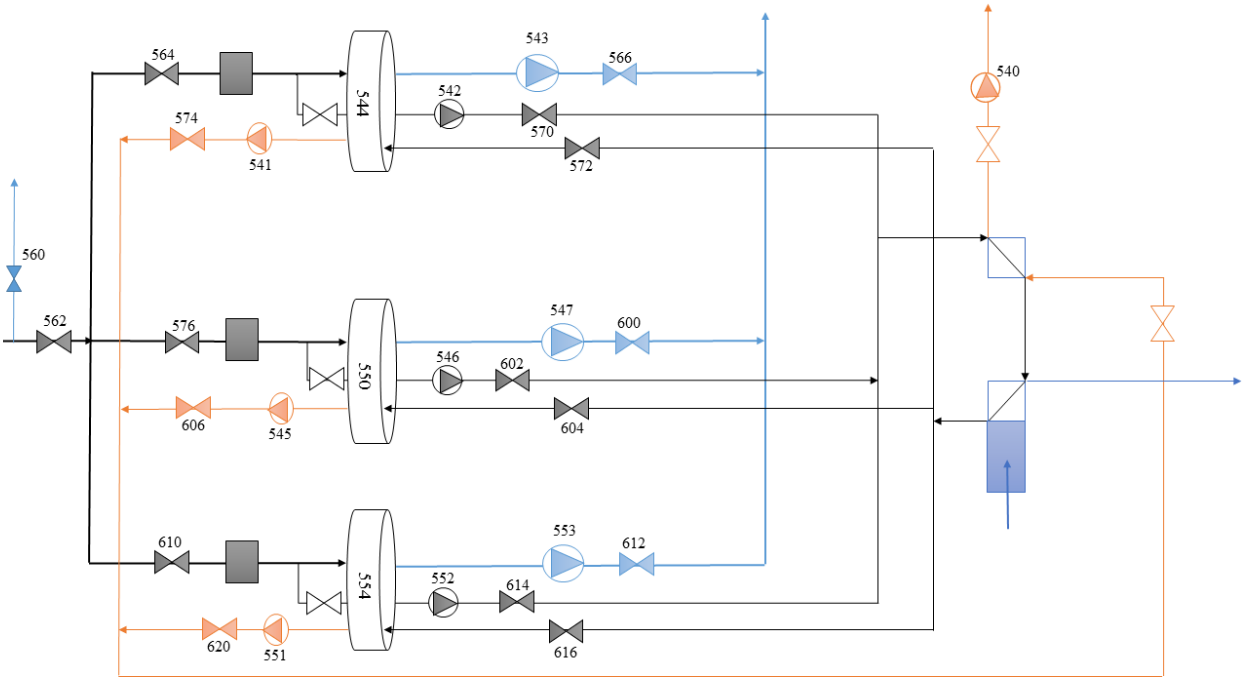
CHALMERS
UNIVERSITY OF TECHNOLOGY



Updating and optimizing a PLC for a zeloite fan at Volvo Cars

Master’s thesis in Systems, Control and Mechatronics

EMELIE JOHANSSON

Electrical Engineering
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

MASTER'S THESIS 2019:NN

Updating and optimizing a PLC for a zeloite fan at Volvo Cars

Emelie Johansson

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department Electrical Engineering
Division of Automation
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2019

Updating and optimizing a PLC
for a zeloite fan at Volvo Cars
EMELIE JOHANSSON

Supervisor: Sarmad Riazi, Electrical Engineering
Supervisor: Patrik Johnasson, Volvo Cars
Examiner:Petter Falkman, Electrical Engineering

Master's Thesis 2019:NN
Department of Electrical Engineering
Division of Automation
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: A vizualization of the system that is being optimized and updated.

Typeset in LaTeX
Printed by Chalmers Reproservice
Gothenburg, Sweden 2019

Updating and optimizing a PLC for a zeolite fan at Volvo Cars

EMELIE JOHANSSON
Department of Electrical Engineering
Chalmers University of Technology

Abstract

This report aims to explain, optimize and rebuild a programmable logic controller, PLC, for a system that is filtering high level of hydrocarbon. The filter that is being used are zeolites and the system is already installed. The new PLC will be from Siemens and is a SIMATIC 7-1500 and the programming is mostly used with function blocks, except for the start/stop sequences that will be programmed using sequences.

The system contains five subsystems where the three in the middle works in parallel filtering the dirty air. If the system is on the subsystem in the beginning and the subsystem in the end are both on. The three subsystems in the middle can be on or off depending on how much flow that is necessary. However, which subsystem to run is controlled manually today, which is both unnecessary and non-energy efficient to let the systems run even if they do not have to.

The system is optimized with focus on energy savings and efficiency. The optimization uses values from experiments on the installed system and information from data sheets on the motors that are used. The method that was used to achieve the minimum cost in energy for the system was integer programming, IP, and the solver was `intlinprog` from MATLAB. `Intlingprog` is a Mixed Integer Linear Programming method, but can be used for only IP too. The result from the optimization was implemented in the controller.

A cascade controller is used to control the system, the optimization is implemented in the inner-loop, the outer-loop is controlled with a PID controller. The input to the controller is the pressure difference in the pipes before the system. The output of the PID is a fractal number between 0-1 that corresponds to 0-100%. That output value goes into the optimization part of the controller that decides which of the three subsystems in the system that should be on/off and on which speed the main fan should run at.

The PLC code and the cascade controller worked well according to PLCsim. The optimization gave a more preferable controller and made sure that the system was not consuming more energy than necessary.

Keywords: PLC, optimization, Cascade controller, PID, Linear Programming, Zeolites, Energy, Volvo Cars.

Acknowledgements

I would love to thank Patrik Johansson who has been my supervisor at Volvo Cars and has helped me during the project. With programming, get contact with the right people and for helping out with the understanding of the system. I also want to thank Stefan Lösnitz for helping me with my PLC programming. Moreover, a big thank you to Sarmad Riazi, my supervisor at Chalmers University of Technology, for helping me with the report, the optimization and for believing in my ideas. Thank you to Anton Albo for creating the thesis and for helping me with my ideas. A thank you to Stefan Odham with employees for having me at your office and for giving me the opportunity to create this. Last but not least I would like to thank my examiner, Petter Falkman, for commenting on my work and helping me get started.

Emelie Johansson, Gothenburg, June 2019

Definitions

TIA-portal - is the software when programming in PLC Siemens. Volvo Cars has libraries that can/has to be used for the construction of PLC for them.

Human Machine Interface - HMI - allows a human/operator to interface with the machine. For example a panel where it is possible to monitor the process and make changes.

PLCtags - is used to connect the HMI panels to the code in the TIA-environment. Makes it possible to monitor the code and their signals.

PLCsim - is a tool for PLC Siemens to be able to simulate the code that has been written in the TIA-portal. In the simulation program it is possible to test for example different values for the input of a PID-controller to see how it changes. Also the PLCsim is used to check if the coding is working and if for example the a motor get the signal to start when it supposed to etc.

Contents

List of Figures

List of Tables

1

Introduction

This report is made to describe the project where a Programmable Logic Controller, PLC is constructed and optimized for a filtering system at Volvo Cars in Torslanda. The filtering system is filtering some of the process air from the painting factory that consists high levels of hydrocarbon. The hydrocarbon is distinguished using zeolites and today the system is controlled by an old SattCon 5 controller [1]. In case of a breakdown it will also be taken in account to fast and easily replace the old SattCon with the new PLC from Siemens. It will also be investigated if it is possible to easily and without any larger disturbances change the old PLC with the new one if a chance is given.

The optimization will result in which of the three subsystems in the middle that should be on/off and on which speed their main fans, (`543, 547, 553`), should run at to give the least consumption of energy. The system that is going to be optimized and controlled is shown in Figure 1.1. In the figure, the black lines corresponds to the process air, the blue is the clean/filtered air and the orange is the air with higher hydrocarbon that is going to an other filtering system before it can be burned.

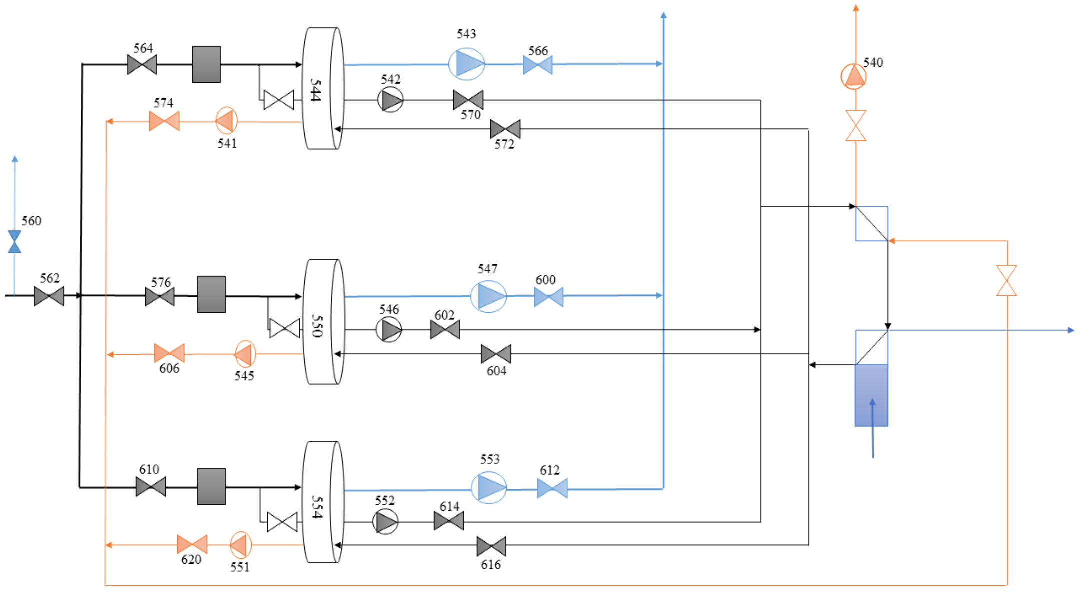


Figure 1.1: The main system that will be programmed and optimized

1.1 Background

Volvo Cars has a plant in Torslanda which is divided into three main factories. The first one is the body-factory where the car body is made. The second factory is the painting and the third is the mounting factory, where the car is utilizable when it leaves the line. In the painting factory a lot of chemicals are being used to be able to paint the bodies, which result in the need of filtering the process air before it is let out in the atmosphere. In one part of the painting factory, the process air is mixed with a solvent of hydrocarbon.

Today Volvo Cars is using zeolites to filter the hydrocarbon from the rest of the air, zeolites are a collected word of crystalline structures that is made of silicon, aluminum and oxygen. The materials are building a kind of frame with small channels and cavities that makes some smaller molecules and atoms pass through, while some of the sticks on the frame. In this case the oxygen is passing through but the hydrocarbon sticks on the frame. Some zeolites are minerals that exists naturally, and some are synthetically made. The zeolites are often spoken of as molecules sieves[2]. In this case the zeolites trap the particles through a rotating filter. The hydrocarbon is then pushed back into another pipe by heated air that transports it to another system where they will be filtered one more time before burning/combustion the hydrocarbon for energy that can produce electrical energy and heat. In Figure 1.1 the numbers `544`, `550` and `554` represent the three rotating zeolite filters.

The control system is a PLC from SattCon5 but it is old and in need of an update. There is no real time monitoring of the system to overview the process and the only knowledge about the control system are in old cases and from the people working at maintenance. This means that the information about the system is limited.

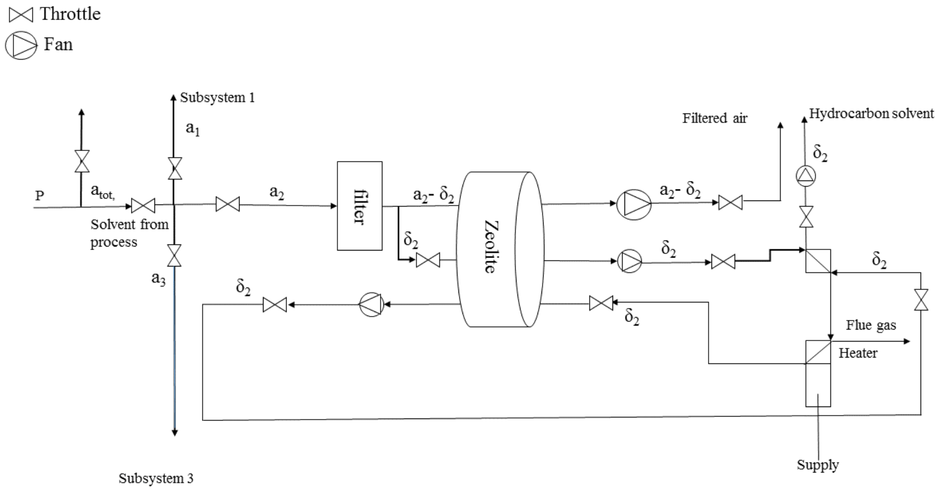


Figure 1.2: One of the three subsystems of the filtering system

Figure 1.2 is showing `subsystem 20` together with the input `subsystem 35-009` and heater `subsystem 35-060`. a and δ are flows in m^3/s and P is the measured differential pressure vacuum that will be used in the new feedback controller.

1.2 Purpose

Since the process today does not have a live monitoring functionality from a computer, the goal is to create a new PLC logic that controls the system with a control feedback from the differential pressure vacuum before the first throttle. An additional goal is to reduce the environmental footprint of the process by optimizing the system with focus on energy efficiency. The optimization will focus on which of the three subsystems,`35-010`,`35-020`,`35-030` , to run and on which speed their main fans,`010-553`,`020-547`,`030-543`, should work on.

1.3 Limitations

This project will not include any analysis regarding replacing the systems fans, zeolite or any other equipment that requires a larger amount of work at site.

Another limitation of the project is that it will not be possible to try the new PLC on the system, but only with help from PLCsim.

The controller will be a PID-controller and will not be programmed in low-level controller but only high-level. This since the system it controls will not be torn down into current or voltage, because the system is to large and the PID will control more than one object. The values of the PID will be tried out using PLCsim and will not be tested on the real system.

The installation will not be investigated deeply, and it will not be any cost functions of that due to the time frame.

2

Theory

2.1 Programmable Logic Controller

PLC is a programmable logic controller and is used to control different systems. There are different kinds of PLC. But in this section only an explanation of PLC from Siemens will be stated.

2.1.1 PLC Siemens

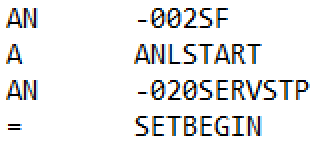
The PLC Siemens is of Boolean type, it could either take the value 0/FALSE or 1/TRUE. But there can also be real values. For example, the signal to make something start is of Boolean but a set point could be a real value. The tool of programming in PLC Siemens is called TIA-portal and is a software created by Siemens. There are different programming languages in PLC programming, and they are as follows:

1. Ladder Logic
2. Function Block Diagram, FBD
3. Structured text, SCL
4. Statement List, STL
5. Step sequence programming (GRAPH, SFC)

[3] In this project only Ladder Logic, Structured Text, Function Block Diagram and for the start and stop sequence Step Sequence Programming will be used.

2.1.1.1 Statement List

A statement list is a textual programming language where the batch number and their row numbers are shown together with the operand and some instruction on what the code is doing. Statement list is what is being used in the old system. In Figure 2.1 a simple statement list is stated, here AN stands for AND NOT, A stands for AND and "=" stands for that if all the stated above is fulfilled the output is 1, otherwise it is 0.-002F and ANLSTART are both signals. In this case if -002F is not 1(=0), ANLSTART is 1 and -020SERVSTP is not 1(=0) then the output SETBEGIN is 1.

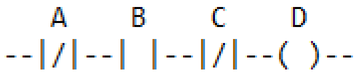


```
AN      -002SF
A       ANLSTART
AN      -020SERVSTP
=       SETBEGIN
```

Figure 2.1: A simple statementlist

2.1.1.2 Ladder Logic

Ladder Logic looks a lot like electrical charts and relay schemes and is common for electricians when programming less sophisticated code. The Ladder Logic however has to be compiled to machine code before it can be run by the PLC. In Figure 2.2 an example of a simple ladder logic is shown, the ladder does the same thing as the statement list above. Where A stands for the first row in the statement list, B for the second and C for the third. | | stands for AND, |/| stands for AND NOT and ( ) stands for equal.



```
    A    B    C    D
--|/|--| |--|/|--( )--
```

Figure 2.2: A simple ladder logic

2.1.1.3 Function Block Diagram

A function block diagram is based on one or more inputs that goes into a block that has a command and outputs from the block. A function block diagram is like a statement list but more visual. In Figure 2.3 the same function as in Figure 2.1 and 2.2 is stated. If the all inputs in the first block is 1 then the output is 1. However, in this case there are two circles where the input port is, this means that the input is inverted, so if the value is 1 it is 0 in the input and vice versa.

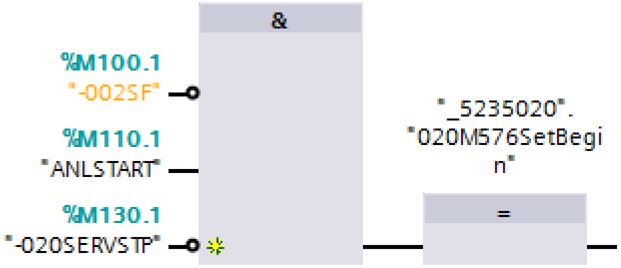


Figure 2.3: A simple function block

2.1.1.4 Step Sequence Programming

Step Sequence Programming is used when the desire is to make something happen in a specific flow, for example, if one wants a throttle to open before the fan does. There are different ways of programming them, it can be unbranched, parallel branch or alternative branch. The difference between them is that in unbranched there are one specific decided way to go through the sequence. But in branched the way can be differently depending on the input. In this project only unbranched versin is used and is showing in Figure 2.4. Steps, S_i is where a certain action is active, for example where something should start or stop. The T_i stands for transition and has to be fulfilled to go to next step, it controls if the sequence is allowed to go from one step to another.

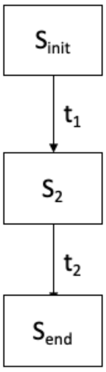


Figure 2.4: An example of a Step Sequence

2.2 Energy efficiency and consumption

Energy (Joule) is in physics the capacity of doing a work, there are different kinds of energy, example; potential, kinetic and thermal energy. In this project however when referring to energy it is electrical energy as in power (Watt) that is discussed, [5] [6].
Energy efficiency can be described as the relationship between consumed energy and the output of a machine. The less energy used to generate the same amount of output the more efficient the machine is. Example: A car that uses 0.7 liters of gasoline to drive 10km is less energy efficient as a car which only uses 0.5 liters of gasoline to drive the same length. [4]

The equation of calculating energy from the SI-Unit Joule to the SI-unit kilo Watt for Power is shown in equation(2.1).

$$1kW = \frac{Joule}{1000 \cdot seconds} \tag{2.1}$$

Since 1 hour corresponds to $60 \cdot 60 = 3600$ seconds therefore converting Joule to kWh is as followed in equation(2.2

$$1kWh = \frac{Joule}{3600000} \tag{2.2}$$

2.3 Controllers

For this project, different controllers need to be used. The old system today is using a built-in PID-controller where it is only needed to insert the gain, integration and derivation values for the controller, but it has not been used for a couple of years.

2.3.1 PID-control

A PID-control is controlled by a feedback and has as derivation part, an integration part and a proportional factor. In this project, only high-level control of the system will be necessary. This means that the PID controller will not be built, the only things that need to be decided is the *proportional, integral* and *derivative* gains, but also *deviation* and maximum and minimum value of the PID. This will be decided with simulations in PLCsim.

The gain is used to change how large impact the controller has on the output. The proportional gain (K_p) is multiplied with the error to deliver a fast feedback to bring the process value near the set point. The integral-gain (K_i) is also multiplied with the error but also multiplied with the cycle time of the controller. The slower the outcome gets to the set point the more impact the K_i has on the output. The deviation-gain (K_d) is multiplied with the ramp rate of the process value, the deviation is predicting where the output is heading and working against the P and I value, this helps the system to not over-shoot and makes it more stabilized. [**?**]
In Figure 2.5 a illustation of a simple PID controller is shown.

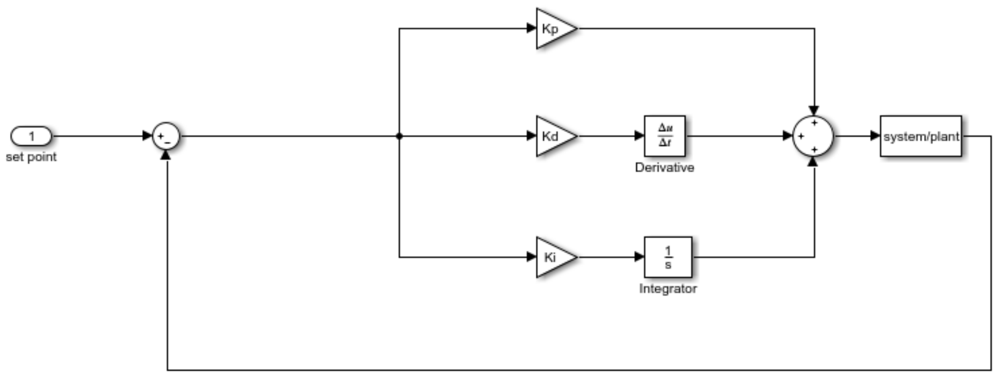


Figure 2.5: A simple PID controller

2.3.2 Cascade Control

In a cascade control architecture, there is an inner-loop and an outer-loop where the inner-loop will give a faster feedback to the controller and the outer-loop is for the entire control part. A cascade control is used to give the actual value a closer value to the set point and a faster feedback. Since this system has three almost alike subsystems a cascade controller would be an option, where the outer-loop is the control of the pressure and the main system and the inner-loop controls the main fans of each subsystem.[8]

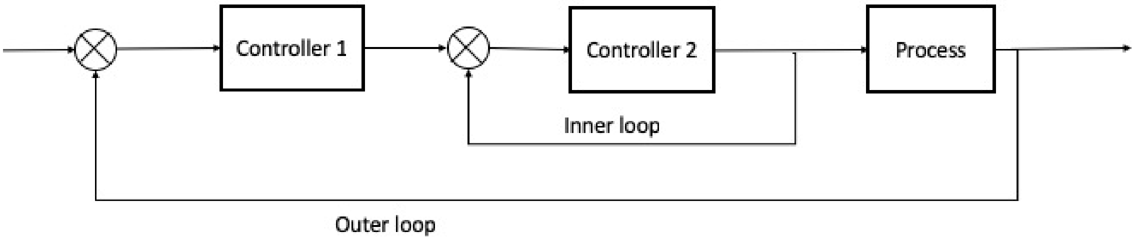


Figure 2.6: A simple figure of a Cascade Controller

2.4 Optimization

There are different methods of solving optimization problems, in this project mixed integer linear programming (MILP) will be used, which is an extension of linear programming (LP).

2.4.1 LP - Linear Programming

Linear Programming is a method used to model an optimization problem in order to find the best possible outcome given a set of constraints. The outcome is formulated by an *objective function*, (2.3) that is the part to maximize or minimize, i.e. the cost or profit. It also includes some constraints (2.4,2.5). There are some constants like the vector **c**, **A** and **b** and some decision variables, **x**, that has to be larger than zero to be linear programming.[9]

$$\textit{maximize} \quad \mathbf{c}^T\mathbf{x} \tag{2.3}$$

$$\textit{subject to}$$

$$\mathbf{Ax} \leq \mathbf{b} \tag{2.4}$$

$$\mathbf{x} \geq \mathbf{0} \tag{2.5}$$

2.4.2 MILP - Mixed integer programming

Mixed Integer Linear Programming is a combination of LP and IP (integer programming), the decision variable **x** can include both integers and real-values variables.

The problem is stated the same way, equation(2.3-2.5), can include the same constraints but without every variable needs to be real-valued [9]. A solver that can be used solving this kind of problems in MATLAB is `intlinprog`. `intlinprog` is a MILP solver that finds the minimum outcome of a problem stated as in equation (2.6) [10].

$$minimum \quad f^T x \quad subject \quad to \begin{cases} x \ integer \\ Ax \leq b \\ A_e x = b_e \\ l_b \leq x \leq u_b \end{cases} \tag{2.6}$$

In equation (2.6), x is the decision variable, A and b are constants when the constraint is an inequality and A_e and b_e is the constants when the constraints are an equality. l_b and u_b stands for the lower and upper bound the decision variable x can have.

In this project, the problem is modelled using integer variables, and to be more specific, only binary variables with 0-1 as their domain are used. The model is hence a special case of MILP, called IP. It can however be solved as usual with MATLAB's built-in MILP solver.

3

Case Study

The project included a large amount of pre-study on the existing system which resulted in the chapter *case study*. To be able to understand the system, its applications and explanations, a deeper explanation is necessary. When using different words such as `subsystem 09`, `mainfan 553` or `mainfan for subsystem 10` the reader can easily understand what it means.

3.1 The system today

To be able to understand the system that is going to be updated, a large case study was carried out. The zeolite filtering system `(52-35-002)`, where `52` stands for which building the system is in, and will not be used again. `35` is the number of the system and `002` is a designation of the entire system, including all of the consisting five sub-systems. The consisting five subsystems are; `35-009`, `35-010`, `35-020`, `35-030` and `35-060`. Where `subsystem 09` is in the beginning, `subsystem 60` is the end and `subsystem 10`,`subsystem 20` and `subsystem 30` are working in parallel in the middle.

In Figure 3.1 the subsystems are indicated with the green dotted lines. Moreover, in Table 3.1 the different components and their connections are explained.

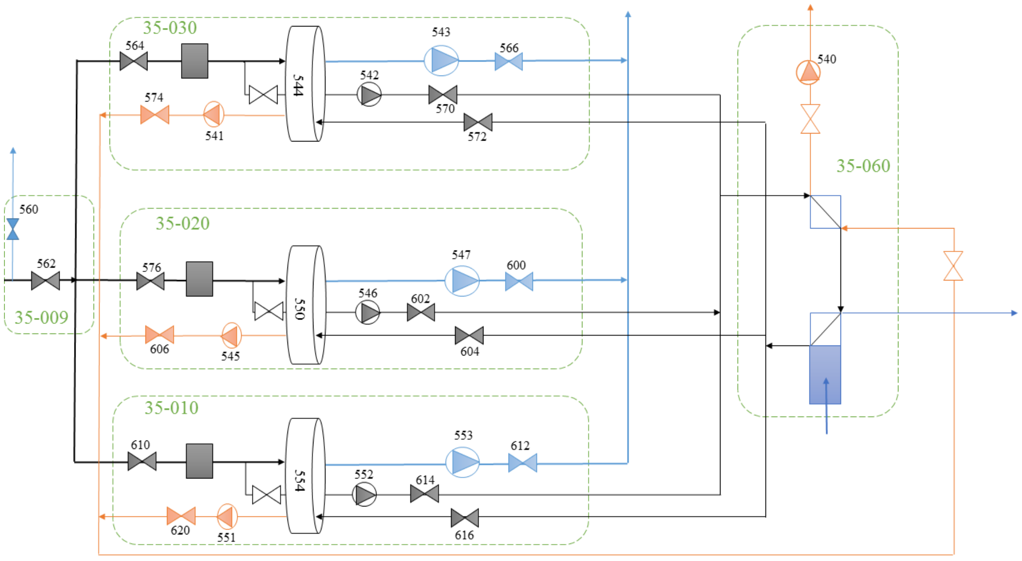


Figure 3.1: The main system including the five subsystems

As explained in the Introduction the subsystems `35-010, 35-020` and `35-030` are the subsystems that will be included and controlled in the optimization. However, all the throttles, fans and the zeolites can be controlled automatically, but they can only run/not run or open/close. But the three throttles that are white in Figure 3.1, can only be opened or closed manually, but they are usually opened.

3.1.1 `Subsystem 09`

The `subsystem 09` is the first subsystem and includes two throttles/valves, `560` and `562`, which can only be opened or closed. If throttle `562`(see Figure 3.1) is opened it means that the process air is supposed to move into the subsystems where the filtering system is. If throttle `562` is opened it means that throttle `560` (see Figure(3.1)) is closed. If throttle `562` is closed it means that `560` can be opened, and the process air that moves through throttle `560` is going directly out in the atmosphere. This throttle only opens if a fire alarm would appear. If `562` is opened `560` can't and vice versa. Figure 3.2 illustrates the connection further.

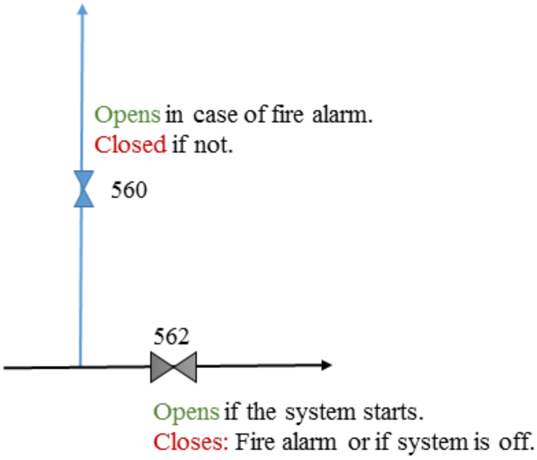


Figure 3.2: `Subsystem 09` with explanations on how the components are controlled

3.1.2 `Subsystem 10, 20 and 30`

The next subsystems are `10,20` and `30`, that includes the filtering processes and they are running in parallel doing the same thing. Each subsystem contains one motor that spins the zeolite filter, three different fans/motors, five valves that is controlled with motors and one last valve that is manually opened and closed.

When the SattCon was implemented 1990 there were more fume hoods, together with the process air, in use than it is today and the system was running at full speed on all three subsystems. A fume hood, which is also called fume cupboard and/or fume closet, is a locker with high ventilation which is usually used in laboratories dealing with hazardous chemicals. The fume hoods that are connected to system 35, this system, are the one that are usually used in laboratories, as the one illustrated in Figure 3.3. Moreover, there are also fume hoods connected that looks like a normal locker with high ventilation where it is possible to keep hazardous equipment and materials.

Figure 3.3: Image of a typical fume hood [11]

However Volvo realized that `subsystem 35-020` fan `547` took all the air from the other subsystems, the gearing is 8% higher to fan `547`, which results in that the fan has a 8% higher maximum speed, than fan `543` and `553`. But it is also a straight way for the air from the `subsystem 09`. To alleviate the problem, an implementation of a PID-controller was made, the PID is controlled by the pressure difference in `subsystem 09` before the two throttles, the same place where the new transmitter will be.

Testing different values and different transmitters on the three fans they finally decided that the transmitter was going to be implemented on fan `547`(`subsystem 35-020`) with a set point on 65% for the pressure difference. This was in 1994 and the number of fume hoods in use has decreased since, which means that the fan speed would be stronger than necessary.

Today, the entire procedure is manually controlled. In the beginning of the project fan `547` is set to run at 32.5Hz which corresponds to a speed of 70%. Fan `543` is running at maximum speed, while fan `553` is turned off together with `subsystem 35-010`. This corresponds to a pressure difference about 36-37%. The rest of the fans connected to subsystem `35-020` and `35-030` was running at full speed, except for the motors that spins the zeolites filters. All the throttles that are in the two subsystems are opened. In the end of the project fan `547` still runs at a speed of 70% but both the other main fans are running at full speed and all the subsystems are turned on.

When Volvo tested how good the zeolites takes up the particles, it was shown that when the motors runs at 15Hz which corresponds to a 30% speed the best filtration result was achieved. Therefore the motors `554`,`550`, `544` is currently running at a speed of 30% of maximum when turned on.

The valves that are connected to each subsystem is only opened when their main fans are on. The same is true with the motors for the fans and the zeolites rotors in the particular subsystem. This results in that the main fans are controlling if the other fans are on/off. The main fans can only be on if throttle `09-562` is opened. The main fans of each subsystem is shown in Table 3.1 and is `030-543`, `020-547` and `010-553`. Explanation on how the components in the three subsystems are connected with the other components are illustrated in Figure 3.4-3.6.

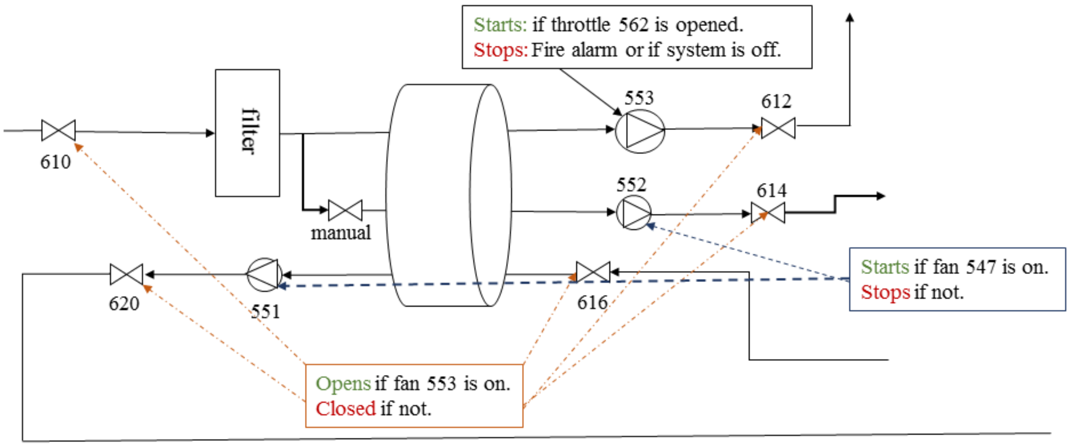


Figure 3.4: `Subsystem 10` with explanations on how the components are controlled

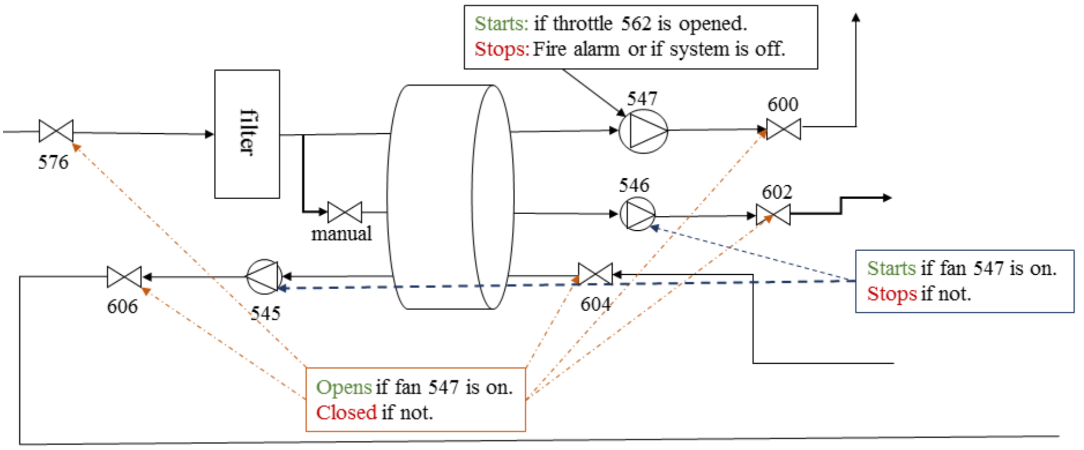


Figure 3.5: `Subsystem 20` explanation

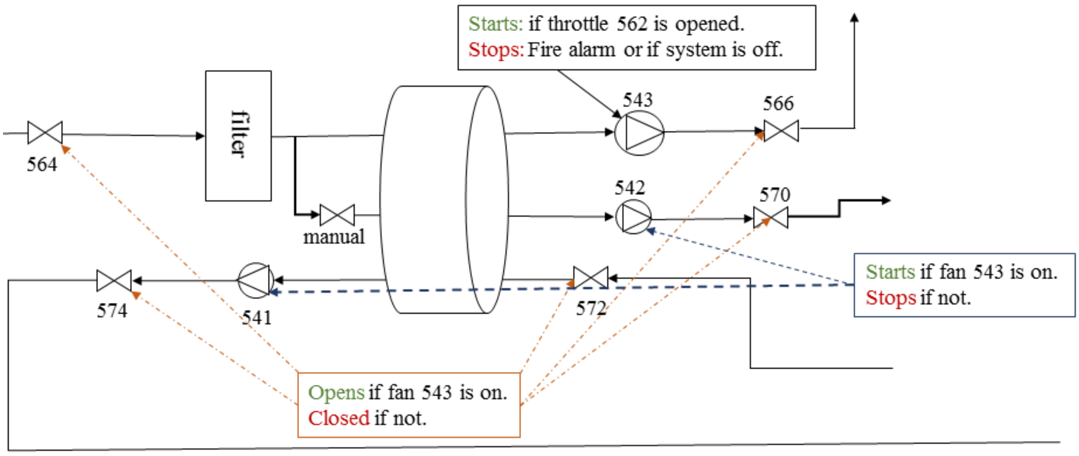


Figure 3.6: `Subsystem 30` explanation

3.1.3 `Subsystem 60`

The last system of the five subsystems is `subsystem 60`. It includes a heater that heating up the process air that is used in the backflow in the filter. It also includes a fan that help the air to get transported to the next system where the high solvent process air will be filtered once again. The heater only gets a signal from the PLC to turn on when the system is running and is not included in the system today. The fan `540` also gets a signal to start running at maximum when one of the mentioned main fans are on. In Figure 3.7 the explanation of when fan `540` is on/off is shown.

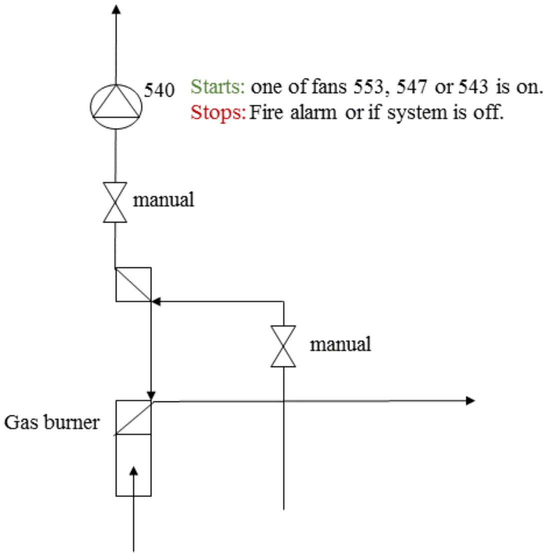


Figure 3.7: `Subsystem 60` explanation

3.1.4 Connection

The three subsystems are today manually turned on/off, and the set point to fan `547` is set and controlled on a transmitter. The new PLC will control all three main fans and will also control which subsystem that will be on/off, but will still be using the given logic that is shown in Figure 3.2, 3.4, 3.5, 3.6 and 3.7. In Table 3.1 the connection is explained as well.

The main fans of each subsystem controls the other fans in that subsystem and the fan in `subsystem 60`,(`540`). The valves in `subsystem 09` is controlling the main fans of each subsystem.

ID	Description
Subsystem 09	
`560` `562`	Throttle/Valve that is opened in case of fire alarm Throttle/Valve that let the process air into the filtering system
Subsystem 10	
`610`, `612`, `614`, `616`, `620`	Thottles/Valves that only opens if the mainfan `553` is running
`553`	Mainfan `553` is the fan that forces air to go through `subsystem 10` and only starts if throttle 562 is opened
`551`, `552`	Smaller fans that starts running when main fan `553` is on
`554`	The zeolite filter that is rotating.
Subsystem 20	
`576`, `600`, `602`, `604`, `606`	Thottles/Valves that only opens if the mainfan `547` is running
`547`	Mainfan `547` is the fan that forces air to go through `subsystem 20` and only starts if throttle 562 is opened
`545`, `546`	Smaller fans that starts running when main fan `547` is on
`550`	The zeolite filter that is rotating.
Subsystem 30	
`564`, `566`, `570`, `572`, `574`	Thottles/Valves that only opens if the mainfan `543` is running
`543`	Mainfan `543` is the fan that forces air to go through `subsystem 30` and only starts if throttle 562 is opened
`541`, `542`	Smaller fans that starts running when main fan `543` is on
`544`	The zeolite filter that is rotating.
Subsystem 60	
`540`	The fan that forwards the filtered air to the next filtering system and is only on if one of each main fans `543`, `547` or`553` is on

Table 3.1: Summary over the specifications of each subsystem/component

4

Methods

This chapter includes ideas and explanations on how the system was built and what has been implemented in the new PLC code and controller.

4.1 Optimization

The optimization is focusing on energy efficiency and the result will present how much energy the optimized system will consume compared to the energy the system consumes today.

No changes in the physical system will be made and subsystem 09 and 60 will always be on if the system is. The optimizer will therefore decide which subsystems between 10,20 and 30 that should run and on which speed fan `543`, `547` and `553` should spin at. The maximum speed on the motors and the fans are shown in Table 4.1, Moreover, in Table 4.2 the speeds at which motors are most efficient are given. However, to be able to evaluate the speeds, two experiments will be carried out. These experiments will probably result in how the energy consumption actually are and will be compared with the given energy efficiency the motors have.

Fan nr.	Max volume flow $[m^3/h]$	max running speed $[rpm]$	motor capacity
`553`	21 000	1540	$18.5kW$ $1470rpm$
`547`	22 800	1600	$18.5kW$ $1470rpm$
`543`	21 000	1540	$18.5kW$ $1470rpm$

Table 4.1: The capacity of the main fans according to the data sheet.[12]

4.1.1 Optimization model

For the decision on which fan to run at which speed, the energy of running each subsystem and the energy of running the three main fans at different speeds is taken into account. Using integer programming an objective and constraints are stated. Since the decision variables are binary, the function is not mixed integer linear programming. In the equation list 4.1 the decision variables and constants

Motor for fan nr	Perentage of speed	motor speed	fan speed	energy efficiency
553	50%	$735rpm$	$770rpm$	92.7
	70%	$1029rpm$	$1078rpm$	92.9
	100%	$1470rpm$	$1540rpm$	91.9
547	50%	$735rpm$	$800rpm$	92.7
	70%	$1029rpm$	$1120rpm$	92.9
	100%	$1470rpm$	$1600rpm$	91.9
543	50%	$735rpm$	$770rpm$	92.7
	70%	$1029rpm$	$1078rpm$	92.9
	100%	$1470rpm$	$1540rpm$	91.9

Table 4.2: The energy efficiency of the motors of each speed according to the motor sheet [12]

are stated.

$$\begin{cases} y(i,j) = \begin{cases} 1 \text{ if the fan in subsystem i is running at the given speed j.} \\ 0 \text{ otherwise} \end{cases} \\ x(i) = \begin{cases} 1 \text{ if subsystem i is running} \\ 0 \text{ otherwise} \end{cases} \\ f(i) = \text{The maximum flow of fan in subsystem i} \\ E(i) = \text{Energy of running subsystem i} \\ Ep_{540} = \text{Energy of running fan } \texttt{540} \\ e(i,j) = \text{Energy of running the fan in subsystem i at speed j} \\ a_{inflow} = \text{the actual inflow into the system} \\ I \in \begin{bmatrix} 1, & 2, & 3 \end{bmatrix} \\ J \in \begin{bmatrix} 1, & 2, & 3, & 4, & 5, & 6 \end{bmatrix} \end{cases} \tag{4.1}$$

$$min \sum_{i \in I} (\sum_{j \in J} e(i,j)y(i,j) + E(i)x(i) + Ep_{540}) \tag{4.2}$$

subject to:

$$x(i) - \sum_{j \in J} y(i,j) = 0 \qquad i \in I \tag{4.3}$$

$$x(2) \geq x(3) \geq x(1) \tag{4.4}$$

$$\sum_{i \in I} f(i)(y(i,1) + 0.9y(i,2) + 0.8y(i,3) + 0.7y(i,4) + 0.6y(i,5) + 0.5y(i,6)) \geq a_{inflow} \tag{4.5}$$

The objective function 4.2 is minimizing the total energy cost of running the system to reach the specific flow, a. In the real system the flow will not be measured. But the optimization will give a good estimation when a specific speed from the fans are needed, depending on the pressure instead of the flow, how to choose the right speed and number of fans.

The second constraint 4.3, ensure that if $x(i) = 1$ then only one of y(i,j) could be equal to 1, and if x(i)=0 then the sum of y(i,j) has to be zero as well. This constraint guarantees that the subsystem cannot be turned on unless the controlled fan is on or reversed. The constraint also ensure that it is not possible to run each fan with more than one speed at the same time. For example, if $y(1,1) = 1$ it means that the controlled fan of subsystem 10 is running at a 100% speed and that means that $y(1,2) = y(1,3) = y(1,4) = y(1,5) = y(1,6) = 0$. In this case $x(1)$ has to be equal to one for the constraint to be fulfilled.

Constraint 4.4 is making the model start the subsystems in a specific order. This to make sure that the optimized model will not switch which subsystem to run when only one or two is necessary. In this case, `subsystem 20` always starts, since that fan is a few percentage stronger. In the beginning of the project fan `030-543` was the second fan/subsystem that was on together with `020-547` therefor the constraint is chosen in the same way. First `subystem 20` starts, then `subsystem 30` and last `subsystem 10` shall start.

The last constraint, 4.5, is the constraint that is depending on the inflow. It says that the sum of the three subsystems capacity with a specific speed on their main fans needs to be larger than the actual inflow, this constraint makes it possible to give a value of when a specific fan need to be run or not to reach the capacity that is necessary.

The solver that will be used is Matlabs solver `intlinprog` where the objective stated above is implemented. The code for the optimization can be seen in Appendix A.

4.1.2 Experiment with the energy

To be able to optimize the system with realistic values two different experiments had to be done. One for what the system is consuming today and one for how much motor 547 is consuming. The different experiments was made to generate, as mentioned, realistic values of the constants $e(i,j)$, $E(i)$ and Ep_{540} for the optimization model.

4.1.2.1 Entire system consumption

The electricity locker that is connected to `system 35` has an energy consumption gauge attached to it. From that the operator/user can read how much energy in kWh that has been consumed since last reset. The test includes looking at the barometer a couple of times and write the time and value down and then compare different values to see the energy consumption during a specific time.

During the experiment all the subsystems where on, the main fans of subsystem 10 and 30 was at maximum speed and the main fan of subsystem 20 was at 35.2Hz which corresponds to a speed of 70%. The zeolite motors, that makes the rotor to spin, are always running at a frequency of 15Hz, which corresponds to a speed around 30% of the maximum speed, if the subsystem is on.

The experiment will give a result of how much energy that is being consumed by the system today, the gauge is showing the consumed energy in kWh.

4.1.2.2 Consumption of energy on different frequencies

To be able to see how much energy the main fans are consuming when running on different frequencies a test had to be carried out. The goal is to measure the power consumption on fan `547`, when the frequency to the motors raises with 1Hz at each time from 25Hz (50%) to 50Hz (100%). The result of the different values of consumption depending on the frequency are implemented in the optimization. The result is the energy consumption of running one of the main fans at a specific speed, $e(i, j)$. This implementation gives the optimization a reliable consumption value of energy. The optimization result will be implemented in the inner-loop of the cascade controller

4.2 Modelling

The main focus of the modelling is the PLC building in the TIA-portal (see Definitions). The construction of the controller is also a big part of the modelling, where a result can be presented. The controller is based on the optimization and the pressure difference given from the transmitter that is calculated with focus on energy efficiency.

4.2.1 Frequency controller

For this project a cascade controller was implemented, there will only be one PID-controller that is based on the input of the pressure difference. The output of the PID will be a scaled value between 0-1, u, which corresponds to a value between 0-100%. The scaled value goes into the inner-loop of the controller where the optimization is implemented. The output of the inner-loop is a set point to the speeds of the three different main fans. The output of the entire cascade controller is depending on what is the most energy efficient/saving and at the same time obtain and keep the pressure set point. The idea of how this controller looks like can be seen in Figure 4.1

The idea of the frequency scaling is that the output of the PID,u, controls the reference input to the fans. For example, if the PID is giving a percentage output between 0-33.33% that will be scaled to give an input to fan `547` to run between 0-100%, if the output is between 33.34-66.66% the PID controls fan `543` and lock fan `547` at a specific speed. If the output of the PID is from 66.67-100% both `543` and `547` is locked to a value from the generated lock-up table, but the PID is controlling fan `553` instead.

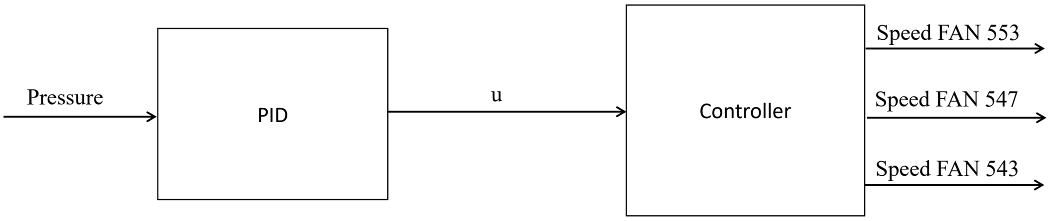


Figure 4.1: How the model of the cascade controller will look like

The PID is from the existing Volvo library and the decision on the parameters of the PID will be made based on simulations using PLCSim (see Definitions).

4.3 Implementation

The main part of the implementation of the system was to take the code from the old system, that was coded as a *statement list*, and convert it to the new system as *function blocks*. The new PLC from Siemens is going to be a *SIMATIC S7-1500*.

4.3.1 PLC programming

The PLC programming in Siemens TIA-portal was divided into three steps, first step was the illustrations and connecting images with the HMI panel (see Definitions), and PLCtags so that when running the system, the screen shows what is on, what is off and if there are any alarms. The second step was to build all the motor blocks and code for them to run, open/close depending on what kind of motor and then connect them to the same PLCtag that the images on the screens have. After that in this case the controller was obtained. The fourth part was to code the start and stop sequences, in which order each motor will start/stop, open/close when the system is started or shutdown. The last part was the structure and creation of the AK/AE locker where all the alarms are obtained.

4.4 Validation

In the validation the PLC code was tested together with the PID controllers response to different pressures using TIA portals, PLCsim. In the PLCsim it is possible to both test how the PID corresponds to different pressure values and test if the programmed code is working or not.

5

Results & Discussion

In this chapter the result is stated together with a discussion on how reliable and how well the result is.

5.1 Optimization

The optimization with focus on energy efficiency includes parameters that must be pre-calculated using two experiments as mentioned in section 4.1.1. After finding the parameters, solving the optimization problem using `intlinprog` in Matlab, the results gave an optimized policy of running the fans. The policy can be encoded as look-up table and be fed in the inner-loop of the cascade controller.

5.1.1 Entire system consumption

The collecting of data for the first experiment was made during two different occasions where the energy consumption was checked various times during around 24 hours. During the experiment all three subsystems (`52-010,52-020,52-030`) plus the additional two subsystems where turned on (`52-009,52-060`). All the motors except `547` and the three motors that drives the zeolites were running at maximum speed. The zeolites were only running at 30% of maximum speed which corresponded to a consumption of 0.55kWh for each one of the three motors. Fan `547` was running at a speed of 70%.

In Table 5.1 and 5.2, the third column indicates the energy consumption since the last reset of the energy measurement device. The fourth column gives the absolute energy consumption since the start of the measurements.

Time	Time since first check (hours)	Energy consumption since reset (kW)	Energy consumption since the first check(kW)
24/4 12:45	0	742711	0
24/4 13:00	0.25	742721	10
24/4 13:15	0.5	742733	22
24/4 13:30	0.75	742745	34
24/4 13:45	1	742757	46
25/4 15:03	26.3	7443959	1248

Table 5.1: The result of energy consumption of the entire system during the first test

Time	Time since first check (hours)	Energy consumption since reset (kW)	Energy consumption since the first check(kW)
29/4 13:15	0	748448	0
29/4 14:15	1	748496	48
29/4 14:30	1.25	748508	60
30/4 10:07	21.133	749446	998
30/4 12:37	23.3667	749565	1117
30/4 12:52	23.6167	749577	1129
30/4 13:00	23.75	749583	1135
30/4 13:15	24	749595	1147

Table 5.2: The result of energy consumption of the entire system during the second test

In Figure 5.1 a plot is illustrating the consumption of energy over time from the first check for the two test. As seen in the figure, the consumption is linear.

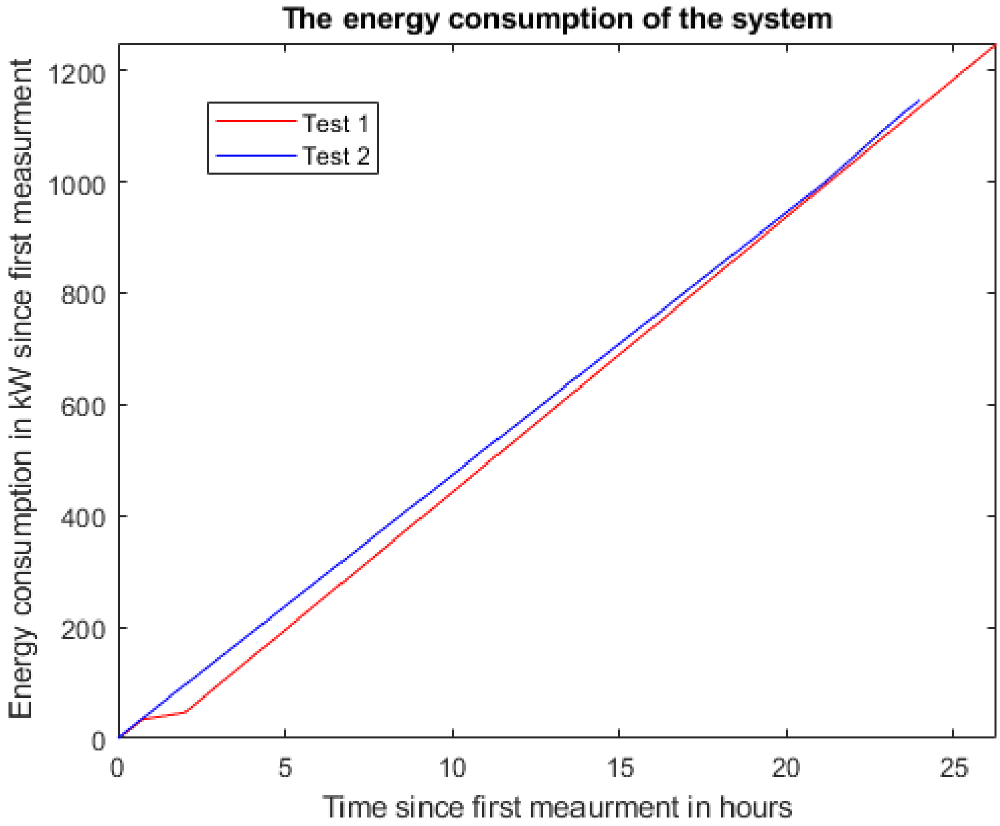


Figure 5.1: The energy consumption for the system over time

The average of energy consumption for one hour using the values from Table 5.1 & 5.2 is shown in Table 5.3, where the $\frac{Test1+Test2}{2}$. is adding the two calculated average of test 1 and test 2 and calculating the average of the two test occasions using $\frac{Test1+Test2}{2}$.

The Test 1 & Test 2 combined, the fourth column, is looking at the two tests as one test, the start value of the "new" test is the start value from test 1 and the last value of the "new" test is the last value of test 2. The hour of test 2 is added with 120.5 hours, this corresponds to that the combined test is about 144.5 hours long in total. This means that the assumption that the system has been on and no one has changed any settings during these days has to be made.

	Test 1	Test 2	$\frac{Test1+Test2}{2}$	Test 1 & 2 combined
Average	44.5572kW	47.77kW	46.43kW	46.44kW
Average with modification	45.70kW	47.77kW	47.02kW	46.98kW

Table 5.3: The average of the two different tests, both combined and separated

In Table 5.3 the modified average is the average without the second measurement from test 1,(Table 5.1 row 2), since that value was a lot smaller then the others. It could be that either the value was read too early or some disturbance in the system, but the value decreased the average a lot so the average of energy consumption will be calculated without it.

However, it can be seen that the average when test 1 and 2 are seen as one long test, does not have a large impact on the value. The assumption that was made earlier can therefore be trusted and is correct. The conclusion of this test is that the energy consumption for one hour for the entire system when running at the specific speeds, is around $47kWh$.

Furthermore since the values between the different occasions is not always $47kWh$ the average is an average and can not be seen as the exact consumption of the system.

5.1.2 Energy consumption of fan `547`

The energy consumption of running fan `547` at different speeds using different frequencies is shown in Table 5.4.

The consumption varied during the test depending on going up in the frequencies or going down. For example when going up 1Hz from 43Hz to 44Hz, the consumption first jumped to about 12kW and then later settled to around 7-8.5kW, both times that step was made. When going down from 45Hz to 44Hz the consumption showed around 6.5-6.8kW. These values results in that the result did not gave an exact value on how much energy it cost to run main fan `547` at different speeds. Moreover, this does not mean that the energy consumption of running the motor/fan at a specific frequency do not have an exact consumption.

For this project the values from the experiment will be used in the optimization. The values in Table 5.4 is an estimation of the values that was shown during the second experiment.

Frequency (Hz)	Speed (%)	Power (kW)		Frequency (Hz)	Speed (%)	Power (kW)
25	50	1.2		38	76	6.2
26	52	1.3		39	78	6.7
27	54	1.3		40	80	7.0
28	56	1.5		41	82	7.1
29	58	1.9		42	84	7.3
30	60	2.3		43	86	7.3
31	62	2.7		44	88	7.4
32	64	3.2		45	90	7.45
33	66	3.5		46	92	7.5
34	68	4.0		47	94	7.6
35	70	4.5		48	96	7.7
36	72	5.0		49	98	8.0
37	74	5.6		50	100	8.5

Table 5.4: The energy consumption of running fan `547` at a specific frequency

In Table 4.2 the energy efficiency of the motors was stated on three different speeds, 50%, 70% and 100%, the Table says that the motor is as most energy efficient at 70%, less efficient at 50% and least efficient at 100%. The experiment that was made on the motor for fan `547` gave another result. As shown in Table 5.4 the motor is consuming the smallest amount of energy when running at 50%,(25Hz) and it also consumes the most energy at 100% speed, (50Hz). The conclusion is that the sheet and experiment does not give the same result. One more thing that is notable though, is that according to the data sheets for the main fans, [12], is that the energy efficiency is based on running at different speeds when the frequency is at 50Hz. In the test however, the control of the motor was with the help from reducing or increasing the frequency but not changing the motors speed, i.e. it is always running at 100% but the frequency is changing. The reason why it is chosen to control the fans using the frequency is because it was a request from Volvo Cars.

Since the experimental value is directly from the system and is not only depending on a data sheet, the experimental result that is shown in Table 5.4 is therefor the values implemented in the optimization. Even though the experimental values was not completely accurate since the energy consumption value was not the same when increasing the frequency as when decreasing the frequency to the same value. This could also be why the data sheet and the experiment did not get the same result. However it is hard to see that the value when running at 100% would almost drop with 4kW and when running at 50% would increase with 2.5kW since 50% was told to be less efficient.

Moreover, since energy efficiency and energy consumption is not the exact synonym to each other, as explained in the Theory Chapter 2.2. That is most likely the reason why the result differs between the experiment and the data sheet.

The reducing of the energy consumption without reducing the output is what the optimization wants to achieve, to get the most efficient way of running the system.

In Table 5.4 the consumption of energy of fan `547` when running at different speeds are stated. An assumption that the energy is the same for all the three main fans since they have the same type of motor that controls them is made. With this assumption a table can be stated on how much energy each component is costing the system today. This is done using the results from the second experiment together with the data sheets on the remaining motors and the energy the zeolite motors consumed according to their transmitter.

5.1.3 Energy consumption with the experiments

The maximum consumption of each component in the system is taken from their specifications and are shown in Table 5.5. In Table 5.6 the actual usage of power is shown of each component during the test, the values for fan `553,547,543` is taken from the experiment result shown in Table 5.4. The motors for the valves are only added when a subsystem is starting or stopping and is only $.0015 \times 5 = .075kW$ which can be disregarded since the impact on the system is rare and very small.

FAN Nr	`543/547/553`	`541/545/551`	`542/546/552`	`560`	`544/550/554`
Maximum consumption in kWh	18.5	1.5	5.5	3	0.55

Table 5.5: The maximum consumption of each motor according to the data sheets.[12][13][14]

FAN Nr	`543/553`	`547`	`541/545/551`	`542/546/552`	`560`	`544/550/554`	total
Consumption in kWh	8.5	4.5	1.5	5.5	3	0.55	47.05 kWh

Table 5.6: The actual consumption of energy for each motor in the system

As seen in Table 5.6 the total amount of energy consumption in kWh is a bit over the average, this is most likely depending on that the effect of running the three main fans at a different speed is not a solid value. The energy consumption is jumping between numbers and the values in Table 5.4 is an average of the actual consumption running at a specific speed.

5.1.4 Optimization

Using the stated objective function (4.2-4.5) together with the values collected from the experiments and data sheets. f(i) from Table 4.1 and e(i,j) given from the generated Table 5.4 from the experiment. Also using the total cost of running a subsystem $E(i) = 1.5 + 5.5 + 0.55 = 7.55kW$, together with the energy cost of running fan `540`, $Ep_{540} = 3kW$. The values from running each subsystem and fan `540` is taken from the result in Table 5.6. As stated in subsection 4.1.1, there will be six different speeds, $50\%, 60\%.70\%, 80\%, 90\%$ and 100%, that is included in the optimization. The following Table 5.7 shows the result from the optimization in MATLAB. It presents which subsystem to run and on which speed.

inflow	running			speed			energy cost kW
m^3/h	fan 553	fan 547	fan 543	fan 553	fan 547	fan 543	
15000	no	yes	no		70%		12.05
17000	no	yes	no		80%		14.50
21000	no	yes	no		100%		16.05
23000	no	yes	yes		60%	50%	18.60
26000	no	yes	yes		60%	60%	19.70
27000	no	yes	yes		70%	60%	21.90
29000	no	yes	yes		90%	50%	23.75
32000	no	yes	yes		50%	100%	24.80
35000	no	yes	yes		100%	60%	25.90
37000	no	yes	yes		100%	70%	28.10
40000	no	yes	yes		100%	90%	31.05
42000	no	yes	yes		100%	100%	32.1
45000	yes	yes	yes	50%	100%	60%	34.65
47000	yes	yes	yes	60%	100%	60%	35.75
50000	yes	yes	yes	70%	100%	60%	37.95
55000	yes	yes	yes	100%	60%	100%	41.95
58000	yes	yes	yes	100%	70%	100%	44.15
60000	yes	yes	yes	90%	100%	90%	46.05
64800	yes	yes	yes	100%	100%	100%	48.15

Table 5.7: The result of the optimization depending on different inflows values

In Table 5.7 the different values on which fan to run and when according to the optimization is stated. An interesting fact is that the result from the optimization says that when it is necessary with a speed of 270% of the total amount of the fans, the best way is to run two fans at 100% and one at 70%. Which means that the old system was running at the most efficient way if that specific speed had been necessary. However, the old code did not have any set point on what the pressure should be, and therefore the new optimized version is better than the old one. Mostly because the preferred pressure might not be reached with the old system, but is most likely obtained using the optimized and new code.

However the energy consumption of running at that speed according to the optimization is around $44.15kW$. Which is around $3kW$ lower then the gauge on the electrical locker said, i.e. what the experiment in Table 4.2 show. Why the optimization is lower/better than the real life experiment is, could depend on a lot of factors. However, it is most reasonable to think it is because the values that are being used in the optimization are from experiments and because of that, as discussed earlier in this Chapter, can not be seen as the absolute truth of the consumption values.

The energy consumption of the fans that are not the main fans, is not tested in the real system, they are only from data sheets. The value from the experiment and from the gauge are just an average and does not necessarily have to be the correct value.

The inner loop of the controller is deciding which motor to run at which speed depending on the PID value,u, the optimization values are implemented there. But some of the values, like when the speed jumps between the flow 26000 and 37000 the speed was configured between fan `547` and `543` to give a more smoother controller and the PID value ,u(see Figure 5.7), is only controlling one of the main fans at a time.

In Figure 5.2 the minimum energy consumption depending on the inflow is shown. The energy consumption is almost linear, which is legit since the subsystems are based on the same energy consumption and the flows are stated from slowest to fastest.

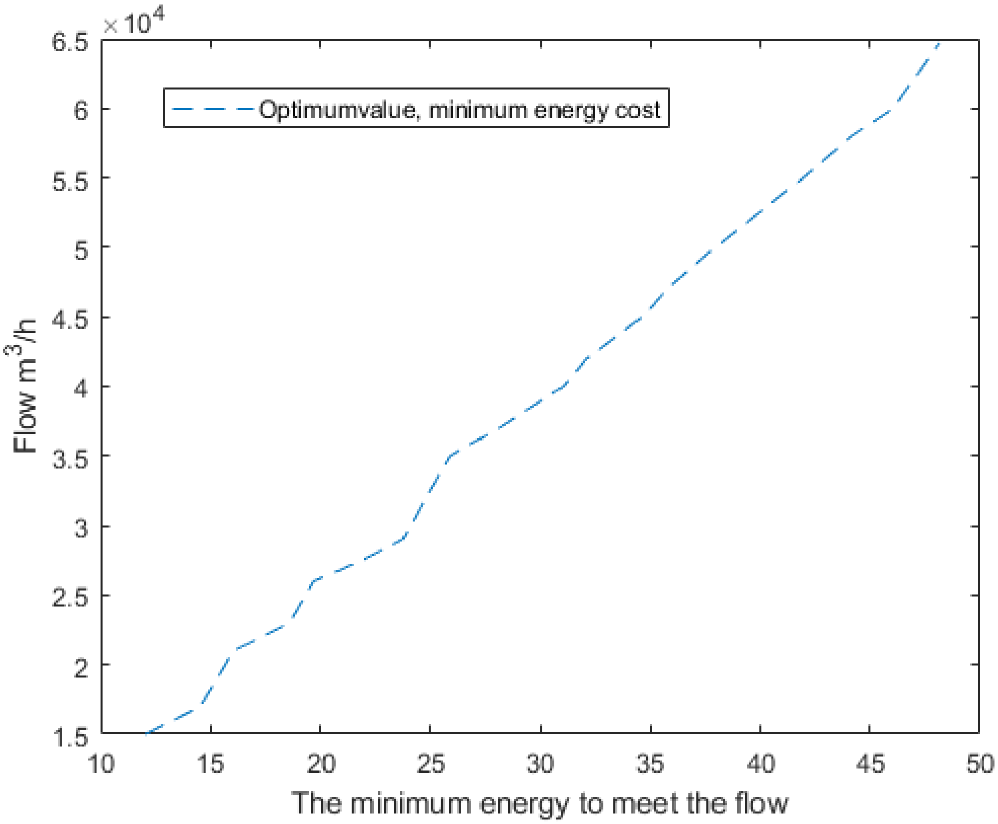
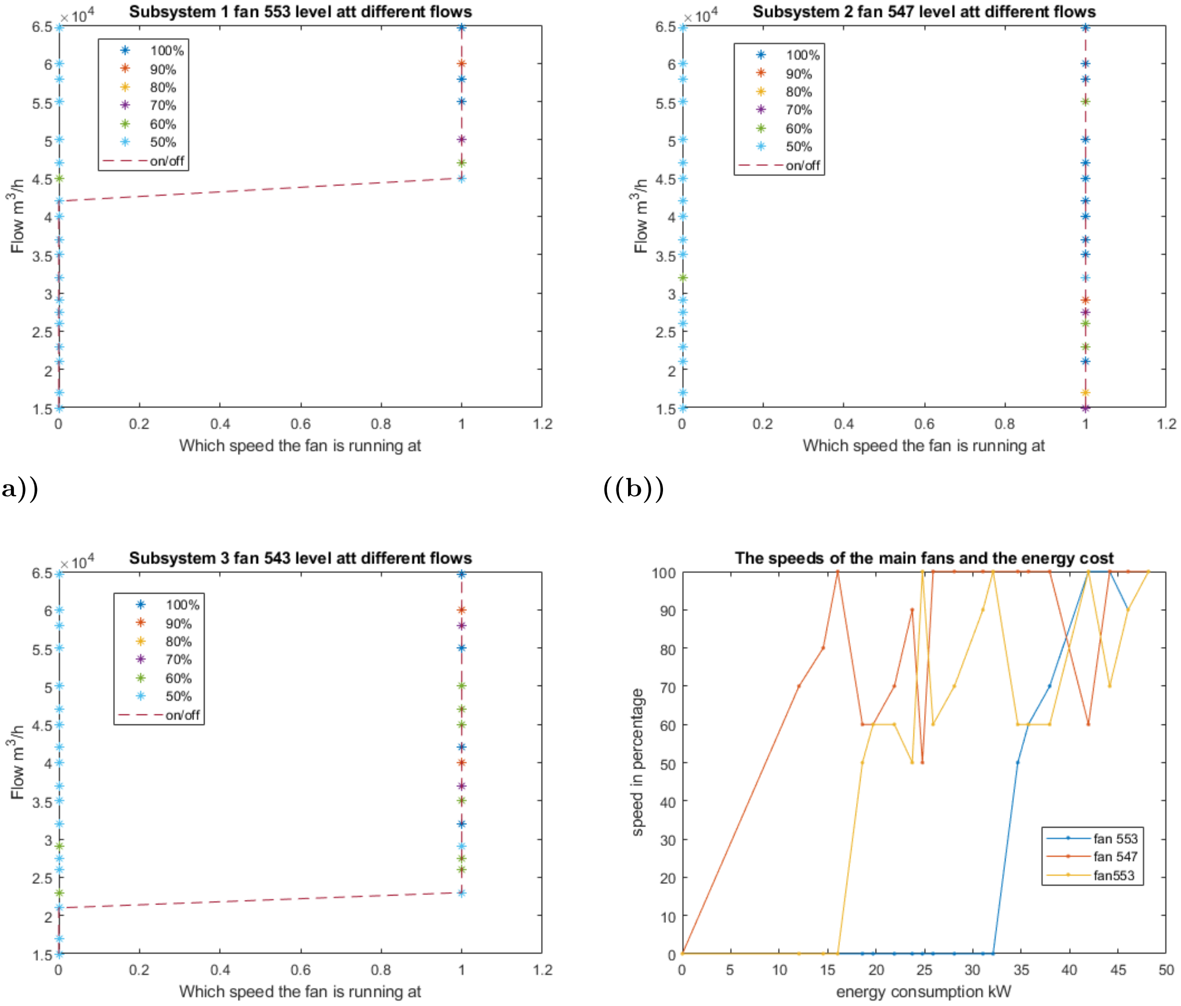


Figure 5.2: The minimum cost of energy to meet the necessary speed in kW



((a)) ((b))

((c)) ((d))

Figure 5.3: a)-c) The necessary fan speed to meet the inflow and d) the speed and the energy consumption

In Figure 5.3,a), b) and c), the three main fans speed are illustrated depending on the inflow. Figure a) is subsystem 10 and fan `553`, figure b) is subsystem 20 and fan `547` and figure c) is for subsystem 30 and fan `543`. The last figure d) is the combined speeds on the fans according to the optimization. The graph illustrates the same result as in Table 5.7 but more graphically. 1 in the graph stands for on and 0 stands for off. The dotted line is showing if the subsystem is turned on or off. The points/starts are equal to one if the main fan is running at that speed. It is zero otherwise, and only one of the six different speeds can be 1 at a time.

5.2 Model och Implementation

The result of the modelling and implementation of the controller and the code was made successfully after many trials and errors

5.2.1 Control

The controller as mentioned in Section 4.2 is a cascade controller with the pressure difference as the input to the PID. The output of the PID is a fractional number between 0 and 1, that corresponds to 0-100% of the speed that is necessary. That value is mentioned as u in Figure 4.1. The value goes into the frequency scaling, the inner-loop of the cascade controller, that gives an input to each of the main fans on what speed they should run at, with the help from the optimization.

5.2.1.1 Real system controller

The controller in the PLC is a PID controller that controls the frequencies of the three main fans. In Figure 4.1 the idea of the controller that is coded below is shown. In the real system a feedback is sent to the controller of the real speeds on the motors. The feedback gives the system a chance to keep the existing speed in case of a disturbance or a smaller change in the pressure. For example a time-delay can be implemented to give the controller the opportunity to not start a subsystem unless it is absolutely necessary. Moreover, since the pressure most likely won't change often or whitin a large interval when the system has been running for a while. Which would depend on the fact that the input to the system does not change that often or with larger disturbances. The timers and the feedback might be unnecessarily, but further testing should be made to be sure.

In Figure 5.4 the actual controller of the inner-loop is shown, the input is the percentage value from the PID and the speed of the main fans. The outputs are the reference speed to the main fans and three signals that tells the subsystems other components to be on/off or open/closed. $0x0\_SYSGO$ is the signal that tells the system if they should be on/off. *Referens0x0* is the signal that gives the set point on what frequency the motor should run at. The input *Frekvensvärdex0* is the feedback on what speed the motor actual has. PID_IN_Ref is the fractal number, u, that controls the function block as explained earlier in this section.

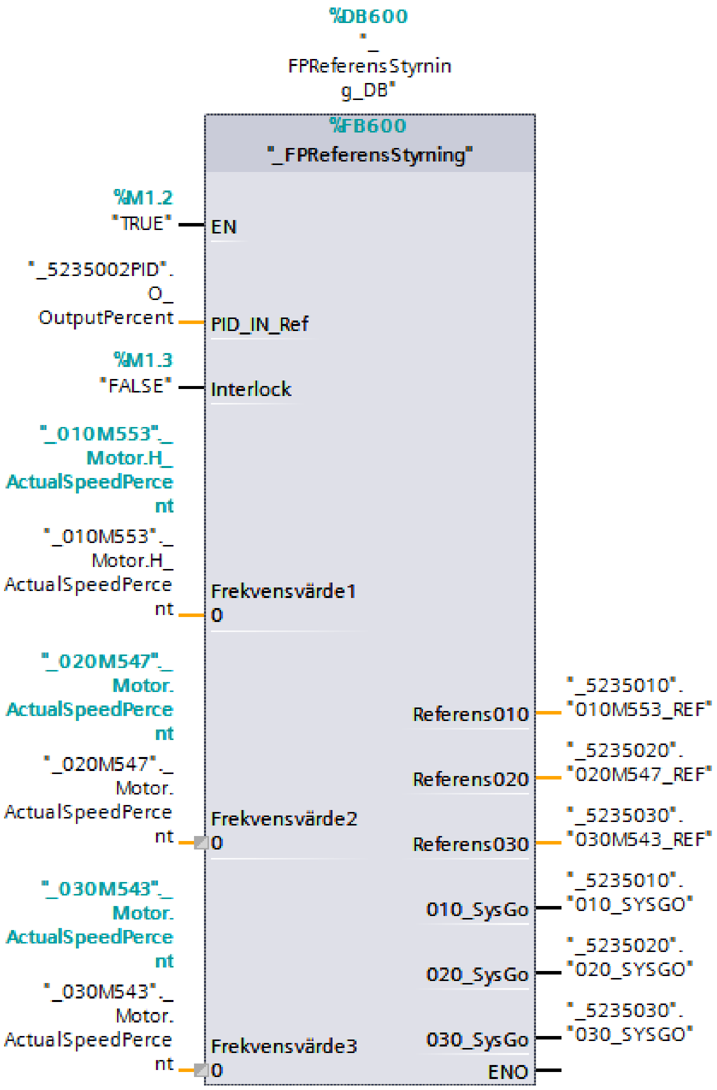


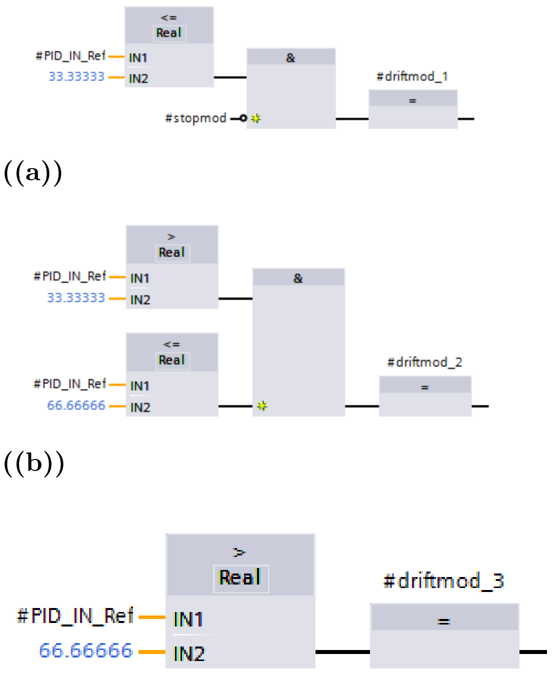
Figure 5.4: How the inner-loop of the controller looks like in the system

The decision on which fan to run at which speed is given from Table 5.7. The result from the optimization needed some minor tuning, as explained earlier, and a look-up table was generated from that. The look-up table depends on the output from the PID and was implemented in the inner-loop of the cascade. The look-up table is shown in Table 5.8.

In Figure 5.5, the code to chose which fan to run depending of the input u is shown, where *a)* is for fan `547`, *b)* is for fan `543` and *c)* is for fan `553`.

	Fan			Energy
PID output, u	553	547	543	consumption
0-33.33%	0	$u \times 3$	0	12.05-16.05
33.34-39.99%	0	60%	$(u - 33.33) \times 3 + 40\%$	16.05-19.7
40-43.33%	0	70%	$(u - 33.33) \times 3 + 30\%$	19.7-21.9
43.34-46.67%	0	90%	$(u - 33.33) \times 3 + 10\%$	21.9-23.75
46.67-66.66%	0	100%	$(u - 33.33) \times 3$	23.75-28.1
66.67-73.33%	$(u - 66.66) \times 3 + 50\%$	100%	50%	28.1-35.75
73.34-76.66%	$(u - 66.66) \times 3 + 40\%$	100%	60%	35.75-37.95
76.67-93.33%	$(u - 66.66) \times 3 + 10\%$	100%	90%	37.95-46.05
93.34-100%	$(u - 66.66) \times 3$	100%	100%	46.06-48.15

Table 5.8: How the output from the PID, u controls the subsystems and main fans



((a))

((b))

((c))

Figure 5.5: The decision code on which fan to control

The decision on which *driftmod* that is deciding over the fans or not is depending on the input u, how u then is controlling the main fans is shown in Figure 5.6 for *driftmod1*, which happens when $u \leq 1/3$, Figure 5.7 - 5.8 for *driftmod2*, when $1/3 \leq u \leq 2/3$ and Figure 5.9 - 5.10 for *driftmod3*, when $u \geq 2/3$. *Driftmod1* is controlling the speed of fan 547, the other two fans are turned off. *Driftmod2* is controlling the speed of fan 547 and 543, where fan 547 gets a specific value, that is depending on the input u but the fan 543 is changing with the input u whitin a specific interval of u. *Driftmod3* is controlling all the fans with the input on u deciding the speeds and fan 553 is the fan that changes during the specific interval of u.

In Table 5.8 the actual speeds of the fans are shown, how the drift modes are changing is shown in Table 5.9 as a complement.

		Fan		
PID output, u	drift mode	553	547	543
0-33.33%	1	0	changing with u	0
33.34-66.6%	2	0	fixed value	changing with u
66.67-100%	3	changing with u	fixed value	fixed value

Table 5.9: How the drift mode is chosen depending on u

In Figure 5.6-5.10 the scaling of the output is the #Referens0x0 which is the reference for each fan at each subsystem, where 10 is for subsystem 10 and therefor fan 553, 20 is for fan 547 and 30 is the reference signal to fan 543.

The function block $NORM_X$ is converting the input *u/PID_IN_Ref* to a corresponding value between 0-100%. MUL is multiplying the value with 100 so that the output will go between 0-100. The function block $MOVE$ is setting the output to the input value and ADD is adding the two inputs and return an output.

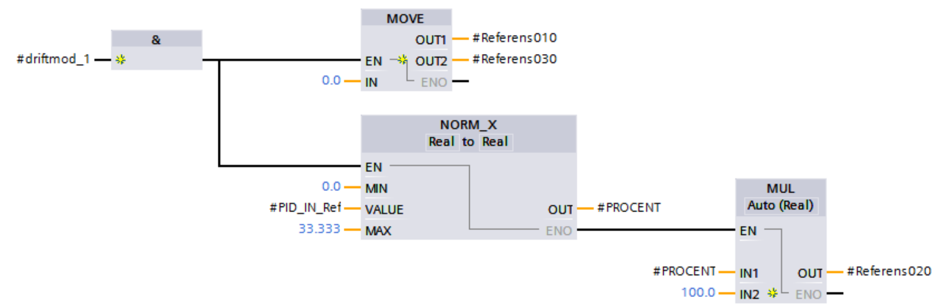


Figure 5.6: How the fans are controlled when they are in #driftmod_1

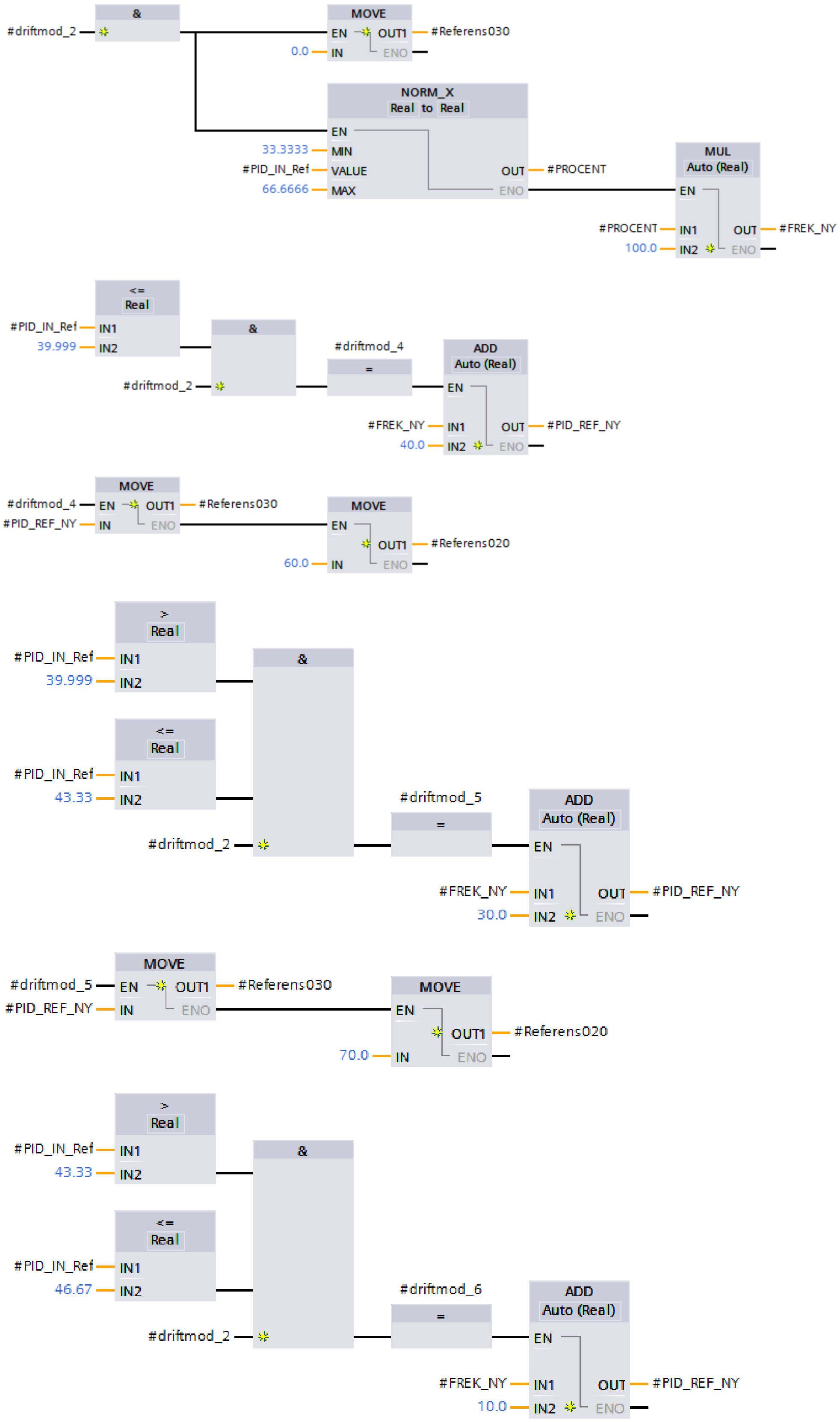


Figure 5.7: How the fans are controlled when they are in #driftmod_2

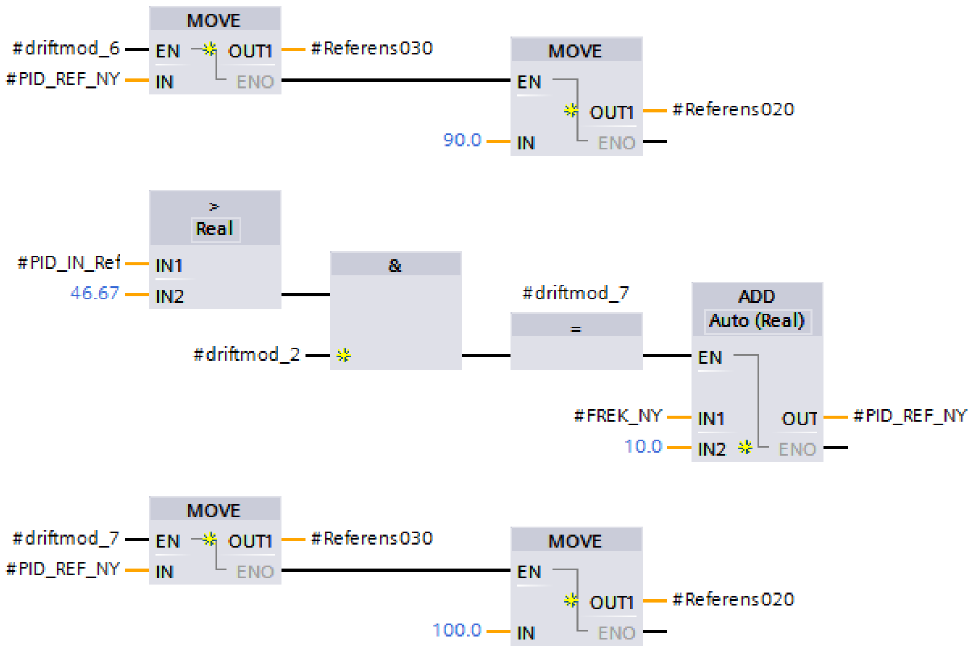


Figure 5.8: How the fans are controlled when they are in #driftmod_2

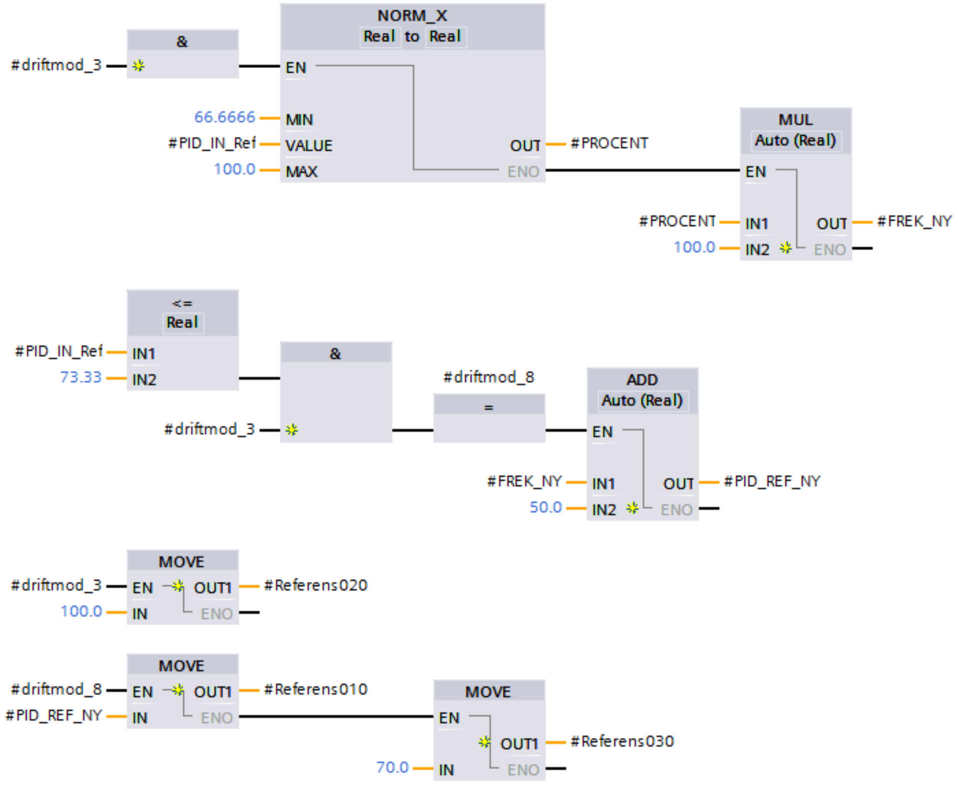


Figure 5.9: How the fans are controlled when they are in #driftmod_3

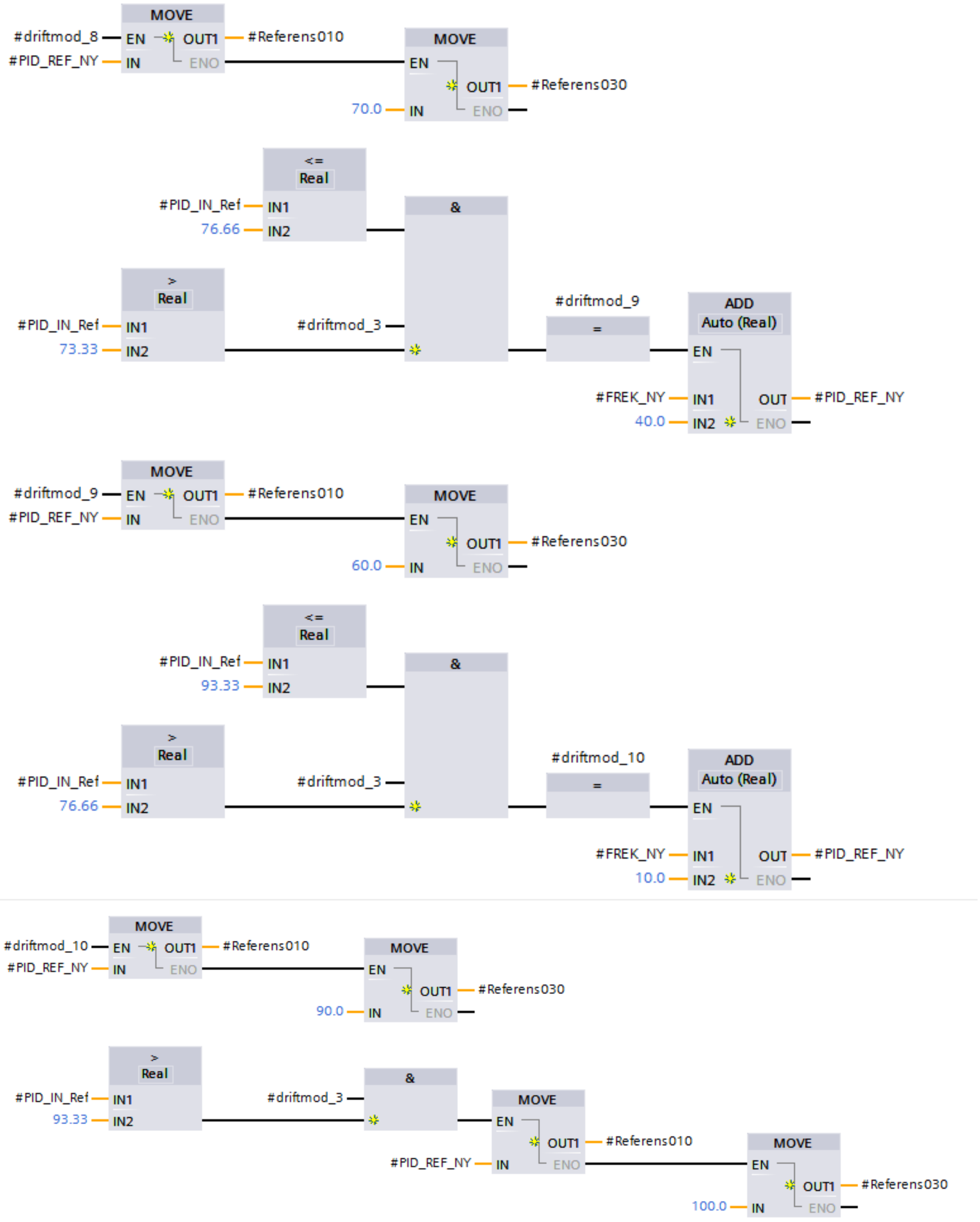


Figure 5.10: How the fans are controlled when they are in #driftmod_3

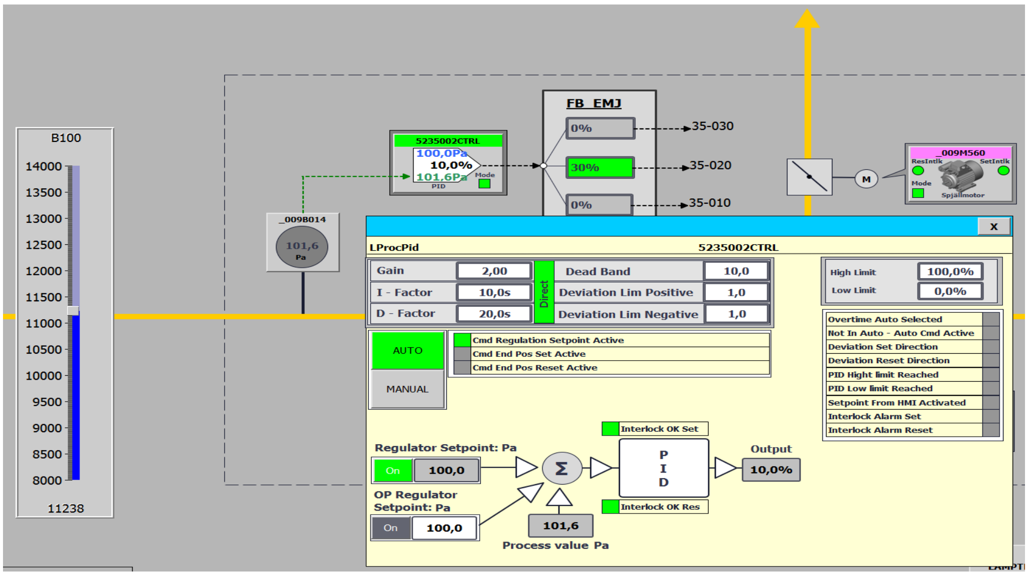
5.2.1.2 PID

The PID that controls the system will be set with a deadband of 10Pa, which means it will not change the output as long as the error is in that interval. This is mainly because when testing how the pressure corresponds to a specific speeds on the fans it was shown that the pressure was not stable, it usually jumped with a interval around 5-10Pa.

Due to energy savings and the fact that subsystems should not be start unless it is necessary the deviation gain will be set to 20 so that the chance to over-shoot reduces. The integration gain will be set to 10 so that the set point will not be reached too fast. The proportional gain will only be set to 2 to not get a too fast feedback since the system cannot reply as fast with the motors and the pressure. Since the inner-loop in the cascade controller can have a delay this could also induce the system to stabilize.

Figure 5.11 is illustrating the HMI panel of `subsystem 09` where the pressure difference is measured. If PLCsim is online the pressure vacuum can be changed with help from the blue barometer and the pressure, PID value and the output to the fans are shown. If the colour behind the percentage value is grey the system is off and if it is green it is on. Furthermore, since the PID values only could be tested using PLCsim and the corresponding from the real system has not been investigated further. It is necessary to tune the PID value swhen implementing the new code in the system.

$$\begin{cases} K_p = 2 \\ K_d = 20 \\ K_i = 10 \\ Deadband = 10Pa \end{cases} \tag{5.1}$$


LProcPid 5235002CTRL

Gain 2,00
I - Factor 10,0s
D - Factor 20,0s
Dead Band 10,0
Deviation Lim Positive 1,0
Deviation Lim Negative 1,0
High Limit 100,0%
Low Limit 0,0%

Regulator Setpoint: Pa On 100,0
OP Regulator Setpoint: Pa On 100,0
Process value Pa 101,6
Output 10,0%

Figure 5.11: The PID controller and the vacuum testing in the TIA environment

5.2.2 New system

The PLC worked as expected and the simulation of the pressure in PLCsim as well. But since the PLC can not be tested on the real system the accuracy and the testing are only simulated and can not be fully trusted. The PLC code and the HMI panel is working together as it should and the components is replying as it should. In Figure 5.12 - 5.15 the screens of the HMI panels are shown.

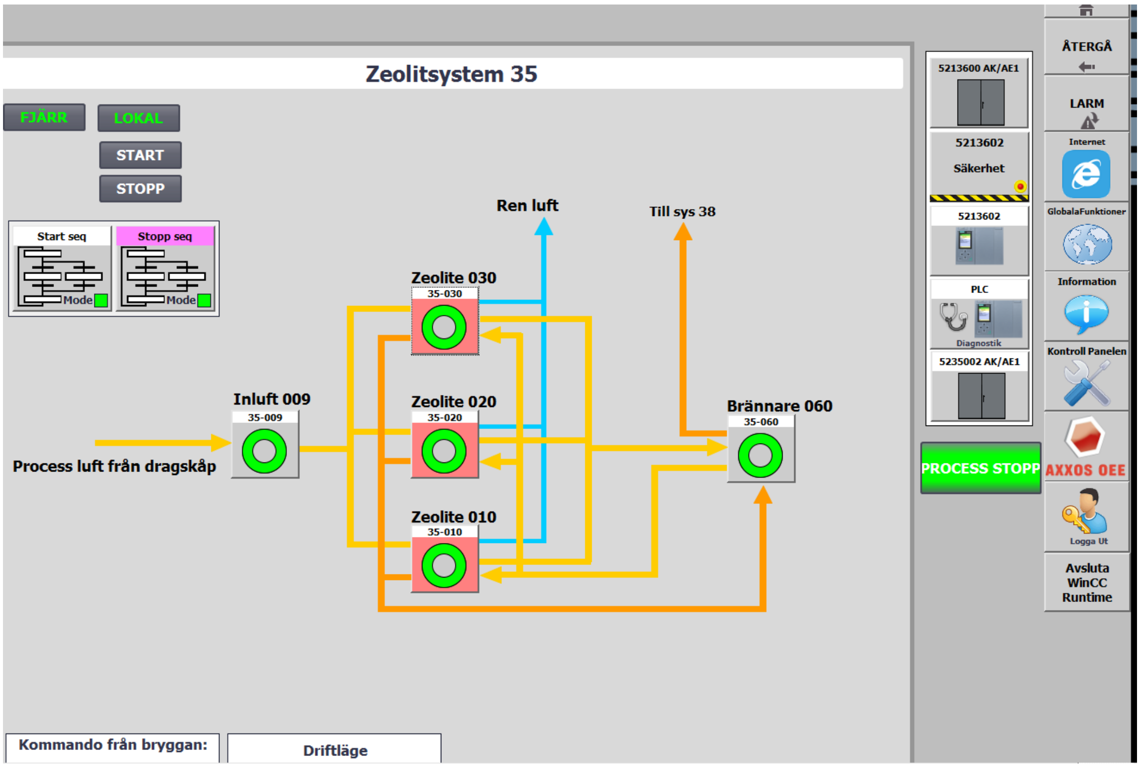


Figure 5.12: The home screen of the panel.

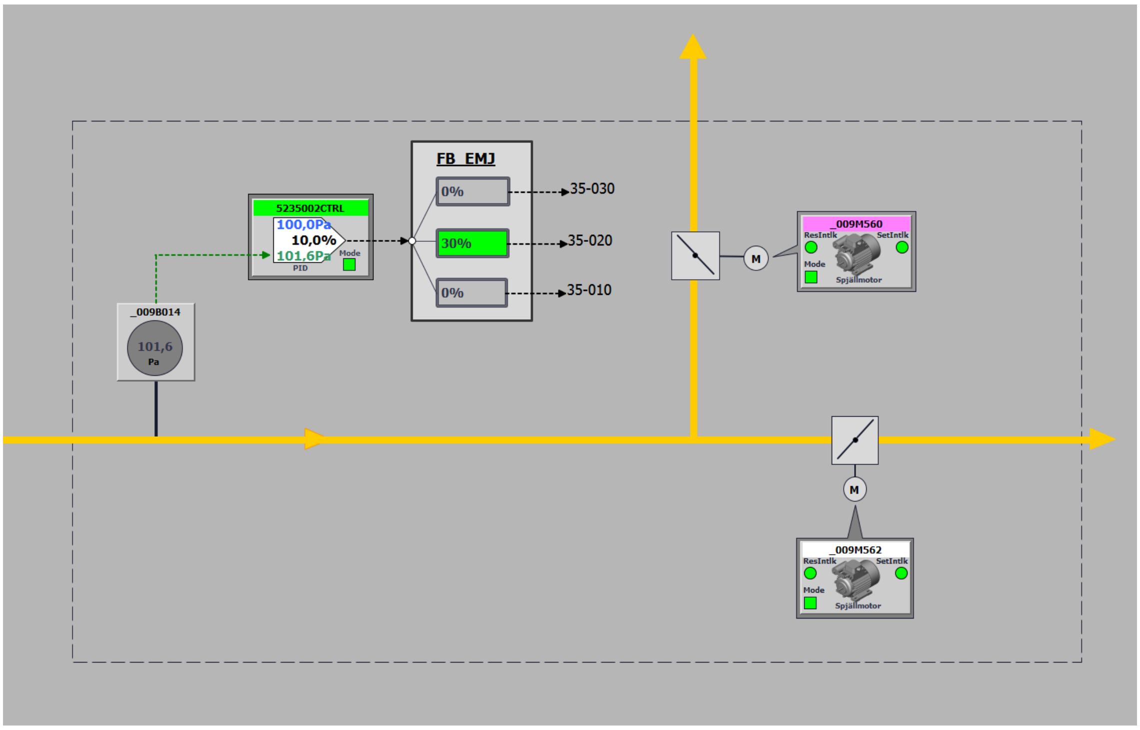


Figure 5.13: The screen of `subsystem 09`

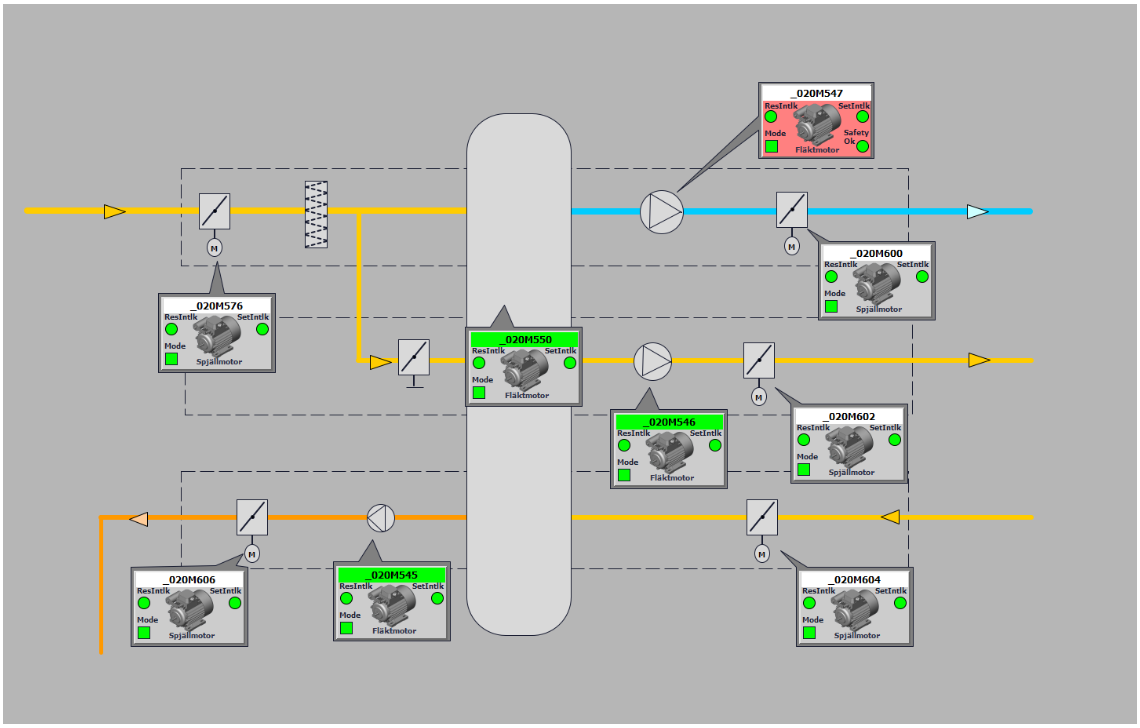


Figure 5.14: The screen of `subsystem 20`, the screen is the same for `subsystem 10` and `30` as well

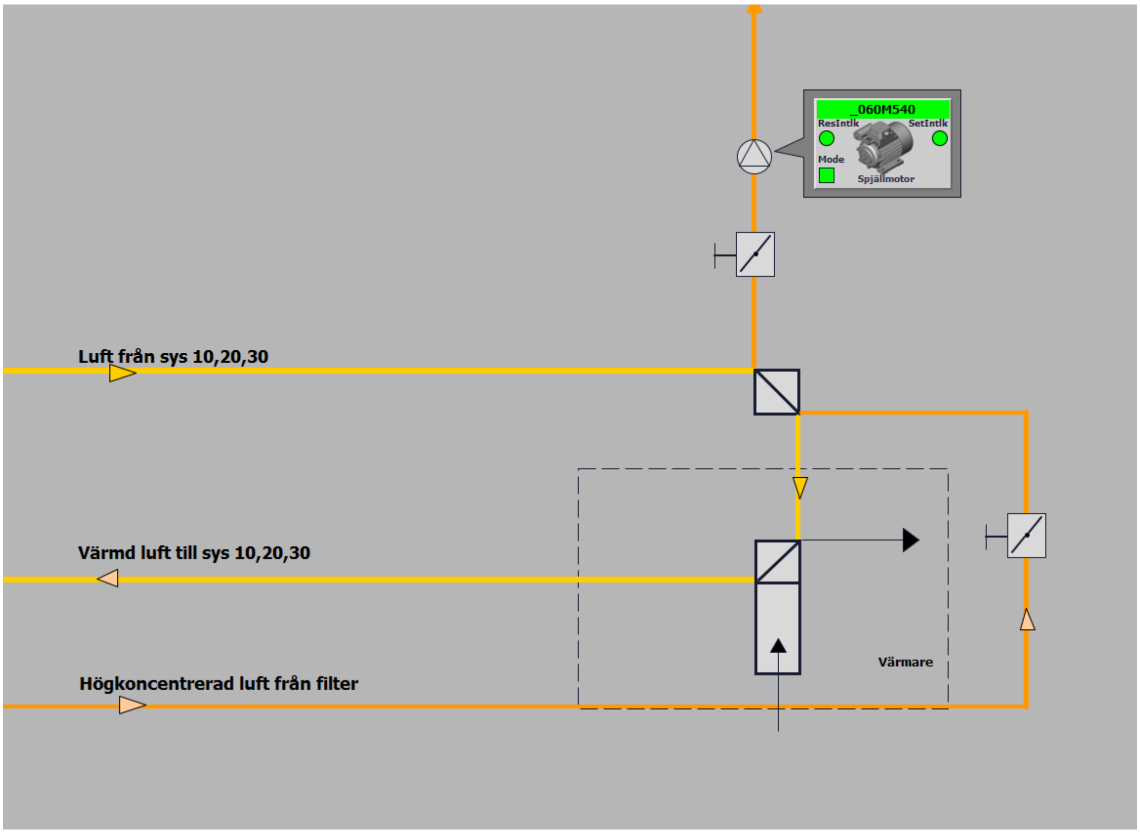


Figure 5.15: The screen of `subsystem 60`

5.2.3 Installation

For the installation and implementation of the new PLC, a decision with Volvo Cars was made to write the code for this system in the same environment as an other system that is placed next to this in the factory.
This will make the installation cheaper than changing the PLC for both systems and also the installation will most likely be easier since there already exist a PLC from Siemens on the other system. Since the installation will not be made during this assignment no further investigation has been made.

6

Conclusion

The new PLC, according to the optimization and the simulation in PLCsim, is more efficient and faster than the old system from SattCon. Although the old system did not really have any control or any optimization, it was just running at the speed the operator had manually written in. Therefore the new controller is smarter in that point of view, but it is also more efficient since the speed of the motors will not be larger than necessary. The implementation of the new PLC was not investigated more then, together with Volvo the decision was made to combine the code for this system together with the system beside this in the factory. The implementation would include uploading the new code to an existing PLC that already is installed in the system beside `35-002`. Moreover, the implementation/installation would also include connect all the cables correctly to the PLC and try the new code. However since the project does not include the actual implementation, the time to install and create new drawings can only be estimated.

6.1 Optimization

The optimization can be improved with even more accurate values of the consumption of energy on different speeds. The objective function and the constraints are well stated and the output of the optimization is meaningful. The result seems to be a good using the methods and applications as stated in the section 4.1.1. The optimization gave a more efficient and also a smarter solution for the process. As seen in the result, it is not necessary to run at maximum speed unless the output, u, from the PID is at maximum.

6.2 Controller

The PID controller worked well and the idea of the cascade controller worked better than predicted. However there can always be improvements, the settings of the pid can be more investigated and could make the process work both smoother and faster. However it will not make any difference until the new PLC together with the controller is installed in the system. It will then be possible to actually test the system and the controller to see if it actually works and how well it performs.

Bibliography

[1] "SattCon-5 Information". [Online]. Available: `https://library.e.abb.com/public/911eb0e0caae112fc12569e7005d9bb7/493067601_-_sv_SC05_Slimline__v4.0__Datablad.pdf` Accessed on Februari 20 2019

[2] "Zeolites - Max Peskov". [Online]. Available: `http://asdn.net/asdn/chemistry/zeolites.php#ref`, Accessed on May 10 2019

[3] "TIA-Portal, IEC Editor - Siemens". [Online]. Available: `https://w3.siemens.com/mcms/automation-software/en/tia-portal-software/step7-tia-portal/step7-professional/tabcards/pages/programming-languages.aspx`, Accessed on February 8 2019

[4] "Energy efficiency 101: What is energy efficiency?". [Online]. Available: `https://www.energysage.com/energy-efficiency/101/what-is-ee/`, Accessed on May 10 2019

[5] "Energy". [Online]. Available: `https://www.britannica.com/science/energy` Accessed on May 10 2019

[6] "What is Energy Efficiency?". [Online]. Available: `https://www.ovoenergy.com/guides/energy-guides/what-is-energy-efficiency.html` Accessed on May 10 2019

[7] "PID Controller Explained - DC I", 2018. [Online]. Available: `https://pidexplained.com/pid-controller-explained/` Accessed on May 8 2019

[8] "Fundamentals of cascade control - Vance Vandoren", 2014. [Online]. Available: `https://www.controleng.com/articles/fundamentals-of-cascade-control/` Accessed on May 8 2019

[9] "Mixed Integer Linear Programming, Combinatorial Problem Solving (CPS) - Javier Larrosa, Albert Oliveras, Enric Rodríguez-Carbonell". [Online]. Available: `https://www.cs.upc.edu/~erodri/webpage/cps/theory/lp/milp/slides.pdf` Accessed on May 4 2019

[10] "`Intlinprog`- MathWorks documentation". [Online]. Available: `https://se.mathworks.com/help/optim/ug/intlinprog.html` Accessed on March 3 2019

[11] "Fumehood picture". [Online]. Available: `https://www.indiamart.com/proddetail/laboratory-fume-hood-13978568133.html` Accessed on April 18 2019

[12] "Detailed information for: 3GAA182031-ADG", the main fans motors. [Online]. Available: `https://new.abb.com/products/3GAA182031-ADG` Accessed on February 28 2019

[13] "Detailed information for: 3GAA092315-ASE", the small fans motors. [Online]. Available: `https://new.abb.com/products/3GAA092315-ASE/m3aa-90ld-4` Accessed on February 28 2019

[14] "Detailed information for: 3GAA092315-ASE" Fan 540 motor. [Online]. Available: `https://new.abb.com/products/3GAA092315-ASE/m3aa-90ld-4` Accessed on February 28 2019

[15] "SIEMENS Simatic S7-1500 Handbook - Siemens". [Online]. Available: `https://www.automation.siemens.com/salesmaterial-as/interactive-manuals/getting-started_simatic-s7-1500/documents/EN/software_complete_en.pdf`, Accessed on February 8 2019

[16] "Dynamic vacuum pressure tracking control with high-speed on-off valves", Gang Yang, Kai Chen, Linglong Du, Jingmin Du and Baoren Li, 2018. [Online]. Available: `https://journals-sagepub-com.proxy.lib.chalmers.se/doi/pdf/10.1177/0959651818780866` Accessed on January 30 2019

[17] "Energy efficiency optimization of air supply system in a water bottle manufacturing system", Vukica Jovanovic, Branislav Stevanov, Dragan Seslija, Slobodan Dudic, Zdravko Tesic, March 2014. [Online]. Available: `https://ac-els-cdn-com.proxy.lib.chalmers.se/S095965261400242X/1-s2.0-S095965261400242X-main.pdf?_tid=be1b5f82-921e-4031-85d6-a2a28ec68c51&acdnat=1547662543_9f20892cd8586ca03bba39129f1f581f` Accessed on January 30 2019

[18] "Industrial Automation, Lecture Notes", Martin Fabian, 2006

A

Appendix

```
clear all;
close all;
clc;



%% Decision Variables

f = [21, 22.8, 21]*10^3; % Maximum fan capacity in m^3/h
sym a; % Flow into system that is calculated from P


e = [8.5;7.45;7;4.5;2.3;1.2];     % Energy of running each fan on
different speeds
E = [7.55; 7.55; 7.55];    % energy of running entire subsystem

x = optimvar('x',3,'LowerBound',0,'UpperBound',1,'Type','integer');
                            % decision variable x
t = optimvar('t',[6,3],'LowerBound',0,'UpperBound',1,'Type','integer');
% decision variable y




prob = optimproblem('Objective',e(1)*t(1,1)+e(2)*t(2,1)...
        +e(3)*t(3,1)+e(4)*t(4,1)+e(5)*t(5,1)+e(6)*t(6,1)
        +E(1)*x(1)+e(1)*t(1,2)+e(2)*t(2,2)+e(3)*t(3,2)...
        +e(4)*t(4,2)+e(5)*t(5,2)+e(6)*t(6,2)+E(2)*x(2)...
        +e(1)*t(1,3)+e(2)*t(2,3)+e(3)*t(3,3)+e(4)*t(4,3)...
        +e(5)*t(5,3)+e(6)*t(6,3)+E(3)*x(3)+3,...
            'ObjectiveSense','min');

prob.Constraints.c1 =  x(1) - sum(t(:,1)) == 0;
prob.Constraints.c2 = x(2) - sum(t(:,2)) == 0;
prob.Constraints.c3 = x(3) - sum(t(:,3)) == 0;

prob.Constraints.c4 = sum(t(:,1)) <= 1;
prob.Constraints.c5 = sum(t(:,2)) <= 1;
prob.Constraints.c6 = sum(t(:,3)) <= 1;

prob.Constraints.c7 = x(2) >= x(3) ;
prob.Constraints.c8 = x(3) >= x(1) ;
```

```
a=0;
b = [15000;17000;21000; 23000; 26000; 27500; 29000; 32000;...
    35000; 37000; 40000; 42000; 45000; 47000; 50000; 55000;...
    58000; 60000; 64700];

for i = 1:19


prob.Constraints.c22 = f(1)*(t(1,1)+t(2,1)*0.9+t(3,1)*0.8+t(4,1)*0.7+...
                       t(5,1)*0.6+t(6,1)*0.5)+f(2)*(t(1,2)+t(2,2)*0.9+...
                       t(3,2)*0.8+t(4,2)*0.7+t(5,2)*0.6+t(6,2)*0.5)+...
                       f(3)*(t(1,3)+t(2,3)*0.9+t(3,3)*0.8+t(4,3)*0.7+...
                       t(5,3)*0.6+t(6,3)*0.5) >= b(i);

problem = prob2struct(prob);
[xt,fval,exitflag,output] = intlinprog(problem);
variable(i,:) = xt;  %% [t1;t2;t3;t4;t5;t6;x];
fmin(i) = fval;
out(i) = output;

a = a+1
end
fmin = fmin';

t1 = variable(:,1:6);
t2 = variable(:,7:12);
t3 = variable(:,13:18);
x = variable(:,19:21);
%% plot
figure()
for i =1:6

    plot(t1(:,i),b,'*')
    hold on
end
plot(x(:,1),b,'--')
title('Subsystem 1 fan 553 level att different flows')
legend('100%','90%','80%','70%','60%','50%','on/off')
xlabel('Which speed the fan is running at')
ylabel('Flow m^3/h')

figure()
for i = 1:6
  plot(t2(:,i),b,'*')
    hold on
end
plot(x(:,2),b,'--')
title('Subsystem 2 fan 547 level att different flows')
legend('100%','90%','80%','70%','60%','50%','on/off')
xlabel('Which speed the fan is running at')
ylabel('Flow m^3/h')

figure()
for i=1:6
    plot(t3(:,i),b,'*')
    hold on
```

```
end
plot(x(:,3),b,'--')
legend('100%','90%','80%','70%','60%','50%','on/off')
title('Subsystem 3 fan 543 level att different flows')
xlabel('Which speed the fan is running at')
ylabel('Flow m^3/h')

figure()

    plot(x(:,1),b,'.','MarkerSize',20)
    hold on

    plot(x(:,2),b,'.','MarkerSize',10)
    hold on

    plot(x(:,3),b,'.','MarkerSize',5)
    hold on
xlabel('If the fan is on or off ')
ylabel('Flow speed in m^3/h')
title('Which fans that runs in different flows')
legend('subsystem 1','subsystem 2', 'subsystem 3')
figure()
  plot(fmin,b,'--')
  legend('Optimumvalue, minimum energy cost')
  xlabel('The minimum energy to meet the flow')
  ylabel('Flow m^3/h')



  %%


  for i = 1:19
 % if x(i,1) == 1 && x(i,2) == 0
      if t1(i,1) >= 0.9
          speed1(i) = 100;
      elseif t1(i,2)>= 0.9
          speed1(i) = 90;
       elseif t1(i,3)>= 0.9
          speed1(i) = 80;
           elseif t1(i,4) >= 0.9
          speed1(i) = 70;
           elseif t1(i,5) >= 0.9
          speed1(i) = 60;
      elseif t1(i,6) >= 0.9
          speed1(i)=50;
      else
          speed1(i) =0;
      end
            if t2(i,1) >= 0.9
          speed2(i) = 100;
      elseif t2(i,2) >= 0.9
          speed2(i) = 90;
       elseif t2(i,3) >= 0.9
          speed2(i) = 80;
           elseif t2(i,4) >= 0.9
```

```
          speed2(i) = 70;
           elseif t2(i,5)>= 0.9
          speed2(i) = 60;
            elseif t2(i,6) >= 0.9
          speed2(i)=50;
            else
                speed2(i)=0;
            end
            if t3(i,1) >= 0.9
          speed3(i) = 100;
      elseif t3(i,2) >= 0.9
          speed3(i) = 90;
       elseif t3(i,3) >= 0.9
          speed3(i) = 80;
           elseif t3(i,4) >= 0.9
          speed3(i) = 70;
           elseif t3(i,5) >= 0.9
          speed3(i) = 60;
            elseif t3(i,6) >= 0.9
          speed3(i)=50;
            else
          speed3(i) = 0;
            end
  end

  for i = 1:19
      sum(i) = speed1(i)+speed2(i)+speed3(i);
      xpid(i) = sum(i)/3;
      speed(:,i) = [speed1(i),speed2(i),speed3(i)];
  end

  S = [b';fmin';speed];


  figure()


   plot([0,b'],[0, speed1],'.-')
  hold on
    plot([0,b'],[0, speed2],'.-')
  hold on
    plot([0,b'],[0, speed3],'.-')
title('The speeds of the main fans depending on the inflow a')
xlabel('flow in m^3/h')
ylabel('speed in percentage')
legend('fan 553','fan 547','fan553')

figure
 plot([0,fmin'],[0, speed1],'.-')
  hold on
    plot([0,fmin'],[0, speed2],'.-')
  hold on
    plot([0,fmin'],[0, speed3],'.-')
title('The speeds of the main fans and the energy cost')
xlabel('energy consumption kW')
ylabel('speed in percentage')
```

```
legend('fan 553','fan 547','fan553')
```