TALLINN UNIVERSITY OF TECHNOLOGY
SCHOOL OF ENGINEERING
Department of Electrical Power Engineering and Mechatronics

PARAMETER ESTIMATION IN NONLINEAR DYNAMIC SYSTEMS USING THE OPTIMIZATION METHODS BASED ON THE EXPERIMENTAL DATA

PARAMEETRI HINDAMINE MITTELINEAARSES DÜNAAMILISES SÜSTEEMIS, KASUTADES KATSEANDMETEL PÕHINEVAID OPTIMEERIMISMEETODEID

MASTER THESIS

Student:	Hossein Alimohammadi
Student Code:	184504MAHM
Superviser:	Kristina Vassiljeva, Associate professor
Co-Superviser	Mart Tamre, Professor

Tallinn 2020

AUTHOR’S DECLARATION

Hereby I declare, that I have written this thesis independently.
No academic degree has been applied for based on this material. All works, major viewpoints and data of the other authors used in this thesis have been referenced.

“.......” .... 20…..

Author: ....
/signature /

Thesis is in accordance with terms and requirements

“.......” .... 20….

Supervisor: ….........................
/signature/

Accepted for defence

“.......”....................20… .

Chairman of theses defence commission: ....
/name and signature/

Non-exclusive Licence for Publication and Reproduction of GraduationThesis[1]

I, Hossein Alimohammadi (date of birth: 16Dec1978) hereby

1. grant Tallinn University of Technology (TalTech) a non-exclusive license for my thesis "Parameter estimation in nonlinear dynamic systems using the optimization methods based on the experimental data"

supervised by Dr. Kristina Vassiljeva

1.1 reproduced for the purposes of preservation and electronic publication, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright;

1.2 published via the web of TalTech, incl. to be entered in the digital collection of TalTech library until expiry of the term of copyright.

1.3 I am aware that the author also retains the rights specified in clause 1 of this license.

2. I confirm that granting the non-exclusive license does not infringe third persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

---

[1] *Non-exclusive Licence for Publication and Reproduction of Graduation Thesis is not valid during the validity period of restriction on access, except the university`s right to reproduce the thesis only for preservation purposes.*

______________ (*signature*)

______________ (*date*)

Department of Electrical Power Engineering and Mechatronics: Mechatronics and Autonomous Systems Centre

THESIS TASK

Student: …Hossein Alimohammadi, 184504MAHM

Study programme, MAHM02/18 - Mechatronics

main speciality:

Supervisor(s): Kristina Vassiljeva, Associate professor, 620 2116

Consultants: …Mart Tamre, Professor

……………………………………………………………… (company, phone, e-mail)

Thesis topic:

(in English) ..Parameter estimation in nonlinear dynamic systems using the optimization methods based on the experimental data

(in Estonian) … Parameetri hindamine mittelineaarsetes dünaamilistes, kasutades katseandmetel põhinevaid optimeerimismeetodeid ….

Thesis main objectives:

1. Unknown parameter estimation in dynamic physic-based modeling
2. Development of new software package based on intelligent search algorithm
3. Involving in industrial application for estimating the HVAC filter's parameter and prediction of clogging in filter

Thesis tasks and time schedule:

No	Task description	Deadline
1.	**Parameter estimation and its methods**	**30.02.2020**
2.	**Implimentation of intelligent algorithm for estimation**	**15.03.2020**
3.	**Development the software package**	**30.03.2020**
	Perform on industrial project	**10.05.2020**

Language: English **Deadline for submission of thesis:** 01 " June 2020

Student: Hossein Alimohammadi .... "......."…………......2020

/signature/

Supervisor: Kristina Vassiljeva ……… "......."............2020

/signature/

Consultant: Mart Tamre …............... "......."...........2020

/signature/

Head of study programme: Mart Tamre "......."..............2020

/signature/

Terms of thesis closed defence and/or restricted access conditions to be formulated on the reverse side

CONTENTS

1 INTRODUCTION ... 1
  1.1 Problem statement ... 1
  1.2 Methodology ... 1
  1.3 Contents ... 2
2 LITERATURE OVERVIEW ... 3
  2.1 Intelligent Optimization ... 3
  2.2 Advantage of Intelligent methods ... 3
  2.3 Particle Swarm Optimization ... 4
  2.4 Survey of Hybridization Approaches ... 4
  2.5 Case study object ... 8
3 OPTIMIZATION TECHNIQUES ... 10
  3.1 GA theory ... 10
  3.2 PSO theory ... 12
  3.3 ABC theory ... 15
  3.4 Introducing Dynamic Systems Identification Software Package ... 16
  3.5 Solving a simple example ... 18
4 PARAMETER ESTIMATION AND CLOGGING PREDICTION ... 22
  4.1 System description ... 22
  4.2 Optimization method for parameter estimation of nonlinear systems ... 23
  4.3 Modeling and parameter estimation of the return filter in HVAC system ... 24
  4.4 Filter clogging estimation and prediction in HVAC systems ... 29
  4.5 Other units ... 31
  4.6 Conclusion ... 33
5 PHYSICS-BASED DYNAMIC MODELING OF HVAC SYSTEM ... 35
  5.1 HVAC Modeling Techniques ... 35
  5.2 White box model (physics-based) ... 36
  5.3 Simulation ... 39
  5.4 Recommendations for future work ... 45
Summary ... 47
Kokkuvõte ... 48
References ... 49
Appendix ... 53

FIGURE CONTENTS

Figure 1. Classification of optimization techniques [22] 10
Figure 2. Pseudo-code of the genetic algorithm 11
Figure 3. Working principle of binary genetic algorithm and its different steps..........12
Figure 4. Search space to find the highest point 13
Figure 5. Scattering of particles to find the highest point 14
Figure 6. Update Particle velocity and location in PSO Algorithm.............................14
Figure 7. Graphic implementation of the model in SIMULINK.................................19
Figure 8. Convergence plot of GA(left) and PSO(right) algorithm 20
Figure 9. Comparison of actual and model proposed by Genetic Algorithm 20
Figure 10. Diagram of filter in AHU of HVAC system...26
Figure 11. Supply filter pressure drop over time, unit 333 27
Figure 12. Supply filter pressure drop tracking 27
Figure 13. Return filter pressure drop, unit 1745 28
Figure 14. Cost function history, ξ estimation 28
Figure 15. Polynomial model result 30
Figure 16. Pressure drop (measured and simulated), inputs (Temperature and flow rate) 30
Figure 17. Tracking of final result 31
Figure 18. Cost function history, clogging estimation..31
Figure 19. Clogging prediction 33
Figure 20. Zone model (Simscape model) 36
Figure 21. Heat transfer in Zone model (Simscape model) 37
Figure 22. Damper model (Simscape model)...38
Figure 23. overview of whole system (Simulink and Simscape model) 39
Figure 24. AHU model (Simulink and Simscape model)..39
Figure 25. Heater model (simulink model) 39
Figure 26. Air temperature after heater 40
Figure 27. Air fan model in Simscape 40
Figure 28. Air fan control model (Simulink model) 40
Figure 29. Air fan validation model in Simulink 41
Figure 30. Air fan sub-model in Simulink 41
Figure 31. Fan orifice pressure drop, open loop simulation, input: ASFAV, output: ASFPDE 41
Figure 32. Close loop simulation, input: Setpoint ASFPDESP, output: ASFPDE 42
Figure 33. Heat Recovery Ventilator simulink model (HRV) 42
Figure 34. Simulation output, air zone supply air temperature...............................42
Figure 35. Simulation output, energy recovery ventilator temperature 43

Figure 36. Simulation output, duct temperature after heater coil 43
Figure 37. Simulation output, supply air duct flow rate 44
Figure 38. Simulation output, fan control using pressure drop of fan orifice.............44

Abbreviations

ABC	Artificial bee colony
ACO	Ant Colony Optimization
AHCSWst	Heat exchanger set point temperature
AHCVO	Air heater valve opening
AHU	Air handling unit
AIS	Artificial Immune Systems
APSO	Accelerated PSO
ASA	Adaptive Simulated Annealing
ASFPDE	Air supply pressure drop
ASFPDESP	Air supply pressure drop set point
BA	Bat Algorithm
BAS	Building automation system
BIM	Building information model
CEED	Combined Economic and Emission Dispatch
CO_2	Carbon dioxide
COW	Cluster of Workstation
CS	Cuckoo Search
DAQ	Data acquisition
DDM	Domain Decomposition Method
DE	Differential Evolution
DF	Dimension Factor
DM	Differential Mutation
DNS	Dynamic sub-swarm number strategy
EA	Evolutionary algorithm
ERV	Energy recovery ventilator
FA	Firefly Algorithm
FCTPs	Fixed Charge Transportation Problems
FEM	Finite element method
FIR	Finite Impulse Response
GA	Genetic algorithm
GMPP	Global Maximum Power Point
GSO	Glow Worm Swarm Optimization
HACPSO	Hybrid Accelerated Cuckoo Particle Swarm Optimization
HAP	Hybrid approach
HCLPSO	Heterogeneous comprehensive learning PSO
HEPGO	Hybrid PSO and Glow Worm Swarm algorithm
HOA	Hybrid Optimization Algorithm

HRV	Heat recovery ventilator
HVAC	Heating, ventilation and air conditioning
IABAP	Integrated algorithm based on ABC and PSO
MEPSO	Multi-Strategy Ensemble PSO
MIMO	Multi input multi output
MISO	Multi input single output
MPI	Message Passing Interface
MSPSO	Multi-swarm particle swarm optimization
NFL	The No Free Lunch Theorem
OOP	Object-oriented Programming
PDS	Purposeful detecting strategy
PHEV	Plug-In Hybrid Electric Vehicle
PPSO	Parallel Particle Swarm Optimization
PSACO	Particle Swarm Ant Colony Optimization
PSO	Particle Swarm Optimization
RMSE	Root mean squared error
SA	Simulated Annealing
SDEA	Swarm and differential evolution algorithm
SE	Standard error
SP	Set point
SRS	Sub-swarm regrouping strategy
SS	State-space
SSE	Sum square error
TF	Transfer function

Nomenclature

Alphabets

A	Area [m^2]
C	Thermal capacitance [J/K]
C_{pa}	Specific heat of air at constant pressure [J/KgK]
C_{pw}	Specific heat of water at constant pressure [J/KgK]
D	Diameter
h	Convection coefficient [W/m^2K]
m	Mass flow rate [kg/s]
P	Pressure [Pa]
Q	Heat transfer rate [W/m^2]
T	Temperature [K]°
T_m	Mixer temperature [K]°

U	Conduction heat transfer coefficient [W/mK]
V	Volume [m^3]
$y(t)$	Output
y	Measured value
$\hat{y}$	Predicted value

Greek Letters

ΔP	Change in pressure [Pa]
ΔT	Change in temperature [C]°
Φ	Angular position [rad]
ε	Effectiveness of heat exchanger
ρ	Density [kg/m^3]

Subscripts

a	air
ai	Inlet air
amb	ambient
ao	Outlet air
cc	Cooling coil
dp	Damper
ea	Exhaust air
eai	Exhaust air at inlet
eao	Exhaust air at outlet
fai	Fresh air at inlet
fao	Fresh air at outlet
int	integration
l	Leakage
sa	Supply air
sd	Supply duct
sdi	Inside the supply duct
w	Water
wa	Wall
wd	Window
wf	Water and floor
wi	water input
wm	Water and metal
wo	Water output
z	Zone

1 INTRODUCTION

1.1 Problem statement

During the design process in research laboratories as well as in industries and also for control purposes, one of the problems encountered after mechanical modeling is the measurement of physical parameters and coefficients. This can sometimes be very time consuming and costly. The purpose of this thesis is to estimate these parameters without a need to build measurement equipment but to use numerical methods. Obviously, one of the simplest ways to estimate the parameters is to visualize the output of the system and correct the parameters by trial and error. Surely this method is not accurate and cannot be used in meticulous design or for MIMO systems. At the same time, it is not usable during the design process of the controller or even the estimation of the control coefficients. Therefore, obtained experimental results should be compared and fit to the first principal model of the process. Obviously, issues such as divergence and local minimum need to be resolved. In addition, some methods for data fitting in nonlinear dynamic systems can suffer from slow and/or local convergence, which should be solved in this thesis. It is assumed that only partial state measurement is available from experiments, and that the parameters appear nonlinearly in the system equations. Simultaneously estimation of two or more parameters are considered. Another goal of this thesis is to provide general software package that can run the Simulink (linear or nonlinear) model separately. It means it is possible easily to evaluate different types of optimization methods and combination of them.

The identification techniques have been used for many years

- To provide a complete set of code, with a focus on the physical parameters of the system.
- General software package
- Comparing the identification methods and using the appropriate one
- Combination of methods in order to find the best one.

1.2 Methodology

As mentioned before, the mechanical kinetic models usually contain unknown parameters, which need to be measured by test facilities. Those experiments sometimes can be expensive and time consuming. Moreover, in some cases such as hysteresis, dead-zone parameters, magnetic parameters and friction coefficients in multi-coupled systems, measuring the parameters is almost impossible. However, the better alternative can be estimation the parameters by optimizing the fit of the model to experimental data. This task can be computationally challenging due to the presence of local optima and disposition. While a variety of optimization methods have been suggested to overcome these issues, it is not obvious how to choose the best one, since many factors can influence their performance. In this thesis, we try to develop a robust method and appropriate code to estimate the parameters for certain types of dynamic systems. To do so, it is assumed that the relevant expert has a thorough understanding of the physics of identified mechatronics systems and is mastered in the mathematical equations of system. The reason is that, in principle, all of the estimated parameters have meaningful physical dimensions and are within reasonable range.

1.3 Contents

In order to complete work on time, the following procedures may progress: Chapter 2 presents an overview of intelligent optimization and a survey of historical and recent developments with PSO hybridization Perspectives. In chapter 3, the dynamic systems identification optimization software package which has been done based on intelligent methods is introduced and a simple example is solved. In chapter 4 estimation of an unknown parameter of filter in HVAC and filter clogging prediction is done using mentioned package. Chapter 5 describes the physic-based modeling and simulation of AHU and HVAC components such as heater, heat recovery ventilator and zone model.

2 LITERATURE OVERVIEW

2.1 Intelligent Optimization

In general, optimal selection and design in projects will produce the best possible answer to a particular situation. The suitable production in various technical and engineering fields depends on the precise design, size, material, cost, machine operating cost, weight and power consumption. For example, there are different shapes and materials to make magnetic actuators, but which one will have the better results? Is aluminum or a special alloy better, or better than a composite material? On the other hand, what is its shape, size and weight depending on the material used? We see that there are numerous design and decision methods in each case, but what is the best way and how to find it? Thus, given the issues raised, we realize the importance of optimization. So: "Our goal is to look for the best answer in the space of possible answers". Since the outcome depends on the solution method, it is important to choose an appropriate method to solve the problem under consideration. For this reason, several optimization methods have been proposed since the 1940s. These include methods such as counting method, classical methods (linear and nonlinear constrained and unconstrained eg. Second-order cone programming, Lagrangian, Simplex, Reduced-Gradient, Multivariate Optimization) and intelligent methods (genetic algorithm, neural networks, species colony searching algorithms, fuzzy and tabu search).

Classical methods are based on mathematical principles which in practice have problems in terms of runtime and large amounts of computing and memory, but are highly accurate.

2.2 Advantage of Intelligent methods

However, solving complex problems using ways such as linear programming and Jacobi slope methods is extremely complex and sometimes impossible. For this reason, nowadays, the motivation of a powerful method in highly sophisticated optimization has led researchers to use intelligent and evolutionary optimization techniques. The general features of these methods are as follows:

1. The intelligent algorithm does not require the derivatives of the objective function and only uses the evaluation of the decision variables in the objective function. Therefore, it does not impose a constraint on the objective function in terms of derivability.

2. The intelligent algorithm performs a search in parallel with a set of decision variables, each of which (equivalent to one chromosome) can be a possible answer to the problem. However, other methods continue to search with only one set of decision variables that are only one possible answer to the problem.
3. The algorithms use random and statistical transmission. These approaches use statistical rules to guide the search. The search process is not based on pure accident. Rather, it uses random search as a tool to achieve a better search domain.
4. Algorithms work with coded decision variables and hence it is a high-coherence optimization program. Based on above mentioned, the optimization problems in different contexts can be easily defined in an intelligent algorithm program.
5. The intelligent algorithms are based on natural evolution. These algorithms do not use for example gradient information to search for the optimal state and since they search for the optimal state in parallel with a population of decision variables, they are therefore more capable of finding the general optimal state. Therefore, they are suitable algorithm for complex optimization.

2.3 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is a global optimization model that has improved in the last two decades due to its ease of use in complex problems, which cannot be achieved using traditional definitive algorithms. Optimization of particles is based on herd behavior and social cooperation of bird and fish and is strongly derived from the evolutionary behavior of these organisms. This thesis provides a comprehensive review of the PSO algorithm with special emphasis on its most basic development as well as some of the latest implementations. Types of PSOs include: Discrete and binary PSO, Hybrid PSO, Gaussian PSO.

2.4 Survey of Hybridization Approaches

The literature on PSO combination algorithms is quite powerful and growing. Sengupta et al. In [1] collected a survey on PSO hybridization. In this section, some of the most prominent works as well as some recent approaches are stated.

Hybridization of PSO using Genetic Algorithms (GA)

Common approaches to GA and PSO hybridization uses the two consecutive or parallel methods or using GA operators such as mutation, and reproduction within the PSO framework. Authors in [2] have used one algorithm to reach the stop criterion to process the final solution in another algorithm for accurate adjustment. How is the selected stop criterion different? They use a method of moving between algorithms when an algorithm cannot repeat the previous results more than a number of repetitions.Over the years, some authors have suggested a fuzzy approach to PSO-GA. Ghamisi and Benediktsson [1] introduced a selection method using PSO and GA hybridization. This method was evaluated in the superstructure data set as well as for road application. This method can automatically select the most instructive features at an acceptable processing time and does not require users to pre-determine the number of desired features. Nick et al. performs GA-PSO and a combination of other hybridization methods to optimize the asphalt coating inspection [2].Using PSO-GA, Lee et al. in [5] Created a mathematical model of the object and determined its highest potential parameter. The results show an increase in speed and superior optimization compared to GA.

A brief list of PSO and GA algorithm is given in Table 1.

Table 1. Hybridized GA-PSO algorithms.

Author/s:	Year	Algorithm	Application
Robinson et al. [3]	2002	PSO-GA	Engineering design
Ghamisi and Benedictsson [1]	2015	GA-PSO	Feature
Nik, Nejad and Zakeri [2]	2016	GA-PSO	Inspection Unit
Li et al. [4]	2018	PSO-GA	Optimization of a field layout

PSO using Differential Evolution (DE)

Difference Evolution (DE) is an effective metaheuristic method for solving global optimization problems. There are several methods of combining DE with PSO in the literatures, some of which are described below. In 2001, Handellas [5] introduced a combination of particle evolution and differentiation (SDEA) algorithms and tested it on a series of experimental problems. The SDEA algorithm acts like a particle swarm, except that DE is implemented as an alternative to move particles from areas with worse performance to better locations.

In [7] Seyed Mahmoudian et al. identified the maximum power point in partial shadow conditions. The simulation and experimental results have confirmed in a variety of partial conditions, and therefore its reliability in global optimal tracking. Boonserm and Sitjongsataporn [8] regulated DE, PSO, and ABC with their regulatory weights, which are determined using a sigmoid membership function. DE has helped eliminate the possibility of premature convergence, and PSO has accelerated the optimization process.

A brief list of PSO and DE algorithm is given in Table 2.

Table 2. Hybridized DE-PSO algorithms.

Author/s:	Year	Algorithm	Application
Hendtlass [5]	2001	SDEA	Global Optimization
Seyed mahmoudian et al. [6]	2015	DEPSO	Power Generation
Boonserm and Sitjongsataporn [7]	2017	Scout DEPSO	Numerical Optimization

PSO using Simulated Annealing (SA)

Zhao et al. [9] proposed a model for activity-based multi objective network and used a new PSO and SA-based exploration to solve the multi objective problem. Sudibyo et al. [10] used PSO-SA to control the temperature of the distillation reaction in a nonlinear model prediction control (NMPC) and pointed to the algorithm's efficiency to find the optimal result of hybridization. Javidrad and Nazari [11] have recently provided a PSO-SA that SA find the best particle if the PSO does not show progress in the performance, which may occur several times during the repetition cycles. This sharing process is sustainable as long as the convergence criteria are not met. Li et al. [12] introduced an efficient energy management plan to increase the fuel efficiency of a Plug-In hybrid electric vehicle.

A brief list of PSO and SA algorithm is given in Table 3.

Table 3. Hybridized SA-PSO algorithms.

Author/s:	Year	Algorithm	Application
Zhao et al. [8]	2005	SDEA	Virtual Enterprise
Sudibyo et al. [9]	2015	SA-PSO	Predictive Control
Javidrad and Nazari [10]	2017	SA-PSO	Global Optimization
Li et al. [11]	2017	SA-PSO	Hybrid Electric Vehicle

PSO using Ant Colony Optimization (ACO)

The Ant Clony Optimization (ACO) proposed by Marco Dorigo [13] took the organized communication resulting using the idea of ant colonies. Shelokar et al. [14] presented the PSACO to have a rapid exploration in the search domain. The first part of the algorithm works to generate initial solutions on the PSO, while the particle position is updated by the ACO in the next part. This strategy has proven to be almost optimal for very non-convex problems.

Mandloi et al. [12] used a hybrid algorithm with a possible novel search method by integrating the axial search approach provided by ants in ACO and the speed-based search-oriented mechanism adopted by the particles in the PSO. The metric of probability used in this algorithm includes weight exploration values obtained from the distance and velocity. Junliang et al. [16] proposed a hybrid optimization algorithm (HOA) that exploits the benefits of global search and rapid convergence in PSO, allowing early convergence to capture ACO. The algorithm converges to the desired solution by setting its initial parameters by PSO.

A brief list of PSO and ACO algorithm is given in Table 4.

Table 4. Hybridized ACO-PSO algorithms.

Author/s:	Year	Algorithm	Application
Marco Dorigo [13]	1991	ACO	Communication
Shelokar et al. [14]	2007	PSACO	continuous optimization
Mandloi and Bhatia [12]	2016	PSO, ACO	Large-MIMO detection
Junliang et al. [15]	2017	HOA	Traveling salesman problem

PSO using Cuckoo Search (CS)

The CS provided by Xin-She Yang and Suash Deb was developed based on the breeding behavior of cuckoos related to the climatic nature of birds and flies. After that, Lotfi [17] introduced the Hybrid PSOCS algorithm, which uses PSO to increase the ability of cuckoos to communicate with each other to reduce the likelihood of their birds being identified and released by host birds. During migration, each cuckoo registers its best personal characters, thus creating the best in the global.

A brief list of PSO and CS algorithm is given in Table 5.

Table 5. Hybridized CS-PSO algorithms.

Author/s:	Year	Algorithm	Application
Ghodrati and Lotfi [16]	2012	Hybrid CS/PSO	Global optimization
Guo et al. [17]	2016	PSOCS	Global optimization
Chi et al. [18]	2019	CSPSO	Optimization

PSO using Artificial Bee Colony (ABC)

The ABC was introduced in 2009 by Karaboga and Basturk and distributed activities in accordance with the colonies of bees. Gao et al. [19] presented an algorithm based on PSO and ABC with the parallel implementation of ABC and PSO and the exchange of information between swarm particles and bee colonies.

Zhou and Young [20] introduced PSO-DE-PABC based on PSO, ABC and DE. Divergence increases with the creation of new locations around random particles through PSO-DE-PABC, while PSO-DE-GABC increases the search rate by creating divergence optimized by differential vectors and Dimension Factor (DF). Sedighizadeh and Mazaheripour [21] used the PSO-ABC algorithm with the best personal scores for each entity and mostly refined it through the PSO and ABC phases.

A brief list of PSO and ABC algorithm is given in Table 6.

Table 6. Hybridized PSO-ABC algorithms.

Author/s:	Year	Algorithm	Application
Shi, et al. [19]	2010	IABAP	Global optimization
Zhou and Yang [20]	2015	PSO-DE-PABC and PSO-DE-GABC	Optimization
Sedighizadeh and Mazaheripour [21]	2017	PSO-ABC	Multi objective

2.5 Case study object

Given the HVAC project in the department of computer systems at Tallinn University of Technology, it was decided to perform a part as a case study of a thesis. Heating, Ventilation and Air Conditioning (HVAC) is a technology for interiors and vehicles. HVAC is an important part of residential structures such as single family homes, apartment buildings, hotels and old residential areas, medium to large industrial and office buildings such as skyscrapers and hospitals, vehicles such as cars, trains, airplanes, ships and so on. Wang Shengwei et al. in [22] provided a structure for supervisory control of building HVAC systems. They concluded that the technique to optimize the solutions should have high computational efficiency and high ability to find global minimums. Research and development in supervisory control of HVAC systems indicates the significant energy saving or cost savings in buildings when using optimal strategies. Sporr et al. [23] gathered the data in Building Information Modeling (BIM) to optimize a prefabricated building and to change existing controllers. They found a way to identify and complete lost information that is necessary to create an air quality controller.

Compared to the classical optimization methods in the parameter estimation process, the intelligent method will be chosen according to its many capabilities. It seems that the PSO method and its combination is more capable and popular than the others for its accuracy and particularly speed. However, as mentioned in the previous section, this method has its limitations and authors have proposed combinatorial methods to overcome its drawbacks. The other issue is that the efficiency of a method will be achieved after observing the results. Depending on the problem, one technique may perform better than the other. The ABC algorithm can overcome the convergence issues faced by PSO. The thesis continues to combine PSO with GA and ABC methods and to compare its results.

3 OPTIMIZATION TECHNIQUES

The goal of optimization is to achieve the "best" design relative to a set of prioritized criteria or constraints, and to optimize (minimize or maximize) factors such as productivity, power, reliability and longevity. This process is known as optimization. Engineering optimization is a topic that uses optimization techniques to achieve design goals in engineering such as multilayer laminar flow coefficient and clogging filter in HVAC system. Depending on the features of the algorithm, there are several ways to classify optimization algorithms (Figure 1).

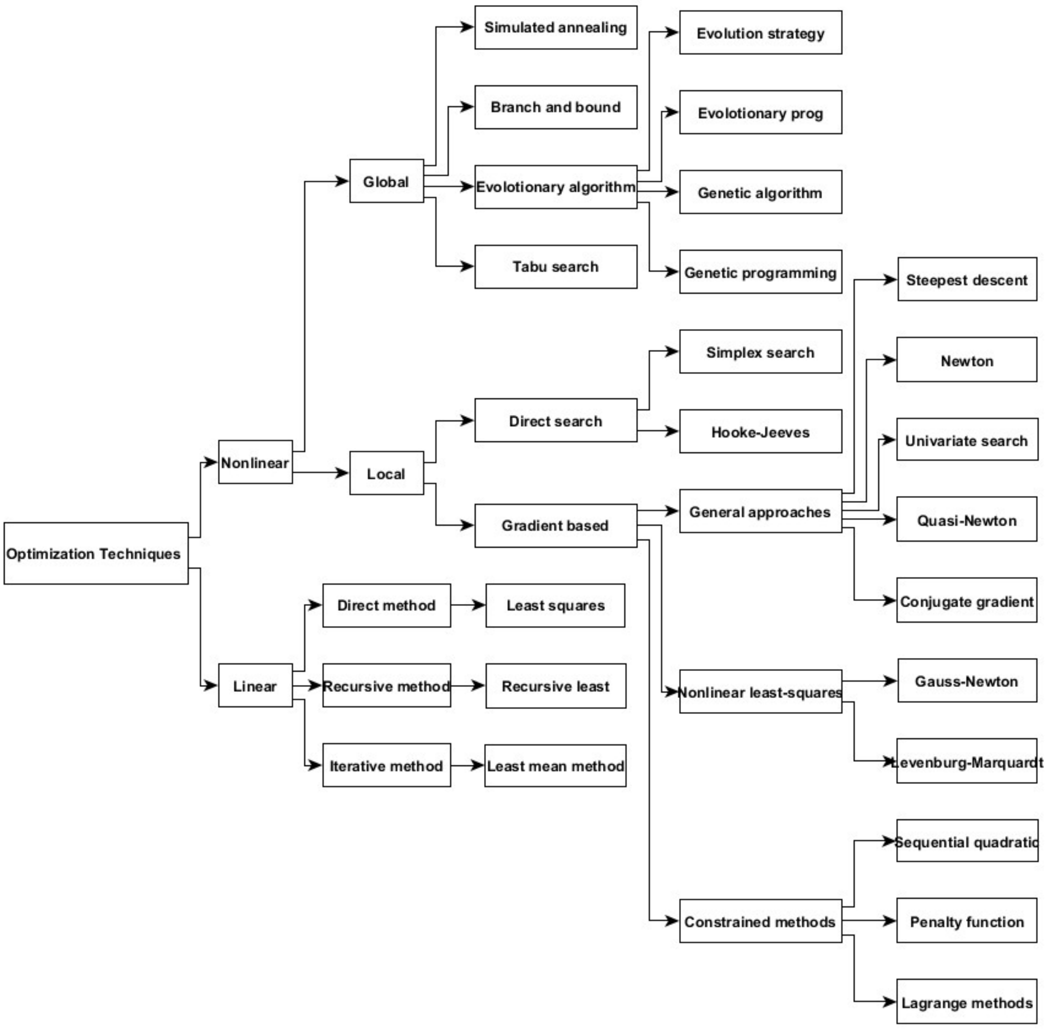


Figure 1. Classification of optimization techniques [22]

3.1 GA theory

Genetic Algorithm is a search and optimization algorithm based on the principles of genetic science and natural selection. Genetic algorithms create a group of individuals and develop in conditions whose overall purpose is to maximize the fitness of the whole population or minimize a population-related cost.

Since the genetic algorithm is derived from both computer science and natural genetics, the terms used in it are a combination of natural and synthetic terms. The basic concepts that are important to understanding a genetic algorithm are:

Chromosomes: The basis of a genetic algorithm is to convert each set of answers into a coding answer. This is called a chromosome. It also called individual, the string and the structure. They can also be called genotypes.

Phenotype: Each chromosome corresponds to a set of answers. The set corresponding to each chromosome is called a phenotype.

Gene: The elements of a chromosome are usually numbers. Genes have been called feature or character.

Location: The location of the gene on the chromosome is called a location.

Population: A set of chromosomes is called a population.

Generation: Each iteration of the algorithm is called a generation.

The pseudo-code of the genetic algorithm is given in Figure 2:

```
Generate the initial population.
Perform the following steps until the conditions are fulfilled:
-   Evaluate, sort, and remove additional members of the population.
-   Select some of the best members of the population as parents, apply
    the intersection operator between them.
-   And create a population of children.
-   Select some of the best members of the population as parents, apply
    the mutate operator between them
-   And create a population of mutants.
-   Merge the main population, the offspring, and the mutant population.

end
```

Figure 2. Pseudo-code of the genetic algorithm

The most important and basic type of genetic algorithm is binary genetic algorithm in which variables are coded binary. This type of genetic algorithm is also called discrete genetic algorithm. Because the variables do not have continuous changes and cannot take any value at all. The set of problem variables, for which the optimal value must be found, is encoded in binary strings and overlapped. Thus a chromosome is obtained from the problem variables. How the binary genetic algorithm works and its different stages can be seen in Figure 3.

3.2 PSO theory

Particle Swarm Optimization (PSO) is a population-based random optimization algorithm that is simulated by the social behavior of a group of birds.

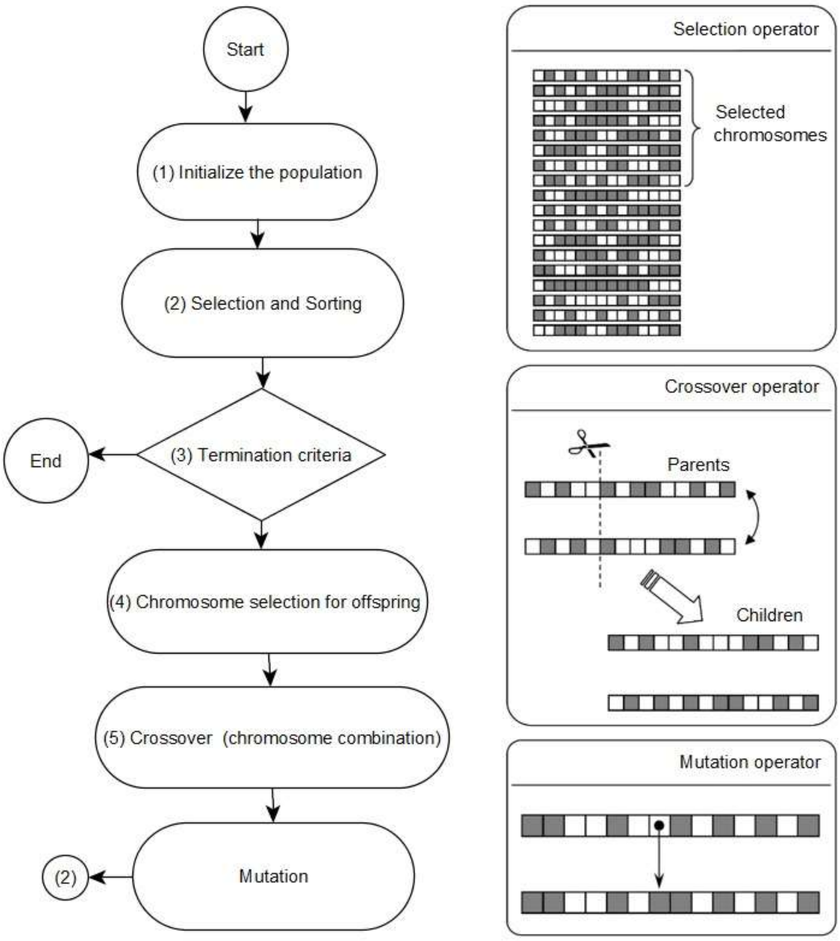


Figure 3. Working principle of binary genetic algorithm and its different steps

In the particle optimization algorithm, there are a number of creatures, which we call particles, and are scattered in a functional search space that we intend to minimize (or optimize). Each particle calculates the value of the objective function in the position of the space in which it is located. It then selects the direction of motion by combining its current location information and the best location it has been in, as well as the information of one or sum of best particles. All particles choose the direction of motion, and once the motion is completed, one step of the algorithm is completed. These steps are repeated several times until the desired result is obtained. In fact, the bulk or swarm of the particles that search for the minimum of a function act like a bunch of birds looking for food. Given this, a new speed is calculated at each stage for the birds. The particle swarm algorithm has two main operators:

- Speed update operator
- Position update operator

At the beginning of the algorithm, a number of birds are randomly generated. Then each of them is assigned a speed. Based on the current speed of the bird and its distance from the best position ever seen or found by the adjacent birds, a new speed is calculated for the bird; and given that the speed value is equal to the amount of bird displacement in one step, the new bird position can be obtained in the next step after updating the position. This process is then carried out to a specified number of iterations and finally the best meeting place for all birds is presented as the answer to the problem.
In order to better understand the assumption, we want to find the highest point in a space of Figure 4. In fact, finding the highest point means to find the objective function in question.

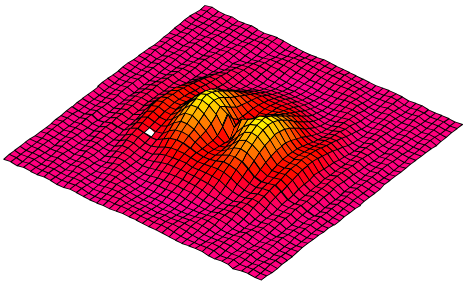


Figure 4. Search space to find the highest point

In order to find the highest point, some particles are randomly generated first, then scattered throughout the space. This is actually like making an initial answer to the problem. For each particle like i, x_i as the particle position, v_i as the particle velocity and y_i as the best position that the particle i itself has ever experienced and defined as the best personal experience. In other words, y_i is the position oh highest point. The highest altitude function, $f^{i,best}$, is the objective function of the problem.
It should be noted that in the first step the y_i is the position of x_i because the particle had only the same experience. Finally, the particle that has the best position in terms of altitude (the objective function of the problem, $f^{g,best}$) among all the particles is defined as $\hat{y}$.
It should be noted that at the beginning of the particle motion (first step) the velocity of all particles is assumed to be zero, and the velocity, v_i, and position, x_i, are then updated according to the formulas of the particle optimization algorithm (PSO). Figure 5 shows the randomly scattered particles in order to find the highest point (the best value of the objective function). Figure 5 also indicates the velocities of each particle by vectors as direction of movement.

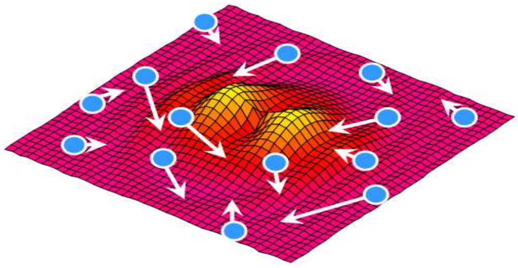


Figure 5. Scattering of particles to find the highest point

After randomizing the particles in space in order to find the highest point and defining the related parameters, the position and velocity of each particle must be updated in order to new search in space. For this purpose, as stated earlier, the motion of each particle, i, is obtained as a form of three vectors, the current vector of the particle, the vector of the best personal experience, and the vector of the best particle position, $\hat{y}$. Figure 6 shows the vectors determining the direction of movement of the particles.

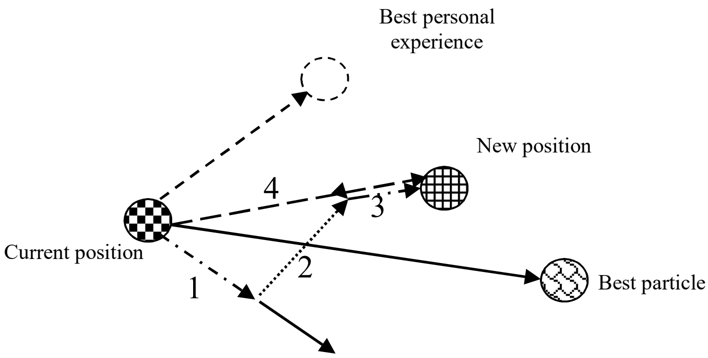


Figure 6. Update Particle velocity and location in PSO Algorithm

It can be seen from Figure 6, vector (1) represents a coefficient as direction of the previous movement of the i -particle. Vector (2) represents a coefficient from the vector of the current position to the location of the best i -particle personal experience,y_i. Vector (3) represents the coefficient as the current position vector to the best particle location, $\hat{y}$. Finally, based on the result of the above three vectors, the new i-particle velocity is denoted as (4). The new position of the i -particle is obtained by transferring the linear position to the new velocity. The relationships are as follows [2]:

$$v_{ij}[t+1] = wv_{ij}[t] + c_1r_1y_{ij}[t] - x_{ij}[t] + c_2r_2\hat{y}_j[t], \quad (1)$$

$$x_{ij}[t+1] = x_{ij}[t] + v_{ij}[t+1] \quad (2)$$

In above equations, w is the inertia coefficient, random numbers with uniform distributed over the interval [0,1] is presented by $r_{1,}r_2$. Learning coefficients are c_1, c_2. Coefficients $r_{1,}r_2$ create a variety of solutions that allow for a more thorough search of space. The learning coefficient, c_1 is related to the personal experiences of each particle, and the learning coefficient, c_2 is related to the collective experiences.

3.3 ABC theory

ABC is a metaheuristic algorithm that was inspired by the seductive behavior of bees, and was proposed by Dervis Karaboga in 2005. This is a simple and powerful algorithm that can be used to solve a wide range of practical and real optimization problems.
The ABC colony consists of three groups of bees including working bees, scouts and onlookers. The number of bees working in the colony is equal to the number of food sources around the hive. The abandoned bee whose food source has been abandoned, becomes a scout and begins to search for a new food source. Onlookers watch the work of working bees and choose food sources depending on their search. The main steps of this algorithm are as follows [24]:

- Initial food sources are produced for all employed bees
- REPEAT
    - Each employed bee goes to a food source in her memory and determines a closest source, then evaluates its nectar amount and dances in the hive
    - Each onlooker watches the dance of employed bees and chooses one of their sources depending on the dances, and then goes to that source. After choosing a neighbour around that, she evaluates its nectar amount.
    - Abandoned food sources are determined and are replaced with the new food sources discovered by scouts.
    - The best food source found so far is registered.
- UNTIL (requirements are met)

In the ABC algorithm, the first half of the crowd includes working bees, and the second half is made up of onlooker bees. The number of working bees is equal to the number of solutions. The ABC randomly generates the initial distributed population of SN solutions (food sources), where the SN shows its size.
Let $X_i = \{x_{i,1}, x_{i,2}, x_{i,3}, \dots x_{i,n}\}$ represent the i^{th} solution in the swarm, where n is the dimension size.
Each employed bee x_j generates a new candidate solution V_i in the neighborhood of its present position as equation below [24]:

$$v_{i,k} = x_{i,k} + \emptyset_{i,k} \times (x_{i,k} - x_{j,k}) \tag{3}$$

where x_j is a randomly selected candidate solution ($i \neq j$), k is a random dimension index selected from the set $\{1,2,\dots,n\}$, and $\emptyset_{i,k}$ is a random number within [-1,1]. Once the new candidate solution v_i is generated, a greedy selection is used. If the fitness value of v_i is better than that of its parent x_i, then update x_i with v_i; otherwise keep x_i unchanged. After all, working bees complete the search process. They share information about their food resources with onlooker bees through wagon dances. The onlooker evaluates the nectar information obtained from all working bees and selects a food source with probability related to the amount of nectar. This choice is probably a roulette wheel selection mechanism, which is described as the following equation:

$$P_i = \frac{fit_i}{\sum_j fit_j} \tag{4}$$

where fit_i is the fitness value of the i^{th} solution in the swarm. As seen, the better the solution i, the higher the probability of the i^{th} food source selected. If a position cannot be improved over a predefined number (called limit) of cycles, then the food source is abandoned. Assume that the abandoned source is x_i, and then the scout bee discovers a new food source to be replaced with i^{th} as equation below :

$$x_{i,k} = lb_k + \emptyset_{i,k} \times (ub_k - lb_k) \tag{5}$$

where $\emptyset_{i,k} = rand(0,1)$ is a random number within [0,1] based on a normal distribution, and lb_k, ub_k are lower and upper boundaries of the k^{th} dimension, respectively.

3.4 Introducing Dynamic Systems Identification Software Package

Based on the concepts introduced in previous chapter, a comprehensive software package is implemented to identify and model dynamic systems using intelligent optimization algorithms using Matlab programming and Simulink software. This package is applicable to any dynamic model graphically designed in Simulink, and therefore offers a unique approach to system modeling and system identification and tools. It should be noted that all parts of this software have been implemented by the executives of this research project. It can be considered as a complementary tool to the Matlab software, which can meet the ever-increasing needs which are expanding in various research fields. In the following we will try to review the structure of this software package and implicitly teach how to use it.

In models with physical descriptions, the actual parameters are used and all the parameters have a meaningful dimension. For example, the value of resistance, the hardness coefficient of a spring, the heat capacity of a material and so on are physical parameters. Experts have unique concepts for these dimensions. In this type of modeling, the main problem is to determine the appropriate and correct parameter values. In this case, the system identification is reduced to parameter identification since the structure of the model is sure enough clear, and the main problem is to find the unknown parameters. But the presence of an expert who is proficient in the details of the system and can provide a physical description of the phenomenon is a prerequisite for this type of identification and modeling.

In contrast, there are models which parameters do not have a clear physical meaning. Rather, the parameters and unknowns in this type of model are a combination or function of the physical parameters. For example, the damping coefficient in a second-order dynamic system (mechanical or electrical) is a function of resistance, capacity and induction coefficient (for electrical system) or mass, spring stiffness and dumping (for mechanical system). Therefore, the parameters themselves have no independent meaning and have only a descriptive effect on the system.

In some cases, there is no model at all and the type and order of the model must be determined, in such case, for non-linear systems, there is basically no precise and universal approach. On the other hand, for linear systems, the type and order of model can be determined by the methods described earlier (Figure 1). However, one of the advantages of intelligent optimization and developed software package is that this tool can also be used to determine the type of a model. As an example of using this package, it can be mentioned in the design of linear and nonlinear controllers. This package also allows us to optimizes the controller's coefficients. For instance, it can be used to find the best PID controller coefficients for nonlinear systems.

Alternatively, this tool can be used to optimize the performance of a control approach (eg sliding mode control or SMC) and adjust its parameters and optimize its performance by means of the algorithms embedded in the package. As a result, the parameters and even the controller structure that performs best and most efficiently will be provided by the software.

The impressive flexibility of this package makes it a simple (practical) and versatile tool. Certainly, examining the models and issues that have been resolved in this thesis confirms the applicability and high use of this software package.

This package has several files; each is described below:

Genetic algorithm implementation files

ga.m	It is a script that contains the implementation of the structure of the continuous genetic algorithm.
ArithmeticCrossover.m	It is a function in which the operator of the computational junction for continuous problems is implemented.
Mutate.m	It is a function in which the mutate operator is implemented for continuous problems.
RouletteWheelSelction.m	It is a function that has been implemented to sample a discrete distribution. This process is known as Roulette Wheel Selection in computational texts and genetic algorithms books and is used to select parents in each generation.

PSO implementation files

pso.m	A script containing the definition and implementation of the particle swarm optimization algorithm, or PSO.

ABC implementation files

abc.m	A script containing the definition and implementation of the artificial bee colony algorithm, or ABC.

PSO-GA implementation files

psoga.m	A script containing the definition and implementation of the hybrid of PSO and genetic algorithm.

PSO-ABC implementation files

psoabc.m	A script containing the definition and implementation of the hybrid of PSO and artificial bee colony algorithm.

Simulation files and connection to Simulink

DefineProblems.m	Practical script in which all the information of the problem is defined. The only place where the user actually needs to make changes is this file.
GetParams.m	It includes a function definition that receives parameter values from the optimization algorithm and prepares them for use in the Simulink model.
LoadNominalData.m	The measured and experimental data is called by this file and prepared for use in the optimization and modeling process.
MyCost.m	containing cost function. In this file, optimization criteria are defined mathematically. In this way, minimizing these criteria means improving the proposed model.
SimulateModel.m	This file implements the core of the optimization section. The parameters’ value obtained by the optimization algorithm are transferred to the Simulink file. The simulation is then performed and the result is returned.

Shared files

GetClockString.m	This file is to create a numeric string using the computer's clock and date. This is done so that, when the modeling program is run several times, the results are stored in separate files and do not loss of the previous modeling results.
PlotResults.m	Plots the obtained best output from the modeling.

3.5 Solving a simple example

One of the most widely used models in scientific studies are populated models that have many applications in the fields of biology, medicine, medical engineering, health, sociology and economics. For example, assume that we have a system which increasing the population of one quantity, x, leads to decreasing another, y and we have a nonlinear differential equation as:

$$\dot{x} = ax - bxy \tag{6}$$

$$\dot{y} = -cx + dxy \tag{7}$$

Where all the parameters a, b, c, d are assumed positive, and of course the equations are implemented in such a way that the values of the states are always positive.

The positive and negative effects of the population of x and y on each other are fully reflected in the above model. The measurements have also been made and we want to find the best values of the 4 parameters of the model, so that it has the best and most compatibility with the measured data.

The dynamic model of the system is implemented in the form of a graphical model in SIMULINK. In this model, unknown parameters are entered with the corresponding variable name. In addition, outputs are delivered to Matlab in the form of arrays with the general name of simout. The target variables and the tracked outputs in the dynamic system must always be stored in the form of the simout and delivered as an array to Matlab (Workspace). Figure 7 shows the graphic design of the model:

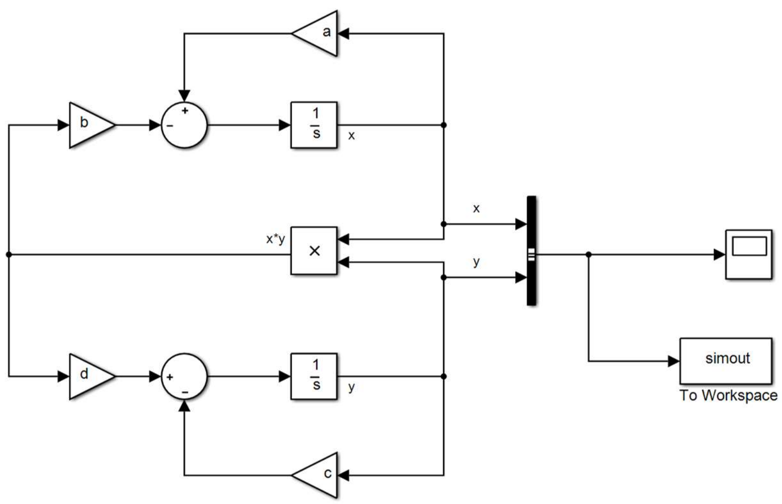


Figure 7. Graphic implementation of the model in SIMULINK

The goal is to minimize the objective function (Eq.(9)). For general approach programming, almost two algorithms use the same functions.

In order to define this problem in software package, first DefineProblem.m file is changed. The code for this problem can be found in appendix 0.

In this program, first the model name (which is the Simulink file name) is defined. Then, different parameters are defined. In the parameter definition, the range of changes are determined. Based on parameters definition, the program calls the optimization algorithm to estimate the parameters.

By applying the changes in the cost function, the synchronization between the optimization problem and the optimal parameters determination can be obtained. This code is shown below. In this code, the sum of the mean squares error between the model output and the experimental data (here nominal data) is calculated and returned as an output of the cost function. the code can be found in appendix 0.

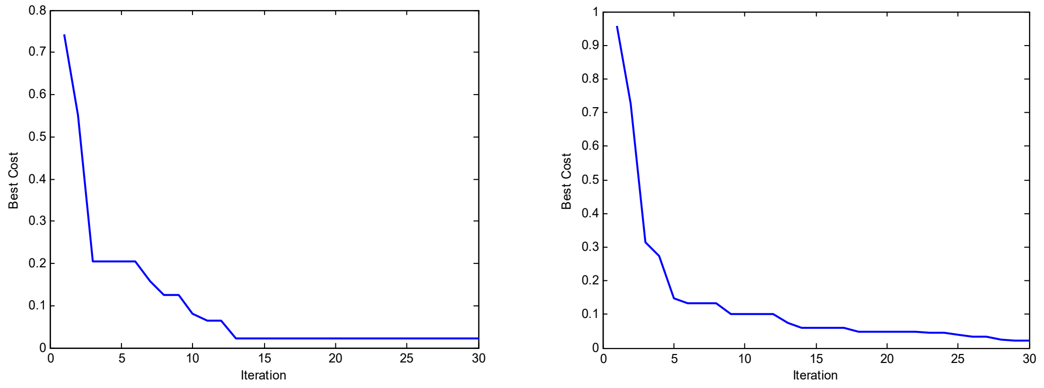


Figure 8. Convergence plot of GA(left) and PSO(right) algorithm

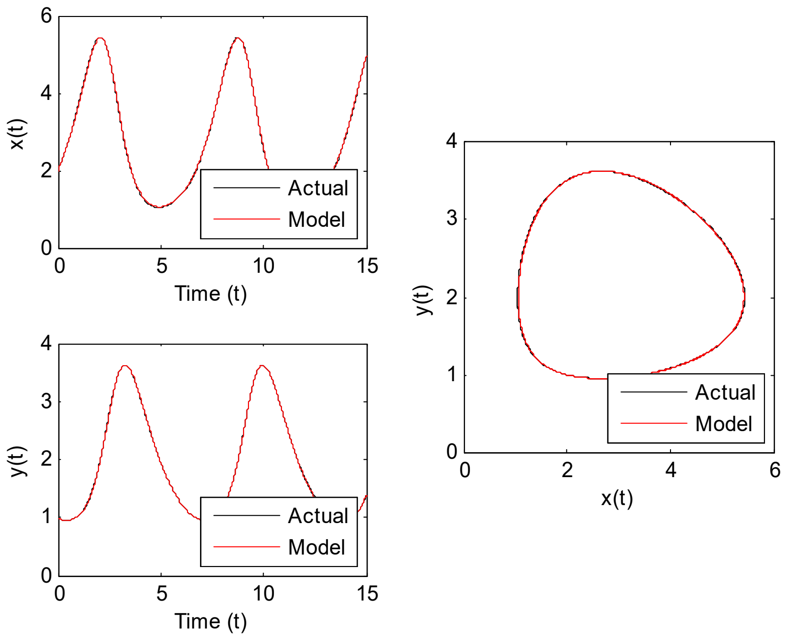


Figure 9. Comparison of actual and model proposed by Genetic Algorithm

Table 7. Simulation results, nonlinear dynamic model

Parameter	PSO	GA	Nominal or desired value
a	1.1908	1.1876	1.2
b	0.5943	0.5915	0.6
c	0.8049	0.8090	0.8
d	0.3028	0.3029	0.3
J, cost function	0.0237	0.0242	0

Figure 8 and Figure 9 show the final results. The first plot relates to the process of decreasing the cost function, known as the evolution or convergence graph. The second is the plot of matching and comparing the modeling results with the nominal model data. Table 7 shows the unknown parameters using two algorithm search. The PSO algorithm has minimized the cost function more compare to GA. In the next chapter, we will see that PSOABC algorithm even is better than PSO for HVAC filter. Compare to nominal value, the results are satisfied. For example, for estimating unknown parameter a, the error is about 0.76%. For b, c, d the deviation is about 0.95%, 0.61% and 0.93% respectively.

4 PARAMETER ESTIMATION AND CLOGGING PREDICTION

This chapter and obtained results is based for conference or journal article. This chapter presents the literature review of the method used to model, estimate unknown parameter and prediction of components in the HVAC systems. Almost all HVAC systems, utilize the filter to provide clean and fresh air to the zones. Having an effective model could be important tools for filter designers, building designers and building energy managers as well as those who are attempting to optimize building energy performance through the use of dynamic model-based control systems.

In the design process in research laboratories as well as in industries and also for control purposes, one of the problems encountered after mechanical modeling is the measurement of physical parameters and coefficients. This can sometimes be very time consuming and costly. The purpose of this work is to estimate parameters based on physical based dynamic model by intelligent optimization methods in applying filter performance models and the identification of model parameter values required to make performance models useful and accurate. A dynamic-based model has been developed which, when combined with standard test data provided by system manufacturers or measured data provided by BIMs, allows the modeler to identify the parameters that govern the behavior of the system particularly pressure drop.The estimated parameters were obtained using real measurements. Different HVAC system data from building sites of Tallinn and Helsinki is used which the results have shown the ABC and PSO-ABC have good performance for estimating the unknown parameters of filters in HVAC systems. Both polynomial and exponential patterns can be used for prediction of filters clogging in some finite range.

The parameter estimation divided into two parts, estimating laminar flow transition in the filter model from sequential (on/off) air handling unit and also prediction of clogging the filter from process (continuous) air handling unit.

4.1 System description

HVAC systems have been used around the world as an important part of energy consumption. Therefore, studies are needed to predict the energy consumption of HVAC components. For this goal, demonstrating these components using the equivalent model and estimating its parameter is an important area for study. Usually manufactures of HVAC components modules provide some information to obtain model parameters using standard test conditions.

In this research a method for parameter estimation of equivalent filter model is proposed using real measurements. Data has been used from the Tallinn site (unit 333, 467, 469, 471,474,479) and Helsinki site (units 1740, 1741, 1742, 1743, 1744, 1745). The application of intelligent algorithms for parameter estimation and clogging prediction of a system based on real measurements is designed. The filter part of HVAC system was selected but the approach to solve the problem is general which can be represented in a form of nonlinear dynamic graphical model in SIMULINK and unknown parameters are entered with the corresponding variable name. The filter was chosen due to find parameters in order to be able predict future clogging values and get to know when we need to change the filter in order to save finances. Furthermore, filter behavior makes it possible to use mentioned techniques.

Methods available in Matlab Parameter Estimator Toolbox are: Gradient descent, Nonlinear least squares, Pattern search, Simplex search. Pattern search including Positive basic Np1-2N, Genetic algorithm, Latin hypercube and Nelder-Mead.

Thus other intelligent pattern search base methods like ABC, ant colony or hybridization of any type of algorithm are not available. Other Additional drawback of the parameter estimation toolbox in Matlab is about cost functions which is only Sum Squared Error and Sum-Absolute Error. Proposed application provides possibility to use this work it is possible to use any type of search algorithm and any metrics to evaluate the accuracy of estimation.

4.2 Optimization method for parameter estimation of nonlinear systems

Consider a nonlinear implicit system modelled by:

$$y = f(x; y; u; p) \tag{8}$$

where that x, y, u, p are states, outputs, inputs and unknown parameters of the model, respectively. In addition, f is a nonlinear function.

The parameter estimation process is formulated as a nonlinear optimization problem. For this purpose an objective function (fitness function) which measures the proximity of the actual system output (y_r) obtained from sampled measures in relation to the output (y) of the solution of implicit mathematical model (Eq.(8)) for value of the parameter vector p_i:

$$\min(J(p)) = \sqrt{\left(\frac{1}{n}\sum_{i=1}^{n}(x_i - \hat{x}_i)^2\right) + \left(\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right)} \tag{9}$$

$$s.t. \qquad p_{imin} < p_i < p_{imax}$$

Where n is the number of samples measured. The search interval of the parameters is defined by p_{imin} and p_{imax}.

The GA, PSO, ABC, PSOGA and PSOABC optimization algorithms were used to estimate the parameters. Obtained values and real measurements are compared and the cost function which is difference between those is minimized by means of parameter changing.

4.3 Modeling and parameter estimation of the return filter in HVAC system

The filters in HVAC systems are components that make resistance through flow and drop the pressure in an air stream network branch. There are two types of common filter called pleated and bag filter.

The consistency of the obtained unknown parameter depends on the geometry, clogging and properties of the filter. However, sometimes it is difficult to calculate these parameters with some additional software like finite element software. In addition, preparation of facilities to measure the parameters undergoes the high cost. Here, we have approached two methods, first using operating points in order to estimate the coefficient of laminar flow transition. Second, using time variable coefficient of clogging in the filter model which is exactly like practical filters in the sites. In first method, the unknown parameter is determined from a nominal operating condition. In other words, it is useful for representing complex components, where it is difficult to determine the theoretical pressure drop of a clogged filter and its geometry. Thus if the pressure drop versus nominal flow rate is mentioned in the datasheet of manufacture, or measured data is available, it is possible to obtain the pressure drop as a function of its mass flow rate during its working time. This filter identification method is useful since the result can be applied to other components like heat coefficients of heat exchanger. The mass flow rate through the inlet must be exactly the same as the mass flow rate through the outlet. Energy can enter and exit the filter through the fluid ports. It is assumed that no heat exchange occurs between the wall and the environment. In addition, no work is done on or by fluid. The energy flow through inlet must be exactly equal to the energy flow through outlet. It is assumed that this parameter is constant during the time.

External forces on the fluid include those due to pressure and those caused by viscous friction in component walls. Body forces like gravity are ignored. The expression of friction forces in terms of a loss factor ξ yields the semi-empirical expression. This type of formulation is to allow for a change in sign as the direction of the stream: [25]

$$\Delta p = \xi \, \frac{\dot{m}\,|\dot{m}|}{2\rho S^2} \tag{10}$$

where:

Δp is the pressure drop through input and output of filter, ξ is the loss factor. Fluid density describes by ρ and the flow area by S.

It is assumed that the resistance is adiabatic. It does not exchange heat with the environment.

The drop of pressure is proportional to the square of the mixture mass flow rate (dry air, water vapor and other gas like CO_2) and inversely proportional to the mixture density (or covertly inlet temperature and pressure). The constant of proportionality (ratio between two directly proportional quantities) is determined from a nominal operating condition and it is used when empirical data on the pressure losses and flow rates through a component is available, but detailed geometry information is unavailable.

For modeling, the loss factor ξ is not required as a parameter. Instead, it is automatically computed from the nominal condition derived in the current thesis as:

$$\frac{\xi}{2S^2} = \frac{\rho_* \Delta p_*}{\dot{m}_*^2} \tag{11}$$

where the asterisk (*) denotes a value at the nominal operating condition. It can be selected from desired and appropriate point of empirical data.

$$\rho = \frac{p_{inlet}}{RT_{inlet}} \tag{12}$$

$$\Delta p = \frac{\rho_* \Delta p_*}{\dot{m}_*^2} (\, \dot{m} \sqrt{\dot{m}^2 + \dot{m}_{lam}^2}) \frac{R\, T_{inlet}}{p_{inlet}} \tag{13}$$

$\dot{m}_{lam}$ mixture mass flow rate threshold for laminar flow and is equal to

$$\dot{m}_{lam} = f_{lam} \times \dot{m}_*, \tag{14}$$

where:

f_{lam} - fraction of nominal mixture mass flow rate for laminar flow transition;

$\dot{m}_*$ - nominal flow rate obtained from filter data sheet or empirical data;

p_{inlet} - inlet pressure Pa;

T_{inlet} - inlet temperature K;

ρ -inlet temperature kg/m^3;

R_a - dry air specific gas constant, 287.048; $J/(kgK)$;

R_w - water specific gas constant, 461.52; $J/(kgK)$;

R_g - trace gas specific gas constant, 188.92; $J/(kgK)$;

The $R = (R_a + R_w + R_g)/3$ is mixture specific gas constant.

The measurements for the estimation (u_r, y_r) were obtained from a building information model (BIM) and building automation system (BAS) and applied for building in Tallinn. This system is presented in Figure 10. There is two filters, one in supply air channel and another in return. The measured pressure in the filters are used for periodic overhaul and replacement of filters in the case of over pressure drop.

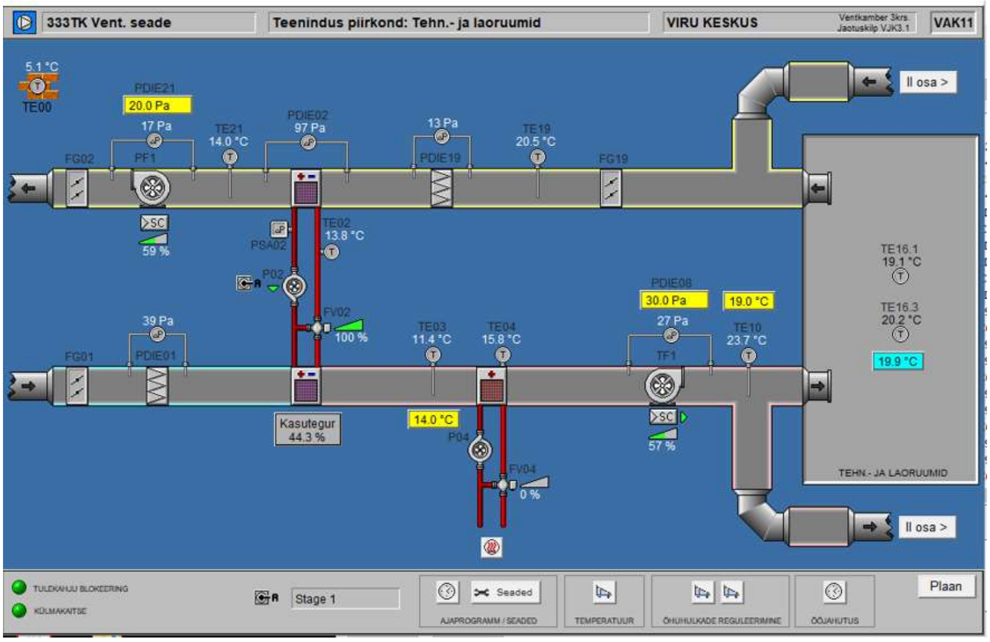


Figure 10. Diagram of filter in AHU of HVAC system

Measurement intervals of pressure drop were registered every 15 min and data is raw without any mean value calculation. The measurements are composed of set of one year. Before using the method, the parameter search region (upper and lower limits of parameters) need to be defined. To define upper and lower limits, a region can be created using as initial value the parameter obtained from datasheet at standard test condition. However, in the case of unavailable datasheet, the expert can set the lower and upper limits in the range of common value in the similar component. Selecting the infinity range will increase the estimation time and leads to other local optima.

The data from unit 333 which operates as on/off controller or thermostat is used for this section. The nominal flow rate 2.71 m^3/s, and nominal pressure drop 43.3 Pa is selected from mean of empirical data. Laminar fraction that is going to be estimated is initially 820e-3 and assumed the density is constant and sample time is equal to 15 min.

Based on the concepts introduced, a comprehensive software package (including Simulink, mfiles and algorithms codes) is implemented to identify and model dynamic systems based on intelligent optimization algorithms using Matlab and Simulink software. This package is applicable to any dynamic model graphically designed in Simulink, and therefore offers a unique approach for system identification and modeling. Each searching algorithm starts Simulink (linear or nonlinear) model independently. Alternatively, this tool can be used to optimize the performance of a control approach (eg sliding mode control or SMC) to adjust its parameters and optimize its performance by means of the algorithms embedded in the package. As a result, the parameters and even the controller structure that performs best and most efficiently will be provided by the software.

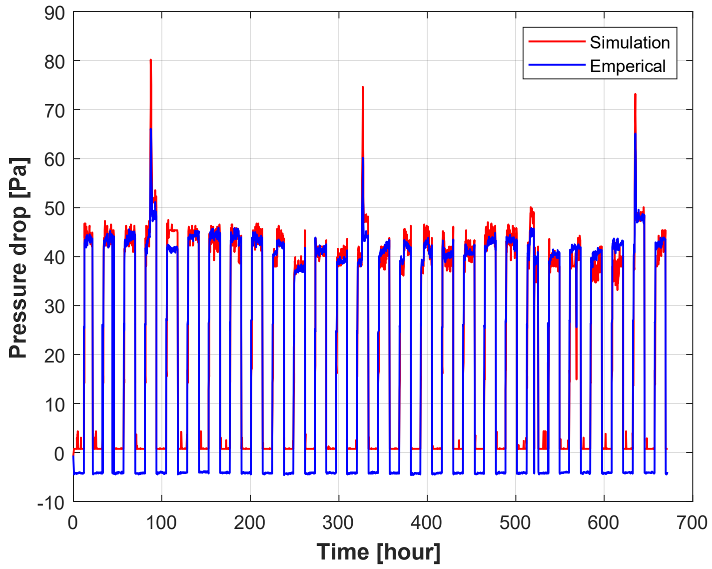


Figure 11. Supply filter pressure drop over time, unit 333

Figure 11 and Figure 12 show the result of modeling and simultaneously using PSOABC algorithm for finding the unknown laminar flow parameter to fit the output with empirical data of unit 333. The offset in Figure 12 is caused by misreadings in DAQ module and is not possible in real situation.

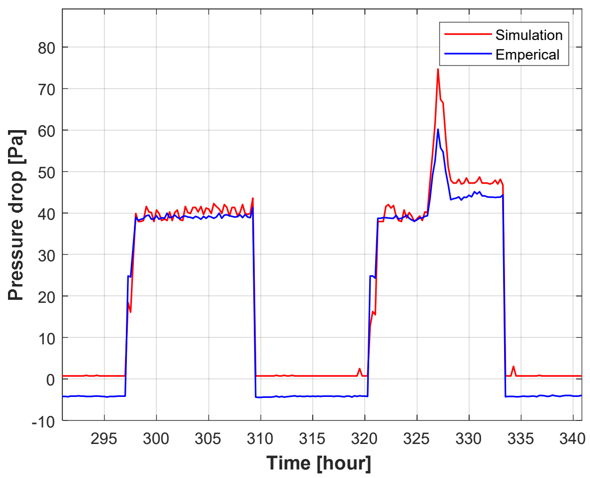


Figure 12. Supply filter pressure drop tracking

The tracking is acceptable, and thus, obtained unknown parameter is estimated. The summary of comparisons among algorithm are listed in Table 8.

The method is also applied to 1745-unit data due to its continuous behavior. The tracking is acceptable and means that obtained unknown parameter is reliable. The set of return filters is selected since the clogging rate of return filter is higher than supply filter due to sources like cooking, odour material and air molecule with more surface groups.

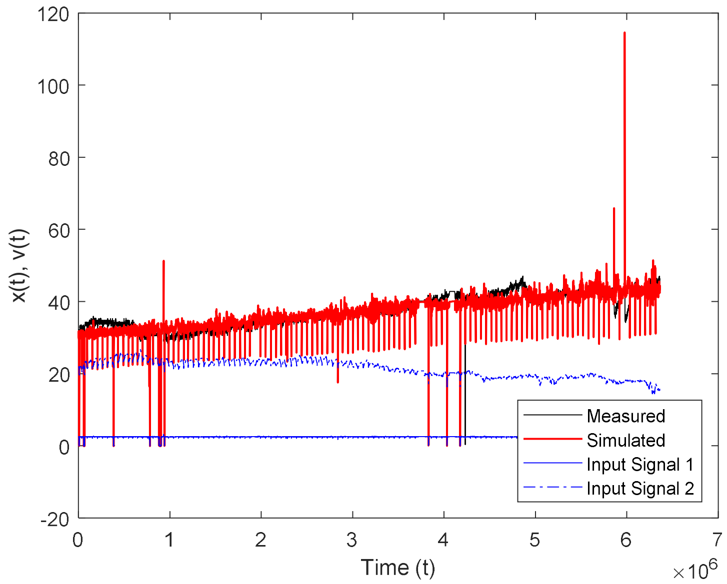


Figure 13. Return filter pressure drop, unit 1745

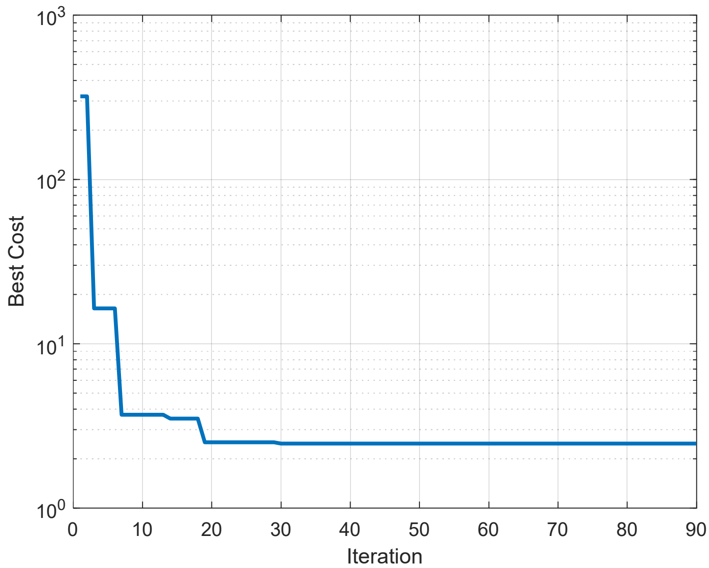


Figure 14. Cost function history, ξ estimation

Table 8. Algorithm Comparison results

Algorithm	GA	PSO	ABC	PSO-GA	PSO-ABC
Pop size	80	80	50	50	50
f_{lam}	800e-3	761e-3	400e-3	687e-3	420e-3
R	0.95	0.94	0.95	0.95	0.99
RMSE	3.18	5.37	2.89	3.22	2.77

In Table 8, the result of algorithms with different sizes of the population is summarized. In order to find the best solution, the root mean square error (RMSE) is calculated which indicates the minimum error or minimum pressure difference between the simulated and measured data. Pearson R correlation coefficient between simulated and measured data is also calculated for observing best tracking beside minimizing error. Comparing the results, the PSOABC method with the population size of 50 is the best solution. The PSOABC algorithm has been used for next part of this research which is estimating the dynamic of filter clogging.

4.4 Filter clogging estimation and prediction in HVAC systems

In previous section, it is assumed that the clogged filter has static behavior during time and we only selected the nominal or working point described before. However, the filter clogging coefficient is changing during the time. This change is much more noticeable in return air duct.

It is possible to present the clogging by the dynamic function which is related to pressure drop, input pressure and temperature to filter and flow rate.

$$f(t)_{clog} = \frac{\Delta p}{\dot{v}\sqrt{\dot{v}^2 + \dot{v}_{lam}^2}\,\frac{p_{inlet}}{R\,T_{inlet}}} \tag{15}$$

In this derived equation in the current thesis, the $\dot{v}$ is flow rate, m^3/s. Since $\dot{v}$ is located in denominator, the output in the case of no flow rate can be diverge. We know that $f(t)_{clog}$ is changing during the time due to molecular aggregation and can result in deterioration of product quality and longer processing times and thus, loss of energy. For model of this occurrence, and to avoid denominator divergence, the polynomial and exponential functions have been presented as the model the clogging, then the coefficients of these function obtained by mentioned intelligent method.

$$f(t)_{\mathrm{clog1}} = K_1\, t^2 + K_2\, t + K_3 \tag{16}$$

$$f(t)_{\mathrm{clog2}} = L_1\, e^{L_2 t} + L_3 \tag{17}$$

It is obvious that the inputs of model including flow rate and temperature which change the density of moist air are time-varying imported from experimental data.

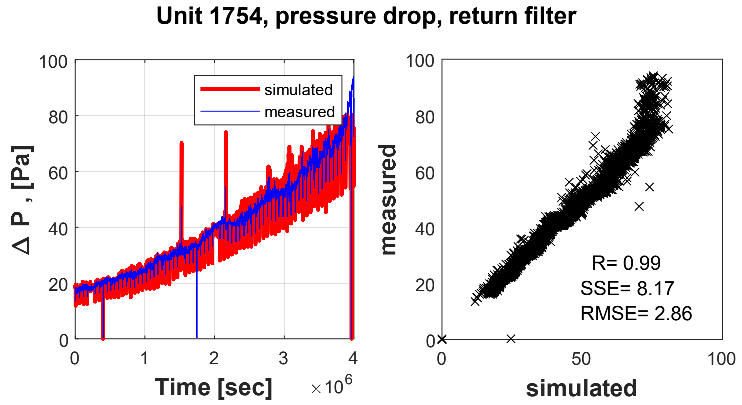


Figure 15. Polynomial model result

The software package runs the model based on PSOABC algorithm and minimizes the cost function to estimate the clogging coefficient which is error of simulated and measured data. Figure 15 shows the result of estimation. The estimated clogged coefficients are shown in Table 9.

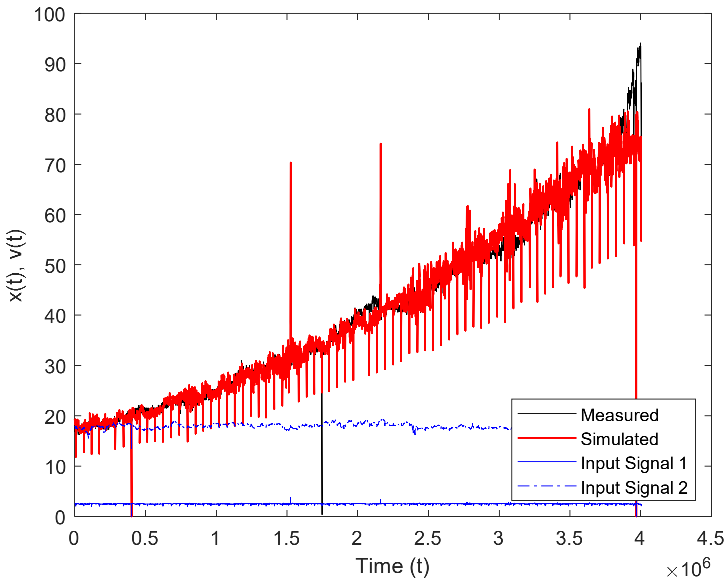


Figure 16. Pressure drop (measured and simulated), inputs (Temperature and flow rate)

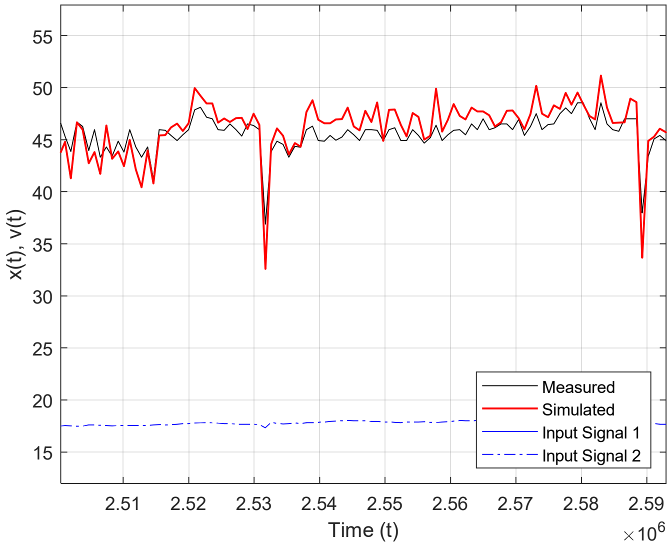


Figure 17. Tracking of final result

Apart from parameter estimation approach, it is necessary to check the graphs to be sure that simulated data tracks the measured data properly (Figure 17). The reason is that; the cost function is calculated by the RMSE equation. It sometimes has been seen the RMSE is low even with unsuitable tracking. In other words, it is possible to have better RMSE with weak tracking or vice versa.

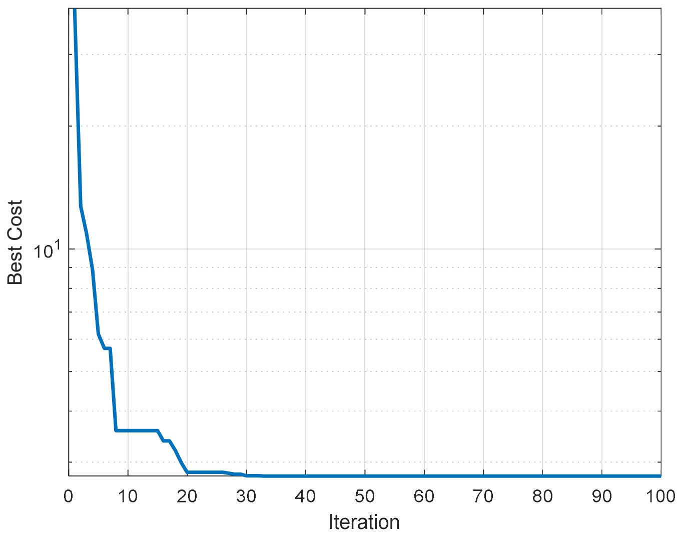


Figure 18. Cost function history, clogging estimation

4.5 Other units

To ensure the verification of result and to predict the clogging behavior, additional data has been used. Those include AHU data from Tallinn site 467, 469, 471,474,479 starting from January 2019 as well as data from Helsinki units 1740, 1741, 1742, 1743, 1744 starting from July 2019.

Above mentioned procedure was applied to all units. The result is summarized in Table 9. For some units, the exponential model has better correlation, for instance, R=0.97 in unit 1740, but in general the polynomial model has better performance. Apart from small RMSE or high R, the tracking of signal should also take into account. Polynomial pattern for clogging estimation in unit 1744 is not satisfactory. The reason is due to high SNR which affected the algorithm to recognize the data as polynomial shape. The same happened in 479 on Tallinn site. The graphs and results are added in the appendix 0.

Table 9. Clogging estimation for different AHUs

Unit	Polynomial	Exponential
1740	K1 = 1.4982e-13 K2 = 1.1279e-20 K3 = 7.6221 R=0.95 RMSE=4.88	L1 = 0.16315 L2 = 5.4651e-07 L3 = 8.1607 R = 0.97 RMSE = 2.63
1741	K1 = 1.6804e-13 K2 = 4.9267e-18 K3 = 11.721 R = 0.95 RMSE = 3.18	L1 = 0.19125 L2 = 7.1593e-07 L3 = 11.541 R = 0.95 RMSE = 3.22
1742	K1 = 1.0771e-13 K2 = 8.0959e-21 K3 = 6.9975 R = 0.94 RMSE =4.96	L1 = 0.20098 L2 = 4.6134e-07 L3 = 7.0652 R = 0.95 RMSE = 4.44
1743	K1 = 2.5828e-13 K2 = 1.9564e-20 K3 = 2.335 R = 0.97 RMSE = 2.57	L1 = 0.58915 L2 = 5.3708e-07 L3 = 1.644 R = 0.97 RMSE =2.56
1744	K1 = 3.3533e-14 K2 = 1.9012e-20 K3 = 0.48454 R = 0.82 RMSE = 4.00	L1 = 0.036141 L2 = 5.5991e-07 L3 = 0.63698 R = 0.78 RMSE = 3.39
1745	K1 = 3.37e-13 K2 = 7.65e-07 K3 = 2.54 R = 0.99 RMSE =2.86	L1 = 0.17109 L2 = 4.2036e-07 L3 = 6.1421 R = 0.98 RMSE = 2.91
467	K1 = 8.8444e-14 K2 = 8.9899e-07 K3 = 15.959 R = 0.86 RMSE = 19.67	L1 = 0.18306 L2 = 7.1988e-07 L3 = 17.131 R = 0.81 RMSE = 21.39
469	K1 = 4.9701e-13 K2 = 3.0123e-07 K3 = 13.914 R = 0.91 RMSE = 23.09	L1 = 1.5794 L2 = 4.6837e-07 L3 = 12.93 R = 0.91 RMSE = 23.73
471	K1 = 1.6705e-15 K2 = 3.3374e-10 K3 = 1.3081 R = 0.95 RMSE = 6.60	L1 = 1.7813e-05 L2 = 6.7614e-07 L3 = 1.4364 R = 0.94 RMSE = 5.83
474	K1 = 1.1353e-14 K2 = 1.2732e-07 K3 = 0.60997 R = 0.92 RMSE = 14.20	L1 = 0.058933 L2 = 3.1593e-07 L3 = 1.1639 R = 0.89 RMSE = 16.05
479	K1 = 1.8719e-21 K2 = 6.3969e-15 K3 = 8.6174 R = 0.94 RMSE = 44.01	L1 = 0.1682 L2 = 2.1296e-07 L3 = 1.2532 R = 0.78 RMSE = 21.58

The next step is prediction of the filter clogging. As the AHU airflow is reduced in clogged filter, less and less air from the duct will pass. In supply air duct, the cooling water in coil is still flowing through the coils but with no air-flow, there is no heat absorbed into the coils and refrigerant. If the temperature of cooling coil falls below the freezing point, moisture in the air around the coil begins freezing. If the air filter is changed, problem remains until the evaporator coil is defrosted. Technicians change filters according to the general rule once or twice per year. In many cases time of change does not much depend on the rate of clogging of the filter. Thus, the overhaul time prediction is important for building sites in order to optimize the energy consumption in HVAC systems.

Figure 19 shows the prediction for Helsinki units. The constant inputs are considered to simplicity. The pressure drop due to clogging is predicted for next 60 days. Result shows that the polynomial pattern for estimating the clogged filter is more realistic. In exponential pattern, the pressure drop increases dramatically which is not very close to real site data.

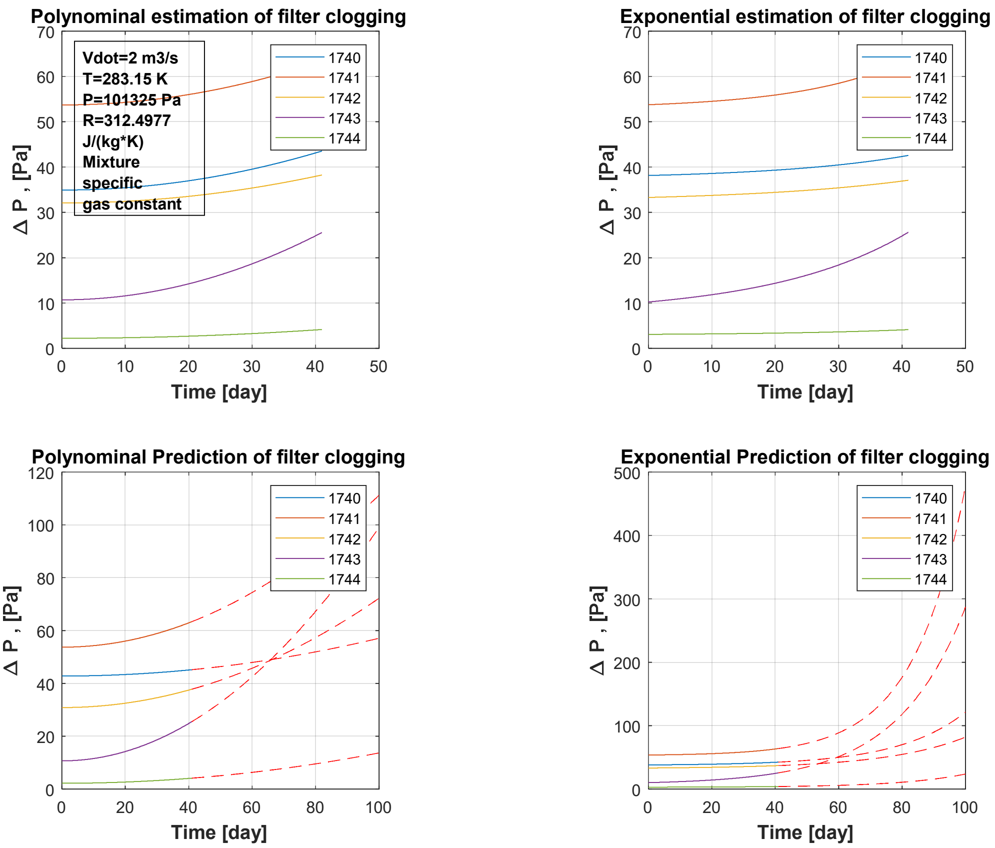


Figure 19. Clogging prediction

4.6 Conclusion

Based on results provided above, the performance of polynomial model for clogged filter leads to acceptable correlation between model and test. However, for more accurate exponential clogged filter estimation, the larger amount of data (without any disruption) in one year is needed. The reason is that maybe in clogged filter, the rate of occlusion is higher than clean filter. In the other words, this occlusion may decrease the ventilation of air through filter exponentially especially in the longer time period. It is worth mentioning that the effect of a clogging on pressure drop can be presented as a second order polynomial, as it can be seen from the results.

Presenting a clogged filter with an exponential function has the advantage that we will never encounter a declining function during estimation and evaluation, which is consistent with the concept of clogging. Although estimation with a polynomial function provides better correlation value, one should take into account that the second-order function is always ascending and has no minimum point.

The obtained results show that clogging of the filter can be estimate with polynomial and exponential function. However, the power function, $f(t)_{\mathrm{clog3}} = P_1\, t^{P_2} + P_3$ can also be examined and compared.

5 PHYSICS-BASED DYNAMIC MODELING OF HVAC SYSTEM

The main purpose of this chapter is to model a dynamic of the HVAC system. Modeling of HVAC systems is essential for the study, evaluation and optimization of energy consumption and indoor air quality. Effective and accessible models of system performance can be an important tool for HVAC component designers and building energy managers, as well as those who use model-based dynamic control systems to optimize building energy performance.

HVAC modeling techniques are divided into data-driven or black box, physics-based, and gray-box models. Models can also be classified as linear or nonlinear, static or dynamic, explicit or implicit, discrete or continuous, definite or possible and deductible, inductive or floating. Gray box models are combined models including block-box and physic-based model. Both physics-based and data-driven methods can lead to linear or nonlinear, static or dynamic models, and explicit or implicit. Physics-based techniques usually lead to continuous and definite models. However, data-driven techniques generally lead to discrete and definite or random models.

During the development of physics-based HVAC system models, static models are commonly used for slow-moving like temperature and humidity process. Dynamic ones are used for fast movement like the temperature of the mixed air and the concentration of carbon dioxide (CO_2) in the mixing box, and the flow rate and water flow through the damper and valve, respectively. Other example of dynamic models are fans and pumps. This is because fast-moving processes are many orders of magnitude faster than slow-moving processes in HVAC systems. Both static and dynamic models can also be developed for a similar subsystem. Sometimes, dynamic physics-based models are presented by the thermal network method. In this method, heat transfer in HVAC components is often modeled by an electrical network in which the resistors and capacitors show the thermal resistance and the capacitor, respectively, while the current and voltage represent the heat and temperature transfer, respectively [26], [27].

5.1 HVAC Modeling Techniques

Major techniques are used for modeling HVAC systems such as:

- Frequency Domain Models [28],
- Data mining algorithm [29],
- Fuzzy Logic Models [30],
- Statistical models [31],
- State-Space [32],

- Geometric (Thin plate spline approximation) [33],
- Case-Based Reasoning [34],
- Stochastic models [35],
- Instantaneous models [36].

5.2 White box model (physics-based)

Physics-based models are also known as the main analytical models, forward models or white box models. These models are based on accurate knowledge of the process and its basic physical principles. They require considerable effort to develop and calibrate. Although physics-based models usually act as time domain differential equations, they can be easily converted to frequency domain TFs (Transfer Function) or time-domain SS (Steady State) representations.

In this project, the Simscape toolbox is used at the beginning. However, using the mathematical equations and modeling in Simulink has its own advantage like simplification, converting to SS model and so on. Thus, the Simulink is applied for modeling and simulation. The first version of HVAC modeling which is presented by Simscape can be seen in the following:

The temperature of a single zone using the mass parameter model in [37] by applying heat equilibrium in the zone air has presented equilibrium in the zone air as follows:

$$\rho_a V_z C_z \frac{dT_z}{dt} = m_{SA} C_{pa}(T_{SA} - T_z) + U_{wa} A_{wa}(T_{wa} - T_z) + U_{wd} A_{wd}(T_{wd} - T_z) + Q_l + Q_{int} \quad (18)$$

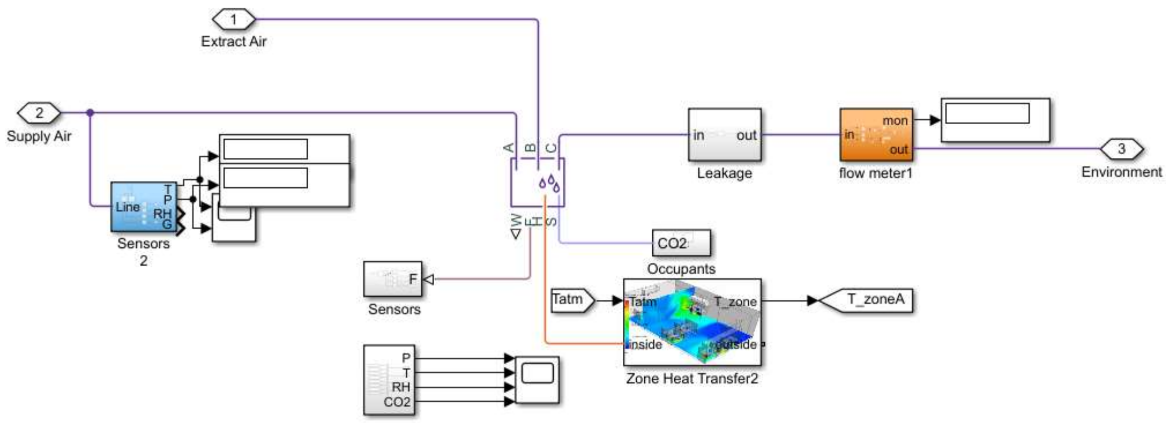


Figure 20. Zone model (Simscape model)

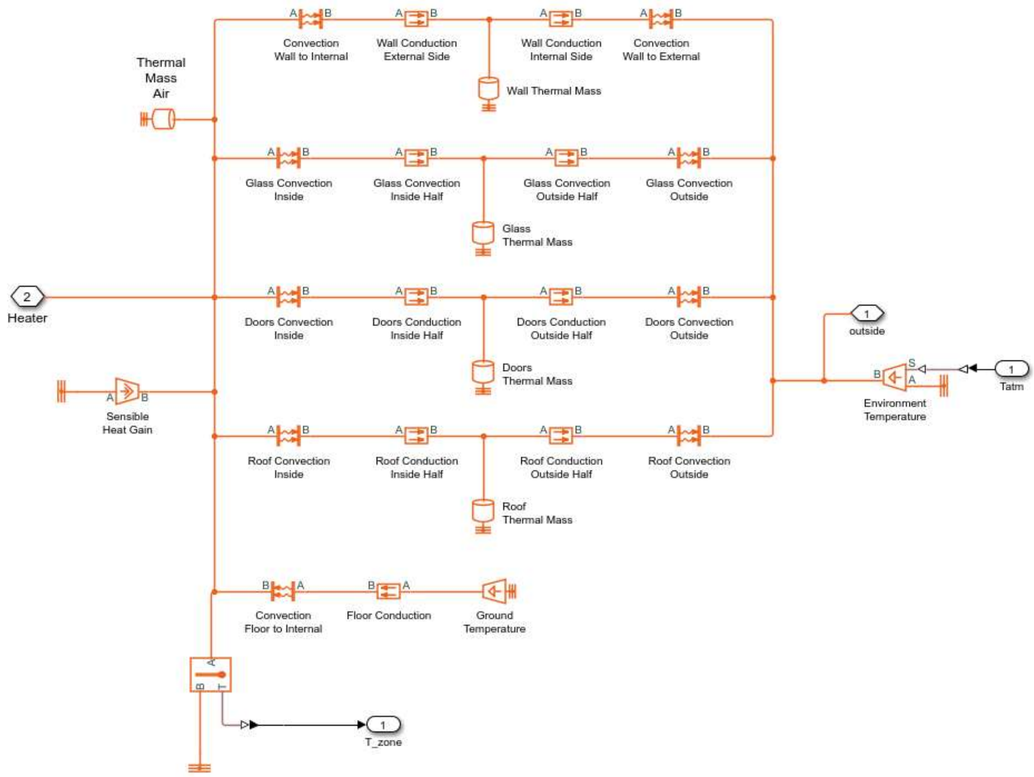


Figure 21. Heat transfer in Zone model (Simscape model)

The dynamic temperature model of the heating and cooling coil can be derived from the energy balance in the air and water heat exchanger. The mass balance on the side of the air gives the ratio of the moisture content of the outlet air. Outlet water and air temperatures are as follows [26]:

$$C_{wm}\frac{dT_{wo}}{dt} = m_w C_{pw}(T_{wi} - T_{wo}) - (UA)_{cc}(T_{wo} - T_{ao}) \tag{19}$$

$$C_{pm}\frac{dT_{wo}}{dt} = (UA)_{cc}(T_{wo} - T_{ao}) - m_a C_{pa}(T_{ao} - T_{ai}) \tag{20}$$

For simplification, the steady state model named effectiveness-NTU method can be applied which is usually use for cooling tower.

$$T_{hot,out} = T_{hot,in} - \frac{\dot{Q}}{(\dot{m}_{hot}\, C_{p,hot})} \tag{21}$$

$$T_{cold,out} = T_{cold,in} + \frac{\dot{Q}}{(\dot{m}_{cold}\, C_{p,cold})} \tag{22}$$

Where

$$\dot{Q} = \varepsilon_{NTU}\, \dot{m}\, C_{min}(T_{hot,in} - T_{cold,in}) \tag{23}$$

The supply duct model represents the heat transfer between the air inside the duct and the ambient [38] given as follows:

$$\frac{dT_{sdi}}{dt} = \frac{4U_{sd}(T_{amb} - T_{sdi})}{C_{sd}D_{sd}\rho_a} \tag{24}$$

The mixed air temperature is the linear combination of return air and fresh outside air temperature given as [39]

$$T_m = \frac{m_0 T_0 + (m_{SA} - m_0) T_z}{m_{SA}} \tag{25}$$

By substituting all temperature variables with the variables of CO_2 concentration or the relative humidity ratio in the above equation, a similar linear relationship can be found to the CO_2 concentration of the mixed air or humidity ratio.

In the damper model presented in [30], the mass flow rate of air through the damper depends on the damper flow coefficient, the pressure difference across the damper, and the location-dependent flow cross-sectional area of the damper. It is assume that control signal is between $u(t) = [0 \quad 1]$

$$m_a = C_{dp} \sqrt{\rho_a \Delta P_{dp}} A_{dp}(\emptyset) \tag{26}$$

Which $A_{dp}(\emptyset)$is changed based on $u(t)$

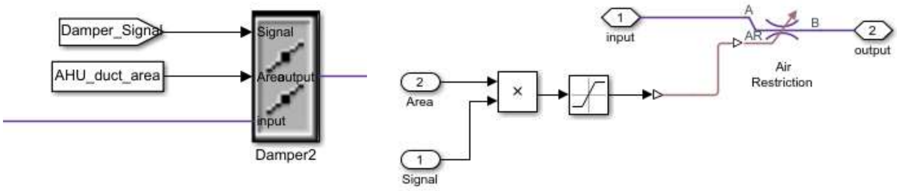


Figure 22. Damper model (Simscape model)

The valve opening depends upon the control signal and the valve authority [40]. The power consumption of the fan or pump depends on the flow rate, pressure difference between inlet and outlet and efficiency of the fan or pump [41]. The temperature of the water in boiler is given with heat supplied and difference in supply temperature [42].

$$\frac{dT_{boiler}}{dt} = \frac{1}{m_w C_{pw}} \left[Q_{heat} + m_w C_{pw} (T_{wi} - T_{wo}) \right] \tag{27}$$

5.3 Simulation

Whole system

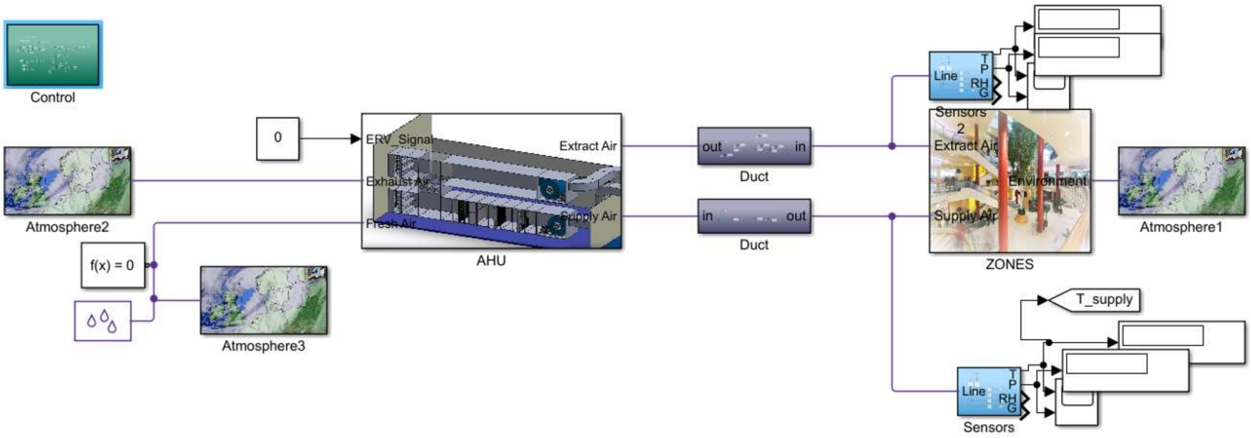


Figure 23. overview of whole system (Simulink and Simscape model)

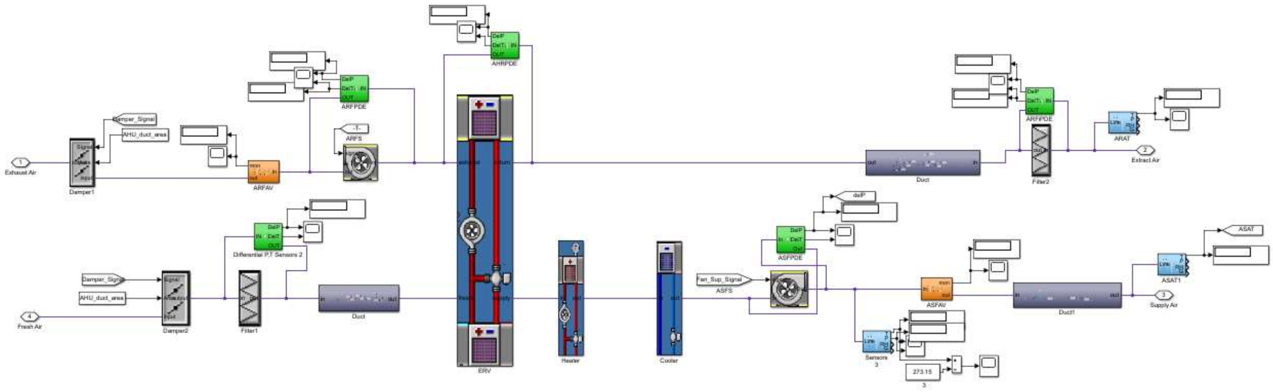


Figure 24. AHU model (Simulink and Simscape model)

Heater

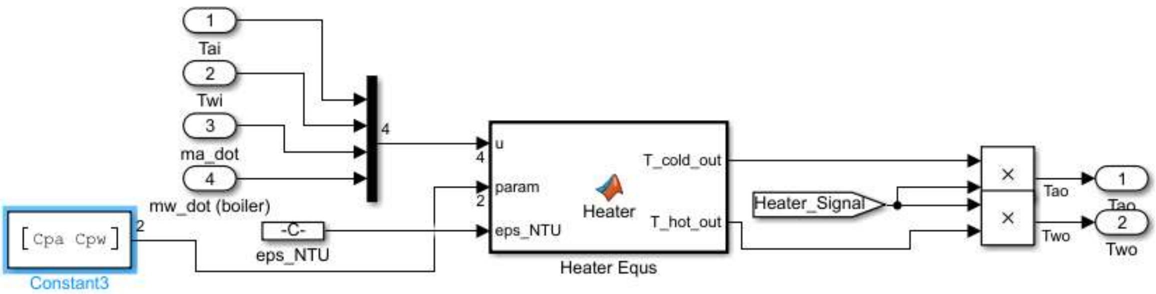


Figure 25. Heater model (simulink model)

Using equations (21)-(23) in function Heater, see Appendix 0, the results provided in Figure 26 were obtained.

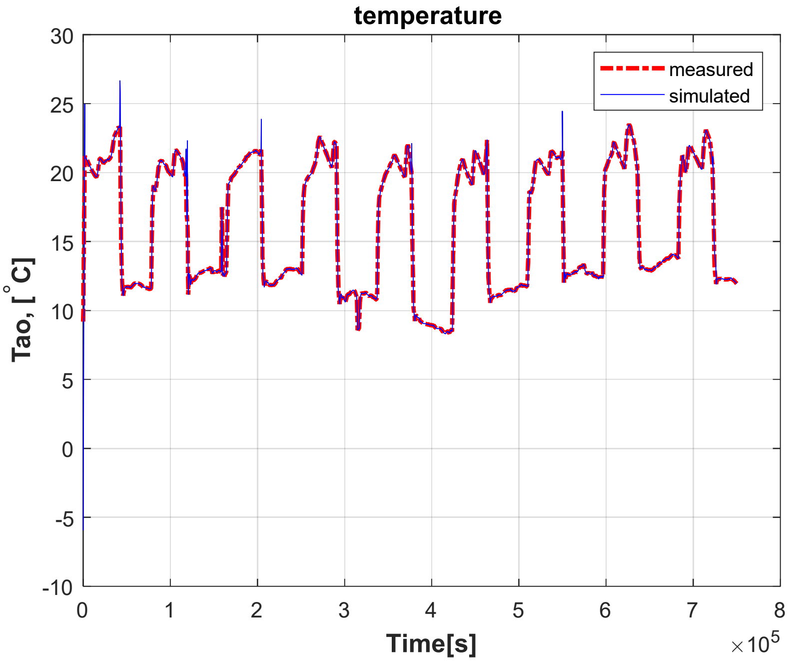


Figure 26. Air temperature after heater

Air fan control

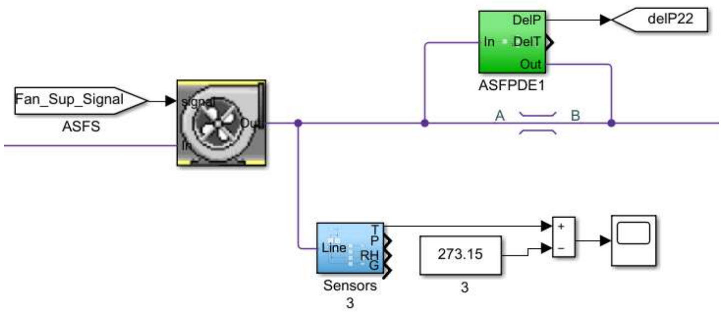


Figure 27. Air fan model in Simscape

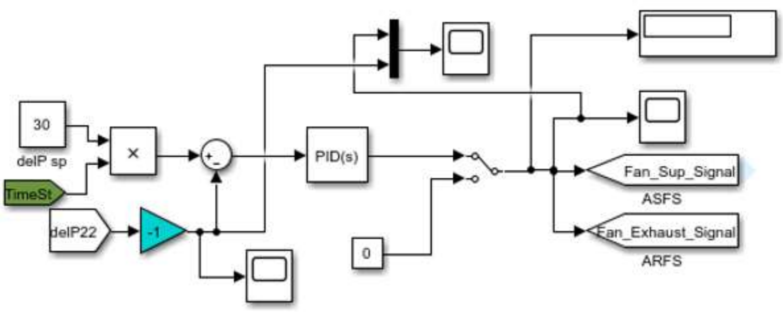


Figure 28. Air fan control model (Simulink model)

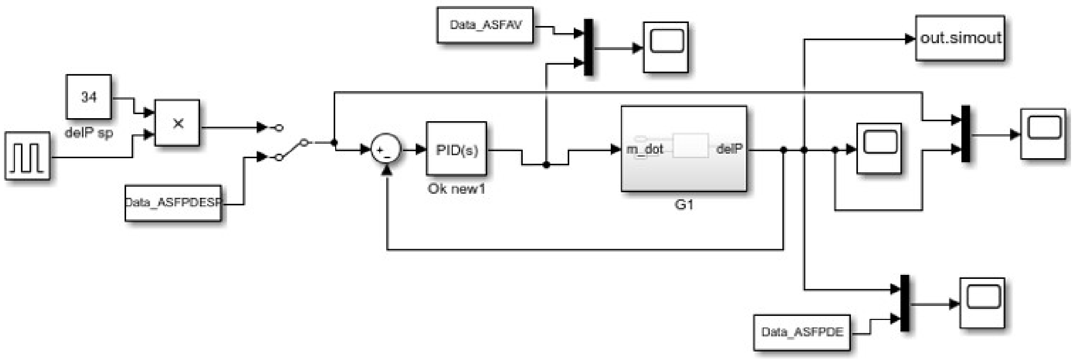


Figure 29. Air fan validation model in Simulink

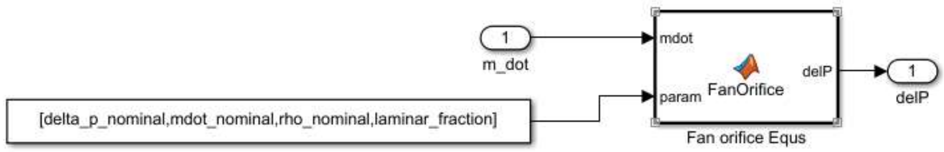


Figure 30. Air fan sub-model in Simulink

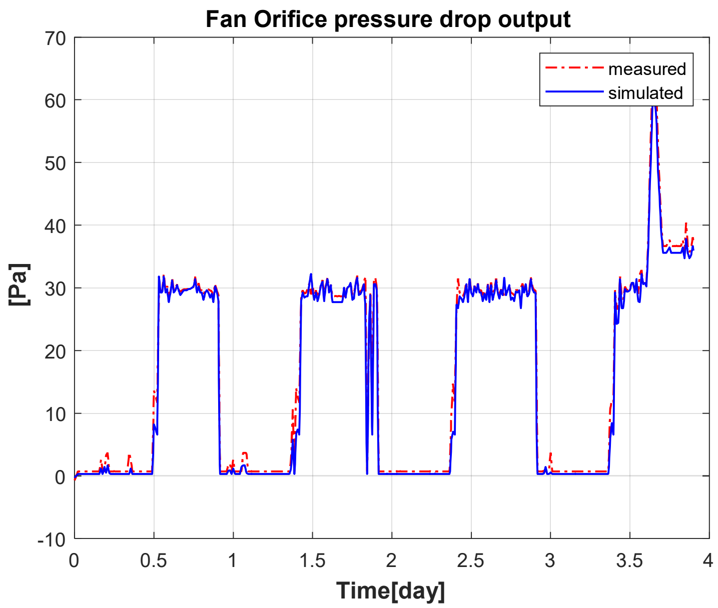


Figure 31. Fan orifice pressure drop, open loop simulation, input: ASFAV, output: ASFPDE

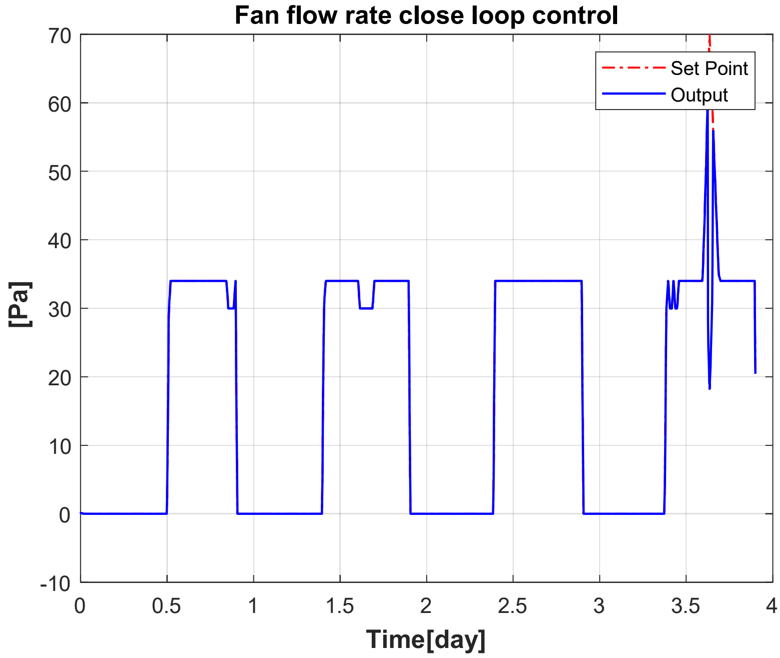


Figure 32. Close loop simulation, input: Setpoint ASFPDESP, output: ASFPDE

Heat Recovery Ventilator (HRV)

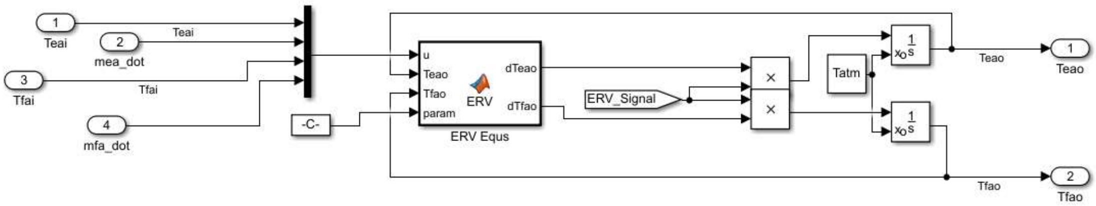


Figure 33. Heat Recovery Ventilator simulink model (HRV)

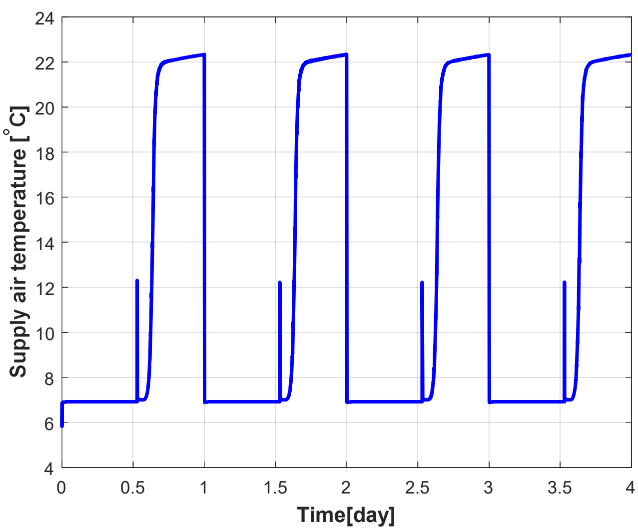


Figure 34. Simulation output, air zone supply air temperature

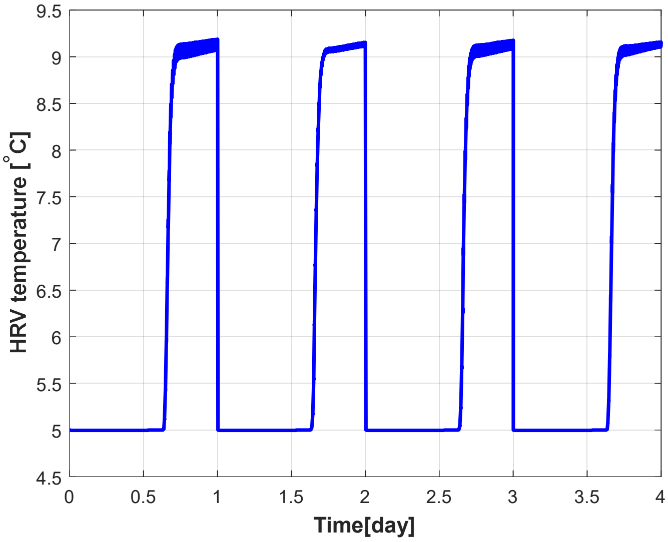


Figure 35. Simulation output, energy recovery ventilator temperature

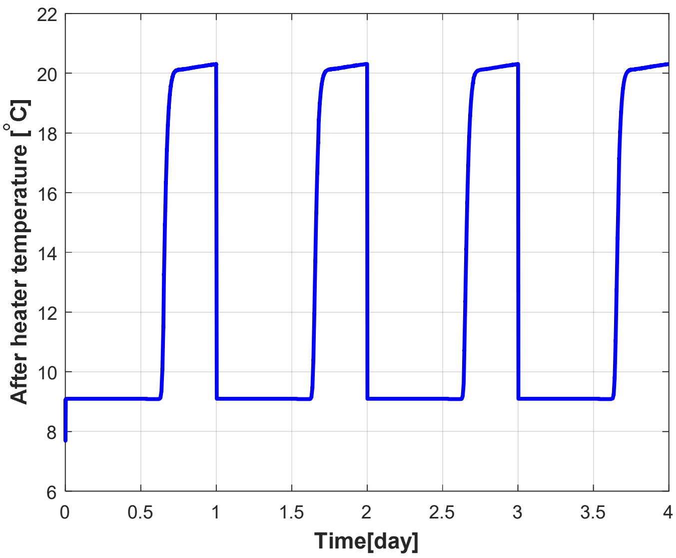


Figure 36. Simulation output, duct temperature after heater coil

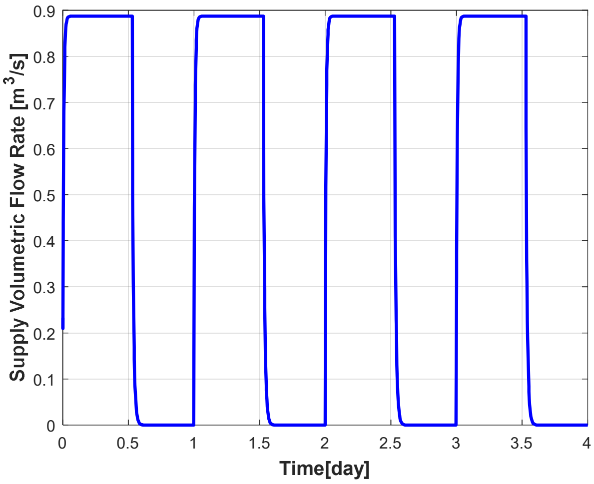


Figure 37. Simulation output, supply air duct flow rate

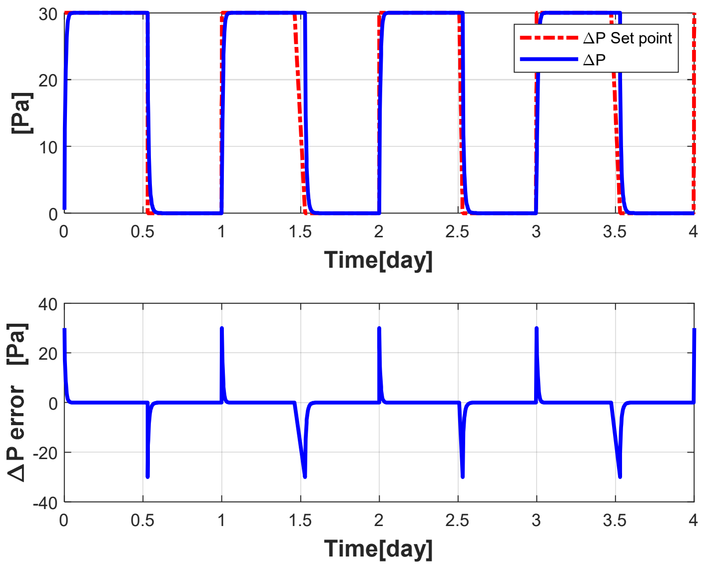


Figure 38. Simulation output, fan control using pressure drop of fan orifice

The simulation is based on the 333 data unit which is on/off type of HVAC system. From Figure 24 to Figure 30, some parts of modeling in Simulink, Simscape and mfile programming environment is shown. Figure 31 shows the open loop output of pressure sensor located after fan or induction motor. The main factor in order to control the flow rate in duct is relative pressure drop located after fan. In fact, the set point to control the flow rate is ΔP. In Figure 32, the PID controller is applied to control the pressure drop after fan (ASFPDE) in order to reach to desire flow rate. The set point (ASFPDESP,) imported from experimental data and controller is applied in modeling environment. Figure 32 also shows the acceptable performance using PID controller to control the fan. Figure 34 shows the temperature simulation output which supplies to the building or zones. The SP temperature command is sent from the BAS and the coupled HVAC system works in such a way that the desired temperature is sent to the zones. As shown in Figure 35, the additional temperature or energy present in the return duct channel is applied to the supply duct, which reduces the energy consumption of the AHU system. Therefore, the return temperature, which is about 9^oC, enters the heat exchanger and the output temperature of the exchanger reaches 20^oC degrees (Figure 36). It should be noted that this graph depends on the AHCVO signal. This is controlled by the AHCSWst command and applies the desired heater temperature to the AHU system. The air supply temperature delivered to the zones is completely dependent on the flow rate. Thus, the BAS calculates the appropriate signal considering the temperatures as well as flow rate. In other words, control commands are the result of calculating a set of processes and the controller considers other coupled components to generate appropriate command (Figure 37). Figure 38 shows that the flow rate is controlled by a separate cascade control system with the help of a pressure sensor. In this figure, ΔP and closed loop controller error are observed. Due to the lack of sufficient information from BAS supervisory control, the separate controller was designed and applied in modeling.

The purpose of this modeling and simulation is to identify the system and examine methods in simulation to reduce energy consumption. It is also helpful to compare standard simulated model with measured data in different units for fault detection.

It should be noted that this thesis is part of the project in Centre for Intelligent Systems and Nearly Zero Energy Buildings research group in TalTech and this section is in progress.

5.4 Recommendations for future work

The aim of this thesis was to

- Development of simulation framework for HVAC system,
- Development of new software package based on intelligent search algorithm,
- Implementation of the method on Tallinn and Helsinki sites,
- Prediction of clogging in filters,
- Estimate the physical parameters in the physic-based dynamic models.

Among studied methods GA, PSO, ABC and their hybridization were selected and applied and simulated output was compared to experimental output.

Experiments showed by using developed package software, it is possible to find multi unknown parameters in MISO modeling. As a case study, multi inputs were dynamic of temperature and volumetric flow rate. Best method is PSOABC which has high correlation, less RMSE and best tracking. The filter clogging is predicted in the presence of other dynamics like measured temperature and measured flow rate. Using long time duration data brings the accuracy of the prediction.

Despite considerable work on unknown parameter estimation, possible areas that require further investigation still exist and are summarized as follows:

- Performance comparison of different cost function metrics,
- Study of other intelligent search algorithms such as SA, DE and GSO,
- Develop the code as MIMO.

Improving the HVAC model and validation with experimental data including:

- Heat and cool coil analysis, identification and modeling,
- Fouling in heat coil section, its effect in power consumption and prediction for periodic cleaning,
- Behavior of damper and modeling the CO_2 control,
- Black-box and gray-box modeling of HVAC system and components,
- Study on factors that affect the efficiency and power consume,
- HVAC control systems.

Summary

The residential HVAC systems can consume more than 60% of the total energy in a house which results in higher operating costs and environmental pollution. The HVAC is a complex system with variable loads caused by the changes in weather and occupancy. The energy consumption of the HVAC systems can be reduced by adapting to the ever changing loads and implementation of energy conservation strategies along with the appropriate maintenance or overhaul service. Almost all HVAC systems, utilize the filter to provide clean and fresh air to the zones. Having an effective and accessible models of the performance of these systems could be important tools for filter designers, building designers and building energy managers as well as those who are attempting to optimize building energy performance through the use of dynamic model-based control systems.

In this thesis, the parameters based on physical based dynamic model by intelligent optimization methods is estimated. The filter clogging in HVAC system on Tallinn and Helsinki sites is estimated and predicted based on this research method. GA, PSO, ABC and hybridization optimization algorithms were used to estimate the parameters of a moist air flow. The estimated parameters were obtained using real measurements. Different HVAC system data from building sites of Tallinn and Helsinki is used which the results have shown the ABC and PSO-ABC have good performance for estimating the unknown parameters of filters in HVAC systems. Both polynomial and exponential patterns can be used for prediction of filters clogging in some finite range. For more precise and longer period predictions engineers need more data without any faults in the systems.

Kokkuvõte

Elamute HVAC süsteemid võivad tarbida üle 60% maja koguenergiast, mille tulemuseks on suuremad tegevuskulud ja keskkonnareostus. HVAC on keeruline süsteem, mille muutuvad koormused mõjutavad seda ilmastiku ja täituvuse muutuste tõttu. HVAC-süsteemide energiatarbimist saab vähendada pidevalt muutuvate koormuste kahanemisega ja energiasäästu strateegiate rakendamisega (s.h. hooldus- või kapitaalremonditeenustega). Peaaegu kõikides HVAC süsteemides kasutatakse filtreid, et tagada tsoonidele puhas ja värske õhk. Selliste süsteemide täpsemad ja efektiivsed mudelid on oluline vahend filtrite disaineritele, hoonete projekteerijatele ning ka neile, kes optimeerivad hoonete energiatõhusust dünaamiliste mudelipõhiste juhtimissüsteemide abil.

Selles lõputöös vaadeldatakse erinevaid intelligentseid optimeerimis meetodeid ja leiakse füüsikaliste dünaamiliste mudelite parameetreid. Hinnatakse ja ennustatakse filtrite ummistumist HVAC süsteemis Tallinna ja Helsingi. GA, PSO, ABC ja hübriidoptimeerimise algoritme kasutati õhuvoolu parameetrite hindamiseks. Hinnangulised parameetrid saadi reaalsete mõõtmiste abil. Kasutatakse erinevate HVAC süsteemide andmeid Tallinna ja Helsingi, kus on nii ON-OFF kui ka pideva juhtimisega süsteemid. Tulemused on näidanud, et ABC ja PSO-ABC algoritmite saab kasutadaH HVAC süsteemide filtrite tundmatute parameetrite hindamisel. Nii polünomiaalsed kui ka eksponentsiaalsed mudeleid saab kasutada filtrite ummistuse prognoseerimiseks. Selleks et garanteerida pikemat prognoositäpsust, on vaja rohkem andmeid kus ei esine süsteemi vigu.

References

[1] P. Ghamisi and J. A. Benediktsson, "Feature selection based on hybridization of genetic algorithm and particle swarm optimization," *IEEE Geosci. Remote Sens. Lett.*, vol. 12, no. 2, pp. 309–313, 2014.

[2] A. A. Nik, F. M. Nejad, and H. Zakeri, "Hybrid Pso And Ga Approach For Optimizing Surveyed Asphalt Pavement Inspection Units In Massive Network," *Autom. Constr.*, vol. 71, pp. 325–345, 2016.

[3] J. Robinson, S. Sinton, and Y. Rahmat-Samii, "Particle swarm, genetic algorithm, and their hybrids: optimization of a profiled corrugated horn antenna," *IEEE Antennas Propag. Soc. Int. Symp.*, vol. 1, pp. 314–317, 2002.

[4] C. Li, R. Zhai, H. Liu, Y. Yang, and H. Wu, "Optimization of a heliostat field layout using hybrid PSO-GA algorithm," *Appl. Therm. Eng.*, vol. 128, pp. 33–41, 2018.

[5] T. Hendtlass, "A combined swarm differential evolution algorithm for optimization problems," *Int. Conf. Ind. Eng. Other Appl. Appl. Intell. Syst.*, pp. 11–18, 2001.

[6] M. Seyedmahmoudian *et al.*, "Simulation and hardware implementation of new maximum power point tracking technique for partially shaded PV system using hybrid DEPSO method," *IEEE Trans. Sustain. energy*, vol. 6, no. 3, pp. 850–862, 2015.

[7] P. Boonserm and S. Sitjongsataporn, "A robust and efficient algorithm for numerical optimization problem: DEPSO-Scout: A new hybrid algorithm based on DEPSO and ABC," *Int. Electr. Eng. Congr.*, pp. 1–4, 2017.

[8] F. Zhao, Q. Zhang, D. Yu, X. Chen, and Y. Yang, "A hybrid algorithm based on PSO and simulated annealing and its applications for partner selection in virtual enterprise," *Adv. Intell. Comput.*, pp. 380–389, 2005.

[9] S. Sudibyo, M. N. Murat, and N. Aziz, "Simulated annealing-Particle Swarm Optimization (SA-PSO): particle distribution study and application in Neural Wiener-based NMPC," *10th Asian Control Conf.*, pp. 1–6, 2015.

[10] F. Javidrad and M. Nazari, "A new hybrid particle swarm and simulated annealing stochastic optimization method," *Appl. Soft Comput.*, vol. 60, pp. 634–654, 2017.

[11] P. Li, N. Cui, Z. Kong, and C. Zhang, "Energy management of a parallel plug-in hybrid electric vehicle based on SA-PSO algorithm," *36th Chinese Control Conf.*, pp. 9220–9225, 2017.

[12] M. Mandloi and V. Bhatia, "A low-complexity hybrid algorithm based on particle swarm and ant colony optimization for large-MIMO detection," *Expert Syst. Appl.*, vol. 50, pp. 66–74, 2016.

[13] A. Colorni, M. Dorigo, V. Maniezzo, and others, "Distributed optimization by ant

colonies," *Proc. first Eur. Conf. Artif. life*, vol. 142, pp. 134–142, 1992.
[14] P. S. Shelokar, P. Siarry, V. K. Jayaraman, and B. D. Kulkarni, "Particle swarm and ant colony algorithms hybridized for improved continuous optimization," *Appl. Math. Comput.*, vol. 188, no. 1, pp. 129–142, 2007.
[15] L. Junliang and others, "A Hybrid Algorithm Based on Particle Swarm Optimization and Ant Colony Optimization Algorithm, Smart Computing and Communication: First International Conference, SmartCom 2016, Shenzhen, China, December 17-19, 2016, Proceedings, 2017."
[16] A. Ghodrati and S. Lotfi, "A hybrid cs/ga algorithm for global optimization," *Proc. Int. Conf. Soft Comput. Probl. Solving*, pp. 397–404, 2012.
[17] J. Guo *et al.*, "Hybrid optimization algorithm of particle swarm optimization and cuckoo search for preventive maintenance period optimization," *Discret. Dyn. Nat. Soc.*, vol. 2016, 2016.
[18] R. Chi, Y. Su, D. Zhang, X. Chi, and H. Zhang, "A hybridization of cuckoo search and particle swarm optimization for solving optimization problems," *Neural Comput. Appl.*, vol. 31, no. 1, pp. 653–670, 2019.
[19] Y. Gao, "An improved hybrid group intelligent algorithm based on artificial bee colony and particle swarm optimization," in *Proceedings - 2018 International Conference on Virtual Reality and Intelligent Systems, ICVRIS 2018*, 2018, vol. 5, pp. 160–163.
[20] F. Zhou and Y. Yang, "An Improved Artificial Bee Colony Algorithm Based on Particle Swarm Optimization and Differential Evolution," *Int. Conf. Intell. Comput.*, pp. 24–35, 2015.
[21] D. Sedighizadeh and H. Mazaheripour, "Optimization of multi objective vehicle routing problem using a new hybrid algorithm based on particle swarm optimization and artificial bee colony algorithm considering Precedence constraints," *Alexandria Eng. J.*, 2017.
[22] S. Wang and Z. Ma, "Supervisory and optimal control of building HVAC systems: A review," *HVAC R Res.*, vol. 14, no. 1, pp. 3–32, 2008.
[23] A. Sporr, G. Zucker, and R. Hofmann, "Automated HVAC Control Creation Based on Building Information Modeling (BIM): Ventilation System," *IEEE Access*, vol. 7, pp. 74747–74758, 2019.
[24] D. Karaboga, "An idea based on Honey Bee Swarm for Numerical Optimization," *Tech. Rep. TR06, Erciyes Univ.*, 2005.
[25] I. E. Idelchik, "Handbook of hydraulic resistance (2nd revised and enlarged edition)," *Washington*, 1986.
[26] B. Tashtoush, M. Molhim, and M. Al-Rousan, "Dynamic model of an HVAC system for control analysis," *Energy*, 2005.

[27] A. Afram and F. Janabi-Sharifi, "Review of modeling methods for HVAC systems," *Applied Thermal Engineering*. 2014.

[28] Q. Bi *et al.*, "Advanced controller auto-tuning and its application in HVAC systems," *Control Eng. Pract.*, 2000.

[29] A. Kusiak and G. Xu, "Modeling and optimization of HVAC systems using a dynamic neural network," *Energy*, 2012.

[30] S. Soyguder and H. Alli, "Predicting of fan speed for energy saving in HVAC system based on adaptive network based fuzzy inference system," *Expert Syst. Appl.*, 2009.

[31] J. Ma, S. J. Qin, B. Li, and T. Salsbury, "Economic model predictive control for building energy systems," in *IEEE PES Innovative Smart Grid Technologies Conference Europe, ISGT Europe*, 2011.

[32] L. Ferkl and Jan Široký, "Ceiling radiant cooling: Comparison of ARMAX and subspace identification modelling methods," *Build. Environ.*, 2010.

[33] J. Nishiguchi, T. Konda, and R. Dazai, "Data-driven optimal control for building energy conservation," in *Proceedings of the SICE Annual Conference*, 2010.

[34] I. Watson, "A case study of maintenance of a commercially fielded case-based reasoning system," *Comput. Intell.*, 2001.

[35] I. Zlatanović, K. Gligorević, S. Ivanović, and N. Rudonja, "Energy-saving estimation model for hypermarket HVAC systems applications," *Energy Build.*, 2011.

[36] Y. Iino, M. Murai, D. Murayama, I. Motoyama, S. Kuzusaka, and K. Ueta, "Hybrid modeling with physical and JIT model for building thermal load prediction and optimal energy saving control," in *ICCAS-SICE 2009 - ICROS-SICE International Joint Conference 2009, Proceedings*, 2009.

[37] X. Xu, S. Wang, and G. Huang, "Robust MPC for temperature control of air-conditioning systems concerning on constraints and multitype uncertainties," *Build. Serv. Eng. Res. Technol.*, 2010.

[38] A. Thosar, A. Patra, and S. Bhattacharyya, "Feedback linearization based control of a variable air volume air conditioning system for cooling applications," *ISA Trans.*, 2008.

[39] M. Mossolly, K. Ghali, and N. Ghaddar, "Optimal control strategy for a multi-zone air conditioning system using a genetic algorithm," *Energy*, 2009.

[40] I. Zajic, T. Larkowski, M. Sumislawska, K. J. Burnham, and D. Hill, "Modelling of an air handling unit: A Hammerstein-bilinear model identification approach," in *Proceedings - ICSEng 2011: International Conference on Systems Engineering*, 2011.

[41] A. P. Wemhoff and M. V. Frank, "Predictions of energy savings in HVAC systems

by lumped models," *Energy Build.*, 2010.

[42] S. Burhenne, J. Radon, M. Pazold, S. Herkel, and F. Antretter, "Integration of HVAC models into a hygrothermal whole building simulation tool," in *Proceedings of Building Simulation 2011: 12th Conference of International Building Performance Simulation Association*, 2011.

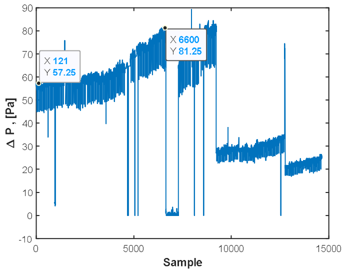
Appendix

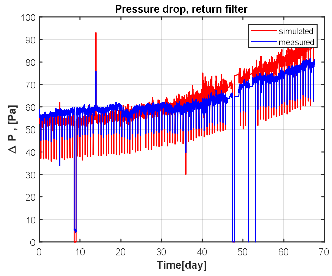
Other units result

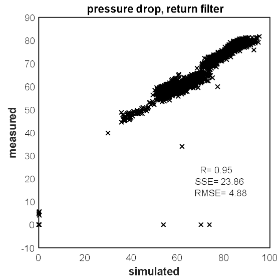
Units 1740,1741,1742,1743,1744,1745

Start 7/2019

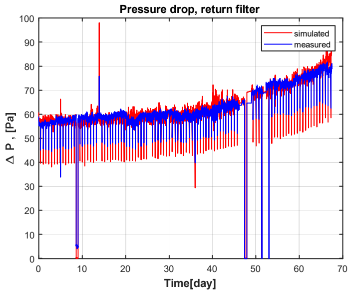
1740

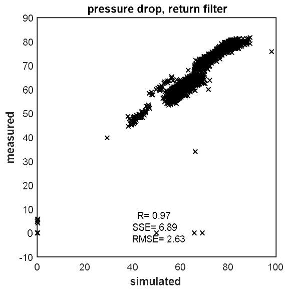






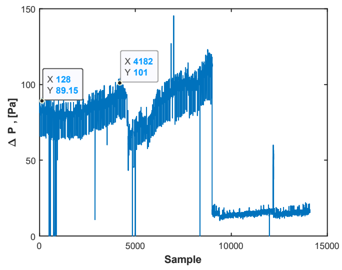
K1 = 1.4982e-13, K2 = 1.1279e-20, K3 = 7.6221

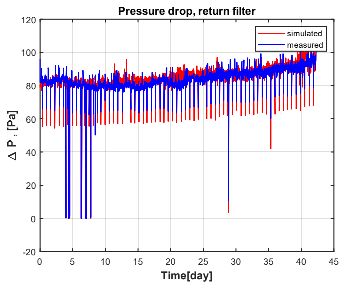


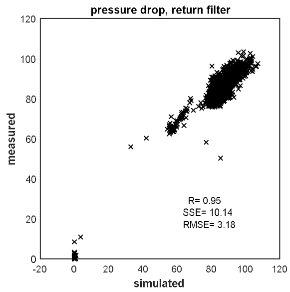


L1 = 0.16315 , L2 = 5.4651e-07, L3 = 8.1607

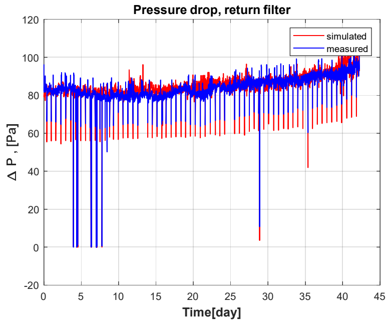
1741

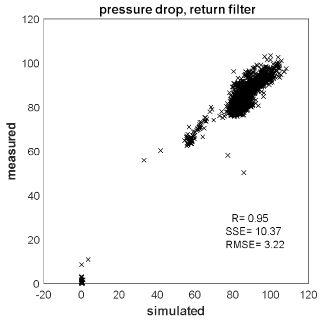






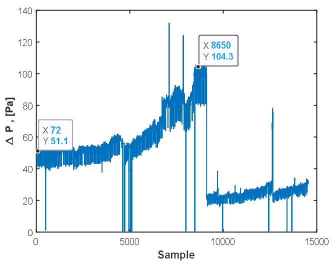
K1 = 1.6804e-13, K2 = 4.9267e-18, K3 = 11.721

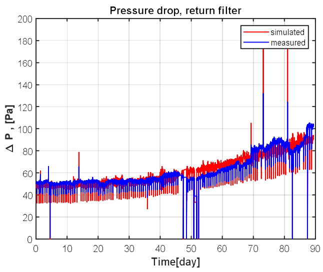


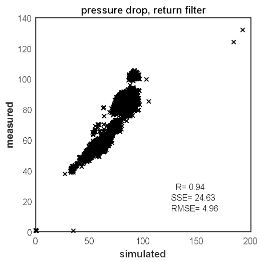


L1 = 0.19125, L2 = 7.1593e-07, L3 = 11.541

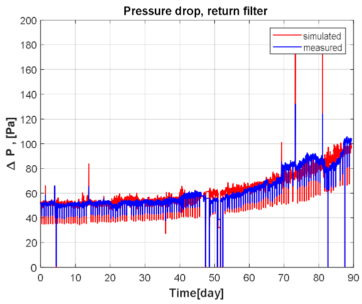
1742

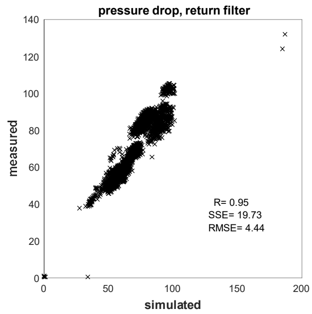






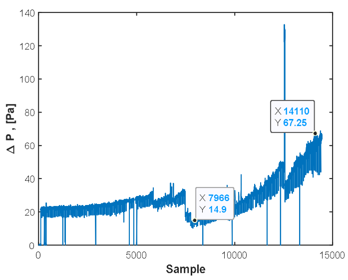
K1 = 1.0771e-13, K2 = 8.0959e-21, K3 = 6.9975

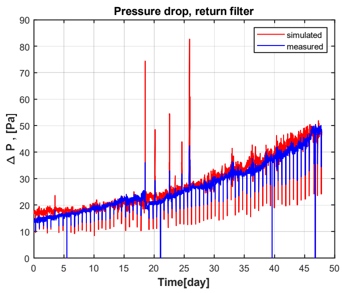


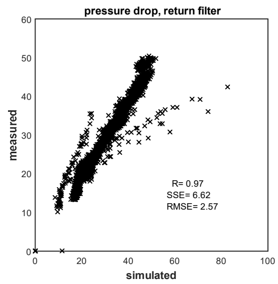


L1 = 0.20098, L2 = 4.6134e-07, L3 = 7.0652

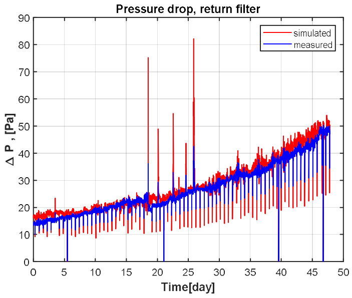
1743

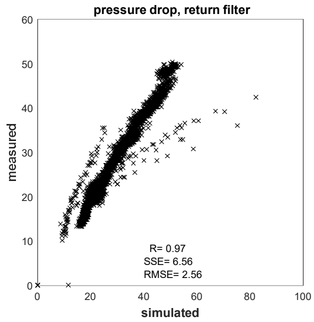






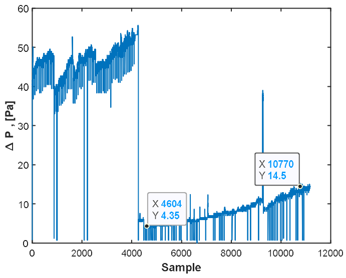
K1 = 2.5828e-13, K2 = 1.9564e-20, K3 = 2.335

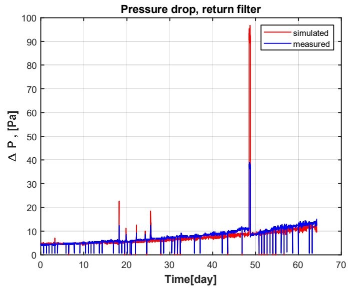


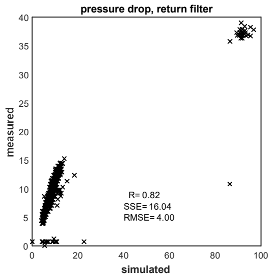


L1 = 0.58915,   L2 = 5.3708e-07,   L3 = 1.644

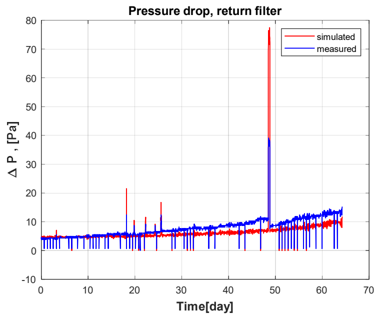
1744

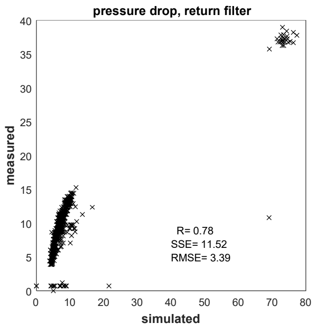






K1 = 3.3533e-14,   K2 = 1.9012e-20,   K3 = 0.48454



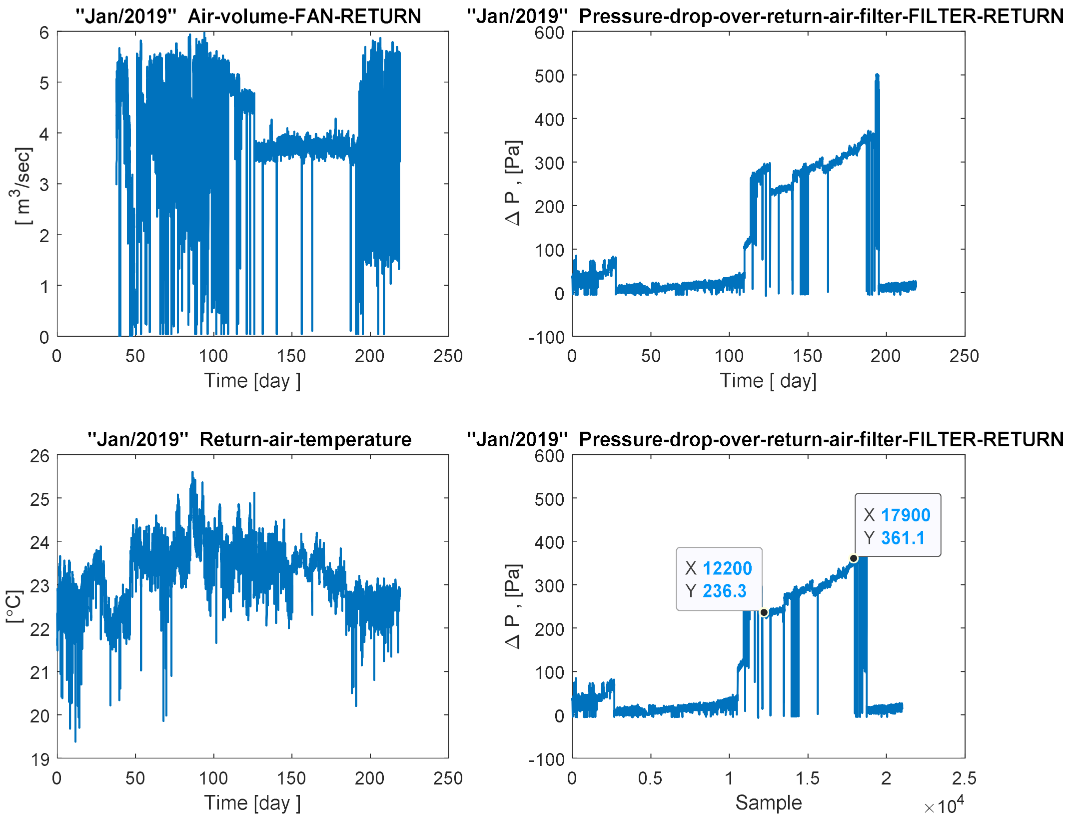


L1 = 0.036141,   L2 = 5.5991e-07,   L3 = 0.63698

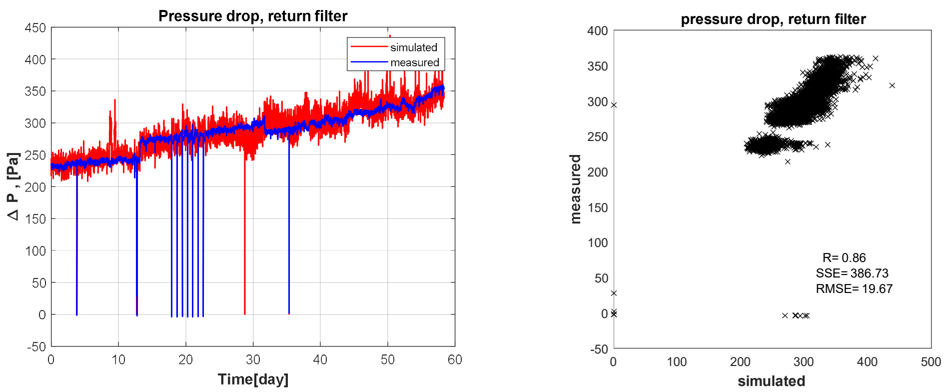
Units 467, 469, 471,474,479

Start January 2019

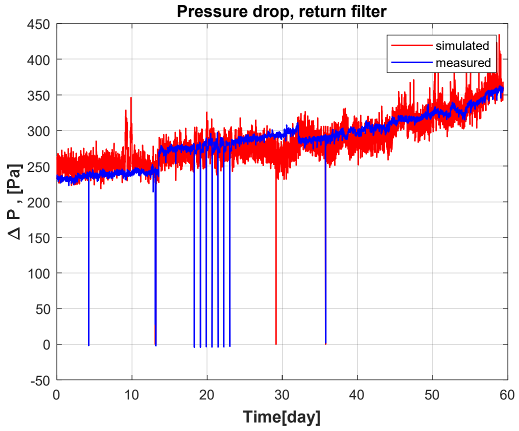
467

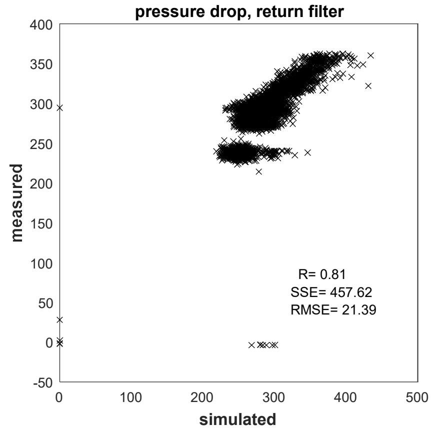


K1 = 1.9185e-14,   K2 = 5.2584e-09,   K3 = 0.35383



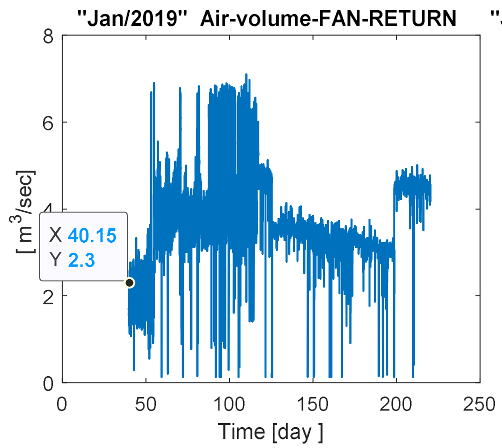
K1 = 8.8444e-14,   K2 = 8.9899e-07,   K3 = 15.959

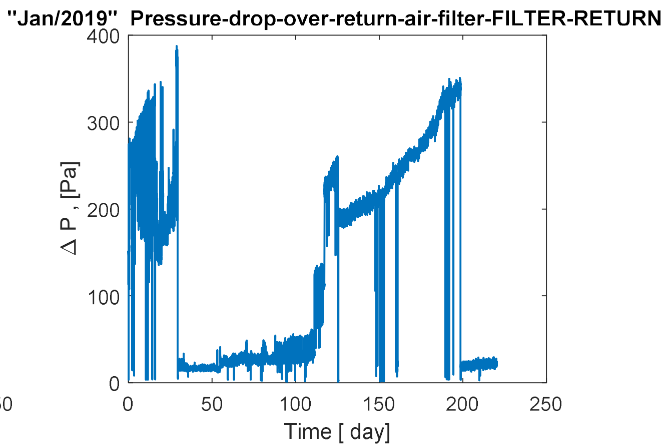


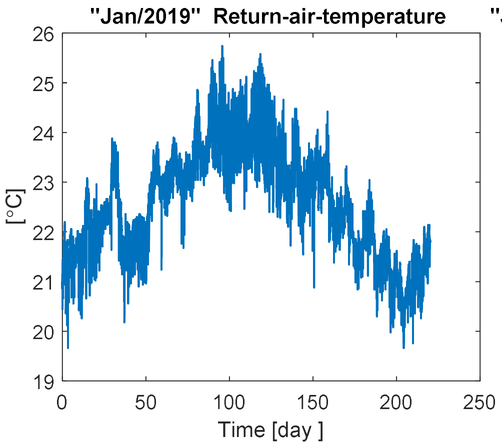


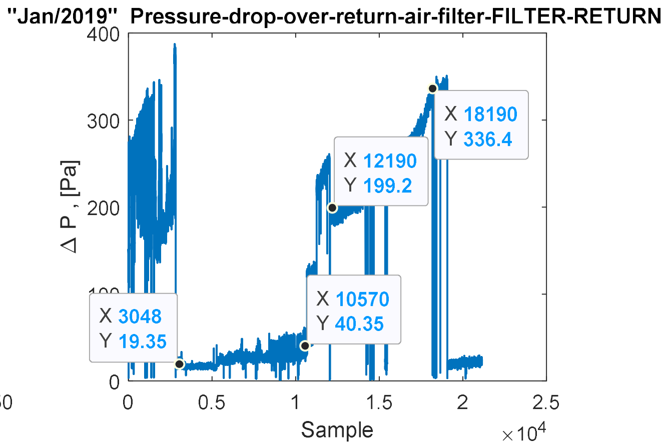
L1 = 0.18306,   L2 = 7.1988e-07,   L3 = 17.131

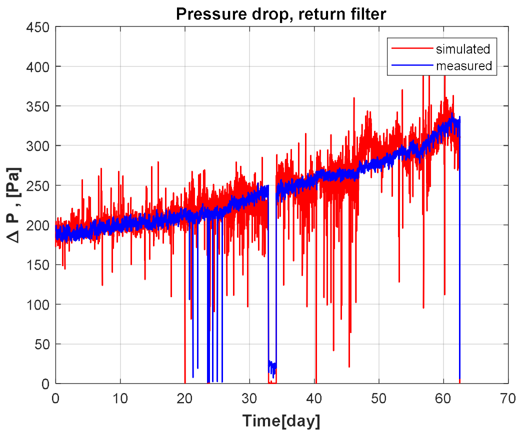
469

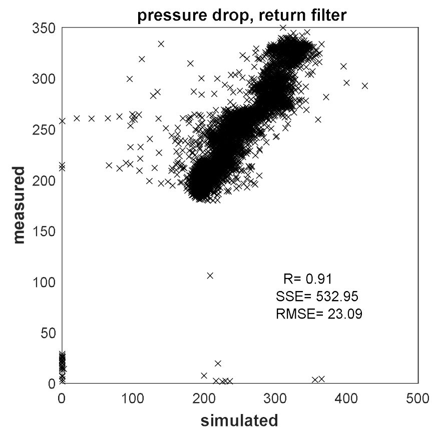




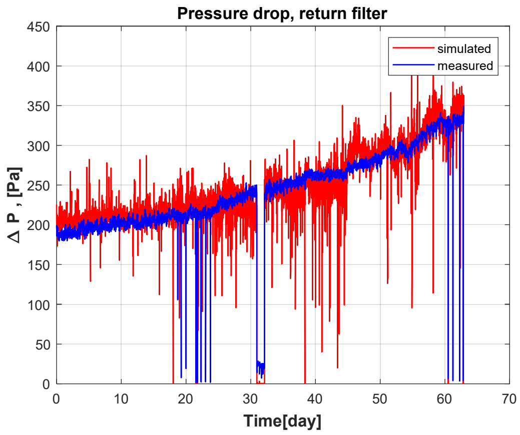


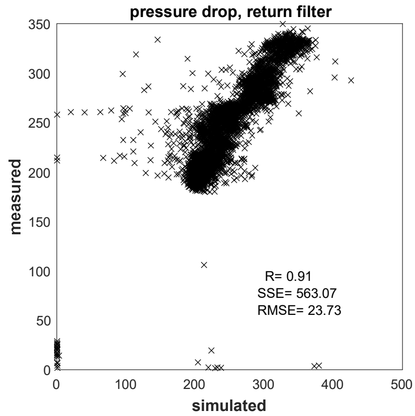






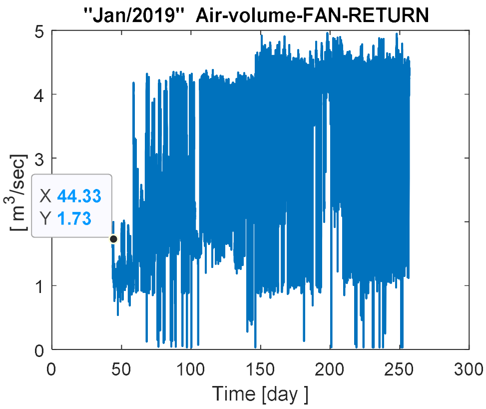
K1 = 4.9701e-13, K2 = 3.0123e-07, K3 = 13.914

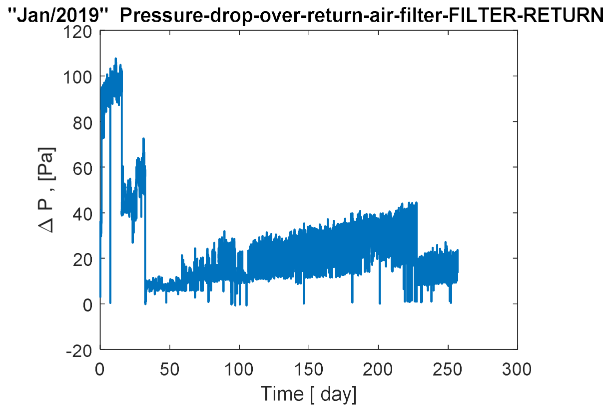


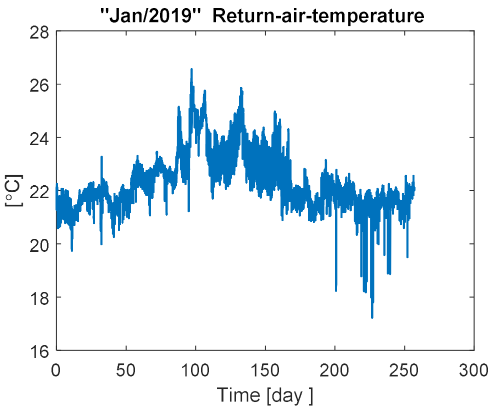


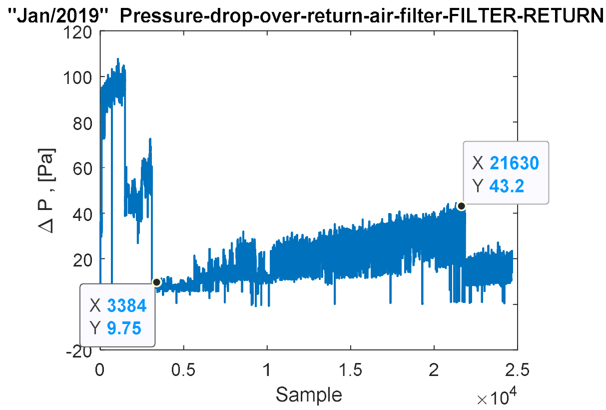
L1 = 1.5794, L2 = 4.6837e-07, L3 = 12.93

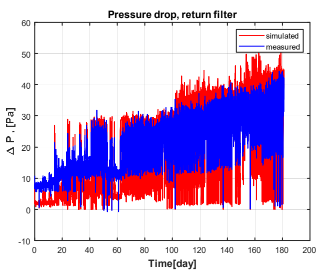
471

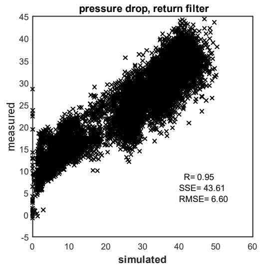


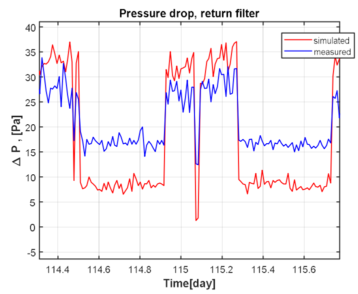




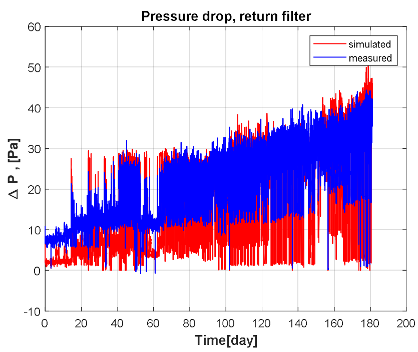


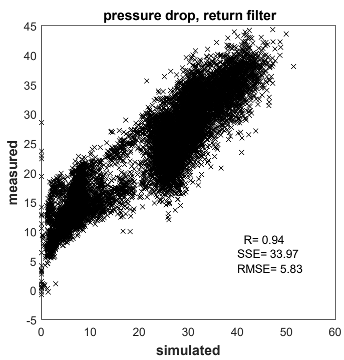






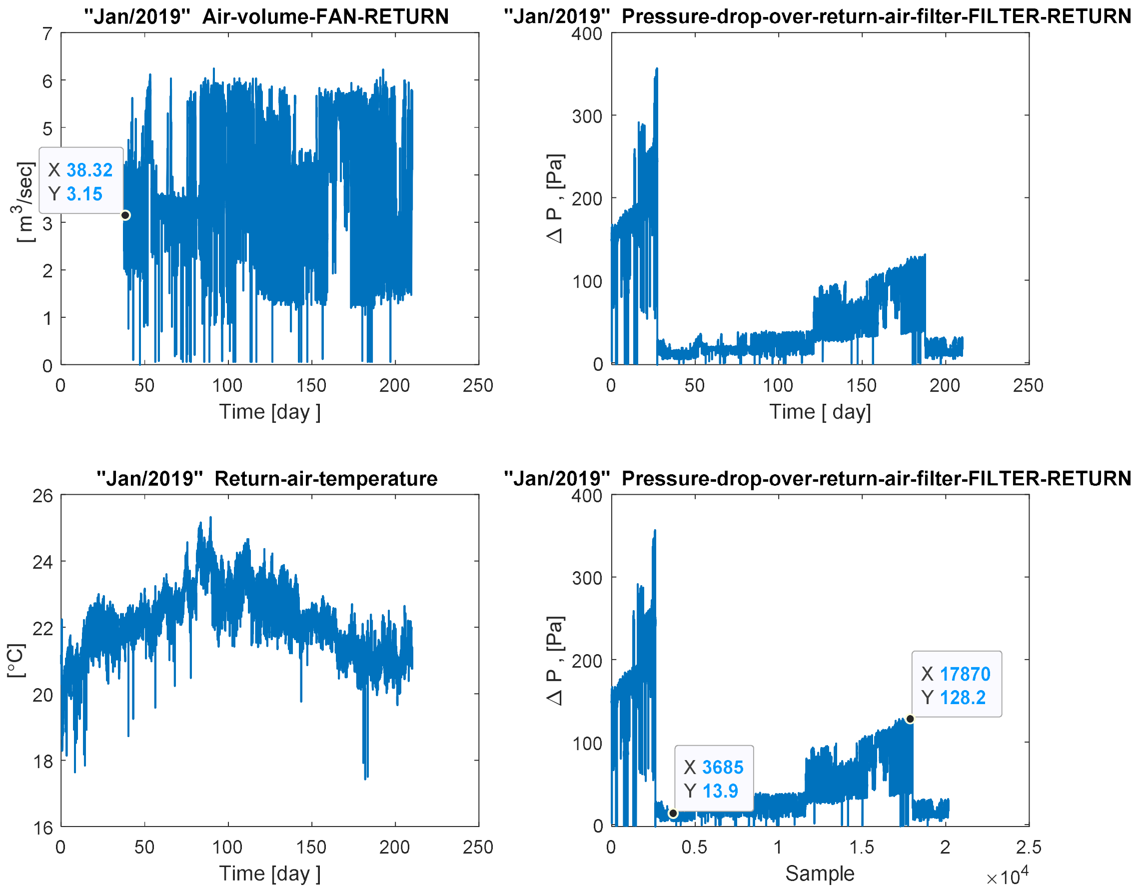
K1 = 1.6705e-15, K2 = 3.3374e-10, K3 = 1.3081

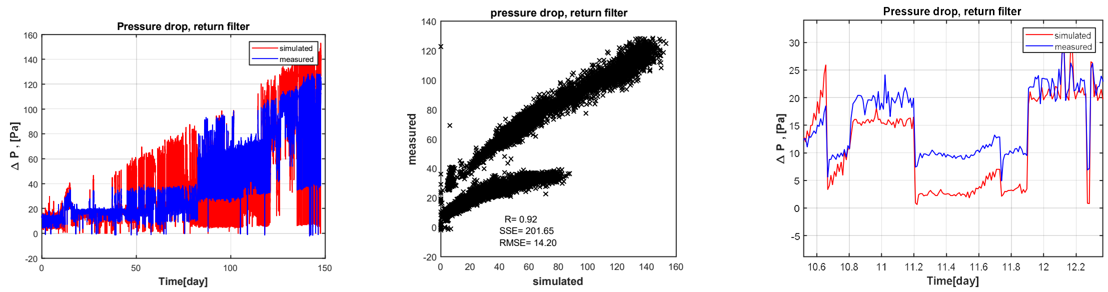




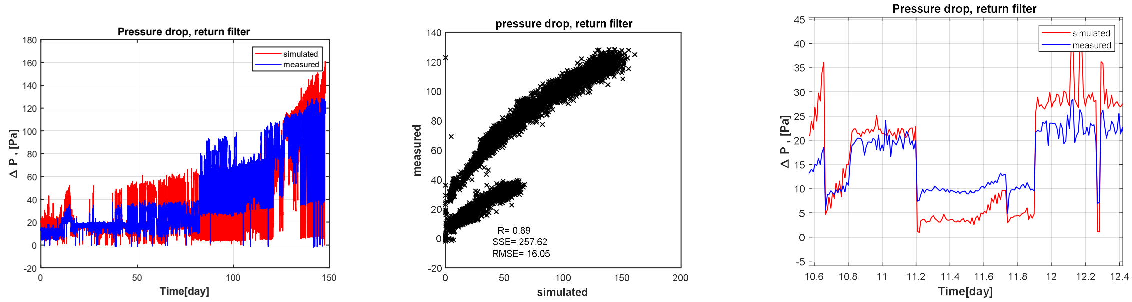
L1 = 1.7813e-05, L2 = 6.7614e-07, L3 = 1.4364

474





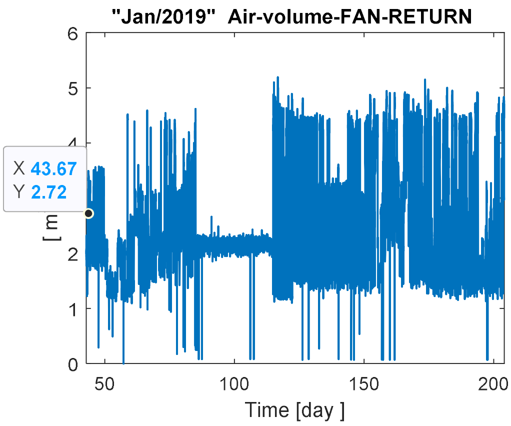
K1 = 1.1353e-14, K2 = 1.2732e-07, K3 = 0.60997

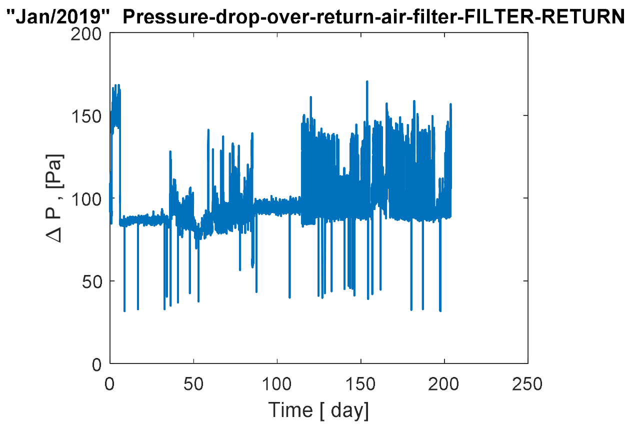


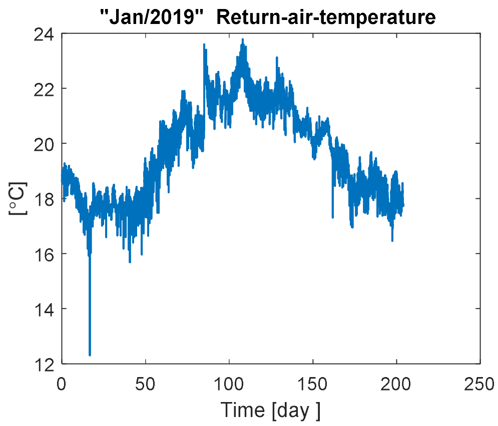
L1 = 0.058933, L2 = 3.1593e-07, L3 = 1.1639

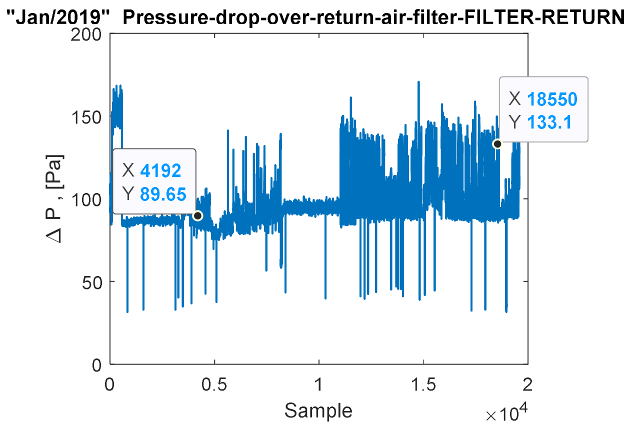
After three times attemp, from figure, it seems the tracking for exponentioal clogging model is not satisfied like polynominal. However, the R is near polynomial result.

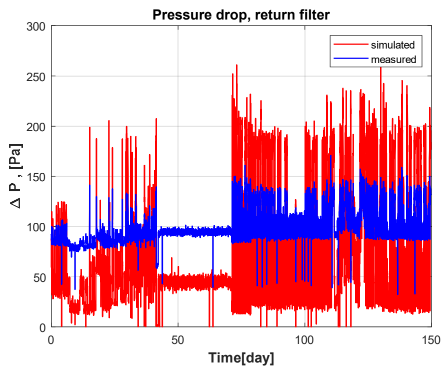
479

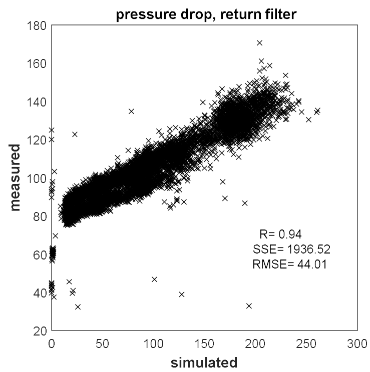












K1 = 1.8719e-21,   K2 = 6.3969e-15,   K3 = 8.6174

M-files

DefineProblem.m

```
%% Load Model Data

modelname='mymodel1';
data=LoadNominalData(modelname);

%% Define Parameters

params.a.Range=[0.9 1.5];
params.a.Value=[];
params.a.IsVariable=true;

params.b.Range=[0.3 0.9];
params.b.Value=[];
params.b.IsVariable=true;

params.c.Range=[0.5 1.1];
params.c.Value=[];
params.c.IsVariable=true;

params.d.Range=[0.1 0.5];
params.d.Value=[];
params.d.IsVariable=true;

nVar=0;
param_names=fieldnames(params)';
for p=param_names
    param=p{1};
    params.(param).Min=min(params.(param).Range);
    params.(param).Max=max(params.(param).Range);
    params.(param).Fixed=~isempty(params.(param).Value);

    if ~params.(param).Fixed
        nVar=nVar+1;
    end
end

data.params=params;

%% Define Decision Variables

VarSize=[1 nVar];    % Size of Decision Variables Matrix

VarMin=0;          % Lower Bound of Variables
VarMax=1;          % Upper Bound of Variables

%% Define Objective Function

CostFunction=@(sol) MyCost(sol,data);     % Cost Function
```

Mycost.m

```
function [z, info]=MyCost(sol,data)

    params=GetParams(sol,data);

    out=SimulateModel(data.modelname,params);

    t=out(:,1);
    x=out(:,2);
    y=out(:,3);

    z=sqrt(mean((x-data.x).^2))+sqrt(mean((y-data.y).^2));

    info.t=t;
    info.x=x;
```

```
    info.y=y;
    info.params=params;

end
```

Fan orifice mode which is used to control the flow rate of fan.

```
function delP  = FanOrifice(mdot,param)

delta_p_nominal=param(1);  % { 'Pa'   }Nominal pressure drop
mdot_nominal=param(2); % = { 'kg/s'  }; % Nominal mass flow rate
rho_nominal=param(3);  % = {0,     'kg/m^3'}; % Nominal density
laminar_fraction=param(4);   %{1e-3,  '1'     }; % Fraction of nominal
mass flow rate for laminar transition
mdot_lam = laminar_fraction * mdot_nominal; % Mass flow rate threshold
for laminar flow

% Square of mass flow rate linearized near zero flow
mdot_sqr = mdot * sqrt(mdot^2 + mdot_lam^2);    % eq  1.a
% % Pressure-flow relation
%     %delP=p_inlet - p_outlet
%     delP = delta_p_nominal / mdot_nominal^2 * mdot_sqr;   % eq 1
rho_outlet=0.955;  % kg/m^3
rho_inlet=rho_outlet;
% Pressure-flow relation
if rho_nominal > 0
    rho_avg = (rho_inlet + rho_outlet)/2;   %   inlet and outlet density
.. compressibility of gas
    delP = rho_nominal * delta_p_nominal / mdot_nominal^2 * mdot_sqr /
rho_avg;   % eq 2
else  % rho_nominal=0
    %delP=p_inlet - p_outlet
    delP = delta_p_nominal / mdot_nominal^2 * mdot_sqr;   % eq 1
end
```

Heat or energy recovery ventilator model.

```
function [dTeao, dTfao ] = ERV(u,Teao,Tfao,param)

Teai=u(1);
mea_dot=u(2);
Tfai=u(3);
mfa_dot=u(4);
Cam=param(1);
Cpa=param(2);
UAcc=param(3);
c1=param(4);
c2=param(5);

dTeao=(1/Cam)*(mea_dot*Cpa*(Teai-Teao)-UAcc*(((Teai+Teao)/2)-
((Tfai+Tfao)/2)))+c1;
dTfao=(1/Cam)*(UAcc*(((Teai+Teao)/2)-((Tfai+Tfao)/2))-mfa_dot*Cpa*(Tfao-
Tfai))+c2;
```

Heatermodel.

```
function [T_cold_out, T_hot_out ] = Heater(u,param,eps_NTU)

T_cold_in=u(1);
T_hot_in=u(2);
m_dot_cold=u(3);
m_dot_hot=u(4);
Cpa=param(1); % 1012   J/(kg*K) Specific heat of air at constant pressure [J/KgK]
Cpw=param(2); % 4181.3 J/(kg*K) Specific heat of water at constant pressure [J/KgK]
Cp_hot=Cpw;
```

```
Cp_cold=Cpa;
if m_dot_hot*Cp_hot > m_dot_cold*Cp_cold
    Q_dot=eps_NTU*m_dot_cold*Cp_cold*(T_hot_in-T_cold_in);
else
    Q_dot=eps_NTU*m_dot_hot*Cp_hot*(T_hot_in-T_cold_in);
end
T_cold_out = T_cold_in+Q_dot/(m_dot_cold*Cp_cold);
T_hot_out = T_hot_in-Q_dot/(m_dot_hot*Cp_hot);
```