

## University of Southampton Research Repository ePrints Soton

Copyright © and Moral Rights for this thesis are retained by the author and/or other copyright owners. A copy can be downloaded for personal non-commercial research or study, without prior permission or charge. This thesis cannot be reproduced or quoted extensively from without first obtaining permission in writing from the copyright holder/s. The content must not be changed in any way or sold commercially in any format or medium without the formal permission of the copyright holders.

When referring to this work, full bibliographic details including the author, title, awarding institution and date of the thesis must be given e.g.

AUTHOR (year of submission) "Full thesis title", University of Southampton, name of the University School or Department, PhD Thesis, pagination

**University of Southampton**

**Faculty of Physical Sciences and Engineering**

Electronics and Computer Science

Recognition of Elementary Upper Limb  
Movements in Nomadic Environment

by

Dwaipayan Biswas

Thesis for the degree of Doctor of Philosophy

April 2015



# Author's Declaration

I, Dwaipayan Biswas, declare that the thesis entitled *Recognition of Elementary Upper Limb Movements in Nomadic Environment*, and the work presented in the thesis are both my own, and have been generated by me as the result of my own original research. I confirm that:

- this work was done wholly or mainly while in candidature for a research degree at this University,
- where any part of this thesis has previously been submitted for a degree or any other qualification at this University or any other institution, this has been clearly stated;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have consulted the published work of others, this is always clearly attributed;
- where I have quoted from the work of others, the source is always mentioned. With the exception of such quotations, this thesis is entirely my own work;
- I have acknowledged all main sources of help;
- where the thesis is based on work done by myself jointly with others, I have made clear exactly what was done by others and what I have contributed myself;
- part of this work have been published as listed in section 1.8.



# University of Southampton

## ABSTRACT

Faculty of Physical Sciences and Engineering  
Electronics and Computer Science

### Doctor of Philosophy

Recognition of Elementary Upper Limb Movements in Nomadic Environment

by Dwaipayan Biswas

ICT enabled body-worn remote rehabilitation system has been projected as an effective means for combating the major socio-economic challenge resulting from the need for quality care delivery for stroke survivors. The two major problems faced in such systems are: 1) while effective for characterising the patient's performance during a constrained "exercise phase" in remote settings, the more natural indicator of rehabilitation status, i.e., the patient's performance in an "unconstrained nomadic environment", are often not considered and; 2) being body-worn and thus constrained by the battery life, their sustainability for long-term continuous monitoring is questionable. These shortcomings motivated the: 1) exploration of effective algorithmic strategies for accurately detecting movement of affected body parts, more specifically, the movement of the upper limb since it frequently gets affected by stroke episodes – in unconstrained scenarios and; 2) translation of the algorithms to dedicated low-power hardware with an aim of enhancing the battery life of a resource constrained body-worn sensor based remote rehabilitation system for its sustained operation satisfying the notion of long-term continuous monitoring.

Following instructions of expert physiotherapists, this work concentrates on detecting three fundamental upper limb movements in unconstrained scenarios: extension/flexion of the forearm; rotation of the forearm about the elbow; and rotation of the arm about the long axis of forearm, using body-worn inertial sensors. After selecting the appropriate type of inertial sensors and their positions through exhaustive experiments, two novel algorithms were proposed to recognize the above mentioned movements: 1) clustering and minimum distance classifier based approach and 2) tracking the orientation of an inertial sensor placed on the wrist. The performances of the algorithms have been evaluated prospectively through an archetypal activity '*making-a-cup-of-tea*' which includes multiple occurrences of the chosen movements. The proposed clustering based approach detected the three movements with an average accuracy of 88% and 70% using accelerometer data and 83% and 70% using gyroscope data obtained from the wrist for healthy subjects and stroke survivors respectively. Compared to that the proposed sensor orientation based methodology using a wrist-worn accelerometer only recognized the three movements with accuracies in the range of 91-99% for healthy subjects and 70%-85% for stroke survivors. However the clustering based approach provides greater flexibility in terms of incorporating new types of movements apart from the ones chosen here and can also be used to track changes in motor functionality over time. Subsequently it was translated into a novel ASIC resulting in dynamic power consumption of 25.9 mW @20 MHz in 130 nm technology. On the other hand, the sensor orientation based approach was also validated in hardware using an Altera DEII FPGA system, for high speed real-time movement recognition.



# Table of Contents

1. Chapter 1	<b>Introduction</b>	1
1.1	Home-based rehabilitation for stroke survivors	2
1.2	Shortcomings of the existing telemedicine modalities	4
1.3	Motivation	5
1.4	Research challenges	6
1.5	Research contributions	6
1.6	Approach	8
1.6.1	Data acquisition	8
1.6.2	Movement recognition algorithms	9
1.6.3	Hardware design	10
1.6.4	Research Constraints	11
1.7	Organisation	12
1.8	Publications	13
2. Chapter 2	<b>Background and Literature review</b>	17
2.1	Introduction	17
2.2	Activity monitoring	20
2.2.1	Rehabilitation exercise platforms	21
2.2.2	Virtual reality based systems - virtual directional exercise platform	24
2.2.3	Commercial gaming devices for tele-rehabilitation	26
2.2.4	Microsoft Kinect for rehabilitation	28
2.2.5	Monitoring activities of daily living (ADL)	29
2.3	Discussion	49
3. Chapter 3	<b>Movement Selection and Sensor Calibration</b>	51
3.1	Introduction	51
3.2	Movement selection	51
3.3	Inertial sensors and their calibration	54
3.3.1	Accelerometer calibration	55
3.3.2	Gyroscope calibration	60
3.4	Estimation of error margin and drift	63
3.5	Discussion	66
4. Chapter 4	<b>Sensor Orientation based Movement Recognition</b>	67
4.1	Introduction	67
4.2	Experimental setup	68
4.3	Algorithm design	70
4.3.1	Pre-processing	74
4.3.2	Orientation detection	74
4.3.3	Sequence detection	75
4.3.4	Action recognition	76
4.4	Results and analysis	78
4.4.1	Healthy subjects	78

4.4.2	Stroke survivors	78
4.5	Summary of explorations	79
4.6	Hardware design	79
4.7	Algorithm to architecture mapping	80
4.7.1	Orientation detection (OD)	81
4.7.2	Sequence detection (SD)	82
4.7.3	Action detection (AD)	84
4.8	Evaluation	84
4.9	Sensor interface for real-time movement recognition	86
4.10	Discussion	87
5.	<b>Chapter 5 Movement Recognition using Supervised Learning Algorithms</b>	89
5.1	Introduction	89
5.2	Approach	90
5.2.1	Data acquisition	90
5.2.2	Data mining	93
5.2.3	Feature extraction	94
5.2.4	Feature selection	95
5.3	Classification using supervised learning techniques	96
5.3.1	Linear/Quadratic Discriminant Analysis	97
5.3.2	Support Vector Machines	100
5.3.3	Classifier performance evaluation	102
5.4	Cross-validation on the training dataset	103
5.4.1	Generalized approach	103
5.4.2	Results for the Generalized approach	104
5.4.3	Personalized approach	106
5.4.4	Results for the Personalized approach	107
5.4.5	Conclusions from the cross-validation step	109
5.5	Evaluation of the trained model - prospective study	110
5.5.1	Healthy subjects	110
5.5.2	Stroke survivors	112
5.6	Discussion	114
6.	<b>Chapter 6 Recognition of Arm Movements using <i>k</i>-means Clustering Classification</b>	119
6.1	Introduction	119
6.2	Approach	120
6.2.1	Data acquisition and pre-processing	122
6.2.2	Feature extraction	124
6.2.3	Feature selection	124
6.2.4	Cluster formation on the training dataset	125
6.2.5	Cross-validation of the training dataset	126
6.2.6	Prospective evaluation with the testing dataset	129
6.3	Results and analysis – prospective evaluation	129
6.3.1	Healthy subjects	130
6.3.2	Stroke survivors	133

6.3.3	Evaluation with LDA and SVM	134
6.4	Summary of exploration	136
6.5	Application framework	137
6.6	Discussion	138
7.	<b>Chapter 7 ASIC Design for Minimum Distance Classifier</b>	141
7.1	Introduction	141
7.2	Approach	142
7.3	Feature extraction	144
7.4	CORDIC fundamentals	145
7.5	CORDIC formulation of features	147
7.5.1	Mean ( $\mu$ )	148
7.5.2	Root mean square (rms)	148
7.5.3	Standard deviation ( $\sigma$ )	149
7.5.4	Index of dispersion (D)	149
7.5.5	Kurtosis	150
7.5.6	Skewness	150
7.5.7	Absolute difference (abs. diff)	151
7.5.8	Information entropy (inf. entropy)	151
7.5.9	Jerk metric	153
7.5.10	Peak number (peaks)	154
7.5.11	Maximum peak amplitude (max_mag)	154
7.6	Architecture and evaluation	154
7.6.1	Architecture	155
7.6.2	Hardware complexity analysis of proposed architecture	159
7.6.3	Hardware complexity analysis of non-CORDIC architecture	161
7.6.4	Synthesis and verification	164
7.7	Minimum distance computation	165
7.8	Synthesis and verification	171
7.9	Chip design	173
7.10	Discussion	175
8.	<b>Chapter 8 Conclusion and Future Works</b>	177
8.1	Future works	181
	Appendix A	187
	References	189



# List of Figures

<b>1.1:</b> Overview of home-based monitoring for stroke survivors.	8
<b>2.1:</b> Overview of a wireless body area network system for remote healthcare monitoring, comprising of on-body sensors for monitoring physiological parameters (ECG, SpO <sub>2</sub> ) and motion and postural data. The collected data is transmitted to a PDA (mobile device) and/or transmitted to a base station (personal computer) or to the respective medical server through GPRS/Bluetooth, physician or caregiver for further intervention [30].	19
<b>2.2:</b> Review structure encompassing various modalities of home-based activity monitoring systems.	21
<b>2.3:</b> H-CAD Portable Unit [77].	22
<b>2.4:</b> The Portable Continuous Passive Motion Elbow Device [80].	23
<b>2.5:</b> Development of RUPERT. Left: Initial design which has a robot arm attached to a chair. Middle: A computer generated picture of the new robot arm. Right: The new RUPERT on a subject's body [81].	24
<b>2.6:</b> Summary of a telemedicine system using virtual environment for stroke rehabilitation [83].	25
<b>2.7:</b> The Microsoft Kinect and the blue markers attached to a therapists' hand [20]	28
<b>2.8:</b> Activity Recognition Process Flow.	37
<b>2.9:</b> Signal patterns generated by a tri-axial gyroscope placed near the elbow for three repetitions of four different arm movements – reach & retrieve; lift hand; swinging arm in the horizontal plane; rotate wrist.	40
<b>3.1:</b> Sensor unit showing direction of positive response for tri-axial accelerometers.	55
<b>3.2:</b> Illustration of Shimmer 9DoF sensor module enclosed in housing and resting on sloped wedges in different orientations during calibration [183].	56
<b>3.3:</b> The eight standard orientations of the Shimmer sensor module when rotated about the Z-axis accelerometer.	57
<b>3.4:</b> Sensor module rotated about the X-axis accelerometer.	57
<b>3.5:</b> Sensor module rotated about the Y-axis accelerometer.	57
<b>3.6:</b> Tri-axial accelerometer output with the module inclined at 0° and rotated about the positive Z-axis. The markers (vertical dashed lines) represent a reading window of 200 sensor data points for each of the four standard orientations (Position A, B, C, D).	58
<b>3.7:</b> Tri-axial accelerometer output with the module inclined at 0° and rotated about the negative Z-axis. The markers (vertical dashed lines) represent a reading window of 200 sensor data points for each of the four standard orientations (Position E, F, G, H).	58
<b>3.8:</b> Example accelerometer calibration data showing highly linear responses over acceleration range $\pm 1 g$ with high correlation ( $R^2$ ) values [183].	60
<b>3.9:</b> Principle of operation for MEMS vibrating gyroscope [125].	61
<b>3.10:</b> Gyroscope output with the sensor module placed in position A at various speeds (stationary, 33 and 45 rpm) alternatively.	62
<b>3.11:</b> Gyroscope calibration curve showing a highly linear response with the X-axis response counter-clockwise to the Y and Z-axis.	63
<b>3.12:</b> A stationary accelerometer with Z-axis facing gravity, showing a normal distribution with a mean acceleration of 9.804 m/s <sup>2</sup> .	64
<b>3.13:</b> Y-axis reading of a stationary gyroscope with a mean rotation rate of -0.0128 °/s.	65
<b>4.1:</b> Operation of a differential capacitive type MEMS accelerometer. The rectangle with broken line shows	

the initial position of the seismic mass when no acceleration is experienced in the direction of the accelerometer axis and the rectangle in solid lines (green) represents the mass in displaced condition.	71
<b>4.2:</b> Predefined orientations of the sensor module with respect to the direction of gravity, showing positive directions of accelerometer axes [183]. An illustration of the sensor worn on the wrist is also presented.	72
<b>4.3:</b> Basic steps of movement detection using sensor orientation[183]	73
<b>4.4:</b> Flowchart for arm movement detection [183].	74
<b>4.5:</b> Transition from Position 3 to Position 6, corresponding to a drinking type of activity (Action B) with sensor on right arm [183].	76
<b>4.6:</b> Setup for the real-time recognition of arm movements using the sensor orientation algorithm.	80
<b>4.7:</b> Architectural overview of the sensor based orientation algorithm.	81
<b>4.8:</b> Architecture for computing the maximum from incoming data samples.	81
<b>4.9:</b> Simulation of the Orientation Detection module with a sample X, Y and Z inputs (scaled up), illustrating the detected orientation states (OrientationType) and the readyOD signal.	82
<b>4.10:</b> Simulation of Sequence Detection with the orientation changes stored in the Type register and the readySD signal.	83
<b>4.11:</b> Architecture for Action Detection (AD).	84
<b>4.12:</b> Simulation showing the detection of Action B for an orientation transition from position 1 to 5 in the Type register.	86
<b>4.13:</b> LCD display for the acceleration data (left) and 7-segment display connections to the Altera FPGA (right) [194].	87
<b>5.1:</b> Methodology used to evaluate data types and learning algorithms.	90
<b>5.2:</b> Typical example of two-class classification in 2-D feature space using LDA [198].	99
<b>5.3:</b> Typical example of two-class classification in 2-D feature space using SVM [125].	100
<b>5.4:</b> An example confusion matrix for four classes.	102
<b>5.5:</b> Overview of the generalized classification approach.	103
<b>5.6:</b> Generalized classification - sensitivities for each task using the wrist accelerometer and gyroscope modulus signal (mod), individual sensor signals (X, Y, Z) and fused signals with LDA, QDA and SVM. The number of features required for each signal and sensor type is shown at the top of each group.	104
<b>5.7:</b> Generalized classification - sensitivities for each task using the elbow accelerometer and gyroscope modulus signal (mod), individual sensor signals (X, Y, Z) and fused signals with LDA, QDA and SVM. The number of features required for each signal and sensor type is shown at the top of each group.	104
<b>5.8:</b> Overview of the personalized classification approach.	107
<b>5.9:</b> Personalized classification – sensitivities for each task using the wrist gyroscope individual signals, with LDA, QDA and SVM for each of the 5 subjects. The number of features required for each subject is shown at the top of each group.	108
<b>5.10:</b> Classification results using LDA with data from wrist gyroscope for Subject 4 (stroke survivor).	114
<b>6.1:</b> Illustration of the clustering and minimum distance classifier based methodology.	121
<b>6.2:</b> Basic stages of the data processing [146].	122
<b>6.3:</b> Overview of cluster formation and cross validation on the <i>Training</i> phase data collected from each sensor and each subject [209].	127
<b>6.4:</b> Data from a tri-axial accelerometer located on the wrist collected while performing arm actions A, B and C from a healthy subject (upper) and a stroke survivor (lower) [183].	129
<b>6.5:</b> Data from a tri-axial rate gyroscope located on the wrist collected while performing arm actions A, B	

and C from a healthy subject (upper) and a stroke survivor (lower) [209].	130
<b>6.6:</b> Change in accuracy with number of features for healthy Subject 2 with accelerometer data using Euclidean distance.	132
<b>7.1:</b> Abstract sensor plane overview	142
<b>7.2:</b> Overview of the offline-online processing – the <i>training</i> dataset is processed offline and the <i>testing</i> dataset is processed online. The computation of the selected features on the <i>testing</i> data and computation of the minimum distance from the pre-computed cluster centroids was done in ASIC for real-time detection of arm movements.	142
<b>7.3:</b> Overview of the envisaged sensor node with the ASIC, microcontroller and memory.	143
<b>7.4:</b> Overview of CORDIC vector rotation from $[X_0, Y_0]$ to $[X_1, Y_1]$ through an angle $\theta$ using a series of micro-rotations in the clockwise or anti-clockwise direction.	145
<b>7.5:</b> Architecture for computing kurtosis.	150
<b>7.6:</b> Architecture for computing skewness.	151
<b>7.7:</b> Architecture for computing maximum and minimum values of a signal.	151
<b>7.8:</b> The selected bins from normal distribution of data samples.	152
<b>7.9:</b> Architecture for computing the probability of each bin of the signal.	152
<b>7.10:</b> Architecture for computation of information entropy.	153
<b>7.11:</b> Pipelined architecture for 4-stage CORDIC for computing <i>rms</i> on data stream having eight samples.	155
<b>7.12:</b> Architectural overview of the CORDIC operation for feature extraction.	156
<b>7.13:</b> Comparison of hardware complexity for a CORDIC and non-CORDIC based architecture for feature extraction, showing variation in number of full adders (FA) required with change in word-length.	164
<b>7.14:</b> Design overview for the offline-online processing involving clustering and minimum distance classification.	165
<b>7.15:</b> Sequence of features extracted from each tri-axial data segment to form a 30-bit <i>feature-code</i> .	166
<b>7.16:</b> Illustration of the minimum distance classification methodology.	167
<b>7.17:</b> Overview of the minimum distance classifier architecture.	169
<b>7.18:</b> Architecture for the minimum distance computation module.	169
<b>7.19:</b> RTL simulation of the hardware module encompassing feature extraction, selection and the computation of the Euclidean distance. The predicted cluster label is A ( <i>group – 01</i> ).	173
<b>7.20:</b> Core chip layout with all pin assignments.	174



# List of Tables

<b>2.1:</b> State-of-the-art systems developed in recent years for ADL monitoring.	47
<b>3.1:</b> Streamlined Wolf Motor Function Test	53
<b>3.2:</b> List of mean sensor readings and error tolerance ( $\pm 3\sigma$ ) when the sensor module is left static on a flat surface for 24 hours.	65
<b>3.3:</b> Drift in the sensitivity ( $S_A$ ) and offset ( $C_A$ ) values for the accelerometer for a sensor module recorded one month apart.	65
<b>3.4:</b> Drift in the sensitivity ( $S_G$ ) and offset ( $C_G$ ) values for a gyroscope recorded one month apart.	65
<b>4.1:</b> Use case activity list - ‘Making-a-cup-of-tea’	68
<b>4.2:</b> Sequence transitions and corresponding actions. Transitions for Action A involve additional processing to distinguish from D.	76
<b>4.3:</b> Recognition of trials for healthy subjects [183].	78
<b>4.4:</b> Recognition of trials for stroke survivors [183].	79
<b>4.5:</b> Computing logic for orientation states.	82
<b>4.6:</b> Recognition of trials for healthy subjects.	85
<b>4.7:</b> Recognition of trials for stroke survivors.	85
<b>4.8:</b> Execution time for each hardware module, with $\text{lenSeq} = 512$ and $\text{lenPos} = 8$ .	85
<b>5.1:</b> Use case activity list - ‘Making-a-cup-of-tea’	91
<b>5.2:</b> Data structure for each of the four sensor-position combinations for the <i>training</i> phase collected from healthy subjects.	93
<b>5.3:</b> Definition of fused signals for each arm movement.	94
<b>5.4:</b> List of the features considered in this exploration.	94
<b>5.5:</b> Generalized classification results using LDA, QDA and SVM.	105
<b>5.6:</b> Personalized classification results with individual sensor signals using LDA.	107
<b>5.7:</b> Personalized classification results with individual sensor signals using QDA.	108
<b>5.8:</b> Personalized classification results with individual sensor signals using SVM.	108
<b>5.9:</b> Data structure for prospective evaluation for each of the four sensor-position combinations using individual sensor signals.	111
<b>5.10:</b> Personalized classification results with individual signals using LDA for healthy subjects	111
<b>5.11:</b> Data structure for prospective evaluation involving stroke survivors for each of the four sensor-position combinations using individual sensor signals.	112
<b>5.12:</b> Cross-validation of the <i>training</i> dataset using a <i>personalized approach</i> with individual signals using LDA for stroke survivors.	113
<b>5.13:</b> Prospective evaluation of the <i>testing</i> dataset with individual signals using LDA for stroke survivors.	113
<b>5.14:</b> Activity-list showing actual and predicted action labels for Subject 4 (stroke survivor).	115
<b>5.15:</b> Overall accuracy for the prospective study for healthy subjects and stroke survivors for each sensor-position.	117
<b>6.1:</b> Use case activity list - ‘Making-a-cup-of-tea’	123
<b>6.2:</b> Data structure for each of the two sensor-position combinations using individual sensor signals.	123
<b>6.3:</b> Recognition sensitivities for each arm movement with accelerometer data for healthy subjects.	131
<b>6.4:</b> Recognition sensitivities for each arm movement with gyroscope data for healthy subjects.	131

<b>6.5:</b> Selected features in ranked order for healthy subjects	132
<b>6.6:</b> Recognition sensitivities for each arm movement with accelerometer data for stroke survivors.	133
<b>6.7:</b> Recognition sensitivities for each arm movement with gyroscope data for stroke survivors.	133
<b>6.8:</b> Selected features in ranked order for stroke survivors.	134
<b>6.9:</b> Sensitivities for each arm movement and overall accuracies for healthy subjects using LDA.	135
<b>6.10:</b> Sensitivities for each arm movement and overall accuracies for stroke survivors using LDA.	135
<b>6.11:</b> Sensitivities for each arm movement and overall accuracies for healthy subjects using SVM.	135
<b>6.12:</b> Sensitivities for each arm movement and overall accuracies for stroke survivors using SVM.	135
<b>7.1:</b> Generalized CORDIC algorithm in three Co-ordinate systems [41].	147
<b>7.2:</b> Computation time for the features employing the CORDIC engine.	159
<b>7.3:</b> Arithmetic operations required in the proposed CORDIC-based architecture for feature extraction	159
<b>7.4:</b> Arithmetic operations required in non-CORDIC-based architecture for feature extraction	161
<b>7.5:</b> Average error between Matlab and RTL simulation for feature extraction	165
<b>7.6:</b> Recognition sensitivities for each arm movement with accelerometer data for 2 healthy subjects.	172
<b>7.7:</b> Recognition sensitivities for each arm movement with gyroscope data for 2 healthy subjects.	172
<b>7.8:</b> Recognition sensitivities for each arm movement with accelerometer data for 2 stroke survivors.	172
<b>7.9:</b> Recognition sensitivities for each arm movement with gyroscope data for 2 stroke survivors.	172
<b>7.10:</b> Average power measurements of the 95 working chips.	174

# Acknowledgement

I would like to thank my supervisor Dr Koushik Maharatna, for providing me with the opportunity to work on the project, 'StrokeBack', which has provided me a platform to work upon new research challenges. I would also like to take this opportunity to thank Dr Andy Cranny, who has worked as a senior research fellow in this project and has provided constant support and guidance to me throughout the duration of my research.

I would also like to thank all the volunteers (healthy subjects and stroke survivors) for their assent and kind cooperation during the experiment sessions. The experiments, data collection and corresponding analysis would have been impossible without the help of the clinicians at the Brandenburg Klinik, Berlin, Germany.

The reviews and suggestions provided by my examiner, Professor Steve Beeby, during the nine month and eighteen moth stages have been really helpful towards writing this thesis.

Finally, I would like to dedicate this thesis to my parents and my wife who have provided constant support and motivation and have stood by my side through these years. They have equally shared my moments of happiness and sadness and have kept me going.



# Abbreviations

<i>GDP</i>	Gross domestic product
<i>BBK</i>	Brandenburgh klinik, berlin, germany
<i>RFID</i>	Radio-frequency identification
<i>WBAN</i>	Wireless body area network
<i>BAN</i>	Body area network
<i>BSN</i>	Body sensor networks
<i>ADL</i>	Activities of daily living
<i>BADLs</i>	Basic activities of daily living
<i>IADLs</i>	Instrumental activities of daily living
<i>AAL</i>	Ambient assisted living
<i>SoC</i>	System on chip
<i>VLSI</i>	Very large scale integration
<i>RTL</i>	Register transfer level
<i>ASIC</i>	Application specific integrated circuit
<i>FPGA</i>	Field programmable gate array
<i>DSP</i>	Digital signal processing
<i>WSN</i>	Wireless sensor network
<i>MEMS</i>	Microelectromechanical systems
<i>IMU</i>	Inertial measurement unit
<i>DoF</i>	9 degree-of-freedom
<i>ECG</i>	Electrocardiogram
<i>EMG</i>	Electromyogram
<i>EEG</i>	Electroencephalography
<i>MRI</i>	Magnetic resonance imaging
<i>CVA</i>	Cerebrovascular accident
<i>FES</i>	Functional electrical stimulation
<i>ILC</i>	Iterative learning control
<i>CIMT</i>	Constrained induced movement therapy
<i>WMFT</i>	Wolf motor function test
<i>FAS</i>	Functional ability scale
<i>FMA</i>	Fugl-meyer assessment
<i>FOG</i>	Freezing of gait
<i>B&amp;B</i>	Box and block
<i>NHP</i>	Nine hole peg test
<i>VR</i>	Virtual reality
<i>LCD</i>	Liquid crystal display
<i>LEDs</i>	Light emitting diodes
<i>GSM</i>	Global system for mobile communications

<i>PHR</i>	Patient health record
<i>PDA</i>	<i>Personal digital assistant</i>
<i>sfs</i>	Sequential forward selection technique
<i>LDA</i>	Linear discriminant analysis
<i>QDA</i>	Quadratic discriminant analysis
<i>SVM</i>	Support vector machines
<i>NB</i>	Naïve bayes
<i>k-NN</i>	<i>K</i> -nearest neighbour
<i>DT</i>	Decision tree
<i>HMM</i>	Hidden markov model
<i>DBNs</i>	Dynamic bayesian networks
<i>CRFs</i>	Conditional random fields
<i>EM</i>	Expectation maximization
<i>ANN</i>	Artificial neural network
<i>MLP</i>	Multi-layer perceptron
<i>STFT</i>	Short time fourier transform
<i>FFT</i>	Fast fourier transform
<i>WT</i>	Wavelet transform
<i>FDR</i>	Fisher discriminant ratio
<i>PCA</i>	Principal component analysis
<i>ICA</i>	Independent component analysis
<i>TPs</i>	True positives
<i>FNs</i>	False negatives
<i>FPs</i>	False positives
<i>ROC</i>	Receiver operating characteristics
<i>rms</i>	Root mean square
<i>inf. entropy</i>	Information entropy
$\sigma$	Standard deviation
$\mu$	Mean
<i>D</i>	Index of dispersion
<i>abs.diff</i>	Absolute difference
<i>CORDIC</i>	Coordinate rotation digital computer
<i>CSA</i>	Carry save arithmetic
<i>FA</i>	Full adder
<i>RCA</i>	Ripple carry adder
<i>CAM</i>	Conventional array multiplier
<i>NAD</i>	Non-restoring array divider
<i>SQRT</i>	Non-restoring iterative cellular square rooter
<i>LUT</i>	Look-up table
<i>CPA</i>	Carry propagate adder

# 1. Chapter 1

## Introduction

Increased life expectancy due to better medical facilities in developed nations has increased the prevalence of health impairments among the constantly increasing ageing population. The world's elderly population (on or above the age of 65 years) which stands at 650 million at present is expected to reach 2 billion by 2050. In the European Union, the 85+ population is expected to increase three times by 2060 [1]. This implies that with more life expectancy there would be a higher prevalence of health impairments necessitating increased expenditure in their healthcare and an intensive support system for their daily life operations [2].

Among the many diseases plaguing the elderly population is Cerebrovascular accident (CVA) or popularly termed as stroke caused primarily due to blockages in the blood supply to the brain, depriving the oxygen supply to the brain tissue which in turn causes the brain cells to dysfunction and eventually die. It is considered as a medical emergency due to its impact leading to death or physical disabilities caused by neurological disorders and ranks second only to coronary heart diseases [3]–[5]. Nearly 2 million people are affected each year in Europe out of which nearly 100,000 people are from the UK [6]. About one-third of the patients who survive, more than 670,000, return home with permanent disabilities leading to a significant reduction in the quality of their lives post-stroke [7]. The effect of stroke on human beings is different for each individual based on the damaged area of the brain but in general leads to paralysis, impaired vision, memory loss, speech and language problems [8], [9]. Due to the prevalence of ageing in the European societies the number of stroke affected people is expected to be on the rise and is predicted to account for 6.2% of the total burden of illness by 2020 [3]. This alarming trend is also prevalent in rest of the world [10].

Survivors require long term rehabilitation and support for leading an independent life, which includes costly human intervention in the form of immediate carers, occupational (ergo therapeutic) and physiotherapeutic sessions and visit of the therapists to the patient's home. The total cost of stroke in the European Union was calculated to be over 38 billion

in 2006 which included direct healthcare costs (about 49% of the total cost), productivity loss of the individuals due to disability and death (about 23% of the total cost) and care expenses (about 29% of the total cost) thus having a very high socio-economic impact. The ‘care expenses’ include inpatient treatment cost, outpatient hospital visits and long-term rehabilitation and care. In recent years, the expenses incurred on long term care have increased from 13% to 48% of the overall ‘care expenses’ [7]. Therefore, there is a need to address the ‘care expenses’ to reduce the incurred costs and also to maximise the number of patients being treated.

Patients who are left paralysed post-stroke are often treated with physiotherapy in order to restore their motor functionality. The level of physiotherapy needs to be as intensive as possible in order to ensure fast recovery [11]. Apart from conventional physiotherapy, rehabilitation therapies have been provided in clinical settings using functional electrical stimulation (FES), often involving the use of a controller like an iterative learning control (ILC) [12], [13]. Robotic technology was also employed in FES systems to assist patients in performing their rehabilitation exercises [14]. Some other methods employed are virtual reality [15] and constrained induced movement therapy [16], [17]. Such different rehabilitation modalities have been discussed at length in Chapter 2.

Although there has been partial success in achieving motor recovery in patients by these rehabilitation methods, the main bottleneck is their suitability in a clinical setting because they involve the use of many appliances and hence necessitates unnecessary patient transfer to the clinic from their respective homes. Moreover, the use of sophisticated robotic framework, correct placement of sensors and the use of specialised software require trained individuals for using the systems, which again diminishes their practicality in home settings. Hence, the clinical outcome of rehabilitation is quite unsatisfactory among a wide group of stroke survivors [5]. The high socio-economic impact of post-stroke rehabilitation necessitates the development of a telemedicine system for devising a long-term and effective treatment strategy in home settings thereby also helping to reduce costly human intervention [18].

## **1.1 Home-based rehabilitation for stroke survivors**

Most survivors feel psychologically and physically more comfortable in their home than in a hospital and therefore home based treatment helps in delivering an increased speed of

rehabilitation [7]. A specialised application of telemedicine for stroke rehabilitation called ‘Telestroke’, has been in practise for over two decades and has had major success in delivering ‘around-the clock’ specialist clinical evaluation of stroke survivors in home settings [19]. The factors that have significantly contributed towards the advent of home-based rehabilitation can be summarised as:

- it provides logistical convenience since it helps to reduce unnecessary patient transfer to the clinic or hospital,
- it is cost-effective since it helps to reduce expenses incurred towards hospitalization and human intervention,
- advances in computer-based technology to provide audio-visual interaction between patients and clinicians and wireless sensor technology have helped in remote monitoring of patients and
- it has further helped to maximize the number of patients being treated at home across a wide spectrum of the population [19].

A typical home-based stroke rehabilitation system comprises of the survivor performing various exercises as prescribed by the clinicians. Many systems further aim to motivate the survivors by emulating the rehabilitation exercises/training through games played over gaming consoles thereby preventing the exercises from being a mundane activity and also expediting the rehabilitation process. Vision-based systems like Kinect motion capture [20] or simple camera-based systems have also been popularly used but are mainly restricted to indoor activities within a defined region and require an un-hindered surveillance of the survivor. Further, installation of video cameras within the home is generally not preferred by the survivors since it intrudes into their privacy. Sensors are also placed within the home environment which helps to track the use of various objects that are typically used in daily life using radio-frequency identification (RFID) [21].

The advancements in wireless sensor networks (WSN) have enabled the monitoring of human body parts across a wide range of applications including remote health monitoring [22], [23], human computer interaction [24], [25] and sports medicine [26], [27]. The advent of low-cost, body-worn, miniaturised inertial sensors have helped in long-term capturing of kinematic data from the human body [28], [29]. Therefore, wearable inertial sensors coupled with the advantages of wireless communication, forming a body area network (BAN) system are widely used for patient monitoring within home settings [30].

The data collected from these sensors are transmitted wirelessly to a remote server where they are analysed to infer the rehabilitation progress of the patients. The data are generally uploaded onto a patient database commonly referred to as the patient health record (PHR). The PHR can be accessed by the respective clinicians for assessing the recovery of the patient at pre-decided intervals of time and if required they can prescribe modifications to the rehabilitation protocol. Hence, the PHR can be utilised as a pre-emptive means to have a regular update on the well-being of the patients [7], [31]-[32].

## 1.2 Shortcomings of the existing telemedicine modalities

The most significant drawback for a majority of remote monitoring systems at present is that they are only effective for characterising the patient's performance during a constrained *exercise phase* without monitoring them in the *nomadic* environment (i.e. real life) wherein the patients undertake various daily activities. A quantitative assessment on the usage of the affected body parts (e.g. impaired limb) in the *nomadic* environment, which essentially involves a longer monitoring duration than the stipulated *exercise phase*, would be a natural indicator of rehabilitation progress and therefore also help the clinicians to formulate an objective feedback.

Moreover, in typical health monitoring systems, the physiological data collected from the patient's body through various sensors are transmitted continuously to a remote server at the back office service platform in two ways - wirelessly over a GSM network or through the patient's Bluetooth enabled mobile device, where the corresponding procedure of analysing the transmitted signal takes place using 'computationally intensive signal processing and classification techniques' [33]. The fundamental problem with continuous data transmission is the energy requirement which is expended at the radio-front end of the sensors. Hence, continuous transmission of data, affects the operational life-time of the battery operated sensors, thereby rendering them ineffective for long-term remote monitoring applications. Therefore, in such wearable systems the data analysis primarily needs to be carried out at the sensor node itself, which has been shown to yield a more energy efficient solution compared to the conventional continuous data transmission based remote monitoring approach, provided the complexity of data analysis algorithms is reduced [33]. Since the energy consumption is proportional to the computational complexity of the data processing algorithm used, the use of high-complexity algorithms (although they may be more 'accurate') will drain the battery faster, defying the objective

of long-term monitoring. Hence, from the perspective of long-term system operation, when implementing a body-worn wireless sensor node comprising of different sensors, it is essential to select data analysis algorithms that are computationally of low-complexity.

### 1.3 Motivation

The highlighted shortcomings act as a motivation to detect the occurrences of the movements performed by the stroke survivors with their impaired body parts at home settings especially in an unconstrained *nomadic* environment. This research work particularly focuses on the domain of upper limb rehabilitation for stroke survivors, since the impairment of the upper limb is a common problem among a majority of survivors which renders them ineffective in performing their essential activities in daily life [34] [35] [35], [36]. Therefore, the aim is to detect the use of the impaired arm of the stroke survivors during the *exercise phase* and during normal daily activities (i.e. *nomadic* environment) using minimal number of sensors. The classification and enumeration of the occurrences of specific arm movements (e.g. prescribed exercises) over time can be indicative of rehabilitation progress since the frequency of these movements is more likely to increase as the motor functionality of the person improves.

In view of this motivation, there is a two-fold objective:

- to explore effective low-complexity algorithmic techniques for detecting specific upper limb movements in an unconstrained *nomadic* environment without compromising on the accuracy of movement detection and
- translating the algorithms to dedicated low-power hardware which can be used within a resource constrained body-worn sensor node with an aim of enhancing its battery life for a sustained operation satisfying the notion of long-term continuous monitoring.

## 1.4 Research challenges

The primary challenges towards fulfilling the research objectives can be summarised as:

- Intending to use only low-cost, wireless, body-worn inertial sensors for collecting and analysing the kinematic data leading to movement detection. Sensor characteristics present a significant challenge for long-term monitoring of activities as the raw sensor data is often corrupted by noise and artifacts (e.g. drift) [37].
- Recognising a particular set of arm movements in an unconstrained *nomadic* environment, using minimal number of inertial sensors (from a convenience and mobility point of view) is challenging owing to the very high degree of variability in human movement [38].
- Developing low-complexity data analysis algorithms which can be conveniently mapped onto a hardware platform requires a holistic approach encompassing algorithm-to-architecture mapping, performance analysis and optimisation for energy consumption for sustained battery operation.

Therefore, a prerequisite is to determine the optimal number/combination of inertial sensors, their placements and development of appropriate algorithmic techniques which can be translated to hardware, enabling accurate detection of a number of movements.

## 1.5 Research contributions

In this work, the focus lies on detecting three elementary types of arm movement, which were chosen since they constitute a significant proportion of the complex movements performed with the upper limb in daily life, besides also resembling three tasks in the standard Wolf Motor Function Test (WMFT) which is an established clinical assessment method for quantifying upper extremity motor ability [32-34]. The three movements (along with examples of their daily occurrence) are - extension/flexion of the forearm (reach and retrieve object); rotation of the forearm about the elbow (drinking action); and rotation of the arm about the long axis of forearm (opening a door, using a key or pouring action). In view of this, the primary contributions of this research work can be outlined as follows:

- Development of three low-complexity algorithmic techniques for recognising the

occurrence of the three arm movements performed within a controlled environment and also in *nomadic* settings. – 1) tracking the orientation of a wrist-worn inertial sensor with respect to the performed movements; 2) *k*-means clustering and minimum distance classifier and 3) movement classification using the supervised learning algorithms: linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and support vector machines (SVM).

- The development of the algorithms also encompasses a detailed exploration to determine the optimal number and placement of inertial sensors on the subject's arm.
- The developed algorithms were evaluated using a 'personalized' approach with the kinematic data collected from four healthy subjects and four stroke survivors who performed the movements in two phases – 1) a controlled environment (within laboratory) and 2) repeated trials of a daily activity (out-of-laboratory, i.e. kitchen). The corresponding results (movement recognition accuracy) were analysed and compared to determine the best performing recognition methodology in accordance to the application scenario. The clustering based methodology and the sensor orientation based methodology proved to be successful.
- An efficient algorithm-to-architecture mapping was done for both the recognition methodologies to achieve an optimised low-power implementation on a hardware platform with an aim of using them in resource constrained body-worn wireless sensor nodes.
- The clustering based methodology which proved to be a flexible and scalable approach (in terms of incorporating new/more category of movements) was developed into a low-power novel Application Specific Integrated circuit (ASIC). On the other hand, the sensor orientation based methodology which was aimed at detecting only these three specific arm movements, yielded higher accuracy and was hence developed as a field programmable gate array (FPGA) based system to detect arm movements in real-time.

The development of two novel movement recognition algorithms (the sensor orientation based methodology and the clustering and minimum distance classifier based methodology), their successful evaluation and their implementation on a hardware platform – a FPGA-based real-time system and a low-power novel ASIC respectively, are the primary novel features of this research work. The developed systems could be used as a clinical tool to assess arm rehabilitation progress amongst stroke survivors by tracking the number of times the person performs specific arm movements with their paretic arm throughout the day.

## 1.6 Approach

The generalized view of a home-based monitoring system for stroke survivors has been illustrated in Figure 1.1. It mainly revolves around the patient at home who is monitored to track rehabilitation progress in two distinct scenarios – a controlled *exercise* phase and an uncontrolled *nomadic* phase. The movements performed by the impaired upper limb of the stroke survivors are recognised and this information could be uploaded to a patient database (PHR system) and referred by the clinicians at periodic intervals. The clinicians can modulate the training schedule and the exercises through the PHR or in direct consultation with the patient.

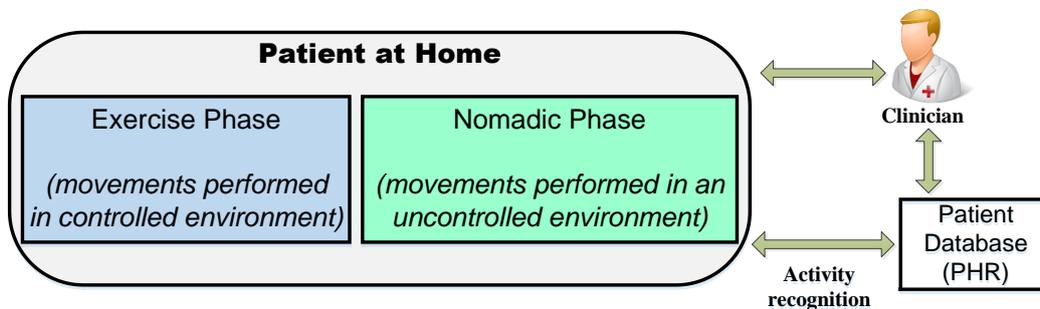


Figure 1.1: Overview of home-based monitoring for stroke survivors.

With respect to this overview, the approach is divided into three groups – 1) data acquisition through optimal sensors, 2) development of movement recognition algorithms and 3) hardware implementation, each of which is discussed briefly in the following sections.

### 1.6.1 Data acquisition

A high degree of variability inherent in human motion necessitates the development of a robust activity recognition system [38]. In view of this, the first step involves the collection of kinematic data using wireless inertial sensors attached to the subjects' body. An experimental protocol was designed where movements are performed in two phases:

- subjects perform multiple trials of the three enlisted movements (cf. section 1.5) in a controlled environment representative of the *training* or *exercise* phase. In this phase, the three movements are performed by the subjects under the supervision of the researcher in a laboratory. The activity recognition algorithm is expected to learn the

movement patterns collected from the sensor data during the *training* phase and recognize the occurrence of these specific movements during the *testing* phase.

- subjects then perform repeated trials of an archetypal activity of ‘*making-a-cup-of-tea*’, which includes multiple occurrences of extension, flexion and rotation of the forearm, representative of the *testing* or the *nomadic* phase. The subjects perform the movements in a semi-naturalistic or an uncontrolled environment, termed in literature as being an intermediate stage between the laboratory settings and a complete natural environment [39]. Here, the subjects perform the movements in an out-of-laboratory (i.e. real world) condition in a naturalistic manner. This might involve the movements being performed naturally and not in a time constrained manner.

Shimmer wireless kinematic sensors with 9 Degree-of-Freedom (DoF) comprising of tri-axial accelerometer, tri-axial rate gyroscope and tri-axial magnetometer are used as the sensing platform [40]. For the experiments performed in this work, only a tri-axial accelerometer and a tri-axial rate gyroscope were used. Two positions of the dominant upper limb proximal to the wrist and elbow were chosen as the sensing positions since they were envisaged to produce the largest sensor response, with respect to the arm movements being investigated.

## **1.6.2 Movement recognition algorithms**

As discussed, three low-complexity recognition approaches were developed to detect the arm movements performed by the subjects during the archetypal activity, each of which is discussed briefly.

### **(1) Sensor Orientation based movement recognition**

A simple recognition methodology was explored using a low-complexity processing technique that does not involve the overheads of training a system and can hence be used within a resource constrained wireless sensor node. Here, the three arm movements are recognized by mapping transitions of six predefined standard orientations of the wrist accelerometer to the corresponding arm movements investigated. The experimental results show that the proposed methodology can successfully recognize the three individual arm movements across the healthy subjects and stroke survivors.

## (2) Classification using LDA, QDA and SVM

Sensor data collected from individual sensor axes (X, Y and Z) and fused signals (modulus of the signals from respective sensors and specific accelerometer-gyroscope combinations) are pre-processed (noise and drift removal) and 30 time domain features are extracted. A wrapper approach based sequential forward selection technique (*sfs*) is used to select the optimum number of features and three learning algorithms are used to classify the movements – LDA, QDA and SVM. The classifiers are trained with subject-specific data collected in the *exercise* phase and evaluated on the data collected from the same subject during the *nomadic* phase. The LDA classifier, chosen due to its low computational complexity, when evaluated in conjunction with individual sensor signals produced the best overall accuracy.

## (3) Clustering and minimum distance classifier based movement recognition

The *exercise* phase (*training*) data are represented by a ranked set of 30 time-domain features. Using *sfs*, for each set of feature combinations three clusters are formed using *k*-means clustering ( $k=3$ ) followed by 10 runs of 10-fold cross validation on the *training* data to determine the best feature combinations. The movements performed in the *nomadic* phase (*testing*) are associated with each cluster label using a minimum distance classifier in a multi-dimensional feature space, comprised of the best ranked features, using Euclidean or Mahalanobis distance as the metric. The three movements were detected successfully with data acquired from the wrist-based accelerometers and gyroscopes across the healthy subjects and stroke survivors.

The algorithms were evaluated using the kinematic data collected from each individual subject – a personalized approach, adopted in view of the large degree of inter-person variability expected amongst the sample population. Furthermore, this would be beneficial when applied to monitoring individual patients who demonstrate differences in levels of impairment depending on their stage of rehabilitation.

### 1.6.3 Hardware design

Although the sensor orientation based methodology resulted in higher overall accuracy in detecting the three movements, the clustering based approach provides greater flexibility as

it can incorporate new types of movement categories other than those selected here and is also scalable since more number of movements can be incorporated by conveniently modifying the algorithm. This is quintessential from the clinical perspective since the prescribed exercise schedule might vary according to the motor impairments of the survivors and therefore new category of movements can be recognised. Hence, the clustering and minimum distance classifier based approach was implemented in hardware as a novel low-power ASIC which involved an efficient mapping of the algorithm to an optimized architecture design. The feature computation, cluster formation on the *training* data (being relatively time and memory intensive) were done in an offline mode (in software). The computation of the selected features on the *testing* data and the distance computation (Euclidean) of the features from the pre-computed cluster centroids was done in hardware. The computation of the time domain features on the *testing* data involving complex arithmetic operations were realized using the CoOrdinate Rotation Digital Computer (CORDIC) algorithm by exploiting the similarities in the mathematical formulation of the features [41]. The developed ASIC is envisaged to be used on a wearable sensor platform, aimed at real-time recognition of arm movements for long-term remote monitoring.

The approach based on sensor orientation which was specifically targeted towards recognising the three movements investigated here, was also validated in hardware and implemented on FPGA to design a system that detects performed arm movements in real-time. Data from a wrist-worn tri-axial accelerometer was transmitted in real-time to a host computer using Bluetooth which was processed and transmitted to the FPGA board through a RS232 cable. The algorithm was coded using a hardware description language and synthesized on the Altera DE2-115 Cyclone IV FPGA board which was used to detect the performed arm movements and display them on a seven segment display in real-time.

#### **1.6.4 Research Constraints**

This research work was part of the European Union Seventh Framework Programme under the project name “StrokeBack: Telemedicine system empowering stroke patients to fight back” [7]. The project envisages empowering the stroke patients with advancements in information and communication technology for rehabilitation within the home environment. Data acquisition from stroke patients was performed at the Brandenburg Klinik (BBK), Berlin, Germany, one of the clinical partners of the project. Kinematic data

was acquired from only four patients who were available within the stipulated time at BBK and agreed to participate in the experiments. Therefore, a selection of an age-matched group which is common for clinical trials was not available for this study. The fabrication, packaging, testing of the ASIC chip and its deployment in a customized body-worn sensor node for use in real-life was entrusted with the company IHP- Innovations for High Performance Microelectronics, Germany, one of the seven StrokeBack partners.

Hence, this work has been partly constrained by the availability of more stroke patients for data acquisition and evaluation of the developed algorithms on a wide population and on a longitudinal scale. Secondly, the use of the fabricated ASIC within a wearable sensor node for real-life patient trial to detect and enumerate the occurrence of the elementary arm movements have not been discussed here.

The primary focus of this work lies in developing low-complexity algorithmic techniques for recognising the occurrence of the three arm movements in a nomadic environment, an efficient algorithm-to-architecture mapping and its implementation on a low-power hardware with an aim of using them in resource constrained body-worn sensor nodes.

## **1.7 Organisation**

The rest of the thesis is organised as follows:

In Chapter 2, the relevant state-of-the-art remote health monitoring systems for activity recognition in a controlled environment and in daily life situations have been discussed. A detailed review of the relevant data processing techniques which includes feature extraction, selection and the recognition algorithms used have been presented in this chapter. Chapter 3 discusses the movement selection and the choice of sensors and the calibration methodologies adopted. In Chapter 4, the low-complexity sensor orientation based methodology to detect the arm movements and an optimized design and implementation of the algorithm to develop the FPGA-based system for real time arm movement detection, have been presented. The proposed supervised learning techniques employing different classification algorithms for recognising the selected upper limb movements have been discussed in Chapter 5. In Chapter 6, the clustering and minimum distance classifier based approach to detect the performed movements have been presented. Chapter 7, presents the architectural design and implementation of the minimum distance

classifier to develop a low-power novel ASIC. Finally, the conclusions are drawn in Chapter 8, where a comparative analysis has been presented for the two recognition methodologies – sensor orientation and clustering based movement recognition, besides discussing on future prospective work.

## 1.8 Publications

The contributions of this thesis can be enlisted in these publications some of which has already been published while a few are under review.

### Journal papers

- Evangelos B. Mazomenos, Dwaipayan Biswas, Andy Cranny, Amal Ranjan, Koushik Maharatna, Josy Achner, Jasmin Klemke, Michael Jöbges, Steffen Ortmann and Peter Langendörfer, “Detecting Elementary Arm Movements by Tracking Upper Limb Joint Angles with MARG Sensors”, *IEEE Journal of Biomedical and Health Informatics*, May 2015 (online), DOI: [10.1109/JBHI.2015.2431472](https://doi.org/10.1109/JBHI.2015.2431472).
- Dwaipayan Biswas and Koushik Maharatna, “A CORDIC-based Low-power Statistical Feature Computation Engine for WSN Applications”, *Circuits, Systems, and Signal Processing*, Apr. 2015 (online), DOI: [10.1007/s00034-015-0041-5](https://doi.org/10.1007/s00034-015-0041-5)
- Dwaipayan Biswas, Andy Cranny, Nayaab Gupta, Koushik Maharatna, Josy Achner, Jasmin Klemke, Michael Jöbges and Steffen Ortmann, “Recognizing Upper Limb Movements with Wrist worn Inertial Sensors using  $k$ -means Clustering Classification”, *Human Movement Science*, vol. 40, pp. 59-76, Apr. 2015, DOI: [10.1016/j.humov.2014.11.013](https://doi.org/10.1016/j.humov.2014.11.013).
- Dwaipayan Biswas, Andy Cranny, Ahmed Rahim, Nayaab Gupta, Koushik Maharatna, Nick Harris and Steffen Ortmann, “On the Data Analysis for Classification of Elementary Upper Limb Movements”, *Biomedical Engineering Letters*, vol. 4, no. 4, pp. 403-413, Jan. 2015, DOI: [10.1007/s13534-014-0160-0](https://doi.org/10.1007/s13534-014-0160-0).
- Dwaipayan Biswas, Daniele Corda, Giovanni Baldus, Andy Cranny, Koushik Maharatna, Josy Achner, Koushik Maharatna, Jasmin Klemke, Michael Jöbges and Steffen Ortmann, “Recognition of elementary arm movements using orientation of a tri-axial accelerometer located near the wrist”, *Physiological Measurement*, vol. 35, no. 9, pp. 1751-1768, Aug. 2014, DOI: [10.1088/0967-3334/35/9/1751](https://doi.org/10.1088/0967-3334/35/9/1751).

- Evangelos B. Mazomenos, Dwaipayan Biswas, Amit Acharyya, Taihai Chen, Koushik Maharatna, James Rosengarten, John Morgan and Nick Curzen, “A Low-Complexity Feature Extraction Algorithm for Mobile Healthcare Applications”, *IEEE Journal of Biomedical and Health Informatics*, vol. 17, no. 2, pp. 459-469, Mar. 2013, DOI: [10.1109/TITB.2012.2231312](https://doi.org/10.1109/TITB.2012.2231312).

### Conference papers

- Dwaipayan Biswas, Gerry Juans Ajiwibawa, Koushik Maharatna, Andy Cranny, Josy Achner, Jasmin Klemke, Michael Jöbges, “Real-time Arm Movement Recognition using FPGA”, *IEEE International Symposium on Circuits and Systems (ISCAS)*, 24-27 May. 2015, Lisbon, Portugal (accepted, to be presented).
- Dwaipayan Biswas, “Recognition of Upper Limb Movements for Remote Health Monitoring”, *IEEE Intelligent Informatics Bulletin*, vol. 15, no. 1, pp. 22-23, 15-17 Dec.2014, ([http://www.comp.hkbu.edu.hk/~iib/2014/Dec/paper1/iib\\_vol15no1\\_paper1.pdf](http://www.comp.hkbu.edu.hk/~iib/2014/Dec/paper1/iib_vol15no1_paper1.pdf)).
- Dwaipayan Biswas, Andy Cranny, Nayaab Gupta, Koushik Maharatna and Steffen Ortmann, “Recognition of Elementary Upper Limb Movements in an activity of Daily Living Using Clustering of Data from Wrist Mounted Accelerometers”, *IEEE International Conference on Healthcare Informatics (IHCI)*, pp. 232-237, 15-17 Sept. 2014, Verona, Italy, DOI: [10.1109/ICHI.2014.40](https://doi.org/10.1109/ICHI.2014.40).
- Evangelos B. Mazomenos, Andy Cranny, Dwaipayan Biswas, Nick Harris, Koushik Maharatna, “An Investigation into the Accuracy of Calculating upper Body joint Angles Using MARG Sensors”, 28<sup>th</sup> Conference on Solid-State Transducers (*Procedia Engineering*), vol. 87, pp. 1330-1333, 07–10 Sept. 2014, Brescia, Italy, DOI: [10.1016/j.proeng.2014.11.693](https://doi.org/10.1016/j.proeng.2014.11.693).
- Emmanouela Vogiatzaki, Yannis Gravezas, Nikos Dalezios, Dwaipayan Biswas, Andy Cranny, Steffen Ortmann, Peter Langendörfer, Ilias Lamprinos, Gioula Giannakopoulou, Josy Achner, Jasmin Klemke and Holger Jost, “Telemedicine System for Game-Based Rehabilitation of Stroke Patients in the FP7-StrokeBack Project”, *IEEE European Conference on Networks and Communications (EuCNC)*, pp. 1-5, 23-26 Jun. 2014, Bologna, Italy, DOI: [10.1109/EuCNC.2014.6882688](https://doi.org/10.1109/EuCNC.2014.6882688).
- Dwaipayan Biswas, Andy Cranny, Ahmed Rahim, Nayaab Gupta, Nick Harris, Koushik Maharatna and Steffen Ortmann, “On the Sensor Choice and Data Analysis for Classification of Elementary Upper Limb Movements”, *IEEE Biomedical and Health Informatics (BHI)*, pp. 744-747, 1-4 Jun. 2014, Valencia, Spain, DOI: [10.1109/BHI.2014.6864471](https://doi.org/10.1109/BHI.2014.6864471).

- Dwaipayan Biswas, Andy Cranny, Koushik Maharatna, Josy Achner, Jasmin Klemke, Michael Jöbges and Steffen Ortmann, “Movement Fluidity of the impaired Arm during Stroke Rehabilitation”, *IEEE Biomedical and Health Informatics (BHI)*, 1-4 Jun. 2014, Valencia, Spain, (<http://eprints.soton.ac.uk/368128/>).
- Dwaipayan Biswas, Evangelos B. Mazomenos and Koushik Maharatna, “ECG compression for remote healthcare systems using selective thresholding based on energy compaction”, *IEEE International Symposium on Signals, Systems, and Electronics (ISSSE)*, pp. 1-6, 3-5 Oct. 2012, Potsdam, Germany, DOI: [10.1109/ISSSE.2012.6374306](https://doi.org/10.1109/ISSSE.2012.6374306).

### **Book chapter**

Dwaipayan Biswas, Andy Cranny and Koushik Maharatna, “Body Area Sensing Networks for Physiological Monitoring”, In: *Telemedicine System for Home-based Stroke Rehabilitation*, Eds: Emmanouela Vogiatzaki and Artur Krukowski, Springer, New York. (in press).



## **2. Chapter 2**

# **Background and Literature Review**

### **2.1 Introduction**

The after-effects of stroke pose a serious socio-economic challenge in terms of loss of lives, disability amongst the survivors and the expenses incurred towards their rehabilitation and care. A common after-effect of stroke in the UK is impaired arm functionality which is seen in around 70% of cases, with 40% of them having a completely non-functioning arm [35], leading to a degeneration of their lifestyle. Hence, rehabilitating the limb is a key factor to help the survivors regain functional independence. Besides helping them to perform their daily activities, an improved rehabilitation mechanism would help in reducing expenses incurred towards care and disability services [36].

Rehabilitation, in general is primarily aimed at restoring a level of physical and psychological functioning within patients that allows them to re-integrate themselves back into their daily life [42]. Rehabilitation guidelines are specifically bespoke to the requirements of each individual with an aim towards optimizing balance, mobility and gait functionality. Intensive physiotherapy [11], use of robotic technology in FES systems involving an ILC controller [12], [13] are some of the popular means adopted within a clinical setting, to restore motor functionality amongst the stroke survivors. Physiotherapy sessions are also carried out at the patient's home. Monitoring of physical activity and specific physiological parameters are the key components in interventions aimed at maintaining health and well-being, preventing falls, reducing motor functionality loss and the risk of recurrent stroke [43].

Physical activity has traditionally been monitored by questionnaires citing its usefulness in covering large subject groups and cost effectiveness [44], [45]. These questionnaires can be filled by the patients themselves on a daily basis from their home, manually or electronically through computing facilities. This data helps to formulate a patient activity log which is monitored periodically by the respective clinicians. There are also a wide

range of clinical tests that are quite popular and performed by the therapists to assess arm rehabilitation of survivors within a clinical environment. The wolf motor function test (WMFT) [5], Box and Block (B&B) test [34] and the Nine Hole Peg (NHP) test [46] are some of the popular means to assess the patient's motor ability while they perform designated tasks under the observation of a therapist. These tests are generally associated with a scoring system which is used to quantify the performance of the subjects. However, subjective measures of physical activity and arm movements may overestimate activity levels as compared to assessments made by objective measures [47]. Hence, sensor-based telemedicine modalities provide an alternative towards objective measurement for remote patient monitoring [48], [49].

Telemedicine has long been influenced by technological advancement in the last century. Modern telemedicine took its flight in the 1960s when a television link was established between Nebraska Psychiatric Institute and Norfolk State Hospital [50] in order for therapists to share findings and discuss medical issues. With the development of information and communication technology (ICT), telemedicine has taken a giant leap towards more flexible solution to home-based healthcare [51]. This has helped in providing improved rehabilitation within home settings. Home-based patient monitoring is generally performed with vision-based, body-worn or ambient sensors. Vision-based sensors although quite effective suffer from occlusion problems and can only monitor movements performed within a designated zone. Ambient sensors (sensors on doors, objects of daily use such as RFID tags) have been used with an aim of developing smart homes that provide health assistance in the subject's living environment, also referred to as ambient assisted living (AAL) [52], [53]. With advancements in wireless communication technologies and integrated circuit design, the development of low-cost, power efficient, un-obtrusive and lightweight body-worn sensors have become very popular and are being used for wearable remote health monitoring applications. They have led to the development of a wireless body area network (WBAN) or body sensor networks (BSN) system which comprises of miniaturized wireless enabled sensor nodes. Apart from sensing, front-end amplification, microcontroller functions and radio transmissions have all been integrated into a single circuit thus resulting in a system-on-chip (SoC) implementation on board the sensors [54] [55].

These sensors can be used for: 1) physical activity monitoring by recording kinematic data and 2) continuous monitoring of significant physiological parameters like heart rate,

respiratory rate, blood pressure, blood oxygen saturation and muscle activity, which were only possible within the hospital settings [30]. Recent advancements in material science have led to the development of e-textile based systems which integrate sensing capability into garments for collecting ECG, EMG and kinematic data [30], [56]. Some health monitoring systems have also combined body-worn sensors and ambient sensors [52], [53]. In this context, information collected by the body-worn sensors can be augmented by the data from ambient sensors distributed throughout the home environment to determine activity patterns of the patients and provide feedback on living behaviors for a better health management.

A holistic overview of a wireless body area network system for remote healthcare monitoring is presented in Figure 2.1, comprising of sensors for capturing physiological parameters and motion data. The obtained measurements from the sensor nodes are transmitted to a central node which can be in the form of a personal digital assistant (PDA), a smart-phone, a pc or a microcontroller-based device. The central node acts as a gateway and can be used for displaying the vital information on a user interface or transmit the clinically relevant information to a remote medical center [57].

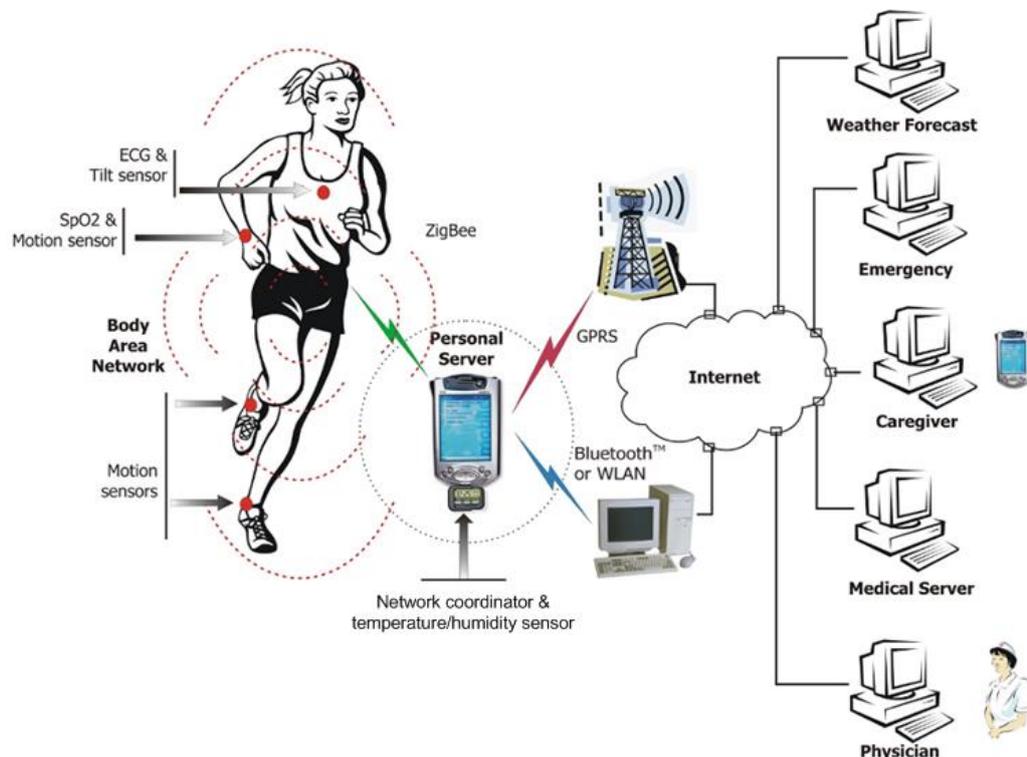


Figure 2.1: Overview of a wireless body area network system for remote healthcare monitoring, comprising of on-body sensors for monitoring physiological parameters (ECG, SpO<sub>2</sub>) and motion and postural data. The collected data is transmitted to a PDA (mobile device) and/or transmitted to a base station (personal computer) or to the respective medical server through GPRS/Bluetooth, physician or caregiver for further intervention [30].

The ever increasing research in remote health monitoring in recent years, has led to the development of many research prototypes [58]–[61], commercially available systems [62]–[68] and smart phone based systems [69]–[73]. These developments have enabled long-term physiological monitoring thereby improving the diagnosis and treatment of life threatening incidents involved in cardiovascular or neurodegenerative diseases [57]. A considerable research effort has also been spent towards fall detection, leading to emergency situations with the elderly population alone at home [74], [75]. This is commonly encountered in freezing of gait (FOG) disorders associated with Parkinson’s disease [76]. Physical activity monitoring in general for helping with motor balance and walking have been performed for post-stroke treatment [43].

In accordance to the research focus, in this chapter, activity monitoring of patients in home and community settings in two circumstances – a controlled exercise phase and an uncontrolled nomadic setting have been looked into.

## **2.2 Activity monitoring**

Activity monitoring is a well-researched and a broad topic. Human activity monitoring has gained prominence with the use of wearable sensors and video-based sensing technologies and is a major area of remote health monitoring systems. For post-stroke rehabilitation, tracking the number of times a patient performs specific movements (e.g. exercises) with their impaired body parts (e.g. paretic arm) during training and also throughout the day can provide useful information on the progress of the patient. The frequency of specific movements and the quality of the movements performed (e.g. fluidity/smoothness) is likely to increase as the motor functionality of the patient improves. It can also provide information on the patient’s compliance to the specific guidelines set by the respective clinicians during rehabilitation training.

The various exercise platforms, virtual reality (VR) based systems, gaming consoles and the widely popular Kinect camera based system used for rehabilitation training have been discussed. These approaches mainly aim to monitor the rehabilitation progress of the patients during the exercise or the training phase in a controlled environment within a designated zone (exercise/gaming platforms and vicinity of camera systems). The main difficulty with this approach is that it offers no possibility to monitor the movement quality of the patients and their compliance with the prescribed exercises in their natural

environment (i.e. while performing daily activities) which are more objective reflections of the actual rehabilitation state and of the effectiveness of the prescribed therapy. Therefore, there has been a growing demand to monitor patient activity in daily life. Hence, monitoring of activities performed in daily life in a nomadic environment have also been focused upon. A brief overview of the review structure is presented in Figure 2.2.

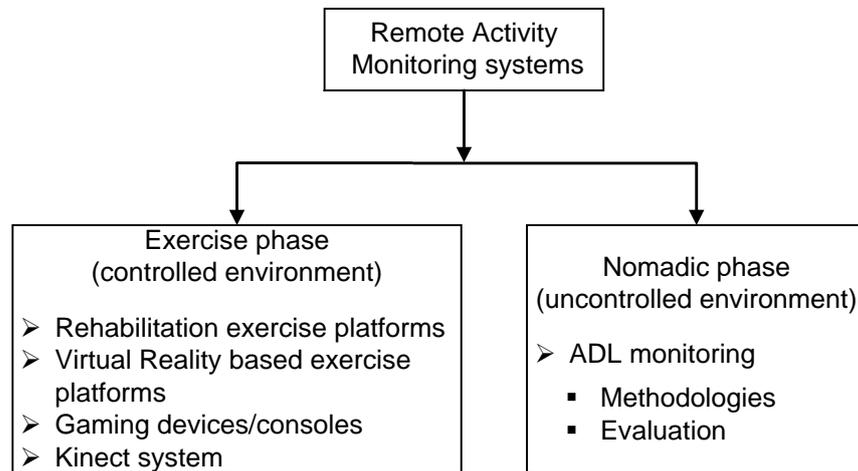


Figure 2.2: Review structure encompassing various modalities of home-based activity monitoring systems.

### 2.2.1 Rehabilitation exercise platforms

As the name suggests, in this section, a few state-of-the-art systems have been discussed which are particularly aimed at monitoring the patients within their home environment while they perform their rehabilitation exercises.

H-CAD (Home-care Activity Desk) built by Signio Motus is a task oriented program comprising of nine functional tasks that help practice every day functional tasks such as picking up keys, turning a switch on and off and opening drawer thus allowing patients suffering from multiple sclerosis, stroke or traumatic brain injury to rehabilitate at home [11]. It is composed of two subsystems: an in-hospital based server and a portable unit to be placed at the patients' home. The portable unit comprises of an activity desk which is placed in the home of the patient containing a computer (e.g. PC) with inbuilt software for controlling and monitoring the different rehabilitation activities, as well as allowing telecommunication facilities with a doctor. There are various sensors like a webcam to capture kinematic information, sensorized keyboard to measure pressure, sensorized table platform to measure pressure while the patient performs various upper limb activities. The setup is shown in Figure 2.3.

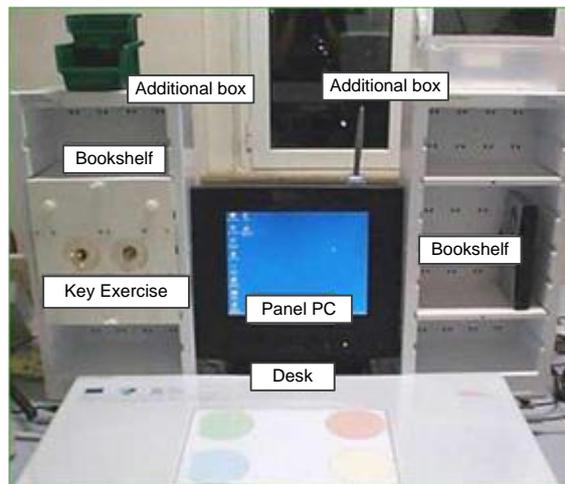


Figure 2.3: H-CAD Portable Unit [77].

The data collected by these sensors are stored on the local PC and sent to the hospital-based server. The server unit located at the hospital allows multiple portable units to be linked with it allowing the appropriate clinician to remotely access results as well as initiate telecommunication sessions with the patients through their portable unit. H-CAD was also used in HELLODOC project (healthcare service linking tele-rehabilitation to disabled people and clinicians) as a further development [78]. A pilot study was carried out in HELLODOC on 81 patients during a one-month period. The overall results of patient rehabilitation as assessed by conventional arm ability measures like action research arm test and nine hole peg test were sent to the clinicians remotely who confirmed the effectiveness of H-CAD system as being on similar lines as conventional physiotherapy.

A more complete home-based rehabilitation system, ‘Stroke Rehabilitation Exerciser’ focussing on compact design and minimal human-machine interaction was developed by Philips Research [79]. It accommodates several components to support the execution of functional exercises and assist the therapist in their assessment: a patient unit equipped with wireless inertial sensors, a graphical user interface and a therapist station. Sensors attached on the body segments to be tracked (for example, torso, shoulder, upper arm) are used to track joint angles and extract postural information in a 3D space which are presented through an animated display to the patient. The raw sensor data and angle information is also used to classify the exercises performed and extract information on the quality of the movements performed (duration, tremor, etc.). These information are used to provide a concurrent feedback to the patients through animations, verbal cues or on-screen text messages. A further comparison of the current performance and outcomes of previous sessions and expected performances serve as a visualization of training progress and enhances the patient’s motivation. The therapist exchanges data and provides information

to the patients on new guidelines through the patient station over a secure internet connection. The need for sensor calibration for postural tracking was a challenge, in addition to the economic viability.

A lightweight, easily programmable and transportable elbow rehabilitation device as shown in Figure 2.4 was built for patients suffering from elbow impairments. Continuous passive motion (CPM) machine is an effective technique to regain the patient's range of motion by moving the forearm about the elbow joint. It consists of a D.C. motor, gearbox, encoder, clutch and brake located in a portable unit, attached through a flexible shaft to an absolute encoder located on an elbow brace [80]. The device has adjustable settings for the entire rehabilitation stage. It is a smart wearable and portable device that uses constant sensor feedback enabling a progressive increase in user's range of motion (torque and motion limits). The device is capable of applying variable resistance about the elbow joint to build muscle mass and also acts as a brace to lock the arm movement in an unwanted direction. The motor-gearbox combination controls the range of motion by means of a current-limiter where the current measurement is converted to torque resistance in the computer and once the pre-programmed limit is exceeded the motor direction is reversed. The electrically controlled clutch acts as a safety feature for the patient. The device also employs a real-time monitoring facility whereby the user can view the real time plots of position and torque being applied to their joint during the exercise routine. The system tested on volunteers show a promising application to stroke rehabilitation at home.

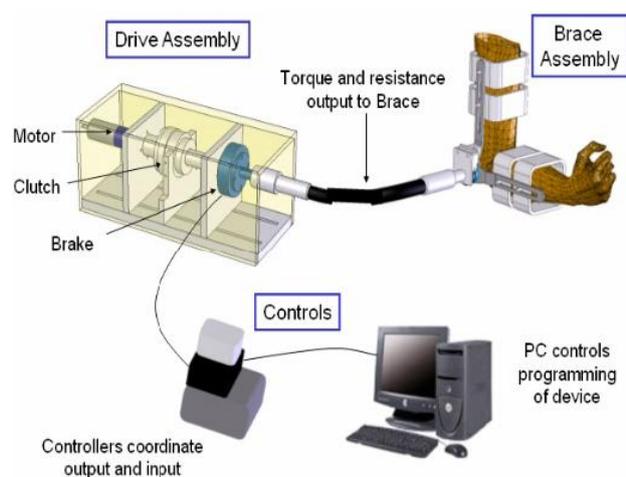


Figure 2.4: The Portable Continuous Passive Motion Elbow Device [80].

Pushing towards portability, a wearable skeletal frame called RUPERT (robot assisted upper extremity repetitive therapy), shown in Figure 2.5, was developed to provide a low cost and easy to use robotic device to assist upper arm rehabilitation at home or clinical

settings. It has four actuated degrees-of-freedom driven by pneumatic muscles (PMs) on the shoulder, elbow and wrist and programmed to actuate the device to move the arm in a 3-D work space without gravity compensation in view of its applicability during activities of daily living. The device interacts with a PC-based biofeedback system and provides the torque at the shoulder, elbow and wrist during reach and retrieve and feeding tasks. The signals from the pressure and angle sensors positioned on shoulder, elbow and wrist of the subject are fed into the computer running specialised software SIMM (software for interactive musculoskeletal modelling), to estimate the force needed to move the subjects arm and the skeletal frame and also kinematic information as joint angles thus providing real-time efficiency of the functional tasks performed. The device was evaluated with stroke patients for performing daily activities like reaching and feeding and the clinical outcome of their rehabilitation was assessed over duration of three weeks through a modified Wolf Motor Function Test (WMFT) and Fugl-Meyer test which showed satisfactory improvements. Further improvements in RUPERT are aimed at developing additional degrees of freedom for complex activities of daily living and use of an adaptive control to interact with users encouraging an active participation during their rehabilitation process [81]. Although effective for monitoring during exercises, both the CPM and the RUPERT are difficult for the subjects to wear on their own and carry for a long-time primarily due to their architecture.

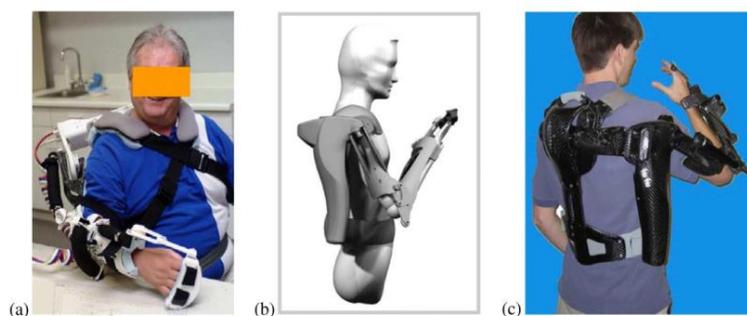


Figure 2.5: Development of RUPERT. Left: Initial design which has a robot arm attached to a chair. Middle: A computer generated picture of the new robot arm. Right: The new RUPERT on a subject's body [81].

### 2.2.2 Virtual reality based systems - virtual directional exercise platform

Virtual reality (VR) technology is being used in several areas of rehabilitation like Parkinson's disease for facilitating gait recovery, children suffering from cerebral palsy, amongst orthopaedic patients after hand or ankle surgery. Robot-assisted training methods have been effective in improving the motor functionality of the upper arm and hence have

been used in stroke rehabilitation. VR creates an environment wherein the intensity of the exercise schedule and corresponding feedback can be systematically modulated to suit the patient's need. This includes the use of a computer, internet connection, video-conferencing facility with the therapist, motivational games, and animations of the movements to provide visual feedback to the patients and the use of an avatar to guide the patient. VR helps the patients to improve their motor functionality through specific training schedule and apply it in the real-world conditions.

A virtual reality based tele-rehabilitation system was developed in [82] as shown in Figure 2.6, where the patient wearing two to three inertial sensors is exposed to a virtual environment (VE) having a motor training system and a videoconference facilities with the therapist. In contrast to the study by Philips Research [79], the sensors attached to the patient did not need calibration every time the system was turned on. The movements performed by the patient, as captured by the on-body sensors and camera systems, is interpreted by the VE system and displayed as an animation on the computer screen for the patient. The motion data is also transmitted to the therapist who can view the animations. This enables the therapists to analyze the movements and correspondingly provide feedback.

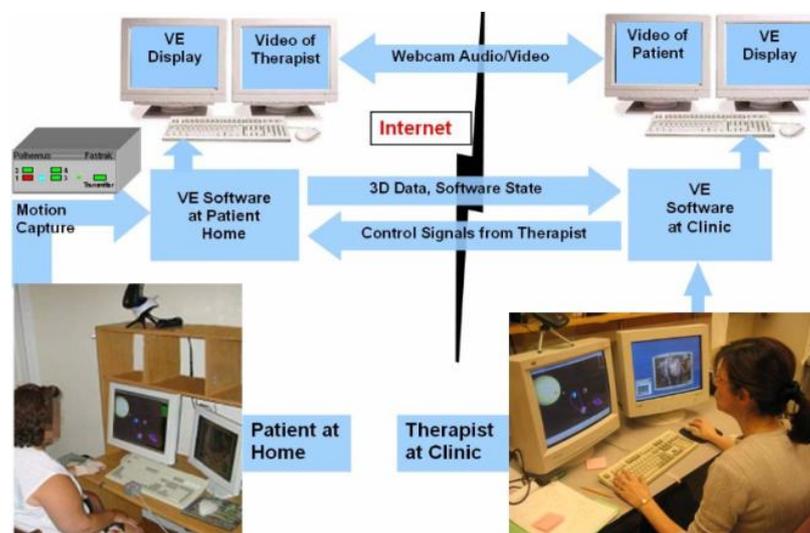


Figure 2.6: Summary of a telemedicine system using virtual environment for stroke rehabilitation [83].

The training methodology employs an avatar wherein the patient tries to follow the movements performed by the 'virtual teacher'; the therapist can control and modulate the movements of the avatar according to the need of the patients. A pilot study was conducted with 12 selected subjects with prior stroke (more than 6 months after recovery), where they performed various hand movements including reaching, grasping and finger movements

with the virtual system. Patients were also clinically evaluated (Fugl-Meyer Test, WMFT, Strength test) before, during and after the training schedule, which revealed significant improvements in their condition. They were able to generalize the motor training received during the VE training schedule to real world performance and even to tasks that were not specifically tested during the training schedule. The system was limited by the internet speed, continuous transmission of sensor data and the use of one camera system.

A study in [83] provides evidence of the effectiveness in transferring functional motion from VR to real-life environment by using cyber gloves to get measurements mainly aimed at testing finger motion and grasping abilities. The cyber glove is a sensitized structure worn on the hand with embedded strain-gauge sensors that measure the metacarpophalangeal (MCP) and proximal interphalangeal (PIP) joint angles of the thumb and fingers as well as finger abduction and wrist flexion. The glove is calibrated at the beginning of each trial. An exoskeletal device known as RMII glove having lightweight custom pneumatic actuators are attached to the tip of the thumb, index, middle and ring fingers aimed at applying force to the finger tips and use infrared sensors to measure the displacement of the fingertip with respect to the palm. Thus the cyber glove is used in the VR exercises that primarily involve position measurement of the patient's fingers and the RMII glove is used in force-exertion exercises. The VR simulation comprises of four exercises aimed at assessing the range of hand movement, the associated speed, finger fractionation and strength. During each trial, the exercise parameters are estimated on-line and a graphical model of the hand is shown to the patient, which is continuously updated to represent the flexion of the fingers and thumb. Patient trial was carried over for two weeks on 3 patients and their rehabilitation was clinically assessed using the Jebsen hand test and the Fugel-Meyer hand assessment, before, during and after the VR session. Objective measurements revealed the improvement of the subjects on most of the hand parameters over the course of the training.

### **2.2.3 Commercial gaming devices for tele-rehabilitation**

In the early 2000s, Sony released the PlayStation 2 EyeToy which was one of the first attempts to create a virtual environment for home gaming [84]. The cost-effectiveness of the device attracts researchers in the application of rehabilitation for both children and adults. It uses computer vision and gesture recognition to process images taken by the camera system allowing the players to interact with games using motion, colour detection

and sound through the microphone. Various clinical settings have used the Sony PlayStation 2 EyeToy for rehabilitation. It has been used successfully as an intervention tool for training the impaired arm during sub-acute rehabilitation after stroke in addition to conventional therapy. In [84] they use the PlayStation 2 with Wish-Washy and Kung-Foo games as the two exercises to simulate rehabilitation environment for stroke patients. Without any active support from the rehabilitation setup, participants having better motor functional ability benefited the most from the exercises while patients with acute stroke could not perform at the same level. The study conducted on twelve stroke patients provides preliminary results in applying game-based VR in rehabilitation. A technical shortcoming was the inability to record the user's performance to track rehabilitation progress. The Nintendo Wii gaming system has also been used in rehabilitation, where the game is controlled by the patient's movement [85].

On the other hand, the use of instrumental music supported therapy has been found to be effective in facilitating subject participation and adherence during exercise schedule as it reduces the perception of monotony, difficulty and discomforts [86]. Music supported therapy was combined with the Playstation3 gaming console and 5DT (five sensors) sensing gloves to build a home-based tele-rehabilitation system aimed at children and adults with chronic hemiplegia post-stroke. A pilot study was carried out on 3 children suffering from severe hemiplegic cerebral palsy aimed at rehabilitating their finger and grasping movements. The hand movements were displayed on a computer screen at the home station and audio-visual feedback was provided on the range and velocity of the motions performed. The computerized data from the sensor gloves generated during the game playing phase are stored in a clinical database to be observed by the respective clinicians. The therapists also upload periodic reports through the secure server for the patients to see their progress. The system also notifies the patient if there are any discrepancies during the training phase. The daily activities of the subjects for a period of 30 minutes were also noted down along with the gaming phase for a period of three months. All the children showed significant improvement in hand functionality thereby proving the effectiveness of remote monitoring. The technical shortcomings of the system such as the need to recalibrate the sensor glove at the beginning of every game consumed valuable time during rehabilitation [87].

### 2.2.4 Microsoft Kinect for rehabilitation

Visual marker based motion capture systems have been effectively used in 3D tracking of human upper limb but they require a complex and costly hardware setup and are intrusive in nature. This problem has been addressed by considering marker-less visual data using model and appearance based approaches. Model-based approaches are computationally intensive and their performance is dependent on the visual information extracted by the multi-camera systems. Appearance based systems, on the other hand, have low computational and hardware complexity but recognize a discrete set of hand movements. In recent times, Microsoft Kinect has been used intensively in hand/arm tracking as shown in Figure 2.7.

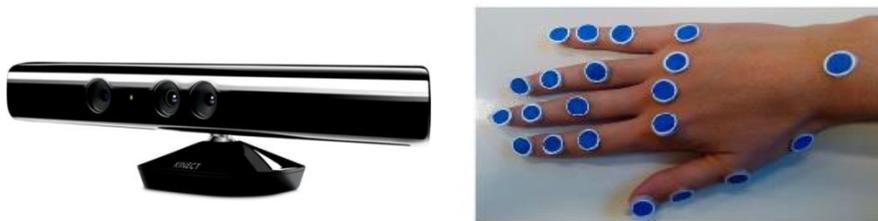


Figure 2.7: The Microsoft Kinect and the blue markers attached to a therapists' hand [20]

It provides an off the-shelf platform that can perform image processing, supporting easy to program languages such as C, C++ and C#. A 3D model of the hand comprising of the palm and five fingers was used in association with the Microsoft Kinect which accurately distinguished the hand articulations in real-time at 15 Hz [88]. The proximity of the hand to the Kinect system was investigated along with the effect of a noisy environment, both of which had an impact within a tolerable limit. In [20] the hand movements involved in daily living activities are performed in an off-line mode by the therapist and the various parameters related to the movement are tracked by the Kinect, based on which a model of the virtual hand is displayed to the patient using a Virtual Reality Toolbox. The movements performed by the virtual hand are imitated by the patient, which is consequently tracked by the Kinect camera and the difference between the prescribed and achieved mobility is displayed as a feedback. It is evaluated on two basic grasping activities with healthy subjects. Since skin tone and the distinction between the subjects' upper limbs with the surrounding environment pose difficulty in using the Kinect, small tracking markers are placed on the joint fingers as shown in Figure 2.7 which can help the tracker algorithm to extract movement data.

In another study [89], tracking information from the Kinect is obtained and compared with data from a more sophisticated acquisition system known as OptiTrack for six upper limb motor tasks that were incorporated into an application for game-based rehabilitation. The performance of the low-cost Kinect is comparable to the high-cost multi-camera lab-based OptiTrack system and hence can be conveniently used in VR based home rehabilitation setup. A study in [90] further uses a smart system having a Microsoft Kinect placed on a table and sensitized cutlery in order to monitor the patients' movement without any wearable sensors.

### **2.2.5 Monitoring activities of daily living (ADL)**

Rehabilitation is primarily carried out by repeated exercises of the impaired limb to maximize the chances of recovery [91]. However, it is well perceived in the medical community that exercises alone do not suffice for achieving a speedy recovery due to various factors. This is due to the lack of motivation among patients to exercise for sustainable period of time and the fact that exercises comprise of only a minor proportion of time and energy spent by a subject as compared to the wide range of activities performed throughout the day. Moreover, patients tend to compensate their paretic arm with their non-impaired arm making rehabilitation progress slower [92][93][94].

Hence, there has been a growing perception to monitor subjects as they perform their daily activities within their home and community settings. Quantifying the daily activities performed by patients would help to ascertain their degree of participation and thereby formulate a qualitative index of their lifestyle [95]. A taxonomy of activities known as Activities of Daily Living (ADLs) developed by [96] gained prominence in the research community owing to its relevance to real-world applications. Typical examples of ADLs include brushing teeth, combing, washing, cooking, bathing or walking [39], [97]–[99][100]. Accordingly, there have been extensive research efforts to assess the accuracy of wearable sensors in classifying ADLs [101]–[105] which has supported medical diagnosis during rehabilitation and augmented traditional medical methods in recovery of chronic impairments [2].

Human activity recognition (HAR) is a challenging and highly researched topic in many diverse fields which include pervasive and mobile computing [106][107], context-aware computing [108][109], remote health monitoring systems which also include ambient

assisted living [110]–[114]. Advances in wireless sensor technology have caused a paradigm shift from low-level data collection and transmission to high-level information integration, processing and activity recognition [115]. The different approaches are variants of the underlying sensor technology, the machine learning models and the environment in which the activities are performed. Before embarking on activity modelling and recognition methodologies it is imperative to understand the different levels of granularity inherent in human behavior.

### **2.2.5.1 Movement categories**

Depending on the complexity of activities performed, they can be categorized into mainly four different levels: gestures, actions, interactions and group activities. Gestures are movements performed by the subject's body parts which are atomic components comprising any holistic movement. Some common examples of gestures are raising the hand or stretching the leg. Actions are activities performed by individuals that are composed of multiple gestures aligned together to form a meaningful movement. For example, walking or reaching and picking a cup can be described as completed actions. Interactions are used to describe human-object or human-human interaction like making tea. Group activities, as the name suggests are performed by multiple persons, for example a group marching together [115][116].

### **2.2.5.2 Modalities of activity recognition**

Home-based activity recognition can be classified as: Vision-based and Sensor-based recognition.

Vision-based activity recognition uses visual sensing facilities such as video cameras and still cameras, to monitor a subject's movement in a designated area. The generated sensor data are video sequences or digitized visual data. Recognition of activities further takes place by the use of computer vision techniques which include feature extraction, structural modelling, movement segmentation, action extraction and movement tracking to analyze visual observations. The use of gaming consoles and the Microsoft Kinect having camera systems in the field of rehabilitation have been discussed in sections 2.2.3 and 2.2.4 respectively. However, they suffer from occlusion problems since they are designated to

mostly indoor activities with their surveillance restricted within a specific zone. Moreover, inferring the performed movements involve complex image processing algorithms [117].

Sensor-based activity recognition was further explored owing to a paradigm shift towards monitoring of activities in unconstrained daily life settings. The sensors used in activity recognition mainly generate time series data of various parameters or state changes. The data is processed through statistical analysis methods, probabilistic models, data fusion or formal knowledge technologies for recognizing the underlying activity. Sensors for activity recognition are generally attached on the body of the subjects as wearable sensors or in the smart phones. Sensors are also embedded within the living environment of the subject and thereby create ambient intelligent applications such as smart environments. For example, sensors attached to objects of daily use can record human-object interaction. Recognition methodologies utilizing multimodal miniaturized sensors present in the environment is termed as dense sensing approach. In this approach, activities are characterized by the objects that are manipulated during the performed movements in real-world settings. They are widely used in ambient assisted living (AAL) through the smart home paradigm [118], [119][120]. Sensors in smart homes are used to initiate a time-bound context-aware ADL assistance. For example, a pressure mat sensor can suggest position and movement of the subject within a defined environment and a switch sensor in bed can suggest sleeping activity of the subject [53]. In general, wearable sensor based monitoring is used in pervasive and mobile computing, while dense-sensing based approach is suitable for intelligent environment enabled applications. They can also work together like in RFID-based activity monitoring, where objects in an environment are instrumented with tags and users wear RFID reader fixed to a glove or a bracelet [114][121].

In the following section wearable sensor-based activity recognition has been covered in depth primarily using inertial sensors, which includes the scope, processing and challenges that are related to it.

### **2.2.5.3 Inertial sensor-based activity recognition**

Activity recognition using wearable inertial sensors primarily involves the capturing of kinematic signals which are used to measure acceleration, velocity, distance, rotation, rate of rotation, angle and time. These measurements help to determine position of a limb segment or the angle of flexion of a limb joint. These classes of signals are widely used as

health indicators encompassing gait, posture, spasticity, tremor, balance, some of which are subjective parameters in clinical assessment. Most of these parameters can be measured with the help of kinematic sensors like accelerometers, gyroscopes and magnetometers [22][122], thereby removing the subjective quotient from symptomatic data [123].

A MEMS accelerometer is probably the most frequently used wearable sensor used for activity recognition and measures the effect of acceleration rather than acceleration directly. Gyroscopes are used primarily for computing rate of rotation ( $^{\circ}/s$ ) and are also combined with accelerometers, known as inertial measurement unit (IMU) [124], where the gyroscopes help to compensate the accelerometers since the latter is orientation dependent. Therefore IMU's can be used to measure velocity, direction and gravitational forces [125]. A magnetometer is another device used in remote monitoring applications that responds to the strength and direction of the Earth's magnetic field. Considering, that the magnetic field vector has a constant direction and magnitude within a predefined area on the earth's surface (given by the latitude and longitude), a magnetometer can be used to track the orientation with respect to this constant field vector [125]. Magnetometers are not extensively used in health monitoring applications due to the fact that the Earth's magnetic field (which acts a reference) can be distorted by the presence of ferromagnetic materials [126]. Patients requiring wheelchair support (with steel frame) or the presence of other ferromagnetic substances within the home environment (e.g. in the kitchen) are likely to distort the reference magnetic field.

#### **2.2.5.4 Data-driven vs Knowledge-driven approach**

Processing of sensor data for recognizing activities can be categorized into two approaches – Data-driven and Knowledge-driven. Development of activity models is important for interpreting the sensor data to infer activities. In the Data-driven approach, sensor data collected as a result of the movements performed by the subjects are used to build activity models with the help of data mining techniques and relevant machine learning algorithms. Since this involves probabilistic or statistical methods of classification driven by the data, the process is generally referred to as Data-driven or bottom-up approach. Although this approach has its advantages in being robust to uncertainties and temporal variation in information but it requires the availability of a large dataset for training the activity model. Further it suffers from reusability and scalability as often it has been seen that activity models developed and evaluated on a particular subject's data does not work on the

movement data of another subject owing to the large degree of variability inherent in human movement [115].

The knowledge-driven approach on the other hand is used to exploit the rich prior domain knowledge to build upon an activity model. This involves knowledge acquisition, formal modelling and representation and is hence referred to as knowledge-driven or top-down approach. It is based upon the observation that most activities are performed in a relatively specific location, time and space. A suitable example would be an act of brushing teeth, which takes place in the morning, evening or at night and involves the use of a toothbrush. Similarly, cooking in the kitchen involves the use of the microwave or cutlery. This implicit relationship between activities, temporal and spatial context and the entities involved provides a rich domain knowledge and heuristics for activity modelling and pattern recognition [115]. This approach is semantically clear, logically simple but weak in cases of uncertainty and temporal information.

Having discussed the different modalities and approaches of human activity recognition, it is imperative to take an in-depth look into the process of activity modelling, classification and recognition using the Data-driven approach. Prior to this, it is worthwhile to look into the various challenges concerning activity recognition.

#### **2.2.5.5 Challenges in activity recognition**

Activity recognition presents more degrees of freedom with respect to system design and implementation as opposed to other fields in computer vision like natural language processing or speech recognition [127]. Owing to the diversity inherent in the same movement performed by different individuals, it requires a careful selection and placement of sensors, data analysis techniques depending on the application scenario and activities to be monitored [128]. These challenges are discussed at length in the following sections.

##### **2.2.5.5.1 Class variability**

A recognition system has to be robust enough to handle intra-class variability. In this context, class refers to the activities that are to be detected by any recognition methodology. This variability is primarily due to the fact that a same activity is performed

differently by different individuals. Further this might also happen with an individual who repeats the same activity over time due to factors as fatigue or environmental changes. Therefore there can be two approaches towards training an activity recognition system.

A system trained with movement data of more than one subject, or a person-independent training system would be susceptible to considerable inter-person variability [128]. To address this issue, the number of data points for each subject can be increased or an alternative approach can be person-dependent training, i.e. training the system on the movement data of single person. This might as well be robust enough towards capturing considerable intra-person variability. This system however requires the collection of a large set of data collected from one individual to train the system thereby capturing as much variability as possible. The choice of the training sample is application dependent and hence a trade-off is required between the selection of a highly specific and discriminative dataset or a generic dataset which is potentially less discriminative but robust across multiple subjects [127]. In general, for remote health applications, formulating a person-centric training data would be beneficial when applied to monitoring of individual patients who demonstrate differences in levels of impairment depending on their stage of rehabilitation [129].

Another interesting challenge in recognizing activities is the similarity in characteristics prevalent across activities [130]. For example, it would be very difficult to distinguish between drinking water from a glass and drinking coffee from a cup based on the kinematic data collected from a wrist-worn sensor pertaining to both the movements. Therefore, in such cases sensors deployed in the environment like RFID tags attached to objects, can prove to be helpful. Therefore this is dictated by the requirements of the application [121].

An intriguing problem occurs during activity recognition on continuous streaming data or real-time monitoring applications where the data needs to be segmented depending on the activities which are to be monitored and those that are irrelevant to the application. This is referred to as the NULL class in relevant literature [131] and is difficult to model since it represents a plethora of activities in infinite space. It can however be identified if the signal characteristics gathered from the sensor data is completely different to the ones that are being monitored and hence involves a threshold-based mechanism to filter out the unwanted data.

Class imbalance is a major problem especially during long-term monitoring where all activities being tracked do not have the similar number of occurrences. A common example would be the number of instances of a drinking action and a walking action [132]. There are however a couple of techniques which can be adapted to get around this problem of class imbalance. Firstly, generating artificial training data for the class which is underrepresented to balance out the inequality and secondly, oversampling or interpolating the smaller class size to match it to the bigger class size [133].

#### **2.2.5.5.2 Ground truth annotation**

Annotating the ground truth of activities being monitored in real life scenarios is another interesting challenge especially with data from wearable inertial sensors as opposed to data obtained from video recordings. With activities performed in the laboratory or controlled environment, annotations of the training data can be performed post-hoc based on video footages. However, in nomadic settings, ground truth annotation of activities is a difficult problem. Researchers generally depend on self-recalling methods [134], experience sampling [135] and reinforcement learning all of which involves testimonies from the subject themselves. Therefore many researchers have based their work on a list of activities performed under a semi-naturalistic condition, where the subjects perform the movements as they would do in normal daily life and another person annotates their activities by means of visual inspection in real time [39]. This therefore helps in gaining the ground truth information required for evaluating the recognition methodology.

#### **2.2.5.5.3 Sensor requirements**

The experimental design gives rise to another challenge that of data collection, sensor selection, placement and the number of sensors to be used. As opposed to other computer vision problems like heart monitoring, brain activity modelling or speech recognition, human activity recognition does not have a standard allocated dataset to start with the data analysis as it is completely dependent on the requirement and experiments are designed in pursuit of recognizing only the selected movements. Sensor characteristics also present a significant amount of challenge for long-term monitoring of activities as hardware failures, sensor drifts and errors in the software aimed at capturing the data can lead to erroneous situations. External factors such as temperature, pressure and change in positioning due to

loose straps can cause the need for frequent recalibration thereby affecting the sensor data being recorded [136], [137].

One of the last challenges is power requirement of the battery operated wireless sensors which are increasingly being used in the field of remote health monitoring. The remote monitoring system in place at present transmits the captured signals collected by the sensor nodes placed on the patient's body to the remote server at the back office service platform wirelessly, where the signals are analysed [33]. This system requires continuous transmission of data from the sensors to the server using wireless protocols taking into account the nomadic environment. The fundamental problem with continuous data transmission is the energy requirement. A result from respective investigations into continuous data transmission at 1 kHz suggests that it can be supported for 24 hr. monitoring using a 1200 mAh battery [138].

The analysis presented in [138] regarding the power consumption and longevity of batteries pertains to transmission energy only, added to it the energy involved in pre-processing the physiological data at the sensor nodes including analog to digital conversion, quantization, filtering and the microcontroller operation, would bring down the effective time of monitoring to 8 - 10 hours, thereby making the entire system power hungry and affecting the life of the batteries. An increased battery capacity like the prismatic zinc-air battery – 1800 mAh operating at 1.4 V used recently in the medical community, would increase the respective sizes of the sensor nodes. Furthermore, the use of Bluetooth transceivers consuming 40-55 mA with operating voltage in the range of 3-3.6 V would necessitate the use of three such zinc-air batteries, making it non-ideal in terms of volume for body-worn applications.

The supply voltage is quadratically proportional to the power dissipation and therefore an optimal power supply to sustain the continuous Bluetooth transmission would have an adverse impact on the operational life-time of the battery powered sensor nodes. Considering Bluetooth as the primary means of communication, the energy dissipation is directly dependant on the packet format of the data being transmitted which can be optimized using standard duty cycling and might eventually lead to delays and packet loss of data which would be highly undesirable for applications involving remote health monitoring [33].

Therefore, from the long-term system operation perspective, when implementing a wireless body area network (WBAN) comprising of heterogeneous sensors, it is imperative to select data analysis algorithms having low-computational complexity. This is because energy consumption is directly proportional to the computational complexity of the processing algorithms used. Therefore, for applications such as real-time movement detection requiring online operation it is imperative to perform the data processing (feature extraction, classification, etc.,) in a low-power way on the sensor platform [33], [128] itself whereas for applications supporting long-term behavioural or trend analysis offline data processing may be sufficient [139].

### 2.2.5.6 Activity recognition – process flow

In this section, a detailed description about the sequence of signal processing and pattern recognition techniques that helps to implement a specific activity recognition behaviour using supervised learning methodologies have been presented. The process flow is depicted in Figure 2.8.

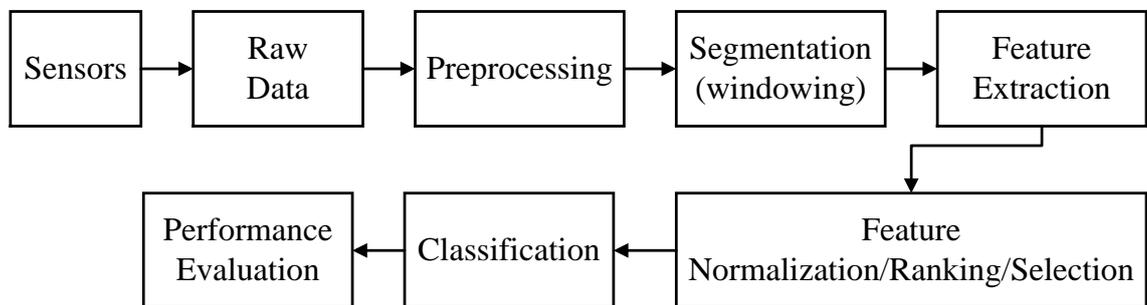


Figure 2.8: Activity Recognition Process Flow.

#### 2.2.5.6.1 Data acquisition and pre-processing

Raw data is collected from multiple sensors attached to the body or sensors placed on objects or in the environment or from both depending on the application requirement as discussed in section 2.2.5.2. Data coming from each sensor sampled at regular intervals results in a multivariate time series. However, the sampling rates of different types of sensors might differ. Therefore there is a need to synchronise multimodal sensor data. Inertial sensor data are generally sampled at low frequencies, 20 – 30 Hz depending on the movements to be monitored. Certain sensors like accelerometers, gyroscopes and

magnetometers can produce data that have multiple dimensions (X, Y and Z axis). Therefore, a multi-dimensional and a multimodal sensor output can be represented by (2.1) where  $S$  represents the sensor output,  $m$  represents the number of sensors and  $d_1...d_n$  represents the sensor-specific data sampled at regular intervals.

$$S_i = [d_1, d_2 \dots d_n], \quad i = 1, \dots, m \quad (2.1)$$

The raw sensor data contains noise and is often corrupted by artifacts caused due to various factors. Artifacts are generally induced into the data due to sensor malfunctioning (e.g. drift) or due to unwanted movements of the body [127]. The pre-processing stage aims to remove the low-frequency artifacts and high-frequency noise components by using high pass and low pass filters [128]. The pre-processing algorithms also synchronise the data coming from various sensors and prepares the data for the next stage of feature extraction. It preserves the signal characteristics which carries the relevant information about the activities of interest. Data from inertial sensors are in general calibrated, their units are converted (as most sensor outputs are arbitrary units), normalized, resampled, synchronised, filtered or fused in the pre-processing stage. Sensor fusion is generally performed where signals from multiple sensor axes are selected a priori, based on the activities being tracked. For example, specific accelerometer and gyroscope axes can be fused (both sensors placed on the wrist) for detecting a reach and retrieve action [128].

#### 2.2.5.6.2 Data segmentation

The pre-processed signal is segmented to identify only those segments that contain information about the activities that are being monitored. This process is also commonly referred to as event detection or activity spotting since it detects the signal frame representative of the activity of interest. The boundary of each segregated time series data is represented by the start and stop time. Thus each segmented time series represents the potential activities to be monitored. As mentioned in section 2.2.5.5.1, segmenting a continuous streaming data is a difficult task especially for monitoring ADL. For example, for a drinking activity, it may start with reaching for the cup or glass, drinking and then putting the cup back on the table or keeping it in the hand. In such circumstances, it is difficult to determine the boundaries of the activity from the signal. There are various segmentation algorithms that are used in relevant research, popular among them being the

sliding window technique, energy-based segmentation, rest-position segmentation and using data from one sensor to segment another sensor reading [127].

The sliding window technique is one of the most popular segmenting schemes followed in diverse applications. As is suggested by the name, a fixed-size window representing definite time duration is used to extract segments of a signal [2]. If a very small interval is chosen, there is a possibility of missing out on a relevant activity whereas a longer window size would pertain to multiple activities, thereby affecting the classification decision. Hence, a dynamic window selection technique based on a data-driven or a probabilistic approach for segmenting each individual activity would be an optimal solution although this increases the computational load [115].

Another popular approach adopted for segmenting different activities is based on the energy content of the signal reflecting the change in intensity levels. The differences in energy levels in the signal are representative of the intensity variations of the activities that produce these kinematic signals. The energy content of a signal  $s(t)$  is given by (2.2).

$$E = \int_{-\infty}^{\infty} |s(t)|^2 dt \quad (2.2)$$

Therefore, a threshold-based mechanism based on the value of  $E$  can help to identify segments of activities which are identical [140]. Researchers have explored energy-based segmentation with the assumption of a rest period between each activity which is particularly useful for gesture recognition involving discrete activities and momentary pauses [141].

#### 2.2.5.6.3 Feature extraction

The choice of features is a fundamental step for classification and a highly problem-dependent task. Although each of the sensors exhibits signal patterns that are distinctive for each of the movements and may be recognizable to the human eye as shown in Figure 2.9, in order for a machine to recognize these patterns a set of characterizing features need to be extracted from the signals. Features represent the transformation of the raw sensor data into another space known as the feature space where ideally, identical activities should be clustered together. It is a measure to characterize the raw data in a quantitative as well as a qualitative manner. Typical feature sets for human activity recognition include statistical functions, time and/or frequency domain features, as well as heuristic features [10].

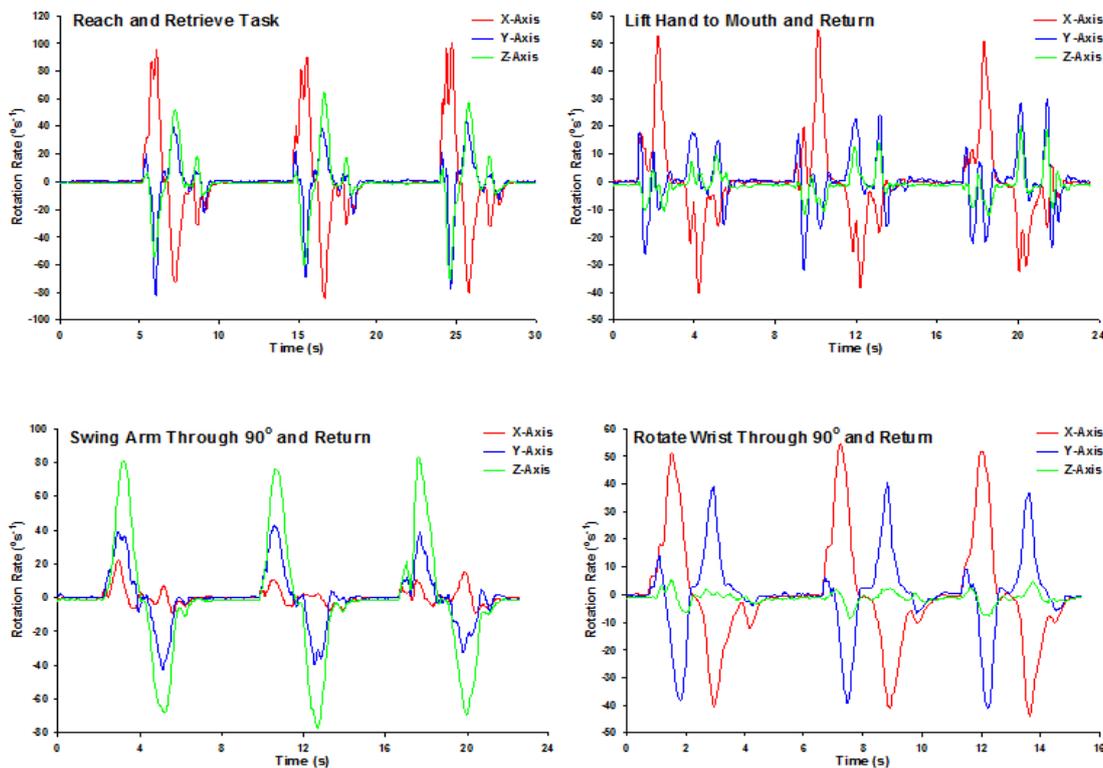


Figure 2.9: Signal patterns generated by a tri-axial gyroscope placed near the elbow for three repetitions of four different arm movements – reach & retrieve; lift hand; swinging arm in the horizontal plane; rotate wrist.

Some of the commonly used time-domain features extracted from sensor data, used in literature are: mean, variance, median, skew, kurtosis, inter-quartile range, root mean square, standard deviation, correlation between axes [28], [142]. Correlation between accelerometer axes can improve recognition of activities involving movements of multiple body parts. For example, walking and climbing stairs might have same periodicity and magnitude of the acceleration signal measured from the limb but walking involves translation in one dimension as compared to climbing stairs which involves translation in multiple dimensions [143]. Similarly, the features variance, inter-quartile range, root mean square and standard deviation are a representative measure of the magnitude of the varying quantity in the acceleration data.

The commonly used frequency domain features which are extracted from the coefficients of various time-frequency transforms like Short Time Fourier transform (STFT), Fast Fourier transform (FFT) and the Continuous or Discrete Wavelet transform (WT). The high frequency components of an accelerometer signal also known as the AC component is primarily related to the dynamic motion of the subject like walking, running, hand shaking while the low-frequency DC component of the signal is related to the gravitational acceleration and hence is informative about the orientation of the body in space and can be used to classify static postural positions [144]. The signal energy and its distribution at

different frequency bands are popular choices for discriminating activities of differing intensities. More specifically some of the commonly used features are spectral centroid, spectral spread, estimation of frequency peak, estimation of the power of the frequency peak and signal power in different frequency bands [28], [144]. Frequency domain entropy calculated as normalized information entropy of the discrete FFT component magnitudes of the signal helps to discriminate activities with similar energy content. For example, biking and running might result in same amounts of energy if captured with an accelerometer placed near the hip. Biking involves a uniform circular motion and discrete FFT of the acceleration data in the vertical direction may show a single dominant frequency component at 1 Hz and low magnitude for all other frequencies. Running on the other hand may have many major FFT components between 0.5 to 2 Hz (considering a sampling frequency of 76.25 Hz) [39].

#### 2.2.5.6.4 Feature selection

For real-time activity recognition, it is imperative to use the minimum number of features with an eye on computational complexity and memory utilization. However, before proceeding with feature selection, the feature vectors need to be pre-processed to remove the outlier points and normalize the features. An outlier is a point that appears as a result of noisy measurement and lies far away from the mean of the corresponding feature vector causing large errors during the training of the classifier. For normally distributed data, a threshold of up to three standard deviations from the mean is used to filter out the outliers. For non-normal distributions, more complex measures like cost functions are considered [145]. Feature normalization is another key step adopted for feature values lying in different numeric ranges, such that features with large values do not dominate the cost function in the design of the classifier. A common technique is linear normalization as shown in (2.3), where the features are normalized by removing the mean from each sample and dividing the samples by their standard deviation. This ensures that each feature has zero mean and unit variance and can be represented as:

$$\tilde{x}_i = \frac{x_i - \mu}{\sigma}, \quad i = 1, 2, 3, \dots, N \quad (2.3)$$

where  $x_i$  represents the respective feature values,  $\mu$  is the mean value,  $\sigma$  is the standard deviation and  $\tilde{x}_i$  represents the normalized feature values. Alternatively, other linear

techniques can be used to normalise the feature values by restricting them between a minimum and a maximum value as expressed in (2.4). Selecting the range depends on the nature of the data.

$$\tilde{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \quad i = 1, 2, 3, \dots, N \quad (2.4)$$

Non-linear methods of normalization are also applied for data which are not evenly distributed about their mean. In such circumstances non-linear functions like logarithmic or sigmoid can be used to transform the feature values within specific intervals [145].

The normalization step is followed by the feature ranking step. Fisher Discriminant ratio (FDR) and Bhattacharya distance are such techniques used to quantify the discriminatory ability of each individual feature between two equi-probable classes. Another class separability technique, based on scatter matrices, can be used for a multiple-class scenario [145]. The rank of each individual feature is determined, where a high rank represents a small within-class variance and a large between-class distance among the data points in the respective feature space [146].

Now, the core problem area is to select a subset of  $l$  features from the best ranked  $m$  features (where  $l \leq m$ ). The two major approaches are – scalar feature selection and feature vector selection. In the scalar feature selection technique, each feature is treated individually and their class separability measure is ascertained using any of the above mentioned criterion  $c(k)$  (FDR, Scatter Matrices etc.) for each feature. Features are then ranked in a descending order according to the criterion  $c(k)$  and  $l$  best features are considered for classification.

Considering features individually involves low computational complexity but is not effective for complex classification problems and for cases where features are mutually correlated. Feature vector selection can be approached in two ways:

(1) **Filter approach**: The features are selected independent of any classification technique. For each combination of features chosen, the class separability criterion as mentioned above is used and the best feature combination is selected. Considering  $m = 10$  and  $l = 5$ , can lead to 252 feature vector combinations which is very large and the number  $l$  is also not known *a priori*.

(2) **Wrapper approach:** The selection of the feature is based in association with the classifier to be employed. For each chosen feature vector combination the classification error probability of the classifier is estimated and the feature combination with the minimum error is chosen. This approach can be further computationally complex depending on the choice of the classifier. However, to reduce the complexity there are some effective searching techniques, which have been proposed to select the best feature vector combination: sequential backward selection, sequential forward selection and floating search method [145].

In relevant literature, the principal component analysis (PCA) or the Karhunen-Loeve transform, which transforms feature vectors into a smaller number of uncorrelated variables called the principle components, is a very popular technique [143][144]. Another popular approach is the Independent component analysis (ICA) applied in problems of blind source separation which mainly attempts to decompose a multivariate signal into statistically independent non-Gaussian signals [147]. The choice of relevant features, the ranking or selection technique is completely dependent on the activities, choice of sensors and the application scenario. The domain of classification which is the next most important step after feature selection is discussed in the following section.

#### 2.2.5.6.5 Classification

A wide range of classifiers have been used for activity recognition in recent years [148]. The determining factors for the selection of the classifier are accuracy, ease of development and speed of real time execution [127]. Two distinct approaches can be used in classifying human activities – supervised and unsupervised learning. In supervised learning, the association of the training dataset comprising of selected feature vectors with each class label is known beforehand [144]. In unsupervised learning, only the number of classes is known and the system assigns a class label to each instance in the training dataset. Clustering based unsupervised learning has been used in the field of activity recognition [146][149]. In human activity recognition the classification schemes used can be broadly categorized into three themes: probabilistic models, discriminative approach and template-based similarity metrics.

(1) **Probabilistic models:** Probabilistic models are quite commonly used for behaviour modelling since they are an efficient means of representing random variables, dependence

and temporal variation. In this approach, the activity samples are modelled using Gaussian mixture, yielding promising results for offline learning when a large amount of data is available for training. Generative probabilistic models such as HMMs have been used to model activity sequences and have been extended to hierarchical models like Conditional Random Fields (CRFs) and Dynamic Bayesian Networks (DBNs) [149]. Hidden Markov Models (HMM) have been very popular in speech recognition and have also been used in applications for hand gesture recognition [150], [151]. In general, the HMM is trained on pre-defined class labels using the Baum-Welch algorithm and is tested on new instances. The Baum-Welch algorithm is a generalized Expectation Maximization (EM) algorithm that computes the maximum likelihood estimates of the parameters of an HMM given the observations as training data [152], [153]. The problem with HMMs is the first-order Markov assumption where the current state depends only on the previous one. Further the probability of a change in the hidden state does not depend upon the time that has elapsed since entering into the current state. Therefore, a time-dependence has been added to HMMs and they have been augmented to semi-HMMs where the hidden process is semi-Markovian rather than Markovian. Coupled HMMs have also gained prominence which is considered as a collection of HMMs, where the state at time  $t$  for each HMM is conditioned by the states at time  $t-1$  of all HMMs in the collection. They are used to model the dynamic relationships between several signals [154].

(2) **Discriminative approach:** The classification is based on the construction of the decision boundaries in the feature space, specifying regions for each class. The decision boundaries are constructed on the feature vectors of the training set, through an iterative or a geometric consideration. The Artificial Neural Network (ANN) commonly used for detecting ADL, consists of inputs and outputs with a processing or a hidden layer in between. The inputs are the independent variable and the outputs represent the dependent variable. The internal (hidden) layers can be adjusted through optimization algorithms such as the resilient back-propagation or scale-conjugate algorithms [104].

The  $k$ -Nearest Neighbor ( $k$ -NN) [155] and the Nearest Mean (NM) classifiers work directly on the geometrical distances between feature vectors from different classes [156]. Support Vector machines (SVM) work by constructing boundaries that maximize the margins between the nearest features relative to two distinct classes. SVM is a very popular technique in machine learning community and generally produces high accuracy rates with moderate computational complexity (depending on the number of support vectors used)

[2], [157]. In principle, it is a binary classifier but has been extended to handle multiple classes using the ‘one versus all’ or the ‘one versus one’ scheme [158]. However both of these methods can be computationally intensive depending on the number of target classes. The Naive-Bayes classifier has also been successfully used over the years [113], [142]. It assumes conditional independence among all feature vectors given a class label and learns about the conditional probability of each feature. They require large amounts of data and do not explicitly model any temporal information which is very important in activity recognition [115]. Finally, binary tree classifiers have been widely popular in the field of human activity recognition, where the classification process is articulated in several different steps. At each step, a binary decision is made based on different strategies like the threshold-based or template-matching. With each stage, the classification is progressively refined as the tree descends along the branches [159]. The *C4.5* Decision Tree (DT) algorithm has been successfully used to recognise daily living activities [28], [39].

(3) **Template-based similarity metrics:** The template matching technique exploits the similarity between the observed data (testing dataset) and the pre-stored activity templates which is user-defined or obtained from the training dataset. They can employ for example, a  $k$ -NN classifier using the Euclidean distance computed between the testing and training dataset having a fixed window size or dynamic time warping [160], [161] in the case of varying window size. Another popular template matching technique used is string matching [162]. The choice of a classifier depends on the trade-off between the computational complexity, memory requirements and the recognition accuracy.

Lastly, one of techniques quite common in classification problems is cross-validation. During estimation of the model parameters, validation of the employed classification algorithm is essential to judge its robustness. This is particularly applicable in supervised learning techniques, i.e. where the class labels of the dataset is known. The available cohort of data is divided into independent training and testing datasets. The training set is used to train the classifier whereas the test set is used to estimate the error rate of the trained classifier. The cross-validation technique, also termed as the resampling method can be divided into three approaches: random sampling,  $k$ -fold Cross-Validation, Leave-one-out Cross-validation [127], [163]. The number of folds is determined by the size of the dataset. A large number of folds on a sparse dataset leads to a more robust classifier.

### 2.2.5.6.6 Performance evaluation

The recognition performance of an activity recognition system can be evaluated in terms of correct classification through True positives (TPs) or False Negatives (FNs). Classification can also lead to false detection of activities that did not occur and can be estimated through False Negatives (FNs) and false positives (FPs). Besides there are few well known performance metrics that are widely used in the research community like confusion matrices, accuracy, precision, recall or Receiver Operating Characteristics (ROC) curves.

A confusion matrix is a popular means of evaluating the classifier performance for multi-class problems. It summarizes the misclassifications of the different activity classes. The rows of the matrix represent the actual number of instances in each activity classes while the columns represent the predicted instances in each activity classes. Precision  $\left(\frac{TP}{TP + FP}\right)$ , recall  $\left(\frac{TP}{TP + FN}\right)$  and the overall accuracy  $\left(\frac{TP + TN}{all}\right)$  can be easily computed for each activity class from the matrix. Normalized confusion matrices are commonly used for unbalanced datasets where there is a significant amount of difference in the number of ground truth annotations of the activity classes [127], [164], [165].

### 2.2.5.6.7 State-of-the-art activity recognition systems

In Table 2.1, a few state-of-the-art activity recognition systems have been listed which have been developed in recent years mainly aimed towards ADL monitoring. Here, the monitored activities, the sensors and their positioning, the classification schemes used and the accuracies obtained have been highlighted. The review of the existing modalities reveals that majority of the published work on activity recognition has been devoted towards monitoring of gross dynamic human movements of such as, sleeping, sitting, standing, cycling, running etc. Apart from them, efforts have also been made to ascertain the activity levels of subjects in daily activities like bathing, combing, toileting, drinking, etc. Monitoring of elementary arm movements or gesture recognition has also been prevalent in the research community aimed at monitoring of specific tasks/movements in rehabilitation based applications, specific hand gestures for human-computer interaction or the monitoring of dietary intake for nutrition monitoring applications. Sensors have been placed on the objects being manipulated [166] or embedded in special gloves [167].

Ref.	Sensors	Activities	Evaluation
[2]	Wrist-watch based sensor containing accelerometer, altimeter, and temperature sensors.	BADLs - Brushing, dressing, walking upstairs/downstairs and sleeping; IADLs – washing dishes, ironing, watching television.	Neural Networks and SVM; average accuracy of 90%; dressing, ironing, brushing and washing are sometimes confused; system is cost effective and non-intrusive.
[38]	Two tri-axial accelerometers in the smartphone.	ADLs such as sitting down, standing up, walking and stopping	SVM; Accuracy of 92 – 100 %; limited by the number of activities and the charge on the smartphone.
[39]	Five bi-axial accelerometers	20 different activities including walking, sitting, standing, lying down, climbing stairs, folding laundry, etc.	Decision Tree; Accuracy of 44 – 94%; riding elevator and stretching confused with other activities; large number of sensors; wired connectivity and need for real time synchronization are the pitfalls.
[117]	Inertial sensor on the thigh for activity recognition; marker based system using twelve cameras to track the 3-dimensional location.	Lying, sitting, standing, walking and transitions from sit-to-stand, lie-to-sit, stand-to-sit and sit-to-lie.	HMM; overall accuracy about 90%; accuracy increases with location data; high set up cost and a fixed coverage area for location tracking using the camera system.
[168]	Accelerometer on thigh and waist; RFID reader on hand glove; RFID tag on objects of daily use	18 activities - hand shaking, rope jumping, brushing, making phone calls, reading, using umbrella and other ADLs	Decision Trees; overall accuracy of 95%; walking was detected by 84% - confused with standing/running. The glove 'iGrabber' is too big to wear for elderly people and tagging of different objects with RFID is exhaustive.
[169]	Infrared sensors to detect location and movement; door contacts; microphones; digital temperature and hygrometry sensor; tri-axial accelerometer and magnetometer	Bathing, dressing, resting, use of toilet, moving in/out of bed/chair, feeding.	SVM; Accuracy of 90% except for dressing (75%) and hygiene (64%) which had fewer occurrences in the training database; use of multiple ambient sensors is expensive; needs to be tested on the elderly population.
[170]	Four accelerometers on the thigh, sternum and lower arm.	Posture of human body, lying, sitting, standing, walking, climbing stairs, cycling, driving, use of wheel chair and running.	Average accuracy of 90%, uses wired sensors and hence is intrusive.
[171]	Tri-axial accelerometer placed on the wrist.	8 WMFT – reach and retrieve tasks, card flipping, grasping etc.	Dynamic Time Warping; average accuracy of 96.5%, the system is small and non-intrusive and uses only one sensor which is advantageous.

Table 2.1: State-of-the-art systems developed in recent years for ADL monitoring.

In addition to basic movements such as sitting, standing and walking, activities involving the upper limb such as keyboard typing, writing on a white-board and handshaking were looked in [172]. Using 12 sensor nodes over the body, the activities were detected successfully (50-90%) using a combination of sensors placed on the wrist, elbow and shoulder in conjunction with a Bayesian classifier. Tool-based workshop activities such as sawing, hammering, drilling, and filing were recognised in [173]. The incoming data from inertial sensors were partitioned using an intensity-analysis on the signals from two microphones based on the fact that each relevant action in the workshop was accompanied by a particular hand tool sound. The focus was to track the progress and sequence of the work in workshop. Hence, single Gaussian HMMs were successfully used for modelling and gestures were detected with an overall accuracy of 84%.

Gestures involved frequently in human feeding motion have been looked in [174] for determining the timing of nutrition intake. This work takes a two-fold approach - first detecting object-interaction gestures, i.e. soup intake involves a spoon, drinking involves a glass/cup/bottle and second, focusing on dietary intake gestures. HMM's were used to model the spatial temporal variations in the input data acquired through five inertial sensors placed on the wrist, upper arm and torso from four young subjects. This study aimed at spotting of sporadically occurring gestures from continuous data stream and highlighted the inter-person variability of the gestures. The classification results suggest an average of 70-80% precision across the gestures monitored. An attempt to classify 10 arm activities as part of assembly-line workers in a car production environment were made in [175] using meta-classifier that fuses information of classifiers operating on individual sensors. Data was collected from 19 sensor nodes distributed over the two arms. Here, HMM's which are a common approach to handle temporal variations in gestures, along with the naïve Bayes classifier have been used. Recognition up to 80% was achieved with data from a single inertial sensor node. Sequence of composite activities constituting of a finite number of atomic or elementary activities involving the arm, have been recognised using a two-layered abstraction model in [176]. In the lower layer, simple atomic activities are recognised using multiple on-body and environmental sensors and these activities are grouped depending on the location of the sensors to infer composite activities. This method was evaluated on a car assembly scenario in the laboratory environment and an overall recall and precision of 77% and 79% was achieved for 11 different composite activities. In all the three works mentioned, the focus was on the detection of the sequence of gestures involved in completion of the designated tasks (feeding activity, workshop activity and assembly line work) and hence HMMs have been effectively used to model the state transitions representing the activities.

Other classifiers such as decision tree, k-NN have also been successfully used for recognising gestures such as drinking water, handling mouse, opening drawer, typing and writing for online recognition on a wrist-worn sensor node [55]. The algorithms were run on a TinyOS platform and implemented on a MSP430 microcontroller on-board the sensor module with a recognition accuracy of 86%. Data collected in office work scenario was used for both the training and testing purposes. Accelerometer based gesture recognition using Continuous Time Recurrent neural Networks (CTRNN) has been performed in [177]. This method has the advantage of operating on the raw data directly rather than using features as used in other classification methods. Eight gestures such as sitting, standing,

reading books, opening drawers, etc., performed in a unconstrained environment were classified with 64% accuracy. An SVM based activity recognition system using objects attached with sensors to recognise drinking, phone use and writing activities was introduced in [178], which achieved a performance of 72%, 84% and 80% respectively for each activity.

In accordance to the research focus, there is a need to explore a methodology for detecting the occurrence of elementary arm movements performed in nomadic settings which is relevant towards rehabilitation for stroke patients who are supposed to exhibit a higher degree of variability within their movement profile as compared to healthy subjects. In relevant literature, the critical aspects like optimal sensor selection and placement, selection of a subject-dependent/independent database and development of associated low-complexity data processing and classification techniques for implementation on a low-power hardware platform did not take priority. This necessitates a complete exploration to achieve the research objectives.

### **2.3 Discussion**

This chapter presents a detailed review of the systems used for physical activity monitoring during rehabilitation training and ADL. Gaming consoles and other camera-based systems can be effective to track rehabilitation of subjects under controlled environments. Given the research focus on upper limb rehabilitation in nomadic settings, in this chapter particular focuses on the systems aimed at ADL monitoring. Objective measures of a subject's participation in daily living activities can be obtained by using wearable sensors (e.g. inertial sensors). But there is a need to limit the number of patches (i.e. sensor nodes) used on a patient's body and the wearable sensors need to be pervasive in nature for long-time use from the convenience perspective of patients. For real-time activity recognition applications the processing of the data takes place on board the sensors and hence it is imperative to select data analysis algorithms that involve low-computational complexity. The essential steps of feature extraction, selection, classification and cross-validation will heavily depend on the system requirements covering important areas – type of activities, number of activities, type of sensors, number of sensors, placement of sensors, multiple sensor fusion, etc. This paves the way for a thorough exploration in recognizing upper limb movements performed in a nomadic environment, considering a realistic implementation scenario which can be translated to actual practice.



# **3. Chapter 3**

## **Movement Selection and Sensor Calibration**

### **3.1 Introduction**

As mentioned in Chapter 1, the primary aim of this research work is to develop low-complexity algorithmic techniques for accurately detecting the impaired upper limb movements of stroke survivors in nomadic settings. The developed algorithm will be translated to dedicated low-power hardware which can be used within a resource constrained framework of a body-worn sensor node aimed at detecting arm movements in real-time for remote monitoring of stroke survivors. Developing a robust arm movement recognition algorithm for nomadic settings on the backdrop of a high degree of variability inherent within the human movement is particularly challenging. The exploration of an accurate algorithmic technique depends on two key factors: 1) the specific arm movements which are to be tracked/detected and 2) the optimal number and placement of appropriate inertial sensors on the body. The latter point necessitates a study on the sensor characteristics along with their calibration. Therefore, in this chapter the focus is on: 1) the selection of specific upper limb movements to be tracked, 2) calibration of inertial sensors, more specifically accelerometers and gyroscopes and 3) estimating the sensor characteristics - noise margin and drift inherent within the sensors under static conditions. This will help to lay the foundation for using these sensors for collecting data during the experiments to be conducted within this research work, helping in the target algorithm development for movement recognition.

### **3.2 Movement selection**

The wrist, elbow and shoulder joints along with the fingers play an important part in completing the majority of upper limb movements performed in daily life. Clinical treatment of the paretic arm post-stroke, involves rehabilitation of these specific

joints/parts by repeated exercises to help survivors regain their motor functionalities. As mentioned in Chapter 2, there are a wide range of clinical tests that are quite popular and performed by the therapists to assess arm rehabilitation of survivors within a clinical environment. The Wolf motor function test (WMFT) is one such standard clinical test used to examine the effects of constraint-induced therapy (CIMT) for stroke survivors and was later modified to be sensitive to the level of motor functioning characteristic of patients with mild to moderate stroke [5]. It comprises seventeen motor tasks that are primarily intended to measure the ability of performing motor tasks in a controlled environment and has an associated scoring system like the 'Functional Ability Scale' (FAS) or 'Fugl-Meyer Assessment' (FMA), [179]-[180] where the patient is given a score for the motor functions that are performed under the supervision of the respective clinician. Therefore, the WMFT can help in characterizing the functional ability of a patient and used as a tool in the prescription of a patient-specific rehabilitation program.

There is a standard protocol for conducting the tests which includes: task descriptions, starting positions for the subjects and the equipment used during the test, verbal instructions to be read out to the subjects prior to each test and a scoring criterion. In addition to the equipment required for each specific task, there are a few general requirements which include – a height adjustable table (approximately 137 cm × 76 cm); a straight back chair (46 cm) or wheelchair with firm back and without armrests; a WMFT template/poster attached on top of the table highlighting the markers and distance measures for each task (e.g. position of arm etc.); a stopwatch to note down the time required to complete each task. The scoring scale varies from 0 to 5 and the subjects are marked according to their performance by the observing clinician. A 0 implies that the subject does not attempt the task with the impaired arm whereas a 5 indicates successful task completion and a normal movement task with the impaired arm.

The seventeen motor tasks of the standard WMFT set [181] have been reduced to a select group of 8 tasks only, as shown in Table 3.1, which is commonly referred to as the streamlined WMFT set [182]. The streamlined WMFT set requires a shorter period of time to administer and is hence popularly used within the medical community, and is deemed sufficient for testing the functional ability of patients suffering from upper limb impairment. The task numbers in Table 3.1, represent those selected from the original list of 17 tasks. All tasks are performed while the patient is seated on a chair by a table.

Motor Function	Description	Functionality
WMFT 1	Forearm to table (side)	Shoulder movement
WMFT 2	Forearm to box (side)	Shoulder movement
WMFT 5	Hand to table (front)	Shoulder/elbow movement
WMFT 6	Hand to box (front)	Shoulder/elbow movement
WMFT 8	Reach and retrieve	Arm and hand
WMFT 9	Lift can	Dexterity/object manipulation
WMFT 12	Stack checkers	Dexterity/object manipulation
WMFT 13	Flip cards	Dexterity/object manipulation
WMFT 15	Turn key in lock	Dexterity/object manipulation
WMFT 16	Fold Towel	Dexterity/object manipulation

Table 3.1: Streamlined Wolf Motor Function Test

As mentioned, these tasks are primarily aimed at assessing the motor functionality of the survivors within a clinical environment. In the nomadic environment, there can be a large number/type of movements performed with the upper limb. However, for this research work, the movement selection was narrowed down following the streamlined version of the WMFT set. From discussions with expert clinicians and trained occupational therapists at the Brandenburg Klinik (BBK), Berlin, Germany, four fundamental arm movements were identified from these set of eight motor tasks (streamlined WMFT set). The four elementary movements of the upper limb are:

- *Action A* – Reach and retrieve an object, monitoring extension/flexion of the forearm;
- *Action B* – Lift cup to mouth, focussing on rotation of the forearm about the elbow;
- *Action C* – Reach out for an object sideways, monitoring rotation of the arm about the shoulder and
- *Action D* – Rotate wrist with arm fully extended, involving rotation of the wrist about long axis of forearm.

In principle, these elementary movements constitute a significant proportion of the complex movements performed with the upper limb in daily life and could also be mapped to the tasks numbered 8, 9, 1 and 15 respectively of the streamlined WMFT set (cf. Table 3.1) [181]. Moreover, the selected movements being elementary arm movements when performed in nomadic settings can also capture a wide range of variability inherent in human movement. Since the target is to develop algorithms that can track arm movements performed in daily life by a generic stroke survivor population, the algorithm needs to be robust against such variability. Therefore, if the model could be built using experimental data obtained from a significant number of subjects executing these fundamental movements, it has a better chance of being more robust to such inter-person variability.

### 3.3 Inertial sensors and their calibration

As highlighted in sections 2.2.5.2 and 2.2.5.3, body-worn inertial sensors are widely used for capturing kinematic information, from which various parameters are extracted that can help in the clinical assessment of the patients. MEMS based inertial measurement units (IMUs) containing three dimensional accelerometers and gyroscopes measuring accelerations and angular rate are popularly used in human activity monitoring. In some cases, three-dimensional magnetometers are also used. For this exploration, the commercially available Shimmer 9DoF wireless kinematic sensor module, consisting of mutually orthogonal tri-axial accelerometers, rate gyroscopes and magnetometers, was selected as the sensing platform. However, only the tri-axial accelerometer and the tri-axial rate gyroscope were used, leaving out the magnetometer since their measurement can be distorted by ferromagnetic materials [126] expected to be present in the nomadic environment of the survivors (cf. section 2.2.5.3).

The Shimmer module has an internal 2 gigabyte data storage capacity (smart card) as well as low-power radio communication capabilities (Bluetooth and IEEE 802.15.4) allowing both long-term data acquisition and real-time monitoring for experimental purposes. The core element of the Shimmer platform is the low-power Texas Instrument MSP430F1611 microprocessor controlling the operation of the device. The processor has 8 ADC channels for 12-bit A/D conversions for capturing sensor data. The firmware running on TinyOS environment enables the real-time streaming or storage of the captured data onto the micro SD card. The shimmer sensor modules are relatively light in weight, weighing 27 g and small in dimension measuring 44.5x20x13 mm and thereby pose minimal hindrance and discomfort for use over long periods. Hence, these features make it an ideal platform to be used for the experimental purpose. The shimmer unit is plugged onto a docking station and PC based application ‘Shimmer Connect’ allows a configurable parameter selection for the sensors [40]. A basic Shimmer sensor unit (housing tri-axial accelerometers) used in the experiments is shown in Figure 3.1. The positive X, Y and Z axis are labelled respectively. Correspondingly, the negative X, Y and Z axis represent the directions on opposite sides (not shown in Figure 3.1 for sake of clarity). These positions are referred as positive/negative X, Y and Z axis during the calibration procedure.

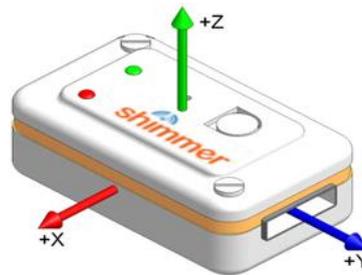


Figure 3.1: Sensor unit showing direction of positive response for tri-axial accelerometers.

Inertial sensors suffer from some existing errors in the accelerometer and gyroscope signals that cause unacceptable drifts and bias. Subsequently, compensation for inherent error sources is essential in the basic sensing device. To minimise these errors, the sensors need to be calibrated. Calibration involves subjecting the accelerometer and the gyroscope to a known acceleration or angular velocity respectively and recording sensor responses. The process of calibration provides the parameters for measurand sensitivity and the sensor offset value. These parameters are used to deduce the physical values of acceleration and angular velocity from the MEMS sensor output.

### 3.3.1 Accelerometer calibration

A typical MEMS accelerometer measures force rather than acceleration directly. This is achieved by measuring the displacement of an internal proof mass and making use of Newton's second law of motion ( $F = ma$ ). It is not necessary to know the value of the proof mass since this is accounted for during the calibration process (and it is assumed that it remains constant) and thus a calibrated MEMS accelerometer produces an output that is directly proportional to the acceleration experienced by it. Because an accelerometer actually responds to force, it is always subjected to the ubiquitous force of gravity. Though seemingly a drawback, this can in fact be quite beneficial. Firstly, it means that gravitational acceleration can be used as the reference value during calibration of the accelerometers. Secondly, in the absence of any other external forces acting on the accelerometer (e.g. when stationary) a simple analysis of the recorded value of gravitational acceleration as experienced by the accelerometer can be used for determining orientation information for postural tracking [125].

Calibrating the accelerometers can be a very simple process, making use of the omnipresent gravitational acceleration ( $g$ ) as a reference standard. The accelerometer is placed on a flat surface with its sensing axis either aligned, opposed or orthogonal to the

direction of gravitational acceleration, causing it to experience accelerations equal to  $+1g$ ,  $-1g$  or  $0g$  respectively. Hence a simple 3-point calibration procedure can be implemented without the need of additional or specialist equipment, and this simple calibration is adequate if it is known that the accelerometer exhibits a linear response to acceleration. However, three calibration points that define the upper, lower and midrange values of a measurand can be deceptive since both a linear or sine function can be fitted to them with high correlation. It is therefore highly recommended that where practical, calibrations are performed with more than three reference values. For an accelerometer, additional reference accelerations can be experienced by placing it on an inclined surface with known inclination angle so that the sensor axis experiences a component of the gravitational acceleration. With bi-axial and tri-axial accelerometers where the axes are mutually orthogonal, this technique has the advantage that a second accelerometer axis simultaneously experiences a different component of the gravitational acceleration based on an angle that is complementary to the slope inclination angle.

To perform this calibration, a rectangular containment unit, with perfectly parallel external faces, was first constructed to securely hold the sensor module. This is essential since the Shimmer sensors do not have an ideal shape for this form of calibration. Its external packaging has tapered sides and certain other features that are slightly in relief on two faces (e.g. nuts, bolts, LEDs), implying that it cannot be laid down perfectly flat on a surface. To extend the range of calibration coefficients available, plastic wedges with faces inclined at  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$  and  $40^\circ$  to the horizontal were also fabricated to provide fractional values of  $g$ , as described previously and as illustrated in Figure 3.2.

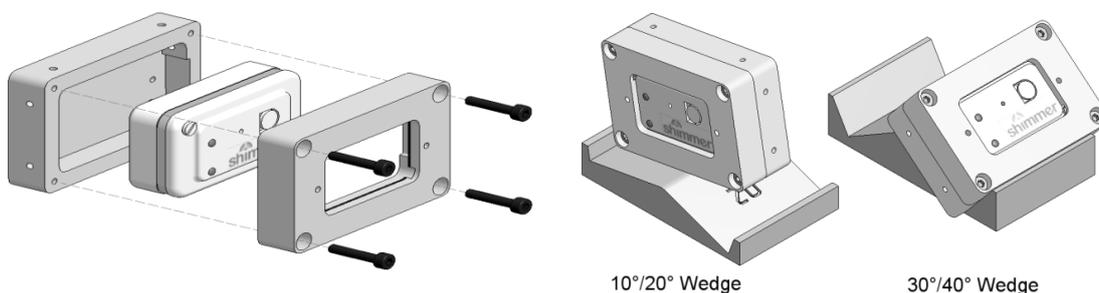


Figure 3.2: Illustration of Shimmer 9DoF sensor module enclosed in housing and resting on sloped wedges in different orientations during calibration [183].

The Shimmer sensors can have 24 possible standard orientations (6 faces, any notional edge pointing up, down, left or right). For illustration, the eight standard orientations of the sensor module when rotated about the positive Z-axis (Positions A – D) and the negative Z-axis (Positions E - H) are shown in Figure 3.3. The X, Y and Z axes are shown in red,

blue and green colours respectively. Similarly, there are eight standard orientations when rotated about the X and Y axes as shown in Figure 3.4 and Figure 3.5 respectively (all positions not shown for sake of clarity). These 24 possible orientations of the sensor module are named accordingly from Position A to Position X.

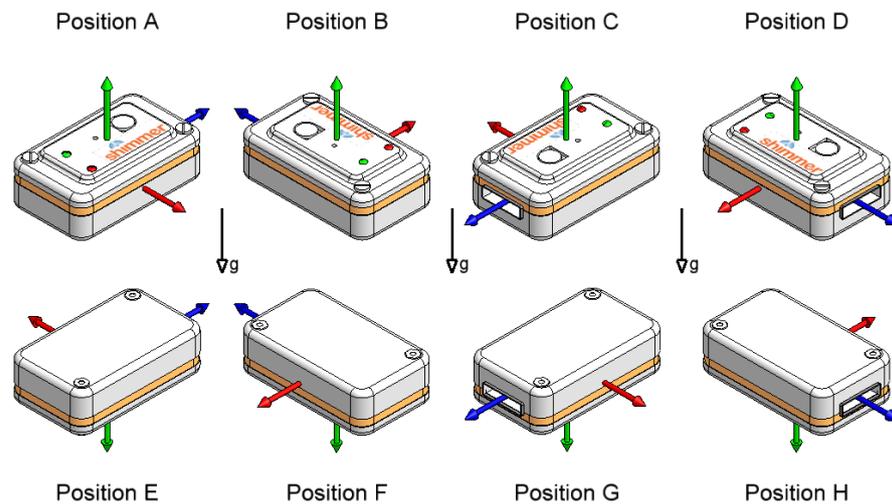


Figure 3.3: The eight standard orientations of the Shimmer sensor module when rotated about the Z-axis accelerometer.

➤ **Rotation about X-axis**



Figure 3.4: Sensor module rotated about the X-axis accelerometer.

➤ **Rotation about Y-axis**

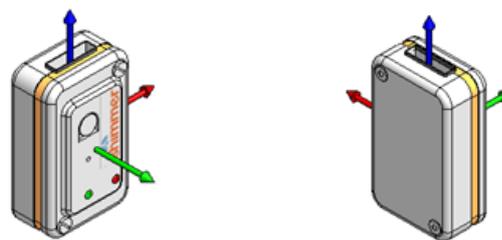


Figure 3.5: Sensor module rotated about the Y-axis accelerometer.

By placing the Shimmer sensor on each sloped surface (having fixed inclinations of  $10^\circ$ ,  $20^\circ$ ,  $30^\circ$  and  $40^\circ$  to the horizontal) and in each of its 24 possible standard orientations, the accelerometer responses from each axis can be recorded [183]. Hence, sensor data is collected at 50 Hz for a period of at least 8 seconds with the sensor module placed at each position (A – X) and is then plotted on separate graphs. A sample graph, showing acceleration data collected while the sensor is placed at an inclination of  $0^\circ$  to the horizontal in eight standard orientations that involve rotations about the Z-axis, is

illustrated in Figure 3.6 and Figure 3.7. The markers on the graph represent each of the eight reading windows comprising of 200 sensor data points, selected within the more stable portion of the displayed data (representing 4 seconds of data collection), corresponding to the orientations (Position A – E). The average values and the standard deviation are computed for the sensor output for each of these reading windows (200 data points) representing the eight positions.

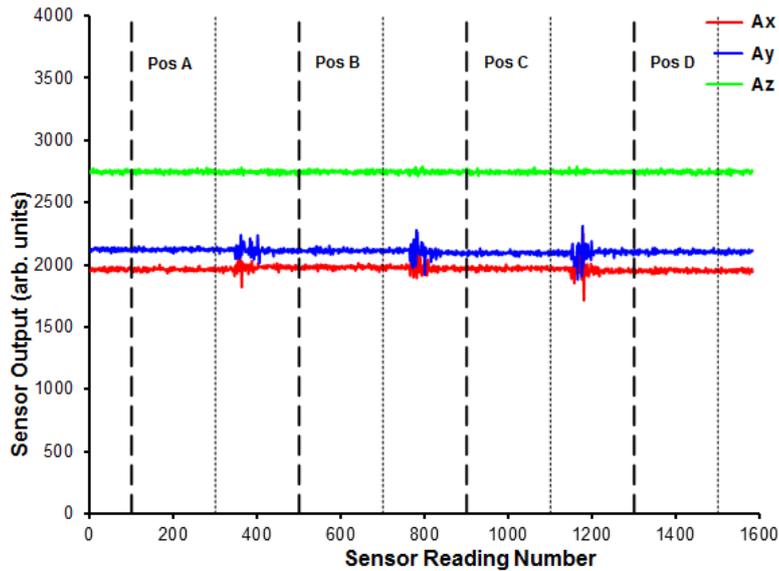


Figure 3.6: Tri-axial accelerometer output with the module inclined at  $0^\circ$  and rotated about the positive Z-axis. The markers (vertical dashed lines) represent a reading window of 200 sensor data points for each of the four standard orientations (Position A, B, C, D).

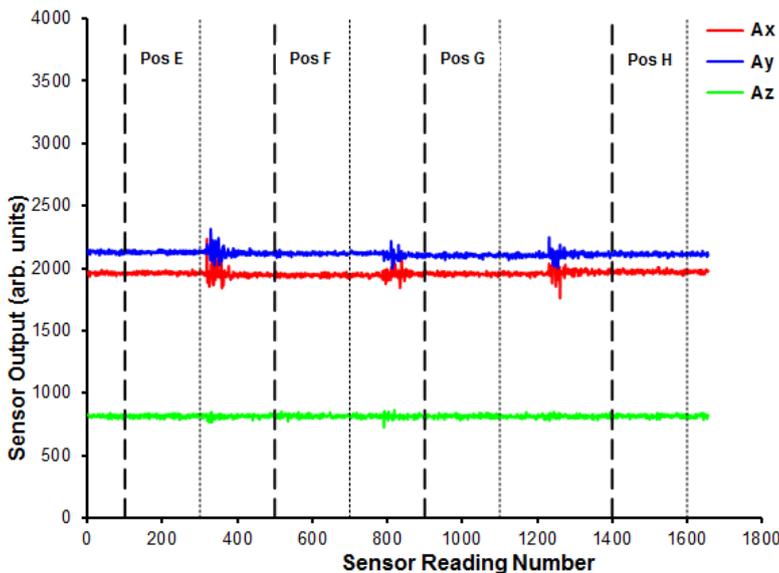


Figure 3.7: Tri-axial accelerometer output with the module inclined at  $0^\circ$  and rotated about the negative Z-axis. The markers (vertical dashed lines) represent a reading window of 200 sensor data points for each of the four standard orientations (Position E, F, G, H).

This procedure is repeated by placing the Shimmer sensor on the flat surface ( $0^\circ$ ), on each sloped surface ( $10^\circ$ ,  $20^\circ$ ,  $30^\circ$  and  $40^\circ$ ) and for each of the orientations (Position A – X) and

subsequently the average sensor values and their standard deviation are computed corresponding to each orientation. Therefore, a list of 120 *measured* average acceleration values (24 positions  $\times$  5 inclination angles) for each axis of the accelerometer was produced. These *measured* average sensor values can be plotted against *theoretical* values of acceleration to produce the required calibration information. Since the accelerometer is under static conditions, the *theoretical* values of acceleration for each accelerometer axis as a function of its standard orientation and wedge inclination angle  $\theta$  to the horizontal can be easily computed.

Using the least mean square linear fit to the calibration data, individual values for acceleration sensitivity and offset were determined for each accelerometer axis using the measured acceleration values of the form:

$$S_n = (m_n \cdot A_n) + C_n \quad (3.1)$$

where  $n = X, Y$  or  $Z$  axes.

$S_n$  = sensor output for axis  $n$

$A_n$  = acceleration experienced by accelerometer axis  $n$

$m_n$  = linearity function (gradient of slope of line of best fit), sensitivity

$C_n$  = offset value for accelerometer  $n$

From the computed values of the various parameters, a transfer function can be established that relates the sensor reading to acceleration values as follows:

$$A_n = \frac{S_n - C_n}{m_n} \quad (3.2)$$

An example of calibration data collected from the Shimmer 9DoF sensor module (plotting *measured* sensor output against *theoretical* acceleration values) is shown in Figure 3.8, which reveals that the accelerometers have a highly linear response over the acceleration range  $\pm 1g$  and quite a large difference in offset values. Since it is established that the accelerometers within the Shimmer module exhibit a highly linear response, henceforth, prior to any experiments performed in this research work the sensors are calibrated using a simple 3-point calibration procedure which is incorporated within the data logging software provided by Shimmer [184]. The accelerometer is calibrated by placing the sensor module in different orientations such that each axis is aligned towards, away or orthogonal

to the direction of gravity. This simple procedure provides a fast and effective solution prior to any data collection and the calibration parameters (sensitivity,  $m_n$  and offset,  $C_n$  thus computed are used to obtain the physical values of acceleration from the recorded sensor data. Routinely performing a simple calibration in this manner helps to reduce temporal errors in the sensor response, such as signal drift, which are inherent or associated with aspects of the sensor manufacturing process.

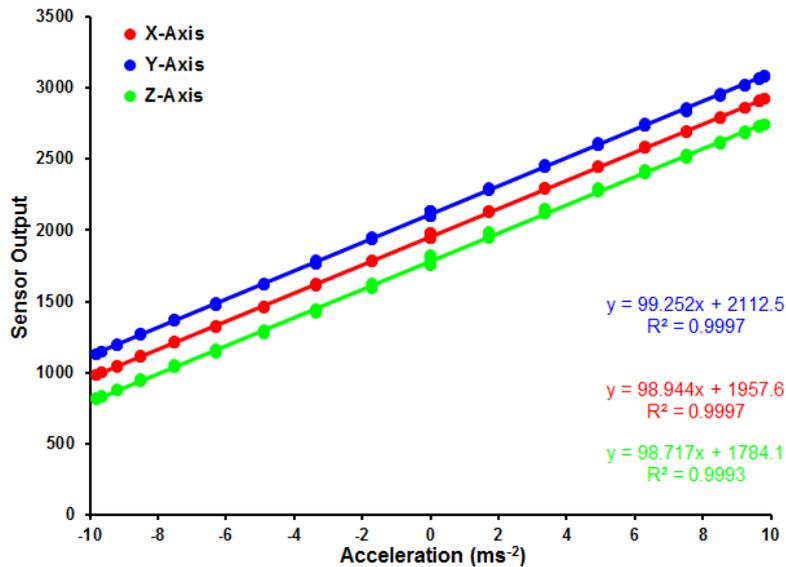


Figure 3.8: Example accelerometer calibration data showing highly linear responses over acceleration range  $\pm 1$  g with high correlation ( $R^2$ ) values [183].

### 3.3.2 Gyroscope calibration

A MEMS gyroscope is used for measuring the rate of rotation ( $^\circ/\text{s}$ ) without a fixed point of reference. Most common MEMS gyroscopes work on the principle of a tuning fork and are comprised of a pair of identical masses ( $m$ ) that are driven to oscillate with equal amplitude but in opposite directions as shown in Figure 3.9. The positive X, Y and Z directions are shown in the figure, the negative direction for each axis represent the opposite directions (not shown for sake of clarity). If one of the masses is moving in the positive X-axis direction with velocity  $V_x$  and an angular rotation  $\Omega_z$  is applied about the Z-axis, then the mass will experience a force ( $F_{\text{Coriolis}}$ ) in the direction of the arrow shown as a consequence of the Coriolis effect. Correspondingly, the other mass moving in the negative X-axis direction with the same velocity ( $V_x$ ) will also experience a Coriolis force of the same magnitude, but acting in the opposite direction [125]. These two forces can be measured by sensing mechanisms built into the MEMS structure (e.g. strain change measured with piezoresistor or deflection measured by capacitance change). In MEMS

based sensors, the change in the geometry of the piezoresistive material or the change in distance between two capacitor plates due to an effect of external force is transduced to an electrical signal obtained at the front end from the sensor device. Since the gyroscope measures the angular rate of rotation, it requires a source of rotation to excite the device. In relevant literature, turntables are generally used for calibrating the gyroscope [185].

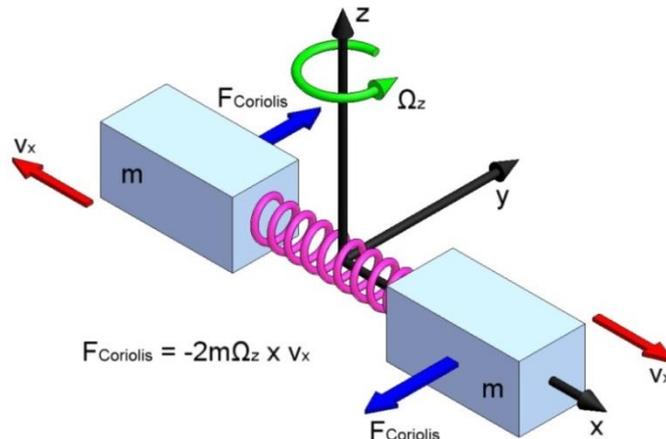


Figure 3.9: Principle of operation for MEMS vibrating gyroscope [125].

The Shimmer sensor module housed in the customized glass frame is placed on a record turntable that has two defined speeds of 33 rpm and 45 rpm. For each 24 standard orientations of the sensor module (Positions A – X) as shown in Figure 3.3 - Figure 3.5, data is collected at a rate of 50 Hz as the turntable rotates at each of these speeds. The turntable is initially kept stationary and then rotated in the following manner:

- 4 complete revolutions at 33 rpm ( $198^\circ\text{s}^{-1}$ ),
- 4 complete revolutions at 45 rpm ( $270^\circ\text{s}^{-1}$ ),
- 4 complete revolutions at 33 rpm ( $198^\circ\text{s}^{-1}$ ),
- 4 complete revolutions at 45 rpm ( $270^\circ\text{s}^{-1}$ ),
- 2 complete revolutions at 33 rpm ( $198^\circ\text{s}^{-1}$ ).

A sample graph, showing gyroscope data collected while the sensor is placed at Position A (cf. Figure 3.3) on the turntable and rotated in the above mentioned sequence is presented in Figure 3.10. A reading window of 100 sensor data points is selected (shown by the vertical markers) within the more stable portion of each different stage of the recorded data (representing 2 seconds of data collection) from which the average values and standard deviations are calculated. This process is repeated for all the 24 different sensor orientations placed on the record turntable.

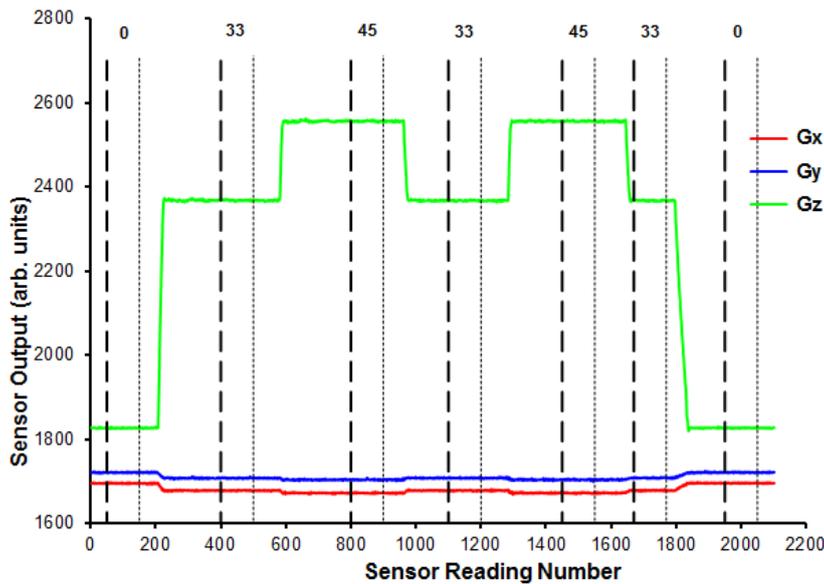


Figure 3.10: Gyroscope output with the sensor module placed in position A at various speeds (stationary, 33 and 45 rpm) alternatively.

Therefore each position (A – X) produces average sensor outputs and the corresponding standard deviation for the stationary phase and for the rotation speeds of 33 rpm and 45 rpm. Hence, a list of 72 *measured* average rotation values (24 positions  $\times$  3 rates of rotation) for each axis of the gyroscope is obtained. These *measured* average sensor values are plotted against *theoretical* values of rotation ( $0, \pm 198^\circ\text{s}^{-1}, \pm 270^\circ\text{s}^{-1}$ ) to produce the required calibration information, as illustrated in Figure 3.11.

Similar to the previous calculation involving the accelerometer reading (cf. section 3.3.1), using the least mean square linear fit to the calibration data, individual values for gyroscope sensitivity and offset were determined of the form:

$$S_n = (m_n \cdot G_n) + C_n \quad (3.3)$$

where,  $n = X, Y$  or  $Z$  axes,

$S_n$  = sensor output for axis  $n$

$G_n$  = angular rotation rate experienced by gyroscope axis  $n$

$m_n$  = linearity function (gradient of slope of line of best fit), sensitivity

$C_n$  = offset value for gyroscope  $n$

From the computed values of the various parameters, a transfer function can be established that relates sensor reading to angular rotation rate as follows:

$$G_n = \frac{S_n - C_n}{m_n} \quad (3.4)$$

Figure 3.11 shows that the gyroscopes are highly linear and that the X-axis gyroscope exhibits a response that is counter clockwise to the Y-axis and Z-axis gyroscope. The response of the X-axis with respect to Y and Z-axis was taken into account in all the sensor recordings performed during any experiments within this research work.

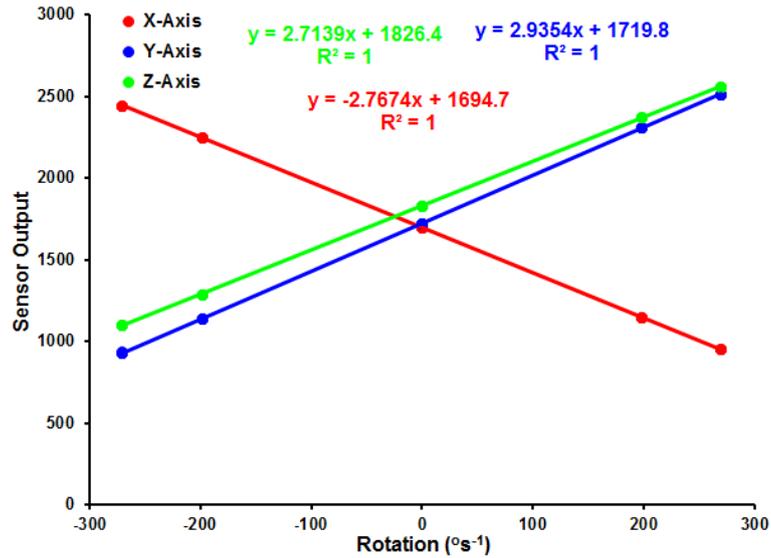


Figure 3.11: Gyroscope calibration curve showing a highly linear response with the X-axis response counter-clockwise to the Y and Z-axis.

Therefore, using the aforementioned calibration methodology, it is established that the accelerometer and gyroscope sensors exhibit a highly linear response. Henceforth, prior to any experiments performed in this research work the gyroscope is calibrated using the procedure incorporated within the data logging software provided by Shimmer [184]. Each axis of the gyroscope is calibrated by rotating the sensor through  $360^\circ$  about that axis, integrating the total response obtained over the time taken (angle  $\times$  rate) and dividing the result by 360 (rate). Rotations are performed both in clockwise and anti-clockwise directions, and measurements are also averaged when the gyroscope is stationary. The calibration coefficients thus determined are used to calibrate the raw sensor data before any further processing.

### 3.4 Estimation of error margin and drift

Inertial sensors are susceptible to several error sources like interference by noise and a drift in the recorded values over time [37]. Hence, it is important to estimate the effects of noise

and drift on the sensors used in for the experiments. To determine the level of noise inherent within the sensor (also referred to as the null bias error [186]), the module is left prone on a flat surface undisturbed for 24 hours. During this time, the sensor continuously streams data at 50 Hz and is oriented such that the Z-axis accelerometer faces gravity and the X and Y axes experiences zero acceleration (cf. Figure 3.1). A histogram plot of the sensor reading for the Z-axis of the accelerometer, illustrated in Figure 3.12, shows a normal distribution. The distribution shows the mean value and the upper and lower ranges for the sensor reading (given by three standard deviations, i.e.  $3\sigma$ ). The  $3\sigma$  value gives the error margin of each sensor axis under static conditions. This is because under the normal distribution hypothesis, 99.9% of the data samples lie within  $\pm 3\sigma$ .

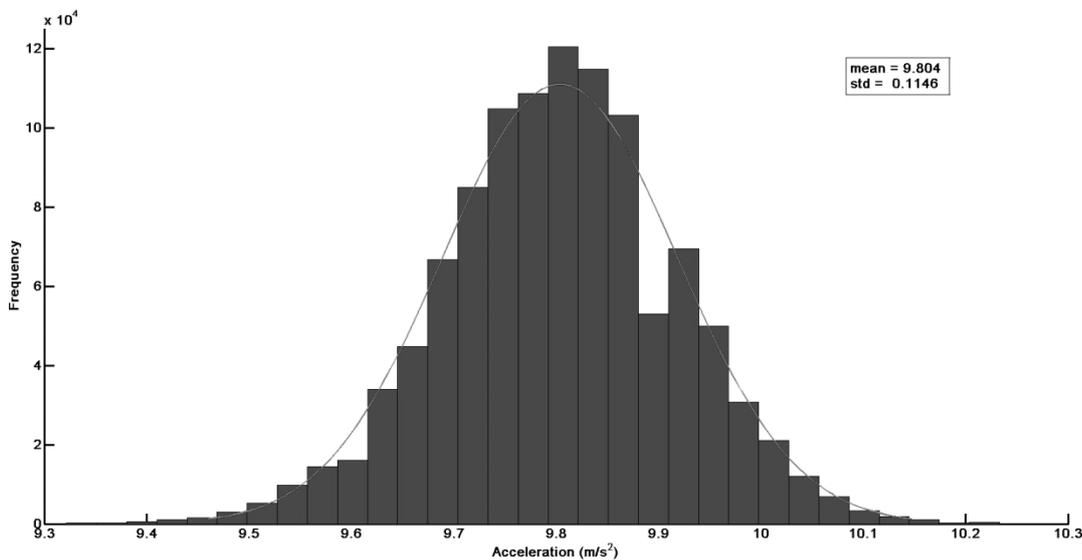


Figure 3.12: A stationary accelerometer with Z-axis facing gravity, showing a normal distribution with a mean acceleration of  $9.804 \text{ m/s}^2$ .

Similarly, the three gyroscope axes should experience no rotation, since the sensor module lies stationary. The distribution of the Y-axis gyroscope shown in Figure 3.13, illustrates the majority of the sensor readings are centered about the mean value ( $-0.01$ ). The mean and error margin for each sensor axes for one sensor module is presented in Table 3.2. The error margins for all the sensors used during the experiments are estimated similarly.

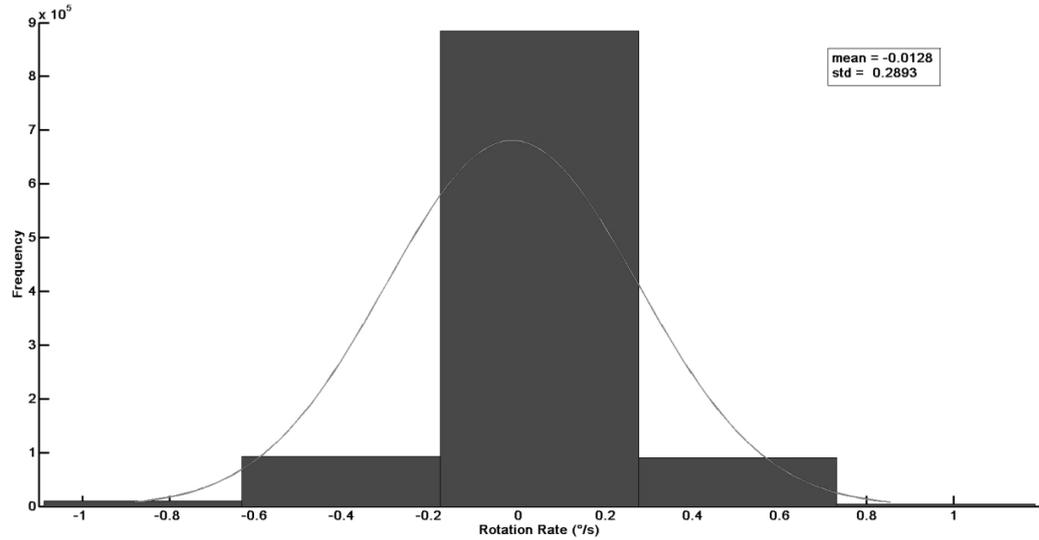


Figure 3.13: Y-axis reading of a stationary gyroscope with a mean rotation rate of  $-0.0128$   $^{\circ}/s$ .

Parameters	Accelerometer ( $ms^{-2}$ )			Gyroscope ( $^{\circ}s^{-1}$ )		
	X	Y	Z	X	Y	Z
Mean	-0.0150	0.0252	9.8037	-0.073	-0.0128	0.0541
Noise Margin ( $\pm 3\sigma$ )	$\pm 0.3243$	$\pm 0.3276$	$\pm 0.3438$	$\pm 0.9432$	$\pm 0.8679$	$\pm 0.9552$

Table 3.2: List of mean sensor readings and error tolerance ( $\pm 3\sigma$ ) when the sensor module is left static on a flat surface for 24 hours.

For estimating sensor drift, the sensors are calibrated one month apart using the simple 3-point calibration method as discussed previously [184]. The difference in the calibration parameters (sensitivity and offset values) represent the drift experienced by each sensor axis. It can be observed from Table 3.3 and Table 3.4 that there is a minor change (% change) in the sensor configuration over a month long duration. The sensitivity and the offset values have been abbreviated as  $S_A$ ,  $S_G$ ,  $C_A$  and  $C_G$  for the accelerometer and the gyroscope respectively in Table 3.3 and Table 3.4. Similar trends are exhibited by the other sensor modules used during the experiments.

	<u>X-axis</u>		<u>Y-axis</u>		<u>Z-axis</u>	
	$C_A$	$S_A$	$C_A$	$S_A$	$C_A$	$S_A$
<b>Before</b>	1897.79	98.01	2084.42	100.32	1658.34	98.63
<b>After</b>	1897.10	98.09	2084.25	100.25	1659.14	98.61
<b>%change</b>	0.03	0.08	0.01	0.06	0.04	0.02

Table 3.3: Drift in the sensitivity ( $S_A$ ) and offset ( $C_A$ ) values for the accelerometer for a sensor module recorded one month apart.

	<u>X-axis</u>		<u>Y-axis</u>		<u>Z-axis</u>	
	$C_G$	$S_G$	$C_G$	$S_G$	$C_G$	$S_G$
<b>Before</b>	1872.41	2.67	1763.41	2.85	1830.77	2.76
<b>After</b>	1872.05	2.66	1762.80	2.83	1830.27	2.75
<b>%change</b>	0.01	0.3	0.03	0.7	0.02	0.3

Table 3.4: Drift in the sensitivity ( $S_G$ ) and offset ( $C_G$ ) values for a gyroscope recorded one month apart.

### 3.5 Discussion

In view of the research focus to recognize occurrences of particular arm movements that are commonly used in daily life, this chapter, lists out the selection of four elementary arm movements to be tracked and their significance, the inertial sensors to be used and their respective calibration. The sensor characteristics in terms of the inherent error caused by interference due to noise and drift in the recorded values over time have also been looked into. Although the primary focus lies in capturing the discriminatory patterns pertaining to each arm movement for classifying them using the data generated by these sensors, it is essential to verify the reliability and repeatability of the sensors which produces the data over a prolonged time. The corresponding higher order statistical analysis performed on the sensor data makes it essential to calibrate the sensors (estimating the sensitivity and the offset values) to obtain a standardized output. The adopted calibration methodology for the tri-axial accelerometer and the gyroscope illustrate that the sensors exhibit a highly linear response. Based on this fact, the sensors are calibrated using a simple 3-point calibration methodology, to obtain the sensitivity and the offset values which are used to deduce the physical values of acceleration and rotation rates, prior to any data acquisition during the experiments conducted within this research work.

In the subsequent chapters, the activity recognition methodologies used to detect the arm movements in nomadic settings using optimal number and placement of the sensors have been presented.

## 4. Chapter 4

# Sensor Orientation based Movement Recognition

### 4.1 Introduction

In this chapter, a simple methodology is presented for recognising three arm movements (reach and retrieve, lift cup to mouth, pouring/(un)locking), performed during an archetypal daily activity of '*making-a-cup-of-tea*'. The methodology is based on determining the orientation of a tri-axial accelerometer located near the wrist. Prior to exploring conventional means of learning methodologies (e.g. classification), here an alternative methodology has been discussed, based on the analysis of the kinematic data with particular impetus on the activities being monitored. The primary motivation for this exploration was to avoid the essential step of learning or a *training* phase which is used in conventional sensor-based activity recognition as discussed in section 2.2.5.6.5, involving complex data processing using feature extraction/selection and a range of learning algorithms.

This led to the development of a simple methodology that allows detection of these three movements performed in an uncontrolled environment with maximal accuracy, while accounting for their temporal and inter-subject variability, using a low-complexity algorithm that can be implemented in a resource-constrained body-worn wireless sensor node [33]. It is important to note that in Chapter 3, section 3.2, four elementary arm movements were mentioned which are to be detected as part of this research. However, in this chapter, the developed algorithm is aimed at detecting three arm movements performed during the process of '*making-a-cup-of-tea*'. These three arm movements are a subset of the four movements to be originally investigated and represent common activities performed in daily life which is further evident from the archetypal activity-list of Table 4.1.

## 4.2 Experimental setup

An activity-list (cf. Table 4.1) was designed which emulated the process of ‘*making-a-cup-of-tea*’, a common activity performed in daily life, having repeated occurrences of three types of arm movement (*actions*) [115]. The activity-list in the experimental protocol comprises 20 individual activities including 10 occurrences of *Action A*, and 5 each of *Action B* and *Action D*.

Activity	Action
1. Fetch cup from desk	A
2. Place cup on kitchen surface	A
3. Fetch kettle	A
4. Pour out extra water from kettle	D
5. Put kettle onto charging point	A
6. Reach out for the power switch on the wall	A
7. Drink a glass of water while waiting for kettle to boil	B
8. Reach out to switch off the kettle	A
9. Pour hot water from the kettle in to cup	D
10. Fetch milk from the shelf	A
11. Pour milk into cup	D
12. Put the bottle of milk back on shelf	A
13. Fetch cup from kitchen surface	A
14. Have a sip and taste the drink	B
15. Have another sip while walking back to desk	B
16. Unlock drawer	D
17. Retrieve biscuits from drawer	A
18. Eat a biscuit	B
19. Lock drawer	D
20. Have a drink	B

Table 4.1: Use case activity list - ‘Making-a-cup-of-tea’

In this investigation, experiments were performed within an open laboratory with an attached kitchen at the University of Southampton (UoS) with four healthy subjects (age range 24 to 40, male, all right arm dominant) and within a treatment centre at the Brandenburg Klinik (BBK), Germany, with four stroke impaired patients (age range 45 to 73, both sexes, both left and right arm dominant). The stroke patients performed the movements under the supervision of the expert physiotherapist members of the research team, using the same set of equipment as used for the healthy subjects. A brief overview of the degree of impairment of the four subjects has been provided in Appendix A. All involved team members had trained together in the use of the equipment prior to commencing these investigations. Healthy participants were requested to perform the chosen activity-list four times within the same session, with a 10 minute rest period between repetitions. By comparison, stroke patients performed the activity-list only twice. The disparity in the number of trials performed by healthy subjects and stroke patients was

due to the fact that the latter tend to tire quickly and were asked to perform the tasks only whilst they felt comfortable to do so. Under the given circumstances, this was the maximum trials that the patients could perform within the stipulated experiment session. For the experiments, there were no restrictions on the seating or standing position with respect to the kitchen surface or the time required to complete the actions. The activity-list was prepared to facilitate the evaluation of the recognition methodology under semi-naturalistic conditions [39] representative of the *nomadic/daily living* phase.

As mentioned in Chapter 3 (cf. section 3.3), the Shimmer 9DoF wireless kinematic sensor module was used as the sensing platform for the experimental purpose. Only the tri-axial accelerometer and the tri-axial rate gyroscope were used, leaving out the magnetometer. Two positions on the dorsal side of the arm (forearm proximal to the wrist, and upper arm proximal to the elbow) were used as the sensing positions and were chosen as those locations were likely to produce the largest sensor responses to the arm movements being investigated. The sensor is placed on the dominant arm for healthy subjects and on the impaired arm for stroke survivors. The XY plane of the sensor module (cf. Figure 3.1) was in contact with the dorsal side of the forearm, AccX (GyroY) points toward the fingers and AccZ (GyroZ) points away from the dorsal aspect. The Shimmer sensors were attached to the arm using elastic straps, providing an intimate, secure, yet un-constraining hold.

Sensor data is collected at a rate of 50 Hz, deemed sufficient for assessing habitual limb movement which is on the higher side compared to assessing holistic activity as in [2], [157]. The accelerometer and gyroscope ranges are selected at  $\pm 1.5g$  and  $\pm 500^\circ/\text{sec}$  respectively. The sensors transmit kinematic data along with a time stamp to a host computer using the Bluetooth wireless standard. Data from multiple streaming sensor modules is synchronised with respect to their individual time stamps and each activity performed by a subject is marked to record the beginning and end of the activities performed during the trial. The start and stop time of the activities were noted down by the researcher observing them as the subjects performed the designated tasks. The corresponding data collected was segmented using the annotations from the researcher who observed the subjects as they performed the designated tasks. For this exploration, data obtained only from the tri-axial accelerometer placed proximal to the wrist is considered, leaving out the gyroscope and the other sensing position (i.e. elbow) since here the aim was to produce a methodology using a minimal number of sensors and with a minimal amount of data processing.

The accelerometers were calibrated prior to performing any measurements following the technique mentioned in section 3.3.1. By applying a least mean square linear fit to the calibration data, individual values for acceleration sensitivity and offset were determined for each accelerometer axis. These calibration coefficients were used to calibrate the data collected during the experimental session. The developed algorithm mines the accelerometer data, analyzing transitions between occurrences of six predefined reference orientations (or states) of the sensor to predict the corresponding arm movements. The methodology was evaluated on the experimental data acquired from four healthy subjects and four stroke survivors as they performed multiple trails of ‘*making-a-cup-of-tea*’.

### 4.3 Algorithm design

Estimation of the upper limb orientation and body positioning is achieved through fusion and processing of heterogeneous sensor data obtained from magnetometers or inertial sensors comprising of accelerometers and gyroscopes attached to the body segment. With the aid of a kinematic model, the position of the individual body segments can be determined in 3D space. The majority of the proposed solutions in literature are based on Kalman Filter and its derivatives as the sensor fusion algorithm for estimating orientation [187]–[191], along with other methods based on complimentary filters and gradient descent methods [192], [193]. However, all these well-known methods which are targeted towards determining position and orientation of the desired body segment (e.g. upper limb) employ complex computations and are hence not applicable for this research exploration where the target is to develop an algorithmic technique that can be implemented in the resource constrained environment of a wireless body-worn sensor node.

The algorithm developed here is based on predicting the most likely orientation of the sensor module at any particular time by assessing which of the three accelerometer axes is the most active at that time. Specific arm movements are then inferred by detecting sequence of transitions between sensor module orientations. This is a relatively simple process when the arm is stationary since the total acceleration measured by the sensor module is equal to the value of the gravitational acceleration ( $g$ ) and its distribution over the three mutually orthogonal accelerometer axes directly indicates the orientation of the sensor module. To appreciate how this simple algorithm works, it is necessary to first understand how a MEMS type accelerometer responds to acceleration. Typically a MEMS accelerometer measures the displacement of a suspended seismic mass either by detecting

changes in strain via a piezoresistive element at the root of a cantilever beam to which the seismic mass is attached, or by detecting changes in capacitance of a micro-machined structure where the seismic mass forms one plate of a capacitor. The Shimmer sensor uses the Freescale MMA7361L tri-axial accelerometer which operates using the capacitive principle, as illustrated in Figure 4.1. Here, the seismic mass is free to move between two fixed plates, effectively forming two back-to-back capacitors ( $C1$  and  $C2$ ), whose values can be measured using a differential charge amplifier. It is known that capacitance is inversely proportional to the separation between the plates (distance), therefore, measurement of capacitor value will be indicative of the position of the seismic mass relative to the sensor frame. In Figure 4.1,  $C1$  and  $C2$  are the upper and lower capacitors respectively, forming a bridge circuit for measuring a differential voltage. The subscript 's' refers to the static condition when the accelerometer only experiences the gravitational acceleration and the subscript 'd' refers to the dynamic condition when the accelerometer experiences the physical acceleration in addition to the gravitational acceleration.

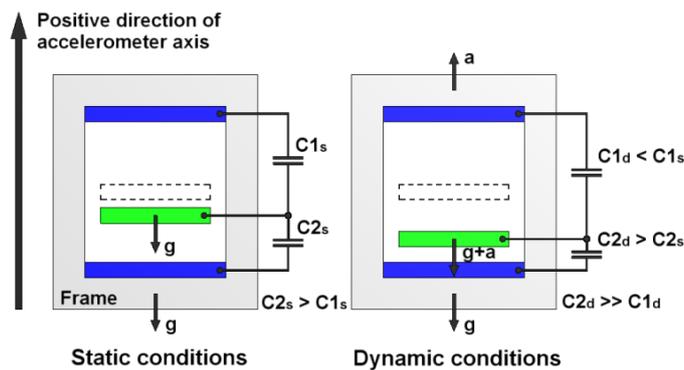


Figure 4.1: Operation of a differential capacitive type MEMS accelerometer. The rectangle with broken line shows the initial position of the seismic mass when no acceleration is experienced in the direction of the accelerometer axis and the rectangle in solid lines (green) represents the mass in displaced condition.

How an accelerometer acts in response to two different types of acceleration (physical and gravitational) is quite different. The force of gravity is ubiquitous and is the process which gives mass the quantity of weight. This attractive force is also bi-directional, meaning that any object on the surface of the Earth is attracted toward the centre of the Earth with the same magnitude of force as that object attracts the Earth toward it. Correspondingly, there are two vectors for acceleration due to gravity: one accelerating an object toward the centre of the Earth and the other accelerating the Earth toward the centre of the object. However, because the Earth is considerably more massive, it is only the acceleration vector that acts on the object accelerating it in a direction toward the Earth that is observable (cf.  $F = ma$ ). Consequently, the seismic mass located within the accelerometer is pulled toward the Earth with an acceleration of magnitude 'g'. By comparison, when an accelerometer is physically

accelerated ( $a$ ) by an external force acting on it, the frame of the sensor moves forward in the direction of acceleration but the suspended seismic mass moves backward (due to inertia) with an equivalent acceleration. This will result in the counter-intuitive situation shown in Figure 4.1 whereby the seismic mass moves in the same direction when under the influence of gravity alone or when physically accelerated in the opposite direction to that of gravitational acceleration.

This can be further explained with reference to Figure 4.1. Under the static conditions, when the accelerometer experiences the gravitational acceleration only, the seismic mass is closer to the bottom plate than the top and hence  $C_2$  has a much larger value than  $C_1$  (i.e.  $C_{2s} > C_{1s}$ ). However, under the dynamic condition, the seismic mass is even closer to the bottom plate and so now  $C_2$  is even greater than  $C_1$  (i.e.  $C_{2d} \gg C_{1d}$ ) and also larger than its previous value under static conditions (i.e.  $C_{2d} > C_{2s}$ ). With the configuration shown in Figure 4.1, the capacitors are connected to processing circuitry such that when  $C_2$  is greater than  $C_1$ , the output of the sensor has a positive value. This means that for the orientation shown in Figure 4.1, an upward acceleration ( $a$ ) in the positive direction of the accelerometer axis results in a positive output voltage [183]. Having this working principle in mind, 6 reference orientations of the sensor module in the horizontal plane were defined, referred to as Positions 1-6, as illustrated in Figure 4.2.

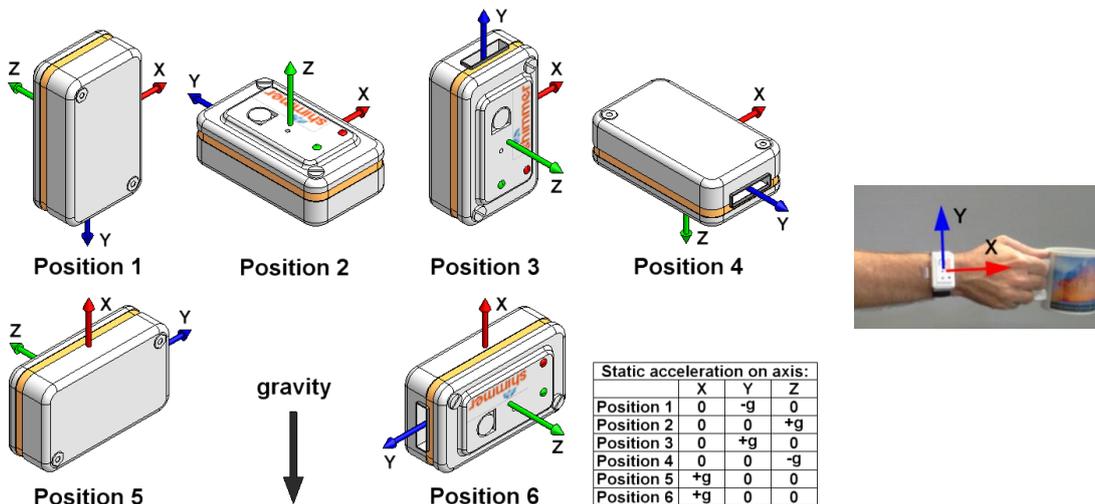


Figure 4.2: Predefined orientations of the sensor module with respect to the direction of gravity, showing positive directions of accelerometer axes [183]. An illustration of the sensor worn on the wrist is also presented.

For each position the dorsal side of the forearm is in contact with the XY plane of the sensor, the X-axis points toward the fingers (illustrated in Figure 4.2) and the Z-axis points away from the dorsal aspect. An illustration of the sensor module worn on the wrist has

also been shown in Figure 4.2. The positions shown cater for all orientations expected when performing the target actions with the sensor module secured to either the right or left arm, and with forearm movement constrained to the horizontal plane. Figure 4.2 also shows the distribution of the gravitational acceleration between the three accelerometer axes when the forearm is stationary at each position.

With reference to Figure 4.2, Positions 1 to 4 represent sequential 90° rotations of the forearm about the median axis of the arm (i.e. rotation about the X-axis accelerometer), and transitions between subsets of these positions occur with activities such as using a key to open or close a lock or when performing a pouring action (i.e. *Action D*). The transition from Position 1 to Position 5 represents a 90° rotation of the forearm about the elbow (i.e. rotation about the Z-axis accelerometer), which occurs when the sensor is worn on the left arm and the forearm is rotated in a drinking action (i.e. *Action B*). The corresponding orientation transition for when the sensor is worn on the right arm is given by Position 3 to Position 6 [183]. The third arm movement of interest (*Action A*) is not as simple to detect since it may not necessarily involve a change in orientation of the sensor. Typically, a reach and retrieve action may be performed with the sensor remaining in Position 2 (e.g. hand palmar side down grasping an object such as when using a computer mouse) or remaining in Position 3 when worn on the right arm or Position 1 when worn on the left arm (e.g. hand midway between pronation and supination such as when grasping a mug). The sensor may also remain in Position 4 during a reach and retrieve action, such as when accepting an object in the palm of the hand, though none of the activities defined in the activity list use this type of movement. Quite frequently, *Action A* involves a change in sensor orientation as the forearm rotates to adopt a particular prehensile shape for the hand prior to grasping an object. Often this means that the orientation of the hand during the retrieve phase of the motion is not the same as that during the reach phase. The key steps involved in the algorithm, which has been implemented in MATLAB, are illustrated in Figure 4.3 and described in the following sections. An overview of the algorithm used to detect and classify each action is also presented in the flow chart shown in Figure 4.4.

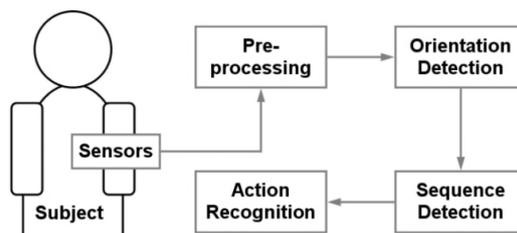


Figure 4.3: Basic steps of movement detection using sensor orientation[183]

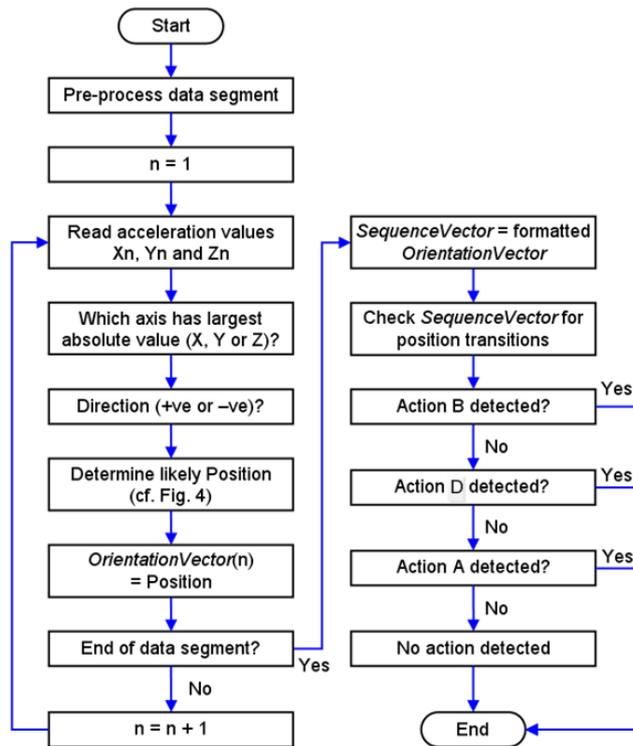


Figure 4.4: Flowchart for arm movement detection [183].

### 4.3.1 Pre-processing

Raw accelerometer data is converted to physical values of acceleration expressed in units of ‘g’ using individual sensor calibration coefficients. The event markers signifying the start and end of each activity that were recorded during data capture are used to segment the data into the 20 activities listed in Table 4.1. Each data segment is then filtered using a 3rd order low-pass Butterworth filter with a cut-off frequency of 5 Hz to suppress the high frequency noise components.

### 4.3.2 Orientation detection

Each data segment contains a number of individual data samples, which in turn are comprised of three acceleration values: X, Y and Z. For each data sample, the maximum absolute value, its direction, and on which axis this occurred were determined. This information is used to determine the most likely orientation of the sensor module and the identifying number of the corresponding position is stored as an element in an *OrientationVector*. The rules used to determine the orientation are as follows:

- If the maximum acceleration occurs on the Y-axis is negative, and lies within the range  $-g \pm 0.5g$ , the corresponding orientation of the sensor is denoted as Position 1.
- If the maximum acceleration occurs on the Z-axis is positive, and lies within the range  $g \pm 0.5g$ , the corresponding orientation of the sensor is denoted as Position 2.
- If the maximum acceleration occurs on the Y-axis is positive, and lies within the range  $g \pm 0.5g$ , the corresponding orientation of the sensor is denoted as Position 3.
- If the maximum acceleration occurs on the Z-axis is negative, and lies within the range  $-g \pm 0.5g$ , the corresponding orientation of the sensor is denoted as Position 4.
- If the maximum acceleration occurs on the X-axis, is positive, and lies within the range  $g \pm 0.5g$ , the corresponding orientation of the sensor is denoted as Position 5 if worn on the left arm or Position 6 if worn on the right.
- If none of these conditions are met, a value of 0 is assigned to the *OrientationVector* indicating an unknown position.

As described above, the decision as to which orientation the sensor module lies is based on acceptable acceleration ranges to allow for the fact that subjects may not orientate their arm such that the sensor module is exactly in one of the orthogonal positions shown in Figure 4.2. In practice, the forearm may be slightly rotated and the sensor module will therefore be tilted with respect to the pre-defined positions. Furthermore, basing orientation determination on a range of acceleration values caters for stroke patients who exhibit varying degrees of tremor in the upper limb depending on their time post-stroke. Limit values for these ranges of  $\pm 0.5g$  were established experimentally and gave the best action detection rates [183].

### 4.3.3 Sequence detection

Having established the most probable orientation of the sensor module for each set of acceleration data recorded, a *SequenceVector* is constructed from the annotated positions stored in the *OrientationVector*, wherein a sequence is defined as a continuous set of orientations of the same type and spanning a minimum duration. This process involves initially passing the *OrientationVector* through a median filter to remove all zeroes (unidentified orientations). The remaining set of position numbers is further scanned and any repeating sequence with a length less than 13 samples (representing 0.26 seconds of data) is discarded. This step is performed since it is considered that the forearm is only

truly orientated in a particular position if it remains in that position for at least one quarter of a second. The remaining set(s) of position identifiers forms the *SequenceVector* [183].

#### 4.3.4 Action recognition

The *SequenceVector* is examined to detect pre-defined transitions between identified positions that correspond to the target actions, as described in Table 4.2.

Position Transitions	Arm	Action
Remaining in Positions 1, 2 or 3	Both	A
1 → 2            or    2 → 1	Left	A
1 → 2 → 1    or    2 → 1 → 2	Left	A
3 → 2            or    2 → 3	Right	A
3 → 2 → 3    or    2 → 3 → 2	Right	A
1 → 5 → 1	Left	B
3 → 6 → 3	Right	B
Any transition between subsets of Positions 1 to 4	Both	D

Table 4.2: Sequence transitions and corresponding actions. Transitions for Action A involve additional processing to distinguish from D.

Firstly, the *SequenceVector* is analysed to check for *Action B*, which is a simple transition from Position 1 to Position 5 then back to Position 1 when the sensor is worn on the left arm or from Position 3 to Position 6 and back to Position 3 when the sensor is worn on the right arm. This movement is indicative of a drinking activity, as depicted in Figure 4.5.

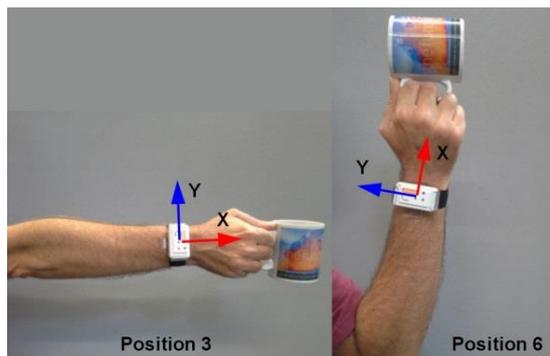


Figure 4.5: Transition from Position 3 to Position 6, corresponding to a drinking type of activity (*Action B*) with sensor on right arm [183].

Secondly, the *SequenceVector* is searched for those orientation transitions that are indicative of *Action D*, which involves rotation of the arm about the median axis. This action may involve a number of different transitions between Positions 1, 2 and 3. For example, a pouring action is described by the position sequence 3 to 2 to 1 (or just 3 to 2) for when the sensor is worn on the right arm, and by the sequence 1 to 2 to 3 (or just 1 to 2)

when the sensor is worn on the left arm. For both left and right arm worn sensors, Position 4 is used when extra torque is required in the rotation action, for example when using a key and lock.

Once a decision about *Action B* and *Action D* have been made, the occurrence of *Action A* (reach and retrieve), which is a very generic activity performed in daily life is determined. *Action A* can result in several different transitions in the activity-list, or demonstrate no transitions at all (cf. Table 4.2). For an action aimed at reaching out for an object in the forward direction is performed, the sensor may remain in Position 1, Position 2 or Position 3 throughout the duration of the activity or it may change between combinations of these Positions depending on the starting position of the forearm and the type of object to be retrieved (i.e. how it will be grasped in the hand). Hence the occurrence of *Action A* has to be confirmed by additional processing.

To identify *Action A*, the *SequenceVector* is first examined to determine how many sequences of consecutively identified orientations it contains. This indicates the number of state changes that occurred during the movement. For each individual sequence in the *SequenceVector*, the accelerometer axis that is aligned with the direction of gravity for that particular orientation (as defined in Figure 4.2) is discarded and the acceleration ranges (maximum value – minimum value) recorded by each of the other two axes over the duration of that sequence is calculated. Specifically:

- For Positions 1 and 3, the acceleration ranges for the X and Z-axes are computed, neglecting the Y-axis.
- For Position 2, the acceleration ranges for the X and Y-axes are computed, neglecting the Z-axis.

Therefore, any non-zero value calculated for the range from either of these accelerometer pairs would indicate:

- Movement of the forearm within the plane defined by the two accelerometer axes; or
- Rotation of the forearm about either or both of the accelerometer axes.

However, forearm rotation will have already been detected earlier in the algorithm when testing for the occurrence of *Action D*, and therefore the second scenario never arises. For

each orientation sequence, the computed acceleration ranges are compared against an acceptance range of  $\pm 0.2$  g. For a movement to be recognised as *Action A*, either or both of the computed acceleration ranges must be larger than this acceptance range (indicating movement within the horizontal plane) for the majority of sequences stored in the *SequenceVector*, otherwise the movement is considered as an *Unknown Action*. Limits for this acceptance range of  $\pm 0.2g$  were determined experimentally and allow for the fact that stroke impaired patients will demonstrate some degree of tremor in their arm movements which could be erroneously interpreted as directional movement.

## 4.4 Results and analysis

The algorithm was evaluated on the data collected in the *semi-naturalistic* setup involving the archetypal activity-list (cf. Table 4.1) emulating the process of ‘*making-a-cup-of-tea*’ comprised of 20 individual activities having 10 occurrences of *Action A*, and 5 each of *Action B* and *Action D*.

### 4.4.1 Healthy subjects

The results for four healthy subjects each performing 4 trials of the activity-list, involving 20 occurrences of the target actions are presented in Table 4.3. It can be observed that the average accuracy of correctly recognising the 3 movements over the 4 trials for all subjects is within a range of 91%-99%.

Subject	Recognised Actions (Out of 20)				Average
	Trial 1	Trial 2	Trial 3	Trial 4	Accuracy (%)
1	19	20	19	20	98
2	16	20	18	19	91
3	19	18	19	18	93
4	20	19	20	20	99

Table 4.3: Recognition of trials for healthy subjects [183].

### 4.4.2 Stroke survivors

The results for four stroke patients each performing two trials of the designated activity-list are presented in Table 4.4. The average recognition accuracy over the two trials for all four patients is within a range of 70%-85%.

Subject	Recognised Actions (Out of 20)		Average Accuracy (%)
	Trial 1	Trial 2	
1	17	17	85
2	14	14	70
3	16	14	75
4	17	11	70

Table 4.4: Recognition of trials for stroke survivors [183].

## 4.5 Summary of explorations

A novel yet simple algorithm has been presented aimed at recognizing three elementary upper limb movements involved in performing a representative activity of daily living, namely the process of ‘*making-a-cup-of-tea*’. In this algorithm, the occurrences of six pre-defined orientations of the sensor module are recognized by examining the distribution of recorded accelerations across the 3 mutually orthogonal accelerometer axes. The sequence of allocated orientations and their transitions are analysed to recognize the performed arm movements. The fact that an accuracy range of 91%-99% for healthy subjects and 70%-85% for stroke patients can be achieved using only a tri-axial accelerometer located on the wrist, demonstrates an alternative approach that reduces the overheads associated with complex data processing algorithms involved in conventional methodologies of human movement recognition. The approach described here has the advantage of negating the requirement of training a system to learn response patterns, as is the case in most conventional pattern recognition systems. However, the pre-defined orientations and the transitions are particularly aimed at recognizing the investigated movements, performed in a horizontal plane. Therefore, this methodology is restrictive to these movements since it is not scalable to adapt to other categories of movements. Nevertheless, it suits the given application requirement of detecting the three chosen arm movements.

The proposed algorithm has low computational complexity, mostly involving comparisons, additions, subtractions and few fixed multiplications. It is therefore suitable for translation into hardware, and can be used as a low-power processing component in a wireless sensor node based activity recognition system aimed at real-time operations.

## 4.6 Hardware design

In view of its low-computational complexity, and an added advantage of not requiring a training system to learn response patterns, the algorithm was implemented on a

reconfigurable platform (i.e. FPGA) for real-time detection of the three arm movements. The algorithm is coded using a hardware description language (System Verilog) and synthesized on the Altera DE2-115 Cyclone IV FPGA board. The DE2 board does not have a Bluetooth receiver and hence for real-time operation, interfacing between the streaming sensor unit (i.e. wrist-worn accelerometer), host PC and the FPGA was done through a combination of Bluetooth, RS232 and application software developed in C# using the .NET framework to facilitate serial port controls. An overview of the hardware setup has been shown in Figure 4.6. A detailed description of the architectural design, its evaluation and the sensor interface with the FPGA for real-time operations have been presented in the subsequent sections.

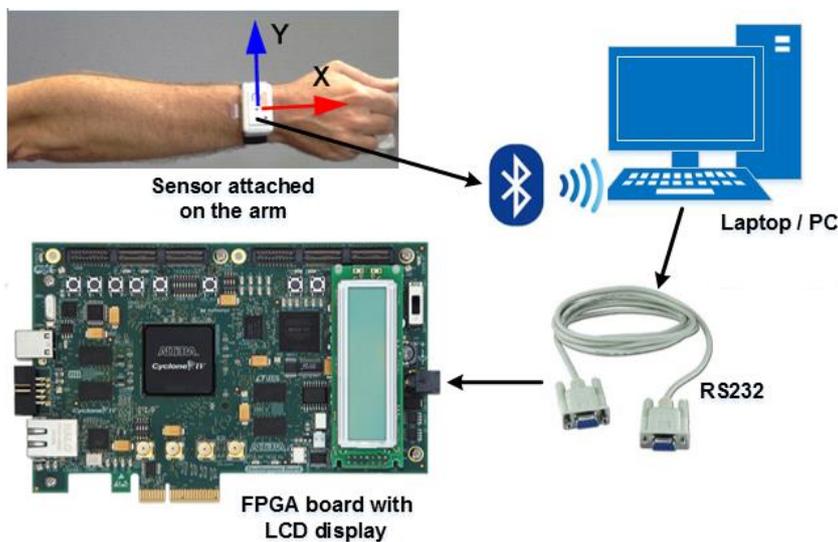


Figure 4.6: Setup for the real-time recognition of arm movements using the sensor orientation algorithm.

## 4.7 Algorithm to architecture mapping

The architecture presented in Figure 4.7, is divided into three modules having three 16-bit inputs for the tri-axial acceleration data and one 4-bit output for the detected arm movement. The *Orientation Detection (OD)* module determines the orientation of the sensor from each tri-axial input data sample (X, Y and Z). The corresponding orientations are sent to a subsequent module called *Sequence Detection (SD)* which tracks the length of the sequence of the orientation states and saves the unique orientations in a register. The module *Action Detection (AD)* is used to infer the performed arm movements (*Action A, B and D*) by looking for pre-defined transitions among the unique orientation states. The detected movements (actions) are displayed on an array of seven segment LEDs as well as a 2×16 monochromatic LCD screen on the FPGA board.

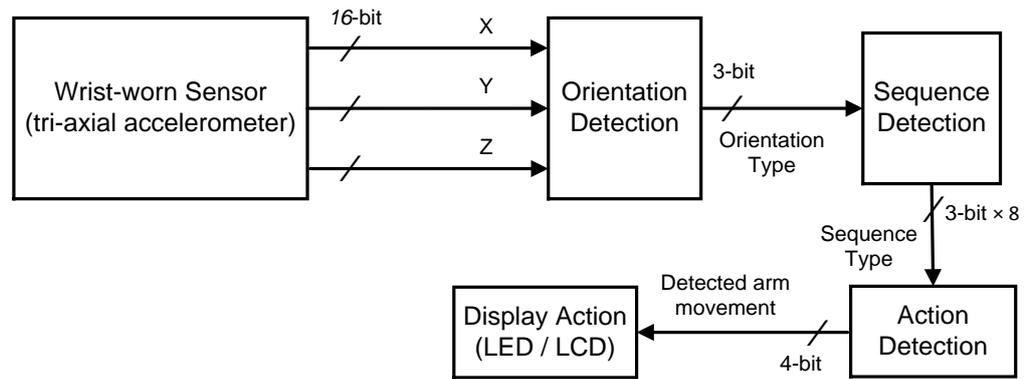


Figure 4.7: Architectural overview of the sensor based orientation algorithm.

#### 4.7.1 Orientation detection (OD)

Each performed movement generates a data segment comprised of individual samples from each accelerometer axis (X, Y and Z). A segment length of 512 samples is considered (which can be represented on a dyadic scale, i.e.  $2^n$  and hence any divisions or multiplications by the segment length can be implemented by  $n$ -bit shift operation), implying duration of 10 seconds, for each movement which is deemed sufficient time, even in view of the stroke survivors exhibiting varying levels of impairment. The absolute maximum acceleration value, its polarity and the corresponding axis for each data segment is computed using a maximum detector as illustrated in Figure 4.8.

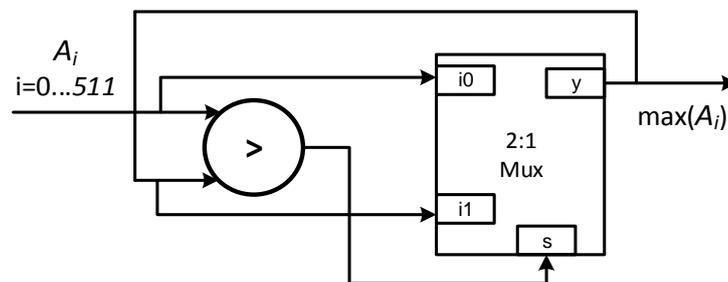


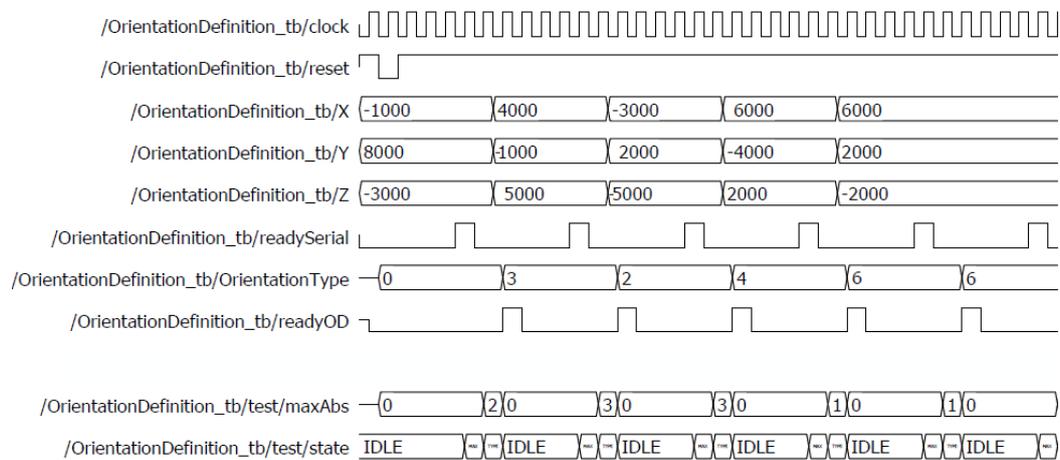
Figure 4.8: Architecture for computing the maximum from incoming data samples.

The maximum acceleration values on respective axes are further compared with a predefined threshold of  $\pm 0.5g$  (cf. Table 4.5) using a comparator module and a multiplexing logic to denote the corresponding orientation state for each sample in the segment. This is in accordance with the algorithm which has been presented in section 4.3.2.

Orientation	Processing
1	maximum acceleration occurs on the Y-axis, is negative, and lies within the range $-g \pm 0.5g$
2	maximum acceleration occurs on the Z-axis, is positive, and lies within the range $g \pm 0.5g$
3	maximum acceleration occurs on the Y-axis, is positive, and lies within the range $g \pm 0.5g$
4	maximum acceleration occurs on the Z-axis, is negative, and lies within the range $-g \pm 0.5g$
5/6	maximum acceleration occurs on the X-axis, is positive, and lies within the range $g \pm 0.5g$ , <i>Orientation 5</i> if the sensor module is worn on the left arm or <i>Orientation 6</i> if worn on the right
0	indicating an unknown position

Table 4.5: Computing logic for orientation states.

A 3-bit orientation state for each incoming data sample is computed on the fly, thereby negating the use of any memory. On successful computation, a 1-bit signal, *readyOD*, is set high which acts as an input flag to the next module, *Sequence Detection*. A simulation snapshot of the *Orientation Detection* module, with a set of tri-axial data samples is shown in Figure 4.9. As is illustrated, the module computes the absolute maximum value among the tri-axial data samples (X, Y and Z) when a *readySerial* signal is detected and infers the orientation (*OrientationType*) based on the aforementioned logic.

Figure 4.9: Simulation of the Orientation Detection module with a sample X, Y and Z inputs (scaled up), illustrating the detected orientation states (*OrientationType*) and the *readyOD* signal.

## 4.7.2 Sequence detection (SD)

Having determined the orientation of the sensor module for each set of tri-axial acceleration data, this module computes a unique sequence of orientation states generated for the performed movement. This module executes each time the *readyOD* signal is set high. A specific orientation is considered part of a sequence only if a continuous set of

orientations of the same type span for more than 13 samples (considering a particular arm position if it lasts for more than quarter of a second). A counter module is used to look through 512 orientation states determined for each data segment pertaining to a performed movement and a comparator is used to compute the changes in orientation states. The simulation waveform of the *Sequence Detection* module is shown in Figure 4.10. The counter operation with respect to the simulation is explained in further detail.

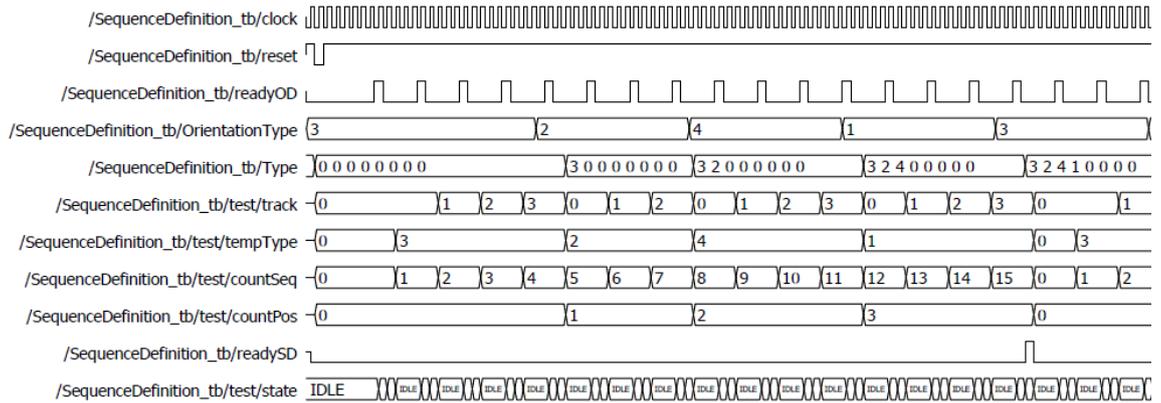


Figure 4.10: Simulation of Sequence Detection with the orientation changes stored in the Type register and the readySD signal.

- A counter *countSeq* increments whenever a *readyOD* signal is detected and therefore counts upto 512 (i.e. all orientation states).
- Each incoming orientation state (*OrientationType*) is stored in a register *tempType* and a sample counter *Track* is incremented until a new orientation appears. This counter is used to check if the present orientation state lasts for more than 13 samples (i.e. quarter of a second). If the counter value exceeds the minimum length threshold, then the *OrientationType* is considered valid and the value stored in *tempType* is stored in the *Type* register bank. Correspondingly, the counter *Track* is reset upon a change in orientation.
- The register bank, *Type* (3-bits  $\times$  8) is used to store up to a maximum of 8 unique orientation states. A counter *countPos* is used to keep a track of the maximum number of unique orientations (i.e. 8) which can be stored in the *Type* register.
- Once *countSeq* reaches 512 (all orientation states), the counter is reset and the signal *readySD* is set high indicating the completion of sequence generation. The unique sequence of orientations is stored in the register bank *Type*.

### 4.7.3 Action detection (AD)

In this module, pre-defined transitions of orientation states (cf. Table 4.2) are looked for within the *Type* register to determine the performed movements. The reverse transitions are also checked since each action involves a reciprocal of the original movement, for example bringing the arm down after raising it to perform a drinking action. The architecture for inferring the movements from the respective orientations in the *Type* register is illustrated in Figure 4.11.

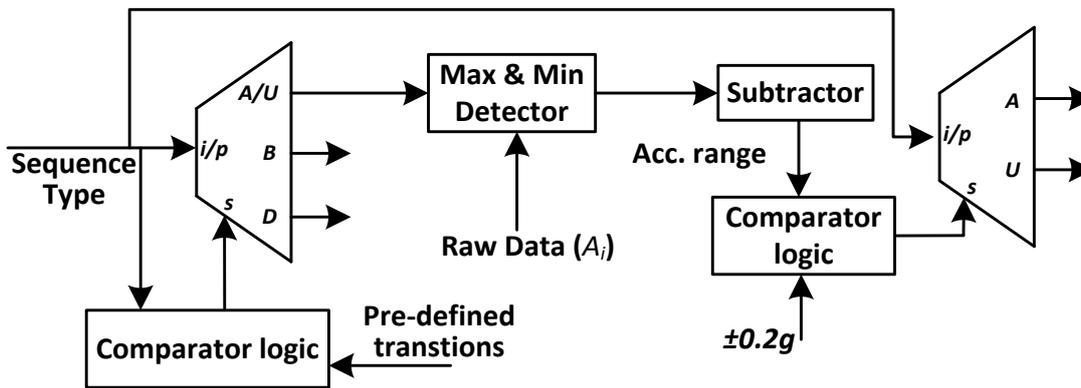


Figure 4.11: Architecture for Action Detection (AD).

*Action B*, involves a transition from Orientation 1 to 5 (left arm) or 3 to 6 (right arm) whereas *Action D*, involves any transitions between orientations 1 to 4. Therefore, a comparator and multiplexing logic are used to infer *Action B* and *D* using the pre-defined transitions, but inferring *Action A* requires additional processing as it can involve different transitions (orientations 1, 2 or 3) or no transition at all as described in section 4.3.4. A subtractor is used to compute the acceleration range (maximum-minimum value) for each orientation sequence and compare it against a pre-defined threshold of  $\pm 0.2g$ , using a comparator logic. The computed acceleration range must be larger than the threshold (indicating movement within the horizontal plane) for the majority of the sequences stored in the register *Type*, otherwise the movement is considered as an *Unknown Action (U)*. A flag *readyAD* is set high once an action has been inferred.

## 4.8 Evaluation

The HDL coding was done using System Verilog and synthesized on the Altera DE2-115 FPGA board, programmed through the USB blaster in Active Serial (AS) mode, providing

non-volatile storage [194]. The synthesized RTL on the FPGA was tested to recognize the movements performed as part of the experimental protocol of ‘*making-a-cup-of-tea*’ (cf. Table 4.1). Test vectors were stored in memory initialisation files (MIF) and the implemented design was tested at 50 MHz. The average accuracy of correctly recognizing the 3 actions over the 4 trials for all healthy subjects is within a range of 85%-96% (cf. Table 4.6) and for all stroke survivors is 63-75% (cf. Table 4.7), representing only slight differences from the corresponding accuracies achieved with the software implementation (cf. Table 4.3 and Table 4.4). The average accuracy dropped by 4.8% for healthy subjects and 6.8% for the stroke survivors. This could be due to the changes in the implemented design where the raw sensor data was not filtered prior to processing.

Subject	Recognised Actions (Out of 20)				Average Accuracy (%)
	Trial 1	Trial 2	Trial 3	Trial 4	
1	18	19	18	18	91
2	16	20	18	18	90
3	18	16	18	16	85
4	18	20	19	20	96

Table 4.6: Recognition of trials for healthy subjects.

Subject	Recognised Actions (Out of 20)		Average Accuracy (%)
	Trial 1	Trial 2	
1	15	15	75
2	12	14	65
3	16	12	70
4	15	10	63

Table 4.7: Recognition of trials for stroke survivors.

The execution time for each module in the worst case is shown in Table 4.8. The variables *lenSeq* is set to 512 (segment length), *lenPos* is set to 8 (number of unique orientation states that can be stored in the register bank *Type*). The *OD* module takes 4 clock cycles for computing the orientation of each data sample. The *SD* computes the *Type* register from a sample length of 512 orientation states in 2050 cycles ( $512 \times 4 + 2$ ) cycles and *AD* takes 10 cycles to infer the performed movement. The design synthesized @50 MHz uses 1804 logic elements and takes 2060 clock cycles ( $\approx 41.2 \mu s$ ) to produce the desired output. The simulation waveform of the top-level module with acceleration values are presented in Figure 4.12. For this simulation, the parameter *lenSeq* has been set to 8 instead of 512 for sake of clarity.

Signal	Module	Clock cycles
<i>readyOD</i>	Orientation Detection	4
<i>readySD</i>	Sequence detection	<i>lenSeq</i> + 1
<i>readyAR</i>	Action Recognition	<i>lenPos</i> + 2

Table 4.8: Execution time for each hardware module, with *lenSeq* = 512 and *lenPos* = 8.

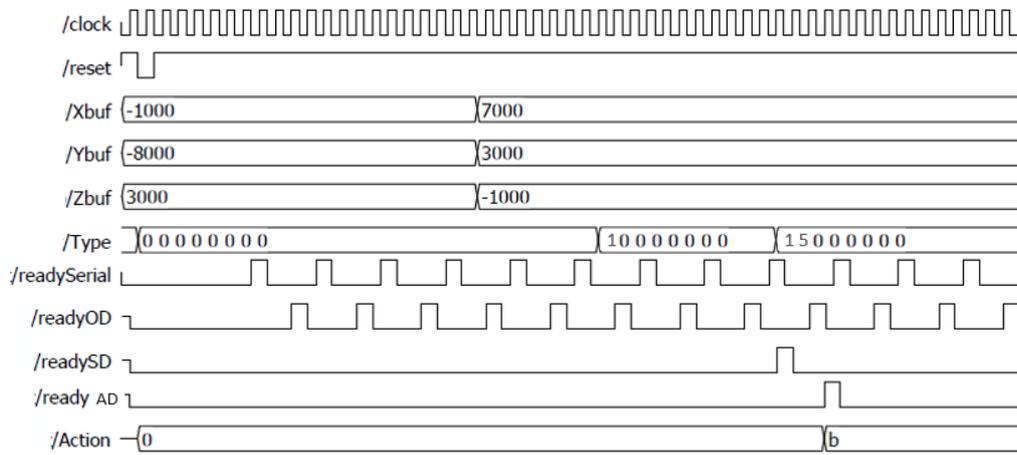


Figure 4.12: Simulation showing the detection of Action B for an orientation transition from position 1 to 5 in the Type register.

A *readySerial* signal which is activated when a set of X, Y and Z values are obtained from the serial receiver. The signal *readyOD* is set high every 4 clock cycles after computing the orientation state for every data sample. A signal *readySD* is set high once the orientation changes are stored in *Type*. Finally, a *readyAD* signal is set high to signify the detected action, *Action B* for a transition in orientation from 1 to 5 (performed with the left arm). In this implementation, the internal RAM was not used since *OD* computes the orientation states on the fly. Furthermore, any multiplication or division was not used in order to minimize the number of synthesized logic elements.

#### 4.9 Sensor interface for real-time movement recognition

An overview of the hardware setup was shown in Figure 4.6. For real-time implementation, the accelerometer transmits data through Bluetooth to a host PC, where the raw sensor data is converted to physical values and transmitted through a RS232 cable to the FPGA board. The RTL implementation of the RS232 receiver and the recognition algorithm were integrated to complete the hardware functionality. The FPGA operates at a much higher frequency (50 MHz) compared to the sensor which streams data at 50 Hz. The application ShimmerConnect was used for the Bluetooth communication between the sensor and the host PC [40]. Using the .NET 4.5 framework, an application software in C# was developed for the serial port control [195]. For transmitting the data from the PC to the FPGA, the baud rate was set to 4800 bits per second, with each set of data being 64-bits wide (16-bits each for X, Y, Z axes and a header code). The header code was used to indicate the start of transmission so that the receiver can determine the correct axes values. On the FPGA, a baud tick generator produces a pulse (based on a counter logic) necessary

for interface synchronization. The LCD screen on the DE2-115 board, using the HD44780 display controller [194], was used to display physical acceleration data derived from the streaming sensor data as shown in Figure 4.13. The recognized arm movements were displayed on a 7-segment display in real-time.

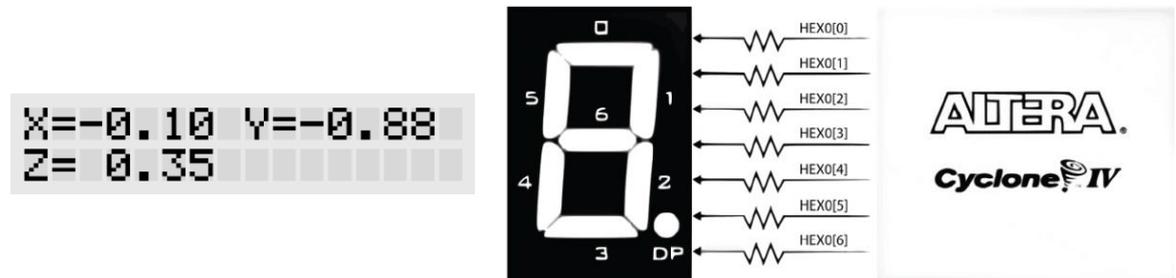


Figure 4.13: LCD display for the acceleration data (left) and 7-segment display connections to the Altera FPGA (right) [194].

For evaluating the system (cf. Figure 4.6) in real-time, the arm movements (*Action A, B* and *D*) were performed multiple times with the sensor worn on the wrist, which were detected successfully and displayed on the LEDs.

## 4.10 Discussion

In this chapter a novel algorithm has been presented to recognize three arm movements by analyzing transitions between six pre-defined orientations of a wrist-worn accelerometer. The algorithm has been developed simply as a proof of concept vehicle for this arm movement recognition application in stroke survivors. The results have been encouraging and show that these particular arm movements can be reliably detected with stroke patients exhibiting moderate levels of involuntary tremor in their movements. The algorithm could be extended for use with patients suffering from other neurodegenerative disorders that might demonstrate greater levels of tremor or less fluidic movement profiles. In such circumstances, additional band-pass filtering can be applied to sensor recordings to suppress higher frequency data associated with rapid or discontinuous movements.

In view of its low-computational complexity, the algorithm was implemented on a reconfigurable platform (i.e. FPGA) for real-time detection of the three arm movements. The synthesized RTL used approximately 1804 logic elements, running at a clock frequency of 50MHz and took 2060 clock cycles to complete thereby taking 41.2  $\mu$ s to generate the desired output. The implemented design does not use any memory element

and avoids the overheads of complex data processing and a dedicated *training* phase, involved in any standard activity recognition system. Although implemented on FPGA, the salient features of the architecture makes it amenable for low-power application in body-worn wireless sensor nodes. The architectural design can be further implemented as a low-power ASIC chip and embedded on a sensor platform along with other vital components such as A/D converter and a de-noising circuit for long-term monitoring of arm movement recognition in real-time.

However, as discussed the six pre-defined orientations and the transitions are particularly aimed at recognising the there investigated arm movements, performed in a horizontal plane. Therefore, this algorithm is not scalable or flexible for incorporating new category of movements which might need to be monitored depending on a patient's impairment and the corresponding guidelines of the clinicians. Hence, in the subsequent chapters the focus is on devising low-complexity activity recognition methodologies which can detect the investigated arm movements in nomadic settings and can be implemented on a low-power hardware platform. Furthermore, upon close examination of the designed activity-list (cf. Table 4.1), it was realised that certain activities in the list (e.g. 3, 6, 8, 10 and 12) could qualify as being denoted *Action C*, since they involve reaching out for an object sideways. These tasks involve a rotation of the arm about the shoulder as originally defined in section 3.2. Hence, in the following chapter where three low-complexity supervised learning algorithms have been explored to recognise the movements performed while '*making-a-cup-of-tea*', the activity-list has been modified to include 5 occurrences each of *Action A*, *B*, *C* and *D*. This also satisfies the movement selection methodology, outlined in section 3.2, derived from the streamlined WMFT set.

## 5. Chapter 5

# Movement Recognition using Supervised Learning Algorithms

### 5.1 Introduction

In this chapter a systematic exploration is presented to recognize the four elementary movements of the upper limb (*Action A, B, C and D* as mentioned in Chapter 3), using data collected through inertial sensors. This involves a detailed description of the experimental protocol for data acquisition from healthy subjects and stroke survivors under two distinct scenarios - *laboratory setup* (controlled environment) and the *semi-naturalistic setup* (uncontrolled environment). As discussed in section 2.2.5.6, in principle there are four steps for human activity recognition using inertial sensors: 1) data capture by appropriate sensors; 2) segmentation of the captured data to identify the beginning and end of an activity, 3) data processing (filtering, feature extraction and selection) and 4) recognition of the activity using appropriate classification techniques. In this chapter, the aim is to detect the number of occurrences of specific arm movements performed by the subjects in the uncontrolled environment. In view of this, the optimal number of sensors, their placement, the best category of sensor signals and the appropriate data processing techniques have also been investigated to enable consistent and accurate detection of these basic arm movements, particularly accounting for temporal and inter-subject variability.

Supervised classification techniques involve two phases – training a model with a given set of observations and evaluating the trained model with new set of observations (testing). Hence, the data collected in the controlled environment is processed to develop the activity recognition model retrospectively following two approaches - a person-dependent (*personalized*) and a person-independent (*generalized*) training dataset. The trained model for both the approaches is cross-validated in association with three supervised learning algorithms independently – linear discriminant analysis (LDA), quadratic discriminant analysis (QDA) and support vector machines (SVM). The trained model (classifier) is then prospectively evaluated on the data collected in the uncontrolled environment in

association with the best performing learning algorithm to detect the performed movements. The overall process is further described in detail in the following section.

## 5.2 Approach

The key steps involved in the overall approach are illustrated in Figure 5.1 and described in detail in the following sections.

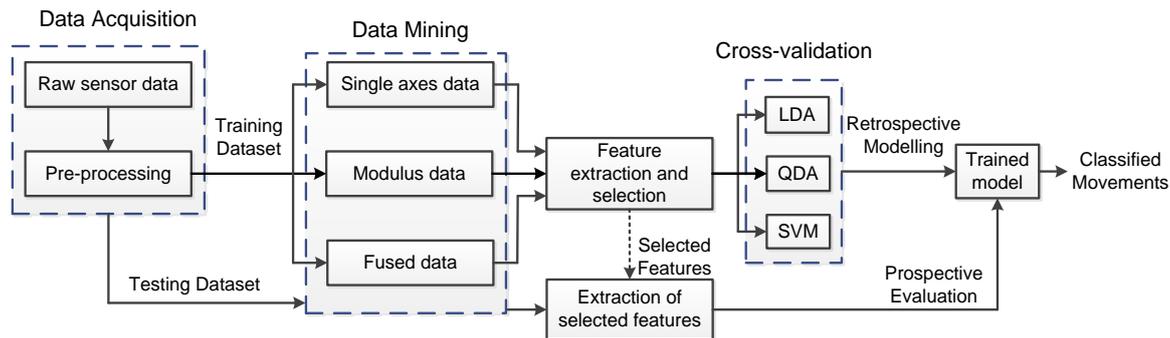


Figure 5.1: Methodology used to evaluate data types and learning algorithms.

### 5.2.1 Data acquisition

The data acquisition step is divided into two phases – sensor data collected through an experiment conducted on healthy subjects and stroke survivors in two distinct scenarios and the pre-processing of the raw sensor data.

#### 5.2.1.1 Experimental setup

In accordance to the requirements of supervised classification, a new experimental setup has been used to collect data for training the classifier in addition to the previously used *seminaturalistic setup* (cf. section 4.2). Therefore, the experimental setup represents two scenarios – *laboratory setup* (controlled environment) representative of the *exercise* phase and the *semi-naturalistic setup* (uncontrolled environment) representative of the *nomadic/daily living* phase.

In the *laboratory setup*, subjects performed multiple trials of the four movements, *Action A*, *B*, *C* and *D* in the laboratory. The subjects were generally encouraged to perform the

tasks in a natural way, as they would normally do when extending, lifting, bending or turning the arm during daily activities. For *Action A*, reaching and retrieving was performed with a full/empty cup, mouse and a paper weight. *Action B* involved lifting a full/empty cup, glass. For *Action C*, subjects reached out for a book, pen, cup kept sideways. Lastly, *Action D* involved rotating the wrist for pouring out from a full/empty cup, glass and a locking/unlocking action. The subjects were asked to perform the actions in a random mix of both sitting as well as standing positions. The subjects were generally encouraged to perform the tasks in a natural way, as they would normally do when extending, lifting, bending or turning the arm during daily activities. In addition, there were no restrictions on the various physical factors of the experiment such as the seating position, height of the chair, distance between the chair and the table, position of the objects on the table and the time required to complete the tasks. Un-constraining the experiment in this manner helps to generate a wider range of variability in the data paving the way for a robust arm movement detection system.

The *semi-naturalistic setup* as already introduced in section 4.2 (cf. Table 5.1) comprises of the archetypal activity-list emulating the process of ‘*making-a-cup-of-tea*’, a common activity performed in daily life. As discussed in section 4.10, the activity-list comprising of 20 individual activities, has been modified to have 5 occurrences each of *Action A*, *B*, *C* and *D* as illustrated in Table 5.1.

	Activity	Action
1.	Fetch cup from desk	A
2.	Place cup on kitchen surface	A
3.	Fetch kettle	C
4.	Pour out extra water from kettle	D
5.	Put kettle onto charging point	A
6.	Reach out for the power switch on the wall	C
7.	Drink a glass of water while waiting for kettle to boil	B
8.	Reach out to switch off the kettle	C
9.	Pour hot water from the kettle in to cup	D
10.	Fetch milk from the shelf	C
11.	Pour milk into cup	D
12.	Put the bottle of milk back on shelf	C
13.	Fetch cup from kitchen surface	A
14.	Have a sip and taste the drink	B
15.	Have another sip while walking back to desk	B
16.	Unlock drawer	D
17.	Retrieve biscuits from drawer	A
18.	Eat a biscuit	B
19.	Lock drawer	D
20.	Have a drink	B

Table 5.1: Use case activity list - ‘Making-a-cup-of-tea’

For this exploration, kinematic data obtained from all the four tri-axial sensor nodes (tri-

axial accelerometer and rate gyroscope – placed on the wrist and the elbow) were considered. Movements performed during the *laboratory setup* are also observed by the researcher to mark the beginning and end time of the activities. Simialr to the *semi-naturalistic setup*, data collected was segmented using the annotations from the accompanying researcher.

The data collected in the *laboratory setup* scenario is used to train the recognition algorithm (classifier/machine) whereby it learns the kinematic characteristics of the specific movements and is later evaluated or tested upon the data collected in the *semi-naturalistic setup*. The primary focus is on detecting the elementary movements used during the archetypal activity of ‘*making-a-cup-of-tea*’. Therefore, in the forthcoming sections the data collected in the *laboratory setup* is referred as *training* phase data or *training* dataset and the data collected in the *semi-naturalistic setup* is referred as *testing* phase data or *testing* dataset. The strategic choice of training the classifier is of utmost importance, given the huge degree of inter-person/temporal variability for the same movement within the human population, and in particular, for people undergoing rehabilitation. Therefore for the *training* phase, sensor data was collected using two types of approaches: a person-independent (*generalized*) and a person-dependent (*personalized*) approach:

- *generalized approach* – kinematic data was collected from a group of 18 healthy subjects in a *laboratory setup*. Each subject performed 20 trials each of *Action A, B, C* and *D*, separated into groups of five repetitions,
- *personalized approach* - kinematic data was collected from five healthy subjects in a *laboratory setup* with each subject performing 120 trials each of *Action A, B, C* and *D*, separated into groups of five repetitions. Four out of these five subjects had already volunteered for the *semi-naturalistic setup* (cf. section 4.2).

Each group of trial comprising of five repetitions for each *action*, was separated by approximately three minutes. This was done to avoid the generation and collection of unrepresentative data due to fatigue and/or boredom, as well as the effects of unconscious self-learning of the activities. At this stage data was collected only from healthy subjects since the intention was to develop the recognition model and evaluate it to deduce if the chosen methodology (sensor positioning, data processing, learning algorithms) were successful in detecting the performed arm movements by a healthy subject population. The data structure for training the model is summarized in Table 5.2.

Attributes	Generalized	Personalized
Number of subjects	18	5
Training dataset	4 actions x 20 times = 80	4 actions x 120 times = 480

Table 5.2: Data structure for each of the four sensor-position combinations for the *training* phase collected from healthy subjects.

### 5.2.1.2 Data pre-processing

The captured raw sensor data was calibrated using the calibration coefficients (sensitivity, offset) obtained for both the accelerometer and the gyroscope sensors, using the simple 3-point calibration methodology discussed in Chapter 3 (cf. section 3.3). The calibrated data is then pre-processed to get rid of any inherent noise and artefacts generally associated with the data acquisition process. The raw sensor data is low-pass filtered with a 3rd order Butterworth filter having a cut-off frequency of 12 Hz to attenuate the high-frequency noise components. The resultant data is passed through a high-pass 3rd order Butterworth filter having a cut-off frequency of 0.1 Hz which attenuates the low-frequency artefacts introduced in the data due to physical effects such as drift [157]. The filter order and cut-off frequency values were experimentally determined using Matlab.

### 5.2.2 Data mining

Data is considered in three forms:

- *Individual sensor signals*: The two Shimmer 9DoF sensor modules transmit data in real-time from a total of 12 individual sensors [(3 x accelerometers and 3 x gyroscopes) x 2 positions], providing a database to search for characterizing patterns.
- *Fused Signals (Modulus)*: modulus of the total acceleration ( $M_a$ ) or total rate of rotation ( $M_g$ ) experienced by the individual limb segments, as given by (5.1). This results in 4 new signals, 2 each for the wrist and elbow sensor modules. Temporal variations in these signals indicate periods of activity of the underlying limb segment.

$$\begin{aligned}
 M_a &= \sqrt{AccX^2 + AccY^2 + AccZ^2} \\
 M_g &= \sqrt{GyroX^2 + GyroY^2 + GyroZ^2}
 \end{aligned}
 \tag{5.1}$$

- *Fused Signals (accelerometer-gyroscope combination)*: Fused signals were further created, based on an *a priori* consideration of the expected trajectory of the subject's arm in relation to the sensor position on the arm and the orientation of the sensor axes when performing the required tasks. For example, Table 5.3 lists the specific accelerometer–gyroscope combinations that are expected to be the most active for each task as a function of their location on the arm. There are 3 unique sensor combinations for the wrist and 2 for the elbow to potentially identify the four tasks. Fusion of these signals takes the simple form of multiplying together the pre-processed data from the appropriate sensor combinations, thus creating 5 unique signals. Fusing data from different sensor nodes were not considered since the aim was to find the minimum number of sensor locations.

Movement	Wrist	Elbow
<b>A</b>	$AccX \times GyroY$	$AccY \times GyroZ$
<b>B</b>	$AccY \times GyroZ$	$AccY \times GyroZ$
<b>C</b>	$AccY \times GyroZ$	$AccY \times GyroZ$
<b>D</b>	$AccZ \times GyroY$	$AccZ \times GyroY$

Table 5.3: Definition of fused signals for each arm movement.

### 5.2.3 Feature extraction

Each data stream exhibit signal patterns that are distinctive for each of the arm movements, which is characterized by a set of features extracted from the signals [157]. In this investigation, 10 time-domain features were considered which are listed in Table 5.4.

No.	Features	Description
1	<i>standard deviation</i>	measure of the variability from the mean of the signal
2	<i>root mean square (rms)</i>	measure of the signal energy normalized by the number of samples
3	<i>information entropy</i>	measure of the randomness of a signal [196]
4	<i>jerk metric</i>	rms value of the derivative of the data normalized with respect to the maximum value of the integral [5]
5	<i>peak number</i>	obtained from gradient analysis of the signal
6	<i>maximum peak amplitude</i>	measure of the amplitude of the peaks obtained after gradient analysis
7	<i>absolute difference</i>	absolute difference between the maximum and the minimum value of a signal
8	<i>index of dispersion</i>	ratio of variance to the mean
9	<i>kurtosis</i>	measure of the 'peakedness' of a signal assuming a non-Gaussian distribution in the data
10	<i>skewness</i>	measure of the symmetry of the data assuming a non-Gaussian distribution in the data [197]

Table 5.4: List of the features considered in this exploration.

Although the last two features (*kurtosis* and *skewness*) are usually associated with defining the shape of a probability distribution, they can still be used as classifying features if they routinely return values that distinguish one pattern of data from another. Hence these 10 one-dimensional features were computed for each movement trial of each subject, considering data from:

- each individual accelerometer ( $AccX$ ,  $AccY$ ,  $AccZ$ ) and gyroscope ( $GyroX$ ,  $GyroY$ ,  $GyroZ$ ) signals for each of the wrist and elbow sensor modules,
- two modulus signals ( $M_a$  and  $M_g$ ) as defined by eqn. (5.1) for each of the wrist and elbow sensor modules,
- five fused data signals (3 for the wrist and 2 for the elbow) as described above in Table 5.3.

#### 5.2.4 Feature selection

Feature selection helps to select the optimal number of features thereby reducing the computational load and helps in achieving the best possible classification accuracy. The extracted features were normalised and as discussed in section 2.2.5.6.4, the Wrapper approach was followed using the sequential forward selection (*sfs*) searching technique. It selects various feature vector combinations to test for the minimal classification error probability and is computationally simple [145]. Here, the selection of the optimal number of features depends strongly on the employed classification algorithm.

The *sfs* technique can be explained with a working example by considering a feature vector comprising of four different features  $[X_1, X_2, X_3, X_4]$ . First, the best ranked feature is computed, say  $X_2$ , and the classification performance is evaluated with  $X_2$ . Secondly, all two-dimensional feature vector combinations with  $X_2$  are computed:  $[X_1, X_2]$ ,  $[X_2, X_3]$ ,  $[X_2, X_4]$  and the classification performance for each of the combinations is evaluated. Thirdly, all three-dimensional feature vector combination with  $X_2$  are computed:  $[X_1, X_2, X_3]$ ,  $[X_1, X_2, X_4]$  and the classification performance is evaluated with both the combinations. Finally, the features forming the best feature vector combination are selected [145].

The best features for a given classification algorithm are selected from a sensor-position specific feature space for each signal type generated as a result of each movement trial performed by each subject.

- $FS_{individual} = [f_{1-X} \cdots f_{10-X}, f_{1-Y} \cdots f_{10-Y}, f_{1-Z} \cdots f_{10-Z}]$

where  $FS_{individual}$ , represents a feature space comprising of 30 features - 10 individual features ( $f_1 \dots f_{10}$ ) extracted from each individual sensor stream (X, Y, Z) for each sensor type (accelerometer, gyroscope) and each position (wrist, elbow),

- $FS_{fused(modulus)} = [f_{1-M} \cdots f_{10-M}]$

where  $FS_{fused(modulus)}$ , represents a feature space comprising of the 10 features extracted from each of the modulus signals ( $M_a, M_g$ ) for each sensor type (accelerometer, gyroscope) and each position (wrist, elbow),

- $FS_{fused(acc-gyro, wrist)} = \begin{bmatrix} f_{1-sig_w1} \cdots f_{10-sig_w1}, f_{1-sig_w2} \cdots f_{10-sig_w2}, \\ f_{1-sig_w3} \cdots f_{10-sig_w3} \end{bmatrix}$

where  $FS_{fused(acc-gyro, wrist)}$ , represents a feature space comprising of 30 features - 10 features extracted from each of the 3 unique signals ( $sig_w1, sig_w2, sig_w3$ ) generated from the specific accelerometer–gyroscope combinations for the wrist sensor node (cf. Table 5.3) ,

- $FS_{fused(acc-gyro, elbow)} = [f_{1-sig_e1} \cdots f_{10-sig_e1}, f_{1-sig_e2} \cdots f_{10-sig_e2}]$

where  $FS_{fused(acc-gyro, elbow)}$ , represents a feature space comprising of 20 features - 10 features extracted from each of the 2 unique signals ( $sig_e1, sig_e2$ ) generated from the specific accelerometer–gyroscope combinations for the elbow sensor node (cf. Table 5.3).

The number of features selected in each of the cases is highlighted in the corresponding results in section 5.4.

### 5.3 Classification using supervised learning techniques

In terms of classification, a review of the literature shows that different machine learning techniques have been used depending on the application area, e.g. Support Vector Machines (SVM) [157], [169], Decision Trees (DT) [28], [157], Naive Bayes (NB) [113], Multi-Layer Perceptron (MLP) [2], Artificial Neural Networks (ANN) [28], or a combination of these techniques [117]. The accuracy of any classification technique will depend on the system requirements covering important areas – type of activities, number of

activities, type of sensors, number of sensors, placement of sensors [38], multiple sensor fusion, etc. Very little work has been reported in terms of activity recognition for elementary arm movements. Further, an aspect which has not been investigated are the differences prevalent among individuals performing the same activities, which is essential considering the variability inherent within a subject population due to various physical factors [29]. Although there are several well-known classification techniques used for human activity recognition as discussed in section 2.2.5.6.5, from the perspective of low or moderate computational complexity, this study was restricted to three supervised learning algorithms – Linear Discriminant Analysis (LDA), Quadratic Discriminant Analysis (QDA) and SVM.

As mentioned in section 2.2.5.6.5, in supervised pattern recognition or supervised learning, the classifier is first trained with a set of data (*training* data) for which the correct class labels are known. During the training of the classifier, the free parameters associated with the classifier are adjusted adaptively to minimize classification errors. The classification performance on the training data can be evaluated using a validation set for which class labels are known, but this data is not used to modify the classifier parameters. Once trained, the classifier is applied to a test dataset where it performs its designated function to determine the most likely condition based on a given data pattern. A good classifier should produce minimum classification error when evaluated on the test data (i.e. the data which it has not been trained upon) [198].

### 5.3.1 Linear/Quadratic Discriminant Analysis

A discriminant function takes an input vector  $X$  and assigns it to one of the  $n$  classes, denoted by  $C_n$ . As the name suggests, linear classifiers use linear decision boundaries, i.e. hyper-planes for separating four or more variables (features). The output predicted by a linear discriminator is given by (5.2).

$$y = \sum_{i=1}^M x_i w_i + b \quad (5.2)$$

where  $y$  is the predicted class,  $M$  is the number of distinct features,  $x_i$  is the  $i^{th}$  feature. Classifier tunable parameters are given by weights ( $w_i$ ) and bias or off-set ( $b$ ), both of which together construct the hyper-plane for separation [198]. The orientation of the

hyper-plane is determined by the weights and the location of the hyper-plane with respect to the origin is determined by the bias.

LDA is widely used due to its powerful discriminatory capability. It generally works on the principle where it maximizes the ratio of intra-class variance to the inter-class variance in any particular dataset thereby maximizing separability between two classes (based on the hyper-plane). If the number of classes is more than two, a natural extension of Fisher Linear discriminant exists using multiple discriminant analysis [199]. The maximization of the ratio (i.e. intra-class variance to the inter-class variance) is done among several competing classes. For  $n$  classes, the intra-class matrix is calculated as:

$$\sum_w = S_1 + \dots + S_n = \sum_{i=1}^n \sum_{x \in c_i} (x - x'_i)(x - x'_i)' \quad (5.3)$$

The inter-class scatter matrix can be computed as:

$$\sum_b = \sum_{i=1}^n m_i (x'_i - x')(x'_i - x')' \quad (5.4)$$

where  $m_i$  is the number of training samples for each class,  $x'_i$  is the mean for each class and  $x'$  is the total mean vector given by:

$$x' = \frac{1}{m} \sum_{i=1}^n m_i x'_i \quad (5.5)$$

The linear transformation,  $\Phi$  to maximize the Rayleigh coefficient, which represents the ratio of the determinant of the inter-class scatter matrix of the projected samples to the intra-class scatter matrix of the projected samples, can be obtained by solving the generalized eigenvalue problem:

$$\sum_b \Phi = \lambda \sum_w \Phi \quad (5.6)$$

Once the transformation,  $\Phi$  is achieved, the classification is then performed in the transformed space based on some distance measure like the Euclidean distance. A new instance  $z$  is classified to  $\mathit{argmin}_n d(z\Phi, x'_n\Phi)$ , where  $x'_n$  is the centroid of  $n$ -th class [199]. Therefore, it is a search for a linear transformation that reduces the dimension of a

$p$ -dimensional statistical model, consisting of  $n$  classes, to  $n-1$  dimensions while preserving a maximum amount of discriminant information in the lower-dimensional model. A representative example of the LDA classifier has been shown in Figure 5.2 which separates two classes by drawing a decision boundary between them in 2-D feature space. The black filled circle and boxes denote the misclassified data points for the two respective classes.

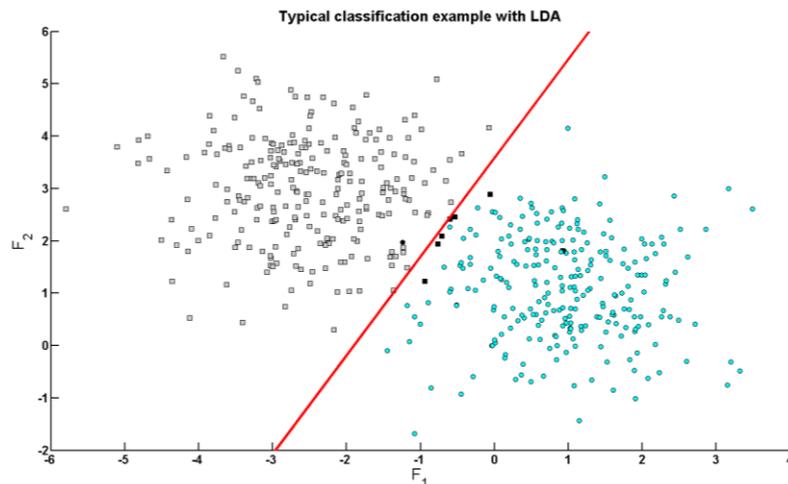


Figure 5.2: Typical example of two-class classification in 2-D feature space using LDA [198].

The LDA classifier gives good result if the data is linearly separable. For many practical biomedical applications, more complex decision boundaries may be required. The easiest way to address this problem is to use higher dimensional kernels. Any input feature vector may be transformed to high dimensional feature space using a polynomial kernel. According to Cover's theorem, it is always possible to obtain a linear boundary or hyperplane if the order of the kernel is gradually increased when the data-set becomes linearly separable [125]. The main problem of this approach is the classifier would not generalize unless the training dataset contains large number of data-points and also results in an increase of the computational complexity. The curse of dimensionality is the main obstacle for using a generalized classifier on limited number of experimental data in majority of classification tasks in biomedical applications. To visualize the effect of higher order kernels, let us consider a 2-D feature vector  $\{x_1, x_2\}$ . A quadratic kernel would map the feature vectors to form a new feature vector as  $\{x_1, x_2, x_1^2, x_2^2, x_1x_2\}$ . Clearly, the new feature vector consists of the original feature vectors, their squared values and their cross-products. Similarly, for a 3-D feature vector  $\{x_1, x_2, x_3\}$ , use of a quadratic kernel will produce a new feature vector as  $\{x_1, x_2, x_3, x_1^2, x_2^2, x_3^2, x_1x_2, x_1x_3, x_2x_3\}$ . Using these higher dimensional features for similar least square based discriminant analysis would produce the well-known quadratic discriminant analysis (QDA) classifier [125].

### 5.3.2 Support Vector Machines

As discussed in section 2.2.5.6.5, the support vector machine is fundamentally a two-class classifier and works by putting special emphasis on the decision boundary. It maximises the distance between those data points (i.e. features) of the training set from both classes that are close to each other as opposed to considering all data points as is common for both LDA and QDA. The points closest to the class separation boundary are called support vectors and the classifier that maximizes the distance between these critical support vectors is known as SVM classifier. Since the SVM classifier maximizes the margin between the classes, it is also known as maximum margin classifier [198]. If the data is linearly separable, the SVM classifier tries to find the optimum hyper-plane which maximizes the margin ( $M$ ) using an optimization routine, subjected to a constraint that all the data points are lying on the appropriate side of the decision boundary. This has been further illustrated with an example of a two-class classification problem in a two-dimensional feature space in Figure 5.3.

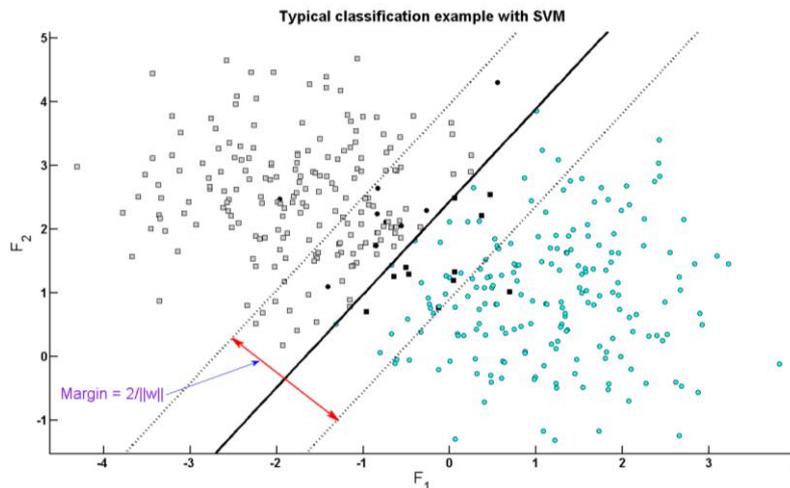


Figure 5.3: Typical example of two-class classification in 2-D feature space using SVM [125].

To simplify the mathematical representation it is considered that the classes are assumed to have a class label of  $y = \pm 1$  which implies the decision boundary to be represented as  $y = 0$ . With input patterns ( $x_i$ ), weight vector ( $w$ ) and the bias ( $b$ ), the decision boundary can be expressed as:

$$y = x_i w + b = 0 \quad (5.7)$$

Therefore, the lines going through the support vectors can be represented as:

$$\begin{aligned} x_i w + b &\geq 1 \quad \text{when } y = +1 \\ x_i w + b &\leq -1 \quad \text{when } y = -1 \end{aligned} \quad (5.8)$$

The above two expressions (5.7) and (5.8), can be combined as follows.

$$y_i (x_i w + b) \geq 1 \quad (5.9)$$

Equation (5.9) signifies that if the parameters  $\{w, b\}$  can be chosen by a suitable optimization framework, the two classes will lie on the appropriate side of the support vectors. The distance of the hyper-plane from the origin can be derived from (5.7), as  $-b/\|w\|$ , where  $\|w\|$  denotes the norm of  $w$  given by:

$$\|w\| = \sqrt{w_1^2 + w_2^2 + \dots + w_n^2} = w^T w \quad (5.10)$$

The distance of the hyper-planes in (5.8) to the origin can be expressed as  $(\pm 1 - b)/\|w\|$ . This can be used to find out the distance between two lines defined by (5.8) simply by subtracting the two distances as shown in (5.11). Hence, the maximum margin ( $M$ ) can be obtained by minimizing  $\|w\|$ .

$$M = \frac{(1-b)}{\|w\|} - \frac{(-1-b)}{\|w\|} = \frac{2}{\|w\|} \quad (5.11)$$

As mentioned in section 2.2.5.6.5, the binary SVM classifier has been extended to handle multi-class problems through the *one-versus-rest* and *one-versus-one* approach. Besides being computationally intensive, there are some inherent problems in both these approaches. In the *one-versus-rest* approach, the training sets are imbalanced and in the *one-versus-one* it takes a significantly longer training time [200]. Therefore, in this exploration the toolbox LIBSVM was used, which is a library for SVM and is efficient for multi-class classification [201]. The linear SVM is effective if the data is linearly separable. For data which are not linearly separable, i.e. linearly non-separable features are often mapped to a high dimensional feature space where they become linearly separable (according to Cover's theorem). In most real world applications, SVM is commonly augmented with complex kernels like Radial Basis Function (RBF) which creates a complex decision boundary based on a higher dimensional feature mapping [125].

### 5.3.3 Classifier performance evaluation

Overall (average) correct classification or accuracy is generally used to measure the performance of a binary classifier which might not always be applicable for multi-class classification because of possible dissimilar classification rates of different classes affecting the overall performance measure. Hence the sensitivity of a given class is measured from the confusion matrix  $N$  following the scheme proposed in [202]. The sensitivity  $S$  of class  $i$  estimates the number of patterns correctly predicted to be in class  $i$  with respect to the total number of patterns in class  $i$  [202]:

$$S_i = \frac{N_{ii}}{f_i} \times 100 \quad (5.12)$$

$$f_i = \sum_{j=1}^c C_{ij} \quad (5.13)$$

where  $i = 1 \dots c$  and  $c$  is the total number of classes. The diagonal and the off-diagonal elements of the confusion matrix correspond to correctly classified and misclassified patterns respectively.  $C_{ij}$  represents the number of times that the patterns are predicted to be in class  $j$  when they really belong to class  $i$ . The overall accuracy of movement detection for each subject is represented by the sum of individual class sensitivities (success rate of individual movements) with respect to the total number of test patterns to be classified. A sample confusion matrix  $N$  is shown in Figure 5.4.

	Predicted 'A' $j = 1$	Predicted 'B' $j = 2$	Predicted 'C' $j = 3$	Predicted 'D' $j = 4$
Actual 'A', $i = 1$	0.95	0.05	0	0
Actual 'B', $i = 2$	0	0.9	0	0.1
Actual 'C', $i = 3$	0.02	0	0.98	0
Actual 'D', $i = 4$	0.05	0	0.05	0.9

Figure 5.4: An example confusion matrix for four classes.

This example shows near perfect classification since all diagonal elements approach unity and all off-diagonal elements approach zero. Therefore, the sensitivity of class A (expressed as a percentage), can be computed as  $S_A = \frac{0.95}{(0.95+0.05)} = 95\%$  and the overall

accuracy (expressed as a percentage) can be computed as  $Accuracy = \frac{(0.95+0.9+0.98+0.9)}{4} = 93.25\%$ . These two metrics – sensitivity and overall accuracy have been used to report the

classification performance in this exploration for movement recognition.

## 5.4 Cross-validation on the training dataset

As mentioned in section 5.2.1.1, the recognition model was developed retrospectively using the data collected in the *laboratory setup* (*training* dataset) using two types of approaches: *generalized* and *personalized* in association with the learning algorithms LDA, QDA and SVM. The model in either approach is verified through cross-validation where a segment of the original dataset is kept for training the model and the rest of the data for evaluating the trained model. This process is repeated till whole of the original dataset has been tested (or cross-validated). Feature extraction from the pre-processed sensor data and classification were performed in Matlab.

### 5.4.1 Generalized approach

The fundamental assumption behind this approach is that if a pool of data encompassing large variability of a particular type of movement from a population is used to train a classifier then it would be able to successfully identify that particular type of movement for a single subject as there is very high probability that the characteristics of the movement of that subject is already embedded within the *training* dataset. To test this hypothesis, as shown in Figure 5.5, a ‘leave-one-subject-out’ validation methodology was performed, wherein one subject was left out of the training data set [128]. This process was repeated for all 18 subjects. Since each subject performs a movement 20 times, for each sensor-position and signal category, a data set consisting of 1440 samples (18 subjects  $\times$  20 trials  $\times$  4 movements) was obtained. One subject’s data of 80 samples (1 subject  $\times$  20 trials  $\times$  4 movements) was used as the *testing* set and the remaining 1360 samples as the *training* set in each iteration to evaluate each of the three classifiers (using LDA, QDA and SVM algorithms) for data from each sensor-position (accelerometer and gyroscope on the wrist and elbow) and each signal category (individual, fused-modulus and fused-combination).

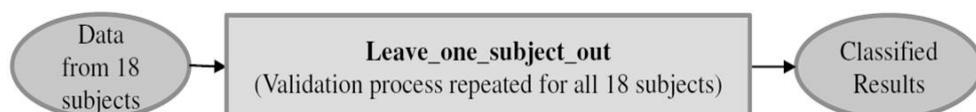


Figure 5.5: Overview of the generalized classification approach.

## 5.4.2 Results for the Generalized approach

The classification results (sensitivity for each arm movement recognised) of the *generalized approach* using the individual sensor data, their moduli and the fused data for each of the learning algorithms LDA, QDA and SVM for the wrist and elbow are presented in Figure 5.6 and Figure 5.7 respectively. The sensitivities and the number of features required for each case are also presented in Table 5.5.

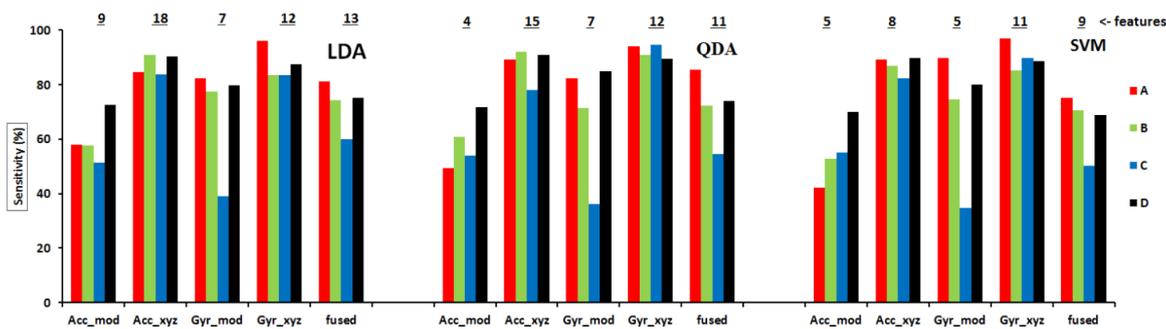


Figure 5.6: Generalized classification - sensitivities for each task using the wrist accelerometer and gyroscope modulus signal (mod), individual sensor signals (X, Y, Z) and fused signals with LDA, QDA and SVM. The number of features required for each signal and sensor type is shown at the top of each group.

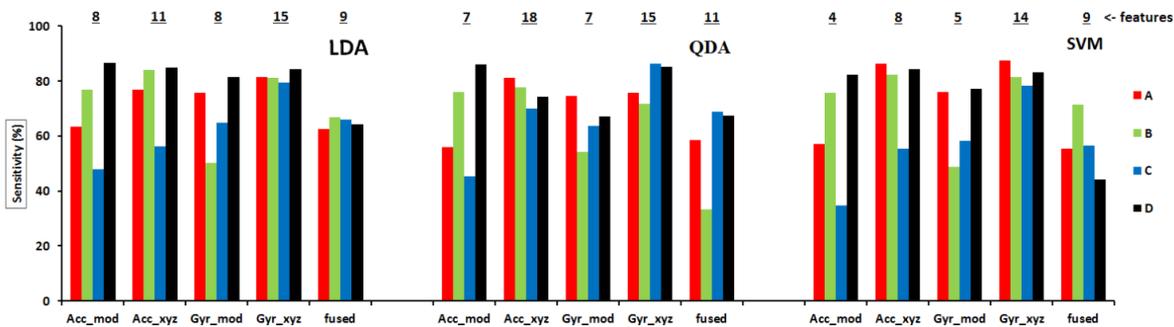


Figure 5.7: Generalized classification - sensitivities for each task using the elbow accelerometer and gyroscope modulus signal (mod), individual sensor signals (X, Y, Z) and fused signals with LDA, QDA and SVM. The number of features required for each signal and sensor type is shown at the top of each group.

Considering LDA with individual sensor signals, the wrist gyroscope recognises the four movements with sensitivities in the range of 83-96% across all movements while the wrist accelerometer also has a similar detection rate with sensitivities in the range of 84-91% across all movements. However, the gyroscope uses only 12 features as compared to the 18 used by the individual sensor signals of the accelerometer (out of a total of 30 -  $3 \times 10$  features) and hence is the obvious choice with regard to a low complexity solution. As can be observed *Action B* has a higher sensitivity (91%) using the accelerometer but that involves a cost of computing 6 extra features.

Classifier	Signal	Wrist					Elbow				
		A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
LDA	Acc_mod	58	58	51	73	9	63	77	48	87	8
	Acc_xyz	85	91	84	90	18	77	84	56	85	11
	Gyr_mod	82	78	39	80	7	76	50	65	81	8
	Gyr_xyz	96	83	83	88	12	81	81	79	84	15
	Fused	81	74	60	75	13	63	67	66	64	9
QDA	Acc_mod	49	61	54	72	4	56	76	45	86	7
	Acc_xyz	89	92	78	91	15	81	78	70	74	18
	Gyr_mod	82	71	36	85	7	74	54	64	67	7
	Gyr_xyz	94	91	95	89	12	76	72	86	85	15
	Fused	86	72	54	74	11	59	33	69	68	11
SVM	Acc_mod	42	53	55	70	5	57	76	35	82	4
	Acc_xyz	89	87	82	90	8	86	82	55	84	8
	Gyr_mod	90	74	35	80	5	76	49	58	77	5
	Gyr_xyz	97	85	90	89	11	88	81	78	83	14
	Fused	75	71	50	69	9	55	71	56	44	9

Table 5.5: Generalized classification results using LDA, QDA and SVM.

A further comparison of the wrist gyroscope results using individual sensor signals with QDA and SVM illustrates that the results for QDA and SVM are marginally higher than LDA, and the number of features required for the three algorithms is nearly the same. Hence, in view of the trade-off between the sensitivity and the complexity involved, LDA being computationally less complex [36] appears as the best choice. The sensitivities achieved using the individual signals from the elbow gyroscope for LDA (A: 81%, B: 81%, C: 79%, D: 84%), QDA (A: 76%, B: 72%, C: 86%, D: 85%) and SVM (A: 88%, B: 81%, C: 78%, D: 83%) are lower than those achieved with the individual signals from the wrist gyroscope LDA (A: 96%, B: 83%, C: 83%, D: 88%), QDA (A: 94%, B: 91%, C: 95%, D: 89%) and SVM (A: 97%, B: 85%, C: 90%, D: 89%). Considering fused signals from the wrist sensors, the sensitivity falls within 60-81% for the four movements with LDA and lies within 54-86% with QDA and 50-75% with SVM whereas for the elbow the sensitivities for the fused signals are within 60-81% with LDA, 33-69% with QDA and 44-71% with SVM. In fact, a close examination reveals that in general the sensitivity for each movement for the signals from the wrist are better than those from the elbow, which is because the wrist is expected to produce the largest sensor response to the arm movements being investigated.

It can be observed that the sensitivity for each movement using the individual sensor signals for both the accelerometer and the gyroscope placed on the wrist and elbow is better than the modulus and the fused signals. The difference in the recognitions rates between modulus and individual signals is due to the fact that for individual sensor signals any bipolar information present in the raw data is retained, whereas the generation of a modulus signal creates, by definition, only unipolar data. Hence, using the individual sensor signals provides the classifier an opportunity to select the optimum number of features from a wider pool of features and hence the recognition rate for the movements is reflected in the higher sensitivities achieved. For the fused signals (accelerometer-gyroscope combinations) the sensitivity is generally lower when compared to results obtained from individual sensors, but better than results obtained when considering the moduli signal.

### 5.4.3 Personalized approach

In contrast to the generalized classification methodology, the basic hypothesis in the personalized approach is that the movement patterns have characteristic associations with specific subjects which may not be possible to capture in a generalized scenario. *Personalized* approach is a further testimony to the fact that each person undergoing any sort of rehabilitation will have different forms and levels of impairment and thus would be prescribed different exercises which would pertain to classifying individual movements. Therefore, a classifier based on the *training* dataset of the movement data of a subject (in a person-centric way) may yield more accurate classification results for that specific subject. The main steps for developing the *personalized* classification strategy are shown in Figure 5.8. To test this hypothesis five healthy subjects were asked to perform the same four movements 120 times each under the same experimental conditions. The collected dataset from a subject is labelled as the training database specific to that particular subject and 10 runs of 10-fold cross validation are carried out on the data collected for each subject. The cross-validation process creates 10 segments of the data sample (120 samples for each *action*) with each segment having 12 samples. In each run of the stipulated 10 runs, one segment is used as the testing set while the rest of the 9 segments are used as the training set. The whole process is repeated for each subject as shown in Figure 5.8, using data from each sensor-position (accelerometer and gyroscope on the wrist and elbow) and signal category (individual, fused-modulus and fused-combination).

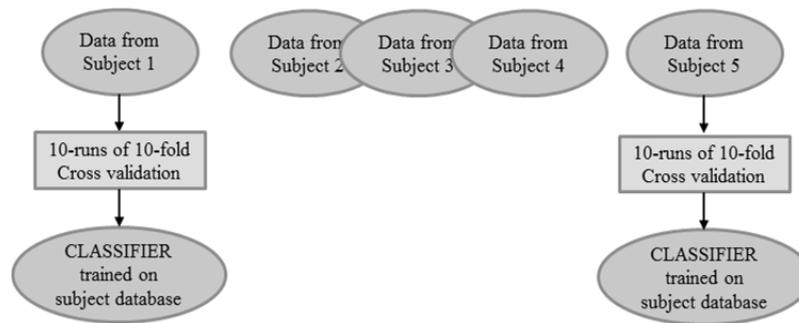


Figure 5.8: Overview of the personalized classification approach.

#### 5.4.4 Results for the Personalized approach

Having established the effectiveness of the individual sensor signals over the moduli and fused signals, the classification results (sensitivity for each task) for the *personalized* approach comparing five healthy subjects using the individual signals for each sensor type and position and the three learning algorithms are presented in Table 5.6 - Table 5.8. In general, all the three classifiers (LDA, QDA and SVM) applied on data from all sensor-position combinations give high levels of classification results across all tasks (above 90%). For sake of brevity, a comparative illustration of the classification results using the individual signals from only the gyroscope wrist has been presented in Figure 5.9.

		Wrist					Elbow				
Sensor	Subject	A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	99	100	100	98	7	100	100	100	99	7
	Subject 2	100	100	100	99	3	100	99	96	98	5
	Subject 3	98	99	97	99	7	98	100	97	99	7
	Subject 4	94	100	96	99	7	94	97	96	98	9
	Subject 5	100	100	100	99	5	98	99	98	100	5
Gyr	Subject 1	100	100	100	100	6	100	99	92	98	8
	Subject 2	100	100	99	100	4	100	100	100	100	5
	Subject 3	98	100	99	99	5	100	100	99	99	7
	Subject 4	98	100	99	99	6	98	99	93	97	7
	Subject 5	99	100	100	98	7	99	99	97	100	10

Table 5.6: Personalized classification results with individual sensor signals using LDA.

		Wrist					Elbow				
Sensor	Subject	A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	100	100	100	98	7	95	99	98	98	5
	Subject 2	100	100	99	99	3	100	98	100	100	4
	Subject 3	99	99	99	99	7	99	99	100	99	5
	Subject 4	94	98	81	91	7	79	95	84	96	5
	Subject 5	100	100	100	100	5	98	98	99	99	5
Gyr	Subject 1	99	100	100	100	6	99	98	99	96	7
	Subject 2	100	100	100	100	3	99	100	100	100	5
	Subject 3	98	98	100	99	6	100	100	100	98	9
	Subject 4	99	100	100	98	6	100	97	100	100	7
	Subject 5	98	99	99	98	5	98	100	98	100	8

Table 5.7: Personalized classification results with individual sensor signals using QDA.

		Wrist					Elbow				
Sensor	Subject	A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	99	100	100	99	7	100	95	99	98	6
	Subject 2	100	100	100	99	3	100	99	98	99	4
	Subject 3	98	100	97	99	6	98	100	98	98	8
	Subject 4	91	100	90	99	6	93	98	90	99	10
	Subject 5	100	100	100	100	5	99	100	98	100	5
Gyr	Subject 1	100	100	100	100	7	100	99	92	98	8
	Subject 2	100	99	100	100	5	100	100	100	100	5
	Subject 3	99	100	100	100	7	99	99	100	99	8
	Subject 4	98	100	100	98	7	93	98	93	99	10
	Subject 5	98	99	98	98	6	94	98	93	99	5

Table 5.8: Personalized classification results with individual sensor signals using SVM.

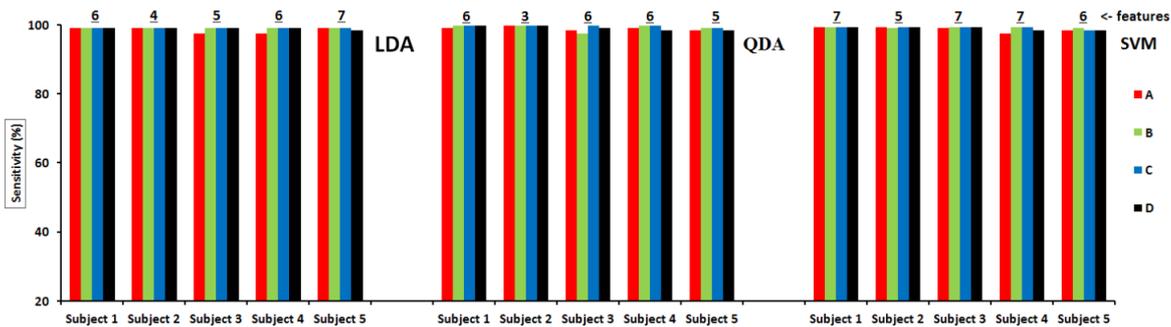


Figure 5.9: Personalized classification – sensitivities for each task using the wrist gyroscope individual signals, with LDA, QDA and SVM for each of the 5 subjects. The number of features required for each subject is shown at the top of each group.

#### 5.4.5 Conclusions from the cross-validation step

A systematic exploration has been made towards developing a robust training model based on a group of subjects doing similar movements and in a subject specific manner to cater to inter-subject variability. The model is verified using cross-validation methodologies with attention on the selection of sensor type, position, and appropriate classification strategies for detecting four fundamental types of upper limb movements that are used in daily life activities. The following conclusions can be drawn from this exploration.

- For the *generalized* approach, it is observed that the sensitivity for each movement for the signals from the wrist are better than those from the elbow. The accelerometer and the gyroscope placed on the wrist can classify all the four movements with accuracy in the range 83-96% when data from individual sensors is used with LDA as the learning algorithm. However, the number of features required to achieve it is on the higher side (12-18) as compared to the *personalized* approach which implies higher computation involved during feature extraction. This can be partly explained by the fact that the classifier requires more feature-specific information to cater to the wider variability inherent in the *generalized* database as compared to the *personalized* approach where there is a high degree of repeatability in the tasks performed by each individual subject and hence can be represented by fewer features.
- Better results are achieved with the individual sensor signals as compared to using the modulus of the accelerometer and gyroscope signals, or the fused signals. Using all the individual sensor signals, rather than a single processed signal (i.e. moduli or fused), provides the classifier an opportunity to select from a wider pool of features and hence the recognition rate for the movements is reflected in the higher sensitivity achieved.
- In the *personalized approach*, exploration with only the individual signals have been presented, and the results demonstrate that LDA gives comparable results when compared with more computationally intensive classification methods such as QDA and SVM and hence is a better choice. The LDA algorithm can independently classify all the four movements with sensitivity in the range 92-100% using a small set of features (6-10) extracted from a tri-axial accelerometer or tri-axial rate gyroscope placed near the wrist or elbow. Therefore any of these two types of sensor or locating positions can be used for the target classification of the *testing phase* data collected in the *semi-naturalistic setup*.

Hence, having established their supremacy, the LDA learning algorithm is used in

conjunction with the *personalized* approach considering only individual sensor signals for the prospective evaluation of the trained model on the data collected from the subjects during the archetypal activity of ‘*making-a-cup-of-tea*’. Furthermore, the *personalized approach* would be beneficial when applied to monitoring individual patients who demonstrate differences in levels of impairment depending on their stage of rehabilitation. Hence, it is essential for subject specific training especially for tracking activities that are susceptible to individual and temporal variation [39].

## 5.5 Evaluation of the trained model - prospective study

Recognition strategies generally follow one of three themes. Firstly, using only data collected under controlled conditions (e.g. in the laboratory) for training as well as testing, which results in high accuracies [28]. Secondly, using both controlled and un-controlled data (e.g. out-of-laboratory) for both training and testing, which results in reasonably high accuracies [27], [39]. Finally, using controlled data for training and only un-controlled data for testing, which generally results in lower accuracies but is more realistic of real-world applications [27], [39].

In this work, controlled data was used for *training* and un-controlled data was used for *testing* in order to explore the levels of recognition accuracy for a robust classification mechanism applicable in the field of home based rehabilitation. The data collected in the laboratory is representative of the scenario where patients are instructed to follow a particular exercise regime involving the impaired arm in a controlled environment (clinic or home) and are later monitored to track occurrences of these specific movements while they perform daily activities with their impaired arm, facilitating a measure of rehabilitation progress in natural settings.

### 5.5.1 Healthy subjects

Data collected (i.e. the *testing* dataset) in the *semi-naturalistic* experiment session (cf. section 4.2), which was performed on a separate day from the *laboratory setup* experiments, was used to evaluate the trained *personalized* model in a prospective manner (the trained model has not been validated on this dataset) to classify the movements performed by each subject during multiple trials of the archetypal activity, ‘*making-a-cup-*

*of-tea*'. The data structure used for this analysis is summarized in Table 5.9.

Attributes	Description
Number of subjects	4
Training dataset	4 actions x 120 times = 480
Testing dataset	[4 actions x 5 times] x 4 trials = 80

Table 5.9: Data structure for prospective evaluation for each of the four sensor-position combinations using individual sensor signals.

The data from the *testing* phase is pre-processed (cf. section 5.2.1.2) and only those features are extracted from each test vector (i.e. each movement trial), which were already selected during the cross-validation of the *personalized* training model (cf. Table 5.6). The results are presented for all the four sensor-position combinations: *acc\_wrist*, *acc\_elbow*, *gyro\_wrist* and *gyro\_elbow* along with the number of features required for each subject in Table 5.10.

Sensor	Subject	Wrist					Elbow				
		A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	100	100	10	75	7	100	0	0	0	7
	Subject 2	90	95	0	15	3	5	90	85	0	5
	Subject 3	0	100	0	10	7	50	50	20	60	7
	Subject 4	90	100	0	100	7	0	100	0	60	9
Gyr	Subject 1	95	95	65	100	6	100	0	0	0	8
	Subject 2	0	100	0	5	4	0	0	0	45	5
	Subject 3	60	100	100	55	5	70	50	40	50	7
	Subject 4	30	95	80	60	6	75	80	100	60	7

Table 5.10: Personalized classification results with individual signals using LDA for healthy subjects

It is evident from Table 5.10 that using individual signals from the accelerometer and the gyroscope on the wrist, relatively high recognition results are achieved but not across all subjects and *actions*. For none of the subjects, all the *actions* are classified up to a satisfactory level (>60%) except for subject1 with signals from the gyroscope wrist. However, it is interesting to note that for these subjects, the results from the cross-validation of the *training* data yielded high sensitivities (cf. Table 5.6). But when the trained model is used to classify the data collected in the *semi-naturalistic* setup, the sensitivities are quite low. This is a typical problem with many classifiers, where they perform well on the *training* dataset but perform poorly on the *testing* dataset or for real-world applications (data on which it has not been trained). Hence, it can be inferred that the

learnt model or the classifier is poorly generalized because it cannot perform well on data collected in a different scenario.

### 5.5.2 Stroke survivors

Having done an initial exploration with healthy subjects which involved the development of the recognition model following a *personalized* approach, finding the optimal sensor type, their position, the best signal category and an evaluation of the trained model, this methodology was further evaluated on movements performed by stroke survivors. Experiments were performed with four stroke survivors (who had volunteered for the *semi-naturalistic* setup, cf. section 4.2) within the same treatment centre at BBK, Germany, under the supervision of the expert physiotherapist members of the research team, to generate the *training* phase data. The four stroke survivors performed – 40 trials each of *Action A, B, C* and *D* (subjects 1 and 4) and 20 trials each of *Action A, B, C* and *D* (subjects 2 and 3) separated similarly into groups of five repetitions for each action in the *laboratory setup*, following the protocol mentioned in section 5.2.1.1. The tasks were performed only whilst they felt comfortable to do so giving rise to the disparity in the number of trials. Similar to the healthy subjects data was obtained from four tri-axial sensor nodes (tri-axial accelerometers and tri-axial rate gyroscopes attached to the wrist and elbow of the impaired arm of each patient).

Similar to the healthy subjects, the data from the *training* phase is pre-processed and used to train an activity model retrospectively in a *personalized* manner in conjunction with the LDA algorithm using only individual sensor signals. The trained model is evaluated on the *testing* dataset prospectively, to classify the movements performed by each subject while ‘*making-a-cup-of-tea*’. The data structure used for this analysis is summarized in Table 5.11.

Attributes	Description
Number of subjects	4
Training dataset	4 actions x 40 times = 160 (patient 1 and 4) 4 actions x 20 times = 80 (patient 2 and 3)
Testing dataset	[4 actions x 5 times] x 2 trial = 40

Table 5.11: Data structure for prospective evaluation involving stroke survivors for each of the four sensor-position combinations using individual sensor signals.

The results for each subject from the cross-validation stage (*training*) for all the four sensor-position combinations: *acc\_wrist*, *acc\_elbow*, *gyro\_wrist* and *gyro\_elbow* along with the number of features required for each subject are presented in Table 5.12 and the results from the prospective evaluation (*testing*) are presented in Table 5.13.

		Wrist					Elbow				
Sensor	Subject	A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	80	100	58	100	5	93	80	73	83	6
	Subject 2	100	100	90	100	5	95	85	80	90	10
	Subject 3	95	85	75	100	5	75	85	95	95	10
	Subject 4	94	100	96	99	10	94	97	96	98	4
Gyr	Subject 1	73	90	65	95	6	88	85	83	90	4
	Subject 2	95	95	95	90	6	100	85	85	90	8
	Subject 3	100	90	90	95	6	95	90	90	90	9
	Subject 4	98	100	99	99	8	98	99	93	97	5

Table 5.12: Cross-validation of the *training* dataset using a *personalized approach* with individual signals using LDA for stroke survivors.

		Wrist					Elbow				
Sensor	Subject	A (%)	B (%)	C (%)	D (%)	Features	A (%)	B (%)	C (%)	D (%)	Features
Acc	Subject 1	10	30	10	20	5	10	20	20	20	6
	Subject 2	0	0	0	50	5	10	10	30	40	10
	Subject 3	20	10	50	50	5	10	50	0	40	10
	Subject 4	10	0	40	40	10	20	30	10	10	4
Gyr	Subject 1	30	20	10	20	6	20	10	30	10	4
	Subject 2	20	10	10	50	6	10	10	10	50	8
	Subject 3	10	0	50	10	6	0	50	10	40	9
	Subject 4	40	20	10	30	8	0	40	10	20	5

Table 5.13: Prospective evaluation of the *testing* dataset with individual signals using LDA for stroke survivors.

From Table 5.12 and Table 5.13, it is evident that for the stroke survivors the trends are quite similar to that of the healthy subjects. Although the sensitivities for each arm movement are quite high in the retrospective cross-validation of the *training* dataset, especially using signals from the wrist-worn accelerometer and gyroscope, but have low sensitivities for each action during the prospective evaluation. When evaluated on the *testing* dataset, the learnt model failed to classify the actions up to an acceptable level. The maximum sensitivity obtained for any action was 50% (cf. Table 5.13) and for none of the subjects, all the four movements are classified with a sensitivity of at least 50%.

## 5.6 Discussion

The classification results obtained for the four healthy subjects and stroke survivors was low, with none of the subjects having all four actions classified with sensitivities up to a minimal threshold (>60%). This reflects a poorly generalized learning model having high sensitivities during retrospective modelling but having low sensitivities when evaluated on datasets collected in a different experimental condition (prospective evaluation). This can also be partly explained by the fact that there is a qualitative difference between the arm movements performed in the *laboratory setup* (*training* dataset) and the *semi-naturalistic setup* (*testing* dataset). Although the subjects performed the movements in the *laboratory-setup* in a natural manner with considerable variability, the movements were performed in a constrained environment within the laboratory emulating the *exercise* phase of patients undergoing rehabilitation within the home environment. This was in contrast to the *semi-naturalistic* movements performed during the ‘*making-a-cup-of-tea*’ session by the subjects in the kitchen. Certain actions of the activity-list (cf. Table 5.1) like ‘switching the power on/off for the kettle’ (no. 6, 8); ‘retrieve biscuits from the drawer’ (no. 17); ‘eat a biscuit’ (no. 18) were quite different to those performed in the *training* set. However, this is in accordance with the application scenario which as discussed in section 5.5, comprises of the patient performing a set of prescribed exercises for a stipulated amount of time within a designated space (e.g. exercise platform at clinic or home). They are later monitored to track the occurrences of those characteristic movements (e.g. rotation of the arm about the elbow) performed in real-life, thereby facilitating a quantification of the usage of the impaired arm in daily activities.

This can be further analysed by looking at the classification results of one of the subjects at a granular level. The confusion matrix shown in Figure 5.10 represents the actual/predicted activity labels for Subject 4 (stroke survivor) as a result of classification (LDA) using the gyroscope data from the wrist. This subject was selected because of being at an early stage of rehabilitation and low on the functional ability scale.

	Predicted ‘A’	Predicted ‘B’	Predicted ‘C’	Predicted ‘D’
Actual ‘A’	4	0	6	0
Actual ‘B’	7	2	0	1
Actual ‘C’	8	0	1	1
Actual ‘D’	4	0	3	3

Figure 5.10: Classification results using LDA with data from wrist gyroscope for Subject 4 (stroke survivor).

Table 5.14 further lists the actual and predicted labels as a result of the classification (LDA) using data from the wrist gyroscope, for each of the 20 activities performed by Subject 4 over the two trials.

	Activity	Action	Predicted1	Predicted2
1.	Fetch cup from desk	A	A	A
2.	Place cup on kitchen surface	A	C	C
3.	Fetch kettle	C	C	A
4.	Pour out extra water from kettle	D	A	A
5.	Put kettle onto charging point	A	C	C
6.	Reach out for the power switch on the wall	C	A	A
7.	Drink a glass of water while waiting for kettle to boil	B	A	A
8.	Reach out to switch off the kettle	C	A	A
9.	Pour hot water from the kettle in to cup	D	A	A
10.	Fetch milk from the shelf	C	A	A
11.	Pour milk into cup	D	D	C
12.	Put the bottle of milk back on shelf	C	A	A
13.	Fetch cup from kitchen surface	A	A	A
14.	Have a sip and taste the drink	B	A	A
15.	Have another sip while walking back to desk	B	A	A
16.	Unlock drawer	D	C	C
17.	Retrieve biscuits from drawer	A	C	C
18.	Eat a biscuit	B	A	D
19.	Lock drawer	D	D	D
20.	Have a drink	B	B	B

Table 5.14: Activity-list showing actual and predicted action labels for Subject 4 (stroke survivor).

It can be observed from Figure 5.10 and Table 5.14 that majority of the confusions or false classifications arise due to the *Actions A* and *C* which are used interchangeably in daily life. *Action A* has been correctly classified four times and falsely classified six times as *Action C*. It can be observed that activities 1 and 13 were correctly classified which bear a closer resemblance to the movements performed in the training set (reach and retrieve a cup/mouse). However, activities 5 and 17 involving a kettle and a biscuit were misclassified which could be due to the fact that a kettle was a heavier object and grasping patterns and wrist/forearm orientations would be quite different to fetching a cup.

Similarly for *Action B*, two of them were correctly classified whereas seven of them were classified as *Action A* and one as *Action D*. Activity 20 involving a sitting and drinking was correctly classified whereas activities 7 and 15 were misclassified as *Action A*. Activity 7 involves a reaching out action prior to a drink whereas activity 15 involves a drinking action while walking. On similar lines, activity 18 involving ‘eating a biscuit’ has considerable difference to a ‘lift cup to mouth action’ (as done in training), was misclassified as *Action A* and *D*. *Action C* had the least number of correctly classified instances out of the 10 occurrences with a majority of them being mis-classified as *Action A*. Activities 10 and 12 involving fetching and keeping back a milk bottle bear a close

resemblance to a reach and retrieve action (*Action A*). Similarly activities 6 and 8 involving a switch have considerable difference to the ‘reach out for an object sideways’ action performed during the training phase.

Lastly, *Action D* was correctly classified for the instances (activities 11 and 19) bearing a close resemblance to the training movements whereas the other instances like pouring out water from kettle were mis-classified. This could again be due the differences in the objects handled during training and testing, inducing a difference in grasping and forearm orientation especially for a subject having poor arm functional abilities.

These observations clearly indicate that activities performed during the ‘*making-a-cup-of-tea*’ session which had a considerable deviation from the training set were misclassified to a large extent. This augurs for devising a *training* set having activities which are a closer match to the ones being classified in the *testing* set. However, as already mentioned in this application scenario the aim is to learn the patterns of specific arm movements (resembling prescribed exercises) and detecting the occurrence of such movements in daily life activities which will involve a plethora of variations especially when it involves elementary movements performed with the arm. Therefore, with this *training* and *testing* set there is a need to explore a different algorithmic technique which can successfully recognize the movements performed in an out-of-laboratory condition.

The employed classification algorithms in this chapter namely LDA, QDA and SVM primarily work based on a decision boundary based system. Data points lying on either side of the decision boundary are classified accordingly to the competing classes (cf. Figure 5.2). LDA and QDA are also affected by outlier data points which might lead to a complicated decision boundary which caters well for the variations of the *training* set but fails to generalize for the data points not used for the modelling (*testing* set). Although SVM caters to outliers by concentrating only on the support vectors that lie proximal to the decision boundary rather than all the data points, none of these methods effectively model the compactness of the data points in the respective feature space. Hence, in the following chapter a clustering based methodology is explored to achieve this. The primary aim is to cater to the underlying data distribution and search for a unique feature space where the data can be represented in compact clusters having a minimal within-class variance.

There are some further important conclusions from this exploration which can be

considered in the forthcoming analysis. From the corresponding results of the prospective evaluation presented in Table 5.10 and Table 5.13 for the healthy subjects and stroke survivors respectively, it is evident that both the accelerometer and the gyroscope sensors placed on the wrist produced better results as compared to the elbow, in terms of individual movement sensitivities and also in terms of the lesser number of features required. A summary of the overall accuracy as a result of the prospective evaluation to detect the movements performed while ‘*making-a-cup-of-tea*’ are presented in Table 5.15.

		Overall accuracy (%)			
Attribute	Subject	Acc_Wrist	Acc_Elbow	Gyr_Wrist	Gyr_Elbow
<b>Healthy subjects</b>	Subject1	71.25	25	88.75	25
	Subject2	50	45	26.25	11.25
	Subject3	27.5	45	78.75	52.5
	Subject4	72.5	40	66.25	78.75
<b>Stroke survivors</b>	Subject1	12.5	17.5	16.25	11.25
	Subject2	12.5	16.25	21.25	15
	Subject3	31.25	22.5	16.25	18.75
	Subject4	20	15	18.75	15

Table 5.15: Overall accuracy for the prospective study for healthy subjects and stroke survivors for each sensor-position.

The results clearly reflect the effectiveness of the wrist over the elbow, which is because it is expected to produce the largest sensor response to the arm movements being investigated. The wrist is the most responsive position producing significant discriminatory sensor responses to the arm movements being investigated. Therefore, in the forthcoming analysis only the individual sensor signals collected from the accelerometer and the gyroscope placed on the wrist are considered, thereby also helping to reduce the amount of data processing involved.

On close examination (cf. Table 5.14) and in further consultation with the respective clinicians, it is observed that there are fundamental similarities in *Action A* and *Action C*, both of which are frequently used interchangeably in daily life for reaching and retrieving objects kept in the front or sideways and involve an extension/flexion of the forearm. Therefore, in the forthcoming analysis these two arm movements are treated to be of similar nature, thereby having to recognize three movements in total (as considered for the exploration in Chapter 4). Henceforth, the arm movements to be recognized in succeeding chapters are: *Action A* (reach and retrieve object), *Action B* (lift cup to mouth) and *Action C* (perform pouring action).



## 6. Chapter 6

# Recognition of Arm Movements using *k*-means Clustering Classification

### 6.1 Introduction

In this chapter, a systematic exploration is performed to recognise the arm movements performed in an out-of-laboratory condition using a clustering and a minimum distance classifier based methodology. From the explorations in Chapter 5 with three supervised learning algorithms, although a high sensitivity across all arm movements could not be achieved, but there were some important conclusions in terms of the sensor positioning and the signal characteristics. With respect to the observations mentioned in section 5.6, here the aim is to recognise only three arm movements performed by each subject, during the archetypal activity of ‘*making-a-cup-of-tea*’, using data collected from only a wrist-worn, wireless tri-axial accelerometer and tri-axial rate gyroscope. This approach is in view of - the suitability of the *personalized* approach for arm movement detection as a rehabilitation indicator, similarities in *Action A* and *Action C*, the effectiveness of the wrist over the elbow as the sensing position and the efficiency of the individual sensor signals in producing discriminative patterns.

The fundamental concept of the exploration presented in this chapter is to first form a set of 3 clusters in multi-dimensional feature space (selected from a ranked set of features), with each cluster representing a particular type of movement, using sets of features generated from person-centric data collected in a constrained *training* phase (e.g. in the laboratory). Subsequent data collected during an unconstrained *testing* phase (e.g. out-of-laboratory) is tested for its proximity to each of the clusters by using a minimum distance classifier. The basic philosophy and the approach adopted are further discussed in detail in the following section.

The highlights of this exploration can be summarised as: (1) demonstrating a completely personalized approach of detecting elementary arm movements accommodating a large

degree of inter-person variability due to different levels of impairment and/or rehabilitation status; (2) analysing kinematic data of healthy subjects and stroke survivors to test the robustness of the methodology; and (3) demonstrating system that does not require periodic training and can therefore be realistically implemented for real-time detection of arm movements in a resource constrained environment. An overview of the application framework for patient monitoring in real-life has also been presented.

## 6.2 Approach

Clustering techniques have been successfully used in diverse fields such as medicine (EEG, Functional MRI), geography or marketing and can be conveniently deployed with limited resources (memory and CPU) [203] [204]. Clustering techniques have also been successfully used in the field of activity recognition for selecting the optimum number of features [205] and for recognising activities as walking, lying, sitting etc from inertial sensor and video data [206][207]. In this work, the application area is further extended to recognise upper limb activities performed by stroke survivors, using the widely popular  $k$ -means clustering algorithm. A major advantage of the  $k$ -means algorithm is its computational simplicity making it an attractive choice for a wide variety of applications [145]. It is a well-perceived fact in the research community that cluster analysis is primarily used for unsupervised learning where the class labels for the training data are not available. However, the  $k$ -means algorithm can also be used for supervised learning where the class labels of the training data are known a priori [198][208]. In this proposed methodology, the class labels for the training data pertaining to the three movements performed in a constrained *training* phase are known. This helps to have a definitive estimate of the underlying cluster structure to be formed on the data (three clusters), thereby facilitating a faster convergence during cluster formation for reduced time complexity [145].

The basic philosophy of the methodology has been illustrated in Figure 6.1, where three clusters A, B and C are formed on the *training* dataset corresponding to the three movements (*Actions A, B and C*) respectively, in a 2-dimensional feature space (Feature 1 ( $f_1$ ) and Feature 2 ( $f_2$ )). The distance of the *test* vector  $T$  from each of the three cluster centroids are represented by the distances  $d_A$ ,  $d_B$  and  $d_C$ . These three distance measures are compared to estimate the proximity of the *test* dataset  $T$  to each cluster and assigned to the nearest one. This methodology can be further scaled up by forming more clusters

corresponding to new categories of movements and associating a new dataset (corresponding to the movement to be detected) to the proximal cluster. The formation of unique clusters corresponding to each performed movement can be achieved by selecting the optimum number of features which help to discriminate movement patterns in the respective feature space.

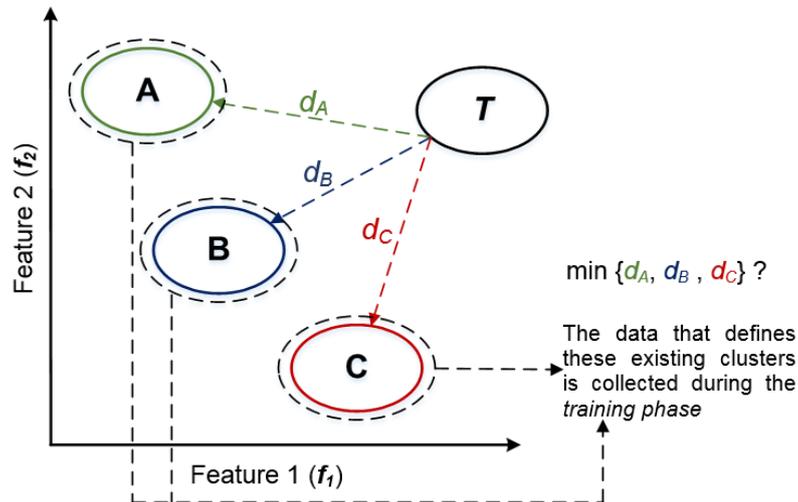


Figure 6.1: Illustration of the clustering and minimum distance classifier based methodology.

The regularized Mahalanobis distance based  $k$ -means clustering technique is used to form the clusters on the person-centric *training* data (collected in the *laboratory* setup) and use 10 runs of a 10-fold cross validation technique to determine the best combination of cluster forming features. A minimum distance classifier based on Euclidean and Mahalanobis distance was used for associating the *test* data (collected during ‘*making-a-cup-of-tea*’) to the formed clusters in the same feature space [209].

To verify the robustness of the proposed methodology two conventional supervised learning algorithms (explored in Chapter 5) are used and the achieved results are compared. The LDA classifier and the SVM classifier with a RBF kernel are used, which are trained on the data collected in the *training* phase and evaluated to predict the movements performed during the *testing* phase using the same best features used by the minimum distance classifier in the proposed methodology.

The primary steps involved in the overall approach are illustrated in Figure 6.2 and described in detail in the following sections.

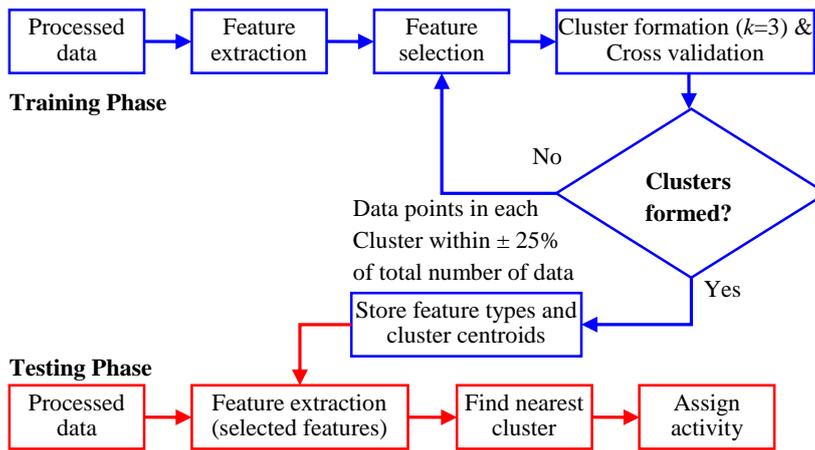


Figure 6.2: Basic stages of the data processing [146].

### 6.2.1 Data acquisition and pre-processing

For this investigation, the experimental data collected from four healthy subjects at UoS and from four stroke survivors at BBK as mentioned in Chapters 4 and 5, was considered. In view of the modifications to the nomenclature of the activities being monitored (cf. section 5.6), the data-structure for the *training* and *testing* phase needs a re-look. The pre-processed sensor data collected from the tri-axial accelerometer and tri-axial rate gyroscope placed on the wrist on the dominant arm for healthy subjects or impaired arm for stroke patients is considered. In view of the similarity between *Action A* and *C*, the three arm movements (*actions*) now considered along with the new nomenclatures are:

- *Action A* – Reach and retrieve an object (extension and flexion of the forearm).
- *Action B* – Lift cup to mouth (rotation of the forearm about the elbow).
- *Action C* – Perform pouring or (un)locking action (rotation of the wrist about long axis of forearm).

For the *training* phase aimed at the target cluster formation, for each of the four healthy participants, 240 trials of *Action A*, 120 trials of *Action B* and 120 trials of *Action C* are considered. For the stroke patients – 80 trials of *A* and 40 trials each of *B* and *C* (patients 1 and 4) and 40 trials of *A* and 20 trials each of *B* and *C* (patients 2 and 3) are considered. This large *training* set offers a greater potential for accurate recognition [39] since the cluster formulation on the *training* data inherently captures the person-centric nature of movement patterns. It is interesting to note that now there are more number of trials for *Action A* in comparison to the other two movements (*Action B* and *C*). However, this is in

line with the fact that *Action A* represents a very generic movement performed more frequently in daily lives.

For the *testing* phase, the archetypal activity-list (cf. Table 5.1) which emulated the process of ‘*making-a-cup-of-tea*’ has been modified accordingly to have repeated occurrences of the three types of arm movement (*actions*). Therefore, the activity-list, shown in Table 6.1, comprising of 20 individual activities now includes 10 occurrences of *Action A*, and 5 each of *Action B* and *Action C*. Sensor data corresponding to four trials of the activity-list by each of the healthy subjects and two trials of the same by each of the stroke survivors are used for this exploration.

Activity	Action
1. Fetch cup from desk	A
2. Place cup on kitchen surface	A
3. Fetch kettle	A
4. Pour out extra water from kettle	C
5. Put kettle onto charging point	A
6. Reach out for the power switch on the wall	A
7. Drink a glass of water while waiting for kettle to boil	B
8. Reach out to switch off the kettle	A
9. Pour hot water from the kettle in to cup	C
10. Fetch milk from the shelf	A
11. Pour milk into cup	C
12. Put the bottle of milk back on shelf	A
13. Fetch cup from kitchen surface	A
14. Have a sip and taste the drink	B
15. Have another sip while walking back to desk	B
16. Unlock drawer	C
17. Retrieve biscuits from drawer	A
18. Eat a biscuit	B
19. Lock drawer	C
20. Have a drink	B

Table 6.1: Use case activity list - ‘Making-a-cup-of-tea’

The data structure for the *training* and *testing* phase used in this analysis is summarised in Table 6.2.

Attributes	Healthy person	Stroke survivors
Number of subjects	4	4
Training dataset	[Action A – 240; Action B – 120; Action C – 120] = 480	[Action A – 80; Action B – 40; Action C – 40] = 160 (patient 1 and 4) [Action A – 40; Action B – 20; Action C – 20] = 80 (patient 2 and 3)
Testing dataset	[Action A – 10; Action B – 5; Action C – 5] x 4 trials = 80	[Action A – 10; Action B – 5; Action C – 5] x 2 trials = 40

Table 6.2: Data structure for each of the two sensor-position combinations using individual sensor signals.

## 6.2.2 Feature extraction

Each accelerometer and gyroscope data stream (X, Y and Z) exhibit signal patterns that are distinctive for each of the arm movements, which is characterized by a set of features extracted from the signals which has already been introduced in section 5.2.3. Ten one-dimensional features are computed on each individual accelerometer ( $AccX$ ,  $AccY$ ,  $AccZ$ ) and gyroscope ( $GyroX$ ,  $GyroY$ ,  $GyroZ$ ) data segment for each movement trial of each subject. The features are: 1) *standard deviation*, 2) *root mean square*, 3) *information entropy*, 4) *jerk metric*, 5) *peak number*, 6) *maximum peak amplitude*, 7) *absolute difference*, 8) *index of dispersion*, 9) *kurtosis*, 10) *skewness*.

The subsequent process of feature selection and cluster formation is performed on the sensor specific feature space (comprising of 30 features), represented as:

$$FS_{individual} = [f_{1-X} \cdots f_{10-X}, f_{1-Y} \cdots f_{10-Y}, f_{1-Z} \cdots f_{10-Z}] \quad (6.1)$$

where,  $FS_{individual}$ , represents the respective feature space for each sensor type (accelerometer or gyroscope). The suffix (X, Y or Z) represents the sensor axis on which the respective feature was computed.

## 6.2.3 Feature selection

The extracted features are linearly normalized and the best features for each subject are selected by using the low-complexity class-separability measure based on scatter matrices which ranks the 30 features for each sensor-movement combination. The scatter matrices quantify the scatter of feature vectors in the feature space. The rank of each individual feature for a multiple-class scenario is determined by the  $R$  value calculated as [145]:

$$R = \frac{S_m}{S_b} \quad (6.2)$$

$$S_m = S_w + S_b \quad (6.3)$$

where  $S_w$  and  $S_b$  are the within-class and between-class scatter matrices respectively and  $S_m$  is the mixture scatter matrix.

$$S_w = \sum_{i=1}^c P_i S_i \quad (6.4)$$

where  $P_i$  denotes the priori probability of a given class  $i = 1, 2, \dots, c$  and  $S_i$  is the respective covariance matrix of class  $i$ .

$$S_b = \sum_{i=1}^c P_i (m_i - m_0)(m_i - m_0)^T \quad (6.5)$$

$$m_0 = \sum_{i=1}^c P_i m_i \quad (6.6)$$

where  $m_0$  is the global mean vector. A high value of  $R$  represents a small within-class variance and a large between-class distance among the data points in the respective feature space [145]. The ranked features are sorted in descending order with respect to their  $R$  values. A sequential forward selection (*sfs*) technique is employed, selecting the first  $i$  features of the ranked feature set in each iteration ( $i = 2, \dots, 30$ ) and it is checked if the data from the *training* phase can be correctly clustered in a multi-dimensional feature space. This has been described in detail in the following section [146].

#### 6.2.4 Cluster formation on the training dataset

The fundamental concept of cluster analysis is to form groups of similar objects as a means of distinguishing them from each other and can be applied in any discipline involving multivariate data [203]. With a given dataset  $X = \{x_i\}$ ,  $i = 1, \dots, n$  to be clustered into a set of  $k$  clusters, the  $k$ -means algorithm iterates to minimize the squared error between the empirical mean of a cluster and the individual data points, defined as the cost function,  $J$ :

$$J(\theta, u) = \sum_{i=1}^n \sum_{j=1}^k u_{ij} \|x_i - \theta_j\|^2 \quad (6.7)$$

where  $\theta_j$  is the cluster center and  $u_{ij} = 1$  if  $x_i$  lies close to  $\theta_j$ , or 0 if otherwise [210]. Initially  $k$  centroids are defined and the data vectors are assigned to a cluster label depending on how close they are to each centroid. The  $k$  centroids are recalculated from the newly defined clusters and the process of reassignment of each data vector to each new centroid is repeated. The algorithm iterates over this loop until the data vectors from the dataset  $X$  form clusters and the cost function  $J$  is minimized [145].

Regularized Mahalanobis distance for cluster formation

The Euclidean distance used to compute the squared distance between the vectors  $x_i$  and the mean of each cluster  $\theta_j$  has an undesirable effect of splitting large and elongated clusters, since most real datasets do not have a well-defined, isolated and spherical underlying cluster structure. By comparison, the use of the Mahalanobis distance which involves computing the covariance matrix of the data vector causes a large cluster to absorb nearby smaller clusters, leading to the creation of unusually large or small clusters. Hence the regularized Mahalanobis distance as mentioned in [210] is used which prevents the clustering algorithm from producing unusually large or small clusters. The distance measure  $J$  is given by:

$$J(x_i, \theta_j) = (x_i - \theta_j)^T [(1 - \lambda)(\Sigma_j + \epsilon I)^{-1} + \lambda I](x_i - \theta_j) \quad (6.8)$$

where  $\Sigma_j$  is the covariance matrix of the  $k$ -th cluster and  $I$  is the  $d \times d$  identity matrix,  $d$  is the input dimensionality (no. of feature vectors representing the data vector) and  $\epsilon$  ( $10^{-6}$ ) is the regularization parameter. The value of  $\lambda$  can be used as a parameter to control the choice of distance measure to be used, with  $\lambda=0$ ,  $J$  is the squared Mahalanobis distance and when  $\lambda=1$ ,  $J$  is the squared Euclidean distance [210]. In this exploration, an initial value of  $\lambda=1$  (Euclidean) is used and after 3 iterations it is changed to  $\lambda=0$  (Mahalanobis). The cluster formation on the *training* dataset is associated with a cross-validation step to determine the best combination of the features, discussed in detail in the following section.

**6.2.5 Cross-validation of the training dataset**

Ten runs of ten-fold cross-validation on the feature vectors are computed from the accelerometer and gyroscope data which characterizes the movement trials of the *training* phase (480 trials for each healthy subject, 160 trials for patients 1 and 4 and 80 trials for patients 2 and 3) to form three clusters representing the three arm movements. The cluster formation and cross-validation steps are illustrated in Figure 6.3. The key steps involved in the cross-validation process have also been highlighted.

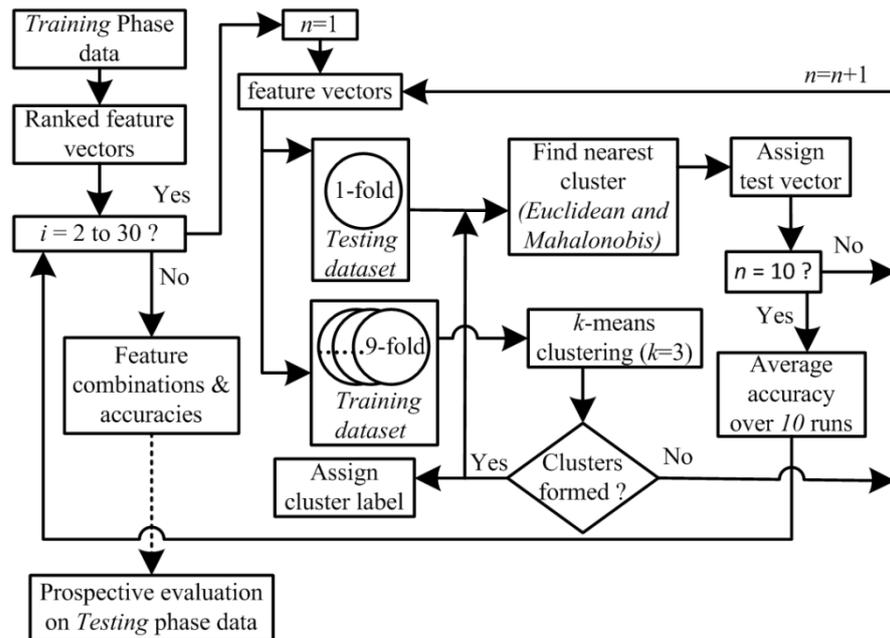


Figure 6.3: Overview of cluster formation and cross validation on the *Training* phase data collected from each sensor and each subject [209].

- The cluster formation using the regularized Mahalanobis distance (as discussed in section 6.2.4) runs on the *training* dataset for each subject comprising of feature vectors (30 features) extracted from each sensor data segment.
- The algorithm runs in conjunction with the *sfs* algorithm sequentially selecting a combination of 2 to 30 ranked features in each step ( $i$ ).
- For a particular set of feature vectors selected ( $i$ ), 10 runs ( $n$ ) of 10-fold cross validation are carried out whereby 10 segments of the *training* data are created.
- In each run ( $n$ ) of the stipulated 10 runs, one segment is used as the *test* dataset while the rest of the 9 segments are used as the *training* dataset.
- A threshold of 25% of the expected number of data points is set for each of the three clusters formed (i.e. for healthy subjects:  $240 \pm 60$  for *Action A* and  $120 \pm 30$  for *Action B* and *Action C*, for patients 1 and 4:  $80 \pm 20$  for *Action A*,  $40 \pm 10$  for *Action B* and *Action C*, for patients 2 and 3:  $40 \pm 10$  for *Action A*,  $20 \pm 5$  for *Action B* and *Action C*). This threshold value was experimentally selected since it produced the best results [146]. If the number of data points in each cluster is within the threshold, it is considered as correctly clustered for that particular combination ( $i$ ) of features selected (where  $i = 2, \dots, 30$ ).
- The distance of the mean of the *training* dataset for each class label from the cluster centroids is computed and thereby each cluster is assigned with the class label that has its closest proximity to that particular class of the *training* dataset.

- Correspondingly a minimum distance classifier is used to compute the distance of the *test* dataset (one segment of the stipulated 10-folds) from the centroid of each cluster in a multi-dimensional feature space (considering the feature combination of the current step,  $i$ ) based upon: a). Euclidean distance and b). Mahalanobis distance. The Mahalanobis distance is used to measure the distance of a point from a data distribution. The data distribution is characterized by the mean and the covariance matrix which defines the shape of how the data is distributed in the feature space and is generally hypothesized as a multivariate Gaussian distribution. Here, the Mahalanobis distance takes into consideration the covariance of the clusters along with their mean for the maximum likelihood estimation of the covariance matrix and hence is effective for clusters with larger variance along one or many directions and in general having an ellipsoidal shape [145].
- The *test* dataset is assigned to a particular cluster depending on the minimum distance computed for each of the two measures.
- The predicted label is verified with respect to the known annotations thereby ascertaining the accuracy of the prediction for a single run ( $n$ ).
- The accuracy of prediction for a particular feature combination is determined by averaging the results produced over the runs ( $n \leq 10$ ) forming successful clusters. This process is repeated for each of the sequentially selected best ranked feature combinations ( $i = 2, \dots, 30$ ).

Therefore at the end of all iterations (i.e.  $i = 30$ ), a detailed list of the feature combinations that resulted in a successful cluster formation and the corresponding accuracies achieved both with Euclidean and Mahalanobis distance measures, for each subject and each sensor type is obtained. This information (minimal number of feature combination, resulting in the best accuracy) is used for the target classification of the *testing phase* data collected in the *semi-naturalistic setup* by associating them to the formed clusters in the chosen feature space. A brief summary of the cross-validation results have been presented here.

For the healthy subjects, the number of features chosen was in the range of 2 to 27 (selected out of total of 30) and the individual sensitivities were in the range of 90% to 100% across both the sensor types. For the stroke survivors, the number of features was in the range of 8 to 30 and the sensitivities were in the range of 80% to 100% using data from the accelerometer or the gyroscope.

### 6.2.6 Prospective evaluation with the testing dataset

The data from the *testing* phase (cf. Table 6.1), collected in the *semi-naturalistic setup* during the ‘*making-a-cup-of-tea*’ session (80 test vectors for each healthy subject [(10A + 5B + 5C) × 4 trials] and 40 test vectors for each stroke patient [(10A + 5B + 5C) × 2 trials]) is pre-processed for each type of sensor, and only those features are extracted from each test vector which resulted in the best accuracy in the cross validation of the *training* data. A Euclidean and Mahalanobis distance based classifier is used to compute the distance of each test vector (represented by the extracted features) from the centroid of each cluster in a multi-dimensional feature space. The test vector is assigned to a particular cluster depending on the minimum distance computed for each of the two measures. The predicted label is verified with respect to the annotations in the activity-list of Table 6.1. The clustering and the minimum distance classification were implemented in Matlab.

## 6.3 Results and analysis – prospective evaluation

The typical variations in accelerometer and gyroscope data recorded during a single example of each action for a healthy and a stroke survivor is illustrated in Figure 6.4 and Figure 6.5 respectively. The difference in movement profiles among the two groups and the longer time taken by the stroke survivors to complete the actions with less smoothness of movement is clearly visible.

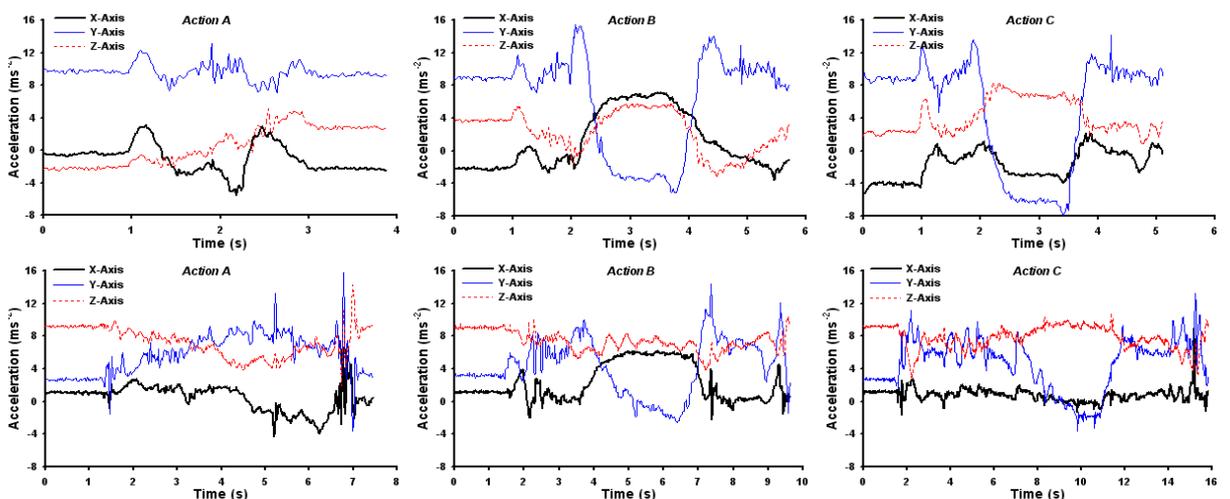


Figure 6.4: Data from a tri-axial accelerometer located on the wrist collected while performing arm actions A, B and C from a healthy subject (upper) and a stroke survivor (lower) [183].

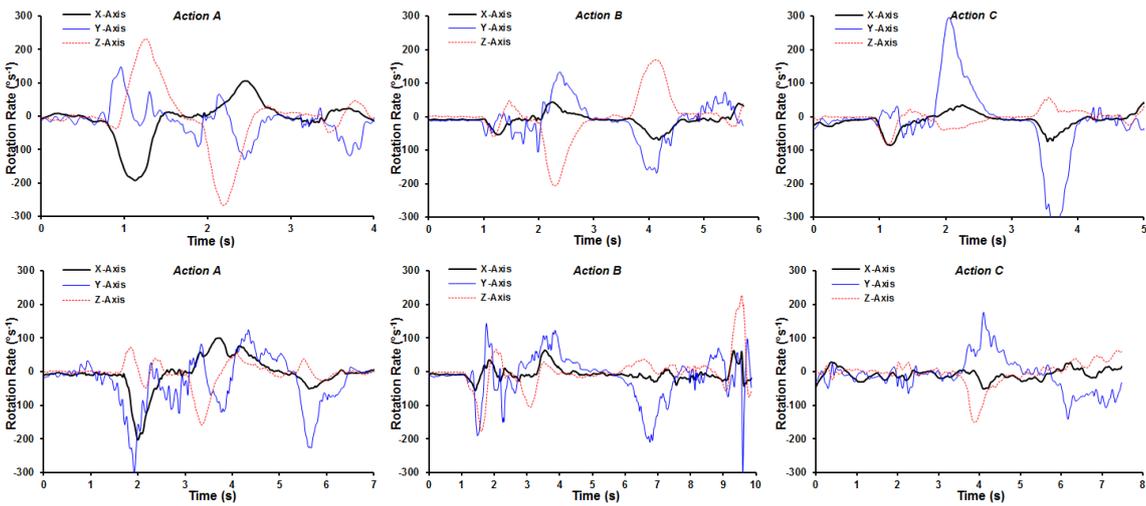


Figure 6.5: Data from a tri-axial rate gyroscope located on the wrist collected while performing arm actions A, B and C from a healthy subject (upper) and a stroke survivor (lower) [209].

There were a total of 80 movement trials (*actions*) to be recognized (40 of A, 20 of B, 20 of C) for each healthy subject and 40 movement trials to be recognized (20 of A, 10 of B, 10 of C) for each stroke patient. The results of the prospective evaluation in terms of the sensitivity of recognizing the movements performed in the *testing* phase for the healthy subjects using accelerometer and gyroscope data are presented in Table 6.3 and Table 6.4 respectively and for the stroke patients are shown in Table 6.6 and Table 6.7 respectively. The tables also show the minimum number of features that were required to successfully form the three clusters for each subject. The number of features has been determined by 10 runs of 10-fold cross validation on the *training* phase data as discussed in section 6.2.5. The results in general show that each subject required a different minimum number of features to successfully form 3 separate clusters from the *training* data, reflecting the variability in arm movement patterns between individuals. The minimum distance classifier used for recognizing the arm movements has also been shown in each table. The right hand column in Table 6.3, Table 6.4, Table 6.6 and Table 6.7 show the overall detection accuracy for each subject (total number of recognized actions expressed as a percentage of the total number of actions performed).

### 6.3.1 Healthy subjects

The overall accuracy covers the range 61% to 100% (average of 88%) using accelerometer data and 60% to 94% using gyroscope data (average of 83%) for all healthy subjects. In general, these recognition accuracies are quite favourable considering the elementary nature of the arm movements being detected and using only the data collected in the

laboratory for *training*. The obvious exceptions are the detection of *Action B* for Subject 2 (5%) using accelerometer and *Action A* for Subject 4 (30%) using gyroscope data respectively.

Subject	Features	Minimum Distance Classifier	Sensitivities (%)			Overall accuracy (%)
			A	B	C	
Subject1	11	Euclidean	100	100	100	100
Subject2	2	Euclidean	80	5	80	61
Subject3	7	Euclidean	95	100	90	95
Subject4	23	Euclidean	95	100	85	94

Table 6.3: Recognition sensitivities for each arm movement with accelerometer data for healthy subjects.

Subject	Features	Minimum Distance Classifier	Sensitivities (%)			Overall accuracy (%)
			A	B	C	
Subject1	10	Euclidean	93	90	100	94
Subject2	27	Euclidean	100	80	60	85
Subject3	18	Mahalanobis	90	90	100	93
Subject4	20	Euclidean	30	95	85	60

Table 6.4: Recognition sensitivities for each arm movement with gyroscope data for healthy subjects.

It is worth mentioning that Subject 2 required the smallest number of features (2) to form clusters from the *training* data. This is somewhat counter-intuitive – fewer features imply sufficient differences in arm movement patterns to make unique cluster formation easier. Whilst this may be the case, however, the low detection accuracy for *Action B* could be accounted for by poor repeatability by the subject in this particular arm movement. Similarly, for Subject 4, using 20 features from gyroscope data and a Euclidean distance classifier the low accuracy for *Action A* may be attributed to poor repeatability.

However, for these specific cases, for Subject 2, the sensitivity of *Action B* with gyroscope data is 80% (27 features) and for Subject 4, the sensitivity of *Action A* with accelerometer data is 95% (with 23 features). Therefore, although the overall recognition accuracies using both accelerometer and gyroscope data are nearly similar, for specific cases considering more than one sensor type can improve the overall accuracy of detection. The detection accuracies for Subject 2 with accelerometer data using additional features are illustrated in Figure 6.6, which reveals that increasing the number of features beyond 2 does not yield successful cluster formations (blank spaces) or improved accuracy.

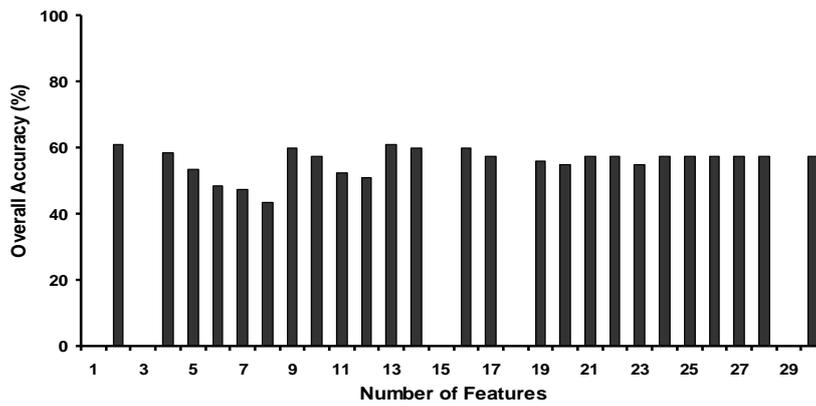


Figure 6.6: Change in accuracy with number of features for healthy Subject 2 with accelerometer data using Euclidean distance.

A list of features selected from a ranked list (sorted in descending order) is presented in Table 6.5, specific to each subject and each sensor, illustrating the difference in the number of features required to form the clusters. The suffix (*x*, *y* or *z*) represents the sensor axis on which the respective feature was extracted. Although for each subject the ranked order of features is different (reflecting the different ways in which they perform a movement) there is a strong commonality in the top ranked features across all subjects for both the sensors. Specifically, the features *stddev\_y* and *rms\_y* are the top two features extracted from the accelerometer data across all subjects except for Subject 3, where it lies within the best five features used. By comparison, the features *rms\_x*, *stddev\_x* and *diff\_x* are routinely amongst the best five features extracted from the gyroscope data.

Accelerometer	
Subject	Ranked Features
Subject1	<i>stddev_y</i> , <i>rms_y</i> , <i>rms_z</i> , <i>stddev_z</i> , <i>rms_x</i> , <i>diff_y</i> , <i>stddev_x</i> , <i>diff_z</i> , <i>max_mag_y</i> , <i>diff_x</i> , <i>max_mag_z</i>
Subject2	<i>stddev_y</i> , <i>rms_y</i>
Subject3	<i>rms_z</i> , <i>rms_x</i> , <i>stddev_y</i> , <i>stddev_x</i> , <i>rms_y</i> , <i>entropy_z</i> , <i>stddev_z</i>
Subject4	<i>stddev_y</i> , <i>rms_y</i> , <i>stddev_x</i> , <i>rms_x</i> , <i>diff_y</i> , <i>max_mag_y</i> , <i>diff_x</i> , <i>max_mag_x</i> , <i>kurtosis_x</i> , <i>kurtosis_z</i> , <i>skewness_z</i> , <i>entropy_y</i> , <i>diff_z</i> , <i>max_mag_z</i> , <i>kurtosis_y</i> , <i>stddev_z</i> , <i>entropy_x</i> , <i>skewness_x</i> , <i>peaks_y</i> , <i>skewness_y</i> , <i>entropy_z</i> , <i>rms_z</i> , <i>peaks_x</i>
Gyroscope	
Subject	Ranked Features
Subject1	<i>rms_x</i> , <i>stddev_x</i> , <i>rms_z</i> , <i>diff_x</i> , <i>diff_z</i> , <i>rms_y</i> , <i>stddev_z</i> , <i>max_mag_x</i> , <i>max_mag_z</i> , <i>stddev_y</i>
Subject2	<i>rms_x</i> , <i>stddev_x</i> , <i>diff_x</i> , <i>max_mag_x</i> , <i>stddev_y</i> , <i>rms_y</i> , <i>diff_y</i> , <i>max_mag_y</i> , <i>rms_z</i> , <i>stddev_z</i> , <i>entropy_y</i> , <i>skewness_z</i> , <i>diff_z</i> , <i>entropy_z</i> , <i>entropy_x</i> , <i>skewness_x</i> , <i>skewness_y</i> , <i>kurtosis_y</i> , <i>kurtosis_x</i> , <i>kurtosis_z</i> , <i>max_mag_z</i> , <i>jerk_y</i> , <i>peaks_x</i> , <i>jerk_z</i> , <i>peaks_z</i> , <i>jerk_x</i> , <i>peaks_y</i>
Subject3	<i>rms_z</i> , <i>rms_x</i> , <i>stddev_z</i> , <i>stddev_x</i> , <i>diff_x</i> , <i>diff_z</i> , <i>max_mag_x</i> , <i>max_mag_z</i> , <i>max_mag_y</i> , <i>entropy_y</i> , <i>diff_y</i> , <i>entropy_z</i> , <i>rms_y</i> , <i>entropy_x</i> , <i>stddev_y</i> , <i>skewness_y</i> , <i>skewness_z</i> , <i>peaks_x</i>
Subject4	<i>rms_x</i> , <i>stddev_x</i> , <i>diff_x</i> , <i>max_mag_x</i> , <i>rms_z</i> , <i>max_mag_y</i> , <i>stddev_z</i> , <i>stddev_y</i> , <i>rms_y</i> , <i>diff_y</i> , <i>skewness_y</i> , <i>skewness_z</i> , <i>diff_z</i> , <i>entropy_x</i> , <i>kurtosis_y</i> , <i>entropy_z</i> , <i>jerk_y</i> , <i>max_mag_z</i> , <i>entropy_y</i> , <i>kurtosis_z</i>

Table 6.5: Selected features in ranked order for healthy subjects

### 6.3.2 Stroke survivors

For the stroke survivors, the overall accuracy is in the range of 40% to 88% (average of 70%) using accelerometer data and 40% to 83% (average of 66%) using gyroscope data as shown in Table 6.6 and Table 6.7 respectively. Under closer examination, it can be observed that for Subject 1 the sensitivities with accelerometer data for individual actions are above 80%, but for Subjects 2 and 3, although the overall accuracy is above 70%, the sensitivity for *Action B* and *Action C* are quite low (20%). When the gyroscope data is taken into consideration, these particular action/subject combinations are improved considerably: 100% detection for *Action B* with Subject 2 and 80% detection for *Action C* with Subject 3. For Subject 4, the overall results with both sensors are not high, although it can be seen that *Action A* can be recognized by 60% (gyroscope), *Action B* by 80% and *Action C* by 60% (accelerometer).

The low overall accuracy can be attributed to the fact that Subject 4 was at an early stage of rehabilitation and the impaired arm being tested was not the naturally dominant arm thereby resulting in poor repeatability. This further emphasizes how detection accuracies may be improved by considering more than one sensor type for specific cases. Moreover, it is worth noting that for stroke patients the Mahalanobis distance based classifier is more effective than the Euclidean distance. This is further reflective of the fact that there is a high degree of variability in their movement profile resulting in clusters having a larger variance along one or many directions, and in such conditions the Mahalanobis distance acts as a more effective distance measure.

Subject	Features	Minimum Distance Classifier	Sensitivities (%)			Overall accuracy (%)
			<i>A</i>	<i>B</i>	<i>C</i>	
Subject1	19	Mahalanobis	80	90	100	88
Subject2	19	Mahalanobis	90	20	100	75
Subject3	21	Mahalanobis	95	100	20	78
Subject4	8	Euclidean	10	80	60	40

Table 6.6: Recognition sensitivities for each arm movement with accelerometer data for stroke survivors.

Subject	Features	Minimum Distance Classifier	Sensitivities (%)			Overall accuracy (%)
			<i>A</i>	<i>B</i>	<i>C</i>	
Subject1	8	Euclidean	90	50	100	83
Subject2	10	Euclidean	60	100	60	70
Subject3	24	Mahalanobis	85	30	80	70
Subject4	30	Mahalanobis	60	40	0	40

Table 6.7: Recognition sensitivities for each arm movement with gyroscope data for stroke survivors.

The ranked list of features selected for each stroke patient using each sensor is presented in Table 6.8. In comparison to healthy subjects, the features selected for each stroke patient are quite different, exhibiting a minimal degree of commonality among the top ranked features. Each of the four subjects had differing levels of functional ability (as assessed by the therapists) and hence, there was a great degree of intra-subject and inter-subject variability in the movement patterns.

Accelerometer	
Subject	Ranked Features
Subject1	stddev_y, stddev_x, rms_x, rms_y, entropy_y, diff_x, entropy_z, max_mag_x, stddev_z, rms_z, peaks_z, entropy_x, diff_y, peaks_x, diff_z, kurtosis_x, max_mag_z, peaks_y, kurtosis_y
Subject2	diff_x, stddev_y, skewness_x, entropy_y, rms_y, stddev_x, rms_x, peaks_x, entropy_z, entropy_x, max_mag_x, peaks_z, kurtosis_y, diff_y, max_mag_y, peaks_y, kurtosis_x, skewness_y, max_mag_z
Subject3	entropy_z, entropy_y, entropy_x, stddev_z, rms_z, rms_y, stddev_y, skewness_z, skewness_x, kurtosis_z, diff_z, kurtosis_x, max_mag_z, peaks_z, kurtosis_y, peaks_x, peaks_y, skewness_y, diff_x, rms_x
Subject4	entropy_y, entropy_z, stddev_y, peaks_x, entropy_x, rms_x, stddev_x, rms_y
Gyroscope	
Subject	Ranked Features
Subject1	stddev_x, rms_x, diff_x, max_mag_x, stddev_z, rms_z, entropy_z, entropy_y
Subject2	diff_z, stddev_z, rms_z, max_mag_z, diff_x, stddev_x, rms_x, entropy_y, max_mag_x, stddev_y
Subject3	diff_z, stddev_z, rms_z, max_mag_z, entropy_y, entropy_z, entropy_x, stddev_x, rms_x, skewness_x, diff_x, max_mag_x, jerk_x, peaks_z, peaks_x, peaks_y, skewness_y, stddev_y, kurtosis_z, jerk_y, kurtosis_y, rms_y, kurtosis_x, diff_y
Subject4	skewness_y, stddev_y, entropy_y, entropy_z, stddev_x, rms_x, peaks_z, max_mag_x, diff_x, peaks_x, entropy_x, peaks_y, rms_y, diff_y, stddev_z, rms_z, skewness_x, diff_z, jerk_y, max_mag_y, kurtosis_y, max_mag_z, kurtosis_z, skewness_z, jerk_x, kurtosis_x, jerk_z, disp_y, disp_z, disp_x

Table 6.8: Selected features in ranked order for stroke survivors.

### 6.3.3 Evaluation with LDA and SVM

For comparing the performance of the proposed methodology the supervised learning algorithms LDA and SVM with radial basis function (RBF) kernel are used for classifying the three arm movements performed during the *testing* phase. These algorithms were used in the previous chapter but for classifying four movements. Therefore, this exploration with three movement categories will be a further testimony to their effectiveness. The data collected in the *training* phase is used to train the classifiers and they are evaluated to detect the movements performed in the *testing* phase while ‘*making-a-cup-of-tea*’ [209], following the approach mentioned in section 5.5. In this investigation the same set of features were used as in the proposed clustering and minimum distance classifier based methodology for both the healthy subjects (cf. Table 6.5) and the stroke survivors (cf. Table 6.8). The classification results are presented in Table 6.9 - Table 6.12.

Subject	Accelerometer				Gyroscope			
	Sensitivities (%)			Overall accuracy (%)	Sensitivities (%)			Overall accuracy (%)
	A	B	C		A	B	C	
Subject1	45	80	0	42	60	0	95	54
Subject2	70	0	85	56	30	15	95	42
Subject3	60	95	15	57	100	15	10	56
Subject4	0	100	0	25	100	0	40	60

Table 6.9: Sensitivities for each arm movement and overall accuracies for healthy subjects using LDA.

Subject	Accelerometer				Gyroscope			
	Sensitivities (%)			Overall accuracy (%)	Sensitivities (%)			Overall accuracy (%)
	A	B	C		A	B	C	
Subject1	35	100	40	52	40	90	100	67
Subject2	15	0	100	32	20	10	100	37
Subject3	60	100	20	60	35	0	70	35
Subject4	95	0	20	52	60	0	60	45

Table 6.10: Sensitivities for each arm movement and overall accuracies for stroke survivors using LDA.

Subject	Accelerometer				Gyroscope			
	Sensitivities (%)			Overall accuracy (%)	Sensitivities (%)			Overall accuracy (%)
	A	B	C		A	B	C	
Subject1	50	95	0	48	85	80	20	67
Subject2	80	5	80	61	87	75	25	68
Subject3	45	95	10	49	90	75	20	69
Subject4	65	95	5	57	95	70	20	70

Table 6.11: Sensitivities for each arm movement and overall accuracies for healthy subjects using SVM.

Subject	Accelerometer				Gyroscope			
	Sensitivities (%)			Overall accuracy (%)	Sensitivities (%)			Overall accuracy (%)
	A	B	C		A	B	C	
Subject1	30	70	100	57	95	60	90	85
Subject2	55	100	70	70	100	0	0	50
Subject3	65	20	90	60	30	0	100	40
Subject4	10	50	70	35	5	0	90	25

Table 6.12: Sensitivities for each arm movement and overall accuracies for stroke survivors using SVM.

Using LDA, for the healthy subjects, the overall accuracy is in the range of 25% to 57% (average of 45%) using accelerometer data and 42% to 60% (average of 53%) using gyroscope data as shown in Table 6.9. For the stroke survivors, the overall accuracy was in the range of 32% - 60% (average of 49%) using accelerometer data and range of 35% - 67% (average of 46%) using gyroscope data as shown in Table 6.10. Using SVM, for the healthy subjects, the overall accuracy is in the range of 48% - 61% (average of 54%) using accelerometer data and 67% - 70% (average of 68%) using gyroscope data as shown in Table 6.11. For the stroke survivors, the overall accuracy was in the range of 35% to 70% (average of 55%) using accelerometer data and 25% to 85% (average of 50%) using gyroscope data as shown in Table 6.12. The overall accuracies using LDA and SVM are

comparatively on the lower side when compared to the results achieved using the proposed methodology (cf. Table 6.3, Table 6.4, Table 6.6 and Table 6.7). The sensitivities for individual actions achieved using LDA and SVM are high for some cases, though none of the subjects have all three movements classified with a sensitivity of greater than 60%. This therefore proves the effectiveness of the proposed methodology.

## 6.4 Summary of exploration

A proof-of-concept methodology based on *k*-means clustering and a minimum distance classifier to recognize three fundamental movements of the upper limb has been presented which uses data collected from a wrist-worn, wireless tri-axial accelerometer and a tri-axial gyroscope. From, the results presented in section 6.3, for four healthy subjects and four stroke patients it can be concluded that:

- the proposed method can recognize the three movements with an overall average accuracy of 88% using just accelerometer data and 83% when using only gyroscope data across all healthy subjects and an average accuracy of 70% using accelerometer data and 66% using gyroscope data across all stroke survivors and arm movement types.
- the minimum sensitivity for detecting each individual arm movement was 80% for healthy subjects and 60% for stroke patients if more than one sensor is used. The results particularly those obtained for the stroke survivors reveal that there is a need to consider more than one sensor type while detecting such elementary arm movements.
- the results in general show that each subject required a different minimum number of features to successfully form 3 separate clusters from the *training* data, reflecting the variability in arm movement patterns between individuals. Although there is a strong commonality in the top ranked features selected for both the sensors across the healthy subject population, for the stroke survivors there was a minimal degree of commonality.
- a further comparison of the achieved results against the classification results obtained using supervised learning algorithms - LDA and SVM, clearly reflect the effectiveness of the proposed methodology in detecting the three investigated arm movements.

## 6.5 Application framework

In view of this exploration and the achieved results, the proposed clustering and minimum distance classifier based methodology can be implemented in real life for monitoring arm rehabilitation progress in remote health monitoring applications. This methodology could be used to track the number of times a patient performs specific arm movements with their paretic arm throughout the day. Therefore, the application framework for a patient monitoring system has been highlighted.

It is evident from the results especially that of the stroke survivors with each subject requiring differing number of features for movement classification, that there is a need for a completely personalized approach of detecting elementary arm movements. Various factors like stage of rehabilitation and the affected arm/natural arm, play a significant role in detection since they affect the level of repeatability of individual movements as well as introduce a high degree of temporal variation.

For any patient suffering from arm dexterity, the specific movements (or exercises) that need to be tracked as defined by clinicians need to be performed multiple times, following an exercise regime or a gaming session, in a controlled environment (clinic or home). The sensor data collected during this phase can be analysed through cross-validation to determine the best cluster forming features and obtain the centroids of each cluster corresponding to each movement. This helps to perform a clinical profiling of the individual patient with respect to their movement quality. Movements performed in the uncontrolled nomadic environment (which can involve daily activities) can be associated to the proximal cluster centroid using the minimum distance classifier.

It is also important to mention that this methodology can be adaptable to the changing movement patterns of the patients over time reflective of an improvement in their motor functionality depending on the rehabilitation. The change in movement patterns over a longitudinal scale can be determined by two means: 1) when the recognition rate of the movements become poor over time – since the movements performed in daily life would have differing patterns in the feature space with respect to the pre-computed cluster centroids and 2) clinical intervention – when the clinicians feel that there has been a considerable change in the patient's movements compared to the time of obtaining the clusters from the *training* data. In such circumstances, the patient's *training* data can be

collected periodically and the cluster centroids and the associated features (new selected feature set) can be recomputed to reflect the changing movement patterns. This information (new cluster centroid and feature set) will be subject-specific due to the inter-subject variability in movement profiles (as evident from section 6.3.2), variation in the rehabilitation profile and the associated functional ability of each individual subject. This information can be further used by the minimum distance classifier to recognize movements performed in daily life.

Given the application framework, this methodology can be implemented for online detection of arm movements in a resource constrained environment of body-worn sensor nodes. For the *training* phase, the key steps of cluster formation and feature selection (being relatively time and memory intensive) need only be done in an offline mode when requested by the clinician, depending on the rehabilitation progress of the patient over time. Further, an online detection module can be used to compute only the required features and the distance to the pre-computed cluster centroids in near real-time, thereby providing an energy efficient solution towards operation of wearable sensors for long durations [33].

## 6.6 Discussion

In this chapter, an algorithmic exploration involving *k*-means clustering and minimum distance classifier has been performed, to classify three arm movements performed in an out-of-laboratory condition by four healthy subjects and four stroke survivors. The results have highlighted the difference in the movement profiles inherent between the healthy subjects and stroke survivors. In view of this, the application framework for real-time patient monitoring within home settings has been discussed. An envisaged remote patient monitoring system needs to encompass various other inter-connected sub-systems along with the proposed recognition methodology. This could involve a patient workstation at home or clinic, having motion capture systems (e.g. Kinect), an interactive touch-screen surface and numerous objects for dexterous and functional hand exercises. This workstation could be used for collecting the data as part of the *training* phase while the patients perform the stipulated movements. The data can be communicated by the workstation to the PHR through a secure internet connection. The collected data can be analysed with the computing facilities at the clinic or the patient's home where the key steps of cluster formation and feature selection can be performed. This information (cluster centroids and

selected features) can be uploaded to the on-board memory of a body-worn sensor node. The recognition logic built within the sensor node, employing the minimum distance classifier can be used for online detection of arm movements. This further necessitates the low-power implementation of the minimum distance classifier on hardware for sustained operation of a resource constrained body-worn sensor node.



# 7. Chapter 7

## ASIC Design for Minimum Distance Classifier

### 7.1 Introduction

Body-worn wireless sensor nodes have facilitated long-term continuous monitoring of patients in remote health applications [211]. The fundamental requirement for the sensor nodes in such a system is low-power operation to prolong the battery life of the sensors owing to its resource constrained nature. The major components of the recognition system include computationally intensive steps like feature extraction from the data acquired by the sensors and its corresponding analysis and therefore traditionally, these are carried out off-line on mainframe computational facilities as discussed in section 2.2.5.5.3. However, for a continuous monitoring scenario it is beneficial to perform these two steps within the sensor node itself for compensating the significant energy required at the radio front-end of the sensors for continuous data transmission. [33]. Therefore, it is of paramount importance to develop a low-power strategy for feature extraction and classification in the resource constrained environment of a sensor node.

In this chapter an architectural implementation of the arm recognition methodology mentioned in Chapter 6 has been presented, employing a clustering and minimum distance classifier based approach. The algorithm has been implemented using a hardware description language (Verilog) aimed at developing an ASIC. The developed hardware can be embedded on-board the sensor nodes for achieving a real-time arm movement recognition system for long-term remote monitoring. It is envisaged that a wearable sensor node having a customized ASIC chip can be used for capturing and on-board processing of the raw data as shown in Figure 7.1. The chip which is dedicated to perform the key steps of feature extraction and classification can be embedded on the sensor platform along with other components aimed at pre-processing (noise and artefact removal) and compressing the data. The inferred observations or results along with the raw sensor data can be stored in memory within the sensor module and can be transmitted to the PHR using standard

communication protocol, Bluetooth or Zigbee at pre-decided intervals of time. Data compression can be achieved using the wavelet-based low-complexity methodology proposed in [212], to store the raw sensor data and the classification results.

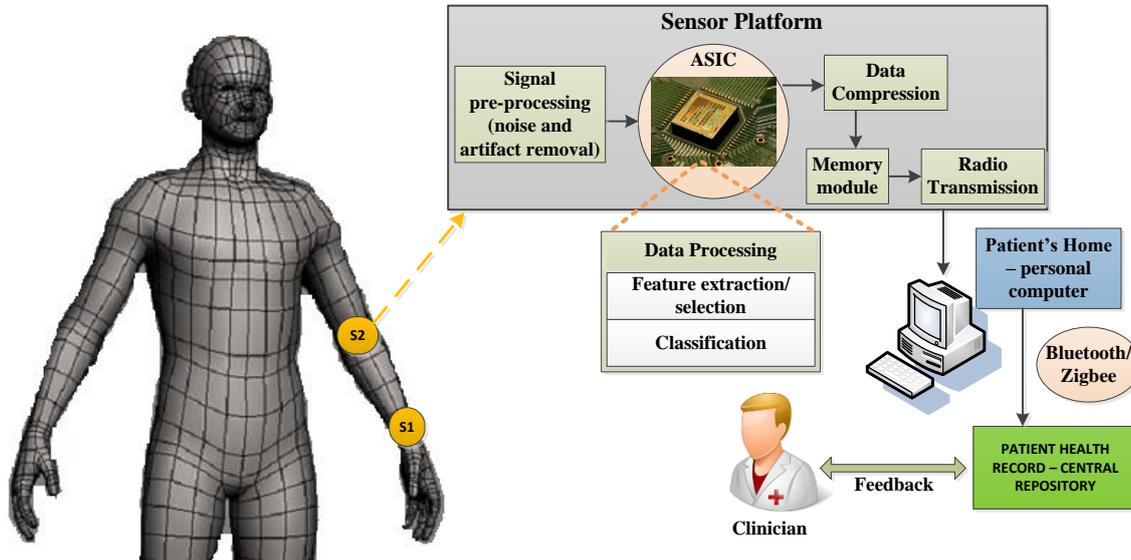


Figure 7.1: Abstract sensor plane overview

## 7.2 Approach

The clustering based approach has been developed in an offline-online resource sharing mechanism, targeted towards an ASIC implementation. The processing overview has been illustrated in Figure 7.2.

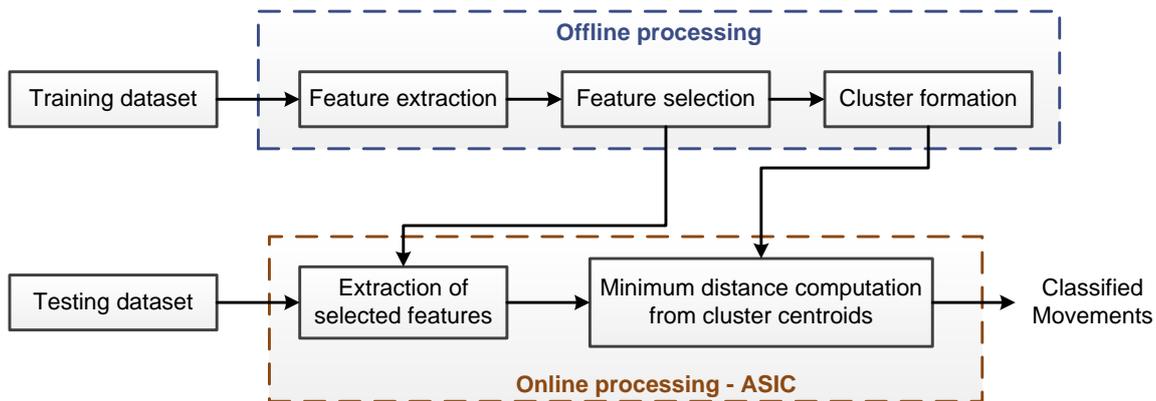


Figure 7.2: Overview of the offline-online processing – the *training* dataset is processed offline and the *testing* dataset is processed online. The computation of the selected features on the *testing* data and computation of the minimum distance from the pre-computed cluster centroids was done in ASIC for real-time detection of arm movements.

The time and memory intensive process of feature computation, selection and cluster

formation, on the *training* data were done in an offline mode (in Matlab). The computation of the selected features on the *testing* data and computation of the minimum distance (Euclidean) from the pre-computed cluster centroids was done in hardware (ASIC) for real-time implementation. This offline-online processing approach satisfies the application requirements of remote monitoring of arm rehabilitation. As highlighted in section 6.5, the collection of the data during the *exercise phase* (*training* data) and the associated processing (feature extraction/selection, clustering) need only be done in an offline mode when requested by the clinician or the recognition rate falls below a threshold over a longitudinal monitoring scale, reflective of the rehabilitation progress of the patient. The online detection module can be used to associate the activities performed in nomadic settings (*testing* data) to the pre-computed cluster centroids in real-time. Therefore the ASIC does not need regular training and can be updated on a periodic basis with the new feature set and cluster centroids (computed in software) which can be used by the hardware to predict arm movements performed in daily life over time. The fabricated ASIC is envisaged to be embedded on a sensor node, illustrated in Figure 7.3, along with a microcontroller and other processing components like A/D converter, memory and de-noising circuit (digital filters), which can be used to recognise arm movements performed in daily life for real-time continuous monitoring.

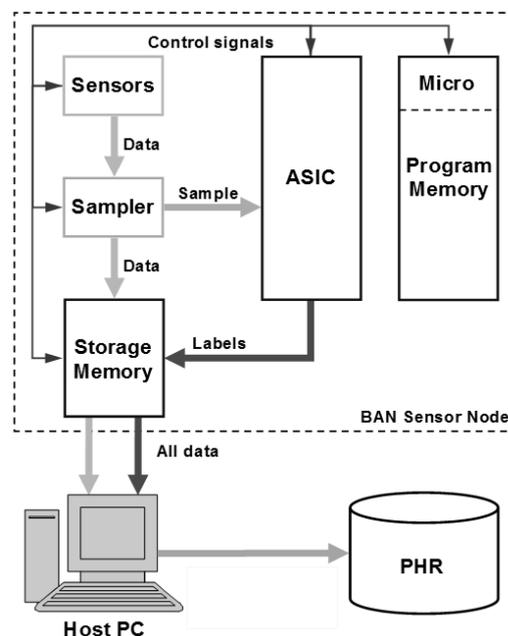


Figure 7.3: Overview of the envisaged sensor node with the ASIC, microcontroller and memory.

The cluster centroids and *feature-code* (selected features) are uploaded to the on-board memory when the sensor module is plugged into the docking station for charging at the end of a monitoring session. The on-board microcontroller is aimed at controlling the address and the data signals to and from the ASIC, placed within the sensor node. The

classification results produced in real-time can also be stored within the memory along with the raw sensor data, which can be uploaded to a local host computer and also to the patient health record system (PHR).

The pre-processing of the raw sensor data for the *training and testing* phases as mentioned in section 5.2.1.2 were not implemented in RTL. This is because a filtering circuit can be added along with the implemented design on the envisaged sensor platform to pre-process the data (cf. Figure 7.1). In this chapter, the focus was particularly on the hardware implementation of the minimum distance classifier. It is also important to note that in this implementation the Euclidean distance has been considered over the Mahalanobis distance for the minimum distance classifier. This is primarily because the Mahalanobis distance involves the calculation of the covariance matrix which is computationally complex. This implementation was targeted as a proof-of-concept vehicle for implementing the minimum distance classifier aimed at movement recognition. In the following sections the design and implementation of a feature extraction engine has been described in detail aimed at computing the time domain features from the *testing phase* dataset and the computation of the minimum distance of the extracted features from the cluster centroids in hardware. The computation of the time domain features and the minimum distance involving arithmetic operations like addition, subtraction, multiplication, square root, division and logarithm on the *testing* data, were realized using the different transcendental functions of the CoOrdinate Rotation Digital Computer (CORDIC) algorithm.

### 7.3 Feature extraction

As discussed in Table 5.4, 10 one-dimensional features are computed on each individual accelerometer ( $AccX$ ,  $AccY$ ,  $AccZ$ ) and gyroscope ( $GyroX$ ,  $GyroY$ ,  $GyroZ$ ) data segment for each movement trial of each subject, which are as follows: 1) *standard deviation* ( $\sigma$ ), 2) *root mean square* ( $rms$ ), 3) *information entropy*, 4) *jerk metric*, 5) *peak number*, 6) *maximum peak amplitude*, 7) *absolute difference*, 8) *index of dispersion* ( $D$ ), 9) *kurtosis*, 10) *skewness*. Most of these features require the basic arithmetic operations like addition, subtraction, multiplication, division, square root and logarithm for their computation. Amongst these arithmetic operations, division, square root and logarithm require special attention for low-power implementation. With the recent advances in VLSI, several effective low-power design techniques have been proposed which include the non-restoring algorithm for division [213] and square root calculations [214] and the piecewise-

polynomial approximation for logarithm calculation [215]. These algorithms provide a good trade-off between accuracy and hardware complexity and hence have been widely employed in digital signal processing applications.

In terms of functional forms most of the features mentioned above have similarities to the different transcendental functions realizable using CoOrdinate Rotation Digital Computer (CORDIC) algorithm. Therefore, the design and implementation of a CORDIC-based low-power engine for computing the 10 features is presented here. The primary motivation for using the CORDIC algorithm is to explore its different transcendental functions and compute the complex arithmetic operations reusing the same architecture which can be implemented at low-cost with basic shift-add operations of the form  $a \pm b.2^{-i}$  [41]. CORDIC is a well-researched subject and several specialized architectural implementations [41], [216]–[220] of it have been proposed over the years which can be utilized for processing algorithms in low-power wireless sensor nodes. The fundamental mathematical processes of the above mentioned features have been formulated in terms of CORDIC and an optimized implementation strategy has been adapted by analysing their shared computational stages.

## 7.4 CORDIC fundamentals

CORDIC is an iterative algorithm for computing different transcendental functions using 2D vector rotation. A vector  $[x_0, y_0]$  can be rotated through an angle  $\theta$  to achieve the final component  $[x_1, y_1]$  through a series of micro-rotations in the clockwise or anti-clockwise direction, as illustrated in Figure 7.4.

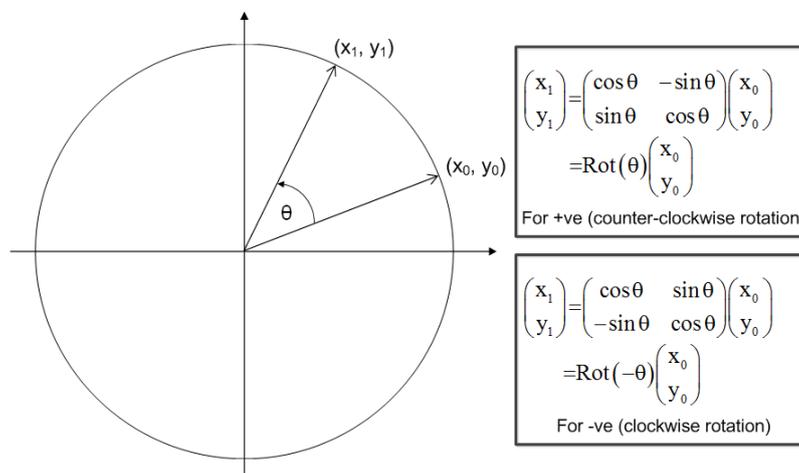


Figure 7.4: Overview of CORDIC vector rotation from  $[X_0, Y_0]$  to  $[X_1, Y_1]$  through an angle  $\theta$  using a series of micro-rotations in the clockwise or anti-clockwise direction.

The final vector is obtained by a matrix product of the initial vector with a rotation matrix  $Rot$  [41] as shown in Figure 7.4. CORDIC can be used to compute this vector rotation by employing the following iterative equation:

$$\begin{aligned}x_{j+1} &= x_j - \mu\sigma_j \cdot 2^{-j} \cdot y_j \\y_{j+1} &= y_j + \sigma_j \cdot 2^{-j} \cdot x_j \\z_{j+1} &= z_j - \sigma_j \cdot \alpha_j\end{aligned}\tag{7.1}$$

where  $[x_j, y_j]^T$  is the intermediate result vector;

$z_j$  is the residual angle;

$\sigma_j \in \{1, -1\}$  is the direction of vector rotation at the  $j$ -th iteration stage;

$\alpha_j$  is the pre-defined angle of rotation at each  $j$ -th iteration stage  $\{= \tan^{-1}(2^{-j})\}$  which add up to make the final target angle of rotation  $\theta$ ;

$\mu \in \{1, 0, -1\}$  being the coordinate of rotation – circular, linear and hyperbolic respectively.

Given an input vector  $[x_0 \ y_0]^T$ , in different coordinate system, CORDIC operates in two modes *viz.* rotation and vectoring, for computing a series of transcendental functions as shown in Table 7.1. In the rotation mode, starting with a vector  $[x_0 \ y_0]^T$  and a target rotation angle ( $z_0$ ) the objective is to compute the final coordinate  $[x_1 \ y_1]^T$  through a series of backward and forward rotation of the vector in an iterative manner to make the target angle zero ( $z_0 \rightarrow 0$ ).

In a similar manner, in the vectoring mode, where the target rotation angle is unknown, the vector  $[x_0 \ y_0]^T$  is rotated towards the  $x$ -axis through a finite number of iterations, so that the  $y$ -component approaches zero ( $y_0 \rightarrow 0$ ) [41]. These two types of operation while executed in different coordinate system ( $\mu$ ) circular, linear and hyperbolic, generates a series of transcendental functions as shown in Table 7.1. These forms of the transcendental functions, more specifically, those generated by the vectoring operation of CORDIC in different coordinate systems could be adapted for computing the target features.

In Table 7.1, the variables  $x_n$  and  $y_n$  correspond to the coordinates of the target vector  $[x_1 \ y_1]^T$  and  $K$  and  $K_h$  are the scale factors corresponding to the CORDIC operation in circular ( $\mu = 1$ ) and hyperbolic ( $\mu = -1$ ) coordinate systems respectively. As long as the number of CORDIC iterations is constant, the scale factor also remains a constant value. The final

result is scaled by a constant factor shown in (7.2), discussed here only for the vectoring operation of CORDIC in circular mode:

$$K = \prod_{i=0}^{b-1} \cos \alpha_i \quad (7.2)$$

where  $b$  is the word-length of the machine. The scale factor,  $K$  remains a machine constant as long as the index  $i$  runs through all of the values from 0 to  $b-1$ , i.e. at the end of all iterations. The sum of all angles of micro-rotations (output angle  $z_n$ ) is equal to the angle of rotation of the vector  $[x_0 \ y_0]^T$  while  $x_n$  corresponds to its magnitude. In this mode (circular) the decision about the direction of the micro-rotation depends on the sign of  $y_j$ . For a positive value of  $y_j$ ,  $\sigma_j = -1$  and for a negative  $y_j$ ,  $\sigma_j = 1$ .

$\mu$	ROTATION MODE ( $Z_0 \rightarrow 0$ )	VECTORING ( $Y_0 \rightarrow 0$ )
1	$x_n = K(x_0 \cos z_0 - y_0 \sin z_0)$ $y_n = K(y_0 \cos z_0 + x_0 \sin z_0)$ $z_n = 0$	$x_n = K\sqrt{x_0^2 + y_0^2}$ $y_n = 0$ $z_n = z_0 + \tan^{-1}(y_0/x_0)$
0	$x_n = x_0$ $y_n = y_0 + x_0 z_0$ $z_n = 0$	$x_n = x_0$ $y_n = 0$ $z_n = z_0 + (y_0/x_0)$
-1	$x_n = K_h(x_0 \cosh z_0 - y_0 \sinh z_0)$ $y_n = K_h(y_0 \cosh z_0 + x_0 \sinh z_0)$ $z_n = 0$	$x_n = K_h\sqrt{x_0^2 - y_0^2}$ $y_n = 0$ $z_n = z_0 + \tanh^{-1}(y_0/x_0)$

Table 7.1: Generalized CORDIC algorithm in three Co-ordinate systems [41].

In this investigation, the vectoring operation of CORDIC is used, represented by the operators  $Vec_c$ ,  $Vec_l$  and  $Vec_h$  in circular ( $\mu = 1$ ), linear ( $\mu = 0$ ) and hyperbolic ( $\mu = -1$ ) coordinate system respectively.

## 7.5 CORDIC formulation of features

For the formulation of the features, the input dataset is represented by  $d_{si}$ , where  $i \in \{0, 1, 2 \dots n-1\}$  and  $d_i$  is the output of vectoring CORDIC operation on  $d_{s(i-1)}$  data sample. With this convention the formulation of the target statistical features in terms of CORDIC operation is described in the following sections. It is important to note here that the features mean, absolute difference, peak number and maximum peak amplitude do not require the use of CORDIC functionalities, but they have also been presented here. The

feature mean ( $\mu$ ) was not in the original list of chosen features, but it has been included here since it is used for computing other features like  $rms$ ,  $\sigma$ ,  $D$ , kurtosis and skewness.

### 7.5.1 Mean ( $\mu$ )

The mean represents the average of a number of data samples calculated by accumulating  $n$  samples and dividing the resultant by  $n$ . If  $n = 2^m$ , where  $m$  is an integer, the final division can be achieved by  $m$  bit right shift of the result.

### 7.5.2 Root mean square (rms)

The  $rms$  is a measure of the signal energy normalized by the number of samples and is given by:

$$rms = \sqrt{\frac{1}{n} \left( \sum_{i=0}^{n-1} d_{si}^2 \right)} \quad (7.3)$$

In terms of the operator  $Vec_c$ ,  $rms$  computation could be represented as:

$$rms = \frac{1}{\sqrt{n}} \left( \prod_{i=0}^{n-1} Vec_c [d_i \quad d_{si}]^T \right)_x \quad (7.4)$$

Physically, (7.4) means that  $d_{si}$  are fed in the  $y$  input of the CORDIC while the  $x$ -component of the output is fed back to the  $x$ -component of its input. Therefore, at every clock cycle as the new data sample  $d_{si}$  arrives the computed  $x$ -component of the CORDIC is given by:

$$d_i = K \sqrt{d_{s0}^2 + d_{s1}^2 + \dots + d_{s(i-1)}^2} \quad (7.5)$$

After every complete CORDIC operation the  $x$ -component of the output is scaled with the scale factor  $K$ . If uncompensated, feeding back this result into the  $x$ -component of the CORDIC input will result in accumulation of this scale factor corresponding to each  $d_{si}$  and therefore (7.5) will not hold true. To avoid this problem, after every complete CORDIC operation (comprising of  $N$  stages) with a set of input data, the scale factor compensation

step needs to be invoked before feeding this output to the  $x$ -input of the CORDIC for the next iteration. With this scale factor compensation step in place, after  $n$  number of operations the final result at the  $x$  output of the CORDIC is multiplied with  $1/\sqrt{n}$  to obtain the true result of  $rms$ . However, since  $n$  is a fixed number the value of  $1/\sqrt{n}$  could be pre-computed and finally multiplied with the CORDIC output using a reduced complexity fixed-number multiplier or multiplier-less shift-and-add technique.

### 7.5.3 Standard deviation ( $\sigma$ )

$\sigma$  represents the variation of the data samples from the mean and is expressed as:

$$\sigma = \sqrt{\frac{1}{n} \left( \sum_{i=0}^{n-1} (d_{si} - \mu)^2 \right)} \quad (7.6)$$

As can be seen from the functional similarity of (7.3) and (7.6) the formulation shown in (7.4) can be reformulated for computing  $\sigma$  in terms of CORDIC operation as:

$$\sigma = \frac{1}{\sqrt{n}} \left( \prod_{i=0}^{n-1} Vec_c[d_i \quad (d_{si} - \mu)]^T \right)_x \quad (7.7)$$

Similar to the  $rms$  computation, here the  $x$  component of the CORDIC output needs to be multiplied with the pre-computed value of  $1/\sqrt{n}$  to obtain the true value of  $\sigma$  which again can be achieved by a reduced complexity fixed-number multiplier or shift-and-add technique. Like  $rms$ , here also the scale factor compensation step needs to run after each complete CORDIC operation.

### 7.5.4 Index of dispersion ( $D$ )

It is a normalized measure of the dispersion of a data distribution, expressed as:

$$D = \frac{\sigma^2}{\mu} \quad (7.8)$$

In terms of CORDIC operation  $D$  may be formulated as:

$$D = \left( \text{Vec}_l[\mu \quad \sigma]^T \right)_z \times \sigma \quad (7.9)$$

Referring to Table 7.1, setting  $\mu$  and  $\sigma$  as the  $x_0$  and  $y_0$  inputs to the CORDIC, operating in vectoring mode in the linear coordinate system, the output will result in  $(\sigma/\mu)$ . This output is then multiplied with  $\sigma$  to obtain the desired value of  $D$  using a multiplier.

### 7.5.5 Kurtosis

Kurtosis is a normalized measure of the dispersion of a data distribution, as expressed in (7.10) and re-framed in (7.11):

$$\text{kurtosis} = \frac{1}{n} \left( \sum_{i=0}^{n-1} (d_{si} - \mu)^4 / \sigma^4 \right) \quad (7.10)$$

$$\text{kurtosis} = \frac{1}{n} \left[ \left\{ (d_{s0} - \mu) / \sigma \right\}^4 + \dots + \left\{ (d_{si} - \mu) / \sigma \right\}^4 \right] \quad (7.11)$$

For each sample  $(d_{si} - \mu)$ , the operator  $\text{Vec}_l$  produces the output  $[(d_{si} - \mu) / \sigma]$  when  $(d_{si} - \mu)$  and  $\sigma$  are set as the  $x_0$  and  $y_0$  inputs to the CORDIC. Two squaring circuits and an accumulator module are then used followed by multiplying it with the pre-computed value of  $1/n$  to achieve the desired value of kurtosis as shown in Figure 7.5.

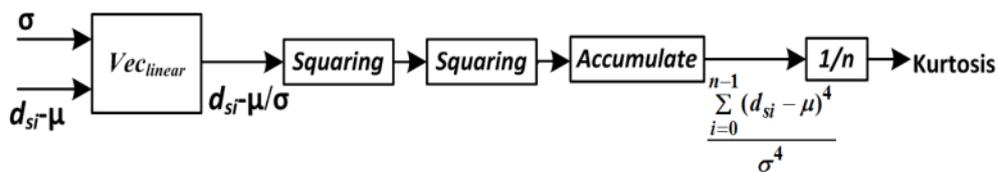


Figure 7.5: Architecture for computing kurtosis.

### 7.5.6 Skewness

Skewness is a measure of the alignment of the probability distribution of a real valued random variable to one side of the mean and mathematically defined as:

$$\text{skewness} = \frac{1}{n} \left( \sum_{i=0}^{n-1} (d_{si} - \mu)^3 / \sigma^3 \right) \quad (7.12)$$

Exploiting the functional similarity of (7.10) and (7.12), the overall architecture for computing *skewness* is shown in Figure 7.6.

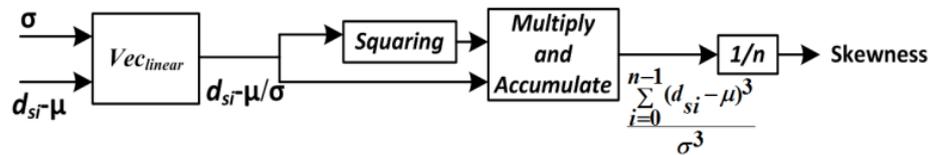


Figure 7.6: Architecture for computing skewness.

### 7.5.7 Absolute difference (abs. diff)

It is the absolute difference between the maximum value and the minimum value of a signal and is given by:

$$abs.diff = abs(\max(d_{si}) - \min(d_{si})) \quad (7.13)$$

This computation can be achieved by using a maximum and minimum detection circuit as shown in Figure 7.7 and then taking the absolute difference of the values corresponding to them.

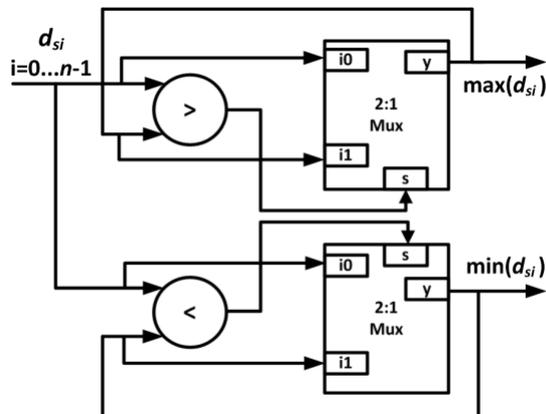


Figure 7.7: Architecture for computing maximum and minimum values of a signal.

### 7.5.8 Information entropy (inf. entropy)

Information entropy is a measure of the randomness present in a signal represented by [168]. To compute it, the histogram plot representing the distribution of data sample is first divided into a number of bins and then the probability distribution is computed by counting the number of samples (frequency of observations) in each bin. For ease of implementation

an approximation was applied where the signal  $d_{si}$ , is divided into four bins ( $bin_k$ , where  $k \in \{0, 1, 2, 3\}$ ) between the *minimum* and *maximum* values of the signal as shown in Figure 7.8 considering a Gaussian distribution as an example.

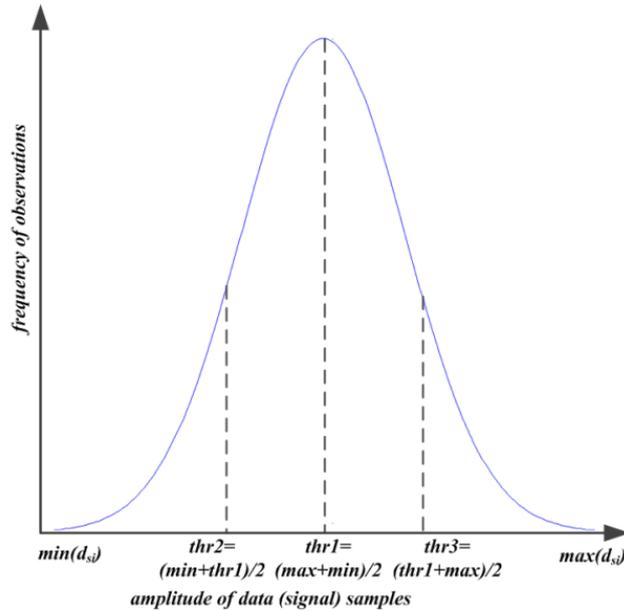


Figure 7.8: The selected bins from normal distribution of data samples.

$$Info.Entropy = - \sum_{i=1}^n p(d_{si}) \log_2 p(d_{si}) \tag{7.14}$$

In hardware, the bin thresholds could be computed by simple add-shift mechanism. The architecture for computing the probability of the signal in each of the four bins is illustrated in Figure 7.9. A comparator logic is used to find out the appropriate bin in which each sample  $d_{si}$  belongs and accordingly for each of the bins a sample counter is used to compute the total number of samples lying in it. The sample count in each bin is multiplied with the pre-computed value of  $1/n$  to calculate the probability of data samples  $p(bin_k)$ .

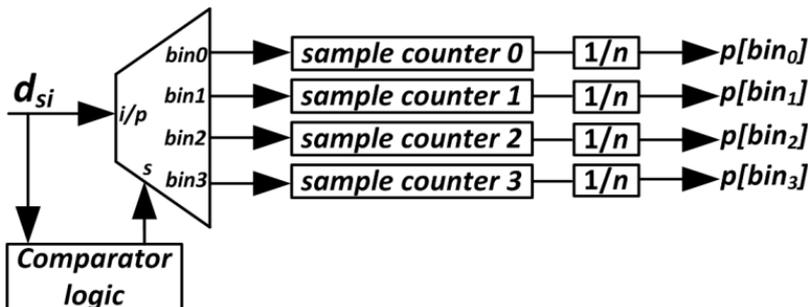


Figure 7.9: Architecture for computing the probability of each bin of the signal.

The logarithm of the respective probability for each bin can be calculated using the CORDIC operator  $Vec_n$  as shown in (7.15). It should be noted that CORDIC computes the

natural logarithm (base  $e$ ), which is further multiplied with a constant scale factor to obtain  $\log_2$  (i.e. base 2). Therefore altogether four CORDIC operations are needed for computing logarithm of the probability corresponding to the four bins. Accordingly (7.14) could be realized using the block diagram shown in Figure 7.10.

$$\ln p(\text{bin}_k) = \left( \text{Vec}_h \left[ \begin{matrix} 1+p(\text{bin}_k) & 1-p(\text{bin}_k) \end{matrix} \right]^T \right)_z \quad (7.15)$$

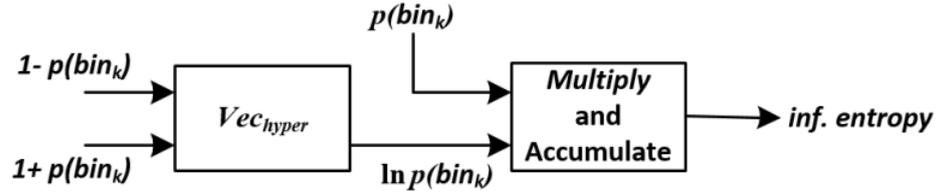


Figure 7.10: Architecture for computation of information entropy.

### 7.5.9 Jerk metric

The *jerk metric* characterizes the average rate of change of acceleration in a movement. It is calculated as the *rms* value of the derivative of the acceleration (jerk) normalized by the maximum value of the integral (velocity) as shown in (7.16). It is important to note here that although the calculation of jerk is physically related to the acceleration data but the same computing logic is also applied to the rotation data from the gyroscope. This is because the *jerk metric* calculated on the gyroscope data serves its purpose as a discriminating feature for characterising and recognising the movements.

$$\text{jerk metric} = \frac{\text{rms} \left[ \frac{d(d_{si})}{dt} \right]}{\max \left[ \int (d_{si}) dt \right]} \quad (7.16)$$

The computation of *jerk metric* requires the CORDIC operations  $\text{Vec}_c$  and  $\text{Vec}_l$ . Since the data samples are equally spaced due to the constant sampling frequency, the first derivative is computed as the difference of the consecutive data samples using a subtractor. The integral of the data is computed using trapezoidal integration which involves the addition of the consecutive data samples and a divide by 2 (implemented as a one-bit right shift).

From (7.4), it can be deduced that the *rms* of the first derivative of the data samples ( $\dot{d}_{si}$ ) can be computed using the operator  $\text{Vec}_c$ , which is shown in (7.17). The samples  $\dot{d}_{si}$  are

fed in the  $y$  input of the CORDIC while the  $x$ -component of the output is fed back to the  $x$ -component of its input. After every complete CORDIC operation the  $x$ -component of the output is scaled with a factor  $K$ .

$$rms = \frac{1}{\sqrt{n}} \left( \prod_{i=0}^{n-1} Vec_c[d_i \quad \dot{d}_{si}]^T \right)_x \quad (7.17)$$

$$jerk\ metric = \left( Vec_l[\max(\int d_{si}) \quad rms(\dot{d}_{si})]^T \right)_z \quad (7.18)$$

The *jerk metric* is finally computed using the CORDIC operator  $Vec_l$  as shown in (7.18). Referring to Table 7.1,  $\max(\int d_{si})$  and  $rms(\dot{d}_{si})$  are set as the  $x_0$  and  $y_0$  inputs to the CORDIC, operating in vectoring mode in the linear coordinate system.

#### 7.5.10 Peak number (peaks)

It represents the tremor during arm movement and will ideally decrease as the patient improves. The *peak number* is obtained from gradient analysis of the data samples ( $d_{si}$ ). The difference of the consecutive data samples is computed using a subtractor. Two consecutive differences are compared (using a comparator) against a pre-defined threshold to count the number of peaks using a sample counter.

#### 7.5.11 Maximum peak amplitude (max\_mag)

It is a measure of the amplitude of the peaks obtained after gradient analysis. It represents the magnitude of the samples where the peaks lie. Hence, this is calculated along with the peak number.

### 7.6 Architecture and evaluation

From the foregoing section, it is clear that the actual CORDIC operation is needed only for the features root mean square ( $rms$ ), standard deviation ( $\sigma$ ), index of dispersion ( $D$ ), kurtosis, skewness, information entropy and jerk metric. In the below section on

architecture the mean ( $\mu$ ) and the *abs. diff* are also included since they are used for computing other features.

### 7.6.1 Architecture

Typically CORDIC is implemented in two ways: iterative and pipelined. The iterative CORDIC architecture utilises a single implementation of (7.1) and computes the final result in  $b$  iterations where  $b$  is governed by the required accuracy and the word-length. Therefore,  $b$  clock cycles are required for completion of one CORDIC operation. The pipelined architecture overcomes this problem by exploiting the identical nature of the CORDIC iterations (shift/add operations) and mapping them onto a pipelined architecture. The first output of a  $N$ -stage pipelined CORDIC is obtained after  $N$  clock cycles (the latency period) and thereafter the outputs will be generated at each clock, helping in achieving a high throughput and is therefore the most popular approach for CORDIC implementation.

The equations (7.4) and (7.5) suggest a tight computing recursion, which indicates a computing loop in the corresponding signal flow graph (c.f. Figure 7.11). Any attempt to pipeline the computation in the loop would lead to inaccurate result and therefore, in this particular application, the mathematical formulations described in section 7.5 cannot be realized using the pipelined CORDIC approach. Without loss of generality, this can be explained by the following example by considering a 4-stage pipelined CORDIC for computing *rms* with a dataset of 8 samples,  $\{d_{s0}, \dots, d_{s7}\}$ . A cycle-by-cycle snapshot of the process, is presented in Figure 7.11, where  $x_{i/p}$ ,  $y_{i/p}$  are the  $x$  and  $y$  component of the CORDIC input and  $x_{o/p}$  is the  $x$ -component of the output,  $d_{si}$  is the input data sample,  $d_i$  is the CORDIC output of the  $d_{s(i-1)}$  data sample.

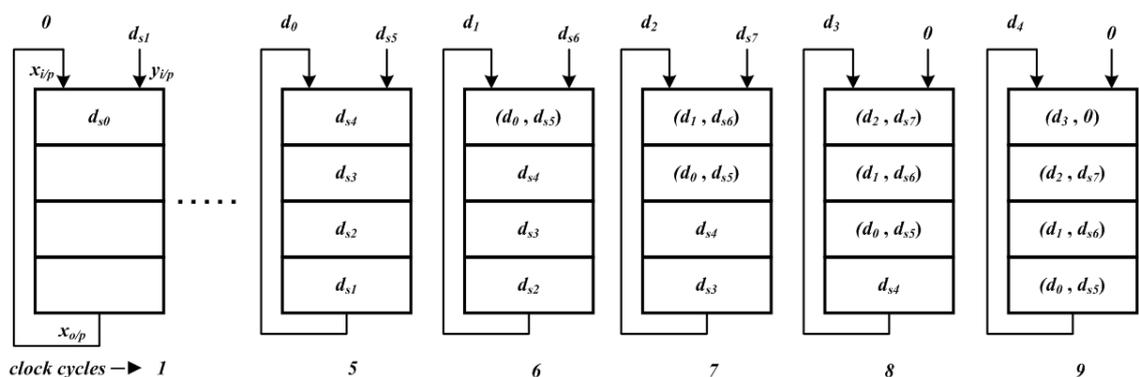


Figure 7.11: Pipelined architecture for a 4-stage CORDIC for computing *rms* on a data stream having eight samples.

In this operation, the final target operation is given by (7.19).

$$rms = \sqrt{\frac{1}{8} [d_{s0}^2 + d_{s1}^2 + \dots + d_{s7}^2]} \tag{7.19}$$

It is clearly evident from Figure 7.11 that the expression in (7.19) cannot be computed using this pipelined architecture. Moreover, if extra registers are used to store the intermediate result of each pipeline stage, it would nullify the advantage of the pipelined architecture and result in increased complexity in the control mechanism and the associated hardware. On the other hand, although the mathematical form in (7.19) can be realized using iterative CORDIC, the overall time required will be significantly high and hence the throughput will suffer. Therefore, to overcome this problem a unit latency design, which coalesce all iterations into one computing stage (clock cycle), is adopted in this work. Carry-save arithmetic (CSA) technique enabling a complete CORDIC operation in one clock cycle is employed [221]. Since, the delay of one CSA adder is equivalent to one carry propagation, for an  $N$ -stage CSA-CORDIC, the overall propagation delay for computing one complete operation is equivalent to  $N$  number of carry propagation delay (equivalent to an  $N$ -bit ripple carry adder) and therefore is achievable in one clock cycle. Accordingly a CSA-based CORDIC has been used in the proposed design. The overall architecture of the statistical feature computation engine is shown in Figure 7.12.

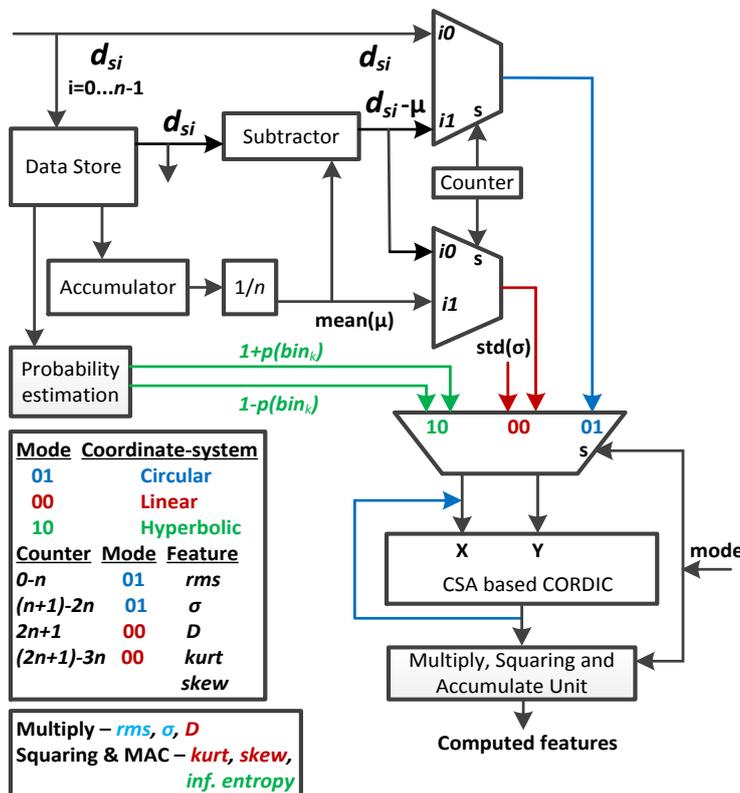


Figure 7.12: Architectural overview of the CORDIC operation for feature extraction.

In principle, the engine consists of a CSA-based CORDIC module, a subtractor, an accumulator, a probability estimator (as shown in Figure 7.9) and a multiply-squaring-accumulate unit. In order to maintain an acceptable level of accuracy (in this case 16-bit) the CORDIC was implemented with 24-bit datapath following the principles described in [219]. A 2-bit *mode* signal (01: circular, 00: linear and 10: hyperbolic) is used to enable the CORDIC operation in different coordinate systems. A control counter is used to input the appropriate data ( $d_{si}$  - for *rms*,  $(d_{si} - \mu)$  - for  $\sigma$  and  $(d_{si} - \mu)$ ,  $\mu$  and  $\sigma$  - for *D*, *kurtosis*, *skewness*) to the CORDIC module at appropriate clock cycle. The components for computing *jerk metric* have not been shown in Figure 7.12 for the sake of clarity but have been described in detail.

For computing  $\mu$ ,  $d_{si}$  are initially stored in the register bank ‘Data Store’ from which they are sequentially passed onto the accumulator and finally after  $n$  number of clock cycles the result is multiplied by the pre-computed value of  $1/n$ .

For *rms* computation, in the first cycle the raw samples  $d_{s0}$  and  $d_{s1}$  are fed into the  $x$  and  $y$  inputs of the CORDIC (*Vec<sub>c</sub> mode*–01, shown in blue in Figure 7.12). The subsequent  $x$ -component of the output of the CORDIC is fed back into its  $x$  input and the next sample  $d_{s2}$  into the  $y$  input of the CORDIC and this process is repeated for  $n$  number of clock cycles while the end result is multiplied by the pre-computed value of  $(1/\sqrt{n})$  to achieve the desired *rms* value.

For computing  $\sigma$ ;  $\mu$  is subtracted from  $d_{si}$  using a subtractor and is fed into the CORDIC module to compute the resulting expression  $\sqrt{\sum_{i=0}^{n-1} (d_{si} - \mu)^2}$  which is multiplied by  $(1/\sqrt{n})$  to obtain the true value of  $\sigma$  at the end of  $2n$  clock cycles.

For computing Dispersion (*D*) the values  $\mu$ ,  $\sigma$  are used as inputs to the CORDIC (*Vec<sub>l</sub> mode*–00, shown in red in Figure 7.12) to compute  $(\sigma/\mu)$  in one cycle which is multiplied by  $\sigma$  to obtain *D*.

Each sample  $(d_{si} - \mu)$  and  $\sigma$  are used as inputs to the CORDIC (*Vec<sub>l</sub>*) to generate the expression  $[(d_{si} - \mu)/\sigma]$  which is used for calculating *Kurtosis* and *Skewness* in  $n$  cycles as shown previously in Figure 7.5 and Figure 7.6 respectively. Hence,  $3n$  clock cycles are required for computing the *kurtosis* and *skewness*.

For computing the *information entropy*, first the values  $p(bin_k)$  is computed for each bin using the probability estimation block and then a pair of adder and subtractor are used for computing  $[1 - p(bin_k)]$  and  $[1 + p(bin_k)]$  which are used as inputs to the CORDIC ( $Vec_h$  mode-10, shown in green in Figure 7.12) and thereby computing  $\ln[p(bin_k)]$  at every clock cycle. Finally, the result is multiplied with a constant scale factor (for computing  $\log_2$ ) and  $p(bin_k)$  and accumulated using a MAC unit as shown in Figure 7.10.

For computing the *jerk metric*, as discussed in section 7.5.9, the samples  $(\dot{d}_{si})$  are used as input to the CORDIC ( $Vec_c$  mode-01) to compute  $\sqrt{\sum_{i=0}^{n-1} (\dot{d}_{si} - \mu)^2}$  which is multiplied by  $(1/\sqrt{n})$  to obtain the *rms* of the *jerk* value in  $n + 1$  clock cycles (one clock cycle is used for computing the first derivative using a subtractor). The max of the first integral is computed in  $n + 1$  cycle. The CORDIC operation  $Vec_l$  is then used to compute the jerk metric obtained in the  $(n + 2)$ -th cycle. The peak number and the maximum peak amplitude involves the computation of the gradient (subtraction of data samples  $d_{si}$ ) and a comparison with a pre-set threshold, and therefore takes  $(n + 2)$  clock cycles.

In this implementation, a signal length of 256 data samples has been considered which can be represented on a dyadic scale and therefore any multiplication or division can be implemented through a shift operation. Following the procedure described above the  $\mu$  and *rms* are computed in 256 ( $n$ ) clock cycles while the  $\sigma$  computation takes 512 ( $2n$ ) clock cycles altogether. The value of  $\sigma$  is used to compute  $D$  in the 513-th ( $2n+1$ ) cycle and *Kurtosis* and *Skewness* are generated in the 768-th ( $3n$ ) cycle. *Information entropy* is independent of the other features and requires 4 clock cycles as the logarithm of the probability of each bin is computed in one CORDIC operation. However, the maximum and the minimum computation takes 256 ( $n$ ) clock cycles followed by the threshold computation for probability estimation. Therefore, the information entropy is computed in the 262 ( $n + 6$ ) clock cycles. The *jerk metric* is computed in 258 ( $n + 2$ ) clock cycles. The peak number and the maximum peak amplitude are calculated in 258 clock cycles. The computation time for each feature is listed in Table 7.2.

Features	Number of clock cycles
$\mu$	256 ( $n$ )
rms	256 ( $n$ )
Abs. difference	257 ( $n + 1$ )
Jerk metric	258 ( $n + 2$ )
Peak number/ max peak amp.	258 ( $n + 2$ )
Information entropy	262 ( $n + 6$ )
$\sigma$	512 ( $2n$ )
$D$	513 ( $2n + 1$ )
Kurtosis/Skewness	768 ( $3n$ )

Table 7.2: Computation time for the features employing the CORDIC engine.

### 7.6.2 Hardware complexity analysis of proposed architecture

A hardware complexity analysis is presented considering a generalized word-length  $b$  of the  $N$ -stage CORDIC module for a single iteration. The hardware resource for one iteration of CORDIC can be reused for multiple iterations (for example, *rms* computation), applicable for all three modes of operation. The complexity is described in terms of the total number of full adders (FA) used. It is important to note here that only those features have been considered which employ the CORDIC module and the operations  $(1/\sqrt{n})$  or  $(1/n)$  have not been included in this estimation since it can be pre-computed. The arithmetic operations with respect to the feature computation is summarised in Table 7.3

Features	CORDIC	Multiplication	Addition/Subtraction	Accumulator	Squaring
<i>rms</i>	1				
$\sigma$	1		1		
$D$	1	1			
<i>kurtosis</i>	1		1	1	2
<i>skewness</i>	1	1	1	1	1
<i>inf. entropy</i>	1	1	5	1	
<i>jerk metric</i>	1		2		

Table 7.3: Arithmetic operations required in the proposed CORDIC-based architecture for feature extraction

- *rms* – CORDIC,
- $\sigma$  - 1 subtractor for computing  $(d_{si} - \mu)$  and CORDIC,
- $D$  – CORDIC and 1 multiplier for multiplying  $(\sigma/\mu)$  with  $\sigma$ ,
- *kurtosis* – 1 subtractor for computing  $(d_{si} - \mu)$ , CORDIC, 2 squaring units and 1 accumulator block,
- *skewness* - 1 subtractor for computing  $(d_{si} - \mu)$ , CORDIC, 1 squaring unit, 1 multiplier and 1 accumulator block,

- *inf. entropy* - 3 adder/subtractor for calculating the bin thresholds for computing the respective probabilities of each bin, 2 adder/subtractor for calculating the inputs  $[1 - p(bin_k)]$  and  $[1 + p(bin_k)]$  for computing  $\log_2 p(bin_k)$ , hence 5 add/sub operations in total. 2 adder/subtractors can be considered in total which can be used for computing both the inputs to the CORDIC and also the bin thresholds. Besides needing the CORDIC, 1 multiplier and accumulator for multiplying  $\log_2 p(bin_k)$  and  $p(bin_k)$  in computing the information entropy is also required as shown Figure 7.10.

- *Jerk metric* – 1 subtractor and CORDIC for computing  $rms\left(\dot{d}_{si}\right)$  and 1 adder for computing  $\int d_{si}$  by trapezoidal integration. Finally CORDIC ( $Vec_I$ ) is used for computing the value of the feature.

As mentioned in the architectural implementation, the features  $rms$ ,  $\sigma$ ,  $D$  and *kurtosis/skewness* are computed sequentially, due to their functional dependencies. However, *kurtosis* and *skewness* can be computed in parallel. The computation of *inf. Entropy* and *jerk metric* is independent of any of the above features. The computation of *inf. Entropy* is dependent on the minimum and maximum computation for obtaining the thresholds and thereby has to wait for  $n$  clock cycles. Similarly the computation of the *jerk metric* is dependent on the  $rms$  of the derivative and maximum of the integral which takes  $n + 1$  cycles.

In view of this operational sequence, one can reuse majority of the arithmetic components thereby saving hardware. Considering such resource sharing, the optimal list of hardware components required for computing all the six mentioned features are:

- the CORDIC module can be reused for computing all the features based on their formulation and architecture as mentioned in section 7.5,
- 1 subtractor is required for computing  $(d_{si} - \mu)$  for the features:  $\sigma$ , *kurtosis*, *skewness*. This subtractor can be reused in the computation of *inf. entropy* (requiring 2 add/sub operations), thereby requiring 1 additional adder/subtractor. These 2 adder/subtractor can be reused for computing *jerk metric*,
- 1 multiplier can be reused for  $D$ , *skewness* and *inf. entropy*,
- 2 accumulator blocks are needed since the computation of *kurtosis* and *skewness* takes place in parallel and can be reused for *inf. entropy*,
- 3 squaring units are needed for *kurtosis* and *skewness*.

For the sake of convenience, 2 squaring units can be considered as 1 multiplier, therefore requiring 2.5 multipliers in total (1 CAM + 3 squaring units). A conventional array multiplier (CAM) requires  $b(b - 2)$  FA,  $b$  half adders (HA) and  $b^2$  AND gates [222]. Considering, 2 HA as 1 FA and 4 AND gates as 1 FA [222] (due to transistor count and area), the total gate count of a CAM can be reduced to  $(1.25b^2 - 1.5b)$  FA. Hence, for 2.5 multipliers,  $A_{mult} = 2.5(1.25b^2 - 1.5b)$  FA, where  $(A^*)$  represents the total number of FA's in each circuit. An accumulator block can be considered to comprise of a FA (any registers associated with the accumulator are not considered, thereby accounting for mathematical operations only). A  $b$ -bit Ripple carry adder/subtractor (RCA) requires  $b$  full adders (FA). Therefore in total  $A_{add/sub} = 4b$  FA (2 adder/subtractors + 2 accumulators) are required. A  $N$  stage  $b$ -bit CORDIC implemented using Carry-Save Arithmetic (CSA) requires  $6Nb$  FA. For this case,  $N=16$ , hence the CORDIC module requires  $A_{CORDIC} = 96b$  FA. Therefore the total gate count for the implementation of these seven features (let us name it as *archit1*) in terms of FA count is  $A_{archit1} = (A_{mult} + A_{add/sub} + A_{CORDIC}) = (3.125b^2 + 96.25b)$  FA. Hence, for a 24-bit datapath,  $A_{archit1} = 4110$  FA.

### 7.6.3 Hardware complexity analysis of non-CORDIC architecture

Now let us consider an alternative architecture (*archit2*) for computing the same six features (*rms*,  $\sigma$ ,  $D$ , *kurtosis*, *skewness* and *inf. entropy*), also summarized in Table 7.4. In this architecture a Ripple carry adder (RCA), conventional array multiplier (CAM), non-restoring iterative cellular square rooter (SQRT), non-restoring array divider (NAD) and multiplicative normalization based logarithm [223] are considered as the arithmetic components for implementing the fundamental mathematical operations. Similar to the proposed CORDIC based design (*archit1*), the operations  $(1/\sqrt{n})$  or  $(1/n)$  are not considered in this estimation.

Features	Division	Multiplication	Addition/Subtraction	Square root	Accumulator	Squaring
<i>rms</i>				1	1	1
$\sigma$			1	1	1	1
$D$	1	1				
<i>kurtosis</i>	1		1		1	2
<i>skewness</i>	1	1	1		1	1
<i>inf. entropy</i>		1	9		1	
<i>jerk metric</i>	1		2	1	1	1

Table 7.4: Arithmetic operations required in non-CORDIC-based architecture for feature extraction

- $rms$  – 1 squaring unit, 1 SQRT and 1 accumulator,
- $\sigma$  - 1 subtractor for computing  $(d_{si} - \mu)$ , 1 squaring unit, 1 SQRT and 1 accumulator,
- $D$  – 1 NAD and 1 CAM,
- $kurtosis$  – 1 subtractor for computing  $(d_{si} - \mu)$ , 2 squaring units, 1 NAD and 1 accumulator block,
- $skewness$  - 1 subtractor for computing  $(d_{si} - \mu)$ , 1 squaring unit, 1 NAD, 1 CAM and 1 accumulator block,
- $inf. entropy$  - 3 adder/subtractor for calculating the bin thresholds for computing the respective probabilities of each bin. For computing the logarithm of each bin,  $\log_2 p(bin_k)$ , 2 variable shifters, a 4:2 adder, a 3:2 adder, a carry propagate adder (CPA), selection module, 2 multiplexers, a look-up table (LUT) and 4 registers [223] are needed. A 4:2 adder corresponds to  $3b$  FA, a 3:2 adder can be realized by  $2b$  FA and a  $b$ -bit CPA requires  $b$  FA, hence requiring  $6b$  FA in total. In addition, 1 CAM is required for multiplying the probability of each bin,  $p(bin_k)$  with its logarithm  $\log_2 p(bin_k)$ . It is important to mention here that the consideration is for one iteration of the logarithm computation. In the end, an accumulator block is used for adding the product  $[p(bin_k) * \log_2 p(bin_k)]$  for each bin.
- $jerk metric$  - 1 subtractor for computing  $\dot{d}_{si}$ ; 1 squaring unit, 1 SQRT and 1 accumulator for computing  $rms\left(\dot{d}_{si}\right)$ ; 1 adder for computing  $(\int d_{si})$ ; and finally 1 NAD.

Most of the components can be reused keeping in view the sequential nature of the feature computation with an exception for  $kurtosis$  and  $skewness$  which are computed in parallel after computing  $\sigma$  and the computation of  $inf. Entropy$  and  $jerk metric$  are independent of the other features. Therefore the optimal list of hardware components required for computing the features are:

- 2 SQRT – 1 SQRT for computing  $rms\left(\dot{d}_{si}\right)$  for  $jerk metric$  and 1 SQRT can be reused for  $rms$  and  $\sigma$ ,
- 1 subtractor is required for computing  $(d_{si} - \mu)$  for the features:  $\sigma$ ,  $kurtosis$ ,  $skewness$  and for computing  $\dot{d}_{si}$  for  $jerk metric$ .
- 1 adder for computing  $(\int d_{si})$  for  $jerk metric$ .

These adder/subtractor can be reused for computing the bin thresholds in  $inf. Entropy$ .

- For computing logarithm, considering  $m$  iterations,  $m6b$  FA are required. For the sake of convenience, considering  $m=16$  (similar to the stages in CORDIC), thereby requiring  $96b$  FA.
- 2 accumulator blocks are needed since the computation of *kurtosis* and *skewness* takes place in parallel, they can be reused for *rms*,  $\sigma$  and *inf. entropy*,
- 1 CAM can be reused for  $D$ , *skewness* and *inf. entropy*,
- 3 squaring units required for *kurtosis* and *skewness* which are computed in parallel, one of the squaring units can be reused for *rms* and  $\sigma$ .
- 2 NAD for *kurtosis* and *skewness*, out of which one can be reused for computing *jerk metric* and  $D$ .

Therefore, 2.5 multipliers are required in total (1 CAM + 3 squaring units), hence  $A_{mult} = 2.5(1.25b^2 - 1.5b)$  FA. The total count for adder/subtractor,  $A_{add/sub} = 100b$  FA (2 adder/subtractor + 2 accumulators +  $96b$  FA). A  $b \times b$  NAD requires  $0.5 \times b(3b - 1)$  FA and  $0.5 \times b(3b - 1)$  XOR gates. Two  $b$ -bit SQRT requires  $0.25 \times b(b + 6)$  FA and XOR gates [222]. Therefore, the total FA count:  $A_{NAD} = (4.5b^2 - 1.5b)$  FA,  $A_{SQRT} = (0.375b^2 - 2.25b)$  FA. The total gate count for computing these seven features using an alternate architecture (*archit2*) in terms of FA count is  $A_{archit2} = (A_{mult} + A_{add/sub} + A_{NAD} + A_{SQRT}) = (8b^2 + 92.5b)$  FA. Hence, for a 24-bit datapath,  $A_{archit2} = 6828$  FA. It is important to note here that for the complexity analysis of computing logarithm, the selection module, 2 multiplexers, a look-up table (LUT) and 4 registers were not considered. The variable shifters were also left out since it cannot be ascertained. This is similar to the assumptions for the proposed CORDIC based design where the comparator and counter logic for computing the probability of each bin  $p(bin_k)$  were not considered.

A comparative analysis has been presented in Figure 7.13, for both the architectures (*archit1* – CORDIC based operation, *archit2* – without using CORDIC, considering  $m = 8, 12$  and  $16$  in logarithm computation), varying the word-length, to show the number of FA being used for computing the seven features. This clearly shows the effectiveness of the proposed CORDIC based feature computation engine in terms of hardware complexity which is further prominent with an increase in word-length. To the best of knowledge there is no unified architecture or design for computing the statistical features considered in this work. Therefore, for the alternate architecture (*archit2* – without CORDIC), a Ripple carry adder (RCA), conventional array multiplier (CAM), non-restoring iterative cellular square rooter (SQRT), non-restoring array divider (NAD) and multiplicative normalization based

logarithm are considered to compute the required features and a hardware complexity analysis is provided in terms of basic arithmetic operations i.e. full adder (FA) counts. This complexity analysis presents a more objective reflection of the qualitative difference between the two architectures. Hence in the next section the synthesis and verification results for only the proposed CORDIC based design has been presented.

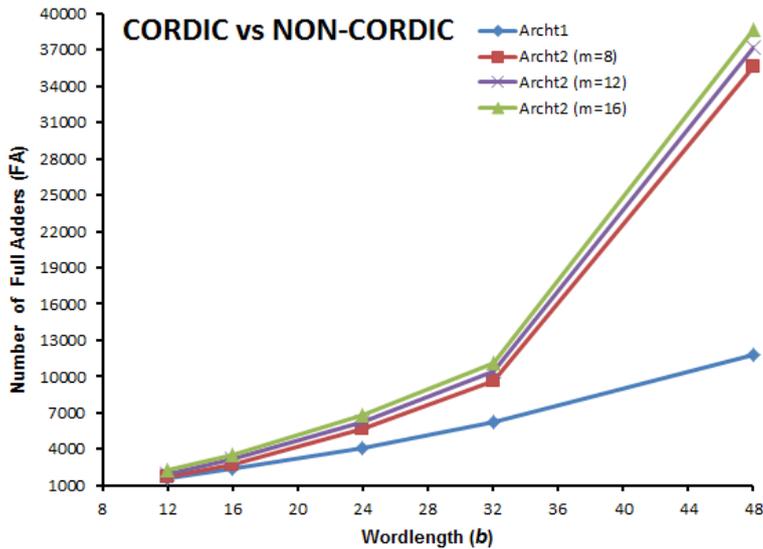


Figure 7.13: Comparison of hardware complexity for a CORDIC and non-CORDIC based architecture for feature extraction, showing variation in the number of full adders (FA) required with change in word-length.

#### 7.6.4 Synthesis and verification

Having established the advantage of the proposed design in terms of the hardware complexity, the architecture was coded using Verilog as the HDL. To verify the functionality of the feature computation engine five datasets of 256 samples in the range of -20 to +20 were randomly generated, and the target features were computed using Matlab. The Verilog output on the same dataset are then compared with the Matlab results and the average errors are calculated as shown in Table 7.5. It is evident that the average error may become significant for the features particularly involving higher-order terms even when the accuracy of the CORDIC itself is set high. To achieve higher accuracy, therefore, adjusting the datapath width for the MAC unit may be necessary depending on the error tolerance of an application.

The design was synthesized using STMicroelectronics 130-nm technology library with a supply voltage of 1.08V and a clock frequency of 50 Hz. The 2-input NAND gate equivalent cell area of the synthesized design was 159 K. The dynamic power for the

synthesized design was 6.4 nW@50 Hz. The design was synthesized at 50 Hz, in view of the application where the sensors are sampled at very low frequencies (20 ~ 50 Hz), but the design was functionally verified at higher clock frequencies (up to 75 MHz) for high speed applications. The engine computes all the features sequentially in  $3n$  clock cycles and could also be utilized for stand-alone feature computation.

Features	Average Error
Mean	0
Absolute Diff	0
RMS	$O(2^{-8})$
Standard Deviation	$O(2^{-10})$
Index of Dispersion	$O(2^{-11})$
Kurtosis	$O(2^{-8})$
Skewness	$O(2^{-12})$
Entropy	$O(2^{-12})$
Jerk	$O(2^{-10})$
Peaks	0
Max_mag	0

Table 7.5: Average error between Matlab and RTL simulation for feature extraction

## 7.7 Minimum distance computation

Having implemented the feature computation engine, the architecture for computing the minimum distance classifier is presented here. The offline-online processing approach as discussed in section 7.2 has been illustrated in Figure 7.14, representing the input-output signal names.

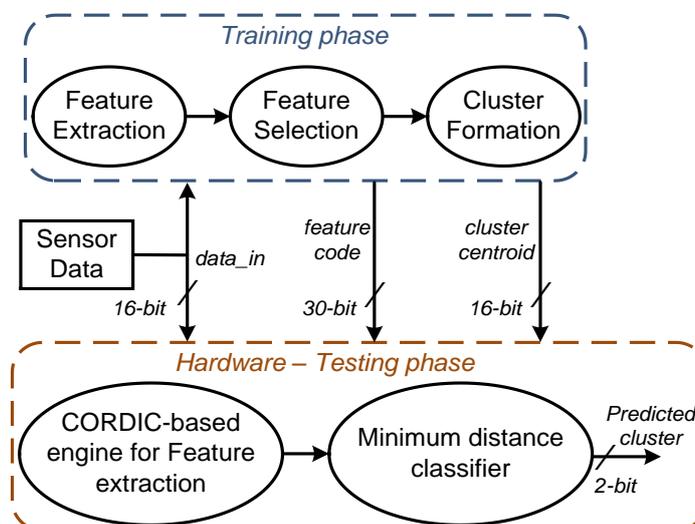


Figure 7.14: Design overview for the offline-online processing involving clustering and minimum distance classification.

The important signals used by the hardware module are:

- *data\_in* - 16-bits input for tri-axial sensor data corresponding to a movement performed in the *testing* phase (Acc\_X, Acc\_Y, Acc\_Z or Gyro\_X, Gyro\_Y, Gyro\_Z),
- *feature-code* – 30-bits input for denoting the features selected out of a total of 30 features during cluster formation on the *training* phase dataset (having ‘1’ for a selected feature else ‘0’).

The sequence of features (10 features) has been illustrated in Figure 7.15, which are extracted from each tri-axial data segment (X, Y and Z) of each sensor type, thereby making up a total of 30 features (cf. section 6.2.2). The features selected (out of a total of 30) during the cluster formation are represented using this *feature-code*. An example of a 30-bit code can be of the form: 000100000000000001001000000000, which represents that the features (3, 17, 20) viz.  $D_x$  (dispersion computed on x-axis data),  $jerk_y$  (jerk metric on y-axis) and  $rms_z$  (rms on z-axis) were selected during the cluster formation.

rms	abs. diff	$\sigma$	D	kurtosis	skew	inf. entropy	jerk	peaks	max_mag
0	1	2	3	4	5	6	7	8	9

Figure 7.15: Sequence of features extracted from each tri-axial data segment to form a 30-bit *feature-code*.

- *cluster centroid* - 16-bits input each for 3 cluster centroids corresponding to the clusters formed from the features selected from the *training* phase data,
- *predicted cluster* – 2-bits output for the predicted cluster depending on the minimum distance of the *test* dataset from the cluster centroids.

The methodology, already presented in Figure 6.1, has been illustrated through a mathematical approach having three clusters in a 2-dimensional feature space and a test vector to be associated based on the computation of the minimum distance. In Figure 7.16, A, B, C represent the three clusters formed using *k*-means on the *training* dataset of the three movements (A, B and C) represented by two features – Feature 1 ( $f_1$ ) and Feature 2 ( $f_2$ ),  $T$  represents the *test* vector. The distance of the *test* vector  $T$  from each of the three cluster centroids are represented by  $d_A$ ,  $d_B$  and  $d_C$ . These three distance measures are compared to estimate the proximity of the *test* dataset  $T$  to the clusters. Let us understand the distance computation in the respective feature space through this working example. According to Figure 7.16, the two dimensional co-ordinates can be represented as:

- Cluster centroid A –  $(f_{A1}, f_{A2})$
- Cluster centroid B –  $(f_{B1}, f_{B2})$
- Cluster centroid C –  $(f_{C1}, f_{C2})$
- *Test* vector  $T$  –  $(f_{T1}, f_{T2})$

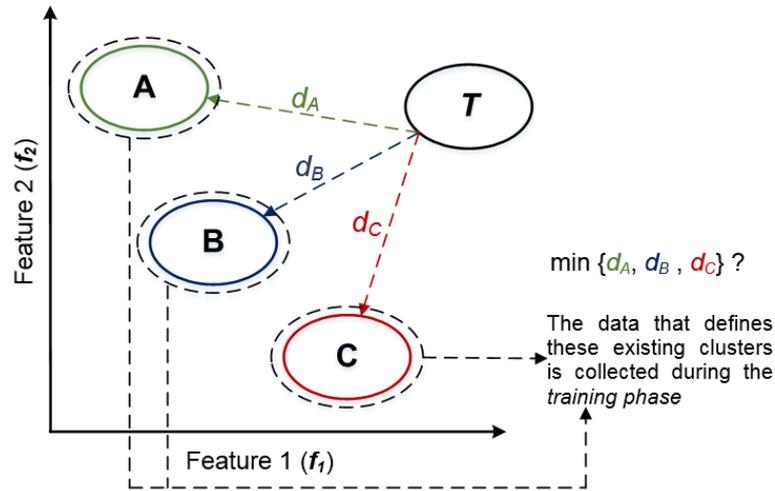


Figure 7.16: Illustration of the minimum distance classification methodology.

The Euclidean distance of the *test* feature vectors from the three cluster centroids can be computed as:

$$d_A = \sqrt{(f_{T1} \sim f_{A1})^2 + (f_{T2} \sim f_{A2})^2} \quad (7.20)$$

$$d_B = \sqrt{(f_{T1} \sim f_{B1})^2 + (f_{T2} \sim f_{B2})^2} \quad (7.21)$$

$$d_C = \sqrt{(f_{T1} \sim f_{C1})^2 + (f_{T2} \sim f_{C2})^2} \quad (7.22)$$

In this example, a two dimensional feature space  $(f1, f2)$  has been considered which can be easily extended to incorporate all the 30 features. The distance computation expressions in (7.20) – (7.22) can be generalized as:

$$d_A = \sqrt{(f_{T0} \sim f_{A0})^2 + (f_{T1} \sim f_{A1})^2 + \dots + (f_{T29} \sim f_{A29})^2} \quad (7.23)$$

$$d_B = \sqrt{(f_{T0} \sim f_{B0})^2 + (f_{T1} \sim f_{B1})^2 + \dots + (f_{T29} \sim f_{B29})^2} \quad (7.24)$$

$$d_C = \sqrt{(f_{T0} \sim f_{C0})^2 + (f_{T1} \sim f_{C1})^2 + \dots + (f_{T29} \sim f_{C29})^2} \quad (7.25)$$

The expressions (7.23) – (7.25) can be reframed as:

$$d_A = \sqrt{d_{As0}^2 + d_{As1}^2 + \cdots + d_{As29}^2} \quad (7.26)$$

$$d_B = \sqrt{d_{Bs0}^2 + d_{Bs1}^2 + \cdots + d_{Bs29}^2} \quad (7.27)$$

$$d_C = \sqrt{d_{Cs0}^2 + d_{Cs1}^2 + \cdots + d_{Cs29}^2} \quad (7.28)$$

where the data samples ( $d_{Asi}$ ,  $d_{Bsi}$ ,  $d_{Csi}$ ,  $i = 0, 1, \dots, 29$ ) are the computed differences between the feature vectors of the *test* dataset and the cluster centroids. The expressions (7.26) – (7.28) can be generalized as:

$$d_A = \sqrt{\left(\sum_{i=0}^{29} d_{Asi}^2\right)} \quad (7.29)$$

$$d_B = \sqrt{\left(\sum_{i=0}^{29} d_{Bsi}^2\right)} \quad (7.30)$$

$$d_C = \sqrt{\left(\sum_{i=0}^{29} d_{Csi}^2\right)} \quad (7.31)$$

The expressions (7.29) – (7.31) have a functional similarity to the *rms* computation mentioned in (7.3) and hence can be realized using the CORDIC operator  $Vec_c$ , represented as:

$$d_A = \left( \prod_{i=0}^{n-1} Vec_c [d_{Ai} \quad d_{Asi}]^T \right)_x \quad (7.32)$$

Similar to the *rms* computation, the samples  $d_{Asi}$  are fed in the *y* input of the CORDIC while the *x*-component of the output,  $d_{Ai}$  is fed back to the *x*-component of its input. Therefore, at every clock cycle as the new data sample  $d_{Asi}$  arrives, the computed *x*-component of the CORDIC is given by:

$$d_{Ai} = K \sqrt{d_{As0}^2 + d_{s1}^2 + \dots + d_{As(i-1)}^2} \quad (7.33)$$

After every complete CORDIC operation the *x*-component of the output is scaled with the scale factor  $K$ . With this scale factor compensation step in place, the final result at the *x* output of the CORDIC is obtained after  $n$  number of operations, where  $n$  is dependant on

the number of features selected ( $1 > n \leq 30$ ). Similarly, the distances  $d_B$ ,  $d_C$  can be computed using the operator  $Vec_c$ .

The architecture for the minimum distance computation for associating the *test* dataset (corresponding to the movement performed in an uncontrolled environment) to pre-computed cluster centroids is shown in Figure 7.17 and Figure 7.18.

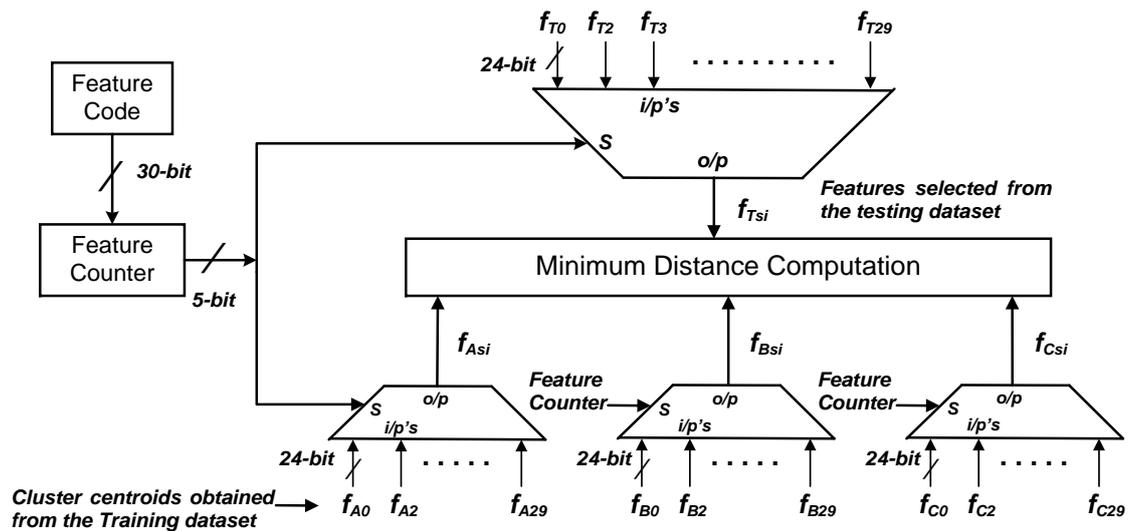


Figure 7.17: Overview of the minimum distance classifier architecture.

The features mentioned in the *feature-code* are selected. The cluster centroid for that corresponding feature is selected through a sample counter (*feature-counter*) which counts through the 30-bit *feature-code*. The features selected from the testing set ( $f_{Tsi}$ ) and the corresponding cluster centroids ( $f_{Asi}$ ,  $f_{Bsi}$ ,  $f_{Csi}$ ) are passed onto a minimum distance computation module.

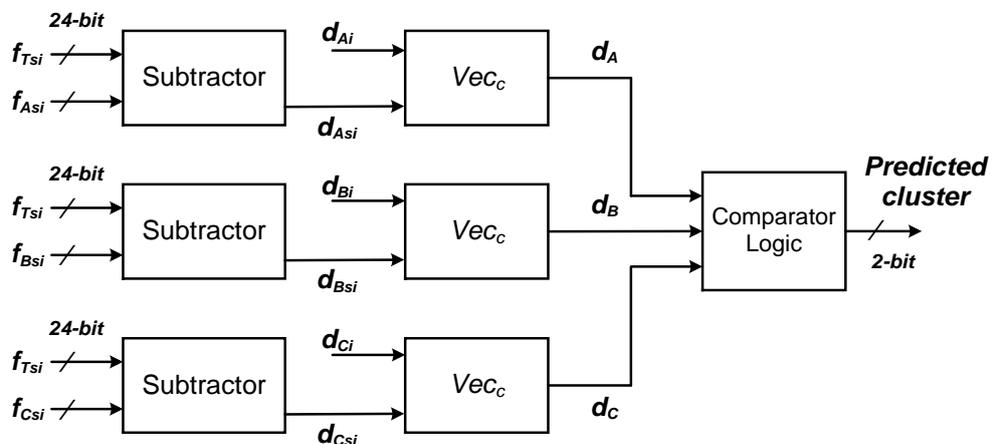


Figure 7.18: Architecture for the minimum distance computation module.

The minimum distance computation module as shown in Figure 7.18 uses a subtractor to compute the difference between the corresponding features and passes it onto the CORDIC module,  $Vec_c$ . The CORDIC operation takes place in accordance with (7.29) – (7.31) to produce the respective distances of the *test* set from each cluster centroid ( $d_A$ ,  $d_B$  and  $d_C$ ). A comparator logic is used to determine the proximal cluster, which is denoted by a 2-bit output ('00' - *A*, '01' - *B* and '10' - *C*).

It is interesting to note here that three CORDIC modules have been used in parallel to compute the distances from each of the three cluster centroids in parallel (cf. Figure 7.18). This can also be achieved by reusing one CORDIC module to compute each distance sequentially but would increase the computation time. For real-time detection of arm movements, a high speed design is more suitable. Using multiple CORDIC modules on the other hand has its effects on the chip area and power. Therefore, a trade-off between the computation time and area-power is necessary to have an optimal design. In the worst case scenario, if all 30 features are selected, then the distance computation from each of the three centroids as mentioned in (7.26) – (7.28) would involve 30 CORDIC operations. If each CORDIC operation takes time  $T$ , the total estimated time is  $30T$ . If the same process was repeated by re-using a single CORDIC, the time taken would be  $3 \times 30T$ , along with the overheads of a control logic (multiplexer, counter, buffers) to handle the data going in and out of the CORDIC module after the distance from each cluster is computed. The feature extraction engine consumes as little as 6.4 nW of power given the low frequency operations (@50 Hz) and therefore, in this design, priority has been given to saving the computation time by using three CORDIC modules in parallel.

Another important factor that requires special mention is the effect of normalization. The clusters are formed in a multi-dimensional feature space where the cluster analysis takes place on the features extracted from the *training* data. These features are linearly normalized with respect to their minimum and maximum value. This is explained in the expression (7.34), where  $x_i$  represents the feature vector,  $\min(x_i)$  and  $\max(x_i)$  represents the minimum and maximum value of the feature vector respectively and  $\tilde{x}_i$  is the normalized feature vector.

$$\tilde{x}_i = \frac{x_i - \min(x_i)}{\max(x_i) - \min(x_i)}, \quad i = 1, 2, 3, \dots, N \quad (7.34)$$

Therefore, the cluster centroids are represented by the normalised values of the selected features. However, during the *testing* phase, when a movement is performed, the relevant features (according to the *feature-code*), are extracted from the corresponding sensor data using the feature extraction engine as discussed in section 7.6.1. These features are further used for computing the minimum distance (Euclidean) from each of the cluster centroids. However, the features lie in different numeric ranges whereas the respective centroids (formed using normalized features, extracted from the *training* data) lie within a range of 0–1. Therefore, prior to computing the Euclidean distance, the cluster centroids are un-normalized and used as inputs to the RTL module. This could have been avoided by taking as inputs the minimum and maximum values for the respective features to the feature extraction module and normalizing the extracted features (from the *test* dataset) in RTL. But this would involve division in hardware which is computationally intensive (power consuming).

## 7.8 Synthesis and verification

The architecture for feature extraction and minimum distance computation was coded using Verilog as the HDL with a target ASIC implementation. The design was functionally verified using the experimental data from 2 healthy subjects and 2 stroke survivors. For each healthy subject, there were 80 test vectors (4 trials of ‘*making-a-cup-of-tea*’, having 20 movements in each trial). Similarly, for each stroke survivor there were 40 movement trials to be recognised (2 trials of ‘*making-a-cup-of-tea*’, having 20 movements in each trial). It is important to note here that since there is a fixed register bank capable of storing 256 samples, for testing with the data already collected during the experimental protocol, an interpolation/extrapolation module in Matlab was implemented to pre-process the *test* data to restrict the sample size to 256 samples. The results using the accelerometer and the gyroscope data from the wrist are shown in Table 7.6 - Table 7.9 for both healthy subjects and stroke survivors.

Two healthy subjects were chosen representing two extreme conditions during the evaluation of the methodology as mentioned in section 6.3. Subjects 1 and 2 had the highest and lowest overall accuracy respectively, with accelerometer data (cf. Table 6.3) and required the maximum and minimum number of features respectively, with the data from the gyroscope (cf. Table 6.4). Similarly, for the stroke survivors, subjects 1 and 4 were selected.

Subject	Features	Recognition accuracies (%)			Overall accuracy (%)
		A	B	C	
Subject1	11	93	90	40	78
Subject2	2	100	100	75	94

Table 7.6: Recognition sensitivities for each arm movement with accelerometer data for 2 healthy subjects.

Subject	Features	Recognition accuracies (%)			Overall accuracy (%)
		A	B	C	
Subject1	10	50	80	100	70
Subject2	27	70	80	70	73

Table 7.7: Recognition sensitivities for each arm movement with gyroscope data for 2 healthy subjects.

Subject	Features	Recognition accuracies (%)			Overall accuracy (%)
		A	B	C	
Subject1	19	70	80	100	80
Subject4	8	20	80	50	43

Table 7.8: Recognition sensitivities for each arm movement with accelerometer data for 2 stroke survivors.

Subject	Features	Recognition accuracies (%)			Overall accuracy (%)
		A	B	C	
Subject1	8	80	60	80	75
Subject4	30	50	50	0	38

Table 7.9: Recognition sensitivities for each arm movement with gyroscope data for 2 stroke survivors.

The results obtained as a result of the RTL simulation are on the lower side when compared to those obtained during the software evaluation (cf. section 6.3). However, there is an exception for healthy subject 2, where the overall accuracy achieved is higher for the RTL simulation. The decrease in individual movement sensitivities and the overall accuracy is primarily due to accumulation of truncation error which is a common phenomenon in fixed-point arithmetic. This is further evident from the fact that the minimal errors occur for healthy subject 2, requiring the computation of minimum number of features, *viz.* 2. Furthermore, in this implementation the test data was modified to 256 samples, which could also have an effect on the obtained results.

The RTL was simulated using Modelsim, a snapshot of the simulation has been presented in Figure 7.19. The variable *data\_in* takes as input the tri-axial data sequentially. The input data is stored in a register bank (*data\_store*) capable of storing 256 elements. The counters *feature\_count\_x/y/z* are used to count the number of features computed on each of the input data stream (X, Y and Z). The register bank *feature\_store* is used to store the extracted features and can save up to 30 features. A 1-bit flag *f\_finish\_x/y/z* is used to denote the end of feature extraction on each data stream. The registers *centroid\_1/2/3* are used to store the centroid variables. The 30-bit *feature\_code* is used to compute the Euclidean distance

between the selected features and the centroid variables. The predicted cluster label is denoted by the 2-bit register *group* and a 1-bit signal *batch\_finish* is set high once all processing is completed.

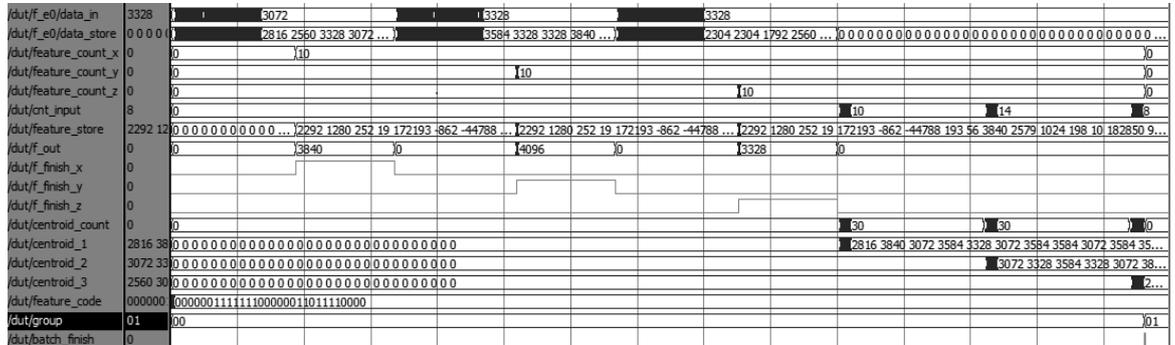


Figure 7.19: RTL simulation of the hardware module encompassing feature extraction, selection and the computation of the Euclidean distance. The predicted cluster label is A (*group* – 01).

The design was synthesized using STMicroelectronics 130-nm technology library with a supply voltage of 1.08V and clock frequency of 50 Hz. The 2-input NAND gate equivalent cell area of the synthesized design was 242 K. The dynamic power for the synthesized design was 65 nW@50 Hz. The design was also functionally verified at a higher clock frequency of 20 MhZ, where the synthesized design occupied an area of 347K (2-input NAND gate equivalent) and the dynamic power consumed was 25.9 mW. The design takes  $(9n + 30)$  clock cycles (where  $n$  is the number of input data samples) in the worst case, considering it has to compute all the 30 features from the *testing* dataset and compute the Euclidean distance to the three cluster centroids.

## 7.9 Chip design

The layout of the synthesized design was performed using the Cadence Encounter tool. The final chip, as shown in the envisaged design in Figure 7.3 was interfaced with a microcontroller, aimed at controlling the address and the data signals to and from the ASIC. The implemented design has only one 16-bit input/output port which is used to input the three sensor data streams (*AccX*, *AccY*, *AccZ* or *GyroX*, *GyroY*, *GyroZ*) sequentially. Same is followed for the three centroids which are passed as inputs sequentially. The 30-bit *feature-code* (cf. Figure 7.15 ) is split into two parts – initially, the lower 16-bits are passed as inputs and secondly the higher 14 bits are padded with two extra zeroes and passed through as inputs into the hardware module. The 2-bit output field signifying the predicted cluster label is padded with 14 zeroes. It is important to note here

that although the input data is 16-bits wide, the width of the datapath in the CORDIC-based feature extraction engine and the minimum distance computation module is 24-bits as mentioned in section 7.6.1. In order to achieve the desired 16-bit accuracy a 22-bit word-length should be selected [219], according to the formulation  $(N + \text{Log}_2N + 2)$  and atleast 16 iterations. Therefore, to obtain a high accuracy a 24-bit CORDIC was used for this implementation.

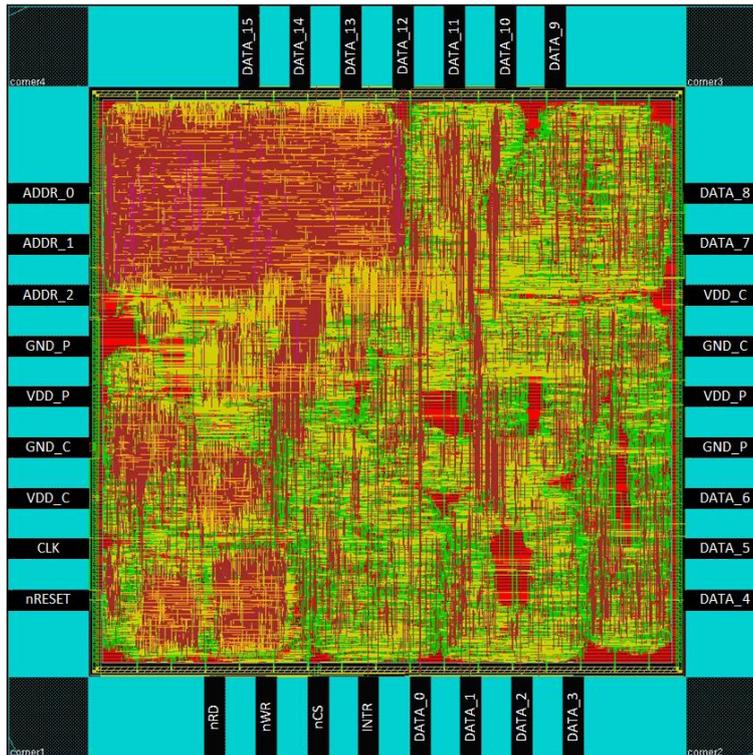


Figure 7.20: Core chip layout with all pin assignments.

The size of the final chip placed in a QFN-48 package was 2.221 mm  $\times$  2.215 mm, having 25 signal pads and 8 power/ground pads. The important pins are the 16-bit input/output data bus (DATA\_0, DATA\_1 ... DATA\_15) and the 3-bit input address bus (ADDR\_0 ... ADDR\_2), the core layout is shown in Figure 7.20. A total of 95 working chips were fabricated. The average power measurement from all the working chips is presented in Table 7.10.

Mode	VDD Core	VDD Pad
Standby	0.166 mA at 1.2V	0.001 mA at 3.3V
Operating	4.366 mA at 1.2V	0.073 mA at 3.3V

Table 7.10: Average power measurements of the 95 working chips.

In view of one of the objectives of this research work to maximize the battery life of the wearable sensors by preventing continuous transmission of the data, it is imperative to

estimate the operational life time of the envisaged sensor node (cf. Figure 7.3) having the ASIC within it enabling on-board data processing. The energy capacity of the latest prismatic zinc-air battery favoured by the medical community is 1800 mAh operating at 1.4V [33]. Considering the average power of the working chips in the operational mode for the core (cf. Table 7.10), it can be deduced that the operational life-time of the sensor node would be approximately 480 hours  $[(1800 \times 1.4)/(4.366 \times 1.2)]$ , implying that the sensor node can work for 20 days without charging. This calculation presents an approximate overview considering only the operational power consumption of the chip core (the primary research focus), other vital components of the sensor node would also have an effect on the operational life-time. It is also important to note that the power consumption in the standby mode have not been considered as well as the current consumption of the pads since the later would vary according to the technology library used. This further reflects the effectiveness of the design whereby the ASIC can be used for long-term monitoring of arm movements in real-time.

## 7.10 Discussion

In this chapter, an architectural implementation of the minimum distance classifier to recognize a movement performed in an out-of-laboratory environment has been presented. Three movements performed initially in a controlled environment or the *training* phase, are processed in software to form clusters in a multi-dimensional feature space. The hardware module is aimed at associating the kinematic data corresponding to the movements performed in the *testing* phase to the proximal cluster centroid. The processing of the data involves feature extraction, selection and computation of the Euclidean distance to the three cluster centroids. The synthesized design consumes 65 nW@50 Hz of dynamic power, occupying 242 K NAND-2 equivalent area. The synthesized RTL was simulated to verify its functionality. The fabricated core has been packaged in a QFN-48 package. However, in view of the designed architecture, there are a few fundamental factors which can be considered in future designs.

First, the size of the register bank to store the incoming data samples (*data\_in*) from the sensors has been fixed at 256 (which can be represented on a dyadic scale,  $2^8$ ) and hence is convenient for feature computation. With the sensor streaming data at 50 Hz, 256 samples represent approximately 5 seconds of kinematic data. This time duration is suitable for the healthy subjects for the completion of the elementary arm movements (actions) chosen for

the experimental protocol. For patients, depending on the level of dexterity, the time duration might be more especially when they are in their initial stage of rehabilitation. The next available window size, in view of representing it in dyadic scale is 512 implying 10 seconds and would suit the requirements of patients needing more time to complete the actions. An alternate approach would be to reduce the sampling frequency in the range of 20~25 Hz which has also been considered to be suitable in human activity recognition [27][2]. Such a case of window selection has been discussed at length in section 2.2.5.6.2.

Secondly, the difference in the results of the software implementation of the methodology (cf. section 6.3) and its RTL implementation have been briefly discussed in section 7.8. The differences in the results are primarily due to the accumulation of truncation error which is a common phenomenon in fixed-point arithmetic operations and occurs due to the implemented logic. In this application, where a data point is being classified based on some decision boundary, this error may accumulate to such an extent that the data point may come very close to the decision boundary or even overshoot it, resulting in misclassification. The software implementation in Matlab presents the obtained results in a 64-bit operating system whereas the CORDIC module implemented in RTL has a datapath width of 24-bits. Since, in this implementation, to achieve 16-bit accuracy, 16 iterations are used and hence this recursive CORDIC operation results in error accumulation to a higher degree. Therefore, the difference of accuracy is further evident especially while computing a higher number of features. Hence, a fundamental exploration in terms of error accumulation and propagation needs to be carried out and accordingly the datapath adjustment for the ASIC implementation needs to be done in view of the target accuracy.

Thirdly, as discussed in section 7.7, the cluster centroids are un-normalized prior to the computation of the Euclidean distance from the *test* features. However, for cases with more number of features it is better to normalize the features computed on the *test* dataset with the minimum and the maximum values of the cluster forming features computed on the *training* dataset. The normalization step involving division can be achieved using the CORDIC operator  $Vec_i$ .

This chapter concludes the explorations performed to achieve the research objectives. In the following chapter conclusions are drawn comparing the performance of the adopted methodologies and relevant future prospective work has been outlined.

## 8. Chapter 8

### Conclusion and Future Works

This work reports on a systematic exploration to recognise three elementary arm movements that are used in daily life. It has also been demonstrated that with particular focus on the algorithm and architecture design, a real-time movement detection system can be achieved suitable for use in a resource constrained environment of a body-worn wireless sensor node. Three approaches have been used to recognize the movements performed by four healthy subjects and four stroke survivors in an out-of-laboratory environment, during the archetypal activity of '*making-a-cup-of-tea*'.

The first approach based on sensor orientation presents a novel yet simple algorithm to recognize the three arm movements by analysing transitions between six pre-defined orientations of a wrist-worn accelerometer. Accuracy in the range of 91%-99% for healthy subjects and 70%-85% for stroke survivors was achieved using only a tri-axial accelerometer on the wrist. Therefore, this methodology reduces the overheads of complex data processing (i.e., feature extraction/selection and classification) and does not require a training system to learn response patterns which is involved in conventional methodologies of human movement recognition. Although this approach has proved to be quite successful, the pre-defined orientations and the transitions are particularly aimed at recognising the investigated movements, performed in a horizontal plane. Therefore, this algorithm is not scalable or flexible for incorporating new category of movements.

For the second approach three classifiers were used, namely, LDA, QDA and SVM for recognizing the performed arm movements. A robust training model was developed retrospectively using the data collected in the *laboratory setup* (*training* dataset) following two types of approach: subject-independent (*generalized*) and subject-dependent (*personalized*), in association with each of the learning algorithms - LDA, QDA and SVM. The model is verified through cross-validation methodologies to determine the best classification methodology, the appropriate sensor signals and its position for recognizing the movements. The LDA learning algorithm involving low-computational complexity had comparable performance to the other two investigated classifiers. Hence, LDA was used in

conjunction with the individual sensor axes signals, to prospectively evaluate the developed model on data collected while ‘*making-a-cup-of-tea*’ (*testing* phase) following a *personalized* approach. However, a satisfactory level of classification ( $> 60\%$ ) was not achieved across all the arm movements for any of the healthy subjects. For the stroke survivors, the maximum sensitivity obtained for any action was 50% and for none of the subjects, all actions were classified with a sensitivity of at least 50%. Although the sensitivities for each arm movement were quite high in the retrospective cross-validation of the *training* dataset, especially using signals from the wrist-worn accelerometer and gyroscope, but when prospectively evaluated on the *testing* dataset, the learnt model failed to classify the actions up to an acceptable level.

In view of the achieved results, the implementation of this classification methodology in hardware was not pursued. However, from this exploration three important conclusions were drawn: 1) effectiveness of the subject-dependant (*personalized*) approach; 2) efficiency of the individual sensor signals over the modulus of the accelerometer and gyroscope signals, or their combined fused signals and 3) effectiveness of the wrist over the elbow as the sensing position for both the accelerometer and the gyroscope.

The third approach considers the formation of three clusters pertaining to the three movements (reach and retrieve; lift arm, rotate arm) performed in the *training* phase in a multi-dimensional feature space and recognizes the movements performed while ‘*making-a-cup-of-tea*’ (*testing* phase), by computing its proximity to the cluster centroids using a minimum distance classifier. The three movements were detected with an overall average accuracy of 88% using the accelerometer data and 83% using the gyroscope data from the wrist, across all healthy subjects and the three arm movement types. The average accuracy across all stroke survivors was 70% using accelerometer data and 66% using the data from the gyroscope. The minimum sensitivity for detecting each individual arm movement was 80% for healthy subjects and 60% for stroke survivors if more than one sensor is used. The results particularly those obtained for the stroke survivors reveal that there is a need to consider more than one sensor type while detecting such elementary arm movements. The achieved results were further compared against the classification results of LDA and SVM, to establish the effectiveness of the proposed clustering-based methodology.

For the sake of generality, to recognise other types of arm movements using the sensor orientation based approach, new sensor positions and their transitions have to be defined.

This requires specific inputs from the therapists regarding the movements they would like to monitor. Correspondingly, the algorithm can be tuned to detect the occurrence of the specific movements. Hence, there is a need for a close coordination between the therapists and data analysts for extending it to future applications. Another interesting factor that needs to be mentioned is that for a supposedly day-long monitoring period, the orientation of the sensor module placed on the wrist needs to be constant. However, from practical experience of wearing a watch it is a well-known fact that the positioning is more likely to change over the course of a day. Therefore, a firm wrist band housing the sensor module, which is comfortable for the patients to wear, needs to be used such that the sensor stays at a fixed position without affecting its orientation.

In comparison, the approach based on clustering and minimum distance classifier appears as a more efficient methodology in view of its flexibility and scalability. It can be modulated to include any other category of movements depending on the clinical requirements. In this work three arm movements have been considered primarily as a proof-of-concept methodology. The choice of the movements were further guided by the consultation with therapists who highlighted the importance of tracking elementary arm movements that constitute majority of the activities performed with the upper limb in daily life. To incorporate more number of movements, more clusters (reflecting the number of unique movements) can be formed and the functionality can be tuned in accordance (Euclidean distance computed from the *test* dataset to the new cluster centroids) to recognize the respective movements. Therefore, this approach is a preferred option for detecting any category of arm movements as compared to the one based on sensor orientation.

The clustering based approach was developed in an offline-online resource sharing mechanism. The time and memory intensive process of feature computation, selection and cluster formation, on the *training* data were done in an offline mode (in Matlab). The computation of the selected features on the *testing* data and computation of the minimum distance (Euclidean) from the pre-computed cluster centroids was realised through a novel low-power ASIC, to be used within a wireless sensor node for real-time continuous monitoring. The fabricated ASIC has a dynamic power consumption of 25.9 mW @20 MHz and a total chip area of 2.221 mm × 2.215 mm. This chip can be embedded on a body-worn wireless sensor node along with other processing components like A/D converter, filtering circuit, memory, power source (cf. Figure 7.1), to associate the

activities performed in daily life (*testing* data) to the pre-computed cluster centroids in real-time. As discussed in the application framework in section 6.5, this methodology can be adaptable to the changing movement patterns of the patients reflective of an improvement in their motor functionality depending on the rehabilitation. The patient's *training* data can be collected periodically as and when requested by the clinician and the cluster centroids and the associated features can be recomputed to reflect the changing movement patterns in offline mode. This information can be further used by the ASIC, embedded within the wireless sensor node to recognize movements performed in daily life.

Given the overall accuracy achieved with the sensor orientation based approach, it was also implemented on a reconfigurable hardware platform, i.e. FPGA, to develop a real-time arm movement recognition system. The synthesized design used 1804 logic elements and recognises the performed arm movements in 41.2  $\mu$ s, @50 MHz clock on the FPGA. This system acts a proof-of-concept and can be implemented as a low-power ASIC chip and embedded on a sensor platform to detect and enumerate the occurrence of the three arm movements in daily life. There was only an opportunity to pursue the fabrication of one design (at IHP, cf. section **Research Constraints**1.6.4). Therefore, the minimum distance classifier based methodology was translated to an ASIC in view of its flexibility and scalability to incorporate changing movement patterns or new categories of movements.

The achieved results, for both the healthy subjects and the stroke survivors using both the approaches can be considered favorable because the methodology was tested to detect activities performed in out-of-laboratory, semi-naturalistic scenario, having a significant degree of variability. The accuracy rates reported for the stroke survivors are acceptable, according to clinicians, since it provides a gross measure of impaired arm use. It is important to mention here that a misclassification or a false detection of a performed movement may not have any significant clinical impact because in this application the final decision on the rehabilitation measure and the corresponding prescription lies with the jurisdiction of the respective clinicians. This methodology could help to augment the clinical findings and provide a quantitative measure on the rehabilitation progress of the patients over time outside the clinical environment.

A completely personalized approach has been presented and the results obtained have been encouraging and show that these particular arm movements can be reliably detected with stroke survivors exhibiting moderate levels of involuntary tremor in their movements. The

developed system could be used as a clinical tool to assess arm rehabilitation progress amongst stroke survivors by tracking the number of times the person performs specific arm movements with their paretic arm throughout the day.

## 8.1 Future works

The research done in this thesis provides a foundation for further exploration and use of advanced techniques to detect arm movements or human movements in general. This work was the first attempt to propose two novel arm movement recognition algorithms and their implementation on a hardware platform, facilitating its application for remote monitoring of patient activity in real-time. The future prospects and other interesting avenues of research are outlined as follows:

### (1) Segmentation of activities

Continuous monitoring of activities in nomadic settings involves the key steps of data segmentation and then the corresponding analysis of the data to recognise the movements of interest as highlighted in section 2.2.5.5.1. Although these two aspects are in practice interrelated but are individually two separate research problems owing to the possible qualitative non-uniqueness of an activity pattern exhibited by an individual subject and due to inter-person variability. In the research reported here, the focus has been only on the activity recognition part as a proof-of-concept methodology. A fixed size sliding window of 256 samples has been used which is in accordance to the requirements of the hardware platform implementing the movement recognition algorithm. However the pitfalls of a fixed size sliding window have already been discussed in section 2.2.5.6.2.

Therefore, as a future work, a segmentation algorithm needs to be developed which can be used as a precursor to the movement recognition algorithms proposed in this work. A dynamic window selection technique based on a data-driven or a probabilistic approach can be further explored. A threshold based segmentation mechanism whereby observing the modulus signal of the tri-axial accelerometers and gyroscopes can help to infer whether an activity has been performed and to some extent the type of an activity. This is because, when the arm is stationary the total acceleration measured by the sensor module is equal to the value of the gravitational acceleration ( $g$ ) and gyroscopes should measure zero as the three axes experience no rotation. Hence, checking a continuous stream of data against

values ( $g \pm x\%$ ) for the accelerometer and ( $0 \pm x\%$ ) for the gyroscope (where  $x$  is the selected threshold value) can help to separate periods of inactivity.

## (2) New movement recognition algorithms

In this present study, time domain features have been used with respect to their popularity in relevant literature. Therefore, new features especially those obtained from frequency domain analysis (Fourier and Wavelet decomposition) can be used in conjunction with new learning algorithms. Some of the common frequency domain features can be: power spectral density from the Fourier analysis; the energy content, entropy and the number of peaks from the signal at various frequency bands obtained from Wavelet decomposition.

Here, three supervised learning algorithms – LDA, QDA and SVM have been explored. These algorithms produced high sensitivities for each investigated arm movement during the retrospective cross-validation of the *training* dataset but the learnt model failed to classify the arm movements during the prospective evaluation on the *testing* dataset. In view of this, new recognition algorithms can be evaluated to recognize the performed arm movements in an out-of-laboratory situation. An initial analysis was also performed using  $k$ -nearest neighbor classifier ( $k$ -NN, varying  $k$  from 3 to 7) and Hidden Markov model (HMM) but it did not lead to a betterment of the individual sensitivities of the movements. Since the main objective was to implement the designed algorithm on a resource constrained hardware platform, commonly used methods like Multi-layer perceptron [2] or Artificial Neural Networks (ANNs) [28], were not explored due to the high complexity involved. These explorations have not been reported here for the sake of brevity. One prime reason, towards the low sensitivities for individual movements obtained is the number of data points. To develop a robust learning model, it is always better to train the classifier with a large pool of data so that it inherently captures the variations in the underlying data samples.

As a future work, different variants of HMMs can be explored, given its popularity in activity recognition literature and its feasibility for implementation onto a hardware platform [224]. Semi-Markovian HMMs or coupled-HMMs can be used for classifying the arm movements. Decision trees (DT) classifiers are also another popular choice [28] [157] and can be used to recognize the performed movements in an uncontrolled environment.

### (3) Quality of movement

The primary objective of stroke rehabilitation is to detect the improvement in the motor functionality of the patients over time. Estimating the quality of the movements performed over a period of time can be a key rehabilitation indicator. This thesis focusses on the recognition methodology of particular arm movements. Enumeration of such specific movements over time could be used as a rehabilitation indicator, since the frequency of these movements is more likely to increase as the motor functionality of the patient improves. Various other parameters can be analysed using data from pervasive body-worn sensors to indicate change in the quality of movements performed within the home environment.

An initial study on the measurement of movement fluidity of the upper arm for four stroke survivors for duration of 3 weeks was performed. Details of these findings have been presented in Appendix A. The four stroke survivors, who had participated in the study for an archetypal activity of '*making-a-cup-of-tea*', were requested to perform a trial of the same over 3 weeks during their regular visit to the clinic (i.e. Brandenburg Klinik, Germany). Kinematic data was collected using tri-axial accelerometers placed near the wrist and analyzed to determine the *jerk metric* [225] and the *peak number* on each orthogonal sensor axes and their moduli. The *jerk metric* characterizes the average rate of change of acceleration in a movement. It is calculated as the negative root mean square value of the derivative of the acceleration data normalized by the maximum velocity and will ideally increase with time as the patient improves. By comparison, the *peak number* is obtained from gradient analysis of the acceleration data. It represents the tremor during arm movement and will ideally decrease as the patient improves. Less number of peaks in the speed signal (integrating the acceleration signal) represents fewer periods of acceleration and deceleration, indicating a smoother movement. During the same time, the patients also undergo three trials of the Box and Block ) test [34] and Nine Hole Peg test test [46] at the beginning of each experiment session, performed over the same 3 week period under the supervision of the clinicians. These two tests are reliable measures of gross manual dexterity and arm functionality, performed by clinicians to assess rehabilitation in clinical settings. An increase in number of blocks transported per minute and number of pegs placed per second indicates improvement in arm functionality [226].

The results of the clinical tests illustrate an increase in the number of blocks transferred per minute for all four subjects over the three weeks. The results from the Nine Hole Peg test

also reflect an increase in the number of pegs placed per second across all subjects. The results for the statistical features, *jerk metric* and *peak number*, calculated on the accelerometer data indicate a decrease in the number of peaks and an increase in the value of *jerk metric* for all the four subjects over the experimental duration of three weeks. This supports the clinical observations as obtained from the B&B and the NHP test scores. These results suggest that low-cost body worn sensors can be used in a pervasive manner to determine the rehabilitation of patients within the home environment.

This work is an initial exploration and presents a preview into the wide area of clinical research on estimating the quality of movements performed by the stroke survivors with their impaired arm. This study can be further extended by analyzing other metrics like joint angles, postures in the three dimensional space as subjects perform their daily activities over a prolonged period for estimating the movement quality. A longitudinal study based on a larger subject base and over a longer time period, can facilitate the formulation of useful rehabilitation indices which can complement the clinical measures for assessing patients within the home environment.

#### **(4) Applicability in real-life**

In this thesis, two novel arm movement recognition algorithms have been proposed and they have been translated to a hardware platform. Given the efficiency of the sensor orientation algorithm, it was also implemented as a Matlab based graphical user interface (GUI). This GUI has been installed at the Brandenburg Klinik, Germany and has been used to test arm movements performed by stroke survivors. Since this algorithm does not require a dedicated training session, it can conveniently be used to track the three arm movements – reach and retrieve, lift and rotate arm.

The clustering and minimum distance classifier based approach which was developed into a low power ASIC can be embedded on a body-worn wireless sensor node (cf. Figure 7.1), to perform movement recognition in real-time and the results can be displayed on a PDA/mobile platform carried by the patient as well as stored in memory for future referral by clinicians. Since this methodology has been implemented as offline-online processing system, it can be trained to detect any category of movements in an offline mode and the ASIC can be used to detect those movements in real-time. This methodology could be extended for use with patients suffering from other neurodegenerative disorders exhibiting less fluidic movement profiles. It can also be extended towards monitoring of lower limb

movements. Real-time detection of arm movements can be very useful in a wide array of applications in the field of sports, human computer interaction or other treatments of arm dexterity. Therefore, the developed system can be used to track movements of required body segments in these respective fields outside a controlled environment.



# Appendix A

The four stroke survivors involved in the experiments at BBK, Germany were selected by the physiotherapists depending on their availability and agreement to participate in the trials. An initial analysis to ascertain their degree of impairment was performed using the streamlined WMFT set. However, consent was not available from BBK to disclose these scores due to ethical issues. On further consultation, a brief summary of their functional state has been presented in Table A.1. The functional ability in Table A.2 reports on the qualitative content of the WMFT score and the physiotherapist's standard evaluation.

Subject	Sex	Natural arm	Impaired arm	Age	Functional ability
Subject1	Female	Right	Right	45	High WMFT score, low impairment
Subject2	Male	Right	Left	65	Average WMFT score, average impairment
Subject3	Male	Right	Right	72	Low WMFT score, high impairment
Subject4	Male	Right	Left	73	Low WMFT score, high impairment, early stage of rehabilitation

Table A.1: Functional details of the four stroke participants as assessed by the therapists.

The following segment on Box and Block test, Nine Hole Peg test and movement fluidity addresses the initial exploration on determining quality of movement (cf. section 8.1).



Figure A.1: Experimental setup for the Box and Block test (left) and Nine Hole Peg test (right).

Subject	Box & Block (blocks/min)		Nine Hole Peg (pegs/sec)	
	Week1	Week3	Week1	Week3
Subject1	29	49	0.25	0.31
Subject2	17	19	0.065	0.067
Subject3	12	34	0.10	0.12
Subject4	47	59	0.35	0.41

Table A.2: Clinical results for assessing the motor functionality of the impaired arm over 3 weeks (only presented for the first and third week) [226]

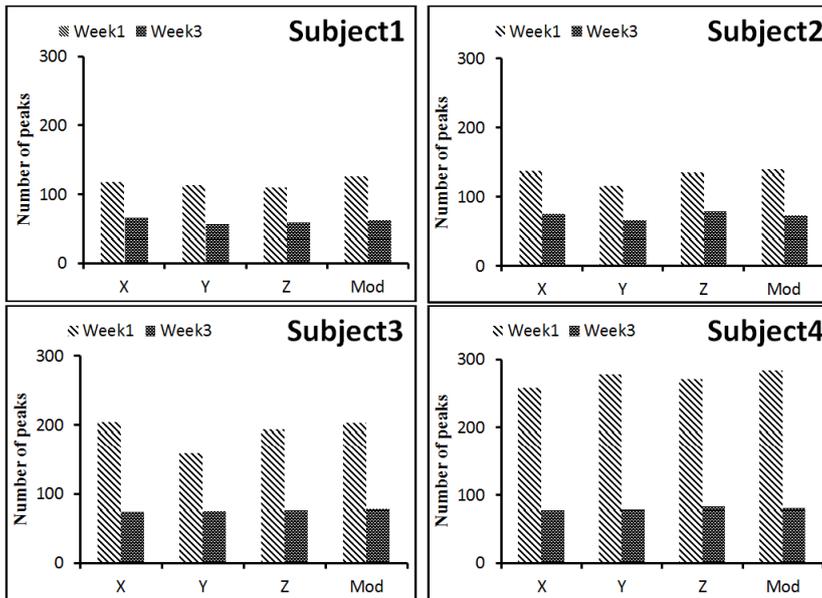


Figure A.2: The decrease in number of peaks for 4 subjects from week 1 to week 3 on X, Y, Z, modulus (horizontal axis) of the accelerometer data [226].

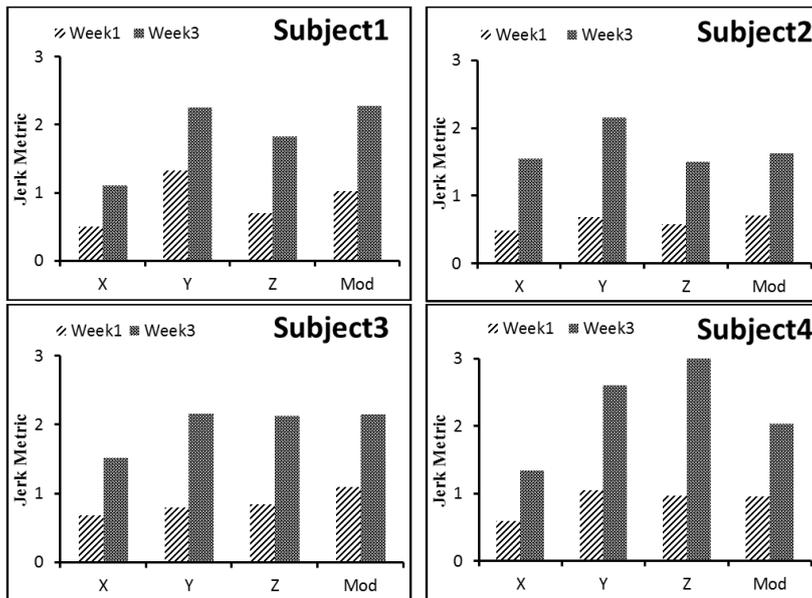


Figure A.3: The increase in jerk metric value for 4 subjects from week 1 to week 3 on X, Y, Z, modulus (horizontal axis) of accelerometer data [226].

# References

- [1] J. A. Botia, A. Villa, and J. Palma, "Ambient assisted living system for in-home monitoring of healthy independent elders," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8136–8148, 2012.
- [2] S. Chernbumroong, S. Cang, A. Atkins, and H. Yu, "Elderly activities recognition and classification for applications in assisted living," *Expert Systems with Applications*, vol. 40, no. 5, pp. 1662–1674, 2013.
- [3] S. Meairs, N. Wahlgren, U. Dirnagl, O. Lindvall, P. Rothwell, J.-C. Baron, K. Hossmann, B. Engelhardt, J. Ferro, J. McCulloch, and others, "Stroke research priorities for the next decade—a representative view of the European scientific community," *Cerebrovascular Diseases*, vol. 22, no. 2–3, pp. 75–82, 2006.
- [4] C. J. Murray and A. D. Lopez, "Alternative projections of mortality and disability by cause 1990–2020: Global Burden of Disease Study," *The Lancet*, vol. 349, no. 9064, pp. 1498–1504, 1997.
- [5] S. Patel, R. Hughes, T. Hester, J. Stein, M. Akay, J. G. Dy, and P. Bonato, "A novel approach to monitor rehabilitation outcomes in stroke survivors using wearable technology," *Proceedings of the IEEE*, vol. 98, no. 3, pp. 450–461, 2010.
- [6] S. Wilson, R. Davies, T. Stone, J. Hammerton, P. Ware, S. Mawson, N. Harris, C. Eccleston, H. Zheng, N. Black, and others, "Developing a telemonitoring system for stroke rehabilitation," *Contemporary Ergonomics*, pp. 505–512, 2007.
- [7] "StrokeBack." [Online]. Available: [www.strokeback.eu](http://www.strokeback.eu). [Accessed: 01–10-2014].
- [8] "Effects of Stroke." [Online]. Available: [http://www.strokeassociation.org/STROKEORG/AboutStroke/EffectsofStroke/Effects-of-Stroke\\_UCM\\_308534\\_SubHomePage.jsp](http://www.strokeassociation.org/STROKEORG/AboutStroke/EffectsofStroke/Effects-of-Stroke_UCM_308534_SubHomePage.jsp). [Accessed: 01–01-2012].
- [9] "Brain - Effects of a Stroke." [Online]. Available: <http://www.ama-assn.org/ama/pub/physician-resources/patient-education-materials/atlas-of-human-body/brain-effects-stroke.page>. [Accessed: 01–01-2012].
- [10] K. Strong, C. Mathers, and R. Bonita, "Preventing stroke: saving lives around the world," *The Lancet Neurology*, vol. 6, no. 2, pp. 182–187, 2007.
- [11] M. Zampolini, E. Todeschini, G. M. Bernabeu, H. Hermens, S. Ilsbroukx, V. Macellari, R. Magni, M. Rogante, M. S. Scattareggia, M. Vollenbroek, and others, "Tele-rehabilitation: present and future," *Annali dell'Istituto superiore di sanità*, vol. 44, no. 2, pp. 125–134, 2007.
- [12] F. Le, I. Markovsky, C. T. Freeman, and E. Rogers, "Identification of electrically stimulated muscle models of stroke patients," *Control Engineering Practice*, vol. 18, no. 4, pp. 396–407, 2010.
- [13] C. T. Freeman, A.-M. Hughes, J. H. Burridge, P. H. Chappell, P. L. Lewin, and E. Rogers, "A model of the upper extremity using FES for stroke rehabilitation," *Journal of biomechanical engineering*, vol. 131, no. 3, p. 031011, 2009.
- [14] C. T. Freeman, E. Rogers, A. Hughes, J. H. Burridge, and K. L. Meadmore, "Iterative learning control in health care: Electrical stimulation and robotic-assisted upper-limb stroke rehabilitation," *Control Systems, IEEE*, vol. 32, no. 1, pp. 18–43, 2012.
- [15] S. V. Adamovich, G. G. Fluet, A. Mathai, Q. Qiu, J. Lewis, and A. S. Merians, "Journal of NeuroEngineering and Rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 6, p. 28, 2009.
- [16] S. L. Wolf, C. J. Winstein, J. P. Miller, P. A. Thompson, E. Taub, G. Uswatte, D. Morris, S. Blanton, D. Nichols-Larsen, and P. C. Clark, "Retention of upper limb function in stroke survivors who have received constraint-induced movement therapy: the EXCITE randomised trial," *The Lancet*

- Neurology*, vol. 7, no. 1, pp. 33–40, 2008.
- [17] E. Taub and D. M. Morris, “Constraint-induced movement therapy to enhance recovery after stroke,” *Current atherosclerosis reports*, vol. 3, no. 4, pp. 279–286, 2001.
- [18] J. Birns, A. Bhalla, and A. Rudd, “Telestroke: a concept in practice,” *Age and ageing*, vol. 39, no. 6, pp. 666–667, 2010.
- [19] S. R. Levine and M. Gorman, “‘Telestroke’ The Application of Telemedicine for Stroke,” *Stroke*, vol. 30, no. 2, pp. 464–469, 1999.
- [20] F. Cordella, F. Di Corato, L. Zollo, B. Siciliano, and P. van der Smagt, “Patient performance evaluation using Kinect and Monte Carlo-based finger tracking,” in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS & EMBS International Conference on*, 2012, pp. 1967–1972.
- [21] F. E. Martínez-Pérez, J. Á. González-Fraga, J. C. Cuevas-Tello, and M. D. Rodríguez, “Activity inference for ambient intelligence through handling artifacts in a healthcare environment,” *Sensors*, vol. 12, no. 1, pp. 1072–1099, 2012.
- [22] B. Najafi, K. Aminian, A. Paraschiv-Ionescu, F. Loew, C. J. Bula, and P. Robert, “Ambulatory system for human motion analysis using a kinematic sensor: monitoring of daily physical activity in the elderly,” *Biomedical Engineering, IEEE Transactions on*, vol. 50, no. 6, pp. 711–723, 2003.
- [23] A. Hadjidj, A. Bouabdallah, and Y. Challal, “Rehabilitation supervision using wireless sensor networks,” in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2011 IEEE International Symposium on a*, 2011, pp. 1–3.
- [24] M. S. Raisinghani, A. Benoit, J. Ding, M. Gomez, K. Gupta, V. Gusila, D. Power, and O. Schmedding, “Ambient intelligence: Changing forms of human-computer interaction and their social implications,” *Journal of digital information*, vol. 5, no. 4, 2006.
- [25] D. Merrill, J. Kalanithi, and P. Maes, “Siftables: towards sensor network user interfaces,” in *Proceedings of the 1st international conference on Tangible and embedded interaction*, 2007, pp. 75–78.
- [26] S. Armstrong, “Wireless connectivity for health and sports monitoring: a review,” *British journal of sports medicine*, vol. 41, no. 5, pp. 285–289, 2007.
- [27] M. Ermes, J. Parkka, J. Mantyjarvi, and I. Korhonen, “Detection of daily activities and sports with wearable sensors in controlled and uncontrolled conditions,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 12, no. 1, pp. 20–26, 2008.
- [28] J. Parkka, M. Ermes, P. Korpipaa, J. Mantyjarvi, J. Peltola, and I. Korhonen, “Activity classification using realistic data from wearable sensors,” *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 1, pp. 119–128, 2006.
- [29] K. Altun, B. Barshan, and O. Tunçel, “Comparative study on classifying human activities with miniature inertial and magnetic sensors,” *Pattern Recognition*, vol. 43, no. 10, pp. 3605–3620, 2010.
- [30] S. Patel, H. Park, P. Bonato, L. Chan, and M. Rodgers, “A review of wearable sensors and systems with application in rehabilitation,” *Journal of neuroengineering and rehabilitation*, vol. 9, no. 1, p. 21, 2012.
- [31] D. T.-P. Fong and Y.-Y. Chan, “The use of wearable inertial motion sensors in human lower limb biomechanics studies: A systematic review,” *Sensors*, vol. 10, no. 12, pp. 11556–11565, 2010.
- [32] D. C. Kaelber, A. K. Jha, D. Johnston, B. Middleton, and D. W. Bates, “A research agenda for personal health records (PHRs),” *Journal of the American Medical Informatics Association*, vol. 15, no. 6, pp. 729–736, 2008.
- [33] K. Maharatna, E. B. Mazomenos, J. Morgan, and S. Bonfiglio, “Towards the development of next-generation remote healthcare system: Some practical considerations,” in *Circuits and Systems*

- (ISCAS), 2012 IEEE International Symposium on, 2012, pp. 1–4.
- [34] T. Platz, C. Pinkowski, F. van Wijck, I.-H. Kim, P. di Bella, and G. Johnson, “Reliability and validity of arm function assessment with standardized guidelines for the Fugl-Meyer Test, Action Research Arm Test and Box and Block Test: a multicentre study,” *Clinical Rehabilitation*, vol. 19, no. 4, pp. 404–411, 2005.
- [35] I. S. W. Party, “National clinical guideline for stroke,” no. 4th edition, 2012.
- [36] L. Shires, S. Battersby, J. Lewis, D. Brown, N. Sherkat, and P. Standen, “Enhancing the tracking capabilities of the Microsoft Kinect for stroke rehabilitation,” in *Serious Games and Applications for Health (SeGAH), 2013 IEEE 2nd International Conference on*, 2013, pp. 1–8.
- [37] G. Welch and E. Foxlin, “Motion Tracking: No Silver Bullet, but a Respectable Arsenal,” *Computer Graphics and Applications, IEEE on*, vol. 22, no. 6, pp. 24–38, 2002.
- [38] D. Fuentes, L. Gonzalez-Abril, C. Angulo, and J. A. Ortega, “Online motion recognition using an accelerometer in a mobile device,” *Expert Systems with Applications*, vol. 39, no. 3, pp. 2461–2465, 2012.
- [39] L. Bao and S. S. Intille, “Activity recognition from user-annotated acceleration data,” in *Pervasive computing*, Springer, 2004, pp. 1–17.
- [40] A. Burns, B. R. Greene, M. J. McGrath, T. J. O’Shea, B. Kuris, S. M. Ayer, F. Strojescu, and V. Cionca, “SHIMMER™-A wireless sensor platform for noninvasive biomedical research,” *Sensors Journal, IEEE*, vol. 10, no. 9, pp. 1527–1534, 2010.
- [41] P. K. Meher, J. Valls, T.-B. Juang, K. Sridharan, and K. Maharatna, “50 years of CORDIC: Algorithms, architectures, and applications,” *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 56, no. 9, pp. 1893–1907, 2009.
- [42] K. D. Nguyen, I.-M. Chen, Z. Luo, S. H. Yeo, and H.-L. Duh, “A wearable sensing system for tracking and monitoring of functional arm movement,” *Mechatronics, IEEE/ASME Transactions on*, vol. 16, no. 2, pp. 213–220, 2011.
- [43] N. Salbach, D. Brooks, J. Romano, and L. Woon, “The Relationship between Clinical Measures and Daily Physical Activity and Participation in Ambulatory, Community-Dwelling People with Stroke,” *J Nov Physiother*, vol. 3, no. 182, p. 2, 2013.
- [44] S. J. Strath, L. A. Kaminsky, B. E. Ainsworth, U. Ekelund, P. S. Freedson, R. A. Gary, C. R. Richardson, D. T. Smith, A. M. Swartz, and others, “Guide to the assessment of physical activity: Clinical and research applications A scientific statement from the American heart association,” *Circulation*, vol. 128, no. 20, pp. 2259–2279, 2013.
- [45] C. Do Lee, A. R. Folsom, and S. N. Blair, “Physical activity and stroke risk a meta-analysis,” *Stroke*, vol. 34, no. 10, pp. 2475–2481, 2003.
- [46] A. Heller, D. Wade, V. A. Wood, A. Sunderland, R. L. Hewer, and E. Ward, “Arm function after stroke: measurement and recovery over the first three months,” *Journal of Neurology, Neurosurgery & Psychiatry*, vol. 50, no. 6, pp. 714–719, 1987.
- [47] B. Resnick, K. Michael, M. Shaughnessy, E. S. Nahm, S. Kopunek, J. Sorkin, D. Orwig, A. Goldberg, and R. F. Macko, “Inflated perceptions of physical activity after stroke: pairing self-report with physiologic measures,” *Journal of physical activity & health*, vol. 5, no. 2, p. 308, 2008.
- [48] D. Rand, J. J. Eng, P.-F. Tang, J.-S. Jeng, and C. Hung, “How active are people with stroke? Use of accelerometers to assess physical activity,” *Stroke*, vol. 40, no. 1, pp. 163–168, 2009.
- [49] F. Naya, R. Ohmura, F. Takayanagi, H. Noma, and K. Kogure, “Workers’ Routine Activity Recognition using Body Movements and Location Information,” in *Wearable Computers, 2006 10th IEEE International Symposium on*, 2006, pp. 105–108.

- [50] R. A. Benschoter, M. T. Eaton, and P. Smith, "Use of videotape to provide individual instruction in techniques of psychotherapy," *Academic Medicine*, vol. 40, no. 12, pp. 1159–61, 1965.
- [51] W. Telemedicine, "Telemedicine Opportunities and developments in Member States: report on the second global survey on eHealth 2009," 2010.
- [52] H. G. Kang, D. F. Mahoney, H. Hoenig, V. A. Hirth, P. Bonato, I. Hajjar, and L. A. Lipsitz, "In situ monitoring of health in older adults: technologies and issues," *Journal of the American Geriatrics Society*, vol. 58, no. 8, pp. 1579–1586, 2010.
- [53] L. Fernández, J. M. Blasco, and J. F. Hernández, "Wireless sensor networks in ambient intelligence," *Technologies for Health and Well-being institute ITACA*, 2009.
- [54] O. Brand, "Microsensor integration into systems-on-chip," *Proceedings of the IEEE*, vol. 94, no. 6, pp. 1160–1176, 2006.
- [55] C. Lombriser, N. B. Bharatula, D. Roggen, and G. Tröster, "On-body activity recognition in a dynamic sensor network," in *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, p. 17.
- [56] L. Caldani, M. Pacelli, D. Farina, and R. Paradiso, "E-textile platforms for rehabilitation," in *Engineering in Medicine and Biology Society (EMBC), 2010 Annual International Conference of the IEEE*, 2010, pp. 5181–5184.
- [57] A. Pantelopoulos and N. G. Bourbakis, "A survey on wearable sensor-based systems for health monitoring and prognosis," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 40, no. 1, pp. 1–12, 2010.
- [58] J.-R. C. Chien and C.-C. Tai, "A new wireless-type physiological signal measuring system using a PDA and the bluetooth technology," *Biomedical Engineering: Applications, Basis and Communications*, vol. 17, no. 05, pp. 229–235, 2005.
- [59] B.-S. Lin, B.-S. Lin, N.-K. Chou, F.-C. Chong, and S.-J. Chen, "RTWPMS: a real-time wireless physiological monitoring system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 10, no. 4, pp. 647–656, 2006.
- [60] U. Anliker, J. A. Ward, P. Lukowicz, G. Troster, F. Dolveck, M. Baer, F. Keita, E. B. Schenker, F. Catarisi, L. Coluccini, and others, "AMON: a wearable multiparameter medical monitoring and alert system," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 8, no. 4, pp. 415–427, 2004.
- [61] M. Sung, C. Marci, and A. Pentland, "Wearable feedback system for rehabilitation," *Journal of neuroengineering and rehabilitation*, vol. 2, no. 17, pp. 0003–2, 2005.
- [62] "Ambulatory ECG." [Online]. Available: <http://www.cardionet.com/>. [Accessed: 01–02-2012].
- [63] "WatchDog." [Online]. Available: <http://www.foster-miller.com/>. [Accessed: 01–12-2011].
- [64] K. J. Heilman and S. W. Porges, "Accuracy of the LifeShirt<sup>®</sup> (Vivometrics) in the detection of cardiac rhythms," *Biological psychology*, vol. 75, no. 3, pp. 300–305, 2007.
- [65] "CleveMed." [Online]. Available: <http://www.clevedmed.com/>. [Accessed: 01–12-2011].
- [66] "Micropaq Monitor." [Online]. Available: <http://www.welchallyn.com/>. [Accessed: 01–12-2011].
- [67] C. B. Liden, M. Wolowicz, J. Stivoric, A. Teller, C. Kasabach, S. Vishnubhatla, R. Pelletier, J. Farrington, and S. Boehmke, "Characterization and implications of the sensors incorporated into the SenseWear armband for energy expenditure and activity detection," *Bodymedia Inc. White Papers: 1*, vol. 7, 2002.
- [68] "Wristcare." [Online]. Available: <http://www.istsec.fi/vivago-pam/>. [Accessed: 01–12-2011].

- [69] R. Fensli, E. Gunnarson, and T. Gundersen, "A wearable ECG-recording system for continuous arrhythmia monitoring in a wireless tele-home-care situation," in *Computer-Based Medical Systems, 2005. Proceedings. 18th IEEE Symposium on*, 2005, pp. 407–412.
- [70] R.-G. Lee, Y.-C. Chen, C.-C. Hsiao, and C.-L. Tseng, "A mobile care system with alert mechanism," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 11, no. 5, pp. 507–517, 2007.
- [71] P. Leijdekkers and V. Gay, "A self-test to detect a heart attack using a mobile phone and wearable sensors," in *Computer-Based Medical Systems, 2008. CBMS'08. 21st IEEE International Symposium on*, 2008, pp. 93–98.
- [72] Z. Jin, J. Oresko, S. Huang, and A. C. Cheng, "HeartToGo: a personalized medicine technology for cardiovascular disease prevention and detection," in *Life Science Systems and Applications Workshop, 2009. LiSSA 2009. IEEE/NIH*, 2009, pp. 80–83.
- [73] N. Oliver and F. Flores-Mangas, "HealthGear: a real-time wearable system for monitoring and analyzing physiological signals," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, p. 4–pp.
- [74] F. Bianchi, S. J. Redmond, M. R. Narayanan, S. Cerutti, and N. H. Lovell, "Barometric pressure and triaxial accelerometry-based falls event detection," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 18, no. 6, pp. 619–627, 2010.
- [75] A. K. Bourke, P. W. Van de Ven, A. E. Chaya, G. M. ÓLaighin, and J. Nelson, "Testing of a long-term fall detection system incorporated into a custom vest for the elderly," in *Engineering in Medicine and Biology Society, 2008. EMBS 2008. 30th Annual International Conference of the IEEE*, 2008, pp. 2844–2847.
- [76] M. Bachlin, M. Plotnik, D. Roggen, I. Maidan, J. M. Hausdorff, N. Giladi, and G. Troster, "Wearable assistant for Parkinson's disease patients with the freezing of gait symptom," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 436–446, 2010.
- [77] "H-CAD Portable Unit." [Online]. Available: [http://www.signomotus.it/h\\_cad\\_site/h\\_cad\\_accessible/portable\\_unit.htm](http://www.signomotus.it/h_cad_site/h_cad_accessible/portable_unit.htm). [Accessed: 10-01-2014].
- [78] H. Hermens, B. Huijgen, C. Giacomozzi, S. Ilsbrouckx, V. Macellari, E. Prats, M. Rogante, M. F. Schifini, M. C. Spitali, S. Tases, and others, "Clinical assessment of the HELLODOC tele-rehabilitation service.," *Annali dell'Istituto superiore di sanità*, vol. 44, no. 2, pp. 154–163, 2007.
- [79] A. Timmermans, P. Saini, R. Willmann, G. Lanfermann, J. te Vrugt, and S. Winter, "Home stroke rehabilitation for the upper limbs," in *Engineering in Medicine and Biology Society, 2007. EMBS 2007. 29th Annual International Conference of the IEEE*, 2007, pp. 4015–4018.
- [80] C. Mavroidis, J. Nikitczuk, B. Weinberg, G. Danaher, K. Jensen, P. Pelletier, J. Prugnarola, R. Stuart, R. Arango, M. Leahey, and others, "Journal of NeuroEngineering and Rehabilitation," *Journal of NeuroEngineering and Rehabilitation*, vol. 2, no. 18, pp. 0003–2, 2005.
- [81] T. G. Sugar, J. He, E. J. Koeneman, J. B. Koeneman, R. Herman, H. Huang, R. S. Schultz, D. Herring, J. Wanberg, S. Balasubramanian, and others, "Design and control of RUPERT: a device for robotic upper extremity repetitive therapy," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 15, no. 3, pp. 336–346, 2007.
- [82] M. K. Holden, T. A. Dyar, and L. Dayan-Cimadoro, "Telerehabilitation using a virtual environment improves upper extremity function in patients with stroke," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 15, no. 1, pp. 36–42, 2007.
- [83] D. Jack, R. Boian, A. S. Merians, M. Tremaine, G. C. Burdea, S. V. Adamovich, M. Recce, and H. Poizner, "Virtual reality-enhanced stroke rehabilitation," *Neural Systems and Rehabilitation Engineering, IEEE Transactions on*, vol. 9, no. 3, pp. 308–318, 2001.
- [84] D. Rand, R. Kizony, and P. T. L. Weiss, "The Sony PlayStation II EyeToy: low-cost virtual reality for use in rehabilitation," *Journal of neurologic physical therapy*, vol. 32, no. 4, pp. 155–163, 2008.

- [85] G. Saposnik, R. Teasell, M. Mamdani, J. Hall, W. McIlroy, D. Cheung, K. E. Thorpe, L. G. Cohen, M. Bayley, and others, "Effectiveness of virtual reality using Wii gaming technology in stroke rehabilitation a pilot randomized clinical trial and proof of principle," *Stroke*, vol. 41, no. 7, pp. 1477–1484, 2010.
- [86] K. A. Schutzer and B. S. Graves, "Barriers and motivations to exercise in older adults," *Preventive medicine*, vol. 39, no. 5, pp. 1056–1061, 2004.
- [87] M. Huber, B. Rabin, C. Docan, G. Burdea, M. E. Nwosu, M. Abdelbaky, and M. R. Golomb, "PlayStation 3-based tele-rehabilitation for children with hemiplegia," in *Virtual Rehabilitation, 2008*, 2008, pp. 105–112.
- [88] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3D tracking of hand articulations using Kinect.," in *BMVC*, 2011, vol. 1, no. 2, p. 3.
- [89] C.-Y. Chang, B. Lange, M. Zhang, S. Koenig, P. Requejo, N. Somboon, A. A. Sawchuk, and A. A. Rizzo, "Towards pervasive physical rehabilitation using Microsoft Kinect," in *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, 2012, pp. 159–162.
- [90] H. M. Hondori, M. Khademi, and C. V. Lopes, "Monitoring intake gestures using sensor fusion (microsoft kinect and inertial sensors) for smart home tele-rehab setting," in *2012 1st Annual IEEE Healthcare Innovation Conference*, 2012.
- [91] H. Feys, W. De Weerd, G. Verbeke, G. C. Steck, C. Capiou, C. Kiekens, E. Dejaeger, G. Van Hoydonck, G. Vermeersch, and P. Cras, "Early and repetitive stimulation of the arm can substantially improve the long-term outcome after stroke: a 5-year follow-up study of a randomized trial," *Stroke*, vol. 35, no. 4, pp. 924–929, 2004.
- [92] L. Legg, "Rehabilitation therapy services for stroke patients living at home: systematic review of randomised trials.," *The Lancet*, vol. 363, no. 9406, pp. 352–356, 2004.
- [93] B. Langhammer, J. K. Stanghelle, and B. Lindmark, "Exercise and health-related quality of life during the first year following acute stroke. A randomized controlled trial," *Brain injury*, vol. 22, no. 2, pp. 135–145, 2008.
- [94] M. C. Cramp, R. J. Greenwood, M. Gill, A. Lehmann, J. C. Rothwell, and O. M. Scott, "Effectiveness of a community-based low intensity exercise programme for ambulatory stroke survivors," *Disability & Rehabilitation*, vol. 32, no. 3, pp. 239–247, 2010.
- [95] N. Gebruers, C. Vanroy, S. Truijen, S. Engelborghs, and P. P. De Deyn, "Monitoring of physical activity after stroke: a systematic review of accelerometry-based measures," *Archives of physical medicine and rehabilitation*, vol. 91, no. 2, pp. 288–297, 2010.
- [96] S. Katz, T. D. Downs, H. R. Cash, and R. C. Grotz, "Progress in development of the index of ADL," *The gerontologist*, vol. 10, no. 1 Part 1, pp. 20–30, 1970.
- [97] G. Pirkl, K. Stockinger, K. Kunze, and P. Lukowicz, "Adapting magnetic resonant coupling based relative positioning technology for wearable activity recognition," in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, 2008, pp. 47–54.
- [98] C. Mattmann, O. Amft, H. Harms, G. Troster, and F. Clemens, "Recognizing upper body postures using textile strain sensors," in *Wearable Computers, 2007 11th IEEE International Symposium on*, 2007, pp. 29–36.
- [99] J. Lester, T. Choudhury, and G. Borriello, "A practical approach to recognizing physical activities," in *Pervasive Computing*, Springer, 2006, pp. 1–16.
- [100] F. I. Mahoney, "Functional evaluation: the Barthel index," *Maryland state medical journal*, vol. 14, pp. 61–65, 1965.
- [101] S.-W. Lee and K. Mase, "Activity and location recognition using wearable sensors," *IEEE pervasive computing*, vol. 1, no. 3, pp. 24–32, 2002.

- [102] K. Van Laerhoven and H.-W. Gellersen, “Spine versus porcupine: A study in distributed wearable activity recognition,” in *Wearable Computers, 2004. ISWC 2004. Eighth International Symposium on*, 2004, vol. 1, pp. 142–149.
- [103] F. Foerster and J. Fahrenberg, “Motion pattern and posture: correctly assessed by calibrated accelerometers,” *Behavior research methods, instruments, & computers*, vol. 32, no. 3, pp. 450–457, 2000.
- [104] K. Van Laerhoven and O. Cakmakci, “What shall we teach our pants?,” in *Wearable Computers, The Fourth International Symposium on*, 2000, pp. 77–83.
- [105] K. Van Laerhoven, K. A. Aidoo, and S. Lowette, “Real-time analysis of data from many sensors with neural networks,” in *Wearable Computers, 2001. Proceedings. Fifth International Symposium on*, 2001, pp. 115–122.
- [106] M. Weiser, “The computer for the 21st century,” *Scientific american*, vol. 265, no. 3, pp. 94–104, 1991.
- [107] T. Choudhury, S. Consolvo, B. Harrison, J. Hightower, A. LaMarca, L. LeGrand, A. Rahimi, A. Rea, G. Bordello, B. Hemingway, and others, “The mobile sensing platform: An embedded activity recognition system,” *Pervasive Computing, IEEE*, vol. 7, no. 2, pp. 32–41, 2008.
- [108] M. Stikic and B. Schiele, “Activity recognition from sparsely labeled data using multi-instance learning,” in *Location and Context Awareness*, Springer, 2009, pp. 156–173.
- [109] C. R. Wren and E. M. Tapia, “Toward scalable activity recognition for sensor networks,” in *Location-and context-awareness*, Springer, 2006, pp. 168–185.
- [110] G. Singla, D. J. Cook, and M. Schmitter-Edgecombe, “Recognizing independent and joint activities among multiple residents in smart environments,” *Journal of ambient intelligence and humanized computing*, vol. 1, no. 1, pp. 57–63, 2010.
- [111] T. Kleinberger, M. Becker, E. Ras, A. Holzinger, and P. Müller, “Ambient intelligence in assisted living: enable elderly people to handle future interfaces,” in *Universal access in human-computer interaction. Ambient interaction*, Springer, 2007, pp. 103–112.
- [112] L. Chen and C. Nugent, “Ontology-based activity recognition in intelligent pervasive environments,” *International Journal of Web Information Systems*, vol. 5, no. 4, pp. 410–430, 2009.
- [113] T. van Kasteren and B. Krose, “Bayesian activity recognition in residence for elders,” pp. 209–212, 2007.
- [114] M. Philipose, K. P. Fishkin, M. Perkowitz, D. J. Patterson, D. Fox, H. Kautz, and D. Hahnel, “Inferring activities from interactions with objects,” *Pervasive Computing, IEEE*, vol. 3, no. 4, pp. 50–57, 2004.
- [115] L. Chen, J. Hoey, C. D. Nugent, D. J. Cook, and Z. Yu, “Sensor-based activity recognition,” *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 42, no. 6, pp. 790–808, 2012.
- [116] N. C. Krishnan and D. J. Cook, “Activity recognition on streaming sensor data,” *Pervasive and mobile computing*, vol. 10, pp. 138–154, 2012.
- [117] C. Zhu and W. Sheng, “Motion-and location-based online human daily activity recognition,” *Pervasive and Mobile Computing*, vol. 7, no. 2, pp. 256–269, 2011.
- [118] N. Vodjdani, “The ambient assisted living joint programme,” *Electronics System-Integration Technology Conference*, pp. 1–2, 2008.
- [119] M. Chan, D. Estève, C. Escriba, and E. Campo, “A review of smart homes—Present state and future challenges,” *Computer methods and programs in biomedicine*, vol. 91, no. 1, pp. 55–81, 2008.

- [120] S. Helal, W. Mann, H. El-Zabadani, J. King, Y. Kaddoura, and E. Jansen, "The gator tech smart house: A programmable pervasive space," *Computer*, vol. 38, no. 3, pp. 50–60, 2005.
- [121] M. Stikic, T. Huynh, K. Van Laerhoven, and B. Schiele, "ADL recognition based on the combination of RFID and accelerometer sensing," in *Pervasive Computing Technologies for Healthcare, 2008. PervasiveHealth 2008. Second International Conference on*, 2008, pp. 258–263.
- [122] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *ACM SigKDD Explorations Newsletter*, vol. 12, no. 2, pp. 74–82, 2011.
- [123] R. Bogue, "MEMS sensors: past, present and future," *Sensor Review*, vol. 27, no. 1, pp. 7–13, 2007.
- [124] R. E. Mayoitia, 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.
- [125] A. Cranny, A. Beriain, H. Solar, G. Tartarisco, and G. Pioggia, "Vital Sign Sensing Technology," in *Systems Design for Remote Healthcare*, Maharatna, Koushik and Bonfiglio, Silvio, Ed. Springer New York, 2014, pp. 55–92.
- [126] C. Kendell and E. D. Lemaire, "Effect of mobility devices on orientation sensors that contain magnetometers," *J. Rehabil. Res. Dev*, vol. 46, no. 7, pp. 957–962, 2009.
- [127] A. Bulling, U. Blanke, and B. Schiele, "A tutorial on human activity recognition using body-worn inertial sensors," *ACM Computing Surveys (CSUR)*, vol. 46, no. 3, p. 33, 2014.
- [128] D. Biswas, A. Cranny, A. Rahim, N. Gupta, N. Harris, K. Maharatna, and S. Ortmann, "On the sensor choice and data analysis for classification of elementary upper limb movements," in *Biomedical and Health Informatics (BHI), 2014 IEEE-EMBS International Conference on*, 2014, pp. 744–747.
- [129] X. Sun, H. Kashima, and N. Ueda, "Large-Scale Personalized Human Activity Recognition Using Online Multitask Learning," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 25, no. 11, pp. 2551–2563, 2013.
- [130] O. Amft, M. Kusserow, and G. Tröster, "Probabilistic parsing of dietary activity events," in *4th International Workshop on Wearable and Implantable Body Sensor Networks (BSN 2007)*, 2007, pp. 242–247.
- [131] O. Amft, "Self-taught learning for activity spotting in on-body motion sensor data," in *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, 2011, pp. 83–86.
- [132] U. Blanke, B. Schiele, M. Kreil, P. Lukowicz, B. Sick, and T. Gruber, "All for one or one for all? Combining Heterogeneous Features for Activity Spotting," in *Pervasive Computing and Communications Workshops (PERCOM Workshops), 2010 8th IEEE International Conference on*, 2010, pp. 18–24.
- [133] A. Bulling, C. Weichel, and H. Gellersen, "EyeContext: Recognition of high-level contextual cues from human visual behaviour," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2013, pp. 305–308.
- [134] K. Van Laerhoven, D. Kilian, and B. Schiele, "Using rhythm awareness in long-term activity recognition," in *Wearable Computers, 2008. ISWC 2008. 12th IEEE International Symposium on*, 2008, pp. 63–66.
- [135] A. Kapoor and E. Horvitz, "Experience sampling for building predictive user models: a comparative study," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2008, pp. 657–666.
- [136] H. Bayati, J. del R Millán, and R. Chavarriaga, "Unsupervised adaptation to on-body sensor displacement in acceleration-based activity recognition," in *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, 2011, pp. 71–78.

- [137] K. Kunze and P. Lukowicz, "Dealing with sensor displacement in motion-based onbody activity recognition systems," in *Proceedings of the 10th international conference on Ubiquitous computing*, 2008, pp. 20–29.
- [138] R. Balani, "Energy consumption analysis for bluetooth, wifi and cellular networks," *Online*. <http://nesl.ee.ucla.edu/fw/documents/reports/2007/PowerAnalysis.pdf>, 2007.
- [139] K. Van Laerhoven and E. Berlin, "When else did this happen? efficient subsequence representation and matching for wearable activity data," in *Wearable Computers, 2009. ISWC'09. International Symposium on*, 2009, pp. 101–104.
- [140] E. Guenterberg, S. Ostadabbas, H. Ghasemzadeh, and R. Jafari, "An automatic segmentation technique in body sensor networks based on signal energy," in *Proceedings of the Fourth International Conference on Body Area Networks*, 2009, p. 21.
- [141] A. F. Bobick, S. S. Intille, J. W. Davis, F. Baird, C. S. Pinhanez, L. W. Campbell, Y. A. Ivanov, A. Schütte, and A. Wilson, "Perceptual user interfaces: the KidsRoom," *Communications of the ACM*, vol. 43, no. 3, pp. 60–61, 2000.
- [142] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," in *Wearable and Implantable Body Sensor Networks, 2006. BSN 2006. International Workshop on*, 2006, p. 4–pp.
- [143] J.-Y. Yang, J.-S. Wang, and Y.-P. Chen, "Using acceleration measurements for activity recognition: An effective learning algorithm for constructing neural classifiers," *Pattern recognition letters*, vol. 29, no. 16, pp. 2213–2220, 2008.
- [144] A. Mannini and A. M. Sabatini, "Machine learning methods for classifying human physical activity from on-body accelerometers," *Sensors*, vol. 10, no. 2, pp. 1154–1175, 2010.
- [145] S. Theodoridis, A. Pikrakis, K. Koutroumbas, and D. Cavouras, *Pattern Recognition*, Fourth. Academic Press, 2009.
- [146] D. Biswas, A. Cranny, N. Gupta, K. Maharatna, and S. Ortmann, "Recognition of Elementary Upper Limb Movements in an Activity of Daily Living using Data from Wrist Mounted Accelerometers," in *Health Informatics (ICHI), 2014 IEEE-Computer Society International Conference on*, 2014, pp. 232–237.
- [147] J. Mantyjarvi, J. Himberg, and T. Seppanen, "Recognizing human motion with multiple acceleration sensors," in *Systems, Man, and Cybernetics, 2001 IEEE International Conference on*, 2001, vol. 2, pp. 747–752.
- [148] A. K. Jain, R. P. W. Duin, and J. Mao, "Statistical pattern recognition: A review," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 22, no. 1, pp. 4–37, 2000.
- [149] L. Atallah and G.-Z. Yang, "The use of pervasive sensing for behaviour profiling—a survey," *Pervasive and Mobile Computing*, vol. 5, no. 5, pp. 447–464, 2009.
- [150] C. Zhu and W. Sheng, "Recognizing human daily activity using a single inertial sensor," in *Intelligent Control and Automation (WCICA), 2010 8th World Congress on*, 2010, pp. 282–287.
- [151] C.-C. Yang and Y.-L. Hsu, "A review of accelerometry-based wearable motion detectors for physical activity monitoring," *Sensors*, vol. 10, no. 8, pp. 7772–7788, 2010.
- [152] N. Oliver, A. P. Pentland, and F. Berard, "Lafter: Lips and face real time tracker," in *Computer Vision and Pattern Recognition, 1997. Proceedings., 1997 IEEE Computer Society Conference on*, 1997, pp. 123–129.
- [153] S. Mitra and T. Acharya, "Gesture recognition: A survey," *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, vol. 37, no. 3, pp. 311–324, 2007.
- [154] P. Natarajan and R. Nevatia, "Coupled hidden semi markov models for activity recognition," in *Motion and Video Computing, 2007. WMVC'07. IEEE Workshop on*, 2007, pp. 10–10.

- [155] K. Kunze, M. Barry, E. A. Heinz, P. Lukowicz, D. Majoe, and J. Gutknecht, "Towards Recognizing Tai Chi Â? An Initial Experiment Using Wearable Sensors," in *Applied Wearable Computing (IFAWC), 2006 3rd International Forum on*, 2006, pp. 1–6.
- [156] K.-T. Song and Y.-Q. Wang, "Remote activity monitoring of the elderly using a two-axis accelerometer," in *Proceedings of the CACS Automatic Control Conference*, 2005, pp. 18–19.
- [157] O. Banos, M. Damas, H. Pomares, A. Prieto, and I. Rojas, "Daily living activity recognition based on statistical feature quality group selection," *Expert Systems with Applications*, vol. 39, no. 9, pp. 8013–8021, 2012.
- [158] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *Neural Networks, IEEE Transactions on*, vol. 13, no. 2, pp. 415–425, 2002.
- [159] M. Mathie, B. G. Celler, N. H. Lovell, and A. Coster, "Classification of basic daily movements using a triaxial accelerometer," *Medical and Biological Engineering and Computing*, vol. 42, no. 5, pp. 679–687, 2004.
- [160] C. Pham, T. Plötz, and P. Olivier, "A dynamic time warping approach to real-time activity recognition for food preparation," in *Ambient Intelligence*, Springer, 2010, pp. 21–30.
- [161] U. Blanke, R. Rehner, and B. Schiele, "South by South-East or Sitting at the Desk: Can Orientation be a Place?," in *Wearable Computers (ISWC), 2011 15th Annual International Symposium on*, 2011, pp. 43–46.
- [162] T. Stiefmeier, D. Roggen, and G. Tröster, "Gestures are strings: efficient online gesture spotting and classification using string matching," in *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, p. 16.
- [163] R. Gutierrez-Osuna, "Lecture 13: Validation," 2012. [Online]. Available: [http://research.cs.tamu.edu/prism/lectures/iss/iss\\_113.pdf](http://research.cs.tamu.edu/prism/lectures/iss/iss_113.pdf). [Accessed: 11–01-2014].
- [164] J. A. Ward, P. Lukowicz, and H. W. Gellersen, "Performance metrics for activity recognition," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 1, p. 6, 2011.
- [165] D. Minnen, T. Westeyn, T. Starner, J. Ward, and P. Lukowicz, "Performance metrics and evaluation issues for continuous activity recognition," *Performance Metrics for Intelligent Systems*, p. 4, 2006.
- [166] S. Antifakos, F. Michahelles, and B. Schiele, "Proactive instructions for furniture assembly," in *UbiComp 2002: Ubiquitous Computing*, Springer, 2002, pp. 351–360.
- [167] G. Fang, W. Gao, and D. Zhao, "Large vocabulary sign language recognition based on hierarchical decision trees," in *Proceedings of the 5th international conference on Multimodal interfaces*, 2003, pp. 125–131.
- [168] Y.-J. Hong, I.-J. Kim, S. C. Ahn, and H.-G. Kim, "Mobile health monitoring system based on activity recognition using accelerometer," *Simulation Modelling Practice and Theory*, vol. 18, no. 4, pp. 446–455, 2010.
- [169] A. Fleury, M. Vacher, and N. Noury, "SVM-based multimodal classification of activities of daily living in health smart homes: sensors, algorithms, and first experimental results," *Information Technology in Biomedicine, IEEE Transactions on*, vol. 14, no. 2, pp. 274–283, 2010.
- [170] J. Bussmann, W. Martens, J. Tulen, F. Schasfoort, H. Van Den Berg-Emons, and H. Stam, "Measuring daily behavior using ambulatory accelerometry: the Activity Monitor," *Behavior Research Methods, Instruments, & Computers*, vol. 33, no. 3, pp. 349–356, 2001.
- [171] R. Muscillo, M. Schmid, S. Conforto, and T. D'Alessio, "Early recognition of upper limb motor tasks through accelerometers: real-time implementation of a DTW-based algorithm," *Computers in biology and medicine*, vol. 41, no. 3, pp. 164–172, 2011.
- [172] N. Kern, B. Schiele, and A. Schmidt, "Multi-sensor activity context detection for wearable computing," in *Ambient Intelligence*, Springer, 2003, pp. 220–232.

- [173] P. Lukowicz, J. A. Ward, H. Junker, M. Stäger, G. Tröster, A. Atrash, and T. Starner, "Recognizing workshop activity using body worn microphones and accelerometers," in *Pervasive Computing*, Springer, 2004, pp. 18–32.
- [174] H. Junker, O. Amft, P. Lukowicz, and G. Tröster, "Gesture spotting with body-worn inertial sensors to detect user activities," *Pattern Recognition*, vol. 41, no. 6, pp. 2010–2024, 2008.
- [175] P. Zappi, C. Lombriser, T. Stiefmeier, E. Farella, D. Roggen, L. Benini, and G. Tröster, "Activity recognition from on-body sensors: accuracy-power trade-off by dynamic sensor selection," in *Wireless sensor networks*, Springer, 2008, pp. 17–33.
- [176] O. Amft, C. Lombriser, T. Stiefmeier, and G. Tröster, "Recognition of user activity sequences using distributed event detection," in *Smart Sensing and Context*, Springer, 2007, pp. 126–141.
- [177] G. Bailador, D. Roggen, G. Tröster, and G. Triviño, "Real time gesture recognition using continuous time recurrent neural networks," in *Proceedings of the ICST 2nd international conference on Body area networks*, 2007, p. 15.
- [178] S. Wang, J. Yang, N. Chen, X. Chen, and Q. Zhang, "Human activity recognition with user-free accelerometers in the sensor networks," in *Neural Networks and Brain, 2005. ICNN\&B'05. International Conference on*, 2005, vol. 2, pp. 1212–1217.
- [179] S. L. Wolf, H. Newton, D. Maddy, S. Blanton, Q. Zhang, C. J. Winstein, D. M. Morris, and K. Light, "The Excite Trial: relationship of intensity of constraint induced movement therapy to improvement in the wolf motor function test," *Restorative neurology and neuroscience*, vol. 25, no. 5, pp. 549–562, 2007.
- [180] D. M. Morris, G. Uswatte, J. E. Crago, E. W. Cook III, and E. Taub, "The reliability of the Wolf Motor Function Test for assessing upper extremity function after stroke," *Archives of physical medicine and rehabilitation*, vol. 82, no. 6, pp. 750–755, 2001.
- [181] S. L. Wolf, P. A. Thompson, D. M. Morris, D. K. Rose, C. J. Winstein, E. Taub, C. Giuliani, and S. L. Pearson, "The EXCITE trial: attributes of the Wolf Motor Function Test in patients with subacute stroke," *Neurorehabilitation and Neural Repair*, vol. 19, no. 3, pp. 194–205, 2005.
- [182] C. Wu, T. Fu, K. Lin, C. Feng, K. Hsieh, H. Yu, C. Lin, C. Hsieh, and H. Ota, "Assessing the streamlined Wolf Motor Function Test as an outcome measure for stroke rehabilitation," *Neurorehabilitation and neural repair*, vol. 25, no. 2, pp. 194–199, 2011.
- [183] D. Biswas, D. Corda, G. Baldus, A. Cranny, K. Maharatna, J. Achner, J. Klemke, M. Jöbges, and S. Ortmann, "Recognition of elementary arm movements using orientation of a tri-axial accelerometer located near the wrist," *Physiological measurement*, vol. 35, no. 9, p. 1751, 2014.
- [184] "Shimmer Research - Shimmer 9dof calibration user manual. rev 0.1a." [Online]. Available: [www.shimmer-research.com](http://www.shimmer-research.com). [Accessed: 01–02–2012].
- [185] W. Fong, S. Ong, and A. Nee, "Methods for in-field user calibration of an inertial measurement unit without external equipment," *Measurement Science and Technology*, vol. 19, no. 8, p. 085202, 2008.
- [186] A. M. Sabatini, "Quaternion-based strap-down integration method for applications of inertial sensing to gait analysis," *Medical and Biological Engineering and Computing*, vol. 43, no. 1, pp. 94–101, 2005.
- [187] X. Yun and E. R. Bachmann, "Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *Robotics, IEEE Transactions on*, vol. 22, no. 6, pp. 1216–1227, 2006.
- [188] H. J. Luinge, P. H. Veltink, and C. T. Baten, "Ambulatory measurement of arm orientation," *Journal of biomechanics*, vol. 40, no. 1, pp. 78–85, 2007.
- [189] H. Zhou, H. Hu, N. D. Harris, and J. Hammerton, "Applications of wearable inertial sensors in estimation of upper limb movements," *Biomedical Signal Processing and Control*, vol. 1, no. 1, pp. 22–32, 2006.

- [190] D. Roetenberg, P. J. Slycke, and P. H. Veltink, "Ambulatory position and orientation tracking fusing magnetic and inertial sensing," *Biomedical Engineering, IEEE Transactions on*, vol. 54, no. 5, pp. 883–890, 2007.
- [191] M. El-Gohary and J. McNames, "Shoulder and elbow joint angle tracking with inertial sensors," *Biomedical Engineering, IEEE Transactions on*, vol. 59, no. 9, pp. 2635–2641, 2012.
- [192] R. Mahony, T. Hamel, and J.-M. Pflimlin, "Nonlinear complementary filters on the special orthogonal group," *Automatic Control, IEEE Transactions on*, vol. 53, no. 5, pp. 1203–1218, 2008.
- [193] S. O. Madgwick, A. J. Harrison, and R. Vaidyanathan, "Estimation of IMU and MARG orientation using a gradient descent algorithm," in *Rehabilitation Robotics (ICORR), 2011 IEEE International Conference on*, 2011, pp. 1–7.
- [194] "Altera, DE2-115 User Manual, 2012." [Online]. Available: [ftp://ftp.altera.com/up/pub/AlteraMaterial/12.1/Boards/DE2-115/DE2\\_115\\_User\\_Manual.pdf](ftp://ftp.altera.com/up/pub/AlteraMaterial/12.1/Boards/DE2-115/DE2_115_User_Manual.pdf). [Accessed: 01–06-2014].
- [195] "SerialPort Class." [Online]. Available: <http://msdn.microsoft.com/en-us/library/system.io.ports.serialport%28v=vs.110%29.aspx>. [Accessed: 06-2014].
- [196] I. Kim, S. Im, E. Hong, S. Ahn, and H. Kim, "Adl classification using triaxial accelerometers and rfid," in *International Conference on Ubiquitous Computing Convergence Technology*, 2007.
- [197] S. A. Lowe and G. ÓLaighin, "Monitoring human health behaviour in one's living environment: A technological review," *Medical engineering & physics*, vol. 36, no. 2, pp. 147–168, 2014.
- [198] J. L. Semmlow, *Biomedical and Medical Image Processing*, 2nd ed. CRC Press, 2008.
- [199] T. Li, S. Zhu, and M. Ogihara, "Using discriminant analysis for multi-class classification: an experimental investigation," *Knowledge and information systems*, vol. 10, no. 4, pp. 453–472, 2006.
- [200] C. M. Bishop and others, *Pattern recognition and machine learning*, vol. 1. springer New York, 2006.
- [201] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 27, 2011.
- [202] J. C. Fernandez Caballero, F. J. Martínez, C. Hervás, and P. A. Gutiérrez, "Sensitivity versus accuracy in multiclass problems using memetic pareto evolutionary neural networks," *Neural Networks, IEEE Transactions on*, vol. 21, no. 5, pp. 750–770, 2010.
- [203] T. Warren Liao, "Clustering of time series data—a survey," *Pattern recognition*, vol. 38, no. 11, pp. 1857–1874, 2005.
- [204] B. Andreopoulos, A. An, X. Wang, and M. Schroeder, "A roadmap of clustering algorithms: finding a match for a biomedical application," *Briefings in Bioinformatics*, vol. 10, no. 3, pp. 297–314, 2009.
- [205] T. Huynh and B. Schiele, "Analyzing features for activity recognition," in *Proceedings of the 2005 joint conference on Smart objects and ambient intelligence: innovative context-aware services: usages and technologies*, 2005, pp. 159–163.
- [206] L. A. Leiva and E. Vidal, "Warped K-Means: An algorithm to cluster sequentially-distributed data," *Information Sciences*, vol. 237, pp. 196–210, 2013.
- [207] J. K. Aggarwal and M. S. Ryoo, "Human activity analysis: A review," *ACM Computing Surveys (CSUR)*, vol. 43, no. 3, p. 16, 2011.
- [208] R. T. T. Hastie and J. Friedman, *The Elements of Statistical Learning*. Springer, 2009.
- [209] D. Biswas, A. Cranny, N. Gupta, K. Maharatna, J. Achner, J. Klemke, M. Jöbges and S. Ortmann, "Recognizing upper limb movements with wrist worn inertial sensors using k-means clustering classification," *Human Movement Science*, vol. 40, pp. 59–76, 2015.

- [210] J. Mao and A. K. Jain, "A self-organizing network for hyperellipsoidal clustering (HEC)," *Neural Networks, IEEE Transactions on*, vol. 7, no. 1, pp. 16–29, 1996.
- [211] H. Alemdar and C. Ersoy, "Wireless sensor networks for healthcare: A survey," *Computer Networks*, vol. 54, no. 15, pp. 2688–2710, 2010.
- [212] D. Biswas, E. B. Mazomenos, and K. Maharatna, "ECG compression for remote healthcare systems using selective thresholding based on energy compaction," in *Signals, Systems, and Electronics (ISSSE), 2012 International Symposium on*, 2012, pp. 1–6.
- [213] P. Nair, D. Kudithipudi, and E. John, "Design and implementation of a CMOS non-restoring divider," in *Region 5 Conference, 2006 IEEE*, 2006, pp. 211–217.
- [214] K. Vijeyakumar, V. Sumathy, P. Vasakipriya, and A. Dinesh Babu, "FPGA implementation of low power high speed square root circuits," in *Computational Intelligence & Computing Research (ICCIC), 2012 IEEE International Conference on*, 2012, pp. 1–5.
- [215] D. De Caro, M. Genovese, E. Napoli, N. Petra, and A. G. Strollo, "Accurate Fixed Point Logarithmic Converter," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 61, no. 7, pp. 526 – 530, 2014.
- [216] K. Kota and J. R. Cavallaro, "Numerical accuracy and hardware tradeoffs for CORDIC arithmetic for special-purpose processors," *Computers, IEEE Transactions on*, vol. 42, no. 7, pp. 769–779, 1993.
- [217] E. Grass, B. Sarker, and K. Maharatna, "A dual-mode synchronous/asynchronous CORDIC processor," in *Asynchronous Circuits and Systems, 2002. Proceedings. Eighth International Symposium on*, 2002, pp. 76–83.
- [218] C. Y. Kang and E. E. Swartzlander, "Digit-pipelined direct digital frequency synthesis based on differential CORDIC," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 53, no. 5, pp. 1035–1044, 2006.
- [219] L. Vachhani, K. Sridharan, and P. K. Meher, "Efficient CORDIC algorithms and architectures for low area and high throughput implementation," *Circuits and Systems II: Express Briefs, IEEE Transactions on*, vol. 56, no. 1, pp. 61–65, 2009.
- [220] C.-H. Lin and A.-Y. Wu, "Mixed-scaling-rotation CORDIC (MSR-CORDIC) algorithm and architecture for high-performance vector rotational DSP applications," *Circuits and Systems I: Regular Papers, IEEE Transactions on*, vol. 52, no. 11, pp. 2385–2396, 2005.
- [221] R. Kunemund, H. Soldner, S. Wohlleben, and T. Noll, "Cordic processor with carry-save architecture," in *Solid-State Circuits Conference, 1990. ESSCIRC'90. Sixteenth European*, 1990, vol. 1, pp. 193–196.
- [222] A. Acharyya, K. Maharatna, B. M. Al-Hashimi, and J. Reeve, "Coordinate rotation based low complexity ND FastICA algorithm and architecture," *Signal Processing, IEEE Transactions on*, vol. 59, no. 8, pp. 3997–4011, 2011.
- [223] M. D. Ercegovac and T. Lang, *Digital arithmetic*. Elsevier, 2003.
- [224] J. Schuster, K. Gupta, R. Hoare, and A. K. Jones, "Speech silicon: An FPGA architecture for real-time hidden Markov-model-based speech recognition," *EURASIP Journal on Embedded Systems*, vol. 2006, no. 1, pp. 10–10, 2006.
- [225] B. Rohrer, S. Fasoli, H. I. Krebs, R. Hughes, B. Volpe, W. R. Frontera, J. Stein, and N. Hogan, "Movement smoothness changes during stroke recovery," *The Journal of Neuroscience*, vol. 22, no. 18, pp. 8297–8304, 2002.
- [226] D. Biswas, A. Cranny, K. Maharatna, J. Achner, J. Klemke, M. Jöbges, and S. Ortmann, "Movement fluidity of the impaired arm during stroke rehabilitation," in *IEEE Biomedical and Health Informatics (BHI)*, 2014.