

## A Measurement System for Human Movement Analysis

*Master of Science Thesis*

**MUHAMMAD ATIF**

**SERKAN SERDAROĞLU**

Department of Signals and System

Division of Biomedical Engineering

CHALMERS UNIVERSITY OF TECHNOLOGY

Göteborg, Sweden, 2011

Report No. EX104/2011

A Measurement System for Human Movement Analysis

© Muhammad Atif, Serkan Serdaroğlu, 2012

Master's Thesis 2012:NN

Department of Signals and Systems  
Chalmers University of Technology  
SE-41296 Göteborg  
Sweden  
Telephone +46-(0)31 772 1000

Cover:

The cover shows average scaled data of triaxial accelerometer & gyroscope

Department of Signals and Systems  
Göteborg, Sweden 2012

# ABSTRACT

Parkinson's disease is a degenerative disorder of the central nervous system which significantly deteriorates the quality of life especially in the elderly. The prognosis of the disease includes impairment of movement related motions e.g. the gait, shaking of hands (tremor), rigidity, bradykinesia and akinesia, and postural instability etc.

The work included in this thesis focuses on developing a system to measure the degree of impairment with patients having this problem. The developed prototype has six degrees of freedom and is an inertial measurement unit to measure and quantify body movements, e.g. Tremor, which is the most well established symptom of this disease. The developed prototype was evaluated and the data acquired from the motion tests was displayed graphically through MATLAB. The prototype showed satisfactory performance and has potential to emerge as a tool in diagnosing and predicting the pace of the disease and a possible feedback system for rehabilitation of patients suffering with Parkinson Disease.

**Key words:** Parkinson's disease, MEMS, 6 DOF Digital IMU, Arduino UNO, MATLAB



# ACKNOWLEDGEMENT

All praise to Allah Almighty all knowing, most wise and perfect sustainer, with whose abundance and unlimited blessing we were able to complete our research thesis.

Deepest gratitude to our supervisor, Ants R Silberberg whose reliable and trustworthy knowledge become a source of enlightened path for us to proceed in the research area. Without his sincere, etch and genuine contribution it was hard to endure our achievements.

*ATIF.* I dedicate my work to my parents & family who always encouraged me to forward ahead in the race of life and excavate positives effects in my life. Especially my beloved mother to whom I love and respect from the depth of my heart. May Allah Almighty always bless her in this world and in the life hereafter (Amen).

I am also thankful to my friends who have been praying for my success and are always there for moral support in every aspect of life specially Ashar, Adnan, Saad, Danial , Umair.

*SERKAN.* First of all, my deepest appreciation to my parents who have always been standing there, supporting me with their pure love, encouraging me to continue my studies and assisting me whenever I needed. I know that I can never pay back what they have done for me. Special gratitude to my brother Serdar Serdaroglu who, all along my life, has been right beside me, taken care of me and my problems generously. Sincere hugs and kisses to my beloved little sister Zeynep Sevde for her unrequited love.

I would like to acknowledge my friends starting from Hasan Caylak; who has all the time guided me with his logical suggestions when I was down and sacrificed a lot for the sake of our friendship, with whom, Cappuccino and Franks Sessions accompanied with South American flavors will always remain as the best memories of Gothenburg days, Murat Haslak; with his honored and brave personality, has been a very good housemate for more than 2 years and helped me to get along with the life in Sweden, Jacobson; having purely devoted life and being perfect example to his students, has always been a good listener, activity organizer and a good friend. I am also thankful to other friends for their enjoyable company and ingenuous support; Ibrahim Avci, Serbay Ozkan, Ashar, Adnan, Umair, Atif.

At last but not least, I am profoundly grateful to the companion of my life Linda for feeling happy with my goodness and becoming worried about me whenever something unpleasant happens. Even though she has always been thousands of kilometers away from me, I have always felt her support, encouragement and pure dearness.



# Contents

1. INTRODUCTION.....	1
2. BACKGROUND.....	3
2.1 CAUSE.....	3
2.2 SYMPTOMS.....	3
2.3 TREATMENT TYPES.....	4
2.4 AIM OF THE STUDY .....	5
3. METHODS.....	6
3.1 PROTOTYPE DEVELOPMENT .....	7
4. HARDWARE SELECTION & SETUP.....	8
4.1 MEMS .....	8
4.2 ACCELEROMETER .....	9
4.3 GYROSCOPE .....	10
4.4 SENSOR SELECTION .....	12
4.5 ADXL345 ACCELEROMETER .....	14
4.6 ITG3200 GYROSCOPE.....	14
4.7 SELECTION OF MICROCONTROLLER BOARD.....	15
4.8 PROTOSHIELD.....	18
4.9 ASSEMBLY.....	18
5. SOFTWARE SECTION.....	20
5.1 ARDUINO.....	20
5.1.1 REGISTERS INITIALIZATION .....	20
5.1.2 WRITE FUNCTION .....	21
5.1.3 READ FUNCTION .....	21
5.1.4 DATA CONVERSION .....	22
5.1.5 SERIAL PORT COMMUNICATION .....	22
5.2 MATLAB .....	23
5.2.1 PORT INITIALIZATION .....	23
5.2.2 SAMPLING RATE .....	23
5.2.3 DATA HANDLING .....	24
5.2.4 GRAPH PLOTS .....	24
6. RESULTS AND DISCUSSION .....	26
6.1 Accelerometer .....	26
6.2 Gyroscope.....	28
7. CONCLUSIONS & FUTURE RECOMMENDATIONS .....	33

APPENDIX A .....	35
APPENDIX B.....	41
REFERENCES.....	45



# LIST OF FIGURES

Figure 1: PROTOTYPE SYSTEM ILLUSTRATION .....	7
Figure 2: Magnified picture of a microactuator <sup>[11]</sup> .....	8
Figure 3: Accelerometer mass on a spring model .....	9
Figure 4: Mechanism illustration of MEMS accelerometer <sup>[12]</sup> .....	10
Figure 5: Mechanical Gyroscope <sup>[13]</sup> .....	11
Figure 6: Gyroscope Model.....	11
Figure 7: Inside look at MEMS gyroscope <sup>[14]</sup> .....	12
Figure 8: IMU, Digital Combo Board (SEN-10121), top and back view <sup>[16]</sup> .....	13
Figure 9: ADXL345; Axes of Acceleration, Output response and Orientation to gravity <sup>[17]</sup> .....	14
Figure 10: ITG3200; Axes Orientation <sup>[18]</sup> .....	15
Figure 11: Arduino UNO top view <sup>[21]</sup> .....	16
Figure 12: Schematic of Arduino and Sensor Board Connection .....	17
Figure 13: Protoshield board and needed components for assembly <sup>[22]</sup> .....	18
Figure 14: Prototype of system.....	19
Figure 15: Protoshield, sensor board and Arduino Uno .....	19
Figure 16: Triaxial accelerometer raw data & scaled data illustration.....	26
Figure 17: Triaxial accelerometer averaged raw data & scaled data illustration .....	27
Figure 18: Triaxial gyroscope raw data & scaled data illustration.....	28
Figure 19: Triaxial gyroscope averaged raw data & scaled data illustration .....	29
Figure 20: Averaged scaled data of accelerometer & gyroscope .....	30



# 1.INTRODUCTION

Neurological diseases affect a vast amount of the elderly population in a negative way, hence resulting in a poor quality of life. Parkinson`s Disease is the most seen neurodegenerative disorder after Alzheimer`s Disease and it causes social, economical and emotional drawbacks with the aging population. These types of disorders are prevalently observed in people usually beyond the age of 55. According to the Population Reference Bureau 25 percent of European population will be over 65 in the year of 2030<sup>[1]</sup>. These facts reveal the need of optimizing the medical treatment and rehabilitation process.

To date Parkinson`s Disease (PD) has remained incurable. Few treatment types have been introduced with the expectation of curing the disease, yet none of them has been permanently effective on the symptoms. Conversely, some medical treatments have even led to adverse side effects. Only subjective and qualitative methods have been utilized in order to scale the degree of the disease which is very unstable as a response to the drug therapy and surgical procedures. At this point, the purpose of the thesis work is to develop a prototype that can sense and quantify the movement patterns of PD patients. The most common symptom of PD is tremor which is involuntary and rhythmic movement of the index finger and the thumb at resting state. Since tremor is observed in the majority of PD patients, the prototype has been designed in the most possible proper way to be attached on dorsum of hand and quantify the movement data temporally. The developed system offers objective and quantitative method to keep track on the PD patients during medical treatments. According to the movement data that the prototype provides; the efficacy of any applied medical treatment can possibly be observed and known, and the drug therapy and surgical procedures can favorably be optimized and arranged in an effective way leading to the best possible method of fighting the disease.

The system includes the implementation of a microcontroller and an inertial measurement unit (IMU) utilizing a triaxial accelerometer and a triaxial gyroscope; relatively measuring acceleration with respect to gravity and angular velocity. Communication between the IMU and the microcontroller is established and the data coming from the sensors are sent to the microcontroller for further processing. With the written program, data is converted into useful information and used as input for serial monitor (it is to be noted here that serial monitor is Arduino software window to display the data). Finally the data is taken from a serial port and processed in MATLAB to provide visual representation of movement patterns in real time for further analysis of clinicians.

## 2.BACKGROUND

Parkinson`s disease is the second most common neurodegenerative and progressive disorder which was initially discovered and characterized by James Parkinson in 1817. The annual incidence of 1/5000 makes it the most prevalent disease after Alzheimer`s disease. Nearly one million people are currently diagnosed with PD in the United States. The disease is usually seen in patients beyond the age of 55. Although the actual cause still remains unclear, the disease occurs due to the progressive degeneration of dopamine possessing neurons in the substantia nigra region located in midbrain. Dopamine is a chemical that is essential for cell communication for normal movement. To date the disease has been remained incurable <sup>[2]</sup>.

### 2.1 CAUSE

According to research on the reasons of developing Parkinson`s disease, only approximately 10 percent of the cases are caused by inherited dysfunction from former generations. The rest of the cases might be due to some unknown factors including an interaction between the environment and genetic susceptibility. These environmental factors incorporate chemicals namely; pesticides that play role in some cases, herbicides whose exposure shows 70 percent of increased risk, insecticides indicating 50 percent of increased risk, and insect killing chemicals which was associated with a 70 percent of increased risk of developing PD <sup>[3]</sup>.

### 2.2 SYMPTOMS

The Parkinson`s disease causes four cardinal symptoms namely; tremor, rigidity, bradykinesia and akinesia, and postural instability.

Tremor is the most common and apparent symptom which is developed in the majority of the patients as the disease progresses. It is maximal at resting state and under stress whereas it decreases or even disappears with voluntary movements and sleep. Tremor of the disease is described as “pill rolling” tremor that is back and forth movement of fingers. The reason why the term is given as pill rolling is the similarity of circular movement of index finger and thumb which almost seems like the former pharmaceutical technique of making pills. The frequency of pill rolling tremor is at 4 to 6 Hz corresponding relatively to 4 to 6 beats per second.

Increased muscle tone results in rigidity of body parts. It is a continuous and excessive contraction of the muscles. When the rigidity is in the facial muscles, it leads the face to have a mask like appearance and the expression is qualified by unblinking stare, wide eyed, and open mouth with involuntary drooling.

Bradykinesia is defined by slowness of movements with which, activities like cutting food, shaving, placing a key into a keyhole and buttoning a shirt become extremely more difficult when the disease progresses. When the disease becomes advanced due to the severely destroyed dopaminergic cells, difficulty of initiating movements arises. This symptom is named akinesia.

Postural instability appears in late stages of PD. The instability causes impaired balance that might lead patients to consecutive falls. This symptom is not observed in younger patients.

Other symptoms roughly include difficulty blinking and swallowing, impaired walking, shorter steps and shuffling, diminished arm swings, impaired speech, decreased range of motion and impaired hand writing <sup>[4, 5]</sup>.

## **2.3 TREATMENT TYPES**

Three types of treatment options for treating the Parkinson`s disease are drug therapy, deep brain stimulation and pallidotomy.

Drug therapy is administrated through injecting precursor of dopamine namely L-dopa which can pass the blood brain barrier. This drug is converted to dopamine subsequently ameliorating the syndromes of the disorder. However, as the efficacy of the therapy lessens adverse side effects develop. Dyskinesia is one side effect that causes impairment of voluntary movement.

Another treatment alternative is deep brain stimulation which involves surgical implantation of a medical device that transmits electrical pulses to brain`s particular regions such as sub thalamic nucleus and the globus pallidus internus. The device is named brain pacemaker comprising of internal pulse generator, extension cable and electrodes. Depending of the degree of the disease, modification can be applied to four parameters namely; electrode polarity, frequency, pulse width and amplitude.

Final procedure for amending some symptoms of PD is pallidotomy which is a surgical operation where an electrical probe is placed in globus pallidus and subsequently heated up to 80 degrees celsius for 1 minute. After this procedure a tiny scar is formed and the brain activity is reduced which somewhat helps relieve symptoms of dyskinesia, rigidity and tremor. This type of treatment is rarely used nowadays since brain stimulation is much less destroying and dangerous compared to pallidotomy <sup>[6, 7& 8]</sup>.

## **2.4 AIM OF THE STUDY**

A common method for assessing and analyzing the status of Parkinson`s disease is the Unified Parkinson`s Disease Rating Scale. The scale comprises of over 40 different inquiries to evaluate the patients` current degree of disease. The assessment is ordinal and interpreted qualitatively. Two problems arise in this method. The first one is that the interpretation can possibly vary from one clinician to another. The second problem is that since the scale is ordinal, it misses a temporal variable. On the other hand Parkinson`s disease possesses temporal attributes such as tremor.

A wearable system that is capable of providing objective quantification of PD attributes and also temporal measurement of tremor can be a key solution for addressing both issues. Such system can be beneficial for quantitative and objective feedback. With the help of this feedback, the selection and dosage of drug therapy and deep brain stimulation parameters can be optimized. Moreover the efficacy of both treatments could be monitored temporally.

Therefore, the aim of the thesis work includes the development of the prototype system. The developed system should comprise of a microcontroller board fused with a sensor board including triaxial accelerometer and triaxial gyroscope. The data coming from the sensors are then sent to a computer for further processing in MATLAB where the movement pattern illustration is performed.

### 3.METHODS

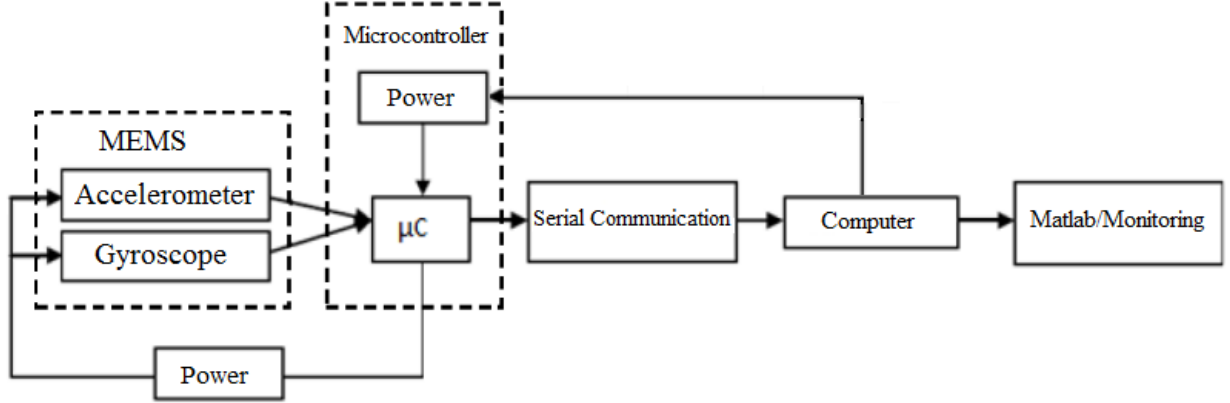
A thorough literature research has been carried out before stepping ahead into the problem and the solution states. First of all numerous available articles and books are read to investigate the Parkinson Disease with its symptoms. It is found that tremor as described in the introduction is the most common symptom of PD and it is observed in most of the patients. That is the reason why the focus is given to quantify the movement patterns of PD patients by designing a device that can measure motion characteristics of human hand.

As a next step, the research is focused on electronics namely Microelectromechanical systems (MEMS) and available microcontroller boards for constructing a favourable prototype system that can sense the tremor motion. A number of articles and research papers are available providing in-depth knowledge of MEMS in different applications. Although, there is a plenty of research regarding the sensors to capture the body movements of patients with neurological disorder e.g. Parkinson`s Disease, very few studies were found in regard to use these sensors as part of rehabilitation process of patients <sup>[9]</sup>.

Finally the hardware of the system is constructed and it is stepped into program writing for both the microcontroller and MATLAB. In this chapter, description of the prototype system with hardware and software sections is mentioned with details.



### 3.1 PROTOTYPE DEVELOPMENT



**Figure 1: PROTOTYPE SYSTEM ILLUSTRATION**

Above is the prototype system that has been developed for the purpose of quantifying movement patterns of PD patients. The system encompasses; MEMS triaxial accelerometer and gyroscope, a microcontroller board, a computer and MATLAB software.

As seen in Figure 1, the MEMS sensors accelerometer and gyroscope sense the motion patterns. After that the data is processed with the help of the written program for the microcontroller and input on serial monitor via USB communication. As a next step MATLAB program receives the data from a serial port and provides visual representation of the real time data in graphical form. The power needed for the microcontroller and the sensors is provided by the computer with the help of the USB connection cable. As a result, the system provides continuous and temporal real time data to be investigated by clinicians.

Under the prototype development section, hardware and software elements are introduced. Hardware is explained with initial theory behind the sensors, the figures of used components and the assembly of the whole system. In a final section, software is introduced with explanations of both the microcontroller and the MATLAB code.

## 4. HARDWARE SELECTION & SETUP

### 4.1 MEMS

The name of the technology differs depending on the location as it is known as Microelectromechanical systems (MEMS) in The United States and as Microsystems technology (MST) in Europe. There are many definitions of these systems. A broad definition might be; ` Portfolio or combination of techniques to design miniature systems which, ranging in size from sub micron to millimeter level, unite the mechanical and electrical components together in order to perform a particular duty by sensing, controlling and activating mechanical processes`<sup>[10]</sup>

The micromachining which, is the collection of fabrication and design tools that accurately form elements and machines at microscale, happened to be the toolbox of MEMS and foundation of micro fabrication that provided the production of Microelectromechanical systems. There are varieties of MEMS applications namely; inkjet printers, accelerometers, gyroscopes, microphones, pressure sensors, biosensors, chemical sensors, transducers, fluid pumps, blood pressure sensors and many more <sup>[10]</sup>.

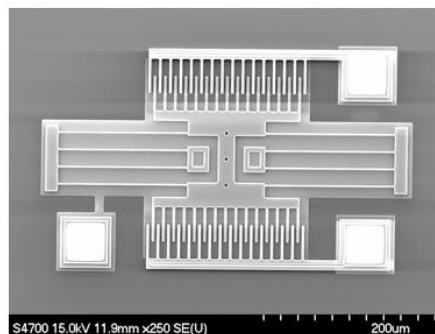


Figure 2: Magnified picture of a microactuator <sup>[11]</sup>

In the following sections, description and working mechanisms of accelerometers and gyroscopes are introduced.

## 4.2 ACCELEROMETER

Accelerometers are devices that measure proper acceleration with respect to the earth gravity. It is not the coordinate acceleration which is change of speed in space, but it is rather the acceleration that gives the orientation according to the weight provided by the gravity with quantity of  $9.8(\text{m/s}^2)$ . For example if an accelerometer stays still on a table, it gives  $1g$  ( $9.8\text{m/s}^2$ ). However when the accelerometer free falls towards the earth it gives  $0g$  because it is in a weightless state although its speed increases.

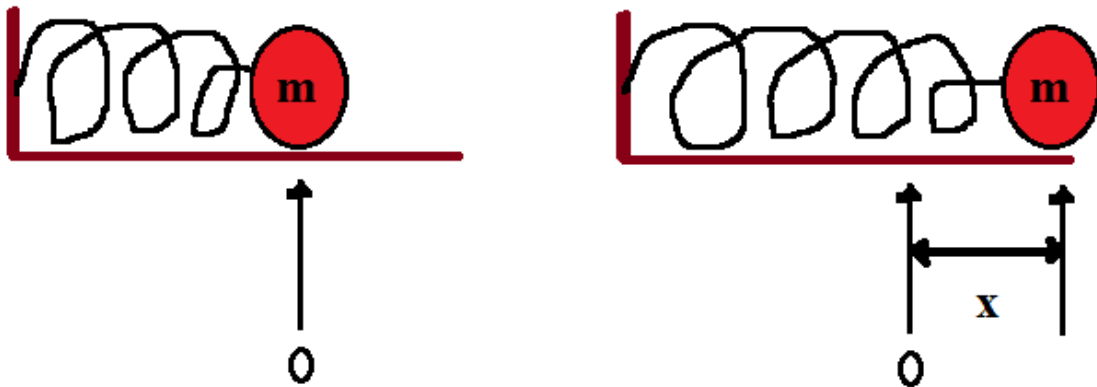
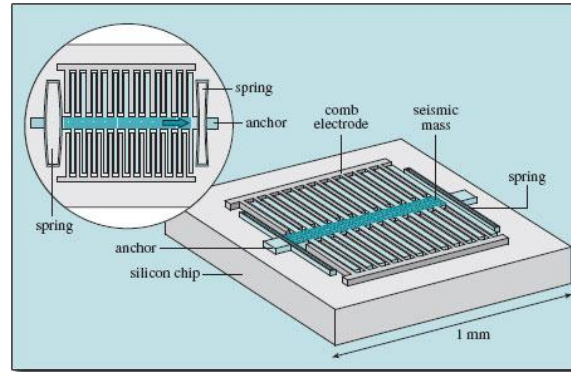


Figure 3: Accelerometer mass on a spring model

The function of acceleration can be explained with the Figure 3 above. The formula  $F = ma$  can be derived from Newton's second law where  $F$  is the force,  $m$  is the mass and  $a$  is the acceleration. Newton's formula is connected with Hooke's law is benefited with formula  $F = kx$  where  $k$  is spring constant and  $x$  is the displacement as seen above. By equalizing the formulas;  $ma = kx$ , hence it is found;  $a = kx/m$ . By doing these calculations, the problem of measuring the acceleration is transformed into measuring the displacement of its mass, which MEMS accelerometers can easily do.

Triple axis MEMS accelerometers are used to sense orientation, vibration, coordinate acceleration and shock by measuring magnitude and direction of the proper acceleration. MEMS accelerometers are used; in automotive industry for safety applications, braking systems, detecting vehicle acceleration, vibration on cars, in game controllers and smart phones to provide input for games and adjusting the screen orientation, in planes for sensing

the orientation, in military applications for weapon fire system, missile launch and rocket deployment, in laptops to protect hard drive at accidental drops, in human activities for measuring running, walking and dancing.



**Figure 4: Mechanism illustration of MEMS accelerometer** <sup>[12]</sup>

As seen in the Figure 4 above, MEMS accelerometers consists of a silicon chip on which the sensing structure is fashioned. Seismic mass region is placed between pairs of electrodes. When acceleration is applied to the sensor, the seismic mass moves towards the spring with respect to the amount of acceleration force. After that the electrodes detect this change and transmit by electronic means an electric signal representing the acceleration. The figure shows single axis accelerometer and the structure is replicated in multiple axis accelerometers.

### 4.3 GYROSCOPE

A gyroscope is a device that is capable of measuring the angular velocity, hence maintaining the orientation. There are mainly two kinds of gyroscopes namely mechanical and optical. In 1852, Leon Foucault invented the first gyroscope which consists of a spinning wheel mounted in a structure as seen in Figure 5 below. When the angular momentum of the wheel is high enough, the wheel fixes its orientation even if an external force is applied to the structure.

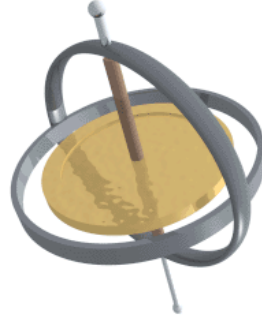


Figure 5: Mechanical Gyroscope <sup>[13]</sup>

The fact that mechanical gyroscopes are big in size and expensive is the reason why they are not used in small electronic devices like mobile phones. However in the last decade, MEMS gyroscopes have been introduced which are extremely small in size and very accurate for the applications of orientation and tilting. Many smart phones and some game consoles like Nintendo Wii use integrated gyroscopes.

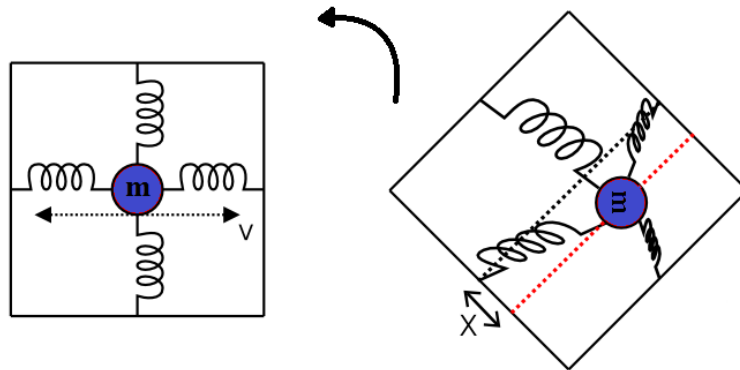


Figure 6: Gyroscope Model

A gyroscope can be modeled as it is depicted in Figure 6 above. The left picture shows a stationary gyroscope while the right one is the state after a significant rotation has been applied. When the device is rotated to the left the mass  $m$  is exposed to a specific force named Coriolis force which causes a second vibration force that is orthogonal to the direction of the first rotation. This is called Coriolis Effect. The second vibration is illustrated in the right image. Coriolis force is represented in the formula of;  $F_c = -2m\Omega \times v$  where  $m$  is the mass,  $\Omega$  is angular velocity and  $v$  is the velocity. Coriolis force is equal to the force explained under the accelerometer model as shown in the following;  $F = kx = F_c = -2m\Omega \times v$  from which

the angular rate can easily be calculated if the spring constant  $k$  and displacement  $x$  are known.

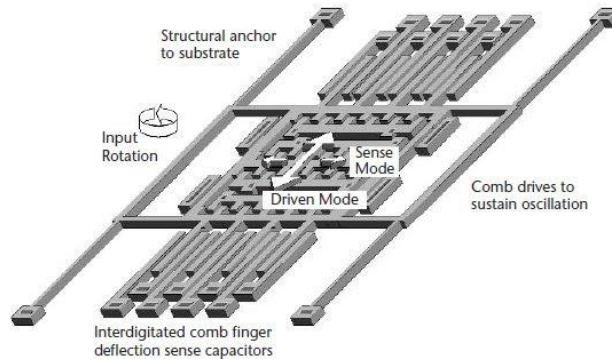


Figure 7: Inside look at MEMS gyroscope <sup>[14]</sup>

Figure 7 shows the inside of a MEMS gyroscope which incorporates; comb driver actuators that excite during rotation and causes oscillation along one axis, subsequently this oscillation excites a second motion along another axis. The second motion is deflected by comb drivers, which are another part of the gyroscopes. The deflection is detected by electronic means and transmitted to the output of the gyroscope.

## 4.4 SENSOR SELECTION

A range of sensors by different manufacturers are available in regard of accelerometer and gyroscope to be used for the development of prototype. For further details of sensors of same category refer <sup>[15]</sup>, where a thorough description of different accelerometers, gyroscopes and the combo boards is presented. After researching and documenting particular sensors that can be used for the implementation of the prototype IMU, Digital Combo Board-6 Degrees of Freedom ITG3200/ADXL345, a product (SEN-10121) by SparkFun was chosen and purchased. The reason why this IMU board, as seen in Figure 8, was used is because it is precise, powerful and widely used nowadays in many applications.

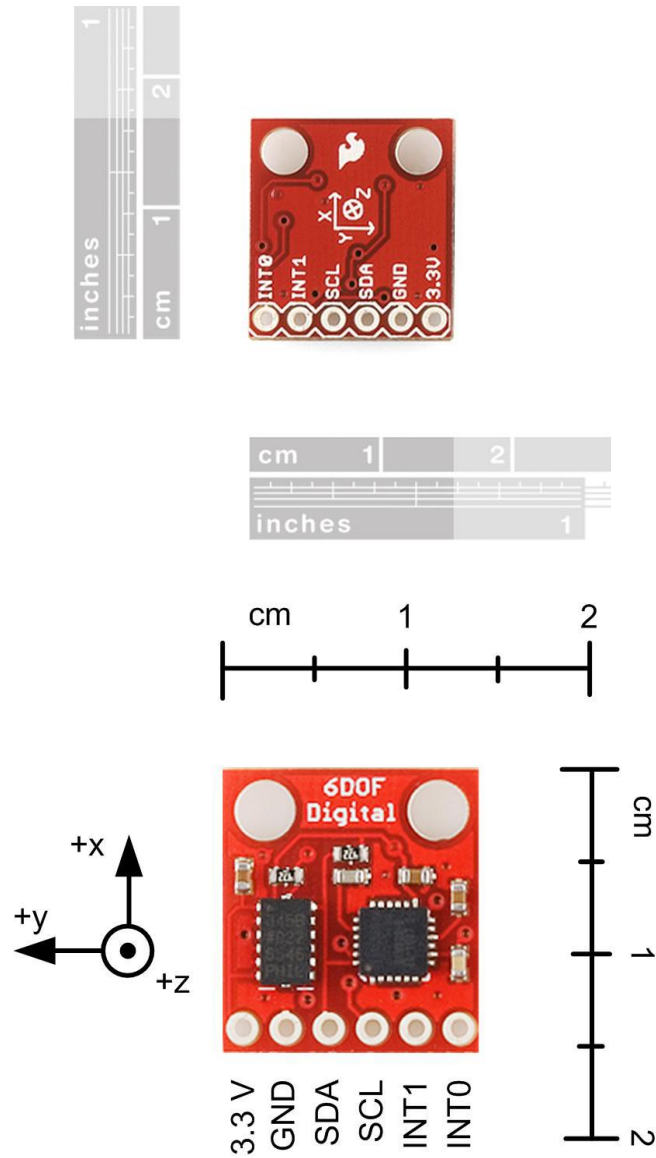


Figure 8: IMU, Digital Combo Board (SEN-10121), top and back view<sup>[16]</sup>

Both sensors that the board offers are digital meaning that they are more intelligent than analog sensors in terms for the capability of converting analog signals coming from their mechanical components into digital ones which are then accessible through a digital communication protocol. Both sensors communicate over the same data line (SDA) and clock line (SCL). There is no need for any external components since they are already built-in. Moreover, each sensor provides different configurable features such as sample rate, scale range, resolution, etc. The features of the board include; tiny size smaller than a coin, two mounting holes, triaxial ADXL345 accelerometer, triaxial ITG3200 gyroscope, 3.3V input and I2C interface.

## 4.5 ADXL345 ACCELEROMETER

ADXL345 is a triaxial and digital output MEMS accelerometer that is produced by Analog Devices. The sensor is very precise and of high quality. It comes with extremely good resolution and range up to  $\pm 16g$  and it is widely used in many different applications. Below is some important features of the sensor;

- X, Y, Z axis acceleration sensors on a single chip
- Low power consumption
- High resolution up to 13 bit with a range of  $\pm 16g$
- I2C and SPI interfaces
- Free fall detection
- Two different interrupt pins

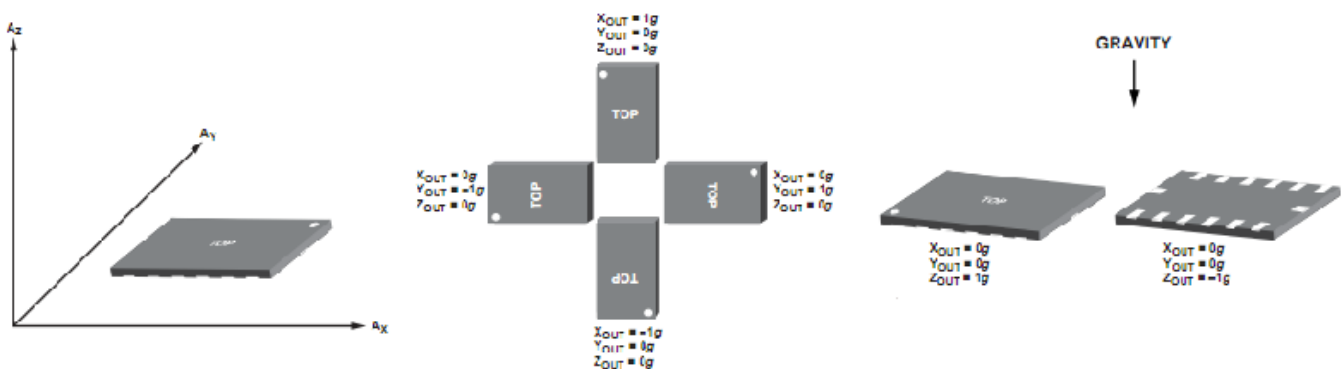


Figure 9: ADXL345; Axes of Acceleration, Output response and Orientation to gravity <sup>[17]</sup>

## 4.6 ITG3200 GYROSCOPE

ITG3200 is triaxial and digital output MEMS gyroscope that is produced by InvenSense. ITG3200 is the world's first single-chip, digital output gyroscope used in many motion based applications such as gaming and 3D mice. Important features are mentioned below;

- X, Y, Z axis angular velocity sensors integrated on one chip
- Configurable scale range of  $\pm 2000\text{deg/sec}$
- 16 bit ADC (analog to digital converter)
- Sensitivity of 14.375 LSBs per  $^{\circ}/\text{sec}$



- Configurable low pass filter
- Integrated temperature sensor
- I2C interface

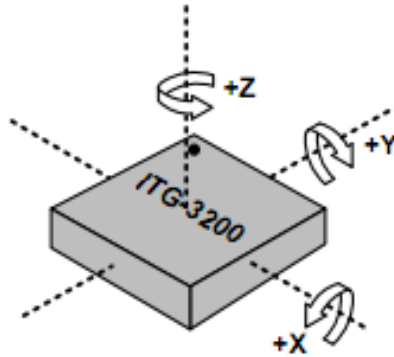


Figure 10: ITG3200; Axes Orientation <sup>[18]</sup>

## 4.7 SELECTION OF MICROCONTROLLER BOARD

There is variety of microcontroller boards available on the market with different sizes and features, for further details refer <sup>[19, 20]</sup>. Arduino UNO microcontroller board is selected, purchased and used for the implementation of the prototype. There are many available versions of Arduino board. The reasons leading to choose Arduino UNO are mentioned below;

- Cheaper in price (28 Euro) than many available boards on the market
- Free Arduino software available at the official website <sup>[20]</sup>
- Easy to use with USB communication
- Many available digital and analog (I/O) pins
- SPI and I2C serial interface
- PWM (pulse width modulation) pins
- Available online community providing numerous sample projects
- Smaller in size than many other versions
- Sophisticated enough as it is one of the last versions of Arduino products

There are many important components assembled on the board and most important ones are described below.

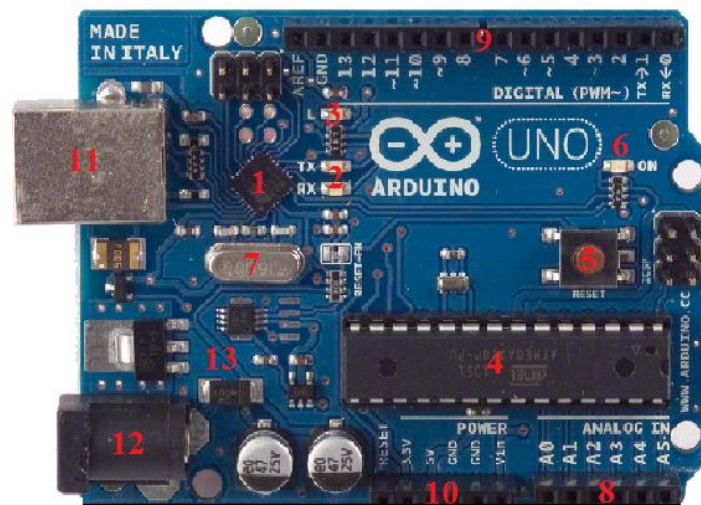


Figure 11: Arduino UNO top view <sup>[21]</sup>

- 1- ATmega16U2; enabling the board to communicate with computers via USB. The chip has a voltage regulator which converts 5V power coming from computer into 3.3V
- 2- TX and RX LED; indicator LEDs turning on when the communication is provided
- 3- Built in LED; connected to digital pin 13
- 4- ATmega328 ( $\mu$ C); brain of the board, comes with a boot loader allowing to upload codes (sketches as in Arduino language), having 32 KB flash memory
- 5- Reset; used for resetting the sketches
- 6- Power LED; turned on whenever the board is connected to any power supply including a computer
- 7- 16MHz Crystal Oscillator; generating the clock signal
- 8- Analog Pins; 6 analog input pins labeled from A0 to A5, each providing 10 bits (0-1023)
- 9- Digital Pins; 14 input/output digital pins of which 6 are PWM featured
- 10- Power Pins; including 1 pin of 5V, 1 of 3.3V, and 2 ground pins
- 11- USB Connection

12- Power Jack; power supply input

13- Components; voltage regulator, capacitors and diodes used for stabilizing power source, protecting from damage and smoothing signals

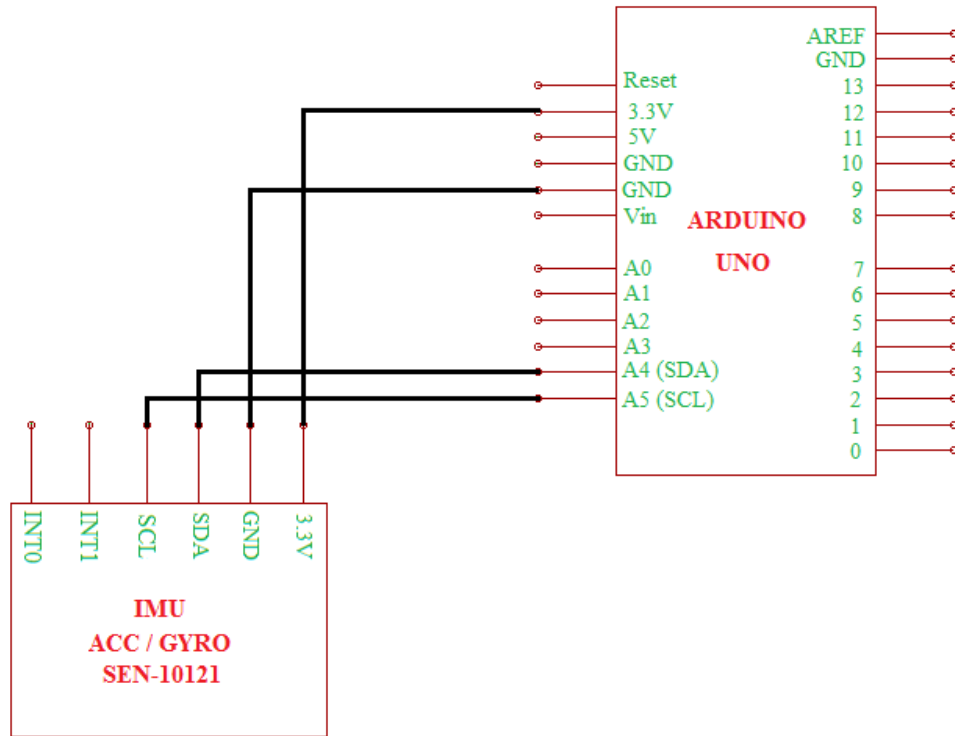


Figure 12: Schematic of Arduino and Sensor Board Connection

The circuit is built as shown in Figure 12. The IMU sensor board possessing accelerometer and gyroscope is fed with 3.3V power coming from an Arduino pin as it is depicted. There are three ground pins that Arduino UNO board provides. One of them is needed to connect to the sensors board's ground pin. The pins of A4 and A5 are SDA and SCL pins respectively both which provides the benefit of I2C protocol which is a two wired interface that allows easier communication. Normally pull up resistors should be connected between DC power and both SDA and SCL pins, however IMU (SEN-10121) has internal pull up resistors integrated. Pull up or pull down resistors are not in the scope of the thesis work but shortly they are used to define logic levels in circuits and to eliminate the noise that might interfere the output accuracy.

## 4.8 PROTOSHIELD

A protoshield is used to extend the accessibility of the Arduino UNO and to make the system compact. The protoshield is also needed to attach further components with the  $\mu$ C board. The sensor chip is attached over the protoshield that fits on the Arduino platform, which makes the system one unit. As shown in Figure 13, the protoshield needs to be assembled and the components should be soldered, before; it can be use with the Arduino UNO  $\mu$ C platform.

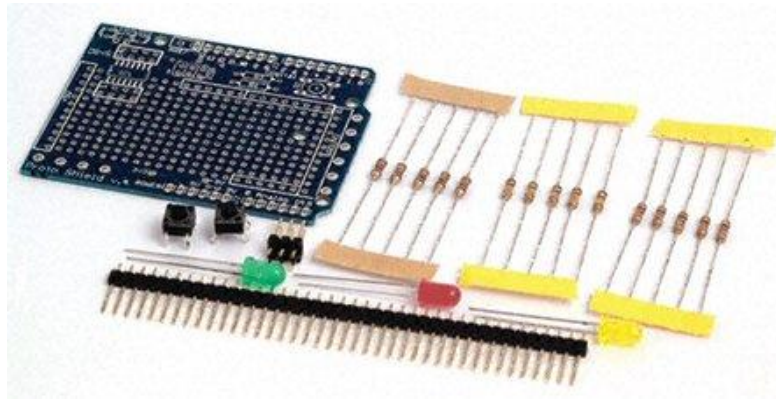


Figure 13: Protoshield board and needed components for assembly <sup>[22]</sup>

## 4.9 ASSEMBLY

The hardware and the components are essentially assembled and soldered before they can be functional. The sensor chip shown in Figure 8 is soldered with male headers and the components shown in Figure 13 are soldered on the protoshield to assemble it. Soldering and assembling of hardware is self done.

Accidentally the male headers are soldered on the sensor board inversely. It is observed that the Z axis of the accelerometer gave reverse output data. The z axis output data is then compensated with a slight change in the Arduino code because resoldering the male headers on the sensor board would be dangerous since the board is very tiny as seen in Figure 8.

Figures 14 & 15 show the hardware section of the prototype system after soldering and assembling.

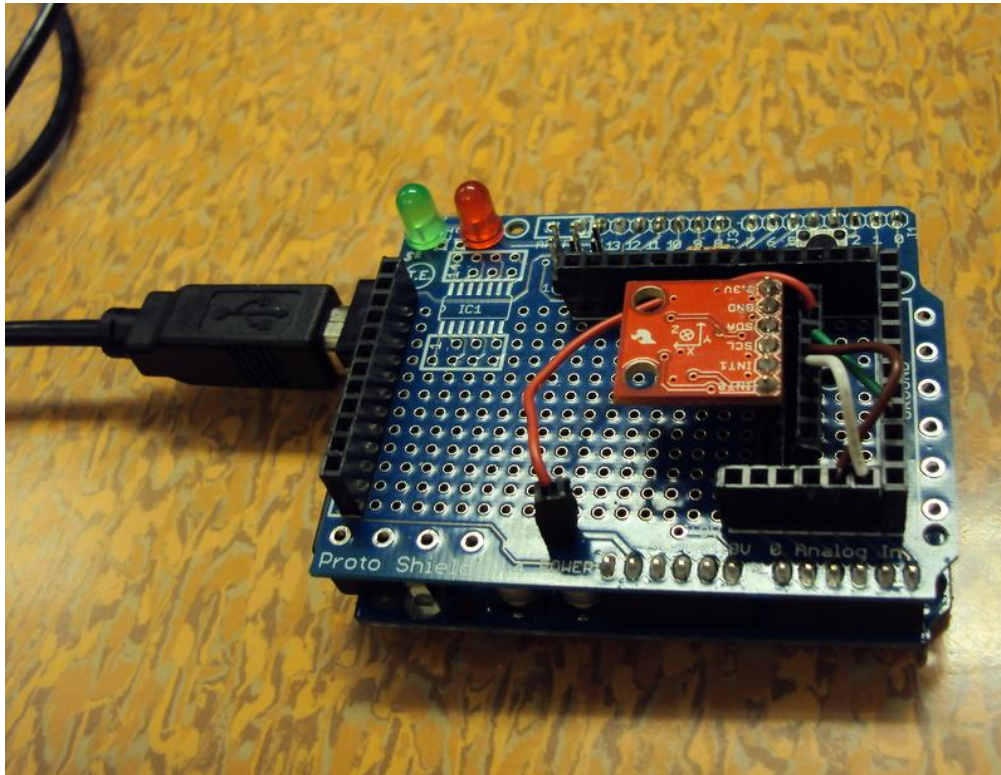


Figure 14: Prototype of system

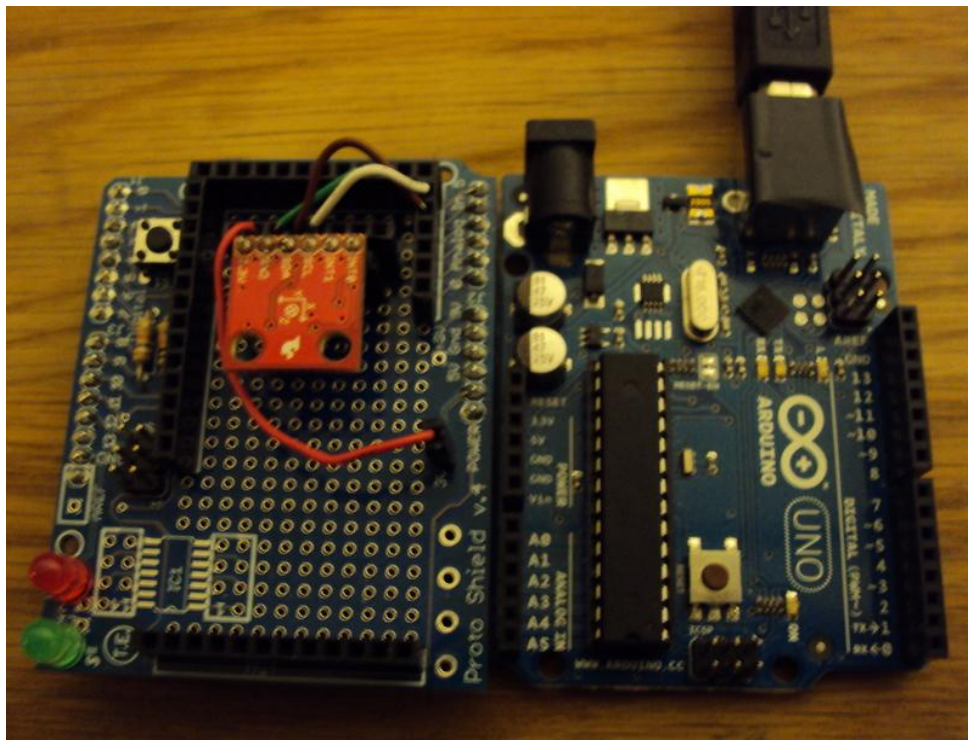


Figure 15: Protoshield, sensor board and Arduino Uno



## 5. SOFTWARE SECTION

### 5.1 ARDUINO

The main logic of the code that is written for the Arduino platform consists of essential parts that each of them performs a particular duty. Combination of these parts allows reading the data from sensors and eventually inputting them on the serial monitor for further processing in MATLAB which will be investigated in the next section. When constructing the code for Arduino, various electronics blogs and websites are utilized, for further details refer <sup>[23, 24& 25]</sup>

All the codes written under the software section are taken from the original code written for the thesis work. Below are the descriptions of some essential parts of the code namely; defining initial registers and addresses, write to function, read from function, buffers and conversion of data and finally inputting the data on serial monitor.

#### 5.1.1 REGISTERS INITIALIZATION

```
#include <Wire.h>           // I2C Library
```

In the beginning of the code (sketch in Arduino language), the library named “wire” is included which allowed the I2C communication with the sensor board as it is seen above. With the help of this library an easy way to communicate is established with two wires.

```
#define ACCELEROMETER (0x53)    // Accelerometer Address  
#define GYROSCOPE (0x68)       // Gyroscope Address
```

After that the ITG3200 gyroscope and the ADXL345 accelerometer addresses are defined according to the datasheets so that the data of accelerometer and gyroscope is distinguished and processing can be performed separately. Even though only one sensor board is used, two different sensors offering two different data type are utilized.

```
#define ACC_READ (6)            // 6 bytes used  
#define GYRO_READ (6)          // 6 bytes used
```

The next step is to define how many bytes to use for 3 data (X, Y, Z) for both accelerometer and gyroscope. Six bytes are used as seen above. Two bytes (16 bits) of storage are allocated for each axis or data.

### 5.1.2 WRITE FUNCTION

```
void writeTo(int SENSOR, byte REGADD, byte VALUE)      {
```

With the help of the wire library mentioned in the previous section, a write function is written. A write function is a very essential section of the code permitting the configuration of specific features of the sensors such as sample rate, range, low pass filter, etc. One example from the code is given below;

```
writeTo(GYROSCOPE, SAMPLE_RATE, 0x07);           // 125 Hz – 8mS
```

Calling the “write to” function allows configuration of the sample rate in the gyroscope. SAMPLE\_RATE is defined in the beginning of the code with its register address. By writing 7 in the sample rate register stated in datasheet of ITG3200, the sample rate is set to 125 Hz corresponding to 8 mS per sample. Any value from 1 mS to 250 mS can be set for the sample interval by just changing the hexadecimal value. Below are a few more examples of different sample rates;

```
writeTo(GYROSCOPE, SAMPLE_RATE, 0x13);           // 50 Hz – 20 mS
```

```
writeTo(GYROSCOPE, SAMPLE_RATE, 0x31);           // 20 Hz – 50 mS
```

```
writeTo(GYROSCOPE, SAMPLE_RATE, 0x63);           // 10 Hz – 100 mS
```

### 5.1.3 READ FUNCTION

```
void readFrom(int SENSOR, byte INITREG, int BYTES, byte BUFFER[])  {
```

The read function is the heart of the code. It allows reading the data coming from the sensors and making it possible to store the data in buffers. Calling this function makes it possible to begin for reading and processing of the data. One example from the code is given below;

```
readFrom(GYROSCOPE, register_1st, GYRO_READ, buffer);
```

```
int register_1st = (0x1D);
```

```
byte buffer [GYRO_READ];
```

The gyroscope address and gyro read are already mentioned and described in the beginning of this chapter. Register\_1st is defined as 0x1D which correspond the first register address of the X axis. So these three lines perform the following; in the gyroscope sensor starting from the first register, read 6 bytes of data by using the buffers. Buffers are explained under the next subtitle.

### 5.1.4 DATA CONVERSION

Since 2 bytes are used for each data, shifting is performed and then the data is converted from bytes to integer and also from raw to scaled data. Below is one example taken from the original code;

```
int GYROFFSET_X = -26;    // Offset X
float sens_cons = 14.375; // sensitivity constant as stated in ITG3200 datasheet
data[0] = (((buffer[0]) << 8) | buffer[1]) - GYROFFSET_X) / sens_cons; // Shifting performed
```

These lines perform storing data into buffers, making two buffers as one integer with shifting operation, adjusting calibration with offset value and finally converting the raw into scaled data. Every gyroscope sensor has got an offset value and it might differ from one sensor to another. It is found reading the data when the sensor stays still. In this case the offset value of X was -26 and it is subtracted from the read value so that it would give zero when the sensor is stationary. Another thing is the sensitivity constant which is used to convert the raw into scaled data. The range of raw data is [-28750 28750] is divided by 14.375 for obtaining the scaled range of [-2000 2000 (deg/sec)]. The final operation is shifting by using “<<” command. Since the output data is 16 bits for each axis of the gyroscope, each axis requires 2 bytes. Therefore shifting is performed so that two bytes can be stored in one integer. X data of the gyroscope is obtained via these three lines in the code.

### 5.1.5 SERIAL PORT COMMUNICATION

```
Serial.print(gyro_data[0]);
Serial.print("g");
Serial.print(gyro_data[1]);
```

The final part of the code includes sending the data on a serial monitor for observation and then the data can be used for MATLAB program. Data 0 is X axis data as it was stored in two buffers and converted to an integer value which are explained in the previous section. As seen above the “g” letter is also printed between two data namely X and Y. The reason is to distinguish the two data so that they can be read and plotted in MATLAB separately. This operation is explained under the MATLAB section with more details. The data type is float for more sensitive and accurate output. The serial print command is used to write the output data.

Datasheets of ITG3200 and ADXL345 are given as a reference to this chapter.



## 5.2 MATLAB

The MATLAB program is written for the purpose of obtaining the data from the serial port that the Arduino microprocessor sends in real time and plotting two graphs of both accelerometer and gyroscope data with different units. Essential parts of the MATLAB program includes setting communication and configuration, adjusting sample rate, reading and separating 6 data from each other, preparing the graphs and finally plotting the data.

### 5.2.1 PORT INITIALIZATION

```
s = serial ('COM8');  
set (s, 'BaudRate', 9600);  
fopen (s);
```

At first the communication with the computer port is established making the data transfer feasible. Here the used port is 8 for Arduino to send the sensor data to a laptop so it is chosen as the same for obtaining the data and processing in MATLAB. As a next step, the baud rate is specified as 9600 meaning that maximum of 9600 bits can be transferred per second. This rate is the same for the Arduino code. Finally, the serial port is opened with the last line. It is necessary to start the program with these lines for performing initializations.

### 5.2.2 SAMPLING RATE

```
SR = 0.155;           // Sample rate in MATLAB
```

Sample rate is set depending on the Arduino code which has got two different sample rate configurations. One is the sample rate of the sensor read data which is explained under the Arduino section 5.1.2 and another one is the extra delay function used to give any delay by choice. Addition of these two configurations equals to the sample rate that ought to be defined in MATLAB. In this program the sample rate is 0.155 seconds per sample. The delay was 0.12 seconds and the read data sample rate was 0.035 seconds. The sample rate is very necessary to define for building up the graph temporally while the data is plotted in real time. In the case that the sample rates clash, the timing would not give correct result.

### 5.2.3 DATA HANDLING

```
str = fscanf(s);
```

Six data values flow on a serial monitor continuously. What this command does is that it reads all the values in a string. As it is stated in the Arduino section, between the data there are some letters just to allow MATLAB to distinguish all 6 data.

```
index1 = find(str == 'g');
```

```
str1 = str(1:index1-1);
```

```
Gx(i) = str2double(str1);
```

Taking into account of only two data flowing on the serial monitor; between these two data there is the letter of “g” which equals to index1 as shown above. As a second step, str1 is equal to the string starting from the first digit to one left digit of “g”. In this way the first data is separated from the second one. This procedure is replicated for every data and in the end there are 6 data obtained one by one separately. Finally the first data (str1) is converted to a number from string by using the str2double command.

### 5.2.4 GRAPH PLOTS

```
subplot(2,1,2),title('Gyroscope Data'),
```

```
ylim([-2000 2000]),xlabel('Time'),ylabel('Deg/sec');
```

The gyroscope graph features are adjusted with the code above. Subplot (2,1,2) specifies the location of the graph in the window (second row and first column). The title and the range of Y coordinate are set. Since the X coordinate is continuous and it varies by time, no fixed range is given. Finally the names of the parameters of X and Y axis are defined.

```
i = 20;
```

```
plot(SR.*(i-10:i),Gx(i-10:i),'--b','LineWidth',2)
```

```
i=i+1;
```

```
drawnow;
```

The defined integer “i” is the data number to be plotted. The longest line above is the main line that plots the data in X and Y coordinates. X is the continuous time that is calculated by multiplying the sample rate with the string of data starting from the 10th data and always storing the last 10 data. The fundamental logic is that;  $time = (sample\ rate) \times (quantity\ of\ data)$  which gives exact and accurate timing. The Y axis is constructed by using the G (i)

operation which is already read and separated from the other data as explained before. As in X axis case, Y axis also stores the last 10 data. The main reason of removing the older data is not to store too much data that might slow down the flowing of graphs. There is another reason that is the flow of the data which corresponds to the sample rate; if the sample rate is less the flow of data is too much which might slow down the plotting of graph. These results are investigated under the results section. Since the data is expected to be in real time, a loop is attached to encompass plot command with increasing data quantity as indicated above. Finally the drawnow command starts drawing the graphs.

## 6. RESULTS AND DISCUSSION

In this chapter, various graphs of accelerometer and gyroscope data are illustrated and evaluated. The graphs are obtained in MATLAB and the data is sent to the serial port of a computer from an Arduino microprocessor board.

### 6.1 Accelerometer

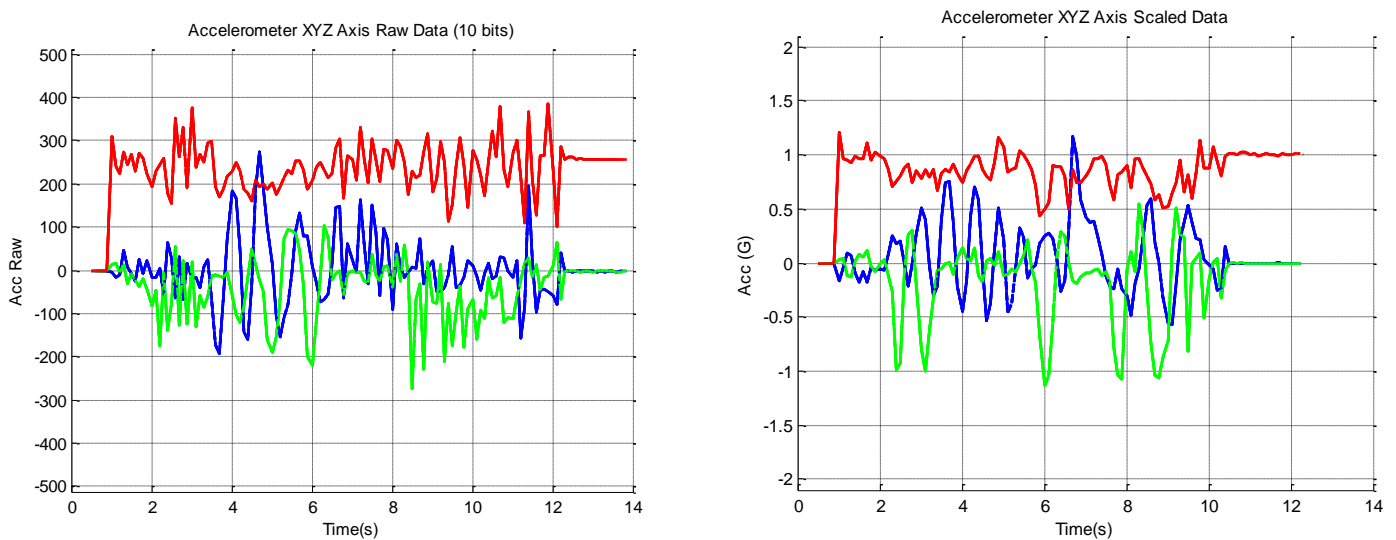


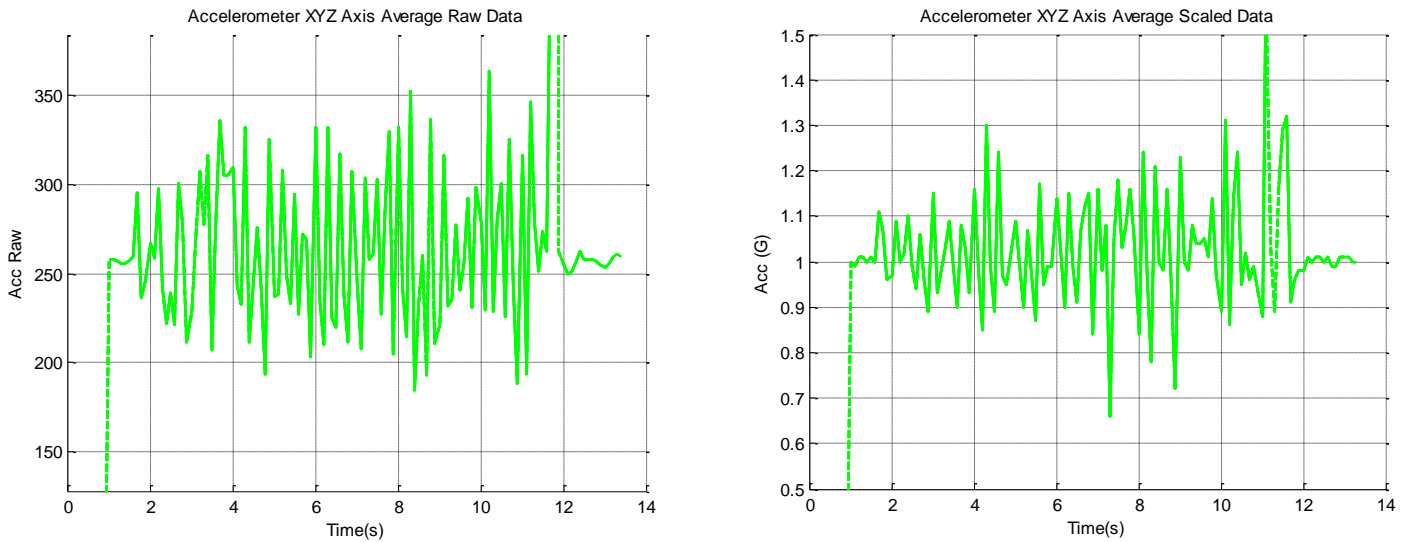
Figure 16: Triaxial accelerometer raw data & scaled data illustration

Triaxial raw data is illustrated in Figure 16 (graph on the left). Data come in 10 bits with the range of (-512 512). The red line is Z, blue is X and green is Y axis. Random orientation is given to the system and quantification of fluctuations is observed as a response. The sample rate of the system is 10 Hz corresponding to 10 data per second or with another means; 0.1 seconds between every sequent data. When the prototype system is stationary as the top of the sensor faces upward, Z axis (red line) stays on 256 while X and Y axes (blue and green respectively) give the value of 0. For general output response and orientation to gravity, please see Figure 9.

Triaxial scaled data is illustrated on Figure 16 (graph on the right). Similarly the red line is Z, blue is X and green is Y axis. Data comes in 10 bits and subsequently it is converted into scaled data by using the scale constant. Multiplying the raw data by 0.0039 or dividing it by 256, permit obtaining scaled data. If the raw data of Z axis is 256, it is 1g in scaled data

means. The reason why scaled data is obtained is because scaled data is the exact unit (1g = 9.8 m/s<sup>2</sup>) of accelerometer so the interpretation of the data is performed much easier. The sample rate of the system is similarly 10 Hz corresponding to 10 data per second or with another means; 0.1 seconds between every sequent data. It should be noted that raw and scaled graphs in Figure 16 are not obtained from the same motion test.

Magnitude and frequency characteristics of PD tremor are introduced and examined in the end of this chapter. The prototype system ought to offer favorable sampling frequency and magnitude range so that the tremor of even advance PD patients can be quantified. In addition when a number of PD patients` tremor is interpreted and quantified with the offered prototype system, it will be easy to classify the degree of tremor behavior.



**Figure 17: Triaxial accelerometer averaged raw data & scaled data illustration**

The raw and scaled triaxial data are averaged and converted into 1 data per each graph. The simple averaging formula of;  $A_v = \sqrt{X^2 + Y^2 + Z^2}$  is used.

The result is never negative as seen on the graphs in Figure 17. It should again be noted that the average raw and scaled graphs are not obtained from the same motion test. This method of averaging offers easy interpretation and classification opportunities. In addition Fast Fourier Transform can be applied to acquire amplitude and frequency graph to assess tremor with further aspects.

## 6.2 Gyroscope

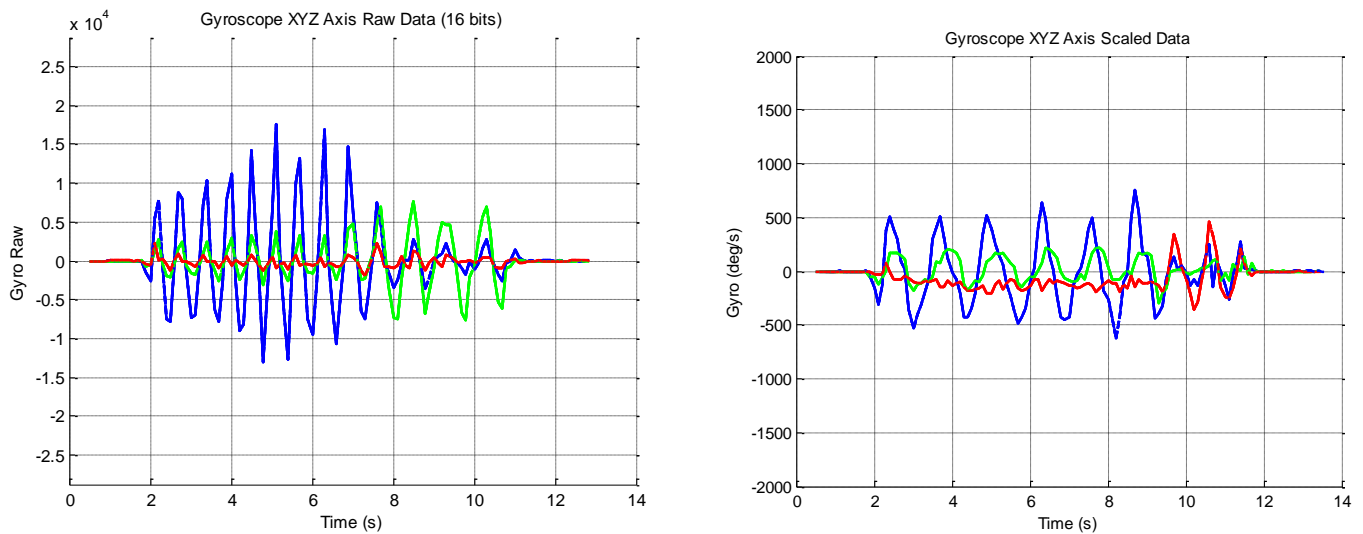


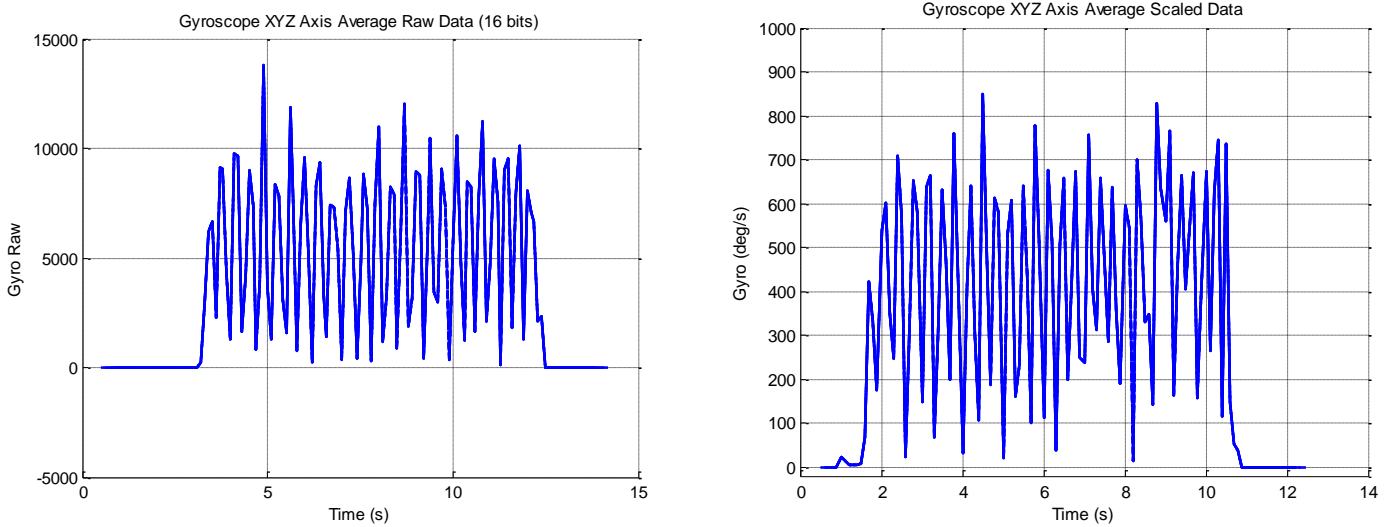
Figure 18: Triaxial gyroscope raw data & scaled data illustration

Triaxial raw data is illustrated on Figure 18 (left graph). Data come in 16 bits with the range of  $(-28750 \ 28750)$ . The red line is Z, blue is X and green is Y axis. Random orientation is given to the system and quantification of fluctuations is observed as a response. The sample rate of the system is similarly 10 Hz corresponding to 10 data per second or with another means; 0.1 seconds between every sequent data. When the prototype system is stationary as the top of the sensor faces upward, all the axes give the value of 0 since the gyroscope values do not depend on the earth gravity. For general axes orientation, please see Figure 10.

As distinct from accelerometer, every gyroscope sensor possesses an offset value for all the axes. Therefore before beginning to read data from the sensor, an offset calibration is performed. Offset values are obtained by reading data out of the sensor when it is at rest. Subsequently the offset values are subtracted from the read data so that the sensor gives all 0 values when it is stationary. For a more detailed explanation please see the software chapter.

Triaxial scaled data is illustrated in Figure 18 (graph on the right). Similarly the red line is Z, blue is X and green is Y axis. Data comes in 16 bits and subsequently it is converted into scaled data by using the scale constant quantity stated in the ITG3200 datasheet. The scaling is simply performed by dividing the raw data into the sensitivity constant which is 14.375. The unit of the scaled data is degrees over second which gives the angular velocity with

respect to time. When the sensor rotates around X axis coordinate, the X data increases or decreases depending on the direction of the rotation. Similarly Y and Z axes change when the system is rotated around Y and Z coordinates respectively. The range of the gyroscope being (-2000 2000) deg/sec is favorably sufficient for quantifying hand motions of PD patients. It should be noted that raw and scaled graphs are not obtained from the same motion test.



**Figure 19: Triaxial gyroscope averaged raw data & scaled data illustration**

Similarly the raw and scaled triaxial data are averaged and converted into 1 data per each graph. The simple averaging formula of;  $A_v = \sqrt{(\bar{X}^2 + Y^2 + Z^2)}$  is used.

The orientation is altered by shaking the prototype repeatedly and the graphs seen above in Figure 19 are acquired. As seen on the scaled graph, maximum amplitude is 850deg/sec. It is observed that even under intense motions, it is almost impossible to obtain 2000deg/sec. The sampling frequency is 10 Hz which is favorable quantity for quantifying the tremor of PD patients since the tremor range is 4 to 6 Hz.

The result is never negative as seen on the graphs in Figure 19. It should be again noted that the average raw and scaled graphs are not obtained from the same motion test. This method of averaging offers easy interpretation and classification opportunities. In addition Fast Fourier Transform can be applied to acquire amplitude and frequency graph to assess tremor with further aspects.

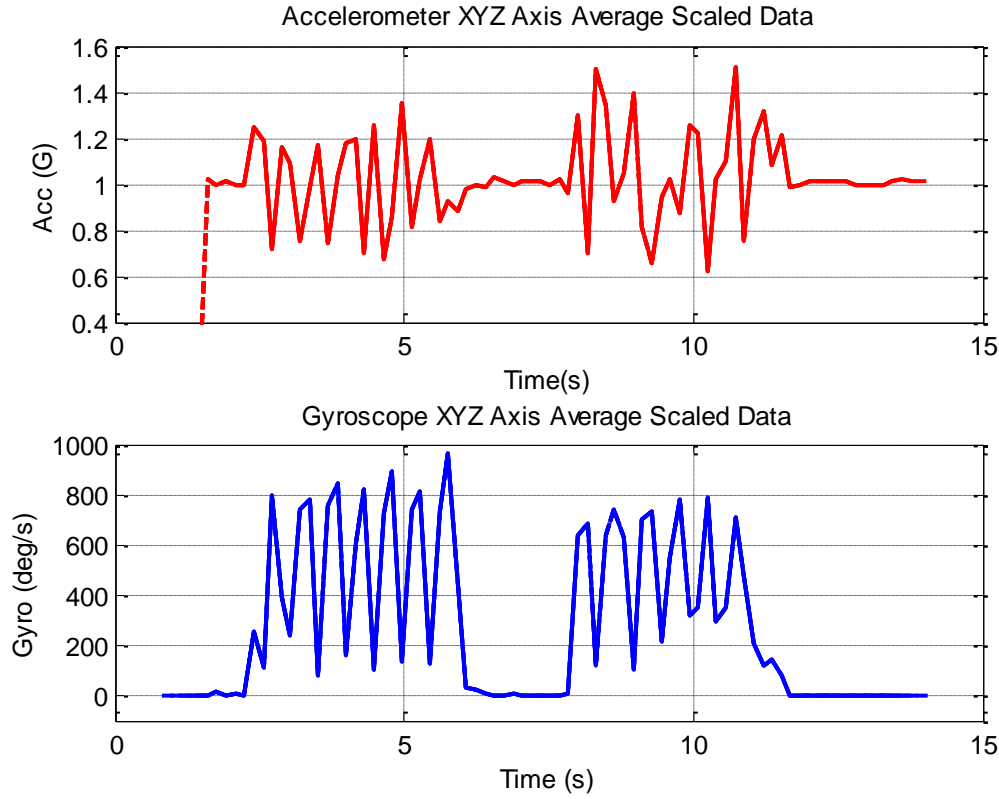


Figure 20: Averaged scaled data of accelerometer & gyroscope

Finally both averaged scaled accelerometer and gyroscope graphs are fused and acquired simultaneously as seen in Figure 20. The prototype system offers satisfactory gyroscope and accelerometer data at the same time and providing 6 degrees of freedom. Even though the data coming from the sensors are satisfactory, some limitations are observed. The main limitation is the sampling frequency. The graphs are obtained with the internal sampling frequency of 6.25 Hz whereas the PD tremor range of (4 - 6) Hz. It is important to be noted here that the result showed is purely a functionality testing of the hardware and this particular result cannot be used for the capturing of tremor in PD patients. As according to the *Nyquist-Shannon sampling theorem* the sampling frequency ( $f_s$ ) should be at least double the highest frequency component ( $f_{\max}$ ) in the signal which means  $f_s > 2f_{\max}$  <sup>[26]</sup>; as in this case the max. frequency of tremor is between (4-6) Hz and we are sampling the signal at 6.25Hz which does not satisfy the condition of the sampling theorem. However, when the both data is being plotted in real time, after 40 seconds the graphs slow down and become less sensitive to motion. The reason of this is thought to be either the insufficient processing power of laptops being used or excessive filling of MATLAB memory due to plotting 2 data simultaneously in real time. Therefore to overcome this problem, remainder logic is added in the MATLAB



program. The remainder is aimed to close and open the graph window continuously to refresh the memory of MATLAB and allow a continuous and sensitive data flow. At first remainder is equalized to a quantity which is the data number. After that when the loop is in action and the remainder number is reached, the systems closes itself and subsequently opens and continue plotting the graph from where it stops before. The code is given below.

```
if (rem (i,200) == 0)
    close;
end
```

In order to find how many seconds the window stays open, data number (remainder) is multiplied by the sampling rate as shown below.

$$\text{Sampling Rate} = 1 / \text{Sampling Frequency}$$

$$T(\text{seconds}) = \text{Remainder} \times \text{Sampling rate}$$

When the sampling frequency is increased from 6.25 Hz to 10 Hz, the system becomes much more sensitive as it can detect more data per second. On the other hand the flow of data on the graph slows down even earlier than 40 seconds which prevents clear observation. Therefore there is inverse proportion between sampling frequency and remainder. When the sampling frequency ascends the remainder should be defined less in quantity for stable plotting.

Another issue to be examined is the amplitude of the PD tremor. Even though many accelerometer studies have been performed to quantify the amplitude of the tremor signal, very few gyroscope included studies are introduced. According to an accelerometer study, it is offered that the PD tremor has got an amplitude range of [0.7 1.3] g <sup>[27]</sup>. Another study suggests less range as [0.9 1.1] g <sup>[28]</sup>. Considering both studies, the accelerometer that the prototype system uses possesses a favorable amplitude range for quantifying PD tremor.

On the other part, the gyroscope with its range of [-2000 2000] deg/sec provides without doubt a sufficient amplitude range. According to one study, the amplitude of PD tremor has been quantified as [-600 600] deg/sec which is less than even half the range of the gyroscope the prototype provides <sup>[29]</sup>.

All in all, the designed prototype, with its sampling frequency limitation (considering the graphical display of results), has got favorable amplitude ranges and accurate readings which make it a feasible system having the potential to emerge as a tool for diagnosing and

predicting the pace of the disease and as a possible feedback system for rehabilitation of patients suffering with Parkinson`s Disease.

## 7. CONCLUSIONS & FUTURE RECOMMENDATIONS

A prototype system was successfully developed and tested using a triaxial accelerometer (ADXL345) and a triaxial gyroscope (ITG3200) interfaced with an Arduino Uno (microcontroller board). The sensor data is interpreted in graphical form using serial communication between hardware and a host PC.

Several technical challenges including the selection of sensor board, soldering & assembling of protoshield were resolved during the development of the prototype system. One challenge was the communication between the hardware parts, and the solution was found in using I2C communication between sensor board and  $\mu\text{C}$  by defining the addresses of both sensors. Another problem was interpretation of real time sensor data in graphical form, Matlab was chosen as a possible solution among other software tools to address this issue.

The prototype system is capable of measuring triaxial acceleration data between range of  $\pm 2g$  and triaxial gyroscope data between  $\pm 2000$  deg/sec. The system has been compensated for the 0'g values of accelerometer and drift in values of gyroscope. The system has a limitation of displaying graphical data for longer period of time, the graphical changes tends to appear slow after certain period of time. This problem mainly appeared because of processing limitations of the host PC, and is, resolved apparently by automatic shutting the graphical window of Matlab after a certain time in a continuous looped program. Apart from that, the system showed satisfactory performance.

This work provides a basic platform for future developments to emerge as a potential tool in diagnosing and predicting the pace of movement disorder diseases.

The functionality of the system can be improved with the data fusion of both sensors. The mathematical expressions can be brought into the software program to get the Euler angles and representing six degree of freedom in angular form.

Fast Fourier Transform (FFT) can be applied to visualise the frequency range of hand movements and tremor in real time.

The system can be integrated with the virtual reality gaming console, such as, Nintendo Wii to make a possible feedback system, allowing the patients to play interactive games in a rehabilitation process of patients suffering with Parkinson Disease. Furthermore, the work can also be extended to make a portable system with wireless transmission.

The prototype system can be utilized to cooperate with a brain stimulation device. With improved features such as wireless transmission, data logging and designed wearable fashion, the prototype can be used as a feedback system for the brain pacemaker. In case the communication between systems are established, 4 parameters of brain pacemaker namely electrode polarity, frequency, pulse width and amplitude could be simultaneously optimized with the help of the continuous quantification of tremor patterns of PD patients.

# APPENDIX A

## Arduino Uno

/\*\*

The program is written on November 2011 for the thesis with the topic of "A measurement system for human movement analysis" performed by Muhammad ATIF and Serkan SERDAROGLU. The program mainly focuses on reading accelerometer and gyroscope data in real time. Thesis aims to develop a quantitative method to assess tremor characteristics of Parkinsonian patients. To date there only exists the quantitative method named Unified Parkinson`s Disease Rating Scale (UPDRS) which simply based on 40 inquiries to be asked to patients. With this new method the diagnosis will be more accurate and depending on the temporal movement data, drug therapy and brain stimulation parameters could be optimized.

\*/

```
#include <Wire.h>          // I2C Library
```

```
// ADXL345
```

```
#define ACCELEROMETER (0x53)  // Accelerometer Address
```

```
#define ACC_READ (6)          // 6 Bytes are going to be read for x, y and z axis (2 bytes  
                               per each)
```

```
// ITG3200
```

```
#define GYROSCOPE (0x68)      // Gyroscope Address (AD0 hooked up to ground)
```

```
#define GYRO_READ (6)         // 6 Bytes are going to be read for x, y and z axis (2 bytes  
                               per each)
```

```

#define POWER_MNG (0x3E)          // *****

#define INTERRUPT_CONF (0x17)     // Defined registers based on ITG3200 datasheet

#define SAMPLE_RATE (0x15)       // See Page 22

#define FS_DLPF (0x16)           // *****

void startACC()

{

writeTo(ACCELEROMETER, 0x2D, 0);  // Accelerometer is turned on, range is -2g to 2g
                                // by default

writeTo(ACCELEROMETER, 0x2D, 16);

writeTo(ACCELEROMETER, 0x2D, 8);

}

void getACC(float * data)

{

int register_1st = (0x32);       // First register of all axes

byte buffer [ACC_READ];         // Buffer for storing data

float scale_cons = 0.0039;       // Constant used for converting data into scaled

// Read the accelerometer data 6 bytes starting from the register_1st

readFrom(ACCELEROMETER, register_1st, ACC_READ, buffer);

data[0] = (((buffer[1]) << 8) | buffer[0]) - 20) * scale_cons;  // Data comes 10 bits

data[1] = (((buffer[3]) << 8) | buffer[2]) - 16) * scale_cons;  // Shifting is performed

data[2] = (((buffer[5]) << 8) | buffer[4]) * -scale_cons;      // Conversion from bytes to integer

```

```

data[0] = sqrt(sq(data[0])+sq(data[1])+sq(data[2]));    // Data is averaged
}

void startGYRO()

{

writeTo(GYROSCOPE, POWER_MNG, 0x00);        // Powers the sensor (Internal
                                           Oscillator)

writeTo(GYROSCOPE, INTERRUPT_CONF, 0x00);    // We do not use interrupt operation

writeTo(GYROSCOPE, SAMPLE_RATE, 0x07);      // 125 Hz sample rate (every 8 ms)

writeTo(GYROSCOPE, FS_DLPF, 0x1D);          // Range +-2000 deg/s / 10 Hz low pass / 1
                                           KHz internal sample rate

}

void getGYRO(float * data)

{

int register_1st = (0x1D);                  // First register of all axes (no temperature registers)

byte buffer [GYRO_READ];                   // Buffer for storing data

// Read the gyroscope data 6 bytes starting from register_1st

readFrom(GYROSCOPE, register_1st, GYRO_READ, buffer);

int GYROOFFSET_X = 25;                      // *****

int GYROOFFSET_Y = 29;                      // Gyroscope`s specific offset values

int GYROOFFSET_Z = 35.5;                   // *****

float sens_cons = 14.375;                   // Constant used for converting data into scaled

```

```

// Data comes 16 bits so shifting is performed

data[0] = (((buffer[0]) << 8) | buffer[1]) + GYROFFSET_X) / sens_cons;

// Data is converted from bytes to integers

data[1] = (((buffer[2]) << 8) | buffer[3]) + GYROFFSET_Y) / sens_cons;

// Gyroscope data is converted to deg/sec by using sensitivity constant (14.375)

data[2] = (((buffer[4]) << 8) | buffer[5]) + GYROFFSET_Z) / sens_cons;


data[0] = sqrt(sq(data[0])+sq(data[1])+sq(data[2]));      // Data is averaged
}

// Writing value to register in ADXL345

void writeTo(int SENSOR, byte REGADD, byte VALUE)

{

Wire.beginTransmission(SENSOR);      // Beginning transmission

Wire.send(REGADD);                    // Sending register address

Wire.send(VALUE);                     // Sending value to write

Wire.endTransmission();               // Ending the transmission

}

// Reading the bytes starting from the first register of the axes

void readFrom(int SENSOR, byte INITREG, int BYTES, byte BUFFER[])

{

Wire.beginTransmission(SENSOR);      // Beginning transmission

Wire.send(INITREG);                  // Sending the address to be read

Wire.endTransmission();              // Ending the transmission

```



```

Wire.beginTransmission(SENSOR);           // Beginning transmission

Wire.requestFrom(SENSOR, BYTES);         // Requesting bytes

int i = 0;

while(Wire.available())

{

    BUFFER[i] = Wire.receive();           // Receive one byte

    i++;                                  // When i increases

}

Wire.endTransmission();                   // Ending transmission

}

void setup()

{

    Wire.begin();                          // Initiating I2C

    Serial.begin(9600);                     // Open the serial monitor

    startACC();                             // Calling the function

    startGYRO();                           // Calling the function

}

void loop()

{

    float acc_data[3];

    float gyro_data[3];

```

```
getACC(acc_data);          // Calling the function

getGYRO(gyro_data);       // Calling the function


Serial.print(gyro_data[0]);    // Gyro averaged data is inputted

Serial.print("a");            // Letter of “a” is inputted for further separating 2 data

Serial.println(acc_data[0]);    // Accelerometer averaged data is inputted

delay(76);                    // Delay is given

}
```

# APPENDIX B

## MATLAB R2011b

```
clear all; clc; close all;

% Removing previous
delete(instrfind({'Port'},{'COM8'}));

% Setting com number for communication
s = serial ('COM8');

% Configuring bit rate
set (s, 'BaudRate', 9600);

% Opening serial port
fopen (s);
hold on

% Variables; Sample rate & Data number
SR = 0.16;
i = 10;

% Infinite loop
while (1)

    % Reading the serial port
    str = fscanf(s);

    % Finding the letter a for seperation
    index3 = find(str == 'a');

    % First data obtained
```

```

str1 = str(1:index3-1);

% Second data is obtained
str2 = str(index3+1:end);

% Converting string to double precision value (Gyro)
Gx(i) = str2double(str1);

% Converting string to double precision value (Acc)
Ax(i) = str2double(str2);

% Figure 1 configuration
subplot(2,1,1),title('Accelerometer   XYZ   Axis   Average   Scaled   Data'),ylim([0.4
1.6]),xlabel('Time(s)'),ylabel('Acc (G)');
set(gca,'YTick',[0.4 0.6 0.8 1.0 1.2 1.4 1.6])
grid on

% Plotting the graph of ACC
% X axis is constructed (data number x sample rate)
% Y axis is constructed by using the read data
% Only last 5 data is stored for memory limitation
plot(SR.*(i-5:i),Ax(i-5:i),'--r','LineWidth',2);
hold on

% Figure 2 configuration
subplot(2,1,2),title('Gyroscope   XYZ   Axis   Average   Scaled   Data'),ylim([-100
1000]),xlabel('Time (s)'),ylabel('Gyro (deg/s)');
set(gca,'YTick',[-200 0 200 400 600 800 1000])
grid on

% Plotting the graph of GYRO
% X axis is constructed (data number x sample rate)
% Y axis is constructed by using the read data

```

```

% Only last 5 data is stored for memory limitation
plot(SR.*(i-5:i),Gx(i-5:i),'--b','LineWidth',2)
hold on
% Remainder for shutting the window periodically
if (rem (i,200) == 0)
    close;
end

% Data number increases & Begin Drawing
i=i+1;
drawnow;

end

```



# REFERENCES

- [1] Lutz, W. “European Demographic Data Sheet 2006”, viewed 2006.
- [2] LeMoyne, R., Coroian, C., Mastroianni, T. “Quantification of Parkinson’s disease characteristics using wireless accelerometers”, viewed 22 February 2009.
- [3] Stephenson, J., Nelson, L. “Link Between Home Pesticide Exposure and Parkinson’s Disease”, American Academy of Neurology, viewed 2000.
- [4] Ferrarin, M., Carpinella, I., Rabuffetti, M., Calabrese, E., Mazzoleni, P., Nemni, R. “Locomotor Disorders in Patients at Early Stages of Parkinson’s disease: a Quantitative Analysis”, viewed 3 September 2006.
- [5] Tortora, G., Derrickson, B., “Essentials of Anatomy and Physiology, Eight Edition”, Published 2010, Page 279.
- [6] Kandel, E. R., Schwartz, J. H., Jessell, T. M. Principles of Neural Science. New York: McGraw Hill, 2000, ch 43.
- [7] Nolte, J. and Sundsten, J. W. The Human Brain An Introduction to Its Functional Anatomy. St. Louis, MO: Mosby, 2002, ch 19.
- [8] Volkmann, J., Moro, E. and Pahwa, R. “Basic algorithms for the programming of deep brain stimulation in Parkinson’s disease,” Mov. Disord., vol. 21, no. S14, pp. S284–S289, June 2006.
- [9] Lyonsa,G.M., Culhane, K.M., Hilton, D., Grace, P.A. Lyons ,D. “A description of an accelerometer-based mobility monitoring technique” Medical Engineering & Physics 27 (2005) 497–504.
- [10] Maluf, N. and Williams, K. *An Introduction to Microelectromechanical Systems Engineering*. Published , 2004, 2nd ed. Boston: Artech House Publishers.
- [11] *What is MEMS Technology?* [Image online] Available at: <<http://www.memsnet.org/about/what-is.html>> [accessed 26 November 2011].
- [12] *Engineering: The Nature of Problems*. [Image online] Available at: <<http://openlearn.open.ac.uk/mod/oucontent/view.php?id=397841&printable=1>> [accessed 26 November 2011].
- [13] *Explain That Stuff*. [Image online] Available at: <<http://www.explainthatstuff.com/yoyos.html>> [accessed 30 November 2011].
- [14] *A Glimpse at MEMS An Introduction to MEMS World*. [Image online] Available at: <<http://knol.google.com/k/a-glimpse-at-mems#>> [accessed 27 November 2011].
- [15] Accelerometer and gyroscope buying guide. Available at: <<http://www.sparkfun.com/tutorials/167>> [accessed 15 august 2011].

[16] *IMU Digital Combo Board - 6 Degrees of Freedom ITG3200/ADXL345*. [Image online] Available at: <<http://www.sparkfun.com/products/10121>> [accessed 28 November 2011].

[17] *ANALOG DEVICES Digital Accelerometer ADXL345*. [Image online] Available at: <<http://www.sparkfun.com/datasheets/Sensors/Accelerometer/ADXL345.pdf>> [accessed 5 December 2011].

[18] *ITG-3200 Product Specification Revision 1.4*. [Image online] Available at: <<http://www.sparkfun.com/datasheets/Sensors/Gyro/PS-ITG-3200-00-01.4.pdf>> [accessed 5 December 2011].

[19] *Telosb platform*

<[http://www.willow.co.uk/html/telosb\\_mote\\_platform.html](http://www.willow.co.uk/html/telosb_mote_platform.html)> [accessed 20 September 2011].

[20] < <http://arduino.cc/> >

[21] *Arduino UNO*. [Image online] Available at: <<http://arduino.cc/en/Main/ArduinoBoardUno>> [accessed 28 November 2011].

[22] Protoshield. [Image online] Available at:

<http://se.rs-online.com/web/p/products/6961677/> > [accessed 28 November 2011].

[23] Fabio Varesano, Fabio., 2010. My First 6 DOF IMU Sensors Fusion Implementation: ADXL345, ITG3200, Arduino and Processing. Available at: <<http://www.varesano.net/blog/fabio/my-first-6-dof-imu-sensors-fusion-implementation-adxl345-itg3200-arduino-and-processing>> accessed at [15 November 2011].

[24] Gadget Gangster, 2010. Accelerometer & Gyro Tutorial. [online] Available at: <<http://www.instructables.com/id/Accelerometer-Gyro-Tutorial/>> [Accessed 20 November 2011]

[25] Jeremy Blum, 2011. ARDUINO TUTORIALS. [online] Available at: <<http://www.jeremyblum.com/category/arduino-tutorials/>> [Accessed 7 November 2011].

[26] Nyquist frequency. Available at :

< [http://en.wikipedia.org/wiki/Nyquist\\_frequency](http://en.wikipedia.org/wiki/Nyquist_frequency) > [Accessed 27 December 2011].

[27] LeMoyne, R., Coroian, C., Mastroianni, T., “Quantification of Parkinson’s disease characteristics using wireless accelerometers”, viewed 22 February 2009.

[28] Lemoyne, R., Mastroianni, T., Cozza, M., Coroian, C., Grundfest, W., “Implementation of an iPhone for characterizing Parkinson’s disease tremor through a wireless accelerometer application”, viewed 4 September 2010.

[29] Suresh, Lo. G., Stocco, A. R., Valenzuela, L., Leung, S. G., V. C. M., “A Wireless Sensor System for Motion Analysis of Parkinson’s Disease Patients”, viewed 2011.



