INTELLIGENT FALL DETECTION AND NOTIFICATION SYSTEM FOR AN IOT BASED SMART HOME ENVIRONMENT

Nayakasinghe Mudiyanselage Tharushi Kalinga

(198045M)

Degree of Master of Science by Research

Department of Electrical Engineering

University of Moratuwa

Sri Lanka

January 2021

INTELLIGENT FALL DETECTION AND NOTIFICATION SYSTEM FOR AN IOT BASED SMART HOME ENVIRONMENT

Nayakasinghe Mudiyanselage Tharushi Kalinga

(198045M)

Thesis submitted in partial fulfillment of the requirements for the degree Master of Science by Research

Department of Electrical Engineering

University of Moratuwa
Sri Lanka

January 2021

DECLARATION

I declare that this is my own work and this dissertation 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 the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my dissertation, 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 MSc thesis under my supervision.

Signature of the Supervisor(s): Date:

Prof.A.G.B.P.Jayasekara

Dr.G.I.U.S.Perera

Abstract

Throughout the history of technology, various mechanisms to support the elderly and the disabled have been introduced as a remedy for the inadequacy of caregivers to provide them with the required assistance in leading an independent and secure living. Among all those mechanisms, smart homes and social robotics appear to play a significant and effective role in assuring a comfortable and safe environment for the elderly and the disabled who prefer to live independently without causing an extra burden on their families.

However, most of the existing assistive systems lack the required levels of accuracy and timeliness which causes increased probability of resulting them in higher risk of damage after encountering an emergency while staying alone at their homes. Therefore, in order to ensure the availability of timely assistance and support, the introduction and development of effective emergency detection and notification systems is an essential necessity in the present world.

This research work introduces a Smart Home System consisting of three subsystems integrated together over an IoT Cloud with the main objective of improving the quality of life of the elderly and the disabled by providing them with ample support in performing their activities of daily living without compromising safety and independence.

The proposed system presents a novel vision based method of detecting falls from standing or walking positions that is also capable of distinguishing the identified falls among three types so that the medical attention could be easily focused. A special subsystem is also introduced for the identification of sitting postures and detection of falls from wheelchairs for the people with mobility impairments.The fall detection and posture identification are carried out with a social robot called MIRob which receives visual input through a Microsoft Kinect Sensor. A novel emergency notification system is also presented where, the notification is performed by implementing a Q-Learning algorithm using a Reinforcement Learning agent via an Android application. Through experimental studies the overall proposed system has promised to guarantee acceptable levels of accuracy and timeliness in providing assistance to the elderly and the disabled.

Keywords- **Fall Detection, Posture Identification, Emergency Notification, Wheelchair Users, Social Robotics, Smart Homes, Independent Living**

DEDICATION

To my family,

for always loving and supporting me...

ACKNOWLEDGMENTS

First and foremost, I would like to express my sincere gratitude to my supervisors, Prof.A.G.B.P.Jayasekara and Dr.G.I.U.S.Perera for the continuous support, motivation and guidance offered throughout my Masters study and research.

I would also like to thank my review panelists, Dr.Kutila Gunasekara and Prof.Chandima Pathirana for their insightful suggestions that encouraged better quality and performance in my research work.

My sincere thanks also goes to the members of the thesis committee for enlightening me with invaluable comments for the improvement of my thesis.

I would also like to extend my heartfelt gratitude to my colleagues at the Robotics and Automation Laboratory, Department of Electrical Engineering, University of Moratuwa for their immense support and valuable opinions.

Last but not least, I would like to thank my family for the unconditional love and strength provided in achieving my dreams throughout the journey of life.

This research work was partially supported by the Center for Advanced Robotics.

Tharushi Kalinga

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background

Elderly is found to be the fastest growing segment of the world population. According to the global statistics, in 2015, one in each eight persons worldwide had aged over 60 years or above and it is anticipated this to become one in every six persons by 2030 and one in every five persons by the middle of the 21^{st} century [1]. The Figure 1.1 shows the trends in global population growth by age groups. The bottom most band represents the global share of children aged up to 14 years. It is seen that this share has decreased from 37% in 1960 to 26% in 2018 and is expected reduce to 21% in 2050. The top most band represents the

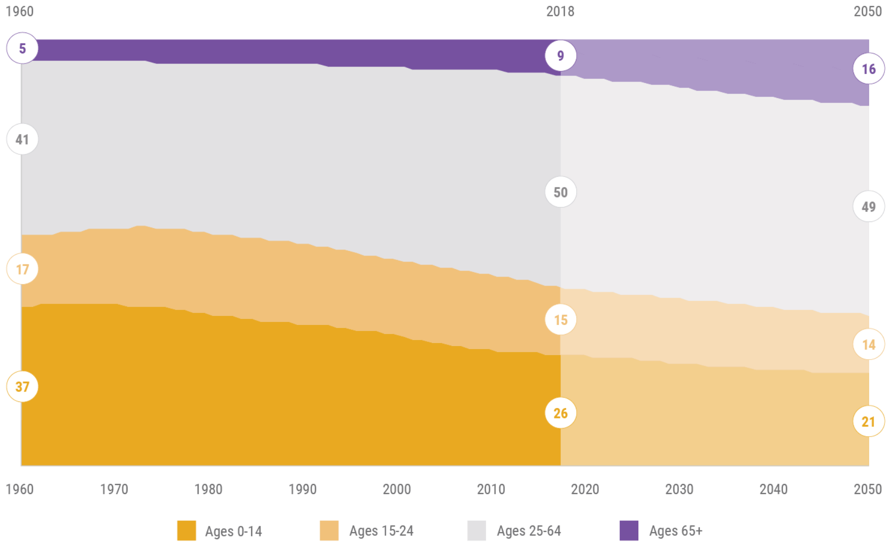

Figure 1.1: Percent of Population by Age Group, 1960 - 2050 [2]

global growth of older adults aged 65 and above. It is observable that this share has increased from 5% in 1960 to 9% in 2018 and is expected to rise to 16% by 2050. Thus, it is quite clear that the world is experiencing a drastic growth in elderly population.

The demographical projections suggest population ageing in all countries causing effects widely on the social, economical and health systems, out of which the health systems are made to encounter major complications [3]. According to the World Health Organization (WHO), over one billion people accounting for around 15% of the world population have been found to live with some form of disability and about 2-4% of them suffers from considerable difficulties in functioning. The WHO predicts the prevalence of disability to increase to greater extents in the coming years, due to the higher ageing populations, increased disability risks in adults and ever-growing rise in chronic health conditions such as cardiovascular diseases, diabetes, and cancer etc. They also state that almost every person will be either permanently or temporary impaired at-least once in their lifetime. They have more over identified that people who live to older ages have an increased probability of facing difficulties in functioning.

The International Classification of Functioning, Disability and Health (ICF), categorizes the problems with human functioning into 3 areas as impairments, activity limitations and participation restrictions. The impairments are the problems in body functions or alterations in body structures such as paralysis or blindness. Difficulties in executing activities such as walking or eating are included in activity limitations. Participation restrictions involve the problems with the engagement in any area of life such as facing discrimination in employment or transportation. Accordingly, disability is referred to as the difficulties encountered in any or all these 3 areas of functioning [4].

In order to facilitate the elderly and the disabled with a safe and comfortable living, and to minimize the negative consequences of the possible emergency situations encountered by them, the availability of timely assistance and support mechanisms is necessary. Thus, the development of effective emergency detection and notification systems is an essential requirement in today's world.

1.2 Problem Statement

Majority of the persons with disabilities and the older adults require assistance and support to achieve a better quality of living by being able to participate in the basic social and economic lives in a way experienced by the others [5]. It must be noticed that ageing and disability not only affect the person being concerned, but also the whole support system of family and friends, who usually need to take care of the older person or the person with the disability.

In order to provide a reasonable standard of living for the disabled people, their families require to meet extra costs on health services, medication, help with daily activities, and disability-specific aids such as assistive devices, personal assistance and house adaptation etc. [6]. While providing care, the care givers are often found to face challenges in physical, emotional, and financial aspects, which are collectively known as the caregiver burden. It is found that the caregivers often live with a huge amount of stress due to increased time spent with the disabled person, increased housework, and lack of sleep etc. and they usually feel isolated and alone [7].

As a result of all these issues the modern world is facing a lack of adequate number of caregivers to provide the necessary support and assistance to accomplish a safe and comfortable living for the elderly and the disabled. Thus, they most of the times need to stay home alone making them more vulnerable to emergency situations. Even though timely detection and notification systems can reduce the negative consequences caused from such occasions, there exists a shortage of suitable systems that provide the required level of accuracy, timeliness, and non-obtrusiveness.

1.3 Thesis Overview

This thesis consists of 6 chapters. Each chapter is briefly described below.

Chapter 1: Provides an introduction to the thesis. It discusses the background that motivated this research and also presents the problem statement addressed by this research work.

Chapter 2: Presents a literature review on related concepts and past research work that are concerned on addressing similar problems.

Chapter 3: Illustrates the proposed Detection System for Falls from Standing or Walking Positions. In the later part of the chapter the system performance is also discussed using experimental results.

Chapter 4: Discusses the proposed System for Posture Identification and Fall Detection of Wheelchair Users. The performance of the system is also presented along with experimental observations and results.

Chapter 5: Describes the proposed Emergency Notification System. The later part of the chapter reveals the performance results with the help of experimental studies.

Chapter 6: Concludes the thesis by providing an evaluation of the overall proposed system along with the observable limitations and possible future developments.

Chapter 2

LITERATURE REVIEW

2.1 Fall Detection Systems

Falling, generally defined as inadvertently coming to rest on the ground, floor or other lower level, excluding intentional change in position to rest in furniture, wall or other objects [8], is the most dominant cause leading to both fatal and non-fatal injuries among people aged 65 and above [9]. The complications with mobility, balance, and muscle strength and the chronic health conditions requiring various kinds of medications often results in increased risk of falling among the adults [10].

It is estimated that 30% of people aged above 65 and 50% of those above 85, falls at least once every year [11]. It is also found that those falls can cause moderate to severe injuries like hip fractures and head traumas [12]. In fact, 23% of injury-related deaths of people older than age 65 and 34% of those older than age 85 are found to occur after a fall [13].

Furthermore Chaccour et al. [14] defines a fall detection system as an assisting device that is capable of sensing, processing and communicating alarm data in the event of a fall under real-life conditions effectively. Thus, real-time detection of falls can help immediate intervention of caregivers and timely treatments resulting in greater probability of saving lives, reducing medical expenses and lowering anxiety within the adult [15].

In order to mitigate the impacts and minimize the consequences caused by falls, many researchers have extensively studied them for over two decades [14] and it has been found that older adults living alone are especially prone to delayed help in the case of falls and it is necessary to have an automated fall detection and notification system to provide the required assurance and timely help [16].

Sposaro et al. [17] has presented an alert system for fall detection using an Android based smart phone with an integrated tri-axial accelerometer. Once a fall is detected, a notification is raised requiring response of the user. If the user fails to respond, the system alerts a predefined list of social contacts. If the social contacts do not respond or confirm a fall, the system alerts an emergency service.

Villaverde et al. [18] has introduced a Machine Learning based fall detection method using a smartwatch accelerometer. The same method has been used in [19] to detect falls by developing a home telecare system. Once a fall is detected an AI-based robot navigates to the location and provides web-based assistance from caregivers by sharing real-time videos and allowing communication. Another smartwatch-based telecare system has been presented in [20], where a tri-axial accelerometer inside the smartwatch detects falls and sends notification to a predefined number.

A fall detection system by having a tri-axial accelerometer based wearable device around the elder's waist has been presented in [21]. As soon as a fall is detected, a GPS/GSM module automatically sends help request to caregivers with patient's location. Santiago et al. [22] has proposed a fall detection system using a wearable device with a motion sensor. Once a fall is detected, it sends an alert to a mobile phone, which notifies a user-defined list of emergency contacts. A robot-assisted emergency system to support elderly living alone has been discussed in [23]. Falls are detected by a wearable device with motion sensors to which a robot is wirelessly connected. Another wearable device for elderly fall detection by the fusion of a tri-axial accelerometer, a gyroscope and a magnetometer has been developed in [24].

Shoaib et al. [25] presents an assistant living environment for the elderly using a network of video cameras. A fall is detected by considering the location of the head with respect to the floor. Once a fall is detected, the system tries to confirm it with the monitored elder and if a positive response or no response is received from the elder, the emergency situation is notified to responsible parties.

A home elder care system where falls are detected by considering skeleton data obtained using a Kinect Sensor has been introduced in [26]. The system identifies an incident as a fall if the acceleration and distance change of Centre of Mass and the recovery time exceeds some defined thresholds and sends alarm to nurses' phones if a fall gets detected. Yang et al. [27] has presented an indoor fall detection method for the elderly using 3D depth images from a Kinect Sensor. The direction of the individual is determined by coefficients of ellipses and a fall gets detected when both the distance from centroids of the body to the floor plane and the angle between the body and the floor plane are lower than some threshold values. A fall detection method using height and velocity of body obtained from depth images generated by a Kinect sensor has been introduced in [28]. Here, the distance from head to floor of the subject is considered as the height and the velocity of the shoulder centre of the subject is considered as the velocity. A fall is identified when the velocity increases beyond a threshold and the height decreases below the previous position of knee joint.

Similar to the above, many researches have been carried out in order to detect and notify falls timely and effectively. However, most of the already proposed wearable devices are obtrusive and uncomfortable and most of the already introduced vision-based methods are unable to satisfy both accuracy and robustness [29].

2.2 Emergency Notification Systems

It is found that, most the elderly people as well as the disabled people prefer to live independently at their own homes which is a familiar environment for them [30]. But, as living alone does not guarantee safety all the times, they are always more vulnerable to emergency incidents such as fall and unconsciousness [25]. In order to provide timely assistance for the home-bound elders and disabled in such situations, it is required to have a constant and reliable emergency notification system which can quickly inform the responsible parties.

Lin et.al [31] introduces a stray prevention system for elders suffering from dementia. It monitors indoor, outdoor, emergency as well as remote locations. Whenever, an incident that requires assistance is detected, a message is sent to the relevant parties such as the caretakers, search volunteers, and family members.

The system called iCare presented in [32], serves as a real-time health monitoring system for the elderly by using wireless body sensors that communicate with a smartphone over Bluetooth, and also as a living assistant for the elderly by providing functions like regular reminders, quick alarms, and medical guidance etc. When the system detects an emergency, the smart phone automatically alerts the emergency center as well as a preassigned list of people including the elder's family and friends. A similar system has been presented in [33] where wearable sensors have been used to monitor the heart rate, body temperature, ECG, respiration rate, tilt and fall of the elders. The data from these sensors are sent to a smartphone that notifies the caretaker during critical situations.

A user-friendly emergency response system for the elderly has been invented in [34]. Once the user triggers an emergency call function, the mobile phone automatically notifies a predetermined list of emergency contacts along with information including current location, and health information such as heartbeat, blood pressure, etc..

Solórzano et.al [35] discusses the design and implementation of a Home Teleassistance system for the elderly and the disabled who live alone or have poor access to emergency services. The details regarding their personal information and the geographical location are sent to their relatives and emergency services using email and SMS, in order to achieve an accelerated response in case of an emergency.

Johnson et. al [36] have designed a mobile robot integrated with smart household technology that monitors the environment as well as the health and behavior of elders and issues alerts to caregivers and emergency personnel during dangerous situations through video communication over the internet. Another robot that acts as a companion for the elderly while ensuring their safety by providing reminders for medication and generating notifications to caregivers during emergencies has been proposed in [37].

2.3 Mobility Disability and Use of Wheelchairs

Mobility is defined as the ability to move and is considered to be one of the most important physical functions essential in achieving an independent living [38]. Independent mobility is critical to individuals of all age, and mobility disability is a highly dynamic process among older persons [39].

Mollaoğlu et.al [40] has studied about the relationship between mobility disability and life satisfaction, which are two of the most important properties related to life quality. They have found that mobility disability was prevalent among majority of the elderly population and has a significant effect on the life satisfaction of them. They have also established that once elderly people are concerned, mobility was affected by the factors such as their age, gender and chronic diseases while life satisfaction was related to their age, level of education and level of health perception.

In the United States, it has been found that 61 million adults live with some sort of a disability and 13.7 % of people with a disability have a mobility disability with serious difficulty walking or climbing stairs [41].

Statistics show that 17 % of world's population is affected by some form of blindness or visual impairment while 6% has deafness or hearing loss. It has also been found that 2.6% of the global population suffer from some form of intellectual disability while 1% of the world need a wheelchair on a daily basis [42]. It is important to note that 1% of the population of world approximately accounts for around 75 million people. Figure 2.1 shows a Manual Wheelchair, which is considered to be one of the most commonly used assistive devices to enhance personal mobility [43].

Several factors that hinder the participation of persons with mobility impairments in public activities have been identified in the literature. The factors such as low quality of wheelchairs and lack of confidence with wheelchair use were particularly noted in [44]. Thus, the solutions designed to solve the problem of mobility disability, should also focus on the restoration and maintenance of independence, as mobility is a precondition for enjoying human rights and living with dignity.

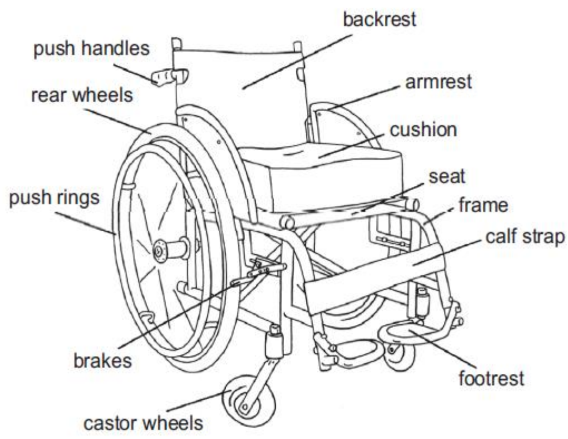

Figure 2.1: Manual Wheelchair [43]

Figure 2.2: George Klein with the World's First Power Wheelchair [45]

So, with that in mind, the world's first power wheelchair shown in Figure 2.2, which was a technological advancement from the manual wheelchair, was invented by George Klein from National Research Council of Canada [45]. This was mainly to assist the people with quadriplegia injured in World War II. George Klein was able to develop a unique set of technologies including the joystick, tighter turning systems and separate wheel drives that are still continuing to be important in developing power wheelchairs.

The Figure 2.3 shows some modern Power Wheelchairs.However, it has been found from a survey that, though the power wheelchair assures to satisfy the needs of certain persons with disabilities, around 40% of the disabled community undergoes difficulty in operating the power wheelchair and between 5-9% find it impossible to operate the power wheelchair without assistance [46].

Figure 2.3: Some Modern Power Wheelchairs [48]

With the intention of overcoming these problems, the Smart wheelchair was introduced. The smart wheelchair was created using the technologies originally developed for mobile robots. It was able to serve as one of the potential solutions for the shortcomings brought about by the power wheelchair and reduced the need for physical, perceptual, and cognitive skills that are required to operate a power wheelchair [47]. Leaman et.al [48] have suggested that the best possible smart wheelchair needs to have the ability to support people with all types of disabilities by utilizing a combination of computer vision, touch, voice, brain control etc.

2.4 Falls from Wheelchairs

Forslund et.al [49] have studied about recurrent falls and fall-related injuries in wheelchair users with spinal cord injury by monitoring a sample of 149 participants over a year. Almost two-thirds (64%) of the participants has fallen and nearly one-third (32%) has fallen recurrently. They have identified wheelchair transfers (to bed/ sofa/ car/ toilet) and pushing of wheelchair (over flat/ uneven grounds) as the most common situations of causing falls from wheelchairs. Further, they have encountered 70 fall-related injuries out of which 67% has been minor involving mostly bruises and scratches, 23% has been moderate causing strains or sprains and 10% has been severe with femoral or tibiae fractures.

Chen et.al [50] have interviewed 95 active wheelchair users regarding wheelchair-related accidents encountered during last 3 years prior to the interview and have found that 54.7% of them have undergone at least 1 accident and 16.8% have faced 2 or more accidents. They have categorized those accidents into 3 types based on the mechanism of the accidents as tips and falls (87.8%), accidental contact(6.8%), and dangerous operations(5.4%). Tips and falls included falling of the wheelchair user with or without reported tipping of the wheelchair. Collisions with mobile or immobile objects not resulting in tips or falls were included in the accident contact category. Dangerous operations were defined involving situations where users are unable to control the wheelchair properly. Out of the reported accidents, 55.4% has resulted in some form of injury with abrasion or laceration (70.7%) as the most common type of injury followed by sprain, head injury, fracture and organ injury. They have also identified that most injuries are located in the upper and lower extremities.

Many researchers have acknowledged wheelchair-related accidents as a significant problem and that a majority of injuries resulted from those accidents are due to tips and falls.

Gavin et.al [51] states that, a wheelchair tip or a fall has the ability to impact the morbidity and mortality of an individual who relies on a wheelchair for mobility by affecting the function, activity, independence, and quality of life. Further they show that serious injuries such as fractures that are resulted from a tip or a fall can lead to extended hospital stay, or extended bed rest, or loss of strength due to immobilization. In fact, they also state that approximately one wheelchair-related death is reported every week in the United States, out of which a majority is due to falls.

A lot of practices such as user precautions, wheelchair modifications, healthcare provider interventions, and environmental modifications are being already undertaken to prevent tips and falls from wheelchairs, but if any such accident takes place causing the user to fall, it is important to have a quick fall detecting mechanism in order to minimise negative consequences.

Sung et.al [52] have studied a group of full time wheelchair users with spinal cord injury and multiple sclerosis and have analyzed the most recent falls undergone by them. They have revealed transferring, wheelchair propulsion, walking short distances, reaching for an object and standing as the 5 main actions causing falls from wheelchairs. Fall detection of elderly is a major public health problem and the earlier a fall is detected the lower the consequences [53].

Ma et.al [54] have developed a smart wheelchair equipped with pressure sensors to detect abnormal sitting postures of wheelchair users in order to avoid falls and wheelchair overturns. They have used supervised machine learning to distinguish among 6 different sitting postures namely; nobody sitting, proper sitting, lean-forward, lean-backward, lean-left, and lean-right. They have further developed their system by attaching inertial measurement units on the user's wrists to extract features of upper limb motion as well [55].

With the aim of creating a healthier and more active lifestyle for the wheelchair users, Hiremath et.al [56] have developed a system using gyroscopes to detect wheelchair overturns and accelerometers worn on upper arm to monitor physical activities.

2.5 Vitruvius Framework

Vitruvius is an advanced 3D motion framework released by Microsoft and Channel 9. The features of this third-party library are available for a wide range of cameras including Intel RealSense D415 & D435, Orbbec Astra & Astra Pro and Microsoft Kinect v2. As the Vitruvius framework is able to provide a series of useful functional characteristics for the analysis of body pose, joints, rotation of a segment in the 3D space, and its distance from the camera etc. [57] [58], it has been used by many researches to carry out their research work.

A Socially Assistive Robot that coaches older adults in physical exercises has been introduced in [57]. It also monitors and evaluates their performance, and provides suitable feedback to encourage further exercise. The application has been developed using C# and WPF, and Vitruvius to detect the user posture; sitting, standing, raising hands, opening arms, etc. Magsino et.al [59] has developed models for lower body exercises used for rehabilitation of post-stroke patients. They have utilized the functionalities of Kinect v2 and Kinect for Windows Software Development Kit (SDK) 2.0 with C# to develop the system. The features such as joint angles of hips and knees, and distances of feet and shoulders have been found using Vitruvius.

A portable Kinect-based system for gait analysis has been presented in [58]. They have developed an application, using C# and WPF with Vitruvius, to interface with a Kinect Sensor in order to detect the skeleton of the user and perform the necessary computations. The feasibility of a game to involve parents in the language development of their hearing impaired child by using sign language has been investigated in [60]. The system has been built using Microsoft Kinect SDK on Vitruvius Kinect framework to recognize letters from the Alphabet in Sign language through hand gesture recognition.

An interactive robotic bear that can mimic the motions of a user in real time using body recognition features of Microsoft Kinect has been developed in [61]. They have recommended the use of Vitruvius for future development of their project to explore and implement more advanced features of the Kinect. Chua et.al [62] have developed a microcontroller based robotic arm capable of mapping the skeletal layout of a person and tracking its movements. For the motion capturing, they have used Kinect sensors programmed on Visual Studio. The development of this program, which has been responsible for getting the angles, the measurements between each joint, and calculating forward and inverse kinematics, has been performed using the Vitruvius framework.

2.6 Summary of the Literature Review

The literature review was done focusing on the two main areas concerned in this research; fall detection and emergency notification. Fall detection was considered from walking or standing position as well as from wheelchairs. A summary of the literature review is given in Table 2.1.

Table 2.1: Summary of Literature Review

Areas concerned	**Existing systems**
Fall Detection	Falls from Standing or Walking Position; -Wearable Devices: -Smartphone [17] -Smartwatch [18] [19] [20] -Waistband [21] -Other Devices [22] [23] [24] -Vision based Devices: -Video Cameras [25] -Kinect Sensor [26] [27] [28] Falls from Wheelchairs; -Wheelchair Modification: -Using Pressure Sensors [54] -Using Gyroscopes [56]
Emergency Notification	Automatic Notification; -Notifying a Predefined List of Contacts [31] [32] [33] [34] [35] [36] [37]

Once the fall detection is considered, mechanisms for fall detection from standing or walking position as well as from wheelchairs were studied. Thereby it was found that, the existing systems for fall detection from standing or walking position can be mainly categorized in to two as wearable devices and vision based devices.

The main problem with wearable devices is that, if the person forgets to wear or carry the device with him/her, then the whole purpose of the entire system will go in vain. Further, most of them are obtrusive and uncomfortable to wear, for instance a waistband, which needs to be specially worn for the expected purpose of detecting a fall. Therefore, researchers have focused on developing vision based devices for fall detection.

When vision based systems are considered, researches have been conducted using both video cameras as well as the Kinect sensor. The video cameras are capable of capturing only the mere physical visual of the human body while the Kinect sensor can seize the skeleton configuration of the human body. Thereby, the Kinect sensor is more accurate in extracting data of the human body and thus, more reliable than the video cameras in developing fall detection systems. Therefore, in this research the Kinect sensor has been utilised to enable vision of a service robot called Moratuwa Intelligent Robot (MIRob) [63], who continuously monitors the elderly or the disabled person.

Further, most of the already proposed vision based methods fail to satisfy required levels of accuracy and robustness. Thus, this research introduces a novel method of detecting fall using the Kinect sensor, which assures to minimise most of the shortages prevailing in the existing vision based systems.

For instance, once the distance change of the centre of mass of the body or even that of the centroid of the body are considered, they are highly dependent upon the body structure of the person. For a considerably shorter person, these centres will lie closer to the ground and thereby the distance change during a fall will be comparatively very lower than that for a taller person.

Therefore, in this research the velocities of three joints; Spine Shoulder, Spine Mid and Spine Base, all lying in the upper half of the body are considered in detecting a fall from standing or walking position. Here, all the three joints under concern show significant changes during falls and thereby makes the system more accurate and reliable.

In addition, this research also presents a novel method of distinguishing falls in to three types; Prone, Crawl and Kneel. This categorisation helps in providing more focused medical attention and more effective treatments as well.

When considering detection of falls from wheelchairs, only a few researches have been done for the development of suitable systems. There also, research has been conducted for the improvement of wheelchairs to accomplish the task of identifying abnormal sitting postures and falling from wheelchairs. Here, these mechanisms are limited only to the particular wheelchair which is specially equipped with the required devices and thereby restrict their applicability and reliability.

This research introduces a novel vision based system to identify abnormal sitting postures in wheelchairs as well as to detect falls from wheelchairs. The proposed system requires no modification in the wheelchair and thereby can be easily implemented in a wider range of applications with higher reliability.

As both the proposed fall detection systems involve the Kinect sensor, they can be easily implemented in real-time. Further, they have the ability to guarantee speedy process and thereby can result in timely detection of falls as well.

When considering the emergency notification systems, a lot of researches have been carried out in order to facilitate quick notification and thereby provide timely assistance.

One of the major drawbacks encountered in most of the existing emergency notification systems is the making of simultaneous calls to a predefined list which causes unnecessary disturbance to everyone on the list. The proposed notification system in this research provides a solution for this drawback, as it notifies a list of persons successively. So that, if the system could make a successful contact, then the rest of the persons on the list will not be disturbed.

Another failure is the lack of attention paid to the response patterns of the contact persons resulting in making many futile attempts. For instance, if a particular person usually do not respond, then we can avoid wasting time on trying to contact him/her. But, this facility is not available in the existing notification systems. The proposed system in this research caters this problem as well, by taking the response patterns of the contact persons in to consideration while making the order of contacting them.

The inadequacy of proper feedback system raising the obvious possibility of the call not getting answered and the elder not getting assisted is also another most common drawbacks found in the existing emergency notification systems. The notification system proposed in this research is capable of obtaining feedback of whether a responsible person gets successfully contacted or not. Thus, it assures assistance to the elderly or the disabled person by continuing the calling process until a successful contact is made.

Further, the proposed Artificial Intelligence based notification system can be triggered manually as well as automatically. This facilitates more reliable and versatile notification during an emergency.

Chapter 3

DETECTION SYSTEM FOR FALLS FROM STANDING OR WALKING POSITIONS

Figure 3.1 and Figure 3.2 give the overview and the model of the system developed to detect falls from standing or walking positions respectively.

It deploys a service robot called MIRob to continuously monitor the elder/ individual by being in his/her surrounding in a non-obtrusive manner. The robot is equipped with a Microsoft Kinect Sensor to enable vision. The RGB and depth data from the Kinect Sensor are extracted by the 'Data Extraction Unit' using Kinect for Windows SDK. The x,y and z position data of joints Spine Shoulder, Spine Mid, Spine Base and Knee are recorded by the 'Data Recorder' every 0.5 seconds for analysis.

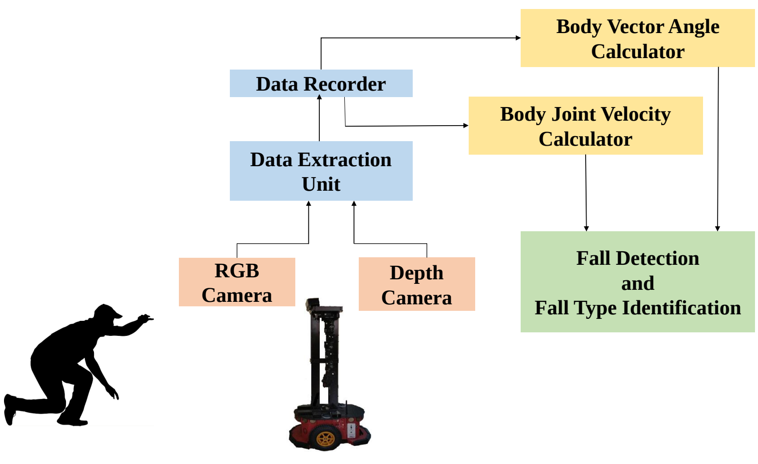

Figure 3.1: Overview of Detection System for Falls from Standing or Walking Positions

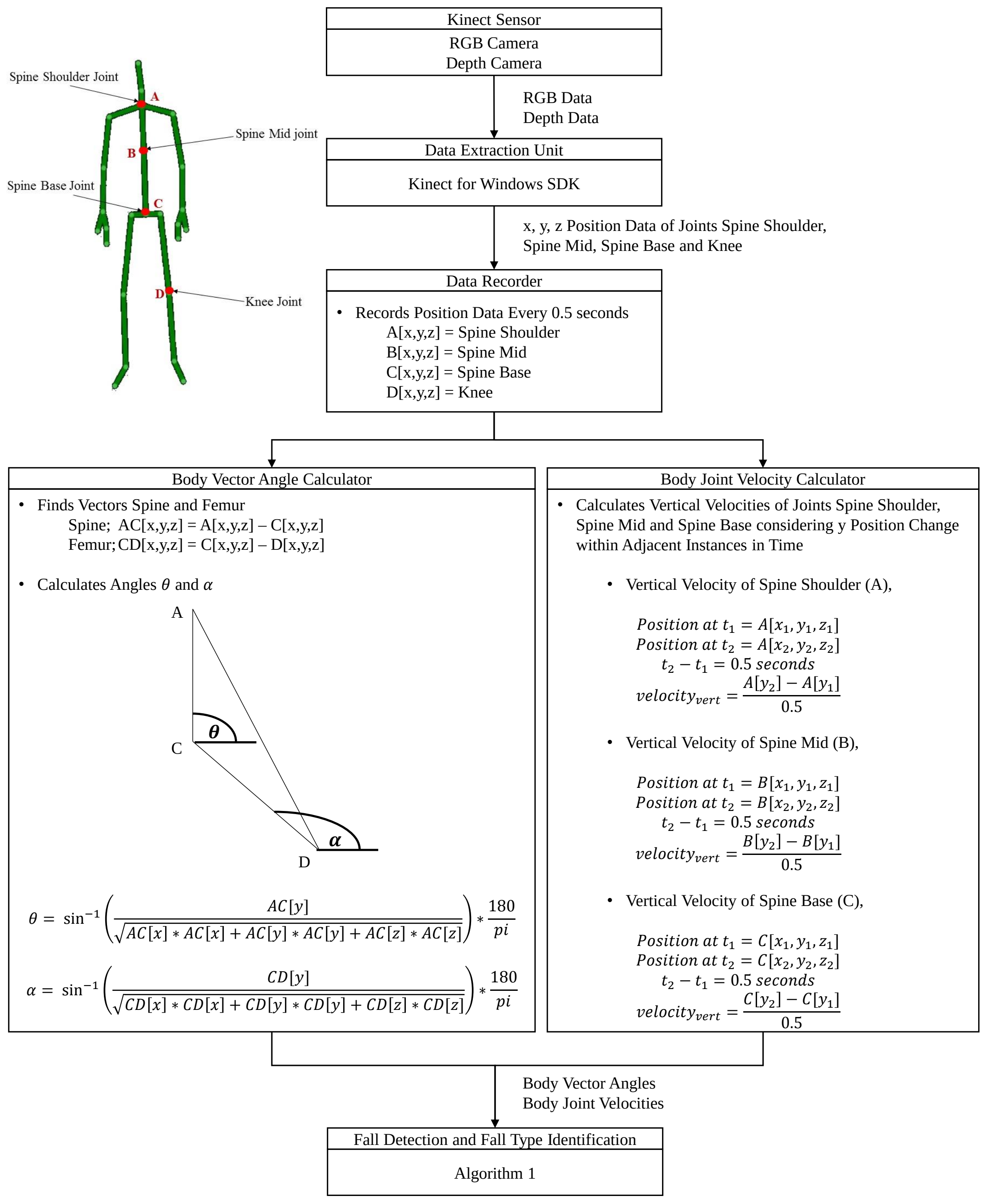

Figure 3.2: Model of Detection System for Falls from Standing or Walking Positions

The fall detection and fall type identification take place by considering the orientation of two limbs; Spine and Femur, and velocities of three joints; Spine Shoulder, Spine Mid and Spine Base. The calculation of body joint velocities and body vector angles happen in the 'Body Joint Velocity Calculator' and the 'Body Vector Angle Calculator' respectively. The method of calculating them are discussed in detail in the Section 3.3.

Based on the findings of these two units, the unit called 'Fall Detection and Fall Type Identification' follows Algorithm 1, which is discussed in Section 3.4. And hence, outputs whether a fall is detected and if detected, the type of that fall is also recognized.

The system was developed with a Windows PC using Kinect for Windows SDK and Visual Studio 2017. The programming was done with C++ language.

3.1 Microsoft Kinect Sensor

The Kinect sensor has been launched by Microsoft in 2010 as a motion-sensing input device for the Xbox360 gaming console. It has become extensively popular within the field of robotics because of the advanced capabilities it offers for the human-robot interaction through high performance in 3D image capture, facial recognition and voice recognition.

As shown in Figure 3.3, the Microsoft Kinect Sensor has 3 vital hardware components working together to generate 3D images of objects and recognize human beings within its field of vision; an RGB color VGA video camera, a depth sensor, and a multi-array microphone.

The RGB camera detects the red, green, and blue color components and carries out the body and facial recognition. It has a pixel resolution of 1280 x 960 and a frame rate of 30 fps (frames per second).

Figure 3.3: Microsoft Kinect Sensor

The depth sensor has a monochrome CMOS sensor and an infrared projector that create 3D imagery throughout the surrounding environment. It also transmits invisible near-infrared light and finds the distance to points of objects by measuring its "time of flight" after getting reflected off the objects. The use of an infrared generator helps in solving the problem of ambient light as it does not register visible light. This has a 640 x 480-pixel resolution and run at 30 fps.

The Kinect microphone is an array of 4 microphones providing a wide-field, conic audio capture. It is capable of isolating voices from other background noises.

The Kinect can distinguish objects' depth within 1 centimeter and height and width within 3 millimeters. For the sensor to work properly, it is recommended to allow about 6 feet (1.8 meters) of space between the target and the sensor. The normal depth vision of the Kinect ranges from around two and a half feet (800mm) to just over 13 feet (4000mm). However, the recommended usage range is 3 feet to 12 feet as the reliability of the depth values degrade at the edges of the field of view. It has view angles of 57 degrees horizontal and 43 degrees vertical.

To work with its hardware, the Kinect developers have built an excellent software as well. They have collected motion-capture data in real-life scenarios and processed them using an artificial intelligence machine-learning algorithm so that the sensor can map the visual data it collects to models representing people who are different in attributes such as age, height, gender, body type, and clothing etc.

3.2 Kinect for Windows SDK (Software Development Kit)

The Kinect for Windows SDK is a set of libraries that allows programming of applications on a variety of Microsoft development platforms using the Kinect sensor as input. It runs only on Windows operating systems and installs several reference applications as well as samples which are written in a combination of C# and C++. These applications provide a starting point for working with the SDK. They show how to use the Kinect SDK and present the best practices for programming with the SDK.

The most distinguishing feature of Kinect is its ability to visually track human skeletal and joint structures. The Kinect SDK processes the IR data received from the infrared camera and produces depth images. In skeleton tracking, the depth image data is processed to establish the positions of various skeleton joints on a human form. Each joint has an identity such as head, shoulder, elbow, etc. and a 3D vector determining its position. The 3D vector is provided by skeleton tracking, where each skeleton joint point gets X, Y, and Z values indicating its position.

The skeleton space is measured in meters with the zero X and Y positions at the center of the depth sensor. The skeleton space uses a right-handed coordinate system where the positive values of the X-axis extend to the right and the positive Y-axis extends upward. The X-axis ranges from -2.2 to 2.2; the Y-axis ranges from -1.6 to 1.6; and the Z-axis from 0 to 4.

Figure 3.4 depicts the skeleton space obtained from the Kinect SDK. The Kinect for Windows SDK can capture up to 2 individuals simultaneously and track 20 joints as shown in Figure 3.5.

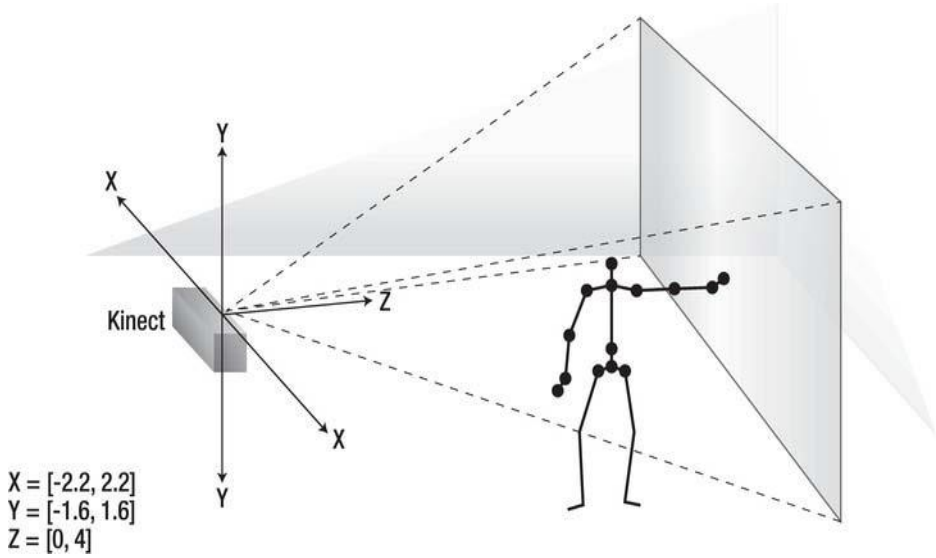

Figure 3.4: Skeleton Space from Kinect SDK [64]

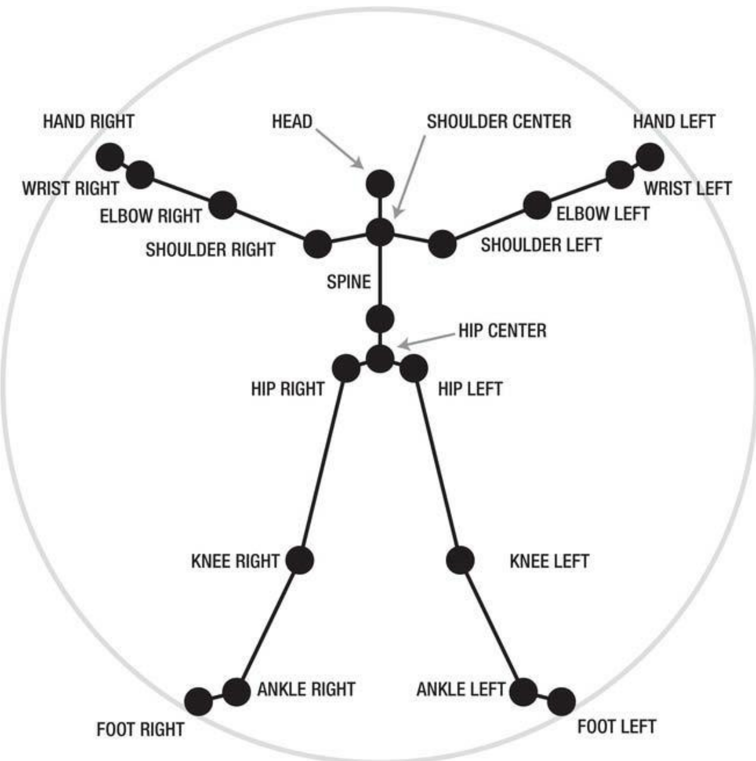

Figure 3.5: Skeleton Joint Points from Kinect SDK [64]

3.3 Calculation of Body Joint Velocities and Body Vector Orientations

The fall detection and fall type identification happen by calculating body joint velocities and body vector angles. For this, the system first monitors and stores x, y and z position data of four joints; the Spine Shoulder joint, Spine Mid joint, Spine Base joint and the Knee joint as shown in the Figure 3.6.

As discussed in Section 3.2, from Kinect for Windows SDK we can obtain the x, y and z positions of the joints. Thus, the developed system records the position details of the selected joints every 0.5 seconds intervals.

When calculating the body joint velocities, the system considers the vertical velocities of three joints; Spine Shoulder, Spine Mid and Spine Base. All these three joints lie in the upper part of the body. These were chosen as they show comparatively higher variations in velocities during a fall from a standing or walking position. Although knee and ankle joints are also subjected to a much higher movement during a fall, their velocities are not considered, as they move fast in a number of daily activities as well.

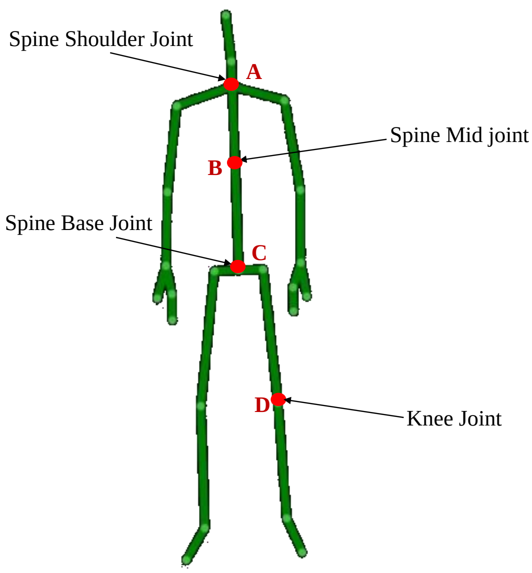

Figure 3.6: Body Joints used to calculate Body Joint Velocities

The vertical velocities of the three considered joints are calculated using the change in their y position within adjacent instances in time as shown in Equations (3.1) to (3.3).

Vertical Velocity of Spine Shoulder(A);

$$\begin{aligned} Position\ at\ t_1 &= A[x_1, y_1, z_1] \\ Position\ at\ t_2 &= A[x_2, y_2, z_2] \\ t_2 - t_1 &= 0.5\ seconds \\ velocity_{vert} &= \frac{A[y_2] - A[y_1]}{0.5} \end{aligned} \tag{3.1}$$

Vertical Velocity of Spine Mid(B);

$$\begin{aligned} Position\ at\ t_1 &= B[x_1, y_1, z_1] \\ Position\ at\ t_2 &= B[x_2, y_2, z_2] \\ t_2 - t_1 &= 0.5\ seconds \\ velocity_{vert} &= \frac{B[y_2] - B[y_1]}{0.5} \end{aligned} \tag{3.2}$$

Vertical Velocity of Spine Base(C);

$$\begin{aligned} Position\ at\ t_1 &= C[x_1, y_1, z_1] \\ Position\ at\ t_2 &= C[x_2, y_2, z_2] \\ t_2 - t_1 &= 0.5\ seconds \\ velocity_{vert} &= \frac{C[y_2] - C[y_1]}{0.5} \end{aligned} \tag{3.3}$$

For the calculation of body vector orientations, the system considers two body vectors; Spine vector and Femur vector, by monitoring and recording position data of three body joints; Spine Shoulder, Spine Base and Knee.

As humans often make random movements with their Tibia, the system does not involve the vector Tibia.

Spine vector is obtained by connecting the joints Spine Shoulder and Spine Base, and Femur vector is obtained by connecting joints Spine Base and Knee Right as shown in the Figure 3.7.

The orientation of these vectors are found by considering the angles they make with the horizontal. Thus the angle made by the Spine vector with the horizontal is named as theta (θ) and the angle made by the Femur vector with the horizontal is named as alpha (α). The Femur vector is found using the right limb of the skeleton. Here, it must be noted that the vectors are used in order to ease the calculation of considered angles.

Figure 3.8 shows the body triangle made by the three joints Spine Shoulder(A), Spine Base(C) and Knee Right(D). Thus, the Spine Vector and the Femur Vector are found by Equations (3.4) and (3.5) respectively.

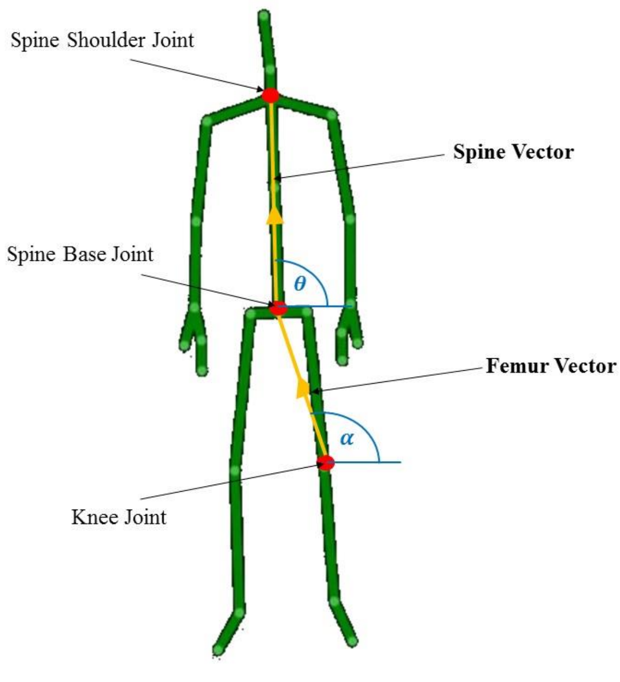

Figure 3.7: Body Vectors used to calculate Body Vector Orientations

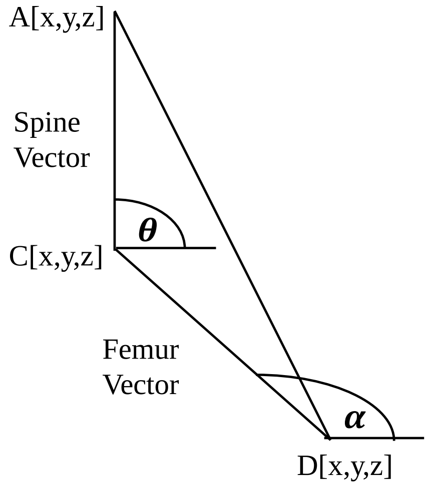

Figure 3.8: Body Triangle made by the Three Joints; Spine Shoulder(A), Spine Base(C) and Knee Right(D)

Spine Vector(AC);

$$AC[x,y,z] = A[x,y,z] - C[x,y,z] \tag{3.4}$$

Femur Vector(CD);

$$CD[x,y,z] = C[x,y,z] - D[x,y,z] \tag{3.5}$$

Then, the angles theta (θ) and alpha (α) are found by the Equations (3.6) and (3.6) respectively.

$$\theta = sin^{-1}\left(\frac{AC[y]}{\sqrt{AC[x]*AC[x]+AC[y]*AC[y]+AC[z]*AC[z]}}\right)*\frac{180}{pi} \tag{3.6}$$

$$\alpha = sin^{-1}\left(\frac{CD[y]}{\sqrt{CD[x]*CD[x]+CD[y]*CD[y]+CD[z]*CD[z]}}\right)*\frac{180}{pi} \tag{3.7}$$

3.4 Fall Detection and Fall Type Identification System

The fall detection and fall type identification system makes use of body joint velocities and body vector orientations which are calculated as discussed in Section 3.3. Falls are distinguished from activities of daily living (ADLs) by considering the vertical velocities of the body joints, as falling always causes higher velocities than ADLs.

The proposed fall detection and fall type identification system is developed based on experimental data. The experiment was conducted in a simulated domestic environment inside the laboratory with the participation of 27 volunteers, who were asked to perform falls in different ways without specifically telling how to fall.

The data obtained were analyzed and were categorized into 3 types of falls namely; Prone Position, Crawl Position, and Kneel Position as depicted in the Figure 3.9.

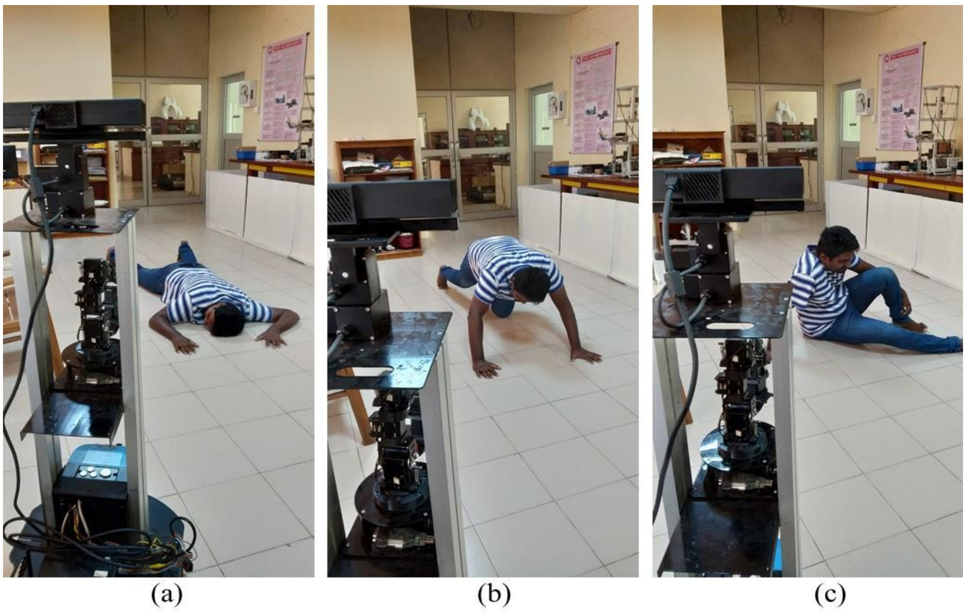

Figure 3.9: Types of Falls (a)Prone Position (b)Crawl Position (c)Kneel Position

Prone Position is where the subject's whole body is on the floor and he/she lies on the chest with face down. Crawl Position is where the subject's knees are on the floor and he/she tries to avoid the body from hitting the ground by applying hand cushioning. Kneel Position is where the subject is seated on the floor with his/her torso approximately upright.

When observed closely, during a fall, there is a considerable change in velocity and this is shown in Figure 3.10. These graphs are drawn considering a single person falling in the above mentioned 3 types of falls.

Equations (3.8), (3.9) and (3.10) give the range of transition velocities during the 3 types of falls; Type 1: Prone Position,Type 2: Crawl Position,Type 3: Kneel Position. Upper limits of the vertical velocities of the 3 types of falls are denoted by lim_{high_type1}, lim_{high_type2} and lim_{high_type3} respectively. Similarly, lower limits of the 3 types of falls are denoted by lim_{low_type1}, lim_{low_type2} and lim_{low_type3} respectively. The actual values of these limits are determined empirically by simulating the falls multiple times.

$$lim_{high\_type1} \geq velocity_{vert} \geq lim_{low\_type1} \tag{3.8}$$

$$lim_{high\_type2} \geq velocity_{vert} \geq lim_{low\_type2} \tag{3.9}$$

$$lim_{high\_type3} \geq velocity_{vert} \geq lim_{low\_type3} \tag{3.10}$$

To confirm a certain movement as a fall, the angles made by 2 selected body vectors with the horizontal are also analyzed. For a single type of fall, a range of values could be observed for these angles, depending on the individual under consideration. The variation of these angles during the 3 types of falls are shown in Figure 3.11.

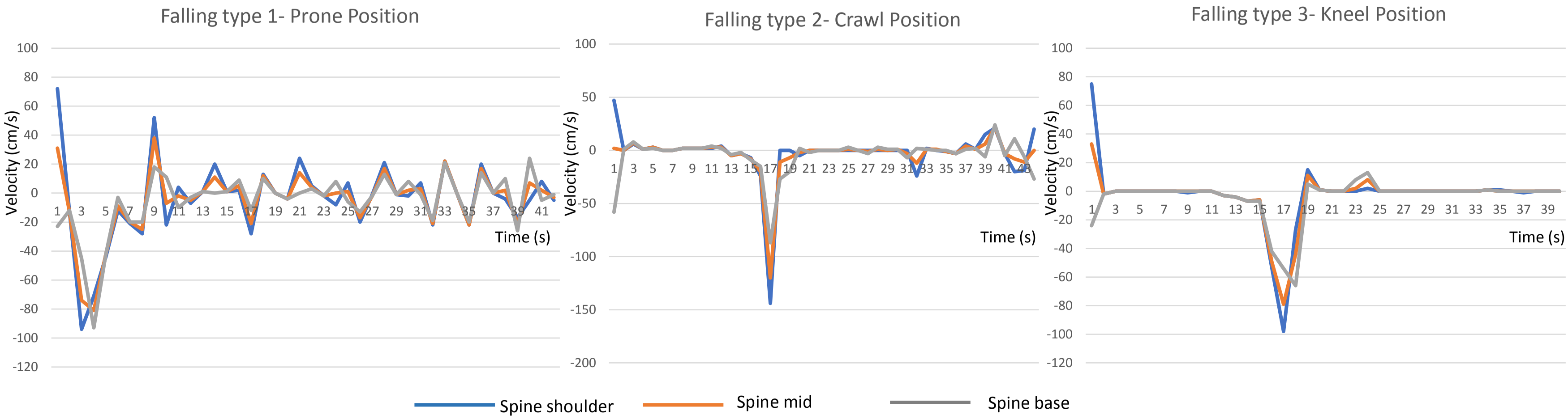

Figure 3.10: Variation of Vertical Velocities of Spine Shoulder, Spine Mid and Spine Base Joints for the 3 Types of Falls against Time

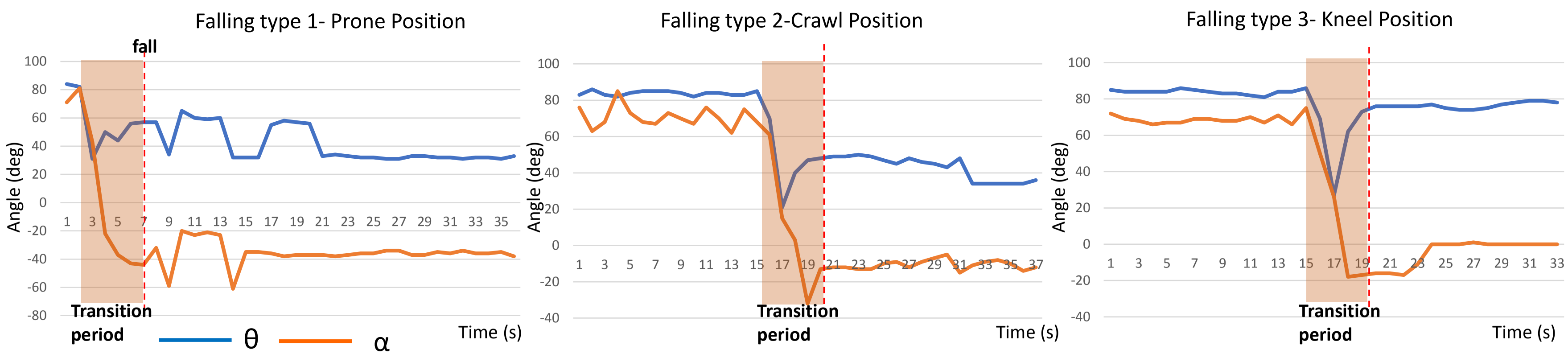

Figure 3.11: Variation of Angles θ and α for the 3 Types of Falls against Time

Equations (3.11) to (3.16) give the angle limits observed for each type of fall. The upper limits of the angles made by spine and femur body vectors with the horizontal for the 3 types of falls are denoted by θ_{high_type1}, α_{high_type1}, θ_{high_type2}, α_{high_type2}, θ_{high_type3}, and α_{high_type3} respectively. Similarly, the lower limits of the angles made by spine and femur body vectors with the horizontal for the 3 types of falls are denoted by θ_{low_type1}, α_{low_type1}, θ_{low_type2}, α_{low_type2}, θ_{low_type3} and α_{low_type3} respectively. The actual values for these variables were found empirically by simulating falls.

$$\theta_{high\_type1} \geq \theta_{type1} \geq \theta_{low\_type1} \tag{3.11}$$

$$\alpha_{high\_type1} \geq \alpha_{type1} \geq \alpha_{low\_type1} \tag{3.12}$$

$$\theta_{high\_type2} \geq \theta_{type2} \geq \theta_{low\_type2} \tag{3.13}$$

$$\alpha_{high\_type2} \geq \alpha_{type2} \geq \alpha_{low\_type2} \tag{3.14}$$

$$\theta_{high\_type3} \geq \theta_{type3} \geq \theta_{low\_type3} \tag{3.15}$$

$$\alpha_{high\_type3} \geq \alpha_{type3} \geq \alpha_{low\_type3} \tag{3.16}$$

The 'Transition Period' denoted in Figure 3.11, corresponds to the time duration in which the velocity changes during a fall. This period differs according to the type of fall and this feature is used to identify a fall and distinguish it in between the 3 types of falls.

To confirm a particular scenario as a fall, the system will analyze it over a timeline. This is due to the fact that sometimes a person may bend down to pick an object or scratch the leg. In such situations, the postures and velocities may be similar to that of a fall, depending on the individual characteristics. However, such occasions prevail only for a very short duration of time. Therefore, the system observes a scenario that is suspected to be a fall, for a few more seconds as soon as a 'Transition Period' is detected.

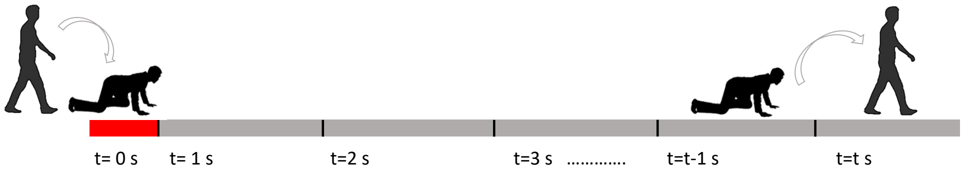

Figure 3.12: Behavior of an Individual during a Fall. The transition from initial posture to a fall (the transition period) is observed at t=0 s. This fall exists for a period of t seconds before the person gets up back to a standing position.

If the conditions continue to exist for a period of 't' seconds as shown in Figure 3.12, the considered scenario is confirmed to be a fall. The actual value of 't' is determined after analyzing a set of simulated falls and the time taken by an individual to get up back to a standing position.

Algorithm 1 shows how a fall is detected and differentiated among the 3 types of falls using the proposed approach. The logical flow diagram of Algorithm 1 is given in Figure 3.13. Firstly, it is required to find the body joint velocities and body vector angles. Then, it must be identified whether a transition period is observed. If so, the system must wait for t seconds to confirm the occurrence of a fall. After that, by comparing the body joint velocities and the body vector angles, the fall type is identified.

Algorithm 1 Identify Fall, Type of Fall

Require: Body Vector Angles, Body Joint Velocities
Ensure: Type of Fall
  State: Transition period observed
    if fall is present throughout t seconds **then**
      Select the corresponding type of fall
      if $\theta_{high\_type1} \geq \theta \geq \theta_{low\_type1}$ AND $\alpha_{high\_type1} \geq \alpha \geq \alpha_{low\_type1}$ AND $lim_{high\_type1} \geq velocity_{vert} \geq lim_{low\_type1}$ **then**
        Type 1 fall
      else
        if $\theta_{high\_type2} \geq \theta \geq \theta_{low\_type2}$ AND $\alpha_{high\_type2} \geq \alpha \geq \alpha_{low\_type2}$ AND $lim_{high\_type2} \geq velocity_{vert} \geq lim_{low\_type2}$ **then**
          Type 2 fall
        else
          if $\theta_{high\_type3} \geq \theta \geq \theta_{low\_type3}$ AND $\alpha_{high\_type3} \geq \alpha \geq \alpha_{low\_type3}$ AND $lim_{high\_type3} \geq velocity_{vert} \geq lim_{low\_type3}$ **then**
            Type 3 fall
          else
            Other fall
          end if
        end if
      end if
    end if

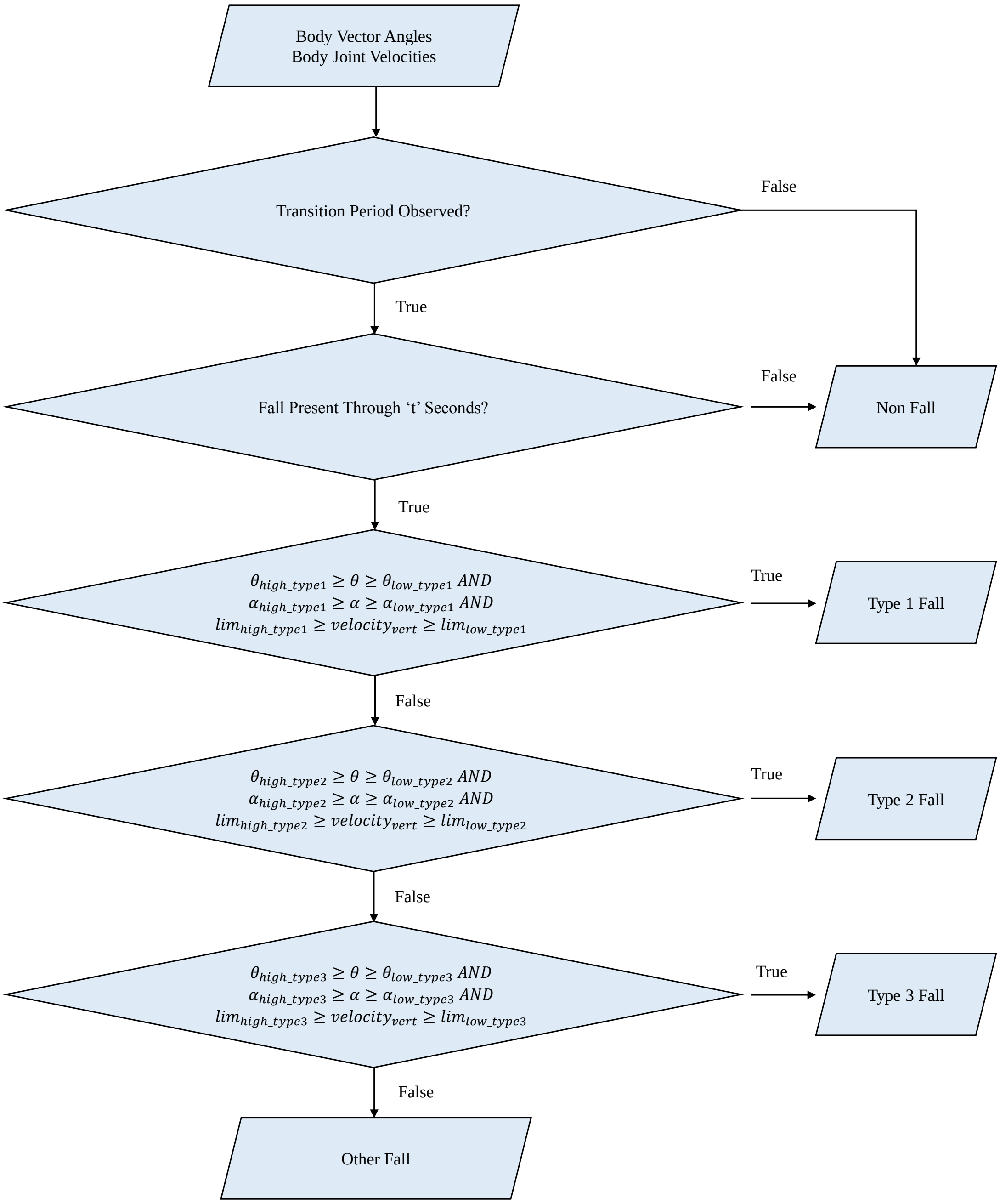

Figure 3.13: Logical Flow Diagram of Algorithm 1

3.5 System Testing and Results

The 3 types of falls were performed in a simulated domestic environment inside the laboratory with the participation of 27 individuals in the age range of 25-58 (Mean-27.6, SD-11.4). Then each fall was categorized into the respective category by the algorithm. The participants were unaware of the way of falling.

It must be noted that, the approach does not consider the dimensions of human body as a parameter to determine falls. The reason behind this approach is that, the body dimensions are unique to each individual and therefore establishing a general criteria upon dimensional changes during fall is unreliable and extremely difficult.

After several empirical studies upon various activities of daily living and incidents of falling, the value of t was determined to be 4 seconds. The limits of lim_{low_type1}, lim_{low_type2} and lim_{low_type3} in Equations (3.8), (3.9) and (3.10) were found to be 80, 85, and 65 cm/s respectively. The actual values for lim_{high_type1}, lim_{high_type2} and lim_{high_type3} determined through the experiments were 100, 150 and 130 cm/s respectively. The Table 3.1 shows these upper and lower limits of velocities.

Table 3.1: Lower and Upper Limits of Velocities in Equations (3.8) to (3.10)

Fall Type	Lower Limit (cm/s)	Upper Limit (cm/s)
Prone Position	80	100
Crawl Position	85	150
Kneel Position	65	130

The lower and upper limits of θ and α in Equations (3.11) to (3.16) were found to be -5, 35, -40, 5, -5, 35, -40, -65, 70, 90, -25 and 0 degrees for the 3 types of falls. The Tables 3.2 and 3.3 shows these lower and upper limits of *theta* and *alpha* respectively. The transition period was recognized by the difference of body joint velocities for a consecutive 2 seconds before the counting of t got started.

Table 3.2: Lower and Upper Limits of Angle *theta*

Fall Type	Lower Limit (degrees)	Upper Limit (degrees)
Prone Position	-5	35
Crawl Position	-5	35
Kneel Position	70	90

Table 3.3: Lower and Upper Limits of Angle *alpha*

Fall Type	Lower Limit (degrees)	Upper Limit (degrees)
Prone Position	-40	5
Crawl Position	-40	-65
Kneel Position	-25	0

Table 3.4 shows the confusion matrix for falls and non-falls for a sample 40 scenarios with the participation of 27 individuals. The table gives the overall scenarios including falls and non-falls and, successful and unsuccessful attempts made by the proposed approach.

Table 3.4: Results obtained for the Fall Detection Algorithm

	Predicted scenarios	
Total scenarios	**Falls**	**Non-falls**
Falls	True Positives (TP) - 21	False Negatives (FN) - 1
Non-falls	False Positives (FP) - 2	True Negatives (TN) -16

Out of the 22 falls recorded, 21 were identified and the recognition accuracy of each type of fall is as follows.

- Type 1 fall (Prone position)- 10 out of 10 (100%)
- Type 2 fall (Crawl Position)- 4 out of 5 (80%)
- Type 3 fall (Kneel Position)- 6 out of 7 (86%)

There was another occasion where type 2 was identified as type 1 as a result of self-occlusion caused during the fall. Recognition of crawl position was difficult due to this reason. Hence type 2 falls were recognized with a relatively low accuracy. Due to the easily-visible body arrangement in prone position, type 1 falls were recognized with the highest accuracy. There was one occasion where type 3 fall was recognized a type 2 fall. In that occasion, the individual transformed into kneel position from crawl during the fall and the system had already recognized the fall as type 2 before the individual settled in kneel position. This behavior could be observed among adults as they try to balance themselves to reduce damage by fast movements before trying to stand on foot after the fall.

The accuracy of the fall detection system was calculated by considering the percentage ration between the sum of TP and TN and the entire sample (Sum of TP, FP, TN and FN). Hence, we received an accuracy of 92.5%.

Similarly, the sensitivity of the system was calculated by considering the percentage ratio between the recognized falls (TP) and the number of actual falls (sum of TP and FN). Hence we received a sensitivity of 95.45%. The reason for this is the inability of the system to perceive scenarios with multiple jerks during a fall. For example if a person hits something, falls down while trying to avoid that incorporates a series of movements. In such scenarios, recognizing a transition period becomes difficult.

The specificity of the system was calculated by taking the percentage ratio between the recognized non-falls (TN) and the number of all non-falls (sum of FP and TN) and this was 88%. The reason for having 2 FPs was that the system was incapable of recognizing movements in activities such as an exercise involving sudden up and down movements.

From the observations we could observe that for fall detection to be even more accurate, human activities have to be distinguished. Even though we receive a satisfactory accuracy from the proposed approach, still we lack the capability of identifying complex scenarios such as activities incorporating sudden movements. In type 1 fall, the person lies on the stomach. This same body arrangement could be observed when a person lies on his/her back. Even rare, such falls should also be distinguished using the proposed approach.

CHAPTER 4

SYSTEM FOR POSTURE IDENTIFICATION AND FALL DETECTION OF WHEELCHAIR USERS

Figure 4.1 and Figure 4.2 respectively depict the overview and the model of the system developed for posture identification and fall detection of wheelchair users.

It involves a service robot called MIRob that continuously monitors the wheelchair user by being in his/her surrounding in a non-obtrusive manner. The robot is equipped with a Microsoft Kinect Sensor that enables vision. The RGB and depth data from the Kinect Sensor are extracted by the 'Data Extraction Unit' using LightBuzz Vitruvius Framework. The x,y and z position data of joints Spine Shoulder, Shoulder Left, Shoulder Right and Spine Base along with the Floor Plane are recorded by the 'Data Recorder' every 0.5 seconds for analysis.

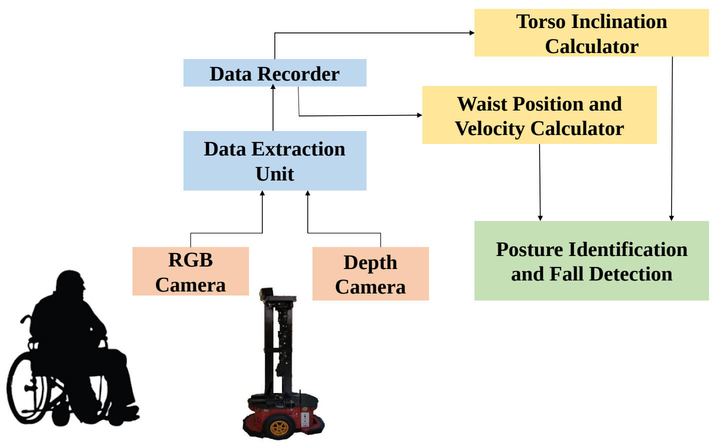

Figure 4.1: System for Posture Identification and Fall Detection of Wheelchair Users

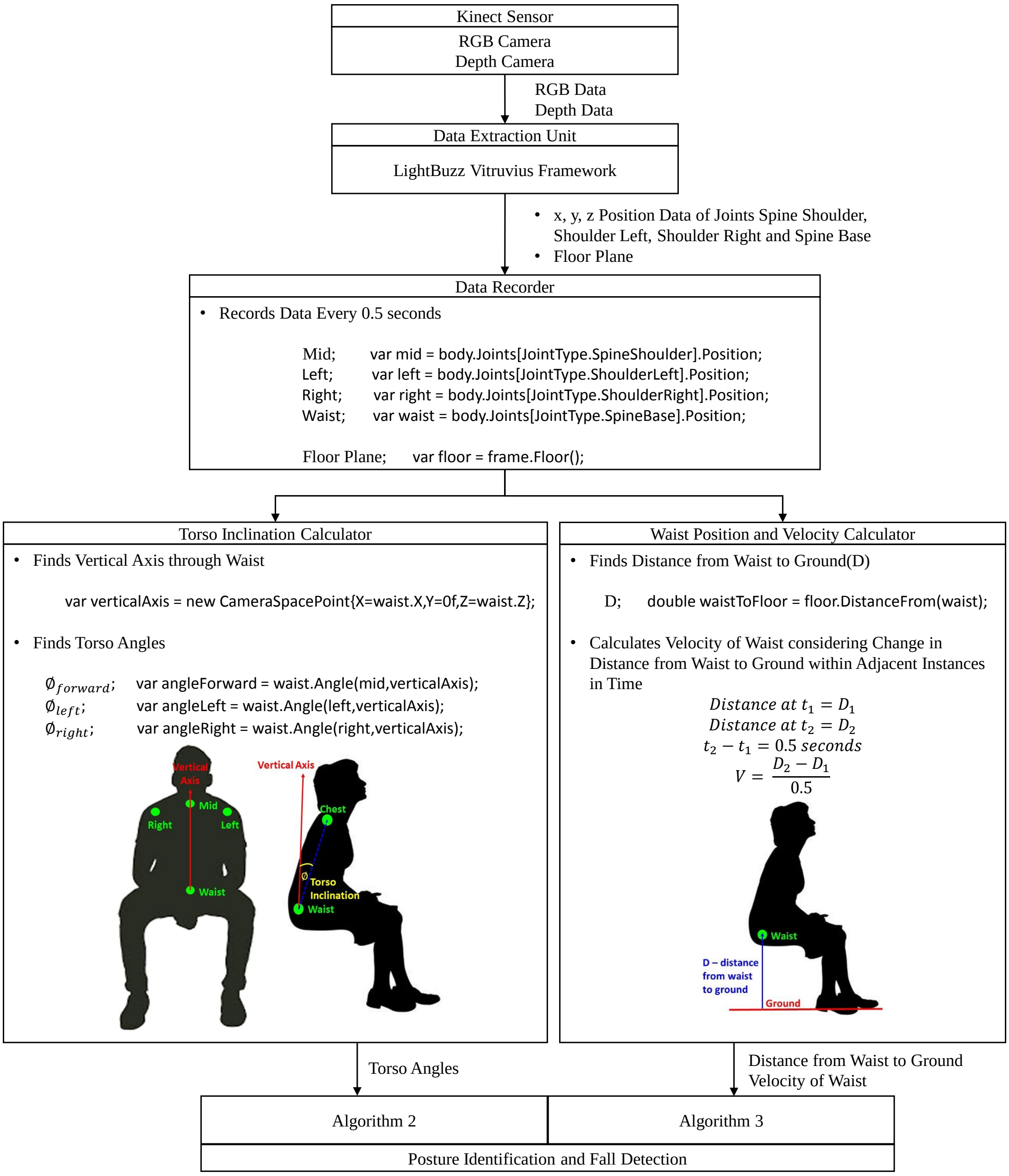

Figure 4.2: Model of System for Posture Identification and Fall Detection of Wheelchair Users

The posture identification occurs by calculating the torso inclination of the wheelchair user in 3 directions inside the 'Torso Inclination Calculator'. The method of calculating the torso angles is described in Section 4.1.1. The fall detection takes place by calculating and analyzing the waist position and the velocity inside the 'Waist Position and Velocity Calculator'. The process of this calculator is explained in Section 4.2.1.

By analyzing the results from these two units, the 'Posture Identification and Fall Detection' unit outputs whether the wheelchair user is seated in an abnormal posture by following Algorithm 2 and whether the wheelchair user is encountering a fall from the wheelchair by following Algorithm 3. The Algorithm 2 and Algorithm 3 are explained in Section 4.1.2 and Section 4.2.1 respectively.

The system was developed using a Windows PC with LightBuzz Vitruvius Academic Version and Visual Studio 2017. The programming was done with C# language.

4.1 Posture Identification of Wheelchair Users

The main objective of wheelchair user posture identification is to detect any abnormal sitting postures that might result in wheelchair overturns. The system has the ability to distinguish among 4 different sitting postures as depicted in the Figure 4.3 namely; proper-sitting, lean-forward, lean-left, and lean-right.

The distinguishing is done by calculating and analyzing the angles made by the torso of the wheelchair user with the vertical axis in 3 directions; forward, left and right. As the system calculates these angles with respect to the vertical axis, the results do not depend on the camera angle.

Figure 4.3: Four Sitting Postures Identified by the System.(a)Proper-Sitting (b)Lean-Forward (c)Lean-Left (d)Lean-Right

4.1.1 Calculation of Torso Inclination

Under Section 3.2, it was mentioned that the Kinect for Windows SDK can track 20 human body joints simultaneously. A joint structure includes three components; the position of the joint in 3D space, the name of the joint and the tracking accuracy. The structure called CameraSpacePoint is used to express the 3D position of a joint in camera space.

The Camera space refers to the 3D coordinate system used by the Kinect sensor, where the origin (x=0, y=0, z=0) of the coordinate system is located at the center of the IR sensor on the Kinect Camera, and each unit corresponds to one meter. Here, the x grows to the left of the sensor, y grows upwards from the sensor depending on its tilt, and z grows out in the direction in which the sensor is facing [65].

The LightBuzz Vitruvius Framework contains special extension methods that supports the measurement of angles using CameraSpacePoint [66]. The extension method supplied by the LightBuzz Vitruvius Framework for angle calculation is the Angle method. To utilize this extension method in measuring angles, the only requirement is to know the position of three points; a starting point, a middle point and an ending point.

For example, if the Vitruvius Framework is used to measure the angle made by the elbow, then it requires only the position data of three points; the shoulder, elbow and the wrist. Then the elbow angle can be easily calculated by using the Statement (4.1).

$$double\ angle = elbow.Angle(shoulder, wrist); \tag{4.1}$$

Here, the joint ShoulderLeft is taken as the starting point ‘shoulder’, the joint ElbowLeft is taken as the middle point ‘elbow’ and the joint WristLeft is taken as the ending point ‘wrist’. In this manner, any human body joint angle can be easily measured by using the Vitruvius Framework.

In the proposed system, for the calculation of the torso inclination of the wheelchair user, a similar approach as discussed above is used. The inclination of the torso is considered in 3 directions by measuring the angles made by 3 joints; Spine Shoulder, Shoulder Left and Shoulder Right, with the vertical axis that goes through the Spine Base as shown in Figure 4.4(a).

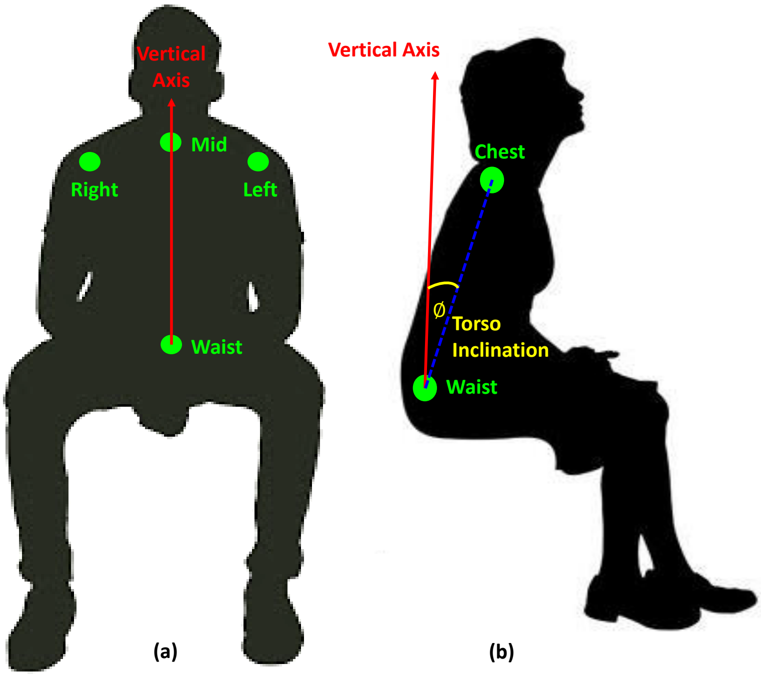

Figure 4.4: (a)Joints considered in measuring the Torso Inclination in Three Directions (b)How Torso Inclination is measured

Here, the joint Spine Base is taken as the point 'Waist', the joint Spine Shoulder is taken as the point 'Mid', the joint Shoulder Left is taken as the point 'Left', the joint Shoulder Right is taken as the point 'Right' and the vertical axis going through the 'Waist' is taken as the axis 'Vertical Axis'. In order to determine the 4 postures accurately, the proposed system considers the 3 directions; forward, left and right by measuring the angles made by these joints.

The x,y and z position data of the four considered joints are recorded every 0.5 seconds using the Statements (4.2) to (4.5).

$$var\ mid = body.Joints[JointType.SpineShoulder].Position; \tag{4.2}$$

$$var\ left = body.Joints[JointType.ShoulderLeft].Position; \tag{4.3}$$

$$var\ right = body.Joints[JointType.ShoulderRight].Position; \tag{4.4}$$

$$var\ waist = body.Joints[JointType.SpineBase].Position; \tag{4.5}$$

Further, Figure 4.4(b) depicts clearly how the angles are being measured with respect to the Vertical Axis from the point 'Waist', where all 3 joints; Mid, Left and Right, are referred together as the 'Chest'.

The Statement (4.6) shows how the Vertical Axis through 'Waist' is obtained.

$$var\ verticalAxis = new\ CameraSpacePoint\{X = waist.X, Y = 0f, Z = waist.Z\}; \tag{4.6}$$

The Statements (4.7) to (4.9) indicate how the torso angles are calculated.

$$var\ angleForward = waist.Angle(mid, verticalAxis); \tag{4.7}$$

$$var\ angleLeft = waist.Angle(left, verticalAxis); \tag{4.8}$$

$$var\ angleRight = waist.Angle(right, verticalAxis); \tag{4.9}$$

The forward inclination of the torso is found using the 'angleForward', which is the angle made by the point 'mid' with the 'verticalAxis'. The left and right inclinations are obtained by considering the 'angleLeft' and 'angleRight' which are the angles made by the points 'left' and 'right' with the 'verticalAxis' respectively.

4.1.2 Determination of Postures

The 4 postures shown in Figure 4.3, are identified by comparing the torso angles measured using the Statements (4.7) to (4.9) against their lower and upper limits as given in Equations (4.10) to (4.12).

$$\phi_{high\_forward} \geq \phi_{forward} \geq \phi_{low\_forward} \tag{4.10}$$

$$\phi_{high\_left} \geq \phi_{left} \geq \phi_{low\_left} \tag{4.11}$$

$$\phi_{high\_right} \geq \phi_{right} \geq \phi_{low\_right} \tag{4.12}$$

The angles 'angleForward', 'angleLeft' and 'angleRight' in Statements (4.7) to (4.9) are represented by $\phi_{forward}$, ϕ_{left} and ϕ_{right} in the Equations (4.10) to (4.12). The upper limits and lower limits of angles $\phi_{forward}$, ϕ_{left} and ϕ_{right} are represented by $\phi_{high\_forward}$, ϕ_{high_left}, ϕ_{high_right} and $\phi_{low\_forward}$, ϕ_{low_left}, ϕ_{low_right} respectively.

Algorithm 2 shows how the proposed system differentiates among the 4 sitting postures. The logical flow diagram of Algorithm 2 is given in Figure 4.5. After the system finds the torso inclination by measuring the torso angles in the 3 directions, the postures are identified by comparing $\phi_{forward}$, ϕ_{left} and ϕ_{right} with their respective lower and upper limits.

Here, the postures 'Proper-Sitting', 'Lean-Forward', 'Lean-Left' and 'Lean-Right' are numbered as 1, 2, 3 and 4 respectively. Therefore, a posture is identified as 'Proper-Sitting' if, $\phi_{forward}$ is within $\phi_{high\_1,forward}$ and $\phi_{low\_1,forward}$, ϕ_{left} is within $\phi_{high\_1,left}$ and $\phi_{low\_1,left}$ and ϕ_{right} is within $\phi_{high\_1,right}$ and $\phi_{low\_1,right}$. Similarly, the other 3 postures could also be identified by considering the corresponding ranges.

Algorithm 2 Posture Identification

Require: Torso Angles

Ensure: Sitting Posture

```
if φ_{high_1,forward} ≥ φ_{forward} ≥ φ_{low_1,forward} AND φ_{high_1,left} ≥ φ_{left} ≥ φ_{low_1,left} AND φ_{high_1,right} ≥ φ_{right} ≥ φ_{low_1,right} then
  Proper-Sitting
else
  if φ_{high_2,forward} ≥ φ_{forward} ≥ φ_{low_2,forward} AND φ_{high_2,left} ≥ φ_{left} ≥ φ_{low_2,left} AND φ_{high_2,right} ≥ φ_{right} ≥ φ_{low_2,right} then
    Lean-Forward
  else
    if φ_{high_3,forward} ≥ φ_{forward} ≥ φ_{low_3,forward} AND φ_{high_3,left} ≥ φ_{left} ≥ φ_{low_3,left} AND φ_{high_3,right} ≥ φ_{right} ≥ φ_{low_3,right} then
      Lean-Left
    else
      if φ_{high_4,forward} ≥ φ_{forward} ≥ φ_{low_4,forward} AND φ_{high_4,left} ≥ φ_{left} ≥ φ_{low_4,left} AND φ_{high_4,right} ≥ φ_{right} ≥ φ_{low_4,right} then
        Lean-Right
      else
        Person out from Wheelchair
      end if
    end if
  end if
end if
```

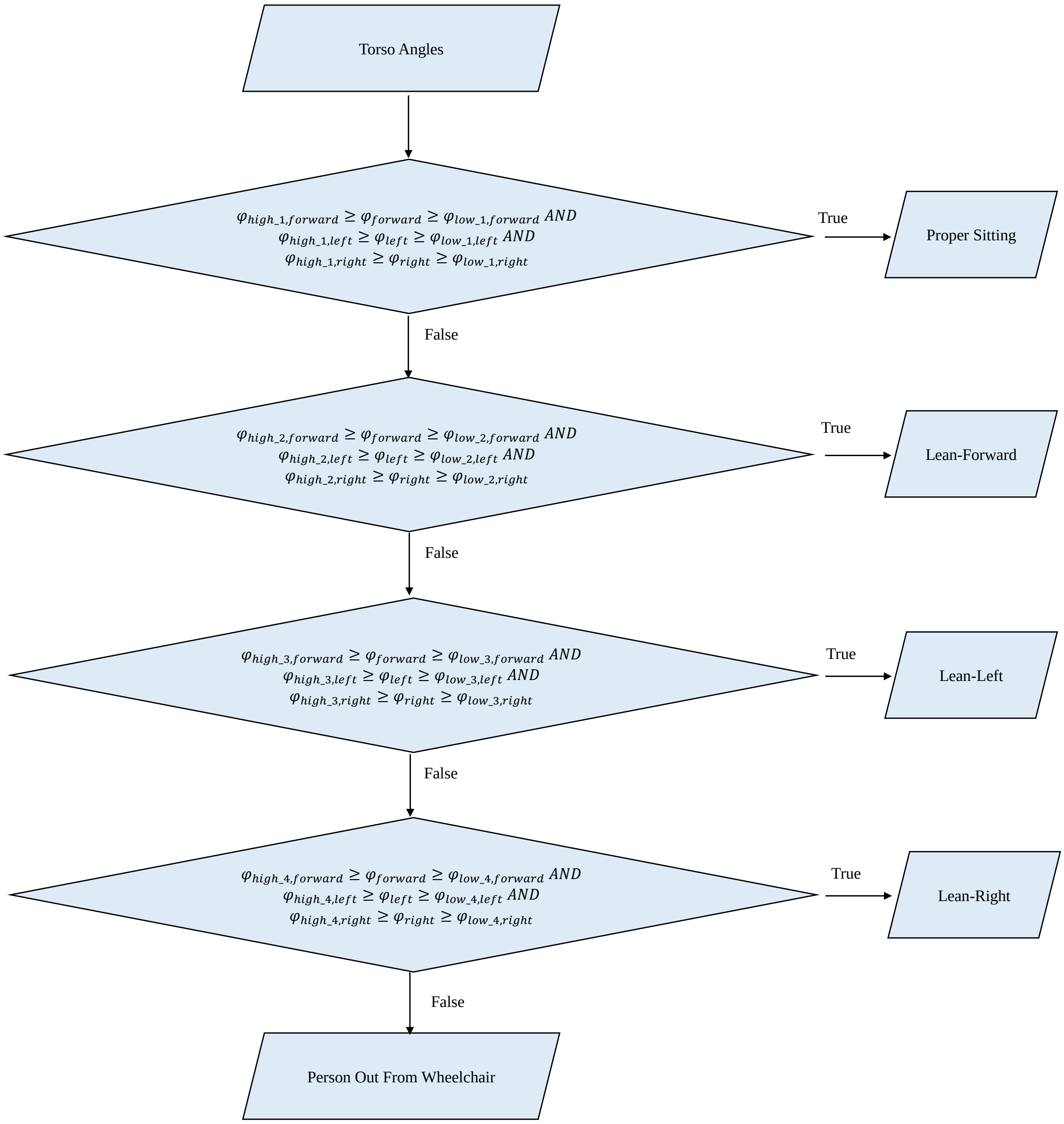

Figure 4.5: Logical Flow Diagram of Algorithm 2

4.1.3 System Testing and Results

The system parameters (upper and lower limits of torso angles) were found by performing the 4 sitting postures on a manual wheelchair with the participation of 29 individuals aged 23 to 65 (Mean-38.9, SD-13.7) in a simulated domestic environment inside the laboratory.

Thus, the values of $\phi_{high\_forward}$, ϕ_{high_left}, ϕ_{high_right} and $\phi_{low\_forward}$, ϕ_{low_left}, ϕ_{low_right}in Equations (4.10) to (4.12) could be found separately for the 4 postures.

It must be noted here that, the system performance does not depend upon the camera angle, because the measuring of the torso angles is done with respect to the world coordinates (vertical axis through the waist).

Table 4.1 to Table 4.4 show the lower and upper limits of torso angles for the 4 postures as obtained from the experimental observations.

Table 4.1: Lower and Upper Limits of Torso Angles for Posture Proper Sitting

Torso Angle	**Lower Limit (degrees)**	**Upper Limit (degrees)**
$\phi_{1,forward}$	335	342
$\phi_{1,left}$	31	37
$\phi_{1,right}$	30	38

Table 4.2: Lower and Upper Limits of Torso Angles for Posture Lean Forward

Torso Angle	**Lower Limit (degrees)**	**Upper Limit (degrees)**
$\phi_{2,forward}$	349	360
$\phi_{2,left}$	24	28
$\phi_{2,right}$	21	26

Table 4.3: Lower and Upper Limits of Torso Angles for Posture Lean Left

Torso Angle	**Lower Limit (degrees)**	**Upper Limit (degrees)**
$\phi_{3,forward}$	33	38
$\phi_{3,left}$	339	346
$\phi_{3,right}$	55	59

Table 4.4: Lower and Upper Limits of Torso Angles for Posture Lean Right

Torso Angle	Lower Limit (degrees)	Upper Limit (degrees)
$\phi_{4,forward}$	329	333
$\phi_{4,left}$	45	50
$\phi_{4,right}$	343	348

Then, the system algorithm (Algorithm 2) was tested upon 116 scenarios and the accuracy of identifying each posture was as follows.

- Proper-Sitting - 24 out of 29 (82.8%)
- Lean-Forward - 25 out of 29 (86.2%)
- Lean-Left - 27 out of 29 (93.1%)
- Lean-Right - 26 out of 29 (89.7%)

The system determined 5 of the ‘Proper-Sitting’ as other postures (4 as Lean-Forward, 1 as Lean-Right), and therefore it has the lowest accuracy of 82.8%. The posture ‘Lean-Forward’ was correctly identified in 25 occurrences and thus has an accuracy of 86.2%. Here, 3 of the ‘Lean-Forward’ were identified as ‘Proper-Sitting’ as, the participants bent forward only slightly. In an another instance of performing ‘Lean-Forward’, the participant tilted to the right and thus, it was identified as a ‘Lean-Right’. The system has an accuracy of 93.1% in identifying the posture ‘Lean-Left’ as it wrongly determined 2 of the ‘Lean-Left’ scenarios as ‘Proper-Sitting’. The posture ‘Lean-Right’ was identified with an accuracy of 89.7% as 2 of the ‘Lean-Right’ were resulted as ‘Proper-Sitting’ and 1 as a ‘Lean-Forward’.

From the observations made, it could be found that the overall system is having an acceptable accuracy in identifying each posture. However, the posture ‘Proper-Sitting’ is having the lowest accuracy and the reason for this is that not every individual is having their torso exactly upright when sitting freely and thereby the angles sometimes deviate from the considered limits.

4.2 Fall Detection of Wheelchair Users

The main objective of the proposed fall detection system for wheelchair users is to detect falls quickly so that a responsible party could be informed in time to minimize any possible negative consequences.

As sitting on a wheelchair always tend to have the waist almost stationary and at the same level from the ground, the distance from waist to the ground can be assumed to be kept constant unless the person rises or falls. As the person considered here is having a mobility impairment and the lower limbs lack adequate strength, regardless of the way in which the person falls, the waist will ultimately rest on the ground as in Figure 4.6.

Thus, a fall can be easily identified by measuring and analyzing the distance from waist (joint spineBase) to ground (floorPlane) as shown in the Figure 4.7.

In the proposed system, an incident is considered to be a fall, if the distance from the waist to the ground gets lower than the usual value, within a short time. Here, both the distance from waist to ground as well as the velocity of waist are considered, in order to discriminate a fall from a normal activity.

Figure 4.6: When a Person with Mobility Impairment falls from a Wheelchair, the Waist will ultimately rest on the Ground

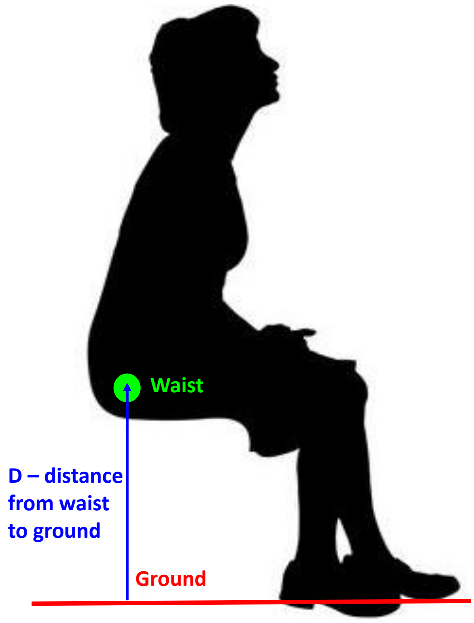

Figure 4.7: How Distance from Waist to Ground is measured

In situations such as, the person sitting on the ground for some special task like meditation, where the velocity is significantly lower than during a fall, the person may be supported by another person, and securely made to sit on the ground slowly, without causing any harm to his/her body, unlike in a fall.

4.2.1 Calculation of Waist Position and Velocity

In order to measure the distance from the waist to the ground, it is required to know the position of the waist and identify the floor plane. The LightBuzz Vitruvius Framework provides special functions to accurately recognize the floor plane in 3D space as well as to measure the distance between a point in 3D space and the floor plane [66].

The position of the waist is obtained by finding the position of the joint Spine Base using the Statement (4.13).

$$var\ waist = body.Joints[JointType.SpineBase].Position; \tag{4.13}$$

Some floor plane detection functionality is already built into the Kinect SDK and with Vitruvius Framework the floor plane can be easily detected by using the Statement (4.14).

$$var\ floor = frame.Floor(); \tag{4.14}$$

To measure the distance between a point in 3D space and the floor plane, the Vitruvius Framework offers the DistanceFrom method which can be used as shown in the Statement (4.15), where the 3D position of the considered point is passed as the argument to the function.

$$double\ distance = floor.DistanceFrom(point); \tag{4.15}$$

In the DistanceFrom method, Vitruvius uses the Point-Plane Distance Formula given in Equation (4.16). Here, D (the distance from the point to the plane) is obtained by projecting w (a vector from the plane to the point) onto v (the normal vector to the plane) as depicted in Figure 4.8.

$$D = \frac{ax_0 + by_0 + cz_0 + d}{\sqrt{a^2 + b^2 + c^2}} \tag{4.16}$$

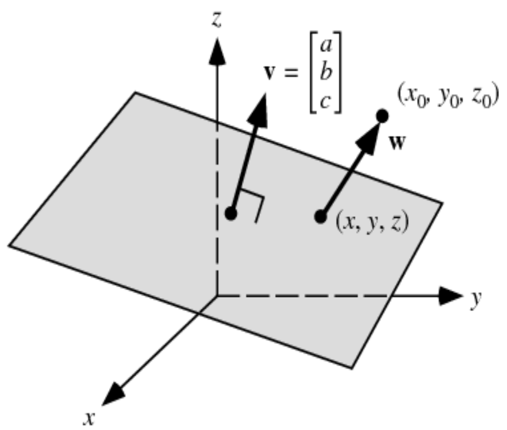

Figure 4.8: Depiction of Point-Plane Distance Formula [67]

Thereby in the proposed system, the distance from waist to the ground is found using the Statement (4.17).

$$double\ waistToFloor = floor.DistanceFrom(waist); \tag{4.17}$$

The velocity of waist (V) is calculated by considering the change in distance from waist to ground (D) during every 0.5 seconds by using the Equation (4.18).

$$\begin{aligned} Distance\ at\ t_1 &= D_1 \\ Distance\ at\ t_2 &= D_2 \\ t_2 - t_1 &= 0.5\ seconds \\ V &= \frac{D_2 - D_1}{0.5} \end{aligned} \tag{4.18}$$

When a person is sitting on a wheelchair, the distance from waist to ground D is approximately constant. But this D is different from one person to another depending on the body structure of the individual. Therefore, as usual values of D, a range is considered. Thereby, if D is in between D_{low_usual} and D_{high_usual}, the person is considered to be "Sitting on Wheelchair".

During a fall, D reduces and reaches a minimum D_{min} while the velocity of waist V increases and reaches a maximum V_{max}. These D_{min} and V_{max} values get varied from one scenario of falling to another. Therefore, the system considers a range for them as shown in Equations (4.19) and (4.20).

$$D_{high\_min} \geq D_{min} \geq D_{low\_min} \tag{4.19}$$

$$V_{high\_max} \geq V_{max} \geq V_{low\_max} \tag{4.20}$$

Thereby, the proposed system identifies an incident as a "Fallen from Wheelchair to Ground" if, D is lower than or equal to D_{high_min} and V is greater than or equal to V_{low_max}.

If a particular incident is neither identified as a "Sitting on Wheelchair" nor as a "Fallen from Wheelchair to Ground", such an incident is considered as a "Out from Wheelchair for an ADL" as presented in Algorithm 3. The logical flow diagram of Algorithm 3 is given in Figure 4.9.

Algorithm 3 Identification of Fall from a Wheelchair

Require: Distance from Waist to Ground, Velocity of Waist

Ensure: Fall from a Wheelchair

```
if D_high_usual ≥ D ≥ D_low_usual then
    Sitting on Wheelchair
else
    if D_high_min ≥ D AND V ≥ V_low_max then
        Fallen from Wheelchair to Ground
    else
        Out from Wheelchair for an ADL
    end if
end if
```

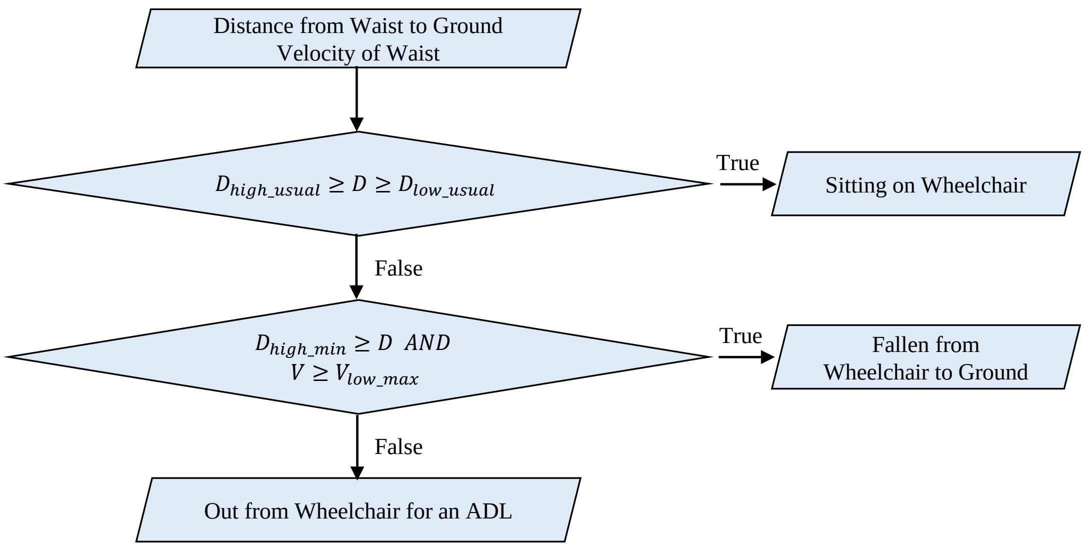

Figure 4.9: Logical Flow Diagram of Algorithm 3

4.2.2 System Testing and Results

Falling from a wheelchair was performed using a manual wheelchair in a simulated domestic environment inside the laboratory with the participation of 23 individuals in the age range of 22-49 (Mean-32.9,SD-9.3). The participants were informed that the person concerned in the proposed system is supposed to have a mobility impairment and thus lacks strength in the lower limbs. However, they were not aware of the way in which the system identifies a fall from a wheelchair.

While the individuals were freely seated on the wheelchair, their waist positions were monitored and thereby their usual D values were found. By analyzing all the D values thus obtained, the actual values of D_{high_usual} and D_{low_usual} were found as 0.52 m and 0.46 m respectively.

When the participants performed different activities including falls from wheelchair and other activities of daily living that requires getting off from wheelchair, the respective D_{min} and V_{max} values were calculated and recorded. After performing several such empirical studies, the higher and lower limits of D_{min} in Equation (4.19) were obtained as 0.13 m and 0.08 m respectively. The magnitudes of V_{high_max} and V_{low_max} in Equation (4.20) were found as 1.28 m/s and 0.96 m/s respectively. Thus, the system identifies an incident as a fall if, D gets lower than D_{high_min} (0.13 m) and V gets higher than V_{low_max} (0.96 m/s).

Figure 4.10 and Figure 4.11 depict how D and V changed during a fall performed by one of the participants. Here, the value of D_{min} was 0.08 m and the magnitude of V_{max} was 1.18 m/s. Therefore, the system was able to successfully identify it as an incident of fall.

Similarly, to test the performance of proposed system in distinguishing falls from activities of daily living that require getting off from wheelchair, an experiment was carried out. The confusion matrix for falls and non-falls for a sample of 33 scenarios with the participation of 23 individuals is given in Table 4.5.

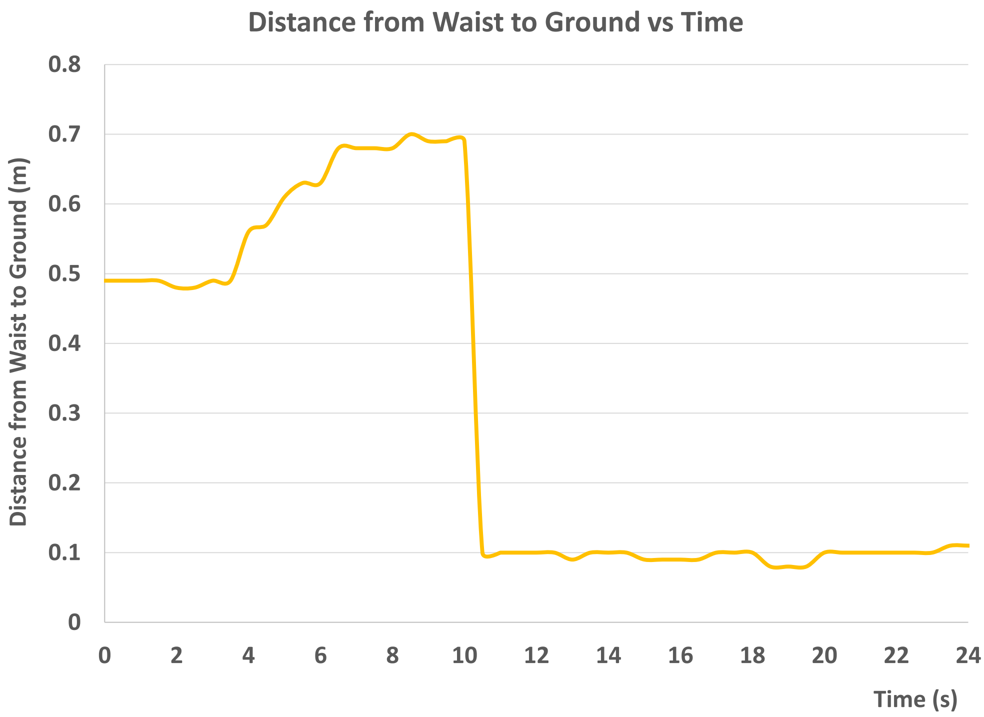

Figure 4.10: Change of Distance from Waist to Ground with Time during a Fall

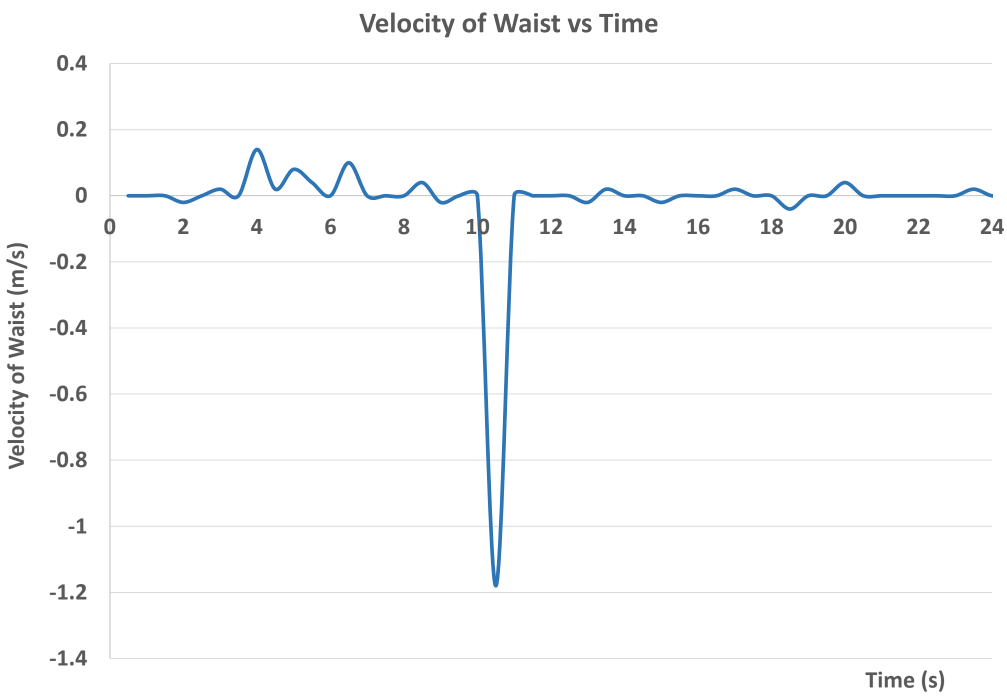

Figure 4.11: Change of Velocity of the Waist with Time during a Fall

Table 4.5: Results obtained for Algorithm of Fall Detection from a Wheelchair

	Predicted scenarios	
Total scenarios	**Falls**	**Non-falls**
Falls	True Positives (TP) - 19	False Negatives (FN) - 2
Non-falls	False Positives (FP) - 0	True Negatives (TN) -12

The total sample consisted of 21 falls and 12 non-falls. The system was able to successfully identify 19 out of the 21 falls and thus, has a sensitivity of 90.5%. The reason for identifying 2 of the falls as non-falls was that, in those 2 scenarios the distance from waist to ground was not lower than D_{high_min}. This was because, the individuals did not completely rest on the ground when performing the fall.

The system has a specificity of 100% as it was able to identify all the 12 non-falls correctly. Therefore, according to the experimental results, the proposed fall detection system for wheelchairs has an acceptable accuracy of 93.9%.

Chapter 5

EMERGENCY NOTIFICATION SYSTEM

Figure 5.1 shows the system developed for the automatic notification of emergencies. Once the system receives a message indicating occurrence of an emergency, it enters the 'GPS Location Tracker' unit where it is checked whether any contact persons are available within 5 kilometer radius from the person under concern. If so, those people are contacted simultaneously and if there are no such people, or if they do not respond, then the system enters the 'Reinforcement Learning Algorithm' unit where a list of contact persons is contacted successively in an order of descending Q-Value. The Q-Values are calculated based on two parameters; 'Probability of Answering a Call' (PA) and 'Level of being Busy' (LB) of persons on the contact list. Once a successful contact is made, the emergency notification happens by an Android Application. This system was developed using Android Operating System with Android Studio 3.0 and Java 8. The Q-Model was trained using Pycharm in Python and the trained Q-Model was sent to the Android application over Firebase Real-time Database.

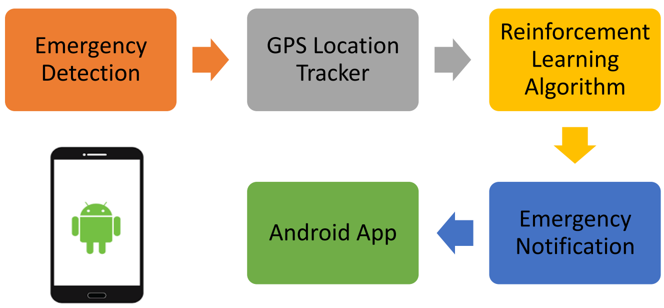

Figure 5.1: Reinforcement Learning based Notification System

5.1 Android Operating System

Mobile apps have become extremely popular over the last few years and offer a great opportunity for the developers as well as the users. As Android offers a unified approach to the application development of mobile devices, it has become one of the most popular mobile systems utilized in the present world with almost 2 billion devices activated. Android is a mobile operating system based on a modified version of Linux, originally developed by a start-up of the same name, Android. In 2005, Google purchased Android and started to take over its development work.

Google developed codes for low-level operations and built the required middleware to power and use electronic devices, and gave Android free of charge to the community who wanted to write codes and build operating systems with it. They even included an application framework, so that third-party applications could also be built and installed.

In 2014, Google introduced Android Studio as the official IDE (Integrated Development Environment) for Android app development and eventually it became the standard. A single downloading of Android Studio includes everything that is needed to begin the development of an Android application. The download package has the Software Development Kit (SDK), which includes all the Android libraries that may be required in developing a mobile application, and the infrastructure to download many Android emulator instances, so that the developers can initially run their applications without the need of real devices.

The best recommended and most convenient way of developing Android applications is by using the Java programming language. Therefore, in developing the proposed system, the standard and official way of developing Android applications that is by using Java with Android SDK and Android Studio IDE was adopted.

5.2 Reinforcement Learning for the Notification System

With the ever-advancing technology, the use of Artificial Intelligence (AI) for tasks such as identifying patterns, learning from experience, and finding novel solutions to challenges etc. is continuously growing. AI simply refers to the simulation of human intelligence in machines which are programmed to think similar to humans and mimic their actions.

Machine learning (ML) is an application of AI that enables systems to automatically learn and improve from experience without being explicitly programmed. There exists 3 types of ML as Supervised Learning, Unsupervised Learning and Reinforcement Learning.

Though both Supervised and Reinforcement learning use mapping between input and output, RL uses rewards and punishments as signals for positive and negative behavior of the agent where as Supervised learning provide the agent with correct set of actions for performing a task.

As compared to Unsupervised learning, Reinforcement learning is different in terms of goals. While the goal in Unsupervised learning is to find similarities and differences between data points, that of Reinforcement learning is to find a suitable action model to maximize the total cumulative reward of the agent.

After comparing the functionalities and the goals of the 3 types of ML, Reinforcement Learning (RL) was chosen as the most suitable for the development of the proposed Notification System, as the scenario of a person answering a phone call is mostly random and the possible situations are unlimited. In addition, it is practically impossible to train an agent for all those situations. But, as RL enables the agent to learn in an interactive environment by trial and error using feedback from its own actions and experiences, it is possible to achieve better results by training the agent for a limited number of situations.

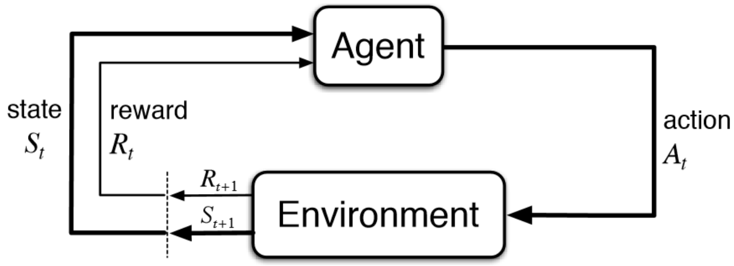

Figure 5.2: Typical Reinforcement Learning Cycle

The typical RL cycle is depicted in Figure 5.2. At time t, the RL Agent observes state $\mathrm{S_t}$ from the environment and receives a reward $\mathrm{R_t}$. The agent then takes an action $\mathrm{A_t}$. In response to action $\mathrm{A_t}$, the environment provides the next state $\mathrm{S_{t+1}}$ and reward $\mathrm{R_{t+1}}$, and the process continues.

The system was developed using Q-Learning which is an off-policy, model-free RL algorithm based on the well-known Bellman Equation. The Markov Decision Process (MDP) for the Q-Model of the proposed system is as follows.

- Agent: Android Mobile App
- Environment: List of Emergency Contacts
- Action: Making Successive Calls
- State: The Contact Person's Status of being 'Free' or 'Busy' at a given instance in time
- Reward: For a successful action, a positive reward proportional to the contact's probability of answering is awarded. For an unsuccessful action, a negative reward is imposed as a failure penalty

5.3 Training of Reinforcement Learning Agent

Building and training of the Q-Model was done using Python in PyCharm platform. It is important to note that the Q-Model was built on arbitrary data of 10 emergency contact persons.

Firstly, each person was assigned with a probability of answering a call, which was a random value in-between 10% to 80%. This was referred to as 'Probability of Answering' (PA), an initial detail of every contact person. This indicates that if a particular contact person receives 100 phone calls, he/she will answer PA number of calls and will not answer (100-PA) number of calls.

At each episode, every contact person was assigned with another detail called 'Level of being Busy' (LB), by assuming 3 persons to be less busy (may include people staying home), 4 persons to be moderately busy and the rest to be highly busy (may include people engaged in an occupation). This was achieved by giving a random value from 0% to 25% for less busy, 25% to 50% for moderately busy and 50% to 100% for highly busy.

Thus, every contact person got 2 details namely 'Probability of Answering' (PA), and 'Level of being Busy' (LB).

For a given instance in time, each contact person got a Contact Status of being 'Free' or 'Busy' which was chosen depending on the respective PA and LB. For an example say, a particular person had a very high PA and a very low LB. Then that person is highly probable to get assigned with the Status of being 'Free' and thereby, it would be significantly effective to contact that person first and therefore would have the highest q-value.

The Q-Model was trained for 100 episodes, meaning 100 random combinations of 'Free' or 'Busy' Contact Status corresponding to 100 different instances in time. At each episode, 50 iterations were made, where each iteration corresponded to a State in the Q-Model. If the n^{th} State has 'x' number of contacts then the $(n+1)^{th}$ State has only 'x-1' contacts. That is, once a particular contact is attempted, that contact is not called again until all the remaining contacts are tried. After calling all the contacts, the Next State gets all 10 contacts again. So that each person can be attempted for a maximum of 5 times within the 50 iterations. After each iteration, the probabilities of answering and the Q-values of the Q-Table were updated.

Equations 5.1 and 5.2 were used to update probabilities of answering after each successful and unsuccessful attempt respectively. The Q-Values of the Q-Table were updated according to the Equation 5.3.

$$new\_probability = (((current\_probability * 100) + 1)/101) * 100 \tag{5.1}$$

$$new\_probability = ((current\_probability * 100)/101) * 100 \tag{5.2}$$

$$\begin{split} new\_q\_value = ((1 - learning\_rate) * current\_q\_value + \\ learning\_rate * (reward + discount\_factor * maximum\_future\_q\_value)) \end{split} \tag{5.3}$$

If the RL Agent chooses a contact with Status 'Free' in a particular iteration, then that episode is escaped and the training of next episode is started so that a new set of 'Free' and 'Busy' status gets trained. If the RL Agent chooses a contact with Status 'Busy', then the iteration loop continues. Each time where the Agent selects a person with Status 'Free', is considered as a successful attempt, and a positive reward proportional to the particular PA of the contact person is awarded. Each time that the Agent selects a contact person with Status 'Busy', is considered as an unsuccessful attempt, and a negative reward is imposed as a failure penalty. After performing all 100 episodes the contact persons' probabilities of answering and the Q-table are uploaded to the Firebase Real-time Database to be accessed by the Android Application.

Algorithm 4 gives the algorithm used for training the Q-Model. The logical flow diagram of Algorithm 4 is given in Figure 5.3.

Algorithm 4 Training of Q Model

Initialize: episode = 1, epsilon = 1
Require: PA of contacts, epsilon decay constant
for episode in range 100 **do**
    Initialize: iteration = 1
    Require: LB of contacts
    Assign: Contact Status 'Free' or 'Busy' depending on PA and LB
    for iteration in range 50 **do**
        Require: random [0:1]
        if random $\geq$ epsilon **then**
            Choose the contact with maximum future q value
        else
            Choose a random contact
        end if
        epsilon = epsilon - epsilon decay constant
        if Contact Status == 'Free' **then**
            Award positive reward
            Update PA of contacts and q values of q table
            Go to next episode
        else
            Award negative reward
            Update PA of contacts and q values of q table
            Go to next iteration
        end if
    end for
    Go to next episode
end for
Upload PA of contacts and q values of q table to Firebase Real-time Database

---

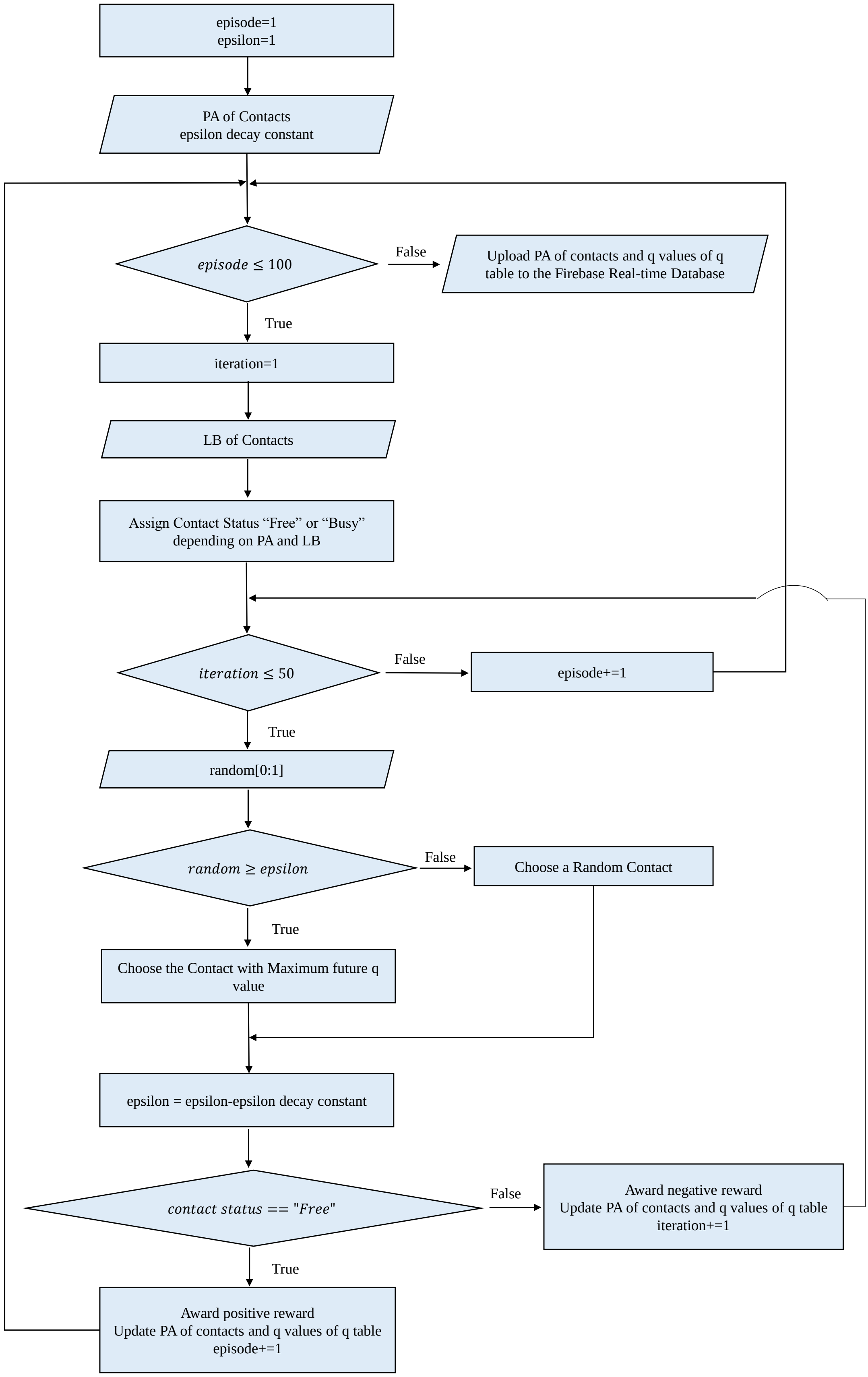

Figure 5.3: Logical Flow Diagram of Algorithm 4

It must noted that, at the beginning of training, epsilon-greedy policy was adopted, where exploration was encouraged. So that at the beginning, random actions were also taken without choosing only the action with max.future q value. With time exploitation dominated. This was achieved by having a decaying epsilon, which decayed by midway of the episodes. This was done by using an 'epsilon_decay_constant' which was calculated as shown in the Equation 5.4.

$$epsilon\_decay\_constant = epsilon/(end\_epsilon\_decay - start\_epsilon\_decay) \tag{5.4}$$

In Equation 5.4, the variable 'epsilon' carried the initial value of epsilon. The variables 'end_epsilon_decay' and 'start_epsilon_decay' corresponded to the episode numbers for starting and ending the decaying of epsilon, which were episode 1 and episode 50 respectively in this application.

5.4 Android Application for Emergency Notification

The Android Application provides both manual as well as automatic emergency notification capabilities. The interface of the developed application has a large icon with the label 'EMERGENCY' as shown in Figure 5.4. By simply touching this icon, the elder can trigger the calling function. The same calling function can be automatically triggered if a fall gets detected by the MIRob in either way described in Chapter 3 or Chapter 4.

Once a notification needs to be done, firstly the system will look whether there is a contact person within 5 kilometer radius from the elder. This is done by comparing GPS location data of the elder and the contact persons. For that, the latitude and longitude values of the location of the elder and the contact persons are updated in a Firebase Real-time Database every 5 seconds as shown in the Figure 5.5.

Figure 5.4: Android Application for Emergency Notification

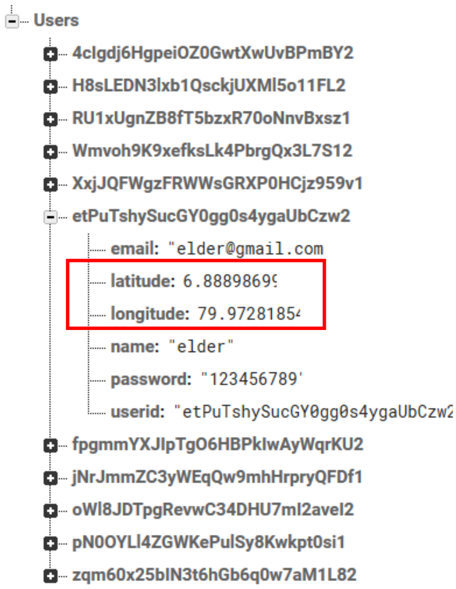

Figure 5.5: Firebase Real-time Database with GPS Location Data

If the system finds a contact person within 5 kilometer radius from the elder as the Contact Person A in Figure 5.6, then that person will be contacted first. If that person fails to respond or if there is no such person within 5 kilometer radius, then the system will enter the Reinforcement Learning based calling algorithm, which makes successive calls to a list of emergency contacts in an order made based on Q-Learning.

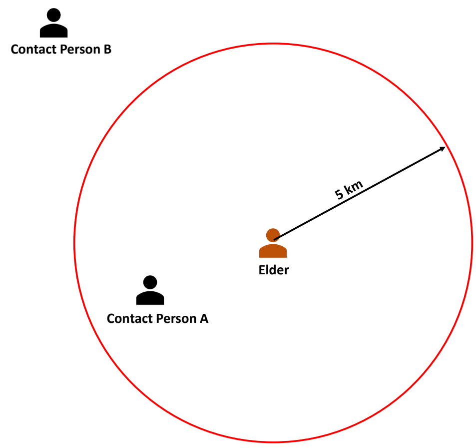

Figure 5.6: Contact Person within 5km Radius from the Elder are called first

The RL based calling algorithm initially uses the probabilities of answering and the q values of the contact persons that were uploaded to the Firebase Real-time Database as described in Section 5.3. By taking those details into consideration the system comes up with an order of calling, where the person with the maximum q-value out of a list of emergency contacts is tried first.

For example, consider the situation shown in Figure 5.7. Here, the Contact Person 5 is having the highest q-value, thus the system calls that person first.

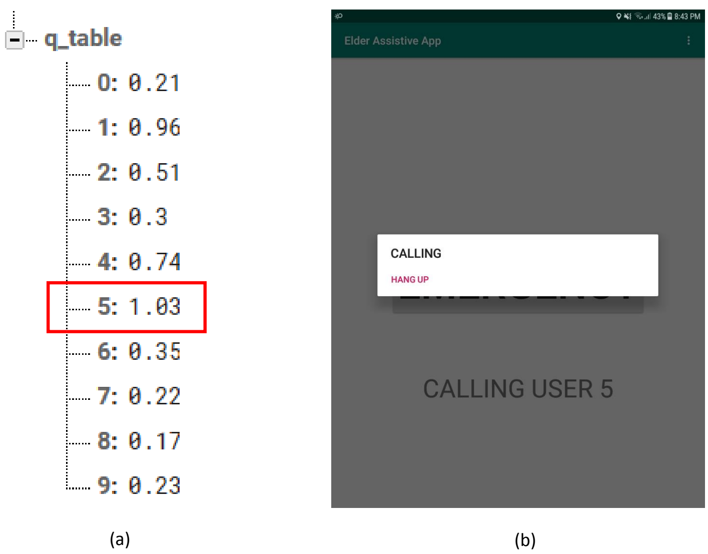

Figure 5.7: Android Application calling Contact Person with Maximum Q Value (a) Q Values of Contact Persons (b) Android Application

If a call gets unsuccessful (rejected, or ended), the contact person with the next maximum q-value is called. It must be noted that, each person will be ringed for a maximum of 10 seconds, and if no response is received, the system will move on to the next person on the list. After each attempt, the probabilities and the q-values are updated on the Firebase Real-time Database. The system will keep on calling the contact persons successively according to the descending order of q-value, until a successful contact is made and the delivery of the emergency notification to a responsible party is accomplished.

5.5 Reinforcement Learning based Calling Algorithm Testing

The proposed Reinforcement Learning (RL) based Calling Algorithm was tested in a simulated environment for different random combinations of probability of answering (PA) to ensure that the system is versatile enough to adapt to various situations.

For each combination of PA, the contact order suggested by the RL Agent was compared with the contact orders presented by 3 middle-aged human volunteers by applying them to 100 unique instances in time. Each instance in time had a random combination of 'Free' or 'Busy' Status based on the respective probability of answering and the level of being busy.

The results obtained for two such combinations of PA are graphically shown in Figure 5.8. The red color graphs correspond to the behavior of the RL Agent and the graphs of the colors yellow, blue and green correspond to the behavior of the 3 human volunteers. As observed in Figure 5.8 (a) for case 1, the RL agent was able to perform considerably better and quicker than the human candidates with a maximum of 3 attempts. In case 2 also, as noticeable in Figure 5.8 (b), the RL agent seemed to perform quite effectively with 88% of the instances being successful from the first attempt itself.

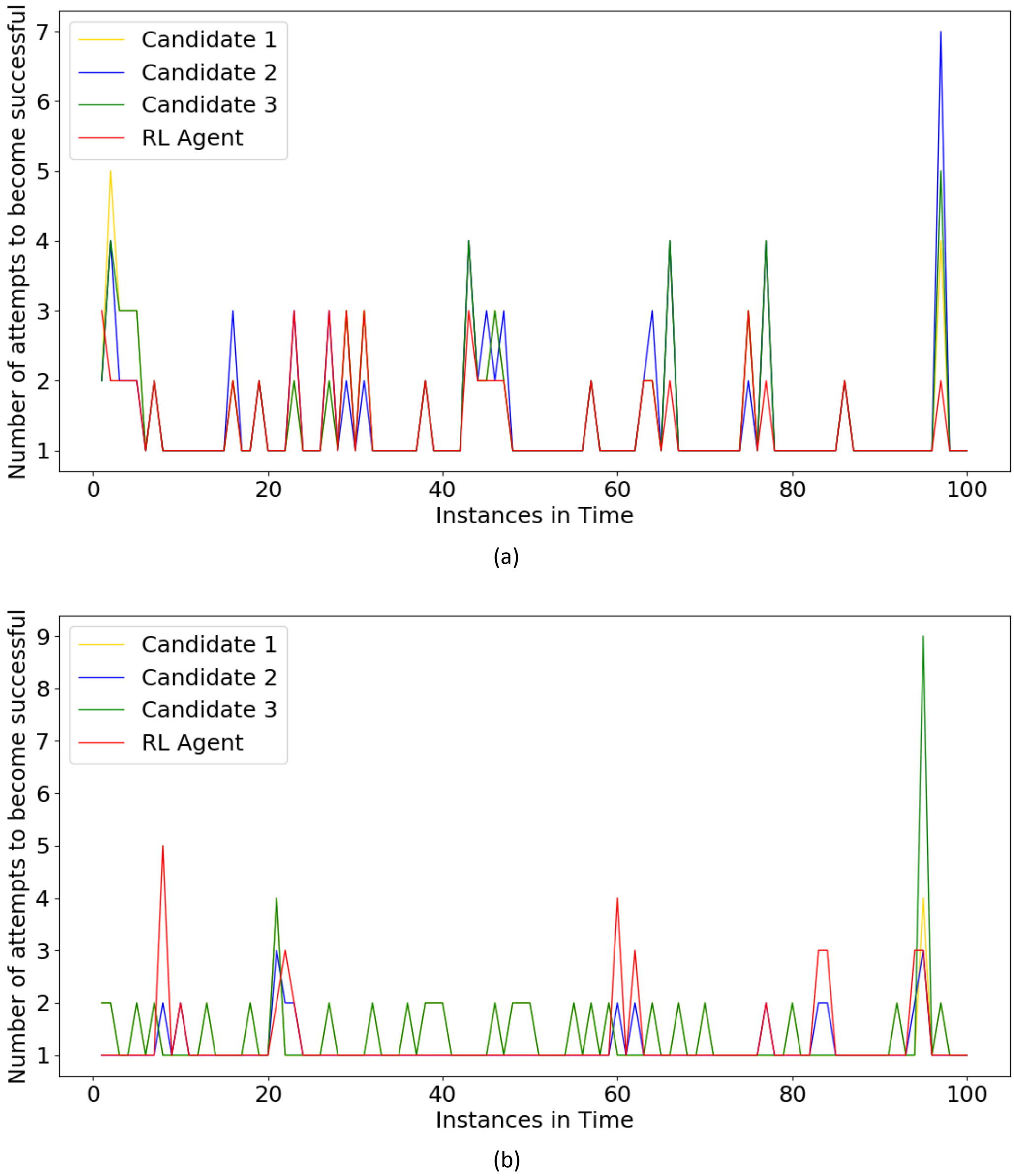

Figure 5.8: Comparison of Results obtained by the Proposed Calling System for Two Unique Cases

Accordingly, it was observable that the RL Agent is able to perform quite similar to a human in most cases and even better in some cases. In situations where multiple contact persons had considerably alike conditions (PA and LB), the RL Agent seemed to perform more easily and quickly than the humans, because humans find those situations hugely confusing.

Chapter 6

CONCLUSIONS

This research proposes a smart home system for the elderly and the disabled with the intention of improving the quality of life by providing ample support in their daily activities while ensuring a safe environment for independent living.

Most often, the elderly and the disabled face troubles when carrying out their daily tasks independently. The mere thought of the inability to perform those activities on their own, makes them frustrated. The insufficiency in support services makes the elderly and the disabled excessively dependent upon their families, which restrains them from being economically active and socially included [68]. In spite of all the emotional stress they have got to undergo, the smart home can help give them exceptional strength and hope for better living conditions [69].

The Smart home technology can highly facilitate aging-in-place or growing old at home, by providing the elderly with emergency assistance, fall prevention/detection, reminder systems, medication administration and many more [70]. It is a promising and cost-effective way of improving home care for the elderly and the disabled in a non-obtrusive way, allowing greater independence, maintaining good health and preventing social isolation [71].

6.1 Evaluation of the System

The overall proposed smart home system is shown in Figure 6.1 and the functional overview of the overall system is given in Figure 6.2.

It basically consists of three subsystems, which are integrated over an IoT Cloud developed using Google Firebase Real-time Database. The subsystem called 'Fall Detection and Fall Type Identification System for Walking or Standing Positions' is discussed under Chapter 3. Chapter 4 explains the subsystem 'Posture Identification and Fall Detection System for Wheelchair Users'. The subsystem 'Reinforcement Learning based Emergency Notification System' is described in Chapter 5.

The three subsystems work together to provide the elderly and the disabled with two main services; Fall Detection and Emergency Notification, and promise to offer great support and safety in leading their daily lives.

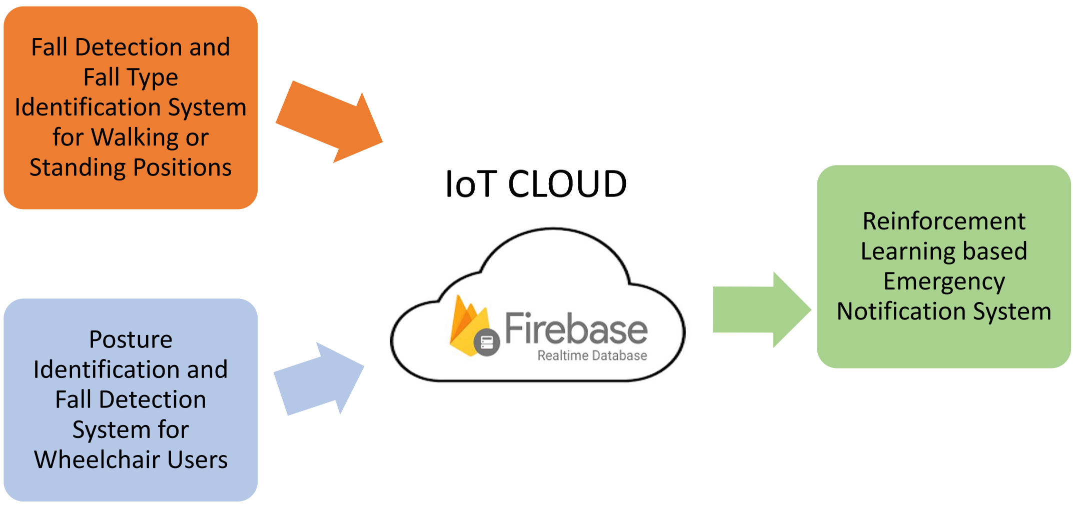

Figure 6.1: Overall Proposed Smart Home System

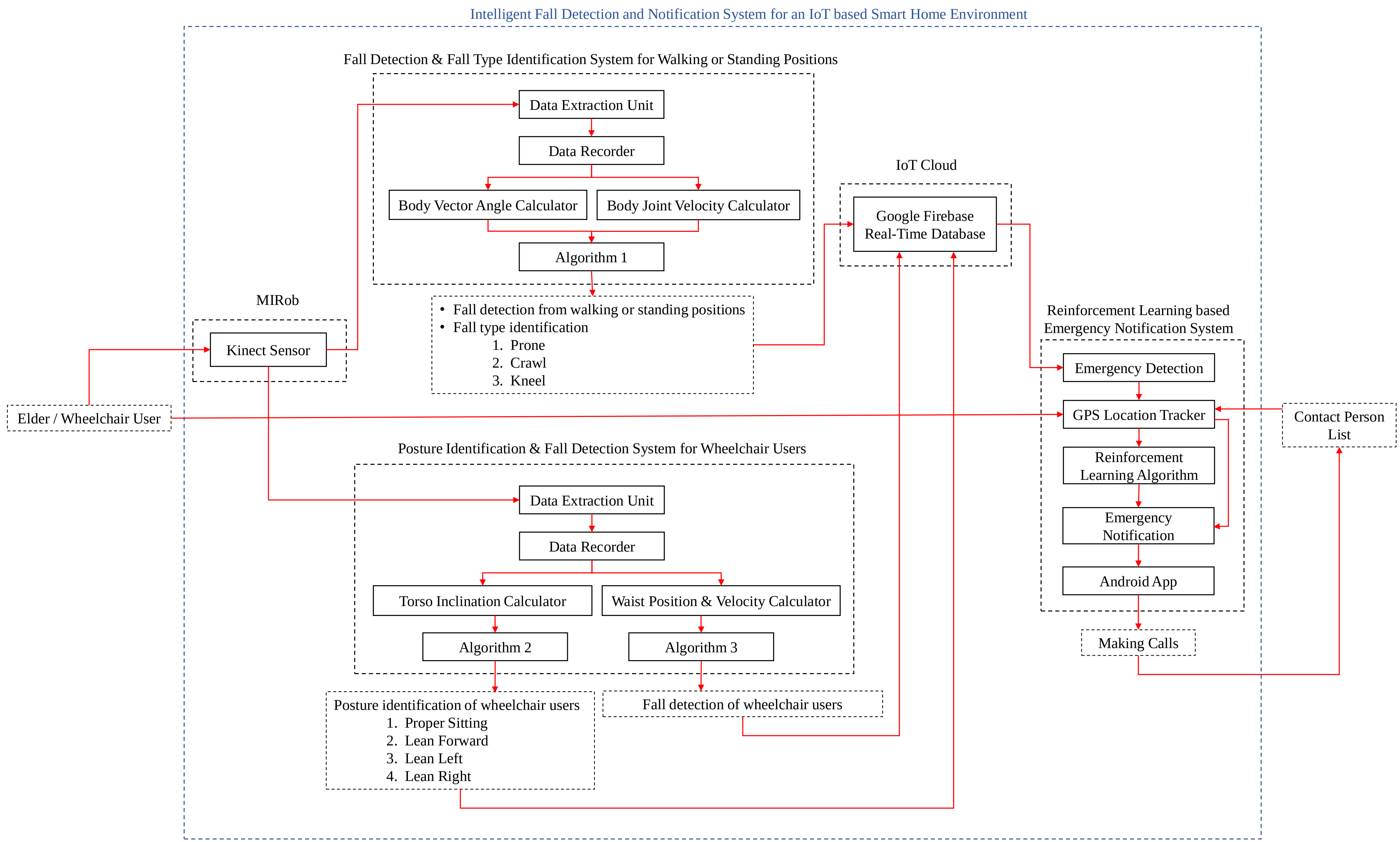

Figure 6.2: Functional Block Diagram of the Overall System

The Fall Detection takes place in two different ways as described below and both of them are capable of automatically triggering the Emergency Notification System.

- Fall Detection from standing or walking position by the MIRob
- Posture Identification and Fall Detection of Wheelchair Users by the MIRob

Fall detection from both standing and seated positions as well as sitting postures are detected by the system automatically with the help of a service robot called MIRob, which obtains visual input through a Microsoft Kinect Sensor.

If the user detects any other emergency that the system cannot identify automatically, then the system provides the user with the facility of signaling the IoT cloud manually by providing an 'Emergency' button in an Android app.

Once the IoT cloud receives an emergency signal either automatically or manually, the emergency notification system starts to operate. Thus, a responsible person will get informed regarding the emergency in time, so that the elder or the disabled person will get support quickly enough to minimize any possible negative consequence.

Thereby, the proposed overall system is capable of providing the elderly and the disabled with adequate support as well as ensure safety and improve their independence and quality of living.

6.1.1 Fall Detection and Fall Type Identification System for Walking or Standing Position

This is a novel vision based approach of detecting falls using the skeleton data obtained from the Kinect sensor. This can be implemented in real-time and can ensure timely detection of falls. Further, the system overcomes the limitations of the existing similar systems, which are discussed in detail under Section 2.6.

A fall is detected by considering the velocities of three body joints; Spine Shoulder, Spine Mid and Spine Base, and the orientations of two body vectors; Spine and Femur. A fall is distinguished from an activity of daily living by waiting for 't' seconds, after encountering a 'Transition Period'.

Once a fall is detected, it is categorized in to one of the three types of falls; Prone, Crawl and Kneel, which will enable more focused medical attention.

The system was tested in a simulated domestic environment inside the laboratory and was found to have an accuracy of 92.5%, a sensitivity of 95.45% and a specificity of 88%. Also, the system was able to discriminate among the three types of falls with accuracy 100% for Prone, 80% for Crawl and 86% for Kneel.

6.1.2 Posture Identification and Fall Detection System for Wheelchair Users

This method of posture identification and fall detection for wheelchair users is a novel vision based mechanism involving a Kinect sensor and Vitruvius Framework. This can operate in real-time and can detect falls fast. This system provides solution to the limitations of similar systems, which are explained in Section 2.6.

The system is capable of discriminating among four sitting postures; Proper Sitting, Lean Forward, Lean Left and Lean Right by considering the angles made by the torso in three directions. A fall from the wheelchair is detected by observing the distance from waist to floor and the velocity of waist.

Experiments were conducted in a simulated domestic environment inside the laboratory and the system could identify Proper Sitting with an accuracy of 82.8%, Lean Forward with 86.2%, Lean Left with 93.1% and Lean-Right with 89.7%. Further, it could detect a fall with an accuracy of 93.9%, a sensitivity of 90.5% and a specificity of 100%.

6.1.3 Reinforcement Learning based Emergency Notification System

This is a novel method of emergency notification using a Reinforcement Learning agent with an Android application. This assures timely notification with a proper feedback mechanism. This system can solve the limitations found in similar systems, which are described in detail in Section 2.6.

The system utilizes two parameters namely; probability of answering and level of being busy of the contact persons to come up with an order of calling based on a Q-Learning algorithm. The GPS location data is also taken in to account when making the order of making calls.

The Reinforcement Learning based calling algorithm was tested on a simulated environment and it was found to perform similar to a human in most cases and even better in some cases. Further, as an elder is more likely to become nervous when making decisions during an emergency, this automatic notification system is more effective in notifying emergencies to responsible parties.

6.2 Limitations of the System

The major drawback in testing the fall detection system was experimenting on actual scenarios. All the experiments were conducted in simulated environments with the participation of healthy individuals. Therefore, the parameter limits found by the experiments may differ from those encountered in actual incidents. However, the proposed algorithms will still guarantee the obtained levels of performance after modifying the parameter limits so as to suit the actual scenarios.

The proposed system cannot automatically identify each and every possible emergency that could occur to a person. It can identify a limited set of emergencies as discussed throughout this thesis, but as the system offers manual triggering as well, the proposed system will assure acceptable performance in a daily basis.

For the continuous performance of the proposed system, we must always ensure an uninterrupted network connection, as every element is interconnected via an IoT cloud. The system cannot be developed over a local database due to 2 major reasons. One is that, the notification system takes GPS location data into consideration when making the required decisions. The other is that, the emergency calling is done over the internet because the mobile network providers do not allow tracking of call status which is an essential feature in operating the proposed notification system.

6.3 Recommendations for Future Developments

The system can be implemented in an elders' home or a hospital and the parameter limits can be adjusted accordingly with values from actual scenarios, so that one of the major drawbacks discussed in Section 6.2 can be overcome. Another possible way of overcoming this drawback is the modification of the system algorithm, so that the system automatically adjusts the parameter limits to suit each individual by getting trained for sometime before being implemented.

The accuracy levels of the emergency detection system can be improved by testing the system with a wider variety of ADLs and falls. Further, health emergencies can also be identified by monitoring and analyzing the body parameters such as heart rate and pressure using suitable sensors.

The emergency notification system can be improved and made more precise by including details regarding the health condition of the elder, and involving more details about the response patterns of the contact persons.

Further, research can be extended towards the smart control and navigation of the wheelchair as well, so that the users are provided with better support and comfort.

LIST OF PUBLICATIONS

1. T. Kalinga, C. Sirithunge, A. G. Buddhika, P. Jayasekara and I. Perera, "A Fall Detection and Emergency Notification System for Elderly," 2020 6th International Conference on Control, Automation and Robotics (ICCAR), Singapore, Singapore, 2020, pp. 706-712, doi: 10.1109/ICCAR49639.2020.9108003.

BIBLIOGRAPHY

[1] United Nations. (2015). World population ageing report. World Popul Ageing Rep, 2015.

[2] United Nations Population Division, World Population Prospects: The 2017 Revision(New York: United Nations,2017)

[3] Chatterji, S., Byles, J., Cutler, D., Seeman, T., & Verdes, E. (2015). Health, functioning, and disability in older adults—present status and future implications. The lancet, 385(9967), 563-575.

[4] World Health Organization. (2001). International classification of functioning, disability and health: ICF. Geneva: World Health Organization

[5] Verdonschot, M. M., De Witte, L. P., Reichrath, E., Buntinx, W. H. E.,& Curfs, L. M. (2009). Community participation of people with an intellectual disability: A review of empirical findings. Journal of Intellectual Disability Research, 53(4), 303-318.

[6] Mitra, S., Palmer, M., Kim, H., Mont, D., & Groce, N. (2017). Extra costs of living with a disability: A review and agenda for research. Disability and health journal, 10(4), 475-484.

[7] Muller-Kluits, N., & Slabbert, I. (2018). Caregiver burden as depicted by family caregivers of persons with physical disabilities. Social Work, 54(4), 493-502..

[8] Yoshida, S. (2007). A Global Report on Falls Prevention: Epidemiology of Falls. Geneva, Swtizerland. World Health Organization.

[9] Bergen, G. (2016). Falls and fall injuries among adults aged 65 years. United States, 2014. MMWR. Morbidity and mortality weekly report, 65.

[10] Florence, C. S., Bergen, G., Atherly, A., Burns, E., Stevens, J., & Drake, C. (2018). Medical costs of fatal and nonfatal falls in older adults. Journal of the American Geriatrics Society, 66(4), 693-698.

[11] World Health Organisation. (2015). WHO World Report on Aging and Health.

[12] Centers for Disease Control and Prevention. (2014). Falls Among Older Adults: An Overview.

[13] Karlsson, M. K., Magnusson, H., von Schewelov, T., & Rosengren, B. E. (2013). Prevention of falls in the elderly?a review. Osteoporosis international, 24(3), 747-762.

[14] Chaccour, K., Darazi, R., El Hassani, A. H., & Andres, E. (2016). From fall detection to fall prevention: A generic classification of fall-related systems. IEEE Sensors Journal, 17(3), 812-822.

[15] Amin, M. G., Zhang, Y. D., Ahmad, F., & Ho, K. D. (2016). Radar signal processing for elderly fall detection: The future for in-home monitoring. IEEE Signal Processing Magazine, 33(2), 71-80.

[16] Skubic, M., Harris, B. H., Stone, E., Ho, K. C., Su, B. Y., & Rantz, M. (2016, August). Testing non-wearable fall detection methods in the homes of older adults. In 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC) (pp. 557-560). IEEE.

[17] Sposaro, F., & Tyson, G. (2009, September). iFall: an Android application for fall monitoring and response. In 2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 6119-6122). IEEE.

[18] Villaverde, A. C., R-Moreno, M. D., Barrero, D. F., & Rodriguez, D. (2016, October). Triaxial Accelerometer Located on the Wrist for Elderly People?s

Fall Detection. In International Conference on Intelligent Data Engineering and Automated Learning (pp. 523-532). Springer, Cham.

[19] Ropero, F., Vaquerizo-Hdez, D., Muñoz, P., Barrero, D. F., & R-Moreno, M. D. (2019). LARES: An AI-based teleassistance system for emergency home monitoring. Cognitive Systems Research, 56, 213-222.

[20] Gjoreski, H., Bizjak, J., & Gams, M. (2016, September). Using smartwatch as telecare and fall detection device. In 2016 12th International Conference on Intelligent Environments (IE) (pp. 242-245). IEEE.

[21] Wu, F., Zhao, H., Zhao, Y., & Zhong, H. (2015). Development of a wearable-sensor-based fall detection system. International journal of telemedicine and applications, 2015, 2.

[22] Santiago, J., Cotto, E., Jaimes, L. G., & Vergara-Laurens, I. (2017, January). Fall detection system for the elderly. In 2017 IEEE 7th Annual Computing and Communication Workshop and Conference (CCWC) (pp. 1-4). IEEE.

[23] Wu, L., Lu, J., Zhang, T., & Gong, J. (2016, October). Robot-assisted intelligent emergency system for individual elderly independent living. In 2016 IEEE Global Humanitarian Technology Conference (GHTC) (pp. 628-633). IEEE.

[24] Pierleoni, P., Belli, A., Palma, L., Pellegrini, M., Pernini, L., & Valenti, S. (2015). A high reliability wearable device for elderly fall detection. IEEE Sensors Journal, 15(8), 4544-4553

[25] Shoaib, M., Elbrandt, T., Dragon, R., & Ostermann, J. (2010, March). Altcare: Safe living for elderly people. In 2010 4th International Conference on Pervasive Computing Technologies for Healthcare (pp.1-4). IEEE.

[26] Xu, T., Zhou, Y., & Ma, Z. (2016, July). AtHoCare: An intelligent elder care at home system. In International Conference on Digital Human Modeling and Applications in Health, Safety, Ergonomics and Risk Management (pp. 298-305). Springer, Cham.

[27] Yang, L., Ren, Y., & Zhang, W. (2016). 3D depth image analysis for indoor fall detection of elderly people. Digital Communications and Networks, 2(1), 24-34.

[28] Nizam, Y., Jamil, M. M. A., & Mohd, M. N. (2016, September). A depth image approach to classify daily activities of human life for fall detection based on height and velocity of the subject. In International Conference on Movement, Health and Exercise (pp. 63-68). Springer, Singapore.

[29] Zhang, Z., Conly, C., & Athitsos, V. (2015, July). A survey on vision-based fall detection. In Proceedings of the 8th ACM international conference on PErvasive technologies related to assistive environments (p. 46). ACM.

[30] Fischinger, David, Peter Einramhof, Konstantinos Papoutsakis, Walter Wohlkinger, Peter Mayer, Paul Panek, Stefan Hofmann et al. "Hobbit, a care robot supporting independent living at home: First prototype and lessons learned." Robotics and Autonomous Systems 75 (2016): 60-78.

[31] Lin, C. C., Chiu, M. J., Hsiao, C. C., Lee, R. G., & Tsai, Y. S. (2006). Wireless health care service system for elderly with dementia. IEEE Transactions on Information Technology in Biomedicine, 10(4), 696-704.

[32] Lv, Z., Xia, F., Wu, G., Yao, L., & Chen, Z. (2010, December). iCare: a mobile health monitoring system for the elderly. In 2010 IEEE/ACM Int'l Conference on Green Computing and Communications & Int'l Conference on Cyber, Physical and Social Computing (pp. 699-705). IEEE.

[33] Megalingam, R. K., Pocklassery, G., Jayakrishnan, V., Mourya, G., & Thulasi, A. A. (2014, April). Smartphone based continuous monitoring system for home-bound elders and patients. In 2014 International Conference on Communication and Signal Processing (pp. 1173-1177). IEEE.

[34] Burnham, A. D. (2017). U.S. Patent No. 9,717,101. Washington, DC: U.S. Patent and Trademark Office.

[35] Solórzano, S., Rojas-Ortiz, M., López-Molina, R. A., Clairand, J. M., & Pozo-Espín, D. (2018, April). Home Tele-assistance System for Elderly or Disabled People in Rural Areas. In 2018 International Conference on eDemocracy & eGovernment (ICEDEG) (pp. 380-385). IEEE.

[36] Johnson, D. O., Cuijpers, R. H., Juola, J. F., Torta, E., Simonov, M., Frisiello, A., ... & Meins, N. (2014). Socially assistive robots: a comprehensive approach to extending independent living. International journal of social robotics, 6(2), 195-211.

[37] Vasavi, K. P., Kumar, N. U., & Prasad, N. S. (2017). Development of Mitthar-The Companion Robot for Lonely Elderly People. Technology, 8(3), 84-94.

[38] Mackenbach, J. P., Borsboom, G. J. J. M., Nusselder, W. J., Looman, C. W. N., & Schrijvers, C. T. M. (2001). Determinants of levels and changes of physical functioning in chronically ill persons: results from the GLOBE Study. Journal of Epidemiology & Community Health, 55(9), 631-638.

[39] Gill, T. M., Allore, H. G., Hardy, S. E., & Guo, Z. (2006). The dynamic nature of mobility disability in older persons. Journal of the American Geriatrics Society, 54(2), 248-254.

[40] Mollaoğlu, M., Tuncay, F. Ö., & Fertelli, T. K. (2010). Mobility disability and life satisfaction in elderly people. Archives of gerontology and geriatrics, 51(3), e115-e119.

[41] Centers for Disease Control and Prevention. (2020). Disability Impacts All of Us Infographic — CDC. [online] Available at: https://www.cdc.gov/ncbddd/disabilityandhealth/infographic-disability-impacts-all.html [Accessed 21 Feb. 2020].

[42] Inclusive City Maker. (2020). Disabled People in the World in 2019: facts and figures. [online] Available at: https://www.inclusivecitymaker.com/disabled-people-in-the-world-in-2019-facts-and-figures/ [Accessed 21 Feb. 2020].

[43] "Fact Sheet on Wheelchairs", Regional Office for South-East Asia, World Health Organization, October 2010.

[44] Giesbrecht, E. M., Miller, W. C., Mitchell, I. M., & Woodgate, R. L. (2014). Development of a wheelchair skills home program for older adults using a participatory action design approach. BioMed research international, 2014.

[45] Bourgeois-Doyle, D. (2016). The Maker: George Klein and the first electric wheelchair. The Future of Health-Care Engineering, (Skulematters 2016), pp.14-16.

[46] Fehr, L., Langbein, W. E., & Skaar, S. B. (2000). Adequacy of power wheelchair control interfaces for persons with severe disabilities: A clinical survey. Journal of rehabilitation research and development, 37(3), 353-360.

[47] Simpson, R., LoPresti, E., Hayashi, S., Nourbakhsh, I., & Miller, D. (2004). The smart wheelchair component system. Journal of Rehabilitation Research & Development, 41.

[48] Leaman, J., & La, H. M. (2017). A comprehensive review of smart wheelchairs: past, present, and future. IEEE Transactions on Human-Machine Systems, 47(4), 486-499.

[49] Forslund, E. B., Jørgensen, V., Franzén, E., Opheim, A., Seiger, Å., Ståhle, A., ... & Wahman, K. (2017). High incidence of falls and fall-related injuries in wheelchair users with spinal cord injury: a prospective study of risk indicators. Journal of rehabilitation medicine, 49(2), 144-151

[50] Chen, W. Y., Jang, Y., Wang, J. D., Huang, W. N., Chang, C. C., Mao, H. F., & Wang, Y. H. (2011). Wheelchair-related accidents: relationship with wheelchair-using behavior in active community wheelchair users. Archives of physical medicine and rehabilitation, 92(6), 892-898.

[51] Gavin-Dreschnack, D., Nelson, A., Fitzgerald, S., Harrow, J., Sanchez-Anguiano, A., Ahmed, S., & Powell-Cope, G. (2005). Wheelchair-related falls:

current evidence and directions for improved quality care. Journal of nursing care quality, 20(2), 119-127.

[52] Sung, J., Trace, Y., Peterson, E. W., Sosnoff, J. J., & Rice, L. A. (2019). Falls among full-time wheelchair users with spinal cord injury and multiple sclerosis: a comparison of characteristics of fallers and circumstances of falls. Disability and rehabilitation, 41(4), 389-395.

[53] Noury, N., Fleury, A., Rumeau, P., Bourke, A. K., Laighin, G. O., Rialle, V., & Lundy, J. E. (2007, August). Fall detection-principles and methods. In 2007 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (pp. 1663-1666). IEEE.

[54] Ma, C., Li, W., Gravina, R., & Fortino, G. (2016, May). Activity recognition and monitoring for smart wheelchair users. In 2016 IEEE 20th International Conference on Computer Supported Cooperative Work in Design (CSCWD) (pp. 664-669). IEEE.

[55] Ma, C., Gravina, R., Li, Q., Zhang, Y., Li, W., & Fortino, G. (2017, October). Activity recognition of wheelchair users based on sequence feature in time-series. In 2017 IEEE International Conference on Systems, Man, and Cybernetics (SMC) (pp. 3659-3664). IEEE

[56] Hiremath, S. V., Intille, S. S., Kelleher, A., Cooper, R. A., & Ding, D. (2015). Detection of physical activities using a physical activity monitor system for wheelchair users. Medical engineering & physics, 37(1), 68-76.

[57] Lotfi, A., Langensiepen, C., & Yahaya, S. (2018). Socially assistive robotics: Robot exercise trainer for older adults. Technologies, 6(1), 32.

[58] Pavone, L., Ricciuti, P., Cafolla, D., & Pasqua, G. (2019). A Kinect-Based Portable Automatic Gait Analysis System-An Experimental Validation. Biomedical Journal of Scientific & Technical Research, 17(1), 12552-12555.

[59] Magsino, A., Suarez, D., Cruz, C., & Vea, L. (2018, October). Model Development of Lower Body Exercise for the Rehabilitation of Level 6 Filipino Post-Stroke Patients Using Microsoft Kinect Sensor V2. In TENCON 2018-2018 IEEE Region 10 Conference (pp. 1177-1182). IEEE.

[60] Van Der Schalk, I.,& Spruit, M. (2017). Sign-Lingo: Feasibility of a Serious Game for Involving Parents in the Language Development of their Deaf or Hearing Impaired Child.

[61] Barth, C. J., Manjarrez, L., & Lopez, E. (2017). Animatronic Kinect Bear.

[62] Chua, S. H., Limqueco, J. S., Lu, E. L., Que, S. W., & Abuan, D. (2018). Development of a Microcontroller-based Wireless Writing Robotic Arm Controlled by Skeletal Tracking. In 2018 IEEE 10th International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM) (pp. 1-6). IEEE.

[63] M. A. V. J. Muthugala and A. G. B. P. Jayasekara, "MIRob: An intelligent service robot that learns from interactive discussions while handling uncertain information in user instructions," 2016 Moratuwa Engineering Research Conference (MERCon), Moratuwa, 2016, pp. 397-402, doi: 10.1109/MERCon.2016.7480174.

[64] Webb, J., & Ashley, J. (2012). Beginning Kinect Programming with the Microsoft Kinect SDK. Apress.

[65] Docs.microsoft.com. 2020. Cameraspacepoint Structure. [online] Available at: https:// docs.microsoft.com/ en-us/ previous-versions/ windows/ kinect/ dn758354 (v=ieb.10) [Accessed 18 May 2020].

[66] Vitruvius. 2020. Vitruvius — Create Stunning Kinect Apps In Minutes. [online] Available at: https://vitruviuskinect.com/ [Accessed 18 May 2020].

[67] Mathworld.wolfram.com. 2020. Point-Plane Distance – From Wolfram Mathworld. [online] Available at: https:// mathworld.wolfram.com/ Point-PlaneDistance.html [Accessed 18 May 2020].

[68] Domingo, M. C. (2012). An overview of the Internet of Things for people with disabilities. journal of Network and Computer Applications, 35(2), 584-596.

[69] Demiris, G., & Hensel, B. K. (2008). Technologies for an aging society: a systematic review of "smart home" applications. Yearbook of medical informatics, 17(01), 33-40.

[70] Cheek, P., Nikpour, L., & Nowlin, H. D. (2005). Aging well with smart technology. Nursing administration quarterly, 29(4), 329-338.

[71] Chan, M., Campo, E., Estève, D., & Fourniols, J. Y. (2009). Smart homes—current features and future perspectives. Maturitas, 64(2), 90-97