DOKUZ EYLÜL UNIVERSITY
GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES

GESTURE ANALYSIS AND RECOGNITION BASED ON MEMS SENSORS

by
Sevda AYDOĞAN

February, 2020
İZMİR

GESTURE ANALYSIS AND RECOGNITION BASED ON MEMS SENSORS

A Thesis Submitted to the
Graduate School of Natural and Applied Sciences of Dokuz Eylül University
In Partial Fulfillment of the Requirements for the Degree of Master of Science
in Mechatronics Engineering Program

by
Sevda AYDOĞAN

February, 2020
İZMİR

M.Sc THESIS EXAMINATION RESULT FORM

We have read the thesis entitled **"GESTURE ANALYSIS AND RECOGNITION BASED ON MEMS SENSORS"** completed by **SEVDA AYDOĞAN** under supervision of **ASSOC. PROF. DR. YAVUZ ŞENOL** and we certify that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

....

Assoc. Prof. Dr. Yavuz ŞENOL

Supervisor

Prof. Dr. Mustafa Gündüzalp

(Jury Member)

Assoc. Prof. Dr. Ahmet Özkurt

(Jury Member)

Prof. Dr. Kadriye ERTEKİN

Director

Graduate School of Natural and Applied Sciences

ACKNOWLEDGEMENTS

Firstly, I would like to thank to my advisor Assoc. Prof. Dr. Yavuz ŞENOL for his support and guidance.

I am grateful to my precious friend Ahmet ÖZKAYA for helping and motivating me at every stage of my work. If he had not supported me, this study would not be completed.

Also, I would like to thank my spiritual sister Sultan AKTİN for her support.

Most importantly, I would like to thank my beloved nephew Göktuğ MAZLUM who added meaning to my life for loving me. His love motivated me throughout my work.

Finally, I wish to thank to my parents and my siblings for their physical and moral support throughout my education and private life.

Sevda AYDOĞAN

GESTURE ANALYSIS AND RECOGNITION BASED ON MEMS SENSORS

ABSTRACT

This study covers the use of Micro-Electro-mechanical system (MEMS) sensors and communication with Bluetooth to transmit sensor data in order to understand the shape drawn into the air. It is important for people, especially children with speech disabilities, to be understood by the characters such as commands, shapes, letters or numbers that are given by their hand or arm movements. For this purpose, the MPU6050 which is type of inertial measurement unit, was used as a MEMS sensor to capture dynamic hand movement. This device is preferred because it is a small integrated structure that transfers data from the gyroscope and accelerometer sensors inside to the computer via Bluetooth. They can be placed easily on the hand and arm.

In this study, some numbers, letters and geometrical shapes, which were generated by hand movements, have been recognized using hidden Markov model algorithm in MATLAB. The obtained data is transferred to a computer by means of wireless communication. The empirical studies have shown that the system can successfully recognize the generated gestures.

Keywords: Inertial measurement unit, Bluetooth, gesture analysis, MEMS, hidden Markov model

MEMS SENSÖRLERİNE DAYALI HAREKET ANALİZİ VE TANIMA

ÖZ

Bu çalışma, havaya çizilen şekilleri anlamlandırmak üzere Mikro-Elektro-Mekanik Sistem (MEMS) sensörlerinin kullanımını ve sensör verilerini aktarmak için Bluetooth haberleşmeyi kapsamaktadır. Kişilerin, özellikle konuşma engelli çocukların el, kol hareketleriyle vereceği komut, şekil, harf veya rakam gibi karakterlerin anlamlandırılması önemlidir. Bu amaçla dinamik el hareketini yakalamak için MEMS sensörü olarak bir tür atalet ölçü birimi olan MPU6050 kullanılmıştır. Bu cihaz, içinde dahili olarak bulunan jiroskop ve ivme ölçer sensörlerinden alınan verileri Bluetooth üzerinden bilgisayar ortamına aktaran tümleşik küçük bir yapıda olması sebebiyle tercih edilmiştir. Onlar el ve kol üzerinde kolayca yerleştirilebilecektir.

Bu çalışmada, el hareketleriyle oluşturulan bazı sayılar, harfler ve geometrik şekiller MATLAB'da gizli Markov model algoritması kullanılarak tanımlanmıştır. Elde edilen veriler kablosuz iletişim yoluyla bir bilgisayara aktarılır. Deneysel çalışmalar, sistemin oluşturulan hareketleri başarıyla tanıyabildiğini göstermiştir.

Anahtar kelimeler: Atalet ölçü birimi, Bluetooth, hareket analizi, MEMS, saklı Markov model

CONTENTS

LIST OF FIGURES

LIST OF TABLES

Page

CHAPTER ONE
INTRODUCTION

Since communication between people takes place both verbally and through body language, sometimes it may be necessary to understand body language instead of verbal communication. For this purpose, human computer interaction (HCI) technology is used to understand the communication between computer and human (Pradipa & Kavitha, 2014). Hand gestures are the most basic element of body language. Communication with this technology has gone one step further, and in many areas such as medical systems and entertainment, hand gesture recognition systems have become more widespread (Li, n.d.).

Gesture detection and recognition can be expressed as a change in the environment of an object or a change in the position of an object relative to a referenced point. Gesture recognition was first used to give simple commands to the computer with the help of a glove based on sign language. These operations were first carried out with the help of sensors such as accelerometer and gyroscope which were mounted on the glove and then with the help of the camera without attaching anything to the glove. One of the first glove studies in the field of motion recognition was Sayre Glove, which was built on the foundation of finger bending in 1977. With the help of a photodiode, the glove was able to detect finger movement using the voltage variation between the photocell and the light source placed on each finger. The first data glove, which was then used with sensors, was developed in 1983 by Gary Gimes. In 1989, Nintendo made power glove by improving the data glove to control the game console in Figure 1.1 (Premaratne, 2014).

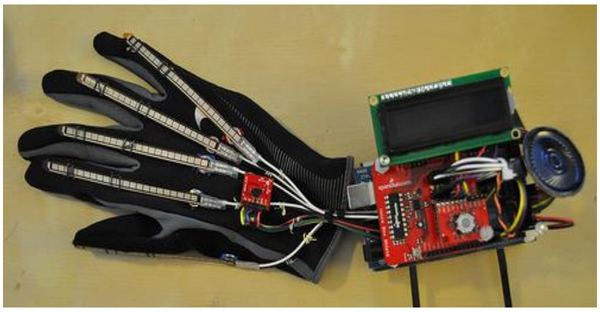

Figure 1.1 Power glove (Grady, 2010)

In paper, gesture recognition and analysis, by using gloves, is still evolving today. In this scope, the accelerometer and Arduino were mounted on a wearable glove. The data set of the Turkish capital letters drawn into the air was transferred to the computer via USB and gesture recognition was performed using random forest algorithm (Ecer, Yetgin, & Celik, 2018).

In addition to gesture recognition with gloves, motion recognition can also be performed using only the camera. In this context, depth (Kinect) camera was used to introduce the alphabets, as well as mathematical operations such as adding and subtracting numbers from zero to nine by hand. A success rate of over 95% was achieved (Murata & Shin, 2014). Such studies show that gesture recognition is successfully performed without the need for a glove or other device such as a sensor.

The sensors can be used in many ways. In this project, mobile phones can be considered as an example of this. In previous studies, using the accelerometer sensor in the mobile phone, the phone was held as a pen then written letters and words in the air. Over 90% success was obtained (Agrawal et al., 2011).

In the article (Meenaakumari & Muthulakshmi, 2013), MEMS accelerometer, wireless for communication and microcontroller were used for motion recognition. In order to obtain letters and numbers, they made sense of the characters by limiting their hand movements to the right, left, up and down. In this way, they aimed to increase the success rate and achieved more than 98% success.

In the study (Tuncer, 2016), acceleration sensor was used for gesture recognition on the English alphabet. When letters are written by different people at different times, DTW algorithm is used to minimize the differences between two signals of the same letter. The data obtained from the acceleration sensor has been processed and over 95% success has been achieved.

As a result, gesture recognition can be obtained by electronic or mechanical means and human movements can be interpreted with the help of algorithms. This process can be performed with devices such as camera, sensor and simple hand or body movements in order to control or communicate the systems without touching the devices.

In this project, in order to recognize hand movement, by using accelerometer and gyroscope, geometric shapes, numbers and alphabets were written into the air and the data was transferred to computer via Bluetooth and a training set was created. This training set was trained with hidden Markov model (HMM) algorithm and then a test data was processed and a study was made to estimate which character or shape it is. In previous studies have generally been seen to focus only on letters, numbers, and shapes.

CHAPTER TWO
GESTURE RECOGNITION AND ALGORITHMS

2.1 Gesture Recognition

The word gesture in gesture recognition can be considered as a non-verbal form of communication (Schechter, 2014). Gesture can be done with the hand or the human body’s motions. These motions are read using sensors or cameras and the collected data is sent to a computer. The data obtained are processed by the system and interpreted mathematically. Gestures are made sense using gesture recognition algorithms and process is realized. Gesture recognition is a type of the perceptual user interfaces (PUI). Other PUIs are voice recognition, retina reading, face recognition (Gesture Recognition, n.d.).

In many projects, during the gesture recognition, analysis and classification stages, as shown in the Figure 2.1, firstly data is obtained and gesture modeling is made and then features of data are extracted and process of gesture recognition or classification is completed (Kumar, 2014).

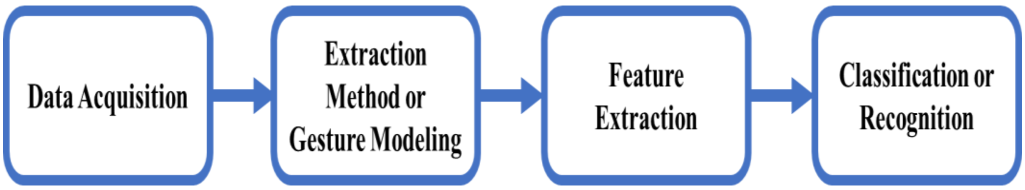

Figure 2.1 Simple block diagram of gesture analysis and recognition

2.1.1 Data Acquisition

At this stage, the data set is gathered for training the algorithm. The device to be used for data set collection is selected depending on the data type. For example, if the data is voice, hand or body recognition, a microphone, acceleration sensor or depth camera can be used respectively (Kumar, 2014).

2.1.2 Extraction Method or Gesture Modeling

Gesture modeling is an important step in the correct conclusion of the gesture recognition process and the aim of this step is to ensure that the Tested Data is not affected by environmental conditions, such as size, noise, and vibration. Therefore, to achieve the targeted movement, it may be necessary to reduce noise on data or to classify of gesture. After these procedures, more accurate results will be obtained in gesture recognition (Kumar, 2014).

2.1.3 Feature Extraction

At this stage, the process should not be affected by the differences in the data in order to obtain correct results from the gesture recognition process. Therefore, it should be well chosen which points to focus on the voice, hand, face or body from which the data is obtained. Moreover, the characteristics of the collected data must be well established, since these features can be used in the training of algorithms used in gesture recognition (Kumar, 2014).

2.1.4 Classification or Recognition

In recognition process which is the last step should clearly state what the movement means. It compares a test data with a trained data set and includes it in the class of the nearest data set (Kumar, 2014). There are some methods for clustering and classification. In this project, K-Means algorithm is used as a clustering algorithm and KNN algorithm is used for classification. K-Means and KNN algorithms are in MATLAB.

Mathematically, K-Means algorithm divides the data set with different properties into clusters and performs the creation of new clusters according to the center points for each class that occurs. The steps of the algorithm are described below.

1- Determination of cluster centers,

2- Classification of samples according to their distance from the center,

3- Determination of new centers according to classification,

4- Repeat steps 2 and 3 until it becomes stable (Şeker, 2008).

Also, KNN is used in this project and it uses cluster centers. KNN is a classification method, which classifies the test data by looking at its nearest k neighbor.

2.2 Algorithm Techniques of Gesture Recognition

In recent years, technology for gesture recognition has been developed rapidly and many projects are being carried out in this field. Many different techniques are used for gesture recognition in these projects. These algorithms and techniques used in gesture analysis can be preferred for classification and clustering according to their weaknesses or strengths. Some of these are briefly described below (Al-Bayaty, 2015).

- ✓ Support Vector Machine (SVM)
- ✓ Artificial Neural Networks (ANN)
- ✓ Random Forest Classification (RF)
- ✓ Dynamic Time Warping (DTW)
- ✓ Hidden Markov Model (HMM)

2.2.1 Support Vector Machine

The SVM algorithm is a simple yet very useful method of classification. It uses supervised learning and so is a good pattern recognition classification technique (Nagshree et al., 2015). SVM, by making this classification between two groups in a plane, creates a model that predicts which category the test data will fall into. One of two group expresses different parts of examples. These groups in the plane are formed by obtaining a different point corresponding to each input after the feature extraction process of each input entering the system (Şeker, 2008).

2.2.2 Artificial Neural Networks

ANN is a classification algorithm used in gesture recognition and is a method that can adapt to changing information in and around the structure. The basis of this method is observation.

ANN has disadvantages and advantages as other algorithms. One of the advantages of this algorithm is that it can work without problems even if some of the individual neurons of the network are broken, so it is fault tolerant. ANN only needs to learn once and can work in any application. As a disadvantage, ANN needs to be trained to work. It may also require a long time for big neural network data (Dhinakaran, 2014).

2.2.3 Random Forest Classification

Another classification method for gesture analysis is Random Forest. This algorithm is a community learning method. A training set is available here and random sets are formed by selecting random samples from this training set. After a trained decision tree is formed with random data sets, the algorithm process is completed by consulting the decision trees again (Ecer, Yetgin, & Celik, 2018).

2.2.4 Dynamic Time Warping

DTW algorithm is also used to detect hand gesture recognition. That is, the data transferred to the computer environment is expressed using the DTW algorithm. The reason for using the DTW algorithm in some projects is that even when the same person draws the same shape at different times, signals of different length may be generated. In this case, the similarity between two signals is found by the DTW method. DTW is more sensitive to alignment of the two signals than the linear method used in previous studies (Tuncer, 2016).

2.2.5 Hidden Markov Model

HMM algorithm is one of the classification techniques used in gesture recognition. This theorem was first studied in the 1940s but it was developed later by Baum, Eagon and Petrie in the 1970s (Sezen, 2014). Nowadays, the HMM is used in many fields such as hand, body movement recognition systems, gene estimation, speech recognition and economic calculations, machine translation, robotics, cryptoanalysis.

HMM is a Bayesian network which is superior to other Bayesian classification techniques.

It is a disadvantage that the movements in the HMM method must be both known and static. It reduces the effect of the HMM if the user wants to add more movement to the first model. Other disadvantage is that they need a lot of data sets to train (Al-Bayaty, 2015).

The purpose of HMM is to predict the future situation of the system from the present situation (Sezen, 2014). It is a model where emission observation can be made in the system but the order of transition between states is not known to generate emissions. It is aimed that sequences of states are independent from the observed data with HMM algorithm (HMM, n.d.).

According to the Figure 2.2, the state at time t depends only on the time t-1, that is, on the previous case, and it does not depend on the conditions (t-2) that took place at times before t-1. Thus, the observed state y (t) depends only on the hidden state x (t). The conditions before it’s had no effect. As other example the state at time t+1 doesn't depend on the state at time t-1. This is known as the Markov property.

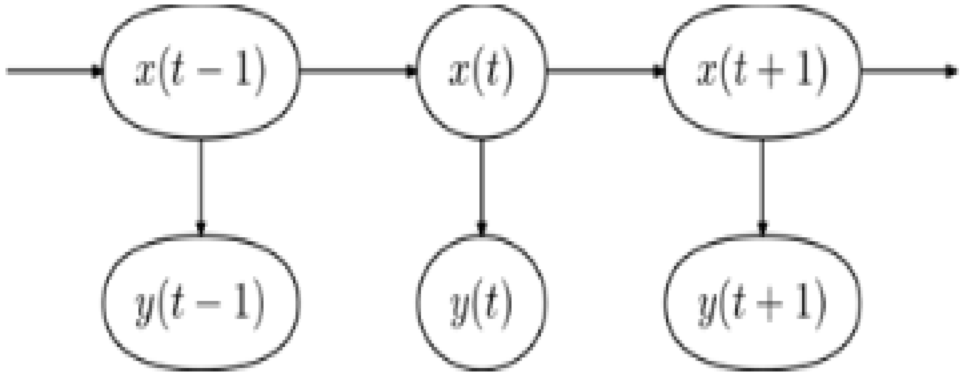

Figure 2.2 Markov model (Sezen, 2014)

The Markov model can predict its next step in uncertainty based on the current state of both social and natural phenomena. There are two types of Markov technique, normal and hidden. In the normal Markov model, states can be seen and only the parameter used is the state transition probability. In the hidden Markov model, situations are not visible, but the outputs affected by the situation can be observed. The parameters of this model are states, possible observations, transition probabilities status and output probabilities (Sezen, 2014). These parameters can be seen in Figure 2.3.

a: Transition probability

b: Output probability

x: Hidden states (It is used as S in formulas.)

y: Observations (It is used as v in formulas.)

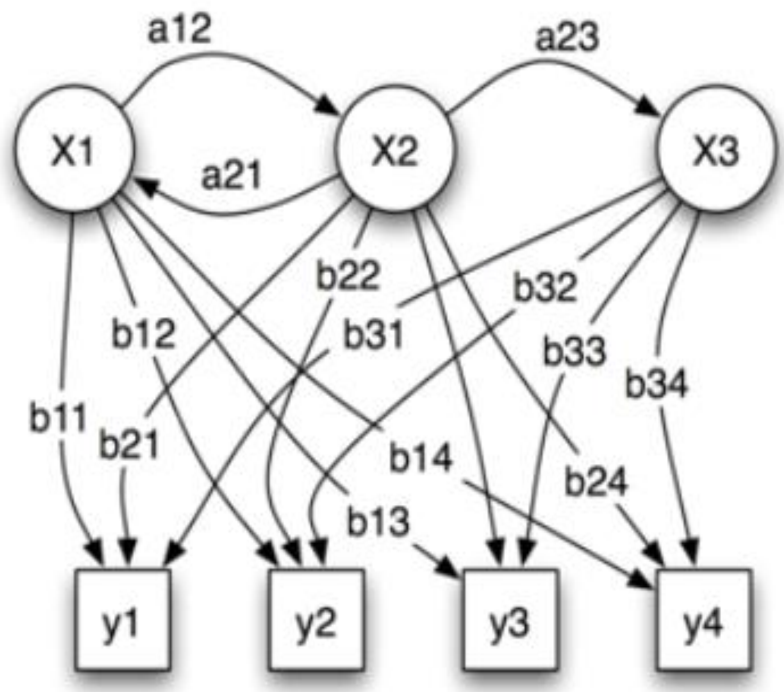

Figure 2.3 Example of hidden Markov model (Sezen, 2014)

2.2.5.1 Features of HMM

- The number of states in the system is indicated by N. Although situations are hidden, they may have some physical meaning. You can also switch from one situation to another. S= {S_1, S_2, S_3, ..., S_N} is the state of the model and q_t is the state at the time t.
- M is the number of observations observed for each case. When the observations (v = {v_1, v_2, v_3, ..., v_M}) are continuous, the number of observations will be infinite (Öcal, 2005).
- State transition probability distributions:

$$A=\{a_{ij}\} \rightarrow a_{ij}=P[q_{t+1}=S_j|q_t=S_i],\ 1\leq i,j\leq N \quad (2.1)$$

- Observation probability distributions of states:

$$B=\{b_j(k)\} \rightarrow b_j(k)=P\ [v_k \text{ at } t|q_t=S_j],\ 1\leq j\leq N \text{ and } 1\leq k\leq M \quad (2.2)$$

- Initial state distribution:

$$\pi=\{\pi_i\} \text{ and } \pi_i=P[q_1=S_i],\ 1\leq i\leq N \tag{2.3}$$

It is important to find the most suitable hidden state in gesture recognition. Before using the HMM algorithm in projects, firstly it is necessary to solve the following basic problems of HMM.

1. **The Evaluation Problem:** Given $O=O_1, O_2, O_3, \ldots, O_T$ sequence of observations and model $\lambda=\{A, B, \pi\}$, how is the probability of observation sequence's $P(O|\lambda)$ calculated (Öcal, 2005)?

The solution to this problem allows us to select the most suitable model for observation from a few models. So, it shows how compatible the observation is with the model (HMM_problems, n.d.).

2. **The Decoding Problem:** Given sequence of observations ($O=O_1, O_2, \ldots, O_T$) and model $\lambda=\{A, B, \pi\}$, how is state sequence ($Q=q_1, q_2, q_3, \ldots, q_T$) corresponding to this observation sequence calculated (Öcal, 2005)?

In this section, it is necessary that hidden states are found. Usually, the correct state sequence can’t be found, but the best state can be found (HMM_problems, n.d.).

3. **The Learning Problem:** Given sequence of observations ($O=O_1, O_2, \ldots, O_T$) how are maximum $P(O|\lambda)$ and model $\lambda=\{A, B, \pi\}$ calculated (Öcal, 2005)?

Model parameters are estimated from the observation series herein. Observation series are called training series and this process is called HMM training process (Öcal, 2005).

HMM has algorithms such as Viterbi, Baum-Welch and Forward algorithms for recognition and learning (Kumar, 2014). Respectively, problem 1, problem 2 and problem 3 can be solved with Forward-Backwards algorithm, Viterbi and Baum-Welch algorithm (HMM_problems, n.d.).

2.2.5.2 Forward-Backward Algorithm

Using the Forward algorithm, the order of the states in the model are found (Sezen, 2014) and also the probability of occurrence of observed states in a model is calculated. With the forward algorithm, the probability of occurrence of an observation is calculated for all states at the start. Backward algorithm is vice versa of Forward algorithm (Ayaz & Alp, 2018). [38]

$$\alpha_t(i) = P(O_1 O_2 \ldots O_T, q_t = S_i | \lambda) \tag{2.4}$$

$$\alpha_1(i) = \pi_i b_i(O_1), \; t=1 \text{ and } 1 \leq i \leq N \tag{2.5}$$

The transition from the states (D_i) at time t to the state (D_j) at the next time (t+1) is calculated with α_{1+1}(j).

$$\alpha_{1+t}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{i+1}) \tag{2.6}$$

$$T = 1, 2, \ldots, T\text{-}1 \text{ and } 1 \leq j \leq N \tag{2.7}$$

Finally, P ($O_t|\lambda$) is calculated with Equation 2.16.

$$P(O_t|\lambda) = \alpha_T(i) = P(O_1 O_2 \ldots O_T, q_t = S_i | \lambda) \tag{2.8}$$

2.2.5.3 Viterbi Algorithm

The Viterbi algorithm was designed by Andrew Viterbi to correct errors in 1967. It was developed in the following years.

The Viterbi algorithm can find the suitable hidden state sequence that makes the output sequence gathered by using the parameters of the model (Fang, 2009).

With the initial probability value of each case multiply the probability value of the first observation. In this case, the variable $\Psi_i(i)$, which specifies the maximum argument, equals zero (Ayaz & Alp, 2018).

$$\delta_1(i) = \pi_i b_i(O_1) \text{ and } \Psi_i(i)=0,\ 1 \le i \le N \tag{2.9}$$

The $\delta_{t-1}(i)$ values obtained for all separately cases are multiplied by the transition probability. The maximum value in these multiplication values is then multiplied by the probability of the current observation. As a result, the state with the maximum value is assigned to the variable $\Psi_i(i)$ (Ayaz & Alp, 2018).

$$\delta_t(i) = \max_{1 \le i \le N}\left[\delta_{t-1}(i)a_{ij}\right]b_j(O_t),\ t=2,3,\ \ldots,\ T \text{ and } 1 \le j \le N \tag{2.10}$$

$$\Psi_i(i) = \operatorname*{argmax}_{1 \le i \le N}\left[\delta_{t-1}(i)a_{ij}\right],\ t=2,3,\ \ldots,\ T \text{ and } 1 \le j \le N \tag{2.11}$$

The maximum of the $\delta_t(i)$ values calculated for the last observation is assigned to P*. Symbol q_T^* shows the state where the maximum selected $\delta_t(i)$ comes from. This gives the optimum state (Ayaz & Alp, 2018).

$$P^* = \max_{1 \le i \le N}[\delta_t(i)] \text{ and } q_T^* = \operatorname*{argmax}_{1 \le i \le N}[\delta_t(i)] \tag{2.12}$$

In the last step, from the last observation is proceeded backwards to the first observation. At the end of this process, the sequence of states that the q_T^* variable receives shows the optimum state sequence (Ayaz & Alp, 2018).

$$q_T^* = \Psi t{+}1(i)\ (\ q_{t+1}^*),\ T=T-1,\ T-2,\ T-3,\ \ldots,\ 1 \tag{2.13}$$

2.2.5.4 Baum-Welch Algorithm

The Baum-Welch algorithm is similar to the Forward algorithm and calculates the probabilities of observations by passing the sequence of observations from start to finish and vice versa. So more precise results can be found (Sezen, 2014).

For the S_i status at time t = 1, the estimate of the initial state distribution at the expected frequency is calculated as $\bar{\pi}=\gamma_1(i)$. Then $\overline{P}_{ij}$ and $\overline{\emptyset}(k)$ are found. Model is total of $\overline{P}_{ij}$ and $\overline{\emptyset}(k)$. Model is showed $\overline{\lambda} = (\overline{P}, \overline{\emptyset}, \bar{\pi})$ and this process continues until the new model approaches the old model (Ayaz & Alp, 2018).

Finally, when started value δ is bigger than $P[O|\overline{\lambda}]-P[O|\lambda]$ the process ends.

CHAPTER THREE

MATERIALS

In this study, it is important for people, especially children with speech disabilities, to be understood by the characters such as commands, shapes, letters or numbers that are given by their hand or arm movements like in Figure 3.1. In general, purpose of this study is to use a system that software and hardware parts working as a whole with the wireless data transfer from the sensor.

Figure 3.1 Writing in the air (Personal archive, 2019)

Nowadays, the speed of technology is increasing day by day and makes our lives much easier. Software and hardware together are used in projects and so the work of many things around us has become automatic. For example, turning on the lights when it gets dark, activating the fire alarm when there is smoke, and turning on the air conditioner when the house is hot, etc. These systems make sense of external signals and react accordingly.

There are many applications like fire detector, weather prediction, park detector, pressure detector etc. Sensors are very important in people life monitoring and detection of dangerous events. Therefore, before using a sensor we must understand what exactly a sensor does, then we should use it.

3.1 What is Sensor?

Sensors are a part of automatic control systems that provide connection with the outside world. Similar to the way people perceive what is happening around them with sensory organs, systems detect temperature, pressure, speed and such similar values through their sensors. After perceiving variable pressure or heat of systems, sensors send the detected input to a microcontroller or a microprocessor as seen in Figure 3.2 and Figure 3.3.

Figure 3.2 Working principle of sensors

As a result of using sensors in electronic and mechanical systems as first step to generate related data, they became an essential part of automated systems in everyday life. As they are the first step in electronic or mechanical systems, they generate an output signal by interpreting the input signals from other systems and this way they can also provide communication between the systems.

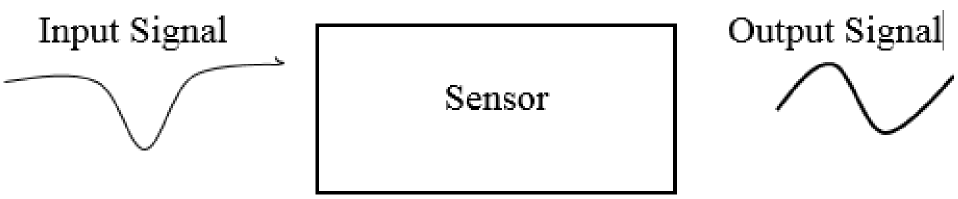

Figure 3.3 Working of sensor

3.1.1 Types of Sensors

Different sensors are used in varied applications. Sensors are used to measure the physical properties like pressure, light, heat, distance, temperature…

Some kinds of sensors;

- ✓ Temperature Sensor
- ✓ Accelerometer
- ✓ LDR
- ✓ Pressure Sensor
- ✓ Ultrasonic Sensor
- ✓ Gas Sensor
- ✓ MEMS
- ✓ Touch Sensor
- ✓ Color Sensor
- ✓ Potentiometer
- ✓ Flex Sensor
- ✓ IR Sensor

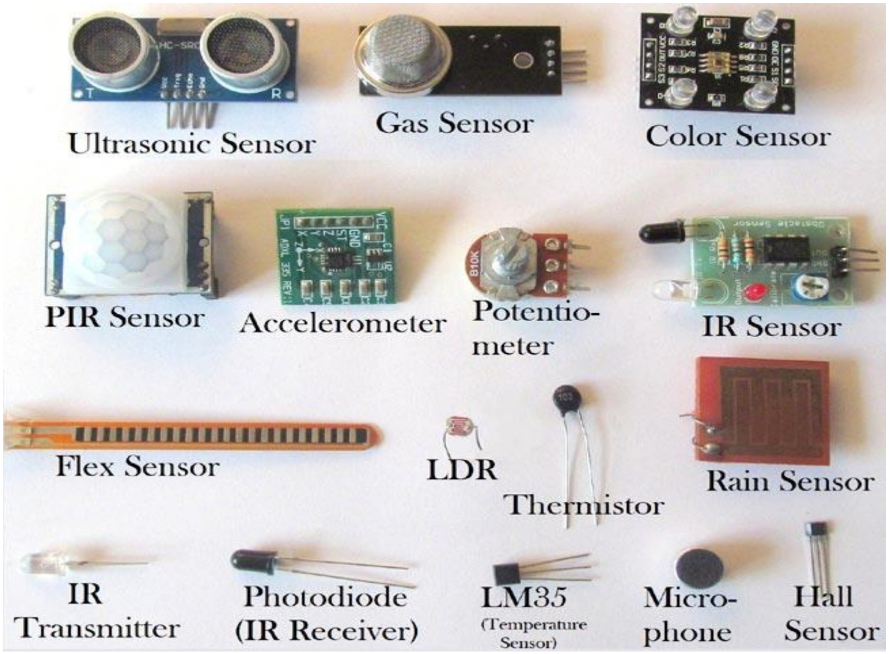

Figure 3.4 Types of Sensors (Thonti, 2018)

Acceleration: Acceleration is the derivative of velocity over time according to the laws of physics and it is a vector quantity. The international unit of acceleration is meter/second^2.

Acceleration also gives the change of both the direction and the velocity of the object over time and is obtained by the expression $a = F/m$. In equation, F is the force and m is the mass of the matter.

Accelerometers Sensor: Accelerometer is used to detect changes in position, velocity, and vibration of object (Vidya, n.d.). It measures the gravity on the object or the acceleration of the object at the moment of sudden acceleration or stop (Samancı, 2011). There are analogue and digital type of it. Analog accelerometers produce a continuous output voltage according to change in acceleration. But digital accelerometers usually provide pulse width modulation (Dimensionengineering, n.d.).

Gyroscope Sensor: Gyroscope is a kind of sensor that can detect angular velocity. Based on the centrifugal principle, it can determine the velocity and direction of the object by comparing the angular ratios on the three axes (Polat, n.d.). The gyroscope is often used in phones, cameras or tablets to determine directions (Aydınoğlu, 2015).

The difference between accelerometers and gyroscope sensors is that the gyroscope detects rotation, while the accelerometer cannot. This means that while the accelerometer cannot detect rotation without acceleration, the gyroscope measures any rotation and is not affected by acceleration (Vidya, n.d.).

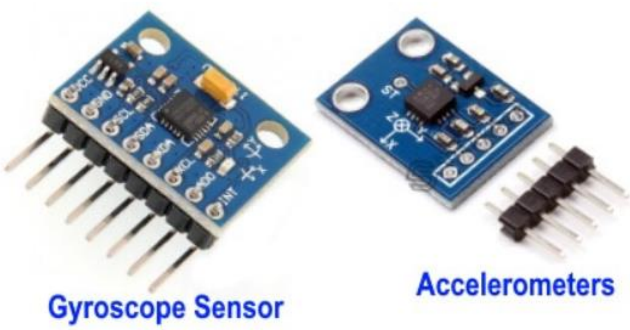

Figure 3.5 Gyroscope sensor and accelerometers (Vidya, n.d.)

Nowadays the accelerometer and gyroscope are used in many areas like agricultural machinery, solar energy, medical instrument, power monitoring, geological monitoring (Wit motion, n.d.), ships, missiles, aircraft, computer and phones systems (Toktaş, 2014). Some of them are given in Figure 3.6.

Figure 3.6 Usage areas of Gyroscope sensor and Accelerometers etc. (Wit motion, n.d.)

In this project, MEMS and motion sensors are emphasized. In the next section, the MPU6050 sensor, a combination of acceleration and gyroscope sensor, and MEMS will be discussed in detail.

3.2 Used Devices for Gesture Recognition

In recent years, many projects have been carried out that make sense of hand or body movement as a result of the advancement of technology and the need for innovations that make life easier. In these projects, a camera or depth camera have been used to display body and hand movements, but an acceleration sensor has often been used to detect movements without a camera (Murata & Shin, 2014).

In general, it has been observed that the communication between the computer and Arduino systems is provided via USB (Ecer, Yetgin, & Celik, 2018). Many different methods are used in the writing studies in the air. Some of these methods are as follows:

- ✓ Smart phone
- ✓ Microsoft Kinect camera
- ✓ Glove with accelerometer and Arduino
- ✓ MEMS (MPU6050)

As shown in Figure 3.7, the acceleration sensors of the phone are used (Agrawal et al., 2011).

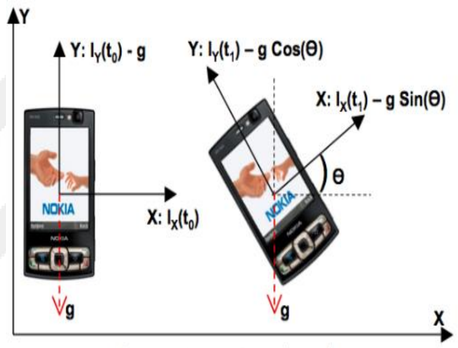

Figure 3.7 Phone used in gesture analysis (Agrawal et al., 2011)

In Figure 3.8, depth camera is used as another method but it is an unfavorable point that it is quite heavy to carry, especially for children.

Figure 3.8 Kinect used in gesture analysis (Tutty, 2017)

Finally, in Figure 3.9, the acceleration sensor and the Arduino are mounted on a glove (Ecer, Yetgin, & Celik, 2018). Arduino also needs a battery or power. The transportation and usability of systems with components such as Arduino or depth camera prevents them from being preferred for daily usage due to their weight and size.

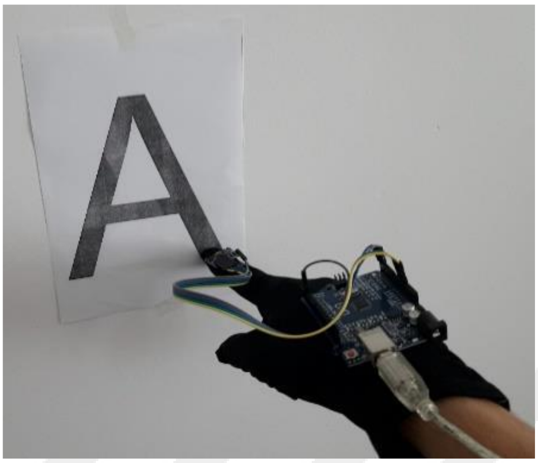

Figure 3.9 Arduino and glove used in gesture analysis (Ecer, Yetgin, & Celik, 2018)

3.2.1 Micro-Electro-Mechanical Systems

MEMS are a combination of the state of miniaturized electronic and mechanical particles. The idea came up when engineers decided it would be easier to use different mechanical and electronic systems on the same circuit for the projects. The structure of the MEMS sensor was first considered in the 1960s, but commercial production has not been started for a long time. Then in 1982, MEMS sensors were used to detect a collision in the airbag systems of automobiles. In 1991, Analog Devices Corporation developed an acceleration sensor for airbag systems using MEMS logic. A gyroscope has been added to the system to obtain more accurate map and orientation information in position systems in automobiles. We can imagine that the first MPU6050 appeared after these developments (MEMS, 2017).

MEMS is a very small device but it contains fixed standing microsensor, microactuator, microelectronic and microstructure structures in Figure 3.10. Microsensor and microactuator elements are important because they are energy conversion part and are called transducer (Bond, 2015).

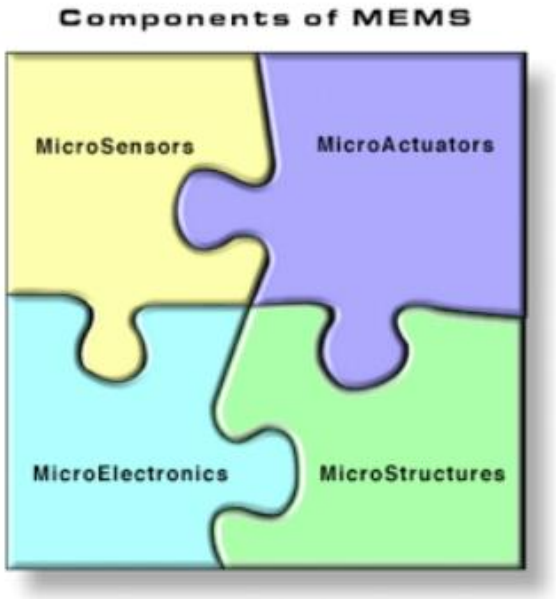

Figure 3.10 Components of MEMS (Bond, 2015)

The reasons why MEMS sensors are preferred is because they can be produced easily and are resistant to external factors such as vibration and radiation. Additionally, they are easy to transport and require very little power inside the system because they have small mass and volume. That’s why, MEMS sensor are used in many fields including space, automotive, health, commercial and military (Gümüş, 2015).

3.2.2.1 MPU6050 (BWT61CL)

As rapidly developing technology facilitates life more, it will be indispensable for people. The usefulness of devices is proportional to their portability. So, this study covers the use of Micro-Electro-Mechanical System (MEMS) sensors and communication via Bluetooth to transmit sensor data (Figure 3.11) in order to interpret the shape drawn into the air or on the floor.

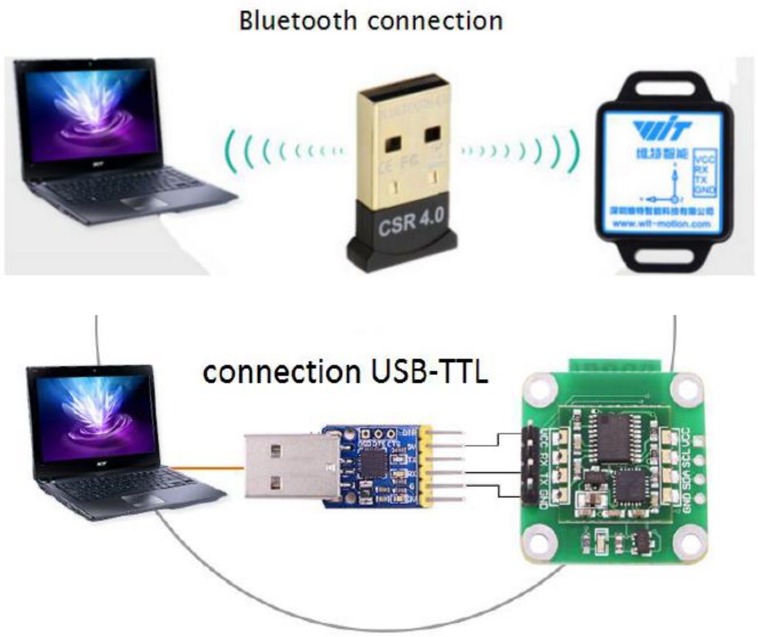

Figure 3.11 Computer and MPU6050 using Bluetooth and USB connection (Wit motion, n.d.)

MPU6050 is type of MEMS and even though it is a small device, it can be connected to a computer via USB or Bluetooth. In addition, the MPU6050 is a very handy sensor as it can be connected to the phone via Bluetooth. The MPU6050 is used as a MEMS sensor to capture dynamic hand movement and is preferred because it has a small integrated structure that transfers data from the gyroscope and accelerometer sensors inside to the computer via Bluetooth. MEMS sensors are especially preferred because of their small structure and they can be placed easily on the hand and arm as shown Figure 3.12.

Figure 3.12 MPU6050 sensor on wrist and hand (Personal archive, 2019)

Specification of BWT61CL is type of MPU6050;

- ✓ It supports serial port and Bluetooth,
- ✓ Bluetooth transmission distance of BWT61CL is 10 meters,
- ✓ It can used with Android,
- ✓ This sensor can accurately display the module's real-time motion output in a dynamic environment using the Kalman filter.
- ✓ The digital filter of this sensor can reduce the noise on the output and provide a more accurate measurement (Wit motion, n.d.).

Parameter	Function
VCC	Module power, 3.3V or 5V input
RX	Serial data input, TTL level
TX	Serial data output, TTL level
GND	Ground

Figure 3.13 Pin description of MPU6050 sensor (Wit motion, n.d.)

Product Parameters:

- ✓ Voltage is between 3.3 Volt and 5 Volt in Figure 3.13
- ✓ Current is smaller than 40 mA
- ✓ Output frequency is 100 Hz
- ✓ Measurement stability of attitude is 0.05 degrees
- ✓ Date interface is serial Transistor-Transistor Logic (TTL) level
- ✓ Stability of angular speed -0.05°/s and accelerated speed is -0.01 g
- ✓ Range of accelerated speed is ±16 g,
- ✓ Range of angular speed is ±2000°/s,
- ✓ Range of angle is ±180°
- ✓ Outputs of sensor are angular speed, accelerated speed, time, angel
- ✓ Baud rate is 115200 and it default value in Figure 3.14 (Wit motion, n.d.).

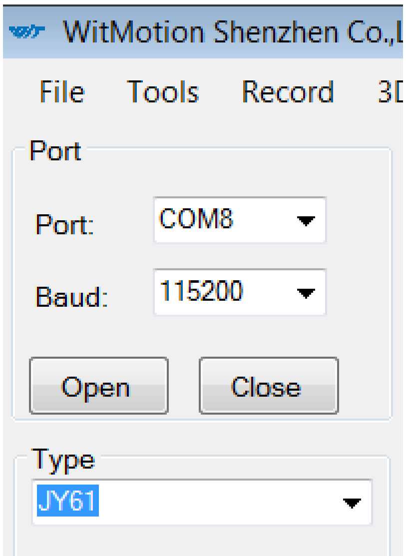

Figure 3.14 Baud rate and port of MPU6050

MPU6050 has;

- ✓ 3-axis Gyroscope
- ✓ 3-axis Accelerometer in Figure 3.15
- ✓ Bluetooth
- ✓ It has battery
- ✓ It is very small and lightweight

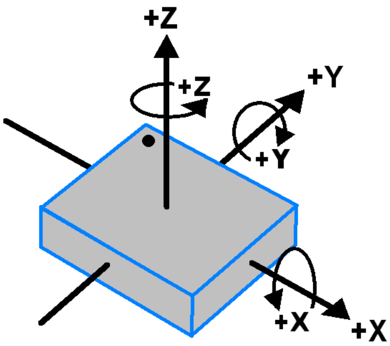

Figure 3.15 X, Y, Z axis of MPU6050 sensor (Wit motion, n.d.)

The MPU6050 sensor can transfer data from the sensor directly to the MATLAB using a Bluetooth or USB cable, or use its own interface for transferring this data to a computer. This interface, in Figure 3.16, shows the angle, acceleration, and angular velocity of the data from the sensor.

Figure 3.16 Interface of MPU6050

3.2.2 Working Principle of MPU6050

When the MPU6050 sensor is moved, it transmits the acceleration, angular velocity of the motion to the computer via Bluetooth in three different parts. These values are calculated by the following equations (Wit motion, n.d.).

There is g (gravity acceleration) =9.8 m/s^2 32768=2^{15} and 15 is bit number,

H=High and **L**=Low, **B**=Byte

a=acceleration (g)

w=angular velocity (deg/s)

Accelerated speed= ±16g

Angular speed=±2000°/s

Angle= ±180°

Table 3.1 Acceleration output

Data number	Data content	Implication
0	0x55	Header of packet
1	0x51	Acceleration pack
2	AxL	x-axis acceleration LB
3	AxH	x-axis acceleration HB
4	AyL	y-axis acceleration LB
5	AyH	y-axis acceleration HB
6	AzL	z-axis acceleration LB
7	AzH	z-axis acceleration HB
8	TL	Temperature LB
9	TH	Temperature HB
10	Sum	Checksum

Formula for calculating acceleration：

$$ax=((AxH<<8)|AxL)/32768*16g \tag{3.1}$$

$$ay=((AyH<<8)|AyL)/32768*16g \tag{3.2}$$

$$az=((AzH<<8)|AzL)/32768*16g \tag{3.3}$$

Table 3.2 Angular velocity output

Data number	Data content	Implication
0	0x55	Header of packet
1	0x51	Angular pack
2	wxL	x-axis angular LB
3	wxH	x-axis angular HB
4	wyL	y-axis angular LB
5	wyH	y-axis angular HB
6	wzL	z-axis angular LB
7	wzH	z-axis angular HB
8	TL	Temperature LB
9	TH	Temperature HB
10	Sum	Checksum

Formula for calculating angular velocity:

$$wx=((wxH<<8)|wxL)/32768*2000(°/s) \tag{3.4}$$

$$wy=((wyH<<8)|wyL)/32768*2000(°/s) \tag{3.5}$$

$$wz=((wzH<<8)|wzL)/32768*2000(°/s) \tag{3.6}$$

Using the above formulas, acceleration, angular velocity and angle values and angles in the Figure 3.17 are obtained.

172,173	0,0972	-0,3911	0,8696	-0,7324	-1,709	0
172,173	0,0967	-0,3901	0,8716	-0,7324	-1,6479	0,1831
172,173	0,0952	-0,3901	0,8721	-0,6104	-1,709	0
172,173	0,0938	-0,3896	0,8647	-0,5493	-1,8311	0,1831
172,173	0,0938	-0,3911	0,8643	-0,5493	-1,8311	0,1831
172,173	0,0962	-0,3926	0,8657	-0,5493	-1,77	0,1831
172,173	0,0957	-0,3896	0,8633	-0,5493	-1,5869	0
172,173	0,0947	-0,3911	0,8677	-0,5493	-1,5259	0,1831
172,173	0,0977	-0,3892	0,8682	-0,6104	-1,6479	0
172,173	0,0986	-0,3906	0,8682	-0,6104	-1,77	0
172,173	0,0967	-0,3901	0,8667	-0,6104	-1,77	0
172,198	0,0981	-0,3906	0,8696	-0,6714	-1,709	0
172,203	0,0967	-0,3911	0,8643	-0,6104	-1,709	0
172,203	0,0962	-0,3931	0,8657	-0,5493	-1,8311	0
172,208	0,0972	-0,3926	0,8687	-0,6104	-1,8311	0
172,309	0,0962	-0,3926	0,8643	-0,6104	-1,77	0,1831
172,314	0,0962	-0,3921	0,8657	-0,6714	-1,77	0,2441
172,314	0,0981	-0,3921	0,8618	-0,6104	-1,77	0,1831
172,314	0,0972	-0,3936	0,8623	-0,6714	-1,8311	0,2441
172,314	0,0967	-0,3921	0,8638	-0,7324	-1,9531	0,1831
172,314	0,0981	-0,3921	0,8623	-0,7324	-2,0142	0,2441
172,314	0,1001	-0,3936	0,8628	-0,7324	-1,9531	0,2441
172,314	0,0996	-0,3921	0,8638	-0,7324	-1,8921	0,1831
172,314	0,1011	-0,3926	0,8647	-0,7935	-1,8921	0,1831
172,325	0,1021	-0,3911	0,8633	-0,7324	-1,8921	0,1831
172,33	0,1011	-0,3931	0,8633	-0,6714	-1,8921	0,2441

Figure 3.17 Sample data (time, ax, ay, az and wx, wy, wz) from the sensor

CHAPTER FOUR
EXPERIMANTAL WORKS AND DISCUSSION

In this project, the MPU6050 was particularly preferred because it has a rechargeable battery and is relatively lightweight and easily portable compared to alternative devices. Data is obtained by moving the sensor in a flat surface or attaching the sensor to the wrist, then it is transferred via Bluetooth. Using the accelerations obtained from the transferred data, graphs which are specific to each gesture have emerged and gestures have been defined from these graphs by using the selected HMM algorithm method as shown in Figure 4.1. Using forward algorithm of HMM, the probabilities of the end-to-end states were found and the test data was classified according to the highest probability.

Hand movement’s data set were not only made up of letters, but were composed of three different data set: geometric shapes, letters and numbers. In this way, successful results were obtained.

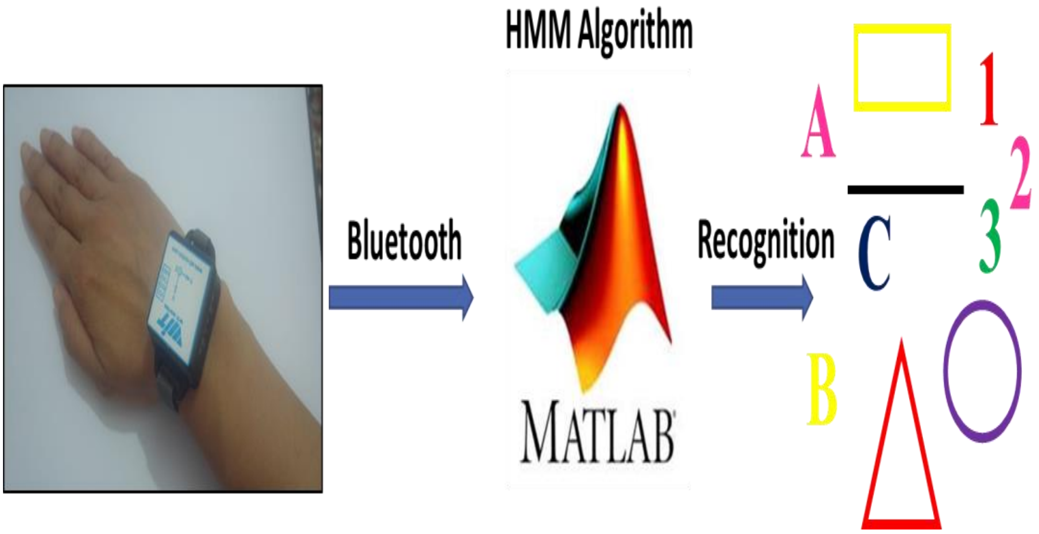

Figure 4.1 Working principle of project (Personal archive, 2019)

4.1 Flowchart of Project

In order to define hand gestures in this project, first of all, the movements must be classified. Therefore, this study is aimed to train the system using HMM algorithm before classifying a gesture. To classify and train the obtained gestures from the sensor, the following ways should be followed:

- ✓ Different hand gestures obtained by the sensor are transferred to the computer via Bluetooth and a data set is created.
- ✓ Some of the data obtained is recorded in the training set to classify hand gestures, and the rest is recorded in the test set.
- ✓ Parameters of hidden Markov model are determined.
- ✓ To minimize the differences between data from different individuals, the data is normalized and classified with K-Means clustering method.
- ✓ Forward algorithm is used to train HMM parameters using the recorded data set.
- ✓ The basic scheme of the classification of test data that is not in the education system is as shown in Figure 4.2 (Gillian, 2019).

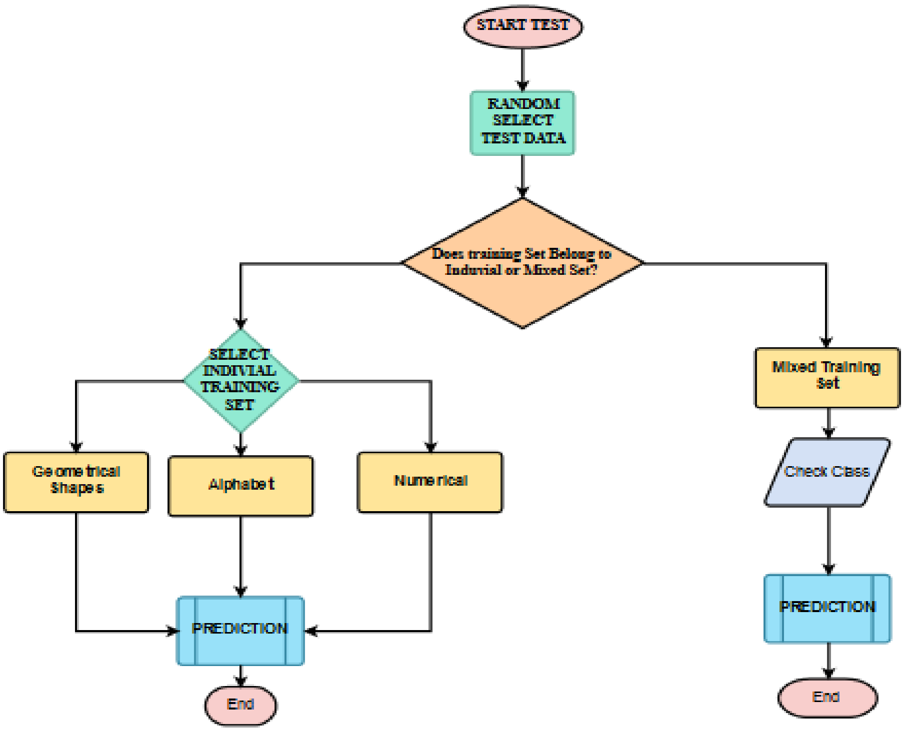

Figure 4.2 Flowchart of project

4.2 Data Preprocessing and Clustering Process

In this project, three data sets that are geometrical shapes, capital letters and numbers are used. In order to use this data collected from individuals in the classification process, the procedures described in the next section should be performed.

4.2.1 Data Acquisition and Feature Analysis

Firstly, a point reference for each shape or character is selected, then the z angle value of the data from the MPU6050 sensor is tried to reduce to be zero as shown in Figure 4.3 and then gesture is drawn in air or on the flat ground. This process is repeated for all data. Data has time, accelerometer and angular velocity and angle values. That’s why, ax, ay, az and wx, wy, wz values of the obtained data are enough for gesture recognition process as in Table 4.1. Therefore, other columns are deleted, all commas are converted to dots, and the files are saved as text files with ASCII encoding for use in MATLAB. An example of the first and final state of a gesture data is shown in Table 4.1 and Table 4.2. In this way, feature extraction is made. Also, during data preparation, since the data is received from more than one person, the initial of the person's name is added to the filename to determine who the data belongs to.

Figure 4.3 Application of MPU6050 sensor

Table 4.1 The first version of circle data from sensor

StartTime: 2019-09-01 02:27:36.362										
Time	**ax**	**ay**	**az**	**wx**	**wy**	**wz**	**AngleX**	**AngleY**	**AngleZ**	**T(°)**
221,943	-0,647	-0,0767	-0,8306	-0,1831	0	0,2441	-174,584	37,9303	0	36,1976
221,943	-0,6479	-0,0771	-0,8311	0	0	0,2441	-174,589	37,9303	0	36,177
221,943	-0,6475	-0,0776	-0,8291	0	0	0	-174,589	37,9303	0	36,1741
221,953	-0,6484	-0,0771	-0,8281	0	0	0	-174,589	37,9303	0	36,1947
221,985	-0,6475	-0,0767	-0,8271	0	0	0	-174,589	37,9303	0	36,2065
221,99	-0,647	-0,0771	-0,8271	0	0	0	-174,589	37,9303	0	36,1947
221,99	-0,647	-0,0791	-0,8276	0	0	0	-174,589	37,9303	0	36,2006
221,995	-0,6465	-0,0771	-0,8301	0	0	0	-174,595	37,9303	0	36,2065
222,001	-0,6489	-0,0757	-0,8296	0	0	0	-174,595	37,9303	0	36,2123
222,072	-0,6484	-0,0801	-0,8271	0	0	0	-174,595	37,9303	0	36,2006
222,077	-0,6479	-0,0796	-0,8276	0	0	0	-174,589	37,9303	0	36,2006
222,077	-0,6455	-0,0801	-0,8291	0	0	0	-174,589	37,9303	0	36,1859
222,077	-0,6465	-0,0796	-0,8271	0	0	0	-174,589	37,9303	0	36,1888
222,077	-0,6445	-0,0801	-0,8262	0	0	0	-174,589	37,9303	0	36,1888
222,082	-0,6455	-0,0776	-0,8291	0	0	0	-174,589	37,9248	0	36,2006
222,087	-0,647	-0,0786	-0,8252	0	0	0,1831	-174,589	37,9248	0	36,1918
222,092	-0,647	-0,0791	-0,8301	0	0	0,1831	-174,589	37,9248	0	36,1829
222,092	-0,647	-0,0796	-0,8281	0	0	0,2441	-174,595	37,9248	0	36,1947
222,123	-0,6479	-0,0786	-0,8286	0	0	0,2441	-174,595	37,9248	0	36,2065
222,128	-0,6479	-0,0771	-0,8276	0	0	0,2441	-174,595	37,9248	0	36,1976
222,133	-0,6465	-0,0796	-0,8271	0	0	0,2441	-174,6	37,9248	0	36,1947
222,133	-0,6465	-0,0791	-0,8271	0	0	0,1831	-174,6	37,9248	0	36,1829
222,165	-0,6455	-0,0791	-0,8262	0	0	0	-174,6	37,9248	0	36,1829
222,17	-0,6475	-0,0791	-0,8232	0	0	0	-174,595	37,9248	0	36,1976
222,175	-0,6479	-0,0781	-0,8252	0	0	0	-174,595	37,9303	0	36,1918
222,175	-0,6489	-0,0791	-0,8232	0	0	0	-174,595	37,9303	0	36,2035
222,21	-0,6465	-0,0776	-0,8257	-0,1831	0	0	-174,6	37,9303	0	36,2006
222,21	-0,6455	-0,0767	-0,8237	-0,1831	0	0	-174,6	37,9303	0	36,2006
222,215	-0,6475	-0,0781	-0,8271	-0,1831	0	0	-174,6	37,9303	0	36,2035
222,215	-0,6475	-0,0771	-0,8281	0	0	0	-174,606	37,9303	0	36,2094
222,273	-0,647	-0,0762	-0,8252	0	0	0	-174,606	37,9303	0	36,1976
222,273	-0,6484	-0,0762	-0,8262	0	0	0	-174,606	37,9303	0	36,1888
222,278	-0,6484	-0,0781	-0,8286	0	0	0	-174,606	37,9303	0	36,1859
222,278	-0,6489	-0,0762	-0,8301	0	0	0	-174,606	37,9303	0	36,1918
222,278	-0,6475	-0,0762	-0,8286	0	0	0	-174,611	37,9303	0	36,1976
222,296	-0,6479	-0,0781	-0,8276	0	0	0	-174,611	37,9303	0	36,2035
222,301	-0,6475	-0,0771	-0,8267	0	0	0	-174,611	37,9303	0	36,1741
222,306	-0,6499	-0,0771	-0,8257	0	0	0	-174,611	37,9303	0	36,1741
222,306	-0,646	-0,0767	-0,8271	0	0	0	-174,611	37,9303	0	36,1888

Table 4.2 Final version of circle data (time, ax, ay, az, wx, wy, wz)

221.943	-0.6470	-0.0767	-0.8306	-0.1831	0.0000	0.2441
221.943	-0.6479	-0.0771	-0.8311	0.0000	0.0000	0.2441
221.943	-0.6475	-0.0776	-0.8291	0.0000	0.0000	0.0000
221.953	-0.6484	-0.0771	-0.8281	0.0000	0.0000	0.0000
221.985	-0.6475	-0.0767	-0.8271	0.0000	0.0000	0.0000
221.990	-0.6470	-0.0771	-0.8271	0.0000	0.0000	0.0000
221.990	-0.6470	-0.0791	-0.8276	0.0000	0.0000	0.0000
221.995	-0.6465	-0.0771	-0.8301	0.0000	0.0000	0.0000
222.001	-0.6489	-0.0757	-0.8296	0.0000	0.0000	0.0000
222.072	-0.6484	-0.0801	-0.8271	0.0000	0.0000	0.0000
222.077	-0.6479	-0.0796	-0.8276	0.0000	0.0000	0.0000
222.077	-0.6455	-0.0801	-0.8291	0.0000	0.0000	0.0000
222.077	-0.6465	-0.0796	-0.8271	0.0000	0.0000	0.0000
222.077	-0.6445	-0.0801	-0.8262	0.0000	0.0000	0.0000
222.082	-0.6455	-0.0776	-0.8291	0.0000	0.0000	0.0000
222.087	-0.6470	-0.0786	-0.8252	0.0000	0.0000	0.1831
222.092	-0.6470	-0.0791	-0.8301	0.0000	0.0000	0.1831
222.092	-0.6470	-0.0796	-0.8281	0.0000	0.0000	0.2441
222.123	-0.6479	-0.0786	-0.8286	0.0000	0.0000	0.2441
222.128	-0.6479	-0.0771	-0.8276	0.0000	0.0000	0.2441
222.133	-0.6465	-0.0796	-0.8271	0.0000	0.0000	0.2441
222.133	-0.6465	-0.0791	-0.8271	0.0000	0.0000	0.1831
222.165	-0.6455	-0.0791	-0.8262	0.0000	0.0000	0.0000
222.170	-0.6475	-0.0791	-0.8232	0.0000	0.0000	0.0000
222.175	-0.6479	-0.0781	-0.8252	0.0000	0.0000	0.0000
222.175	-0.6489	-0.0791	-0.8232	0.0000	0.0000	0.0000
222.210	-0.6465	-0.0776	-0.8257	-0.1831	0.0000	0.0000
222.210	-0.6455	-0.0767	-0.8237	-0.1831	0.0000	0.0000
222.215	-0.6475	-0.0781	-0.8271	-0.1831	0.0000	0.0000
222.215	-0.6475	-0.0771	-0.8281	0.0000	0.0000	0.0000
222.273	-0.6470	-0.0762	-0.8252	0.0000	0.0000	0.0000
222.273	-0.6484	-0.0762	-0.8262	0.0000	0.0000	0.0000
222.278	-0.6484	-0.0781	-0.8286	0.0000	0.0000	0.0000
222.278	-0.6489	-0.0762	-0.8301	0.0000	0.0000	0.0000
222.278	-0.6475	-0.0762	-0.8286	0.0000	0.0000	0.0000
222.296	-0.6479	-0.0781	-0.8276	0.0000	0.0000	0.0000
222.301	-0.6475	-0.0771	-0.8267	0.0000	0.0000	0.0000
222.306	-0.6499	-0.0771	-0.8257	0.0000	0.0000	0.0000
222.306	-0.6460	-0.0767	-0.8271	0.0000	0.0000	0.0000
222.311	-0.6460	-0.0762	-0.8306	0.0000	0.0000	0.0000
222.341	-0.6475	-0.0771	-0.8291	0.0000	-0.1831	0.0000
222.346	-0.6475	-0.0776	-0.8262	0.0000	-0.1831	0.0000
222.346	-0.6465	-0.0781	-0.8281	0.0000	0.0000	0.0000

4.2.2 Normalization of Data

As mentioned in the previous section, the hand gestures were obtained from more than one person. When different people are typing in the air, the height or movement speed of the hand from the ground may be different. In addition, environmental factors such as hand tremor can make a difference between the hand movements obtained. In this project, differences between the data were observed and normalization process was performed for mixed training and test sets. Considering the largest and smallest values in each column of data, all columns are processed and normalize so that the new values of the data are between 0 and 1 with the following formula. X is a data;

$$X_{new}=(X-\min(X))\,/(\max(X)-\min(X)) \tag{4.1}$$

While normalization process plays an active role in the classification of the data obtained from different individuals, it was observed that the success of the classification did not change much if the training set and test set belonged to the same person. In the following sections, the samples will be examined in detail.

4.2.3 K-Means Clustering

After the preprocessing of the data obtained, we perform the clustering process using the K-means algorithm in MATLAB with the number of observations, an important parameter of HMM, in training system. The number of observations, a parameter of HMM, was used as the number of clustering in the K-means algorithm. Example clustering for each different class of a data set is shown in Figure 4.4. It was used 240 as the number of clusters for this example. This means that the training data set is divided into 240 clusters. The distribution, in 240 clusters, of each of the 17 classes in the training set was visualized. After, by using the values obtained from the K-means algorithm, each data in the training set is processed and the observation series are formed.

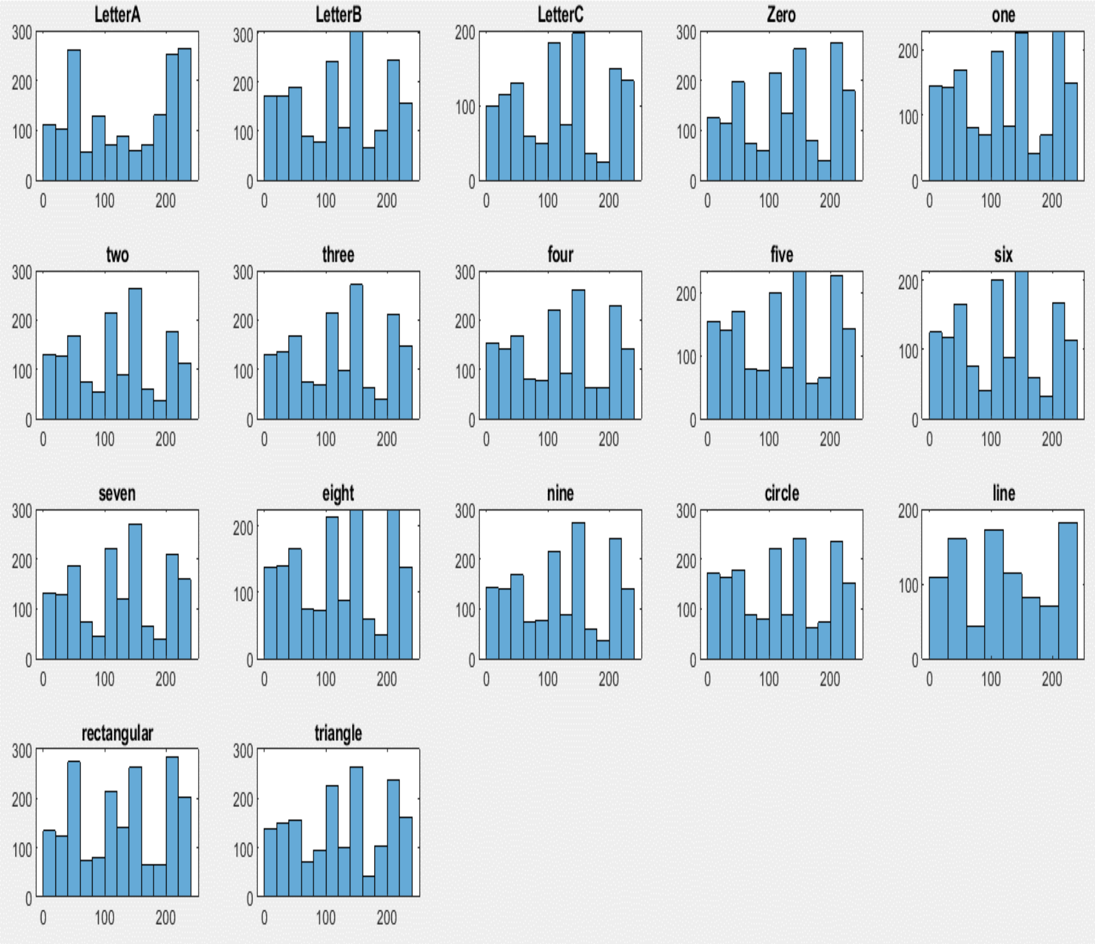

Figure 4.4 Clustering for each different class

4.3 System Training with HMM Algorithm

At this stage, an HMM model is created for each training data and model parameters is calculated using the observation series obtained after the clustering process as well as the initially defined numbers of state and observations. Forward Algorithm is used for modeling, and A, B, alpha, beta parameters have been calculated with this algorithm. For each data in the training set, probabilities value increased over time and after 30 iterations, $P(O|\lambda)$ value was observed to be fixed as in Figure 4.5. In this way, the training process of the system for classification is completed.

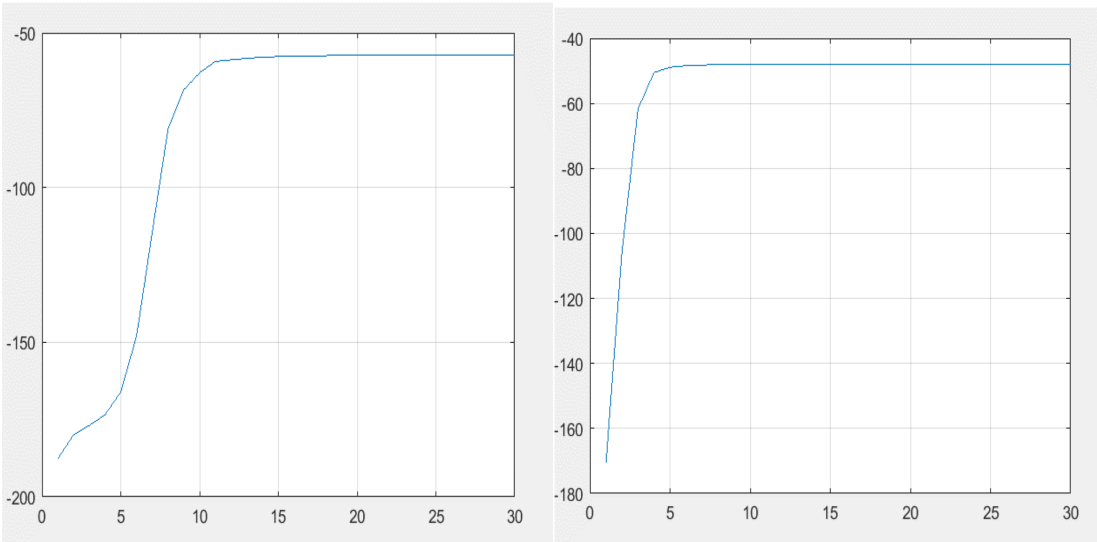

Figure 4.5 P(O|λ) for iteration 30

To express the training of the system with an example, the same movement is repeated a certain number of times for each of the 17 classes in the mixed training set. The training of the system is showed in Figure 4.6. The letter i is used as a symbol to indicate which class in the system is used. This refers to the order of classes from 1 to 17. In the explanations, Ns and No abbreviations are used respectively to express the number of state and the number of observations. How these parameters are selected is described in the next section.

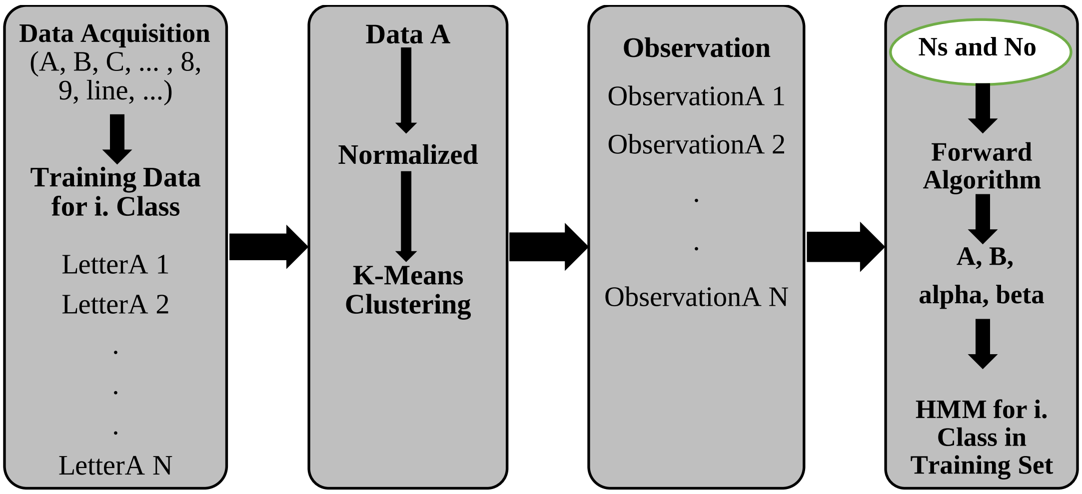

Figure 4.6 Block diagram for HMM of system

4.4 Classification of Test Data

Firstly, unused data in the training set is selected for the test set. Then, the center points and HMM parameters in the training system are loaded to test system. The HMM parameters and also value obtained by the KNN classification algorithm are used in the Forward algorithm to estimate which class the test data belongs to. This process is shown in Figure 4.7.

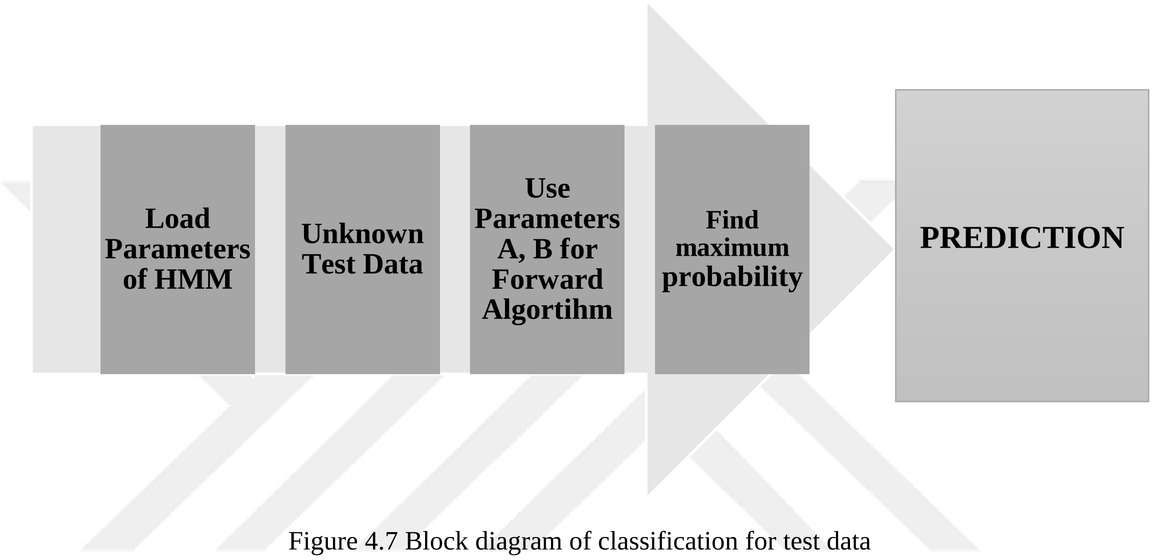

Figure 4.7 Block diagram of classification for test data

4.5 Results of Test

After the preprocessing in Section 4.2, the training sets were created by taking equal number of samples from each class. The test sets were selected from the data not in the training set.

Firstly, 85 data are obtained from first person for preliminary study. Geometrical shapes set consists circle, line, rectangular and triangle. There are selected 4 sample for each class and training set is formed with 16 data. The test set of geometrical shapes contains a total of randomly selected 4 data. Alphabet set is formed A, B and C letters. The training set and test set of this set consist of 12 and 3 data, respectively. Finally, set of numbers consists of 0, 1, 2, 3, 4, 5, 6, 7, 8, 9 numbers. Training set for numbers has 40 data and test set has 10 data. All of them can be seen in Table 4.3 and Table 4.4.

Secondly, in this part, both the mixed training and the mixed test set were created by the same person and a 73.5% success rate was achieved by using only the test set of this person. There is a total of 170 data from second person, of which 40% is used for the training set and 60% is used for the test set, as shown in section 4.3.4.

Finally, more data is saved to create a mixed set from the first, second and third person. The shapes, letters and numbers are combined to form a new training set, then the system is trained. This mixed training set was formed by taking 6 samples from three people for each of the 17 classes. Number of data in training set is 106. Using this mixed training system, 102 data of the only second person were tested and maximum 68.6% success was achieved. After, when the mixed training set remained the same, a mixed test set was created with data from three people. The rate of success was observed to be around 50%.

In Appendix 1 MATLAB GUI results were given for some test data.

Table 4.3 List of three separate training data from first person

Numbers	Geometrical Shapes	Alphabet
Number0-1y.txt	ShapeCircle3y.txt	LetterA1y.txt
Number0-2y.txt	ShapeCircle4y.txt	LetterA2y.txt
Number0-3y.txt	ShapeCircle5y.txt	LetterA3y.txt
Number0-4y.txt	ShapeCircle6y.txt	LetterA4y.txt
Number1-1y.txt	ShapeLine1y.txt	LetterB1y.txt
Number1-2y.txt	ShapeLine2y.txt	LetterB2y.txt
Number1-3y.txt	ShapeLine3y.txt	LetterB3y.txt
Number1-4y.txt	ShapeLine6y.txt	LetterB6y.txt
Number2-1y.txt	ShapeRectangular1y.txt	LetterC3y.txt
Number2-2y.txt	ShapeRectangular2y.txt	LetterC4y.txt
Number2-5y.txt	ShapeRectangular3y.txt	LetterC5y.txt
Number2-6y.txt	ShapeRectangular4y.txt	LetterC6y.txt
Number3-1y.txt	ShapeTriangle1y.txt	
Number3-2y.txt	ShapeTriangle2y.txt	

Table 4.4 continues

Number3-3y.txt	ShapeTriangle5y.txt	
Number3-4y.txt	ShapeTriangle6y.txt	
Number4-3y.txt		
Number4-4y.txt		
Number4-5y.txt		
Number4-6y.txt		
Number5-1y.txt		
Number5-4y.txt		
Number5-5y.txt		
Number5-6y.txt		
Number6-1y.txt		
Number6-2y.txt		
Number6-3y.txt		
Number6-4y.txt		
Number7-1y.txt		
Number7-2y.txt		
Number7-3y.txt		
Number7-4y.txt		
Number8-1y.txt		
Number8-2y.txt		
Number8-3y.txt		
Number8-6y.txt		
Number9-1y.txt		
Number9-2y.txt		
Number9-3y.txt		
Number9-4y.txt		

Table 4.5 List of three separate test data from first person

Numbers	Geometrical Shapes	Alphabet
Number0-5y.txt	ShapeCircle2y.txt	LetterA5y.txt
Number1-6y.txt	ShapeLine4y.txt	LetterB4y.txt
Number2-3y.txt	ShapeRectangular5y.txt	LetterC1y.txt
Number3-5y.txt	ShapeTriangle4y.txt	
Number4-2y.txt		
Number5-2y.txt		
Number6-5y.txt		
Number7-6y.txt		
Number8-4y.txt		
Number9-6y.txt		

The mixed system was trained and then induvial systems were trained by HMM algorithm and generally it was observed that the classification could be fixed after 30 iterations. In order to obtain the estimation results, the number of hidden states (Ns) was selected between 2 and 20 and the observed number (No) was selected between 100 and 300. The results were obtained using the cross-validation method and the best result are selected.

For preliminary work, Ns=5 and No=200 were used, then algorithm finds correct predictions for all test set of geometrical shapes. Ns=10 and No=220 values were used for alphabet and it was observed that the highest success rate did not change even if Ns or No were changed for the numbers, so Ns=5 and No=220 were used.

In this part, Table 4.5 shows the variation of the success rates of the data tested according to the Ns and No numbers using the mixed training set obtained from the second person. The highest success for 102 test data appears to be achieved when the number of states is 14 or 18. So, state number can be selected one of them. Graphs in section 4.3.4 are shown for 14.

The abbreviation OSP in Table 4.5 and Table 4.6 means only the second person.

Table 4.5 Prediction rate for test and training set of only second person

Ns	No	True Prediction Number	Total Test Number	Success	Training Set	Test Set
20	210	72	102	70.6	OSP	OSP
18	190	**75**	102	**73.5**	OSP	OSP
16	220	72	102	70.6	OSP	OSP
14	200	**75**	102	**73.5**	OSP	OSP
12	240	72	102	70.6	OSP	OSP
10	240	73	102	71.6	OSP	OSP
5	240	70	102	68.6	OSP	OSP
4	210	72	102	70.6	OSP	OSP
2	210	64	102	62.8	OSP	OSP

Finally, Table 4.6 shows the variation of the success rates of the data tested according to the Ns and No numbers using the mixed training set obtained from the three people. The test consisted of the data obtained from the second person first and then the most successful Ns and No numbers were determined. This time, using these Ns and No values, success was measured with obtained test set from three person, but the success rate decreased as shown last two line in Table 4.6. Graphs in section 4.34 are shown for 12.

Table 4.6 Prediction rate for mixed training set from three person

Ns	No	True Prediction Number	Total Test Number	Success	Training Set	Test Set
20	210	59	102	57.8	Mixed	OSP
18	190	53	102	52	Mixed	OSP
16	220	54	102	53	Mixed	OSP
14	200	59	102	57.9	Mixed	OSP
12	240	**70**	102	**68.6**	Mixed	OSP
10	240	**70**	102	**68.6**	Mixed	OSP
5	240	56	102	55	Mixed	OSP
4	210	40	102	39.2	Mixed	OSP
2	210	45	102	44.1	Mixed	OSP
10	240	50	106	47.2	Mixed	Mixed
12	240	46	106	43.4	Mixed	Mixed

4.5.1 Only Geometric Shapes Test for Preliminary Study

Predictions of geometrical shape test set for Ns=5 and No=200 are shown from Figure 4.8 to Figure 4.11.

Data-1: 'True Data' 'ShapeCircle2y.txt' ' Predicted Data' 'Circle'

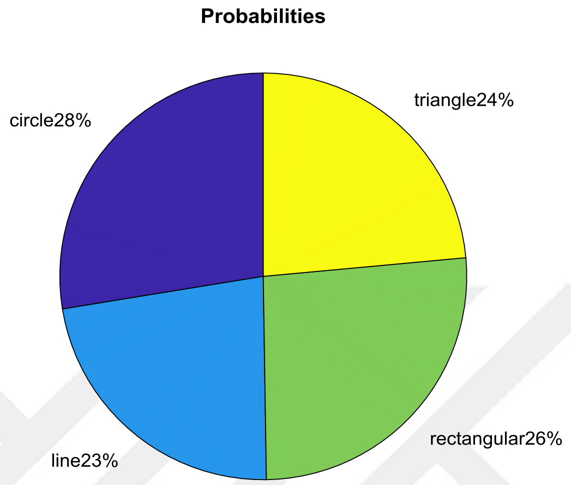

Figure 4.8 Circle probability for Ns=5 and No=200

Data-2: 'True Data' 'ShapeLine4y.txt' ' Predicted Data' 'Line'

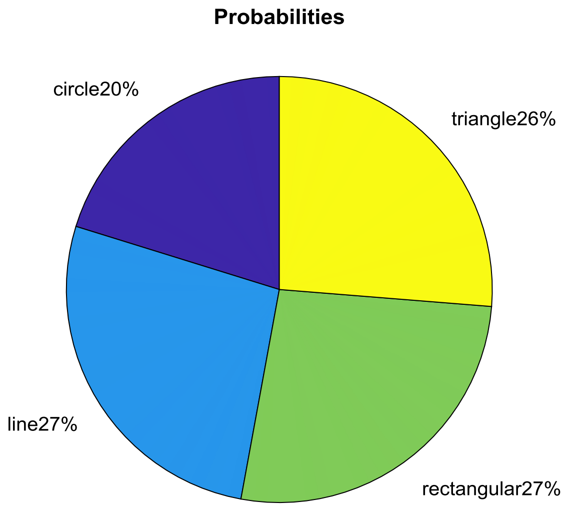

Figure 4.9 Line probability for Ns=5 and No=200

As seen in the Figure 4.9, the line movement is estimated correctly for No. Although the probability of being a line and a rectangle seems to be the same, the gesture is line as seen from actual probability of them. Actual values are “0.2021 0.2690 0.2662 0.2627” for circle, line, rectangular and triangle.

Data-3: 'True Data' 'ShapeRectangular5y.txt' ' Predicted Data' 'Rectangular'

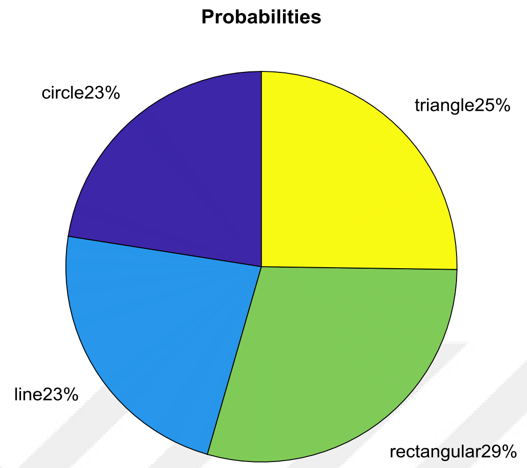

Figure 4.10 Rectangular probability for Ns=5 and No=200

Data-4: 'True Data' 'ShapeTriangle4y.txt' ' Predicted Data' 'Triangle'

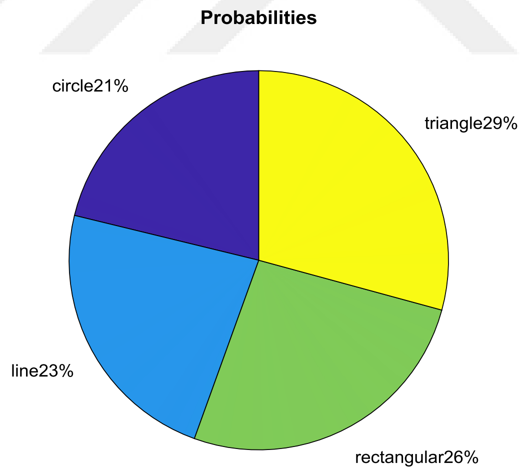

Figure 4.11 Triangle probability for Ns=5 and No=200

When Ns is constant but No value is changed from 200 to 220, if we compare the accuracy of the test data for both cases, it is seen that the accuracy of Data-1 in the second case decreases by 1% and Data-3 and Data-4 increase 1%. It is observed that the correct estimation is made for Data-2. So, we take No value as 200, the test success is 100%, but if No is 220, the success rate is about 67%.

4.5.2 Only Alphabet Test for Preliminary Study

Ns=5 and No=220 are selected for the alphabet training set, there is one wrong prediction. As seen as actual values of probability, B is higher than C letter. Their probabilities respectively are 0.2809, 0.3607, 0.3584.

Predictions of alphabet shape test set for Ns=5 and No=220 are shown from Figure 4.12 to Figure 4.14.

Data-1: 'True Data' 'LetterA5y.txt' ' Predicted Data' 'A'

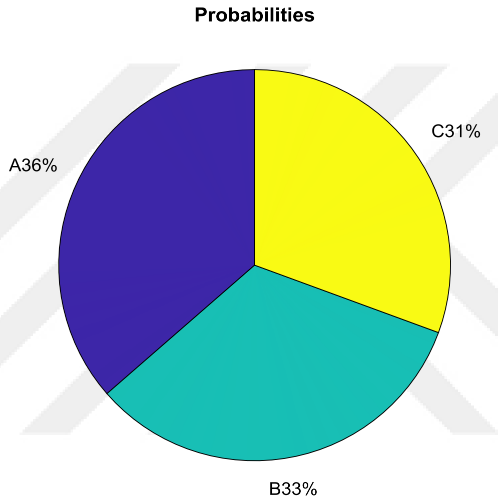

Figure 4.12 A for probability Ns=5 and No=220

Data-2: 'True Data' 'LetterB4y.txt' ' Predicted Data' 'B'

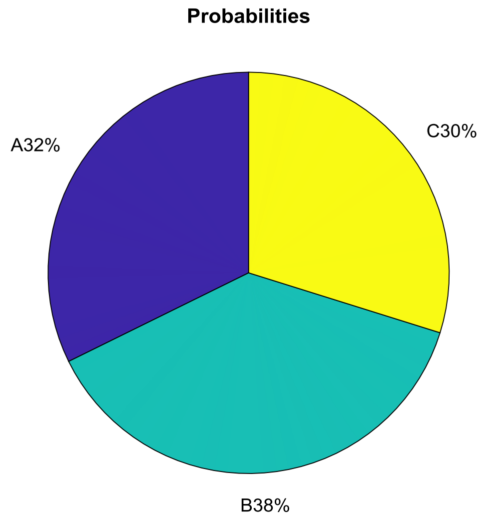

Figure 4.13 B for probability Ns=5 and No=220

Data-3: 'True Data' 'LetterC1y.txt' ' Predicted Data' 'B'

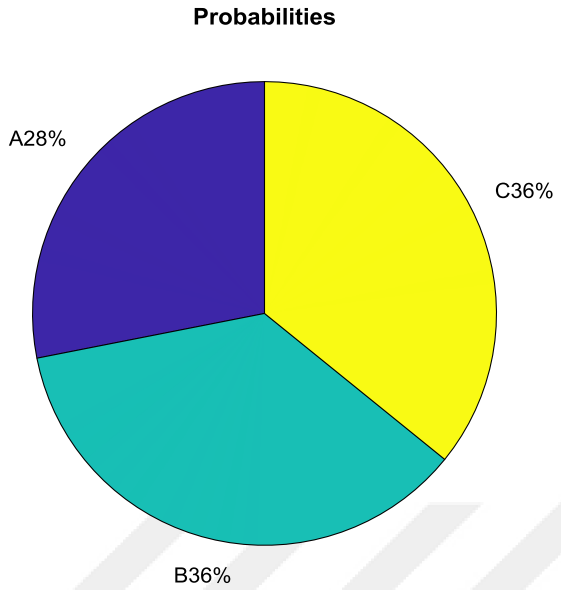

Figure 4.14 C for probability Ns=5 and No=220

Predictions of alphabet test set for Ns=10 and No=220 are shown from Figure 4.15 to Figure 4.17.

Data-1: 'True Data' 'LetterA5y.txt' ' Predicted Data' 'A'

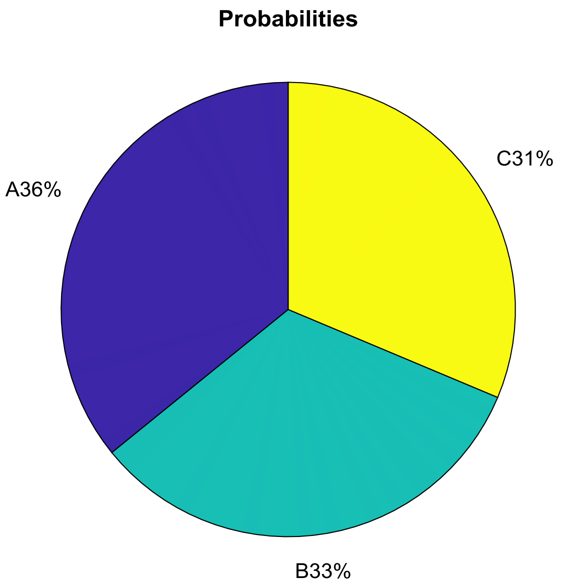

Figure 4.15 A for probability Ns=10 and No=220

Data-2: 'True Data' 'LetterB4y.txt' ' Predicted Data' 'B'

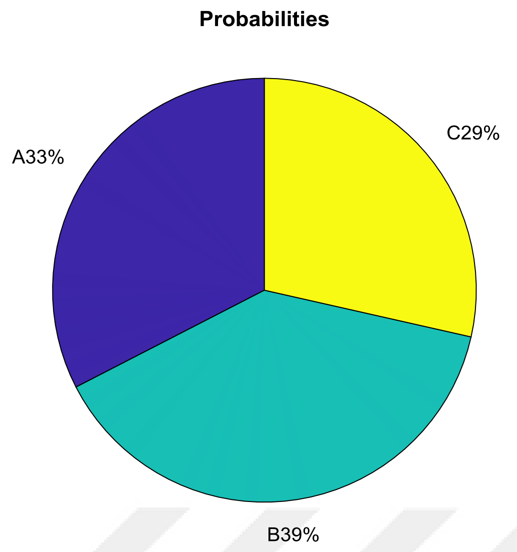

Figure 4.16 B for probability Ns=10 and No=220

Data-3: 'True Data' 'LetterC1y.txt' ' Predicted Data' 'C'

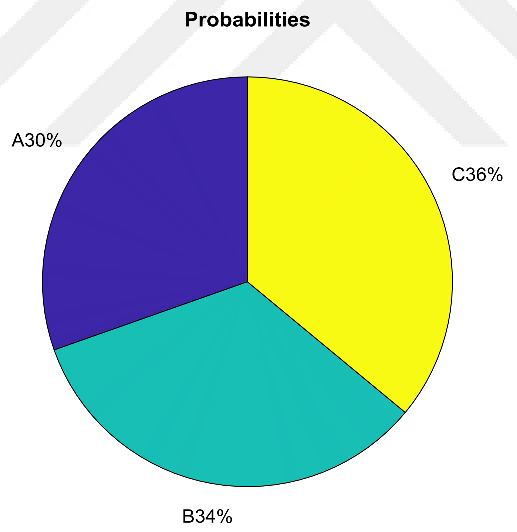

Figure 4.17 C for probability Ns=10 and No=220

When No is constant but Ns value is changed from 5 to 10, if we compare the accuracy of the test data for both cases, it is seen that the accuracy of Data-1 in the second case stay same value and Data-2 increases 1%. It is observed that the correct estimation is made for Data-3. So, if we take Ns value as 10, the test success is 100%, but if Ns is 5, the success rate is about 67%.

4.5.3 Only Numerical Test for Preliminary Study

Firstly, Ns=5 and No=220 were selected for the numbers training set, and there were seen two wrong predictions. To correct these wrong predictions, different values were tried and it was observed that the best result had two wrong predictions as previous. After, when Ns=5 and No=200 were selected, it was seen that only the wrong type changed. Therefore, test data of numerical set for Ns=5 and No=220 are shown in below figures. Eventually, if we take Ns value as 5 and No is 220, the test success is 80%, or if Ns is constant and No is 200, the success rate is 80% still.

Predictions of numerical test set for Ns=5 and No=220 are shown from Figure 4.18 to Figure 4.27.

Data-1: 'True Data' 'Number0-5y.txt' ' Predicted Data' 'four'

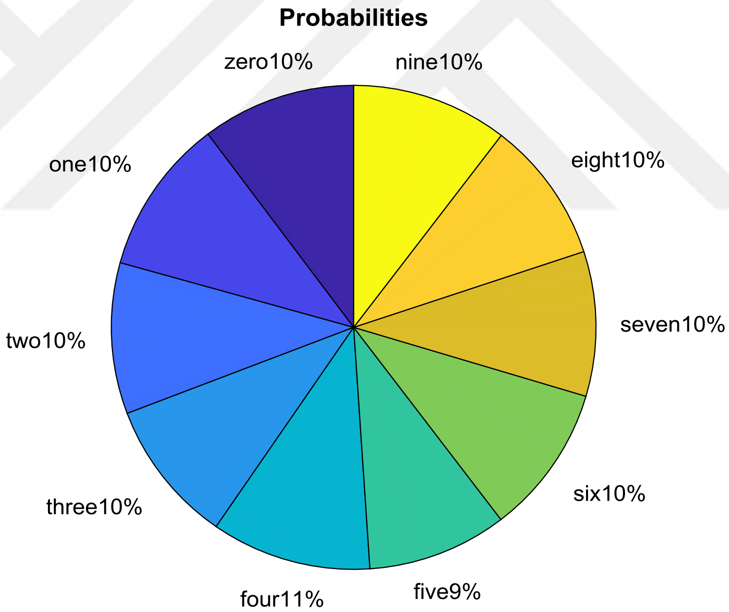

Figure 4.18 Zero for probability Ns=5 and No=220

Data-2: 'True Data' 'Number1-6y.txt' ' Predicted Data' 'one'

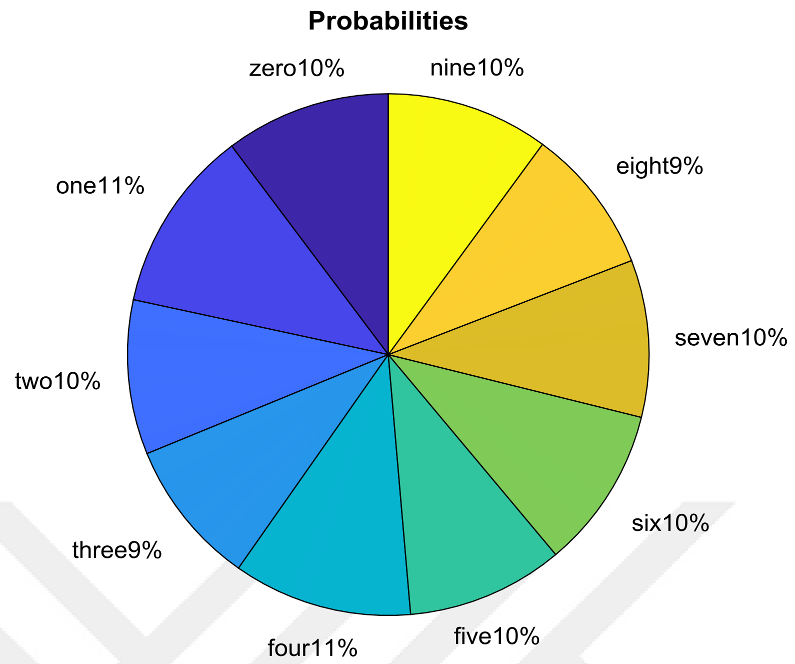

Figure 4.19 One for probability Ns=5 and No=220

Data-3: 'True Data' 'Number2-3y.txt' ' Predicted Data' 'two'

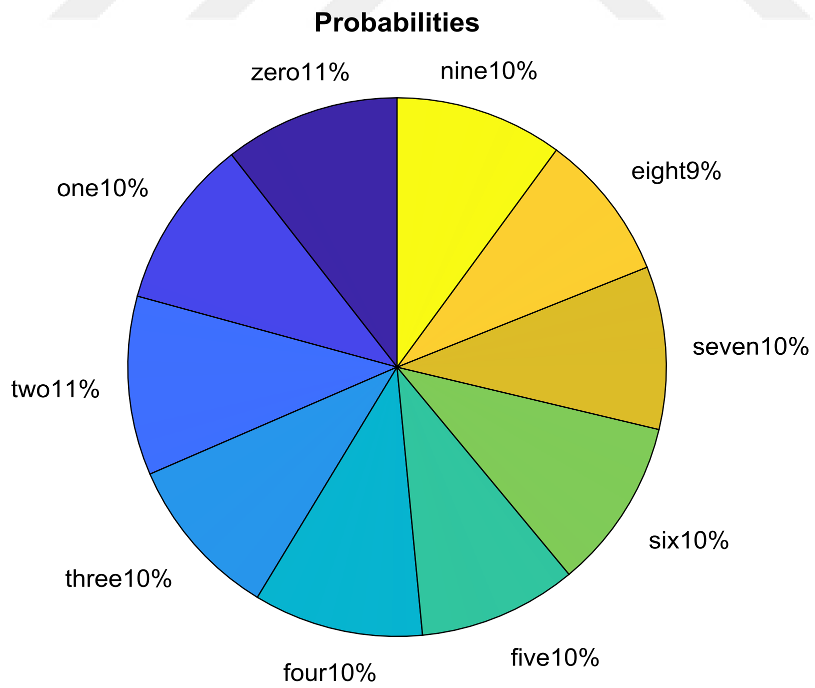

Figure 4.10 Two for probability Ns=5 and No=220

Data-4: 'True Data' 'Number3-5y.txt' ' Predicted Data' 'three'

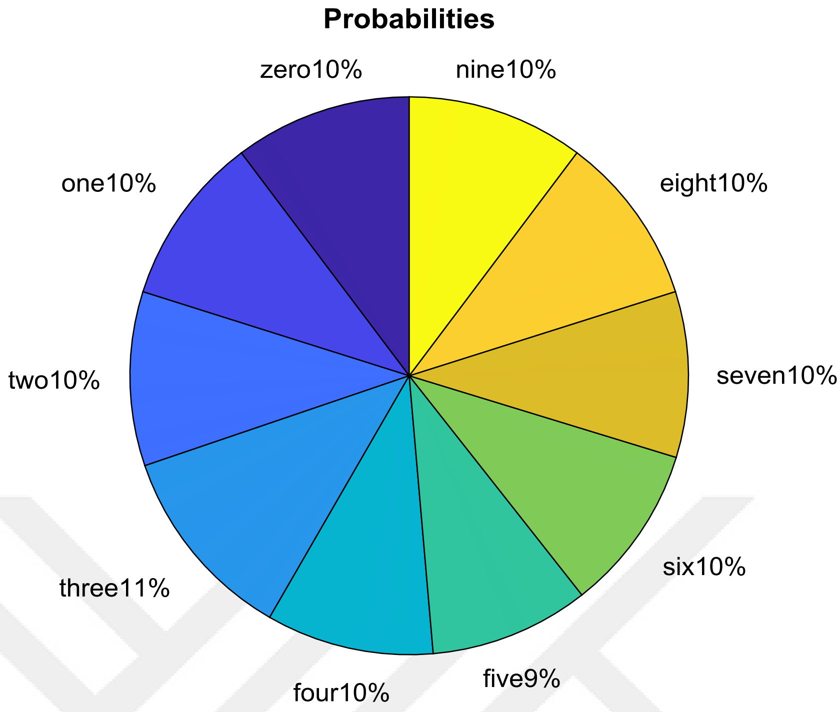

Figure 4.21 Three for probability Ns=5 and No=220

Data-5: 'True Data' 'Number4-2y.txt' ' Predicted Data' 'four'

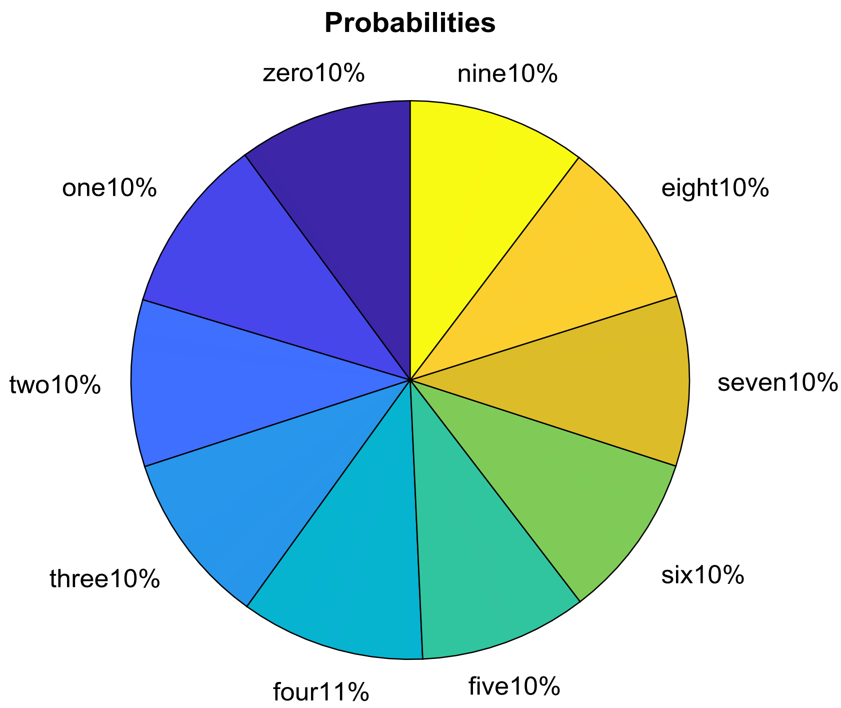

Figure 4.22 Four for probability Ns=5 and No=220

Data-6: 'True Data'   'Number5-2y.txt'   ' Predicted Data'   'five'

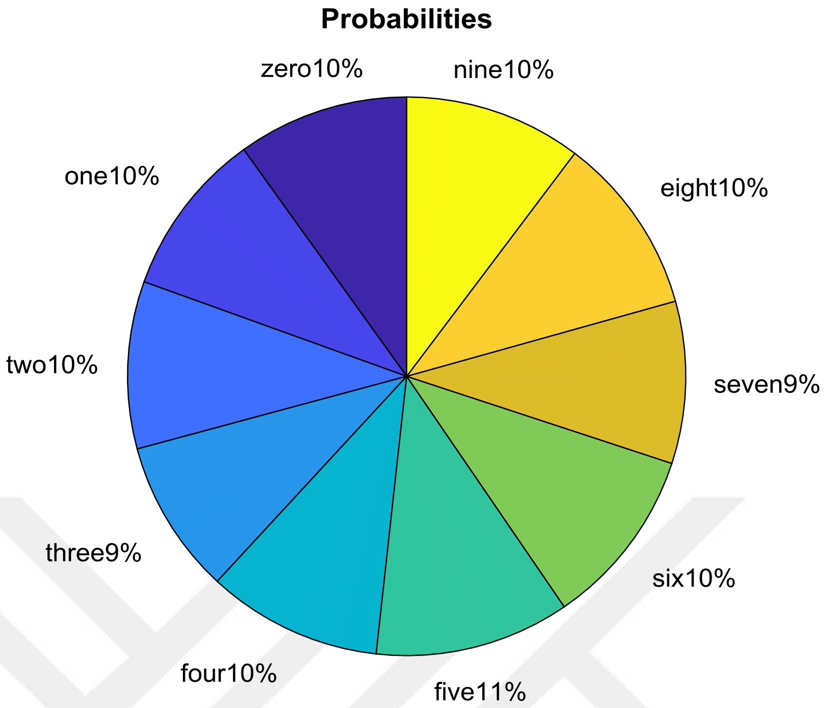

Figure 4.23 Five for probability Ns=5 and No=220

Data-7: 'True Data'   'Number6-5y.txt'   ' Predicted Data'   'zero'

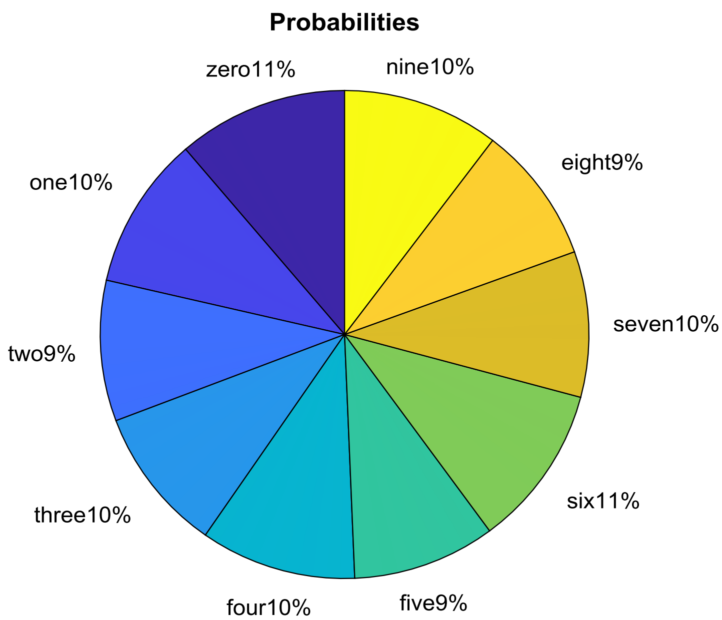

Figure 4.24 Six for probability Ns=5 and No=220

In fact, the probability values are as follows 0.1128, 0.1015, 0.0932, 0.0961, 0.1031, 0.0945, 0.1074, 0.0965, 0.0913, 0.1036 before they are rounded. So even though the possibilities zero and six seem equal, they are not actually equal.

Data-8: 'True Data' 'Number7-6y.txt' ' Predicted Data' 'seven'

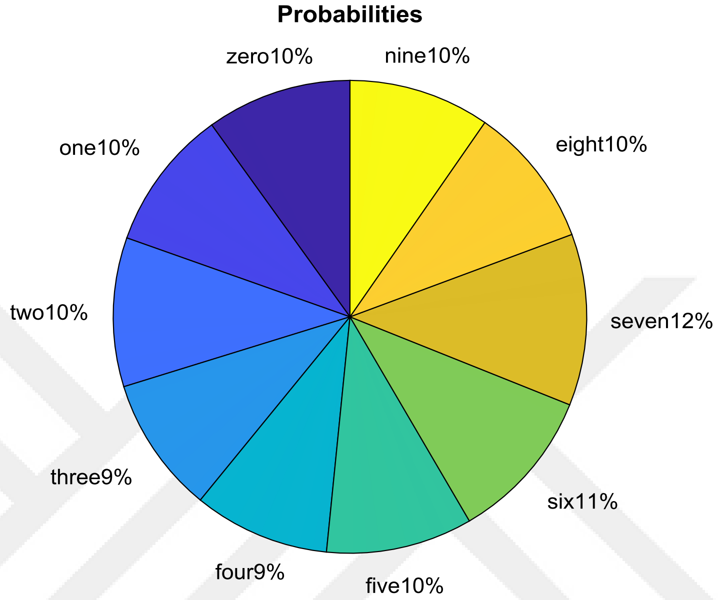

Figure 4.25 Seven for probability Ns=5 and No=220

Data-9: 'True Data' 'Number8-4y.txt' ' Predicted Data' 'eight'

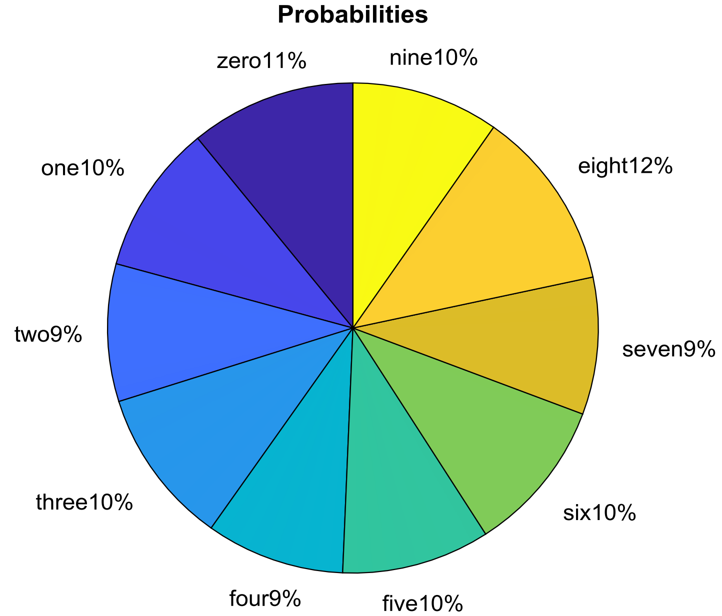

Figure 4.11 Eight for probability Ns=5 and No=220

Data-10: 'True Data'   'Number9-6y.txt'   ' Predicted Data'   'nine'

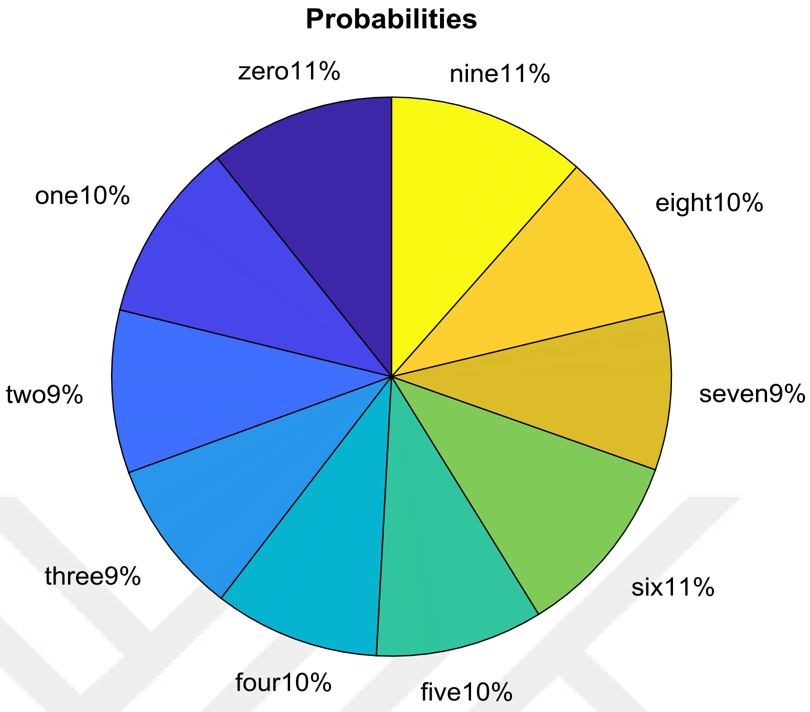

Figure 4.27 Nine for probability Ns=5 and No=220

In fact, the probability values are as follows 0.1071, 0.1045, 0.0942, 0.0899, 0.0956, 0.0967, 0.1078, 0.0916, 0.0974, 0.1150 before they are rounded. So even though the possibilities of zero and nine and others seem equal, they are not actually equal.

4.5.4 Mixed Training Set Obtained from Only Second Person

In previous parts, results of numerical, alphabet and shapes are seen that each set is individually processed and tested in itself. In this section, a system of three different sets (geometric shapes, alphabet and numerical data which are obtained by only second person) are trained together and there are 102 test data and 68 training data. The test and training data set used are shown in Table 4.7 and Table 4.8.

Table 4.7 List of three class mixed test data

LetterA1f.txt	Number1-3f.txt	Number5-5f.txt	ShapeCircle1f.txt
LetterA2f.txt	Number1-4f.txt	Number5-6f.txt	ShapeCircle2f.txt
LetterA3f.txt	Number1-5f.txt	Number6-1f.txt	ShapeCircle3f.txt
LetterA4f.txt	Number1-6f.txt	Number6-2f.txt	ShapeCircle4f.txt
LetterA5f.txt	Number2-1f.txt	Number6-3f.txt	ShapeCircle5f.txt
LetterA6f.txt	Number2-2f.txt	Number6-4f.txt	ShapeCircle6f.txt
LetterB1f.txt	Number2-3f.txt	Number6-5f.txt	ShapeLine1f.txt
LetterB2f.txt	Number2-4f.txt	Number6-6f.txt	ShapeLine2f.txt
LetterB3f.txt	Number2-5f.txt	Number7-1f.txt	ShapeLine3f.txt
LetterB4f.txt	Number2-6f.txt	Number7-2f.txt	ShapeLine4f.txt
LetterB5f.txt	Number3-1f.txt	Number7-3f.txt	ShapeLine5f.txt
LetterB6f.txt	Number3-2f.txt	Number7-4f.txt	ShapeLine6f.txt
LetterC1f.txt	Number3-3f.txt	Number7-5f.txt	ShapeRectangular1f.txt
LetterC2f.txt	Number3-4f.txt	Number7-6f.txt	ShapeRectangular2f.txt
LetterC3f.txt	Number3-5f.txt	Number8-1f.txt	ShapeRectangular3f.txt
LetterC4f.txt	Number3-6f.txt	Number8-2f.txt	ShapeRectangular4f.txt
LetterC5f.txt	Number4-1f.txt	Number8-3f.txt	ShapeRectangular5f.txt
LetterC6f.txt	Number4-2f.txt	Number8-4f.txt	ShapeRectangular6f.txt
Number0-1f.txt	Number4-3f.txt	Number8-5f.txt	ShapeTriangle1f.txt
Number0-2f.txt	Number4-4f.txt	Number8-6f.txt	ShapeTriangle2f.txt
Number0-3f.txt	Number4-5f.txt	Number9-1f.txt	ShapeTriangle3f.txt
Number0-4f.txt	Number4-6f.txt	Number9-2f.txt	ShapeTriangle4f.txt
Number0-5f.txt	Number5-1f.txt	Number9-3f.txt	ShapeTriangle5f.txt
Number0-6f.txt	Number5-2f.txt	Number9-4f.txt	ShapeTriangle6f.txt
Number1-1f.txt	Number5-3f.txt	Number9-5f.txt	
Number1-2f.txt	Number5-4f.txt	Number9-6f.txt	

Table 4.8 List of three classes mixed training data

LetterA7f.txt	Number1-8f.txt	Number5-9f.txt	Number9-10f.txt
LetterA8f.txt	Number1-9f.txt	Number5-10f.txt	ShapeCircle7f.txt
LetterA9f.txt	Number1-10f.txt	Number6-7f.txt	ShapeCircle8f.txt
LetterA10f.txt	Number2-7f.txt	Number6-8f.txt	ShapeCircle9f.txt
LetterB7f.txt	Number2-8f.txt	Number6-9f.txt	ShapeCircle10f.txt
LetterB8f.txt	Number2-9f.txt	Number6-10f.txt	ShapeLine7f.txt
LetterB9f.txt	Number2-10f.txt	Number7-7f.txt	ShapeLine8f.txt
LetterB10f.txt	Number3-7f.txt	Number7-8f.txt	ShapeLine9f.txt
LetterC7f.txt	Number3-8f.txt	Number7-9f.txt	ShapeLine10f.txt
LetterC8f.txt	Number3-9f.txt	Number7-10f.txt	ShapeRectangular7f.txt
LetterC9f.txt	Number3-10f.txt	Number8-7f.txt	ShapeRectangular8f.txt
LetterC10f.txt	Number4-7f.txt	Number8-8f.txt	ShapeRectangular9f.txt
Number0-7f.txt	Number4-8f.txt	Number8-9f.txt	ShapeRectangular10f.txt
Number0-8f.txt	Number4-9f.txt	Number8-10f.txt	ShapeTriangle7f.txt
Number0-9f.txt	Number4-10f.txt	Number9-7f.txt	ShapeTriangle8f.txt
Number0-10f.txt	Number5-7f.txt	Number9-8f.txt	ShapeTriangle9f.txt
Number1-7f.txt	Number5-8f.txt	Number9-9f.txt	ShapeTriangle10f.txt

The following results were obtained by changing the state and observation numbers, which are HMM parameters, to train the mixed training system. Ns value was changed from 2 to 20 and No value was changed from 100 to 300 as shown in Table 4.9. The best results were obtained when Ns was14 or 18 and No was 200 or 190. 75 of the 102 test data were accurately estimated and 73.5% successful.

Table 4.9 Prediction and success rate according to changing state numbers

No	Nstate	Success (%)	True Prediction Number
300	2	47.1	48
300	4	50	51
300	5	57.8	59
300	10	68.6	70
300	12	69.6	71
300	14	69.6	71
300	16	67.7	67
300	18	64.7	66
300	20	69.6	71
240	2	47.1	48
240	4	62.7	64
240	5	68.6	70
240	10	71.6	73
240	12	70.6	72
240	14	70.6	72
240	16	65.7	67
240	18	71.6	73
240	20	66.7	68
220	2	53.9	55
220	4	64.7	66
220	5	61.8	63
220	10	69.6	71
220	12	67.7	69
220	14	70.6	72
220	16	70.6	72
220	18	70.6	72
220	20	65.7	67
210	2	62.8	64
210	4	70.6	72
210	5	66.7	68
210	10	66.7	68
210	12	70.6	72
210	14	72.6	74
210	16	66.7	68
210	18	66.7	68
210	20	70.6	72
200	2	52.9	54
200	4	60.8	62
200	5	63.7	65
200	10	67.7	69
200	12	64.7	66

Table 4.9 continues

200	14	73.5	75
200	16	64.7	66
200	18	59.8	61
200	20	68.6	70
190	2	56.9	58
190	4	59.8	61
190	5	57.8	59
190	10	68.6	70
190	12	68.6	70
190	14	69.6	71
190	16	66.7	68
190	18	73.5	75
190	20	61.7	63
100	2	52	53
100	4	63.7	65
100	5	52.9	54
100	10	68.6	70
100	12	70.6	72
100	14	61.8	63
100	16	58.8	60
100	18	58.8	60
100	20	57.8	59

When No is constant, graphs of success rate according to changing state numbers are shown from Figure 4.28 to Figure 4.34. Looking at these graphs, it can be said that with increasing number of states to a certain point, success also increases. However, fluctuations are observed after state number is 14. The lowest number of predictions was taken at Ns 100. This emphasizes the importance of No as well as Ns.

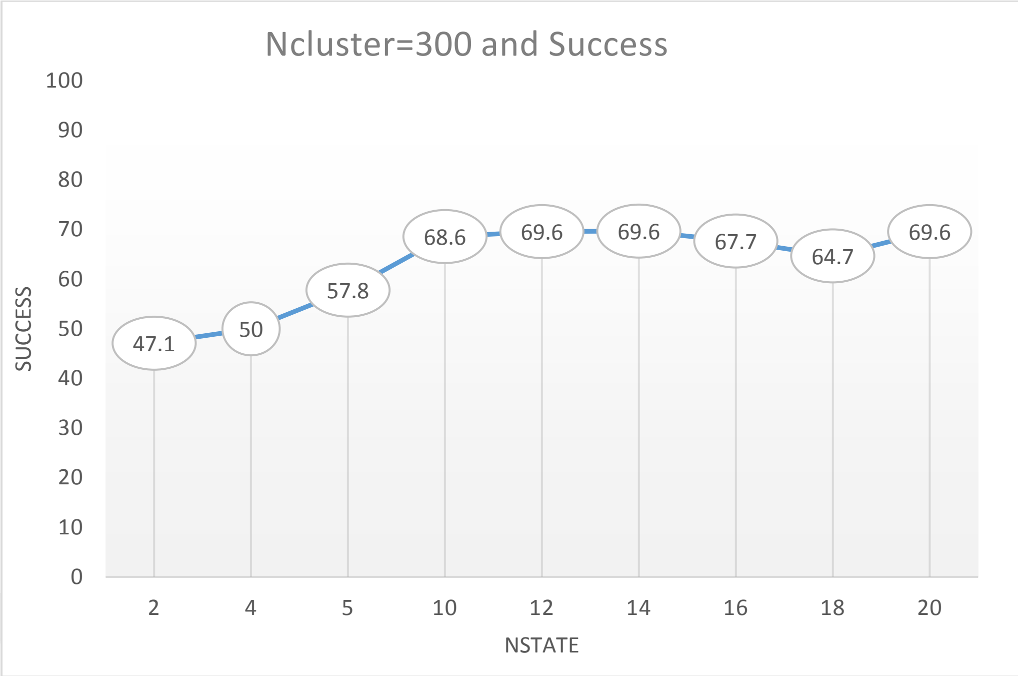

Figure 4.28 Success graph for variable Ns and No is 300

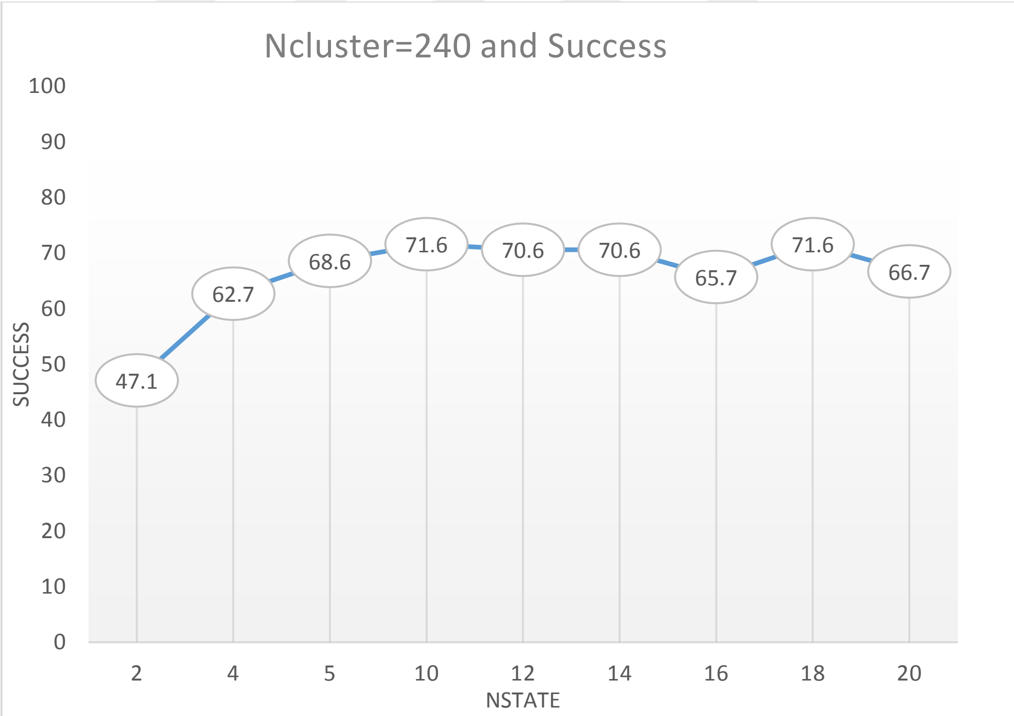

Figure 4.29 Success graph for variable Ns and No is 240

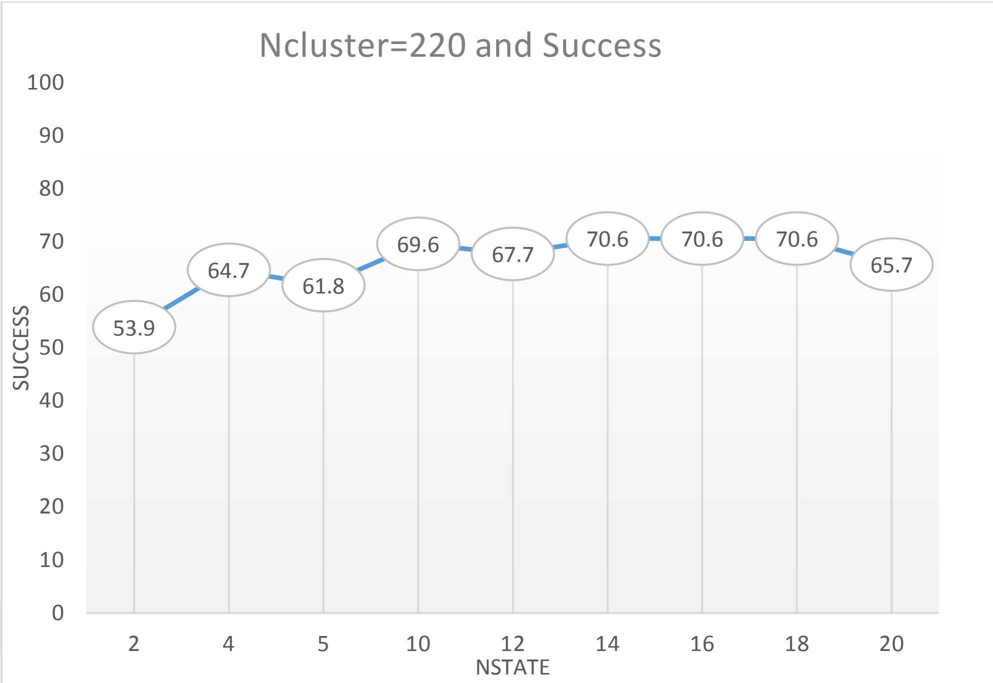

Figure 4.30 Success graph for variable Ns and No is 220

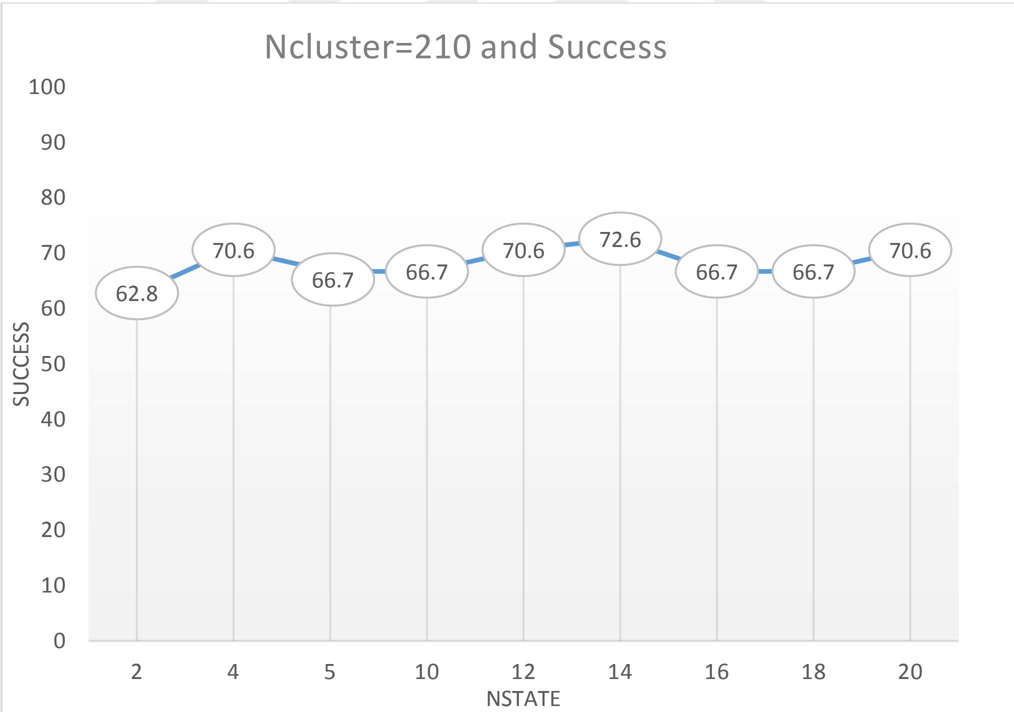

Figure 4.31 Success graph for variable Ns and No is 210

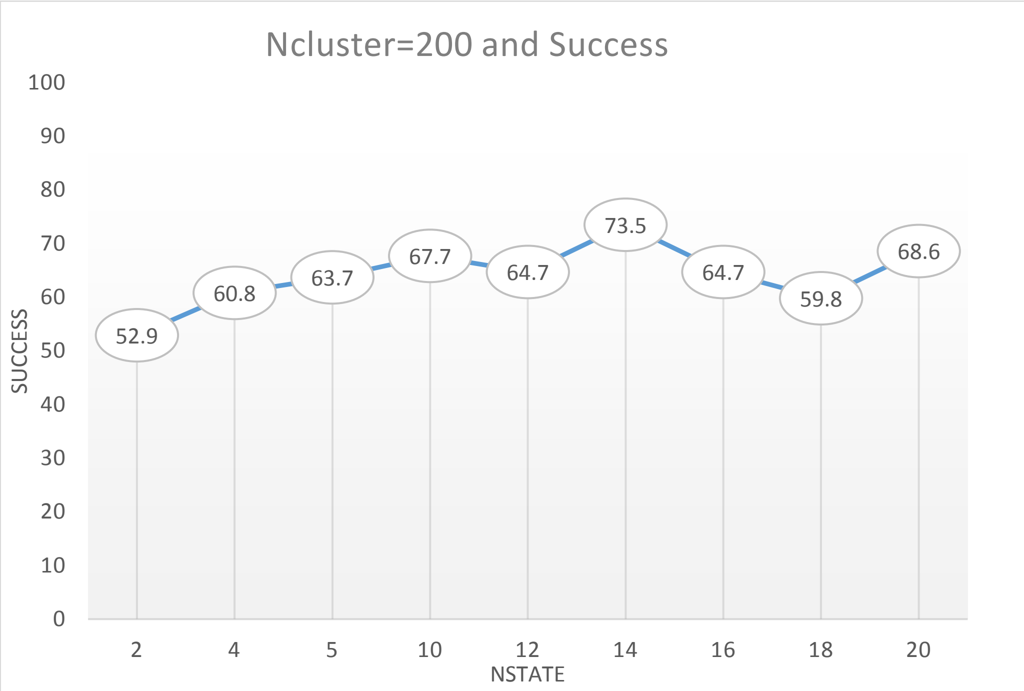

Figure 4.32 Success graph for variable Ns and No is 200

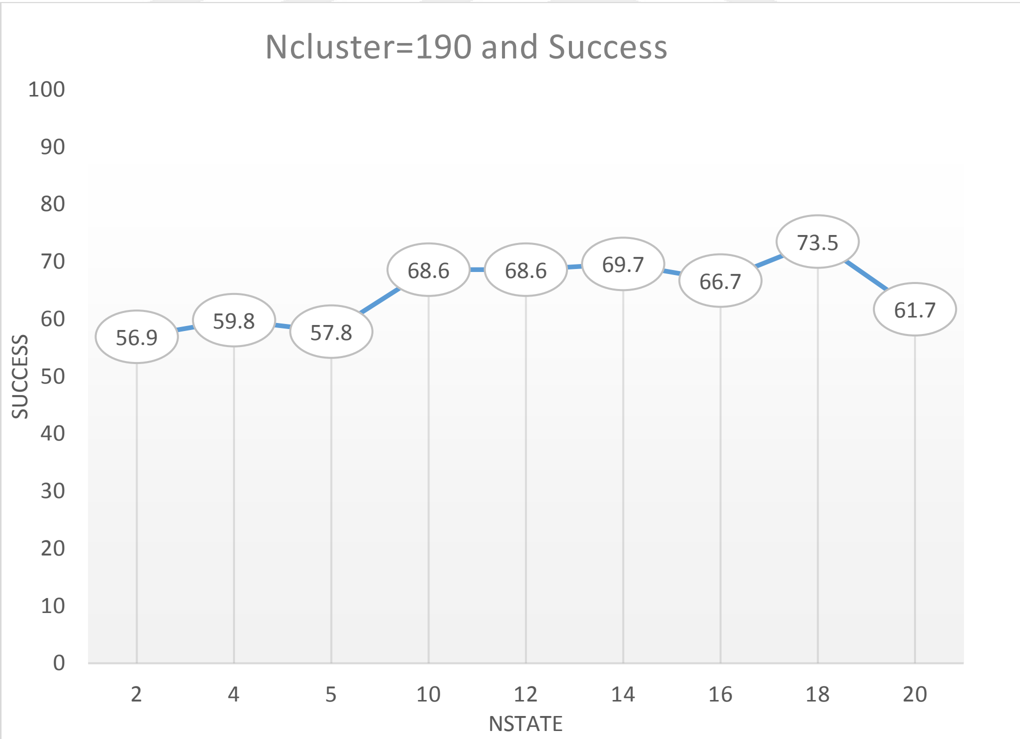

Figure 4.33 Success graph for variable Ns and No is 190

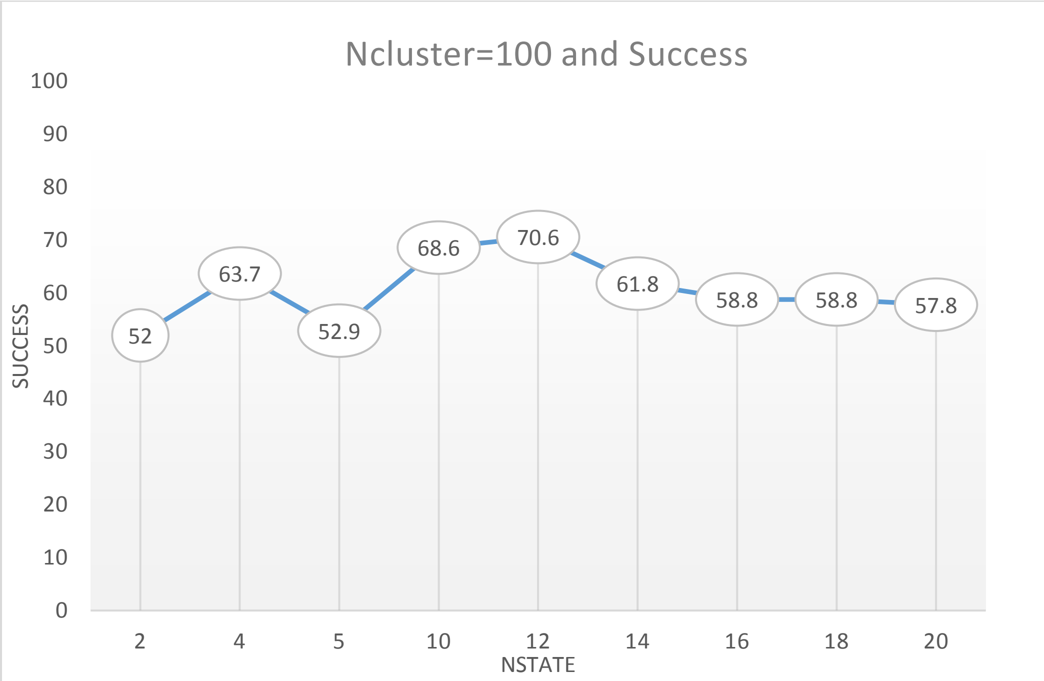

Figure 4.34 Success graph for variable Ns and No is 100

Some examples predictions of mixed test and training set obtained from only second person for Ns=14 and No=200 are shown from Figure 4.35 to Figure 4.51.

Data-1: 'True Data' 'LetterA2f.txt' ' Predicted Data' 'A'

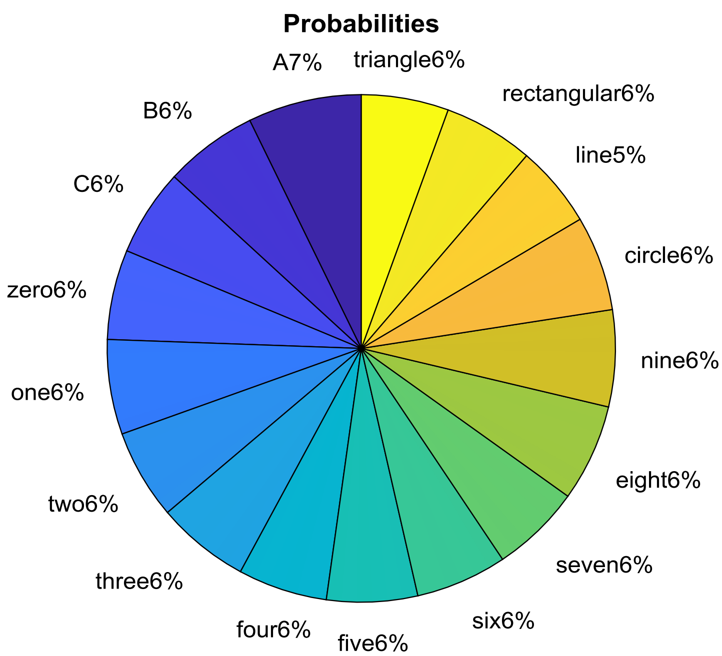

Figure 4.35 A probability for test of second person Ns=14 and No=200

Data-2: 'True Data' 'LetterB1f.txt' ' Predicted Data' 'B'

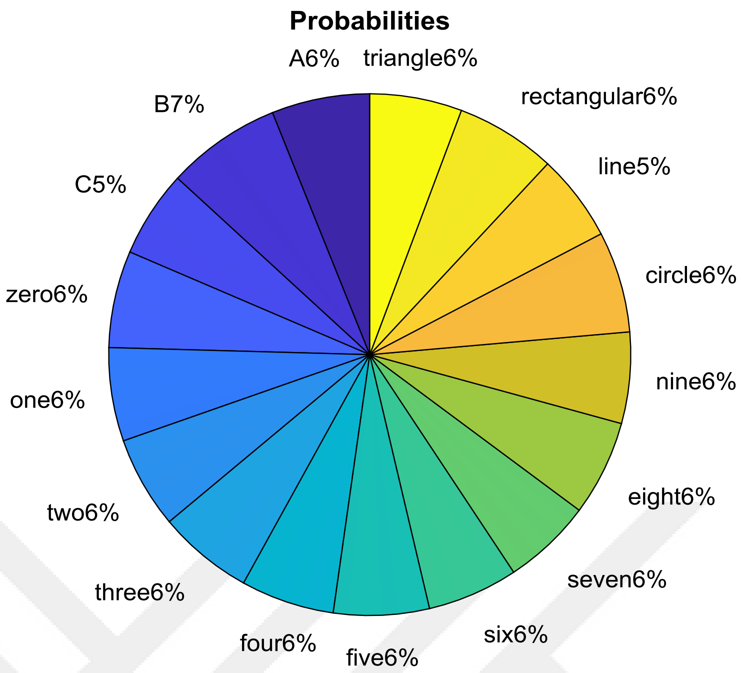

Figure 4.36 B probability for test of second person Ns=14 and No=200

Data-3: 'True Data' 'LetterC5f.txt' ' Predicted Data' 'C'

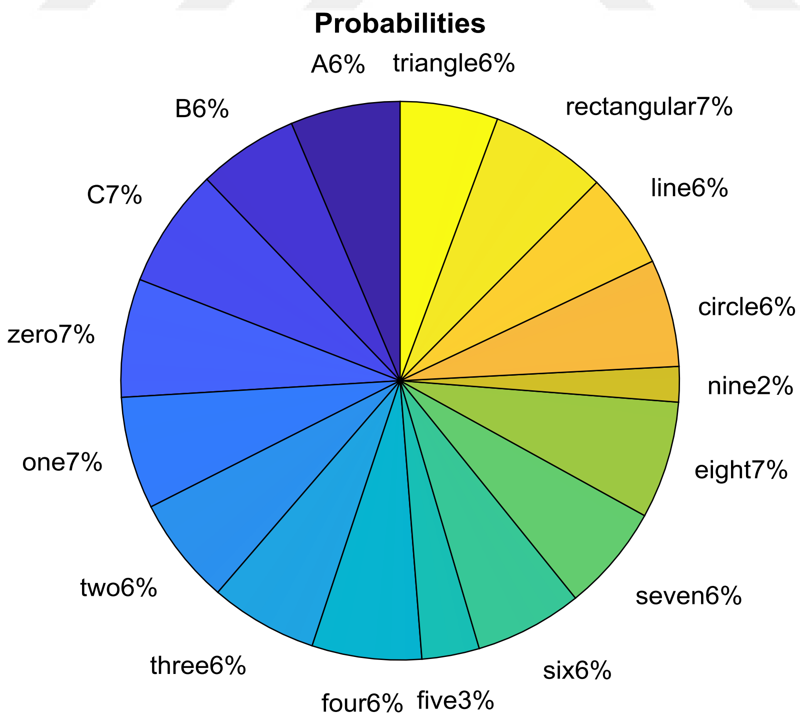

Figure 4.37 C probability for test of second person Ns=14 and No=200

Data-4: 'True Data' 'Number0-2f.txt' ' Predicted Data' 'zero'

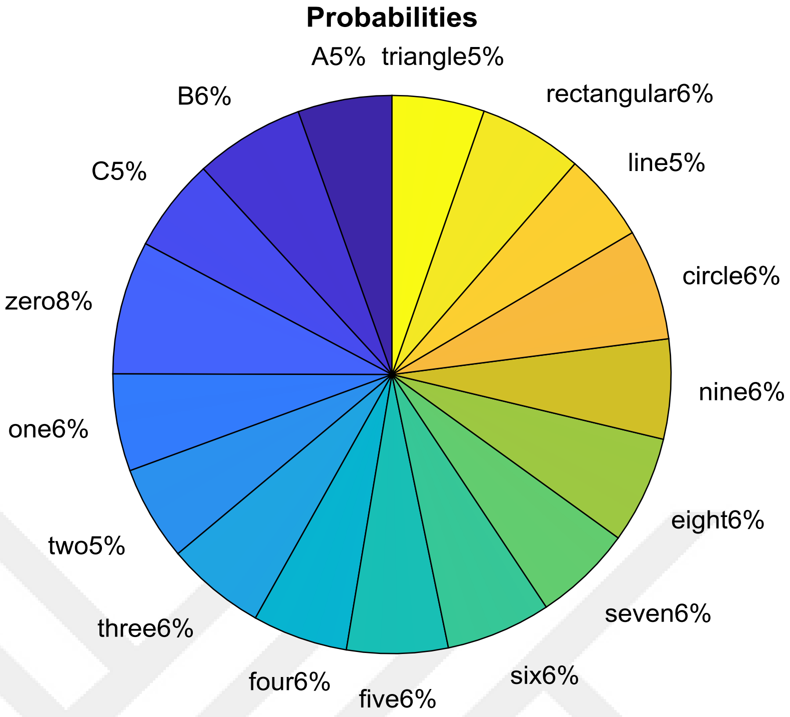

Figure 4.38 Zero probability for test of second person Ns=14 and No=200

Data-5: 'True Data' 'Number1-5f.txt' ' Predicted Data' 'one'

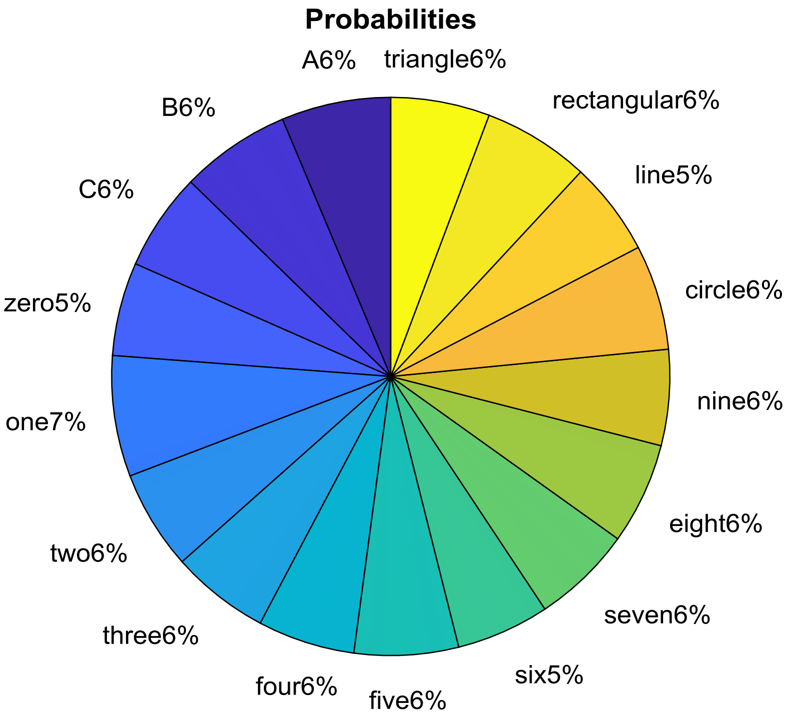

Figure 4.39 One probability for test of second person Ns=14 and No=200

Data-6: 'True Data' 'Number2-4f.txt' ' Predicted Data' 'two'

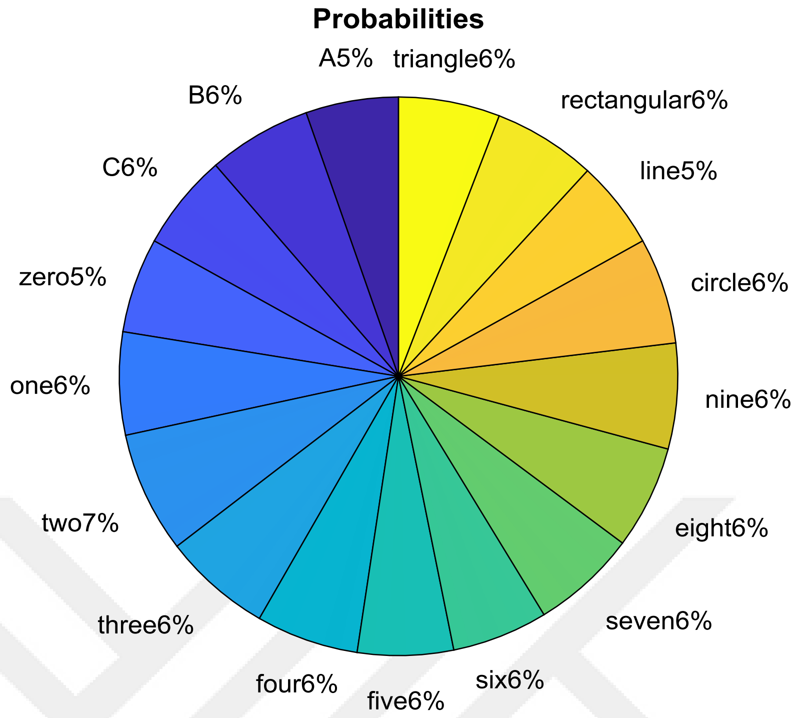

Figure 4.40 Two probability for test of second person Ns=14 and No=200

Data-7: 'True Data' 'Number3-1f.txt' ' Predicted Data' 'three'

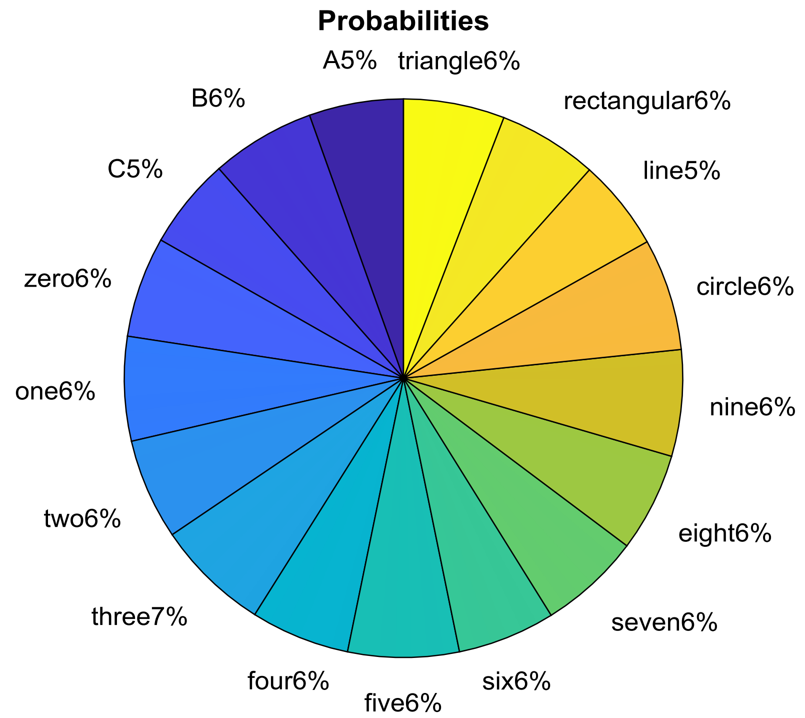

Figure 4.41 Three probability for test of second person Ns=14 and No=200

Data-8: 'True Data' 'Number4-4f.txt' ' Predicted Data' 'four'

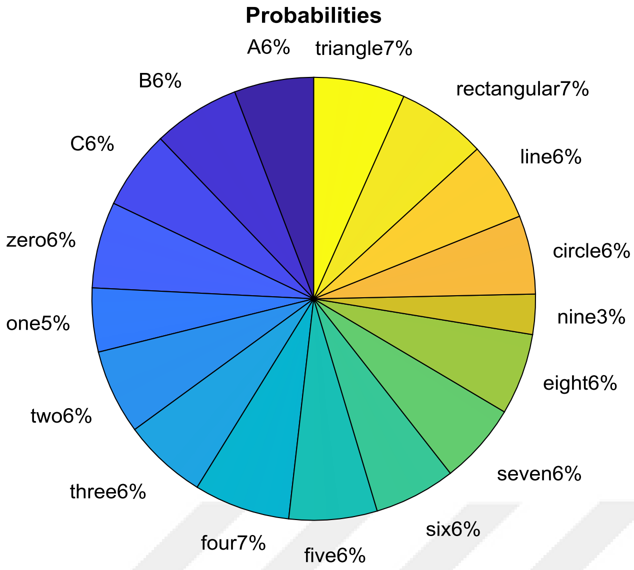

Figure 4.42 Four probability for test of second person Ns=14 and No=200

In fact, the probability values are as follows 0.0580, 0.0635, 0.0576, 0.0630, 0.0466, 0.0618, 0.0614, 0.0701, 0.0643, 0.0594, 0.0591, 0.0594, 0.0293, 0.0574, 0.0574, 0.0653, 0.0665 before they are rounded. So even though the possibilities four and triangle seem equal, they are not actually equal. Probability of four is 0.0701 and probability of triangle is 0.0665.

Data-9: 'True Data' 'Number5-6f.txt' ' Predicted Data' 'five'

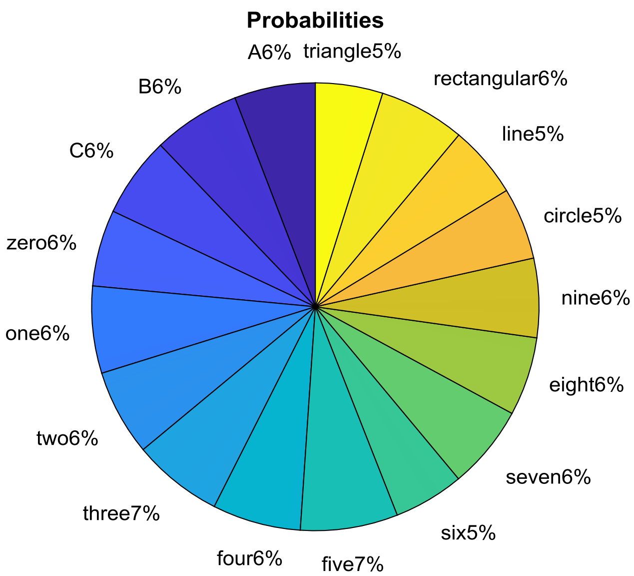

Figure 4.43 Five probability for test of second person Ns=14 and No=200

Data-10: 'True Data' 'Number6-2f.txt' ' Predicted Data' 'six'

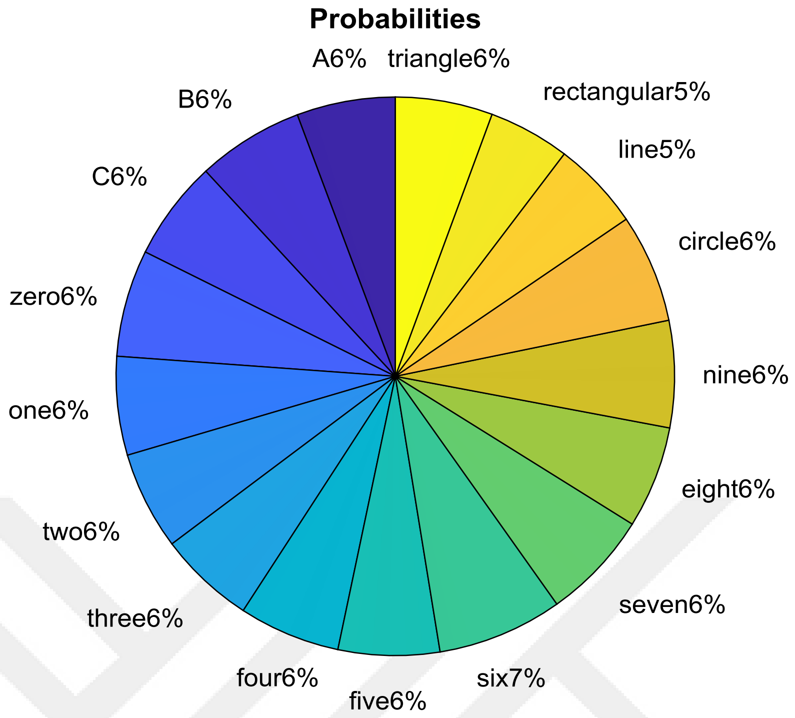

Figure 4.44 Six probability for test of second person Ns=14 and No=200

Data-11: 'True Data' 'Number7-2f.txt' ' Predicted Data' 'seven'

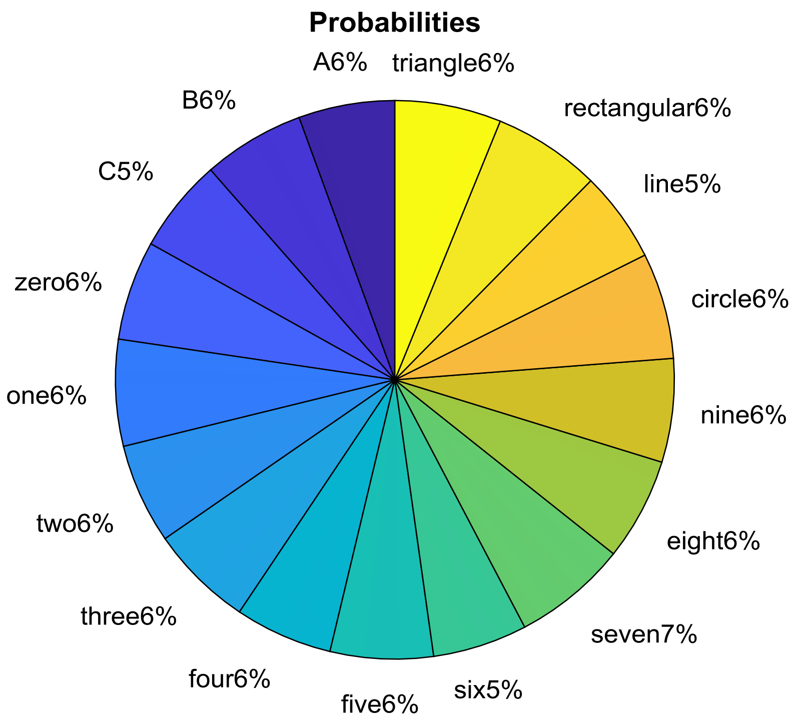

Figure 4.45 Seven probability for test of second person Ns=14 and No=200

Data-12: 'True Data'   'Number8-4f.txt'   ' Predicted Data'   'eight'

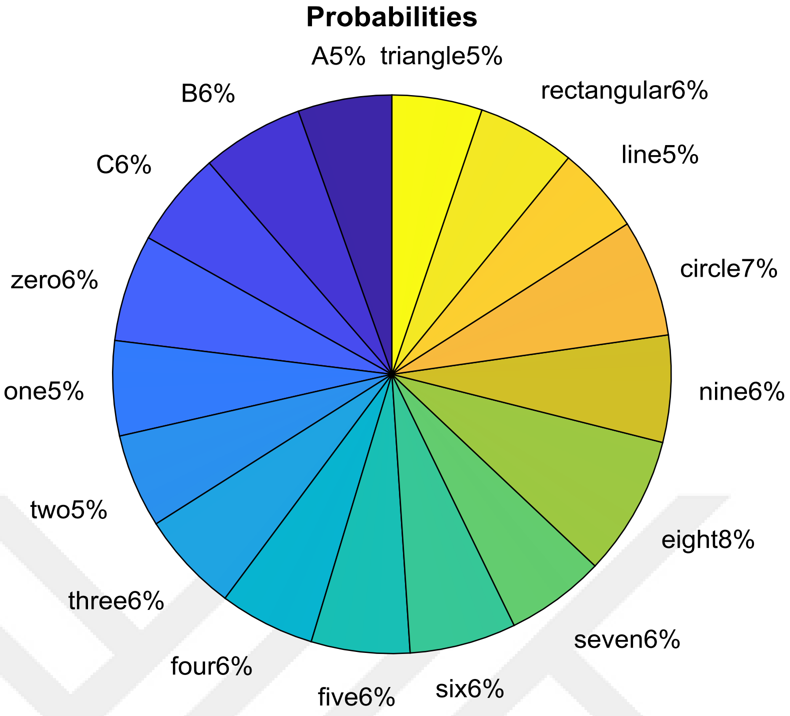

Figure 4.46 Eight probability for test of second person Ns=14 and No=200

Data-13: 'True Data'   'Number9-3f.txt'   ' Predicted Data'   'nine'

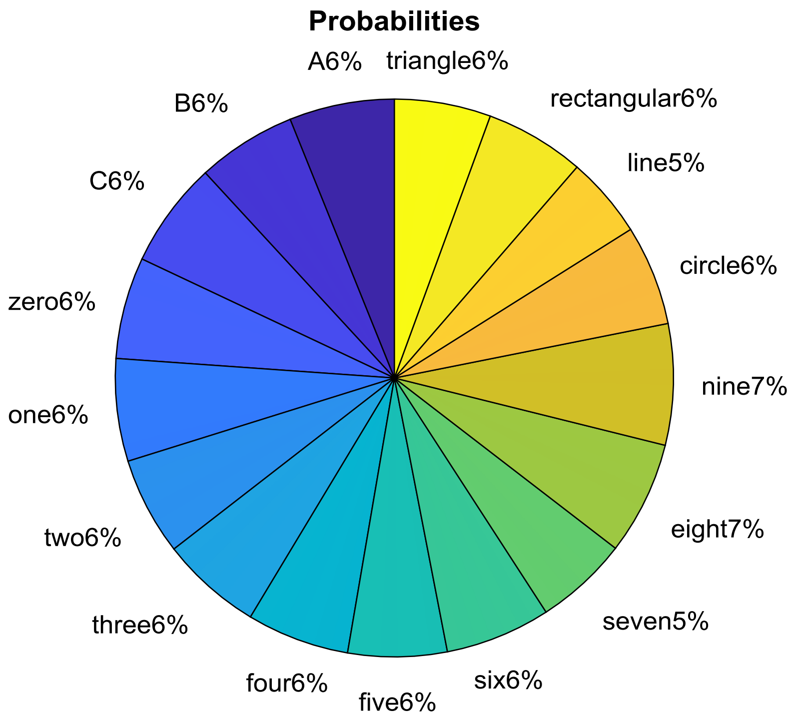

Figure 4.47 Nine probability for test of second person Ns=14 and No=200

Data-14: 'True Data' 'ShapeCircle1f.txt' ' Predicted Data' 'circle'

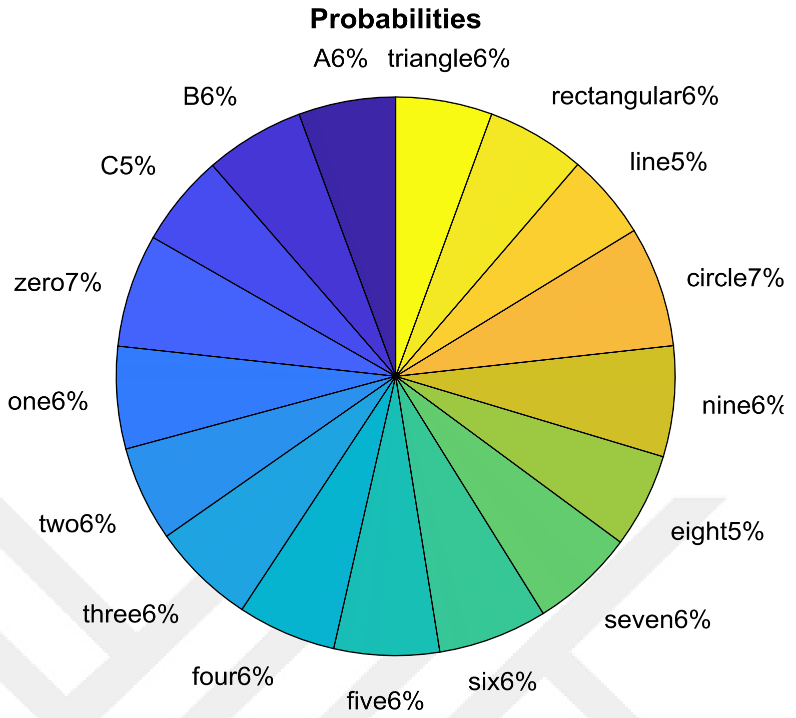

Figure 4.48 Circle probability for test of second person Ns=14 and No=200

Data-15: 'True Data' 'ShapeLine6f.txt' ' Predicted Data' 'circle'

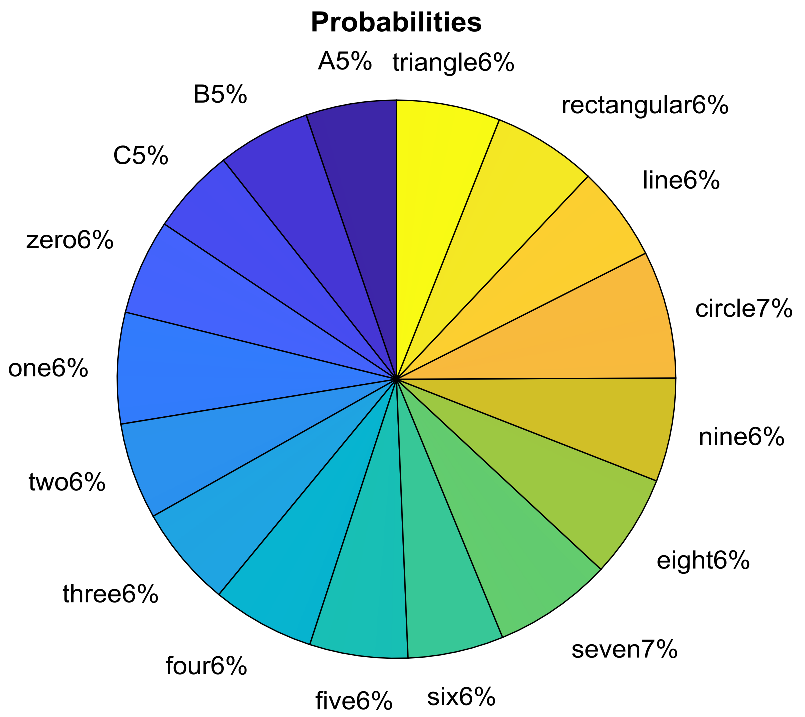

Figure 4.49 Line probability for test of second person Ns=14 and No=200

Data-16: 'True Data' 'ShapeRectangular4f.txt' ' Predicted Data' 'rectangular'

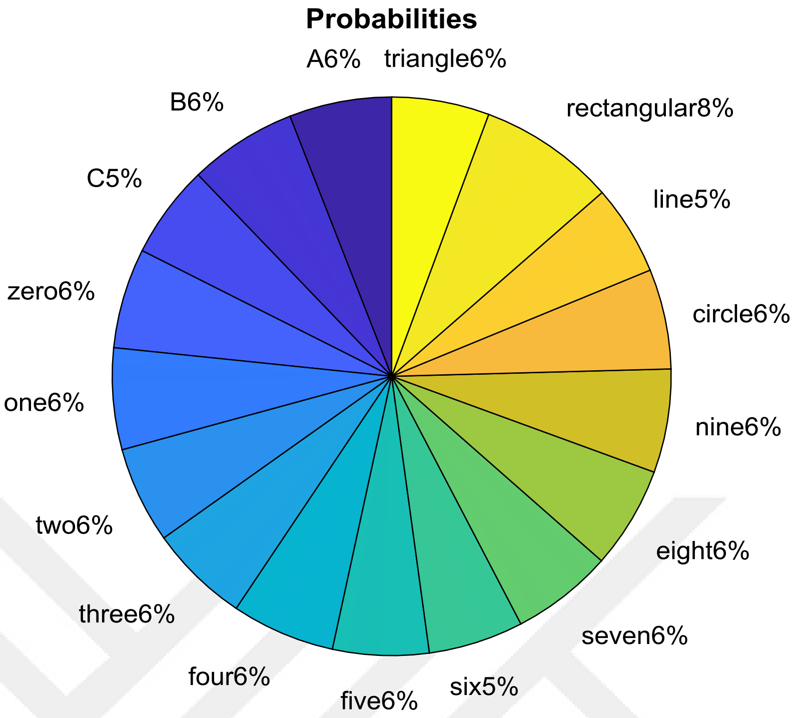

Figure 4.50 Rectangular probability for test of second person Ns=14 and No=200

Data-17: 'True Data' 'ShapeTriangle3f.txt' ' Predicted Data' 'triangle'

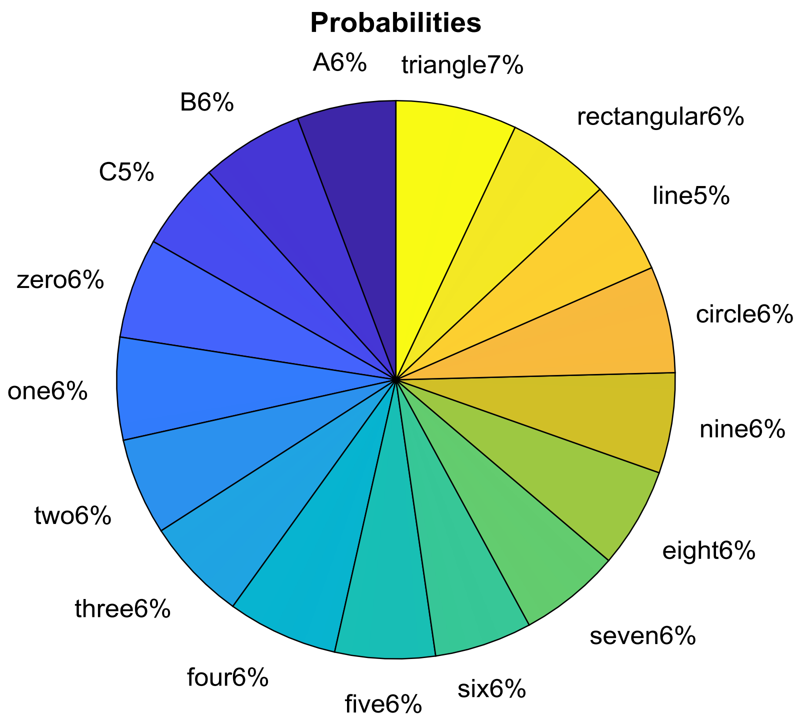

Figure 4.51 Triangle probability for test of second person Ns=14 and No=200

4.5.5 Mixed Training Set from Three Person

In this section, a training set was created with data from three people and 70 of them were estimated correctly using only 102 test data of the second person. In order to achieve this success, the data were normalized. It was observed that the success rate before normalization was lower.

Finally, both the training set and the test set consisted of data from three people and the success rate was observed. The second column in Table 4.10 shows the test data and the last column shows the estimated class. When the Ns number was 10 and the No number was 240, 106 test data were used, of which only 50 were correctly estimated.

Table 4.10 Mixed training and test set obtained from three person

'True Data'	'LetterA1a.txt'	'Predicted Data'	'A'
'True Data'	'LetterA1f.txt'	'Predicted Data'	'A'
'True Data'	'LetterA1y.txt'	'Predicted Data'	'five'
'True Data'	'LetterA2a.txt'	'Predicted Data'	'A'
'True Data'	'LetterA2f.txt'	'Predicted Data'	'A'
'True Data'	'LetterA2y.txt'	'Predicted Data'	'six'
'True Data'	'LetterA3a.txt'	'Predicted Data'	'A'
'True Data'	'LetterB1a.txt'	'Predicted Data'	'B'
'True Data'	'LetterB1f.txt'	'Predicted Data'	'B'
'True Data'	'LetterB1y.txt'	'Predicted Data'	'A'
'True Data'	'LetterB2a.txt'	'Predicted Data'	'B'
'True Data'	'LetterB2f.txt'	'Predicted Data'	'B'
'True Data'	'LetterB2y.txt'	'Predicted Data'	'triangle'
'True Data'	'LetterC1a.txt'	'Predicted Data'	'C'
'True Data'	'LetterC1f.txt'	'Predicted Data'	'C'
'True Data'	'LetterC1y.txt'	'Predicted Data'	'line'

Table 4.10 continues

'True Data'	'LetterC2a.txt'	'Predicted Data'	'two'
'True Data'	'LetterC2f.txt'	'Predicted Data'	'C'
'True Data'	'LetterC2y.txt'	'Predicted Data'	'two'
'True Data'	'LetterC3a.txt'	'Predicted Data'	'C'
'True Data'	'Number0-1a.txt'	'Predicted Data'	'six'
'True Data'	'Number0-1f.txt'	'Predicted Data'	'four'
'True Data'	'Number0-1y.txt'	'Predicted Data'	'three'
'True Data'	'Number0-2a.txt'	'Predicted Data'	'eight'
'True Data'	'Number0-2f.txt'	'Predicted Data'	'circle
'True Data'	'Number0-2y.txt'	'Predicted Data'	'B'
'True Data'	'Number1-1a.txt'	'Predicted Data'	'one'
'True Data'	'Number1-1f.txt'	'Predicted Data'	'six'
'True Data'	'Number1-1y.txt'	'Predicted Data'	'B'
'True Data'	'Number1-2a.txt'	'Predicted Data'	'seven'
'True Data'	'Number1-2f.txt'	'Predicted Data'	'three'
'True Data'	'Number1-2y.txt'	'Predicted Data'	'B'
'True Data'	'Number2-1a.txt'	'Predicted Data'	'three'
'True Data'	'Number2-1f.txt'	'Predicted Data'	'two'
'True Data'	'Number2-1y.txt'	'Predicted Data'	'one'
'True Data'	'Number2-2a.txt'	'Predicted Data'	'two'
'True Data'	'Number2-2f.txt'	'Predicted Data'	'two'
'True Data'	'Number2-2y.txt'	'Predicted Data'	'B'
'True Data'	'Number3-1a.txt'	'Predicted Data'	'three'
'True Data'	'Number3-1f.txt'	'Predicted Data'	'B'
'True Data'	'Number3-1y.txt'	'Predicted Data'	'one'
'True Data'	'Number3-2a.txt'	'Predicted Data'	'three'
'True Data'	'Number3-2f.txt'	'Predicted Data'	'B'
'True Data'	'Number3-2y.txt'	'Predicted Data'	'B

Table 4.10 continues

'True Data'	'Number4-1f.txt'	' Predicted Data'	'four'
'True Data'	'Number4-1y.txt'	' Predicted Data'	'three'
'True Data'	'Number4-2a.txt'	' Predicted Data'	'triangle'
'True Data'	'Number4-2f.txt'	' Predicted Data'	'four'
'True Data'	'Number4-2y.txt'	' Predicted Data'	'triangle'
'True Data'	'Number5-1a.txt'	' Predicted Data'	'five'
'True Data'	'Number5-1f.txt'	' Predicted Data'	'rectangular'
'True Data'	'Number5-1y.txt'	' Predicted Data'	'B'
'True Data'	'Number5-2a.txt'	' Predicted Data'	'five'
'True Data'	'Number5-2f.txt'	' Predicted Data'	'B'
'True Data'	'Number5-2y.txt'	' Predicted Data'	'B'
'True Data'	'Number6-1a.txt'	' Predicted Data'	'six'
'True Data'	'Number6-1f.txt'	' Predicted Data'	'six'
'True Data'	'Number6-1y.txt'	' Predicted Data'	'triangle'
'True Data'	'Number6-2a.txt'	' Predicted Data'	'six'
'True Data'	'Number6-2f.txt'	' Predicted Data'	'six'
'True Data'	'Number6-2y.txt'	' Predicted Data'	'line'
'True Data'	'Number6-3a.txt'	' Predicted Data'	'six'
'True Data'	'Number7-1a.txt'	' Predicted Data'	'B'
'True Data'	'Number7-1f.txt'	' Predicted Data'	'five'
'True Data'	'Number7-1y.txt'	' Predicted Data'	'triangle'
'True Data'	'Number7-2a.txt'	' Predicted Data'	'triangle'
'True Data'	'Number7-2f.txt'	' Predicted Data'	'seven'
'True Data'	'Number7-2y.txt'	' Predicted Data'	'triangle'
'True Data'	'Number8-1a.txt'	' Predicted Data'	'eight'
'True Data'	'Number8-1f.txt'	' Predicted Data'	'eight'
'True Data'	'Number8-1y.txt'	' Predicted Data'	'five'
'True Data'	'Number8-2a.txt'	' Predicted Data'	'eight'

Table 4.10 continues

'True Data'	'Number8-2f.txt'	' Predicted Data'	'eight'
'True Data'	'Number8-2y.txt'	' Predicted Data'	'B'
'True Data'	'Number9-1a.txt'	' Predicted Data'	'six'
'True Data'	'Number9-1f.txt'	' Predicted Data'	'B'
'True Data'	'Number9-1y.txt'	' Predicted Data'	'A'
'True Data'	'Number9-2a.txt'	' Predicted Data'	'five'
'True Data'	'Number9-2f.txt'	' Predicted Data'	'nine'
'True Data'	'Number9-2y.txt'	' Predicted Data'	'B'
'True Data'	'ShapeCircle1a.txt'	' Predicted Data'	'six'
'True Data'	'ShapeCircle1f.txt'	' Predicted Data'	'triangle'
'True Data'	'ShapeCircle1y.txt'	' Predicted Data'	'zero'
'True Data'	'ShapeCircle2a.txt'	' Predicted Data'	'eight'
'True Data'	'ShapeCircle2f.txt'	' Predicted Data'	'four'
'True Data'	'ShapeCircle2y.txt'	' Predicted Data'	'line'
'True Data'	'ShapeCircle3a.txt'	' Predicted Data'	'circle'
'True Data'	'ShapeLine1a.txt'	' Predicted Data'	'line'
'True Data'	'ShapeLine1f.txt'	' Predicted Data'	'line'
'True Data'	'ShapeLine1y.txt'	' Predicted Data'	'three'
'True Data'	'ShapeLine2a.txt'	' Predicted Data'	'line'
'True Data'	'ShapeLine2f.txt'	' Predicted Data'	'line'
'True Data'	'ShapeLine2y.txt'	' Predicted Data'	'three'
'True Data'	'ShapeRectangular1a.txt'	' Predicted Data'	'circle'
'True Data'	'ShapeRectangular1f.txt'	' Predicted Data'	'rectangular'
'True Data'	'ShapeRectangular1y.txt'	' Predicted Data'	'nine'
'True Data'	'ShapeRectangular2a.txt'	' Predicted Data'	'rectangular'
'True Data'	'ShapeRectangular2f.txt'	' Predicted Data'	'rectangular'
'True Data'	'ShapeRectangular2y.txt'	' Predicted Data'	'zero'
'True Data'	'ShapeTriangle1a.txt'	' Predicted Data'	'triangle'

Table 4.10 continues

'True Data'	'ShapeTriangle1f.txt'	' Predicted Data'	'triangle'
'True Data'	'ShapeTriangle1y.txt'	' Predicted Data'	'A'
'True Data'	'ShapeTriangle2a.txt'	' Predicted Data'	'triangle'
'True Data'	'ShapeTriangle2f.txt'	' Predicted Data'	'rectangular'
'True Data'	'ShapeTriangle2y.txt'	' Predicted Data'	'triangle'

Some predictions of mixed training shape test set for Ns=10 and No=240 are shown from Figure 4.52 to Figure 4.54. They are the test results of the letter A, taken from three different people. Test data were called as LetterA1a.txt, LetterA1f.txt, LetterA1y.txt.

Data-1: ‘True Data’ ‘LetterA1y.txt’ ‘ Predicted Data’ ‘five’

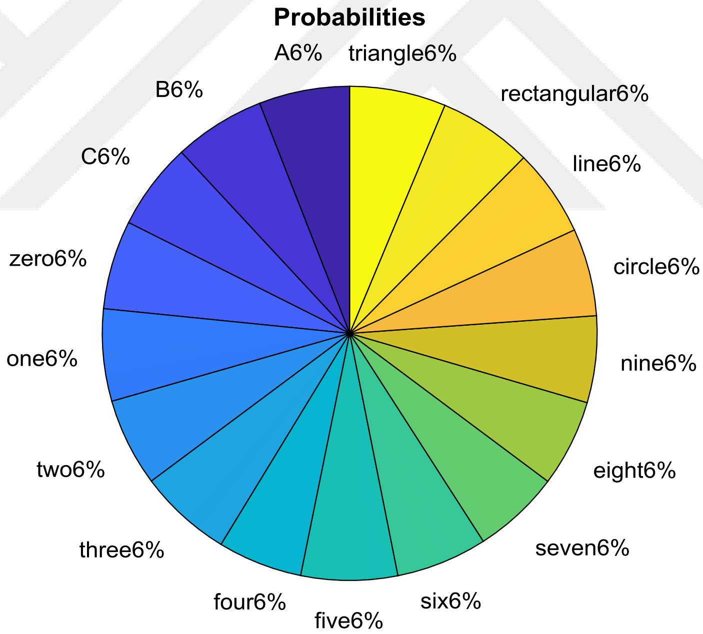

Figure 4.52 A probability with a mixed test set from three person Ns=10 and No=240

There is wrong prediction. Because, the probability values are as follows 0.0593, 0.0597, 0.0569, 0.0581, 0.0603, 0.0577, 0.0610, 0.0553, 0.0630, 0.0596, 0.0570, 0.0570, 0.0570, 0.0573, 0.0571, 0.0612, 0.0628 before they are rounded. So even though the possibilities letter A and five seem equal, they are not actually equal. Probability of letter A is 0.0593 and probability of five is 0.0630.

Data-2: 'True Data'   'LetterA1a.txt'   ' Predicted Data'   'A'

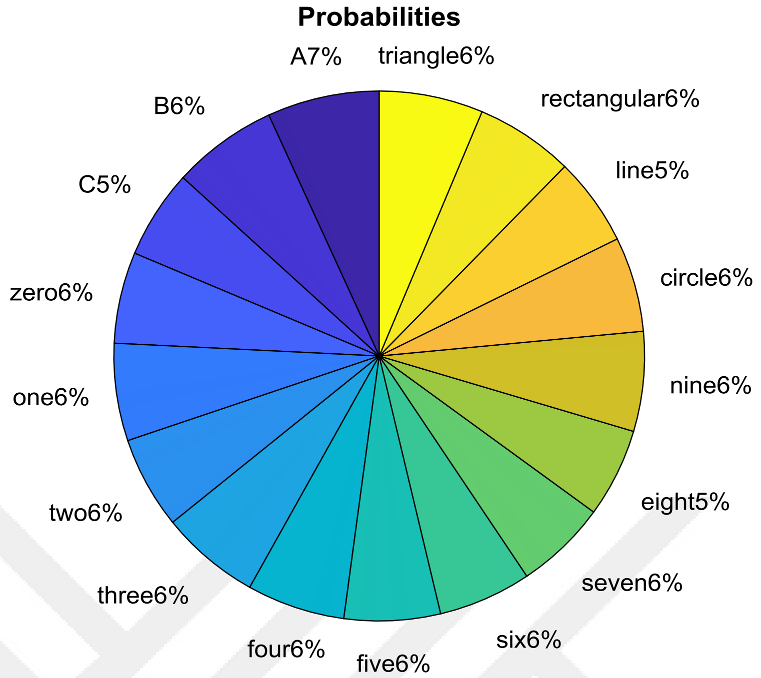

Figure 4.53 A probability with a mixed test set from three person Ns=10 and No=240

Data-2: 'True Data'   'LetterA1f.txt'   ' Predicted Data'   'A'

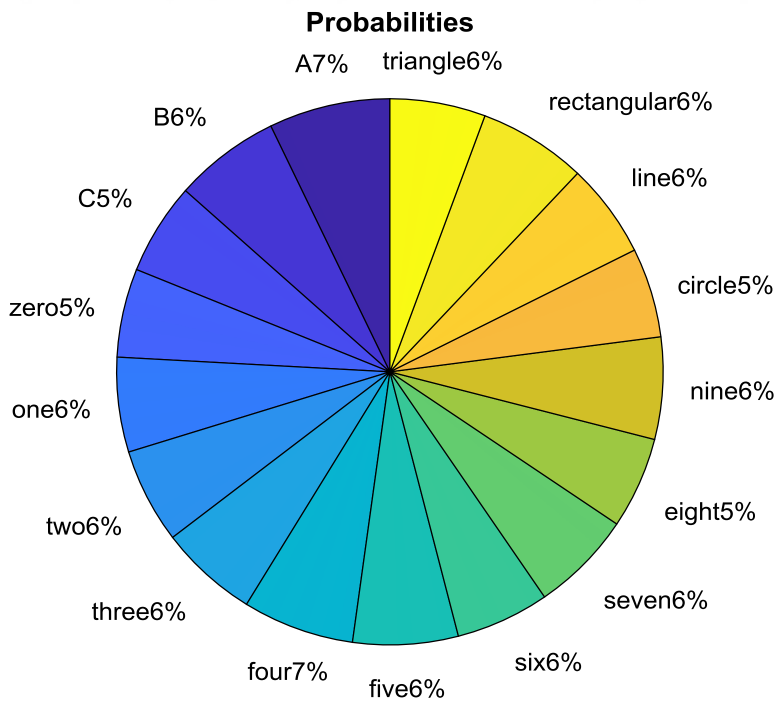

Figure 4.54 A probability with a mixed test set from three person Ns=10 and No=240

CHAPTER FIVE

CONCLUSION

Gesture recognition has become quite common nowadays. In this study, various tools and algorithms used in previous gesture recognition projects are briefly introduced and advantages and disadvantages of algorithms are mentioned.

In this work, MPU6050 sensor is used to recognize the characters written in air by one user. The device used is observed to be more portable than alternatives due to being smaller. So, air writing technology with MPU6050 will be a technology that can replace digital-based writing in the future.

Firstly, a total of three sets were created by the first person for the preliminary study. Then, this three different test and training sets, which contained shapes, numbers and capital letter, were used to train an HMM based recognition system by using values of No and Ns. In this way three different recognition systems were generated. Recognition of a test data was done in respect to selected group class. When the experimental results were examined for the geometrical shape, alphabetical and numerical test data sets, it was seen that 75%, 67% and 80% successes were obtained respectively using Ns. However, when the rates of success were seen to be low, the Ns and No values were changed and the systems were trained again and the success rate was observed to increase. It was seen that 100% success was achieved in geometric shape and alphabetical test data sets, but the numerical success rate remained the same. Since a small number of data are used for this preliminary study, the number of data has been increased considering that the success rates do not reflect the reality. Data were collected from three different individuals and mixed training sets and test sets were created.

Secondly, both the mixed training and the mixed test set were created by the second person and there are totally 170 data. This mixed test set including shapes, numbers and alphabet was used and the success rate was found to be about 74%.

Finally, more data was saved to create a mixed set that included 17 class obtained from three person. Using this mixed training system, 102 test data of the only second person were tested and 68.6% success was achieved. After, a mixed test set obtained from three people, was tested with the mixed training set obtained three people. The rate of success was observed to be around 50%. In order to increase this success rate, one-to-one movement can be drawn in the same place at the same time. Thus, the differences between the data are reduced and the success for the mixed test set can be increased.

306 data used in this study were transferred to computer via sensor interface. Also, sensor can connect to computer via MATLAB but it was observed that the data received via Bluetooth cannot be transferred to the graph simultaneously. Improvements can be made in later studies and data can be obtained instant from the sensor, can be recognized and analyzed.

REFERENCES

Agrawal, S., Constandache, I., Gaonkar, S., Choudhury, R. R., Caves, K., & Ruyter, F. D. (2011). *Using mobile phones to write in air*. Retrieved August 23, 2019, from https://synrg.csl.illinois.edu/papers/mobi198-agrawal.pdf.

Al-Bayaty, R. (2015). *On gesture recognition from motion sensor data*. Retrieved August 26, 2019, from https://github.com/ralbayaty/GraduateProjects/blob/master/gestureRecognition/Paper/EEL6825_Final_Report_Albayaty.pdf.

Ayaz, O., & Alp, S. (2018). Saklı Markov modeli kullanılarak İstanbul'daki üniversite öğrencilerinin GSM operatör tercihlerini etkileyen faktörlerin analizi. *Çukurova University Journal of the Faculty of Engineering and Architecture, 33*(4), 203-212.

Aydınoğlu, C. (2015). *Jiroskop sensörü nasıl çalışır?*. Retrieved August 15, 2019, from https://www.elektrikport.com/teknik-kutuphane/jiroskop-sensoru-nasil-calisir/16721#ad-image-0.

Bond, B. (2015). *1 MEMS (microelectromechanical systems) growing in a shrinking world a seminar on*. Retrieved October 10, 2019, from https://slideplayer.com/slide/7404384/.

Dhinakaran, V. (2014). *Artificial neural network for hand gesture recognition*. Retrieved August 26, 2019, from https://www.slideshare.net/VigneshwerViki/artificial-neural-network-for-hand-gesture-recognition?from_action=save.

Dimensionengineering, (n.d). *A beginner's guide to accelerometers*. Retrieved August 17, 2019, from https://www.dimensionengineering.com/info/accelerometers.

Ecer, O., Yetgin, Z., & Celik, T. (2018). Air write letter recognition using random forest classification on Arduino dataset. *International Journal of Scientific and Technological Research, 4* (7), 1-9.

Fang, C. (2009). *From Dynamic Time Warping (DTW) to hidden Markov model (HMM).* Retrieved August 29, 2019, from https://pdfs.semanticscholar.org/048c/d300b39c49eab62a4f91457ea3a0aa6b9bb7.pdf.

Gesture Recognition, (n.d.). Retrieved August 18, 2019, from https://www.techopedia.com/definition/618/gesture-recognition.

Gillian, N. (2019). How to train parameters for HMM (gesture recognition)? [Msg 1]. Message posted to https://www.researchgate.net/post/How_to_train_parameters_for_HMM_Gesture_Recognition10.

Grady, E. (2010). *Sign language glove.* Retrieved August 25, 2019, from https://www.sparkfun.com/news/453.

Gümüş, Ö. F. (2015) *MEMS (mikro elektrik-mekanik sistemler) nedir?.* Retrieved August 16, 2019, from https://www.elektrikport.com/universite/mems-(mikro-elektrik-mekanik-sistemler)-nedir/16797#ad-image-0.

Hidden Markov models (HMM), (n.d.). Retrieved August 23, 2019, from https://in.mathworks.com/help/stats/hidden-markov-models-hmm.html#bq_i1wh.

HMM_problems, (n.d). *The three basic problems for HMMs.* Retrieved August 26, 2019, from http://www.phon.ox.ac.uk/jcoleman/new_SLP/Lecture_2/HMM_problems.htm.

Kumar, H. (2014). *Gesture recognition using hidden Markov models augmented with active difference signatures.* M.Sc. Thesis, Rochester Institute of Technology, Rochester, NY.

Li, X. (n.d.). *Hand gesture recognition technology in human-computer interaction.* Retrieved August 17, 2019, from https://pdfs.semanticscholar.org/d466/aad270523d8a4823184e76f0fefedf8c32fe.pdf.

Meenaakumari, M., & Muthulakshmi, M. (2013). MEMS accelerometer based hand gesture recognition. *International Journal of Advanced Research in Computer Engineering & Technology (IJARCET), 2*(5), 1886-1892.

MEMS, (2017). Retrieved August 18, 2019, from https://ethw.org/MEMS.

Murata, T., & Shin, J. (2014). *Hand gesture and character recognition based on kinect sensor*. Retrieved August 22, from 2019, https://journals.sagepub.com/doi/full/10.1155/2014/278460.

Nagashree, R. N., Michahial, S., Aishwarya, G. N., Azeez, B. H., Jayalakshmi, M. R., & Rani, R. K. (2015). Hand gesture recognition using support vector machine. *The International Journal Of Engineering And Science (IJES), 4*(6), 42-46.

Öcal, K. (2005). *Otomatik konuşma tanıma algoritmalarının uygulamaları.* M.Sc. Thesis, Ankara Üniversitesi, Ankara.

Polat, Ç. (n.d.). *Jiroskop sensörü nedir?.* Retrieved August 16, 2019, from https://www.erasistem.com/sss/soru/jiroskop-sensoru-nedir.

Pradipa, R., & Kavitha, S. (2014). Hand gesture recognition – analysis of various techniques, methods and their algorithms. *2014 International Conference on Innovations in Engineering and Technology (ICIET'14), 3* (3), 2003-2010.

Premaratne, P. (2014). *Historical development of hand gesture recognition.* Retrieved September 6, 2019, from https://pdfs.semanticscholar.org/3416/9a33e0666bb82fbe927f5d6020e2f28bef96.pdf.

Samancı, B. (2011). *Accelerometer, Gyroscope, IMU nedir?.* Retrieved August 16, 2019, from http://www.barissamanci.net/Makale/26/accelerometer-gyroscope-imu-nedir/.

Schechter, S. (2014). *What is gesture recognition? Gesture recognition defined.* Retrieved August 18, 2019, from https://www.marxentlabs.com/what-is-gesture-recognition-defined/.

Sezen, F. (2014). *Saklı Markov modeli.* Retrieved August 26, 2019, https://prezi.com/pccnewuqfolu/sakl-markov-modeli/.

Şeker, Ş. E. (2008). *SVM (Support vector machine, destekçi vektör makinesi).* Retrieved August 26, 2019, from http://bilgisayarkavramlari.sadievrenseker.com/2008/12/01/svm-support-vector-machine-destekci-vektor-makinesi/.

Şeker, Ş. E. (2008). *K-Ortalama algoritması (K-Means Algorithm).* Retrieved January 10, 2020, from http://bilgisayarkavramlari.sadievrenseker.com/2008/12/15/k-ortalama-algoritmasi-k-means-algorithm/.

Thonti, V. (2018). *Different types of sensors and their working.* Retrieved August 15, 2019, from https://circuitdigest.com/tutorial/different-types-of-sensors-and-their-working.

Toktaş, A. (2014). *İvmeölçer nedir? Nasıl çalışır?.* Retrieved August 13, 2019, from https://www.elektrikport.com/teknik-kutuphane/ivmeolcer-nedir-nasil-calisir/12216#ad-image-0.

Tuncer, E. (2016). *Accelerometer based handwritten character recognition using dynamic time warping.* M.Sc. Thesis, İzmir Institute of Technology, İzmir.

Tutty, J. (2017). *How to avoid vomit-inducing interactive experiences | Mumbrella360 video.* Retrieved August 25, 2019, from https://mumbrella.com.au/how-to-avoid-vomit-inducing-interactive-experiences-mumbrella360-video-480691.

Vidya, M. (n.d.). *What is a sensor? Different types of sensors with applications.* Retrieved August 14, 2019, from https://www.electricaltechnology.org/2018/11/types-sensors-applications.html.

Wit motion, (n.d.). Retrieved August 15, 2019, from http://www.wit-motion.com/english.php?m=text&a=index&classify_id=46396.

APPENDICES

APPENDIX 1: MATLAB GUI

The following Figure A.1, Figure A.2, Figure A.3, Figure A.4 and Figure A.5, Figure A.6 show the MATLAB GUI for test data of geometric shapes, alphabets, numbers, mixed for only second person and triple training with triple test set and test of only second person.

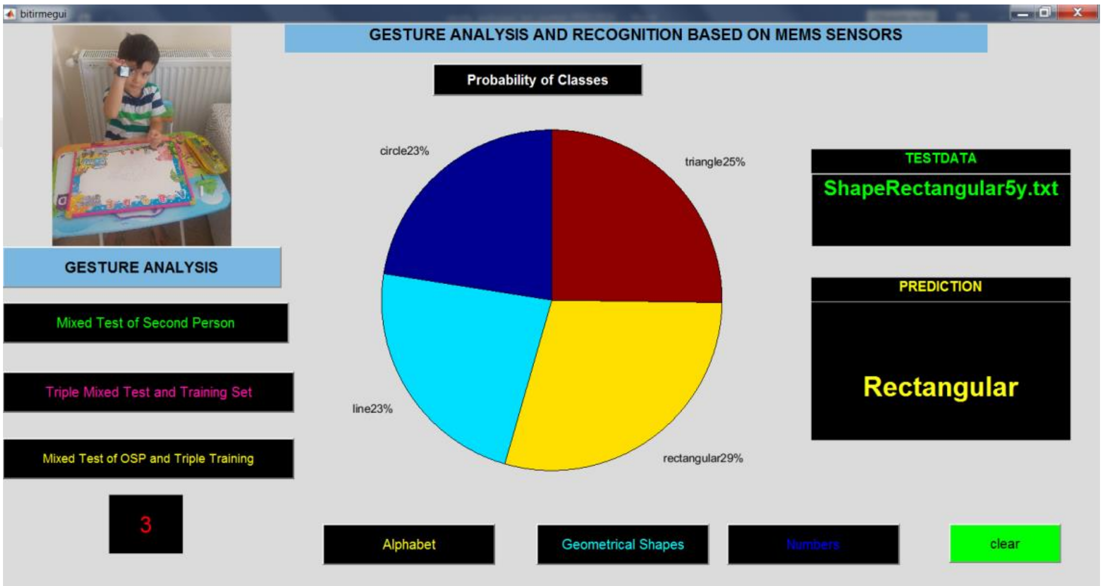

Figure A.1 Geometrical shapes

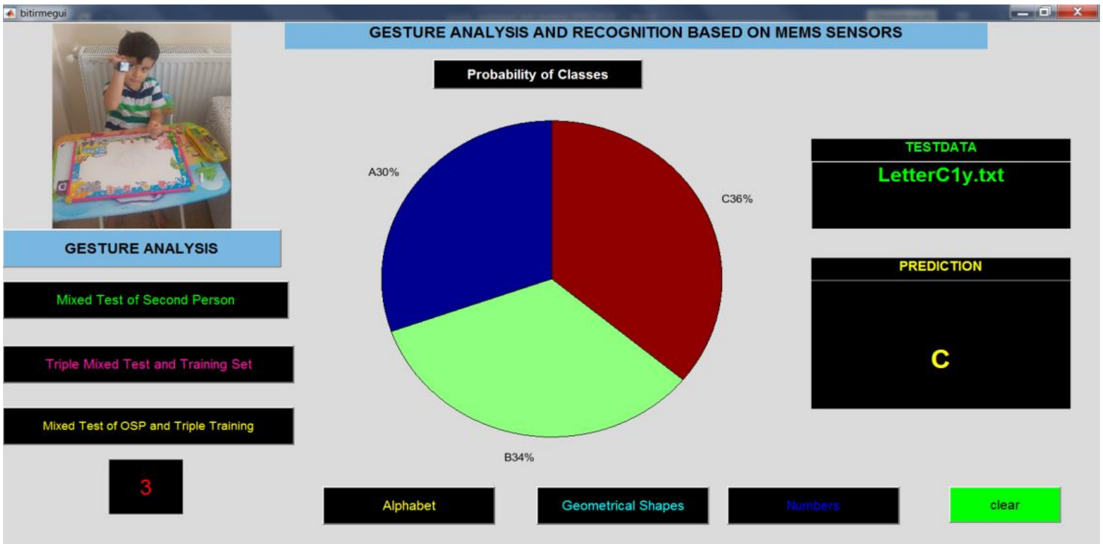

Figure A.2 Alphabet

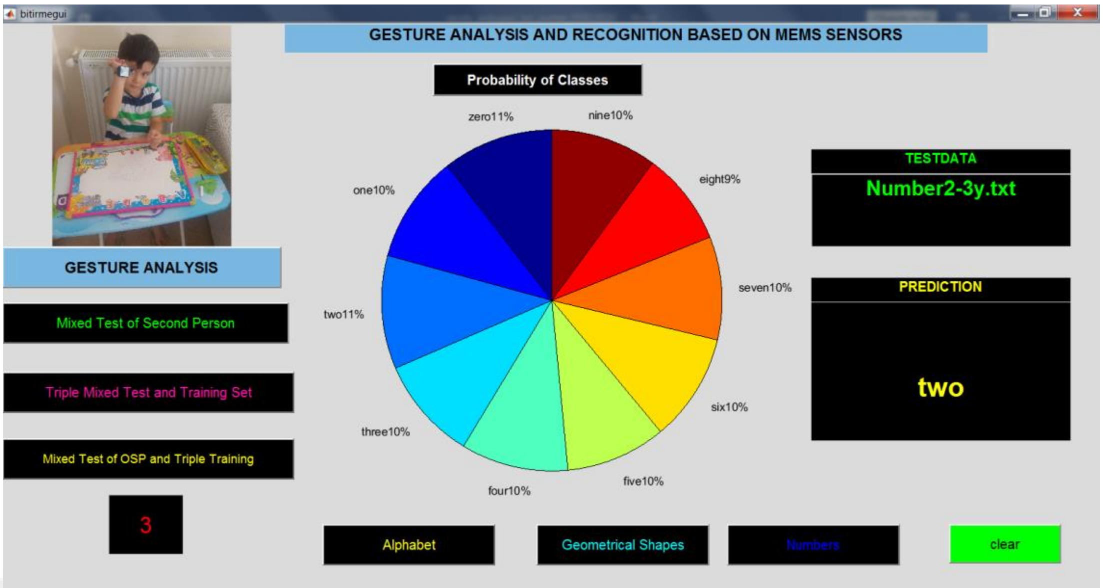

Figure A.3 Numbers

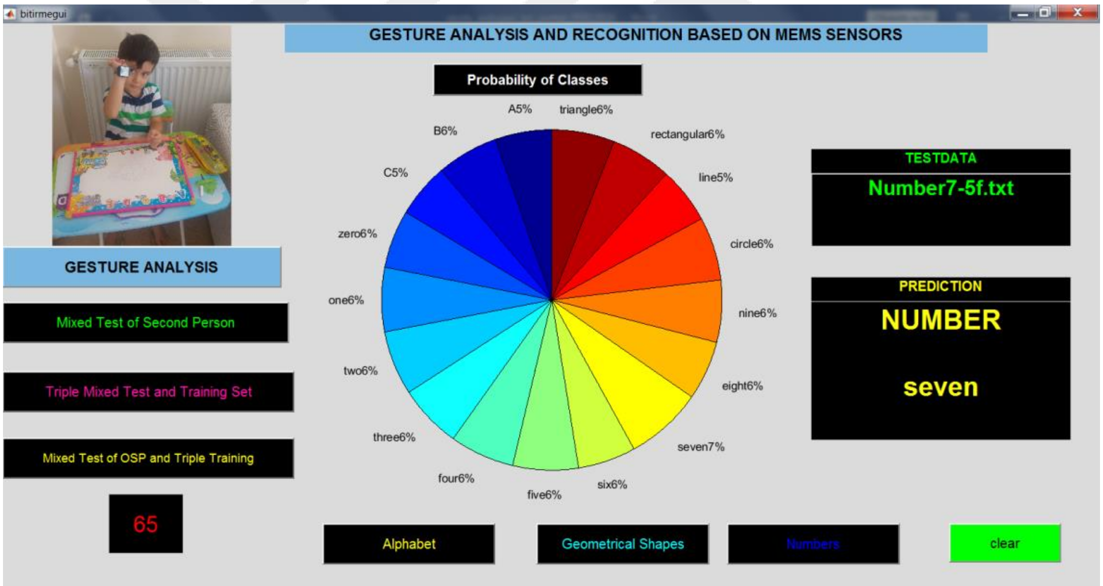

Figure A.4 Mixed training and test set of only second person

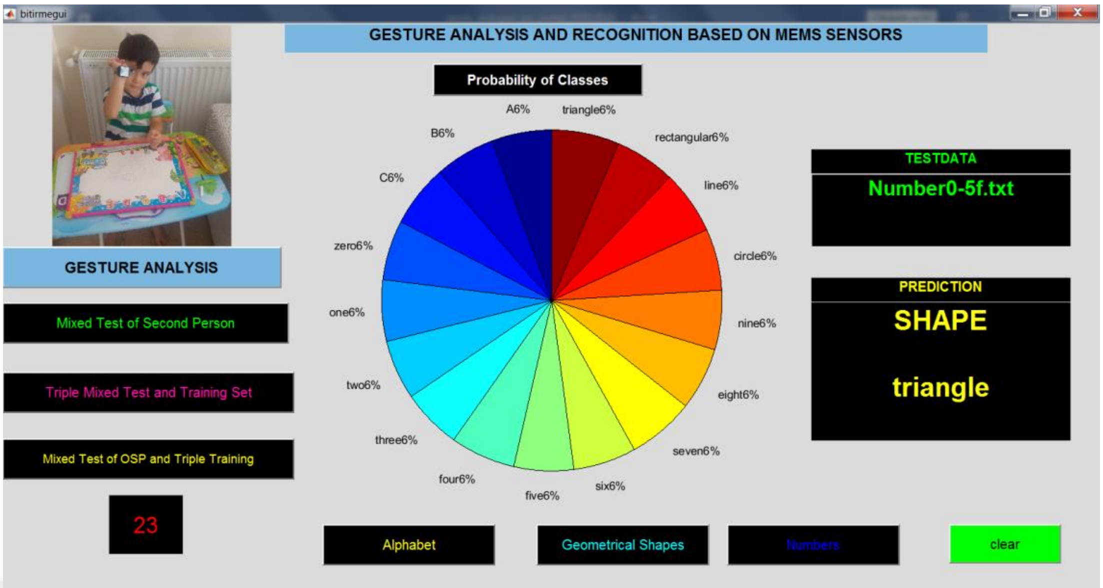

Figure A.5 Triple training set and test set of only second person

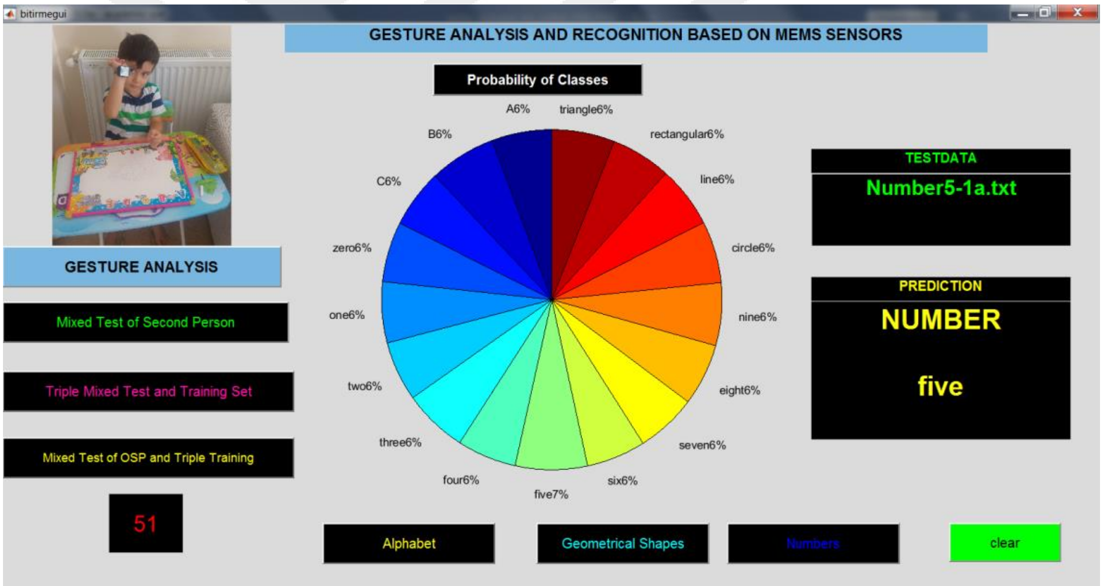

Figure A.6 Triple mixed test and training set

APPENDIX 2:

ABBREVIATIONS

ANN	Artificial Neural Networks
DTW	Dynamic Time Warping
GUI	Graphical User Interface
HCI	Human Computer Interaction
HMM	Hidden Markov Model
IMU	Inertial Measurement Unit
KNN	K Nearest Neighborhood
MEMS	Micro-Electro-Mechanical Systems
MPU	Motion Processing Unit
OSP	Only Second Person
PUI	Perceptual User Interfaces
RF	Random Forest
SVM	Support Vector Machine
USB	Universal Serial Bus
Ns	State number
No	Observation number