MOTION LEARNING AND CONTROL FOR SOCIAL ROBOTS
IN HUMAN-ROBOT INTERACTION

by

NAMRATA BALAKRISHNAN

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

MASTER OF SCIENCE IN ELECTRICAL ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2015

Acknowledgements

I would like to start by saying how deeply indebted I am to my supervising professor Dr. Dan O. Popa for giving me the opportunity to undertake the project. I am grateful to him for constantly encouraging and motivating me towards my goals. It would not be right if I were to forget to thank the University of Texas at Arlington for providing the enriching environment to successfully carry out the project. I wish to serve my gratitude to Dr. Michael Manry and Dr. Frank Lewis for taking the time off their schedule in order to serve in my defense committee.

I thank faculty associate researcher Dr. Indika Wijayasinghe for his guidance. I also wish to thank all of my colleagues at NGS especially Isura Ranatunga, Sumit Das, Dr. Corina Bogdan, Abhishek Thakurdesai, Fahad Mirza and Sandesh Gowda for the constant support. I cannot forget to thank Ranjani Muthukrishnan for helping with the basics of Data Analysis and Joe Sanford for reviewing my thesis document.

My most profound thanks go to my parents who have not only constantly encouraged me in the pursuit of my studies, but have also provided invaluable help by inspiring me.

April 17, 2015

Abstract

MOTION LEARNING AND CONTROL OF SOCIAL ROBOTS IN HUMAN ROBOT INTERACTION

Namrata Balakrishnan, MS

The University of Texas at Arlington, 2015

Supervising Professor: Dan O. Popa

In the domain of social robotics, robots have recently been used in conversational interaction with humans. In this thesis, research was conducted to help create a system for imitation learning. In this system, a trainer trains a robot to be a teacher. The robotic teacher interacts with other humans in order to teach them the task that the robot was trained on. The method of ‘Teaching by Demonstration’ was used, where an ideal motion is performed by a trainer. This ideal motion is learned by the robot and replayed in the subsequent interaction with humans. If the replayed motion is copied by the human, the closeness of the motion performed by the human and the robot are compared. The task is then repeated until the desired optimal motion (as performed by the trainer) is obtained from the human subject. The main focus of the thesis is to define a general imitation system that can encode different motions which are beneficial in social robotics. A technique called Dynamic Movement Primitives (DMPs) was selected as the method for recording and generating the generalized robotic motions. The human-robot interaction gestures are compared using another algorithm called Dynamic Time Warping (DTW) and the validity of DTW as a comparison metric was also studied.

DMPs are a set of non-linear differential equations which are used as the framework for describing human motion in a generalizable manner. A motion can be expressed as a combination of the learnt movement primitives. The DMPs have the flexibility to encode any motion into a set of differential equations by just adjusting certain parameters. The task/ motion that the robot is to teach a human subject is learnt using the DMPs.

Once the motion is taught to the human subject, the gesture performed by the subject and the motion executed by the robot are juxtaposed and analyzed using the DTW method. DTW is an algorithm which analyzes motion series that change temporally. Similarities between the gestures performed by the robot and the imitation done by the human are studied using DTW. Trials are performed to validate the utility of DTW as an effective measure for comparison.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

1.1. Motivation for motion learning related to Human Robot Interaction

Interactive robotics is a major research area which involves Human-Robot Interaction (HRI) enabled by Artificial Intelligence, Control theory and Mechanical innovation [1]. On further inspection, it can be seen that HRI involves machine learning [2], robotic vision [3] and embedded real-time control [4]. HRI finds utility in an extensive range of applications like entertainment, education, and rehabilitation therapy, among other fields. The current research focused on these applications is aimed towards a general betterment of the society.

Active research in the area of robot assisted therapy is on the rise. Using robots for rehabilitation, where the robot is used as a therapy device, has been reported by various research groups [5] [6] [7]. Traditionally these assistive robots were used for assisting patients to overcome their motor impairments. Recently there has been progress on using robots for assistance with cognitive impairments, such as Autism therapy [8] [9] [10]. Autism is a disorder which impairs social interactions and inhibits sensory development as well as verbal communication [11] [12]. It is a neurodevelopmental disorder which becomes more prominent around the age of 2- 3 years [13]. The Autism Spectrum Disorders represents three disorders of which Autism Disorder is one[14]. It is also characterized by repetitive behavior in daily activities and is associated with weakness in motor skills like use of fingers for grasping objects [15] [16]. Therapy sessions involving robots are useful as autistic individuals, especially children, are attracted towards robots [17]. This inclination is used as a motivation to begin introducing robots for therapy with the autistic individuals who generally refrain from social interactions.

Currently subjective judgments of physicians are used for the diagnosis of Autism. Behavioral components of Autism are one of the main criterion used for the diagnosis [18]. There has been a rise in the use of robotics for Autism therapy [17]. But there has not been any objective criterion for a reliable indication of childhood autism. The need for a quantitative tool for the purpose of diagnosis and treatment of Autism has been a huge motivation factor for this thesis.

For a robot to teach a human any action, the robot has to be taught the action first. As the basic research in robotic development is often application specific, a robot is generally programmed separately for separate tasks. In the domain of service and social robotics, interactions between humans and robots are high and the interaction environment is not constant [19]. With the changes in the surroundings, repeated changing of the program, to suit the task at hand, can be futile.

Thus, a system for imitation learning is necessary where tasks can be generalized and encoded onto a pre-set structure. This pre-set structure must define the tasks that are to be performed by the robot, during its interaction with the human, so that the concept of task specific re-programming can be eliminated [19]. The prime inspiration for conducting this research was to facilitate a system where a robot learns a task and teaches the motion to a human and subsequently adapts itself.

1.2. Challenges involved in Human Robot Interaction

At the beginning of any Human Robot Interaction (HRI) three aspects – recognition of the human, realization of the surroundings and definition of robot`s kinematics – must be considered. Once this is done, the response of the human to the robot is to be noted and then further actions are to be taken. Such interactions involve numerous challenges ranging from object perception to motion planning and then re planning according to any changes. Currently, performing a gesture, analyzing a gesture and then replaying the changed gesture using a robot requires tedious manual programming. New systems which need not be manually tuned every time are required.

Playing a human like motion on a robot has many challenges. Understanding and replicating human range of motion, human kinematics and human dynamics is very important. Any action/ gesture performed with a robot faces the challenge of the motion looking mechanical. This is because, human constraints and kinematics of a robot are different. Correct translations of human motions to a robot play a huge role in motion planning for the type of application where human motion is to be reproduced by a robot.

Another problem in HRI surfaces when a robot copies the human motion. As discussed before, the human and robot's constraints in their kinematics are in effect here as well. The actuator movements must be mapped from the workspace of the human. Workspace of a human is defined by Gupta in [20] as "A set of reachable pose states for a typical human within the scene". The robot also suffers from singularities which must be taken into account. Singularities are "Configurations in which there is a change in the expected number of instantaneous degrees of freedom" [21]. Copying of an action also involves visualization of the motion followed by its execution. Timing constraints are to be considered here to keep the motion real-time.

The major problem in HRI is the lack of many systems which define multiple motion trajectories using a pre-defined framework. HRI involves repeated tweaking and changing of the position values of the servos of a robot to suit the change required in the interaction. Incorporation of a single system having the ability to change the motion path internally without manual intervention is needed. The system to be defined must have the ability to perform all tasks and must also have a comparison factor with which change can be made.

1.3. Details of the research carried out

In the thesis, a system for encoding a motion used in imitation learning was formulated and implemented. The defined system records a motion and replays the motion on a robot by the use of an architecture called Dynamic Movement Primitive (DMP). Subjects were asked to imitate the robot and their motion was compared with the robot`s motion using Dynamic Time Warping (DTW). The thesis also validates the use of DTW, as a comparison metric, through experimental results. A system for imitation learning with validation is presented.

The work started by reviewing the literature regarding learning by demonstration methods. Studies depicting how humans learn, components of a human motion and how humans adapt to these motions were studied. Recent work on how robots have been used as assistive teachers was also reviewed. How human joints are actuated and how humans move was studied. Literature on how to generate a human like motion was surveyed and thoroughly understood. Work related to motion comparison were researched.

Once the survey was done, the development of the system which assists in imitation learning was developed. The DMP algorithm was selected and implemented on a robot model. The implemented DMP motion was compared with human motion. Experiments were conducted where a human mimics a robot. The data was recorded and DTW was used as a comparison measure. The validation of DTW`s use as a comparison metric was done statistically using hypothesis testing.

The robot interactions were performed using a robot called Zeno. Zeno is an expressive humanoid robot developed by Hanson Robotics [22]. It was used in research conducted at the UNT Health Science Center with Autistic children and at University of Texas at Arlington`s Next-Gen Systems Lab with healthy subjects. The research included collecting data using the robot when a scripted motion was performed by the robot and people were asked to follow Zeno. The complete software for Zeno is written in LabVIEW. For demonstrating DMP and implementing the system, this robot was used. Using the Kinect motion sensor, an ideal motion performed by a trainer was recorded. This recorded motion was replayed on the Zeno Robot by incorporating the motion into the DMP architecture. The DMP architecture is a set of non- linear differential equations which can generate motion by just changing the time, and the trajectory end points. A subject was asked to imitate this learned motion performed by the robot. The robot would later be made to adapt its motion to match the subjects' motion by changing certain parameters in DMP.

DTW was used to compare the motions. To validate the use of DTW in these experiments, another set of experiments were conducted, where Zeno was made to perform certain scripted motions such as a wave motion, a tummy rubbing motion and a fist bumping motion. These gestures were scripted in LabVIEW. Participants were made to follow the robot while holding different weights in their hands. How closely the participants followed the robot was analyzed using DTW algorithm, which was also coded in the LabVIEW software. The objective was to conclude if DTW could determine the change in motions when participants were subjected to different weights. The purpose of the experiment was to find out the reliability of use of DTW in finding out motion impairment. These results would then be extended to validate DTW as an objective measure of impairment in limb motions for early detection and diagnosis of Autism.

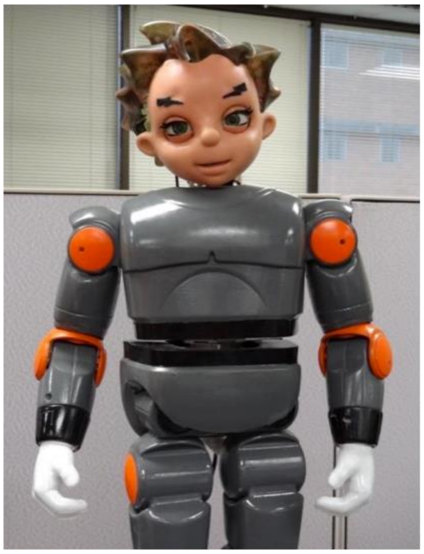


Figure 1.1 Zeno Robot [23]

1.4. Research contributions of this thesis

The contributions of this thesis consists of gesture mapping and comparison for HRI. The gestures are developed, analyzed and encoded using the algorithms Dynamic Movement Primitives and Dynamic Time Warping. These gestures are implemented on a humanoid, the Zeno Robot.

- This thesis proposes DMP for upper limb motion generation of the Zeno robot. The ability of the user to change the motion profiles of the robot after its training is explored. The proposal is validated through simulations and experiments. DMPs, used for the motion generalization, are a set of non- linear equations. General set of motions like hand waves are applied using this architecture.
- Validation of DTW as a metric for imitation quality is presented. This is established statistically through a set of experiments involving healthy subjects imitating a robot. The results are based on experiments with 56 different human subjects.

DMP and DTW are then used in an imitation learning system where gestures are encoded on the Zeno robot using DMP and the human responses are compared using DTW. The DMP motion is then adapted according to the DTW error value and the system is re run.

The research contributions are summarized in the following papers

- Isura Ranatunga, Namrata Balakrishnan and Dan Popa, “User Adaptable Tasks for Robot Differential Teaching”, submitted to 6-th Assistive Robotics Workshop, PETRA.
- Indika Wijayasinghe, Isura Ranatunga, Namrata Balakrishnan, Dan Popa, “Human- Robot Gesture Analysis” for Diagnosis of Autism”, to be submitted to International Journal of Social Robot.

The research about Dynamic Movement Primitives has been presented at the ACES 2014 (The Annual Celebration of Excellence of Students) Symposium at University of Texas at Arlington through a poster – Namrata Balakrishnan, Isura Ranatunga and Dan Popa, ‘Adaptive Robotic Teacher for Gesture Imitation Learning’.

1.5. Thesis Organization

Chapter 2 is the background survey of the topics: 1) Research conducted in Human Robot Imitation, 2) Research in Dynamic Time Warping and 3) Research in Dynamic Movement Primitives.

Chapter 3 describes the motion encoding algorithm Dynamic Movement Primitives in detail. It also discusses the gesture adaptation technique. It describes the system overview for experiment of imitation analysis using DMP on the Zeno robot

Chapter 4 discusses DTW algorithm and the experimental set up for DTW validation by gesture analysis on the Zeno robot.

Chapter 5 presents the results obtained from both the experiments.

Chapter 6 concludes by summarizing the thesis and also discusses the future work that can be done to extend this thesis.

Chapter 2

Background Survey

Human-Robot Interaction has been proposed to solve many challenges for service robots. For robots to interact with humans, simple as well as complex tasks must be taught to them. In the literature, humans act as teachers and robots learn the demonstrated gesture [24]. Robots acting as teachers and humans as learners have also been demonstrated [24]. Teaching a robot to perform human motion has been of interest for a very long time. The sections in this chapter discusses briefly the post research advances made in the field of HRI.

2.1. Research conducted in Human Robot Imitation Domain

Under HRI, especially human mimicry, it becomes very difficult to map human motion directly onto the robot because of many issues like difference in degree of freedom, singularities of the robot as discussed before. These issues have been studied by many researchers. The joint angles and velocities of a robot are far more limited than that of a human. Pollard [25] suggests that the human motion data that is captured by any motion sensor be scaled to the level of the robot. The scaling was done by limiting the joint angles and the joint velocities. Dillmann [26] suggests using the traditional method of inverse kinematics where joint angles are determined from position and orientation of the link. He uses a “sensorimotor transformation model” to map the angles of joints of human into quaternions [26]. These quaternions were then mapped onto the robots joint angles from the inverse kinematics solution. Kim [27] proposes a motion capture database to generate human like gestures.

Other methods for mapping human motion on to a robot include similarity mapping, affine mapping and variable similarity mapping. Kuchenbecker [28] suggests these three new mapping methods along with the traditional mapping technique.

Traditional mapping is about scaling the human motion and to add an offset to get the robots motion. The three new mapping methods proposed in [28] are “similarity transformation, affine transformation and variable similarity transformation”. Kuchenbecker states that “similarity transformation consists of a scaling, a rotation, a reflection, and/or a translation” [28]. Kuchenbecker also stated that “affine transformation consists of a strain, a shear, a rotation, a reflection, and a translation” [28]. Variable similarity mapping consists of warping. Scaling and then adding offset to the human motion to convert it into robotic motion.

While joint angle mapping has been proposed by many, Ishiguro [29] proposes a method to map the appearance of the human on to the robot with less error. This research suggests determining the posture of the human and the android and then comparing them [29]. Markers are used to obtain points from human body and similar number of markers are placed on the robots body. A neural network is used to train the human’s posture to the robots desired joint angles. Here differences are noted at the visible surfaces. With the degrees of freedom differing between a robot and a human, the direct mapping of joint angles would make the motion appear not human-like. This method works with the physical limitation of the robot and improves human likeness.

Dillmann [30] also works with mapping human motion onto a robot using motion marker system where a Motion Map is utilized to represent a human motion on a humanoid robot. This model incorporates the body segmenting property from biomechanics, thus emphasizing that when the segments are transferred onto the robot the motion would be more human-like. Changes in the position is introduced rather than change in joint angle mapping so as to maintain the human-like characteristic of the segmented data.

Atkenson [31] programs robotic behavior to be more human-like by understanding how humans learn a behavior. Trajectory planning, learning from demonstration and oculomotor control are discussed in [31]. For joint angle mapping, Atkenson suggests using “redundant inverse kinematics algorithm known as extended Jacobian method” [31]. Here joint angles in the joint space are searched with a minimum energy and time constrain. Joint mapping is also learnt from demonstration where the joint angles are scaled and fit to the robot. Alternative method of trajectory planning is stated in [31] where movement trajectories are stored in memory and retrieved whenever necessary. The trajectories are defined as movement primitives which only require speed and amplitude to generate discrete rhythmic movements [31].

Another method of defining trajectories are, based on “Gaussian Mixture Models (GMMs)” as discussed in [32], a combination of “Gaussian/ Bernoulli distributions (GMM/BMM)” as discussed in [33] and also estimated by “Gaussian approximation of quasi-linear key phases” [34] . The features to be imitated are extracted from the dataset and linearly transformed by Principle Component Analysis (PCA). The extracted data set is then analyze using Dynamic Time warping (DTW) and then it is encoded using GMM/ BMM. The Gaussians in the GMM are used in segmenting the trajectories [34]. The basic skills acquired are reorganized and a continuous trajectory is reproduced. The trajectory is computed taking the robots kinematic constrains and the goal positions into consideration [33].

Ijspreet [35] introduces a concept of Control Policies (CPs) to encode the non-linear dynamical system of trajectories. Control Policies are autonomous non-linear set of differential equations which encodes an entire landscape rather than a single trajectory pertaining to a set of discrete motions. CPs are stable and robust against external perturbations [35]. This has led to the development of the - Dynamic Movement Primitives (DMP) framework.

An application of using Control Policies for encoding a trajectory is studied in [36]. The Nearest Neighbor technique is used for training. In this work a differential drive robot is simulated and trained by learning control policies taught by a user to intercept balls [36]. As the robot performs the task, the human critiques it by using the same control policy. Thus critiquing is simpler than hand coding. Billard includes the human as a teacher as well as a participant in [32] and [37]. This is understood as incremental learning.

2.2. Research conducted in Dynamic Movement Primitives

Dynamic Movement Primitives has been a popular architecture for defining trajectories for a very long time. The versatility and the easy generalization has attracted researchers to use DMPs in various applications. DMPs can define any type of complex trajectory as a set of differential equations which are non- linear in nature. The ability to encode discrete as well as oscillatory motions have made them more appealing than Control Policies.

One such example is using DMP for handwriting generation. The problem involves obtaining complex handwritings and converting them into the versatile primitives. Kulvicius [38] uses modified primitives to simulate handwriting and join these primitives to form a continuum in space. The modified form defined can be easily used in applications where a task is to be disintegrated into smaller components and later joined into one. A method of overlapping the kernels is defined to overcome the problem of the velocities reaching close to zero when primitives are joined by the conventional method.

Another example of the use of DMP in HRI is shown in [39]. In [39], an object is handed over between a human and a robot. With the goal moving, this modification of

DMP works by handling the destination position of robots arm without knowing the specific location. A velocity feedback term is used so that the direction of the moving target is known. Thus, an introduction of change in the human hand position produces on-line change in the robots motion [39].

The generalization ability of DMPs is explored in [40] and [41]. In [40], a robot reproduces pick and place and water pouring tasks using the non-linear set of equations i.e. DMP. Pastor [40] introduces DMP parameters and describes its framework. This framework is used as the building blocks to learn the action performed by a human. DMPs generalize a motion by just changing the time, beginning and end points of the trajectory, thus allowing easy learning. As these tasks can have different goal positions, adaptation is easily done. DMPs are also robust against perturbations and can be used in obstacle avoidance [41].

Calinon [34] includes the Gaussian Mixture problem with the Dynamic Movement Primitive Model. The weight activation mechanism in the DMP model is primarily done by locally weighted projection regression (LWPR) [34]. Instead, GMR learns the weights. It is proposed to be easier than LWPR [34].

2.3. Research conducted in Dynamic Time Warping

Dynamic Time Warping is an algorithm which has been in use for comparing temporal sequences for a very long time. From this algorithm the least distance attained by aligning the two sequences is obtained. Dynamic Time Warping although initially introduced for speech recognition [42] [43], has now been used in various fields like medicine [44], data mining [45] [46], entertainment [47] to name a few.

Dynamic Time Warping was bracketed out of Dynamic Programming. Dynamic Programming [48] algorithms are used for optimization of sequences by solving the problem by breaking them down into sub problems [49]. Dynamic Programming applications pertained to two large classes of problems- sequence alignment and hidden Markov model problems [49]. Our focus is drawn towards two sequence alignment which defines the comparison problem involving a sequence and then perturbed sub sections of the second sequence resulting in a cost variable to define the mismatch. Minimum cost is sought [49].

The use of Dynamic Time Warping was first documented in [50] for speech discrimination. The algorithm of finding a class nearest to the sequence to be determined is explained using Dynamic Programming. Sakoe [42] describes Dynamic Programming Algorithm with time based normalization for speech recognition. The time normalization effect in this paper is non-linear in nature. The minimum distance between the two sequences is calculated and time difference between the two sequences is eliminated by time warping.

Christiansen [43] introduced time-warping algorithm for detecting words in speech. In this paper, similarity is measured between speech samples based on template matching, linear prediction and similarity measurement. Most of the applications of Dynamic Time Warping are for the speech recognition domain [44] [51].

In the domain of data mining, Dynamic Time Warping has been used for data retrieval from comparison. Lijffijt [47] talks about Dynamic Time Warping as an optimal aligning tool for two sequences. In this research, Lijffijt studies “the matching of time series and their performance using of Dynamic Time Warping” and “constrained Dynamic Time Warping” in music notes for music queries from a database [47].

[45] and [46] use the popular DTW algorithm for the comparing hand gestures from the American Sign Language. The hand gesture query is compared to a database of hand gestures using this DTW algorithm. Here they combine the time series of hand gestures with hand appearance and use this algorithm for finding out similarities.

Normalization of sequences before running DTW is studied. It is stated that the length of the two time sequences that are to be compared must be the same as the DTW algorithm is biased against longer database matches.

Chapter 3

Imitation Analysis on Zeno Robot using Dynamic Movement Primitives

3.1. Concepts of Dynamic Movement Primitives.

Dynamic Movement Primitives are used for controlling joint motions. The advantage of generalizing a motion into a set of non- linear dynamic equations has been fascinating and has drawn a lot of attention from many researchers. The desirable properties of DMP - easy to generalize a motion by just changing a few parameters - enable easy learning. They can easily model both discrete time and rhythmic movements [19].

Dynamic Movement Primitives are nonlinear differential equations used as control policies for encoding a robotic motion. They were first introduced by Ijspreet [35]. This chapter explains the concept of DMP and describes it original form. Modified form of DMP is also discussed in this chapter. The system of Imitation analysis, where DMP is used, is explained in detail in this chapter.

3.1.1 Dynamic Movement Primitive

The Dynamic Movement Primitive system consists of three parts: the canonical system, the modulation function, and a stable converging dynamic system. The dynamic system is a second order differential equation. It is an attractor landscape pulling the state variable from initial to the final position through time. The definition used for the dynamic system is obtained from [40]:

$$\tau\ddot{x}(t) = k\left(x(t_f) - x(t)\right) - D\dot{x}(t) + \left(x(t_f) - x(0)\right)f(s) \qquad (1)$$

The dynamic system is a spring-damper system called the transformation system. The system starts from time, t = 0 and continues untill time,$t = t_f$. $\ddot{x}(t)$, $\dot{x}(t)$ and $x(t)$ depict the acceleration, velocity, and position respectively. $x(t_f)$ defines the final position. k is the spring constant, D is the damping coefficient and τ is a time scaling factor. s is a phase variable which affects the driving force of this spring-damper system. $f(s)$ is the force that drives the attractor landscape towards the goal through a specific motion. It is by modification of the function $f(s)$ that any type of motion can be obtained using just the second order differential equation.

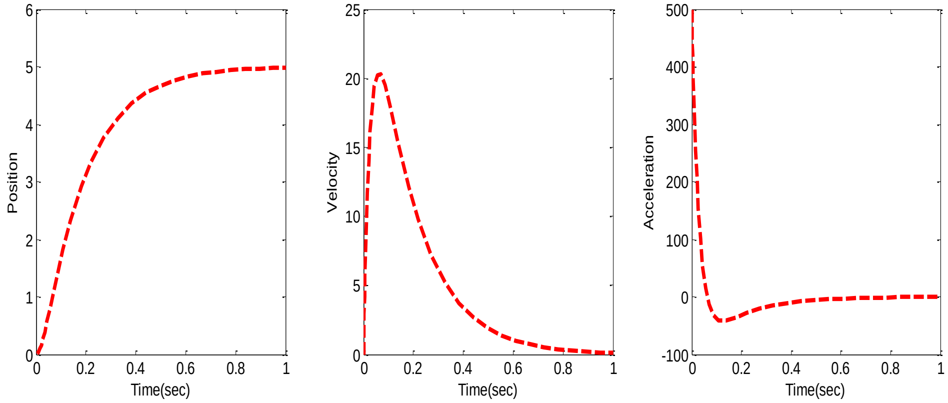


Figure 3.1 Depiction of the transformation equation without any external force

With the values of f being 0, k and D taken according to [40], the representation of equation (1) is shown in Figure 3.1. The change of the motion in position, velocity and acceleration is seen for $\tau = 1$. The goal is set at 5 with initial condition given 0. This forms a globally stable linear attractor system [52]. To obtain more complex trajectories pertaining to motor movements of a robot, the external force f is changed into a non-linear function. This function is called the modulation function which is a function of phase variable. The modulation function is defined as

$$f(s) = \frac{\sum_{1=1}^{N} \psi_i(s) w_i s}{\sum_{i=1}^{N} \psi_i(s)} \tag{2}$$

Here, $\psi_i(s)$ are the Gaussian basis functions of $f(s)$ which are defined as $exp(-h_i(s - c_i)^2)$. The Guassian basis function has a height of h_i and center c_i. N is the total number of Gaussian basis functions used for the modulation function. The modulation function has adjustable weights w_i and also contains the phase variable s. The canonical function which generates the input s for equation (2) is

$$\tau s = -\alpha t + 1 \tag{3}$$

Here τ is the scaling factor and α is a constant. The canonical function is a basic linear equation with a negative slope. It generates the phase variable s, which drives the modulation function. The phase variable is a function of time, t. On solving the equation (3) for a particular value of α the result shown in Figure 3.2 is obtained.

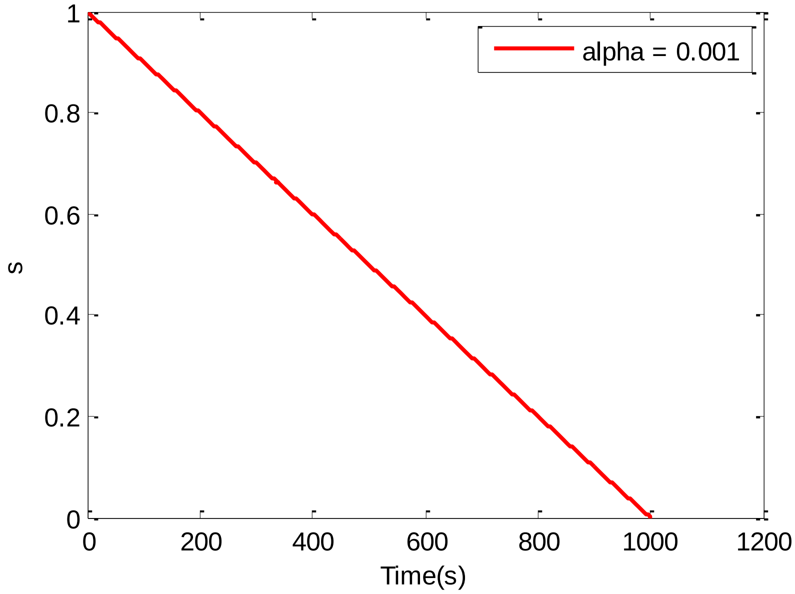


Figure 3.2 Variation of the phase variable through time

The weights of the modulation function are assumed to be bounded. As the canonical form also is stable and bounded, the complete system is assured to be stable and thus can converge to the desired goal, $x(t_f)$. The convergence of phase variable ensures the attractor to reach the goal. Thus different values of α can be used depending upon the speed and the degree of convergence required for the experiment.

Figure 3.3 depicts the plot of the Gaussian basis function, $\psi_i(s)$ which is defined as $exp(-h_i(s - c_i)^2)$ with h at 1000 and center c at 0.01 distance. On using random weights initially, the solution for the equation (2) over the phase variable is depicted in Figure 3.4. It clearly depicts how the Gaussian kernels affect the modulation function with the help of weights.

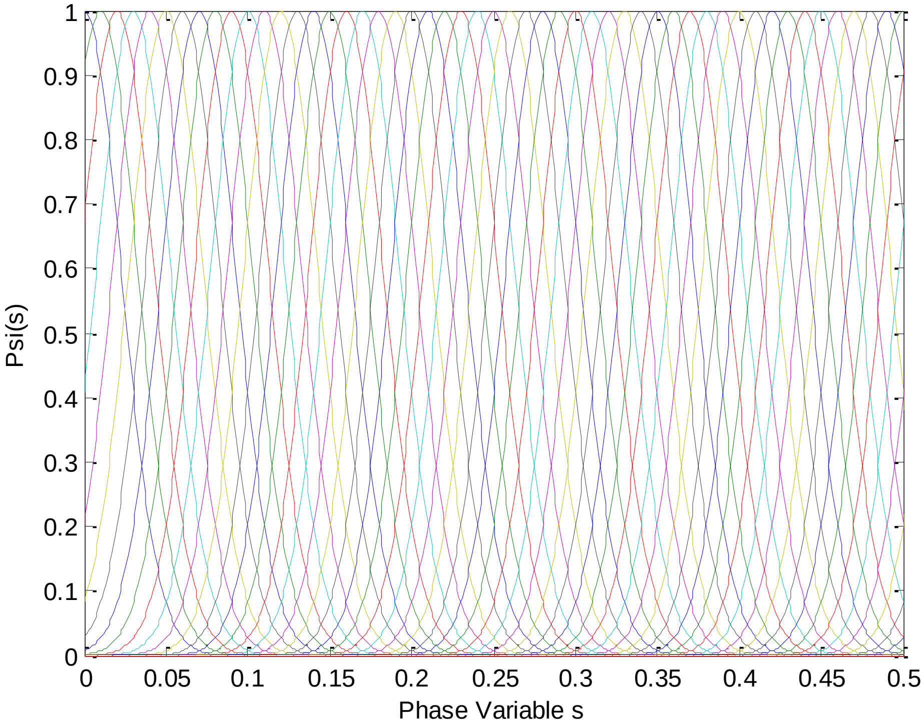


Figure 3.3 Gaussian kernels, 1000 in number with centers spaced at 0.01 plotted over the phase variable

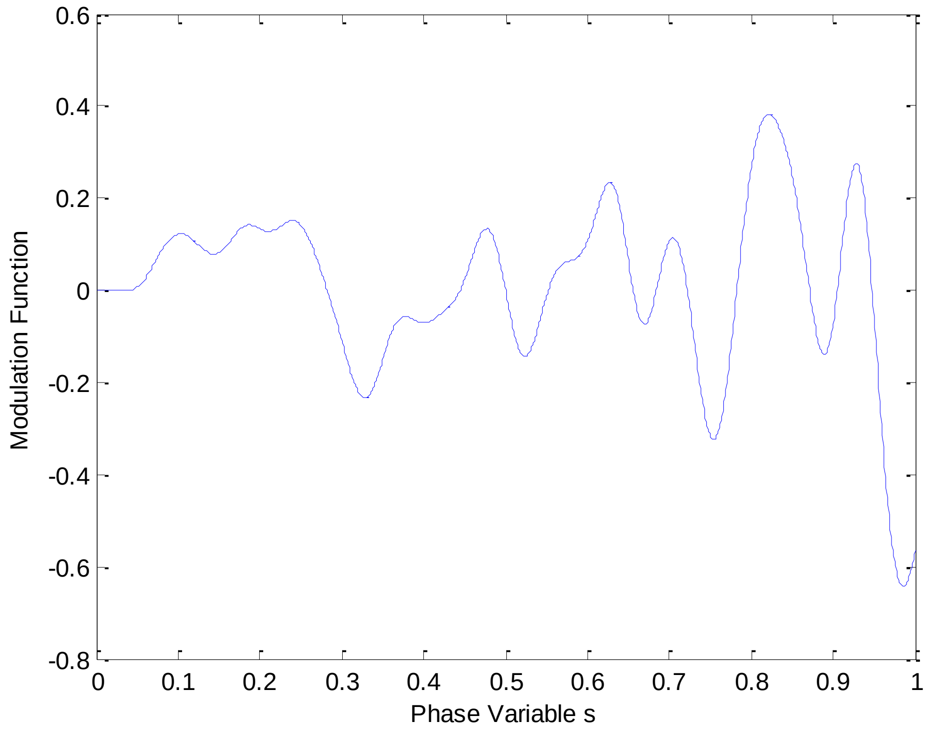


Figure 3.4 Variation of modulation function through phase variable

3.1.2 Training of Dynamic Movement Primitives

The DMPs have the desirable characteristics of learning a motion and replaying it with just the beginning and end point. For the system to do so, the DMP must be trained initially, so that it can modify itself and replay the motion whenever required. Training of DMP involves the modulation function where weights appropriate to the experiment are found from the training process. The model used for training is depicted in Figure 3.5. For training of the modulation function, a target variable is found from the transformation system’s equation as given below.

$$f = \frac{-k\left(x(t_f) - x\right) + D\dot{x} + \tau\ddot{x}}{x(t_f) - x(0)} \tag{4}$$

The training motion is given in x, which is an array of positions through time. $\dot{x}$ and $\ddot{x}$ are also found and subsequently zero padded, so that x, $\dot{x}$ and $\ddot{x}$ are all of the same size. $x(0)$ depicts the first value in x array and $x(t_f)$ depicts the final value in x array. The canonical equation is used as an input for this training process. For training the weights of the modulation function, s and f are generated. The training process is described in a flow diagram in Figure 3.5.

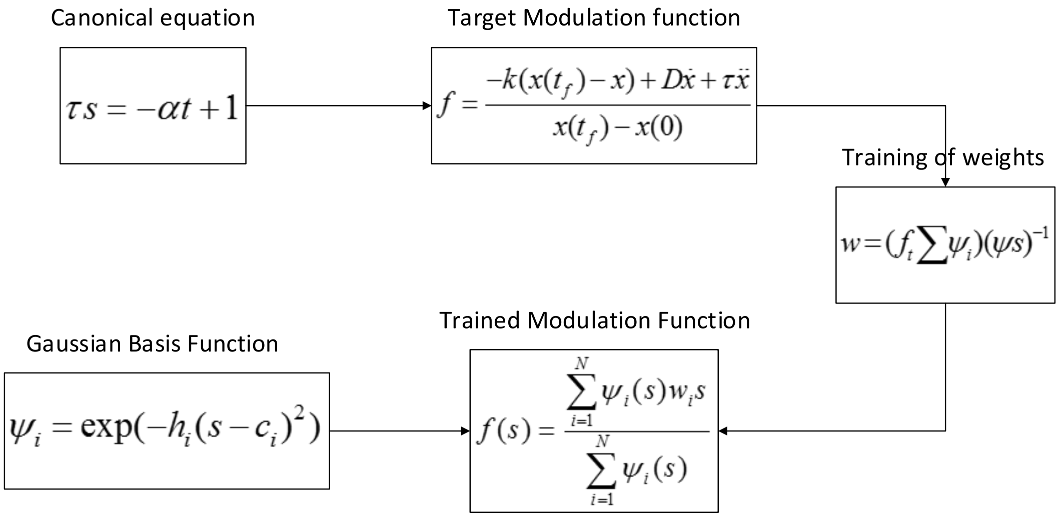

Canonical equation
$\tau s = -\alpha t + 1$
Target Modulation function
$f = \frac{-k(x(t_f) - x) + D\dot{x} + \tau\ddot{x}}{x(t_f) - x(0)}$
Training of weights
$w = (f_t \sum \psi_i)(\psi s)^{-1}$
Trained Modulation Function
$f(s) = \frac{\sum_{i=1}^{N} \psi_i(s) w_i s}{\sum_{i=1}^{N} \psi_i(s)}$
Gaussian Basis Function
$\psi_i = \exp(-h_i(s - c_i)^2)$

Figure 3.5 Diagram depicting the DMP training process

Here weights in the modulation function are determined using closed form solution. Substituting the target modulation function from equation (4) into the equation (2), and solving for w, would give the weights for the modulation function. The training of weights can also be done using locally weighted projection regression model [52]. A fixed number of kernels in the Gaussian Basis function are used to approximate the data. The Figure 3.6 also depicts the variation in the modulation function after training of weights. It can be seen in Figure 3.6 that weights generated from training affect the height of the Gaussian kernels. The cumulative effect of all the Gaussian kernels act on the modulation function thus producing a graph as visible in Figure 3.6. This new generated modulation function is compared with the one used for training. This comparison is shown in Figure 3.7. The success of the training process is depicted by this comparison graph.

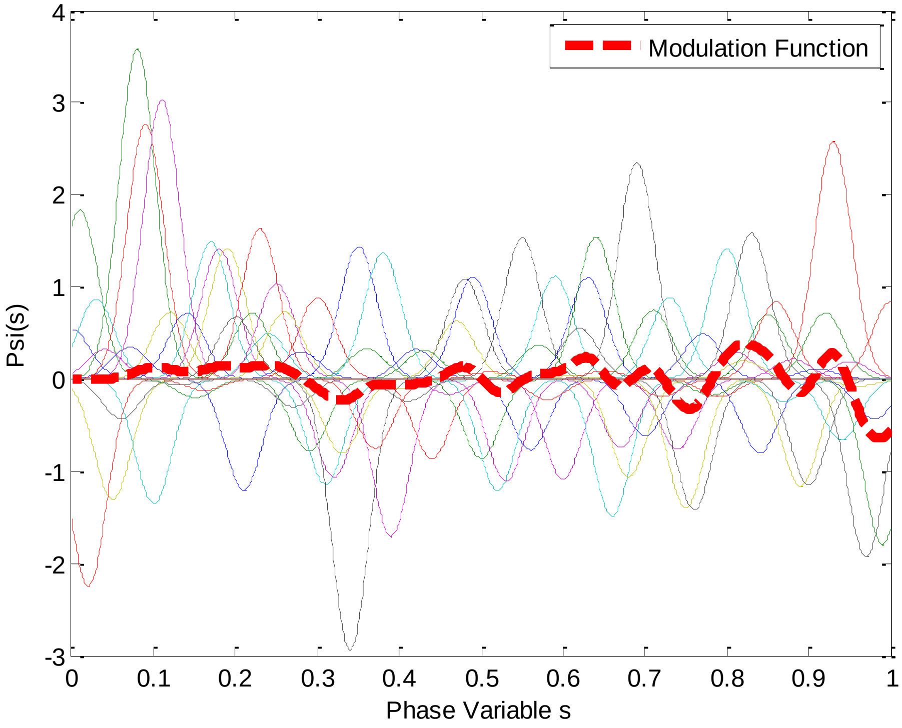


Figure 3.6 Weighted Gaussian curves over phase variable

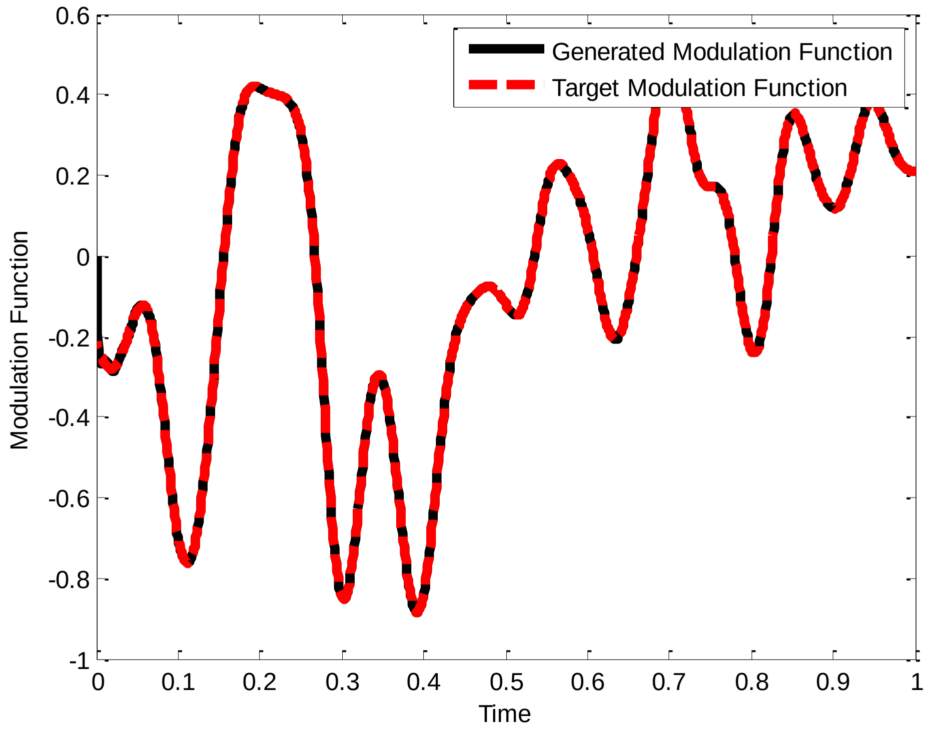


Figure 3.7 Target modulation function v/s generated modulation function

3.1.3 Running of Dynamic Movement Primitives

Once the modulation function is trained, then just by giving a start and a final point, the architecture outputs a trajectory which is driven by the modulation function. The transformation system and the canonical systems are integrated. The newly trained modification function, along with the weights, modifies the transformation system. Figure 3.8 depicts the original motion and the motion learned by the trained DMP parameters.

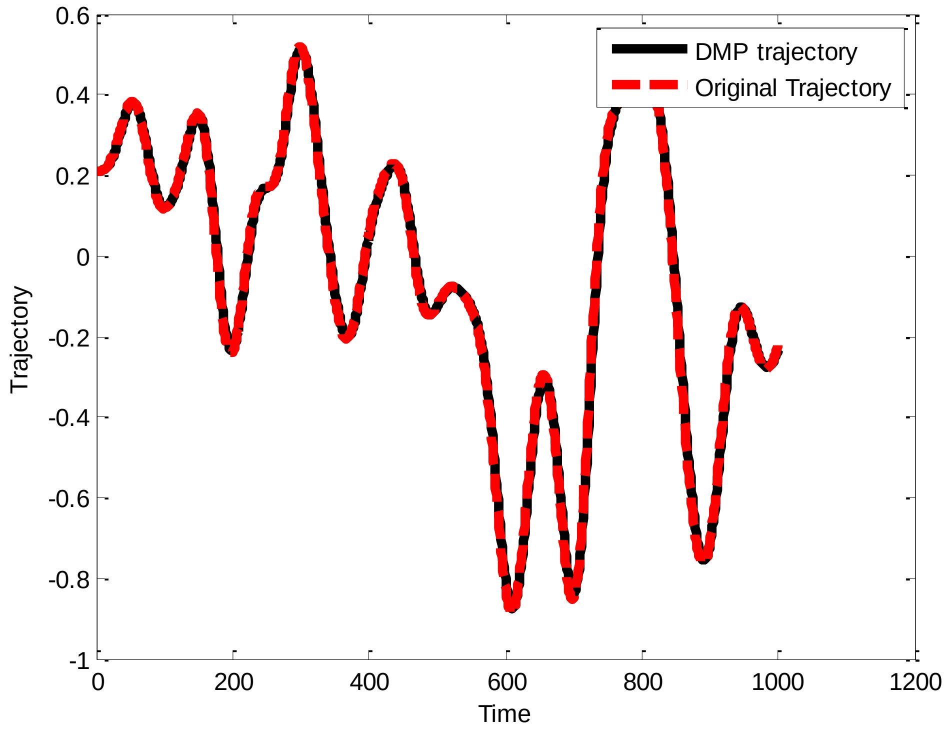


Figure 3.8 Motion that is to be trained (Original trajectory) v/s the learnt motion (DMP trajectory)

3.1.4 Modifications in Dynamic Movement Primitive

While experimenting around DMP by changing the parameters, it was seen that the original DMP formulation did not work for certain conditions. If the starting and goal position of a trajectory is same, then the DMP would produce an output as depicted in Figure 3.9. This is due to the fact that the modulation function was weighted by the difference in the beginning and end position of the trajectory. If the difference becomes zero, the effect of the modulation function would be nullified. A modified DMP as defined in [40] was observed to overcome the problem.

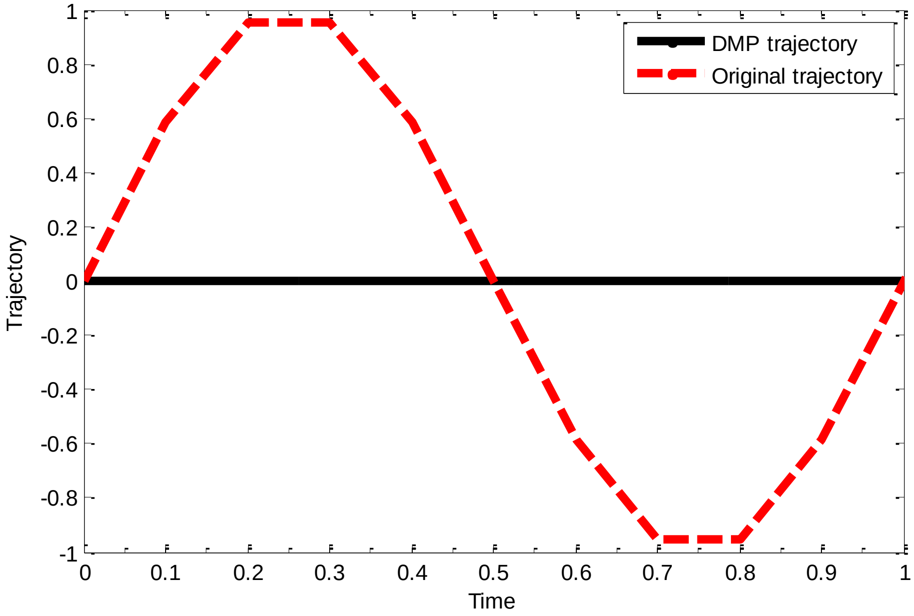


Figure 3.9 Original formulation of DMP v/s Trajectory to be learnt

This modified approach involves a little change in the transformation system dynamics equation while retaining the canonical systems equation as is. The equation (5) is defined in [40].

$$\tau \ddot{x}(t) = k\big(x(t_f) - x(t)\big) - D\dot{x}(t) - k\big(x(t_f) - x(0)\big)s + kf(s) \qquad (5)$$

This form differs from the original form as the modulation function is now not weighted by the difference in the present position and the goal. Thus for the points where the starting and ending point are same, there will not be any problems with DMP generation. This case was implemented and the observed results are depicted in Figure 3.10.

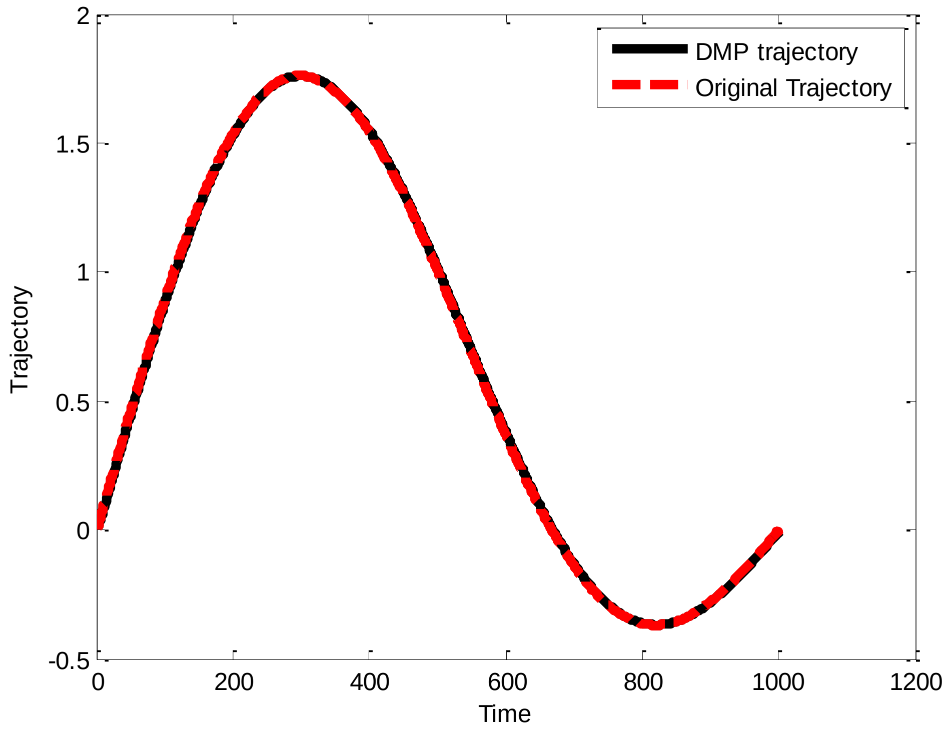


Figure 3.10 Modified DMP v/s Original trajectory

For this formulation, the equation for the training of the modulation function is depicted as follows.

$$f = -\big(x(t_f) - x(0)\big)s + \big(x(t_f) - x\big) + \frac{\tau\ddot{x} - D\dot{x}}{k} \tag{6}$$

Due to the visible advantage, the system generated for the experiment uses the modified formulation of DMP.

3.1.5 Properties of Dynamic Movement Primitives

The presence of several favorable features make DMP the most used form to encode a motion. In this section, the properties advantageous to the experiment are noted.

Multiple degrees of freedom: DMP can be observed in single degree of freedom as well as multiple degrees of freedom. This can be visualized by using a single canonical form for all the degrees and different transformation system defining the state in different dimensions [53].

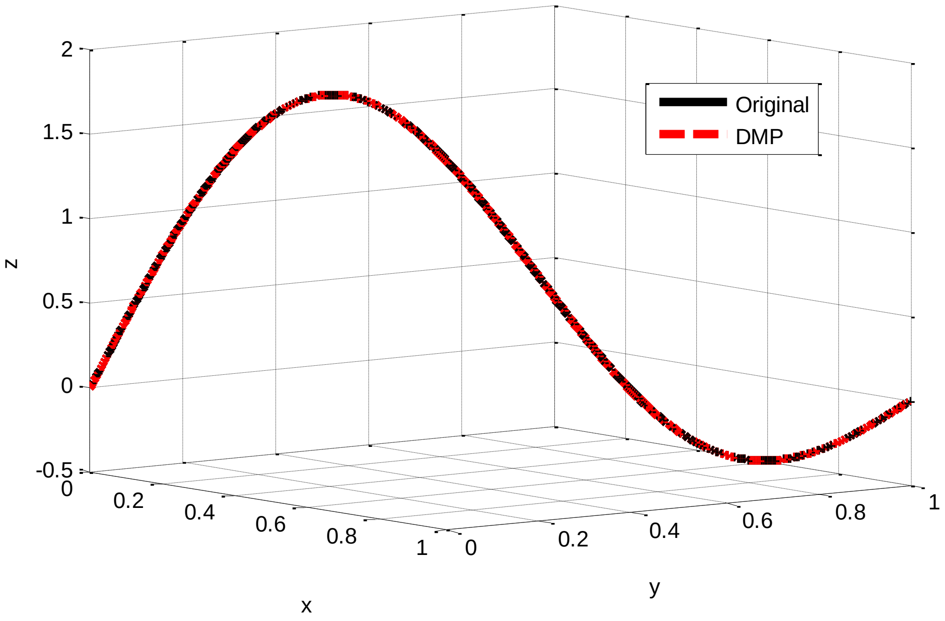


Figure 3.11 Multi degree freedom DMP with comparison of original trajectory and learnt DMP trajectory

Thus all the degrees of freedom would have the same phase. Hence there would be a valid relationship between the multiple degrees of freedom. Figure 3.11 depicts a 3D sine wave learnt and reproduced by the 3D DMP model. The 3D DMP model has the similar structure as the original DMP with a different transformation system equation per degree of freedom. Thus equation (1) will be ran three times and learnt with three different modulation functions and different weights.

DMP`s are used as building blocks of a motion.  Thus to generate a complex motion, many simple DMP`s can be integrated together. As DMP is a point to point motion generation system, a rhythmic motion can be obtained by superposition principle.

Identification of movements is possible using DMP. Individual movements have specific weights. By simply changing the start and final points or duration of a movement, alterations can be done using DMP, but the weights do not change. This factor can be used in identification of a specific gesture by simple classification using the weights.

DMP have the usefulness of being very robust. If there are any obstacles in the way, the DMPs have the inherent ability to overcome the hurdles. While the DMP is being executed, if unknown hurdles are inserted, DMP has the ability to adapt online by just changing its goal position during execution.

DMP also have the property of adapting to changing goal positions. Once a DMP is learnt, if the goal positions are changed at the beginning of trajectory or during the motion of trajectory, DMP has the ability to adjust its parameters and adapt to the goal efficiently. This is shown in the Figure 3.12 and Figure 3.13. Figure 3.12 shows the effect of changing the goal position after training of DMP has been completed. The trained DMP can be seen to perfectly reach the set goal points of [1, 1, 0.5] and [1, 1,-0.5]. Figure 3.13 shows how the DMP reaches the set goal points even after command to track the original trajectory has been given. In this testing, after 1/5 time of DMP execution had been elapsed, the goal points were changed from [1,1,0] to [1,1,0.5] and [1,1,-0.5] in two different runs. The DMP reaches the goal points during online adaptation successfully.

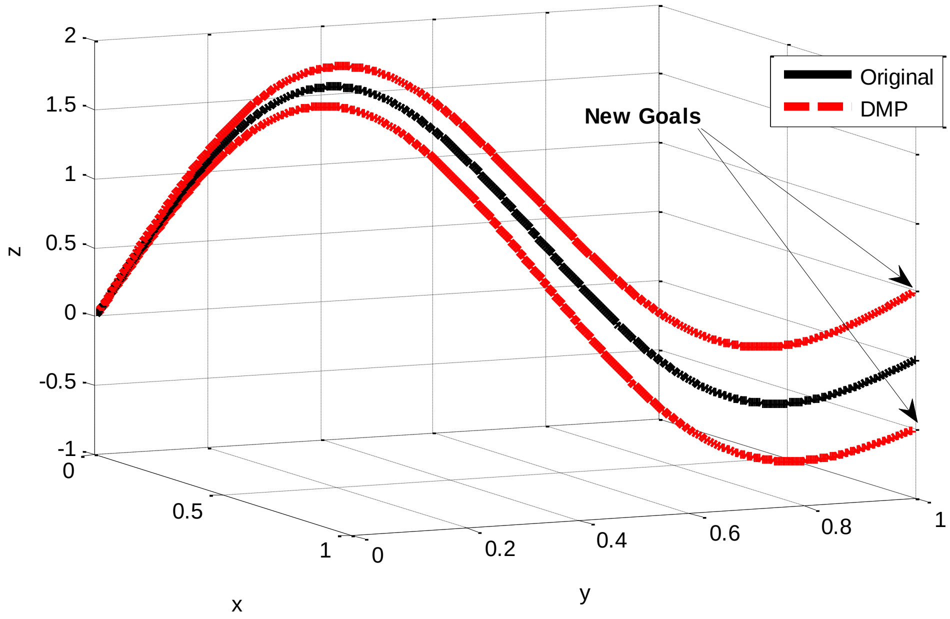


Figure 3.12 Effect of change in goal position on DMP

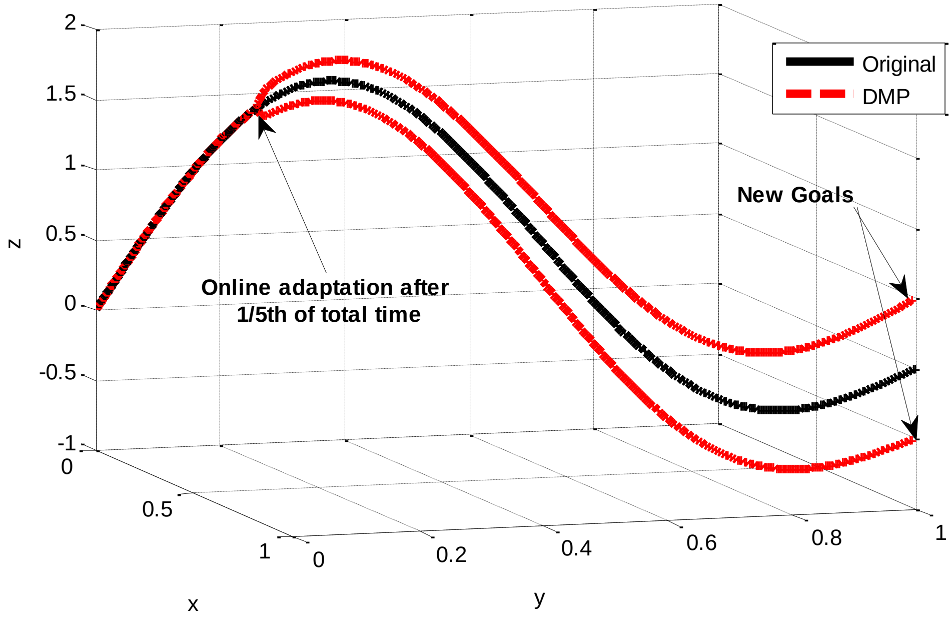


Figure 3.13 Change in goal position after DMP has been executed

3.2. Imitation analysis system using DMP on Zeno

A system for imitation learning is proposed in this thesis. While considering the advantages of DMP, a system for gradual improvement in learning is developed. The set of non- linear differential equations are used to generalize a motion. This motion is to be mimicked by an individual. Each motion is recorded after it has been demonstrated. The demonstrated gesture is trained on the Zeno Robot. Zeno learns the motion and replays it. The individual, who is to learn the demonstrated gesture, tries to imitate the robot. The gestures performed by both the trainer and the subject are captured using Kinect – motion sensor. The point cloud data collected from the trainer is used in the formation of differential equation for the robotic motion. The subjects' actions, also captured by the Kinect, are compared with that of Zeno and further analysis is done.

3.2.1 System for visual capture of motion

The purpose of the experiment is to make a robot imitate a human. Imitation involves sensing of motion and replaying it on the robot. For the purpose of sensing, the most commonly used sensor device is a Microsoft Kinect Sensor. Kinect is a Microsoft sensor used in Xbox 360 for gaming purposes. The ability of Xbox 360 Kinect sensor to track a human body makes it a very useful sensor for gesture imitation project [54]. The Kinect sensor is used to track the human skeletal model.

Along with sensing, the Kinect sensor is also used as a feedback device. Once the robot performs the desired motion, the subject is asked to imitate the robot. The subjects' skeletal framework is captured by the Kinect sensor and used as a feedback. The difference between the motion performed by the subject and the robots' motion are published.

Figure 3.14 Kinect

3.2.2 Hardware Background

The experimental set up uses a miniature humanoid robot called Zeno. Zeno is a 2 feet tall humanoid robot developed by Hanson Robotics and Hanson Robokind. The robot has an expressive face with movable eye lids and lips. It has 4 degrees of manipulation in each limb and 1 degree of freedom for torso movement and rigid legs attached to a base [55]. The degrees of freedom is attained by Dynamixel RX-28 servo motors. These motors are present at each joint of the robot. It is controlled externally using a Dell Quad core laptop using the software LabVIEW.

Torres states that for this robot, each arm has “four degrees of freedom” which corresponds to alpha, beta, gamma and theta angles [55]. This is shown in Figure 3.15. The alpha angle corresponds to the flexion and extension in the sagittal plane. The beta angle corresponds to the abduction of the upper limb in the frontal plane. The gamma angle denotes the supination and pronation of the arm. The beta angle corresponds to the flexion and extension of the elbow joint in the sagittal plane.

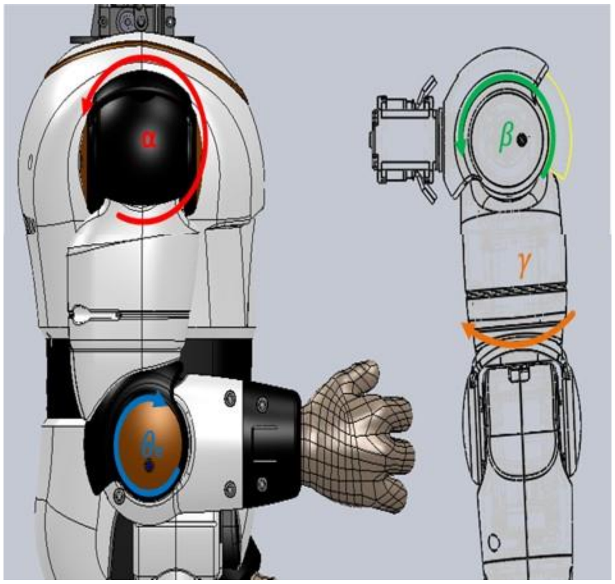


Figure 3.15 Zeno joint angles [55]

These joint angles are manipulated by dynamixel servo motors using LabVIEW software. Dynamixel Servos are actuators for robots having feedback functionality and programmability. Figure 3.16 shows the Dynamixel motor RX-28 present in Zeno’s arm.

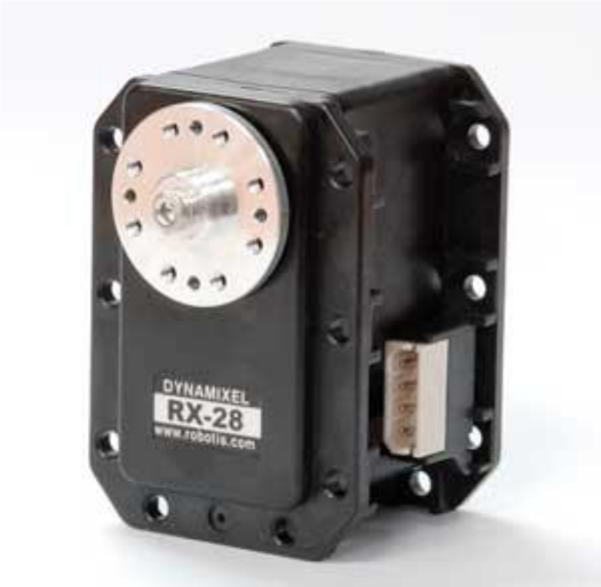


Figure 3.16 Dynamixel RX – 28 [56]

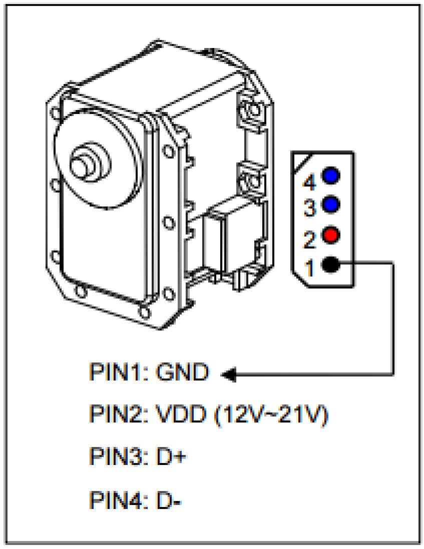


Figure 3.17 Pin connections [56]

To operate this servo motor, a power supply of 14.4 V is required [56]. The pin assignment of a connector is shown in Figure 3.17. A power connector circuit is designed with a power switch and a LED indicator. For the purpose of communication, the main controller supports a RS485 UART communication method. Thus, the motor communications/ commands from the Laptop must be directed to the Dynamixel through a USB to RS485 convertor. The program for controlling Zeno`s motion is coded in LabVIEW and then using USB2Dynamixel communicated to the Dynamixel motors.

Once Zeno performs the DMP generated gesture, the subjects are asked to imitate the motion. The subjects’ movements are captured using Microsoft Kinect. Kinect is an RGB camera and also a depth sensor. It is used to capture 3D motion. The 3D data points of the motion through time is recorded and then using inverse kinematics, the variations in the value of human arm joint angles were found. The hardware setup of the complete system is shown in Figure 3.18.

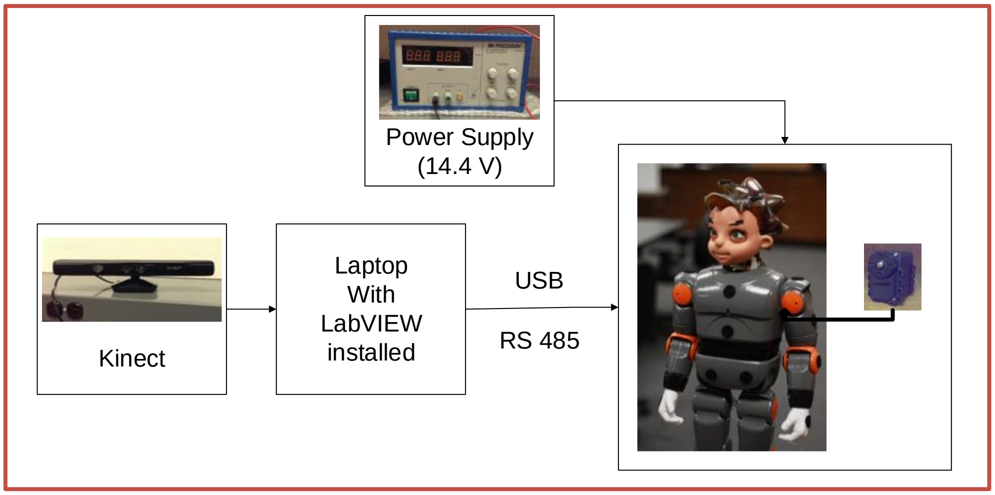


Figure 3.18 Zeno Hardware Overview

3.2.3 System Description- (Software framework)

The imitation learning system is a system proposed and implemented in both MATLAB and LabVIEW. During the initial phase of understanding the theory and code testing, the software MATLAB was used. For the part of application on Zeno robot, LabVIEW was the software used during the implementation. Both these tools are run on the operating system of Windows for the project.

MATLAB is a language for technical computing developed by MathWorks and released in 1984. It has since become popular in the industry and the academia. This is an interactive environment where programming can be done and visualized [57]. MATLAB is an acronym for Matrix Laboratory. The environment is programming language friendly with the ability to code in c, c++, java and python [57]. The software is also accessible in Linux and Macintosh OS. MATLAB also supports object oriented programming [58]. It also supports hardware such as web camera, raspberry Pi and Arduino micro controller [57].

MATLAB has packages and toolboxes rendering it useful for many direct applications. Some of the toolboxes available are Statistical toolbox, Robotics System

Toolbox, Computer toolbox to name a few. It is a very popular tool used in control systems engineering.

LabVIEW is a software developed by National Instruments. This software is preferred in industries due to the wide range of applications that it can be used in [59]. The programming done in LabVIEW is graphical and hence very easy to understand and design [59].

Due to the advantages of efficient coding, LabVIEW is used in a wide of application industries like Control, Signal Processing, Testing, Embedded Systems and also in Academia for teaching [60].

Similar to MATLAB, LabVIEW has toolkits which are made available to the users for different applications. These can be third party developed software solutions [61].

3.2.4 DMP Package

The inspiration for encoding Dynamic Movement Primitives in MATLAB and LabVIEW is from a ROS repository of Scott Niekum [62]. The package contained implementation of DMP with Fourier and radial approximation of basis functions. This implementation is robot- agnostic and general [62]. It was a PR2 dependent ROS repository where a DMP trajectory node was created.

A DMP project specific to the Zeno Robot is developed in this thesis. In this implementation of DMP, Training of weights is done by closed loop form solution and Gaussian basis kernels are used. To execute the trajectory on the Zeno robot, there must be a kinematics controller included in the project. For this purpose, a dynamixel controller is used to convert the DMP generated trajectory into serial output as required by the dynamixels for performing these motions.

LabVIEW contains a motion recorder code. When the trainer performs an ideal motion, it is recorded and saved using Microsoft Kinect. Kinect Initialization, Configuration, reading, skeletal mapping and inverse kinematics is done in this portion. This recorded motion is stored in a database and retrieved from it whenever required. This motion is then passed through the Zeno DMP code and then the DMP motion generated is replayed on the robot using the motion code. Thus separate code is written for motion capture, DMP generation and for redirecting motion onto the robot.

3.2.5 DMP Architecture Description

The experiment conducted involves an ideal motion to be displayed by the trainer. Kinect captures the poses of the trainer and sends it to the system. The system is a second order differential equations used for encoding the system. The motion points obtained from the system are fed to the robot. The subject is made to mimic the motion displayed by the robot. The mimicking ability of the subject is tested here. The motion displayed by the subject is captured by Kinect sensor. The original motion, and the subjects' motions are compared. The comparison in the ability is quantified using DTW. The complete system is described in the Figure 3.19.

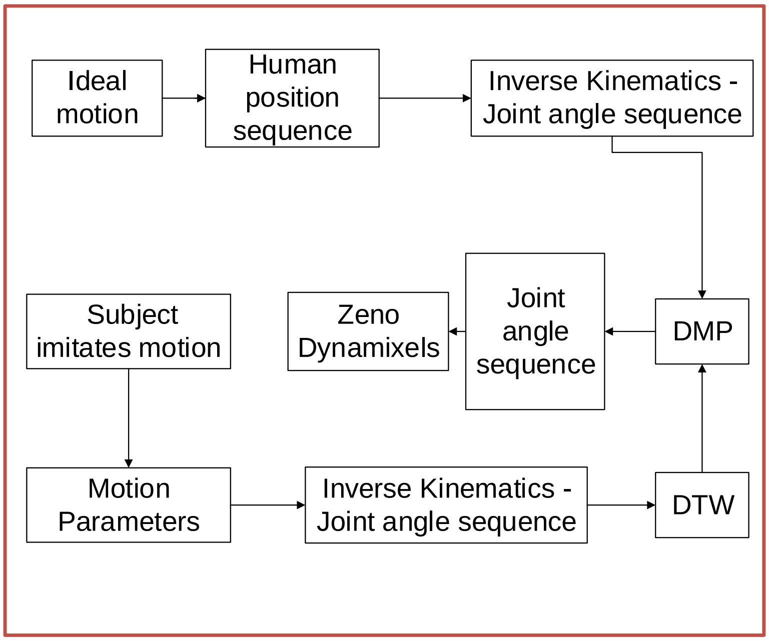


Figure 3.19 System overview

To learn the trajectory, the motion data obtained from the Kinect sensor must go through the DMP architecture. The DMP architecture consists of two systems – The system where the ideal trajectory is planned (The Gesture encoding system) and the system where trajectory is carried out (The adaptive system). The system is described in Figure 3.20.

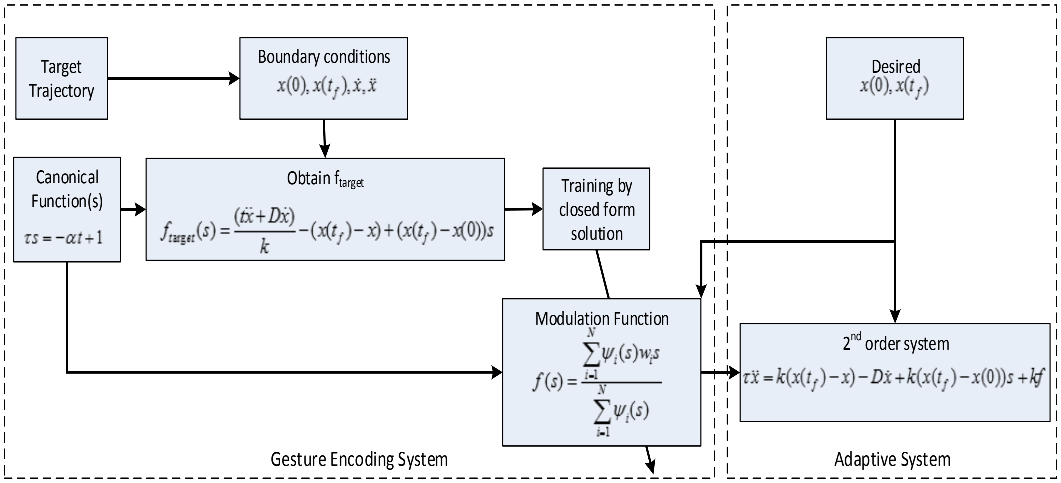

Target Trajectory
Boundary conditions $x(0), x(t_f), \dot{x}, \ddot{x}$
Desired $x(0), x(t_f)$
Canonical Function(s)
$\tau s = -\alpha t + 1$
Obtain f_{target}
$f_{target}(s) = \frac{(t\ddot{x} + D\dot{x})}{k} - (x(t_f) - x) + (x(t_f) - x(0))s$
Training by closed form solution
Modulation Function
$f(s) = \frac{\sum_{i=1}^{N} \psi_i(s) w_i s}{\sum_{i=1}^{N} \psi_i(s)}$
2nd order system
$\tau\ddot{x} = k(x(t_f) - x) - D\dot{x} + k(x(t_f) - x(0))s + kf$
Gesture Encoding System
Adaptive System

Figure 3.20 DMP architecture

The motion is the combination of simpler blocks also called primitives. A simple second order system describes the transformation system. The transformation system gives the current state of the limb. Position, velocity and the acceleration of the joints are the state variables of the transformation system. It is an attractor landscape which converges towards the goal. The goal here is the final point of the demonstrated trajectory. This equation is modified by a modulating function. The non- linear modulating function depends upon the state variable of a canonical equation. Canonical equation is a function with a negative slope. It is shaped so, so that it converges like the transformation equation. The error from the DMP transformation system and the target equation, through weights is given to the modulation function. The modified function is trained according to the error and is fed to the transformation system. The equations (5), (6), (3)and (2) define the system used in the experiment.

The training is done using closed form solution model. In order to train the DMP, the output of the training model, which are the weights obtained from calculation of the errors, and the phase variable from canonical form are used. The values of the parameters used in the DMP formulation is stated in the Table 3.1.

Table 3.1 Values

Constant	Value
n	25
α	1
k	100
D	$2\sqrt{k}$

Chapter 4

Dynamic Time Warping applied on Zeno

Dynamic Time Warping is a comparison measure that has been used in speech processing for a long time [63]. It is an algorithm which measures the similarity between two linear sequences. Sequences are warped non- linearly and the closest matched alignment is chosen. In this thesis, DTW is used as a comparison metric to compare two motion sequences. The similarity between two gestures are found using DTW. DTW has been used to study gestures before but not a lot of experimental data hold true to the claim that DTW can distinguish between gestures accurately. Present experiment validates this claim.

The success of this validation would provide success for the use of DTW algorithm as an objective measure of the impairment in limb motions for early diagnosis of Autism. Thus it could be used as a replacement of physician’s subjective judgment.

4.1. Dynamic Time Warping Algorithm

DTW is a distance measuring process which is similar to finding Euclidean distance. DTW compares the signals for shortest distance, thus maximizing similarity. Prior to comparing the gestures, DTW aligns the signal according to the shortest distance and then compares the motions. This is known as warping. As the motion of robot and human differs, temporal warping seems appropriate while comparing the gestures.

For two signals A and B of equal duration, DTW finds the shortest distance using the formula given in [64].

$$D(i,j) = d\left(A, B_j\right) + \min\{D(i-1, j-1), D(i-1, j), D(i, j-1)\} \tag{7}$$

Here d is given by the Euclidean distance which can be calculated as

$$d(A,B) = \sqrt{\sum_{i=1}^{n}(a_i - b_i)^2} \tag{8}$$

DTW algorithm assumes the first element to be perfectly aligned. Keeping an element from one series constant, distance of that element from all the elements in the second series is calculated. This procedure is repeated by keeping the elements from the second series constant.

Minimum Euclidean distance is chosen after this calculation. For two sequences, given in Figure 4.1, DTW algorithm when run, would compare them as depicted in Figure 4.2. Each point from one sequence would be mapped on the other sequence and the least difference is found. The grey lines indicate the closest or the least distance of one point in sequence with the data points in the second sequence.

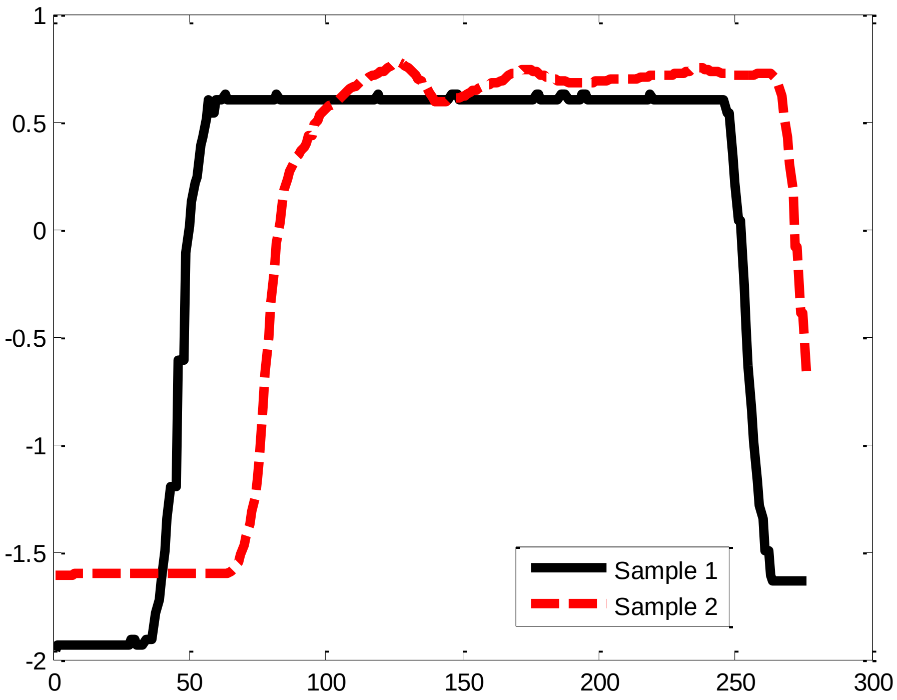


Figure 4.1 Input sequences examples for DTW algorithm

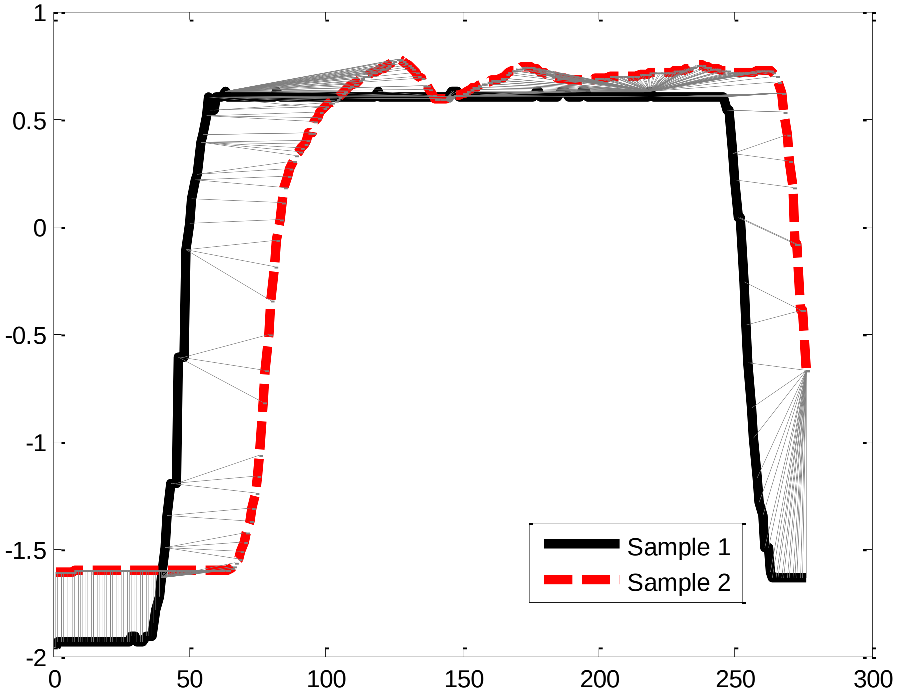


Figure 4.2 Comparison of the sequence using DTW algorithm

4.2. System Description for the validation of DTW algorithm

For validating the DTW algorithm, an experimental setup similar to that for the imitation learning system was implemented. The Zeno robot was used as a trainer and subjects were asked to imitate the gestures performed. Zeno performed upper body motions like waving, rubbing of tummy and fist bumping motion for this experiment. These hand motions were to be imitated by the subjects first without holding any weight in the hand and then thrice by carrying different weights. The set of weights to be carried by the subjects were 5, 10 and 15 pounds. The gesture mimicked by the subject is recorded through Kinect and then matched with the robot`s motion. This comparison is done using DTW algorithm. The system setup is shown in the Figure 4.3.

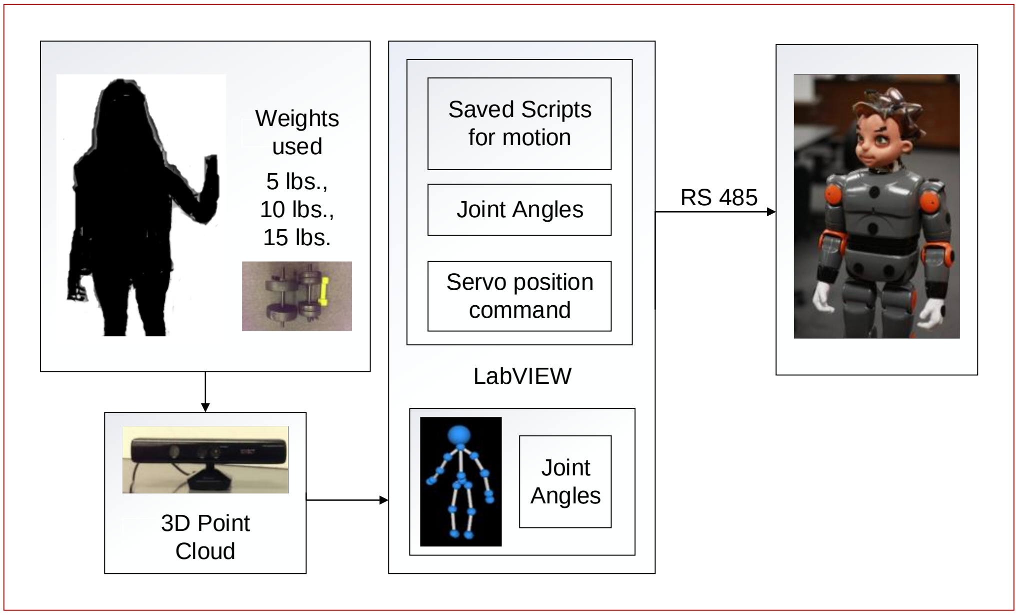


Figure 4.3 System overview

4.2.1 Software Framework

Once the hardware is set up, the software for running the experiment is brought in place. The program for running the experiment is coded in LabVIEW. The data collection and the DTW analysis is done using VI`s coded in LabVIEW. The data cleaning and data sorting is done using the software MATLAB and the Data analysis using hypothesis testing is done using Excel.

Zeno robot has two modes of operation. One is the scripted mode which runs a pre-stored set of gestures and the other mode is where the robot imitates a human [55]. For the purpose of this project, the scripted mode of Zeno is used. The project has two components: Kinect VI for capturing human motion data and the Read Motion VI for the motor control. The front panel of the two VI`s are shown in Figure 4.4 and Figure 4.5.

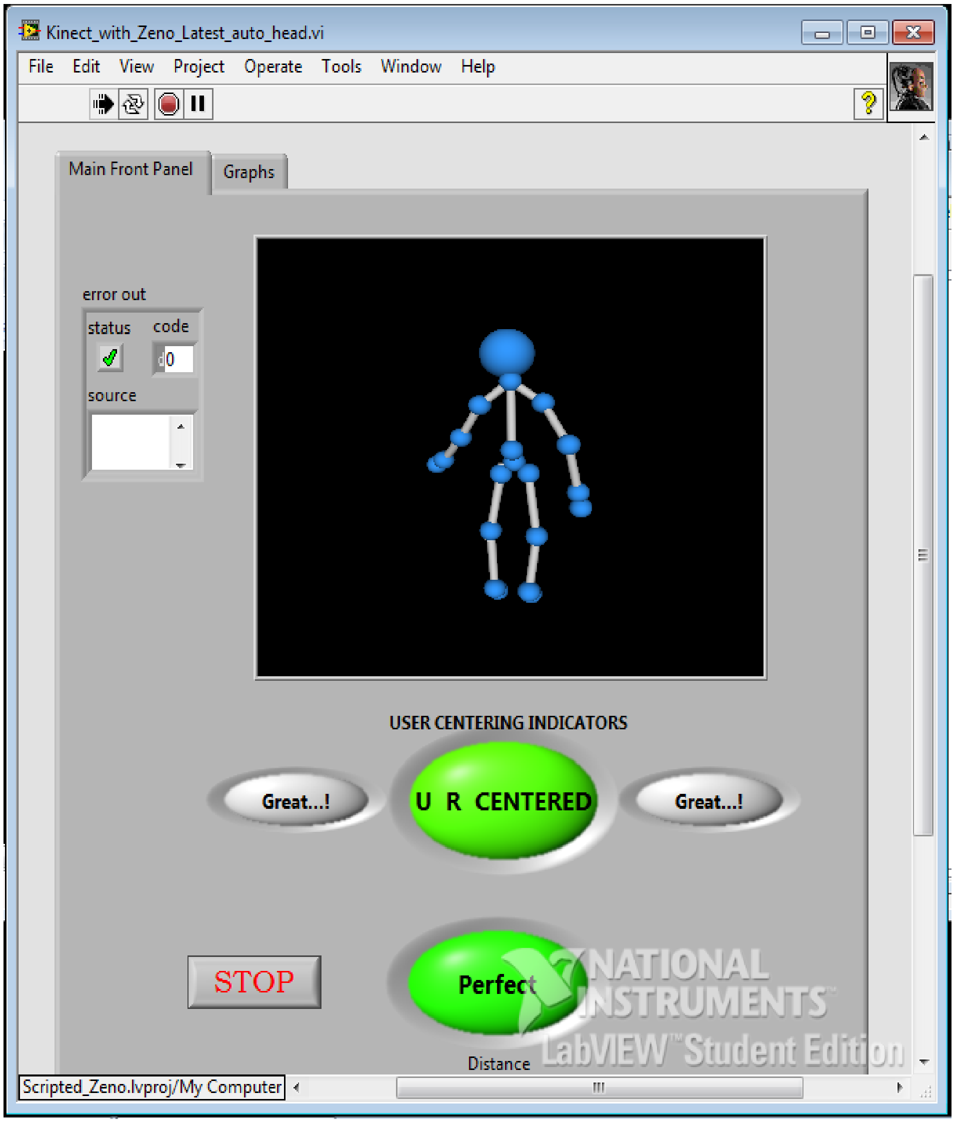
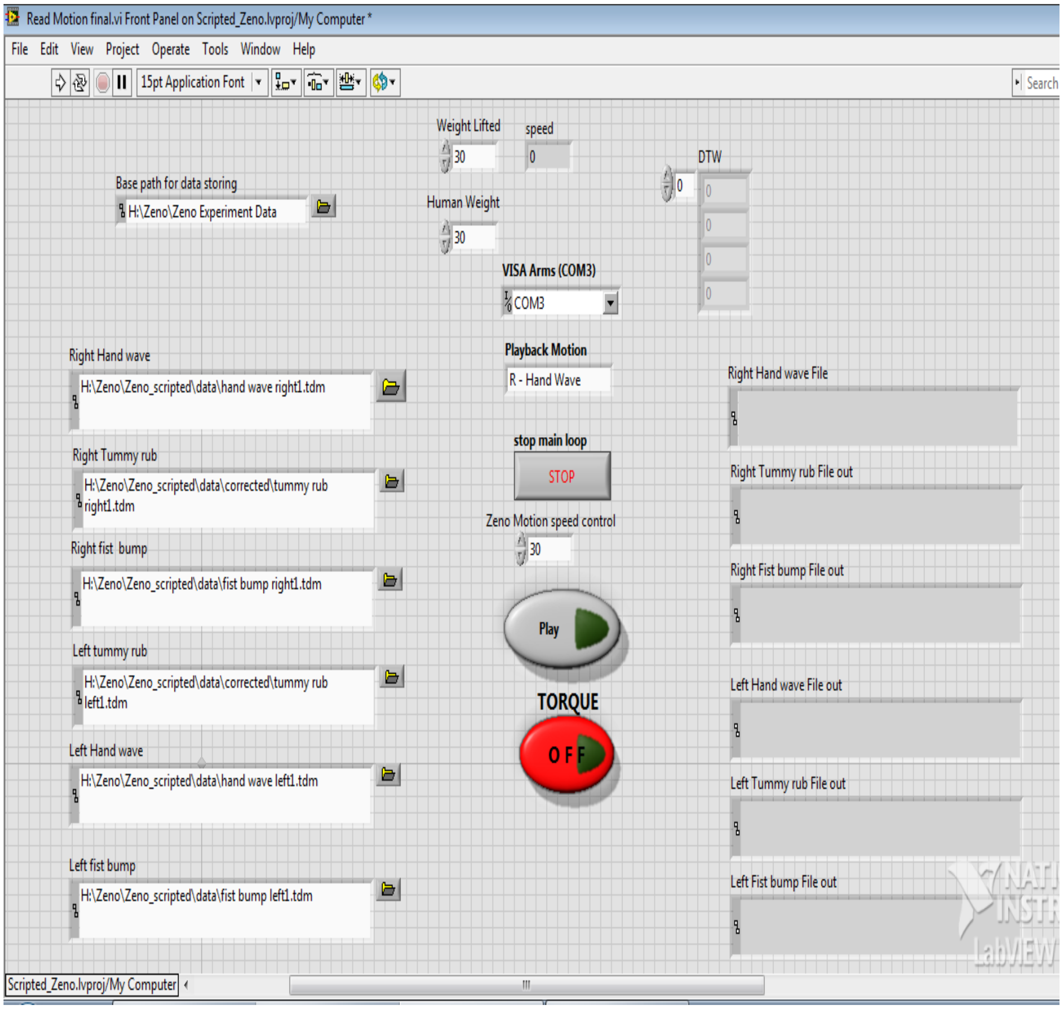

Kinect_with_Zeno_Latest_auto_head.vi
File Edit View Project Operate Tools Window Help
Main Front Panel
Graphs
error out
status
code
0
source
USER CENTERING INDICATORS
Great...!
U R CENTERED
Great...!
STOP
Perfect
Distance
NATIONAL INSTRUMENTS
LabVIEW Student Edition
Scripted_Zeno.lvproj/My Computer

Figure 4.4 Front Panel of Kinect VI



Read Motion final.vi Front Panel on Scripted_Zeno.lvproj/My Computer *
File Edit View Project Operate Tools Window Help
15pt Application Font
Search
STOP
Play
TORQUE
OFF
Scripted_Zeno.lvproj/My Computer

Weight Lifted 30

speed 0

Human Weight 30

DTW 0

0 0 0 0

Base path for data storing H:\Zeno\Zeno Experiment Data

VISA Arms (COM3) COM3

Playback Motion R - Hand Wave

stop main loop

Zeno Motion speed control 30

Right Hand wave H:\Zeno\Zeno_scripted\data\hand wave right1.tdm

Right Tummy rub H:\Zeno\Zeno_scripted\data\corrected\tummy rub right1.tdm

Right fist bump H:\Zeno\Zeno_scripted\data\fist bump right1.tdm

Left tummy rub H:\Zeno\Zeno_scripted\data\corrected\tummy rub left1.tdm

Left Hand wave H:\Zeno\Zeno_scripted\data\hand wave left1.tdm

Left fist bump H:\Zeno\Zeno_scripted\data\fist bump left1.tdm

Right Hand wave File

Right Tummy rub File out

Right Fist bump File out

Left Hand wave File out

Left Tummy rub File out

Left Fist bump File out

Figure 4.5 Front Panel of Motion VI

The Kinect VI first involves detection of the human skeleton. The human skeletal coordinates are then converted into joint angle coordinates using inverse kinematics. The joint angles of the arms are used in this experiment. In the meanwhile, the scripted motion VI runs the pre-recorded set of gestures. The pre-recorded set is a sequence of servo motion positions. These are read and then sent to the Dynamixel motors using RS485 communication. By selecting the appropriate serial port, the communication can be established.

The comparison of the joint angles is done using DTW. A MathNode is created for this where all four joint angles of Zeno and the human are first normalized. Then the DTW algorithm is run on the data set. The DTW value compares the corresponding joint angles of Zeno and the human. Thus the Output of one gesture when imitated is four DTW values. For every motion run on Zeno and imitated by a human, the joint angle values and DTW values are recorded in Excel sheets. This data is then used for validation of DTW.

4.2.2 Experiment

The designed experiment consisted of 100 volunteers participating in the procedure to test the algorithm. The experiment consisted of the robot being instructed to perform a series of upper body movements and the participants were asked to imitate Zeno. The set of motions that were performed by the robot were: Right Hand Wave, Right Hand Tummy Rub, Right Hand Fist Pump, Left Hand Wave, Left Hand Tummy Rub, and Left Hand Fist Pump. The imitated gestures performed by the participants were captured by Microsoft`s Kinect. The captured data i.e the joint positions were converted into joint angles and were stored in excel files. The process of performing the set of six motions was repeated four times. Barring the first set of motions, the next three processes involved the participants to perform the motions while holding onto 5 lbs. 10 lbs. and 15 lbs. respectively.

Before the experiments were conducted, each participant was given an Informed Consent Document to read and sign. Each volunteer was assigned a subject number. The motion capture data (the joint angle values) of a volunteer was stored with the subject ID.

For a motion like wave, the gesture is broken down into four trajectories of joint angle variations. The four trajectories of joint angle variations of the robotic motion are compared individually with the four joint angle trajectories of the subject. At the end of the experiment, 4 values of DTW will be obtained for alpha, beta, gamma and theta angles respectively.

These DTW values are then combined to produce one output value for one subject. Ranatunga [64] explains the method for combining the angle trajectories. For this purpose, the range of Zeno`s angle movement per motion is recorded. The range is obtained from the equation (9) [64]. A particular trajectory depicted as A can be any of the 3 gestures that were performed and the range is W.

$$W = MAX(A) - MIN(A) \tag{9}$$

This weight is calculated individually for each joint angle trajectory to produce $W_\alpha, W_\beta, W_\gamma$ and W_θ which represents the range of Zeno`s alpha angle, beta angle, gamma angle and theta angle respectively. The DTW values are combined as shown in equation (10) [64].

$$A_w = \frac{W_\alpha D_\alpha + W_\beta D_\beta + W_\gamma D_\gamma + W_\theta D_\theta}{W_\alpha + W_\beta + W_\gamma + W_\theta} \tag{10}$$

Where, $D_\alpha, D_\beta, D_\gamma$ and D_θ represent the DTW values calculated for Zeno`s alpha angle, beta angle, gamma angle and theta angle respectively.

Ranatunga [64] suggests that the angle used best in the imitation concerning these set of gestures are the theta and the beta angles. Thus this weighted average is taken only for the beta and the theta angles.

Chapter 5

Results from Experiments

5.1. Experimental results for the validation of DTW.

For every experiment involving Data Analysis, seven key steps are to be taken [65]. For any data to be analyzed, it must be collected properly, cleaned for errors, arranged for better understanding and then analyzed for obtaining what was set to attain. Once the basic steps of deciding the objective/ outcome of the experiment, which here is to prove the reliability of use of DTW algorithm as a measure of level of limb impairment, we move to the steps of data collection, data modelling, data cleaning and then finally data analysis.

5.1.1 Data Collection

For the Validation of DTW, 56 people were recruited. Each person was asked to imitate Zeno. First set of motions - Wave, Rubbing of tummy and Fist Bump from both the hands – were carried out without any weight in the volunteers' hand. For the next three sets, weight of 5 lbs., 10 lbs. and 15 lbs. were held.

The aim of our experiment is to validate the reliability of DTW algorithm as a measure of the level of limb impairment. For the purpose of emulating limb impairment, individuals were made to hold weights in hands while imitating the robot. Table 5.1 shows a portion of excel sheet as saved from the experiment. Each row depicts DTW results for a particular gesture. Gestures were performed in the same order as it was mention in the experiment setup explained in chapter 5. The value under the column of Theta corresponds to the DTW value obtained on comparison of the variations in theta angle of Zeno and human when performing the different gestures. This table is continued to hold the data for the trials of 5 lbs., 10 lbs. and 15 lbs.

Table 5.1 DTW values of Joints of Subject 1 when no weight lifted

Type of Motion	Human Weight	Weight Lifted	Speed	Theta	Gamma	Beta	Alpha
R Hand Wave	110	0	30	64.89976	113.8699	99.75154	208.1274
R Tummy Rub	110	0	30	37.66691	183.8677	105.4256	163.5452
R Fist Bump	110	0	30	49.67777	46.88298	13.43332	109.9783
L Hand Wave	110	0	30	46.98202	280.3508	64.36042	227.2815
L Tummy Rub	110	0	30	31.44966	131.9934	137.7247	95.98658
L Fist Bump	110	0	30	52.09961	111.806	69.97854	106.4875

Similar to Table 5.1, data was collected for each weight for all the subjects. Figure 5.1, Figure 5.2, Figure 5.3 and Figure 5.4 are the graphs depicting the variation of the four angles for a right hand wave motion between Zeno and human when imitation was done without any weights held in hand. It can be seen that the figure for theta and beta angle are similar with just it being temporally shifted. Whereas for the motion of a hand wave, the variations in alpha and gamma angle are not descriptive of the experiment. This can be concluded due to the limited usage of those angles during a hand wave motion.

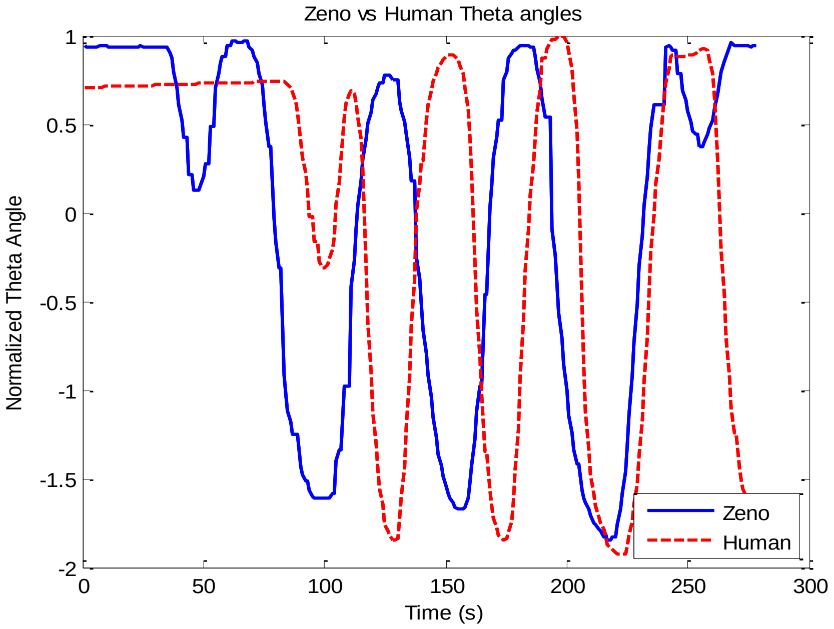


Figure 5.1 Subject 1 Right Hand Wave - Theta Angle comparison

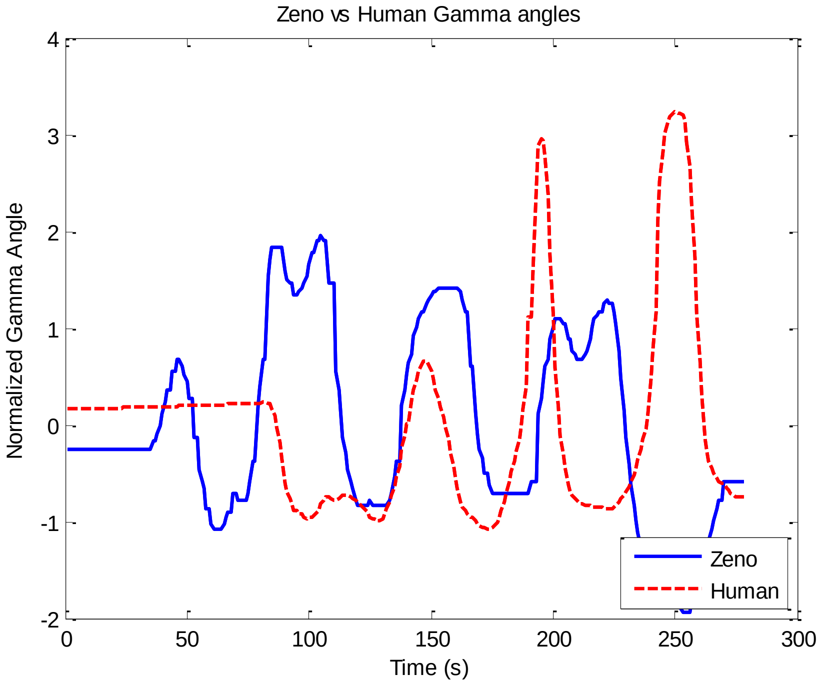


Figure 5.2 Subject 1 Right Hand Wave - Gamma Angle comparison

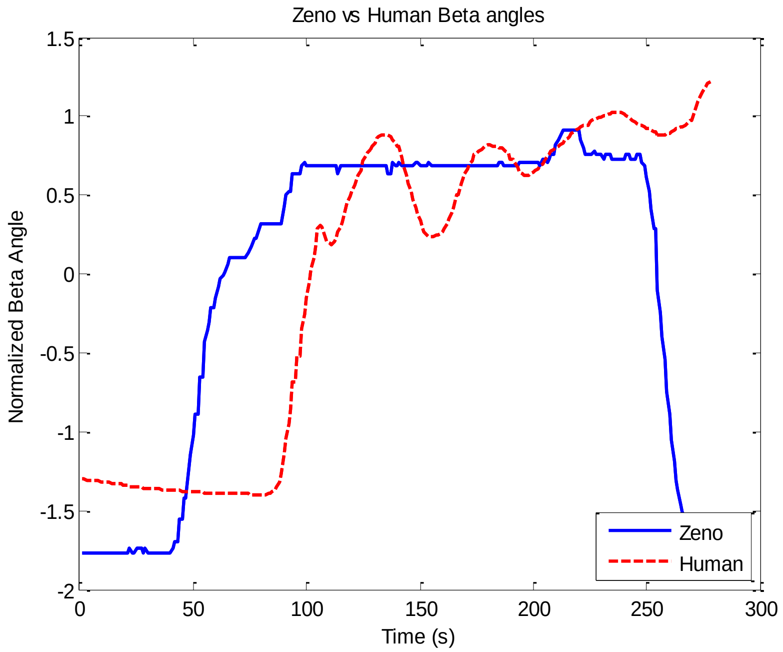


Figure 5.3 Subject 1 Right Hand Wave - Beta Angle comparison

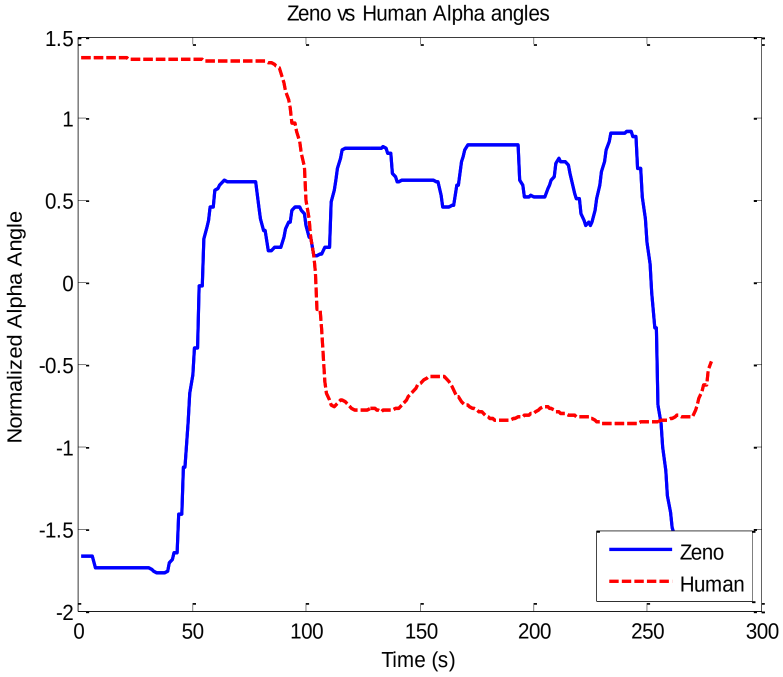


Figure 5.4 Subject 1 Right Hand Wave - Alpha Angle comparison

Figure 5.5, Figure 5.6, Figure 5.7and Figure 5.8 shows the comparison between all the four angles for subject 1 for the gesture of Right Tummy Rub.For this set of graphical results, it can be seen that a tummy rub motion affects the alpha and theta angle more than the beta and the gamma angle. This can be seen from the result as without holding any weight, the action of zeno and that of the human subject are similar.

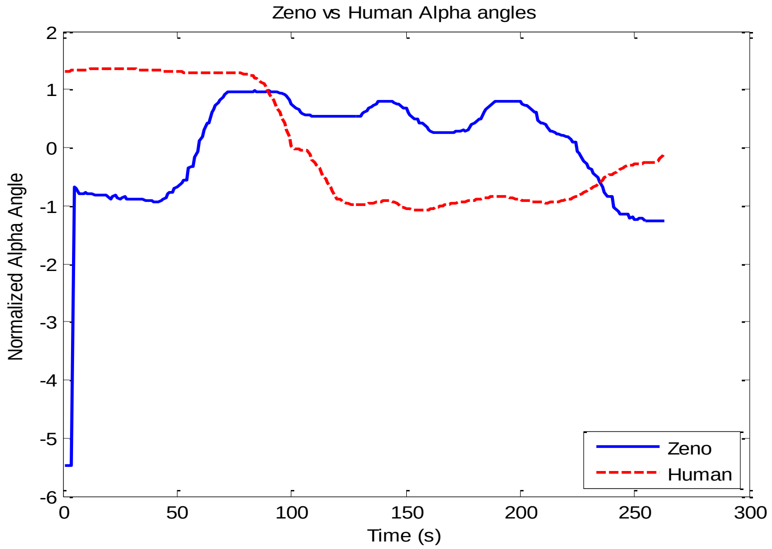


Figure 5.5 Subject 1 Right Tummy Rub - Alpha Angle comparison

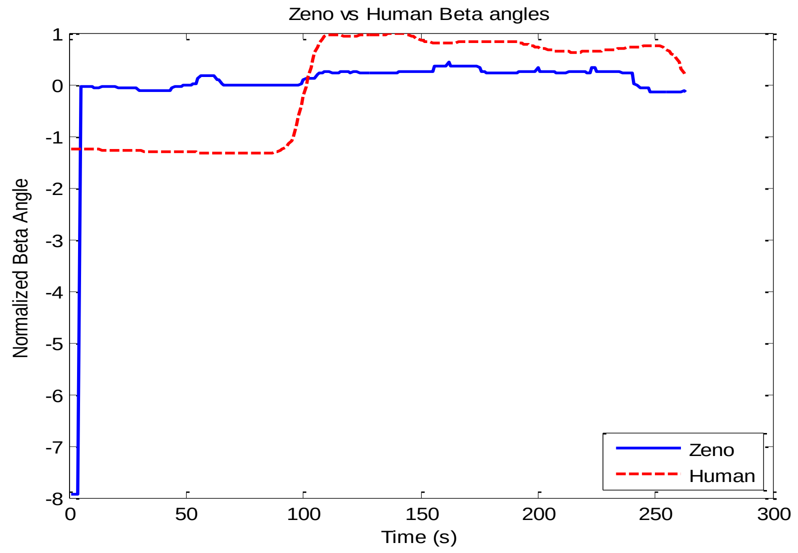


Figure 5.6 Subject 1 Right Tummy Rub - Beta Angle comparison

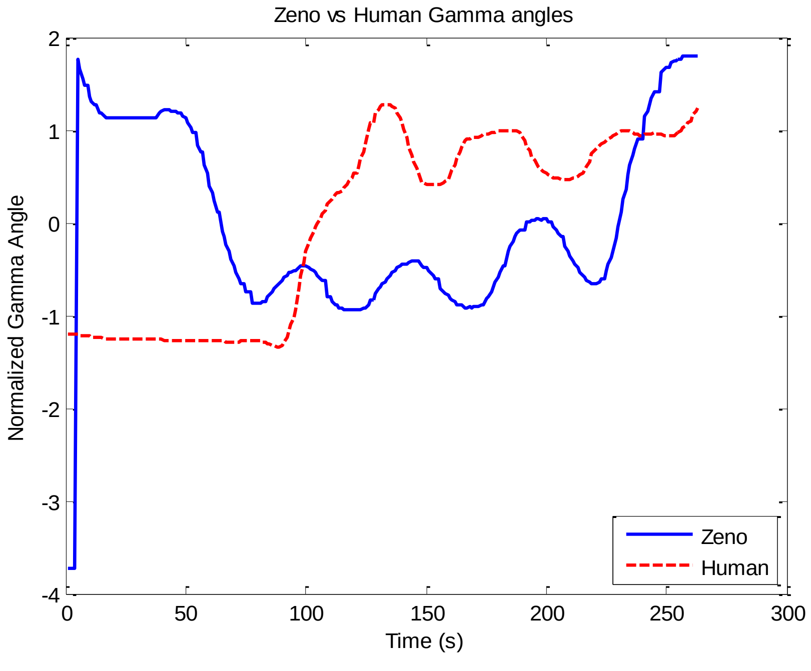


Figure 5.7 Subject 1 Right Tummy Rub - Gamma Angle comparison

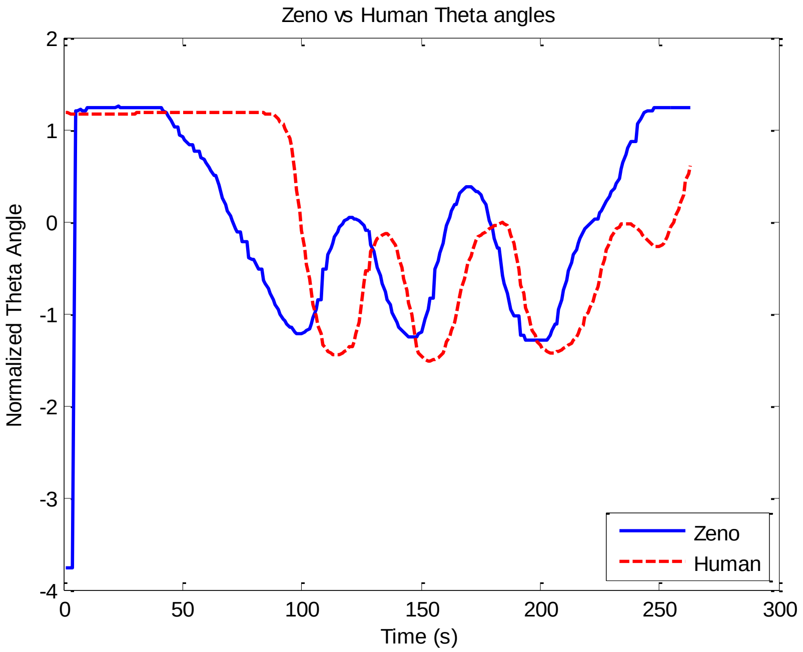


Figure 5.8 Subject 1 Right Tummy Rub - Theta Angle comparison

Figure 5.9, Figure 5.10, Figure 5.11 and Figure 5.12 depicts the variation in Right hand fist bump between Subject 1 and Zeno. The fist bump experiment without weights held in the hand has all the four angles varying throughout the motion. The human`s motion and the robot`s motion are almost closely related with some temporal delay.

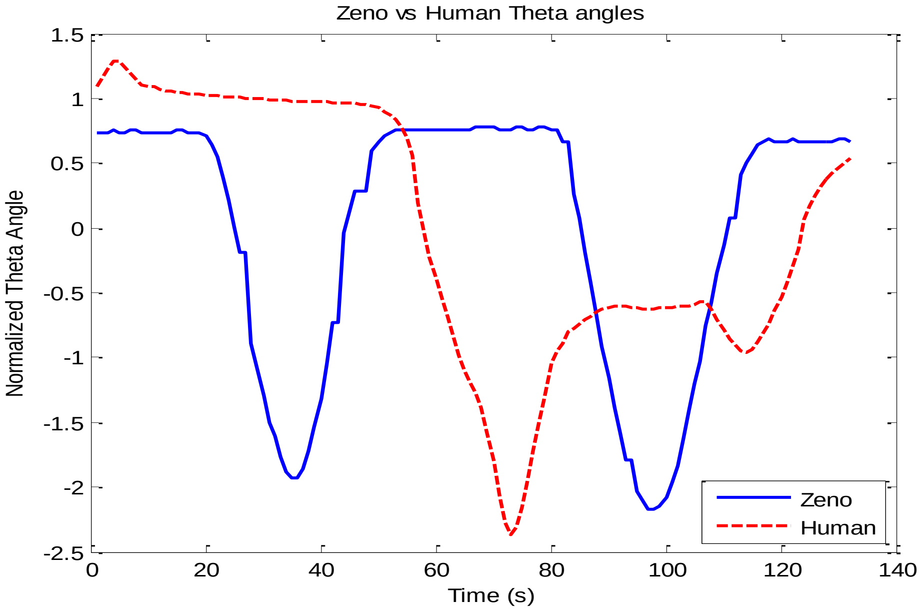


Figure 5.9 Subject 1 Right Fist Bump - Theta Angle comparison

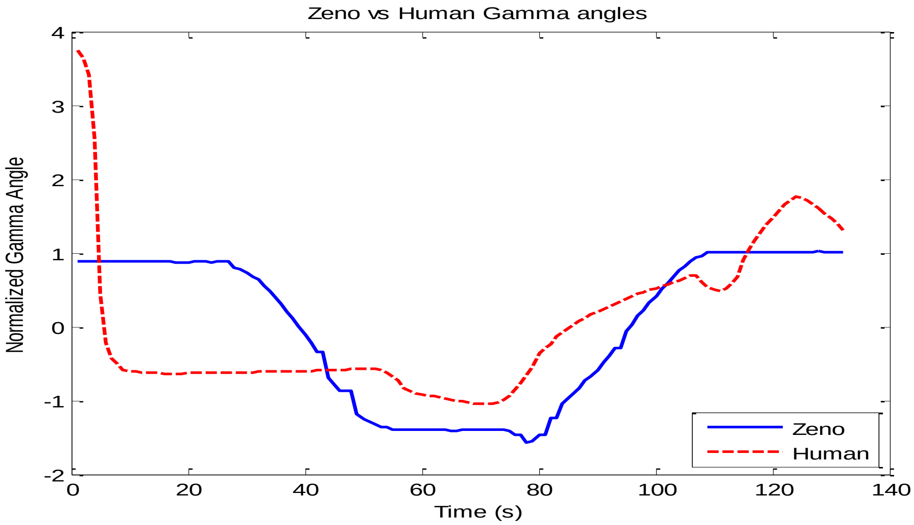


Figure 5.10 Subject 1 Right Fist Bump - Gamma Angle comparison

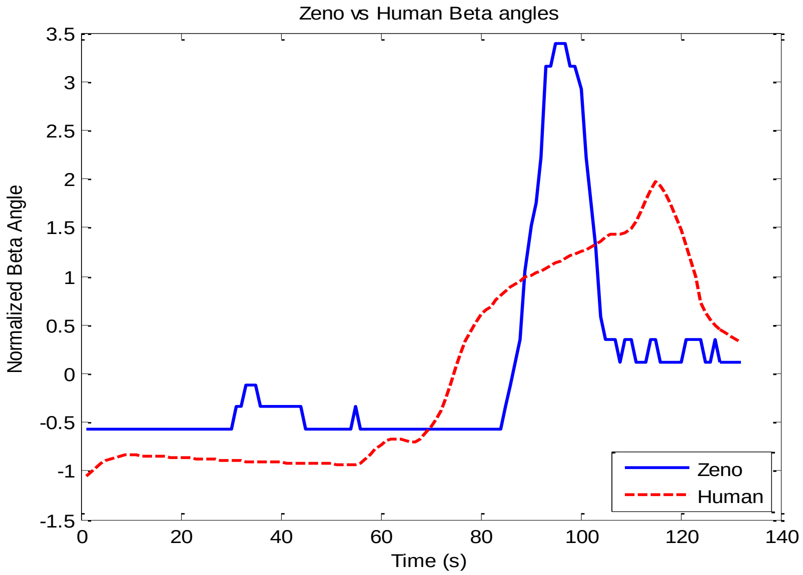


Figure 5.11 Subject 1 Right Fist Bump - Beta Angle comparison

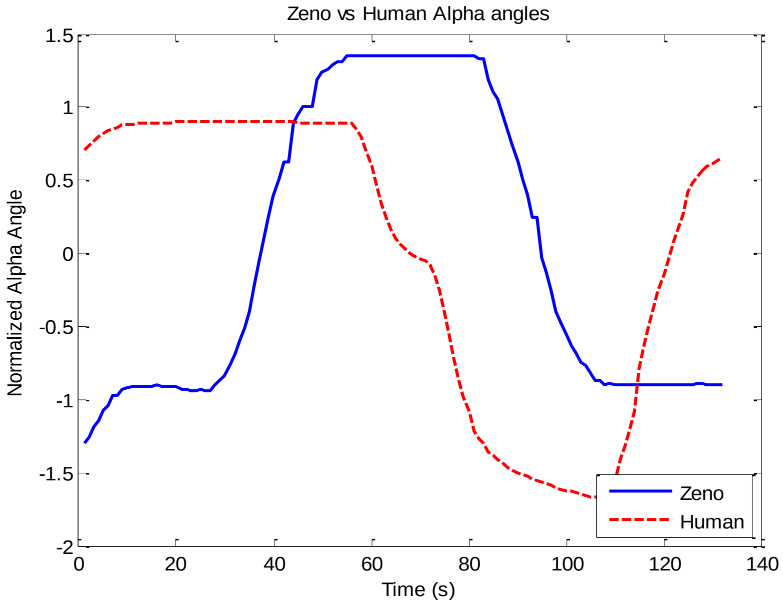


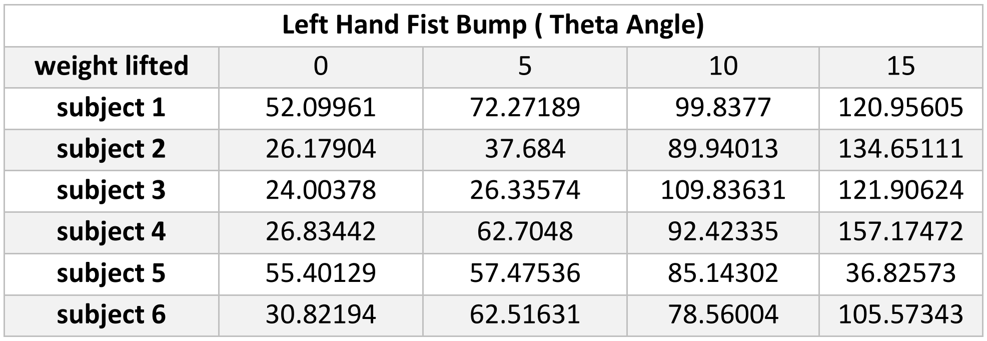
Figure 5.12 Subject 1 Right Fist Bump - Alpha Angle comparison

5.1.2 Data Sorting/ Modelling

As seen from Table 5.1, each subject produced 6 sets of data for each trial. As there were four trials (0, 5, 10 and 15 lbs.), each subject contributed to 24 sets of data. For the purpose of analyzing the reliability of DTW, we compared the DTW of each gesture with weight and without weight. Hence, for a right hand wave, every volunteer`s right hand wave`s DTW comparison data while imitating without weight must be grouped together, DTW data obtained while holding 5 lbs. must be grouped together and so on. This task was accomplished using MATLAB. A MATLAB code was written to read all the excel files, sort through the data and separate the data according to gesture first and then according to weights lifted and finally according to the joint angles.

Thus at the end of Data sorting there were 6 excel workbooks, one excel workbook dedicated for one gesture. Each excel workbook contained four sheets one per joint angle. Each sheet in a workbook contained four columns and 56 rows where columns were labelled as no weight, 5 lbs. 10lbs. and 15lbs. and the rows denoted the number of participants. Table 5.2 depicts a portion of excel sheet at the end of data sorting. The leftmost column labels the data according to the subject number. The next four columns are the DTW values obtained after comparing the Left Hand Fist Bump for the

Table 5.2 A Part of Sorted Data



Left Hand Fist Bump (Theta Angle)				
weight lifted	0	5	10	15
subject 1	52.09961	72.27189	99.8377	120.95605
subject 2	26.17904	37.684	89.94013	134.65111
subject 3	24.00378	26.33574	109.83631	121.90624
subject 4	26.83442	62.7048	92.42335	157.17472
subject 5	55.40129	57.47536	85.14302	36.82573
subject 6	30.82194	62.51631	78.56004	105.57343

Table 5.2—*Continued*

subject 7	34.05075	54.99555	38.25065	455.93224
subject 8	27.77337	53.44356	37.65265	99.95701
subject 9	29.9949	66.50887	105.16471	108.48554
subject 10	25.01447	70.90899	82.46578	102.72097

Once the data is sorted, the DTW values for all the four angles for a particular subject is combined according to equation (11) and the results obtained are depicted in the Table 5.3. The columns correspond to the weighted DTW value for Left Hand Fist Bump motion after combining all the four joint angles.

Table 5.3 Weighted DTW values

Left Hand Fist Bump				
weight lifted	**0**	**5**	**10**	**15**
subject 1	54.08613	68.64466	40.56976	72.14034
subject 2	31.19171	40.81682	44.42366	65.896
subject 3	27.78762	31.21612	52.96732	28.5686
subject 4	28.8792	60.20399	51.71011	53.71894
subject 5	58.31141	62.01287	41.9411	40.35882
subject 6	59.88418	60.93008	65.76842	65.72475
subject 7	35.99444	57.45825	43.95732	59.68432
subject 8	33.1959	46.4773	42.82166	43.72914
subject 9	34.29835	32.9339	67.47913	80.79222
subject 10	28.72829	69.95811	35.42698	46.20269
subject 11	61.57486	63.68365	47.52114	49.36075
subject 12	35.09508	45.52151	57.16369	67.15715
subject 13	72.86421	86.1781	78.46126	82.39858
subject 14	53.43329	62.18842	58.46194	53.67367
subject 15	55.12059	57.73593	42.79794	54.21049

5.1.3 Data Cleaning

The next step is to improve the data quality. Not every subject imitated the robot as expected. Errors arise due to improper gesture imitation, use of wrong hand to name a few. These erroneous data`s would lead to the presence of outliers. An outlier is indicative of measurement error. It is the data which does not lie in the same data pool. It often lies at a significant distance from the rest of the data. Potential outliers were found using the inter quartile range. A data set is divided into four equal parts. The lower and upper quartiles are found by the equation (11)and (12).

$$Q_1 = \frac{1}{4}(n+1)th\ value \tag{11}$$

$$Q_3 = \frac{3}{4}(n+1)th\ value \tag{12}$$

$$\mathrm{IQR} = \ Q_3 - Q_1 \tag{13}$$

Any value less that $Q_1 - (1.5 \times IQR)$ or more than $Q_3 + (1.5 \times IQR)$ is rejected. Table 5.4 shows the cleaned data set. The Table 5.4 is almost the same as the Table 5.3

Table 5.4 Cleaned Data Set

Left Hand Fist Bump				
weight lifted	**0**	**5**	**10**	**15**
subject 1	54.08613	68.64466	40.56976	72.14034
subject 2	31.19171	40.81682	44.42366	65.896
subject 3	27.78762	31.21612	52.96732	28.5686
subject 4	28.8792	60.20399	51.71011	53.71894
subject 5	58.31141	62.01287	41.9411	40.35882
subject 6	59.88418	60.93008	65.76842	65.72475
subject 7	35.99444	57.45825	43.95732	59.68432
subject 8	33.1959	46.4773	42.82166	43.72914
subject 9	34.29835	32.9339	67.47913	80.79222
subject 10	28.72829	69.95811	35.42698	46.20269
subject 11	61.57486	63.68365	47.52114	49.36075
subject 12	35.09508	45.52151	57.16369	67.15715
subject 13	72.86421		78.46126	82.39858
subject 14	53.43329	62.18842	58.46194	53.67367
subject 15	55.12059	57.73593	42.79794	54.21049

5.1.4 Data Analysis

The cleaned dataset is now analyzed using a two sample one tail hypothesis testing (0). We are interested in analyzing how each pair of sample means compare to each other and what these observations translate to.

$$\mu_{0lb} < \mu_{5lb}, \mu_{0lb} < \mu_{10lb}, \mu_{0lb} < \mu_{15lb} \tag{14}$$

In order to test this, one tail hypothesis testing is done with two samples. Column of DTW data for each weight is individually analyzed with the DTW value of no weight. The hypothesis testing is done in MATLAB.

In order to prove equation (14) two hypothesis are to be determined according to hypothesis testing. Both the hypothesis must be contradictory such that if one were true, it would be not statistically possible to state the other to be true as well. Thus statement (14) must form one hypothesis. The hypotheses for this experiment are as follows:

H_0(Null hypothesis): The mean DTW value for the case of 5lbs weight is equal to the mean DTW value for the case of 0 weight.

$$\mu_5 = \mu_0 \tag{15}$$

H_a(Alternate hypothesis): The mean DTW value for the case of 5lbs weight is greater than mean DTW value for the case of 0 weight.

$$\mu_5 > \ \mu_0 \tag{16}$$

Because of the nature of the Alternative Hypothesis, we proceed with a one tail hypothesis test, which is conducted on our data set. Here the sample mean and the population mean are compared. A t-test is conducted on the sample dataset. A confidence bound of 95% is taken. Thus for the sample size of 56 and the probability of observing the null hypothesis at 0.05, the value for t statistic score rejection region is >1.645 (0). In order to find the test statistics, from the data summary, the mean, the standard deviation and sample count is obtained.

Mean, Standard Deviation and Count Value is noted and used for finding the value of t which is given in equation (17)

$$\mathrm{t} = \frac{\bar{x}_5 - \bar{x}_0}{\sqrt{\sigma_1^2/n_1 + \sigma_2^2/n_2}} \tag{17}$$

Here, $\bar{x}_5$ is the mean of 5lbs. weight dataset, $\bar{x}_0$ is the mean of 0 weight data set, $\sigma_1^2$is the variance of the 5lbs weight data set and σ_2^2 is the variance of 0 weight dataset and $n_1$and n_2 are the respective sample sizes. On solving (17) for the values from the Table 5.4, the value of t comes to be 5.07.

In order to reject the null hypothesis, the value of t must lie in the rejection region. As the rejection region is >1.645, we reject the Null hypothesis. Thus we can say that the mean of 5lbs weight is higher than the mean of 0 weight at a level of significance of 0.05%.

This process is done for all the three weights through all the four joint angles for all the six gestures. Thus this process is done 24s times while keeping the same null and alternate hypothesis Table 5.5 summarizes the result obtained for all the processes. The mean of the sample of 5lbs is greater than the mean of the sample obtained for no weight for all the gestures except for the left hand tummy rub. Similarly, the sample mean of 10lbs is also significantly greater for all the gestures except the right hand tummy rub. The sample mean for 15lbs is greater than no weight for all the gestures.

Table 5.5 Summary of Data Analysis

Gesture	Weights	5lbs.	10lbs.	15lbs.
R Hand Wave		Reject H0	Reject H0	Reject H0
R Tummy Rub		Reject H0	Not significant	Reject H0
R Fist Bump		Reject H0	Reject H0	Reject H0
L Hand Wave		Reject H0	Reject H0	Reject H0
L Tummy Rub		Not significant	Reject H0	Reject H0

Table 5.5—*Continued*

L Fist Bump		Reject H0	Reject H0	Reject H0

5.2. Experimental results for the Gesture Imitation Project

The gesture imitation project in LabVIEW was implemented for the same hand wave gesture as used in the DTW validation experiment. The trajectory encoded/ saved for the DTW experiment was used and from that motion database this gesture was extracted and replayed on the robot. The gesture was generated again using DMP to utilize all the properties and advantage for this project.

5.2.1 Software results of DMP in LabVIEW

The Figure 5.13 depicts the results obtained from inputting the saved right hand wave motion into the DMP algorithm in LabVIEW. The Figure 5.13 shows the DMP to be following the original algorithm. All the four angles involved in the right hand wave motion are DMP trained separately. The four mini graphs in the top row are the original signal which is recorded by the trainer. The four mini graphs at the bottom row are the DMP generated joint angle trajectories.

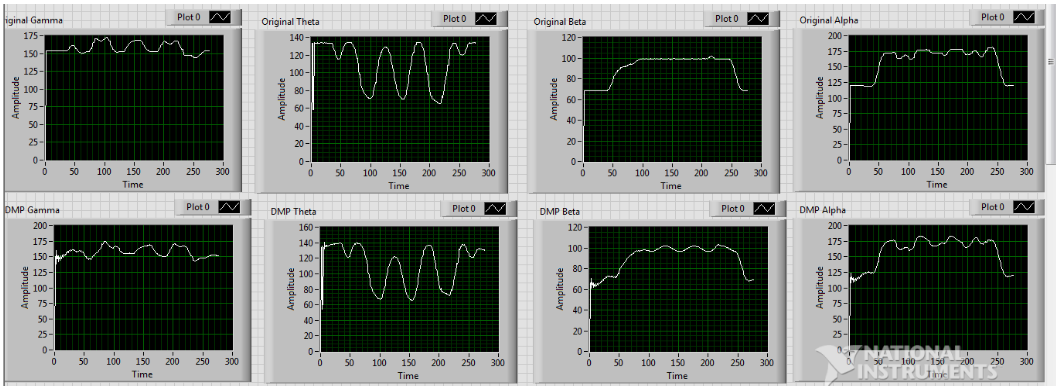


Figure 5.13 DMP in LabVIEW

These four joint angle trajectories are then sent to the Dynamixel VI and then converted to RS485 communication format to send to the robot. As the robot performs this motion, Kinect captures the imitation performed by a subject.

The motion of the robot could be changed easily by changing a few parameters, thus confirming the properties of DMP. The robot`s motion data was stored and was used for plotting the joint angle sequence in MATLAB. Here for the wave motion, the variation in joint angles is analyzed for changes with DMP parameters. Changes in τ in the range of 1 and 2 is shown through Figure 5.14, Figure 5.15 and Figure 5.16. τ is the temporal scaling factor. So it can be seen how the change in its value affects the change in the DMP trajectory. As the value of τ increases, the motion gets delayed in time.

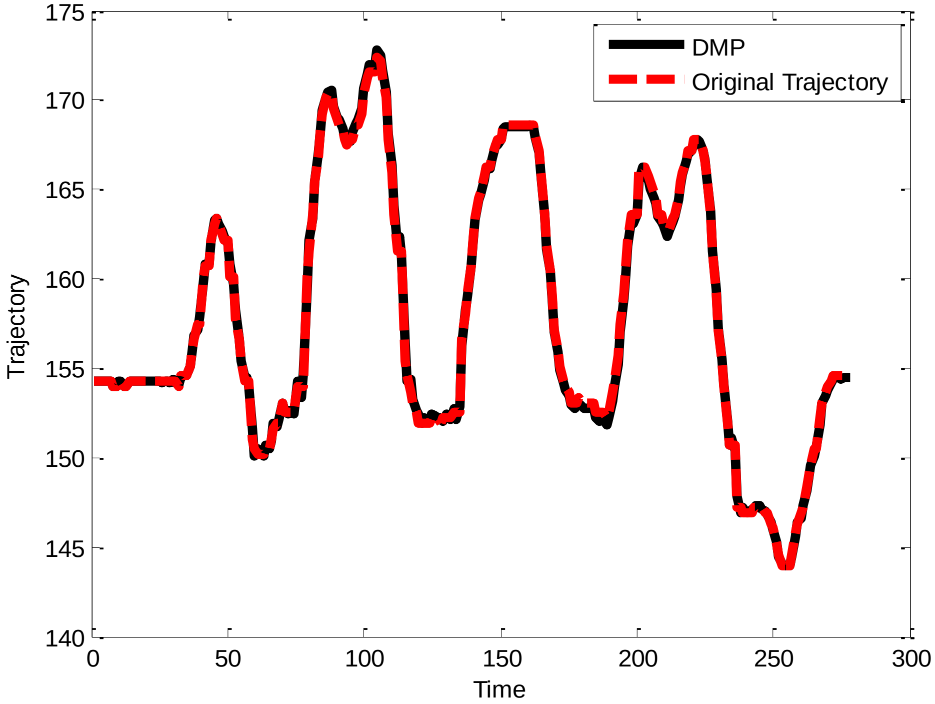


Figure 5.14 Variation in gamma angle through time for tau = 1

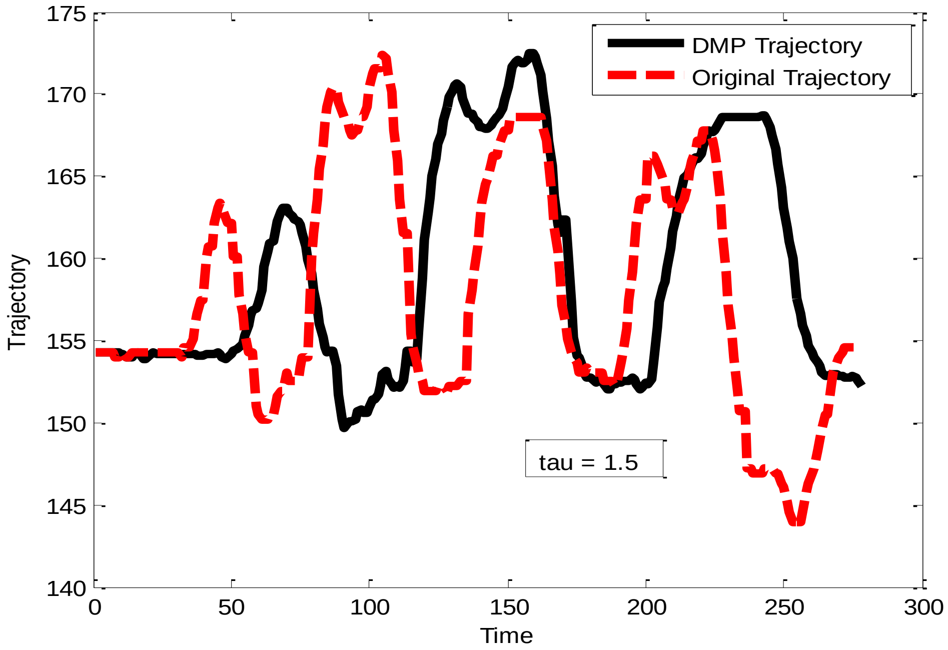


Figure 5.15 Variation of gamma angle through time for tau = 1.5

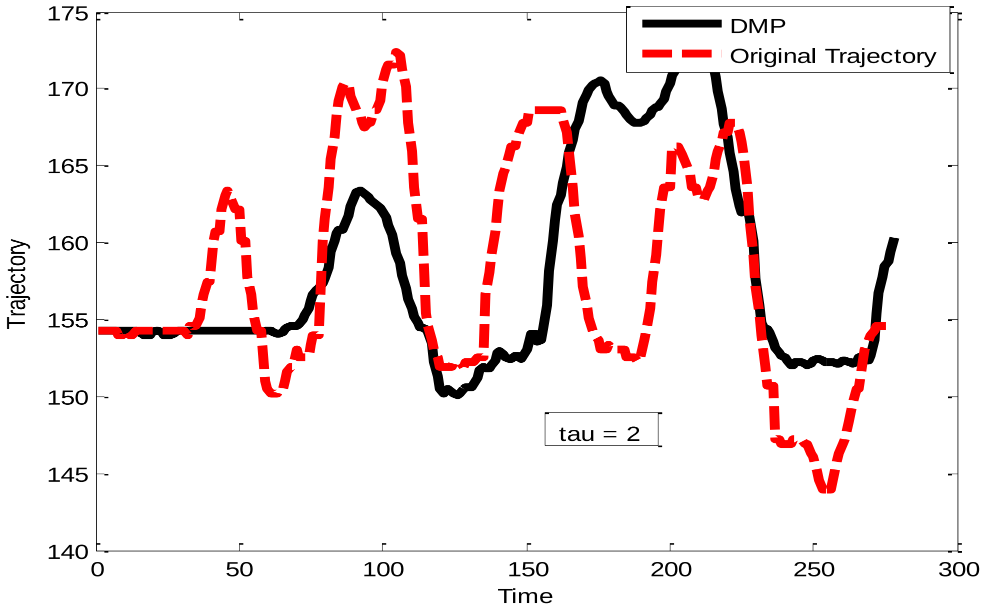


Figure 5.16 Variation of gamma angle through time for tau = 2

Changes in the width of Gaussian also changes the DMP trajectory. The effect of this variable is also tested and results are displayed in Figure 5.17, Figure 5.18, Figure 5.19, Figure 5.20 and Figure 5.21. For the range of 1 to 50. The lower the width of the Gaussian, lower is the frequency of the motion sequence. It can be seen that as h increases, the higher frequencies are also translated correctly. The testing was done for 1, 5, 10, 25 and 50. Thus amplitude and the frequency components are altered by just tweaking a parameter.

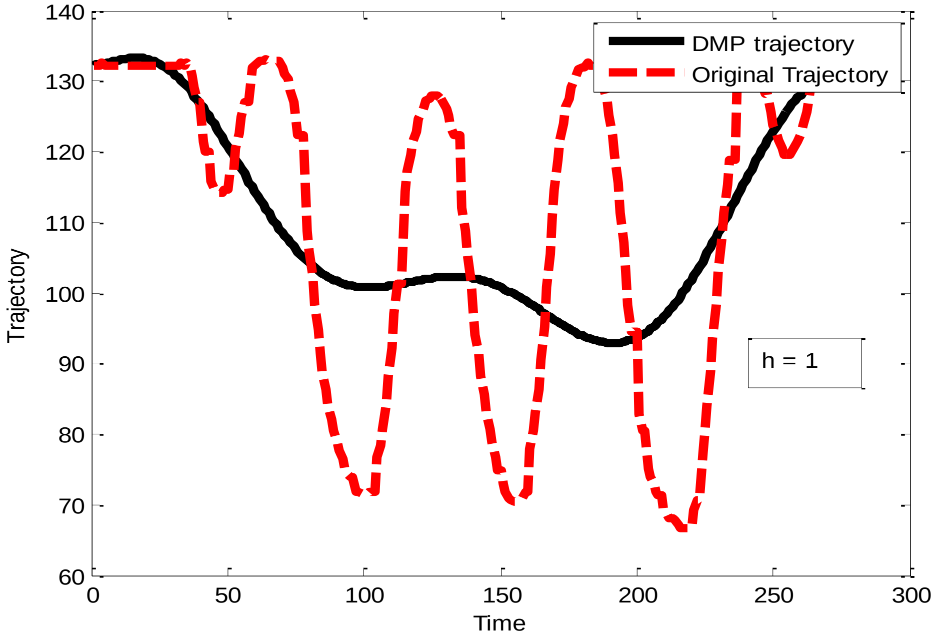


Figure 5.17 Variation of theta angle through time for t width of Gaussian as 1

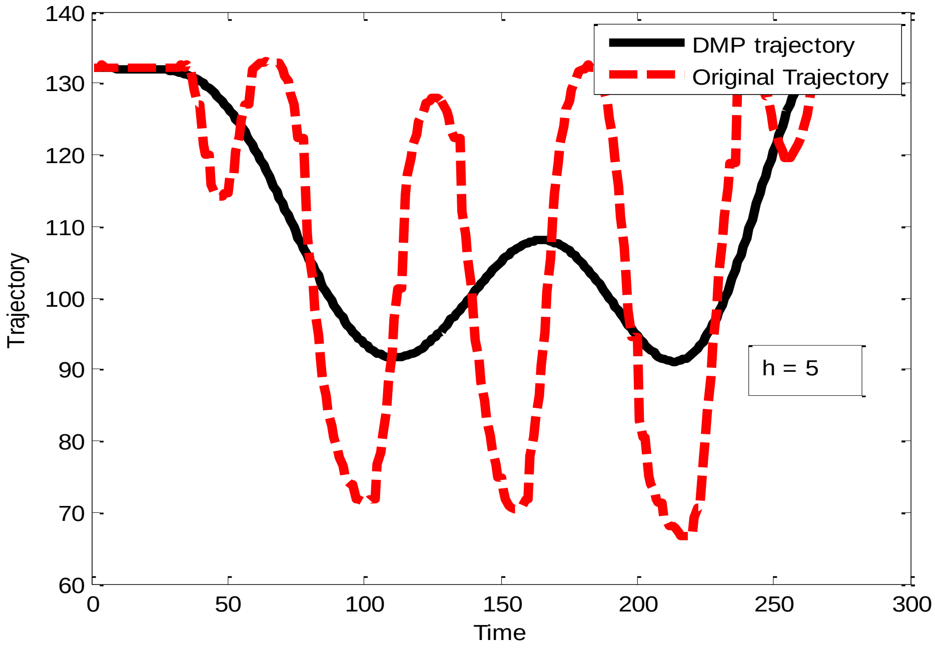


Figure 5.18 Variation of theta angle through time for t width of Gaussian as 5

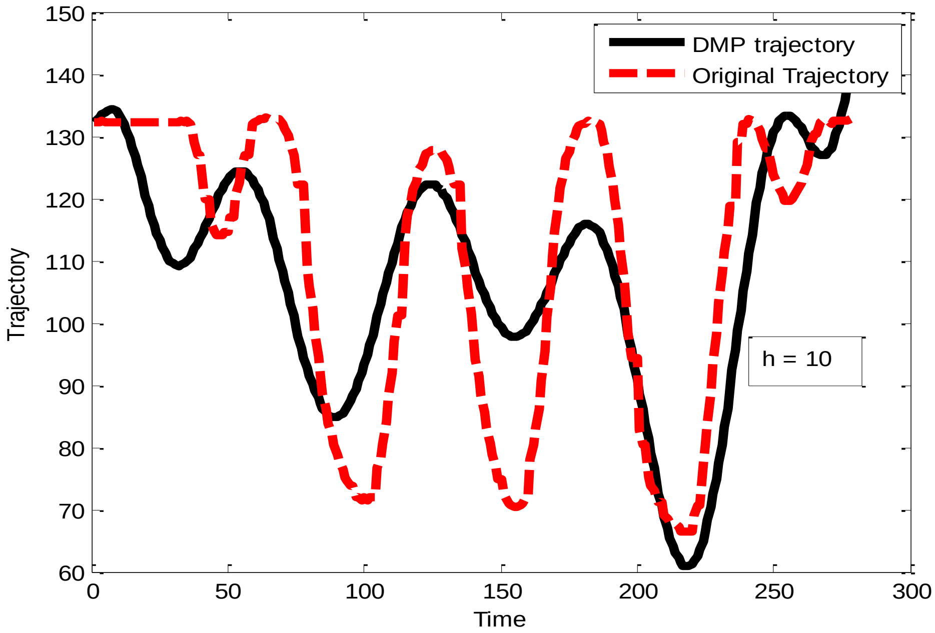


Figure 5.19 Variation of theta angle through time for t width of Gaussian as 10

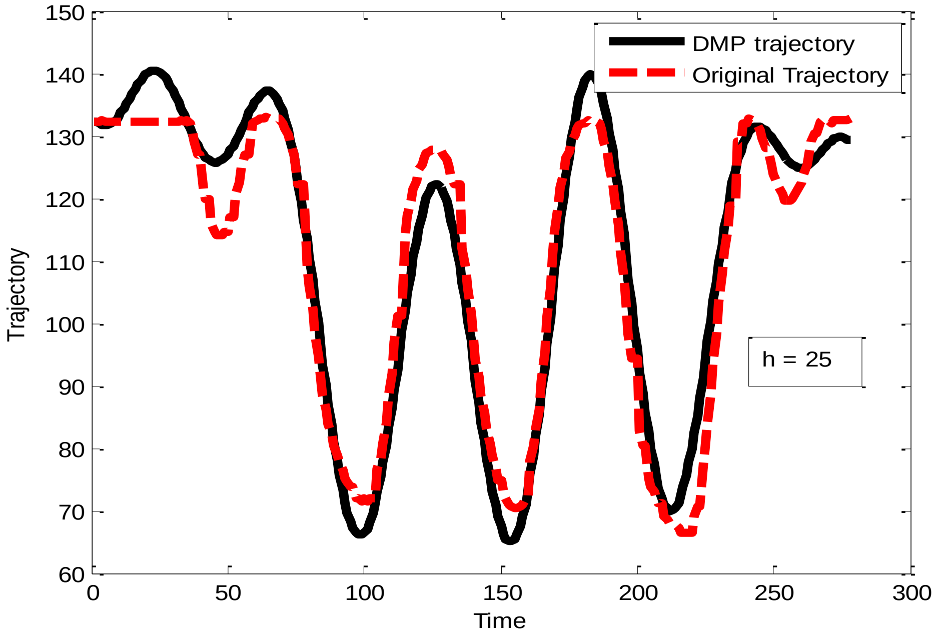


Figure 5.20 Variation of theta angle through time for t width of Gaussian as 25

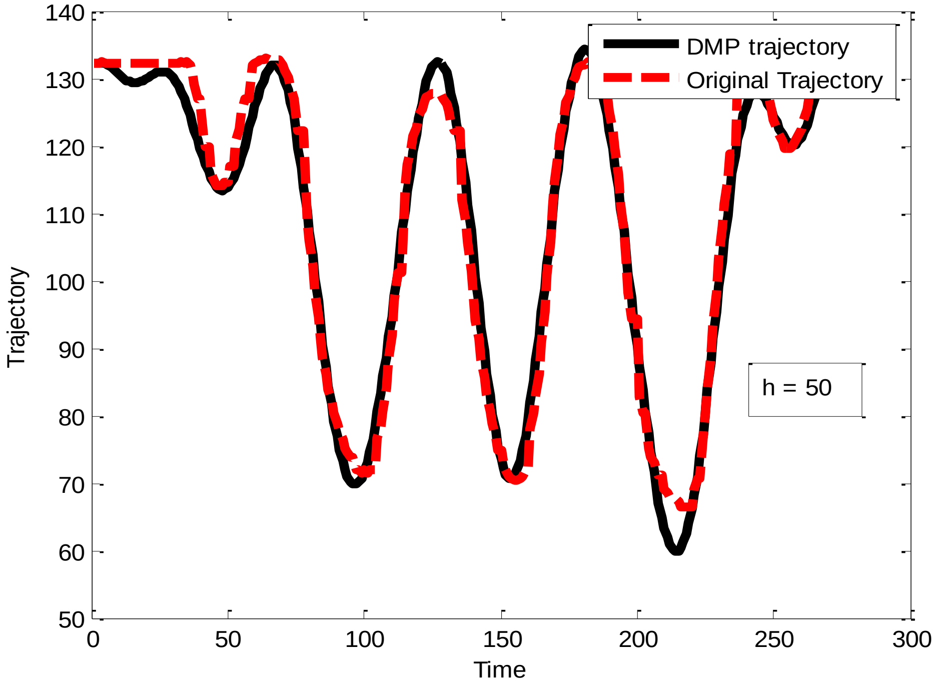


Figure 5.21 Variation of theta angle through time for t width of Gaussian as 50

Changes in the weight of Gaussian also changes the DMP trajectory. The effect of this variable is also tested and results are presented in Figure 5.22 and Figure 5.23 for the multiplication of weight by 1.5 and 2. It can be seen that as w increases, the scaling of DMP is done in amplitude.

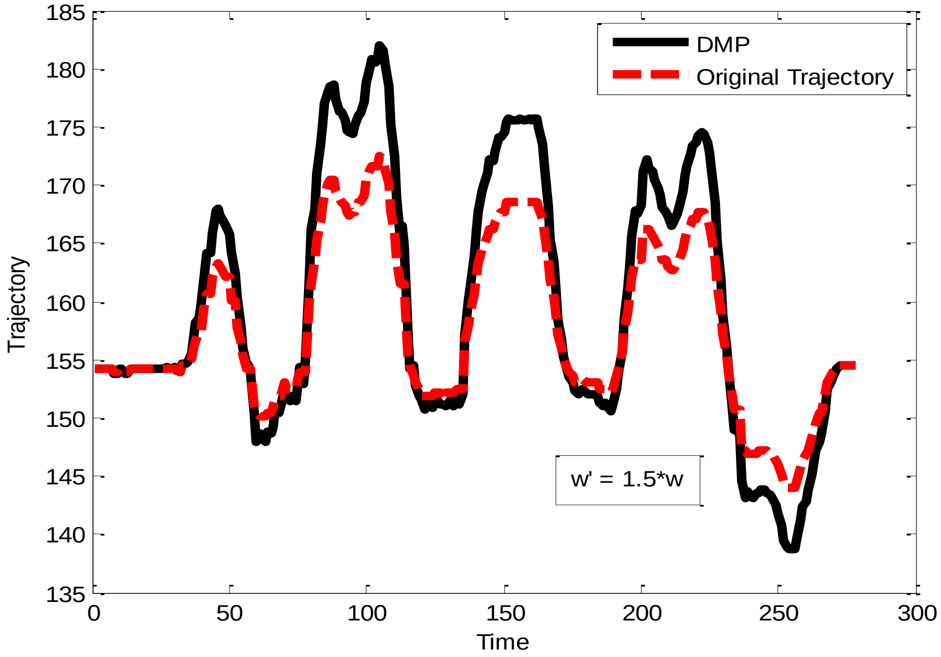


Figure 5.22 Variation of gamma angle through time for w’=1.5*w

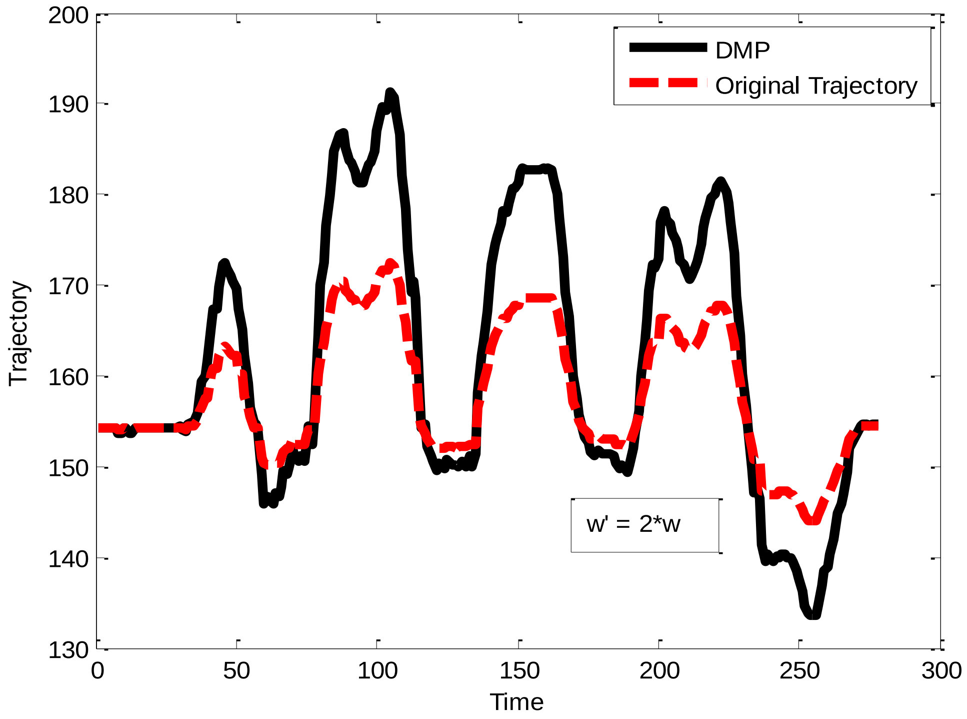


Figure 5.23 Variation of gamma angle through time for w’=2*w

Chapter 6

Conclusion and Future Work

6.1. Conclusion

In this thesis we addressed some problems regarding Human – Robot gesture Imitation and Analysis on Robots. The main aim was to develop better human like gestures.

Here, we discussed and studied a novel gesture encoding technique called Dynamic Movement Primitive and also worked on Dynamic Time Warping. The motivation of using these algorithms is to make a more human- like robot movement system, so that human- robot interactions can become easier.

6.1.1 Imitation Analysis using Dynamic Movement Primitive

In this portion of the thesis, we study trajectories defined by a parameterized set of differential equations called DMP. With the manipulation of certain DMP parameters, we can modify the robot joint or Cartesian trajectories. Thus the robot motion can be changed by just changing the time, frequency and amplitude of the DMP.

This algorithm is implemented on the Zeno robot for joint angle trajectories. Experimental results demonstrate the effectiveness of this method, and future applications of the robot as an Adaptive teacher can now be contemplated.

For the Adaptive Robotic Teacher Project, three parameter control of DMP resulting in time, amplitude and frequency is studied. The results obtained are as expected from the theory. This progress can be explored further for the project and is discussed as future work.

6.1.2 Gesture Comparison using Dynamic Time Warping

In this portion of thesis, scripted motion of Zeno, used for Autism therapy is studied. Experiments were conducted with healthy volunteers at the Next- Gen Systems

Lab, UTA. These experiments were conducted to assess the reliability of using DTW as a quantitative measure of level of limb impairment. The results helped us in establishing the algorithm of DTW as a valid measure for limb impairments, thus it can help us in early detection and diagnosis of Autism.

It was determined statistically that the DTW algorithm can distinguish between a motion that is done normally and a motion that is impaired by a weight. Weighted DTW values were taken as certain joint angles vary very little for certain motions, thus not providing a significant information. The choice of the joint angle can be determined by the calculating the variance of the joint angles through motion. This would give a better insight as to what joint angles must be chosen for the weighted DTW formula.

6.2. Future Work

Further research in the field of Gesture Imitation and analysis would include implementing gesture adaptation and learning on a real robot. Beneficial gestures can be tested on a control environment with Dynamic Movement Primitive Algorithm. This research could be applied on Zeno robots framework. We can aim to teach Autistic kids beneficiary gestures adaptively by manipulating DMP parameters.

The positive results of the manipulation of DMP parameters can be utilized in adaptive robotic teaching of Autistic kids. An adaptive control structure is to be implemented to compare and analyze the robotic DMP gestures and the kids` gesture.

Here, a trainer would chose a gesture, teach the robot using DMP system and the motion will be formed using the DMP architecture. The time, duration, speed and amplitude can be changed in order to adaptively teach the kid.

For Gesture Comparison using Dynamic Time Warping, as the initial results are satisfactory, we still have certain joint angles motions whose results were not significantly different. In order to make this approach more efficient, variance calculation can be incorporated into the work before calculating the weighted DTW value for the purpose of selection and elimination of joint angles in the weighted DTW formula.

Appendix A

Hypothesis Testing

Hypothesis testing is a method of statistical inference to a problem. A result is said to be statistically significant if it would have occurred unlikely by an error [66]. In order to infer to any result, two hypotheses are generated. One is the null hypothesis which states that there is no relation between the text samples. The other hypothesis is called the alternative hypothesis which states the opposite of null hypothesis, or rather what one wants to prove.

The testing process for any problem is as follows.

- Initial premise is unknown.
- State the appropriate null and alternate hypothesis.
- Decide on an appropriate test.
- Calculate the difference in the mean from the hypothesized value over the standard error.
- If the difference lies in the rejection region, then the null hypothesis would be rejected and it is more likely that the alternate hypothesis is true.

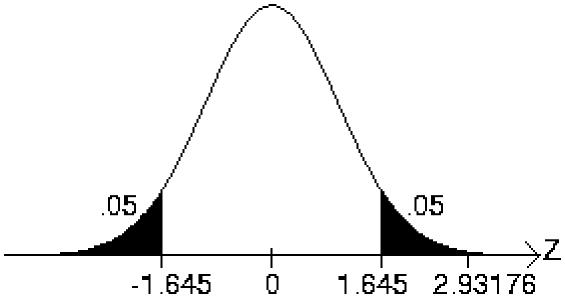


Figure A.1 Hypothesis testing Bell curve [67]

For a null hypothesis, the mean of the sampling distribution and the mean of the population distribution would be the same. Thus the white portion of the Figure A.1 Hypothesis testing Bell curve would correspond to the null hypothesis. The curve denotes a z- score which is given by the equation (13). For a null hypothesis, as the mean is same, the z- score would be zero. The Figure A.1 shows that only beyond a certain standard distribution does the mean of the sample distribution not match the population. The point beyond which the null hypothesis does not hold valid is called the rejection region. This is shown in black in the Figure A.1. Rejection region value depicts the fact that how far would we go to prove the null hypothesis. The value at the rejection region is decided by the sample size of the population and the level of significance that we would want to set. Generally the most used level is that of 95% with 5% in the rejection region. Thus the value of the level of significance would be 0.05. The z- score would be found by either calculating the t- test or the p- value. For a population size over 30, t- test is done and thus, t- table is followed.

A t- table is given below.

Notes:

I. The shaded column headings at the top of the table:

a. “Outside” is used with two-sided t-tests (alternative hypotheses containing the symbol≠)

b. “Within” is used with t confidence intervals

c. “>” is used with right-sided alternative hypothesis (those containing the symbol >)

d. For left sided rejection regions (those containing the “<” symbol), use the > row but change the sign to negative.

II. The rows correspond to the degrees of freedom of the t-test which changes depending on the statistical procedure.

t-table						Direction
Deg. Of Freedom	0.20	0.10	0.05	0.02	0.01	outside
	0.80	0.90	0.95	0.98	0.99	Within
	0.10	0.05	0.025	0.01	0.005	>
1	3.0777	6.3138	12.7062	31.8205	63.6567	
2	1.8856	2.9200	4.3027	6.9646	9.9248	
3	1.6377	2.3534	3.1824	4.5407	5.8409	
4	1.5332	2.1318	2.7764	3.7469	4.6041	
5	1.4759	2.0150	2.5706	3.3649	4.0321	
6	1.4398	1.9432	2.4469	3.1427	3.7074	
7	1.4149	1.8946	2.3646	2.9980	3.4995	
8	1.3968	1.8595	2.3060	2.8965	3.3554	
9	1.3830	1.8331	2.2622	2.8214	3.2498	
10	1.3722	1.8125	2.2281	2.7638	3.1693	
11	1.3634	1.7959	2.2010	2.7181	3.1058	
12	1.3562	1.7823	2.1788	2.6810	3.0545	
13	1.3502	1.7709	2.1604	2.6503	3.0123	
14	1.3450	1.7613	2.1448	2.6245	2.9768	
15	1.3406	1.7531	2.1314	2.6025	2.9467	
16	1.3368	1.7459	2.1199	2.5835	2.9208	
17	1.3334	1.7396	2.1098	2.5669	2.8982	
18	1.3304	1.7341	2.1009	2.5524	2.8784	
19	1.3277	1.7291	2.0930	2.5395	2.8609	
20	1.3253	1.7247	2.0860	2.5280	2.8453	
21	1.3232	1.7207	2.0796	2.5176	2.8314	
22	1.3212	1.7171	2.0739	2.5083	2.8188	
23	1.3195	1.7139	2.0687	2.4999	2.8073	
24	1.3178	1.7109	2.0639	2.4922	2.7969	
25	1.3163	1.7081	2.0595	2.4851	2.7874	
26	1.3150	1.7056	2.0555	2.4786	2.7787	
27	1.3137	1.7033	2.0518	2.4727	2.7707	
28	1.3125	1.7011	2.0484	2.4671	2.7633	
29	1.3114	1.6991	2.0452	2.4620	2.7564	
30	1.3104	1.6973	2.0423	2.4573	2.7500	
31	1.3095	1.6955	2.0395	2.4528	2.7440	
32	1.3086	1.6939	2.0369	2.4487	2.7385	
33	1.3077	1.6924	2.0345	2.4448	2.7333	
34	1.3070	1.6909	2.0322	2.4411	2.7284	
35	1.3062	1.6896	2.0301	2.4377	2.7238	

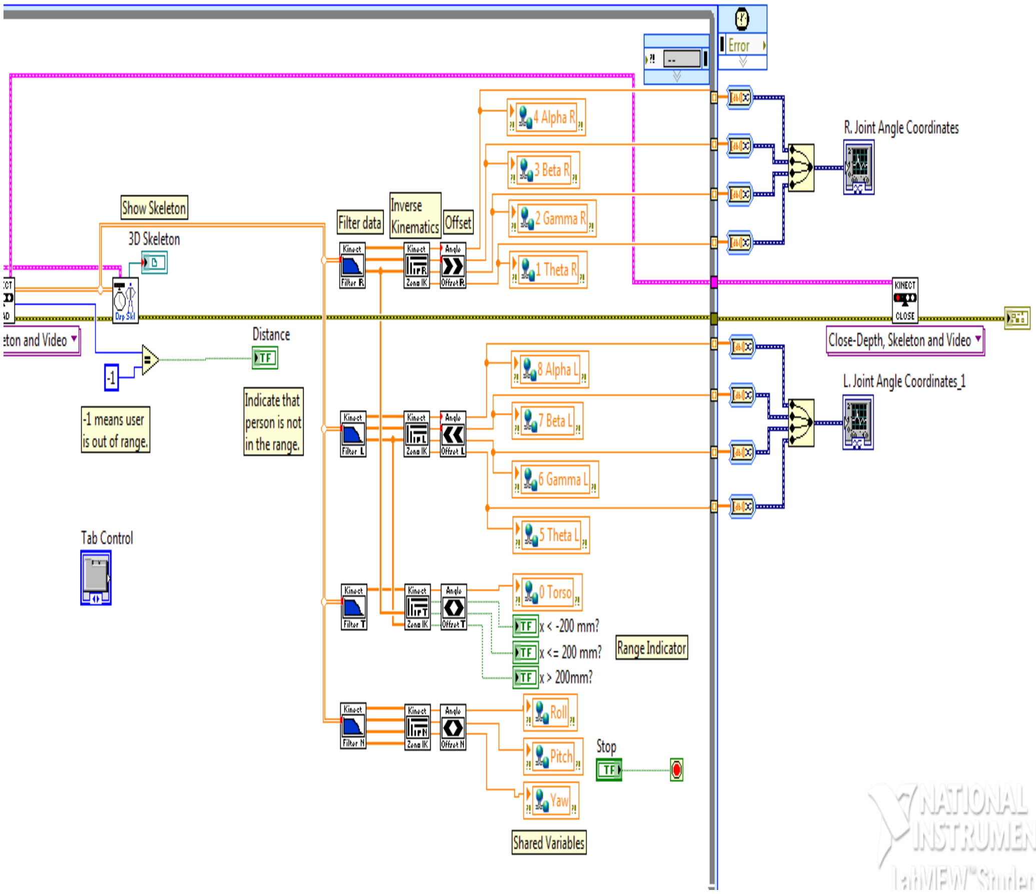
t-table						Direction
Deg. Of Freedom	0.20	0.10	0.05	0.02	0.01	outside
	0.80	0.90	0.95	0.98	0.99	Within
	0.10	0.05	0.025	0.01	0.005	>
36	1.3055	1.6883	2.0281	2.4345	2.7195	
37	1.3049	1.6871	2.0262	2.4314	2.7154	
38	1.3042	1.6860	2.0244	2.4286	2.7116	
39	1.3036	1.6849	2.0227	2.4258	2.7079	
40	1.3031	1.6839	2.0211	2.4233	2.7045	
41	1.3025	1.6829	2.0195	2.4208	2.7012	
42	1.3020	1.6820	2.0181	2.4185	2.6981	
43	1.3016	1.6811	2.0167	2.4163	2.6951	
44	1.3011	1.6802	2.0154	2.4141	2.6923	
45	1.3006	1.6794	2.0141	2.4121	2.6896	
46	1.3002	1.6787	2.0129	2.4102	2.6870	
47	1.2998	1.6779	2.0117	2.4083	2.6846	
48	1.2994	1.6772	2.0106	2.4066	2.6822	
49	1.2991	1.6766	2.0096	2.4049	2.6800	
50	1.2987	1.6759	2.0086	2.4033	2.6778	
51	1.2984	1.6753	2.0076	2.4017	2.6757	
52	1.2980	1.6747	2.0066	2.4002	2.6737	
53	1.2977	1.6741	2.0057	2.3988	2.6718	
54	1.2974	1.6736	2.0049	2.3974	2.6700	
55	1.2971	1.6730	2.0040	2.3961	2.6682	
56	1.2969	1.6725	2.0032	2.3948	2.6665	
57	1.2966	1.6720	2.0025	2.3936	2.6649	
58	1.2963	1.6716	2.0017	2.3924	2.6633	
59	1.2961	1.6711	2.0010	2.3912	2.6618	
60	1.2958	1.6706	2.0003	2.3901	2.6603	
61	1.2956	1.6702	1.9996	2.3890	2.6589	
62	1.2954	1.6698	1.9990	2.3880	2.6575	
63	1.2951	1.6694	1.9983	2.3870	2.6561	
64	1.2949	1.6690	1.9977	2.3860	2.6549	
65	1.2947	1.6686	1.9971	2.3851	2.6536	
66	1.2945	1.6683	1.9966	2.3842	2.6524	
67	1.2943	1.6679	1.9960	2.3833	2.6512	
68	1.2941	1.6676	1.9955	2.3824	2.6501	
69	1.2939	1.6672	1.9949	2.3816	2.6490	
70	1.2938	1.6669	1.9944	2.3808	2.6479	

t-table						Direction
Deg. Of Freedom	0.20	0.10	0.05	0.02	0.01	outside
	0.80	0.90	0.95	0.98	0.99	Within
	0.10	0.05	0.025	0.01	0.005	>
71	1.2936	1.6666	1.9939	2.3800	2.6469	
72	1.2934	1.6663	1.9935	2.3793	2.6459	
73	1.2933	1.6660	1.9930	2.3785	2.6449	
74	1.2931	1.6657	1.9925	2.3778	2.6439	
75	1.2929	1.6654	1.9921	2.3771	2.6430	
76	1.2928	1.6652	1.9917	2.3764	2.6421	
77	1.2926	1.6649	1.9913	2.3758	2.6412	
78	1.2925	1.6646	1.9908	2.3751	2.6403	
79	1.2924	1.6644	1.9905	2.3745	2.6395	
80	1.2922	1.6641	1.9901	2.3739	2.6387	
81	1.2921	1.6639	1.9897	2.3733	2.6379	
82	1.2920	1.6636	1.9893	2.3727	2.6371	
83	1.2918	1.6634	1.9890	2.3721	2.6364	
84	1.2917	1.6632	1.9886	2.3716	2.6356	
85	1.2916	1.6630	1.9883	2.3710	2.6349	
86	1.2915	1.6628	1.9879	2.3705	2.6342	
87	1.2914	1.6626	1.9876	2.3700	2.6335	
88	1.2912	1.6624	1.9873	2.3695	2.6329	
89	1.2911	1.6622	1.9870	2.3690	2.6322	
90	1.2910	1.6620	1.9867	2.3685	2.6316	
91	1.2909	1.6618	1.9864	2.3680	2.6309	
92	1.2908	1.6616	1.9861	2.3676	2.6303	
93	1.2907	1.6614	1.9858	2.3671	2.6297	
94	1.2906	1.6612	1.9855	2.3667	2.6291	
95	1.2905	1.6611	1.9853	2.3662	2.6286	
96	1.2904	1.6609	1.9850	2.3658	2.6280	
97	1.2903	1.6607	1.9847	2.3654	2.6275	
98	1.2902	1.6606	1.9845	2.3650	2.6269	
99	1.2902	1.6604	1.9842	2.3646	2.6264	
100	1.2901	1.6602	1.9840	2.3642	2.6259	
110	1.2893	1.6588	1.9818	2.3607	2.6213	
120	1.2886	1.6577	1.9799	2.3578	2.6174	
infinity	1.2816	1.6449	1.9600	2.3263	2.5758	

Appendix B

Program

Figure B.**1** is the screen shot of the Motor Control VI



shows the reading of the sequence of motion from a pre-recorded set.

Figure B.**1** Kinematics Mapping to Joint Angles

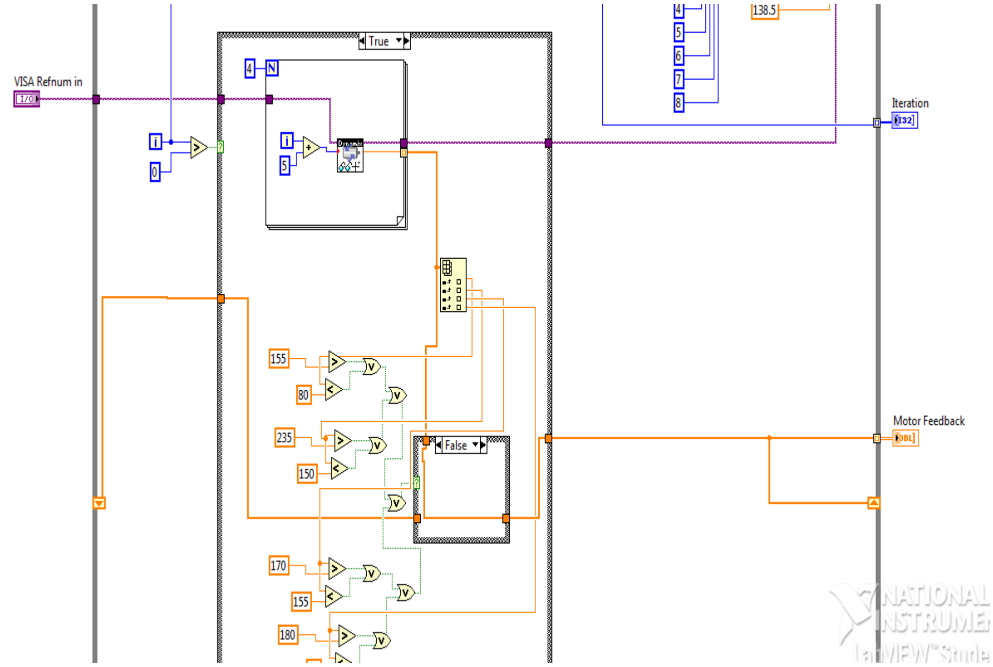
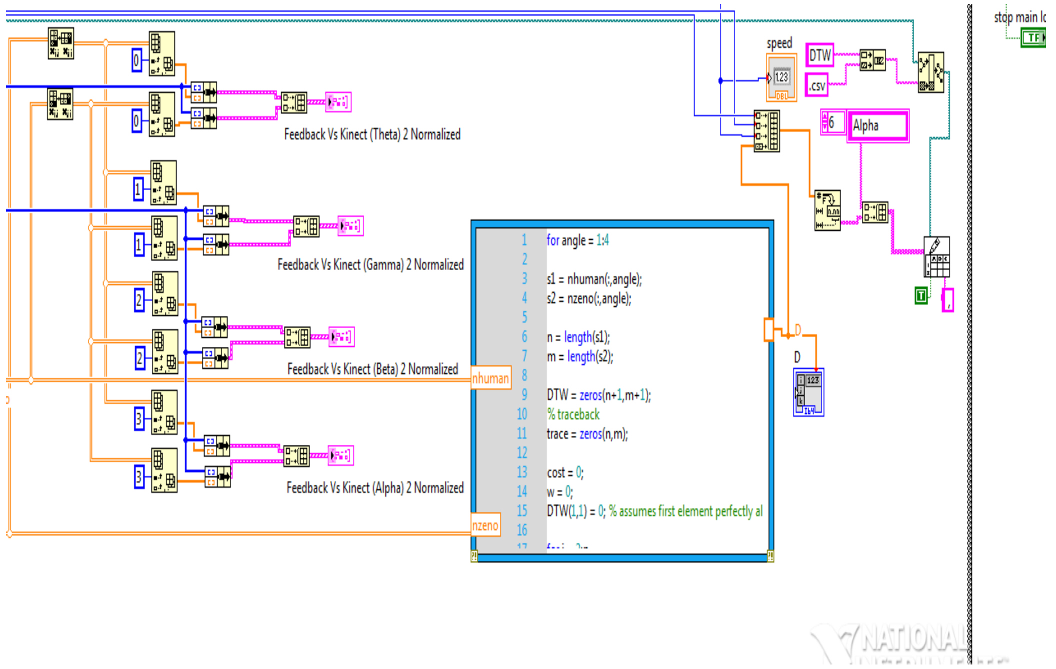


Figure B.2 Code snippet of reading the motor feedback value

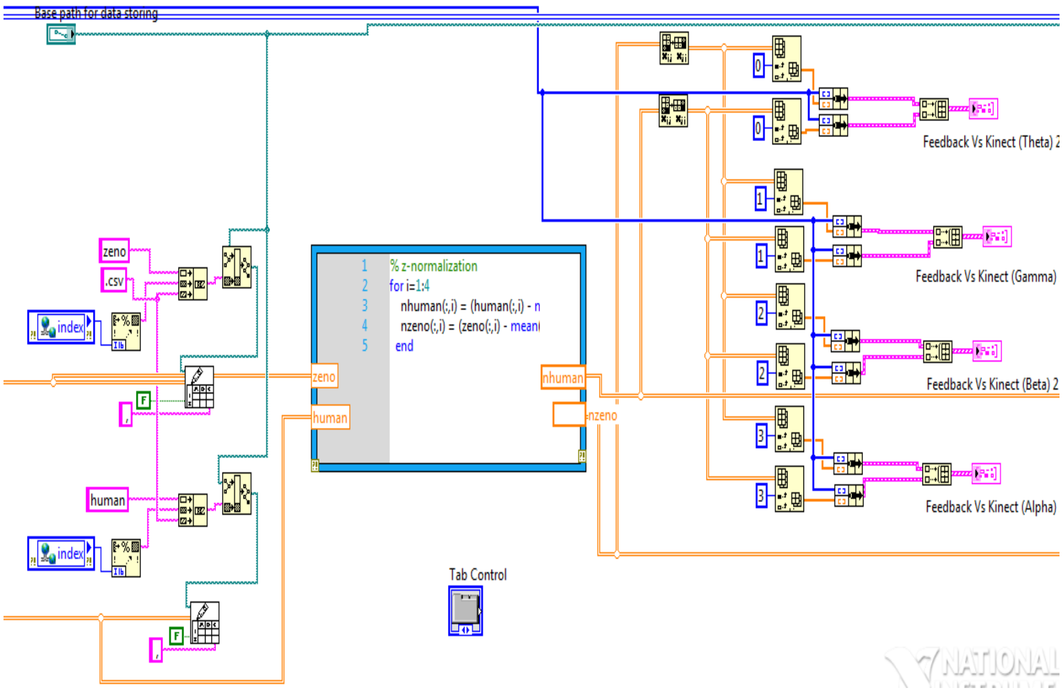
This VI also includes a portion for the comparison of Zeno`s joint angles and humans` joint angles. The code snippet can be found in the Figure B.2.



```
for angle = 1:4

s1 = nhuman(:,angle);
s2 = nzeno(:,angle);

n = length(s1);
m = length(s2);

DTW = zeros(n+1,m+1);
% traceback
trace = zeros(n,m);

cost = 0;
w = 0;
DTW(1,1) = 0; % assumes first element perfectly al
```

Feedback Vs Kinect (Theta) 2 Normalized
Feedback Vs Kinect (Gamma) 2 Normalized
Feedback Vs Kinect (Beta) 2 Normalized
Feedback Vs Kinect (Alpha) 2 Normalized
nhuman
nzeno
speed
DTW
.csv
Alpha
D
D
stop main lc

Figure B.3 DTW LabVIEW code



```
% z-normalization
for i=1:4
   nhuman(:,i) = (human(:,i) - n
   nzeno(:,i) = (zeno(:,i) - mean(
  end
```

Base path for data storing
zeno
.csv
index
human
index
zeno
human
nhuman
nzeno
Tab Control
Feedback Vs Kinect (Theta) 2
Feedback Vs Kinect (Gamma)
Feedback Vs Kinect (Beta) 2
Feedback Vs Kinect (Alpha)

Figure B.4 Normalization code

DMP Matlab Code

```
% Author: Isura Ranatunga and Namrata Balakrishnan
%  email address: isura@ieee.org
% Website: isura.me
% Created: 02/13/2015
% Modified: 02/13/2015 Isura
% Modified: 02/17/2015 Namrata
%

%------------- BEGIN CODE --------------
clc
clear all
clf

% Assumptions
% 1 - vectors are row vectors by default

% Phase variable
% This produces a variable which is a straight line with negative
slope
s = 0:0.001:1;
s = ones(length(s),1)' - s;

figure(1)
% Plot the phase variable
plot(s,'r', 'LineWidth',2);
xlabel('Time(s)');
ylabel('s');
title('Variation of Phase Variable through time');


%%
% Target Modulation function.
load targetData

% Gaussian Basis Function

% center of the gaussian kernels
c = 0:0.01:1;
% width control (Standard deviation) for smaller width, height of
the
% gaussian must be large. So large value of h for a thinner
gaussian.
h = 1000*ones(1,length(c));
% weight
w = randn(1,length(c));
```

```
% Calculating psi for linear phase variable
for k = 1:length(c)
   psi(k,:) = exp(-h(k)*ones(length(s),1)'.* (s -
c(k)*ones(length(s),1)').^2);
end


figure(2)

% Plot the guassian functions
plot(s,psi);
xlabel('Phase Variable s');
ylabel('Psi(s)');
title('Gaussian Curves over the linear phase variable');

%%
% Calculating the modulation function with linear phase variable

for k = 1:length(s)
   f(k) = sum(w*psi(:,k)*s(k))/sum(psi(:,k));
end

figure(3)

% Plot the modulation functions
plot(s,f);
xlabel('Phase Variable s');
ylabel('Modulation Function');
title('Variation of Modulation Function through the phase
variable');

%%
weight = repmat (w,length(s),1);
multiply = weight'.* psi;

figure(4)
% Plot of the weighted gaussians
plot(s, multiply);
xlabel('Phase Variable s');
ylabel('Psi(s)');
title('Weighted Gaussian Curves over the linear phase variable');
hold on

handlevector(1)=plot(s,f, '--r','LineWidth',4);
hold off
legend(handlevector(1),'Modulation Function');

%% Calculating the weights using closed form solution.
```

```
psis = psi.* repmat (s,length(c),1);
LHS = ft.* sum(psi);
w = LHS * pinv(psis);


%%
% Calculating the new modulation function for linear phase
variable
for k = 1:length(s)
   fn(k) = sum(w*psi(:,k)*s(k))/sum(psi(:,k));
end

figure(5)
% Plot the modulation functions
plot(s, fn, 'k', 'LineWidth',4)
xlabel('Phase Variable s');
ylabel('Modulation Function');
title('Target modulation function v/s Generated modulation
function');
hold on
plot(s, ft, '--r', 'LineWidth',4)
hold off
legend('Generated Modulation Function','Target Modulation
Function');
%%
% Learning from the trained DMP

clear all
clc

% Phase variable
delT = 0.001;
s = 0:delT:1;
s = ones(length(s),1)' - s;

% time = 0:delT:1;
% [T,S] = ode45(@canonicalEquation,time,1);
% s = S';

% Gaussian basis function

% center of the gaussian kernels
c = 0:0.02:1                ;
% width control (Standard deviation) for smaller width, height of
the
% gaussian must be large. So large value of h for a thinner
gaussian.
h = 1000*ones(1,length(c))  ;
```

```
% From repeated testing these data were found. For higher K,
error was high
% and for lower value of K change in goal position doesnt produce
any
% effect
% Lower value of D produced better result.
% K = 6.25;
% D = 0.001 ;

K = 1000000;
D = 2500;

tau = 1 ;
load targetData

% Lets assume the desired trajectory is ft from before
x   = ft                ; % traj to be learnt
xd  = [ 0 diff(x) ]/delT;
xdd = [ 0 diff(xd)]/delT;

g = x(end)  ;
x0= x(1)    ;

w = randn(1,length(c))     ; % weight

% Calculating psi for linear phase variable
for k = 1:length(c)
   psi(k,:) = exp(-h(k)*ones(length(s),1)'.* (s -
c(k)*ones(length(s),1)').^2);
end


% New formation target modulation function from transformation
equations
f_target = ( tau*xdd + D*xd ) / K - ( g - x ) + ( g - x0 )*s;

% Learning the target trajectory by weight learning
psis = psi.* repmat (s,length(c),1);
LHS = f_target.* sum(psi);
w = LHS * pinv(psis);

% Learnt modulation function
for k = 1:length(s)
   f_learnt(k) = sum(w*psi(:,k)*s(k))/sum(psi(:,k));
end

figure(6)
```

```
plot( f_learnt, 'k', 'LineWidth',4)
hold on
xlabel('Phase Variable s');
ylabel('Modulation Function');
title('Variation of Modulation function through phase variable');
plot( f_target, '--r', 'LineWidth',4)
hold off
legend('Learnt Modulation Function','Target Modulation
Function');

%%
%g = 1;
g=x(end);
%g = input('Enter the goal point,g =');

tau =2;

% Solving the transformation systems equation
xo(1)  = x0;
xod(1) = xd(1) ;

duration = length(x);
adap = round(1/5*duration);
for k = 2:length(x)

%     if(k >= adap)
%         g = -0.4;
%     else
%         g = x(end);
%     end
    xodd(k) = ( K*(g - xo(k-1)) - D*xod(k-1) - K*(g - x0)*s(k) +
K*f_learnt(k) );
    xod(k)  =  xod(k-1)  + xodd(k)*delT  / tau ;
    xo(k)   =    xo(k-1)  +   xod(k)*delT /tau        ;
end

figure(7)
plot(xo, 'k', 'LineWidth',4)
hold on
xlabel('Phase Variable s');
ylabel('Trajectory');
title('Variation of Trajectory through phase variable');
plot( x, '--r', 'LineWidth',4)
hold off
legend('DMP trajectory','Original Trajectory ');

disp('xo(k)=');
disp(xo(k));
disp('x(k)=');
disp(x(k));
.
```

References

[1] D. Feil-Seifer and M. J. Matari{\'c}, "Human RobotHuman--robot interaction (HRI) InteractionInteraction human robot," in *Encyclopedia of Complexity and Systems Science*, Springer, 2009, pp. 4643-4659.

[2] E. Short, D. Feil-Seifer and M. Matari{\'c}, "A comparison of machine learning techniques for modeling human-robot interaction with children with autism," in *Proceedings of the 6th international conference on Human-robot interaction*, 2011.

[3] S. Valibeik and G.-Z. Yang, "Segmentation and tracking for vision based human robot interaction," in *Web Intelligence and Intelligent Agent Technology, 2008. WI-IAT'08. IEEE/WIC/ACM International Conference on*, 2008.

[4] Y. Lei, W. Hongpeng, T. Dianxiong and W. Jue, "A real-time hand gesture recognition algorithm for an embedded system," in *Mechatronics and Automation (ICMA), 2014 IEEE International Conference on*, 2014.

[5] I. Carrera, A. Campos and H. A. Moreno, "Concept Design and Analysis of a Novel Robot for Rehabilitation and Assistance," in *Robotics Symposium and Latin American Robotics Symposium (SBR-LARS), 2012 Brazilian*, 2012.

[6] R. Riener, "Control of robots for rehabilitation," in *Computer as a Tool, 2005. EUROCON 2005. The International Conference on*, 2005.

[7] K. Low, "Robot-assisted gait rehabilitation: From exoskeletons to gait systems," in *Defense Science Research Conference and Expo (DSR), 2011*, 2011.

[8] N. Giullian, D. Ricks, A. Atherton, M. Colton, M. Goodrich and B. Brinton, "Detailed requirements for robots in autism therapy," in *Systems Man and Cybernetics (SMC), 2010 IEEE International Conference on*, 2010.

[9] J. Hwang, M.-G. Kim and K. Cho, "Developing interaction scenarios of robot-mediated imitation training for children with Autism Spectrum Disorders," in *RO-MAN, 2013 IEEE*, 2013.

[10] Z. Zheng, L. Zhang, E. Bekele, A. Swanson, J. Crittendon, Z. Warren and N. Sarkar, "Impact of robot-mediated interaction system on joint attention skills for children with autism," in *Rehabilitation Robotics (ICORR), 2013 IEEE International Conference on*, 2013.

[11] I. Noens, V. Berckelaer-Onnes, R. Verpoorten, G. Van Duijn and others, "The ComFor: an instrument for the indication of augmentative communication in people with autism and intellectual disability," *Journal of Intellectual Disability Research,* vol. 50, no. 9, pp. 621-632, 2006.

[12] F. R. Volkmar, R. Paul, A. Klin and D. J. Cohen, Handbook of Autism and Pervasive Developmental Disorders, Diagnosis, Development, Neurobiology, and Behavior, vol. 1, John Wiley \& Sons, 2005.

[13] S. J. Rogers, "What are infant siblings teaching us about autism in infancy?," *Autism Research,* vol. 2, no. 3, pp. 125-137, 2009.

[14] C. P. Johnson, S. M. Myers and others, "Identification and evaluation of children with autism spectrum disorders," *Pediatrics,* vol. 120, no. 5, pp. 1183-1215, 2007.

[15] "Diagnostic Criteria," 02 2015. [Online]. Available: http://www.cdc.gov/ncbddd/autism/hcp-dsm.html.

[16] "Signs and Symptoms," 02 2015. [Online]. Available: http://support.robotis.com/en/product/dynamixel/rx_series/rx-28.htm.

[17] B. Scassellati, H. Admoni and M. Mataric, "Robots for use in autism research," *Annual Review of Biomedical Engineering,* vol. 14, pp. 275-294, 2012.

[18] D. H. Geschwind, "Advances in autism," *Annual review of medicine,* vol. 60, p. 367, 2009.

[19] K. Glatz, *Adaptive Learning from Demonstration using Dynamic Movement Primitives,* August, 2012.

[20] A. Gupta, S. Satkin, A. A. Efros and M. Hebert, "From 3d scene geometry to human workspace," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, 2011.

[21] P. Donelan, Kinematic singularities of robot manipulators, INTECH Open Access Publisher, 2010.

[22] D. Hanson, S. Baurmann, T. Riccio, R. Margolin, T. Dockins, M. Tavares and K. Carpenter, "Zeno: A cognitive character," in *AI Magazine, and special Proc. of AAAI National Conference, Chicago*, 2009.

[23] "Programmed to Treat Autism," 2011. [Online]. Available: http://texasmrc.org/programmed-to-treat-autism/.

[24] M. N. Nicolescu and M. J. Mataric, "Experience-based representation construction: learning from human and robot teachers," in *Intelligent Robots and Systems, 2001. Proceedings. 2001 IEEE/RSJ International Conference on*, 2001.

[25] N. S. Pollard, J. K. Hodgins, M. J. Riley and C. G. Atkeson, "Adapting human motion for the control of a humanoid robot," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2002.

[26] T. Asfour and R. Dillmann, "Human-like motion of a humanoid robot arm based on a closed-form solution of the inverse kinematics problem," in *Intelligent Robots and Systems, 2003.(IROS 2003). Proceedings. 2003 IEEE/RSJ International Conference on*, 2003.

[27] S. Kim, C. H. Kim and J. H. Park, "Human-like arm motion generation for humanoid robots using motion capture database," in *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, 2006.

[28] R. M. Pierce and K. J. Kuchenbecker, "A data-driven method for determining natural human-robot motion mappings in teleoperation," in *Biomedical Robotics and Biomechatronics (BioRob), 2012 4th IEEE RAS \& EMBS International Conference on*, 2012.

[29] T. Minato and H. Ishiguro, "Generating Natural Interactive Motion in Android Based on Situation-Dependent Motion Variety," 2009.

[30] S. G{\"a}rtner, M. Do, T. Asfour, R. Dillmann, C. Simonidis and W. Seemann, "Generation of human-like motion for humanoid robots based on marker-based motion capture data," in *Robotics (ISR), 2010 41st International Symposium on and 2010 6th German Conference on Robotics (ROBOTIK)*, 2010.

[31] C. G. Atkeson, J. G. Hale, F. E. Pollick, M. Riley, S. Kotosaka, S. Schaul, T. Shibata, G. Tevatia, A. Ude, S. Vijayakumar and others, "Using humanoid

robots to study human behavior," *IEEE Intelligent Systems and their applications,* vol. 15, no. 4, pp. 46-56, 2000.

[32] S. Calinon and A. Billard, "Active teaching in robot programming by demonstration," in *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on*, 2007.

[33] S. Calinon, F. Guenter and A. Billard, "On learning, representing, and generalizing a task in a humanoid robot," *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on,* vol. 37, no. 2, pp. 286-298, 2007.

[34] S. H. Lee, I. H. Suh, S. Calinon and R. Johansson, "Learning basis skills by autonomous segmentation of humanoid motion trajectories," in *Humanoid Robots (Humanoids), 2012 12th IEEE-RAS International Conference on*, 2012.

[35] A. J. Ijspeert, J. Nakanishi and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation, 2002. Proceedings. ICRA'02. IEEE International Conference on*, 2002.

[36] B. Argall, B. Browning and M. Veloso, "Learning by demonstration with critique from a human teacher," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007.

[37] S. Calinon and A. Billard, "Incremental learning of gestures by imitation in a humanoid robot," in *Proceedings of the ACM/IEEE international conference on Human-robot interaction*, 2007.

[38] T. Kulvicius, K. Ning, M. Tamosiunaite and F. Worgotter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications

exemplified on handwriting," *Robotics, IEEE Transactions on,* vol. 28, no. 1, pp. 145-157, 2012.

[39] M. Prada, A. Remazeilles, A. Koene and S. Endo, "Dynamic Movement Primitives for Human-Robot interaction: comparison with human behavioral observation," in *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*, 2013.

[40] P. Pastor, H. Hoffmann, T. Asfour and S. Schaal, "Learning and generalization of motor skills by learning from demonstration," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009.

[41] H. Hoffmann, P. Pastor, D.-H. Park and S. Schaal, "Biologically-inspired dynamical systems for movement generation: automatic real-time goal adaptation and obstacle avoidance," in *Robotics and Automation, 2009. ICRA'09. IEEE International Conference on*, 2009.

[42] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 26, no. 1, pp. 43-49, 1978.

[43] R. Christiansen and C. K. Rushforth, "Detecting and locating key words in continuous speech using linear predictive coding," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 25, no. 5, pp. 361-367, 1977.

[44] F. Ferr{\`e} and P. Clote, "BTW: a web server for Boltzmann time warping of gene expression time series," *Nucleic acids research,* vol. 34, no. suppl 2, pp. W482--W485, 2006.

[45] P. Jangyodsuk, C. Conly and V. Athitsos, "Sign language recognition using dynamic time warping and hand shape distance based on histogram of oriented

gradient features," in *Proceedings of the 7th International Conference on PErvasive Technologies Related to Assistive Environments*, 2014.

[46] H. Wang, A. Stefan, S. Moradi, V. Athitsos, C. Neidle and F. Kamangar, "A system for large vocabulary sign search," in *Trends and Topics in Computer Vision*, Springer, 2012, pp. 342-353.

[47] J. Lijffijt, P. Papapetrou, J. Hollm{\'e}n and V. Athitsos, "Benchmarking dynamic time warping for music retrieval," in *Proceedings of the 3rd international conference on pervasive technologies related to assistive environments*, 2010.

[48] R. Bellman and S. Dreyfus, "Functional approximations and dynamic programming," *Mathematical Tables and Other Aids to Computation,* pp. 247-251, 1959.

[49] S. Dreyfus, "Introduction to the 2010 Edition," 2010. [Online]. Available: http://press.princeton.edu/chapters/i9234.pdf.

[50] T. K. Vintsyuk, "Speech discrimination by dynamic programming," *Cybernetics and Systems Analysis,* vol. 4, no. 1, pp. 52-57, 1968.

[51] L. R. Rabiner and B.-H. Juang, Fundamentals of speech recognition, vol. 14, PTR Prentice Hall Englewood Cliffs, 1993.

[52] S. Schaal, J. Peters, J. Nakanishi and A. Ijspeert, "Learning movement primitives," in *Robotics Research. The Eleventh International Symposium*, 2005.

[53] A. J. Ijspeert, J. Nakanishi and S. Schaal, "Learning rhythmic movements by demonstration using nonlinear oscillators," in *Proceedings of the ieee/rsj int. conference on intelligent robots and systems (iros2002)*, 2002.

[54] "Kinect: Gestures," [Online]. Available: http://support.xbox.com/en-US/browse/xbox-360/kinect/Gesture.

[55] N. A. Torres, N. Clark, I. Ranatunga and D. Popa, "Implementation of interactive arm playback behaviors of social robot Zeno for autism spectrum disorder therapy," in *Proceedings of the 5th International Conference on PErvasive Technologies Related to Assistive Environments*, 2012.

[56] "Dynamixel RX-28 : User`s Manual," 1999. [Online]. Available: http://support.robotis.com/en/product/dynamixel/rx_series/rx-28.htm.

[57] "Documentation," [Online]. Available: http://www.mathworks.com/help/matlab/index.html?refresh=true.

[58] "Object Oriented Programming in MATLAB," [Online]. Available: http://www.mathworks.com/discovery/object-oriented-programming.html.

[59] "LabVIEW System Design Software," [Online]. Available: http://www.ni.com/labview/.

[60] "Application Areas," [Online]. Available: http://www.ni.com/labview/applications/.

[61] "LabVIEW Tools Network," [Online]. Available: http://www.ni.com/labview-tools-network/.

[62] "dmp," 12 2014. [Online]. Available: https://github.com/sniekum/dmp.

[63] F. Itakura, "Minimum prediction residual principle applied to speech recognition," *Acoustics, Speech and Signal Processing, IEEE Transactions on,* vol. 23, no. 1, pp. 67-72, 1975.

[64] I. Ranatunga, M. Beltran, N. A. Torres, N. Bugnariu, R. M. Patterson, C. Garver and D. O. Popa, "Human-robot upper body gesture imitation analysis for autism spectrum disorders," in *Social Robotics*, Springer, 2013, pp. 218-228.

[65] G. Shapira, "The Seven Key Steps of Data Analysis," [Online]. Available: http://www.oracle.com/us/corporate/profit/big-ideas/052313-gshapira-1951392.html.

[66] "Statistical hypothesis testing," [Online]. Available: http://en.wikipedia.org/wiki/Statistical_hypothesis_testing.

[67] "Hypothesis Testing," [Online]. Available: http://www.stat.wmich.edu/s216/htests/htests.html.

[68] F. Mirza, "Dynamixel Lab Manual," 2014.

[69] F. Edno, "Documentation," 12 2014. [Online]. Available: http://wiki.ros.org/.

[70] D. Eiken, "Dr. Eakin's BSTAT 3321/5301 Web Site," [Online]. Available: http://wweb.uta.edu/faculty/eakin/bstat3321.html.

[71] T. Hester, "Introduction to ROS Programming," 03 2013. [Online]. Available: http://www.cs.utexas.edu/~todd/cs378/slides/Week8a.pdf.

[72] L. Y. Ku, "RVIZ: a good reason to implement a vision system in ROS," 11 2012. [Online]. Available: https://computervisionblog.wordpress.com/2012/11/18/rviz-a-good-reason-to-implement-a-vision-system-in-ros/.

[73] E. Guizzo, "Willow Garage Giving Away 11 PR2 Robots Worth Over $4 Million," 05 2010. [Online]. Available:

http://spectrum.ieee.org/automaton/robotics/robotics-software/050410-willow-garage-giving-away-11-pr2-robots-worth-over-4-million.

Biographical Information

Namrata Balakrishnan was born in Bangalore, India. She graduated under the Central Board of Secondary Education from high school in Delhi, India in 2009. She attained her Bachelors of Technology degree in Electronics and Instrumentation Engineering from VIT University Vellore, India in 2013 and her Masters of Science degree in Electrical Engineering from the University of Texas at Arlington, Texas in 2015. Her main research area is in Robotics and control.