Title	Heterogeneous Sensing for Temperature Control in Cyber-Physical Smart Home Systems
Author(s)	Chung, Linh Thuy
Citation	
Issue Date	2015-12
Type	Thesis or Dissertation
Text version	author
URL	http://hdl.handle.net/10119/12984
Rights	
Description	Supervisor:Lim Yuto, School of Information Science, Master

Heterogeneous Sensing for Temperature Control in Cyber-Physical Smart Home Systems

CHUNG, Linh Thuy

School of Information Science
Japan Advanced Institute of Science and Technology
December, 2015

Master's Thesis

Heterogeneous Sensing for Temperature Control in Cyber-Physical Smart Home Systems

1310254 CHUNG, Linh Thuy

Supervisor :	Associate Professor Yuto Lim
Main Examiner :	Associate Professor Yuto Lim
Examiners :	Professor Yasuo Tan
	Professor Yoichi Shinoda

School of Information Science
Japan Advanced Institute of Science and Technology

November, 2015

Abstract

Nowadays, a smart home has been developed to automatically achieve some services using sensors and actuators with the goal to improve the occupant experience, e.g., comfortable and easier life environment. Smart home system is one of Cyber-Physical System applications, which is defined as tight integrations of computation, communication, and control for active interaction between physical and cyber elements in which embedded devices, such as sensors and actuators, are wireless or wired networked to sense, monitor and control the physical world. It is an appropriate and efficient way to design the home control system. It is believed that in both the academic and industrial communities that CPS will have great technical, economic and social impacts in the future. CPS environment contains the different terms in its own elements e.g, sensors, actuators, communication media. In real scenario where users need a single result from whole system, handling the heterogeneity of sensors requires to manage the collaborative nature of sensors, that leads to difficulty in processing or estimating desired parameters in high accuracy. Heterogeneous data from heterogeneous and CPS-based oriented sensor, which are equipped on different appliances, have different sensing performance information(e,g. operating range, response time, accuracy, setting interval), that might cause by the unpredictable change of environment

This paper proposes a new framework, the heterogeneous data processing and estimating system (HDPES) that can provide a highly accurate sensed data and/or estimate a desired data using the CPS-oriented and heterogeneous sensors in the cyber-physical smart home environment. The design of HDPES is considered in heterogeneity of sensing performance and sensing data to increase the reliability and accuracy of the temperature control system in Smart Home

By using the raw data from experiments, we analyze and evaluate our proposed framework in the home environment by using R software, a useful program for statistical computing and data analysis. Through multiple data estimation methods, simulation results reveal that our proposed system HDPES is adaptable and feasible for satisfying normalisation sensing error and estimation the desired parameter at a particular estimating point in cyber-physical smart home environment.

Acknowledgements

I wish to express my sincere appreciation to all those who have offered me help during the whole period of my research.

First and most, I wish to express my sincere gratitude to my supervisor, Associate Professor Yuto Lim, for all of his assistance and guidance. He helps me to finish my master study with his knowledge, enthusiasm and patience, not only for my research but also for many other skills like how to prepare a good slides as well as, how to make a good presentation. He always take a lot of time to explain and discuss with me when I have problem in research. I especially treasure the experience gained over the past year from him.

I would like to thank Professor Yasuo Tan, my sub supervisor for his encouragement and support throughout my study and experiments. Thanks to his comments and guidance, not only to my research, but also presentation skill, make me understand my research better and express views more clearly.

I would like to express my thanks to Professor Yoichi Shinoda for taking time to attend my defence and review my thesis. His comments given at my mid-term defence have been very helpful for my research, which make me better understand and give me important views about my research.

Many thanks to my friends, their helps, encouragements make me through many difficult periods.

Last but not least, thank to my parents, their great love and fully understanding through all these years is my strongest motivation to go on my study and pursue my life goals.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Research Background

1.1.1 Cyber-Physical Systems and their Applications

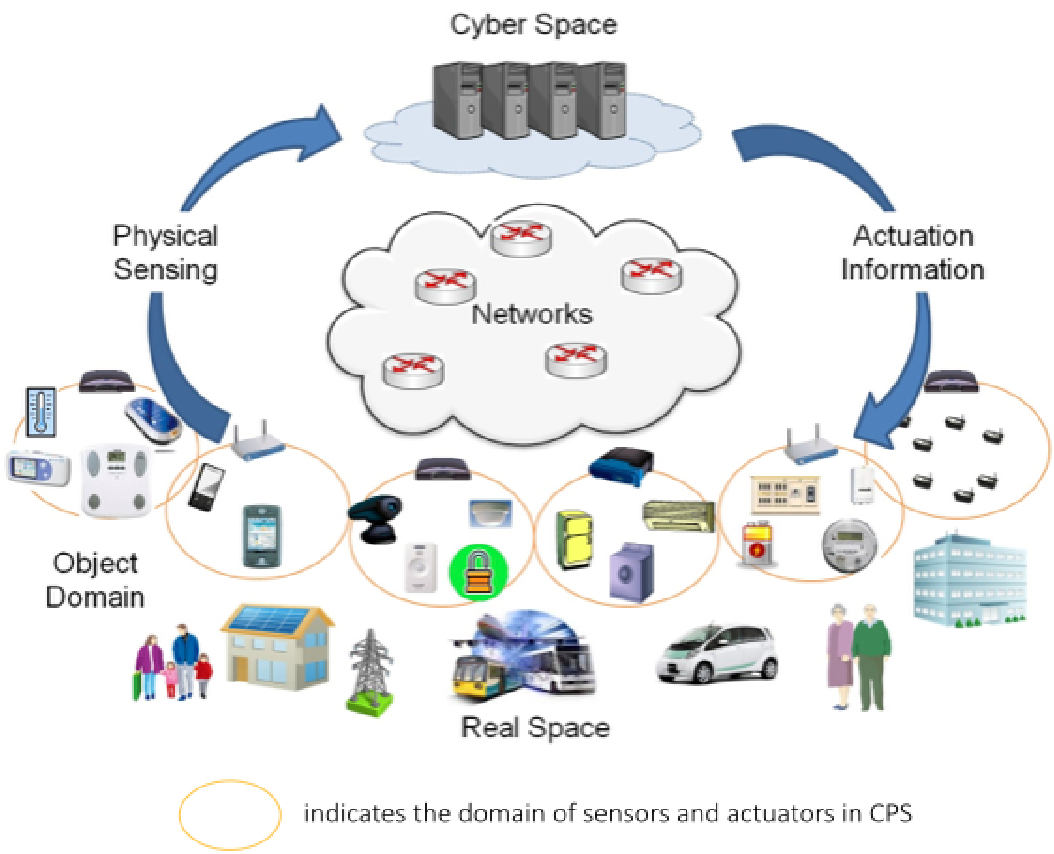


Figure 1.1: Cyber-Physical Systems Environment

Cyber-Physical Systems (CPS) are integrations of computation, networking, and physical processes. Embedded computers and networks monitor and control the physical processes, with feedback loops where physical processes affect computations and vice versa. CPS integrates the dynamics of the physical processes with those of the software and networking through a lot of sensors and actuators, providing abstractions and modelling, design, and analysis techniques for the integrated whole[1]

CPS brings many benefits by merging computation and communication with physical processes. CPS applications bring advance in many areas such as: health care and medicine, disaster detection and recovery, energy, robotics, smart transportation, smart home and another smart structure.

1.1.2 Smart Homes

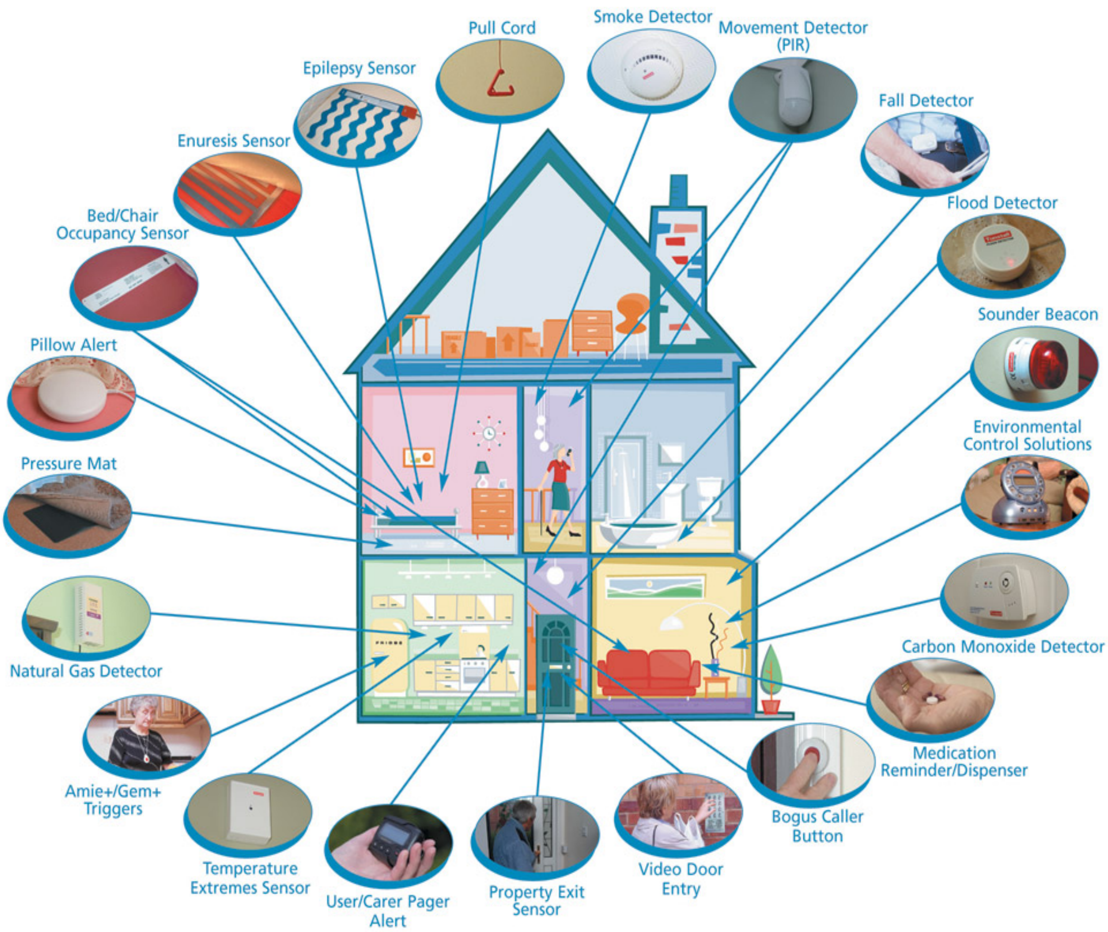


Figure 1.2: SmartHome Environment

Smart Home is a living environment that incorporates the appropriate technology, called Smart Home technology, to meet the resident goals of comfort living, life safety, security and efficiency. Smart Home technology started for more than a decade to introduce the concept of networking devices and equipment in the house. It is a home automation system that allows for controlling over a home environment, media systems, home security, and integrates with an easily accessible user interface through home network. The devices and systems in Smart Home environment can communicate with each other and can be controlled automatically in order to interact with the household members and improve the quality of their daily life.

In a Smart Home system, one of CPS applications, many devices and appliances are equipped with sensors and actuators to meet the demands or preferences of occupants such as: controllable door, fire alarm, lighting control, human present detector, temperature control, etc. These different devices and appliances lead to the presence of heterogeneous sensor in Smart Home.

1.1.3 Heterogeneous and Homogeneous Sensor

Nowadays, sensors used in CPS environment can be classify into two types: homogeneous and heterogeneous sensors. Homogeneous sensors are identical in term of energy, hardware complexity, performance. Heterogeneous sensors consist of multiple physically difference types of sensor.

Table 1.1: Type of sensors

Heterogeneous Sensors	*Homogeneous Sensors*
They consist of sensors have differences in processing power, resource, communication module, AC power supply.	They have the same device characteristics and sensing features from the same manufacturer
They are also different in sensing performance in terms of response time, accuracy, resolution, unit and operating range	

1.2 Motivation

In Cyber-Physical Smart Home environment, to collect data from the domain of heterogeneous sensors and actuators, two types of data collection method can be implemented.

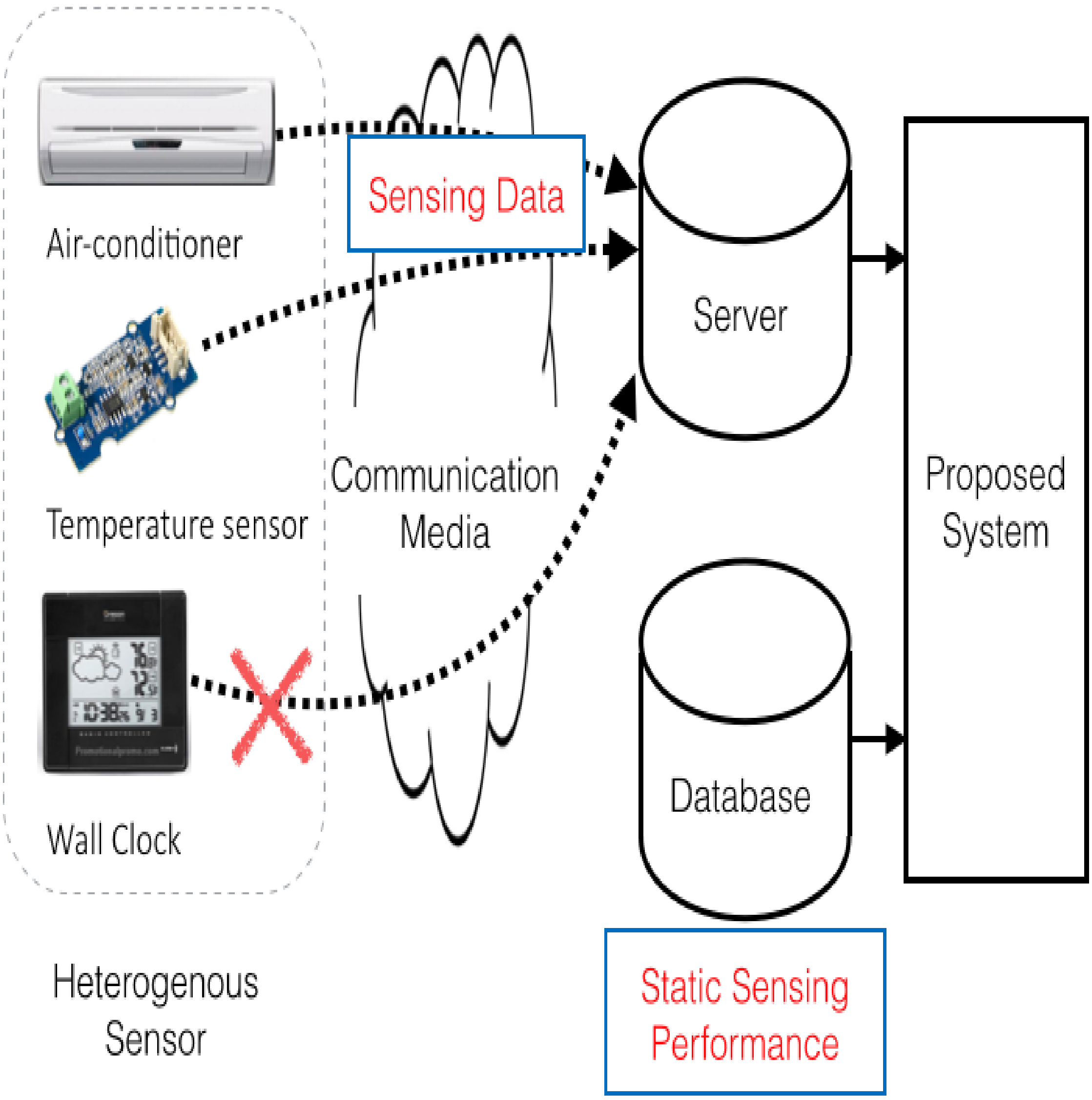


Figure 1.3: Look-up table Data Collection Method

Look-up table Data Collection Method is a feasible method to collect the sensed data from traditional heterogeneous sensor and retrieve the sensing performance. Usually, the sensor performance data sheet is commonly static from a pre-prepared database, which is provided by the manufacturer.

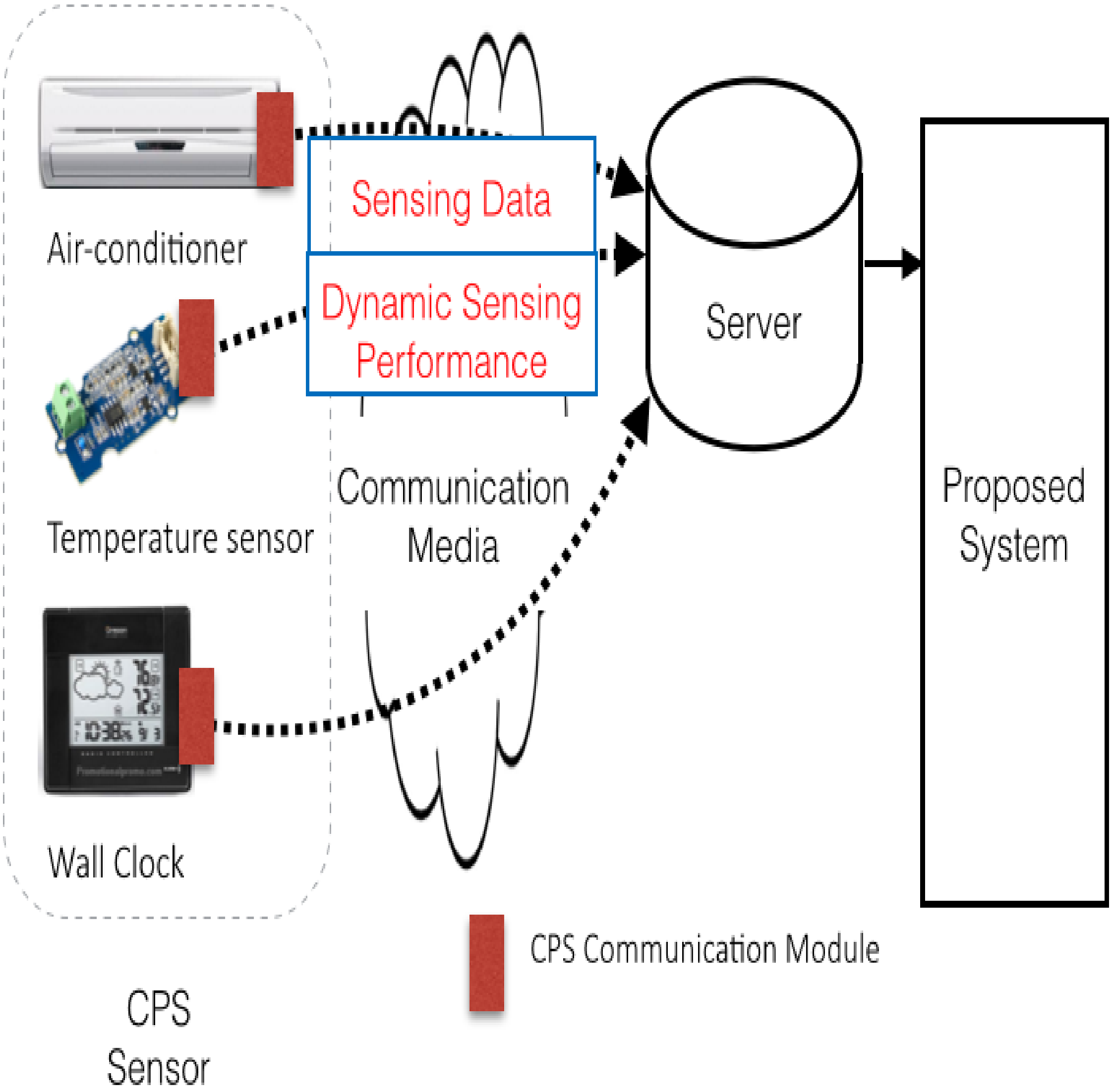


Figure 1.4: CPS-based Data Collection Method

CPS-based Data Collection Method: proposed method to be expected to collect not only the sensed data, but also the dynamic sensing performance from CPS sensor in timely manner. In this research, CPS sensor is defined as a sensor that is able to send its own sensing performance that might cause by the unpredictable change of environment using a CPS communication module.

1.2.1 Sensor Performance

Each of sensor has its own performance parameters (e.g., resolution, accuracy, repeatability, operating range, response time), which are different from others. These performance parameters can contribute error to the accuracy of sensed data, which are changing according to dynamic environment. Sensor performance characteristics give a technical information for certain sensor performance parameter with the specified definition and meaning. Figure 1.6 and 1.5 shows the sensor performance parameters in data-sheet of sensor SHT7x (including SHT71 and SHT75), relative humidity and temperature sensors with pins.

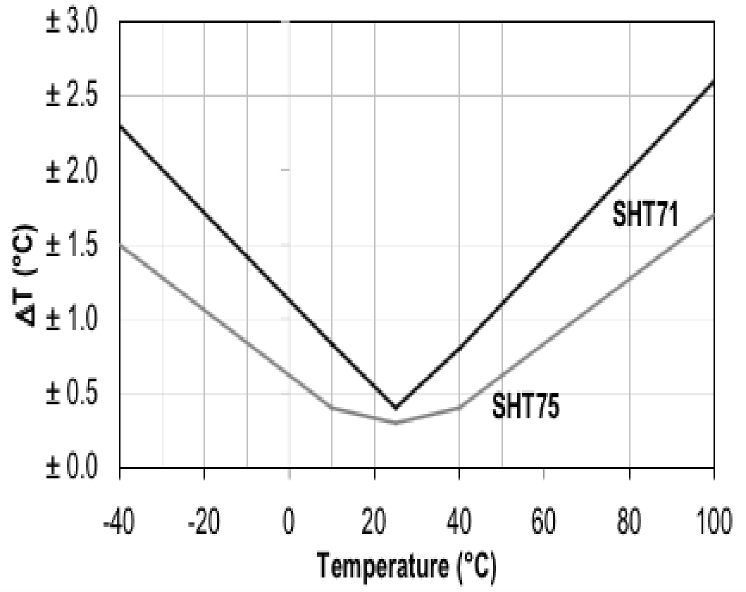


Figure 1.5: Maximal T-tolerance per sensor type

Temperature

Parameter	Condition	min	typ	max	Units
Resolution[1]		0.04	0.01	0.01	°C
		12	14	14	bit
Accuracy[2] SHT71	typ		±0.4		°C
	max	see Figure 3			
Accuracy[2] SHT75	typ		±0.3		°C
	max	see Figure 3			
Repeatability			±0.1		°C
Operating Range		-40		123.8	°C
		-40		254.9	°F
Response Time [6]	tau 63%	5		30	s
Long term drift			< 0.04		°C/yr

Figure 1.6: Performance Information of Sensor SHT71 and SHT75

In this research, the influence of sensor performance parameter is analyzed, especially considering in accuracy, operating range and response time

1.2.1.1 Operating Range and Accuracy

Sensor works stable within recommended normal range, after return to normal range it will slowly return towards calibration state by itself. Changes in the temperature of the surrounding area can result in significant shifts in the dielectric constant of area, which introduces inaccuracies in the sensor readings. Accuracy parameter is obtained in sensor performance to describe generally as the largest expected error between actual and ideal output signals. Error comes from influence of measured temperature in operating range on accuracy parameter is described as this equation:

$$\Delta d = \begin{cases} d_{opt},\ d = d_{opt} \\ (\Delta d_{min} - \Delta d_{opt})/(d_{opt} - d_{min})\cdot \mid d - d_{opt} \mid + \Delta d_{opt},\ d < d_{opt} \\ (\Delta d_{max} - \Delta d_{opt})/(d_{max} - d_{opt})\cdot \mid d - d_{opt} \mid + \Delta d_{opt},\ d < d_{opt} \end{cases}$$

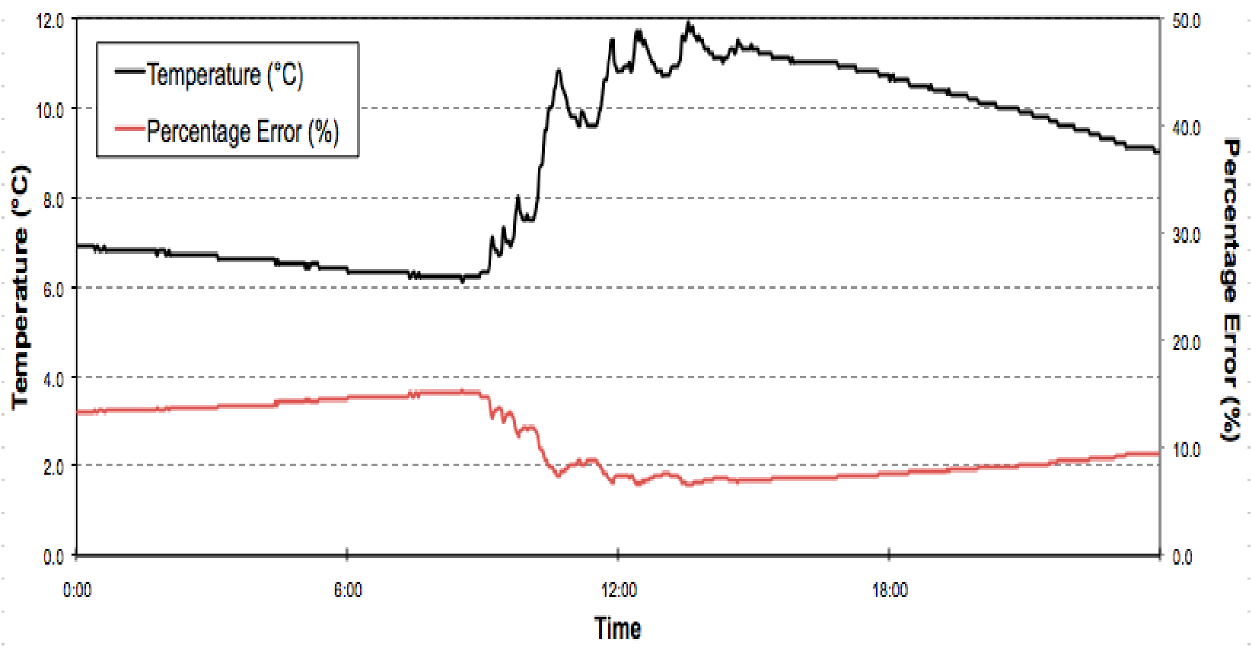


Figure 1.7: Relative of measured temperature and accuracy

Figure 1.7 shows the relative of measured temperature and sensor accuracy performance parameter in percentage unit. Before 12:00 AM the temperature is lower than 10°C, percentage error is higher than 10% . However, after 12:00, the temperature increases and the percentage error decreases

1.2.1.2 Response Time

Response time, is an expression of how quickly a sensor responds to temperature changes. Time constant is a particular case of response time, which is defined as the length of time it takes a sensor to reach 63% of a step temperature change.

The response time depends on heat capacity of and thermal resistance to sensor substrate, that means sensor with different type of thermocouple will give different average response time. A rapid response time is essential for accuracy in a system with sharp temperature changes [19]. Measured temperature d_{ik} of sensor S_i at time t_k is represent in the following equation

$$d_{ik} = d'_{ik} - (d'_{ik} - d_{i(k-1)}) \cdot e^{\frac{-I_{set}}{\tau}} \tag{1.1}$$

where τ is response time of sensor S_i at time t_k, I_{set} is the setting time on sensor made by engineer to give an output sensing value, d'_{ik} is the actual temperature

$$\Rightarrow d'_{ik} - d_{ik} = d'_{ik} - d_{i(k-1)} \cdot e^{\frac{-I_{set}}{\tau}} \tag{1.2}$$

An error fraction function is defined:

$$\frac{d'_{ik} - d_{ik}}{(d'_{ik} - d_{i(k-1)})} = e^{\frac{-I_{set}}{\tau}} \tag{1.3}$$

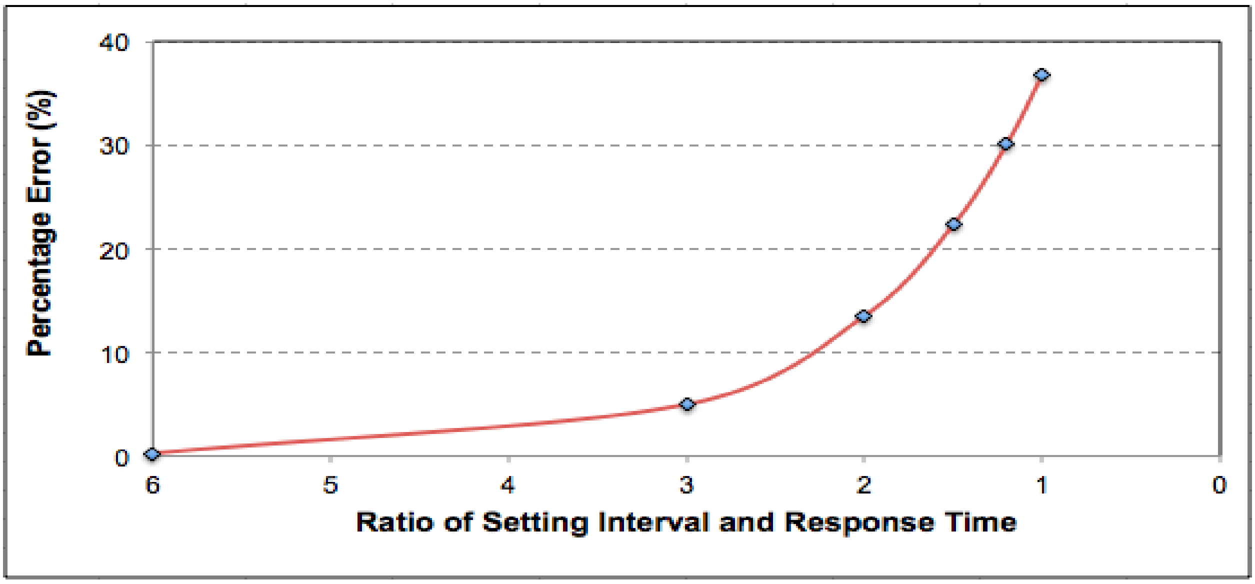


Figure 1.8: Relative of response time and percentage error

Figure 1.8 represent percentage error betweens actual temperature and measured value of sensor, which is inversely proportional to the ratio of setting interval and response time

1.2.2 Sampling Interval and Empty Value

Some definition of time in sampling data are defined:

- Setting interval I_{set} is the time, which is set by engineer on sensor, to give sensing data. I_{set} is greater or equal to the minimum response time of sensor
- Sampling interval I_{samp} is the period betweens two samples, which is made by request from user/application
- Sampling time t_{samp} is the time to observe sensor

When the fraction $\frac{I_{samp}}{I_{set}}$ is not an integer and each sensor starts with different time although they have the same sampling interval, empty value presents in sample.

This figure shows the of measured sensor and refer sensor in sampling interval 30 seconds, setting interval of each sensor is 120 seconds. Each sensor starts with different time although they have the same sampling interval. This leads to the presence of empty data when the system read sensors' reading at each sampling time.

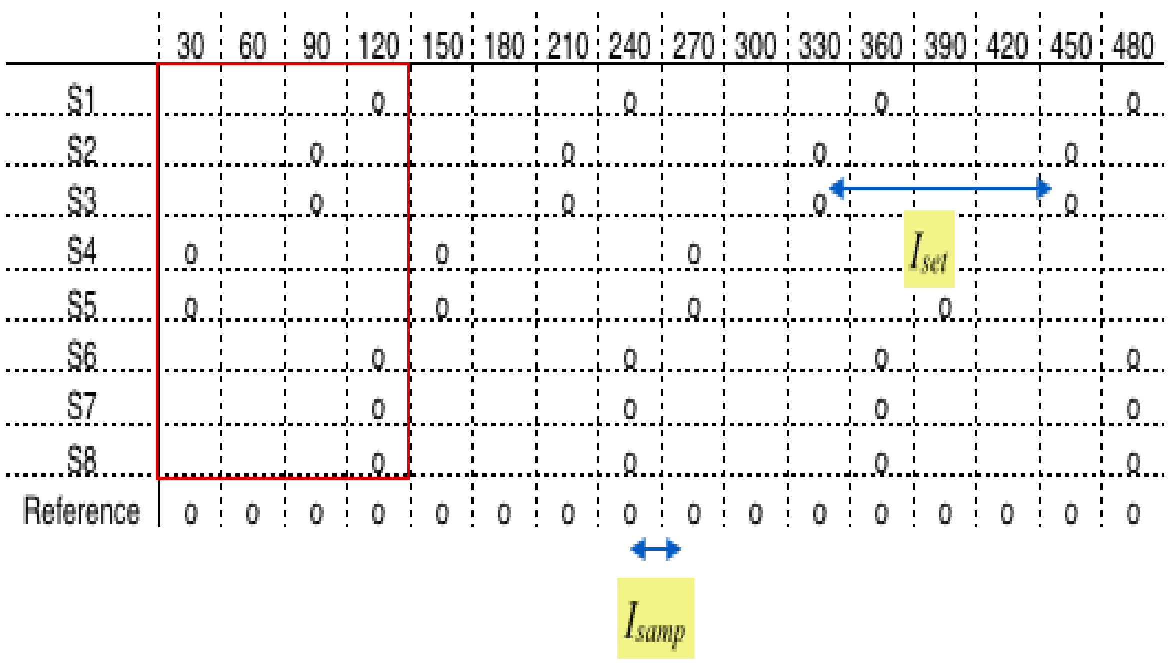


Figure 1.9: Data distribution of measured sensor and refered sensor

1.3 Thesis Objective and Contributions

This research aims to develop the heterogeneous data processing and estimating system (HDPES) that can provide a highly accurate sensed data and/or can estimate a desired data using the CPS-oriented and heterogeneous sensors in the cyber-physical smart home environment.

The contribution of this research is divided into 3 folds: (*i*)Specify a new framework for using CPS sensors with heterogeneous sensing data from cyber-physical smart home environments. Emphasis on resolving dynamic total error by selecting only some of input sensors using the minimum error first (MEF) algorithm; (*ii*) To propose a novel estimation method, minimum error method (MEM) to improve the accuracy of the parameter considered (temperature) at specific location; (*iii*)To study and analyse the relationship between total error and the performance of the proposed framework. By using a simulator, which is written in R language.

1.4 Thesis Outline

- Chapter 1 explains the research background, motivation, objective and contributions.
- Chapter 2 gives some related work, shows our designed proposed framework, proposed algorithm to reducing error in sensing data and estimating the desired parameter in CPS smart home environment
- Chapter 3 evaluates our proposed system by conducting some simulated studies and data analysis.
- Chapter 4 concludes the thesis, points some research challenges and draws our future works.

Chapter 2

Design of Heterogeneous Data Processing and Estimating System

2.1 Overview

Indoor applications developed intelligent diversification in many areas to bring utilities to the user in everyday life or toward the maximum saving of energy for electrical equipment in the smart home. The Home Automation field is expanding rapidly as electronic technologies converge. The home network encompasses communications, entertainment, security, convenience, and information systems [20]. Advancements in the fields of intelligent home systems such as doors control system for safety purposes, healthcare; the warning systems: fire alarm, gas leak detection; environment adjustment of living space.

This research focuses on the gathering, handling heterogeneous data from sensors of different appliances to serve the same purpose is to estimate and control a specific variable. Specific variables could be as temperature, illumination, humidity, occupants' locations, sound, etc. The proposed system can be applied to achieve this aim. However, in this research temperature, one of importance information for thermal comfort in smart home, is used as the particular parameter to study and analysis data, system performance.

2.2 Related Works

Temperature is the most basic environmental parameter. Almost the response of appliances in smart home are closely related to the temperature. Temperature has an interconnection influences with people's daily life. Some related works of temperature control in home environment can be found in [2], [3], [4] and [5] . These works considering in optimization accuracy of estimated temperature in room based on homogeneous sensors. We can se the overview through the following history graph:

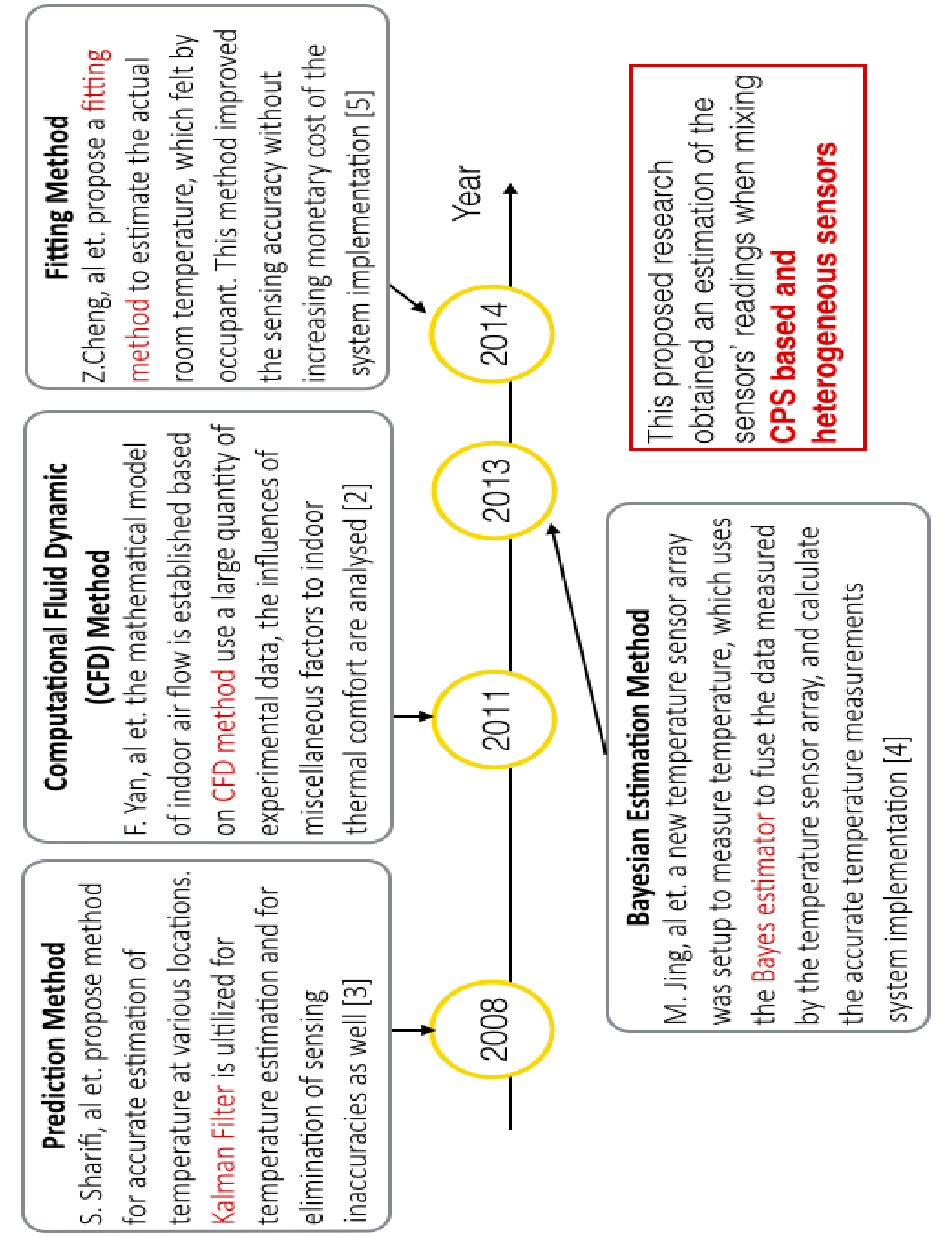


Figure 2.1: Related works of temperature estimation

2.2.1 Kalman Filterring (KF) Method

S.Sharifi, proposed method for accurate estimation of temperature at various locations on a chip considering the inaccuracies in thermal sensor readings due to limitations mainly on thermal sensor placement and sensor noise. Kalman filter (KF) is ultilized for temperature estimation and for elimination of sensing inaccuracies as well. This technique typically reduces the standard deviation and maximum value of temperature estimation errors by about an order of magnitude. The most important of this technique is efficiently in order to estimate the temperatures at the locations of interest where no sensor is available. Model order reduction is used to reduce the size of the model and generate a much smaller yet accurate linear system. Kalman Filtering estimates the temperature at different locations on the chip based on the inaccurate temperature readings at sensor locations and inaccurate power consumption estimates [2].

The thermal network is represented in state space form with the grid cell temperatures as states and the power consumption as inputs to this system. The outputs of this state space model are the temperatures at the sensor locations which can be observed by sensor readings.

2.2.2 Computational Fluid Dynamic (CFD) Method

F.Yan et al. based on CFD theory and Airpak software, the mathematical model of indoor air flow was established. CFD method use a large quantity of experimental data, the influences of miscellaneous factors such as the influences of wind velocity, wind frequency, and temperature to indoor thermal comfort are analysed. Considering the dynamics of the simulated wind velocity, indoor air flow is generally incompressible and of low turbulence, the turbulence model of Airpak.

This paper found when the wind velocity is in wave patterns, the increment of wind velocity can also bring preferable thermal comfort, even if the temperature reaches to a high level. Then specific values are given to some typical cases. The research can provide theoretic reference and beneficial experience to building ventilation design. The RNG $K-\varepsilon$ model is used, which is established based on the Standard $K-\varepsilon$ model, widely applied in the condition of turbulence [3]

2.2.3 Bayesian Estimation Method

M.Jing, a new temperature sensor array was setup to measure temperature, which uses the Bayes estimation to fuse the data measured by the temperature sensor array, and calculate the accurate temperature measurements system implementation. To measure temperatures, the new sensor array are used and made a rapid accurate judgment to the change of the environmental temperature. The sensor of the sensor array is quartz tuning fork non-contact temperature sensor based on the polymer line, with the high precision and the low cost and the simple manufacturing process. The system will use the Bayes estimation to fuse the measured temperature data the array get, and it overcomes that the precision is not high [4]

2.2.4 Fitting Method

Z.Cheng proposes a fitting method to estimate the actual room temperature, which felt by occupant. By using the design idea of CPS, a hybrid temperature control (HTC) system was proposed. It enables to monitor and maintain the room temperature in the desired interval. Through simulations and field experiments, the relationship between control performance and sensing accuracy was captured. This method improved the sensing accuracy without increasing monetary cost of the system implementation. To increase the sensing accuracy is crucial to improve the efficiency of the control system, linear regression method is used to establish the fitting function. With the help of Simulink Design Optimization toolbox in Matlab software, the mean value of square error is minimized based on trust region reflective algorithm [5]

2.3 Design of Heterogeneous Data Processing and Estimating System

2.3.1 Architecture and Design Framework

In this research, the proposed system can work with data come from two sources: traditional heterogeneous sensors and CPS sensors with different types of data collection methods: Look-up table data collection method and CPS-based data collection method.

In CPS-based data collection method, besides sensing data, sensing performance of CPS sensor is also sent directly to server by using CPS communication module. In the other hands, server will send a request message to database storage to retrieve static sensing performance of normal heterogeneous sensor, which does not have any mechanism to update sensing performance in timely manner

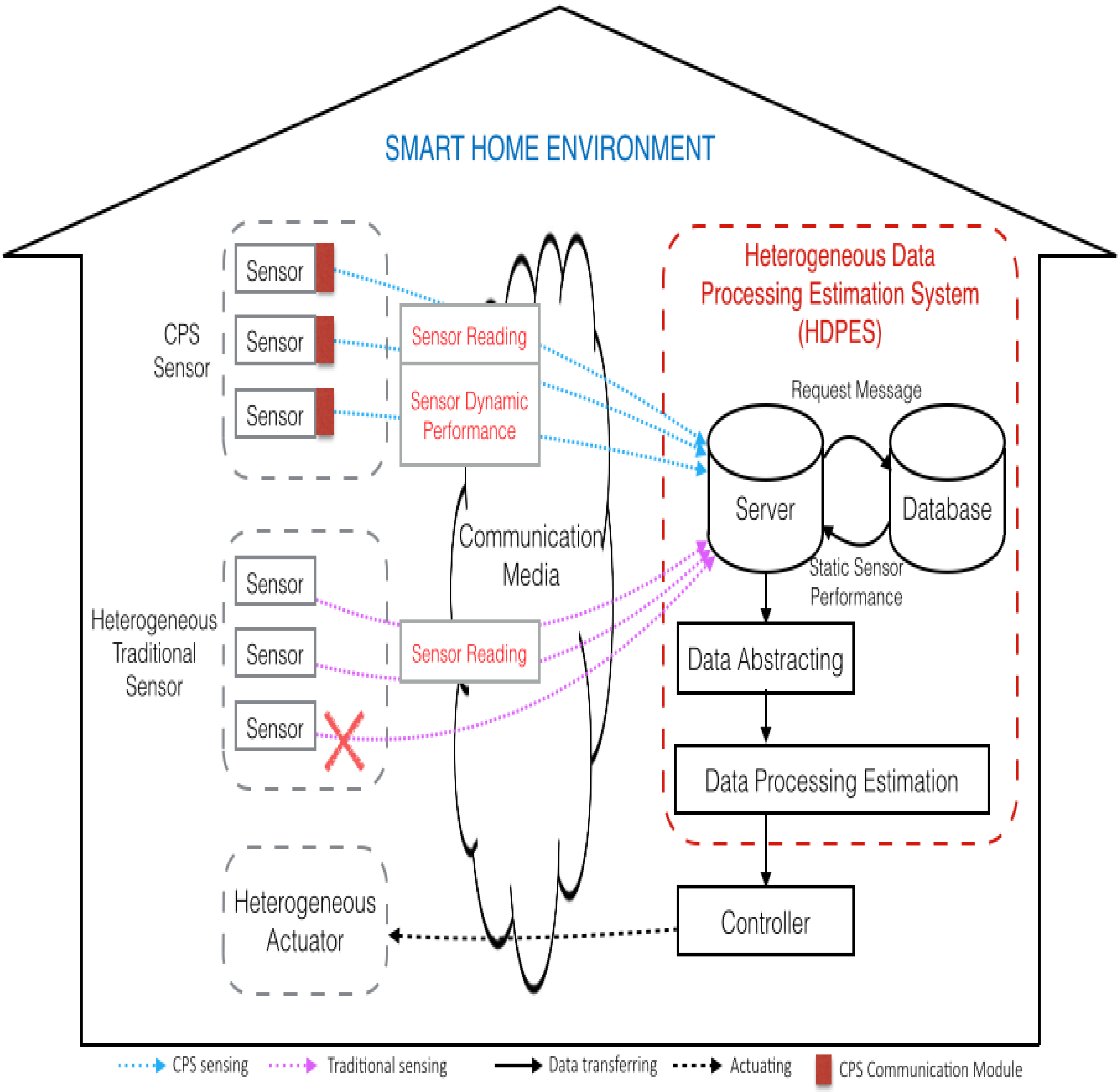


Figure 2.2: Heterogeneous Data Processing and Estimating System

Figure 2.2 shows the basic architecture of HDPES. In this architecture, cyber world and physical world are defined. These two worlds connect together through a communication media (e.g, WSAN). In particular, the WSAN comprises of two components: sensors and actuators. The sensors in the physical side send data included the environment temperature for inside room periodically to data storage in the cyber side.

Subsequently, the sensed data and performance parameter from server is extracted and sent to Data Abstracting block. Inside Data Abstracting block, total error from the influence of unpredictable changing environment and performance parameters on sensing accuracy is appreciated to reduce sensing error before supply input for next block. Based on the data from previous block, Data Processing and Estimating block use a special estimate method, which is proposed by the authors, to predict desired parameter at estimating point and send to controller. The mission of controller is to compute a control signal to achieve appropriate actuators to perform the corresponding task to influence the home temperature.

2.3.2 Data Abstracting

Functions of the Data Abstracting are:

1. To reduce sensing error of the input desired parameter by using the Fitting Method
2. To recover empty data in each sampling interval immediately by using a temporal model, Autoregressive Integrated Moving Average (ARIMA)

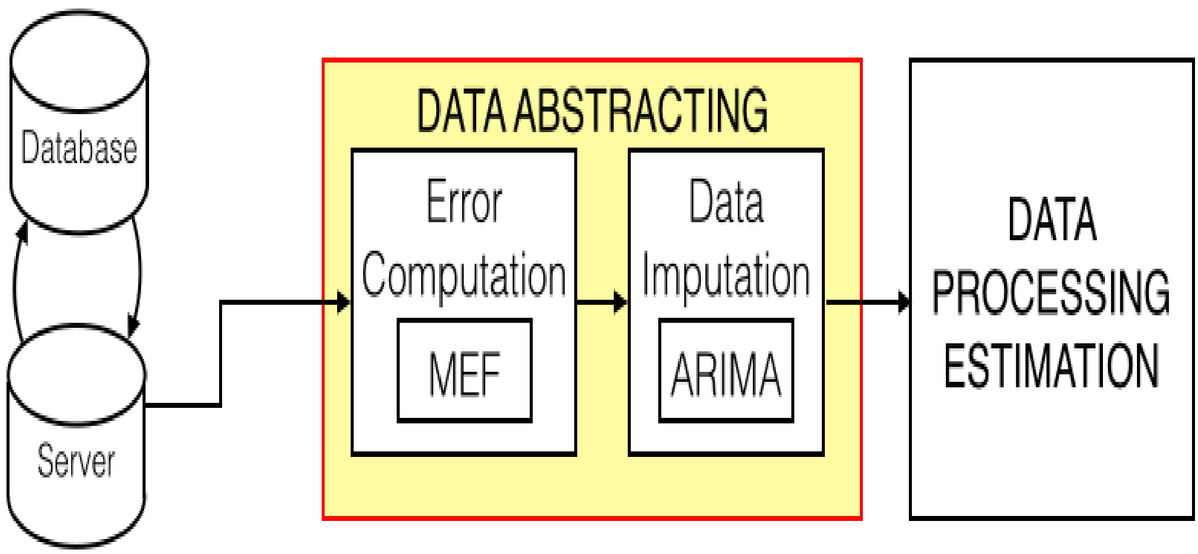


Figure 2.3: Data Abstracting block diagram

Data Abstracting consists of Fitting Method and ARIMA model with the goal to supply better input data for estimating a highly accurate value of the desired parameter in Data Processing and Estimating block

2.3.2.1 Error Computation

To optimise the accuracy of desired parameter, this method also makes an error computation, which is an adding bounding condition for choosing a good input data suitable with scenario sampling. To choose better sensors' readings with lower error for the inputs by using Minimum Error First (MEF) Algorithm.

The measured value d_{ik} of S_i at time t_k consists of actual value d'_{ik} and random generated error in range $\pm E_{ik}$:

E is the total expected error representing the overall change in the sensor performance which is caused by the unpredictable change of the surrounding environment as well as sensors unreliability. E sometimes is provided by the sensor specification.

In this simulated study, assuming that E of sensor S_i at time t_k : E_{ik} consist of error come from influence of operating range to accuracy and response time parameter

$$E_{ik} = \sum_{k=1}^{T} \varepsilon_{o,ik} + \varepsilon_{r,ik} \tag{2.1}$$

The block diagram of reducing sensing error mission is shown in figure 2.4 . In this block, server checks the existing of total error band in sensing performance information, and figure out E_{ik} if it does not in static prepared database.

Subsequently, Minimum Error First Algorithms (MEF) collects all the generated error from the total number of sensors, sort in increasing order, then choosing the number of input sensors (n) based on minimum E and send data output to Data Imputation block.

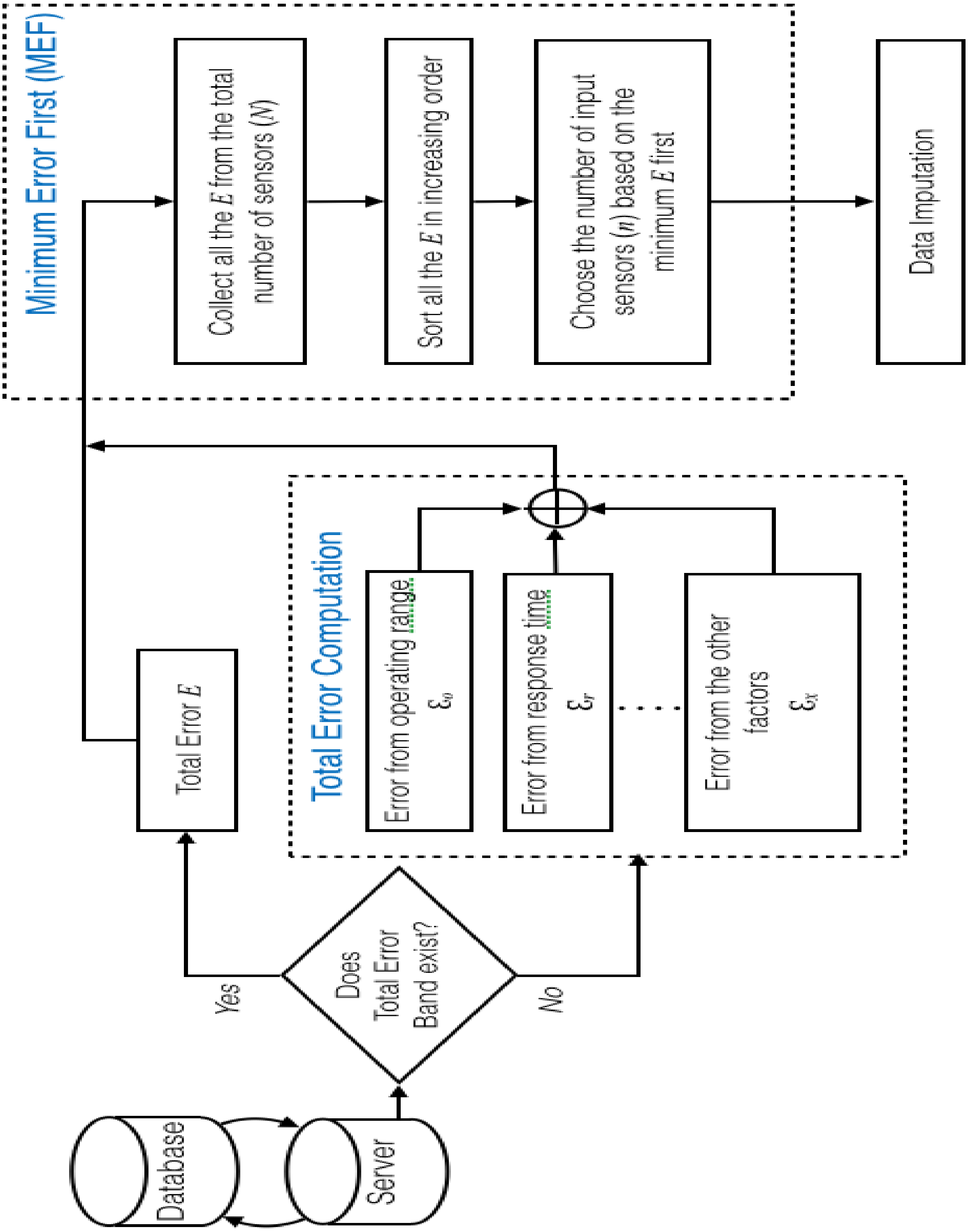


Figure 2.4: Error Computation block diagram

2.3.2.2 Autoregressive Integrated Moving Average model (ARIMA)

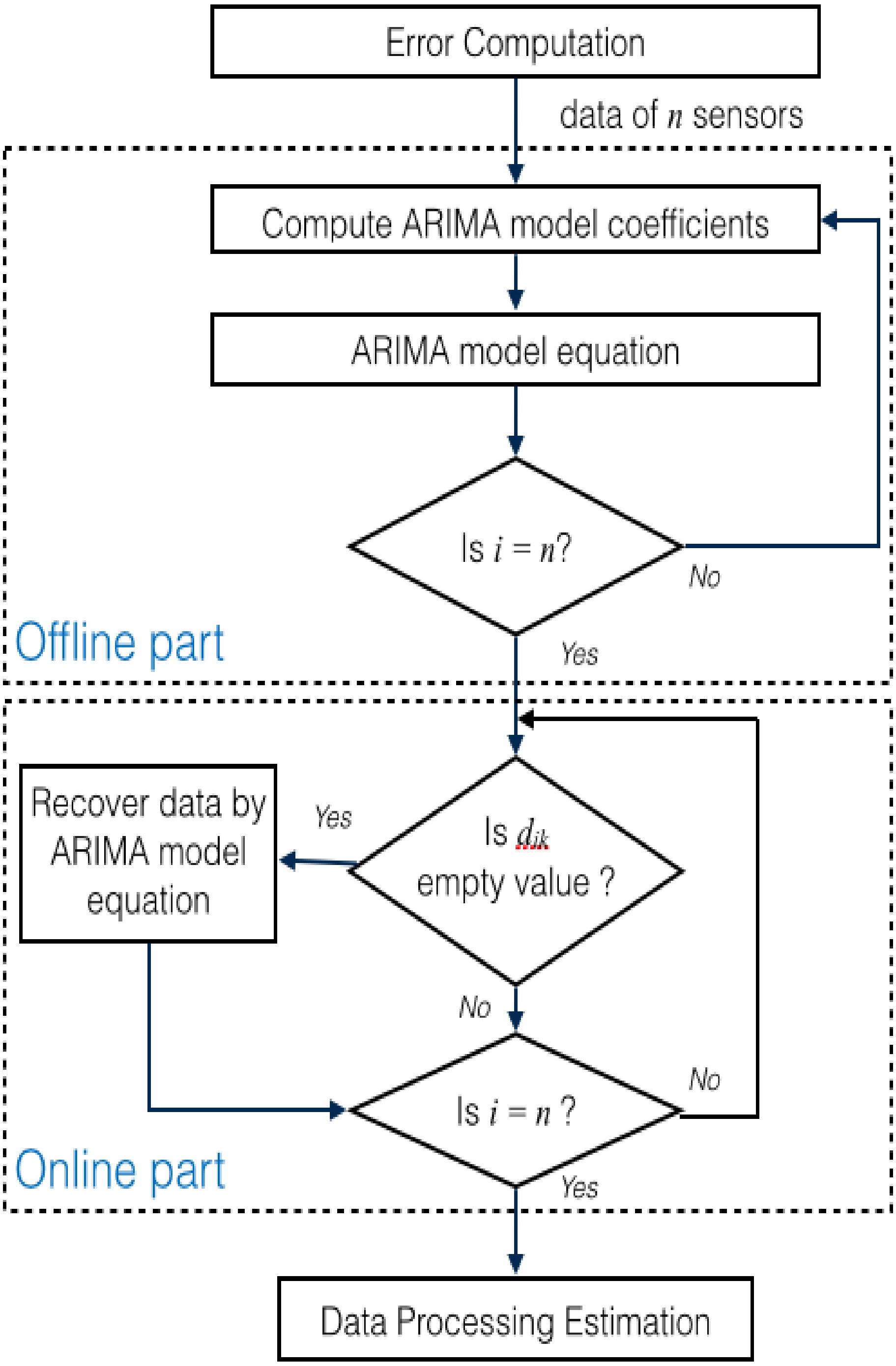


Figure 2.5: Data Imputation block diagram

The time series data that has inexplicable changes in direction, is analysed and build a temporal model by modelling it in ARIMA model ARIMA(p,q) models are a class of linear models, that are capable of representing stationary and non-stationary time series. ARIMA model rely heavily on autocorrelation patterns in data both ACF and PACF are used to select an initial model.

The model is generally denoted to as an ARIMA(p,d,q) model where, parameters p, d, and q are non-negative integers used to refer to the order of the auto-regressive, the amount of differencing, and moving average parts of the model respectively

$$\widehat{d_{ik}} = \Phi_0 + \Phi_1 d_{1(k-1)} + \Phi_2 d_{2(k-2)} + ... + \Phi_p d_{i(k-p)} - \Theta_1 \varepsilon_{1(k-1)} + \Theta_2 \varepsilon_{2(k-2)} + ... + \Theta_q \varepsilon_{i(k-q)} \quad (2.2)$$

An auto-regressive (AR) model is a simplified version of ARIMA model which describes random time-varying process. The AR model specifies that the output variable depends linearly on its own previous values. The AR model of sensor data with order p is defined as follows

$$\widehat{d_{ik}} = \Phi_0 + \Phi_1 d_{1(k-1)} + \Phi_2 d_{2(k-2)} + ... + \Phi_p d_{i(k-p)} \quad (2.3)$$

where p is the order of auto-regressive terms, Φ_1, Φ_2, ...,Φ_p are the parameter of the model

A q-order moving average model, or MA(q), is a linear regression of the current and previous error of a random series. A model with autoregressive terms can be combined with a model having moving average terms to get an ARIMA(p,q) model

$$\widehat{d_{ik}} = \Theta_0 + \Theta_1 \varepsilon_{1(k-1)} + \Theta_2 \varepsilon_{2(k-2)} + ... + \Theta_q \varepsilon_{i(k-q)} \quad (2.4)$$

where q is the number of moving average terms, Θ_1, Θ_2, ...,Θ_p, ε_q is white noise.

To recover empty data, ARIMA model of each sensor selected from the previous block, is contributed based on history data in offline part. Empty data is recover in real-time in online part.

2.3.3 Data Processing and Estimation

Functions of the Data Processing estimates the temperature by using Data Estimation Methods to obtain an accurate value at the considered point. This block uses the history embedded data output of the previous Data Imputation Block.

2.3.3.1 Average Method (AM)

The commonly used method for control system estimating the actual room temperature at estimating point is using the average value of data sensed by the equipped sensors [8]. This is the equation of average method:

$$\widehat{d_k^A} = \frac{\sum_{i=1}^{N} d_{ik}}{N} \quad (2.5)$$

2.3.3.2 Root Mean Square (RMS)

This is the improvement of average method to reach the higher accuracy of estimated value. In statistics, the root mean square value, also known as the quadratic mean, is a statistical measure defined as the square root of the arithmetic mean of the squares of a set of values. The RMS value is always greater than or equal to the average

$$\widehat{d_k^R} = \sqrt{\frac{\sum_{i=1}^{N} d_{ik}^2}{N}} \tag{2.6}$$

2.3.3.3 Fitting Method (FM)

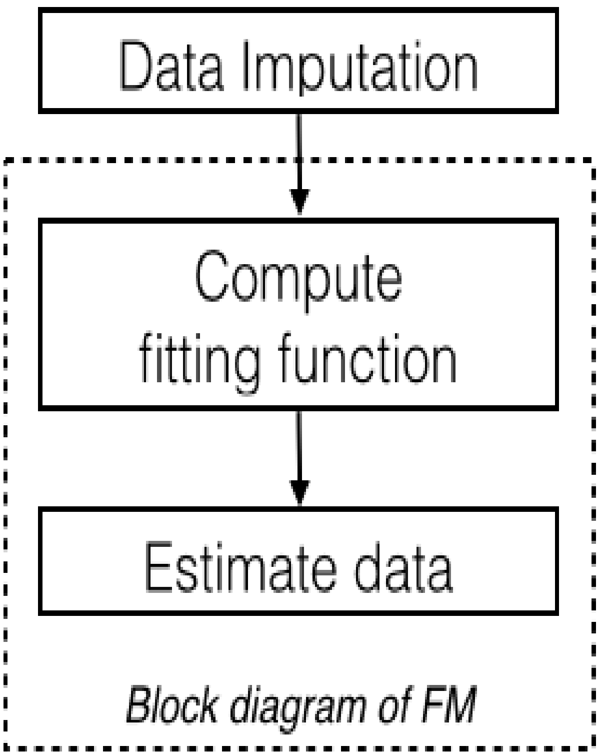


Figure 2.6: Fitting Method block diagram

This is the method proposed by Cheng [5]. In this research, fitting function is conducted by using training data based on history value of measured sensors, selected in previous block.

Linear regression was the first type of regression analysis to be studied rigorously, and to be used extensively in practical applications. In linear regression, the relationships are modelled using linear predictor functions whose unknown model parameters are estimated from the data. The block diagram of FM shows how to build a fitting function in fitting method.

This is the equation of fitting function:

$$\widehat{d_k^F} = \beta_1 d_{1k} + \beta_2 d_{2k} + ... + \beta_i d_{ik} \tag{2.7}$$

2.3.3.4 Most Minimum Error Method (MMEM)

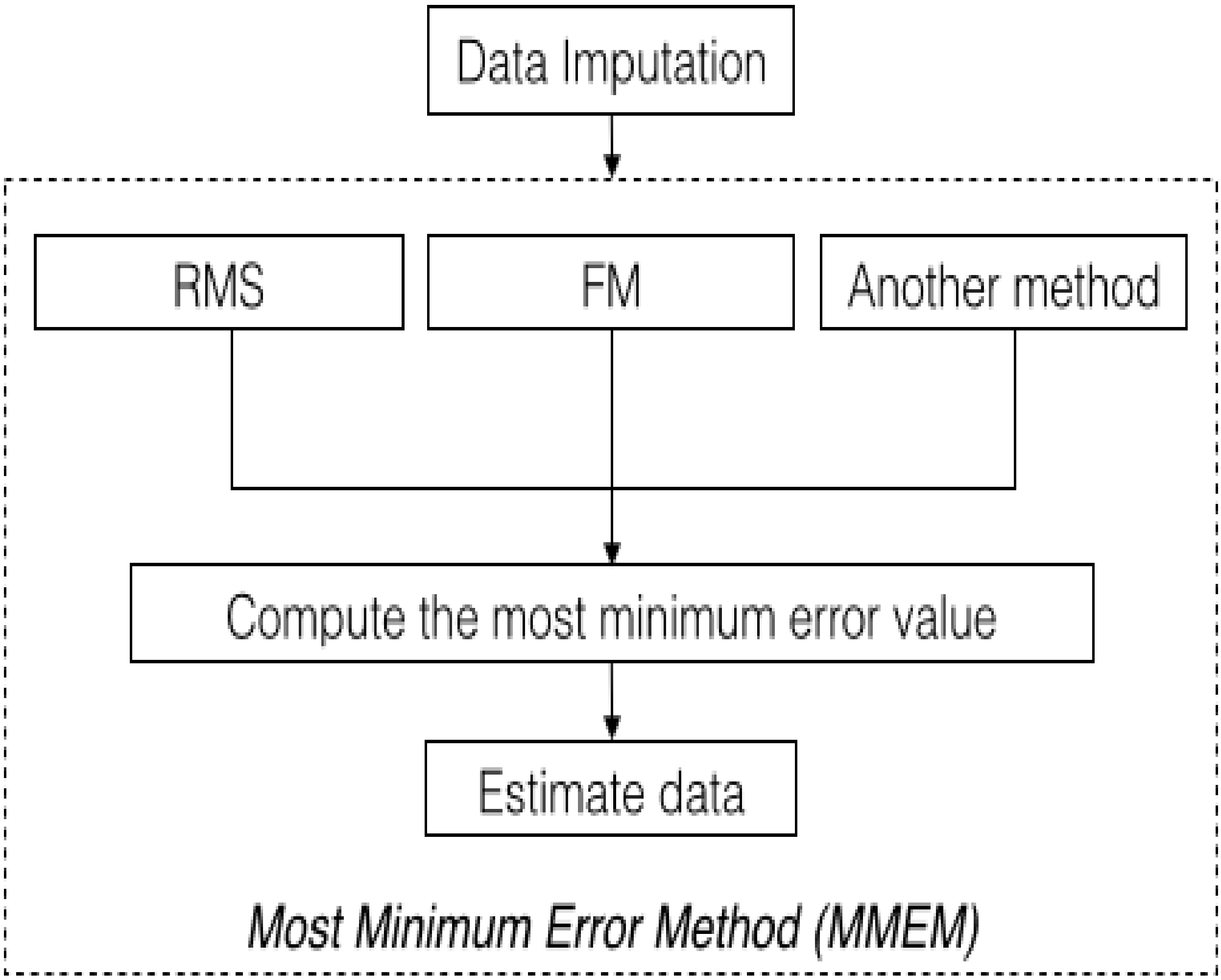


Figure 2.7: Most Minimum Error Method block diagram

This is the proposed method, using fitting method (FM), root mean square(RMS), to obtain an additional representation of the current values of sensors' readings. Choosing the readings with lowest error (closest to the estimated value obtained from FM).

According to the correlation coefficients between the measured sensors and refer sensor, make regression analysis which analyses the relationship between the data fitting. Therefore, when the mutual amendment is made in the use of a higher correlation coefficient failure without valid data of relevant measured sensors can be used to supplement it. Considering in this concept, MMEM is a method to compute the minimum error among the sensors' readings and FM,RMS or another estimate method in each sampling interval and choose the best values for estimated data.

This is the mathematical equation to represent for MMEM estimated value:

$$|d_{Ik} - \widehat{d_k^F}| = Min \begin{pmatrix} |d_{1k} - \widehat{d_k^F}| \\ |d_{2k} - \widehat{d_k^F}| \\ ... \\ |d_{ik} - \widehat{d_k^F}| \end{pmatrix} \tag{2.8}$$

$$\widehat{d_k} = d_{Ik}$$

Chapter 3

Evaluation of Heterogeneous Data Processing and Estimating System

3.1 Evaluation Methods

To understand how our framework will act in the physical word, simulations by using R software and simulated studies conducted from experiment data in the intelligent home environment are adopted to verify the proposed system. R is derived from an original set of notes describing the S and S-Plus environments written in 1990 by Bill Venables and David M. Smith when at the University of Adelaide. R is an integrated suite of software facilities for data manipulation, calculation and graphical display. R software is the commonly software using in statistical and computing

It is important from the research perspective, as well as from a practical view, to be able to decide on an algorithm that matches the domain and the task of interest. The standard way to make such decisions is by comparing a number of algorithms offline using some evaluation metric. Many evaluation metrics have been used to rank algorithms, some measuring similar features, but some measuring drastically different quantities.

To evaluate the performance of the proposed system, three of evaluation metric are computed: the root mean square error (RMSE), the mean absolute error (MAE) and the integral of absolute error (IAE)

3.1.1 Root Mean Square Error (RMSE)

The RMSE is a frequently used measure of the difference between values estimated by an algorithm and the values actually measured from the real environment. An algorithm estimation with respect to the estimated value, the RMSE value is defined as the square root of the mean squared error as written as:

$$RMSE = \sqrt{\frac{\sum_{k=1}^{T} \left(\widehat{d_{Vk}} - d_{Vk}\right)^2}{T}} \tag{3.1}$$

3.1.2 Mean Absolute Error (MAE)

The MAE is another statistical measurement that used to measure how close the estimated values are to the measured values. The MAE measures the average magnitude of the errors in a data set, over the verification sample of the absolute values of the differences between forecast and the corresponding observation, without considering their direction. In other words, it measures the accuracy for the continuous variables.

The MAE and the RMSE can be used together to analyse the variation in the errors of the data set. The value of RMSE will always be greater or equal to the MAE. The value of RMSE will always be greater or equal to the MAE

$$MAE = \frac{1}{T} \sum_{k=1}^{T} \left| \widehat{d_{Vk}} - d_{Vk} \right| \tag{3.2}$$

3.1.3 Integral of Absolute Error (IAE)

The IAE is a widely used performance metric in control community, which is recorded to measure the performance of the control application. The IAE is calculated as follows, where, t denotes total simulation time. In general, the larger the IAE values imply the worse the performance of the control algorithm

$$IAE = \int_0^k \left| \widehat{d_V}(k) - d_V(k) \right| dk \tag{3.3}$$

3.2 Simulation and Data Analysis

3.2.1 Setup, Scenario and Setting

Figure 3.1: Experiment intelligent home environment – iHouse

In this section, I verify and examine how the proposed Heterogeneous Data Processing Estimation System (HDPES) will behave in estimated data by making the simulation conducted with R software tool. In the simulation, I use the raw data from the experiments that were conducted at the intelligent house environment, iHouse, which is located at Nomi city, Ishikawa, Japan. Figure 3.1 shows the overview of iHouse. Three parts are included in the simulations.

Use the measured data of the sensor located at the centre of Bedroom A of the iHouse as a reference reading. Layout of the bed room A showed in figure 3.2

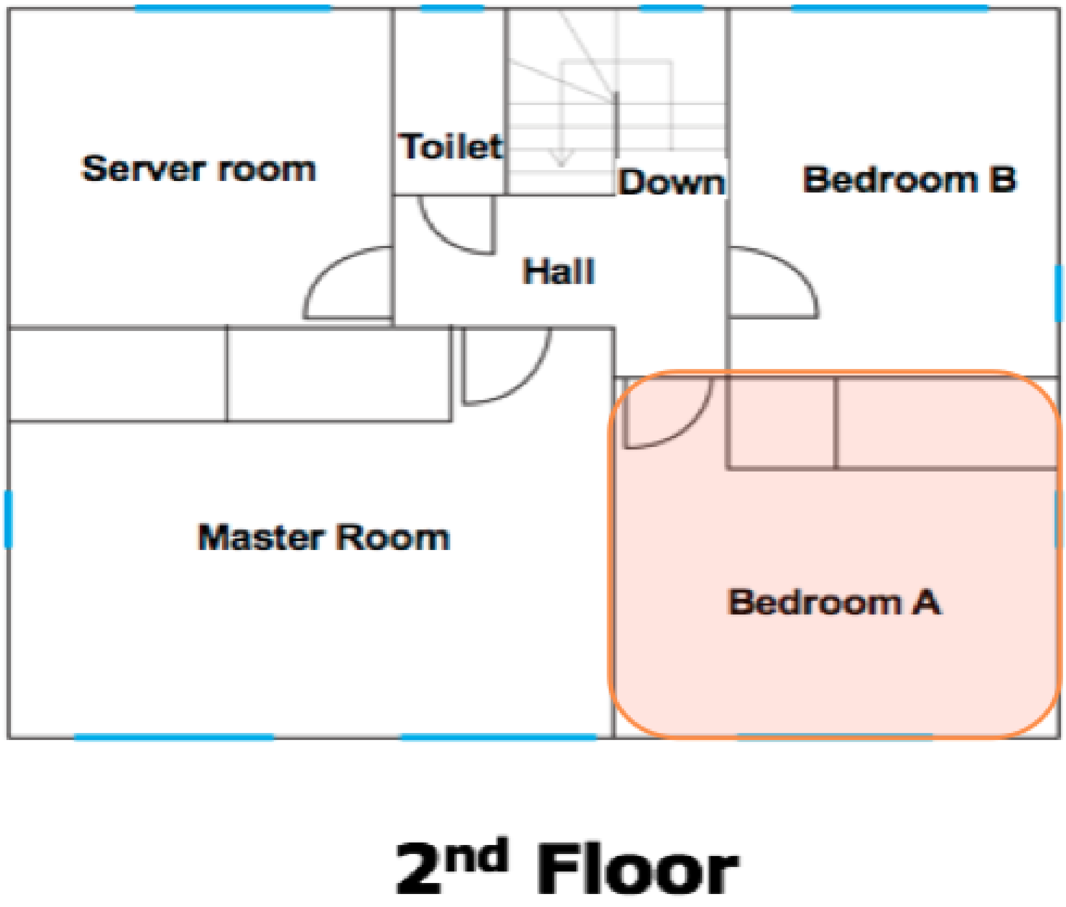


Figure 3.2: Room layout

Scenarios: Evaluate the proposed system, 8 CPS-sensors from 4 types (A,B,C,D) with different sensor performance. Data for those sensors is created based on actual measured data of the reference sensor with different generated random error with the maximum total error band as described in table 3.1

Table 3.1: Sensor Specification

Quality	*Type*	*Operating range* °C	*Accuracy* °C *(Optimum Values* °C*)*	*Response Time (seconds)*
2	A (High accurate)	-40 $\rightarrow$ 150	$\pm$ 0.25 (25)	2
4	B (Typical sensor)	-40 $\rightarrow$ 123.8	$\pm$ 0.25 (25)	17.5
1	C (Sensor inside wall-clock)	-40 $\rightarrow$ 70	$\pm$ 1.5 (25)	30
1	D (Sensor inside air-conditioner)	-50 $\rightarrow$ 80	$\pm$ 2 (25)	10

The reference source to refer the sensor performance is shown in table 3.2

Table 3.2: Sensor Specification Reference

Type	*Refer Source*
A (High accurate)	*http : //www.analog.com/media/en/technical-documentation/datasheets/ADT*7420*.pdf*
B (Typical sensor)	*http : //www.sensirion.com/fileadmin/user_upload/customers/sensirion/Dokumente/Humidity/Sensirion_Humidity_SHT*7*x_Datasheet_V*5*.pdf*
C (Sensor inside wallclock)	*http : //www.acurite.com/timex*115*atomicdigitalwallclock-withtemperature − moon − phasecalendar*75331*t.html*
D (Sensor inside air-conditioner)	*http : //www.smartclima.com/airconditionertemperature-sensor.htm* *http : //www.vishay.com/docs/*29053*/ntcintro.pdf*

3.2.1.1 Part I: Analysis of Data Imputation Block

In this part, the error computation and data imputation are combined in the system and evaluate the result . The data I used in simulation is conducted at the iHouse master room, which is mainly used for work and study. The values and parameters used in the simulation are shown in Table 3.3 .

Table 3.3: Simulation parameters and settings: Part I

Parameter	*Value*
$V_{room}(L \times W \times H)$: volume of room	$5.005m \times 4.095m \times 2.4m$
I_{set}: setting interval sensors	2mins
I_{samp}: sampling interval of system	30secs, 1min, 2mins, 3mins, 4mins, 6mins
t_{samp}: period time to observe data	1day
N: number of measured sensor	$2 \rightarrow 8$
Observation Time for reference sensor	15 and 16 - December 2013

3.2.1.2 Part II: MMEM performance

This part evaluate the performance of proposed algorithm Most Minimum Error method. The data is used in simulation is conducted at the iHouse master room, which is mainly used for work and study. The values and parameters, other than those shown in Table 3.3 , used in the simulation are shown in Table 3.4

Table 3.4: Simulation parameters and settings: Part II

Parameter	*Value*
I_{set}: setting interval sensors	2mins
I_{samp}: sampling interval of system	3mins
t_{samp}: period time to observe data	1day

All temperature sensor are used in this simulation have the same unit and meaning (to measured room's temperature). However, in practical environment, sensors can be different unit (Celsius, Kelvin, Fahrenheit) and different meaning or different function (measure room's temperature, measure temperature inside an appliance/ device, etc.). Because of these reasons, before apply MMEM algorithm, the unit of all used sensors must be synchronised to the standard unit (Celsius) by the proposed system. Besides, a threshold for inside room's temperature is necessary to make a boundary for temperature value in room.

Linear regression, root mean square method, minimum square error method, are the different represent of average method. These methods are easy to implement and save time in computing. The disadvantage of these methods is extremely sensitive to extreme values. Hence, using FM, RMS, AM for data sets of sensors' readings containing a few extreme values is not a good solution. In this case, median value of a large data set can be a better alternative (Gaussian).

With a few sensors, the present solution is to normalise all sensor's readings d_i by using percentage of relative value between sensor reading and measured temperature of reference sensor $\widehat{d_{ref}}$.

$$d_i \rightarrow d_i' = \frac{d_i}{\widehat{d_{ref}}} \cdot 100$$

3.2.2 Results

3.2.2.1 Part I: Analysis of Data Imputation Block

The results we report in figure 3.3 is conducted by using average method on reduced error data and raw data for estimating the desired parameter at estimating point RMSE is shown of the system with and without the Data Imputation Block. With data imputing, RMSE decreases up to 16% (at $I_{samp} = 30$s)

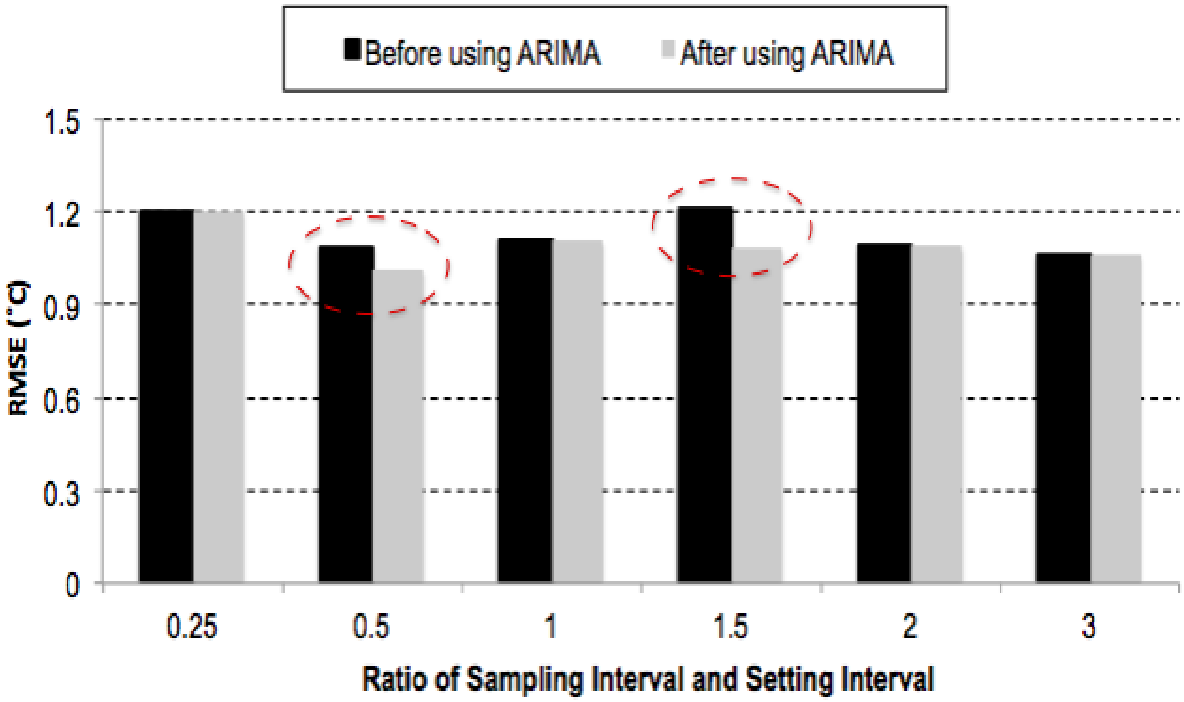


Figure 3.3: Performance of Data Imputation

3.2.2.2 Part II: MMEM performance

This simulation is to evaluate the performance of MMEM in estimating accurate temperature. In this simulation, the input sensed data of measured sensor are already imputing to fulfil all empty value, which caused by different setting interval and the start time to sample of each sensors. Apply MMEM in the following 4 cases:

- 8 sensors (4 type A, 2 type B, 1 type C, 1 type D) $\rightarrow$ worst case

- 4 sensors (2 type A, 2 type B) → not shown
- 3 sensors (2 type A, 1 type B) → not shown
- 2 sensors(2 type A) → best case

3.2.2.2.1 Accuracy

After using Minimum Error First algorithm to choose sensors, which has lower error band, the estimated temperature is closer to the measured temperature of reference sensor. The difference for 4 cases with different the number of high error band sensors, the result is better with the decreasing of randomly generated error E. In figure 3.4 , case of using 2 high accurate sensor is the best case, and the case of using 8 sensors (included large error band sensor, which are in air-conditioner and wall-clock).

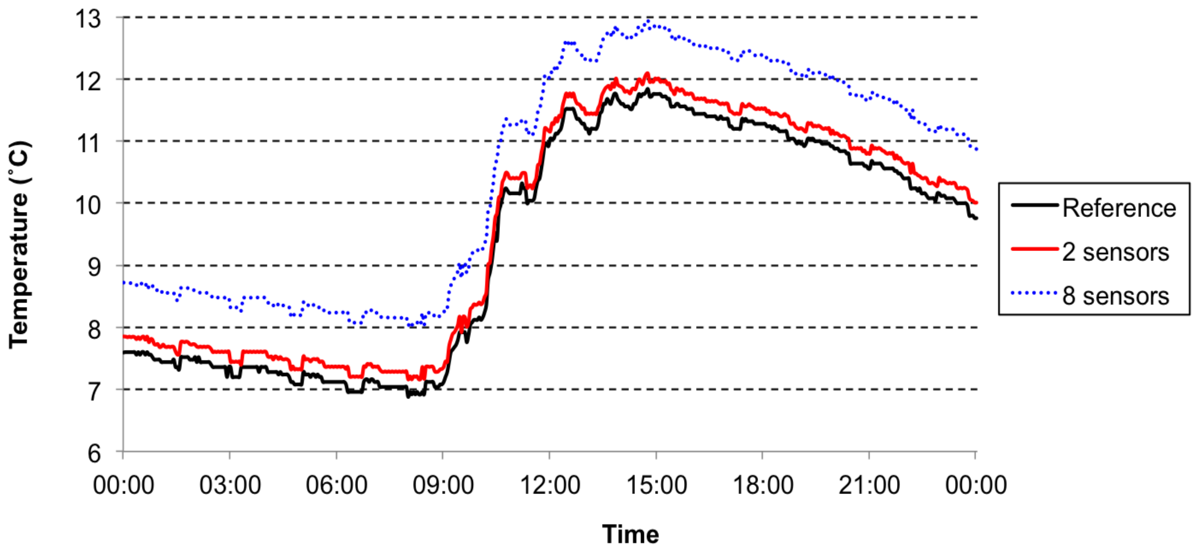


Figure 3.4: Performance of MMEM

The result of MMEM also represents the influence of Data Imputation block in case of sampling interval is : 30secs, 1min, 2mins, 4mins and 6mins. The results is reported in figure 3.5 is conducted by using Most Minimum Error method on reduced error data and raw data for estimating the desired parameter at estimating point. RMSE, MAE is shown of the system with the Data Imputation Block. With data imputing, RMSE in case of sampling interval equal to 3 minnutes is the best case in this simulation (0.27°C)

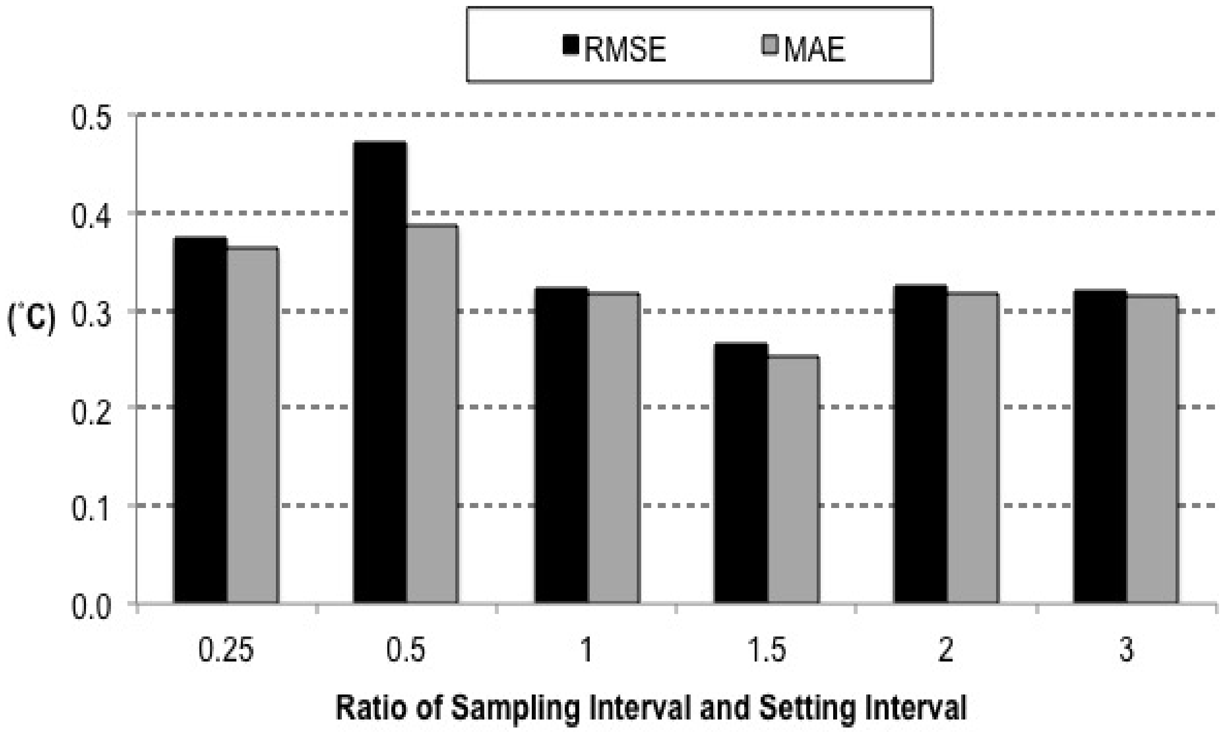


Figure 3.5: Performance of MMEM in different sampling interval

In figure 3.6, 3.7 and 3.8, the estimated value of two methods: Fitting Method(FM) and the proposed method Most Minimum Error Method(MMEM) are compared with temperature from reference sensor. The difference for both RMSE, MAE, IAE between MEM and FM increases with the average of total error E. However, in this simulation, the increasing of RMSE, MAE, IAE is not much.

The average of total error is represented in the following equation:

$$E = \frac{\sum_{j=1}^{n} \frac{\sum_{k=1}^{T} E_{ik}}{T}}{n} \tag{3.4}$$

- 8 sensors (4 type A, 2 type B, 1 type C, 1 type D): 40.10%
- 4 sensors (2 type A, 2 type B): 33.78%
- 3 sensors (2 type A, 1 type B): 32.61%
- 2 sensors(2 type A): 30.26%

- RMSE of MMEM is 87% less than for FM.
- MAE of MMEM is 89% less than for FM.
- IAE of MMEM is 34% less than FM.

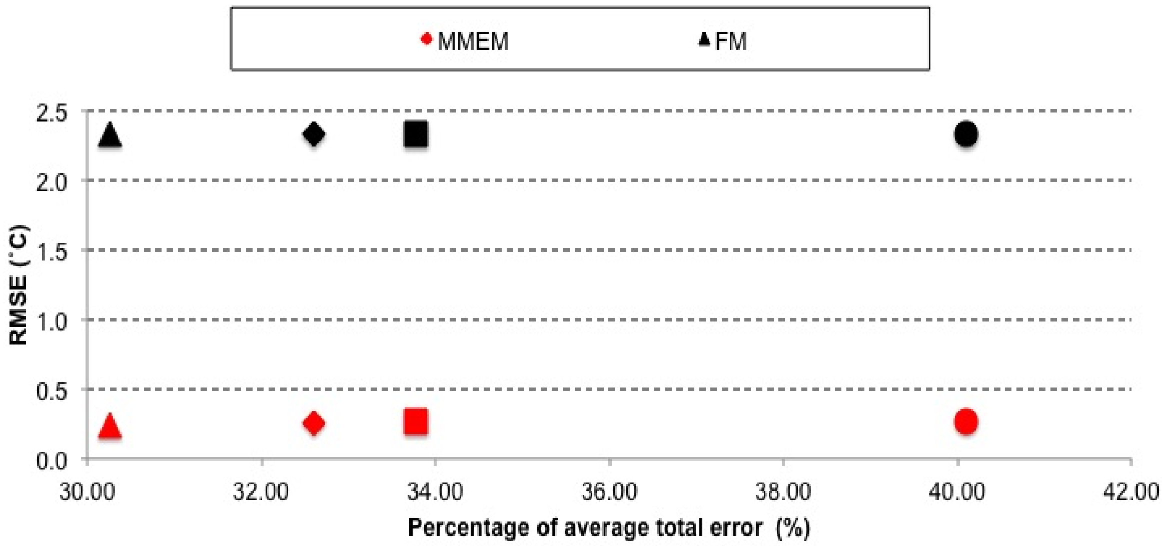


Figure 3.6: Performance of MMEM vs FM: RMSE

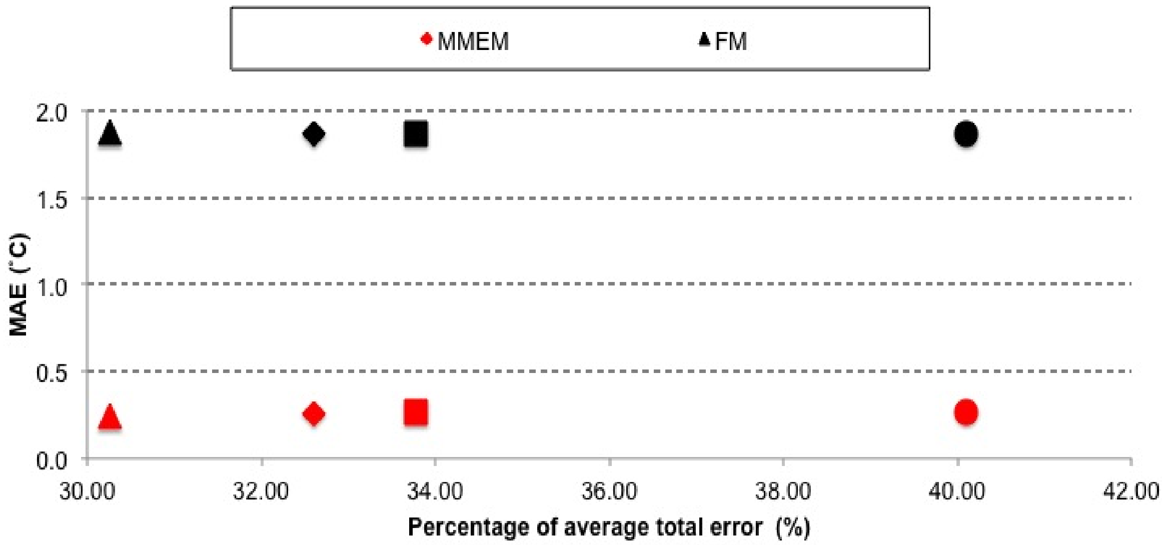


Figure 3.7: Performance of MMEM vs FM: MAE

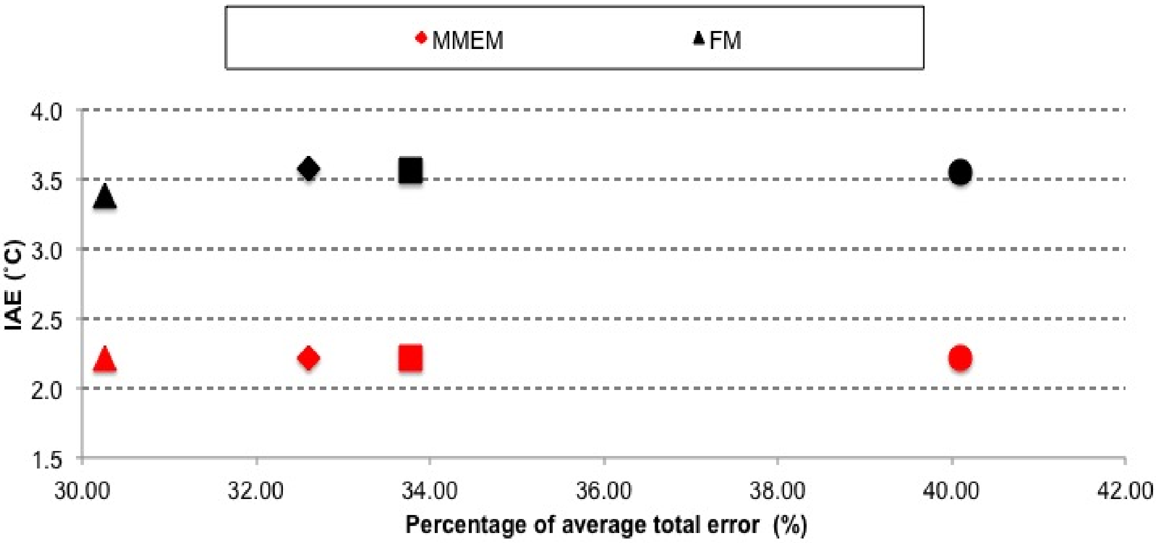


Figure 3.8: Performance of MMEM vs FM: IAE

3.2.2.2.2 Elapsed Time

These simulations take place on a computer with following specification: CPU speed (1.7GHz Core i7), Memory (8GB), OS: Macintosh. The result in figure 3.9 is conducted after taking account in 10 times to get the average value of elapsed time.

Elapsed is directly proportional to the number of sensors. Elapsed time of 2 sensors is 58% less than the elapsed time of 8 sensors

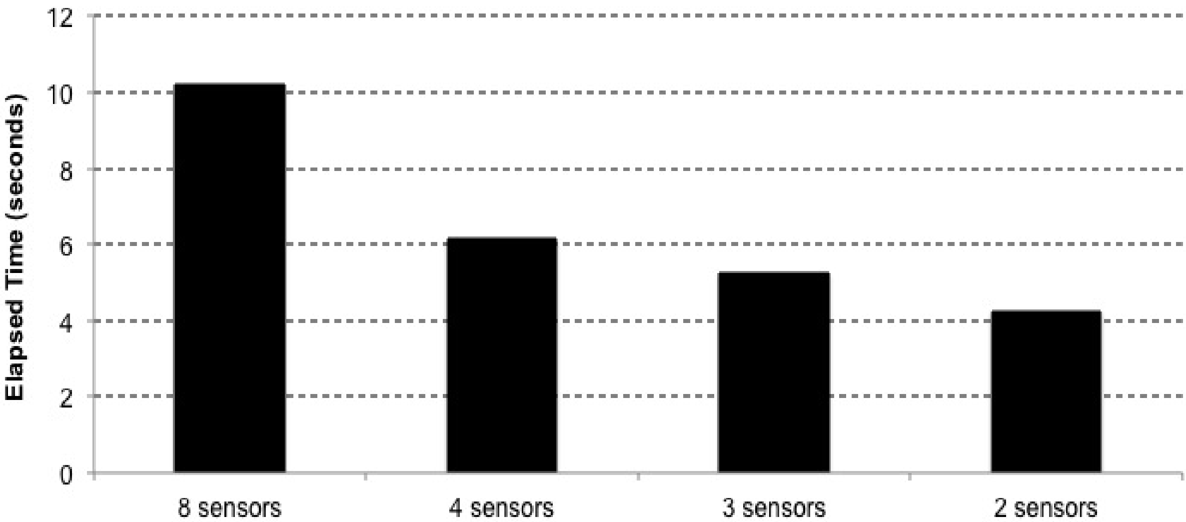


Figure 3.9: Performance of MMEM: Elapsed time

Chapter 4

Conclusion

4.1 Concluding Remarks

In this research, a new framework is specified for using CPS sensors with heterogeneous sensing data from cyber-physical smart home environments. The design of Heterogeneous Data Processing and Estimating System is presented with expected collection data method for heterogeneous sensing. Emphasis on resolving dynamic total error by selecting only some of input sensors using the minimum error first (MEF) algorithm. A novel estimation method, minimum error method (MEM) is proposed to improve the accuracy of the parameter considered (temperature) at specific location. The relationship between total error and the performance of the proposed framework is studied and analysed by using a simulator, which is written in R language.

By comparison simulation result of different sensors, which are equipped on different appliances, to evaluate and verify HDPES; the algorithm of MMEM inside the proposed system can obtained to estimate the desired parameter with highly accuracy at estimated point.

Sensing performance of specified sensors, which is designed for different purpose, have various factors (e,g resolution, precision, accuracy, hysteresis, operating range, humidity). Through data analysis, the affect of environment and sensing performance factors have a strong impaction on accuracy of sensed data, this is also the reason of difference betweens estimated and actual value.

For the continuous work, I will make a deeply detail survey about sensor performance factors of usual sensor types in home appliances and influence of these factors on sensed value. This survey supports to do a completely error computation to give a better condition for choosing input data in processing and estimating desired parameter.

4.2 Research Challenges and Directions

The complexity of Cyber-Physical Systems, resulting from their intrinsically distributed nature, the heterogeneity of physical elements (i.e. sensors and actuators), the lack of reliability in communications, variability of the environments in which they are employed, makes data analysis, processing and estimating as a complex task[6]. Base on these views, my future research direction will aim to expand the proposed system HDPES with a fully API such as an interface to integrate between the proposed system and the other home application/controller, CPS oriented sensor. This API will be developed in CPS-base oriented to obtain these below folds:

1. To consider the case when abnormally long response time sensor causing communication delay longer than the communication cycle

2. Spatial correlation also need to be considered same as temporal correlation has been studied in this research

3. To formally define the interface between the already in use controllers and its protocols on one hand and the proposed system on the other hand

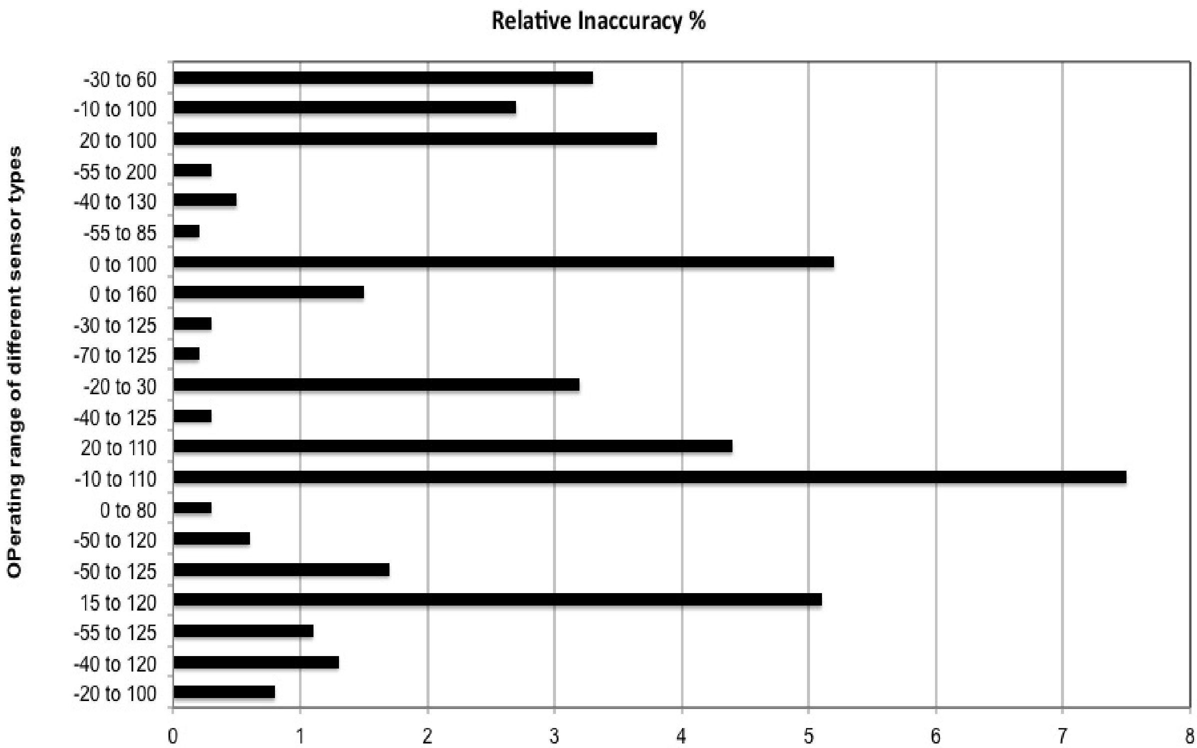
I expect that the machine learning technique can be developed for improving the performance of error computation and estimating algorithm in the proposed system. This is a new and promising research domain for CPS approach especially in smart home environment. Since the outcomes of the HDPES system using the co-design framework can be varied dynamically due to the unpredictable changing of environment factors, the machine learning technique may be able to improve the predictive variables accuracy of estimating method, it may leads to the entire system is adaptable to the dynamic change of smart home environment under different sensing performance of CPS-based oriented sensors.

Appendix A

Survey of Sensor Operating range

Use data from Smart Temperature Sensor Performance Survey [7], this graph show the affect of measured temperature to sensor accuracy of many type of sensor Relative inaccuracy corresponds to the slope of an imaginary boo placed around the sensor's error by this formula:

$$\frac{\text{PPIA}}{\text{Specified temperature range}} \cdot 100$$



Name	Component materials		Operating temperature range Overheat limit in ( )	Properties	Standard thermal EMF table, and authority
	+ side	-side			
Platinum-Rhodium	Contains 20% Rhodium. Platinum-Rhodium alloy	Contains 5% Rhodium. Platinum-Rhodium alloy	300 to 1500°C (1800°C)	Usable at high temperatures. Small thermal EMF. Otherwise, same as Type R.	Appendix Table B22
	Contains 40% Rhodium. Platinum-Rhodium alloy	Contains 20% Rhodium. Platinum-Rhodium alloy	1100 to 1600°C (1800°C)		
Tungsten-Rhenium	Contains 5% Rhodium. Platinum-Rhodium alloy	Contains 26% Rhodium. Platinum-Rhodium alloy	0 to 2400°C (3000°C)	Suitable for reducing environments, inert gasses, hydrogen gas. Fragile.	Appendix Tables B20 ASTM E988-84
	Contains 3% Rhodium. Platinum-Rhodium alloy	Contains 25% Rhodium. Platinum-Rhodium alloy			Appendix Table B22 ASTM E988-84
	Tungsten	Contains 26% Rhodium. Platinum-Rhodium alloy			
Platinel	Alloy, primarily Palladium, Platium, and Gold	Alloy, primarily Gold and Palladium	0 to 1100°C (1300°C)	High resistance to abrasion. Thermal EMF nearly the same as that of Type K.	Appendix Table B22 NBS Journal of Research, vol. 68C. N08
Chromel/ Gold-Iron	Alloy, primarily Nickel and Chrome o4?0 (Chromel)	Contains 0.07 mole% Iron. Gold-Iron alloy	1 to 300K	Thermal EMF relatively large at 20 K and below. Good EMF linearity.	ASTM SPT430 Appendix Table B21

Appendix B

Source code for MEF algorithm

```
function  MEF(data,n)
{
% 'data' is a matrix include sensing data from all measured
   sensor,
    nrow <- nrow(data)
    ncol <- ncol(data)
    %Initial the error array
    E <- matrix(data = NA,nrow,ncol)
    estimate <- c()
    for(i in 1 : nrow)
    {
        for(j in 1:ncol)
        {
        %Estimate Total Error of sensors' readings
        % e.o : the error from operating range, e.r: the error
           from response time
            E[i,j] <- e.o[i,j]  + e.r[i,j]
        }
    }
    % Estimate the average value of total error of each sensor
    E.average <- colMeans(E)
    % sort() function return array of average total error of all
        measured sensors in increasing order
    E.average.sort <- sort(data.frame(E.average), decreasing=
       FALSE)
    list.sensors <- colnames(E.average.sort[c(1:n)])

    %Return data of n selected sensors
    return(data[,c(list.sensors)])
}
```

Appendix C

Source code for Data Imputation

```
function dataimputation(m,olddata,order,samples)
{
    %m: data of selected sensor, olddata: history data of
       selected sensors, samples: the number of predicted value
       by ARIMA model
    model <- arimamethod(m,order)
    pdata <- m
    i <- 1,  j <- 1
   avgrow <- m[,ncol(m)]
   counter  <- 1

    % Offline part
    % Contribute arima model for each sensor in selected sensors
       from Error Computation
    for(c in 1:ncol(m))
    {
                fit <- arima(ts(olddata[,j]),order)
                pdata[,c] <- forecast.Arima(fit,samples)$upper
                   [,2]
    }
    % This can be separated in Online part
    % Recover empty values
    for(i in 1:nrow(m))
    {
         for(j in 1:ncol(m))
        {
            val <- m[i,j]

          % Check which is the empty value
            if(!is.na(val))
```

```
            next
else
{
    if(i == 1 || i == 2)
    {
                    k <- 1
                    while(k<=10)
                    {
                            if(!is.na(m[k+1,j]))
                            {
                                    m[i,j] <- m[k+1,
                                        j]
                                    break
                            }
                            k = k+1
                    }
    }

    % arg1, arg2, arg3, are coefficients of ARIMA
       model
    % pval[1] , pval[2] is the 2 previous empty
       value of current value
    else
    {
            arg1 <-  as.numeric(model[1,j])
            arg2 <-  as.numeric(model[2,j])
            arg3 <- as.numeric(model[3,j])

            x <- 1
            pval <- c()
            k <- i

    % Find 2 previous non-empty value
            while(x <= order[1] && k>=2)
            {
                    if(!is.na(m[k-1,j]))
                    {
                            pval[x] <- m[k-1,j]
                            x = x+1
                    }
                    k = k-1
            }
```

```
                    pval <- as.numeric(pval)

                    %Apply arima model equation
                    if(!is.na(arg1) && !is.na(arg2))
                    {
                         val <- arg1*pval[1] + arg2*pval[2] +
                             arg3*(pval[1]-pval[2])
                         if(val > 0.8*rowMeans(m[i-1,]))
                                             m[i,j] <- val
                         else
                                             m[i,j] <- rowMeans(m[i
                                                  -1,])
                    }
                    else
                              m[i,j] <- pdata[i,j]

               }
          }
     }
  }
  return(m)
}
```

Appendix D

Source code for MMEM algorithm

```
function estimate = MMEM(data, refer)
{
% 'data' is a matrix include sensing data from all measured
   sensor, average value (form AM), fittied value (from FM)
    nrow <- nrow(data)
    ncol <- ncol(data)

    %Initial the error array
    e <- matrix(data = NA,nrow,ncol)

    estimate <- c()
    for(i in 1 : nrow)
    {
        for(j in 1:ncol)
        {
        %Calculate Absolute Error betweens temporal estimated
            data and referent data
            e[i,j] <- abs(data[i,j]-refer)
        }
        % which.min() function return index of minimum value
        min <- which.min(e[i,])

        estimate[i] <- data[i,min]
    }
    %Return Estimated value
    return(estimate)
}
```

Appendix E

Source code for FM algorithm

```
function FM(olddata,data)
{
        % Contribute fitting function from history data of n
            selected sensor from Data Abstracting block in
            previous day
        formula <- "olddata[,9] ~ 0 "

        nrow <- nrow(data)
        ncol <- ncol(data)

        m <- data

        for(j in 1:ncol)
        {
                formula <- str_c(formula,"+ olddata[,",j,"]")
        }
        %fit : contains fitting function
        fit <- lm(as.formula(formula))

        %predict new temperature of each selected sensor in
            current day
        d.fit <- predict(fit,m)

        %m: included fitted values and sensors' readings of n
            selected sensors
        m <- cbind(m,d.fit)
        colnames(m) <- c(colnames(data),"FM")

        return(m)
}
```

Bibliography

[1] E.A. Lee, "Cyber-Physical Systems Design Challenge," *International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing (ISORC)*, pp.363-369, 2008.

[2] S.Sharifi, "Accurate Temperature Estimation for Efficient Thermal Management," *IEEE 9th Quality Electronic Design (ISQED)*, pp.137-142, 2008.

[3] F. Yan, "Simulation of Indoor Dynamic Thermal Comfort based on Computational Fluid Dynamic," *IEEE Intelligent Computation Technology and Automation (ICICTA)*, pp.860-865, 2011.

[4] M.Jing,"Research of high precision quartz temperature sensor system based on data fusion technology," *IEEE Measurement, Information and Control (ICMIC)*, pp.1051-1054, 2013

[5] Z. Cheng, Y. Tan, and A.O. Lim, "Fitting method for hybrid temperature control in smart home environment," *IEEE International Conference on Modelling, Identification and Control (ICMIC)*,pp.300-305,2014.

[6] G.Andrea, S.Giandomenicoand, and V.Andrea, "Rainbow: an Intelligent Platform for Large-Scale Networked Cyber-Physical Systems," *International Workshop on Networks of Cooperating Objects for Smart Cities (UBICITEC)*, pp.70-85, 2004.

[7] K.A.A. Makinwa, "Smart Temperature Sensors in Standard CMOS," Procedia Engineering, pp. 930-939, Sept. 2010.

[8] M. Maasoumy, Q. Zhu, C. Li, F. Meggers, and A. Sangiovanni Vincentelli, "Co-design of control algorithm and embedded platform for building HVAC systems," *In Proc. of the ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 61-70, 2013

[9] H.Sandberg, M.Rabi, M.Skoglund and K.H.Johansson, "Estimation over Heterogeneous Sensor Network", *In Proc. of the 47th IEEE Conference on Decision and Control*, pp.4898-4903, 2008

[10] W.Elmenreich, R. Leidenfrost, "Fusion of Heterogeneous Sensors Data," *IEEE Intelligent Solutions in Embedded Systems*, pp.1-10, 2008

[11] N.Nower, Y.Tan, Y.Lim, "Efficient Temporal and Spatial Data Recovery Scheme for Stochastic and Incomplete Feedback Data of Cyber-physical Systems," *IEEE 8th International Symposium on Service Oriented System Engineering (SOSE)*, pp.192-197, 2014

[12] R.H.Shumway, S.S.David, "Time Series Analysis and Its Applications," *Springer Text in Statistics*, pp.83-171, 2010

[13] B.G. LipTk, "Instrument Engineer Handbook: Process Control and Optimization," *CRC Press Taylor and Fracis Group, 4th ed.*, 2006

[14] R.H.Shumway, S.S.David, "Vector Autoregressive Models for Multivariate Time Series," *Springer Modeling Financial Time Series with S-PLUS*, pp.385-429, 20006

[15] M.C.Vurcan, O.B.Akan, I.F.Akyildiz, "Spatio-temporal correlation: theory and applications for wireless sensor networks," *Springer Sensor Networks and Configuration*, pp.75-98, 2007

[16] J.Yick, B.Mukherjee, D.Ghosal, "Wireless sensor network survey," *Computer Networks, International Journal of Computer and Telecommunications Networking, ISSN: 1389-1286*, pp.2292-2330, 2008

[17] P.Samundiswary, P.Priyadarshini, P.Dananjayan, "Performance Evaluation of Heterogeneous Sensor Networks," *IEEE International Conference on Future Computer and Communication (ICFCC)*, pp.264 - 267, 2009

[18] Thomas D.McGee, "Principles and methods of temperature measurement," *Willey-Interscience Publication*, 1988

[19] Robin E. Bentley, "Theory and Practice of Thermoelectric Thermometry," *Springer Handbook of Temperature Measurement Vol.3*, 1998

[20] R.J. Robles, T.Kim, "Applications, Systems and Methods in Smart Home Technology: A Review," *SERSC International Journal of Advanced Science and Technology Vol.15*, pp.37-47, 2010

[21] Z.Hong-bing, Z.Zhe-zhao, "A fitting Method of the Temperature Characteristic Curve of Sensor," *IEEE, Wireless Communications, Networking and Mobile Computing, 5th International Conference on*, pp.1-4, 2009

[22] A. Gunawardana, G.Shani, "A Survey of Accuracy Evaluation Metrics of Recommendation Tasks," *Journal of Machine Learning Research (JMLR) Vol.10*, pp.2935-2962, 2009

[23] J. Wan, H. Yan, H. Suo and F. Li, "Advances in cyber-physical systems research,? *KSII Transactions on Internet and Information Systems, vol. 5, no. 11*, pp.1891-1908, 2011

[24] F. Xia, X. Kong and Z. Xu, "Cyber-physical control over wireless sensor and actuator networks with packet loss," *Wireless Networking Based Control, Springer*, pp. 85-102, 2011

[25] K. Wan, K.L. Man and D. Hughes, "Specification, analyzing challenges and approaches for cyber-physical systems (CPS),? *Engineering Letters, vol. 18, no. 3*, pp. 308-305, 2010

[26] E.A. Lee, "CPS foundations," *ACM/IEEE Design Automation Conference (DAC), Anaheim, USA*, pp.737-742, 2010

[27] F.Xia, T.Mukherjee, Y.Zhang and Y.Song, "Sensor Networks for High-Confidence Cyber-Physical Systems," *Int. J. of Distributed Sensor Networks Vol.201*, Article ID 245734, 2011

[28] R.J. Hyndman, "Yule-Walker estimators for continuous-time autoregressive models?, *J. of Time Series Analysis, vol.14*, pp.281?296, 1993

[29] A. Azadeh, S.M. Asadzadeh, R.J. Marandi, S.N. Shirkouhi, G.B. Khoshjhou and S. Talebi, "Optimal estimation of missing values in randomized complete block design by genetic algorithm," *Knowledge-Based Systems, Elsevier, vol.37*, pp.3747, 2013

[30] C.J. Wilmott, K. Matsuura, "Advantages of the mean absolute error (MAE) over root means square (RMSE) in assessing average model performance," *Climate Research, vol.30*, pp.79-82, 2005