Plausibility check and energy management in a semi-autonomous sensor network using a model-based approach

Mehrdad Babazadeh

Universität Bremen 2010
Plausibility check and energy management in a semi-autonomous sensor network using a model-based approach

Vom Fachbereich für Physik und Elektrotechnik
der Universität Bremen

zur Erlangung des akademischen Grades
Doktor-Ingenieur (Dr.-Ing.)
genehmigte Dissertation

von
MSc. Mehrdad Babazadeh

Referent: Prof. Dr.-Ing. Walter Lang
Korreferent: Prof. Dr.-Ing. Walter Anheier

Eingereicht am: 8. März 2010
Tag des Promotionskolloquiums: 3. Mai 2010
In the Name of Allah

To my dear Country, Iran

To my beloved spouse, Nasim and my dear son, Mohammad Taha

&

To all those I love.
Acknowledgment

I would like to express my gratitude to all those who gave the possibility to realize this research work. In particular, I wish to express my appreciation to Mr. Yousef Jameel for his financial support through the PhD scholarship. In this I would also like to include my gratitude to my supervisor, Professor Walter Lang for his continued encouragement and help during this work, Professor Hans Jörg Kreowski for very kind contribution and Professor Steven X. Ding, head of Institute for Automatic Control and Complex Systems (AKS), University of Duisburg-Essen, for his helpful directions. Also I appreciate Professor Rolf Isermann, Head of Institute for Automatic Control in Darmstadt University of Technology, for his useful hints. Also thanks to both Institute for Microsensors, -actuators and –systems (IMSAS) and International graduate school (IGS) to contribute in the experiments, participate in the presentations as well as good incites. I am also grateful to Ishwar Lal for his contribution through his Master of Science thesis.

Finally, I would like to thank my family. The encouragement and support from my beloved wife, Mrs. Nasim Moein and our always positive and joyful son, Mohammad Taha was a powerful source of inspiration and energy. A special thought is devoted to my parents for a never-ending support.
Abstract

The present dissertation carries out both energy management and model-based fault detection while using Wireless Sensor Networks (WSNs). It deals with an application of a WSN which uses scattered sensor nodes inside a closed space container to monitor environmental variables, temperature and relative humidity.

Since the environmental system under discussion is non-linear, multivariable and time variant, a hybrid mathematical model is extracted. A novel approach to simplify the hybrid model and decouple the monitoring variables is introduced for the first time in this research. This outstanding idea, so-called Floating Input Approach (FIA) exploits system identification as well as the properties of a distributed measurement systems to simplify the modeling task. It performs a Multi Input-Single Output (MISO) linear dynamic model and estimates environmental variables on a desired sensor node as output by using actual measured variables from surrounding sensor nodes as inputs.

Developing both on-line and off-line model identifications based on the FIA, model-based fault detection and energy saving of the wireless sensor network without performance degradation is successfully achieved. The FIA-based techniques detect and discriminate different fault types in sensors and system under discussion. Moreover, in the basis of the proposed mathematical dynamic model, an effective technique is introduced to enlarge life time of the sensor nodes. A combinational fault detection and energy management is introduced at the end.

Benefits of the addressed techniques are verified using simulations and implementations on a progressive platform of WSN, Imote2. They can also be developed simply for a wide variety of applications in the future.
Table of Contents

Acknowledgment .............................................................................................................. i
Abstract ............................................................................................................................. ii
Table of Contents ................................................................................................................ iii
List of Figures ...................................................................................................................... v
List of Tables ....................................................................................................................... viii
Synonyms, Abbreviations .................................................................................................. ix

1. Introduction...................................................................................................................... 1
   1.1 General context ........................................................................................................ 1
   1.2 Contribution of the thesis ..................................................................................... 2
   1.3 Outline of the thesis ............................................................................................... 3

2. Brief review of wireless sensor networks ................................................................. 4
   2.1 Specifications ........................................................................................................ 4
   2.2 Remarkable characteristics of Imote2 .................................................................. 6

3. Environmental model making ...................................................................................... 8
   3.1 Problem description ............................................................................................... 8
   3.2 Measurements used for model making ................................................................. 13
       3.2.1 Measurement test I .................................................................................... 13
       3.2.2 Measurement test II ................................................................................... 14
   3.3 A grey-box hybrid model .................................................................................... 16
   3.4 Floating Input Approach (FIA) ............................................................................ 23
       3.4.1 New topology of WSN and system identification based on FIA ............... 23
           3.4.1.1 Overview of Linear System Identification ........................................... 28
           3.4.1.2 Different model structures ................................................................. 28
           3.4.1.3 Different data numbers ................................................................... 32
           3.4.1.4 Different Fit indexes ....................................................................... 33
           3.4.1.5 Model order selection and number of KSNs .................................... 35
           3.4.1.6 Sensor position ................................................................................ 39
   3.5 Summary and conclusion chapter 3 ...................................................................... 42

4. Fault detection and diagnosis based on FIA .............................................................. 43
   4.1 Introduction .......................................................................................................... 43
   4.2 Faults in the system .............................................................................................. 44
   4.3 Measurements used for FDD .............................................................................. 45
   4.4 Fault detection mechanisms ................................................................................ 47
List of Figures

Figure 2.1. Network topology ........................................................................................... 5
Figure 2.2. Battery board of Imote2 .................................................................................. 6
Figure 2.3. Processor board of Imote2 .............................................................................. 7
Figure 2.4. Sensor board .................................................................................................... 7
Figure 3.1. Different states of normal air. ........................................................................ 9
Figure 3.2. A container equipped with scattered wireless sensor nodes. .......................... 9
Figure 3.3. Location of data loggers during the test in a closed space container ........... 13
Figure 3.4. Measurement results of T for a number of data loggers inside a container .. 14
Figure 3.5. Arrangement of data loggers during the test inside a closed room ............... 14
Figure 3.6. Measurement results of T and H during the experiment in a closed room ... 15
Figure 3.7. Container as an Input-Output grey-box model.............................................. 16
Figure 3.8. Psychrometric chart. ...................................................................................... 19
Figure 3.9. A container with inlet and three SNs. .......................................................... 21
Figure 3.10. Outputs when T, H and F in input change. ............................................... 22
Figure 3.11. Connections between the KSNs (K1... Km) and each DSN (S1 or S2)...... 24
Figure 3.12. One MIMO model for a pair KSN-DSN..................................................... 25
Figure 3.13. Two separate SISO models for a pair KSN-DSN based upon FIA............. 25
Figure 3.14. Measured T inside the container in three points (T_s= 150 s)..................... 26
Figure 3.15. Flowchart for offered estimation technique. ............................................ 27
Figure 3.16. ARMAX model structure .......................................................................... 29
Figure 3.17. Simulation based on a model. ................................................................. 30
Figure 3.18. One-step a head predictor. ............................................................. 30
Figure 3.19. Prediction using different estimation methods........................................ 31
Figure 3.20. Comparison of different data number used for model making. ........... 32
Figure 3.21. Off-Line estimation using 300/429 samples ............................................ 33
Figure 3.22. Comparing the result of prediction of Temperature (T)......................... 36
Figure 3.23. Comparing the result of prediction of Relative Humidity (H).............. 37
Figure 3.24. Measured signals by three sensor nodes (K1, K2 and S1)...................... 40
Figure 3.25. Comparing the result of prediction of R. Humidity \((H)\) while two estimators near and far from S1 are chosen. ............................................................................................................. 40
Figure 3.26. Impression of the KSNs on a DSN ............................................................. 41
Figure 4.1. Truck under test to measure \(T\) and \(H\) by Data loggers (iButton) .............. 45
Figure 4.2. (a) Setpoint and door position. (b) Actual measured \(T\) by different SNs. (c) Relative humidity. ............................................................................................................. 46
Figure 4.3. Basic scheme of fault detection with signal models ................................. 48
Figure 4.4. Multiple regions of operations and possible fault areas ......................... 50
Figure 4.5. Different situation of measurements of a DSN and KSN ......................... 52
Figure 4.6. Detecting faults for two near \(H\)-sensor .................................................... 54
Figure 4.7. Detecting faults for two far SNs which measure H ..................................... 54
Figure 4.8. Basic scheme of fault detection with process models ............................. 57
Figure 4.9. Basic scheme of fault detection based on process models-based by FIA .... 57
Figure 4.10. Fault detection by using residual of off-line estimation ....................... 61
Figure 4.11. (a) Measured \(T\). (b) Estimation using inlet and K1; \(T_s= 150\) s ............. 62
Figure 4.12. Detecting faults for two near \(H\)-sensor .................................................... 64
Figure 4.13. Prediction with: (a) Ordinary method. (b) Old estimated parameters. (c) Normalized \(P(t)\). ............................................................................................................. 65
Figure 4.14. Prediction when there is an input-output sudden change with updated matrix \(P\) based on prediction error (with update parameters). ......................... 66
Figure 4.15. Output, with and without faults, prediction using \{(a) Improved prediction. (c) Normal prediction\}. Residual using: \{(b) Improved prediction. (d) Normal prediction\}. Fault detection signal (e). ............................................................................ 66
Figure 4.16. Parameter variations when forgetting factor is chosen big \((\lambda =1)\) ........ 67
Figure 4.17. Parameter variations when forgetting factor is chosen small \((\lambda =0.5)\) ...... 68
Figure 4.18. Adaptive threshold generator in the area of Laplace transform (S) ......... 69
Figure 4.19. Input, prediction, output with and without fault and residual .................. 70
Figure 4.20. Topology of single detector ........................................................................ 71
Figure 4.21. Topology of multiple detectors ................................................................. 72
Figure 4.22. Report issued by A KSN during an experiment ....................................... 72
Figure 4.23. Report issued by Main station an experiment ........................................... 73
Figure 4.24. Block diagram of Combinational Fault Diagnosing .................................. 74
# List of Tables

Table 2.1. Characteristics of *T-sensor* and *H-sensor* on Imote2 ....................................... 7
Table 3.1. Sensor positions ....................................................................................................... 15
Table 3.2. Initial conditions .................................................................................................... 22
Table 3.3. The most important Common linear models with input .......................................... 29
Table 3.4. Evaluations with different number of data, indexes, orders, SISO and MISO for estimating $H$ .................................................................................................................................... 38
Table 4.1. Sensor locations during the measurement test ......................................................... 45
Table 4.2. Fault models ........................................................................................................... 53
Table 4.3. Qualitative comparison of properties of fault detection methods for linear processes ................................................................................................................................... 55
Table 4.4. Survey of important identification methods ............................................................ 56
Table 5.1. Maximum time each node survived with single recharge of battery ...................... 89
Table 5.2. Power consumption of Imote2 in different modes .................................................. 91
Table 5.3. Imote2 operating specifications .............................................................................. 92
## Synonyms, Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ARX</td>
<td>Auto Regressive with eXternal (exogenous) input(s) method</td>
</tr>
<tr>
<td>ARMAX</td>
<td>Auto Regressive Moving Average with eXternal input(s) method</td>
</tr>
<tr>
<td>BJ</td>
<td>Box-Jenkins method</td>
</tr>
<tr>
<td>CFD</td>
<td>Computational fluid dynamics</td>
</tr>
<tr>
<td>DSN</td>
<td>Desired Sensor Node</td>
</tr>
<tr>
<td>ESM</td>
<td>Energy Saving Mode</td>
</tr>
<tr>
<td>EV</td>
<td>Environmental Variables</td>
</tr>
<tr>
<td>$F$</td>
<td>Air Flow</td>
</tr>
<tr>
<td>FD</td>
<td>Fault detection</td>
</tr>
<tr>
<td>FDD</td>
<td>Fault Detection and Diagnosis</td>
</tr>
<tr>
<td>FIA</td>
<td>Floating Input Approach</td>
</tr>
<tr>
<td>$H$</td>
<td>Relative Humidity</td>
</tr>
<tr>
<td>$H$-sensor</td>
<td>Relative Humidity sensor</td>
</tr>
<tr>
<td>KSN</td>
<td>Key Sensor Node</td>
</tr>
<tr>
<td>LTI</td>
<td>Linear Time Invariant</td>
</tr>
<tr>
<td>MIMO</td>
<td>Multi Input - Multi Output</td>
</tr>
<tr>
<td>MISO</td>
<td>Multi Input - Single Output</td>
</tr>
<tr>
<td>NC</td>
<td>Normalized Covariance</td>
</tr>
<tr>
<td>OE</td>
<td>Output Error method</td>
</tr>
<tr>
<td>SISO</td>
<td>Single Input - Single Output</td>
</tr>
<tr>
<td>SN</td>
<td>Sensor Node</td>
</tr>
<tr>
<td>SS</td>
<td>State Space method</td>
</tr>
<tr>
<td>$T$</td>
<td>Temperature</td>
</tr>
<tr>
<td>$T$-sensor</td>
<td>Temperature sensor</td>
</tr>
<tr>
<td>$u$</td>
<td>System input</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
</tr>
<tr>
<td>$y$</td>
<td>Actual measured output</td>
</tr>
<tr>
<td>$\hat{y}$</td>
<td>Estimated output</td>
</tr>
</tbody>
</table>
1. Introduction

1.1 General context

Enhanced attention in the academic community and in industry due to increasing demands on the performance of the processes since about 1960 caused significant progresses in the application of automation on the industrial processes. Later on after about 1975 which microcomputers were available to contribute effectively to solve automation problems, specialists tried to interacting them in the practice. At the same time further progresses made further feasibility for the application of automation. The next necessity was automatic supervision of such systems. A high degree of system safety and reliability was required for the complicate processes to achieve more cost efficient and high performance systems.

The problem under discussion is improving the performance of a monitoring system. A Wireless Sensor Network (WSN) monitors Environmental Variables (EVs) containing Temperature ($T$) and relative Humidity ($H$) in different positions of the internal space, top, bottom, on the surfaces and possibly inside the freights loaded inside the containers which transport cargo. To have a reliable environmental monitoring in this scenario, a Fault Detection and Diagnosis (FDD) system is necessary whereas performance of the system may degrade with unexpected faults. Further on, a novel approach to reduce total power consumption of Sensor Nodes (SNs) is introduced since the energy depletion of batteries of the SNs is a main source of arising sensor faults. In fact, the present research work involves two fundamental concepts, developing model-based methodologies for fault detection and energy saving in the sensor nodes.
Faults reported via this semi-autonomous supervision system might be signs to occurrence of risks for the quality of the loaded freight items. These faults can also be evaluated in a higher level by a main station to make necessary predictive decisions.

1.2 Contribution of the thesis

This research deals with a time variable, nonlinear and multivariable environmental system. It suggests novel methods for energy saving in a wireless sensor network as well as fault detection and diagnosis for both the network elements and the environment.

Based on thermodynamical relations and empirical behavior of the environmental variables, a new approximate hybrid model of the environment is primarily developed. It can be considered as a grey-box model for this environmental system whereas full information is not accessible for such complex dynamic system and accurate modeling of this system is not our main objective.

Three important variables, temperature, relative humidity and air flow are considered in the first stage of model making to achieve a hybrid model. After that, from a new view of distributed monitoring systems, the model is simplified mathematically. It infers a new Floating Input Approach (FIA) to be used for both fault detection and energy management. This novel approach performs a Multi Input-Single Output (MISO) linear model by using advantages of a sensor network. In the other word, getting help from surrounding sensor nodes as inputs of a MISO model, information of a desired sensor node as output can be estimated.

The techniques are generally independent from the type of utilized ventilation system and can also be effectively applied to several applications to increase the reliability of related monitoring system. The acquired results are supported by real experiments and some practical rules to attain a near optimal estimation of the EV are introduced. An energy saving method, introduced for the first time by the present research might be considered as a novel method in this area. It can be applied to the network even while any other kind of energy saving techniques exists. Furthermore, many types of faults in the system are detected and diagnosed by using the proposed model-based methods in this report. All the mentioned techniques have been implemented on a wireless platform by using Imote2 and a full implemented scenario has been tested successfully.
1.3 Outline of the thesis

The first chapter expresses an introduction to the problem and research objective where the second chapter reviews WSNs in brief. Third chapter describes a mathematical hybrid grey-box model for the environmental variables. In this chapter a new idea for model making in a space containing a sensor network has been addressed. Several alternatives have also been studied to achieve the best model structure and characters. After that a new Floating Input Approach (FIA) to simplify the hybrid model has been introduced. The forth chapter develops model-based fault detection methods. It applies the existing methods of fault detection and then develops powerful tools to discriminate additional fault types, malfunctions and failures. An adaptive threshold method is also developed and combined with the other methods. In addition to fault detection, a fault diagnosis is provided and about forty types of possible sensor and system faults are distinguished for two important environmental variables, temperature and relative humidity. More information about a novel technique for Energy Saving based on the FIA is given in the chapter five. The last step is combining energy saving and fault diagnosis units which is also proposed in the chapter five.

Finally, all remarkable conclusions are briefly summarized and a few suggestions for future research direction are given.
2. Brief review of wireless sensor networks

2.1 Specifications

Physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, at different locations can be cooperatively monitored by a Wireless Sensor Network (WSN) consisting of spatially scattered autonomous devices using sensors. Although they have been motivated originally via military applications, they are now used in many industrial and civilian applications, including industrial process monitoring and control, machine health monitoring, environment and habitat monitoring, healthcare applications, home automation, and traffic control.

Each node of a WSN is typically equipped with a radio transceiver or other wireless communication devices, a small microcontroller, and a limited energy source, usually a battery. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and bandwidth. A WSN as briefly categorized in the following contains several components, depending on its application, its deployment strategy, packet routing, its size and final data collection.

✓ Components

- Wireless sensor node
  - Power Supply
  - Processor Unit
  - Sensing Hardware
  - Communication module
According to Figure 2.1, among the existing topologies of a network, three important types, applicable to a wireless sensor network are defined in the following:

**Star and extended Star Topology**- One of the most popular topologies is the star and extended star topology. It is easy to setup and relatively cheap. The Star Topology works by connecting each node to a central device. This central connection allows us to
have a fully functioning network even when other devices fail. The only real threat to this topology is that if the central device goes down, so does the entire network.

The Extended Star Topology is a bit more advanced. Instead of connecting all devices to a central unit, we have sub-central devices added to the mix.

**Hierarchical Topology**- It is a Tree Topology, much like the Star Topology, except that it doesn’t use a central node. This type of topology suffers from the same centralization flaw as the Star Topology. If the device on top of the chain fails, consider the entire network down.

### 2.2 Remarkable characteristics of Imote2

Utilized in this research as a pilot wireless sensor module, Imote2 contains a 32 MB flash memory and the Intel PXA271 CPU and operates in a low voltage (0.85V), low frequency (13MHz) mode, hence enabling very low power operation. As shown in Figure 2.2, it can be powered via a battery board as well as an on-board mini-B USB connector. The frequency is scaled from 13MHz to 416MHz with Dynamic Voltage Scaling. The Imote2’s CC2420 radio is tuned within the IEEE 802.15.4 channels that are numbered from 11 (2.405 GHz) to 26 (2.480 GHz), each separated by 5 MHz. The processor has a number of different low power modes such as sleep and deep sleep.

![Figure 2.2. Battery board of Imote2](image)

The following list summarizes the changes that are incrementally applied to the processor to achieve low power consumption:

1. Normal : CPU and peripherals are active.
2. Idle : CPU is inactive.
3. Deep-idle : Core PLL is disabled.
4. Standby: Peripheral PLL is disabled.

5. Sleep: Low-voltage power domains and internal SRAM are disabled.

6. Deep-sleep: High-voltage power domains are disabled.

Figure 2.3 below illustrates processor module via Cross Bow Technologies, the module has an integrated communication handling hardware and a strip antenna.

Sensing hardware consists of various sensors and their respective signal conditioning circuits (Figure 2.4).

Table 2.1 illustrates specifications of T-sensor and H-sensor. One might refer to [48] to see further technical details about it.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>$H$ (%)</th>
<th>$T$ (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Temperature and Humidity Sensor</td>
<td></td>
<td></td>
</tr>
<tr>
<td>(Sensirion SHT15, 2 Channels):</td>
<td>12 bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td>10 - 90% @</td>
<td>+/-2%</td>
</tr>
<tr>
<td></td>
<td>0 - 100% @</td>
<td>+/-4%</td>
</tr>
<tr>
<td></td>
<td>14 bit</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - 40 @</td>
</tr>
<tr>
<td></td>
<td></td>
<td>-40 - 120 @</td>
</tr>
<tr>
<td>Digital Temperature Sensor (TI TMP175 with I2C interface)</td>
<td></td>
<td>-25 - 85 @</td>
</tr>
</tbody>
</table>
3. Environmental model making

3.1 Problem description

A model is defined as a representation of the essential aspects of a system which presents knowledge of the system in a usable form. Identification, modeling and control of Environmental Variables (EVs) in the air conditioned closed spaces have gained a lot of attractions during the last few years. Containers utilized for transporting goods are exemplar of such spaces. Simple and precise mathematical models play a key role in this area and developing either linear or nonlinear models is very vital on this issue. Different kinds of disturbances and unwanted conditions exist in this environment. Different fault types, failure or malfunctions might also arise in the corresponding monitoring system. Freights have their own initial temperature and humidity before loading in to the container and alter the EVs of surrounding after loading. The cooling system inside the container starts up whilst various initial conditions of the environment inside the container, Temperature ($T$) and relative Humidity ($H$) in different positions exist. It takes some time to regulate $T$ which is fully affected from thermodynamical parameters and nature of the freights.

**Relative humidity** - It is a ratio (in percent) of the actual amount of water vapor in the atmosphere compared to the saturation amount, if the saturation amount decreases, the ratio must increase. If the temperature and pressure change, the relative humidity will change too. Higher temperature allows the atmosphere to hold more water. It is possible to compare how much water vapor is present in the air to how much water vapor would be in the air if the air were saturated.
CHAPTER 3. Environmental model making

Figure 3.1 represents measured values of relative humidity in different states when temperature or pressure is changed. P is the system pressure and $P_{\text{H}_2\text{O}}(T)$ is the saturated vapor pressure of water at temperature $T$. Nevertheless $H$ in this space depends on several parameters and cannot be controlled directly.

![Figure 3.1. Different states of normal air.](image)

State (I)
- $P = 101.3$ kPa
- $T = 70 \, ^\circ\text{C}$
- $H = 50\%$
- $P_{\text{H}_2\text{O}}(70\, ^\circ\text{C}) = 31.2$ kPa

State (II)
- $P = 101.3$ kPa
- $T = 80 \, ^\circ\text{C}$
- $H = 33\%$
- $P_{\text{H}_2\text{O}}(80\, ^\circ\text{C}) = 47.4$ kPa

State (III)
- $P = 201.3$ kPa
- $T = 70 \, ^\circ\text{C}$
- $H = 99.4\%$
- $P_{\text{H}_2\text{O}}(70\, ^\circ\text{C}) = 31.2$ kPa

As shown in Figure 3.2 below, a WSN has been utilized to monitor the environmental variables in different positions. Nonlinear multivariable nature and interconnections between the variables of the EVs in addition to the presence of a load as an unpredictable, immeasurable disturbance, effects of fluid dynamic, influence of surfaces inside the container increase complexity of the model which we are looking for.

![Figure 3.2. A container equipped with scattered wireless sensor nodes.](image)
Changing Temperature ($T$), relative Humidity ($H$) or even air Flow ($F$) in inlet may change both $T$ and $H$ in all positions of the desired space. Affected by disturbances, measurements might be different even in the same place, over the similar experiments.

Lots of research activities in the field of fluid dynamics have been dedicated to create thermodynamic models for Environmental Variables (EVs) in closed spaces. They usually look for a way to improve cooling systems, ventilation or to attain homogeneity in the space. They try to either modify the quality of supply chain or decrease the energy consumed by the cooling systems.

There are white, grey, and black-box models of $T$ for air-handling units have been addressed in some previous works. It is represented in [1] that the model of the air handling unit elements is nonlinear and $T$ and $H$ as controlled variables are coupled. It assumes constant air flow as a parameter influenced on the other parameters. Also it assumes that $T$ and $H$ change with a constant speed. It uses grey-box approach to combine theoretical modeling, parameter identification of discrete models and partially known models by using optimization techniques. It uses energy balance to achieve transfer functions of transducers. It acquires a model for any device and then identifies unknown parameters by using separate tests. Assuming special conditions it decouples $T$ and $H$ and uses separate linear transfer functions for them.

Analytical and numerical models to describe the dynamics of the cryogenic freezing tunnel system have been studied in [2]. By a composite model, it uses finite difference methods for sizing the tunnel freezer. It also talks about freezing and freezer dynamics that is useful to have a view of such systems. It argues that heat transfer with phase change is a highly non-linear problem.

Reference [3] is a brief review of numerical models of $F$ in refrigerated food applications using (k-c) model and also a data-base mechanistic modeling technique. They obtain partial differential equations using Computational Fluid Dynamics (CFD) which are without general analytical solution. It is a simulation tool for modeling of fluid flow problems based on the solution of the governing flow equation. Although this method gives high precision, we can’t use it, because this process is necessarily iterative and requires the solution of a huge number of equations at each step.

To model a 3-D spatio-temporal temperature distribution in an imperfectly mixed forced ventilated room for control purposes they exploit a second order model in [4]. It
delivers good definitions of different types of models (white, grey and black) in a cooling system. It introduces a hybrid between the extremes of mechanistic and data based modeling. This so-called Data Based Mechanistic (DBM) models provide a physically meaningful description of the dominant internal dynamics of heat and mass transfer. It uses static experiments to examine the effect of the ventilation rate on the spatial temperature homogeneity, whilst keeping the average temperature inside the ventilated chamber constant. It points that increasing the ventilation rate decreases the standard deviation of temperature in different places. In a specific rate maximum uniformity is achieved. It fits a curve to $T$ in different places and uses model-based predictive method to optimal control of spatial $T$ distribution. But, it doesn’t consider relative humidity.

A combination of CFD and DBM methods is investigated in [5]. It outlines a methodology to achieve an accurate model of $T$ in a closed space. First of all using k-$\varepsilon$ model, turbulence is modeled and then a DBM model is formulated from an energy balance equation. It can reduce complexity of CFD using identification technique. It doesn’t consider relative humidity. Some first order models between inlet and individual zones are considered assuming a constant air flow rate.

There are also a lot of papers which use neural networks to make model in this area. Authors of [6] exemplify a Neural network based Nonlinear Auto Regressive with eXternal input model (NNARX) for modeling the internal greenhouse temperature as a function of outside air temperature and humidity. Because of slow nature of the system, it doesn’t need of frequent retuning the parameters.

Numerical and experimental characterization of air flow within a semi trailer enclosure with pallets has been reported in [7]. The effect of air flow pattern on $T$ is given by this paper. The numerical modeling of $F$ is performed using CFD code fluent and second–moment closure, the Reynolds Stress Model (RSM). It indicates importance of air ducts in decreasing temperature differences throughout the cargo. It implies that prediction using k-$\varepsilon$ models are often not accurate. It investigates numerically and experimentally the air flow pattern throughout a vehicle enclosure loaded with two rows of pallets with and without an air duct system.

Using CFD method flow pattern inside the working area of a pilot scale clean room has been numerically investigated in [8]. Two versions of the k-$\varepsilon$ turbulence model have
been tested. To solve transport equations the surfaces bounding the domain has been
defined clearly during this work and comparisons between turbulence models have been
done as well. As mentioned in [9], there are two ways to define a grey box model. One
way emanates from the black box model frame. A prior knowledge is incorporated as
constraints on model parameters or variables. Second way is to begin with a model
originating from mathematical relations, which describe the behavior of the system. This
means the starting point is a specific model structure based on physical relations.

The transport planning for goods with different temperature requirements form a
special case of a vehicle routing problem [10]. The planning can be improved by
analysis and prediction of local temperature deviations. The assignment of transport
items to different temperature zones and trucks can be done more accurately. The risk
for temperature abuse may be evaluated based on the predicted temperature curve for
the position of the item inside the truck or container.

All mentioned models in above are acquired between system input so-called inlet
and a point in the corresponding space. With the mentioned models, the environmental
variables in a desired point changes due to variation in the inlet. Some of these models
either linear or nonlinear do not consider interconnections of the EVs. Likewise,
particular conditions and limited range of parameter variations of such models are
required. Despite the high precision, complexity makes some of them impractical and
the rest inaccurate of course in some applications.
3.2 Measurements used for model making

Two series of measurements were utilized to accomplish an approximate hybrid environmental grey-box model. The model was improved considering extra simulation results and mathematical viewpoints. These are treated in the following sections.

3.2.1 Measurement test I

Figure 3.3 represents the place of data loggers during a field test in cooperation with a German food retailer [11] on 17.04.2007 to capture spatial temperature profiles. Up to 40 data loggers had been mounted at the walls of the compartment for fish and meat. A 2-point control turned on the ventilation if $T$ below the refrigeration unit rose above a given setpoint. Reefer unit is located near to sensor No. 8, shown with red color. And closed doors are near to sensors 23 and 40 shown with green color.

For the sake of transparency, Figure 3.4 gives measured $T$ by only 20 selective sensors. A few measurements will be used in the simulations during the next parts.

We will try to choose the worst cases for simulations to be sure about the performance of the techniques while using in real applications. It is important to note that because of practical constraints in real applications, only a few sensor nodes might...
be utilized. When cooling system starts, all measured $T$ come down in the very first part of the plot. The different measurements correlate with each other, but they have different values as well as dynamics. The biggest changes are related to restarting cooling system when $T$ differs from the setpoint.

![Figure 3.4. Measurement results of $T$ for a number of data loggers inside a container](image)

### 3.2.2 Measurement test II

From Figure 3.5, a three days measurement in a closed space room was done according to the following arrangement of data loggers (*iButton*) starting from 30/04/2008.

![Figure 3.5. Arrangement of data loggers during the test inside a closed room](image)
The *iButton* is a computer chip enclosed in a 16 mm thick stainless steel can. They measured $T$ and $H$ with sampling time equal with 10 minutes during 2360 min (39.3 hours) of whole test period. Table 3.1 and Figure 3.6 show data loggers positions and their measurement results respectively. The occasionally sunshine during the day time acts as a disturbance ($1080 < t < 1400$ min) and influences on the room environment. The lowest values show $T$, measured during midnight. The lower graph illustrates measured $H$ and a reverse relationship exists between $T$ and $H$.

### Table 3.1. Sensor positions

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Place</th>
<th>Sensor name</th>
<th>X (m)</th>
<th>Y (m)</th>
<th>Z (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>T</em> &amp; <em>R. Humidity</em> (%)</td>
<td>By the window</td>
<td>F1</td>
<td>2</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>By the window</td>
<td>F4</td>
<td>0.1</td>
<td>0.1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>On the drawers</td>
<td>F10</td>
<td>2</td>
<td>2.5</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>On the desk</td>
<td>F11</td>
<td>1.5</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F12</td>
<td>1.5</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>F13</td>
<td>1</td>
<td>1.5</td>
<td>1</td>
</tr>
</tbody>
</table>

![Figure 3.6. Measurement results of $T$ and $H$ during the experiment in a closed room](image-url)
3.3 A grey-box hybrid model

Figure 3.7 draws a general input-output scheme of an environmental system inside the container between the inlet (reefer unit) and a SN. It is a time varying, multi-variable system with three inputs, three outputs, disturbance, and measurement noise.

As the first step of model making based upon the approximate behavior of this thermodynamical system, a simple mathematical linear model was chosen. With this linear transfer function matrix between input (inlet) and outputs (SNs) shown in (3.1), there will be an independent MIMO model for a pair inlet-SN in terms of the EVs. Therefore, for all the SNs, several MIMO models are assigned. The arrays of the matrix denote the effects of variation in inlet on respective SN in the domain of Z-transform.

\[
\begin{pmatrix}
    T_{\text{SN}}(z) \\
    H_{\text{SN}}(z) \\
    F_{\text{SN}}(z)
\end{pmatrix} =
\begin{pmatrix}
    G_T(z) & -G_{HT}(z) & 0 \\
    -G_{TH}(z) & G_H(z) & 0 \\
    0 & 0 & G_F(z)
\end{pmatrix}
\begin{pmatrix}
    T_{\text{inlet}}(z) \\
    H_{\text{inlet}}(z) \\
    F_{\text{inlet}}(z)
\end{pmatrix}
\]  

(3.1)

For ith sensor, \((T_{\text{SN}}, H_{\text{SN}} \text{ and } F_{\text{SN}})\) in the above matrix equation denote measured value of the EVs in an output sensor based on those from inlet \((T_{\text{inlet}}, H_{\text{inlet}} \text{ and } F_{\text{inlet}})\). Transfer functions \(G_T, G_H\) and \(G_F\) represent direct impress of \(T, H\) and \(F\) from inlet to the same variable in the selected SN. Interconnections between \(T\) and \(H\) are written by \(G_{HT}\) and \(G_{TH}\). Increasing either \(T\) or \(H\) reduces another one where a negative sign for interconnections implies these influences which accords with the measurement test II as shown already in Figure 3.6. The parameters, \(T\) and \(H\) don’t influence on \(F\).
The air flow is assumed to impress only the speed of the other variables and not their steady state values. As shown in (3.2) and (3.3), time constant of the transfer functions have been considered as a function of \( F \). It can be an exponential function in \( f(.), g(.), h(.) \) and \( m(.) \). References [1] and [5] have already suggested a first order transfer function for \( T \) in a closed space. Our thermodynamical system is slow thus to work with steady state responses and to make simplicity, first order transfer functions is used for both direct effects and interconnections.

\[
G_f = K_f \prod_{j=1}^{nF} \frac{(Z - Z_j)}{(Z - \alpha \cdot P_j)} \quad , \quad G_H = K_H \prod_{j=1}^{nH} \frac{(Z - Z_j)}{(Z - \beta \cdot P_j)} , \quad \alpha = f(\text{flow}) , \beta = g(\text{flow}) \quad (3.2)
\]

\[
G_{TH} = K_{TH} \prod_{j=1}^{nTH} \frac{(Z - Z_j)}{(Z - \gamma \cdot P_j)} \quad , \quad G_{HT} = K_{HT} \prod_{j=1}^{nHT} \frac{(Z - Z_j)}{(Z - \lambda \cdot P_j)} , \quad \gamma = h(\text{flow}) , \lambda = m(\text{flow}) \quad (3.3)
\]

To study the validation area of the model (3.1) with regard to steady state behavior, a reverse lemma is given with three assumptions in different border conditions:

**Assumption 1:** in case \( T_{\text{inlet}} = T_{\text{max}} \quad , \quad H_{\text{inlet}} = H_{\text{min}} = 0 \) \quad (3.4)

\[
T_{\text{oss}(SN)} = \lim_{t \to \infty} Z^{-1}(G_f \bullet T_{\text{max}}) \quad , \quad H_{\text{oss}(SN)} = -\lim_{t \to \infty} Z^{-1}(G_{TH} \bullet T_{\text{max}}) \quad (3.5)
\]

With \( T_{\text{oss}} \) and \( H_{\text{oss}} \) which are steady state values of measured output, we should have:

\[
T_{\text{oss}(SN)} \leq T_{\text{max}} \quad , \quad H_{\text{oss}(SN)} \geq 0 \xrightarrow{T_{\text{th}} \to \infty} 0 \leq |G_f| \quad , \quad |G_{TH}| \leq 1 \quad , \quad |T_{\text{max}}| \geq 0 \rightarrow H_{\text{oss}(SN)} \leq 0 \quad (3.6)
\]

But negative \( H \) is not admissible. With maximum permissible values of output sensor \( (T_{\text{oss}} \) and \( H_{\text{oss}}) \) to find permissible margins we have:

**Assumption 2:** in case \( T_{\text{inlet}} = T_{\text{max}} \quad , \quad H_{\text{inlet}} = H_{\text{min}} \neq 0 \) \quad (3.7)
CHAPTER 3. Environmental model making

\[ 0 \leq Z^{-1}(G_H \cdot H_{\min} - G_{TH} \cdot T_{\max}) \leq H_{o \max} \quad (3.8) \]

Subsequently:
\[ Z^{-1} \left( \frac{G_H \cdot H_{\min} - H_{o \max}}{G_{TH}} \right) \leq T_{\max} \leq Z^{-1} \left( \frac{G_H \cdot H_{\min}}{G_{TH}} \right) \quad (3.9) \]

And:
\[ T_{0 \min} \leq Z^{-1}(G_T \cdot T_{\max} - G_{HT} \cdot H_{\min}) \leq T_{0 \max} \quad (3.10) \]

Then:
\[ Z^{-1} \left( \frac{G_T \cdot T_{\max} - T_{0\max}}{G_{HT}} \right) \leq H_{\min} \leq Z^{-1} \left( \frac{G_T \cdot T_{\max} - T_{0\min}}{G_{HT}} \right) \quad (3.11) \]

The equation (3.11) represents the limitation while using model (3.1) as a simple approximate model where \( T_{\min} \) and \( T_{\max} \) are minimum and maximum of permissible \( T \).

Assumption 3: in case
\[ Z^{-1} \left( \frac{G_H \cdot H_{\max} - H_{o \max}}{G_{TH}} \right) \leq T_{\min} \leq Z^{-1} \left( \frac{G_H \cdot H_{\max}}{G_{TH}} \right) \quad (3.12) \]

\[ T_{0 \min} \leq Z^{-1}(G_T \cdot T_{\min} - G_{HT} \cdot H_{\max}) \leq T_{0 \max} \quad (3.13) \]

\[ Z^{-1} \left( \frac{G_T \cdot T_{\min} - T_{0\max}}{G_{HT}} \right) \leq H_{\max} \leq Z^{-1} \left( \frac{G_T \cdot T_{\min} - T_{0\min}}{G_{HT}} \right) \quad (3.14) \]

Finally we will have:
\[ T_{\max} = Z^{-1} \left( \frac{G_H \cdot H_{\min}}{G_{TH}} \right), \quad T_{\min} = Z^{-1} \left( \frac{G_H \cdot H_{\max} - H_{o \max}}{G_{TH}} \right) \quad (3.15) \]

\[ H_{\min} = Z^{-1} \left( \frac{G_T \cdot T_{\max} - T_{0\min}}{G_{HT}} \right), \quad H_{\max} = Z^{-1} \left( \frac{G_T \cdot T_{\min} - T_{0\min}}{G_{HT}} \right) \quad (3.16) \]

Having \( H_{\min} \) and \( H_{\max} \), minimum and maximum of permissible \( H \) in inlet, other input limitations are figured out. Hence, there are specific bands for inputs in order that outputs of linear model stay in the admissible areas. The linear model (3.1) cannot be a
suitable model for whole range of variations of $T$ and $H$. Therefore a more consistent hybrid model is made based on the basic knowledge of the nonlinear nature of the interconnections. This nonlinearity can be represented by a psychrometric chart which presents physical and thermal properties of moist air in a graphical form as shown in Figure 3.8. It is helpful whilst working about environmental problems.

![Psychrometric Chart](image)

Figure 3.8. Psychrometric chart.

A psychrometric chart is a graph of the thermodynamic properties of moist air at a constant pressure (often equated to an elevation relative to sea level). The properties are:

**Dry-Bulb Temperature (DBT)** is that of an air sample, as determined by an ordinary thermometer, the thermometer’s bulb being dry, horizontal axis of the graph.

**Wet-Bulb Temperature (WBT)** is that of an air sample after it has passed through a constant-pressure, ideal, adiabatic saturation process. In practice, this is the reading of a thermometer whose sensing bulb is covered with a wet sock evaporating into a rapid stream of the sample air. When the air sample is saturated with water, the WBT will read the same as the DBT. The slope of the line of constant WBT reflects the heat of vaporization of the water required to saturate the air of a given relative humidity.
Dew Point Temperature (DPT) is that temperature at which a moist air sample at the same pressure would reach water vapor “saturation.” At this point further removal of heat would result in water vapor condensing into liquid water fog or (if below freezing) solid hoarfrost. The dew point temperature is measured easily and provides useful information, but is normally not considered an independent property of the air sample. It duplicates information available via other humidity properties and the saturation curve.

The versatility of the psychrometric chart lies in the fact that by knowing three independent properties of some moist air (one of which is the pressure), the other properties can be determined. By using the above chart, nonlinear functions for the interactions are driven from a few basic thermodynamic relations expressed in (3.17).

\[
H = H_0 \cdot 2^{-\frac{(T-T_0)}{10.1}} \quad \text{or} \quad T = T_0 - \frac{10.1}{\ln 2} \cdot \ln \frac{H}{H_0}
\]  

(3.17)

T and H in above represent secondary values and T0 and H0 initial values of temperature and relative humidity. According with above equations, 10.1°C increasing of T in the area of a SN reduces measured H to the half. The hybrid model, a combination of linear direct effects and nonlinear interconnections is written now as (3.18) in time domain:

\[
\begin{align*}
T_{SN}(t) &= Z^{-1}(G_{T,F} \cdot T_{inlet}) + g(H_{inlet}, F_{inlet}) + N_T \\
H_{SN}(t) &= f(T_{inlet}, F_{inlet}) + Z^{-1}(G_{H,F} \cdot H_{inlet}) + N_H \\
F_{SN}(t) &= Z^{-1}(G_{F} \cdot F_{inlet}) + N_F
\end{align*}
\]  

(3.18)

In this \(N_T\), \(N_H\), and \(N_F\) represent the SNs’ measurement noises. \(Z^{-1}\) is reverse transfer in domain of Z-transform. Functions \(g(\cdot)\) and \(f(\cdot)\) denote nonlinear interconnection terms influenced by \(F\). \(G_{T,F}\) and \(G_{H,F}\) are transfer functions for \(F \rightarrow T\), \(F \rightarrow H\) and \(F \rightarrow F\) for each pair of inlet-SN. \(\Delta T\) and \(\Delta H\) denote nonlinear parts of \(T\) and \(H\) which contain measurement noise.

Consisting of nonlinear interconnections, the model (3.18) is used for any pair of inlet-SN to attain an estimation of the EVs in the SN. It is noted also that although the variable \((F)\) in the primary formula was written, it is not considered directly in the
measurements and other applications during this research. However, its effects on the
other variables \((T\) and \(H\)) are considered. Nonlinear interconnections are extracted from
\((3.17)\) in the following where \(T_0\) and \(H_0\) are initial conditions of the EVs for the SNs,
respectively.

\[
\Delta T(t) = T_0 - \frac{10.1}{\ln 2} \ln \frac{Z^{-1}(G_H \cdot H_0) + N_H(t)}{Z^{-1}(K_H \cdot H_0)}
\]  
\[
\Delta H(t) = (2^{1.10} \cdot e) \frac{-Z^{-1}(G_H \cdot T_0) + N_H(t)}{10^{\frac{1}{1.10}}} + Z^{-1}(K_H \cdot H_0) + N_H(t)
\]

\[
\Delta T(t) = g(.) + N_T, \quad \Delta H(t) = f(.) + N_H
\]

\[
T_{SN}(t) = T_{linear (fromT_{calc})} + T_{nonlinear (fromH_{calc})}
\]

\[
H_{SN}(t) = H_{linear (fromH_{calc})} + H_{nonlinear (fromT_{calc})}
\]

The model is not due to a real super position. That is only an assumption and its
properties will be treated in the following parts. Over the next chapter, exploiting the
advantages of plurality of measuring points in this WSN, above hybrid model is
simplified. For simulation mean, Figure 3.9 illustrates three SNs accompany with some
surrounding objects with different initial conditions written in Table 3.2. The objects
change \(F\) and the rate of the EVs near to SNs.

![Figure 3.9. A container with inlet and three SNs.](image)
Table 3.2. Initial conditions

<table>
<thead>
<tr>
<th></th>
<th>$T_0$ (°C)</th>
<th>$T_{\text{delay}}$</th>
<th>$H_0$ (%)</th>
<th>$H_{\text{delay}}$</th>
<th>$F_0$ (m/s)</th>
<th>$F_{\text{delay}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inlet</td>
<td>10</td>
<td>---</td>
<td>30</td>
<td>---</td>
<td>15</td>
<td>---</td>
</tr>
<tr>
<td>K1</td>
<td>9</td>
<td>5</td>
<td>28.5</td>
<td>7</td>
<td>13.5</td>
<td>2</td>
</tr>
<tr>
<td>K2</td>
<td>8.5</td>
<td>3</td>
<td>27</td>
<td>4</td>
<td>3</td>
<td>5</td>
</tr>
<tr>
<td>S1</td>
<td>8</td>
<td>8</td>
<td>25.5</td>
<td>2</td>
<td>10</td>
<td>8</td>
</tr>
</tbody>
</table>

In the above table, $(T_0, H_0, \text{and } F_0)$ and $(T_{\text{delay}}, H_{\text{delay}}\text{, and } F_{\text{delay}})$ are initial conditions and delay time of the EVs between inlet and the SNs, respectively.

While reducing $T$ in reefer unit, $T$ in the SNs decreases and because of its reverse effect on $H$, relative humidity increases. Changing the rate of $F$ changes the speed of the responses of simulated $T$ and $H$ and higher air flow makes their response faster. As illustrated in Figure 3.10, curves of K1, K2 and S1 are according with the data extracted from models introduced in equations (3.18) and red colored curves related to the inlet are the setpoints of the corresponding EVs. The relations of $T$, $H$ and interconnections are updated based on the amount of $F$ at the related instant of simulation.

Figure 3.10. Outputs when $T$, $H$ and $F$ in input change.
3.4 Floating Input Approach (FIA)

3.4.1 New topology of WSN and system identification based on FIA

The main proposal of this research for energy management of the WSN has been published over [12], [13], [14], [15] and [16]. As shown in Figure 3.11, as members of a cluster, there will be a few special Key Sensor Nodes (KSNs) which try to estimate the EVs for a Desired Sensor Node (DSN). Hence, there might be a same scenario inside each cluster of a large scale network. In a cluster, KSNs may also transmit direct commands to the related DSN.

A KSN might differ from a cluster head. It can be called either an estimator or predictor. In a cluster, several KSNs send the required information to the cluster head to be sent to the main processor (base station) according to the corresponding routing procedure. They might be distributed everywhere inside the container, even near the door or near to the inlet. If KSNs are located in some key points, mismatch error due to no considering unpredictable phenomenon will be avoided. That is because depending on the floating input approach, uncertainties and disturbances are considered indirectly as the input change. This method is independent of the ventilation system. We introduce here a semi autonomous topology in which some decisions can be made directly by the KSNs where the global decision might be made by the base station. The following tasks are devoted to KSNs:

1. They measure environmental variables periodically.
2. The KSNs stay in active mode during the normal operating mode.
3. They evaluate measured values and predict the EVs for the DSNs and update previous models after measuring and receiving some new data (this task is used in energy saving mode and fault detection).
4. Failed SNs can be considered as a target of prediction for KSNs (predictors).
5. They may take DSNs to sleep or deep sleep mode when the operational conditions are normal and there are no big changes in the EVs (this operating mode is used in energy saving mode and is discussed in chapter 5.4.1).
Based on the existing methods of modeling, variation in inlet changes the EVs of the DSNs. Instead, in accordant to Figure 3.11, a new topology based upon FIA simplifies modeling problem in an environmental system by using advantage of plurality of measuring points in a sensor network.

It assumes that each DSN (S1 or S2) is influenced only from surrounding KSNs and not from actual input of the system (cooling system input). The proposed model is considered like a linear Multi Input-Single Output (MISO) model of KSNs-DSN in each cluster. A cluster head can be among the KSNs as shown with white colored circles among the other sensor nodes.

![Figure 3.11. Connections between the KSNs (K1...Km) and each DSN (S1 or S2).](image)

Disturbance in the introduced environmental system might be applied either to the input, system or even to the output. Any way, it may impress some of the SNs. The highest excited KSNs are candidates to be inputs of a new MISO system which influence a DSN. On the other hand, every non-modeled disturbance is considered as input change, not a pure disturbance.

Unknown parameters of this MISO model are acquired using an identification method having noise-corrupted data of the KSNs-DSN.

According with [38], two nodes are said to be connected whether the distance between them is less than transmission range. Subsequently, if the behavior of EVs in the KSNs and DSN are close enough, MIMO model (3.18) of a KSN-DSN is converted to a set of SISO models of each EV. The selective KSNs during system normal operation is chosen considering the following points:

- Large number of data of KSNs-DSN, enough for estimation is necessary.
• Covariance matrix of measured EVs for KSNs-DSN should be computed.
• After sorting the Normalized Covariance (NC), the best estimators are those with bigger NC. They are also among the best predictors.
• Picking up the number of the estimators for each DSN in a cluster depends on the number of all SNs and processing capability of the KSNs and required accuracy.

Based on the FIA, there will be two approximate linear models written for any pair KSN-DSN in terms of \( T \) and \( H \) as illustrated in Figure 3.12 and Figure 3.13.

For a noise free system, transfer functions of SISO models can be written simply as:

\[
T_{DSN}(Z) = G_T(Z) \cdot T_{KSN} \quad (3.24)
\]

\[
H_{DSN}(Z) = G_H(Z) \cdot H_{KSN} \quad (3.25)
\]

For a MIMO model of KSNs-DSN written in (3.26), an optimized combination of SISO models in the domain of discrete time (\( Z \)) for \( T \) and \( H \) should be solved.

\[
\begin{bmatrix}
T_{DSN} \\
H_{DSN}
\end{bmatrix} = 
\begin{bmatrix}
\sum \alpha_i T_{(KSN_i \rightarrow DSN)} \\
\sum \beta_i H_{(KSN_i \rightarrow DSN)}
\end{bmatrix} 
\quad (3.26)
\]
(T_{DSN}, H_{DSN}) in the above relation are respectively values of T and H measured by DSN. The parameters α and β are coefficients for the effects of different KSNs participated in a cluster to estimate the EVs on the corresponding DSN (T_{(KSNi → DSN)} and H_{(KSNi → DSN)}). Figure 3.14 below illustrates the selective worst case for two KSNs (K1, K2) and a DSN (S1) during an actual measurement test. In real systems there exist fewer variations during longer periods of operations.

![Actual measured signals from three sensors (K1, K2, S1)](image)

**Figure 3.14. Measured T inside the container in three points (T_s = 150 s).**

The sensory data with big and fast variation belongs to K1 which is located near to the inlet. The other SNs have been located far from each other and also far from K1.

In the above measurement different dynamics and non-linearities between the measured data of different SNs is due to their relatively large distance. In practice it might also be made by a SN located inside a closed space box or even a wrong reading by a faulty SN. When a sensor node is in the sleep mode, it is shut down except a very low-power timer which is on, just to wake up that node at a later time. In this way, once
the KSNs are in the active mode and the DSN in the sleep mode or it is faulty, there will be a separate MISO model for both $T$ and $H$ devoted to pairs (K1-S1) and (K2-S1). Unknown parameters of these models can be determined using an off-line identification technique. Having new inputs in K1 or K2, new predictions of EVs in the DSN are computed. The high correlated parts of input-output data help to find fixed model which is consistent of all utilized data in the estimation phase. In the prediction part input data which are correlated with the past data helps to have better prediction. Figure 3.15 represents a flowchart of an off-line predictor based on FIA.

Figure 3.15. Flowchart for offered estimation technique.
The off-line identification can be start with data recording from KSNs and DSN. When the number of sensory data is enough to have good parameter identification, it start to make off-line model wit predefined structure. After that past data of KSNs-DSN as well as new data of KSNs together with the obtained model is utilized to predict the future of the DSN. By the above description, the prediction can be used in different approaches like energy management (refer to chapter 5) and estimation of correct values of faulty sensors. The following factors, influenced on the quality of estimation are discussed in the next sections:

1. Using different model structures such as Auto-regressive exogenous input methods (ARX, ARMAX), Output Error (OE), Box- Jenkins (BJ) and State Space (SS) to see the corresponding differences.
2. Using different number of data-samples in learning stage.
3. Investigation of different adaptation indexes
4. Using different number of KSNs and model.
5. Using relevant KSNs.
6. Using either online or offline predictions.

3.4.1.1 Overview of Linear System Identification

As mentioned in [32], the models in linear system identification are distinguished into parametric and non-parametric approaches. 

**Parametric models** describe the true process behavior exactly with a finite number of parameters. A typical example is a differential or difference equation model. The parameters have a direct relationship to physical quantities of the process. 

**Non-parametric models** like an impulse response need an infinite number of parameters to describe the system. Relatively small number of parameters, by using optimization methods according to some objectives is determined by parametric methods.

3.4.1.2 Different model structures

Two categories of linear models and their important types have been written in the following Table 3.3. In the below table stochastic input v(t) is white noise and A(q), B(q), C(q), D(q), and F(q) are polynomials of forward shift operator q with degrees na,
nb and nc, nd and nF, respectively where nk represents delay time of input signal u(t) and y(t) is the output.

| Equation error | ARX- Auto Regressive with eXternal input | \( y(k) = \frac{B(q)}{A(q)} \cdot u(k) + \frac{1}{A(q)} \cdot v(k) \) |
| Output error   | ARMAX- Auto Regressive Moving Average with eXternal input | \( y(k) = \frac{B(q)}{A(q)} \cdot u(k) + \frac{C(q)}{A(q)} \cdot v(k) \) |
|               | FIR- Finite Impulse Response           | \( y(k) = B(q) \cdot u(k) + v(k) \) |
|               | OE- Output Error model                 | \( y(k) = B(q) \cdot u(k) + v(k) \) |
|               | BJ- Box Jenkins                        | \( y(k) = \frac{B(q)}{F(q)} \cdot u(k) + \frac{C(q)}{D(q)} \cdot v(k) \) |

Figure 3.16 shows an Auto-Regressive Moving Average with eXternal input (ARMAX) model (3.27) and the polynomials in (3.28):

\[
A(q^{-1}) \cdot y(t) = B(q^{-1}) \cdot u(t - n_\text{b}) + C(q^{-1}) \cdot v(t) \tag{3.27}
\]

\[
\begin{bmatrix}
A(q^{-1}) \\
B(q^{-1}) \\
C(q^{-1})
\end{bmatrix} =
\begin{bmatrix}
1 + a_1 q^{-1} + \ldots + a_{n_\text{a}} q^{-n_\text{a}} \\
b_1 q^{-1} + \ldots + b_{n_\text{b}} q^{-n_\text{b}} \\
1 + c_1 q^{-1} + \ldots + c_{n_\text{c}} q^{-n_\text{c}}
\end{bmatrix} \tag{3.28}
\]

\[
y(t) = [b_1 u(t-1) + \ldots + b_{n_\text{b}} u(t-n_\text{b})] - [a_1 y(t-1) + \ldots + a_{n_\text{a}} y(t-n_\text{a})] + [v(t) + c_1 v(t-1) + c_{n_\text{c}} v(t-n_\text{c})] \tag{3.29}
\]
The ARMAX model differs than the other models in the sense that it contains the past error signals which makes it rather more flexible for modeling processes with noisy data or unmeasured disturbances. In other word, the ARMAX model is based on the ARX model plus the part of the moving average noise model. The most important application of a model is forecasting the future behavior of a process. Two cases are defined here:

**Simulation**- As illustrated in Figure 3.17, based on this method response of the model to an input sequence is calculated while the process outputs are unknown.

![Figure 3.17. Simulation based on a model.](image1)

**Prediction**- Accordant with this method, shown in Figure 3.18 the process outputs are known up to a time instant, say k-1, and it is asked for the model output l steps in the future. If the prediction horizon l becomes very large the importance of the information about the previous process output decreases and prediction approaches simulation.

![Figure 3.18. One-step a head predictor.](image2)
Both forecasting methods mentioned above will be described mathematically in detail in sections 4.5.1 and 4.5.2.

Assuming only one KSN (k1) as estimator and one DSN (S1) as the object of estimation, and having its actual measurement, we will attain different results using ARX, ARMAX, OE, BJ and SS methods in two separate experiments. Whereas order one can’t cause a good performance, a third order linear model is chosen and unknown parameters are obtained via different methods. The estimation results for OE and BJ are the same with ARMAX and have not shown in the following Figure 3.19. About 691 samples of measurement have been used in the estimation procedure for both learning and validation. It represents that the ARMAX method provides a better fit to actual measurement than ARX and State Space (SS) method. Due to flexibility of ARMAX structure in noise modeling as well as less computation in compare with Box-Jenkins structure, it can be a candidate to be studied during the next parts.

![Figure 3.19. Prediction using different estimation methods.](image-url)
3.4.1.3 Different data numbers

The very first question is that how many samples are enough for an accurate estimation? Having ARMAX method, Figure 3.20 depicts that increasing data number up to 500 provides better performance and increasing more than 500 samples up to 691 (whole data number) changes the quality less.

In the worst cases 70% of whole range of data horizon is enough to have an acceptable prediction in 30% of the rest, provided that enough data (in real cases usually more than 50 measurements) have already been recorded.

![Figure 3.20. Comparison of different data number used for model making.](image)

Apparently, reducing the number of utilized data degrades the performance of estimation. The estimated model can be used precisely to predict the EVs provided that the data utilized in the horizon of prediction consists of relatively similar variations of the learning section.

If the measurements have no big variations, model is not sensitive to the number of measured samples. On the other hand, one may use less number of sensory data and then
exploit that model to predict output, accurately. Nevertheless, existing big variations in required data, extra samples are needed whilst acquiring the model.

### 3.4.1.4 Different Fit indexes

The simplest way to estimate the EVs in a SN is finding the mean (average) of the corresponding variables from the surrounding SNs. It delivers reliable values not far from the other measurements. However, as illustrated in Figure 3.21 it is not a good estimation particularly when the data of KSNs are close and they are different in compare with those measured by a DSN. This method is useful when the normal estimation fails.

![Different Off-Line Estimation (SISO and MISO) with 300/429 samples](image)

**Figure 3.21. Off-Line estimation using 300/429 samples**

In [39], the procedure of locating a faulty sensor has been formalized statistically to compare the reading at a SN with those of its neighbors. It has used median instead of mean value of measurements because the sample mean cannot represent well the centre of a sample when some values of the sample are extreme. However, median is a robust estimator of the centre of a sample. It implies that whether the difference of reading of a
SN differs from median, it is faulty. Authors of [40] utilize different methods of interpolating the temperature data of intermediate positions to find a prediction of a desired position which has no sensor. They employ this method to minimize sensor number scattered in a closed space. Due to existing uncertainties in this nonlinear model, different fit indexes as defined in the following might candidate different estimators (KSNs). When a model is created, one should validate the estimates predicted by the model. There are several ways for validation of different models. In linear time invariant systems one can use these indexes to find the best estimators. A good measure that tests how well estimated data fits the outputs from the system is:

$$Fit\% = \frac{1 - \|\tilde{y} - \hat{y}\|}{\|y - \tilde{y}\|} \cdot 100$$  \hspace{1cm} (3.30)$$

Where, $\tilde{y}$ and $\hat{y}$ are the predicted value and mean of output, respectively. Another index, the covariance is a measure of how much the deviation of two or more variables or processes match.

For two signals $u$ and $y$, if they are not closely related (uncorrelated) the covariance will be small, otherwise if they are similar (correlated) the covariance will be large. It is defined as (3.31).

$$cov(U, Y) = \sigma_{uy} = E[(U - \bar{u})(Y - \bar{y})] = E(U - \bar{u})E(Y - \bar{y})$$ \hspace{1cm} (3.31)$$

For two signal vectors measured by $K1$ and $S1$, covariance matrix has four elements.

$$Covariance(K1, S1) = C = \begin{pmatrix} C_{ii} & C_{ij} \\ C_{ji} & C_{jj} \end{pmatrix}$$ \hspace{1cm} (3.32)$$

To compare different signals we compare normalized value of covariance matrix as follows:

$$NC = \text{Normalized Covariance} = \frac{C_{ij}}{\sqrt{C_{ii} \cdot C_{jj}}}$$ \hspace{1cm} (3.33)$$
When the correlation between the model error and the model input approach to zero: 
\[ \text{corr} \{u, e\} = 0. \]
AIC (Akaike Information Theoretic Criterion) and FPE (final prediction error) techniques defined in the following must become minima that the models order can be taken:

\[ \text{AIC} = \text{Log}(v) + 2 \cdot \frac{\text{dim}(\theta)}{N} \]

\[ v = \text{det} \left( \frac{1}{N} \sum_{1}^{N} \varepsilon \cdot (\varepsilon)^T \right) \]  \hspace{1cm} (3.34)

\[ \text{FPE} = V \cdot \frac{1 + \frac{\text{dim}(\theta)}{N}}{1 - \frac{\text{dim}(\theta)}{N}} \]  \hspace{1cm} (3.35)

\[ \text{SSE} = \sum_{1}^{N} (\text{error})^2 \]  \hspace{1cm} (3.36)

The parameter N is the number of estimation data and dim (θ) is the size of the unknown parameter vector.

### 3.4.1.5 Model order selection and number of KSNs

In order to have low computational overhead as well as involving simple numerical operations, we would like to reduce the number of the KSNs. But, simulations will illustrate that the accuracy will be increased with increasing the number of these estimators (KSNs). Figure 3.22 and Figure 3.23 represent estimation by using K1, K2 separately (in the form of SISO model) and together (in the form of MISO model) respectively for \( T \) and \( H \). In both figures, subplot (a) represents actual measurement from three sensor nodes (K1, K2 and S1). Based upon these, the estimations (utilizing K1) are compared with actual S1 with subplot (b) for low and high order models. About the same story is also given by K2 in subplot (c). Instead, subplot (d) draws estimation by using MISO model as well as average of K1 and K2. As figures represent, the measurement of \( T \) has less and slower variation than \( H \) and results better accuracy.

To cope with the nonlinearity of the system while making linear model, higher order transfer functions behaves more suitable. However, to be less time consuming and prevent over fitting problems and to make simplicity in implementation, model orders
more than three are not taken in this application. MISO models consider environment from different sides of a DSN and provide better fit and more robustness than SISO models. Nevertheless by using appropriate KSN, SISO model needs less calculation and still gives reasonable prediction. The evident difference of mean of \((K1, K2)\) with \(S1\) proves that this criterion (average) is not suitable to be used instead of estimation. However, very good fit from MISO and acceptable fit from SISO models guarantees the capability of the proposed identification method based on the FIA.

Figure 3.22. Comparing the result of prediction of Temperature \((T)\).
Figure 3.23 in below gives the same meaning. It also highlights the effect of the location of different KSNs on the performance of prediction.

Table 3.4 summarizes the results of the comparative study in case various conditions applied to estimators. It evaluates the effect of the number of data (100, 200, 300, 400 and 427 out of 427), low and high model orders (one and three), the mentioned indexes of fitting (NC, Fit% and SSE) candidate different estimators. More NC and FIT% and less SSE cause better estimation. Two columns in the left side of the table represent the assessment of the sensor reading whilst computing NC and SSE for different numbers of samples. For instance where we have NC(K1-S1) = -0.92 and NC(K2-S1) = 0.94, it means that K2-S1 is more correlated than K1-S1. Third column has been devoted to
average of measured K1 and K2 in compare with S1. Estimation results in the next
columns interpret the situation of estimation of S1, firstly by K1 and K2 in the form of a
SISO model and secondly (K1, K2) with MISO model. The red colored numbers in the
rows candidate the best estimators where the blue colored numbers represent higher
correlation between KSN and DSN. In this test K2-S1 is more correlated than K1-S1 as
highlighted in the second column (left). Mostly high order MISO model (KSNs-DSN)
gives better performance than the others. However, in a few cases estimation by using
high order K1-S1 is better than that by either K2-S1 or K1K2-S1 (estimation by K1 and
K2 as MISO form inputs).

The first reasonable result is that more correlated measured signal of K2-S1 than
K1-S1 gives also more correlated estimation of K2-S1 than K1-S1. In spite of this,
mostly using both K1 and K2 in the form of a MISO model to estimate S1 gives better
performance than SISO models by either K1 or K2 individually. It is because a MISO
model considers the effect of environment around of a DSN from different directions.
Increasing the number of estimators will increase covariance of the response.

Another notably result is that using average method has less covariance than
estimation by using either SISO or MISO model provided that in MISO model
appropriate KSN (K2) is chosen. To select either one or more KSNs provided that there

<table>
<thead>
<tr>
<th>Measurements</th>
<th>Estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Measured K1</td>
<td>Measured K2</td>
</tr>
<tr>
<td>-0.92</td>
<td>0.94</td>
</tr>
<tr>
<td>256.9</td>
<td>93.3</td>
</tr>
<tr>
<td>-0.41</td>
<td>0.93</td>
</tr>
<tr>
<td>413.7</td>
<td>148.7</td>
</tr>
<tr>
<td>0.30</td>
<td>0.97</td>
</tr>
<tr>
<td>456.2</td>
<td>162.9</td>
</tr>
<tr>
<td>0.28</td>
<td>0.97</td>
</tr>
<tr>
<td>548.3</td>
<td>193.0</td>
</tr>
<tr>
<td>0.32</td>
<td>0.97</td>
</tr>
<tr>
<td>560.7</td>
<td>195.7</td>
</tr>
</tbody>
</table>
are no additional conditions, a large number of data of KSNs-DSN, enough for estimation is required. Furthermore the KSNs should be sorted based on their normalized covariance. Picking up the number of the estimators for each DSN depends on the number of all KSNs and the DSNs and capability of the processor and required accuracy. MISO model is more robust than SISO model. However, with competent KSNs, an off-line SISO model needs less calculation and gives reasonable prediction. Hence, we recommend using either the MISO models or SISO model by using KSNs which have higher correlation with the representing DSN. The column so-called selected results, gives the amount of parameters such as NC, %FIT and SSE for different kinds of estimation. In a few cases in order to nonlinearity of the actual system the best estimation are acquired by low order MISO models. Another reason is that with less number of data used for estimation, model doesn’t contain the whole conditions and therefore cannot be used for prediction.

The obtained results show that higher order models cause better estimation (higher covariance). Therefore, K2 is prior to K1 to be predictor in this test.

3.4.1.6 Sensor position

The positions of SNs inside the container are not fixed forever and they may be located in different positions. As said before, models of $T$ and $H$ are decoupled whether appropriate KSNs are chosen as the estimators. Figure 3.24 below graphs the measured $H$ by three SNs (460 recorded samples). The curve with the less variation is related to a node far from the inlet, behind a fruit box. The first part of the curves till $t = 30$ minute is related to loading and turning-on the ventilation system and the last part ($t \geq 255$ min) shows permanent turning-off, opening the door and unloading the freight. Figure 3.25 represents off-line estimation results by K1, K2 and both. The K1, located near to inlet far from S1, makes poor performance where K2 near to S1 gives better estimation. Using both K1 and K2 in a MISO model gives acceptable prediction.

As a result, choosing the SNs in vicinity in a cluster provides better off-line estimation of a DSN by the KSNs. The KSNs and DSN near to each other form linear dynamic models. In such a case they are simply sorted based on an index like sum of squared error or covariance and they might be chosen with respect to their priorities. One possibility to find the members of a cluster is using their signal strength rate.
CHAPTER 3. Environmental model making

Actual measured signals from three sensors (K1, K2, S1)

Figure 3.24. Measured signals by three sensor nodes (K1, K2 and S1)

Different Off-Line Estimation (SISO and MISO) with 300/429 samples

Figure 3.25. Comparing the result of prediction of R. Humidity (H) while two estimators near and far from S1 are chosen.
A theoretical drawback of FIA arises for systems containing big delays in the sense that there might be a pair of KSN-DSN, where the DSN (output) is prior to KSN (input). According to Figure 3.26, this happens because in a practical case there is no information about the location of the SNs and no simple rule to choose a KSN which its reading is earlier than a DSN in the same cluster. In such cases there will be a non casual system and therefore a poor model is identified.

Using data just from a KSN-DSN to create single input-single output (SISO) model cannot fully present surrounding influences. It can only interpret variations of the EVs on a DSN from side of that specific KSN where the EVs from side of the other KSNs make separate influences as well.

Figure 3.26 exemplifies a three dimensional space in which K1 and K2 are considered more operative than K3 due to flow direction. In fact, prediction using multi input-single output model (MISO) causes better accuracy than SISO models, provided that corresponding KSNs are prior than the DSN. As it will be shown in the next parts, the more effective KSNs, the better performance in estimation and prediction gives. That is because once a fault in a KSN occurs, other KSNs continue to estimate. The KSNs which do not influence on the DSN might not increase the accuracy. They are among the SNs either very far from DSN or in a location with different nature in order that they have no much correlation with that DSN.

![Impression of the KSNs on a DSN](image)

Using several KSNs instead of one, considers the effects of the EVs from different sides around a DSN. However, chaotic direction of the $F$ and need to detect the flow direction by expensive sensors causes it impractical. Therefore, there will be a mismatch error clearly due to considering non effective KSNs in the estimation process.
3.5 Summary and conclusion chapter 3

First of all a hybrid thermodynamic model was made for modeling environmental variables, temperature ($T$), relative humidity ($H$), and air flow ($F$) inside a closed space container. To achieve this approximate model, some thermodynamical relations as well as experimental results were employed. After that a novel method, Floating Input Approach (FIA) was introduced to simplify the hybrid model. Further on, topology of the network, new nomenclatures and the scenario were quoted.

The FIA was studied in detail and various identification methods in addition to several alternatives influenced on the performance of identification were also investigated. The preference of the FIA in compare with the usual methods which employ model of inlet-SN was addressed as well.

Among different model structures, ARMAX model selected and a MISO model was introduced for each environmental variable where surrounding key sensor nodes (KSNs) was considered as inputs and a desired one as output. Because of flexibility of the proposed techniques and decoupling between the variables, models can be used in different SISO or MISO configurations for each variable.

Analysis of the results showed that MISO model acts better than SISO whereas it considers environment from different sides, employing different KSNs instead of one direction by one KSN. In addition to model structure, there were some other peripheral factors influenced on the quality of estimations such as location of the estimators, the number of estimators and the number of utilized sensory data which were fully investigated. The selective identification techniques based on the FIA will be applied for fault-detection and energy management in the next two chapters.
4. Fault detection and diagnosis based on FIA

4.1 Introduction

Fault diagnosis has been received great attention since the early 1970s. A large variety of methodologies have already been studied and developed. One of the most complicate features in this field is related to time varying, multivariable and nonlinear systems. Such case dependent problems cannot be solved straightforward as the problem defined in this research is. The present chapter reports the result of implementing a reliable monitoring system by using a WSN for an intelligent container which is a part of a big supply chain network. Some modern methods of system identification and fault diagnosis have been applied and successfully implemented by using a high performance wireless Imote2 module.

As the most noteworthy references, [18] has surveyed the most often methods of detecting sudden changes in stochastic linear systems. In [19], a review of model-based fault diagnosis techniques from basic principles to the properties and limitations of several methods has been treated. It introduces a two-stage structure of model-based fault diagnosis, first stage to produce residual generation and the second for decision making based on the residuals. The authors of [20] give some assumptions for faults to be detected by model-based methods and this method is argued in detail by [21].

There are some methods of signal processing, very often among the traditional fault detection and diagnosis (FDD) methods like fault dictionary approach, limit checking and plausibility tests and also parallel redundancy as mentioned in [22]. Implementing these methods which compare system variables with pre-set limits showed that they are
simple and reliable. However the variables may change in accordance with different operating points and those limits might not be valid forever. Afterward, available measurements can be compared with a priori information indicated by a model of the system by generating residuals. We replaced an estimation of measured value of a faulty SN by using other SNs and we applied our recently introduced Floating Input Approach (FIA). It gives a quantitative model as treated in detail in chapter 3.4.

4.2 Faults in the system

Secure data collected by a monitoring system plays the main role in a logistic process and related decision making. According with definitions of faults in [22], a fault is an unpermitted deviation of at least one characteristic feature of the system from the acceptable, usual, standard condition. It might initiate a failure or a malfunction. Failure is a permanent interruption of a system’s ability to perform a required function under specified operating conditions where malfunction is a temporary interruption of a system’s function. Faults may cause three types of failures: **Hard failures** which are step changes in the sensor outputs (stepwise). **Soft failures** which are small step changes and/or drifts, incipient (drift wise). They might also be **intermittent** (with interrupts). We define here two sources of faults in the system under discussion:

1. **Faults in the environment-** Due to behaviors which do not correspond to the specification. In case of either opening the door (if very big difference between weather inside and outside of container exists) or abnormal operating of the cooling system, cargos become more or less influenced. Then after investigating the monitored EVs, related warnings are released in case exceeding defined borders.

2. **Faults in monitoring devices-** system operates normally, but wrong measurements lead it to make wrong decision. Such faults are mainly caused by:

   - Low battery power, low signal strength or mechanical-electrical faults.
   - Network connection failure.
   - Internal unexpected failure in the SNs before they run out of battery.
   - Application failure such as software failure and bugs.
4.3 Measurements used for FDD

A field test was performed on a truck in the University of Bremen on 26.08.2008 (shown in Figure 4.1). The truck was equipped with up to 20 data loggers (*iButton*) to measure the EVs. They were mounted at the walls, top, bottom, inside a closed box, and outside the truck. Some of the data loggers (F1…F13) measured both $T$ and $H$ and the rest (B1…B45) only measured $T$. Several boxes were loaded inside the truck to be located against the normal air stream like that in a real environmental system.

![Figure 4.1. Truck under test to measure $T$ and $H$ by Data loggers (*iButton*)](image)

Table 4.1 represents the location of the data loggers. Measured $T$ and $H$ outside during the test were about 20 °C and 80%, respectively.

<table>
<thead>
<tr>
<th>Sensor locations during the measurement test</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T$ (°C) &amp; $H$ (%)</td>
</tr>
<tr>
<td>Inside</td>
</tr>
<tr>
<td>-------</td>
</tr>
<tr>
<td>X</td>
</tr>
<tr>
<td>Y</td>
</tr>
<tr>
<td>Z</td>
</tr>
</tbody>
</table>

Figure 4.2 (a) illustrates the variation of setpoint of $T$ with blue color as well as the periods that door was manually opened (green color). Figure 4.2 (b) and (c) show measured $T$ and $H$ by sensors. After the first closing door at $t$=30 min, cooling system
tries to decrease $T$. Based upon the adjusted setpoint, reefer unit in the truck provides only desired $T$ where both $F$ and $H$ are dependent variables. Obstacles against $F$ make it chaotic and unpredictable. There are different initial conditions for different data loggers because of their positions or even corresponding measurement errors. The figure represents that the measurements are directly influenced by input variations as well as disturbances such as opening the door. Maximum differences in measurements appear between the points near to door and inlet while door is open (see the difference between $F_1$ and $F_{11}$ in subplots b and c). System follows predefined setpoint by automatically turning on and off. To have the effect of disturbance during the measurement, door was opened for unequal durations (for one minute at $t=150$, $330$ min, and two minutes at $t=220$ min) which in turn increased $H$ and also absolute humidity.

![Figure 4.2. (a) Setpoint and door position. (b) Actual measured $T$ by different SNs. (c) Relative humidity.](image)

It is because during the test, weather outside the truck was more humid than inside. The measurements with the least variations are related to $F_{12}$ (located inside a box) and
F13 (far from inlet). Only the SN near the reefer unit (F1) has different rhythm of variations in compare with the others. The minimum and the fastest measure of $T$ and also the maximum $H$ belong to F1. The minimum $H$ is recorded in the bottom of the truck. The last part of the curves depicts permanent turning off ($t \geq 380$ min) and permanent opening the door at $t = 420$ min.

4.4 Fault detection mechanisms

As noted in [22] fault detection can be established based upon signal and process models. We have categorized different techniques in the following. It is noted that most of the offered methods have been used and developed in this research and the most appropriate methods have been implemented on Imote2 platform.

- Limit checking - Trend checking of single signals
  - Fixed threshold
  - Adaptive threshold
  - Change detection
- Single or multiple-signal models
  - Correlation
  - Spectrum analysis
  - Wavelet analysis
- Process models and fault modeling
  - Parameter estimation
  - Neural networks
  - State observer
  - State estimation
  - Parity equation
- Multi variant data analysis
  - Principal component

A signal model-based method in the system is shown by Figure 4.3. It uses measured signals and evaluates them having a predefined normal behavior of the system. The
CHAPTER 4. Fault detection and diagnosis based on FIA

feature generation block prepares measured signals or their trends to be compared with normal values to show whether they exceed permissible values.

![Figure 4.3. Basic scheme of fault detection with signal models](image)

**4.4.1 Limit and Trend checking of single signals**

In this method, direct measured variable \( y(t) \) is compared with the permissible limits if their absolute values or trends exceed certain thresholds. Checking their plausibility could be also further possibility. Corresponding with [22], one may use either limit or trend checking for fault detection. A combination of limit checking for absolute values and trends can also be considered as (4.1) to (4.7).

\[
\begin{align*}
    y_{\text{min}} & < y(t) < y_{\text{max}} \\
    \dot{y}_{\text{min}} & < \dot{y}(t) < \dot{y}_{\text{max}}
\end{align*}
\]

where \((y_{\text{min}}, y_{\text{max}})\) and \((\dot{y}_{\text{min}}, \dot{y}_{\text{max}})\) in these inequalities are minimum and maximum of permissible variables and their trends, respectively. These limits are assumed as known...
initial data in all SNs. For the first step two sets of limitations for $T$ as well as $H$ are defined as follows:

$$T_{\text{min}} < T(t) < T_{\text{max}}, \quad H_{\text{min}} < H(t) < H_{\text{max}}$$ \hspace{1cm} (4.3)

$$\dot{T}_{\text{min}} < \dot{T}(t) < \dot{T}_{\text{max}}, \quad \dot{H}_{\text{min}} < \dot{H}(t) < \dot{H}_{\text{max}}$$ \hspace{1cm} (4.4)

$$\dot{T}(t) = \frac{\Delta T}{T_s}, \quad \dot{H}(t) = \frac{\Delta H}{T_s}; \quad \text{where } T_s = \text{Sensor sample time}$$ \hspace{1cm} (4.5)

For example:

$$T_{\text{min}} = -50^\circ C, \quad T_{\text{max}} = 60^\circ C$$ \hspace{1cm} (4.6)

$$H_{\text{min}} = 0\%, \quad H_{\text{max}} = 100\%$$ \hspace{1cm} (4.7)

For a container equipped with continuous $T$ control system, the fastest $T$ or $H$ can be measured by the SNs near the door. However, with on-off temperature control, the fastest response is found either near the door or near the inlet because the space between these two points acts like a low pass filter against the variations. The measured values out of the above limits are represented immediately as fault on the SN.

Specifying the accurate values of permissible trends is not simple in general. They depend on several factors such as the ventilation system, internal space of the container and desired sensitivity of FDD mechanism. Therefore, in the noisy, time variant system which is influenced by both disturbances and operating of automatic cooling system, trend limits should be chosen as big as possible such that system issues not wrong fault alarms. In this way limit-trend checking methods can only perform conservative indexes to detect big and abrupt faults and there are still several faults that cannot be detected. The next section goes one step ahead and develops the limit-trend checking methods to be more applicable in the system dealing with disturbances.
4.4.2 Development of limit-trend checking methods, two-level limitations

There is seldom a sharp borderer between normal and faulty states. Then, as represented in Figure 4.4, defining two-level limitations of output and trends, there will be three areas:

I. Green - normal operation area
II. Yellow - suspect status area
III. Red - explicit fault area

To decide about output or trend values in yellow area the auxiliary variables (multiple signals) are required. A measured parameter of DSN (single signal) which exceeds limitations of $y_{max2}$, $y_{min2}$, $\hat{y}_{max2}$ or $\hat{y}_{min2}$ is shown in Figure 4.4 as faults such as F-1, F-2…F-6.

Checking some physical laws under which a process component works, shows if a loss of the plausibility has been occurred. It provides us with the information about that fault. Some logical rules are needed based on our expectation of the system parameters. These rules of the measurement allow rough description of the expected behavior of the process under normal conditions.

If these rules are not satisfied either the process or the measurements are faulty. Any negative measured value of $H$ represents a faulty measurement. The performance of fault detection by the two-level limitation method is better than the usual limit-checking or trend-checking methods and faults on $T$ or $H$ or both are detected carefully.

![Figure 4.4. Multiple regions of operations and possible fault areas](image-url)
4.4.3 Multiple-signal models

As mentioned in the previous section, by using single signal methods, conservative limits can detect some of big and abrupt faults. Regarding to different areas of operation shown already in Figure 4.4, there are still several fault types which don’t lead to exceeding the predefined limitations although they might be among the most probable sensor faults. The following signal modeling methods employ auxiliary measurements:

1. Considering measurements of a specific EV from a pair KSN-DSN.
   - Measurements of a DSN stay at zero for a long time (more than five steps), Figure 4.5 (a).
   - Both KSN-DSN measure zero for a long time (it might be because of disconnecting measurement part of the SNs), Figure 4.5 (b).
   - Measurements from DSN stay at a fix value for a long time, Figure 4.5 (c). This situation should be separated from which the EVs are approximately fixed in stable states. In this case, using another variable of the EV is helpful.
   - Measured value of a DSN jumps fast and stays at a fix value, Figure 4.5 (d).
   - Using difference of a similar measured variable of a pair KSN-DSN.

2. Considering various measured variables of the specific SN.

The above procedure detects only single faults (one fault at the moment) not multiple faults. To detect multiple simultaneous faults, measured values of several KSNs together with respective DSN in each cluster should be considered. Figure 4.5 illustrates above rules by using measurements of a pair KSN-DSN. It presents the situation under which faults (F-9, F-11, F-13, F-15 or F-17) arise. A few extra faults such as F-10, F-12, F-14, F-16 and F-18 also represent those permanent faults which have been started already or they occur after a different kind of faults. It becomes clear that according with the proposal of this research, after detecting a fault on a DSN, other KSNs can estimate supposed measured values of the faulty SN by either SISO or MISO models.

Abrupt variation on both KSN-DSN depicts a disturbance like opening door or existence of an influencing cold, hot or wet freight to the vicinity of them. Instead, no variation of $T$ besides big variation of $H$ measured by the same SN represents a faulty
sensor. Additional embedded sensors such as flow-sensors help distinguishing sensor fault from disturbance.

To detect additional faults and their duration, different models of more often faults have been illustrated in Table 4.2. To distinguish faults in $T$-sensor from those in $H$-sensor of each SN, fault recognition number for $T$ will differ with that for $H$ by a coefficient (shown in second and third columns). Exceeding high limit of measured values of $T$ or $H$, FDD system releases fault-1 or fault-100 respectively. A common delay between KSN-DSN may provide a big difference and is considered in the rules of signal model-based methods.
Above signal model-based techniques detect most fault types in the system and sensors. However, some faults can only be detected by Process model-based methods which are exhausted in the next section. Figure 4.6 draws fault detection while choosing KSN-DSN with relatively similar behavior in the middle of the container.

<table>
<thead>
<tr>
<th>Fault type</th>
<th>Detection rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 (No Fault)</td>
<td>Fault(t&lt;5 * T_s) = 0 or if no fault is being occurred, Fault(t) = 0 or Back to normal mode from a faulty condition: y(t−1) − y(t−2) = 0 and y(t) &gt; Y_{max1} or y(t−1) = y(t−2) = 0 and y(t) &lt; Y_{min1}</td>
</tr>
</tbody>
</table>

### Faults in the environment

<table>
<thead>
<tr>
<th>Single Signal (SS)</th>
<th>Detection rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>1, 2, 3, 4</td>
<td>y(t) &gt; Y_{max2}, y(t) &lt; Y_{min2}</td>
</tr>
<tr>
<td>5, 6</td>
<td>y(t) ≥ Y_{max2} &amp; y(t) ≥ Y_{max2}</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Multiple Signal (MS)</th>
<th>Detection rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>7</td>
<td>Y_{max1} ≤ y(t) ≤ Y_{max2} &amp; 0 ≤ 0(t) ≤ Y_{max1}</td>
</tr>
<tr>
<td>8</td>
<td>Y_{min2} ≤ y(t) ≤ Y_{min1} &amp; 0 ≤ 0(t) ≤ 0</td>
</tr>
</tbody>
</table>

### Internal faults of sensors

<table>
<thead>
<tr>
<th>Multiple Signal (MS)</th>
<th>Detection rule</th>
</tr>
</thead>
<tbody>
<tr>
<td>9</td>
<td>y(t)=0 &amp; y(t+1)=0 for DSN (Long time stay at 0), but KSN has not been staying at 0 provided that there was already no fault detected.</td>
</tr>
<tr>
<td>10</td>
<td>If a fault has already been detected in addition to case 9.</td>
</tr>
<tr>
<td>11</td>
<td>DSN &amp; KSN have been staying at Zero for Long time provided that there was already no detected fault.</td>
</tr>
<tr>
<td>12</td>
<td>If a fault has already been detected in addition to case 11.</td>
</tr>
<tr>
<td>13</td>
<td>y(t=0,1,2,3,4,5)=0, y(t)=0 for DSN (Long time stay at Fix value),</td>
</tr>
<tr>
<td>14</td>
<td>If a fault has already been detected in addition to case 13.</td>
</tr>
<tr>
<td>15</td>
<td>There has had a variation (y(t−1) ≥ Y_{max1}) and then parameters are fixed to a value not equal with 0 provided that there was not already a detected fault.</td>
</tr>
<tr>
<td>16</td>
<td>If a fault has already been detected in addition to case 15.</td>
</tr>
<tr>
<td>17</td>
<td>First, there is a variation (y(t−1) &lt; Y_{max1}) and then parameters are fix to a value not equal with 0 provided that there was not already a detected fault.</td>
</tr>
<tr>
<td>18</td>
<td>If a fault has already been detected in addition to case 17.</td>
</tr>
<tr>
<td>Single Signal (SS)</td>
<td>19, 1900 No signal received.</td>
</tr>
</tbody>
</table>

Based on the experimental setup shown in Figure 4.1, subplot (a) gives measured values by a KSN (F10) and a DSN (F11) and several faults on DSN have been highlighted by ovals. Subplot (b) shows |y(t)| and y(t) where subplot (c) represents results of fault detection by limit-trend checking method. It detects different fault types according with Table 4.2. Also Figure 4.7 illustrates fault detection whilst choosing another KSN-DSN (F1-F11), far from each other with bigger difference in compare with previous pair (F10-F11).
Figure 4.6. Detecting faults for two near H-sensor

Figure 4.7. Detecting faults for two far SNs which measure H
It represents that more fault types on F11 can be detected by F1 in compare with F10 and Fault_900, 1000, 400, 1300, 1400 and 1500 are issued in respective events. 

Comparing two subplots (c) in above two examples depicts different performance of fault detectors for various KSNs and that, neither drift wise faults nor stay at zero (0<t<30 and 50<t<70) and staying at a fix value (150<t<180 and 250<t<300) is detected by limit-trend checking method.

By using the results of surveys in [22], characteristics of parity equation, state estimation and parameter estimation in linear time invariant systems have been brought in Table 4.3. Obviously, parameter estimation is more applicable than other techniques because of advantages, highlighted by blue color.

**Table 4.3. Qualitative comparison of properties of fault detection methods for linear processes.**

<table>
<thead>
<tr>
<th>Criterion</th>
<th>Parity equations</th>
<th>State estimation</th>
<th>Parameter estimation</th>
</tr>
</thead>
<tbody>
<tr>
<td>assumption</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Model structure</td>
<td>Exactly known</td>
<td>Exactly known</td>
<td>Known</td>
</tr>
<tr>
<td>Model parameters</td>
<td>Known, constant</td>
<td>Known, constant</td>
<td>Unknown, time varying</td>
</tr>
<tr>
<td>Disturbance Models for</td>
<td>Exactly known</td>
<td>Exactly known</td>
<td>Exactly known</td>
</tr>
<tr>
<td>unknown inputs</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Noise</td>
<td>small</td>
<td>small</td>
<td>medium</td>
</tr>
<tr>
<td>Stability of detection</td>
<td>No problem</td>
<td>Depends on design</td>
<td>No problem</td>
</tr>
<tr>
<td>scheme</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Excitation by the input</td>
<td>Additive faults: no multiplicative faults: yes</td>
<td>Additive faults: no multiplicative faults: yes</td>
<td>Additive faults: no multiplicative faults: yes</td>
</tr>
</tbody>
</table>

**Detectable faults**

<table>
<thead>
<tr>
<th></th>
<th>Abrupt, Drift, Incipient</th>
<th>Single faults</th>
<th>Multiple faults</th>
<th>Fault isolation</th>
<th>Additive</th>
<th>Multiplicative</th>
<th>Robustness parameter changes</th>
<th>Nonlinear processes</th>
<th>Static processes</th>
<th>Computational effort</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>yes</td>
<td>yes</td>
<td>yes</td>
<td>SISO: no; MIMO: yes</td>
<td>SISO: yes</td>
<td>yes</td>
<td>No</td>
<td>problematic</td>
<td>Many classes possible</td>
<td>yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>SISO: no; MIMO: yes</td>
<td>SISO: yes</td>
<td>yes</td>
<td>No</td>
<td>problematic</td>
<td>limited</td>
<td>no</td>
</tr>
<tr>
<td></td>
<td></td>
<td>yes</td>
<td>yes</td>
<td>SISO: yes; MIMO: yes</td>
<td>SISO: yes</td>
<td>yes</td>
<td>No</td>
<td>problematic</td>
<td>Many classes possible</td>
<td>yes</td>
</tr>
</tbody>
</table>

Furthermore, for time-variant systems below Table 4.4 gives more details in terms of permissible disturbance and reachable accuracy where TVS, MIMO and NLS mention to time variant, multi input multi output and nonlinear systems respectively.
We use parametric methods because we have already assigned a fix structure for our model. Therefore, parametric estimation method is more suitable than the other methods for time variant systems. It provides very good accuracy and large disturbances are allowed. Neural net techniques are not appropriate for time varying system identification although they handle small and medium disturbances.

### 4.5 Process model-based methods

The process model-based fault detection methods are designed to detect any discrepancy between real system and model behaviors. The residual signal should be related to a fault. Nevertheless the same difference signal can respond to model mismatch or noise in real measurements, which is detected as a fault. Then designer must design some robust methods to cope with that. In this way particular attention is paid to the discrimination between actual faults and errors due to model mismatch.

Figure 4.8 represents a scheme of time invariant process model-based fault detection. Usual FDD methods extract special features such as model parameters and state variables which should be compared with normal values to produce corresponding residuals. Finally, change detection methods will generate analytical symptoms.
Referring to 4.3, on the basis of FIA, above scheme is altered to a new one illustrated in below Figure 4.9. It employs internal variables instead of common input-output of system. The measured values from process in this figure are gathered by different key sensor nodes (inputs), surrounding a desired sensor node (output).

![Basic scheme of fault detection with process models](image)

**Figure 4.8. Basic scheme of fault detection with process models**

![Basic scheme of fault detection based on process models-based by FIA](image)

**Figure 4.9. Basic scheme of fault detection based on process models-based by FIA**
The next two sections apply different identification methods to find residuals utilized in fault detection.

### 4.5.1 Output residual generation and evaluation using off-line models

An identification technique is said to be of the off-line type when a large number of input-output data is required to be stored. In this way according to a prescribed cost function the best fit of parameters are estimated. Black-box model for a linear time invariant system can be achieved by an off-line identification method during a fault-free period. As a result of chapter 3, An ARMAX model structure is chosen with input delay time (nk) as follows:

$$\begin{align*}
A(q^{-1}), y(k) &= B(q^{-1})u(k-nk) + C(q^{-1})e(k) \\
A(q^{-1}) &= \begin{bmatrix} 1 + a_1q^{-1} + \ldots + a_{na}q^{-na} \\ b_1 + b_2q^{-1} + \ldots + b_{nb}q^{-nb+1} \\ 1 + c_1q^{-1} + \ldots + c_{nc}q^{-nc} \end{bmatrix} \\
B(q^{-1}) &= \begin{bmatrix} 1 + a_1q^{-1} + \ldots + a_{na}q^{-na} \\ b_1 + b_2q^{-1} + \ldots + b_{nb}q^{-nb+1} \\ 1 + c_1q^{-1} + \ldots + c_{nc}q^{-nc} \end{bmatrix} \\
C(q^{-1}) &= \begin{bmatrix} 1 + a_1q^{-1} + \ldots + a_{na}q^{-na} \\ b_1 + b_2q^{-1} + \ldots + b_{nb}q^{-nb+1} \\ 1 + c_1q^{-1} + \ldots + c_{nc}q^{-nc} \end{bmatrix}
\end{align*}$$

(4.8)

Based on the above relation (4.8), model output $y(k)$ at the time instant $k$ is obtained by a combination of present values of input $u(k)$ and noise $e(k)$ and past values of output, input and noise. Whereas we don’t know about measurement noise at time instant $k$, we name it prediction error which can be considered as $(e = y_{\text{actual}} - y_{\text{prediction}})$ as follows:

$$e_{\text{ARMAX}}(k) = \frac{A(q^{-1})}{C(q^{-1})} y(k) - \frac{B(q^{-1})}{C(q^{-1})} u(k-nk)$$

(4.10)

Represented in (4.10), the prediction error of an ARMAX model is nonlinear in its parameters because of the filtering with $1/C(q^{-1})$ although we may express it in a pseudo-linear form as follows:

$$e_{\text{ARMAX}}(k) = -a_1 \cdot y(k-1) + \ldots + y - a_{na} \cdot y(k-na) - b_1 \cdot u(k-1) - \ldots - b_{nb} \cdot y(k-nb) - c_1 \cdot u(k-1) - \ldots - c_{nc} \cdot e(k-nc)$$

(4.11)
Since prediction errors of the ARMAX model ($e(k)$) before modeling is not available, the best way to estimate unknown parameters ($a_i$, $b_i$ and $c_i$) is employing multistage least squares which is sometimes called Extended Least Square (ELS). According to [32], this technique contains three following steps:

1) Estimate an ARX model by using existing input-output data having $(k \geq na + nb + nc + nk)$. Having measured values of $u$ (input), $y$ (output) for an ARX model $\varphi$ (regression vector) and $\psi$ (vector of $\varphi$ in different time instant) is then:

$$
\Psi = \begin{bmatrix}
\varphi_0^T \\
\varphi_1^T \\
\vdots \\
\varphi_k^T
\end{bmatrix} = 
\begin{bmatrix}
- y_{-1} & \ldots & - y_{-na} & u_{0-nk} & \ldots & u_{1-nb-nk} \\
- y_{0} & \ldots & - y_{1-na} & u_{1-nk} & \ldots & u_{2-nb-nk} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
- y_{k-1} & \ldots & - y_{k-na} & u_{k-nk} & \ldots & u_{k+1-nb-nk}
\end{bmatrix}
$$  \hspace{1cm} (4.12)

$$
y = \begin{bmatrix}
y_0 \\
y_1 \\
\vdots \\
y_t
\end{bmatrix}
$$  \hspace{1cm} (4.13)

$$
SS = \sum_{i=m}^{t} (y_i - \varphi_i^T \theta)^2, \quad t \geq na + nb + nk
$$  \hspace{1cm} (4.14)

To minimize the sum of squared error as the performance index (4.14) as mentioned in [17] we should have:

$$
\theta = [a_1 \ldots a_{na} \quad b_1 \ldots b_{nb}]^T = (\Psi^T \Psi)^{-1} \Psi^T \bullet y
$$  \hspace{1cm} (4.15)

2) Calculate the prediction errors of this ARX model as follows:

$$
e_{\text{ARX}}(k) = \hat{A}(q^{-1})y(k) - \hat{B}(q^{-1})u(k-nk)
$$  \hspace{1cm} (4.16)
3) By approximating the ARMAX model residuals as $\varepsilon_{ARMAX}(k-i) \sim e_{ARX}(k-i)$, the new parameters of ARMAX model is computed after above time $k$. For $t > k$ the new $\varepsilon(t)$ can be calculated by using the new ARMAX model as 
$$
\varepsilon(t) = y - \hat{y}, \theta
$$

$$
\varphi^T(t) = [-y(t-1) \ldots -y(t-na) \ u(t-1) \ldots u(t-nb) \ \varepsilon(t-1) \ldots \varepsilon(t-nc)]
$$

$$
\Psi = \begin{bmatrix}
-\varphi^T_0 \\
-\varphi^T_1 \\
\vdots \\
-\varphi^T_t \\
\end{bmatrix} = 
\begin{bmatrix}
- y_{-1} & \ldots & - y_{-na} & u_{0-nk} & \ldots & u_{1-nb-nk} & \varepsilon_{-1} & \ldots & \varepsilon_{-nc} \\
- y_{0} & \ldots & - y_{1-na} & u_{1-nk} & \ldots & u_{2-nb-nk} & \varepsilon_{0} & \ldots & \varepsilon_{1-nc} \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
- y_{t-1} & \ldots & - y_{t-na} & u_{t-nk} & \ldots & u_{t+1-nb-nk} & \varepsilon_{t-1} & \ldots & \varepsilon_{t-nc}
\end{bmatrix}
$$

$$
\theta^T = [a_1 \ldots a_{na} \ b_1 \ldots b_{nb} \ c_1 \ldots c_{nc}] \quad (4.19)
$$

$$
\theta = (\Psi^T \ast \Psi)^{-1} \ast \Psi^T \ast y \quad (4.20)
$$

$$
\hat{y} = \Psi \ast \theta \quad (4.21)
$$

It is worth noting however that the off-line ARX model has been utilized in energy management where the developed ARMAX model has been used in off-line fault detection to compare with on-line method. Model-based FDD utilizes process models which do not fully agree with real process due to model uncertainties and generated residuals deviate from zero even in fault-free cases. Then, such process model-based methods are useful just in case existing significant and abrupt faults not drift wise or small faults. In addition to actual measured signals due to two KSNs and a DSN, Figure 4.10 graphs estimation of the DSN by using different KSNs and also corresponding residuals. For estimating unknown parameters of ARMAX models of K1-S1 and K2-S1 to be used in residual generation, 300 data-samples were used. Furthermore, two faults on the DSN, stay at zero (300<t<320) as well as stay at fix value (350<t<400) were applied. Subplot (c) illustrates that the first fault provides a detectable big residual
where the second one provides about the same residual in compare with non faulty status which cannot be detected.

One of the advantages of the FIA is that disturbances in vicinity of SNs can be detected. Figure 4.11 (a) illustrates measured values and setpoint of $T$, when a disturbance is applied in the vicinity of K1 and S1 at $t= 25000$ second without influence on K2. The detectability of this disturbance is represented in the following by using regular model of inlet-SN in compare with FIA method. First of all an ARMAX model is acquired under normal operating mode before disturbance acts. Indicated in Figure 4.11 (b), estimation of S1 due to regular model of inlet-S1 can’t show this influence because the disturbance is not affected on inlet (input). Instead, FIA reflects it while estimating S1 because at least one KSN (input) senses it.
4.5.2 Output residual generation and evaluation using on-line models

On-line fault detection in a WSN is of permanent importance due to safety related factors. It needs on-line identification method which satisfies the following conditions: (1) It doesn’t need a special input. (2) All the data need not be recorded. (3) After each sampling instant a recursive algorithm adjusts the estimates. (4) Computation for model adjustment is a fraction of the sampling period.

To predict next steps of output and new parameters of the model, a few steps a head predictor is needed. The last input-output steps in addition to current data of the input (KSN) are used to predict future of the DSN in advance. This procedure gives a predicted model which is valid for the corresponding moment and can be utilized for generating residuals as well. To cope with time-varying systems in on-line identification methods, recursive algorithms might be applied. It is eligible to be used in short horizon predictions and fault diagnosis of time variant systems. Unknown parameters of the last model are determined by the data of input-output from a few steps ago and the parameters will be updated step by step. Starting from an ARMAX model we have:
\begin{equation}
A(q^{-1})y(t) = B(q^{-1})u(t-nk) + C(q^{-1})e(t)
\end{equation}

\begin{equation}
y(t) = \varphi^{T}(t)\theta(t) + e(t) \quad , \quad \hat{A}(q^{-1})y(t) = \hat{B}(q^{-1})u(t) + \varepsilon(t)
\end{equation}

\begin{equation}
\varphi^{T}(t-1) = [-y(t-1) \ldots -y(t-na) \quad u(t-1) \ldots u(t-nb)\nonumber \\
\varepsilon(t-1) \ldots \varepsilon(t-n\epsilon)]
\end{equation}

\begin{equation}
SS = V(\theta, y, \hat{y}) = \sum_{i=m}^{t} \hat{\lambda}^{t-i} (y_{i} - \varphi^{T} \theta_{i}^{2}) \quad , \quad t \geq m \quad , \quad 0 < \lambda \leq 1
\end{equation}

\begin{equation}
\theta^{T} = [a_{1} \ldots a_{m} \quad b_{1} \ldots b_{n} \quad c_{1} \ldots c_{m}]
\end{equation}

Where \( \lambda \) is forgetting factor and the model of predictor based on prediction error method is written as a linear regression neglecting \( \varepsilon(t) \):

\begin{equation}
\hat{y}(t) = \varphi^{T}(t-1) \hat{\theta}(t-1) \quad , \quad \varepsilon(t) = y(t) - \hat{y}(t)
\end{equation}

Above algorithm constitutes a scheme for estimating the parameters of a first order ARMAX model. The Least Square (LS) method determines the parameters which makes the Sum of Squared (SS) prediction errors as small as possible. It is an indispensable theoretical tool in experimental research. The idea behind LS is to fit a model to measurements in such a way that weighted errors between the measurements and the model are minimized. In this way, prior information about the model is used with the measurements to derive a posterior model. This idea can be used in linear or non-linear modeling. When the time variant system changes slowly, the recursive algorithm tracks corresponding parameters, describing such a process. To exploit this method we can complete the recursive algorithm by using either the forgetting factor or a Kalman filter. We used least-square method with forgetting factor (\( \lambda \)) because of simplicity of implementation and sufficiency in performance criterion. This method dedicates less weight to older measurements that are no longer representative for the system.
\[
\hat{\theta}(t) = \hat{\theta}(t-1) + K(t)\mathcal{E}(t)
\]
(4.28)

\[
K(t) = P(t-1)\varphi(t-1)\varphi^\top(t-1)P(t-1)^{-1}
\]
(4.29)

\[
P(t) = (I - K(t)\varphi^\top(t-1))P(t-1)/\lambda
\]
(4.30)

Figure 4.12 draws performance of on-line fault detection while choosing KSN-DSN in the middle of the container. Subplot (a) gives measured values of KSN (F10) and DSN (F11) as well as one-step a head prediction of the DSN. There are several faults on DSN which have been highlighted by ovals. Predicted values are very close to measurements and yield residual illustrated in subplot (b).

![Figure 4.12. Detecting faults for two near H-sensor](image)

Illustrated in above Figure 4.12, in addition to faults, disturbance also has provided relatively big residual. It makes a complexity while evaluation of the residuals.
Properties of an on-line prediction are described here while there is: (1) An abrupt single fault just on DSN. (2) A common abrupt variation in input-output sensor due to a disturbance which is not a real fault. Figure 4.13 illustrates results of on-line prediction in example 1, when there is a faulty reading \((H = 0)\) only on DSN for \(20 < t < 21\), distinguished by red colored plot. In this, subplot (a) gives the results of applying ordinary covariance matrix \((P)\) and old parameters in (4.27). In subplot (b), new estimated parameters have been used in each step of recursive procedure. Subplot (c) employs new parameters and updated matrix \(P\) to achieve the best output tracking. In each step of recursive technique, existing model is updated and prediction is forced to track actual value even in faulty mode. Then, in the best status, the residual converges to zero immediately after a big variation of input or output and fault duration is lost.

![Figure 4.13. Prediction with: (a) Ordinary method. (b) Old estimated parameters. (c) Normalized P(t).](image)

Starting point or end of a fault, not its duration might be detected in case a big residual exists. Although improved \(P(t)\) yields better prediction, it provides small residuals which are not helpful in fault detection. Then, we use ordinary \(P(t)\) just as defined in (4.30). As the second example, a big sudden change both in KSN-DSN for \(20 < t < 21\) min and prediction is illustrated in Figure 4.14. Despite one step delay, the predictor follows the changes. This delay must be considered while prediction otherwise an incorrect fault will be issued. There are some faulty circumstances such as long time
staying at zero (50<t<70) and stay at fix value (100<t<140 and t>155), significant short time change (t= 25) for $H$ in Figure 4.15.

![Figure 4.14. Prediction when there is an input-output sudden change with updated matrix P based on prediction error (with update parameters).](image1)

![Figure 4.15. Output, with and without faults, prediction using {(a) Improved prediction. (c) Normal prediction}. Residual using: {(b) Improved prediction. (d) Normal prediction}. Fault detection signal (e).](image2)
CHAPTER 4. Fault detection and diagnosis based on FIA

Most faults are still detected by the developed limit-trend and plausibility check and not by the residuals yielded from on-line estimation as clarified in subplot (d). Particularly for a fault on DSN such as stay at a fix value (100 < t < 140), residual is negligible and fault cannot be detect by on-line prediction technique.

4.5.2.1 Residual due to on-line model parameter identification

As it can be noted, a fault occurrence leads to a discrepancy between the model and the process outputs, resulting in an increase of $\varepsilon(t)$ as well as variance of the estimates. This might be a source of changing $\lambda$ in [26]. According with that, estimating parameters on-line with a long time horizon allows following their slow variations. These are not considered as faults, but rather due to normal evaluation of process. An on-line ELS algorithm with $\lambda=1$ figures out the long horizon estimates, Figure 4.16. This algorithm results in reference parameter estimation. A second estimator based on a short time horizon allows following fast variations, considered as symptoms of a fault (as shown in Figure 4.17). It results in a tracking model [27].

![Figure 4.16. Parameter variations when forgetting factor is chosen big ($\lambda=1$)](image-url)
In order to detect the duration of faults which cannot be characterized by using the residual of output, we may exploit residual of parameters. The reasonable result is that based upon the above Figure 4.17 parameters change even without fault occurrence in this system. Even with \( \lambda = 1 \), they change when door is opened because, system model changes while acting disturbances. Then using this method we won’t be able to discriminate faults from disturbances.

4.5.2.2 Residual evaluation using adaptive thresholds

As published in [30] and [31], defining a threshold means to find out the tolerant limit for disturbances and model uncertainties under fault-free operation conditions. It should consider dynamics of the residual generator as well as bounds of the unknown inputs and model uncertainties. In [21], threshold computation has been detailed based on different types of uncertainties. Also [28] quotes some conditions that threshold should satisfy to be robust against parameter uncertainty with sensitivity to small faults.

Since the generated residuals deviate from zero even without faults, they may contain a static part which is proportional to the input and a dynamic part dependent to
the trend of input. In [29] an adaptive threshold is introduced which uses a first order high pass filter to enlarge the threshold as shown in Figure 4.18. It contains also a proportional enlargement (K2) and finally, a low pass filter smoothes the threshold.

\[
L_{th}(s) = \frac{1}{1 + \frac{T_2}{T_1} T_D S} \left[ \left( \frac{T_D}{T_1} S \right) u + \| K_2 \| u + k_i \right]
\]

\[
L_{th}(z) = \frac{1 - e^{-\frac{T_2}{T_1} z^{-1}}}{1 - e^{-\frac{T_2}{T_1} z^{-1}}} \left( \frac{T_D}{T_1} z^{-1} u + \| K_2 \| u + k_i \right)
\]

T1 and T2 are selected according to the dominating time constant of the system. TD/T1 depends on the uncertainty of the dynamics. They are chosen according to the slowest process Eigen value or the cut-off frequency of the state variable filter, respectively. The amplitude of threshold and sample time are |L_{th}| and T_s, respectively.

Relation (4.32) still needs some improvements whereas it has an inherent delay from input signal (u). This delay makes threshold very sensitive to residuals so that it might meet residuals just when they have abrupt variations. To cope with this weak point the
CHAPTER 4. Fault detection and diagnosis based on FIA

formula should be used one step ahead (omitting $Z^{-1}$ from coefficient). In this way threshold obtained from KSN (input) will be earlier than signal from DSN (output). It is the state of the art in practice that thresholds are generally characterized based on experiences or by means of real tests and simulations.

From Figure 4.19, output sensor faults, “stay at fix value” have been marked by black colored ovals, where the orange ovals show the time once the door is opened (no fault). Upper subplot (a) represents measured signals of $H$ by a (KSN-faulty DSN) and one-step a head prediction of DSN. Below Figure 4.19 (b) gives residual ($y_{actual} - y_{predicted}$) and adaptive threshold as well.

---

Figure 4.19. Input, prediction, output with and without fault and residual.
Subplots (c) and (d) depict fault detection by residual-based and signal-based methods respectively. Residual is still big for opening the door (disturbance) as well as significant faults such that they could not be discriminated simply without using an adaptive threshold. A ramp shaped fault which had not been modeled by our signal-model based methods, is now detected by adaptive threshold technique. Besides, there may be delay between KSN-DSN. Since disturbance varies both KSN-DSN, this can be an idea to diagnose disturbance from sensor fault which only affects the DSNs.

4.6 Implementation of FDD on IMOTE2

Two types of topologies were implemented for fault detection by using Imote2 sensor network. Based on the first topology, according with the cluster-based topology of WSN shown in Figure 4.20, a KSN (K1) investigates several DSNs (S1…S3) in a cluster and detects only their related faults. In case no signal received from a DSN, it issues in turn a No-Signal warning for that DSN. Although simple in implementation, a faulty KSN results in missing fault detection of all DSNs in the corresponding cluster. The maximum allowed number of sensors or their identification number is defined in the program to allow specific DSNs to be member of that cluster.

In the second implemented topology, shown in Figure 4.21, there is no limitation regard to the number of permitted DSNs for each cluster. If the signal strength is enough to be received by a KSN, corresponding DSN will be also member of the cluster and clusters overlap each other. Multiple KSNs (K1 & K2) contribute to detect faults occurred in common DSNs. Despite great accuracy, more complicate evaluation is required.
A report of real time situation is given by the KSNs in detail (if connected to a computer). As represented by Figure 4.22, Sensor number, measured values of KSN and DSN(s), an estimation of DSN(s) and possible Fault-number for both $T$ and $H$ is given respectively. Amount of adaptive threshold gives a sense about the permissible difference between actual value and corresponding estimation. The second part of below report represents a fault on $H$, issued by both adaptive threshold technique and the proposed signal based detectors as fault number 2000 and 300 respectively.

---

**Sensor Name and sample Nr. ::> S0 ( 2 )**

- $T$ (KSN) ::> 23.78
- $T$ (DSN) ::> 23.08
- $T$ (estimation) ::> 23.1
- Threshold ::> 2.7

**Sensor Name and sample Nr. ::> S0 ( 3 )**

- $T$ (KSN) ::> 23.78
- $T$ (DSN) ::> 25.42
- $T$ (estimation) ::> 23.09
- Threshold ::> 2.83

Fault detected by Residual based method:
- $T$ (Fault Nr.) = 0 , $H$ (Fault Nr.) = 2000
- $T$ (Fault Nr.) = 3 , $H$ (Fault Nr.) = 300

---
Another report can be issued also by the Main station which gives information about the kind of fault and identification number of the faulty sensors. The faults can be issued by signal based mechanisms or adaptive threshold method, represented by Figure 4.23.

<table>
<thead>
<tr>
<th>ID. Number :</th>
<th>KSN (Head Cluster) = 463 ; DSN = 443</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Name and step: S1 ( 2 )</td>
<td></td>
</tr>
<tr>
<td>Fault Number : .........................................................</td>
<td></td>
</tr>
<tr>
<td>Fault (T) : 0 ; Fault (H) : 0</td>
<td></td>
</tr>
<tr>
<td>Model-based Fault (T) : 0 ; Model-based Fault (H) : 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID. Number :</th>
<th>KSN (Head Cluster) = 463 ; DSN = 3798</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Name and step: S0 ( 3 )</td>
<td></td>
</tr>
<tr>
<td>Fault Number : .........................................................</td>
<td></td>
</tr>
<tr>
<td>Fault (T) : 3 ; Fault (H) : 300</td>
<td></td>
</tr>
<tr>
<td>Model-based Fault (T) : 0 ; Model-based Fault (H) : 2000</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID. Number :</th>
<th>KSN (Head Cluster) = 463 ; DSN = 443</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Name and step: S1 ( 3 )</td>
<td></td>
</tr>
<tr>
<td>Fault (T) : 0 ; Fault (H) : 0</td>
<td></td>
</tr>
<tr>
<td>Model-based Fault (T) : 0 ; Model-based Fault (H) : 0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>ID. Number :</th>
<th>KSN (Head Cluster) = 463 ; DSN = 3798</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor Name and step: S0 ( 4 )</td>
<td></td>
</tr>
<tr>
<td>Fault (T) : 0 ; Fault (H) : 0</td>
<td></td>
</tr>
<tr>
<td>Model-based Fault (T) : 20 ; Model-based Fault (H) : 2000</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.23. Report issued by Main station an experiment.

In the third step of above report (S0 (3)), a fault on $H$ has been detected by two methods (signal-based and model-based) where fault on $T$ has been detected just by one technique. In the fourth step (S0 (4)), other faults on $T$ and $H$ has been detected only by model-based adaptive threshold technique. There is no fault reported on the other sensor (S1).

KSNs don’t need to transmit all the informative data to main station except yielded information about faults. Therefore, the process operates in a semi autonomous way according with the proposal of the present thesis.

In both above cases calculation for a pair of KSN-DSN starts as soon as KSN receives data from that DSN. Hence, any DSN is analyzed by a common loop in program in its turn. In addition to measuring task, on-line process model-based FDD and a developed signal-model based method are included in KSNs, Figure 4.24.
Three programs have been devoted for DSNs, KSNs and Main processor as follows:

1. A program written for DSNs consisting of 226 lines of code in C# includes measuring and transmitting data in each sample time. The number of utilized DSNs is optional and any new member gets new membership from related KSN.

2. A very important program for KSNs with 1124 lines of code in C# contains measuring, receiving, evaluation of received data from DSNs and sending data to the main processor. In addition to on-line system identification methods based on FIA, signal-model based methods proposed in 4 have been considered in this. KSNs equipped with this program diagnose 20 types of faults for each variable of environment that is 40 types of most probable faults for $T$ and $H$.

   In the model making part written for KSNs, one may choose: different model order, delays and forgetting factor ($\lambda$). Either old or new parameters while computing one-step ahead prediction are also chosen. The number of back step utilized in fault detection part can be selected by the user, based on the
application. In case a DSN has already become a member of a KSN, its unsuccessful communication due to a failure is diagnosed after running common loop in the program for all members of that cluster.

3. Main processor’s program with 127 lines of code in C#, manages received data from KSNs and issues respective fault alarms. If all the KSNs show an extended faulty situation in all clusters, Main processor releases a system-fault warning.

It is important to note that the communication between DSNs-KSN use different channel with that for KSNs-Main station.

![Figure 4.25. Flowchart of fault detection in a KSN.](image-url)
To avoid releasing wrong fault alarms because of possible difference between the actual and estimated values during the starting periods of the predictor, for a few steps (five steps in this application) no fault alarm is released.

Choosing SNs in vicinity as members of a cluster, synchronism problem between the measured data of the pairs KSN-DSN is omitted. For instance, when container’s door is opened some SNs near to door sense the variation and after a while the others far from the door sense it. Another source of such delay is not receiving signal from a DSN to the corresponding KSN for awhile. In this status data in the related vector will be shifted in time and delays arise. This kind of delay degrades the performance of the computational approaches. More accurate FDD is acquired by:

- Using a MISO model of KSNs-DSN and investigate the residuals using the topology of Figure 4.20.
- Sharing FDD by several KSN-DSN due to multiple SISO models by using the topology of Figure 4.21.

4.7 Summary and conclusion chapter 4

A combination of different methods is required for accurate fault detection as shown in Figure 4.26. The combinational method was implemented on IMOTE2 to achieve better performance of fault detection. Therefore, in addition to process model-based techniques, regular limit-trend checking method and signal model-based methods were developed and 40 types of system-fault and sensor-fault were distinguished.
Based on the proposed novel techniques, faults on \textit{T-sensor} as well as \textit{H-sensor} and no-signal received from a DSN can be diagnosed, separately and similar detection based on the residuals due to several SNs represents a disturbance instead of sensor-fault.

To apply process model-based methods by using Floating Input Approach (FIA), the following results were obtained for two types of FIA:

\textbf{Off-Line FIA:}

(1) Estimation was more accurate than average and median methods.

(2) Failed SNs were considered as the DSNs which should be estimated.

(3) Off-line FDD was not so powerful due to uncertainties and time variant.

\textbf{On-Line FIA:}

(1) Output observer was realized by a “one-step a head prediction”.

(2) Residual due to output (measured value of DSN) was more effective than it acquired from estimated parameters ($\hat{\theta}$).

(3) The best threshold for residual evaluation was developed by an adaptive method.
5. Energy saving in a WSN

5.1 Introduction

Sensor networks have emerged as a revolutionary technology for querying the physical world and hold promise in a wide variety of applications. However the extremely energy constrained nature necessitates that their design and operation be done in an energy aware manner. A typical Wireless Sensor Network (WSN) is expected to work without human intervention for a long time period. Hence, energy is of paramount importance in WSNs in order to achieve maximum network lifetime. In order to the energy constraint of sensor devices, WSN necessitates an energy-aware design to ensure the longevity of the network.

While most WSNs use battery-operated computing and sensing devices, new technologies like energy harvesting have been emerging and getting much attention in the research community recently. In addition, each layer of the protocol stack can employ various techniques to conserve energy whilst the hardware designs of motes can focus on extending its lifetime. Other applications also develop some recipes to save energy for the sensor networks.

The SNs in our desired network are supposed to be supplied by batteries which need to be exchanged regularly. The main goal of this chapter is introducing a novel technique for prolonging the overall life time of a wireless sensor network without energy harvesting. It tries to reduce power consumption of the SNs by reducing their active operation time even existing other methods of energy saving.
Battery depletion is among the most probable faults which can influence monitoring system. Any attempt to prolong the operation of the SNs is valuable from side of economic and operative activities. Energy consumption of batteries depends on several parameters such as weather temperature that they are in, sampling time of SNs, amount and regime of consumption (continuous or discontinuous) in different mode of operation and discharge curve of batteries. Figure 5.1 draws discharge curve of a battery in terms of temperature. The rate of decrease of voltage with increasing discharge will also be higher at lower temperatures, as will the capacity. In practice, if there is no way to scavenging or harvesting energy for the SNs, operators have to change the batteries regularly at least once a week.

![Figure 5.1. Discharge curve of a battery in different temperatures.](image)

Paper [33] uses extra knowledge about the environment to increase system lifetime. It argues that significant improvements in usable system lifetime can be achieved if the task allocation is aligned with the spatio-temporal characteristics of energy availability. A thermogenerator module is employed to harvest energy from temperature gradients between a heat source and the ambient by [34].

Authors of [35] describe a battery-less WSN that uses the combination of an electric double layer capacitor equipped with a small solar cell as its energy source to drive SNs with such source. Communication module has to wait till capacitor is fully charged. This behavior by the node seriously affects the design of the communication mechanism in battery-less SNs. It gives a practical way to address this problem with a new communication mechanism based on a real application scenario of environmental monitoring applications. Development of a self-powered system is described by [36]. They use liquid crystal display to provide the system output and an infra-red link to transmit the data output. Using super-capacitors in addition to batteries to prolong
lifetime of the batteries has been addressed in [37]. It uses a system that intelligently manages energy transfer for perpetual operation without human intervention or servicing. It employs a multi-stage energy transfer system that reduces the common limitations of single energy storage systems.

In [38] authors describe architectural and algorithmic approaches to enhance the energy awareness of a WSN. They mention various factors that affect system lifetime. A suite of techniques are introduced that perform aggressive energy optimization while targeting all stages of sensor network design, from individual nodes to the entire network. Authors of [39] consider the problem of positioning data collecting base stations in a sensor network. They show that choice of positions has a marked influence on the data rate, or equivalently, the power efficiency.

Optimal clustering in sensor networks is studied by [40]. It illustrates an optimal algorithm for clustering the SNs such that each cluster is balanced and the total distance between the SNs and master nodes is minimized. It reduces the communication overhead and hence the energy dissipation. Paper [41] uses powerful cluster-heads and basic SNs. All the nodes are randomly deployed in a specific area. To better balance the energy dissipation, it uses a simple mixed communication modes where the SNs can communicate with cluster-heads in either single-hop or multi-hop mode. Given the initial energy of the basic SNs, it drives the optimal communication range and identifies the optimal mixed communication mode to minimize the WSN’s lifetime. The research done over [42] reports an initial foray into the design of self configuring mechanisms for WSNs as well.

The article [43] presents and analyzes a variety of regular deployment topologies, including circular and star deployments as well as deployments in square, triangular, and hexagonal grids. In [44] novel protocols are developed for WSNs in order to ensure reliable packet transmission and maximize lifespan at the same time. The optimal transmission energies are derived which guarantee that the packets are received by the Base Station with a given reliability subject to achieving the longest possible lifespan.

Paper [45] studies different ways to firstly increasing single-hop throughput and reducing power consumption to prolong battery life. The second scheme focuses on energy-aware routing. The third scheme contributes to a dynamic increase of the lifetime of the sensor network. The fourth scheme prolongs the lifetime of wireless
sensor networks by cross-layer interaction. Finally, the fifth scheme focuses on the major energy wastage sources whilst achieving good scalability and collision avoidance capability. It is noted that some results of implementation has been reported from [46] which has been done under the direction of the author during a master thesis.

5.2 Energy Saving Methods

According to Figure 5.2, there are different methods of energy saving. In contrast to energy scavenging methods energy saving methods concentrate on controlling the amount of energy consumed by the network components on node level and on network level. Energy-saving methods are further classified as power management methods and data driven methods. No single method of energy saving is yet very effective to provide satisfactory results, but a hybrid approach can yield considerable results.

![Energy saving methods diagram](image)

**Figure 5.2. Further classification of power management methods**

5.2.1 Power Management Methods

Power management methods look for saving power by turning off the system components or taking the whole system into power down mode when a full operation is not required. Power management methods are further classified into two branches:

- sleep, wake scheduling
- energy efficient Media Access Control (MAC) protocols

Sleep and wake scheduling looks for an opportunity to save power by turning of some components in a way that it may not harm primary objective of network. A simple approach is “sense and send data” and stay in sleep mode for the rest time but a more complex
scheduling is needed for large self organizing networks. In big multi hop networks sleep wake scheduling becomes very complex as when a sensor node transmits some data the surrounding nodes must be in a receiving state otherwise packet will be lost. So a well synchronized sleep wake algorithm is mandatory. Energy efficient MAC protocols reduce the energy consumption by reducing the communication overhead.

These methods work on the MAC level, routing level and topology level to ensure that not a single bit of data is transmitted unnecessarily. In multi hop networks efficient routing algorithm are also employed to keep the network energy balanced, otherwise nodes which forward more packets gets power exhausted fast as compare to others.

5.2.2 Data Driven Methods

Data driven methods save energy of nodes and the whole network by controlling the amount of data they generate and how they generate. They are further divided in:

- Data reduction techniques
- Energy efficient sampling techniques

Data reduction methods decide whether a particular data is reasonable or not and decide the need to transmit data or not. Another way to reduce the data transmitted over the network is aggregating the samples as they propagate through different nodes. Energy efficient sampling methods focus on saving the energy consumed by sensor modules on a wireless sensor node. One such technique is adaptive sampling, where sampling time is reduced on times when high sampling rate does not reveal much detail.

5.3 Energy saving based upon FIA

Energy saving in wireless sensor networks using a floating input approach is a new idea proposed by the author for the first time. It has already been deeply described in chapter3.4 as well as in [12]-[16] for a closed space container.

The term floating means not fixed and floating input means an input quantity which is not referenced to any reference point. Floating input approach considers the knowledge of environmental variables at different points in the space and uses this knowledge to create relations between environmental variables at different points. This relation is usually a mathematical model. According with the statements of chapter 3.4,
a while after loading the container, when the EVs inside the container have less variations, it is the best time to take more DSNs to sleep mode and to predict their EVs instead of their direct measurement. By this way the number of communications, the most battery-power consumer will be reduced.

5.3.1 Simulations for implementation

The simulations shown in previous section are result of total offline study. In case of an offline study one has total control over the method and the parameters. Although the method used is an offline identification method (as mentioned in detail over the previous section 4.5.1) but for real applications it is not possible to change the model free parameters in run time. For example whilst working on ones desktop one can try different options for same data set, for example change the model structure, change the orders, change the number of training samples and etc.

For a real system where offline system identification algorithm is running on an embedded system, it is not possible to change the model structure or change the other options. Therefore some practical steps must be applied so that the approach will change a bit. In the new approach the algorithm first waits for $N_{T_{\text{min}}}$ number of measured samples from input and output. $N_{T_{\text{min}}}$ is the minimum number of samples to be used for parameter estimation, it is a pre-programmed number. At this point also let us define $N_T$ as the number of samples used to estimate unknown parameters of a model.

After collecting enough samples, parameters for chosen model structure are estimated. The estimated model is than validated over a prediction horizon of $N_v$ samples. During validation part both the KSN and DSN are in active mode in the sense that the algorithm can calculate error between actual and estimated data. If the estimated model passes validation test, the KSN starts estimating further data points over pre-defined prediction horizon $N_p$ samples. The cycle then repeats itself again. If the estimated model cannot pass the validation test, number of training samples is increased by a factor of $N_{\text{step}}$ and a new model is estimated, number of training samples is increased as long as they are less than maximum number of training samples $N_{T_{\text{max}}}$.

After that only latest $N_{T_{\text{max}}}$ samples are used all other samples are discarded. When a model is got successfully validated $N_T$ is set to $N_{T_{\text{min}}}$ and the cycle goes on and on. With this new approach, algorithm keeps on updating the model after certain time which
assures that model incorporates the latest behavior of system. A point must be noted here that we are dealing with a nonlinear system and we are trying to approximate it with a linear model. Practically a nonlinear system cannot be approximated by a linear model over its whole range of operation, so updating the model is necessary. The validation assures that the estimated model gives satisfactory results in order that fake estimation is avoided.

Figure 5.3 shows the measured and estimated values of $T$ in a room and their error. With this new approach there is very less error in estimated and measured values. Sensory data of $H$ have much dynamic behavior but still the algorithm is able to capture the system dynamics. Those portions where estimated data is falling down, displays the learning part. A first order ARX model is used to approximate the system dynamics. The SN in this experiment is in active mode and never goes to sleeping mode because the main goal is to achieve high performance prediction.

![Figure 5.3. Measured and estimated $H$ with new approach (No sleeping)](image)
5.3.2 Program Flow of Main Application

The main application has three parts. The first one is Global Data area, second one is Main thread and third one is Ident thread. Figure 5.4 shows its pictorial view.

**Global Data** - The global data area holds the data which is used by both the threads. Most of the data is global so that both the threads can easily access the data.

**Main Thread** - Main thread is responsible for organizing all the resources of the system and running the different tasks. Figure 5.5 in below points the flow chart of Main thread.
Main thread starts with initializing global variables and then creates Ident Thread. Later for most of the time, Main Thread deals with radio. It receives the radio packets and extracts the data of environmental variables from received packets. The extracted data is then saved in respected data buffers in global data area.

**Ident Thread** - This thread, shown in Figure 5.6, is activated at predefined time interval, this interval is same as the sampling time of the network. Each time the thread is activated it increments the sample-counter. Based on value of sample-counter decides that what to do and currently in which part of energy cycle the system for instance learning part, validation part or estimation part is.

![Ident Thread flow chart](image)

Figure 5.6. Ident thread flow chart

When sample counter reaches the value of number of training samples $N_T$, it estimates the model parameters. From next sample till $N_V$ number of samples, it predicts the environmental variables. When sample counter reaches the value of $N_T + N_V$ it validates
the model. If model passes validation criterion it keeps on estimating environmental variables otherwise it goes for another model.

Once started the Ident Thread runs an infinite loop where at the start of loop it goes into sleep state for time $T_s$ (the sampling time) and after that it runs the whole energy saving algorithm and once more goes into sleep state for time $T_s$ (the sampling time), the cycle then goes on and on.

### 5.4 Experiment demonstration

After implementing the program a comparison between the results acquired from Imote2 (sensor node) and the MATLAB simulation is made from same data set. Figure 5.7 addresses a comparison between implemented results and simulated results. One can hardly see any difference between two results, which shows that the implementation works fine. For comparison a test was performed on Imote2 and after words same data was fed to MATLAB program and the result of both is then plotted here for comparison.

![Compare Matlab and Imote2 Results](image)

**Figure 5.7. Comparison between results of implementation and MATLAB simulation**

Another test of final algorithm with energy saving enabled is also performed. In this test three sensor nodes are used. One acts as KSN (key sensor node), other as DSN (desired
sensor node) and one extra node called S2. DSN is enabled with energy saving algorithm and S2 does not implement energy saving algorithm. All three nodes have a sampling time of one minute. DSN and S2 are battery powered while the KSN is attached to computer USB Port. DSN and S2 go into deep sleep mode when they need not to transmit and receive data to save more energy. In energy saving mode DSN sends one measured sample every five minutes. The nodes are allowed to sense and send data to KSN until battery expires. Figure 5.9 depicts the result of experiment based on the setup represented in Figure 5.8.

![Figure 5.8. Experimental setup](image)

The portions of plot where error is perfectly zero shows learning part. In this part the algorithm estimates the model and validates it. The part of plot where error has some other value hints the actual estimation.

One strange phenomenon is observed from the results that error always goes negative with a similar pattern this phenomenon was not observed in previous tests. The reason for this is self heating and cooling of the sensor and other nearby electronic components. In previous tests the sampling time was very short and the DSN did not really go into energy saving mode. It was kept like this for the purpose of testing, so that it can be made sure that the actual and estimated data has less error.
In this final test when DSN goes into energy saving mode its sampling rate is reduced very much, during this time all the electronics of circuit except some important parts is turned off so the sensor gets enough time to cool down, this results the strange behavior of error. This test also produces results as shown in Table 5.1. It represents how long each node survived on battery. During this test both the DSN and S2 were powered with batteries of same brand, capacity and same charge. It is clear that the sensor node with energy saving enabled survived for longer time (approximately 74%) than one which was not running energy saving algorithm.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Samples</th>
<th>Minutes</th>
<th>Hours</th>
<th>Days</th>
</tr>
</thead>
<tbody>
<tr>
<td>KSN (Estimator)</td>
<td>5207</td>
<td>5207</td>
<td>86.783</td>
<td>3.616</td>
</tr>
<tr>
<td>DSN (With Energy Saving)</td>
<td>5207</td>
<td>5207</td>
<td>86.783</td>
<td>3.616</td>
</tr>
<tr>
<td>S2 (Without Energy Saving)</td>
<td>2994</td>
<td>2994</td>
<td>49.9</td>
<td>2.079</td>
</tr>
</tbody>
</table>
Figure 5.10 depicts result of another test with a small modification. In this test if the error grows more than a pre-determined threshold, the KSN sends a message to the DSN to stop energy saving mode and continue normal operation. This effect is shown at very beginning just before sample number 200. So when error grows behind the threshold DSN returns to normal operation.

![Image of Measured and Estimated Data](image)

**Figure 5.10.** Final implementation results with three nodes & error checking

### 5.4.1 Analysis of Energy Saving by FIA

It is hard to describe the accurate AC or DC values of current and power for an embedded system running different software modules. Instantaneous values also don’t work well, so the average of current consumption and finally average power consumption over defined time can be used. Measurement procedure is simple but it is time taking and cumbersome. To find a meaningful value of current and power we need to take average of the quantity over time, to converge to a better result it is recommended to repeat the experiment for several times, then for each experiment an average of the quantity should be found and finally average of all the averages.

Figure 5.11 shows electronic setup used to measure current consumption. Normally it is not possible to measure current though a device using an oscilloscope. But if a one ohm
resistance of sufficient wattage is placed in series of the device then the voltage drop across one ohm resistance is directly proportional to the current through it. Care must be taken whilst picking up a resistor, its power ratting should be greater than square of the maximum current drawn by the device otherwise a damage may arise.

![Figure 5.11. Current measurement setup for Imote2](image)

According to the mesh rule, same current fellows at a given time through all devices connected in a loop. So the voltage across resistor is a direct measure of current consumed by Imote2. It is difficult or impossible to realize with continuous time equipment so a digital storage oscilloscope must be used. Table 5.2 represents the measurement results where $I_{\min}$, $I_{\max}$ and $I_A$ is the minimum, maximum and average current consumption respectively. $V_S$ is the supply voltage, $P_A$ is average power, $T_s$ is sampling time, the time over which averaging of current is done.

<table>
<thead>
<tr>
<th>Operation Mode</th>
<th>$I_{\min}$ (mA)</th>
<th>$I_{\max}$ (mA)</th>
<th>$I_A$ (mA)</th>
<th>$V_S$ (V)</th>
<th>$P_A$ (mW)</th>
<th>$T_s$ (s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sensor node</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Idle Mode (Radio off)</td>
<td>29.6</td>
<td>38.4</td>
<td>32.4</td>
<td>4.5</td>
<td>154</td>
<td>9</td>
</tr>
<tr>
<td>Idle Mode (Listening)</td>
<td>48.8</td>
<td>58.4</td>
<td>54.0</td>
<td>4.5</td>
<td>243</td>
<td>9</td>
</tr>
<tr>
<td>Deep Sleep (Radio off)</td>
<td>-</td>
<td>-</td>
<td>0.387</td>
<td>4.5</td>
<td>1.7</td>
<td>-</td>
</tr>
<tr>
<td>Sense &amp; Send (transmit mode)</td>
<td>49.6</td>
<td>130.4</td>
<td>84.1</td>
<td>4.5</td>
<td>245</td>
<td>60</td>
</tr>
<tr>
<td>KSN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sense + Receive</td>
<td>51.2</td>
<td>144</td>
<td>81.4</td>
<td>4.5</td>
<td>266</td>
<td>2</td>
</tr>
<tr>
<td>Save (Learning part) + (listening mode)</td>
<td>51.2</td>
<td>144</td>
<td>81.4</td>
<td>4.5</td>
<td>266</td>
<td>2</td>
</tr>
<tr>
<td>Parameter Estimation (listening mode)</td>
<td>51.2</td>
<td>144</td>
<td>81.4</td>
<td>4.5</td>
<td>266</td>
<td>2</td>
</tr>
<tr>
<td>Prediction mode (listening mode)</td>
<td>51.2</td>
<td>144</td>
<td>81.4</td>
<td>4.5</td>
<td>266</td>
<td>2</td>
</tr>
<tr>
<td>Sense + Send &amp; Idle &amp; Listen (Radio on)</td>
<td>58.4</td>
<td>128.8</td>
<td>105.6</td>
<td>4.5</td>
<td>239</td>
<td>300</td>
</tr>
<tr>
<td>DSN</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Normal mode:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sense (every 1 min) + Send (every 1 min)</td>
<td>0.387</td>
<td>128.8</td>
<td>19.1</td>
<td>4.5</td>
<td>47.1</td>
<td>60</td>
</tr>
<tr>
<td>+Deep Sleep + (Transmit mode)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Energy saving mode:</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Sense (every 5 min) + Send (every 5 min)</td>
<td>0.387</td>
<td>128.8</td>
<td>2.47</td>
<td>4.5</td>
<td>11.1</td>
<td>300</td>
</tr>
<tr>
<td>+Deep Sleep + (Transmit mode)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Above $V_s$ refers to actual measured voltage of four batteries of Imote2 and relation
($P_A = V_s \times I_A$) is used to find power consumption (mW) in each cases. Whereas the
measured currents are not constant, maximum and minimum values can be recorded and
$I_A$ is average of those current in mA.

The amount of energy is directly proportional to number of samples predicted
($N_p=150$). It is equal with the time for which energy is saved. It is also inversely
proportional to the number of samples used to train ($20 \leq N_t \leq 100$) and validate ($N_v \geq 19$)
the model. So maximum energy is saved when less number of samples is used to train
and validate the model and more samples are predicted using the model.

The DSN wakes up and senses and sends one sample every 5 minute of prediction
period (150 min). It is used for validation of prediction. If residual of measured value
and predicted value satisfies the predefined thresholds, prediction can be continued for
the next 5 minutes. Different types of sensors support different sets of modes and even if
they support the same set of modes, they may use different terminology. For an Imote2
the important modes are given through Table 5.3 and following definitions:

- **Normal mode** - The CPU and peripherals are active.
- **Idle mode** - The CPU is inactive
- **Deep-idle mode** - The core PLL (Phase Locked Loop) is disabled
- **Standby mode** - The peripheral PLL is disabled
- **Sleep mode** - The low voltage power domains and internal SRAM are disabled
- **Deep sleep mode** - USB hardware is shuts-down. When the Imote2 recovers from the
deep sleep state, it is a reboot. The high-voltage power domains are disabled.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Operating current</th>
</tr>
</thead>
<tbody>
<tr>
<td>Current in deep sleep mode</td>
<td>387 μA</td>
</tr>
<tr>
<td>Current in active mode</td>
<td>13 MHz, radio off</td>
</tr>
<tr>
<td>Current in deep sleep mode</td>
<td>13 MHz, radio Tx/Rx</td>
</tr>
<tr>
<td>Current in deep sleep mode</td>
<td>104 MHz, radio Tx/Rx</td>
</tr>
</tbody>
</table>
The above typical values are different in compare with measured values in the Table 5.2 which have been measured under real conditions with full programmed Imote2. Based on actual specification which we have obtained from different modes of operation there will be the minimum and maximum energy saved amount which can be given by $ES_{\text{min}}$ and $ES_{\text{max}}$ as follows:

$$N_p = 150$$
$$N_v = 19$$
$$N_{\text{max}} = 100$$
$$N_{\text{min}} = 20$$
$$P_{\text{normal}} = 48.7 \text{ mW}$$
$$P_{\text{es}} = 11.1 \text{ mW}$$

$$ES_{\text{min}} = \frac{N_p}{N_{\text{tmin}} + N_v + N_p} \left( \frac{P_{\text{normal}} - P_{\text{es}}}{P_{\text{normal}}} \right) \times 100 = 43\%$$ \hspace{1cm} (5.1)

$$ES_{\text{max}} = \frac{N_p}{N_{\text{tmax}} + N_v + N_p} \left( \frac{P_{\text{normal}} - P_{\text{es}}}{P_{\text{normal}}} \right) \times 100 = 61\%$$ \hspace{1cm} (5.2)

$N_{\text{min}}$ and $N_{\text{max}}$ are minimum and maximum number of training samples, respectively. The first value is a predefined value. $P_{\text{saved}}$ represents the power that is saved with the help of energy saving method and $P_{\text{normal}}$ is the power consumption under normal operation mode and $P_{\text{es}}$ is the power consumption of system during energy saving mode. They can be obtained from technical specifications of sensor node in different modes. According to energy saving based on FIA, it works in the following limits: (43% $<$ Energy Saved $<$ 61%). Figure 5.12. Energy saving based upon FIA draws a whole procedure of energy saving based upon FIA.

The respective part of actual operating mode in each sample time is given by Figure 5.13. It contains short period of deep sleep, some time for wake up, sense and send. The last one, Figure 5.14 displays procedure of a long time prediction consist of deep sleep, wake up, sense and send.
Figure 5.12. Energy saving based upon FIA

Figure 5.13. Normal operating mode of sensor node

Figure 5.14. Long deep sleep while prediction based on FIA
5.5 Modification of predictions

Figure 5.15 illustrates that due to inherent nonlinearity, uncertainty and time varying, prediction based on acquired model from previous data doesn’t fit actual measured values. Although this difference is reduced by using a MISO model, it never becomes zero. Looking to a longer period of operation, different stages are observed in Figure 5.16. A difference exists between new measured values of a new learning part and last predicted value, published in [47].

Figure 5.15. Estimation using K1.

Figure 5.16. Whole stage of estimation and prediction
Input is always available from a KSN where output from a DSN during prediction stage is not available. Here a mathematical technique is applied to improve continuity between prediction and measurements of EVs. It also helps to have more accurate off-line prediction. Figure 5.17 displays an actual measurement (in practice last point of actual curve is accessible) and a predicted curve. To move the green curve, acquired by predictors to the place of blue curve so that their last points coincide with each other, equation (5.3) in the following has been used. Other points of prediction will also be moved linearly and the first point of prediction won’t change. Blue solid curve points the capability of this method.

\[
\hat{y}_{\text{new}}(t) = \hat{y}_{\text{old}}(t) + \hat{y}_{\text{last}} \cdot \frac{t - t_0}{t_1 - t_0}
\]  

(5.3)

In the above formula, \(t_0\) and \(t_1\) are the starting time of first and second measurements, respectively. \(\hat{y}_{\text{last}}\) gives the last point of prediction and \(y_{\text{first}}\) is the first point of second measurement area and \(\hat{y}_{\text{new}}\) refers the new improved prediction and \(\hat{y}(t)\), prediction in time moment (t).

Figure 5.17. Comparing the result of primary prediction and its improvement.
5.6 Combinational Energy saving and Fault detection

The given FDI techniques can favorably be combined with the energy saving method. Accordant with Figure 5.18, the energy saving method works even while environment is faulty but not if sensor (DSN) is faulty (refer to section 4.2 to see the definitions of fault types). Whilst saving energy fault detection is stopped.

Figure 5.18. Flowchart of the system while working in both FDD and ESM.
5.7 Summary and conclusion chapter 5

Many techniques have been presented so far to prolong the network life. Energy saving using Floating Input Approach is a new idea which was offered during this PhD thesis and implemented during a master project.

Most of the energy saving methods concentrates on media access control layer protocols to reduce the amount of the energy wasted in communication. Some methods deal with energy wasted due to unnecessary sampling and data processing and try to save this energy. By using prediction of the environmental variables by means of the surrounding sensor nodes, energy saving based upon the Floating Input Approach eliminates the need for extra sensing, data processing and communication until it is really required. Comparison between respective simulations and implementations indicates that this method of energy saving is practically used in a WSN. The given approach can be used along with other existing energy saving approaches to increase the overall performance. From above discussion and the work presented in this chapter following points are concluded:

- A test revealed that (for that particular case) a sensor node with energy saving enabled survived for 74% more time in network as compared to other nodes. During this test maximum accuracy of 0.8 °C was observed for temperature measurements.
- Use of higher order models and other complex polynomial structures does not help much in improving the results in contrast to energy cost of the models.
- Choice of KSN (key sensor node) or the estimator node is very essential.
- In order to inherent nonlinearities in the system a very large sampling time may cause large error in estimations.
- There is no guaranty that each time KSN will be able to take a DSN to energy saving mode. It may take several tries for a KSN to find a good model.
- In favorable conditions method can save a maximum of 60% energy as compared to normal energy consumption in the particular case (refer to section 5.4.1).
6. Conclusions, future prospects

6.1 Conclusions

A novel method named Floating Input Approach (FIA) was introduced. It was utilized as a tool to achieve two important objectives, energy saving and fault detection. The FIA simplified a complicate hybrid model by using properties of distributed monitoring assuming the surrounding Key Sensor Nodes (KSNs) as inputs and a Desired Sensor Node (DSN) as output. The simplified linear model was employed to estimate environmental variables, temperature and relative humidity in place of a DSN in on-line and off-line forms. The effects of size of data-samples, various performance indexes as well as numbers of employed sensor nodes on the accuracy of predictions were studied by using MATLAB programs. The model making stage terminated choosing an Auto Regressive Moving Average with eXternal input (ARMAX) model.

Several signal & fault model-based methods were developed for fault detection. Furthermore, non-modeled fault types were also distinguished by using an on-line process model-based method and a recursive technique determined unknown parameters of ARMAX models. To cope with the system containing time varying and evaluate residual yielded by predictions and measurements, an adaptive threshold method was formed. Off-line method of identification based on FIA was used for a high value of energy saving in sensor nodes and was compared with the best existing methods. It was also utilized to estimate the environmental variables on a faulty sensor node.

Finally, a combinational method was developed for simultaneous energy saving and fault detection. The capability of the proposed techniques was endorsed by using simulations, mathematical proofs and actual implementation results.
6.2 Future prospects

There might be still some rare drift-wise faults that cannot be detected confidentially by using the proposed techniques. Then, investigation of following techniques can be of interest for the next steps of implementation:

- Decision making sharing fault detection results due to multiple SISO models.
- Using MISO models in each cluster and investigate the yielded residuals.
- Decision making based on the residuals due to both Temperature and Relative Humidity at the same time.
- Using augmented version of fault detector, consisting of proposed limit-trend, On-line and Off-line model-based techniques.

From energy saving point of view, combining the quoted recipe with the existing battery management techniques to achieve a better performance of energy saving can be very interesting. Furthermore, implementing simultaneous fault detection and energy saving can also be a venue for the future works.
References


Appendix: List of publications


