ABSTRACT

FUZZY LOGIC FOR TASK SPACE TELEMANIPULATION OF A A FIVE FINGERED ROBOTIC HAND

by

Raaaghavann Srinivasan

This work presents a fuzzy logic based HandyMan (Hand gesture commands for grasping and manipulation) system to recognize the operator's hand gestures during task space telemanipulation. A combination of joint positions may be shared between at least two manipulation gestures. To avoid misinterpretation of gestures by the gesture recognition system, six new manipulation gestures were introduced. The gestures are produced by the user wearing a CyberGlove^{TM}.

This system replaces the previous HandyMan gesture recognition method. The output of the fuzzy system drives the state machines to implement the gestures with the robot hand.

The experimental results show that the proposed method can be used for telemanipulation of 15-DOF robot hand in task space. Manipulation in six degrees of freedom and pistol grip manipulation is achieved with a good repeatability percentage and no burst errors. The number of fingers used by the user does not affect the number of fingers used by the robot hand during manipulation. Hence, the same manipulation gesture can be used for 2 finger mode, 3 finger mode and 5 finger mode manipulation.

FUZZY LOGIC FOR TASK SPACE TELEMANIPULATION OF A FIVE FINGERED ROBOTIC HAND

by

Raaghavann Srinivasan

A Thesis

Submitted to the Faculty of

New Jersey Institute of Technology

in Partial Fulfillment of the Requirements for the degree of

Master of Science in Biomedical Engineering

Department of Biomedical Engineering

January 2013

APPROVAL PAGE

FUZZY LOGIC FOR TASK SPACE TELEMANIPULATION OF A FIVE FINGERED ROBOTIC HAND

Raaghavann Srinivasan

Dr. Richard A. Foulds, Thesis Advisor Date
Associate Professor of Biomedical Engineering, NJIT

Dr. Neal Y. Lii, Thesis Advisor Date
Institute of Robotics and Mechatronics, German Aerospace Center

---

Dr. Sergei V. Adamovich, Committee Member Date
Associate Professor of Biomedical Engineering, NJIT

Dr. Max Roman, Committee Member Date
Assistant Research Professor of Biomedical Engineering, NJIT
Director, MS Program

BIOGRAPHICAL SKETCH

Author: Raaghavann Srinivasan

Degree: Master of Science

Date: January 2013

Undergraduate and Graduate Education:

- Master of Science in Biomedical Engineering,
  New Jersey Institute of Technology, Newark, NJ, USA, 2013

- Bachelor of Engineering in Biomedical Engineering,
  Anna University, Chennai, Tamil Nadu, India, 2010

Major: Biomedical Engineering

Blank Page

ACKNOWLEDGMENT

I would like to thank Dr. Neal Lii for his continuous support as thesis advisor. I would like to extend my thanks to my co-advisor Dr. Richard Foulds for his guidance and motivation during the time of work.

Special thanks to Dr. Sergei V. Adamovich and Dr. Max Roman for serving as committee members for this thesis.

I am grateful to Deustches Zentrum fur Luft und Raumfahrt (German Aerospace Center) for providing me an opportunity to do my thesis in their Institute of Robotics and Mechatronics.

I finally offer my gratitude to my family members for their support and trust.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF FIGURES

(Continued)

Figure **Page**

CHAPTER 1

INTRODUCTION

1.1 Objective

The aim of the thesis is to telemanipulate a 15-DoF (Degrees of Freedom) dexterous robotic hand in 6-DoF and perform pistol grip manipulation using Fuzzy Logic based gesture recognition system in order to reduce the drift error to zero and thereby create a better interface between the user and the end effector. The Sband link is the only real time data link to the International Space Station. This data link is shared between robot hand information, robot arm information, information for mobile platform etc**.** The baud rate of Sband link is only 256K. Hence, a low information rate full hand manipulation is desirable. In task space manipulation the user's intent is mapped to the robot hand and the robot hand executes the user's actions [1]. The TaskMan (TASK space grasping and MANIPULATION) concept uses HandyMan (Hand gesture commands for grasping and manipulation) for recognition user's hand gestures while commanding the robot hand [1]. Although the refresh cycle of the robot is 1000 frames / sec, the refresh cycle of the cybergloves is 90 frames/ sec. Hence, the size of each command sent to the robot hand is 90 8bit frames/sec, thus using only very low bandwidth for communication. Matlab commands were used to create the HandyMan gesture recognition engine [1]. This resulted in drift errors, which is undesirable for remote operation purposes. The kinematics of robot hand and human hand are different. Since the robot hand resembles the user's hand intent, task space manipulation avoids the problem of dissimilar kinematics between the robot end effector and operator's hand [2].

Teleoperation of a robotic hand is a trained process. The joint angles of the user were obtained using the CyberGlove. The Cyberglove was configured to match the human hand shape and size. The library of gestures was created using the fuzzy inference system. The joint angle measurements from the 22 glove sensors were used to create the rules of the fuzzy system.

The fuzzy logic based gesture recognition engine identifies the number of fingers to be used for grasping and manipulation. Once the system recognizes the number of fingers used by the user to perform teleoperation, it then detects the manipulation type performed by the user and transmits it for execution by the robot hand.

The manipulation types of the robot hand, XR (Rotation in X direction), XT (Translation in X direction), YR (Rotation in Y direction), YT (Translation in Y direction), ZR (Rotation in Z direction), ZT (Translation in Z direction), Pistol Grip manipulation, are based on the output of the fuzzy inference system, which is determined by the user hand gestures. The manipulation in 6-DoF covers all the manipulation type performed by humans in everyday life. The fuzzy logic base gesture recognition engine indentifies the hand gestures of the user based on the rules.

1.2 Background Information

Humans have a good knowledge of manipulation tasks performed in day to day life, most of which are learnt through everyday practice. A telemanipulation system provides a human operator a way to interact with the environment, which is otherwise inaccessible or unsafe. Examples are use of teleoperated robotic systems in surgery and in teleoperated robotic systems in space applications, which considerably reduces the risk to humans. The manipulators used for each task are different from the other. One used for the industrial purpose is different from the one used in space application, which is different from one used for robotic surgery. Though the manipulators differ, the same concept can be used for each manipulator.

Remote operation of a multifingered robot hand requires commands from the Master (user) which are transmitted to the slave (robot end effector). The commands to the robot can be sent using devices such as joystick. A joystick allows user to input scaling values of position and force to the manipulator [3]. Virtual reality provides a new intuitive and user friendly human-robot interface [3]. There are different methods for controlling a unilateral / bilateral robot. The techniques include from passivity control, adaptive control [4, 5, 6, 7, 8, 9, 10, 11, 12]. Robots can be controlled by controlling their position, force or force and position together. Measure of the manipulating ability in positioning and orientation of the robot hand helps in obtaining best hand postures and task planning [13]. Adaptive control of position/force method was proposed by W.H Zhu and S. E Salcudean for performing teleoperation in a flexible environment. But this method does not deal properly with large uncertainities and requires four channels of communication [14]. In recent past robot controllers has taken advantage of neural network. Robot controllers require processing of huge amount of data.

Conventionally, mapping between the user is hand and the robot hand is either done via joint space to get the similar joint space configurations during manipulation or in Cartesian space to get the same fingertip position [15]. In joint space mapping user's joints are mapped to the corresponding joints in the robot hand. This provides the similar finger joint position. In [16] point to point mapping is presented and the fingertip motion of the user is reproduced by three fingered gripper. Rohling and Hollerbach presented optimized fingertip mapping for teleoperation of dexterous robotic hand for reducing the human-robot fingertip position error. But this model requires an accurate human hand model [17].

Neural networks were used to map the fingertips in Cartesian space [24]. The main issue with the Cartesian space mapping is that for almost every single fingertip of the human, there should exist a fingertip position of the robot hand and the users should know their fingertip position and should predict artificial finger movement [24]. During certain tasks there are some positions of the human hand that lie outside the workspace of the robot hand [24].

Ekvall and Kragic have presented evaluation, recognition and modeling of human grasp during transportation sequence in learning-by-demostration framework. But, for certain poses, this method may lock the robot hand [19]. Virtual object mapping allows the user to make natural movements and have the robot perform similar movements. This mapping assumes the virtual sphere is held between the thumb and index finger. Important parameters of the transformed object are scaled. This modified virtual object is then used to compute robotic fingertip location. This method provides better manipulation range than point to point mapping [9]. Forces and movements exerted by the robotic hand on the grasped object are not guaranteed in the virtual sphere solution used in [21]. The robot hand differs in kinematics, dynamics, programming, and method of control. Joint space and Cartesian space mapping may not work as the user intends because of dissimilar kinematics between human and robot hand. Mapping of human and robot synergies in task space is done using a virtual sphere. This mapping aims at moving the reference joint in a synergistic way. The main advantage of this mapping is that it avoids the problem of dissimilar kinematics. The robot hand mimics the human hand gestures [2]. But, this method may exceed the low bandwidth requirements in some teleoperation applications [25]. Designing and providing the tools that form-fit the end effectors provides a good the teleoperation performance.

TaskMan (Task space grasping and Manipulation) concept uses a library of tasks based on gesture commands [1]. HandyMan has a library of intuitive task gesture commands for grasping and manipulation [1]. The gesture recognition engine is created using the Matlab commands [1]. When the hand gesture is recognized it is then transferred to the robot hand [1]. Since the command is sent to a local controller on the slave side, the local controller executes it and commands the robot, thus reducing the bandwidth required for communication and also taking advantage of shared autonomy [6, 30]. Task space telemanipulation has a higher manipulation success rate and ease of operation [1]. Three translational and three rotational movements were realized since it covers most manipulation gestures performed in day to day life. At the same time the command algorithm - based HandyMan gesture commands suffers from drift problem (displacement in the unintended DoF) and hence requires more robust gesture recognition system [1].

Classifying human hand manipulation is still a field of research. The problem still exists in classifying the manipulation types [23]. Fuzzy control of the robots proves to be robust in theoretical analysis and industrial applications [18, 19, 22, 25].

This research proposes a new method for framing a gesture recognition engine using fuzzy logic to perform the task space grasping and manipulation in 6-DOF, thereby reducing the undesirable drift error to zero. Fuzzy logic controller can incorporate formal reasoning on uncertain input information and is more effective in providing a realistic output [17]. Figure 1.1 shows a basic flowchart for hand gesture based telemanipulation. The users joint angles, joint rates are measured using the cybergloves which is then transferred to a system which executes the commands from the CyberGlove and transmits it to the robotic hand and make it to mimic the users' gestures.

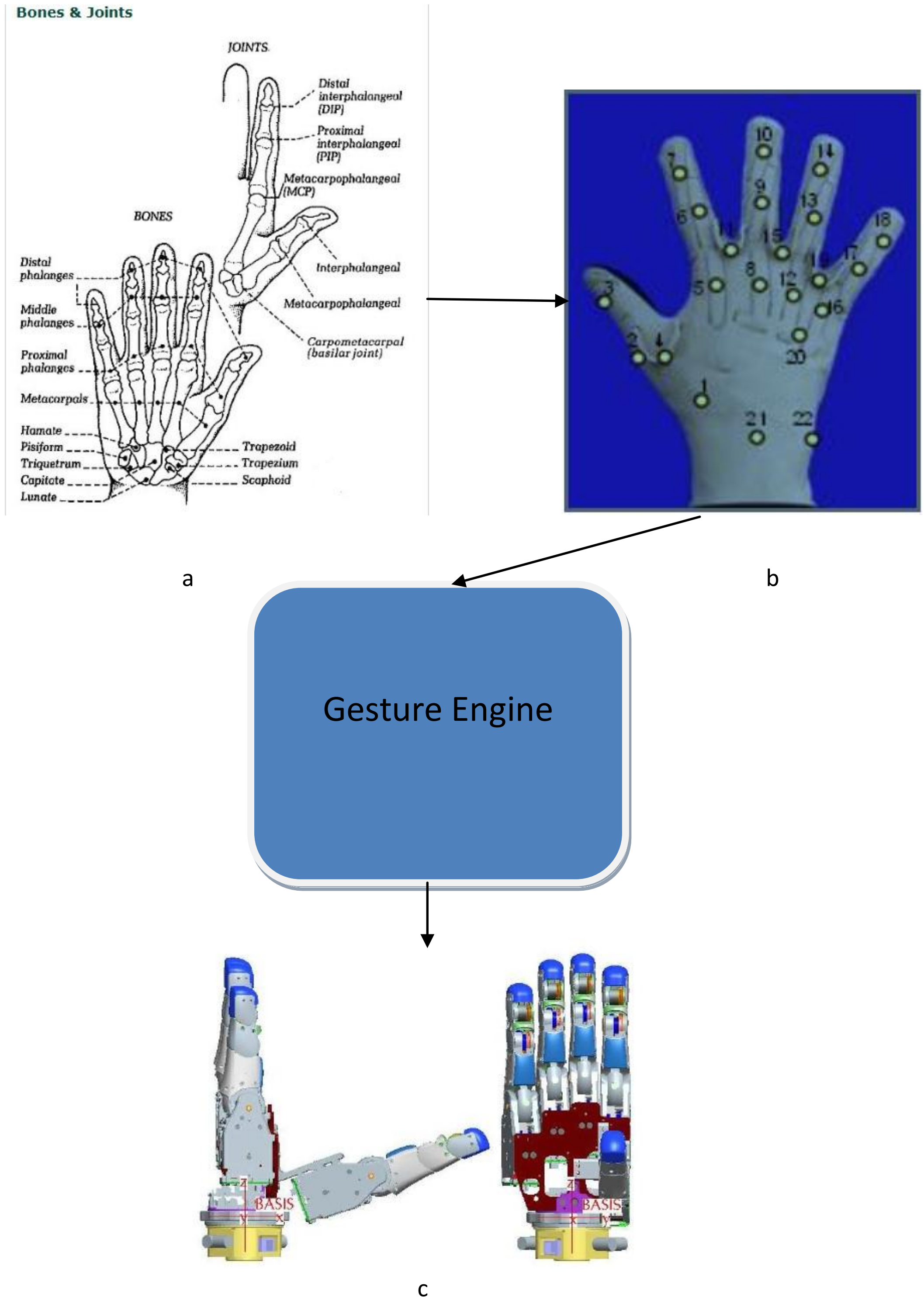

Figure 1.1: Telemanipulation Flowchart

Image (a): Source: [27]

Image (b): Source: [26]

Image (c): Source [1]

CHAPTER 2

SYSTEM ARCHITECTURE

2.1 TaskMan [1]

To overcome the problems faced by joint space mapping, Cartesian space mapping and object space mapping, task space mapping is employed to perform the grasping and manipulation actions. The TaskMan concept uses a library of tasks based on gesture commands [1, 6]. Since the teleoperation in task space is not specific to any end effector type, it can be applied to different end effectors with different kinematics [2]. Two dissimilar TaskMan state machines are employed on the master side and the slave side for communication between HMI (Human machine interface) and the end effector [1]. The TaskMan state machine on the master side and the end effector side (slave side) communicate using an 8bit UDP (User Datagram Protocol) channel [1]. The gesture recognition engine delivers the gesture type, which drives the Master side state machine. The transitions in the master side state machine drive the transitions in the slave side state machine. Figure 2.1 shows a detailed view of slave side TaskMan state machine.

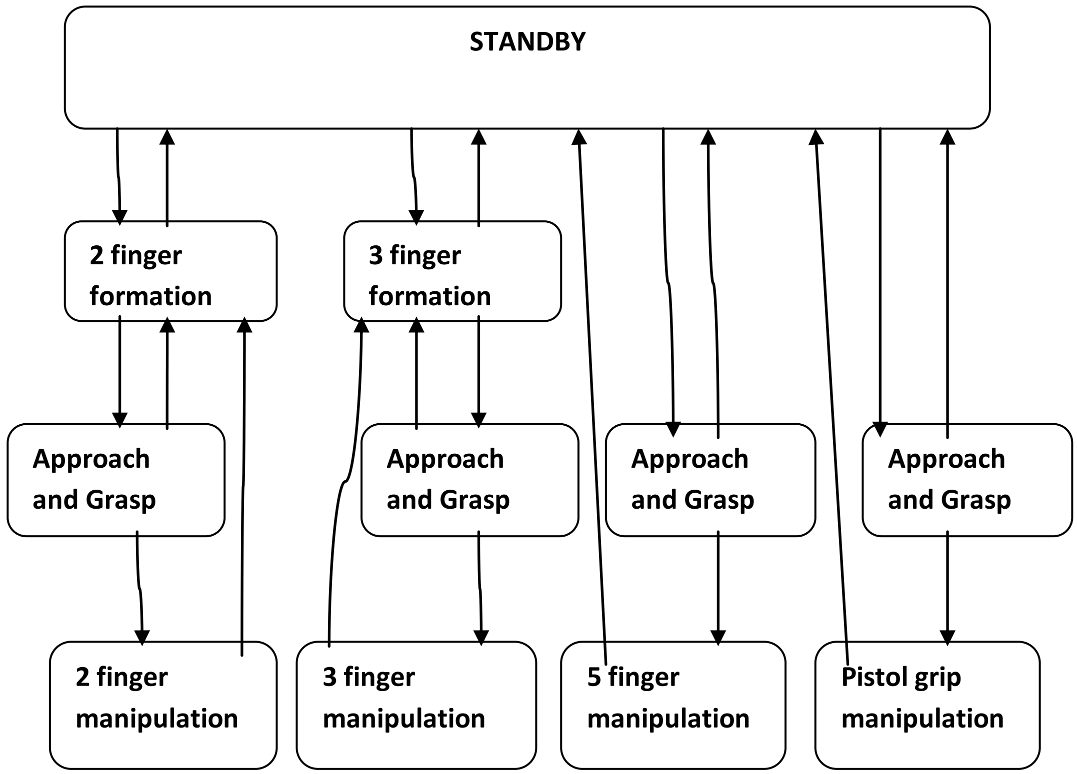

Figure 2.1 Master side state machine

Source: [1]

The Task space teleoperation is based on two dissimilar state machine on the master side and slave side [1]. The dissimilarity reflects the dissimilar dynamics of human and robot hand [1]. Only relevant task states were implemented in the state machines. The formation task state is implemented only in the master side state machine [1]. The formation state in the master side serves to narrow down the possible task gesture to be performed by the user, thus simplifying gesture recognition process [1]. The formation task state is not implemented in the slave side state machine because it is unnecessary to have non-task states in the state machine [1].The two state machines allow the master side and slave side to synchronize and perform teleoperation [1]. The 6-DoF manipulation performed in 3 finger mode and 5 finger mode, are realized in the manipulation state [1]. Only 5-DoF can be performed in 2 finger mode since, it is impossible to perform Z-Rotation in 2 finger mode.

The research focuses mainly on 6-DoF manipulation as it covers every action performed by the humans' every day [1]. Additional pistol grip tool manipulation is also implemented. Figure 2.2 shows the state machine in the slave side.

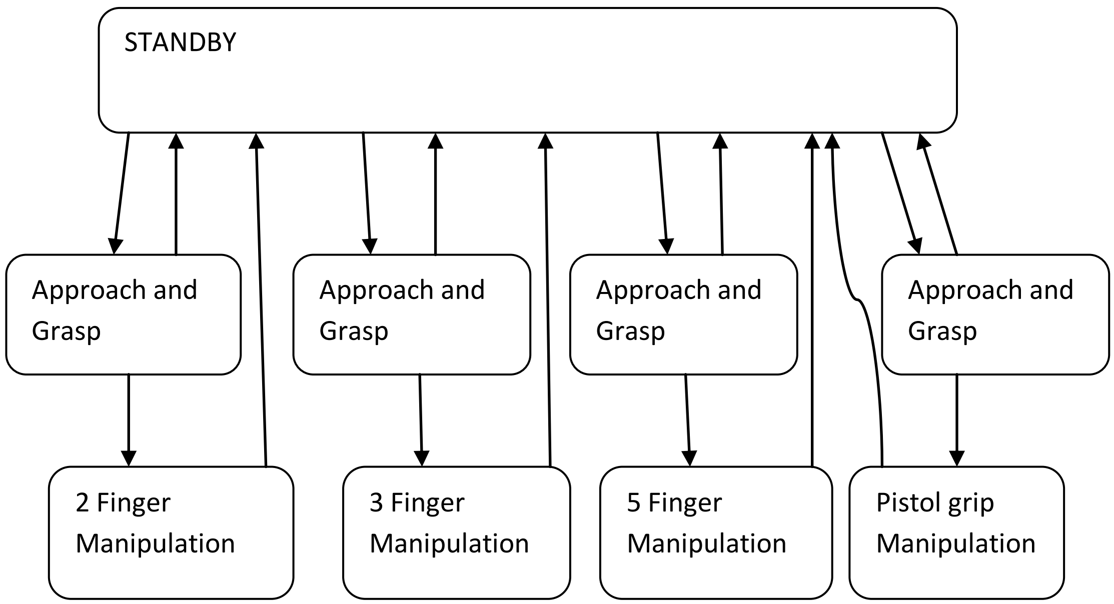

Figure 2.2 Slave side state machine

Source: [1]

The approach and grasp of pistol grip manipulation are preprogrammed as static gestures unlike the joint space mapping due to the lack of online grasp planner [1]. Certain applications require low bandwidth communication between master and slave. Since in task space telemanipulation, the intent of the user is executed by the robot hand. Task space manipulation requires low bandwidth for communication between the master and the slave. Only 8kb/sec is transferred when the gesture command is provided to the slave system.

2.2 HandyMan [1]

The Hand gesture command for grasping and manipulation (HandyMan) is employed to facilitate task space telemanipulation [1]. The HandyMan gesture recognition engine in the original work was created using Matlab commands crisp logic. This work uses fuzzy logic based HandyMan gesture recognition engine. As opposed to the original work with 43 task gestures, only 14 task gestures were designed for grasping and manipulation in all modes of operation. The gestures were created so that each gesture is unique and can be clearly distinguished from other gestures in the library.

The fuzzy rules were framed to recognize both the motion gestures and the static gestures in all modes of operation. For a standby command the user's hand s fully open [1]. This specifies the intent of the user to start a command or to restart the whole manipulation process [1]. The HandyMan standby gesture command from the use / fully open hand of the user triggers the TaskMan state machine [1]. Figure 2.3 shows the standby gesture of the human hand and the robot hand executing it.

Figure 2.3 Standby mode

There are two formation gestures (two finger formation and three finger formation). The standby gesture also acts as the five finger formation. Hence, five finger formation was not created. For three finger mode, the unused finger (pinky and ring) are flexed to command the robot end effector the intent of the user in using 3 finger mode. The robot hand does not retract the finger while commanding the intent of the user in using 2 finger or 3 finger mode, thus helping in reducing the work space obstruction [1].

Figure 2.4 Three finger formation

Four gestures of 14 gestures created are grasp gestures. Figure 2.5 shows three finger grasp. Seven unique manipulation gestures were created. No two combinations of hand joint positions are similar. This prevents overlapping of the manipulation gesture thus assisting in removal of possible drift errors during manipulation.

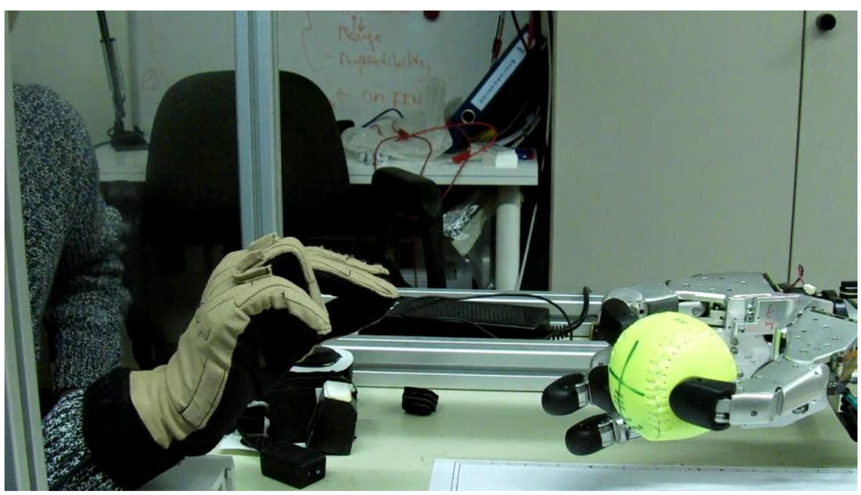

Figure 2.5 Three finger grasp

Although the manipulation gestures created in the original were robust, it was found that the manipulation gestures were not unique enough to avoid the false recognition by the gesture recognition engine. This led to drift errors i.e. displacement in the unintended DoF.

Each gesture in the fuzzy logic based task gesture library is framed are unique. The rules in the gesture engine aim to simplify the gesture for the user. Since the configuration of the human hand and the robot hand are different, the manipulation by robot hand does not exactly mimic the human gestures [1]. For example, X-rotation gesture performed by the human requires only the proximal joint of the thumb moving from one side to another, which will be reproduced by the robot hand. In case of Y-translation and Z-translation, the user will use the wrist abduction/adduction and flexion/extension action respectively. Whereas, the robot hand will follow the user's command by moving the fingers base joint and thumb's proximal joint. This proves the dexterity of the robot hand and inability of the human hand to perform certain tasks.

Totally 12 sensors of instrumented gloves are used for every gesture that the user performs. The fuzzy system recognizes the gesture with the combination of joint angle value and the rate of change of joint angle measurement.

In the original work the manipulation gesture types were recognized using the unique sensor rate combination with weight function:

$$fw = XPi - XNi - X|Zi| \qquad (1)$$

where Pi, Ni and Zi are the three types of measured rates (positive, negative, and non-moving) observed by each CyberGlove sensor [1]. In each data process cycle, the weight function fw is calculated for all 12 possible manipulation types [1]. The manipulation type with the highest fw value is the most likely the gesture type currently performed [1].

Unlike the original work [1] in this work when the user performs a gesture, the fuzzy inference system in HandyMan recognizes the unique combination of joint angle measurement and the rate of change of joint sensor measurement . The input from the user is mapped to the fuzzy rules the output of which is then mapped to a corresponding output, thus recognizing the gesture performed by the user. The identified gesture is then commanded to the robot.

Figure 2.6 CyberGlove

Source: [17]

Table 2.1 Sensors

Sensor Number	**Sensor Name**
1	Thumb roll / thumb base
2	Thumb proximal
5	Index base
6	Index proximal
8	Middle base
9	Middle proximal
12	Ring base
13	Ring proximal
16	Pinky base
17	Pinky proximal
21	Wrist flexion / P2
22	Wrist arch / P3

2.4 Implementation

The HandyMan and TaskMan were realized using 5 finger dexterous anthropomorphic hand which is based on DLR/HIT II hand [1]. The anthropomorphic hand has 15 degrees of freedom. Each finger of the anthropomorphic hand is identical and has three degree of freedom. The base joint of each finger is capable of performing abduction, adduction and flexion, extension. The proximal joint and the distal joints of the fingers are coupled. The thumb is placed opposing the index finger.

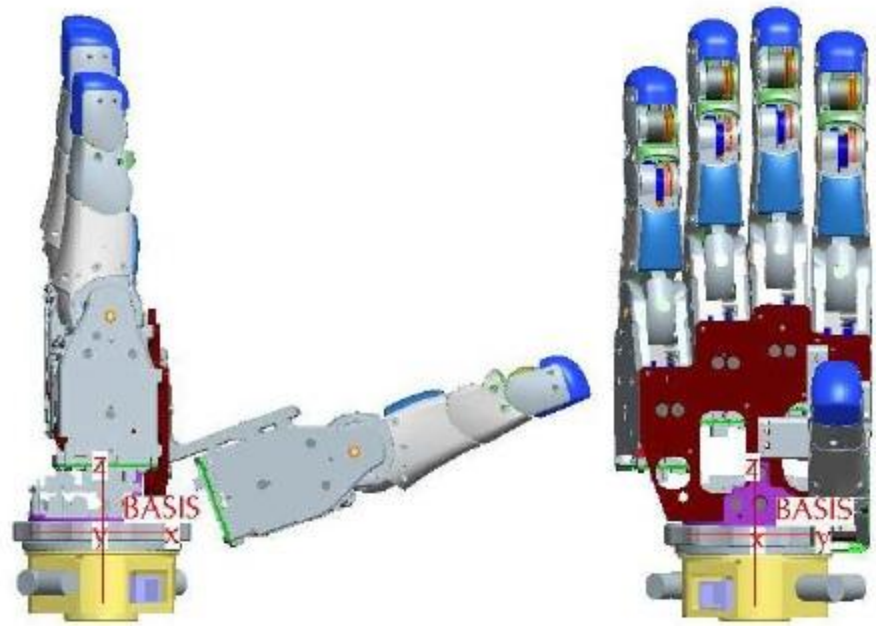

Figure 2.7 Five finger hand based on DLR/ HIT Hand II

Source: [1]

The HandyMan gesture recognition engine is implemented in a LINUX system using the Fuzzy Logic Toolbox in Matlab/Simulink. The Cyberglove is connected to the same linux system to retrieve the joint angle measurements and the rate measurements which are the inputs to the gesture recognition engine. The real time clock from the QNX real time system is used to synchronize with the robot hand. The fuzzy system is exported from the file where it is saved to the Matlab workspace every time before the Human Machine Interface (HMI) Simulink model is compiled. The inputs to the gesture recognition engine are inputted in the same order as that of the input variables in the fuzzy system. The impedance control to the robot is provided to the robot hand [1]. Controller is implemented on the ONX [1]. Linux system is used to host the model running on QNX[1].

In task space grasping and manipulation, the joint positions are not commanded directly and rather the tasks performed are recognized and commanded to the slave. The controller on the slave side receives the commands and executes it, the required bandwidth for communication between master and slave is reduced [1].

CHAPTER 3

MANIPULATION TYPE CLASSIFICATION

3.1 A Fuzzy Approach

Fuzzy logic is a problem solving technique that uses an imprecise spectrum of data to provide an approximate output that is most accurate. In a binary logic proposition output is either one or zero, and no intermediate values are allowed. In fuzzy logic everything is a matter of degree and is approximate. Fuzzy logic aims in providing an output/reasoning that is approximate and not exact. The Higher the complexity of the problem, the more generality of fuzzy logic is required. Knowledge about the world and the knowledge about the underlying probabilities play a vital role in decision making process. Fuzzy logic deals with the vagueness and the imprecision of input and provides inference that uses human reasoning ability to be applied to knowledge based systems. Fuzzy logic is based on a collection of variables that determine the reasoning capability of the system. In fuzzy logic everything is a matter of degree and exact reasoning is viewed as a limiting case. Fuzzy logic makes a human-like decision when the input is imprecise, vague or missing [28, 29].

Since the output of the CyberGlove is imprecise, fuzzy logic is used in this experiment. Fuzzy logic deals with giving an approximate result rather than one that is exact. Fuzzy logic arrives at a conclusion with the imprecise or vague input by mapping the input to the rules and thus deciding on the output. Manipulation type in this thesis is determined using the joint angle and joint rate as input to the fuzzy inference system. Fuzzy logic is a robust approach to many problems in the real world. It can be employed in complex systems at a low cost [11] and can be formulated in natural language, and works well with the imprecise or contradictory input, thus proving to be a powerful tool in dealing the ambiguity and nonlinearity. Fuzzy logic has a disadvantage of rule chaining, where this number of rules can grow exponentially with the accuracy and complexity of the problem [28, 29]. Figure 3.1 shows the basic fuzzy structure.

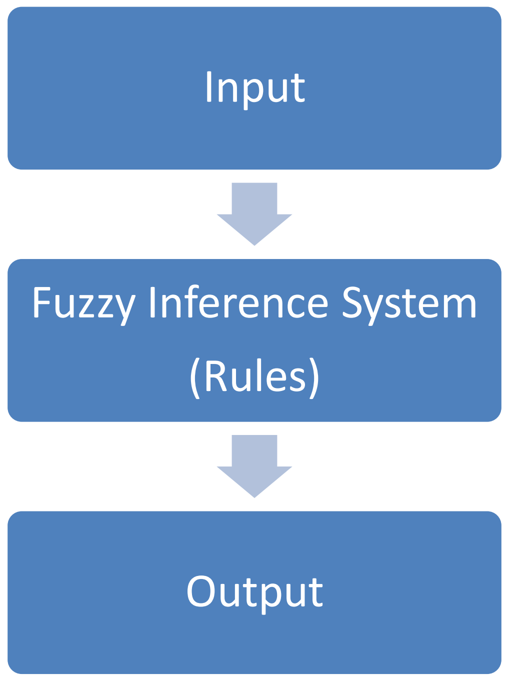

Figure 3.1 Basic structure of Fuzzy Logic**.**

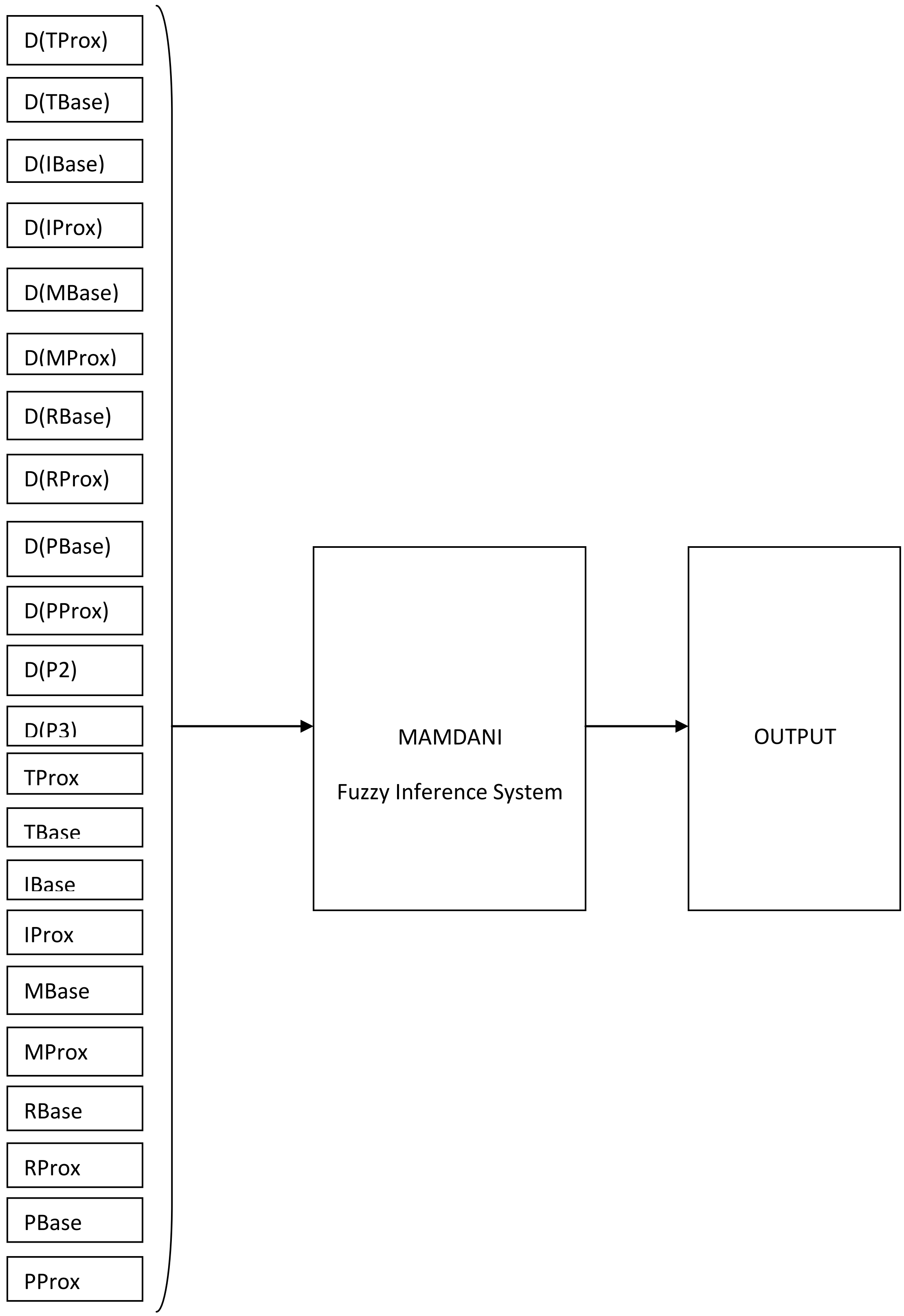

Figure 3.2 Fuzzy Logic structure

3.2 Gesture Recognition Engine

In this research, teleoperation of a 15-DoF robotic hand is realized using fuzzy logic in 6-DoF. The gesture recognition library consists of task gestures to perform manipulation in 6-DoF and an additional special purpose gesture for pistol grip manipulation. The Gesture recognition engine in this thesis is created using fuzzy logic toolbox in Matlab. Fuzzy system has five important parameters.

Type: Mamdani

There are two types of fuzzy interface methods. (1) Mamdani (2) Sugeno. The type used in the research is Mamdani, since it is well suited for the imprecise human input. The main difference between the two types is that Sugeno has linear output membership function whereas the Mamdani type has nonlinear output membership function [21].

Logical Operation: AND

Two logical operators in a fuzzy system are AND and OR. Default values are taken for both the logical operators. Hence AND method will be 'min' and OR method will be 'max'. Therefore the logical output minimum of two inputs is provided to the fuzzy system while using the AND operator and output while using OR operator is the maximum of two inputs [21].

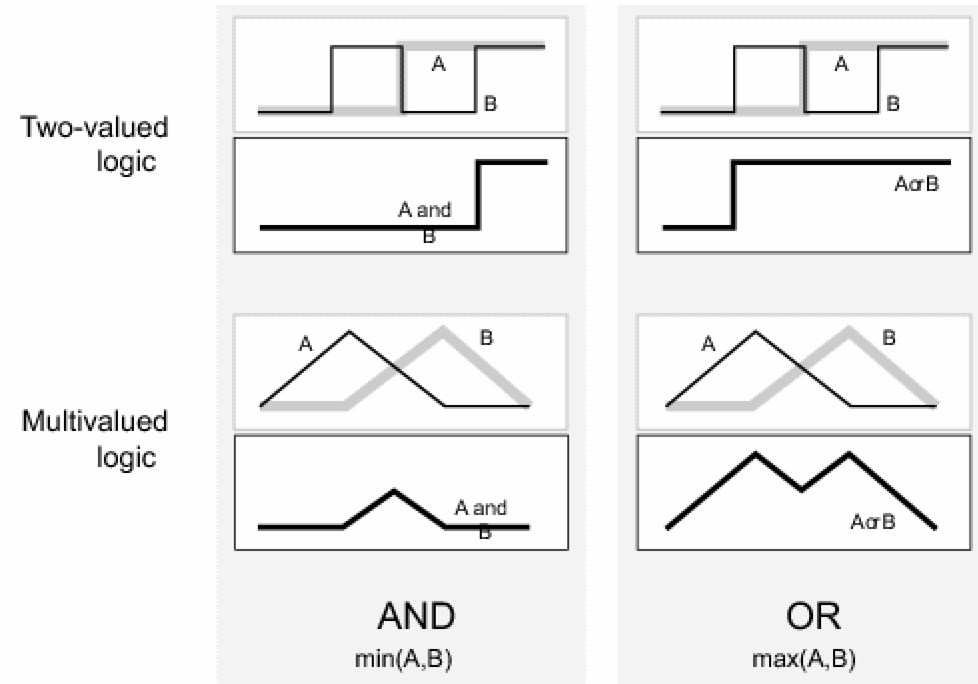

Figure 3.3 Logical operators

Source: [21]

Table 3.1 Fuzzy 'And' and 'Or'

Source: [21]

AND

A	B	min(A,B)
0	0	0
0	1	0
1	0	0
1	1	1

OR

A	B	max(A,B)
0	0	0
0	1	1
1	0	1
1	1	1

Implication Method: Min

Interpretation of an if-then rule involves two distinct parts. If the antecedent is true to some degree of membership, implication modifies the output fuzzy set to the degree specified by the antecedent. There are two types of implication method. 'min' truncates the output fuzzy set and 'prod' scales the output fuzzy set [21]. The type of implication used in this paper is 'min'. An example of 'min' implication is shown in figure 4.2

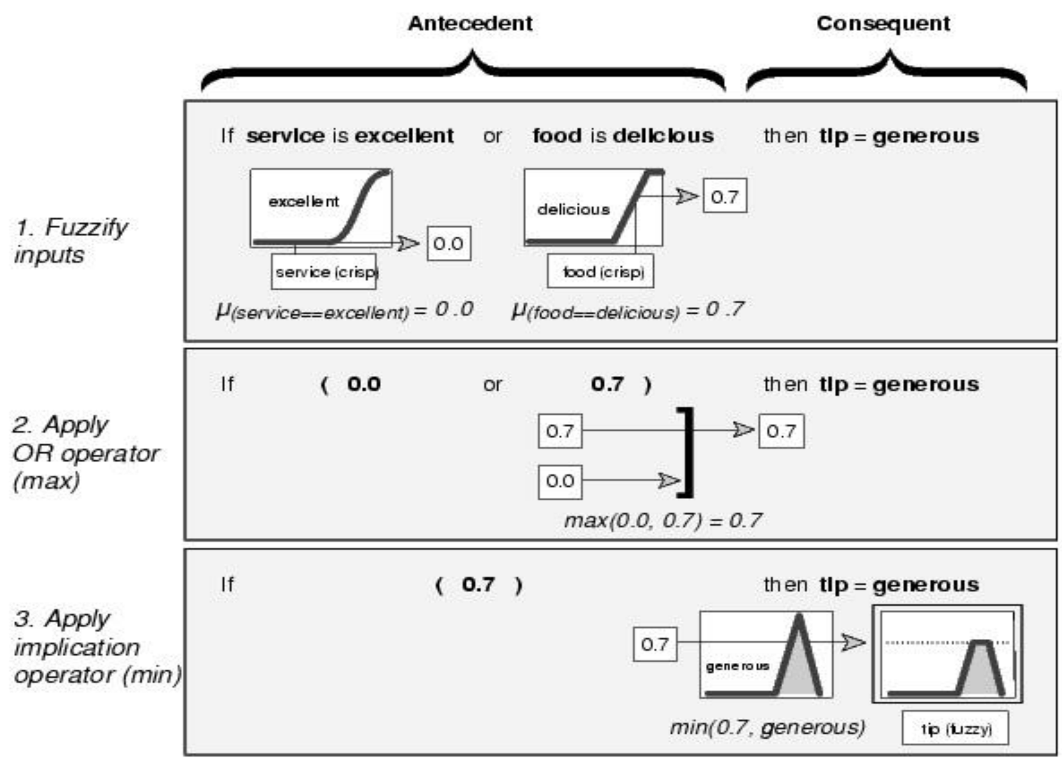

Figure 3.4 Implication example

Source: [21]

Aggregation Method: Max

Aggregation is the process by which the fuzzy sets that represent the outputs of each rule are combined into a single fuzzy set. The aggregation occurs before defuzzification. The input of aggregation process is the list of truncated output functions returned by the implication for each rule. Output of aggregation process is one fuzzy set for each variable. The 'max' operator is by-far the most common implementation of rule aggregation operation. According to this the overall fuzzy output is calculated by taking the maximum truth value from set of individual outputs where one or more terms overlap. The other common operators in aggregation of fuzzy actions are 'sum' and 'probor' [13].

Defuzzification: Centroid

The main aim of the defuzzification method is to provide a crisp output value. The design of defuzzification method is very important since it will affect the interpretation of fuzzy response. The technique used in this work is centroid defuzzification. The crisp output value is obtained by deriving the centroid of the output membership function. Other defuzzification methods are bisector, largest of maximum, smallest of maximum and middle of maximum. The centroid defuzzification technique provides more appropriate output value than the other defuzzification types [13].

3.2.1 Creating Gesture Recognition Engine

Defining the input variables: The inputs to the fuzzy system are the joint angle measurements and the joint rate measurements. The rate of change of joint sensor measurement determines the speed at which the particular joint is moving. Joint angle measurements define the angle of the joints and play a key role in differentiating the manipulation type along with the rate measurement. The input variables in this thesis are joint angle measurements, Tbase, Tprox, Ibase, Iprox, Mbase, Mprox, Rbase, Rprox, Pbase, Pprox and rate measurements, D(Tbase), D(Tprox), D(Ibase), D(Iprox), D(Mbase), D(Mprox), D(Rbase), D(Rprox), D(Pbase), D(Pprox), D(P2), D(P3).

Setting range and creating the membership functions: Membership function is a graphical representation of input value range. The fuzzy rules use the membership functions as a weighting factor to determine their influence on the fuzzy output set. Ultimately the membership functions determine an output response. There are different types of membership functions in real world application. Among them are triangular, trapezoidal, gaussian, piecewise linear and bell shaped. Membership functions can take any form but user defined shapes increase the complexity of the fuzzy system. The membership function used in this thesis is triangular membership function (Trimf) because of its less mathematical complexity and represents a rational basis in decision making processes.

The membership function is defined and the number and type of membership functions are selected. The rate measurements have ten membership functions making the gesture recognition engine recognize fine task based movements produced by the user and joint angle measurements have two or three membership functions depending on use of the joints during manipulation. This determines the joint position while performing a gesture.Once the input variables are defined and the membership functions are set the range of each input variable are set. The range is set depending on the range of motion of each joint and the range may vary for every joint. Based on the experiment carried out, the range has been optimized to obtain the most comfortable finger positions pertaining to the research.

Defining the output variable: The output variable is ‘Manipulation Type’. The membership is defined for the output variable and has eight membership functions of type ‘Triangular’. Each membership function defines standby state or a formation state or a manipulation type.

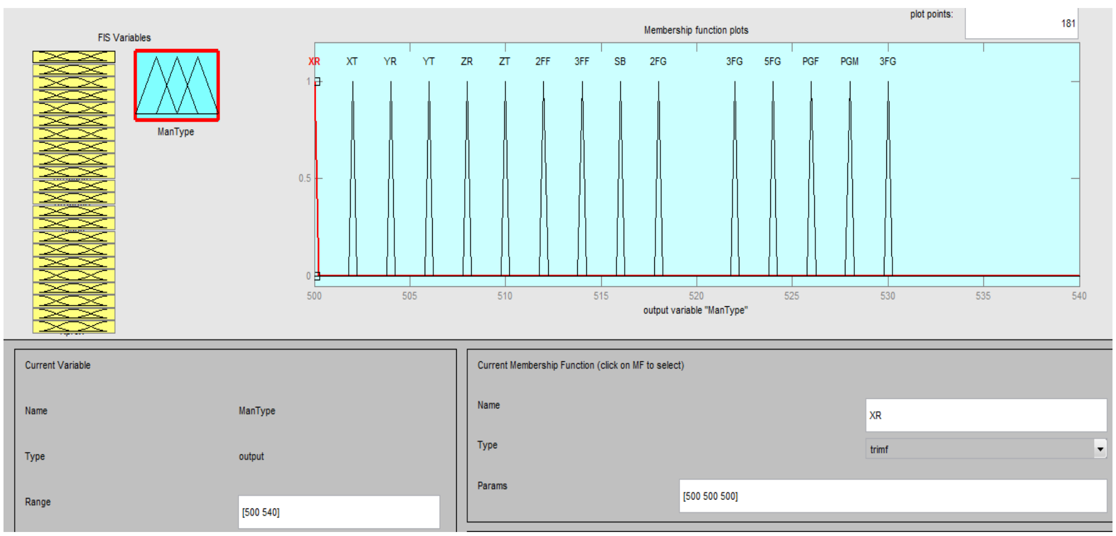

Figure 3.5 Output membership function

Gesture Recognition Rules: The rules were created in such a way that the system does not detect and perform manipulation in unintended DoF. A set of rules were created for each manipulation type, formation and standby. Table 3.2 shows the names of membership functions and the range of values for every joint angle.

Table 3.2 Input membership function and range

Membership Function / **Joint Angle**	**5**	**45/50**	**90**
TBase	55 to 97.8	91.39 to 193.6	187.2 to 212.8
Ibase	35 to 61.1	53.21 to 182.1	174 to 206
Mbase	10 to 90.55	82.78 to 207.2	199.5 to 230.5
Rbase	50 to 65.55	57.78 to 182.2	174.5 to 205.5
PBase	40 to 68.67	58.33 to 191.7	183.3 to 216.7
Tprox	80 to 112.2	110.7 to 210	-
IProx	10 to 56.67	-	48.33 to 200
MProx	45 to 72.22	66 to 163.9	157.8 to 182.2
RProx	55 to 70	62.5 to 182.5	175 to 205
PProx	30 to 85.22	80 to 174	167.8 to 192.2

5, 45/50 and 90 are the names of the membership function. Change in the name of a membership function will not affect the fuzzy rules that were constructed.

Table 3.3 Output membership function

Manipulation Type	Fuzzy Output
X-Rotation	500
X-Translation	502
Y-Rotation	504
Y-Translation	506
Z-Rotation	508
Z-Translation	510
2 Finger Formation	512
3 Finger Formation	514
Standby	516
Pistol Grip Manipulation	528

The gesture recognition rules rely heavily on the joint angle measurements so as to avoid the false recognition that posed a problem in the original work. While performing manipulation, every manipulation gesture has at least one combination of joint positions which shared by one or more manipulation gestures. To avoid false recognition by the gesture recognition engine, every gesture was created to be unique. This work aims at creating gestures that look similar to the tasks to be performed by the robotic hand. The gestures have at least one aspect that differentiates it from the other gestures, yet, allows the user to perform the gestures that is similar to the task movements. Table 3.4 shows the gestures created to perform teleoperation in two finger, three finger and five finger mode of operation and number of fuzzy rules used for each gesture type.

Table 3.4: Gesture type and number of rules

GESTURE TYPE	NUMBER OF FUZZY RULES
Standby	5
Two finger formation	1
Three finger formation	5
XR	78
XT	18
YR	9
YT	9
ZR	38
ZT	11
Pistol grip manipulation	11

The joints must be positioned within the range specified in the rules to be recognized as the corresponding gesture. The rules of standby, two finger formation and three finger formation are shown.

Standby: The rules for the standby state are created in such a way to make it robust. Since the Standby is a static gesture the rate of change of joint sensor was not considered while creating the rules. The output of the fuzzy system is Standby for different combinations of joint angles. When the joint angle value of TBase is 5, IBase is 5, MBase is 5, RBabse is 50, PBase is 50, Tprox is 5, IProx is 5, MProx is 5, RProx is 50, PProx is 5 or

if the joint angle value of TBase is 5, IBase is 5, MBase is 5, RBabse is 50, PBase is 50, Tprox is 5, IProx is 5, MProx is 5, RProx is 50, PProx is 50 or.

if the joint angle value of TBase is 5, IBase is 45, MBase is 5, RBabse is 50, PBase is 50, Tprox is 5, IProx is 5, MProx is 5, RProx is 50, PProx is 5 or.

if the joint angle value of TBase is 5, IBase is 45, MBase is 5, RBabse is 50, PBase is 50, Tprox is 5, IProx is 5, MProx is 5, RProx is 50, PProx is 50 the gesture is identified as standby.

Figure 3.6 Standby

Formation: The rules were created to recognize the two finger formation and three finger formation. Since, during formation there is no movement in the finger joint angles, rate measurements were not considered while creating the rules.

Two Finger Formation: The thumb base joint angle is 5, Index base joint angle is 5, middle finger's base joint angle is 45, ring finger's base joint angle is 50, pinky's base joint angle is 50 for a two finger formation.

Figure 3.7 Two finger formation

Three Finger Formation: Four combinations of joint angles were used to make the three finger formation robust.

(1) TBase is 5, IBase is 5, MBase is 5, RBase is 50, PBase is 50, TProx is 5, IProx is 5 MProx is 5, RProx is 90, PProx is 50.

(2) TBase is 5, IBase is 5, MBase is 5, RBase is 50, PBase is 50, TProx is 5, IProx is 5 MProx is 5, RProx is 5, PProx is 50

(3) TBase is 5, IBase is 5, MBase is 5, RBase is 50, PBase is 50, TProx is 5, IProx is 5 MProx is 5, RProx is 50, PProx is 50

(4) TBase is 5, IBase is 5, MBase is 5, RBase is 50, PBase is 50, TProx is 50, IProx is 5 MProx is 5, RProx is 50, PProx is 50.

If the user performs one of these combinations then the system recognizes the gesture as three finger formation.

Figure 3.8 Three finger formation

Five Finger Formation: The joint angle measurements for the five finger formation are same as that of the joint angle measurements used for standby position. Hence, no separate rule was created for five finger formation.

MANIPULATION: Manipulation is a complex process. When the user performs the manipulation gesture, each gesture must be unique and should not overlap any other manipulation gesture. Overlapping of gestures while performing manipulation may result in false recognition of gestures. False recognition of gestures leads to drift error.

The fuzzy logic-based task space telemanipulation uses a threshold (range of values) within which the joints should be positioned while performing the gestures. Every manipulation gesture created has at least one aspect that differentiates it from other manipulation gestures. Joint angle plays a major part in making each manipulation gesture unique. Table 3.4 shows the joints that make the manipulation gestures unique.

Table 3.5: Joints that make the gestures unique

MANIPULATION TYPE	**JOINT(S) THAT MAKE THE GESTURES UNIQUE**
XR	Index Proximal
XT	Ring Proximal and Pinky Proximal
YR	Ring Proximal and Pinky Proximal
YT	Pinky Proximal
ZR	Thumb Proximal
ZT	Middle Proximal

The gesture recognition engine recognizes the manipulation gesture by identifying the combination of joint positions and the specified rate of change of joint sensor measurement. For example, during X-Translation, the pinky's proximal joint is placed between the range 45 and ring finger's proximal joint is placed between the range 90. No other manipulation gesture is created to have the pinky's proximal joint and ring finger's proximal joint in the same range. Thus with unique pinky and ring finger proximal joint positions, overlapping of X-Translation gesture with other manipulation gestures was avoided thereby, reducing the drift error to zero. Similarly, every manipulation type uses at least one joint to make the manipulation gesture unique. Figure 3.9 shows the difference in each manipulation gesture.

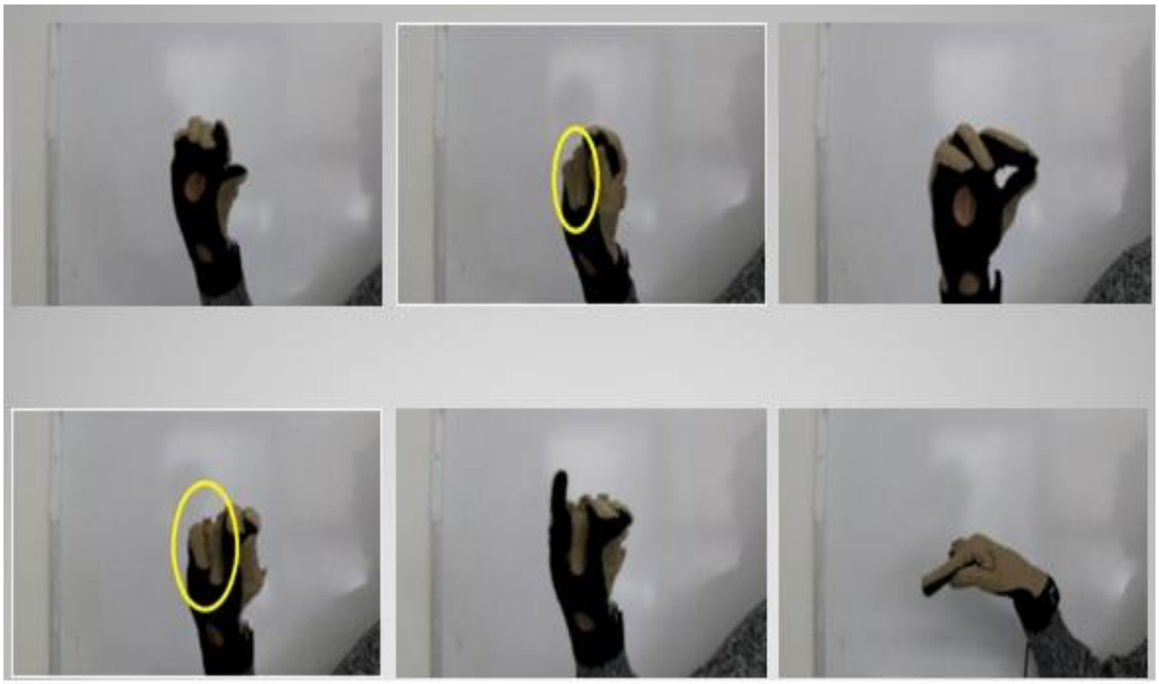

Figure 3.9: Manipulation gestures

X-Rotation: Thumb proximal joint velocity is not zero. Index proximal joint position makes the manipulation gesture unique.

Figure 3.10 X-Rotation

X-Translation: Although the velocity of index proximal, middle proximal, thumb proximal, index base, middle base, thumb base are non zero while performing the gesture, the fuzzy rules were created to consider only the velocity of index base and thumb proximal joint in order to recognize the gesture. Particular combination of finger joint positions is maintained while performing the gesture. The ring finger proximal joint and the pinky finger proximal joint helps avoiding the overlap of gestures.

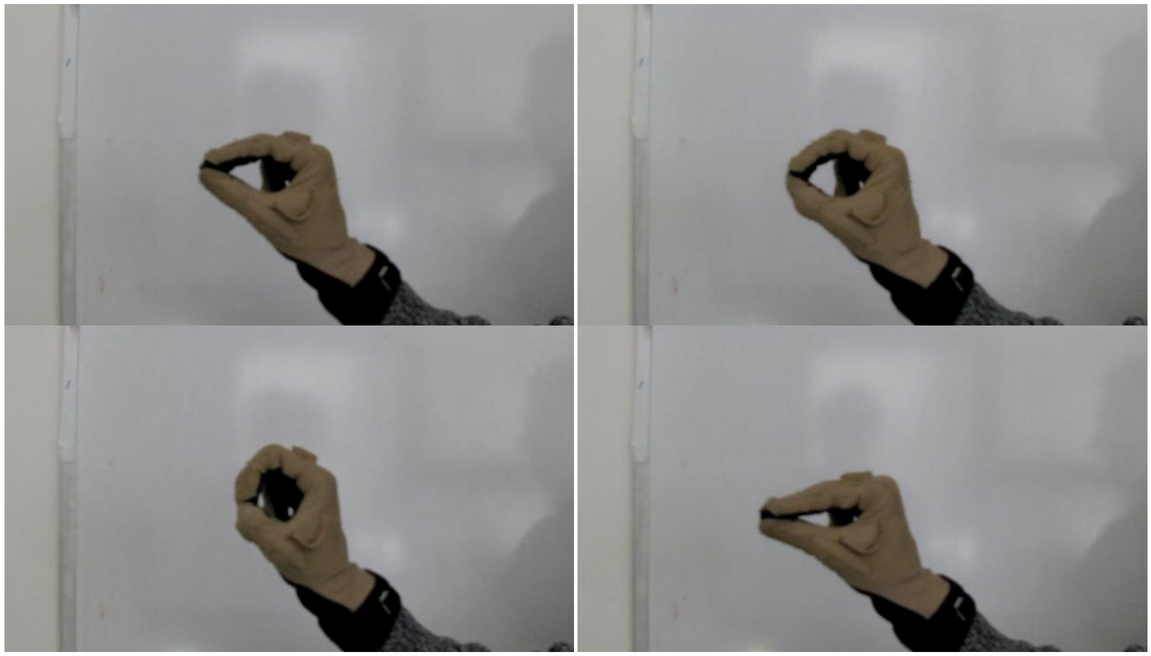

Figure 3.11 X-Translation

Y-Rotation: Similar to the X-Translation, during Y-Rotation only the index proximal and thumb base joint velocity is considered along with a particular combination joint positions. Ring finger proximal joint and pinky's proximal joint position during the manipulation makes the gesture unique. The movements of the thumb base and index proximal joint should be in the opposite direction.

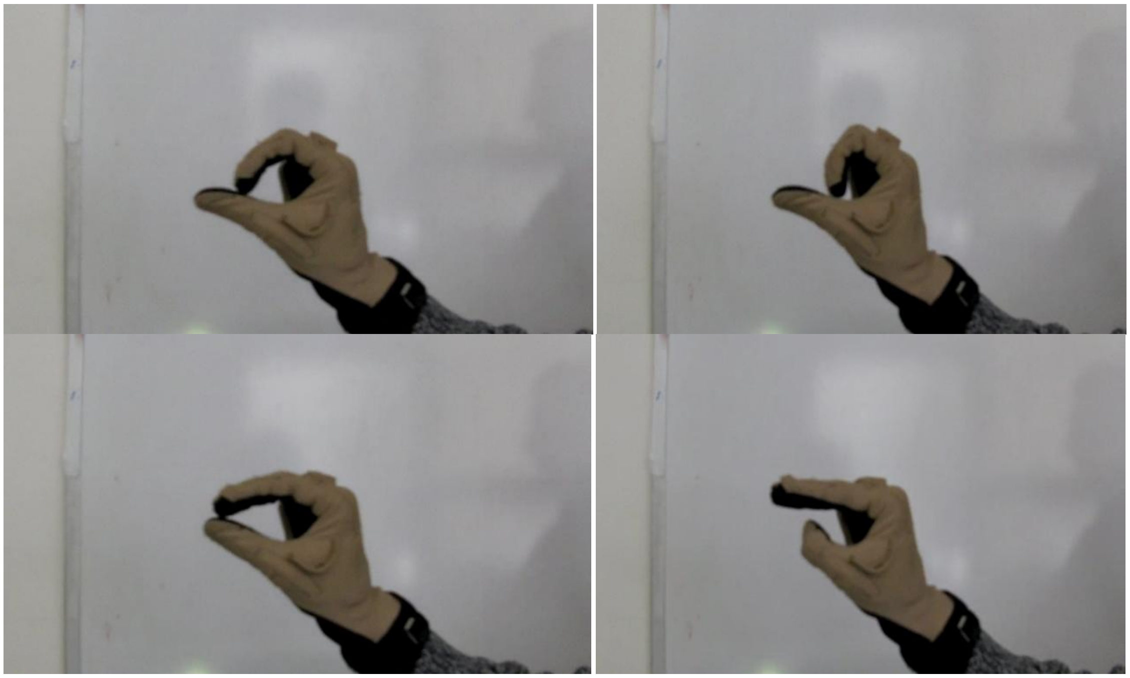

Figure 3.12 Y-Rotation

Y-Translation: The gesture recognition engine recognizes the rate of change of wrist arch sensor measurement. While performing the Y-Translation gesture, humans have tendency to perform Z-Translation. The pinky proximal joint position again plays a major role in distinguishing the overlap of Y-Translation gesture on Z-Translation. Thus, reducing the drift in the unintended DoF to zero.

Figure 3.13 Y-Translation

Z-Rotation: The rules were created so that the gesture recognition engine considers the middle finger's base joint velocity along with a particular combination of other joint positions. Thumb proximal joint position aids in making the gesture unique.

Figure 3.14 Z-Rotation

Z-Translation: While performing Z-Translation gestures, humans tend to perform Y-Translation. The middle finger's proximal joint assists in preventing the superimposing of Z-Translation gesture and Y-Translation gesture. The gesture recognition engine considers the rate of change of wrist flexion sensor measurement for identifying the gesture type.

Figure 3.15 Z-Translation

Pistol Grip Manipulation: Additionally fuzzy rules were also created to grasp and manipulate the pistol grip tools. The gesture recognition engine recognizes the rate of change of the index finger proximal joint sensor along with the other joint angle position.

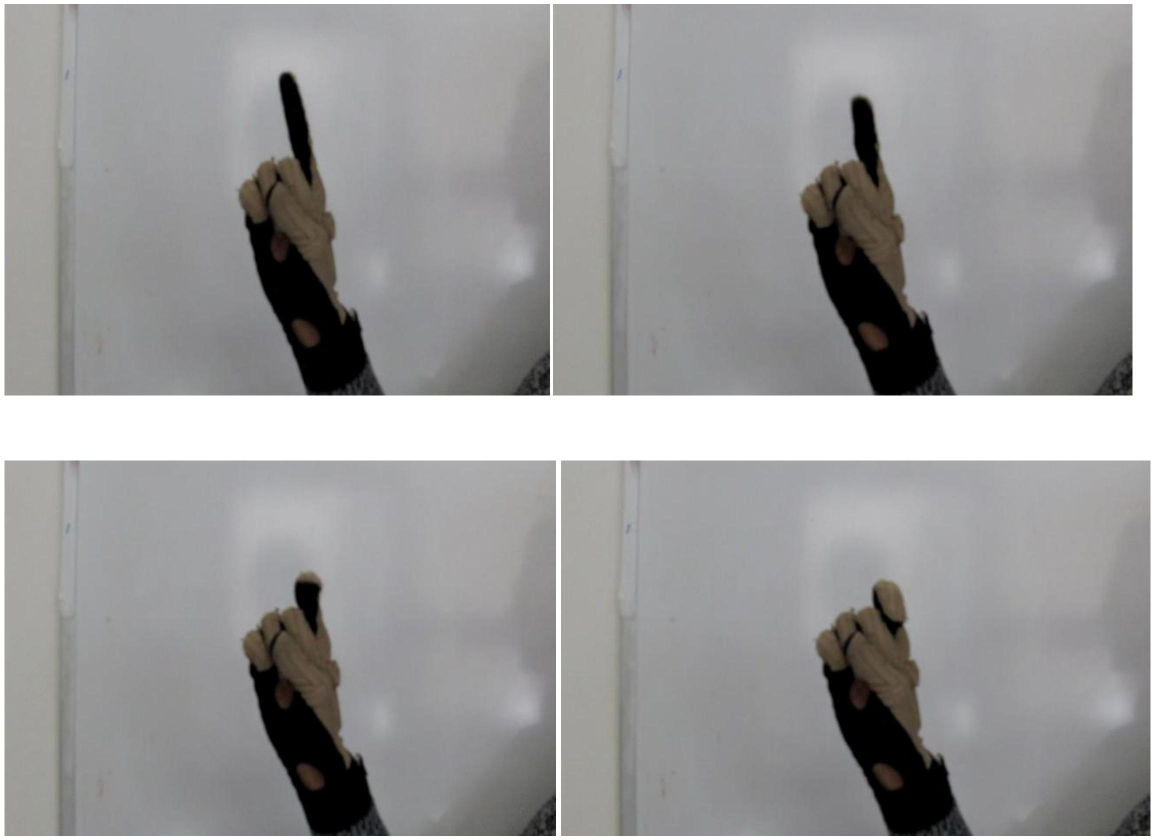

Figure 3.16 Pistol Grip Manipulation

3.3 Displacement Processor

Once the manipulation type is determined, the translational and rotational displacement is determined. The determined displacement is then commanded to the robot hand. The rate of change of the joint sensor measurement along with the gesture recognition engine output determines the displacement in the intended DoF. The displacement is calculated for every manipulation type. The rate of change of sensor value determines the displacement in the intended DoF. The displacement value is determined using the following formula.

$$\text{Displacement} = \text{kgain}^{*}\ (\text{DoFCurrent} - \text{DoFPrevious}) \qquad (3.1)$$

DoFCurrent is the current glove sensor value and DoFPrevious is the glove sensor value during the time of grasp/manipulation. If the K_{gain} value is low, the manipulation is fine.

Even if the operator continues to perform multiple DoF manipulation, robot hand performs analogous manipulation without dropping the object, until the TaskMan state machine moves to formation state from manipulation state on the HMI side. The displacement processor helps in identifying the displacement / drifts in unintended DoFs. Table 2.2 shows the manipulation type and the sensors that are considered to determine the manipulation type.

Table 3.6 Manipulation type and displacement sensors

Manipulation Type	**Sensor**
X-Rotation	Thumb Proximal
X-Translation	Index base and Thumb Proximal
Y-Rotation	Index Proximal and Thumb base
Y-Translation	Wrist Arch
Z-Rotation	Middle Proximal
Z-Translation	Wrist Flexion
Pistol Grip	Index Proximal

Once the displacement is calculated, the displacement command is then sent to the robot hand. 90 8bit displacement commands per second is sent to the robot to perform the translational and rotational displacement.

CHAPTER 4

RESULTS AND ISSUES

The result in this work is based on 3 finger manipulation, since the manipulation in 6-DoF requires minimum of 3 fingers. 6-DoF telemanipulation using Fuzzy Logic base gesture recognition system was compared against the earlier command algorithm based gesture recognition system during task space telemanipulation. The original HandyMan (command algorithm) based telemanipulation provided DoF commands which had an advantage very good repeatability, but manipulation in one DoF had drifting of other DoF commands [1]. This is due to the recognition of two or more commands simultaneously, in other words due to false recognition of DoF commands. The drift error becomes high and consistent making the DoF uncommandable [1]. During manipulation full range of motion is not always achieved, which is due to the respective joint position during manipulation or may be due to cyberglove error.

The fuzzy logic based gesture recognition engine uses rate of change of sensor measurement and the joint angle measurement together to create the gesture library. By creating new gestures that have at least one aspect different from the other gestures, the drift error is reduced to zero. Each grasping/manipulation type rule(s) has at least one difference from other manipulation type rule(s). For example the value of pinky's proximal joint during Y-translation falls between 30 to 85.22, which is not the same for any other manipulation type. Since joint positions are considered to recognize the manipulation type, the operator should be careful during teleoperation since changes in joint angle may cause drift errors. Since, the fuzzy logic based task space telemanipulaiton uses a threshold (range of values) within which the joints should be positioned while performing the gestures, the technique is less robust. The operator may feel uncomfortable during the teleoperation as every joint has to be within a particular angle range. If there is a change in joint angle during telemanipulation, the intended DoF command does not reach the maximum.

Figures 4.1 and 4.2 shows the dominant axis recognized while performing the X-translation gesture. The fuzzy logic based task space telemanipulation shows no drift as opposed to the original work. Each frame contains all the manipulation gestures i.e. every frame has information about all the gestures and the figures 4.1 and 4.2 shows the dominant axis recognized and other manipulation gestures detected while performing a manipulation type.

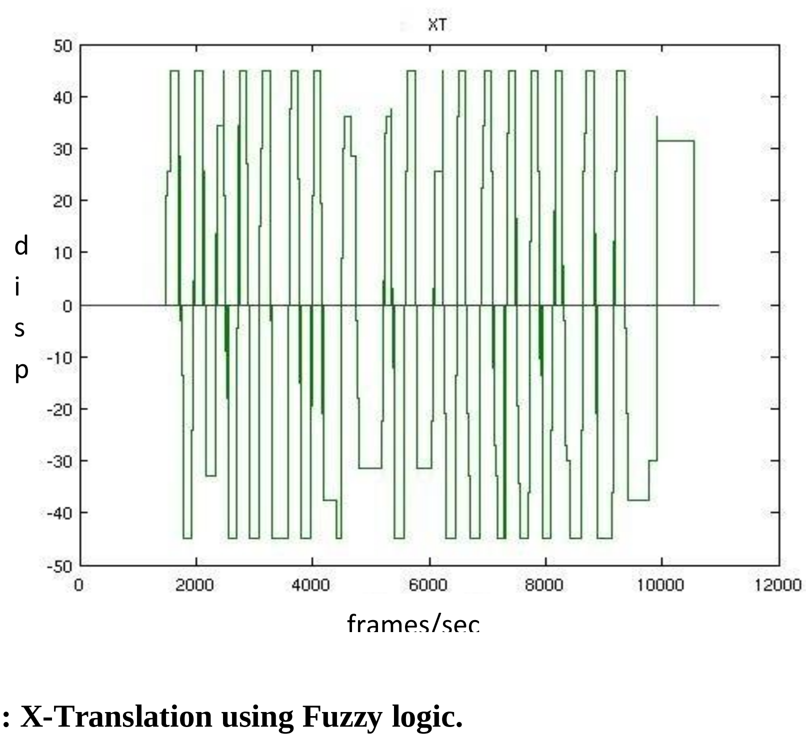

Figure 4.1: X-Translation using Fuzzy logic.

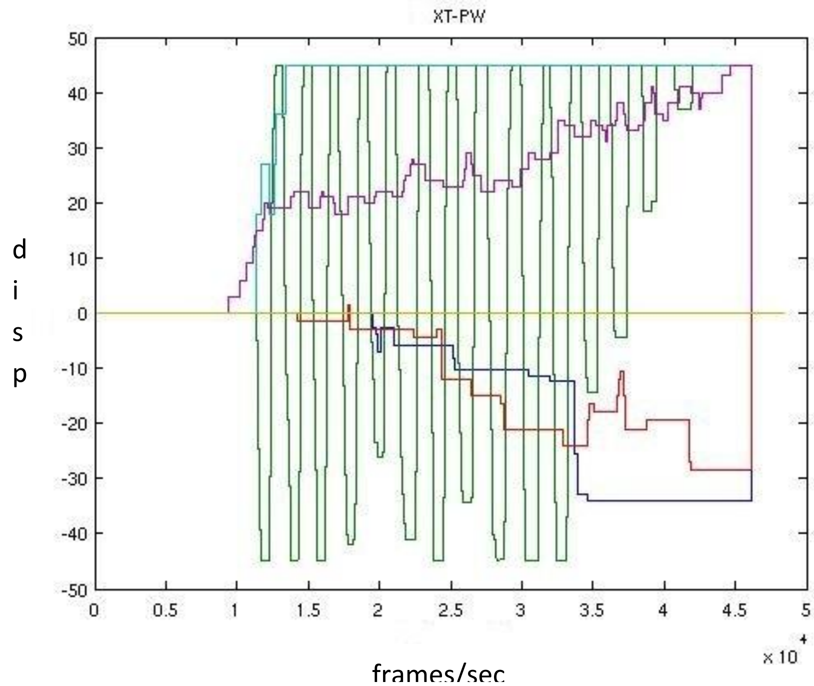

Figure 4.2: X-Translation using command algorithm

As can be seen in Figure 4.1 only X-Translation was detected while X-Translation gesture was performed and the other five unintended manipulation gestures do not get recognized in fuzzy based system. Whereas in the original work (Figure 4.2) while performing X-Translation gesture, the gesture recognition engine falsely recognizes the unintended manipulation gestures. The new gestures that were introduced solved the drift problem, but it has complexity issue since every joint should be within a particular range while performing the gestures. The hand gestures in this research are complex and hence it is very difficult for someone to learn the gesture and perform teleoperation. The range of motion achieved during the manipulation tasks was proved to be better than the original work. Another main issue is the jitter problem. Jitter problem occurs mainly because of the failure by the user to maintain the complex hand gestures during manipulation. Jittering of displacement commands may cause damage to the hardware.

But during the teleoperation experiments the jitter problem was found very low. The jitter problem can be avoided completely with good practice of gestures.

Figures 4.3 and 4.4 show the dominant axis recognized while every frame has information about all the gestures. In case of X-Rotation (Fuzzy based) only the intended action is recognized and every other manipulation gesture stays at zero. Whereas in the original work (figure 4.4) unintended manipulation types are also recognized.

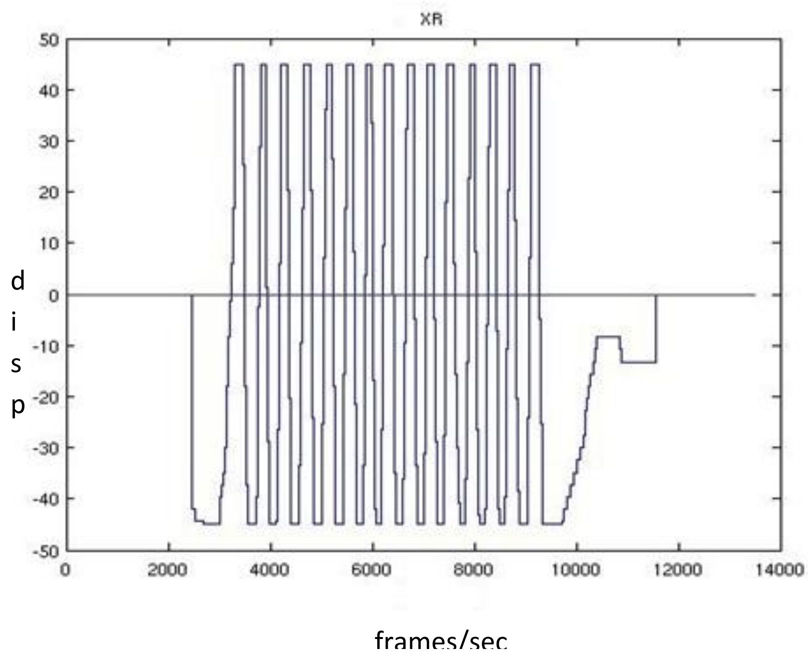

Figure 4.3: X-Rotation using Fuzzy logic

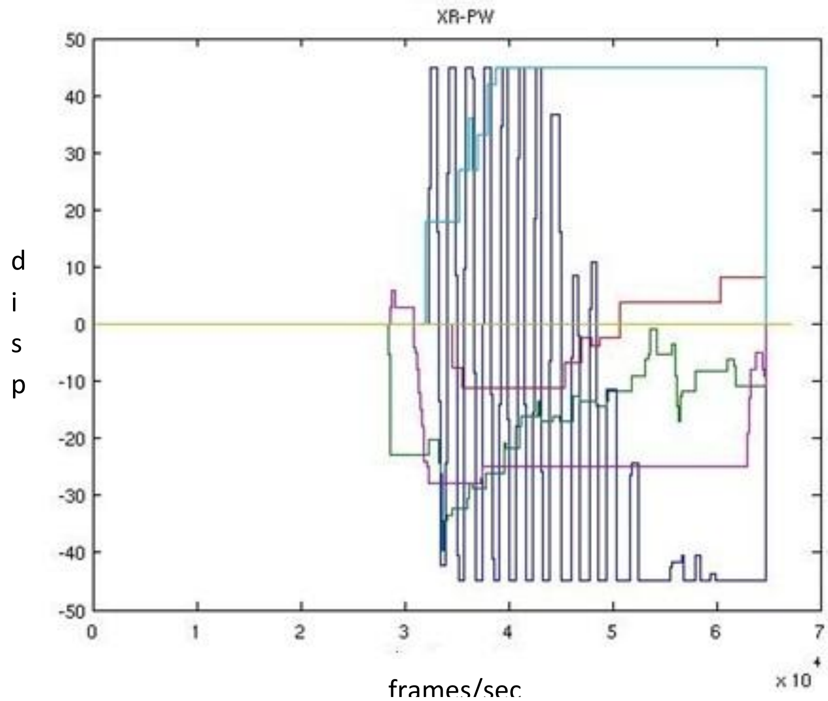

Figure 4.4: X-Rotation using command algorithm

Figure 4.3 shows a jitter problem while starting the manipulation process. It should also be noted that the jitter problem is present only while starting the manipulation. The remaining part of the plot shows fine manipulation.

It has also been found that of using 17 different manipulation gestures that were created before employing only 6 manipulation for 6-DoFmanipulation in 3 finger and five finger modes and 5-DoF manipulation in 2 finger mode may be more stressful since the user has to get used to 17 different combination of joint positions. Six gestures are enough to perform the manipulation in all 3 modes. Once the state machine enters a formation state, the number of fingers used to perform a manipulation type will not affect the number of fingers used by the robot hand i.e. if the HMI state machine is in two finger manipulation state, the user can still perform three finger manipulation gestures and make robot hand perform two finger manipulations. Hence, the gestures used for two finger manipulation, three finger manipulation and five finger manipulation are the same in this thesis. This is because the HMI state machine will not change its state until it is commanded by the user. However, using 17 different gestures will also provide the same result as using 6 gestures. Using only 6 gestures takes comparatively lesser time to get used to.

Pistol grip manipulation was also performed with no drift errors but has an issue of jittering if the user does not get used to the hand gesture and should have proper finger joint angles while entering into the manipulation state to avoid the jitter problem. The pistol grip manipulation using the command algorithm does not have any drift errors or jitter problem. Comfortable joint positions, no jitter problem during the pistol grip manipulation using command algorithm makes it superior compared to the pistol grip manipulation using the fuzzy based recognition system.

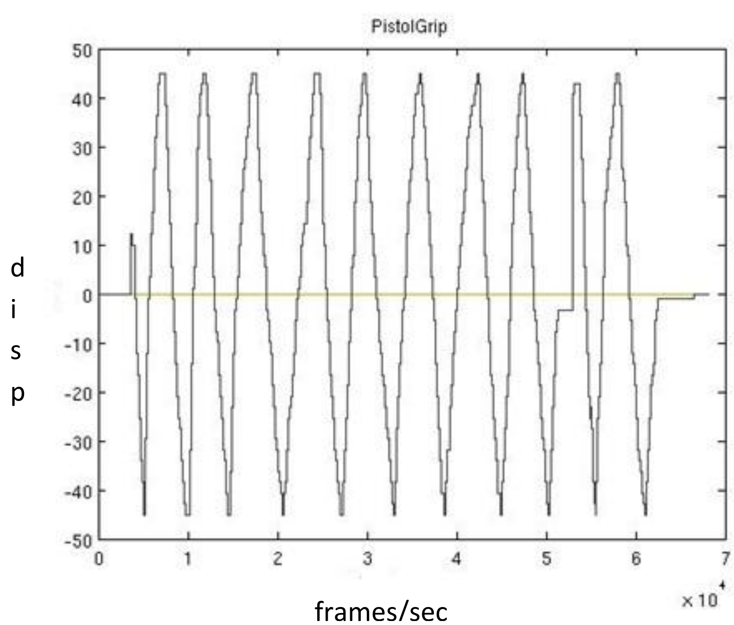

Figure 4.5 Pistol Grip Manipulation using Fuzzy Logic

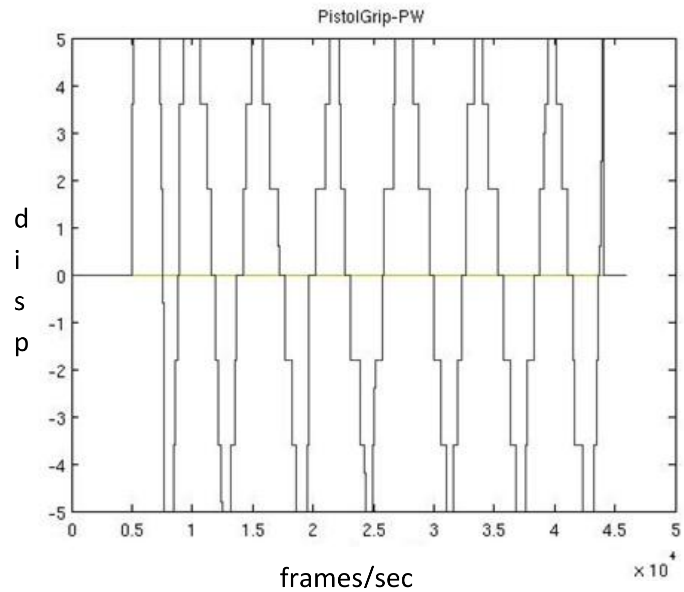

Figure 4.6 Pistol Grip Manipulation using Command Algorithm

The remaining manipulation types, Y-Rotation, Y-Translation, Z-Rotation and Z-Translation show similar patterns (like figures 4.1, 4.2, 4.3, 4.4) and prove that, the fuzzy logic based hand gesture recognition system to be better in reducing the drift error than the original work in 6-DoF (XR, XT, YR, YT, ZR, ZT). Although, both fuzzy based and command algorithm based pistol grip manipulation does not have any drift error, the fuzzy based pistol grip has jitter problem and complex hand gesture unlike the original work. This makes pistol grip manipulation using command algorithm to be superior. The task library used in this work can be easily improved and various gestures can be added.

CHAPTER 5

CONCLUSIONS

This work presents a fuzzy logic based gesture recognition engine for task space telemanipulation. The input to the fuzzy inference is joint angle measurement and velocity of the joint. The input is retrieved using the Cybergloves and the unwanted sensor measurements are terminated before being fed to the fuzzy logic controller. The membership functions of the input were created by trial and error technique. This work concentrates on 6-DoF manipulation and an additional pistol grip manipulation. Unique gestures were created to avoid the drifting in unintended DoF.

Experiments were carried out by hooking the robot hand to the Human Machine Interface (HMI) Matlab/Simulink model. Comparisons were made between the task space manipulation using fuzzy logic and the original work. Use of fuzzy rules and unique manipulation gestures does not have any drift error during manipulation. But the gestures are complex when compared to the original work.

The complex hand gesture problem should be solved with a robust fuzzy rule based task gesture library. The membership function has to be remodeled so as to avoid the jitter problem that was faced in this work.

The gesture library consists of task gestures and can be applied only for rigid objects. It should be expanded to non rigid objects and complex objects like the pistol grip tool manipulation gesture that was presented in this work. The telemanipulation experience can be increased by introducing force feedback.

APPENDIX A

CODE FOR CALCULATING JOINT ANGLE DISPLACEMENT

The joint displacement is calculated for every manipulation type is calculated. There may be two or more joints having displacement while performing manipulation. The joints that get displaced during each manipulation type are identified.

```
function [clockout,TP, fzout] ...
   = fcn(clockinPrevious, rawglovedata,tps,TPin,...
       TPPrevious, fzin)


fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%thumb proximal
if clockinPrevious==clockout

   TP=TPPrevious;
% if no manipulation is performed then no displacement
elseif fzin==520
   TP=0;
%if manipulation type is XR,YR,XT then thumb proximal is joint angle value is the
displacement
elseif fzin==500||504||502
    TP=TPin;
% if the XR,XT or YR continues then change in joint angle value is added to
% the previous thumb proximal
else
   if fzin==500||504||502
      TP=TPPrevious+tps;
   else
      TP=TPPrevious;
   end
end

function [clockout,TB, fzout] ...
   = fcn(clockinPrevious, rawglovedata,tbs,TBin,...
       TBPrevious, fzin)


fzout=fzin;
```

```
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%thumb base
if clockinPrevious==clockout

   TB=TBPrevious;
 % if no manipulation is performed then no displacement
elseif fzin==520
   TB=0;
 %if manipulation type is XT then thumb base is joint angle value is the displacement
elseif fzin==502
    TB=TBin;
% if the XT continues then change in joint angle value is added to
% the previous thumb base value
else
   if fzin==502
      TB=TBPrevious+tbs;
   else
      TB=TBPrevious;
   end
end


function [clockout,IB, fzout] ...
   = fcn(clockinPrevious, rawglovedata,ibs,IBin,...
       IBPrevious, fzin)


fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%index base
if clockinPrevious==clockout

   IB=IBPrevious;
 % if no manipulation is performed then no displacement
elseif fzin==520
   IB=0;
 %if manipulation type is PG,XT then index base is joint angle value is the displacement
elseif fzin==502||528
    IB=IBin;
% if the PG,XT  continues then change in joint angle value is added to
% the previous index base
else
   if fzin==502||528
      IB=IBPrevious+ibs;
   else
```

```
    IB=IBPrevious;
   end
end


function [clockout,IP, fzout] ...
   = fcn(clockinPrevious, rawglovedata,ips, IPin,...
       IPPrevious, fzin)



fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%index proximal
if clockinPrevious==clockout

   IP=IPPrevious;

elseif fzin==520
   IP=0;
 %if manipulation type is PG,YR,XT then index proximal is joint angle value is the
displacement
elseif fzin==504||502||528
    IP=IPin;
% if the XT,YR,PG continues then change in joint angle value is added to
% the previous index proximal
else
   if fzin==504||502||528
      IP=IPPrevious+ips;
   else
      IP=IPPrevious;
   end
end


function [clockout,MB, fzout] ...
   = fcn(clockinPrevious, rawglovedata,mbs,MBin,...
       MBPrevious, fzin)



fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%middle base
if clockinPrevious==clockout

   MB=MBPrevious;
```

```
 % if no manipulation is performed then no displacement
elseif fzin==520
   MB=0;
%if manipulation type is XT then middle base is joint angle value is the displacement
elseif fzin==502
    MB=MBin;
% if the XT continues then change in joint angle value is added to
% the previous middle base
else
   if fzin==502
      MB=MBPrevious+mbs;
   else
      MB=MBPrevious;
   end
end


function [clockout,MP, fzout] ...
   = fcn(clockinPrevious, rawglovedata,mps,MPin,...
       MPPrevious, fzin)


fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%middle proximal
if clockinPrevious==clockout

   MP=MPPrevious;
 % if no manipulation is performed then no displacement
elseif fzin==520
   MP=0;
 %if manipulation type is YR, ZR,XT then middle proximal is joint angle value is the
displacement
elseif fzin==504||508||502
    MP=MPin;
% if the YR,ZR or XT continues then change in joint angle value is added to
% the previous middle proximal
else
   if fzin==504||508||502
      MP=MPPrevious+mps;
   else
      MP=MPPrevious;
   end
end
```

```
function [clockout,P2, fzout] ...
    = fcn(clockinPrevious, rawglovedata,p2,P2in,...
         P2Previous, fzin)


fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%wrist flexion
if clockinPrevious==clockout

    P2=P2Previous;
 % if no manipulation is performed then no displacement
elseif fzin==520
    P2=0;
 %if manipulation type is ZT then wrist flexion is joint angle value is the displacement
elseif fzin==510
     P2=P2in;
% if the ZT continues then change in joint angle value is added to
% the previous
else
    if fzin==510
        P2=P2Previous+p2;
    else
        P2=P2Previous;
    end
end


function [clockout,P3, fzout] ...
    = fcn(clockinPrevious, rawglovedata,p3,P3in,...
         P3Previous, fzin)


fzout=fzin;
clockout=rawglovedata(1,1);
%if there is no change in clock, there will be no change in displacement of
%wrist arch
if clockinPrevious==clockout

    P3=P3Previous;
 % if no manipulation is performed then no displacement
elseif fzin==520
    P3=0;
 %if manipulation type is YT then wrist arch is joint angle value is the displacement
elseif fzin==506
     P3=P3in;
```

```
% if the YT continues then change in joint angle value is added to
% the previous wrist arch
else
   if fzin==506
      P3=P3Previous+p3;
   else
      P3=P3Previous;
   end
end
```

APPENDIX B

CODE FOR CALCULATING DOMINANT AXIS RECOGNITION AND ZERO ADJUSTMENT

Dominant axis recognized is the axis along which the translational or rotational displacement takes place i.e. displacement for every gesture is calculated. Once the displacement is calculated, zero adjustment is done.

```
function XRD = fcn(fz,tp,XRDPrevious)
%#codegen

%if manipulation type is XR then change in thumb proximal is the
%displacement.
if fz==500
   XRD=tp;
else
   XRD=XRDPrevious;
end

function XTD = fcn(fz,mp,XTDPrevious)
%#codegen
%if manipulation type is XT then change in middle proximal is the
%displacement.
if fz==502
   XTD=mp;

else
   XTD=XTDPrevious;
end

function YRD = fcn(fz,mp,YRDPrevious)
%#codegen
%if manipulation type is YR then change in middle proximal is the
%displacement.
if fz==504
   YRD= mp;

else
   YRD=YRDPrevious;
End
```

```
function YTD = fcn(fz,p3,YTDPrevious)
%#codegen

%if manipulation type is YT then change in wrist arch is the
%displacement.
if fz==506
   YTD=p3;
else
   YTD=YTDPrevious;
end

function ZRD = fcn(fz,mp,ZRDPrevious)
%#codegen

%if manipulation type is ZR then change in middle proximal is the
%displacement.
if fz==508
   ZRD=mp;
else
   ZRD=ZRDPrevious
end

function ZTD = fcn(fz,p2,ZTDPrevious)
%#codegen

%if manipulation type is ZT then change in wrist flexion is the
%displacement.
if fz==510
   ZTD=p2;
else
   ZTD=ZTDPrevious;
end

function PGD = fcn(fz,ip,PGDPrevious)
%#codegen

%if manipulation type is PG then change in index proximal is the
%displacement.
if fz==528
   PGD=ip;
else
   PGD=PGDPrevious;
end

function XTD = fcn(inp, threshold)
```

```
if ((inp>threshold) && (inp<=12.5))
   XTD=threshold-inp;
elseif inp>12.5
   XTD=(-1)*threshold;
elseif (inp<threshold) && (inp>=9.5)
   XTD=threshold-inp;
elseif inp<9.5
   XTD=threshold;
elseif inp==threshold
   XTD=0;
else
   XTD=threshold;
end

function YRD = fcn(inp, threshold)
if ((inp>threshold) && (inp<=130))
   YRD=threshold-inp;
elseif inp>130
   YRD=((-1)*threshold);
elseif (inp<threshold) && (inp>=90)
   YRD=threshold-inp;
elseif inp==threshold
   YRD=0;
else
   YRD=threshold;
end
function YTD = fcn(inp, threshold)

if ((inp>threshold) && (inp<11.2))
   YTD=threshold-inp;
elseif inp>11.2
   YTD=(-1)*threshold;
elseif (inp<threshold) && (inp>6.2)
   YTD=threshold-inp;
elseif inp==threshold;
   YTD=0;
else
   YTD=threshold;
end

function ZRD = fcn(inp, threshold)

if ((inp>threshold) && (inp<14))
   ZRD=threshold-inp;
elseif inp>14
   ZRD=(-1)*threshold;
```

```
elseif (inp<threshold) && (inp>7.0)
   ZRD=threshold-inp;
elseif inp==threshold
   ZRD=0;
else
   ZRD=threshold;
end

function ZTD = fcn(inp, threshold)

if ((inp>threshold) && (inp<17.6))

   ZTD=threshold-inp;
elseif inp>17.6
   ZTD=(-1)*threshold;
elseif (inp<threshold) && (inp>8.6)
   ZTD=threshold-inp;
elseif inp==threshold
   ZTD=0;
else
   ZTD=threshold;
end

function PGD = fcn(inp, threshold)

if ((inp>threshold) && (inp<9))

   PGD=threshold-inp;
elseif inp>9
   PGD=(-1)*threshold;
elseif  (inp>5) && (inp<threshold)
   PGD=threshold-inp;

elseif inp==threshold
   PGD=0;
else
   PGD=threshold;
End
```

APPENDIX C

CODE FOR CALCULATING OFFSET ERROR

When the manipulation gestures are performed one after another continuously, changes in the displacement of every rotational and translational movements are held at the same position unless the system is commanded by the user's hand.

```
function [clockout,XR,XT,YR,YT,ZR,ZT,PG] ...
  = fcn(clockinPrevious, rawglovedata,fz,...
     XRD,XTD,YRD,YTD,ZRD,ZTD,PGD,...
   XRPrevious, XTPrevious, YRPrevious, ...
   YTPrevious, ZRPrevious, ZTPrevious,PGPrevious)


fzout=fz;

clockout=rawglovedata(1,1);
% If clock hasn't changed then no change in the Task Position
if (clockout==clockinPrevious)
   XR=XRPrevious;
   XT=XTPrevious;
   YR=YRPrevious;
   YT=YTPrevious;
   ZR=ZRPrevious;
   ZT=ZTPrevious;
   PG=PGPrevious;

% if there is no manipulation performed no changes in the task Postion
elseif fz==520
      XR=XRPrevious;
      XT=XTPrevious;
      YR=YRPrevious;
      YT=YTPrevious;
      ZR=ZRPrevious;
      ZT=ZTPrevious;
      PG=PGPrevious;
```

```
else
   %If the manipulation type is XR then perform XR else no change in the
   %position
 if fz==500
      XR=XRD;
   else
      XR=XRPrevious;
   end
   %If the manipulation type is XT then perform XT else no change in the
   %position
   if fz==502
      XT=XTD;
   else
       XT=XTPrevious;
   end
   %If the manipulation type is YR then perform YR else no change in the
   %position
   if fz==504
      YR=YRD;
   else
    YR=YRPrevious;
   end
   %If the manipulation type is YT then perform YT else no change in the
   %position
   if fz==506
      YT=YTD;
   else
     YT=YTPrevious;
   end
   %If the manipulation type is ZR then perform ZR else no change in the
   %position
   if fz==508
      ZR=ZRD;
   else
    ZR=ZRPrevious;
   end
   %If the manipulation type is ZT then perform ZT else no change in the
   %position
   if fz==510
      ZT=ZTD;
```

```
    else
        ZT=ZTPrevious;
    end
    %If the manipulation type is PG then perform PG else no change in the
    %position
    if fz==528
            PG=PGD;
     else
     PG=PGPrevious;

    end
    %if the user's hand moves back to formation state from manipulation state or
    %if user's hand is in formation state or if user's hand is in standby
    %state then Robot hand goes back to standby position.
    if fz==519 || fz==514 || fz==516 || fz==512

    XR=0;
    XT=0;
    YR=0;
    YT=0;
    ZR=0;
    ZT=0;
    PG=0;


    end
end
```

REFERENCES

[1] N Y. Lii , Zhaopeng Chen, Maximo A. Roa, Annika Maier, Benedikt Pleintinger, and Christoph Borst, "A Task Space Framework for Gesture Commanded Telemanipulation," in *IEEE International Symposium on Robot and Human Interactive Communication*, pp. 3745-3752, 2012

[2] G. Gioioso, G. Salvietti, M. Malvezzi and D. Prattichizo, "Mapping of synergies from human to robotic hands with dissimilar kinematics: an object based approach," in *IEEE Int. Conf. on Robotics and Automation, Workshop on Manipulation Under Uncertainty*, 2011.

[3] S. Tosunoglu and D. W Repperger, "A Survey of Telesensation and Teleoperation Technology with Virtual Reality and Force Reflection Capabilities," in *Intl. Journal of Modeling and Simulation*, vol. 20, no. 1, pp. 79–88, 2000.

[4] J. Cui, S. Tosunoglu, R. Roberts, C. Moore, D.W. Repperger, "A Review of teleoperation system control", in *Proc. Florida Conference on Recent Advances in Robotics, FCRAR, Boca Raton, Florida,* May 8-9,2003.

[5] R. J. Anderson and W. Spong, "Bilateral Control of Teleoperation with Time Delay", in *IEEE Transaction Automation and Control*, Vol. 34, My 1989.

[6] W. Conklin, and S. Tosunoglu, " Conceptual Design of a Universal Bilateral Manual Controller," in in *Proc. Florida Conference on Recent Advances in Robotics, FCRAR, Boca Raton, Florida,* April 11-12,1996.

[7] G. Niemeyer, and J. E. Slotine, "Using Wave Variables For System Analysis and Robot Control," *IEEE International Conference Robotic & Automation*, Albuquerque, New Mexico, April, 1997.

[8] J. P. Urban, J. L. Buessler and J. Gresser, "Modular Neuro-Controllers for Reaching Movements," *Proceeding of SMC'98*, San Diego, California, October 1998.

[9] N. Xi, and T. J. Tarn, "Stability Analysis of Non-time reference Internet-Based telerobotics Systems", *Robotics and Autonomous System 32*, pp. 173-178, 2000.

[10] H. C. Cho, and J. H. Park, "Impedance Controller Design of Internet-Based Teleoperation Using Absolute Stability Concept," *International Conference on Intelligent Robots and Systems EPFL*, Lausanne, Switzerland, Oct. 2002.

[11] F. Cuesta, A. Ollero, B. C. Arrue, and R. Braunstingl, "Intelligent Control of Nonholonomic Mobile Robots with Fuzzy Perception," *Fuzzy Sets and Systems*, 134, 2003.

[12] C. Garcia, J. Posto, and C. Soria, "Supervisory Control for a Telerobotics System: a hybrid control approach" *Control Engineering Practice*, Vol. 11/7, pp. 805-817, January 2003.

[13] T. Yoshikawa, "Manipulability of robotic mechanisms," *in Intl. Journal of Robotics Research*, 1985.

[14] W. H. Zhu, and S. E. Salcudean, "Stability Guaranteed Teleoperation: An Adaptive Motion/Force Control Approach," *IEEE Transactions on Automatic Control*, pp. 1951–1969, 2000.

[15] M. Ciocarlie and P. Allen, "Hand posture subspaces for dexterous robotic grasping," *Int. J. Robotics Research*, vol. 28, no. 7, pp. 851-869, 2009.

[16] A. Peer, S. Einenkel, and M. Buss. "Multi-fingered telemanipulation - mapping of a human hand to a three finger gripper," *in Robot and Human Interactive Communication, RO-MAN, the 17th IEEE International Symposium*, pp. 465–470, August 2008.

[17] R. N Rohling and J. M Hollerbach. "Optimized fingertip mapping for teleoperation of dexterous robot hands," in *Proc. IEEE Intl. Conf. Robotics and Automation*, pp.3:769-775, Atlanta, May, 1993.

[18] M. Fischer, P. van der Smagt and, G. Hirzinger, "Learning techniques in a dataglove based telemanipulation system for the DLR hand," in *Proc. IEEE Int. Conf. on Robotics and Automation*, pp. 1603-1608, 1998,.

[19] S. Ekvall and D. Kragic. "Interactive grasp learning based on human demonstration," in *Proc. IEEE International Conference on, volume 4, pages 3519– 3524*, May 2004.

[20] W. B. Griffin, R. P. Findley, M. L. Turner, and M. R. Cutkosky, "Calibration and mapping of a human hand for dexterous manipulation," in *ASME International Mechanical Engineering Congress and Exposition, Symposium for Haptic Interfaces for Virtual Environments and Teleoperator Systems*, 2000.

[21] J. Liu and Y. Zhang. "Mapping human hand motion to dexterous robotic hand," in *IEEE Intl. Conf. on Robotics and Biomimetics (ROBIO)*, pages 829–834, December 2007.

[22] G. Hirzinger, K. Landzettel, D.Reintsema, C.Preusche, A. Albu Schaffer, B. Rebele and M. Turk, “ROKVISS – robotic component verification on ISS,” in *The 8th International Symposium on Artificial Intelligence, Robotics and Automaiton in Space – iSAIRAS,* 2005.

[23] I. M. Bullock and A. M. Dollar, “Classifying human manipulation behavior,” in *IEEE Int. Conf. on Rehabilitation Robotics,* pp. 532-537, 2001.

[24] S. Y. Yi, and M. J. Chung, “Robustness of Fuzzy Logic Control for an Uncertain Dynamic System,” *IEEE Trans. Fuzzy System 6*, 1998.

[25] W. Lin, C. Tsai, and J. Liu, “Robust Neuro- Fuzzy Control of Multivariable System By Tuning Consequent Mmembership Function,” *Fuzzy Sets and Systems*, 124, 2001.

[26] Ch. S. Kim, W. H. Seo, S. H. Han, Oussama Khatib, “ Fuzzy logic control of a robot manipulator based on visual servoing” *in Proc. IEEE Int. Conf. on Industrial Electronics*, vol. 3, pp. 1597-1602, 2001.

[27] Human Hand Anatomy. [Online]. Available: http://www.assh.org/Public/HandAnatomy/Pages/default.aspx Retrieved: November 2012.

[28] L. Zadeh, “Fuzzy Logic Systems: Origin, Concepts, and Trends," Available: http://wi-consortium.org/wicweb/pdf/Zadeh.pdf Retrieved: November 2012.

[29] Fuzzy Logic. [Online]. Available: Mathworks.com. Retrieved: November 2012.

[30] D. T. Pham and M. Castellani, “Action aggregation and defuzzification in mamdani-type fuzzy systems,” in *Proc. Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering and Science*, 2002.

[31] T. Yoshikawa, “Manipulability of robotic mechanisms,” *in Intl. Journal of Robotics Research*, 1985.