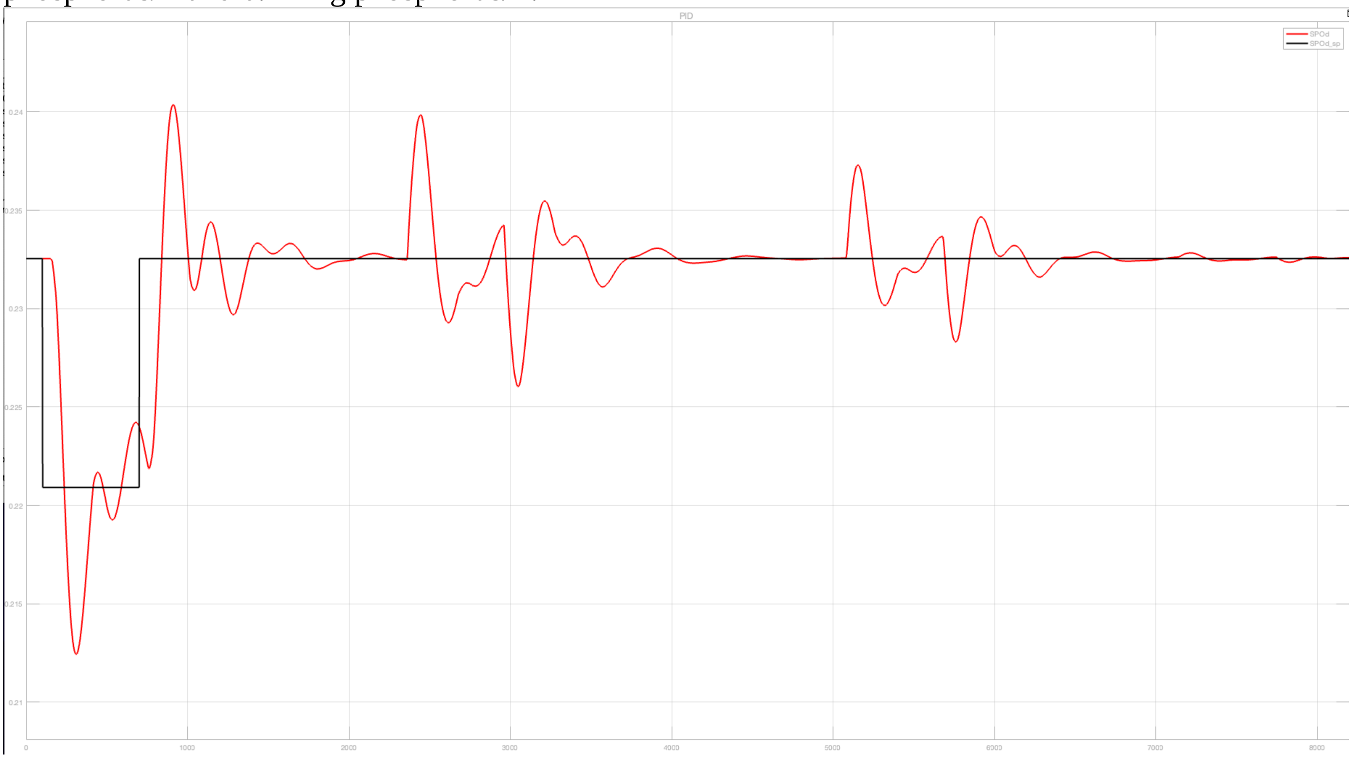

Figure 21 Test procedure trends for control variable S_{POd} (red graph) and setpoint (black graph) using the PID controller, y-axes being mg P/L and x-axis being in minutes

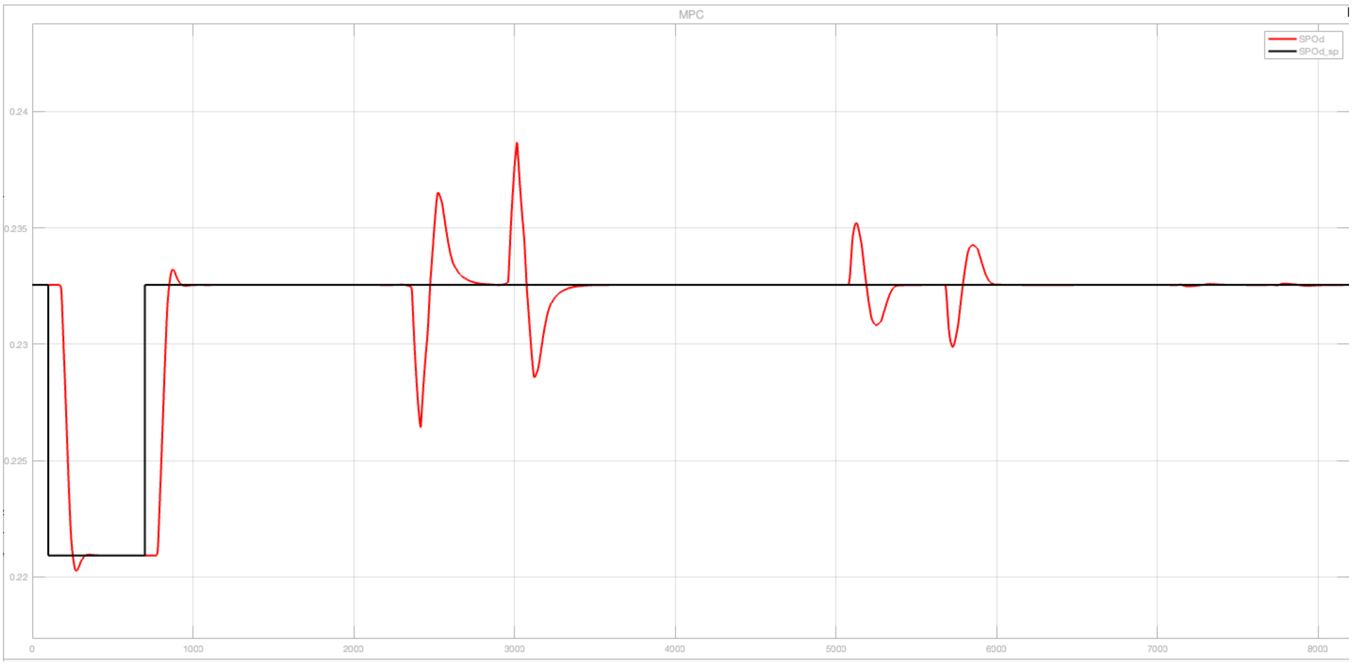

Figure 22 Test procedure trends for control variable S_{POd} (red graph) and setpoint (black graph) using the MPC controller, y-axes being mg P/L and x-axis being in minutes

Figure 23 illustrates the test procedure trends for the aeration rates (F_O), which is the manipulated variables (MVs) for the PID controller. By visual inspection it's possible to see the disturbances and the set point changes which is also seen in Figure 21 and Figure 22. The constraints set as -0.3 to 0.3 is also visible. Except for test procedure for FO4s. The reason for this is because of the gain being higher than for the FO5s. Figure 24 presents the same things as Figure 23 just for the MPC. However, the MVs are much smaller for the MPC than for the PID.

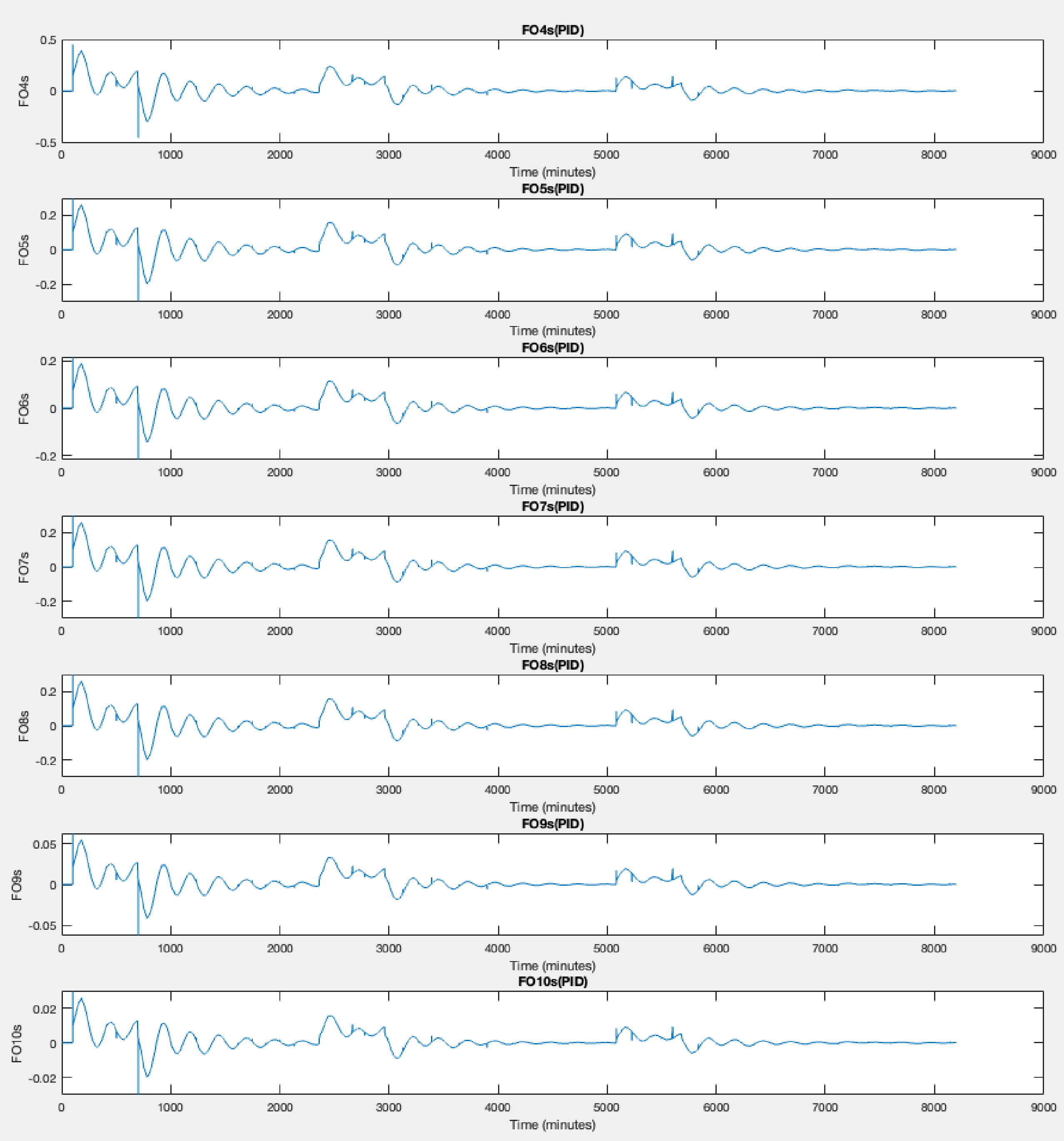

Figure 23 Test procedure trends for MV1 (FO4s), MV2 (FO5s), MV3 (FO6s), MV4 (FO7s), MV5 (FO8s), MV6 (FO9s), MV7 (FO10s) for PID

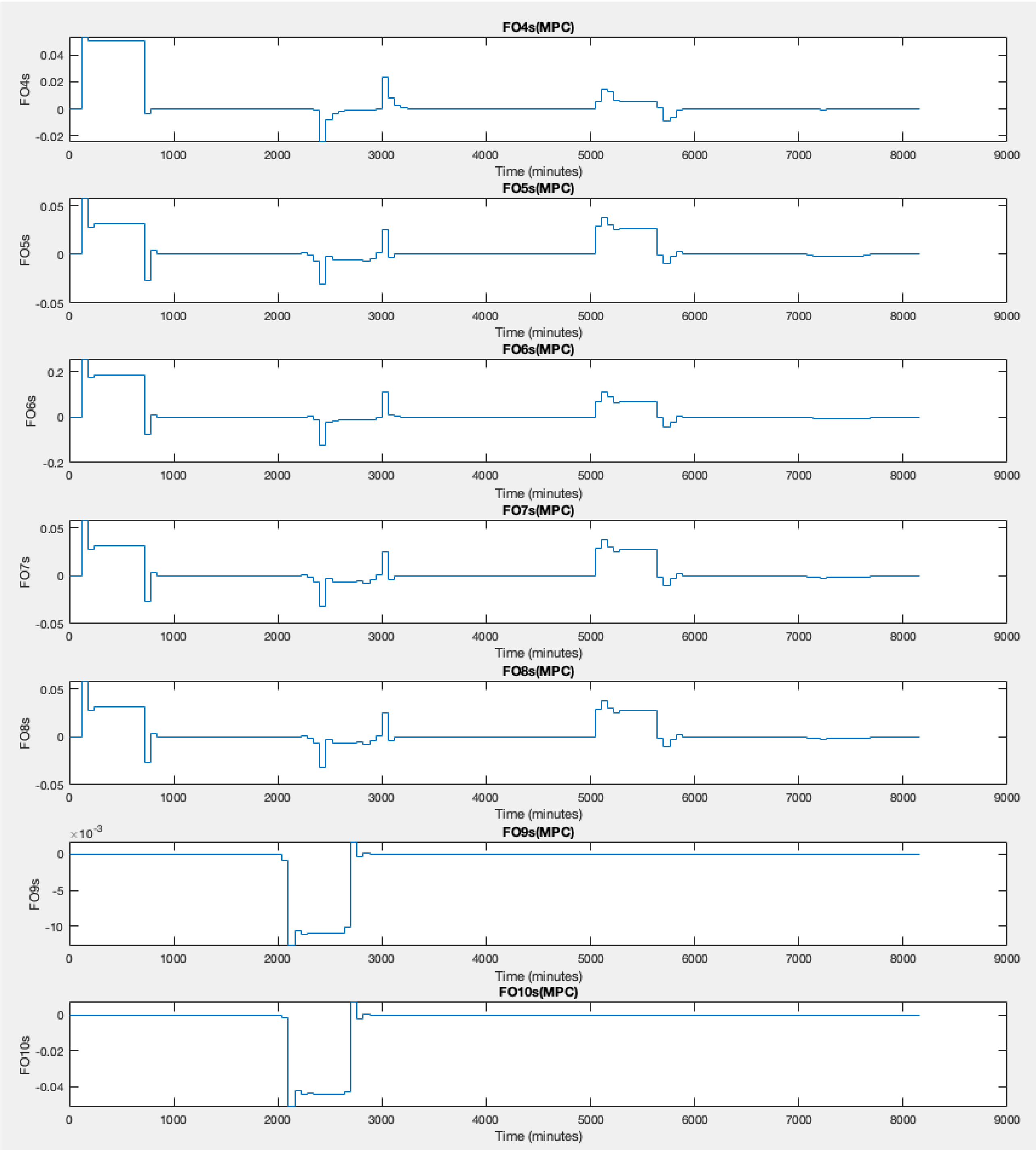

Figure 24 Test procedure trends for MV1 (FO4s), MV2 (FO5s), MV3 (FO6s), MV4 (FO7s), MV5 (FO8s), MV6 (FO9s), MV7 (FO10s) for MPC

The test procedure trends for the disturbance variables (DVs) are based on Table 5 seen in III.K section. The DVs are only five percent of the mean of the DV in question. The reason for this is to provide stability to the system while also introducing small disturbances at varying time instances to see how it effect the system. If all the DVs was introduced to the system at ones the system would be unstable and no analysis of the behavior of the DVs would be possible. The direction of the DVs is based on the hypothesis that when the flow rate of wastewater (F_S) goes up (because of rain for the most part) then soluble chemical oxygen demand (S_{Sins}) will go down, since it will be diluted. NOX will also be diluted and should go down. However, it goes up here instead. The effect of NOX is not significant; therefore, it doesn't matter as much which directing it is. The disturbances are visible in Figure 21 and Figure 22.

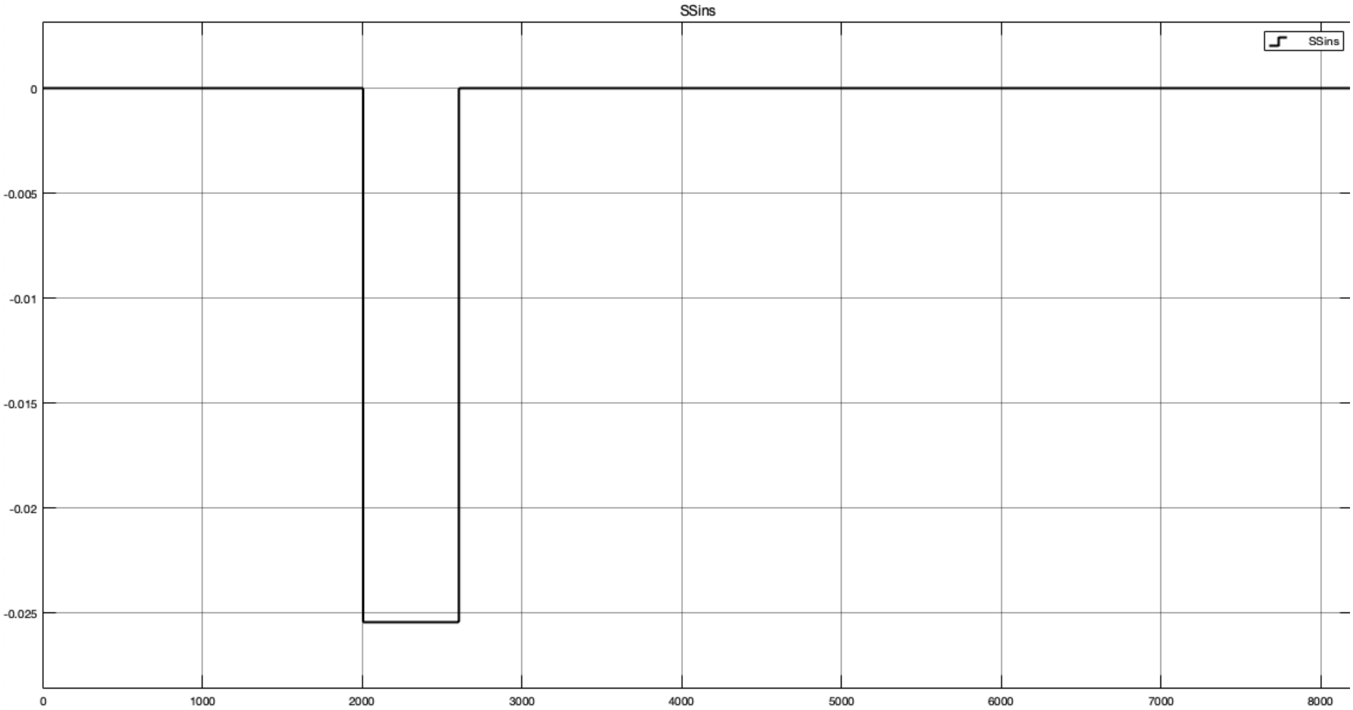

Figure 25 Test procedure trend for DV1 (S_{Sins}), y-axis being mg COD/L and x-axis being in minutes

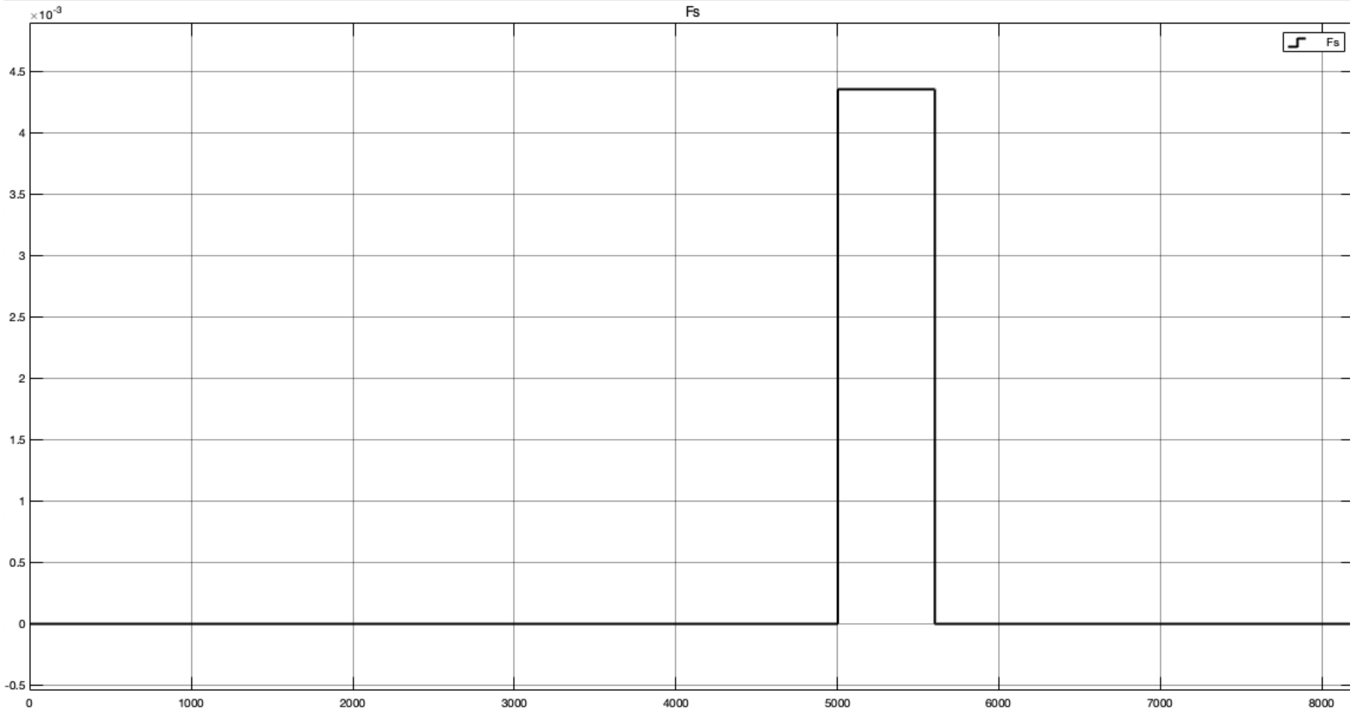

Figure 26 Test procedure trend for DV2 (F_S), y-axis being L/h and x-axis being in minutes

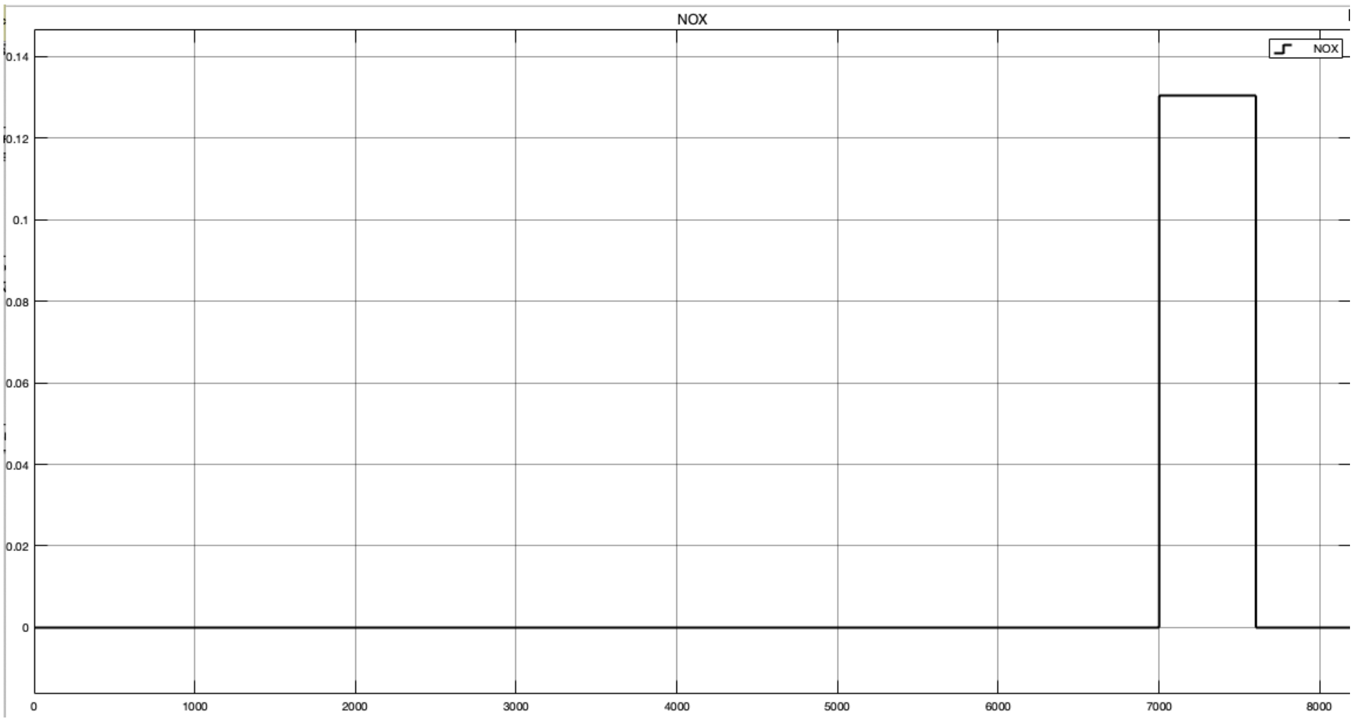

Figure 27 Test procedure trend for DV3 (NOX), y-axis being $\mu g/m^3$and x-axis being in minutes

H. Control error indicies and computational time for control strategies

Table 13 presents the Integral of absolute error (IAE) for the model predictive controller (MPC) and the Proportional-integral-derivative (PID) controller, and the difference between them. The difference being the IAE_$\boldsymbol{S_{POd}}$ for MPC divided by the IAE_$\boldsymbol{S_{POd}}$ for the PID and multiplying with 100%. Having a low IAE indicates that the difference between the set point and the measurement are not very large. The MPC has significantly lower IAE_$\boldsymbol{S_{POd}}$ than the PID which illustrates the MPCs abilities.

Table 14 describes the Integral of total movement in manipulated variables (IAMV), and the difference between them. This shows the changes in the control systems input over time. A low IAMV would indicate that the manipulated variables are able to track the set point. The MPC has a lower IAMV than the PID.

The computational time is presented in Table 15. The PID achieves better computational time. The computational time for the MPC was 4.859 seconds, and 2.782 seconds for the PID.

The cumulative sum of the aeration rate ($\mathrm{F_O}$) described in Table 16 provides an estimate of the total aeration that of the system. The actual value column in this table takes the total sum of the $\mathrm{F_O}$ before the MPC and PID is implemented. One of the reasons why this value is so high might be because of the scales in the system. However, the scales are the same for all of them. The MPC has exceptionally lower cumulative sum for the aeration rate than that of the actual value and the PID.

Table 13 Integral of absolute error (IAE), and the difference between them

Controller parameters	***IAE_ S_{POd}***
PID	10.19
MPC	5.009
ΔIAE_S $_{POd}$ ($\frac{MPC}{PID}$·100%)	46.4

Table 14 Integral of total movement in manipulated variables (IAMV), and the difference between them

Controller parameters	***IAMV_ FO4s***	***IAMV_ FO5s***	***IAMV_ FO6s***	***IAMV_ FO7s***	***IAMV_ FO8s***	***IAMV_ FO9s***	***IAMV_ FO10s***	***Sum_IAMV***
PID	394.1	218.9	158.6	98.94	83.69	45.56	21.89	$1.0217 \cdot 10^3$
MPC	40.62	47.98	194.1	47.98	47.98	6.834	27.32	412.8140
ΔIAMV_ F_O ($\frac{MPC}{PID}$·100%)	10.31	21.92	122.38	48.49	57.33	15	124.81	40.4

Table 15 Computational time for MPC and PID

Controller parameters	***Time (seconds)***
PID	2.782
MPC	4.859

Table 16 Cumulative sum of Fo

	Cumulative sum of F_O(MVs)
Actual value	$6.3413 \cdot 10^{11}$
PID	$7.2625 \cdot 10^5$
MPC	422.6788

V. DISCUSSION

A. Limitations

An industrial dataset will produce more noise and there are more uncertainties. The fitness index will not be as high because of this. A fitness index of 50% is seen as a very good performance in this case.

Temperature was unfortunately not measured with online data. This is an important disturbance variable. Rudi et al. found that the temperature effected phosphorus removal capabilities of the Hias process significantly. Incorporating this would be beneficial.

There was not made online measurements of soluble oxygen demand (S_S) for basin 7 which resulted in there not being made TFs with S_S as output. Making control strategies that control S_S could be explored in future projects.

The control strategies works very well because the model that gets tested has the exact same parameters as the control algorithms. Normally the models get tested against first principle's models with ordinary differential equations that represents the same system. However, development of control strategies is very important work towards industrial implementation.

B. Data preprocessing

The data provided is very noisy with many sudden changes which illustrates the dynamic changes that occurs in a such a process. There could have been added several different data preprocessing techniques. However, capturing the dynamic changes that may occur in real life is important as well. Therefore, balancing between capturing the dynamic changes in the process and achieving better results must be taken into consideration. Achieving even better correlation between the variables would also be preferable.

The raw data is static and will not take the time delay that occurs throughout the process. That's why a time delay is added to the process. This will make the simulations dynamic and will take an important element into consideration in the process.

The correlation matrix illustrates that the correlation between polyphosphate (S_{POd}) and the aeration rate (F_O) variables have a high correlation. The soluble oxygen demand (S_{ins})-, the wastewater flow rate (F_S)- and NO_2/NO_3 (NOX) in the inlet has an adequate correlation with S_{POd}. These variables were reasonable to use as inputs for estimating phosphate. The pair plots described many of the same things as the correlation matrix and was able to visualize the correlation matrix.

C. System identification for the dynamic linear models

The P2DL model shown in Table 8 was an attempt on getting the criteria as close as possible to be as representative to the Hias process as possible. The parameters could have been even closer to the desired parameters. It achieved low final prediction error (FPE) and mean squared error (MSE) which indicates that the model is accurate. The fit to estimation- and validation data was also low. This should have preferably been higher. However, the reason why it is not that high can be several reasons. One of them being that there are too many inputs that predicts the output. However, there have been conducted tests that checks this hypothesis. Fewer inputs did not achieve better results. The scaling of the variables could possibly be different as well. Better data preprocessing with the emphasis on achieving better correlation between the variables is also a reason for low fit to validation data results.

In theory polyphosphate (S_{POd}) should have a tenth order TF since S_{POd} reacts with the bacteria in all the basins. That's why a higher order output model will in theory fit better than lower order equations.

The proportional gain (K_p) values of the transfer function for FO7 (FO7s_S_{POd}) and FO8 (FO8s_S_{POd}) should have been positive. This could have been done in system identification. However, the understanding of how detrimental a negative K_p value would affect the system was not understood early enough in the project. These two transfer functions were replaced with FO5 (FO5_S_{POd}) which provided sufficient results for the modeling part of the project.

The "+/-" values that comes after gain K_p, time constant T_p, and time delay T_d values varies a lot. If the "+/-" value are very big it indicates that there are somewhat of an uncertainty around the actual value and vice versa.

The system identification part could have been performed again where the emphasis on getting all the aeration rates to have negative values was higher. This would have given better insight in how well the final prediction error, mean squared error, fit to estimation- and validation data would have been with the right criteria. The solution of swapping FO7s and FO8s with FO5s was the best solution that could have been taking at that stage of the project. The correlation FO5s has with FO7s and FO8s are also very high which justifies the swap. The system identification results were sufficient to use them for control strategies.

After doing all the tests on all the TFs for system identification with S_{ins} for the validation dataset (week 49) without interpolating for the peek that appears, it was decided to not redo all the TFs. It would improve results if this peek would have been reduced. This also applies many other places as well. However, it was also discovered that using week 51 as the validation dataset and week 50 as the estimation dataset significantly improved the results. The week 49 dataset is very noisy and not the best dataset to achieve good results on. Fine tuning the datasets with preprocessing techniques will improve results. However, if the result is adequate for doing control strategies it would be beneficial to move forward with worse results than to use too much time on this task. The main objective is getting the model output to follow the trends of the validation dataset as good as possible with gain and time constants that makes sense.

In theory dissolved oxygen (S_O) will either be a process where oxygen in air bubbles gets taken up or it is a process where the bacteria consume the oxygen in the basins. This is both first order reactions which fits well with the models developed in the system identification process.

NOX will have a big impact on the bacteria at the inlet and for the anerobic basins. The bacteria will rather denitrify NOX (NOX→N2 gas) than eat carbon and release polyphosphate, PO4-P, which will result in a poor effect in the aerobic basins. NOX will however not influence the oxygen uptake in the aerobic basins, it will rather indirectly affect the bacteria consumption of oxygen in the aerobic basins. That’s why it’s not included for the dissolved oxygen transfer functions. It’s just included for the transfer function for polyphosphate, S_{POd} (tf_s51sfnfspo).

D. Control strategies

The model predictive controller (MPC) performance was mostly better than that of the Proportional-Integral-Derivative controller (PID). The MPC setpoint tracking was significantly better. The settling time of the multivariate controller after each disturbance introduced is notably smaller than that of the PID controller. The MPC has much better disturbance rejection as well. This can be seen by visual inspection of Figure 21 and Figure 22 seen in IV.G.2) section. The IAE index showcases the same as the outputs, with the MPC having 46.4% of the IAE as the PID. The IAMV values for the MPC for the manipulated variables was also superior to that of the PID IATV_S_{POd} result except for IAMV_FO6 and IAMV_FO10 when looking at Table 14 in section IV.H. The IAMV index illustrated the similar results as the outputs, with the MPC having 40.4% of the IAMV as the PID. The computation time gave the PID the advantage. Generally, a PID structure is simpler than that of a MPC which will result in less computation time.

There could have been conducted more tests to determine the direction of test procedure trends of the disturbance- and control variable. However, the results obtained for the control strategies was optimal. Therefore, further testing was not performed. There could also been performed tests with different procedures and step changes. Some inaccurate tests were also performed earlier in the project.

The control strategies were able to track the setpoint and adjust after the setpoint. This is mostly because of the closed-loop structure of the control strategies. A closed-loop system can correct itself when errors occur. This is highly advantageous when dealing with systems with volatile changes.

The constraints could possible been different. The upper limit was 0.3 and the lower limit was -0.3. The upper limit could have been higher. This would have been more realistic since there are some values above 0.3 that does not get taken into consideration. However, by limiting the control strategies the process will operate in safer conditions and will ensure better efficiency.

The initial tuning parameters derived by the Skogestad and Seborg et al. was good starting points for getting the right parameters for the control strategies. However, these were not applied in the end since some additional tuning with trial-and-error testing in Simulink achieved better results for the strategies. Therefore, somewhat of a difference between the theoretically correct parameters and the parameters that achieves the optimal results can be the case in some systems.

In section IV.G.2), Table 16 presents the cumulative sum of the aeration rate (F_O). The cumulative sum for the system without control strategies was $6.3413 \cdot 10^{11}$. While for the PID it was $7.2625 \cdot 10^{5}$ and 422.6788 for the MPC. The cumulative sum of the F_O for the MPC being such low values compared to the cumulative sum of F_O for both the actual value and the PID showcases how energy effective the MPC would be in a real system. However, the cumulative sum for the PID and the actual value seems to be unrealistic. It's possible that the actual value accounts for more values than the MPC and PID. This can be because of the constraints set on the MPC and PID. There are some uncertainties if this is the correct cumulative sum for each of them. Therefore, further investigation is needed.

This project is a novel control strategy for the Hias wastewater treatment and water resource recovery facilities (WRRF). There are possibilities of comparing the results achieved with other people's findings. Comparing how well this approach is compared to others would be insightful. However, this comparison has not been made.

VI. CONCLUSION

The data preprocessing could have been improved with other techniques than K-nearest-neighbor for filling the missing values. However, the dynamic changes of the industrial dataset were captured and utilized. K-nearest neighbor were still able to achieve satisfying correlations between the variables as well. The results of the system identification part were sufficient since parameters where able to represent the Hias Wastewater treatment and water resource recovery facilities (WRRF) adequately. The dynamic linear model used for this project was somewhat different from the dynamic linear model derived from the system identification. However, it did not differentiate itself too much from the measurement of polyphosphate at the disk filter (S_{POd}). This thesis aim was to assess the energy efficiency by utilizing a MPC to control the phosphorus levels in the Hias process and see if the PIDs performance was better. The control strategies achieved preferable results. With closed-loop control strategies the errors in a system that occurs gets corrected very well. Both control strategies can be implemented at the Hias WRRF with the tuning parameters in this project. The model predictive controller (MPC) showcased a better performance than the Proportional-integral-derivative (PID) controller. With Integral of total movement in manipulated variables (IAMV), Integral of absolute error (IAE) for the control variables being lower for the MPC. The cumulative sum of aeration rate or the total amount of aeration was substantially lower for the MPC than that of the PID and for the actual system. This indicates that the MPC will be notably energy efficient if implemented in the Hias WRRF. However, for industrial implementation, PID controllers is for the most part preferred because of its simplicity and faster computation time, even though the MPC performance is better. The MPC takes approximately twice as long to compute as the PID does. The results show that both control strategies are able to follow setpoint changes and reject disturbances. As future work, this should be validated with first principles simulator.

VII. FUTURE WORK

Other data preprocessing for the missing values could have been utilized. Using other methods such as cubic spline could have improved the correlation between the variables. Other resampling methods could have been explored as well.

Making transfer functions for two weeks in January to get some more comparisons would also be an important task to perform.

Developing a first principle's model to test the control strategies would also be very important to implement for a future project involving the same strategies in this project.

Developing the feedforward control strategy can achieve preferable results. Comparing a feedforward controller with the PID and MPC controller would give suitable insight in how well each controller would perform in the Hias process.

Controlling Nitrogen dioxide (NO_2) and Nitrate (NO_3) with advanced control strategies would also be beneficial for the environment. Controlling soluble oxygen demand (S_S) would be very beneficial for a wastewater treatment and water resource recovery facilities as well.

There was a lot of time spent on trying to implement PID controllers in a cascade where seven PIDs was implemented to control the aeration rate (F_O) to get the right dissolved oxygen (S_O) values. However, there should have been transfer functions between S_O and polyphosphate (S_{POd}) to get this to work. This was not realized until it was too late to implement it. The idea was to make a cascade structure with a PID controller that was a part of the outer loop, which makes it the master PID that sets the set point. The valves being a part of the inner loop and has one slave PID controller for each of them that controls how much air that will be applied to the system. Each of these PID controller would have different impacts on maintaining the setpoint for outer loop PID. The first valve has the most impact since the aeration rate is the strongest there, and the next valves has descending impacts on the outer loop PID. However, this did not work out after all. The idea for the MPC was to implement it in some of the same ways as the cascade PID structure described above. With the MPC just replacing the inner loop PIDs. Some of the work on the system identification part has been presented in the appendix section IX.D. Some of the simulation models is presented in the appendix section IX.E.1). The results have been better with a different m-script. However, due to time limited reasons the poor results that occurs now, this will not be presented.

A stochastic model predictive controller (SMPC) is a great MPC algorithm that could be implemented to this process. A SMPC is an extension of MPC that takes the presence of uncertainty of the nutrient composition and flow rate in the influent into consideration and utilizes probabilistic models to predict future behavior and optimize control actions (Ali Meshbah, 2016). Both SMPC and MPC will optimize multiple controlled variables by using information of process influent variables and dynamic models. Developing a SMPC could enhance the performance even further.

VIII. REFRENCES

Ali Meshbah. Stochastic Model Predictive Control: An Overview and Perspectives for Future Research. (2016). *IEEE Control Systems*, *36*(6), 30–44. https://doi.org/10.1109/mcs.2016.2602087

Araki, M. (2009). *CONTROL SYSTEMS, ROBOTICS ANDS AUTOMATION – Volume VII – PID Control*.

Avløp - Miljødirektoratet. (2022). Miljødirektoratet/Norwegian Environment Agency. https://www.miljodirektoratet.no/ansvarsomrader/forurensning/avlop/regulering-avlop/

Darby, M. L., & Nikolaou, M. (2012). MPC: Current practice and challenges. *Control Engineering Practice*, *20*(4), 328–342. https://doi.org/10.1016/j.conengprac.2011.12.004

Discriminatory Analysis. Nonparametric Discrimination: Consistency Properties on JSTOR. (2023). Jstor.org. https://www.jstor.org/stable/1403797?seq=3

European Commision (2021). Urban Wastewater Directive Overview. https://environment.ec.europa.eu/topics/water/urban-wastewater_en

Forskrift om begrensning av forurensning (forurensningsforskriften) - Del 4. Avløp - Lovdata. (2021). Lovdata.no; Lovdata. https://lovdata.no/dokument/SF/forskrift/2004-06-01-931/KAPITTEL_4#KAPITTEL_4

Ghoneim, W. A. M., Helal, A. A., & Wahab, M. G. A. (2016). Renewable energy resources and recovery opportunities in wastewater treatment plants. 2016 3rd International *Conference on Renewable Energies for Developing Countries (REDEC)*. https://doi.org/10.1109/redec.2016.7577509

Helness, H., & Ødegaard, H. (1999). Biological Phosphorus Removal in a Sequencing Batch Moving Bed Biofilm Reactor. *Water Science and Technology*, *40*(4-5), 161–168. https://doi.org/10.2166/wst.1999.0588

Komulainen T.M., et al. (2023), Estimation of nutrient composition in Hias MBBR-EBPR process, IWA Science & Technology.

Kommunalt avløpsvann - Miljøstatus. (2023). Miljøstatus. https://miljostatus.miljodirektoratet.no/tema/forurensning/kommunalt-avlopsvann/

Ljung, L. (1999). System identification: theory for user.

Nair, A., Hykkerud, A., & Ratnaweera, H. (2022). Estimating Phosphorus and COD Concentrations Using a Hybrid Soft Sensor: A Case Study in a Norwegian Municipal Wastewater Treatment Plant. *Water*, *14*(3), 332. https://doi.org/10.3390/w14030332

Nair, A. M. (2020). Innovative Surveillance and Process Control in Water Resource Recovery Facilities. *Unit.no*. https://doi.org/978-82-575-1735-9

Nair, A. M., Fanta, A., Haugen, F. A., & Ratnaweera, H. (2019). Implementing an Extended Kalman Filter for estimating nutrient composition in a sequential batch MBBR pilot plant. *Water Science and Technology*, *80*(2), 317–328. https://doi.org/10.2166/wst.2019.272

Meijering, E. (2002). A chronology of interpolation: from ancient astronomy to modern signal and image processing. *Proceedings of the IEEE*, *90*(3), 319–342. https://doi.org/10.1109/5.993400

Phosphorus in Wastewater, Analysis, Removal Strategies. (2023). Ysi.com. https://www.ysi.com/PHOSPHORUS-IN-WASTEWATER

Rudi, K., Goa, I. A., Saltnes, T., Sørensen, G., Angell, I. L., & Eikås, S. (2019). Microbial ecological processes in MBBR biofilms for biological phosphorus removal from wastewater. *Water Science and Technology*, *79*(8), 1467–1473. https://doi.org/10.2166/wst.2019.149

Seborg. D.E, Edgar, T.F., ., Mellichamp, D. A., & Doyle, F. J. (2017). Process Dynamics and Control (4th EMEA ed.). Hoboken NJ:Wiley.

Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning. *Journal of Process Control*, *13*(4), 291–309. https://doi.org/10.1016/s0959-1524(02)00062-8

Villard, D., Torgeir Saltnes, Gjermund Sørensen, & Rudi, K. (2022, February 24). *Spatial fractionation of phosphorus accumulating biofilm: stratification of polyphosphate accumulation and...* ResearchGate; Taylor & Francis. https://doi.org/10.1080/08927014.2022.2044475

Villard, D., Torgeir Saltnes, Gjermund Sørensen, & Rudi, K. (2022, February 24). *Spatial fractionation of phosphorus accumulating biofilm: stratification of polyphosphate accumulation and...* ResearchGate; Taylor & Francis. https://doi.org/10.1080/08927014.2022.2044475

Xylem YSI Municipal Water. (2021). Phopshorus in wastewater - a complete guide to phophorus limits, online analysis & removal strategies. Retrieved 2022, from https://www.ysi.com/File%20Library/Gated%20Documents/Phosphorus-in-Wastewater-e-book.pdf

IX. Appendix

A. Appendix 1

1) M-script for the master's thesis

```
%%%
%Hias MBBR process parameters and variables
%PACBAL project
%ACIT5900 Master's Thesis
%by Einar Nermo (s331440@oslomet.no)
%%%



%To switch datasets just switch the place of where the variables
%for the datasets are
%The "not used" remarks shows that that part was not used for the final
%product. However, they were a part of the process.

%Online data
%Load data and set them into variables for week 49 dataset
%This dataset was not used after all, it was replaced for week 51 dataset
Data1 = load('Data49.mat');

Ssin= Data1.Hiasonlinedataw49.Ssin; %Soluble chemical oxgyen demand (COD) in the inlet, mg COD/L, DV, not used
F= Data1.Hiasonlinedataw49.F; %Wastewater flow into the system, L/s, DV, not used

Ssins= Data1.Hiasonlinedataw49.Ssins; %Soluble chemical oxgyen demand (COD) in the inlet, mg COD/L, DV, scaled
Fs= Data1.Hiasonlinedataw49.Fs; %Wastewater flow into the system, L/s, DV, scaled
NOX=Data1.Hiasonlinedataw49.NOX; %NO2 and NO3 combined, DV


FO4= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 4, L/h, MV
FO5= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 5, L/h, MV
FO6= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 6, L/h, MV
FO7= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 7, L/h, MV
FO8= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 8, L/h, MV
FO9= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 9, L/h, MV
FO10= Data1.Hiasonlinedataw49.FO4; %Flow rate of oxygen (aeration) in basin 10, L/h, MV

FO4s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 4, L/h, MV
FO5s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 5, L/h, MV
FO6s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 6, L/h, MV
FO7s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 7, L/h, MV
FO8s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 8, L/h, MV
FO9s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 9, L/h, MV
FO10s= Data1.Hiasonlinedataw49.FO4s; %Flow rate of oxygen (aeration) in basin 10, L/h, MV


SO4= Data1.Hiasonlinedataw49.SO4; %Dissolved oxygen in basin 4, mg O2/L, CV
SO5= Data1.Hiasonlinedataw49.SO5; %Dissolved oxygen in basin 5, mg O2/L, CV
SO6= Data1.Hiasonlinedataw49.SO6; %Dissolved oxygen in basin 6, mg O2/L, CV
SO8= Data1.Hiasonlinedataw49.SO8; %Dissolved oxygen in basin 8, mg O2/L, CV
SO9= Data1.Hiasonlinedataw49.SO9; %Dissolved oxygen in basin 9, mg O2/L, CV

SO7= (SO6+SO8)/2; %Dissolved oxygen in basin 7, mg O2/L, CV
SO10= SO9-(SO8-SO9); %Dissolved oxygen in basin 10, mg O2/L, CV

SPOd= Data1.Hiasonlinedataw49.SPOd; %Soluble phosphate (PO4) that comes out of the disk filter, mg P/L, CV
```

```
%Online data
%Load data and set them into variables for week 50 dataset

Data1 = load('Data50.mat');


Ssin= Data1.Hiasonlinedataw50.Ssin; %Soluble chemical oxygen demand (COD) in the inlet, mg COD/L, DV, not used
F= Data1.Hiasonlinedataw50.F; %Wastewater flow into the system, L/s, DV, not used

Ssins= Data1.Hiasonlinedataw50.Ssins; %Soluble chemical oxygen demand (COD) in the inlet, mg COD/L, DV scaled
Fs= Data1.Hiasonlinedataw50.Fs; %Wastewater flow into the system, L/s, DV scaled
NOX=Data1.Hiasonlinedataw50.NOX; %NO2 and NO3 combined, DV


FO4= Data1.Hiasonlinedataw50.FO4; %Flow rate of oxygen (aeration) in basin 4, L/h, MV, not used
FO5= Data1.Hiasonlinedataw50.FO5; %Flow rate of oxygen (aeration) in basin 5, L/h, MV, not used
FO6= Data1.Hiasonlinedataw50.FO6; %Flow rate of oxygen (aeration) in basin 6, L/h, MV, not used
FO7= Data1.Hiasonlinedataw50.FO7; %Flow rate of oxygen (aeration) in basin 7, L/h, MV, not used
FO8= Data1.Hiasonlinedataw50.FO8; %Flow rate of oxygen (aeration) in basin 8, L/h, MV, not used
FO9= Data1.Hiasonlinedataw50.FO9; %Flow rate of oxygen (aeration) in basin 9, L/h, MV, not used
FO10= Data1.Hiasonlinedataw50.FO10; %Flow rate of oxygen (aeration) in basin 10, L/h, MV, not used

FO4s= Data1.Hiasonlinedataw50.FO4s; %Flow rate of oxygen (aeration) in basin 4, L/h, MV
FO5s= Data1.Hiasonlinedataw50.FO5s; %Flow rate of oxygen (aeration) in basin 5, L/h, MV
FO6s= Data1.Hiasonlinedataw50.FO6s; %Flow rate of oxygen (aeration) in basin 6, L/h, MV
FO7s= Data1.Hiasonlinedataw50.FO7s; %Flow rate of oxygen (aeration) in basin 7, L/h, MV
FO8s= Data1.Hiasonlinedataw50.FO8s; %Flow rate of oxygen (aeration) in basin 8, L/h, MV
FO9s= Data1.Hiasonlinedataw50.FO9s; %Flow rate of oxygen (aeration) in basin 9, L/h, MV
FO10s= Data1.Hiasonlinedataw50.FO10s; %Flow rate of oxygen (aeration) in basin 10, L/h, MV


SO4= Data1.Hiasonlinedataw50.SO4; %Dissolved oxygen in basin 4, mg O2/L, CV
SO5= Data1.Hiasonlinedataw50.SO5; %Dissolved oxygen in basin 5, mg O2/L, CV
SO6= Data1.Hiasonlinedataw50.SO6; %Dissolved oxygen in basin 6, mg O2/L, CV
SO8= Data1.Hiasonlinedataw50.SO8; %Dissolved oxygen in basin 8, mg O2/L, CV
SO9= Data1.Hiasonlinedataw50.SO9; %Dissolved oxygen in basin 9, mg O2/L, CV

SO7= (SO6+SO8)/2; %Dissolved oxygen in basin 7, mg O2/L, CV
SO10= SO9-(SO8-SO9); %Dissolved oxygen in basin 10, mg O2/L, CV

SPOd= Data1.Hiasonlinedataw50.SPOd; %Soluble phosphate (PO4) that comes out of the disk filter, mg P/L, CV




%Online data
%Load data and set them into variables for week 51 dataset
Data1 = load('Data51.mat');

SSin= Data1.Hiasonlinedataw51.Ssin; %Soluble chemical oxygen demand (COD) in the inlet, mg COD/L, DV, not used
F= Data1.Hiasonlinedataw51.F; %Wastewater flow into the system, L/s, DV, not used
```

```
SSins= Data1.Hiasonlinedataw51.Ssins; %Soluble chemical oxygen demand (COD) in the inlet, mg COD/L, DV,
scaled
Fs= Data1.Hiasonlinedataw51.Fs; %Wastewater flow into the system, L/s, DV, scaled
NOX=Data1.Hiasonlinedataw51.NOX; %NO2 and NO3 combined, DV


FO4= Data1.Hiasonlinedataw51.FO4; %Flow rate of oxygen (aeration) in basin 4, L/h, MV, not used
FO5= Data1.Hiasonlinedataw51.FO5; %Flow rate of oxygen (aeration) in basin 5, L/h, MV, not used
FO6= Data1.Hiasonlinedataw51.FO6; %Flow rate of oxygen (aeration) in basin 6, L/h, MV, not used
FO7= Data1.Hiasonlinedataw51.FO7; %Flow rate of oxygen (aeration) in basin 7, L/h, MV, not used
FO8= Data1.Hiasonlinedataw51.FO8; %Flow rate of oxygen (aeration) in basin 8, L/h, MV, not used
FO9= Data1.Hiasonlinedataw51.FO9; %Flow rate of oxygen (aeration) in basin 9, L/h, MV, not used
FO10= Data1.Hiasonlinedataw51.FO10; %Flow rate of oxygen (aeration) in basin 10, L/h, MV, not used

FO4s= Data1.Hiasonlinedataw51.FO4s; %Flow rate of oxygen (aeration) in basin 4, L/h, MV
FO5s= Data1.Hiasonlinedataw51.FO5s; %Flow rate of oxygen (aeration) in basin 5, L/h, MV
FO6s= Data1.Hiasonlinedataw51.FO6s; %Flow rate of oxygen (aeration) in basin 6, L/h, MV
FO7s= Data1.Hiasonlinedataw51.FO7s; %Flow rate of oxygen (aeration) in basin 7, L/h, MV
FO8s= Data1.Hiasonlinedataw51.FO8s; %Flow rate of oxygen (aeration) in basin 8, L/h, MV
FO9s= Data1.Hiasonlinedataw51.FO9s; %Flow rate of oxygen (aeration) in basin 9, L/h, MV
FO10s= Data1.Hiasonlinedataw51.FO10s; %Flow rate of oxygen (aeration) in basin 10, L/h, MV


SO4= Data1.Hiasonlinedataw51.SO4; %Dissolved oxygen in basin 4, mg O2/L, CV, not used
SO5= Data1.Hiasonlinedataw51.SO5; %Dissolved oxygen in basin 5, mg O2/L, CV, not used
SO6= Data1.Hiasonlinedataw51.SO6; %Dissolved oxygen in basin 6, mg O2/L, CV, not used
SO8= Data1.Hiasonlinedataw51.SO8; %Dissolved oxygen in basin 8, mg O2/L, CV, not used
SO9= Data1.Hiasonlinedataw51.SO9; %Dissolved oxygen in basin 9, mg O2/L, Cv, not used

SO7= (SO6+SO8)/2; %Dissolved oxygen in basin 7, mg O2/L, CV
SO10= SO9-(SO8-SO9); %Dissolved oxygen in basin 10, mg O2/L, CV


SPOd= Data1.Hiasonlinedataw51.SPOd; %Soluble phosphate (PO4) that comes out of the disk filter, mg P/L, CV



%Assumptions:
%Assuming FO4(s)-FO10(s) is manipulated variables (MVs)
%Assuming SSin(s), F(s) and NOX is disturbance variables (DVs)
%Assuming SO4-SO10 and SPOd is control variables (CVs)


%This is the mean for week 51 variables and will be used to get initial
%values and used for deviation variables and set point
FO4m= mean(FO4s); %FO4m=2.9980
FO5m= mean(FO5s); %FO5m=1.8193
FO6m= mean(FO6s); %FO6m=1.5715
FO7m= mean(FO7s); %FO7m=1.0550
FO8m= mean(FO8s); %FO8m=0.8462
FO9m= mean(FO9s); %FO9m=0.6239
FO10m= mean(FO10s); %FO10m=0.5214

SO4m= mean(SO4); %SO4m =5.3729
SO5m= mean(SO5); %SO5m =6.0622
SO6m= mean(SO6); %SO6m =5.8070
SO7m= mean(SO7); %SO7m =5.4728
SO8m= mean(SO8); %SO8m =5.1387
SO9m= mean(SO9); %SO9m =5.0493
SO10m= mean(SO10); %SO10m =4.9598
```

```
Fsm= mean(Fs); %Fsm=0.0870
SSinsm=mean(SSins); %SSinsm=0.5088
NOXm=mean(NOX); %NOXm=2.6089

SPOm=mean(SPOd); %SPOm=0.2325





%Interpolate measurement vectors from 144 sample/day to 1440 samples/day
%Current sampling of measurement 144/day.
%This will make the dataset go from having 817 datapoints to 8170, this
%will make x-axis be in minutes, which is the same as the system
%identification models or transfer functions

N0=max(size(Fs));
N=(N0-1)*10;
t0=0:10:N;
t1=0:1:N;


%Interpolate
Fs=interp1(t0,Fs,t1);
SSins=interp1(t0,SSins,t1);
NOX=interp1(t0,NOX,t1);
SPOd=interp1(t0,SPOd,t1);

FO4s=interp1(t0,FO4s,t1);
FO5s=interp1(t0,FO5s,t1);
FO6s=interp1(t0,FO6s,t1);
FO7s=interp1(t0,FO7s,t1);
FO8s=interp1(t0,FO8s,t1);
FO9s=interp1(t0,FO9s,t1);
FO10s=interp1(t0,FO10s,t1);

SO4=interp1(t0,SO4,t1);
SO5=interp1(t0,SO5,t1);
SO6=interp1(t0,SO6,t1);
SO7=interp1(t0,SO7,t1);
SO8=interp1(t0,SO8,t1);
SO9=interp1(t0,SO9,t1);
SO10=interp1(t0,SO10,t1);


%Cumulative sum of the FOs

FO4s_c = cumsum(FO4s);
FO5s_c = cumsum(FO5s);
FO6s_c = cumsum(FO6s);
FO7s_c = cumsum(FO7s);
FO8s_c = cumsum(FO8s);
FO9s_c = cumsum(FO9s);
FO10s_c = cumsum(FO10s);
sum (FO4s_c);

sum (FO4s_c+FO5s_c+FO6s_c+FO7s_c+FO8s_c+FO9s_c+FO10s_c) % 1.0e+11 * 6.3413


%To get each of the subplots switch places and have the plot you want
```

```
%in the end of the plots section
%Plots for dissolved oxygen variables (SO)
subplot(7,1,1);
plot(SO4);
title('SO4');
xlabel('Time (minutes)');

subplot(7,1,2);
plot(SO5);
title('SO5');
xlabel('Time (minutes)');

subplot(7,1,3);
plot(SO6);
title('SO6');
xlabel('Time (minutes)');

subplot(7,1,4);
plot(SO7);
title('SO7');
xlabel('Time (minutes)');

subplot(7,1,5);
plot(SO8);
title('SO8');
xlabel('Time (minutes)');

subplot(7,1,6);
plot(SO9);
title('SO9');
xlabel('Time (minutes)');

subplot(7,1,7);
plot(SO10);
title('SO10');
xlabel('Time (minutes)');

%Plots for all the aeration variables
subplot(7,1,1);
plot(FO4s);
title('FO4s');
xlabel('Time (minutes)');

subplot(7,1,2);
plot(FO5s);
title('FO5s');
xlabel('Time (minutes)');

subplot(7,1,3);
plot(FO6s);
title('FO6s');
xlabel('Time (minutes)');

subplot(7,1,4);
plot(FO7s);
title('FO7s');
xlabel('Time (minutes)');

subplot(7,1,5);
plot(FO8s);
title('FO8s');
xlabel('Time (minutes)');
```

```
subplot(7,1,6);
plot(FO9s);
title('FO9s');
xlabel('Time (minutes)');

subplot(7,1,7);
plot(FO10s);
title('FO10s');
xlabel('Time (minutes)');

%Plots for the rest of the variables
subplot(4,1,1);

plot(Fs); % plot all input data
title('Fs');
xlabel('Time (minutes)');

subplot(4,1,2);
plot(SSins);
title('SSins');
xlabel('Time (minutes)');

subplot(4,1,3);
plot(NOX);
title('NOX');
xlabel('Time (minutes)');

subplot(4,1,4);
plot(SPOd);
title('SPOd');
xlabel('Time (minutes)');



%The out variable in workspace that gets generated after running the
%simulink for the PID must be deleted after generating the test procedure trends for FO
%for the PID and then run the simulink for the MPC to generate plots for
%the both of them
%This code won't run the first time becouse of this section, since this
%section require the out variable that is generated by the simulink model

%Plotting all the test procedure trends for FO for the PID
subplot(7,1,1);
plot(out.FO4sPID);
title('FO4s(PID)');
xlabel('Time (minutes)');

subplot(7,1,2);
plot(out.FO5sPID);
title('FO5s(PID)');
xlabel('Time (minutes)');

subplot(7,1,3);
plot(out.FO6sPID);
title('FO6s(PID)');
xlabel('Time (minutes)');

subplot(7,1,4);
plot(out.FO7sPID);
title('FO7s(PID)');
xlabel('Time (minutes)');
```

```
subplot(7,1,5);
plot(out.FO8sPID);
title('FO8s(PID)');
xlabel('Time (minutes)');

subplot(7,1,6);
plot(out.FO9sPID);
title('FO9s(PID)');
xlabel('Time (minutes)');

subplot(7,1,7);
plot(out.FO10sPID);
title('FO10s(PID)');
xlabel('Time (minutes)');

%Plotting all the test procedure trends for FO for the MPC
subplot(7,1,1);
plot(out.FO4sMPC);
title('FO4s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,2);
plot(out.FO5sMPC);
title('FO5s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,3);
plot(out.FO6sMPC);
title('FO6s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,4);
plot(out.FO7sMPC);
title('FO7s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,5);
plot(out.FO8sMPC);
title('FO8s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,6);
plot(out.FO9sMPC);
title('FO9s(MPC)');
xlabel('Time (minutes)');

subplot(7,1,7);
plot(out.FO10sMPC);
title('FO10s(MPC)');
xlabel('Time (minutes)');



%Transpose
%This is needed to not get dimension error for the variables, transposing
%the variables will give the right directions of the columns and rows.
Fs = [t1' Fs'];
SSins = [t1' SSins'];
NOX = [t1' NOX'];
SPOd = [t1' SPOd'];
```

```
FO4s = [t1' FO4s'];
FO5s = [t1' FO5s'];
FO6s = [t1' FO6s'];
FO7s = [t1' FO7s'];
FO8s = [t1' FO8s'];
FO9s = [t1' FO9s'];
FO10s = [t1' FO10s'];

SO4 = [t1' SO4'];
SO5 = [t1' SO5'];
SO6 = [t1' SO6'];
SO7 = [t1' SO7'];
SO8 = [t1' SO8'];
SO9 = [t1' SO9'];
SO10 = [t1' SO10'];



%These are the transfer functions between FO and SO which will not be used
%after all
%There are also other codes that have been tested out and not been a part
%of the final product.
%There are also many different simulink models where a lot of time has been spent on
%developing it but will not be a part of the final product after all.
%The same goes for the system identification part.

%Transfer function for tf_b4_sffoso, with FO4s, Fs and SSins as input and SO4 as output
FO4s_SO4=tf(0.00076014);
SO4_SO4=tf(0); %just to get right dimensions
Fs_SO4=tf(14.62,[120.38 1]);
SSins_SO4=tf(4.0728,[121.65 1]);

tf_SO4=[FO4s_SO4 SO4_SO4 Fs_SO4 SSins_SO4];

%Transfer function for tf_b5_sffoso, with FO5s, SO4, Fs and SSins as input and SO5 as output
FO5s_SO5=tf(0.56743);
SO4_SO5=tf(0.56743);
Fs_SO5=tf(11.544,[300 1]);
SSins_SO5=tf(-0.11662,[300 1]);

tf_SO5=[FO5s_SO5 SO4_SO5 Fs_SO5 SSins_SO5];

%Transfer function for tf_b6_sffoso, with FO6s, SO5, Fs and SSins as input and SO6 as output
FO6s_SO6=tf(0.63581);
SO5_SO6=tf(0.80471);
Fs_SO6=tf(-0.14453,[228.01 1]);
SSins_SO6=tf(0.048607,[240.18 1]);

tf_SO6=[FO6s_SO6 SO5_SO6 Fs_SO6 SSins_SO6];

%Transfer function for tf_b7_sffoso, with FO7s, SO6, Fs and SSins as input and SO7 as output
FO7s_SO7=tf(1.4266);
SO6_SO7=tf(0.38308);
Fs_SO7=tf(15.348,[420 1]);
SSins_SO7=tf(1.5279,[420 1]);

tf_SO7=[FO7s_SO7 SO6_SO7 Fs_SO7 SSins_SO7];


%Transfer function for tf_b8_sffoso, with FO8s, SO7, Fs and SSins as input and SO8 as output
```

```
FO8s_SO8=tf(5.6197);
SO7_SO8=tf(0.23002);
Fs_SO8=tf(-8.8233,[41.37 1]);
SSins_SO8=tf(0.23383,[480 1]);

tf_SO8=[FO8s_SO8 SO7_SO8 Fs_SO8 SSins_SO8];


%Transfer function for tf_b9_sffoso, with FO9s, SO8, Fs and SSins as input and SO9 as output
FO9s_SO9=tf(3.4222);
SO8_SO9=tf(0.75981);
Fs_SO9=tf(-3.6539,[125.15 1]);
SSins_SO9=tf(-1.1004,[75.244 1]);

tf_SO9=[FO9s_SO9 SO8_SO9 Fs_SO9 SSins_SO9];



%Transfer function for tf_b10_sffoso, with FO10s, SO9, Fs and SSins as input and SO10 as output
FO10s_SO10=tf(3.4631);
SO9_SO10=tf(0.89267);
Fs_SO10=tf(-8.5198,[750 1]);
SSins_SO10=tf(-1.1618,[342.08 1]);

tf_SO10=[FO10s_SO10 SO9_SO10 Fs_SO10  SSins_SO10];


%not used
plantSO=[tf_SO4 tf_SO5 tf_SO6 tf_SO7 tf_SO8 tf_SO9 tf_SO10];



%This is also not used
%SO4
%t5 increase 5%
t5=240;
step1=1.05;
%t6 decrease 5%
t6=840;
step2=-0.05;


%SO5
%t7 increase 5%
t7=300;
step1=1.05;
%t8 decrease 5%
t8=900;
step2=-0.05;


%SO6
%t9 increase 5%
t9=360;
step1=1.05;
%t10 decrease 5%
t10=960;
step2=-0.05;


%SO7
%t12 increase 5%
```

```
t11=420;
step1=1.05;
%t12 decrease 5%
t12=1020;
step2=-0.05;


%SO8
%t13 increase 5%
t13=480;
step1=1.05;
%t14 decrease 5%
t14=1080;
step2=-0.05;


%SO9
%t15 increase 5%
t15=540;
step1=1.05;
%t16 decrease 5%
t16=1140;
step2=-0.05;


%SO10
%t17 increase 5%
t17=600;
step1=0.05;
%t18 decrease 5%
t18=1200;
step2=-0.05;

%PID FO4
tau4=10;
tauc4=10;
Ti4=min(tau4, 4*tauc4);
Kc4=1000;

%PID FO5
tau5=10;
tauc5=10;
Ti5=min(tau5, 4*tauc5);
Kc5=1000;

%PID FO6
tau6=10;
tauc6=10;
Ti6=min(tau6, 4*tauc6);
Kc6=1000;

%PID FO7
tau7=10;
tauc7=10;
Ti7=min(tau4, 4*tauc7);
Kc7=1000;

%PID FO8
tau8=10;
tauc8=10;
Ti8=min(tau8, 4*tauc8);
Kc8=1000;
```

```
%PID FO9
tau9=10;
tauc9=10;
Ti9=min(tau9, 4*tauc9);
Kc9=1000;

%PID FO10
tau10=10;
tauc10=10;
Ti10=min(tau4, 4*tauc10);
Kc10=1000;

%PID SPOd
tauSPOd=1;
taucSPOd=1;
TiSPO=min(tauSPOd, 4*taucSPOd);
KcSPO=1000;

%This is the end of things that are not used after all, from this point
%everything else is used


%Transfer function for tf_s51sfnfspo, with SSins, Fs, NOX in, FO4s, FO5s...FO10s as input and SPOd as output
SSins_SPOd=tf(0.43109, [139.992 62.332 1],'IODelay',359.72); %(60s+1)(2.3332s+1)=139.992s^2+62.332s +1
Fs_SPOd=tf(1.2924,[277.653 39.2551 1],'IODelay',79.46); %(30s+1)(9.2551s+1)=277.653s^2+39.2551s+1
NOX_SPOd=tf(0.0023115,[0.12464 46.598675 1],'IODelay',147.61); %
(46.596s+1)(0.0026751s+1)=0.12464s^2+46.598675s+1
FO4s_SPOd=tf(-0.0029016,[472.577701 43.898 1],'IODelay',47.75); %
(18.919s+1)(24.979s+1)=472.577701s^2+43.898s+1
FO5s_SPOd=tf(-0.029011,[671.19725 52.085 1],'IODelay',81.34); %
(23.394s+1)(28.691s+1)=671.19725s^2+52.085s+1
FO6s_SPOd=tf(-0.047375,[303.0221745 55.6611 1],'IODelay',57.45); %
(49.545s+1)(6.1161s+1)=303.0221745s^2+55.6611s+1
FO7s_SPOd=tf(-0.029011,[671.19725 52.085 1],'IODelay',81.34); % (60s+1)(30s+1)=1800s^2+90s+1
FO8s_SPOd=tf(-0.029011,[671.19725 52.085 1],'IODelay',81.34); % (40.808s+
1)(30s+1)=1224.24s^2+70.808s+1
FO9s_SPOd=tf(-0.053278,[800.52 73.342 1],'IODelay',304.8); % (60s+1)(13.342s+1)=800.52s^2+73.342s+1
FO10s_SPOd=tf(-0.21394,[79.53 61.3255 1],'IODelay',315.41); % (60s+1)(1.3255s+1)=79.53s^2+61.3255s+1


%This is the original FO7s_SPOd and FO8_SPOd, replaced with FO5 since
%FO7s_SPOd and FO8s_SPOd has positive Kp values

%FO7s_SPOd=tf(0.13821,[1800 90 1],'IODelay',132.74); % (60s+1)(30s+1)=1800s^2+90s+1
%FO8s_SPOd=tf(0.12275,[1224.24 70.808 1],'IODelay',132.74); % (40.808s+1)(30s+1)=1224.24s^2+70.808s+1



tf_SPOd=[SSins_SPOd Fs_SPOd NOX_SPOd FO4s_SPOd FO5s_SPOd FO6s_SPOd FO7s_SPOd
FO8s_SPOd FO9s_SPOd FO10s_SPOd];



%SPOd
%t1 increase 5%
t1=100;
step1=-0.05;
%t2 decrease 5%
t2=700;
step2=0.05;
```

```
%SSins
%t5 decrease 5%
t3=2000;
step1=-0.05;
%t6 increase 5%
t4=2600;
step2=0.05;


%Fs
%t3 increase 5%
t5=5000;
step3=0.05;
%t4 decrease 5%
t6=5600;
step4=-0.05;

%NOX
%t7 decrease 5%
t7=7000;
step1=-0.05;
%t8 increase 5%
t8=7600;
step2=0.05;



%The mean for SO will be set as sp for FO:
%SO4m =5.3729, SO5m =6.0622, SO6m =5.8070, SO7m =5.4728, SO8m =5.1387, SO9m =5.0493, SO10m
=4.9598

SO4sp=5.3729; %SO4m =5.3729
SO5sp=6.0622; %SO5m =6.0622
SO6sp=5.8070; %SO6m =5.8070
SO7sp=5.4728; %SO7m =5.4728
SO8sp=5.1387; %SO8m =5.1387
SO9sp=5.0493; %SO9m =5.0493
SO10sp=4.9598; %SO10m =4.9598




%Ratio controllers based on mean of FO, and then tuned futher
rFO4=1.8;
rFO5=1;
rFO6=0.72418;
rFO7=0.4519;
rFO8=0.38225;
rFO9=0.2081;
rFO10=0.10;


%PID FO5 to SPOd

Tp1=40.1606; %originally 28.6533
Tp2=30.5810; %originally 23.4192

theta=81.34;
tauc=theta;
Kp=-0.02901;
```

```
Kc=1/(Kp)*(Tp1/(tauc+theta)); %-8.5098
taui=min(Tp1,(4*(tauc+theta))); %40.1606
taud=Tp2;%30.5810




%%%
% MPC controller %
%%%
plant3=setmpcsignals(tf_SPOd,'MD',[1, 2, 3]);

%Define MPC sampling time
%1/10 of 600min, which is the time it takes from basin 1 to 10 (this is not
%based on the calculated sampling time but rather the sampling time
%conveyed early on in the project)

%Sampling time Ts
Ts=60;

%Horizon for prediction and control
Tsettling=600;
N=Tsettling/Ts; %10

M=10; %N/3<M<N/2
P=610; %P=N+M


%Constraints for the manipulated(input)variables
MV1=struct('Min',-0.3,'Max',0.3);
MV2=struct('Min',-0.3,'Max',0.3);
MV3=struct('Min',-0.3,'Max',0.3);
MV4=struct('Min',-0.3,'Max',0.3);
MV5=struct('Min',-0.3,'Max',0.3);
MV6=struct('Min',-0.3,'Max',0.3);
MV7=struct('Min',-0.3,'Max',0.3);




MV=[MV1 MV2 MV3 MV4 MV5 MV6 MV7];

%Constraints for the controlled (output) variable
OV1=struct('Min',-0.3,'Max',0.3);
OV=OV1;

%Q equal weighting between controlled variables
Q=10;

%Ru zero weighting for the values of the manipulated variables(u)
Ru=[0 0 0 0 0 0 0];


%0.01
%Rd weighting for the changes in the manipulated variables(du)
Rd=[0.1 0.1 0.1 0.1 0.1 0.1 0.1];


W=struct('ManipulatedVariables',Ru,'ManipulatedVariablesRate',Rd,'OutputVariables',Q);

%Specifies MPC controller with prediction horizon (p), control horizon(m)
%and input
```

```
%and the properties of manipulated variables (MV), outputvariables (OV) and
%input disturbance.

mpcB=mpc(plant3,Ts,P,M,W,MV,OV);
XmpcB=mpcstate(mpcB);



%Cumalative sum will generate error first time running the m-script
%Cumulative sum of the FOs for the PID

FO4sPID_c = cumsum(out.FO4sPID);
FO5sPID_c = cumsum(out.FO5sPID);
FO6sPID_c = cumsum(out.FO6sPID);
FO7sPID_c = cumsum(out.FO7sPID);
FO8sPID_c = cumsum(out.FO8sPID);
FO9sPID_c = cumsum(out.FO9sPID);
FO10sPID_c = cumsum(out.FO10sPID);


sum (FO4sPID_c+FO5sPID_c+FO6sPID_c+FO7sPID_c+FO8sPID_c+FO9sPID_c+FO10sPID_c); %7.2625*e+05


%Cumulative sum of the FOs for the MPC

FO4sMPC_c = cumsum(out.FO4sMPC);
FO5sMPC_c = cumsum(out.FO5sMPC);
FO6sMPC_c = cumsum(out.FO6sMPC);
FO7sMPC_c = cumsum(out.FO7sMPC);
FO8sMPC_c = cumsum(out.FO8sMPC);
FO9sMPC_c = cumsum(out.FO9sMPC);
FO10sMPC_c = cumsum(out.FO10sMPC);

sum (FO4sMPC_c+FO5sMPC_c+FO6sMPC_c+FO7sMPC_c+FO8sMPC_c+FO9sMPC_c+FO10sMPC_c);
%422.6788
```

B. Appendix 2

1) Simulation models for the linear models

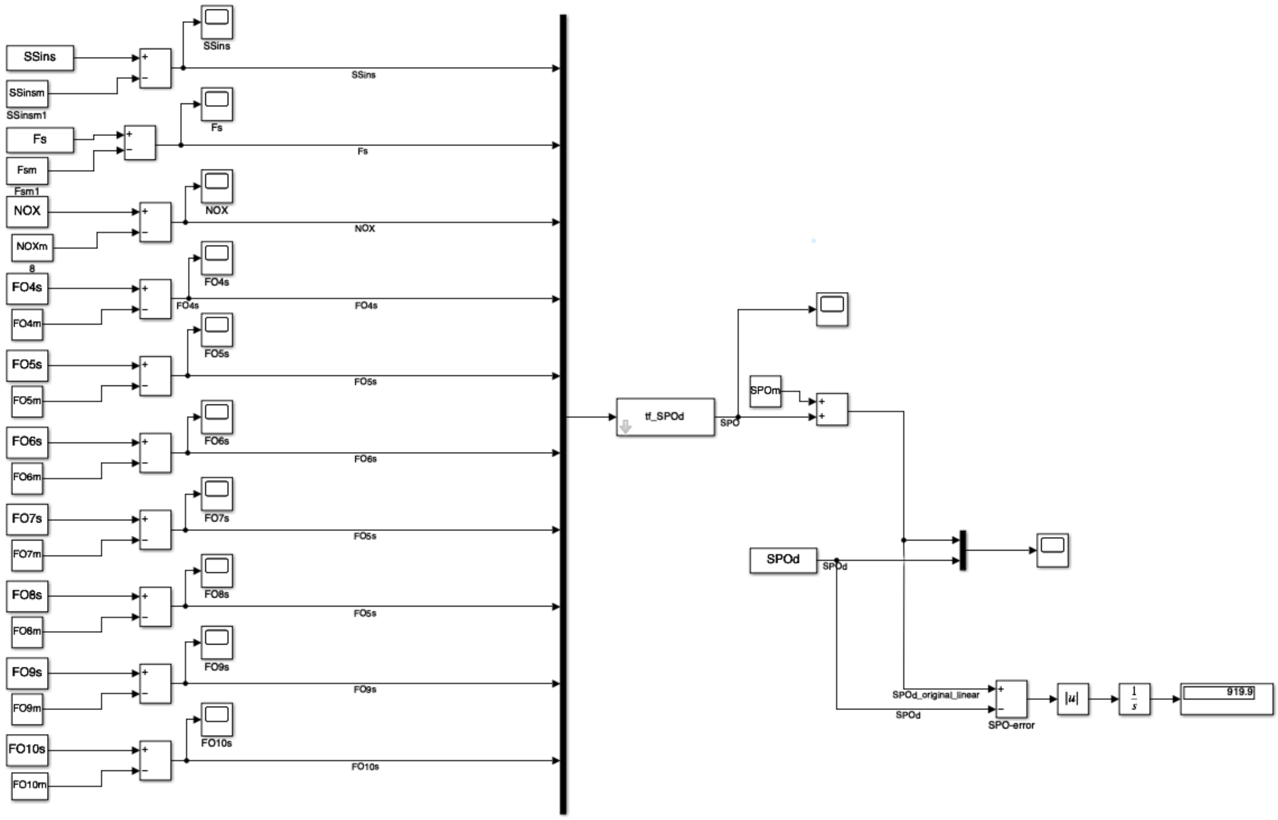

Figure 28 Simulation model for the original dynamic linear model

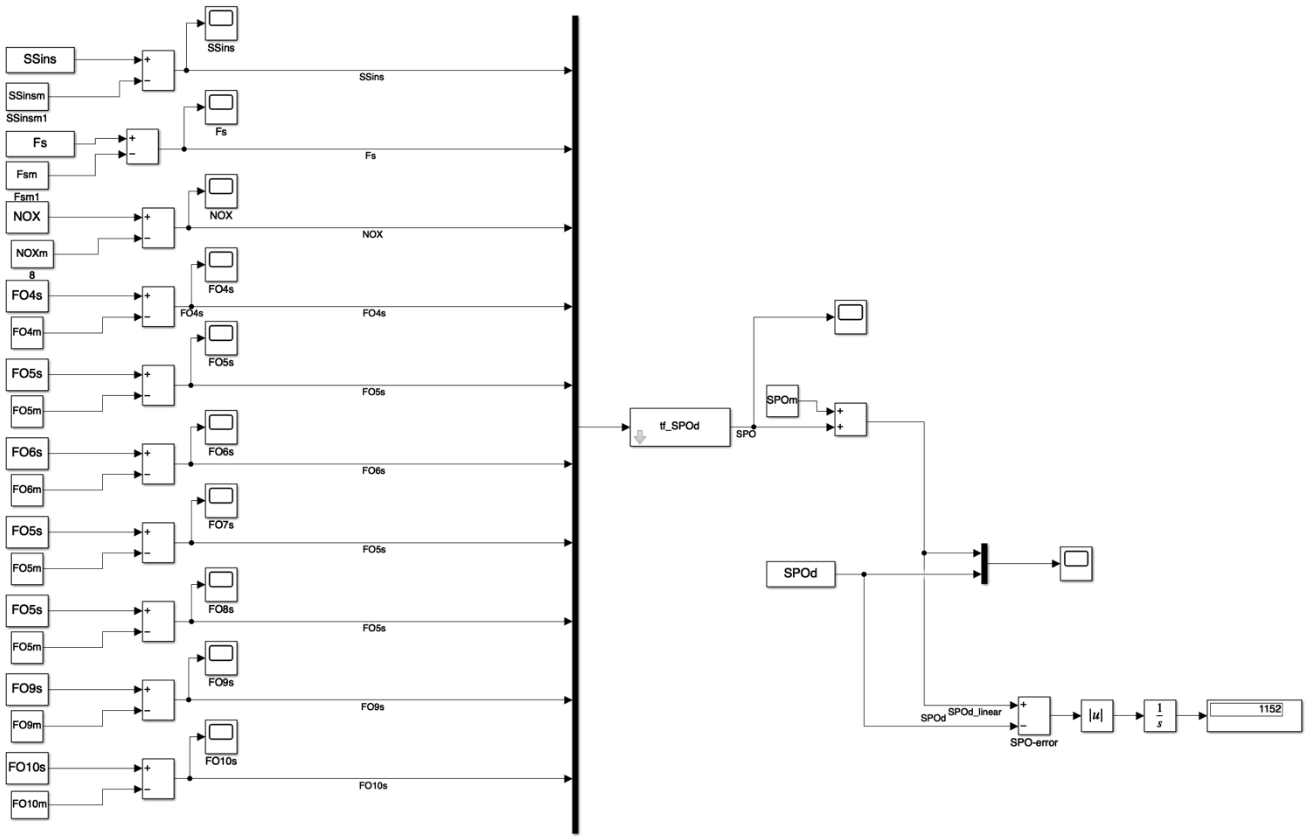

Figure 29 Simulation model for the dynamic linear model

2) Simulation model for the PID

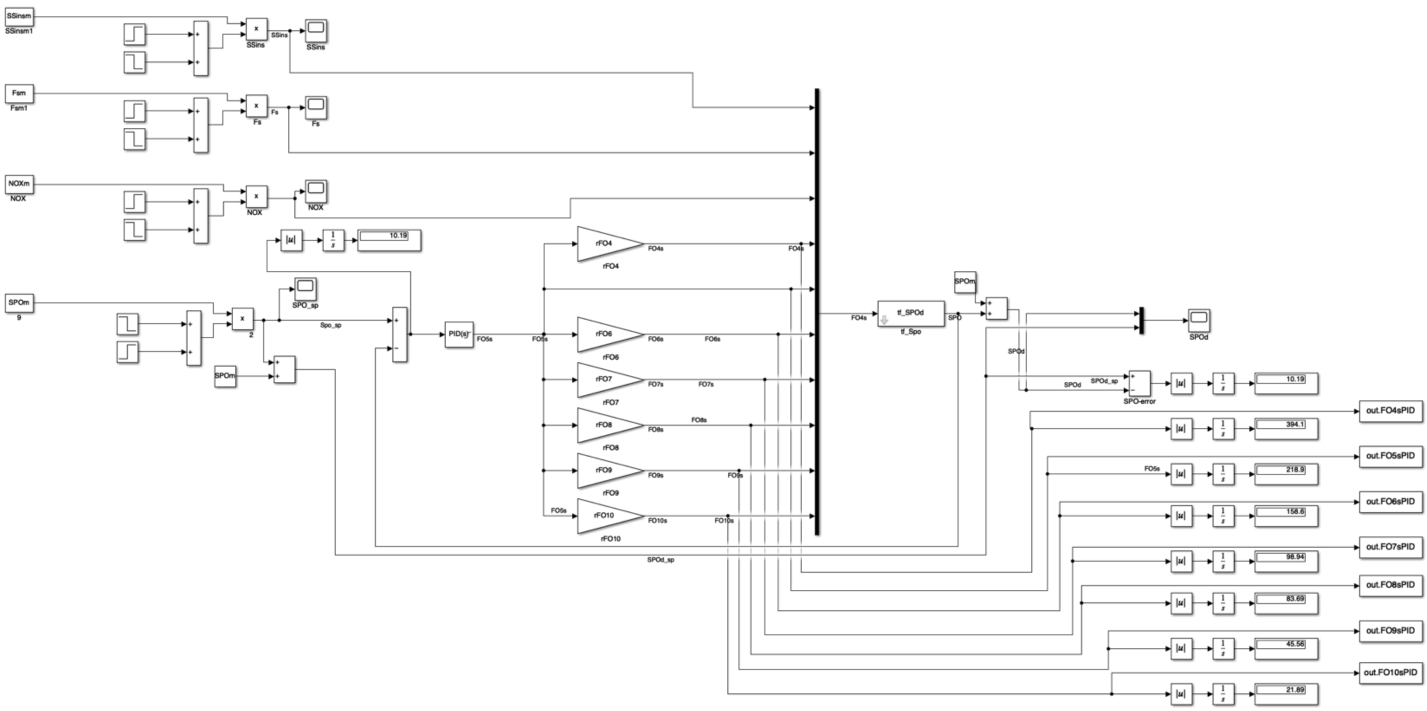

Figure 30 simulation model for the PID

3) Simulation model for the MPC

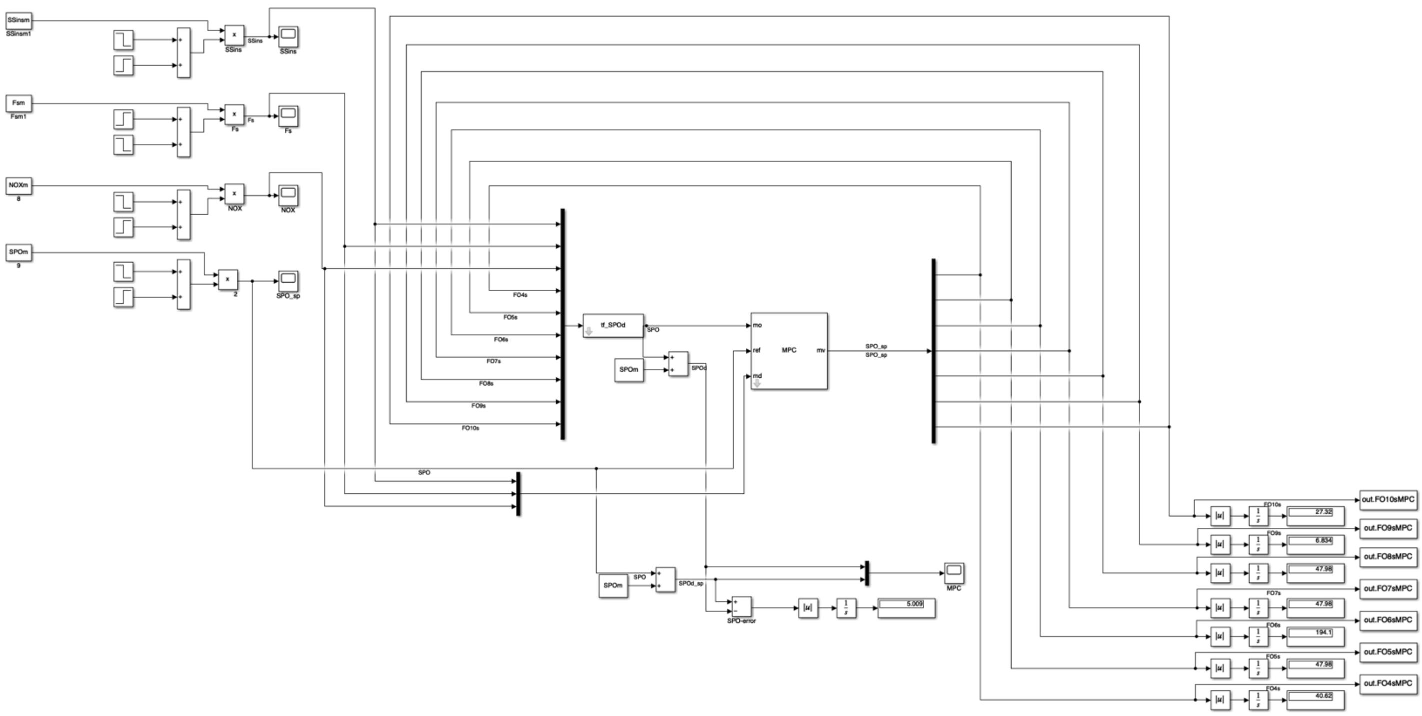

Figure 31 simulation model for the MPC

C. Appendix 3

1) Code for filling missing values in the whole december dataset with heat map and pair plot for december and for just week 51 dataset:
https://github.com/s331440/DAVE3625/blob/main/5900_project.ipynb

D. Appendix 4

1) tf_b4_foso with week 49 as validation dataset

This appendix is incomplete, there was not time to fill everything out. There should also have been system identification results for the failed attempts for the transfer functions for polyphosphate (S_{POd})

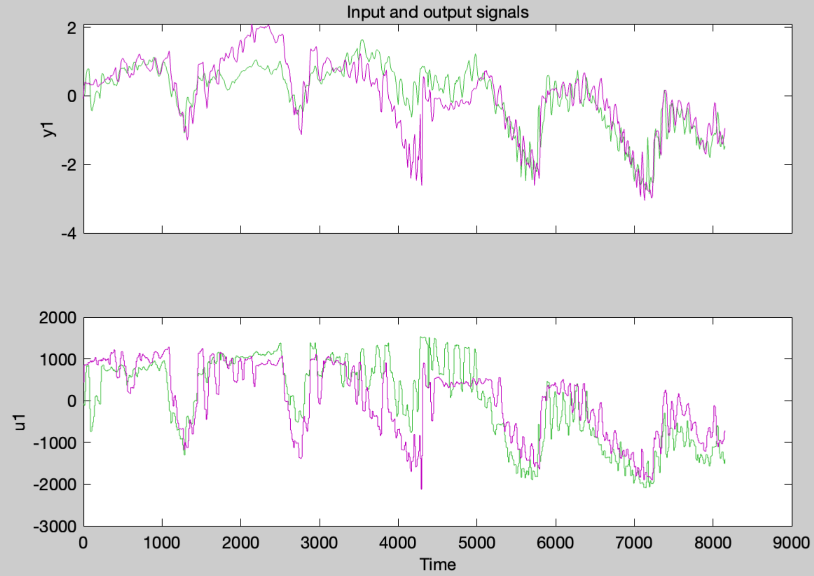

Figure 32 tf_b4_foso time plot for week 50 (purple graph) and week 49 (green graph) with FO4 as input and SO4 as output for channel 1 which is FO4

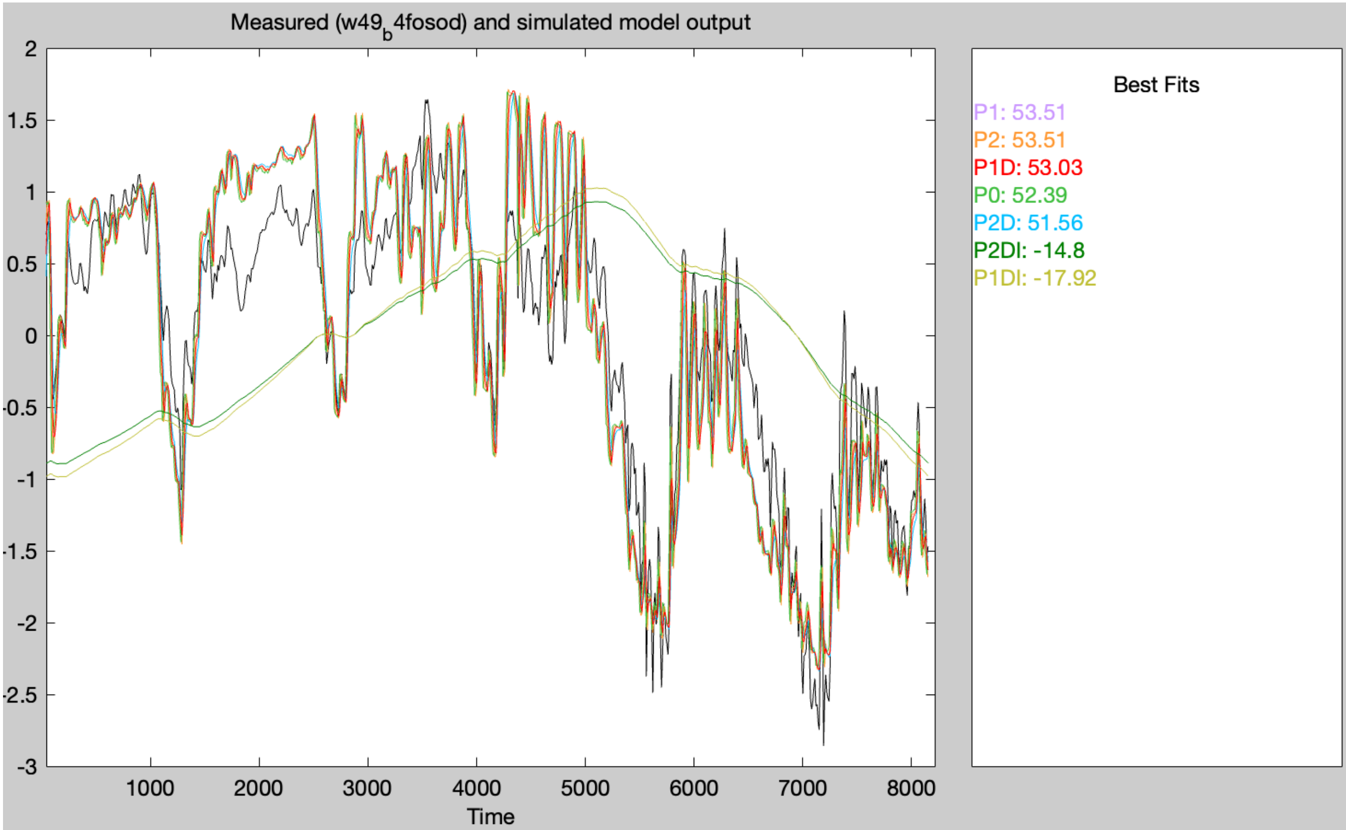

Figure 33 tf_b4_foso fitness index and plots of all model outputs with FO4 as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

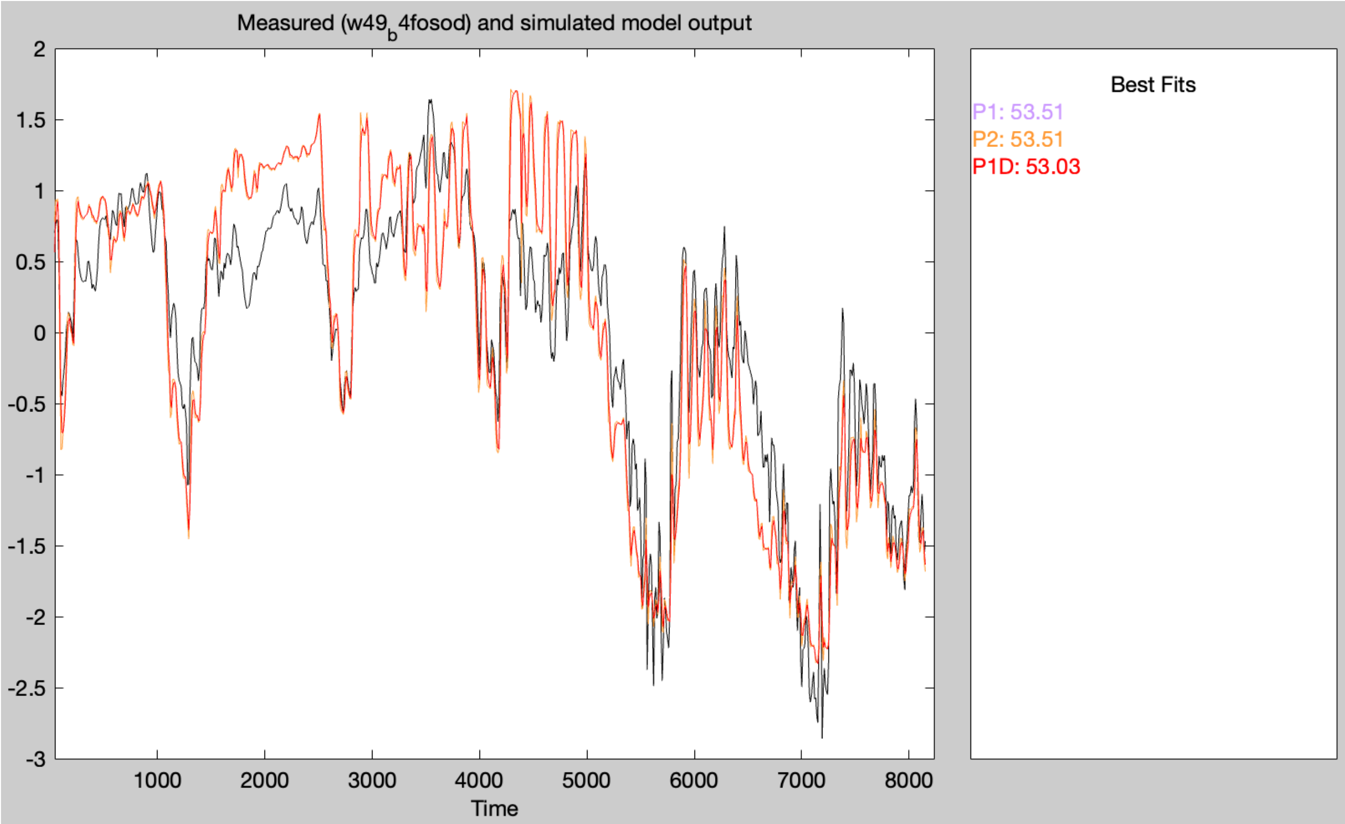

Figure 34 foso fitness index and plots of the best model outputs with FO4 as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

Table 17 tf_b4_foso transfer functions with FO4 as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P1	$1.1119 \cdot 10^{-3}$	$4.8569 \cdot 10^{-2}$	0.2235	0.2224	55.38	53.51
P2	$1.1119 \cdot 10^{-3}$	0.33052 and $4.7603 \cdot 10^{-4}$	0.224	0.2224	55.38	53.51
<u>P1D</u>	$1.1244 \cdot 10^{-3}$	9.63 and Td=0	0.2313	0.2296	54.66	53.03
P0	$1.1075 \cdot 10^{-3}$		0.2404	0.2398	53.66	52.39

P2D	$1.1402 \cdot 10^{-3}$	19.887 and 0.13695	0.2449	0.2425	53.4	51.56
P2DI	$5.7207 \cdot 10^{-7}$	$2.7228 \cdot 10^{-2}$ and 1.8597	0.7501	0.7373	18.74	22.96
P1DI	$6.3059 \cdot 10^{-7}$	$2.7708 \cdot 10^{-2}$ and Td=0	0.7413	0.7323	19.02	-17.92

2) tf_b5_foso first attempt with unscaled datasets

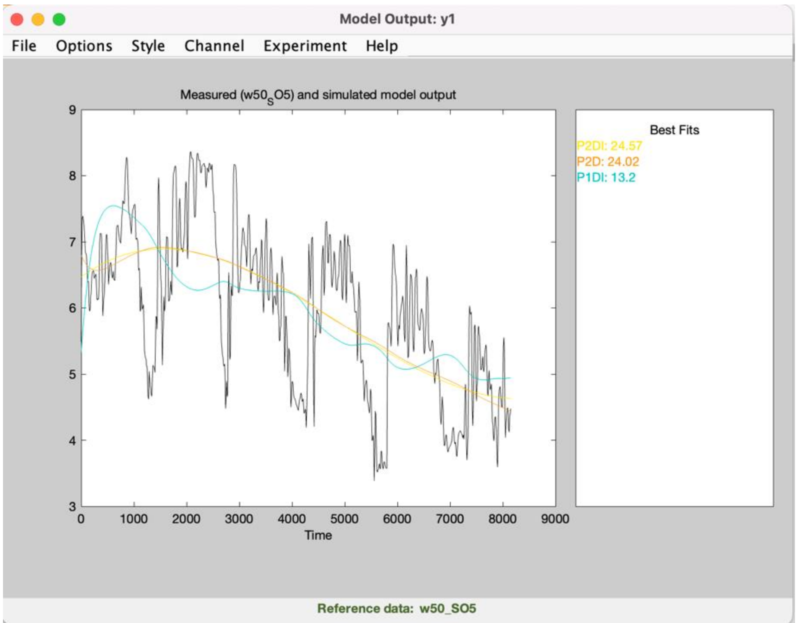

Figure 35 Fitness index and plots of all model output with FO5 and SO4 as input and SO5 as output with unscaled datasets

Notably, the documentation of this attempt was insufficient, and the parameters of the models could not be successfully determined. However, there is room for improvement in future attempts to better document the process and increase the likelihood of success in finding the parameters.

3) tf_b5_foso with week 49 as validation dataset

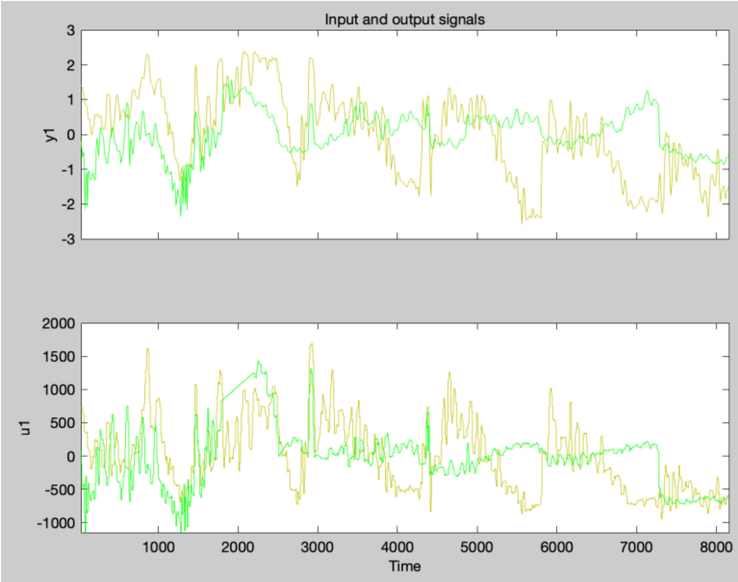

Figure 36 tf_b4_foso time plot for week 50 (yellow graph) and week 49 (green graph) with FO5 and SO4 as input and SO5 as output for channel 1 which is FO5

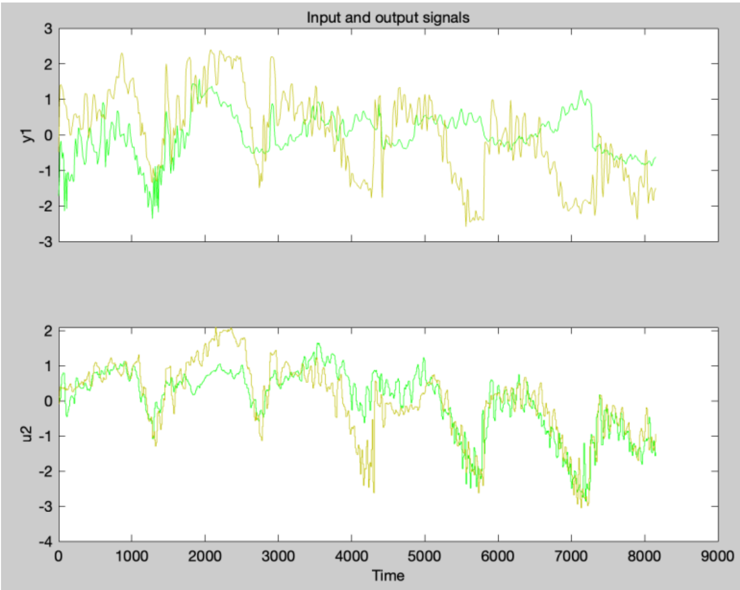

Figure 37 tf_b4_foso time plot for week 50 (yellow graph) and week 49 (green graph) with FO5 and SO4 as input and SO5 as output for channel 2 which is SO4

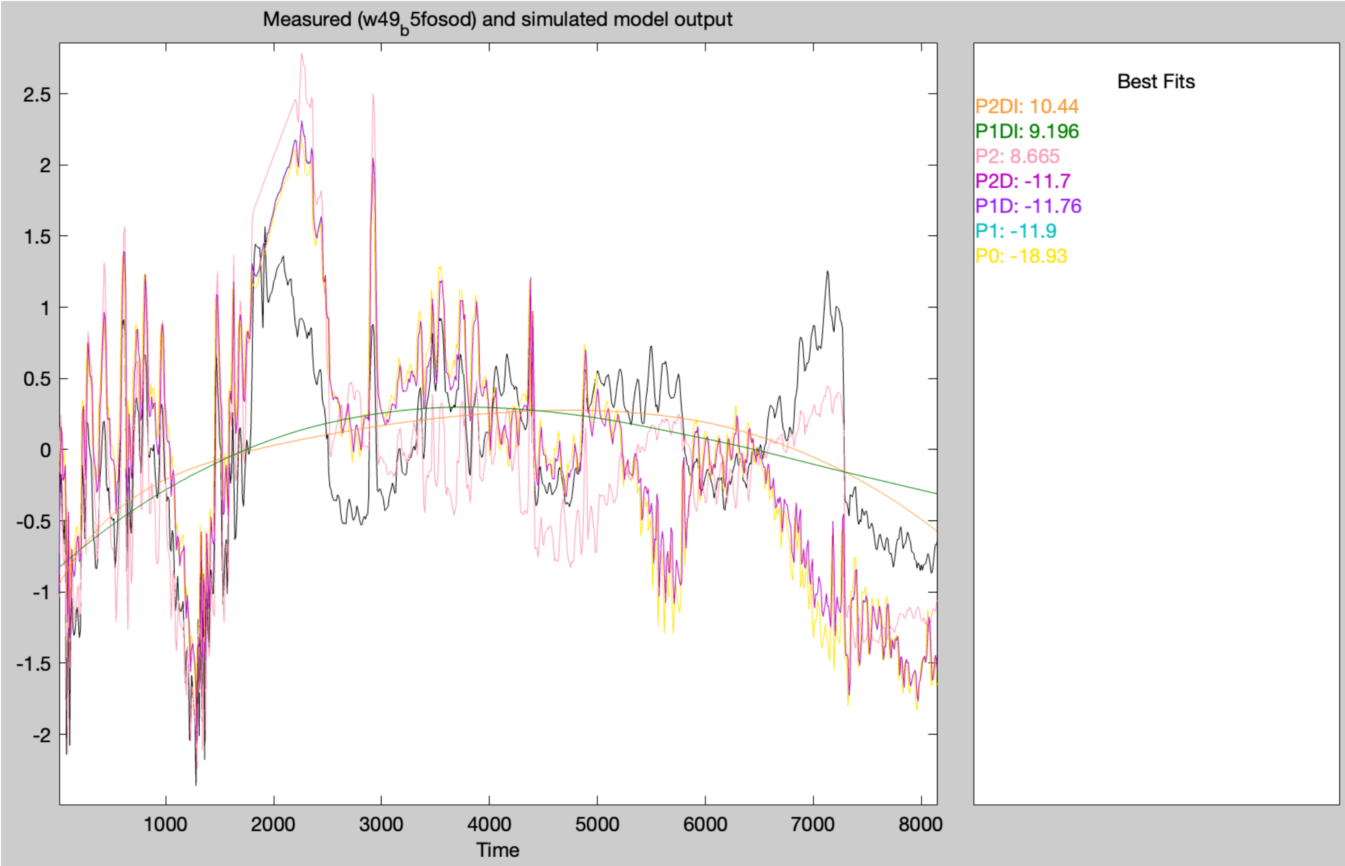

Figure 38 tf_b4_foso fitness index and plots of all model outputs with FO5 as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

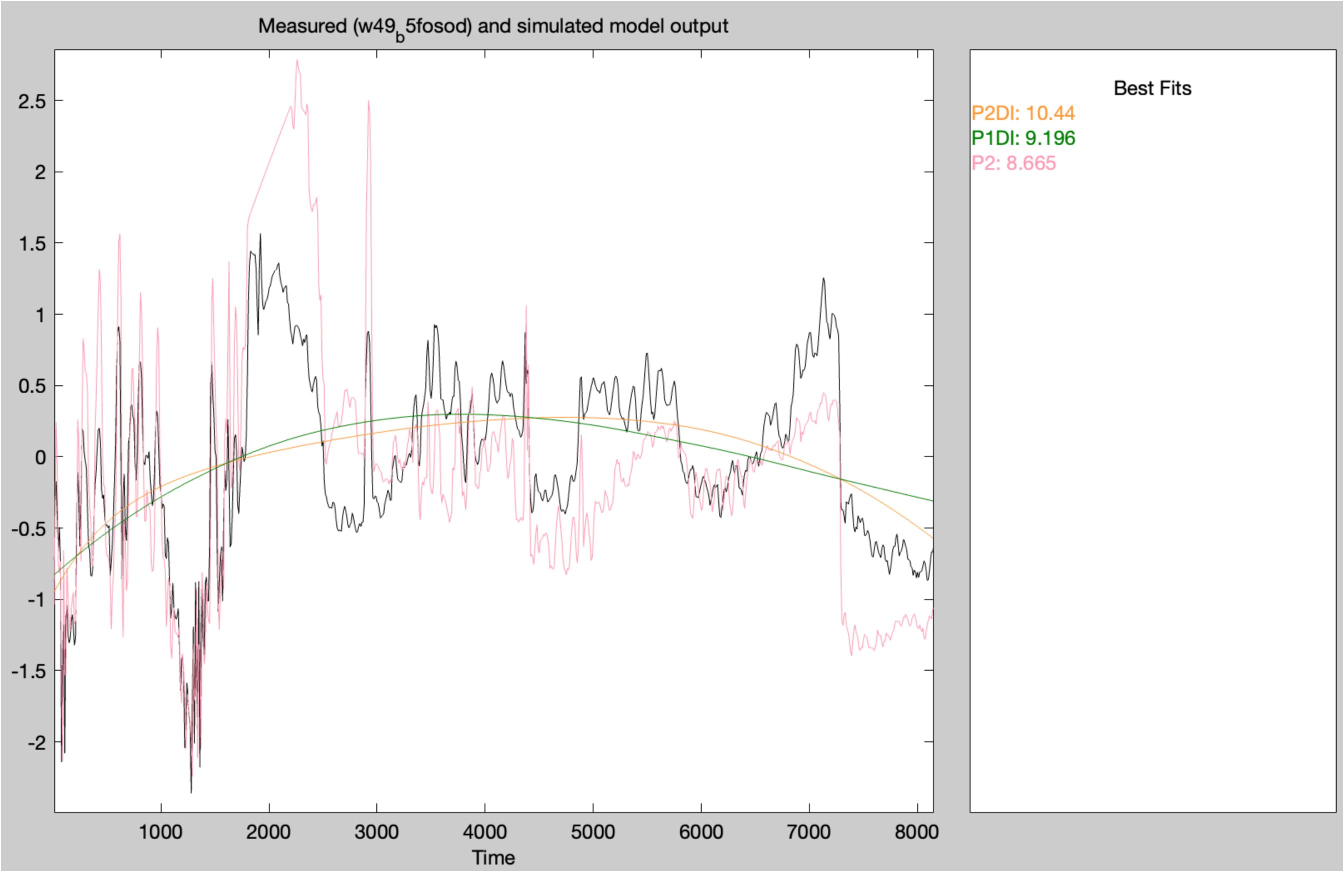

Figure 39 tf_b4_foso fitness index and plots of the best model outputs with FO5 as input and SO5 as output, and week 49 as validating data and week 50 as estimation data with unscaled datasets

Table 18 tf_b5_foso transfer functions with FO5 and SO4 as input and SO5 as output, and with week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P2DI	$-1.032\cdot 10^{-6}$	8256.3 and 5668, and Td=20.68	$7.4004\cdot 10^{-4}$	732.46 and 10 000 and Td=297.5	0.8662	0.837	24.83	10.44
P1DI	$-1.8985\cdot 10^{-7}$	10 000 and Td=15.83	$2.6828\cdot 10^{-4}$	7522.7 and Td=75.24	0.8694	0.8484	24.33	9.196
P2	$1.9998\cdot 10^{-3}$	$1\cdot 10^{-6}$ and 4.1599	-1.4081	10 000 and $1.5347\cdot 10^{-3}$	0.2914	0.2872	55.97	8.665

P2D	$1.288\cdot 10^{-3}$	$1.11166\cdot 10^{-3}$ and $1.1352\cdot 10^{-3}$	0.53597	1.8277 and 0.19284 and Td=0	0.1047	0.1026	73.68	-11.7
P1D	$1.2871\cdot 10^{-3}$	0.47415 and Td=0.35	0.53653	0.25632 and Td=4.24	0.1041	0.1026	73.68	-11.76
P1	$1.2936\cdot 10^{-3}$	$1.8903\cdot 10^{-3}$	0.53651	0.22468	0.1036	0.1026	73.69	-11.9
P0	$1.1741\cdot 10^{-3}$		0.61317		0.06651	0.06619	78.86	-18.93

The fitness index is not very high for tf_b4_foso and tf_b5_foso. That's why it is necessary to add more inputs. The flowrate of the wastewater is an important disturbance variable. There are fortunately online data for this variable, and the following transfer functions will have this input included. There would also be a good idea to add temperature as an input as well. However, this DV was not measured unfortunately. Adding SSin as an input will also be investigated later.

4) tf_b4_ffoso with unscaled datasets and with week 49 as validation dataset

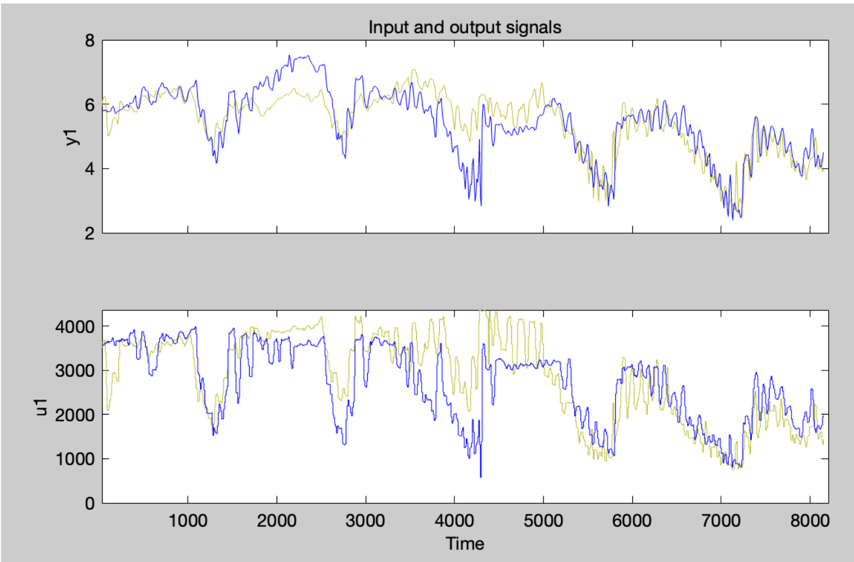

Figure 40 tf_b4_ffoso time plot for week 50 (blue graph) and week 49 (yellow graph) with FO4 and F as input and SO4 as output for channel 1 which is FO4 with unscaled datasets

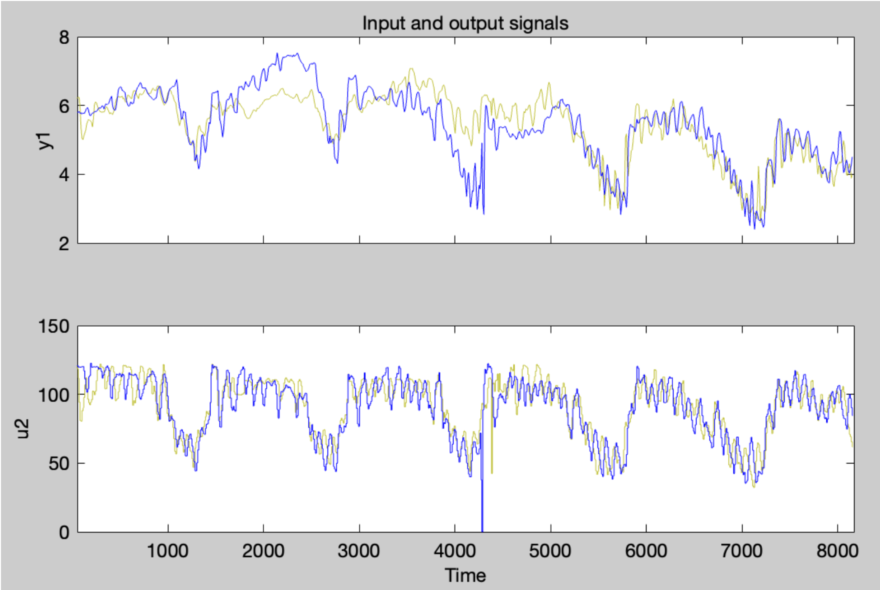

Figure 41 tf_b4_ffoso time plot for week 50 (blue graph) and week 49 (yellow graph) with FO4 and F as input and SO4 as output for channel 2 which is F with unscaled datasets

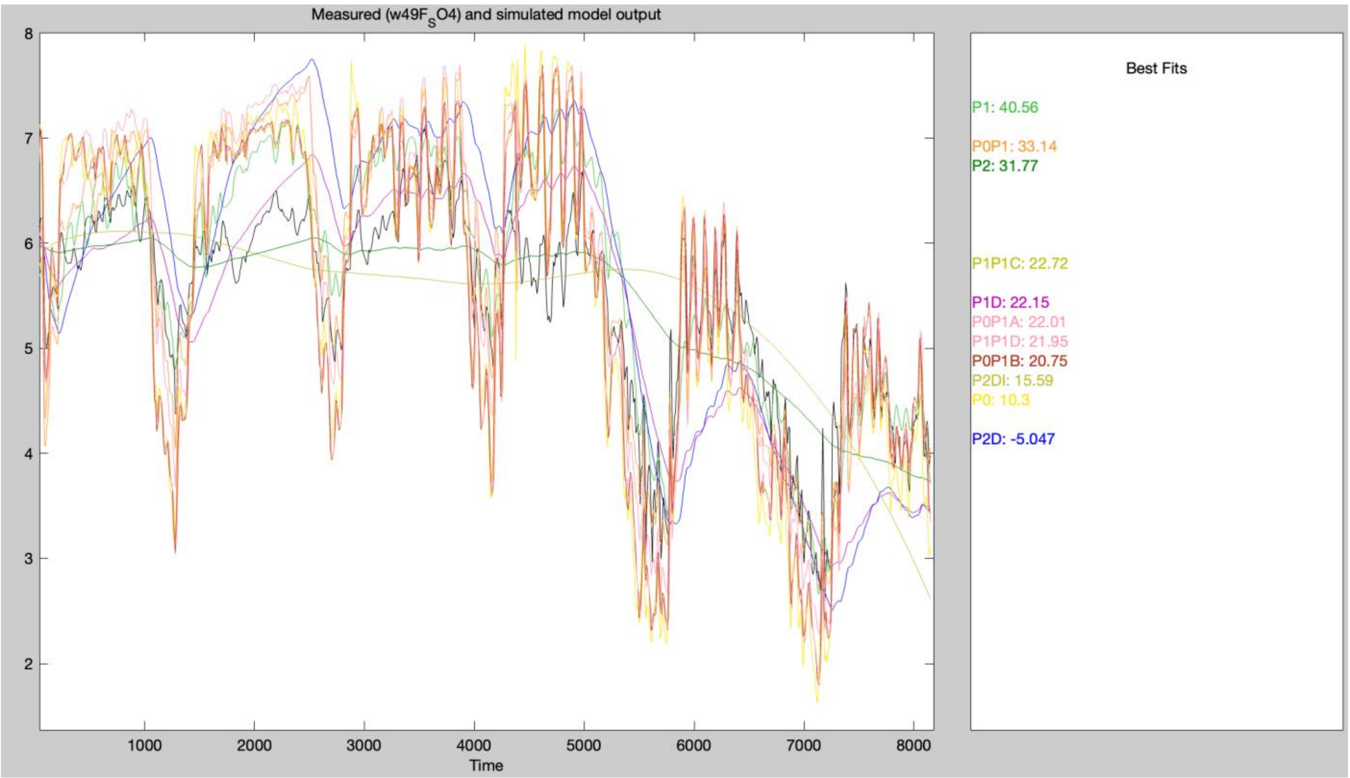

Figure 42 tf_b4_ffoso fitness index and plots of all model outputs with FO4 and F as input and SO4 as output, and week 49 as validating data and week 50 as estimation data with unscaled datasets

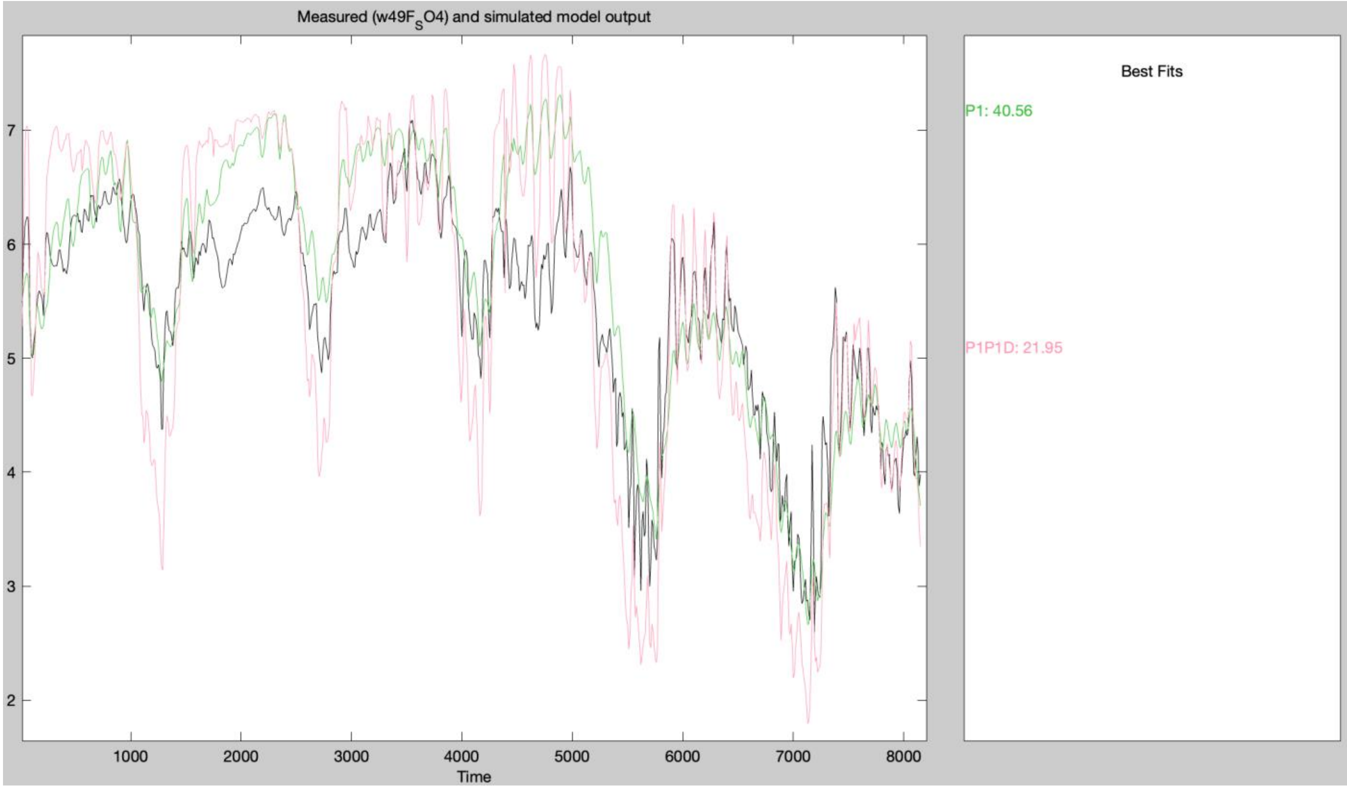

Figure 43 tf_b4_ffoso fitness index and plot of the best model outputs with FO4 and F as input and SO4 as output, and week 49 as validating data and week 50 as estimation data with unscaled datasets

Table 19 tf_b4_ffoso transfer functions with FO4 and F as input and SO4 as output with unscaled datasets, and with week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P1	$1.1696\cdot 10^{-3}$	436.69	$2.5519\cdot 10^{-2}$	16.472	0.2097	0.2067	56.98	40.56
P0P1	$1.1776\cdot 10^{-3}$		$2.639\cdot 10^{-2}$	488.18	0.2166	0.215	56.12	33.14
P2	$1.6903\cdot 10^{-3}$	2416.5 and $5.3445\cdot 10^{-4}$	$5.0435\cdot 10^{-2}$	10 000 and 10 000	0.6341	0.6188	25.56	31.77
P1P1C	$1.0372\cdot 10^{-3}$	0.39726	$2.788\cdot 10^{-2}$	20	0.5513	0.5433	20.25	22.72

P1D	$1.6786\cdot 10^{-3}$	428.35	$1.2834\cdot 10^{-2}$	10 000	0.4703	0.4612	35.74	22.15
P0P1A	$1.2372\cdot 10^{-3}$		$2.4169\cdot 10^{-2}$	200	0.3221	0.3198	46.49	22.01
<u>P1P1D</u>	$1.0558\cdot 10^{-3}$	0.9882	$2.7422\cdot 10^{-2}$	20	0.5859	0.5787	28.01	21.95
P0P1B	$1.0605\cdot 10^{-3}$		$2.7273\cdot 10^{-2}$	20	0.57	0.5658	28.82	20.75
P2DI	$9.2955\cdot 10^{-6}$	39.261 and 10 000	$-2.4033\cdot 10^{-4}$	0.76605 and 9625.5, and Td=2.71	1.272	1.229	-4.907	15.59
P0	$1.2831\cdot 10^{-3}$		$2.0255\cdot 10^{-2}$		0.6489	0.6458	23.96	10.3
P2D	$1.8769\cdot 10^{-3}$	300.98 and 0.28751	$3.7147\cdot 10^{-3}$	229.26 and 0.034272, Td=245.12	0.493	0.4787	34.53	-5.047
P1DI	$-6.9057\cdot 10^{-7}$	1897.2, Td=63.26	$1.0714\cdot 10^{-4}$	2738.6	0.75	0.7319	19,04	-171

In (Table 19 tf_b4_ffoso transfer functions with FO4 and F as input and SO4 as output) the P1 model gave the best fitness index. However, this model had a very high time constant (Tp1=436,69) for the FO4 input. This is very unrealistic, since 436,69 minutes (which is the sampling time) is too long. The aeration rate will be between 30 minuets and 90 minutes for F and SO4. That's why it was necessary to reduce the time constant Tp1 by setting limits. This will make the model more realistic and unfortunately reduce the fitness index. Model structure with integrator did not give good results, fitness index was under 0. That's why this was not included.

5) tf_b4_ffoso with scaled datasets and with week 49 as validation dataset

After some new information came to light about the preprocessing, there were made some adjustments to improve the results. Rescaling the datasets with the remove means function in system identification significantly improved the results. Which will be the following time plots, model outputs and transfer functions for tf_b4_ffoso and so on.

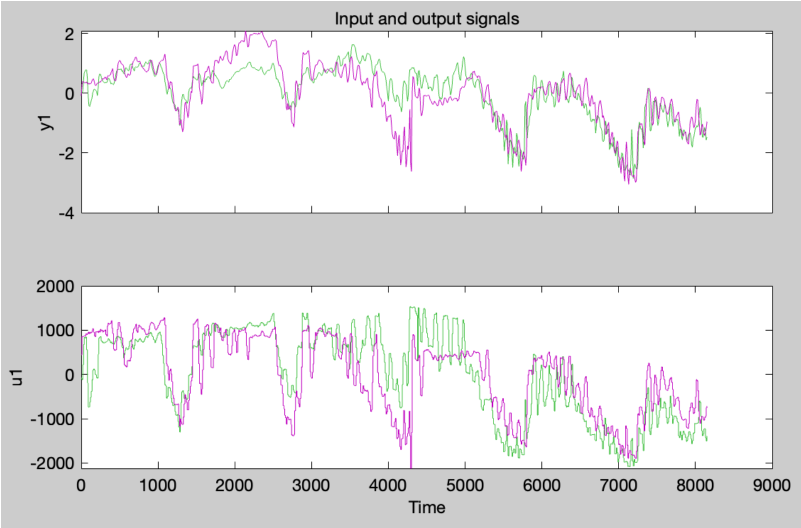

Figure 44 tf_b4_ffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO4 and F as input and SO4 as output for channel 1 which is FO4

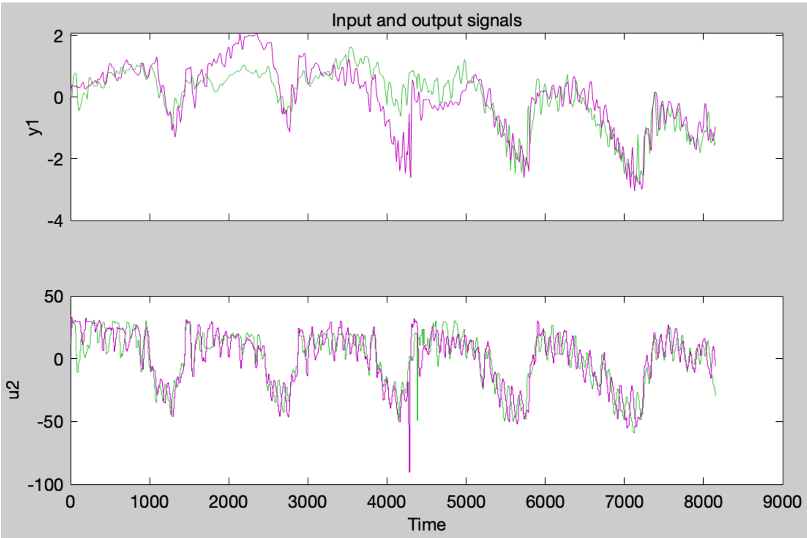

Figure 45 tf_b4_ffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO4 and F as input and SO4 as output for channel 2 which is F

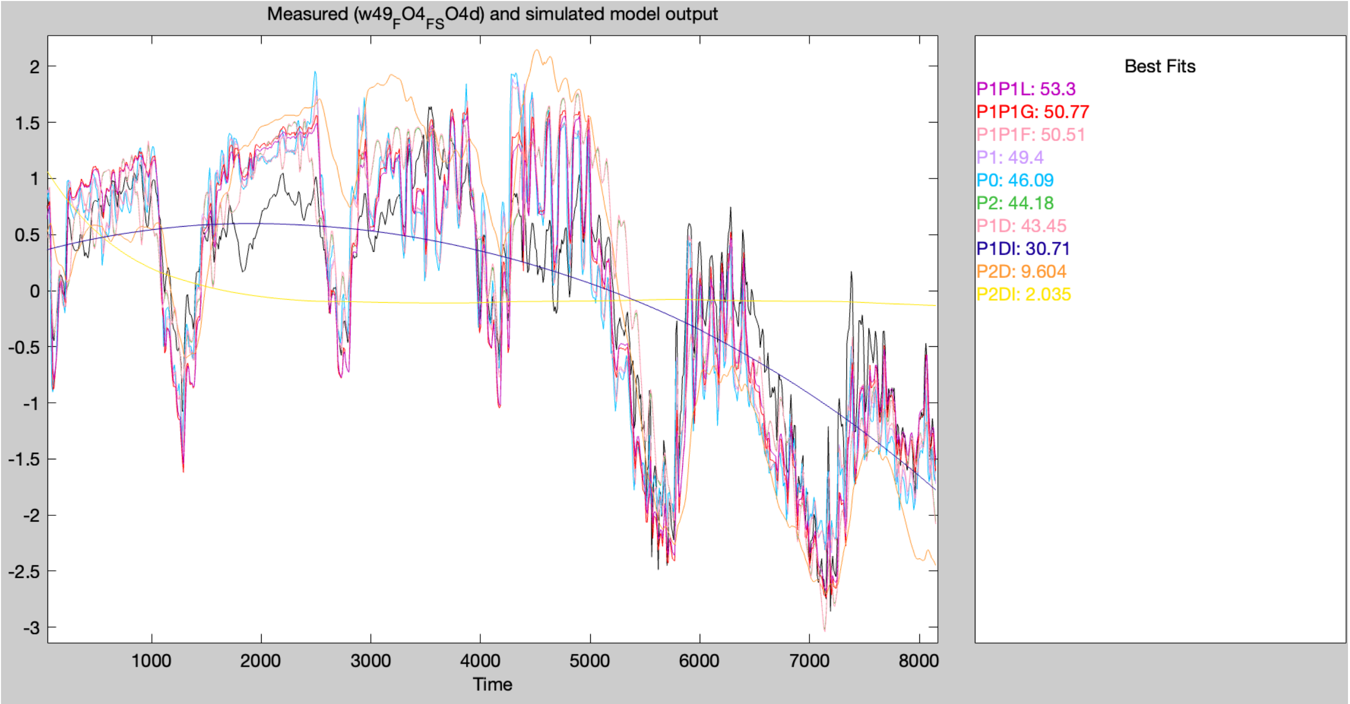

Figure 46 tf_b4_ffoso fitness index and plot of the best model outputs with FO4 and F as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

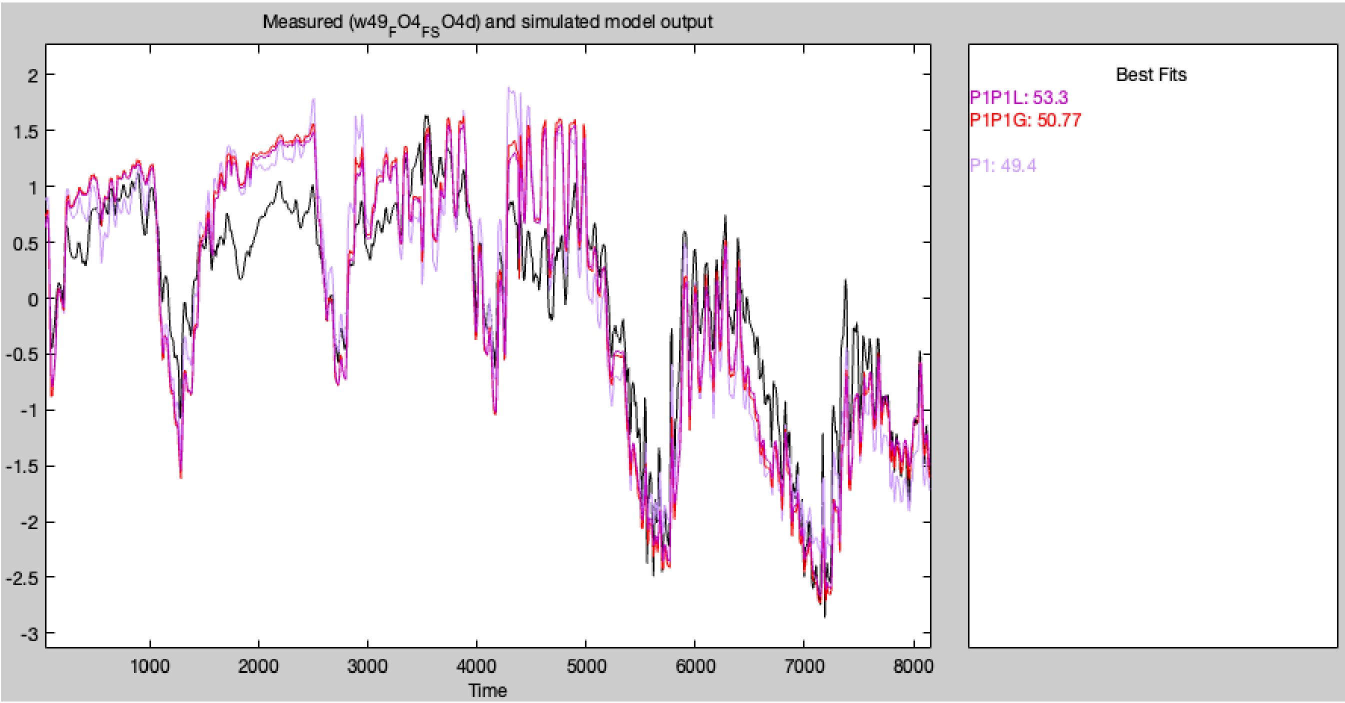

Figure 47 tf_b4_ffoso fitness index and plot of the best model outputs with FO4 and F as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

Table 20 tf_b4_ffoso transfer functions with FO4 and F as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
<u>P1P1L</u>	$1.006 \cdot 10^{-3}$	6.2589	$1.76717 \cdot 10^{-2}$	240	0.2055	0.2025	57.42	53.3
P1P1G	$1.0010607 \cdot 10^{-3}$	0.17136	$1.5918 \cdot 10^{-2}$	240	0.2274	0.2252	55.1	50.77
P1P1F	$1.1975 \cdot 10^{-3}$	0.21143	$2.639 \cdot 10^{-2}$	$9.5452 \cdot 10^{-3}$	0.2214	0.2192	55.69	50.51
P1	$1.2235 \cdot 10^{-3}$	0.32173	$-5.2662 \cdot 10^{-3}$	14.591	0.2203	0.2182	55.8	49.4
P0	$1.2739 \cdot 10^{-3}$		$-8,4902 \cdot 10^{-3}$		0.2267	0.2256	55.05	46.09
P2	$1.2694 \cdot 10^{-3}$	581.81 and $1.2894 \cdot 10^{-3}$	$2.6547 \cdot 10^{-2}$	8.7285 and 8.7112	0.2301	0.2267	54.94	44.18
P1D	$1.2873 \cdot 10^{-3}$	588.26 and Td=0	$2.6278 \cdot 10^{-2}$	9.7725 and Td=7.28	0.2296	0.2263	54.99	43.45
P1DI	$1.2922 \cdot 10^{-6}$	30769 and Td=127.56	$-8.6827 \cdot 10^{-4}$	$1.003 \cdot 10^{-6}$ and Td=19.93	0.7053	0.6882	21.5	30.71
P2D	$2.3345 \cdot 10^{-3}$	341.45 and 0.24376 and Td=0	$-7.6652 \cdot 10^{-2}$	697.47 and 0.62596 and Td=187.88	0.296	0.2902	49.02	9.604
P2DI	$3.3455 \cdot 10^{-8}$	2.9987 and 782.38 and Td=0	$-7.4046 \cdot 10^{-7}$	0.66601 and 4.1836 and Td=2.18	0.6812	0.6582	23.23	2.035

Unfortunately, there was not made good enough documentation of how P0P1, P0P1A, P0P1B, P1P1C, P1P1D was made in the process models tab. However, the values are still available and there is probably a way of recreating them. However, it is decided that using new models was better for time limited reasons. P1P1D, P1P1L and P1PG have very similar structures and can be regarded as almost the same method.

6) tf_b5_ffoso reducing the peek somewhat with interpolation

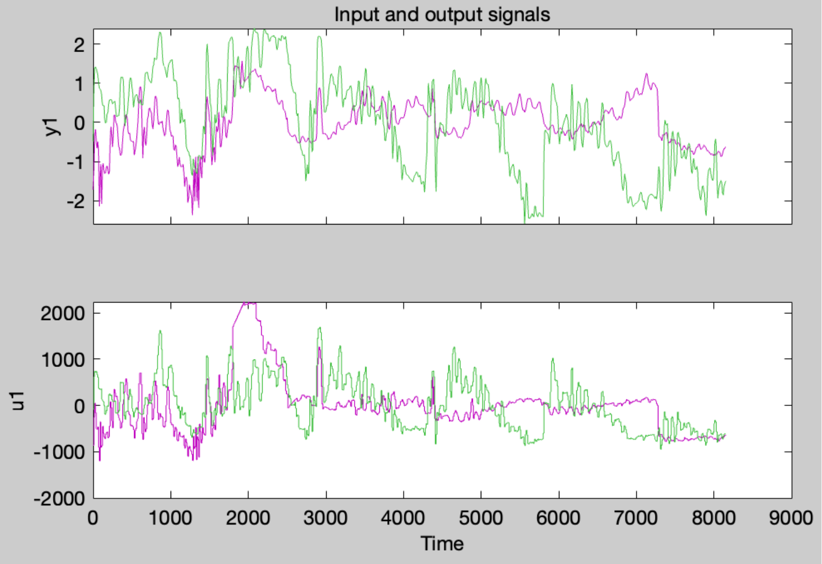

Figure 48 tf_b5_ffoso time plot for week 50 (green graph) and week 49 (purple graph) with FO5, SO4 and F as input and SO5 as output for channel 1 which is FO5

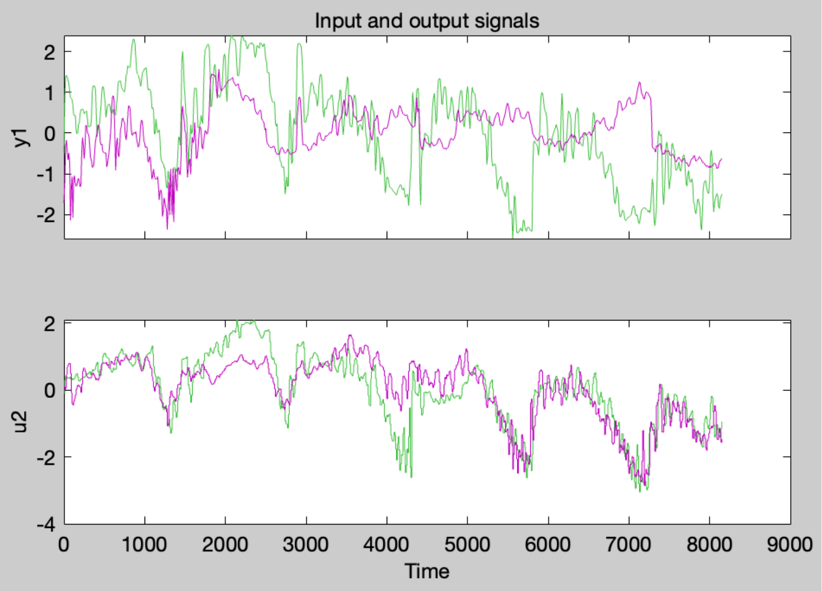

Figure 49 tf_b5_ffoso time plot for week 50 (green graph) and week 49 (purple graph) with FO5, SO4 and F as input and SO5 as output for channel 2 which is SO5

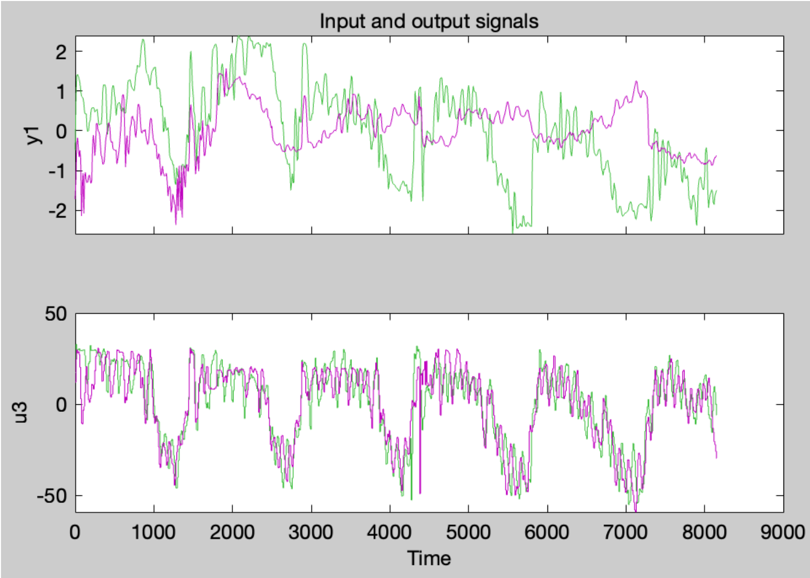

Figure 50 tf_b5_ffoso time plot for week 50 (green graph) and week 49 (purple graph) with FO5, SO4 and F as input and SO5 as output for channel 3 which is F

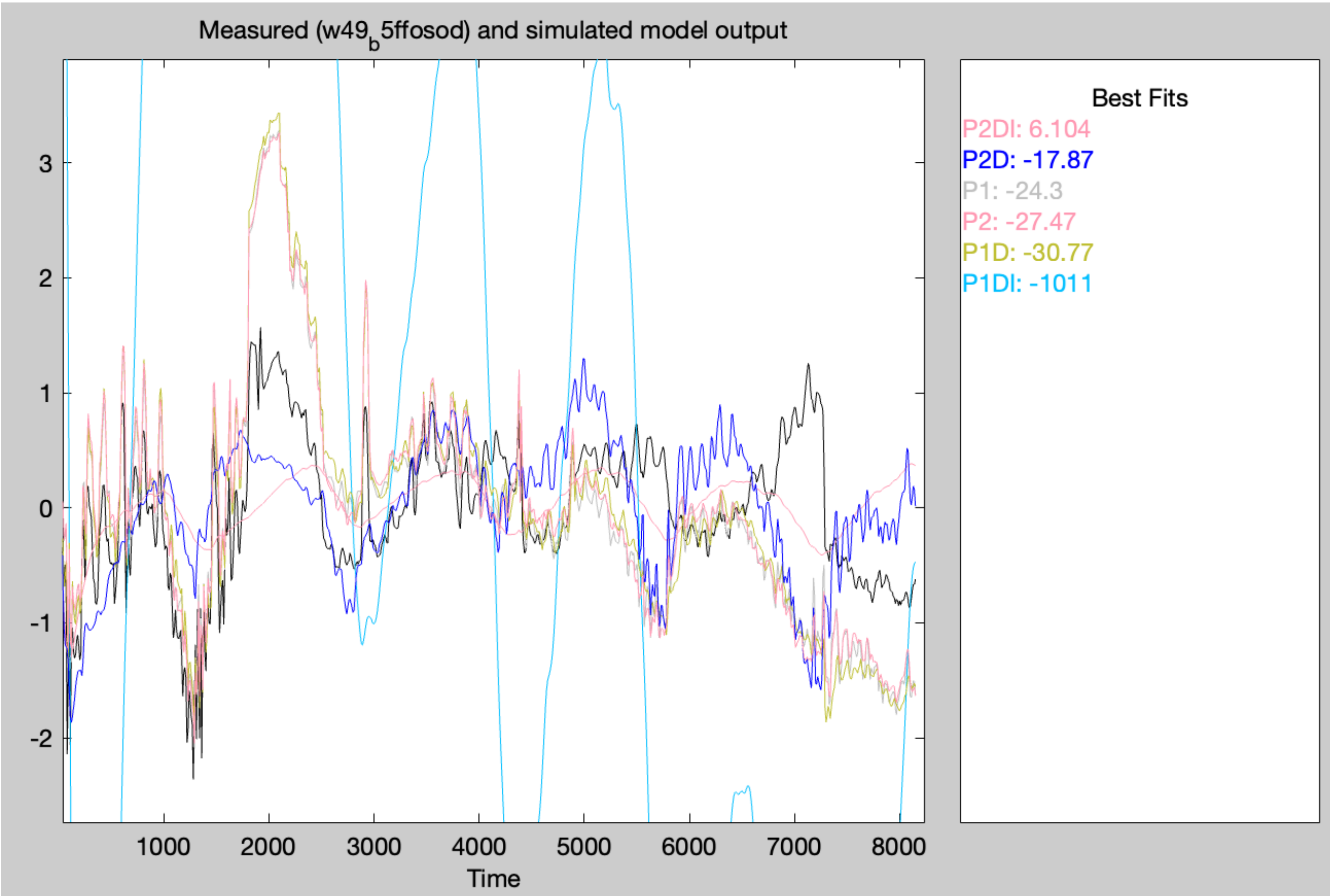

Figure 51 f_b5_ffoso fitness index and plot of all the model outputs with FO5, SO4 and F as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

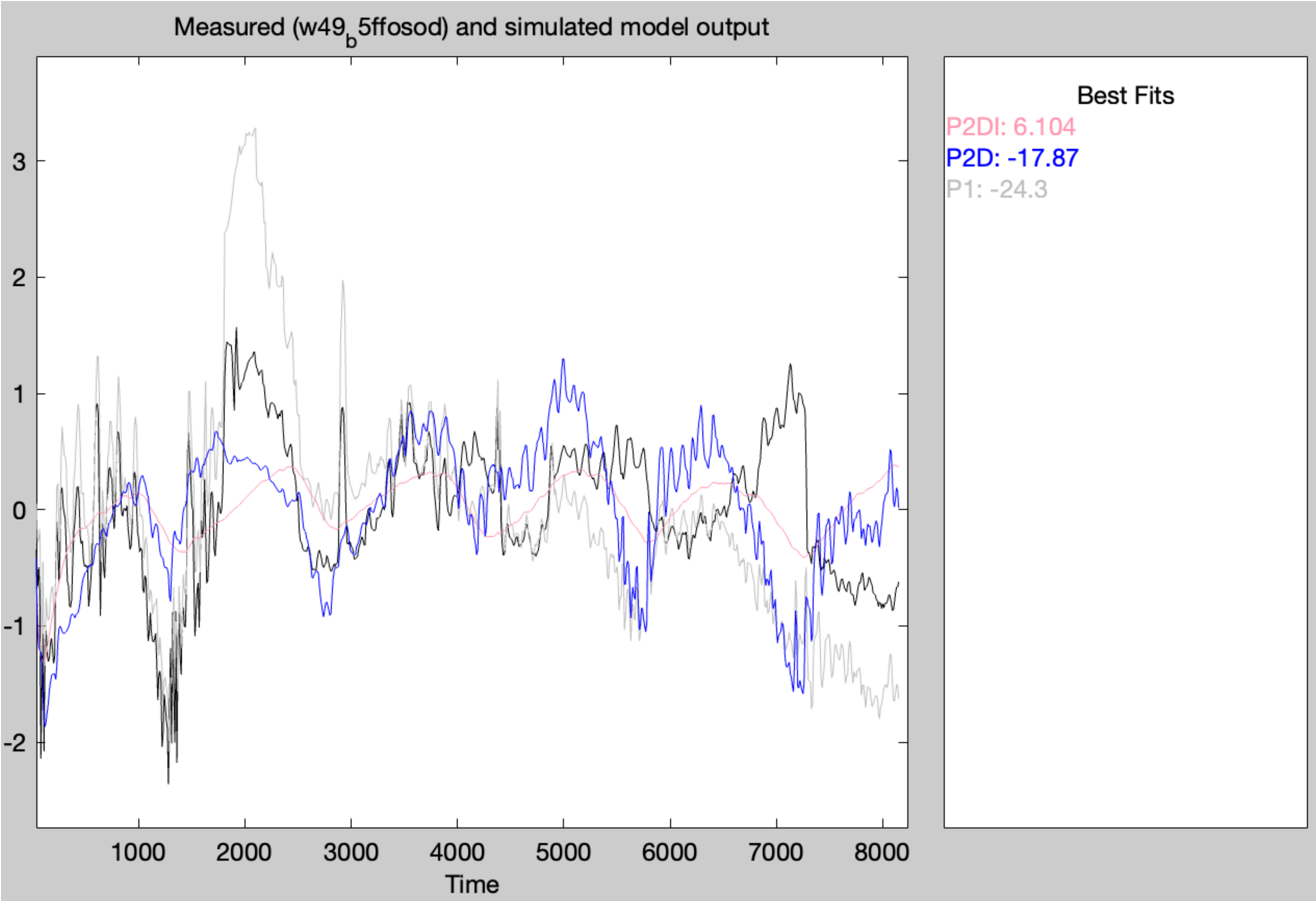

Figure 52 f_b5_ffoso fitness index and plot of the best model outputs with FO5, SO4 and F as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

Table 21 tf_b5_ffoso transfer functions with FO5, SO4 and F as input and SO5 as output, and week 49 as validating data and week 50 as estimation data with somewhat reduced peek

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P2DI	$-4.2819\cdot 10^{-7}$+/-$1.0443\cdot 10^{-2}$	125.57+/-$2.8519\cdot 10^{7}$ and 10 000+/-$7.7856\cdot 10^{5}$ and Td=0.17 +/-$2.077\cdot 10^{5}$	$4.8388\cdot 10^{-4}$	118.27+/-28482 and 61.586+/-6707.4 and Td=0.64 +/- $1.3626\cdot 10^{5}$	$5.33\cdot 10^{-5}$+/-$8.8882\cdot 10^{-3}$	0.96707 +/-$7.7991\cdot 10^{5}$ and 1.9061+/-$7.8122\cdot 10^{5}$ and Td=1.9061+/-836.05	0.8316	0.7899	26.98	6.104
P2D	$-1.2363\cdot 10^{-2}$+	10000 +/-8194.3 and 9.2009	0.56725 +/-	3.3575 +/-774.52 and 4.1315+/-536.72 and	0.24731 +/-0.4192	6216.7 +/-10552 and	0.4567	0.437	45.69	-17.87

	/- 1.1085$\cdot 10^{-2}$	+/- 249.81 and Td=28.67 +/- 242.8	0.056691	Td=0+/- 340.69		3.695 +/- 631.39 and Td=0+/- 615.36				
P1	1.2999 $\cdot 10^{-3}$+/- 3.2782$\cdot 10^{-5}$	1.9252 +/- 3.5962	0.53373 +/- 0.018902	5.0717+/- 1.5801	-0.20441 +/- 0.96307	62119+/- 3.0987$\cdot 10^{5}$	0.108	0.1064	73.2	-24.3
P2	1.2505$\cdot 10^{-3}$ +/- 939.74	3.277 $\cdot 10^{-4}$+/- 5.4236 $\cdot 10^{10}$	0.50383 +/- 11.774	14.365+/- 331.45 and 1.1089$\cdot 10^{-5}$+/- 4.9162	4.6235$\cdot 10^{-3}$+/- 10047	3.1555$\cdot 10^{-6}$+/- 4.2971$\cdot 10^{9}$ and 1.8239$\cdot 10^{-4}$+/- 2.403$\cdot 10^{11}$	0.1025	0.1003	73.98	-27.47
P1D	1.3521$\cdot 10^{-3}$+/- 88.338	5.2641 $\cdot 10^{-3}$+/- 9.2054 $\cdot 10^{11}$ and Td=0+/- 42921	0.55851 +/- 2.0383$\cdot 10^{-2}$	41.858+/- 7.142 and Td=0+/- 4.7772	1.9321 $\cdot 10^{-3}$+/- 557.87	8.1142$\cdot 10^{-4}$+/- 2.5221$\cdot 10^{12}$ and Td=400+/- 2.0013$\cdot 10^{-3}$	0.1172	0.1146	72.18	-30.77
P1DI	-1.2598$\cdot 10^{-5}$+/- 9.3445$\cdot 10^{-6}$	109.3+/- 137.39 and Td=0+/- 188.7	-6.03$\cdot 10^{-3}$+/- 7.7486$\cdot 10^{-3}$	720.71+/- 1285.8 and Td=2.62+/- 812.4	7.61667 $\cdot 10^{-4}$+/- 2.0212$\cdot 10^{-4}$	20.556+/- 110.52 and Td = 0 +/- 96.769	22.24	21.44	-280.4	-1011

7) tf_b5_ffoso and with week 49 as validation dataset

Table 22 tf_b5_ffoso transfer functions with FO5, SO4 and F as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %

P1										-11

8) tf_b4_sffoso with week 49 as validation dataset

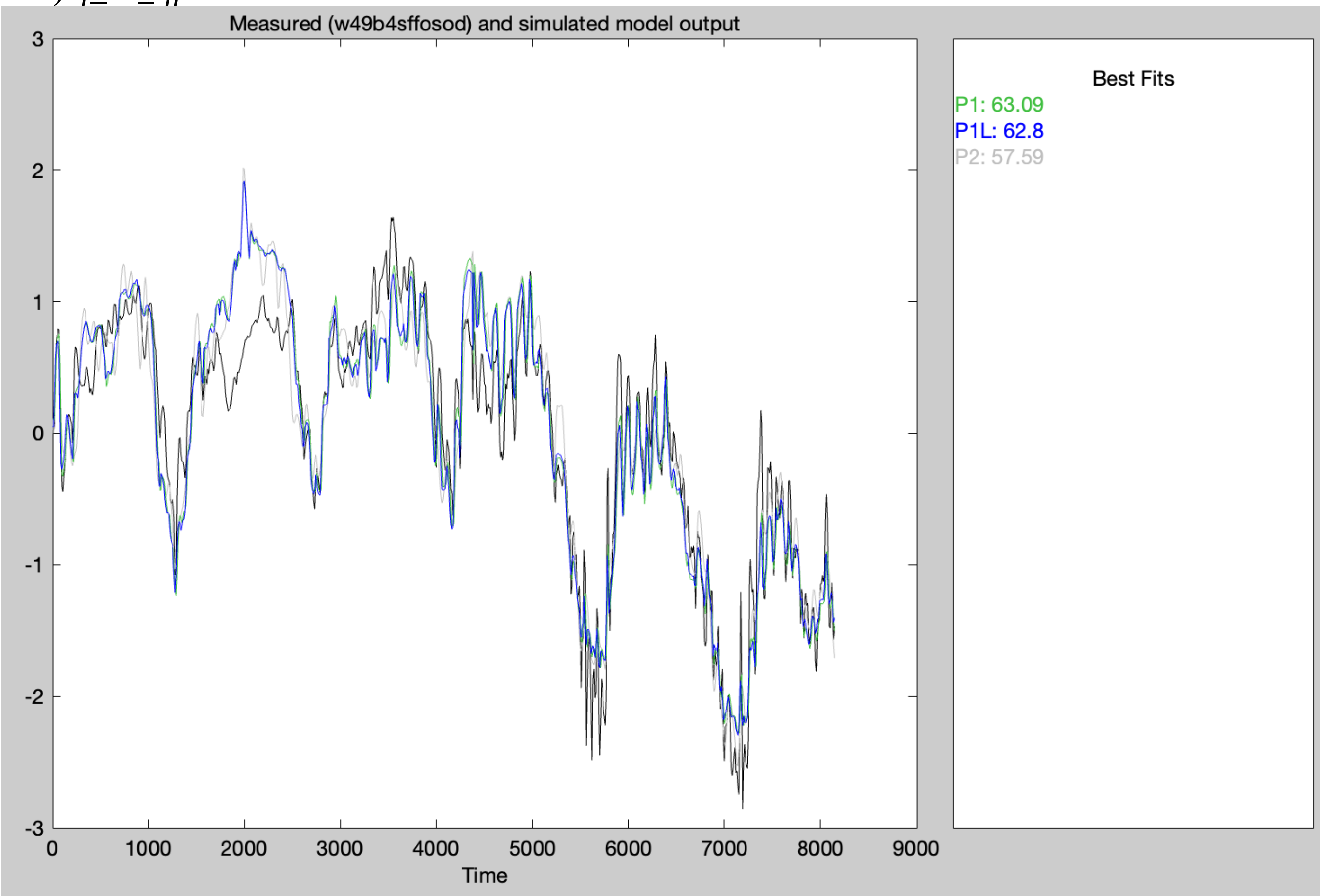

Figure 53 f_b5_ffoso fitness index and plot of the best model outputs with FO4, F and SSin as input and SO4 as output, and week 49 as validating data and week 50 as estimation data

Table 23 tf_b4_sffoso transfer functions with SO4s, Fs and SSins as input and SO5s as output, and week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P1	6.2508 · 10^{-4}+/-	1.3666· 10^{-7}+/-	2.0606· 10^{-2}+/-	129.69 +/- 12.846	5.3183· 10^{-3} +/-	= 53.59 +/- 8.5287	0.1174	0.1157	67.81	63.09

	1.4353	$3.1868 \cdot 10^{7}$	$1.3224 \cdot 10^{-3}$		$2.2563 \cdot 10^{-4}$					
P1L	$5.8591 \cdot 10^{-4}$ +/- $3.0671 \cdot 10^{-5}$		$2.2703 \cdot 10^{-2}$+/- $1.2525 \cdot 10^{-3}$	129.27 +/- 11.679	$5.5161 \cdot 10^{-3}$+/- $2.21 \cdot 10^{-4}$	55.843 +/- 8.5198	0.1181	0.1167	67.68	62.8
P2	$5.2681 \cdot 10^{-4}$+/- $4.9312 \cdot 10^{-3}$	214.23 +/- 2000.7 and $1.6529 \cdot 10^{-3}$ +/- 19.642	$2.7703 \cdot 10^{-2}$+/- $1.7368 \cdot 10^{-3}$	18.064 +/- 5.1176 and 3.8363 +/- 3.249	$5.7826 \cdot 10^{-3}$+/- 89.74	49.026 +/- $7.6083 \cdot 10^{5}$ and $7.2962 \cdot 10^{-7}$+/- $9.9379 \cdot 10^{-4}$	0.1506	0.1473	63.68	57.59

9) tf_b5_sffoso, accidently used week 50 as validation data and week 49 as estimation data and documented it

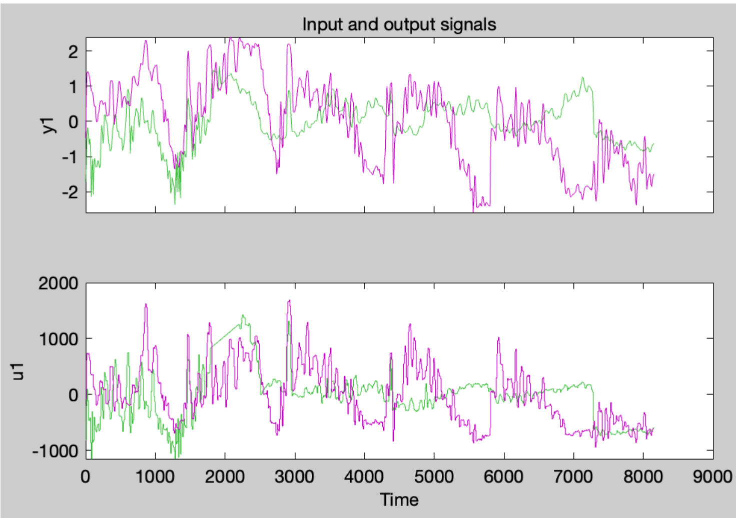

Figure 54 tf_b5_sffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO5, SO4, F and SSin as input and SO5 as output for channel 1 which is FO5

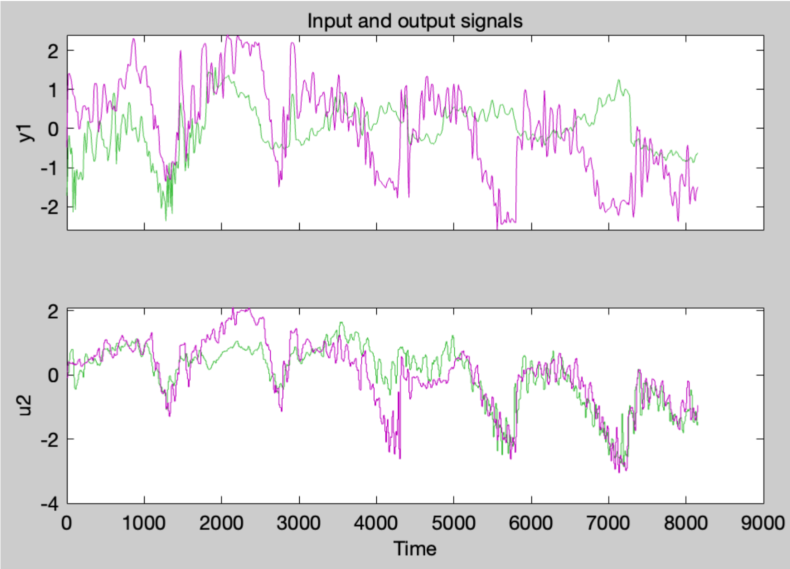

Figure 55 b5_sffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO5, SO4, F and SSin as input and SO5 as output for channel 2 which is SO4

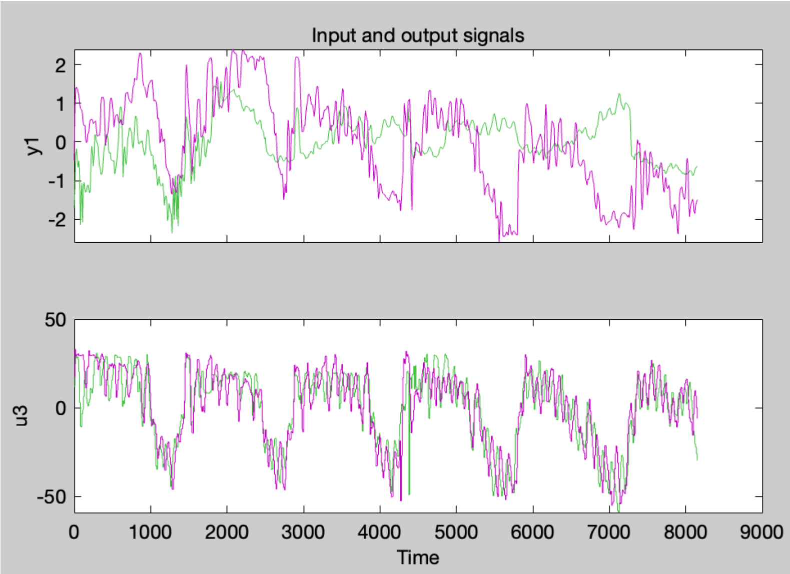

Figure 56 b5_sffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO5, SO4, F and SSin as input and SO5 as output for channel 3 which is F

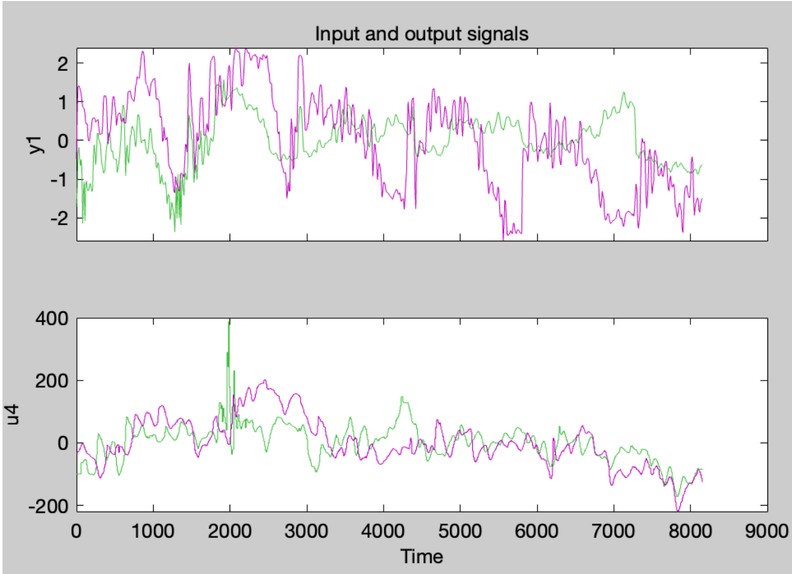

Figure 57 b5_sffoso time plot for week 50 (purple graph) and week 49 (green graph) with FO5, SO4, F and SSin as input and SO5 as output for channel 4 which is SSin

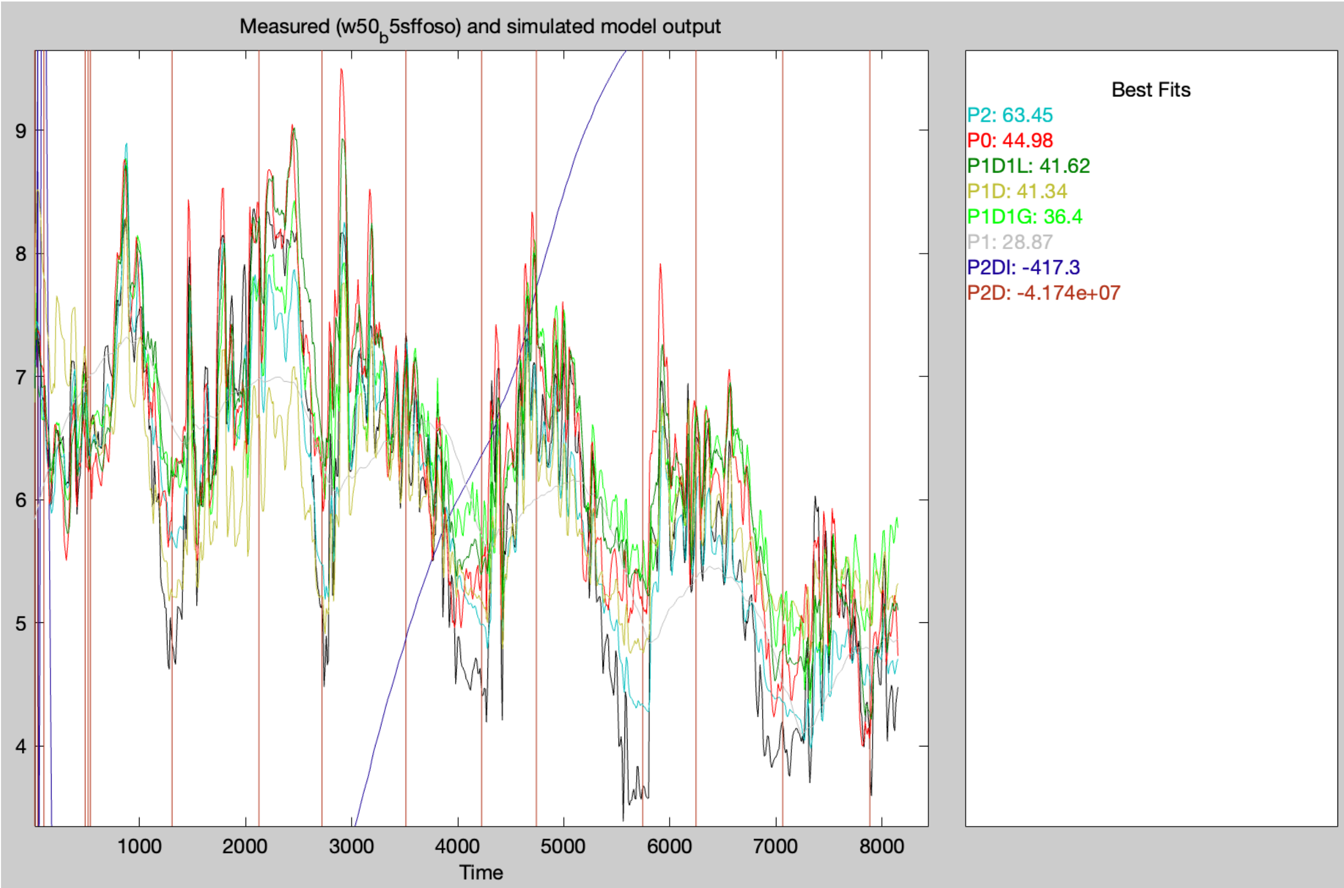

Figure 58 f_b5_sffoso fitness index and plot of all the model outputs with FO5, SO4, F and SSin as input and SO5 as output, and week 49 as estimation data and week 50 as validation data

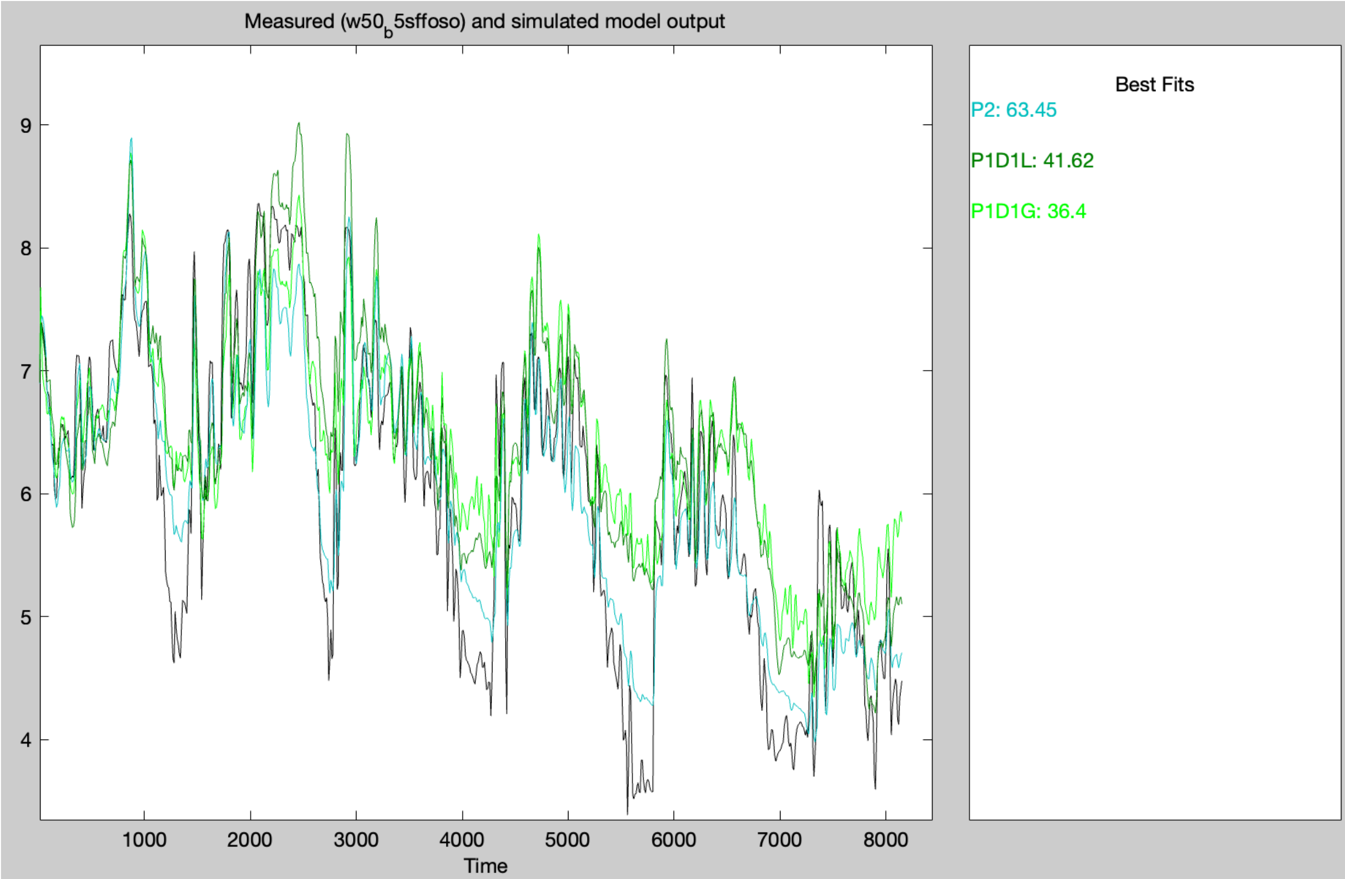

Figure 59 f_b5_sffoso fitness index and plot of the best model outputs with FO5, SO4, F and SSin as input and SO5 as output, and week 49 as validating data and week 50 as validation data

Table 24 tf_b5_sffoso transfer functions with FO5, SO4, F and SSin as input and SO5 as output, and week 49 as estimation data and week 50 as validation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P2	$1.3295 \cdot 10^{-3}$+/-3.5275	$1 \cdot 10^{-6}$ +/- 11.56 and 10.336 +/- 27416	-433.36 +/- $3.0844 \cdot 10^{7}$	$2.2872 \cdot 10^{6}$+/-$1.6352 \cdot 10^{11}$ and 10000 +/- $3.8306 \cdot 10^{6}$	$2.8454 \cdot 10^{-2}$+/-$3.7234 \cdot 10^{-2}$	663.37 +/-1025 and 1.1221 +/- 72.862	$1.423 \cdot 10^{-2}$+/-2.5706	74.473 +/-1576 and 10811+/-$1.9834 \cdot 10^{6}$	0.1351	0.1287	43.79	63.45

P1D1L	$1.0129 \cdot 10^{-3}$+/-$6.1252 \cdot 10^{-5}$	1.1193 +/- 4262.2	$-4.7537 \cdot 10^{-2}$ +/- $4.4722 \cdot 10^{-2}$	10+/-116.55 and Td = 0+/-136.17	$1.2583 \cdot 10^{-2}$+/-$2.1346 \cdot 10^{-3}$	178.42 +/- 51.061 and Td=55.39+/-29.847	$7.6844 \cdot 10^{-3}$ +/- $4.9853 \cdot 10^{-4}$	20+/-6.9048 and Td = 2.69+/-5.2847	0.4461	0.4332	-3.136	41.62
P1D	$1.1744 \cdot 10^{-3}$+/-$3.241 \cdot 10^{-5}$	5.122 +/- 7.9659 and Td = 0+/-16.145	0.48989 +/- 0.16627	1202.8 +/- 329.54 and Td=255.21 +/- 45.907	$4.6955 \cdot 10^{-3}$+/-$8.0957 \cdot 10^{-4}$	5.2176 +/- 17.712 and Td=204.03 +/- 22.159	0.18876 +/- 6.002	$3.925 \cdot 10^{5}$+/-$1.2606 \cdot 10^{7}$ and Td=0+/-623.19	0.1063	0.1022	49.91	41.34
P1D1G	$9.0156 \cdot 10^{-4}$+/-$1.657 \cdot 10^{-4}$	1.5335 +/- 363.72 and Td=5.88 +/- 977.05	-0.39618 +/- 2053.4	$2.4144 \cdot 10^{-3}$+/-$1.7431 \cdot 10^{11}$ and Td=10 +/- $6.2986 \cdot 10^{9}$	$4.1911 \cdot 10^{-2}$+/-0.17876	300+/-1301.8 and Td=0 +/-12.4	$6.553 \cdot 10^{-3}$+/-$5.5857 \cdot 10^{-4}$	16.981 +/- 7.4119 and Td=0+/-6.9301	0.2863	0.2753	17.78	36.4
P1	$-1.9593 \cdot 10^{-2}$+/-145.91	$3.8541 \cdot 10^{5}$+/-$2.8706 \cdot 10^{9}$	-0.65356 +/- $4.5676 \cdot 10^{5}$	$1.1345 \cdot 10^{6}$+/-$4.5976 \cdot 10^{5}$	$6.828 \cdot 10^{-2}$+/-$5.0442 \cdot 10^{-2}$	867.4 +/- 537.93	0.1786 +/- 1984.1	$3.925 \cdot 10^{5}$+/-$4.3599 \cdot 10^{9}$	0.3885	0.3772	3.757	28.87
P2DI	$1.4777 \cdot 10^{-6}$ +/- $3.7546 \cdot 10^{-7}$	1.7482 +/- 59199 and 2.6295 +/- 32998 and Td = 0.68+/-26213	$5.2419 \cdot 10^{-7}$ +/- $9.9901 \cdot 10^{-6}$	$5.2419 \cdot 10^{-7}$+/-$9.9901 \cdot 10^{-6}$ and $3.8975 \cdot 10^{-3}$+/-$6.2301 \cdot 10^{-8}$	$1.2164 \cdot 10^{-7}$ +/- $6.4476 \cdot 10^{-6}$	18.117 +/- 2291.6 and 73.218 +/- 53.756 and Td=24.39 +/- 13393	3.7716e-08 +/- 9.5309e-08	5.3546 +/- 2247.7 and 8.7439 +/- 1588.9	0.3945	0.3683	4.901	-417.3
P2D	$3.4512 \cdot 10^{-3}$ +/- $8.758 \cdot 10^{7}$	38418 +/- $9.8144 \cdot 10^{-14}$ and 9.8047 +/- $2.765 \cdot 10^{11}$	0.1213 +/- $2.9239 \cdot 10^{7}$	431.95 +/- $1.1927 \cdot 10^{11}$ and 63.69 +/- 5.4883	$3.9617 \cdot 10^{6}$ +/- $5.9191 \cdot 10^{18}$	$1 \cdot 10^{-6}$ +/- $6.5726 \cdot 10^{7}$ and 31106 +/- 4.65	$-1.1303 \cdot 10^{-2}$ +/- $5.2537 \cdot 10^{5}$	973.05 +/- $3.0186 \cdot 10^{10}$ and 41.241 +/- 1.1093·	$2.616 \cdot 10^{15}$	2.515e+15$\cdot 10^{15}$	$-7.85 \cdot 10^{9}$	$-4.174 \cdot 10^{7}$

		and Td=11.22 +/- $2.6229 \cdot 10^{11}$		$\cdot 10^{10}$ and Td=108.67 +/- $4.0582 \cdot 10^{10}$		$\cdot 10^{16}$ and Td=103.09 +/- 1.1644e $\cdot 10^{9}$		10^{10} and Td = 84.53 +/- $9.9747 \cdot 10^{9}$				

Unfortunately, there was not adequate documentation of how P0P1, P0P1A, P0P1B, P1P1C, P1P1D was made in the process models tab. However, the values are still available and there is probably a way of recreating them. However, it is decided that using new models was better for time limited reasons. P1P1D, P1P1L and P1PG have very similar structures and can be regarded as almost the same method. To improve the result even more it would be appropriate to interpolate values for SSin for the week 49 dataset to reduce the peek that is occurring. P2D and P2D1 keep producing bad results and will be disregarded for the rest of the TFs.

10) *tf_b5_sffoso with week 49 as validation dataset*
11) *tf_b6_sffoso with week 49 as validation dataset*
12) *tf_b7_sffoso with week 49 as validation dataset*
13) *tf_b8_sffoso with week 49 as validation dataset*
14) *tf_b9_sffoso with week 49 as validation dataset*
15) *tf_b10_sffoso with week 49 as validation dataset*

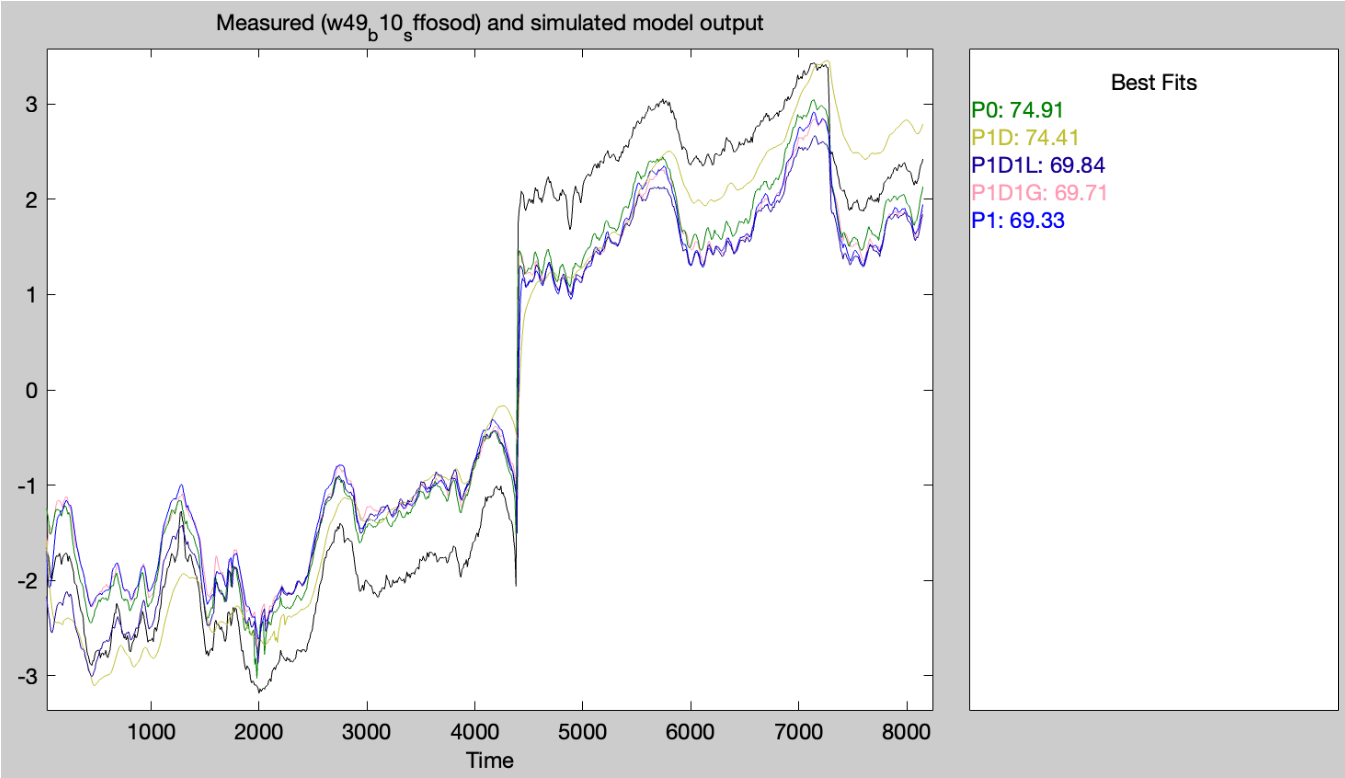

Figure 60 f_b10_sffoso fitness index and plot of all the model outputs with FO5, SO4, F and SSin as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

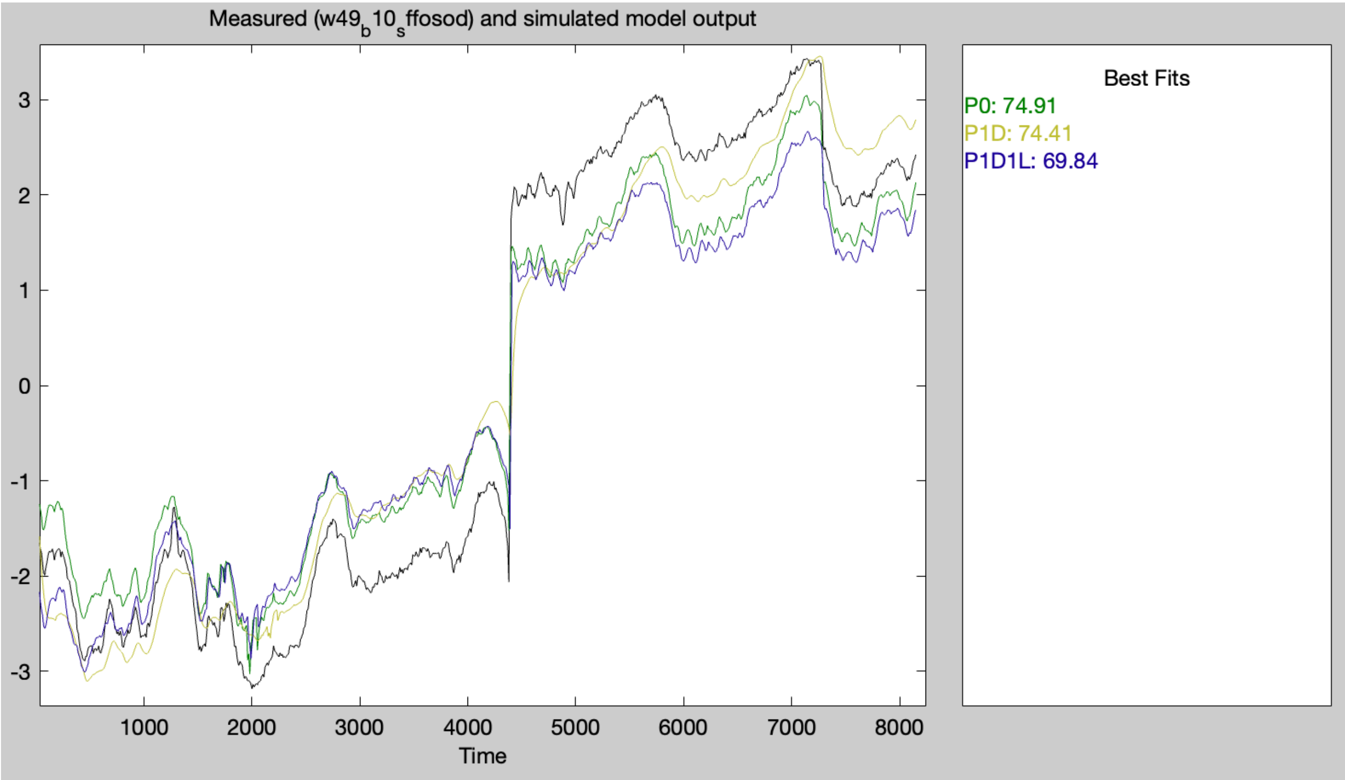

Figure 61 f_b10_sffoso fitness index and plot of the best model outputs with FO5, SO4, F and SSin as input and SO5 as output, and week 49 as validating data and week 50 as estimation data

16) tf_spo with week 49 as validation dataset

17) *tf_sfspo with week 49 as validation dataset*

18) *tf_sfnfspo without scaling FO with week 49 as validation dataset*

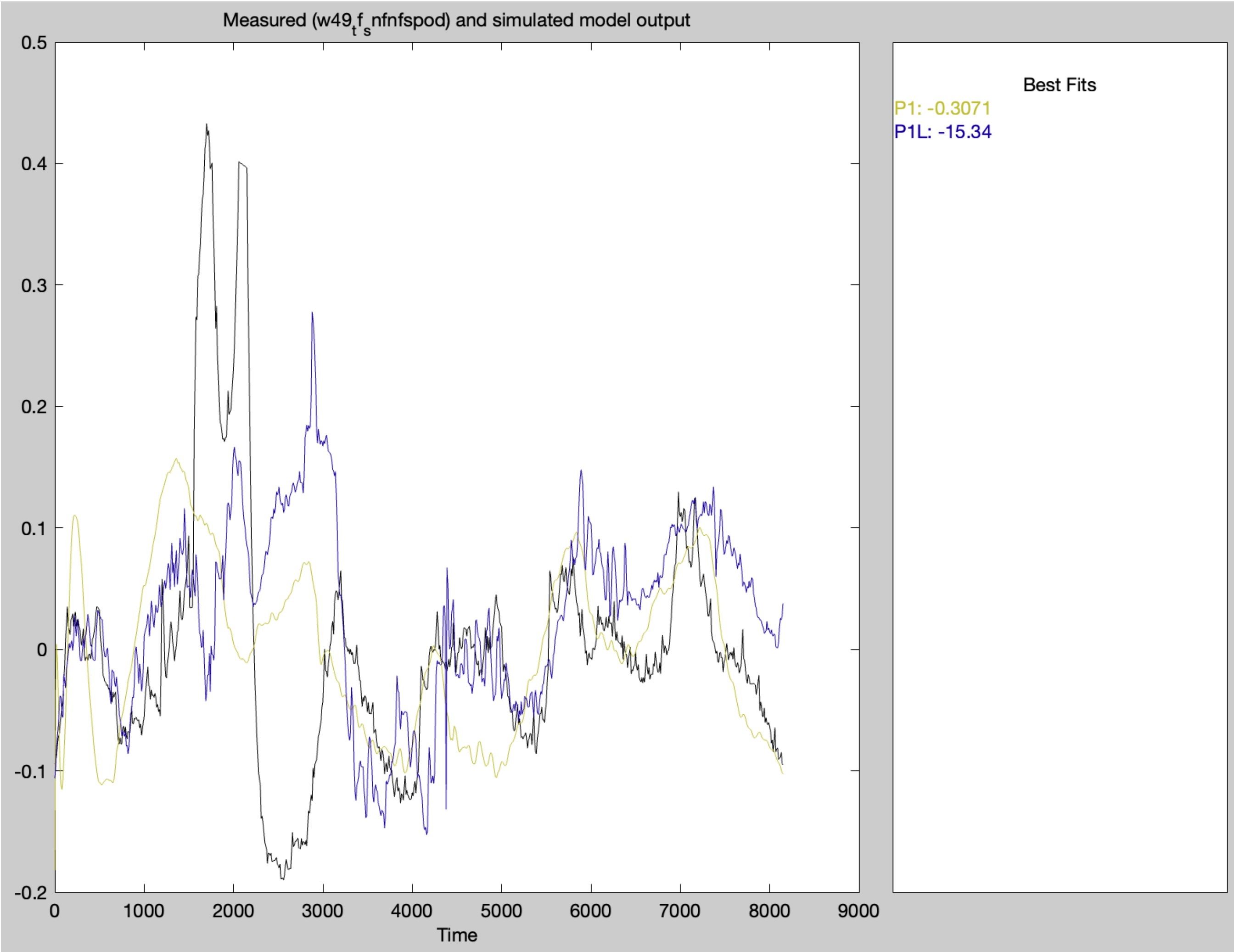

Figure 62 tf_sfnfspo fitness index and plot of the best model outputs with SSin, F, NOX, FO4, FO5, FO6, FO7, FO8, FO9, FO10 as input and SPOd as output, and week 49 as validating data and week 50 as estimation data

Table 25 tf_sfnfspo transfer functions with with SSin, F, NOX, FO4, FO5, FO6, FO7, FO8, FO9, FO10 as input and SPOd as output, and week 49 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Process Gain Kp5, mg/L	Time constant Tp5, min	Process Gain Kp6, mg/L	Time constant Tp6, min

P1L	$3.6103 \cdot 10^{-3}$+/- $4.7115 \cdot 10^{-4}$	600 +/- 90.178	$-1.4463 \cdot 10^{-3}$ +/- $2.7783 \cdot 10^{-4}$	48.698 +/- 13.305	-0.26948 +/- $3.7689 \cdot 10^{-2}$	555.88 +/- 97.801	$2.4035 \cdot 10^{-5}$ +/- $8.2596 \cdot 10^{-5}$		$-3.4996 \cdot 10^{-5}$ +/- $1.6926 \cdot 10^{-5}$		$7.2131 \cdot 10^{-5}$ +/- $2.1733 \cdot 10^{-5}$	
	Process Gain Kp7, mg/L	Time constant Tp7, min	Process Gain Kp8, mg/L	Time constant Tp8, min	Process Gain Kp9, mg/L	Time constant Tp9, min	Process Gain Kp10, mg/L	Time constant Tp10, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P1L (sameone)	$1.0015 \cdot 10^{-5}$+/- $4.8459 \cdot 10^{-5}$		$-4.9646 \cdot 10^{-5}$ +/- $8.2364 \cdot 10^{-5}$		$6.016 \cdot 10^{-5}$ +/- $1.0989 \cdot 10^{-4}$		$0.00076315 \cdot 10^{-7}$+/- $9.4947 \cdot 10^{-5}$		$4.04 \cdot 10^{-3}$	$3.885 \cdot 10^{-3}$	-9.483	-15.34

19) tf_sfnfspo with week 49 as validation dataset

Table 26 tf_b7_sffoso transfer functions with FO7s, SO6, Fs, and SSins as input and SO7 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s)	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error	Mean squared	Fit to estimati	Fit to validation

		Tp1, min							(FPE)	error (MSE)	on data in %	data (fitness index) in %
<u>POP1L</u>	1.4266 +/- 7.6289· 10^{-2}		0.38308 +/- 1.1257· 10^{-2}		15.348 +/- 1.07	420 +/- 61.971	1.5279 +/- 0.21365	420 +/- 58.098	5.972· 10^{-2}	5.857· 10^{-2}	69.15	41.32
P0	1.601 +/- 0.067446		0.30381 +/- 0.010756		8.2646 +/- 0.52772		3.3309 +/- 0.1296		6.218· 10^{-2}	6.157· 10^{-2}	68.37	34.26
P1	1.6703 +/- 8.3486· 10^{-2}	1.6315 +/- 12.283	0.21778 +/- 1.387· 10^{-2}	9.8743 +/- 2.5117	10.844 +/- 0.65573	2.0238 +/- 10.228	3.7137 +/- 84322	0.044469 +/- 2.5504· 10^{12}	8.146· 10^{-2}	7.909· 10^{-2}	64.15	33.35

20) tf_b8_sffoso

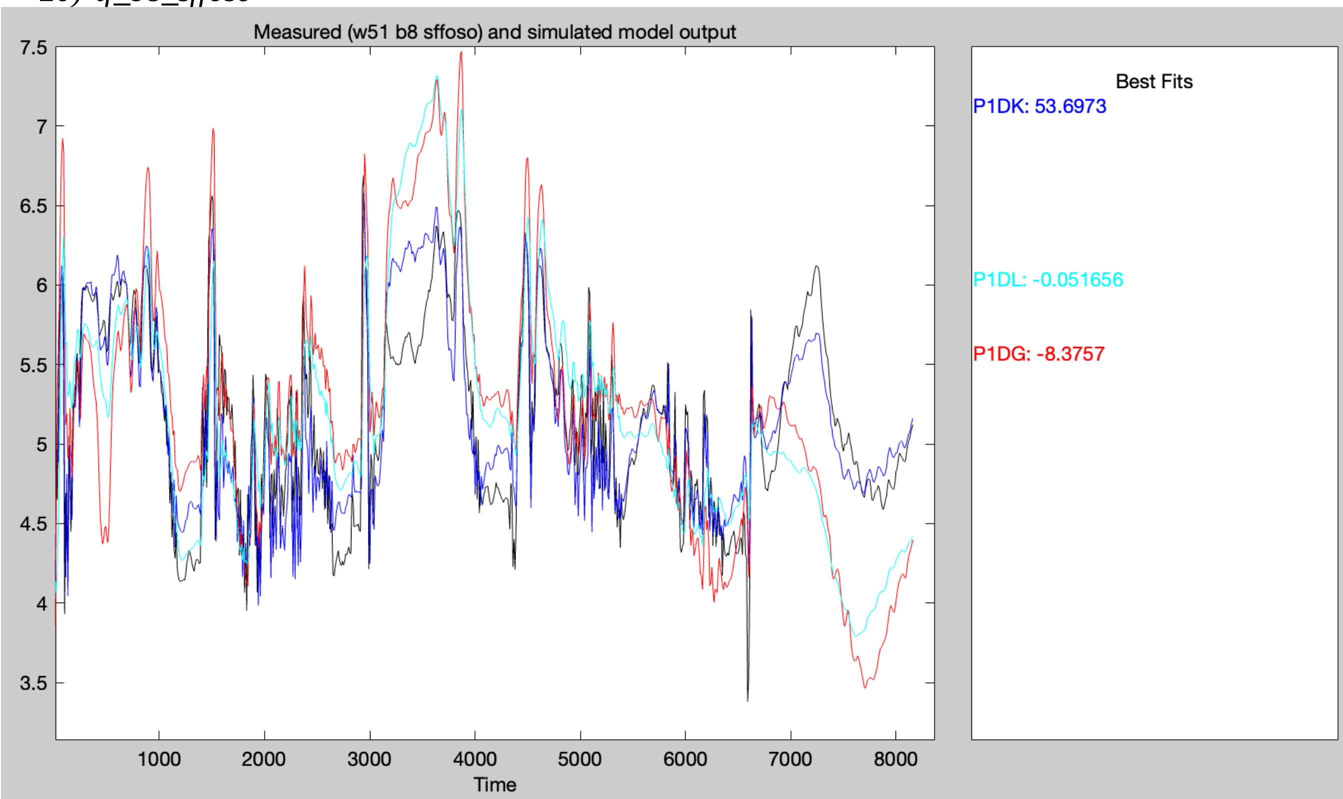

Figure 63 f_b8_sffoso fitness index and plot of the best model outputs with FO8s, SO7 Fs and SSins as input and SO8 as output, and week 51 as validating data and week 50 as estimation data

Unrelated

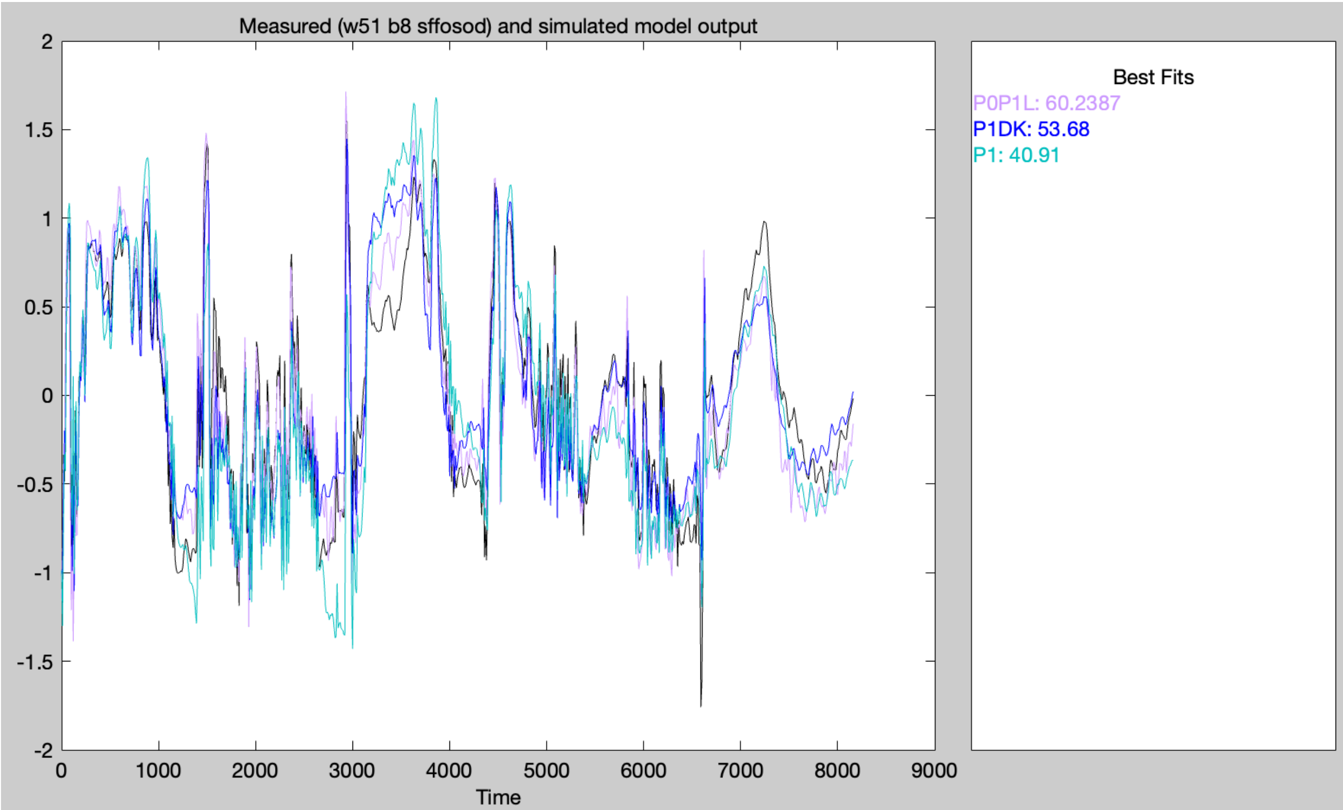

Table 27 tf_b8_sffoso transfer functions with FO8s, SO7, Fs, and SSins as input and SO8 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P0P1L	2.1858+/-0.10737		0.48261 +/- 0.019925		-23.548 +/- 0.54545	65.337 +/- 3.2489	-3.1567 +/- 0.14636	217.25 +/- 25.694	$3.795 \cdot 10^{-2}$	$3.739 \cdot 10^{-2}$	58.78	60.2387

P1DK	3.1675 +/- 0.22823	5.7307 +/- 3.5647 and Td = 0 +/- 5.4759	0.81652 +/- 7810.4	$3.1016\cdot 10^5$+/- $2.9663\cdot 10^9$ Td = 0.01 +/- 13590	-15.72 +/- 0.96833	68.868 +/- 11.456 and Td = 21.03 +/- 6.405	74.638 +/- $7.2511\cdot 10^5$	$1.0236\cdot 10^6$+/- $9.9392\cdot 10^9$ Td = 400 +/- 2317.8	$7.38\cdot 10^{-2}$	$7.096\cdot 10^{-2}$	43.21	53.7
P1	3.0537 +/- $1.0419\cdot 10^5$	6.8536e-05 +/- $1.498\cdot 10^{10}$	0.9369 +/- $4.3929\cdot 10^{-2}$	193.37 +/- 15.882	-34.088 +/- 1.0155	153.93 +/- 7.3955	-7.54 +/- 0.34026	227.2	$3.424\cdot 10^{-2}$	$3.358\cdot 10^{-2}$	60.93	40.91

21) tf_b4_sffoso

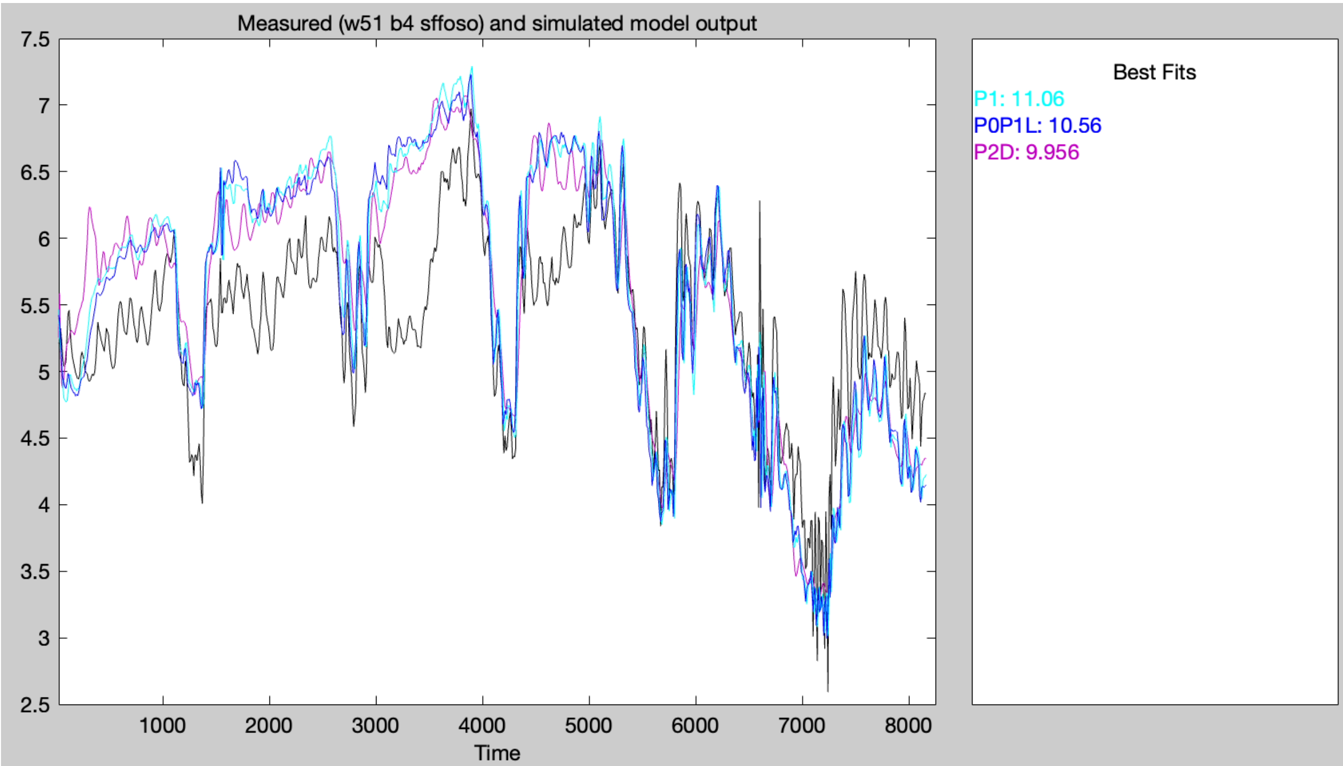

Figure 64 f_b4_sffoso fitness index and plot of the best model outputs with FO4s, Fs and SSins as input and SO4 as output, and week 51 as validating data and week 50 as estimation data

Table 28 tf_b4_sffoso transfer functions with with FO4s, Fs, and SSins as input and SO4 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness

										index) in %
P1	7.8821· 10^{-4} +/- 1.7364 · 10^{-4}	0.38238 +/- 2.6854· 10^{8}	14.79 +/- 0.84909	162.83 +/- 23.537	3.9018 +/- 0.16162	56.446 +/- 10.698	0.1157	0.1131	68.17	11.06
P0P1L	7.6014· 10^{-4} +/- 2.8603· 10^{-5}		14.62 +/- 0.89273	120.38 +/- 18.264	4.0728 +/- 0.13927	121.65 +/- 21.76	0.1238	0.1217	66.98	10.56
P2D	7.2902· 10^{-4} +/- 3.381· 10^{-4}	57.82 +/- 4.2767 · 10^{4} and 0.67903 +/- 6.495· 10^{6} and Td=0.07 +/- 6.5654· 10^{6}	12.834 +/- 1.0762	21.599 +/- 21.762 and 4.194 +/- 58.749 and Td = 0 +/- 49.444	4.5128 +/- 3.6394 · 10^{3}	0.65066 +/- 1.6997· 10^{10} and 1.6683 +/- 3.6893· $10^{6}$and Td = 0 +/- 2.7823· 10^{10}	0.1694	0.1621	61.9	9.956

22) *tf_b5_sffoso*

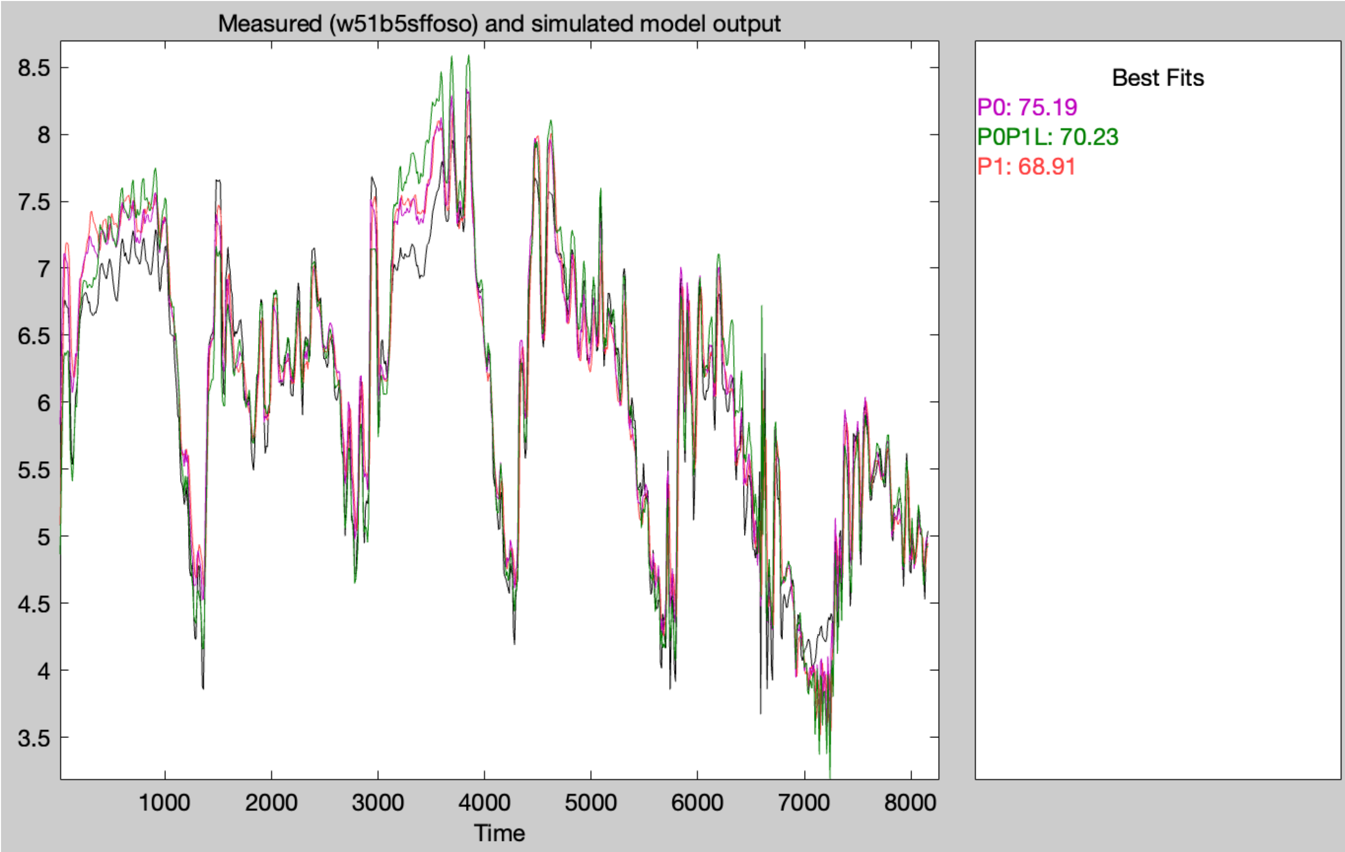

Figure 65 f_b5_sffoso fitness index and plot of the best model outputs with FO5s, SO4 Fs and SSins as input and SO5 as output, and week 51 as validating data and week 50 as estimation data

Table 29 tf_b5_sffoso transfer functions with FO5s, SO4, Fs, and SSins as input and SO5 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P0	1.0324 +/- $2.4972\cdot 10^{-2}$		0.55741 +/- $1.5893\cdot 10^{-2}$		4.9863 +/- 0.55411		1.68 +/- 0.1133		$6.793\cdot 10^{-2}$	$6.727\cdot 10^{-2}$	78.69	75.19
P0P1L	1.203 +/- 2.712		0.56743 +/- $2.2813\cdot 10^{-2}$		11.544 +/- 1.1354	300 +/- 57.206	-0.11662 +/- 0.24236	300 +/- 42.824	$7.18\cdot 10^{-2}$	$7.04\cdot 10^{-2}$	78.2	70.23

	$6 \cdot 10^{-2}$											
P1	1.0141 +/- $4.1083 \cdot 10^{-2}$	8.3128 +/- 1.5102	0.45484 +/- $2.7553 \cdot 10^{-2}$	5.2591 +/- 2.1929	8.5186 +/- 0.92979	5.3731 +/- 4.8978	2.2038 +/- 1.5873	0.38767 +/- $1.6251 \cdot 10^{9}$	0.1095	0.1063	73.21	68.91

23) tf_b6_sffoso

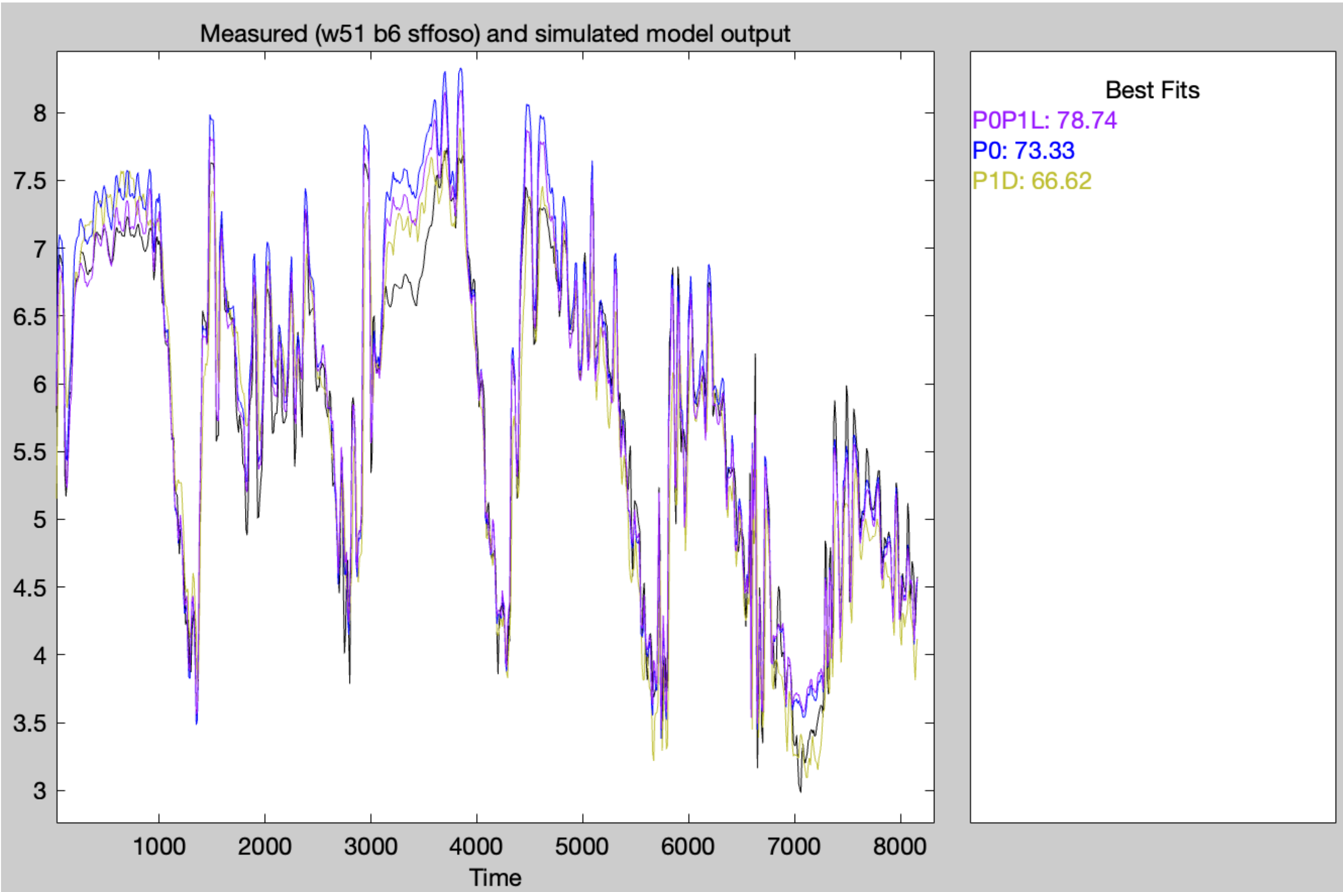

Figure 66 f_b6_sffoso fitness index and plot of the best model outputs with FO6s, SO5 Fs and SSins as input and SO6 as output, and week 51 as validating data and week 50 as estimation data

Table 30 tf_b6_sffoso transfer functions with FO6s, SO5, Fs, and SSins as input and SO6 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness

													index) in %
P0P1L	0.63581 +/- $3.9349 \cdot 10^{-2}$		0.80471 +/- $2.4212 \cdot 10^{-2}$		-0.14453 +/- 0.81994	228.01 +/- 1882.2	$4.8607 \cdot 10^{-2}$ +/- 0.14468	240.18 +/- 1955.6	$5.821 \cdot 10^{-2}$	$5.708 \cdot 10^{-2}$	83.5	78.74
P0	0.72742 +/- $2.8064 \cdot 10^{-2}$		0.80311 +/- $1.9523 \cdot 10^{-2}$		1.2094 +/- 0.51764		-0.31017 +/- 0.12084		$5.249 \cdot 10^{-2}$	$5.198 \cdot 10^{-2}$	84.25	73.33
P1D	0.48919 +/- 4.0063e+09	0.12966 +/- $2.8881 \cdot 10^{12}$ Td = 3.66 +/- $2.0458 \cdot 10^{19}$	12.444 +/- 1.938e+10$\cdot 10^{10}$	1.5937 +/- 3.2679e+08 Td = 1.59 +/- $5.9773 \cdot 10^{8}$	12.444 +/- $1.938 \cdot 10^{10}$	1.5937 +/- $3.2679 \cdot 10^{8}$ Td = 1.59 +/- $5.9773 \cdot 10^{8}$	-2.0552 +/- 2.0408e+09	9789.4 +/- 8447.1 Td = 159.18 +/- $8.1755 \cdot 10^{8}$	0.169	0.1625	72.15	66.62

24) *tf_b7_sffoso*

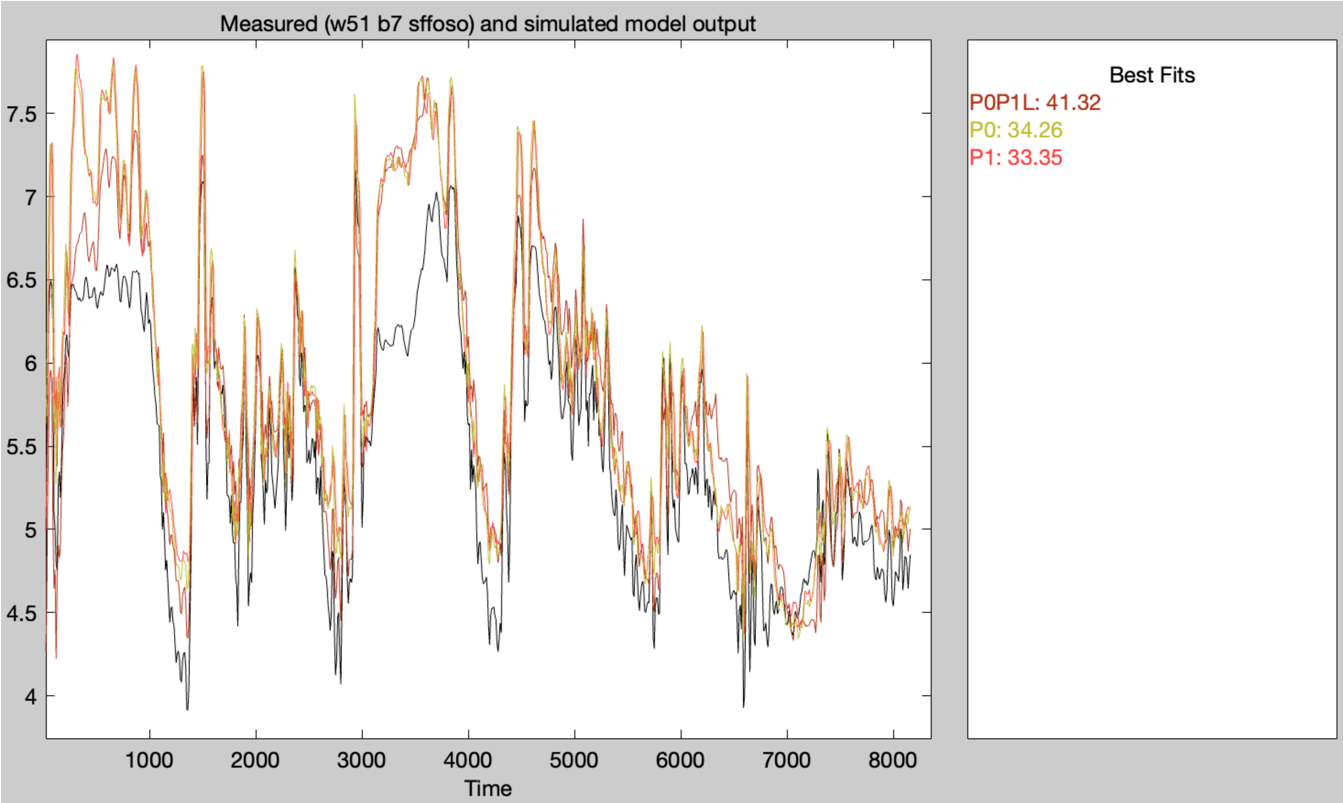

Figure 67 f_b7_sffoso fitness index and plot of the best model outputs with FO7s, SO6 Fs and SSins as input and SO7 as output, and week 51 as validating data and week 50 as estimation data

Table 31 tf_b7_sffoso transfer functions with FO7s, SO6, Fs, and SSins as input and SO7 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
<u>POP1L</u>	1.4266 +/- $7.6289 \cdot 10^{-2}$		0.38308 +/- $1.1257 \cdot 10^{-2}$		15.348 +/- 1.07	420 +/- 61.971	1.5279 +/- 0.21365	420 +/- 58.098	$5.972 \cdot 10^{-2}$	$5.857 \cdot 10^{-2}$	69.15	41.32
P0	1.601 +/- 0.067446		0.30381 +/- 0.010756		8.2646 +/- 0.52772		3.3309 +/- 0.1296		$6.218 \cdot 10^{-2}$	$6.157 \cdot 10^{-2}$	68.37	34.26
P1	1.6703 +/- 8.348	1.6315 +/- 12.283	0.21778 +/-	9.8743 +/- 2.5117	10.844 +/-	2.0238 +/- 10.228	3.7137 +/- 84322	0.044469 +/-	$8.146 \cdot 10^{-2}$	$7.909 \cdot 10^{-2}$	64.15	33.35

	$6\cdot 10^{-2}$		$1.387\cdot 10^{-2}$		0.65573			$2.5504\cdot 10^{12}$				

25) tf_b8_sffoso

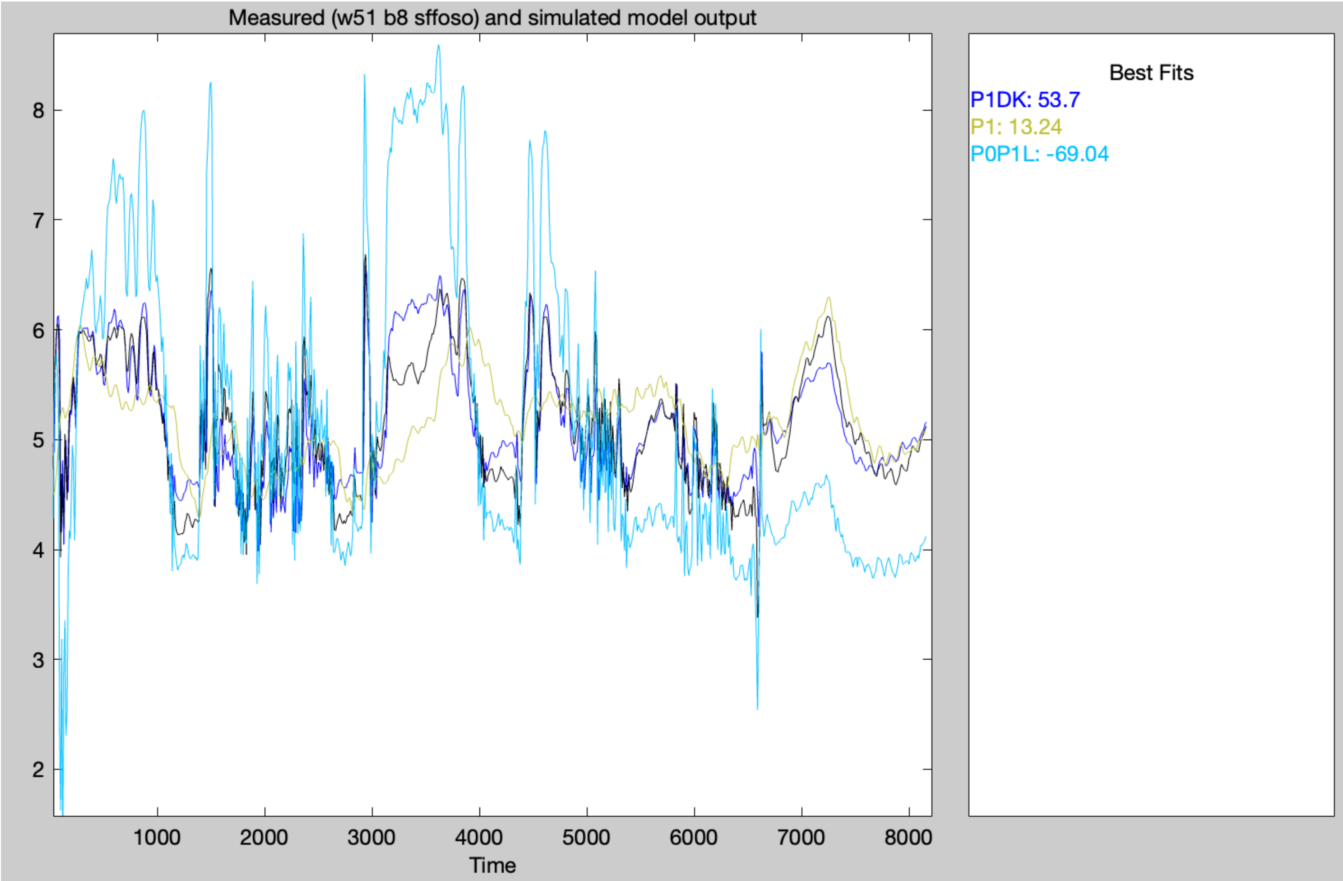

Figure 68 f_b8_sffoso fitness index and plot of the best model outputs with FO8s, SO7 Fs and SSins as input and SO8 as output, and week 51 as validating data and week 50 as estimation data

Table 32 tf_b8_sffoso transfer functions with FO8s, SO7, Fs, and SSins as input and SO8 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
<u>P0P1L</u>	5.6197 +/- 0.22108		0.23002 +/- $4.3428\cdot 10^{-2}$		-8.8233 +/- 1.0477	41.37 +/- 13.422	0.23383 +/- 0.20908	480 +/- 701.83	0.2325	0.2291	-2.041	-69.04

P1DK	3.1675 +/- 0.22823	5.7307 +/- 3.5647 and Td = 0 +/- 5.4759	0.81652 +/- 7810.4	$3.1016 \cdot 10^5$+/- $2.9663 \cdot 10^9$ Td = 0.01 +/- 13590	-15.72 +/- 0.96833	68.868 +/- 11.456 and Td = 21.03 +/- 6.405	74.638 +/- $7.2511 \cdot 10^5$	$1.0236 \cdot 10^6$+/- $9.9392 \cdot 10^9$ Td = 400 +/- 2317.8	$7.38 \cdot 10^{-2}$	$7.096 \cdot 10^{-2}$	43.21	53.7
P1	6.3431 +/- 0.34623	16312 +/- 18391	1.0385 +/- 0.15556	71.739 +/- 20.862	-36.916 +/- 4.1841	105.04 +/- 20.486	-6.1636 +/- 1.2638	148.88 +/- 44.917	$7.92 \cdot 10^{-2}$	$7.698 \cdot 10^{-2}$	40.85	13.24

26) tf_b9_sffoso

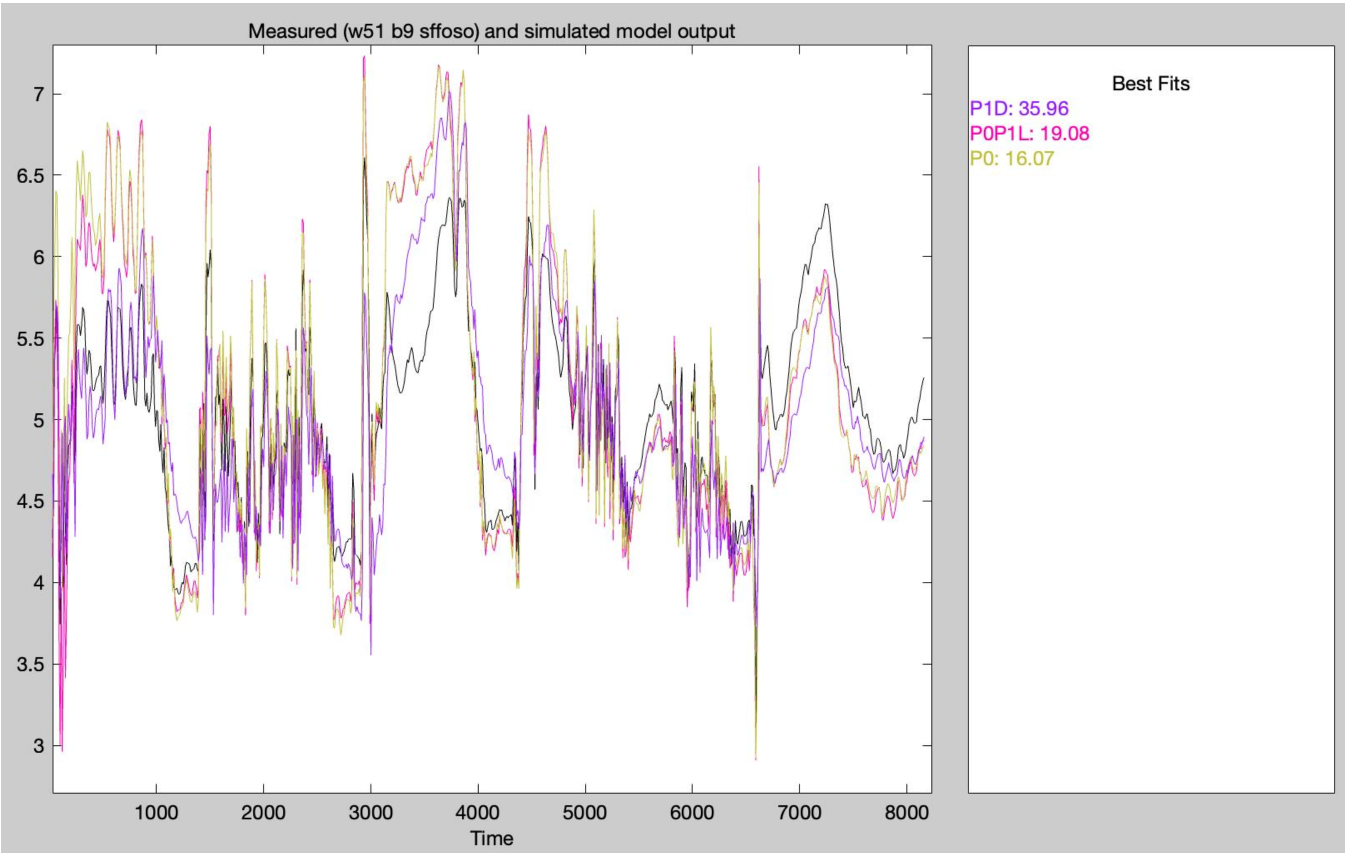

Figure 69 f_b9_sffoso fitness index and plot of the best model outputs with FO9s, SO8 Fs and SSins as input and SO9 as output, and week 51 as validating data and week 50 as estimation data

Table 33 tf_b9_sffoso transfer functions with FO9s, SO8, Fs, and SSins as input and SO9 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P1D	4.3145 +/- 0.17703	1.6306 +/- 16.473 Td= 5.57 +/- 43.769	0.97863 +/- 0.031209	429.05 +/- 40.313 Td = 0 +/- 11.045	-12.282 +/- 0.68861	= 63.698 +/- 7.5864 Td = 17.95 +/- 3.8439	-3.0792 +/- 0.16585	149.44 +/- 18.513 Td = 179.93 +/- 8.5705	$3.756\cdot 10^{-2}$	$3.612\cdot 10^{-2}$	61.37	35.96
P0P1L	3.4222 +/- $9.64\cdot 10^{-2}$		0.75981 +/- $1.0668\cdot 10^{-2}$		-3.6539 +/- 0.33536	125.15 +/- 23.695	-1.1004 +/- $7.9293\cdot 10^{-2}$	75.244 +/- 16.354	$2.509\cdot 10^{-2}$	$2.473\cdot 10^{-2}$	68.04	19.08
P0	3.2567 +/- 0.10489		0.77257 +/- $1.2052\cdot 10^{-2}$		-1.697 +/- 0.25611		-1.3569 +/- $8.9435\cdot 10^{-2}$		$3.048\cdot 10^{-2}$	$3.018\cdot 10^{-2}$	64.69	16.07

27) *tf_b10_sffoso*

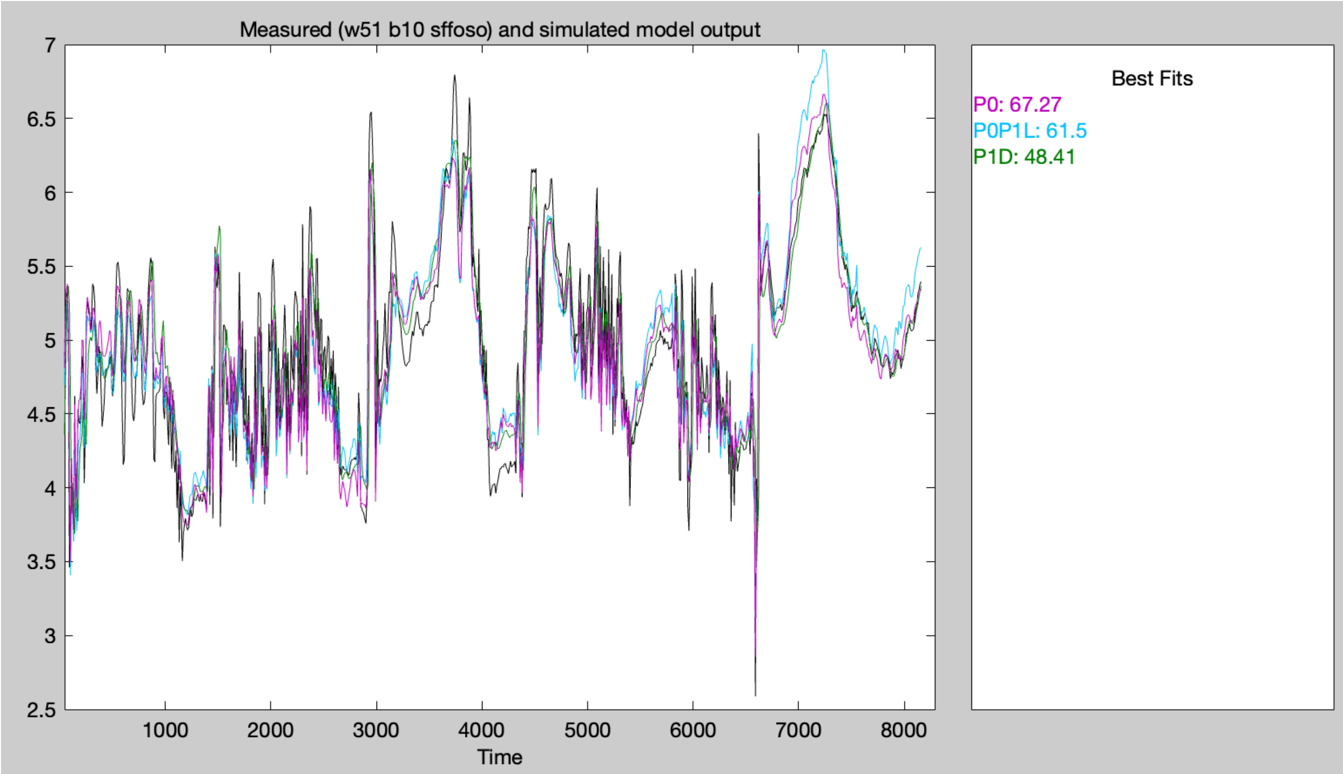

Figure 70 f_b10_sffoso fitness index and plot of the best model outputs with FO10s, SO9, Fs and SSins as input and SO10 as output, and week 51 as validating data and week 50 as estimation data

Table 34 tf_b10_sffoso transfer functions with FO10s, SO9, Fs and SSins as input and SO10 as output, and week 51 as validating data and week 50 as estimation data

Model name	Process gain Kp1, mg/L	Time constant(s) Tp1, min	Process Gain Kp2, mg/L	Time constant Tp2, min	Process Gain Kp3, mg/L	Time constant Tp3, min	Process Gain Kp4, mg/L	Time constant Tp4, min	Final prediction error (FPE)	Mean squared error (MSE)	Fit to estimation data in %	Fit to validation data (fitness index) in %
P0	1.9203 +/- 0.10018		0.96055 +/- 0.012481		-1.9643 +/- 0.28012		-1.4773 +/- 0.087927		0.03776	0.03739	69.45	67.27
P1L	3.4631 +/- 0.21263		0.89267 +/- 0.018051		-8.5198 +/- 1.8709	750 +/- 203.64	-1.1618 +/- 0.25166	342.08 +/- 113.69	0.03594	0.03524	70.34	61.5
P1D	1.9102 +/-	146.36 +/- 133.43	0.96124 +/-	8.9938 +/- 4.7501	-4.2547 +/-	10481 +/-	-0.89088 +/-	207.35 +/- 145.21	0.06389	0.06143	60.84	48.41

	1.0381	and Td=0 +/- 69.981	$4.8447\cdot 10^{-2}$	and Td = 0+/- 5.2148	13.316	$1.1942\cdot 10^{5}$and Td=0 +/- 2728.9	0.39291	and Td=120 +/- 78.095				

E. Appendix 5

1) Old simulation models with transfer functions between the aeration and dissolved oxygen were implemented

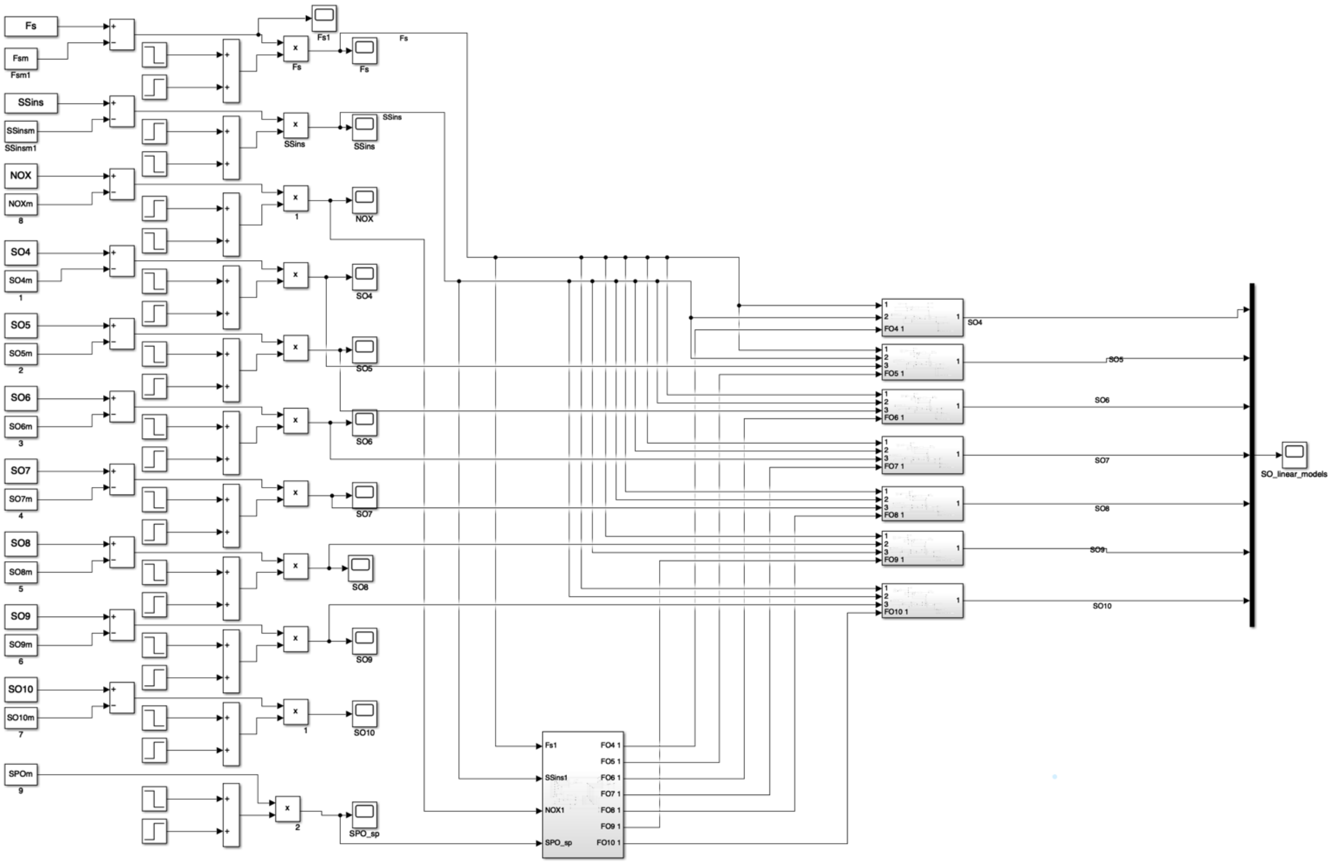

Figure 71 Simulation model for the whole system with the subsystem being where linear models and PID controllers for each of the CVs (SO4-SO10, and S_{POd}) have been developed

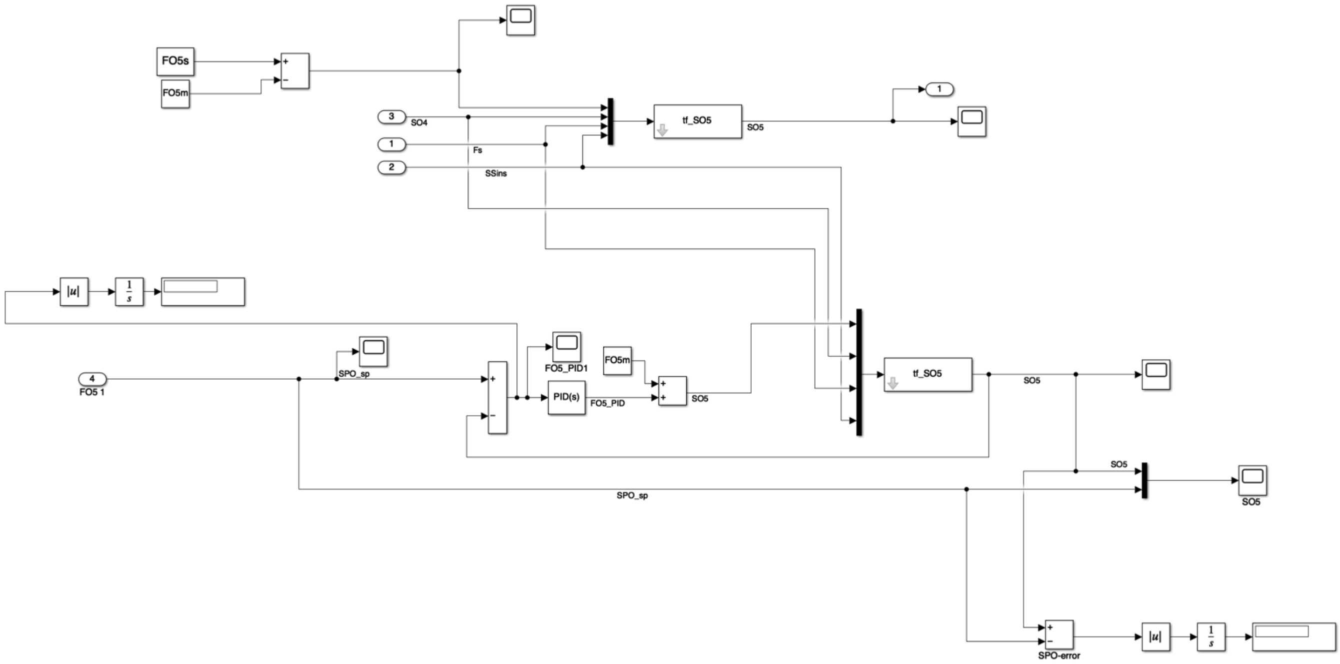

Figure 72 One of the subsystems where the SO5 CV was developed, with the linear model and the PID that controls FO5s. The other subsystems have the exact same concept just for each of the S_O in question

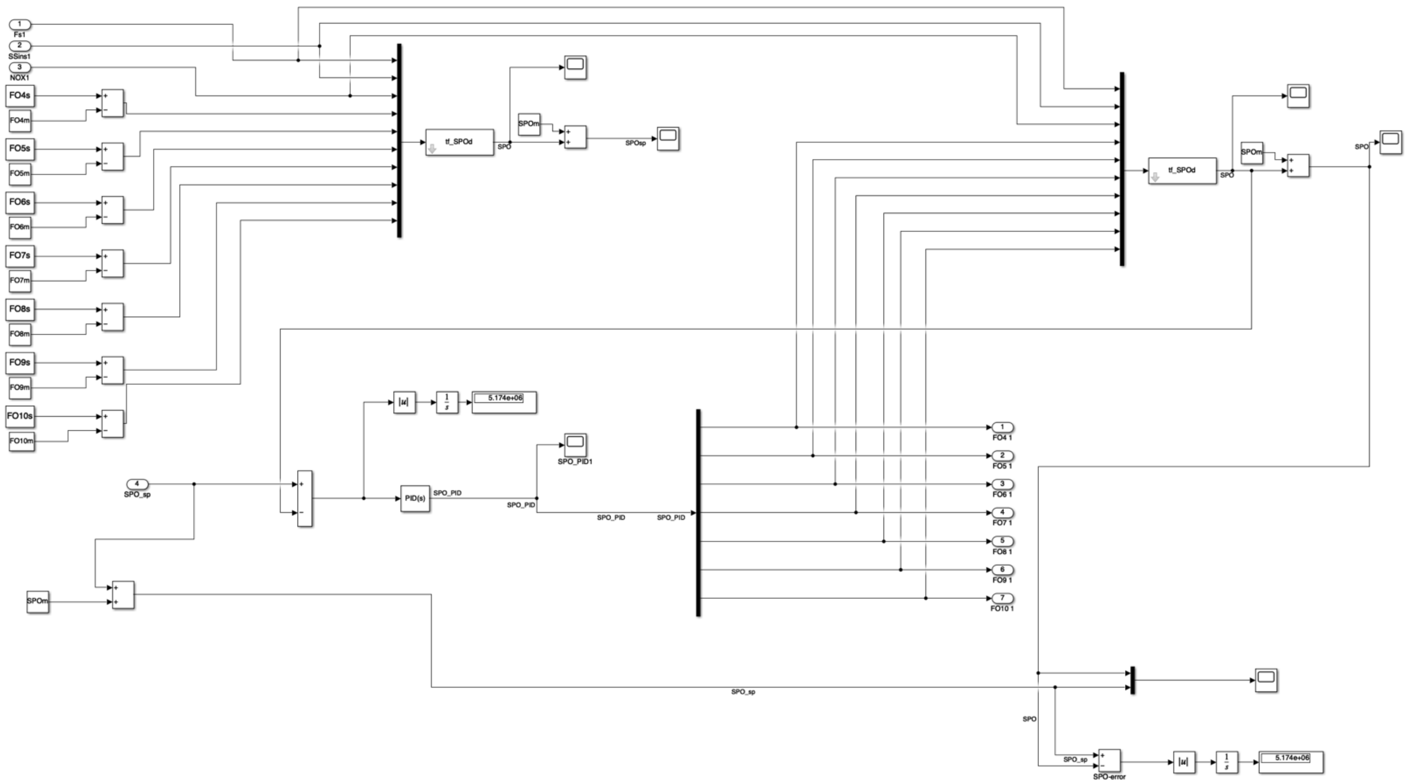

Figure 73 The subsystem for S_{POd} CV was developed, with the linear model and the PID that sets the setpoint for F_O, where the setpoint is the mean of S_{POd}

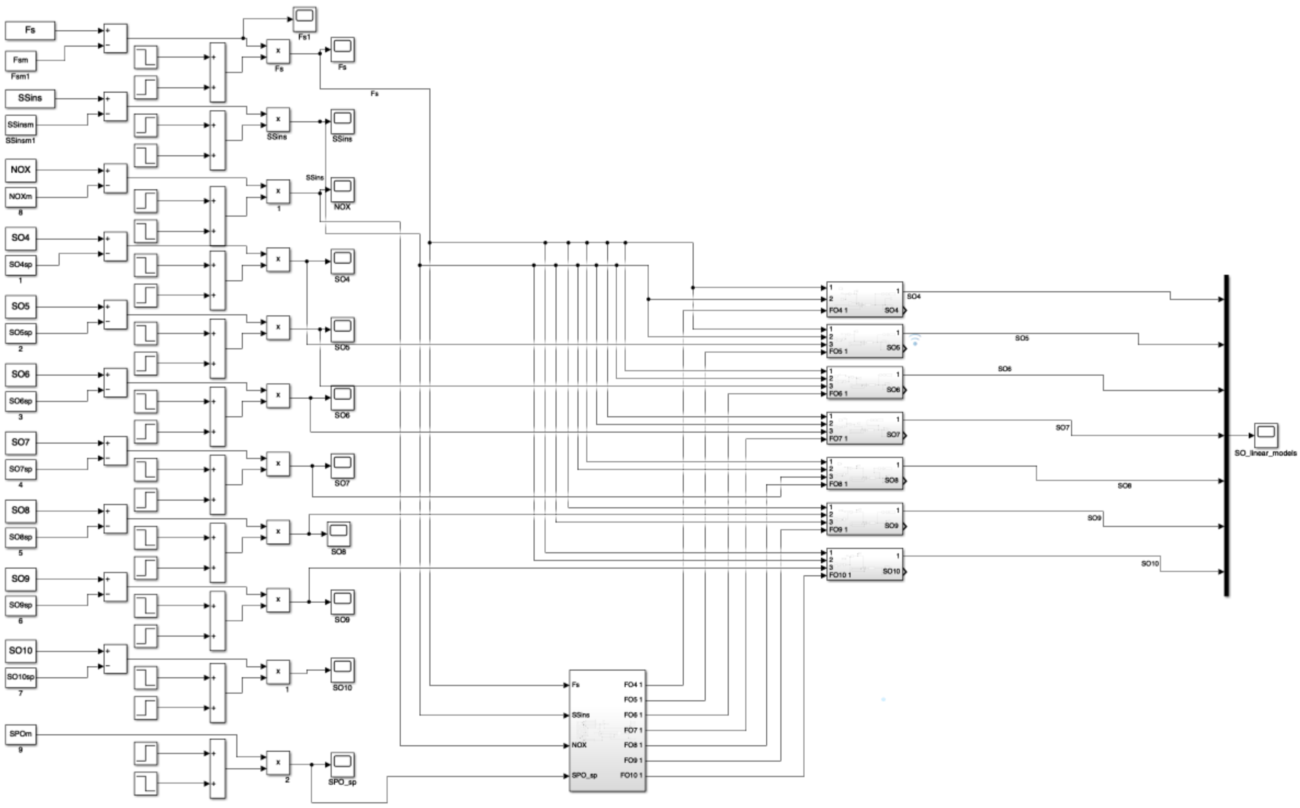

Figure 74 Simulation model for the whole system with the subsystem being where MPC controllers for each of the CVs (SO4-SO10, and S_POd) have been developed

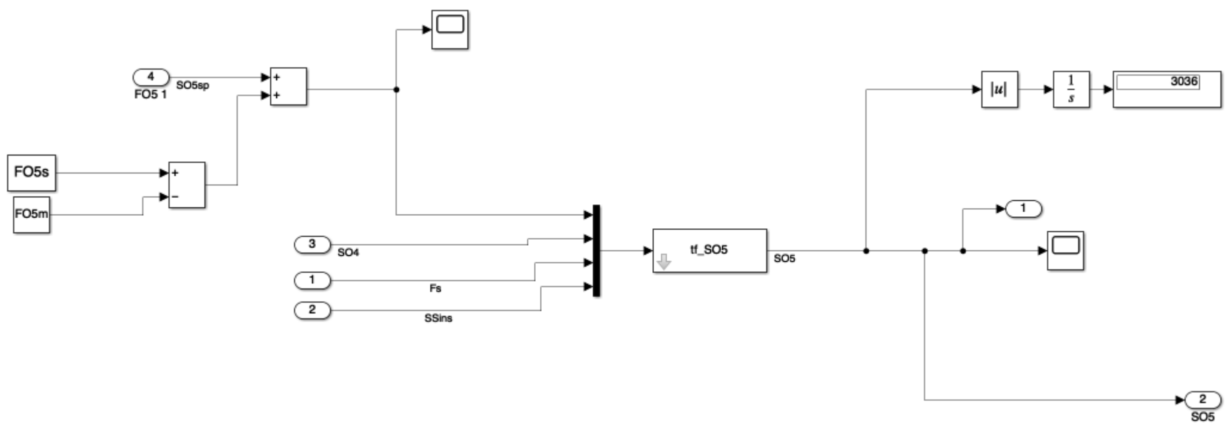

Figure 75 One of the subsystems where the SO5 CV was developed. The other subsystems have the exact same concept just for each of the S_O in question

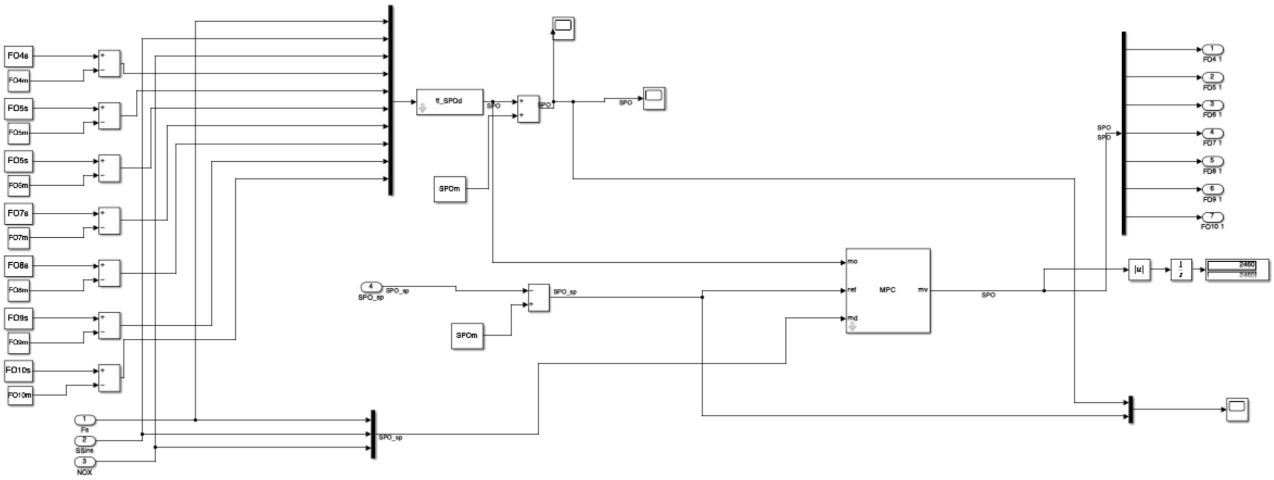

Figure 76 The subsystem for S_{POd} CV was developed, with the MPC.

F. Appendix 6

1) Gantt diagram

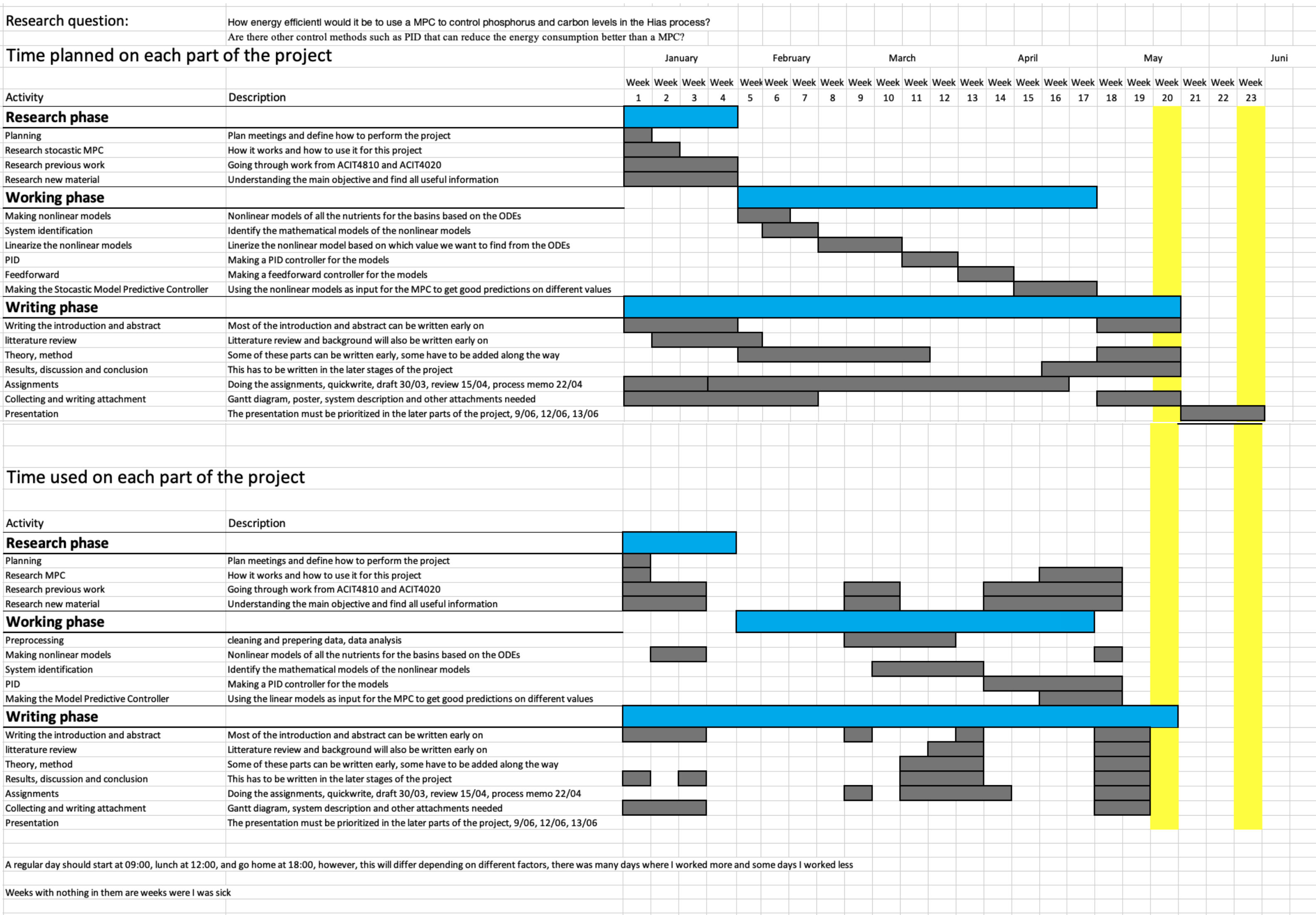

Figure 77 Gantt diagram