ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

GESTURE RECOGNITION FOR HUMANOID ROBOT ASSISTED INTERACTIVE SIGN LANGUAGE TUTORING

M.Sc. THESIS

Bekir Sıtkı ERTUĞRUL

Department of Computer Engineering

Computer Engineering Programme

SEPTEMBER 2014

ISTANBUL TECHNICAL UNIVERSITY ★ GRADUATE SCHOOL OF SCIENCE ENGINEERING AND TECHNOLOGY

GESTURE RECOGNITION FOR HUMANOID ROBOT ASSISTED INTERACTIVE SIGN LANGUAGE TUTORING

M.Sc. THESIS

Bekir Sıtkı ERTUĞRUL
(504111503)

Department of Computer Engineering

Computer Engineering Programme

Thesis Advisor: Asst. Prof. Dr. Hatice KÖSE

SEPTEMBER 2014

İSTANBUL TEKNİK ÜNİVERSİTESİ ★ FEN BİLİMLERİ ENSTİTÜSÜ

İNSANSI ROBOT DESTEKLİ ETKİLEŞİMLİ İŞARET DİLİ EĞİTİMİ İÇİN İŞARET TANIMA

YÜKSEK LİSANS TEZİ

Bekir Sıtkı ERTUĞRUL
(504111503)

Bilgisayar Mühendisliği Anabilim Dalı

Bilgisayar Mühendisliği Programı

Tez Danışmanı: Yrd. Doç. Dr. Hatice KÖSE

EYLÜL 2014

Bekir Sıtkı Ertuğrul, a **M.Sc.** student of ITU **Graduate School of Science Engineering and Technology** student ID **504111503**, successfully defended the **thesis** entitled "**GESTURE RECOGNITION FOR HUMANOID ROBOT ASSISTED INTERACTIVE SIGN LANGUAGE TUTORING**", which he prepared after fulfilling the requirements specified in the associated legislations, before the jury whose signatures are below.

Thesis Advisor :	**Asst. Prof. Dr. Hatice KÖSE** İstanbul Technical University	.....
Jury Members :	**Assoc.Prof.Dr. Hazım Kemal EKENEL** İstanbul Technical University	.....
	Assoc. Prof. Dr. Songül ALBAYRAK Yıldız Technical University	.....

Date of Submission : 05 May 2014
Date of Defense : 01 September 2014

To my family,

FOREWORD

I have been working on Human Robot Interaction topics since 2013 when I took a course related with Human Robot Interaction. Besides, I have also studied on Machine Learning, motion sequence recognition and image processing topics.

I will be always grateful for all that my advisor Asst. Prof. Hatice Köse have done to help me and thanks for her supports. I am so thankful to Ömer Sinan Saraç for the time he spent to help me with my project on Machine Learning topics.

I owe gratefulness to my family for all the years they have been by my side, supporting me and encouraging me.

I would also like to thank to my colleague Cemal Gürpınar, Mennan Güder, Hasan Er and Kemal Taşkıran, Enes Özbay and other friends for their technical support and motivation.

I would also like to express my appreciation to Assoc.Prof.Dr. Hazım Kemal Ekenel and Assoc. Prof. Dr. Songül Albayrak for kindly accepting to be my jury members.

September 2014

Bekir Sıtkı ERTUĞRUL
Computer Engineer

TABLE OF CONTENTS

Page

ABBREVIATIONS

ANN : Artificial Neural Network
HMM : Hidden Markov Model
DTW : Dynamic Time Warping
GMM : Gaussian Mixture Model
HRI : Human-Robot Interaction
SL : Sign Language
DOF : Degree of Freedom
TSL : Turkish Sign Language
ASD : Autism Spectrum Disorder
ASL : American Sign Language
GUI : Graphical User Interface
SDK : Software Development Kit
BoW : Bag of Words
BoVW : Bag of Visual Words

LIST OF TABLES

Page

LIST OF FIGURES

Page

GESTURE RECOGNITION FOR HUMANOID ROBOT ASSISTED INTERACTIVE SIGN LANGUAGE TUTORING

SUMMARY

In the recent years, depending on the rapid progress in the robotic technology, the researchers develop various applications on Human-Robot Interaction. At this point, besides the rapid progress in the robotic field, some topics like, how robots behave in the social life, what their limits are and how a robot communicates with people, become inevitable for researchers. Robots are in use with the people in the industry and many fields, including therapy, entertainment and education.

When the people communicate with each other, they need an interaction point. Usually this interaction point is verbal; but if the people cannot communicate verbally, gestures and sign language are in use. Especially hearing-impaired people use sign language including upper torso gestures and/or mimics to improve their expressions. Consequently, the sign language is an important part of interaction for people who have communication disorders including people with autism spectrum disorder (ASD).

In this work, it is intended to teach sign language to the children with imitation based turn taking games with the help of humanoid robots, which are significantly developed in the recent years. Hearing-impaired children and children with ASD are focused in this project.

In this study, a humanoid robot (NAO H25 or R3) helps the human teacher in teaching some signs and basic upper torso actions, which are observed and imitated by the participants. Along with the robots cameras, RGB-D sensor camera (Kinect) is used in the user tests. Due to its capability of capturing the human skeletal data and joint position information, the Kinect sensor is used instead of regular RGB cameras. Besides, it gives more motion freedom to the users than the wearable motion capture devices, and do not require additional markers, or special gloves.

The system developed within the scope of this thesis is focused on the gesture/sign recognition part of this work. With the use of Kinect, the upper torso information of the human participants is gathered and with using the proposed methods, it is recognized in real-time so the robot is able to give feedback to the participants about their actions. We propose three methods to recognize the signs/gestures namely; K-Means with Hidden Markov Model (HMM), Neural Networks (NN) with HMM and Real Time Gesture Recognition. All of the methods work base on the Hidden Markov Model.

HMM is the very typical model for a stochastic sequence of a finite number of observations. In this study, every sign or gesture is represented with a sequence of frames. Each gesture/sign is composed of *n* states that are based on these frames. However, these states are hidden, not directly observable. The representation of frames as a sequence of observations is an issue in this solution. To overcome this

issue, other heuristics such as the kinematics, k-means and ANN based methods are proposed within the thesis.

In the first method that is named as K-Means with HMM, frames are transformed into observation sequence by means of k-means algorithm. In the second method (ANN with HMM), generation of observation sequence is handled by a neural network. Therefore, HMM takes the emission probabilities from this ANN module. In the last method the DTW is added to the ANN with HMM method and result of DTW and ANN with HMM methods is combined.

In the first method, Baumwelch training algorithm is used for HMM training and it is not cost effective. The training time of second method is better than the first method. Nevertheless, in the test step, the first method is faster than the second method. In terms of effectiveness, if the data for NN is labeled correctly, and an optimum parameter set is selected (hidden layer count and node counts per layers), the second method has higher success rates than the first approach. The results is empowered with the adding of third method to the system.

The system is used in the recognition phase of the above-summarized user cases with children and the results are discussed and published in the related conferences and publications.

İNSANSI ROBOT YARDIMLI İŞARET DİLİ ÖĞRETİMİ İÇİN İŞARET DİLİ TANIMA

ÖZET

Son yıllarda robot teknolojilerindeki hızlı ilerlemeye bağlı olarak, robotların sosyal hayatta kullanılması amacıyla çesitli uygulamalar geliştirilmeye çalışılmaktadır. Bu noktada robot teknolojilerindeki ilerlemenin yanında robotların sosyal yaşamda nasıl kullanılmaları gerektiği, sınırları, robotların insanlar ile etkileşimlerinin nasıl olması gerektiği konularında araştırmalar yapılması kaçınılmaz olmuştur. Robotlar başta endüstriyel alan olmak üzere hayatın çeşitli alan ve seviyelerinde insanlarla iç içe olmaya başlamışlardır. Robotların insanlarla bu tarz etkileşimlerini belirlemek amacıyla insan robot etkileşimi araştırma alanı olarak ortaya çıkmıştır.

İnsanlar birbirleriyle karşılıklı iletişim kurmak istedikleri zaman bir etkileşim noktasına ihtiyaç duyarlar. Bu etkileşim noktası genellikle sözlü olmakla beraber, sözlü iletişim kuramayan insanlarda iletişim işaret diliyle gerçekleşir. Sözlü olarak iletişim kurabilen insanlar bile ifadelerini güçlendirmek için jest ve mimiklerini kullanırlar. Dolayısıyla işaret dili, etkileşimin önemli bir parçasıdır.

Bu çalışmada son yıllarda gelişen robot teknolojisinin ve insansı robotların yardımıyla, işaret dilinin taklit bazlı ve robotların insanlarla sıralı olarak etkileşmesiyle öğretilmesi amaçlanmıştır. Projenin asıl amacı, iletişim sorunları yaşayan çocukların işaret dilini öğrenerek iletişim bozukluğu olmayan insanlarla etkileşimlerini artırabilmeleridir.

Projenin hedef topluluğu işitme engelli çocuklar ve Otizm Spektrum Bozukluğu (ASD) olan çocuklardır. ASD kısıtlı sosyal etkileşim sorunları, iletişim sorunları ve kısıtlı hayal gücü gibi sorunları kapsar. ASD'li bir çocuk öğretmeniyle veya diğer insanlarla iletişim kurmakta güçlük çeker. Fakat yapılan denemelerde çocukların robotlara diğer insanlardan daha fazla ilgi gösterdiği görülmüştür. Bu nedenle robotların, çocuklar ve öğretmenleri/ebeveynleri arasında sosyal etkileşim kurma konusunda destek olarak kullanılabileceği düşünülmüştür.

Çalışma kapsamında önceden belirlenen bazı işaret dili hareketlerinin ve basit üst vücut hareketlerinin, izleme ve taklit yoluyla çocuklara öğretilmesi hususunda öğretmenlere NAO H25 isimli insansı robot kullanılarak yardımcı olunmaktadır. Nao robotun insansı ve oyuncak gibi görünmesi çocukların daha fazla ilgisini çekmesine sebep olmuştur. Nao robot üzerindeki kameralar ile birlikte Microsoft firmasının Kinect isimli RGBD kamerası da proje kapsamında kullanılmıştır. Kinect algılayıcısı, insan iskeleti yapısını ve eklem pozisyon bilgilerini alabilme kabiliyeti, kullanım kolaylığı, kullanıcıya hareket serbestliği sağlaması gibi yetenenekleri nedeniyle kullanılmıştır.

Tez kapsamında yapılan sistem bir oyun üzerine kurgulanmıştır. Bu oyunda öğretilecek her hareket için bir oyun kartı oluşturulmuştur. Öğretici, herhangi bir kartı seçerek robota gösterir, robot kartı tanır ve o kartla ilişkilendirilen hareketi yapar. Hareketi tamamladıktan sonra sırayı karşısındaki çocuğa bırakır. Robot hareketi yapacak kişinin hareketlerini takip eder ve hareketi doğru yapıp yapmadığını anlamaya çalışır. Eğer katılımcı hareketi doğru yaptıysa görsel ve sesli olarak hareketi doğru yaptığına dair geri bildirimde bulunur. Eğer hareteket katılımcı

tarafından yanlış yapıldıysa, katılımcıya hareketi yanlış yaptığına dair görsel ve sesli uyarıda bulunularak, katılımcının hareketi tekrar denemesi istenir.

Bu oyunun gerçekleştirilmesinde iki önemli adım vardır. Birinci adım oyun kartının tanıınması, ikinci adım ise işaret dilinin tanınmasıdır. Oyun kartlarının tanınmasında BoVW(Bag of visual words) denilen bir yöntem kullanılmıştır. Bu yöntemde kartların tanımlayıcı özellikleri çıkarılarak, bu tanımlayıcı özellikler K-Means denilen bir algoritma vasıtasıyla gruplanır. Bu grupların merkezleri sistemdeki tüm görsel sözlüğümüzü oluşturur. Daha sonra gelen bir resmin tanımlayıcı özellikleri çıkarılarak bu sözlükteki kelimelere göre histogramı oluşturulur ve histogram karşılaştırma yöntemi kullanılarak kartlar tanınır.

Tez kapsamında asıl üzerinde durulan konu ise işaret dili tanımadır. Bu kapsamda üç yaklaşım denenmiştir. Birinci metotta K-means ile beraber Saklı Markov Modeli(SMM) kullanılmış daha sonra ise Yapay Sinir Ağı (YSA) ile beraber Saklı Markov Modeli kullanılmıştır. Son olarak dinamik zamak bükmesi(DTW) algoritması sisteme eklenerek sonuçların iyileşmesi sağlanmıştır. Kullanılan üç metotta da temel olarak Saklı Markov Modeli önemli yer tutar. SMM sonlu sayıda stokastik sıralı gözlem kümesi üzerinde çok kullanılan bir yöntemdir. Projede işaret dili hareketleri Kinect'ten gelen sıralı veriler ile ifade edilmiştir. Her işaret dili hareketinin **n** adet durumdan oluşmaktadır. Örneğin kolları yana doğru açma hareketi (side) 'nin 3 durumdan oluştuğu varsayılmaktadır. Bu durumlar; kolların vücuda yapışık olduğu durumdan, omuzların vücut ile 45 derecelik açı yaptığı durum ve kolların yere paralel olduğu durumlardır. Eğer kollar, bahsedilen 3 durumda sıralı olarak görülebilirse kişinin bu hareketi doğru olarak yaptığı söylenilebilir. Bu durumda hareketin durumlarının iyi belirlenmesi ve bu durumların zaman sıralı olarak doğru gelmesi gerekmektedir.

Fakat bu durumlar doğrudan gözlemlenemez. Gelen verilere çeşitli işlemler uygulanarak anlık pozisyonlar gözlem dizisine çevrilir. Çalışmada veri olarak Kinect'ten elde edilen bir gözlem dizisi kullanılmakta ve bu gözlem dizisine bakarak sistemin istenilen durumlardan sıralı olarak geçmiş olma olasılığı hesaplanıp buna görede reaksiyon verilmektedir.

Bu noktada bazı problemler ile karşılaşılmaktadır. Mesela Kinectten alınan veriler derinlik bilgilerini içeren vektörlerden oluşmaktadır. Bu vektörlerin yorumlanarak SMM'in anlayacağı gözlem dizisine ve parametrelerine çevrilmesi gerekmektedir.Sorunların çözümü için kinematic, K-means, DTW ve YSA'dan oluşan metotlar kullanılmıştır.

Yukarıda bahsedilen üç yaklaşımda da sensörden uzamsal verilerin alınarak açı verilerine çevrilmesi ve sistem kapsamında kullanılacak olanların değerlendirilmesi aynı yolla yapılmıştır. Fakat bu sonuçların SMM'nin anlayacağı parametreleri dönüştürme işleminde farklılıklar vardır.

Birinci yaklaşımda sisteme gelen sensör verilerinin değerlendirilerek gözlem dizisine dönüştürülme işlemi K-means ile yapılmıştır. İkinci yaklaşımda ise SMM'nin ihtiyaç duyduğu gözlem dizisinin olasılıklarını hesaplama işlemi YSA modelleri ile yapılmaktadır. Son yaklaşımda iki yöntemin birleşmesinden oluşmaktadır. Birinci yöntemde SMM'nin ihtiyaç duyduğu gözlem dizisinin olasılıklarını hesaplama işlemi yine YSA modelleri ile yapılmaktadır. İkinci metotta ise DTW kullanılmatadır. DTW içingözlem dizisine çevirme işlemi gerekmemektedir. DTW'ye açı değerlerine çevrilen uzamsal koordinatlar doğrudan gönderilmektedir.

Birinci yaklaşımda, Baumwelch isimli bir algoritma ile SMM eğitimi yapılmıştır. Bu algoritma zaman bakımından ikinci yaklaşıma gore çok masraflıdır. Bu da eğitim zamanının çok artmasına neden olmaktadır. Fakat bu algoritma sonucunda oluşan gözlem olasılıkları matrisi sayesinde test zamanında ikinci yaklaşıma gore üstünlük sağlamaktadır. Bunun sebebi ise ikinci yaklaşımda SMM eğitiminin olmamasıdır. SMM başlangıç ve geçiş matrisleri sisteme doğrudan verilir. Bu yaklaşımda sadece YSA eğitilir ve SMM, gözlem olasılığı istediğinde matristen değil YSA'dan hesaplama istenerek, buradan sağlanmaktadır.

Sonuç olarak ikinci yaklaşımda etiketleme işleme doğru yapılarak, en uygun parametreler ile eğitilebilirse ikinci yaklaşımın birinci yaklaşımdan daha başarılı olduğu görülmüştür.

Üçüncü yaklaşımda ise ikinci yaklaşıma ek olarak DTW eklenmiştir. Bu yaklaşımda, DTW açı değerlerini doğrudan alarak her hareketin kendi DTW modeline gönderir ve her hareket modeli gelen açı dizisinin o hareket olup olmadığını gösteren bir olasılık değeri hesaplar. Diğer taraftan eş zamanlı olarak ikinci yaklaşımdaki metod yardımıyla, her hareket modeline(SMM'li YSA modeli) gelen açı dizisi gönderilir. Her hareket modeli(SMM'li) gelen açı dizisinin o hareket olup olmadığını gösteren bir olasılık değeri hesaplar. Bu iki algoritmadan dönen olasılık değerleri, eğitim verisinden elde edilen ağırlık değerleriyle ağırlıklandırma yapılarak birleştirilir ve sonuç elde edilir. Sistemin performansı DTW ve ikinci yaklaşımın birleştirilmesiyle daha da artmıştır.

1. INTRODUCTION

Communication is a vital requirement for human life. Language acquisition is an extremely crucial process for brain development and intelligence. Sign Language (SL) is an alternative way of communication for hearing impaired or autistic children who cannot communicate verbally. Sign Language is a visual language that is based on upper body movements (including hands, fingers, arms, upper torso, head and neck) and facial gestures. There are studies on visual recognition of sign language, and sign language tutoring with 2-D visual aids to hearing-impaired people [1-6]. In addition, several robots and robotic hands were utilized to implement sign language [7-8]. In the studies [9-11] visual games are employed for sign language tutoring.

The studies introduced in this thesis have been realized as part of an on-going project "Robot Sign Language Tutor", which is supported by the Scientific and Technological Research Council of Turkey under the contract TUBITAK KARIYER 111E283. The project aims to utilize humanoid robots for assisting sign language tutoring due the lack of sufficient educational material. Also in terms of children's sign language education, 2-D instructional tools are found to be incompetent. Therefore using humanoid robots as an assistive tool in sign language tutoring for children will be very beneficial. In the proposed system, it is intended that a child-sized humanoid robot is going to perform and recognize various elementary signs (currently basic upper torso gestures and words from sign languages) so as to assist teaching these signs to children with communication problems. This will be achieved through interaction games based on non-verbal communication, turn taking and imitation that are designed specifically for robot and child to play together. Imitation based non-verbal interaction games with humanoid robots were successfully used with children and adults previously within the project [12-16].

The robot should be aware of what the children are doing in front of it. To ensure this matter, the best way is that children's movements are perceived by the robot via gesture recognition. Moreover, for children not becoming bored, the communication between the robot and the child should be fast. Thus, the robot should sense the

child’s gesture and give feedback to child in a short time. Our main motivation comes out from this point, which is recognizing the gestures quickly and accurately in real time. So, there are two important parameters in our work which are speed and accuracy.

In order to enhance these parameters, we propose some methods. One of the methods is spatial data transformation. Each child has different length of upper torso. According to the experiments in different studies, this matter leads inaccurate results. To increase the accuracy in recognition process, we transform the spatial values of joint points coming from Kinect to angle values to make the data person-independent. To optimize the recognition speed, we applied three methods which are K-Means with HMM, ANN with HMM, DTW and ANN with HMM respectively. We gain maximum increase in recognition speed in the last method.

As a summary, we made two main contributions in this project. One of them is using angle values of the joint points of the human body. The other one is creating a new method to recognize the sign language gesture by combining the Dynamic Time Warping, Artificial Neural Network with Hidden Markov Model to increase the recognition accuracy.

2. RELATED WORK AND ALGORITHMS

2.1 Literature Review

As a result of the rapid progress on new technologies, various methods are developed for human machine interaction. Gesture recognition has been very active research area from early of 90's for human machine interaction.

Gesture recognition is tracking and interpreting the human hand, arm, face and head movements. To fulfill gesture recognition, some mathematical and statistical methods should be utilized. Hidden Markov Model, Random Forest Methods, Dynamic Time Warping Method, Neural Network Methods and some other statistical methods are promising tools that could be employed.

Current literature is concentrated on different applications to make interactions more easy, natural and convenient without help of any extra device.

Indeed, interaction is handled through human body (hand, arm etc.) then it evolves to the conventional devices like mouse and keyboard [17]. The data, which is obtained from human with different ways, can be much more than the data taken from conventional devices. So the conventional device based interaction can be improved.

There are two main gesture recognition approaches in the literature; device based and vision based. At the beginning, some devices were used to capture the data thus recognition process was called as device based gesture recognition. These recognition systems were based on accelerometers (Soapbox, Wii mote, PlayStation Move has three-axis accelerator), optical/magnetic tracking and on glove (Data glove) based equipment [17] [18].

Other alternative is vision based methods [19]. These methods use camera for inputs and extract features from them. These methods is neutral way to obtain input and have some advantages compared to the device based approaches like human who has not wear any device, just use his/her body. However, they have some disadvantages. The vision base information depends on environment, placement of camera,

luminance effects, human posture change, obstacles etc. In the current vision based application, Kinect is commonly employed [20] [21] [22].

The main methods in gesture recognition are DTW [23], Hidden Markov Model [24] [25] [26], Kalman Filtering and HMM [27], Discrete Cosine Transform and K-NN (K-nearest neighbor) [22]. The related literature is given in the following paragraphs:

Rung-Hui et.al [28] tried to recognize 250 vocabularies Taiwanese Sign Language (TWL) using HMM. In their model they used DataGlove to obtain data and they interpreted the real-time continuous data [28]. They separated the gesture to four parameters, which were posture, position, orientation and motion. Their sentence success rate was 80.4% for 51 postures, 6 orientations and 8 motions. This success rate was dependent to signer.

In the early of 90's, 6 different tennis strokes were recognized by Yamato et.al [29]. They used 25x25 pixels image. Their recognition methodologies were Discrete HMM.

Then in the same years Thad Starner and Alex Pentland [30] used the HMM to recognize American Sign Language. They used colored gloves. For the 40 words word recognition success rate is 99%.

Device based approach is used by Aran and Akarun [27]. They tried to recognize seven signs in TSL so they designed an interactive sign language game. According to the game, a test subject who wears the gloves (device base recognition) was selecting a sign from the program then a video was played to show signs to the test subject. After the video completion, the program was waiting for implementation of gesture from the test subject. They used Kalman Filtering and HMM for recognition and their success rate is 95%. In this work, they specified their success rate as 76% for 19 signs from American Sign Language.

With the releasing of Kinect SDK (2011) some recognition methods based on Kinect data were released. In the one of the works, Zafrulla and et.al [21] used the Kinect depth camera's data. They tried to recognize 1000 ASL, their success rate was 51.5%, and sentence verification rate is 76.12%.

Kinect based RGB and depth data was used in the Memis and Albayrak's works [22]. They used motion difference and accumulation approach for temporal gesture analysis. They tried to recognize 1002 signs which belong to 111 words from

Turkish Sign Language. For recognition processes, Discrete Cosine Transform (DCT) and K-Nearist Neighbor classifier were chosen for this work and their success rate was 90%.

There is a lot of use of DTW and HMM distinctly for recognition but it is not obvious which one is better for gesture recognition. Carmona and Climent [31] compared these two methods with using different criteria. They created a dataset, which included 75 samples of ten different gestures corresponding to the 0-9 numbers. Moreover, they applied DTW and HMM then, they compared according to recognition rates, computing times and some other criteria. In their test, DTW gave better success rate than HMM. DTW gave 98.8% success rate and HMM gave 96.46% success rate with optimal parameters. To obtain similar recognition rate, HMM was required more training sample than DTW, because of DTW needs just one reference sequence. HMM had lower recognition time than DTW in their tests.

2.2 K-Means

2.2.1 Description

K-Means clustering algorithm is an unsupervised classification algorithm. Originally, it is a method of vector quantization, which divides large number of points into the number of clusters given by the user. Nowadays this algorithm is popular on machine learning, pattern recognition, and data mining.

Since k-means is an unsupervised classification algorithm; the original points have no labels, and at the beginning only data points and *k* value, which is the desired cluster count, are given.

Let's assume that **d** dimensional **n** data points are given. Our data points vector is $X=\{X_1, X_2, \ldots, X_n\}$ and each point(X_i) is in the data vector(X) is a d-dimensional vector like $X_i =\{X_i^1, X_i^2,\ldots, X_i^d\}$. And a **k** ($k<n$) value is given. **N** points will be partitioned into **k** sets as $\mathbf{S= \{S_1, S_2, \ldots, S_k\}}$ [32] then each $\mathbf{S_i}$ is called a **cluster** and these clusters include the set of data points that are assigned to the cluster with the nearest mean. And the centroids are given like $\mu= \{\mu_1, \mu_2,\ldots, \mu_k\}$ that $\boldsymbol{\mu_i}$ is **the mean of points** in $\mathbf{S_i}$ and $\boldsymbol{\mu_i}$ called as **centroid of cluster** $\mathbf{S_i}$.

2.2.2 How it works

K-means has an iterative structure to find local optimum point. K-means algorithm applies two steps to the data points at every iteration. These steps are assignment and update steps as in the expectation maximization algorithm. The assignment step is the same with the expectation step and the update step is the same with the maximization step [33]. The main idea of the k-means iterations is minimizing the within-cluster sum of squares:

$$\arg\min_{S} = \sum_{i=1}^{k} \sum_{x_j \in S_i} \left\| X_j - \mu_i \right\|^2 \quad [33] \qquad \textbf{(2.1)}$$

where $\boldsymbol{\mu_i}$ is the mean of points in $\mathbf{S_i}$ [33].

The k-means algorithm can be summarized with four steps which are mentioned below;

1- Select the initial centroids (μ_k values) randomly and assign the initial label of data points according to the closeness to the centroids.
2- Calculate the new centroids according to the new labels.
3- Assign the label of data points according to the closeness to new centroids.
4- Loop until the no assignment changes in the partition or specified number of iteration count.

At the step 1 mean vector, have to be initialized and initial label of data points have to be assigned. The initial values of mean vectors are chosen arbitrary. It can be assigned random values, or it can be picked k points from the data points vector(X) and uses as the initial means. Then the data points are assigned to closest means. As well as there are some assignment methodologies like Euclidian distance, Manhattan distance, Chebyshev distance measurement, generally Euclidian distance is used for these assignment [34]**.** The Euclidian distance calculation of any point(X_i) from data points to any centroid(μ_k) shown in equation **(2.2)**;

$$dist(X_i, \mu_k) = \sqrt{\sum_{a=1}^{d} (X_i^a - \mu_k^a)^2} \quad \text{(d dimension of a point)} \qquad \textbf{(2.2)}$$

The step 1 operations illustrated in Figure 2.1 for 13 data points (n=13) and three cluster (k=3). In the figure, the randomly picked centroids are demonstrated with "X" mark and cluster borders shown with dashed line.

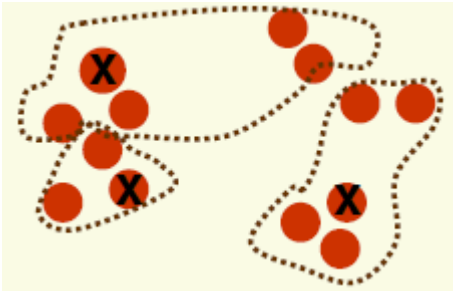

Figure 2.1 : Initial centroids and clusters [34].

After then the initial means are created as in shown in Figure 2.1 data points are assigned into the clusters. The initial cluster assignment to the points is same with the assignment step, which will explain at the step 3.

Then the turn comes to calculation of the new centroids. In the second step, the centroids of the clusters will updated according to the new assignment of the clusters. A cluster Si has set of data points, which are assigned to it. Sum of all points in this cluster (Summation of each dimension will be separately) divided by total number of points which belongs to this cluster then new centroids of clusters are found. This is shown with formulas (2.3).

$$\mu_i = \frac{1}{|S_i|} \sum_{X_j \in S_i} X_j \qquad \textbf{(2.3)}$$

As shown in Figure 2.2 the sum of all points in dashed line divided by total number of red points in the same dashed line then the new clusters, which are shown with "X" mark are found.

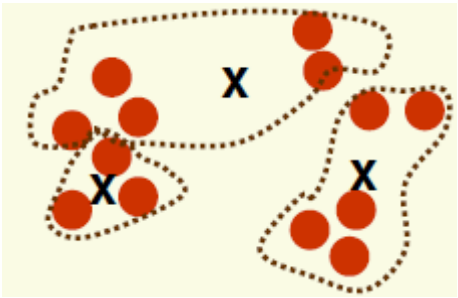

Figure 2.2 : The clusters updated after assignment [34].

At the step, 3 points are assigned to the clusters. Sequentially a data point from dataset is picked then the distances to all clusters are calculated then the point is

assigned into the new cluster with the minimum distance. This operation is shown in formulas (2.4).

$$\arg\min_{S_i} = \|X_j - \mu_i\|^2 \quad \text{Or} \quad \arg\min_{S_i} = dist(X_j, \mu_i) \tag{2.4}$$

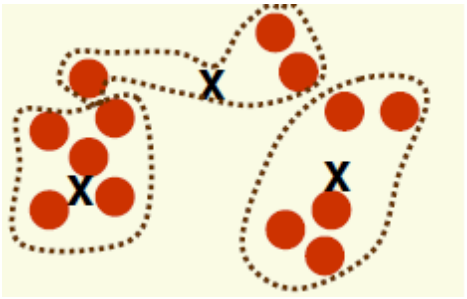

Figure 2.3 : Reassignment all sample to closest centroid [34].

In the Figure 2.3, the new assignment of red point shown. Before these points are in the different dashed line (cluster), this is illustrated in Figure 2.2 but after the step 3 the clusters was like the Figure 2.3.

After the step 3, clusters are changed so the new centroids have to be calculated then algorithm goes back to step two then iterate repeatedly. This iteration continues until the no assignment changes in the partition or specified number of iteration count.

2.3 Markov Model and Hidden Markov Model

2.3.1 Introduction

Many machine learning techniques assume that the data points are independent and identically distributed. This means that the likelihood function is derived from the products of the likelihoods of the individual instances [35]. Speech recognition, handwriting recognition, word letters recognition, bioinformatics (DNA sequence recognition), musical score following, and similar problems cannot be solved by the basic probability distribution methods because the successive instances are dependent. For example, the probability of place of ‘h’ letter after ‘t’ letter more probable than ‘x’ letter after ‘t’ letter [36]. This situation comes with the sequence of observation, which each observation depends on before. In probability theory, these kinds of problems are called as stochastic process.

"*In probability theory, a stochastic process, or sometimes random process (widely used) is a collection of random values; this is often used to represent the evolution of some random variable, or system, over time. Instead of describing a process which can only evolve in one way (as in the case, for example, of solutions of an ordinary differential equation), in a stochastic or random process there is some indeterminacy: even if the initial condition (or starting point) is known, there are several (often infinitely many) directions in which the process may evolve.*"[**37**]

In this kind of problem, Markov Model and Hidden Markov Model can be used to predict future instance from previous sequence.

2.3.2 Markov model

Consider a system comprising of N distinct state (S_1, S_2, S_3,…, S_N) at any time **t** and each state is dependent on most recent state except the previous states. The $\mathbf{q_t}$ stands for the actual state at time t (t=1, 2, ...) and the $\mathbf{q_t = S_i}$ represents that the system is in state S_i at time **t**. Rabiner [38] specifies the being on state **j** at time t+1($q_{t+1} = S_j$) depends on previous state transitions and it is shown like

$$P(q_{t+1} = S_j | q_t = S_i, q_{t-1} = S_k, q_{t-2} = S_l, \dots) \quad \textbf{(2.5)}$$

and as a special case, first order Markov chain, is being on state j at time t+1 ($q_{t+1} = S_j$) depends to only most recent state except the previous states(previous times):

$$P\left(q_{t+1} = S_j \middle| q_t = Si, q_{t-1} = Sk, q_{t-2} = Sl, \dots\right) = P\left(q_{t+1} = S_j \middle| q_t = Si\right) \quad \textbf{(2.6)}$$

At any time the system moves with probability a_{ij} from state S_i to state S_j and this is demonstrated as:

$$a_{ij} = P\left(q_{t+1} = S_j \middle| q_t = Si\right) \quad \textbf{(2.7)}$$

Here is a_{ij} is independent from time, and the state occurrence at any time and a_{ij} must be

$$a_{ij} \geq 0 \; and \; \sum_{j=1}^{N} a_{ij} = 1 \quad \textbf{(2.8)}$$

In the Markov model **A** is a matrix called transition matrix, which is used for showing state to state transition probabilities (a_{ij} values). As it is pointed out above, the system has N different state so $A = [a_{ij}]$ is an N x N matrix and its rows summation has to be **one**.

Markov model holds one more term known as **start probability**. At t=1 in the time sequence, the system has no transition so it has to start from the initial point to calculate following probabilities. The start probability is represented as π_i and it stands for the initial probability of stateS_i, which is the first state in the sequence.

$$\pi_i \equiv P(q_1 = S_i) \tag{2.9}$$

$$\sum_{i=1}^{N} \pi_i = 1 \tag{2.10}$$

The vector $\prod = [\pi_i]$ has N elements and the summation of elements has to be one [36].

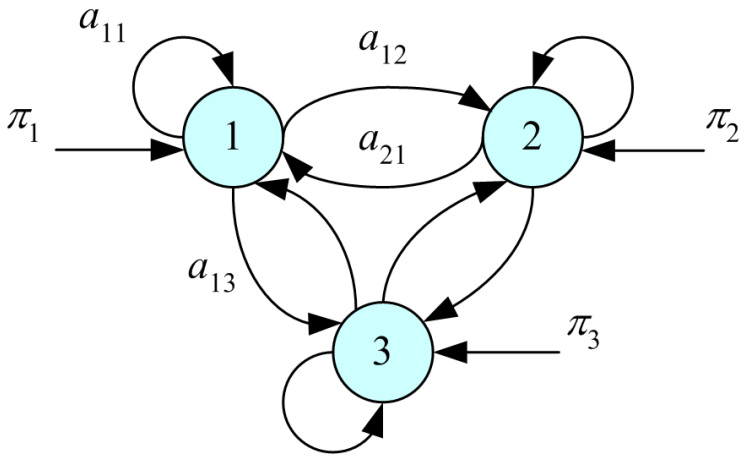

Figure 2.4 : Markov model illustration [36].

In the Markov model, states are visible to the observer. Any watcher can observe the system with the sequence O=Q={q_1,q_2,q_3,…., q_t}. With this information, the probability of any observation sequence can be calculated with the equation below.

$$P(O = Q|A, \Pi) = P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) = \pi_{q1} a_{q_1 q_2} \dots a_{q_{t-1} q_t} \tag{2.11}$$

Alpaydin [36] clarifies urns and balls example to demonstrate this problem. According to this example, there is **N** urn in a room and each urn holds red balls, green balls, or blue balls. There is a man who picks a ball among urns and shows its color. In this case, states are become S_1=red, S_2=blue, S_3=green and q_t indicates the drawn color at time t.

Assume that state's initial probabilities are

$$\Pi = [0.5 \quad 0.2 \quad 0.3]^T$$

In addition, the transition matrix is

$$A = \begin{bmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{bmatrix}$$

As it is also understood from transition matrix that a_{ij} is the probability of transition, which is drawing color i ball then drawing color j from any urn. After that given **Π**, **A** and an observation sequence like red, blue, blue, green with corresponding states O={S_1, S_2, S_2, S_3} , the probability of this observation can be calculated as follows:

$$P(O|A,\Pi) = P(S_1).P(S_2|S_1).P(S_2|S_2).P(S_3|S_2)$$

$$P(O|A,\Pi) = \pi_1.a_{12}.a_{22}.a_{23}$$

$$P(O|A,\Pi) = 0.5 \cdot 0.3 \cdot 0.6 \cdot 0.2 = 0.018$$

2.3.3 Hidden markov model

Suppose you are a company owner and you want to organize a campaign according to the location of people. You have only profile pictures whose backgrounds are cleaned, taken from a social media site for every day. People, who are in this dataset, have worn three kinds of dress and they can be at three types of locations. They wear suit, tracksuit or jeans and they have been at home, work or outside. You have some pictures for different sequential day for a person, and you can see only him/her dress in the pictures and you want to guess its locations in corresponding times. According to this problem, a person's location will be called a state in the concept of HMM. Besides, he/she can wear any of three dresses with different probability at each location and these dresses will be called an observation in this concept.

States are not observable in HMM, only observations are observable at each state with a certain probability. In this example, observations will occur with different probabilities at each state and will be discrete. They will be shown with {v_1, v_2, …. v_m}. The observation or emission probability in the state S_j is shown with b_j (m) [36]. In the example, b_{work} (suit) represents the probability of observing a person with suit at the work.

$$b_j(m) \equiv P(O_t = v_m | q_t = S_j) \quad [36] \qquad \textbf{(2.12)}$$

All observations (suit, tracksuit and jeans) can be observed at any state (home, work or outside) with different probability. HMM observation sequence is represented with ***O*** and corresponding state sequence is represented with ***Q*** but the state sequence is hidden that the observer cannot observe, it is inferred from observation sequence [36]. In the example, the observer can observe the *O* = {jeans, jeans, suit, tracksuit} and the system infers more than one state sequence (*Q*) with different probabilities but the highest probability of generated sequence will be taken into consideration.

For abbreviations of states, "**h**" for home, "**w**" for work and "**o**" for outside will be used. "s" for suit, "t" for tracksuit and "j" for jeans will be used for abbreviations of observations.

HMM consists of three parameters which are *A*, *B*, Π and an HMM model is shown like λ= (*A*, *B*, Π). Let's see the means of parameters:

1. N is the system state count, states of the system is denoted as

$$S = \{S_1, S_2, \dots, S_N\} \;\; in\ the\ example\ S = \{h, w, o\}$$

2. M is the observation count, normally each state can generate own observations, it can be independent from other state. In this case vector V have to include all of the observations that they are produced by each state.

$$V = \{v_1, v_2, \dots, v_M\} \;\; in\ the\ example\ V = \{s, t, j\}$$

3. "*A*" is the state transition matrix. It is same within the discrete Markov Model; it holds the transition probabilities from previous state to next state.

$$A = [a_{ij}] \ and \ a_{ij} \equiv P(q_{t+1} = S_j | q_t = S_i) \ [36]$$

4. "*B*" is the observation probability matrix. It holds the probability of observing m when the system is in the state j.

$$B = [b_j(m)] \ where \ b_j(m) \ \equiv P(O_t = v_m | q_t = S_j)$$

5. "Π" is the initial start probability of a state. It is same with the discrete Markov Model

$$\Pi = [\pi_i] \ and \ \pi_i \equiv P(q_1 = S_i)$$

2.3.3.1 Three main questions of hidden markov model

Assume an observation sequence $O = \{O_1 O_2 \ldots O_T\}$ is given. Three major problems can come out as stated below:

1. Given the HMM model parameters λ= (*A, B*, Π) and a sequence of observations *O*, we would like to find the probability of the observed sequence. P (*O*| λ) stands for the probability of observation sequence and this calculation will be done with forward or backward algorithm, which we get back later.

2. HMM model parameters λ= (*A, B*, Π) are known, observation sequence *O* is known and we would like to find most likely sequence of states $Q = \{q_1 q_2 \ldots q_T\}$ which generates the *O*. The highest probability of sequence *Q** which maximizes the P (*Q*|*O*, λ), is calculated with the Viterbi algorithm and it will be mentioned in the next topics.

3. In the beginning, any but not the exact HMM model parameters λ= (*A, B*, Π) exist, so you have only set of observation sequence $X = \{O^k\}_k$, and you would like to find a HMM parameters λ* which maximizes the probability of generating *X* , P(*X*| λ). In this problem Baum-Welch algorithm will be used and it will be touched on next headers.

Now in the next header, the HMM representation of example, which is given at the above, will be shown then solutions to three main HMM question will be given in the example.

2.3.3.2 HMM representation of example

Remember that there are three states home, work and outside and there are three observations suit, tracksuit, jeans in the previous example. Let's see the parameters of the example like in HMM parameters section systematically.

1. $S = \{h, w, o\}$ since there are three states, N=3.
2. The observations are expected as distinct and all states generate same observations $V = \{s, t, j\}$ so M=3.
3. Assume that the transition matrix "A" is given as below.

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} a_{hh} & a_{hw} & a_{ho} \\ a_{wh} & a_{ww} & a_{wo} \\ a_{oh} & a_{ow} & a_{oo} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.3 & 0.2 \\ 0.4 & 0.5 & 0.1 \\ 0.2 & 0.1 & 0.7 \end{bmatrix}$$

4. Moreover, assume that emission matrix "*B*" is given below.

$$B = \begin{bmatrix} b_1(1) & b_1(2) & b_1(3) \\ b_2(1) & b_2(2) & b_2(3) \\ b_3(1) & b_3(2) & b_3(3) \end{bmatrix} = \begin{bmatrix} b_h(s) & b_h(t) & b_h(j) \\ b_w(s) & b_w(t) & b_w(j) \\ b_o(s) & b_o(t) & b_o(j) \end{bmatrix} = \begin{bmatrix} 0.2 & 0.5 & 0.3 \\ 0.7 & 0.1 & 0.2 \\ 0.1 & 0.3 & 0.6 \end{bmatrix}$$

5. "Π" is the initial start probabilities of states. It is same with the discrete Markov Model

$$\Pi = [0.5 \quad 0.2 \quad 0.3]^T$$

The HMM representation of the HMM model parameters λ= (*A*, *B*, Π) that specified above, is demonstrated in Figure 2.5.

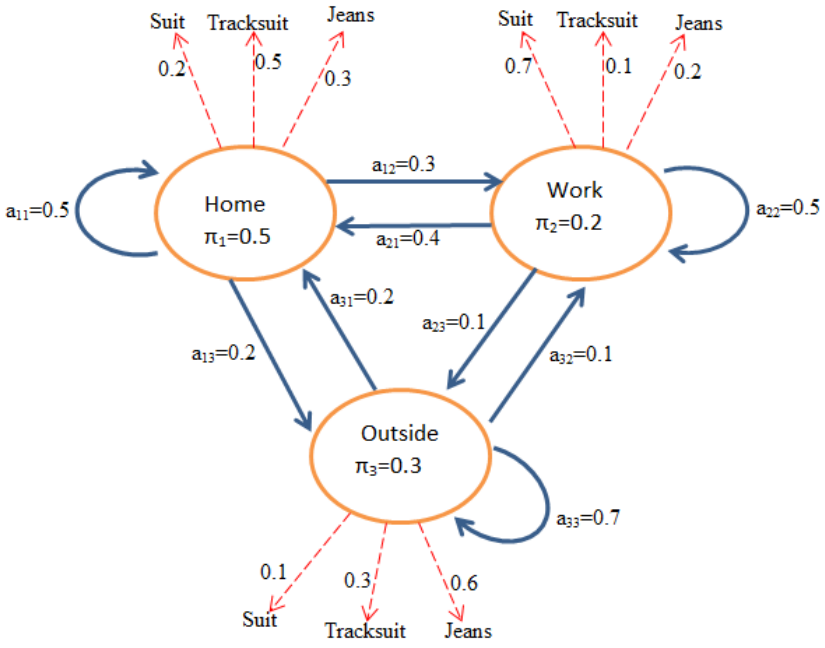

Figure 2.5 : HMM Representation.

2.3.3.3 Answers to three main questions of HMM

Problem 1: Calculation of the probability of observation

In this question, HMM model parameters λ= (*A*, *B*, Π) are given and we would like to calculate the probability of observing sequence *O* that refers with P (*O*| λ). Suppose that an observation sequence *O*= {$O_1O_2…O_T$} and a fixed state sequence *Q*= {$q_1q_2…q_T$} are given. If the question is the probability of observing *O* at the corresponding state sequence *Q*, the answer will be simply

$$P(O|Q,\lambda) = \prod_{t=1}^{T} P(O_t|q_t,\lambda) = b_{q1}(O_1) \cdot b_{q2}(O_2) \cdots b_{qt}(O_T) \quad \textbf{(2.13)}$$

However, here the state sequence *Q* is unknown and each observation at time t in *O* can be produced by any of states so the probability of any state sequence *Q* can be shown as [38].

$$P(Q|\lambda) = P(q_1) \prod_{t=2}^{T} P(q_t|q_{t-1}) = \pi_{q1} a_{q1q2} \cdots a_{qt-1qt} \quad \textbf{(2.14)}$$

The probability of occurring *O* and *Q* at the same time is shown with the joint probability of two equations,

$$P(O,Q|\lambda) = P(O|Q,\lambda)P(Q,\lambda) \tag{2.15}$$

It can be thought that O and *Q* occur simultaneously, but in fact *O* is given and it has to be calculated on all possible states. Since all states can generate the observation O_T at any time t, the calculation changes to,

$$\begin{aligned} P(O|\lambda) &= \sum_{all\ possible\ Q} P(O|Q,\lambda)P(Q|\lambda) \\ &= \sum_{q_1,q_2,\cdots,q_T} \pi_{q1} b_{q1}(O_1) a_{q1q2} b_{q2}(O_2) \cdots a_{qt-1qT} b_{qt}(Q_T) \end{aligned} \tag{2.16}$$

Rabiner [38] says that the equation above requires $2TxN^T$ calculations so this is inefficient. Instead of this calculation, Forward-Backward algorithm is used to calculate the probability of observation sequence (P (*O*| λ)) for given model λ. Auxiliary variable α_t (i) is used as forward variable.

$$\alpha_t(i) \equiv P(O_1 \cdots O_t, q_t = S_i|\lambda) \tag{2.17}$$

Forward algorithm computes the probability of the partial observation sequence until time t $\{O_1 \ldots O_t\}$, where the underlying Markov process in state S_i at the time t [36].

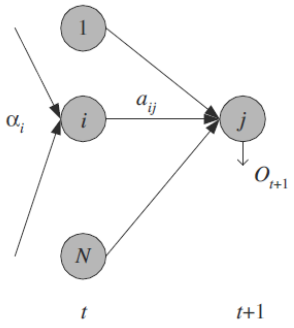

Figure 2.6 : Calculation of a α_t (i) [36].

The problem can be solved inductively.

1. Initialization

$$\alpha_1(i) = \pi_i b_i(O_1), \quad 1 \le i \le N \tag{2.18}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}), \qquad 1 \le t \le T-1\,, 1 \le j \le N \tag{2.19}$$

3. Termination

$$\mathrm{P(O|\lambda)} = \sum_{i=1}^{N} \alpha_T(i) \tag{2.20}$$

These operations are illustrated in below, Figure 2.7.

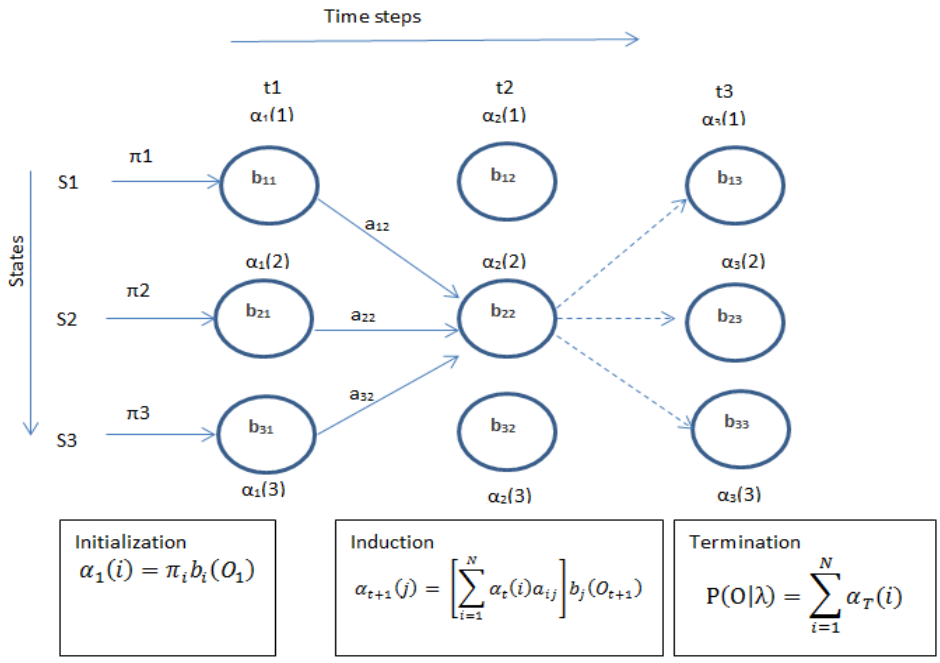

Figure 2.7 : Forward algorithm's steps.

The observation sequence can also be evaluated with the backward algorithm similar to forward algorithm. The backward variable, β_t (i) means the probability of observing partial sequence $O_{t+1}\ldots O_T$ then being state S_i at time t [36].

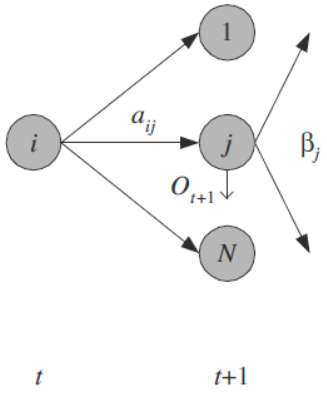

Figure 2.8 : Calculation of a βt (i) [36].

It is formulated with the below equation nearly identical with the forward variable.

$$\beta_t(i) \equiv P(O_{t+1} \cdots O_T | q_t = S_i, \lambda) \tag{2.21}$$

Just as the forward variable, the problem can be solved inductively with the backward variable.

1. Initialization

$$\beta_T(i) = 1, \quad 1 \leq i \leq N \quad [38] \tag{2.22}$$

2. Induction

$$\beta_t(i) = \left[\sum_{j=1}^{N} a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)\right], \tag{2.23}$$
$$t = T-1, T-2, \cdots, 1\ , 1 \leq i \leq N \ [38]$$

3. Termination

$$\mathrm{P}(\mathrm{O}|\lambda) = \sum_{j=1}^{N} \pi_j b_j(O_1) \beta_1(j) \tag{2.24}$$

Problem 2: Most likely state sequence

The problem is decoding the most likely state sequence Q= $\{q_1 q_2 \ldots q_T\}$, with given model λ= (A, B, Π) and observation sequence *O*= $\{O_1 O_2 \ldots O_T\}$. Rabiner [38] emphasizes that the problem two has not exact solution and it is difficult to solve. According to [38] optimal state sequence, which has conjunction with the

observation sequence *O*, has to be optimal, but there are more than one optimality criteria and this situation makes things difficult. Problem two also has an auxiliary variable named $\gamma_t(i)$ which defines the given *O* and λ, probability of being at S_i at time t.

$$\gamma_t(i) = P(q_t = S_i|O,\lambda) = \frac{P(O|q_t = S_i,\lambda)P(q_t = S_i|\lambda)}{P(O|\lambda)} \quad \textbf{(2.25)}$$

In terms of forward-backward variable,

$$\gamma_t(i) = \frac{\alpha_t(i)\beta_t(i)}{P(O|\lambda)} = \frac{\alpha_t(i)\beta_t(i)}{\sum_{j=1}^{N}\alpha_t(j)\beta_t(j)} \quad \textbf{(2.26)}$$

In the equation, $\alpha_t(i)$ stands for the beginning part of the sequence until time t ($O_1O_2\ldots O_t$) and the backward variable $\beta_t(i)$ stands for the remaining part of the sequence until time T ($O_{t+1}Ot+_2\ldots O_T$).

From the variable $\gamma_t(i)$ it can be found the highest probability state at time t,

$$q_t = \underset{1\le i\le N}{\operatorname{argmax}}[\gamma_t(i)], \quad 1 \le t \le T \quad \textbf{(2.27)}$$

It is guarantee that $\sum_i \gamma_t(i) = 1$ by dividing to multiplication of $\alpha_t(i)$ and $\beta_t(i)$ at the all possible states at time t [36][38].

This equation solves the problem by choosing the most likely state for each t [38]. Nevertheless, when the transition variable is equals to zero, the problem still will be occurred, and the equation will give invalid state sequence. The solution to this problem based on the dynamic programming methods is the Viterbi algorithm [38].

In the Viterbi algorithm, variable $\delta_t(i)$ shows the path which has the highest probability at time t, that accounts for the first t observations and ends in state S_i [36][38].

$$\delta_t(i) \equiv \max_{q_1q_2\cdots q_{t-1}} p(q_1q_2\cdots q_{t-1}, q_t = S_i, O_1\cdots O_t|\lambda) \quad \textbf{(2.28)}$$

In this algorithm, $\delta_{t+1}(i)$ can be calculated recursively. Then variable $\psi_t(j)$ is used to find optimal path. The variable $\psi_t(j)$ keeps the track of the path that maximizes $\delta_t(j)$ up to time t and the best previous state [36]. The Viterbi algorithm works as follows:

1. Initialization:

$$\delta_1(i) = \pi_i b_i(O_1)$$
$$\psi_1(i) = 0 \quad \textbf{(2.29)}$$

2. Recursion:

$$\delta_t(j) = \max_i \delta_{t-1} a_{ij} b_j(O_t)$$
$$\psi_t(j) = \operatorname{argmax}_i \delta_{t-1}(i) a_{ij} \quad \textbf{(2.30)}$$

3. Termination:

$$p^* = \max_i \delta_T(i)$$
$$q_T^* = \operatorname{argmax}_i \delta_T(i) \quad \textbf{(2.31)}$$

4. Path backtracking:

$$q_t^* = \psi_{t+1}(q_{t+1}^*), \qquad t = T-1, T-2, \dots, 1 \quad \textbf{(2.32)}$$

The Viterbi algorithm works similar to forward algorithm. Instead of summation in forward algorithm it uses the maximum. At time t, each state keeps the previous state (at time t-1) which comes with the highest probability of observation sequence until time t [40].

Problem 3: Parameter training

In the beginning, not exact HMM model parameters λ= (*A*, *B*, Π) exist, so you have only set of observation sequence $X = \{O^k\}_k$, and you would like to find an HMM parameters λ* which maximizes the probability generating *X* P(*X*| λ). There is no known way to solve analytically maximizing the P (*O*| λ) problem. However, it is possible to find locally maximum point with an iterative procedure Baum-Welch[38].

Initially the variable $\xi_t(i, j)$ has to be defined which represents the probability of being in state S_i at time t and being state S_j at time t+1, given the model and observation sequence, i.e. Figure 2.10 [38].

$$\xi_t(i,j) \equiv P(q_t = S_i, q_{t+1} = S_j | O, \lambda) \quad \textbf{(2.33)}$$

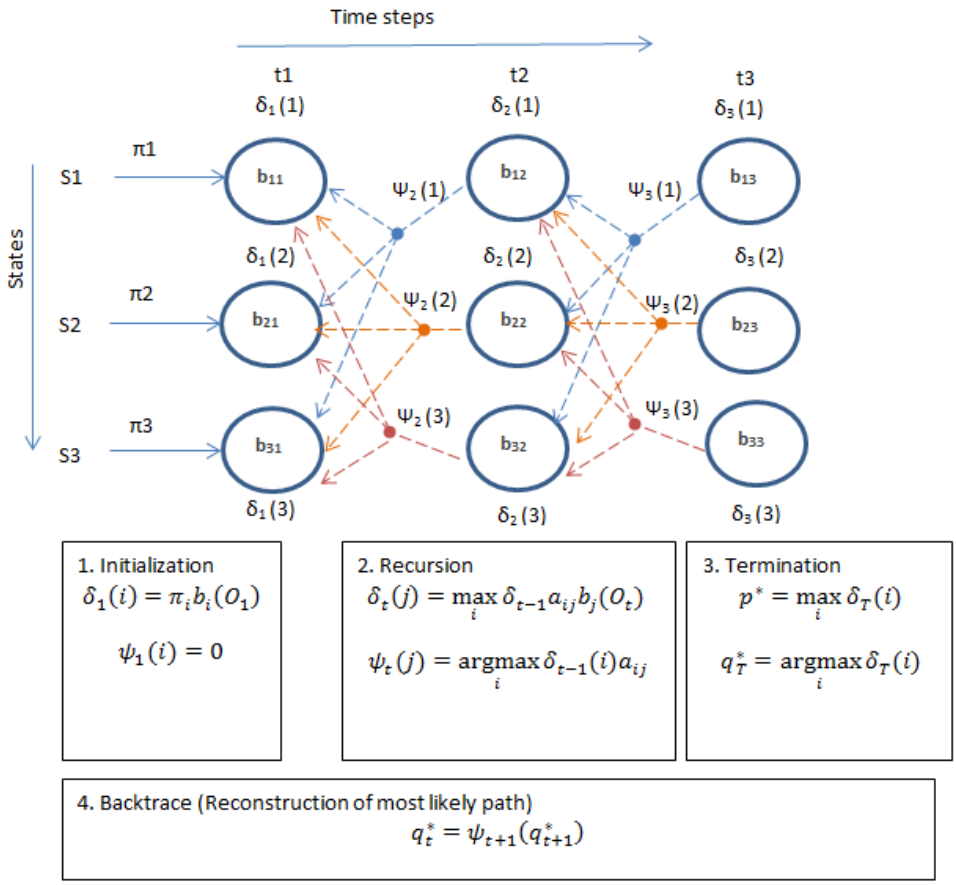

Figure 2.9 : Viterbi path [40].

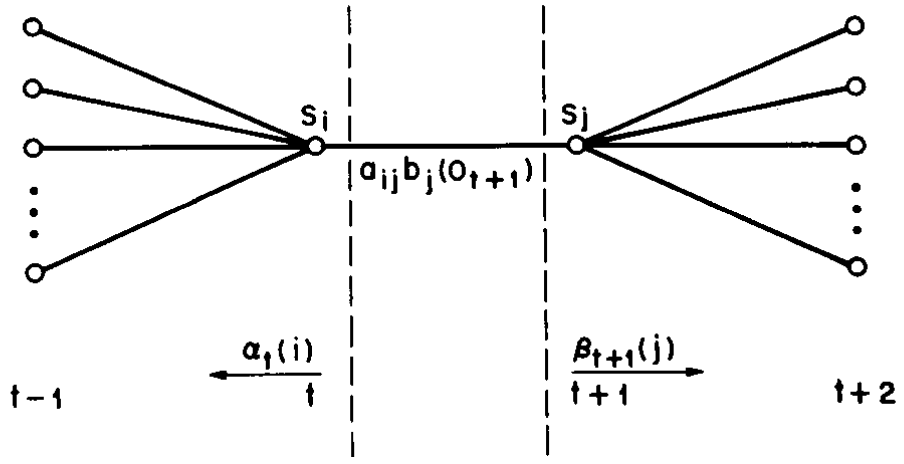

Figure 2.10 : ξt(i,j) illustration [38]**.**

In terms of forward-backward variable it will be,

$$\xi_t(i,j) = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{P(O|\lambda)} = \frac{\alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)}{\sum_{i=1}^{N} \sum_{j=1}^{N} \alpha_t(i) a_{ij} b_j(O_{t+1}) \beta_{t+1}(j)} \quad \textbf{(2.34)}$$

As it is illustrated in Figure 2.10, $\alpha_t(i)$ shows the first t observation from beginning to state S_i, then the system moves to state S_j with the probability of a_{ij}. After that, the system observes the observation t+1, then generates the rest of the observation. The equation is normalized with the dividing by all possible couple of states that visited at time t and t+1 [36].

In the problem two the variable $\gamma_t(i)$ is defined, which the probability of being in state S_i at time t, given observation sequence *O* and model λ. If this variable related to $\xi_t(i, j)$ the equation will be,

$$\gamma_t(i) = P(q_t = S_i|O, \lambda) = \sum_{j=1}^{N} \xi_t(i,j) \quad \textbf{(2.35)}$$

If $\gamma_t(i)$ is summed over time t, this gives the expected number of transitions from Si and summation of to $\xi_t(i, j)$ over time t gives the expected number of transitions from S_i to S_j [38].

$$\sum_{t=1}^{T-1} \gamma_t(i) = expected\ number\ of\ transitions\ from\ S_i \quad \textbf{(2.36)}$$

$$\sum_{t=1}^{T-1} \xi_t(i,j) = expected\ number\ of\ transitions\ from\ S_i\ to\ S_j \quad \textbf{(2.37)}$$

Baum-Welch is an expectation maximization algorithm. Calculation of $\gamma_t(i)$ and $\xi_t(i, j)$ using given $\lambda = (A, B, \Pi)$ is the expectation part of baumwelch. In the maximization step $\bar{\pi}_i, \bar{a}_{ij}, \bar{b}_i(j)$ will be updated with the new values using current $\gamma_t(i)$ and $\xi_t(i, j)$ values.

$$\bar{\pi}_i = expected\ number\ of\ times\ in\ state\ S_i\ at\ time\ (t=1) = \gamma_1(i) \quad \textbf{(2.38)}$$

$$\bar{a}_{ij} = \frac{expected\ number\ of\ transitions\ from\ state\ S_i\ to\ state\ S_j}{expected\ number\ of\ transitions\ from\ S_i} = \frac{\sum_{t=1}^{T-1} \xi_t(i,j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \textbf{(2.39)}$$

$$\bar{b}_j(k) = \frac{expected\ number\ of\ times\ in\ state\ j\ and\ observing\ v_k}{expected\ number\ of\ times\ in\ state\ j} = \frac{\sum_{\substack{t=1 \\ s.t.O_t=v_k}}^{T} \gamma_t(j)}{\sum_{t=1}^{T} \gamma_t(j)} \quad \textbf{(2.40)}$$

HMM has some stochastic constraints, which are shown in the equation below. These constraints are automatically satisfied at each iteration in the maximization step.

$$\sum_{i=1}^{N} \bar{\pi}_i = 1 \tag{2.41}$$

$$\sum_{j=1}^{N} \bar{a}_{ij} = 1, \quad 1 \le i \le N \tag{2.42}$$

$$\sum_{k=1}^{M} \bar{b}_j(k) = 1, \quad 1 \le j \le N \tag{2.43}$$

If the training data consists of multiple observation sequence, $X = \{O^k\}_k$, then each observation is assumed as independent. Therefore:

$$P(X|\lambda) = \prod_{k=1}^{K} P(O^k|\lambda) \tag{2.44}$$

In this case, calculation of HMM parameters evolve into averages over all observation in all sequence [36]:

$$\bar{a}_{ij} = \frac{\sum_{k=1}^{K} \sum_{t=1}^{T_k - 1} \xi_t^k(i,j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k - 1} \gamma_t^k(i)} \tag{2.45}$$

$$\bar{b}_j(m) = \frac{\sum_{k=1}^{K} \sum_{\substack{t=1 \\ s.t. O_t = v_k}}^{T} \gamma_t(j)}{\sum_{k=1}^{K} \sum_{t=1}^{T_k} \gamma_t^k(j)} \tag{2.46}$$

$$\bar{\pi}_i = \frac{\sum_{k=1}^{K} \gamma_1^k(i)}{K} \tag{2.47}$$

2.4 Dynamic Time Warping

The DTW is an algorithm, which computes an optimal matching between two temporal sequences, which may vary in speed or time [31]. It is commonly used in speech recognition, shape matching, and gesture recognition. If any data can be illustrated in linear sequence, it can be analyzed with DTW. The DTW algorithm calculates the distance between two sequences of feature vector, which allows examination of stretched and compressed section of the sequences, finds the optimal match between them and computes the cumulative distance between each possible pair of features (points in vector). It also compares and aligns these two sequences.

Assume we have two sequences which are $A=\{a_1,a_2,\ldots,a_n\}$ and $B=\{b_1,b_2,\ldots, b_m\}$ of length n and m respectively. Each point (a_i or b_i) in the vectors (A or B) is a d-dimensional vector like $a_i=\{a_i^1, a_i^2, \ldots, a_i^d\}$ or $b_i=\{b_i^1, b_i^2, \ldots, b_i^d\}$. We want to measure similarity of these two sequences.

An n x m distance matrix (D) is constructed from the A and B matrices. Each cell (i, j) in the D defines distance between i – th feature from A and j-th feature from B. There are some distance calculation methods like Euclidian distance, Manhattan distance, Chebyshev distance measurement; Euclidian distance is the most commonly used one.

$$d_{ij} = \left|a_i - b_j\right|, \qquad i = 1,n \quad j = 1,m \tag{2.48}$$

The distance matrix includes n rows and m columns of terms above. Finding the shortest path in distance matrix can be done through brute force methods but generally this problem is solved by dynamic programming. *cij* is minimum shortest path up to the adjacent cells.

$$c_{ij} = d_{ij} + \min\left(c_{i-1,j-1}, c_{i-1,j}, c_{i,j-1}\right) \tag{2.49}$$

Sakoe and Chiba [41] summarized some constraints in DTW to obtain good path and to limit the explored paths.

- Monotonicity: The alignment path will not turn back in time index. Both i and j indexes either stay same or increase, they never decrease It is guaranteed that features are not repeated in the alignment with this condition.
- Continuity: The alignment path jumps one step at a time. This condition guarantees that important features are not omitted with this condition.
- Boundary: The alignment starts in left-down corner c11 and ends in right-up corner cnm. It is guaranteed through that condition that the sequences are not considered only partially.

Warping window: A good alignment path is unlikely to wander too far from the diagonal. Through warping window condition it is not allowed to stick at similar feature or the path can’t jumps to different features.

3. GESTURE RECOGNITION

As explained in "A CASE STUDY: INTERACTIVE ISIGN GAME" section, the gesture of the test subject must be recognized and must be given feedback based on the recognition result by robot. Gesture recognition process will be expressed in this section.

3.1 Overview

Sign Language is a visual language based on upper body movements (including hands, fingers, arms, upper torso, head and neck) and facial gestures. The gesture recognition is used for forming more powerful interaction between machines and humans and this enables humans to interact without any controller. In other words, the gestures will be used as an input by machines. The studies introduced in this thesis aim to utilize humanoid robots for assisting sign language tutoring due to the lack of sufficient educational material [42]. Therefore, the gesture of the person who learns the sign language from humanoid robot, should be recognized and the person should be able to control the robot with his/her gestures. There are various research in computer science dealing with gesture recognition. This currently active and focused topic generally uses the camera and computer vision algorithms for gesture recognition. Depending on the input data type, there are different approaches to gesture recognition, i.e. Figure 3.1.

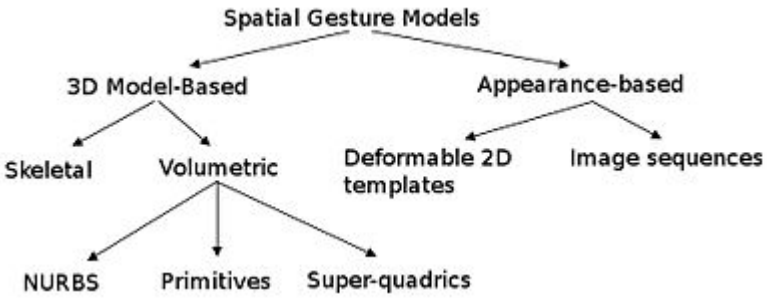

Figure 3.1 : Some ways of tracking and analyzing gesture [43].

Sign language recognition comprises of two basic processes, namely feature extraction and feature recognition.

Feature extraction is a special form of dimensionality reduction [44]. Since the original data is too large to be processed, and some parts of it are unnecessary, the input data is converted into the feature vector on the purpose of summarizing it nearly keeping its originality. The extracted features should represent the original data as much as possible without significant data loss, and should be computationally cost effective.

In the feature extraction step, the data of the joint points provided by RGDB camera of the Kinect is used as features. The data of the joint points is sent in the form of spatial data by Kinect. This situation causes changes in the results depending on the length of the arm, leg and other joints of the test subject. For preventing these undesired changes in the results, spatial data are converted into the angle data. Spatial data to angle data conversion will be mentioned in Transformation of Kinect's Spatial Data to Angle Data section and the description of the gesture is given in Gesture section.

In the feature recognition step, there are three methods used, which are K-means with HMM, ANN with HMM and real time gesture recognition. These methods will be explained in the following K-Means and HMM Method, Neural Network and HMM Method and Real-time Gesture Recognition sections.

3.2 Transformation of Kinect's Spatial Data to Angle Data

Several image processing and computer vision techniques are used to detect the skeletal model of human successfully using RGBD cameras i.e. Kinect Sensor. Thus, the job to determine a good feature set for gesture classification becomes easier with the availability of Kinect.

In this study, the Kinect's depth sensors are employed, and the depth data is used as a feature to recognize gestures.

RGBD camera (Kinect Sensor) starts the input stream (30 frames per second) and sends every gestural motion data frame-by-frame. All frames include 20 joints in the body, which Kinect can track, and joint spatial coordinates (x, y, and z) of skeletal structure for each joint are generated. Joints are shown in Figure 5.3.

All 20 joints provided by Kinect cannot be used in this study, because of the disabled children's and NAO's constraints. Some gestures are predetermined that can be made

by five joints for recognition. So in the scope of this thesis, only the five of joints will be used in recognition process. These are “Head”, “Shoulder Left”, “Shoulder Right”, “Elbow Left”,” Elbow Right” which are demonstrated in Figure 3.3.

The sign language consists of upper body based movements since the robot has some kinematic constraint. NAO has 25 degrees of freedom illustrated in Figure 3.2 .

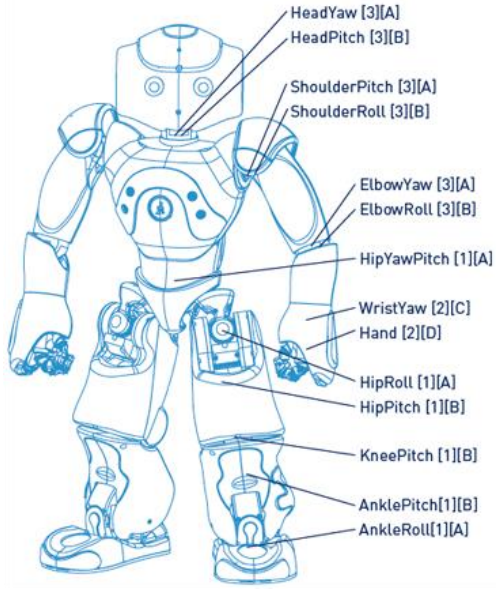

Figure 3.2 : Nao H-25 Joints [45].

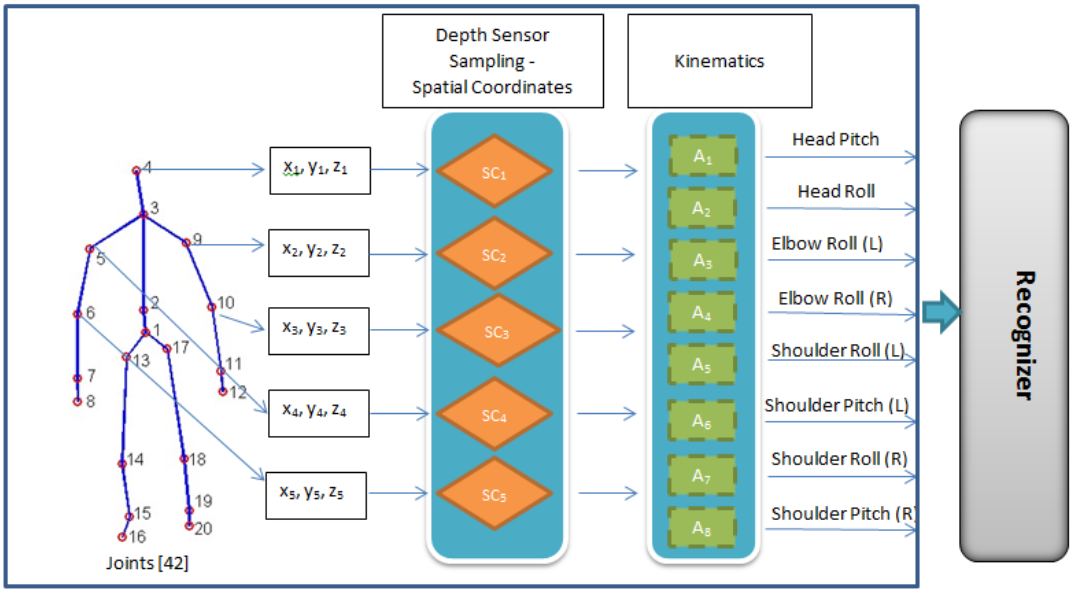

Figure 3.3 : Evolution of a data frame.

In order to provide robustness, every frame in gestural motion data is expressed as a single vector of angle values (Roll, Pitch, Yaw) which is computed from spatial position values (x, y, z) for every joint node of skeletal model. Some of these angles

are rejected because they are redundant for predetermined gestures or some of them cannot be used due to kinematic constraints of NAO H-25 robot. Finally, an angle vector or frame is created which is consisted of “Head Pitch”, “Head Roll”, “Left Elbow Roll”, “Right Elbow Roll”, “Left Shoulder Pitch”, “Left Shoulder Roll”, “Right Shoulder Pitch”, and “Right Shoulder Roll” angles.

Every frame has a spatial data of the test subject’s joint points. Because every person implementing a gesture in front of the Kinect has a different length of arm, leg and other joint points, the data in question entails a change in the result of recognition process. In order to eliminate the dependence on people, the spatial data are transformed to the angle data.

Different gestural motion data taken from RGBD camera (Kinect Sensor) can have different number of frames. Thus, representation of every gesture pattern should be carefully modeled so that recognition process meets the performance criteria for robust recognition.

3.3 Gesture

A gesture is a sequence of frames, which involves angle vector that is transformed from spatial data. Angle vector generated by transforming five joints to the eight angles is depicted in Figure 3.3.

In the recognition phase, the gestures of the test subjects are recognized by using statistical models such as k-means, HMM and ANN. Due to statistical learning’s nature, a dataset including training, validation and test samples is needed. For constituting aforesaid dataset, test subjects are asked to execute the previously determined gestures and all gestures are recorded from the beginning of the gesture to the end of the gesture. Gestures are required in both of training and test phases. In order to create sufficient amount of gesture, people execute each gesture more than once. A portion of these gestures is separated for training step and remaining part are reserved for test step.

A gesture file is created for every test subject by adding angle vector converted from spatial data obtained from Kinect. The frame numbers, which represents the duration of the gesture, specifies the size of the file (shown in Figure 3.4.).

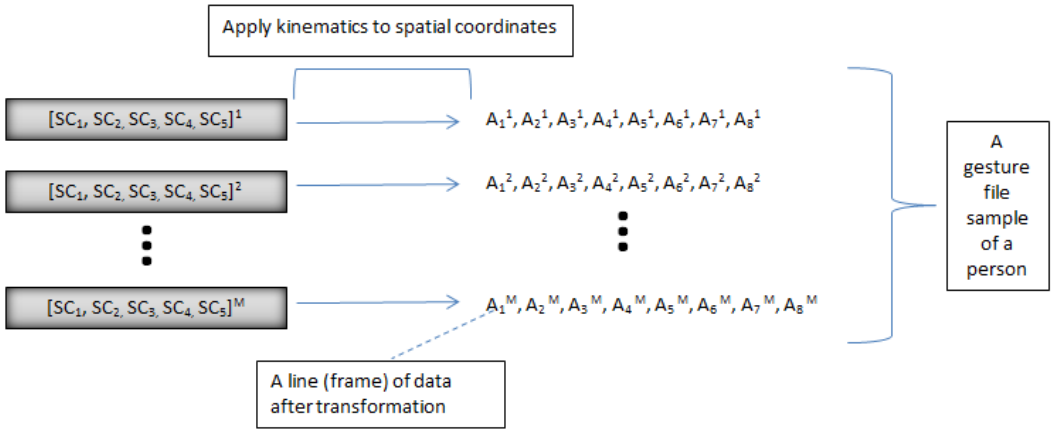

Figure 3.4 : Sample gesture file of a person.

The main challenge in the recognition processes is detecting a gesture from continuous Kinect data. In order to read Kinect data discretely, window-sliding method is used. In this method, a window size is determined as the slowest person completion time for detecting the beginning of the gesture to recognize it. A window slides through the continuous data and the data under the window is sent to the related HMM model. According to the goodness of the result of the HMM model, window is continued to slide on the data.

The sliding window method is used for continuously reading problem but besides this method another brute force method, the comparison of angle, is also used for recognizing the gesture. These two methods work well together but there are many limitations. The gestures, which used in the system, have to be completely distinctive, and the angles needed to be at the edge point. These problems were prompted to another recognition methods that will mentioned in the next topics. These methods included some pattern recognition and machine learning techniques.

3.4 K-Means and HMM Method

In the scope of this method six motions, “Car”, ”Dad”, ”Forward”, “Table”, “Side”, ”Up”, which are nearly isolated because of the constraints mentioned in the collection of the data part, are used.

K-means and Hidden Markov Model are implemented in the scope of this thesis.

There are two phases in this method. First one is to represent gesture pattern (Sign Language (SL) word), in the terms of k-means and HMM. A gesture data consists of

a file whose line count is M, shown in Figure 3.4. During gesture implementation, a person starts from an initial position then follows a trajectories and the motion is finished at the end point.

Assume that there are two people, one of them implements a gesture and the other is an observer. The Observer tries to recognize the implemented gesture is done successfully or not. Each person can do the same gesture with different speed or can skip some positions of the gesture. However, generally, detecting the fundamental positions of the joints for the gesture is sufficient to recognize the gesture. This doctrine can be applied to the algorithm behind the gesture recognition in the same manner. The fundamental positions of the joints correspond to **states** in this method. For instance, while doing the "Side" motion illustrated in Figure 3.5, the upper torso is in many positions during at least 60 frames, but three states of the body shown in Figure 3.5 is enough to recognize this motion. Therefore, if these states are detected, it can be said that the "Side" motion was implemented successfully.

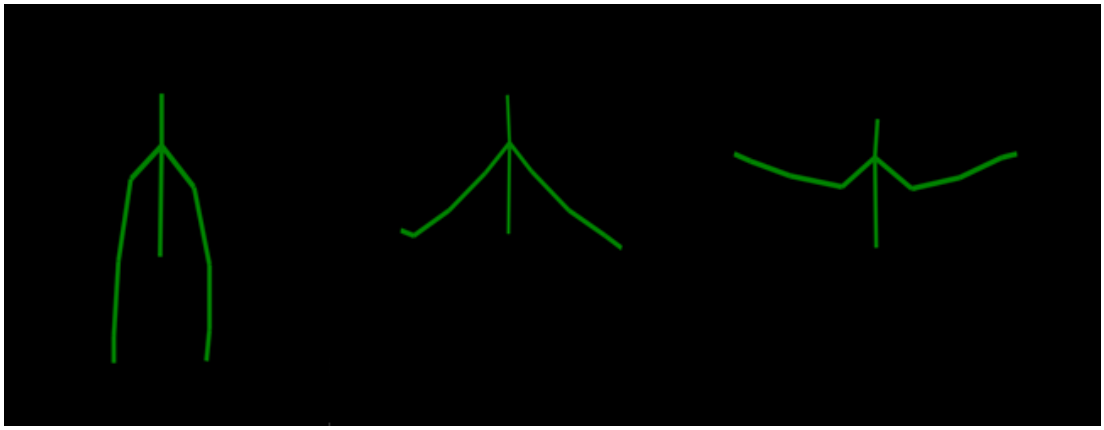

Figure 3.5 : "Side" gesture motion of a participant.

In the first phase of this method's test step, the features have to be transformed into terms of k-means and HMM. These operations are demonstrated in Figure 3.6**.** The demonstration in the figure belongs to a single motion and a single HMM model. The number of the HMM model equals to the number of motion in the system.

According to this model, a motion file which consists of angle lines are given to the k-means algorithm. The k-means algorithm clusters the each line of the motion file then assigns labels to each cluster. Finally it produces a corresponding label sequence for the file. The label sequence indicates the life cycle (how the motion occurred) of the motion.

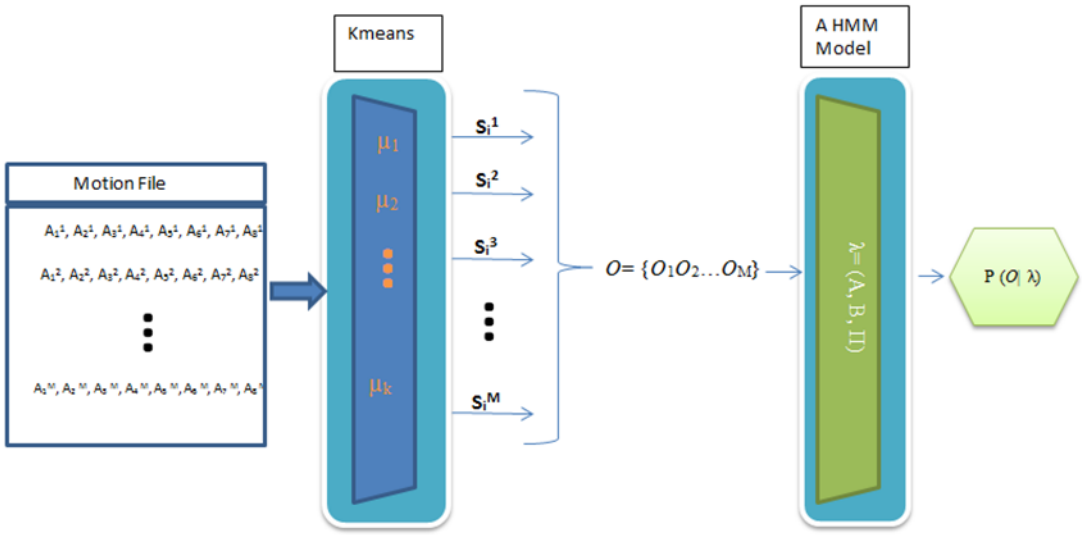

Figure 3.6: Evaluation of motion file in a gesture model.

Based on the clustered data coming from the K-Means algorithm, an observation sequence is generated for HMM. The HMM model of related k-means takes this observation sequence and calculates the probability of it via forward algorithm, which is mentioned in the Hidden Markov Model topic.

The second phase of the test step is model decision. In this method, every motion has a model, which consists of a k-means and HMM. When a file is inserted to the system, each model takes this file as an input, then each model calculates the probability of this file, and finally these probabilities are used for making decision. HMM has a model selection decision, which is formulated as:

$$P(\lambda_i|O) = \frac{P(O|\lambda_i)P(\lambda_i)}{\sum_j P(O|\lambda_j)P(\lambda_j)} \quad \textbf{(3.1)}$$

P (λ_i) is the prior probability of model *i*. Notice that all of the predetermined models for this method have the same probability of occurrence so within the scope of this method, the decision will be taken according to highest probability. The second phase of the test step is shown in Figure 3.7.

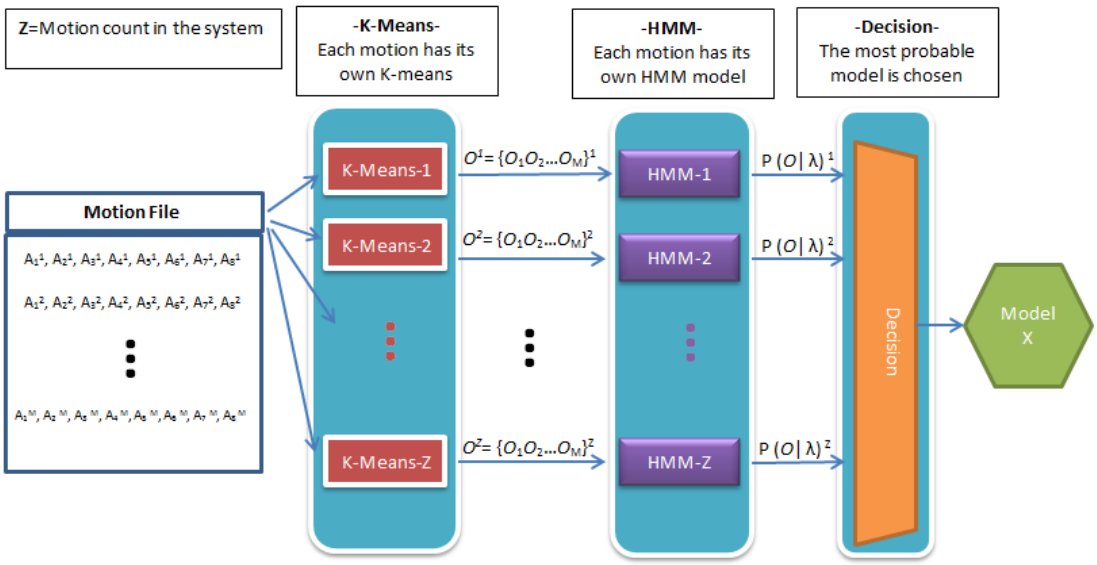

Figure 3.7: Model decision of method.

3.5 Neural Network and HMM Method

In the previous method, states of the HMM was being determined by the k-means. Since k-means is an unsupervised learning algorithm and the results of it are affected significantly by the selection of the initial centroid, it may cause to wrong state and observation assignments. If we explain this problem with an instance, assume that an observation sequence O = $\{O_1O_1O_2O_2O_3O_3\}$is expected to send from k-means, but O = { $O_3O_1O_2O_1O_3O_3\}$ observation sequence is sent from k-means because of the change in the initial centroid. While calculating the probability of observing sequence given λ, P (O| λ), forward algorithm is used as stated in the previous section. Suppose that a transition matrix, A, is given as below:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 0.5 & 0.5 & 0 \\ 0 & 0.5 & 0.5 \\ 0 & 0 & 1 \end{bmatrix}$$

According to the expected observation sequence, first O_1 is observed, then O_1 is observed again. When computing the induction phase of the forward algorithm as in the formula below:

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] b_j(O_{t+1}) \qquad \textbf{(3.2)}$$

$a_{ij} = a_{11}$ is used 0.5 as in the table above. However, according to the observation sequence sent from k-means, first O_3 is observed, then O_1 is observed. So, $a_{ij} = a_{31}$ is used 0 as in the table above and the result will be completely different. Depending on this calculation, the model decision can be wrong.

Second problem for previous method is long training time. The Baumwelch is not a cost effective algorithm and depending on the training observation size and training data size, it can be taken more than an hour to train. If a new sign/gesture is added to the system, it takes very long time to adopt it.

These problems lead us to the new method. This new method is obtained by combining the HMM algorithm with the ANN algorithm while computing the result of the forward algorithm. The difference from the prior method is making use of ANN for acquiring the emission matrix and the observation sequence instead of k-means algorithm.

When a gesture file comes to system, feature transformations (spatial data to angle data conversion) are applied to the file. Then processed file is thought as observation sequence and line by line it is sent to the neural network. Consequently, the neural network returns the probability of line for each state. These probabilities are used at time t (line number) for corresponding states emissions.

In this method, the model of the HMM which is λ= (*A*, *B*, Π) is altered to λ= (*A*, Π, ANN). Moreover, the forward algorithm depicted in Figure 3.8 can solve inductively as in the formulas below:

In the condition of ANN formula which is:

$$z_h = sigmoid(w_h^T x) = \frac{1}{1 + exp\left[-\left(\sum_{j=1}^{d} w_{hj} x_j + w_{h0}\right)\right]}, h = 1, \dots, H \quad \textbf{(3.3)}$$

$$y_i = v_i^T z = \sum_{h=1}^{H} v_{ih} Z_h + v_{i0} \quad \textbf{(3.4)}$$

1. Initialization

$$\alpha_1(i) = \pi_i ANN_i(O_1), \quad 1 \le i \le N \quad \textbf{(3.5)}$$

2. Induction

$$\alpha_{t+1}(j) = \left[\sum_{i=1}^{N} \alpha_t(i) a_{ij}\right] ANN_j(O_{t+1}), \quad 1 \le t \le T-1 \ \ and \ 1 \le j \le N \tag{3.6}$$

3. Termination

$$P(O|\lambda) = \sum_{i=1}^{N} \alpha_T(i) \tag{3.7}$$

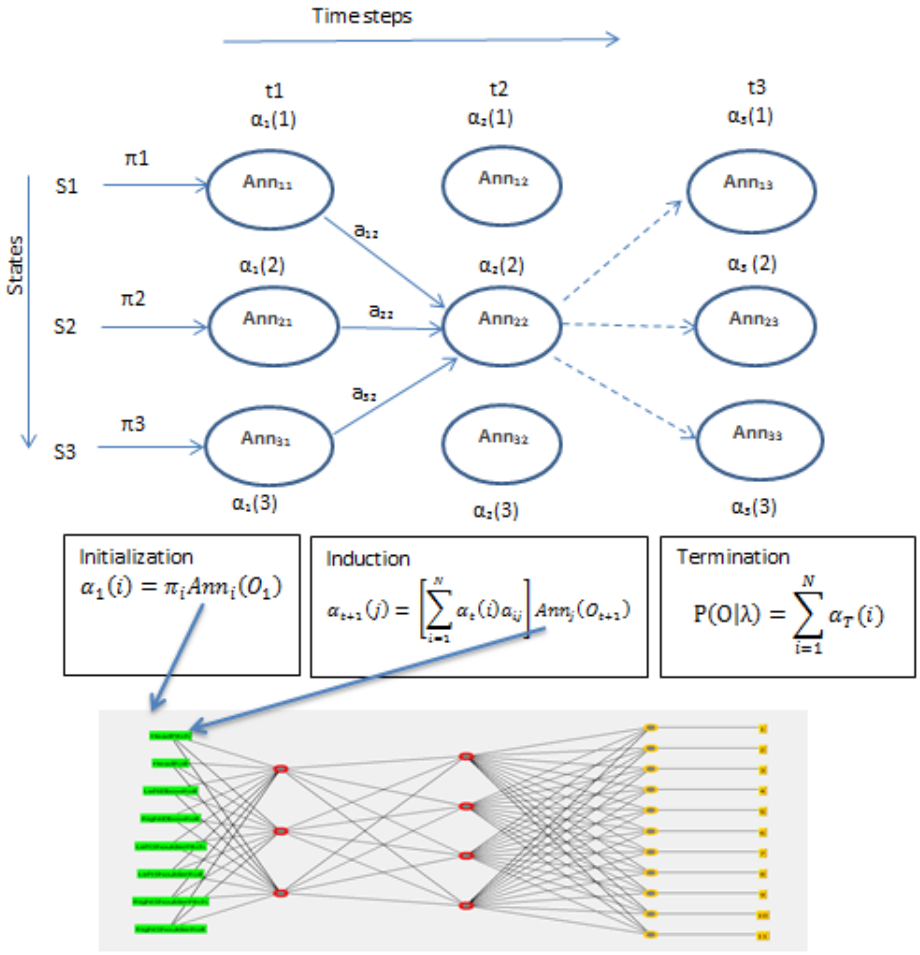

Figure 3.8 : ANN in Forward Algorithm.

Within the context of this method, two approaches are utilized.

In the first approach, every sign/gesture has its own states. For example, the side gesture states are shown in Figure 3.9 and the up gesture states are shown in Figure 3.10.

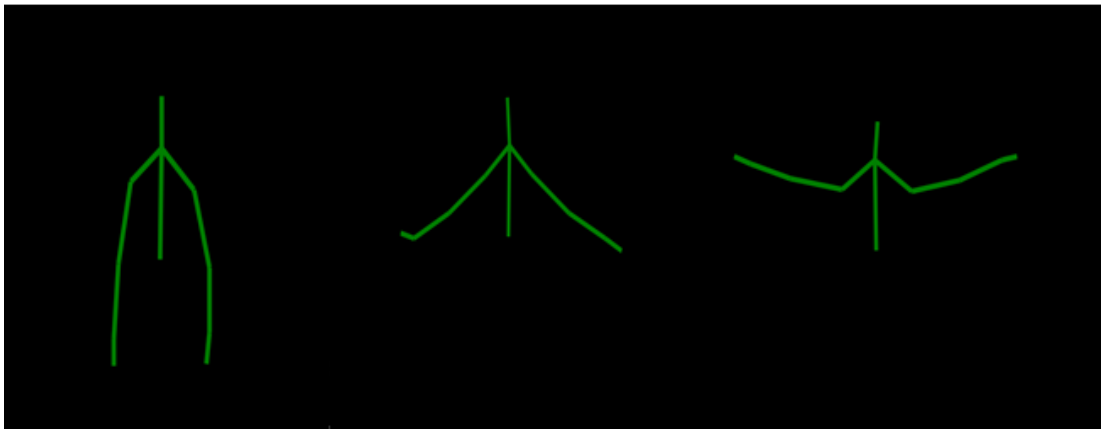

Figure 3.9 : Side gesture states.

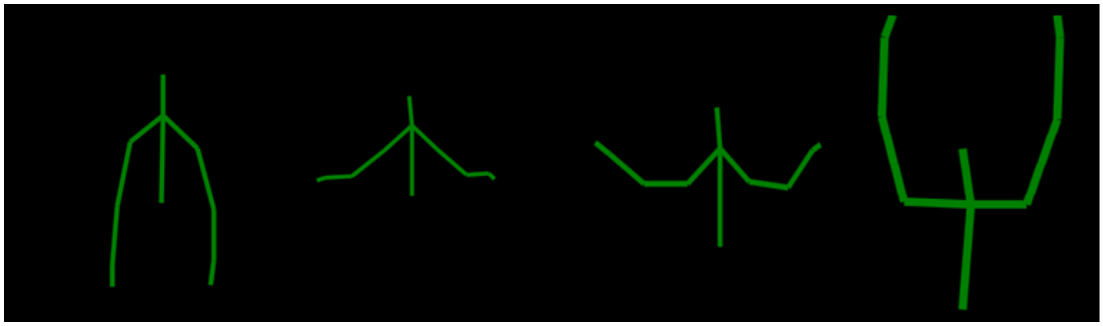

Figure 3.10 : Up gesture states.

According to these states, a neural network is trained as stated in the prior sections for each sign/gesture. Finally, each HMM model uses the result of corresponding neural network as an emission probability for the related sign/gesture. The approach is summarized in Figure 3.11 .

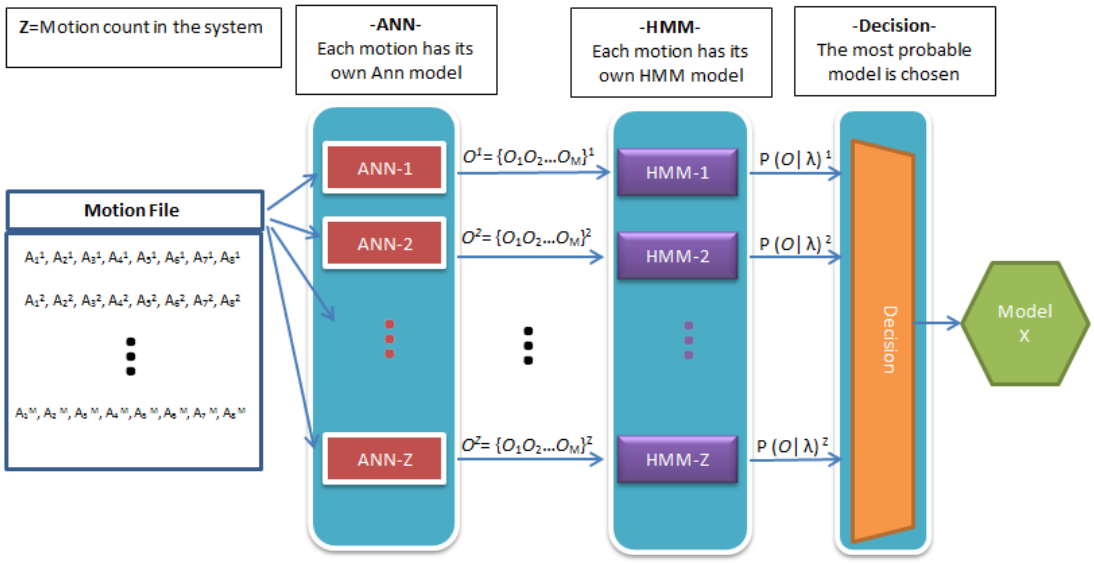

Figure 3.11: ANN with HMM Method Approach-1.

First approach has problems with discrimination of some gestures which are similar to each other. For example, when “up” gesture shown in Figure 3.10 comes to system, in order to find out corresponding gesture, the data is sent to the all ANN with HMM models as shown in Figure 3.11. Normally, “Up” gesture consists of 4 states. However, when the data of “Up” gesture is sent to the model of “side” gesture that is composed of 3 states, it assigns the lines of data related with fourth state to the third state. Because, the model of “side” gesture has no state named 4. As a result of this situation, there is no difference between the “up” gesture and “side” gesture and at the end, the examined data can be assigned to the “side” gesture instead of “up” gesture. This contradiction leads us to the new approach.

In the second approach, whole system has just one neural network. 11 states is specified for all gestures(Figure 3.13). The human body can stand in 11 distinct states along in all gestures. In this approach the HMM models includes the state sequence which consists of the global state point. In example “Dad” gesture is formed of “S_1,S_9,S_{10}” state sequence, “Spring” gesture is consist of “S_1,S_9,S_{10} S_{11}” and “Table” gesture is comprised “S_1,S_2,S_7 S_8” and these gestures are illustrated in Figure 3.14. The difference from the first approach is to train only one ANN instead of the number of ANN which is equal to the number of gestures.

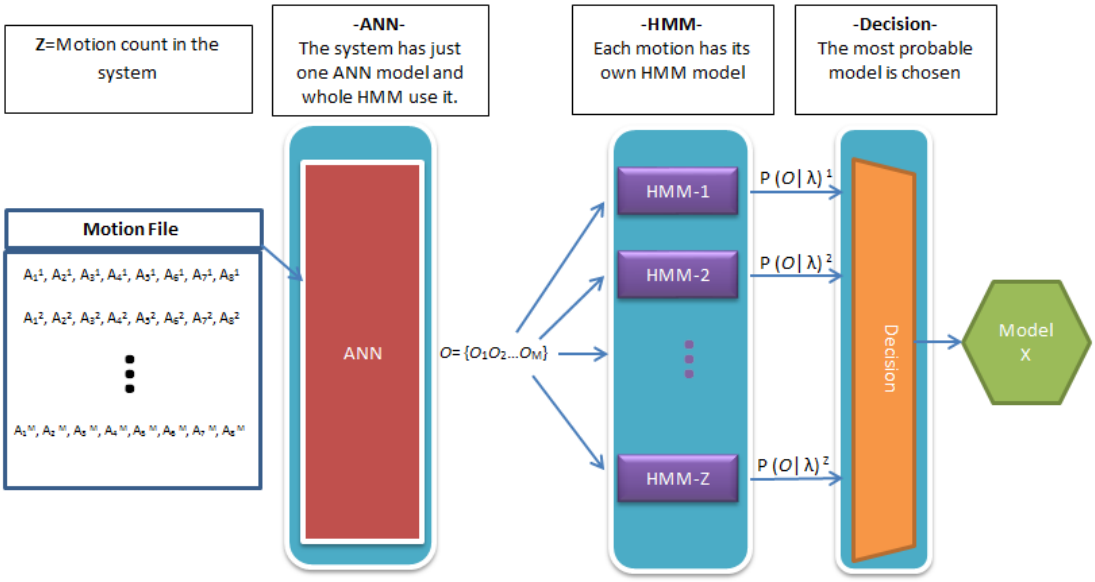

Figure 3.12 : ANN with HMM Approach-2.

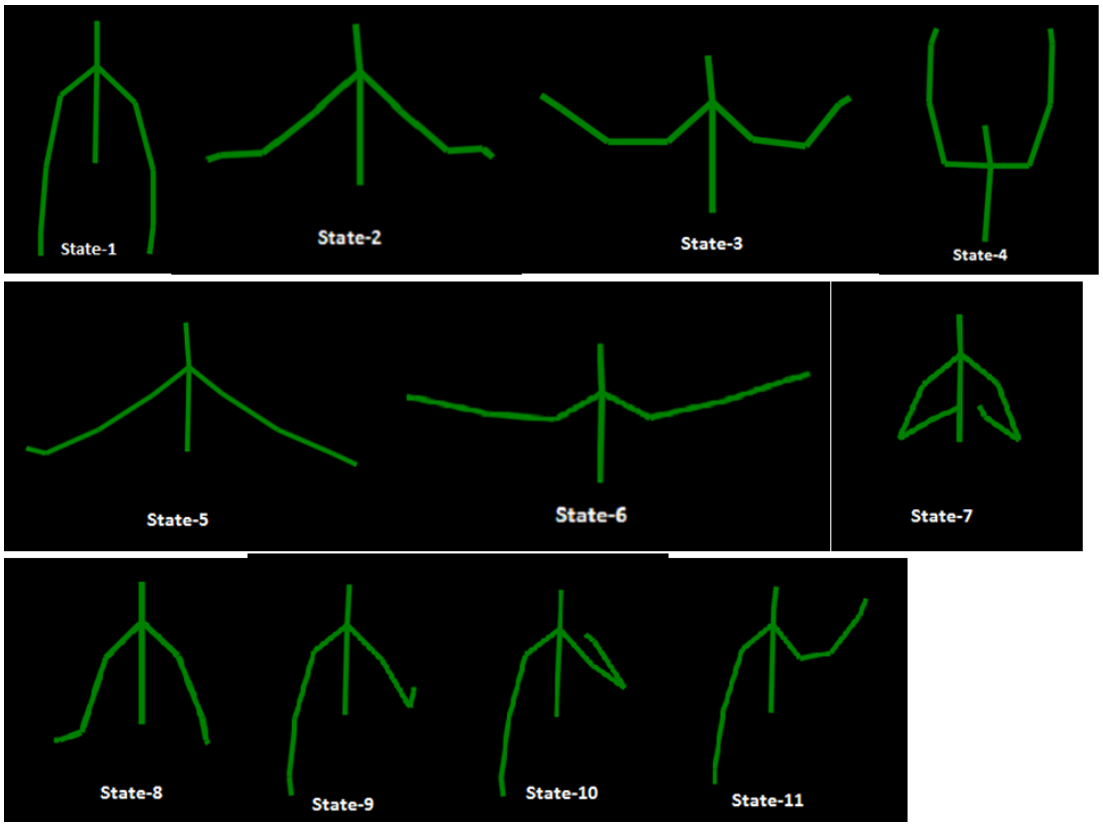

Figure 3.13: Approach-2 all states.

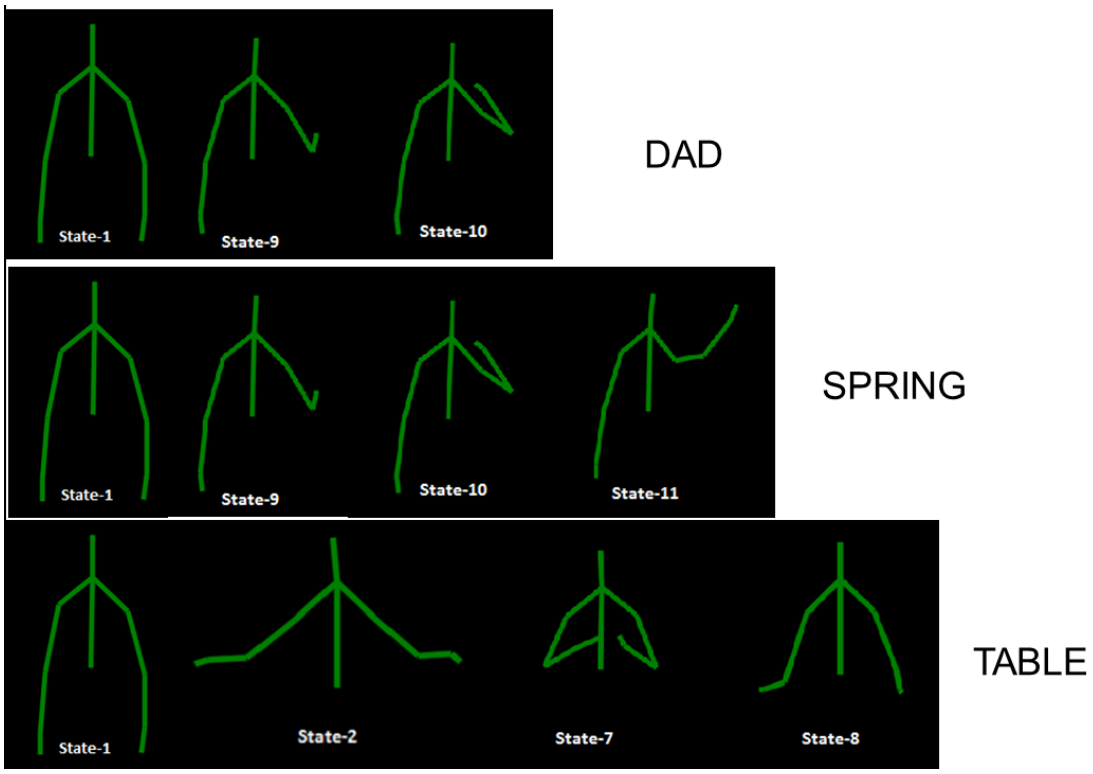

Figure 3.14: Approach-2 sample gesture.

3.6 Real-time Gesture Recognition

In the scope of this method, person-independent real time gesture recognition algorithm is proposed to recognize seven gestures, “Side”, “Table”, ”Forward”, “Small”, “Up”, ”Dad” and “Spring”.

The real time gesture recognition has some critical problems. One of the problems is the continuity problem. We want to calculate likelihood values between the pre-recorded gesture model in gesture dataset and the implemented gesture by test subject. If a test frame sequence is given to the system discretely, which is mentioned in the “Neural Network and HMM Method” section the method can calculate the probability easily. But in this case the length of the implemented gesture is unknown. To overcome this problem, window sliding method (Section 3.3) is used. According to this method, a window slides through the continuous data and the data under the window is sent to the related HMM model. The size of this window is specified as the slowest person gesture completion time. In this case the test subject could implement the gesture faster than this length and this irredundant part of the window results in bad performance and lower probability. In fact, the sliding window is not exact solution to our problem. The real solution for continuity problem is tracking the discontinuity of gesture trajectories. But in the proposed method, every gesture has same starting point so the separation point of gesture can be found out from this position.

Other problem is the start point detection of gestures. This problem is generally solved by the gesture trajectory discontinuity detection [28]. But in our case the gestures in our dataset have the same start positions. Based on this information, the detection of this specific start position solves our problem for this case.

The last problem is person-independent feature extraction. The Kinect tracks the joint and sends spatial coordinates for each joint. The spatial coordinates depends on the person’s body parts length. After the system training if a test subject, who has not training data in the system, comes for testing to the system, the system cannot recognize the gesture. To prevent this problem the transformation operation (Section 3.2) is applied to spatial coordinates.

In the proposed method, DTW (Section 2.4) is added to previous method (Neural Network and HMM Method section). As stated in the “Dynamic Time Warping”

section, the DTW is an algorithm computes an optimal matching between two temporal sequences. It calculates the distances between two sequences not directly recognizing the gesture. According to this expression we need two sequences to find matching score. One of these sequences is the test sequence which is coming continuously from Kinect, and we need one more sequence called reference sequence.

In the HMM, it is better as training sample size is bigger so the training quality will be bigger. Because HMM calculates some statistics from these samples. For example how many data sample corresponds to S1 state or how many data sample transits from S1 to S2. Unlike HMM, DTW has no training phase. In the DTW, there is no difference between 10000 training sample or 100 training sample. The DTW uses just one reference or template model. Because of this reason it uses one of the 10000 training sample.

The number of gesture count equals to the number of DTW model in the system. When a test sequence comes to the system, it is sent to each DTW model then the matching score is obtained from each one. As it is mentioned before, we need a reference sequence to compare this test sequence. Generally one of the gestures from dataset can be used as reference sequence but in the proposed method a dummy gesture is implemented.

As stated in the "Neural Network and HMM Method" section, the body can stand in 11 distinct states and the each gesture is comprised of some of these 11 states. These 11 states are illustrated in 6.12. Each state is recorded separately and saved distinctly. A reference frame constructor takes the state sequence for corresponding gesture and reads related state file respectively then constitutes an ideal reference sequence.

3.6.1 Combination of ANN-HMM method and DTW

In the proposed method, the frame sequence, which is taken from Kinect, is transformed to the angle sequence. Angle sequence is sent to both DTW models and ANN with HMM Models simultaneously. Both models calculate the matching scores and feed them to the formula below (3.8).

$$R_i = r_{m1}^i w_{m1} + r_{m2}^i w_{m2} \quad \textbf{(3.8)}$$

where R_i is the result of i-th gesture recognition, w_{m1} is the weight of DTW model, w_{m2} is the weight of ANN with HMM model, r_{m1}^i is the matching score of the DTW Model of the i-th gesture and r_{m2}^i is the matching score of the ANN with HMM Model i-th gesture.

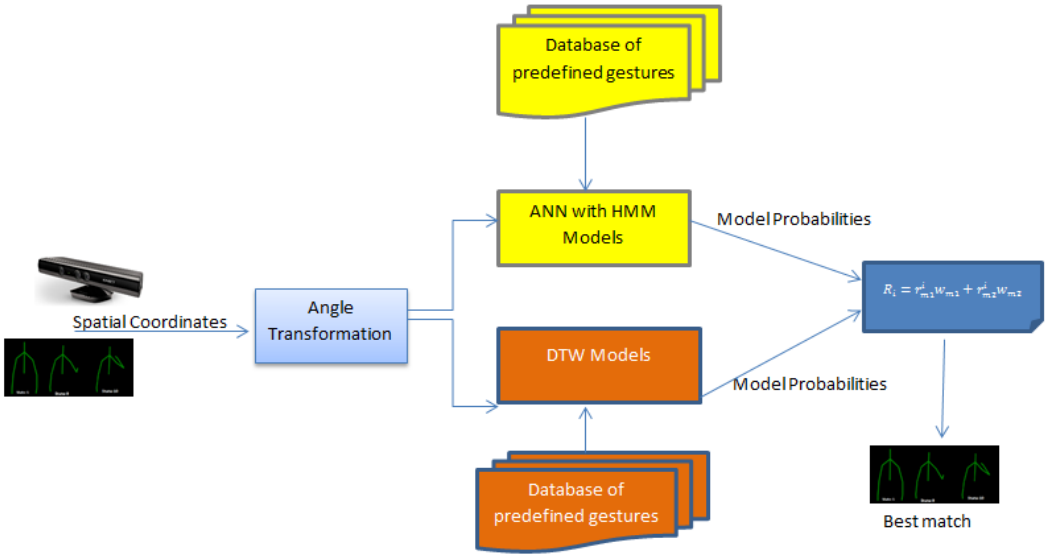

Figure 3.15 : Recognition of real time gesture.

Based on the validation set's results, $w_{m1}\ and\ w_{m2}$ weights are given to each method.

Consequently, the best matching gesture is chosen as the biggestR_i.

4. TEST AND RESULTS

4.1 Data Collection Methodology

The implemented actions were selected specially based on the advices of the therapists so that the tests can be also applied to the children with autism or mental disabilities. In addition, the actions, which can be implemented by the NAO robot (due to Kinematic constraints), were selected among these advised actions. In order to avoid boredom, confusion and fatigue in children during the tests, number of the actions and the repeats were set to a minimum, as possible. The actions were introduced to half of the participants (6 participants) using the real physical robot and the simulated robot, and by a real human to the rest of the participants. All of the participants were asked to stay on a special sign on the floor and implement the signs to make sure that they keep the same distance to the Kinect camera during the tests (1.5 mt). The participants were not given any information except the fact that they should follow the robot/human and imitate the actions afterwards, and their actions are recorded with the Kinect camera.

4.2 K-Means with HMM Method's Test and Results

In this part of thesis, 6 actions, namely, three basic sign language action (from TSL) and three basic upper torso actions from 15 university students were recorded with Kinect. For each action, each participant was asked to perform 4 times, of which two were used for training the system and two were used for testing.

Six Hidden Markov Models belonging to six signs were trained with 219 training sample and tested with 80 test samples. In this test, every class has different states and event numbers. The data, which belongs to these classes, was clustered corresponding to these event and state numbers. The confusion matrix exhibits the tests with these parameters are shown in the Table 1. These state and event numbers are the ones that give the best solution in the test trials. In the tests, 8 states and 10 events were used for "side", 8 states and 10 events were used for "forward", 7 states

and 9 events were used for “table”, 4 states and 5 events were used for “car”, 4 states and 6 events were used for “up”, 6 states and 9 events were used for “dad”.

Successfully matched classes were located in the diagonal of the table. 54 of the 80 test samples were classified successfully. When the unsuccessful groups are studied, it is observed that the groups with similar features cause these faults (similar actions).

For example “table” was recognized as “forward” 1 times, and “dad” and “up” signs were mislabeled. Another reason for the mislabeled signs is that some participants realized some actions wrong (not similar to the human/humanoid teacher), partly or as a whole.

Table 4.1 : Confusion Matrix.

		Predicted classification					
		Car	Dad	Forward	Table	Side	Up
Actual classification	Car	**13**	0	0	1	1	0
	Dad	0	**11**	1	0	1	1
	Forward	0	0	**8**	0	0	5
	Table	3	1	1	**7**	0	0
	Side	3	1	3	0	**6**	0
	Up	1	0	2	1	0	**9**

There were no statistically significant difference between the performance of the actions imitated from human teacher and humanoid teacher (simulated/physical).

Within the studies, one child with normal developments, and two children with ASD (all in the age group 6-7) participated, as well as the university students. The children did not stay stable during the recording of data, which makes it very hard to get data. The child with the normal development and one of the children with ASD could finish all the actions, yet not exactly same with the adults. Unfortunately, since they did not stay still, were tired, and leave the test before enough data was collected, the Kinect system could not be trained to recognize their actions. We are working on the necessary improvements to get data from children as fast and robust as possible. The system should be fast, and the time for calibration should be as short as possible.

4.3 ANN with HMM Method's Test and Results

In the scope of this method, seven gestures, "Side", "Table", "Forward", "Small", "Up", "Dad" and "Spring" are used.

Seven Hidden Markov Models were trained for the correspondent seven signs. Nearly 2000 lines are used in the neural network training. For the method's recognition test, 83 of test samples that belong to the different gestures of various people are used. This method tests have two main aspects; the neural network evaluation and method success rate evaluation. Firstly, ANN success rate is discussed then secondly the ANN with HMM method success rate.

As stated before, there are two approaches in the ANN with HMM method. In the first approach, a neural network is trained for each gesture. In the second approach only one neural network is trained for all gestures and it has 11 states. Approximately 2000 training samples (lines of data) are used for neural network trainings.

The average result for neural network in the second approach is given in Table 4.2. According to this table, the neural network success rate is 90.2314%.

Table 4.2 : Neural network confussion matrix sample.

		Predicted classification										
Actual classification		**a**	**b**	**c**	**d**	**e**	**f**	**g**	**h**	**i**	**J**	**k**
	State a	653	11	0	0	1	0	0	0	0	0	0
	State b	30	155	5	0	1	0	13	0	2	0	0
	State c	0	5	112	0	0	0	0	2	0	1	0
	State d	5	0	3	106	3	1	0	0	0	2	3
	State e	5	2	0	0	27	1	0	0	0	0	0
	State f	0	0	4	0	2	97	0	0	0	0	0
	State g	0	5	1	1	0	0	188	7	0	1	0
	State h	0	1	0	0	0	0	12	65	0	0	0
	State i	13	2	0	0	0	0	0	0	45	9	1
	State j	20	3	0	0	0	1	0	0	5	212	4
	State k	0	0	0	0	0	0	0	0	0	2	95

The recognition rate using second approach for this method is given in Table 4.3. According to this table, the method success rate is 85.5422%.

Table 4.3 : Best result's confusion matrix of Neural Network with HMM Method.

		Predicted classification						
		Side	Forward	Table	Small	Up	Dad	Spring
Actual classification	Side	12	0	0	0	0	0	0
	Forward	0	12	0	0	0	0	0
	Table	0	2	7	3	0	0	0
	Small	0	0	0	11	1	0	0
	Up	0	0	0	0	11	0	0
	Dad	0	0	0	0	0	10	2
	Spring	1	0	0	0	0	3	8

The accuracy of initial neural network layer is very important in this method. In the given results above, the neural network has 90% success rate. This means that when a line comes to the system, the neural network finds the correct states with 90% probability on other words there are still 10% wrong state assignment.

The HMM accuracy highly depends on the neural network layer accuracy. Due to misclassification and the left-to-right HMM's nature, some state transitions become zero. Zero transition occurrences results in lower success rate.

4.4 Real-time Gesture Recognition Method's Test and Results

This method consists of a combination of ANN with HMM model, and the DTW algorithm which is frequently used in motion recognition system. In the scope of this method, seven gestures, "Side", "Table", "Forward", "Small", "Up", "Dad" and "Spring" are used. The success rate of this method is given below in two-step. In the first step DTW success rate is discussed then secondly the real time gesture recognition success rate.

The success rate of DTW is 77.1% in the system. The confusion matrix of DTW algorithm is given in Table 4.4.

The Table 4.4 shows us, if only DTW were used in the system, the recognition rate for "Up" and "Spring" will be 100%, for "Side", "Forward" and "Table" gesture will be 91.6%, for "Small" will be 25%, and lastly for "Dad" will be 41.66%.

Table 4.4 : DTW confusion matrix results.

		Predicted classification						
		Side	Forward	Table	Small	Up	Dad	Spring
Actual classification	Side	11	1	0	0	0	0	0
	Forward	0	11	1	0	0	0	0
	Table	0	1	11	0	0	0	0
	Small	0	0	7	3	0	2	0
	Up	0	0	0	0	11	0	0
	Dad	0	0	0	2	0	5	5
	Spring	0	0	0	0	0	0	12

If this method is compared with the ANN with HMM method, the better results are obtained with the DTW for some gestures. However, for the rest of gestures, DTW performs worse than the ANN with HMM method. In the final method evaluation, this performance difference is taken into account. The weights are given to the gestures for each model(DTW and ANN with HMM), considering which methods recognize better the gesture the weight of this gesture for this method is given bigger than the other method.

The results are obtained from gesture implementation in front of Kinect by seven test subjects. In these tests, the static data which was used previous methods, is not used. The test subject is selected from the people whom data was not used in the training set.

Table 4.5 : Real time recognition results.

		Predicted classification						
		Side	Forward	Table	Small	Up	Dad	Spring
Actual classification	Side	7	0	0	0	0	0	0
	Forward	0	7	0	0	0	0	0
	Table	0	0	5	2	0	0	0
	Small	0	0	0	7	0	0	0
	Up	0	0	0	0	7	0	0
	Dad	0	0	0	0	0	7	0
	Spring	0	0	0	0	0	4	3

The system has 89.79% success rate and thus the DTW improves succeed rate of real time gesture recognition method.

The DTW has not a training phase, which is mentioned in the section “Real-time Gesture Recognition”, so the system has not any training cost. At the test phase of the real time gesture recognition method, the system can calculate the ANN with

HMM output and DTW output simultaneously so there is no time overhead to system.

4.5 Overall Results

The three method, which are implemented in the scope of this thesis, are K-Means with HMM, ANN with HMM, Real time Gesture Recognizer respectively. K-means with HMM has 67.5% success rate and it is the worst result in the system. With the purpose of improvement of this result, the ANN with HMM is developed and the result is enhanced to 85.54%. Lastly, the DTW added to the ANN with HMM model and the results raised to 89.79%.

Table 4.6 : Overall Results.

K-Means with HMM	ANN with HMM	DTW	DTW and ANN with HMM
67.5%	85.54%	77.1%	89.79%

5. A CASE STUDY: INTERACTIVE ISIGN GAME

Interactive Isign game is a study performed with autistic children as a part of a PhD course entitled as Autism and Computational Aspects, which aims to bring researchers in computer science and robotics with non-engineer experts from psychology, neurology, speech therapy, and autism therapists, to train students who want to work in this field, and for a long term brain storming event on possible computational solutions that can be used within autism therapy (recognition of autism and other related issues were not included in the course schedule due to time limitations). This thesis presents one of the projects, which are produced as an output of this collaboration, and it is planned to use the system and the game in the collaborative special schools on autism.

Autism Spectrum Disorder (ASD) involves communication impairments, limited social interaction, and limited imagination. Researchers are interested in using robots in treating children with ASD [46-50]. Many such children show interest in robots and find them engaging. Robots can facilitate interaction between the child and the teacher. Every child with autism has different needs. Robot behavior needs to be changed to accommodate individual children's needs and as each individual child makes progress.

The game will be based on the visual cards, the cards will be shown to the robot to select among several signs from ASL and basic upper torso motion (hands side, forward, up etc.) Then the robot will perform the sign and wait for the child to imitate. The imitated action will be evaluated using an RGB-D camera (Kinect) and robot will give a motivating comment when the action is imitated with success [51][52].

5.1 Technical Hardware

Kinect and Nao Humanoid robot which are explained below are used for the purpose of this game.

5.1.1 Microsoft kinect

Kinect [53] is a sensor device, which involves cameras, a microphone array, and an accelerometer as well as a software pipeline that processes color, depth, and skeleton data. It is shown in Figure 5.1 and Figure 5.2.

Figure 5.1 : Kinect [53].

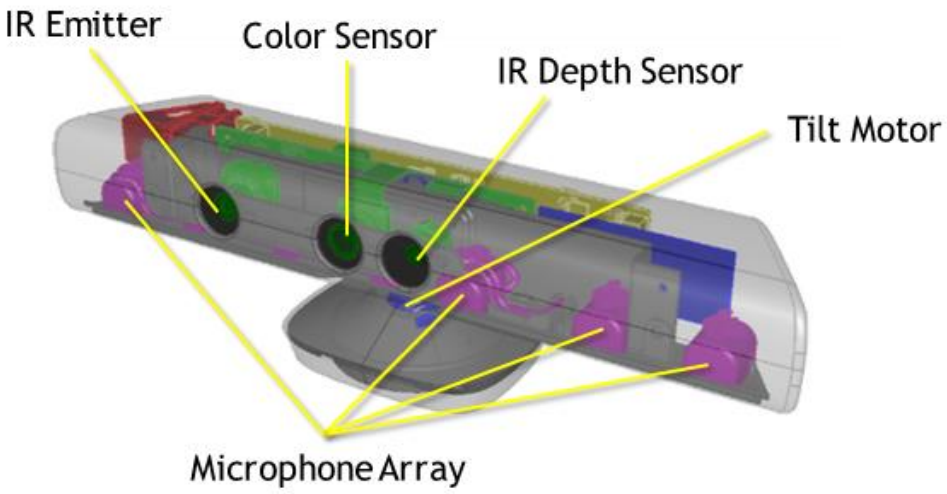

Figure 5.2 : Kinect Sensors [53].

Its depth and color stream frame rate is 30 frames per second (FPS). Kinect RGB camera can capture three channel data in 1280x960 resolutions [53]. It has an infrared (IR) emitter and an IR depth sensor. The depth information obtained via emitter and depth sensor. Emitter emits infrared light beams and the depth sensor reads the depth information from that beams. It has four microphones and it uses them to find the location of sound and direction of the audio wave besides the recording sounds. Kinect tracks 20 joints in the body and joint spatial coordinates (x, y, and z) of skeletal structure for each joint are generated. This is shown in Figure 5.3.

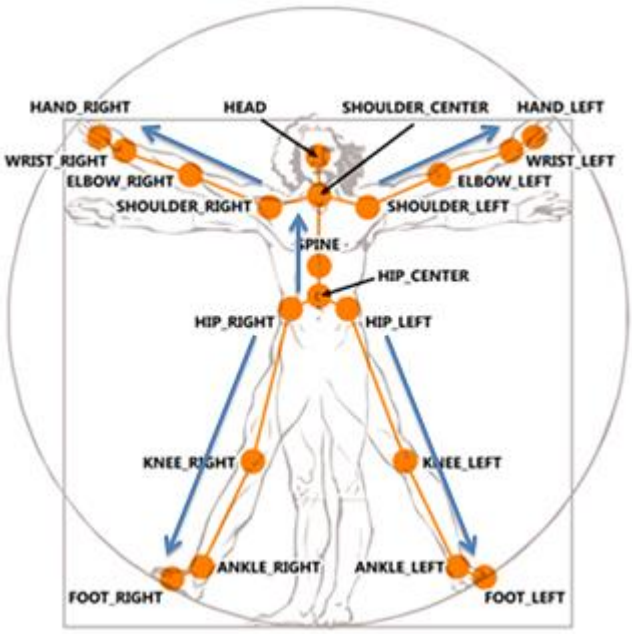

Figure 5.3 : Skeleton Position and Tracking State [53].

5.1.2 NAO humanoid robot

In this thesis, the goal is to empower teacher that works with children with ASD and to customize robot behavior to suit the needs of each child. NAO H-25 humanoid robot is used during the field studies. Since NAO is a small size humanoid robot, it is suitable to implement basic signs in the ASL and TSL, it is also robust and safe to work with children. For further studies, a bigger size humanoid robot platform with five fingers and more DOF on arms will be used within this thesis.

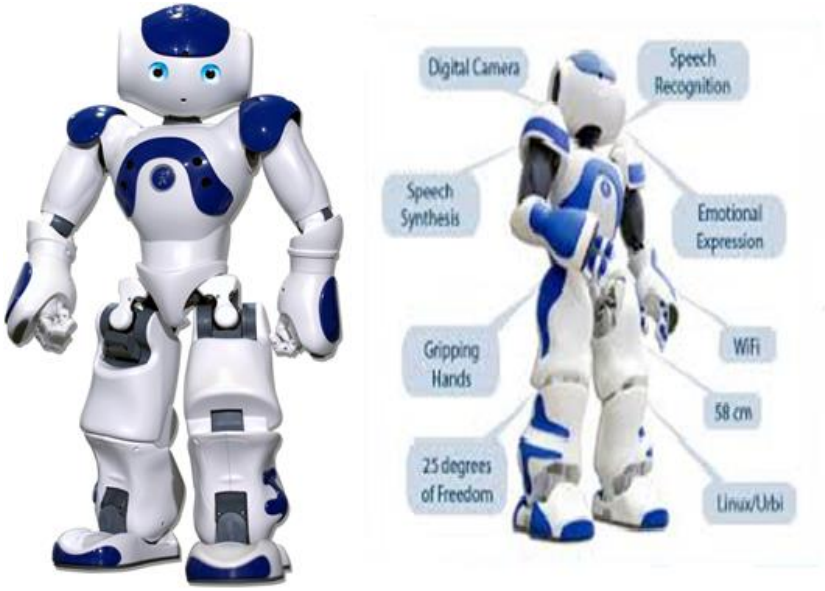

Figure 5.4 : Nao H-25 Robot [54].

Aldebaran Robotics offers several software tools for use with the NAO robot. Choregraphe can be used for face detection, face recognition, speech, speech recognition, walking detection, recognizing special marks and dances, and individual control of the robot's joints. The movements can be performed in sequence or in parallel. Choregraphe needs to be used with a robot proxy, real or simulated.

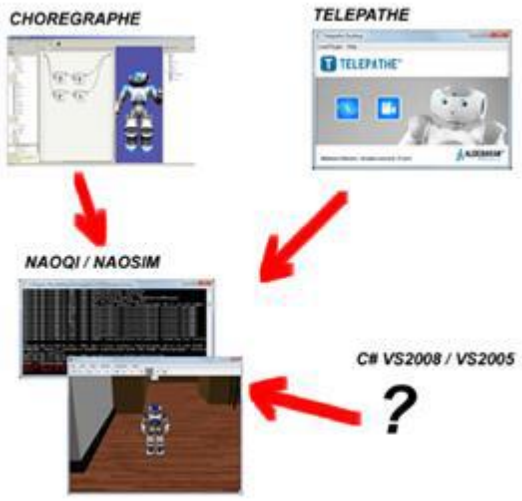

Figure 5.5 : Interaction of Aldebaran software for the NAO robot.

The simulated proxy can be NAOqi or a sophisticated simulator such as NaoSim [55]. NAOqi is a piece of software that simulates the robot for Choreographe and tests it before trying on the actual robot. NAOsim is simulation tool that allows for robot simulation in an apartment having simulated furniture. Whereas NAOqi only simulates the robot, NAOsim simulates an environment with which the robot can interact. Monitor, which was called Telepathe until the current release, allows the user to access the robots memory, see through the robots two cameras and observe the environment as the robot senses it. In addition, it is possible to use some of program languages, Python or C++ to program the NAO.

During the implementation phase, different methods for the kinematic modeling of the humanoid robots are available as [42].

5.2 Game Flow

As stated before, as an outcome of the PhD level course on Autism and Computational Aspects, an imitation game was constructed using Kinect and Nao H25, based on the recommendations of autism therapists. Our main goal was to

extent our studies on sign based interaction games with humanoid robots for autism therapy.

Figure 5.6 : “up”, and “side” actions.

The sign imitation game is an extension of the sign language game. As an initial step, we used basic upper torso gestures, i.e. opening the arms sides, up, forward, waving hand, etc., in the long run, we plan to use signs from TSL as well. The aim of the game is to teach the children to recognize and imitate the gestures/signs, within a turn-taking interaction game. The demonstrator will be the robot and the therapist will be able to assist the child manually, when the child fails to imitate the action successfully. Within this game, it is possible to locate many of the exercises already being used as a part of the autism therapy.

The game consists of three stages. In the first stage, the child will learn how to play doing the gestures one by one, the sequence and the quality of the gestures were chosen by the therapist or the child. When they show a picture of the gesture to the robot, the robot does the gesture and waits for the child to repeat the action. Using a Kinect camera, we can evaluate child’s actions and send the robot feedback. If the child can repeat the action then robot says, “You did the action good” (The experts suggested us that we have to praise the action of the child, it is not enough to say “it’s good” or “congratulations”). Else, the therapist helps the child manually to do the gesture.

In the second stage, the game is like a sports work out, each action/gesture is repeated several times without the picture display and the therapist gets involved less. The child is assumed to learn each action by then.

In the third stage, we turn the game into a musical play. Robot sings a song related to the actions and does the actions one by one and the child is expected to repeat the sequence of actions.

Figure 5.7 : Therapists help children in the first stage of the game.

The robot will record the success rate of the child and the experimenter will record the therapist's corrections and child's success.

These games were usually played with the therapist, or the video of the therapist, or another autistic child. The robot will act as a playmate in these games.

5.3 Bag of Visual Words

Cards are recognized in Isign game by Bag of Visual Words Method.

To navigate through the exercises, the instructor used a set of cards with different images. Each image represents a different exercise. This requires image recognition software which was coded using C++ and OpenCV. The algorithms used in this project included SURF[56] feature detection, Bag-of-Words, K-means clustering and support vector machines (SVM).

OpenCV has many algorithms to detect and describe local feature. SURF was chosen for feature extraction and object detection. This is one of the most common methods.

The Bag of words (BoW) approach is more commonly used in natural language processing [57] [58]. When used for image processing, an image and its graphical

features are analogous to the document and to the words, respectively. After image representations had been obtained with BoW, SVM supervised learning was used for classification [59]. SVM takes an input array which consists of data and a label. The label represents the class to which data belongs. There are many hyperplanes that could be used to classify the data sets. The best hyperplane is the one that gives the greatest separation or margin between the classes.

The training database consisted of a scanned set of Walt Disney cartoon characters printed on cards. Every card was identified by a number which became the SVM label for that card. Training began by extracting the features from the images and computing image descriptors:

Then, the dictionary of graphical features was determined with K-means clustering as shown below. The result set of BowKmeansTrainer was written to file in YAML format. That improves speed of image recognition. The last step in training was training the SVM. CvSVM, an implementation included in OpenCV was used. The SVM training results were also written in a file to improve performance.

Once training was complete, the cartoon characters printed on the cards can be recognised by simply detecting the features of the images and sending them to the SVM for prediction.

For current techniques, training set must be extended and more experiments must be done. If a picture is texture based, and has a lot of key points, our approach gives a very successful result. The database in the iSign game consisted of about seven pictures with distinctive characteristics (Disney cartoon characters), and the collected test set accuracy was very high. This algorithm is used in a user test with with 30 hearing impaired children of 7-16 years old, with a different set of 9 colored flashcards in 2014, where every child showed 3 cards to the robot to be recognized, and only at one instance the card was not recognized.

5.4 Conclusion

This game was implemented and demonstrated in pilot studies in a special rehabilitation center for ASD [51,52]. In 2014, an updated version of the game will be tested in the collaborative special schools of hearing-impaired children.

6. CONCLUSION

This thesis is part of an ongoing project, which aims to teach sign language with imitation based turn taking games by the help of humanoid robots, which are significantly developed in the recent years. In this work we focused on the children who have communication disorders including hearing impaired children and children with ASD. A NAO H25 humanoid robot helps to human teacher to teach some signs and basic upper torso actions, which are observed and imitated by the participants. Besides the Nao robots cameras, RGBD sensor camera (Kinect) is used in the user tests.

The system, which developed in the scope of this thesis, is focused on the gesture/sign recognition. With the use of Kinect, the upper torso information of the human participants is gathered and using the proposed methods, it is recognized in real-time so the robot is able to give feedback to the participants about their actions.

We propose three methods to recognize the signs/gestures namely; K-Means with Hidden Markov Model (HMM), Neural Networks (NN) with Hidden Markov Model and Real time Gesture Recognition. All of these methods are based on the Hidden Markov Model. The distinction of these methods is how they represent observed human motion states.

In the first method named as K-Means with HMM, frames are transformed into observation sequence by means of k-means method. In the second method (HMM with NN), generation of observation sequence is handled by a neural network. Therefore, HMM takes the emission probabilities from this neural network module. In the third method, the DTW algorithm is added to ANN with HMM method.

In the first method, Baumwelch training algorithm is used for HMM training and it is not cost effective. The training time of the second method is better than the first one, while in the test step, the first method is faster than second method. In terms of effectiveness, if the data for NN is labeled correctly, and an optimum parameter set is

selected (hidden layer count and node counts per layers), the second method has higher success rates than the first method.

In the third method the ANN with HMM method and the DTW results are combined then the result of system is empowered.

As a summary, we made two main contributions in this project. One of them is using angle values of the joint points of the human body. The other one is creating a new method to recognize the sign language gesture by combining the Dynamic Time Warping, Artificial Neural Network with Hidden Markov Model to increase the recognition accuracy.

In addition to these contributions, to the best of our knowledge there is no other work on sign language tutoring by robots (especially for children), in the literature. Plus this work aims to fill this gap by supporting the hearing impaired people as a social responsibility project.

In the future, we would like to combine RGB data with depth data to increase the success rate. In addition, facial expressions could be integrated to distinguish similar looking signs.

REFERENCES

[1] **Staner, A. T., and A. Pentland**, "Real-Time American Sign Language Recognition from Video using Hidden Markov Models", Technical Report TR-306, Media Lab, MIT.

[2] **Kadous, W.** (1995). "Grasp: Recognition of Australian sign language using instrumented gloves" MSc. Thesis, University of New South Wales

[3] **Murakami, K., & Taguchi, H**. (1991). Gesture recognition using recurrent neural networks. In Proceedings of the SIGCHI conference on Human factors in computing systems (pp. 237-242). ACM.

[4] **Aran, O., & Akarun, L.** (2010). A multi-class classification strategy for Fisher scores: Application to signer independent sign language recognition. Pattern Recognition, 43(5), 1776-1788.

[5] **Aran, O., Ari, I., Akarun, L., Dikici, E., Parlak, S., Saraclar, M., Campr,P.,& Hruz, M.** (2008). Speech and sliding text aided sign retrieval from hearing impaired sign news videos. Journal on Multimodal User Interfaces, 2(2), 117-131.

[6] **Caplier, A., Stillittano, S., Aran, O., Akarun, L., Bailly, G., Beautemps, D., Aboutabit,N., & Burger, T.** (2007). Image and video for hearing impaired people. EURASIP Journal on Image and Video Processing, Special Issue on Image and Video Processing for Disability

[7] **Jaffe, D. L.** (1994). Evolution of mechanical fingerspelling hands for people who are deaf-blind. Journal of rehabilitation research and development, 31(3), 236-244.

[8] **Hersh, M. A., & Johnson, M. A. (Eds.)**. (2003). Assistive technology for the hearing-impaired, deaf and deafblind.Springer. XIX, 319 p. 168 illus., Hardcover, ISBN: 978-1-85233-382-9.

[9] **Adamo-Villani, N.** (2006). A virtual learning environment for deaf children: design and evaluation. IJASET–International Journal of Applied Science, Engineering, and Technology, 16, 18-23.

[10] **Lee, S., Henderson, V., Hamilton, H., Starner, T., Brashear, H., & Hamilton, S.** (2005, April). A gesture-based American Sign Language game for deaf children. In CHI'05 Extended Abstracts on Human Factors in Computing Systems (pp. 1589-1592). ACM.

[11] **Greenbacker, C., and K. McCoy.** (2008, Feb) The ICICLE Project: An Overiew. First Annual Computer Science Research Day, Department of Computer & Information Sciences, University of Delaware.

[12] **Shen, Q., Kose-Bagci, H., Saunders, J., & Dautenhahn, K.** (2009, September). An experimental investigation of interference effects in human-humanoid interaction games. In Robot and Human Interactive

Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on (pp. 291-298). IEEE.

[13] **Kose-Bagci, H., Dautenhahn, K., & Nehaniv, C. L.** (2008, August). Emergent dynamics of turn-taking interaction in drumming games with a humanoid robot. In Robot and Human Interactive Communication, 2008. RO-MAN 2008. The 17th IEEE International Symposium on (pp. 346-353). IEEE.

[14] **Kose-Bagci, H., Ferrari, E., Dautenhahn, K., Syrdal, D. S., & Nehaniv, C. L.** (2009). Effects of embodiment and gestures on social interaction in drumming games with a humanoid robot. Special issue on Robot and Human Interactive Communication, Advanced Robotics vol.24, no.14.

[15] **Kose-Bagci, H., Dautenhahn, K., Syrdal, D. S., & Nehaniv, C. L.** (2010). Drum-mate: interaction dynamics and gestures in human–humanoid drumming experiments. Connection Science, 22(2), 103-134.

[16] **Dautenhahn, K., Nehaniv, C. L., Walters, M. L., Robins, B., Kose-Bagci, H., Mirza, N. A., & Blow, M.** (2009). KASPAR–a minimally expressive humanoid robot for human–robot interaction research. Applied Bionics and Biomechanics, 6(3-4), 369-397.

[17] **Zufferey, D.** (2009). Device based gesture recognition. In Written as part of a master seminar about gesture recognition. Department of Informatics (DIUF), University of Fribourg, Fribourg, Switzerland.

[18] **Du, W., & Li, H.** (2000). Vision based gesture recognition system with single camera. In Signal Processing Proceedings, 2000. WCCC-ICSP 2000. 5th International Conference on (Vol. 2, pp. 1351-1357). IEEE.

[19] **Schlömer, T., Poppinga, B., Henze, N., & Boll, S.** (2008, February). Gesture recognition with a Wii controller. In Proceedings of the 2nd international conference on Tangible and embedded interaction (pp. 11-14). ACM.

[20] **Wang, Y., Yang, C., Wu, X., Xu, S., & Li, H.** (2012, August). Kinect based dynamic hand gesture recognition algorithm research. In Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on (Vol. 1, pp. 274-279). IEEE.

[21] **Zafrulla, Z., Brashear, H., Starner, T., Hamilton, H., & Presti, P.** (2011, November). American sign language recognition with the kinect. In Proceedings of the 13th international conference on multimodal interfaces (pp. 279-286). ACM.

[22] **Memiş, A., & Albayrak, S.** (2013, December). A Kinect based sign language recognition system using spatio-temporal features. In Sixth International Conference on Machine Vision (ICMV 13) (pp. 90670X-90670X). International Society for Optics and Photonics.

[23] **Reyes, M., Dominguez, G., & Escalera, S.** (2011, November). Feature weighting in dynamic time warping for gesture recognition in depth data. In Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on (pp. 1182-1188). IEEE.

[24] **Gehrig, D., Kuehne, H., Woerner, A., & Schultz, T.** (2009, December). Hmm-based human motion recognition with optical flow data. In Humanoid Robots, 2009. Humanoids 2009. 9th IEEE-RAS International Conference on (pp. 425-430). IEEE.

[25] **Starner, T., & Pentland, A.** (1997). Real-time american sign language recognition from video using hidden markov models. In Motion-Based Recognition (pp. 227-243). Springer Netherlands.

[26] **Wilson, A. D., & Bobick, A. F.** (1999). Parametric hidden markov models for gesture recognition. Pattern Analysis and Machine Intelligence, IEEE Transactions on, 21(9), 884-900.

[27] **Aran, O., & Akarun, L.** (2007, June). Sign Language Processing and Interactive Tools for Sign Language Education. In Signal Processing and Communications Applications, 2007. SIU 2007. IEEE 15th (pp. 1-4). IEEE.

[28] **Liang, R. H., & Ouhyoung, M.** (1998, April). A real-time continuous gesture recognition system for sign language. In Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on (pp. 558-567). IEEE.

[29] **Yamato, J., Ohya, J., & Ishii, K.** (1992, June). Recognizing human action in time-sequential images using hidden markov model. In Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on (pp. 379-385). IEEE.

[30] **Starner, T. E.** (1995). Visual Recognition of American Sign Language Using Hidden Markov Models. MIT Cambridge Dept. of Brain and Cognitive Sciences.

[31] **Carmona J.M, Climent J**, (2012). "A Performance Evaluation of HMM and DTW for Gesture Recognition", Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications Lecture Notes in Computer Science Volume 7441, pp 236-243 ISBN: 978-3-642-33275-3

[32**] MacQueen, J. B.** (1967). "Some Methods for classification and Analysis of Multivariate Observations". Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability 1. University of California Press. pp. 281–297. MR 0214227. Zbl 0214.46201. Retrieved 2009-04-07.

[33] **URL-1** <http://en.wikipedia.org/wiki/K-means_clustering#cite_note-macqueen1967-1>, date retrieved 29.05.2014

[34] **URL-2** <http://www.csd.uwo.ca/faculty/olga/Courses/CS434a_541a/Lecture15.pdf >, date retrieved 29.05.2014

[35] **Bishop,C.M** (2006).Pattern Recognition and Machine Learning, Cambridge U.K

[36] **Alpaydin E**.(2010) , Introduction to Machine Learning , Second edition, London, England

[37] **URL-3** <http://en.wikipedia.org/wiki/Stochastic_process>, date retrieved 29.05.2014

[38] **Rabiner, L.** (1989). A tutorial on hidden Markov models and selected applications in speech recognition. Proceedings of the IEEE, 77(2), 257-286. doi:10.1109/5.18626.

[39] **Rabiner, L., & Juang, B. H.** (1986). An introduction to hidden Markov models. ASSP Magazine, IEEE, 3(1), 4-16.

[40] **URL-4** <https://edit.ethz.ch/ife.ee/education/wearable_systems_1/>, date retrieved: 29.05.2014

[41] **Sakoe, H., & Chiba, S.** (1978). Dynamic programming algorithm optimization for spoken word recognition. Acoustics, Speech and Signal Processing, IEEE Transactions on, 26(1), 43-49.

[42] **Ertugrul, B.S.,Gurpinar, C. , Kivrak, H. , Kulaglic, A., Kose, H.**,(2013) "Gesture Recognition for Humanoid Assisted Interactive Sign Language Tutoring", Advanced in Computer-Human Interactions (ACHI).

[43] **URL-5 <**http://en.wikipedia.org/wiki/Gesture_recognition>, date retrieved: 29.05.2014

[44] **URL-6 <**http://en.wikipedia.org/wiki/Feature_extraction>, date retrieved: 29.05.2014

[45] **URL-7** <https://community.aldebaran-robotics.com/doc/1-12/nao/hardware/product-range.html>, date retrieved: 29.05.2014

[46] **Feil-Seifer, D. J., M. P. Black, E. Flores, A. B. St. Clair,E. K. Mower, C. Lee, M. J. Mataric, S. Narayanan, C. Lajonchere, P. Mundy, and M. Williams**. (2009). Development of socially assistive robots for children with autism spectrum disorders. Technical Report CRES-09-001, USC Interaction Lab, Los Angeles, CA.

[47] **Goodrich, M. A., Colton, M. A., Brinton, B., & Fujiki, M.** (2011, March). A case for low-dose robotics in autism therapy. In Proceedings of the 6th international conference on Human-robot interaction (pp. 143-144). ACM.

[48] **Billard A., Robins B., Nadel J., Dautenhahn K.** Building Robota, a Mini-Humaoid Robot for the Rehabilitation of Children with Autism. RESNA Assistive Technology Journal, vol.19, 2006

[49] **Dautenhahn K.** (2003) Roles and Functions of Robots in Human Society - Implications from Research in Autism Therapy. Robotica 21(4), pp. 443-452.

[50] **Kozima, H., Nakagawa, C., & Yasuda, Y.** (2007). Children–robot interaction: a pilot study in autism therapy. Progress in Brain Research, 164, 385-400.

[51] **Ertugrul, B. S., Kivrak, H., Daglarli, E., Kulaglic, A., Tekelioglu, A., Kavak, S., Ozkul,A., Yorganci, R. & Kose, H.** (2012, October). iSign: Interaction Games for Humanoid Assisted Sign Language Tutoring. In International Workshop on Human-Agent Interaction (iHAI 2012) (Vol. 11).

[52] **Kivrak, H., Ertugrul, B. S., Yorgancı, R., Daglarli, E., Kulaglic, A., & Kose, H.** (2012, October). Humanoid Assisted Sign Language Tutoring. In 5th International Workshop on Human-Friendly Robotics, Brussels.

[53] **URL-8** <http://msdn.microsoft.com/en-us/library/hh855347.aspx>, date retrieved: 29.05.2014

[54] **URL-9** <http://www.robotics.com.hk/index.php?option=com_wrapper&Itemid=122> , date retrieved: 29.05.2014

[55] **URL-10** <http://www.aldebaran-robotics.com/en/>, Available: 29.05.2014

[56] **Schmitt, D., & McCoy, N**. (2011). Object classification and localization using SURF descriptors.

[57] **Tirilly, P., Claveau, V., & Gros, P**. (2008, July). Language modeling for bag-of-visual words image categorization. In Proceedings of the 2008 international conference on Content-based image and video retrieval (pp. 249-258). ACM.

[58] **Sivic, J., Schaffalitzky, F., & Zisserman, A**. (2004). Efficient object retrieval from videos. na.

[59] **Cortes, C., & Vapnik, V**. (1995). Support-vector networks. Machine learning, 20(3), 273-297.

CURRICULUM VITAE

Name Surname: Bekir Sıtkı ERTUĞRUL

Place and Date of Birth: Sandıklı/1985

E-Mail: bsertugrul@gmail.com

B.Sc.: Ege University Computer Engineering 2010

PUBLICATIONS/PRESENTATIONS ON THE THESIS

- **Ertugrul, B.S.,** C. Gurpinar, H. Kivrak, A. Kulaglic, H. Kose, "Gesture Recognition for Humanoid Assisted Interactive Sign Language Tutoring", The Sixth International Conference on Advances in Computer-Human Interactions (ACHI 2013), February 24 - March 1, 2013 - Nice, France (presented by Bekir S. Ertuğrul).
- **Ertugrul, B.S.,** H. Kivrak, E. Daglarli, A. Kulaglic, A. Tekelioglu, S. Kavak, A. Ozkul, R. Yorgancı, H. Kose, "iSign: Interaction Games for Humanoid Assisted Sign Language Tutoring", International Workshop on Human-Agent Interaction (iHAI 2012), 11 October, 2012, Vilamoura, Algarve, Portugal
- H. Kivrak, **Ertugrul, B.S.,** R. Yorgancı , E. Daglarli, A. Kulaglic, H. Kose, "Humanoid Assisted Sign Language Tutoring", 5th Workshop on Human-Friendly Robotics (HFR 2012), October, 2012, Brussels,
- **Ertugrul, B.S.,** C. Gurpinar, H. Kivrak, H. Kose, Insansı Robot destekli Interaktif İşaret Dili Eğitimi için İşaret Tanıma, 21. Sinyal İşleme ve İletişim Uygulamaları (SİU) Kurultayı, 24-26 Nisan 2013, Girne, KKTC, pp.1-4, 2013
- Kose, H., N. Akalin, R. Yorganci, **B. S. Ertugrul**, H. Kivrak, S. Kavak, A. Ozkul, C. Gurpinar, P. Uluer ve G. İnce, iSign: Humanoid Assisted Sign Language Tutoring for Children, Intelligent Assistive Robots-Recent advances in assistive robotics for everyday activities,The Springer Tracts in Advanced Robotics (STAR), *in press,* 2014