An Inertial Sensing Module for Movement and Posture Monitoring for Assisted Living

A dissertation

submitted in partial fulfillment of

the requirements for the degree of

Master of Technology

by

Praveen Kumar

(113300024)

under the supervision of

Prof. P. C. Pandey

Biomedical Engineering Group
Department of Biosciences and Bioengineering
Indian Institute of Technology Bombay

June 2013

Indian Institute of Technology Bombay

M. Tech. Dissertation Approval

This dissertation entitled **"An Inertial Sensing Module for Movement and Posture Monitoring for Assisted Living"** by **Praveen Kumar** (Roll No. 113300024) is approved, after the successful completion of *viva voce* examination, for the award of the degree of **Master of Technology in Biomedical Engineering.**

Supervisor .... (Prof. P. C. Pandey)

Examiners .... (Prof. V. Rajbabu)

.... (Prof. L. R. Subramanyan)

Chairman .... (Prof. Joseph John)

Date: 28th June 2013

Place: Mumbai

Declaration

I declare that this written submission represents my ideas and my own words and where other's ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the institute and can also evoke penal action from sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

Praveen Kr.

Date: 28 June, 2013

Praveen Kumar

Place: Mumbai

(Roll No. 113300024)

Praveen Kumar / Prof. P. C. Pandey (Supervisor): "An Inertial Sensing Module for Movement and Posture Monitoring for Assisted Living," *M. Tech. dissertation*, Department of Biosciences and Bioengineering, Indian Institute of Technology Bombay, June 2013.

ABSTRACT

Fall is a major problem for the elderly persons and persons suffering from neuromuscular disorders. Movement and posture monitoring can help in alerting the emergency service and relatives in case of such an occurrence. The project objective is to develop a sensor module for tracking motion of a body part to which it is attached by continuously monitoring its orientation and acceleration. For this purpose, an inertial sensing module (ISM) is developed. It consists of a microcontroller, an integrated sensor having tri-axial accelerometer and tri-axial gyroscope for sensing the linear acceleration and angular velocity, and non-volatile memory for recording the sensed variables. Bluetooth has been used for controlling the ISM operations and for data transfer. Data acquisition can be carried out using sample-by-sample transfer or burst mode transfer. The sensor data have been tested for accuracy and precision. A real-time fall detection algorithm has also been implemented. The module can be used as a recording device for actigraphy for diagnosis of sleep disorders. It can record data continuously for approximately 2 hours at a sampling frequency of 100 Hz. A number of such modules wirelessly connected to a central processing unit can be used as a system for continuous monitoring of movement and posture of the person and can serve as an aid for assisted living.

CONTENTS

LIST OF ABBREVIATIONS

Abbreviation	Explanation
ADC	analog to digital convertor
ADL	activities of daily life
CMG	control moment gyroscope
CS	chip select
DAQ	data acquisition
DMP	digital motion processor
FIFO	first in first out
I2C	inter integrated circuit
ID	identity
I/O	input/output
IRQ	interrupt request
ISM	inertial sensing module
ISR	interrupt service routine
MEMS	micro electro mechanical systems
MISO	master in slave out
MOSI	master out slave in
OSCO	oscillator out
PC	personal computer
PCB	printed circuit board
PIC	peripheral interface controller
RAM	random access memory
SCK	serial clock

SD	standard deviation
SDA	serial data
SPI	serial peripheral interface
UART	universal asynchronous receiver/transmitter

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Background

One of the important tasks of a medical healthcare system is to provide assisted living to elderly persons and patients suffering from chronic diseases. A continuous monitoring of posture is required for people suffering from sleep disorders as they are prone to increased risk of heart attack, hypertension, and accidents. An individual suffering from neuromuscular disorder has an increased chance of losing control over his or her body which may lead to a fall. A severe fall may cause serious injuries and may lead to several complications and even death, if immediate and appropriate action is not taken. A continuous monitoring of the movement and posture of such individuals is required so that in case of an emergency the concerned persons can be informed to deliver the required assistance and treatment. Further, a continuous log of movement and posture data can be used for diagnostic purposes and for assessing the effect of treatments being provided to such persons. Such a log is also useful in actigraphy for diagnosis and treatment of sleep disorders.

1.2 Project objective

The aim of this project is to develop a system with multiple movement sensing modules, which are attached to different parts of the body, e.g. wrists, ankle, and trunk. These modules will wirelessly transmit the data related to movement and orientation to a central unit on a real-time basis. The software on the central unit will receive the data from these modules, and use them for preparing a map of relative position of the sensor modules for monitoring the posture and movement of the person without using magnetometers. The central unit can also be body worn. Wireless connectivity is required to avoid any movement restrictions. In an earlier design of a sensor module by Adithya [1], two sensor ICs (one tri-axial accelerometer IC and one tri-axial gyroscope IC) were interfaced to a microcontroller and a flash memory was provided for storing the sensed data. In the current project, an inertial sensing module (ISM) is designed by upgrading the earlier design for making it more compact and accurate and to provide wireless connectivity. An integrated sensor with tri-axial accelerometer and tri-axial gyroscope is used. An integrated sensor avoids errors of the sensor axes caused by misalignment during assembly. It also reduces the chip count and makes the module more compact. Wireless connectivity is provided by using a Bluetooth module (RN-42). Different ways of signal acquisition are implemented and tested along with a run-time fall detection algorithm.

1.3 Report outline

The second chapter provides literature survey for posture and movement tracking. The hardware design of inertial sensing module (ISM) is presented in Chapter 3. The signal acquisition methods and implementation of real-time fall detection algorithm are described in Chapter 4 and Chapter 5 respectively. The last chapter gives the summary of the work done and conclusions.

Chapter 2

LITERATURE SURVEY

2.1 Introduction

Various motion monitoring products available in the market are based on different sensing systems: optical tracking system, image based systems, acoustic tracking systems, and inertial tracking systems. Optical tracking systems detect motion by sensing light patterns, and hence are immune to disturbances due to ferromagnetic and conductive material in the environment [2], [3]. Image based systems use multiple cameras for body and face tracking [4], [5]. Use of multiple cameras makes it a very expensive technique. Acoustic sensing systems track motion by determining the velocities and displacement directions of acoustic wave in various combinations [6]. Inertial tracking systems use accelerometers and gyroscopes to detect linear acceleration and angular velocities [7] – [10]. Various other sensors like tilt sensor and magnetometer have also been used to determine the angle of inclination and direction of movement.

The inertial tracking system has certain advantages over other systems. The sensors used in this system are of low cost and compact size. Compared with magnetic tracking systems, they do not have interference problems. Unlike optical and image based systems, they do not impose restrictions on the space in which movement can be tracked. These advantages make it the most suitable system for movement and posture monitoring for assisted living. Some of the methods for fall detection using accelerometers are reviewed in Section 2.2 while those using gyroscopes are reviewed in Section 2.3. Fall detection using multiple sensors are reviewed in Section 2.4 and conclusions are presented in the last section.

2.2 Fall detection using accelerometers

Kangas *et al.* [11] used three tri-axial capacitive accelerometers to monitor falls and activities of daily living. The accelerometers were attached to the non-dominant wrist, same side of the waist, and the front side of the forehead. The three axes of head and waist accelerometers were aligned to the mediolateral, anteroposterior and vertical directions. The axis of the wrist worn accelerometer was also aligned to the same direction when the subject's arms were stretched to the sides keeping palm downwards. Each accelerometer had a data logger

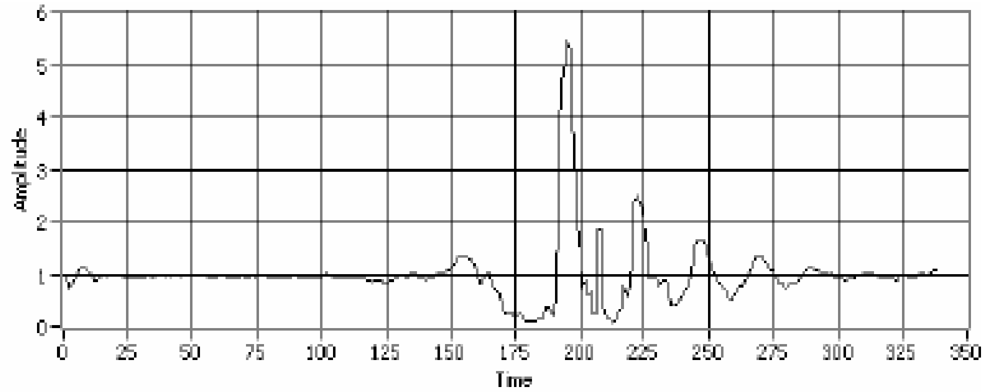


Figure 2.1: SV_{TOT} , as given in [11].

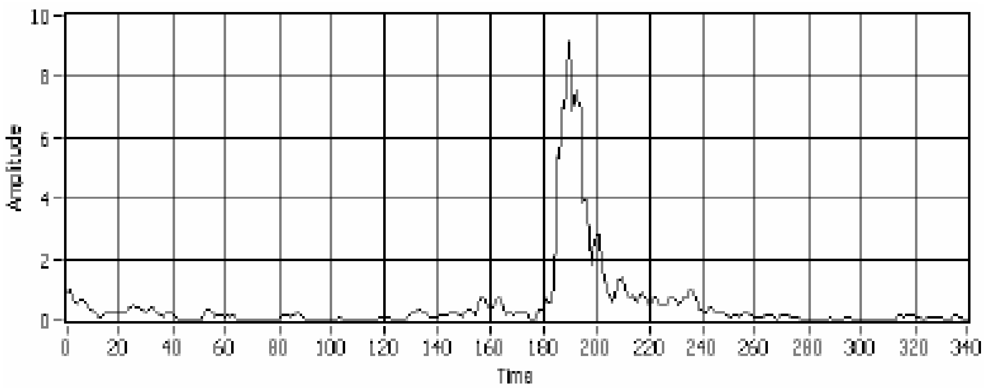


Figure 2.2:SV_{maxmin}, as given in [11].

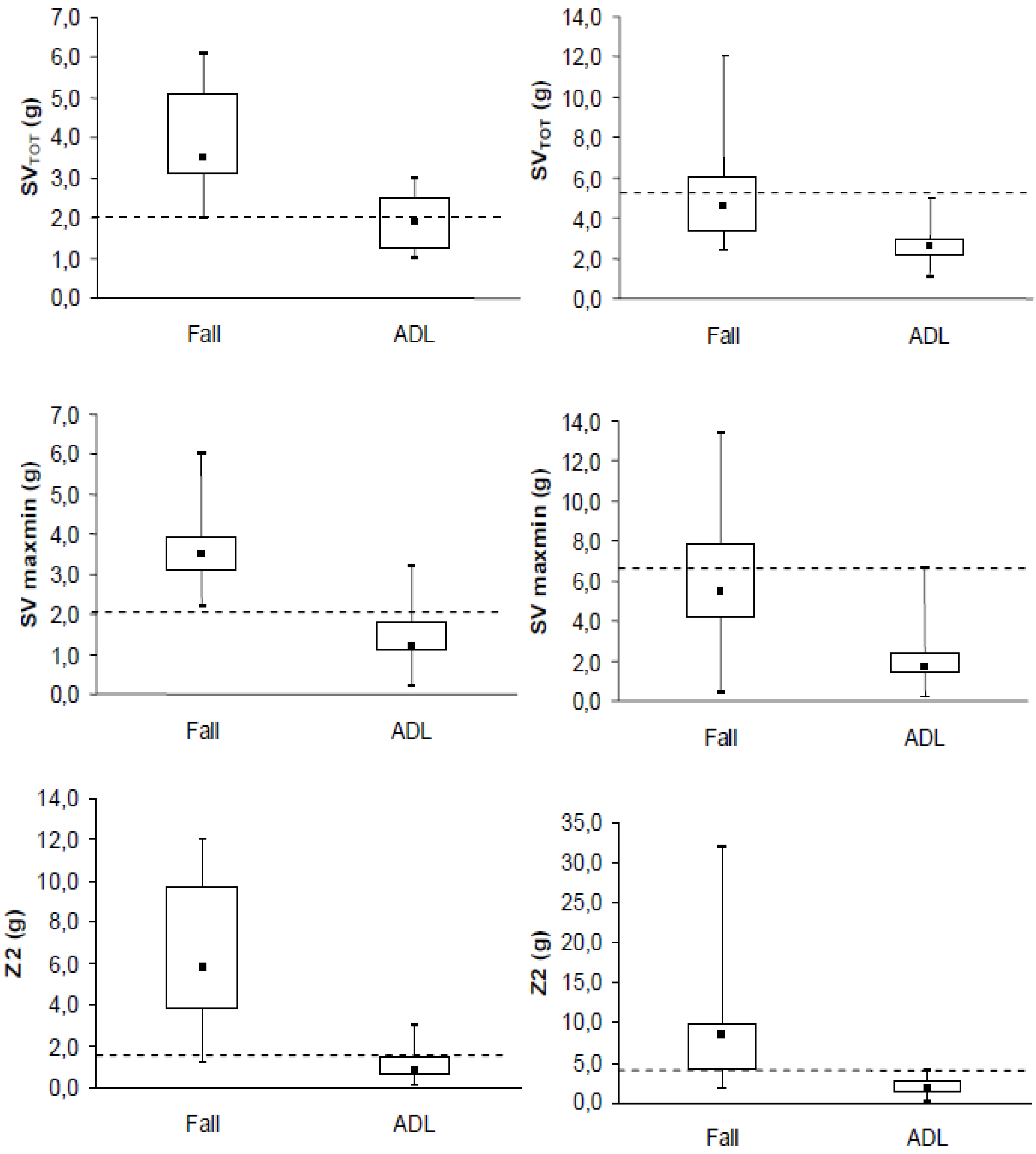
attached with a sampling frequency of 400 Hz. The data collected from two subjects (22 and 38 years) were transferred to a computer and were converted into gravitational units and processed. Activities of daily living (like walking, using stairs, picking up objects from the floor, posture transitions) and falls (like forward fall, backward fall, and lateral falls) were monitored. They prepared two different sets of data by filtering the accelerometer data using a low-pass filter and a high-pass filter with cut-off frequency of 0.25 Hz. The low-pass filtered data were used for posture detection while the high-pass data were used for motion analysis. Parameters like total sum vector SV_{TOT}, dynamic sum vector SV_D, and vertical acceleration Z_2 were calculated using the following equations,

$$SV_{TOT} = \sqrt{[A_x^2 + A_y^2 + A_z^2]}$$

$$SV_D = \sqrt{[A_{xHP}^2 + A_{yHP}^2 + A_{zHP}^2]}$$

$$Z_2 = (SV_{TOT}^2 + SV_D^2 - G^2) / 2G$$

where A_x, A_y, and A_z are the three components of the acceleration, subscript "HP" refers to high-pass filtered signals and *G* is the gravitational acceleration. An another parameter SV_{maxmin} was calculated by taking the difference between the maximum and minimum acceleration values of each axis with a sliding window of size 0.6 s. SV_{maxmin} was calculated to determine the rate of change of acceleration for deciding whether the posture transition was



(a) (b)

Figure 2.3: Quartile box plots of parameters measured from (a) waist (b) wrist, during falls and ADL. Selected threshold values (th) are marked (---) [11].

intentional. As can be seen in Figure 2.1, there is a valley before the maximum peak. The dip is associated with the start of a fall while the peak of SV_{TOT} (Figure 2.1) and SV_{maxmin} (Figure 2.2) gave the impact. The difference between the minimum value of SV_{TOT} in the valley and the maximum value of SV_{TOT} at the peak gave the fall duration. Integrating the area under the valley gave the velocity just before the impact. The low-pass filtered data 2 s after the impact were used to determine the final posture of the body. As can be seen in Figure 2.3, there was a

Table 2.1: Threshold values for fall detection algorithm as given in [11].

	Waist		Wrist		Head	
Parameter	Th	Se / Sp (%)[a]	Th	Se / Sp (%)	Th	Se / Sp (%)
SV_{TOT}	2.0 g	100/100	5.2 g	45/100	2.0 g	100/100
SV_D	1.7 g	100/100	5.1 g	32/100	1.2 g	100/100
SV_{maxmin}	2.0 g	100/100	6.5 g	41/100	1.7 g	100/100
Z_2	1.5 g	95/100	3.9 g	75/100	1.8 g	100/100

Th = threshold, Se / Sp = sensitivity and specificity (%), g = acceleration of gravity. [a] Posture detection after fall included

slight overlapping of fall and ADL values of SV_{TOT}, SV_{maxmin}, and Z_2 measured from the waist, while the overlapping of the same parameters measured from the wrist was more. This overlapping of the values can be clearly seen in the figure. There was no overlapping between fall and ADL values measured from the head unit. The different thresholds applied for various parameters have been summarized in Table 2.1. As can be seen from the table, the sensitivity and specificity of fall detected from head unit was always 100% as there was no overlapping of values. The waist unit also had a 100% sensitivity and specificity with an exception of 95% sensitivity due to Z_2 values. The wrist unit had a specificity of 100% for all three variables but the sensitivity was poor. Thus head and waist proved to be good locations while wrist was found to be not a suitable location for placing the sensor module.

Bourke *et al.* [12] also used tri-axial accelerometers for data acquisition but they located these sensors at the trunk (anterior part of sternum) and the thigh (mid-point of femur bone). The signals were acquired with sampling frequency of 1 kHz and resolution of 12 bits. Two separate studies were performed with 10 subjects each for monitoring fall and activities of daily living. Younger subjects of age 21 − 29 years were used for fall detection study while elder subjects of age between 70 − 83 years were used for ADL study. Eight different types of fall were performed by the younger subjects, while the elder ones performed ADL, like (1) sitting down and standing up from an armchair, a kitchen chair, a toilet seat, a low stool, and a bed; (2) getting in and out of a car seat, (3) walking 10 m, and (4) sitting down on and standing up from a bed. The resultant acceleration was calculated by using the same formula as given in [11]. Figure 2.4 shows the valley and the maximum peak during impact. An upper fall threshold (UFT) and lower fall threshold (LFT) were selected for thigh and trunk. Figure 2.5 shows the peak values for fall and various daily life activities. They found that the upper fall threshold of the trunk module differentiated fall from ADL with an accuracy of 100%. The upper peak value of the fall with knee flexion was lower than the upper peak value of fall

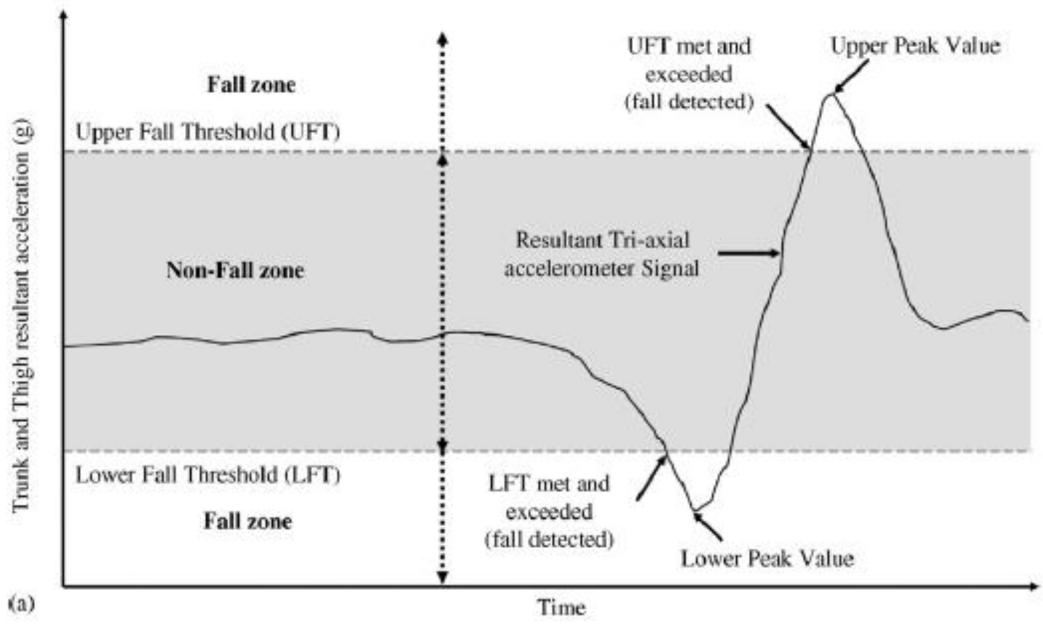


Figure 2.4: Trunk resultant vector signals as given in [12].

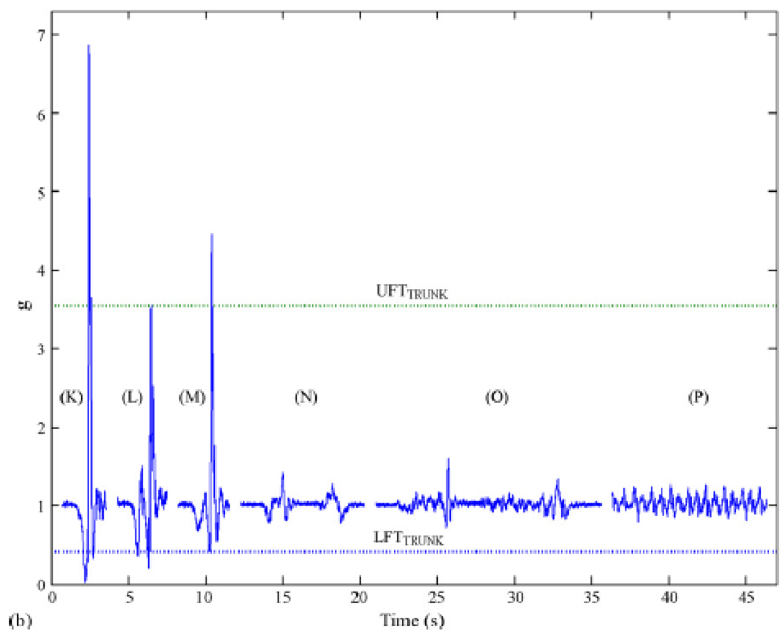


Figure 2.5: Peaks (K), (L), (M) shows fall and peaks (N), (O), and (P) show ADL as in [12].

with legs straight. The upper peak values of different types of fall and ADL is shown in Figure 2.6. Hence trunk and thigh also proved to be good location for sensor placement.

Helmi and Almodarresi [16] used LIS3LV02DQ tri-axial accelerometer. The accelerometer was attached to the waist of three male adults. The LIS3LV02DQ has a user selectable full scale of ±2 g and ±6 g. The full scale of ±2 g was used in their study. A 12-bit ADC was used to sample the output of the accelerometer at 22 Hz. They performed ten trials on each person for the activities related to moving forward, jumping, going upstairs and going

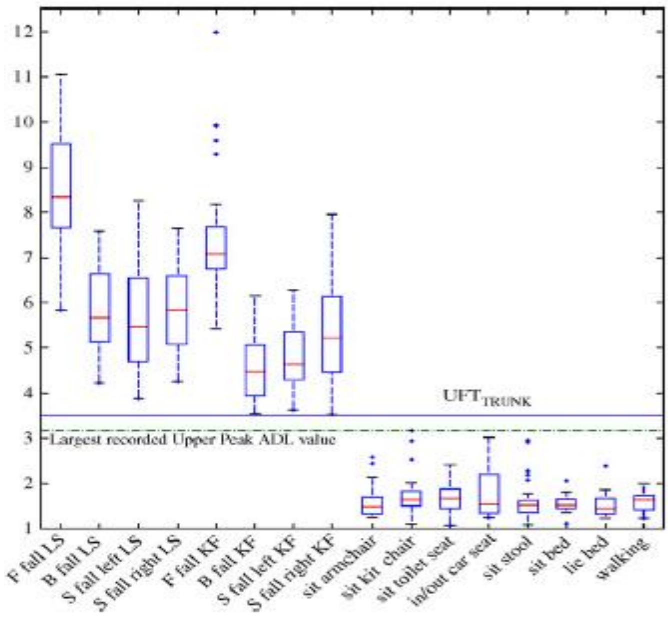


Figure 2.6: Box plot for trunk upper peak values as given in [12].

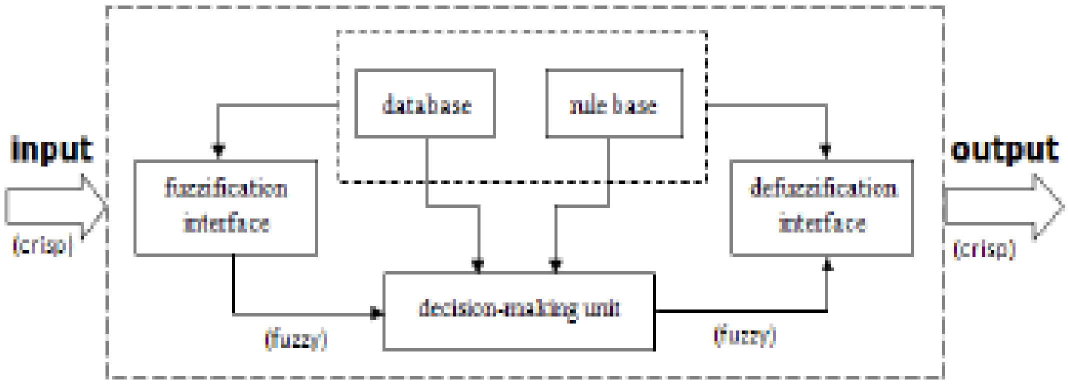


Figure 2.7: Structure of fuzzy inference system, as given in [16].

downstairs. They used fuzzy inference system for classification. Fuzzy logic requires less mathematics, less memory and executes faster. It comprises of five functional blocks as shown in Figure 2.7. It has a database which contains the membership functions of input and output of the system. The rule base block contains a set of if-then rules. These rules are implemented by fuzzy operators like AND, OR in the decision making unit. The fuzzification interface block takes the crisp input and determines the extent to which they belong to a particular fuzzy set. This fuzzification is done by comparing the crisp inputs with the membership functions. In the defuzzification interface block, output of each rule is aggregated into a single fuzzy set. Several defuzzification methods are then applied to resolve the single fuzzy set into a crisp output value. The centroid calculation method was used here in the defuzzification step. The overall accuracy achieved by them using fuzzy inference system was 95%. The accuracy obtained for moving forward (one step), jumping, going

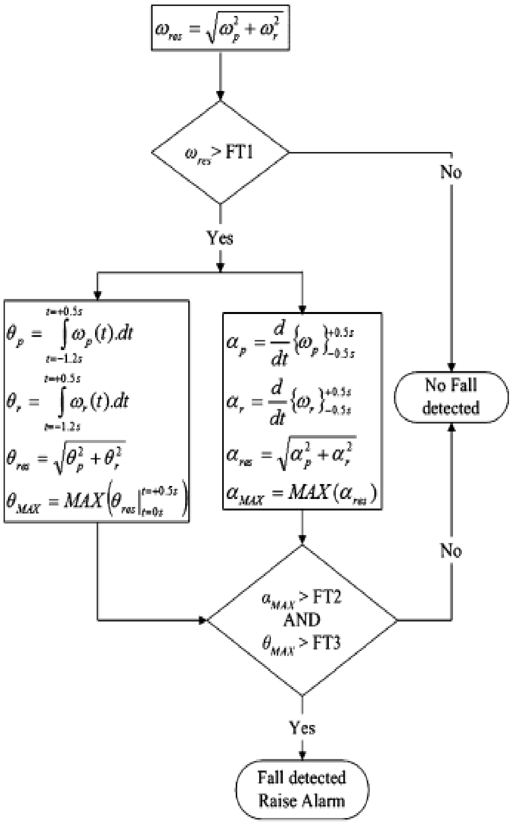


Figure 2.8: Fall detection algorithm used by Bourke *et al.* as given in [13].

upstairs (two stairs), and going downstairs (two stairs) were 100, 96.7, 93.3, and 93.3% respectively. Main problem faced was in distinguishing between going upstairs and going downstairs.

2.3 Fall detection using gyroscopes

Bourke *et al.* [13] used bi-axial gyroscopes instead of accelerometers for fall detection keeping all other parameters of data acquisition same as [12] as described in Section 2.2. They divided the data into two cases. One in which lowest peak value of fall is greater than highest peak value of ADL and the other in which fall and ADL regions are overlapping. They calculated the resultant angular velocity using the pitch, roll values and applied a threshold FT1 to it. If the resultant angular velocity crossed the threshold, two more parameters were calculated. The resultant angle movement was calculated by integrating the angular velocity for a time 1.2 s before the impact to 0.5 s after the impact. The resultant angular acceleration was calculated by differentiating the angular velocity for a time 0.5 s before the impact to 0.5 s after the impact. Two different thresholds FT2 and FT3 were applied to the calculated parameters angular movement and angular acceleration respectively. If both the thresholds FT2 and FT3 were crossed then fall was detected and alarm got raised. The whole algorithm is shown in Figure 2.8. They obtained a sensitivity and specificity of

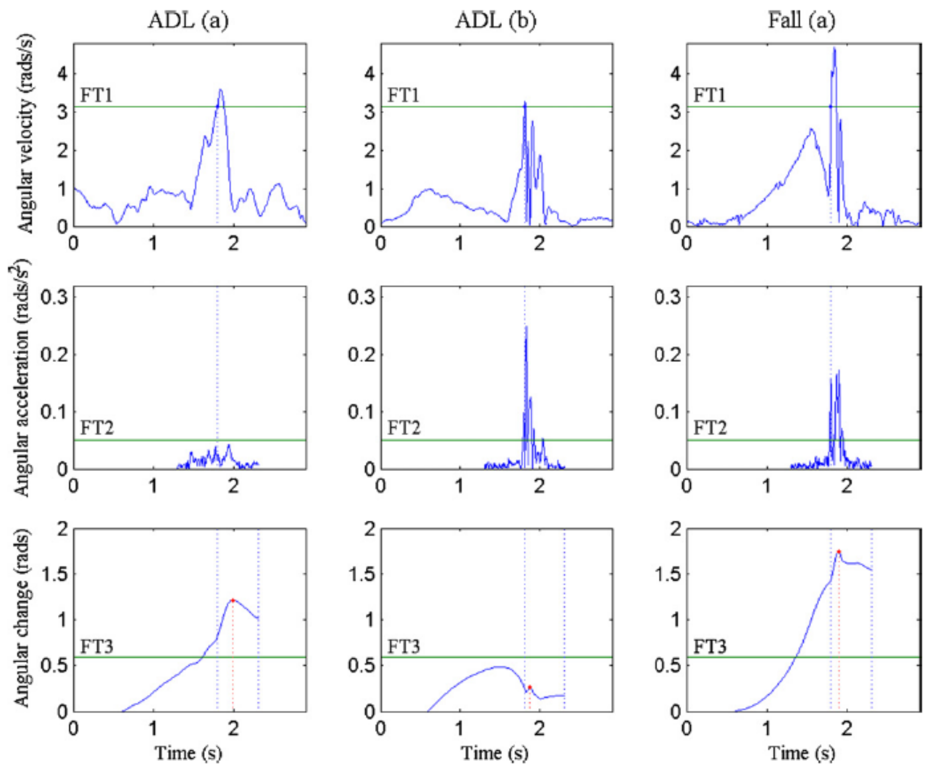


Figure 2.9: Thresholds applied to the resultant angular velocity, angular acceleration, and angular change of ADL and fall as given in [13].

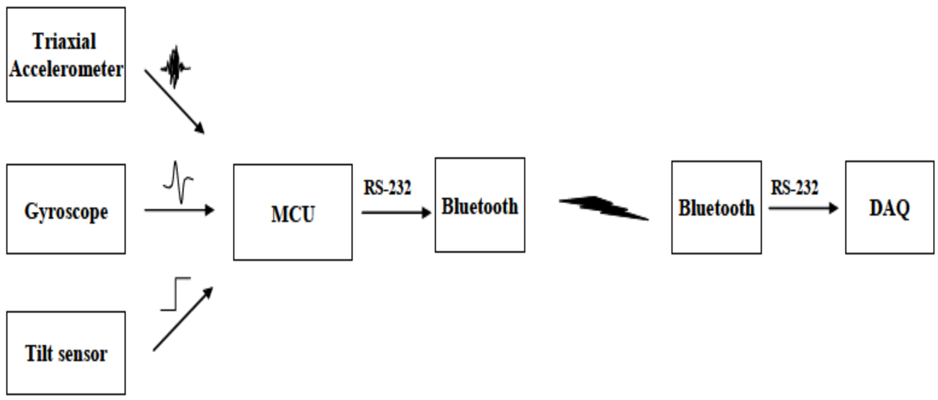


Figure 2.10: Real-time monitoring system for fall detection, as used in [14].

100%. As can be seen from Figure 2.9, in ADL (a) and ADL (b) the peak value crossed any two thresholds. In fall condition, all the three thresholds were crossed.

2.4 Fall detection using multiple sensors

Hwang et al. [14] used a system composed of real-time sensing module and a communication module as shown in Figure 2.10. The sensing module consisted of a tri-axial accelerometer, a gyroscope, and a tilt sensor. The accelerometer containing three piezoelectric sensing elements oriented in three orthogonal linear axes measured kinetic force in each direction during a fall. The gyroscope which could measure angular velocity up to ±400^0/s was used to

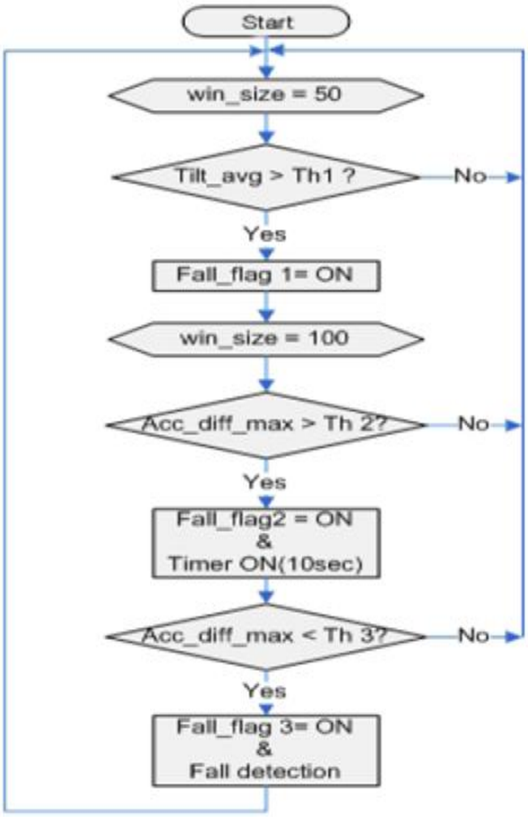
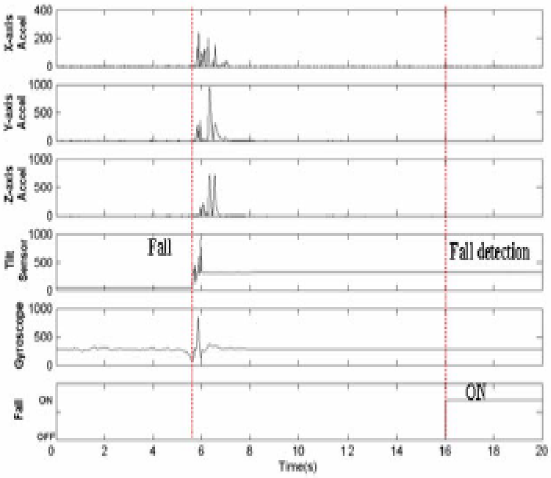


Figure 2.11 : Fall detection algorithm flow chart, as used in [14].

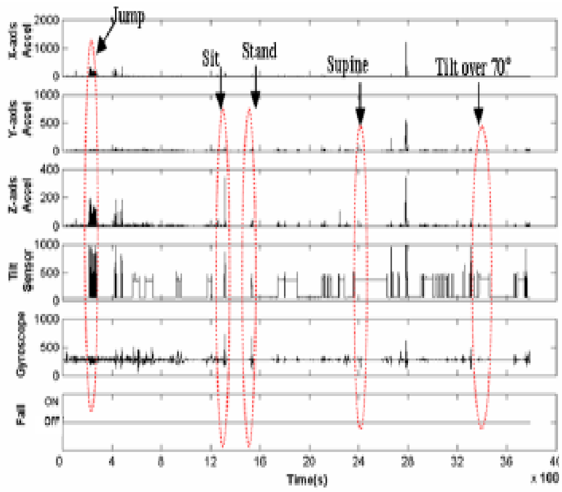
estimate posture transition while the tilt sensor detected whether the person is supine or standing. The system used an AVR4433 microcontroller integrated with the sensing part for real-time monitoring. The data obtained from the sensors were transmitted through Bluetooth to a data acquisition (DAQ) system. The algorithm used by them is summarized in Figure 2.11. If a person falls, acceleration rapidly changes, gyroscope gives various signals according to the fall pattern, and the tilt sensor is switched on as the body leans over 70^0. The tilt signal is then buffered and averaged in window 1 of size 50. Fall_flag1 becomes on if the average tilt value exceeds Threshold 1, initiating the inspection of acceleration variations of each axis. The acceleration difference values are then buffered in window 2 of size 100. If the maximum value among the buffered values crosses the Threshold 2, Fall_flag2 becomes on enabling the timer for 10 s. The values in window 2 are reconfirmed after 10 s. If the maximum is below Threshold 3, Fall_flag3 becomes on raising an alarm (because a person who falls does not move and there is no acceleration variation). This algorithm was integrated with the DAQ system. To test the accuracy, they attached the system to the subject's chest and experimented for five cases: (1) falling forward, (2) falling backward, (3) falling aside, (4) sitting and standing, and (5) daily life activity, repeating the first four cases ten times each. Table 2.2 summarizes the experimental result in which out of 123 times, success was achieved 119 times, thereby giving an accuracy of 96.7%. Figure 2.12 (a) shows the sensors' output during fall forward. Tilt sensor was switched on, acceleration signal exceeded the

Subject	Fall Detection									
	Forward		Backward		Side		Sit and Stand		Daily life	
	S	F	S	F	S	F	S	F	S	F
1	10	0	10	0	8	2	10	0	1	0
2	10	0	10	0	9	1	10	0	1	0
3	10	0	10	0	9	1	10	0	1	0
Total	30	0	30	0	26	4	30	0	3	0
	Success: 119, Fail: 4									

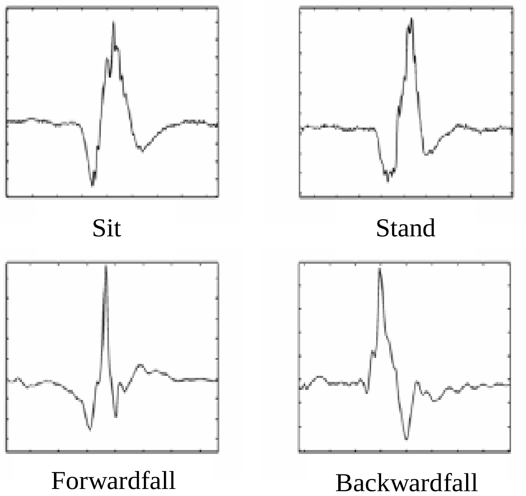
Table 2.2: Experiment result, as given in [14]



(a)



(b)



(c)

Figure 2.12: (a) Sensors' output during fall forward; (b) Signal pattern during daily life activity; (c) Signal from gyroscope for sit, stand, forward fall, and backward fall, as given in [14].

threshold, and gyroscope produced the signal pattern of forward fall thus detecting a forward fall. Figure 2.12 (b) shows the signal pattern during daily life activity. Fall detection signal remained off during this time. Figure 2.12 (c) gives the signal from gyroscope for sit, stand, forward fall, and backward fall. Forward fall and backward fall could be differentiated by the gyroscopic signal, but the pattern was same for sit and stand.

Qiang *et al*. [15] used two TEMPO (Technology-Enabled Medical Precision Observation) 3.0 sensor nodes which consisted of a tri-axial accelerometer and a tri-axial gyroscope as shown in Figure 2.13 (a). One of the sensor nodes was attached on the chest (Node A) and the other on thigh (Node B) as shown in Figure 2.13 (b). The tri-axial accelerometer, a "Freescale Semiconductor MMA7261QT"could monitor acceleration within a range of ±10g, while the tri-axial gyroscope consisted of an "InvenSense IDG-300" dual-

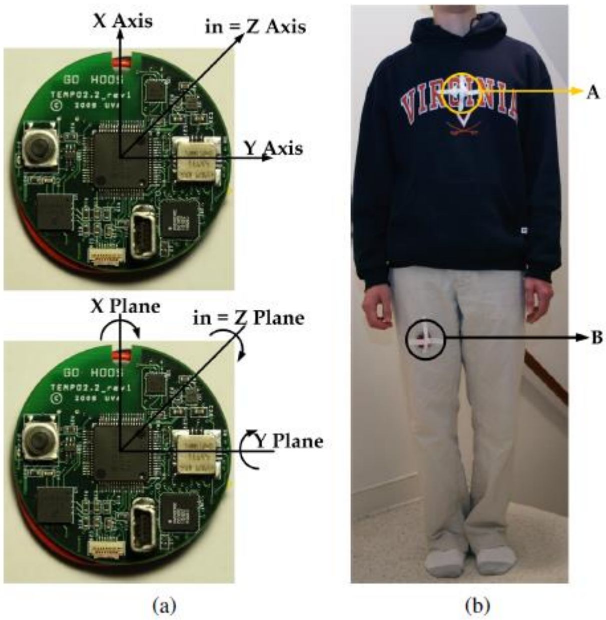


Figure 2.13: (a) The TEMPO 3.0 sensor node; (b) The placement of two TEMPO 3.0 nodes [15].

Algorithm 1: The three-phase fall detection process

▷ Monitor if people are static or dynamic during the present time segment.
if $|a_{A_{max}} - a_{A_{min}}| < 0.4g \wedge |a_{B_{max}} - a_{B_{min}}| < 0.4g \wedge |\omega_{A_{max}} - \omega_{A_{min}}| < 60°/s \wedge |\omega_{B_{max}} - \omega_{B_{min}}| < 60°/s$ **then**
    ▷ Recognize the present static posture: is it lying?
    if $\theta_A > 35° \wedge \theta_B > 35°$ **then**
        ▷ Determine if the transition before the present lying posture is intentional.
        if $a_{A_{max}} > T_{a_A} \wedge a_{B_{max}} > T_{a_B} \wedge \omega_{A_{max}} > T_{\omega_A} \wedge \omega_{B_{max}} > T_{\omega_B}$ **then**
            return *Yes*
return *No*

$$a_A = \sqrt{a_{A_x}^2 + a_{A_y}^2 + a_{A_z}^2}$$
$$a_B = \sqrt{a_{B_x}^2 + a_{B_y}^2 + a_{B_z}^2}$$
$$\omega_A = \sqrt{\omega_{A_x}^2 + \omega_{A_y}^2 + \omega_{A_z}^2}$$
$$\omega_B = \sqrt{\omega_{B_x}^2 + \omega_{B_y}^2 + \omega_{B_z}^2}$$

$$\theta_A = \arccos \frac{a_{A_x}}{g}$$

$$\theta_B = \arccos \frac{a_{B_x}}{g}$$

Figure 2.14: Fall detection algorithm, as used in [15].

axis gyroscope and an "Analog Devices ADXRS300" z-axis gyroscope which could monitor angular velocity between ±500^0/s and ±300^0/s respectively. The data from the sensors were fed to a "TI MSP430F1611" microcontroller. The sampling rate was fixed at 120 Hz as the characteristic response of human movement never exceeded 60 Hz. They collected four continuous data sets for 5 s for each of the activities of daily living (walk on stairs, walk, sit, jump, lay down, run, run on stairs), fall-like motions (quickly sit-down upright, quickly sit-down reclined), flat surface falls (fall forward, fall backward, fall right, fall left), inclined falls (fall on stairs) and static posture (standing, bending, sitting, and lying) to find out the accuracy of the proposed algorithm shown in Figure 2.14. They divided the

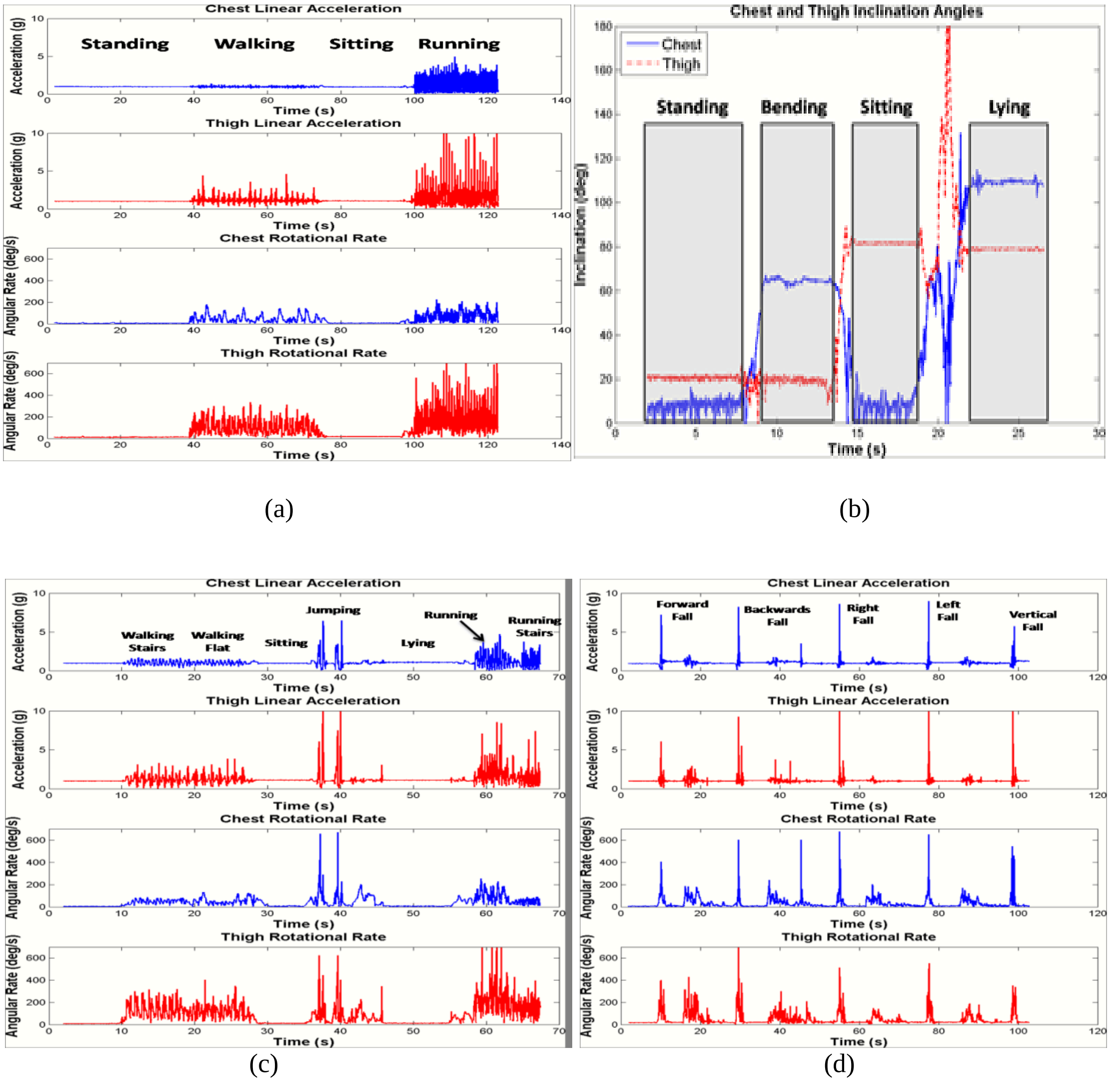


Figure 2.15: (a): Linear acceleration, and rotational rate of the trunk and thigh for standing, walking, sitting, and running, as given in [15]; (b) Inclination angles of the trunk and thigh for four static postures: standing, bending, sitting, and lying, as given in [15]; (c) Linear acceleration, and rotational rate of the trunk and thigh for ADL, as given in [15]; (d) Linear acceleration and rotational rate of the trunk and thigh falls, as given in [15].

algorithm into three steps: activity intensity analysis, posture analysis, and transition analysis. The data collected were segmented into 1 s intervals. According to the line 2 of Algorithm 1 shown in Figure 2.14, if the data from the accelerometers and gyroscopes were in the specified range, it was classified as static posture else a dynamic transition was assumed. Here α_A and α_B are the magnitudes of the chest and thigh vector magnitude linear acceleration vectors respectively, while ω_A and ω_B are measures of aggregate rotational rate. Figure 2.15 (a) shows the acceleration and rotational rate readings from nodes A and B for standing, walking, sitting, and running. If static posture got detected, the angle between the trunk and the gravitational vector, θ_A and the angle between the thigh and the gravitational vector, θ_B was calculated. For static posture α_A and α_B should always be near 1 g. Figure 2.15 (b) shows θ_A and θ_B for standing, bending, sitting, and lying. For bending posture, θ_A was greater than

35° and θ_B was less than 35°. For sitting posture, θ_A was less than 35° and θ_B was greater than 35°. Both θ_A and θ_B were less than 35° for standing posture, and greater than 35° for lying posture. If lying posture was detected, the third step determined whether it was an intentional or unintentional transition to the lying posture by applying thresholds to peak values of acceleration (α) and angular rate (ω) from the two nodes, as given in line 6 of Algorithm 1 given in Figure 2.14. Figure 2.15 (c) shows the linear acceleration of the chest and thigh for ADL while Figure 2.15 (d) illustrates the acceleration and rotational rate of different types of fall. They evaluated the accuracy of the system by first studying two special cases, and then running a continuous monitoring system. Unlike other systems [16] [17] [18], this system was able to distinguish between a fall and sitting down fast. The accelerometer reading crossed the threshold at both the instances; therefore sitting down fast was not distinguishable for systems using only accelerometers [16]. Figure 2.16 (a) and Figure 2.16 (b) shows the inclination angles, linear acceleration, and rotational rate of the trunk and thigh for sitting fast, with ending postures as sitting straight and leaning back. In the second case the system was able to differentiate between bending and falling on stairs. The other systems which used trunk position only [17] [18] to detect falls failed to differentiate between the two because the trunk position is inclined at both the instances. On the other hand this system gets the inclination data of trunk as well as thigh which gives an accurate result. Figure 2.16 (c) and Figure 2.16 (d) shows the linear acceleration rotational rate and inclination angles of the trunk and thigh for falling on stairs. Continuous monitoring was done for daily life activities, fall-like motions (e.g. sitting down fast, jumping, going up/downstairs, stumbling, and lying down) and different kinds of fall (e.g. falling forward/backward/leftward/rightward/vertically). Figure 2.16 (e) shows the false negative performance of the algorithm and Figure 2.16 (f) shows the false positive performance of the algorithm. Each type of fall was detected accurately except for the fall against wall ending up with sitting position. The sensitivity of algorithm was observed to be 91% from 70 records. Most of the fall-like activities got excluded. However, quickly lying sometimes triggered false positives. The specificity was observed to be 92% from 70 records.

2.5 Summary

From the review of the literature it was found, that the use of multiple sensors enabled the system to recognize more number of activities with better accuracy. Only accelerometers or only gyroscopes provided good results for restricted movement in specific direction. Systems which used multiple sensors faced problems in detecting movements like fall against wall ending up in sitting position, fall aside, and quickly lying. It is also preferable to have sensors located at different positions of the body to measure the relative movement of various body parts. This also cancels the error introduced by the acceleration due to gravity. Head, waist,

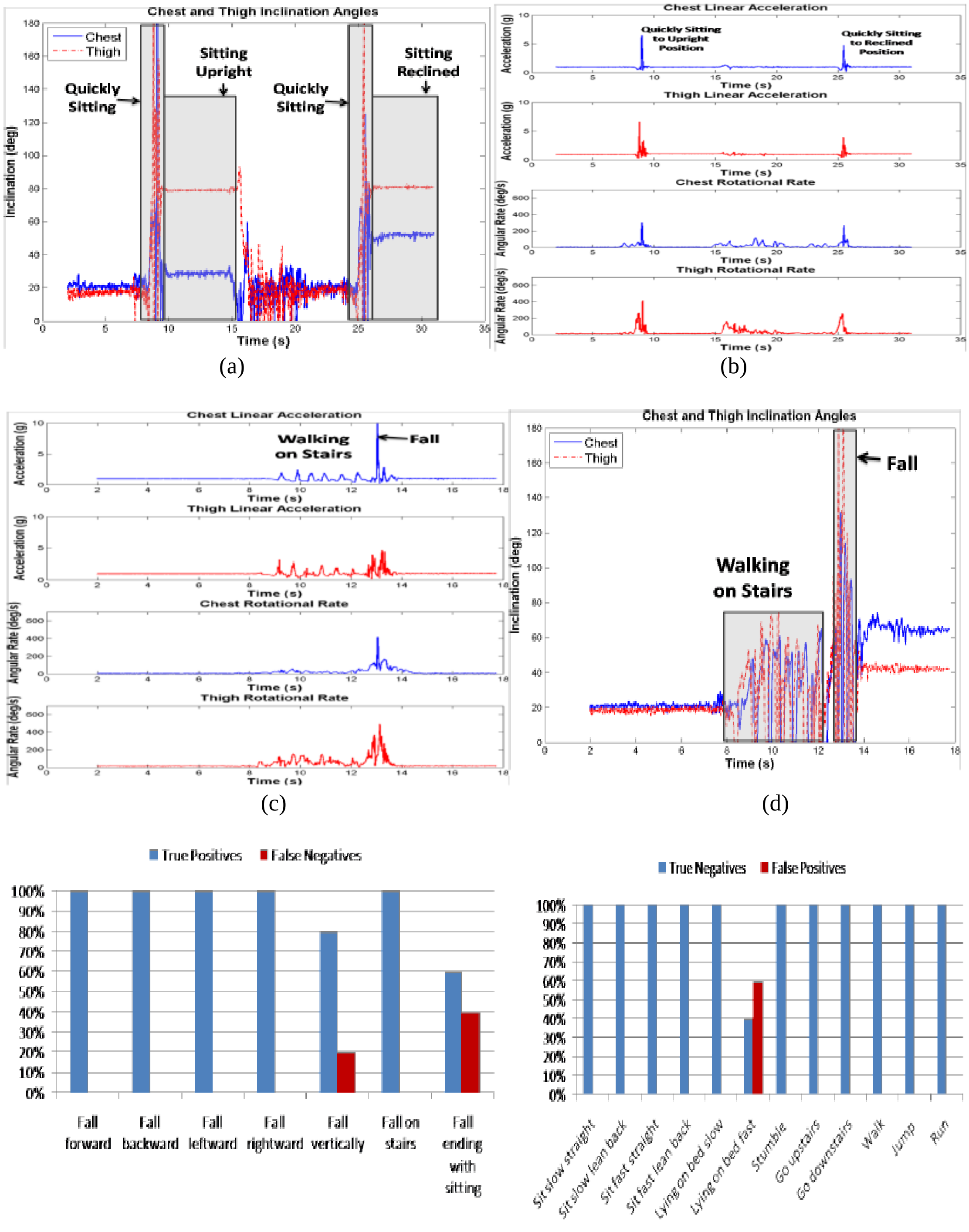


Figure 2.16: Results of [15]: (a) Chest and thigh inclination angles for sitting fast, ending postures are sitting straight and leaning back; (b) Linear acceleration and rotational rate of the trunk and thigh for sitting fast, ending postures are sitting straight and leaning back; (c) Linear acceleration and rotational rate of the trunk and thigh for falling on stairs; (d) Linear acceleration and rotational rate of the trunk and thigh for falling on stairs;(e) False negatives performance; (f) False positives performance.

trunk, and thigh proved to be good locations for sensor placement while wrist did not. Placing sensors on head can be too uncomfortable for the user and hence waist, trunk and thigh are the most suitable places for sensor location.

Fuzzy inference system proved to be a powerful algorithm as it enhanced the accuracy of the system using a single accelerometer but the algorithm itself is too complex and may not be suited for real-time applications. Threshold based fall detection algorithms are less complex, easy to implement, and require less amount of processing power yet achieving 100% sensitivity and specificity in most cases.

Chapter 3

HARDWARE OF INERTIAL SENSING MODULE

3.1 Introduction

The inertial sensing module (ISM) is designed with the primary objective of continuously acquiring the acceleration and angular velocity data for tracking the movement and orientation of a body part to which the unit is attached. It should have a settable sampling frequency of 100 Hz or higher and should be able to wirelessly transfer the data to a central unit for integration and processing of the data from multiple ISMs. It may incorporate processing to detect emergency situations and send an alarm to the base station. Further it should have nonvolatile memory to store the acquired data in a time-stamped manner which can be transferred in a burst mode. This storage feature is needed for actigraphy and other research applications. As the unit is to be used as a body-worn device, it has to be light in weight, compact in size, and battery powered with low power consumption. With these objectives, the circuit has been designed using a microcontroller and a MEMS sensor with integrated tri-axial accelerometer and tri-axial gyroscope. A block diagram of the module is shown in Figure 3.1. It has a serially interfaced flash memory for storing the acquired data. To keep the unit compact, it is designed without any connectors and switches. A serially interfaced Bluetooth module is provided for wireless control of operations and also for transfer of data, either sample-by-sample or in burst mode.

3.2 Component selection

The module is designed with "InvenSense MPU-6000" [2] as the integrated MEMS based tri-axial accelerometer and tri-axial gyroscope sensor module. It has on-chip A/D converter and processor for simultaneous conversion of the six analog sensed signals as 16-bit digital outputs. It has SPI and I2C ports for controlling its operation and for transferring the sensed data. It has an internal clock for setting the sampling rate and internal processing. The sensor module is available in QFN package and operates with 2.375 – 3.46 V supply with a nominal operating current of 3.9 mA.

The microcontroller used is "Microchip Technology PIC24FJ64GB004" [1], available in 44-pin TQFP package and with 25 I/O pins with programmable functions. It is a 16-bit

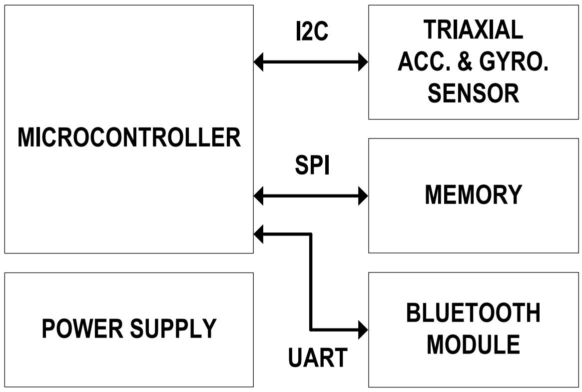


Figure 3.1: Block diagram of the Inertial Sensing Module.

controller with 64K program memory and 8K data RAM. It operates with 2.0 – 3.6 V supply, with current drain of 2.9 mA at 3.3 V and processor clock of 8 MHz. It has two SPI, two I2C, two UART, and one USB channel ports. There are five different modes for clock which can be selected under program control. The processor starts at power-on with an internal RC oscillator with clock F_{RC} of 8 MHz. A "postscalar" can be used for dividing F_{RC} by a factor of 2 to 256 to obtain the processor clock F_{OSC}. The internal oscillator has an accuracy of 0.25%. The sampling frequency of the sensor output is derived from its internal clock and is not affected by the accuracy of the processor clock. Hence the module is designed without using an external clock for the microcontroller The internal instruction cycle clock F_{CY} is half of the processor clock F_{OSC}, which can be changed through software by selecting the post-scaling factor. To conserve power, it should be set at the minimum clock frequency required for processing the signals.

The module has been designed with the flash memory "Microchip Technology SST25VF064C" [3] in 8-pin SOIC package. It is a 64 Mb (8 Mbytes) serial I/O flash, with SPI and an input buffer of 256 bytes. The data are written into the input buffer using SPI, and are saved into the flash memory by sending the starting address and a page program instruction. The page write takes about 2.5 ms. For sample-by-sample signal acquisition and storage with a sampling frequency of 100 Hz, the values of six sensor channels result in 12 bytes every 10 ms. These are appended with two bytes as time index and written to the input buffer of the flash memory using SPI. With F_{CY} = 4 MHz, the operation of transferring 14 bytes of data to the buffer and page write operation together are estimated to take about 1.64 ms. Therefore, the memory write speed is adequate for storing the input data even on sample-by-sample basis. The operating voltage is 2.7 – 3.6 V with supply current of 12 mA during read operation and 25 mA during write/erase operation, and 5 µA in standby mode.

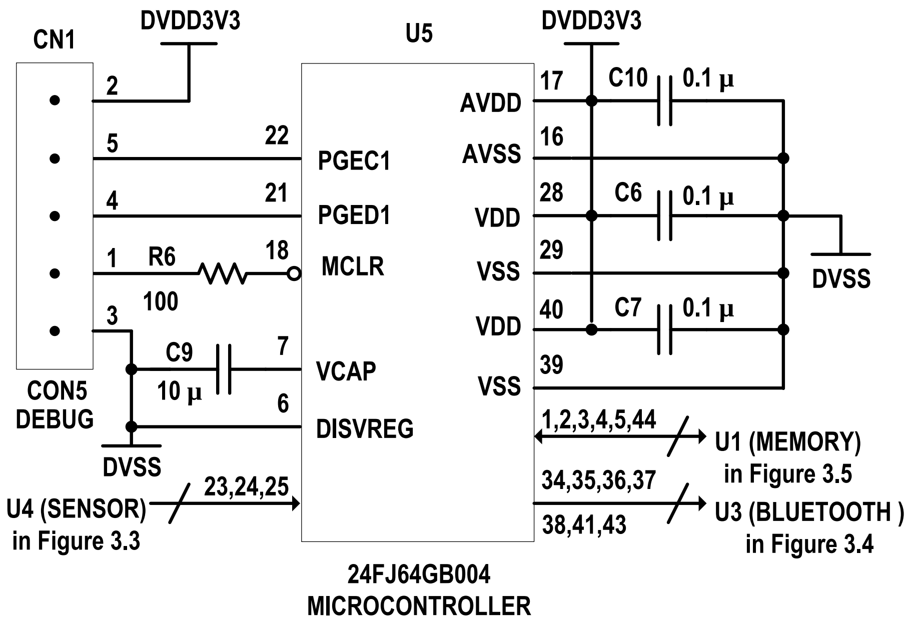


Figure 3.2: The microcontroller pin connections.

For short-range wireless data transfer, Zigbee and Bluetooth are the two main options. We have opted for Bluetooth as it is widely used in hand-held devices. The Bluetooth module "Roving Network RN-42" [4] is used for wireless communication. It is a “class-2” radio with a 20 m range. For wireless transfer, it has UART with serial port protocol (SPP), supporting data rate up to 240 Kbps in slave mode and 300 Kbps in master mode. For microcontroller to Bluetooth data transfer, the rate is settable between 1200 bps to 921 kbps. The operating voltage of the module is 3.0 – 3.6 V with supply current of 26 µA in sleep mode, 3 mA when connected, and 30 mA during wireless data transfer.

3.3 Microcontroller

The connections to the microcontroller U5 are shown in Figure 3.2. The power and inline programming connections are made as per the information in its datasheet. As the analog blocks of the microcontroller are not used, the AVDD and AVSS pins are shorted to DVDD and DVSS, respectively. Capacitors C6, C7, C10 (0.1 µF, ceramic) are decoupling capacitors, each placed across VDD-VSS, within a distance not more than 6 mm from the pins. The DISVREG pin is connected to ground to enable the internal voltage regulator. To stabilize the output voltage of the on-chip voltage regulator, capacitor C9 (10 µF) is connected between VCAP and DISVREG pins at a distance less than 6 mm from the pins. The connector CN1 is a 5-pin connector provided for inline programming and debugging. PGEC/PGED pins are used for in-circuit serial programming and debugging. The MCLR pin is used for device programming and debugging as well as device reset. A resistor R6 (100 Ω) is connected to MCLR pin to limit the current entering the pin, and to meet V_{IH} and V_{IL} specifications at that

Table 3.1: Assignment of port pins of the microcontroller (U5).

Pin No.	Pin Label	Function
1	SCK1	Serial clock of SPI connected to serial clock of flash memory U1
2	SDO1	Serial data out of SPI connected to serial data in of flash memory U1
3	RP23	Digital output connected to write protect of U1 flash memory U1
4	SDI1	Serial data in of SPI connected to serial data out of flash memory U1
5	RP25	Digital output connected to chip enable of flash memory U1
23	SDA2	Serial data of I2C connected to serial data of sensor U4
24	SCL2	Serial clock of I2C connected to serial clock of sensor U4
25	INT	Digital input connected to interrupt of sensor U4
34	RA4	Digital output connected to reset of Bluetooth module U3
35	RA9	Digital output connected to PIO6 of Bluetooth module U3
36	RP19	UART Rx connected to 5–pin connector CN2
37	RP20	UART Tx connected to 5–pin connector CN2
38	RC5	Digital output connected to PIO4 of Bluetooth module U3
41	RB5	Digital output connected to PIO2 of Bluetooth module U3
43	RB7	Digital output connected to PIO3 of Bluetooth module U3
44	RP8	Digital output connected to HOLD of flash memory U1

pin. Pins 23, 24, 25 (SDA2, SCL2, INT) are used for interfacing the sensor U4 as described in Section 3.4 and Figure 3.3. Flash memory U1 is interfaced to pins 1, 2, 3, 4, 5, 44 (SCK1, SDO1, RP23, SDI1, RP25, RP8) as described in Section 3.6 and Figure 3.5. The Bluetooth module is connected to the pins 34, 35, 36, 37, 38, 41, 43 (RA4, RA9, RP19, RP20, RC5, RB5, RB7) as shown in Figure 3.4 and described in Section 3.5. The assignment of microcontroller port pins used for interfacing sensor, flash memory, and Bluetooth module is given in Table 3.1.

3.4 Sensor

The connections to the sensor U4 (MPU-6000) and its interfacing to the microcontroller U5 are shown in Figure 3.3. The sensor chip has options of I2C and SPI for serial communication. I2C needs 2 lines (SCK, SDA) for communication, while SPI needs 4 lines (SCK, MOSI, MISO, CS). SPI does not have a built-in mechanism for acknowledgement and addressing. Although, I2C has a lower data rate than SPI, its data rate 100 kHz is sufficient for transferring the data for sampling rate of up to 1 kHz as needed in our application. Therefore I2C has been used for serial communication between the sensor and the controller.

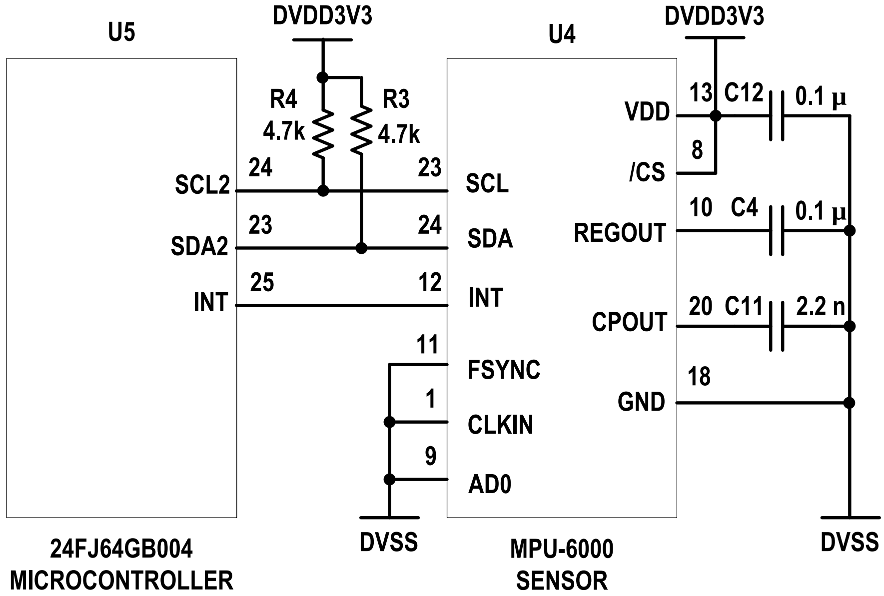


Figure 3.3: Sensor interfacing.

The SDA and SCL pins for I2C communication are open drain and therefore pull-up resistors R3 and R4 of 4.7 kΩ are provided. Capacitor C11 (2.2 nF, ceramic) is used to bypass the output of an on-chip charge pump which generates the high voltage for internal MEMS oscillator. Capacitor C4 (0.1 µF) is connected to REGOUT as regulator filter capacitor. The sampling rate of the on-chip ADC can be selected from 3.9 Hz to 8 kHz. The gyroscope has settable full-scale range of ±250°/s, ±500°/s, ±1000°/s, and ±2000°/s. The accelerometer has settable full-scale range of ±2 g, ± 4 g, ±8 g, and ±16 g. The full-scale range setting is common for the three axes. An auxiliary I2C port is available for interfacing with an external digital sensor such as a magnetometer. The sensor module has an on-chip digital motion processor (DMP) which can be used for reducing the computation requirement of the host processor. It provides three types of motion detection: free-fall detection, zero-motion detection, and threshold-based motion detection. The threshold and duration for each type of detection can be set using the corresponding registers. There is an interrupt associated with each of them. When the value remains above the specified threshold for the specified duration, an interrupt is raised. Option of motion detection using DMP is under software control. At each sampling instant, the 16-bit values of the six internal channels and the three external channels acquired through the auxiliary I2C port are combined as a single 144-bit word and stored as 18-byte data. If the auxiliary port is not used, the values of the six internal channels are stored as 12-byte data. In our design, the auxiliary port of the sensor module is not used. Therefore, sample-by-sample mode involves interrupt based 12-byte data transfer after each sampling. The sensor module has an on-chip 1024-byte FIFO buffer. As the buffer gets filled an interrupt is generated after which the data can be read in burst mode. This mode

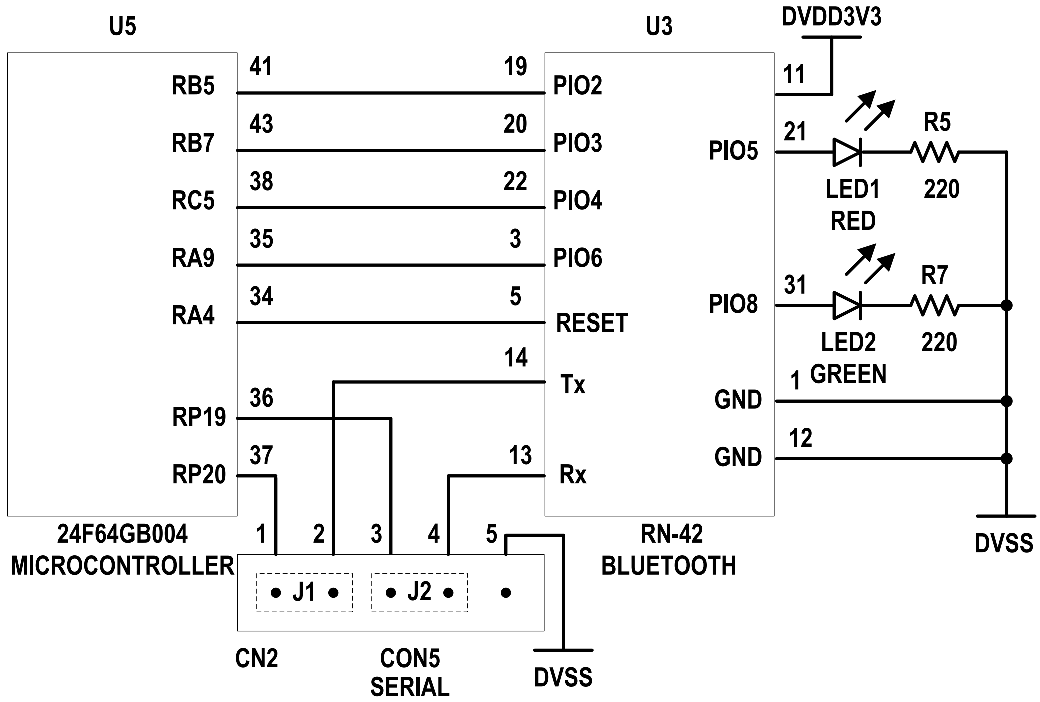


Figure 3.4: Serial communication and Bluetooth wireless interface.

permits a larger fraction of time on the microcontroller for processing of the acquired data. The FIFO buffer has a counter associated with it for tracking the number of bytes in the buffer. The counter value can be read at any time for carrying out polled data transfer from the buffer.

3.5 Bluetooth module

The data are transferred from microcontroller to the Bluetooth module serially using UART. The connection of Bluetooth module U3 with microcontroller U5 is shown in Figure 3.4. Rx and Tx pins of Bluetooth module and RP19 and RP20 pins of the microcontroller are connected to a 5-pin connector CN2. This 5-pin connector makes serial data pins of the microcontroller available for debugging purpose. For Bluetooth connection, jumpers J1 and J2 are used to short RP19 to Rx and RP20 to Tx, respectively. The module has a programmable pin PIO7 to set the baud rate at power-on after which the baud rate can be set to a desired value by programming. If the pin is kept at logic high the baud rate is 9600 bps at power-on which cannot be changed later by programming and if the pin is given logic low or is not connected the baud rate is 115 kbps at power-on which can be changed later by programming. In our design PIO7 is not connected and therefore the power-on baud rate is 115 kbps. Baud rate of 115 kbps is sufficient for our design and therefore it is not changed by programming. There are two LEDs to indicate the status of the module. The blinking rate of LED1 (red) indicates the connection status of Bluetooth. It blinks twice per

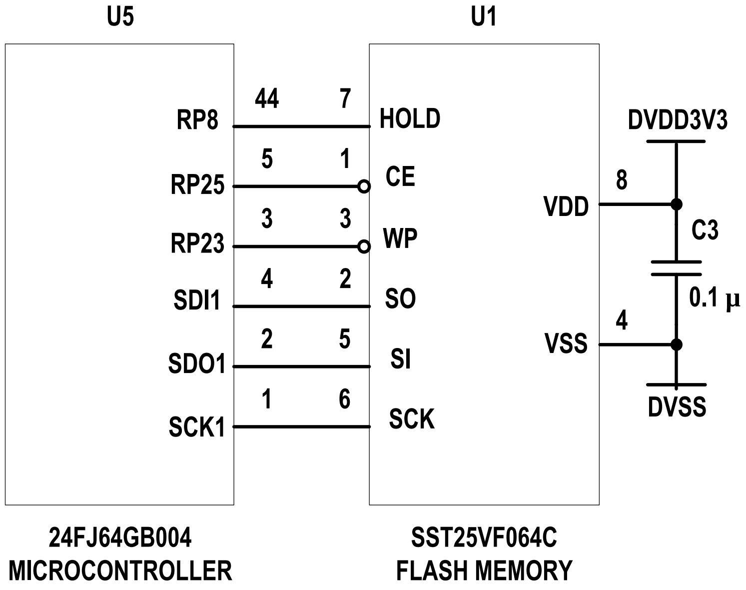


Figure: 3.5 Memory interfacing.

second at boot up. Blinking at a rate of 10 times per second indicates that the module is in the configuration mode. Blinking once per second indicates that the device is idle and discoverable. If it is continuously lit, it indicates that the connection is established. LED2 (green) blinks continuously during data transfer and remains continuously lit otherwise. All the settings of Bluetooth like address, name, baud rate, parity, mode (master, slave), pin code, remote address, etc. can be set in the configuration mode. This mode is entered by sending '$$$' to the module through serial port. Pin 14 and 13 are used for serial transmission and reception respectively. Programmable pin PIO6 is used to set the module in master mode, PIO4 is used to set factory defaults, PIO3 is used to enter auto discovery mode, and PIO2 is used to convey the connection status to the microcontroller.

3.6 Flash memory

Serial Peripheral Interface (SPI) protocol has been used for connecting memory U1 to the microcontroller U5 as shown in Figure 3.5. The memory chip "Microchip Technology SST25VF064C" is a 64 Mb (8 Mbytes) serial I/O flash. Each sensor axis data is of 2 bytes and hence 6-axis sensor data (tri-axial accelerometer and tri-axial gyroscope) is 12 bytes. Therefore in total 699050 samples (8 Mbytes / 12 bytes) can be stored in the memory. At a sampling frequency of 100 Hz during fall detection, the memory can record data for 6990.5 s (699050 / 100 Hz) which corresponds to 1 hour 57 minutes. In applications like actigraphy where sampling frequency of 10 Hz is sufficient, the memory can store data for 19 hours 24 minutes. The chip has a fast read dual I/O mode in which two pins are assigned for reading data simultaneously, pin SO for odd data bits and pin SI for even data bits. This mode has not

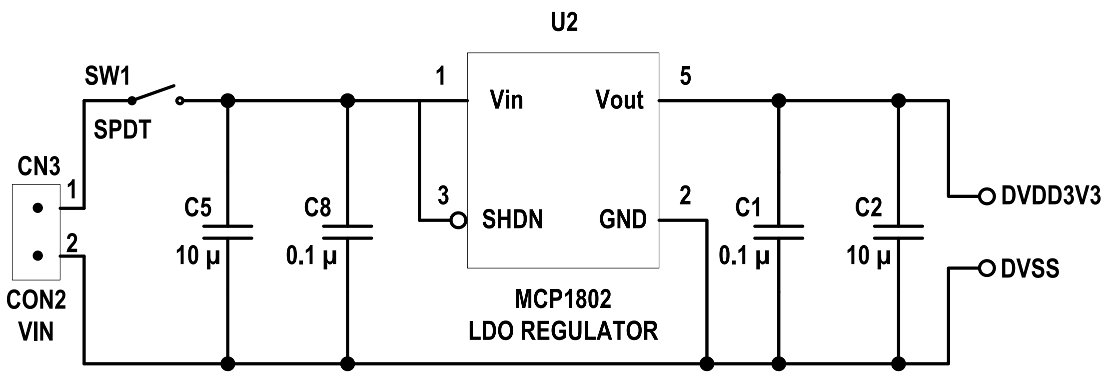
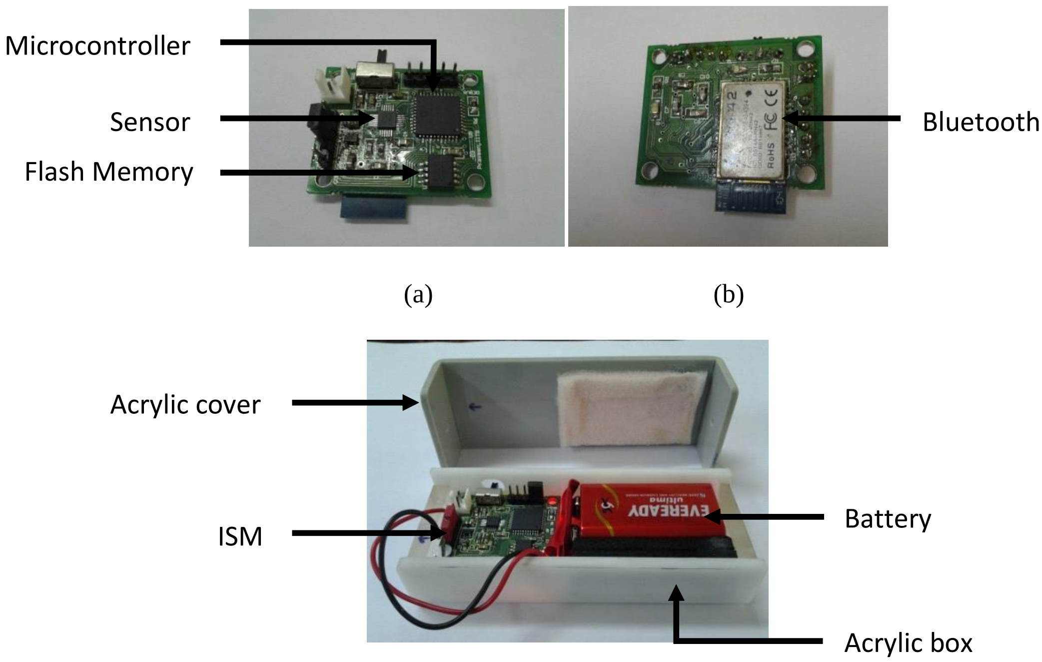


Figure 3.6: Power supply.



(a) (b)

(c)

Figure 3.7: (a) Top view of ISM; (b) Bottom view of ISM; (c).Packaged ISM

been used in our design as it would increase software overheads. Capacitor C3 (0.1µF) has been used as decoupling capacitor between the VDD-VSS pins. WP Pin is provided for hardware control of write protection while HOLD pin is used to pause a serial sequence. HOLD pin also provides a power-on reset to the memory and can be used as a reset pin by clearing the EHLD (Enable Hold) bit of the memory.

3.7 Power supply

All ICs used in the unit can be powered by 3.3 V. The maximum current consumption of all components on the board is estimated to be $I_L \approx 50$ mA (microcontroller U5: 2.9 mA, sensor U4: 3.9 mA, memory U1: 12 mA, Bluetooth module U3: 30 mA). Thus we need a 3.3 V regulator with 50 mA output current. Linear regulator "Microchip MCP 1802" [5] has been used as the regulator U2 with output voltage V_{OUT} = 3.3 V, as shown in Figure 3.6. It is a low drop-out voltage regulator with 200 mV drop at 100 mA. It has 3.5 – 12 V input, with maximum current I_{MAX} = 300 mA and maximum continuous power dissipation $P_D \approx 250$ mW. We need to have

$$I_L < I_{MAX}\ ,((V_{IN})_{MAX} - V_{OUT})\ I_L < P_D$$

For I_L = 50 mA, we get $(V_{IN})_{MAX} < P_D/I_L + V_{OUT} = 8.3$ V. Therefore, the regulator can be operated with an input voltage of 3.5 – 8.3 V for a load current of 50 mA. Actual maximum current consumption of the circuit will be lower than 50 mA, as the memory read and wireless data transfer from the Bluetooth module will not be carried out simultaneously. To improve the stability of the regulator, a parallel combination of capacitors C5 (10 µF), and C8 (0.1 µF, ceramic) is connected at the input. A combination of parallel capacitors C1 (0.1 µF, ceramic) and C2 (10 µF, ceramic) is connected at the output. The regulator IC has been used with SOT-23-5 package.

3.8 PCB design and ISM packaging

A 2-layer 36.32 mm x 29.21 mm PCB has been designed for the circuit. To minimize the board area, both sides of the board have been used for placing components. Figures 3.7(a) and 3.7(b) show the top and bottom view of the assembled board respectively. The ISM board, the Bluetooth module along with the battery are housed in a custom made acrylic box of 10.5 cm x 4 cm x 2.3 cm, as shown in Figure 3.7(c).

Chapter 4

SIGNAL ACQUISITION AND TESTING

4.1 Introduction

The development of the software was carried out in a stage-wise manner. Initially, a program "Oscillator" was written to test the oscillator and debugging connections. Bluetooth module was interfaced next and a program "Bluetooth" was written to test the Bluetooth connectivity and bi-directional data transfer. Flash memory was then interfaced to facilitate data storage and a program "Memory" was written to test the same. The sensor was interfaced last and was tested using software "Sensor" by reading its device ID and acquiring linear acceleration and angular velocity data. These programs developed for testing the blocks of the module are described in Section 4.2.

Three different programs "Sample-by-sample", "Burst mode interrupt", and "Burst mode polling" are written to acquire data in sample-by-sample mode and burst modes. The three programs are identical in all other respects except for the UART ISR to start DAQ. In "Sample-by-sample" program the UART ISR for start DAQ operation is written to acquire data in sample-by-sample mode while in the other two programs it is written to acquire data in burst mode. In "Burst mode interrupt" program the data are acquired in bursts using interrupt while in "Burst mode polling" program the data are acquired in bursts by polling method. After power-on, the main part of the programs "Sample-by-sample", "Burst mode interrupt", and "Burst mode polling" starts with configuring the internal RC oscillator. It then initializes the I/O port pins, serial communication modules (SPI, I^2C, and UART), UART and external interrupts. Sensor IC and flash memory are then configured after which the microcontroller is placed in sleep mode to conserve power consumption. It remains in sleep mode until it is interrupted by UART interrupt generated by the Bluetooth module in response to an external command. The Bluetooth module is used to control ISM operations and also for data transfer from ISM. Specific operations are handled by corresponding interrupt service routines (ISR's). The different UART ISRs written for the three programs are described in Sections 4.3.3, 4.3.4, and 4.3.5.

A graphical user interface (GUI) is developed to control the operations of the ISM through a PC. Controls are provided in the GUI to save the data and load the previously saved data. The accelerometer hex data are converted into gravitational unit (g) and the gyroscope data are converted into angular rate data (°/s) and these are displayed as table provided on the GUI screen. A plot of the sensor data can be used to test its validity. Data conversion algorithm is described in Section 4.4 and the sensor data testing is presented in Section 4.5.

4.2 Programs for testing the hardware blocks

The following microcontroller programs have been written for testing the hardware blocks.

- `Oscillator`: It was used to test the basic connections of the controller. It generated a square wave of frequency 8 MHz by using the internal fast RC oscillator at the oscillator out pin OSCO of the microcontroller.
- `Bluetooth`: To test the Bluetooth module, wireless connection was established between the Bluetooth of ISM and a Bluetooth module RN-42 connected to a desktop PC through its serial port. Bi-directional data transfer for Bluetooth connectivity was verified by writing a program to echo the data sent to ISM through Bluetooth. The baud rate was set at 115 kbps. As a character was received at UART-Rx register from Bluetooth, an interrupt was generated. The ISR for this interrupt read the character from UART-Rx register and wrote it to UART-Tx register for transmitting through Bluetooth.
- `Memory`: This program was written to test the memory connections. It receives input sent from the computer over Bluetooth and saves it in the memory. After receiving an "enter", all previous bytes of data are retrieved from the memory and sent sequentially to the computer. The baud rate was set at 115 kbps and the SPI clock was set at 4 MHz. A UART interrupt was generated each time a character was received. The UART ISR compared the character with "enter". If the character received was not "enter", it was written into memory using SPI protocol and the memory pointer was incremented by 1. If the character received was "enter", the memory was read from the first location to the current location and the data were written to the UART-Tx register for transmitting through Bluetooth. It verified proper working of the memory and correct implementation of the SPI protocol.
- `Sensor`: This program was written to read the device ID of the sensor using I2C protocol. To test the internal DAQ hardware of the sensor, the six 16-bit registers which store the value of the individual axis of tri-axial accelerometer and tri-axial gyroscope are read. The output rate and sampling rate of both the gyroscope and accelerometer are fixed at 100 Hz. The baud rate was set at 115 kbps, I2C clock was set at 1 MHz and the sensor interrupt was set in sample-by-sample mode. Each time the registers got filled with data, a

UART interrupt was generated. To service this interrupt, the microcontroller read the data from all the six registers using I2C protocol and transmitted them through UART.

4.3 Programs for signal acquisition

Three different programs were written for signal acquisition. The three programs were identical in all respect except for the UART ISR for start DAQ operation. Each program started with configuring the internal modules of the microcontroller and other components of ISM as described in Section 4.3.1. After configuring, the microcontroller is put to sleep mode from where it can come out only by a UART interrupt or a sensor interrupt. UART interrupt is used to control the ISM operation (start DAQ, stop DAQ, retrieve data, erase memory, and retrieve last *N* bytes of data) as described in Section 4.3.2, while sensor interrupt is used only when a UART IRQ is raised to start DAQ. Therefore whenever a UART IRQ is raised to start DAQ the sensor interrupt is enabled and whenever a UART IRQ is raised to stop DAQ the sensor interrupt is disabled. The UART ISR for start DAQ operation for the programs "Sample-by-sample", "Burst mode interrupt", and "Burst mode polling" are described in Section 4.3.3, Section 4.3.4, Section 4.3.5 respectively.

4.3.1 Initial configuration setting of the programs

The flowchart for configuring the programs is shown in Figure 4.1. The system clock F_{OSC} is set at 8 MHz and therefore internal instruction cycle clock F_{CY} is fixed at 4 MHz ($F_{CY} = F_{OSC}/2$). The microcontroller I/O port pins are initialized as input/output depending on their function. UART1, SPI1 and I2C2 modules of the microcontroller are enabled. The baud rate for UART1 is set to 115K and the receive interrupt has been enabled with highest priority. The SPI1 clock is same as the internal instruction clock F_{CY} which is 4 MHz while the I2C2 clock is set at 100 kHz.

The sensor is reset initially and the full-scale range of accelerometer is set to ±4 g while that of gyroscope is set to ±500°/s. The FIFO is enabled as the data are transferred in sample-by-sample mode as well as in burst mode. The DMP (Digital Motion Processor) is disabled as neither any auxiliary sensor is used nor the internal motion detection algorithm is used. The internal high-pass filter and low-pass filter are also disabled as all the calibration and processing is to be done during signal processing. The sampling frequency of both accelerometer and gyroscope is set at 100 Hz.

The flash memory is configured to be read in the normal mode as fast mode is not required and will consume more current. The BPL (Block Protection Lock) bit is set to allow hardware control over status register write protection using WP (Write Protection) pin.

The UART interrupt is enabled with a priority greater than the sensor interrupt. The microcontroller is then put to sleep mode to save power and can come out of this mode on

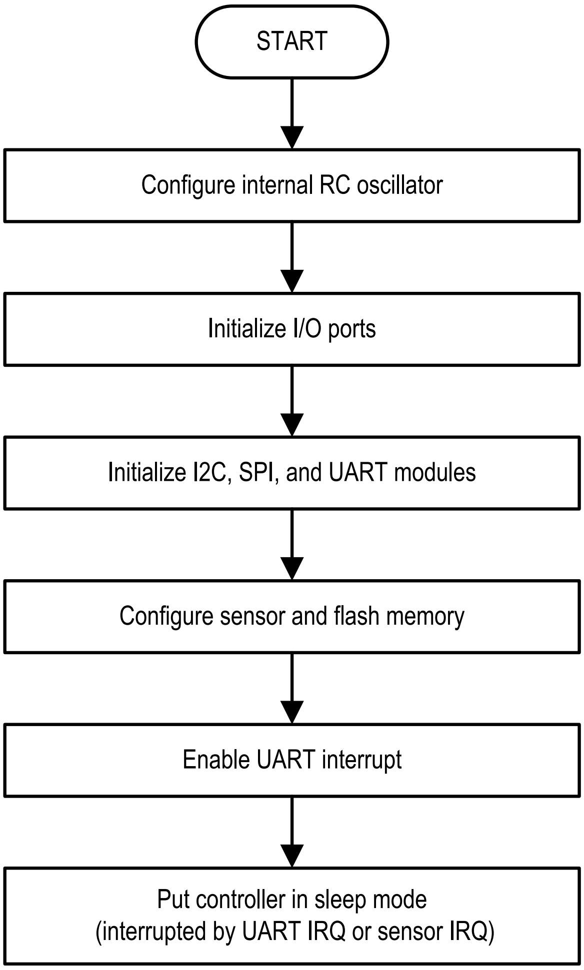


Figure 4.1 Flowchart for initial configuration setting of programs.

receiving a UART interrupt or a sensor interrupt to service the respective ISR. On receiving a UART interrupt for data acquisition the sensor interrupt will be enabled and data will be transferred from sensor to the microcontroller by servicing the sensor ISR. To stop data acquisition a UART interrupt is raised which having the highest priority will disable sensor interrupt, stop data acquisition and the microcontroller will return to sleep mode.

4.3.2 UART interrupt logic to control ISM operation

The control and data transfer operations are handled through the Bluetooth module which is connected to UART of the microcontroller. Single character codes are assigned for these functions. On receiving a character UART interrupt is raised which gets the microcontroller out of sleep mode to service the respective ISR. A flowchart for the operations is shown in Figure 4.2. Data acquisition is started by sending ‘p’. To stop data acquisition, ‘s’ is sent. The stored data is retrieved by sending ‘r’, while 'n' is to be sent to retrieve last *N* bytes. The flash memory of ISM can be erased by sending ‘e’. On receiving ‘p’ different ISR is executed in the three programs which is described in the next three subsections, while on receiving other characters the ISR executed is same for all three programs.

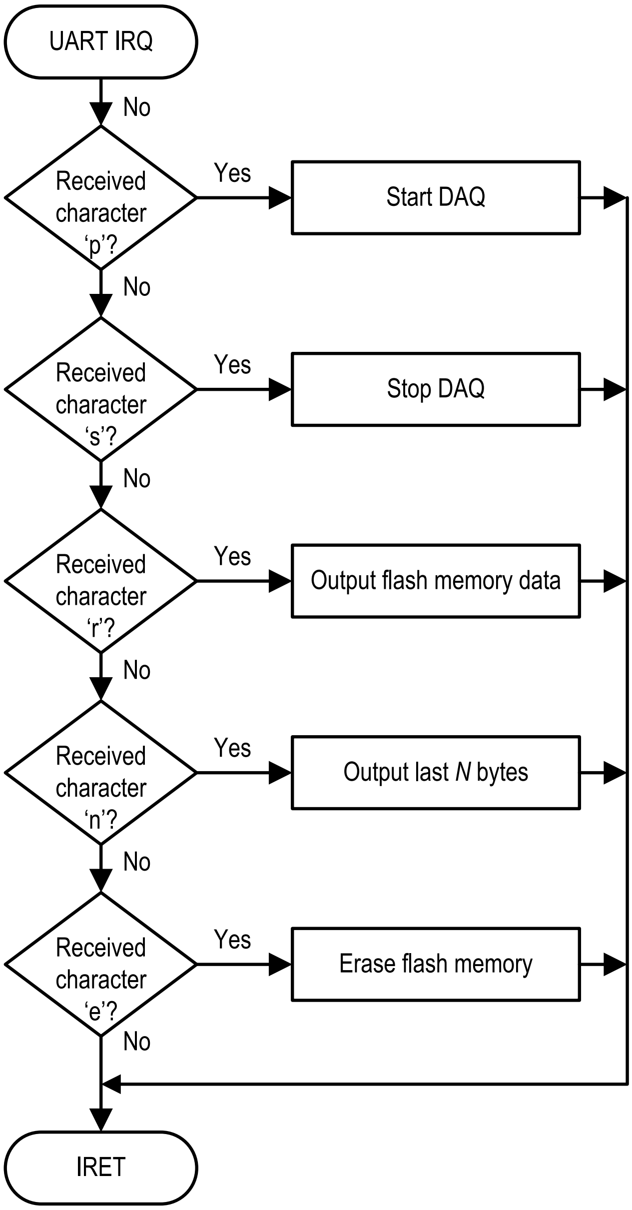


Figure 4.2 Flowchart for UART interrupt logic to control ISM operation.

4.3.3 UART ISR for data acquisition in "Sample-by-sample" program

Initially a program "Sample-by-sample" was written to acquire signal data in sample-by-sample mode. As character 'p' is received on the UART-Rx register, the sensor interrupt is enabled and set to data ready enable mode after which the microcontroller is again put to sleep mode. Since the sensor interrupt is enabled now in data ready mode, every time the sensor axes are filled with data an interrupt will be raised and the sensor ISR will be executed. The flow chart for sensor ISR for "Sample-by-sample" program is shown in Figure 4.3. Whenever the sensor interrupt is raised, the 6-axis sensor data are read from the sensor and stored in a buffer of size 252 bytes (21 samples) in the data RAM of the microcontroller. This buffer is created to facilitate page program operation of the memory by which 256 bytes can be written into the flash memory at a time. The buffer pointer is then incremented by 12, the

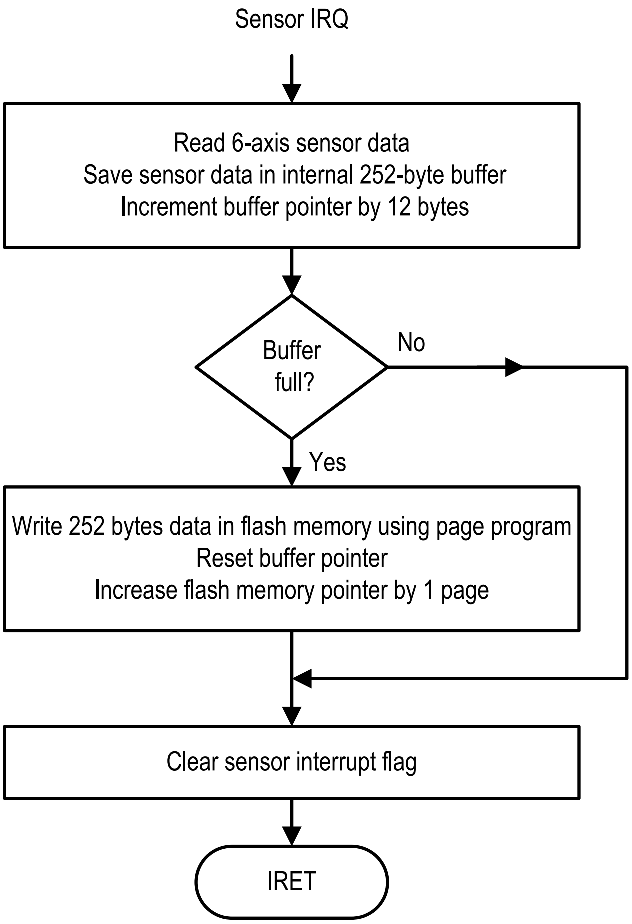


Figure 4.3: Sensor ISR in "Sample-by-sample" program.

external interrupt flag is cleared and the microcontroller is put to sleep mode if the buffer is not full. If the buffer is full with sensor data then the 252 bytes (21 samples) are written into the flash memory using page program instruction and the memory pointer is incremented by 1 page (256 bytes). The internal buffer and buffer pointer both are reset, the interrupt flag is cleared and the microcontroller is put to sleep mode until another sensor IRQ is raised. The microcontroller can come out of this continuous acquisition process only by a UART IRQ to stop data acquisition. For 100 Hz sampling frequency, an interrupt is generated every 10 ms and therefore not much time is available for signal processing. To avoid the overheads associated with sample-by-sample data transfer, burst mode data transfer can be used. Two such modes applied in "Burst mode interrupt" program, and "Burst mode polling" program are described in Section 4.3.4 and Section 4.3.5 respectively.

4.3.4 UART ISR for data acquisition in "Burst mode interrupt" program

In the program "Burst mode interrupt" on receiving character 'p', the data ready enable interrupt is disabled, FIFO interrupt is enabled and the microcontroller is then put to sleep mode. Since the size of the sensor FIFO buffer is 1024 bytes (85 samples and 4 bytes) and the sampling rate is 100 Hz, a sensor interrupt is raised after every 860 ms, thus a good amount of

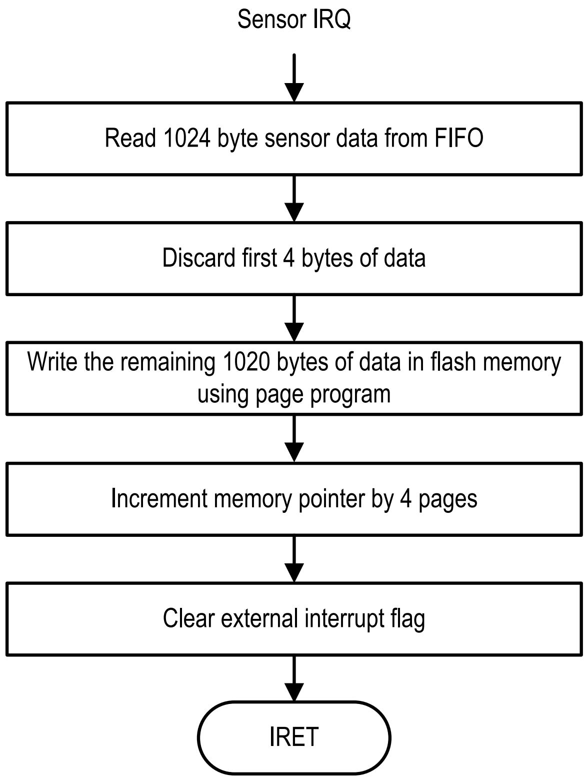


Figure 4.4: Sensor ISR in "Burst mode interrupt" program.

time will remain for signal processing after writing the data into the flash memory. The disadvantage with this method is that one sample is lost in every burst read. This happens because the buffer size is not an integer multiple of 6-axis data size (12 bytes) and by the time the interrupt is raised and the routine is serviced the remaining 8 bytes of the 86^{th} sample is pushed into the FIFO and thus first 8 bytes of the first sample are lost leaving the last 4 bytes of first sample in the FIFO. Therefore whenever the interrupt is raised, 1024 bytes of data are read from the sensor in burst mode but the first 4 bytes of data are discarded. The remaining 1020 bytes of data (85 samples) are written into the flash memory using page program instruction. The memory pointer is incremented by 4 page size, the interrupt flag is cleared, and the microcontroller is put to sleep mode. The flow chart for sensor ISR for "Burst mode interrupt" program is shown in Figure 4.4. The microcontroller can come out of this continuous acquisition process only by a UART IRQ to stop data acquisition. Due to its disadvantage another method was used for data acquisition which is explained in the next subsection.

4.3.5 UART ISR for data acquisition in "Burst mode polling" program

In the program "Burst mode polling", the sensor interrupt is not enabled; instead the FIFO counter registers are read continuously to check if *N* bytes have been acquired. As the FIFO is filled with *N* bytes, the data are read and stored in an internal buffer of *N* bytes. The data are then written into the flash memory using page program instruction. Meanwhile if a request is

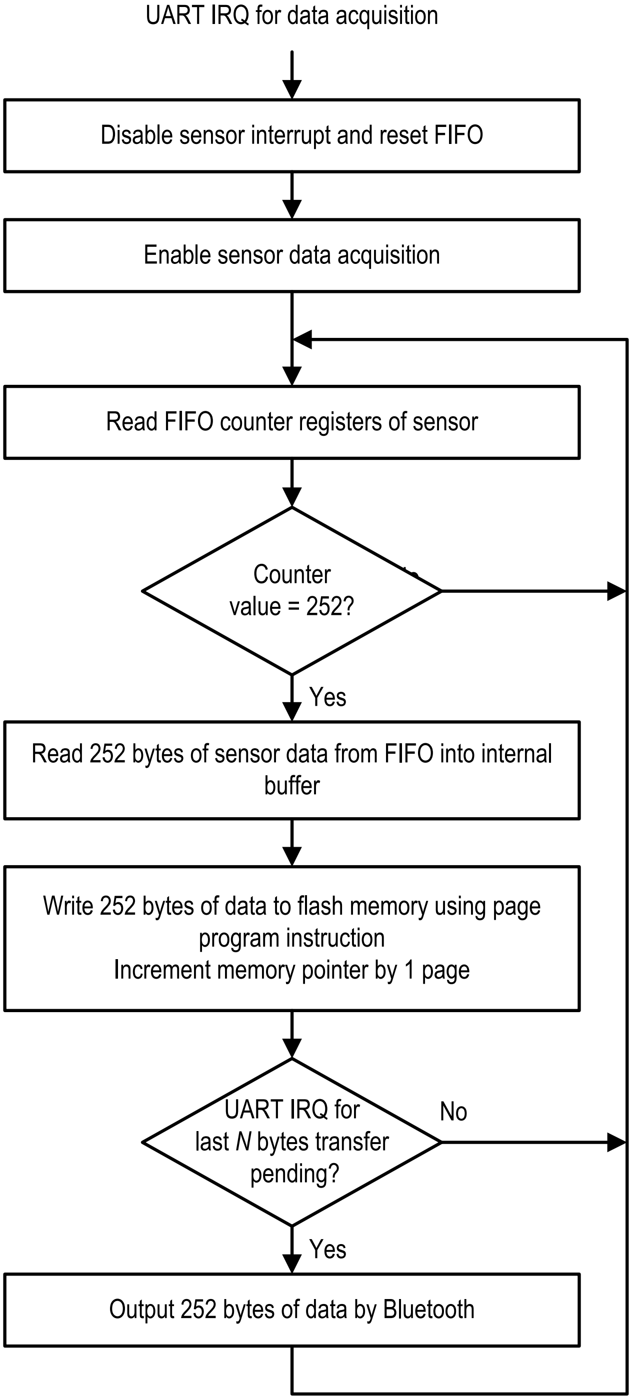


Figure 4.5: Data acquisition in "Burst mode polling" program.

raised by the user to send the last N bytes of data, then the same N bytes of data are written to the UART-Tx register to output them through Bluetooth. The interrupt flag is not cleared so that this data acquisition process continues in a loop. The microcontroller can come out of this loop only by a UART IRQ to stop data acquisition. The flow chart for this method is shown in Figure 4.4. The main advantage of this method is that samples are not lost as well as we can choose the value of N such that the time required to acquire $N/12$ samples is more than the time needed for processing the data block. The value of N was chosen as 252, 21 samples. At

Figure 4.6: Control Moment Gyroscope Model 750 from Educational Control Products [24].

100 Hz sampling, the time required for signal acquisition is 210 ms. One page of flash memory is 256 bytes, and hence the data can be written into the flash memory in just one page program instruction. The time required for transferring these values from FIFO to flash memory is relatively very small.

4.4 Data conversion

The sensor output is in 2's complement form which is transmitted by ISM in hexadecimal format. A PC is used for communicating with ISM and the received sensor data are saved in a single text file with each row starting with the sample number followed by three axis data of accelerometer and the three axis data of the gyroscope. This text file is then read using MATLAB code "Data conversion algorithm" and all the columns are saved as different variables (time, accl_x, accl_y, accl_z, gyro_x, gyro_y, and gyro_z). Each column except the 'time' column has data in hex format which has to be converted to g or °/s data. The accelerometer and gyroscope hex data are first converted to decimal form and then the accelerometer data are divided by 8192 and gyroscope data by 65.5 to get the data in required form of g and °/s respectively. The division is done because 16 bits (+2^15 to −(2^15 - 1)) represents 5 g for accelerometer and 500 °/s for gyroscope. This converted data are then saved in a 2D matrix for further processing and plotting.

4.5 Testing of ISM operations

To test the ISM operation, the module was fixed on the central platform of the Control Moment Gyroscope (CMG) "Model 750" from "Educational Control Products" [24], available in the Dynamics and Control Lab of the Department of Aerospace Engineering. The CMG has a central platform with two outer rings, as shown in Figure 4.6, allowing it a freedom of three degrees of rotation. An encoder is present at each axis of the CMG to record the angle of

Table 4.1: Accelerometer and gyroscope output for different orientations

Sensor	Accl-X		Accl-Y		Accl-Z		Gyro-X		Gyro-Y		Gyro-Z	
Orientation	Mean(g)	SD(g)	Mean(g)	SD(g)	Mean(g)	SD(g)	Mean(°/s)	SD(°/s)	Mean(°/s)	SD(°/s)	Mean(°/s)	SD(°/s)
X_Z_Y_0	-0.0166	0.0042	0.0836	0.0034	0.9139	0.0094	0.2440	0.4518	-2.0372	0.2516	-2.7231	0.0692
X_Z_Y_30	-0.0091	0.0055	-0.4149	0.0471	0.8137	0.0207	0.2324	0.1748	-1.8331	0.2647	-2.8497	0.0759
X_Z_Y_45	-0.0038	0.0117	-0.6285	0.0683	0.6743	0.0246	0.2279	0.0590	-1.8179	0.2667	-2.8471	0.0737
X_Z_Y_60	-0.0034	0.0167	-0.7994	0.0866	0.4860	0.0457	0.2263	0.0492	-1.8297	0.0580	-2.8462	0.0606
X_Z_Y_90	0.0143	0.0022	-0.9879	0.0021	-0.0223	0.0034	0.2249	0.0482	-1.8219	0.0537	-2.8502	0.0585
Y_Z_X_0	-0.0166	0.0043	0.0826	0.0337	0.9130	0.0308	0.2439	0.4516	-2.0369	0.2516	-2.7231	0.0691
Y_Z_X_30	-0.5005	0.0621	0.0789	0.0335	0.7611	0.0314	0.2523	0.0835	-1.9738	0.3112	-2.8238	0.0698
Y_Z_X_45	-0.7084	0.0039	0.0689	0.0026	0.5975	0.0133	0.2227	0.0651	-1.8272	0.2776	-2.8465	0.0731
Y_Z_X_60	-0.8496	0.0329	0.0635	0.0153	0.4032	0.0451	0.2236	0.1746	-1.8137	0.2790	-2.8416	0.0959
Y_Z_X_90	-0.9699	0.0583	0.0081	0.0145	-0.0692	0.1485	0.2182	0.0577	-1.8108	0.2904	-2.8477	0.0986
Z_X_Y_0	0.9373	0.3626	-0.0085	0.0171	-0.0253	0.1371	0.2375	0.0509	-1.8408	0.1963	-2.8993	0.0608
Z_X_Y_30	0.8243	0.3340	0.4739	0.0884	-0.0650	0.1450	0.2358	0.0498	-1.8494	0.1318	-2.8918	0.0629
Z_X_Y_45	0.6851	0.3041	0.6682	0.1310	-0.0885	0.1498	0.2307	0.0515	-1.8474	0.0833	-2.8870	0.0590
Z_X_Y_60	0.4421	0.2515	0.8540	0.1720	-0.1462	0.1621	0.2865	0.0636	-1.9417	0.0984	-2.8442	0.0658
Z_X_Y_90	-0.0122	0.1230	0.9902	0.1627	-0.0998	0.1237	0.1395	0.6742	-1.8608	0.1043	-2.1146	0.1451

rotation about that particular axis. The initial position of the platform and the outer rings can be fixed by applying the brakes provided for each axis. The CMG is connected to a PC which records the values of each encoder along with the time and therefore provides data of angular location with respect to time. The angular positions can be differentiated to get angular velocities.

To test the static data the brakes were applied on two axes such that the angle between them was 90° and ISM was rotated about the third axis and fixed at angles of 0°, 30°, 45°, 60°, 90°. The sensor data were recorded for each angle in sample-by-sample mode at a sampling frequency of 100 Hz for 10 s. This process was repeated for all the three axes. The mean and standard deviation of 1000 samples for each angle around each axis was calculated. These values are shown in Table 4.1. The first letter in each name depicts the axis about which ISM was rotated. The second and third letters of the name suggest the axis which had an initial vertical and horizontal orientation respectively. The number at the end of the name is the angle rotated about the fixed axis from the initial position in clockwise direction. The gyroscope values should be ideally 0 °/s for each axis in all orientations. An offset of around -

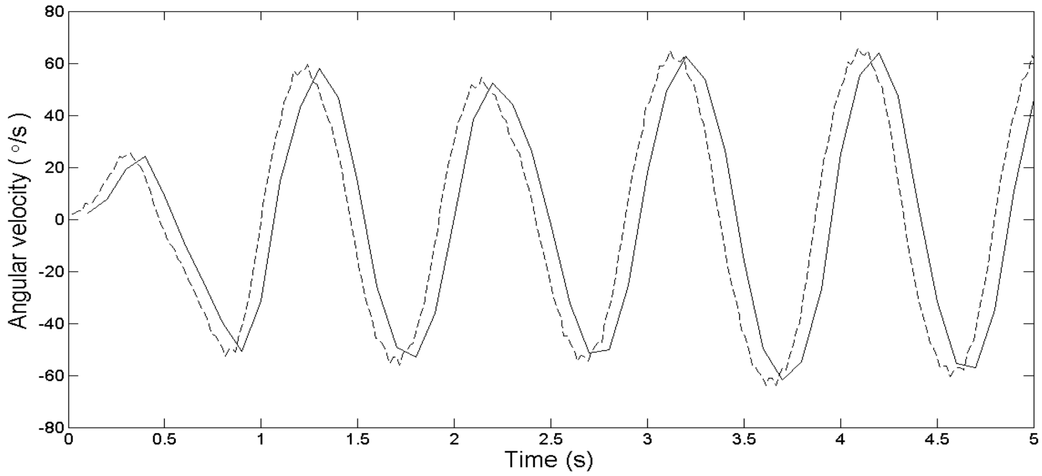


Figure 4.7: Gyroscope output about X-axis (solid line shows the output from ISM and broken line shows the output from CMG).

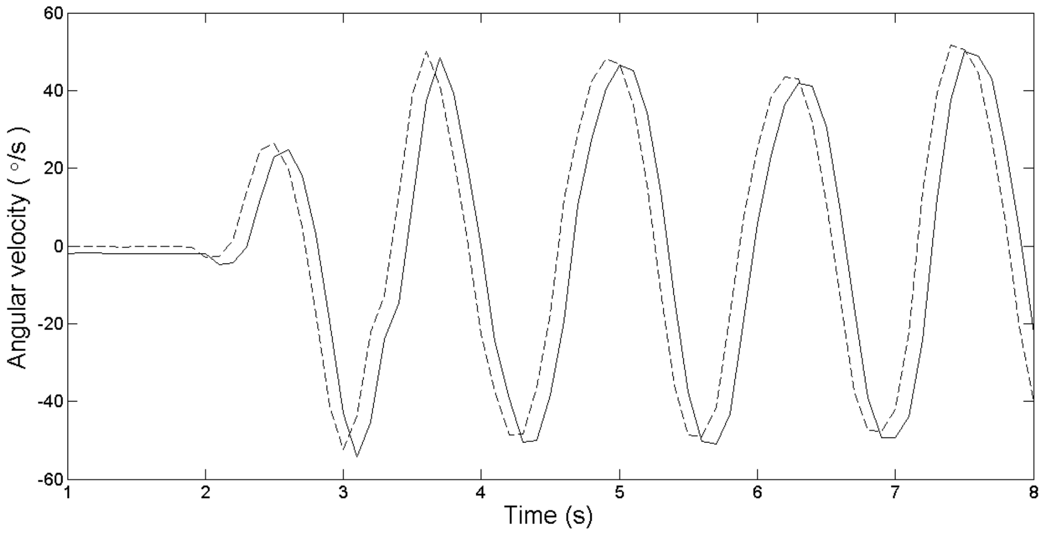


Figure 4.8: Gyroscope output about Y-axis (solid line shows the output from ISM and broken line shows the output from CMG).

2.8 °/s is observed in the z-axis of the gyroscope in each orientation. The maximum SD observed for that axis is 0.1451 °/s and thus the drift in the offset is acceptable. Some of the SD's are large while testing about z-axis which may have happened due to the vibration caused by other instruments in the lab. The accelerometer outputs on rotation of ISM change as expected. The X, Y, and Z axis values of the accelerometer when kept parallel to the gravitational axis were 0.9373 g, 0.9902 g, and 0.9139 in the orientation Z_X_Y_0, Z_X_Y_90, and X_Z_Y_0 respectively. The error from the expected value of 1 g was 0.0627 g, 0.0098 g, and 0.0861 g respectively. Some percentage of the total error introduced in the output may be due to the error in alignment of the PCB inside the box of ISM, the error introduced during placement of ISM on the CMG, and the error introduced due to the resolution of the encoders.

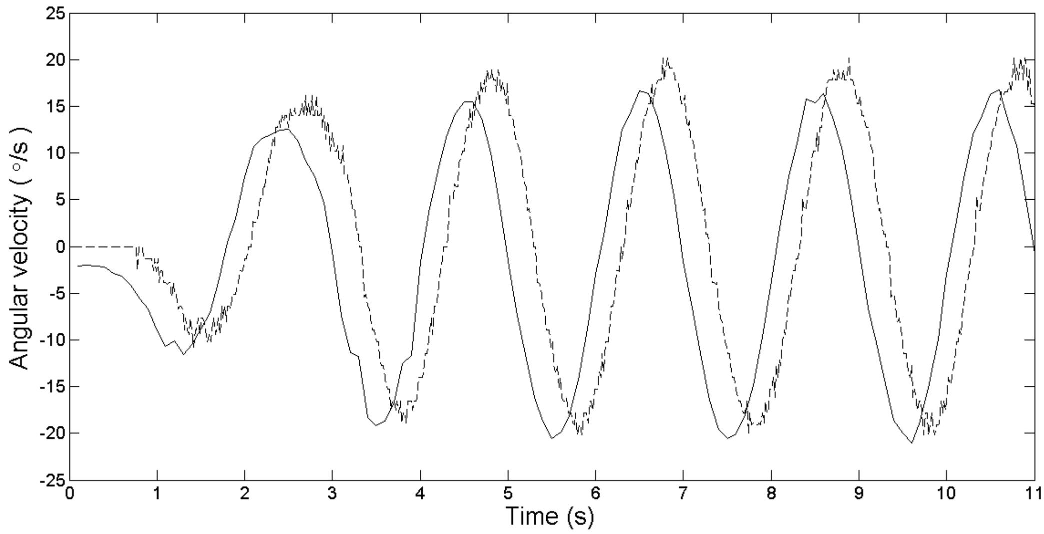


Figure 4.9: Gyroscope output about Z-axis (solid line shows the output from ISM and broken line shows the output from CMG).

To test the gyroscope data, the ISM attached at the centre of CMG platform was provided an oscillatory motion about each of the three axes by moving the platform or the outer rings to and fro. The keys to start the data acquisition from CMG and ISM were pressed simultaneously for proper alignment of data. Data were simultaneously collected at a sampling frequency of 100 Hz from ISM as well as the encoder of CMG. The program "Gyro plot" converted the angular location data from CMG to angular velocity data by differentiating it using the MATLAB function "diff" and plotted the same along with ISM data as shown in Figures 4.7, 4.8 and 4.9 to check the proper working of the gyroscope. The initial offset observed in these graphs are almost similar to the values found in Table 4.1 and as can be seen the ISM gyroscope data follows the data from the CMG therefore confirming the proper working of the sensor.

Chapter 5

REAL-TIME FALL DETECTION

5.1 Introduction

In the literature, as reviewed in Chapter 2, the most commonly used parameters for detecting falls are the magnitudes of the acceleration vector and angular velocity vector as sensed by tri-axial accelerometer and tri-axial gyroscope, respectively. Thresholds are applied on the parameters for detecting different types of fall. As the orientation of the sensor axes is not known, detection using magnitude alone may lead to errors. Fall detection method involving fusion of accelerometer and gyroscope data are computation intensive. We have investigated a computationally simple method, in which the fall detection is based on supra-threshold changes in the acceleration in any one of several directions. Let us represent the acceleration components along the three axes as *x*, *y*, *z*. It is proposed to monitor changes in the acceleration along the three axes, in the magnitudes of the acceleration in three orthogonal planes, and the total magnitude of the acceleration. Therefore, the variables used for fall detection are A_x, A_y, A_z, $A_{xy} = \sqrt{(A_x^2 + A_y^2)}$, $A_{yz} = \sqrt{(A_y^2 + A_z^2)}$, $A_{zx} = \sqrt{(A_z^2 + A_x^2)}$, $A_{xyz} = \sqrt{(A_x^2 + A_y^2 + A_z^2)}$.

5.2 Recordings with simulated falls

For an exploratory experiment, ISM was attached to the top of a wooden stick with length equal to the height of the waist of a normal person. The orientation of the module was such that its Y-axis was in vertical direction; Z-axis was horizontal in forward direction while X-axis was horizontal in sideways direction. It was then allowed to fall freely in the forward, backward, left, and right directions. The data were acquired using burst transfer by polling method. The seven parameters were plotted as shown in Figures 5.1, 5.2, 5.3, and 5.4. From the figures, we can observe that A_y plot starts with a value of 1 g as it is parallel to the gravitational axis initially, while A_x and A_z start from 0 g. All other plots except A_{zx} plot starts with a value of 1 g. There is a sudden dip in the value of Y-axis from 1 g to 0 g because of free fall under the influence of the gravity. This dip is reflected in all plots except A_{zx} plot. After the dip, there is an oscillation in all the three axes maximum frequency on

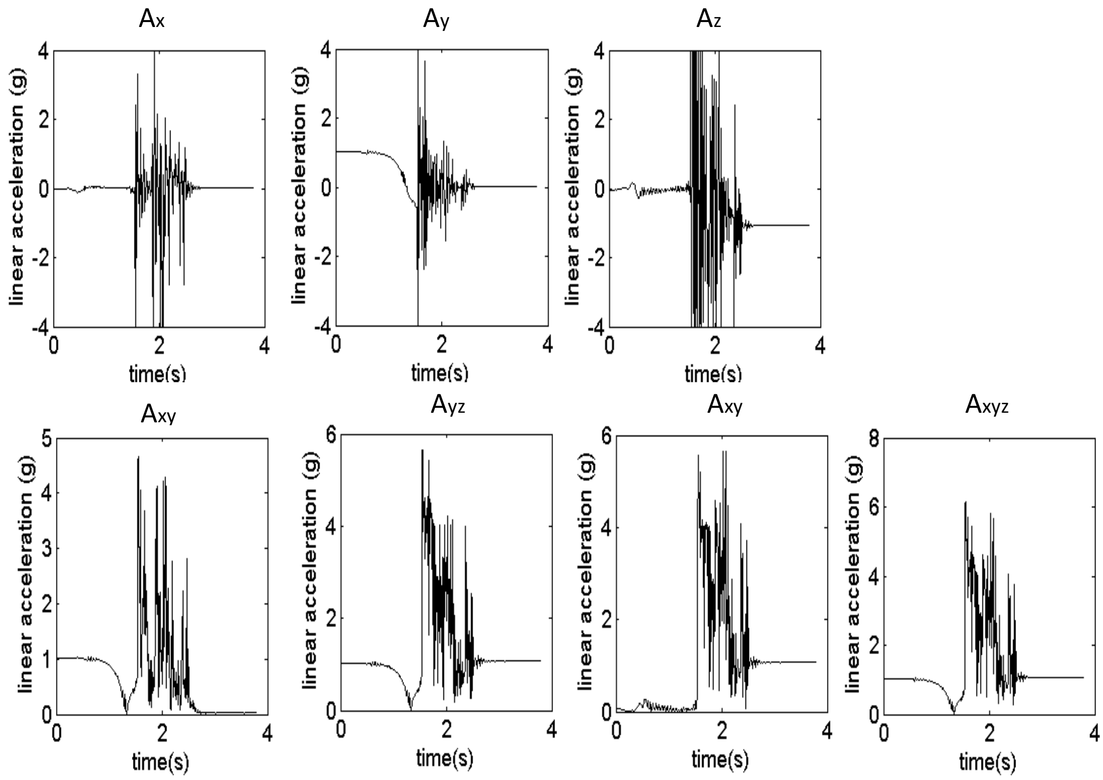


Figure 5.1: Fall forward: Accelerometer outputs and calculated magnitudes.

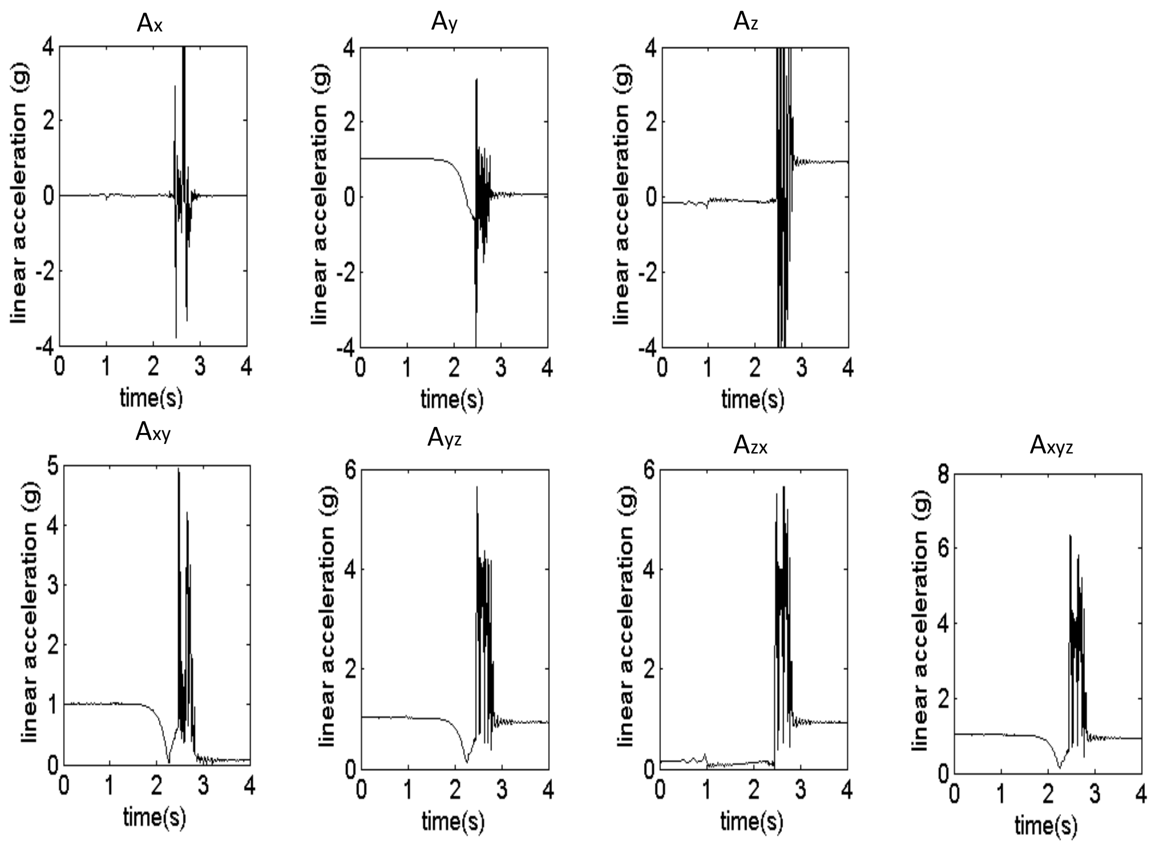


Figure 5.2: Fall backward: Accelerometer outputs and calculated magnitudes.

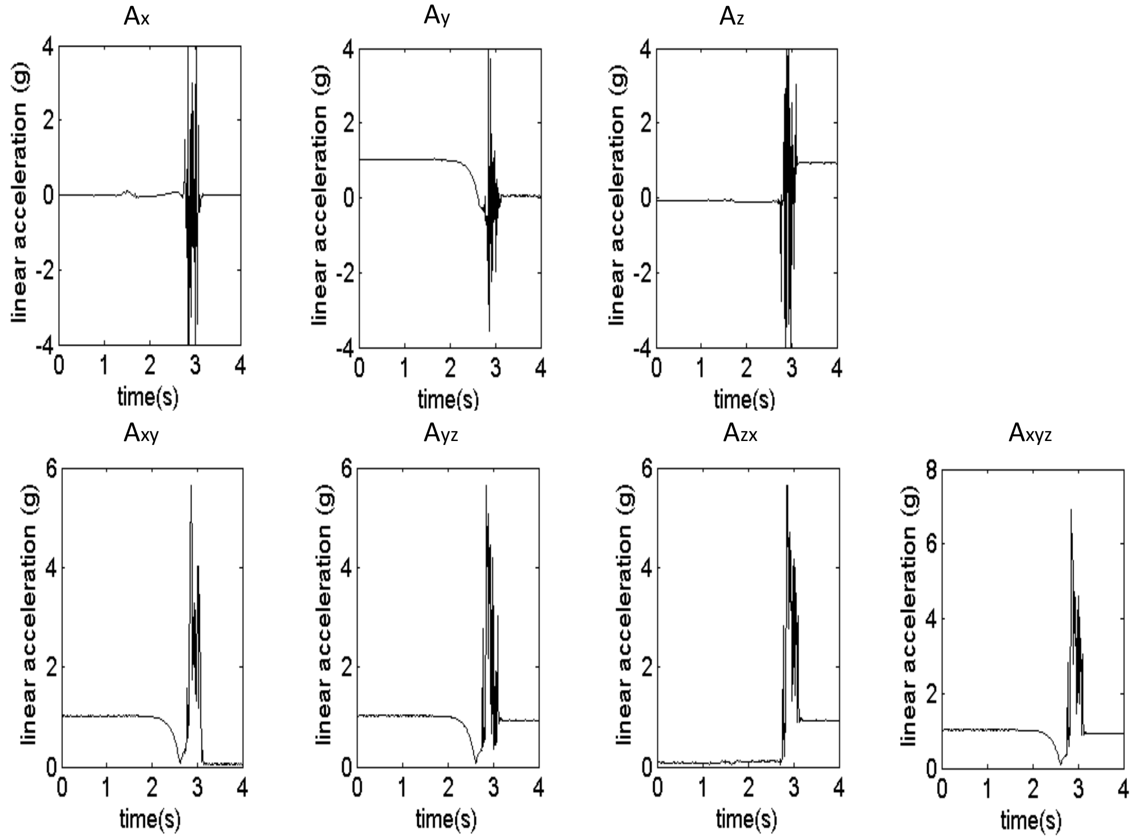


Figure 5.3: Fall left: Accelerometer outputs and calculated magnitudes.

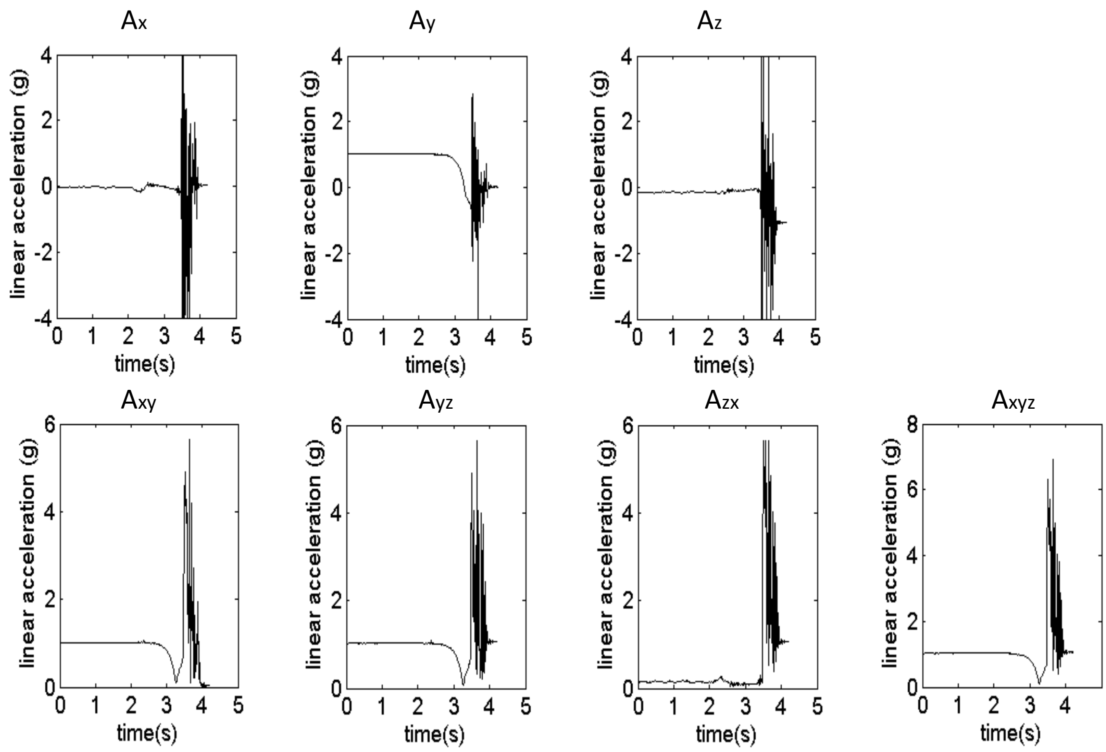


Figure 5.4: Fall right: Accelerometer outputs and calculated magnitudes.

z-axis in forward and backward fall and on x-axis in left and right fall. This is due to the impact of the fall. This oscillatory duration comes out as the aggregation of several peaks in positive direction in A_{xy}, A_{yz}, A_{zx}, and A_{xyz} plots thus making application of threshold easier. After the oscillation subsides a straight line is seen.

5.3 Fall detection algorithm

The fall detection algorithm monitors the three axial components of the acceleration, magnitudes of the acceleration in three orthogonal planes, and the magnitude in the three-dimensional space. The fall detection is based on the logic that a fall is associated with a large change from the mean value in at least one of these seven variables and the change should last for a certain minimum duration. Since the falls are sudden, the change in acceleration cannot exist for too long a duration, and therefore a change extending over a long duration is considered to be associated with ADL and not with fall.

At each sampling instant, we calculate seven variables from the three acceleration components as the following:

$$v_1(n) = A_x(n)$$
$$v_2(n) = A_y(n)$$
$$v_3(n) = A_z(n)$$
$$v_4(n) = \sqrt{((A_x(n))^2 + (A_y(n))^2)}$$
$$v_5(n) = \sqrt{((A_y(n))^2 + (A_z(n))^2)}$$
$$v_6(n) = \sqrt{((A_x(n))^2 + (A_z(n))^2)}$$
$$v_7(n) = \sqrt{((A_x(n))^2 + (A_y(n))^2 + (A_z(n))^2)}$$

For each of these variables, a 100-point moving average is calculated as the following

$$m_i(n) = m_i(n-1) + [v_i(n) - v_i(n-100)]/100$$

With a sampling rate of 100 Hz, this corresponds to moving average over 1 s duration. Change in the variable from the mean value is calculated as

$$d_i(n) = |v_i(n) - m_i(n)|$$

Each d_i is compared with a threshold θ, and if the value remains supra-threshold for a duration greater than t_1 but not greater than t_2, it is considered to be indicating a fall. The duration thresholds $t_1$and t_2 are the minimum and maximum estimates of the duration for which a fall can last. This fall-duration window detection part of the fall detection algorithm is shown in Figure 5.5. If d_i becomes less than the threshold θ before the duration t_1 then the variable duration is reset and the algorithm starts afresh. If d_i remains greater than θ for duration more than t_1, it is checked for becoming smaller than θ. If it becomes smaller before the duration t_2, a fall condition is declared, and the detection starts afresh. If d_i remains greater than θ for duration more than t_2 then it is inferred that the variable is in continuous oscillation due to activities of daily life and hence is ignored until d_i becomes less than θ, after which the

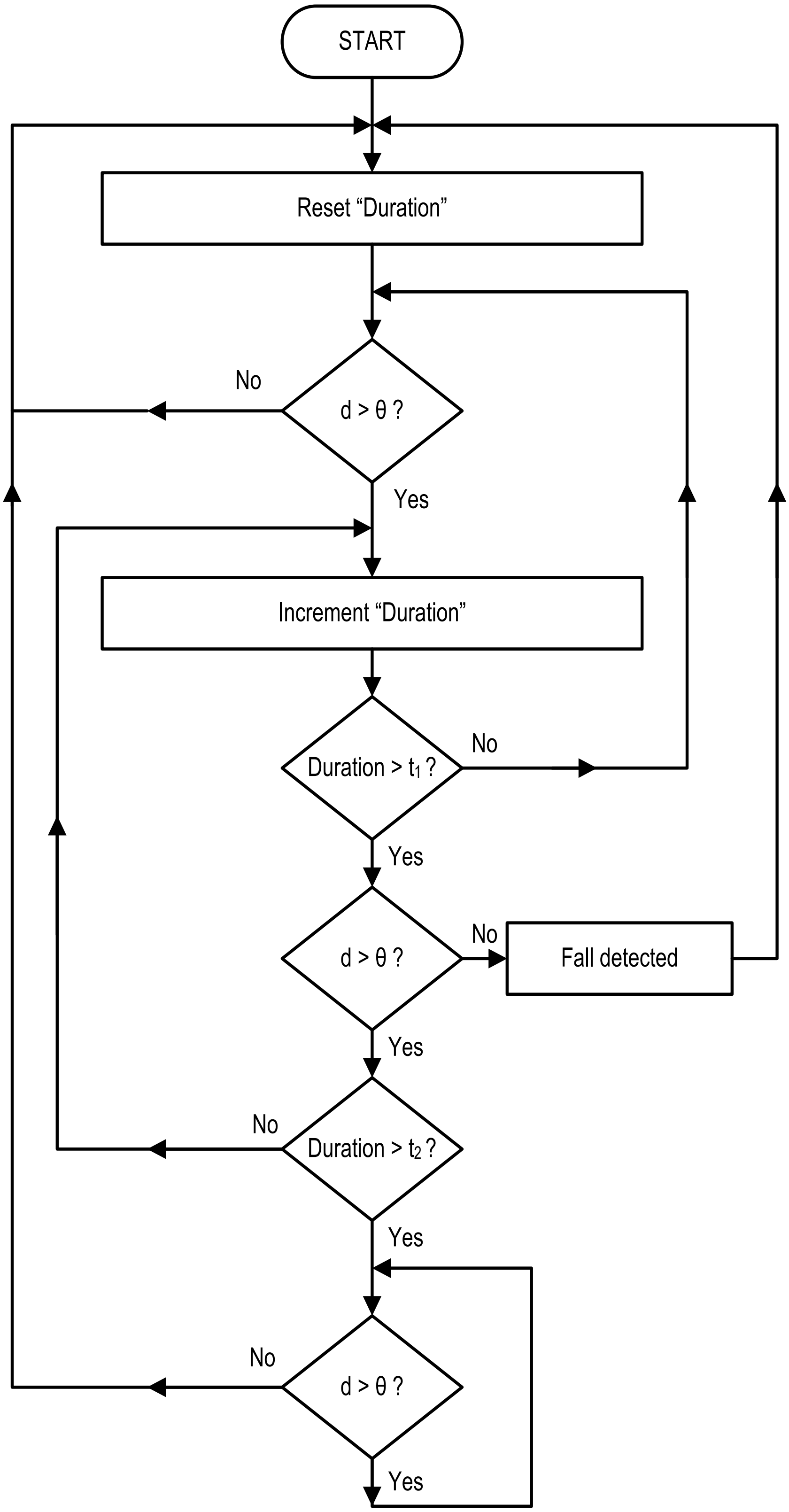


Figure 5.5: Fall-duration window detection

variable duration is reset and the fall-duration window detection starts afresh. The fall detection method is continuously applied on all seven variables. The fall is declared if it is detected for at least one of the variables.

(a) (b)

Figure 5.6: Experimental setup for testing: (a) fall; (b) activities of daily life.

5.4 Results

Fall detection algorithm was tested for various types of fall and activities of daily life (ADL). Fall detection was carried out using simulated falls with ISM attached to a stick as shown in Figure 5.6(a). For ADL, it was worn on the waist as shown in Figure 5.6(b).

To test detection of different types of fall, ISM was attached to the top of a stick of height approximately equal to the waist level. Different types of fall (fall forward, fall backward, fall left, fall right and rotate and fall) were simulated using the stick. Each type of fall was repeated for 5 times. All falls were detected correctly, leading to 100% sensitivity. To test the specificity, the ISM was worn by a person on the waist. There was no fall detection during these tests. Thus the results indicated 100% specificity as well.

Examples of fall and ADL plots are given in Appendix D and Appendix E respectively. ADL was separated from fall by choosing $\theta = 2$ g, $t_1 = 250$ ms, and t_2 as 850 ms. An example of fall detection due to different variables is shown in Figure D.1 in which 5 variables satisfied the criteria for fall. Similar results were observed in all other falls. The variable along the gravitational axis and the variables for acceleration magnitude in the three orthogonal planes and in the 3D space always detected the fall while the other two variables did not detect it. In ADLs skipping, jogging, and sit fast d_i crossed θ but for duration less than t_1 and hence these activities were not detected as fall. In other ADL's the acceleration magnitudes never crossed the threshold.

Chapter 6

SUMMARY AND CONCLUSIONS

The objective of the project was to develop an inertial sensing module capable of continuously monitoring the body movement, transmitting data wirelessly, and alerting a base station on detecting fall. An integrated MEMS sensor with 3-axis accelerometer and 3-axis gyroscope along with a microcontroller has been used in the design. The sensed data are in the form of 6 two-byte words and can have sampling frequency of up to 1 kHz. A 64 Mb flash memory has been used for data logging. It can record data continuously for approximately 2 hours at a sampling frequency of 100 Hz. A Bluetooth module has been incorporated to provide wireless connectivity for controlling the operation and data transfer without using any switches and connectors. The design has been implemented on a 2-layer PCB. The module has been tested for proper working of all blocks of the hardware. The circuit operates at 3.3 V with a maximum current consumption of 50 mA with the Bluetooth module operating, 35 mA during sensing and storage, and 3 mA during sleep mode. The module has been assembled to work with a 9 V battery and a linear regulator.

Software has been developed for data acquisition in sample-by-sample, and burst modes. It also has the facility for periodically transferring data in a time multiplexed manner from multiple ISMs. Operation of the sensor and data transfer operation has been verified by mounting ISM on a control moment gyroscope for different static tilts and oscillatory movements. A real-time algorithm for fall detection using the acceleration data has been developed and tested for satisfactory detection of simulated falls. It applies a threshold on the change from the running average on seven variables: acceleration along the three axes, and magnitude of the acceleration in three orthogonal planes and in 3D space. If the change remains above the threshold for duration greater than t_1 but less than duration t_2, a fall is declared. If the change crosses the threshold for duration less than t_1, the algorithm starts again. If the change crosses the threshold for duration more than t_2, the algorithm waits for the difference to go below the threshold after which it starts again.

As part of future work, the developed prototype needs to be more extensively tested, particularly under realistic fall and ADL conditions on a large number of subjects. Fusion of accelerometer and gyroscopic data needs to be carried out. Multiple modules communicating with a base unit in a time multiplexed manner can be used for simultaneous tracking of different body parts and devising a real-time classifier for fall and other activities. A power source using a chargeable and compact battery needs to be devised.

Appendix A

COMPONENT LIST

Table A.1: Component list of the ISM

Component designator	Component description	Part number / Value	Quantity
C1, C3, C4, C6, C7, C8, C10, C12	Capacitor, ceramic, surface mounted	0.1 µF	8
C2, C5, C9	Capacitor, ceramic, surface mounted	10.0 µF	3
C11	Capacitor, ceramic, surface mounted	2.2 nF	1
R3, R4	Resistor, surface mounted	4.7 kΩ	2
R5, R7	Resistor, surface mounted	220.0 Ω	2
R6	Resistor, surface mounted	100.0 Ω	1
LED1	LED, surface mounted	RED	1
LED2	LED, surface mounted	GREEN	1
CN1, CN2	Connector, 5-pin	CON5	2
CN3	Connector, 2-pin	CON2	1
J1, J2	Jumper	2 pin	2
SW1	SPDT	SW1	1
U1	IC, serial flash, 8 pin, SOIC	SST25VF064C	1
U2	IC, regulator, 5-pin, SOT	MCP1802	1
U3	Bluetooth module	RN-42	1
U4	IC, 6-axis, accelerometer and gyroscope, 24 pin, QFN	MPU-6000	1
U5	IC, microcontroller, 44-pin, TQFP	24FJ64GB004	1

Appendix B

SCHEMATIC DIAGRAM OF ISM

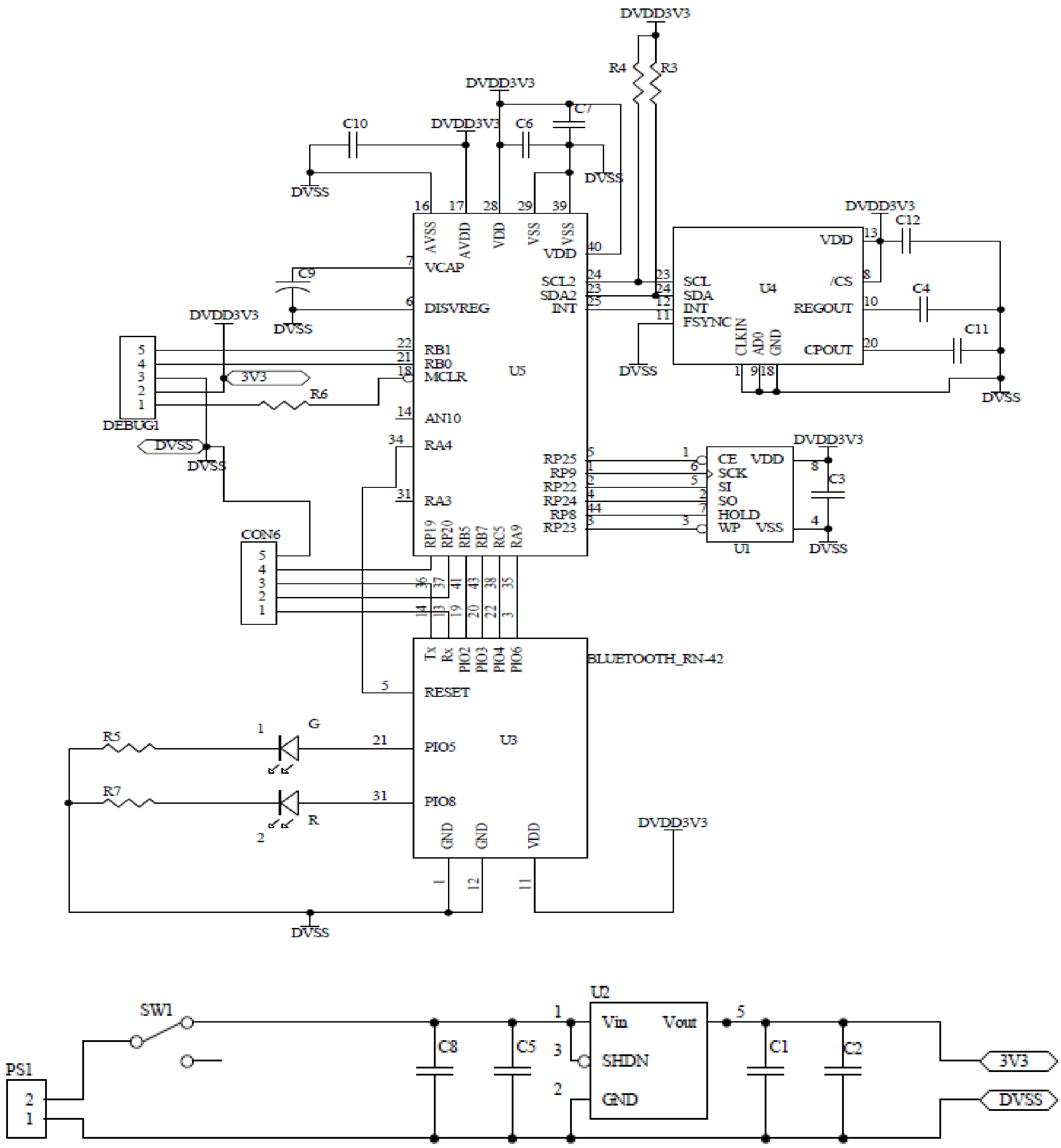


Figure B.1: Schematic diagram of Inertial Sensing Module

Appendix C

PCB LAYOUT

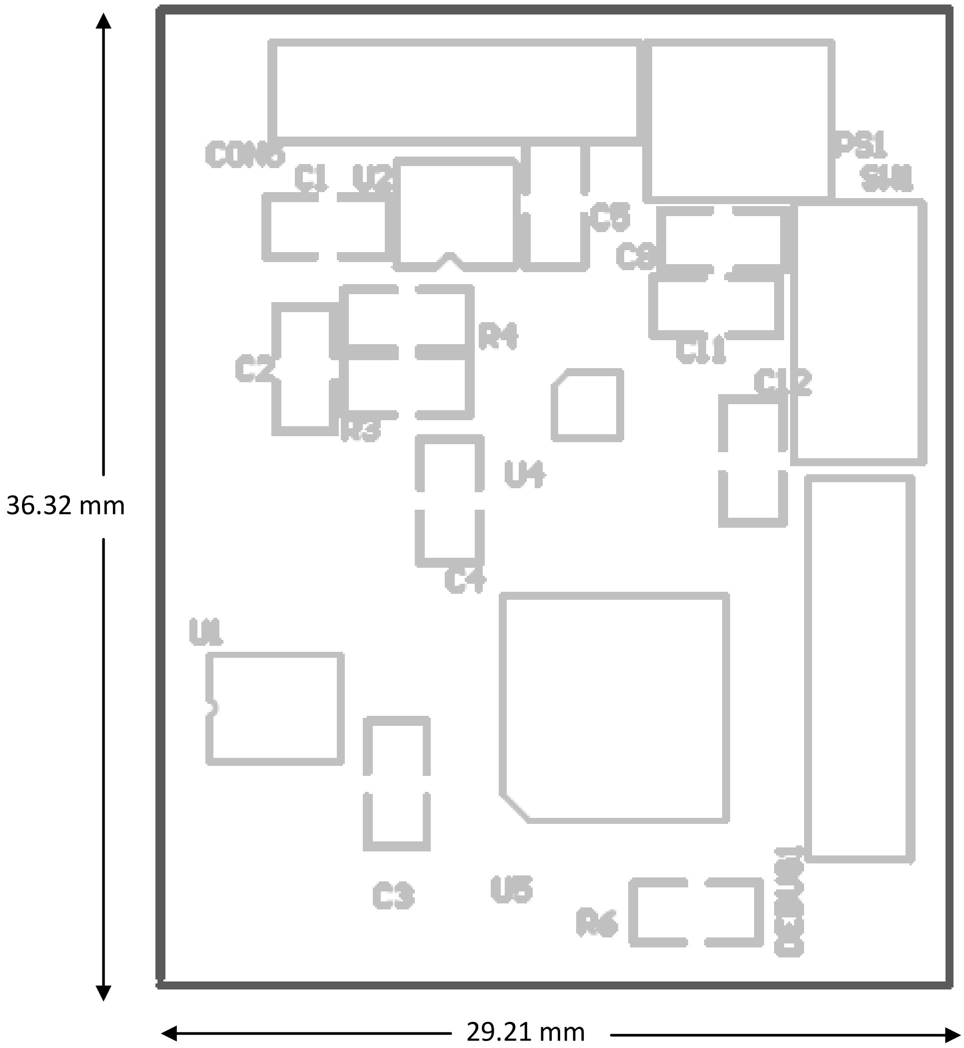


Figure C.1: Top overlay layer of PCB

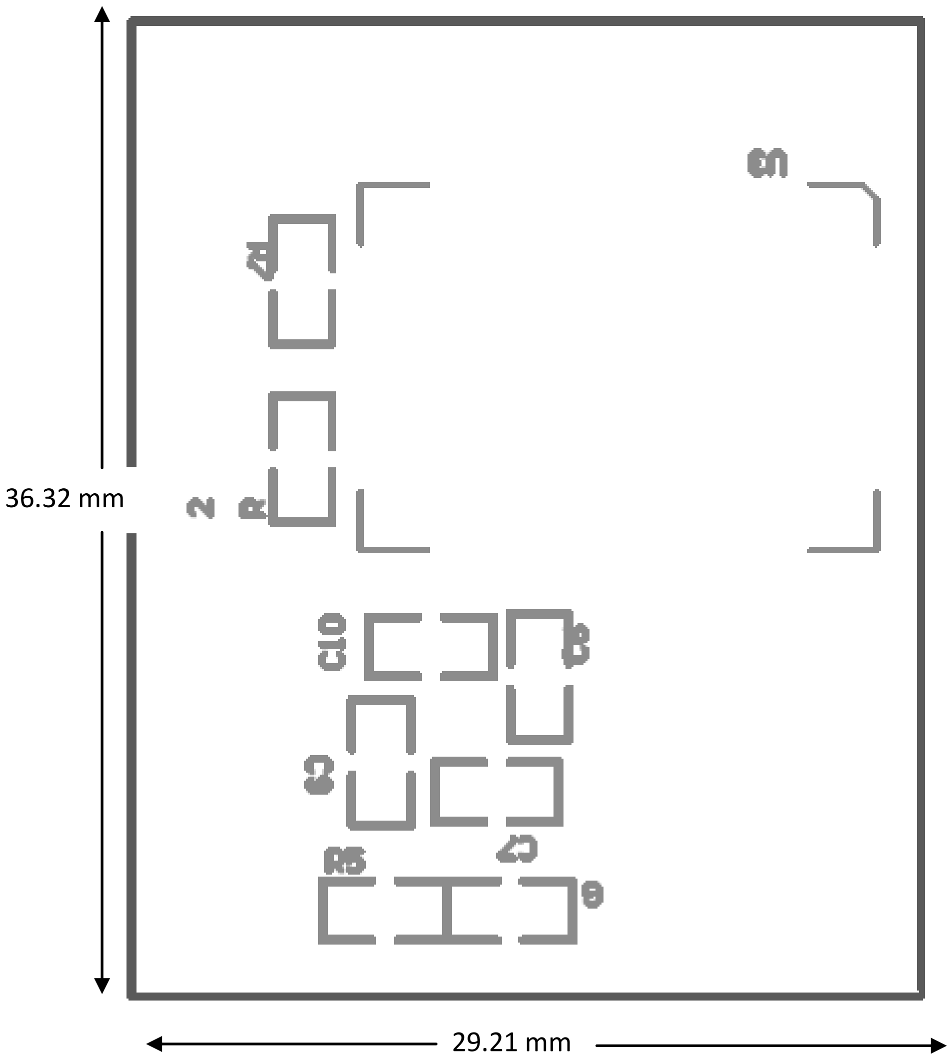


Figure C.2: Bottom overlay layer of PCB

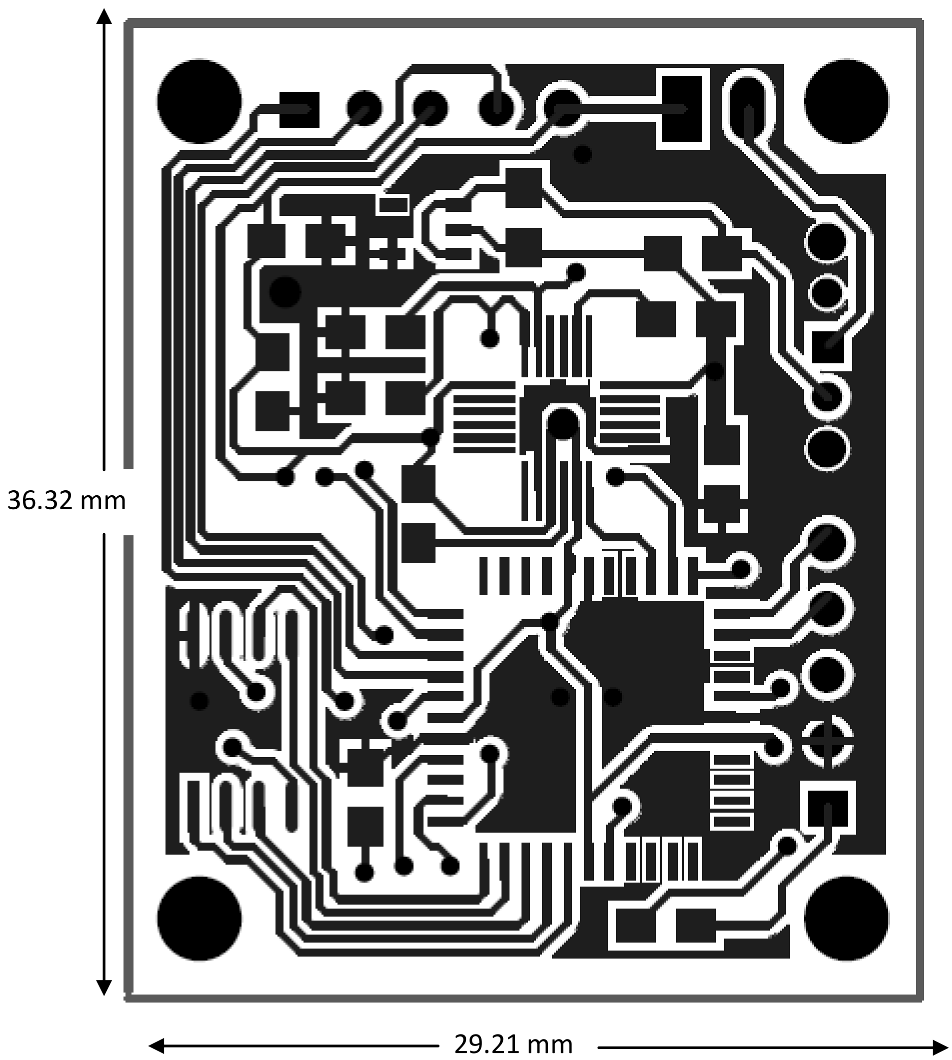


Figure C.3: Top layer of PCB

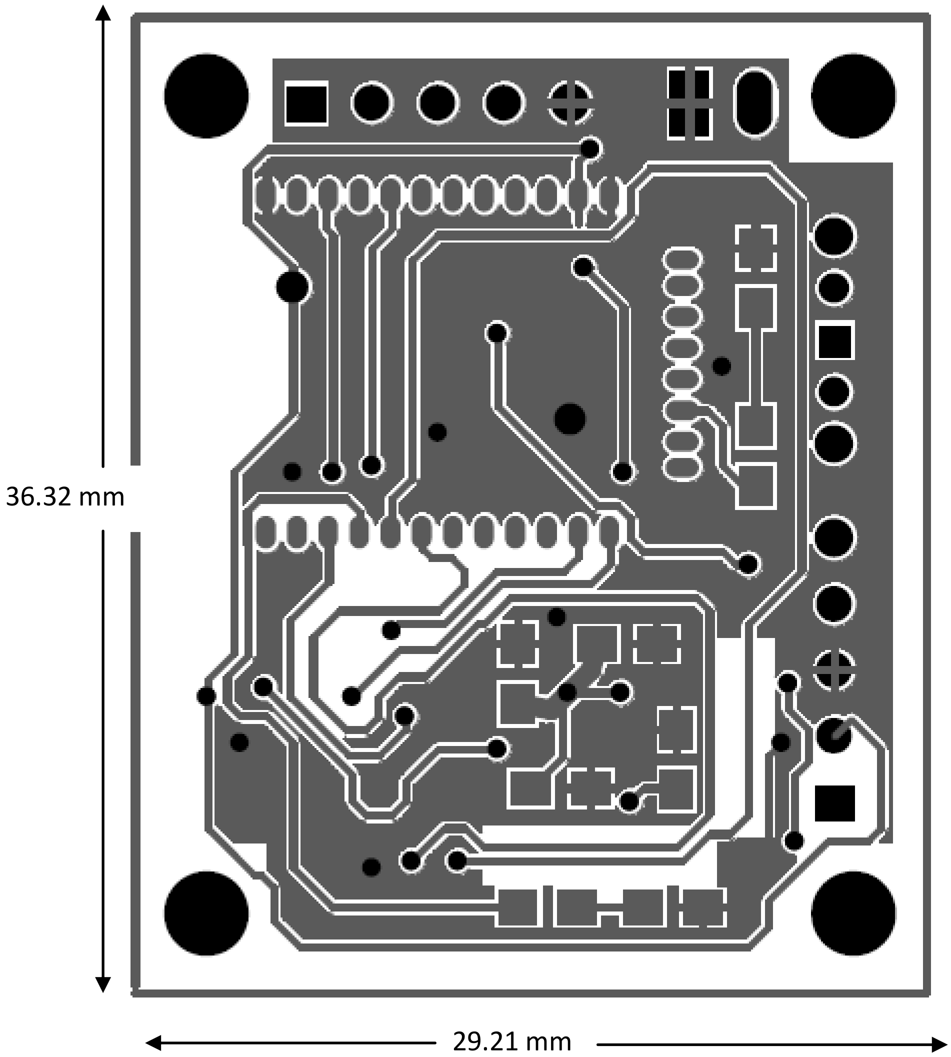


Figure C.4: Bottom layer of PCB

Appendix D

FALL EXAMPLES

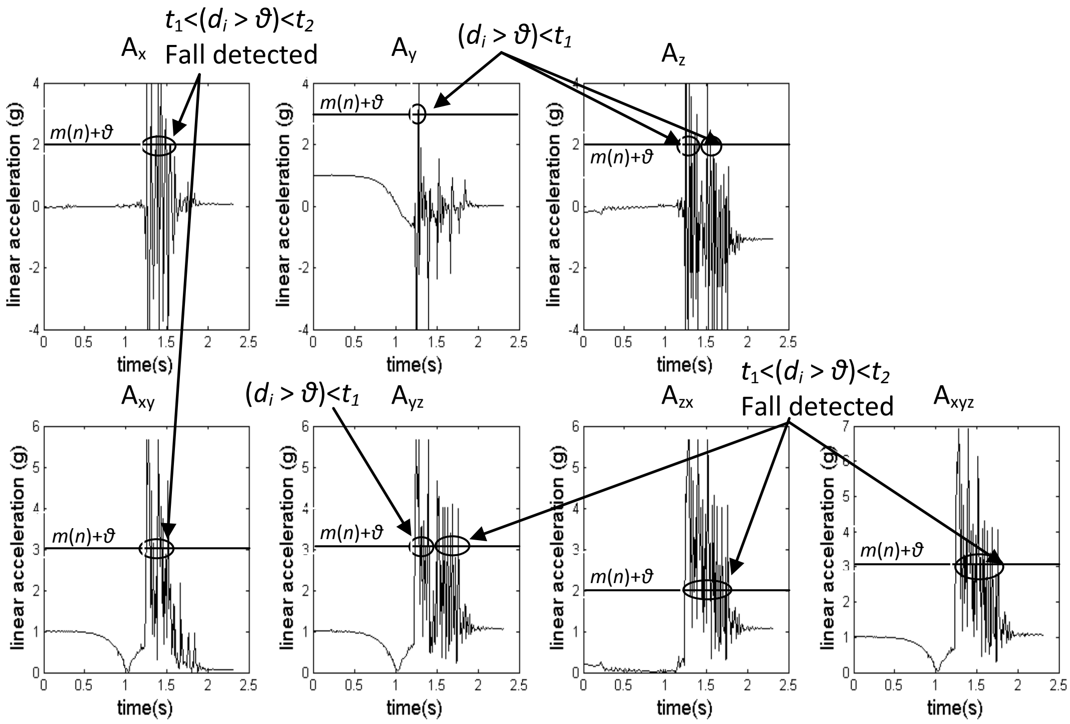


Figure D.1: Fall forward: Accelerometer outputs and calculated magnitudes.

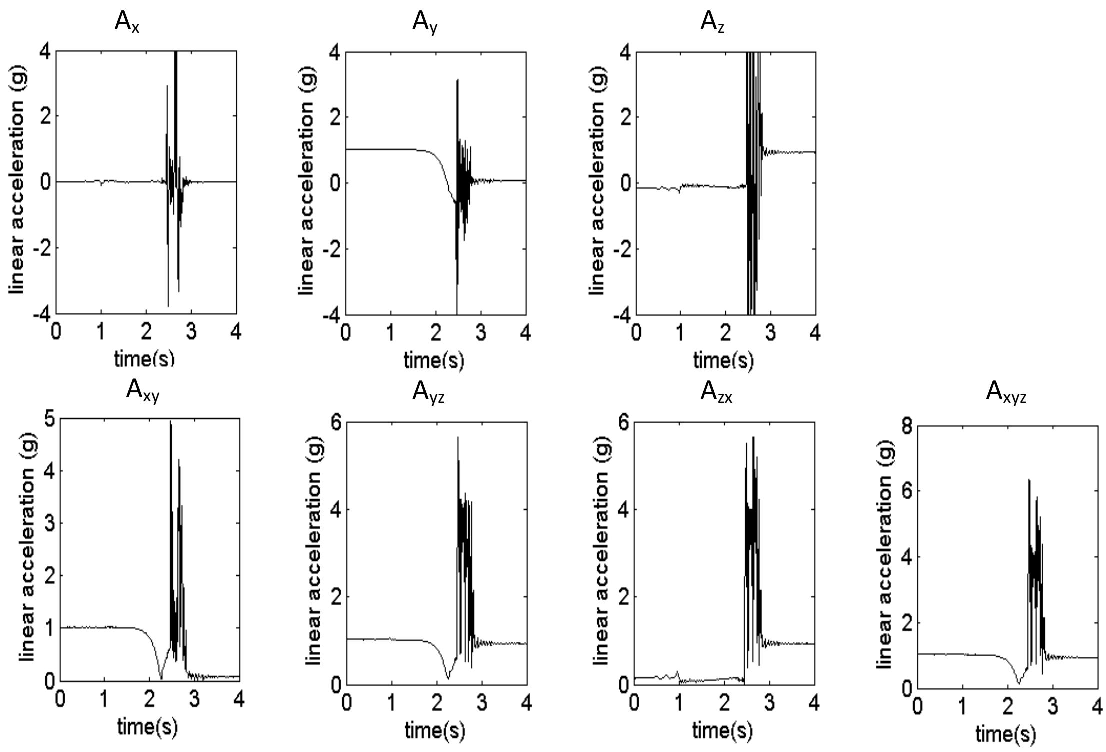


Figure D.2: Fall backward: Accelerometer outputs and calculated magnitudes.

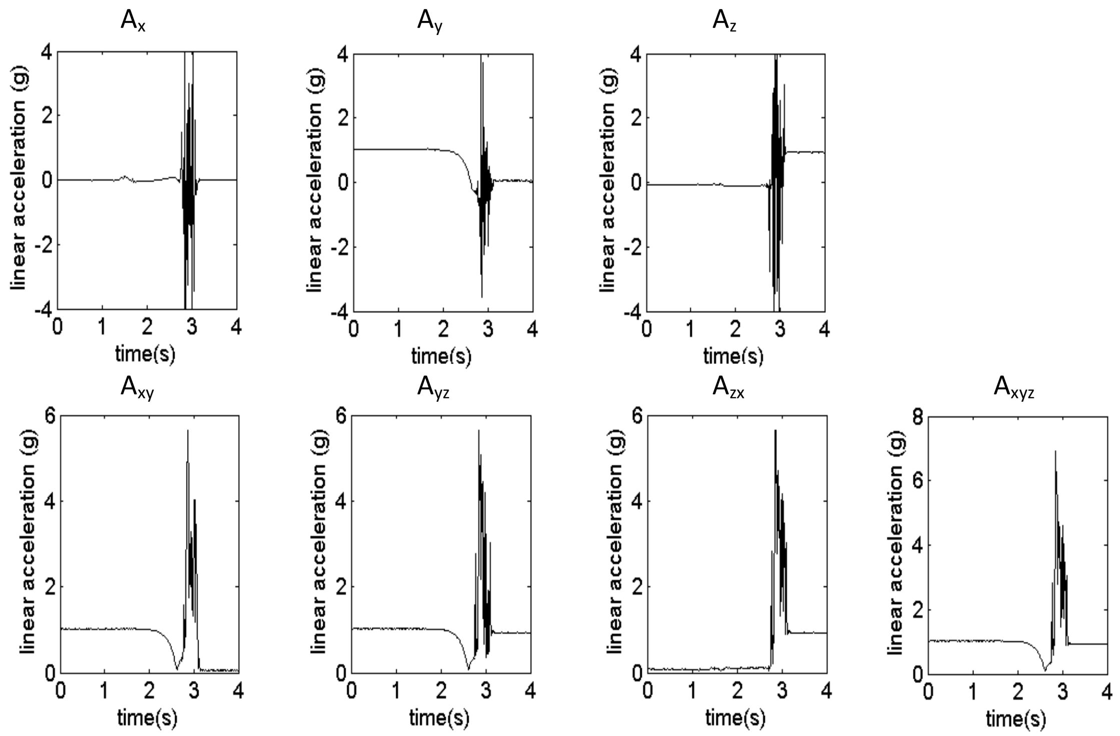


Figure D.3: Fall left: Accelerometer outputs and calculated magnitudes.

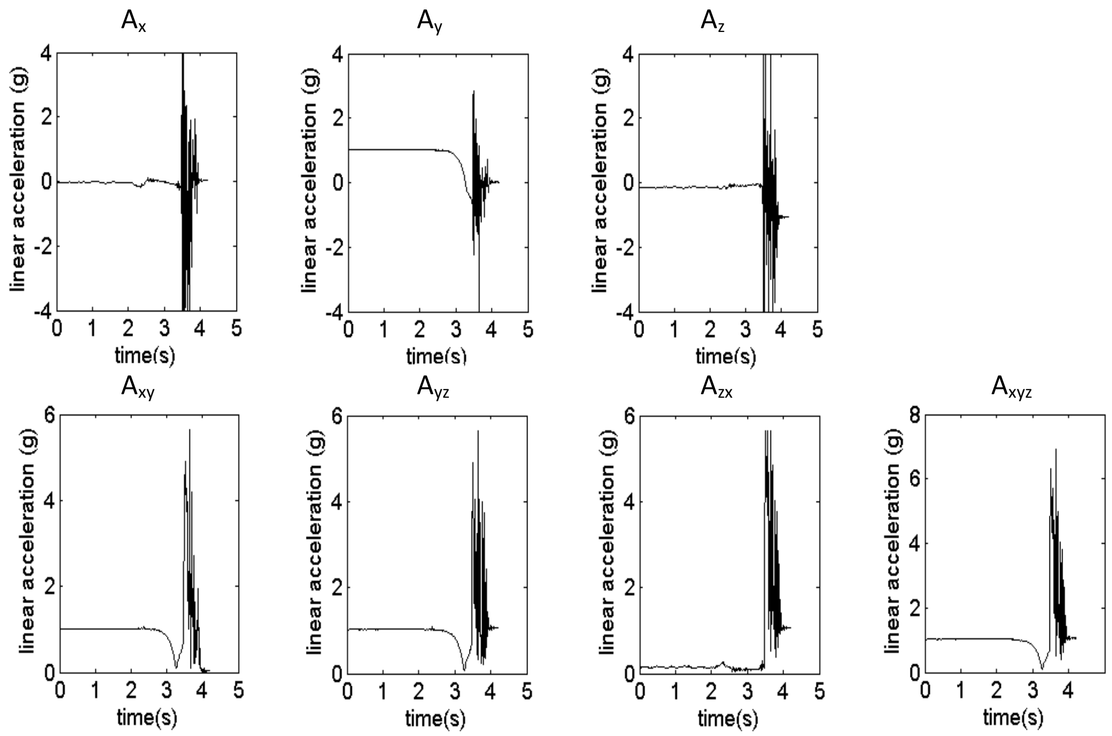


Figure D.4: Fall right: Accelerometer outputs and calculated magnitudes.

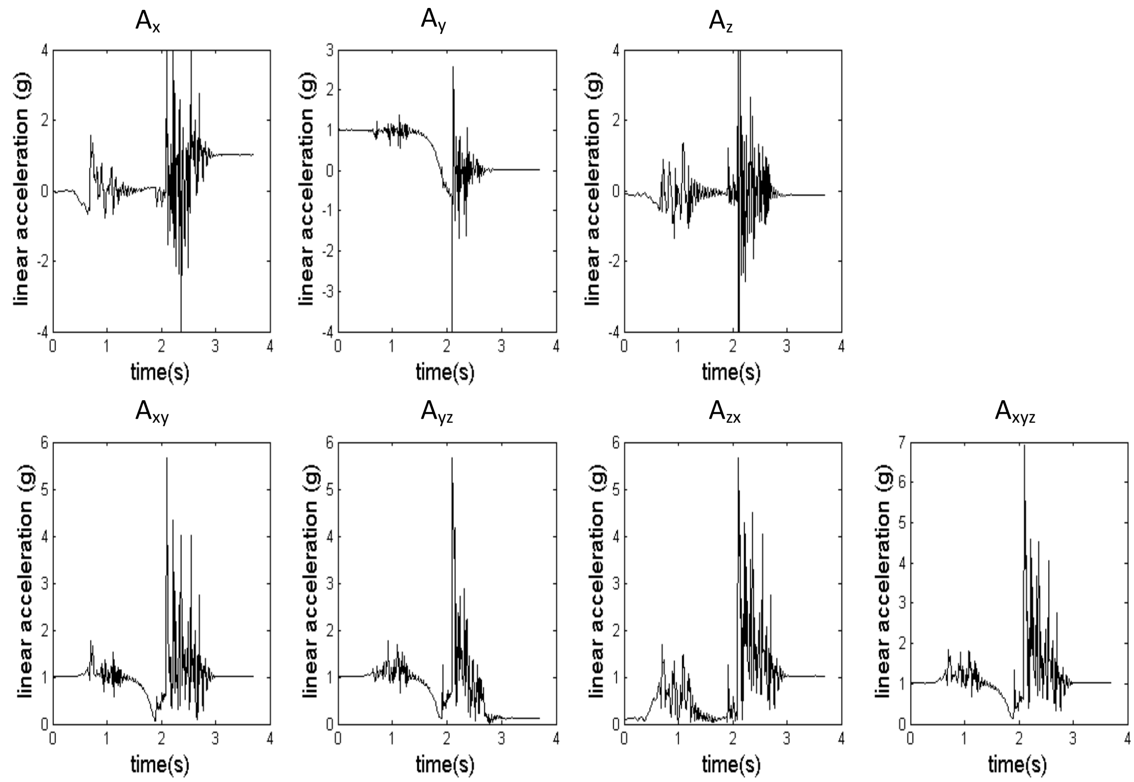


Figure D.5: Rotate and fall: Accelerometer outputs and calculated magnitudes.

Appendix E

ADL EXAMPLES

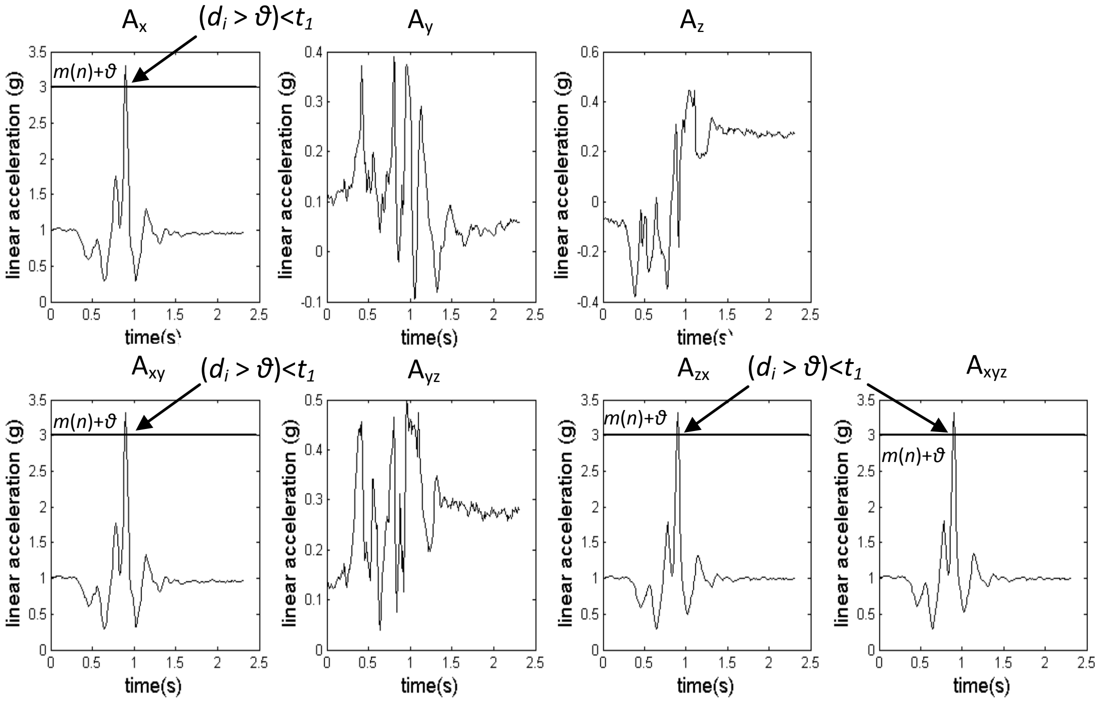


Figure E.1: Sit fast: Accelerometer outputs and calculated magnitudes.

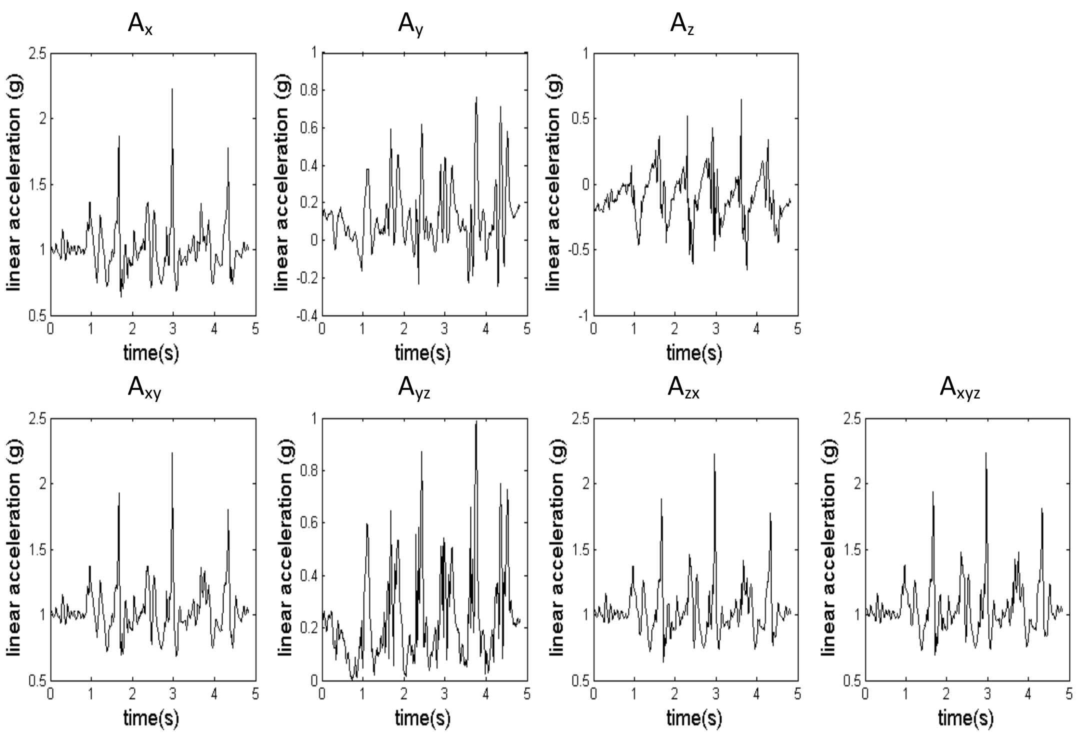


Figure E.2: Walking: Accelerometer outputs and calculated magnitudes.

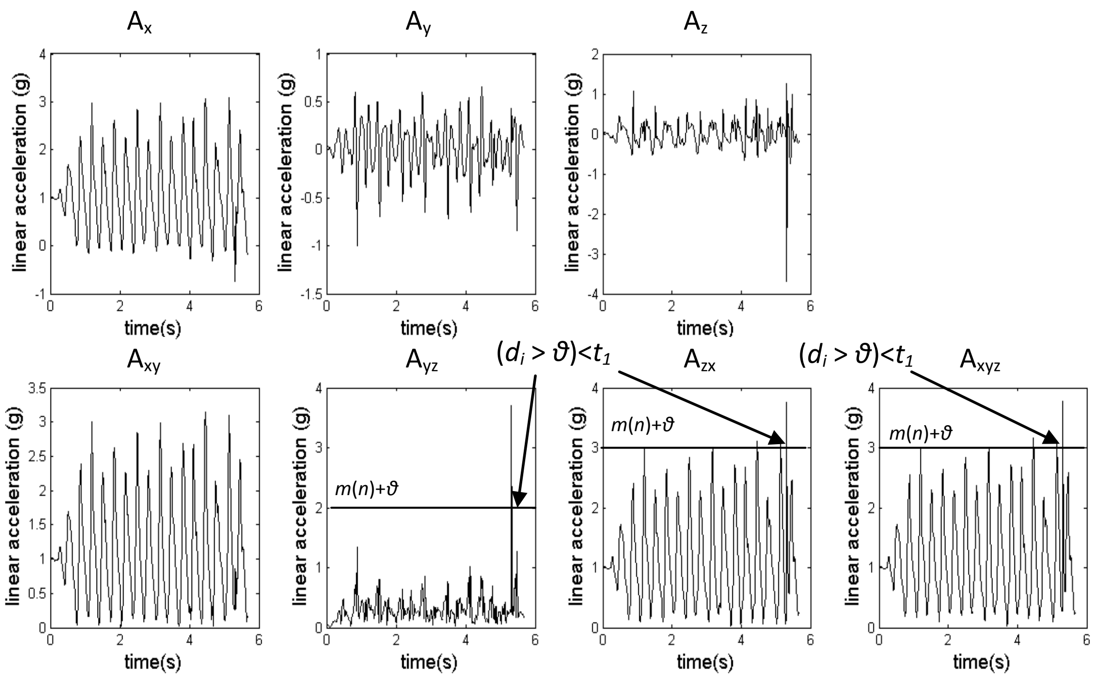


Figure E.3: Jogging: Accelerometer outputs and calculated magnitudes.

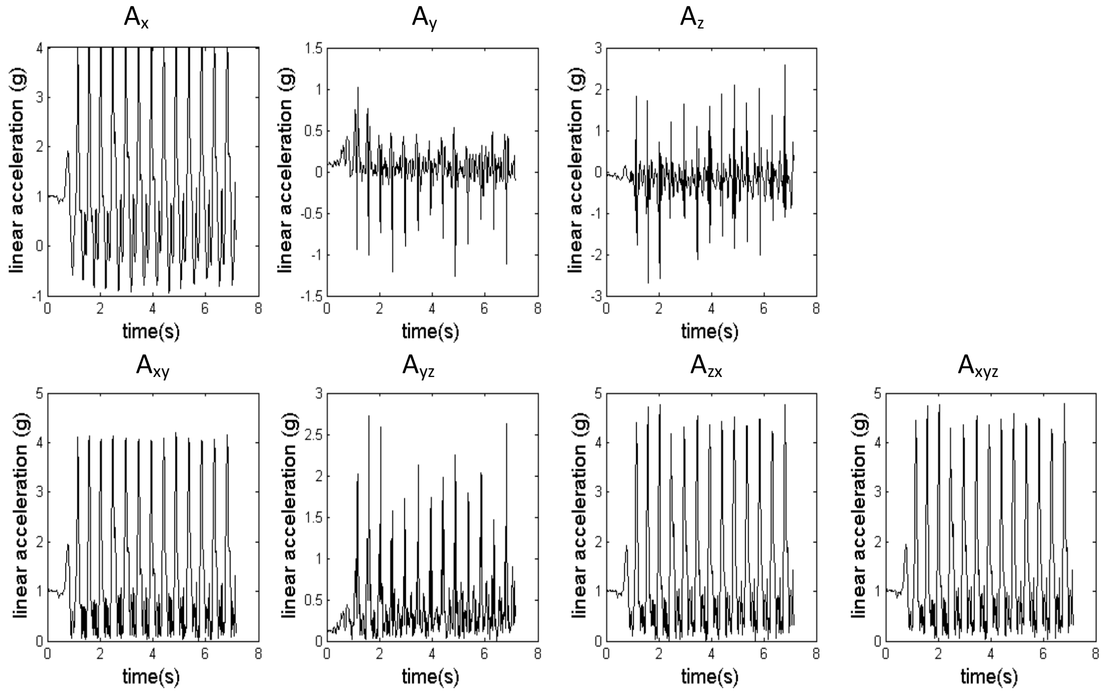


Figure E.4: Skipping: Accelerometer outputs and calculated magnitudes.

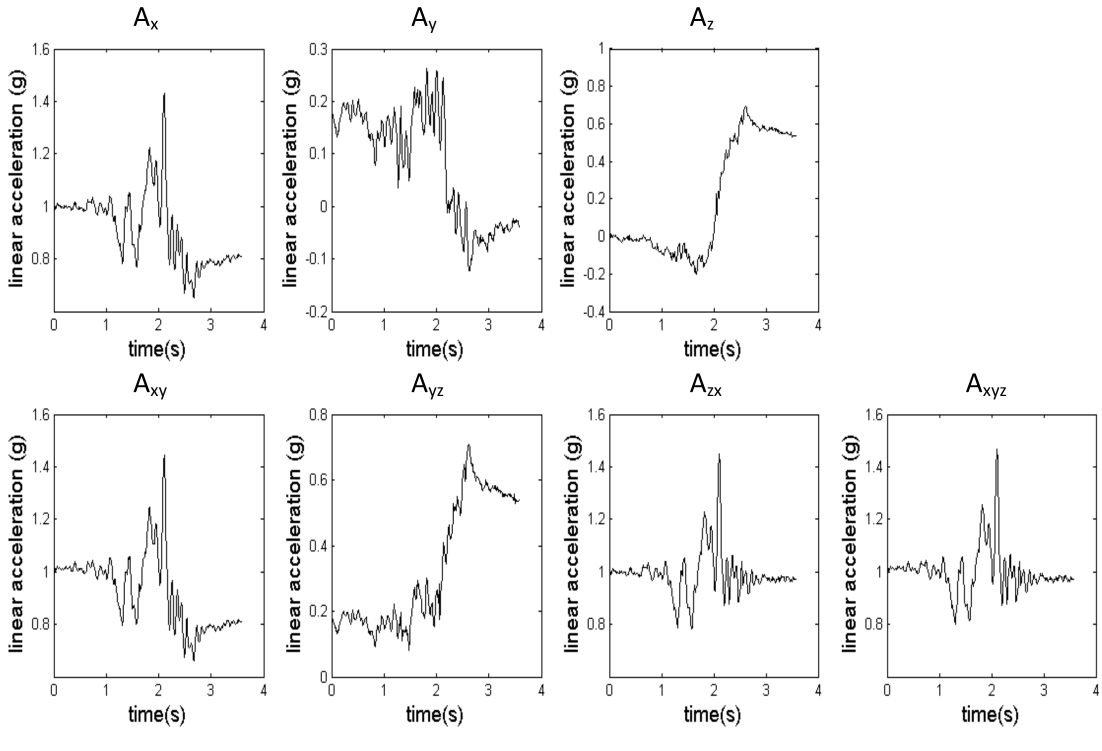


Figure E.5: Sit slowly: Accelerometer outputs and calculated magnitudes.

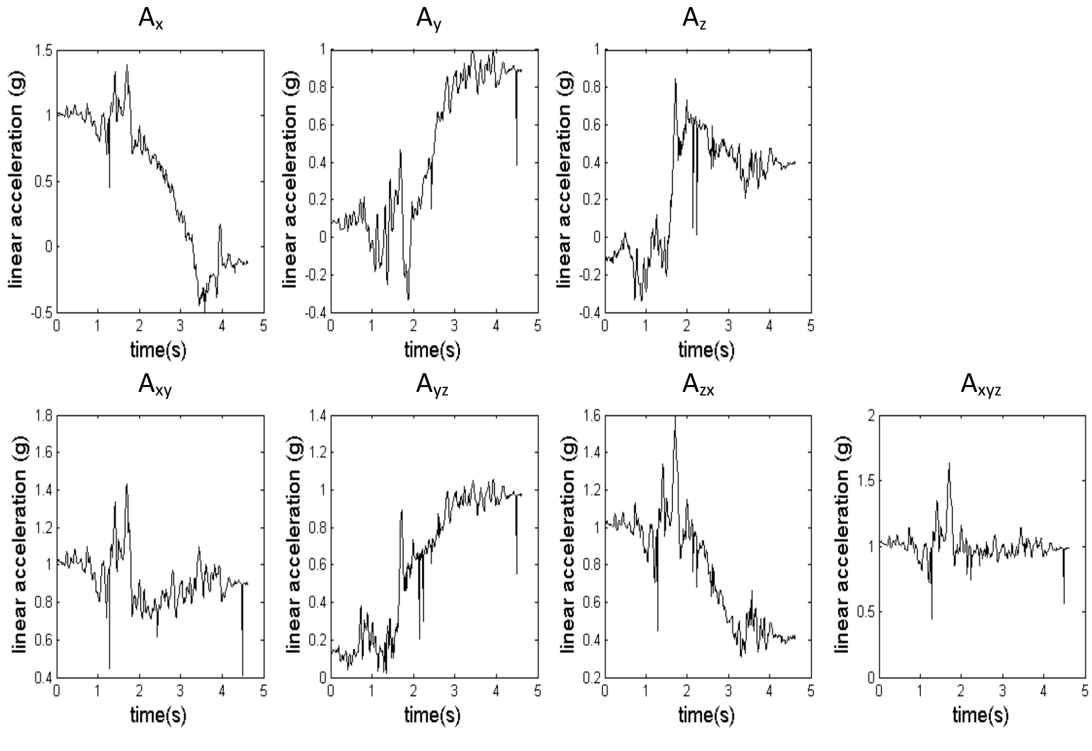


Figure E.6: Lying: Accelerometer outputs and calculated magnitudes.

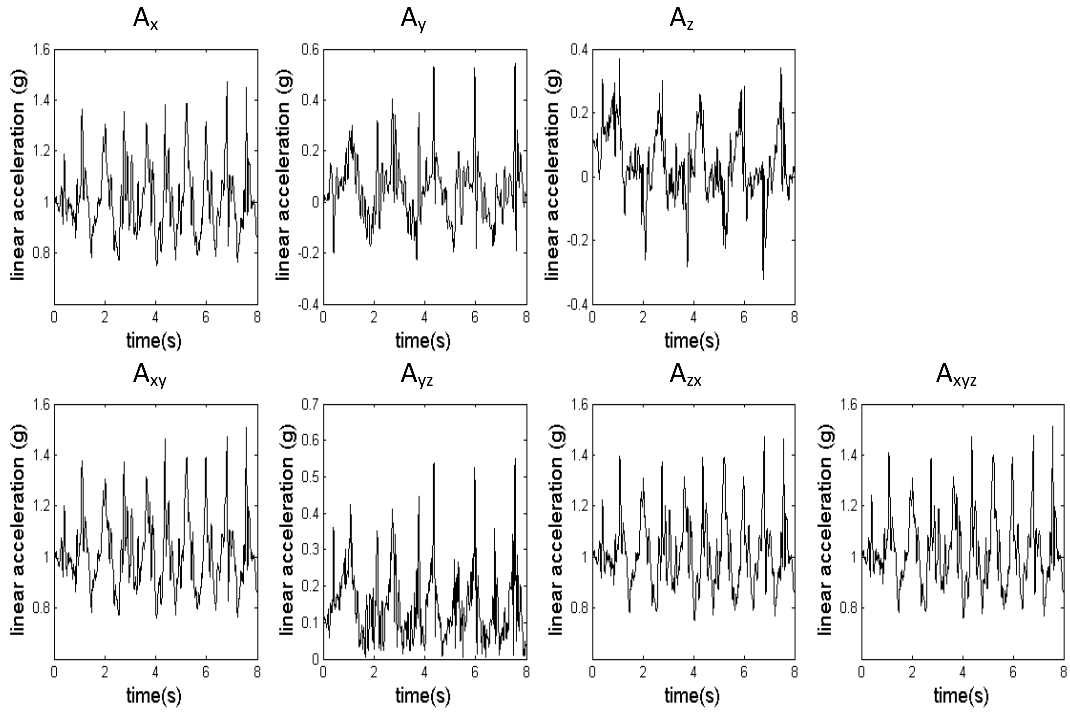


Figure E.7: Going upstairs: Accelerometer outputs and calculated magnitudes.

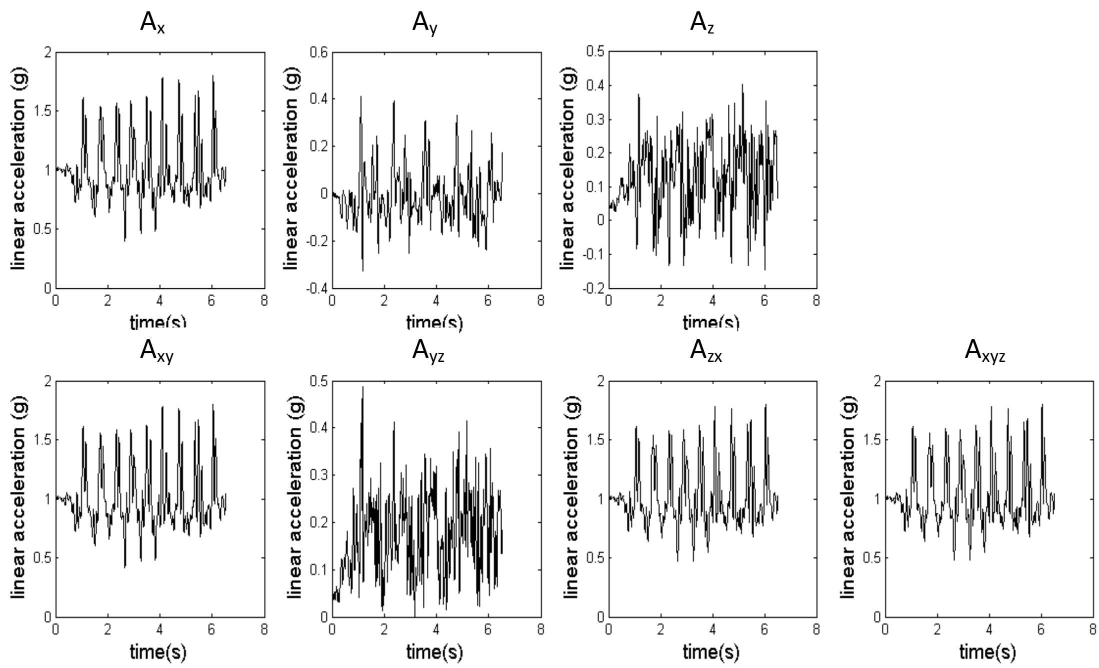


Figure E.8: Going downstairs: Accelerometer outputs and calculated magnitudes.

REFERENCES

[1] V. Adithya, "Motion and posture tracking for assisted living," *M. Tech. Dissertation,* Dept. Biosci. Bioeng., IIT Bombay, Mumbai, 2012.

[2] G. Welch, G. Bishop, L. Vicci, S. Brumback, K. Keller, and D. Coluccit, "The HiBall tracker: high-performance wide-area tracking for virtual and augmented environments," in *Proc. ACM Symp. Virtual Reality Softw. Technol.,* London, Dec. 1999, pp. 1-11.

[3] W. Korb, D. Engel, R. Boesecke, G. Eggers, B. Kotrikova, R. Marmulla, J. Raczkowsky, H. Worn, J. Muhling, and S. Hassfeld (2003, Apr.). Development and first patient trial of a surgical robot for complex trajectory milling. *Comput. Aided Surg.* [Online]. 8(5). 247-256. Available: http://www.ncbi.nlm.nih.gov/pubmed/ 15529954?dopt=Citation.

[4] T. Lee and A. Mihailidis, "An intelligent emergency response system: preliminary development and testing of automated fall detection," *Telemed and Telecare,* vol. 11, pp. 194-198, 2005.

[5] Vicon, "Case study: The dance of love", Available: http://www.vicon.com/company/ documents/Northumbria_owres.pdf, downloaded on 1st June 2012.

[6] O. Gabai and H. Primo, "Acoustic motion capture," (2011, Jan. 13). *Patent Application 20110009194* [Online]. Available: http://appft1.uspto.gov/netacgi/nph-Parser?Sect1= PTO1&Sect2=HITOFF&d=PG01&p=1&u=/netahtml/PTO/srchnum.html&r=1&f=G&1 =50&s1=2-0110009194.PGNR.

[7] X. Yun and R. Bachmann, "Design, implementation, and experimental results of a quaternion-based Kalman filter for human body motion tracking," *IEEE Trans. Robotics.,* vol. 22, no. 6, pp. 1216–1227, Dec. 2006.

[8] A. Gallagher, Y. Matsuoka, and W. T. Ang, "An efficient real-time human posture tracking algorithm using low-cost inertial and magnetic sensors," in *Proc. IEEE Int. Conf. Robot. Autom.*, Sendai, Japan, Sep. 28–Oct. 2, 2004, pp. 2967-2972.

[9] J. S. Wang, Y. L. Hsu, and J. N. Liu, "An inertial measurement unit based pen with a trajectory reconstruction algorithm and its applications," *IEEE Trans. Ind. Electron.*, vol. 57, no. 10, pp. 3508-3521, Oct. 2010.

[10] R. Zhu and Z. Zhou, “A real-time articulated human motion tracking using tri-axis inertial/magnetic sensors package,” *IEEE Trans. Neural Syst. Rehab. Eng.*, vol. 12, no. 2, pp. 295-302, Jun. 2004.

[11] M. Kangas, A. Konttila, I. Winblad, and T. Jamsa, “Determination of simple thresholds for accelerometry-based parameters for fall detection,” in *Proc. 28th IEEE Int. Conf. Engineering in Medicine and Biology Society*, Lyon, France, 13-28 Aug. 2007, pp. 1367-1370.

[12] A. K. Bourke, J. V. O'brien, and G. M. Lyons, "Evaluation of a threshold-based tri-axial accelerometer fall detection algorithm," *Gait & posture* 26.2, 2007, pp. 194-199.

[13] A. K. Bourke, and G. M. Lyons, "A threshold-based fall-detection algorithm using a bi-axial gyroscope sensor," *Medical Engineering and Physics* 30.1, 2008, pp. 84-90.

[14] J. Y. Hwang, J. M. Kang, Y. W. Jang, H. C. Kim, "Development of novel algorithm and real-time monitoring ambulatory system using Bluetooth module for fall detection in the elderly," in *Proc. 26th IEEE Int. Conf. Engineering in Medicine and Biology Society,* San Francisco, CA, 1-5 Sept. 2004,vol.1, pp.2204-2207.

[15] Q. Li, J. A. Stankovic, M. A. Hanson, A. T. Barth, J. Lach, and G. Zhou, "Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information," in *Proc. 6th Int. Workshop on Wearable and Implantable Body Sensor Networks,* Berkeley, CA, USA, 3-5 June 2009,pp. 138-143.

[16] M. Helmi and S. M. T. Almodarresi, “Human activity recognition using a fuzzy inference system,” in *Proc. 18th IEEE Inter. Conf. Fuzzy System,* Jeju island, Korea, 2009, pp.1897-1902.

[17] J. Y. Yang, Y. P. Chen, G. Y. Lee, S. N. Liou, and J. S. Wang, “Activity recognition using one tri-axial accelerometer: a neuro-fuzzy classifier with feature reduction,” in *Proc. Int. Conf. Entertainment Computing,* Shangai, China, 2007, vol. 4740/2007, pp. 395-400.

[18] C. Dinh, D. Tantinger, and M. Struck, “Automatic emergency detection using commercial accelerometers and knowledge-based methods,” *Comput. Cardiology,* vol. 36, pp. 485-488, 2009.

[19] InvenSense, “Motion processing unit product specification rev. 3.2, MPU-6000”, 16th Nov, 2011. [online]. Available: http://www.invensense.com/mems/gyro/documents/PS-MPU-6000A.pdf

[20] Microchip Technology, “PIC24FJ64GB004 family data sheet”, 15th July, 2010. [online]. Available: http://www.microchip.com/downloads/en/DeviceDoc/39940d.pdf

[21] Microchip Technology, “64-Mbit SPI serial dual I/O flash, SST25VF064C”, 11th June, 2011. [online]. Available: http://www.microchip.com/downloads/en/DeviceDoc/25036A.pdf

[22] Roving Networks, “Class 2 Bluetooth module version 1.08, RN-42”, 11th Aug, 2010. [online]. Available: http://www.rovingnetworks.com/resources/download/19/RN_42.pdf

[23] Microchip Technology, "MCP 1802", 8th Dec, 2006. [online]. Available: http://www.microchip.com/downloads/en/DeviceDoc/22053C.pdf

[24] Educational Control Products, "Model 750 Control Moment Gyroscope", 12th June, 2012. [online]. Available: http://www.ecpsystems.com/docs/ECP_Gyroscope_Model_750.pdf

ACKNOWLEDGEMENT

I am highly grateful to my guide Prof. P. C. Pandey for his guidance. I want to give thanks to Mr. Vidyadhar Kamble and Mr. Vismay Desai for their continuous support during the MTP. I want to convey my deep gratitude towards my lab mates in Electronic and Instrumentation Lab and Signal Processing Lab, and Mr. Sourav Chatterjee from Department of Aerospace Engineering. I am sincerely thankful to Ms. Geetanjali Shinde from WEL lab for her support and time. I dedicate my work to my loving family.

Praveen Kumar

Author's Resume

Praveen Kumar: He received B.E. in electronics and telecommunication engineering from College of Engineering Pravaranagar, University of Pune, Maharashtra in June 2010. Presently, he is pursuing M.Tech.in biomedical engineering at Indian Institute of Technology Bombay. His areas of interest include embedded system design, electronic system design, and VLSI design.