ROBOT COMPANION FOR ADAPTIVE HOME ENVIRONMENT

Subasiri Mudiyanselage Supun Saranga Senarathna

Degree of Master of Science

Department of Electrical Engineering

University of Moratuwa
Sri Lanka

June 2018

ROBOT COMPANION FOR ADAPTIVE HOME ENVIRONMENT

Subasiri Mudiyanselage Supun Saranga Senarathna

(159288F)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Since in Industrial Automation

Department of Electrical Engineering

University of Moratuwa
Sri Lanka

June 2018

DECLARATION

I declare that this is my own work and this thesis does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief, it does not contain any material previously published or written by another person except where due reference is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature :…………………….. Date :…………………..

The above candidate has carried out research for the Master’s thesis under my supervision.

Dr. A.G.B.P. Jayasekara

Signature : ..………………. Date :…………………

Abstract

Robot companions are being develop to assist humans in domestic environment. Sooner rather than later, robot buddies will be a piece of our typical family's day by day life attempting to help with local errands and dealing with us when required. Notwithstanding, before this turns into a reality, imperative issues should be tended to in the Human-Robot Interaction (HRI) field so as to accomplish social robots equipped for communicating with people also to which people associate with each other.

One of the greatest difficulties in this field is to endow robot associates with those social capacities expected to collaborate with a man amid a consistent period. These aptitudes can be upgraded by utilizing robot's own sensors or outer tactile frameworks introduced around the trial condition, so robots could know about the logical data. The fuse of these capacities hopes to enhance the association and the robot colleagues acknowledgment by people. Individuals have desires when initially experiencing a buddy robot, particularly in domestic environments, where the capacity to mingle and impart in a human-like manner and distinguishing the real necessity of the user by utilizing human like knowledge by recognizing user behavior without user to mention it, are crucial highlights to fuse keeping in mind the end goal to accomplish the coveted level of collaboration expected by people.

This Research Project was based on to achieve a robot companion which would identify the actual requirement of the user by using human like intelligence by identifying the user behavior without user to mention the requirement by himself and to have the ability to socialize and communicate with the user and with using these two major capabilities control the lighting level, temperature and humidity of the home which would aid the elderly people, differently abled people and also the normal family. This developed intelligent robot companion named HomeBot (HB) that has ability to control the ambient conditions of the smart home environment based on the detected user behavior for improved the user experience. In order to enhance the interaction ability the interaction between the user and the HB is integrated with a vocal interaction module.

After identifying the above mentioned current requirements which need to be addressed by an assistive home robot companion with the use of literature reviewing and brain storming sessions, HB was designed and developed which is capable of adapting the ambient conditions in accordance to the user behavior. The user behavior identification is facilitated by an artificial neural network that has been trained to detect the different postures of humans such as sitting and standing. Based on the identified user behavior, the robot controls the smart devices available in the home to realize the adaptation of ambient conditions. The smart devices and companion robot connect over a wireless network. Furthermore, voice interaction capabilities have also been incorporated to the robot companion to facilitate voice interaction based controlling of ambient conditions. A prototype of the system has been developed and the capabilities of the system have been validated experimentally.

A robot companion capable of providing assistance and companionship to the users of all kinds such as to normal family, elderly people or to differently abled people will be the new technology to be experienced by the people in near future. Hence a robot companion who has the capability to provide both assistance, companionship and to provide smart home appliance controlling to provide most suitable ambient condition in the home in accordance to the user behavior was the aim to be achieved by this Research Project. Identification more postures and gestures by training the developed neural network and there by providing more user experience are proposed as further improvements.

Keywords — Robot companions, Behavioral Pattern Recognition, Smart Homes

ACKNOWLEDGEMENTS

I would like to thank my project supervisor Dr. A.G.B.P. Jayasekara, Senior Lecturer and course coordinator Msc/PG diploma in Industrial Automation, Department of Electrical Engineering, University of Moratuwa and all the people who provided me with the facilities being required and conductive conditions for my project.

I express my warm thanks to Eng. D.D.A. Namal (Director General), Eng. D.M. Punchibanda (Director of Mechatronic Engineering Department), Eng. M.A.T.M.Kumara (Research Engineer), Eng. W.M.S.P.K.Wansinghe (Research Engineer), Miss. S.M Ediriweera (Technical Assistant), Mr. J.M.H. Udaya Bandara (Technical Assistant) and the staff of the Department of Mechatronic Engineering for their support and guidance at Department of Mechatronic Engineering in National Engineering Research and Development Centre of Sri Lanka.

I am using this opportunity to express my gratitude to everyone who supported me throughout the special studies project in Industrial Automation. I am thankful for their aspiring guidance, invaluably constructive criticism and friendly advice during the project work. I am sincerely grateful to them for sharing their truthful and illuminating views on a number of issues related to the project.

TABLE OF CONTENTS

List of Figures

List of Tables

1 INTRODUCTION

This chapter discusses the background of this project, its aim, objectives, and methodology with brief overview of each chapter.

1.1. Background

The elderly population around the world is steadily increasing. The number of people 60 years old and older increased to almost 900 million in 2015 and is forecasted to reach 2 billion by 2050 [1]. Elderly people are an important asset to society and should be cared for in a physical and social environment that is friendly for elderly people. In modern days, elderly people face the problem of feeling the loss of independence and loneliness though have the services such as adult day care, long term care, and nursing homes which provide the elderly with healthcare, nutritional, social, and other daily living support. It is more comfortable and easy to live in the surrounding that one has lived for many years than to adopt to live in a place where new relationships need to be developed from the start. This is more relevant to elderly people as they are having less physical capability to adopt for new environment of elderly care services. Hence, the most suitable way of providing elderly care would be to provide aiding facilities that elders required in the home itself. For that purpose, smart home concept and incorporating assistive robots for the aiding of elders can be used. By using these innovative concepts, elderly people would be able to live and enjoy their own home environment with an innovative smart home automation system which is capable of providing aid to day to day activities which elderly people are having hardship to perform. A combined system of an assistive robot and a home automation system will work in such a way that identification of the need of the user will be recognized by the assistive robot and intelligently recognizing the need of the user and then controlling of the home automation system by the robot for the required need of the user.

Apart from elderly people who will be benefited from an assistive robot companion which provides an autonomous home automation system for the user requirement, differently abled persons will also be find this unit to be of an essential device. If the assistive robot identifies the current home environment such as the temperature of the room, humidity of the room and the activity carried out by the differently abled person currently performing, and then provide information to those users in order to achieve the best comfort level in the room and control the light fittings of the room for the user need, then such a system will become a good assistance to those differently abled persons.

Domestic family will also be able to use such an assistive robot to ease the activities in the house. Sooner rather than later, robot buddies will be a piece of our typical family's day by day life attempting to help with local errands and dealing with us when required. Notwithstanding, before this turns into a reality, imperative issues should be tended to in the Human-Robot Interaction (HRI) field so as to accomplish social robots equipped for communicating with people also to which people associate with each other.

One of the greatest difficulties in this field is to endow robot associates with those social capacities expected to collaborate with a man amid a consistent period. These aptitudes can be upgraded by utilizing robot's own sensors or outer tactile frameworks introduced around the trial condition, so robots could know about the logical data. The fuse of these capacities hopes to enhance the association and the robot colleagues acknowledgment by people.

People have desires when initially experiencing a buddy robot, particularly in domestic environments, where the capacity to mingle and impart in a human-like manner and distinguishing the real necessity of the user by utilizing human like knowledge by recognizing user behavior without user to mention it, are crucial highlights to fuse keeping in mind the end goal to accomplish the coveted level of collaboration expected by people.

This Research Project was based on to achieve a robot companion which would identify the actual requirement of the user by using human like intelligence without user to mention the requirement by himself and to have the ability to socialize and communicate with the user and with using these two major capabilities control the lighting level, temperature and humidity of the home which would aid the elderly people, differently abled people and also the normal family. This developed intelligent robot companion named HomeBot (HB) that has ability to control the ambient conditions of the smart home environment based on the detected user behavior for improved the user experience. In order to enhance the interaction ability, the interaction between the user and the HB is integrated with a vocal interaction module.

1.2. Research Aim and Objectives

1.2.1. Aim

The main aim of this research is to devise methods to automate the activities associated with customized service delivery for smart home environments by exploiting the recent advancements in the field of social robotics and wireless sensor network.

1.2.2. Objectives

The main objectives of this research are:

1. To formulate the specifications and requirements of the configuration process, taking into consideration the needs of users, the demands of services and environmental conditions of smart home environment.
2. To design and implement a prototype stationary robot companion, incorporated with inhabitant behavioral pattern recognition system.
3. To design and integrate a vocal interaction system into the robot for improving the user experience.
4. To enable the adaptive home environment using smart objects and robot through wireless sensor network.
5. To assess and validate the efficiency and effectiveness of the proposed approach by implementing a domestic environment.

1.3. Methodology

- Detailed literature survey to found out smart home environment, robot technologies, artificial intelligence systems, wireless sensor networks and robot integrated smart home environments.
- Therefore, selected most suitable hardware systems, software algorithms, artificial intelligence method and wireless network topologies for developing of system.

- Preparing of data set including deferent human postures for tanning artificial neural network. It is used for developing of inhabitant behavioral pattern recognition system.
- Identified a suitable method and software algorithm for developing of voice recognition system and voice response generation. Hence, developing of vocal interaction system.
- Identified a suitable method of developing of wireless sensor network and smart object.
- Designed phenotype stationary robot companion and smart object using above designs
- Testing and validating capabilities of the overall system in domestic environment

1.4. Chapter overview

The project report has seven chapters. A brief description of each section is given below.

Chapter 1: This includes the introduction with the research gap, aims, objective and methodology.

Chapter 2: This includes the literature survey related to the concept of the smart home environment, applications in smart home environments, important of automatic configuration of smart home environments, Types of Automatic Configuration in Smart Home Environment, Automatic Configuration Systems Solutions, Knowledge Representation Technology Base On Smart Home, Smart home Systems Based on Autonomous Robotics Technology and Evolution of Robotics Technology the objective of this chapter is to identify available technologies in smart home systems and critically analyze the research gap.

Chapter 3: This includes the methodology regarding robot companion of adaptive home environment. It is containing the overview of the Kinect, Field of View, Software SDK of the Kinect, Image and voice streams selection for system, skeletal tracing, Mathematical formulation for finding joint angle of the human skeleton, preparing a dataset for Artificial Neural Network, preparing ANN for detecting human posture, designing of vocal interaction system, designing of wireless sensor network for communicating in-between robot and smart object and development of Robot Graphical User Interface (GUI). Finally design of the robot companion incorporated with above modules.

Chapter 4: This chapter includes the design of the prototype robot companion with integrated modules. Hence, prove that the intelligent control and output result in domestic environment and findings (From chapter 2 and 3) and discusses the variation between design and actual data.

Chapter 5: This chapter includes the overall conclusion of the thesis with the recommendations for future works.

Chapter 6: This includes the overall references, which used to fulfill the research gap of this project.

2 LLTERATURE REVIEW

This chapter discusses the overall literature review pertaining to the smart environments and robotics technology.

2.1. The Concept of the Smart Home Environment

The smart home environment is a growing concept driven by advances in technologies in the modern world such as the social robots, general computing, Internet of Things (IoT), data mining and the increased availability of various type sensors and actuators in the market [3]. The conventional home environment is often unable to fulfil its main role of supporting various human activities because of its inflexible and static nature: it offers very limited flexibility and adaptation to deal with the various needs and activities of its users. To overcome these limitations, the concept of the smart home environment has been proposed to improve the quality of life by enhancing users' daily activities, supporting their various needs automatically, by altering its behavior without deliberate human intervention as well as facilitating independent living for their special needs such as elderly people, disabled people, and patient [4].

The smart home is also known as the adaptive home or automated home or intelligent home. It is a digital, adaptive environment, sensitive and responsive to inhabitant needs [5]. It is explicitly defined by Aldrich [6] as "a residence equipped with computing and information technology, which anticipates and responds to the needs of the occupants, working to promote their comfort, convenience, security and entertainment through the management of technology within the home and connections to the world beyond". Smart home is described as conjuring up the idea of an imminent future in which people are surrounded by very many fine-grained distributed networks [7]. These networks consist of sensors, actuators and various computational electronic devices that are inconspicuously used to enhance daily objects such as clothes and furniture, thus collectively constitute digital environments that are adaptive, sensitive and responsive to the inhabitants' presence [8].

Thus, the smart home environment should be able to collect and analyze information about residents' daily lives and use it to enhance their living environment. This entails that it flexibly and securely controls home appliances, monitors residents' health status, advises them on actions (in case of any irregularities occurring) in the home environment, and enables users to easily access the required information [9]. Recent advances in sensor technology, embedded systems and robotics indicate that the realization of the smart home concept is now practicable, particularly if essential security and privacy concerns are addressed [10].

One of the crucial challenges faced by smart homes is to provide intelligent support reliably within the home environment. Intelligence is an important feature of the smart home. It refers to self-awareness, problem-solving and the ability to memorize, learn, understand, plan and reason about the environment [9]. The smart home is required to be imbued with an awareness of its physical context (e.g. temperature, lighting, house layout), its occupants' context (e.g. preferences, location, activities), as well as temporal context (hour of day, day, week, season, year).

Such an environment imbued with context-aware reasoning makes it conceivable for users to obtain customized services such as temperature and lighting settings based on their preferences and the monitoring of their health status [11]. In the smart home environment, computer software acts as an intelligent agent capable of perceiving the states of the physical context and its users by means of sensors, reasoning about these states using AI techniques, making decisions and taking actions to accomplish certain goals, such as maximizing users' safety, maintaining their comfort and minimizing consumption of resources [12].

2.2. Applications in Smart Home Environments

Research efforts to make the environment 'smart' have been actively carried out across various domains under the umbrella of Ambient Intelligence (AmI) [13]. A smart home is an example of an environment enhanced with AmI [14]. AmI builds on the idea of general computing to deliver services which are imperceptibly integrated into human's lives through connected devices that are progressively embedded into the environment [15]. Typical service applications in the home environment are energy management and health monitoring [16].

Energy management is no longer a luxury in the domestic environment, but essential for normal householders, who are beginning to seek easy ways to reduce their energy consumption because of high energy prices and environmental concerns [17]. Smart homes equipped with smart energy management automate energy conservation activities to reduce consumption without compromising users' quality of life, by easing their operational burden [18]. Energy management systems within smart home environments have attracted much research interest as an application of smart grid technologies, to extend their capabilities (e.g. automation) into the home [17]. The smart home environment plays a vital role in the interaction between the grid and the consumer [19]. Automation technology is a key feature of the smart grid which enables utility companies to adjust and control each individual device from a central location [20].

The smart home also facilitates the assessment of its users' health [21]. For instance, researchers have tried to create a linkage between alterations in users' motion patterns and the commencement of dementia symptoms, benefiting from motion sensors and the evaluation of specific parameters such as walking speed and distance covered [11]. Smart home technologies have also been used for early-childhood autism screening and performing accustomed routines for initiating new health behaviors.

The smart home provides essential infrastructure to enable healthcare services to deliver care to people in their own homes. Lifestyle monitoring, which is a typical telecare application, potentially offers a new technique for providing a safe assistive home environment for old and vulnerable people in case unexpected events occur. It is based on the potential to determine and manage peoples' care or health status through remote monitoring of their behavioral characteristics and parameters concerning their interactions with and within their local environment. Thus, any changes in peoples' normal activity over time, detected through sensors embedded in the home environment, can be recognized as unusual and responded to [22]. In addition, computer algorithms have been generated to envisage and detect users' activities in the home and to identify their behaviors, emotions and even gestures.

However, argue that there is little comprehensive understanding of how to deliver complete and efficient lifestyle monitoring systems, notwithstanding the great numbers

of marketable worldwide installations. In other words, although all these health monitoring technologies offer considerable opportunities for the smart home, they also come with some risks and challenges concerning users' privacy and security.

Many people are still hesitant to use sensing technologies in their homes, being afraid of allowing others to monitor their digital trails and make use of them [11]. However, they might be well received if they are properly introduced and used, cost-effective and properly resourced. In addition, such monitoring might be acceptable if it enhances peoples' feelings of security and safety in the home, increases the available care options and supports the carers' role [23]. Thus, an effective realization of the smart home environment essentially requires such challenges to be addressed.

To dispel users' concerns, researchers should address such challenges by providing alternative ways to identify users without compromising their privacy and safety, such as by using digital representations with users' consent, instead of using video footage. Users should be ensured of their decisive authority to control the system by imposing restraints to prevent their home from making undesired decisions and taking harmful actions.

This further indicates that it is essential to benchmark the attitudes and opinions of telecare users, informal carers and care professionals before introducing any telecare system to their environment. Research shows that the deployment of telecare is expected to yield important consequences [24]. Examples of these consequences are electronic monitoring of patients, accurate and secure electronic patient records, reduction of cost and time by providing virtual consultations, thus eliminating the need for patients to go to hospital, as well as fewer admissions to residential care units [25].

Both energy management and health monitoring applications have attracted considerable attention from academic and industry researchers, mainly because of increasing healthcare costs and energy consumption, and recent technological advances in miniature devices, smart textiles and wireless communications [11] However, the existing smart home systems are not intelligent enough, for three main reasons:

The wide availability of various services and devices in contemporary networked smart environments makes the management and deployment of dynamic smart home systems difficult without the intentional involvement of human actors, because devices have been innovatively used beyond their original design across several application domains (e.g. energy consumption for daily activity monitoring) and because different types of devices can achieve the same functionalities (e.g. user location can be obtained via GPS, RFID, Wi-Fi)[12]. It has also been observed that the development of smart home services has begun to decouple from the development of smart home devices. Rather than being pre-bundled with smart home devices, smart home services have been developed and deployed independently, with the consequence that the systems demand significant manual works to create efficient linkages between services and appropriate devices, because of the large number of possibilities and combinations [12].

These service applications require a large number of sensors and actuators. In the context of the IoT, there will be even more of these, coexisting within one physical area (e.g. the home). Several sensor types (e.g. PIR) are found to be used across application areas such as home security, telecare and energy management services. The same type of sensor and actuator is also used to detect and create different phenomena, depending on its context and application area. Different types of sensor and actuator are used to detect and create the same phenomena, thus augmenting the complexity of smart home systems.

These systems lack the elasticity and scalability needed to deal with users' different requirements, needs and changing states[26]. They are unable to deal efficiently with dynamic contexts, where devices may be added or removed during runtime. This highlights the need for an open, intricate, agile and flexible architecture to deal efficiently with all the changing circumstances[4].

In summary, a large number of possible combinations of services and devices, as well as diverse user preferences, have made the management and deployment of dynamic systems difficult without the intentional involvement of human actors. This has led to interoperability issues and prevents various smart home components from communicating with each other unless certain gateways or adapters are utilized. This has in turn highlighted the necessity of having auto-configurable systems to connect the correct devices to application services and deliver smart home services appropriately and autonomously.

2.3. The Important of Automatic Configuration of Smart Home Environments

Automatic Configuration has become essential in any smart home environment, due to the complexity of the interactions among applications and devices [27]. This complexity and the dynamic nature of smart home environments make it impossible, at the time of design, to take account of which components will interact in what ways in the environment. Thus, it is essential that they can be configured during the deployment of the system, or auto configured at runtime. Besides changing devices and/or their parameters, one important aspect of configuration is to adjust the connections among them using multiple communication options such as Ethernet, Wi-Fi and power-line, in line with the actual tasks to be undertaken or unexpected changes or failures in the physical environment.

The complexity of such management tasks is beyond the capability of most households. Thus, automatic configuration becomes essential to ensure the flexibility and robustness of smart home environments. In detail, it will be actively necessary to identify appropriate components for service applications and create appropriate connections in each situation without direct human intervention.

Compared to manual configuration, which requires intensive human effort and knowledge, in auto-configuration the correct interconnections among the components are decided automatically. Automatic configuration has also been defined as the ability to adjust the system's configuration dynamically to adapt to altered circumstances, hence to enable the adding, removing or modifying of entities, without interrupting the normal operation of the whole service [28]. Thus, significant efforts are required to maintain linkages between services and devices so as to cope with any changes of requirements of the target environment. This in turn requires considerable expenditure of money, effort and time for experts to maintain such complex systems.

Most of the endeavors to make environments 'intelligent' have addressed mainly the technical features of building elements, but lack of studying on the dynamic interrelationships between the designed environment and its users. This has in turn made it difficult to address communication problems among users, their activities and smart systems, which may cause user dissatisfaction [29]. This does not achieve the flexibility and agility of smart home systems, which are intended to adapt to changes in environment and users' requirements.

Overall, the abovementioned issues have stimulated the need to design dynamic auto-configurable applications to provide communication models among these distributed systems autonomously. Recent research in smart home technology aims to enable interaction between smart home devices and the networking infrastructure with no obvious user control [30].

The smart home systems must help the ubiquity of their services' task. This implies any smart home systems should dependably be accessible, despite any adjustments in the service environment. All the more particularly, a smart system should be naturally interoperable with assets in its present service environment, instead of statically pre-modified for its condition. Manual configuration requires major inputs of money, effort and time for experts to maintain such complex systems. The smart home vision highlights the need of automatic configuration in smart home environments by indicating that users are not meant to manage the smart home system themselves, as professional administrators do with conventional distributed system [12].

The integration of automatic configuration technology into smart homes ensures peace of mind, increased comfort, health monitoring, safety and security, as well as economic benefits (e.g. lower energy consumption) [5, 3].

This literature review shows that the full potential of smart home environments has not yet been realized, due to the complexity of the essential automation systems [4, 12, 15]. Achieving the automatic configuration of smart home systems is a complex task [31]. Adjusting the smart home system accurately to its users’ requirements requires profound expert knowledge. Thus, [32] argue that much work is needed to establish the efficiency of automatic monitoring functions and automatic adaptation to users’ changing needs.

To deal effectively with these issues, smart home environments require automatic configuration to enable services to utilize any required device through any network within a given smart home environment, automatically, anywhere and at any time. It helps to deal with the intricacy of the interactions among smart home applications as it reaches levels beyond the human ability to control them, while ensuring the required quality of flexibility and robustness of smart home systems. This would allow them to integrate and cooperate continuously in order to meet inhabitants’ objectives and fulfil their changing needs throughout the environment. Overall, this will serve the global trend towards providing flexible and agile services or products for users at low prices and at the appropriate times [8].

2.4. Types of Automatic Configuration in Smart Home Environment

Research into automatic configuration was mainly carried out in the web services area, followed by work on AmI and distributed robotics, either with help from web services or directly employing traditional AI techniques [33]. Within these research areas, the term ‘configuration’ refers to patterns of cooperation, collaboration and information exchange amongst multiple software components to learn and reason about the environment in order to execute actions when none is predefined [34-36].

Automatic configuration systems help to minimize the overheads of the existing configuration process with minimum user intervention [37]. Achieving auto-configuration in smart home environments could fulfil users’ need for an easy and flexible environment.

In the areas of AmI and network robotics, the concept of configuration is generally employed to refer to organizational and/or structural variance. Achieving configuration is a key function of an autonomic system, defined as “a system that operates and serves its purpose by managing itself without external intervention even in the case of environmental changes” [38].

2.5. Automatic Configuration Systems Solutions

The evaluation process adopted here focuses on the ability of configuration approaches to meet the requirements of smart home environments and cope with their dynamics. The success of automatic configuration will enable the smart home environment to adapt quickly to changes in users’ needs in order to provide a comfortable and adaptable environment.

It is expected that the key gains from employing automatic configuration systems will be in the form of enhanced performance and reduced operating cost. The auto-configuration features will enhance the performance of the smart home system, thanks to the better adaptation to changing system characteristics. Thus, assessment criteria will focus on the methodologies to evaluate the gains that can be attained using auto-configuration systems: flexibility of automating smart home systems online with minimum or no user intervention and enhancing the overall performance of the smart home system.

Research has been done to achieve automatic configuration by concentrating on knowledge representation technology [30, 39-42] and autonomous robotics technology [33, 43-45]. The criteria and performance metrics for the assessment are flexibility, reconfiguration type and configuration executor. Flexibility indicates that the system deals elegantly with the complexity of smart home environmental conditions [46]. Since the aim of this research work is to achieve automatic configuration within smart home environments without user intervention, we have added three other criteria: on-line configuration; the targeted environment to determine the domain of automatic configuration (whether it is a smart home or not); and the configuration executor, to determine whether the user should interfere to support the configuration process or whether it is a wholly machine-based process.

Although the user is certainly the final determinant of what the system does and how it does it, minimum or no user intervention is preferable, particularly in the case of dementia patients, for the sake of making their lives easy. The transfer of responsibility for an individual’s wellbeing to software entails adopting safe, reliable and secure automatic configuration techniques. Appropriate levels of user privacy should be established, taking into account what is considered appropriate from their point of view.

The identification of the criteria was based on the main aim of this research: to achieve automatic configuration within a smart home environment at runtime with minimum user intervention. Previous work is analyzed in light of these criteria to determine whether these benchmarks were achieved or not. Fulfilling the smart home requirements will help to make the whole system more agile. Such auto-configuration ensures the up-to-date and smooth functioning of the smart home environments. This seamless operation of the smart home system leads to a calm home environment, which enhances the quality of the services provided within smart home environments. This in turn satisfies users’ needs and reduces the overall system cost. All of the studies discussed above have been analyzed in terms of these benchmarks. The evaluation process adopted was also used for subsequent evaluation of the proposed approach. In general, automatic configuration studies can be categorized, according to the enabling technology employed, as follows:

2.5.1. Knowledge Representation Technology Base On Smart Home

Service-Oriented Context-Aware Middleware (SOCAM) architecture is proposed for the building and rapid prototyping of context-aware services [42]. SOCAM provides an effective support for discovering and accessing numerous contexts to construct context-aware services. A formal context model based on ontology using the Ontology Web Language (OWL) has been built to tackle issues such as context classification, semantic representation, context reasoning and dependency. However, test results indicate that the ontology reasoning time is much longer than the user-defined rule-based reasoning time.

An architecture for automatic configuring system has been constructed using knowledge representation technology [41]. It was quite easy, taking a model-driven approach, to transform the behavioral models and message sequence charts into a framework of executable code to which the more specific functionality (including execution of the OWL-based framework of the SW (OWL-S) ontology) could be added. However, some of the subtests failed, so the architecture needs to be improved.

A new proposed approach to achieving fast reconfiguration of modular manufacturing systems relied on an ontology-based reconfiguration agent without human intervention [40]. However, the problem-solving skills of the proposed agents would need to be improved to solve more intricate problems. This could possibly be achieved by extending the reasoning engine architecture or adding new components which could use entirely decidable reasoning capabilities and ontology web language description logic (OWL DL).

Autonomic communications architecture has also been proposed to solve the automatic configuration problem. Using the OSGi framework, an autonomic context-aware element was developed in order to identify and personalize the service offer for a particular user [47]. The autonomic element senses devices linked to the home network considering user preferences. The services are modelled in OWL ontology.

The autonomic element also uses Semantic Web Rule Language (SWRL) reasoning to infer suitable services and offer a personalized service to each user [48]. Although the autonomic element has provided successful results, it must evolve to a more mature state to deal with more complex issues. An ontology-based infrastructure called Sixth-Sense has been proposed to facilitate quick prototyping of artefact-related applications [49]. This approach is based on Semantic Web technologies including ontology to represent significant aspects artefact-human communication and reasoning of high level context from data collected by sensors. Although experienced users found it easy to customize the ontology, it was difficult for less experienced users.

Another novel approach called the Smart Home Ontology Model has been proposed to build a smart home in conjunction with an autonomic system whose aim is to monitor the home and its residents [39]. In particular, it aims to provide the elderly with intelligent support and assistance in any circumstances at any time using wireless sensor technologies to control devices and other smart home components as required. However, this model needs to be enhanced to make reasoning components more complete and to provide more intelligent decisions and actions.

Shen et al have proposed an ontology-based approach to represent Product Extension Services (PES) knowledge configuration and developed a PES configuration system [50]. This was achieved by implementing efficient configurations through unambiguous sharing and reuse of knowledge. Nevertheless, this approach can be considered only a partial solution to the configuration problem, as configuration optimization and reconfiguration were not considered.

An Autonomic Ontology Driven Architecture (AODA) has been proposed for the automatic configuring of resources, utilizing service and event-oriented communications [51]. The AODA follows a top-down approach and is implemented as an ecosystem-wide ontology intended to characterize the properties of services and events relevant to producers, entities and consumers, and to participate in the dynamic collaborative environment. However, more validation of the AODA in more complex scenarios is required to manage energy consumption in the smart metering business case, as it involves a larger number of connected heterogeneous machines.

2.5.2. Smart home Systems Based on Autonomous Robotics Technology

Recent research has investigated the principles and techniques needed for automatic configuration within the field of autonomous robotics. McKee has proposed a novel Task-Directed Configuration of Networked Robotic Agents [43]. This model integrates the concepts of a task factory, which produces task modules, and a set of modules demonstrating robotic components distributed in the environment, such as sensors and actuators. It specifically focuses on automatic configuring networked robotic agents where robotic components are extracted from a common pool and assembled in relation to a high-level task description. In this method, ad hoc robotic architectures are automatically created with just the components needed for each specific task. Task descriptions include information about the kinds of components required, about their physical location and about their requisite connectivity. However, this model needs to be further improved to develop the capability to form more intricate robotic agent architectures.

A plan-based approach is proposed to control the smart environment and support interaction between it and its users [52]. Users in this approach need not learn how to operate all devices and control their functions, but simply communicate their needs to the environment. Planning is utilized to develop strategies on the ways in which each function can be performed on each device to accomplish the users' defined goal. This approach is very similar to traditional action planning, rather than configuration planning. Automated Synthesis of Multirobot Task Solutions through Software Reconfiguration (ASyMTRe) is proposed to increase the autonomous capabilities of heterogeneous robot systems [44]. It enables a robot coalition to connect schemas dynamically within and across robots to perform a single-robot task by coalitions of multiple robots. The approach is based on a centralized ASyMTRe configuration algorithm and the ASyMTRe-D negotiation protocol. The distributed negotiation process provides a more flexible and robust method for establishing coalitions. On the other hand, it introduces a trade-off between robustness and solution quality. A reactive approach to auto-configuration based on the ecology of physically embedded intelligent systems (PEIS Ecology) has been proposed by [33]. This approach employs SW services to independently generate a configuration to execute a cooperative navigation task, automatically changing this configuration when any component fails. One disadvantage is that it may produce non-optimal configurations. Another is that it may fail to identify a configuration which exists.

Lundh have proposed a plan-based approach to automatically generate a desired configuration of a robot ecology, set of resources and environment [53]. In particular, hierarchical task planning was applied to configuration generation [54]. Configuration methods have a body which lists all the functionalities which comprise them. These functionalities are the system's basic components, defined by their functioning parameters, their input and output signals, and the preconditions which must hold in the ecology for their correct functioning. Configuration planning is achieved by a best-first search of the set of functionality instances which conform to the functionality of the body of each configuration method. This type of search allows activation of the configuration with minimum cost for every planned action. The state of the ecology is automatically attained at design-time and is also monitored during implementation in order to reconfigure it if any functionality fails. They have validated their approach by showing that a scenario which was previously hand-coded can currently be run autonomously in the adopted PEIS-Ecology testbed.

2.6. Evolution of Robotics Technology

During the last half century, in response to the development of the social and commercial needs of human beings, robotics technology has undergone a profound revolution which can be represented in terms of three generations [55]. A robot is defined as "an autonomous machine able to process information elicited from sensors, upon which it can make its own decisions and act in the environment" [56]. Autonomy in robots is defined as "the capacity to operate in the real-world environment without any form of external control, once the machine is activated and at least in some areas of operation, for extended periods of time" [57].

The first generation was dominated by industrial robotics (e.g. manipulators performing tasks such as assembling, polishing, welding and painting) and the second by service robotics (personal robots, domestic robots, medical robots, humanoid robotics etc.), leading to the current exploitation of ubiquitous robotics. This third generation is the outcome of developments merging ubiquitous computing and robotic technologies [58]. Industrial robots were introduced into production lines in the early 1960s and remained dominant until the 1990s. The Robotic Industries Association defines an industrial robot as "an automatically controlled, reprogrammable, multipurpose manipulator programmable in three or more axes which may be either fixed in place or mobile for use in industrial automation applications" [59]. The first one was manufactured by Unimate and was installed by General Motors in 1961.

They were used to free human beings from hazardous and harmful tasks, to enhance the speed and accuracy of tasks such as painting and welding, and to reduce the cost of various manufacturing processes. The industrial robot is pre-programmed and has relatively simple tasks such as manipulating and moving objects and cooperating with the environment [60].

However, it has been recognized that there is a need to develop more flexible robotic systems for use in unstructured environments. Compared with industrial robots, service robots take advantage of recent advances in mobility, perception and algorithmic research, which have the control, sensing and decision-making abilities that are necessary for them to work in unstructured, three-dimensional environments. These features enable robots to localise in two-dimensional maps of the world and to navigate unknown two-dimensional environments. Since 1995, developments in service robotics have allowed the construction of animal-like robots and mobile robots. Figure 2.1 illustrates the evolution of robotics over the last five decades.

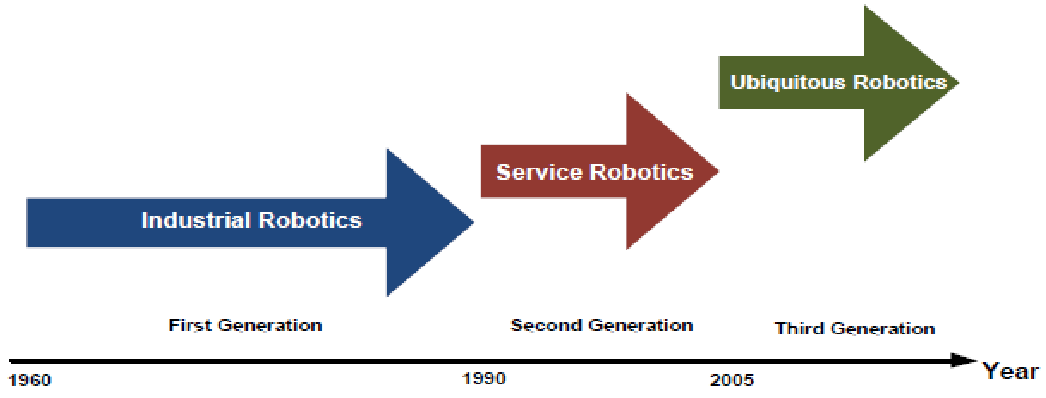

Figure 2.1: Evolution of Robotics

These technological advances and increasing social demands are the main driving forces of new applications emergence such as surgery assistance, automatic refuelling and rehabilitation, home cleaning and museum exhibitions [60]. The International Federation of Robotics identifies a service robot as one which operates fully or semi-autonomously to deliver services useful to the wellbeing of humans and equipment [61]. Service robots can execute daily tasks, assist people with disabilities and work as companions or pets. Other types are surveillance robots, military robots, security robots, construction robots, field robots, medical robots, rehabilitation robots, and office and tour guide robots.

Ubiquitous robotics technology is the latest generation robotic technology utilising the ubiquitous devices (e.g. sensors, actuators) embedded everywhere in the inhabited environments and even inside human bodies via networks. Through these devices, the ubiquitous robots can gain better picture of the world and manipulate the world beyond its own capabilities [62]. The ubiquitous robotics revolution has been accelerated by the development of autonomous robotics and intelligent environments, giving rise to the multidisciplinary research field of intelligent robotic environments or ubiquitous robotics, which merges ubiquitous computing and ubiquitous environments with robotic technology [63].

It can be observed that the robotics revolution is driven by the more and more complex tasks and higher expectation such as able to communicate autonomously with users, understand their requests and provide them with services accordingly. The usage of knowledgebase has increased with each robotics generation [64]. Knowledge extraction, representation and usage can enable a grounded and shared model of the world appropriate for high-level tasks [65]. The main challenges are how to extract knowledge from natural languages, to benefit from several areas of knowledge in decision making and to recognize what knowledge is missing.

Bearing in mind that it would be virtually impossible to equip a robot with completely comprehensive knowledge before it was put to use, robots need to be developed to be able to acquire missing knowledge automatically from somewhere at runtime, to achieve the tasks set by users. Thus, the DARPA Robotics Challenge (DRC) aims to improve semi-autonomous robots which are able to perform "complex tasks in dangerous, degraded, human-engineered environments" [66].

Robots should also be able to use various types of knowledge, since they might need to encompass several areas from more than one knowledge resource to achieve a single task. They should be able to identify gaps between the implicit forms of knowledge and experience. They already possess and that required for the task at hand, then be able to search for the pertinent knowledge from open sources.

These requirements have stimulated continual research efforts to develop intelligent robots, which are a part of ubiquitous robotics technology. In this respect, cognitive systems make an on-going contribution to increasing robots' cognitive capabilities, enabling ubiquitous robots to automatically acquire and utilize shared knowledge and to reason based on decision-making mechanisms to achieve runtime tasks. Thus, ubiquitous robots are able to collectively comprehend the users' needs or preferences, even when no direct orders are given, and provide a constant and seamless service [67].

2.7. Summary of Literature Review

The approaches reviewed have some similarities with smart home concept. This section shows that the research can benefit from ubiquitous robotics and knowledgebase technologies in the home environment. It can be considered an effective approach to achieving automatic configuration, because it is able to intermediate between services and users in order to satisfy their individual needs. It could satisfy users' needs by positively enhancing and facilitating their home activities without disturbing their peaceful lives or intruding into their privacy. It helps smart home applications to provide users with the necessary services, accurately and inconspicuously, when and where needed.

Although the approaches reviewed above have accomplished good results, none of them has met all the requirements of the smart home environment; each fails to meet one or more of them. As shown in Table 1.1, if the approach meets the flexibility criterion, it either requires user intervention or is not conducted at runtime. In other words, there is no adequate solution to allow the smart home system to adapt dynamically to changing circumstances, hence to enable the correct interconnections among its components without user intervention and without interrupting the whole service. In short, the above-mentioned technologies and approaches have not yet evolved to a sufficiently mature state.

To bridge these research gaps, we propose a Human Behavior Pattern Recognition robot companion for Smart Home Environments as a way forward to enable automatic configuring systems. It is incorporated with vocal interaction system for improving user experience. The powerful features of the robot companion are described in detail in later chapters. The robot companion to perform automatic configuration and meet the requirements of the smart home.

The proposed approach should also be able to automate the activities associated with customized service delivery for dynamic home environments, reliably and precisely, to facilitate context-aware services.

Table 1.1: A Comparison of Configuration-related Research

Technology	Reference	Approach Name	Flexibility	Configuration Type (Online)	Targeted Location. (Home)	Configuration executor (No Direct User Intervention)
Knowledge Representation	[42]	"A service-oriented middleware for building context-aware services"	✓	X	✓	X
	[41]	"A Semantic Web-driven approach to self-Configuring Computer Systems"	X	✓	✓	✓
	[40]	"An ontology-based reconfiguration agent for intelligent mechatronic systems"	✓	X	X	✓
	[45]	"Dynamic auto-configuration of an ecology of robots"	X	X	✓	✓

Knowledge Representation	[47]	“An autonomic approach to offer services in osgi-based home gateways”	✓	✓	✓	X
	[49]	“An Ontology-based Programming Platform for Smart Artifact Systems”	✓	✓	X	✓
	[39]	“POSTECH's U-Health Smart Home for elderly monitoring and support”	✓	✓	✓	X
	[50]	“Configuration of product extension services in servitization using ontology”	✓	X	X	✓
	[51]	“AODA: An Autonomic and Ontology-Driven Architecture for service-oriented and event-driven systems””	X	✓	✓	X

Robotics	[43]	“Task-directed configuration of networked robotic agents”	✓	X	X	X
	[52]	“Smart environments and self-organizing appliance ensembles”	X	X	✓	X
	[44]	“Building Multirobot coalitions through automated task solution synthesis”	X	✓	X	X
	[35]	“Reactive auto-configuration of an ecology of robots”	X	✓	✓	✓

3 HOMEBOT: ROBOT COMPANION FOR SMART HOME

This chapter discusses the methodology regarding the robot companion for an adaptive home environment. In fact, it will be discussed why each module such as of the robot complain is selected, what they give as the results, how the results are merged to get the desired outcome and finally how the robot companion is developed using these modules.

3.1. HomeBot: System Overview

HomeBot (HB) is a stationary robot companion with facilitate to auto-configuration of home appliances using behavior pattern recognition in inhabitant. Inbuilt vocal interaction system (voice recognition and Voice responds) used to improve the robot capabilities and it is resultant to enable a user-friendly environment of the humans. Overview of overall system is depicted in Figure. 3.1 in a modular format.

Smart Object is other device of the main system. It used to control home appliances according to commands of the robot. Wi-Fi enable warless sensor network has provided the communication link in-between robot and smart object.

Low power ARM (Advanced RISC Machines) Cortex-M processer has used to develop Smart object and Inbuilt Wi-Fi stake of the processor has used to develop the warless communication facility of the smart objects. All hardware of the robot is embedded with high power single board computer. It is powered by Intel Atom base CPU with x86 architecture. Main processing unit is used to process the software algorithms, data handling and communicate with smart objects for controlling home appliances through Wi-Fi network.

The voice and image streams of the user is obtained by using Microsoft Kinect sensor. A skeletal information of the user is derived by using of Kinect image streams and Kinect software SDK (Software Development Kit). Microsoft Cooperation has developed the SDK for Kinect sensor. The joint coordinates of the human skeleton are obtained by using Vector algebra. Joint coordinates are used to develop behavior pattern recognition system.

Pattern classification model has been implemented with Artificial Neural Network (ANN) for detecting and classification of user behavior patterns. Various types of posture patterns of the humans have been used to train the ANN. Large amount of data is used to improve accuracy of ANN. All the learned data of ANN is stored in an inbuilt database of the system.

.

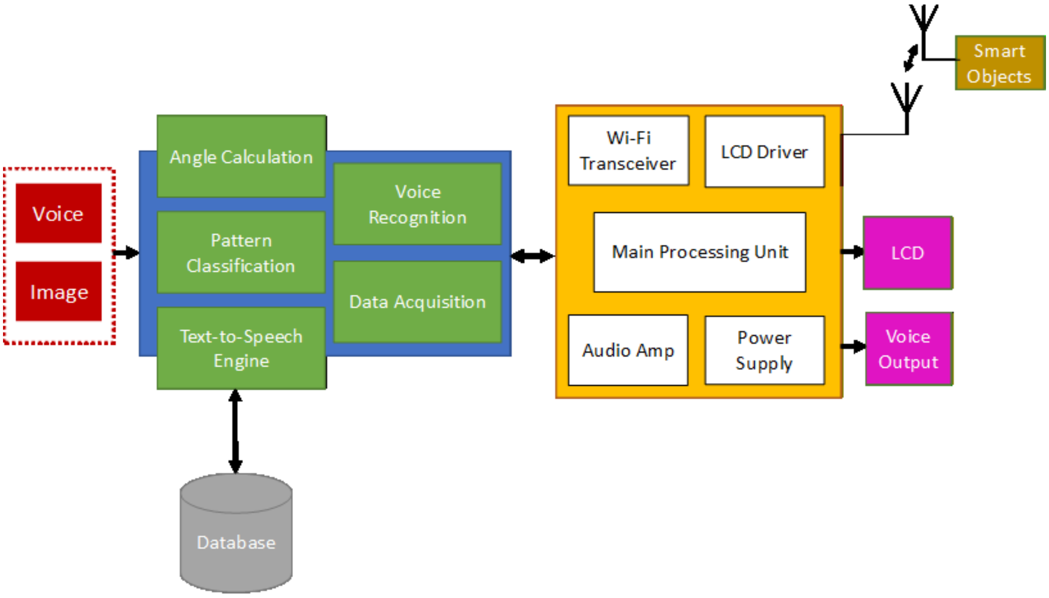

Figure 3.1 – Overall functionality of the system of robot control module

Voice recognition and understanding units of the system convert speech into text strings and identify various types of voice of deferent humans. Microsoft Speech Recognition is a speech to text converter and it is used to implement the Voice recognition and understanding unit.

Converted text output of the speech is tokenized and verified using the stored speech grammar in the memory. Kinect voice stream is used for developing of voice recognition system. Voice response generation module has been implemented using Microsoft Speech API.

Data acquisition unit is a software to hardware connector and it has been developed by using a software algorithm written with C# language. It facilitates the processing of output data in a manner which can be sent to the hardware units.

Audio amp amplifies the voice output and LCD driver drives the LCD that displays the basic information of the system. Power supply is used to provide the power to all the circuitries.

Overall architecture of the robot companion has been described above in brief. Some of the modules of this architecture are driven by number of software coded algorithms. Key modules of this architecture such as Kinect Sensor, Kinect skeleton tracking, Graphical User Interface (GUI), Hardware platform, Behavioral Pattern Recognition System, Vocal Interaction System and Wireless Sensor Network will be discussed further under sub topics.

3.2. Kinect Sensor

The Kinect sensor has been produced and patented [67] by Microsoft Company initially under a venture called Natal in 2006. The goal to make a progressive amusement controller for Xbox 360 was started by the divulging of the Wii support at the 2005 Tokyo Game Show meeting. The reassure presented another gaming gadget called the Wii Remote, which can identify development along three tomahawks and contains an optical sensor that identifies where it is pointing. This initiated the Microsoft's Xbox division to begin on a focused gadget, which would outperform the Wii. Microsoft made two contending groups to think of the planned gadget: one working with a PrimeSense innovation and other working with innovation created by an organization called 3DV. In the end, the last item has been named Kinect for Xbox 360 and was based on the PrimeSense's profundity detecting innovation.

As of now, Microsoft offers two variants of the Kinect gadget. The first, Kinect for Xbox 360, is focused on the excitement with Xbox 360 reassure and was propelled in November 2010. After the Kinect was hacked and numerous different applications spread through the Internet, Microsoft saw the presence of a radical new market. Based on this discovering Microsoft outlined a moment rendition of the sensor, Kinect for Windows, directed on the advancement of business applications for PC. In fact, there are just slight contrasts between the two renditions; nevertheless, the official Software Development Kit (SDK) from Microsoft limits the help of Kinect for Xbox 360 for advancement as it were. The most essential contrast between Kinect for Xbox 360 and Kinect for Windows is particularly in an extra help of profundity detecting in close range that empowers the sensor to see from 40 centimeters remove rather than 80 centimeters.

3.2.1. Inside the Kinect

The Kinect gadget is in view of a depth sensing technology [68] that comprises of an Infra– Red (IR) camera and IR producer situated in a specific separation between them. The rule of the depth sensing is a radiating of a predefined design by the IR producer and a catching of its reflected picture that is distorted by physical items utilizing the IR camera.

The processor at that point looks at the first example and its distorted reflected picture and decides a profundity based on 7 varieties between the two patterns. The subsequent profundity picture has a level determination of 640 pixels, vertical 480 and profundity determination of 8 meters partitioned by millimeters. The gadget is also furnished with the shading (RGB) camera with up to 1280 960 pixels' determination, which might be utilized as another information hotspot for acknowledgment. Other gadget's part is a multi– exhibit mouthpiece for spatial voice contribution with capacity to perceive a course of a voice source. The gadget's tilt edge is conceivable to set utilizing an engine in go from - 27 to 27 degrees which expands a last vertical sensor's field of view. Moreover, the gadget contains a 3–pivot accelerometer utilized for deciding a gadget's tilt edge yet it can be utilized for extra further applications. Figure 3.2 depicts a format of the Kinect's segments.

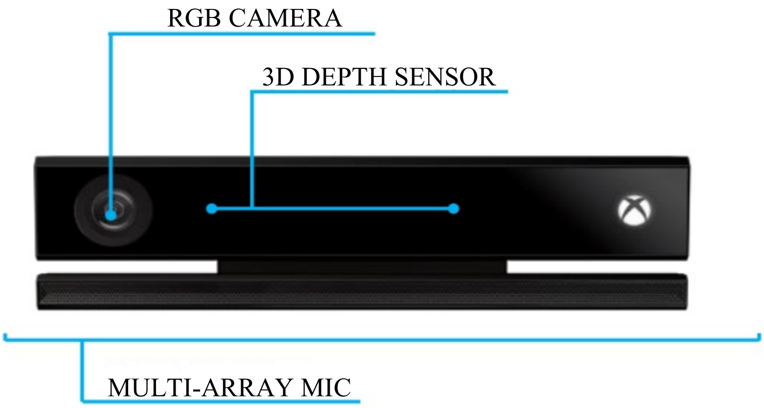

Figure 3.2 – Kinect for Windows sensor components.

3.2.2. Field of View

The Kinect sensor works from numerous points of view comparatively to a camera; it likewise can see just a restricted piece of the scene confronting it. This piece of the scene that is obvious for the sensor, or camera mostly, is called Field of View (FOV) [68]. The accompanying vertical and even points depict the sensor’s FOV for both profundity and shading camera in [68]. The flat point is 57.5 degrees and the vertical edge is 43.5 degrees.

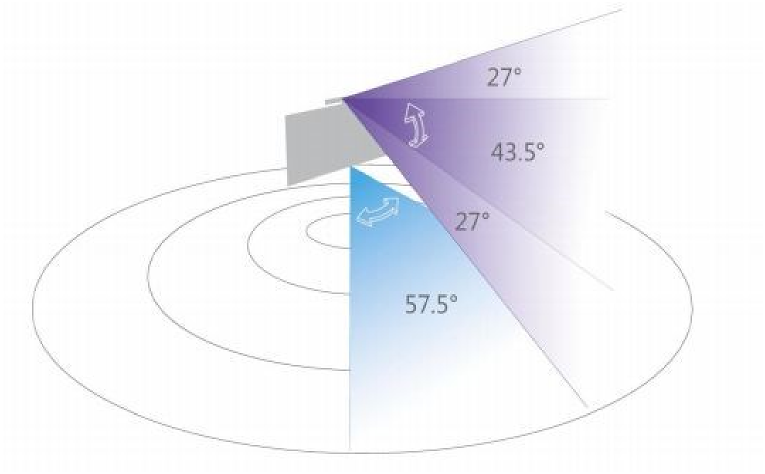

Figure 3.3 – Kinect for Windows sensor field of view

The vertical point can be moved inside range from - 27 to +27 degrees here and there by utilizing the sensor tilt. Furthermore, the depth camera is restricted in its view remove. It can see inside range from 0.4 meter to 8 meters yet for the commonsense use, there are prescribed esteems inside 1.2 meter to 3.5 meters. In this range, the articles are caught with insignificant twisting and negligible clamor. The Figure 3.3 shows the sensor's FOV.

3.2.3. Microsoft Kinect for Windows SDK

Microsoft distributed an authority SDK after it had understood the Kinect's potential in opening another market. The SDK bolsters an improvement in C++, C#, VB.NET, and other .NET based dialects under the Windows 7 and later working frameworks. The most recent rendition of the SDK is accessible for nothing on its official site [68]. The Kinect for Windows SDK began by its first beta form that was discharged in July 2011. The beta was just a review form with an impermanent Application Programming Interface (API) and enabled clients to work with profundity and shading information and furthermore bolstered a progressed Skeletal Tracking which, in examination with an open– source SDKs, did not as of now require T–stance to instate skeleton following as is required in other Skeletal Tracking libraries.

Since the main beta Microsoft refreshed the SDK step by step up to rendition 1.7 and incorporated some of extra capacities. The principal significant refresh joined the 1.5

form that incorporated a Face Tracking library and Kinect Studio, a device for recording and replaying groupings caught by the sensor. The following form 1.6 broadened SDK by the likelihood of perusing an infrared picture caught by the IR camera lastly uncovered the API for perusing of accelerometer information. The as of now most recent Kinect for Windows SDK variant 1.7 was discharged in March 2013 and included propelled libraries, for example, Kinect Fusion, a library for 3D filtering and recreation, and a library for hand grasp location which has opened entryways for more common method for collaboration. The API of the Kinect for Windows SDK gives sensor's profundity, shading and skeleton information in a type of information streams. Every one of these streams can deliver genuine information outline by surveying or by utilizing an occasion that is raised each time another casing is accessible [69]. The accompanying points depict specific information streams and their choices.

3.2.4. Depth Stream

Information from the Kinect's depth camera are given by the depth stream. The depth information are spoken to as a casing made up of pixels that contain the separation in millimeters from the camera plane to the closest object as is outlined by the Figure 3.4.

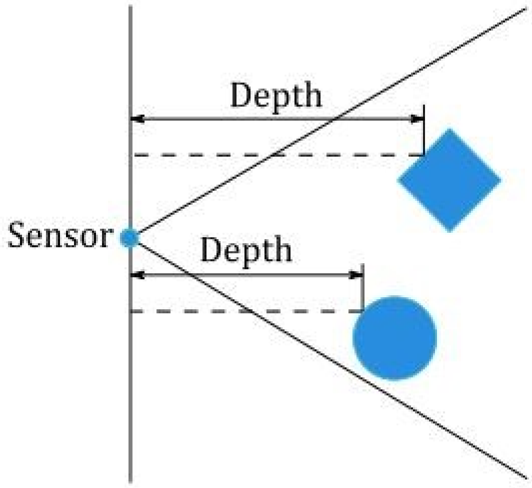

Figure 3.4 – An illustration of the depth stream values

The pixel blends the separation and player division information. The player division information stores data about a connection to the followed skeleton that empowers to connect the followed skeleton with the profundity data utilized for its following. The profundity information are spoken to as 16– piece unsigned whole number esteem where the initial 3 bits are saved for the player division information and the rest 13 bits for the separation. It implies that the maximal separation put away in the profundity information can be up to 8 meters. The Figure 3.5 represents the depth information portrayal. The profundity outline is accessible in various resolutions. The most extreme determination is 640 480 pixels and there are additionally accessible resolutions 320 240 and 80 60 pixels. Profundity outlines are caught in 30 outlines for every seconds for all resolutions. The profundity camera of the Kinect for Windows sensor can find in two territory modes, the default and the close mode. On the off chance that the range mode is set to default, esteem the sensor catches profundity esteems in run from 0.8 meter to 4.0 meters, generally when the range mode is set to close esteem the sensor catches profundity esteems in run from 0.4 meter to 3.0 meters. As indicated by the depiction of profundity space extend portrayed in [69] the maximal caught profundity esteem might be up to 8.0 meters in both range modes. In any case, nature of the profundity esteem surpassing a cutoff estimation of 4.0 meters in default mode and estimation of 3.0 meters in close mode might be corrupted with separate.

3.2.5. Color Stream

Color information accessible in various resolutions and organizations are given through the color stream. The color picture's organization decides if color information are encoded as RGB, YUV or Bayer.

The RGB design speaks to the color picture as 32– piece, straight X8R8G8B8– organized color bitmap. A color picture in RGB arrange is refreshed at up to 30 outlines for each seconds at 640 480 determination and at 12 outlines for every second in high– definition 1280 960 determination. [70]

The YUV design speaks to the color picture as 16– piece, gamma– adjusted direct UYVY– organized color bitmap, where the gamma revision in YUV space is proportional to standard RGB gamma in RGB space. As per the 16– piece pixel portrayal, the YUV design utilizes less memory to hold bitmap information and designates less cushion memory. The shading picture in YUV arrange is accessible just at the 640 480 determination and just at 15 fps. [70]

The Bayer organize incorporates more green pixels esteems than blue or red and that makes it nearer to the physiology of human eye [70]. The organization speaks to the color picture as 32– piece, direct X8R8G8B8– designed color bitmap in standard RGB color space. color picture in Bayer arrange is refreshed at 30 outlines for every seconds at 640x480 resolutions and at 12 outlines for each second in high– definition 1280x960 determination. [70]

Since the SDK adaptation 1.6, custom camera settings that permit streamlining the color camera for genuine natural conditions have been accessible. These settings can help in situations with low light or a splendidly lit scene and permit modifying tone, splendor or complexity keeping in mind the end goal to enhance visual clearness.

Furthermore, the color stream can be utilized as an Infrared stream by setting the color picture arrangement to the Infrared organization. It permits perusing the Kinect's IR camera's picture. The essential use for the IR stream is to enhance outer camera alignment utilizing a test design saw from both the RGB and IR camera to all the more precisely decide how to delineate starting with one camera then onto the next. Likewise, the IR information can be utilized for catching an IR picture in murkiness with a gave IR light source.

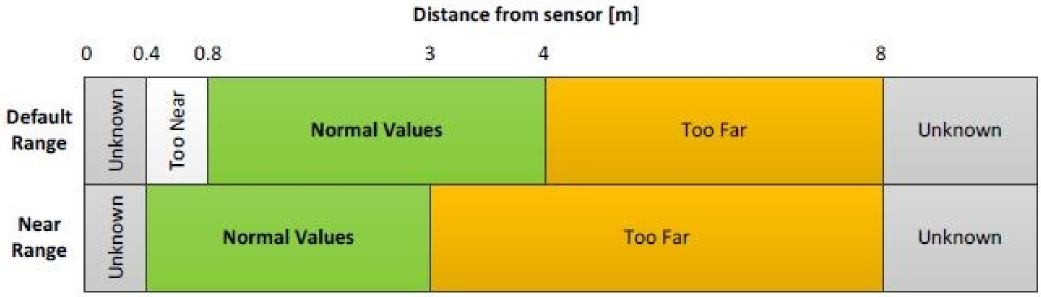

Figure 3.5 – Range of the depth stream values

3.2.6. Voice Stream

The Kinect sensor incorporates a four-component, microphone array. The microphone array catches sound information at a 24-bit determination, which permits precision over a wide powerful scope of voice information, from typical discourse at least three meters to a man shouting. The microphone array empowers a few client situations, for example, High-quality sound catch, Focus on sound originating from a specific heading with beamforming, Identification of the bearing of sound sources, improved discourse acknowledgment because of sound catch and beamforming and Raw voice information access.

3.3. Skeletal Tracking

The vital usefulness gave by the Kinect to Windows SDK is the Skeletal Tracking. The skeletal following permits the Kinect to perceive individuals and take after their activities. It can perceive up to six users in the field of perspective of the sensor, and of these, up to two users can be followed as the Figure 3.6 represents the skeleton comprised of 20 joints that speak to areas of the key parts of the user's body. The joints areas are really arranges with respect to the sensor and estimations of X, Y, Z organizes are in meters. The Figure 3.7 outlines the skeleton space.

The tracking algorithm is intended to perceive users confronting the sensor and in the standing or sitting stance. The tracking sideways stances is trying, as a component of the user isn't noticeable for the sensor. The users are perceived when they are before the sensor and their head and abdominal area is unmistakable for the sensor. No particular posture or alignment move should be made for a user to be followed. Figure 3.8 outlines skeleton examples of the basic stances is given by Kinect.

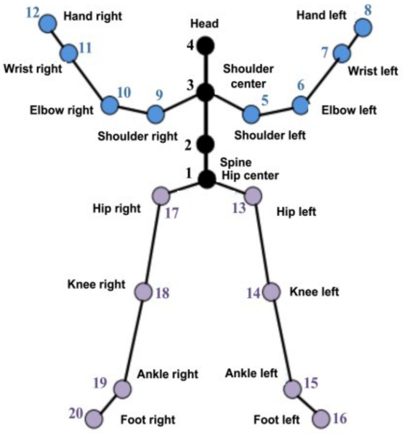

Figure 3.6 – Tracked skeleton joints

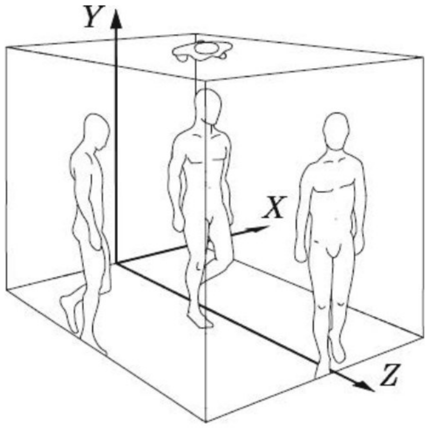

Figure 3.7 – An illustration of the skeleton space

The skeletal tracking can be utilized as a part of both range methods of the depth camera, see additionally 3.2.4 By utilizing the default range mode, users are followed out yonder in the vicinity of 0.8 and 4.0 meters away, however a reasonable range is between 1.2 to 3.5 meters because of a constrained field of view. If there should be an occurrence of close range mode, the user can be followed in the vicinity of 0.4 and 3.0 meters away, however it has a useful scope of 0.8 to 2.5 meters.

The tracking algorithm gives two methods of tracking. The default mode is intended for following every one of the twenty skeletal joints of the user in a standing stance. The situated mode is planned for following the user in a situated stance. The situated mode tracks just ten joints of abdominal area. Every one of these modes utilizes diverse pipeline for the following. The default mode identifies the user in light of the separation of the subject from the foundation.

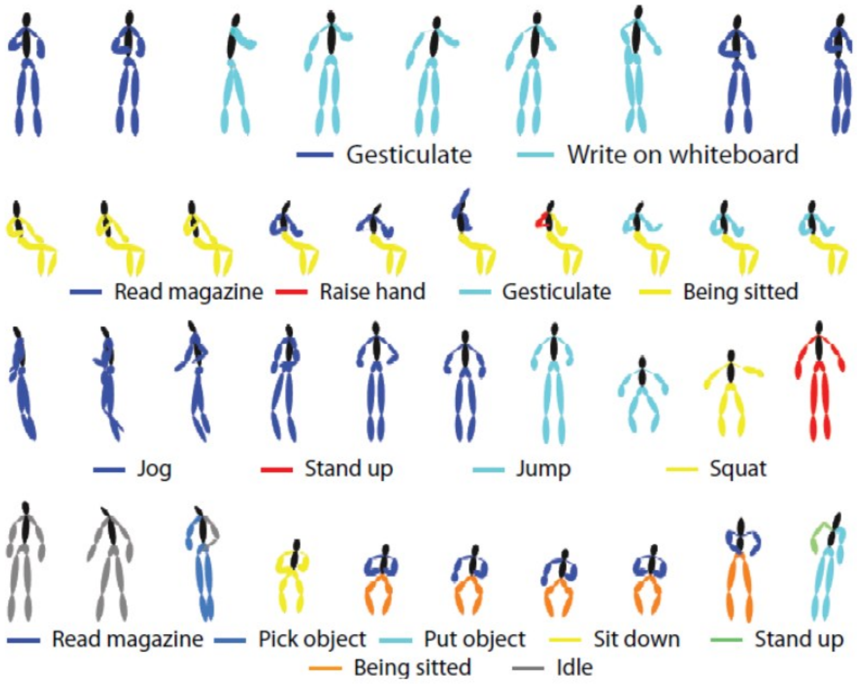

Figure 3.8- skeleton patterns of the Common postures

The situated mode utilizes development to recognize the user and recognize him or her from the foundation, for example, a couch or a seat. The situated mode utilizes a greater number of assets than the default mode and yields a lower throughput on a similar scene. In any case, the situated mode gives the most ideal approach to perceive a skeleton when the depth camera is in close range mode. By and by, just a single tracking mode can be utilized at once so it isn't conceivable to track one user in situated mode and the other one in default mode utilizing one sensor. The skeletal tracking joint data might be mutilated because of clamor and mistakes caused by physical constraints of the sensor. To limit jittering and settle the joint positions after some time, the skeletal tracking can be balanced crosswise over various casings by setting the Smoothing Parameters. To decrease the clamor of the yield, smoothing channel in view of the Holt Double Exponential Smoothing technique utilized for measurable investigation of monetary information. The channel furnishes smoothing with less inertness than other smoothing filter algorithms.

3.4. Graphical User Interface

Graphical User Interface (GUI) is the software dashboard of the overall system and it is depicted in Figure.3.9. Dashboard indicates the zone. parameters (temperature, Humidity, Light level and Output power), threshold reach alarms, current activated zone, controlling mode (voice mode and pattern mode), detected human pattern (sitting, standing, reading), human detected or not and selected angle of the detected patterns. Software has been coded by using C# language.

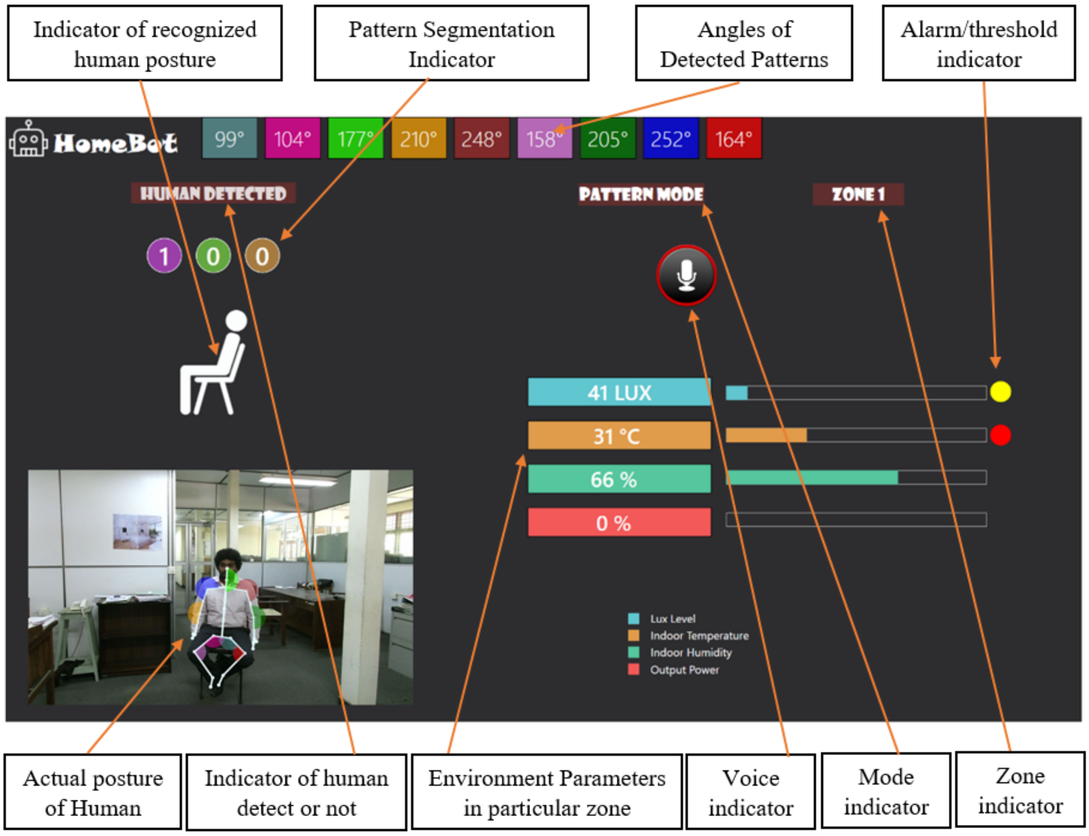

Figure 3.9 – Graphical user interface of the robot

3.5. Hardware platform

Robot Graphical User Interface (GUI) and sensor interface are integrated into the LattePanda board. This board is main platform of the robot. LattePanda is the advancement board that can run a full form of Windows 10. It is turbocharged with an Intel Quad Core processor and has great network, with three USB ports and coordinated Wi-Fi and Bluetooth 4.0. Inbuilt Wi-Fi module is used for developing wireless communication system of the robot. All software, including GUI and software algorithms is embedded in the LattePanda board. The voice response generation of the robot is developed by using an integrated audio circuitries in LattePanda board. LCD and LCD controller of the robot are driven by using inbuilt HDMI circuitries. Appearance of the of the LattePanda is illustrated in Figure 3.10.

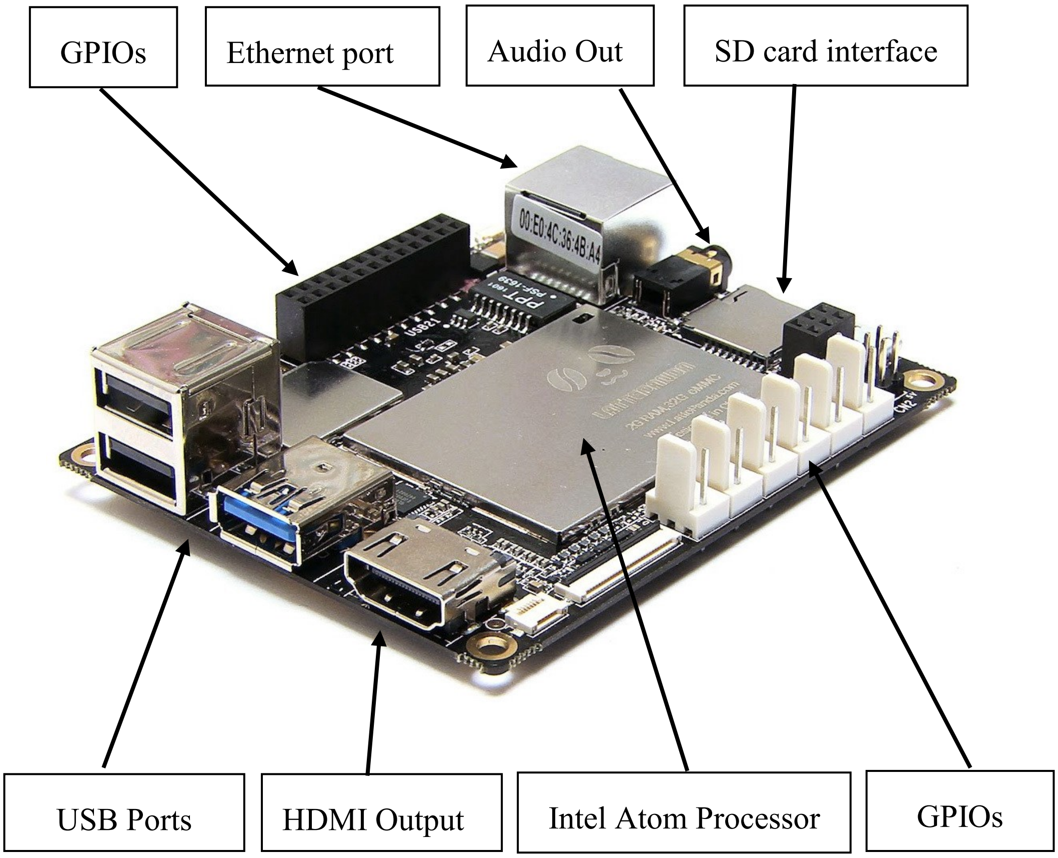

Figure 3.10 – LattePanda Single board computer.

NodeMCU is another development board, it’s used to develop smart object and provide wireless connectivity in between robot and smart object. The module consume low energy and it operating voltage is 3.3 v. NodeMCU is an open source Internet of Think (IoT) stage. It incorporates firmware, which keeps running on the

ESP8266 Wi-Fi System on Chip (SoC) from Espressif Systems, and equipment that depends on the ESP-12 module. This module can be coded by many languages such as Lua, C and C++. Arduino C/C++ compiler or standard Lua base compiler can be used to program final code into the device.

This module has been integrated general purpose input/output (GPIO), for controlling output device. Appearance of the NodeMCU is illustrated in Figure 3.11.

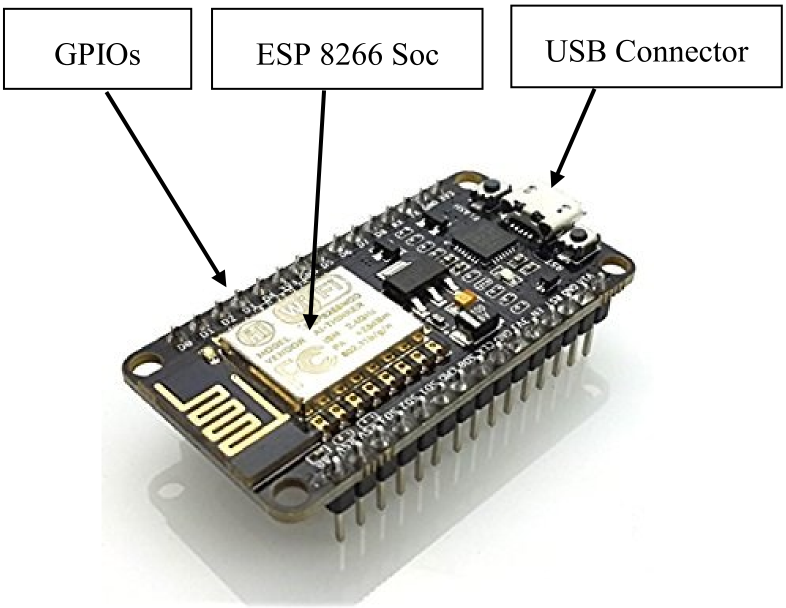

Figure 3.11 – NodeMCU module.

3.6. Behavioral Pattern Recognition System

Behavioral Pattern Recognition system (BPRS) was developed for identifying real time user behavior patterns in domestic environment. It can be used to recognize different human activities in daily routine of the humans. This system has incorporated with robot and implement as adaptive home environment. This system consists with behavior base comfort controlling, light controlling, providing alert and etc.

Kinect provided image streams are used for as raw data for this system. The Figure 3.12 illustrates the block diagram of the BPRS.

Depth Processing Unit

Depth processing unit is processed the actual depth coordinate of the human body using Kinect provided raw RGB and Depth streams.

Coordinate Segmentation Unit

Coordinate segmentation is a software algorithm, used for collecting depth coordinate in particular order.

Body Point Classifier

Body point classifier is a software algorithm, used for classified the human model using depth coordinates and store the coordinate in a database.

Human Finding Unit

Human finding unit match the actual data in to the classified coordinate model. The two unit is used for identifying and detecting whether the object is human or not.

Skeleton Constructor Unit

After detecting human object, the human skeleton is built by Skeleton Constructor using combination of each coordinates.

Joint Extraction Unit

Joint extraction unit is developed for sorting the required coordinates of each joints using constructed human skeleton.

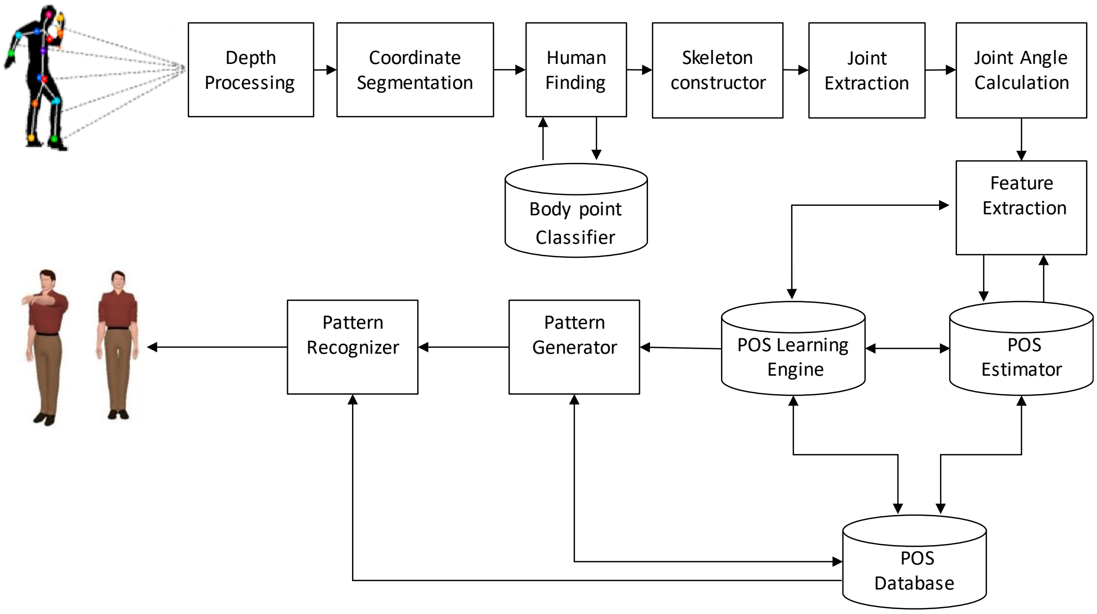

Figure 3.12 – Behavioral Pattern Recognition system

Joint Angle Calculation Unit

Joint angle calculation is a software algorithm, developed using vector coordinate system and dot product of two vectors to calculate joint angles in constructed human skeleton.

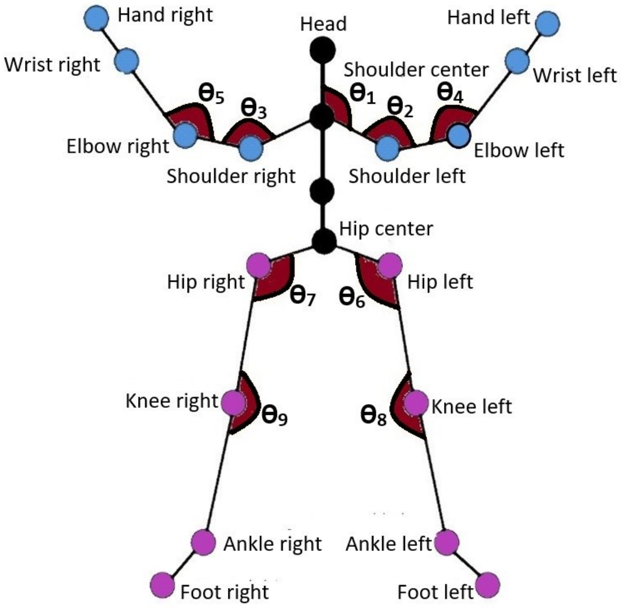

Figure 3.13 – Human skeleton with joint Angles.

Feature Extraction Unit

Feature Extraction model is used for extracting the features of the real time postures into the detected skeleton patterns. The nine major joint angles of human skeleton are used for developing of human posture identification system. Constructed human skeleton and major joint angels (ϴ1, ϴ2, ϴ3, ϴ4, ϴ5, ϴ6, ϴ7, ϴ8 and ϴ9) are depicted in Figure 3.13. Joint angles of each joints can be obtained by using vector dot product. The corresponding angle related two vectors could be obtained by using joint coordinate of each joints.

3.6.1. Vector Coordinate System

The calculating of joint angles are used skeleton coordinates and vector dot product. It is used for derive the general equation for finding any joint angle of the human skeleton. Figure 3.14 is illustrated the vector diagram of knee, hip and ankle joints related skeleton coordinates. It has used as an example and base approach of the developing joint angle calculation algorithm. The law of vectors can be used to obtain the coordinate of each joints related to the origin.

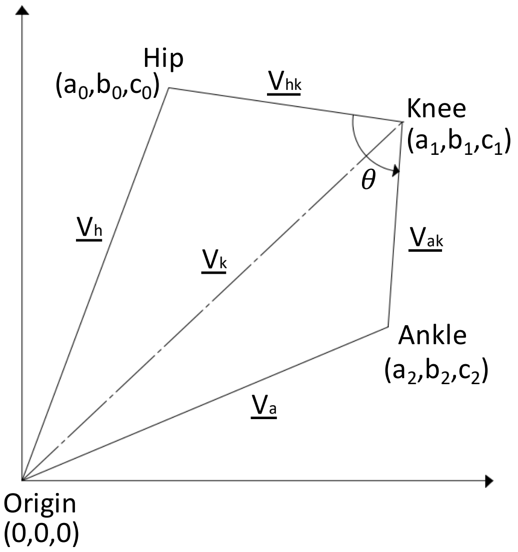

Figure 3.14 – vector diagram of knee, hip and ankle joint and coordinates of the human

Considering vectors of above diagram, the equation (1) and (2) represent the vectors of the particular joints and it can be use to obtain joint coordinates of each joints. Coefficient of each vector is represented as a_0, b_0, c_0, a_1, b_1, c_1, a_2, b_2 and c_2. Using dot product of two vectors, can be obtained the angle of each joints. The equation (3)

$$\overrightarrow{V_{hk}} = (a_0 - a_1)i + (b_0 - b_1)j + (c_0 - c_1)k \ - (1)$$
$$\overrightarrow{V_{ak}} = (a_2 - a_1)i + (b_2 - b_1)j + (c_2 - c_1)k \ - (2)$$

is represented vector dot product and equation (4) is represented with angle between two vectors. It is derived using base equation of vector dot product. Apply of the equation (1) and (2) into equation (4), it can be obtained cosine value of the angle between two vectors. Equation (5) is represented the cosine value of the angle. Finally take the Inverse cosine of the equation (6) and it can be used to derive the joint angle of the human skeleton. Unit vectors along with X, Y, Z directions are represented as i, j and k.

$$\overrightarrow{V_{hk}} . \overrightarrow{V_{ak}} = \left|\overrightarrow{V_{hk}}\right|\left|\overrightarrow{V_{ak}}\right| \cos\theta - (3)$$

$$\cos\theta = \frac{\overrightarrow{V_{hk}} \cdot \overrightarrow{V_{ak}}}{|\overrightarrow{V_{hk}}||\overrightarrow{V_{ak}}|} - (4)$$

Above equations are derived only for finding knee angle of the human skeleton. Considering these equations, they can be used to obtain general equation (7) for finding any angle of human skeleton. Ai and Bi represent as the vectors of the any angle between in human skeleton. Derived general equation can be used to calculate the values of the major nine angle of the human skeleton. After calculating values of each angles store in to the POS Database for developing of artificial neural network model.

$$cos\theta = \frac{\begin{pmatrix} a_0 - a_1 \\ b_0 - b_1 \\ c_0 - c_1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix} \cdot \begin{pmatrix} a_2 - a_1 \\ b_2 - b_1 \\ c_2 - c_1 \end{pmatrix} \begin{pmatrix} i \\ j \\ k \end{pmatrix}}{\sqrt{(a_0 - a_1)^2 (b_0 - b_1)^2 (c_0 - c_1)^2} \sqrt{(a_2 - a_1)^2 (b_2 - b_1)^2 (c_2 - c_1)^2}} - (5)$$

$$\theta_{ha} = \cos^{-1} \left(\frac{[(a_0 - a_1)(a_2 - a_1)] + [(b_0 - b_1)(b_2 - b_1)] + [(c_0 - c_1)(c_2 - c_1)]}{\sqrt{(a_0 - a_1)^2 (b_0 - b_1)^2 (c_0 - c_1)^2} \sqrt{(a_2 - a_1)^2 (b_2 - b_1)^2 (c_2 - c_1)^2}} \right) - (6)$$

$$\theta_{ij} = \cos^{-1} \sum \frac{A_i \cdot B_i}{|A_i||B_i|} - (7)$$

3.6.2. Development of Neural Network

A feed forward neural network is a kind of a system, which has a progression of layers and a weighted aggregate of info streams forward way as it were. Every one of the layers has association just from its quickly past layer. The layers between the input and output layers are called hidden layers. In spite of the fact that there can be numerous hidden layers, one concealed layer with a few neurons fits any input output mapping. On the off chance that palatable outcomes are not acquired, the quantity of neurons in the shrouded layer is fluctuated.

Keeping in mind the end goal to acquire a mapping between two spaces (input-output) an irregular arrangement of weights must be adjusted with the goal that mistake between desired and produced output is limited. Back-propagation algorithm is one such model where weight adaption continues from the last to first layer. There are a few weight adjustment strategies for actualizing Back-propagation algorithm. Out of them, Levenberg-Marquardt optimization has been utilized as it outperforms inclination plummet pursuit and conjugate angle (or quadratic guess) strategies for medium measured issues.

A feedforward neural network shown in Figure 3.15 has been designed for the user behavior identification. Those nine joint angles are fed as the input of the neural network. The nine joint angles are considered by the neural network to detect the behavior of the user.

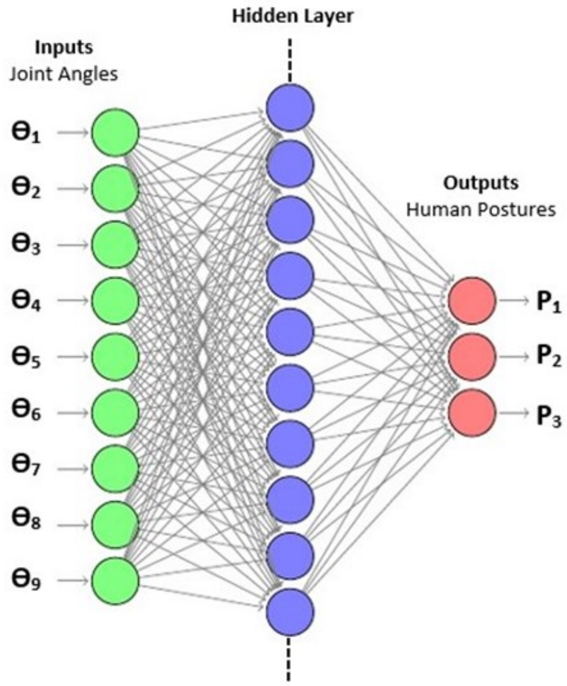

Figure 3.15– Neural network model of the system

This system is planned one hidden layer having 100 neurons for obtaining best performance of neural network. The inputs are taken by the 9 neurons in the input layer. The output layer consists with 3 neurons. The output of the neural network is the classified user behavior.

The neural network has been trained using a data set of 5000 samples for a single posture. Tanning, testing and validation of neural network is indicated in Figure 3.16. According to the learning curves, neural network is very accurate in selected dataset.

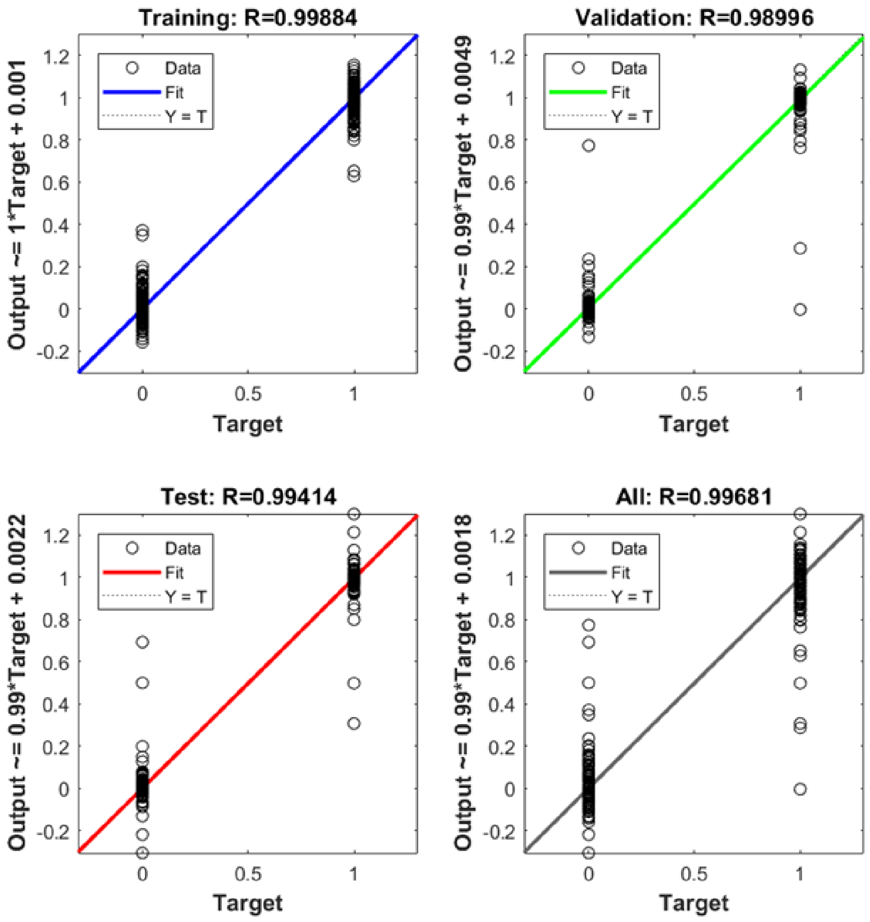

Figure 3.16 – Testing, tanning and validation of neural network

The neural network performance and error histogram are illustrated in Figure 3.17 and Figure 3.18

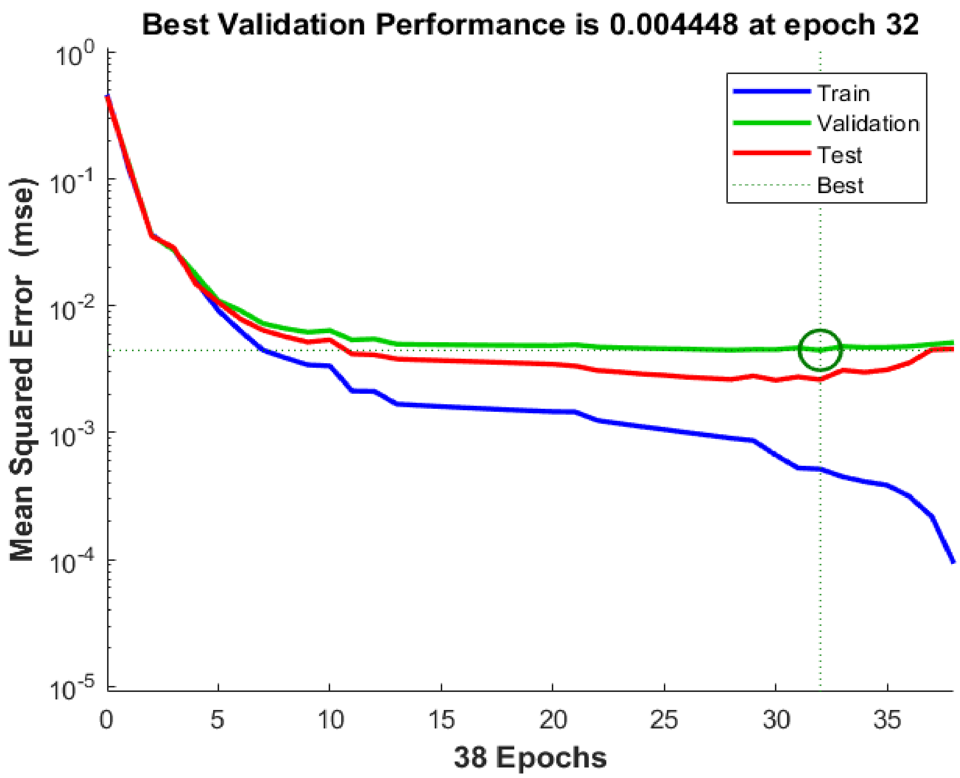

Figure 3.17 – Performance of neural network

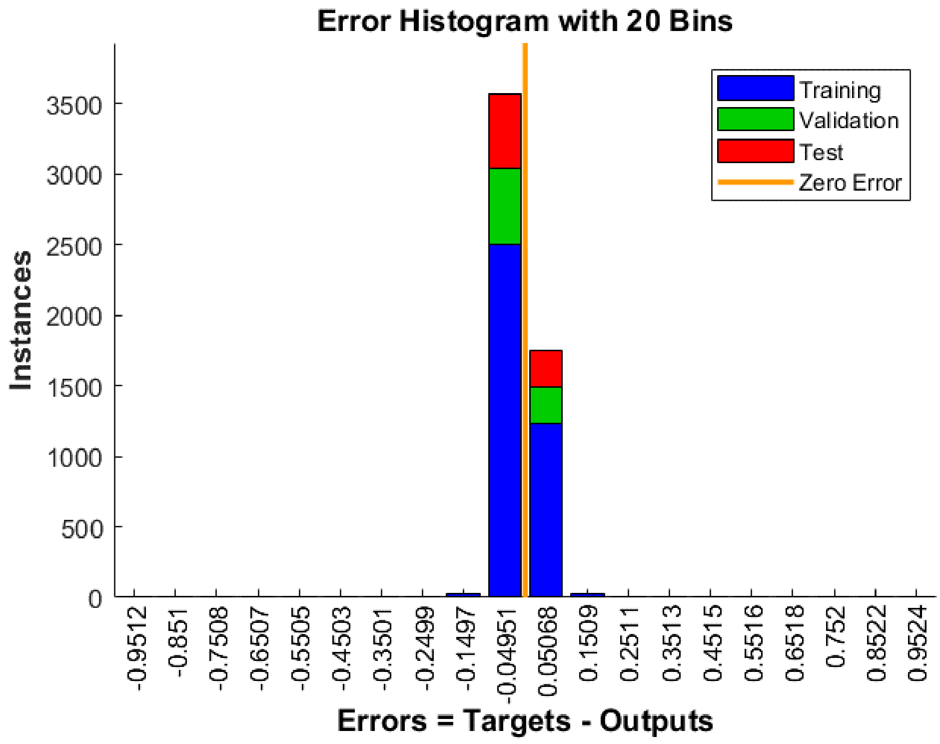

Figure 3.18 – Error Histogram of the Neural Network

POS Learning Engine and POS Estimator

POS Learning Engine and POS Estimator are brain of the behavior pattern recognize system. POS learning engine is consisted of a software algorithm for training neural network. POS estimator is consisted software algorithm for deciding final output. POS Learning Engine and POS Estimator can be used to handle the learning accuracy and estimate the closeness of real time posture, collaborated with artificial neural network.

POS Database

POS Database is used to store the training data values of the neural network for detecting human postures.

Pattern Generator

The estimated final patterns of the humans are generated using Pattern Generator. Pattern Generator obtains the final output of the behavior pattern recognition system.

Pattern Recognizer

Pattern recognizer algorithm is used to recognize the final classified output of the detected human postures.

3.7. Vocal Interaction System

Vocal interaction system is the other important part of the robot. It consists with two major units such as voice input system and voice response system. Voice input system is used for provide the voice command in to the robot by user. Voice response system is used for generating voice output for the user by robot. Overall functionality of the vocal interaction system is depicted in Figure.3.19 in a modular format.

Kinect voice stream is used as an input of the voice input system. Inbuilt microphone array of the Kinect is generated the voice stream and noise cancelation unit is reduced the background noise of the input voice signal. The microphone array enables High- quality audio capture. It is focused on audio coming from a particular direction with beamforming and identified the location of the audio sources.

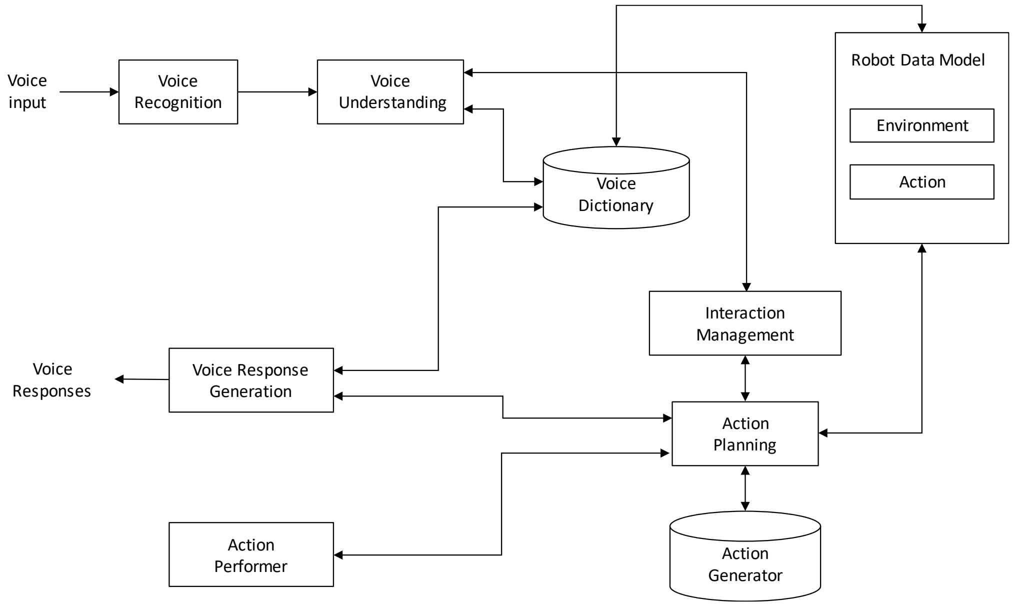

Figure 3.19 – Vocal Interaction System

Voice Recognition Unit

Voice recognition system take the voice stream and convert into the system related text strings.

Voice Dictionary

Voice Dictionary has stored predefined command strings used by the robot. User can be stored any text string into voice dictionary in programming mode.

Voice Understanding Unit

Voice Understanding unit is used for validating of stored predefined command string in the Voice Dictionary. It provides real time text strings of the voice recognition system. Microsoft speech recognizer engine has been used to implement the voice recognition functionality of the robot.

Robot Data Model

Robot Data Model is temporary data storing system of the robot. It contains two data units such as environment data unit and action data unit. Environment data unit contains real-time environment data like temperature, humidity, light level etc. Those data are updated by inbuilt sensors of the robot. Action data unit stores the temporary generated action data provided by robot.

Interaction Management Module (IMM)

The Interaction Management Module (IMM) manages voice interactions between the user and the robot.

Action Generator

Action Generator obtains control command as the text strings in-between robot and smart object.

Action Planning Unit

Action Planning unit schedules the command string sequence and voice string sequence in particular order. Other one is that, it is used to update the text string in suitable parameters associate with action data unit.

Voice Response Generation Unit

Voice Response Generation section is a text to speech converter implemented using Microsoft Speech API (Application Program Interface). It is used to convert voice string into the analog audio output associated with audio hardware module. Voice input is used in this system to identify a user instruction that are compatible to the operation domain of the robot companion such as controlling of an appliance. Voice responses of the robot are used to provide the alerts and status of the environment and zones. A simple questionnaire such as asking name, time, date, temperature, humidity and zone condition can also answered by the robot with the help of both voice input and voice response systems. Action Performer is prepared control string in to the data object for transmission in between robot and smart object via wireless sensor network.

3.8. Wireless Sensor network

Wireless Sensor Network (WSN) is used for establish the communication link in-between robot and smart objects. Wi-Fi router has enabled the communication bridge facility in-between robot and smart object. Star topology has used to implement wireless network. Two hardware module has been used to enable Wi-Fi feature of the system, such as Wi-Fi hardware module of the robot and embedded Wi-Fi hardware module of the smart object. Smart devices embedded communication system has been developed by using ESP8266 chip with inbuilt Wi-Fi stake. The communication system of the robot has been developed by using a single board computer inbuilt Wi-Fi module. TCP/IP (Transmission Control Protocol/Internet Protocol) protocol has been used to implement communication tunnel in-between robot and smart object.

Communicating system of the robot is illustrated at Figure.3.20 and communication system of smart object is illustrated at Figure.3.21.

Action Performer

Considering robot side, Action Performer prepares the data object for communication.

Action Validator

Action Validator checks the validity of the transmission data. If data is not validated, it rejects all data and refresh communication tunnel.

Network Data Memory unit

The IP (Internet Protocol) address, MAC (Media Access Control) address and network related temporary data are being stored by using Network Data Memory unit.

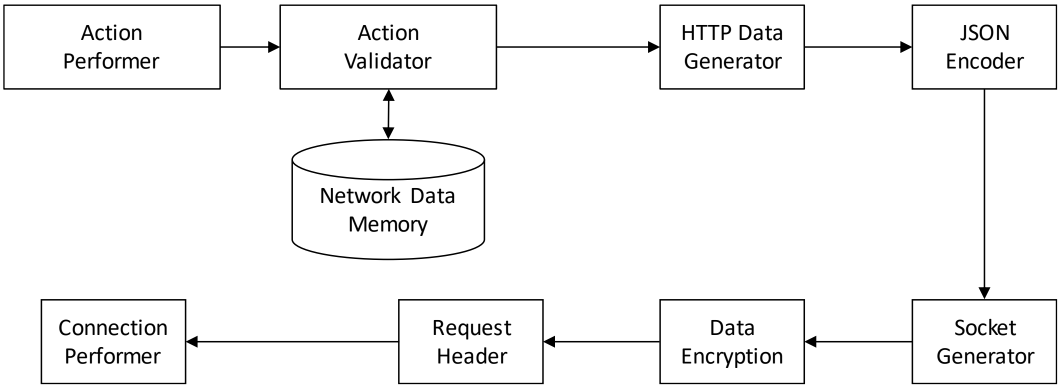

Figure 3.20 – communication System of the robot

HTTP Data Generator

After checking validity of the data, HTTP (Hypertext Transfer Protocol) Data Generator fetches the usable data in to the HTTP header.

JSON Encoder

JSON (Java Script Object Notation) encoder converts raw data in to the JSON format. JSON is used for this system; it required low data bandwidth during transmuting and reserving periods.

Socket Generator

Socket Generator assigns the socket to the TCP/IP transmission protocol. Socket is used to assign the separate medium in Ethernet communication for high speed data transmission.

Data Encryption Algorithm

Data Encryption Algorithm is used for data encryption and translates data into another form, or code, so that only people with access to a secret key (formally called a decryption key) or password can read it.

Request Header

Request header is a software algorithm used to finalize the data packet for sending over the network.

Connection Performer

Establish the connection of the robot and smarts object is handled by Connection Performer.

Responds Header

Considering the communication system of the smart object, the incoming data of the robot are handled by using Responds Header for controlling home appliances.

Socket Serializer

Socket Serializer is used to validates socket of the HTTP header and it is used to serialize object, buffer, string, number and boolean with the minimum of transformations in order to improve performance.

HTTP Serializer

The unwanted part (IP address, MAC address) of the incoming data object is removed by HTTP Serializer.

Data Decryption Algorithm

Decryption is the process of taking encoded or encrypted text or other data and converting it back into text that smart object can read and understand.

JSON Serializer.

JSON object of the input data stream is converted into the raw data by using JSON serializer.

Network Data Memory

The IP address, MAC address and network related temporary data of the smart object are being stored by using network data memory unit.

Action validator

Action validator is checked the valid action included into the incoming data.

Action performer

Action performer performs final action collaborated with hardware unit.

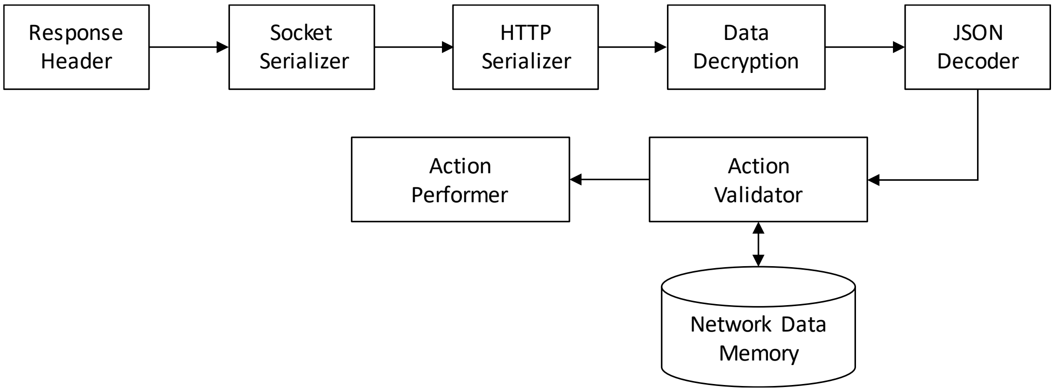

Figure 3.21 – communication System of the smart object

4 INTELLIGENT CONTROL AND RESULTS OF THE ROBOT

This chapter discusses the intelligent control of the robot and output results of the overall system. Hence prove that the parameters found from the above chapters (Chapter 2 and 3) and the results of the developed system. Design includes the design of behavior pattern recognition system, vocal interaction system, design of structure of the robot etc. And also includes the final output of the system once the robot is assembled. Finally, overall module test and validate in domestic environment.

4.1. Experimental Prototype

To validate the use of the robot configuration approach within the context of the smart home environment, a testbed was constructed to replicate the setup of a real house and a set of experiments were conducted to assess the Robot's ability to automatically configure logical connections among services and devices within the smart home environment. The Robot is expected to enable any service applications, once delivered, to utilize the available and connected devices, taking into account the smart home settings, to meet the user's requirements. The testbed floor layout is shown in Figure 4.1 as the representative environment for the delivery of various services (e.g. Environment Alert, Lighting control and energy management) to the smart home.

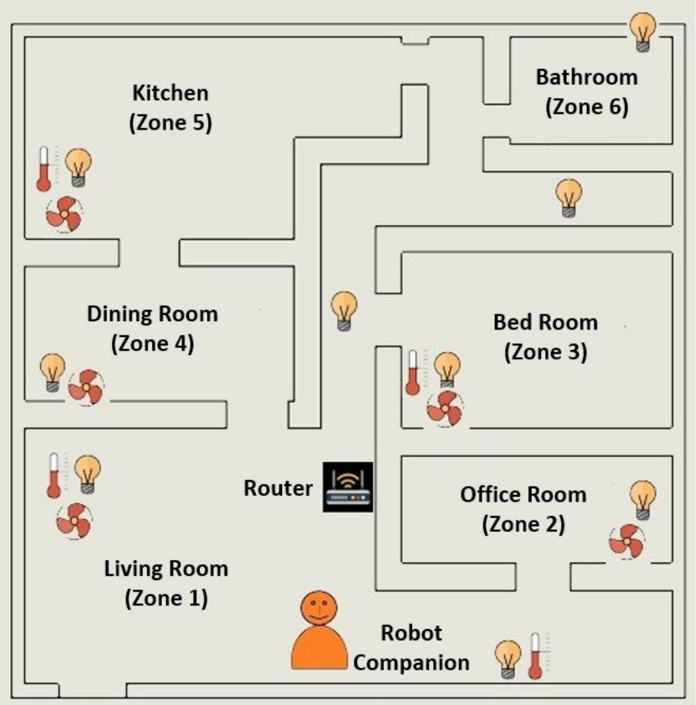

Figure 4.1 – Home with smart object placement

As Figure 4.1 shows, the testbed contains all representative functional areas for viability assessment. It is desirable because the characteristics of different functional areas or rooms within the smart home environment are likely to be closely linked to the available devices and services. The floor layout of the testbed consists of several indoor and outdoor functional areas. A functional area is classified as indoor if it is assumed that the room has a roof.

Based on their physical locations, the indoor functional areas are grouped into the ground floor and first floor, by convention and according to the requirements of potential services. Each room is connected to adjacent rooms via entrances. Similarly, the outdoor area is linked to the indoor area via entrance(s).

The testbed is divided into separate zone and provide the ID for each zone. All smart node communicates using the robot via Wi-Fi communication. User should introduce the zone ID to robot for controlling the appliances that particular zone. Every zone equipped with multiple devices. In particular, a Humidity sensor, a temperature sensor, and a light level sensor are provided real time environment condition into the robot. A AC/Fan/Light controller can be used to control the home appliances such as Light, Fan, AC and etc. according to the robot commands. Symbols of the Figure 4.1 is indicated smart object placement at home.

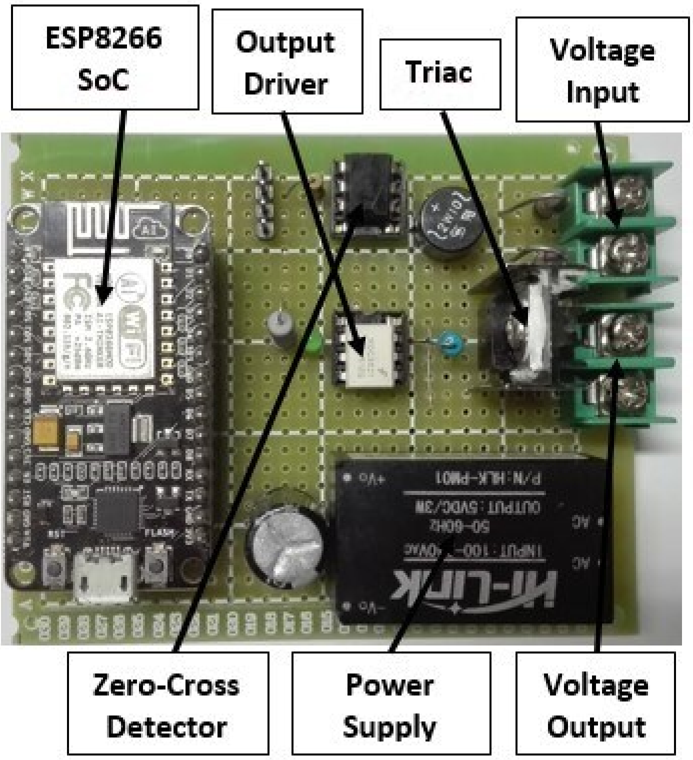

Figure 4.2 – Inside the smart object

Inside of the smart object (wireless control module) is depicted in Figure 4.2. It can handle 230v AC with 15A of average current. Inbuilt PWM power control method is used to control the output power of the connected appliances and ON/OFF control method is used to switch the appliances. Appearance of the developed robot companion is shown in Figure 4.3.

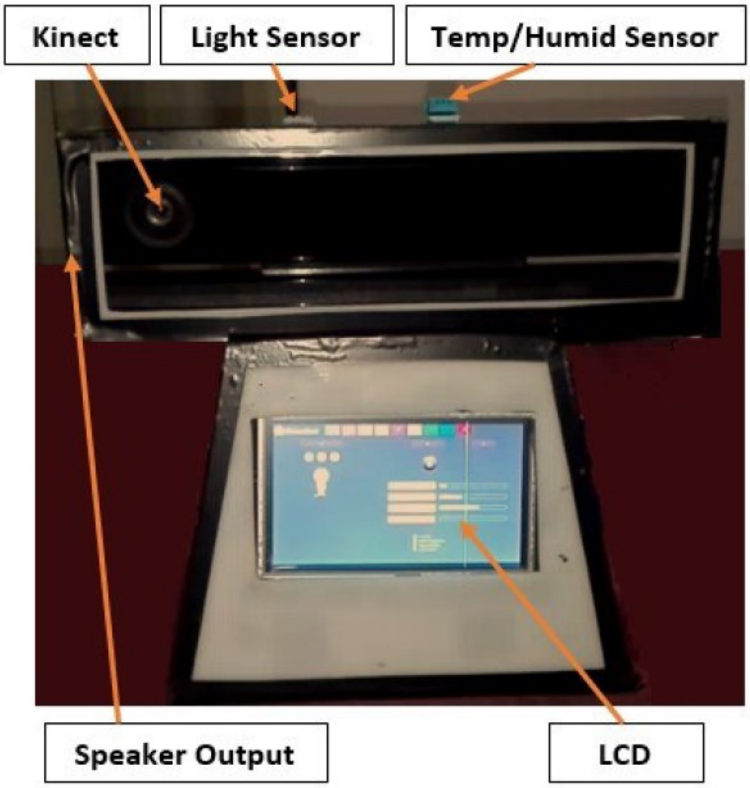

Figure 4.3 – Robot Companion with inbuilt module.

Automating energy conservation (light control, Fan control and AC control) activities is considered an effective way to reduce energy consumption in the home environment and thus lower energy costs, without affecting the occupants' comfort. In the smart home environments, energy conservation strategies are devised to mimic the occupants' conservation activities to address wastage. Inhabitant activity base demand control approach becomes necessary solution to address that issue.

4.2. Behavior Pattern Recognition System

Experiments have been conducted to verify the behavior pattern recognition system. For this, several people were asked to be engaged in different activities and the responses of the system during the experiment have been recorded for the analysis. The detected patterns of the developed system has been discussed following subtopics. These patterns of the domestic environment have been recognized by using of robot companion incorporated behavior pattern recognition system.

4.2.1. Case I: Results of the Sitting posture

Following results are presented the sitting posture of the humans in five various sitting patterns. Figure 4.4(a) to 4.4(e) are illustrated the detected patterns by the robot.

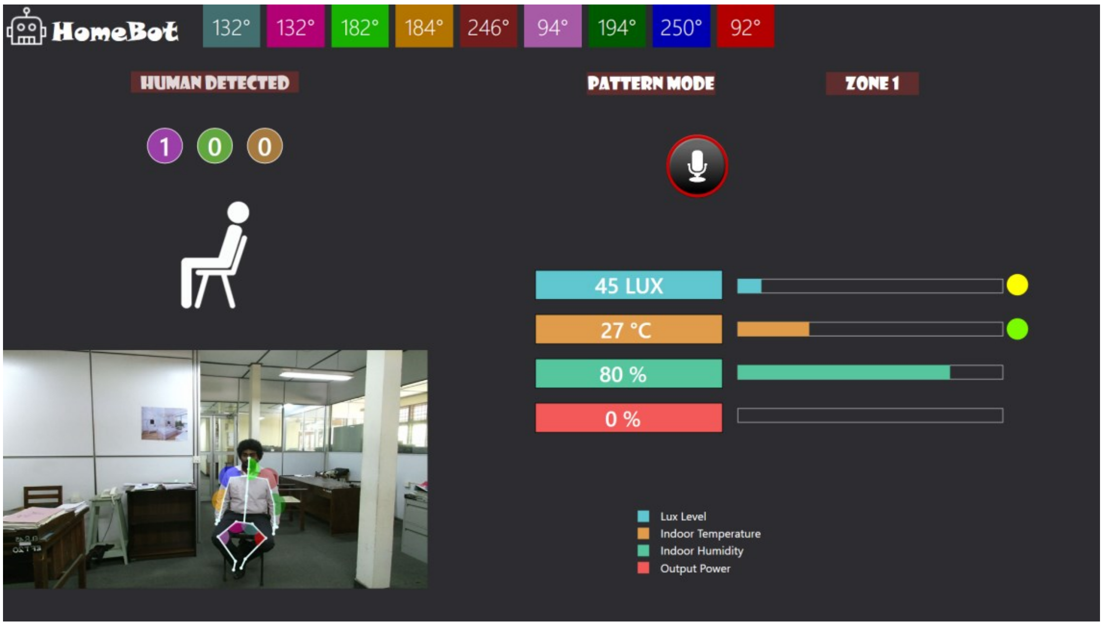

Figure 4.4(a) – Result obtained for "Sitting" of the actual pattern of the human vs detected pattern (Sitting Pattern 1).

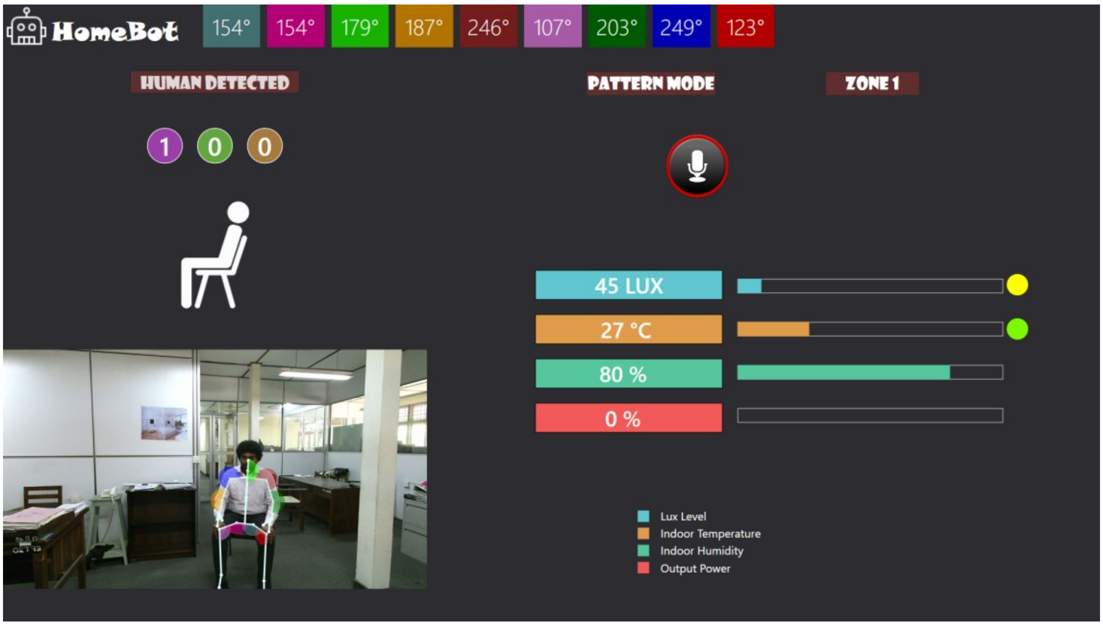

Figure 4.4(b) – Result obtained for "Sitting" of the actual pattern of the human vs detected pattern (Sitting Pattern 2).

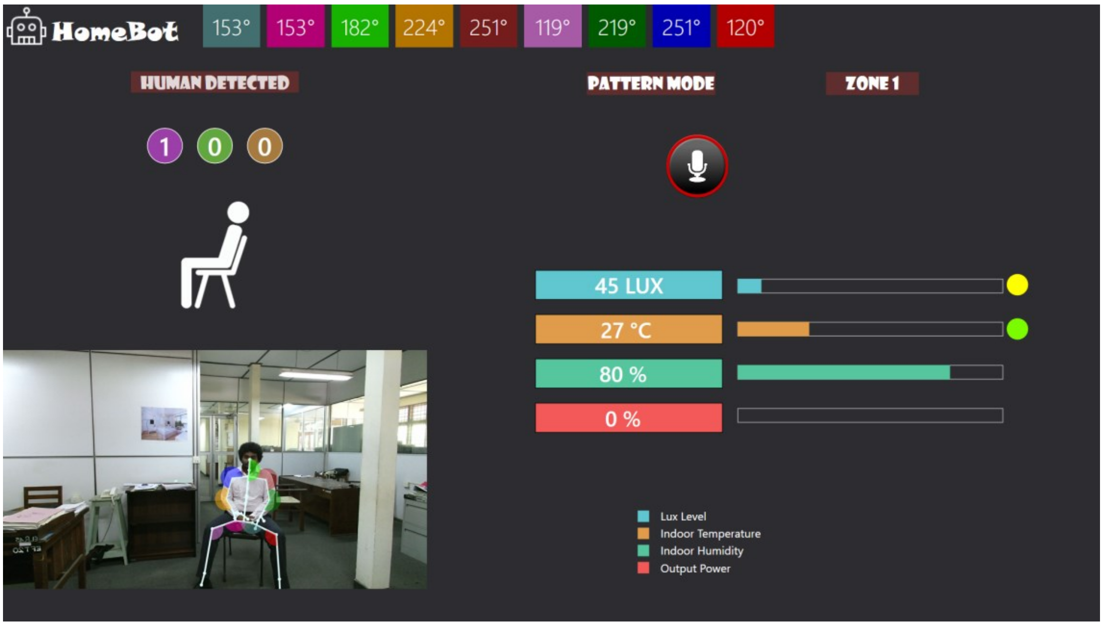

Figure 4.4(c) – Result obtained for "Sitting" of the actual pattern of the human vs detected pattern (Sitting Pattern 3).

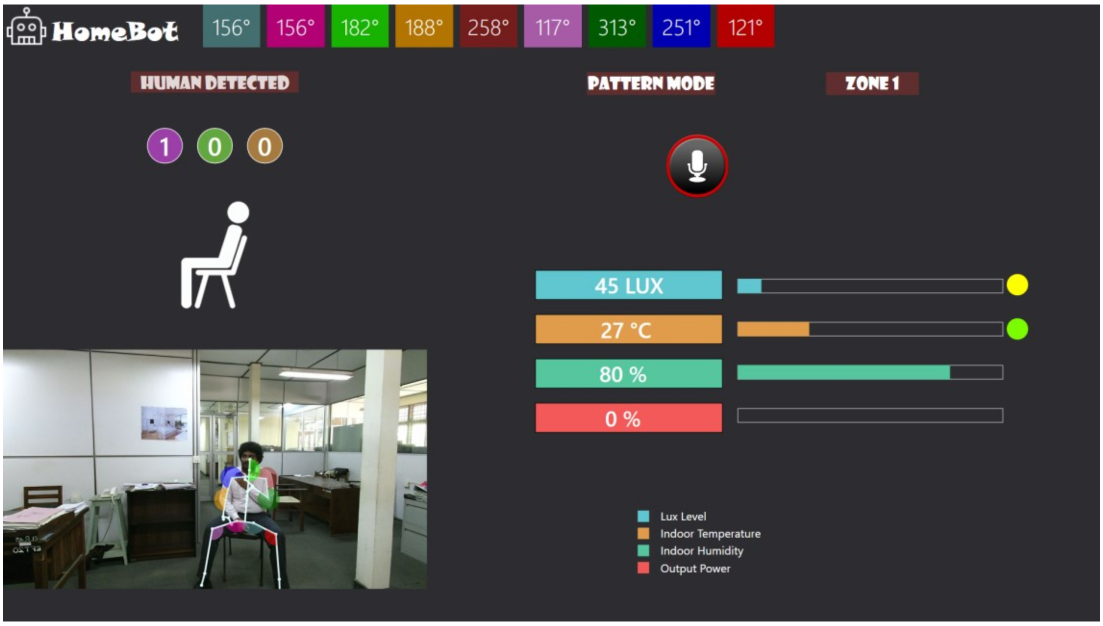

Figure 4.4(d) – Result obtained for "Sitting" of the actual pattern of the human vs detected pattern (Sitting Pattern 4).

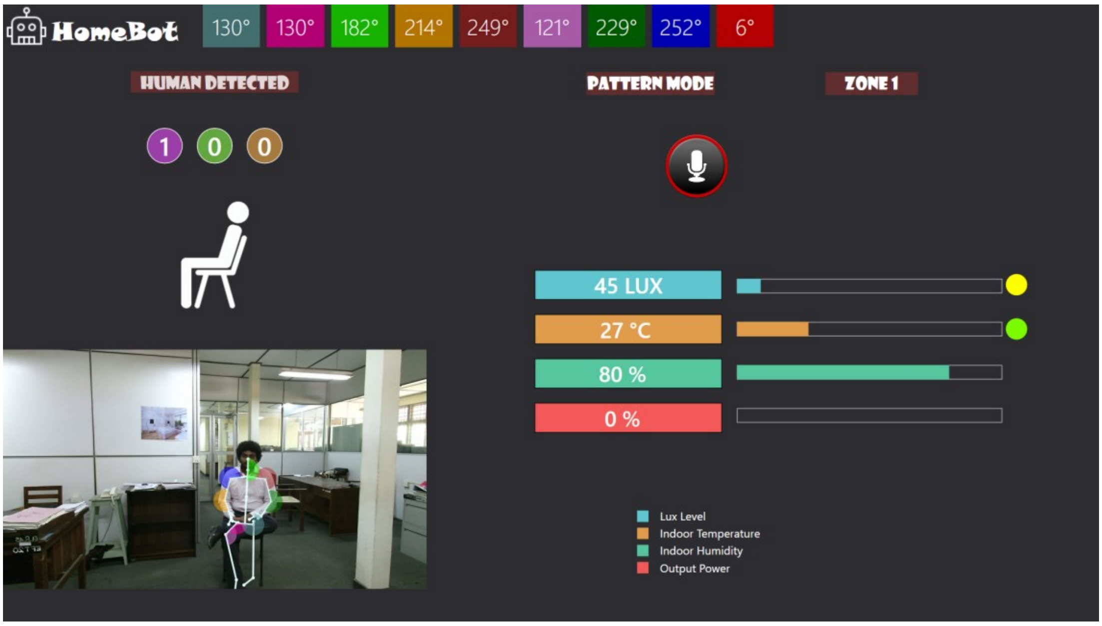

Figure 4.4(e) – Result obtained for "Sitting" of the actual pattern of the human vs detected pattern (Sitting Pattern 5).

4.2.2. Case II: Results of the Sitting with Reading posture

Following results are presented the sitting with reading posture of the humans in five various sitting patterns. Figure 4.5(a) to 4.5(e) are illustrated the detected patterns by the robot.

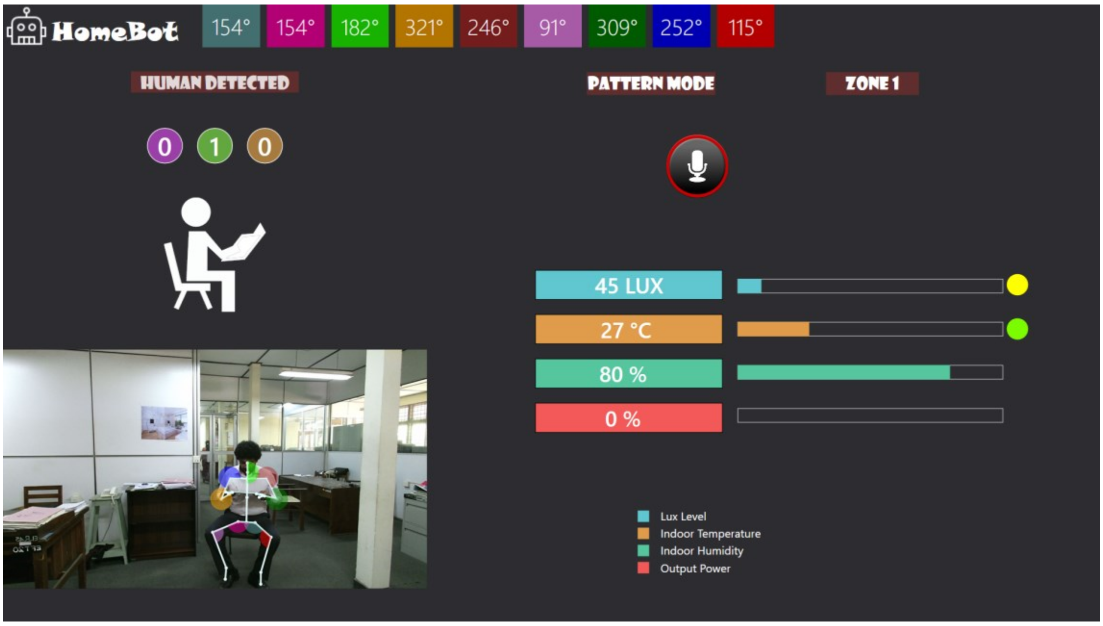

Figure 4.5(a) – Result obtained for "Sitting with Reading" of the actual pattern of the human vs detected pattern (Sitting with Reading Pattern 1).

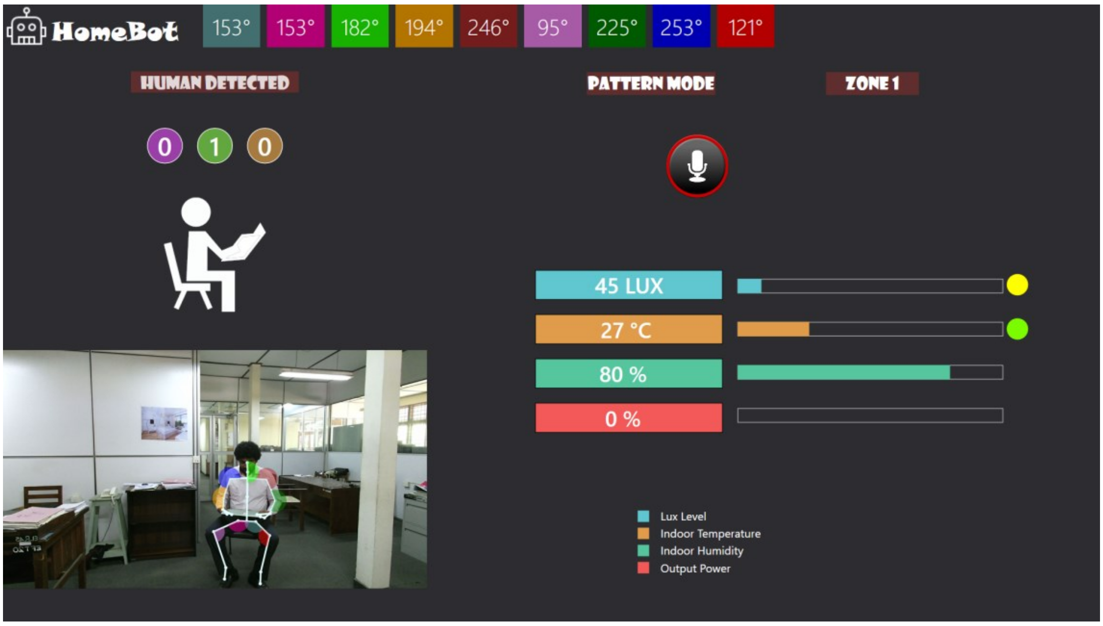

Figure 4.5(b) – Result obtained for "Sitting with Reading" of the actual pattern of the human vs detected pattern (Sitting with Reading Pattern 2).

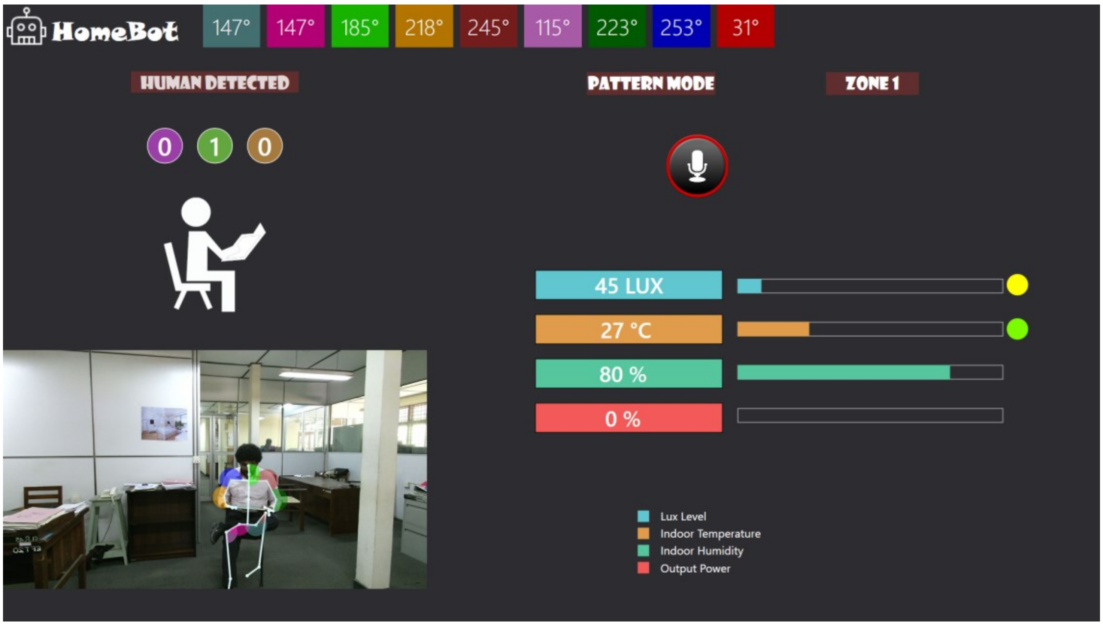

Figure 4.5(c) – Result obtained for "Sitting with Reading" of the actual pattern of the human vs detected pattern (Sitting with Reading Pattern 3).

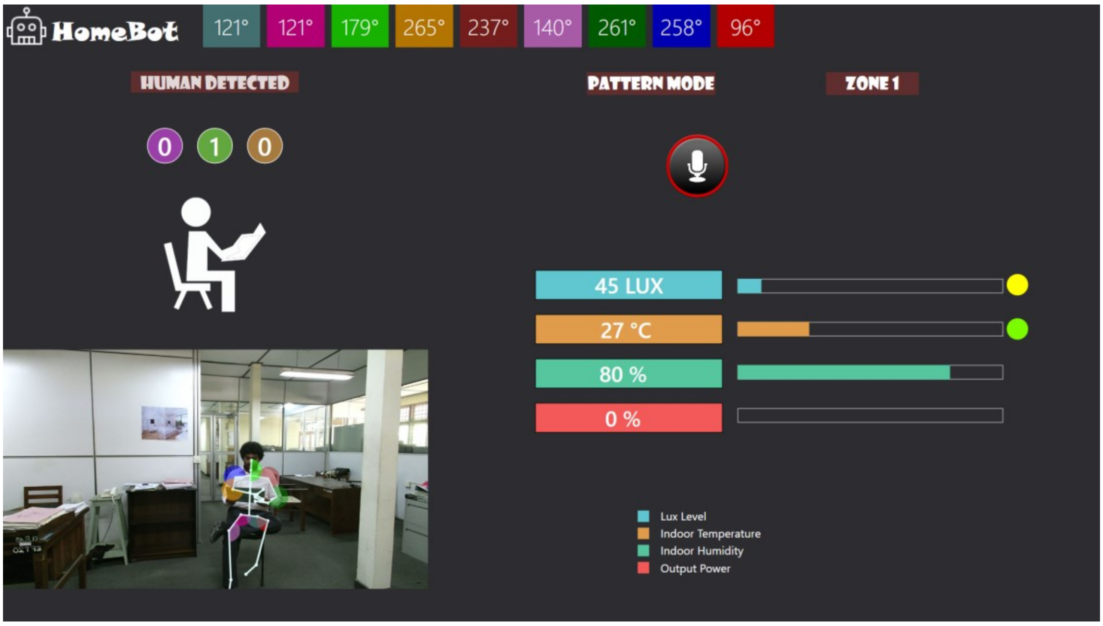

Figure 4.5(d) – Result obtained for "Sitting with Reading" of the actual pattern of the human vs detected pattern (Sitting with Reading Pattern 4).

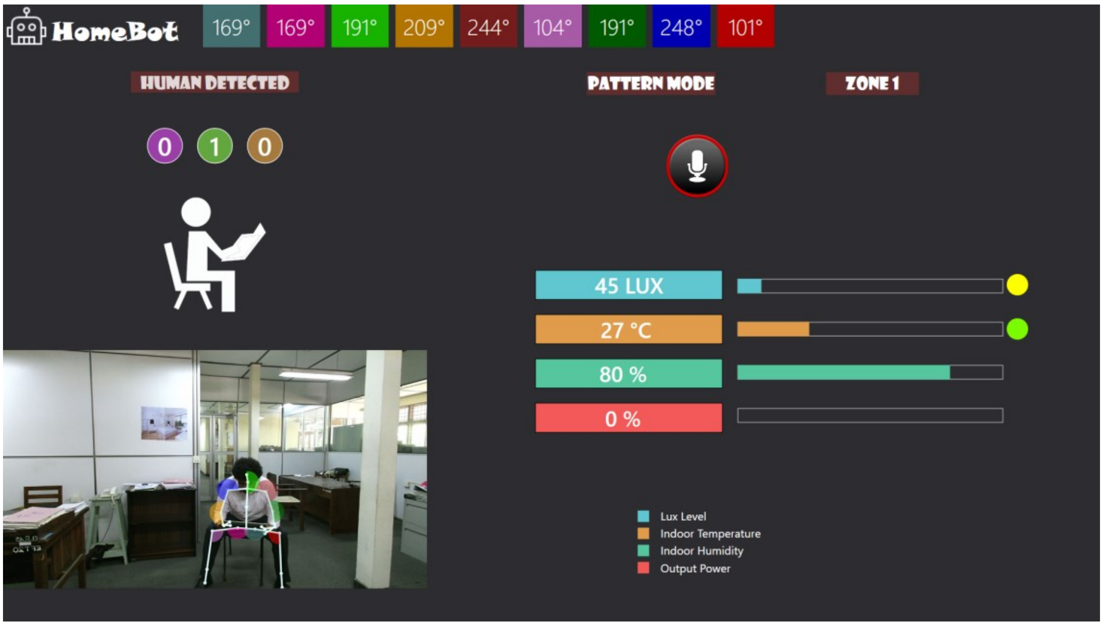

Figure 4.5(e) – Result obtained for "Sitting with Reading" of the actual pattern of the human vs detected pattern (Sitting with Reading Pattern 5).

4.2.3. Case III: Results of the Standing posture

Following results are presented the standing posture of the humans in five various sitting patterns. Figure 4.6(a) to 4.6(e) are illustrated the detected patterns by the robot.

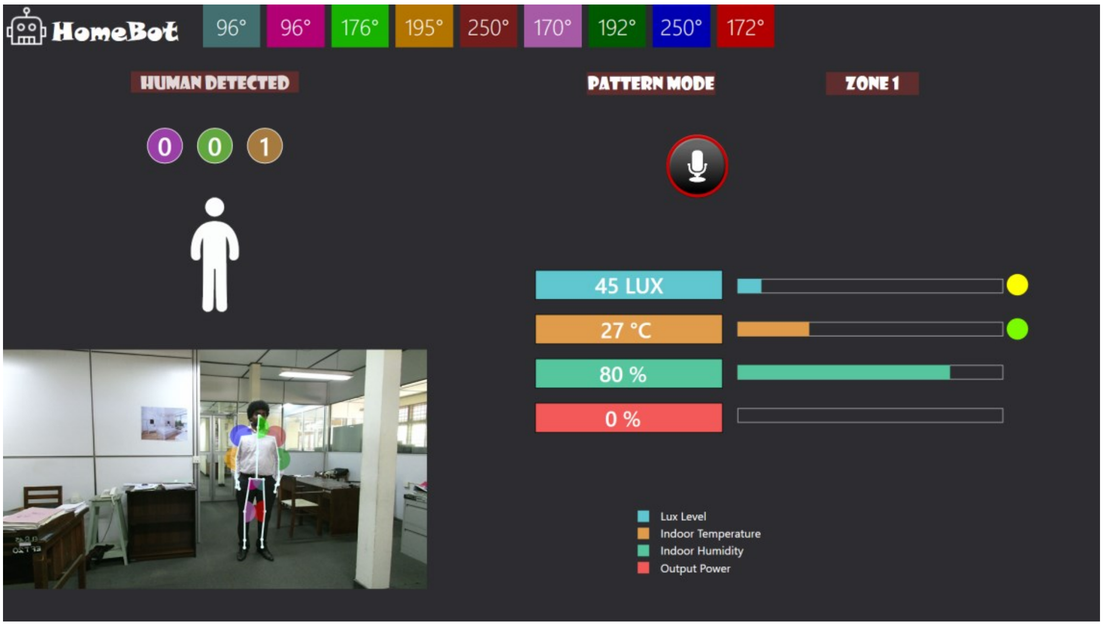

Figure 4.6(a) – Result obtained for "Standing" of the actual pattern of the human vs detected pattern (Standing Pattern 1).

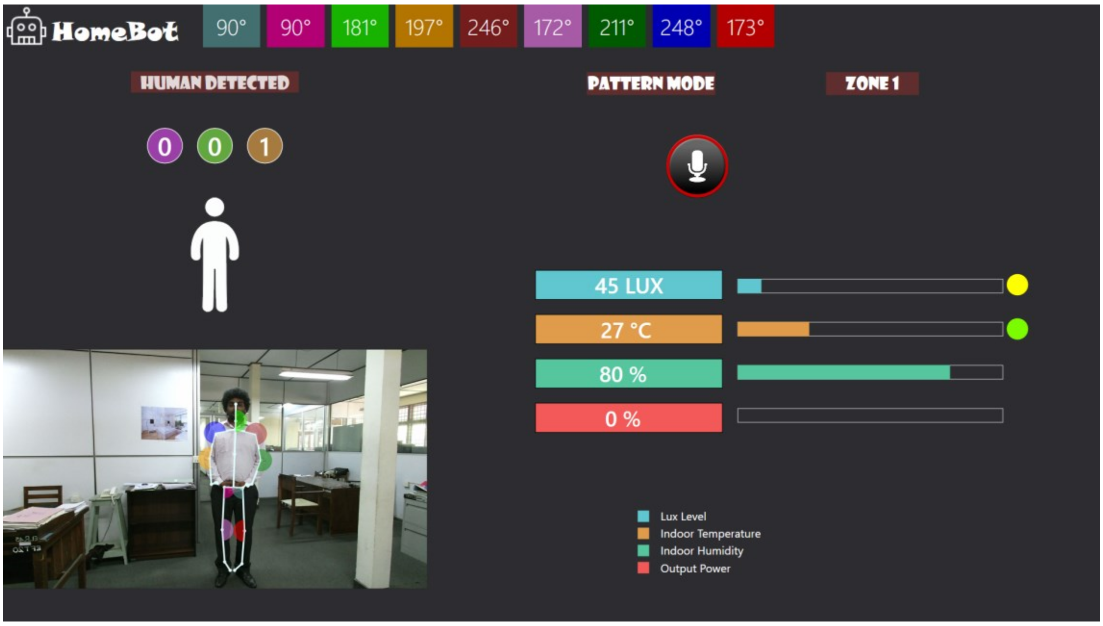

Figure 4.6(b) – Result obtained for "Standing" of the actual pattern of the human vs detected pattern (Standing Pattern 2).

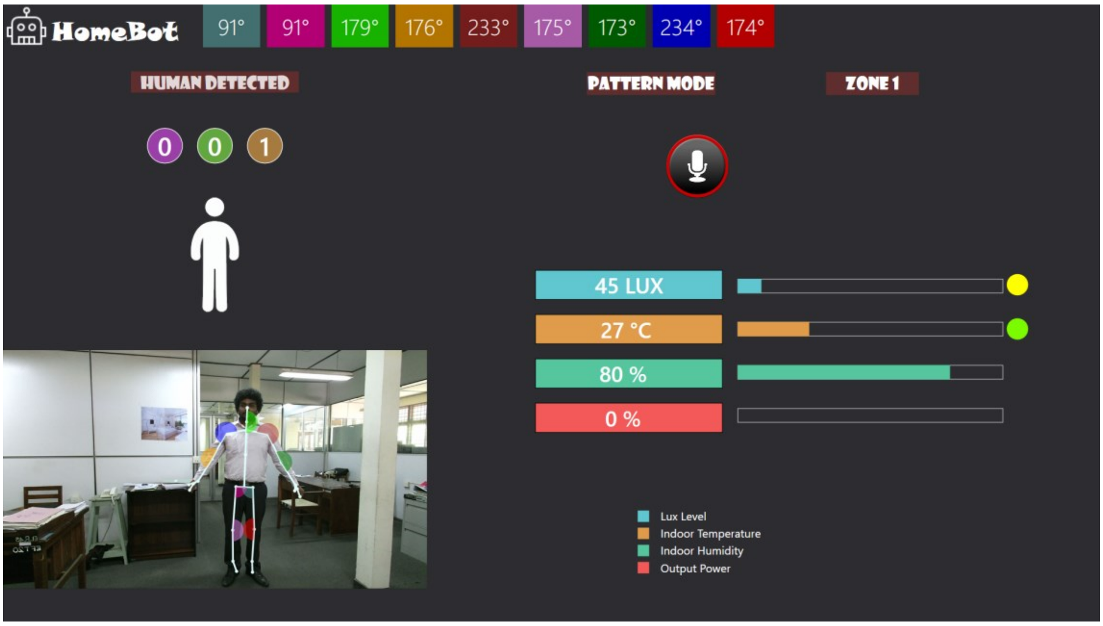

Figure 4.6(c) – Result obtained for "Standing" of the actual pattern of the human vs detected pattern (Standing Pattern 3).

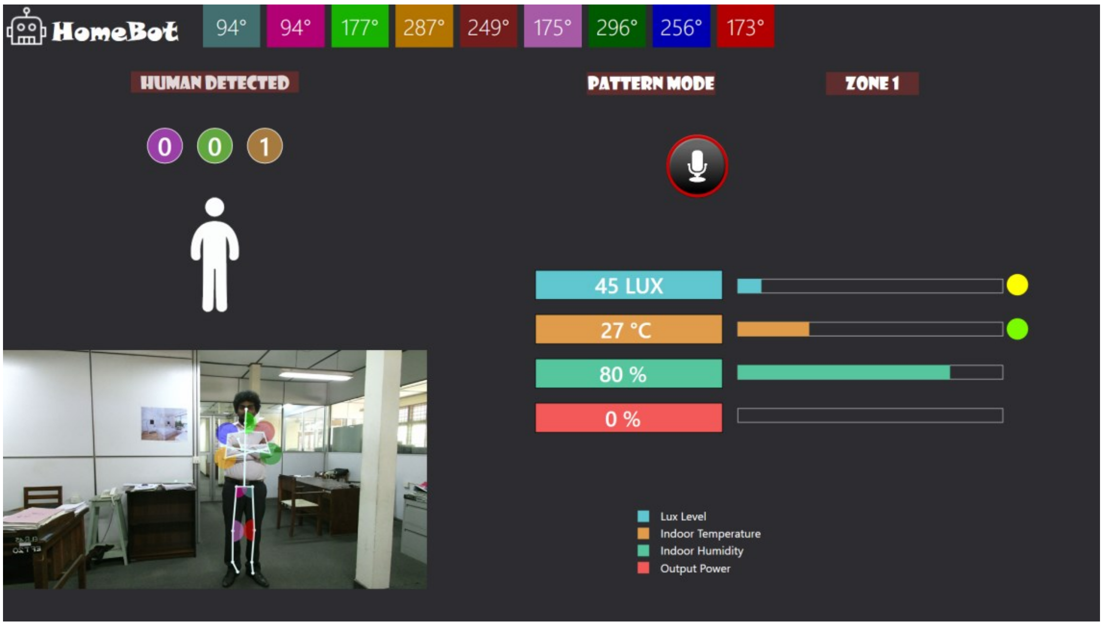

Figure 4.6(d) – Result obtained for "Standing" of the actual pattern of the human vs detected pattern (Standing Pattern 4).

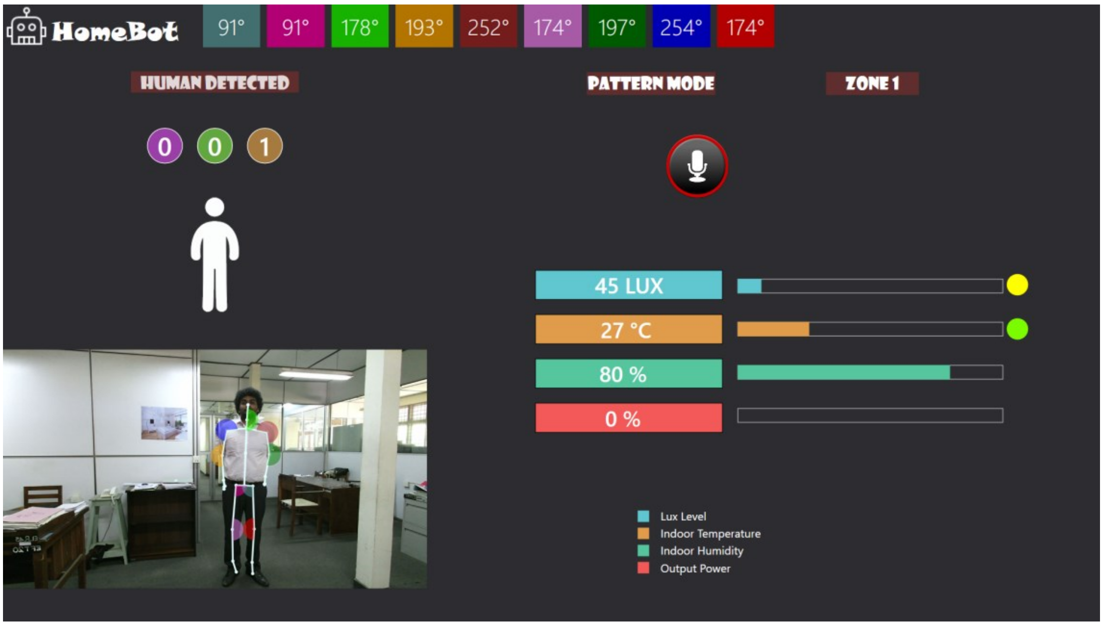

Figure 4.6(e) – Result obtained for "Standing" of the actual pattern of the human vs detected pattern (Standing Pattern 5).

In case (I), the user was in a sitting pose and the corresponding skeleton view captured from Kinect is shown in above figures. The corresponding inputs for the neural network used in the behavioral pattern recognition system were theta is indicated at GUI in deferent colors. The behavior of the user was detected as ‘Sitting’ by the system. For sitting situation, the system had been setup to have a low level of light intensity. Hence, the light intensity level was set as ‘low’ and the other ambient parameters were not affected. In case (II), the user was reading a book. The system detected the user behavior as ‘Reading’ and set the ambient light condition to ‘high’ since it has been defined to have ‘high’ light intensity for the reading behavior. In case (III), the user was in a sitting position. The system detected that behavior as ‘sitting’ and the ambient conditions were set to the given parameters.

The corresponding internal parameters of the behavior pattern recognition system and the ambient conditions are given in Table 4.1 and Table 4.2 for those sample cases.

This validates the capability of the robot companion in adjusting the ambient condition according to the behavior of the user. However, the other ambient parameters such as temperature and humidity have not been varied during these behaviors. It only depend for environment ambient changers. Among them, those parameters are indented to be adjusted to activities like exercising etc. The adjusting of those parameters is proposed for the future work.

Table 4.1: Posture related Light Control

Joint Angles (Degrees) of the Skeleton									Recognized Posture	Current Light Level (Lux)	Status of Light (Particular Zone)
Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7	Θ_8	Θ_9			
187	242	243	186	187	092	093	176	175	Sitting	Very Low	Light on at low Brighter
189	238	225	195	210	149	137	116	129	Setting and Reading	Very Low	Light on at high Brighter
172	244	246	274	251	140	130	128	134	Sanding	Very Low	Light on at low Brighter
187	242	243	186	187	092	093	176	175	Sitting	Low	Light OFF
189	238	225	195	210	149	137	116	129	Setting and Reading	Low	Light on at high Brighter
172	244	246	274	251	140	130	128	134	Sanding	Low	Light OFF

187	242	243	186	187	092	093	176	175	Sitting	High	Light OFF
189	238	225	195	210	149	137	116	129	Setting and Reading	High	Light on at low Brighter
172	244	246	274	251	140	130	128	134	Sanding	High	Light OFF

Table 4.2: Posture and Environment parameter related Fan/AC Control

Joint Angles (Degrees) of the Skeleton									Recognized Posture	Current Temp (°C)	Humid (%)	Status of Fan/AC (Particular Zone)
Θ_1	Θ_2	Θ_3	Θ_4	Θ_5	Θ_6	Θ_7	Θ_8	Θ_9				
187	242	243	186	187	092	093	176	175	Sitting	28	66	Fan/AC ON
189	238	225	195	210	149	137	116	129	Setting and Reading	28	66	Fan/AC ON
172	244	246	274	251	140	130	128	134	Sanding	28	66	Fan/AC ON
187	242	243	186	187	092	093	176	175	Sitting	24	66	Fan/AC OFF
189	238	225	195	210	149	137	116	129	Setting and Reading	24	66	Fan/AC OFF
172	244	246	274	251	140	130	128	134	Sanding	24	66	Fan/AC OFF

4.3. Vocal Interaction System

For testing the vocal interaction system, the users were asked to interact with the system through discussions. The users were explained about the possible voice instructions that could be understood by the robot companion before the experiments. A segment of dialogues between the robot companion and a user is given below.

1) ***User**: What is your name?*
2) ***Robot**: My name is HomeBot. May I help you?*
3) ***User**: What is the time now?*
4) ***Robot**: Time is now 10.30 PM*
5) ***User**: What is the current environment condition?*
6) ***Robo**t: current environment condition is 54 lux in light level, 29 Celsius in Temperature and 75 Percent of percentage humidity.*
7) ***User**: Take Zone 1?*
8) ***Robot**: Zone 1 Recognized.*
9) ***User**: Voice Mode Enable?*
10) ***Robot**: Voice Mode Recognized.*
11) ***User**: Pattern Mode Enable?*
12) ***Robot**: Pattern Mode Recognized.*
13) ***User**: Light ON?*
14) ***Robot**: Light ON.*
15) ***User**: Light OFF?*
16) ***Robot**: Light OFF.*
17) ***User**: Light MID?*
18) ***Robot**: Light MID.*

In the first dialogue, the user asked the name of the robot. Then the robot replied with its name in the second dialogue. Furthermore, the user was asked about any requirements by the robot. In dialogue 3, the user asked the time from the robot. When this question was asked, the robot inquired the Real Time Clock (RTC) module integrated in the system to obtain the current time and replied in dialogue 4. In the dialogue 5, the user asked about the current ambient conditions. The robot replied with dialogue 6 by expressing the lux level, current temperature and humidity obtained from the sensory information. In the dialogue 7, the user provide the command into the robot for identifying zone. The zone number are sown in Figure 4.1. The robot replied with dialogue 8 by recognizing particular zone and configuring smart object of the particular zone for controlling equipment. In the dialogue 9 the user provide the command into the robot for enable the voice controlled operation mode of the robot. . The robot replied with dialogue 10 by voice controlled operation is enabled. In the dialogue, 11 the user provide the command into the robot for enable the pattern controlled operation mode of the robot. The robot replied with dialogue 12 by pattern controlled operation is enabled. In the dialogue, 13 the user provide the command into the robot for switch ON the light in selected zone. The robot replied with dialogue 14 by operation is successfully completed. In the dialogue, 15 the user provide the command into the robot for switch OFF the light in selected zone. The robot replied with dialogue 16 by operation is successfully completed. In the dialogue, 17 the user provide the command into the robot for switch on the light at low Brighter in selected zone. The robot replied with dialogue 18 by operation is successfully completed. Therefore, this validates the ability of the vocal interaction system.

5 CONCLUSION AND RECOMMENDATIONS

5.1. Research Summary

The aim of this research study was to device a novel approach capable of accurately and automatically configuring smart home systems. The main benefit of auto-configuration is that it promotes the evolution of such systems towards the provision of a more comfortable home environment. The literature review indicates that auto-configuration within the smart home environment has generally proven to be a challenging task. The challenge arises from unanticipated changes to external factors such as the status of physical devices and user's preferences, which can significantly affect the system's behavior throughout its lifecycle. This requires considerable effort to maintain such complex systems. These challenges have stimulated the need for dynamic auto-configurable services amongst such distributed systems.

However, an evaluation of the state-of-the-art approaches reveals no satisfactory solutions to achieve auto-configuration and respond to system dynamics without direct user intervention, thus maintaining the quality of the smart home. These solutions are deemed incomplete, as they lack the ability to meet the requirements of the smart home environment, such as flexibility and avoiding the need for direct user intervention. Consequently, it is necessary to develop an efficient, agile and flexible mechanism that adapts to the new and dynamic requirements of the smart home environment without user intervention.

5.2. Conclusion

- This thesis has proposed a novel approach named HomeBot which is an intelligent robot companion that is capable of controlling the ambient conditions based on the behavior of the user has been developed. A vocal interaction system has been incorporated to the design of the robot companion in such a way that it facilitates the operation with the domain of the robot companion such as controlling and monitoring of the home appliances through the voice instructions.

- Prototype stationary robot has been tested with three postures which are sitting, standing and reading at real time. Very large dataset is improved the accuracy of final output. Robot can be recognized these postures very accurately. The neural network has been designed to identify the real time postures of humans.
- Vocal interaction system of the robot is developed by using Microsoft speech API. this system has two parts such as voice recognition and voice responds. Considering voice recognition system, it was checked with same word in different pitches. Robot can be recognized same words in deferent pitches. Voice response generation is implemented using audio hardware of the single board computer and Microsoft text-to-speech API. Robot can generate voice responds with deferent sound formats.
- Communication system of the robot and smart object was based on 2.4Ghz low power Wi-Fi standard with proprietary HTTP protocol. Home broadband router is used to enable the bridge facility of the network. Star topology is used to develop the network function in between robot and smart object. Data encryption and decryption algorithms for protecting network data was developed. Home was divided in to separate zones and implemented smart objects each zones. Overall function of the system has been tested with connecting different devices in to the smart object.

5.3. Recommendations for Future Work

The area of auto-configuration is broad and still largely unexplored. This research study has investigated some issues within this area and envisages some other suitable avenues of investigation. The following paragraphs consider possible improvements to the present approach and the ramifications of this study, which are expected to prove particularly significant for future research into auto-configuration.

- Further work is specifically needed to improve the auto-configuration robot prototype, particularly its architecture, decision-making ability and analytical functionalities. The architecture could be improved by integrating more technologies to build more robust and efficient systems to automate smart home services. The unified framework of elements, descriptions and configuration models could be improved to encompass the semantic richness of these elements and tasks. In particular, increasing the complexity of the inference rules will help the robot to achieve the configuration of more complex systems.

- The Behavior Pattern Recognition (BPR) could also be enhanced to better represent the features and specifications of more complex smart environments and more advanced telecare systems. This BPR would permit the identification of more complex relations amongst services and devices operating in a smart environment, or related to interference with existing devices and user profiles.
- Further scenarios should be investigated to evaluate the applicability of the proposed approach to certain areas such as the safety and security of smart home environments, utilizing the BPR models created here and the individuals of the smart home devices and services. For example, when the house is unoccupied, both the “away from home” security service feature and the energy management service are active. However, these services have incompatible goals, since the energy management service will scheduled the refrigerator and the TV off to save energy, while the security service will switch them on to give the impression that the home is occupied. The conflicting requirements of these two services mean that they are incompatible when attempting to control smart home appliances simultaneously.

6 REFERENCES

[1] Who.int, 'Ageing and health', 2018. [Online]. Available: http://www.who.int/mediacentre/factsheets/fs404/en/ [Accessed: 23- Jun-2018].

[2] Duque Garcia, Ismael “Adapting Robot Behavior in Smart Homes: a Different Approach Using Personas” IEEE RO MAN 2013.

[3] A. Dohr, R. Modre-Opsrian, M. Drobics, D. Hayn, and G. Schreier, "The internet of things for ambient assisted living," in Information Technology: New Generations (ITNG), 2010 Seventh International Conference on, 2010, pp. 804-809.

[4] M. Arndt, S. Wille, L. de Souza, V. F. Rey, N. Wehn, and K. Berns, "Performance evaluation of ambient services by combining robotic frameworks and a smart environment platform," *Robotics and Autonomous Systems,* 2013.

[5] C. Reinisch, M. J. Kofler, and W. Kastner, "ThinkHome: A smart home as digital ecosystem," in *Digital Ecosystems and Technologies (DEST), 2010 4th IEEE International Conference on*, 2010, pp. 256-261.

[6] F. K. Aldrich, "Smart homes: past, present and future," *Inside the smart home,* vol. 1, pp. 17-39, 2003.

[7] S. Schmid, M. Sifalakis, and D. Hutchison, "Towards autonomic networks," in *Autonomic Networking*, ed: Springer, 2006, pp. 1-11.

[8] W. Feng, "Remote service provision for connected homes," 2010.

[9] X. Zhu, Y. Yu, Y. Ou, D. Luo, C. Zhang, and J. Chen, "System Modeling of a Smart-Home Healthy Lifestyle Assistant," in *Agents and Data Mining Interaction*, ed: Springer, 2013, pp. 65-78.

[10] H. Lee and J. Kwon, "The Smart Home Service System Architecture for Healthy and Safe Human Life," 2013.

[11] S. Brownsell, D. Bradley, S. Blackburn, F. Cardinaux, and M. S. Hawley, "A systematic review of lifestyle monitoring technologies," *Journal of telemedicine and telecare,* vol. 17, pp. 185-189, 2011.

[12] D. J. Cook, "How smart is your home," *Science,* vol. 335, pp. 1579-1581, 2012.

[13] C. Ramos, J. C. Augusto, and D. Shapiro, "Ambient intelligence—The next step for artificial intelligence," *Intelligent Systems, IEEE,* vol. 23, pp. 15-18, 2008.

[14] D. J. Cook, J. C. Augusto, and V. R. Jakkula, "Ambient intelligence: Technologies, applications, and opportunities," *Pervasive and Mobile Computing,* vol. 5, pp. 277-298, 2009.

[15] H. Lee and J. Kwon, "The Smart Home Service System Architecture for Healthy and Safe Human Life," 2013.

[16] F. Viani, F. Robol, A. Polo, P. Rocca, G. Oliveri, and A. Massa, "Wireless Architectures for Heterogeneous Sensing in Smart Home Applications: Concepts and Real Implementation," 2013.

[17] I. Khan, A. Mahmood, N. Javaid, S. Razzaq, R. Khan, and M. Ilahi, "Home Energy Management Systems in Future Smart Grids," *arXiv preprint arXiv:1306.1137,* 2013.

[18] J. Paradiso, P. Dutta, H. Gellersen, and E. Schooler, "Guest Editors' Introduction: Smart Energy Systems," *Pervasive Computing, IEEE,* vol. 10, pp. 11-12, 2011.

[19] V. C. Gungor, D. Sahin, T. Kocak, S. Ergut, C. Buccella, C. Cecati*, et al.*, "Smart Grid and Smart Homes: Key Players and Pilot Projects," *Industrial Electronics Magazine, IEEE,* vol. 6, pp. 18-34, 2012.

[20] S. Ahmad, "Smart metering and home automation solutions for the next decade," in *Emerging Trends in Networks and Computer Communications (ETNCC), 2011 International Conference on*, 2011, pp. 200-204.

[21] D. Niyato, L. Xiao, and P. Wang, "Machine-to-machine communications for home energy management system in smart grid," *Communications Magazine, IEEE,* vol. 49, pp. 53-59, 2011.

[22] T. Buracchio, H. H. Dodge, D. Howieson, D. Wasserman, and J. Kaye, "The trajectory of gait speed preceding mild cognitive impairment," *Archives of neurology,* vol. 67, p. 980, 2010.

[23] M. Hogan, "Technophobia amongst older adults in Ireland," 2006.

[24] S. Brownsell and D. Bradley, *Assistive technology and telecare: forging solutions for independent living*: The Policy Press, 2003.

[25] G. Williams, K. Doughty, and D. A. Bradley, "A systems approach to achieving CarerNet-an integrated and intelligent telecare system," *Information Technology in Biomedicine, IEEE Transactions on,* vol. 2, pp. 1-9, 1998.

[26] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer, "Easyliving: Technologies for intelligent environments," in Handheld and ubiquitous computing, 2000, pp. 12-29.

[27] A. Kuutti, A. Dvoryanchikova, A. Lobov, J. L. M. Lastra, and T. Vantera, "A device configuration management tool for context-aware system," in Industrial Informatics (INDIN), 2012 10th IEEE International Conference on, 2012, pp. 10-15.

[28] G. Bin, "An Ontology-based Programming Platform for smart artefact system. ," Unpublished MA thesis vol. Keio University 2009.

[29] T. Perumal, A. R. Ramli, C. Y. Leong, S. Mansor, and K. Samsudin, "Interoperability for smart home environment using web services," International Journal of Smart Home, vol. 2, pp. 1-16, 2008

[30] C. Fabbricatore, H. Boley, and A. P. Karduck, "Machine learning for resource management in smart environments," in Digital Ecosystems Technologies (DEST), 2012 6th IEEE International Conference on, 2012, pp. 1-6

[31] G. Mantas, D. Lymberopoulos, and N. Komninos, "Security in Smart Home Environment," Wireless Technologies for Ambient Assisted Living and Healthcare: Systems and Applications, p. 170, 2010

[32] S. Brownsell, D. Bradley, F. Cardinaux, and M. Hawley, "Developing a systems and informatics based approach to lifestyle monitoring within eHealth: part I-technology and data management," in Healthcare Informatics, Imaging and Systems Biology (HISB), 2011 First IEEE International Conference on, 2011, pp. 264-271.

[33] M. Gritti, M. Broxvall, and A. Saffiotti, "Reactive self-configuration of an ecology of robots," in Proc of the ICRA-07 Workshop on Network Robot Systems, Rome, Italy, 2007.

[34] G. D. Abowd, A. F. Bobick, I. A. Essa, E. D. Mynatt, and W. A. Rogers, "The aware home: A living laboratory for technologies for successful aging," in Proceedings of the AAAI-02 Workshop "Automation as Caregiver, 2002, pp. 1-7.

[35] G. Weiss, Multiagent systems: a modern approach to distributed artificial intelligence: The MIT press, 1999.

[36] S. A. Shafer, B. Brumitt, and J. Cadiz, "Interaction issues in context-aware intelligent environments," Human-Computer Interaction, vol. 16, pp. 363-378, 2001.

[37] Y. Alsafi and V. Vyatkin, "Ontology-based reconfiguration agent for intelligent mechatronic systems in flexible manufacturing," Robotics and Computer-Integrated Manufacturing, vol. 26, pp. 381-391, 2010.

[38] S. Schmid, M. Sifalakis, and D. Hutchison, "Towards autonomic networks," in Autonomic Networking, ed: Springer, 2006, pp. 1-11.

[39] J. Kim, H.-s. Choi, H. Wang, N. Agoulmine, M. J. Deerv, and J. W.-K. Hong, "POSTECH's U-Health Smart Home for elderly monitoring and support," in World of Wireless Mobile and Multimedia Networks (WoWMoM), 2010 IEEE International Symposium on a, 2010, pp. 1-6.

[40] Y. Al-Safi and V. Vyatkin, "An ontology-based reconfiguration agent for intelligent mechatronic systems," in Holonic and Multi-Agent Systems for Manufacturing, ed: Springer, 2007, pp. 114-126.

[41] J. A. Johnsen, "A Semantic Web-driven Approach to Self-Configurating Computer Systems," Norwegian University of Science and Technology, 2006.

[42] T. Gu, H. K. Pung, and D. Q. Zhang, "A service-oriented middleware for building context-aware services," Journal of Network and computer applications, vol. 28, pp. 1-18, 2005.

[43] G. T. McKee, D. I. Baker, and P. S. Schenker, "Task-directed configuration of networked robotic agents," in Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on, 2002, pp. 2752-2757.

[44] L. E. Parker and F. Tang, "Building multirobot coalitions through automated task solution synthesis," Proceedings of the IEEE, vol. 94, pp. 1289-1305, 2006.

[45] R. Lundh, L. Karlsson, and A. Saffiotti, "Dynamic self-configuration of an ecology of robots," in Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, 2007, pp. 3403-3409.

[46] P. Clements, R. Kazman, and M. Klein, Evaluating software architectures: Addison-Wesley Reading, 2001.

[47] J. E. López de Vergara, V. A. Villagrá, C. Fadón, J. M. González, J. A. Lozano, and M. Álvarez-Campana, "An autonomic approach to offer services in OSGi-based home gateways," Computer Communications, vol. 31, pp. 3049-3058, 2008.

[48] M. Rodrıguez-Muro and D. Calvanese, "Quest, an OWL 2 QL reasoner for ontology-based data access," OWLED 2012, 2012.

[49] B. Guo, "An Ontology-based Programming Platform for Smart Artifact Systems," 2009.

[50] J. Shen, L. Wang, and Y. Sun, "Configuration of product extension services in servitisation using an ontology-based approach," International Journal of Production Research, vol. 50, pp. 6469-6488, 2012.

[51] G. Gharbi, M. B. Alaya, C. Diop, and E. Exposito, "AODA: an Autonomic and Ontology–Driven Architecture for service–oriented and event–driven systems," International Journal of Collaborative Enterprise, vol. 3, pp. 167-188, 2013.

[52] T. Heider and T. Kirste, "Smart Environments and Self-Organizing Appliance Ensembles," Mobile Computing and Ambient Intelligence, vol. 5181, 2005.

[53] R. Lundh, L. Karlsson, and A. Saffiotti, "Dynamic self-configuration of an ecology of robots," in Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on, 2007, pp. 3403-3409.

[54] L. Karlsson, "Conditional progressive planning under uncertainty," in IJCAI, 2001, pp. 431-438.

[55] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," Robotics & Automation Magazine, IEEE, vol. 14, pp. 90-103, 2007.

[56] P. Lin, K. Abney, and G. Bekey, "Robot ethics: Mapping the issues for a mechanized world," Artificial Intelligence, vol. 175, pp. 942-949, 2011.

[57] G. A. Bekey, Autonomous robots: from biological inspiration to implementation and control: The MIT Press, 2005.

[58] J.-H. Kim, Y.-D. Kim, and K.-H. Lee, "The third generation of robotics: Ubiquitous robot," in Proc of the 2nd Int Conf on Autonomous Robots and Agents, 2004.

[59] G. Bekey and J. Yuh, "The status of robotics," Robotics & Automation Magazine, IEEE, vol. 15, pp. 80-86, 2008.

[60] E. Garcia, M. A. Jimenez, P. G. De Santos, and M. Armada, "The evolution of robotics research," Robotics & Automation Magazine, IEEE, vol. 14, pp. 90-103, 2007.

[61] C. Buiu, F. Cazan, and R. Ciurlea, "Developing of a Service Robot to Recognize and Sort Waste," in Proceedings of the 16th International Conference on Control Systems and Computer Science, 2007, pp. 22-26.

[62] M. Weiser, "Some computer science issues in ubiquitous computing," Communications of the ACM, vol. 36, pp. 75-84, 1993.

[63] M. Weiser, "The computer for the 21st century," Scientific American, vol. 272, pp. 78-89, 1995.

[64] M. Tenorth, U. Klank, D. Pangercic, and M. Beetz, "Web-enabled robots," Robotics & Automation Magazine, IEEE, vol. 18, pp. 58-68, 2011.

[65] X. Chen, J. Xie, J. Ji, and Z. Sui, "Toward Open Knowledge Enabling for Human-Robot Interaction," Journal of Human-Robot Interaction, vol. 1, pp. 100-117, 2012.

[66] M. Inja, N. Heijne, S. Nugteren, and M. de Waard, "Project ai-the darpa robotics challange-footloose," Project Report, Universiteit van Amsterdam, 2013

[67] support.xbox.com, 'Kinect for Windows v2 or an Xbox Kinect sensor', 2014. [Online]. Available: http://www.microsoft.com/en-us/kinectforwindows/.. [Accessed: 23- Jan- 2018].

[68] msdn.microsoft.com,' Kinect for Windows Sensor Components and Specifications', 2014. [Online]. Available: http://msdn.microsoft.com/en-us/library/jj131033.aspx. [Accessed: 18- Feb- 2018].

[69] msdn.microsoft.com,' Getting the Next Frame of Data by Polling or Using Events', 2014. [Online]. Available: https://msdn.microsoft.com/en-us/library/hh973076.aspx. [Accessed: 22- Feb- 2018].

[70] msdn.microsoft.com,' Color Stream', 2014. [Online]. Available: https://msdn.microsoft.com/en-us/library/jj131027.aspx. [Accessed: 25- Feb- 2018].

7 APPENDIX

APPENDIX A - MATHLAB Function for Developing of Artificial Neural Network

```
function [A] = Test9(in1,in2,in3,in4,in5,in6,in7,in8,in9)
A = myNeuralNetworkFunction([in1;in2;in3;in4;in5;in6;in7;in8;in9]);
end


function [Y,Xf,Af] = myNeuralNetworkFunction(X,~,~)
%MYNEURALNETWORKFUNCTION neural network simulation function.
%
% Generated by Neural Network Toolbox function genFunction, 05-Dec-2017 22:24:21.
%
% [Y] = myNeuralNetworkFunction(X,~,~) takes these arguments:
%
%   X = 1xTS cell, 1 inputs over TS timesteps
%   Each X{1,ts} = 9xQ matrix, input #1 at timestep ts.
%
% and returns:
%   Y = 1xTS cell of 1 outputs over TS timesteps.
%   Each Y{1,ts} = 3xQ matrix, output #1 at timestep ts.
%
% where Q is number of samples (or series) and TS is the number of timesteps.

%#ok<*RPMT0>

% ===== NEURAL NETWORK CONSTANTS =====

% Input 1
x1_step1_xoffset = [0;0;0;0;0;0;0;0;0];
x1_step1_gain                                                                   =
[0.00917431192660551;0.0072202166064982;0.00706713780918728;0.00602409638554217;0.006
2111801242236;0.0122699386503067;0.0127388535031847;0.0111731843575419;0.011363636363
6364];
x1_step1_ymin = -1;

% Layer 1
b1 = [-3.1625132944198859;-7.3862728403466651;-2.6367860349243379;7.328723831161847;-
0.82538657295576856;3.0587174366787631];

IW1_1    =    [0.37444841065763013    -1.4129577512708835    -1.9820179860343781    -
0.96597197929904688  -0.081970406721104505  -1.3326140949796201  -1.2244491798918469  -
1.1187575042779068      -0.87035998362423461;3.6128809018999175      3.2173951205557367
0.97754992065566815    10.338277416233556    4.4493408541859756    18.625609153503092
16.332561518743539   -20.362008645919843   -26.580325001198773;-2.5624568223803195   -
2.5249495374986712   0.22522009190015813   -3.4438289616037419   -3.280031190449014   -
9.3289917171942172    -15.677936280543257    7.2849479523255329    12.278743202759596;-
51.084825520115814     23.015873732625444     30.94750176830598     3.6550679706568587
25.740635753820296    -36.405167885624216    -10.72879717951059    15.671413795987336
17.208346343380416;0.50212152741271698   -1.6101907886173501   -0.91678922467797352   -
```

```
0.44374655763784682    -0.8641754370199376    1.2708542462485748    0.54348842062360714
0.48444568155563666     2.4298446967979626;1.9042255769052392     0.80404686730400665
1.1335955206754409     1.381990267112347     1.1220096113197597     0.99541503615800497
1.2129916341762605 1.0737270518963984 0.31176872838885755];


% Layer 2
b2 = [-1.000000883412842;1.3749353572479331e-05;-1.0000174482665234];
LW2_1   =   [-0.0062422957139925241   1.2885404630670678e-08   1.0000001739369313   -
1.2558787766337351e-09 5.0102954479073749e-08 0.99375878738306456;0.83476939205943046
1.0000001932009654 1.6757095910432272e-07 1.0000000739226675 -9.8544692687228781e-08
-0.16524451785092353;-1.0573694974960179  -1.0000002325309683  -1.000000283480835 5  -
1.0000000871135086 9.1930231633440667e-08 -0.057351951305608759];


% Output 1
y1_step1_ymin = -1;
y1_step1_gain = [2;2;2];
y1_step1_xoffset = [0;0;0];


% ===== SIMULATION ========


% Format Input Arguments
isCellX = iscell(X);
if ~isCellX, X = {X}; end;


% Dimensions
TS = size(X,2); % timesteps
if ~isempty(X)
    Q = size(X{1},2); % samples/series
else
    Q = 0;
end


% Allocate Outputs
Y = cell(1,TS);


% Time loop
for ts=1:TS


    % Input 1
    Xp1 = mapminmax_apply(X{1,ts},x1_step1_gain,x1_step1_xoffset,x1_step1_ymin);


    % Layer 1
    a1 = tansig_apply(repmat(b1,1,Q) + IW1_1*Xp1);


    % Layer 2
    a2 = repmat(b2,1,Q) + LW2_1*a1;


    % Output 1
    Y{1,ts} = mapminmax_reverse(a2,y1_step1_gain,y1_step1_xoffset,y1_step1_ymin);
```

```
end

% Final Delay States
Xf = cell(1,0);
Af = cell(2,0);

% Format Output Arguments
if ~isCellX, Y = cell2mat(Y); end
end

% ===== MODULE FUNCTIONS ========

% Map Minimum and Maximum Input Processing Function
function y = mapminmax_apply(x,settings_gain,settings_xoffset,settings_ymin)
y = bsxfun(@minus,x,settings_xoffset);
y = bsxfun(@times,y,settings_gain);
y = bsxfun(@plus,y,settings_ymin);
end

% Sigmoid Symmetric Transfer Function
function a = tansig_apply(n)
a = 2 ./ (1 + exp(-2*n)) - 1;
end

% Map Minimum and Maximum Output Reverse-Processing Function
function x = mapminmax_reverse(y,settings_gain,settings_xoffset,settings_ymin)
x = bsxfun(@minus,y,settings_ymin);
x = bsxfun(@rdivide,x,settings_gain);
x = bsxfun(@plus,x,settings_xoffset);
end
```

APPENDIX B - C# Cording for Developing interface in- between MATHLAB and C#

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
namespace NeuralInter;
{
    public interface IFindPos
    {
        double[] Test9(double Num1, double Num2, double Num3, double Num4, double
Num5, double Num6, double Num7, double Num8, double Num9);
```

```
    }
}
```

APPENDIX C - C# Cording for Interfacing Kinect into C#

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;

using System.Windows;

using System.Windows.Controls;

using System.Windows.Data;

using System.Windows.Documents;

using System.Windows.Input;

using System.Windows.Media;

using System.Windows.Media.Imaging;

using System.Windows.Navigation;

using System.Windows.Shapes;

using LightBuzz.Vitruvius;

using System.Speech.Recognition;

using System.Speech.Synthesis;

using Newtonsoft.Json;

using Advantech.Adam;

using NeuralNetwork2;

using Microsoft.Kinect;

using System.Windows.Threading;

using System.Threading;

using System.Net.Http;

using Advantech.Common;


namespace HomeBot

{

    /// <summary>

    /// Interaction logic for MainWindow.xaml

    /// </summary>

    public partial class MainWindow : Window

    {
```

```
 FindPos2IFindPos net = new FindPos2IFindPos();

 KinectSensor _sensor;

 MultiSourceFrameReader _reader;

 PlayersController _playersController;

 HomeBotData _SensorData = new HomeBotData();

 DispatcherTimer MainTimer;

 SpeechRecognitionEngine recEngine = new SpeechRecognitionEngine();

private static ComPort Com1 = new ComPort(3);



 private static System.Timers.Timer ComTimer;

 private static System.Timers.Timer NetTimer;



 JointType _start1 = JointType.ShoulderRight;

 JointType _center1 = JointType.ElbowRight;

 JointType _end1 = JointType.WristRight;


 JointType _start2 = JointType.ElbowLeft;

 JointType _center2 = JointType.ShoulderLeft;

 JointType _end2 = JointType.SpineShoulder;


 JointType _start3 = JointType.AnkleRight;

 JointType _center3 = JointType.KneeRight;

 JointType _end3 = JointType.HipRight;


 JointType _start4 = JointType.HipLeft;

 JointType _center4 = JointType.KneeLeft;

 JointType _end4 = JointType.AnkleLeft;


 JointType _start5 = JointType.SpineShoulder;

 JointType _center5 = JointType.ShoulderRight;

 JointType _end5 = JointType.ElbowRight;


 JointType _start6 = JointType.WristLeft;

 JointType _center6 = JointType.ElbowLeft;
```

```
JointType _end6 = JointType.ShoulderLeft;


JointType _start7 = JointType.Head;

JointType _center7 = JointType.Neck;

JointType _end7 = JointType.SpineShoulder;


JointType _start8 = JointType.SpineBase;

JointType _center8 = JointType.HipLeft;

JointType _end8 = JointType.KneeLeft;


JointType _start9 = JointType.KneeRight;

JointType _center9 = JointType.HipRight;

JointType _end9 = JointType.SpineBase;



public MainWindow()

{

    InitializeComponent();

    MainTimer = new DispatcherTimer();

    MainTimer.Interval = new TimeSpan(0,0,0,0,500);

    MainTimer.Tick += MainTimer_Tick;



    // TimerCallback tmCallback = SensorCom;

    // Timer timer = new Timer(tmCallback,null, 1000, 1000);

    //Console.WriteLine("Press any key to exit the sample");

    // Console.ReadLine();


    //_dtTimer = new DispatcherTimer();

    /// _dtTimer.Tick += new System.EventHandler(SensorCom);

    // _dtTimer.Interval = new TimeSpan(0, 0, 0, 1); //Timespan of 2 seconds

    //  _dtTimer.Start();


    ComTimer = new System.Timers.Timer(2000);

    // Hook up the Elapsed event for the timer.

    ComTimer.Elapsed += ComTimer_Elapsed;
```

```
            ComTimer.AutoReset = true;

            ComTimer.Enabled = true;


            NetTimer = new System.Timers.Timer(2000);

            NetTimer.Elapsed += NetTimer_Elapsed;

            NetTimer.AutoReset = true;

           // NetTimer.Enabled = true;


            ComOpen();

            MainTimer.Start();


            Thread.Sleep(3000);




        }

void  Reader_MultiSourceFrameArrived(object  sender,  MultiSourceFrameArrivedEventArgs 
e)

        {

            var reference = e.FrameReference.AcquireFrame();


            // Color

            using (var frame = reference.ColorFrameReference.AcquireFrame())

            {

                if (frame != null)

                {

                    if (viewer.Visualization == Visualization.Color)

                    {

                        viewer.Image = frame.ToBitmap();

                    }

                }

            }


            // Body

            using (var frame = reference.BodyFrameReference.AcquireFrame())

            {

                if (frame != null)
```

```
                {
                    var bodies = frame.Bodies();

                    _playersController.Update(bodies);

                    Body body = bodies.Closest();

                    if (body != null)
                    {
                        viewer.DrawBody(body);

                        angle1.Update(body.Joints[_start1],     body.Joints[_center1],
body.Joints[_end1], 50);
                        angle2.Update(body.Joints[_start2],     body.Joints[_center2],
body.Joints[_end2], 50);
                        angle3.Update(body.Joints[_start3],     body.Joints[_center3],
body.Joints[_end3], 50);
                        angle4.Update(body.Joints[_start4],     body.Joints[_center4],
body.Joints[_end4], 50);
                        angle5.Update(body.Joints[_start5],     body.Joints[_center5],
body.Joints[_end5], 50);
                        angle6.Update(body.Joints[_start6],     body.Joints[_center6],
body.Joints[_end6], 50);
                        angle7.Update(body.Joints[_start7],     body.Joints[_center7],
body.Joints[_end7], 50);
                        angle8.Update(body.Joints[_start8],     body.Joints[_center8],
body.Joints[_end8], 50);
                        angle9.Update(body.Joints[_start9],     body.Joints[_center9],
body.Joints[_end9], 50);

                        tblAngle1.Text = ((int)angle1.Angle).ToString();
                        tblAngle2.Text = ((int)angle2.Angle).ToString();
                        tblAngle3.Text = ((int)angle3.Angle).ToString();
                        tblAngle4.Text = ((int)angle4.Angle).ToString();
                        tblAngle5.Text = ((int)angle5.Angle).ToString();
                        tblAngle6.Text = ((int)angle6.Angle).ToString();
                        tblAngle7.Text = ((int)angle7.Angle).ToString();
                        tblAngle8.Text = ((int)angle8.Angle).ToString();
                        tblAngle9.Text = ((int)angle9.Angle).ToString();
```

```
                    _SensorData.neck = Convert.ToInt16(tblAngle7.Text);

                    _SensorData.shoLeft = Convert.ToInt16(tblAngle2.Text);

                    _SensorData.shoRight = Convert.ToInt16(tblAngle5.Text);

                    _SensorData.ElboLeft = Convert.ToInt16(tblAngle6.Text);

                    _SensorData.ElboRight = Convert.ToInt16(tblAngle1.Text);

                    _SensorData.HipLeft = Convert.ToInt16(tblAngle8.Text);

                    _SensorData.HipRight = Convert.ToInt16(tblAngle9.Text);

                    _SensorData.KneeLeft = Convert.ToInt16(tblAngle4.Text);

                    _SensorData.KneeRight = Convert.ToInt16(tblAngle3.Text);



                }

            }

        }


    }
```

APPENDIX D - C# Cording for Getting sensor information's

```
using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using System.Threading.Tasks;


namespace HomeBot

{

    class HomeBotData

    {

        public  double[] NuralData;


        public  int neck;

        public  int shoLeft;

        public  int shoRight;
```

```
public  int ElboLeft;

public  int ElboRight;

public  int HipLeft;

public  int HipRight;

public  int KneeLeft;

public  int KneeRight;


public static double lux;

public static double Tempatrature;

public static double Humidity;

public static double OutputPower;


public static string RStr;


public  int sit;

public  int sitR;

public  int sta;



public  bool HumenState;


public  bool NotHumanDetect = true;

public  bool HumanDetect = true;


public  bool sitting = true;

public  bool sitReading = true;

public  bool standing = true;

public  bool Invalid = true;



public  bool VoiceMode = false;

public  bool PatternMode = false;

public  bool EnableZone1 = false;

public  bool EnableZone2 = false;
```

```
public  bool Light1On = false;

public  bool Light1Off = false;

public  bool Light1Mid = false;


public bool Light2On = false;

public bool Light2Off = false;

public bool Light2Mid = false;


public bool AutoMode = false;

public bool ManualMode = false;




public bool Zone1LightH = true;

public bool Zone1LightL = true;


public bool Zone2LightH = true;

public bool Zone2LightL = true;




//light and temp setting


public bool LightH = false;

public bool LightM = false;

public bool LightL = false;


public bool TempH = false;

public bool TempL = false;


//temp Alert System

public bool TempAlHi = true;

public bool TempAlLo = true;


//light level Alert System

public bool LightAlHi = true;
```

```
        public bool LightAlMd = true;

        public bool LightAlLo = true;


        public bool LightAlHi1 = true;

        public bool LightAlMd1 = true;

        public bool LightAlLo1 = true;


        public bool LightAlHi2 = true;

        public bool LightAlMd2 = true;

        public bool LightAlLo2 = true;


        public bool LightAlHi3 = true;

        public bool LightAlMd3 = true;

        public bool LightAlLo3 = true;


        public bool LightAlHi4 = true;

        public bool LightAlMd4 = true;

        public bool LightAlLo4 = true;


        public bool LHi = true;

        public bool LMd = true;

        public bool LLo = true;




    }

}
```

APPENDIX E - C++ Cording for interfacing Smart Object

```
#include<ESP8266WiFi.h>

const char* ssid = "xxx";

const char* password = "xxxxxxx";


WiFiServer server(80);

int val = 0;

int dimming = 128;


void setup() {

        Serial.begin(9600);


  pinMode(D1, OUTPUT);// Set AC Load pin as output

  pinMode(D5, INPUT_PULLUP);

  attachInterrupt(digitalPinToInterrupt(D5), zero_crosss_int, RISING);


        Serial.println();

        Serial.println();

        Serial.print("Connecting to network");

        Serial.println(ssid);

        WiFi.mode(WIFI_STA);

        //WiFi.hostname("Smart Lamp");

        //WiFi.enableSTA(1);

        WiFi.begin(ssid, password);

        WiFi.config(IPAddress(192,  168,  43,  105),  IPAddress(192,  168,  43,  105),
IPAddress(255, 255, 255, 0));


        while (WiFi.status() != WL_CONNECTED) {

                delay(500);

                Serial.print(".");

        }

        Serial.println("");

        Serial.println("WiFi connected");


        // Start the server
```

```
        server.begin();

        Serial.println("Server started");



        // Print the IP address

        Serial.println(WiFi.localIP());

}

void zero_crosss_int()  //function to be fired at the zero crossing to dim the light

{



  int dimtime = (75*dimming);    // For 60Hz =>65

  delayMicroseconds(dimtime);    // Wait till firing the TRIAC

  digitalWrite(D1, HIGH);   // Fire the TRIAC

  delayMicroseconds(10);                         //   triac   On   propogation   delay
// (for 60Hz use 8.33) Some Triacs need a longer period

  digitalWrite(D1, LOW);     // No longer trigger the TRIAC (the next zero crossing
will swith it off) TRIAC


}

void loop() {


        WiFiClient client = server.available();

        if (!client) {


                return;

        }


        // Wait until the client sends some data

        Serial.println("new client");

        while (!client.available()) {

                delay(1);

        }


        // Read the first line of the request

        String req = client.readStringUntil('\r');
```

```
        Serial.println(req);
        client.flush();

        // Match the request

        if (req.indexOf("/gpio/0") != -1) {
                val = 128;
        }
        else if (req.indexOf("/gpio/1") != -1) {
                val = 96;
        }

        else if (req.indexOf("/gpio/2") != -1) {
                val = 64;
        }

        else if (req.indexOf("/gpio/3") != -1) {
                val = 32;
        }

        else if (req.indexOf("/gpio/4") != -1) {
                val = 0;
        }
        else if (req.indexOf("/gpio/5") != -1) {
                String p = "HTTP/1.1 200 OK\r\nContent-Type: text/html\r\n\r\n<!DOCTYPE HTML>\r\n<html>\r\n";
                p += (val) ? "ON" : "OFF";
                p += "</html>\n";

                // Send the response to the client
                client.print(p);
                delay(1);
                return;
        }

        else {
```

```
                Serial.println("invalid request");
                client.stop();

                return;
        }
        // Set GPIO2 according to the request
        //analogWrite(D2, val);
 dimming = val;

        client.flush();

        // Prepare the response
        String  s  =  "HTTP/1.1  200  OK\r\nContent-Type:  text/html\r\n\r\n<!DOCTYPE
HTML>\r\n<html>\r\nLIGHT is now ";
        s += (val) ? "ON" : "OFF";
        s += "</html>\n";

        // Send the response to the client
        client.print(s);
        delay(1);
        Serial.println("Client disonnected");

        // The client will actually be disconnected
        // when the function returns and 'client' object is detroyed

}
```