

Hybrid Wheelchair Controller for Handicapped and Quadriplegic Patients

Dissertation

For obtaining the academic title
Doctor Engineer (Dr.-Ing.)

At the Faculty of Computer Science and Electrical Engineering
University of Rostock

Prepared at the Center for Life Science Automation (celisca)



Submitted by:

Mohammed Faeik Ruzaij Al-Okby, born on 15th January 1977 in Thi-Qar, Iraq
Rostock, Germany, 2017

**Printed with the support of the German Academic Exchange Service
“Deutscher Akademischer Austauschdienst – DAAD”**

Reviewers:

1. Reviewer:

Prof. Dr. -Ing. habil. Kerstin Thurow
Institute of Automation
University of Rostock, Germany

2. Reviewer:

Prof. Dr. med. habil. Regina Stoll
Institute of Preventive Medicine
University Medicine Rostock, Germany

3. Reviewer:

Prof. Dr. Emil Jovanov
University of Alabama, Huntsville, USA

Date of Submission: 28.06.2017

Date of Defense: 08.12.2017

To my mother with love

Acknowledgement:

The work in this dissertation would not have been completed without the guidance and unlimited help and support of several persons.

First and foremost, I would like to express my utmost gratitude to my supervisors **Prof. Dr.-Ing. habil. Kerstin Thurow** and **Prof. Dr.-Ing. Norbert Stoll** for their abundant and elaborate guidance, unlimited support and prolific advises. I would especially like to express my thanks and appreciations for **Prof. Kerstin Thurow** and **Prof. Norbert Stoll** to give me the opportunity of this dissertation at the Center for Life Science Automation (celisca) with the very interesting topic under the excellent working conditions as well as for the inspiring conversations and discussions.

Also, I would like to thank **Dr.-Ing. Sebastian Neubert**, the group leader of the LSA-Information Technologies for his inspirational advice and constant encouragement during the preparation of this work.

Furthermore, I would like to thank all my colleagues of the Institute of Automation and the Center for Life Science Automation (celisca) at the University of Rostock, Germany. Without their kinds of support and help to my work and daily life in Germany, this dissertation would not have been finished on time. Especially I would like to thank **Prof. Dr.-Ing. habil. Mohit Kumar**, **PD Dr.-Ing. habil. Heidi Fleischer**, **PD Dr.-Ing. habil. Hui Liu**, **Dr.-Ing. Thomas Roddelkopf**, **Dr.-Ing. Steffen Junginger**, **M.Sc. Martin Adam**, **Dipl.-Ing. Lars Woinar**, **Mr. Heiko Engelhardt**, **Dipl.-Ing. André Geißler**, **Dipl.-Ing. Volker Gatz** and **Dipl.-Ing. Peter Passow**.

Particularly, I would like to express my deepest gratitude and appreciation to my colleagues who have shared with me the hard times and the joyful hugs, **M.Sc. Mohammed M. Ali**, and also, I would like to thank **M.Sc. Mostafa Haghi**, **M.Sc. Ali. A. Abdulla**, **M.Sc. Xianghua Chu**, **M.Sc. Quang Vinh Do**, **M.Sc. Xiangyu Gu**, and **M.Sc. Mazen Ghandour** for kind cooperation.

Last but not least, I would like to dedicate this dissertation to my family: my dear mother **Mrs. Wafiqah Mahmood**, who has spent her life lighting our way to be successful peoples, my father **Mr. Faeik Alokby**, my wife **M.Sc. Manar Alsultani**, and my children **Waeil**, **Ali**, and **Faeik**, for their patience, great love, unlimited support and their relentless motivation in my life.

Table of Contents

Table of Contents	i
List of Figures	v
List of Tables	ix
List of Abbreviations	xi
1– Introduction	- 1 -
1.1 Motivation.....	- 1 -
1.2 Target Users	- 1 -
1.3 Structure of the Dissertation	- 3 -
2- State of the Art	- 5 -
2.1 Drawbacks and Problems Statement	- 5 -
2.1.1 Available Signals for Quadriplegia Patients.....	- 6 -
2.2 Wheelchairs and Rehabilitation Robotics	- 8 -
2.3 Voice and Speech Recognition	- 11 -
2.3.1 Overview.....	- 11 -
2.3.2 Voice Recognition Algorithms	- 23 -
2.3.3 Voice Recognition Errors	- 32 -
2.4 Electromyogram (EMG)	- 33 -
2.5 Electroencephalogram (EEG)	- 38 -
2.6 Motion and Gesture Recognition	- 42 -
2.7 Orientation Sensor	- 47 -
2.7.1 Accelerometer.....	- 47 -
2.7.2 Magnetic Field Sensor (Magnetometer)	- 48 -
2.7.3 Gyroscope	- 48 -
2.8 Summary and Conclusion	- 52 -
3- Work Concept	- 55 -
3.1 System Requirements	- 56 -
3.2 Input Units	- 56 -
3.2.1 Voice Recognition Modules.....	- 57 -

3.2.2 Head Tilt Controller	- 59 -
3.3 The Processing Unit	- 60 -
3.4 Output Units.....	- 62 -
4- System Development	- 65 -
4.1 Microcontroller Unit (MCU)	- 65 -
4.1.1 Cortex-M3 Core.....	- 68 -
4.1.2 Memory System Controller (MSC).....	- 69 -
4.1.3 General Purpose Input/Output (GPIO)	- 69 -
4.1.4 Energy Modes.....	- 69 -
4.1.5 EFM32 UART/USART.....	- 71 -
4.1.6 Inter-Integrated Circuit Interface I ² C.....	- 71 -
4.1.7 Programming Integrated Development Environment IDE.....	- 71 -
4.2 Voice Recognition	- 72 -
4.2.1 Speaker Dependent VR	- 72 -
4.2.2 Speaker Independent VR	- 72 -
4.2.3 Voice Controller Embedded Solution.....	- 73 -
4.3 Orientation Detection.....	- 80 -
4.3.1 Euler Angles.....	- 81 -
4.3.2 Orientation Detection Sensor BNO055	- 82 -
4.4 Test Platforms	- 84 -
4.4.1 Dr. Jaguar Lite Robot.....	- 84 -
4.4.2 Meyra Ortopedia Smart 9.906 Wheelchair	- 88 -
4.4.3 Speed Feedback Unit.....	- 90 -
5- Voice Controller.....	- 93 -
5.1 Realization	- 93 -
5.2 Types of Voice Commands	- 96 -
5.3 Evaluation.....	- 97 -
5.4 Conclusion for the Voice Controller.....	- 104 -
6- Head Tilt Controller.....	- 105 -
6.1 Overview of the Head Tilt Controller.....	- 105 -

6.2 Autocalibrated Algorithm	- 111 -
6.3 Speed Compensation Algorithm	- 114 -
6.4 Experimental Results.....	- 118 -
6.4.1 Test Results for Jaguar-Lite Robot.....	- 118 -
6.4.2 Test Result for Meyra Smart 9.906 Wheelchair	- 120 -
6.5 Functions Embedded in the Head Tilt Controller	- 123 -
6.5.1 Command Confirmation Function	- 123 -
6.5.2 Wrong Orientation Handling Function	- 125 -
6.5.3 Orientation Sensor Error Handling Function.....	- 126 -
6.6 Conclusion for Head Tilt Controller	- 127 -
7- Voice and Head Tilt Controllers Integration	- 129 -
7.1 Voice and Head Tilt Combination	- 129 -
7.2 Testing of Final System Integration	- 131 -
7.3 Time and Distance for Controllers Stop Command.....	- 133 -
7.4 Questionnaire for System	- 136 -
7.5 Conclusion.....	- 138 -
8- Summary and Outlook.....	- 139 -
8.1 Summary.....	- 139 -
8.2 Comparison of Selected Solutions with Previous Work	- 140 -
8.2.1 Voice Recognition.....	- 140 -
8.2.2 Orientation Detection.....	- 140 -
8.3 Outlook.....	- 141 -
References	- 143 -
List of Publications	- 163 -
Dissertation Theses	- 165 -
Abstract.....	- 167 -
Zusammenfassung.....	- 169 -

List of Figures

Figure 1-1: Spinal cord area where injury can cause quadriplegia [3], [4]	- 2 -
Figure 1-2: Active area for quadriplegia patient [5].....	- 3 -
Figure 2-1: Traditional electrical wheelchair [6].....	- 8 -
Figure 2-2: Joystick main functions	- 9 -
Figure 2-3: Segway system for the lower limb paralyzed user [10]	- 10 -
Figure 2-4: EMG-based exoskeleton [13]	- 10 -
Figure 2-5: Voice recognition process flowchart	- 11 -
Figure 2-6: ANN based voice-controller robot [35].....	- 13 -
Figure 2-7: The basic principles of voice controlled wheelchair [40].....	- 14 -
Figure 2-8: Wireless voice controller for robotic systems [41]	- 15 -
Figure 2-9: Voice Controlled Wheelchair Using DSK TMS320C6711 [44]	- 16 -
Figure 2-10: Diagram of functional voice-activated wheelchair [47]	- 17 -
Figure 2-11: Sound Waveform [50].....	- 18 -
Figure 2-12: The robot communicates with the smartphones [51].....	- 18 -
Figure 2-13: System structure of voice-controlled robots [52].	- 19 -
Figure 2-14: FFT for a sample signal [56]	- 20 -
Figure 2-15: Voice recognition unit for a hybrid driving two-arm robot [57] ..	- 21 -
Figure 2-16: Structure of VRCS system [58]	- 22 -
Figure 2-17: Sub-system block diagram [59]	- 23 -
Figure 2-18: Time alignment of two time-dependent sequences [60].....	- 24 -
Figure 2-19: Cost matrix of the two-time sequences [60].....	- 25 -
Figure 2-20: Paths of index pairs for X (N=9) and Y (M=7) [60]	- 26 -
Figure 2-21: Example of the Markov process [70]	- 27 -
Figure 2-22: An example of a hidden Markov model [70]	- 29 -
Figure 2-23: The trellis algorithm [70]	- 31 -
Figure 2-24: Induction step of the forward algorithm [70]	- 31 -
Figure 2-25: SI system structure.....	- 32 -
Figure 2-26: The Levator Scapulae Muscles (LSM) [88].....	- 34 -
Figure 2-27: CyberLink device [90]	- 34 -
Figure 2-28: EMG (B11) in relax state and active state [90].....	- 35 -
Figure 2-29: Schematic diagram of BPANN classification [91].....	- 36 -
Figure 2-30: Framework of the proposed classification system [102]	- 37 -
Figure 2-31: Structure of the system [113]	- 39 -
Figure 2-32: BCI system configuration [115].....	- 40 -

Figure 2-33: EPOC headset [118].....	- 40 -
Figure 2-34: System block diagram [119]	- 41 -
Figure 2-35: System structure of the hand-glove control system [140]	- 43 -
Figure 2-36: Gesture recognition system [141].....	- 44 -
Figure 2-37: The structure of the system [136]	- 45 -
Figure 2-38: Kinect-based wheelchair controller [147]	- 45 -
Figure 2-39: Ultrasonic head controlled wheelchair [138].....	- 47 -
Figure 2-40: Coordinate of the orientation sensor [172]	- 50 -
Figure 2-41: AHRS electronic module [173]	- 50 -
Figure 2-42: View of the system [181].....	- 51 -
Figure 3-1: Schematic overview of the system	- 55 -
Figure 3-2: VR success probability using one and two VR modules.....	- 58 -
Figure 3-3: Suggested system structure	- 64 -
Figure 4-1: EFM32GG990F1024 microcontroller block diagram [194].....	- 68 -
Figure 4-2: Speaker-dependent VR system	- 72 -
Figure 4-3: Speaker-independent VR system	- 73 -
Figure 4-4: SpeakUp click voice recognition module (VR1).....	- 75 -
Figure 4-5: SpeakUp software v 1.00	- 76 -
Figure 4-6: SpeakUp Click module communication with microcontroller	- 76 -
Figure 4-7: Easy VR module.....	- 77 -
Figure 4-8: Easy VR_Commander 3.8.0 software.....	- 78 -
Figure 4-9: Quick T2SI™ lite v_3.1.14 software	- 78 -
Figure 4-10: Easy VR communication with ARM microcontroller	- 79 -
Figure 4-11: Euler angles axes	- 81 -
Figure 4-12: Adafruit BNO 055 Orientation Module	- 82 -
Figure 4-13: BNO055 output forms.....	- 83 -
Figure 4-14: Jaguar-Lite robot	- 85 -
Figure 4-15: Jaguar-Lite robot internal structure [201].....	- 86 -
Figure 4-16: CRC checksum	- 88 -
Figure 4-17: Meyra Smart 9.916 wheelchair	- 89 -
Figure 4-18: SRG0531 wheelchair motor	- 89 -
Figure 4-19: Sabertooth 2x32 Amp motor driver.....	- 90 -
Figure 4-20: Magnetic encoder output signal	- 91 -
Figure 5-1: Voice controller structure	- 94 -
Figure 5-2: Motors speed control for voice controller	- 95 -
Figure 5-3: System status notification on LCD Display	- 96 -
Figure 5-4: VR accuracy for VR1 and VR2.....	- 99 -

Figure 5-5: VR errors for VR1-SD, VR2-SD, and VR2-SI - 99 -

Figure 5-6: Flowcharts for “AND” and “OR” algorithms..... - 100 -

Figure 5-7: “AND” and “OR” VR accuracy - 102 -

Figure 5-8: “AND” and “OR” VR error rate..... - 102 -

Figure 5-9: Flowchart for OR algorithm with FP cancellation function - 103 -

Figure 5-10: VR errors rate for OR-FP cancelation..... - 103 -

Figure 6-1: Block diagram of the head tilt controller..... - 106 -

Figure 6-2: Tilt angles for the head tilt controller..... - 107 -

Figure 6-3: System motors control by head tilt - 109 -

Figure 6-4: Speed control algorithm of head tilt controller - 110 -

Figure 6-5: Head tilt controller at different road slope angles - 112 -

Figure 6-6: Flowchart for auto-calibrated algorithm - 113 -

Figure 6-7: Flowchart for speed compensation algorithm - 114 -

Figure 6-8: Speed compensation procedure for each individual motor - 116 -

Figure 6-9: Tilt to speed ratio of the head tilt controller - 117 -

Figure 6-10: Compensation time of the system motors..... - 119 -

Figure 6-11: Head tilt controller commands success rate..... - 121 -

Figure 6-12: Tasks for system indoor test..... - 122 -

Figure 6-13: System tests for passing a long ramp - 123 -

Figure 6-14: Command confirmation function - 124 -

Figure 6-15: Wrong orientation notification - 125 -

Figure 6-16: Flowchart for orientation modules error handling function..... - 126 -

Figure 7-1: System flowchart for transition between the controllers - 130 -

Figure 7-2: System tilt to speed ratio..... - 131 -

Figure 7-3: Tasks map for system tests - 132 -

Figure 7-4: Voice commands accuracy test results..... - 133 -

Figure 7-5: Stop command reaction time for sub-controllers - 135 -

Figure 7-6: Reaction time and distance for stop command..... - 135 -

Figure 7-7: Questionnaire result for the system - 136 -

List of Tables

Table 3-1: Comparison between selected processing units	- 62 -
Table 4-1: Comparison of selected microcontrollers	- 65 -
Table 4-2: Energy modes for EFM32 microcontroller [194]	- 70 -
Table 4-3: Comparison of embedded VR modules	- 74 -
Table 4-4: Comparison of orientation sensors features.....	- 80 -
Table 4-5: BNO055 operation modes [200]	- 84 -
Table 5-1: System voice commands	- 96 -
Table 5-2: SD tests for VR1	- 97 -
Table 5-3: SD tests for VR2	- 98 -
Table 5-4: SI –HMM tests for VR1	- 98 -
Table 5-5: OR tests for VR1-SD and VR2-SD	- 101 -
Table 5-6: AND tests for VR1-SD and VR2-SD	- 101 -
Table 6-1: Head tilt controller commands description	- 108 -
Table 6-2: Command accuracy tests for system at 2.5 km/h speed	- 120 -
Table 6-3: Test results for the head tilt controller	- 120 -
Table 7-1: System test results	- 133 -
Table 7-2: Questionnaire evaluation questions	- 136 -
Table 7-3: Questionnaire results	- 137 -

List of Abbreviations

ADC	Analogue to Digital Converter
AHRS	Attitude and Heading Reference Systems
ARM	Advanced RISC Machines
ANN	Artificial Neural Network
BCD	Binary Coded Decimal
BCI	Brain Computer Interface
CCS	Code Composer Studio
CMOS	Complementary Metal Oxide Semiconductor
DAC	Digital to Analogue Converter
DSP	Digital Signal Processing
DTW	Dynamic Time Warping
EEG	Electroencephalogram
EMG	Electromyogram
EOG	Electrooculography
FIS	Fuzzy Inference System
FFT	Fast Fourier Transform
FPS	Frame Per Second
GUI	Graphical User Interface
GPIO	General Purpose Input Output
HMI	Human Machine Interface
HMM	Hidden Markov Model
IC	Integrated Circuit
IR	Infrared
LED	Light Emitted Diode
LCD	Liquid Crystal Display
LPC	Linear Predictive Coding
LSM	Levator Scapulae Muscles
MAV	Mean Absolute Value
MCU	Micro Controller Unit
MEMS	Micro Electro Mechanical System
MFCC	Mel Frequency Cepstral Coefficients
NLP	Natural Language Processing
ODS	Obstacle Detection System
OS	Orientation Sensor

PC	Personal Computer
PEN	Performance Evaluation Network
PIC	Peripheral Interface Controller
PCM	Pulse-Code Modulation
PRS	Peripheral Reflex System
PWM	Pulse Width Modulation
RISC	Reduced Instruction Set Computing
SD	Sound Dependent
SI	Sound Independent
SR	Speech Recognition
SDK	Software Development Kit
SNR	Signal to Noise Ratio
SMAPI	Speech Manager Application Programming Interface
TIN	Task Identification Network
TTL	Transistor-Transistor Logic
TTS	Text-To-Speech
USART	Universal Asynchronous Serial Receiver and Transmitter
VR	Voice Recognition

1 – Introduction

1.1 Motivation

The primary goal of this dissertation is to design and develop an intelligent multi-input controller to help patients with quadriplegia, and those who are elderly, paralyzed, have had arms amputated, and are handicapped to drive wheelchairs or rehabilitation robots easily and comfortably. Since some of these users have problems with upper and lower limbs and cannot use the traditional electrical powered wheelchairs which are normally controlled by a joystick, a flexible controller is needed that can be comfortably adapted for them to use for controlling a wheelchair. Instead of a traditional joystick controller, they can use the available body signals in upper body regions which are not affected by their spinal cord injuries or diseases. Even quadriplegia patients with four paralyzed limbs can still control the motion of the head motion, the voice and muscles in the face, neck, and shoulders. The available signals from quadriplegia patient should be used to control a traditional electrical wheelchair without the need for the help of others. Simple and comfortable wheelchair control will compensate for the loss of lower limb functions and enable patients to move around inside the home and also outdoors. This will give these patients the chance to be in contact with the outside world and give them the opportunity to improve their social life.

1.2 Target Users

Quadriplegia and paralyzed patients are the primary users of this work. Quadriplegia is a type of disability affecting the upper and lower limbs of the patient. It is caused by upper spinal cord injury and some diseases which affect the nervous system. Quadriplegia or tetraplegia happens when a person has a spinal cord injury above the first thoracic vertebra T1 (see Figure 1-1), where paralysis usually affects the cervical spinal nerves C1 to C8 resulting in the paralysis of all four limbs. This type of spinal cord injury may lead to partial or complete paralysis of the arms as well as full paralysis of the legs. Some nervous system diseases can cause similar disabilities, such as transverse myelitis, multiple sclerosis, cerebral palsy and strokes. Recently, the number of quadriplegia patients have increased rapidly due to various reasons such as motor vehicle accidents (48%), falls (21%), violence (15%), sports accidents (14%), and

other causes (2%). In the USA, there are 12,000 spinal cord injuries each year [1], [2].

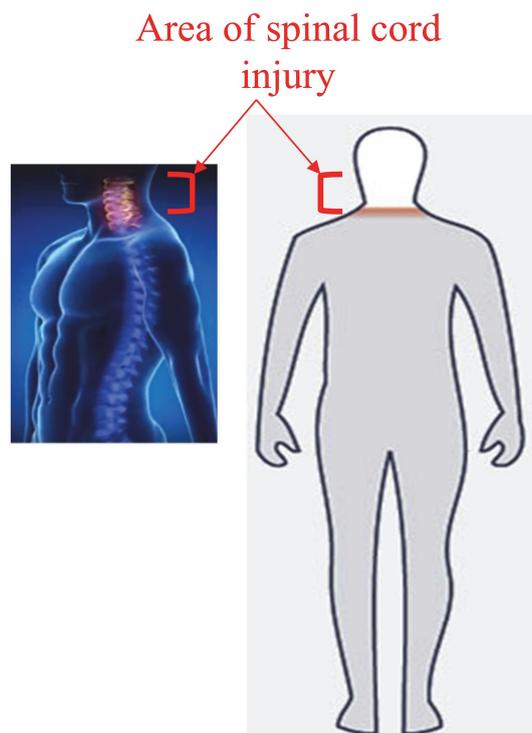


Figure 1-1: Spinal cord area where injury can cause quadriplegia [3], [4]

The life of a quadriplegic patient is very difficult with the loss of control of the four limbs. These patients need the help of other people to perform most of their daily activity. This research aims to help quadriplegic patients to move alone using a traditional electrical wheelchair. The use of electrical wheelchairs requires a clear and meaningful control signal from the user. Unfortunately, quadriplegia patients have lost the easiest way to control a wheelchair by the hand and fingers using a joystick unit. The system proposed in this study must use the remaining controllable signals and actions possible for quadriplegia patients. Some signals require intensive training before they can be used for control purposes.

Figure 1-2 shows the active body regions for the quadriplegia patient marked in yellow. These are the parts of the body that can be used by the patient to send a specific signal that can be used to control the wheelchair or rehabilitation robot. Some control signals from these regions need training to be used correctly by the user as a control signal, such as with the electrical activity of brain cells (EEG) or of the contraction of the muscles in the face, neck, and shoulders (EMG) [5]. The same concept for quadriplegia patients can be applied to help in many similar

disability situations in which the user cannot use a joystick controller, such as people with amputated arms and elderly or paralyzed patients. In addition, the same solution may also be applicable for some robotic control applications in rehabilitation engineering.

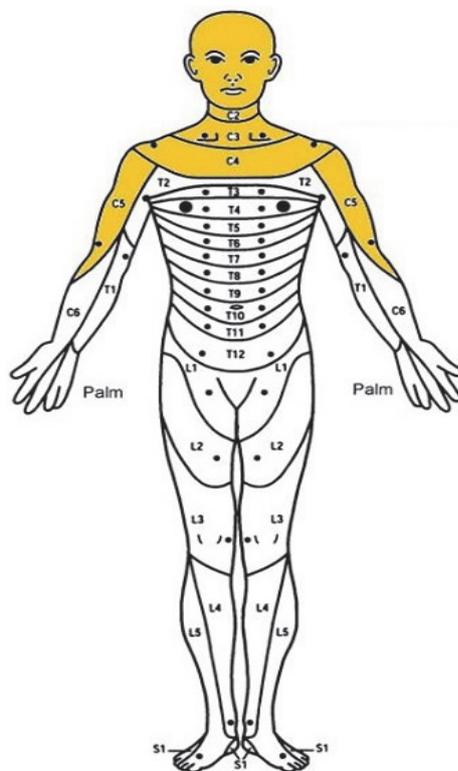


Figure 1-2: Active area for quadriplegia patient [5]

1.3 Structure of the Dissertation

This dissertation consists of eight chapters. This introductory chapter gives a brief description and some statistical facts about the problem to be tackled in this research. It also explains the motivation and overall contribution of this work.

Chapter 2 provides a general description of the traditional electrical wheelchair and its main units. It also includes a short review of the state of the art of available intelligent control systems for wheelchairs and rehabilitation robotics, including voice control, the electroencephalogram, the electromyogram and gesture and motion control. A short review of the available obstacle detection systems for wheelchairs finishes this chapter.

Chapter 3 describes the concept of the presented work. It discusses the research problem and user limitations and design constraints. Introducing the concept of the system, its main parts are introduced, and a short discussion of the solutions available for each part of the system is given. The implementation strategy is then introduced in this chapter.

Chapter 4 describes the system development. The control solution used and technology included are presented. The chapter also gives a brief description of the working principles of the sensors and modules used.

Chapter 5 describes the testing and implementation of the voice controllers that have been used as one of the input methods in the present study. It includes the tests of the voice recognition modules individually and after their combination. The chapter explains the algorithms used to increase the accuracy of voice recognition and also to reduce errors in voice recognition. The results are explained using tables and charts, which clearly show the system performance in different conditions.

Chapter 6 explains the structure and tests of the head tilt controller with the auto-calibrated and speed compensation algorithms used as the second input method in the present study. It describes in detail the algorithm used to enhance the head tilt controller, especially for outdoor environments. The system test result indoors as well as in outdoor environments are discussed.

Chapter 7 describes the integration of the voice and head tilt controller. It includes the flowcharts, diagram that explains the integration process and how the user can activate and use each controller. The chapter contains the experimental results of the validating test and also questionnaire about the system performance for the users of the system.

Chapter 8 provides a summary and comparison of the solutions in the state of the art chapter and the proposed solution. It is also included the outlooks section which discusses the possible future works which can be conducted to improve the current system.

2 - State of the Art

The huge growth in the numbers of elderly people in society, and increases in injuries due to war, accidents, and paralysis are the main reasons for developing new control methods for rehabilitation equipment. Quadriplegics, people who have had hands amputated, and paralyzed patients cannot use the traditional electrical wheelchairs with a joystick controller. They need to use special control methods depending on body signals available in the region of the shoulders, neck, and head. One of the goals of this research is to help these people to control an electric wheelchair by themselves.

One type of active control signal is the voice of the user, which can represent a simple solution to control rehabilitation applications. Voice recognition (VR) technology can be used to achieve this goal. VR technology converts audio signals into electrical signals which can be digitized and processed by a computer or microcontroller as a control signal.

Another control signal can be taken from the user's head, using head movement or head orientation data. This is another possibility for controlling a wheelchair and rehabilitation applications. This method can be implemented using orientation detection (OD) units which can sense and measure any change in the head's position in the three principals (x, y, and z) axes, which refer to the vertical, horizontal, and depth orientation. Many researchers have proposed multi-input control systems including one or more of the different input signals or data from a user's body, such as the electroencephalogram (EEG), electromyogram (EMG), electrooculogram (EOG), voice recognition and body orientation as a controller for a wheelchair or robotic application. The following section explains the general background of wheelchairs and rehabilitation robotics and also some of the different solutions to help elderly, quadriplegic and handicapped people suggested in previous research work.

2.1 Drawbacks and Problems Statement

The main goal of this thesis is to help quadriplegia patients to implement some daily motion activity without the need for others to help them. There are several levels of quadriplegia based on the degree of damage and its location in the spinal cord. Some quadriplegia patients have lost the ability to move any body part below the spinal cord injury in the neck. Other quadriplegia patients have

the ability to move their arms but they cannot control their hands and fingers and cannot perform specific hand and finger motions such as grasping or turning keys on or off. The majority of quadriplegia patients have lost the ability to walk and so that they cannot move around on their own. The loss of movement ability creates real problems and can cause physical and psychological side effects.

A proposed control system for quadriplegic patients must cover all the functions of the joystick unit (see Figure 2-2). A suggested system must be built based on the signals available in the areas of head, neck, and shoulders and the control concept must take into consideration the need to be applicable with minimum user effort. One serious design issue is how to allow the quadriplegic user to control the system's speed and direction easily and comfortably. Which signal or body action for quadriplegics can be used to implement this task?

The present work aims to help quadriplegic patients to perform some of the most vital and important daily activities which involve movement. A basic rehabilitation solution for people with disabilities is the use of an electrical wheelchair. Unfortunately, most quadriplegia patients cannot use the electrical wheelchairs controlled by joystick units because they have lost control of their hands and fingers. Fortunately, there are several signals and actions which can still be controlled by quadriplegic patients depending on their spinal cord injury. The signal chosen for controlling a wheelchair or rehabilitation robot must be carefully selected so as to be well adapted to the user's requirements. The control method must let the user easily and comfortably execute the control commands for the wheelchair.

2.1.1 Available Signals for Quadriplegia Patients

The available signals for quadriplegia patients can be taken from the shoulder, neck, and head areas. Some of the valuable control signals in quadriplegics are the:

- Electrical activity of brain cells (Electroencephalography - EEG)
- Eye movements (Electrooculography - EOG)
- Electrical activity of muscles (Electromyogram - EMG)
- Voice
- Head movement

The electrical activity of brain cells is one of the most important signals for quadriplegia patients. Brain activity generates electrical signals that can be picked up from the scalp using EEG. The EEG signal can be adapted with a BCI unit so as to analyze and interpret brain signals as specific control commands. Brain signals have very small electrical values and are strongly affected by noise from power sources and other body signals. They need to be amplified hundreds or thousands of times before being used. The use of the brain signals to control wheelchairs or rehabilitation robots faces various difficulties. The use of these signals needs very close concentration by the user to generate a specific sample for control commands. This means that the user needs to think only about the command which needs to be given to the wheelchair, and this prevents normal communication with the surrounding environment. This method can be used to generate control commands such as to move forward, but it is very difficult to use it to control the speed of the system in forward mode. Switching among control commands requires time for the user to change thoughts. The use of this method also requires the use of expensive instruments and a computer, which leads to high cost.

The use of eye movements and the EMG encounters similar problems to the EEG. Both can be used to generate single control commands by triggering a signal based on programmable thresholds, but it is very difficult to use such signals for speed control, which requires a continuous and flexible control signal that allows the user change it gradually to increase or decrease the speed of the system. Both signals are also affected by noise and require intensive training, especially with EMG.

The user's voice can be exploited as a controller for quadriplegic patients. It is a simple control signal where huge effort may not be needed to generate voice commands. The voice can be used for both motion and speed and direction commands and can cover all the required functions (see Figure 2-2).

The movement of the head can also be an important solution for speed and direction control. This is an easy control signal that does not require a huge effort or intensive concentration to generate control commands. The user's head movements and tilt around the principal axes can be translated into specific commands and the angle of motion can be used for speed increments and decrements. Head motions cannot cover all control commands in the joystick unit but it can be used with a voice controller to generate a powerful method to satisfy the necessary control requirements for quadriplegic patients.

2.2 Wheelchairs and Rehabilitation Robotics

Wheelchairs and rehabilitation robotics is one of the most important fields in robotics technology. It is used to help the people with disabilities to continue their independent life without the need for external assistance in daily activities like movement using a wheelchair in indoor and outdoor environments.

In the traditional electrical wheelchair, the user needs to use his arms and fingers to control the movement and direction of the wheelchair via a joystick controller. The joystick will convert the user's command into an electrical signal, which is further sent to the motor driver unit and to the light controller. The motor driver unit controls the motor speed and also the direction of the wheelchair. Figure 2-1 shows a traditional electrical wheelchair.



Figure 2-1: Traditional electrical wheelchair [6]

The traditional wheelchair controller consists of two types of control commands. The first type is motion commands for direction and speed which allow the user to drive the wheelchair towards a target direction at the speed required. The motion commands implemented for traditional wheelchairs use the joystick. The user chooses the required direction by changing the joystick direction to the same required direction and the speed of the wheelchair is increased by increasing

the joystick tilt in the selected direction. Some wheelchairs have different speed levels which can be chosen using the joystick keyboard. The second type of control commands are the peripheral control commands which are used to control lights, direction signals, sound alarms and other ancillary equipment. Figure 2-2 explains the main functions of the joystick control unit.



Figure 2-2: Joystick main functions

In many situations, several types of wheelchair users cannot use a joystick to control the wheelchair. In this case, it is important to find another control method that can better interface the user with the wheelchair. These types of wheelchairs are often called “intelligent wheelchairs”.

Recently, a new piece of transportation equipment called the Segway (Segway Inc. New Hampshire, USA) was introduced in 2001. This equipment has been used for persons with lower limb paralysis or amputated legs, and there are different models to solve the resulting physical challenges. In one model, the lower body part of the paralyzed person can be fixed and the patient can move freely by using the top of the body to control the motion of the Segway. Another design uses a chair to let the user sit while driving the Segway [7]–[10]. Figure 2-3 shows the Segway system for paralyzed users.



Figure 2-3: Segway system for the lower limb paralyzed user [10]

Another new rehabilitation robotics technology called Exoskeleton has been developed to help the physically challenged and disabled users in their daily movements. This technique supports the body joint in the paralyzed lower limbs and takes commands from the user via several kinds of the controller like the keypad, EEG-BCI and EMG [11]–[18]. Figure 2-4 shows an EMG-based exoskeleton system.

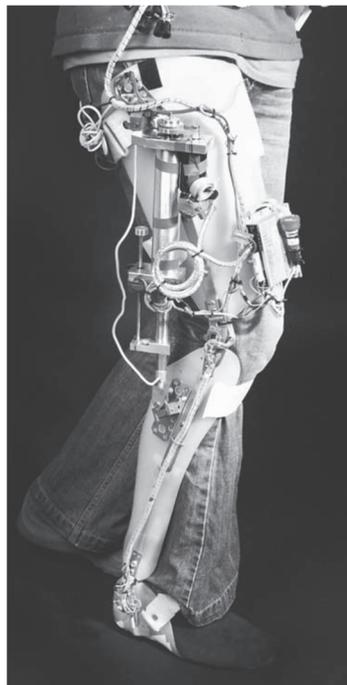


Figure 2-4: EMG-based exoskeleton [13]

2.3 Voice and Speech Recognition

2.3.1 Overview

JIM BAUMANN et al. described two main approaches to implementing voice recognition, which are template matching and feature analysis. Template matching is the easier method and has better accuracy ($\geq 98\%$) when it is used correctly. In this technique, the user needs to speak voice commands into a microphone. The microphone converts the acoustic signal into an electrical signal which is sent to an analog-to-digital converter (ADC) to be digitalized and stored in the memory or microcontroller registers. Each stored command has a specific meaning. When the user gives a live voice command, the system tries to interpret the significance of this command. It compares and matches live commands with previously stored digital command templates in the memory, each of which has a definite meaning. Figure 2-5 explains the process of voice recognition.

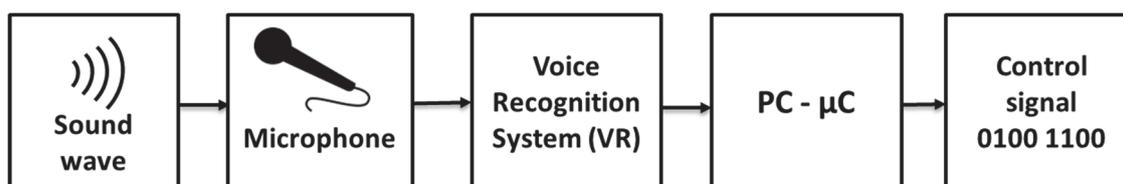


Figure 2-5: Voice recognition process flowchart

Since each user has different voice features, users who want to use a voice recognition system need to train their voice commands with the system before using it. This kind of voice recognition system is called speaker dependent (SD). These systems can be used by a limited number of users. SD systems can be trained with a few hundred voice commands. They give better performance compared with other voice recognition techniques, with accuracy about 98% compared to 90% for speaker-independent systems.

Feature analysis techniques have been used in order to develop more general voice recognition systems which can be used by any user without prior training; these systems are called speaker-independent (SI). Instead of trying to find an exact match between actual and already stored voice commands using template matching techniques, the feature analysis method attempts to process and analyze the voice command to find its statistical features using Fourier transformation or

linear predictive coding (LPC). This technique compares the characteristics and features of the expected live voice command and those of already processed and stored commands. This method can be used with a wide range of users, and its accuracy is between 90-95 % which is less than that of SD [19].

ROCKLAND et al. proposed a voice-activated wheelchair which depends on the use of an embedded IC HM2007 for voice recognition [20]. This IC can be programmed to store 40 spoken commands in 40 registers in the training process. The recognition process depends on a comparison between a spoken command and the 40 stored command templates. This IC receives the voice command through a microphone and converts it into a digital code, which is further compared to already stored commands. The output of this IC is binary-coded decimal (BCD) so that it can easily be understood by the microcontroller. Theoretically, the HM2007 IC can achieve more than 95% accuracy in a non-noisy environment. A similar voice-based wheelchair implementation for HM2007 IC indicates that the system needs to be enhanced to get acceptable accuracy for a real-time application, which is in the range of 90% in noisy environments [21]–[23].

KUBIK et al. proposed a voice controller for a robot by using a cellular phone [24]. The system consists of three personal computers (PCs). The first PC is used to connect the cell phone with the system and to analyze voice commands. The second PC is used to analyze the artificial vision of the robot, and the third computer controls the robot's movements. The system uses special software from IBM to implement voice recognition. This software uses pattern comparisons between stored command samples and new live voice commands. The voice command picked up by the cellular phone is processed by the first PC and the voice recognition results are then sent to the second control PC. However, the use of three computers makes the system very costly and more complex and not suitable for the real application.

SAJKOWSKI et al. reviewed voice recognition algorithms that have been used in voice-activated robotic wheelchairs [25]. Voice processing algorithms, verification methods for the recognition results, sensor data processing algorithms, path planning algorithms and the structure of the control system and master control algorithms proposed in various studies [26]–[29] were analyzed and discussed. The study concludes that it is hard to compare voice recognition system performance depending only on the results published in research papers. Voice recognition accuracy cannot be measured for different systems unless precisely

defined criteria are used because the comparison must be made under the same conditions of vocabulary, noise, microphone type and so on. The authors advise the use of more sensors to avoid the limitations and drawbacks of the voice recognition process and also suggest using fuzzy logic or expert system techniques in the master controller [30]–[32].

CHATTERJEE et al. proposed a particle swarm optimization (PSO) method for training a voice-controller for the robot based on an artificial neural network (ANN). It consists of four main parts: a speech recognizer (SR), task identification network (TIN), performance evaluation network (PEN), and the robot system itself. The human voice command is picked up by a microphone and converted into electrical value used to perform the desired action. The SR will generate a pseudo-sentence, which is then compared to the vocabulary stored in the OOV (out-of-vocabulary) words. The SR operates based on a hidden Markov model (HMM) [33] [34]. The proposed system can be further improved by training the entire FNN trained completely by the PSO. This process requires the automatic determination of membership functions. The authors did not provide information regarding the voice recognition accuracy achieved or the noise levels in the test environments. Figure 2-6 shows the construction of the system [35].

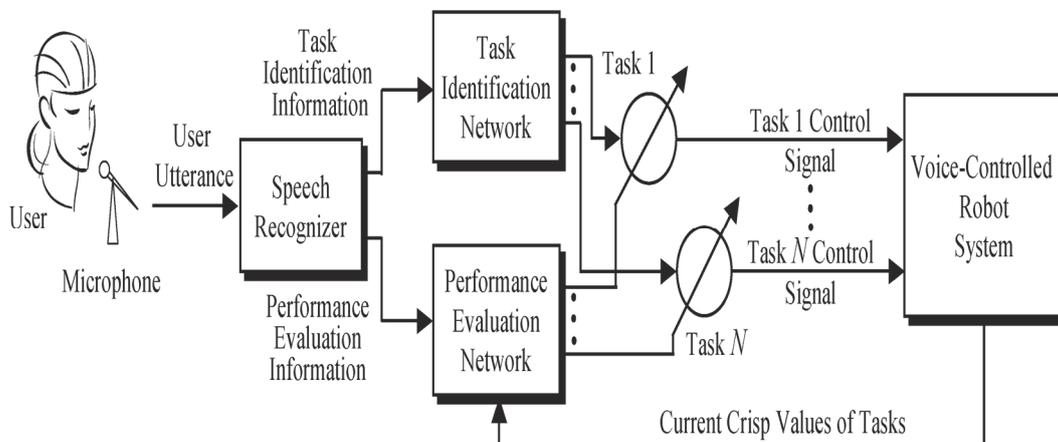


Figure 2-6: ANN based voice-controller robot [35]

PACNIK et al. proposed a voice-operated intelligent wheelchair using a neural network algorithm to realize voice recognition. The proposed work adapted the previous work of **RABINER and JUANG** [36], **FURUI** [37], **ZEGERS** [38] and **KIRSCHING** [39]. The voice recognition process starts by sampling the input signal, then conducting word separation, coefficient dimension reduction, LPC analysis and trajectory sample recognition using a fixed-point approach to neural networks. Figure 2-7 shows the sequence in the application window. This work has some

limitations that lead the system to fail to distinguish between control commands and a simple conversation. The system was tested in two noisy environments, but the noise levels were not measured which is very important for the comparison of results. The system was tested with only two persons and it would need to be tested with more users and different languages. The recognition of voice commands differed from one user to another and the results show the lowest recognition accuracy of 76%. This is not sufficient for rehabilitation purposes, which need excellent recognition with accuracy over 90% [40].

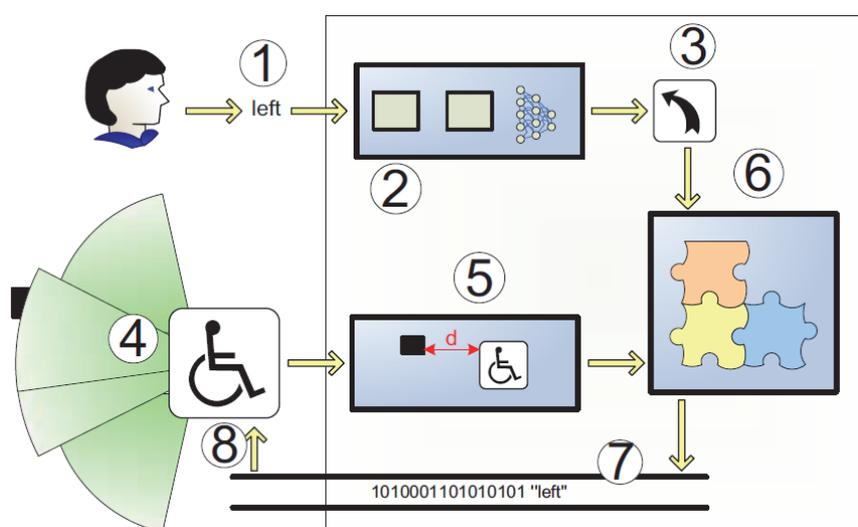


Figure 2-7: The basic principles of voice controlled wheelchair [40]

1. Wheelchair user utters command "left" ("levo" in the Slovene language);
2. Voice recognition system recognizes the command;
3. Voice command transmitted to the system unit designed for evaluation and execution;
4. Wheelchair surrounded by dynamic obstacles is being observed at all times during operation;
5. Special component which measures the distance to the obstacle;
6. Control system picks up all the data and decides whether or not to execute the recognized command;
7. In the case that the recognized command leads to no harm to the user and other people, the voice command is sent to the wheelchair for execution;
8. Wheelchair receives the command and executes it.

LV et al. proposed a speech controller for a robot. In this work, two personal computers were used (PC1 and PC2). A microphone connected to the first PC is used to pick up the voice command. PC1 analyzes the voice command and transfers the required action to the second PC. PC2 is connected to the robot chassis. Communication and the transfer of information between PC1 and PC2 are accomplished using a Wi-Fi connection. PC2 is responsible for control of the robot movement's and Figure 2-8 shows the structure of the system.

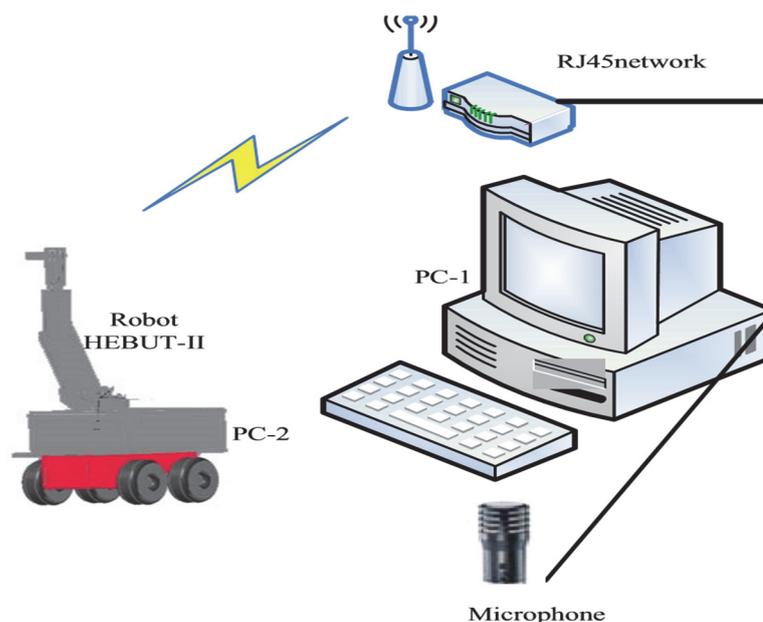


Figure 2-8: Wireless voice controller for robotic systems [41]

A feature extraction technique operating on an acoustic signal received by the microphone has been used for voice recognition. It involves Mel Frequency Cepstral Coefficients (MFCCs) and linear prediction coefficients (LPCs). The recognition process depends on pattern comparison using dynamic time warping (DTW). The use of 2 PCs, however, makes the system more expensive and complicated [41], [42].

QADRI et al. proposed a voice-controlled wheelchair for persons with disabilities. The voice recognition process is performed by measuring the sound energy, zero crossing and standard deviation of the spoken command. This process is completed using special signal processing kits from Texas Instruments Inc. (TMS320C6711 DSP kit). The picked-up voice command is converted into an analog electrical signal and fed to the analog input of the DSP kit. The DSP kit gives specific analog output depending on the input signal from the microphone. The maximum output of the microphone is 520 mV, and the output signal from the DSP kit is amplified using the 741 operational amplifiers and then converted to DC voltage using a bridge rectifier. The DC signal is utilized by the PIC 16F876A microcontroller (Microchip Technology Inc., Arizona, USA) to control the wheelchair's movement. Figure 2-9 shows a block diagram of the system. This system uses a complicated procedure to identify the spoken word. The command must be limited to 0.704 sec, which is enough for a one-word command only. A pre-

analysis of the voice command is required for every person. This system will not give good efficiency if there are any changes in the attitude of the voice, and so it is not suitable for real-time applications. The authors also fail to provide sufficient information regarding the noise levels and recognition accuracy in their tests, which are important when comparing results [43], [44].

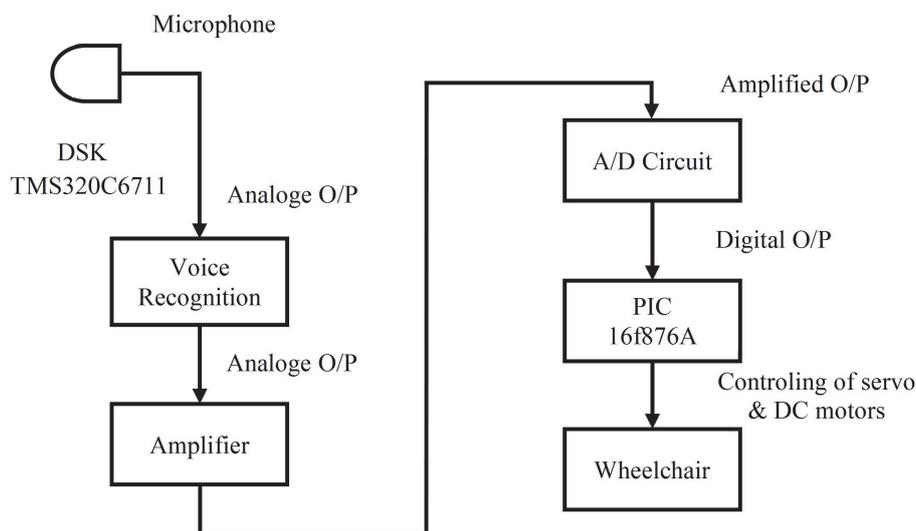


Figure 2-9: Voice Controlled Wheelchair Using DSK TMS320C6711 [44]

MURAI et al. proposed a new functional voice-activated wheelchair. The system uses a commercial electronic powered wheelchair, voice recognition system and laptop with seven peripheral interface controllers (PICs). A headset microphone picks up the user's voice command and the microphone output is sent to the voice recognition software which includes two programs installed on the system laptop. The first program is a grammar-based recognition parser named Julius (Nagoya Institute of Technology, Japan) [45] and the second program is called AquesTalk (AQUEST Corp., Kanagawa, Japan) to allow the system to operate interactively [46]. The laptop also controls the movement of the wheelchair motors. Figure 2-10 shows a diagram of the system. One of the significant drawbacks of this system is the slow recognition process because of the confirmation command which may take around 2-3 seconds, which is a long time. The user needs to wait for the system response to the given command to see if the system has recognized the command correctly. Then the user needs to give another voice command to confirm the first command. The long recognition process makes it inadequate for real-time applications, and the complexity and cost of this system will be high [47]. Similar work is available in [48], [49].

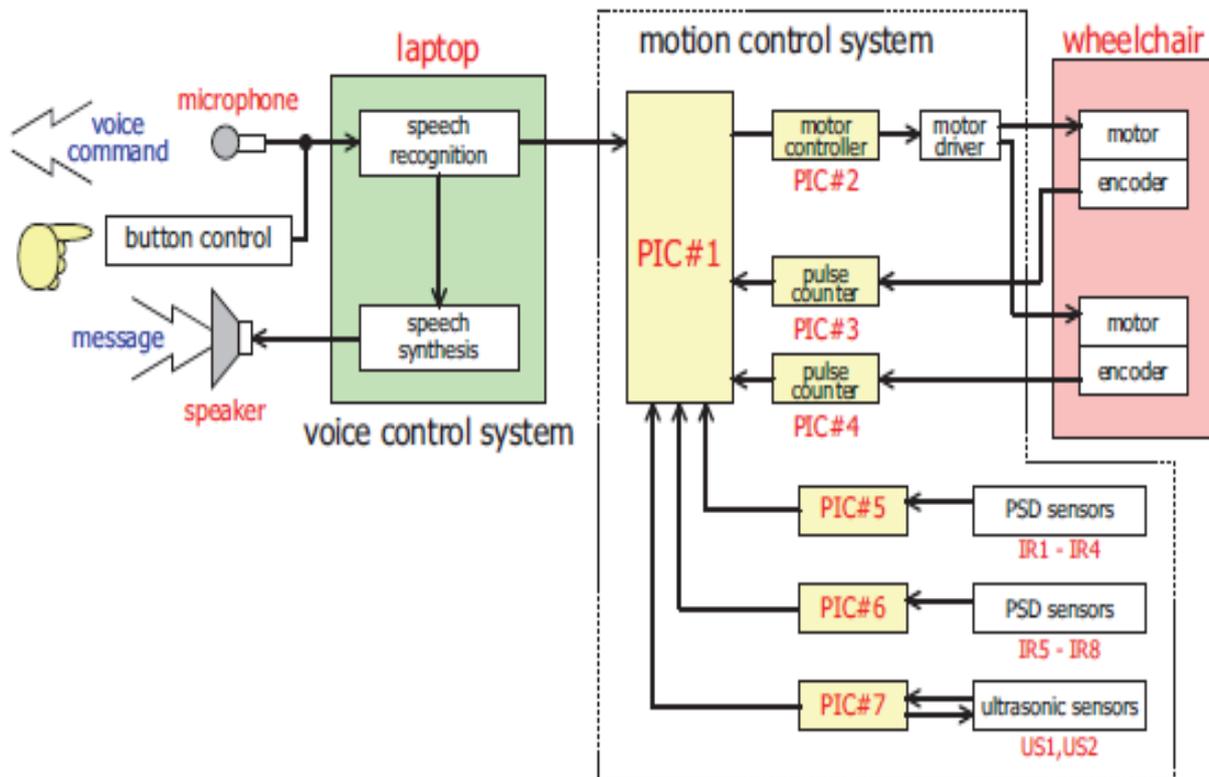


Figure 2-10: Diagram of functional voice-activated wheelchair [47]

QIDWAI et al. proposed the design of a speech-controlled wheelchair using a Fuzzy Inference System (FIS). The voice recognition implementation was conducted using MATLAB software. The process starts by picking up voice commands from the user and sending them continuously to the MATLAB program installed on a laptop. The program extracts the voice features and then checks with the fuzzy system. For instance, when a user says the Arabic word *Yameen* (i.e., right) the sound acquisition system receives it as the waveform shown in Figure 2-11. After extracting the exact word and removing the entire silent portion, the resulting waveform is then passed through the filter bank. The signal subsequently passes through all the filters in the filter bank, resulting in a specific value for each voice command between 0 and 1. The heights command value represent the recognition result of the FIS. The final output values are provided by appropriate motor control signals. The publication mentions that noisy environments affect system performance but give no details of the noise levels in their tests. It is recommended that commands are repeated at greater volume in noisy environments. The need for a laptop increases the cost of this system [50].

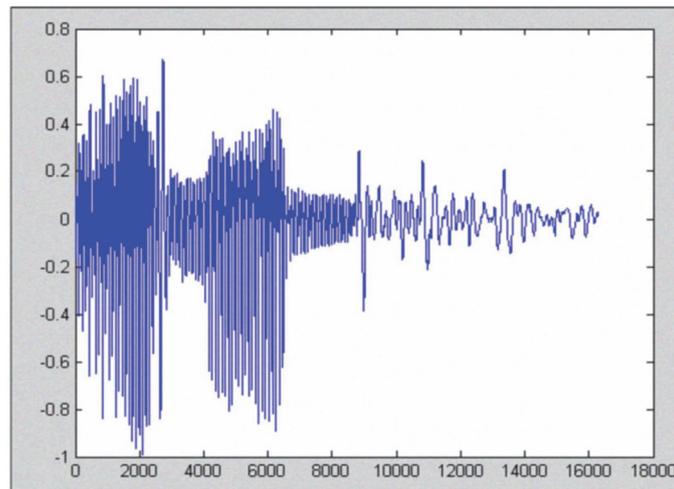


Figure 2-11: Sound Waveform [50]

JEON et al. proposed a voice and captured image robotic controller using a smartphone (Android OS and Apple OS). The phone represents the client, and the robot computer represents the server. Communication between server and client is achieved using Wi-Fi. The user's voice is recorded from a microphone using the natural language processing (NLP) tool of the client. Voices are captured and transmitted to the NLP server. The development kit of the natural speaking software Dragon (Nuance Communications Inc., Burlington, MA, USA) (SDK) was used to analyze the voices. The publication provides information about neither the success rate of the voice recognition software used nor any drawbacks or errors associated with the VR implementation. Figure 2-12 shows the schematic of the proposed system [51].

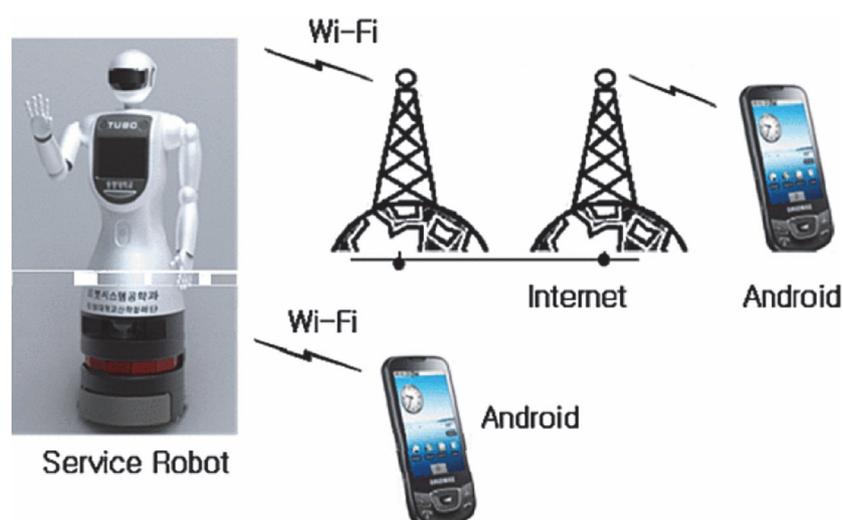


Figure 2-12: The robot communicates with the smartphones [51]

PHOOPHUANGPAIROJ et al. proposed a voice controlled robot based on the use of multiple Hidden Markov Model (HMM) algorithms. This system has three operational modes, which are gender-independent, gender-dependent and gender acoustic. Speech recognition is implemented using three stages. The voice command is first recorded in 16-bit pulse-code modulation (PCM) format at 11,025 samples per second using a close-talk microphone. In the second stage, the speech features are critical for accurate robot command recognition, and 39 acoustic dimension features consisting of 12 mathematical coefficients for sound modeling MFCCs with energy as well as 1st and 2nd order derivatives are used as speech features to recognize the robot's commands. In the final stage, the HMM acoustic models corresponding to commands are connected based on the pronunciation found in the pronunciation dictionary and the command grammar. This work is only HMM-based, and this can cause the system to produce false positive errors which are dangerous in medical and rehabilitation applications. The use of a laptop to implement the voice recognition also makes the system more costly. Figure 2-13 explains the system structure of the voice-controlled robots [52].

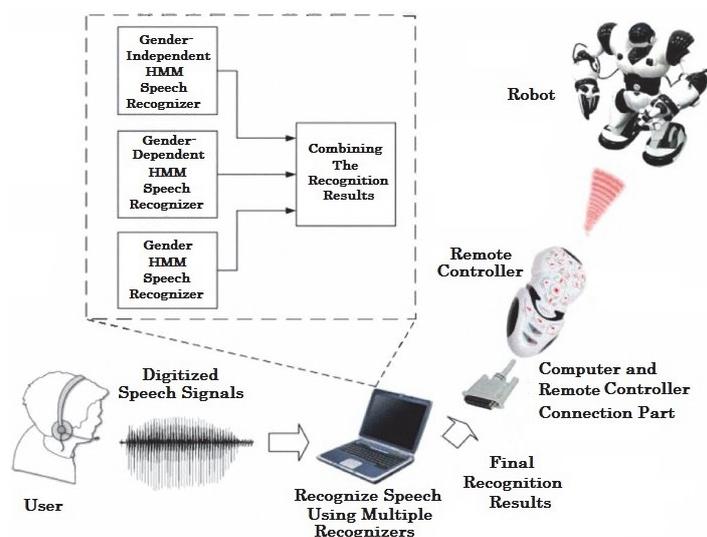


Figure 2-13: System structure of voice-controlled robots [52].

QIDWAI et al. proposed an Arabic voice controller for people with disabilities. The system design is based on the Easy VR voice recognition module [53]. The new system is an enhancement of a previously proposed project [50], realizing the control of the wheelchair without using a laptop to connect to the system. The authors mention that the new system still needs further improvements, such as

executing more user-based calibrations to make it easy to adapt to diverse users and environments. The proposed work is a prototype with a small motor and is not a real-time application. [54].

KATHIRVELAN et al. proposed a low-cost automatic wheelchair controlled by oral commands using a stand-alone control system. In this work, the voice recognition process is completed using embedded LabVIEW and a compact reconfigurable input/output FPGA Card-cRIO-9074 (National Instruments, Texas, USA) to dump the LabVIEW code. This makes the entire system a stand-alone application. The signal is picked up using a microphone set and then fed to an amplifier which supplies the amplified signal to the analog input of NI-9201(AI0) (National Instruments, Texas, USA). The analog signal is processed by LabVIEW, which analyzes the fast Fourier transform (FFT) spectrum of the input voice signal to find suitable ranges for the commands. Figure 2-14 shows the FFT of a sample signal. This system was not tested in different noisy environments and was not tested for medical applications [55].

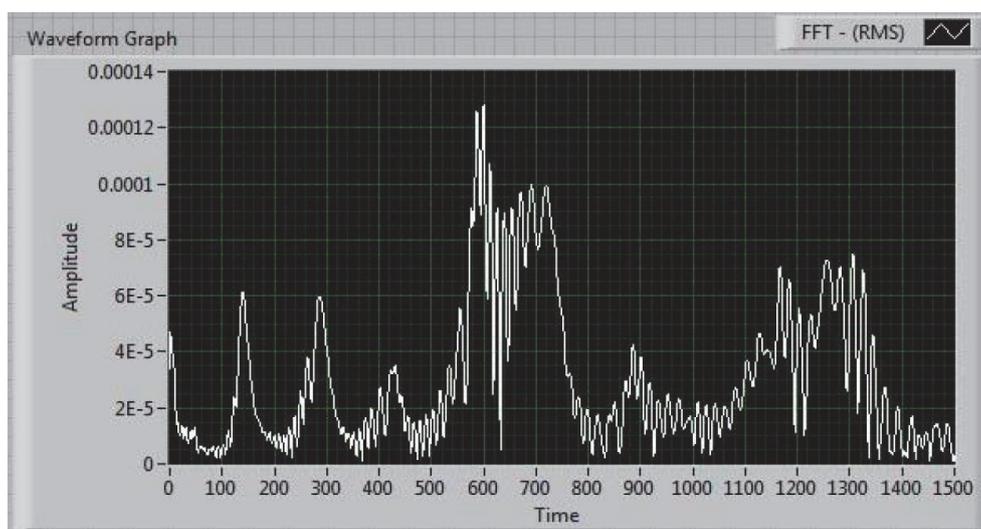


Figure 2-14: FFT for a sample signal [56]

HUANG et al. proposed a hybrid driving two-arm robot based on voice recognition. The voice recognition is achieved using the LD3320 module (Electrodragon Inc., China). The system uses an Atmega16L Microcontroller (Atmel Corporation, California, USA) to adapt the output of the LD3320 module with an original robot controller. The LD3320 module has an integrated speaker-

independent (SI) voice recognition algorithm. This SI mode of operation is not sufficient in rehabilitation applications because the system must obey the commands of the system user only, while SI systems respond to any users who give the same command. The performance of SI mode is profoundly affected by noise and has a recognition accuracy of $\leq 90\%$ which is lower than that of SD ($\leq 98\%$). The authors do not mention the recognition accuracy achieved and no information is given about the noise levels in the test environment.

Figure 2-15 shows a block diagram of the voice recognition unit [57].

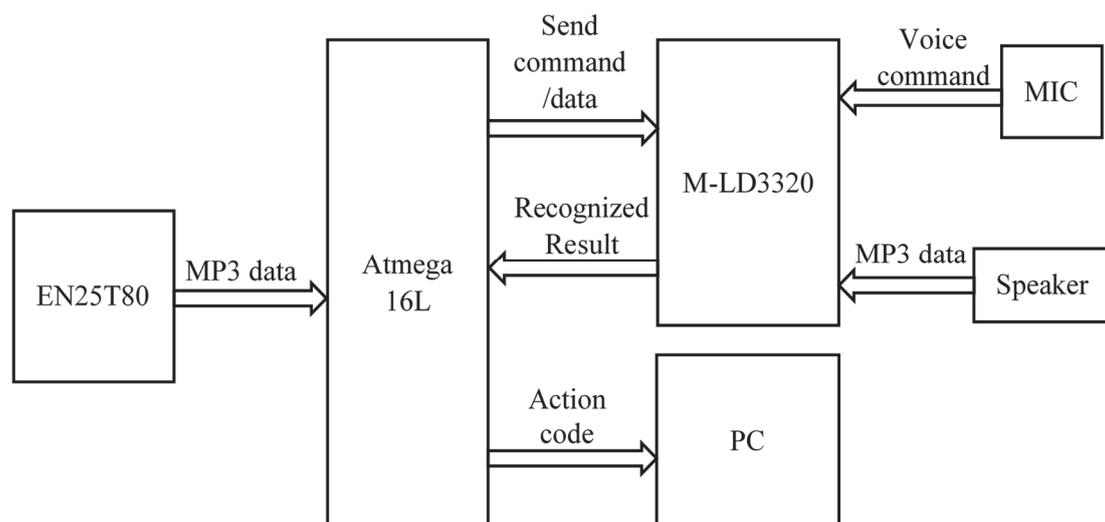


Figure 2-15: Voice recognition unit for a hybrid driving two-arm robot [57]

SHIM et al. have proposed an intelligent controller for mobile robots based on voice commands. This work concentrates on noise and reverberation cancellation by using an outlier-robust generalized side-lobe canceller (RGSC) and feature-space noise suppression. The system was designed to recognize voice commands from different users of different ages. This task has been implemented using two decoders depending on the age of the user (adult or child). Figure 2-16 describes the sub-sections of each module of the VRCS system. This work faced some drawbacks, such as the decreasing signal-to-noise ratio (SNR), and the distortion of acoustic input features which are usually unknown. The reliability of the acoustic input features could not be estimated. The authors used single channel approaches and multi-conditional acoustic model training. This system needs to be tested in noisy environments before being used for rehabilitation applications [58].

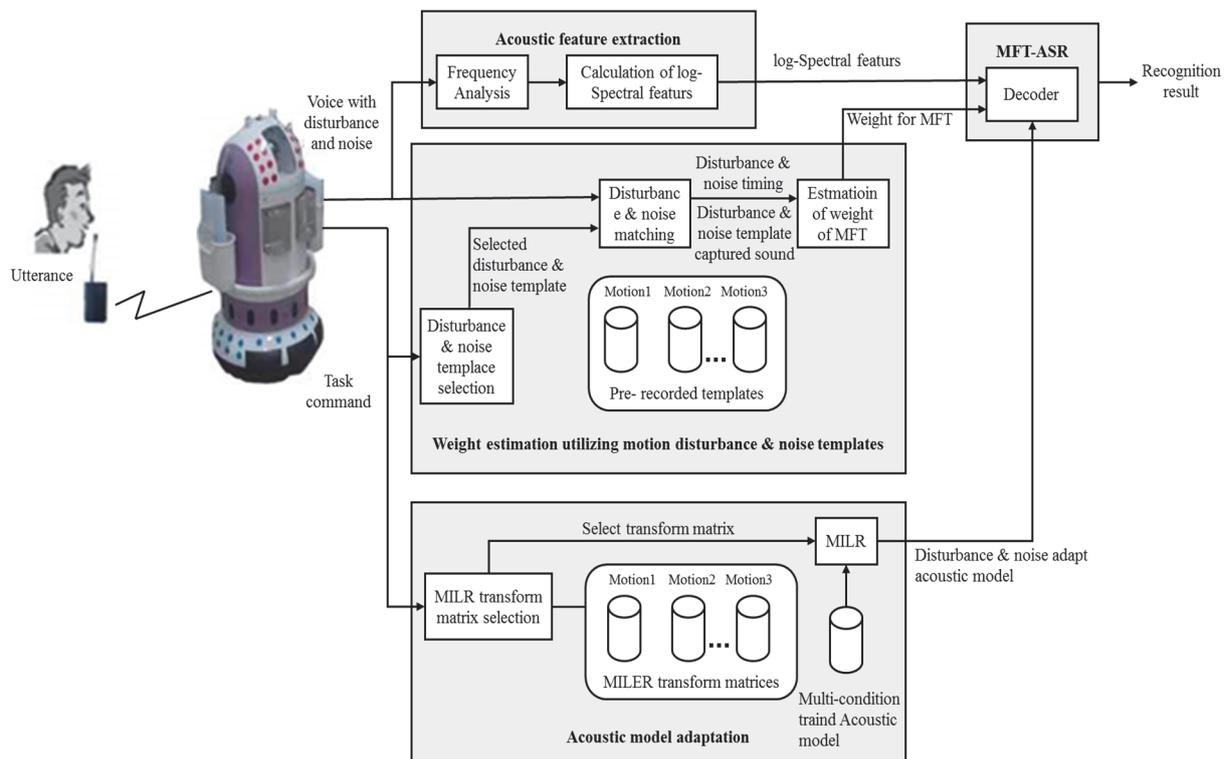


Figure 2-16: Structure of VRCS system [58]

SANTOS et al. proposed a command interface and driving strategy to control the movement and position of endoscope robotic arms. A block diagram of the system is shown in Figure 2-17. The voice recognition system implemented depends on the use of IBM SAPI (Speech Manager Application Programming Interface, IBM Corporation, New York, USA). It converts spoken words into a string of text. The text string is processed and converted to a command that controls the application. The process starts with the microphone port acquiring the input audio signal at 22 kHz, 16 bits. The resulting data is sent to an acoustic pre-processor, which optimizes the signal and sends it to a stochastic recognition algorithm. The SAPI has a library of word lists provided as arrays of text strings. This array can be activated or deactivated in the run-time. The recognition system compares the audio signal acquired with a set of features extracted from each word in the active vocabulary and then chooses the one that most likely matches the recorded sound. If there is no matching probability, the spoken command will fall under an adjustable rejection threshold, the system will ignore the command, and a false-negative exception is raised. The accuracy of this system needs to be improved and further testing is needed in different noise environments before using it for elderly or paralyzed users [59].

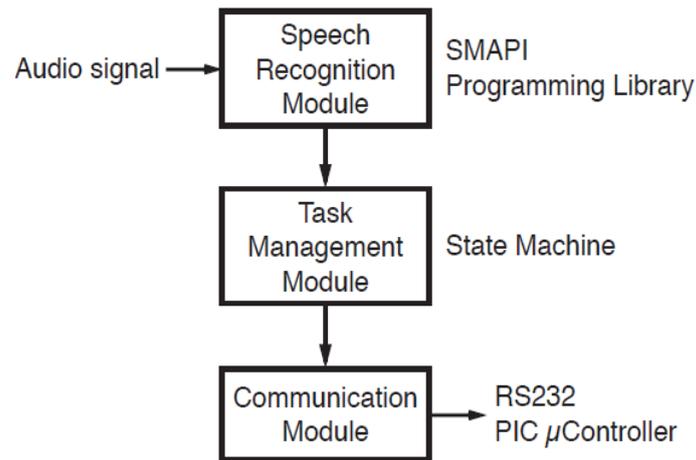


Figure 2-17: Sub-system block diagram [59]

There are many speech and voice recognition algorithms used in voice recognition software and hardware systems and techniques. The following discussion considers the operation of two models of VR using two different algorithms to implement voice recognition, which are dynamic time warping (DTW), and the hidden Markov model (HMM).

2.3.2 Voice Recognition Algorithms

2.3.2.1 Dynamic Time Warping (DTW) Algorithm

The Dynamic Time Warping Algorithm is a technique used to measure the similarity between two sequences which may vary in time and speed for the pattern. One of the two patterns is used as a reference for comparison (trained before) and the other is a real sample. DTW is used in time series analysis for classification and clustering tasks, and especially for isolated word recognition tasks and to calculate the optimal warping path between two time-series of different lengths [36]. The optimal path represents the minimum distance between the test and reference template.

Classical DTW Algorithm

The purpose of DTW is to compare the two time-independent sequences, let us say they are:

$$X = (x_1, x_2, \dots, x_N) \text{ of length } N \in \mathbb{N}$$

And

$Y = (y_1, y_2, \dots, y_M)$ of length $M \in \mathbb{N}$

The sequences could be discrete signals (time series) or featured signals sampled at a similar point in time. If F represents the feature space, then we get:

$$x_n, y_m \in F \text{ for } n \in [1: N] \text{ and } m \in [1: M]$$

The comparison of the two sequences is based on the comparison of two different features, x, y , where $x, y \in F$. A local cost measure, or local distance measure, is needed which is defined as the following function:

$$c: F \times F \rightarrow \mathbb{R}_{\geq 0}. \quad (2.1) [60]$$

where $c(x, y)$ is small (low cost) if x and y are similar to each other, or $c(x, y)$ is large (high cost). Calculating the local cost measure for each pair of elements of the sequences X and Y can be obtained from the cost matrix as:

$$C \in \mathbb{R}^{N \times M} \text{ defined by } C(n, m) = c(x_n, y_m)$$

Figure 2-18 shows the time alignment of two time-dependent sequences where the arrows represent the aligned points.

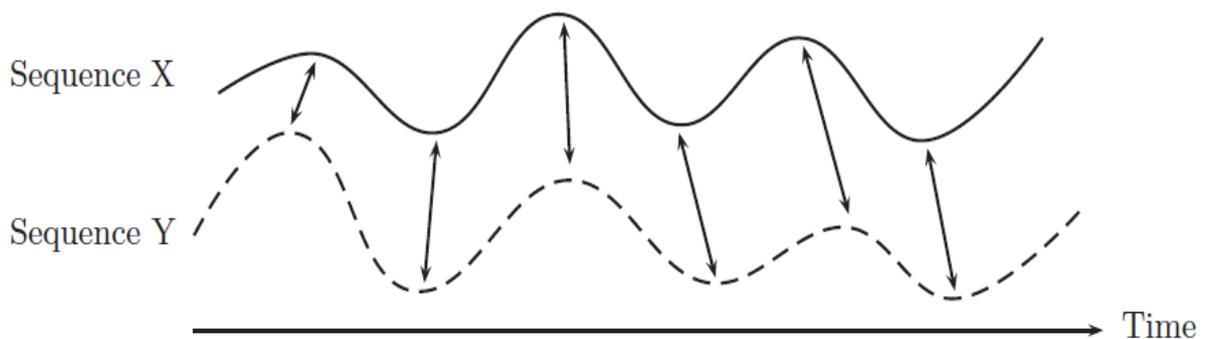


Figure 2-18: Time alignment of two time-dependent sequences [60]

Figure 2-19 shows the cost matrix of the two-time sequences. The first is the real-valued sequences, X , represented on the vertical axis and the second is the absolute value of the difference, Y , represented on the horizontal axis. The dark colors represent regions of low cost, and the light colors indicate regions of high cost [60]–[62].

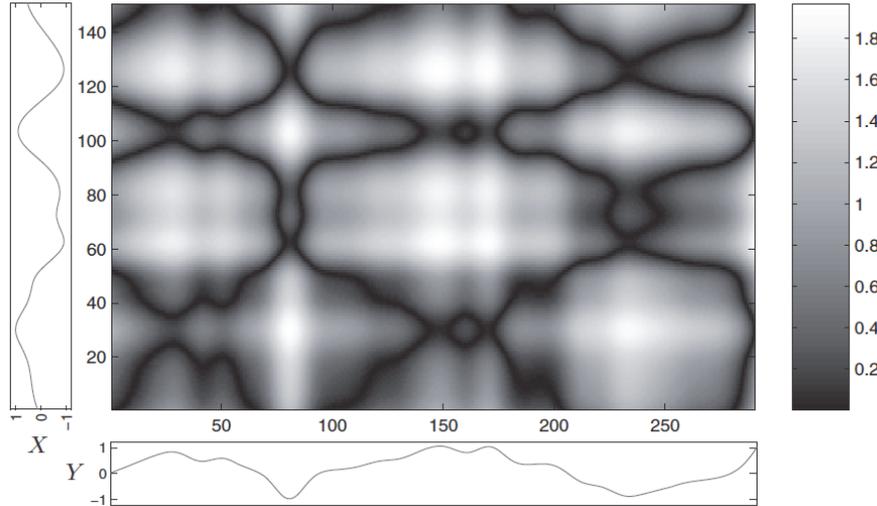


Figure 2-19: Cost matrix of the two-time sequences [60]

The warping path for (N, M) represented by a sequence $p = (p_1, \dots, p_L)$ where

$$p_\ell = (n_\ell, m_\ell) \in [1:N] \times [1:M] \text{ for } \ell \in [1:L]$$

Three conditions must be satisfied which are:

- (i) Boundary condition: where $p_1 = (1, 1)$ and $p_L = (N, M)$.
- (ii) Monotonicity condition: where $n_1 \leq n_2 \leq \dots \leq n_L$ and $m_1 \leq m_2 \leq \dots \leq m_L$.
- (iii) Step size condition: where $p_{\ell+1} - p_\ell \in \{(1, 0), (0, 1), (1, 1)\}$ for $\ell \in [1: L-1]$.

The step size condition (iii) implies the monotonicity condition(ii), which nevertheless has been quoted explicitly for the sake of clarity. The warping path $p = (p_1, \dots, p_L)$ for (N, M) is defined as an alignment between the two-time sequences $X = (x_1, x_2, \dots, x_N)$ and $Y = (y_1, y_2, \dots, y_M)$ by specifying the element x_{n_ℓ} of X to the element y_{m_ℓ} of Y . The boundary condition compels the first elements of sequence X and Y and also the last elements of sequences X and Y to be aligned with each other. The monotonicity condition shows the requirement of faithful timing: if an element in the X sequence precedes the second element, this should also hold for the corresponding elements in sequence Y and vice versa. There is a kind of continuity condition expressed by the step size condition which is that no omissions or replications in the alignment are acceptable for the elements of sequences X and Y . Figure 2-20 explains the three conditions where: (a) an admissible warping path satisfies conditions (i), (ii), and (iii); (b) the boundary

condition (i) is violated; (c) the monotonicity condition (ii) is violated; and (d) the step size condition (iii) is violated [60].

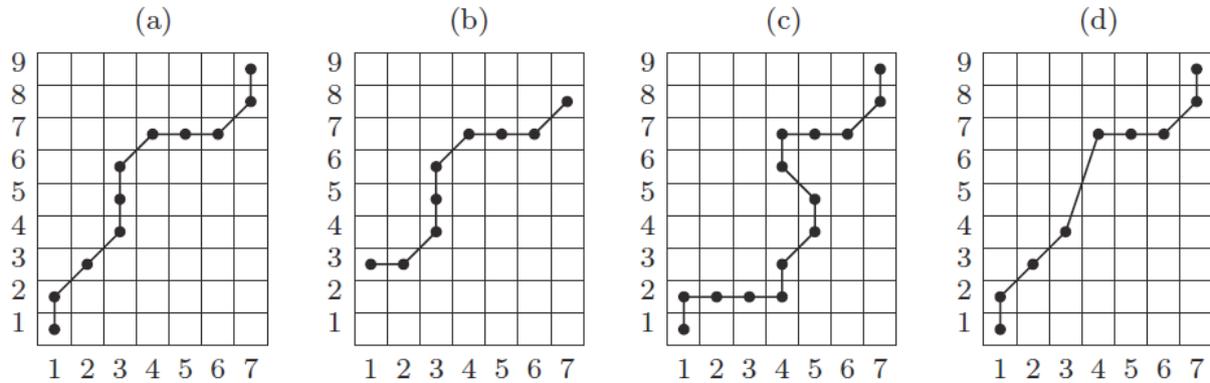


Figure 2-20: Paths of index pairs for X (N=9) and Y (M=7) [60]

The total cost of a warping path p $c_p(X, Y)$ between the two sequences X and Y with respect to the local cost c is defined as follows:

$$c_p(X, Y) = \sum_{\ell=1}^L c(x_{n\ell}, y_{m\ell}). \tag{2.2} [60]$$

The p^* represents the optimal warping path between X and Y and it has the minimal total cost among all possible warping paths. The dynamic time warping distance $DTW(X, Y)$ between X and Y is defined as the total cost of p^* :

$$TW(X, Y) = c_{p^*}(Y, X) \tag{2.3} [60]$$

$$= \min\{c_p(X, Y) \mid P \text{ is an } (N, M) - \text{warping path}\}$$

More details of DTW can be found elsewhere [63]–[68].

2.3.2.2 Hidden Markov Model (HMM)

The HMM is a powerful statistical tool for a discrete-time stochastic process model. It is a kind of signal processing technique that has been used successfully in speech and handwriting character recognition [69]. It is used for modeling generative sequences that can characterize an underlying process generating observable sequences. HMMs appear in a broad range of applications. They can be

used in many fields of interest in signal processing, and in particular speech processing. They have also applied with success in low-level natural language processing (NLP) tasks and functions such as speech tagging, phrase chunking, and extracting target information from documents [70].

The name of this algorithm was given by Andrei Markov to the initial version of HMM in the early twentieth century [71], before Baum and his colleagues developed the theory of the HMM in the 1960s [72].

Figure 2-21 shows an example of a Markov process. The presented model describes a simple sample for the stock market index. The model has three main states, Bull, Bear and Even, and three index observations, which are up, down and unchanged. The model represents a finite state automaton, with probabilistic transitions between states. It gives a sequence of observations like up-down-down where it can easily be verified that the state sequence that produces the observation is Bull-Bear-Bear. The probability of the sequence represented by the product of the transitions, in this case, is $0.2 \times 0.3 \times 0.3$ [70].

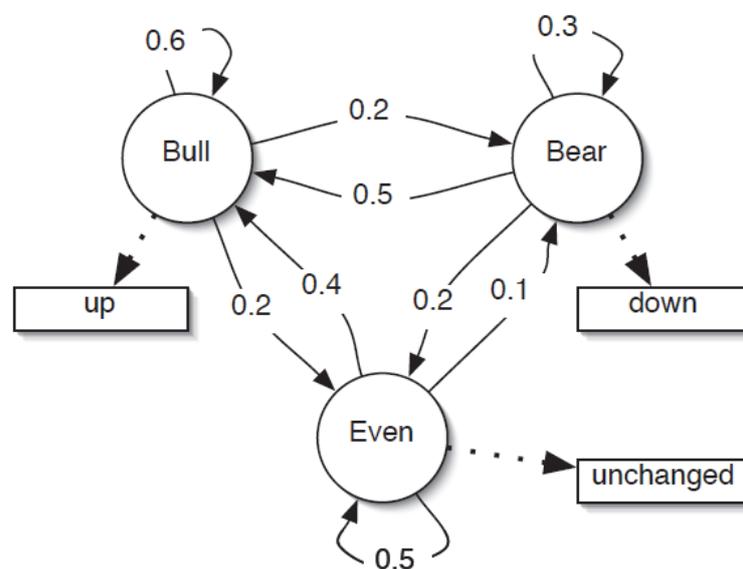


Figure 2-21: Example of the Markov process [70]

Figure 2-22 shows a case of the extension of the Markov process to a hidden Markov model. The new HMM allows all the observation symbols to be emitted in each state with a finite probability. This improvement makes the model more expressive and better represent intuition; in this case, that the bull market would

have both good and bad days, but there would be more good ones. The key difference is that from the observation sequence up-down-down it cannot be said exactly which state sequence produced this observation, and so the state sequence is 'hidden.' The probability of the model sequence production can be calculated as well as which state sequence is the most likely to have produced the observations [70].

The formal definition of an HMM is as follows:

$$\lambda = (A, B, \pi) \quad (2.4) [70]$$

S represents the state alphabet set, and V is the alphabet observation set:

$$S = (s_1, s_2, \dots, s_N) \quad (2.5) [70]$$

$$V = (v_1, v_2, \dots, v_M) \quad (2.6) [70]$$

Q is a fixed state sequence of length T, and corresponding observations O:

$$Q = q_1, q_2, \dots, q_T \quad (2.7) [70]$$

$$O = o_1, o_2, \dots, o_T \quad (2.8) [70]$$

A represents the transition array, where when storing the probability of state j following state i, the state transition probabilities are independent of time:

$$A = [a_{ij}], a_{ij} = P(q_t = s_j | q_{t-1} = s_i) \quad (2.9) [70]$$

B represents the observation array, storing the probability of observation k produced from the state j, independent of t:

$$B = [b_i(k)], b_i(k) = P(x_t = v_k | q_t = s_i) \quad (2.10) [70]$$

π is the initial probability array:

$$\pi = [\pi_i], \pi_i = P(q_1 = s_i) \quad (2.11) [70]$$

The model sets two assumptions. The first is the Markov assumption, which supposes that the current state is only dependent on the previous state. It represents the model memory:

$$P(q_t | q_1^{t-1}) = P(q_t | q_{t-1}) \quad (2.12) [70]$$

The output observation at time t which represents the independence assumption states is dependent only on the current state. It is fully independent of previous observation states and is given as:

$$P(o_t|o_1^{t-1}, q_1^t) = P(o_t|q_t) \tag{2.13} [70]$$

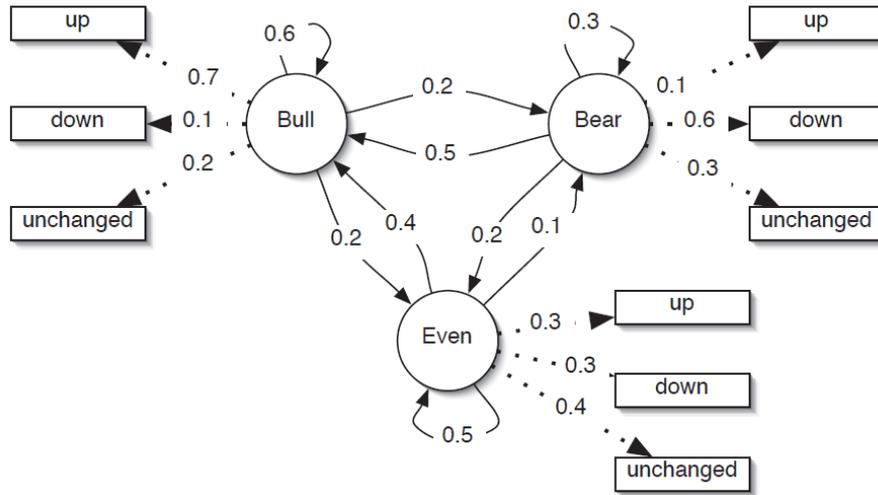


Figure 2-22: An example of a hidden Markov model [70]

The calculation of the probability of the observation sequence $P(O|\lambda)$ for the given sequence of observation will view the evaluation of how the model predicts a given observation sequence and allow us to choose the appropriate model from a set.

The probability of the observations O for a specific state sequence Q is:

$$P(O|Q, \lambda) = \prod_{t=1}^T P(o_t|q_t, \lambda) = b_{q_1}(o_1) \times b_{q_2}(o_2) \dots b_{q_T}(o_T) \tag{2.14} [70]$$

The probability of the state sequence is given by:

$$P(Q|\lambda) = \pi_{q_1} a_{q_1 q_2} a_{q_2 q_3} \dots a_{q_{T-1} q_T} \tag{2.15} [70]$$

So, the probability of the observation can be calculated as follows:

$$\begin{aligned} P(O|\lambda) &= \sum_Q P(Q|Q, \lambda) P(Q|\lambda) \\ &= \sum_{q_1 \dots q_T} \pi_{q_1} b_{q_1}(o_1) a_{q_1 q_2} b_{q_2}(o_2) \dots a_{q_{T-1} q_T} b_{q_T}(o_T) \end{aligned} \tag{2.16} [70]$$

The probability of O can be used to evaluate the results, but for a direct evaluation will be exponential in T .

There are many redundant calculations which would be made by direct evaluation (equation 2.14). The complexity can be reduced by using the caching of calculations. The cache can be implemented as a trellis of states at each time step. The cache value (α) has been calculated for every single state as a sum of calculations over all states of the previous time step. α represents the probability of the particular observation sequence $o_1, o_2 \dots o_t$ and state s_i at time t . Figure 2-23 shows the trellis algorithm. The forward probability variable is defined as:

$$\alpha_t(i) = P(o_1 o_2 \dots o_t, q_t = s_i | \lambda) \quad (2.17) [70]$$

The filling of the trellis in the value of α makes the final column of the trellis equal to the probability of the observation sequence. The forward algorithm used to implement this process operates as follows:

1. Initialization:

$$\alpha_1(i) = \pi_i b_i(o_1), 1 \leq i \leq N. \quad (2.18) [70]$$

2. Induction:

$$\alpha_{t+1}(j) = [\sum_{i=1}^N \alpha_t(i) a_{ij}] b_j(o_{t+1}), 1 \leq t \leq T-1, 1 \leq j \leq N. \quad (2.19) [70]$$

3. Termination:

$$P(O|\lambda) = \sum_{i=1}^N \alpha_T(i). \quad (2.20) [70]$$

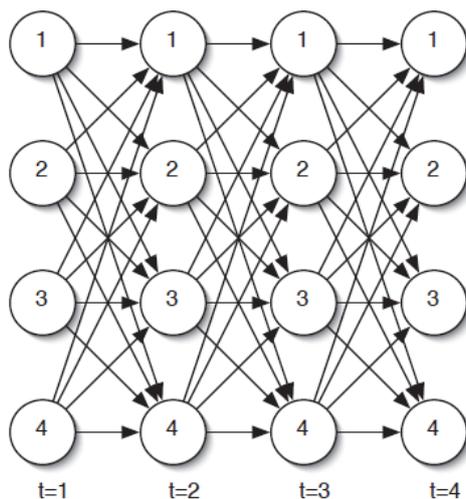


Figure 2-23: The trellis algorithm [70]

The induction step represents the key of the forward algorithm. Each state s_j , $\alpha_j(t)$ stores the probability of arriving in a state that observed the observation sequence in time t . Figure 2-24 shows the induction step of the forward algorithm.

The forward algorithm reduces the complexity of calculation by caching α values involved to N^2T rather than $2TN^T$. The analogous backward algorithm is the exact reverse of the forward algorithm with the backward variable:

$$\beta_t(i) = P(o_{t+1} o_{t+2} \dots o_T | q_t = s_i, \lambda) \quad (2.21) [70]$$

where the probability of the partial observation sequence is from $t + 1$ to T , starting in state s_i . More details about the HMM can be found elsewhere [73]–[75].

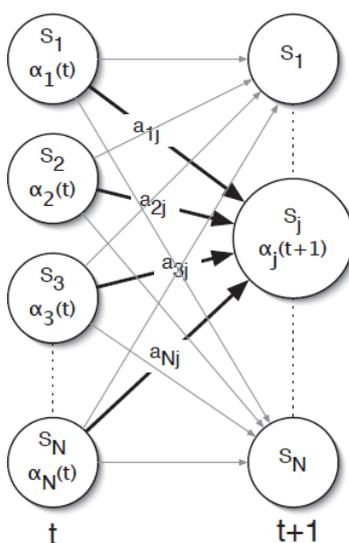


Figure 2-24: Induction step of the forward algorithm [70]

2.3.2.3 Sensory Text to Sound Independent (T2SI™)

The text-to-sound-independent algorithm T2SI™ (Sensory Inc. USA), is a powerful phonemic recognition algorithm using a combination of both a neural network (NN) and HMM [76]. The T2SI algorithm converts a voice command from any user (SI) into text and then compares this text with the texts in a library of voice commands which is stored in the voice controller before the recognition process. The user writes the required command as a text, and the algorithm converts the voice command into a text to accomplish recognition. This process will allow any user to use the voice controller. Figure 2-25 shows the structure of an SI voice recognition system.

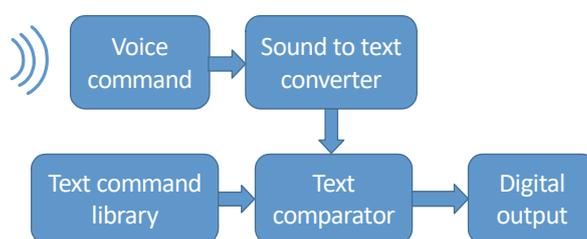


Figure 2-25: SI system structure

2.3.3 Voice Recognition Errors

Through the voice recognition process, different types of errors can occur due to various reasons such as noise and command pronunciation interference. Two types of voice recognition error are taken into consideration in the current work. The first is false positive (FP) errors where the voice recognition system misrecognizes the given command such as when the user gives the command “Forward” and the VR result of the system is “left”. This means that the system has recognized the voice command but with the wrong meaning, and it could result in the incorrect implementation of a command. This may be very dangerous and harmful in medical and rehabilitation applications. FP errors must be reduced as far as possible.

The second type of voice recognition error is false negative (FN) errors. An FN error means that the voice recognition system cannot recognize the voice command. This type of error is less harmful to the control system and resulting problems can be solved by repeating the required commands. The use of voice recognition systems in control application requires a safety condition to avoid dangerous situations if any VR errors occur, such as an obstacle detection and avoidance system in voice-activated wheelchair and rehabilitation robotics.

2.4 Electromyogram (EMG)

One important human body function is the electrical activity in the muscle, which can be recorded as an electromyogram (EMG). The EMG represents a technology used for evaluating and measuring the electrical activity and electrical potential generated by muscle cells. To use this technique the patient has to be equipped with special types of electrodes (surface electrode, needle electrode) to pick up the electrical activity in the muscle at the surface of the skin covering the muscle area. The origin order that makes the muscle generate this electrical activity comes from the brain. When anybody wants to move any part of his or her body, a nervous impulse is generated in the brain and transmitted by the nervous system to the target muscle. Even though some patients have spinal cord injury or quadriplegia they are still able to control this bio-signal in some regions of the face, neck and shoulder muscles. Handicapped and quadriplegic patients can use this technique as a solution for the control of wheelchair and rehabilitation applications. More details about this technique and its specification can be found in [77]–[86]. Many researchers are working in this field.

MOON et al. proposed a robotic wheelchair controlled by active potential (EMG) signals generated by the levator scapulae muscle (LSM) in the contraction of the right and left shoulder muscles. Figure 2-26 shows the levator scapulae muscles. The EMG signals are picked up using wet-type surface electrodes, which are then amplified by 60 dB. After that, the signals pass through a band-pass filter with 50 to 500Hz bandwidth to remove some of the noise and interference. The EMG signals are sampled at a rate of 2 kHz and a notch filter is used to reduce the 60 Hz noise effect. The preprocessing is implemented using the MP150 EMG system (BIOPAC System Inc., California, USA) [54]. Then the output signals are sent to the computer for classification and reprocessing using LabVIEW software. The core processing implemented depends on the calculation of the mean absolute value (MAV) of the EMG signals. This work faces some drawbacks such as in the discerning the user's intention from unintentional behavior. This problem causes a delay in recognition and hence in the system's response. The user needs to have a proper training to control the LSM muscles and also the system still needs to be combined with a laptop to control the wheelchair which adds higher cost [87].

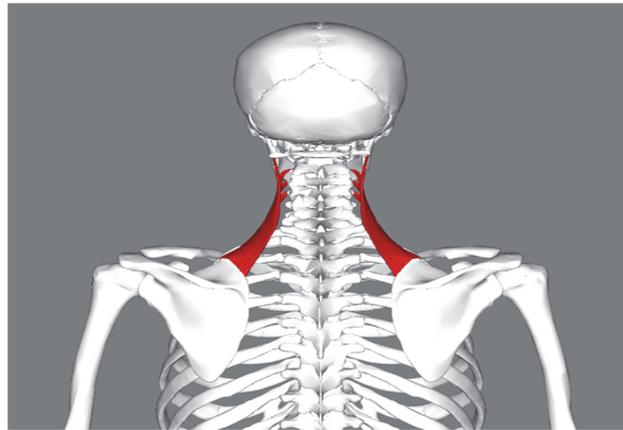


Figure 2-26: The Levator Scapulae Muscles (LSM) [88]

TSUI et al. proposed hands-free control using forehead EMG and EOG signals. The aim of this work was to assist elderly and physically challenged people to control the movement of an electrical wheelchair. The CyberLink signal acquisition device (Brain Actuated Technologies, Inc., Ohio, USA) was used for signal acquisition and processing, which is a small wearable device that acquires bio-signals from three electrode sensors on a headband. Figure 2-27 shows the CyberLink device [89].



Figure 2-27: CyberLink device [90]

This device can read three human biosignals, the EEG, EOG, and EMG, from the forehead surface. After the data acquisition phase is completed, the information is processed by a PC software suite called Brainfingers (Brain Actuated Technologies, Inc., Ohio, USA), which implements various signal processing techniques such as active segment detection, feature extraction, pattern recognition, and logic control. In this work, the EMG signals from the

forehead muscle (via B11 Brainfingers software, see a) Relax state
b) Active state

Figure 2-28 is used for motion command control while the EOG signal is used only for speed control. The system uses a two-second time period to generate the control command. For example, if the system moves in any direction and only one EMG signal (EMG-click) is presented and detected in the 2-sec time period, the system will generate a stop command. If a second EMG-click appear after the stop command, the system will generate a forward command. If there are two EMG-click after the stop command, the system will generate a left command and so on. All the EMG-click must have a value higher than the threshold level of the Brainfingers software. Users need prior training with the system before they can use it. They must implement the proper control of their forehead muscles. For elderly patients, this method is not the easiest solution. Also, this method cannot be used in indoor environments because it cannot give flexible motion control and, for example, a curved trajectory of motion is not available.

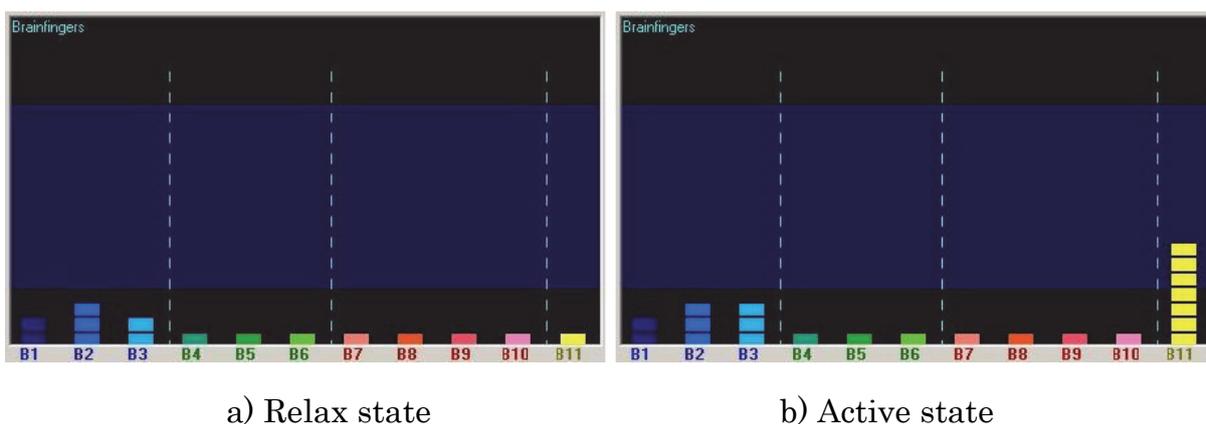


Figure 2-28: EMG (B11) in relax state and active state [90]

Yi et al. present similar work [90]. Their new method uses the back-propagation artificial neural network (BPANN) algorithm to classify the measured EMG signal, and the training of users is also needed for them to give specific muscle contractions for each of the main five commands. Figure 2-29 shows a schematic diagram of the BPANN classification. This method has a problem of false recognition because it is profoundly affected by noise in the signals coming from facial muscles. The system has the same problem of not allowing curved motion, which is very important in narrow environments. The BPANN algorithm needs more improvement, and thus this system is not ready for clinical and real-time applications.

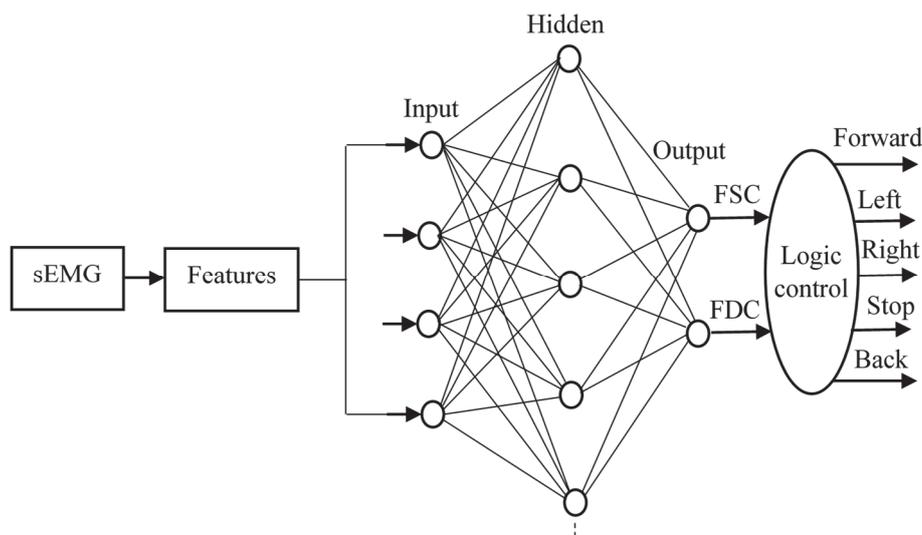


Figure 2-29: Schematic diagram of BPANN classification [91]

HASHIMOTO et al. proposed a third control method using Cyberlink which depends on the following three gestures:

- Wrinkled forehead detected from EOG
- Look left/right detected from EOG
- Jaw closure detected from EMG

The control algorithm uses EOG and EMG signals. The gesture of closing the jaw/single click (G1) moves the wheelchair forward, closing the jaw/double click (G2) moves it backward, and wrinkling the forehead (G3) stops it. The eye motions of looking towards the right and left (G4 and G5) turn the wheelchair in a pivot turn. This method has some drawbacks, such as the existence of different recognition thresholds for each user, and it is strongly affected by signal noise and interference from the facial muscles [90], [92]–[94].

OHISHI et al. proposed an EMG-based wheelchair system which uses surface EMG signals from the tibialis anterior muscles in both lower extremities. The signals are picked up and amplified by active surface electrodes (NM-512G, NIHON KODEN), a multi-telemeter system (WEB-5000, NIHON KODEN) and a biomedical amplifier. The measured EMG data are sent to a personal computer (XW4400/CT, HP) via a 16-bit analog-to-digital converter board (LPC-321416, Interface Inc.). The system determines a control command for every control cycle (100 ms). This system is designed for users who can control their muscles but cannot walk. It depends on the EMG signal being picked up from muscles in the

feet, and so this method is not suitable for a quadriplegic, or paralyzed patients or those with upper spinal cord injury. In addition, the use of this method over a long period causes painful muscle fatigue [95]–[97].

RIVERA et al. proposed a power wheelchair controlled using a single surface EMG (sEMG) sensor. In this method, only two features are extracted from a single EMG signal and used for command signal classification. One of these features is based on the independent component analysis (ICA), which is a traditional blind source signal separation method [98], [99]. and is a powerful technique for EMG signal separation [100]. It is assumed that an EMG sensor captures a combination of statistically independent motor unit action potential trains (MUAPT) due to cross-talk [101]. Each MUAPT is a sum of many activities in the muscle. This study aimed to separate the entire MUAPT originating from a single muscle. To apply ICA, an accurate sensor is placed nearby to the muscle that generates the MUAPT signals. The sensor must capture each EMG signal. This method needs further enhancement before it can be used in real-time applications. It requires a higher classification rate than those already in use and needs a number of gestures to cover the commands for the basic movements required to control a wheelchair. Figure 2-30 shows the framework of the proposed classification system [102].

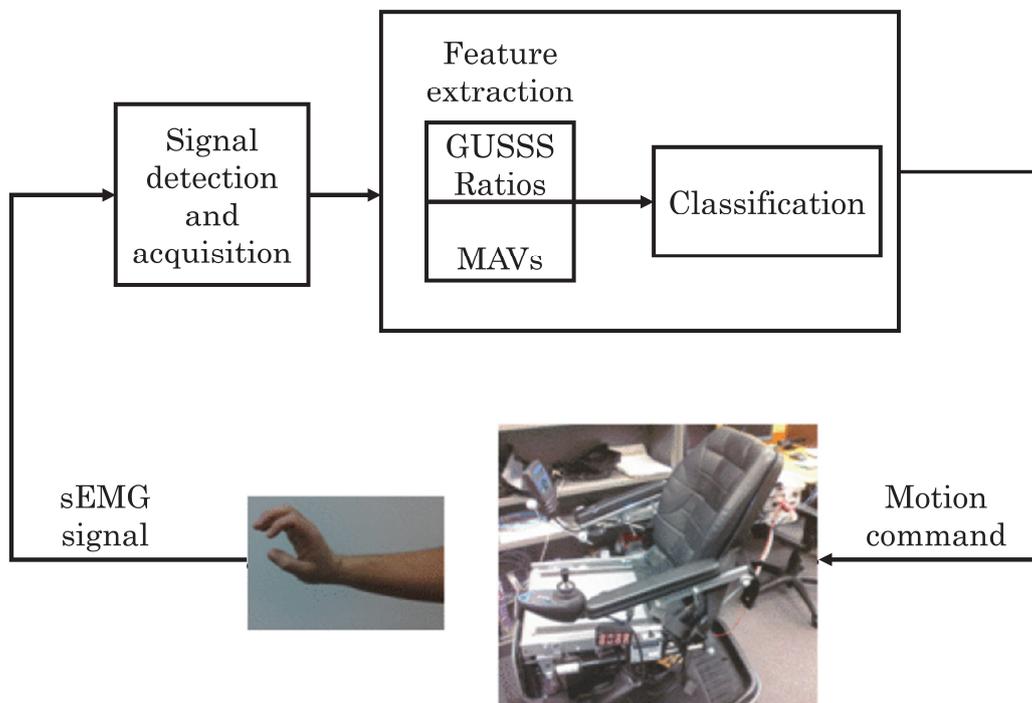


Figure 2-30: Framework of the proposed classification system [102]

2.5 Electroencephalogram (EEG)

The electroencephalogram represents the electrical activity of the brain cells. When a person uses the brain, it generates several electrical signals with frequencies that differ in different areas of the brain. The value and frequency of these electrical signals differ depending on the used thinking style. Each of these signals represents a particular kind of brain activity and emotion, such as thinking, sleeping, hungry, or angry. This electrical activity can be picked up and recorded using a special technology called electroencephalography (EEG). The EEG instrument interfaces with the user using special types of EEG electrodes arranged in a specific position to gain appropriate measurements of brain activity. This arrangement of electrodes is called the 10-20 system, which is an internationally recognized method for EEG electrodes positioned on the brain scalp. The name of this system refers to the relationship between electrode positions which cover 10% to 20% of the total brain cortex in the front, left, right and rear sides [103]. In recent years, many researchers have been working in the field of the use of brain signals as input signals for smart systems so as to interpret the meaning of each brain signal. One important research area is the use of these signals as control signals for wheelchairs and rehabilitation applications for particular kinds of patients. This procedure includes the classification and processing of brain signals using a computer system. This technique is called brain-computer interface (BCI). The main idea in this field is to pick up Beta, Alfa, Theta, and Delta brain signals using electroencephalography (EEG) or any other technology and for a computer to classify the signals by understanding user commands based on brain signals. More details about this technology are available elsewhere [104]–[111].

TANAKA et al. proposed an intelligent wheelchair controlled by EEG brain signals. The system picks up EEG signals from the scalp using surface electrodes connected to an electrode box. The EEG electrodes are distributed according to the 10-20 international electrode system [112]. A band-pass filter is used to reduce the interference of the EMG and EOG signals. The EEG signals are amplified and then digitalized using an analog-to-digital converter (ADC). The final signal is processed by a PC to check the template matching between the EEG signal for training and a direct EEG signal for a user control command. This system was designed for testing only two commands, which are left and right. The results show that the system is still affected by EMG and EOG interference even after using a band-pass filter to reduce it. Users of the system were asked to reduce their body

motions and to fix their eyes to enhance performance. The results show that this system is not suitable for real-time applications. Figure 2-31 shows the structure of the system [113].

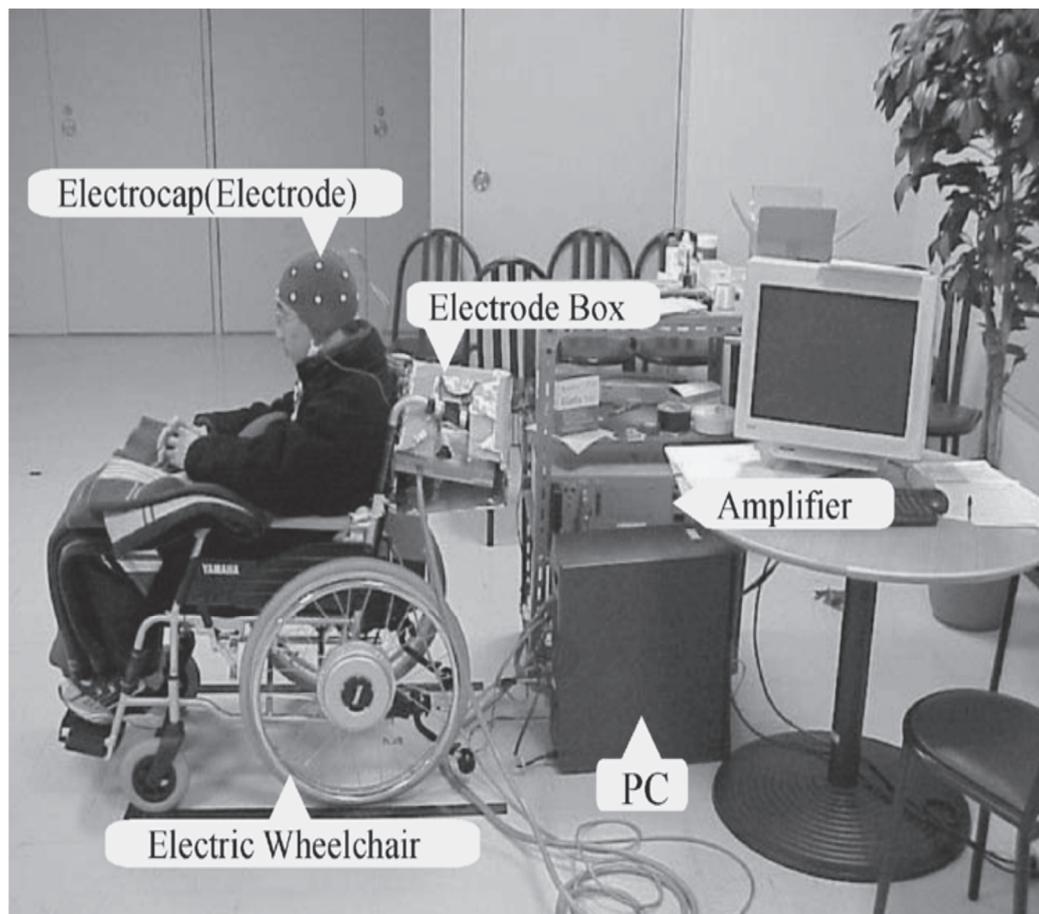


Figure 2-31: Structure of the system [113]

ITURRATE et al. proposed a brain-actuated wheelchair. The system uses an EEG signal to feed an asynchronous P300 brain-computer interface unit [114]. The P300 is integrated with an autonomous navigation system to guide the wheelchair in familiar environments. The P300 BCI system executes the acquisition, processing, and classification of brain signals. It is connected to a personal computer which runs the P300 system software to process the acquired signals. The system configuration is shown in Figure 2-32 [115]. This system needs previous training for users with a BCI unit to get accurate results. One of the harmful drawbacks is that the system is strongly affected when the user loses concentration because the control commands generated depend on the user's thoughts. Therefore, it is not suitable for use in real-time applications.

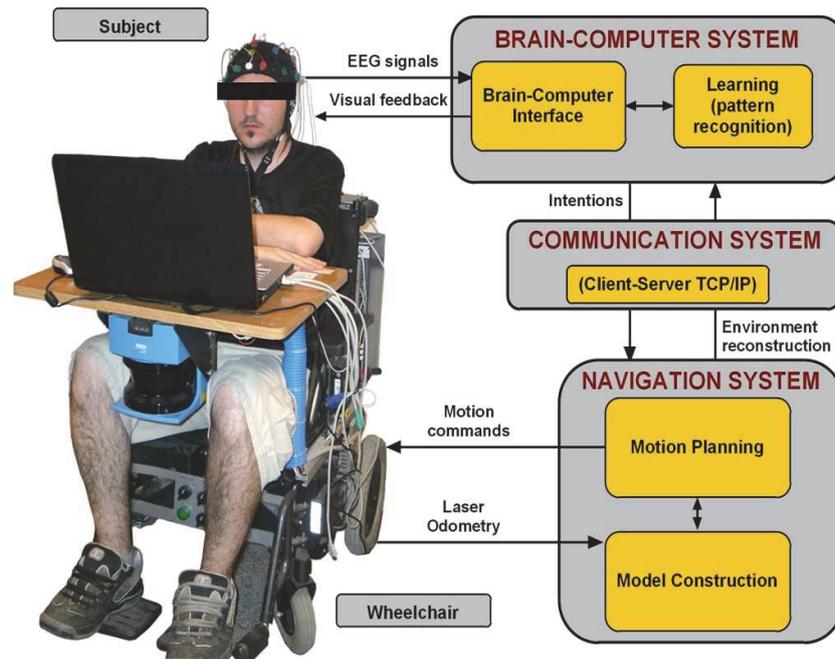


Figure 2-32: BCI system configuration [115]

A new EEG device, the Emotiv EPOC Headset, was developed by Emotiv Systems (Sydney, Australia). The EPOC headset looks similar to a bicycle helmet and consists of 16 EEG sensors. Data are recorded through 14 channels for active electrodes plus two channels for reference electrodes. The electrodes in the headset are positioned according to the 10-20 EEG system. The device works with a sampling rate of 2,048 Hz and includes a low-pass filter with a cut-off frequency of 86 Hz and a high-pass filter with cut-off frequency of 0.16 Hz. Most useful electrical activity in the brain can be picked up within these two ranges. Fifty Hz and 60 Hz notch filters are also included in the headset to reduce interference from the power source. Figure 2-33 shows the EPOC headset [116], [117].



Figure 2-33: EPOC headset [118]

Many researchers have used this technology in wheelchair and robot system control. WIJAYASEKARA et al. proposed the use of the EPOC headset to control the movement of a Lego NXT prototype robot. The control signal is picked up from 6 trained users for 30 minutes. Each user is trained using three types of headset management software, which are the Emotiv Control Panel, the EmoKey application, and the Cyton Control application. A graphical user interface (GUI) has been designed and implemented using the C++ environment. It combines the processed signals from the Emotiv SDK and the output control signals are sent to the mobile robot to interface the user's signals with the robot. This system requires further improvements for the detection, acquisition, and classification of brain signals [119]. Figure 2-34 shows a block diagram of the system, and similar work has been proposed in other studies [120]–[123].

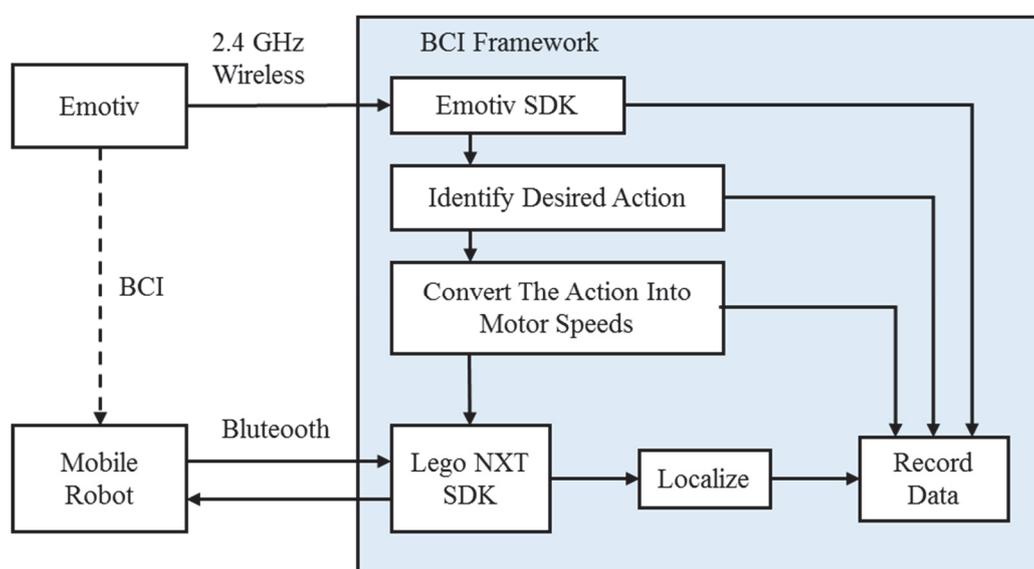


Figure 2-34: System block diagram [119]

The use of EEG signals to control wheelchairs and mobile robots is still in the development and testing phase and is not yet applicable to traditional control systems and for commercial products. One of the drawbacks of using EEG signals is the small electrical values of brain signals which are profoundly affected by noise and interference from other body signals and are also affected by peripheral noise in the surrounding environment. These drawbacks prevent this method from being very useful in control systems. The second problem is the difficulty of training some types of wheelchair users when they have a severe disability such as brain stroke or paralysis. More details about EEG signals and their drawbacks can be found elsewhere [124]–[135].

2.6 Motion and Gesture Recognition

In another field of research work, a different control method is considered for the control of wheelchairs and rehabilitation robots. This method depends on the recognition of the motions and gestures of the patient's body. Recognition can be accomplished using different types of cameras, sensors, and image processing algorithms. The success of these methods depends on the ability of the patient to control the motion of the body and limbs. Quadriplegia patients can use head movements to control the wheelchair using different procedures and sensors. Infrared and ultrasonic sensors have been used in some studies to detect head movements and use them in controlling wheelchairs [136]–[138]. Other researchers use cameras to detect head gestures for wheelchair control [139]. Recently, many researchers have proposed the use of accelerometers and orientation sensors in controllers for wheelchair and rehabilitation applications.

AKMELIAWATI et al. proposed a hand-glove control system for a wheelchair. Users wear an instrumented glove fitted with flex or bend sensors to control the movement and direction of the wheelchair. The system uses uni-directional wireless communication between the gloves and the system controller which is interfaced between the seat and the system wheels. When the user bends a finger in the glove, a small voltage develops in the bending sensor. The sensor voltage is transferred to the control board which processes the signal and sends it to the motor drivers to control the motion of the wheelchair. Figure 2-35 shows the structure of the hand-glove control system. This method is useful for rehabilitation wheelchairs when the patients can move and control their fingers. This method cannot be applied to quadriplegia and paralyzed patients [140].

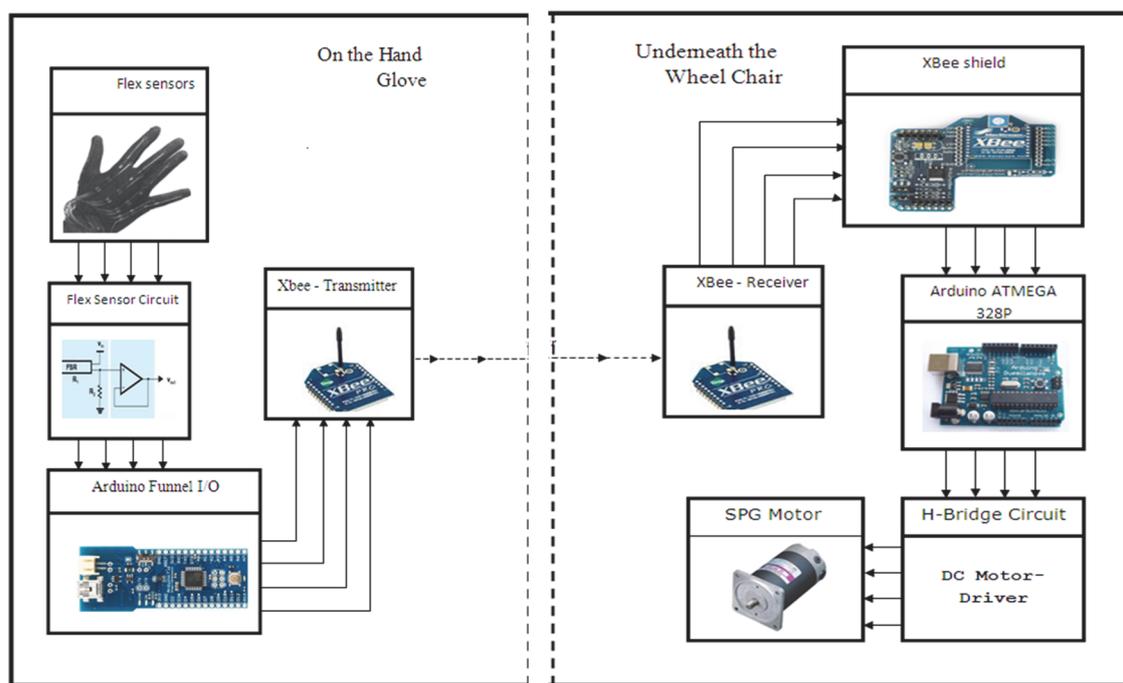


Figure 2-35: System structure of the hand-glove control system [140]

HU et al. proposed a head gesture recognition system for the control of an intelligent wheelchair. The system is based on tracking head motions using a PC, camera and image processing algorithms. The system compares the location of the lips with a fixed rectangular window. Head gesture commands are determined by the algorithm and include head-up, head-down, turn-left and turn-right. The movements of the intelligent wheelchair such as turning left or right and going forward or backward are determined correspondingly. Experiments showed that this approach can achieve the purpose of controlling the intelligent wheelchair by head gestures. Lips are detected using the Adaboost algorithm [141]. Head gestures determined according to the positions of the lips control the movement states of the intelligent wheelchair. Figure 2-36 shows the gesture recognition system. The system tests results show that the use of head gestures to control the intelligent wheelchair movements has good robustness. Due to the complexities of face patterns, performance in conditions of shelter occlusion, illumination, imaging conditions and other factors need to be further improved. Further details with a similar method can be found in subsequent publications [142]–[146].

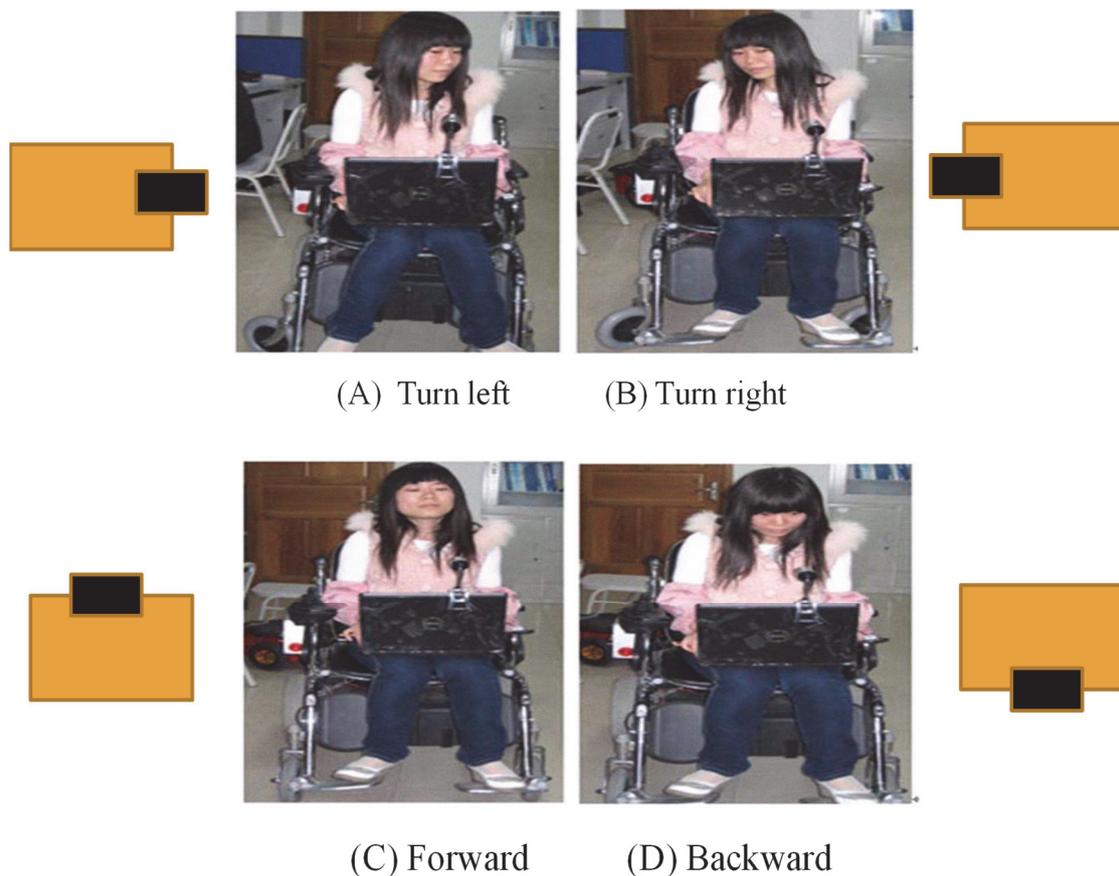
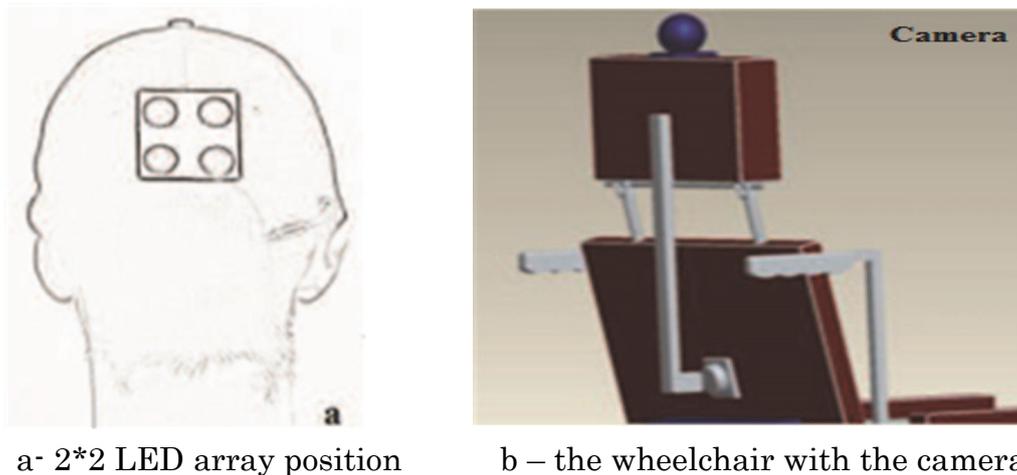


Figure 2-36: Gesture recognition system [141]

KUPETZ et al. proposed a head motion controller for an electrical wheelchair. The system is primarily designed for quadriplegia patients and depends on the use of camera-based motion tracking with an infrared LED array. The system can control the speed and direction of the wheelchair. The infrared LED array is positioned on the back of the user's head and consists of a 2*2 array. When the head of the user moves, an IR camera picks up the position of the array and sends details to a computer program. The program uses the segmentation of the image to detect the desired position or direction. This method is affected by the limitations of the camera in outdoor environments, and furthermore, it is not easy for the user to adjust the speed control. The system applies a braking action by hitting the head rest on the user's head. The system requires a computer to analyze and process the data, and this will add a high cost to the system. Figure 2-37 shows the structure of the system [136].



a- 2*2 LED array position

b – the wheelchair with the camera

Figure 2-37: The structure of the system [136]

KONDORI et al. proposed a Kinect sensor-based electric wheelchair operated by head movements to control movement and direction. The system captures images of the head using the Kinect camera fixed in front of the user's head. It scans the head's position at specific angles. The system can apply primary motion commands (forward, backward, left, right, stop) depending on the angle of the head compared to a reference angle. Figure 2-38 shows an overview of the system. The user of this type of system needs prior training to select the correct angles for each motion command. The practical results show that the system might face difficulties while operating outdoors due to the effect of sunlight on the Kinect [147]. This drawback can be solved by employing other depth sensors. The system still needs to be tested with a real wheelchair.

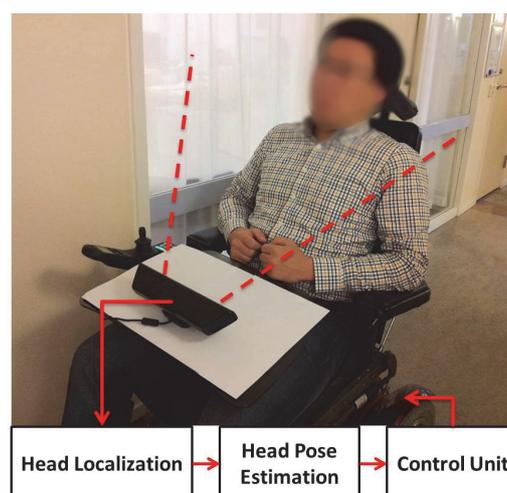


Figure 2-38: Kinect-based wheelchair controller [147]

ZHAN et al. proposed a pressure cushion controller for an intelligent wheelchair [148]. The system depends on the patient's body weight pressure at a particular point in the chair which is used to control the movement of the wheelchair. When the user moves his body to the right side, the pressure sensors on that side are activated and send electrical signals to the control circuit of the wheelchair. The same procedure is implemented in other directions. A fuzzy adaptive algorithm for pressure signals is used for the self-adjustment of the first primary pressure produced when the patient is attached to the chair. One of the drawbacks of this system is the different weights of different users, which produces different values of pressure. This method is not suitable for quadriplegia and paralyzed patients because they cannot fully control their body motions and weight pressure directions [149]–[151].

FORD et al. proposed an ultrasonic sensor-based head controller for electrical powered wheelchairs [138]. The system consists of two ultrasonic units. The first is used as a transmitter and is fixed on a holder surrounding the user's head, and it generates ultrasonic pulses (audio waves) which are transmitted through the air until collision and reflection by the user's head. The second ultrasonic sensor is used to pick up the reflected ultrasonic pulses to extract the positional information of the user's head by calculating the sending and reflecting the time of the audio waves. The system was designed to replace joystick operation. As the proposed system uses sound waves, it can be profoundly affected by the weather. In addition, a computer is required to extract and calculate the position data, which adds high cost to the system. Figure 2-39 explains the structure of the ultrasonic head controller system. Similar work with an infrared sensor instead of ultrasound has also been reported [137].

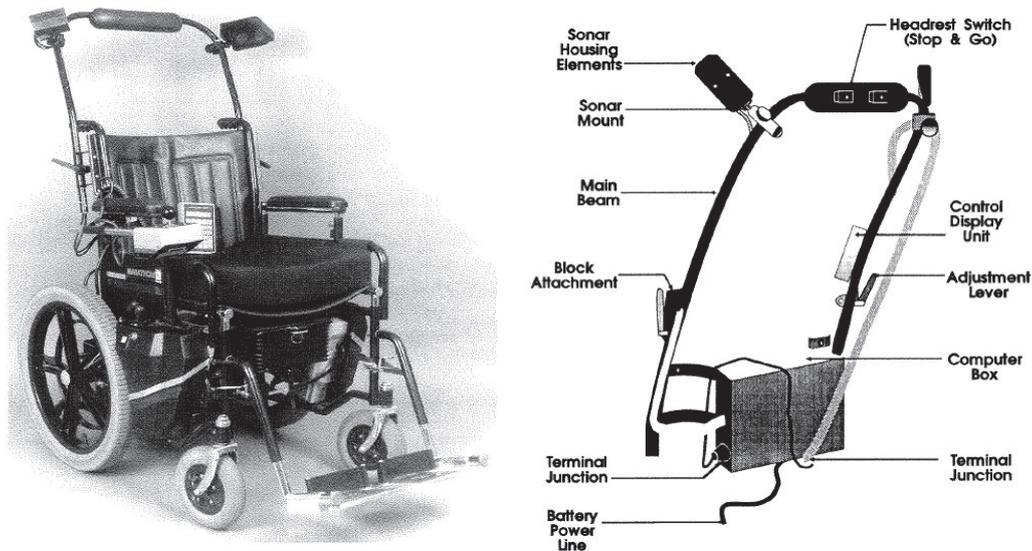


Figure 2-39: Ultrasonic head controlled wheelchair [138].

2.7 Orientation Sensor

The orientation sensor is an important type of measurement device that can be used to measure and detect the position, acceleration, tilt, and rotation of a target object. This source of orientation information allows the use of this sensor in many fields and applications such as the military, sports, aircraft, biomedical engineering, rehabilitation, and robotics. There are many types of orientation sensors which vary in structure and information processing modes, such as optical, mechanical, or electrical. The most well-known type of orientation modules consists typically of three accelerometer sensors measuring linear accelerations, three gyroscopes measuring angular velocity, and three magnetometers sensing the earth's magnetic field. This kind of orientation sensor is called an attitude and heading reference system (AHRS). In an AHRS system, every sensor is placed orthogonally to others of the same modality to provide measurements along each axis [152]–[159]. The orientation sensor can include one or more of the following devices.

2.7.1 Accelerometer

The accelerometer is an electromechanical device used to measure the acceleration of an object, where acceleration is the change in velocity over a specific time. The measuring unit of the accelerometer is (g) which is equal to 9.81m/s^2 . There are many types of accelerometers which differ in working principles, such as piezo electric, MEMS-based, capacitive, and piezo resistive devices. All of them convert a measured acceleration into an electrical value that can be further processed by a PC or microcontroller. The output of the accelerometer can be in

analog or digital form depending on need, design and the application field. It can be used to measure the tilt, vibration, impact, and motion of an object. The most common uses of accelerometers are rotation sensors in smartphones and impact sensors in cars to trigger airbags in the case of an accident. The measured signal can be classified into two types: a gravity acceleration signal and an external acceleration signal. The former can be used to measure the tilt of the accelerometer by determining the direction of sensor tilt after filtering external accelerations. When the target goal is to measure orientation in three axes, three axes orientation sensors must be used [160].

2.7.2 Magnetic Field Sensor (Magnetometer)

The magnetometer is a device that used to measure the magnetization of the material as well as the strength, direction and the relative changes in the magnetic field. It is used widely in many application like navigational purposes, metal detector, defense applications, anti-lock braking systems in vehicles, space explorations, geographical surveys and archeological surveys. The magnetometers are classified basically into two main groups which are the vector magnetometers and the scalar magnetometers. The vector magnetometer measures the magnitude and direction of the magnetic field in a specific direction in 3-dimensional space. The scalar magnetometer measures only the magnitude of the vector passing through the sensor regardless of the direction. There are several types of magnetometer based on their working principles such as torque magnetometer, optical magnetometer, Hall sensor, proton magnetometer, overhauser magnetometer and so on. The device output can be generated in the form of a change in voltage due to a magnetic field or a change in resistance due to the magnetic field. The resolution of a magnetometer is the smallest change in the magnetic field that can be resolved or reported by the magnetometer. [161]–[163].

2.7.3 Gyroscope

A gyroscope is a device used to measure angular rotation and the orientation of an object based on the principles of angular momentum. The orientation of the spin axes changes in response to external torque. There are many types of gyroscopes, such as mechanical, MEMS, solid-state, ring lasers, fiber optic devices, and the extremely sensitive quantum gyroscope. They can be used for object orientation detection, inertial navigation systems, and the stabilization of flying vehicles, and can be combined with accelerometer sensors to configure a highly sensitive orientation sensor. The MEMS gyroscope is the most sophisticated, small, power-saving, lightweight and low-cost devices. It requires very good

calibration because it is affected by environmental conditions like temperature and electromagnetic fields [164]–[166].

Many relevant studies used orientation sensors. **JOVANOV** et al. proposed a multi-sensor system for body position monitoring. In this work, an inexpensive real-time motion tracking system was developed for rehabilitation purposes. Intelligent inertial sensors are used for tracking and monitoring some individual body segments. Each intelligent sensor consists of 3 axis accelerometers and 2 axis gyroscopes. All the sensors in the sensor network are controlled by a master node which works as a gateway for communication with the acquisition devices. This system can be used for various applications such as for rehabilitation, virtual and augmented reality, gaming and so on. More details about this work and related work by the same authors are available [167]–[170].

GARNER et al. proposed an absolute non-contact orientation sensor for wrist motion control. In this work, the design and analysis of sensors for wrist-like actuator motion control have been implemented. The system is based on the use of vision-based sensors using a distinctive grid pattern. It allows the encoding of absolute orientation onto a spherical object. The design was tested in a system of motor driver control. This design is not appropriate for use at the present time since new and more sophisticated and accurate AHRS orientation sensors are more efficient, smaller and much lighter in weight [171].

Aiyun et al. proposed an application of low-cost orientation sensors for a robot navigation system. It consists of two sensors combined together which are a magnetometer and an accelerometer. It can provide accurate and real-time absolute orientation data for an autonomous mobile robot. It compares both the referenced coordinates O-XR-YR-ZR and the sensor coordinates O-X-Y-Z. The output of the sensor is processed using the DSP TMS320C2407A signal processing board. Data is collected by a sensor fixed onto the head of the robot. Figure 2-40 shows the coordinates of this orientation sensor [172].

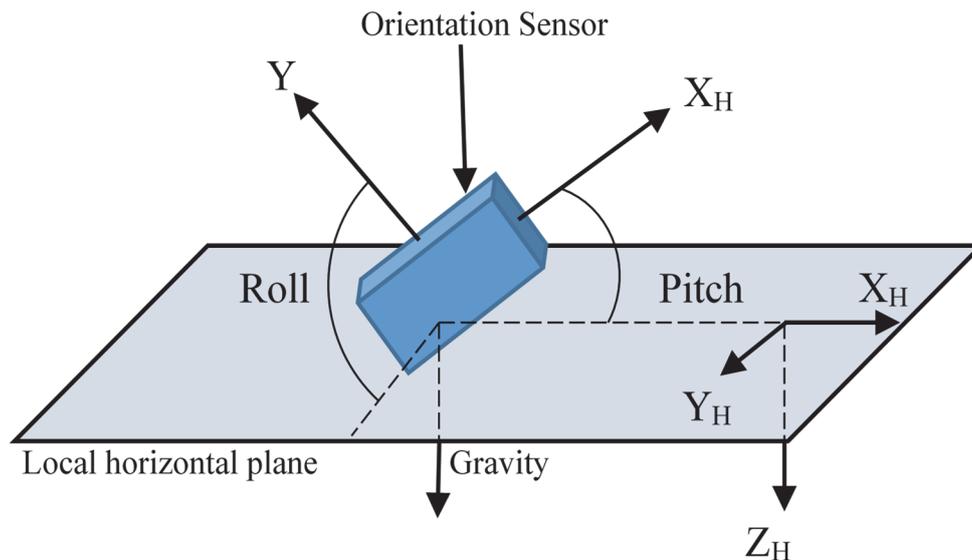


Figure 2-40: Coordinate of the orientation sensor [172]

HARMS et al. presented the design, implementation, and evaluation results of a miniature attitude and heading reference system (AHRS). The system can be used for human motion analysis and was tested with a sampling frequency of up to 128 Hz. The system includes various MEMS devices, including an accelerometer, gyroscope, magnetic field sensor, temperature sensor, and a power monitor system. Two sensor fusion algorithms were used to estimate body orientation, which are a quaternion-based Kalman filter and a complementary filter [173]. Figure 2-41 shows the AHRS module of the system. Similar studies are reported elsewhere [174]–[177].

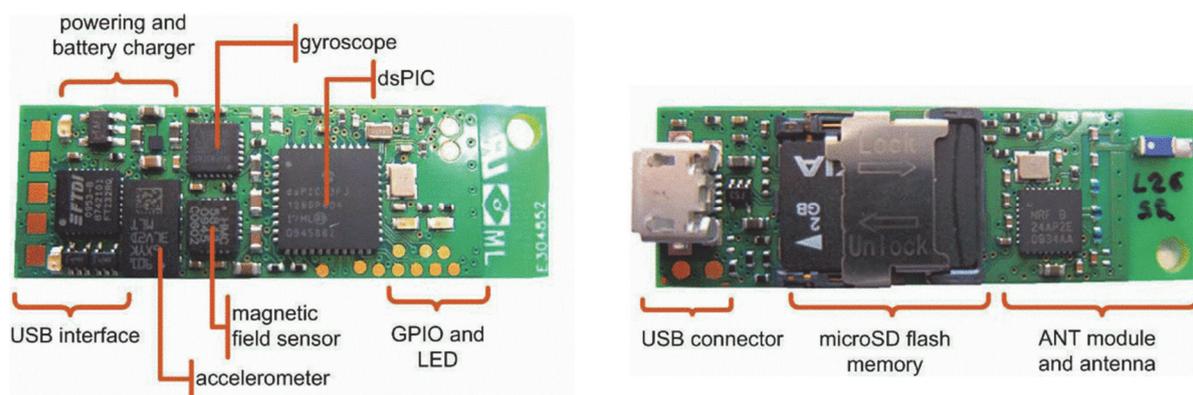


Figure 2-41: AHRS electronic module [173]

The three different research groups of **MANOGNA et al.**, **LU et al.** and **RATHORE et al.** have proposed prototypes for wheelchairs with control based on accelerometer sensors. The output of the accelerometers is digitalized and fed to the main microcontroller of the system. None of these systems has a speed control algorithm for the wheelchair, however, but apply a threshold point for direction control. When the tilt of the sensor goes to the right-hand side, the system gives a trigger to electromechanical switches via the microcontroller to make the motor turn the wheelchair in the right direction without sending any speed feedback to control the speed depending on the value of tilt angle (open loop control). The accelerometer sensor used to measure orientation is fixed to the head or hand of the user. All three systems are being tested as prototypes and need further enhancement before being used in a real-time application [178]–[180].

MILENKOVIĆ et al. proposed the use of smartphones for a smart wheelchair. This work includes a recording of the user's vital parameters and physical activity for a manual wheelchair using smartphone sensors. The system also includes the measurement of the tilt of a smartphone which is fixed to the wheelchair chassis. Smartphone tilt is similar to wheelchair tilt and tilt information is measured by the phone's accelerometer, gyroscope and magnetometer sensors to track the wheelchair's reference orientation. The availability of smaller low-cost orientation sensors means that the use of smartphones in the present study would not be the best choice. Figure 2-42 shows a general description of the system, and more details of this work and other related work can be found elsewhere [181], [182].



Figure 2-42: View of the system [181]

2.8 Summary and Conclusion

The available control methods give a broad range of solutions for the control of a wheelchair or rehabilitation robot by using brain output signals. Most of the work described above needs to be enhanced and developed so as to be applicable in real-time solutions, especially for medical and rehabilitation applications.

The voice recognition implementation faces the problems of false recognition in noisy environments and also the accuracy of correct voice command recognition is not sufficient to be used for rehabilitation applications. This problem occurs since the majority of the proposed systems use only one voice recognition processing algorithm. Most studies do not provide the option of using more than one mode of speaker dependent (SD) and speaker-independent (SI) voice recognition. Most of the above described work requires one or more PC to complete the process of voice recognition and to generate control commands, which adds to the cost to the system. These drawbacks can be avoided by using stand-alone voice recognition modules with powerful microcontrollers instead of computers.

The proposed EMG systems require the appropriate training of the user to control the muscles so as to give the correct EMG signal to the target muscle, which is then captured by the EMG electrodes to perform the required control action. This type of EMG training may be not easy and accessible for elderly and paralyzed patients. One of the drawbacks in using this technique for quadriplegia patients is the interference of noise signals from other muscles near the target muscles such as those in the face and around the eye. The use of EMG needs to fix special types of electrodes to the patient's body. Most EMG systems also use a PC for EMG signal classification, which again increases the cost of the EMG-based control system.

The use of EEG signals in control systems is still under development and is not commercially applicable to medical and real-time applications. It requires extensive complex training for the user to be able to generate the required brain signals. In this process, the user is expected to be able to concentrate and limit the thinking to specific control commands. Many drawbacks need to be taken into consideration in using EEG, such as the minuscule electrical values of brain signals which are easily affected by other electrical signals in the body such as electrocardiogram (ECG), EMG and EOG. Also, interference from noise from other power sources in the surroundings and the complex electrode system makes the use of this method not the easiest choice for wheelchair and rehabilitation robotic control.

One of the relevant solutions in the state-of-the-art in this field is the use of the head and body motions to control the movement of a wheelchair. Different methods and different sensors have been used to achieve this goal. Most previous work uses orientation sensors to control the motion of the wheelchair without using speed control. Recently, sophisticated low-power MEMS devices have been used in the construction of new generations of orientation sensors. It is recommended that these should be used as the second type of controller for the system proposed in this dissertation.

3 - Work Concept

Chapter 2 has introduced the most popular methods available for a user with a disability to control traditional wheelchairs. Every control method has advantages and disadvantages depending on the hardware and software elements and components used. The present work endeavors to overcome the weaknesses in existing solutions by designing a real-time solution that can be used in practice by paralyzed and quadriplegic patients. The suggested system will consist of three main parts which are the sensors and modules used for acquiring the input control signal, the processing unit, and the controller of the wheelchair's functions. A lot of details have to be taken into consideration for these three parts of the system so as to be better adapted for use by paralyzed patients. Figure 3-1 shows a schematic overview of the suggested system.

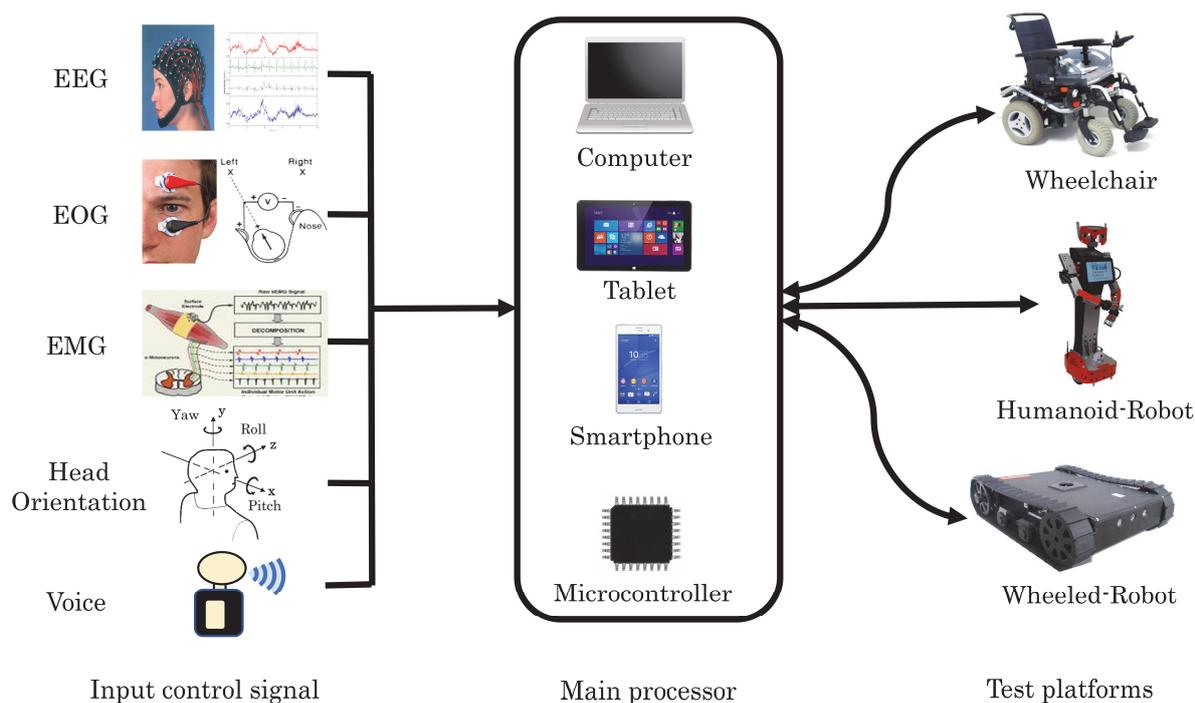


Figure 3-1: Schematic overview of the system

Each input signal from the user needs special sensors and detectors for data acquisition and processing. The processing unit must have the ability to deal with different types of data and information formats that will be measured by the input sensors. The processing unit must have the ability to control and communicate with the selected platforms, which may be wheelchairs or rehabilitation robots.

3.1 System Requirements

There are several requirements of the suggested system to allow it implements the main tasks required for the target users. One of the important requirement is the ability for easy adaptation between the system and the user because the target user has a type of disability that causes a severe impairment in some of the body regions which makes the interaction with the system more difficult. The system control commands must be simple as much as possible with minimum user efforts to avoid adding an additional effort for the disabled users. Medical and rehabilitation application required a safety function to protect the user in case of any an expected error or problem affect the system functionality.

There are several tasks and functions for the suggested system have to be considered in the design and development which are:

- 1- As the user is a person with special needs, the system must have more than one option/method to control it to allow the user to select the more comfortable method.
- 2- The control method must cover all the control commands and operate as much as possible at the same level of accuracy and functionality as the original design for normal user (joystick controller).
- 3- The user should be able to use all the system functions without the need for help from other people.
- 4- Additional protection functions must be added to sense if there is any abnormality in the sensors data to stop the system in case of any wrong data or user behavior (Unconsciousness) measured by the system sensors.
- 5- The system design must avoid using uncomfortable sensors or signal acquisition electrodes.

3.2 Input Units

From the above discussion of available signals in quadriplegia, two types of input signals have been selected to be used in the control of the wheelchair or rehabilitation robot, which are the user's voice and head movements. Each type of input signal requires a special device or sensors for acquisition and analysis. The user's voice and head motion are suggested as the best solution for paralyzed and quadriplegia patients to use to control a wheelchair with minimal patient effort. The design takes into consideration safety and the comfort of handling by the user, flexible system control and the costs of the system to achieve a comprehensive

control solution which can be used in various wheelchair systems and also for rehabilitation and handicap supporting robotic systems.

The present work endeavors to overcome the weaknesses of previous works. Some of the previously described signals were employed in the previous research, and the present work uses different strategies to implement the voice controller and also the head tilt controller.

3.2.1 Voice Recognition Modules

The user's voice will be picked up by a sensitive microphone. Microphones are transducers which detect acoustical sound pressure and produce an electrical form of the sound; that is, they produce a voltage or a current which is proportional to the sound signal. The most common microphones are dynamic, ribbon, or condenser microphones. Besides the variety of basic mechanisms, microphones can be designed with different directional patterns and impedances. For the proposed work, more than one voice recognition module is used so that the user's voice commands are processed by two or more voice processors. This can give a higher likelihood of correct voice recognition and reduced rates of recognition errors [183], [184]. The selected microphone must be precisely adapted for the modules used, or otherwise, a specific microphone for each voice recognition module is needed; normally, the microphone specifications are given by the manufacturer or vendor. The microphone must be fixed in a flexible location near the user's mouth (<20 cm). This can be achieved by using headset microphones, hanging the microphone on the user's shirt or using a fixed microphone with a movable arm. A microphone with a wide frequency spectrum, such as 50-20000 Hz, can be used to cover the differences in sound frequency of different users' voices. The selection of microphone impedance depends on the distance between the microphone and the voice processing unit. For a wheelchair user the microphone position will be near the voice recognition module (<1.5 m) so that a medium microphone impedance range (600 Ω -10,000 Ω) is optimal for the proposed system [185]. As the microphone used for voice recognition is a sound pressure level (SPL) application, a sensitivity of ≤ 65 dB will cover the requirements of the proposed system [186].

The suggested voice controller will consist of two or more voice recognition (VR) modules. The user voice command will be picked up and sent to all of the modules in the voice controller at the same time. The main microcontroller will be programmed to collect the voice recognition results from the different modules and

compare the VR results to check if there are differences in the recognition results. For better results, the selected voice recognition modules must differ in the voice processor used (from different vendors or manufacturers) to achieve more credible recognition results. The selection of modules that work in different VR modes, such as speaker-dependent (SD) and speaker-independent (SI) modes, will allow the system to be used for different applications. For example, an application designed to be used by only one specific person requires a VR system that works in SD mode. This will prevent unauthorized persons from using the system even when giving the same already-stored commands. SI mode can be used for other applications designed to deal with different users, such as games and telephones.

The main reason for using more than one VR module in the suggested system is to let the system recognize the FP errors which are the most harmful VR errors. The use of only one module will prevent the ability to compare the VR results of the voice commands and with only one VR result, it is difficult for the system to detect correct or wrong voice commands. When the system has more than one VR result, it can check if the recognition result is identical or not. If they are not identical, the system can ignore the implementation of the given voice command or ask the user to repeat the voice command. Also, the use of more than one VR module will increase the probability of the system achieving the correct VR if any one of the modules is unable to distinguish voice commands. Figure 3-2 explains the probability of successful VR when using one or two VR modules. The number 1 in the output refers to successful VR recognition and the number 0 refers to failure. Successful VR does not mean that the system recognizes the command correctly, but only that it has recognized a voice command and this can be a correct or false recognition.

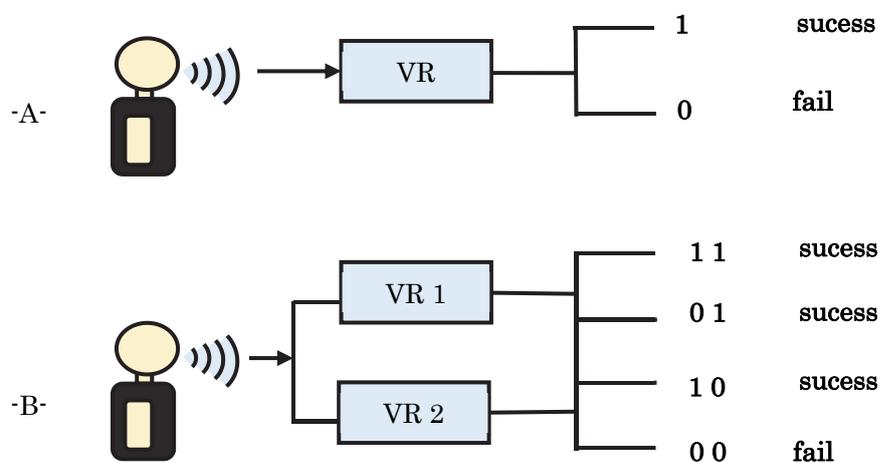


Figure 3-2: VR success probability using one and two VR modules

3.2.2 Head Tilt Controller

In chapter 2, Kinect cameras, ultrasonic sensors, infrared sensors and inertial sensors were introduced as means to measure head movements. The use of a Kinect camera requires high-performance computers for the handling and analysis of video images; in addition, the large size of the Kinect camera (31*6.5 cm) is not convenient compared to that of available inertial and orientation sensors. Ultrasonic and infrared sensors can be used for head tilt tracking, but they are not the best choice for systems that may be used for outdoor navigation due to the effects of weather and sunlight on their performance. Furthermore, the effective angle for most ultrasonic sensors is less than 15°, and this not sufficient to track head movements in all directions.

The best choice for motion tracking is currently the use of inertial sensors. Several types of sophisticated MEMS inertial orientation sensors or attitude and heading reference systems (AHRSs) are available which can provide body orientation data in different formats with high resolution and sufficient accuracy. For the current system, a small and lightweight sensor with dimensions of 20x20x10 mm in width, length and height can be used. The small size and light weight make it easily adapted to a headset or other wearable device to measure the user's head orientation. Most orientation and AHRS modules consist of an accelerometer, gyroscope, and magnetometer, or just two of these components. An acceleration range of ± 2 g of an embedded accelerometer and a gyroscope with a range of ± 2000 degrees per second are sufficient parameters for orientation determination. The orientation sensor used must provide a standard communication bus to offer flexible data transfer to the processing unit.

The head tilt controller must replace the original joystick functionality of the electrical wheelchair. The user's head tilt needs to be interpreted at any point as particular speeds and directions of the wheelchair or rehabilitation robot. The proposed system replaces the joystick with the user's head and head motions and tilt are similar to joystick motions and must have similar control effects. The design of the head tilt controller must take into consideration the user's vision level so that, for the forward tilt, the user's head tilt must maintain the user's eyes facing the front as normal. The system design needs to use sophisticated orientation modules that can give accurate readings of the motion of the head.

A reference orientation for the system can be used in the head tilt controller to enhance system performance when the wheelchair or robot passes long non-straight roads. Two orientation modules need to be used to measure the user's

head tilt and wheelchair orientation. The measurement of the head and wheelchair orientation allows the system to avoid change in head orientation when the system ascends or descends a hill.

3.3 The Processing Unit

There are several options for a system processor to control the wheelchair or rehabilitation robot, including PCs, tablets, smartphones, and microcontrollers. The selection of the system processor has to be based on parameters such as processing speed, cost, size, and power consumption. The proposed system will deal with several sensors and modules with different operating concepts, data formats, sampling rates, and communications ports.

Processor speed is an important factor in the selection of a processing unit. The selection of a faster processor will help to get better performance in data processing and multi-task implementation. At the same time, the selection of a faster processor increases the cost and power consumption of the system. In the present work, the processor speed required is based on the speed required by the peripheral sensors and modules (voice recognition modules, head orientation sensors, and motor driver unit). Most of the available traditional sensors and modules use the standards as universal asynchronous receiver and transmitter (UART), serial peripheral interface (SPI), I²C, and USB communication ports to communicate with the host processor.

The baud rate of the UART port can reach 460,800 bps (bits per second) with a bit processing time of 2.17 μ s. Only two pins are required for communication whereby only one device uses the TX-pin for transmission and the RX-pin for the reception. Most available modules and sensors with UART use standard baud rates between 9600 and 115200 bps and a maximum processing time of 8.68 μ s which can be processed by any processor with a speed greater than 10 MHz [187].

The SPI port is used for specific applications especially in combination with microcontrollers. It is not as widely used as UART and I²C buses. The SPI bus is used for synchronous serial communication between one host processor and one or more slave sensors or devices. It requires a minimum of 4 pins for communication between the host microcontroller and the slave devices. The SPI is faster than the UART and can operate up to 50 Mbps [188].

The I²C protocol is used widely in modern sensors and in micro-electro-mechanical systems (MEMSs) which are used in most modern orientation sensors. The I²C bus allows the host processor to communicate with up to 127 sensors using

only one bus. The maximum speed for the I²C port is 5 MHz and it operates in so-called ultra-fast mode (UFM). In most modern I²C applications, the following standard modes are used: fast mode with 400 kHz, a high-speed mode with 3.4 MHz, and fast mode-plus with 1MHz. All of these speed variations can be supported by modern computers and microcontrollers [189].

A laptop can be used to be the processor of the system due to its high performance, but this will increase cost and size and a suitable location on the wheelchair will be needed to locate the laptop with input and output units. The use of a laptop requires more booting time than a microcontroller which is turned on with a short booting time (<1 sec). A system with a laptop requires previous settings for the programs that receive information from the sensors and modules and the programs that send commands to the wheelchair or robot using socket programming (server-client). For the above reasons, the laptop is not the best choice for the proposed system.

The use of a tablet or smartphone as a controller for the system will provide ready-to-use Bluetooth and motion tracking sensors that can be used to measure the reference orientation of the wheelchair. They can operate with high data processing performance and also will reduce the size of the system in comparison with a laptop. Tablets and smartphones are not very flexible in dealing with the ports required to cover the input/output unit sensors and modules which may need different communication buses like I²C, UART, SPI, or CAN bus. The costs of these devices are still high compared with the use of a microcontroller and they need to be set by another person because of the limitations of quadriplegia.

From the above information about sensors, communication ports and data transfer rates, a microcontroller processor with a speed of more than 16 MHz or a computer, tablet or smartphone can be used to communicate with the available sensors. Microcontrollers are very cheap compared to computers, tablets, and smartphones and this gives them an advantage in the proposed system. Also, the size of the microcontroller is another advantage which will allow its flexible integration in small objects and spaces like, for example, headsets for orientation detection. The third advantage of using a microcontroller is its low power consumption of less than 300 μ A/MHz in modern devices. This makes it an optimal selection for battery-powered applications. The use of a computer is the most flexible solution for programming tasks compared to microcontrollers, especially when dealing with several sensors and modules. Table 3-1 compares the technical features of selected laptops, tablets, smartphones and microcontrollers [190]–[192].

Table 3-1: Comparison between selected processing units

Processor	Speed	No. of Cores	No. of Threads	Blue tooth	GPIO	USB	I ² C	UART	WiFi	Size L*W*H	Price
Lenovo G50 i7 core laptop	2.4 GHz	2	4	Y	N	Y	N	N	Y	39*27*2.5 cm	600€
Samsung T530 tablet	1.2 GHz	4	4	Y	N	Y	N	N	Y	25*18*0.8 cm	350€
Apple iPhone 4S	1 GHz	2	2	Y	N	Y	N	N	Y	11.6*5.9*0.93 cm	300€
STK3700 32GG990 MCU	40 MHz	1	1	N	86	Y	2	5	N	8*5*.05	25€

3.4 Output Units

As the system is to be used by quadriplegics and paralysis patients, wheelchairs and the rehabilitation robots are used as test platforms. The output units for these platforms are the motor driver units, the lights and signals unit, and the display unit. Each robot or wheelchair may use a special type of motor driver unit depending on the motors used and the type of communication bus between the host processor and the motor driver unit. All motion control methods in wheelchair execution are based on sending control commands to the motor driver unit. This unit is responsible for sending power to the wheelchair or robot motors and it controls the speed and direction of motor rotation by changing the power duty cycle and power polarity. Each motor driver unit has a vendor-specific communication protocol for the motors to receive specific speed and direction commands. Most modern electrical wheelchairs use CAN bus-based motor driver units. There are several types of CAN bus, each of which uses different communication protocols depending on the manufacturer. In the proposed system, a general-purpose motor driver unit can be used to make the system easily adaptable to several types of platform. It must be able to supply the power required by the wheelchair and robot motors, which is rated in the range of approximately 220 to 400 Watts.

The speed sensitivity control of the system motor is one of the most important tasks of the proposed system. The selected input parameters of the user's voice and head orientation need smooth speed control. Most electrically powered wheelchairs use three levels or more in their speed range, which are basic, medium and fast, and these can be changed using two buttons in the joystick board (see

Figure 2-2). In the present work, the head tilt controller can be programmed to cover only the control of motor speed and direction at one level and the voice commands can be used easily to change between the three-speeds levels.

For each speed level, the head tilt controller can control the speed of the motors from 0 to the maximum speed at the selected level. That means for a wheelchair at normal speed a maximum of 2 km/h can be reached, with medium speed 4 km/h and with fast speed 6 km/h. Head tilt can control the speed at each individual level from 0 to the maximum speed by tilting the head from 0° to 25° .

Other features, like a sound alarm, lights, and signals, can be controlled using general purpose input/output ports of the host microcontroller. Most wheelchair horns and lights operate with 12-24 DC volts. The easiest way for the microcontroller to control these types of horns and lights is by using a relay with 3.3 V for the coil and a minimum of 24 V, 10 A for the terminals. Only one GPIO pin is sufficient for the main system's lights and another GPIO for the horn. The same procedure can be used for left, right and stop light signals. The difference with the operation of lights is that they only use on/off from the GPIO pins, while the signals need to program the GPIO pins with pulse width modulation (PWM) to let the signal to be powered with a duty cycle of $\approx 70\%$ each second. Three GPIO pins need to be used for left, right and stop light signals.

The display can be used to send an alert message to the user in the case of any error or abnormal situation facing the system. One such important situation is when the system faces an obstacle and the obstacle avoidance system stops the wheelchair motor. The system can send an indication to the display to inform the user about the obstacle and give a suggestion for the next movement. It can also be used to indicate errors when the head orientation sensor tilt further than the normal range (0° - 25° for forward), where the system can send an error statement that instructs the user to go to the stop control region (-10° - 0° for stop). The display must show and continuously update system status information, such as which controller is activated and also which commands are being executed. Figure 3-3 shows the structure of the suggested system.

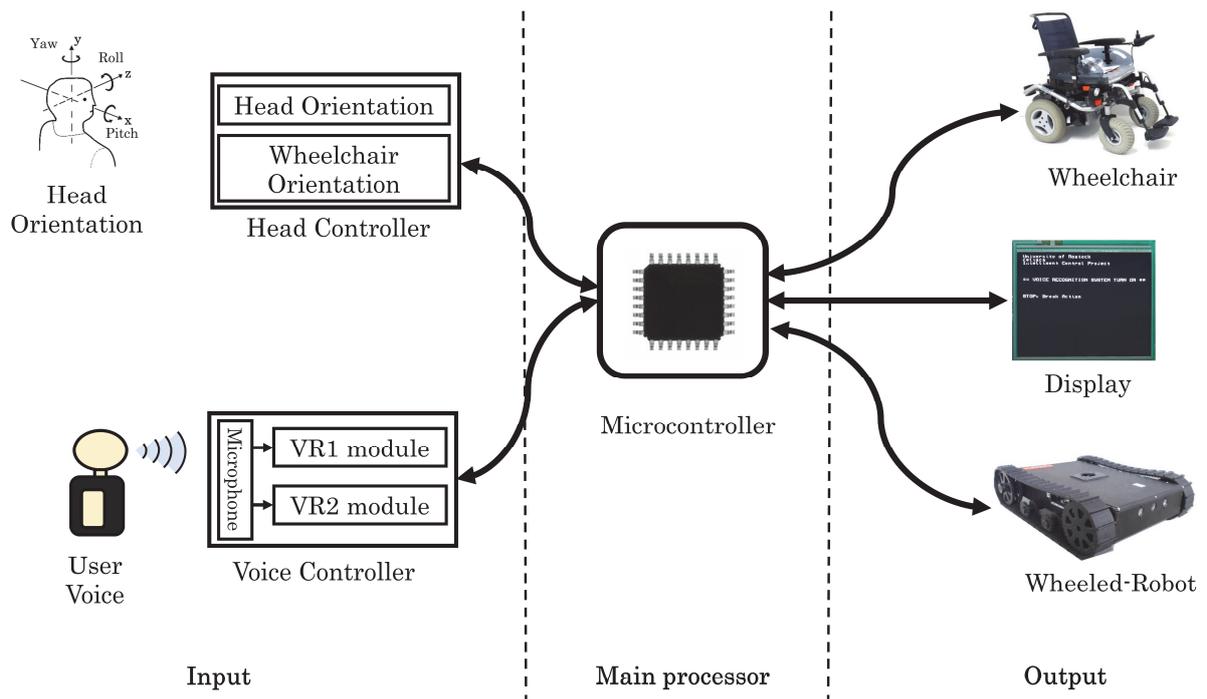


Figure 3-3: Suggested system structure

4 - System Development

4.1 Microcontroller Unit (MCU)

As the proposed work is a multi-input human machine interface (HMI) system, the system design requires the use of several modules and sensors as input to the microcontroller. The system needs to be driven by a powerful microcontroller that can receive information from all sensors and modules, process incoming data and information and make optimal decisions for wheelchair control. It must be able to manage the prioritizing of command implementation depending on the system's current situation as informed by the external sensors and units. It needs to have more peripheral communication and GPIO ports to allow the system to interface with the input lines from the recognition unit and sensors, and with the output line to the motor driving unit and LED screen. Additionally, simple updating is one of the future needs. Table 4-1 explain a comparison of the main embedded features of selected microcontrollers from different companies.

Table 4-1: Comparison of selected microcontrollers

Microcontroller	Speed	Flash	RAM	GPIO	USB	I ² C	UART	Timer	Price
Microchip* PIC16F877	20 MHz	8 K	368 bytes	32	N	1	1	3	3 €
Silicon Lab** EFM32GG990	48 MHz	1 M	128 k	86	Y	2	7	9	9 €
STM*** 32F030R8T6	48 MHz	64 k	8 k	55	N	2	6	11	5 €
Atmel**** ATmega2561	16 MHz	256 k	4 k	54	Y	1	2	6	12 €
Texas Inst.***** MSP430G2353IN20	16 MHz	4 k	256 k	24	Y	1	1	2	3 €
Atmel AVR**** AT32UC3A0512	66 MHz	128 K	256 k	69	Y	1	4	3	15 €
Tiva™ ***** TM4C129XNCZAD	120 MHz	1 M	256 k	140	Y	10	8	8	20 €

* Microchip Technology Inc., Arizona, United States

** Silicon Labs, Austin, Texas, United States

*** STMicroelectronics, Geneva, Switzerland

**** Atmel Corporation, San Jose, California, United States

***** Texas Instruments Inc., Dallas, Texas, United States

There are several types of microcontrollers from different embedded technology companies which can be used for the current work. The Silicon Labs, Austin, Texas, USA, ARM Cortex M3 EFM32GG990F1024 microcontroller has been selected to be the core of the system. This microcontroller is chosen to achieve high performance, low cost, low power consumption and small size. It uses an ARM architecture with a 32-bit processor, 1 MB Flash memory, 28 KB RAM and a broad selection of peripherals that can implement all the processing and control tasks requested from the application side. Many reasons related to the target applications have encouraged the selection of this MCU in the present work, including the following:

- Multi-universal synchronous/asynchronous receiver/transmitter UART/USART ports: The voice recognition modules need to communicate through two UART ports, and the robot/wheelchair needs one for motor driver communication. The selected microcontroller has 5 UART/USART ports which cover the current and future system requirements.
- Two I²C bus interface: The I²C bus is required for the orientation detection BNO055 modules communication. The current implementation requires only one I²C bus and the second one can be reserved for future work.
- 3 Pulse counters: the system needs a two-pulse counter to communicate with the motor encoder to calculate the speed feedback of the system's motors.
- Power consumption: This enables active mode processing for running application code at all energy modes at EM0-EM4 with an approximate current consumption between 219 μ A/MHz to 20nA/MHz for only 114 μ A/MHz with a 3 V power supply
- Wake up time: It has a short wake-up time of 2 μ s. The wake-up time is the time required to change the energy mode from different sleep modes to active mode.
- Processing time: The 32-bit ARM low power and high-performance MCU reduces the data processing time in all operation energy states.
- Ultra-low standby current of 9 μ A
- Peripheral Reflex System (PRS): It can implement fast signaling and predictable interfacing between peripherals without CPU intervention.

- Different energy modes: It has five different energy modes with different levels of power consumption (EM0-EM4)
- Energy efficient peripherals: It has a wide range of extremely energy-efficient MCU peripherals, such as an Analog comparator with 100 nA, low energy UART EUART at 9600 bps with 150 nA, LCD controller of 8x36 segments with 0.55 μ A and analog to digital converter ADC of 12-bit, 1 Msps at 350 μ A [193].

The above unique features make the ARM EFM32GG990F1024 one of the most energy efficient microcontrollers in the world [194]. It has a unique combination of powerful ARM 32-bit Cortex-M3, is innovative in excellent low-energy techniques, and has a wide selection of peripherals and a short wake-up time from energy saving modes. The EFM32GG990F1024 microcontroller is a very good choice for any battery-operated application as well as other systems requiring high-performance and low-energy consumption. It covers all the requirements of the multi-input/output units of the proposed system. It can work with up to 48 MHz frequencies that make it flexible enough to deal with data processing and communication management from many ports and peripherals at the same time, which is strongly recommended for the proposed system. It has 86 general purpose input-outputs (GPIOs), 3 \times universal synchronous / asynchronous receiver / transmitter (USARTs), 2 \times universal asynchronous receiver / transmitters (UARTs), 2 \times low energy UARTs, 2 \times I²C interfaces and many other important embedded features such as a group of pulse counters (PCs), real time counters (RTCs), analog comparators, operational amplifiers, a 12-channel peripheral reflex system (PRS) for autonomous inter-peripheral signaling, a 12-channel direct memory access DMA controller, and 16 asynchronous external interrupts. The universal asynchronous serial receiver and transmitter (UART) is a flexible serial I/O module. It supports full- and half-duplex asynchronous UART communication and will be used as an input port for the VR2 Easy_VR module which communicates with the host processor only through the UART communication ports. The UART port is also used for communication between the ARM MCU and the Jaguar-Lite robot. The 86 GPIO pins are divided into ports with up to 16 pins each. These pins can be individually configured for either output or input. The GPIO pins are overridden by the peripheral pin connections, such as the pulse width modulation (PWM) output of the timer or the USART communication, which can be routed to several locations on the device. Many other

active features are included in this ARM Cortex M3 microcontroller. Figure 4-1 shows the microcontroller block diagram [194].

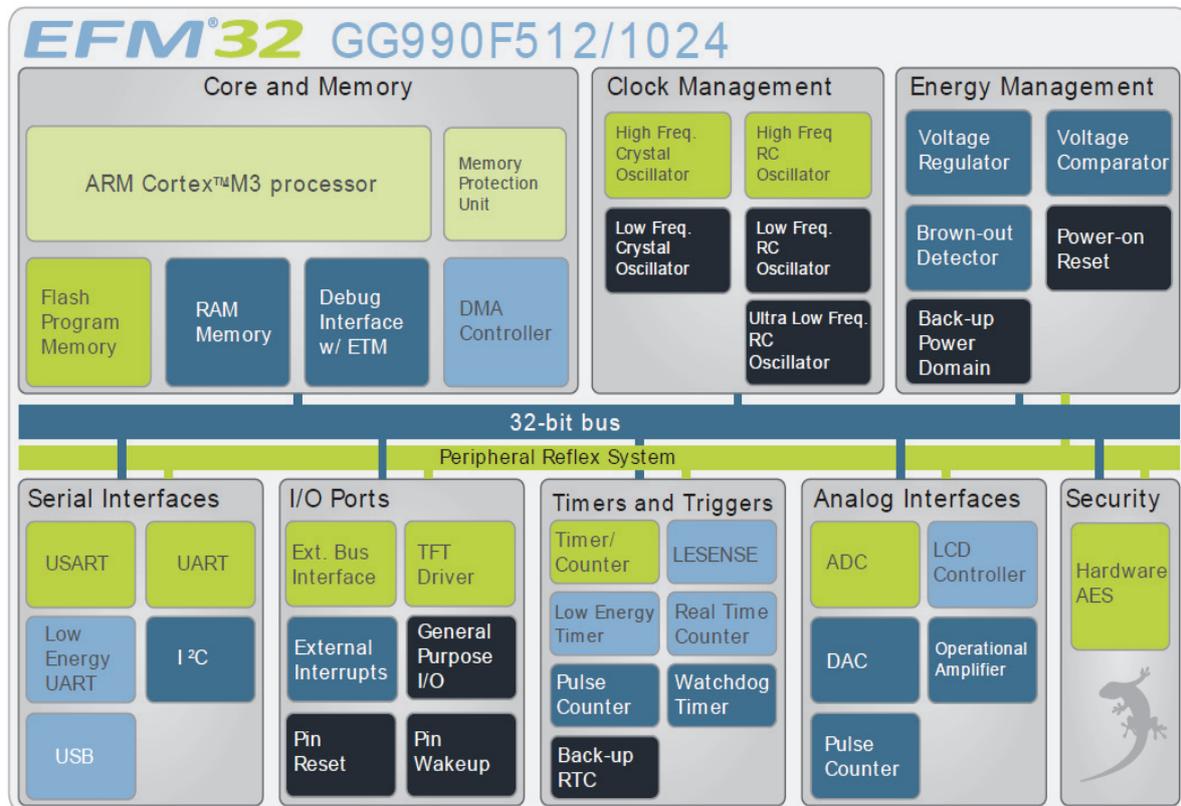


Figure 4-1: EFM32GG990F1024 microcontroller block diagram [194]

The following sections introduce the important features of the selected microcontroller:

4.1.1 Cortex-M3 Core

ARM Cortex-M3 microcontrollers have a 32-bit RISC processor. It can achieve more than 1.25 Dhrystone MIPS/MHz. It has a memory protection unit that can support up to 8 memory segments. It has a wake-up interrupt controller for managing interrupts during CPU sleep. The combination of the powerful 32-bit ARM processor, the innovative low-energy technology, a wide range of peripherals and short wake-up time from energy saving modes make the EFM32 microcontrollers a good choice for any battery-operated application that needs high performance with low energy consumptions [194].

4.1.2 Memory System Controller (MSC)

The memory system controller (MSC) represents the program memory of the EFM32GG microcontroller family. It has readable and writable flash memory that can be used by both the Cortex-M3 and the direct memory access (DMA) controller. The flash memory has two blocks: the main block and the information block. The main block is used for the program code writing while the information block is used for user data and the flash lock bit. There are also read-only parts of the information blocks used for storing system and calibration data.

4.1.3 General Purpose Input/Output (GPIO)

The EFM32GG990 microcontroller has 86 general purpose input/output (GPIO) pins. They are divided into ports named A, B, C, D, E and F. Each port includes a standard 16 pins, while some ports have fewer pins. Each pin can be individually set as an input or output pin. Advanced configuration can also be applied for open-drain, filtering, and drive strength. The GPIO pins can be overridden by peripheral ports like USART and Timer PWM communication, which can be routed to many locations in the device's GPIO ports. The input value of the individual pin can be routed via the peripheral reflex system (PRS) to any other peripherals. The GPIO ports support asynchronous interrupt up to 16 external pins, which can enable an interrupt from any device pin.

4.1.4 Energy Modes

The EFM32GG microcontroller family has five different energy modes from EM0 to EM4. A high degree of autonomous operation can be achieved for this type of microcontroller in low energy mode. The short 2 μ s wake-up time and the intelligent combination of peripherals, DMA, low power oscillator and RAM with data retention make the EFM32 family attractive to stay in low energy mode for a long time with effective low energy consumption. Table 4-2 explains the different energy modes of the EFM32 microcontroller.

Table 4-2: Energy modes for EFM32 microcontroller [194]

Energy Mode	Name	Description
	EM0 Energy Mode 0 (Run mode)	In EM0, the CPU is running and consuming as little as 219 $\mu\text{A}/\text{MHz}$, when running code from flash. All peripherals can be active.
	EM1 Energy Mode 1 (Sleep Mode)	In EM1, the CPU is sleeping and the power consumption is only 80 $\mu\text{A}/\text{MHz}$. All peripherals, including DMA, PRS and memory system, are still available.
	EM2 Energy Mode 2 (Deep Sleep)	In EM2 the high frequency oscillator is turned off, but with the 32.768 kHz oscillator running, selected low energy peripherals (LCD, RTC, LETIMER, PCNT, LEUART, I ² C, LESENSE, OPAMP, USB, WDOG and ACMP) are still available. This gives a high degree of autonomous operation with a current consumption as low as 1.1 μA with RTC enabled. Power-on Reset, Brown-out Detection and full RAM and CPU retention is also included.
	EM3 Energy Mode 3 (Stop Mode)	In EM3, the low-frequency oscillator is disabled, but there is still full CPU and RAM retention, as well as Power-on Reset, Pin reset, EM4 wake-up, and Brown-out Detection, with a consumption of only 0.8 μA . The low-power ACMP, asynchronous external interrupt, PCNT, and I ² C can wake-up the device. Even in this mode, the wake-up time is a few microseconds
	EM4 Energy Mode 4 (Shutoff Mode)	In EM4, the current is down to 20 nA and all chip functionality is turned off except the pin reset, GPIO pin wake-up, GPIO pin retention, Backup RTC (including retention RAM) and the Power-On Reset. All pins are put into their reset state.

4.1.5 EFM32 UART/USART

The UART/USART ports have an important role in the proposed system because they are used for communication between the ARM microcontroller and the two voice recognition modules. One of the important factors in current microcontroller selection is the availability of 3 USART and 2 UART modules that can work separately at the same time, so this one covers the current and future requirements of the present application. The UART/USART modules have a programmable baud rate and a wide selection of operational modes. They can work in full duplex and half duplex and with separate TX / RX channels. They have separate interrupt vectors for receiving and transmitting interrupts and can detect hardware collision which allows the avoidance of data corruption if two pieces of hardware use the same bus for data transmission.

4.1.6 Inter-Integrated Circuit Interface I²C

The I²C module provides an important interface between the microcontroller and the modules and sensor that uses the I²C bus for communication and data exchange as master or slave. It allows for communication with the minimum possible energy consumption and uses only two I/O pins which are a serial data line (SDA) and a serial clock line (SCL). In the present work, the I²C bus is used for communication between the microcontroller and two BNO055 orientation detection sensors used by the head tilt controller. The I²C bus can communicate with up to 128 sensors that must have different I²C addresses to allow the master microcontroller to address each sensor at the bus. The EFM32 microcontroller has two I²C buses that can run in a standard mode with a bit rate of up to 100 kbit/s, a fast mode with a bit rate of 400 kbit/s and fast mode plus with a bit rate up to 1Mbit/s.

4.1.7 Programming Integrated Development Environment IDE

Several IDEs can be used to program the ARM EFM32GG990F1024 microcontroller, such as the IAR Embedded Workbench for ARM, Atollic TrueSTUDIO for ARM, Rowley Associates - CrossWorks for ARM, Code Sourcery and Keil MDK-ARM [195]. The current work has been written in Keil μ Vision5 C/C++ IDE. This IDE combines several functions in a single powerful environment, such as project management, run-time environment, source code editing, and code building and debugging facilities.

4.2 Voice Recognition

Voice recognition technology has been established based on the conversion of spoken words into an electrical signal which is further processed by a computer and microcontroller and then converted into control commands. There are two types of isolated voice recognition depending on the response to the speaker's sound features, which are speaker-dependent (SD) and speaker-independent (SI). Each involves a different training process and can be used for different applications.

4.2.1 Speaker Dependent VR

The speaker dependent mode (SD) means that the system response to specific speaker depending on its voice characteristic. It is used signal processing techniques to compare the feature of the reference and the live voice command. It requires prior training by the user to record the sound characteristics, so that if two people give the same command to the system one of them is trained before and the other not, the system will respond to the individual who trained to the system before. Figure 4-2 shows the SD system.

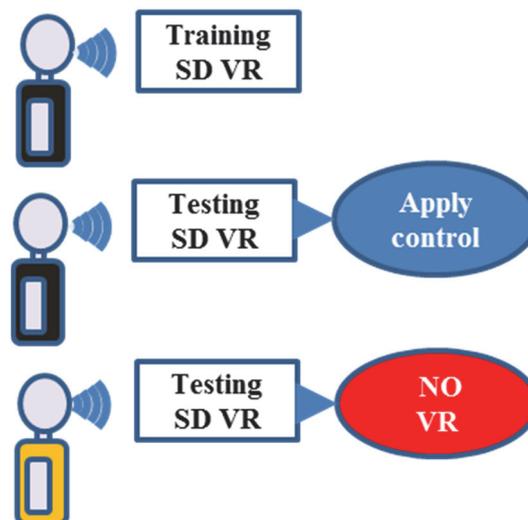


Figure 4-2: Speaker-dependent VR system

4.2.2 Speaker Independent VR

Speaker-independent mode (SI) means that the system responds to any user without depending on individual sound characteristics. This type of voice recognition technology converts the voice command into text. Then the text is compared with text commands selected from the command vocabulary library. This type of voice recognition does not require previous training and interaction with any user who gives an already selected command from the vocabulary

command library is possible. Figure 4-3 shows the speaker-independent voice recognition system.

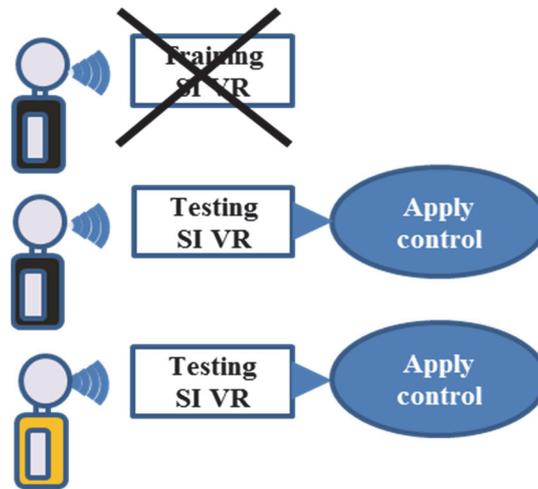


Figure 4-3: Speaker-independent VR system

4.2.3 Voice Controller Embedded Solution

Since the present design uses an embedded device (EFM32 MCU) as the core of the system, the sensors and modules used should be compatible with the selected microcontroller to achieve the required performance. One of the important factors in the selection of voice modules is the type of communication bus, which is important to avoid the complexity of adding new hardware for communication with the microcontroller. The second factor in the selection of the VR module is its size of the module, which is an important factor in the shape of the final system design such as a wearable device. The third factor in selection is cost, where the low cost is an important factor that would allow more patients to use the system.

Generally, there are two main types of solutions when implementing voice recognition, which are as follows:

- Software solution: Voice recognition (VR) is implemented using VR software such as Dragon, Julian, AquesTalk, or MATLAB. This solution is more expensive and a computer needs to be used as the system processor which adds additional cost.
- Embedded solution: using VR modules like V3, HM2007, Easy VR, Smart VR, and Speak Up Click. Most embedded modules use one type of VR algorithm and most are small size and cheap compared to the use of software solutions.

In the current work, the embedded solution has been selected to implement the voice controller. There are several embedded voice recognition modules produced by different companies. Table 4-3 explains the features of several embedded voice recognition modules which could be used in the proposed system.

Table 4-3: Comparison of embedded VR modules

VR module	No of Commands		Command duration	Bus	Response Time	SD	SI	Price
	Total	Active						
HM2007*	40	20	1.92 sec	BCD	300 ms	Y	N	170 €
Easy VR**	32	12	3 sec	UART	200 ms	Y	Y	35 €
V2 module***	15	5	1.3 sec	UART	700 ms	Y	N	30 €
SpeakUP Click****	200	100	3 sec	UART	250 ms	Y	N	40 €
Simple VR***	64	64	3 sec	UART	700 ms	N	Y	34 €

* Images SI Inc., Staten Island, New York, USA

** ROBOTTECH SRL, Pisa, Italy

*** Elechouse comp., Shen Zhen, China

**** MikroElektronika D.O.O., Belgrade, Serbia

Two sophisticated VR modules have been selected to implement the voice controller. The integration of two VR modules will help to recognize wrong recognition. The first selected module is the SpeakUp Click (MikroElektronika D.O.O., Belgrade, Serbia). This module is able to operate 100 active voice commands at a short response time of ≈ 200 ms. The second selected VR module is the Easy VR (ROBOTTECH SRL, Pisa, Italy), which can be operated in SD as well as SI modes. The voice controller is one of the input units for the proposed system. These two VR modules use different voice processing algorithms. They can be combined to build a strong VR unit that increases VR accuracy and reduces error. The modules used are explained in detail in the following sections.

4.2.3.1 SpeakUp Click Module (VR1)

The first VR module is the SpeakUp Click module (MikroElektronika D.O.O., Belgrade, Serbia). It works only in speaker-dependent mode (SD). In SD mode, the VR process occurs depending on the sound characteristics of the user so that the system will respond only to a user who has trained the system before. VR execution in this module is realized using a dynamic time warping (DTW) algorithm. DTW

is a well-known signal processing technique used to detect and measure the similarity between two information sequences. It calculates the optimal warping path between two-time series of different lengths. One of them represents the training reference sample, and the other is the test sample. This unit can train with up to 200 voice commands and can be adapted to the external host using 12 GPIO pins and also by the UART communication port. The voice command time can be adjusted up to 3 seconds. The output of the module can be in the form of high 3.3 v, low 0 v or pulse for a specific time on the GPIO. It can also send the voice recognition results as a command index through the UART communication port to the host processor or microcontroller [13]. Figure 4-4 shows the SpeakUp click voice recognition module.

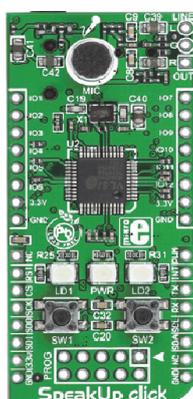


Figure 4-4: SpeakUp click voice recognition module (VR1)

This Speak Up module uses a template matching algorithm which is a kind of pattern recognition. Each voice command is represented as a template set of feature vectors and each command is stored in a separate template. The module requires previous training to be used in VR. The user needs to give voice samples speaking the required commands to be saved in the module memory. This process allows the module to compare the live command with already stored samples. In the present work, each user gives four samples for each voice command. The training process can be conducted directly using buttons on the module board or using special software from the module production company called SpeakUp V1.00. The SpeakUp software is much easier to use and has more options to select the types of VR output such as UART index or GPIO signal with selectable duty cycle. The SpeakUp Click module has to be connected to a PC before starting up the training software and the software will show the connectivity with the module in the GUI. Figure 4-5 shows the software used.

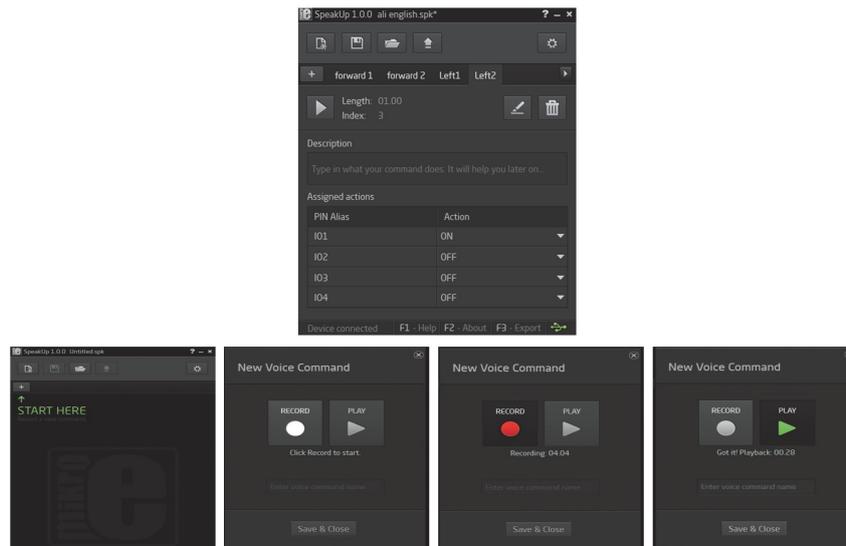


Figure 4-5: SpeakUp software v 1.00

The SpeakUP Click module use three types of communication with the main processor, which are via the UART, USB and programmable GPIO pins. The UART communication protocol communicates with a host processor using the RX and TX pins in the module for receiving and sending commands and data respectively. The UART speed can be set from 4800 to 115200 bps. The module sends a 16-bit index number for each voice command (0-199) via UART as well as via USB. The module has 12 programmable GPIOs that can be used for simple switching purposes. Figure 4-6 shows the communication bus between the SpeakUp Click module and the ARM EFM32 microcontroller.

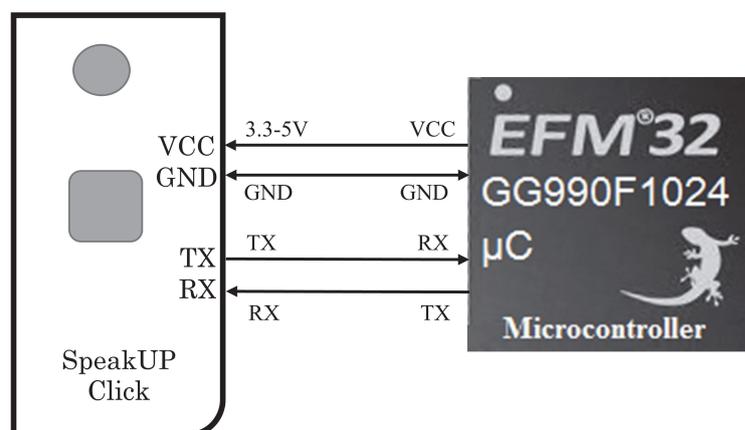


Figure 4-6: SpeakUp Click module communication with microcontroller

4.2.3.2 Easy VR Module (VR2)

The EasyVR (ROBOTECH SRL, Pisa, Italy) is a multi-purpose voice recognition module designed for embedding robust, versatile and low-cost voice recognition capabilities in any application. The EasyVR module can be used with any host with a UART interface powered at 3.3V – 5V, such as ARM, PIC, and Arduino boards. This module is used as the second VR module. This unit can operate in two modes of voice recognition, which are speaker-dependent (SD) and speaker-independent (SI). The SD mode uses the same voice recognition algorithm as in the first module, which is DTW and can be used in any available language. In the speaker-independent SI mode, the system will respond to any user saying a specific command which is included in the command vocabulary library. This mode depends on converting the text command to a reference voice sample and making a comparison with the live voice commands from the user. The Easy VR module can work with only eight global languages (US English, UK English, German, French, Italian, Spanish, Korean and Japanese) in SI mode, due to the availability of command vocabulary libraries for these languages supported by the manufacturer. It can support up to 28 SD commands and up to 32 SI commands, which is enough for the requirements of the present work. The SI voice processing in this module is conducted using a hidden Markov model (HMM) algorithm. Figure 4-7 shows the Easy VR module.



Figure 4-7: Easy VR module

The SD mode needs previous training to store the user's voice samples in the module. Each user trains the module with a minimum of two samples. The training of the Easy VR module is accomplished using special software called EasyVR Commander 3.8.0. Figure 4-8 shows the used software.

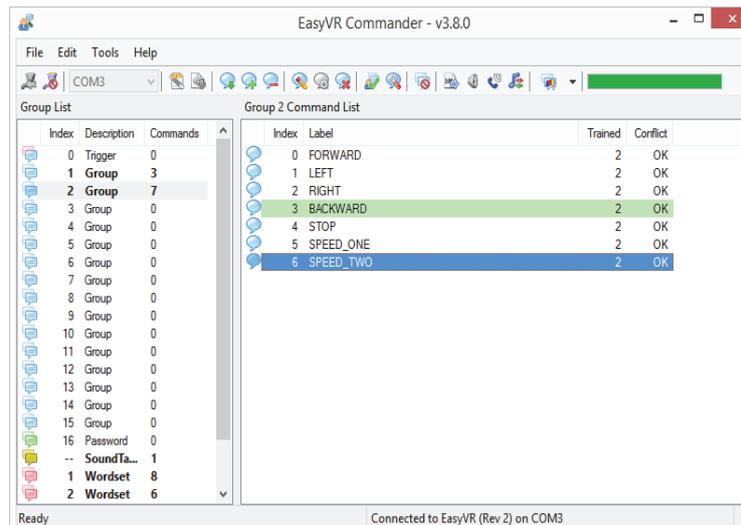


Figure 4-8: Easy VR_Commander 3.8.0 software

The SI algorithm needs to write the wanted and selected control commands as text in the algorithm vocabulary library to allow the module to implement the textual comparison of the selected text command with the live user command, which will further be converted into text. This process is performed using professional software called Quick T2SI™ Lite v_3.1.14 (Sensory Inc., California, USA). The Quick T2SI™ Lite allows the user to enter the control commands as text from the PC keyboard and it converts the text command into an execution file compatible with the Easy VR module. Figure 4-9 shows the software used.

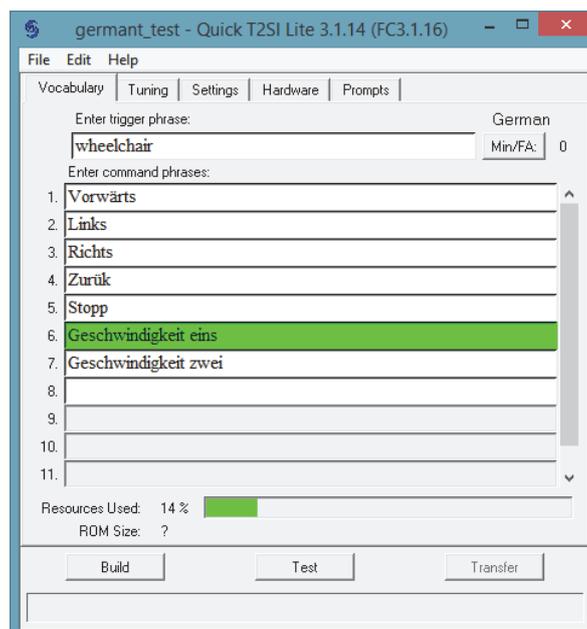


Figure 4-9: Quick T2SI™ lite v_3.1.14 software

The Easy VR module uses the standard UART communication interface to communicate with a host processor or peripheral controller. The standard UART port uses the logical levels 3.3-5 V TTL/CMOS according to the VCC power voltage. The initial UART setting of the Easy VR module is at 9600 baud rate, 1 bit stop, no parity, 8-bit data. The baud rate can be changed after module tuning to a specific value between 9600 and 115200 depending on the host microcontroller UART port's baud rate. Figure 4-10 explains the Easy VR module's communication with the host microcontroller.

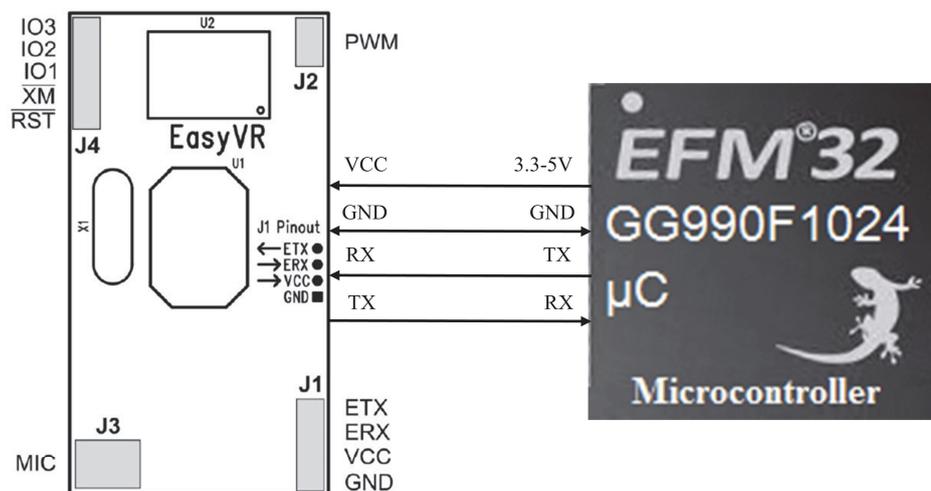


Figure 4-10: Easy VR communication with ARM microcontroller

The communication protocol uses ASCII characters as control or status commands to communicate and implement recognition processes with the host processor which drives the communication process. The characters are sent via the TX line of the Easy VR and received by the RX line of the host microcontroller UART port. The Easy VR module serial interface is software-based and requires a very short delay time (0-9 ms) to transmit the command characters to the host microcontroller. This is a good feature for a fast processor like the ARM EFM32GG990F1024 microcontroller and the delay time can be increased for a slow processor to the range of 10 - 90 ms, and 100 ms - 1 s [53].

4.3 Orientation Detection

The body orientation can be defined as the body's position in relation to true north, to points on the compass, or to a specific place or object [195]. There are two types of body orientation, which are absolute and relative orientation. Absolute orientation means the orientation of the body with respect to the earth and its magnetic field, while relative orientation represents the orientation of the body with respect to a specific place as a reference [196]. Body or object orientation can be measured in different forms such as in Euler angles, quaternions, or linear acceleration. Recently Euler angles and quaternions have been widely used for orientation detection, especially in sensor fusion software for orientation detection modules to represent body orientation. Compared to quaternions, Euler angles are simpler and lend themselves to simple analysis and control. Furthermore, Euler angles are limited by a phenomenon called the "Gimbal Lock". In applications where the sensor will never operate near pitch angles of +/- 90 degrees, Euler angles are an excellent choice [197].

Several companies have produced types of orientation detection modules which can be used for sensing and tracking the movements of the head and system reference orientation. For this purpose, this section concentrates on orientation modules that use more than one sensor to detect body movement (accelerometer, gyroscope, and magnetometer) to achieve better accuracy. Table 4-4 shows the main features of some available orientation sensors.

Table 4-4: Comparison of orientation sensors features

Orientation Dete. module	Sensor			Euler Angles			Bus	Measu. Time	Sample Rate	Price €
	Acc.	Gyr.	Mag.	Pit.	Roll	Yaw				
SQ-SI-360DA*	Y	Y	Y	Y	Y	N	UART	25 ms	40 Hz	50
VN-100**	Y	Y	Y	Y	Y	Y	SPI,USB, UART	2.5 ms	400 Hz	500
9-DOF L3GD20H +LSM303 ***	Y	Y	Y	N	N	N	I ² C	-	-	19
BNO055 ****	Y	Y	Y	Y	Y	Y	UART, I ² C	10 ms	100 Hz	35
UM7 *****	Y	Y	Y	Y	Y	Y	UART, SPI	4 ms	250 Hz	155

* Signal Quest, LLC, New Hampshire, USA

** VectorNav Technologies, LLC, Texas, USA

*** Adafruit, New York, USA

**** Bosch sensortec.Inc., Reutlingen, Germany

***** CH Robotics LLC, Utah, US

4.3.1 Euler Angles

The Euler angles are three angles originally defined by Leonhard Euler to represent the orientation of a rigid body in relation to a global coordinate system [198]. Euler angles provide an approach to describe the 3-D orientation of the body using an aggregate of three rotations about the x-, y-, and z-axes which are called pitch, roll, and yaw. Pitch, roll, and yaw angles represent the orientation of a body's rotation around a principal axis. Figure 4-11 shows the Euler angles axes.

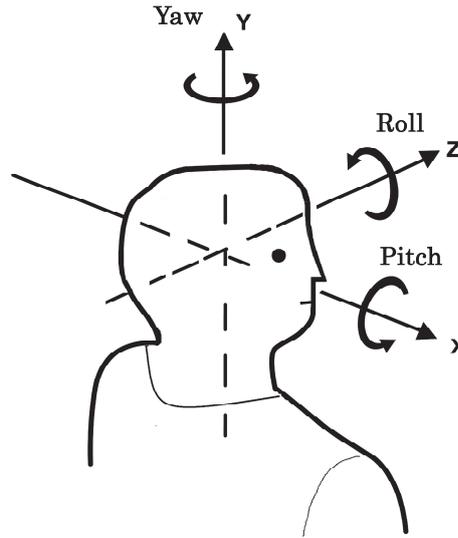


Figure 4-11: Euler angles axes

The rotation of yaw angle around the x-axis is defined as:

$$R_x(y) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos p & -\sin p \\ 0 & \sin p & \cos p \end{bmatrix} \quad (3.22)$$

The rotation of pitch angle around the y- axis is defined as:

$$R_y(p) = \begin{bmatrix} \cos r & 0 & \sin r \\ 0 & 1 & 0 \\ -\sin r & 0 & \cos r \end{bmatrix} \quad (3.23)$$

The rotation of roll angle around the z-axis is defined as:

$$R_z(r) = \begin{bmatrix} \cos y & -\sin y & 0 \\ \sin y & \cos y & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (3.24)$$

The angles p, r, and y represent pitch, roll, and yaw which are the Euler angles [199].

4.3.2 Orientation Detection Sensor BNO055

The BNO055 (Bosch Sensortec, Germany) orientation detection module has been selected to build the head tilt controller. The selection of the module is done based on the sufficient sample rate 100 Hz, the low-cost price of 35 € and the favorable output orientation format of the Euler angles. The head tilt controller consists of two BNO055 orientation detection modules. The BNO055 module includes three MEMS sensors which are an accelerometer, gyroscope, and magnetometer. The three sensors are combined in one chip with a powerful ARM Cortex M0 processor to access data from the three sensors. The BNO055 is a 9-degrees-of-freedom (DOF) module. It can use each MEMS sensor separately or use more than one sensor in the fusion mode, which can give the orientation of the module in different forms like Euler angles, rotation vector, linear acceleration, or quaternion vector. In the present work, Euler angles were used to describe orientation. Figure 4-12 shows the BNO055 orientation sensor module used.

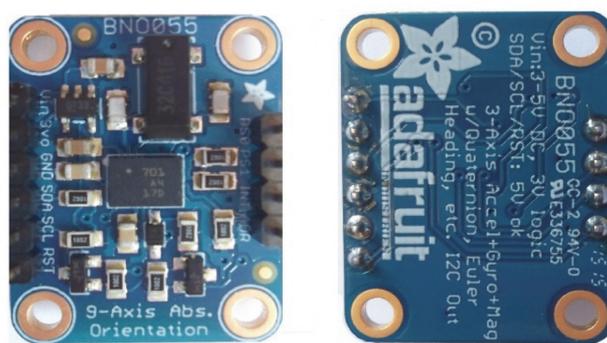


Figure 4-12: Adafruit BNO 055 Orientation Module

The BNO055 is an orientation detection module. It is a system-in-package (SIP) device. Three MEMS sensors have been included in the module, which are a triaxial 14-bit accelerometer, a triaxial 16-bit gyroscope with a wide range of ± 2000 degrees per second, and a triaxial geomagnetic sensor. The three sensors are managed by a 32-bit Cortex M0+ microcontroller in a single package. The corresponding chipsets are integrated into one single 28-pin LGA with a 3.8 mm x 5.2 mm x 1.1 mm housing. The BNO055 has peripheral communication ports including digital bidirectional I²C and UART interfaces. The I²C port can be programmed to run with the HID-I²C protocol which allows the BNO055 to turn into a plug-and-play sensor hub solution for the Windows 8.0 or 8.1 operating system [200].

4.3.2.1 BNO055 Operation Modes

The BNO055 module provides different output forms of orientation, which can be selected by choosing the appropriate operation mode. Each of the MEMS sensors in the BNO055 can be used individually or with another sensor depending on the task required. In the current system, the three MEMS sensors have been used to find the absolute orientation.

Figure 4-13 shows the orientation output form of the BNO055 module.

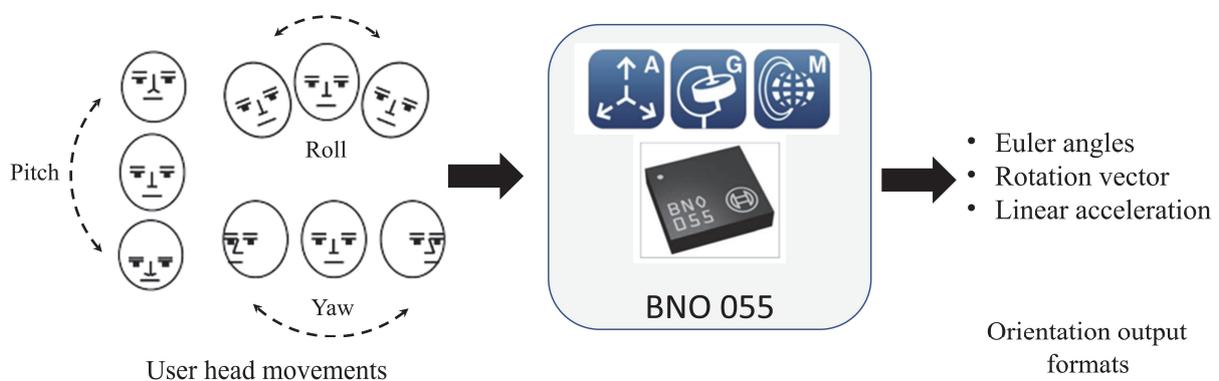


Figure 4-13: BNO055 output forms

Table 4-5 explains the available operation modes, the activated sensors and the type of orientation measurement of the BNO055 orientation module. The table shows that body orientation detection can be achieved in absolute or relative orientation by using sensor fusion mode. For the current application, a nine degrees of freedom (NDOF) mode has been used to represent the user's head and system reference orientation. The NDOF means that the device can measure the body orientation in 9 axes (3 accelerometers, 3 gyroscopes, and 3 magnetometers) which can measure the body's motion in any direction of movement, where a degree of freedom (DOF) is an independent direction in which motion can occur. The sampling rate of this mode is 100 samples/sec, which covers the system sample rate requirements for the sensor's measurement of head tracking (> 50 sample/s).

Table 4-5: BNO055 operation modes [200]

Operating Mode		Available sensor signals			Fusion Data	
		Accel	Mag	Gyro	Relative Orientation	Absolute Orientation
Configuration Mode		-	-	-	-	-
Non-fusion modes	ACC. ONLY	X	-	-	-	-
	MAG. ONLY	-	X	-	-	-
	GYRO. ONLY	-	-	X	-	-
	ACC-MAG	X	X	-	-	-
	ACC-GYRO	X	-	X	-	-
	MAG-GYRO	-	X	X	-	-
	AMG	X	X	X	-	-
Fusion modes	IMU	X	-	X	X	-
	COMPASS	X	X	-	-	X
	M4G	X	X	X	X	-
	NDOF_FMC_OFF	X	X	X	-	X
	NDOF	X	X	X	-	X

4.4 Test Platforms

The system has been designed to be used for wheelchairs and rehabilitation robotics. Various different methods can be used to communicate with wheelchairs and robots. Most modern wheelchairs use the CAN bus for communication between the joystick and motor driver controller and also with light controllers. Another communication bus can also use UART or SPI and so on. The proposed system has been tested on two different platforms using the UART communication bus because it is available on both platforms as well as on the microcontroller side. The first platform is the Jaguar-Lite robot (Dr. Robot Inc., Canada) and the second platform is the Meyra Ortopedia Smart 9.906 (MEYRA GmbH, Germany). The system was adapted to the motor driver units of both platforms.

4.4.1 Dr. Jaguar Lite Robot

The Jaguar-Lite robot (Dr. Robot Inc., Canada), is a type of mobile robot which has no fixed place or a limited space of work and is able to move in the required operating environments. The robot has two wheels, and its concept of control and the construction of the motor are similar to those of a wheelchair. The present control system interfaces with the Jaguar-Lite robot using the UART

communication port. The ARM microcontroller sends the control commands to the BMS5005 robot controller. A Sabertooth 2x25 motor driver unit realizes the final motor control action. The robot has two DC motors operated at 24 V and a rated current of 7.5 A. Figure 4-14 shows the Jaguar-Lite robot.

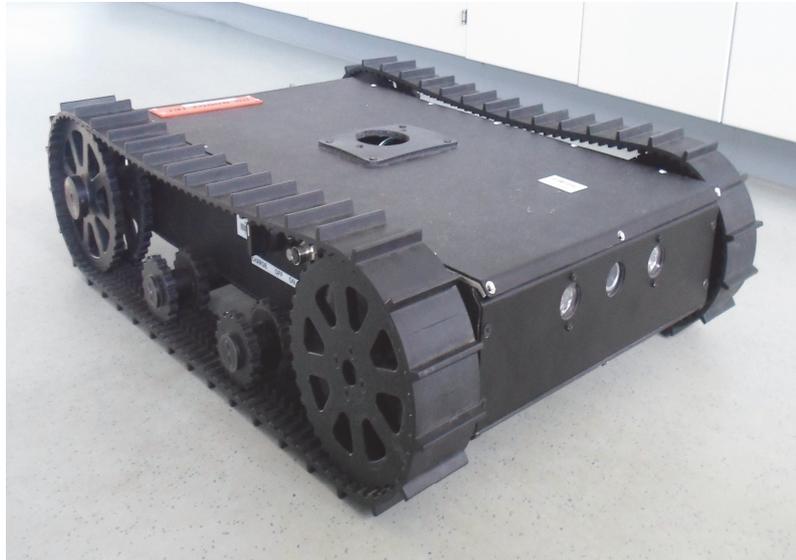


Figure 4-14: Jaguar-Lite robot

4.4.1.1 Jaguar Lite Essential Features

There are several important features of the Jaguar-Lite robot which make it a good selection as a test platform. It is a rugged and reliable mobile robot platform for indoor and outdoor applications with premium handling and stability. It can operate in in different harsh terrains indoors and outdoors. This robot has a shield for water and weather resistance and it can climb slopes up to 45° stairs of a maximum 180 mm in height. It has autonomous navigation with outdoor GPS with an integrated high-resolution video camera with audio. The robot is lightweight at approximately 14 Kg with a large payload capacity.

4.4.1.2 Jaguar Robot Structure

The internal structure of the Jaguar Lite robot is shown in Figure 4-15. All the sensors, modules and units are connected to the main motion and sensing PMS5005 controller. The main components of the Jaguar are as follows:

- The motor driver unit is an H-Bridge 2-channel type and has an input operating voltage of 6~24V and 30V absolute max, and a maximum input

current up to 25A continuous power for each channel with a peak up to 50A for a few seconds. This unit is responsible for converting the microcontroller's low power control signals into sufficient power to be fed to the DC motors. It also controls the speed and direction of the motors.

- The motion and sensing controller represents the brain of the robot and works with 5V operating voltage. It has 6 PWM output, and three motor control modes of PWM control, velocity control, and position control.
- The video camera works with 5V operating voltage. It has a 4.4mm lens with 47° horizontal view, F2.0, fixed iris and fixed focus. It has (640*480) to (160*120) resolution. It has a frame rate of 264:30 at all resolutions.
- The GPS works with 5V operating voltage and a 5Hz update rate. It has a minimum sensitivity of -185 dB.

The Jaguar robot has other components with the optional selection. Figure 4-15 shows the internal structure of the Jaguar-Lite robot, more information available elsewhere [201].

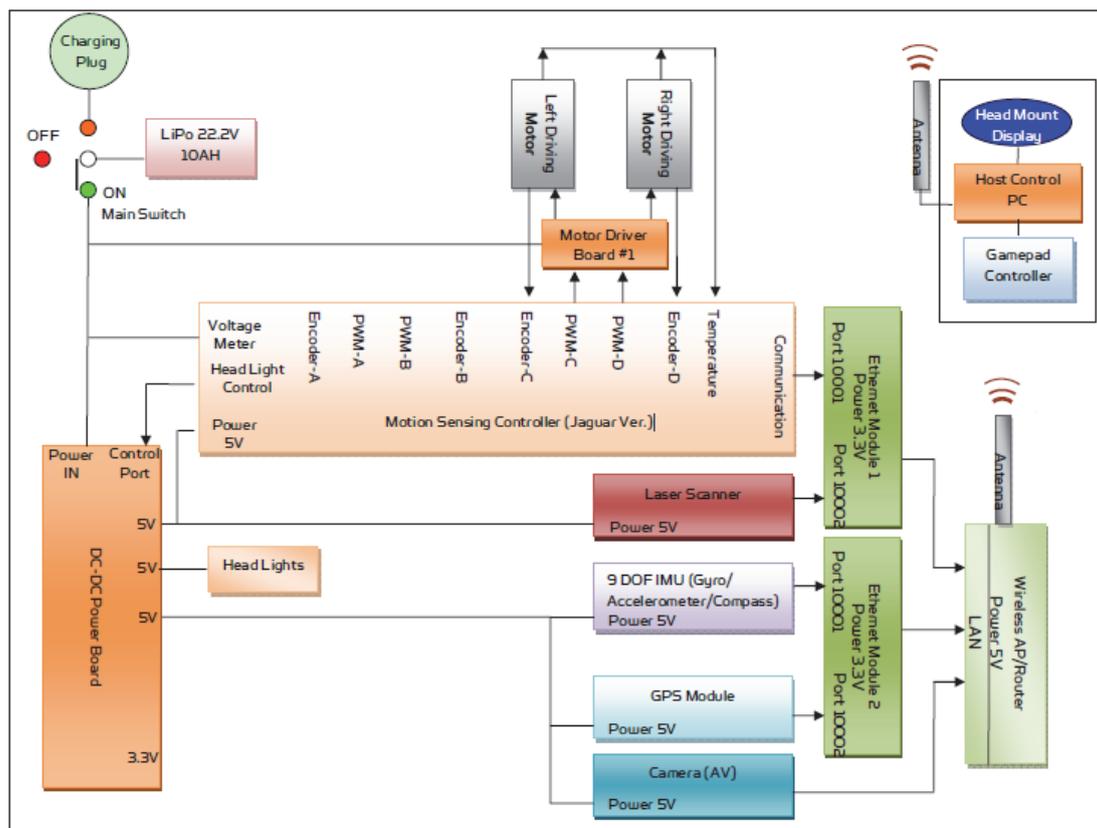


Figure 4-15: Jaguar-Lite robot internal structure [201]

4.4.1.3 Jaguar Robot Communication Protocol

In the present work, the communication with the Jaguar-Lite robot is implemented using a UART communication port with settings of 115200 kbps, N, 8, 1. The microcontroller sends control commands using a special communication protocol. Each command is sent as a packet of data as follows:

STX	RID	Reserved	DID	LENGTH	DATA	CHECKSUM	ETX
-----	-----	----------	-----	--------	------	----------	-----

Where:

1. Start-of-transmission indicator (STX)
2. Robot ID (RID)
3. Reserved byte for future use
4. Data ID (DID)
5. A length field to specify the total number of bytes in the Data field
6. Data, if any
7. A checksum
8. An end-of-transmission indicator (ETX)

The robot receives the control packet as an array, and it recognizes the start of a new control command packet by the STX and recognizes the end of the control packet by the ETX. The robot ID is used to differentiate among a group of robots, which is not applicable to the current application. The data ID is used to recognize the type of control data; for example, to control the robot motor with PWM the DID No. 5 needs to be used, and for controlling the motor's specific position DID No. 3 should be used. More details about this protocol can be found elsewhere [202].

4.4.1.4 Communication Security

The cyclic redundancy check (CRC) algorithm has been used for communication security. CRC is used especially in digital network to detect any errors which can occur during data transition between the sender (the ARM microcontroller) and the receiver (the Jaguar-Lite robot). The use of a CRC checksum is very important for the present study due to the application being used for disabled people or for rehabilitation and medical applications where user safety is given higher priority. The CRC algorithm used is the same one as that used by the Jaguar Lite robot's PMS5005 motherboard to communicate with the host processor via a UART port. The CRC prevent the implementation of any control command if there is any change in the data received compared to that sent, and

thus it will be guaranteed that all the data on the receiver side is the same as that in the sender side. When any error is detected, the receiver will send a negative acknowledgment to the sender to resend the corrupted message again. Figure 4-16 shows the CRC checksum used.

```
1322 |
1323 | unsigned char CRC(char *lpBuffer, const int nSize)
1324 | {
1325 |     unsigned char shift_reg, sr_lsb, data_bit, v;
1326 |     int i, j;
1327 |     unsigned char fb_bit;
1328 |
1329 |     shift_reg = 0; // initialize the shift register
1330 |
1331 |     for (i = 0; i<nSize; i++)
1332 |     {
1333 |         v = (unsigned char)(lpBuffer[i] & 0x0000FFFF);
1334 |         for (j = 0; j<8; j++) // for each bit
1335 |         {
1336 |             data_bit = v & 0x01; // isolate least sign bit
1337 |             sr_lsb = shift_reg & 0x01;
1338 |             fb_bit = (data_bit^sr_lsb) & 0x01; // calculate the feed back bit
1339 |             shift_reg = shift_reg >> 1;
1340 |             if (fb_bit == 1)
1341 |                 shift_reg = shift_reg ^ 0x8C;
1342 |             v = v >> 1;
1343 |         }
1344 |     }
1345 |     crc = shift_reg;
1346 |     return shift_reg;
1347 |
1348 | }
```

Figure 4-16: CRC checksum

4.4.2 Meyra Ortopedia Smart 9.906 Wheelchair

The Meyra Smart 9.906 wheelchair is used as the second test platform. It has two types of controller. The first is a motion controller which is implemented using a joystick module. The second is the light controller which uses a button on the control dashboard. The joystick and light buttons communicate with the power motion and light controller using the CAN bus communication protocol. Two types of CAN bus are used in the wheelchair controller. The first is the SAE J2411 single-wire CAN bus, which is used for motion control. It uses only a single wire with ground to communicate with other nodes on the bus. It works with low requirements regarding bit rate and bus length (33.3 kbit/s, 83.3 kbit/s in high-speed mode for diagnostics) and is used widely in comfort electronics networks in motor vehicles [203]. The second CAN protocol is the low-speed CAN bus (iso11898-3) protocol which is used for light control. It is used widely in the automotive industry and consists of two twisted wires called CAN_H and CAN_L. The operating bit rate for the wheelchair CAN bus is 35.715 kbit/s. Figure 4-17 shows the Meyra Smart 9.906 wheelchair.



Figure 4-17: Meyra Smart 9.916 wheelchair

4.4.2.1 Wheelchair Motors

Two DC motors SRG0531 (AMT Schmid GmbH & Co. KG, Germany) have been used to drive the wheelchair. It has a gear box including a helical gearset on the input side and spur gear set on the output side. It works with 24 V operating voltage with a rated power from 220 to 400 Watt. It has a total load capacity up to a maximum of 160 kg and output speed from 105 to 200 rpm (revolutions per minute). The motor operates with low-level noise less than 58 dB. It has an electromagnetic brake working with 12/24 V. Figure 4-18 shows the SRG0531 wheelchair motor [204].



Figure 4-18: SRG0531 wheelchair motor

4.4.2.2 Motor Driver Unit

The original CAN bus-based Meyra motor driver has been replaced with a new Sabertooth 2*32 Amp motor driver unit (Dimension Engineering LLC., USA; see Figure 4-19). This is necessary for the current system because the original CAN controller is not open source, and the manufacturer would not allow us to change the original program. It has two main power channels that can drive two motors with up to 32 A per motor with a maximum peak current of 64 A per channel. It has another two auxiliary power channels with up to 8 A that can be used for electromagnetic brakes or for controlling a medium power load. The Sabertooth motor driver can be connected to any DC power supply from 6 to 33.6 V. In the present work, two 12 V lead-acid batteries have been used to feed the motor driver unit. The two batteries are connected in series to supply 24 V, which is the voltage required for the system motors. The Sabertooth has a 5-volt power terminal that can be used for feeding a microcontroller or host processor that is used to control the motor driver without the need for an external power source. More details can be found in [205].



Figure 4-19: Sabertooth 2x32 Amp motor driver

4.4.3 Speed Feedback Unit

The speed feedback unit consists of two quadrature magnetic encoders fixed onto the shaft of the wheelchair and robot motors. The magnetic encoder senses the motor's rotation and gives the output in the form of square pulses for each rotation. The operation of the encoder used is based on the Hall Effect phenomenon. Each rotation cycle generates 320 square pulses. Each encoder has

two channels (A and B) with a 90° phase shift between them. The magnetic encoder can be used to measure the speed and position of the wheelchair or robot. Figure 4-20 shows an oscilloscope screenshot of the two magnetic encoders when the wheelchair starts moving.

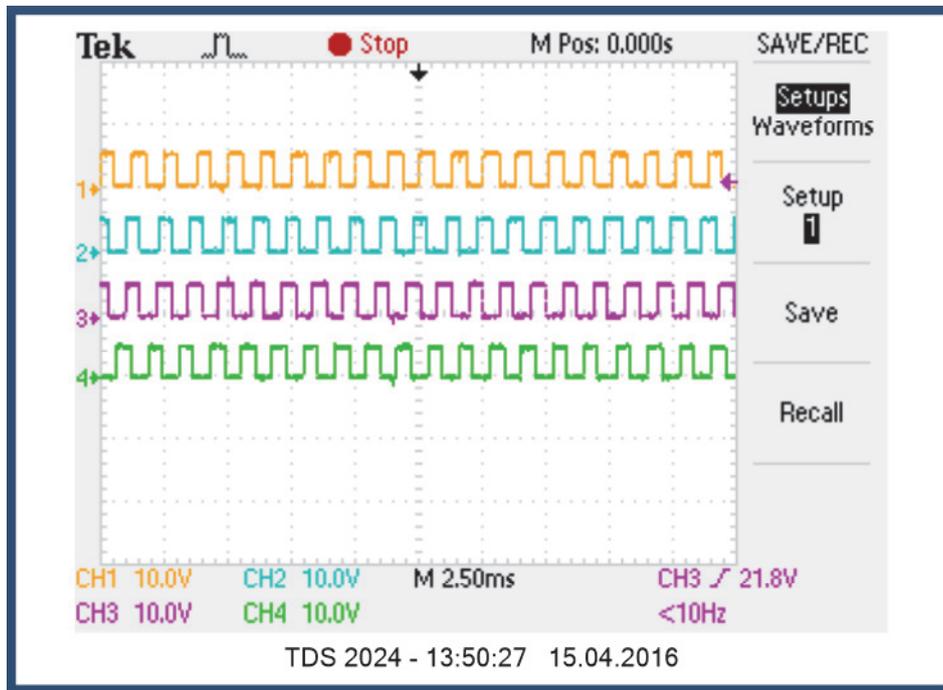


Figure 4-20: Magnetic encoder output signal

5 - Voice Controller

5.1 Realization

The voice controller is one of the optimal solutions to help people with upper and lower limb disabilities. In the present work, two embedded voice recognition modules have been selected to build a novel, strong, and low-cost voice controller. Figure 5-1 shows the voice controller structures. The main unit in this figure is the ARM EFM32 microcontroller, which works as the intelligent core of the system. The microcontroller manages the operation of the system. It receives data from the two voice recognition modules and other input units and sensors. It processes the incoming information, makes the required decisions and sends them as control commands to the motor driving unit and other output units.

A small clip-on boundary microphone with an impedance of 1 k Ω and operating frequencies of 50-13000 Hz has been used to pick up voice commands. The microphone is fixed on a headset that is also used to house the head tilt controller. It operates as a transducer by converting the sound wave (the user's voice) into an electrical signal which is then sent to the VR1 and VR2 voice recognition modules.

The VR1 module can communicate with the host EFM32 microcontroller via UART or GPIO ports. As the VR1 operates only in SD mode, it requires prior training by the users. The live voice commands are compared inside the module with previously trained and stored commands using template matching techniques. The VR results are sent to the microcontroller which retains it as an integer variable. Each command has different values. When the user gives a different command, the recognition results come with a different value; for example, the command “forward” has an index value of 00, and the command “left” has a value of 01 and so on. The microcontroller is programmed to keep listening to the VR1 for any change in the given voice command.

The VR2 module communicates with the microcontroller using only a UART communication port. It is connected to the ARM microcontroller at the UART0 communication port, which consists of two GPIO pins, PE0 and PE1, in location 1 of the μ C pins. This module can work in both SD and SI voice recognition modes. The SD mode requires previous training and works in the same way as VR1. For the SI mode, the desired voice commands need to be selected or written as texts to

the command vocabulary library. The choice of SD and SI modes is performed by the host microcontroller. The recognition results of the VR2 module will be saved as another character variable in the EFM32 microcontroller. Each command from the VR2 is sent via the UART as a character. The microcontroller will carry on listening to the VR2 module via the UART port to check if there are any changes in the recognition results. The microcontroller uses the voice recognition results of VR1 and VR2 which have been stored in the given variables to implement the required control commands and also to check if there are any FP or FN errors.

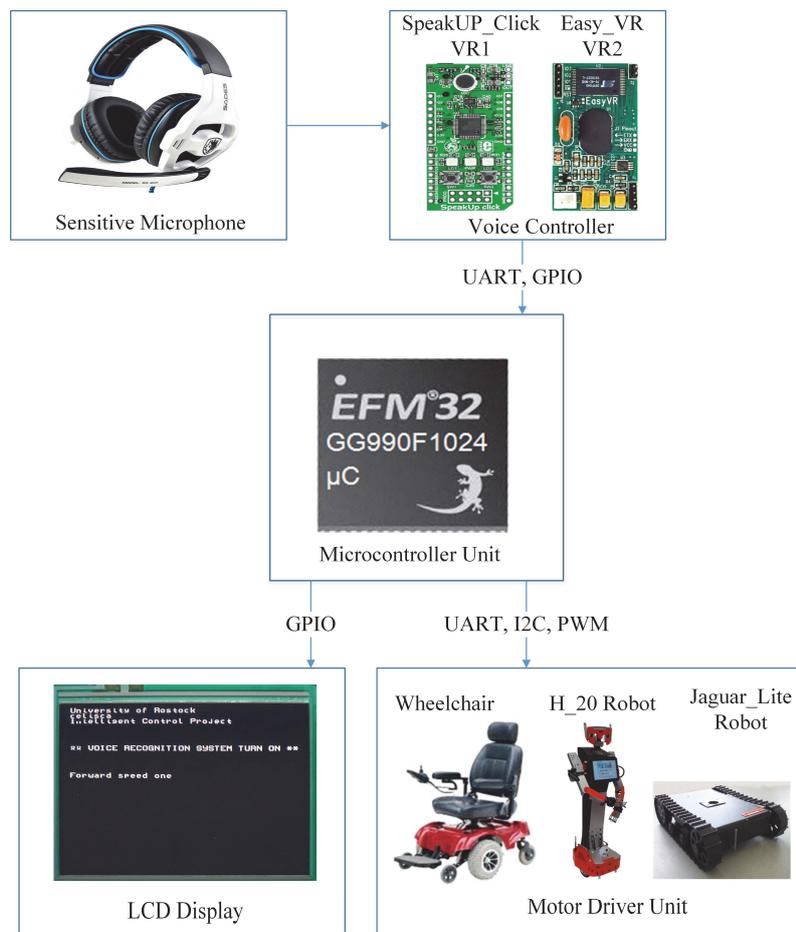


Figure 5-1: Voice controller structure

The EFM32 microcontroller has been programmed to send a specific data packet for each control command depending on the action required and also depending on the test platform used. The data in each packet changes depending on the type of control commands, which may be motion commands such as forward or backward or other general commands for turning on/off other components such as lights or signals.

The voice controller was tested with two platforms, which are the Jaguar-Lite robot and the Meyra Smart 9.906 electrical wheelchair. Each uses different control packets for the motor driver unit. Thus, the commands need to change depending on the motor driver unit used. The motion commands (forward, backward, left, right, stop, speed one and speed two) have fixed values of speed for each test platform depending on their maximum speeds. The maximum speed for the Jaguar-Lite robot is 8.5 km/h while it is 6 km/h for the Meyra wheelchair. When the voice controller is activated as the main controller, the system uses a discrete speed control with specific speed values for each sound command. For example, if the user gives the “Forward” command, the microcontroller will send a specific speed value (2.5 Km/h) to the motor driver unit. To increase the speed, the user needs to give another voice command, which is “Speed one”, and for the maximum speed for the voice controller, the user needs to use another voice command of “Speed two”. Figure 5-2 explains the control of the Jaguar-Lite robot's speed using the voice controller.

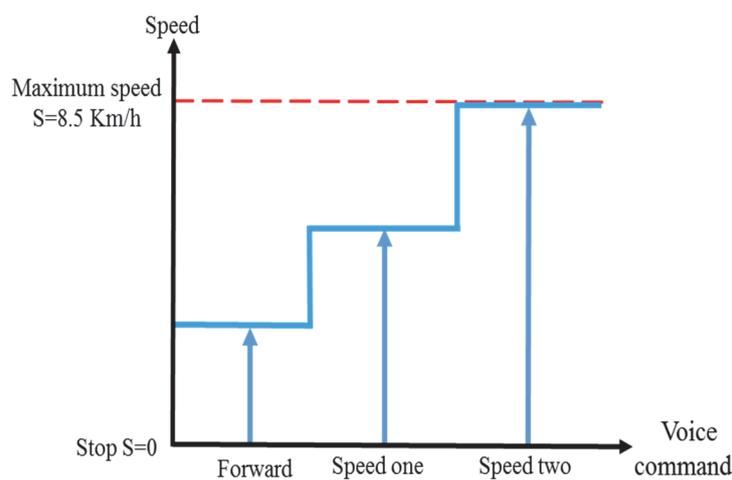


Figure 5-2: Motors speed control for voice controller

The microcontroller is programmed to send another packet of data to the LCD display to show the system status. The LCD display helps the user in case of errors or problems in the system and also shows indications and suggestions to the user to avoid error situations. For example, if the wheelchair faces obstacles, the microcontroller informs the user via LCD by showing the message “Obstacle detected, please go back”. The microcontroller sends another indication when it cannot recognize the voice command and shows the message “Say again please!”. The microcontroller shows which is the active controller and the current control command, which also helps the user to check whether the system is responding correctly. Figure 5-3 shows a sample view of the LCD display.

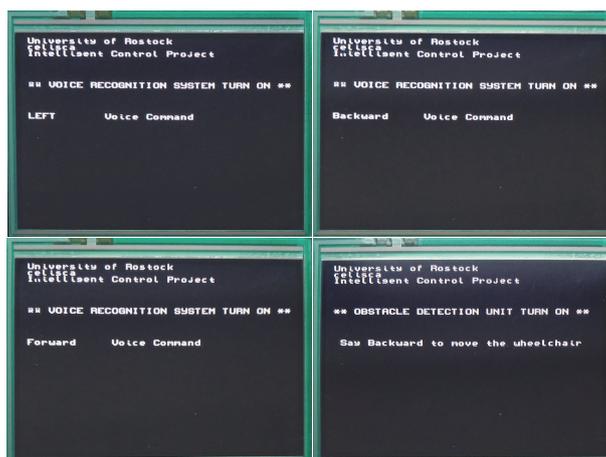


Figure 5-3: System status notification on LCD Display

5.2 Types of Voice Commands

There are two groups of voice commands used in the proposed system. The first is the motion control commands and the second the general control commands. The motion control commands are used to control the movement of the wheelchair and are active only when the user selects the voice as a motion controller and deactivated when the user selects the head tilt as a motion controller. The second group is the general control commands used for controlling other features of the wheelchair such as lights, signals, and sound alarm. It is also used to control the tilt to speed ratio of the head tilt controller. Table 5-1 explains the voice commands used with short descriptions.

Table 5-1: System voice commands

Command	Definition
Forward	Both wheelchair motors rotate forward
Left	Right motor rotates forward and the left rotate backward
Right	Left motor rotates forward and the right rotate backward
Backward	Both motors rotate backward
Stop	Both motors stop
Speed one*	Increase the speed or decrease speed two
Speed two**	Increase the speed
Light on	Turn lights on
Light off	Turn light off
Right signal	Turn right signal on
Left signal	Turn left signal on
Signal off	Turn signal off

*, ** these commands are used also to change the user's head tilt to speed ratio

5.3 Evaluation

The VR unit consists of the two VR modules VR1 and VR2. The VR1 module has one VR algorithm, which is SD-DTW. The VR2 module has two VR algorithms, which are SD-DTW and SI-HMM. The three VR algorithms of the VR1 and VR2 modules were tested first individually to check their performance in environments with two different noise levels.

The algorithms were tested with the three languages English, German, and Chinese. Each algorithm was tested with 1,260 voice samples at two different noise levels. The test environment was selected carefully based on levels of noise. The first place was a low noise environment with a maximum noise of 42 dB, whereas the second test place was a noisy laboratory with 72 dB of noise. Seven control motion commands have been used in the tests. Each command was repeated ten times by each user in each test environment. The tested voice commands are forward, backward, left, right, stop, speed one and speed two.

The following Table 5-2, Table 5-3, and Table 5-4 show the average accuracy of the VR modules at the two noise levels. Each table includes the tests in three languages. The symbol P represents the number of persons who performed the test; each person trained each voice command ten times for the system. The abbreviation Acc. represents the percentage of correct voice recognition.

Table 5-2: SD tests for VR1

Command	English VR Acc.			German VR Acc.			Chinese VR Acc.		
	P	42 dB	72 dB	P	42 dB	72 dB	P	42 dB	72 dB
Forward	7	91.4%	95.7%	7	91.4%	95.7%	4	100%	95.0%
Left	7	98.5%	94.2%	7	91.4%	90.0%	4	100%	85.0%
Right	7	97.2%	92.8%	7	97.2%	94.2%	4	90.0%	85.0%
Backward	7	94.2%	95.7%	7	87.1%	87.1%	4	97.5%	100%
Stop	7	98.5%	97.2%	7	91.4%	91.4%	4	97.5%	90.0%
Speed one	7	97.2%	88.5%	7	88.5%	87.1%	4	82.5%	95.0%
Speed two	7	94.2%	90.0%	7	94.2%	94.2%	4	85.0%	92.5%

Table 5-3: SD tests for VR2

Command	English VR Acc.			German VR Acc.			Chinese VR Acc.		
	P	42 dB	72 dB	P	42 dB	72 dB	P	42 dB	72 dB
Forward	7	90%	91.4%	7	94.2%	91.4%	4	95%	97.5%
Left	7	95.7%	78.5%	7	98.5%	80.0%	4	92.5%	80.0%
Right	7	98.5%	90.0%	7	100%	90.0%	4	100%	87.5%
Backward	7	98.5%	91.4%	7	97.2%	74.2%	4	100%	95.0%
Stop	7	100%	87.1%	7	100%	84.2%	4	100%	67.5%
Speed one	7	94.2%	85.7%	7	91.4%	82.8%	4	100%	80.0%
Speed two	7	97.2%	77.1%	7	87.1%	72.8%	4	92.5%	90.0%

Table 5-4: SI –HMM tests for VR1

Command	English VR Acc.			Command	German VR Acc.		
	P	42 dB	72 dB		P	42 dB	72 dB
Forward	7	87.1%	82.8%	Vorwärts	7	87.1%	82.5%
Left	7	91.4%	87.1%	Links	7	90%	87.1%
Right	7	97.2%	77.1%	Rechts	7	82.8%	77.1%
Backward	7	82.8%	67.1%	Zurück	7	75.7%	57.1%
Stop	7	80%	65.7%	Stopp	7	87.1%	67.1%
Speed one	7	87.1%	78.2%	Gesch. eins	7	88.5%	82.8%
Speed two	7	91.4%	82.8%	Gesch. zwei	7	90%	85.7%

The tests of the VR1 and VR2 algorithms show that the SD-VR1 algorithm shows good stability against the change in noise levels with a voice recognition accuracy of 93.66% and 92.2% for 42 to 72 dB noise respectively, where the stability of the system is the stability of output against the change of input. It also shows a VR error rate of 6.34% including 0.00% FP errors and 6.34% FN errors at 42 dB noise. The VR errors increased slightly at 72 dB noise to 7.8%, including 0.87% of FP errors and 6.93% of FN errors.

The SD-VR2 algorithms tests show good performance at low-level noise with VR accuracy of 96.23% and VR errors of 3.77% including 0.07% FP errors and 3.7% FN errors. The VR accuracy of the SD-VR2 was reduced when the noise increased to 72 dB at 84.50% and the VR errors increased to 15.62% including 0.07% FP errors. The SD-VR2 tests show the lowest FP error rate of 0.07% at the 42 and 72

dB noise levels compared to the other algorithms. The VR2- SI algorithm tests show the lowest VR accuracy of 87.10% at 42 dB and 72 dB noise and also the highest VR errors for both FP and FN errors compared with VR1-SD and VR2-SD. Figure 5-4 and Figure 5-5 show the VR accuracy and VR errors of the algorithms tested.

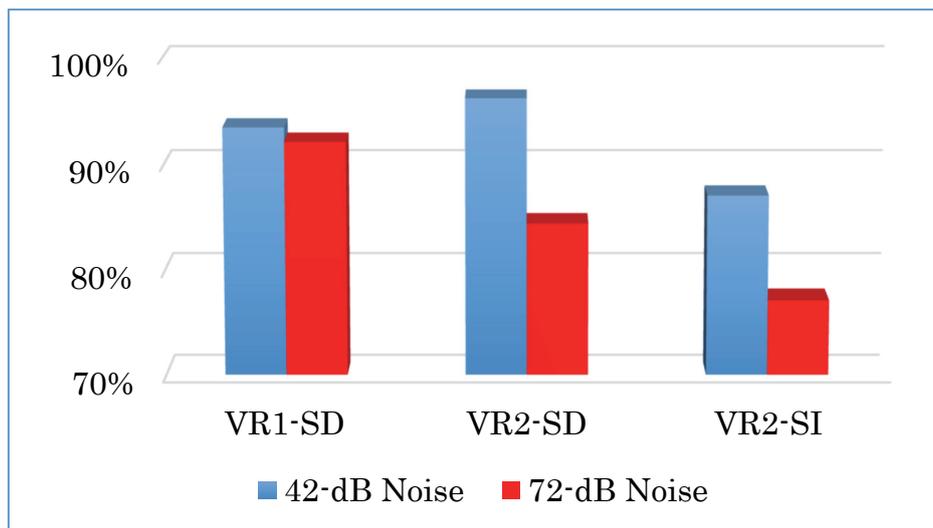


Figure 5-4: VR accuracy for VR1 and VR2

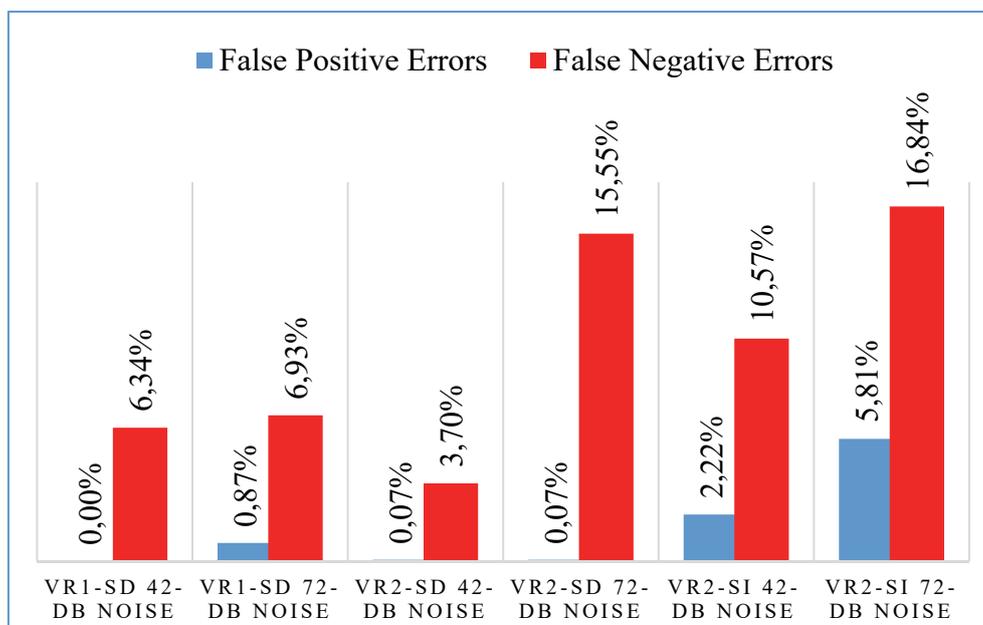


Figure 5-5: VR errors for VR1-SD, VR2-SD, and VR2-SI

The limited performance of the VR2-SI algorithm makes it not the best choice for the proposed system or for rehabilitation applications, but it is still required for applications where many users use the system and give the same commands.

As the system is designed to be used for wheelchair and rehabilitation applications, it is required that only the system owner can control it using voice commands. The speaker-dependent (SD) algorithm is the best choice for current work. VR1-SD and VR2-SD were selected to implement the tests of the voice controller. Two logical algorithms have been used to combine the VR1-SD and VR2-SD so as to increase VR accuracy and reduce VR error. The first algorithm uses the “OR” logical condition between the two modules, which means that the system will implement the control commands when any one or both of the VR1 and VR2 modules recognize the voice command. The second algorithm uses the “AND” logical condition to implement the voice command, which means that both VR modules VR1 and VR2 must recognize the given commands correctly. If only one module recognizes them or both modules generate different recognition results for the same command (FP error), the system will not implement the given command. Figure 5-6 explains the “OR” and “AND” algorithms flowcharts.

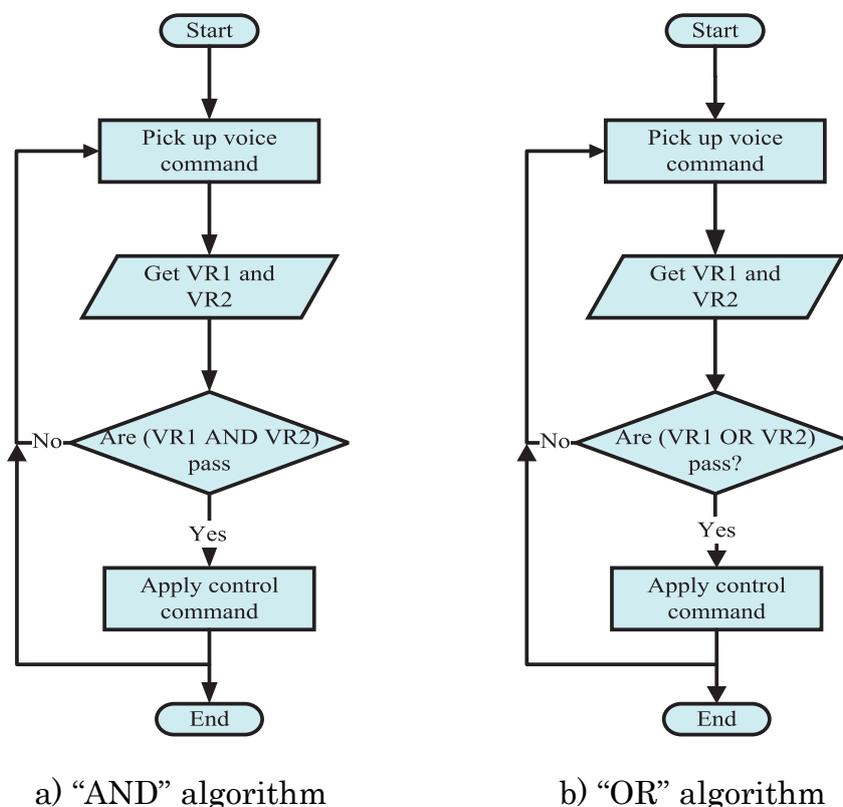


Figure 5-6: Flowcharts for “AND” and “OR” algorithms

The results are shown in Table 5-5 and Table 5-6 indicate that the “OR” algorithm has the highest VR accuracy at both 42 dB and 72 dB with $\approx 98.20\%$ and $\approx 94.5\%$ respectively. The VR error rate for “OR” algorithm at 42 dB noise is 1.82%, including 0.87% FP errors and 0.95% FN errors, whereas the error increased to 5.20% including 0.87% FP errors and 4.36% FN errors in the environment with 72 dB of noise.

The “AND” algorithm tests show that it has the lowest FP error in both 42 dB and 72 dB environments at 0%, which is a very good feature for the current work and rehabilitation applications. Unfortunately, the FN error rate for “AND” is also the highest at 12.61% with 42 dB noise and 24.9% with 72 dB. The VR accuracy of “AND” is strongly affected by increasing noise and is reduced from $\approx 87.40\%$ at 42 dB to $\approx 74.20\%$ at 72 dB of noise.

Table 5-5: OR tests for VR1-SD and VR2-SD

Command	English VR Acc.			German VR Acc.			Chinese VR Acc.		
	P	42 dB	72 dB	P	42 dB	72 dB	P	42 dB	72 dB
Forward	7	95.7%	92.8%	7	92.8%	95.7%	4	87.5%	97.5%
Left	7	100%	95.7%	7	98.5%	97.2%	4	100%	100%
Right	7	100%	100%	7	100%	100%	4	100%	95.0%
Backward	7	100%	97.2%	7	98.5%	84.2%	4	100%	97.5%
Stop	7	100%	97.2%	7	100%	92.8%	4	100%	77.5%
Speed one	7	97.2%	98.5%	7	94.2%	98.5%	4	100%	77.5%
Speed two	7	100%	100%	7	98.5%	90.0%	4	100%	100%

Table 5-6: AND tests for VR1-SD and VR2-SD

Command	English VR Acc.			German VR Acc.			Chinese VR Acc.		
	P	42 dB	72 dB	P	42 dB	72 dB	P	42 dB	72 dB
Forward	7	85.7%	84.2%	7	90.0%	90.0%	4	75.0%	70.0%
Left	7	94.2%	84.2%	7	92.8%	81.4%	4	100%	57.5%
Right	7	94.2%	87.1%	7	94.2%	87.1%	4	92.5%	75.0%
Backward	7	97.2%	87.1%	7	88.5%	61.4%	4	97.5%	95.0%
Stop	7	97.2%	72.8%	7	82.8%	74.2%	4	100%	57.5%
Speed one	7	71.5%	75.7%	7	75.7%	67.1%	4	70.0%	55.0%
Speed two	7	91.4%	65.7%	7	75.7%	57.1%	4	70.0%	72.5%

Figure 5-7 and Figure 5-8 show the VR accuracy and the VR error rate results of the (OR) and (AND) algorithms in the two different noise levels.

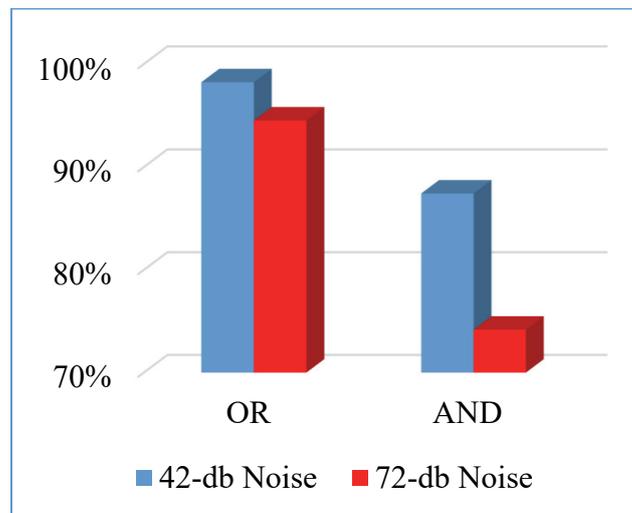


Figure 5-7: “AND” and “OR” VR accuracy

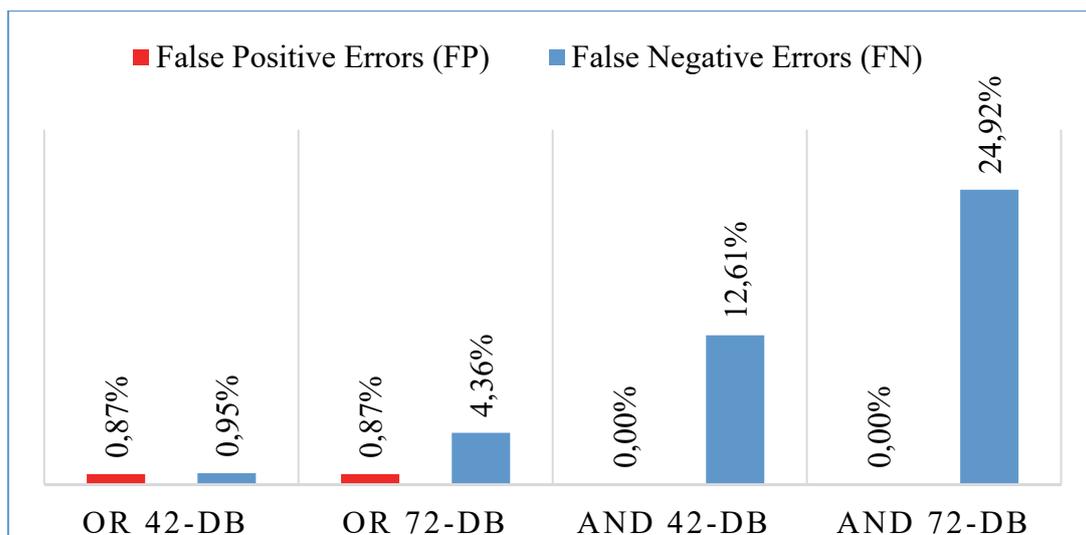


Figure 5-8: “AND” and “OR” VR error rate

The test results revealed that the “OR” algorithm exhibits better performance at the two noise levels compares with the “AND” algorithm. At the same time, the “AND” algorithm has the lowest FP error rate at 0% compared with less than 1% for the “OR” algorithm. For the current system, the OR algorithm has been selected to implement the voice controller. An FP cancellation function has been added to the “OR” algorithm to reduce FP errors and is called the OR-FP. This function will check if there is a difference in VR results (for example, when the first module recognizes “left” and the second module recognizes “right”) from VR1 and VR2. In this case, the function will cancel the VR result. Thus, the

FP errors will be converted to FN errors, which will increase the safety condition of the system. Figure 5-9 shows the new modified OR algorithm flowchart with the FP cancellation function.

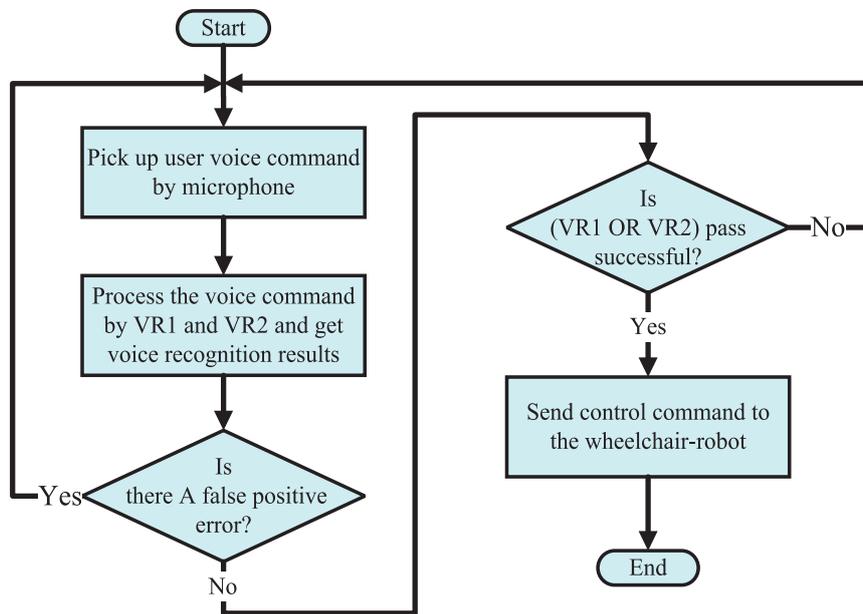


Figure 5-9: Flowchart for OR algorithm with FP cancellation function

Figure 5-10 shows the new VR error results for “OR” algorithm before and after FP error cancellation function. It can be clearly seen that the FP error rate of the OR-FP algorithm becomes 0% for the test of current samples, which is an ideal situation for the system. FP errors can still occur when only one module recognizes the voice command, but incorrectly. Most FP errors can be significantly reduced by adding a third VR module to the system.

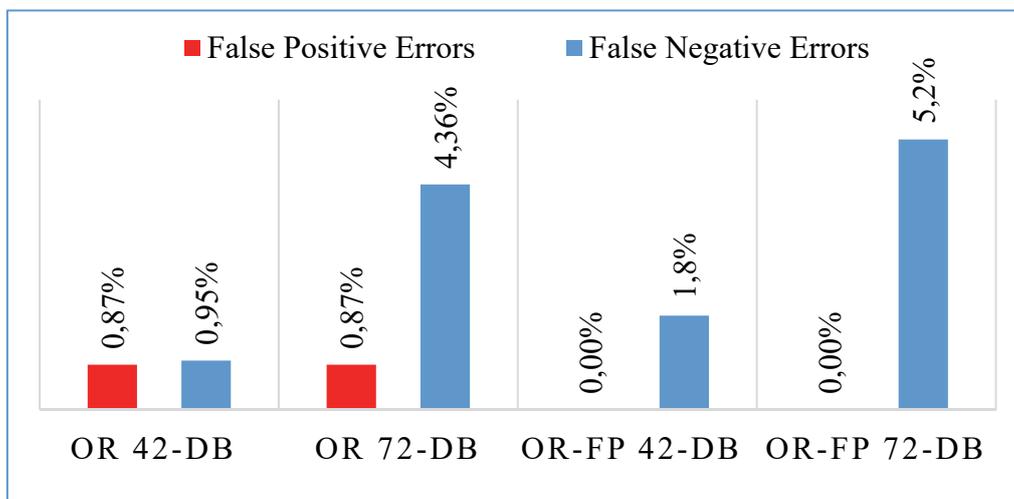


Figure 5-10: VR errors rate for OR-FP cancelation

5.4 Conclusion for the Voice Controller

In this chapter, a multi-mode voice controller for rehabilitation purposes has been presented. The system aims to help quadriplegic, handicapped, elderly and paralyzed patients to control a robotic wheelchair using voice commands instead of a traditional joystick controller. The system design takes into consideration future needs for modifications or expansions where new control units could be added using the GPIO, I²C and UART ports of the ARM EFM32GG990F1024 Microcontroller. The results show acceptable improvements in VR accuracy using the OR-AND algorithms compared to the use of an individual algorithm (SD-VR1, SD-VR2) with approximately a 5% increment in recognition accuracy with a low level of noise of 42 dB and approximately 3% with a higher level of noise of 72 dB. The AND program can be used effectively in a low noise environment with FP error rates $\approx 0\%$, which is safer for rehabilitation applications.

The VR accuracy of the OR program reaches more than 98% in non-noisy environments using the English language. Compared to the findings of other studies [24], [40]–[42], [206]–[210], this outcome has the following advantages:

- The system can avoid and profoundly reduce the FP errors, which are the most harmful errors for VR rehabilitation applications.
- The maximum percentage accuracy of VR is achieved in noisy and non-noisy environments up to 72 dB of noise. Most previous works conducted tests without measuring the noise level, which makes the results useless for comparison with those at specific noise levels.
- The system was tested with three different languages and it has the flexibility to use any language in SD voice control and 8 global languages in SI voice control. The majority of previous studies performed tests with only one language.
- The system was tested with more than 10 people with different accents, which exceeds the total numbers of users in most previous works.
- The system works as stand-alone without needing to use a PC.

The system was tested successfully with the Jaguar Lite robot and the Meyra smart 9.906 wheelchair and it is designed to work with any robot or wheelchair after modifying the motor driver parameters.

6 - Head Tilt Controller

6.1 Overview of the Head Tilt Controller

The orientation of the head has been selected to be the second controller in the current work. The orientation sensors have to be used to design an intelligent head tilt controller which interprets the head orientation around the x, y, and z-axes to control commands that control the speed and direction of the wheelchair or rehabilitation robot. The combination of the voice controller and head tilt controller produces a reliable system with two types of controllers that can give the user the chance to choose or change control mode easily and comfortably.

Figure 6-1 shows a block diagram of the head tilt controller. It consists of two BNO055 modules. The output of this module is given in different forms of orientation. The current system uses the Euler angles of pitch, roll, and yaw form to implement the head tilt controller algorithms. The first orientation sensor is fixed on a headset that is worn by the user on his head. It is used to measure the head orientation in the form of pitch (P) and roll (R) Euler angles and fed it to the microcontroller. The pitch Euler angle represents the head tilt for forward and backward which are interpreted as forward and backward movement commands. The roll Euler angle represents head movements to the left and right directions which are interpreted as left and right control commands. The tilt angles from 0° to -10° for pitch and from -5° to 5° for roll are used for a stop command. The second orientation detection module is used to measure the reference orientation of the wheelchair or robot system. It is fixed on the chassis of the wheelchair or robot. The reference orientation is used to measure road slope angles to allow the system to recognize non-straight roads like hills or slopes. It plays an important role in the implementation of an auto-calibrated algorithm and the speed compensation algorithm that is used to enhance the performance of the head tilt controller.

The two BNO055 orientation modules communicate with the main microcontroller using the inter-integrated circuit I²C bus. Each orientation module has a different I²C address to allow the microcontroller to recognize each of them. The first module has the I²C address (0x28) and the second (0x29). The orientation modules send orientation data at a rate of 100 samples per second. This sampling rate covers the requirements of the proposed system as introduced in chapter 3.

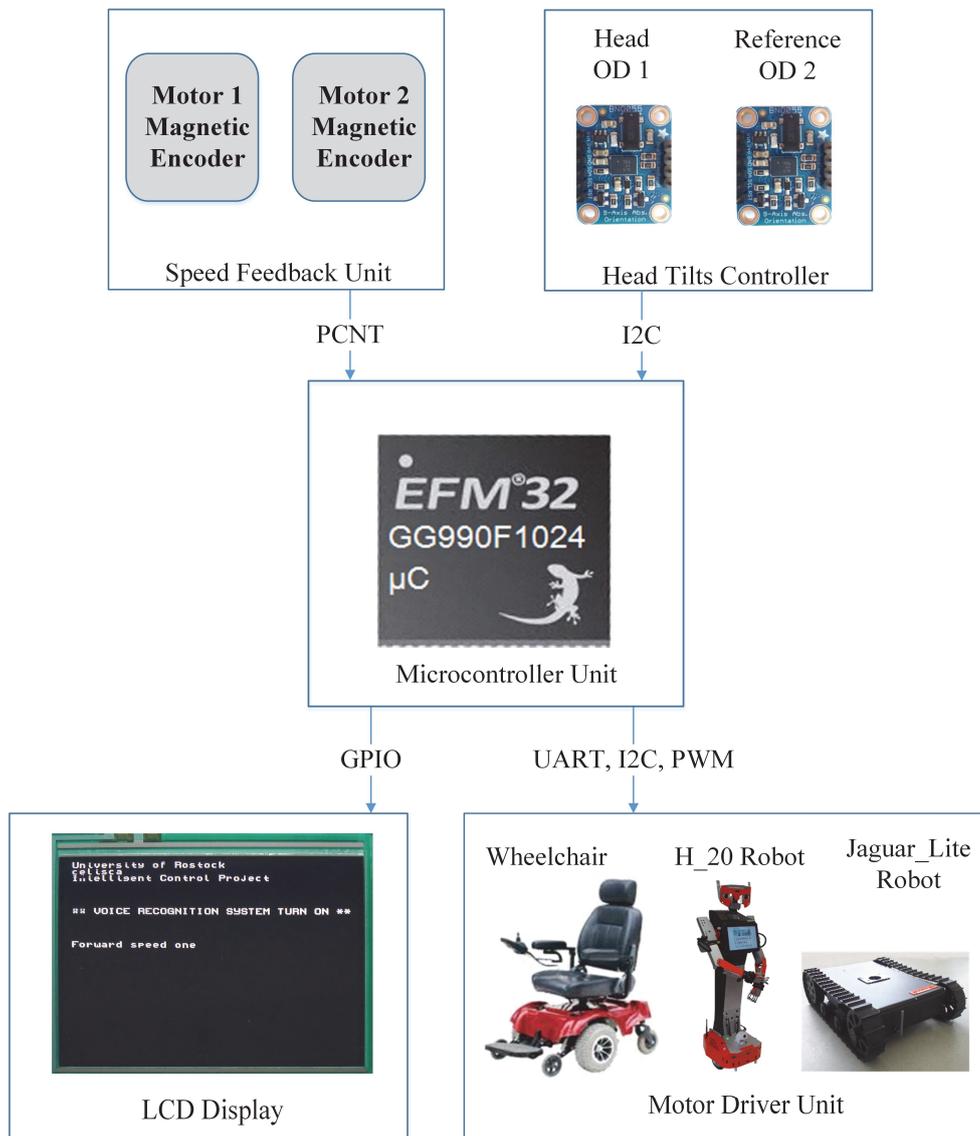


Figure 6-1: Block diagram of the head tilt controller

Each control command of the head tilt controller has a specific orientation angle that represents the command's active area. The standard angle for the forward command starts from a pitch tilt angle of 0° to 25° which is proportional to a system speed from 0 to maximum speed depending on the application used (Jaguar robot or wheelchair). The forward command includes not only the forward direction but also forward to the right and forward to the left, which are achieved by tilting the user's head to forwards with a left or right direction. The stop command has a pitch tilt angle between 0 and -10° and roll tilt angle of -5° to 5° . The backward command has pitch tilt angles between -10° and -45° to start from 0 to maximum speed which is programmed at a limited speed of 2 km/h. The backward command includes not only the backward direction but also backward to the left and backward to the right which is programmed to be similar to when

driving a car in reverse. The left control command has a roll tilt angle between -5° and -45° and the right control command has a roll tilt angle between 5° to 45° to start from 0 to a maximum speed of 2 km/h. The remaining tilt angle ranges have been programmed with safety functions to protect the user in case of any incorrect range of orientation detected. This problem can happen when the headset is dislodged from the user's head or when the user loses consciousness. This function will stop the system immediately to protect the user, and the system will give an indication on the LCD display about the expected problem and ask the user to return his head to the stop region to restart the system. This function keeps the system halted and does not let the system return to work until the headset returns to the stop area of the orientation sensor. Figure 6-2 shows the command tilt angles for the forward, stop and the backward control commands. Table 6-1 explains the system control commands and their angles.

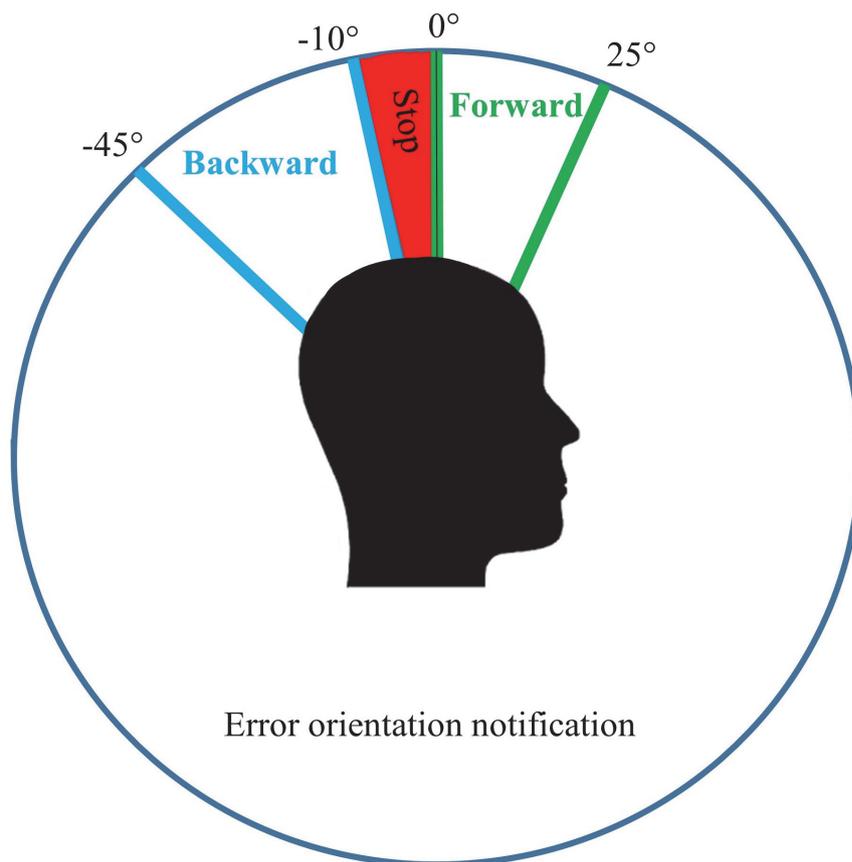


Figure 6-2: Tilt angles for the head tilt controller

Table 6-1: Head tilt controller commands description

Command	Pitch angle	Roll angle	Definition
Forward	$0^\circ < P < 25^\circ$	$5^\circ \geq R \geq -5^\circ$	Both wheelchair motors rotate forward
Left	$0^\circ \geq P \geq -10^\circ$	$5^\circ < R < 45^\circ$	Right motor rotates forward and the left rotate backward
Right	$0^\circ \geq P \geq -10^\circ$	$-5^\circ > R > -45^\circ$	Left motor rotates forward and the right rotate backward
Backward	$-10^\circ > P > -45^\circ$	$5^\circ \geq R \geq -5^\circ$	Both motors rotate backward
Forward-Left	$0^\circ < P < 25^\circ$	$5^\circ < R < 45^\circ$	Right motor rotates faster than left motor in forward direction
Forward-right	$0^\circ < P < 25^\circ$	$-5^\circ > R > -45^\circ$	Left motors rotate faster than right motor in forward direction
Backward-Left	$-10^\circ > P > -45^\circ$	$5^\circ < R < 45^\circ$	Right motor rotates faster than left motor in Backward direction
Backward-right	$-10^\circ > P > -45^\circ$	$-5^\circ > R > -45^\circ$	Left motors rotate faster than right motor in Backward direction
Stop	$0^\circ \geq P \geq -10^\circ$	$5^\circ \geq R \geq -5^\circ$	Both motors stop

The head tilt controller is programmed to control the speed of the wheelchair motors depending on the P and Y angles values of the user's head with reference to the origin point ($P=0$, $R=0$). If the user maintains his head tilt angle ($P > 0^\circ$) and ($-5^\circ < R < 5^\circ$), then both wheelchair motors will rotate in the forward direction at the same speed. If the user tilts his head in the forward direction with a right-side tilt (see Figure 6-3) this means that the user wants the wheelchair to move forward in a direction to the right. In this case, the left motor (M2) speed will be increased ($M2 = P + R$) and the right motor (M1) speed will be decreased ($M1 = P - R$) depending on the value of the head tilt in the y-axis. The same concept applies when the user needs to go forward with a left curve, where the speed of the right motor will increase and the speed of the left motor will decrease. The backward speed control has the same functionality. The user can control the backward command motion with a straight motion or curved to the left or right. When the user tilts his head less than -10° ($P < -10^\circ$) and the roll angle is kept in ($-5^\circ < R < 5^\circ$), the wheelchair motors will rotate at the same speed in the backward direction. When the user tilts his head in the backward direction and to the right, the system will move backward with a right curved path by increasing the left motor M2 speed and reducing the right motor M1 speed. If the user tilts his head backward in the left direction, the system will move the wheelchair backward with a left curve. To

implement the backward with a left curve, the right motor M1 speed will increase ($M1=P+R$) and the left motor M2 speed will decrease ($P-R$) depending on the values of both tilt angles.

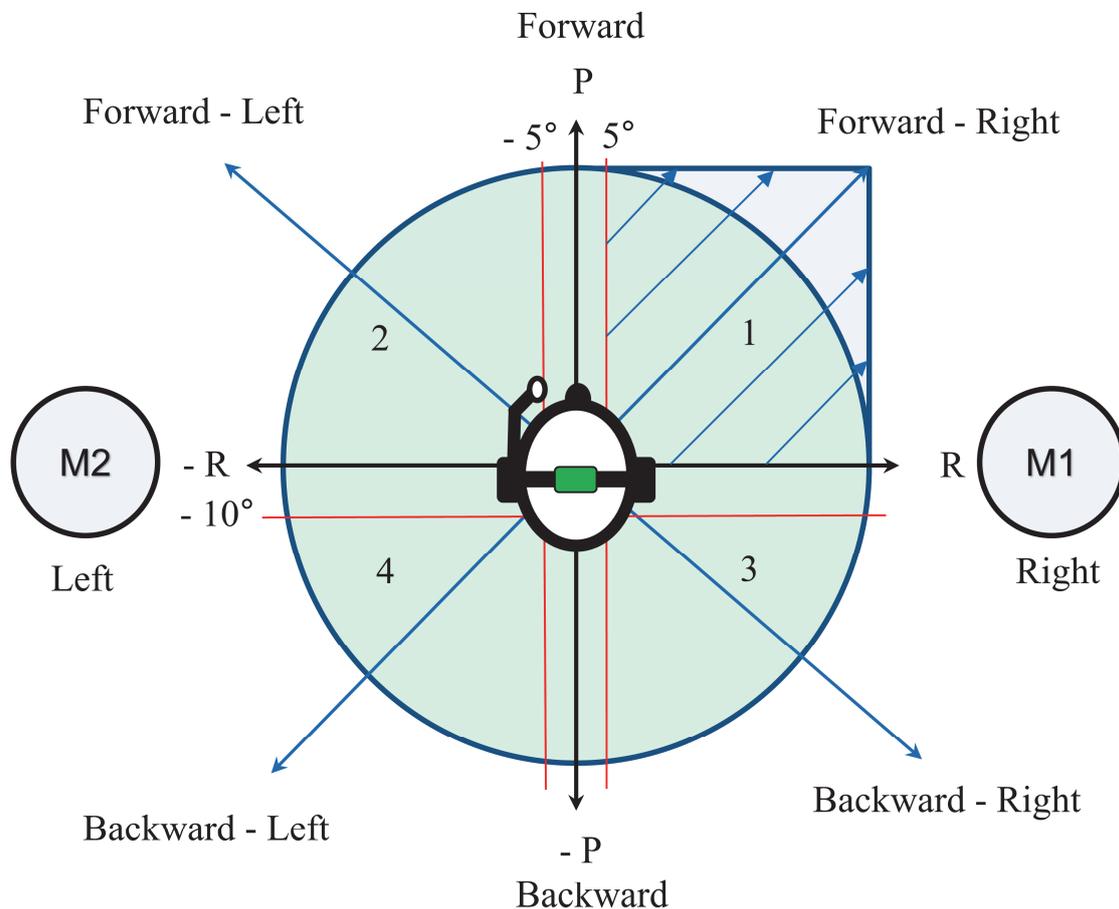


Figure 6-3: System motors control by head tilt

The user can rotate the wheelchair to the left or right at the same point by tilting the head left or right and keeping the value for the pitch angle in the range of $(-10^\circ \leq P \leq 0^\circ)$. The “right” control command will make the left motor (M2) rotate forward and the right motor (M1) rotate backward so that the wheelchair will rotate to the right direction at the same point. The “left” control command makes the right motor (M1) rotate forward and the left motor (M2) rotate backward to let the wheelchair rotate in the left direction at the same point. Figure 6-4 explains the speed control algorithm of the head tilt controller.

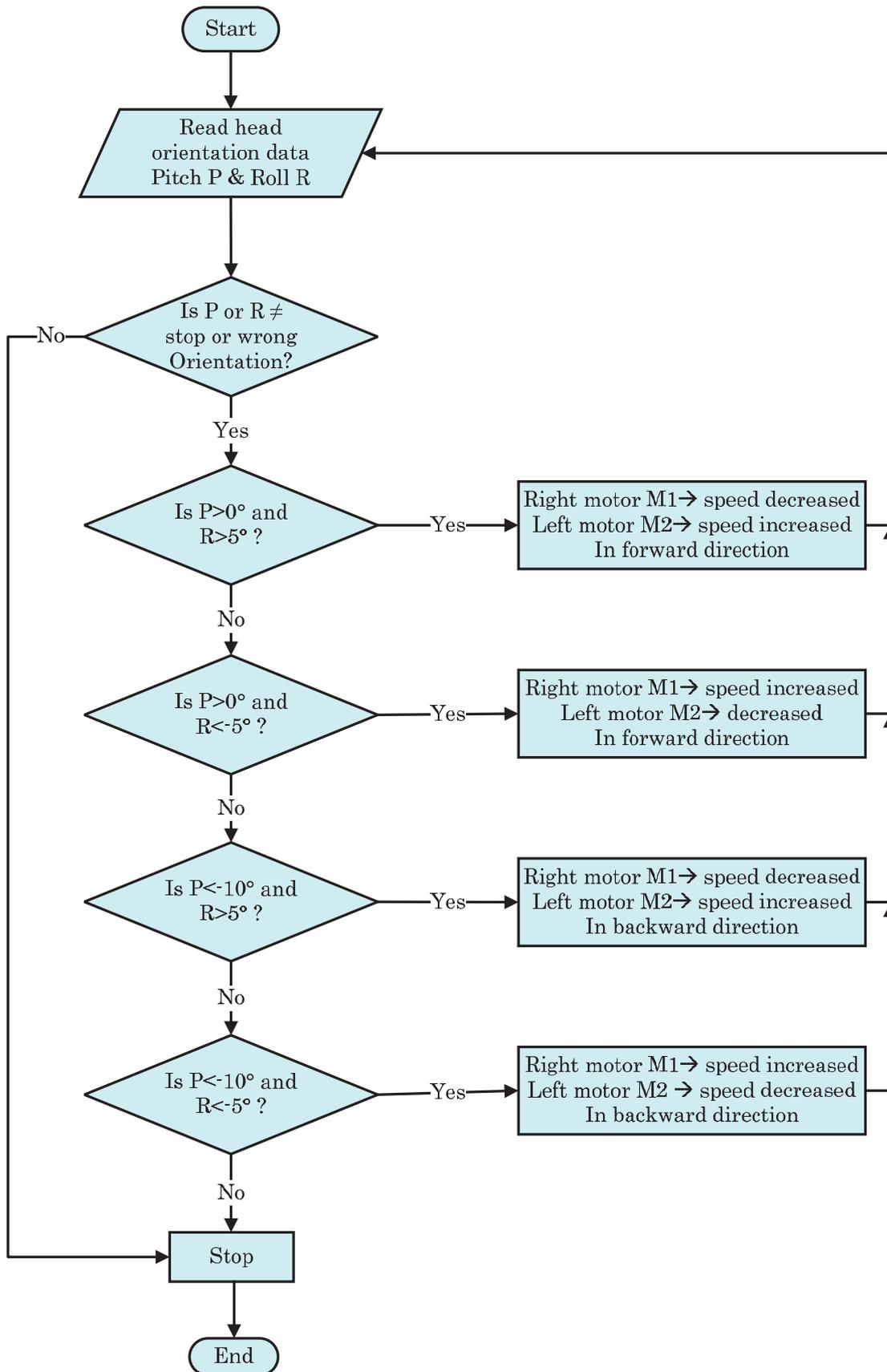


Figure 6-4: Speed control algorithm of head tilt controller

The first version of head tilt algorithm was previously tested with Jaguar Lite robot [211], [212]. It covers specific points of head movements to control the motion of the robot and it does not cover all the motion points of the user's head in the motion domain. For example, if the user wants to move the robot forward with a left curve he needs to keep his head in the middle line between +P and +R Euler angles and any head tilt out of this line of points will not affect the system control. The new control algorithm in Figure 6-4 is an improvement of the first head tilt algorithm. It covers all the possible head motion and responds optimally to the required direction similar to the response of joystick controlling. Each head tilt point which is represented by P and R Euler angle has its own speed and direction parameters and the system response to any head tilt motion in any direction. The new algorithm enhances the performance of the head tilt controller by making it much easier, smoother and has a similar direction response of the original joystick controller.

6.2 Autocalibrated Algorithm

When the robot passes along a straight road, the auto-calibrated algorithm will not be activated since the values of the two Euler angles of the two orientation detection modules will be near the straight position limit (-10° to $+10^\circ$ road slope). In the case of an ascending or a descending slope in the road, the value of the Euler angles of the second orientation module at the robot chassis will change relative to the value of the slope of the road. The first orientation module on the headset also takes the same road slope and the difference between the Euler angles of the two orientation detection modules will be kept in the range of (-10 to +10 degrees) and this will avoid drift on a non-straight road.

The use of only one orientation module on the headset to control the wheelchair motion without a reference orientation will make the system control difficult when it passes along non-straight roads. When the road slopes up or down, this will lead to harmful errors in orientation control. The user's head orientation will change relative to the change in the road slope. When the road slope reaches a threshold level it can give incorrect control commands. The auto-calibrated algorithm cancels the effect of road slope by using the orientation of the wheelchair as the reference orientation. In this case, the system will not reach the threshold level when the road is not straight. It will reach the threshold level only when the user's head moves in the specific movement to choose the direction and speed of the system. The user needs to keep his head perpendicular to his body as

in a normal situation to apply the stop command. Figure 6-5 shows the situation in which the auto-calibrated head controller is required.

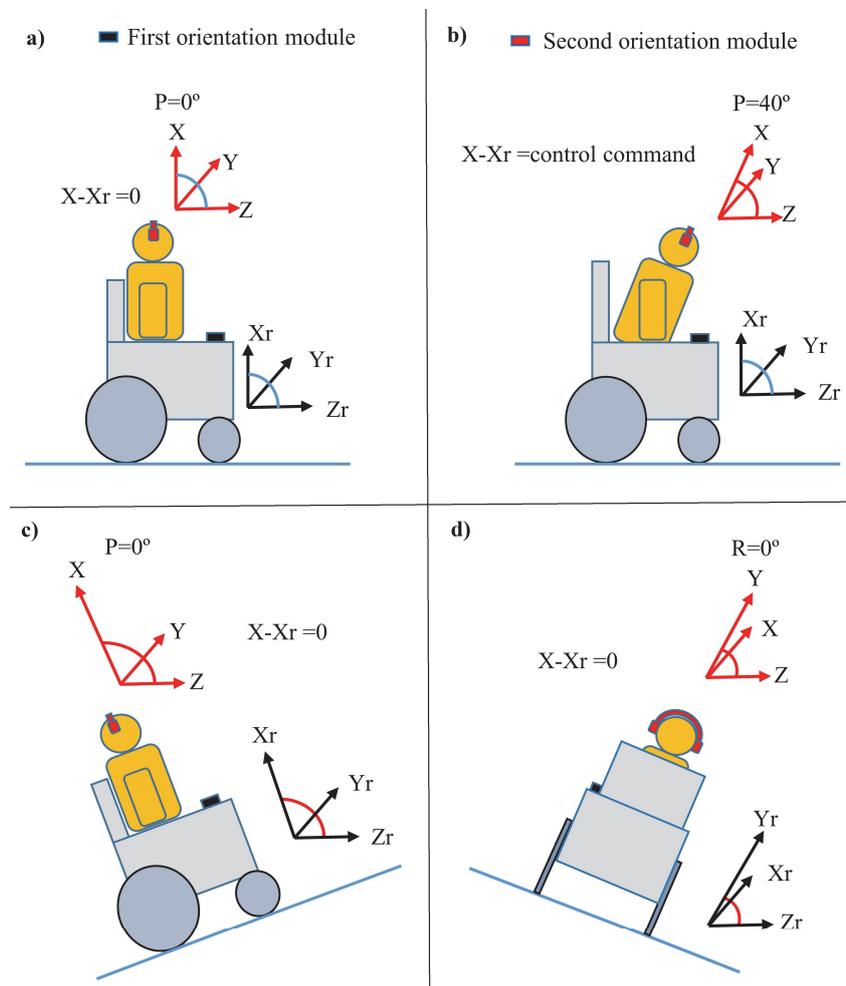


Figure 6-5: Head tilt controller at different road slope angles

Figure 6-6 shows a flowchart of the auto-calibrated algorithm for the head tilt controller. The algorithm starts the acquisition of orientation data from the two orientation modules. The orientation of each module is recorded at six internal 8-bit registers in the orientation module. The value of both high and low registers represents the value of one Euler angle as a 16-bit representation (8-bit high + 8-bit low). This 16-bit value can be changed into degrees or radians by controlling the internal register (UNIT_SEL) of the BNO055 module to represent the value of the Euler angle in a specific measuring unit. After data acquisition, the algorithm compares the value of each Euler angle from the first orientation module with the equivalent from the second orientation module. The algorithm checks whether there are any errors in head position and, if so, sends an alarm to the LCD display

to instruct the user to correct his head position. If there is no error, it checks the value of the comparison of Euler angles. If it reaches the threshold of any control command, it will send the required command to the motor driver unit in the Jaguar Lite robot or Meyra wheelchair.

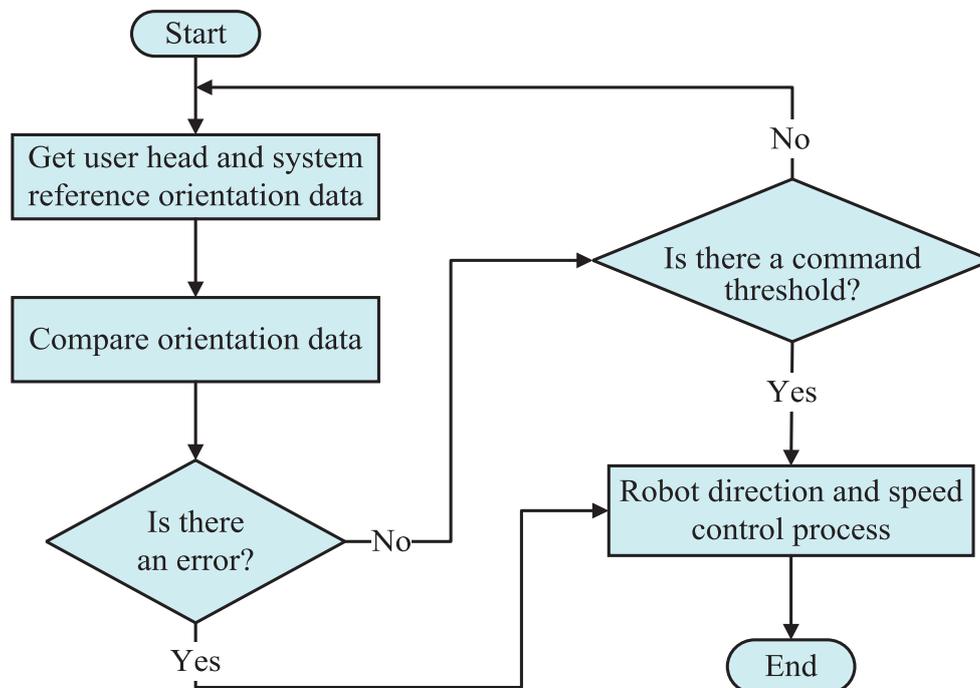


Figure 6-6: Flowchart for auto-calibrated algorithm

When the wheelchair passes along straight roads, the difference between the reference orientation and head orientation is approximately zero, and when the user keeps his head in the center region this is used to stop the wheelchair. The head tilt in any direction will be subtracted from the reference orientation until it reaches the threshold of the wheelchair motion command. The command threshold will set the direction and required speed depending on the value of the head tilt angle. When the wheelchair is ascending or descending a ramp, the user will obviously appreciate the activity and advantage of the auto-calibrated algorithm. Thus, if the wheelchair is ascending a ramp without using the auto-calibrated algorithm, an incorrect backward command would be activated since the system takes the road slope as a head tilt. The same concept is applied for the descent of a ramp or if the roadside altitude is not symmetrical.

6.3 Speed Compensation Algorithm

The speed compensation algorithm is used with the head tilt controller to compensate for the speed change in one or both system motors due to gravity forces when it is ascending or descending a non-straight road. This algorithm is applied to control the Jaguar-Lite robot since it has two optical encoders connected to the robot motors. The algorithm can be applied to any other system, such as a wheelchair, after adding encoders to the system motors used. The speed compensation algorithm is activated in various situations. The first is when the wheelchair is passing along a ramp and the speed of the motor starts to decline due to the increment of gravity force. In this case, the algorithm starts first to check if there is a side slope of the road (see Figure 6-5-d). If the value of side slope angle (roll Euler angle) reaches the programmed threshold it will guide the speed compensation algorithm to select the appropriate left or right motor. Then the required speed compensation for the target motor will be calculated.

The second situation is when the wheelchair passes along a ramp with a slope in the pitch angle (see Figure 6-5-c). The algorithm will check whether the pitch angle reaches the threshold value and then it starts to compare the speed given by the user and the real speed measured by the encoders. The algorithm will calculate the speed factor and send it to the motor driver to compensate for the wheelchair speed. The same procedure is implemented in reverse when the wheelchair is descending a ramp due to the reduction in gravity force. Figure 6-7 gives a flowchart illustrating the speed compensation algorithm.

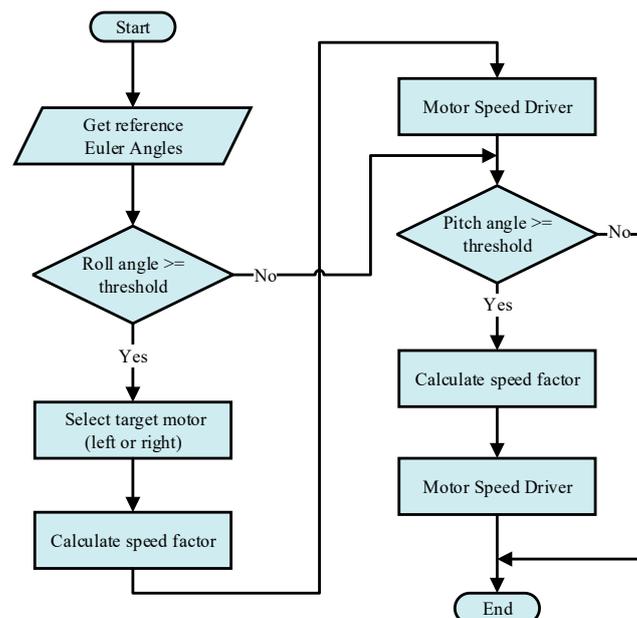


Figure 6-7: Flowchart for speed compensation algorithm

The motor speed feedback is measured using the magnetic encoders. Each encoder is fixed onto one of the system's DC motor shafts. When the motor rotates, the encoder senses the shaft rotation and starts to generate square pulses depending on the number of shaft cycles. Each wheel rotation generates 320 encoder pulses. The system speed feedback for each motor is calculated using the following equation:

$$S = (D_2 - D_1) / (t_2 - t_1) \quad (6-1)$$

Where:

S = Motor speed

D₁ = Distance at time 1

D₂ = Distance at time 2

t₁ = time at start of encoder measurements

t₂ = time at end of encoder measurements

$$D = R * \pi * W \quad (6-2)$$

Where:

D = distance

R = number of wheel rotation

W = wheel Diameter

$$R = E_p / E_r \quad (6-3)$$

Where:

E_p = encoder pulses

E_r = encoder resolution = 320 pulse per rotation

Each encoder output is sent to a pulse counter in the microcontroller. The PCNT0 and PCNT1 pulse counters have been used to implement this task. Each pulse counter has been programmed to make an increment of the counter for each pulse generated by the encoders. Each rotation generates 320 pulses from the encoder and appears as 320 counts in the pulse counter. The motor speed is calculated using equation 6-1.

The speed compensation algorithm has been applied when a side slope in the reference wheelchair orientation has been detected. The algorithm starts by

calculating the left and right motor speed depending on the magnetic encoder feedback. The algorithm checks whether there is any difference between the two motor speeds. It keeps both of them at the same required speed given by the user and cancels the speed added by gravity force in the case of ascending or descending ramps. The algorithm compensates for the motor speed depending on the change in the speed factor of each motor individually and makes a comparison and compensation according to the difference between the speeds of both motors for a fixed time period. It checks whether there are differences in the motor speed for the forward control command, and then the algorithm will reduce the speed of the highest speed motor until it is equal to that of the second motor. Figure 6-8 shows the speed compensation procedure for each individual wheelchair motor.

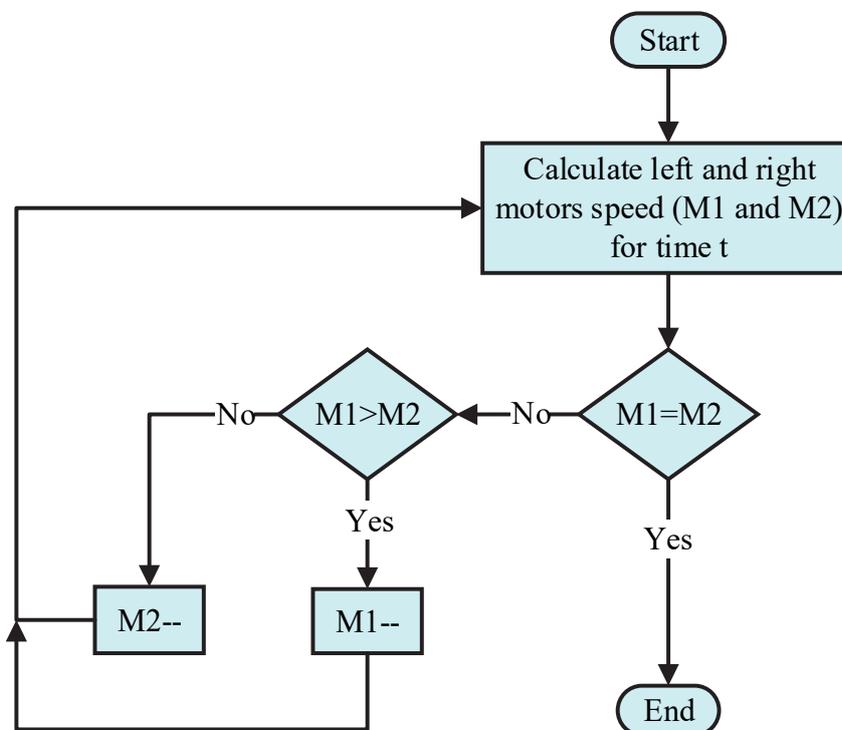


Figure 6-8: Speed compensation procedure for each individual motor

The speed control of the head tilt controller is directly proportional to the value of the user's head tilt angle. Each control command starts with the minimum speed when it reaches the command threshold value of tilt angle and the maximum speed when it reaches the maximum command tilt angle. It is clear that the value of motor speed will increase if the head tilts angle increases and vice versa. The ratio between the tilt angles and the maximum speed of the wheelchair can be programmed and adjusted depending on the user's preference. Basically, three speeds have been selected for the system, which are the "basic speed", "speed

one” and “speed two” to be used for indoor, outdoor, and long distances respectively. The number of speed levels can be increased or decreased depending on user requests. The basic speed starts from 0 Km/h until it reaches one-third of the original system's maximum speed, which is 6 Km/h for the Meyra wheelchair and 8.5 Km/h for Jaguar-Lite robot. “Speed one” starts from 0 Km/h until it reaches two-thirds of the original system's maximum speed and “speed two” starts from 0 Km/h until it reaches the original system's maximum speed. The three tilt to speed ratios are 2, 4 and 6 Km/h for the Meyra wheelchair and 2.5, 5 and 8.5 Km/h for the Jaguar-Lite robot. The user can change the system tilt to speed ratio using the voice commands “forward” for basic speed, “speed one” for a speed ratio increment or decrement from “speed two” and “speed two” for maximum system speed. Figure 6-9 explains the relationship between head tilt angle and system speed.

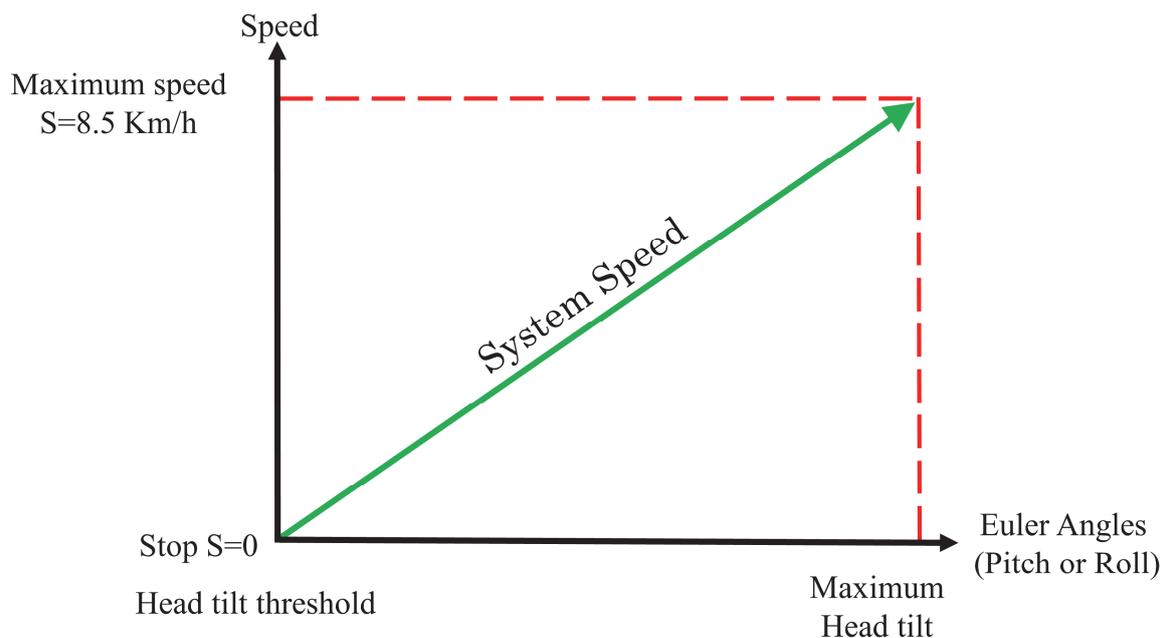


Figure 6-9: Tilt to speed ratio of the head tilt controller

6.4 Experimental Results

The head tilt controller has been tested with the Jaguar-Lite robot as well as with the Meyra Smart 9.906 wheelchair. There are several important differences between the two test platforms. The user in the Jaguar robot test will control the robot from the surrounding area while the user will be positioned on the Meyra wheelchair system and this will add extra load to the system motors and affect system performance when passing along a non-straight road. The second difference is the type of wheels used. The Jaguar-lite robot uses a tank drive track compared to normal wheels for the Meyra wheelchair. In addition, the Jaguar motors have magnetic encoders that allow the programming of the speed compensation algorithm using speed feedback from the system encoders, whereas the normal wheelchair motors on the Meyra lack encoders. Thus, the results are discussed below separately.

6.4.1 Test Results for Jaguar-Lite Robot

The tests measured the time required for the speed compensation algorithm to compensate for the speed of the Jaguar-Lite robot which has asynchronous motor speed. The tests show the performance of the speed compensation algorithm in cases of ascending and descending an artificial ramp. Figure 6-10 shows the results for the forward command response time in the case of ascent and descent with 10 samples for each angle. The Jaguar robot runs at a test speed of 2.5 km/h where the maximum speed is 8.5 km/h. The test starts with a 15° slope angle increasing to a 45° slope angle with 5° step increments for both ascent and descent. The compensation time is the processing time of the speed compensation algorithm to make both the left and right motors rotate at the same speed for the given command by compensating for the difference between the speed desired by the control command and the actual speed measured by the encoders. The practical results show acceptable compensation time for path correction depending on the compensation of the motor speed using the feedback from the encoders. This is an important feature in rehabilitation and medical applications.

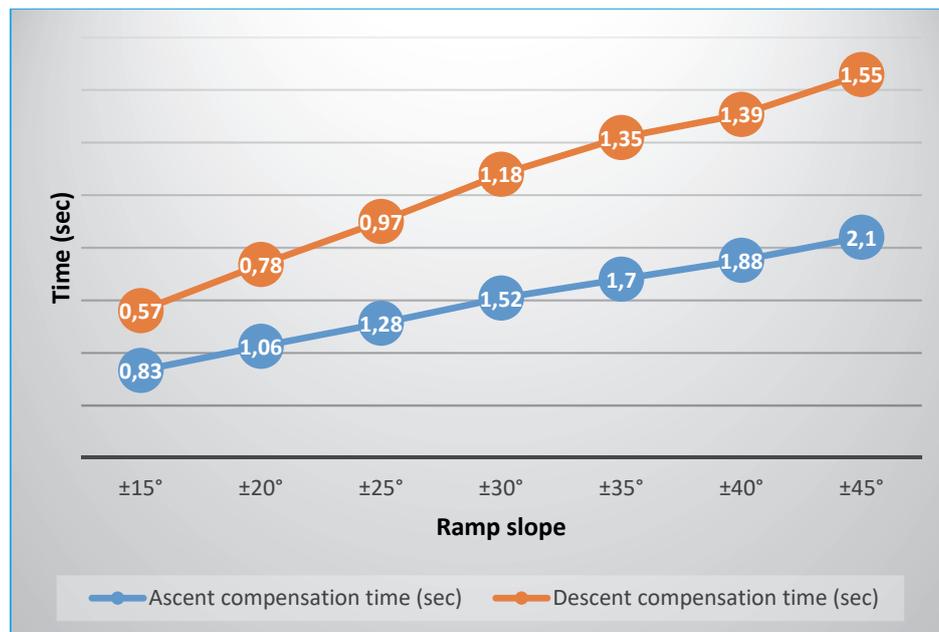


Figure 6-10: Compensation time of the system motors

The second test checked the accuracy of control commands from the auto-calibrated head tilt controller in the case of passing along a ramp. This test is very important to check the calibration tolerance of the BNO055 orientation modules used in the system. This test investigates the percentage success rate of each individual control command at different slope angles of an artificial ramp at a fixed speed of 2.5 km/h. The user gives the test commands after stopping on the ramp at a specific slope angle. An artificial ramp with a manually changeable slope has been used to implement the test. Five users performed the test by giving each control command 20 times to the system at each ramp angle. Two additional control commands have been tested, which are speed one and speed two. These two commands were used to change the sensitivity of the system proportional to the tilt angle. The results shown in Table 6-2 indicate the success rate of each control command at different road slope angles. Each row includes the control command and its percentage accuracy when given by 5 users and repeated 20 times for each slope angle. The system success rate of the control commands reached more than 97.5%. The error levels in the test are less than 2.5% and all of them are false negative (FN) errors. FN errors mean that the system does not respond to the given command or there is a time delay in implementing the control command. This could occur because of the auto-calibration of the MEMS triaxial gyroscope, triaxial accelerometer and triaxial magnetometer sensors of the BNO055 orientation detection module, and should be dealt with in future work.

Table 6-2: Command accuracy tests for system at 2.5 km/h speed

Commands	Command Success rate for artificial ramp slope angle					
	15°	30°	45°	-15°	-30°	-45°
Forward	100	100	98	99	99	97
Left	98	98	97	98	99	98
Right	98	98	97	97	97	98
Backward	97	98	96	99	96	96
Stop	98	99	97	99	96	97
Speed one	94	98	98	99	96	97
Speed two	94	98	98	97	97	97
Accuracy %	97%	98,4%	97,2%	98,2%	97,1%	97,1%

6.4.2 Test Result for Meyra Smart 9.906 Wheelchair

The head tilt controller has been tested in indoor environments with a mall path and numerous obstacles as well as in an outdoor environment via passing hills and non-straight roads. Several different tasks have been implemented to test the system for indoor motion control. In the primary test, individual control commands for the head tilt controller were checked without previous training. Five users tested seven control commands activated by head tilt around the x- and y-axes. The tested control commands are forward, backward, left, right, stop, forward-left, forward-right. Each user tested each command 10 times. The system speed was fixed to 2 km/h which is the basic ratio of the Meyra smart 9.906 wheelchair. The same test was repeated for the same previous basic commands after training the same users with the head tilt controller for approximately 30 minutes. The test results before and after 30 minutes training for the control command tests are summarized in Table 6-3, where the abbreviation NTU refers to the non-trained user and the abbreviation TU refers to a user trained for half an hour.

Table 6-3: Test results for the head tilt controller

Commands	NTU	TU
Forward	96 %	100 %
Left	82 %	98 %
Right	78 %	100 %
Stop	92 %	100 %
Backward	94 %	100 %
Forward-left	82 %	100 %
Forward-right	76 %	98 %

The result in the table refers to the successful implementation of the control task. For example, to implement the forward control command the user needs to drive the wheelchair in a straight line between two points by tilting his head in a forward direction and keep the roll angle between $-5 < R < 5$. The forward command result shows that there are two failed attempts and 48 successful attempts of total 50 attempts which is = 96% successful rate for the non-trained before user. The result after 30 min training shows that the user gets 100 % of successful implementation of the control commands. The same procedure has been implemented for all the head tilt controller commands in Table 6-3. Figure 6-11 explains the difference in the command success rate before and after 30 minutes of user training.

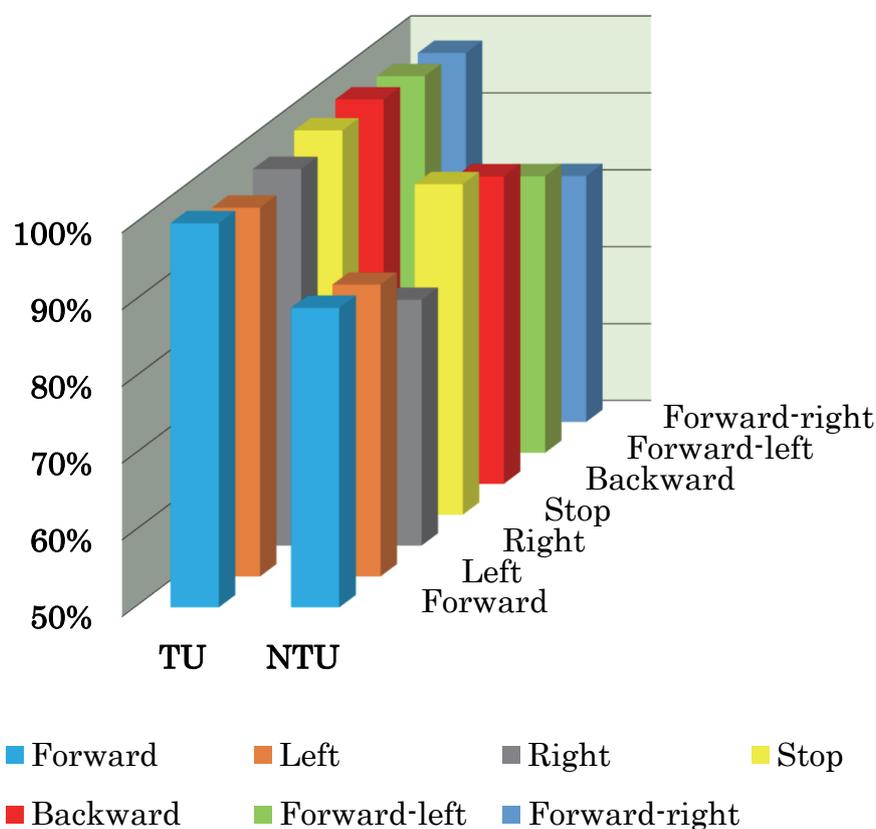
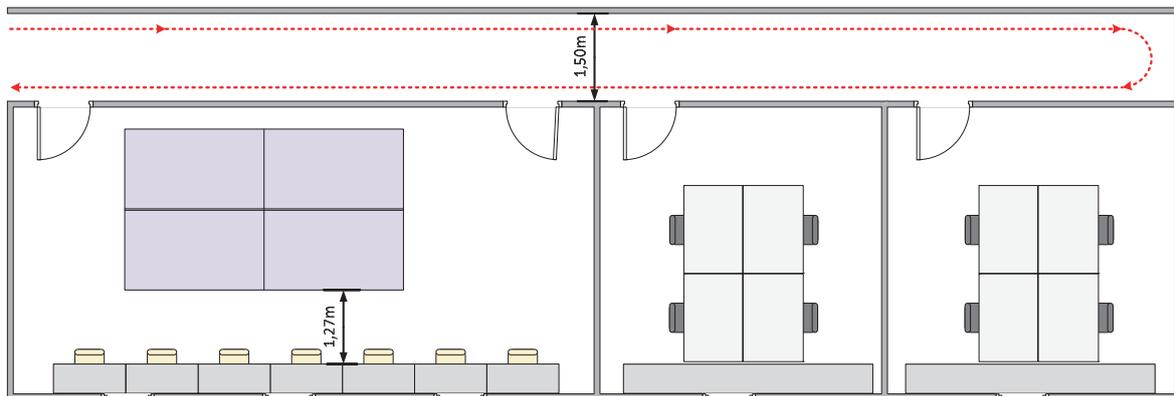


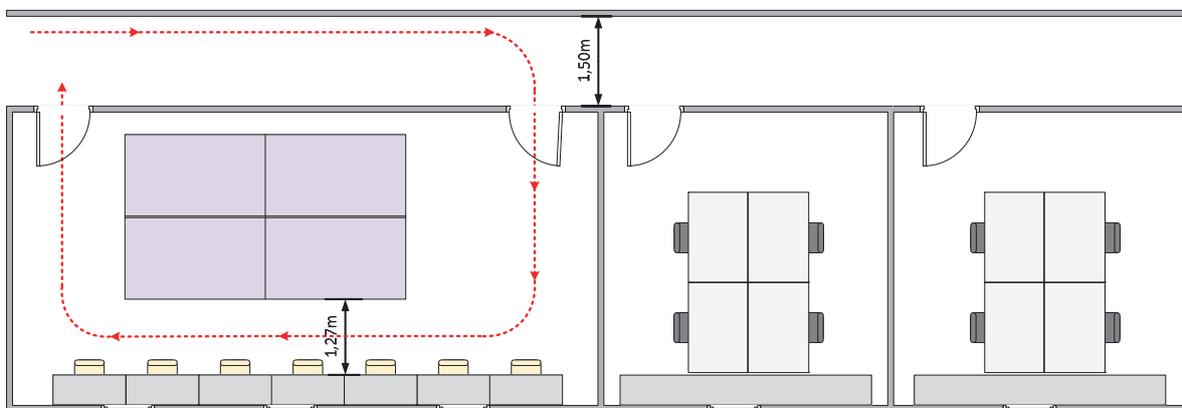
Figure 6-11: Head tilt controller commands success rate

The results in Table 6-3 and Figure 6-11 show the control command success rates. A success is the correct implementation of the required command at the required speed and direction from the start to the end point. The test results revealed that the success rate increased when the users were first trained in using the system.

The second test for indoor motion control is to drive the wheelchair in a narrow indoor area. Two tasks have been tested, which are passing along a narrow corridor and 2) passing along a pathway in a furnished hall. Both tasks have been taken in the celisca building by the same five users who were trained with the system before. The test results show the successful performance of the head tilt controller. All 5 users completed tasks 1 and 2 without collision and followed the required paths correctly. Figure 6-12 shows the test paths.



1- Passing a narrow corridor



2- Passing a pathway in a furnished hall

Figure 6-12: Tasks for system indoor test

The head tilt controller was also tested for outdoor environments, and especially for the use of the auto-calibrated algorithm designed to cancel the effect of road slope in cases of ascending or descending a hill. The tests took place on a small ramp with a height of 80 cm and a slope angle of 7 degrees. Five users repeated the test twice, once without the auto-calibrated algorithm and the second time with the auto-calibrated algorithm. The tilt to speed ratio was fixed to a 25°

tilt angle to 2 km/h. The test results show that the auto-calibrated algorithm simplifies wheelchair movement while traversing a hill and makes it similar to driving the wheelchair on a straight road with normal head tilt angles. The test of the system for ascending a hill without the auto-calibrated algorithm revealed that the control of the wheelchair is more difficult. While ascending the hill the user has to tilt the head an extra 7 degrees down compared to the motion with the auto-calibrated algorithm to compensate for the hill slope (7°) and vice versa. Figure 6-13 shows the results of tests of the system with and without an auto-calibrated algorithm.



a) Without auto-calibrated algorithm, 7° more in the head tilt



b) With auto calibrated algorithm

Figure 6-13: System tests for passing a long ramp

6.5 Functions Embedded in the Head Tilt Controller

There are several functions included in the head tilt controller to overcome some problems which occurred during the system tests. The functions will be explained in the following sections.

6.5.1 Command Confirmation Function

During the testing of the system, the head tilt controller is faced with a harmful situation as a result of user head movement around the borders of the stop command area. This problem occurs when the user tries to keep his head in the stop region ($-10^{\circ} \leq P \leq 0^{\circ}$, $-5^{\circ} \leq R \leq 5^{\circ}$) and involuntarily moves the head out of the stop borders to motion commands borders for less than 50 ms without intending

the system to move. This problem results in continuous, discrete motion-stop commands that can cause concern to the user and also unwanted motion may occur. A new function has been added to the head tilt controller to avoid this unwanted situation by increasing the number of measurements for the desired control command from the head orientation sensor. Instead of using only one reading in the range of $0^\circ < \text{tilt angle} < 25^\circ$ for the forward command, the system uses 10 continuous measurements in the region of $0^\circ < \text{tilt angle} < 25^\circ$ for the forward command tilt area in order to implement the forward command. The same procedure is used for all other commands. This problem occurred frequently before using this function. The problem has been totally solved using the command confirmation function. Figure 6-14 explains the tilt angles for the forward command before and after using the command confirmation function.

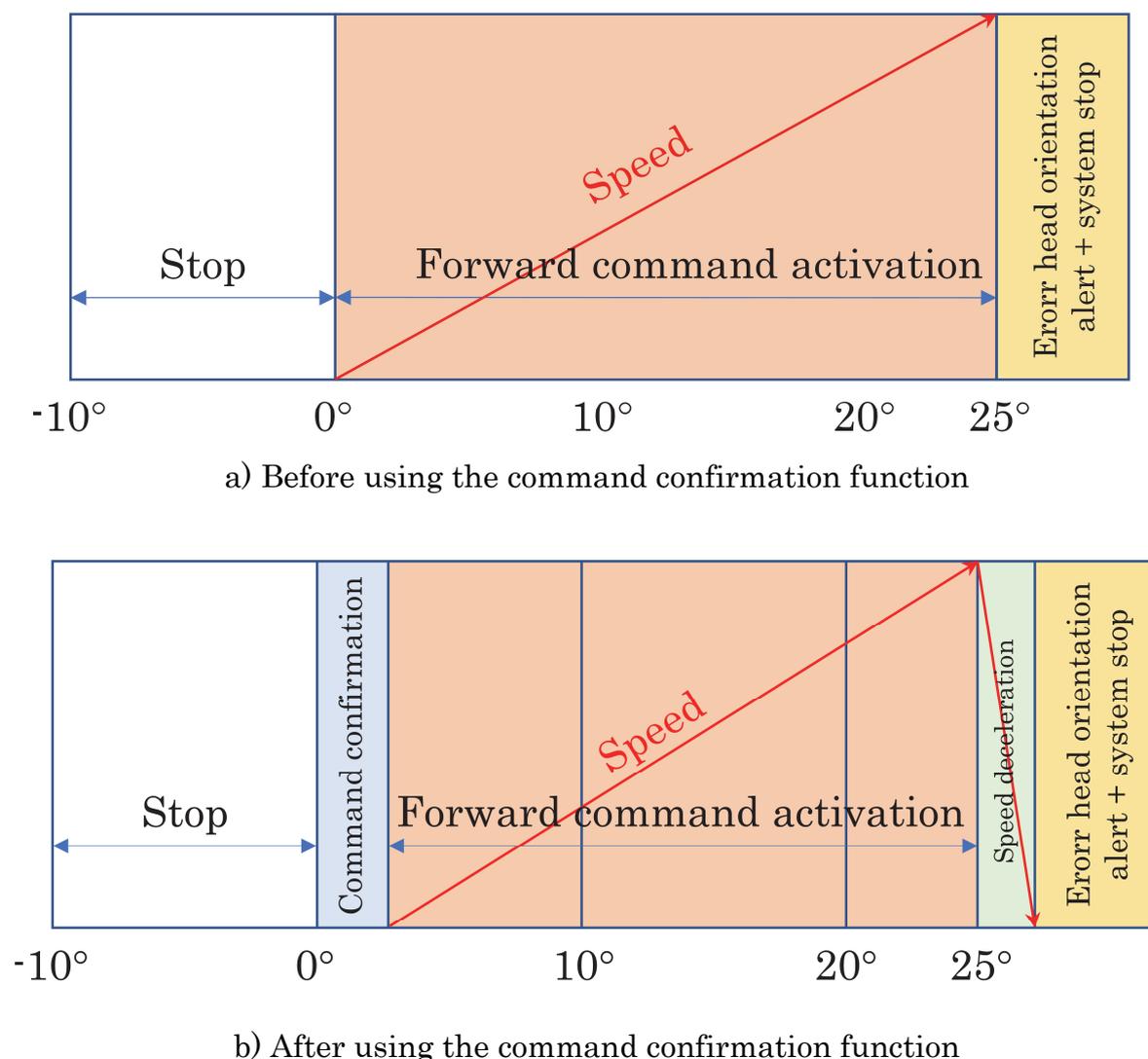


Figure 6-14: Command confirmation function

6.5.2 Wrong Orientation Handling Function

The wrong orientation handling function has been programmed to stop the system and protect the user if the head orientation goes out of the programmed ranges of the control commands tilt angles. The function stops the wheelchair and suspends the control commands of the head tilt controller if the reading of the head tilt angle is beyond the normal control commands borders. The function stops the wheelchair until the headset returns to the center stop region ($-10^{\circ} \leq P \leq 0^{\circ}$, $-5^{\circ} \leq R \leq 5^{\circ}$). This function protects the user in case of the headset falls down or the user loses consciousness. This function has been tested by 10 users. The user has to apply a wrong head orientation by tilting his head out of the motion control commands ranges ($-45^{\circ} > P > 25^{\circ}$, $-45^{\circ} > R > 45^{\circ}$) and check the system behavior. The function test result shows 100% successful emergency stop and the LCD display showed alarm indications about the wrong head orientation (see Figure 6-15).

The function has been programmed to stop the wheelchair gradually to make a smooth stopping because the function will be activated after the head tilt reaches the maximum speed tilt angle. Figure 6-14-b explains the gradually stop in the green color area after 25° forward tilt angle.

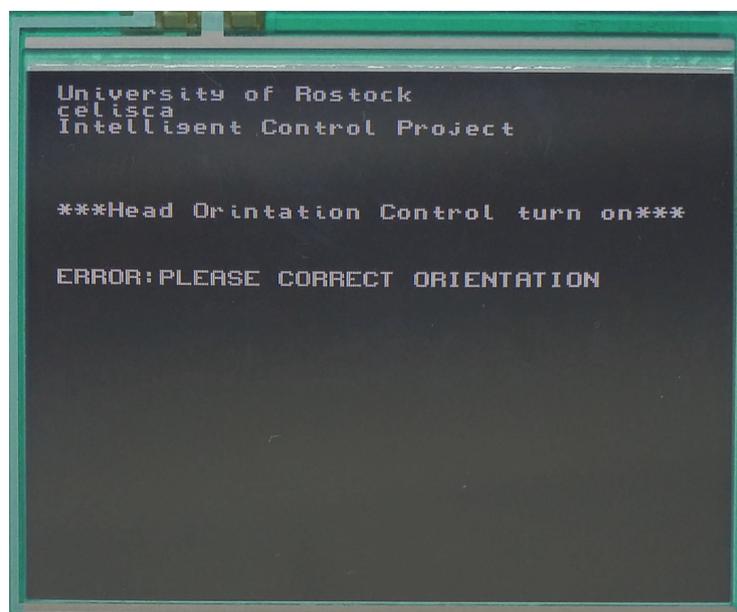


Figure 6-15: Wrong orientation notification

6.5.3 Orientation Sensor Error Handling Function

Orientation sensor error handling is one of the most important tasks in the present study since the system's motion can cause the loss of the instantaneous updating of orientation data. This error affects the system suddenly and not frequently. This error can happen because of the instantaneous loss of power in the orientation module due to strong vibration when the system passes bumpy roads especially, in outdoor environments or any problem stopped the orientation modules. This can affect the microcontroller performance and may cause restarting or hanging in the system. An error handling function has been programmed to avoid missing orientation data. The error handling function tracks the changes and the updating of the orientation data and sends an alert notification to the user if there are any communication problems with the orientation modules. It also resets and recalibrates the orientation sensors after a limited number of unchanged readings. Figure 6-16 shows the flowchart of the system's error handling function.

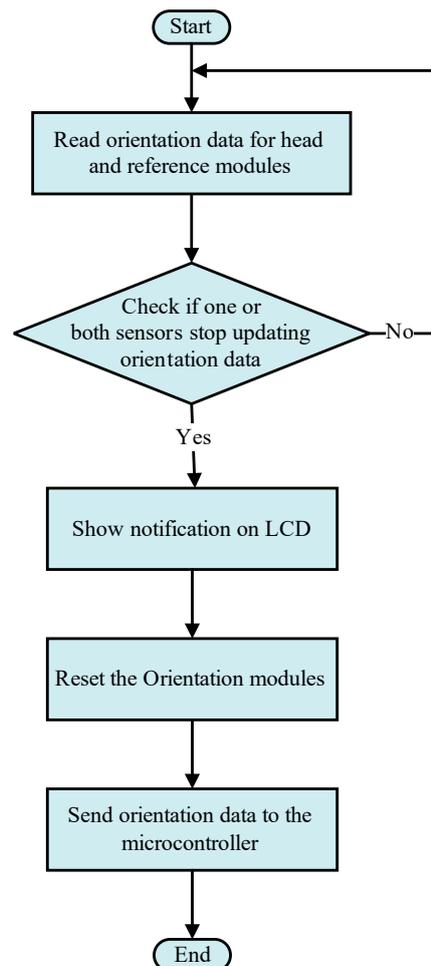


Figure 6-16: Flowchart for orientation modules error handling function

6.6 Conclusion for Head Tilt Controller

In this chapter, the realization and evaluation of the head tilt controller have been presented. An auto-calibrated head orientation controller and speed compensation algorithms for wheelchairs and rehabilitation applications are proposed. The system aims to help quadriplegic, paralyzed and elderly patients to control robotic-wheelchairs by using their head movements instead of traditional joystick controllers. The system employs a novel method to avoid errors in control commands due to any change in the user's head orientation and position in cases of passing along non-straight roads, which is an important feature allowing the use of the system in outdoor environments. The results show high control performance with accuracy levels of approximately 97.4%. The speed compensation algorithm adds another important speed calibration feature to the system. It keeps the wheelchair speed constant and gives more stability against gravitational force in ascents or descents during outdoor navigation.

The head tilt controller tests revealed that it can be used for both indoor as well as outdoor motion control. The indoor tests showed accurate performance with a successful implementation of tasks 1 and 2. The outdoor tests showed that the system is ideal for non-straight roads and especially for movement on hills with the auto-calibrated algorithm. The head tilt controller tests revealed that previous training of the user is required before using the head tilt controller to allow a better adaptation to the system and to enhance system performance.

The system has been tested with the Jaguar Lite robot and it can be used directly with Hawk and H20 robots (Dr. Robot Inc., Canada) which have the same controller PMS5005 that used in Jaguar light robot. The system can be used to control the movement of any robot or wheelchair after the modification of motor driver parameters.

The speed compensation algorithm is used to enhance the auto-calibrated head tilt controller by correcting the speed and direction of the wheelchair motors in cases of an ascent or descent up or down a ramp. This problem appears when the system passes along a ramp or not-straight road because of the effect of gravitational forces or in cases where the road has a side slope. One of the important goals of this work is to give the system more stability, which is important for quadriplegic and handicapped users of the system. The system test results show a good response and acceptable compensation time, with a maximum correction time of approximately 2.1 seconds. The speed compensation algorithm was tested with the Jaguar Lite robot because it has built-in magnetic encoders.

Conventional wheelchair motors do not have encoders, which have to be added to use this algorithm. It is still under development and needs further improvement in order to be used for wheelchair rehabilitation applications. The auto-calibrated head tilt controller has been tested with a Jaguar-Lite robot and shows excellent performance with command accuracy levels of more than 97.5%.

7 - Voice and Head Tilt Controllers Integration

7.1 Voice and Head Tilt Combination

The combination of the voice and head tilt sub-controllers produces a simple, flexible and comfortable controller that allows disabled users to select and change between the voice and head tilt controller easily. The voice controller can be activated at any time because it is used for both, motion and other stuff (lights, speed levels, and sound alarm) control commands while the head tilt controller will be activated only for motion control. The system starts by listening to the user's voice command to select which motion control type is required. The user must give the name of one of the controllers "Voice" or "Orientation" as a voice command. When the user selects the voice controller, the head tilt controller will be deactivated. The user can move his head in all directions without any control action affecting the system. The user can change the controller by giving the voice command "Orientation". In this case, the head tilt will control the speed, and the direction of the wheelchair-rehabilitation robot and the voice controller will be deactivated for speed and direction control, but it is still active for other signals like turn on and off lights, sound alarm and so on. Figure 7-1 shows the system flowchart for the transition between the voice and head tilt controllers.

The tests and results in chapters 4 and 5 revealed that the head tilt controller is more simple and comfortable for motion control because of the fast response time ($Tr < 100$ ms) compare with ($Tr \approx 1$ s) for voice controller. The voice controller can be used effectively in controlling parameter which cannot be covered by head tilt controller like, lights, signal lights, sound alarm and speed ratio control. The voice commands "voice" and "orientation" have been used to select or change the motion controller. Since the voice controller was tested before, the SD voice recognition mode has been selected to control the wheelchair, which is the best choice for rehabilitation application, to keep the wheelchair respond only to a specific user and not respond to any person giving the same trained commands. The voice commands "forward", "speed one" and "speed two" have been used to control the user's head tilt to speed ratio. This is important to let the user increase or decrease the system speed proportional to the head tilt angle value. The command "forward" makes the tilt to speed ratio from 0° - 25° tilt angles as 0-2

km/h wheelchair speed with ≈ 80 m/h speed for each tilt degree. The command “speed one” increases the speed ratio to $(0^\circ\text{-}25^\circ / 0\text{-}4$ km/h) with ≈ 160 m/h speed for each tilt degree. The command “speed two” increases the tilt to speed ratio to $(0^\circ\text{-}25^\circ / 0\text{-}6$ km/h) with ≈ 240 m/h speed for each tilt degree. Figure 7-2 shows the tilt to speed ratio of the tested voice commands.

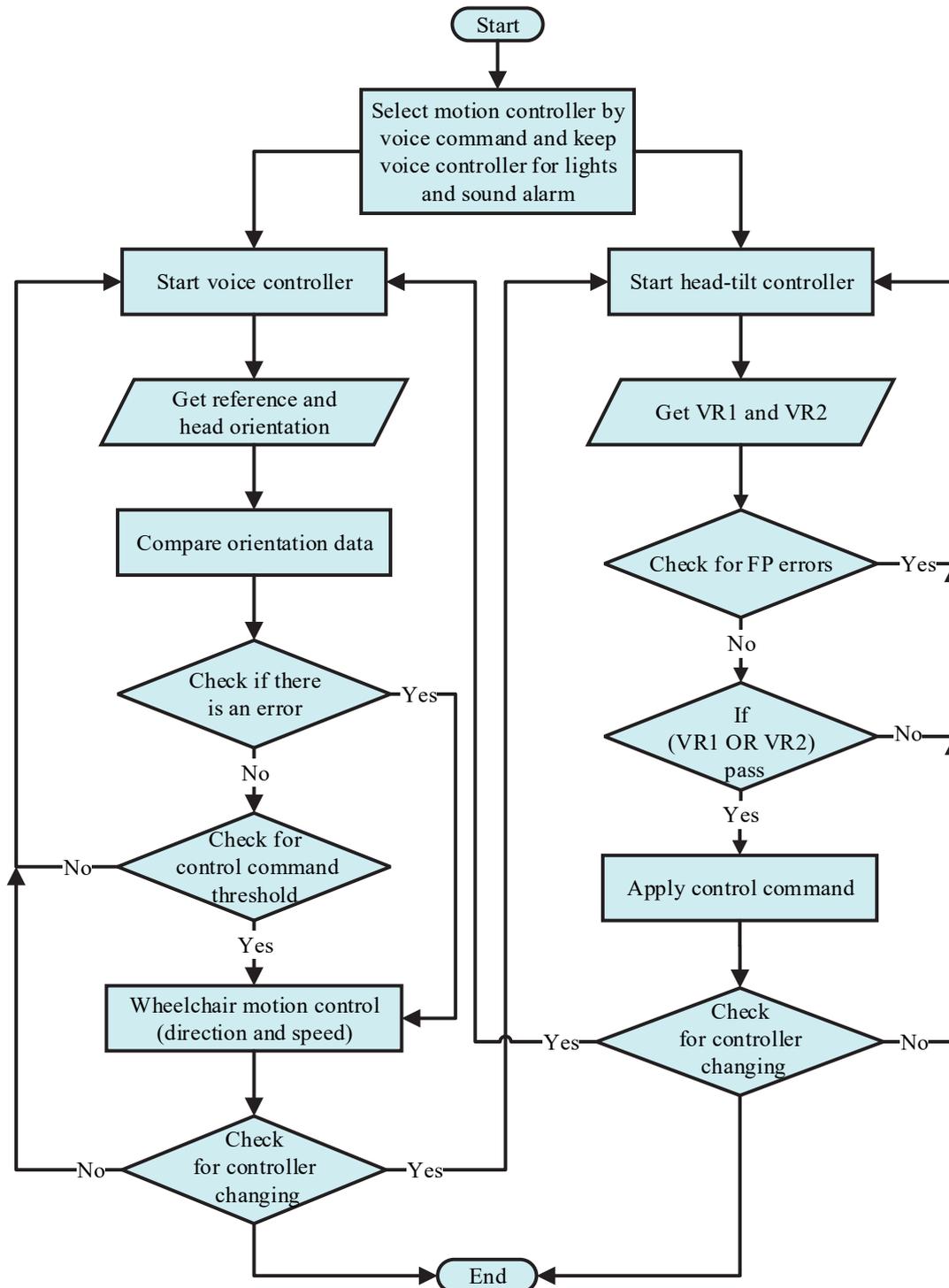


Figure 7-1: System flowchart for transition between the controllers

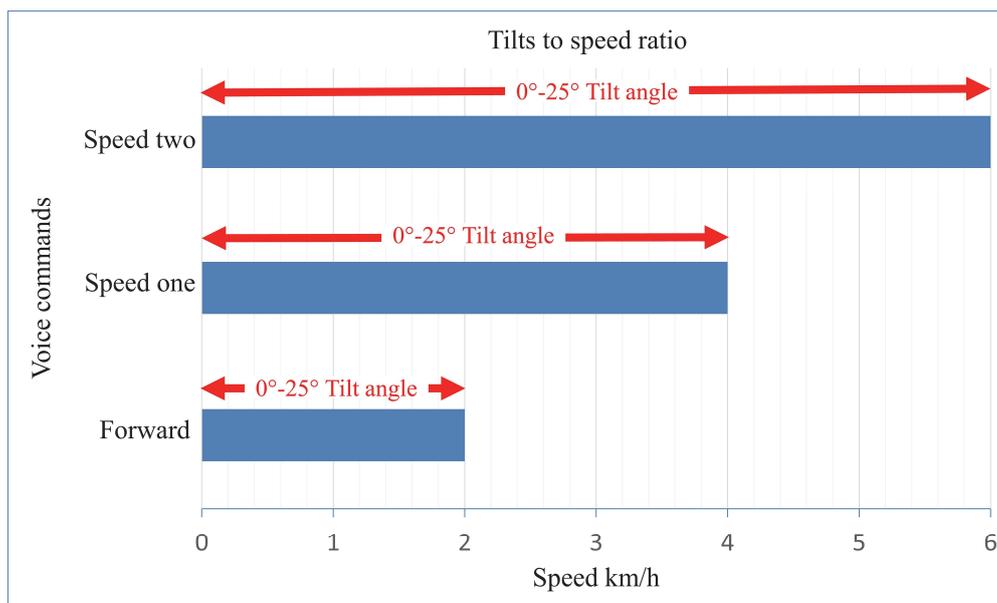


Figure 7-2: System tilt to speed ratio

7.2 Testing of Final System Integration

This test is aimed to check the performance of the combination of the two controllers (voice and head tilt). New voice commands have been tested in this test which are:

- “forward= set the tilt to speed to the basic ratio”,
- “speed one= increase the tilt to speed ratio”,
- “light on= turn lights on”,
- “light off= turn lights off” and “stop”.

The test has been taken by 14 users, each user repeated the test 10 times. The test includes different tasks to check the most important functions of the hybrid controller. The test tasks were implemented while the wheelchair was passing through a furnished narrow area (laboratory). In the first part of the test, the user must drive the wheelchair through a narrow corridor of about 1.5 m width. In this part, the user needs to change the tilt to speed ratio basic sensitivity (0°-25°=0-2 km/h) to the median ratio (0°-25°=0-4 km/h) by giving the voice command “speed one” (Task No. one T1). The user needs to use the voice command “light on” to turn on the wheelchair lights (T2) and the user also needs to return to the basic ratio when the wheelchair reaches the door of the laboratory by giving

also by using clear pronounced voice commands like replacing the similar pronounced commands speed one and speed two by different pronounced commands faster and slower which increases the VR accuracy to 100%.

Table 7-1: System test results

Task	Acc.	FP	FN
T1	95.71 %	0	6
T2	96.42 %	0	5
T3	95.71 %	0	6
T4*	100.00 %	0	0
T5*	100.00 %	0	0
T6	97.85 %	0	3
T7	96.42%	0	5

* Only head tilt controller has been used

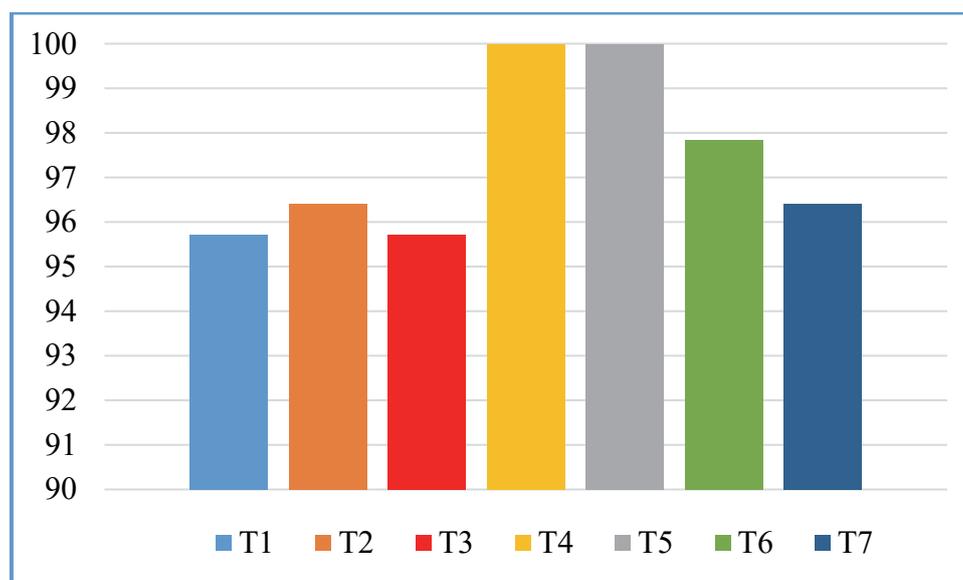


Figure 7-4: Voice commands accuracy test results

7.3 Time and Distance for Controllers Stop Command

Another Test has been used to measure both of the reaction time and the required distance to stop the wheelchair after giving the stop command using voice, head tilt, and joystick controllers. This test will deliver information

regarding the performance of the voice and head tilt controllers compared to the original wheelchair joystick controller. This test requires identical conditions for the test parameters, especially for the wheelchair speed, user weight, noise in the test environment and test place ground. The system speed has been fixed to ≈ 1.716 km/h for all the three controllers. The DT-2234C digital tachometer has been used for speed adjustment by calculating the wheel rotation per minute and fixing all the controllers at the same wheel rotation ratio which is 27.6 RPM. The user weight was 85 kg and the noise was ≈ 54 dB.

The reaction time and the required distance to stop the wheelchair after giving the stop command have been measured using the Casio EX-FH 100 high-speed camera (Casio Computer Co., Ltd., Tokyo, Japan) with up to 1000 frames per second (fps). Each controller has been tested for stop command 10 times and the average speed and the distance have been calculated from the picture frames of the time period starting from giving the command till the wheelchair completely stopped. The same procedure has been used for all the three controllers. The user needs to give the stop command using voice command “stop”, the head tilt stop command and the original joystick stop command. The user needs to give the stop voice command and lift his arm up at the same moment to indicate exactly the time and distance spent after giving the commands when the wheelchair reaches a marked point on the ground of the test place. In the head tilt controller test, the user needs to stop the wheelchair at the same marked point using his head orientation by returning the head to the stop area of the head tilt controller ($-10^{\circ} \leq P \leq 0^{\circ}$, $-5^{\circ} \leq R \leq 5^{\circ}$). Two measurements have been taken after giving the stop command which are 1) the time required to stop the wheelchair after giving the command and 2) the distance required to stop the wheelchair after giving the stop command. Figure 7-5 shows the calculation procedure of reaction time for the three controllers.

The test results revealed that the voice controller has the highest reaction time of $\approx 1,551$ ms and it required a ≈ 476.65 mm distance to stop the wheelchair. The joystick controller has a stop reaction time of ≈ 827.91 ms and it required ≈ 243.44 mm to stop the wheelchair. The head tilt controller has the shortest reaction time of ≈ 666.60 ms and it required the shortest distance of ≈ 171.2 mm to stop the wheelchair. Figure 7-6 explains the comparison between the three controllers.



a) Head tilt controller

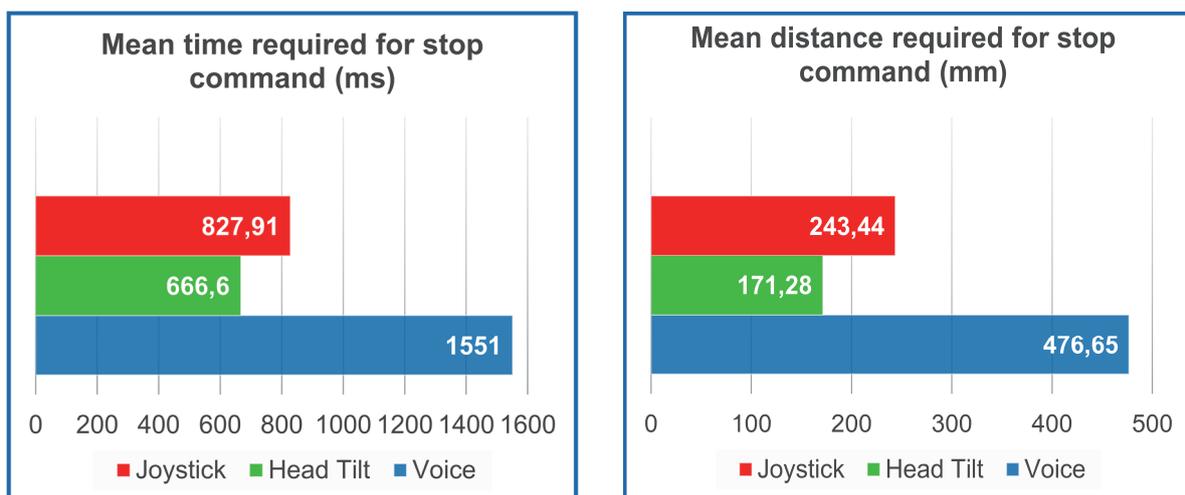


b) Joystick controller



c) Voice controller

Figure 7-5: Stop command reaction time for sub-controllers



a) Stop time

b) Stop distance

Figure 7-6: Reaction time and distance for stop command

The test results show that the head tilt and the joystick controller have an approximately similar range of performance compared to the voice controller, which has a long reaction time and also it required more distance to stop the wheelchair. The reaction time of the voice controller is longer due to the voice recognition process, as well as the FP cancellation function consumes time that reflexes in this test result.

7.4 Questionnaire for System

For the evaluation of the system usability, fourteen users have been asked seven questions about their feeling, adaptation, and opinions about the system. Each question should get a numerical evaluation from 0 minima up to 10 maximums. Table 7-2 summarizes the evaluation questions. Figure 7-7 and Table 7-3 explain the questionnaire results.

Table 7-2: Questionnaire evaluation questions

Question No.	Question
Q1	Is it easy to use head tilt controller?
Q2	How do you find the comfortability of wearing a headset?
Q3	How do you find the reaction time for head tilt controller?
Q4	How do you find the reaction time for voice controller?
Q5	How do you find the voice command response?
Q6	Is it easy to pass narrow area?
Q7	Is it easy to control lights and signals?

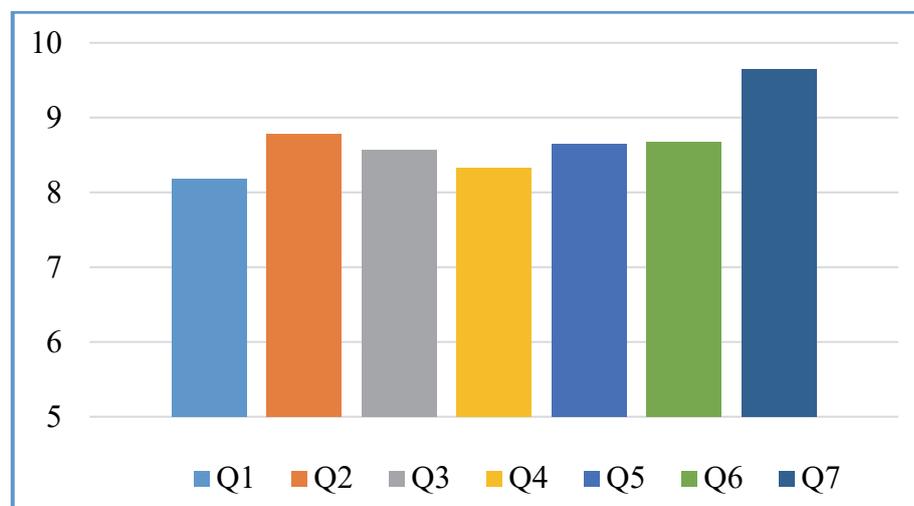


Figure 7-7: Questionnaire result for the system

Table 7-3: Questionnaire results

User	Q1	Q2	Q3	Q4	Q5	Q6	Q7
U1	7,5	8	7,5	7,5	9	10	10
U2	7,5	8	7	9	7,5	10	10
U3	9	10	10	8	10	7,5	10
U4	8,5	9	8	7,5	9	9	10
U5	7	8	7,5	7	7,5	8	9
U6	7,5	9	10	9	10	8	10
U7	9	8	10	10	8	10	10
U8	9	8	9	10	8	8	10
U9	8	9	10	9	10	8	10
U10	9	10	10	8	10	8	10
U11	8,5	9	8	7,5	9	9	10
U12	8	10	8	8	8	9	9
U13	9	9	8	8,5	7	10	8
U14	7,5	8	7	8	8	7,5	9
Mean	8,214	8,785	8,571	8,3573	8,642	8,714	9,642
SDV	0,726	0,801	1,206	0,928	1,063	0,974	0,633

The abbreviation SDV refers to the standard deviation. The results of the questionnaire show differences in the system evaluation between the users. The heights mean value with the minimum standard deviation goes to the (Q7); most of the users found the control of lights and signal using voice controller is easy. The highest difference between the users' answers has a standard deviation of 1.206 for (Q3). The answer to this question may get this difference because the users not previously informed about the command confirmation function of the head tilt controller that makes a 100 ms delay in the starting of each motion command.

7.5 Conclusion

In the presented work, a hybrid wheelchair controller for handicapped and quadriplegic patients is presented. The system has two control methods to control the motion and other system parameters, which are the voice controller and the head tilt controller. The two controllers help the quadriplegia, elderly and handicap patients to control the robot-wheelchairs by the available body signals for this kind of users like user voice and user's head tilt as input control methods instead of traditional joystick controller. The two controllers have novel implementation and improvement compared with the previous research work in this domain. The results show good improvement in the VR accuracy by using OR-FP algorithms which allow the voice controller to avoid and strongly reduce the FP errors.

The combination of the two sub-controllers in one system enables the user to control all the function and service in the wheelchair similar to the original joystick and buttons. The validating tests revealed that the head tilt controller has the shortest stop command reaction time ≈ 666.6 ms compare with the original joystick ≈ 827.91 ms and the voice controller $\approx 1,551$. This result guides us to select the head tilt controller as the main controller for the system, especially in indoor environments. The voice controller can be used in outdoor as well as in low noise environments <60 dB.

The system design takes into consideration future requirement, and it can be easily modified by adding any new control unit or sensors by interfacing them through the GPIO, SPI, I²C and UARTs ports of the ARM Microcontroller.

The system was tested successfully with a wheeled Jaguar light robot and Meyra Smart 9.916 wheelchair and can also be used directly with H2O and Hawk robots. The system can be used for any wheelchair or rehabilitation robots after modifying the drivers, motors and wheel parameters.

8 - Summary and Outlook

8.1 Summary

Nowadays a significant increase in the number of older persons can be seen in all countries of the world. This part of society as well as patients with quadriplegia, permanently paralyzed, and/or amputated arms need to use special control systems to control the rehabilitation robots and wheelchairs instead of using traditional electrical wheelchairs, which are controlled by a joystick. The system design and development must take into consideration the available control signal for this kind of patients. The most important control signal can be received from the region of shoulder, neck, and head of the user. Some of the useful signals in this area are the voice, head orientation, Electroencephalogram (EEG), Electromyogram (EMG) and Electrooculogram (EOG).

In this dissertation, a multi-input hybrid control system for rehabilitation application is proposed. The multi-input makes the system more flexible to adapt to the available body signals, which are used as control signals for the rehabilitation and medical applications. The first input is a voice controller unit. It includes voice controllers with two modes of operation to maximize the voice recognition accuracy and reduce the voice recognition errors. The voice controller has a novel implementation using two different voice recognition modules which add an intelligent feature to the system that allows it to identify wrong command recognition and also increases the voice recognition probability. The second input is the head orientation control unit. It uses the user head movements around the x, y, and z-axes to control the wheelchair movement and speed in all directions. This control unit includes two 9 DOF (degree of freedom) orientation modules. Each module has three MEMS sensors (micro-electro-mechanical systems), which accelerometer, gyroscope, and magnetometer are combined in one module to give proper orientation data in the format of Euler angles. An auto-calibrated algorithm has been embedded into the head tilt controller to calibrate the control of movement and speed when the wheelchair passes through a hill or non-straight roads. Other functions such as command confirmation function, wrong orientation handling function, and orientation sensor error handling functions have been added to the head tilt controller to enhance the performance of the controller and to add more safety to the system users. The low-cost design is taken into consideration as it allows more patients to use this system.

8.2 Comparison of Selected Solutions with Previous Work

Based on the available state-of-the-art solutions to control electrical wheelchairs without using a traditional joystick controller, two types of controller are selected in this study for use by quadriplegic patients, which are the voice controller and the head tilt controller. Most previous work has used one or more computer to process and classify the acquired signals. This adds high costs and complexity to the system. In the present work, a powerful ARM microcontroller has been selected to use instead of computers to reduce the system's cost. Most body signals such as EEG, EMG, and EOG are profoundly affected by electrical interference from the user's body and power sources. Special electrodes in contact with the body need to be used to pick up these signals and this makes this approach uncomfortable for the patient. The small electrical values of these signals make their processing and use more complicated as control signals for wheelchairs and rehabilitation systems. The user's voice and head motions can instead be used effectively for this control by quadriplegics and paralyzed patients.

8.2.1 Voice Recognition

Compared to previous research, this work is innovative in the sense that it uses two voice recognition modules built based on two different voice recognition algorithms, dynamic time warping (DTW) and a hidden Markov model (HMM), as a hybrid primary voice controller. This controller can work with speaker dependent (SD) and speaker-independent (SI) modes. The use of these two VR modules in one control unit enhances allow the system to detect the wrong command recognition and improve the VR accuracy by selecting the most accurate recognition results and reducing the VR errors. A microcontroller program has been written to include a false positive (FP) error cancellation function. This works stand-alone, without the need to use a computer to perform data processing and there is no need to fit complex, sensitive electrodes onto the user's body. The proposed system can operate in any language in speaker-dependent mode, and works with only eight global languages in speaker- independent mode.

8.2.2 Orientation Detection

In this work, an auto-calibrated head tilt controller and speed compensation algorithms for wheelchair and rehabilitation robotics are proposed. The system uses orientation data of the Euler angles of head tilt movements as a controller for the intelligent application. The movement of the user's head around the x- and y- axes is translated into motion in the forward, backward, left and right directions.

The system can adjust its speed depending on the slope of the road. The system uses Euler angles for pitch, roll, and yaw to detect the head's orientation. The Euler angles are picked up using three MEMS sensors which are an accelerometer, gyroscope, and magnetometer. The three sensors are combined to build a highly accurate orientation sensor. The system uses two orientation sensors, the first being fixed to the wheelchair chassis to supply a reference orientation. The second sensor is fixed in a wearable headset on the user's head to detect head movements and give orientation data to the microcontroller.

Compared to previous work, the proposed head tilt controller uses two control algorithms to enhance its performance. The first is an auto-calibrated algorithm which is used to avoid the effect of road slope in case of the wheelchair climbing a hill or traveling along non-straight roads. It prevents any incorrect command when the head position enters the control threshold area because of the change in head position due to the slope or curvature of the road. It has the flexibility to perform the head tilt control when the system is used in outdoor environments with non-straight roads. The algorithm gives the flexibility to the user to change the speed of the wheelchair in all directions from the minimum to the maximum speed depending on the head tilt angle. The use of Euler angles makes the system more accurate and enhances the sensitivity of the controller.

8.3 Outlook

There are several ideas and suggestions to improve the system performance. One suggestion is to add magnetic or optical encoders to the wheelchair motors. This will allow the system to measure the real speed of the wheelchair motors and compare it to the speed given by the control commands. The speed feedback will allow the system to compensate the difference between the intended and measured/real speed. Further, encoders will allow to add a closed loop PID controller to the wheelchair and apply a speed compensation algorithm to compensate the speed changes in case of ascending or descending ramps due to gravity forces. The implementation of speed compensation algorithm with the wheelchair will improve the performance of the head tilt controller via auto adjustment of the required speed for each individual motor in case of ascending and descending ramps and hills.

The communication between the head orientation sensor and the main microcontroller can be changed to wireless by using Bluetooth modules. This will increase the ease and comfortability of wearing and take off the headset.

Adding a robotic arm can be another important improvement for the proposed system especially if is designed for peoples with useless arms. The robotic arm can be fixed on the wheelchair chassis and can be controlled using voice commands. The robotic arm can be used by the user for implementing simple daily tasks such as pick up an object, pressing buttons (electrical switches for light or elevator buttons) and so on.

The performance of the voice controller can be further improved by adding a new VR module. The new module will increase the probability of voice command recognition and decrease the chance of FP errors to \approx zero.

Through the technical test of the system, we find that the user weight is affecting the system speed inversely so that when the weight increased the speed decreased. We suggest adding a weight sensor array to measure the user weight and then calibrate the speed ratio of the system motors to compensate the speed depending on the measured user weight.

Another suggestion is to add an obstacle detection unit. It is a necessary unit for systems to be used by paralyzed patients. This unit will improve the safety of the system and protect the user in cases of any problems or errors which affect the system. There are several types of obstacle detection and avoidance approaches that can be adopted in the system. The present study does not focus on this unit as it is already available. The selection of the type of obstacle detection unit is related to the processor unit used in the system. The use of cameras and vision sensors for obstacle detection means that a high-speed processor is required and can be implemented using computers or tablets. Ultrasound sensors can be effectively used with the microcontroller unit and it is necessary to adopt more than one sensor in the system to cover all directions due to the sensor coverage area. The ultrasound sensor can be used to detect an obstacle from 2 to 400 cm distance and it can check if there are pitfalls in the wheelchair's path at the same range [213], [214]. The use of eight ultrasound sensors fixed on the front, back and corners of the wheelchair can cover all the directions that the wheelchair may take. Each sensor can be adjusted by the host processor to detect a specific distance based on the design requirements.

References

- [1] “Spinal Cord Injury Statistics,” Spinal Cord Injury Support - Paraplegic and Quadriplegic. [Online]. Available: <http://www.apparelyzed.com/statistics.html>. [Accessed: 15-Jan-2016].
- [2] “Quadriplegia and Paraplegia Information and Infographic,” Disabled World. [Online]. Available: <http://www.disabled-world.com/disability/paraquad.php>. [Accessed: 15-Jan-2016].
- [3] Robert Jarchi, “What is Quadriplegia C1-C4 Complete?” [Online]. Available: <https://www.youtube.com/watch?v=svu4bV0SyrU>. [Accessed: 20 January 2017].
- [4] Zuni, “Quadriplegia & Paraplegia,” [Online]. Available: <http://www.slideshare.net/zuni1412/3-quadriplegia-paraplegia>, [Accessed: 05 January 2017].
- [5] “Quadriplegia | Find all Inability Adaptation Kit,” TITAN. | Find all Inability Adaptation Kit. [Online]. Available: <http://www.association-titan.fr/la-tetraplegie/>. [Accessed: 09-Apr-2016].
- [6] Meyra Smart 9.906 Wheelchair, “Elektromobil Meyra Smart Model 9.906 6km/h *09.” [Online]. Available: <http://www.ebay.de/itm/Elektromobil-Meyra-Smart-Model-9-906-6km-h-09-/262838577617>. [Accessed: 09-Mar-2017].
- [7] L. J. Pinto, D. H. Kim, J. Y. Lee, and C. S. Han, “Development of a Segway robot for an intelligent transport system,” in Proc. of 2012 IEEE International Symposium on System Integration (SII), Fukuoka, Japan, 2012, pp. 710–715.
- [8] M. H. Khan, M. S. Chaudhry, T. Tariq, Q. u A. Fatima, and U. Izhar, “Fabrication and modelling of Segway,” in Proc. of 2014 IEEE International Conference on Mechatronics and Automation (ICMA), Tianjin, China, 2014, pp. 280–285.
- [9] M. U. Draz, M. S. Ali, M. Majeed, U. Ejaz, and U. Izhar, “Segway electric vehicle,” in Proc. of 2012 International Conference on Robotics and Artificial Intelligence (ICRAI), Islamabad, Pakistan, 2012, pp. 34–39.
- [10] “Segway-Like Mobility Device Gets Paraplegics Around Easily | Gadgets, Science & Technology.” [Online]. Available: <http://gajitz.com/segway-like-mobility-device-gets-paraplegics-around-easily/>. [Accessed: 03-Mar-2016].
- [11] K. Anam and A. A. Al-Jumaily, “Active Exoskeleton Control Systems: State of the Art,” in Proc. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012), Sarawak, Malaysia, 2012, pp. 988–994.
- [12] D. Wang, K.-M. Lee, J. Guo, and C. Yang, “Adaptive Knee Joint Exoskeleton Based on Biological Geometries,” IEEE/ASME Transactions on Mechatronics, Vol. 19, no. 4, pp. 1268–1278, 2014.
- [13] C. Fleischer, A. Wege, K. Kondak, and G. Hommel, “Application of EMG signals for controlling exoskeleton robots,” Biomedical Engineering, Vol. 51, No. 5_6, pp. 314–319, 2006.

-
- [14] D. Miranda-Linares, G. Alrezage, and M. O. Tokhi, "Control of lower limb exoskeleton for elderly assistance on basic mobility tasks," in Proc. of 2015 19th International Conference on System Theory, Control and Computing (ICSTCC), Cheile Gradistei, Romania, 2015, pp. 441–446.
- [15] A. Gams, T. Petrič, T. Debevec, and J. Babič, "Effects of Robotic Knee Exoskeleton on Human Energy Expenditure," IEEE Transactions on Biomedical Engineering, Vol. 60, No. 6, pp. 1636–1644, 2013.
- [16] R. Huang, H. Cheng, Q. Chen, H.-T. Tran, and X. Lin, "Interactive learning for sensitivity factors of a human-powered augmentation lower exoskeleton," in Proc. of 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 2015, pp. 6409–6415.
- [17] D. Liu, Z. Tang, and Z. Pei, "The motion control of lower extremity exoskeleton based on RBF neural network identification," in Proc. of 2015 IEEE International Conference on Information and Automation, Changshu, China, 2015, pp. 1838–1842.
- [18] E. Palermo, "New Robotic Exoskeleton Is Controlled by Human Thoughts," LiveScience.com. [Online]. Available: <http://www.livescience.com/51940-mind-controlled-exoskeleton-robot.html>. [Accessed: 03-Mar-2016].
- [19] J. Baumann, "VoiceRecognition", [Online]. Available: www.hitl.washington.edu/scivw/EVE/I.D.2.d.VoiceRecognition.html/. [Accessed: 20-May-2016].
- [20] "HM2007 Data sheet", [Online]. Available: www.imagesco.com/. [Accessed: 20-Mar-2016].
- [21] R. H. Rockland and S. Reisman, "Voice activated wheelchair controller," in Proc. of IEEE 24th Annual Northeast Bioengineering Conference, Pennsylvania, USA, 1998, pp. 128–129.
- [22] M. F. Ruzaj and S. Poonguzhali, "Design and implementation of low cost intelligent wheelchair," in Proc. of 2012 International Conference on Recent Trends In Information Technology (ICRTIT), Chennai, India, 2012, pp. 468–471.
- [23] C. Aruna, A. Dhivya Parameswari, M. Malini, and G. Gopu, "Voice recognition and touch screen control based wheel chair for paraplegic persons," in Proc. of 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, 2014, pp. 1–5.
- [24] T. Kubik and M. Sugisaka, "Use of a cellular phone in mobile robot voice control," in Proc. of the 40th Annual Conference SICE 2001. International Session Papers, Nagoya, Japan, 2001, pp. 106–111.
- [25] M. Sajkowski, "Voice control of dual-drive mobile robots-survey of algorithms," in Proc. of the Third International Workshop on Robot Motion and Control, 2002. RoMoCo '02, Poznań, Poland, 2002, pp. 387–392.
- [26] P. Cusi, J.-P. Hosom, J. Shalkwyk, S. Sutton, and R. A. Cole, "Connected digit recognition experiments with the OGI Toolkit's neural network and HMM-based recognizers," in Proc. of 1998 IEEE 4th Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA '98). Torino, Italy, 1998, pp. 135–140.

-
- [27] L. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, Vol. 77, No. 2, pp. 257–286, 1989.
- [28] "digital processing of speech signals (rabiner & schaffer 1978).pdf," Google Docs, 09-Dec-2015. [Online]. Available: <https://drive.google.com/file/d/0B8lDery-ZA7XMU9MS09aLUc2WGM/view>. [Accessed: 09-Dec-2015].
- [29] D. B. Roe and J. G. Wilpon, "Voice Communication between Humans and Machines," Washington, D.C.: National Academies Press, 1994.
- [30] R. C. Simpson and S. P. Levine, "Adaptive shared control of a smart wheelchair operated by voice control," in *Proc. of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '97)*, Grenoble, France, 1997, pp. 622–626.
- [31] S. Grochowski, "Modeling of the speech units for Polish: PROSODY 2000," in *Proc. of Speech Recognition and Synthesis Workshop*, Krakow, Poland, 2000, pp. 2-5.
- [32] Theo J. A. de Vries, C. V. Heteren and L. Huttenhuis "Modeling and Control of a Fast Moving, Highly Maneuverable Wheelchair," in *Proc. of the International Biomechatronics Workshop*, Enschede, Netherlands, 1999, pp. 110-115.
- [33] L. Rabiner and B. H. Juang, "An introduction to hidden Markov Models," *IEEE Acoustics, Speech, and Signal Processing (IEEE ASSP) Magazine*, Vol. 3, No. 1, pp. 4–16, 1986.
- [34] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system," in *Proc. of 1990 International Conference on Acoustics, Speech, and Signal Processing (ICASSP-90)*, Albuquerque, New Mexico, USA, 1990, pp. 129–132.
- [35] A. Chatterjee, K. Pulasinghe, K. Watanabe, and K. Izumi, "A particle-swarm-optimized fuzzy-neural network for voice-controlled robot systems," *IEEE Transactions on Industrial Electronics*, Vol. 52, No. 6, pp. 1478–1489, 2005.
- [36] L. Rabiner and B.-H. Juang, "Fundamentals of Speech Recognition," Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1993.
- [37] S. Furui, "Digital Speech Processing, Synthesis, and Recognition", New York, Marcel Dekker Inc., 2000.
- [38] P. Zegers, "Speech recognition using neural networks," M.S. thesis, Faculty of the Electrical and Computer Engineering, University of Arizona, 1998.
- [39] I. Kirschning, H. Tomabechi, M. Koyama, and J.-I. Aoe, "The time-sliced paradigm—a connectionist method for continuous speech recognition," *Information Sciences*, Vol. 93, No. 1–2, pp. 133–158, 1996.
- [40] G. Pacnik, K. Benkic, and B. Brecko, "Voice operated intelligent wheelchair - VOIC," in *Proc. of the IEEE International Symposium on Industrial Electronics (ISIE 2005)*, Dubrovnik, Croatia, 2005, pp. 1221–1226.
- [41] X. Lv, M. Zhang, and H. Li, "Robot control based on voice command," in *Proc. of IEEE International Conference on Automation and Logistics (ICAL 2008)*, Qingdao, China, 2008, pp. 2490–2494.
- [42] H. R. Singh, A. Mobin, S. Kumar, S. Chauhan, and S. S. Agrawal, "Design and development of voice/joystick operated microcontroller based intelligent

- motorised wheelchair,” in Proc. of the IEEE Region 10 Conference (TENCON 1999), Cheju Island, Korea, pp. 1573–1576.
- [43] “A prototype Lab Box with DSK 'C6711/13 for rapid DSP algorithm development | Bertini G.,” Europeana. [Online]. Available: http://www.europeana.eu/portal/record/2023823/istituti_ProdottoDellaRicerca_html_cds_074_id_160898.html. [Accessed: 09-Dec-2015].
- [44] M. T. Qadri and S. A. Ahmed, “Voice Controlled Wheelchair Using DSK TMS320C6711,” in Proc. of International Conference on Signal Acquisition and Processing (ICSAP 2009), Kuala Lumpur, Malaysia, 2009, pp. 217–220.
- [45] T. Kawahara and A. Lee, “Open-Source Speech Recognition Software Julius,” Journal of Japan. Society of Artificial Intelligent, Vol. 20, No. 1, pp. 41–49, 2005.
- [46] “AquesTalk pico/AquesTalk2/AquesTalk,” [Online]. Available: <http://www.a-quest.com/aquestalk/index.html/>. [Accessed: 09-Dec-2016].
- [47] A. Murai, M. Mizuguchi, M. Nishimori, T. Saitoh, T. Osaki, and R. Konishi, “Voice activated wheelchair with collision avoidance using sensor information,” in Proc. of International Joint Conference (ICCAS-SICE 2009), , Fukuoka, Japan, 2009, pp. 4232–4237.
- [48] M. Nishimori, T. Saitoh, and R. Konishi, “Voice controlled intelligent wheelchair,” in Proc. of SICE Annual Conference 2007, Kagawa, Japan, 2007, pp. 336–340.
- [49] A. Murai, M. Mizuguchi, T. Saitoh, T. Osaki, and R. Konishi, “Elevator available voice activated wheelchair,” in proc of the 18th IEEE International Symposium on Robot and Human Interactive Communication, Toyama, Japan ,2009, pp. 730–735.
- [50] U. Qidwai and F. Ibrahim, “Arabic speech-controlled wheelchair: A fuzzy scenario,” in Proc. of 2010 10th International Conference on Information Sciences Signal Processing and their Applications (ISSPA), Kuala Lumpur, Malaysia, 2010, pp. 153–156.
- [51] Y.-H. Jeon and H. Ahn, “A multimodal ubiquitous interface system using smart phone for human-robot interaction,” in 2011 8th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Ncheon, South Korean, 2011, pp. 764–767.
- [52] R. Phoophuangpaioj, “Using multiple HMM recognizers and the maximum accuracy method to improve voice-controlled robots,” in Proc. of 2011 International Symposium on Intelligent Signal Processing and Communications Systems (ISPACS), Chiang Mai, Thailand, 2011, pp. 1–6.
- [53] EasyVR 2.0 User Manual R.3.6.6., TIGAL KG, Vienna, Austria, 2014.
- [54] U. Qidwai and M. Shakir, “Ubiquitous Arabic voice control device to assist people with disabilities,” in 2012 4th International Conference on Intelligent and Advanced Systems (ICIAS), Kuala Lumpur, Malaysia, 2012, pp. 333–338.
- [55] J. Kathirvelan, R. Anilkumar, Z. C. Alex, and A. Fazul, “Development of low cost automatic wheelchair controlled by oral commands using standalone controlling system,” in Proc. of 2012 IEEE International Conference on Computational Intelligence Computing Research (ICCIC), Coimbatore, India, 2012, pp. 1–4.

-
- [56] “Sound Spectrum Pro Screenshots.” [Online]. Available: http://www.lindentree.eu/sounds_shots.php. [Accessed: 11-Apr-2016].
- [57] R. Huang and G. Shi, “Design of the control system for hybrid driving two-arm robot based on voice recognition,” in Proc. of 2012 10th IEEE International Conference on Industrial Informatics (INDIN), Beijing, China, 2012, pp. 602–605.
- [58] B.-K. Shim, K. Kang, W.-S. Lee, J.-B. Won, and S.-H. Han, “An intelligent control of mobile robot based on voice command,” in Proc. of 2012 12th International Conference on Control, Automation and Systems (ICCAS), Jeju Island, South Korea, 2012, pp. 1878–1881.
- [59] E. T. P. Santos and I. A. Cestari, “Command interface and driving strategy for a voice activated en-doscope positioning arm,” in Proc. of 2014 5th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics, São Paulo, Brazil, 2014, pp. 66–69.
- [60] M. Müller, “Dynamic time warping,” *Information retrieval for music and motion*, Ch. 4, pp. 69–84, 2007.
- [61] V. Tiwari, “MFCC and its applications in speaker recognition,” *International Journal on Emerging Technologies*, Vol. 1, No. 1, pp. 19–22, 2010.
- [62] S. C. Sajjan and C. Vijaya, “Comparison of DTW and HMM for isolated word recognition,” in Proc. of 2012 International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME), Salem, India, 2012, pp. 466–470.
- [63] Ran Yaniv and D. Burshtein, “An enhanced dynamic time warping model for improved estimation of DTW parameters,” *IEEE Transactions on Speech and Audio Processing*, Vol. 11, No. 3, pp. 216–228, 2003.
- [64] T. Giorgino and others, “Computing and visualizing dynamic time warping alignments in R: the dtw package,” *Journal of Statistical Software*, Vol. 31, No. 7, pp. 1–24, 2009.
- [65] D. Gaoming, X. Di, H. Wei, D. Zhantao, and L. Wenlong, “Dynamic Time Warp Based Template Analysis,” in Proc. of 2012 Second International Conference on Instrumentation, Measurement, Computer, Communication and Control, Harbin, China, 2012, pp. 470–473.
- [66] C. A. Ratanamahatana and E. Keogh, “Everything you know about dynamic time warping is wrong,” in Proc. of Third Workshop on Mining Temporal and Sequential Data, Seattle, Washington, USA, 2004, pp.1–11.
- [67] C. Myers, L. R. Rabiner, and A. E. Rosenberg, “Performance tradeoffs in dynamic time warping algorithms for isolated word recognition,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, Vol. 28, No. 6, pp. 623–635, 1980.
- [68] T. Zaharia, S. Segarceanu, M. Cotescu, and A. Spataru, “Quantized dynamic time warping (DTW) algorithm,” in Proc. of 2010 8th International Conference on Communications (COMM 2010), Bucharest, Romania, 2010, pp. 91–94.
- [69] S. C. Sajjan and C. Vijaya, “Comparison of DTW and HMM for isolated word recognition,” in Proc. of 2012 International Conference on Pattern Recognition, Informatics and Medical Engineering (PRIME), Salem, India, 2012, pp. 466–470.

-
- [70] P. Blunsom, "Hidden markov models," Lect. Notes, Vol. 15, pp. 1–7, 2004.
- [71] A. A. Markov, "An Example of Statistical Investigation in the Text of "Eugene Onyegin" Illustrating Coupling of Tests in Chains," In Proc. Academy of Sciences of St. Petersburg, 7, pp. 153–162, 1913.
- [72] L. E. Baum, T. Petrie, G. Soules, and N. Weiss, "A maximization technique occurring in the statistical analysis of probabilistic functions of Markov chains. Ann Math Stat," The Annals of Mathematical Statistics, Vol. 41, No. 1, pp. 154–171, 1970.
- [73] A. Atwi, K. Savla, and M. A. Dahleh, "A case study in robust quickest detection for hidden markov models," in Proc. of American Control Conference (ACC 2011), San Francisco, California, USA, 2011, pp. 780–785.
- [74] Y. Jia, L. Sun, and H. Teng, "A comparison study of hidden Markov model and particle filtering method: Application to fault diagnosis for gearbox," in Proc. of 2012 IEEE Conference on Prognostics and Health Management (PHM), Denver, Colorado, USA, 2012, pp. 1–7.
- [75] S. Zhang et al., "A Markov Chain Model with High-Order Hidden Process and Mixture Transition Distribution," in Proc. of 2013 International Conference on Cloud Computing and Big Data, Fuzhou, Fujian, China, 2013, pp. 509–514.
- [76] Quick T2SI Toolkit for the RSC-4x family, Product brief, [Online]. Available: <http://www.sensoryinc.com/support/docs/80-0245-C.pdf>. [Accessed: 28-Nov-2014].
- [77] K. Ogino and W. Kozak, "Spectrum analysis of surface electromyogram (EMG)," in Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP '83), Boston, Massachusetts, USA, 1983, pp. 1114–1117.
- [78] G. Hefftner and G. G. Jaros, "The electromyogram (EMG) as a control signal for functional neuromuscular stimulation. II. Practical demonstration of the EMG signature discrimination system," IEEE Transactions on Biomedical Engineering, Vol. 35, No. 4, pp. 238–242, 1988.
- [79] R. N. Khushaba, S. Kodagoda, D. Liu, and G. Dissanayake, "Electromyogram (EMG) based fingers movement recognition using Neighborhood Preserving Analysis with QR-decomposition," in Proc. of 2011 Seventh International Conference on Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP), Adelaide, Australia, 2011, pp. 1–6.
- [80] K. Choi and S. Hiki, "Possibility of automatizing computer program for pulse density demodulation (PDD) processing of surface electrode electromyograms (EMG)," in Proc. of the 20th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Hong Kong Sar, China, 1998, pp. 2635–2638.
- [81] M. Yoshikawa, M. Mikawa, and K. Tanaka, "Real-Time Hand Motion Estimation Using EMG Signals with Support Vector Machines," in Proc. of SICE-ICASE, 2006. International Joint Conference, Busan, South Korea, 2006, pp. 593–598.
- [82] S. Kouchaki, R. Boostani, S. Shabani, and H. Parsaei, "A new feature selection method for classification of EMG signals," in Proc. 2012 16th CSI International Symposium on Artificial Intelligence and Signal Processing (AISP), Shiraz, Iran, 2012, pp. 585–590.

-
- [83] R. N. Khushaba and S. Kodagoda, "Electromyogram (EMG) feature reduction using Mutual Components Analysis for multifunction prosthetic fingers control," in Proc. of 2012 12th International Conference on Control Automation Robotics Vision (ICARCV), Guangzhou, China, 2012, pp. 1534–1539.
- [84] P. Geethanjali and K. K. Ray, "A Low-Cost Real-Time Research Platform for EMG Pattern Recognition-Based Prosthetic Hand," *IEEE/ASME Transactions on Mechatronics*, Vol. 20, No. 4, pp. 1948–1955, 2015.
- [85] W. Ikehara, T. Uchida, K. Choi, and S. Hiki, "Computer programmed pulse density demodulation (PDD) processing for surface electrode electromyograms (EMG) of human extremities," in Proc. of IEEE 17th Annual Conference of the Engineering in Medicine and Biology Society, Montreal, Canada, 1995, pp. 1349–1350.
- [86] M. Aoi, M. Kamijo, and H. Yoshida, "Relationship between Facial Expression and Facial Electromyogram (f-EMG) Analysis in the Expression of Drowsiness," in Proc. of 2011 International Conference on Biometrics and Kansei Engineering (ICBAKE), Takamatsu, Japan, 2011, pp. 65–70.
- [87] I. Moon, M. Lee, J. Ryu, and M. Mun, "Intelligent robotic wheelchair with EMG-, gesture-, and voice-based interfaces," in Proc. of 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003), Las Vegas, Nevada, USA, 2003, pp. 3453–3458.
- [88] "Muscle Monday: Levator Scapula," Hummingbird Bodyworks, 01-Mar-2015. [Online]. Available: <http://hummingbirdbodyworks.com/levator-scapula-neck-and-shoulder-pain/>. [Accessed: 10-Apr-2016].
- [89] "Brainfingers hands-free computer controls," [Online]. Available: www.brainfingers.com/. [Accessed: 10-Oct-2016].
- [90] C. S. L. Tsui, P. Jia, J. Q. Gan, H. Hu, and K. Yuan, "EMG-based hands-free wheelchair control with EOG attention shift detection," in Proc. of IEEE International Conference on Robotics and Biomimetics (ROBIO 2007), Sanya, China, 2007, pp. 1266–1271.
- [91] Y. Zhang, X. Zhu, L. Dai, and Y. Luo, "Forehead sEMG signal based HMI for hands-free control," *The Journal of China Universities of Posts and Telecommunications*, Vol. 21, No. 3, pp. 98–105, 2014.
- [92] M. Hashimoto, K. Takahashi, and M. Shimada, "Wheelchair control using an EOG- and EMG-based gesture interface," in Proc. of IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM 2009), Singapore, 2009, pp. 1212–1217.
- [93] Z. Yi, D. Lingling, L. Yuan, and H. Huosheng, "Design of a surface EMG based human-machine interface for an intelligent wheelchair," in Proc. of 2011 10th International Conference on Electronic Measurement Instruments (ICEMI), Chengdu, China, 2011, pp. 132–136.
- [94] S. Ohishi and T. Kondo, "A proposal of EMG-based wheelchair for preventing disuse of lower motor function," in Proc. 2012 Annual International Conference on Instrumentation, Control, Information Technology and System Integration (SICE), Akita, Japan, 2012, pp. 236–239.
- [95] Y. Sasaki and T. Kondo, "A proposal of EMG-based teleoperation interface for distance mobility," in Proc. of 2011 IEEE International Conference on

- Systems, Man, and Cybernetics (SMC), Anchorage, Alaska, USA, 2011, pp. 2904–2909.
- [96] T. Kondo, O. Amagi, and T. Nozawa, “Proposal of anticipatory pattern recognition for EMG prosthetic hand control,” in Proc. of IEEE International Conference on Systems, Man and Cybernetics (SMC 2008), Singapore, 2008, pp. 897–902.
- [97] S. K. Choudhary, D. Chakraborty, N. M. Kakoty, and S. M. Hazarika, “Development of Cost Effective EMG Controlled Three Fingered Robotic Hand,” in Proc. of 2012 Third International Conference on Computer and Communication Technology (ICCCT), Allahabad, Uttar Pradesh, India, 2012, pp. 104–109.
- [98] Aapo Hyvärinen, “Survey on Independent Component Analysis,” *Neural Computing Surveys*, Vol. 2, pp. 94–128, 1999.
- [99] T.-P. Jung, S. Makeig, M. J. McKeown, A. J. Bell, T.-W. Lee, and T. J. Sejnowski, “Imaging brain dynamics using independent component analysis,” in Proc. of IEEE, Vol. 89, No. 7, pp. 1107–1122, 2001.
- [100] Y. Guangying, “Study of Myoelectric Prostheses Hand based on Independent Component Analysis and Fuzzy Controller,” in Proc. 8th International Conference on Electronic Measurement and Instruments (ICEMI '07), Xian, China, 2007, pp. 174–178.
- [101] C. De Luca, “Electromyography,” *Encyclopedia of Medical Devices and Instrumentation*, New Jersey, John Wiley & Sons, Inc., 2006.
- [102] L. A. Rivera and G. N. Desouza, “A power wheelchair controlled using hand gestures, a single sEMG sensor, and guided under-determined source signal separation,” in Proc. of 2012 4th IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2012), Rome, Italy, 2012, pp. 1535–1540.
- [103] V. Jurcak, D. Tsuzuki, and I. Dan, “10/20, 10/10, and 10/5 systems revisited: Their validity as relative head-surface-based positioning systems,” *NeuroImage*, Vol. 34, No. 4, pp. 1600–1611, 2007.
- [104] D. L. Sherman, A. M. Brambrink, D. Walterspacher, V. K. Dasika, R. Ichord, and N. V. Thakor, “Detecting EEG bursts after hypoxic-ischemic injury using energy operators,” in Proc. of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, Illinois, USA, 1997, pp. 1188–1190.
- [105] X. Kong, X. Lou, and N. V. Thakor, “Detection of EEG changes via a generalized Itakura distance,” in Proc. of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, Chicago, Illinois, USA, 1997, pp. 1540–1542.
- [106] A. Ademoglu and T. Demiralp, “Quadratic phase coupling of electroencephalogram (EEG) and evoked potentials (EP),” in Proc. of the 1992 International Biomedical Engineering Days, Istanbul, Turkey, 1992, pp. 146–150.
- [107] A. S. Gevins, “Analysis of the Electromagnetic Signals of the Human Brain: Milestones, Obstacles, and Goals,” *IEEE Transactions on Biomedical Engineering*, Vol. BME-31, No. 12, pp. 833–850, 1984.
- [108] D. Wang, T. Kochiyama, S. Lu, and J.-L. Wu, “Measurement and analysis of electroencephalogram (EEG) using directional visual stimuli for brain

- computer interface,” in Proc. of the 2005 International Conference on Active Media Technology (AMT 2005), Kagawa, Japan, 2005, pp. 34–39.
- [109] M. N. Teli and C. Anderson, “Nonlinear dimensionality reduction of electroencephalogram (EEG) for Brain Computer interfaces,” in Proc. of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC 2009), Minneapolis, Minnesota, USA, 2009, pp. 2486–2489.
- [110] K. Brigham and B. V. K. V. Kumar, “Subject identification from electroencephalogram (EEG) signals during imagined speech,” in Proc. of 2010 Fourth IEEE International Conference on Biometrics: Theory Applications and Systems (BTAS), Washington, D.C., USA, 2010, pp. 1–8.
- [111] O. Gunaydin and M. Ozkan, “Design of a Brain Computer Interface system based on electroencephalogram (EEG),” in Proc. of 2010 4th European DSP Education and Research Conference (EDERC), Nice, France, 2010, pp. 150–154.
- [112] G. H. Klem, H. O. Lüders, H. H. Jasper, and C. Elger, “The ten-twenty electrode system of the International Federation. The International Federation of Clinical Neurophysiology,” *Journal of Electroencephalography and clinical neurophysiology Supplement*, Vol. 52, pp. 3–6, 1999.
- [113] K. Tanaka, K. Matsunaga, and H. O. Wang, “Electroencephalogram-Based Control of an Electric Wheelchair,” *IEEE Transactions on Robotics*, Vol. 21, No. 4, pp. 762–766, 2005.
- [114] S. H. Patel and P. N. Azzam, “Characterization of N200 and P300: Selected Studies of the Event-Related Potential,” *International Journal of Medical Sciences*, Vol. 2, No. 4, pp. 147–154, 2005.
- [115] I. Iturrate, J. Antelis, and J. Minguez, “Synchronous EEG brain-actuated wheelchair with automated navigation,” in Proc. of IEEE International Conference on Robotics and Automation (ICRA '09), Kobe, Japan, 2009, pp. 2318–2325.
- [116] T. Jijun, Z. Peng, X. Ran, and D. Lei, “The portable P300 dialing system based on tablet and Emotiv EPOC headset,” in Proc. of 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Milan, Italy, 2015, pp. 566–569.
- [117] M. Ahmad and M. Aqil, “Implementation of nonlinear classifiers for adaptive autoregressive EEG features classification,” in Proc. of 2015 Symposium on Recent Advances in Electrical Engineering (RAEE), Islamabad, Pakistan, 2015, pp. 1–5.
- [118] “Wearables for your brain | EEG.” [Online]. Available: <https://emotiv.com/>. [Accessed: 10-Apr-2016].
- [119] D. Wijayasekara and M. Manic, “Human machine interaction via brain activity monitoring,” in Proc. of 2013 The 6th International Conference on Human System Interaction (HSI), Sopot, Poland, 2013, pp. 103–109.
- [120] S. Grude, M. Freeland, C. Yang, and H. Ma, “Controlling mobile Spykee robot using Emotiv Neuro headset,” in Proc. of 2013 32nd Chinese Control Conference (CCC), Xi'an, China, 2013, pp. 5927–5932.
- [121] W. Ouyang, K. Cashion, and V. K. Asari, “Electroencephalograph based brain machine interface for controlling a robotic arm,” in Proc. of 2013 IEEE

- Applied Imagery Pattern Recognition Workshop (AIPR): Sensing for Control and Augmentation, Washington, DC, USA, 2013, pp. 1–7.
- [122] M. K. Hazrati and U. G. Hofmann, “Avatar navigation in Second Life using brain signals,” in Proc. of 2013 IEEE 8th International Symposium on Intelligent Signal Processing (WISP), Madeira, Portugal, 2013, pp. 1–7.
- [123] A. Kawala-Janik, M. Podpora, M. Pelc, P. Piatek, and J. Baranowski, “Implementation of an inexpensive EEG headset for the pattern recognition purpose,” in Proc. of 2013 IEEE 7th International Conference on Intelligent Data Acquisition and Advanced Computing Systems Technology and Applications (IDAACS), Berlin, Germany, 2013, vol. 01, pp. 399–403.
- [124] H. Blaiech, M. Neji, A. Wali, and A. M. Alimi, “Emotion recognition by analysis of EEG signals,” in Proc. of 2013 13th International Conference on Hybrid Intelligent Systems (HIS), Gammarth, Tunisia, 2013, pp. 312–318.
- [125] D. A. Craig and H. T. Nguyen, “Adaptive EEG Thought Pattern Classifier for Advanced Wheelchair Control,” in Proc. of 29th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBS 2007), Lyon, France, 2007, pp. 2544–2547.
- [126] B. Rebsamen et al., “Controlling a wheelchair using a BCI with low information transfer rate,” in Proc. of IEEE 10th International Conference on Rehabilitation Robotics (ICORR 2007), Noordwijk, Netherlands, 2007, pp. 1003–1008.
- [127] M. Palankar et al., “Control of a 9-DoF Wheelchair-mounted robotic arm system using a P300 Brain Computer Interface: Initial experiments,” in Proc. of 2008 IEEE International Conference on Robotics and Biomimetics (ROBIO 2008), Bangkok, Thailand, 2009, pp. 348–353.
- [128] B. Rebsamen et al., “A Brain Controlled Wheelchair to Navigate in Familiar Environments,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 18, No. 6, pp. 590–598, 2010.
- [129] K. Kanewaran, K. Arshak, E. Burke, and J. Condron, “Towards a brain controlled assistive technology for powered mobility,” in Proc. of 2010 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), Buenos Aires, Argentina, 2010, pp. 4176–4180.
- [130] A. R. Satti, D. Coyle, and G. Prasad, “Self-paced brain-controlled wheelchair methodology with shared and automated assistive control,” in Proc. of 2011 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB), Paris, France, 2011, pp. 1–8.
- [131] S. M. T. Muller, T. F. Bastos-Filho, and M. Sarcinelli-Filho, “Using a SSVEP-BCI to command a robotic wheelchair,” in Proc. of 2011 IEEE International Symposium on Industrial Electronics (ISIE), Gdansk, Poland, 2011, pp. 957–962.
- [132] D. Huang, K. Qian, D.-Y. Fei, W. Jia, X. Chen, and O. Bai, “Electroencephalography (EEG)-Based Brain-Computer Interface (BCI): A 2-D Virtual Wheelchair Control Based on Event-Related Desynchronization/Synchronization and State Control,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, Vol. 20, No. 3, pp. 379–388, 2012.
- [133] F. Carrino, J. Dumoulin, E. Mugellini, O. A. Khaled, and R. Ingold, “A self-paced BCI system to control an electric wheelchair: Evaluation of a

- commercial, low-cost EEG device,” in Proc. of 2012 ISSNIP Biosignals and Biorobotics Conference: Biosignals and Robotics for Better and Safer Living (BRC), Manaus, Brazil, 2012, pp. 1–6.
- [134] J. Long, Y. Li, H. Wang, T. Yu, and J. Pan, “Control of a simulated wheelchair based on a hybrid brain computer interface,” in Proc. of 2012 Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC), San Diego, California, USA, 2012, pp. 6727–6730.
- [135] I. Pathirage, K. Khokar, E. Klay, R. Alqasemi, and R. Dubey, “A vision based P300 Brain Computer Interface for grasping using a wheelchair-mounted robotic arm,” in Proc. of 2013 IEEE/ASME International Conference on Advanced Intelligent Mechatronics (AIM), Wollongong, Australia, 2013, pp. 188–193.
- [136] D. J. Kupetz, S. A. Wentzell, and B. F. BuSha, “Head motion controlled power wheelchair,” in Proc. of 2010 IEEE 36th Annual Northeast Bioengineering Conference (NEBEC), New York, USA, 2010, pp. 1–2.
- [137] H. V. Christensen and J. C. Garcia, “Infrared Non-Contact Head Sensor, for Control of Wheelchair Movements,” Amsterdam, IOS Press, 2003.
- [138] J. M. Ford and S. J. Sheredos, “Ultrasonic head controller for powered wheelchairs,” *Journal of Rehabilitation Research and Development*, Vol. 32, No. 3, pp. 280–4, 1995.
- [139] F. Abedan Kondori, S. Yousefi, L. Liu, and H. Li, “Head operated electric wheelchair,” in Proc. of 2014 IEEE Southwest Symposium on Image Analysis and Interpretation (SSIAI), San Diego, California, USA, 2014, pp. 53–56.
- [140] R. Akmeliawati, F. S. B. Tis, and U. J. Wani, “Design and development of a hand-glove controlled wheel chair,” in Proc. of 2011 4th International Conference on Mechatronics (ICOM), Kuala Lumpur, Malaysia, 2011, pp. 1–5.
- [141] Z. Hu, L. Li, Y. Luo, Y. Zhang, and X. Wei, “A novel intelligent wheelchair control approach based on head gesture recognition,” in Proc. of 2010 International Conference on Computer Application and System Modeling (ICCASM 2010), Taiyuan, China, 2010, pp. 159–163.
- [142] M. Baklouti, M. Bruin, V. Guitteny, and E. Monacelli, “A Human-Machine Interface for assistive exoskeleton based on face analysis,” in Proc. of 2nd IEEE RAS EMBS International Conference on Biomedical Robotics and Biomechatronics (BioRob 2008), Scottsdale, Arizona, USA, 2008, pp. 913–918.
- [143] S. Rehman, B. Raytchev, I. Yoda, and L. Liu, “Vibrotactile rendering of head gestures for controlling electric wheelchair,” in Proc. of IEEE International Conference on Systems, Man and Cybernetics (SMC 2009), San Antonio, Texas, USA, 2009, pp. 413–417.
- [144] A. K. Arumbakkam, T. Yoshikawa, B. Dariush, and K. Fujimura, “A multi-modal architecture for human robot communication,” in Proc. of 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids 2010), Nashville, Tennessee, USA, 2010, pp. 639–646.
- [145] Y.-J. Tu, C.-C. Kao, and H.-Y. Lin, “Human computer interaction using face and gesture recognition,” in Proc. of Asia-Pacific Signal and Information

- Processing Association Annual Summit and Conference (APSIPA 2013), Kaohsiung, Taiwan, 2013, pp. 1–8.
- [146] J. Zhang, L. Zhuang, Y. Wang, Y. Zhou, Y. Meng, and G. Hua, “Video Demo: An Egocentric Vision Based Assistive Co-robot,” in Proc. of 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), Portland, Oregon, USA, 2013, pp. 48–49.
- [147] F. Abedan Kondori, S. Yousefi, L. Liu, and H. Li, “Head operated electric wheelchair,” in Proc. of 2014 IEEE Southwest Symposium on Image Analysis and Interpretation, San Diego, California, USA, 2014, pp. 53–56.
- [148] J. Zhan, “A pressure cushion for control of intelligent wheelchair movements,” in Proc. of IEEE 10th International Conference on Computer-Aided Industrial Design and Conceptual Design (CAID&CD 2009), Wenzhou, China, 2009, pp. 155–157.
- [149] S. Yokota, H. Hashimoto, Y. Ohyama, J. She, H. Kobayashi, and P. Blazevic, “The electric wheelchair controlled by human body motion,” in Proc. of 2nd Conference on Human System Interactions (HSI '09), Catania, Italy, 2009, pp. 247–250.
- [150] D. Chugo, K. Fujita, S. Yokota, Y. Sakaida, and K. Takase, “A depressurization assistance control based on the posture of a seated patient on a wheelchair,” in Proc. of 2011 IEEE International Conference on Rehabilitation Robotics (ICORR 2011), Zurich, Switzerland, 2011, pp. 1–6.
- [151] S. Yokota, H. Hashimoto, Y. Ohyama, D. Chugo, J. She, and H. Kobayahsi, “Improvement of body motion measurement using pressure data on Human Body Motion Interface,” in Proc. of 2011 4th International Conference on Human System Interactions (HIS 2011), Yokohama, Japan, 2011, pp. 144–149.
- [152] A. D. Young, M. J. Ling, and D. K. Arvind, “Orient-2: A Realtime Wireless Posture Tracking System Using Local Orientation Estimation,” in Proc. of 4th Workshop on Embedded Networked Sensors, New York, USA, 2007, pp. 53–57.
- [153] E. R. Bachmann, X. Yun, D. McKinney, R. B. McGhee, and M. J. Zyda, “Design and implementation of MARG sensors for 3-DOF orientation measurement of rigid bodies,” in Proc. of IEEE International Conference on Robotics and Automation (ICRA '03), Taipei, Taiwan, 2003, pp. 1171–1178.
- [154] A. Lynch, B. Majeed, B. O’Flynn, J. Barton, F. Murphy, K. Delaney and S. C. O’Mathuna, “A wireless inertial measurement system (WIMS) for an interactive dance environment,” *Journal of Physics: Conference Series* 15, Vol. 15, No. 1, pp. 95–100, 2005.
- [155] “MTx - Products,” Xsens 3D motion tracking. [Online]. Available: <https://www.xsens.com/products/mtx/>. [Accessed: 26-Apr-2016].
- [156] E. Vergaro, M. Casadio, V. Squeri, P. Giannoni, P. Morasso, and V. Sanguineti, “Self-adaptive robot training of stroke survivors for continuous tracking movements,” *Journal of Neuro Engineering and Rehabilitation*, Vol. 7, No. 1, pp. 1–12, 2010.
- [157] D. Bannach, O. Amft, K. S. Kunze, E. A. Heinz, G. Troster, and P. Lukowicz, “Waving Real Hand Gestures Recorded by Wearable Motion Sensors to a Virtual Car and Driver in a Mixed-Reality Parking Game,” in Proc. of IEEE

- Symposium on Computational Intelligence and Games (CIG 2007), Honolulu, Hawaii, USA, 2007, pp. 32–39.
- [158] H. Harms, O. Amft, D. Roggen and G. Tröster, “Rapid prototyping of smart garments for activity-aware applications,” *Journal of Ambient Intelligence and Smart Environments*, Vol. 1, No. 2, pp. 87–101, 2009.
- [159] R. E. Mayagoitia, A. V. Nene, and P. H. Veltink, “Accelerometer and rate gyroscope measurement of kinematics: an inexpensive alternative to optical motion analysis systems,” *Journal of Biomechanics*, Vol. 35, No. 4, pp. 537–542, 2002.
- [160] B. B. Graham, “Using an accelerometer sensor to measure human hand motion,” Thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, Cambridge, 2000.
- [161] DC Hovde, MD Prouty, I Hrvoic, RE Slocum, “Commercial magnetometers and their application,” *Optical Magnetometry - Chapter 20*, Cambridge, Cambridge University Press, 2013.
- [162] P. Jain, “Magnetometers,” [Online]. Available: <https://www.engineersgarage.com/articles/magnetometer>. [Accessed: 29-May-2017].
- [163] T. Agarwal, “What are Magnetometers - Overview about Its Types & Applications,” [Online]. Available: <https://www.elprocus.com/magnetometers-types-applications/> [Accessed: 29-May-2017].
- [164] A. M. Shkel, C. Acar, and C. Painter, “Two types of micromachined vibratory gyroscopes,” in *Proc. of the 4th IEEE conference on Sensors (IEEE Sensors 2005)*, Irvine, California, USA, 2005, pp. 531–536.
- [165] M. Wen, Z. Luo, W. Wang, and S. Liu, “A characterization of the performance of MEMS vibratory gyroscope in different fields,” in *Proc. of 2014 15th International Conference on Electronic Packaging Technology (ICEPT)*, Chengdu, China, 2014, pp. 1547–1551.
- [166] G. Xue, T. Li, and H. Zhang, “Research status and development of magnetically suspended rotor gyroscopes,” in *Proc. of International Conference on Applied Superconductivity and Electromagnetic Devices (ASEMD 2009)*, Chengdu, China, 2009, pp. 373–376.
- [167] E. Jovanov et al., “Avatar; A multi-sensory system for real time body position monitoring,” in *2009 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Minneapolis, Minnesota, USA, 2009, pp. 2462–2465.
- [168] E. Jovanov, V. Milutinovic, and A. R. Hurson, “Acceleration of nonnumeric operations using hardware support for the Ordered Table Hashing algorithms,” *IEEE Transactions on Computers*, Vol. 51, No. 9, pp. 1026–1040, 2002.
- [169] P. Madhushri, A. A. Dzhagaryan, E. Jovanov, and A. Milenkovic, “A Smartphone Application Suite for Assessing Mobility,” in *Proc. of the 2016 38th Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, Orlando, Florida, USA, 2016, pp. 3117–3120.
- [170] E. Jovanov, D. Raskovic, A. O. Lords, P. Cox, R. Adhami, and F. Andrasik, “Synchronized physiological monitoring using a distributed wireless intelligent sensor system,” in *Proceedings of the 25th Annual International*

- Conference of the IEEE Engineering in Medicine and Biology Society, Cancun, Mexico, 2003, pp. 1368–1371.
- [171] H. Garner, M. Klement, and K.-M. Lee, “Design and analysis of an absolute non-contact orientation sensor for wrist motion control,” in Proc. of 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Como, Italy, 2001, pp. 69–74.
- [172] Z. Aiyun, Y. Kui, Y. Zhigang, and Z. Haibing, “Research and application of a robot orientation sensor,” in Proc. of 2003 IEEE International Conference on Robotics, Intelligent Systems and Signal Processing, Changsha, China, 2003, pp. 1069–1074.
- [173] H. Harms, O. Amft, R. Winkler, J. Schumm, M. Kusserow, and G. Troester, “ETHOS: Miniature orientation sensor for wearable human motion analysis,” in Proc. of 2010 IEEE Sensors Conference (Sensors 2010), Kona, Hawaii, USA, 2010, pp. 1037–1042.
- [174] J. Chahl and A. Mizutani, “Biomimetic Attitude and Orientation Sensors,” IEEE Sensors Journal, Vol. 12, No. 2, pp. 289–297, 2012.
- [175] G. Xianfeng, Y. Jing, and W. Wei, “Design of high accuracy orientation sensor of MWD,” in Proc. of 2011 International Conference on Electric Information and Control Engineering (ICEICE), Wuhan, China, 2011, pp. 297–300.
- [176] T. Beravs, J. Podobnik, and M. Munih, “Three-Axial Accelerometer Calibration Using Kalman Filter Covariance Matrix for Online Estimation of Optimal Sensor Orientation,” IEEE Transactions on Instrumentation and Measurement, Vol. 61, No. 9, pp. 2501–2511, 2012.
- [177] B. Florentino-Liano, N. O’Mahony, and A. Artes-Rodriguez, “Human activity recognition using inertial sensors with invariance to sensor orientation,” in Proc. of 2012 3rd International Workshop on Cognitive Information Processing (CIP), Baiona, Spain, 2012, pp. 1–6.
- [178] S. V. S. Manogna, “Head Movement Based Assist System for Physically Challenged,” in Proc. of 2010 4th International Conference on Bioinformatics and Biomedical Engineering, Chengdu, China, 2010, pp. 1–4.
- [179] T. Lu, “A motion control method of intelligent wheelchair based on hand gesture recognition,” in Proc. of 2013 8th IEEE Conference on Industrial Electronics and Applications (ICIEA), Melbourne, Australia, 2013, pp. 957–962.
- [180] D. K. Rathore, P. Srivastava, S. Pandey, and S. Jaiswal, “A novel multipurpose smart wheelchair,” in Proc. of 2014 IEEE Students’ Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2014, pp. 1–4.
- [181] A. Milenković, M. Milosevic, and E. Jovanov, “Smartphones for smart wheelchairs,” in Proc. of the 2013 IEEE International Conference on Body Sensor Networks, Cambridge, Massachusetts, USA, 2013, pp. 1–6.
- [182] M. Milenkovic, E. Jovanov, J. Chapman, D. Raskovic, and J. Price, “An accelerometer-based physical rehabilitation system,” in Proc. of the Thirty-Fourth Southeastern Symposium on System Theory, Huntsville, Alabama, USA, 2002, pp. 57–60.

- [183] M. F. Ruzaij, S. Neubert, N. Stoll, and K. Thurow, "Design and testing of low cost three-modes of operation voice controller for wheelchairs and rehabilitation robotics," in Proc. of the IEEE 9th International Symposium on Intelligent Signal Processing (WISP), Siena, Italy, 2015, pp. 1–6.
- [184] M. F. Ruzaij, S. Neubert, N. Stoll, and K. Thurow, "Hybrid Voice Controller for Intelligent Wheelchair and Rehabilitation Robot Using Voice Recognition and Embedded Technologies," Journal of Advanced Computational Intelligence and Intelligent Informatics, Vol. 20, No. 4, pp. 615–622, 2016.
- [185] "Microphone Impedance." [Online]. Available: <http://www.mediacollege.com/audio/microphones/impedance.html>. [Accessed: 23-Mar-2017].
- [186] Audio-Technica, "What Is a Microphone's Sensitivity and Why Is It Important to Know?" [Online]. Available: <http://blog.audio-technica.com/week-microphones-sensitivity-important-know>. [Accessed: 16-Jun-2016].
- [187] Wikipedia, "Universal asynchronous receiver/transmitter," [Online]. Available: https://en.wikipedia.org/wiki/Universal_asynchronous_receiver/transmitter. [Accessed: 09-Mar-2017].
- [188] Total Phase Blog, "SPI vs. UART: Similarities and Differences," Total Phase Blog. [Online]. Available: <https://www.totalphase.com/blog/2016/06/spi-vs-uart-similarities-differences/>. [Accessed: 24-Mar-2017].
- [189] Wikipedia, "I²C," [Online]. Available: <https://en.wikipedia.org/wiki/I%C2%B2C>. [Accessed: 19-Mar-2017].
- [190] "Lenovo G50 Technical Specification | Lenovo India." [Online]. Available: <http://shopap.lenovo.com/in/en/tech-specs/laptops/lenovo-g50/>. [Accessed: 26-Mar-2017].
- [191] "Samsung Galaxy Tab 4 10.1 - Full tablet specifications." [Online]. Available: http://www.gsmarena.com/samsung_galaxy_tab_4_10_1-6247.php. [Accessed: 26-Mar-2017].
- [192] "Apple iPhone 4s - Full phone specifications." [Online]. Available: http://www.gsmarena.com/apple_iphone_4s-4212.php. [Accessed: 26-Mar-2017].
- [193] "EFM32™ the world's most energy-friendly microcontrollers PRODUCT SELECTOR GUIDE." [Online]. Available: <http://www.silabs.com/products/mcu/32-bit/Pages/32-bit-microcontrollers.aspx>. [Accessed: 05 September 2016].
- [194] EFM32GG990 DATASHEET, [Online]., Available: <http://www.silabs.com>. [Accessed: 20 November 2015].
- [195] "Orientation | Define Orientation at Dictionary.com." [Online]. Available: <http://www.dictionary.com/browse/orientation>. [Accessed: 28-Oct-2016].
- [196] B. K. Horn, "Relative orientation," International Journal of Computer Vision Vol. 4, No. 1, pp. 59–78, 1990.
- [197] "Understanding Euler Angles | CH Robotics." [Online]. Available: <http://www.chrobotics.com/library/understanding-euler-angles>. [Accessed: 10-Oct-2016].

-
- [198] “Euler angles,” Wikipedia. [Online]. Available: https://en.wikipedia.org/wiki/Euler_angles/. [Accessed: 10-Oct-2016].
- [199] G. G. Slabaugh, “Computing Euler angles from a rotation matrix,” Retrieved August Journal, Vol. 6, No. 2000, pp. 39–63, 1999.
- [200] BNO055 data sheet, [online]. Available: http://www.bosch-sensortec.com/en/homepage/products_3/sensor_hubs/iot_solutions/bno055_1/bno055_4. [Accessed: 15 Jun 2015]. .
- [201] Dr. Jaguar Light robot user guide. [Online]. Available: http://jaguar.drrobot.com/images/Jaguar_lite_manual.pdf. [Accessed: 20 December 2015].
- [202] PMS5005 Protocol Reference Manual. [Online]. Available: http://http://www.drrobot.com/products/item_downloads/sentinel_4.pdf. [Accessed: 02-Mar-2016].
- [203] M. D. Natale, H. Zeng, P. Giusto, and A. Ghosal, “Understanding and Using the Controller Area Network Communication Protocol: Theory and Practice. “New York, Springer Science & Business Media, 2012.
- [204] “AMT - Auto Move Technologies > Home > Products > Geared Motors > SRG 04.” [Online]. Available: http://www.amt-schmid.com/en/products/right_angled_drives/srg_05/srg_05.php. [Accessed: 10-Oct-2016].
- [205] “Sabertooth 2x32 regenerative dual motor driver.” [Online]. Available: <https://www.dimensionengineering.com/products/sabertooth2x32>. [Accessed: 23-Oct-2016].
- [206] R. H. Rockland and S. Reisman, “Voice activated wheelchair controller,” in Proc. of the IEEE 24th Annual Northeast Bioengineering Conference, Hershey, Pennsylvania, USA, 1998, pp. 128–129.
- [207] M. F. Ruzaij and S. Poonguzhali, “Design and implementation of low cost intelligent wheelchair,” in Proc. of 2012 International Conference on Recent Trends in Information Technology (ICRTIT), Chennai, India, 2012, pp. 468–471.
- [208] C. Aruna, A. Dhivya Parameswari, M. Malini, and G. Gopu, “Voice recognition and touch screen control based wheel chair for paraplegic persons,” in Proc. of 2014 International Conference on Green Computing Communication and Electrical Engineering (ICGCCEE), Coimbatore, India, 2014, pp. 1–5.
- [209] M. Sajkowski, “Voice control of dual-drive mobile robots-survey of algorithms,” in Proc. of Third International Workshop on Robot Motion and Control (RoMoCo '02, 2002), Bukowy Dworek, Poland, 2002, pp. 387–392.
- [210] P. Cosi, J.-P. Hosom, J. Shalkwyk, S. Sutton, and R. A. Cole, “Connected digit recognition experiments with the OGI Toolkit’s neural network and HMM-based recognizers,” in Proc. of 1998 IEEE 4th Workshop on Interactive Voice Technology for Telecommunications Applications (IVTTA '98), Torino, Italy, 1998, pp. 135–140.
- [211] M. F. Ruzaij, S. Neubert, N. Stoll, and K. Thurow, “Auto calibrated head orientation controller for robotic-wheelchair using MEMS sensors and embedded technologies,” in 2016 IEEE Sensors Applications Symposium (SAS), Catania, Italy, 2016, pp. 433–438.

- [212] M. F. Ruzaij, S. Neubert, N. Stoll, and K. Thurow, “Multi-sensor robotic-wheelchair controller for handicap and quadriplegia patients using embedded technologies,” in 2016 9th International Conference on Human System Interactions (HSI), London, UK, 2016, pp. 103–109.
- [213] A. Gawade, R. Patil, S. Hanamshet, and P. Chougule, “Fuel Monitoring and Headlight Intensity Controller using Embedded Control Network,”. Multidisciplinary Journal of Research in Engineering and Technology, pp. 22-34, 2015.
- [214] Google Documents, “HC-SR04 User’s_Manual,” [Online]. Available: https://docs.google.com/document/d/1Y-yZnNhMYy7rwhAgyL_pfa39RsB-x2qR4vP8saG73rE/edit. [Accessed: 21-Apr-2017].

DECLARATION

This dissertation 'Hybrid Wheelchair Controller for Handicapped and Quadriplegic Patients' is a presentation of my original research work. Wherever contributions of others are involved, every effort is made to indicate this clearly, with due reference to the literature, and acknowledgement of collaborative research and discussions. The work of this dissertation has been done by myself at the University of Rostock, Germany. Also, the dissertation has not been accepted for any degree and is not concurrently submitted in candidature of any other degree.

Rostock, 28 June, 2017

Mohammed Faeik Ruzaij Al-Okby

List of Publications

1. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Hybrid Voice Controller for Intelligent Wheelchair and Rehabilitation Robot Using Voice Recognition and Embedded Technologies”, *Journal of Advanced Computational Intelligence and Intelligent Informatics*, Vol. 20 No. 4-2016, Fuji Technology Press Ltd, Tokyo, Japan, pp. 615-622, 2016. (Scopus, Web of Science)
2. **Mohammed Faeik Ruzaij Al-Okby**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Low-Cost Hybrid Wheelchair Controller for Quadriplegias and Paralysis Patient”, *Advances in Science, Technology and Engineering Systems Journal (ASTESJ)*, Vol. 2, No. 3, pp. 687-694, 2017.
3. **Mohammed Faeik Ruzaij Al-Okby**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Complementary functions for intelligent wheelchair head tilts controller”, in 2017 IEEE 15th International Symposium on Intelligent Systems and Informatics (SISY), Subotica, Serbia, 14-16 September 2017, pp. 000117 - 000122. (IEEE, Scopus)
4. **Mohammed Faeik Ruzaij Al-Okby**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Development and Testing of Intelligent Low-Cost Wheelchair Controller for Quadriplegics and Paralysis Patients”, in 2nd International Conference on Bio-engineering for Smart Technologies Biosmart2017, Paris, France, 30 August-1 September, 2017, pp. 1-4. (IEEE, Scopus)
5. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Design and implementation of low-cost intelligent wheelchair controller for quadriplegias and paralysis patient”, in Proc. 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMi), Herl'any, Slovakia, 26-28 January 2017, pp. 399-404. (IEEE, Scopus)
6. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “A speed compensation algorithm for a head tilts controller used for wheelchairs and rehabilitation applications”, in Proc. 2017 IEEE 15th International Symposium on Applied Machine Intelligence and Informatics (SAMi), Herl'any, Slovakia, 26-28 January 2017, pp. 497-502. (IEEE, Scopus)
7. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Multi-Sensor Robotic-Wheelchair Controller for Handicap and Quadriplegia Patients Using Embedded Technologies”, in Proc. 2016-9th International Conference on Human System Interactions (HSI), Portsmouth, United Kingdom, 6-8 July 2016. pp.103-109. (IEEE, Scopus)
8. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Auto Calibrated Head Orientation Controller for Robotic-Wheelchair Using MEMS Sensors and Embedded Technologies”, in Proc. 2016 IEEE Sensors Applications Symposium (SAS 2016), Catania, Italy, 20-22 April 2016, pp. 433-438. (IEEE, Scopus)
9. **Mohammed Faeik Ruzaij**, Sebastian Neubert, Norbert Stoll, Kerstin Thurow, “Design and Testing of Low Cost Three-Modes of Operation Voice Controller for Wheelchairs and Rehabilitation Robotics”, in Proc. 9th IEEE International Symposium on Intelligent Signal Processing WISP2015, Siena, Italy, May 15 - 17, 2015, pp. 114-119. (IEEE, Scopus)

Dissertation Theses

- 1- Quadriplegia and paralyzed patients are the primary targets of this work. They need to use an electrical wheelchair or rehabilitation robots in their daily movement.
- 2- Quadriplegia or tetraplegia happens when a person has a spinal cord injury where paralysis usually affects the cervical spinal nerves resulting in the paralysis of all four limbs.
- 3- Limited effective and useful control signal from quadriplegia patient are available in the head, neck, and shoulder regions.
- 4- User's voice is one of the effective solutions for quadriplegic patients to control a wheelchair or rehabilitation applications.
- 5- Embedded voice recognition modules with a microcontroller can be covered the required number of voice commands need to be used for control purpose and also provide a low-cost and a small size solution.
- 6- Speaker dependent mode of voice recognition is the best choice for wheelchair and rehabilitation application to let the system response to the user commands only. Speaker independent mode can be used for general purpose control application such as lab automation based on the application requirements.
- 7- Speaker dependent mode can be used for any language compare with specific language for speaker independent.
- 8- Using two or more voice recognition modules improve the voice controller performance and make the system able to detect and avoid false recognition and false positive errors.
- 9- Head tilts can be used as a control solution for quadriplegic patients to control the wheelchairs using head orientation.
- 10- Measuring the head orientation and wheelchair reference orientation add an important feature to the head tilts controller to avoid the change in the control commands thresholds in the case of ascending or descending ramps.
- 11- A command confirmation function has embedded in the head tilts controller and used to avoid the system response for involuntary head motion.
- 12- Orientation sensor error handling function used to inform the used if any one

or both of the orientation modules stop updating orientation data due to an instantaneous loss of power in the orientation module caused by strong vibration when the system passes bumpy roads in outdoor environments or for any reason.

- 13- A wrong orientation handling function has been programmed to stop the system and protect the user if the head orientation goes out of the programmed ranges of the control commands tilts angles.
- 14- The Integration of voice and head tilts controller in one hybrid controller produces a simple, fixable and comfortable controller that allows quadriplegics and disabled users to select and change the voice and head tilts controller easily.

Abstract

Nowadays a significant increase in the number of older persons can be seen in all countries of the world. This part of society as well as patients with quadriplegia, permanently paralyzed, and/or amputated arms need to use special control systems to control the rehabilitation robots and wheelchairs instead of using traditional electrical wheelchairs, which are controlled by a joystick. The system design and development must take into consideration available control signals for this kind of patients. The most important control signals can be received from the region of shoulder, neck, and head of the user. Some of the useful signals in this area are the voice, head orientation, Electroencephalogram (EEG), Electromyogram (EMG) and Electrooculogram (EOG).

In this dissertation, a multi-input control system for rehabilitation application is proposed. The multi-input makes the system more flexible to adapt to the available body signals, which are used as control signals for the rehabilitation and medical applications. The first input module is a voice controller unit. It includes voice controllers with two modes of operation to maximize the voice recognition accuracy and reduce the voice recognition errors. The voice controller has a novel implementation using two different voice recognition modules which add an intelligent feature to the system that allows it to identify wrong commands recognition and also increases the voice recognition probability. The second input module is the head orientation control unit. It uses the user head movements around the x, y, and z-axes to control the wheelchair movement and speed in all directions. This control unit includes two 9 degree of freedom orientation modules. Each module has three MEMS sensors (accelerometer, gyroscope, and magnetometer) which are combined in one module to give proper orientation data in the format of Euler angles. An auto-calibrated algorithm has been embedded into the head tilts controller to calibrate the control of movement and speed when the wheelchair passes through a hill or non-straight roads. Other functions such as command confirmation function, wrong orientation handling function, and orientation sensor error handling functions have been added to the head tilts controller to enhance the performance of the controller and to add more safety to the system users. The low-cost design is taken into consideration as it allows more patients to use this system.

Zusammenfassung

Gegenwärtig ist in allen Ländern weltweit eine signifikante Zunahme der Anzahl älterer Menschen zu verzeichnen. Dieser Teil der Gesellschaft sowie Patienten mit Tetraplegie, die permanent gelähmt sind und / oder amputierte Arme haben, müssen, alternativ zu herkömmlichen elektrischen Rollstühlen mit Joysticksteuerung, auf Rehabilitationsroboter und Rollstühle mit speziellen Kontrollsystemen zurückgreifen. Das Systemdesign und die Entwicklung berücksichtigt verfügbare Steuersignale für diese Patienten. Die wichtigsten Steuersignale vom Benutzer können dabei aus dem Bereich der Schulter, des Nackens und des Kopfes erfasst werden. Die nützlichsten Signale in diesem Bereich sind die Stimme, die Kopforientierung, das Elektroenzephalogramm (EEG), das Elektromyogramm (EMG) und das Elektrokulogramm (EOG).

In dieser Dissertation wird ein Multi-Input-Hybridsteuerungssystem für Rehabilitationsanwendungen vorgeschlagen. Der Multi-Input macht das System flexibler für die Anpassung an die verfügbaren Bio-Signale, die als Steuersignale für Rehabilitations- und medizinische Anwendungen verwendet werden. Das erste Eingangsmodul ist eine Sprachsteuerungseinheit. Sie umfasst einen Sprachkontroller mit zwei Betriebsmodi, um die Genauigkeit der Spracherkennung zu maximieren und die Spracherkennungsfehler zu reduzieren. Der Sprachkontroller hat eine neuartige Implementierung, die zwei unterschiedliche Spracherkennungsmodulare einsetzt, die das System um ein intelligentes Merkmal erweitern, das es ermöglicht, falsche Befehlsenerkennungen zu identifizieren und auch die Spracherkennungswahrscheinlichkeit zu erhöhen. Das zweite Eingangsmodul ist die Kopforientierungssteuereinheit. Aus den Bewegungen des Benutzerkopfs um die x-, y- und z-Achse wird die Bewegung und die Geschwindigkeit des Rollstuhls in alle Richtungen gesteuert. Diese Steuereinheit enthält zwei Orientierungsmodulare mit 9 Freiheitsgraden. Jedes Modul hat drei MEMS-Sensoren (Beschleunigungsmesser, Gyroskop und Magnetometer), die in einem Modul kombiniert sind, um korrekte Orientierungsdaten im Format der Euler-Winkel zu erhalten. Ein automatisch kalibrierter Algorithmus wurde in den Kopforientierungskontroller eingebaut, um die Bewegungssteuerung und die Geschwindigkeit an Hügeln oder ungeraden Straßen anzupassen. Weitere Funktionen, wie die Kommandobestätigungsfunktion, die Funktion zur Handhabung falscher Orientierungen und die Funktion zur Behandlung von Fehlern durch den

Orientierungssensor, wurden in der Kopforientierungssteuerung ergänzt, um die Leistung des Controllers zu verbessern und den Systembenutzern mehr Sicherheit zu geben. Das kostengünstige Design unterstützt eine hohe Verbreitung des Systems.