Fire Detection Robot using Type-2 Fuzzy Logic Sensor Fusion

Xuqing Le

A thesis presented for the degree of Master of
Applied Science in Engineering

uOttawa

L'Université canadienne
Canada's university

Department of Graduate Studies in Mechanical Engineering,
University of Ottawa, Canada

Abstract

In this research work, an approach for fire detection and estimation robots is presented. The approach is based on type-2 fuzzy logic system that utilizes measured temperature and light intensity to detect fires of various intensities at different distances. Type-2 fuzzy logic system (T2 FLS) is known for not needing exact mathematic model and for its capability to handle more complicated uncertain situations compared with Type-1 fuzzy logic system (T1 FLS). Due to lack of expertise for new facilities, a new approach for training experts' expertise and setting up T2 FLS parameters from pure data is discussed in this thesis. Performance of both T1 FLS and T2 FLS regarding to same fire detection scenario are investigated and compared in this thesis. Simulation works have been done for fire detection robot of both free space scenario and new facility scenario to illustrate the operation and performance of proposed type-2 fuzzy logic system. Experiments are also performed using LEGO MINDSTROMS NXT robot to test the reliability and feasibility of the algorithm in physical environment with simple and complex situation.

Acknowledgement

I would like to express my sincere appreciations to my supervisor Dr. Dan Necsulescu for his patience, support, inspiration, direction and suggestions. I overcome countless difficulties in the research with his help, and finally, find the right direction of my thesis. I also want to thank Dr. Jurek Sasiadek and Dr. Davide Spinello for their time and helpful suggestion. I would also like to thank my colleagues Ying Dong, Jin Bai, and Guangqi Nie in my research lab for their advices and encouragement. Finally, I would like to thank my parents for their care, patient and love.

Contents

List of Figures

List of Tables

1. Introduction

1.1 Background of Mobile Robot

Robots represent devices which are able to perform some activities similar to a human. Mobile robots can be considered as a mobile platform with large mobility within its work environment, e.g. air, land and underwater.

A mobile robot consists of as an integration of various physical hardware and computational software components. With respect to hardware components, a mobile robot is a collection of subsystems with different functionalities including locomotion, sensing, reasoning and communication. Locomotion is the basic function of mobile robot which determines how the robot moves through its environment. Sensing is data collection process which provides the information about how the robot measures properties of itself and its environment. Reasoning is an intelligent process that shows how the robot maps the measurements into actions, whereas the communication is how the robot communicates with an outside operator as well as a computer or another robot.

There has been tremendous amount of studies focusing on mobile robot recently. With respect to industrial applications, mobile robots started to replace humans to do dull, dangerous and poorly paid works in natural resources sectors such as mining, oil and gas and agriculture.

Mobile robots can be classified into four different types based on the environment they are working in. The lists are shown in below:

- Unmanned Ground Vehicles (UGVs). Working on land, they are wheeled, tracked or legged robots with two or more legs. Land and home robots are usually classified into this part.
- Unmanned Aerial Vehicles (UAVs). Working in the air, aerial robots are usually referred to this type.
- Autonomous Underwater Vehicles (AUVs) are robots work underwater.
- Polar Robots. This kind of robot is designed to navigate in icy filled environments.

Based on the device that robots used to move, UGV mobile robots can be further classified into three types:

- Legged Robot: Human-like legs (humanoid robot) or animal-like legs (resembling animals or insets).
- Wheeled Robot.
- Continuous Track Robot.

Mobile robots with legs do have high degrees of freedom, but they are often mechanically complicated and are expensive to design and manufacture. On the other hand robot with wheels and tracks are easier and cheaper to manufacture, and they can move with higher speed. The wheeled mobile robot is also the main object of study in our research.

1.2 Motivation

Fire is very useful for human being as long as it is under our control, it serves lots of valuable purposes for us, but once it goes out of our control, fire can lead to gigantic destruction and result in terrible disaster. Based on the different environment fire happened, it can be classified into different types. Compare with all different types of fire, structure fires are the most numerous situation lead to loss of life and property. According to NFPA (National Fire Protection Agency, USA) 2013 statistics

- 1,240,000 fires were reported in the U.S in 2013.
- 3240 civilian fire deaths in 2013
- 15925 civilian fire injuries in 2013
- One civilian fire death occurred every 2 hours and 42 minutes
- 16,500 civilian fire injuries
- One civilian injury occurred every 32 minutes
- $12.4billion in property damage
- A fire department responded to a fire every 23 seconds
- 480,500 of these fires were structure fires.

To avoid fire disaster, it is extremely important to detect fires earlier in their development stage when there is still enough time for the safe evacuation for residents or industrial workers. Early detection also plays a significant role in protecting potential property loss for a company. Property loss could be reduced for the operation minimized through early detection because emergency measures are executed while the fire is still controllable.

Fire alarm systems are the most wildly used techniques to protect human and properties from a potential fire disaster. These systems have their numerous advantages, but they still have limitations. One important concern is that for the safety consideration, fire alarm systems are often quite sensitive. It will result in large amount of false alarms due to lack of further information about the fire. Huge amount of unnecessary human resource and energy are wasted because of fire false alarms every year. Another limitation of fire alarm systems is their performance. A YES/NO result is often not good enough for fire rescue. More information about the occurred fire is desired to perform a better evacuation or rescue. If further information such as fire intensity can be obtained, fire rescue can be better organized and performed.

To perform the tasks described above, there is a need of using an autonomous robot which can detect and approach a potential fire source in its early stage and give the estimation information of the fire intensity. With more information and less false alarms detected by robot regarding to the fire, a better evacuation or rescue can be organized and performed to minimize the loss from the fire disaster. This is extremely important and helpful in the situation when human observations are not able to perform 24/7 at the place that potential dangerous may lead to a fire disaster.

1.3 Research Objective and Method

Fires lead to various physical phenomena and influence numerous physical states such as temperature, humidity and light intensity. Therefore, a sensor fusion algorithm is needed for fire detection to obtain and process data from multiple sensors. Fuzzy logic system is chosen because it gives the better results when exact mathematical model cannot be built. In the case of fire, because fire influences various physical conditions simultaneously with no exact mathematical pattern, fuzzy logic is a suitable choice due to its capacity to handle uncertainty. However there are some limitations in regular type-1 fuzzy logic. For the case of multiple experts work on the same problem, different membership functions have to be developed by different experts for same fuzzy set. Type-1 fuzzy logic provides no solution to handle this issue.

In order to overcome above limitation, a new algorithm that carries the advantage of fuzzy logic system and has more expandability is needed, in this case type-2 fuzzy logic. We used type-2 fuzzy logic to build our proposed fire detection system. By performing a lot of simulations and experiments to check the feasibility of our proposed algorithm, we verified that type-2 fuzzy logic can be used reliably and efficiently for fire detection.

Another limitation with respect to both type-1 and type-2 fuzzy logic is that fuzzy logic algorithm usually requests prior experts’ experience to build a reliable system. Experts can provide suitable expertise if available. However for the situation of a new facility, lack of expertise will limit the usage of fuzzy logic system. To overcome this limitation, instead of formulating the type-2 fuzzy logic system directly based on available prior knowledge of experts, a new approach was proposed for training experts’ expertise and setting up type-2 fuzzy logic system with its related parameters from simulation data.

1.4 Thesis Outline

This thesis consists of eight chapters that refer to the designing and executing of the application of type-2 fuzzy logic system for fire detection robot.

Chapter 2 provides a literature review on fire detection, sensor fusion and type-2 fuzzy logic related topics

Chapter 3 presents the creation of type-2 fuzzy logic system for fire detection robot, and a new method to build fuzzy logic system without prior expertise is proposed.

Chapter 4 illustrates the hardware architecture and the mechanical design of the LEGO MINDSTORMS NXT robot which is used for our experiments.

Chapter 5 introduces the software architecture which will be used in our simulation and experiment.

Chapter6 lists the fire simulation process in both MATLAB and PyroSim. The performance of proposed sensor fusion algorithm is tested and applied for both free space situation and new floor plans scenarios.

Chapter 7 presents all experimental results. The experiments are designed to reproduce the situations in the simulation works. Results prove the proposed sensor fusion approach can be utilized in practical situations.

Finally, Chapter 8 provides conclusions of this thesis, our research contributions, and discusses the topics of future work.

Chapter 2

Literature Review

2.1 Fire Detection Method

In [1], a wildfire detection method using social media is introduced by Viktor Slavkovikj, et. The author indicated that with the increasing number of social networks and services, there has been countless information sharing on the internet. In this case the social media played a critical role in disaster detection in the past decade. A review of now days existing systems and approaches which using the social media as a human-centric sensor for large scale disaster detection has been presented at the beginning of the paper. Then the author proposed the general architecture of his Wildfire Social Sensor (WSS) Platform. The WSS Platform is separated into several stages by its different functionalities. Collection is an initial status to obtain information from the social media. Aggregation and filtering stage works on grouping different kinds of data gathered from multiple social media and filtering redundant information from the gathered dataset. Verification stage is used for estimating the reliability of the gathered data, whereas the Analysis stage is used for automatic processing of the gathered data and filtering high level information. Last stage is the central coordination stage which provides secure and timely exchange information between the groups, individuals and organizations involved in the incident. Results show that social media could be used effectively to provide human-centric sensor network for early detection of large scale disaster such as fire.

In [2], fire detection system using fuzzy logic has been developed by Aiswarya Muralidharan and Fiji Joseph. Instead of depending on a single sensor output for detecting fire, multiple sensors are utilized in their fuzzy logic system to obtain more accurate and no error decision. Even though efficient, the existing problems about complex mathematical or control systems have been discussed in the paper. The multi sensor based fire detection systems (MSbFD) introduced by the authors basically takes inputs from three different sensors (temperature, light and smoke) and evaluates and process the sensor signals by using fuzzy logic. There are five final input variables implemented, namely absolute Temperature (Ta), differential Temperature (Td), absolute smoke density (Sa), fluctuation in smoke density (SF) and light intensity variation (Lv). Result shows that the fuzzy logic application is able to provide precise results about fire under different conditions. The implementation of fuzzy logic also made the system much simpler and closer to human way of thinking.

In paper "Autonomous Fire Fighting Mobile platform", Teh Nam Khoon, et al [3] have presented and designed a novel autonomous robot for firefighting. This robot which is called by them, Autonomous Fire Fighting Mobile Platform (AFFPM, has both flame sensor and obstacle avoidance systems installed. The AFFPM can move follows a known path and utilizes a guide rail or markers such as black painted line or a tape to navigate through the environment until a high possibility of fire is detected. When a fire is detected, the robot will leave its prior track and approach the fire. Finally it will stop at 30 cm away from the flame. It then would engage the fire extinguisher which is implemented on the robot. Once it has left the 9 prior navigation routes, the obstacle avoidance system will be activated and the system will be able to guide the robot closer to the fire source. After it extinguished the fire completely, robot would then return to its guiding track to carry on with its further investigation of any other potential fire.

In [4], a new fire detection method is presented, which is based on utilizing a stereo camera system to compute the distance from the fire region to the camera, and to reconstruct the 3D surface of the fire front. Selected fire source is identified by utilizing background difference model and generic color models for further fire detection. Because fire frames change constantly, it’s hard to illustrate the frames by exact mathematic model. Fuzzy logic is then chosen to describe the fire frames. Gaussian membership functions (GMFs) for shape, size, and motion variation of the fire are generated. These three GMFs are then implemented to fuzzy logic system for real-time fire detection. After division of fire sources from left and right images, feature points are obtained by utilizing the matching algorithm. Then the distance estimation and 3D surface reconstruction are calculated by the difference detected from two images. The experimental results showed that the proposed algorithm can produce a more robust fire detection result than the regular methods discussed in this paper. In addition, the author indicated that the result is more reliable and accurate within a 5-m distance from the stereo camera so that the proposed algorithm is more suitable for an indoor environment or for a fire which is in a short distance away from an outdoor situation.

Recent developments about Video based Fire Detection (VFD) are discussed in [5]. The author illustrated that video cameras and computer vision system are widely used these days in many different security applications including fire detection. VFD can help reduce the detection time needed comparing to the currently available algorithms and sensors in both indoors and outdoors environment because vision systems and cameras do not have transport delay that traditional sensors suffer from. Multiple types of video fire detection algorithms are introduced then in this paper. For example, color detection approaches in VFD uses RGB color space to distinguish a fire, whereas moving object detection approaches detect a fire by treating flames and smoke as moving object. Another method introduced by the author is called dynamic texture and pattern analysis. A dynamic texture or pattern in video, such as smoke, flames, water and leaves in the wind can be simply defined as a texture with motion. The author also introduced later the infrared (IR) spectral range fire detection method when dealing with the situation that there is no or very little visibility or the color of the fire to be detected is very similar to the background.

In [6], an automatic fire detection system using adaptive fusion algorithm for firefighting robot was discussed by Kuo L.Su. The author developed six different systems for his firefighting robot. It included obstacle avoidance and driver system, software development system, fire detection system, remote supervise system and other system. Three flame sensors were implemented on the mobile robot. The weight value of each sensor is time-varying in different condition in this research work. The author used his adaptive decision method to obtain the optimal values for the weight of each sensor. To achieve the computation complication, the firefighting robot transmits signals measured from these flame sensors to the supervised computer using wireless RS232 interface. The output decision will then transmit back to the mobile robot from the computer.

2.2 Sensor Fusion Method

In the book [7], the author discussed the knowledge with respect to target and background signature-generation phenomena, sensor design, signal processing algorithms, available communications types and bandwidths, and end use of the fusion products. From chapter 3 to chapter 9, different sensor fusion data fusion methods were introduced respectively such as the classical inference, Bayesian inference, Dempster-Shafer evidential theory, artificial neural networks, voting algorithm, fuzzy logic and fuzzy neural networks.

In [8], fire detection system based on multi-sensor data fusion is proposed. The data are gathered by using a wireless network of environmental sensors scattered at the supervising URI area and a vision sensor that monitors the same geographical area. Each sensor reading is sent to an integrated control unit. A probability is established for the lower lever fusion for each individual sensor node. Next step the results data will be sent and fused with the second level, the vision sensors that monitor the same geographical area. Different data were fused both in the first level and the second level. The first step fusion is mandatory to reduce the latency of the fire detection process, when the second step uses complementary type of sensors to deal with possible conflicts and ignorance induced by outliers or malfunctioning sensor nodes. The authors believe that fusing fire possibilities induced by vision system together with the corresponding probabilities obtained from in-field environmental sensors is a novel and very promising approach for fire detection.

2.3 Type-2 Fuzzy Logic Algorithm

In [13], a fire detection algorithm using type-2 fuzzy logic system is presented by A.K. Singh and Harshit Singh. The author illustrated that fuzzy logic works better for fire detection because there is an uncertainty about temperature, humidity and light intensity which is involved to cause a fire. A brief introduction about type-1 fuzzy logic system and type-2 fuzzy logic system is presented first. The authors took four input factors in their application which are temperature, light intensity, humidity and carbon mono-oxide density. Each of its input parameter is separated into three membership functions which are low, medium and high. The output is probability of fire which has five different membership functions as very low, low, medium, high and very high. The results show that the proposed fuzzy type-2 logic approach can deal with the uncertainty presented in the data effectively and can also give the best results with less false alarm rate compared to a traditional algorithm. The author believes that the results generated from type-2 fuzzy logic system are more accurate than the results obtained from type-1 fuzzy logic system.

A review about type-2 fuzzy logic approach in clustering, pattern recognition and classification has been discussed in [21] by Patricia Melin and Oscar Castillo. The author stated that type-2 fuzzy logic has obtained the popularity in a wide range of applications due to its ability to handle higher levels of uncertainty which has helped to improving the results gathered from type-1 fuzzy logic. Type-2 fuzzy logic used in clustering and classification has been discussed first. Such application has been developed for vehicle classification, micro-calcification detection and the battlefield ground vehicles classification. The author then introduced type-2 fuzzy logic which is used in pattern recognition application. Such approach for computer aided detection, edge detection, fingerprint recognition, pattern recognition, medical diagnosis and face recognition are discussed respectively.

In [22], the classification for medical data using wavelets and interval type-2 fuzzy logic system was presented by Thanh Nguyen, et. The feature extraction methods Wavelet transformation is used in this research to reduce the data dimension in order to prepare the inputs for interval type-2 fuzzy logic. The fuzzy logic would be more powerful in approximation and classification with the help of Wavelet. The FCM clustering method is implemented by the author to initialize the GA population for type-2 fuzzy logic system training. GA then work as a robust evolutionary algorithm for type-2 fuzzy logic parameter tuning. The results shows that the Wavelet transformation in this research shows more efficient ability about extracting data compared to the PCA method. With utilizing the wavelet features, the performance of the entire classification system enhances.

In [23], a type-2 fuzzy logic system based energy management system for hybrid electrical vehicles has been introduced by Javier Solano Martinez, Robert I.john, et. The author discussed and developed a fuzzy logic controller which is able to manage the energy in a hybrid electrical vehicle implemented with three different energy sources. The energy management system aims to the design for different customer needed such as energy consumption minimization or comfort driving. The membership functions and rule bases for fuzzy logic systems in this research are generated by either data from the system or the human experience. A survey was also performed and 10 experts were invited to help build the fuzzy sets and rules of the fuzzy logic controller. The proposed energy management based on fuzzy logic controller was then tested by computer simulation work. Result shows that the type-2 fuzzy logic energy management system can achieve the specified goals. Future work for this research will be the real-time implementation and testing of the type-2 fuzzy logic energy management strategy in the electrical vehicle. A general type-2 sets may also be investigated instead of interval type-2 sets.

An overview of the applications of interval type-2 fuzzy logic in intelligent control field has been discussed in [24] by Tijuana. The major focus of this paper is on bio-inspired optimization methods. Three classic paradigms, genetic algorithms, particle swarm optimization and ant colony optimization that help the designing of optimal type-2 fuzzy logic controllers have been discussed. In the first section, topic as achieving the output regulation of a servomechanism with backlash, evolving type-2 fuzzy logic controllers for real world autonomous robots and type-2 fuzzy logic system cascaded with neural network were introduced. The next section about this paper illustrated most successful application about improving fuzzy logic by using particle swarm optimization. This algorithm severed for the type-2 fuzzy logic system which is used in uncertainty in control and presence of noise topic of research. The Ant Colony Optimization evolving algorithms has been presented at last. These algorithms are used mainly for type-2 fuzzy logic applications which are designed for uncertainty in mobile robots and navigation, modeling uncertainty in control and test different controllers.

Chapter 3

Fuzzy Logic Sensor Fusion

Fuzzy logic gives a possibility for representing analog processes in a digital framework. Processes that are implemented through fuzzy logic are usually not easily separated into discrete segments and it may also be difficult to model with conventional mathematical paradigms which require hard boundaries or decisions. Therefore fuzzy logic is valuable where the boundaries between sets of values are not sharply defined or there is partial occurrence of an event. [15]

In this chapter, regular fuzzy logic will be first introduced in section 3.1. The introduction of a higher uncertainty fuzzy logic, type-2 fuzzy logic, will be presented in section 3.2. Section 3.3 will focus on our proposed fire estimation approaches based on type-2 fuzzy logic system.

3.1 Fuzzy Logic

3.1.1 Basic Elements of Fuzzy System

There are three basic elements in a fuzzy system. They are fuzzy sets, membership functions, and production rules. Fuzzy sets are the fundamental elements of fuzzy logic system, whereas the membership functions are used to graphically represent fuzzy sets. The production rules work as human’s brain which helps the fuzzy logic system to make a decision.

3.1.1.1 Classical Sets and Fuzzy sets

- **Classical Sets**

The concept of a set is fundamental in mathematics and could be illustrated as a collection of objects possibly linked through some properties. Very clear boundaries are signed to a classical set, for example $\mathrm{x} \in \mathrm{A}$ or $\mathrm{x} \notin \mathrm{A}$ which exclude any different possibility [14].

Let X be a set and let A be a subset of X ($A \subseteq X$). Then we can define the function

$$X_A(x) = \begin{cases} 1 & if \quad x \in \mathrm{A} \\ 0 & if \quad x \notin \mathrm{A} \end{cases} \tag{3.1}$$

Which is called the characteristic function of the set A in X.

Classical sets and relative operations can be achieved by their characteristic functions.

- **Fuzzy sets**

Fuzzy sets were introduced by L. Zadeh in [13]. The definition of a fuzzy set given by L. Zadeh is as follows: A fuzzy set is a class with a continuum of membership grades. So a fuzzy set A in a referential (universe of discourse) X is characterized by a membership function A which associates with each element. [14]

A fuzzy set *A* and its fuzzy subset X is defined as a mapping:

$$\mathrm{A}: \mathrm{X} \rightarrow [0, 1] \tag{3.2}$$

where A(x) is the membership degree of x to the fuzzy set A. We denote by *F*(X) the collection of all fuzzy subsets of X.

Every classical set is also considered as a fuzzy set. Fuzzy sets are generalizations of the classical sets represented by their characteristic functions $x_A : X \to \{0,1\}$. In the case A(x) = 1 expresses full membership of x in A, whereas A(x) = 0 means there is no membership of x in A, but in contrary to the classical case other membership degrees are allowed.

3.1.1.2 Membership Function

The membership function of a fuzzy set is a generalization of the indicator function in classical sets. In fuzzy logic, the membership function represents the degree of truth. Membership functions allow us to illustrate a fuzzy set using graph. The x-axis in the graph stands for the universe of discourse, whereas the y-axis stands for the membership degrees in [0 1] interval. A membership function for a fuzzy set A on the universe of discourse X is defined as μA: X → [0, 1], where each element of X is mapped to a value between 0 and 1. This value is known as degree of membership or quantifies the grade of membership of the element in X to the fuzzy set A.

Usually simple functions are chosen to build membership functions for fuzzy sets since we are defining fuzzy concepts. Using more complex functions does not add too many precision. Three commonly used membership functions are demonstrated as: Figure 3.1 below defined a triangular shape membership function:

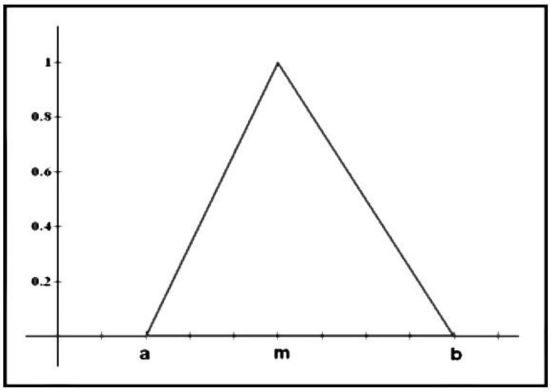


Figure 3.1 Triangular membership functions

which satisfied:

$$\mu_A(x) = \begin{cases} 0, & x \le a \\ \dfrac{x-a}{m-a}, & a < x \le m \\ \dfrac{b-x}{b-m}, & m < x < b \\ 0, & x \ge b \end{cases} \tag{3.3}$$

Where the triangular membership function defined by a lower limit a, an upper limit b, and a value m, where a < m < b

Figure 3.2 below defined a trapezoidal shape membership function:

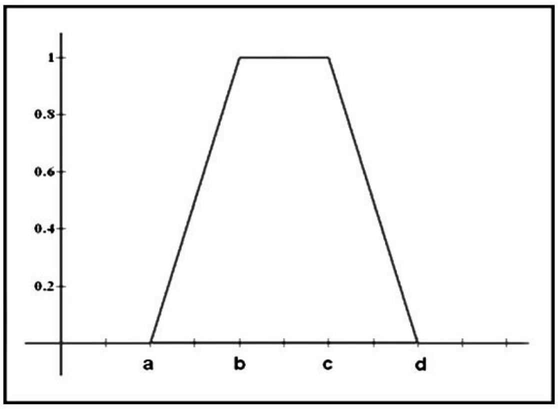


Figure 3.2 Triangular membership functions

which satisfied:

$$\mu_A(x) = \begin{cases} 0, & (x < a)\ or\ (x > d) \\ \frac{x-a}{b-a}, & a \le x \le b \\ 1, & b < x < c \\ \frac{d-x}{d-c}, & c \le x \le d \end{cases} \tag{3.4}$$

Where the trapezoidal membership function defined by a lower limit a, an upper limit d, a lower support limit b, and an upper support limit c, where a < b < c < d

Figure 3.3 below defined a Gaussian shape membership function:

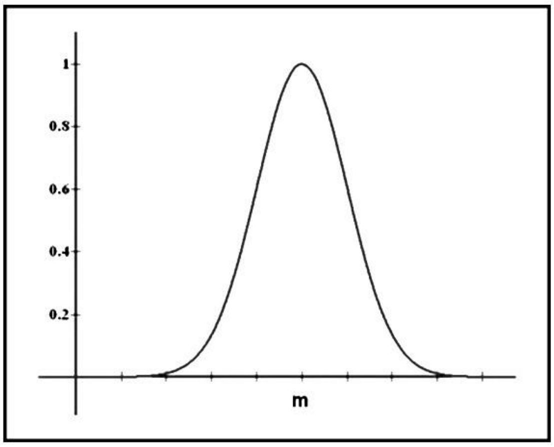


Figure 3.3 Triangular membership functions

which satisfied:

$$\mu_A(x) = e^{-\frac{(x-m)^2}{2k^2}} \tag{3.5}$$

Where the Gaussian membership function defined by a central value m and a standard deviation k > 0, the smaller k is, the narrower the “bell” is.

3.1.1.3 Production rules

Fuzzy logic system use production rules represent “IF-THEN” logic statements in human knowledge. IF-THEN statements are known as integral part of an expert system in artificial intelligence field. However, expert systems usually rely on classical logic and probability to develop the inferences used in the production rules. Fuzzy sets incorporate vagueness into the production rules since they represent less precise linguistic terms, for example, very hot, not very close, and small. The production rules operate in parallel and influence the output of the fuzzy logic system to changing degrees. These logical processing using fuzzy sets are known as fuzzy logic. [15]

3.1.2 Fuzzy Logic Processing

Fuzzy logic processing is illustrated in Figure 3.4(a) and Figure 3.4(b). The processing sequence can be divided into three different sections, the fuzzification, inference and defuzzification. Fuzzification is the processing that crisp inputs are transformed into fuzzy sets, which then can be used by fuzzy logic system. Inference processing begins with the development of the production rules in the form of IF-THEN statements. The antecedent or condition block of the rule starts with IF statements and the consequent or conclusion block starts with THEN statements. The value assigned to the consequent block is equal to the logical product of the activation values of the antecedent membership functions that characterize the boundaries of the fuzzy sets. The activation value is equal to the value of the membership function at which it is intersected by the input variable at the time instant being evaluated. A defuzzification operation is performed to convert the fuzzy values, represented by the logical products and consequent membership functions, into a fixed and crisp output. Defuzzification may be implemented in several ways. Most applications execute a center-of-mass or fuzzy centroid computation on the consequent fuzzy set. This is equivalent to finding the mode of the distribution if it is symmetric and unimodal [15]. The integrated fuzzy logic processing is shown in Figure 3.4(a) and Figure 3.4(b)

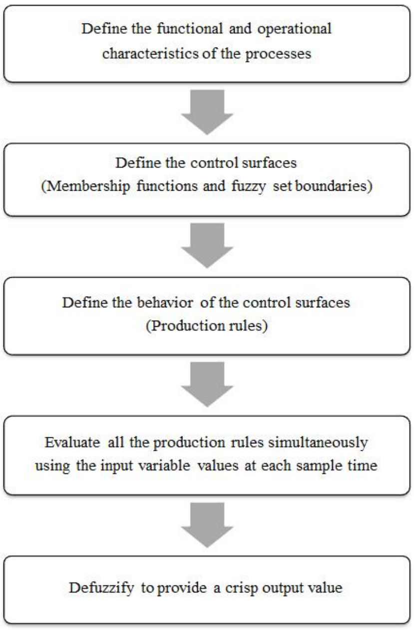


Figure 3.4(a) Fuzzy logic processing

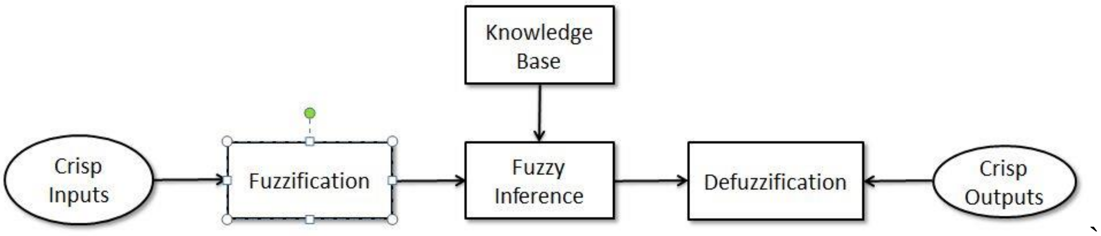


Figure 3.4(b) Fuzzy logic processing

3.2 Type-2 Fuzzy Logic System

A fuzzy logic system that is described completely in terms of type-1 fuzzy sets, the regular fuzzy sets, is called type-1 fuzzy logic system, whereas a fuzzy logic system that is expressed using at least one type-2 fuzzy set is called a type2 fuzzy logic system. In spite of their name, type-1 fuzzy logic systems are usually unable to directly handle complicated uncertainties because the type1 fuzzy sets they used are certain. The membership function for a type-1 fuzzy set is fixed and there is less flexibility for it. On the other hand, type2 fuzzy logic systems are very useful in circumstances where it is difficult to determine an exact membership function for a fuzzy set. Hence, they can be used to handle more complicated uncertainties [16].

3.2.1 Type-2 Fuzzy Sets

Type-2 Fuzzy sets are first introduced by Lotfi A. Zadeh in [18], when he proposed more complicated fuzzy sets. A type-2 fuzzy set give us the opportunity to incorporate uncertainty about the membership function into fuzzy set theory.

A type-1 fuzzy set has a grade of membership that is crisp, where as a type-2 fuzzy set has grades of membership that are fuzzy, so it is called a “fuzzy-fuzzy set.” This kind of set is useful in situations where it is difficult to determine the exact membership function for a fuzzy set. If all uncertainty disappears, then a type2 fuzzy set reduces to a type1 fuzzy set [17].

3.2.2 Membership Function for Type-2 Fuzzy Sets

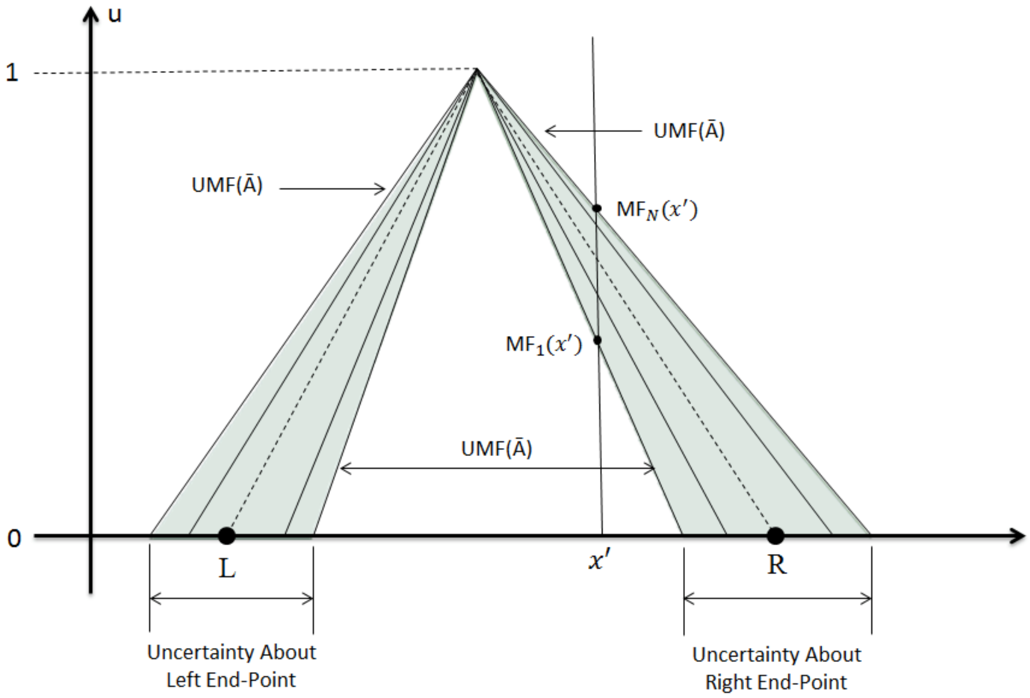


Figure 3.5 Membership function for type-2 fuzzy set.

As can be seen in Figure 3.5, if the uncertainties about the left-end and right-end points disappear, then only the dashed triangle survives. This is similar to what happens to probability, when randomness degenerates to determinism. In brief, a type1 fuzzy set is embedded in a type2 fuzzy set.

3.2.3 Type-2 Fuzzy Logic Processing

A rule-based type-2 fuzzy logic system contains four components. They are rules, fuzzifier, inference engine and output processor. As shown in Figure 3.6, once the rules have been established, a type-2 fuzzy logic system can be viewed as a mapping from inputs to outputs, and this mapping can be expressed quantitatively as y=f(x).

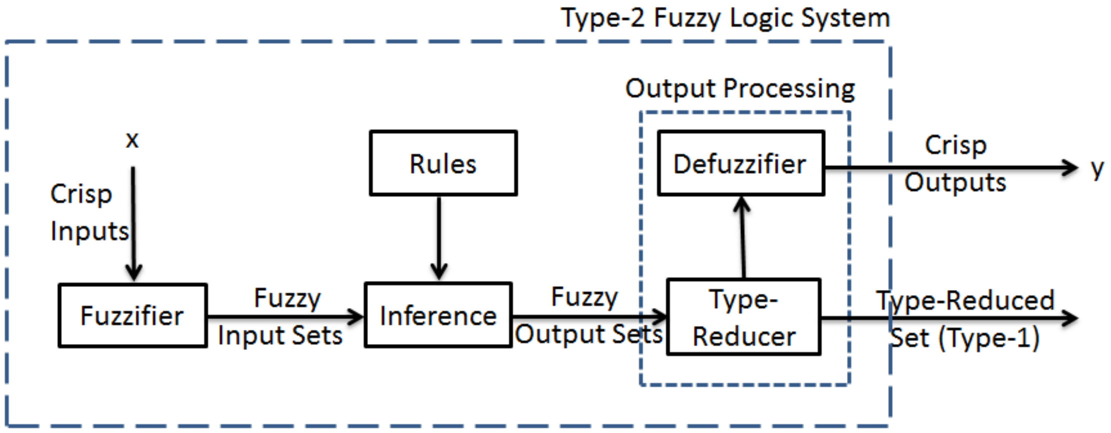


Figure 3.6 Type-2 fuzzy logic processing

As shows in Figure 3.6, the output processing for type-2 fuzzy logic system consists of two parts. One is the type-reduction (TR) and another is Defuzzification. The defuzzification stage is same as type-1 fuzzy logic system. On the other hand, the type-reduction methods are considered as type-2 extensions of type-1 defuzzification methods, each of which relies on some kind of centroid calculation. Regardless of the kind of type-reduction, they all need calculation the centroid of an interval type-2 fuzzy set (IT2 FS). These two iterative algorithms for computing the centroid are known as the Karnik-Mendel (KM) algorithms. This algorithm will be discussed in next section. [19]

3.2.4 Karnik-Mendel Algorithm

KM algorithms are two iterative algorithms for calculating the centroid of an interval type-2 fuzzy set. KM algorithm consists of two parts, one is used for computing y_l and the other is used for computing y_r.

Define

$$f_l(k) = \frac{\sum_{n=1}^{k} \underline{x}_n \overline{w}_n + \sum_{n=k+1}^{N} \underline{x}_n \underline{w}_n}{\sum_{n=1}^{k} \overline{w}_n + \sum_{n=k+1}^{N} \underline{w}_n} \tag{3.6}$$

$$f_r(k) = \frac{\sum_{n=1}^{k} \overline{x}_n \underline{w}_n + \sum_{n=k+1}^{N} \overline{x}_n \overline{w}_n}{\sum_{n=1}^{k} \underline{w}_n + \sum_{n=k+1}^{N} \overline{w}_n} \tag{3.7}$$

Where $\{\underline{x}_n\}$ and $\{\overline{x}_n\}$ have been sorted in ascending order, respectively. Furthermore, in this thesis it is assumed that $\{\underline{x}_n\}$ $\{\overline{x}_n\}$ has no duplicate elements, which can be easily achieved by combining the weights for duplicate elements. Then, y_l and y_r can also be re-expressed as:

$$y_l = \min_{k=1,\ldots,N-1} f_l(k) \equiv f_l(l) \tag{3.8}$$

$$= \frac{\sum_{n=1}^{l} \underline{x}_n \overline{w}_n + \sum_{n=l+1}^{N} \underline{x}_n \underline{w}_n}{\sum_{n=1}^{l} \overline{w}_n + \sum_{n=l+1}^{N} \underline{w}_n} \tag{3.9}$$

$$y_r = \max_{k=1,\ldots,N-1} f_r(k) \equiv f_r(r) \tag{3.10}$$

$$= \frac{\sum_{n=1}^{r} \overline{x}_n \underline{w}_n + \sum_{n=r+1}^{N} \overline{x}_n \overline{w}_n}{\sum_{n=1}^{r} \underline{w}_n + \sum_{n=r+1}^{N} \overline{w}_n} \tag{3.11}$$

where l and r is switch points satisfying

$$\underline{x}_l \le y_l < \underline{x}_{l+1} \tag{3.12}$$

$$\overline{x}_r < y_r \le \overline{x}_{r+1} \tag{3.13}$$

KMA for computing y_l:

1) Sort $\underline{x}_n - n = 1, 2, ..., N$ in increasing order and call the sorted $\underline{x}_n$ by the same name, but now $\underline{x}_1 < \underline{x}_2 < \cdots < \underline{x}_N$. Match the weights W_n with their respective $\underline{x}_n$ and renumber them so that their index corresponds to the renumbered $\underline{x}_n$.
2) Initialize w_n by setting

$$w_n = \frac{\underline{w}_n + \overline{w}_n}{2}, n = 1, 2, ..., N \tag{3.14}$$

and then compute

$$y = \frac{\sum_{n=1}^{N} \underline{x}_n w_n}{\sum_{n=1}^{N} w_n} \tag{3.15}$$

3) Find switch point $l - 1 \le l \le N$ such that

$$\underline{x}_l < y \le \underline{x}_{l+1} \tag{3.16}$$

4) Set

$$w_n = \begin{cases} \overline{w}_n, \mathrm{n} \le l \\ \underline{w}_n, n > l \end{cases} \tag{3.17}$$

and compute

$$y' = \frac{\sum_{n=1}^{N} \underline{x}_n w_n}{\sum_{n=1}^{N} w_n} \tag{3.18}$$

If $y' = y$, stop and set $y_l = y$ otherwise, set $y = y'$ and go to Step 3

KMA for Computing y_r**:**

1) Sort $\overline{x}_n - n = 1, 2, ..., N$ in increasing order and call the sorted $\overline{x}_n$ by the same name, but now $\overline{x}_1 < \overline{x}_2 < \cdots < \overline{x}_N$. Match the weights W_n with their respective $\overline{x}_n$ and renumber them so that their index corresponds to the renumbered $\overline{x}_n$.
2) Initialize w_n by setting

$$w_n = \frac{\underline{w}_n + \overline{w}_n}{2}, n = 1, 2, ..., N \tag{3.19}$$

and then compute

$$y = \frac{\sum_{n=1}^{N} \overline{x}_n w_n}{\sum_{n=1}^{N} w_n} \tag{3.20}$$

3) Find switch point $r - 1 \le r \le N$ such that

$$\overline{x}_r < y \le \overline{x}_{r+1} \tag{3.21}$$

4) Set

$$w_n = \begin{cases} \underline{w}_n, \mathrm{n} \le r \\ \overline{w}_n, n > r \end{cases} \tag{3.22}$$

and compute

$$y' = \frac{\sum_{n=1}^{N} \overline{x}_n w_n}{\sum_{n=1}^{N} w_n} \tag{3.23}$$

If $y' = y$, stop and set $y_r = y$ otherwise, set $y = y'$ and go to Step3.

3.2.5 Issues Regarding Subjective-Expert Inputs to Type-2 FS

The application of type-2 fuzzy logic system requires inputs of subjective nature available from practical experience:

-shape of membership function

-number of membership functions

-weight factors etc.

Experts might provide these inputs if practical experience are available. For new facilities this becomes a difficult problem due to lack of expertise. The solution proposed in this thesis is to formulate a direct problem, a model of cause (fire) to effects (temperature and light) for new floor plans and run simulations for expert training. This approach is presented in Ch. 3.3.

3.3 Proposed Type-2 Fuzzy Logic Sensor Fusion Approach for Fire Detection and Estimation

In our proposed approach, type-2 fuzzy logic system is used for fire detection and estimation purposes. The major functionality of our proposed sensor fusion approach is to search and estimate various intensities of fire, such as large fire, medium fire and small fire, at different distances. As shown in Figure 3.7, the process takes measured temperature value and light value as inputs variables and produces the estimated fire intensity and the distance between the fire and robot as outputs. For example, results for our application could be: a large fire at close distance or a medium fire at far distance.

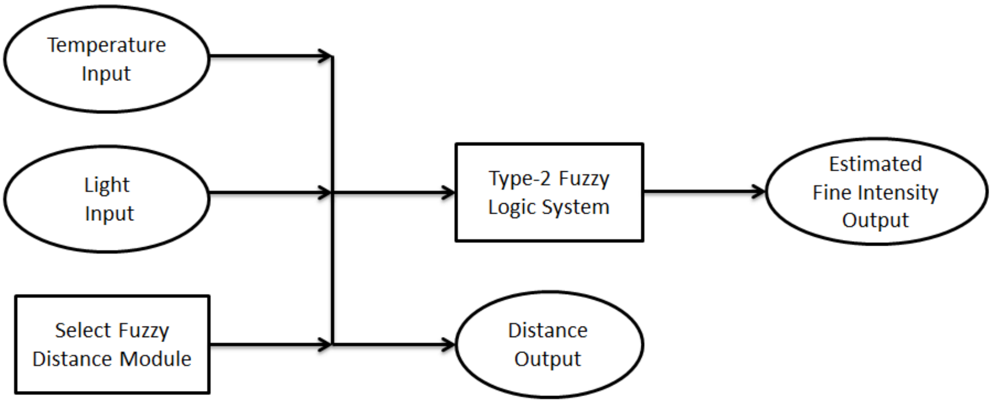


Figure 3.7 Process of proposed type-2 fuzzy logic approach

Instead of building fuzzy logic system directly from experts' experience, we proposed the method to build fuzzy logic system only on data. This method is based on a direct and inverse problem concept [19]. The inverse problem for our application is to get estimated fire intensity using temperature value and light value. To solve this problem, large fire, medium fire and small fire was set up in MATLAB Simulation for our direct problem. Different fires generated respective temperature value and light value at different distances between the measurement point and the center of fire. These data are recorded, gathered, analyzed and used to build type-2 fuzzy logic system. The sensor fusion system is then used to solve the inverse problem in our research which takes the real time temperature measurements and light measurements and gives the estimation of fire intensity.

In our research, we focused on the problems about how to train the experts' expertise requested for building fuzzy logic system from data and how to set up the parameters for type-2 membership functions, such as structuring type-2 membership function based on multiple type-1 membership function. These two issues will be discussed in 3.3.3 and 3.3.4.

Two of my colleagues were invited to develop this type-2 fuzzy system together with me. We were acting as three experts for our application. Instead of using type-1 fuzzy logic, type-2 fuzzy logic system is used for our application because type-1 fuzzy logic has limited performance at handle the situations shown below:

- The words that are used in antecedents and consequents of rules can mean different things to different people.
- Consequents obtained by polling a group of experts will often be different for the same rule, because the experts will not necessarily be in agreement.
- Only noisy training data are available for tuning the parameters of an interval type2 fuzzy logic system.
- Noisy measurements activate the fuzzy logic system.

3.3.1 Fuzzy Logic Module Based on Distance

In order to achieve more precise result, three different sub-modules were developed and embedded in our proposed type-2 fuzzy logic system. As shown in Figure 3.9, different modules are activated based on different distance between fire and robot. The distance information could be obtained from an external range sensor or localization system. Far-distance module will be activated at first when the distance between robot and fire is relatively far. The output from the fuzzy module will give a rough estimation for fire intensity which occurred. Medium-distance module will then be activated when the distance between fire and robot is getting closer. Close-distance module will be activated at last stage and give a precise estimation about the detected fire intensity.

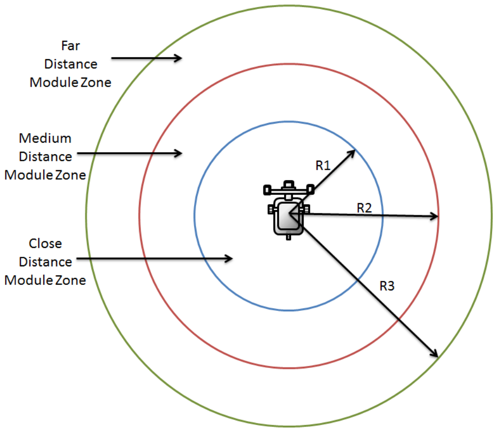


Figure 3.8 Three different distance zones

3.3.2 Data Collection and Process

In order to develop a reliable type2 fuzzy logic system for fire detection and estimation, high quantity temperature data and light data are needed for experts to train their expertise for building fuzzy logic system later on. Reliable data can be generated either through simulations or obtained from actual experiments.

Considering the factor of safety, in our proposed fire intensity estimation system, temperature measurement data and light measurement data were both generated by MATLAB and PyroSim Simulation. These data are related to fire of various intensities which occurred at different distances and are used to help experts train their expertise.

3.3.3 Expertise Training for Type-2 Fuzzy Logic System

The expertise used for developing type-2 fuzzy logic system should be gathered based on expert’s experience or data. An expert can gather his expertise for specific field with accumulated experience or additional new data for specific problem. In our research, we will focus on the problem regarding to training expertise with measurement data.

In our proposed approach, the job for an expert in this specific problem is to estimate the intensity of the fire by analyzing both temperature value and light value. To set up the type-2 fuzzy logic system for fire estimation purpose, experts need describe different intensities of fire using linguistic variables such as a large fire, a medium fire and a small fire. Consequently, these linguistic variables are chosen to describe output fuzzy sets for our application.

Table 1 and Table 2 show the data sample for our proposed fire detection system which was generated by MATLAB Simulation.

Table 1: Temperature-distance relationship data

linguistic dis (m)	dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
center	0	1000	600	200	20
close	3	131.11	86.67	42.22	20
	4	82.5	57.5	32.5	20
	5	60	44	28	20
medium	6	47.78	36.67	25.56	20
	7	40.41	32.24	24.08	20
	8	35.63	29.38	23.13	20
far	9	32.35	27.41	22.47	20
	10	30	26	22	20
	11	28.26	24.96	21.65	20

Table 2: Light-distance relationship data

linguistic dis(m)	dis(m)\light	Large Fir(lm)	Medium Fire(lm)	Small Fire(lm)	No Fire(lm)
center	0	2000	1200	400	30
close	3	252.22	163.33	74.44	30
	4	155	105	55	30
	5	110	78	46	30
medium	6	85.56	63.33	41.11	30
	7	70.82	54.49	38.16	30
	8	61.25	48.75	36.25	30
far	9	54.69	44.81	34.94	30
	10	50	42	34	30
	11	46.53	39.92	33.31	30

The algebra version of Table 1 and Table 2 are shown in Tables below.

Table 3: Temperature-distance relationship data

dis(m)\temp	Large Fire℃	Medium Fire℃	Small Fire℃	No Fire℃
Close	LT1	MT1	ST1	NT1
	LT2	MT2	ST2	NT2
	LT3	MT3	ST3	NT3
Medium	LT4	MT4	ST4	NT4
	LT5	MT5	ST5	NT5
	LT6	MT6	ST6	NT6
Far	LT7	MT7	ST7	NT7
	LT8	MT8	ST8	NT8
	LT9	MT9	ST9	NT9

Where LT1-LT9 represents the temperature value measured for a large fire, MT1, ST1 and NT1 represent the lowest temperature value measured for a medium fire, a small fire and no fire respectively.

Table 4: Light-distance relationship data

dis(m)\light	Large Fir(lm)	Medium Fire(lm)	Small Fire(lm)	No Fire(lm)
Close	LL1 LL2 LL3	ML1 ML2 ML3	SL1 SL2 SL3	NL1 NL2 NL3
Medium	LL4 LL5 LL6	ML4 ML5 ML6	SL4 SL5 SL6	NL4 NL5 NL6
Far	LL7 LL8 LL9	ML7 ML8 ML9	SL7 SL8 SL9	NL7 NL8 NL9

Where LL1-LL9 represents the light value measured for a large fire, ML1, SL1 and NL1 represent the lowest light value measured for a medium fire, a small fire and no fire respectively.

Linguistic variables were used by our experts to evaluate the intensities of both temperature value and light value in order to build the type-2 fuzzy logic system later on. Consequently, these linguistic variables are chosen to describe the input fuzzy sets for our application. Table 5(a) and Table 5(b) show the example about how an expert interpreted the date.

Table 5(a): Example of data analyzing for temperature value by expert

dis(m)\temp	Large Fire℃	Medium Fire℃	Small Fire℃	No Fire℃
Close	high temperature	medium temperature	medium temperature	very low temperature
Medium	medium temperature	medium temperature	low temperature	very low temperature
Far	low temperature	low temperature	very low temperature	very low temperature

Table 5(b): example of data analyzing for light value by expert

dis(m)\light	Large Fir(lm)	Medium Fire(lm)	Small Fire(lm)	No Fire(lm)
Close	high light value	medium light value	medium light value	very low light value
Medium	medium light value	medium light value	low light value	very low light value
Far	low light value	low light value	very low light value	very low light value

With studying and analyzing the data gathered from multiple simulations, our experts can get the expertise which is needed for building our fire detection and estimation fuzzy logic system later on.

3.3.4 Type-2 Fuzzy Logic System Parameters Setup

3.3.4.1 Numbers of Fuzzy sets

The number of fuzzy sets for both inputs and outputs should be chosen as needed. The results can be altered based on how the data was analyzed by experts. It is reasonable to have three input fuzzy sets for temperature value if the expert considered temperature value can be classified as high, medium and low. In another case, the input fuzzy sets for light value should be two if an expert classified the light value as only high and low.

In our proposed type-2 fuzzy logic approach, four different linguistic variables were used by our experts to describe the different temperature intensity and light intensity for fuzzy inputs. High, medium, low and very low are the four chosen linguistic variables to describe four different fuzzy sets in our proposed fuzzy logic system for both input temperature and light. The fire intensity output was described by four other linguistic variables with respect to no fire, small fire, medium fire and large fire.

3.3.4.2 Model of Type1 Membership Function

Membership function is one of the most important parts of fuzzy logic system. It decides how input variables are converting into fuzzy sets. Boundaries and endpoints of each membership function should be chosen carefully based on simulation or experiment data. In our proposed approach, in order to get a reliable fuzzy logic system, all experts developed type-1 membership function for each fuzzy set based on its own membership function model. The process of how to build membership function model is discussed in this section. Since there were three experts, three different type-1 membership functions were built for single fuzzy set which will then form a new type-2 membership function later on. The formation of type-2 membership function will be discussed in next section.

An example of temperature data and how an expert interpreted it are shown in Table 6 and Table 7.

Table 6: Temperature-distance relationship data

dis(m)\temp	Large Fire℃	Medium Fire℃	Small Fire℃
Close	LT1 LT2 LT3	MT1 MT2 MT3	ST1 ST2 ST3

where LT1-LT3 represents the temperature value measured for a large fire, MT1, ST1 and NT1 represent the lowest temperature value measured for a medium fire, a small fire and no fire respectively.

Table 7: Temperature data after expert interpreting

dis(m)\temp	Large Fire℃	Medium Fire℃	Small Fire℃
Close	high temperature	medium temperature	low temperature

Three different model of membership functions with different endpoint and overlaps could be generated from data depended on different input temperature value

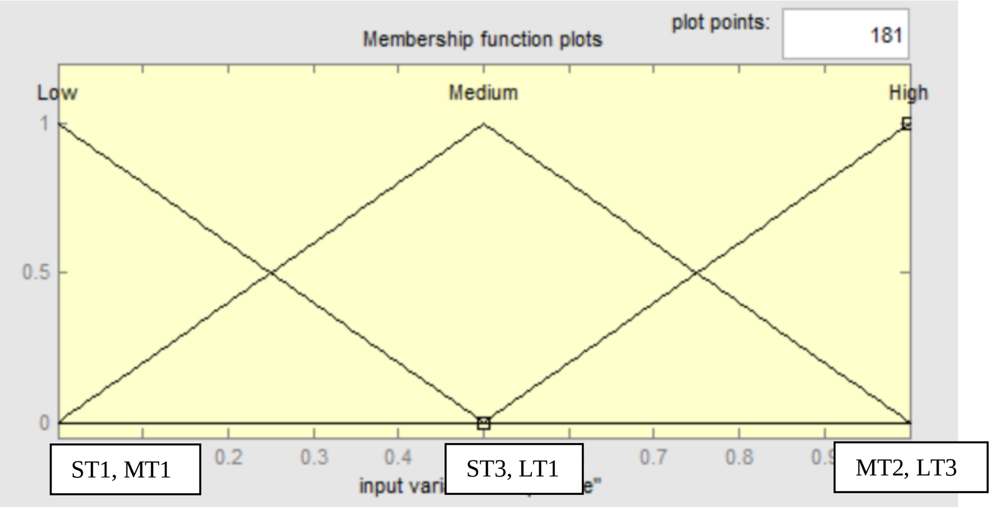


Figure 3.9 Membership function Model A

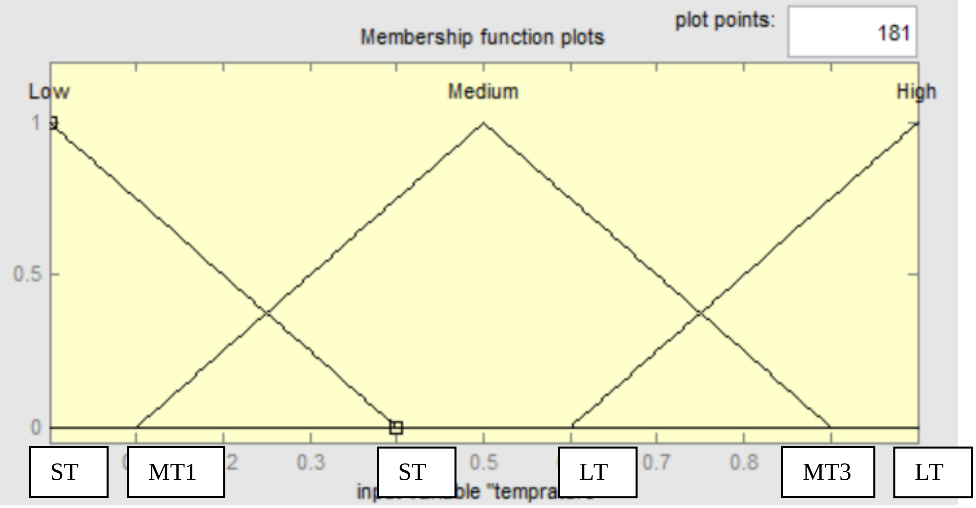


Figure 3.10 Membership function Model B

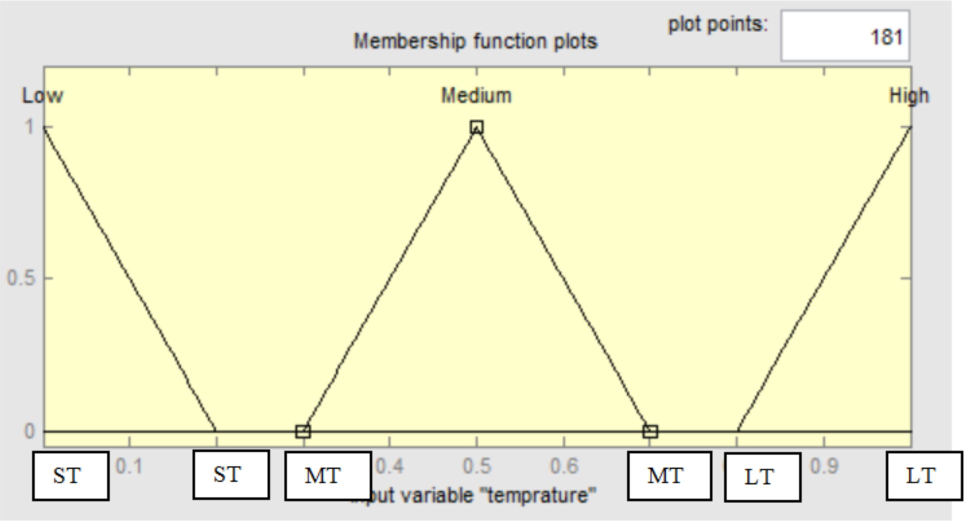


Figure 3.11 Membership function Model C

As shown in Figure 9, Figure 10 and Figure 11, three different models of membership functions could be generated from data depending on different input temperature values produced by various intensities of fire. Model C is generated if there’s no overlap between input temperature value LT1-LT3, MT1-MT3 and/or ST1-ST3, whereas Model A is generated if there is lots of overlap between LT1-LT3, MT1-MT3 and/or ST1-ST3.

Referring to membership function Model A and Model C, Model B normally membership function model for fuzzy logic system is chosen with appropriate overlap. Fortunately, it's easy and reasonable for an expert to modify the original membership function only gathered from data in order to satisfy the requirement of fuzzy logic system.

3.3.4.3 Type2 Membership Function

It's common to have several experts working on same problem. Different type-1 fuzzy logic system would be built by different experts using their specific expertise. It's not appropriate to arbitrarily choose from one expert over another. A better method is to combine different type-1 fuzzy logic membership functions to form a new type-2 membership function. In our proposed approach, three different type-1 membership functions were built for single fuzzy set by our three experts. These type-1 membership functions were then used to structure a new type-2 membership function.

As we have mentioned, the robot will activate different fuzzy logic sub system based on the distance between itself and the fire source to obtain more accurate results. These three sub systems named Far-Module Fuzzy Logic System, Medium-Module Fuzzy Logic System and Close-Module Fuzzy Logic System are introduced in this section. How we designed and developed type-2 membership functions for these fuzzy logic sub systems for temperature input are also discussed separately in this section.

Membership Function Designing for Far-Module Fuzzy Logic

Table 3.8(a): Temperature data for far distance

linguistic dis (m)	dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
far	9	32.35	27.41	22.47	20
	10	30	26	22	20
	11	28.26	24.96	21.65	20

Table 3.8(b): Temperature classified by expert

dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
far	high temperature	medium temperature	low temperature	very low temperature

The membership function model for temperature input for far-module fuzzy logic then can be built based on the data obtained from simulation. Figure 12(a) shows the type-1 fuzzy logic membership function model.

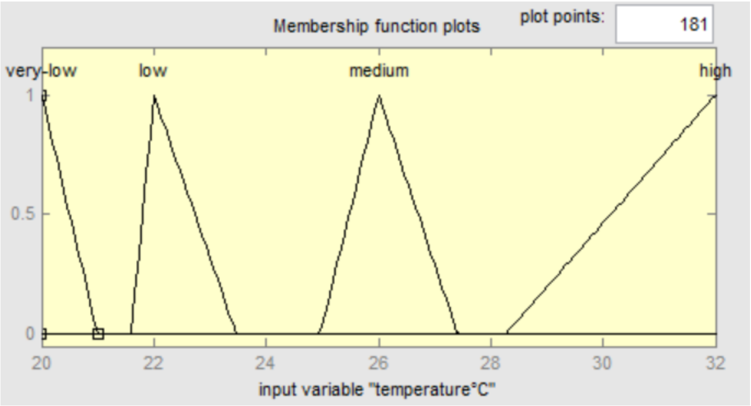


Figure 3.12(a) Membership function model with no overlap for far distance temperature from data

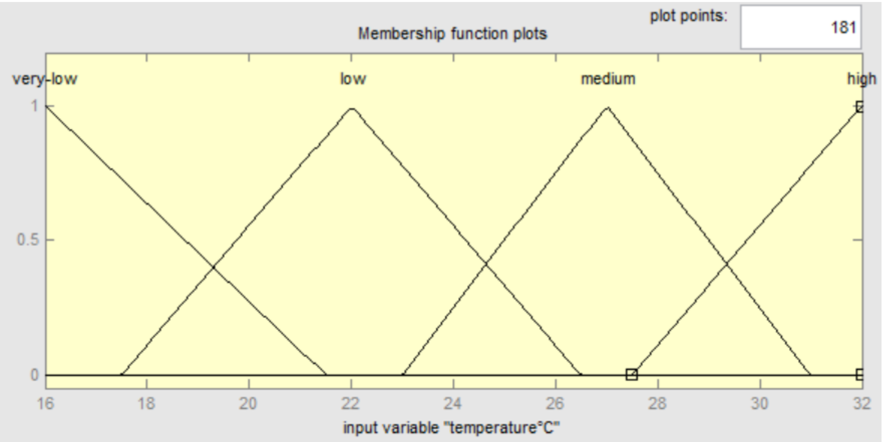


Figure 3.12(b) Membership function for far distance temperature from expert A

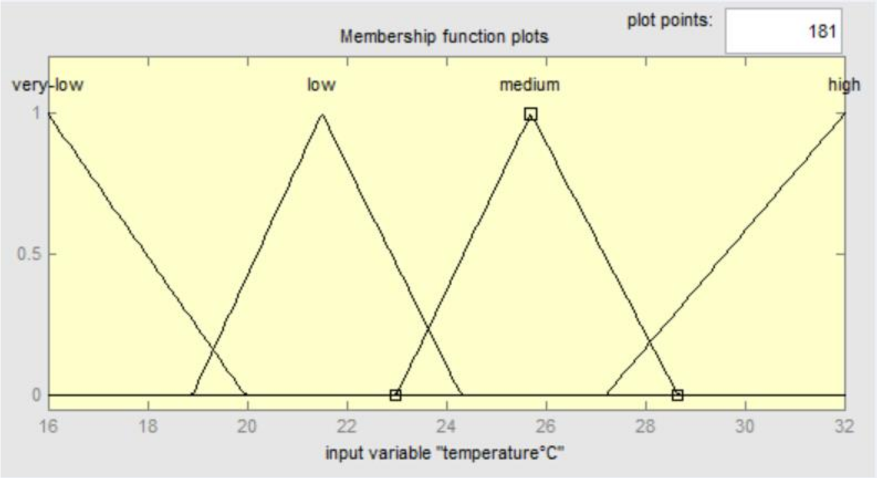


Figure 3.12(c) Membership function for far distance temperature from expert B

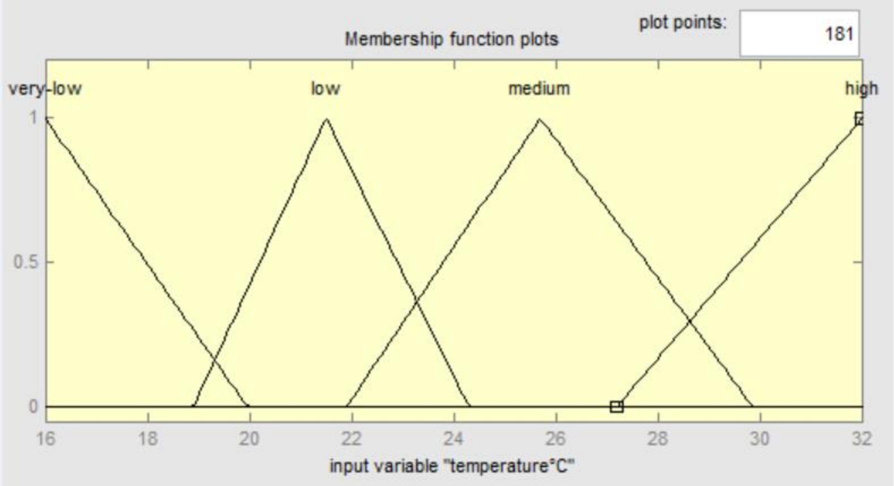


Figure 3.12(d) Membership function for far distance temperature from expert C

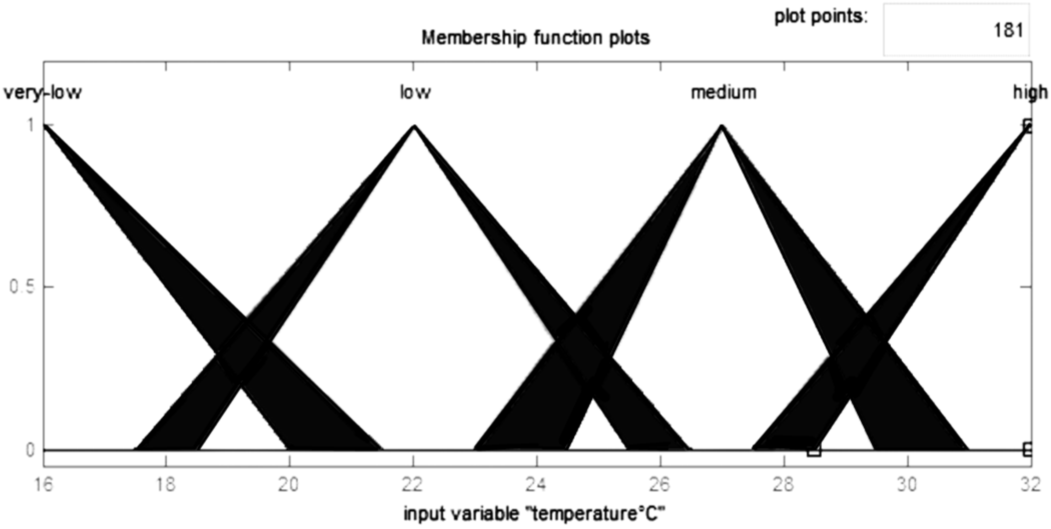


Figure 3.12(e) Mixed Type-2 Membership function for far distance temperature

As shown in Figure 12(b), 12(c) and 12(d), three different type-1 fuzzy logic membership functions were developed individually by our three experts based on the model. Finally, the type-2 membership function, which shows in 12(e), was generated by mixing three type-1 membership functions.

Membership Function for Medium-Module Fuzzy Logic

Table 3.9(a): Temperature data for medium distance

linguistic dis (m)	dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
medium	6	47.78	36.67	25.56	20
	7	40.41	32.24	24.08	20
	8	35.63	29.38	23.13	20

Table 3.9(b): Temperature classified by expert

dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
medium	high temperature	medium temperature	low temperature	very low temperature

The membership function model for temperature input for medium-module fuzzy logic then can be built based on the data obtained from simulation. Figure 13(a) shows the type-1 fuzzy logic membership function model.

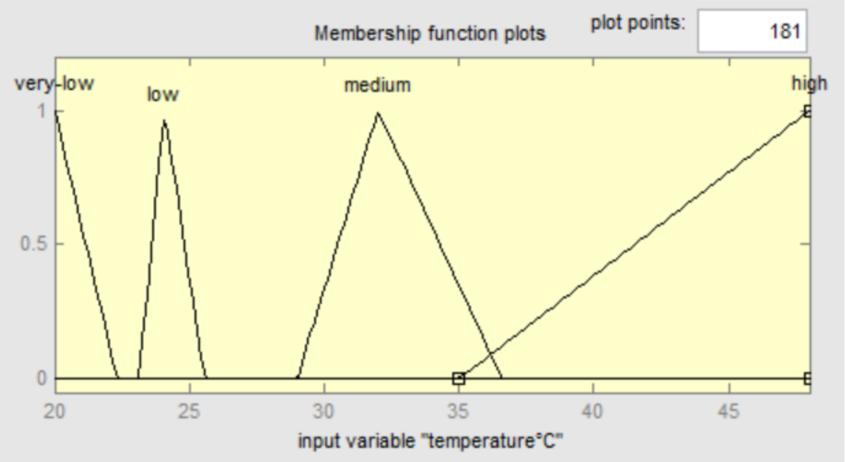


Figure 3.13(a): Membership function model with no overlap for medium distance temperature from data

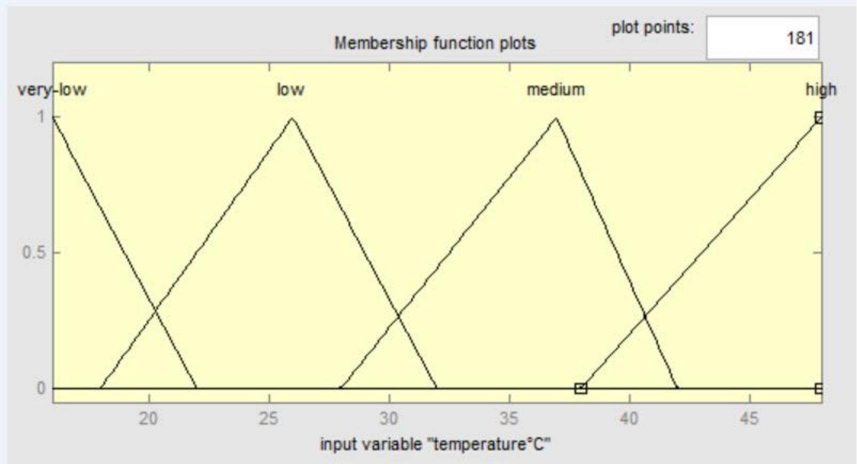


Figure 13(b): Membership function for medium distance temperature from expert A

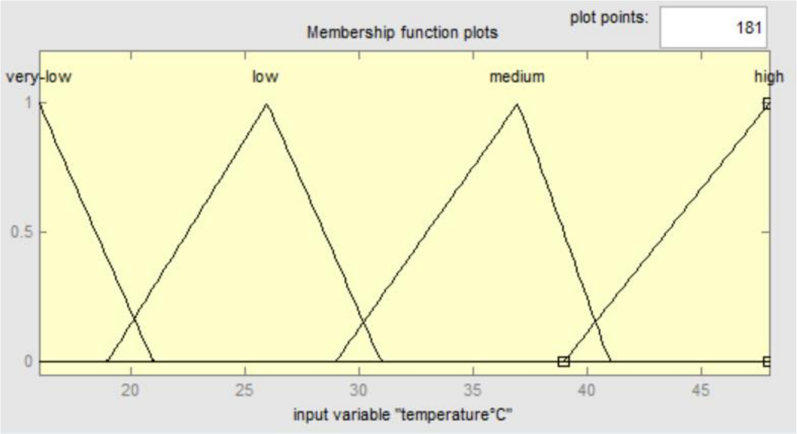


Figure 3.13(c) Membership function for medium distance temperature from expert B

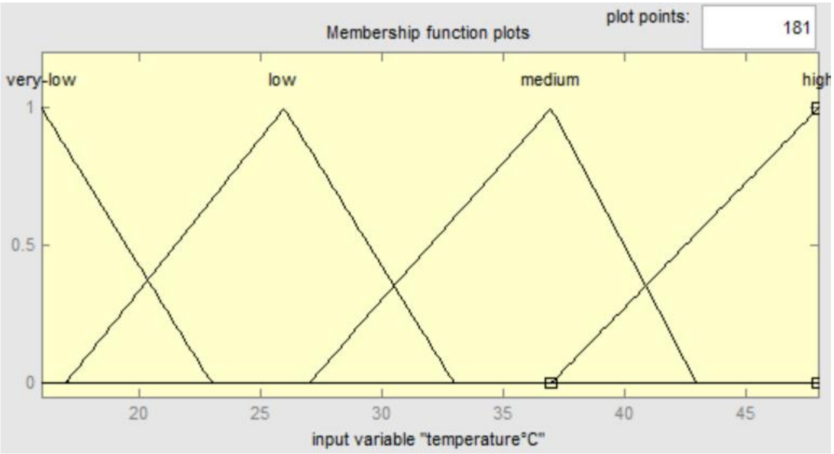


Figure 3.13(d) Membership function for medium distance temperature from expert C

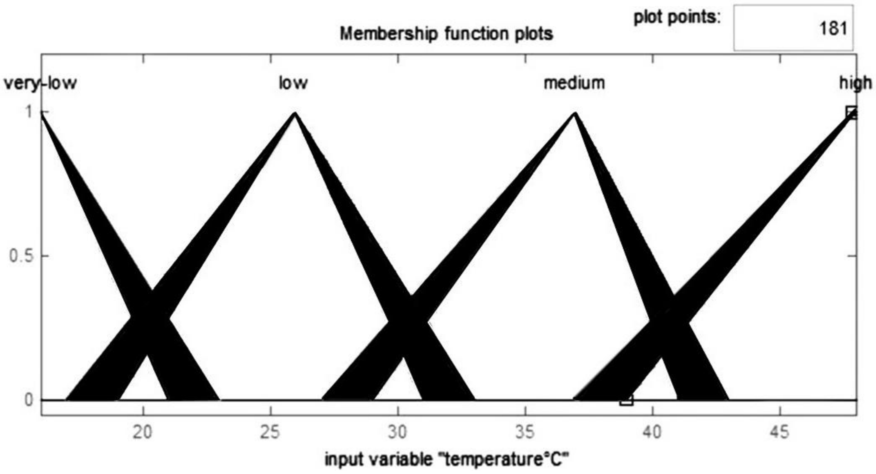


Figure 13(e): Mixed Type-2 Membership function for medium distance temperature

Similar as before, the model of type-1 fuzzy logic membership function is shown in 13(a). As shown in 13(b), 13(c) and 13(d), three different type-1 fuzzy logic membership functions were developed individually by our three experts based on the model. Finally, the type-2 membership function, which shows in 13(e), was generated by mixing three type-1 membership functions.

Membership Function for Close -Module Fuzzy Logic

Table 3.10(a): Data from Simulation

linguistic dis (m)	dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
close	3	131.11	86.67	42.22	20
	4	82.5	57.5	32.5	20
	5	60	44	28	20

Table 10(b): Temperature classified by expert

dis(m)\temp	Large Fire°C	Medium Fire°C	Small Fire°C	No Fire°C
Close	high temperature	medium temperature	low temperature	very low temperature

The membership function model for temperature input for close-module fuzzy logic then can be built based on the data obtained from simulation. Figure 14(a) shows the type-1 fuzzy logic membership function model.

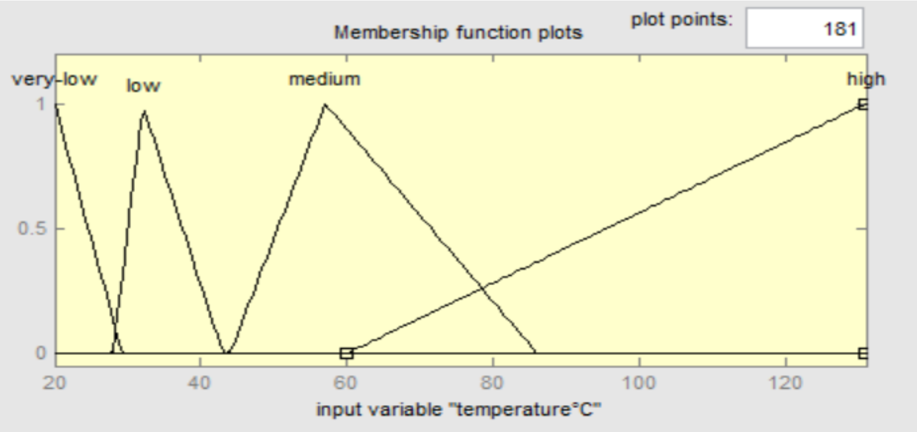


Figure 3.14(a) Membership function model for temperature from data

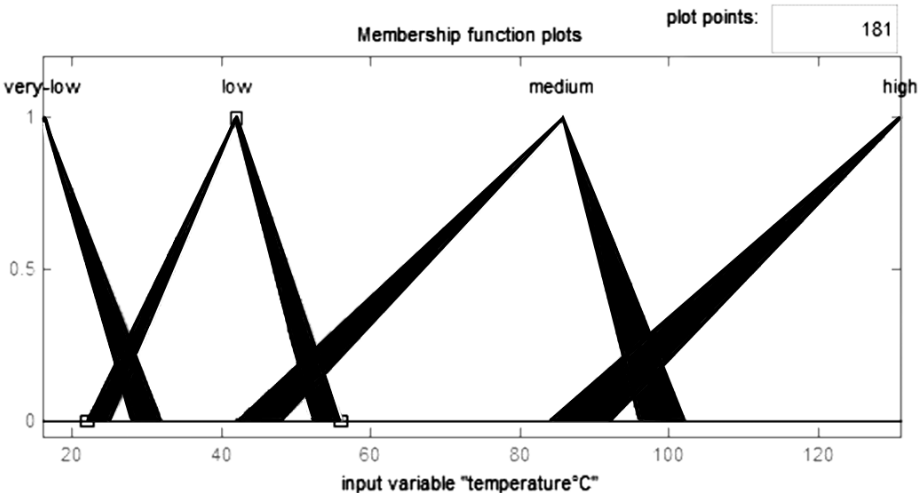


Figure 3.14(b): Mixed Type-2 Membership function for temperature

The formed type-2 membership function for temperature in put for close-distance fuzzy module is shown in 14(b).

3.3.4.4 Fuzzy inference rule bases

Fuzzy inference rule bases should be developed by experts using their expertise which can be gathered either from experience or from new data for a specific solution. Expertise training for experts for fire estimation purpose has been illustrated in last section. With appropriate design for both input fuzzy sets and output fuzzy sets, fuzzy rules can be designed comparative easily for our proposed type-2 fuzzy logic system by our experts.

Table 3.11: Fuzzy Inference Rules Base

Estimated Fire Intensity	Light Value Very Low	Light Value Low	Light Value Medium	Light Value High
Temperature Very Low	Very Low	Very Low	Very Low	Very Low
Temperature Low	Very Low	Low	Low	Medium
Temperature Medium	Very Low	Low	Medium	High
Temperature High	Very Low	Medium	High	High

As Table 11 shows, 16 different inference rules were designed for our proposed type-2 fuzzy logic system. These rules were discussed and developed by our experts together with their expertise.

3.3.5 Real Time Computation

Iterative Algorithm with Stop Condition (IASC)

One of the major drawbacks for type-2 fuzzy logic system is its computation complexity. In real time computation, our approach used two efficient algorithms for computing the generalized centroid of IT2 FSs, which is also used in type-reduction of IT2 FLSs. This faster algorithm, called iterative algorithm (KM algorithm) with stop condition. It is based on the fact that y_l first monotonically decreases and then monotonically increases with the increase of k, and y_r in first monotonically increases and then monotonically decreases with the increase of k [20]. The process of calculating y_l and y_r is illustrated bellow:

IASC for computing y_l :

1) The same as Step of KMA for computing y_l .
2) Initialize

$$a = \sum_{n=1}^{N} \underline{x}_n \underline{w}_n \ , \qquad b = \sum_{n=1}^{N} \underline{w}_n$$

$$y_l = \underline{x}_N \, , \qquad l = 0$$

3) Compute

$$l = l + 1$$
$$a = a + \underline{x}_l (\bar{w}_l - \underline{w}_l)$$
$$b = b + \bar{w}_l - \underline{w}_l$$
$$c = a / b$$

4) If $c > y_l$, set $l = l - 1$ and stop otherwise, set $y_l = c$ and go to Step3.

IASC for computing y_r :

1) The same as Step –1 of KMA for computing y_r .
2) Initialize

$$a = \sum_{n=1}^{N} \bar{x}_n \bar{w}_n \ , \quad b = \sum_{n=1}^{N} \bar{w}_n$$

$$y_r = \bar{x}_1 \, , \ r = 0$$

3) Compute

$$r = r + 1$$
$$a = a - \bar{x}_r (\bar{w}_r - \underline{w}_r)$$
$$b = b - \bar{w}_r + \underline{w}_r$$
$$c = a / b$$

4) If $c > y_r$, set $r = r - 1$ and stop otherwise, set $y_r = c$ and go to Step3

Chapter 4

Hardware Architecture

According to our proposed algorithm and equipments in our laboratory, LEGO MINDSTROMS NXT is chosen for our research and experiments. This is a programmable robotics kit released by Lego in 2006. Lego released three generations of MINDSTORMS kit, the Robotics Invention System, the MINDSTORMS NXT 2.0 and the newest MINDSTORMS EV3 until now. In our experiment, we used MINDSTORMS NXT 2.0 to achieve our targets.

The Lego MINDSTORMS NXT 2.0 was launched on 5 August 2009. It contains 619 pieces (includes sensors and motors), two Touch Sensors, an Ultrasonic Sensor, and introduced a new Color Sensor. The NXT 2.0 uses Floating Point operations whereas earlier versions use integer operation. [21]

LEGO provides a friendly programming environment for researchers, students, hobbyists and any other users. Rather than code programming, LEGO NXT is designed for command box programming. However code programming is also supported by LEGO and other third parts if there has request. The NXT supports many kinds of languages such as [21]:

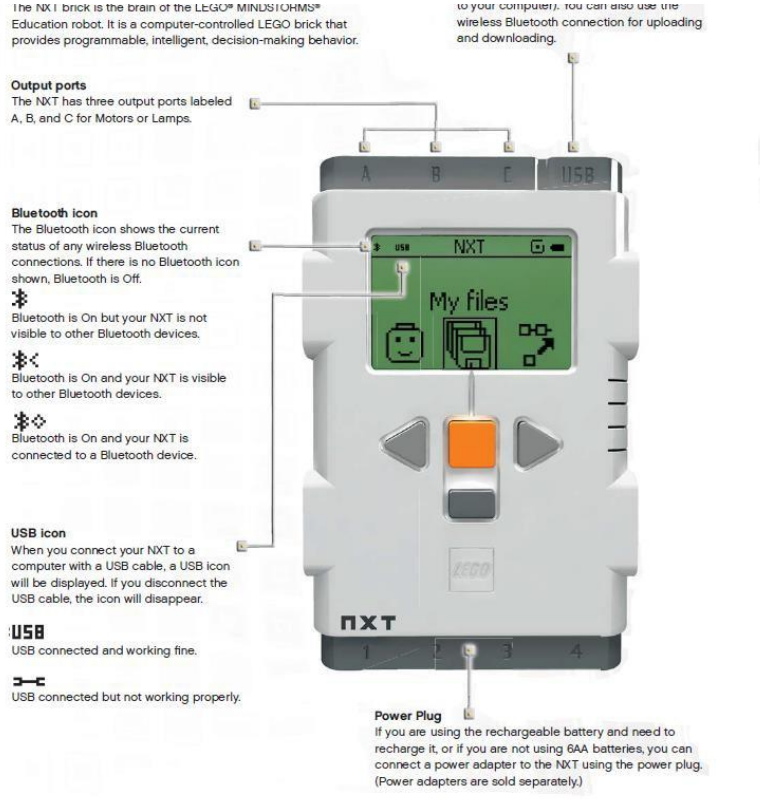
- RCX Code (included in the MINDSTORMS consumer version sold at toystores)
- ROBOLAB (based on LabVIEW and developed at Tufts University)
- Popular third-party languages:

- GNAT GPL: Allows programming NXT using the Ada language for real-time and embedded programming.
- LeJos: A port of Java
- Not eXactly C: (NXC), an open source C-like high-level programming language,
- Not Quite C: (NQC)
- RoboMind: Simple educational scripting language for virtual and LEGO NXT robots.
- ROBOTC: C-Based Programming Language with an Easy-to-Use Development Environment.
- Simulink: Graphical Signal Processing and Control Design tool from which C code is auto-generated and deployed onto the NXT.
- pbFORTH: Extensions to Forth
- pbLua: Version of Lua
- Visual Basic: Via the COM+ interface supplied on the CD

This chapter describes the hardware components of LEGO NXT 2.0 robot platform as well as the mechanical structure design with respect to our experiments. Section 4.1 provides an introduction of the NXT Intelligent Brick of LEGO MINDSTORMS robot. Section 4.2 is a description of the sensors which were chosen and used in experiments. Section 4.3 gives a brief introduction of LEGO robot’s servo motors. Finally, section 4.4 gives an overview of the robot mechanical structure design and architecture used in our research.

4.1 NXT Intelligent Brick

The NXT brick is the center brain of the LEGO MINDSTORMS mobile robot. It is a computer-controlled LEGO brick that allows intelligent, programmable and decision-making behavior. The NXT brick has three output ports which labeled A B and C for LEGO Motors or Lamps. At the meantime, the NXT brick supports four input ports for attaching sensors. The four input ports are labeled 1, 2, 3 and 4 respectively. Figure 4.1 and Figure 4.2 illustrate the details of NXT Intelligent Brick.



The NXT brick is the brain of the LEGO® MINDSTORMS® Education robot. It is a computer-controlled LEGO brick that provides programmable, intelligent, decision-making behavior.

to your computer). You can also use the wireless Bluetooth connection for uploading and downloading.

Output ports
The NXT has three output ports labeled A, B, and C for Motors or Lamps.

Bluetooth icon
The Bluetooth icon shows the current status of any wireless Bluetooth connections. If there is no Bluetooth icon shown, Bluetooth is Off.

Bluetooth is On but your NXT is not visible to other Bluetooth devices.

Bluetooth is On and your NXT is visible to other Bluetooth devices.

Bluetooth is On and your NXT is connected to a Bluetooth device.

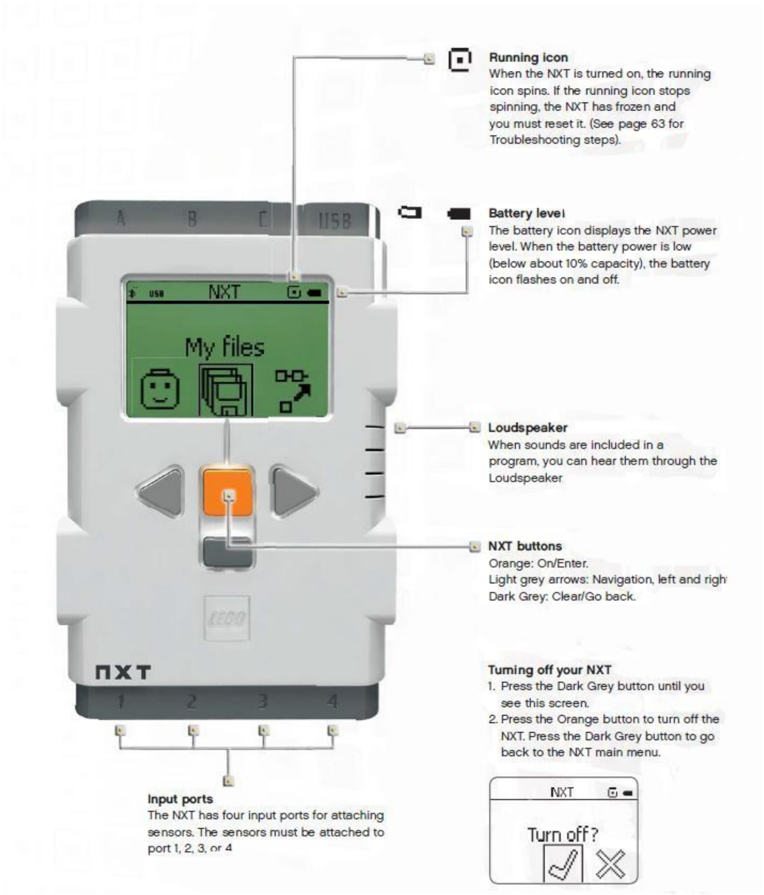
USB icon
When you connect your NXT to a computer with a USB cable, a USB icon will be displayed. If you disconnect the USB cable, the icon will disappear.

USB
USB connected and working fine.

USB connected but not working properly.

Power Plug
If you are using the rechargeable battery and need to recharge it, or if you are not using 6AA batteries, you can connect a power adapter to the NXT using the power plug. (Power adapters are sold separately.)

Figure 4.1 Illustration of NXT intelligent brick



Running icon
When the NXT is turned on, the running icon spins. If the running icon stops spinning, the NXT has frozen and you must reset it. (See page 63 for Troubleshooting steps).

Battery level
The battery icon displays the NXT power level. When the battery power is low (below about 10% capacity), the battery icon flashes on and off.

Loudspeaker
When sounds are included in a program, you can hear them through the Loudspeaker

NXT buttons
Orange: On/Enter.
Light grey arrows: Navigation, left and righ
Dark Grey: Clear/Go back.

Turning off your NXT
1. Press the Dark Grey button until you see this screen.
2. Press the Orange button to turn off the NXT. Press the Dark Grey button to go back to the NXT main menu.

Input ports
The NXT has four input ports for attaching sensors. The sensors must be attached to port 1, 2, 3, or 4

Figure 4.2 Illustration of NXT intelligent brick

The details regarding to NXT Intelligent Brick’s functions are listed as follow [22]

- 32-bit Atmel AT91SAM7S256 main microcontroller (256 KB flash memory, 64 KB RAM)
- 8-bit Atmel ATmega48 microcontroller @ 4 MHz (4 KB flash memory, 512 Bytes RAM)
- 100x64 pixel LCD Screen
- four 6-pin input ports (ports 1-4)
- three 6-pin output ports (ports A-C)
- USB Port

- Bluetooth Class II V2.0
- Loudspeaker - 8 kHz sound quality, 8-bit resolution, 2–16 kHz sample rate
- Four Push Buttons
- Orange button: On/Enter
- Light grey arrows: moving left and right in the NXT menu
- Dark grey button: Clear/Go back
- Powered by six AA batteries or the NXT Rechargeable DC Battery

As we just mentioned, The NXT Intelligent Brick can take input from up to four different sensors and control output up to three motors. In the experiment for our research, four input ports are connected to 2 Light Sensors, a Thermal Infrared Sensor (TIR Sensor) and an Ultrasonic Sensor. Two output ports are connected to left servo step motor and right servo step motor respectively which are used to drive the NXT Robot. Details about the sensors used in our experiment will be illustrated in the

4.2 Sensors

Sensors are devices which we use to measure physical variables such as temperature, pH value, flow rate, rotational rate, velocity, pressure and many others. Rather than provide an analog scale reading (like a thermometer), the modern sensors usually produce a voltage or a digital signal that is indicative of the physical variable they measured. Those signal data are usually transferred to computer programs, stored in files, analyzed and plotted on computers. [23] All sensors we implemented for our mobile robot will be introduced in this section. Table 4.1 shows the overall usage of sensors in our research work.

Sensor Name	Quantity	Usage
Ultrasonic Sensor	1	Distance Measure
Light Sensor	2	Detect Light
TIR Sensor	1	Detect Temperature
Rotation sensor	3	Measurement of robot turning angle

Table 4.1 Usage of sensors in research

4.2.1 Ultrasonic Sensor

Ultrasound is a sound pressure wave with oscillating with a frequency greater than the upper limit of the human hearing range. The approximately upper boundary of a sound that people can hear is about 20000 hertz. If the frequency of a sound is higher than 20000 hertz, it is called ultrasound. Figure 4.3 illustrates the frequency range. [24]

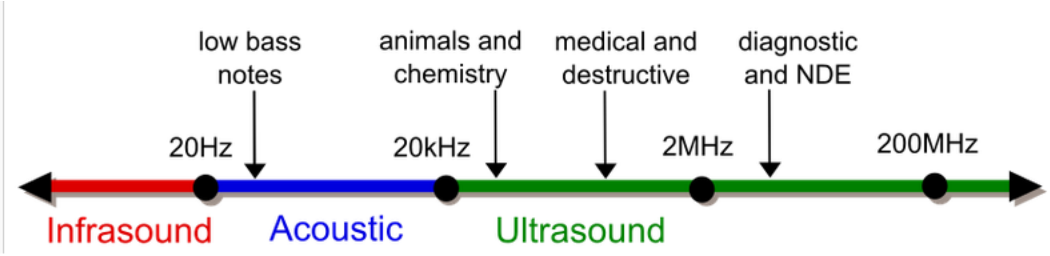


Figure 4.3 Approximate frequency ranges corresponding to ultrasound

The Ultrasonic Sensor allows robot to see and detect objects. It is usually used by a robot to avoid obstacles, sense and measure distance, and detect movement. Figure 4.3 shows the features of an ultrasonic sensor.

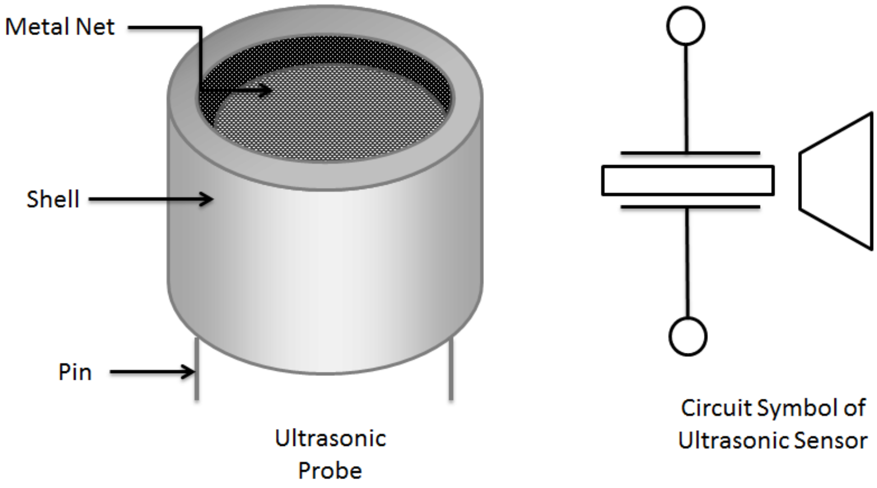


Figure 4.4 An Ultrasonic probe and its symbol

An ultrasonic sensor is consisted as an ultrasonic transmitter, an ultrasonic receiver, control circuit and power source. Ultrasonic Sensor applies the same principle as bats: it measures the distance by calculating the time it takes for a sound wave to hit an object and return. It works just like an echo. Figure 4.5 and Figure 4.6 shows the principle of ultrasonic ranging and how it is achieved by ultrasonic sensor. [24]

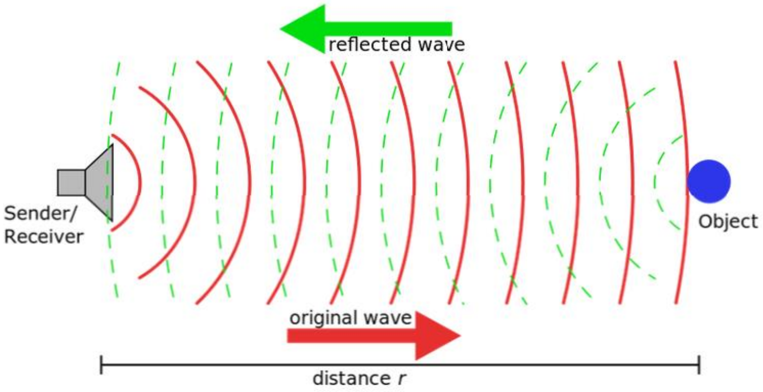


Figure 4.5 Principle of ultrasonic ranging

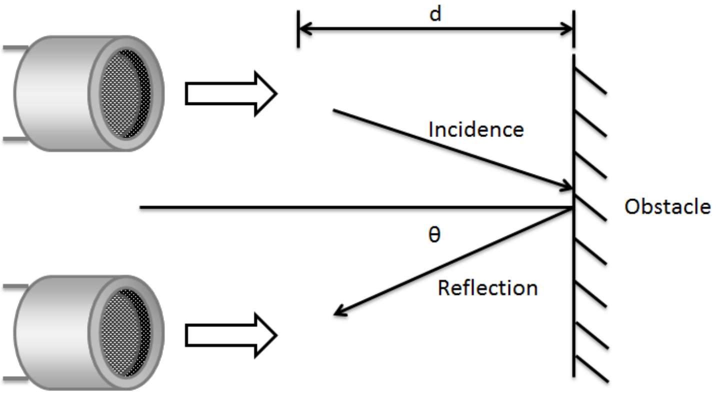


Figure 4.6 Principle of ultrasonic ranging by sensor

Ultrasonic transmitter launches ultrasonic wave in a given direction and at the same time the timer starts to calculate. When the ultrasonic wave hits a wall or an obstacle, it reflects and transmits back to the ultrasonic receiver. After the ultrasonic receiver catches the reflected wave, the timer will stop counting immediately. The distance d between obstacle and the ultrasonic transmitter can be calculated by:

$$D = \frac{t \cdot v}{2}[m] \tag{5.1}$$

where t is the time period measured by the timer and v is the velocity of ultrasonic wave in the air medium and can be expressed as Function 4.2. [25]

$$v = 331.5 + 0.6T[m/\sec] \tag{5.2}$$

where T is the air temperature in degrees Celsius.

Figure 4.7 LEGO MINDSTORM ultrasonic sensor

The LEGO MINSSTORM ultrasonic sensor is shown in Figure 4.7. The ultrasonic sensor is able to measure distance both in inches and in centimeters. It can measure distances from 0 to 255 centimeters with a precision of +/- 3 cm. Large sized objects with hard surfaces return better readings, whereas objects made of soft fabric such as a ball, or are very thin or small can be more difficult for the LEGO ultrasonic sensor to detect.

4.2.2 Thermal Infrared Sensor

The Thermal Infrared Sensor (TIRS) will measure land surface temperature in two thermal bands with a new technology that applies quantum physics to detect heat. The Thermal Infrared Sensor (TIR) we used in our experiment is developed by Dexter Industries. This sensor is produced specialized for LEGO MINDSTORM robot. By detecting the infrared radiation from an object, the sensor could read object temperatures from -90°F to 700°F (-70℃ and +380℃). The Thermal Infrared Sensor can read both the ambient temperature (the temperature of surrounding air) and the surface temperature of the object which the TIR sensor is pointed towards. Figure 4.8 and Figure 4.9 shows the front side and the backside of the sensor. [26]

Figure 4.8 The front side of Thermal Infrared Sensor

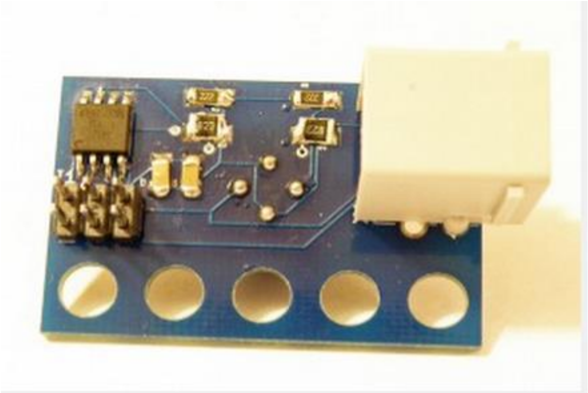


Figure 4.9 The backside of Thermal Infrared Sensor

The features of the Thermal Infrared Sensor used in our experiment are shown in list below:

- Capacity of reading surface temperature of the objects from a distance
- Capability of reading object temperatures between -70 ℃ and +380 ℃.
- Have an accuracy of 0.5 ℃ and a resolution of 0.02 ℃.
- It is capable of reading both ambient and surface temperatures
- Capable of reading the emissivity values.
- The detection angle of vision is total of 90 °, from -45 ° to 45 °.
- NXT-G blocks have been developed for the Thermal Infrared Sensor prior by its manufactory.

4.2.3 Color sensor

LEGO color sensor uses RGB LED and it continuously shines red, blue and green light on the object. The reflected light from the object is then collected by a light sensor which is sensitive to all length of wave. The color sensor used for our research is shown in Figure 4.10.

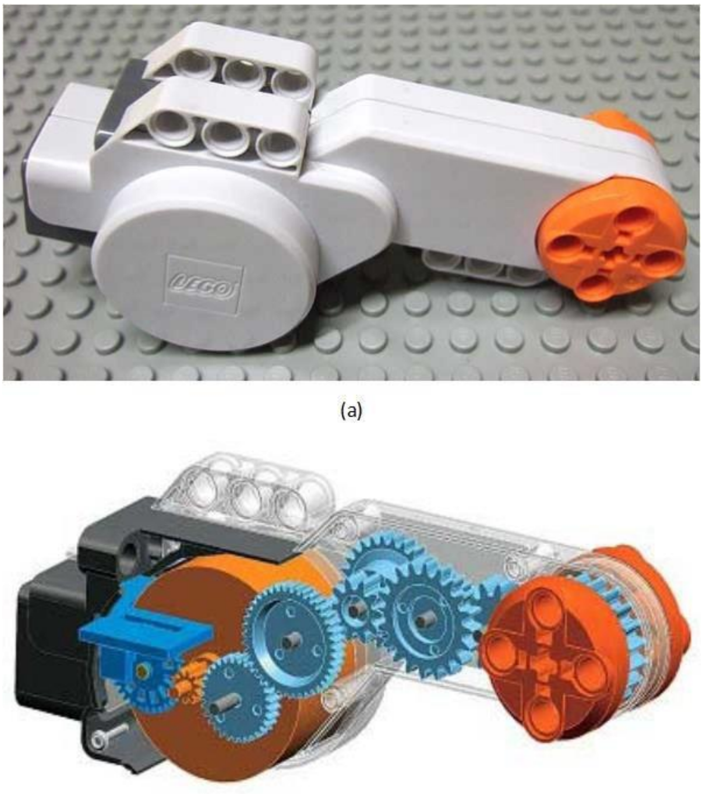
Figure 4.10 Mindstorms® NXT Color Sensor

The features of the Color Sensor are shown in list below:

- Using the NXT brick, the Color Sensor is able to perform three unique functions.
- It can work as a Color Sensor distinguishing between six colors
- It can act as a Light Sensor detecting light intensities for both reflected and ambient light.
- It works as a Color Lamp, emitting red, green or blue light.
- It detects colors using separate Red, Green and Blue (RGB) components
- The angle of vision of this sensor is also 45 °on either side of the normal axis.
- To detect color, it needs to be very close to the colored object (1~2 cm).
- Its output values range from 0 to 255 for the luminance. [28]

4.3 Servo Motor

The LEGO NXT Servo Motor has a built-in rotation sensor that is used to measure speed and distance, and send information back to the NXT Intelligent Brick. This allows NXT bricks to achieve precise steps and motor control within one degree of accuracy. Multiple motors can be aligned to drive the LEGO robot at the same speed. The structure and design of the servo motor are shown in Figure 4.11. Table 4-2 gives some main characteristics of the LEGO NXT servo motor.



(a)

(b)

Figure 4.11 (a) NXT servo motor (b) Mechanical design of NXT servo motor [27]

Table 4.2 Main characteristics of the NXT servo motor

Characteristics		**Values**
Weight		80g
No-load	Rotation Speed	170rpm
	Current	60mA
Stalled	Torque	50N*cm
	Current	2A
Loaded Characteristics (at minimum 4.5V Voltage)	Torque	16.7N*cm
	Rotation Speed	33rpm
	Current	600mA
	Mechanical Power	0.58W
	Electrical Power	2.7W
	Efficiency	21%
Loaded Characteristics (at maximum 12V Voltage)	Torque	16.7N*cm
	Rotation Speed	177rpm
	Current	580mA
	Mechanical Power	3.10W
	Electrical Power	6.96W
	Efficiency	45%

The curves in Figure 4.12 [27] below shows NXT motor rotation speed (Rotations per Minute) vs. motor power level (supply duty cycle).

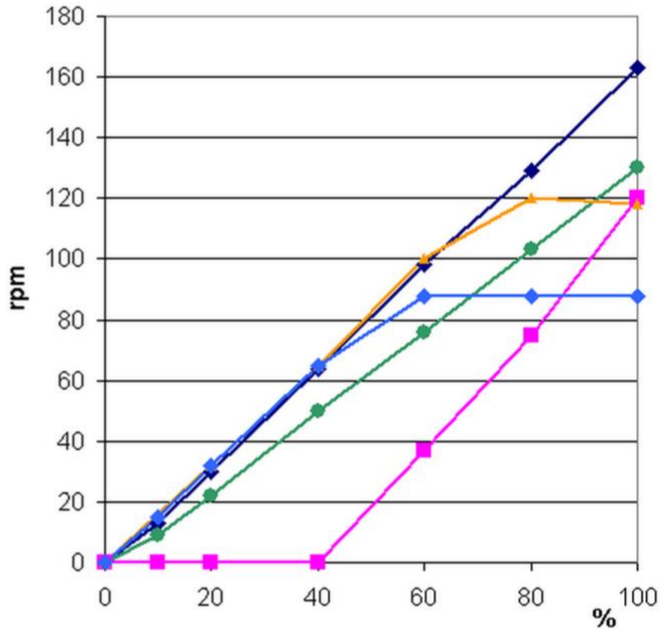


Figure 4.12 NXT motor rotation speed vs. motor power level

—◆— represents motor not loaded, 9V NXT power.

—●— shows motor not loaded, 7.2V NXT power.

—■— is when motor behavior with a 11.5 N.cm load applied, no Power Control, 9V NXT power.

—▲— is the curve for motor loaded with a 11.5 N.cm, 9V NXT power.

—◆— denotes the motor loaded with a 11.5 N.cm, 7.2V NXT power.

4.4 Mechanical Structure of LEGO NXT Robot Based on Experiments

The mechanical structure of robot was designed to meet requirements of experiments done for this thesis research. Mechanical structure of the robot is easy to design, architecture and change according to different experiment demands because of the smart LEGO BRICK design. The overall design of the robot for fire detection and estimation is shown in Figure 4.13 (a), (b), (c) and (d).

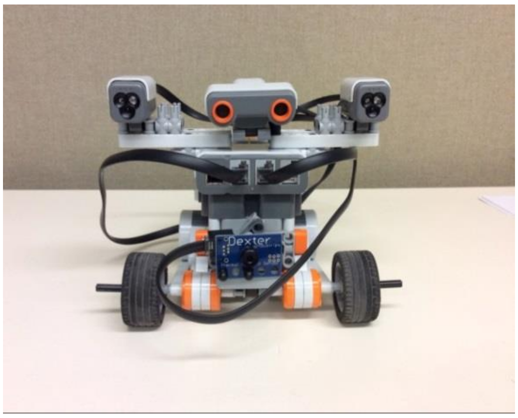


Figure 4.13(a)

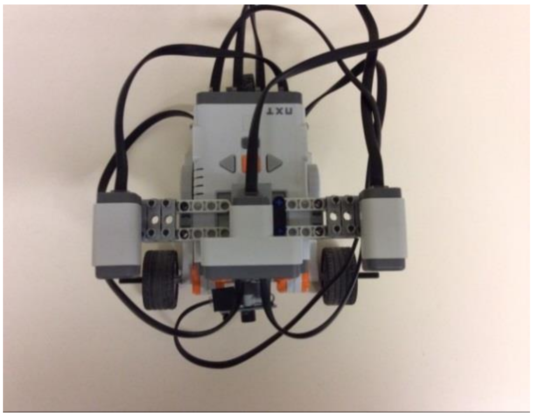


Figure 4.13(b)

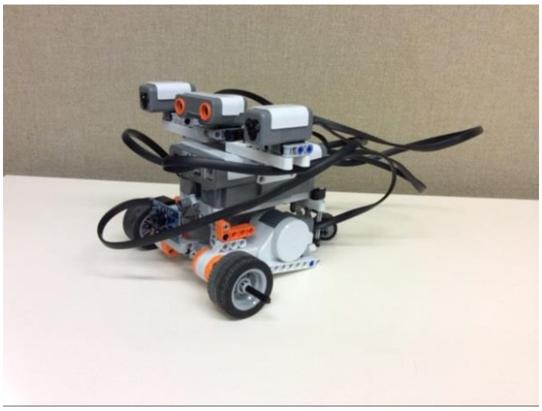


Figure 4.13(c)

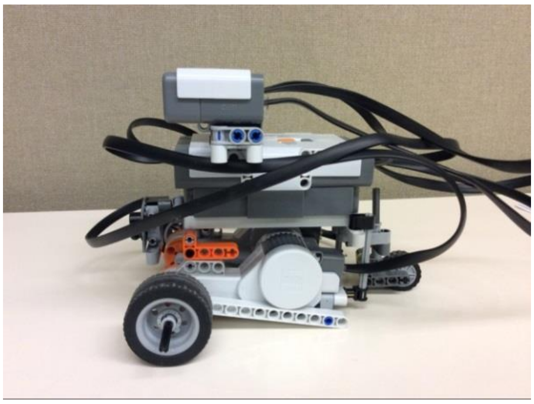


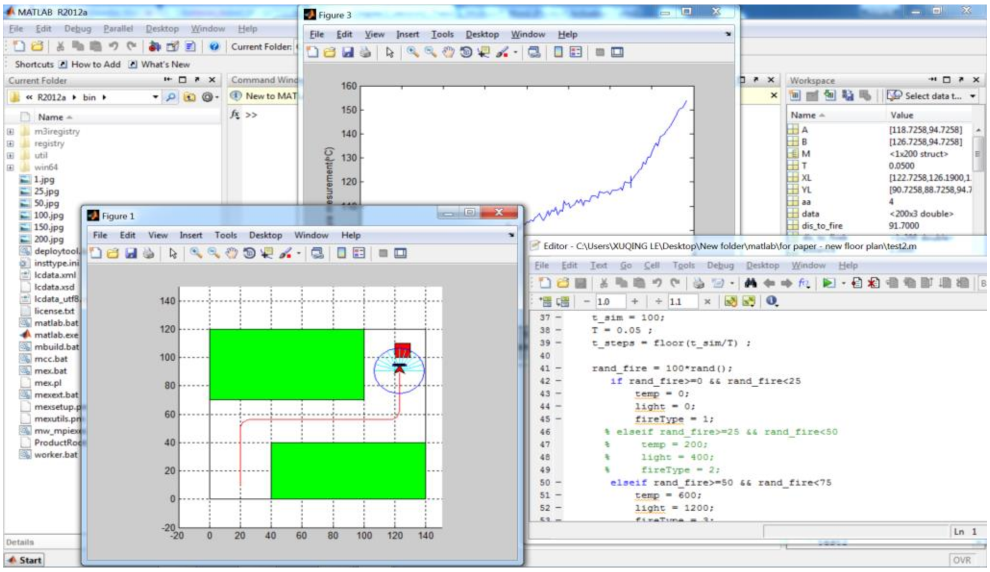
Figure 4.13(d)

Chapter 5

Software Architecture

5.1 MATLAB^{TM}

The software we used for research simulation is MATLAB 2012a developed by MathWorks. MATLAB is a matrix based computing environment and a high-level programming language utilized by engineers and scientists for quick data analyzing, developing and testing algorithms and creating models and applications. Because of the MATLAB language, toolboxes and built in mathematic functions, MATLAB enables users to reveal solutions faster than these platforms using traditional programming languages such as C/C++ or Java. MATLAB is capable of robotic simulations due to its powerful matrix computation and algorithms simulation. In our research, MATLAB simulations are used to analyze the feasibility of our proposed algorithm in different situations. In such case experiments for our research can be processed in next step based on the algorithm we proposed. Figure 5.1 shows the operation interface of MATLAB.



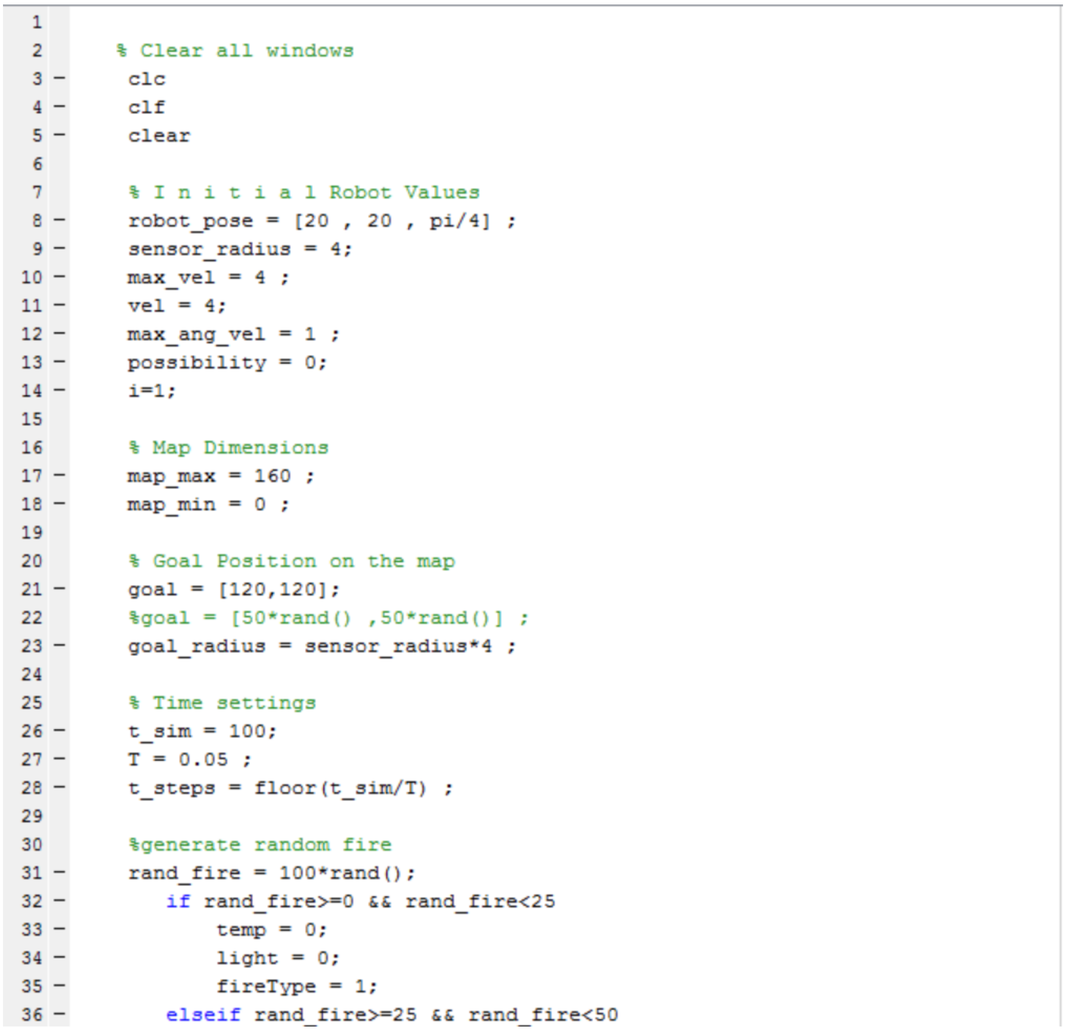
Name	Value
A	[118.7258,94.7258]
B	[126.7258,94.7258]
M	<1x200 struct>
T	0.0500
XL	[122.7258,126.1900,1
YL	[90.7258,88.7258,94.7
aa	4
data	<200x3 double>
dis_to_fire	91.7000

```
    t_sim = 100;
    T = 0.05 ;
    t_steps = floor(t_sim/T) ;

    rand_fire = 100*rand();
       if rand_fire>=0 && rand_fire<25
           temp = 0;
           light = 0;
           fireType = 1;
      % elseif rand_fire>=25 && rand_fire<50
      %      temp = 200;
      %      light = 400;
      %      fireType = 2;
       elseif rand_fire>=50 && rand_fire<75
           temp = 600;
           light = 1200;
```

Figure 5.1 Operation interface of MATLAB

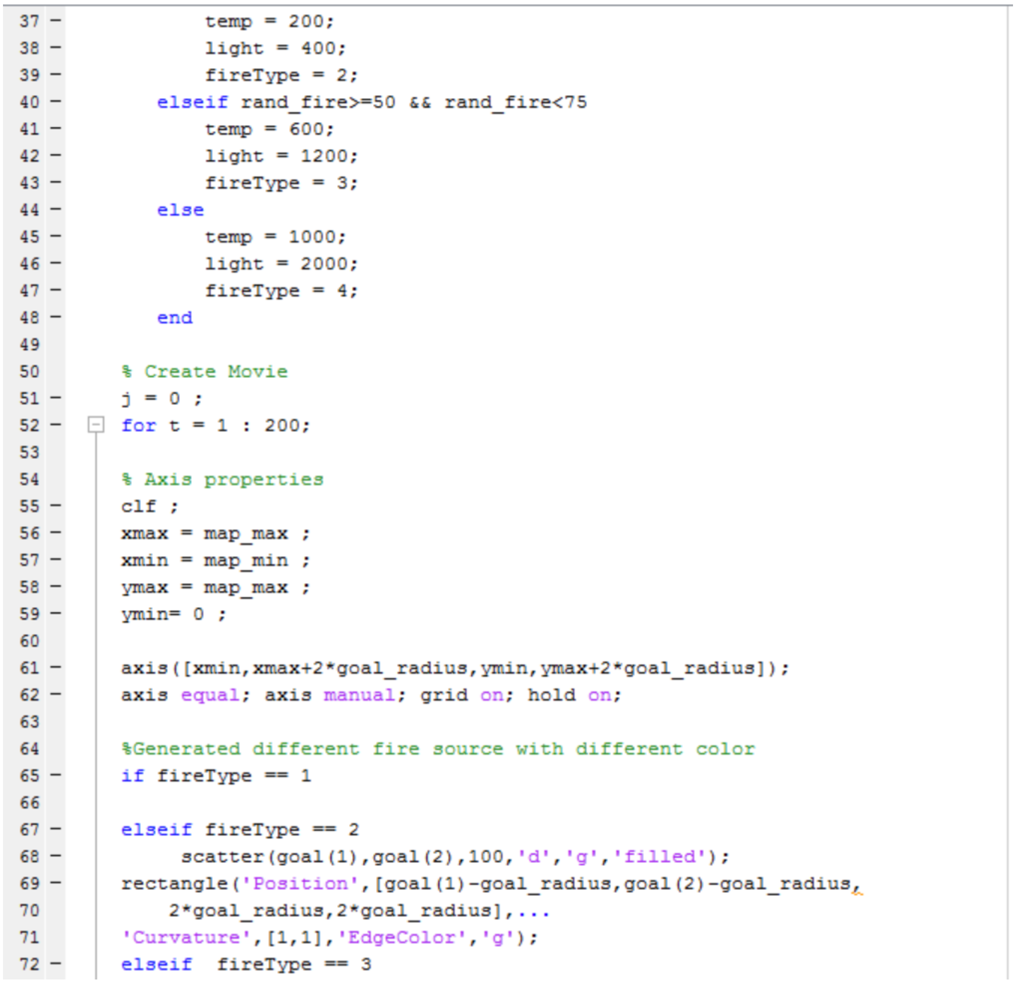
The following piece of code shown in Figure 5.2(a) – (d) is the main function for our application in MATLAB. All MATLAB code will run starting from the main function. The name of the function is "robot_main.m". Several sub-functions are called during the running of main function such as statement 99nd and 103nd.

All the simulation results are illustrated in Chapter 6, at the meantime all programs of MATLAB are shown in Appendix with respect to the simulations in Chapter 6.



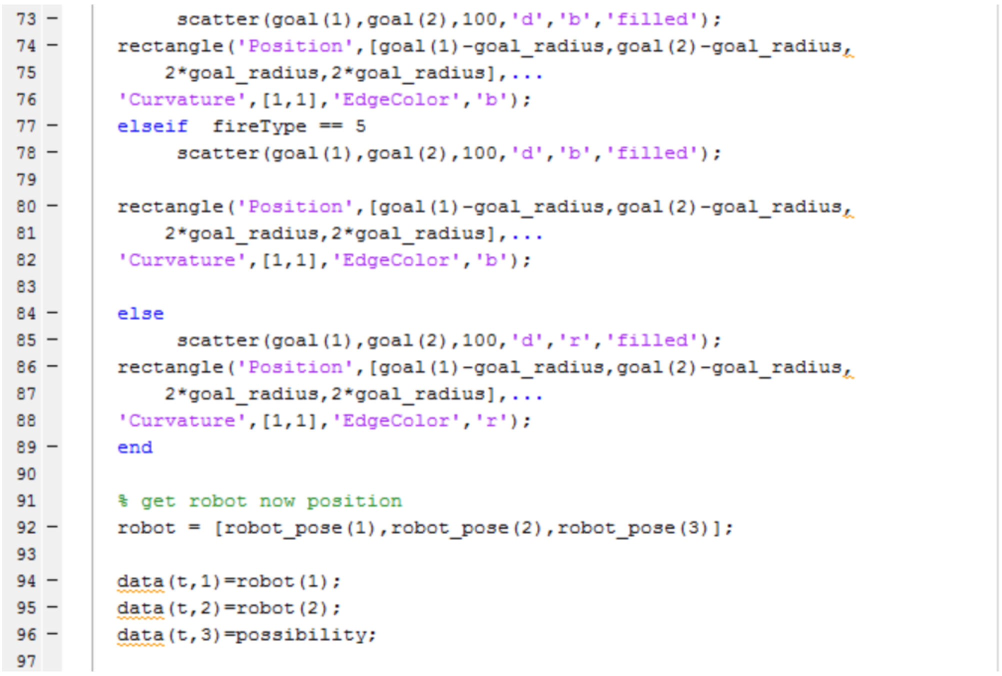
```
 1      
 2       % Clear all windows
 3 -      clc
 4 -      clf
 5 -      clear
 6      
 7        % I n i t i a l Robot Values
 8 -      robot_pose = [20 , 20 , pi/4] ;
 9 -      sensor_radius = 4;
10 -      max_vel = 4 ;
11 -      vel = 4;
12 -      max_ang_vel = 1 ;
13 -      possibility = 0;
14 -      i=1;
15      
16        % Map Dimensions
17 -      map_max = 160 ;
18 -      map_min = 0 ;
19      
20        % Goal Position on the map
21 -      goal = [120,120];
22        %goal = [50*rand() ,50*rand()] ;
23 -      goal_radius = sensor_radius*4 ;
24      
25        % Time settings
26 -      t_sim = 100;
27 -      T = 0.05 ;
28 -      t_steps = floor(t_sim/T) ;
29      
30        %generate random fire
31 -      rand_fire = 100*rand();
32 -         if rand_fire>=0 && rand_fire<25
33 -             temp = 0;
34 -             light = 0;
35 -             fireType = 1;
36 -         elseif rand_fire>=25 && rand_fire<50
```

Figure 5.2(a) Main function written with MATLAB language



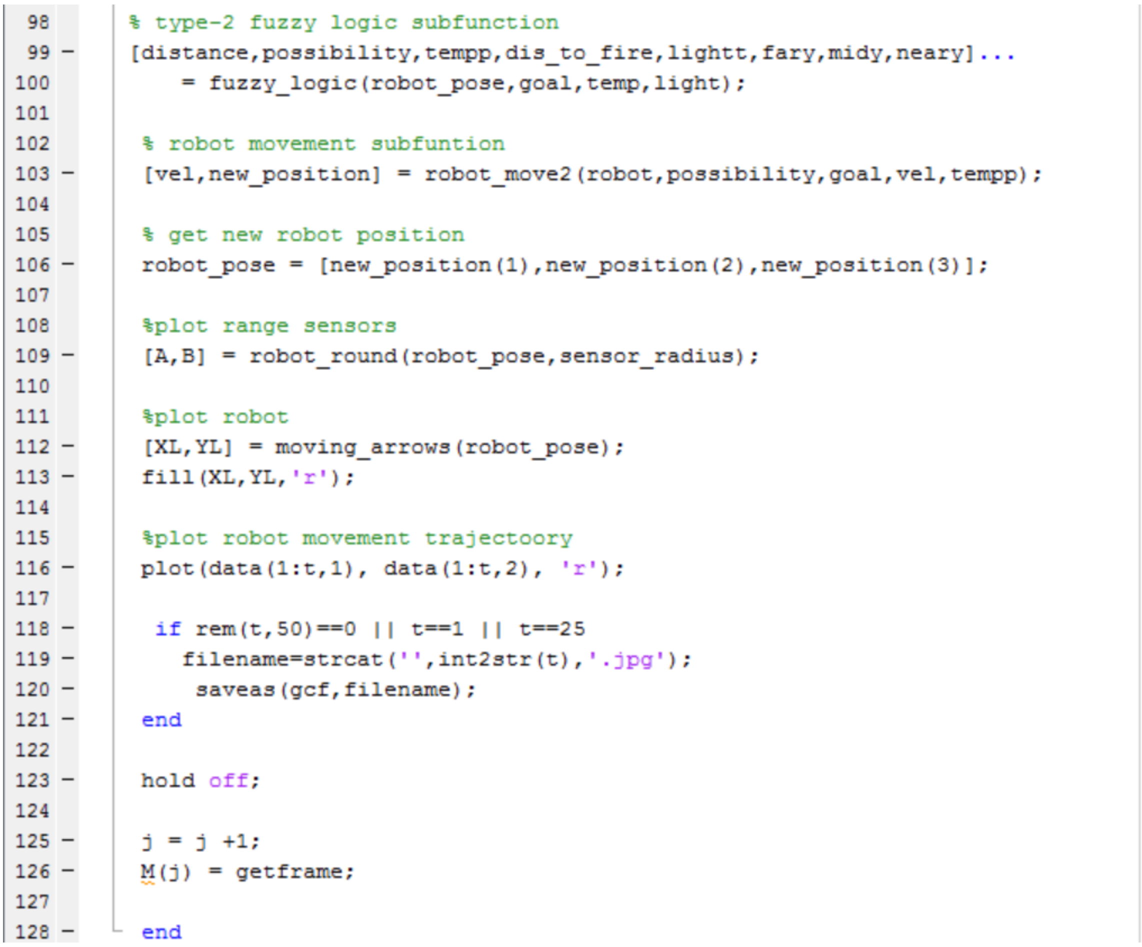
```
       temp = 200;
       light = 400;
       fireType = 2;
   elseif rand_fire>=50 && rand_fire<75
       temp = 600;
       light = 1200;
       fireType = 3;
   else
       temp = 1000;
       light = 2000;
       fireType = 4;
   end

% Create Movie
j = 0 ;
for t = 1 : 200;

% Axis properties
clf ;
xmax = map_max ;
xmin = map_min ;
ymax = map_max ;
ymin= 0 ;

axis([xmin,xmax+2*goal_radius,ymin,ymax+2*goal_radius]);
axis equal; axis manual; grid on; hold on;

%Generated different fire source with different color
if fireType == 1

elseif fireType == 2
     scatter(goal(1),goal(2),100,'d','g','filled');
rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,
    2*goal_radius,2*goal_radius],...
'Curvature',[1,1],'EdgeColor','g');
elseif  fireType == 3
```

Figure 5.2(b) Main function written with MATLAB language



```
     scatter(goal(1),goal(2),100,'d','b','filled');
rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,
    2*goal_radius,2*goal_radius],...
'Curvature',[1,1],'EdgeColor','b');
elseif  fireType == 5
     scatter(goal(1),goal(2),100,'d','b','filled');

rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,
    2*goal_radius,2*goal_radius],...
'Curvature',[1,1],'EdgeColor','b');

else
     scatter(goal(1),goal(2),100,'d','r','filled');
rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,
    2*goal_radius,2*goal_radius],...
'Curvature',[1,1],'EdgeColor','r');
end

% get robot now position
robot = [robot_pose(1),robot_pose(2),robot_pose(3)];

data(t,1)=robot(1);
data(t,2)=robot(2);
data(t,3)=possibility;

```

Figure 5.2(c) Main function written with MATLAB language

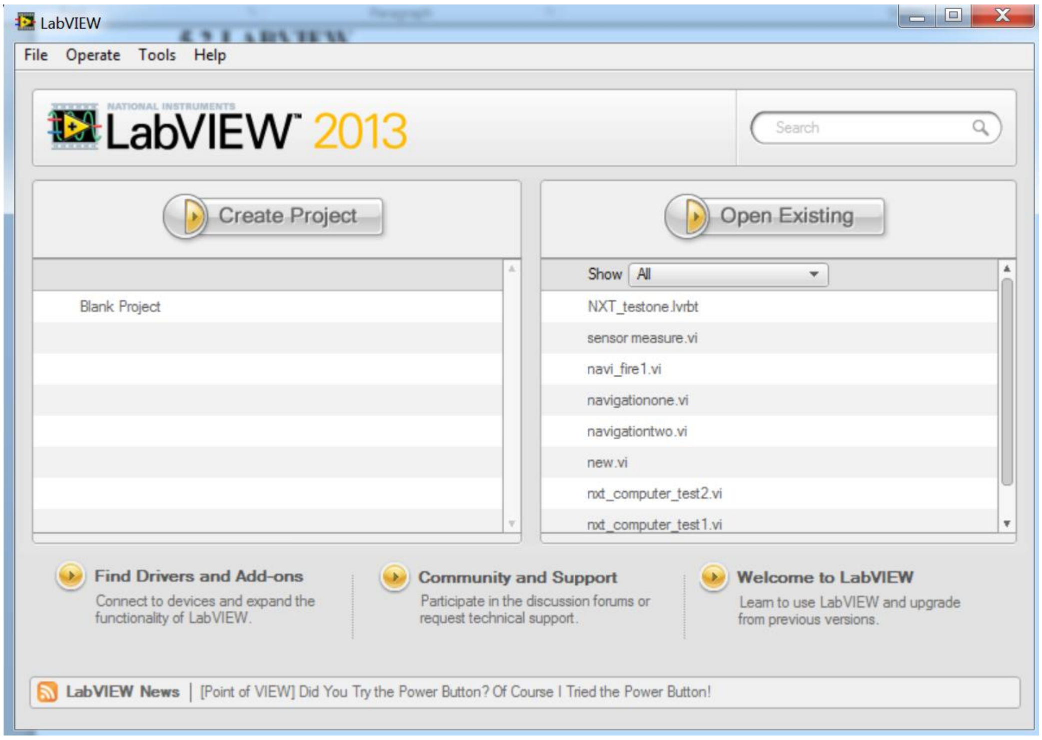


```
% type-2 fuzzy logic subfunction
[distance,possibility,tempp,dis_to_fire,lightt,fary,midy,neary]...
    = fuzzy_logic(robot_pose,goal,temp,light);

 % robot movement subfuntion
 [vel,new_position] = robot_move2(robot,possibility,goal,vel,tempp);

 % get new robot position
 robot_pose = [new_position(1),new_position(2),new_position(3)];

 %plot range sensors
 [A,B] = robot_round(robot_pose,sensor_radius);

 %plot robot
 [XL,YL] = moving_arrows(robot_pose);
 fill(XL,YL,'r');

 %plot robot movement trajectoory
 plot(data(1:t,1), data(1:t,2), 'r');

  if rem(t,50)==0 || t==1 || t==25
    filename=strcat('',int2str(t),'.jpg');
     saveas(gcf,filename);
 end

 hold off;

 j = j +1;
 M(j) = getframe;

 end
```

Figure 5.2(d) Main function written with MATLAB language

5.2 NI LABVIEW™

LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) is a platform and development environment for graphical programming developed by National Instruments. The common usage for NI LabVIEW is data acquisition, device control, and industrial automation. It is a development environment for problem solving, accelerated productivity, and continual innovation. [28] Figure 5.3 shows the workbench of LabVIEW.



Blank Project

NXT_testone.lvrbt
sensor measure.vi
navi_fire1.vi
navigationone.vi
navigationtwo.vi
new.vi
nxt_computer_test2.vi
nxt_computer_test1.vi

Connect to devices and expand the functionality of LabVIEW.

Participate in the discussion forums or request technical support.

Learn to use LabVIEW and upgrade from previous versions.

[Point of VIEW] Did You Try the Power Button? Of Course I Tried the Power Button!

Figure 5.3 NI LabVIEW workbench

Compared to other development environments, one benefit that LabVIEW provide is the extensive support for accessing hardware. Hardware drivers and abstraction layers for many different types of instruments and devices are built in LabVIEW. The abstraction layers usually provide standard software interfaces to communicate with hardware devices. With the help of this powerful software, even people with limited coding experience can write applications and test solutions in a relatively short time period when compared to other more conventional software.

In our search work, we are using LEGO NXT robot for experiments. This hardware is also included in LabVIEW which means LabVIEW will manage the driver and abstraction layer for us. It then became quick simple and time-saving to develop our application in the graphic interfaces provided by LabVIEW. The LabVIEW operation interface for our application is shown in Figure 5.4.

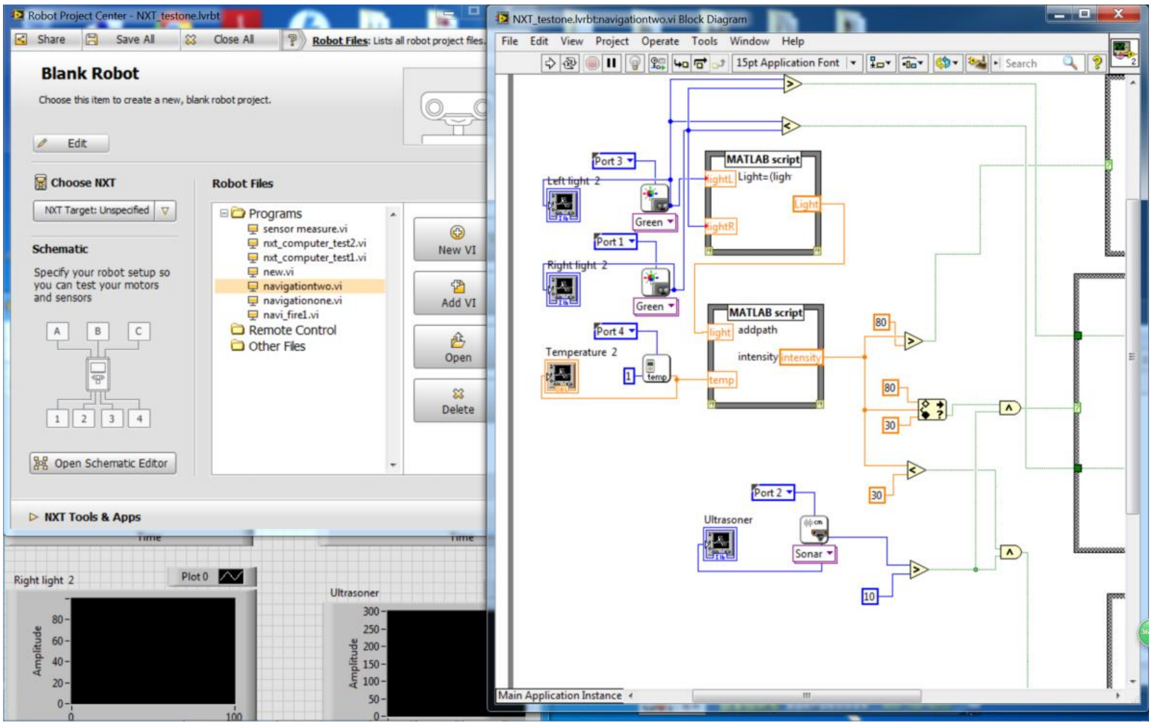


Figure 5.4 Operation interface for NI LabVIEW

5.3 FDS™ and PYROSIM™

Fire Dynamics Simulator (FDS) is a large scale computational fluid dynamics (CFD) model of fire-driven fluid flow which is powered by Google. The Fire Dynamics Simulator solves equations which describe the evolution of fire. The FDS was written by FORTRAN™ and it works based on text file. The FDS obtain input parameters from a text file, calculated a numerical solution to the governing equations, and writes user-specified output data to a separated file. Smokeview™ is an additional program comes with the FDS which can read FDS output text files and show the results graphically. [29] An example of Smokeview is shown in Figure 5.5

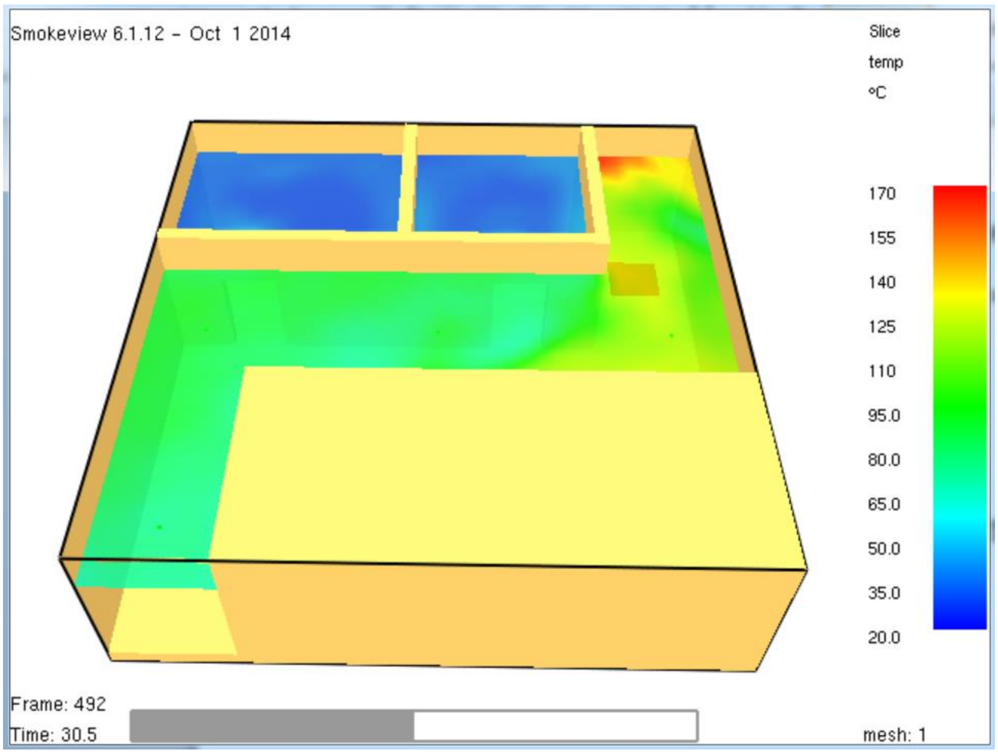
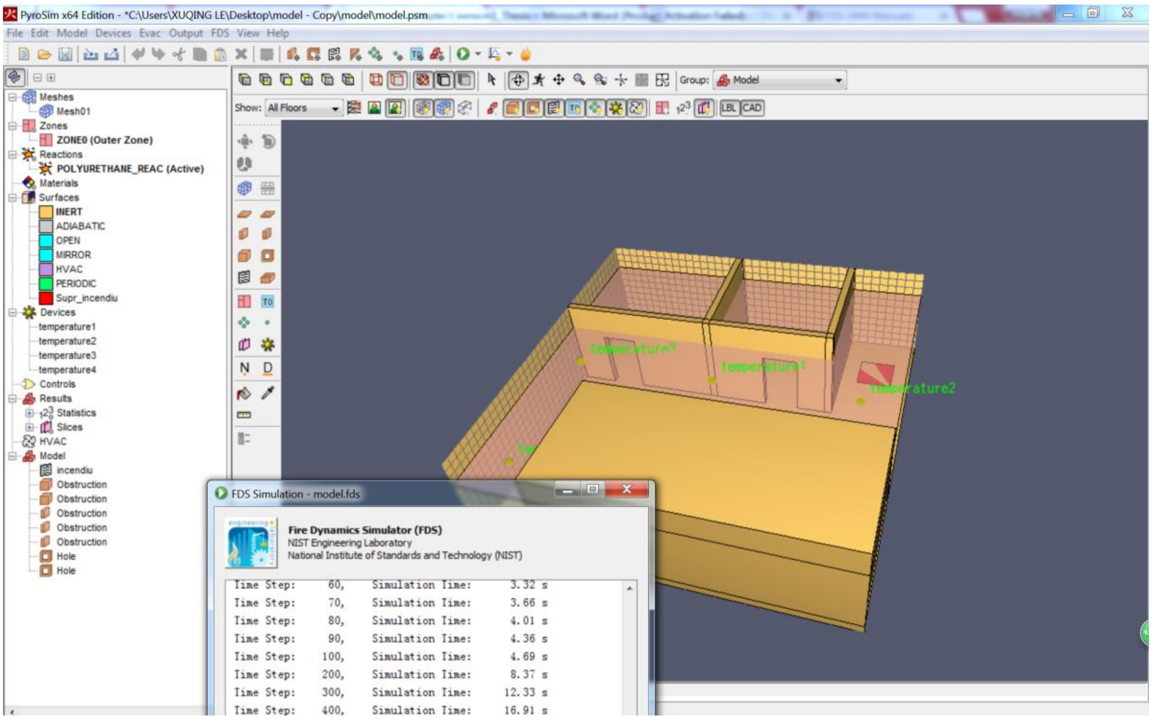


Figure 5.5 an example of Smokeview

PyroSim is a user graphical interface designed for Fire Dynamics Simulator (FDS). Instead of typing tedious FDS inputs variables in text file, PyroSim allows user to quickly create and manage complicated fire models by dragging objects in its user interface. Figure 5.6 shows the user interface of PyroSim



```
Time Step:     60,    Simulation Time:     3.32 s
Time Step:     70,    Simulation Time:     3.66 s
Time Step:     80,    Simulation Time:     4.01 s
Time Step:     90,    Simulation Time:     4.36 s
Time Step:    100,    Simulation Time:     4.69 s
Time Step:    200,    Simulation Time:     8.37 s
Time Step:    300,    Simulation Time:    12.33 s
Time Step:    400,    Simulation Time:    16.91 s
```

Figure 5.6 User interface of PyroSim

PyroSim was used for our research study for complicated fire simulation. We used PyroSim to create a new corridor floor plan first. Fire with different intensities was then setup and simulated in the corridor. The Evolution of each fire has been recorded. The output data we obtained from the PyroSim fire simulation are utilized to build our type-2 fuzzy logic system later.

Chapter6

Simulation Result

Our research is to investigate the performance of type-2 fuzzy logic system used for fire detection robot. This chapter concentrates on the simulation works we have done with MATLAB and PyroSim in different situations. The features of the software we used were presented in Chapter4. MATLAB Simulations were utilized to analyze the feasibility of the algorithms we proposed in different situations so that we can process our experiments in the next step based on these algorithms.

In our research work, all of our system performances testing simulations were carried out in the MATLAB environment. Since the programming codes are long and complicated, we will use the flowcharts to visually and briefly illustrate the methodology of our program. Flowcharts of simulations in Chapter 6 and experiments in Chapter 7 are shown below. All the programs codes of the simulations will be given in appendix of this thesis.

Figure 6.1 shows the flowchart of the main function of the simulation. Figure 6.2 shows the calling sequence of sub functions in main function.

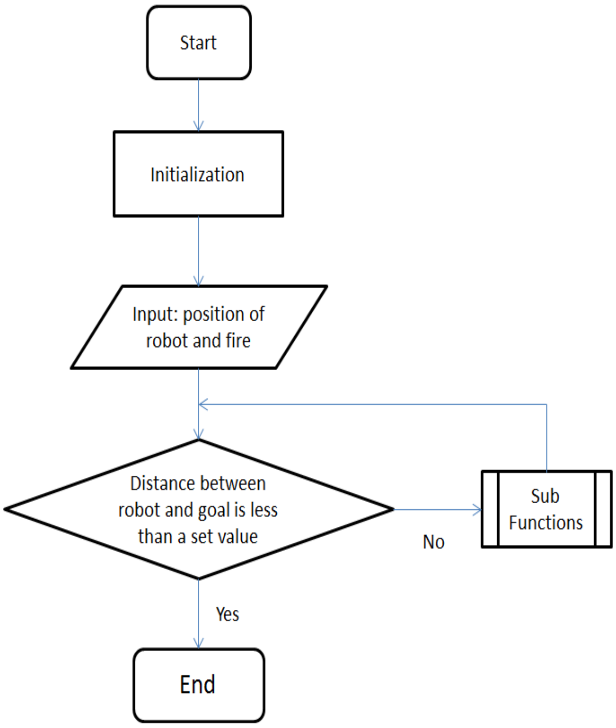


Figure 6.1 Flowchart of main function

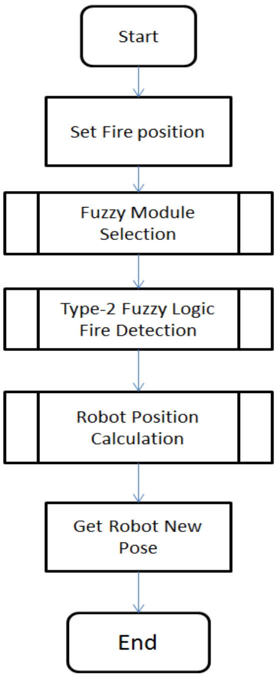


Figure 6.2 Sub functions

Figure 6.3 – 6.5 give the flowcharts of three sub functions:

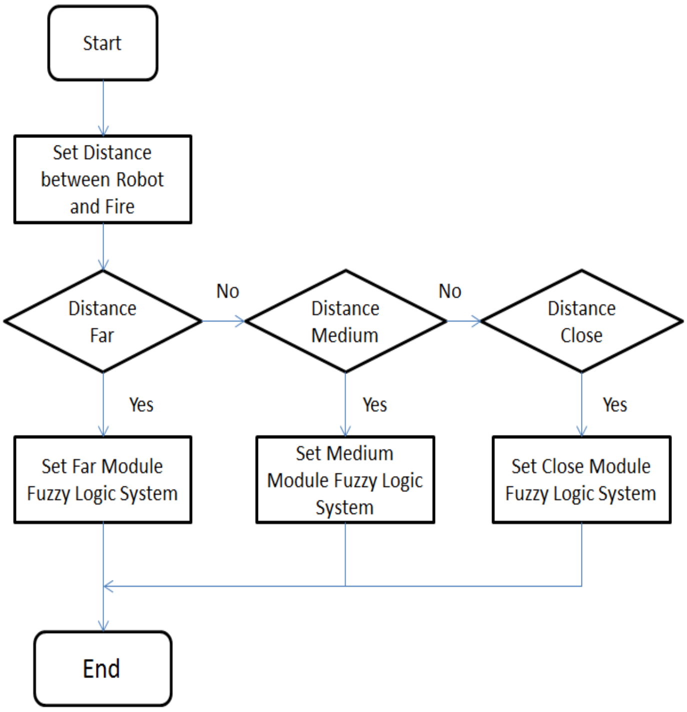


Figure 6.3 Fuzzy module selection sub function

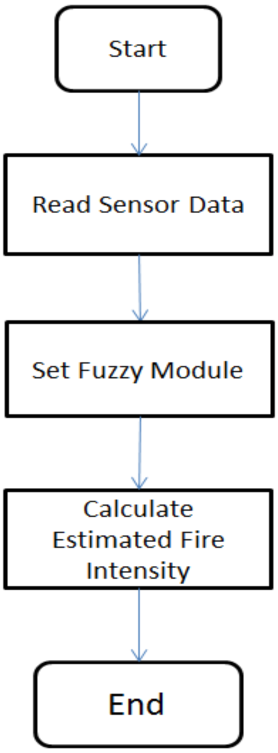


Figure 4: fuzzy logic system sub function

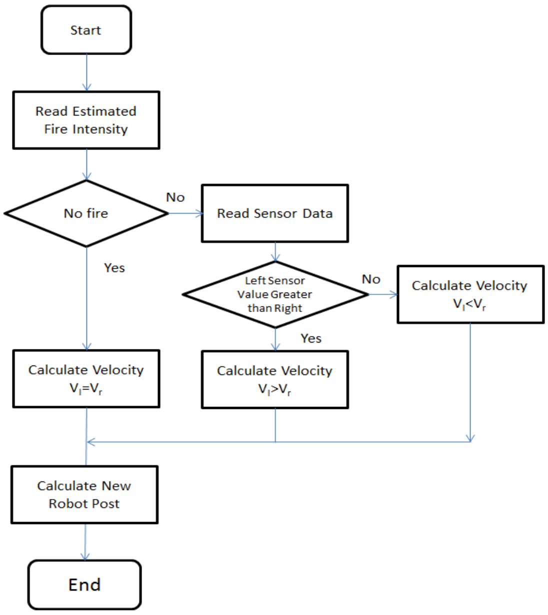


Figure 5: Robot new post calculation

6.1 Fire Simulation

6.1.1 Fire Simulation with MATLAB

MATLAB is used for fire simulation in this section. Fire is considered as a power source with high temperature value and light value in its center area. The diffusion of both temperature and light for a fire were illustrated based on inverse square law. A fire was modeled in MATLAB using these algorithms:

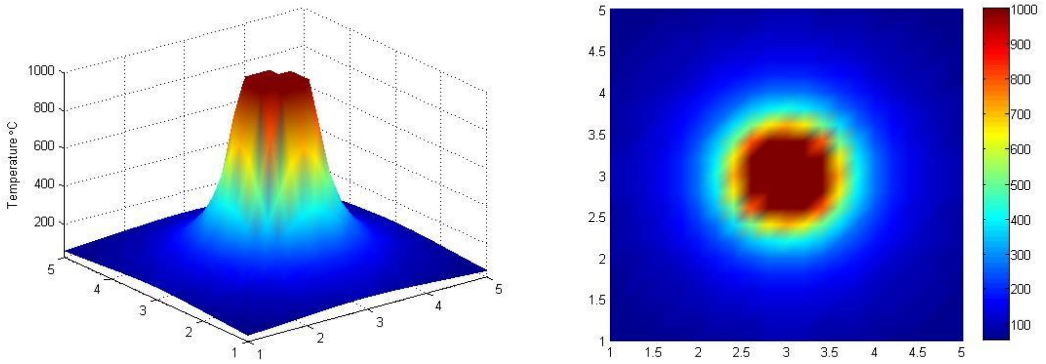
$$T=\begin{cases}\frac{r_c{}^2}{d^2}*T_c & d>r_c\\ T_c & d\le r_c\end{cases} \tag{6.1}$$

Where r_c represents the radius for center area of a fire, Tc represents the temperature in the center area of a fire. d represents the distance between the measuring point and fire.

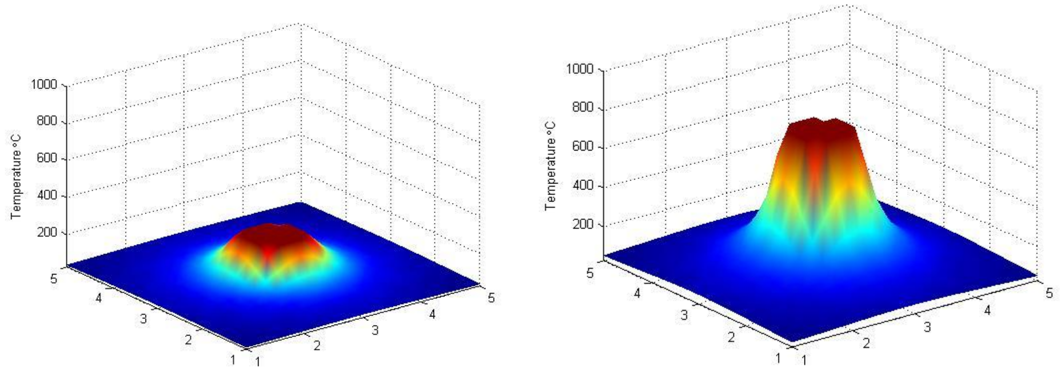
$$L = \begin{cases} \dfrac{r_c^{\,2}}{d^2} * L_c & d > r_c \\ L_c & d \le r_c \end{cases} \tag{6.2}$$

Where r_c represents the radius for center area of a fire, L_c represents the light value in the center area of a fire, d represents the distance between the measuring point and fire.

Three different fires, namely large fire, medium fire and small fire were simulated in MATLAB. Large fire was assumed with Tc = 1000℃ Lc=2000lumen, medium fire was simulated with Tc = 600℃ Lc=2000lumen, whereas a small fire has Tc = 200℃ Lc=400lumen as parameters. Consider the no fire situation, the ambient temperature designed for the simulation is 20℃ and the light intensity is 50lumen. Figure 6 and Figure 7 show the temperature and light diffusion regarding to various fires.

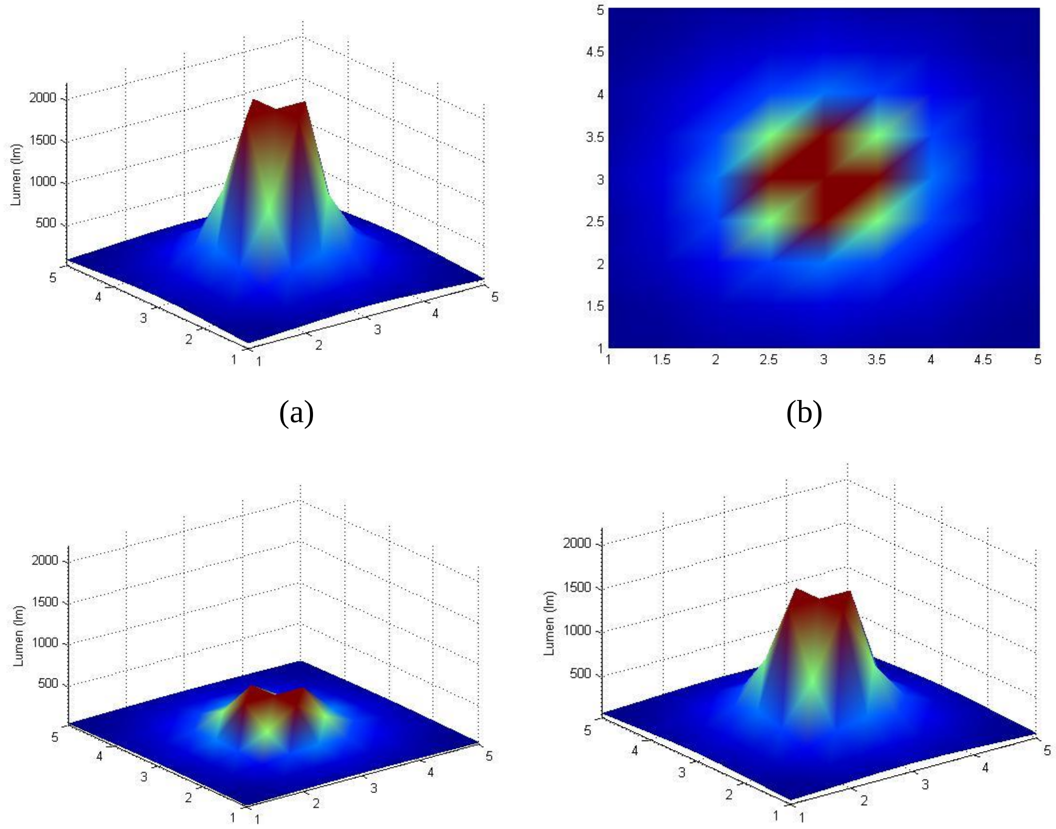


(a) (b)



(c) (d)

Figure 6.6(a) Temperature diffusion of large fire; (b) Top view of large fire temperature diffusion; (c) Small fire temperature diffusion; (d) Medium fire temperature diffusion



(a) (b)

(c) (d)

Figure 6.7(a) Light diffusion of large fire; 7(b) Top view of large fire light diffusion; 7(c) Small fire light diffusion; 7(d) Medium fire light diffusion

The fire simulation data is gathered and stored and then used to build type-2 fuzzy system for fire detection robot. This type-2 fuzzy logic is then tested in a free space scenario to investigate the feasibility of our proposed approach.

6.1.2 Fire Simulation with PyroSim

To test our proposed algorithm in more relative real world situation, a more powerful fire simulation tool is needed for more complex fire model simulation. The fire simulation work for a new floor plan has been done in this section using PyroSim. A new floor plan with two rooms and a corridor was designed with PyroSim modeling for our further simulation work. A fire was then set up and simulated at the corner of the floor which is represented by red rectangular. Figure 8(a) - 8(b) show the front view and top view structure of the designed floor.

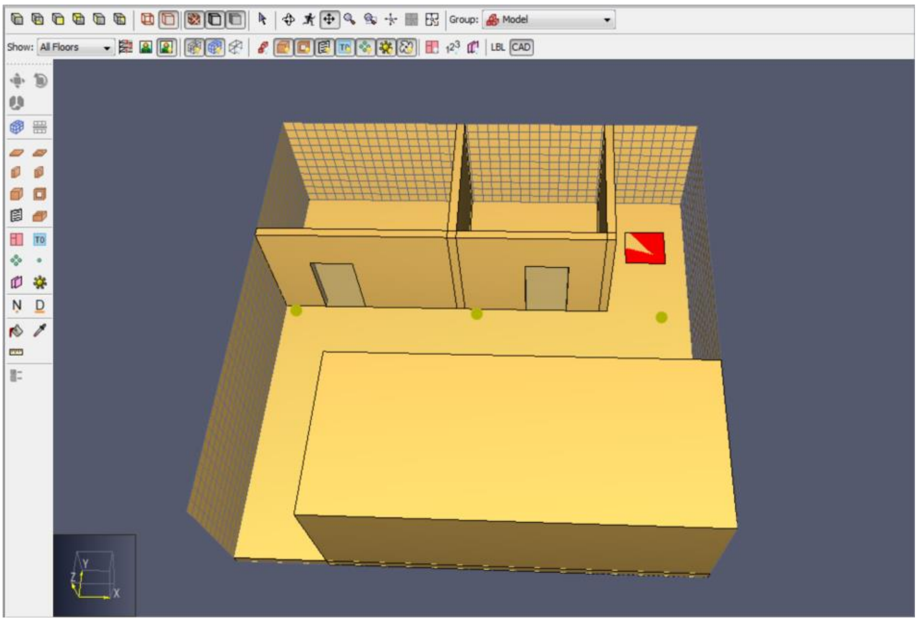


Figure 6.8(a) Front view of 3-D structure diagram

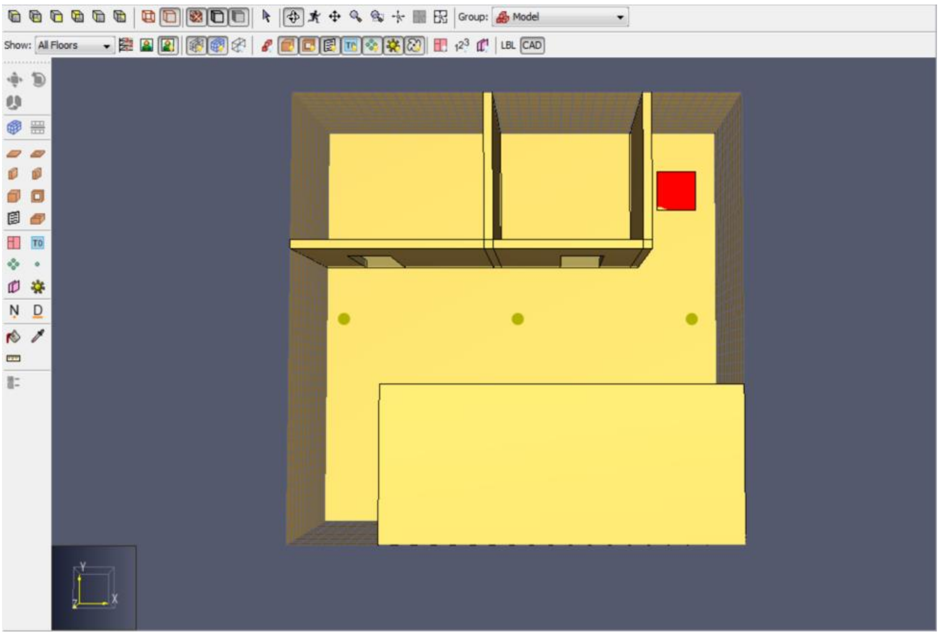


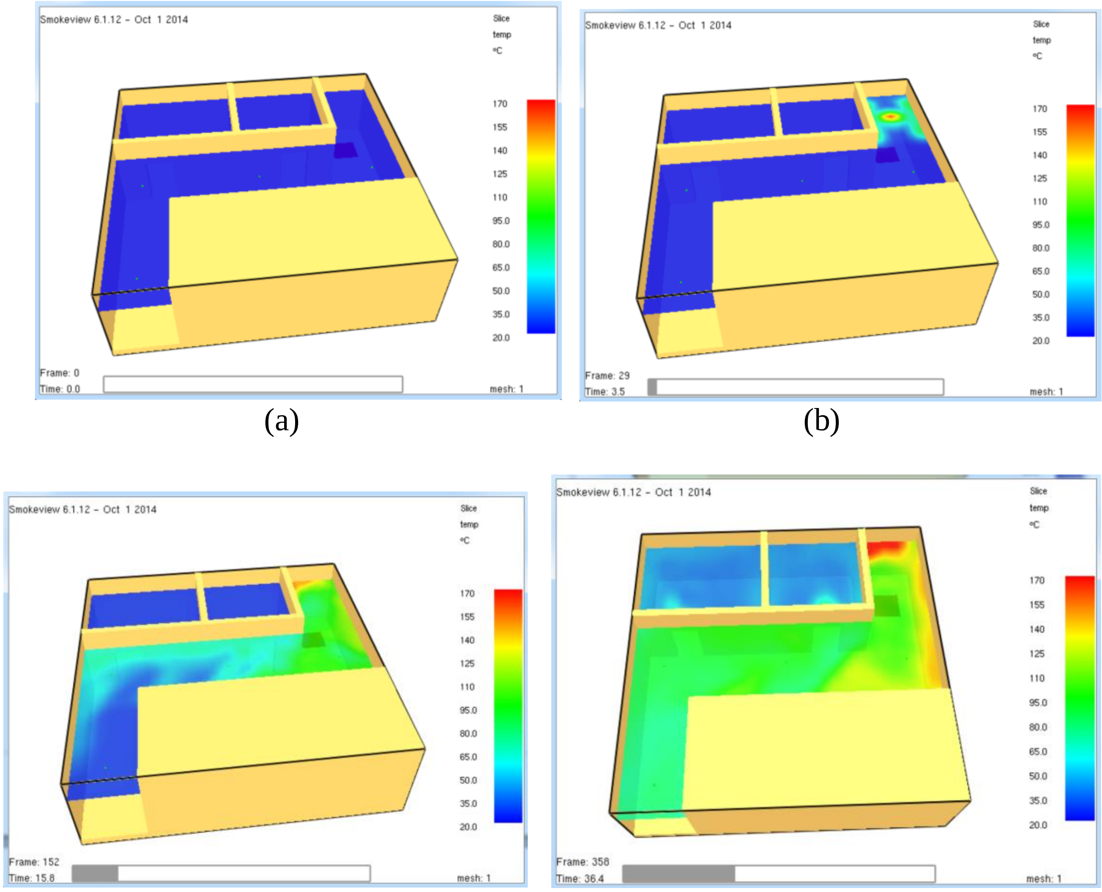
Figure 6.8(b) Top view of 3-D structure diagram

Three different fires were simulated respectively at the corner of the floor, namely a large fire with 400KW/m^2 heat release rate per area (HRRPUA), a medium fire with 250KW/m^2 HRRPUA and last a small fire with 150KW/m^2 HRRPUA. The diffusion process of temperature and light regarding to a simulated fire are shown in Figure 9(a) – Figure 9(f) below. Data gathered from the fire simulation are then used to build type-2 fuzzy logic system later. The most important feathers regarding to the fire simulation are listed as:

- The Boundary for the floor is 15m*15m*3m
- The Size of burner is 1m*1m,
- Ambient Temperature: 20℃
- Ambient Light : 40lm
- Ambient pressure: 1.01325E5 Pa
- Ambient Oxygen Mass Fraction: 0.232378 kg/kg

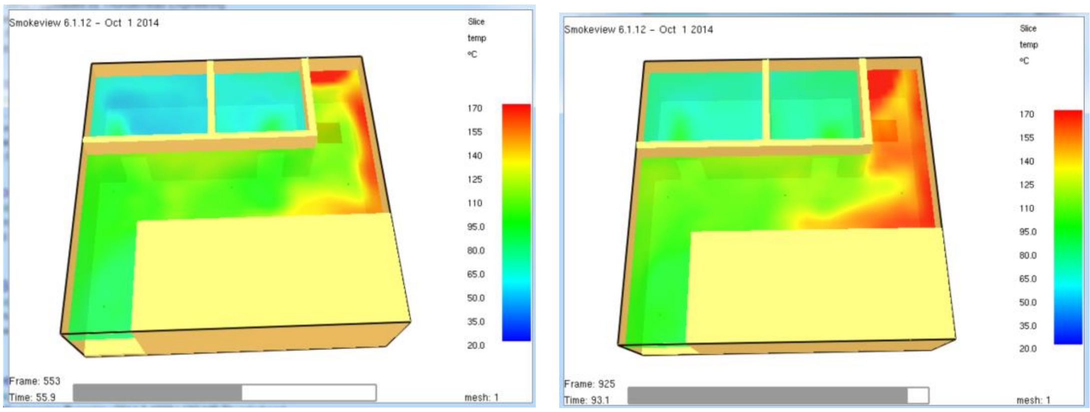
- Ambient Carbon Dioxide Mass Fraction: 5.95E-4 kg/kg
- Relative Humidity: 40%
- The default wall boundary condition representing a smooth wall with fixed temperature, TMPA, and emissivity 0.9.
- Burner has 400KW/m^2 HRRPUA for a large fire, 250 KW/m^2 for a medium fire and 150 KW/m^2 for a small fire.

The diffusion process of temperature regarding to a simulated fire are shown in Figure 9(a) – 9(f) below.



(a) (b)

(c) (d)



(e) (f)

Figure 6.9(a) - (f): Diffusion of temperature with respect to simulated fire

The fire simulation data obtained from PyroSim was used for expert training and building type-2 fuzzy system for fire detection robot. This type-2 fuzzy logic is then tested in a new floor scenario to investigate the feasibility of our proposed approach.

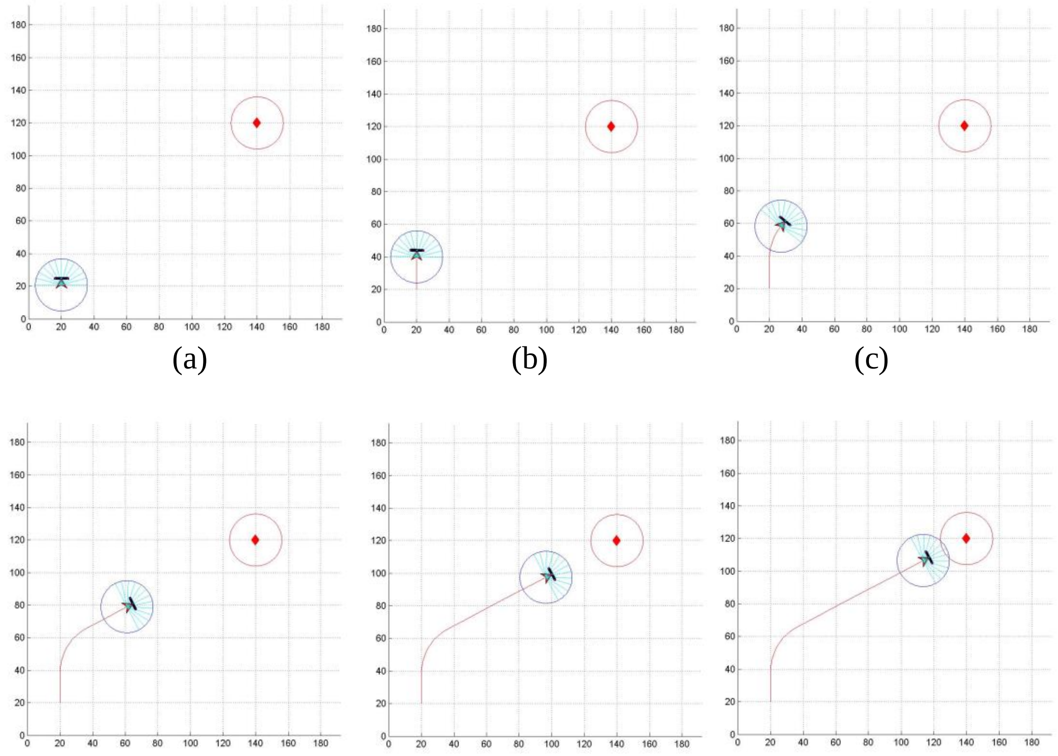
6.2 Simulation for Free Space Scenario

In this section, the performance of type-2 fuzzy logic system designed for fire detection robot regarding to free space scenario was investigated in MATLAB Simulation. The robot was designed to approach fires of various intensities including large fire, medium fire, small fire and no fire. The simulation results with respect to robot moving towards a large fire and a heat bar will be presented in this section.

Figure 10(a) – (f) show the process of robot approaching a large fire source with the following symbols:

- The red diamond represents the large fire.
- The circle represents the center area of fire.
- The arrow represents the current direction of robot.

- The circle with radials represents the detection range
- The red line represents the trajectory of robot.

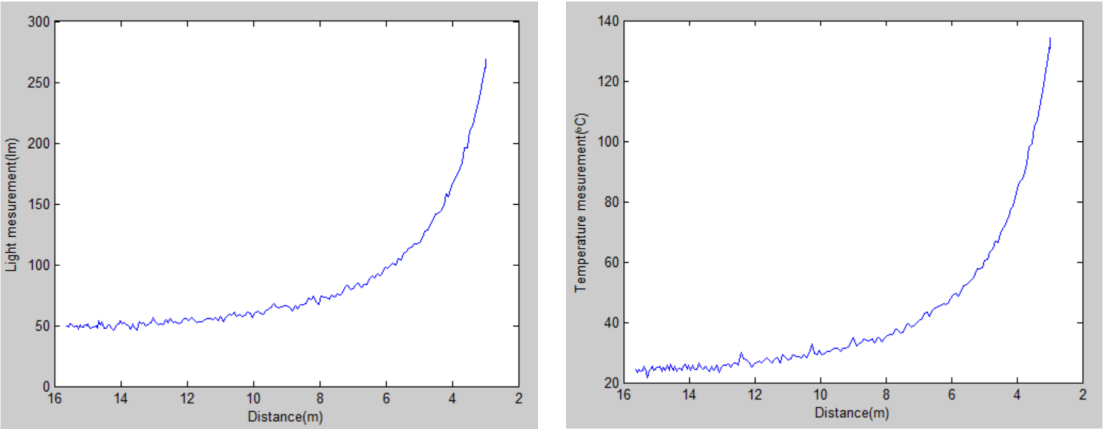


(a) (b) (c)

(d) (e) (f)

Figure 10(a) – (f): Process of robot approaching large fire

Figure 6.10(g) and Figure 10(h) gives the information about real time simulated sensor measurements with respect to both temperature and light including Gaussian white noise.



(g) (h)

Figure 6.10(g) Real time simulated light measurements; (h) Real time simulated temperature measurements

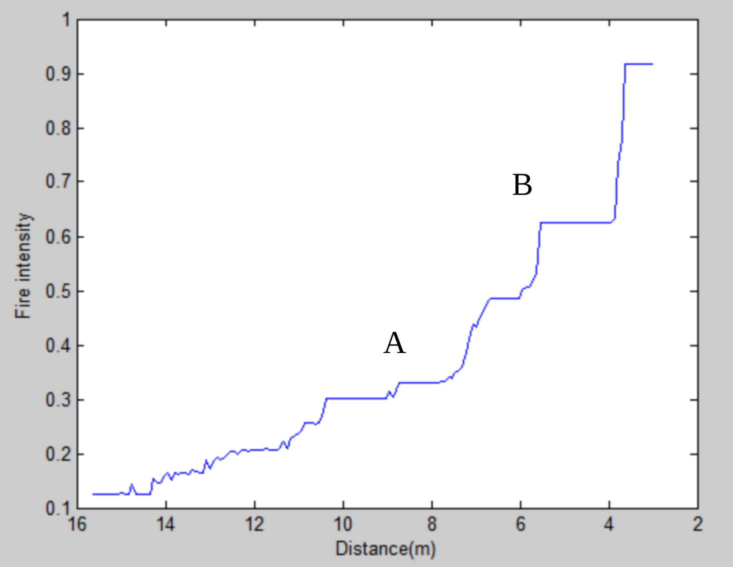
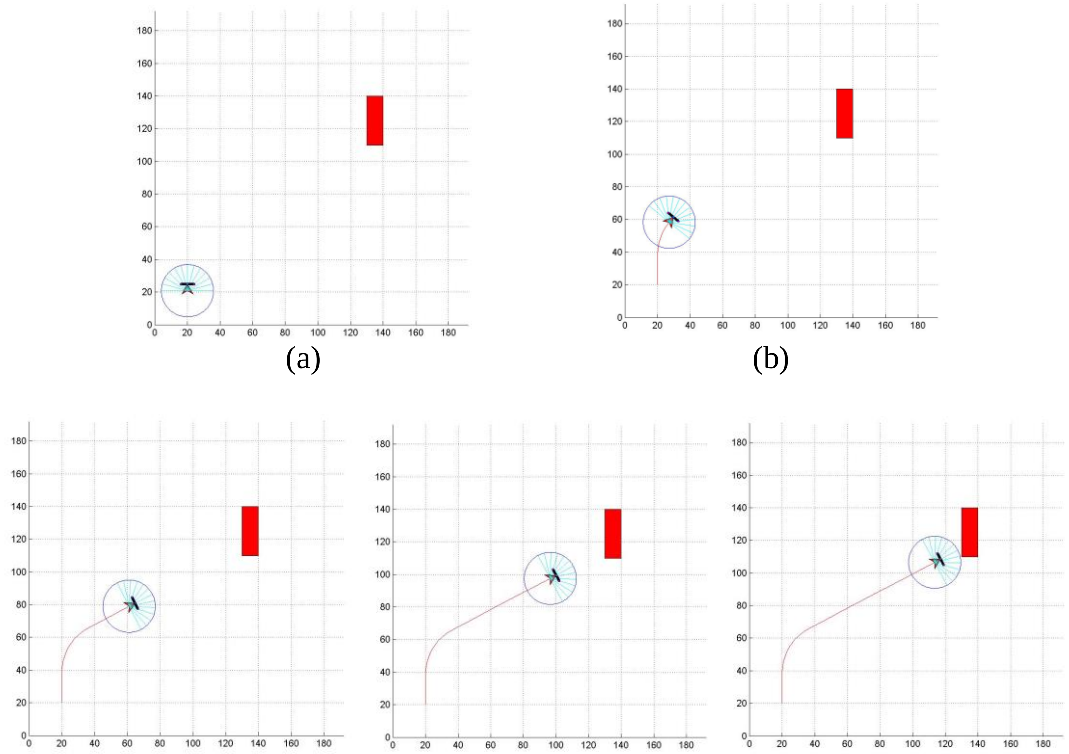


Figure 6.10(i): Result of Estimated fire intensity

Figure 6.10(i) shows the changing of estimated fire intensity when robot approached the fire source. Three different fuzzy modules were activated in order based on the distance between robot and fire. Far-module was activated first and gave a quick fire intensity evaluation. Fire intensity was re-estimated start at A because medium fuzzy module was activated. Finally the close-module fuzzy logic was triggered at B and gave a final estimation of fire intensity. In this case, a large fire at far distance was estimated at beginning by far-module and with robot approaching the source, a final decision was made by close model that there was a large fire at close distance.

Figure 11(a) – (e) show the process of robot approaching a heat bar which represents no fire situation.

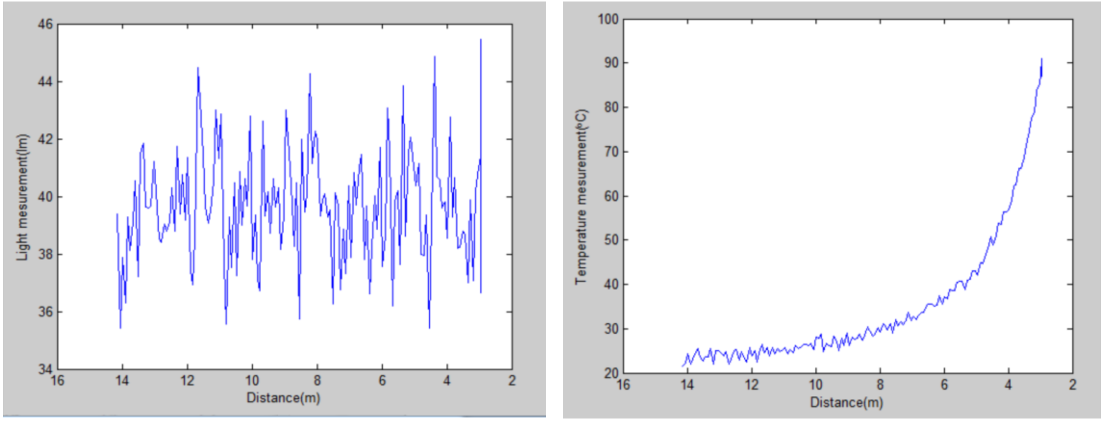


(a) (b)

(c) (d) (e)

Figure 6.11(a) – (e): Process of robot approaching a heat bar

Figure 11(f) and Figure 11(g) gives the information about real time simulated sensor measurements with respect to both temperature and light including Gaussian white noise.



11(f) 11(g)

Figure 6.11(f) Real time simulated light measurements; (g) Real time simulated temperature measurements

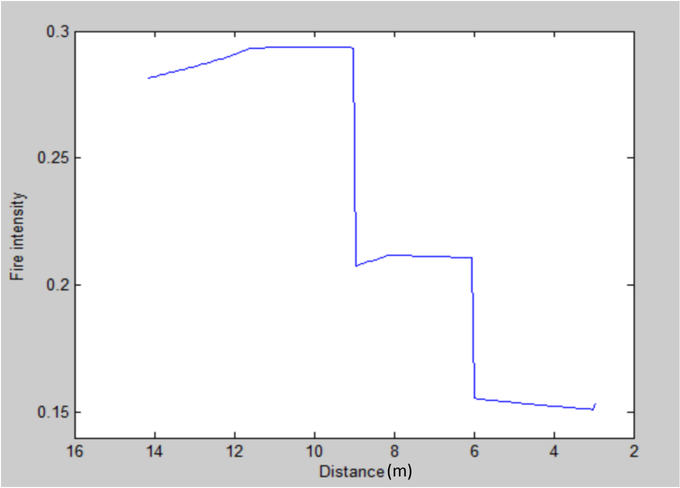


Figure 6.11(i) Result of Estimated fire intensity

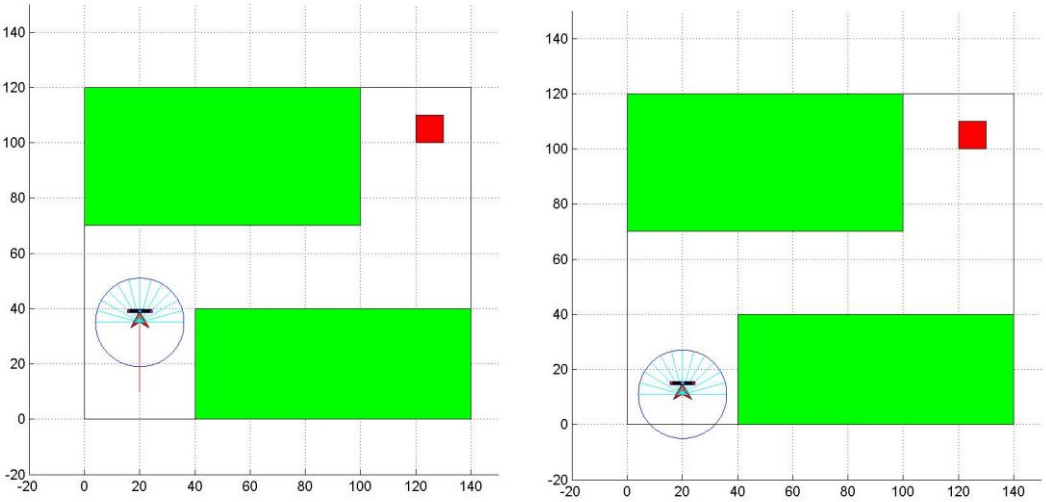
In this case, a small fire at far distance was estimated at beginning by far-module due to abnormal temperature had been detected, and with robot approaching the source, it found that there was less possibility for a fire. A final decision was made by close model that there was no fire.

The results presented in this section show that the proposed type-2 fuzzy logic system could be used reliably and efficiently for fire detection robot.

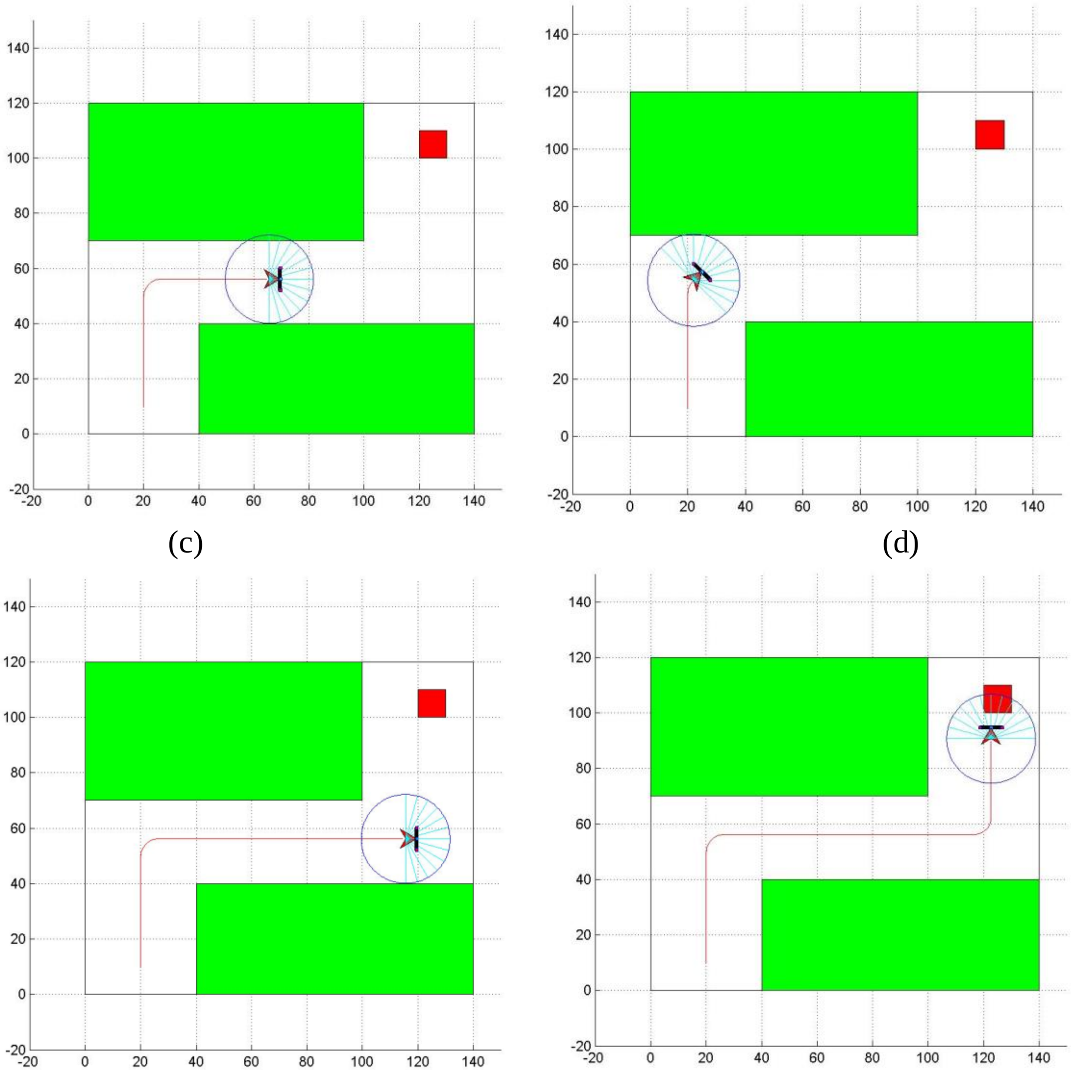
6.3 Simulation for New Floor Plan Scenario

In this section, the new floor plan we designed in PyroSim was modeled in MATLAB. Type-2 fuzzy logic fire detection and estimation system was built based on the temperature and light data obtained from fire simulation using PyroSim. This type-2 fuzzy logic system is then tested using MATLAB Simulation. The situations of robot searching large fire, medium fire and small fire were investigated. The result regarding to robot searching a large fire is presented in this section.

Figure 12(a) – 12(f) show the process of robot approaching the fire source.



(a) (b)

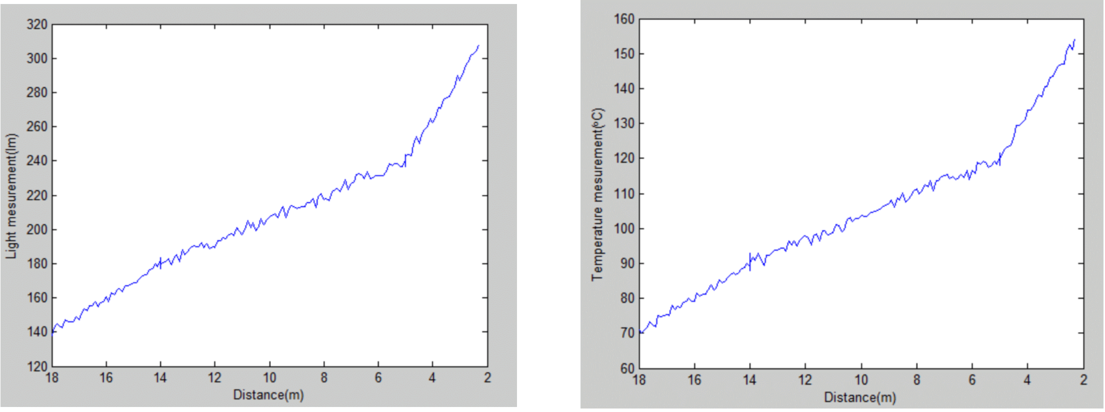


(c) (d)

(e) (f)

Figure 6.12(a) – (f) Robot approach fire at corner

Figure 12(g) and Figure 12(h) gives the information about real time simulated sensor measurements when a large fire was set up.



(g) (h)

Figure 6.12(g) Real time simulated light measurements; (h) Real time simulated temperature measurements

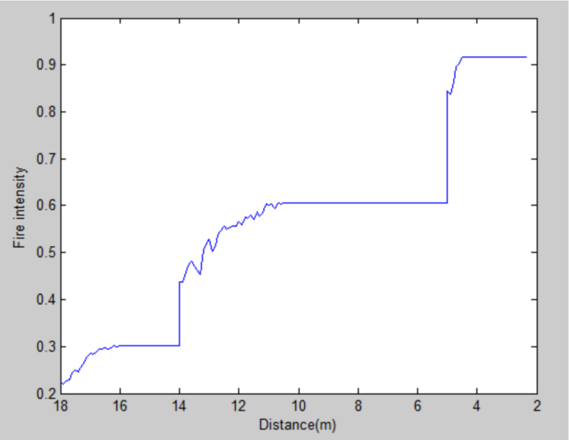


Figure 6.12(i) Result of fire intensity estimation for a large fire

The result of estimated fire intensity is shown in Figure 12. The final fire intensity result produced by type-2 fuzzy system is large fire at far distance, large fire at medium distance and large fire at close distance

A medium fire and a small fire were also simulated in MATLAB respectively. The performance of our proposed algorithm is shown in Figure 13 – Figure 14 and Table 1 – Table 2.

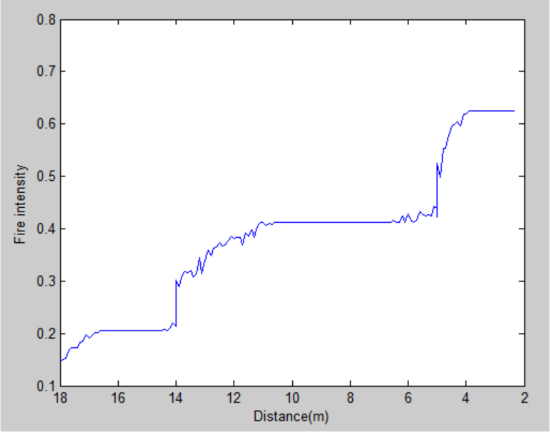


Figure 6.13 Result of fire intensity estimation for a medium fire

Table 6.1: Fire intensity result for medium fire

Module	Far Module	Medium Module	Close Module
Results	Medium Fire	Medium Fire	Medium Fire

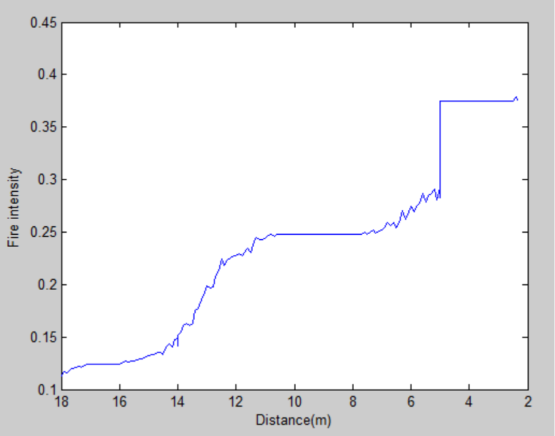


Figure 6.14 Result of fire intensity estimation for a small fire

Table 6.2: Fire intensity result for small fire

Module	Far Module	Medium Module	Close Module
Results	Small Fire	Small Fire	Small Fire

The results presented in this section show that our proposed algorithm is able to give the correct fire intensity estimation regarding to fires of various intensities. The proposed approach, which used the proposed type-2 fuzzy logic system, avoids the need for prior expertise to build a fuzzy logic system by direct problem simulation which will help the experts to design a more reliable fuzzy logic system to solve problems in unknown areas.

6.4 Comparison of Type-1 and Type-2 Fuzzy Logic

6.4.1 Analysis of the confidence in the results of type-1 and type-2 Fuzzy Logic

A Comparison of Type-1 fuzzy logic system (T1 FLS) and Type-2 fuzzy logic system (T2 FLS) are discussed in this section. One type-2 fuzzy logic system and three type-1 fuzzy logic systems are simulated and compared with same simulation environment. Three T1 FLSs, namely Fuzzy Logic System A, Fuzzy Logic System B and Fuzzy Logic System C are developed individually by expert A, expert B and expert C. T2 Fuzzy Logic system is consisted of all three T1 FLSs that is developed by three experts together. In order to focus on the difference of the performance between T1 FLS and T2 FLS, exactly the same rule bases are implemented for all four Fuzzy Logic Systems.

The goal of the proposed fuzzy logic system is to output estimated fire intensities when robot approaches the fire source. In this section, the performance of all four different fuzzy logic systems are tested and compared through nine different simulation scenarios. They are: a large fire setting up at far distance, a large fire at medium distance, a large fire at close distance, a medium fire at far distance, a medium fire at medium distance, a medium fire at close distance, a small fire at far distance, a small fire at medium distance and a small fire at close distance. Table3 below shows the results of the simulation.

	Large Fire		**Medium Fire**		**Small Fire**	
	Distance	Result	Distance	Result	Distance	Result
Type2 Fuzzy Logic System	Far	Correct	Far	Correct	Far	Correct
	Medium	Correct	Medium	Correct	Medium	Correct
	Close	Correct	Close	Correct	Close	Correct
Type1	Far	Correct	Far	Wrong	Far	Correct

Fuzzy Logic System A	Medium	Correct	Medium	Wrong	Medium	Correct
	Close	Correct	Close	Wrong	Close	Correct
Type1 Fuzzy Logic System B	Far	Correct	Far	Correct	Far	Correct
	Medium	Correct	Medium	Correct	Medium	Correct
	Close	Correct	Close	Correct	Close	Correct
Type1 Fuzzy Logic System C	Far	Wrong	Far	Wrong	Far	Correct
	Medium	Correct	Medium	Correct	Medium	Correct
	Close	Correct	Close	Correct	Close	Correct

Table 6.3: Results of the fuzzy logic system simulation

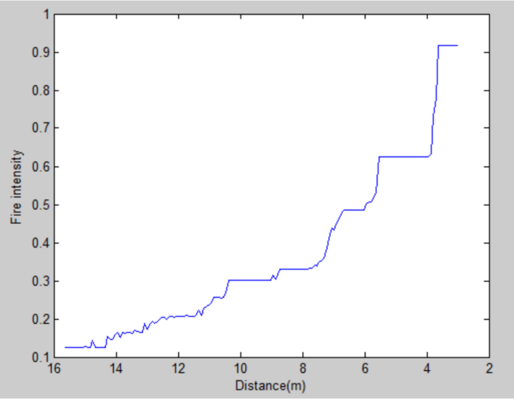
As shown in table above, the results from Type-2 Fuzzy Logic System are relevant. The system was able to generate correct fire intensity in all nine simulation scenarios. Also, the output curves are smooth which is ideal for developing robot’s navigation system. Although type-1 FLS B results in all 9 correct answers, there are some wrong results in both FLS A and FLS C. These two FLSs failed in some circumstances because of inefficient membership function choice by expert A and expert C. These inefficient membership functions have full weight in its corresponding system. Using the same membership function as MF component, Type-2 fuzzy logic system is able to handle these inefficient choices and detect the correct fire intensity by reducing the weight of each membership function. Opinions from multiple experts were considered when building the membership functions. Results in table above illustrate that type-2 fuzzy logic system has more reliability than type1 fuzzy logic system.

6.4.2 Comparison of type-1 and type-2 under different noise level

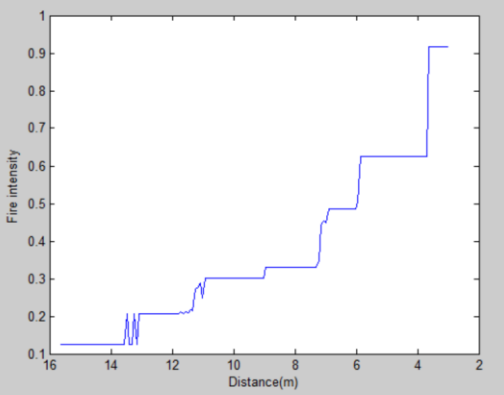
Sensors can obtain inaccuracy results due to numerous reasons. Significant disturbances can exist in robot’s working environment which can result in large noise for sensors. Also inexpensive sensor itself can lead to inaccurate results. A good sensor fusion algorithm should have good features to handle input noise.

The goal of the simulation work presented in this section is to test how type-1 fuzzy logic system and type-2 fuzzy logic system handle the input noise. Gaussian white noise is simulated and used as basic noise for inputs in these simulations. The two sensor fusion algorithms which are chosen to compare are Type-2 FLS and Fuzzy Logic System B were presented in the last section.

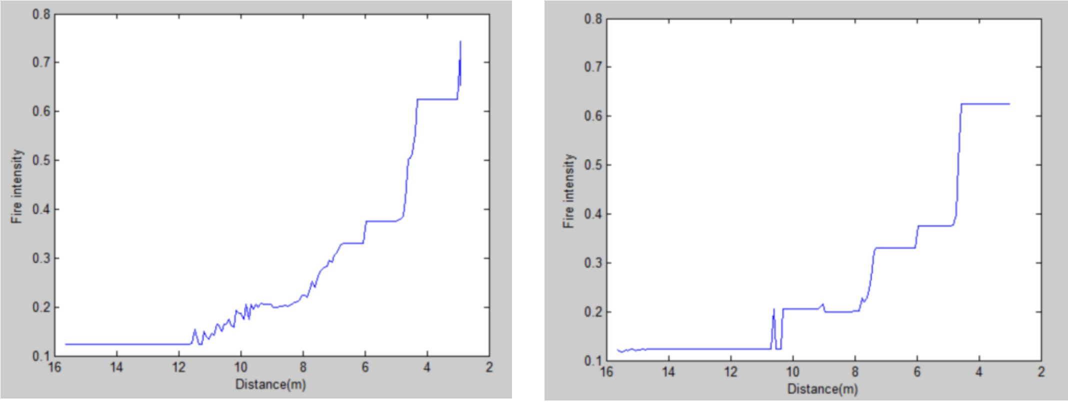
With regular noise level, both T1 FLS and T2 FLS are able to produce the correct answer and their output curves are smooth. Results are shown in Figure 15(a) - (d)



(a)



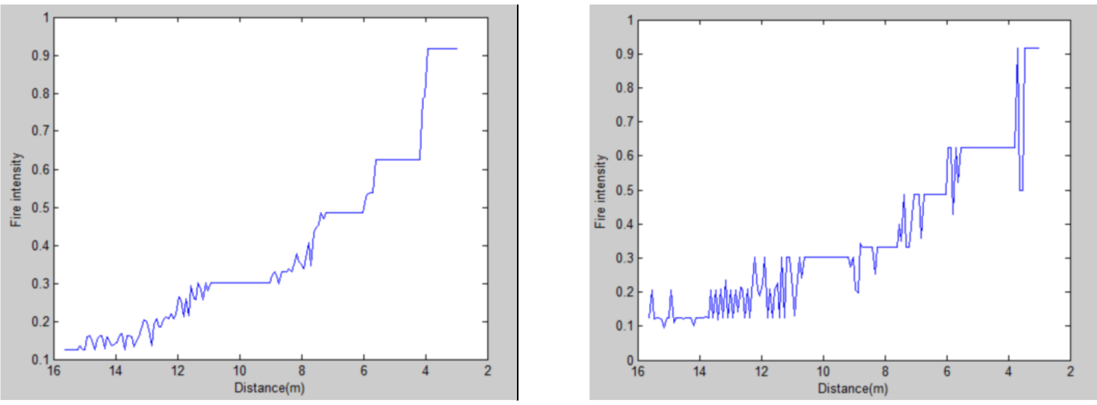
(b)



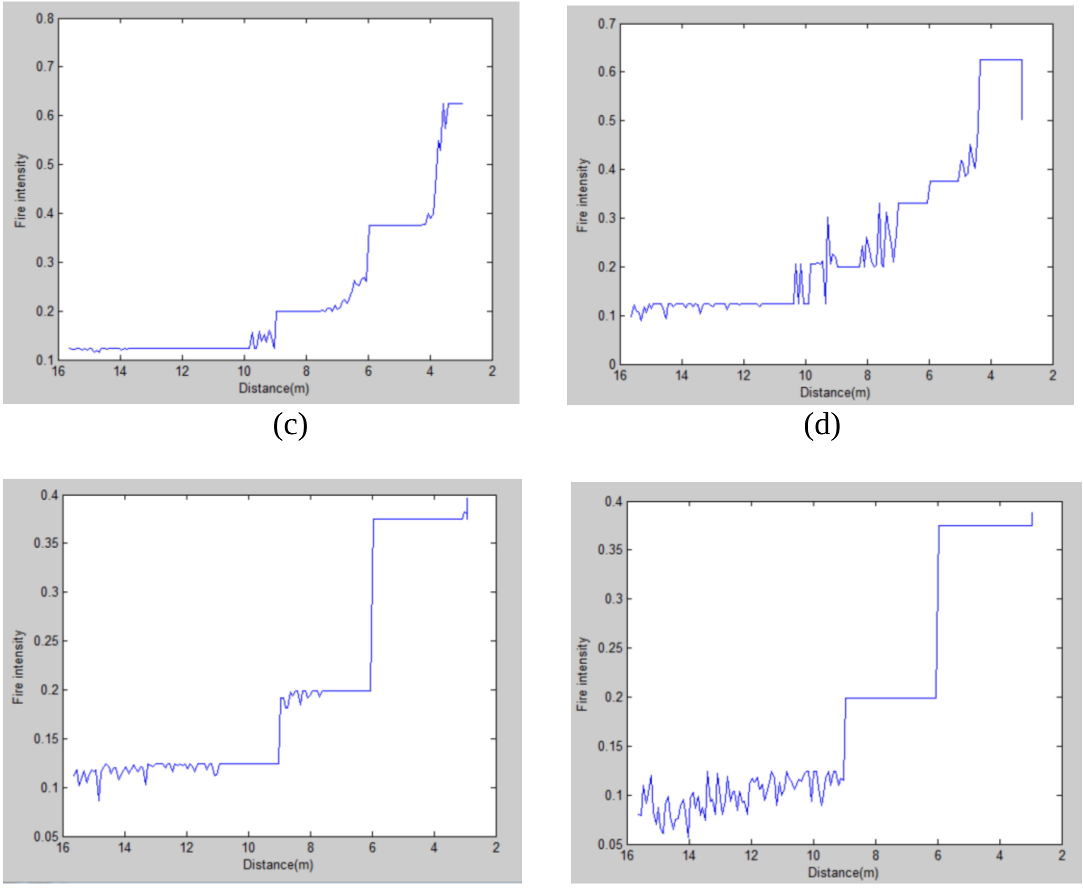
(c) (d)

Figure 6.15(a) Type-2 result for robot searching large fire; (b) Type-1 result for robot searching large fire; (c) Type-2 result for robot searching medium fire; (d) Type-1 result for robot searching medium fire

Figure 16(a) - (f) below shows how T1 FLS and T2 FLS respond when the noise intensity increases. By amplifying the Gaussian noise for amplitude, although T1 FLS is still capable to produce correct result, the system starts to become unstable. There have been significant oscillations in the results generated by T1 FLS, whereas T2 FLS is still able to produce relative stable and clear results for fire intensity estimation.



(a) (b)



(c) (d)

(e) (f)

Figure 6.16(a) Type-2 result for robot searching large fire; (b) Type-1 result for robot searching large fire; (c) Type-2 result: searching medium fire; (d) Type-1 result for robot searching medium fire; (e) Type-2 result for robot searching small fire; (f) Type-2 result for robot searching small fire

The results presented in this section show that type-2 fuzzy system is general outperform type-1 fuzzy system. By comparing the performance with type-1 fuzzy system, it has been found that type-2 fuzzy gives better results when large amount of uncertainty has to be concerned.

Chapter 7
Experimental Result

Experimental work has been carried out to verify the simulation result in an actual physical environment. By analyzing the results from experiments, it can be verified that the proposed type-2 fuzzy logic fire detection algorithm is feasible at both theory and practical. The robot we utilized for the experiment is LEGO NXT robot. The structure, sensors and the designing of the robot were discussed in Chapter 4.

The overall designing of our experimental work is presented in Section 7.1. In section 7.2, the experiments with respect to robot approaching a source placed in sight was performed to test the feasibility and reliability of our proposed fire detection system. In section 7.3, experiments regarding to robot searching a source in a cluster environment was made to reproduce the situation of the new floor plan simulation in Charter 6.

7.1 Experiment Design

Experiments were performed by LEGO NXT robots. Four sensors were implemented in our robot. Two light sensors, a thermal infrared sensor, an ultrasonic sensor are connected to robot input port A, B, C and D respectively.

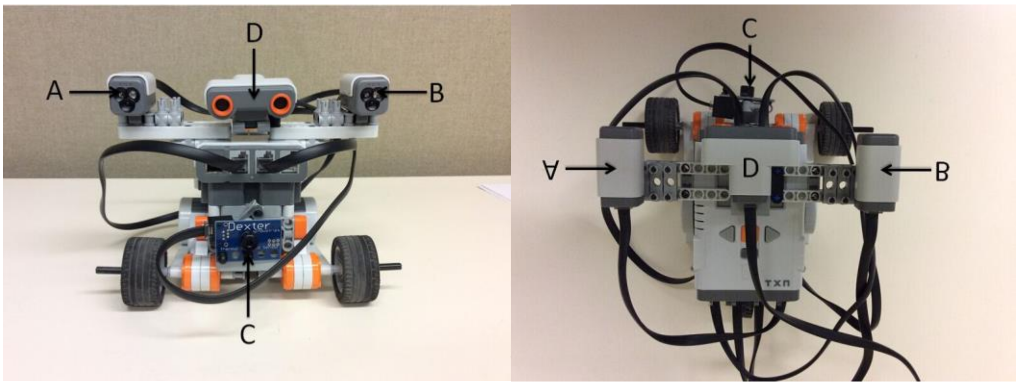


Figure 7.1 Mechanical structure for LEGO NXT

For safety concerns, Thermal Infrared Heat Lamp was used as the fire source for our experiment. The source was 375 Watt, 120-130 Volts with a bulb head and medium base. To test the performance of our proposed type-2 fuzzy logic system for fires of various intensities, fire source with different intensity was assumed and established as follow:

- Lamp turned off stands for No fire
- Lamp turned on stands for large fire
- Lamp turned on with plastic cover in front of it stands for small fire

Figure 7.2 Thermal Infrared Heat Lamp with different condition

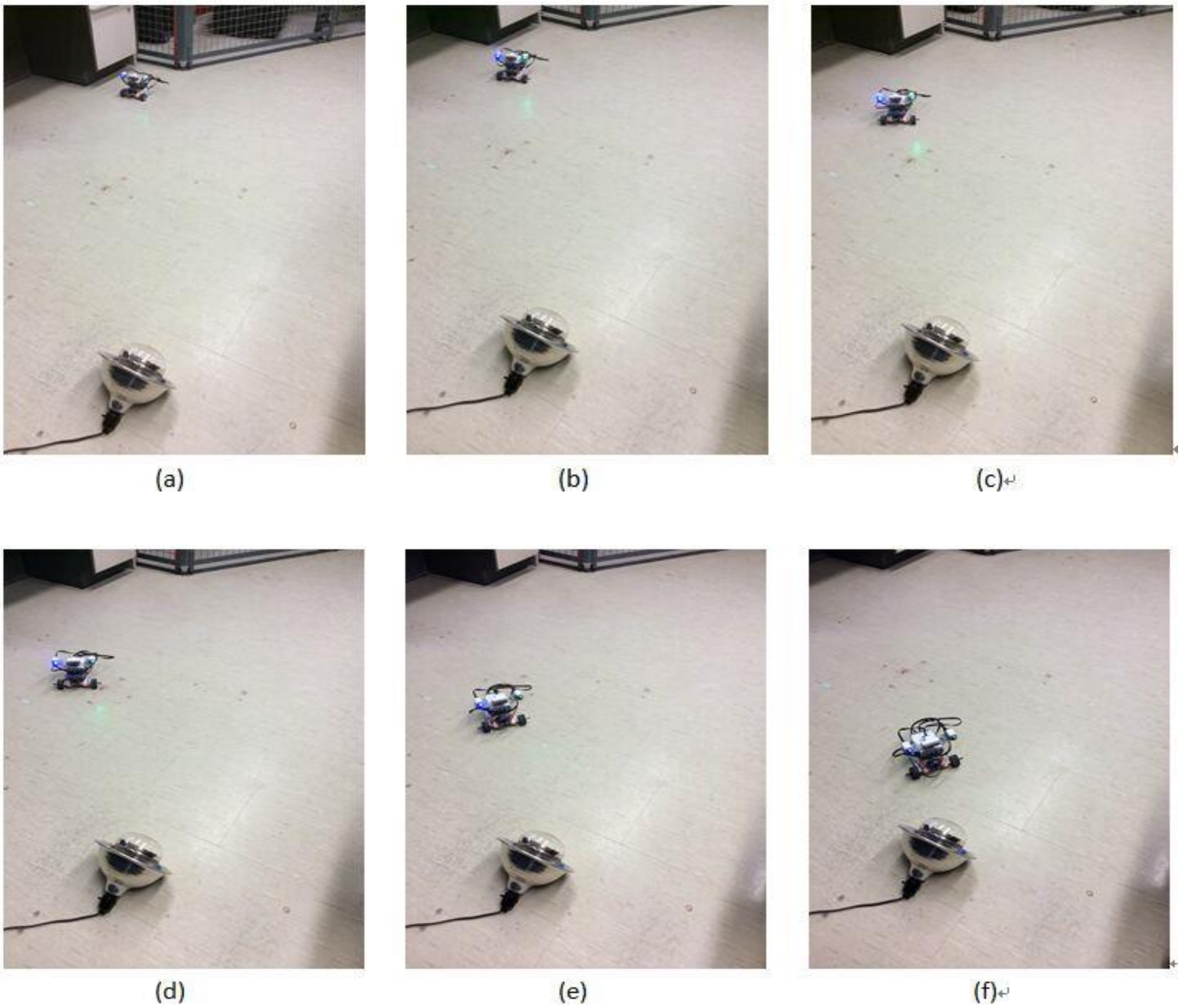
Since the Thermal Inferred Lamp produced different level of temperature and light value compare to a regular fire. Further data measuring and collection are needed before the experiments. The type-2 fuzzy logic system implemented on our fire detection robot was built based on these data. The system was then tested in six independent scenarios as list:

- Source (Large fire) placed in sight but not directly in front of the robot
- Source (No fire) placed in sight but not directly in front of the robot
- Source (Small fire) placed in sight but not directly in front of the robot
- Source (Large fire) placed in complex terrain
- Source (No fire) placed in complex terrain
- Source(Small fire) placed in complex terrain

7.2 Experiments of Fire Detection Robot in Free Space

Scenario 1: Target (No fire) placed in sight but not directly in front of the robot

In this experiment, robot was assigned to search the lamp. Figure 7.3(a) – (f) shows the process of robot approaching target. Figure 7.4(a) – (c) shows the real time light sensor measurement and TIR sensor measurement. We can see from the results that both measured temperature and light keeps on a low level. The fire estimation result is shown in Figure 7.4(d). The type-2 fuzzy logic system produced no fire as final result.



(a) (b) (c)

(d) (e) (f)

Figure 7.3 Robot approaching the target

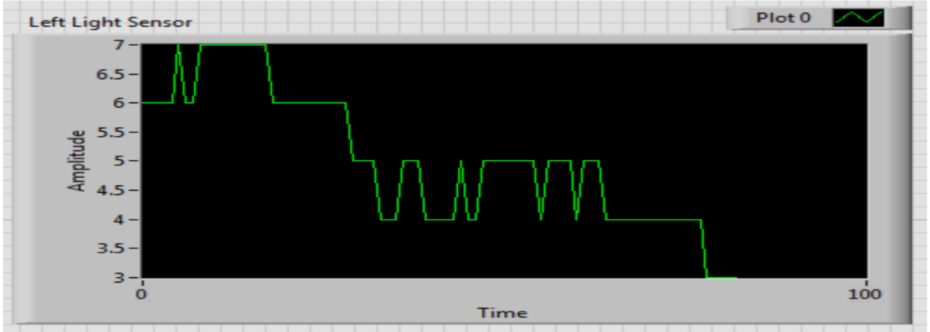


Figure 7.4(a) Real time light measurement

Figure 7.4(b) Real time light measurement

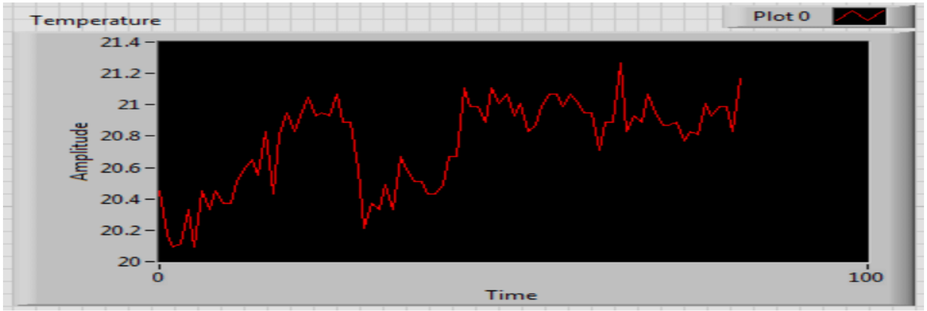


Figure 7.4(c) Real time temperature measurement

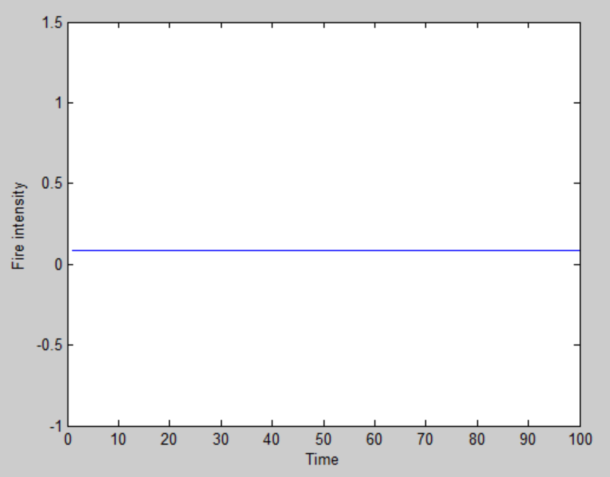


Figure 7.4(d) Real time computation for fire intensity

Scenario 2: Target (Large fire) placed in sight but not directly in front of the robot

In this experiment, robot was assigned to search the assumed large fire. Figure 7.5(a) – (f) below shows the process of robot approaching target. Figure 7.6(a) – (c) shows the real time color sensor measurement and TIR sensor measurement. We can observe from results that light measurment started at 75 and ended at 100 and temperature value started at 20℃ and finally reached 120℃. The type-2 fuzzy logic system produced small fire at far-distance (because of very low temperature value), large fire at medium-distance and large fire at close-distance as result.

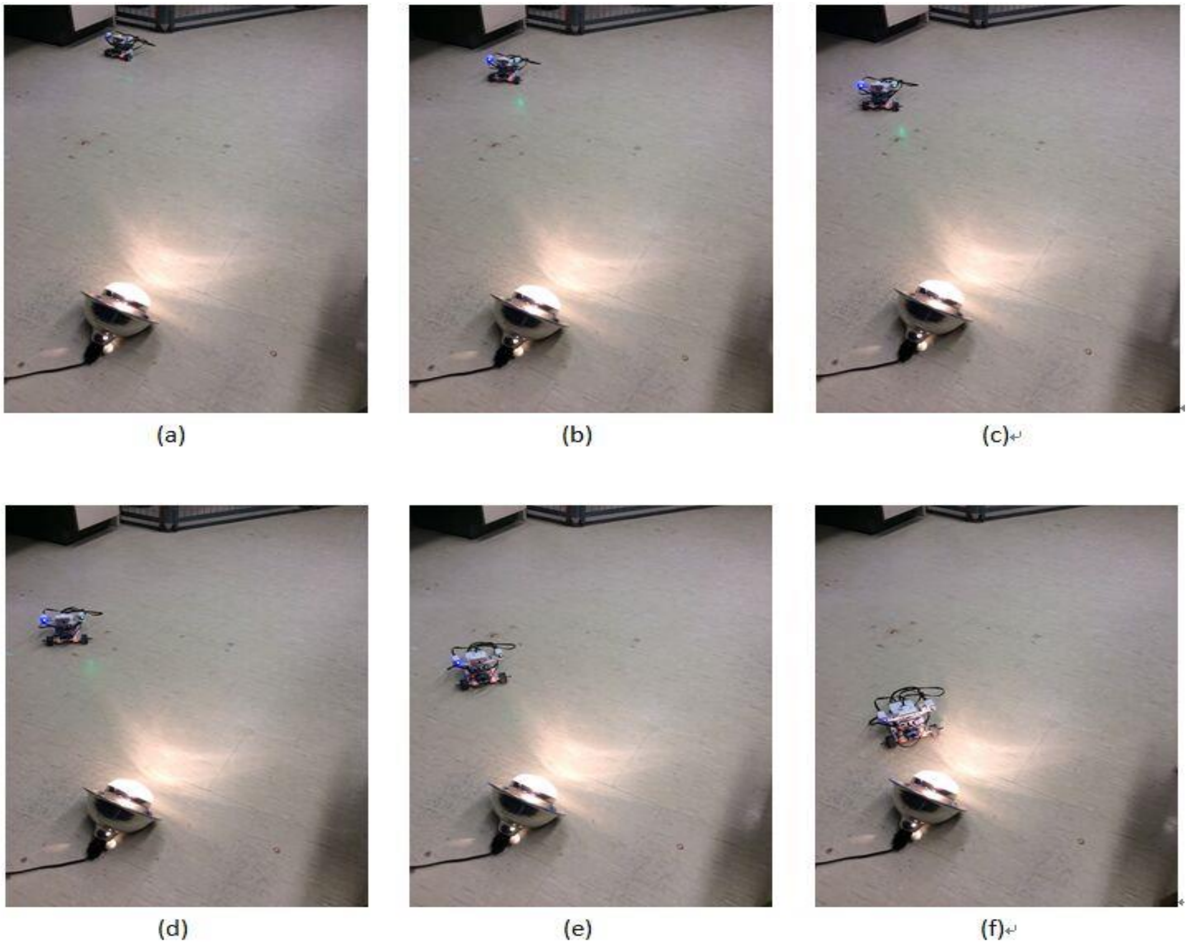


Figure 7.5 Robot approaching assumed large fire

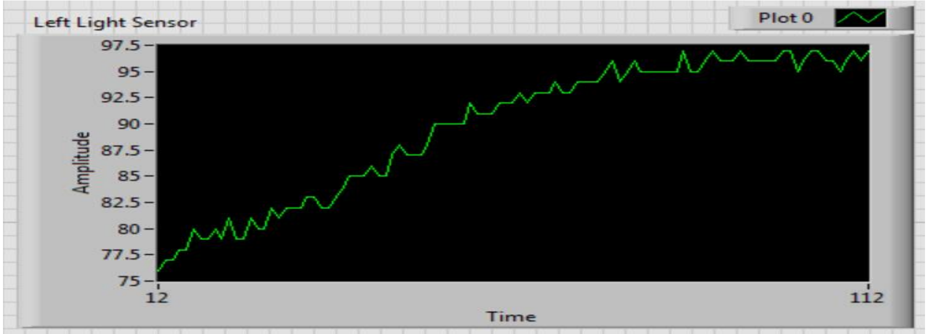


Figure 7.6(a) Real time light measurement

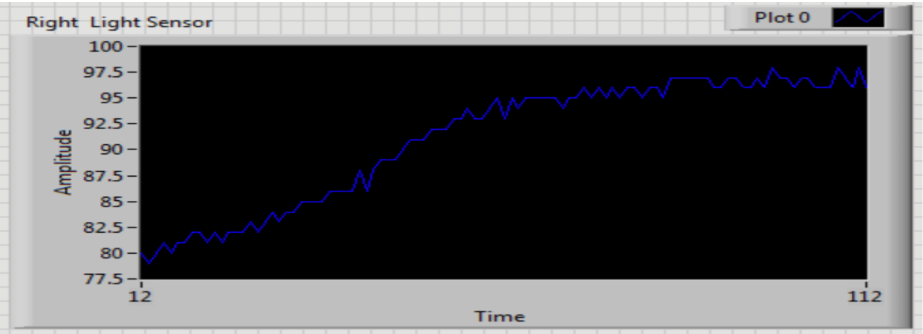


Figure 7.6(b) Real time light measurement

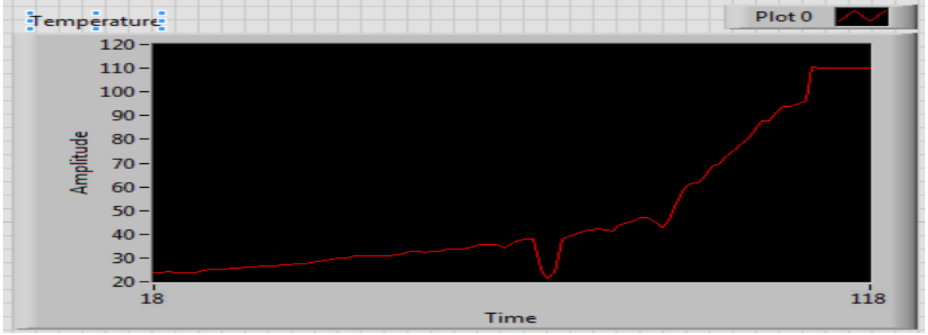


Figure 7.6(c) Real time temperature measurement

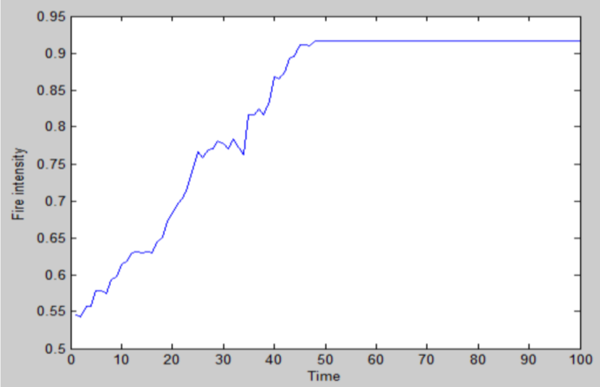
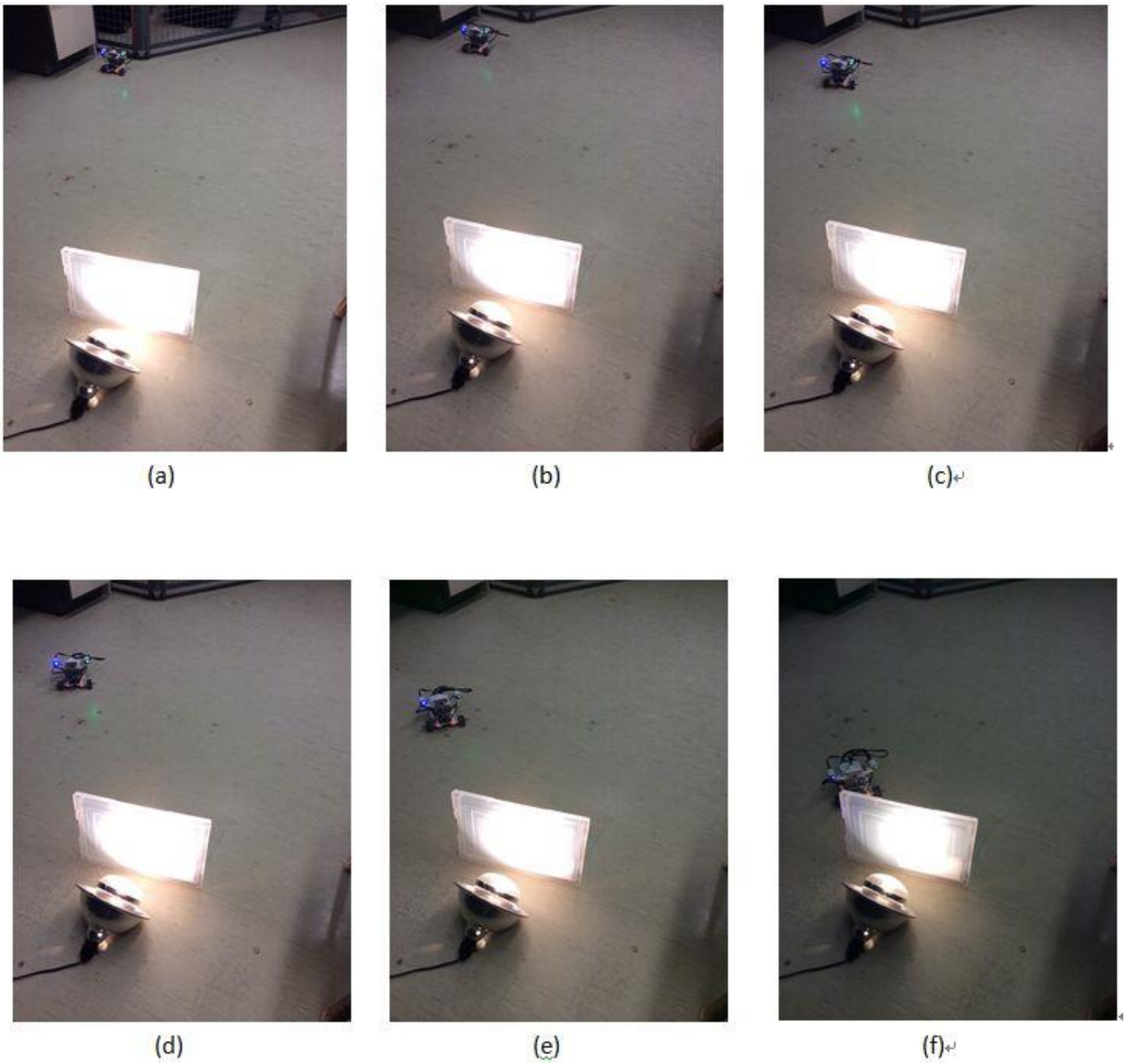


Figure 7.6(d) Real time computation for fire intensity

Scenario 3: Target (Small fire) placed in sight but not directly in front of the robot

In this experiment, robot was assigned to search the assumed medium fire. Figure 7.7(a) – (f) below shows the process of robot approaching target. Figure 7.8(a) – (c) shows the real time color sensor measurement and TIR sensor measurement. We can see from results light value started at 45 and ended up at 80 and temperature value started at 21℃ and finally reached about 26℃. The type-2 fuzzy logic system produced No fire at very beginning and switch to small fire very quickly. The type-2 fuzzy system produced small fire as final result.



(a) (b) (c)

(d) (e) (f)

Figure 7.7 Robot approaching assumed small

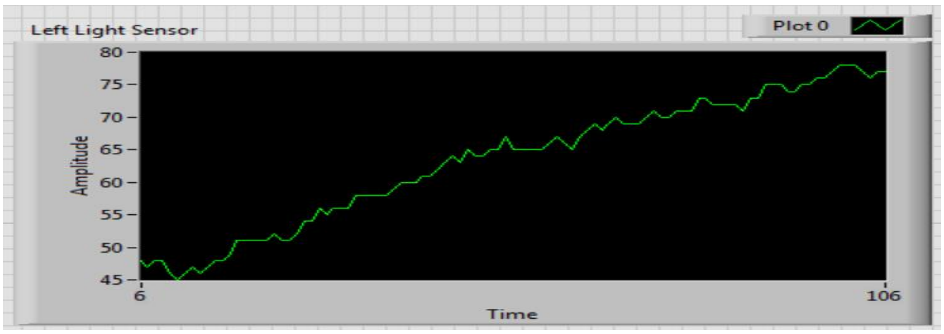


Figure 7.8(a) Real time light measurement

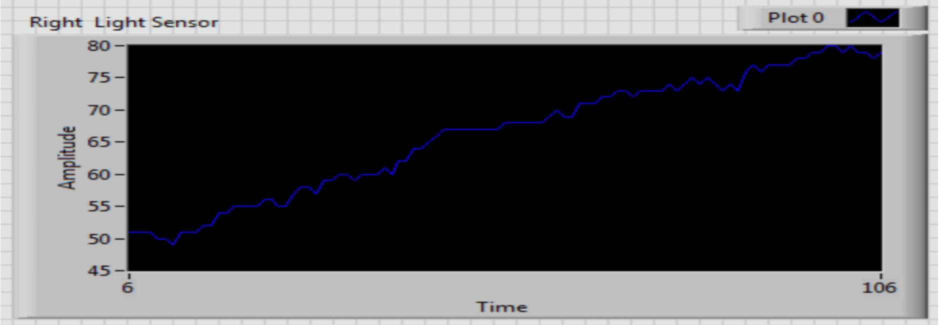


Figure 7.8(b) Real time light measurement

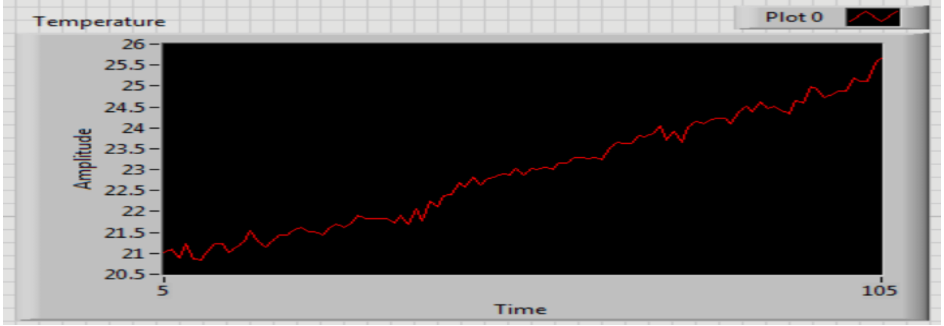


Figure 7.8(c) Real time temperature measurement

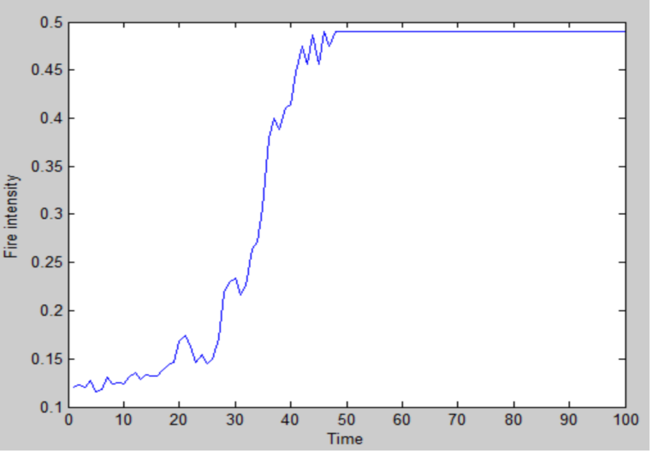
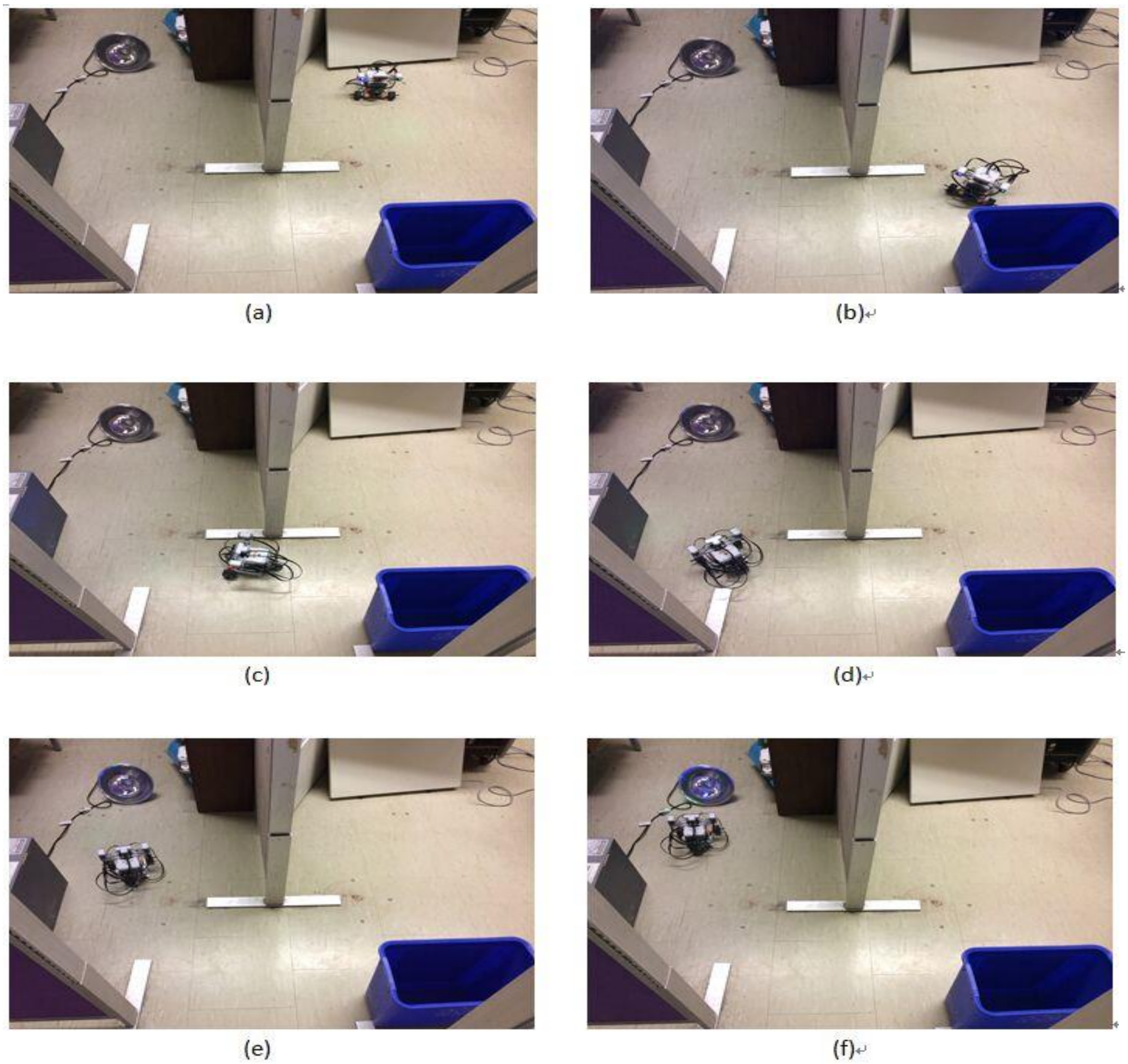


Figure 7.8(d) Real time computation for fire intensity

7.3 Experiments of Fire Detection Robot in Cluster environment

Scenario 4: Target (No fire) placed in complex terrain

Robot was assigned to search the lamp in cluster environment in this experiment. As usual Figure 7.9(a) – (f) below shows the process of robot approaching target. Figure 7.10(a) shows the real time measurement between robot and obstacles from ultrasonic sensor. Figure 7.10(b) – (d) shows the real time color sensor measurement and TIR sensor measurement. We can see from the results that both measured temperature and light keeps on a low level. The type-2 fuzzy logic system produced no fire as final result.



(a) (b)

(c) (d)

(e) (f)

Figure 7.9 Robot approaching the target

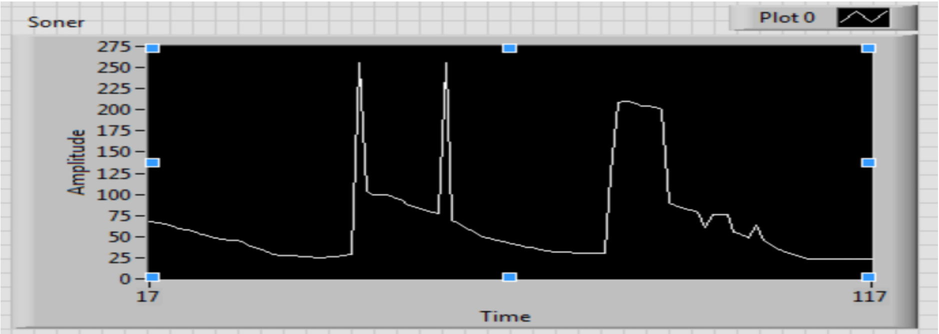


Figure 7.10(a) Real time distance measurement between robot and obstacles

Figure 7.10(b) Real time light measurement

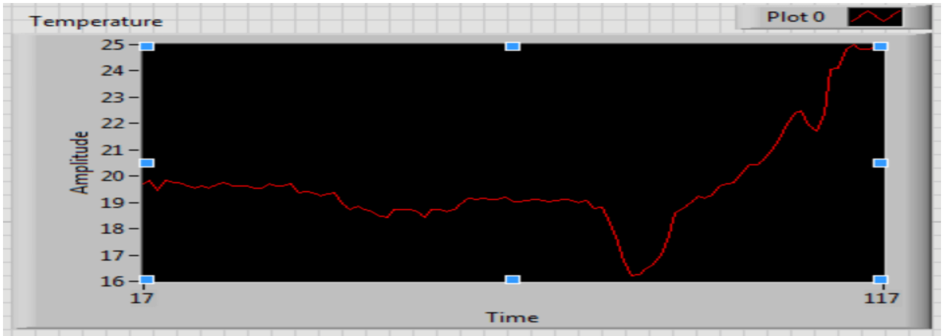


Figure 7.10(c) Real time temperature measurement

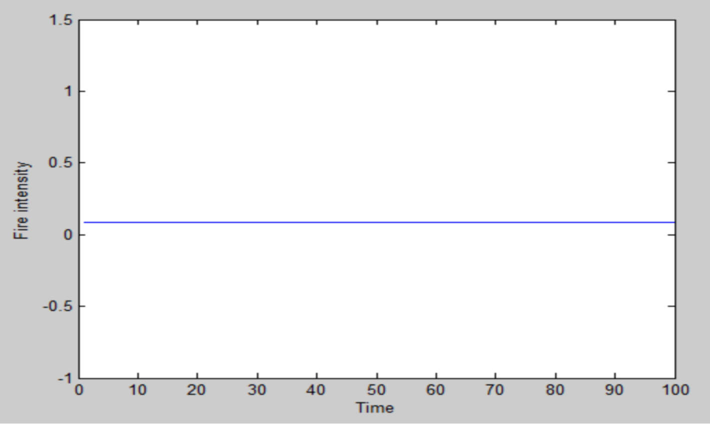
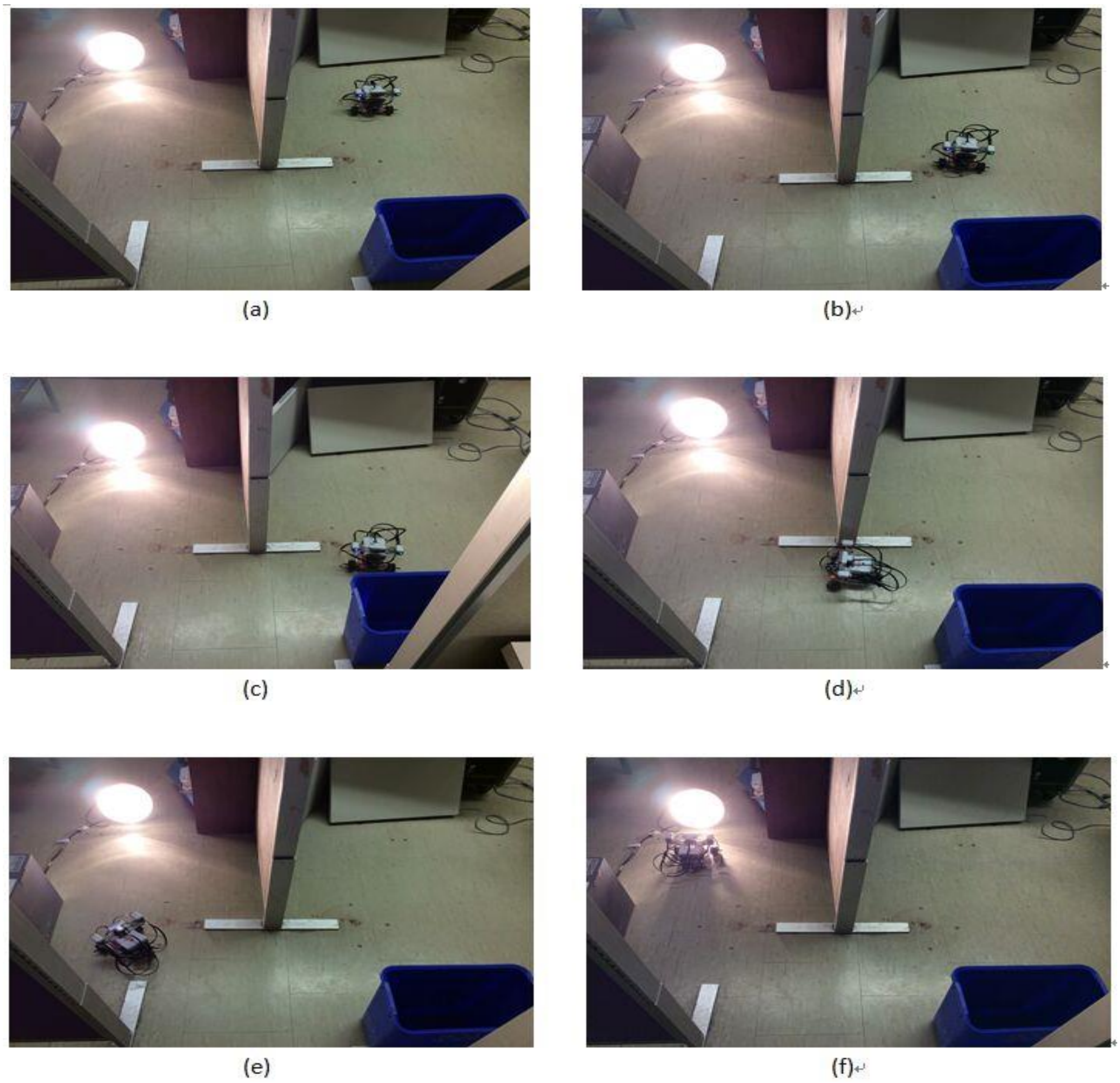


Figure 7.10(d) Real time computation for fire intensity

Scenario 5: Target (Large fire) placed in complex terrain

Robot was assigned to search the assumed large fire in cluster environment in this experiment. Figure 7.11(a) – (f) below shows the process of robot approaching target. Figure 7.12(a) shows the real time measurement between robot and obstacles from ultrasonic sensor. Figure 7.12(b) – (d) shows the real time color sensor measurement and TIR sensor measurement. The type-2 fuzzy logic system produced no fire at far-distance (low temperature and low light), small fire at medium-distance (high light but still low temperature) and large fire (high light and high temperature) at close-distance as result.



(a) (b)

(c) (d)

(e) (f)

Figure 7.11 Robot approaching the target

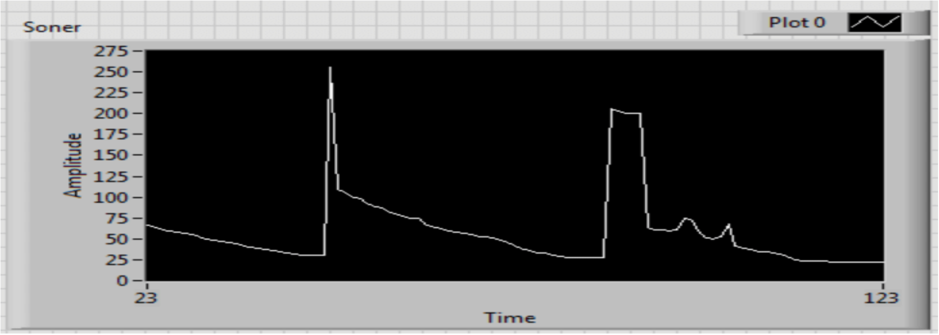


Figure 7.12(a) Real time distance measurement between robot and obstacles

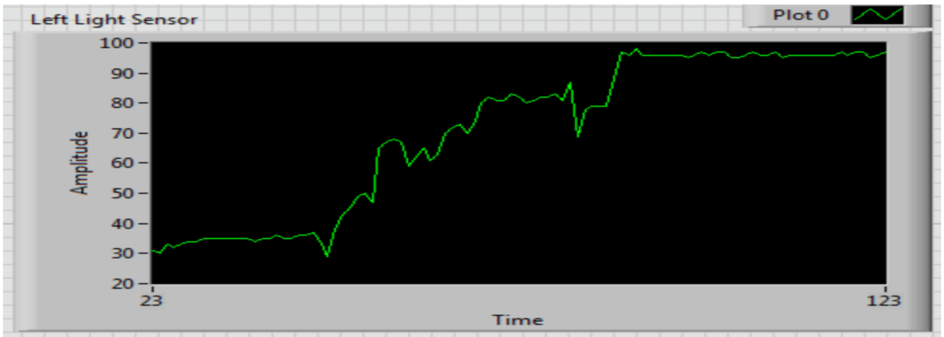


Figure 7.12(b) Real time light measurement

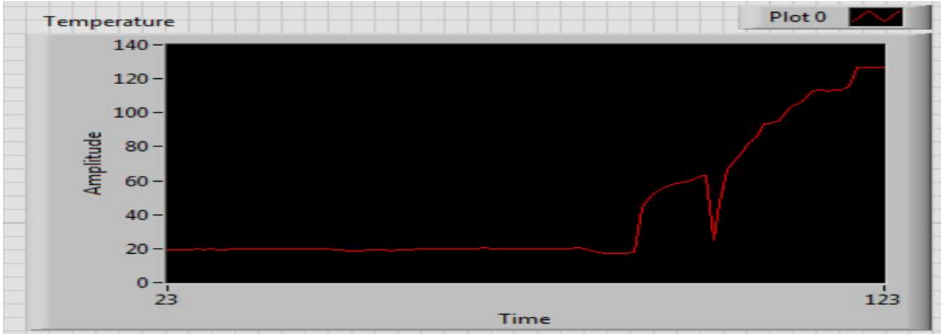


Figure 7.12(c) Real time temperature measurement

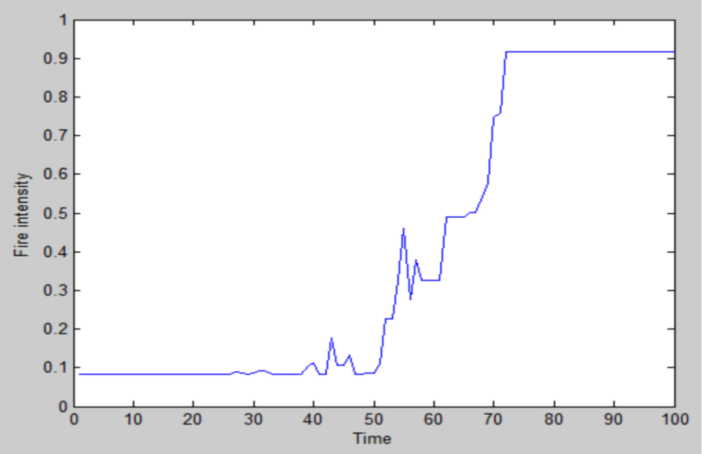
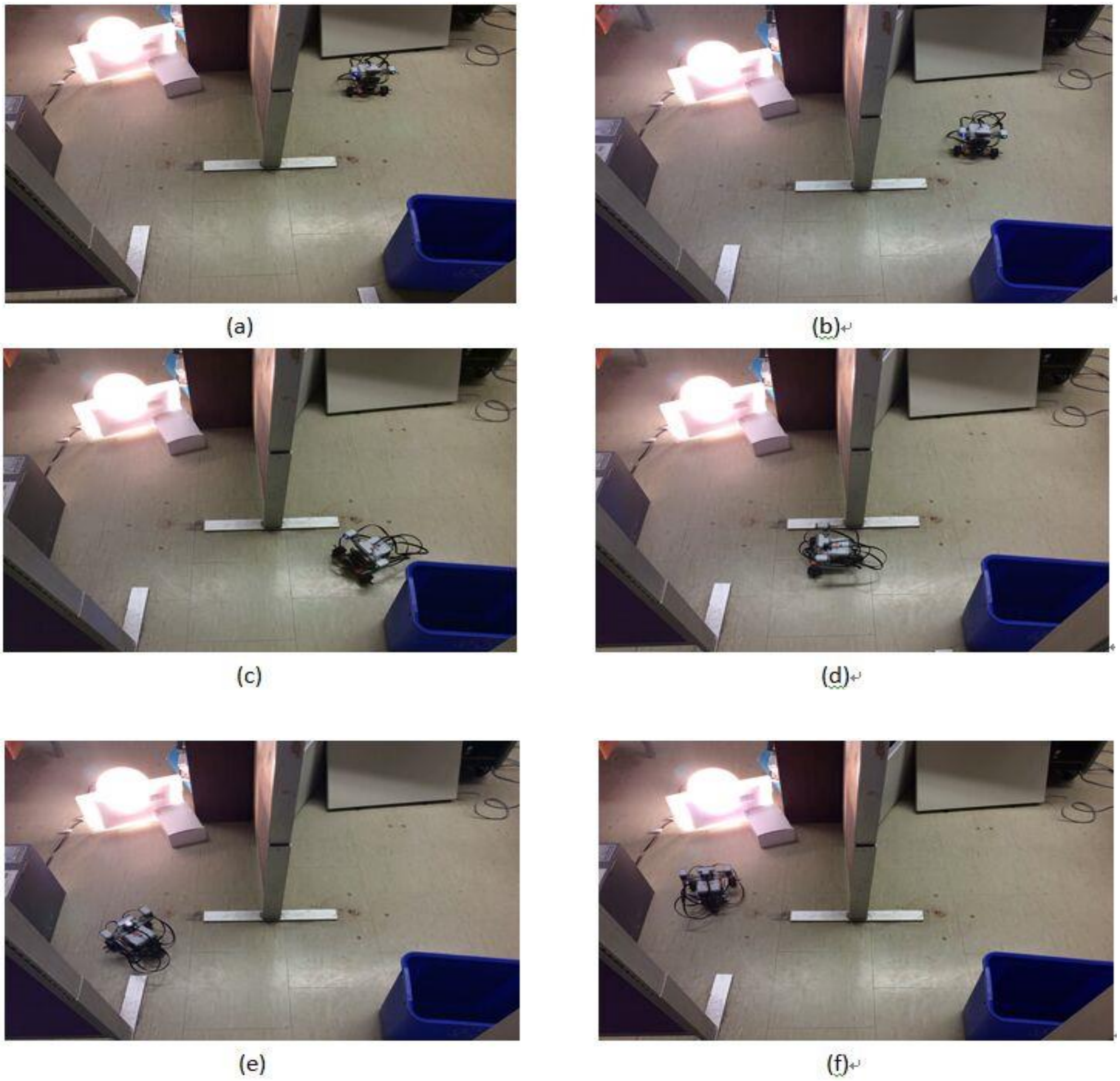


Figure 7.12(d) Real time computation for fire intensity

Scenario 6: Target (Small fire) placed in complex terrain

Robot was assigned to search the small fire in cluster environment in this experiment. Figure 13(a) – (f) below shows the process of robot approaching target. Figure 14(a) shows the real time measurement between robot and obstacles from ultrasonic sensor. Figure 14(b) – (d) shows the real time color sensor measurement and TIR sensor measurement. The type-2 fuzzy logic system produced no fire at far-distance, no fire at medium-distance and small fire at close-distance as result.



(a) (b)

(c) (d)

(e) (f)

Figure 7.13 Robot approaching the target

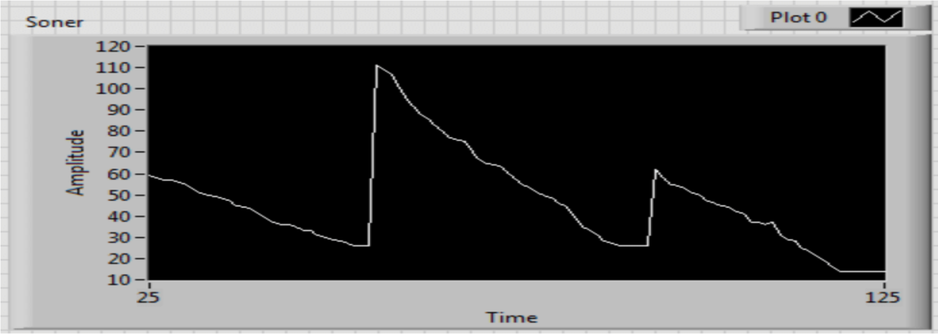


Figure 7.14(a) Real time distance measurement between robot and obstacles

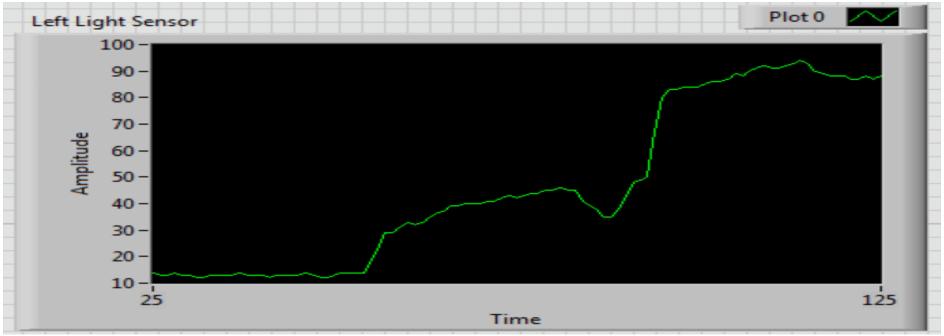


Figure 7.14(b) Real time light measurement

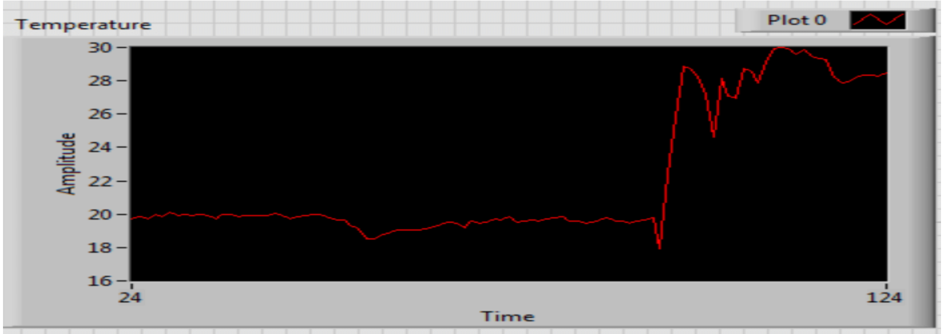


Figure 7.14(c) Real time temperature measurement

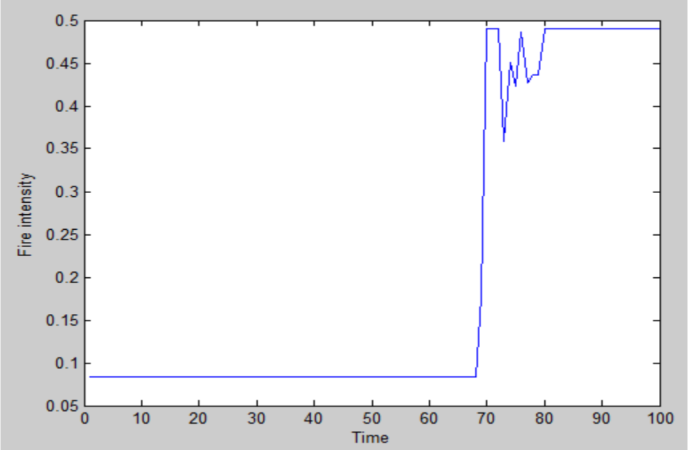


Figure 7.14(d) Real time computation for fire intensity

Chapter 8

Conclusions

8.1 Summary

The purpose of our research is to design a fire detection system using an autonomous robot. In order to achieve this target, we proposed a new approach based on type-2 fuzzy logic system that utilizes measured temperature and light intensity to detect fires of various intensities at different distances.

We started our researching by investigating type-1 fuzzy logic which is known for not needing exact mathematic model. In order to overcome some issues existing in type-1 fuzzy logic such as no solutions for taking opinions from multiple experts and handle more complicated uncertainty in our system, we introduce type-2 fuzzy logic to form our sensor fusion algorithm.

Instead of formulated the type-2 fuzzy logic system directly from prior knowledge of experts available for known facilities, a new approach for training experts’ expertise and setting up type-2 fuzzy logic system with its related parameters from data is discussed for the case of new facilities.

Simulations for testing the reliability and feasibility of our sensor fusion algorithms were carried out in MATLAB. In the simulation, we designed different scenarios to test the performance of the fire detection algorithm of our robot under different situations. Simulation results confirmed that our proposed type-2 fuzzy logic fire detection algorithm results in overcoming the lack of needed expertise to build a fuzzy logic system. The performance of type-2 fuzzy logic when compared to type-1 fuzzy logic for a fire detection robot is also discussed in this thesis.

Next step, we tested our proposed algorithm using NXT LEGO robot in the real physical environment. The experimental results confirmed the performance of the type-2 fuzzy logic fire detection system and validity of the simulation results.

8.2 Research Contribution

Fire detection has always been a critical challenge in our life. To avoid a fire disaster or perform a better rescue, there is an urgent need of an autonomous robot which can detect and approach a potential fire source in its early stage and give the estimation information of the fire intensity. In this research work, an approach for fire detection and estimation robots is presented. The approach is based on type-2 fuzzy logic system that utilizes measured temperature and light intensity to detect fires of various intensities at different distances. Simulation work was carried out on free space to test the feasibility and on complex new floor plan to test the reliability and efficiency of our proposed algorithm. Both simulation results and experimental results show the high performance of our proposed approach.

Fuzzy logic algorithm usually requests prior experts' experience to build a reliable system. Experts can provide suitable expertise if available. However for the case of a new facility, lack of expertise will limit the usage of fuzzy logic system. Our proposed method, avoids the need for prior expertise to build a fuzzy logic system by direct problem simulation. A model of cause (fire) to effects (temperature and light) is built in simulation. Data obtained from simulations are used to build type-2 fuzzy logic system. This helps the experts to design a more reliable fuzzy logic system to solve problems in unknown areas.

A comparison between type-1 fuzzy logic and type-2 fuzzy logic was also investigated in the thesis. By comparing the performance with type-1 fuzzy system, it has been found that type-2 fuzzy gives better results when large amount of uncertainty has to be concerned.

8.3 Future Work

In further research, a more comprehensive and accurate fire detection system could be developed with a robot platform with stronger processor and more sensor input ports. In our current studies, the scope of our proposed fire detection application is less comprehensive due to the limitation of LEGO NXT robot. Because there were no Smoke Sensor and Humility Sensor for NXT robot, we were not able to consider smoke and humidity in our research which are two very important features of fire. Further study can be easily done by considering more physical quantities produced by a fire with a more complex robot.

Another possibility of the future research is to include the fire extinguishing technics to the robot. By achieving this functionality, human being's involvement in a dangerous scenario such as a fire situation will be minimized. Because different type of fires request corresponding extinguishing tools to put it out, messing up with the extinguishing tools may lead to further disaster. Fire types study must be performed before this method can be implemented on an autonomous robot in order to successfully identify and put out different kinds of fire.

Reference

[1] V. Slavkovikja, S. Verstockta, S.V. Hoeckea, R.V Wallea. “Review of wildfire detection using social media”, Fire Safety Journal, volume 68, August 2014, pages 109-118.

[2] A. Muralidharan, F. Joseph. “Fire Detection System Using Fuzzy Logic”, International Journal of Engineering Sciences & Research Technology, 2014.

[3] N. Khoon, P. Sebastian. “Autonomous Fire Fighting Mobile Platform”, International Symposium on Robotics and Intelligent Sensors, 2012, pages 1145-1153.

[4] B. Ko, J. Jung, J. Nam, “Fire detection and 3D surface reconstruction based on stereoscopic pictures and probabilistic fuzzy logic”, Fire Safety Journal, August 2014, Pages 61-70.

[5] A.E. Çetin, K. Dimitropoulos, “Video fire detection-Review”, Digital Signal Processing, December 2013, Pages 1827-1843.

[6] T.N. Khoon, P. Sebastian, and A.B. Saman. “Autonomous Fire Fighting Mobile Platform,” Procedia Engineering, Janary 2012, pages 1145–1153.

[7] L.A. Klein, “Sensor and data fusion: a tool for information assessment and decision making”, Spie.org, 2004.

[8] E. Zervas, A. Mpimpoudis, C. Anagnostopoulos, O. Sekkas, and S. Hadjiefthymiades, “Multisensor data fusion for fire detection,” Inf. Fusion, July 2011, pages 150–159.

[9] A.K. Singh, H. Singh. “Forest Fire Detection through Wireless Sensor Network using Type-2 Fuzzy System”, International Journal of Computer Applications, August 2012, pages 19-23.

[10] P. Melin, O. Castillo. “A review on the applications of type-2 fuzzy logic in classification and pattern recognition”, Expert Systems with Applications, October 2013, pages 5413-5423.

[11] T. Nguyen. “Medical data classification using interval type-2 fuzzy logic system wavelets”, Applied Soft Computing, May 2015, pages 812-822.

[12] J.S. Martinez, R.I. Jone, D.Hissel, M.Pera. “A survey-based type-2 fuzzy logic system for energy management in hybrid electrical vehicles”, Information Science, May 2012, Pages 192-207.

[13] O. Castillo, P. Melin. “A review on interval type-2 fuzzy logic applications in intelligent control”, Information Sciences, September 2014, pages 615-631.

[14] Barnabas Bede. “Mathematics of Fuzzy Sets and Fuzzy Logic”, Spring-Verlag Berlin Heidelberg, Volume 295, 2013.

[15] L.A. Klein. “Sensor and Data Fusion: A Tool for Information Assessment and Decision Making”, Society of Photo-optical Instrumentation Engineers, 2004.

[16] J.M. Mendel, “Interval Type-2 Fuzzy Logic Systems Made Simple”, IEEE Transactions On Fuzzy Systems, December 2006, pages 808-821.

[17] J.M. Mendel, “Type-2 Fuzzy Sets and Systems: An Overview”, IEEE Computational Intelligence Magazine, February 2007, pages 20-29.

[18] L.A. Zadeh, "The Concept of a Linguistic Variable and Its Application to Approximate Reasoning–1," *Information Sciences*, 1975, pages 199–249.

[19] J. M. Mendel, "Advances in type-2 fuzzy sets and systems," Information Sciences, 2007, pages 84–110.

[20] J. M. Mendel, "Computing with words: Zadeh, Turing, Popper and Occam," IEEE Computational Intelligence Magazine, November 2007, pages 10-17.

[21] http://en.wikipedia.org/wiki/Lego_Mindstorms [Accessed: 20-Mar2015].

[22] LEGO MINDSTORMS User Guide [Accessed: 20-Mar2015].

[23]http://www.facstaff.bucknell.edu/mastascu/eLessonsHTML/Sensors/SensorsIntro.htm [Accessed: 20-Mar2015].

[24]http://en.wikipedia.org/wiki/Ultrasound [Accessed: 20-Mar2015].

[25]http://en.wikipedia.org/wiki/Speed_of_sound [Accessed: 20-Mar2015].

[26]http://www.dexterindustries.com/manual/lego-mindstorms-sensors/thermal-infrared-sensor/ [Accessed: 20-Mar2015].

[27]http://www.philohome.com/nxtmotor/nxtmotor.htm [Accessed: 20-Mar2015].

[28] http://www.ni.com/labview [Accessed: 20-Mar2015].

[29] https://code.google.com/p/fds-smv/ [Accessed: 20-Mar2015].

Appendix A

MATLAB: Fire Detection Robot Using Type-2 Fuzzy Logic Sensor Fusion

A.1 Main Function

```
% Clear all windows
 clc
 clf
 clear

 % Initial Robot Values
 robot_pose = [20 , 20 , pi/4] ;
 sensor_radius = 4;
 max_vel = 4 ;
 vel = 4;
 max_ang_vel = 1 ;
 possibility = 0;
 i=1;



 % Map Dimensions
 map_max = 160 ;
 map_min = 0 ;

 % Goal Position on the map
 goal = [140,120];
 %goal = [50*rand() ,50*rand()] ;
 goal_radius = sensor_radius*4 ;
```

```
% Time settings
t_sim = 100;
T = 0.05 ;
t_steps = floor(t_sim/T) ;

% random set up a large fire, medium fire, small fire or no fire
rand_fire = 100*rand();
  if rand_fire>=0 && rand_fire<25
     temp = 0;
     light = 0;
     fireType = 1;
  elseif rand_fire>=25 && rand_fire<50
     temp = 200;
     light = 400;
     fireType = 2;
  elseif rand_fire>=50 && rand_fire<75
     temp = 600;
     light = 1200;
     fireType = 3;
  else
     temp = 1000;
     light = 2000;
     fireType = 4;
  end


% Create Movie, Step
j = 0 ;
for t = 1 : 200;

% Axis properties
clf ;
xmax = map_max ;
xmin = map_min ;
ymax = map_max ;
ymin= 0 ;

axis([xmin,xmax+2*goal_radius,ymin,ymax+2*goal_radius]);
axis equal; axis manual; grid on; hold on;
```

```
 % Create different source image with different color by fire type
 if fireType == 1

 elseif fireType == 2
      scatter(goal(1),goal(2),100,'d','g','filled');
 rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,2*goal_radius,2*goal_radius],...
 'Curvature',[1,1],'EdgeColor','g');
 elseif  fireType == 3
      scatter(goal(1),goal(2),100,'d','b','filled');
 rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,2*goal_radius,2*goal_radius],...
 'Curvature',[1,1],'EdgeColor','b');
 elseif  fireType == 5
      scatter(goal(1),goal(2),100,'d','b','filled');

 rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,2*goal_radius,2*goal_radius],...
 'Curvature',[1,1],'EdgeColor','b');

 else
      scatter(goal(1),goal(2),100,'d','r','filled');
 rectangle('Position',[goal(1)-goal_radius,goal(2)-goal_radius,2*goal_radius,2*goal_radius],...
 'Curvature',[1,1],'EdgeColor','r');
 end



 % get robot position information
 robot = [robot_pose(1),robot_pose(2),robot_pose(3)];

 % calling fuzzy_logic sub-function
[distance,possibility,tempp,dis_to_fire,lightt,fary,midy,neary] =
fuzzy_logic(robot_pose,goal,temp,light);

% prepare variables for ploying results after application
faryy(i) = fary;
midyy(i) = midy;
nearyy(i) = neary;
possib(i)= possibility;
time(i)=i;
dis_to_fireb(i)=dis_to_fire;
```

```
tempb(i) = tempp;
lightb(i) = lightt;
distanceb(i) = distance;
i=i+1;

 %robot movement subfunction
 [vel,new_position] = robot_move2(robot,possibility,goal,vel,tempp);

 %calculate robot new pose
 robot_pose = [new_position(1),new_position(2),new_position(3)];

 % plot robot on movie
 [A,B] = robot_round(robot_pose,sensor_radius);

 % plot robot direction on movie
 [XL,YL] = moving_arrows(robot_pose);
 fill(XL,YL,'r');

 % plot robot trajectory
 plot(data(1:t,1), data(1:t,2), 'r');

 % capture images on certain frame
 if rem(t,50)==0 || t==1 || t==25
    filename=strcat('',int2str(t),'.jpg');
     saveas(gcf,filename);
 end

 hold off;

 j = j +1;
 M(j) = getframe;

 end



 %plot the fire intensity result
  figure;
 plot(distanceb/10,possib/100)
 xlabel('Distance(m)');
 ylabel('Fire intensity');
```

```
set(gca,'XDir','reverse')

%plot real time temperature measurements
figure;
r = 0;
tempbb=tempb+r;
 plot(distanceb/10,tempbb)
xlabel('Distance(m)');
ylabel('Temperature mesurement(\circC)');
set(gca,'XDir','reverse')

%plot real time light measurements
 figure;
 r = 0;
lightbb=lightb+r;
 plot(distanceb/10,lightbb)
xlabel('Distance(m)');
ylabel('Light mesurement(lm)');
set(gca,'XDir','reverse')
```

A.2 Type-2 Fuzzy Logic Main Function

```
function
[distance,possibility,tempp,dis_to_fire,lightt,fary,midy,neary] =
fuzzy_logic(robot_pose,goal,temp1,light1)

    %get distance from robot to goal
    dis_x = robot_pose(1)-goal(1);
    dis_y = robot_pose(2)-goal(2);

    %setup temperature and light value from orrcued fire
    temp = temp1;
    light = light1;

    %calculation total distance
    dis_total = abs(sqrt((dis_x)*(dis_x)+(dis_y)*(dis_y)));
    dis_total_temp = dis_total/10;
```

```
    %real time simulated temperature and light value at robot 
position
    temp_input = 20+(temp)/(dis_total_temp*dis_total_temp);
    light_input = 40+(light)/(dis_total_temp*dis_total_temp);

    fary=0; midy=0; neary=0;

    % far distance type-2 fuzzy logic system activated
    if dis_total_temp>=9
    [Y1 YL1 YR1]=MYIT2_far(temp_input,light_input);
    fary = Y1;

    % far distance type-2 fuzzy logic system activated
    elseif dis_total_temp>=6 && dis_total_temp<9
        [Y1 YL1 YR1]=MYIT2_mid(temp_input,light_input);
    midy = Y1;

    % far distance type-2 fuzzy logic system activated
    elseif dis_total_temp>=1 && dis_total_temp<6
        [Y1 YL1 YR1]=MYIT2_near(temp_input,light_input);
    neary = Y1;
    end
    [Y YL YR]=MYIT2(temp_input,light_input);



    %prepare outputs to main funtion
    distance = dis_total;
    possibility = Y1;
    dis_to_fire = Y;
    tempp = temp_input;
    lightt = light_input;

end
```

A.3 Far-Module Type-2 Fuzzy Logic System Parameter setting up

```
function[Y YL YR]=MYIT2_far(a,b)

%set up type-2 membership function parameters for temperature input
A1=16; A2=16; A3=16; A4=21.5; A5=16; A6=16; A7=16; A8=20.5; A9=1;
B1=17.5; B2=22; B3=22; B4=31; B5=18.5; B6=22; B7=22; B8=31; B9=1;
C1=27.5; C2=32; C3=32; C4=29.5; C5=27.5; C6=32; C7=32; C8=28.5; C9=1;
G1=27.5; G2=40; G3=40; G4=40; G5=28.5; G6=40; G7=40; G8=40; G9=1;

%set up type-2 membership function parameters for light input
D1=A1*2; D2=A2*2; D3=A3*2; D4=A4*2; D5=A5*2; D6=A6*2; D7=A7*2;
D8=A8*2; D9=A9;
E1=B1*2; E2=B2*2; E3=B3*2; E4=B4*2; E5=B5*2; E6=B6*2; E7=B7*2;
E8=B8*2; E9=B9;
F1=C1*2; F2=C2*2; F3=C3*2; F4=C4*2; F5=C5*2; F6=C6*2; F7=C7*2;
F8=C8*2; F9=C9;
H1=G1*2; H2=G2*2; H3=G3*2; H4=G4*2; H5=G5*2; H6=G6*2; H7=G7*2;
H8=G8*2; H9=G9*2;

% building type-2 fuzzy logic inputs and rules
X=[A1 A2 A3 A4 A5 A6 A7 A8 A9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 H1 H2 H3 H4 H5 H6 H7 H8 H9];
```

```
% setting up type-2 output parameters and rules
Y=[7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   36.5 38.5
   36.5 38.5
   36.5 38.5
   61.5 63.5
   61.5 63.5
   61.5 63.5
   90.7 92.7
   90.7 92.7
   90.7 92.7];

%get inputs array
x=[a b];

%fuzzy calculation subfuntion
[y, yl, yr]=IT2FLS(x,X,Y);

%prepare outputs to main function
Y = y;
YL = yl;
YR = yr;
```

A.4 Medium-Module Type-2 Fuzzy Logic System Parameter setting up

```
function[Y YL YR]=MYIT2_mid(a,b)

%set up type-2 membership function parameters for temperature input
A1=16; A2=16; A3=16; A4=23; A5=16; A6=16; A7=16; A8=21; A9=1;
```

```
B1=17; B2=26; B3=26; B4=33; B5=19; B6=26; B7=26; B8=31; B9=1;
C1=27; C2=37; C3=37; C4=43; C5=29; C6=37; C7=37; C8=41; C9=1;
G1=37; G2=48; G3=48; G4=48; G5=39; G6=48; G7=48; G8=48; G9=1;

%set up type-2 membership function parameters for light input
D1=A1*2; D2=A2*2; D3=A3*2; D4=A4*2; D5=A5*2; D6=A6*2; D7=A7*2;
D8=A8*2; D9=A9;
E1=B1*2; E2=B2*2; E3=B3*2; E4=B4*2; E5=B5*2; E6=B6*2; E7=B7*2;
E8=B8*2; E9=B9;
F1=C1*2; F2=C2*2; F3=C3*2; F4=C4*2; F5=C5*2; F6=C6*2; F7=C7*2;
F8=C8*2; F9=C9;
H1=G1*2; H2=G2*2; H3=G3*2; H4=G4*2; H5=G5*2; H6=G6*2; H7=G7*2;
H8=G8*2; H9=G9*2;

% building type-2 fuzzy logic inputs and rules
X=[A1 A2 A3 A4 A5 A6 A7 A8 A9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 H1 H2 H3 H4 H5 H6 H7 H8 H9];

% setting up type-2 output parameters and rules
Y=[7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
```

```
   36.5 38.5
   36.5 38.5
   36.5 38.5
   61.5 63.5
   61.5 63.5
   61.5 63.5
   90.7 92.7
   90.7 92.7
   90.7 92.7];

%get inputs array
x=[a b];

%fuzzy calculation sub-function
[y, yl, yr]=IT2FLS(x,X,Y);

%prepare outputs to main function
Y = y;
YL = yl;
YR = yr;
```

A.5 Close-Module Type-2 Fuzzy Logic System Parameter setting up

```
function[Y YL YR]=MYIT2_near(a,b)

%set up type-2 membership function parameters for temperature input
A1=16; A2=16; A3=16; A4=23; A5=16; A6=16; A7=16; A8=23; A9=1;
B1=17; B2=42; B3=42; B4=52; B5=17; B6=42; B7=42; B8=52; B9=1;
C1=42; C2=86; C3=86; C4=98; C5=42; C6=86; C7=86; C8=98; C9=1;
G1=90; G2=131; G3=131; G4=131; G5=90; G6=131; G7=131; G8=131; G9=1;

%set up type-2 membership function parameters for light input
D1=A1*2; D2=A2*2; D3=A3*2; D4=A4*2; D5=A5*2; D6=A6*2; D7=A7*2;
D8=A8*2; D9=A9;
```

```
E1=B1*2; E2=B2*2; E3=B3*2; E4=B4*2; E5=B5*2; E6=B6*2; E7=B7*2;
E8=B8*2; E9=B9;
F1=C1*2; F2=C2*2; F3=C3*2; F4=C4*2; F5=C5*2; F6=C6*2; F7=C7*2;
F8=C8*2; F9=C9;
H1=G1*2; H2=G2*2; H3=G3*2; H4=G4*2; H5=G5*2; H6=G6*2; H7=G7*2;
H8=G8*2; H9=G9*2;

% building type-2 fuzzy logic inputs and rules
X=[A1 A2 A3 A4 A5 A6 A7 A8 A9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   A1 A2 A3 A4 A5 A6 A7 A8 A9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 D1 D2 D3 D4 D5 D6 D7 D8 D9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   B1 B2 B3 B4 B5 B6 B7 B8 B9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 E1 E2 E3 E4 E5 E6 E7 E8 E9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 H1 H2 H3 H4 H5 H6 H7 H8 H9;
   G1 G2 G3 G4 G5 G6 G7 G8 G9 F1 F2 F3 F4 F5 F6 F7 F8 F9;
   C1 C2 C3 C4 C5 C6 C7 C8 C9 H1 H2 H3 H4 H5 H6 H7 H8 H9];

% setting up type-2 output parameters and rules
Y=[7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   7.3 9.3
   36.5 38.5
   36.5 38.5
   36.5 38.5
   61.5 63.5
   61.5 63.5
   61.5 63.5
   90.7 92.7
   90.7 92.7
```

```
   90.7 92.7];

%get inputs array
x=[a b];

%fuzzy calculation sub-function
[y, yl, yr]=IT2FLS(x,X,Y);

%prepare outputs to main function
Y = y;
YL = yl;
YR = yr;
```

A.6 Type-2 Fuzzy Logic Algorithm Calculation

```
function [y,yl,yr]=IT2FLS(x,X,Y)

% y=IT2FLS(x,X,Y)
%
% x: Input to the IT2 FLS. Assume the FLS has N consequents and 1 
consequent. Then, x can be a 1*N vector
%    or M*N matrix, each row corresponding to an input vector.
%
% X: Matrix describing the consequents part of the rulebase. Assume 
the rulebase has K rules:
%    IF x1 is X11 and x2 is X21, THEN y is y1
%    ...
%    IF x1 is X1K and x2 is X2K, THEN y is yK
% where X11-X2K are IT2 FSs and y1-yK are crisp intervals. Then,
%      X=[MF of X11, MF of X21;
%         ...
%         MF of X1K, MF of X2K]
% Note that each MF is represented by 9 points.
%
% Y: Matrix describing the consequent part of the rulebase. For the 
above rulebase,
%    Y=[y1;
%       ...
```

```
%       yK];
%
% y: Output of the IT2 FLS. It has as many rows as x, each row is the
output for the corresponding row of x.
%    y=(yl+yr)/2
%
% yl: Lower bounds of the intervals output by the type-reducer.
%
% yr: Upper bounds of the intervals output by the type-reducer.
%
% Dongrui Wu, GE Research (drwu09@gmail.com)
%

if nargin<3
    display('Function IT2FLS: Must have three arguments. Abort.');
return;
end

if size(X,1)~=size(Y,1)
    display('Function IT2FLS: X must have the same number of rows as
Y. Abort.'); return;
end

if size(x,2)~=size(X,2)/9
    display('Function IT2FLS: The number of columns in x must be
equal to the number of IT2 FSs in each row of X. Abort'); return;
end

y=zeros(size(x,1),1);
yr=y; yl=y;
for i=1:size(x,1)
    fl=ones(1,size(X,1)); fu=fl;
    for j=1:size(X,1)
        for k=1:size(X,2)/9
            fu(j)=fu(j)*mg(x(i,k),X(j,9*(k-1)+(1:4)));
            fl(j)=fl(j)*X(j,9*k)*mg(x(i,k),X(j,9*(k-1)+(5:8)));
        end
    end
    [y(i),yl(i),yr(i)]=EIASC(Y(:,1)',Y(:,2)',fl,fu);
end
```

```
function [y,yl,yr,l,r]=EIASC(Xl,Xr,Wl,Wr,needSort)

%
% function [y,yl,yr,l,r]=EIASC(Xl,Xr,Wl,Wr,needSort)
%
% function to implement the EIASC algorithm in:
%
% D. Wu and M. Nie, "Comparison and Practical Implementation of Type-Reduction Algorithms for Type-2
% Fuzzy Sets and Systems," IEEE International Conference on Fuzzy Systems, Taipei, Taiwan, June 2011.
%
% Dongrui WU, GE Research (drwu09@gmail.com), 7/18/2010
%
% Xl: A row vector containing the lower bounds of x
% Xr: A row vector containing the upper bounds of x
% Wl: A row vector containing the lower bounds of w
% Wr: A row vector containing the upper bounds of w
% needSort: “1” if at least one of Xl and Xr is not in ascending order.
%            “0” if both Xl and Xr are in ascending order. Default “1.”
%y: (yl+yr)/2
%yl: lower bound of the type-reduced output
%yr: upper bound of the type-reduced output
%l: switch point for yl
%r: switch point for yr

ly=length(Xl); XrEmpty=isempty(Xr);
if XrEmpty;  Xr=Xl; end
if max(Wl)==0
    yl=min(Xl); yr=max(Xr);
    y=(yl+yr)/2;  l=1; r=ly-1; return;
end
index=find(Wr<10^(-10));
if length(index)==ly
    yl=min(Xl); yr=max(Xr);
    y=(yl+yr)/2; l=1; r=ly-1; return;
end
```

```
Xl(index)=[]; Xr(index)=[];
Wl(index)=[]; Wr(index)=[];
if nargin==4;  needSort=1; end

% Compute yl
if  needSort
    [Xl,index]=sort(Xl); Xr=Xr(index);
    Wl=Wl(index); Wr=Wr(index);
end
Wl2=Wl; Wr2=Wr;
for i=length(Xl):-1:2 % Make Xl unique
    if Xl(i)==Xl(i-1)
        Wl(i)=Wl(i)+Wl(i-1);
        Wr(i)=Wr(i)+Wr(i-1); Xl(i)=[];
        Wl(i-1)=[]; Wr(i-1)=[];
    end
end
ly=length(Xl);
if ly==1
    yl=Xl;  l=1;
else
    yl=Xl(end); l=1;
    a=Xl*Wl'; b=sum(Wl);
    while l < ly && yl > Xl(l)
        a=a+Xl(l)*(Wr(l)-Wl(l));
        b=b+Wr(l)-Wl(l);
        yl=a/b;   l=l+1;
    end
end

% Compute yr
if ~XrEmpty && needSort==1
    [Xr,index]=sort(Xr);
    Wl=Wl2(index); Wr=Wr2(index);
end
if ~XrEmpty
    for i=length(Xr):-1:2 % Make Xr unique
        if Xr(i)==Xr(i-1)
            Wl(i)=Wl(i)+Wl(i-1);
            Wr(i)=Wr(i)+Wr(i-1); Xr(i)=[];
            Wl(i-1)=[]; Wr(i-1)=[];
```

```
        end
    end
end
ly=length(Xr);
if ly==1
    yr=Xr; r=1;
else
    r=ly; yr=Xr(1);
    a=Xr*Wl'; b=sum(Wl);
    while r>0 && yr < Xr(r)
        a=a+Xr(r)*(Wr(r)-Wl(r));
        b=b+Wr(r)-Wl(r);
        yr=a/b;  r=r-1;
    end
end
y=(yl+yr)/2;



function u=mg(x,xMF,uMF)

% u=mg(x,xMF,uMF)
% function to compute the membership grades of x on a T1 FS
% Dongrui WU, GE Research (drwu09@gmail.com), 7/18/2010
%
% xMF: x-coordinates of the T1 FS
% uMF: u-coordinates of the T1 FS; default to be [0 1 1 0]
% u: membership of x on the T1 FS

if nargin==2
    uMF=[0 1 1 0];
elseif length(xMF)~=length(uMF)
    display('Function mg: xMF and uMF must have the same length.
Abort.'); return;
end

[xMF,index]=sort(xMF); uMF=uMF(index);

u=zeros(size(x));
for i=1:length(x)
    if x(i)<=xMF(1)
```

```
        if xMF(1)==xMF(2)
            u(i)=1;
        end
    elseif x(i)>=xMF(end)
        if xMF(end-1)==xMF(end)
            u(i)=1;
        end
    else
        left=find(xMF<x(i),1,'last');     right=left+1;
        u(i)=uMF(left)+(uMF(right)-uMF(left))*(x(i)-
xMF(left))/(xMF(right)-xMF(left));
    end
end
```

A.7 Robot Movement Calculation

```
function [now_vel,new_position] =
robot_move2(robot,possibility,goal,vel,tempp)

%get robot old pose and position
x = robot(1);
y = robot(2);
theta = robot(3);
radius=4;
r=radius/2;
k=radius/2;

% calculate robot's position to fire source
dis = abs(sqrt((x-goal(1))^2+(y-goal(2))^2));

O=[x+r*cos(theta),y+r*sin(theta)];
A=[O(1)-k*cos(pi/2-theta),O(2)+k*sin(pi/2-theta)];
B=[O(1)+k*sin(theta),O(2)-k*cos(theta)];



disA = abs(sqrt((A(1)-goal(1))^2+(A(2)-goal(2))^2));
disB = abs(sqrt((B(1)-goal(1))^2+(B(2)-goal(2))^2));

% robot moving based on different light value sensored by simulation
```

```
if disA>disB && disA-disB < 10 && possibility>40 && possibility < 80
&& dis>30
    max_vel = 0.8;
    theta = robot(3)- pi/120;
elseif disA<disB && disB-disA < 10 && possibility >40 && possibility
< 80 && dis>30
    max_vel = 0.8;
    theta = robot(3)+ pi/120;
elseif dis<=30;
    max_vel = 0;
else
    max_vel = 0.8;
    theta = robot(3);
end

    % setting up new pose and position
    delta_d = [max_vel*cos(theta),max_vel*sin(theta)];
    new_position = [robot(1)+delta_d(1),robot(2)+delta_d(2),theta];
    now_vel = max_vel;

end
```

A.8 Plot the simulated robot and corresponding sensors

```
function[A,B,circle] = robot_round(Q,radius)

% setting up initial parameters
x = Q(1);
y = Q(2);
theta = Q(3);
r=radius;
k=radius;

% draw simulated ultrasonic beans
for beam_angle = (theta-pi/2) : pi/12 : (theta+pi/2);
```

```
    beamX = [x,x+4*radius*cos(beam_angle)];
    beamY = [y,y+4*radius*sin(beam_angle)];
    plot(beamX,beamY,'c');
end

% draw front bar implemented on robot
O=[x+r*cos(theta),y+r*sin(theta)];
A=[O(1)-k*cos(pi/2-theta),O(2)+k*sin(pi/2-theta)];
B=[O(1)+k*sin(theta),O(2)-k*cos(theta)];

plot([A(1),B(1)],[A(2),B(2)],'k-','LineWidth',3)

% plot simulated two light sensors and temperature sensor
rectangle('Position',...
    [A(1)-0.5,A(2)-0.5, 1, 1],...
    'Curvature',[0,0],'EdgeColor','b','facecolor','r');

rectangle('Position',...
    [B(1)-0.5,B(2)-0.5, 1, 1],...
    'Curvature',[0,0],'EdgeColor','b','facecolor','r');

rectangle('Position',...
    [O(1)-0.5,O(2)-0.5, 1, 1],...
    'Curvature',[0,0],'EdgeColor','b','facecolor','c');

% plot circle based on detection range for ultrasonic sensor
circle = rectangle ( 'Position',....
[ x-4*radius, y-4*radius , 8 * radius , 8 *
radius ],'Curvature',[1,1],'EdgeColor','b') ;
```

Appendix B

LabView: Fire Detection Robot using Type-2 Fuzzy Logic