

A WEARABLE SENSORY SYSTEM DESIGN FOR SPINE MONITORING

by

HAZAN EZGİ İMER

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of
Master of Science

in

Electrical and Electronics Engineering



**KOÇ
UNIVERSITY**

September 13, 2017

**A WEARABLE SENSORY SYSTEM DESIGN FOR SPINE
MONITORING**

Koç University

Graduate School of Sciences and Engineering

This is to certify that I have examined this copy of a master's thesis by

HAZAN EZGİ İMER

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Engin Erzin (Advisor)

Assoc. Prof. Özden Özyemişçi Taşkiran

Assoc. Prof. Burak Acar

Date: _____



To my family

ABSTRACT

The body posture and spine structure play important roles in human health. Besides the unintentional injuries and accidents, wrong sejan postures, slouching and a static life style, in the course of time, may damage the functions of the spine. Therefore, the physical tracking of the spine structure, and determination and diagnosis of any existing complications are of capital importance for a healthy life. The aim of this thesis is to monitor the movement of the spine via real time visualization while capturing the angle values of the spine using flex sensors. For this purpose, hardware and software systems are developed. While the circuit, composed of a micro controller and flex sensors, is designed in the hardware system, the software system includes algorithms, written in Java programming language, for data acquisition, processing and visualization in real time. Various medical vest designs and sensor mounting solutions, such as medical tapes, are investigated for the wearable sensory system. Additionally, a dataset of sensory measurements from eight different spine exercise poses are collected, and a support vector machine classifier is utilized for identification of the spine exercise poses with high accuracy.

ÖZETÇE

Vücut duruşu ve omurga yapısı insan sağlığında önemli rol oynar. İstem dışı oluşan yaralanmalar ve kazaların yanı sıra, yanlış oturma pozisyonları, kambur yürüme ve statik bir yaşam tarzı gibi etkenlerde zaman içerisinde omurganın işleyişine zarar verebilmektedir. Dolayısı ile omurga yapısının fiziksel takibi ve varolan sorunların tespit ve teşhisi sağlıklı bir yaşam için çok önemlidir. Bu tezin amacı, esnek sensörler ile omurların açısal değerlerini yakalayarak gerçek zamanlı bir görsellemeyle omurga takibini yapmaktır. Bu amaçla donanım ve yazılım sistemleri geliştirildi. Donanım sisteminde, sensörlerden ve mikroişlemciden oluşan devre tasarımı yapılırken, yazılım sisteminde, JAVA dilinde geliştirilmiş bir algoritma ile veri toplama, işleme ve gerçek zamanlı olarak görselleştirilmesi sağlandı. Giyilebilir sensörlü sistemler için çeşitli tıbbi yelek tasarımları ve tıbbi bantlar gibi sensör yerleştirme çözümleri araştırıldı. Ek olarak, sekiz farklı omurga egzersiz pozlarından sensörlü ölçümlerin veri seti toplandı ve omurga egzersiz pozlarının yüksek doğrulukla teşhisi için bir destek vektör makine sınıflandırıcısı kullanıldı.

ACKNOWLEDGMENTS

First and foremost, I would like to express my deepest respect to my advisor Assoc. Prof. Dr. Engin Erzin who causes me to write this thesis for his patience, support and efforts.

I would like to thank Assoc. Prof. Özden Özyemişçi Taşkıran, M.D for her support, efforts and guidance. I have never seen such an understanding and smiling person like her. I am also grateful to Assoc. Prof. Dr. Yücel Yemez, Assoc. Prof. Nazan Canbulat, M.D and Assoc. Prof. Dr. Metin Sezgin for their feedback and helps in this thesis. I would especially thank Prof. Dr. Tekin Dereli who supported me to start the graduate program. I would like to thank Assoc. Prof. Dr. Burak Acar for his feedback and advices.

I would like to thank Selim Ölçer and Dr. Selçuk Çakmak for help me to design the hardware and software systems in this thesis.

I would like to thank my mother, Sevim Ünger, and brother, Berk Utku İmer, my greatest supporters in my life, for always staying with me and never forgiving their love. Without them, I would never have come to this point. I thank to my father, Rıza İmer who has a great engineering mind that I know for his advices about my thesis. I am very lucky to have a perfect man who has great heart and soul. I would like to thank Dr. Ali Ümit Cemal Hardal for his understanding, support and guidance.

I thank Ayşe Ekin Karayığit - Bulut and Dilara Aksoy, my friends for many years with whom I shared the same house and spent hours while drinking coffee in the kitchen, for being there during this hard period, listening my troubles and giving me serenity.

I thank Devrim Merve Saimler, my dearest friend who has been always there for

me for last two years. She provided her help and support, without any expectations of personal gain, to solve the problems that I faced in academic and personal life. I thank Meltem Civas who has the most colorful and kind personality in the world for her support and friendship.

I thank my office-mates for attending to my experiments and creating a great atmosphere in office. Also I would like to thank Emine Büyükdurmuş and Elif Tüysüz for their great helps and understanding.



TABLE OF CONTENTS

List of Figures	x
Chapter 1: Introduction	1
Chapter 2: Contributions	4
Chapter 3: Wearable Sensory System Design	5
3.1 Flex Sensors	5
3.2 Data Acquisition Hardware Design	8
3.3 Data Acquisition Software Design	10
3.3.1 Arduino Software (IDE)	10
3.3.2 MATLAB Software	11
3.3.3 JAVA Software	12
3.4 Wearable Vest Design	15
3.4.1 Using a Medical Vest	15
3.4.2 Using a Lycra T-shirt	16
3.4.3 Using a Lycra Vest	17
3.4.4 Using a Designer Vest	18
3.4.5 Using Medical Tapes	19
Chapter 4: Experimental Evaluations	22
4.1 System Test on a Flexible Stick	22
4.2 System Analysis on Spine Data	25
4.3 Data Collection	27
4.4 Statistical Analysis of Data	28

4.5 Pose Classification Using Sensory Data	34
Chapter 5: Discussion and Conclusions	37
Bibliography	39



LIST OF FIGURES

3.1	Visual samples and dimensions of the flex sensor.	5
3.2	Sensor readings: left column for each 30 degree, right column for each 10 degree.	7
3.3	Flex Sensor Schematic	8
3.4	Phases and different components of the hardware system	9
3.5	Four sample frames from motion capture in Matlab	12
3.6	Illustration of data visualization according to the algorithm written in Matlab.	13
3.7	Illustration of data visualization according to the algorithm written in Java	14
3.8	Samples of lowpass filtered raw sensor data with normalized cutoff frequencies: (a) 0.1; (b) 0.07; (c) 0.04.	15
3.9	Sample images from the medical vest	16
3.10	Sample image of the lycra T-shirt	17
3.11	Sample images from the lycra vest	18
3.12	Illustration of the vest design	19
3.13	Sample images from the designer vest	19
3.14	Sample images from the medical tapes	20
3.15	Sensor placement on human body	21
4.1	Test on a flexible stick: (a) sensors are attached on a flexible stick, (b,c) flexible stick at two different bending positions.	22
4.2	Calculation of sensor angle from visual observation of the flexible stick	23

4.3	The calculation of RMSE: (a) Plot of the sensor readings along the ground truth; (b) RMSE for each experiment; (c) RMSE in five degree interval; (d) RMSE for each degree.	25
4.4	Body posture and visualization in Java	26
4.5	System analysis on spine: (a) Data visualization, black line, and subject's back curve, white line; (b) Applying the rotation method on data visualization; (c) The displacement of data visualization along the x-axis.	27
4.6	Histograms of 11 sensors for two sample poses of subject 1: Forward flexion 1 and 2	29
4.7	The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject 1.	31
4.8	The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject 2.	32
4.9	The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject independent setting.	33

Chapter 1

INTRODUCTION

Body posture plays an important role in human health. Besides the random accidents and injuries, long hours of computer work and maintaining a static posture with inactive lifestyle may increase disorders in the neck, upper and lower limbs. These types of disorders may cause postural dysfunctions and further deteriorate pains if being careless of these disorders. Various studies in the literature discuss the effects of postural dysfunction in human life. In particular, Griegel-Morris et al. [1] investigated posture and pain in the thoracic, cervical and shoulder regions. They also examined the relations between posture deviation and pain. Some associations were found between fixed posture, neck and upper back pain [2]. Furthermore, approximately 70%-84% of people experience low back pain at some period in their lives [3].

Back pain limits the lifestyles of the patients and may cause economic burdens. Therefore, designing a system to monitor body posture becomes an important issue. These systems can be used in the natural environments of the patients without the necessity of direct observation by doctors or physiotherapists and can alert patients to correct their posture. Such a system targets patients to monitor their progress and improve their quality of life. Such systems should be available at low prices, assemble easily and be portable. As they are also sufficiently comfortable, these systems are good candidates to ease the daily actions of the patients. To this end, various systems have been proposed in the literature, including accelerometers [4, 5, 6], gysroscopes [4, 6], inertial sensors [7, 8, 9], bend sensors [10, 11, 12], optical bend sensors [13, 14] or video/optical coordinate capturing system [15, 16].

Tognetti et al. [7] presents an upper limb kinesthetic garment for gesture, posture

and movement detection using elastomers as sensors. This approach is similar to that of studied by Mattmann et al. [8] who examined the upper body posture. The authors presented a garment prototype using thermal elastomeric elongation strain sensors, as well. In addition, they focused on the torso and were able to distinguish between 27 different postures while fifteen of them were sitting and rest standing postures.

Some intriguing devices, including a smart shirt, a leotard and a glove, have been implemented to monitor the movements of arms and fingers and also the upper trunk with similar sensors [9]. A bend sensor has been used for monitoring seated spinal posture, as well [13]. The authors further evaluated four bend sensors which are a polypyrrole-coated open-cell foam sensors, two piezo resistive sensors and plastic optical fiber sensor. While both resistive sensors and optical fiber sensor achieved a good degree of accuracy, the foam sensor displays poor and variable response.

To cover the specific spine area, sensor length diversity plays an important role in choosing a sensor. While the lengths of two resistive sensors are definite, the optical fiber sensors have an unlimited length. As a result, a plastic optical fiber sensor is used for monitoring seated spinal posture. Tsuchiya et al. [10] developed a wearable sensor system, which contains three flex sensors and two accelerometers to estimate lumbosacral alignment and lumbar load. In addition, this system is used to measure the parameters of anteflexion and the change in dimensions of the lumbar spine from changes in posture. The authors accounted for individual differences to improve the accuracy of the estimation. They measured the lumbosacral dimension by using X-ray images and calibrated the estimation.

On the other hand, video/optical coordinate capturing systems [15] represent the gold-standard apparatuses for measuring angular displacements [17]. While these methods have maximum reliability and precision, they can only be easily applied in laboratory. Therefore, they are expensive and not portable. Consequently, sensor selection is the main part of measuring human body kinematics. It should be at low cost and have good precision and mobility.

In this case, a flex sensor can be shaped to fit their size to a body joint and

requires simple electronic interfaces [18]. A flex sensor is not only sensitive and repeatable but also strong and robust [17]. In addition, it is inexpensive and can be used in different research areas such as robotics, gaming, medical devices, musical instruments and physical therapy. In the health sector, it can be used in physiotherapy [19], rehabilitation [20, 21], tele-health [22] or long-term monitoring [23, 17].



Chapter 2

CONTRIBUTIONS

The aim of this thesis study is to use angular changes measured on the flex sensors to determine the bending angle of the corresponding spine. By doing so, we shall be able to track the general position and the movement of the spine.

A successful implementation of our ideas require engineering a fully controlled system. To this end, we developed and designed both hardware and software which specifically comply with our purposes. The hardware system is comprised of a micro controller and flex sensors. The software system consists of two main parts which are graphical user interface and data processing algorithms. To be able to find the angular variation of the sensors calculated by using the data obtained from the microcontroller, a numerical algorithm was written in Matlab programming language. In addition, the algorithm, written in Java programming language, performs real-time data acquisition and visualization. A user can monitor his/her motion in real time by using the graphical user interface (GUI). The corresponding data can be stored and revitalised when needed, as well.

In order to make sensory system wearable, we tried various types of vest designs and medical tapes. Each design and tape has been tested with the same spine movement patterns and then the results are compared with each other.

Data analysis was performed for eight different poses on the data collected with the best wearable system. Thus, we performed statistical characterization of the sensor data for each pose and implemented a SVM based pose classification system.

Chapter 3

WEARABLE SENSORY SYSTEM DESIGN

3.1 Flex Sensors

Flex sensors are passive resistive devices and typically fabricated by resistive characteristics of a conductive material, in ink form, on a flexible substrate. The flexibility of the sensor mostly rely on the substrate's properties. The change of resistance value depends on sensor's cracks. When flex sensors are bent, the ink separated into many micro cracks and so the resistance is increased. When it is returned to the flat position, the distance between the cracks' faces shrink and then the resistance is decreased. Spectra Symbol, Abrams-Gentile and Flexpoint Sensor Systems are some of the companies which have commercialized flex sensors. Since they are easily available in the local market, we used the spectra Symbol's sensors. Spectra Symbol's flex sensors are available in two sizes which are 2.2" (5.588 cm) and 4.5" (11.43 cm) long. Visualization and dimensions of the flex sensor are given in Figure 3.1.

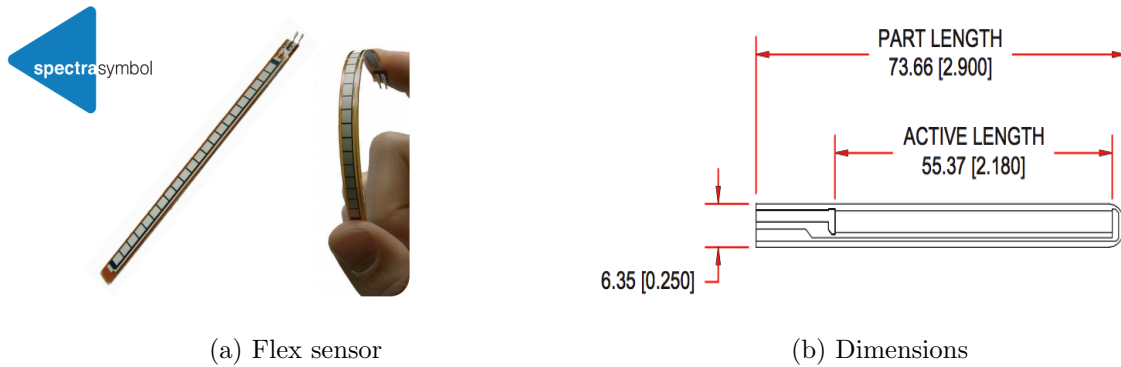


Figure 3.1: Visual samples and dimensions of the flex sensor.

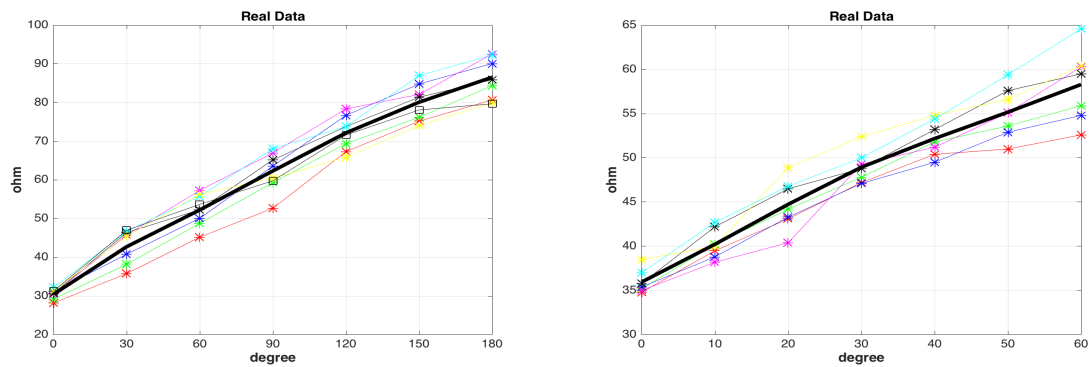
To test flex sensor sensibility, we measure the change of sensor value for different

bending angles by using a multimeter. We use two main steps to control these sensors by using multimeter sensor measurements for each 30 and 10 degrees of bending angles, and comparing readings of flat sensors after repeated experiments.

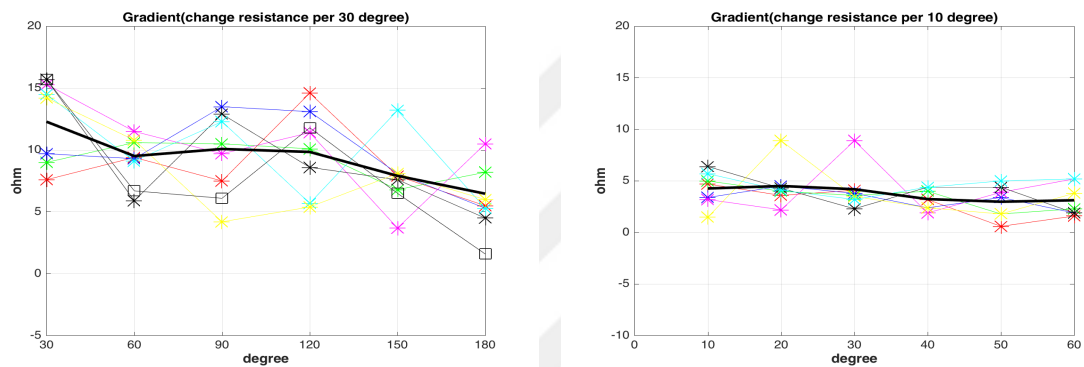
First, we measure the sensor value for each 30 degree from 0 to 180 degree. This experiment is repeated eight times and the data is plotted on the left column of Figure 3.2(a). In this figure, each color represents one experiment. Left column of Figure 3.2(b) plots the change of sensor readings for each 30 degree. For instance, the pink plot shows the change of readings as 15.3 ohm from 0 to 30 degree and 11.5 ohm from 30 to 60 degree.

The sensor's initial value can change at every experiment. Left column of Figure 3.2(c) plots readings of flat sensor in each experiment. Sensor's resistance value at flat position is 28.2 ohm in the first experiment and in the second experiment, it is 29.2 ohm. The reason of this hysteresis can be the sensor's inability to return to the flat position after bending.

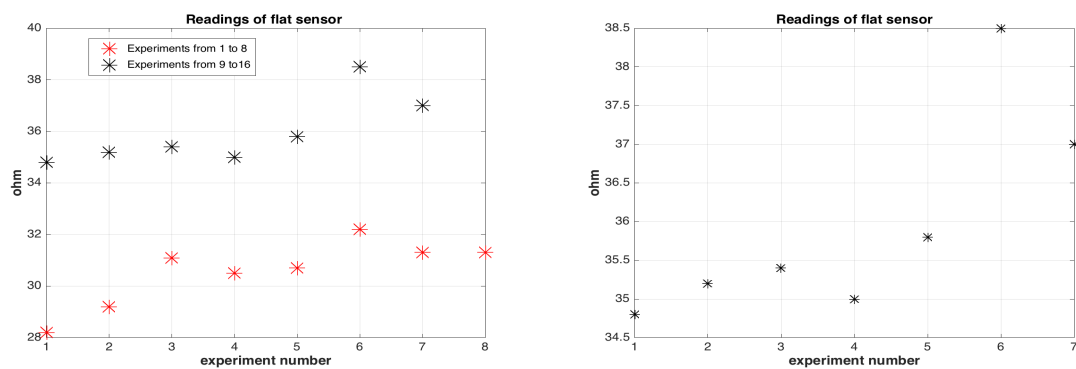
Also, we check the sensor's inability with 10 degree increments. The same procedure is implemented. With these results, the flex sensor is observed to be sensitive even 10 degree angular changes. Another issue is the change of starting point at a flat position of the sensor. Totally, sixteen experiments are done. First 8 experiments are examined with 30 degree increments and the rest are examined with 10 degree increments. In the right column of Figure 3.2(c), each star represents the initial point of one experiment. As observed, when the bendings are executed repeatedly, initial point of sensor can change and increase with time. In our final system, we solve this problem by registration of sensor readings at the beginning of each data acquisition.



(a) Plot of the sensor resistance readings vs bending angle degrees



(b) Sensor reading increments between bending angles



(c) Flat sensor readings in each experiment

Figure 3.2: Sensor readings: left column for each 30 degree, right column for each 10 degree.

3.2 Data Acquisition Hardware Design

The flex sensor cannot plug in Arduino directly and Arduino cannot measure flex sensors resistance. Thus, we build a simple voltage divider circuit to connect flex sensor and Arduino.

A simple voltage divider made of two resistors connected in series. The source voltage is divided among the resistors in direct proportion to their resistance. Since the two resistors are in series, the same current flows in both of them. The voltage output of a voltage divider is given as,

$$V_{out} = V_{in} \frac{R_2}{(R_1 + R_2)} \quad (3.1)$$

This voltage divider circuit is designed as shown in Figure 3.3. We use flex sensor as R_1 and $22\text{ k}\Omega$ resistor as R_2 . V_{out} is connected to the analog input port of Arduino with V_{in} being 5V.

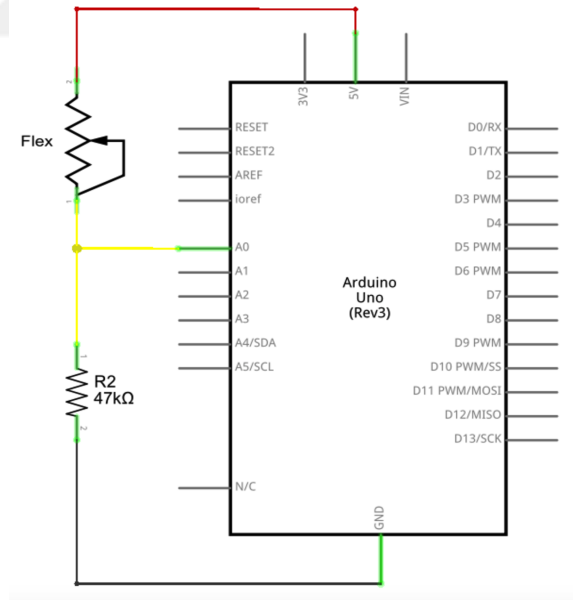


Figure 3.3: Flex Sensor Schematic

We have 11 flex sensors hence 11 resistors, which are $22\text{ k}\Omega$. Those components are connected on PCB (printed circuit board). A PCB is a thin board usually made

of fiberglass. According to the circuit design, conductive pathways are printed onto board. These pathways provide proper connections of the circuit components. The circuit is designed by using Proteus software as shown in Figure 3.4(a). Figure 3.4(b) shows the printed circuit board. According to circuit design, the circuit components are soldered on the PCB. Figure 3.4(d) presents the final sensory system.

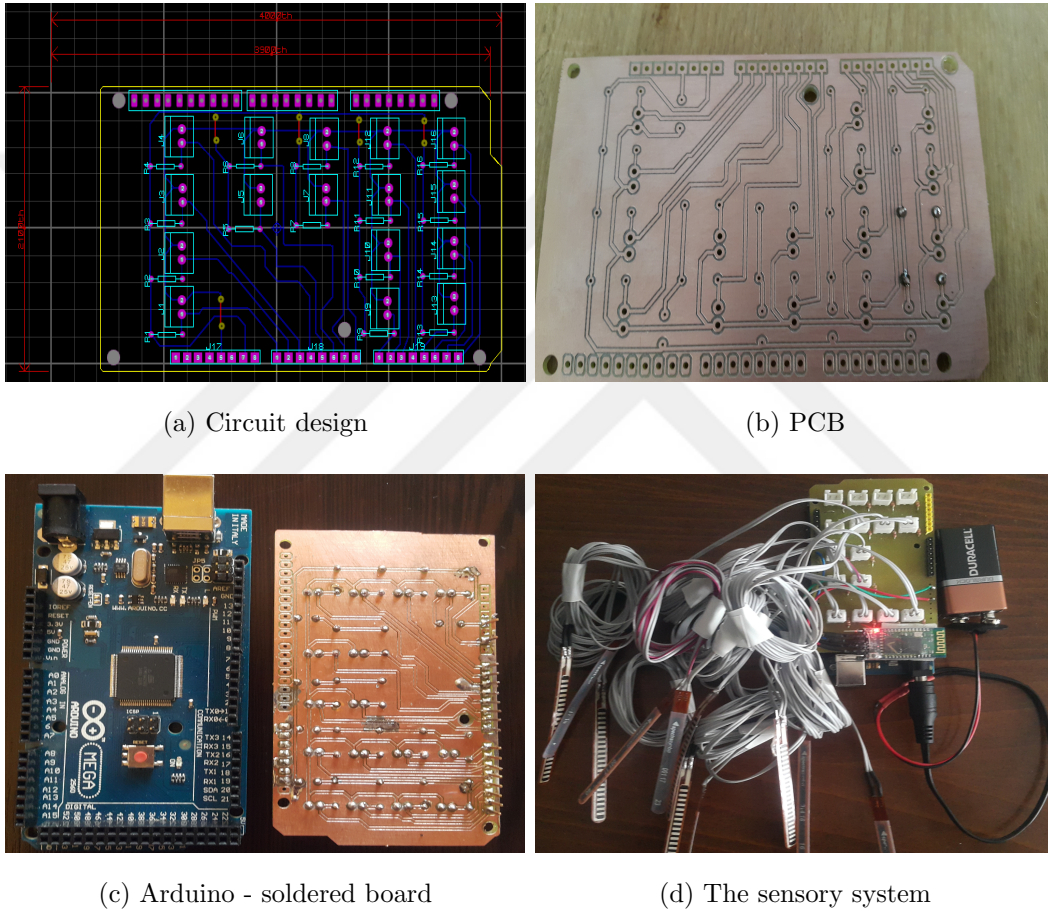


Figure 3.4: Phases and different components of the hardware system

Flex sensors are attached to the board with the white connectors as shown in Figure 3.4(d). The board is connected to the Arduino via headers. In order to increase the range of movement of the user, the Bluetooth (HC05) is used for connection between the Arduino and the computer.

3.3 Data Acquisition Software Design

3.3.1 Arduino Software (IDE)

For the data acquisition process, a system is built to record flex sensor data in real-time. The system is comprised of a micro controller and flex sensors. Arduino mega 2560 is used as a micro controller. Arduino mega 2560 has 16 analog inputs, 54 digital I/O pins an, 4 UARTs (hardware serial ports) and a 16 MHz crystal oscillator. It simply connects to a computer with a USB cable or power with a AC-to-DC adapter or battery.

Main functionality of the Arduino is to perform analog to digital conversion of multiple sensor readings in a synchronous manner, so that the computer can record these values in real-time. The arduino's ADC(analog to digital converter) can read voltages from 0 - 5 V with 10 - bit resolution. The Arduino IDE (integrated development environment) is written in Java. Using Arduino IDE, we can write and compile our code and save results into a computer. We develop a software on the computer to communicate with the Arduino and to take the sensor readings. The main steps of this software are listed as:

i. `Serial.begin()`

Open serial port and set the data rate in bits per second (baud) for serial data transmission. The data rate 9600 is used in our system.

ii. `analogRead()`

Read all sensor values from a specified analog pin.

iii. `map()`

Provide re-map from one range to another. To arrange map function, we measure sensor's value when bending and flat position of a sensor and map those values to zero and ninety degree. Then it estimates the bend angle based on those calibration values.

iv. `constrain()`

Make sure value is within a range.

v. `Serial.print()`

Pass the results to the serial port.

In addition, instead of the delay function of Arduino, we use timer interrupt routine. The delay function pauses the program at given time intervals. On the other hand, timer interrupt tells the processor to stop what its working and wait for further instruction for a certain interval. When an interrupt occurs, the processor starts the execution of an Interrupt Service Routine (ISR). A piece of code in ISR tells the processor what to do. After the execution of ISR, the processor returns back to the instruction. In our system, we set the sampling rate to 10 Hz, i.e. Arduino reads data from the sensors at every 100 ms.

3.3.2 MATLAB Software

In the beginning phase of this study, an algorithm is written in Matlab to track the motion using data visualisations. First, the data obtain from Arduino is saved in a text file. The algorithm makes a two-dimensional visualization using this data. The visualized data is synchronized with a video which is recorded by a camera throughout the entire motion. Then the visualized data is merged on the synchronized video. Figure 3.5 presents sample frames from motion capture video.

Each sensor is visualized as a line with a starting coordinate (X_n, Y_n) , length γ , and angle Θ as shown in Figure 3.6. While the black points show the beginning and end of the sensor in Figure 3.6, the red points show the middle point of the sensor. Hence, given the $(n - 1)^{st}$ sensor the starting coordinate of the n^{th} sensor can be extracted as:

$$\begin{aligned} X_n &= \gamma * \cos(90 - \Theta) + X_{n-1}, \\ Y_n &= \gamma * \sin(90 - \Theta) + Y_{n-1}. \end{aligned} \tag{3.2}$$

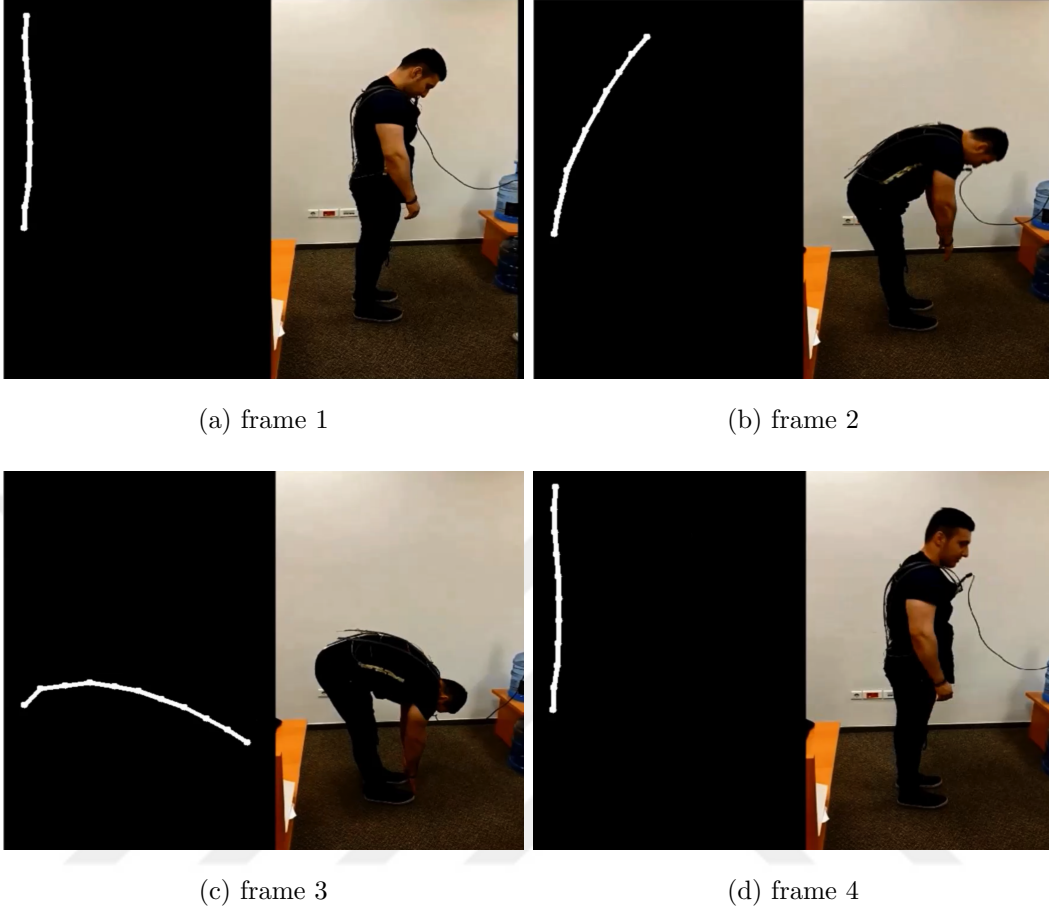


Figure 3.5: Four sample frames from motion capture in Matlab

For instance, the angular values of the first and second sensors are 0 and α degrees. That is, while the second sensor is bent to α , the first sensor remains stationary. In addition, X_1 and Y_1 are initial points and the γ is the length of sensor. In this case, starting coordinates of the sensor 1 and 2 can be extracted using equation (3.2) as given in Figure 3.6.

3.3.3 JAVA Software

Instead of a USB cable, the bluetooth module is used to establish a connection between the Arduino and the computer. After the computer recognizes the bluetooth, our algorithm handles all the communication. There are two main parts in this algorithm,

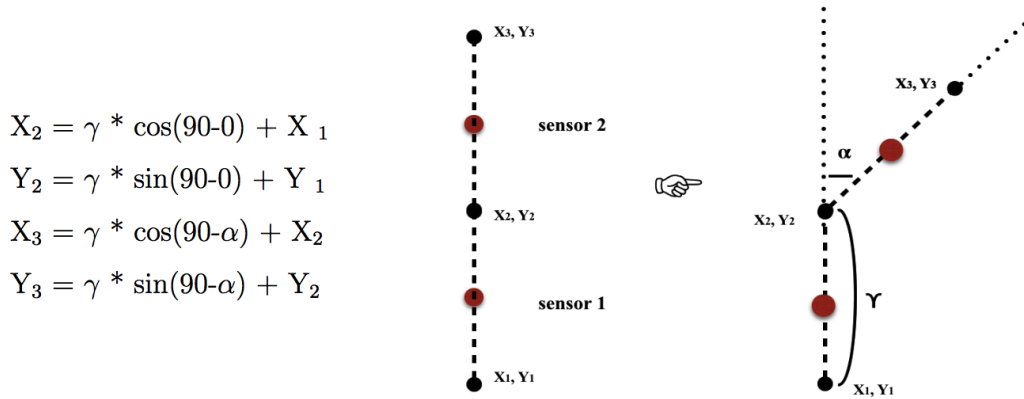


Figure 3.6: Illustration of data visualization according to the algorithm written in Matlab.

the registration part and real-time data acquisition and visualisation.

Although the sensors are calibrated in Arduino, some deviations are observed in the sensor's readings. To prevent these deviations and to get more precise results, an algorithm is written for the registration part. In the registration part, before starting the motion, the serial port is opened and the values of the sensors in their stationary position are collected and saved.

In real time visualisation part, the sequence of data coming from the serial port is subtracted from the data sequence obtained in registration part throughout the entire movement. The modified data is visualized in real time. In addition, the data of the movement can be saved as a text file and then ready to visualize again when needed.

In the Matlab implementation, each sensor is visualized as a line and the implementation is based on the start and end points of the sensor. However, in realtime Java implementation, we choose to represent each sensor with the middle point as shown in Figure 3.7. Since the distances between middle points of sensors are different from each other, the system takes these distances as an input. Thus, the visualisation model is proportional to the actual distances. Location of the n^{th} sensor

can be extracted as referenced to the $(n - 1)^{st}$ sensor as:

$$\begin{aligned} X_n &= \gamma_{n-1} * \cos(90 - \Theta) + X_{n-1}, \\ Y_n &= -\gamma_{n-1} * \sin(90 - \Theta) + Y_{n-1}. \end{aligned} \quad (3.3)$$

The angular values of the first and second sensors are α and β degrees. X_1 and Y_1 are initial points and the γ is the distance between middle points of sensors. In this case, locations of sensors 1 and 2 can be extracted using equation (3.3) as given in Figure 3.7.

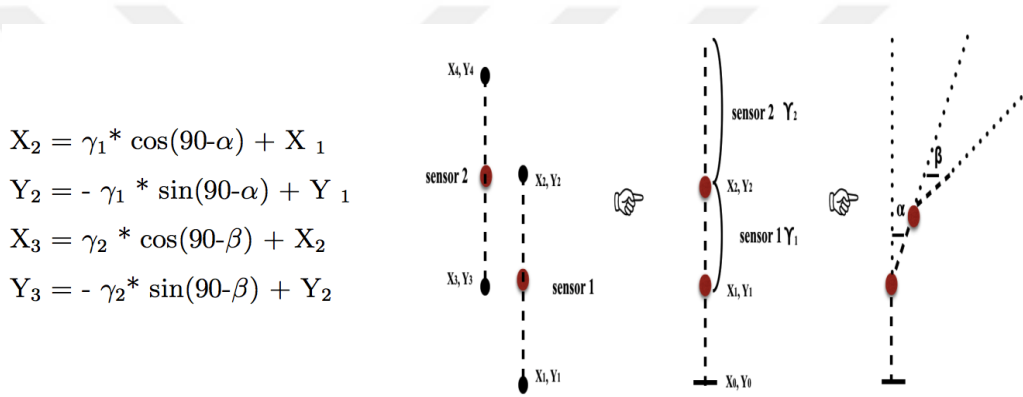


Figure 3.7: Illustration of data visualization according to the algorithm written in Java

When there is no change in the movement, it is expected to observe a still spine contour. However data is noisy and therefore visualizations carry this noise. To reduce the noise on the data, we implement a low pass filter with the following transfer function:

$$H(z) = \frac{Y(z)}{X(z)} = \frac{b_1 + b_2 * z^{-1} + \dots + b_{n+1} * z^{-n}}{a_1 + a_2 * z^{-1} + \dots + a_{m+1} * z^{-m}}, \quad (3.4)$$

where the filter coefficients a_n and b_n are extracted using the Butterworth filter design toolkit of Matlab. We utilize three fourth order low-pass filters with three different normalized cutoff frequencies 0.1, 0.07 and 0.04. Figure 3.8 presents samples of noisy and filtered sensor readings with these three filters.

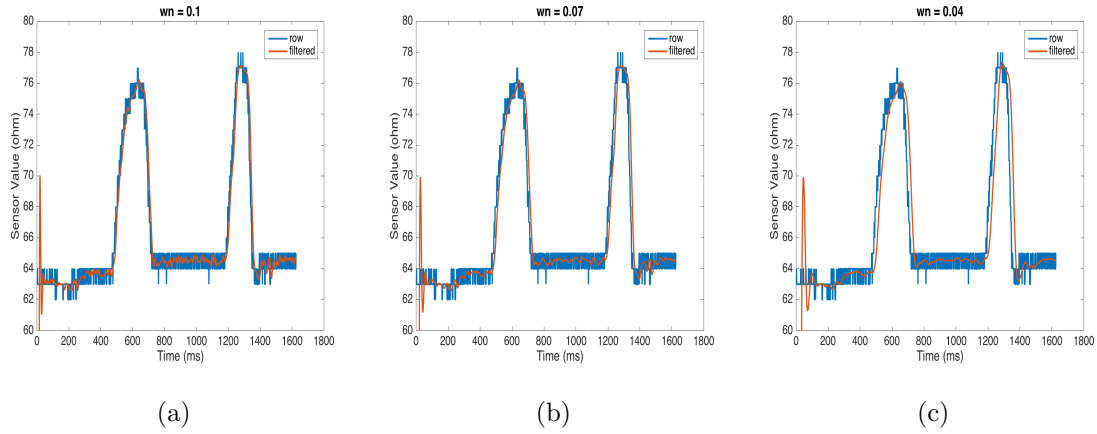


Figure 3.8: Samples of lowpass filtered raw sensor data with normalized cutoff frequencies: (a) 0.1; (b) 0.07; (c) 0.04.

In the real time Java visualization software, these three lowpass filters are available as optional preprocessing of the sensor readings before the visualization. The lowpass filtering runs in real time without a significant delay.

3.4 Wearable Vest Design

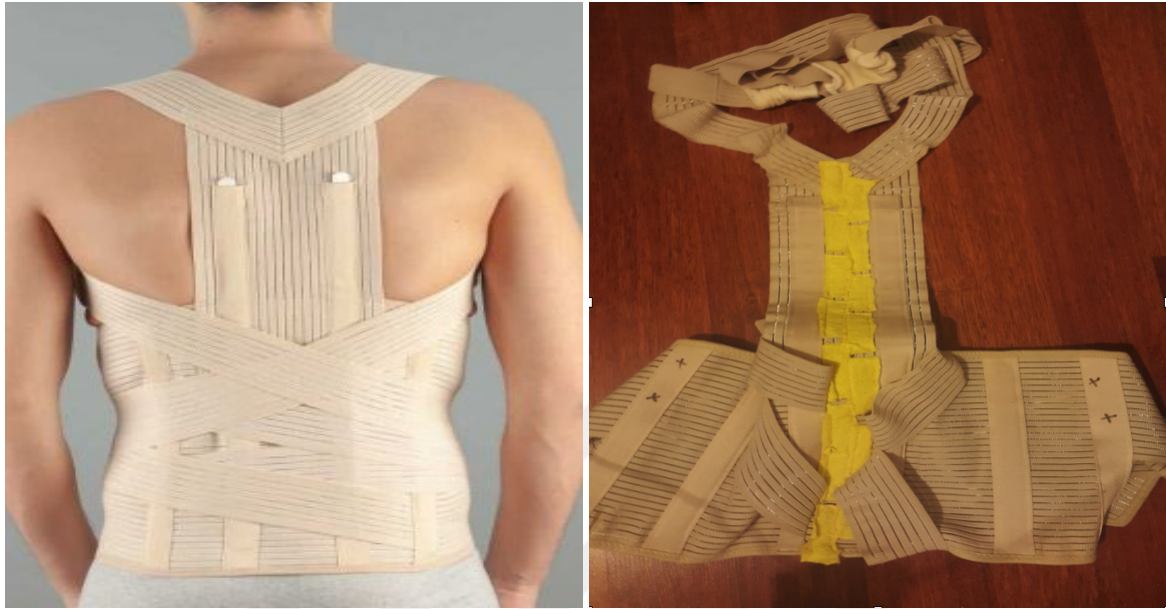
In order to make sensory system wearable, various types of vest designs and medical tapes are tested with similar spine movement patterns. In each section, the advantages and disadvantages of vest or medical tapes are explained and various techniques for placing sensors are tested.

3.4.1 Using a Medical Vest

First, we investigate a medical vest, which is shown in Figure 3.9(a). The iron bars on the left and right side of the spine are removed and the pockets are sewed on the vest as shown Figure 3.9(b). The sensors are placed in this pockets.

Although this vest surrounds body well, we can not get accurate data from flexible sensors, since it is very thick and not stable when moving. In addition, it has been

observed that this vest limits spine motion to a large extent.



(a) Medical vest on a subject

(b) Medical vest after modifications

Figure 3.9: Sample images from the medical vest

3.4.2 Using a Lycra T-shirt

In the first trial, we observe that the vest should be thin and slim such as a lycra t-shirt. For this purpose, a lycra t-shirt is modified as shown in Figure 3.10. To prevent the movement of the sensors, we sew the sensors on the vest. But then, the yarns apply a pressure to the sensors. This pressure causes not only to change the sensors' values but also cause damage to the sensors. Moreover, this vest can not turn back to the starting locations on the spine after the user completes the movement. Hence, sensor readings correspond different spine locations as sensors change their locations.

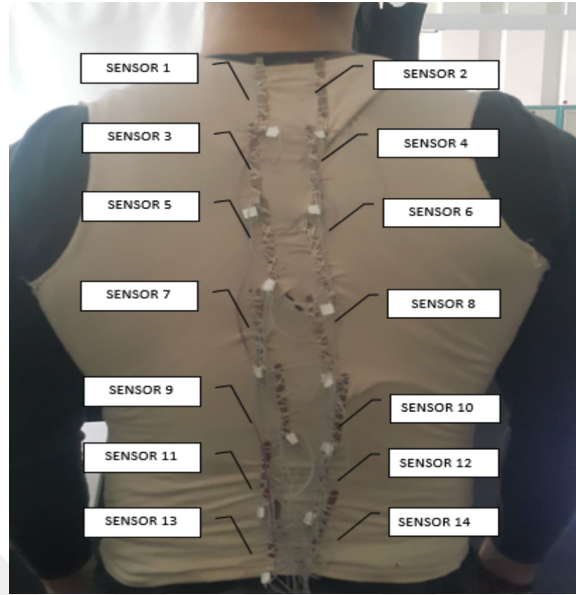


Figure 3.10: Sample image of the lycra T-shirt

3.4.3 Using a Lycra Vest

In this vest, we use a double-sided tape to place the sensors on the vest. The yellow parts are a double-sided tape in Figure 3.11(a). One side of the tape is on the vest and the sensors are put on the other side. To protect the sensors, they are covered with tape.

This vest also surrounds the body. Two lycra yarns, which are tied to the legs, are used to force the vest to return its original position after the spine movements. Although the sensors can be placed on exact vertebrae via double-sided tape, it is observed that this tape cannot hold the sensors on the vest during the motion.

The first three vests are designed with in-house modifications using trial and error type of approaches. Although the problems can be reduced, we can not find or modify perfect vest to collect accurate data without any problem. In that point, we found a designer to design a more proper vest.



Figure 3.11: Sample images from the lycra vest

3.4.4 Using a Designer Vest

The designer's vest is shown in Figure 3.12. In Figure 3.12(a), there is a pocket to put the Arduino. The cables are covered by fabric. When needed, it can be opened via velcro. In Figure 3.12(b), there are three black lines. The middle line shows the location of spine. Both sides show the locations of sensors. There is also extra fabric to cover the back. There are some fabrics under the vest to tie to the legs.

Interlining tape and double-sided tape are used in this design. First, interlining tape is pasted on the vest with using an iron. The double-sided tape is pasted on the interlining tape as shown in Figure 3.13(b-c). The sensors are placed on the double-sided tape. Thus, we prevent the sensors to fall down. It is observed that there is a space between a waist region and the vest. Thus, we couldn't obtain accurate data from the sensors.

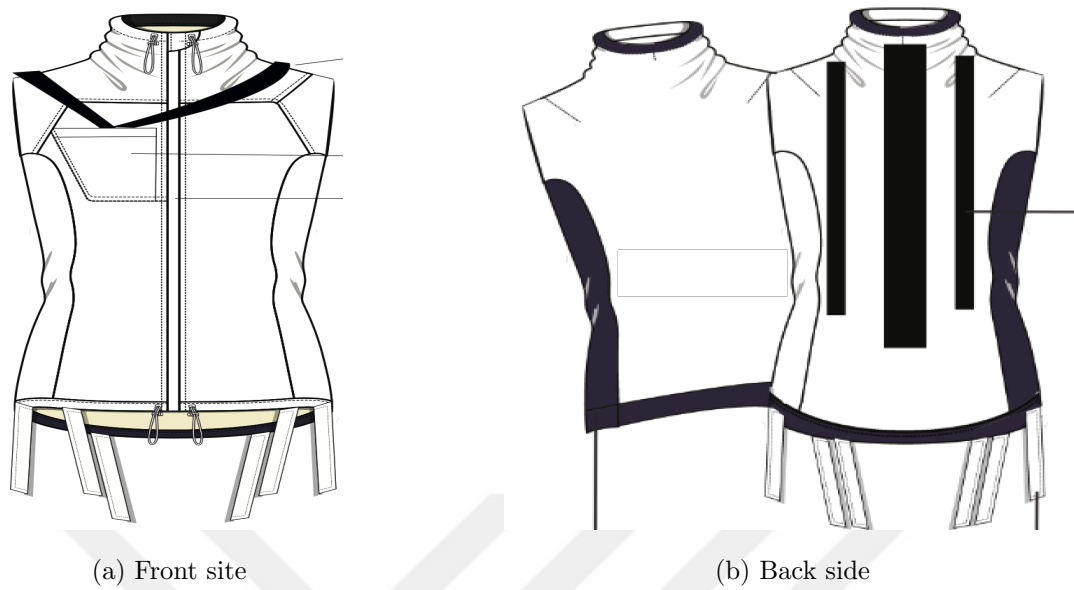


Figure 3.12: Illustration of the vest design



Figure 3.13: Sample images from the designer vest

3.4.5 Using Medical Tapes

As the designed jacket can not generate the desired output, next, we test medical tapes. In this case the sensors directly place on the human body.

First, we use two different tapes to place the sensors on the body and prevent them from falling during the motion. Although these tapes, shown in Figure 3.14(a-b), are very thin and flexible, we observe that the sensors are shifting during the motion as shown in Figure 3.15(a). In addition, the adhesive side of the tape, which is on the sensor, can damage the upper part of sensors in the long term.

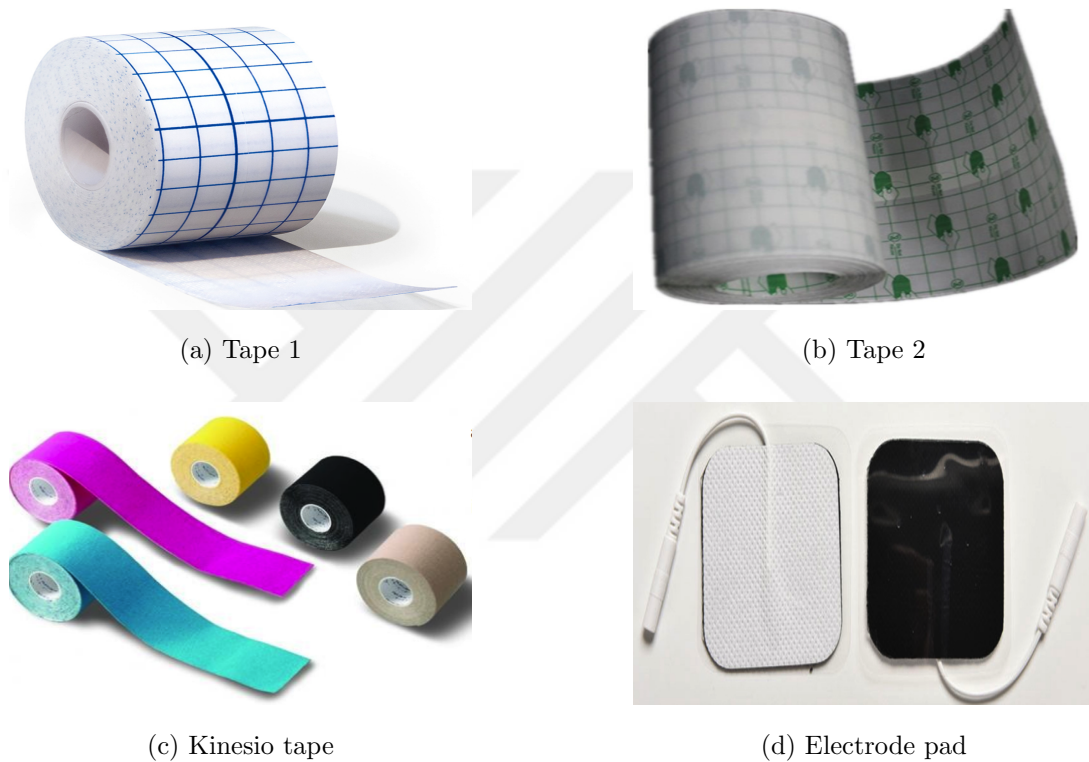


Figure 3.14: Sample images from the medical tapes

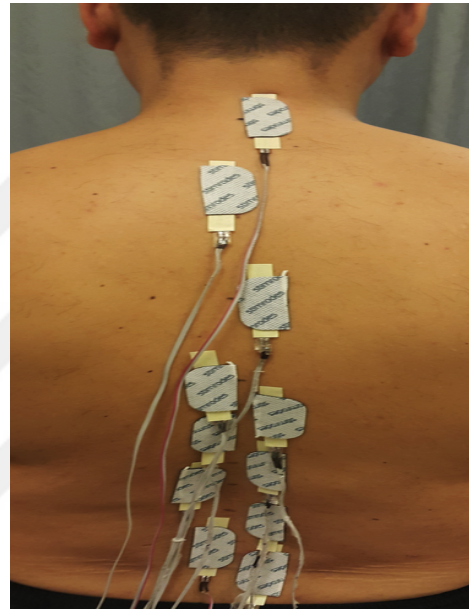
Then, we use a Kinesio tape which is shown in Figure 3.14(c). The elasticity of this tape is similar to that of human skin, therefore, the sensors may either fall down during the movement when the sensors place on the tape or squeeze and are damaged when placing under the tape because of in-elasticity of the sensor.

Finally, we use electrode pads shown in Figure 3.14(d). In order to prevent any shifting of the sensor locations on the human body and to be able to track the same determined vertebrae during the motion of the body, the sensors are fixed at their

midpoints as shown in Figure 3.15(b). In order to receive data from the sensors, the whole surface of the sensor must touch the body completely. Thus, a tight and stretchy t-shirt is used. Due to the thickness of the tape, we observe that wearing a t-shirt doesn't apply enough pressure to the sensors to force a sufficient contact with the body.



(a) Using the tapes 1 and 2



(b) Using the electrode pads

Figure 3.15: Sensor placement on human body

Under these circumstances, the tape should be thin enough to allow the t-shirt to apply pressure to the sensors completely and has a strong adhesive to hold the sensors on the body during the motion. Hence, tape 1, shown in Figure 3.14(a), is used to fix the sensors on the body from their midpoints. Wearing a tight and stretchy t-shirt helps the sensors to contact on the body completely. Therefore, we obtain accurate data from sensors.

Chapter 4

EXPERIMENTAL EVALUATIONS

4.1 *System Test on a Flexible Stick*

In order to test our system, an algorithm is implemented in Matlab. We bend the sensors using a flexible stick in these experiments. First, we look at the angular differences by overlapping the two images. Then we calculate the angles of each sensor via the image processing.

In order to observe the angular differences between the image and visualization obtained from the Java software, we put the sensors on the stick and bend it with five different angles. In each angular bending, we take a photo of the stick and capture the visualization of sensor angles obtained from the Java software. Then these two images are overlapped as shown in Figure 4.1.

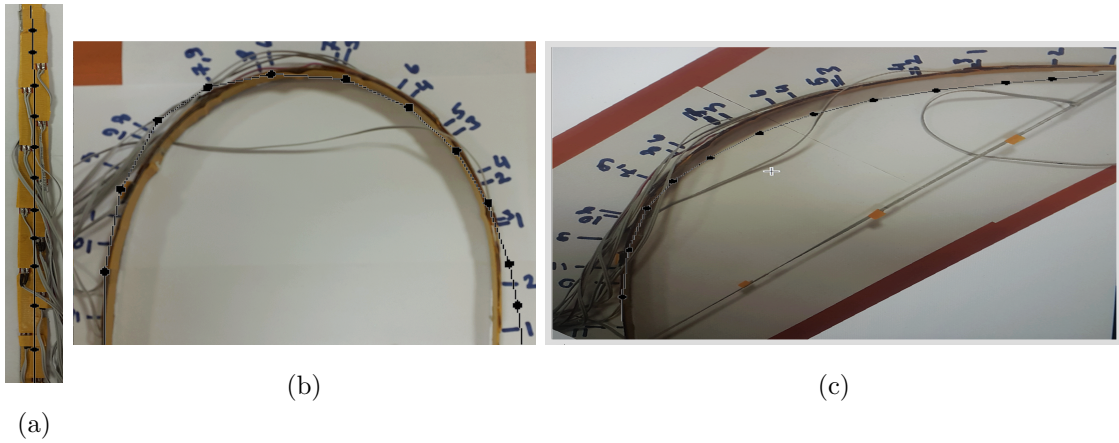


Figure 4.1: Test on a flexible stick: (a) sensors are attached on a flexible stick, (b,c) flexible stick at two different bending positions.

To be able to calculate the sensor angles, an algorithm is implemented. In Matlab, three points are selected on the image for each sensor. These three points represent the start, middle and end points of the sensor. A curve fitting through these three points is applied. The start and end points of the sensor are combined with a line. Tangent lines are drawn from the start and end points of the sensor. These two tangent lines are intersected at one point. The half of the angle, $\Delta/2$, shown in Figure 4.2(a), gives us a sensor angle.

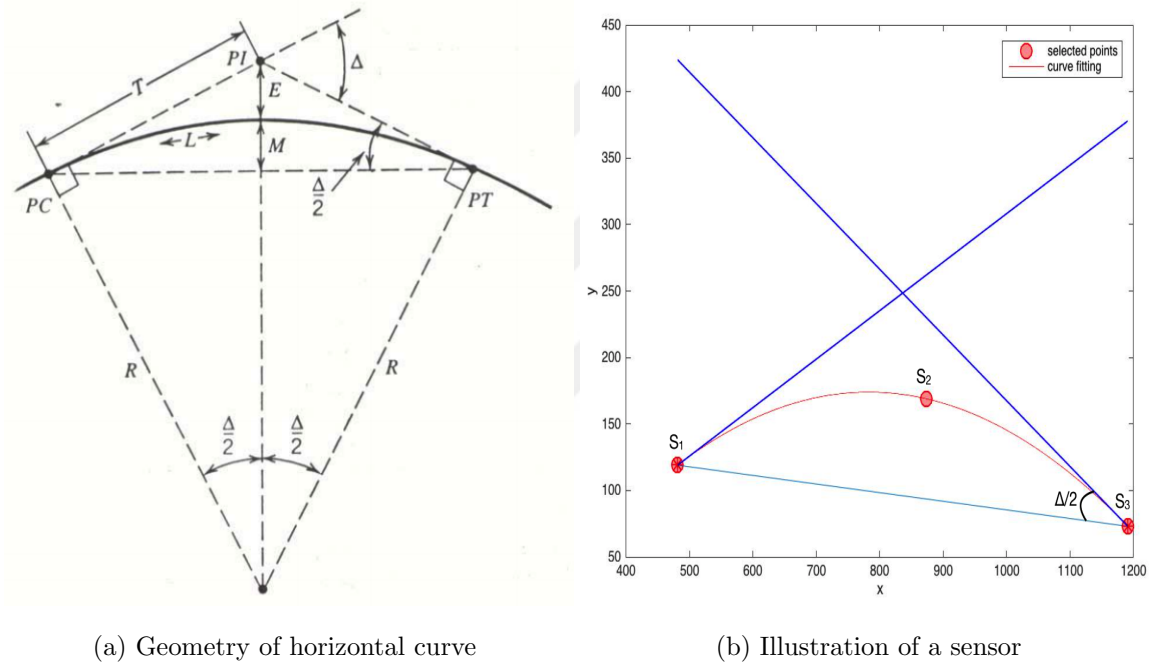


Figure 4.2: Calculation of sensor angle from visual observation of the flexible stick

The matlab function, which is fit function, creates the fit a quadratic curve to the given data with the linear model Poly2 is:

$$f(x) = p_1x^2 + p_2x + p_3 \quad (4.1)$$

where p_1 , p_2 and p_3 are the coefficients of model. The x is the point which is the either the start or end point of the sensor. This equation gives a tangent line for a

specific point. The derivative of $f(x)$ function,

$$f'(x) = 2p_1x + p_2, \quad (4.2)$$

is the slope of the tangent at the point x , given by equation (4.1).

Suppose that the slopes of the two tangent lines are m_1 and m_2 . The angle Δ is obtained as,

$$\begin{aligned} \tan(\Delta) &= \left| \frac{m_1 - m_2}{1 + m_1 m_2} \right| \\ \Delta &= \arctan\left(\frac{m_1 - m_2}{1 + m_1 m_2}\right) \end{aligned} \quad (4.3)$$

According to Figure 4.2(a), if we divide Δ by two, we can find the angle of sensors based on the image. To compare the sensor angles obtained from visual inspection and sensor reading, the Root Mean Square Error (RMSE) is calculated. The RMSE is used to measure the difference between sensor angles which are obtained from sensor readings and visual calculations. The RMSE for a sequence of sensor readings is computed as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (x'_i - x_i)^2}{n}} \quad (4.4)$$

where n is the total number of readings, x_i is the visually calculated ground truth angle for the i -th measurement and x'_i is the i -th sensor reading.

Fifteen experiments are performed. Our results are presented in Figure 4.3. In Figure 4.3(a), x and y-axes represent the sensor angles that are obtained from visual calculations and sensor readings, respectively. Figure 4.3(b) presents the RMSE for each experiment. The first experiment's RMSE is approximately 1.88 degree and second one is 1.21 degree. In Figure 4.3(c), we calculate RMSE from zero to twenty-five degree in five degree intervals. RMSE of angle values measure between 0 and 5 degrees is 1.337 degree. Also we calculate RMSE for each degree shown in Figure 4.3(d). These results show that the error rate is smaller than 2.5 degree for each degree.

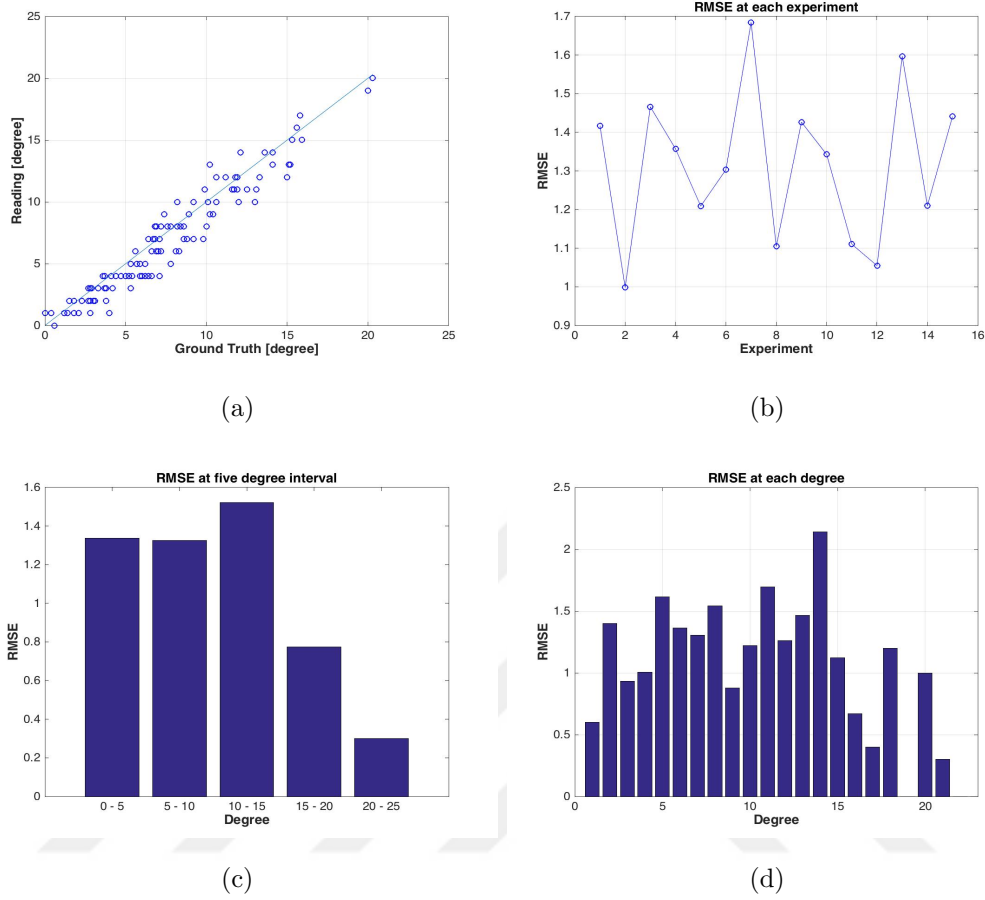


Figure 4.3: The calculation of RMSE: (a) Plot of the sensor readings along the ground truth; (b) RMSE for each experiment; (c) RMSE in five degree interval; (d) RMSE for each degree.

4.2 System Analysis on Spine Data

We also test the sensors on human body posture. The person is photographed when s/he reaches the desired spine pose as shown in Figure 4.4(a). In this pose, the data visualization is obtained from the Java software which is shown in Figure 4.4(b).

The image is cropped to take the part of the spine by using a Matlab algorithm. To find the boundaries of objects within cropped images, the edge detection method, which is an image processing technique, is used. Thus to capture a subject's back

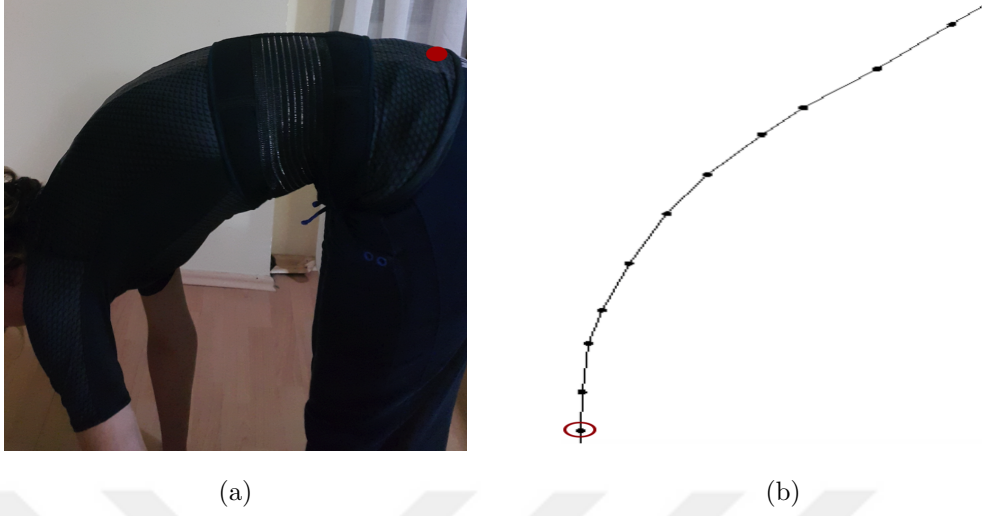


Figure 4.4: Body posture and visualization in Java

curve, the edge detection method is applied on this part of the image. As seen in Figure 4.5(a), after the image which is applied edge detection methods is flipped, we combine these two images. The last sensor, which is the red circle as shown in Figure 4.4, is known in both images. The coordinates of the last sensors are taken from both images. One of the images is rotated until the y -coordinates of these two images are the same. Figure 4.5(b) presents the obtained image after rotation. After rotation, according to the new coordinates of the last sensors which are the same y -coordinates, the difference of those x -coordinates is calculated. This difference is used for shifting the image in the x -axis therefore the transformation matrix for translation constituted with using this difference. In general, for translation, the transformation matrix is:

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix} \quad (4.5)$$

where t_x and t_y specifies the displacement along the x -axis and y -axis. After created a transformation structure (TFORM), that defines an affine transformation, the `imtransform` function of Matlab is used to apply the translation.

The basic Matlab algorithm can be given as:

```

 $t_x = \text{abs}(\text{firstImg}(1) - \text{secondImg}(1));$ 
 $T = [1 \ 0 \ 0 ; 0 \ 1 \ 0 ; t_x \ 0 \ 1];$ 
 $tform = \text{maketform}('affine', T);$ 
 $TransImage = \text{imtransform}(image, tform);$ 

```

After translation, the end points of these two images are the same. Then these images are combined as shown in Figure 4.5(c).

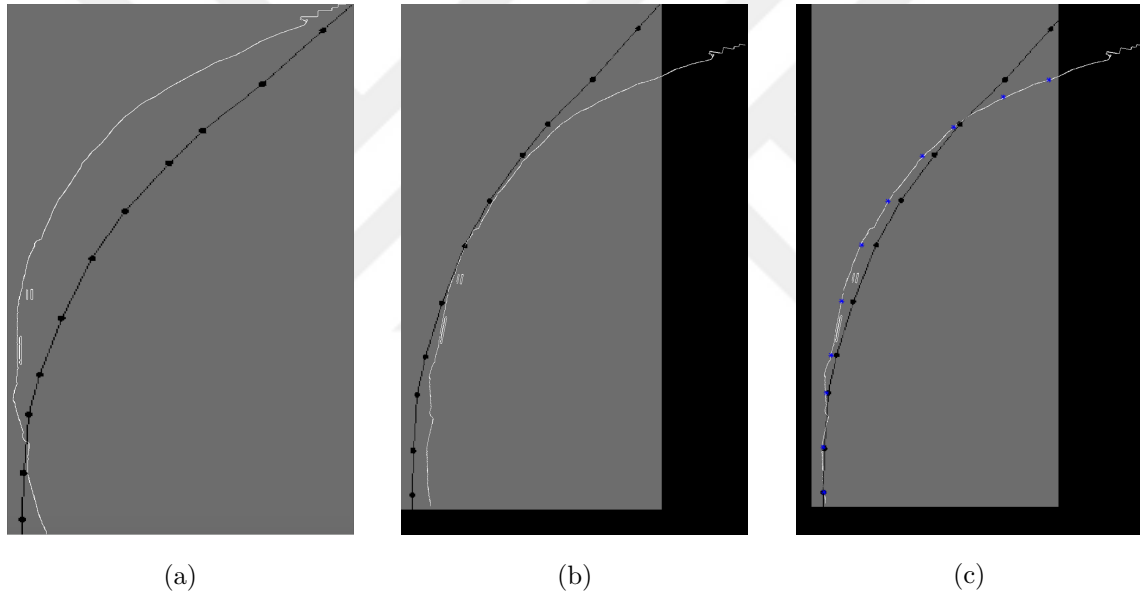


Figure 4.5: System analysis on spine: (a) Data visualization, black line, and subject's back curve, white line; (b) Applying the rotation method on data visualization; (c) The displacement of data visualization along the x-axis.

4.3 Data Collection

In order to measure the sensitivity of the flex sensors on different movements, we test different poses on two people. These poses are actually based on four main movements

which are cat position, child position, pelvic tilt and forward flexion. In total, we have eight different poses together with the initial body postures of these movements. For each pose, seventy data sequences are collected from each individual.

We analyze and classify these poses under subject dependent and independent settings. In subject independent case, there are 140 data sequences, in subject dependent case, 70 data sequences exist for each pose and each subject.

4.4 Statistical Analysis of Data

The aim of the statistical analysis is to determine the differences between poses. For this purpose, we use the Kullback-Leibler distance (KLD). The KLD is defined between two probability mass functions P and Q as,

$$D_{KL}(P \parallel Q) = \sum_i P(i) \log \frac{P(i)}{Q(i)}. \quad (4.6)$$

Our data includes 11 sensor readings, S_k for $k = \{1, 2, 3, \dots, 11\}$, and 8 spine poses, Q_m for $m = \{1, 2, 3, \dots, 8\}$. We can define pose conditional sensor reading probability mass functions (pmf) as $P(S_k|Q_m)$. These pmf can be approximated by the histograms, which are extracted for two sample poses of subject 1 as given in Figure 4.6.

To measure the statistical difference between readings of two poses, we use a symmetric KL distance as,

$$d_k(i, j) = 0.5(D_{KL}(P(S_k|Q_i)||P(S_k|Q_j)) + D_{KL}(P(S_k|Q_j)||P(S_k|Q_i))), \quad (4.7)$$

where $d_k(i, j)$ defines the statistical distance between poses i and j for the k -th sensor reading. We also define an average KLD over the sensor readings to have a global distance between poses i and j , as:

$$KLD_{avg}(i, j) = \frac{1}{k} \sum_k d_k(i, j) \quad (4.8)$$

The eight spine poses are defined in Table 4.1. The KLD distance $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, are plotted in Figure 4.7 and 4.8 for subjects 1 and 2, respectively.

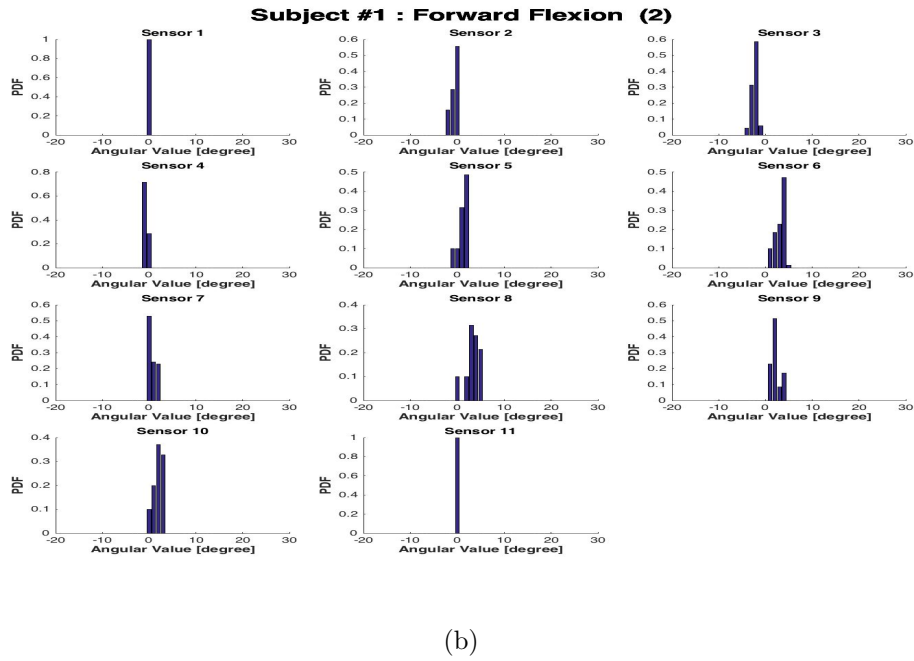
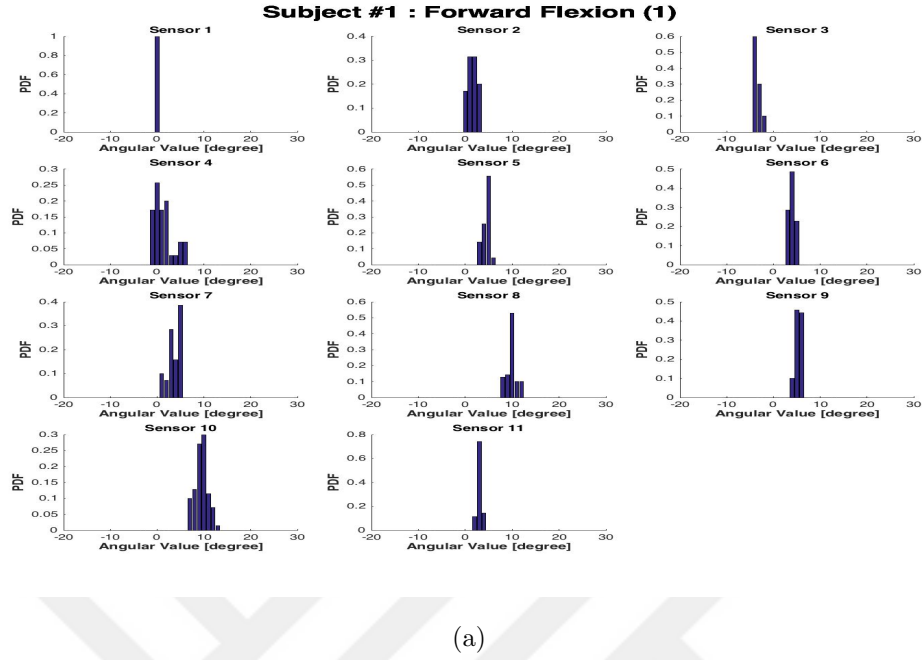


Figure 4.6: Histograms of 11 sensors for two sample poses of subject 1: Forward flexion 1 and 2

The diagonal elements of each matrix are equal to zero, since the two poses are the same. This follows from the fact that the statistical difference between readings of two poses approaches the zero as their similarities increase. The active part of the spine can be varied according to the poses. While the lumbar region of the body is more active in pelvic tilt (2), it is more passive than the upper part of the body in cat position (2). Figures 4.7, 4.8 and 4.9 present pose dependent sensor activities, which help us to discriminate poses and sensor activities. According to the flexibility of the spine, the sensor activities can change from person to person.

Table 4.1: The eight different spine poses

m	Q_m
1	Cat Position (2)
2	Cat Position (1)
3	Child Position (2)
4	Child Position (1)
5	Pelvic Tilt (2)
6	Pelvic Tilt (1)
7	Forward Flexion (2)
8	Forward Flexion (1)

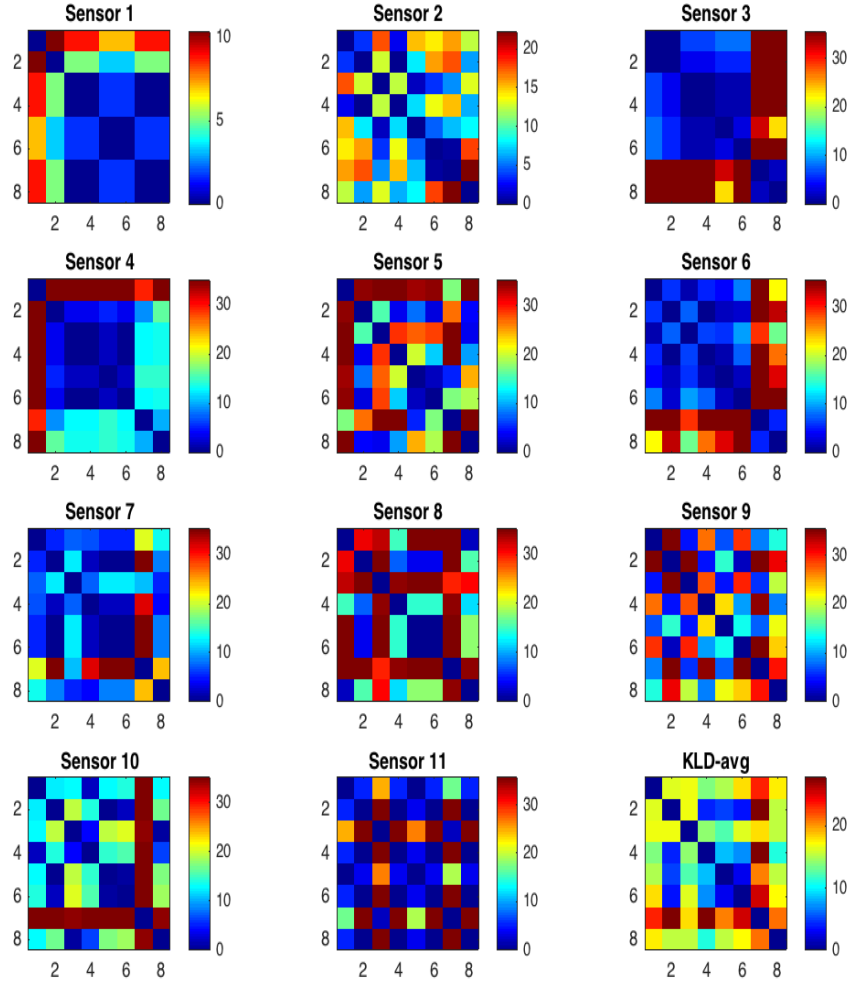


Figure 4.7: The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject 1.

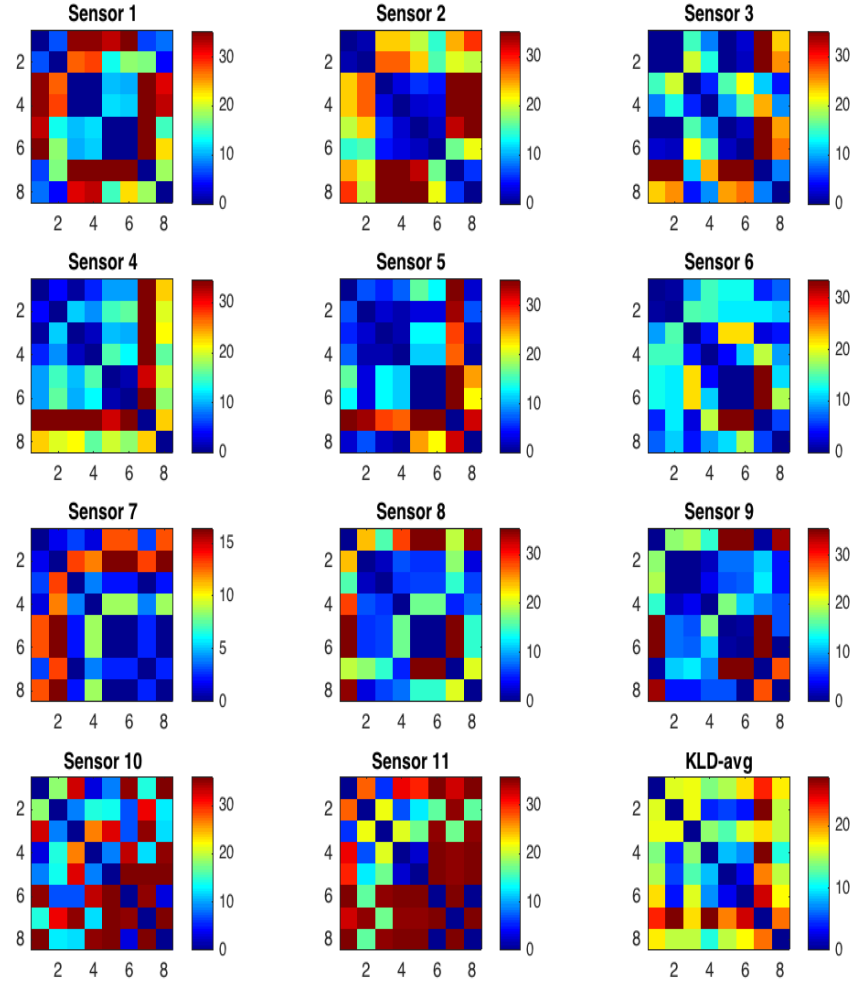


Figure 4.8: The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject 2.

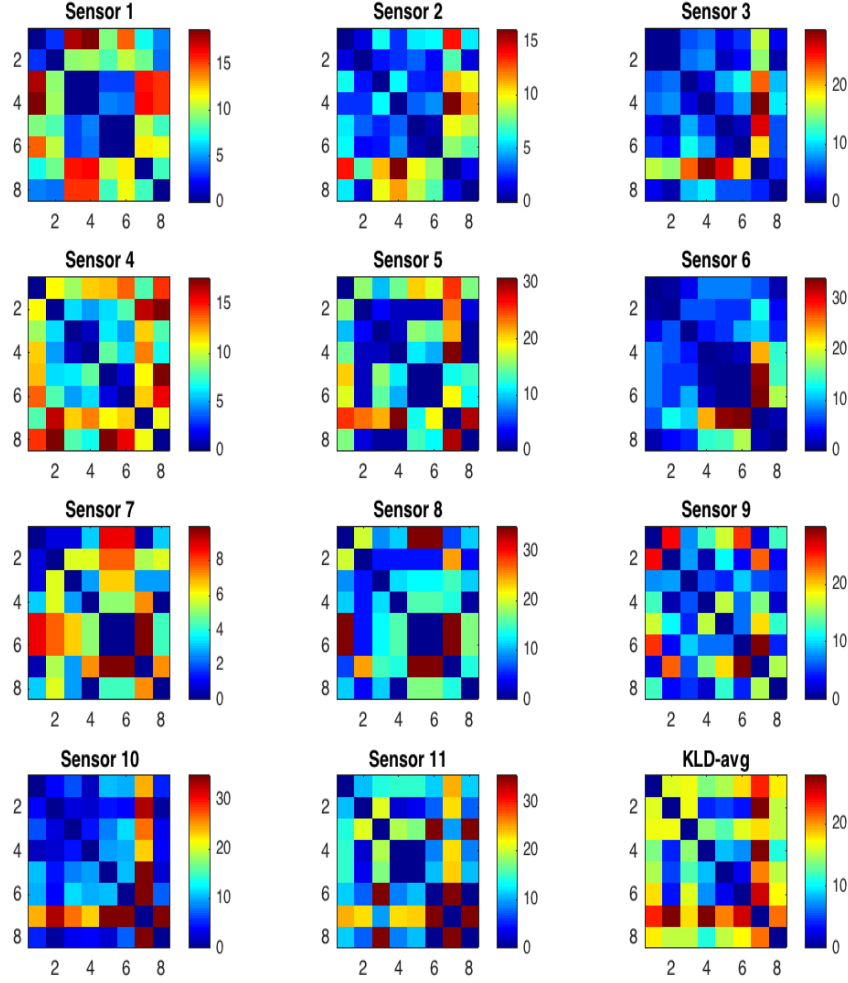


Figure 4.9: The KL distances $d_k(i, j)$, for each sensor k and the average distance, $KLD_{avg}(i, j)$, for subject independent setting.

4.5 Pose Classification Using Sensory Data

Support vector machine (SVM) with k-fold cross validation is used for classification of poses. Briefly, in k-fold cross-validation, the original data is randomly splited into k folds or subsets. Of the k folds, a single fold is used as testing data and the remaining (k-1) folds are used as training data. After this process is repeated k times, the k results are obtained. To produce a single estimation, these k results are averaged.

Eight pose, defined in Table 4.1, are classified according to subject dependent and independent settings. In order to determine the misclassified data, a confusion matrix is created. As can be seen in Tables 4.2, 4.3 and 4.4, there is a clear separation between poses. In Table 4.2, a small confusion occurs between cat position (1), Q_2 , and pelvic tilt (1), Q_6 . This means that 5 out of 70 data sequence are misclassified as pose 6 instead of pose 2 for subject 1. The average accuracy is measured as 97.68% and corresponding confusion matrix is provided in Table 4.2. In Table 4.3, a small confusion, 7.15%, is observed between cat position (1), Q_2 , and cat position (2), Q_1 . The average accuracy is 98.04% and Table 4.3 represents the confusion matrix for subject 2.

Some pose classification differences between subject dependent and independent settings can be observed. The pose 7, Q_7 , is classified with 100% accuracy in Table 4.2 and Table 4.3 and with 98.58% accuracy in Table 4.4. Pose differences across subjects is one possible reason of this difference. The average subject independent accuracy is measured as 96.88% and corresponding confusion matrix is provided in Table 4.4.

Table 4.2: The confusion matrix for subject 1

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
Q_1	100.0	0	0	0	0	0	0	0
Q_2	0	92.85	0	0	0	7.15	0	0
Q_3	0	0	100.0	0	0	0	0	0
Q_4	0	4.29	0	95.71	0	0	0	0
Q_5	0	0	0	0	100.0	0	0	0
Q_6	0	5.72	0	0	0	94.28	0	0
Q_7	0	0	0	0	0	0	100.0	0
Q_8	0	0	0	1.43	0	0	0	98.57

Table 4.3: The confusion matrix for subject 2

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
Q_1	97.14	0	0	0	0	0	0	2.86
Q_2	7.15	92.85	0	0	0	0	0	0
Q_3	0	0	100.0	0	0	0	0	0
Q_4	0	0	0	100.0	0	0	0	0
Q_5	0	0	0	0	100.0	0	0	0
Q_6	0	0	0	5.72	0	94.28	0	0
Q_7	0	0	0	0	0	0	100.0	0
Q_8	0	0	0	0	0	0	0	100.0

Table 4.4: The confusion matrix for subject independent

	Q_1	Q_2	Q_3	Q_4	Q_5	Q_6	Q_7	Q_8
Q_1	92.85	0	0	0	0	0	0	7.15
Q_2	0	91.14	0	0	0	0	0	8.86
Q_3	0.71	0	98.58	0	0	0	0.71	0
Q_4	0	0	0	100.0	0	0	0	0
Q_5	0	0	0	0	100.0	0	0	0
Q_6	0	0	0	0	0	100.0	0	0
Q_7	1.42	0	0	0	0	0	98.58	0
Q_8	0	1.42	0	0	0	0	0	98.58

Chapter 5

DISCUSSION AND CONCLUSIONS

In this thesis, we measured the angles of specific vertebral bones by using flex sensors in real time. In order to monitor the movement of spine, we developed the hardware and software systems. While the hardware system was comprised of a micro controller and flex sensors, the software system enabled to communicate between the hardware system and computer and visualize the obtained data.

In the hardware system, the circuit was designed by using the Proteus software and the circuit components are soldered on the PCB according to the circuit design. For the data acquisition process, a system was built to achieve flex sensor data in real-time by using the Arduino IDE. For data visualization process, the algorithm was implemented in the Java software to monitor the movement of spine in real time. Since we observed that the visualizations carry the noise, we implemented a low pass filter to reduce this noise on the data.

In order to make sensory system wearable, various techniques for placing sensory system and different types of vest designs and medical tapes were tested with similar spine movement patterns. The advantages and disadvantages of vest or medical tapes were explained. One of the medical tapes which gave the most accurate data was used in our experiments.

The angles of the sensors were calculated in Matlab according to pictures taken from the stick with the image processing method. To compare the sensor angles obtained from visual calculations and sensor readings, the RMSE was calculated for each experiment, for each degree and in five degree intervals according to the fifteen experiments. Also, we tested visualized data obtained from the Java software on human body posture. For this, the edge detection method was used to find the

boundaries of subject's back curve within the image. In order to overlap the two images properly, the rotation and transformation techniques were used.

In addition to forward flexion, totally eight different poses were collected from two people to measure the sensitivity of the flex sensors on different movements. We used a symmetric Kullback-Leibler distance (KLD) to determine the statistical differences between poses for the each sensor and also calculated the average KLD to have a global distance between poses. In order to classify these eight poses, support vector machine (SVM) was used. The data analysis and classification were applied under subject dependent and independent settings.

As future work, instead of medical tapes, a better vest design can be developed by a professional designer to make the system more practical. Therefore, the user can monitor and test the movement of spine on his/her own. The visualization part of the system can be improved and made in three dimensions instead of two dimensions. A hardware system can be designed to prevent the sensors from being damaged during a movement.

BIBLIOGRAPHY

- [1] Patricia Griegel-Morris, Keith Larson, Krissann Mueller-Klaus, Carol A Oatis, et al. Incidence of common postural abnormalities in the cervical, shoulder, and thoracic regions and their association with pain in two age groups of healthy subjects. *Physical therapy*, 72:425–425, 1992.
- [2] Sam Murphy, Peter Buckle, and David Stubbs. Classroom posture and self-reported back and neck pain in schoolchildren. *Applied ergonomics*, 35(2):113–120, 2004.
- [3] Afıtap İçağasıoğlu, Yasemin Yumuşakhuylu, Ayşegül Ketenci, Naciye Füsün Toraman, Gülçin Kaymak Karataş, Ömer Kuru, Yeşim Kirazlı, Kazım Çapacı, Esmâ Eriman, and Sema Haliloğlu. Burden of chronic low back pain in the turkish population. In *24th National Congress of Physical Medicine and Rehabilitation*, pages 58–64, 2013.
- [4] Qiang Li, John A Stankovic, Mark A Hanson, Adam T Barth, John Lach, and Gang Zhou. Accurate, fast fall detection using gyroscopes and accelerometer-derived posture information. In *Wearable and Implantable Body Sensor Networks, 2009. BSN 2009. Sixth International Workshop on*, pages 138–143. IEEE, 2009.
- [5] James A Levine, Lorraine M Lanningham-Foster, Shelly K McCrady, Alisa C Krizan, Leslie R Olson, Paul H Kane, Michael D Jensen, and Matthew M Clark. Interindividual variation in posture allocation: possible role in human obesity. *Science*, 307(5709):584–586, 2005.
- [6] Kamiar Aminian and Bijan Najafi. Capturing human motion using body-fixed

- sensors: outdoor measurement and clinical applications. *Computer animation and virtual worlds*, 15(2):79–94, 2004.
- [7] Alessandro Tognetti, Federico Lorussi, Raphael Bartalesi, Silvana Quaglini, Mario Tesconi, Giuseppe Zupone, and Danilo De Rossi. Wearable kinesthetic system for capturing and classifying upper limb gesture in post-stroke rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2(1):8, 2005.
- [8] Corinne Mattmann, Oliver Amft, Holger Harms, Gerhard Troster, and Frank Clemens. Recognizing upper body postures using textile strain sensors. In *Wearable Computers, 2007 11th IEEE International Symposium on*, pages 29–36. IEEE, 2007.
- [9] Danilo De Rossi, Federico Carpi, Federico Lorussi, Alberto Mazzoldi, Rita Paradiso, Enzo Pasquale Scilingo, and Alessandro Tognetti. Electroactive fabrics and wearable biomonitoring devices. *AUTEX Research Journal*, 3(4):180–185, 2003.
- [10] Yoshio Tsuchiya, Takashi Kusaka, Takayuki Tanaka, Yoshikazu Matsuo, Makoto Oda, Tsukasa Sasaki, Tamotsu Kamishima, and Masanori Yamanaka. Calibration method for lumbosacral dimensions in wearable sensor system of lumbar alignment. In *Engineering in Medicine and Biology Society (EMBC), 2015 37th Annual International Conference of the IEEE*, pages 3909–3912. IEEE, 2015.
- [11] Giancarlo Orengo, Antonino Lagati, and Giovanni Saggio. Bend sensors modelling for fast signal recovering in human motion analysis. In *Proceedings of the 3rd International Conference on Sensor Device Technologies and Applications (SENSORDEVICES), Rome, Italy*, pages 19–24, 2012.
- [12] FMA Hossain and Masuda A Tonima. A study on the effects of heavy backpack and development of a preventative prototype. *Journal of Sensors*, 2017, 2017.

- [13] Lucy E Dunne, Pauline Walsh, Sonja Hermann, Barry Smyth, and Brian Caulfield. Wearable monitoring of seated spinal posture. *IEEE transactions on biomedical circuits and systems*, 2(2):97–105, 2008.
- [14] Lucy E Dunne, Pauline Walsh, Barry Smyth, and Brian Caulfield. Design and evaluation of a wearable optical sensor for monitoring seated spinal posture. In *Wearable Computers, 2006 10th IEEE International Symposium on*, pages 65–68. IEEE, 2006.
- [15] Thomas Beth, Ingo Boesnach, Martin Haimerl, Jörg Moldenhauer, Klaus Bös, and Veit Wank. Characteristics in human motion—from acquisition to analysis. In *IEEE International Conference on Humanoid Robots*, pages 56–75, 2003.
- [16] Thomas B Moeslund and Erik Granum. A survey of computer vision-based human motion capture. *Computer vision and image understanding*, 81(3):231–268, 2001.
- [17] Giovanni Saggio, Francesco Riillo, Laura Sbernini, and Lucia Rita Quitadamo. Resistive flex sensors: a survey. *Smart Materials and Structures*, 25(1):013001, 2015.
- [18] Giancarlo Orengo, Antonino Lagati, and Giovanni Saggio. Modeling wearable bend sensor behavior for human motion capture. *IEEE Sensors Journal*, 14(7):2307–2316, 2014.
- [19] Ryan Keyfitz. Design of a range of motion sensor for ankle rehabilitation monitor. 2010.
- [20] Pawel Pyk, David Wille, Edith Chevrier, Yves Hauser, Lisa Holper, Ismael Fatton, Roger Greipl, Sonja Schlegel, Luzia Ottiger, Barbel Ruckriem, et al. A paediatric interactive therapy system for arm and hand rehabilitation. In *Virtual Rehabilitation, 2008*, pages 127–132. IEEE, 2008.

- [21] P Daponte, J De Marco, L De Vito, B Pavic, and S Zolli. Electronic measurements in rehabilitation. In *Medical Measurements and Applications Proceedings (MeMeA), 2011 IEEE International Workshop on*, pages 274–279. IEEE, 2011.
- [22] Saba Bakhshi, Mohammad H Mahoor, and Bradley S Davidson. Development of a body joint angle measurement system using imu sensors. In *Engineering in Medicine and Biology Society, EMBC, 2011 Annual International Conference of the IEEE*, pages 6923–6926. IEEE, 2011.
- [23] Luis M Borges, Norberto Barroca, Fernando J Velez, and Antonio S Lebres. Smart-clothing wireless flex sensor belt network for foetal health monitoring. In *Pervasive Computing Technologies for Healthcare, 2009. PervasiveHealth 2009. 3rd International Conference on*, pages 1–4. IEEE, 2009.