UNIVERSITY OF ÇUKUROVA

INSTITUTE OF NATURAL AND APPLIED SCIENCES

Ph.D THESIS

Cabbar Veysel BAYSAL

PRESHAPING CONTROL FOR DEXTEROUS MANIPULATION WITH A MULTIFINGERED ROBOT HAND

DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

2006 ,ADANA

ABSTRACT

Ph.D THESIS

PRESHAPING CONTROL FOR DEXTEROUS MANIPULATION WITH A MULTIFINGERED ROBOT HAND

Cabbar Veysel BAYSAL

UNIVERSITY OF ÇUKUROVA
INSTITUTE OF NATURAL AND APPLIED SCIENCES
DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

Supervisor: Prof. Dr. Süleyman GÜNGÖR

Co-Supervisor: Prof. Dr. Aydan ERKMEN

Year:2006, Pages: 162

Jury : Prof. Dr. Süleyman GÜNGÖR

Assoc. Prof. Dr. Mehmet TÜMAY

Assist. Prof. Dr. Hüseyin CANBOLAT

Assist. Prof. Dr. Ulus ÇEVİK

Assist. Prof. Dr. Turgay İBRİKÇİ

In a dextrous manipulation, one of the major problems that remains to be solved is the determination of an appropriate preshape for grasping a given object, under a manipulation task. In order to have a successful grasp execution, the system should have a successful grasp preparation at the preshaping phase. Grasp Performance Measure analyses are important for grasping with a robot hand as well as for the preshape.

In our work, we design and implement an "Intelligent Control System" for precision grasping beginning from hand preshaping , synthesizing from object properties, manipulation task characteristics, kinematic analysis and evaluation of contact wrench patterns via approximate reasoning with learning abilities , which is finally demonstrated in an antropomorphic robot hand simulation.

Keywords: Dextrous Manipulation, Robot Hand, Grasp Synthesis,Grasp Preshaping, Preshaping Controller.

ÖZ

DOKTORA TEZİ

ÇOK PARMAKLI ROBOT ELİN MAHARETLİ TUTUŞ BİÇİMİ KONTROLU

Cabbar Veysel BAYSAL

ÇUKUROVA ÜNİVERSİTESİ
FEN BİLİMLERİ ENSTİTÜSÜ
ELEKTRİK VE ELEKTRONİK MÜHENDİSLİĞİ ANABİLİM DALI

Danışman: Prof.Dr. Süleyman Güngör

2. Danışman: Prof. Dr. Aydan ERKMEN

Yıl:2006, Sayfa:162

Jüri: Prof. Dr. Süleyman GÜNGÖR

Doç. Dr. Mehmet TÜMAY

Yrd. Doç. Dr. Hüseyin CANBOLAT

Yrd. Doç. Dr. Ulus ÇEVİK

Yrd. Doç. Dr. Turgay İBRİKÇİ

Robot El'in maharetli tutuş işlemi konusunda devam eden çalışmalarda, çözüm bekleyen temel problemlerden birisi de tutulacak nesneye ve uygulanacak göreve uygun bir El ön biçimlenmesinin tespitidir.

Bu çalışmamızda, insan eli benzeri Robot El'in hassas tutuş eylemleri için, el ön biçimlenmesinden başlayarak, nesnelerin geometrik özelliklerinin, tutuş görevinin ve tutuş noktalarının kinematik analizlerinin, öğrenme yetenekli yaklaşık karar verme sistemi ile sentezlenmesi sonucunda , "Akıllı Denetim Sistemi" tasarlanmıştır. Bu sistem, 3B sanal ortamda, bir Robot El simulasyonu ile de test edilip, başarısı değerlendirilmiştir.

Anahtar Kelimeler: Robot El, Maharetli Tutuş, Tutuş Ön Hazırlığı, Tutuş Sentezleme , Akıllı Denetim Sistemi, Önbiçimlenme Denetleyici.

ACKNOWLEDGEMENTS

I would like to express my gratefull respects to everyone who helped or supported me during this thesis.

I would like to especially thank to Prof. Dr. Süleyman Güngör, for his guidance of my Ph.D study. His patience and support were very precious for me throughout the study.

I would like to especially thank to Prof .Dr. Aydan Erkmen , co-supervisor from METU, for her critics and guidance for my Ph.D work. Her acute insight into robotics and intelligent control , enabled me to develop this thesis. Her criticism about quality of the study , encouraged me to reach a better performance .

I would like to express my respects to my Qualifiying and Thesis Jury Members so that they spent their valuable time criticising me, for further development.

I would like to thank to Department Faculty of EE in CU, to all, for encouraging me , helping me, and giving me a friendly atmosphere .

I would like to devote my work to my family who show patiance during my study and spending their time for waiting . They all sacrificed from their holidays, vacations.

I would like to thank to anyone who taught me even a sentence, throughout my life, my parents, all my instructors, my friends and my competitors.

Thank You, all...

TABLE OF CONTENTS **PAGE**

LIST OF FIGURES **PAGES**

LIST OF TABLES **PAGES**

1. INTRODUCTION

1.1.Motivation and Goals

One of the prominent issues in robotics is the dextrous manipulation of robot hands. Due to many aspects , such as desire to improve flexibility of the current generation industrial robotic systems, desire to improve prosthetic devices and applications in humanoids.

Dextrous Manipulation means handling objects skillfully according to desired tasks. A Dextrous Robot Hand is the one with multiple articulated fingers, each of which is an open kinematic chain distributed around a possibly palmar surface. It contains a large number of degrees of freedom, and can achieve fine resolution with high accuracy and at high speeds. Each of the fingers should be eqipped with position and force sensors and may have additionally tactile sensors. Dextrous Robot Hands poseses a number of advantages over conventional grippers, by providing versatility for fine motions, and hence are less error prone. They also allow control of gripping forces which provide more efficient grasps. This control ranges from securely grasping an object in order to prevent slipping to compliantly grasping a delicate object so that it is not damaged. In addition, a Multifingered Dextrous Robot Hand can grasp unusually shaped objects which a parallel gripper cannot.

Dextreous manipulation, which is only a subfield of robotics, is a broad area in itself, comprising of many topics such as gripper design, grasping, control, tactile sensing, and task planning.

In dextrous manipulation, one of the major problems that remains to be solved is the determination of an appropriate preshape for grasping given an object, under a predefined manipulation task. In order to have a successful grasp execution, the system should have a successful grasp preparation at preshaping phase. Hence, the grasp preparation or preshaping is one of the critical issues for manipulation with a dextrous robot hand.

In this thesis work , our objective is to achieve preshaping control to have a successful grasp execution in dextrous manipulation, integrating preshaping and graps initialization phases .In order to reach our objective we state following goals:

- task analysis and decomposition into primitives,
- partitioning task space by relative approach planes
- combination of object geometry and task primitives
- estimation of patterns representing kinematic effects of grasp initialization
- development of grasp quality measures
- approximate reasoning used for pattern matching of task and grasp
- integration of developed methods into an Intelligent controller architecture.

To reach the goals stated, we describe our methodology in the next section.

1.2.Methodology

The Dexterous Manipulation (DM) has consisted of phases such as grasp planning , approach , grasping and manipulation . In the literature, there are works about DM that have been done in individual areas of the phases of dexterous manipulation which we will discuss in chapter 2. These works are grouped into two major approaches: physiological (knowledge based) and mechanical (analytical) grasp approaches. However, there are relatively few works dealing with integration of these grasp phases . Moreover, many existing approaches are not complimentary such as to form a universal integrated solution to the dexterous manipulation problem.

In this thesis work , we consider a combinational point of view. From the physiological to the mechanical aspects, considering the problem of preshaping and grasp as a sole continium. As a result, in order to have a successful dexterous manipulation, grasp planning, approach and grasping phases should all be analyzed and handled together. This perspective is somewhat similar to physiological grasp situations observed in human infants and primates .

Oztop and Arbib (2004) , presents a an analysis of such a physiological model for Infant Grasp Learning.

Christel and Fragazsy (2000) analyzes the grasping behaviour of Capuchin Monkeys. Both works show that preparation of hand posture before grasp is influenced by task and object.

Adapting this point of view, our work develops Preshaping Control System which includes grasp planning coming from task requirements, approaching and grasping which is related to object, its pose and environment .

Although in literature there are works for preshaping such as Cutkosky (1986), Lyons (1985), which uses human hand based taxonomies , preshaping is not integrated to the DM phases. We foresee that preshaping control must be planned in a virtual situation , before any actual movement. In this perspective, the task is analyzed and and decomposed into primitives of axial translational and axial rotational displacements in 3D environment, as done in most of the previous works, such as Michelman (1998).

Object pose, shape , together with environment constraints are also analyzed to select approaching and grasp characteristics. We include Virtual Grasp Kinematics into preshaping control so that at the planning phase, the controller able to deduce what kind of outcome manipulations, the selected preshape can lead to, well before execution of the actual grasp and manipulation. Our controller takes inputs from the virtual environment and yields a resultant grasp preshape, after handling, decomposed task attributes, and selecting a virtual approach to the predefined object by an analysis of possible virtual grasps with kinematic estimations and grasp quality measures. The controller use the all estimated attributes as inputs, some of which are in forms of Fuzzy Sets. The kinematic estimations are based on virtual grasp contacts .

To conclude, we design an implement an Intelligent Controller architecture based on Incremental Learning of task and objects which will estimate the preshape that will contribute the execution of the task.

1.3.Contribution of the Thesis

The main contribution of this thesis work are twofold.

- We develop a model of preshaping for grasping, bridging the two fields of preshaping and that of grasping. A Preshaping Control is generated, yielding the appropriate forces and torques for an optimal initialization of the grasping task. Our Preshape Controller involves grasp planning and grasp synthesis using the attributes coming from each grasp phases in order to yield a preshape that finalizes into a successful grasp. Towards this aim, the controller integrates information from task, object and the grasp kinematics for grasp planning and preshape selection .
- Preshaping Control aiming at initializing the grasp in an optimal way for beginining the manipulation task requested , necessitates an appropriate contact model representing the initial force and torque generated upon landing on an object with a given preshape. We use screw systems in our contact models for post contact kinematic considerations at the initial phase of manipulation . Screw systems in the literature are used for generating grasp quality measures and were not at all related to preshaping to achieving a high grasp quality to better initiate manipulation. This is the major novelty of our approach in this thesis work.

1.4.Thesis Outline

After introducing dexterous manipulation, in chapter 2 we continue with the survey of the previous works at this area. Although there exists numerous theoretical and applied works, we restate here basis forming works.

Chapter 3 presents our Controller Model for the Grasp Synthesis for Preshaping phase, with a given task . In this part, we define our framework and assumptions, model of our controller and introduce its components.

Chapter 4 presents the 3D computer implementation for our model developed in chapter 3. We run several demonstrative examples in the same chapter.

Chapter 5 presents the implementation results and discussions about the results .

Chapter 6 disscuses the performance analysis of the controller in terms of parametric sensitivity analyses.

Finally , Chapter 7 provides summary of our approach and concludes with further possible research works.

2.SURVEY OF PREVIOUS WORKS

Since the last two decades, there has been a large amount of research in dextrous manipulation. Basically, there are four main areas: grasping, dextrous manipulation, high-level task planning, and design of hands and tools for manipulation. The main issue has been how to control mechanical hands so that they can perform manipulation tasks with the same dexterity and sensitivity as the human hands. Hence, there are two main approaches to the problems of grasping which are those motivated by the study of human hands and those motivated by the physical and mechanical properties of grasps. We start with further descriptions for robotic grasp and later state these two approaches.

2.1.Descriptions for Robot Hand Grasping

The system in which the desired object is gripped by the fingers of a multi-fingered hand is called a grasp. Grasps can be categorized into three general groups: precision grasps, power grasps and partial grasps.

Precision grasps grip an object with fingertips. The way a pen is held for writing is one example of a precision grasp. Fine motions can be imparted on the object simply by moving the fingers. Although this type of grasp has a high degree of manipulability, it does not have a large capability to resist loads and disturbances.

The grasp of a hammer or wrench is an example of a power grasp. The fingers enclose an object, there are multiple contacts on each finger and palm contact is usually involved. There is no ability to manipulate the object within the hand but the ability to resist forces on the object is greatly enhanced.

Partial grasps do not totally constrain the movement of the grasped object. They are used to perform tasks on objects whose motion is limited in other ways. One example of a partial grasp is the hook grasp with arched fingers used to open a drawer with recessed handles.

It would seem that the type of grasp is aligned with the task involved – precision grasps for manipulation and power grasps for resisting forces. There are situations, however, where force must be applied to the object but physical obstacles prevent a power grasp.

The process of obtaining a grasp on an object can be separated into four phases:

- Object detection – determining the type and position of the target object.
- Grasp synthesis – determining the appropriate hand position and configuration for type of object and task.
- Approach – positioning the hand near the object.
- Grasp execution – obtaining the actual grasp of the object.

In the literature, The grasping issues are outlined .In a broad sense, the problems involved , described by Scwartz (1986), include following issues: Control Issues, Geometric Issues, Integration Issues, Task Planning Issues, Static issues .

2.2.Main Approaches For Robot Hand Grasping

In attempting to solve problems of the grasping issues , researchers have taken two basic approaches; one is a physiological approach motivated by the study of human hands and the other is motivated by the physical and mechanical properties of hands and objects. The human hand is a versatile end effector which can perform intricate tasks. Studying human grasps can lead to a grasp taxonomy which can assist robot hands in choosing an appropriate grasp to perform a specific task, and can also be used to design efficient grippers.

Another approach involves studying the physical and mechanical properties of grasps and understanding how the actions of the robot affect the motions of the object.

It should be noted that these two approaches are not competing, they complement each other. Whereas the mechanical approach seems more suitable for low level tasks (homogeneous manipulations), the physiological approach is likely to be more suitable for high level tasks.

We start to present here the first approach of which deals with studies on human grasping and grasp-taxonomies, and then the second part which deals with physical and mechanical issues such as contact types, number of fingers required to achieve grasps, equilibrium, stability, and compliance.

2.3.Physiological Approach (Grasp Taxonomies)

In an effort to simplify the choice of grasp a robot has to make, numerous attempts have been made to categorize grips by studying grasps of the human hand. The human hand is capable of a wide range of grasps, and the grasp selection depends on the size and shape of the object as well as the task to be performed. Studying how these factors influence the grip choice leads to a taxonomy which provides a systematic way to choose an appropriate grasp for a particular set of task requirements and object characteristics.

Human grasps were first categorized in medical literature and are summarized in Taylor and Schwartz (1955). This classification is based on the shape of the object rather than the task to be performed. The six types of grasps are the cylindrical grasp, the tip grasp, the hook grasp, the palmar grasp, the spherical grasp and the lateral grasp.

Napier (1956) argues that when one grasps an object for whatever purpose, the major concern is the stability of the grasp. He therefore bases his classification on how stability is achieved, which leads to two major types of grips:

Power Grip: The object is held in a clamp formed by the partly flexed fingers and the palm, while counter pressure is applied by the thumb which lies more or less in the plane of the palm.

Precision Grip: The object is pinched between the flexor aspects of the fingers and the opposing thumb.

The factors which influence the posture of the hand are the shape of object and the size of the object . Another important influence is the intended activity. Finally, the pattern of the chosen grip is based upon the nature of the task.

Cutkosky and Wright (1986) approached the problem in order to design efficient robot grippers for a manufacturing environment. They carried out a series of protocol analysis experiments with machinists and classified these and other machining grips. These have led to the taxonomy of human grasps which they organized as a hierarchical tree. They extended Napier's basic categories by considering the task and geometric considerations that influence the pattern of the grip. The taxonomy presented in figure 2.1

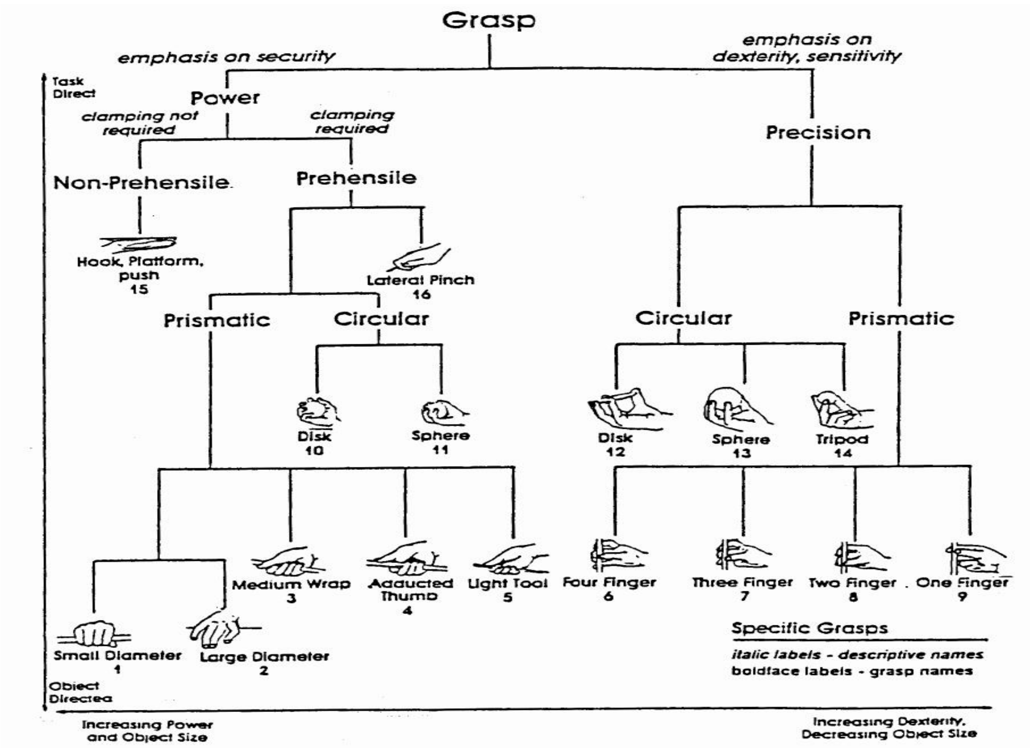


Figure 2.1. Cutkosky's Taxonomy (Cutkosky ,1986)

Moving from left to right along the tree, the grasps become less powerful yet more dextrous and the grasped objects become smaller. The wrap grips are the most powerful yet least dextrous and therefore require that all manipulation must be done with the wrist. By contrast, the precision grips are much more delicate. Moving from top to bottom, the choices range from task-directed grasps to object-directed grasps.

Iberall's (1987) classification is based on the fact that in practice, the human hand uses a combination of grasps. Iberall introduces the virtual finger concept.

The concept of virtual fingers provides independence from physical hand structure and particular object models. Figure 2.2 illustrates the virtual fingers.

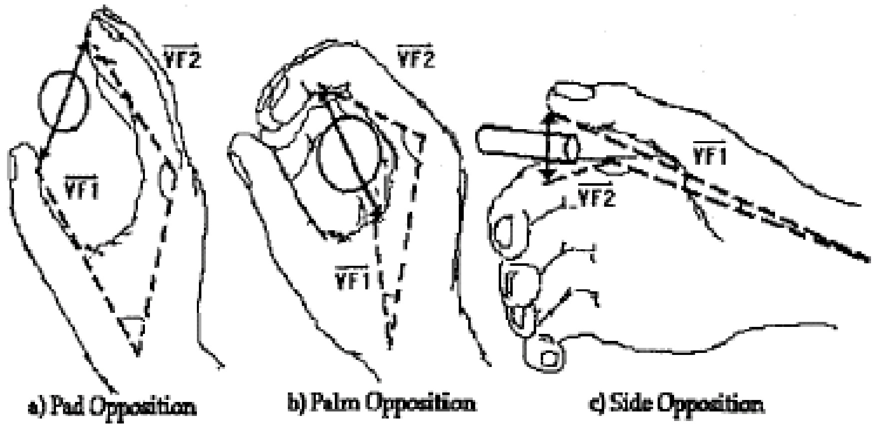


Figure 2.2. Iberall’s Virtual Fingers (Iberall ,1987)

According to Iberall, there are three basic methods the hand can employ to apply opposing forces around an object for a given task, each of which use two VF's (virtual fingers).

1. Pad Opposition is between the thumb pad (VF1) and the finger pads (VF2). This grip provides great flexibility in finely controlled motions, but little stability.

2. Palm Opposition is between the palm (VF1) and the digits (VF2). It sacrifices flexibility in favor of stability.

3. Side Opposition is either between the thumb pad (VF1) and the side of the index finger (VF2) or between the sides of the fingers. This grip is a compromise between flexibility and stability.

Iberall also concentrates on the planning phase of the grasping task and includes a definition of the task to be performed. Figure 2.3 illustrates the grasp planning methodology. The task is defined in a high-level language structure and passed to the planner, where it is broken down into task phases. Hand postures for each phase are selected using information from the object database, task knowledge base, and grasp knowledge base. The object database is constructed of geometric representations of specific objects (i.e., a screwdriver or hammer). The task knowledge base consists of information concerning the use of an object for a specific task (i.e., picking up or turning a screwdriver). The grasp knowledge base is the set of grasp configurations used to preshape the hand.

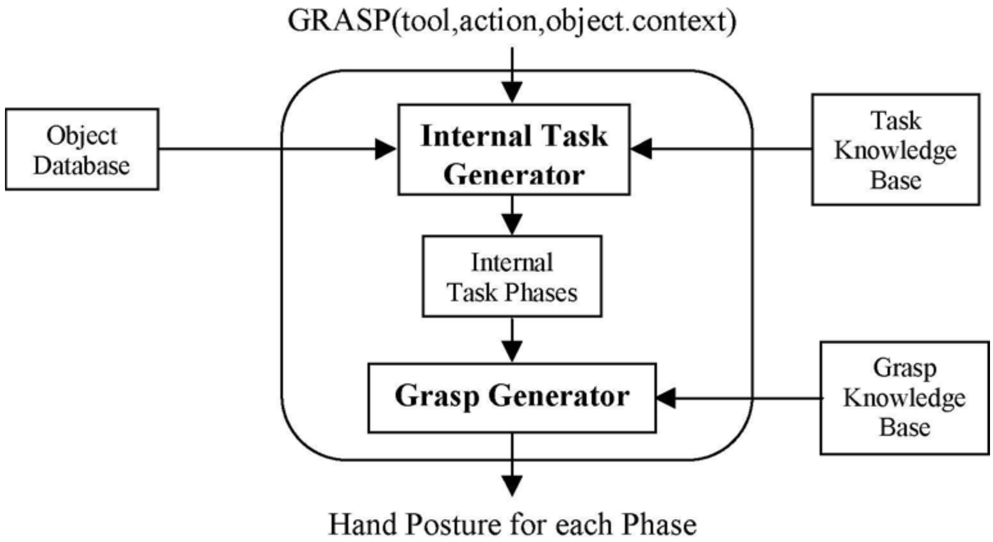


Figure 2.3. Iberall' s Grasp Planning Method (Iberall ,1987)

Lyons (1985) used the concept of virtual fingers to classify grips as part of a computational model which he built. He classifies grasps into three major categories and calls this classification SSG, a Simple Set of Grasps. As in Iberall's classification, there is a power grasp, a precision grasp, and a grasp which provides some power and some precision.

1. Encompass grasp. This is a power grasp; it provides no manipulative ability but secure affixment. The hand completely envelopes the object. Manipulation can only be accomplished through the arm and wrist and this is used for basic `pick-and-place' tasks.

2. Lateral grasp. This is basically the pinch grasp. The flat surfaces of the fingers are used to grip the object. It provides some manipulative ability and some stability.

3. Precision grasp. The object is held between fingertips. It allows arbitrary motion but provides little stability. Each of these grasps must be particularized to suit the size and shape of the target object.

Lyons divides grasping into three phases:

1. Preshape: This is where the hand is configured in preparation to grip the object while it is reaching for the object.

2. Acquisition or Gripping : As the fingers move to contact the object, they have low stiffness, and once the contacts are in place, the contact forces required to secure the object are calculated.

There are three cases. Simple Acquisition: The hand moves toward the object in the preshape configuration and grasps it. Two-phase Acquisition: The object is acquired in a temporary grasp and is then manipulated to the final configuration. Constrained Acquisition: It is impossible for the hand to approach the object with the chosen grasp and preshape. Therefore either the grasp or preshape must be modified.

3. Manipulation: A table is used to determine how to move the object based on the degrees of freedom left in the hand. This ranges from moving only the wrist to the hand being able to impart arbitrary forces and moments to the object.

Liu (1989) defines task requirements by the qualitative weighting of four task attributes: stability, manipulability, torquability (tangential rotability), and radial rotability. The weightings of the attributes guide the selection and sequence of control heuristics that align the hand with the object and execute the grasp.

In general, the basic elements of taxonomy based grasping methodologies can be categorized as follows:

- Reduce possible hand configurations to a set of standard grasps.
- Develop a database of target objects.
- Align and Preshape the hand by selecting a standard grasp using information about the task and object.
- Grasp the object using automated local control schemes and sensory feedback. The complexity of selecting a hand configuration is avoided by defining a limited set of grasps.

We can study how humans perform tasks in particular domains or see if there are better ways to perform grasps, and design hands to perform those domain specific tasks efficiently, or we can use these taxonomies and try to better understand how to particularize a grasp in order to perform a specific task and how to achieve these precise and stable grasps.

2.4.Mechanical and Physical Approach

A grasp is generally defined as a system in which an object is gripped by the fingers of a robotic or human hand. The configurations of the hand, the object and the contacts on the object have an effect on the properties of the grasp. A clear understanding of the important attributes of a grasp is necessary in order to interpret grasp analysis and synthesis. Murray (1994) gives a detailed analysis of the kinematics of a precision grasp. This section summarizes the analysis and identifies significant grasp attributes. The following assumptions are made in this analysis:

- The object and finger links are rigid bodies.
- The contacts are fixed. No rolling or slip occurs at the contact points.
- Accurate models of the object and the hand exist.

The analysis first examines the properties of the contact interface between the fingers and the object. The effects on the object of forces at the contact points are then described. Finally, the kinematic constraints of the grasp are derived.

2.4.1.Grasp Kinematics

A hand may be modeled as a collection of manipulators representing each finger of the hand. A two-finger example from Murray (1994) , which illustrates the significant reference frames of the hand, contacts, and grasped object, is shown in Figure 2.4 .

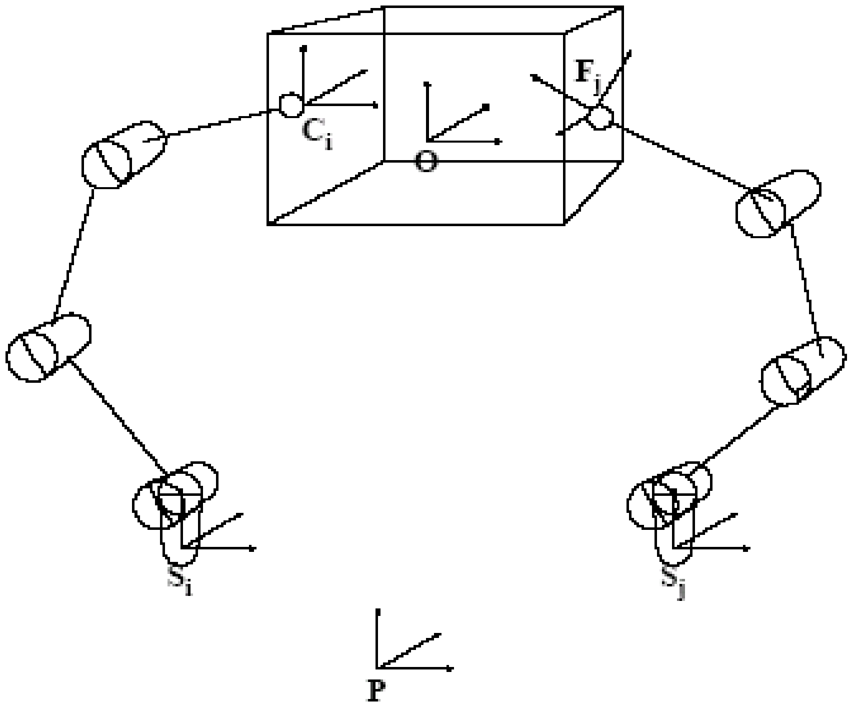


Figure 2.4. Murray's Coordinate Frames (Murray, 1994)

Each finger is assigned a base frame, $\mathbf{S_i}$ and a fingertip frame, $\mathbf{F_i}$ .The palm frame, **P**, is affixed in a convenient location to the hand structure that supports the fingers, usually the attachment point to the wrist. The object frame, **O**, and the contact frame, $\mathbf{C_i}$ are also indicated. Examining the grasping constraints for each finger identifies the relationship between finger motion and movement of the grasped object. The grasping constraints for a finger are the directions, relative to the contact frame, in which motion is not allowed. Motion is constrained in the directions in which forces can be exerted, as defined by the contact model.

In developing the kinematics of the grasp problem , first, a method is required to calculate the fingertip forces from a desired force/torque wrench on the object. The basis for this calculation is the Grasp Jacobian relationship.

The Grasp Jacobian (or grasp map), $\mathbf{G}$, can be obtained by resolving each fingertip force to a common coordinate frame embedded in the object. For each fingertip i, this force resolution results in the mapping matrix

$$\mathbf{G_i} \ . \quad \mathbf{f}_{\mathbf{obj}} = \mathbf{G.f_{tip}} \tag{2.4.1}$$

$$\mathbf{f_{obji}} = \mathbf{G_i.f_{tipi}} \tag{2.4.2}$$

In above equations , the force vectors $\mathbf{f}$ are generalized vectors: they may include both forces and torques. The individual mapping matrices $\mathbf{G_i}$ are concatenated to form the grasp map $\mathbf{G}$, and the fingertip force vectors are also grouped into one vector. The details of subject can be found in Kerr and Roth ‘s (1986) work.

The Grasp Jacobian developed above allows us to calculate the required contact forces from the desired force on the object. In order to produce these forces at the fingertips, we now develop a Hand Jacobian, which will allow us to calculate the joint torques from the contact forces . The Hand Jacobian, $\mathbf{J_h}$, is based on the standard Jacobian, which relates end effeector forces to individual joint torques for a robotic manipulator. The Hand Jacobian describes the motion of all contact frames in relation to the finger joint velocities.

$$\tau_{\mathbf{i}} = \mathbf{J^T{}_i.f_{tipi}} \tag{2.4.3}$$

$$\tau = \mathbf{J^T{}_h.f_c} \tag{2.4.4}$$

$$\mathbf{f}_{\mathbf{obj}} = \mathbf{G.f_c} \tag{2.4.5}$$

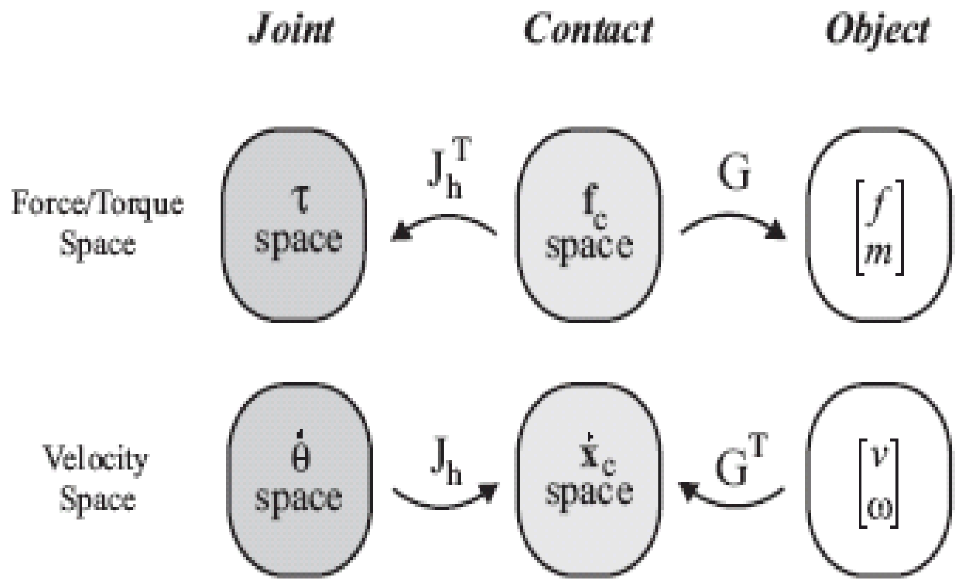


Figure 2.5. Grasp and Hand Jacobians (Okamura ,2000)

2.4.2.Contacts

In order to consider the interaction between the fingertips and the object being grasped, it is important to study the mechanics of constraint and freedom that take effect when two bodies come into contact. The contact causes forces and moments to be transmitted to the objects, and consequently, the relative motion between the two bodies is determined by the geometric and friction properties of the contact. The contact model defines the connectivity at the contact point between a finger and the grasped object. Three generally accepted models are frictionless point contact, point contact with friction (hard finger), and soft finger contact. point contact model allows the finger to exert a force on the object only in the direction of the object surface normal at the contact point. The hard finger model adds tangential friction force components at the contact point, limited by a friction coefficient, μ .

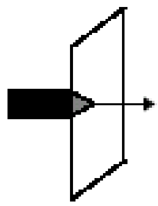
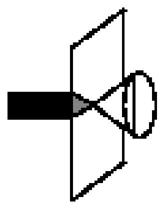
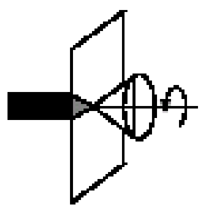
Contact Type	Picture	Wrench Basis	FC
Frictionless point contact		$\begin{bmatrix}0\\0\\1\\0\\0\\0\end{bmatrix}$	$f_1 \geq 0$
Point contact with friction		$\begin{bmatrix}1&0&0\\0&1&0\\0&0&1\\0&0&0\\0&0&0\\0&0&0\end{bmatrix}$	$\sqrt{f_1^2+f_2^2} \leq \mu f_3$ $f_3 \geq 0$
Soft finger		$\begin{bmatrix}1&0&0&0\\0&1&0&0\\0&0&1&0\\0&0&0&0\\0&0&0&0\\0&0&0&1\end{bmatrix}$	$\sqrt{f_1^2+f_2^2} \leq \mu f_3$ $f_3 \geq 0$ $\lvert f_4 \rvert \leq \gamma f_3$

Figure 2.6. Contact Types (Murray, 1994) .

The set of possible contact forces is defined by a friction cone about the object surface normal. The soft finger model adds the ability to generate a torque about the surface normal, limited by a torsional friction coefficient, γ . Each of these models can be described by where $\boldsymbol{F}$ is the wrench imposed on the object at the point of contact, $\boldsymbol{B_{ci}}$ is the wrench basis that defines the connectivity at the contact and $\boldsymbol{FC_{ci}}$ defines the constraints on the contact forces. Figure 2.6 illustrates each contact model and provides the definitions of the wrench bases and force constraints. The contact reference frame is chosen such that the z-axis is collinear with the inward surface normal at the point of contact.

Salisbury's (1982) analysis is based on screw theory which uses six parameters, so there are six degrees of freedom (relative motion), and hence six classes of contacts (excluding the trivial case of no contact). A surface can experience three types of contact: point contact, line contact, and plane contact. By studying how the motion of a body is constrained by the various types of contact and considering the effect of friction, he derives the contact types shown in the table below.

Table 2.1 Salisbury's Contact Types

CONTACT TYPE	REMAINING DEGREES OF FREEDOM
No Contact 6	6
Frictionless Point Contact	5
Frictionless Line Contact	4
Friction Point Contact	3
Frictionless Plane Contact	3
Soft Finger	2
Friction Line Contact	1
Planar Contact with friction	0

The constraints imposed by each type of contact are described in terms of the degrees-of-freedom allowed and in terms of screw systems of motion and force associated with each contact type. This formulation is neat and makes the calculations simple, however it is not always accurate for certain kinds of contact. For instance, it does not take into account surface deformation under load, which, in addition to friction and surface geometry, also effects the strength and stability of a grip.

Cutkosky (1989) defines a different set of models. Rather than studying just contacts between rigid bodies, he considers different types of fingertips and defines a set of models that may be used to characterize the fingertip geometry: point contacts , hard curved contacts, flat contacts , soft curved contacts , and very soft contacts.

Human fingertips exhibit rolling and deformation by soft and curved fingertips. Salisbury provides a simple classification of contact types, while Cutkosky does a more in depth study of contacts and attempts more realistic models of contacts. However, we shall see in the coming sections, that most analyses consider the point-contact with friction, since it is simplest to deal with mathematically, although quite a few consider the soft finger model which models a human finger.

2.4.3.Grip Map

The effect of the wrenches at each contact on the object is determined by transforming the wrenches to the object frame. The object frame is typically located at the mass center of the object, O, as shown in Figure 4. The position and orientation of each contact frame relative to the object frame. The wrench imparted on the grasped object from a single contact point is an object wrench. The contact wrench basis, $\boldsymbol{B_{ci}}$, defines the directions in the contact frame in which forces can be exerted for a particular type of contact. The resultant contact map, $\boldsymbol{G_i}$,describes the wrench imposed on the grasped object due to a wrench at the i^{th} contact. This contact map is linear. Therefore, the total object wrench is determined by superimposing the object wrenches that result from each contact wrench.

The matrix $\mathbf{G}$ is the grip map that defines the resultant object wrench due to the wrenches applied at the contact points for an n-fingered grasp. Note that only the contact model and contact locations on the object are factors in determining the grip map. The configuration of the hand is not addressed and does not even need to be defined.

Salisbury identifies the external and internal forces acting on the object in a grip matrix which is used to determine how to apply forces and control the motion and stiffness of the object. He shows that when a grasp totally constrains an object there are contact and frictional forces it can apply to the object while maintaining its equilibrium, and he then shows how to identify these forces.

In the course of performing a task with a grasped object, the object will be subjected to external forces. In order to measure these forces, as well as the internal forces acting on the body, Salisbury defines the Grip Transform, $\mathbf{G}$, which like the Jacobian $\mathbf{J}$, is used to map forces between finger joints and grasped object coordinates. $\mathbf{G}$ is a square matrix whose dimension and rank is the number of contact wrenches acting on the object, and so it is constant for a particular grasp on an object.

The grip transform maps a set of forces and moments acting on the fingers, to external and internal forces acting on the fingers. Similarly, given a desired force we wish to exert on the object, **G** produces the forces and moments needed for each finger. Additionally, **G** maps the finger velocities to the body's velocities, where **V** is a vector of twist intensities per unit time acting along the wrench axes at each contact. **G** provides a basis for applying forces and controlling small motions with an articulated hand.

2.5.Grasp Properties

A definition of the properties that constitute a “good grasp” must be decided upon before a grasp may be synthesized. Shimoga (1996) surveyed current grasp synthesis algorithms available in current literature and categorized the properties that were utilized into five basic types:

- Force closure – the ability of a grasp to resist an arbitrary external force
- Equilibrium – the ability to maintain a grasp on an object
- Stability – the ability to resist external disturbances
- Dexterity – the ability to impart motion to the grasped object
- Dynamic response – the compliance of the grasp

Force closure, equilibrium and stability are required elements of a successful grasp.

The dexterity and dynamic response properties provide a measure of the grasp's capability to perform certain tasks. Dynamic response as a subset of stability and the elastic properties of the grasp must meet some predetermined goals as well as reject load disturbances.

Force Closure :

Force closure, the ability to resist arbitrary external loads, is one essential property that must be achieved in a successful grasp. Shimoga provides a summary of current literature related to achieving force closure in multi-fingered grasps. The majority of works listed, however, are restricted to the investigation of planar grasps. Most of the works that do examine spatial grasps are limited to, at most, three fingers.

Efficient algorithms for constructing force closure grasps on polygons and polyhedra were developed in Nguyen (1987). The complete set of force closure grasps on a set of faces in 3D was found by solving a set of linear inequalities constructed from the wrenches produced at each contact. Independent regions of contact were constructed for each finger where force closure would be satisfied. He characterized the force closure grasps of 3D, polyhedral objects for hard finger contacts. The necessary linear conditions for three and four-finger force closure grasps were formulated and implemented as a set of linear inequalities in the contact positions. Finding the set of all force closure grasps is then cast as a problem in projecting a polytope onto a linear subspace.

Both of the previous analyses were limited to polyhedral objects and concentrated on producing independent regions of contact for each finger. No method was presented to compare individual grasps within the set of force closure solutions. A more general approach for determining force closure is presented in Ferrari (1992). No restrictions are placed on the object model and hard-finger contacts are assumed. Primitive wrenches on the object due to forces at the contacts are calculated for a particular grasp. Force closure is satisfied if the convex hull of the primitive wrenches encloses the origin of wrench space. A quality measure that gives an indication of the "efficiency" of the grasp is defined by the distance from the wrench space origin to the nearest facet of the convex hull. This grasp efficiency measure furnishes a means for comparing force closure grasps.

Equilibrium:

Force closure only guarantees that the grasp can resist any arbitrary external load, or, conversely, that the motion of the object is fully constrained. An equilibrium analysis verifies that the hand is capable of generating the necessary forces to constrain the object for a given external load. A grasp satisfies equilibrium if the following constraints are met

- force equilibrium in the object space,
- friction and positivity constraints in contact space,
- joint torque limits in joint space.

In other words, the hand must be able to generate the joint torques that result in the positive contact forces that balance the external load without exceeding friction limits at the contacts. Finding the contact forces that satisfy all the constraints is the principal objective of equilibrium analysis. This problem is made more difficult since grasps that satisfy force closure are underconstrained. The contact force solution is the sum of the forces that balance the external load and the internal grasping forces for which there is a multiplicity of solutions. The optimum contact forces are those in which the internal grasping forces, and the demands on the hand, are minimized. The force components at each contact force are represented by individual wrenches on the object. The equilibrium equations are represented as the linear sum of the individual wrenches and augmented by adding internal grasping forces between pairs of fingers. The contact forces can then be calculated through a simple matrix inversion. A computationally efficient algorithm for obtaining the contact forces without matrix inversion is also presented. This method would fit very well with the force closure technique presented in Ferrari , since both are based on the primitive contact wrenches. However, the definition of the internal grasping forces does not generalize to five hard-finger contacts.

According to Shimoga, Kerr (1987) formulates the solution for the contact forces as an optimization problem. The linearized friction constraints and joint torque constraints are combined into a compact inequality relation that is used to form the objective function of the optimization problem. The optimum contact forces balance the external loads while minimizing the objective function. An algorithm for constructing and solving the linear programming problem is presented in Shimoga (1996). This method assumes the grasp is force closure and an equilibrium solution exists since the method will otherwise fail to produce any results.

Stability:

The definition of grasp stability takes on two different meanings in current literature. The categories for stability analyses as follows:

- 1st order stability –will the object slip from the grasp
- 2nd order stability – rejection of load disturbances which could be

classified as stability for general and compliant grasps.

First order, or kinematic, stability is similar to equilibrium analysis. The aim of this analysis, however, is to define the set of external loads which meet the contact and joint torque constraints rather than verifying that a given load set meets the constraints. A 1st order stability analysis does not consider the elastic and dynamic behavior of the grasp. Accordint ot Shimoga, Jameson introduces a method for predicting the 1st order stability of grasps. The external load is represented as the product of a unit wrench and a wrench intensity. All possible motions of the object with respect to the hand are examined to determine the set of wrench intensities for which the grasp is stable. The method is limited to the examination of contact constraints and assumes the external load is produced by a pure force on the object. The latter assumption decreases the complexity of the analysis by reducing the region of stable external wrenches to a line in $\mathbf{R}^6$ wrench space.

Hanafusa and Asada (1977) introduced the concept of stable prehension, similar to the one discussed earlier. They did their analysis in two dimensions using a robot hand with elastic fingers and rollers on the fingertips to eliminate friction.

Nuyen (1987) introduces an algorithm for constructing 2nd order stable grasps. The compliance of the grasp is modeled by virtual springs at the contacts. The grasp is stable if the resulting grasp stiffness matrix is positive definite. Algorithms for achieving a desired grasp stiffness matrix are presented. The number and type of contacts are also considered. Nguyen also proves that all 3D force closure grasps can be made stable, assuming the hand control system includes a form of compliance control. This is of considerable importance for grasp synthesis since the synthesis algorithm does not need to include stability analysis if the selected grasp configuration satisfies force closure.

Nguyen extended this work in finding stable grasps in the plane and in three dimensions. His criteria for stability are that the object be in static equilibrium and the grasp be force closure. For the 2D case, he considered objects that were planar polygons, multiple point contacts without friction, and the fingers were modeled as virtual springs with controllable stiffness.

Dexterity :

Dexterity criteria provide a means to compare the suitability of different grasps. The definition of dexterity is expanded here to include not just manipulability but ability to achieve a certain condition. In other words, dexterity measures encompass both manipulation and force-related criteria.

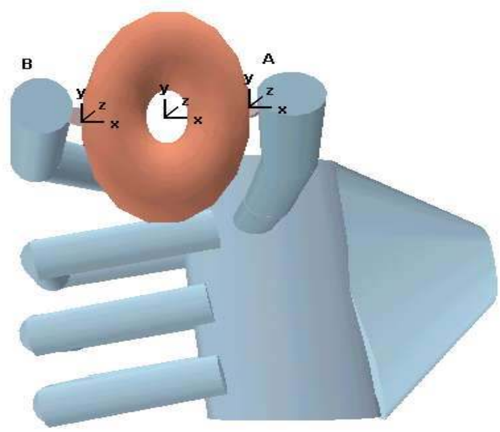
Most dexterity measures involve the grasp matrix $\mathbf{G_h}$. Shimoga (1996) presents a well-rounded list of dexterity measures. Nearly all were developed for redundant manipulators, but their extension to multi-fingered hands is discussed. Criteria reviewed in the survey include:

- Determinant of $\mathbf{G_h}$
- Condition number of $\mathbf{G_h}$
- Smallest Singular Value of $\mathbf{G_h}$
- Joint angle deviation

Those criteria are reasonable selections for comparing grasps in a general sense, but it is difficult to relate them to specific motion or force related task requirements. For example, maximizing the determinant of $\mathbf{G_h}$ physically implies that the maximum object velocity vector is obtained for a unit velocity vector of all finger joints. The question that must still be answered is does the object velocity vector lie in a suitable direction.

Hsu ,Li and Sastry (1988) present a method for grasp synthesis that utilizes a detailed description of the task requirements. The load and manipulation requirements for the task are defined as ellipsoids in the object twist space and wrench space. The approach to grasp synthesis methodology is based on specifying the grip map **G** and hand Jacobian $\mathbf{J_h}$ so that some performance measure is maximized. The task ellipsoids, grip map and hand Jacobian are used to develop structured quality measures in the twist space and wrench space.

2.6.Grasp Wrench Space



$$W_A = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad W_B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}$$

Figure 2.7. Example Grasp with Contact Wrenches

One main topic in this field is the static analysis of forces and torques that can be applied to an object through grasp contacts. The evaluation of the efficiency of a grasp to counteract disturbances such that the object will be firmly fixed in the gripper is the next important research topic.

A basic quality criterion for a grasp is the force closure property, first proposed for grasping applications . Efficient tests for this property have been developed using different models, like the grasp matrix or the grasp wrench spaces . Also, many approaches deal with the construction of a force closure grasp.

The force closure property, however, is only a minimal quality requirement for a grasp. It deals with how efficiently a grasp can compensate for arbitrary disturbances or balance a special set of disturbances that is expected when executing a desired task. To quantify this efficiency of a grasp, the concept of wrench spaces can be used. The set of all wrenches that can be applied to the object through the grasp contacts is called the *Grasp Wrench Space (GWS)*. A commonly used efficient way to approximate the GWS is to calculate the convex hull over the discretized friction cones . The problem with all these approaches is the discretization of the friction cones, where significant errors may be introduced when approximating the cone with only a few vectors to achieve fast computation. Moreover, Mishra et al (1987) showed that there are problems to be expected with this method for large friction coefficients. This GWS approximation corresponds to the idea that the sum of all applied forces is constrained to one, which has only a weak physical interpretation for multifingered grippers. Ferrari and Canny (1992) proposed a method for calculating a physically well interpretable GWS approximation, where the forces applied in the contact sum up to the number of contacts, by calculating the convex hull over the Minkowski sum of the friction cones. The drawback with directly calculating this GWS approximation is that it is computationally expensive. To rate the quality of a grasp, task directed and task independent measures were introduced. They also use the largest wrench sphere that just fits within the GWS as a task independent quality measure of the grasp. The measure is not scale invariant and depends on the selection of the torque origin .

To achieve invariance to the selection of the reference point, Li and Sastry propose to use the volume of an ellipsoid generated by the grasp matrix as a measure of grasp quality. They also suggest to model the task wrench space as a six dimensional ellipsoid and fit it in the GWS. The problem with this approach is how to model the task ellipsoid for a given task, which they state to be quite complicated. Pollard (1994) introduces the Object Wrench Space (OWS) which incorporates the object geometry into the grasp evaluation. With her approach, however, this OWS represents the best grasp that can be achieved for an object and gives no direct measure for an arbitrary grasp. To rate the quality of a grasp, the largest inscribed sphere is used. To achieve scale invariance, the torque component of the wrenches is scaled with the length of the longest object axis. It is also proposed to evaluate grasps using disturbance forces in order to overcome the problem of torque origin selection and to take the geometry of the object into account.

The quality evaluation method is highly reliable, but the complexity of this approach is very high while the geometric information has to be evaluated for each grasp candidate .These are the major drawbacks of the proposed grasp measures, so far.

2.7.Control Structures for Dexterous Manipulation

In the literature, the control of dexterous manipulation is be decomposed into three main levels , as shown in Table 2.2.

Table 2.2 Levels of Control for Dexterous Manipulation (Okamura,2000)

High Level	Task planning, discrete event systems, grasp choice
Mid Level	Phases, transitions, event detection
Low Level	Operational space dynemics, cooperative object impedance control , kinematics, forces.

High-level control includes task and motion planning and grasp choice. Most of these issues have been discussed in issues previous sections. Grupen (1998), in his works, introduced DEDS, which implements high level control tasks. In his work, basically , controller is a discrete event controller with predefined states.

Mid-level control includes manipulation phases, for example whether the fingers are operating independently or cooperatively, and whether force or motion control is required. Transitions between these phases must be accomplished smoothly, and events must be sensed to trigger the transitions.

The mid level of control involves the management of the various phases of a task, as well as the specification of control laws to be used when transitioning between phases. Hyde (1995) presents an Object Oriented Framework for Event-Driven Dextrous Manipulation.

Low-level control includes basic strategies (e.g. force or impedance control) . It also includes the formulations of the control problem in the appropriate space (e.g. the operational space of the grasped object). One that stands out is the operational space formulation, which has been used in object-centered dexterous manipulation . The basic idea of this approach is to control motions and contact forces through the use of control forces that act at the operational point of a grasped object. Through joint space/operational space relationships formed using Jacobians and a Lagrange or Newton-Euler formulation, an operational space dynamic model of the system is created.

2.8.Summary of Previous Works

Much of the research has in literature provided useful tools that may bring us closer to more versatile hands. Salisbury and Cutkosky studied contact models, and how they affect the freedom of the object. According to Mishra and Silver (1989), Sharir, and Schwartz as well as Markenscoff, Ni, and Papadimitriou calculated how many fingers are needed to obtain grasps on certain objects. Salisbury introduced the notion of the Grip Transform. He used it to neatly describe the forces present in a grip and what forces can be applied to the grip without disturbing the equilibrium. Hanafusa and Asada introduced the notion of stability, but only determined stable grasps for a very simplified model. The analysis works for the case of no friction, but has limited usefulness since friction is an important consideration. Nguyen addressed the stiffness of objects grasped by virtual springs. He added the stiffness consideration to Hanafusa and Asada's work, and also extended it to three dimensions. However, in trying to generalize the calculations, they became too complex, so he only dealt with limited simplified cases. Cutkosky obtained other measures for grasps in addition to stability. He showed how to measure the stability, stiffness, and resistance to slippage, which are other important considerations for choosing a grasp. For humans, grasping an object seems like a simple task, yet analyzing the details involved shows us that grasping is very complex and there are many issues to be considered. The primary focus until now has been establishing statically stable grips, such as was done by Hanafusa and Asada and also Nguyen, although researchers are now dealing with finding grasps suitable to the performance of tasks. What emerges from all this is the importance of finding a stable grasp. A lot of work has been put into this problem, but most of the researchers seem to come up with the same mathematical calculations (much of which were omitted in the descriptions). The main idea seemed to be to obtain a stable grasp by identifying force relationships, programming the stiffness of the fingers, and reducing the degrees of freedom of the object. Jacobians were obtained and calculations were done using standard linear algebra. However, all these calculations included constraints which were added to obtain results, yet it remains unclear how to determine some of these constraints and why they were chosen.

The most unsatisfactory situation in the whole area concerns the problem of synthesizing a grasp. While the physiological approach has succeeded in a limited way to address these issues, there is little or no progress by the researchers using analytical techniques. The work of Mishra and Schwartz does provide an algorithm to synthesize a positive grip, but this is only worked out in the case of an idealized hand, and no effort has been made to come up with a grip that satisfy additional criteria of ``goodness." Techniques to synthesize grasps and to integrate these grasps into homogeneous manipulations must be developed, and this is where ideas from both approaches can be used together. Although many of the problems involved in grasping are far from being solved, some progress has been made: problems have been identified and tools have been established.

In studying the physiological aspects of grasping, researchers developed numerous grasping taxonomies. These taxonomies have been used to develop robot grippers and to assist robot hands in grasping certain objects. Since human hands are efficient and multi-purpose grippers, studying how humans grasp objects can hopefully explain the science of grasping. While some systems have been built which use taxonomies to choose an appropriate grip, this does not always work due to unusually shaped objects and additional unknown factors. Therefore, the physical and mechanical properties must be studied.

The literature is more extensive in the areas of identifying contact types and their effects and restrictions on grips. Salisbury's grip transform is an important tool for describing force relationships between the fingers and the grasped object. The main results are the identification of certain properties of grips, such as the potential energy, stability, and compliance, and the establishment of the relationships between forces and movements of the hand and the object. The physiological and mechanical approaches are complementary. The physical and mechanical results are needed to understand the low level elements involved in grasping. This includes contact types, effects of friction, compliance, slipping, and what is involved in obtaining a stable grasp. However, for higher level elements such as task planning, studying how human hands grasp objects and perform tasks is more useful. For the purpose of performing a task, it might not be sufficient to have a stable grasp. Often, there are many different possible stable grasps, and so we will have to choose which one, if any, is appropriate for the task to be performed.

In the literature, majority of mechanical approach works focused on low level analysis of grasp control. The works based on physiological methods focused on grasp synthesis more. The synthesis methods combined successive phases of grasp for control, such as preshaping , approach and final closure, as in Lyons' work and in Iberall's work. However, these methods did not combined mechanical properties of grasp into planning.

There has been little success in evaluating the suitability of a grasp to perform a particular task in terms of its mechanical properties, and therefore, these grasp taxonomies (and expert systems) can better assist the dextrous hand in choosing a grasp. Moreover, the controller design for grasp synthesis is still an active research point.

To conclude, grasp synthesis methods using heuristics to combine mechanical properties of grasp into synthesis phase are required to bridge gap between these two approaches. Using this perspective, the grasp synthesis should start from preshaping phase and hence control should also include preshaping phase. The combination of the mechanical and physiological approach can be better performed at preshaping phase in which determination of the appropriate hand position and configuration for type of object and task occurs.

Therefore, for a successful final grasp, a good estimation of preshape using mechanical properties becomes a critical point. A control strategy using such approach would perform better because it integrates two control perspectives that of grasp preparation and that of landing on an object to initiate grasp . The control of grasp using that integrated approach therefore will start with the control of preshaping towards a goal initialization of actual grasp after contacting the object. Therefore, in our work, we develop a integrated approach by designing a predictive learning controller that assures a good initialization of the subsequent phase of grasping.using this approach.

3. STRUCTURE OF GRASP PRESHAPE CONTROLLER

3.1.Implementation of the Intelligent Controller

In order to reach our preshape determination goal, we need to design a controller structure whose properties are partially explained at section 1.2 . The desired controller for our goal should be able perform grasp planning based on combination of grasp phases and it should also be implemented as the integration of task requirements, object properties and grasp kinematics. In next sections , we will individually analyze task requirements, object properties and grasp kinematic features. Preshaping Control System should include grasp planning coming from task requirements, approaching and grasping which is related to object, its pose and environment . It should also be perform kinematic analysis of Virtual Grasp which is to be included into control so that at planning phase controller would be able to deduce what kind of final manipulations can be performed with the proposed grasp preshape before execution of the actual grasp.

In order to design such a high level controller, we need to go over available control structures. As it can be inferred from the requirements and nature of the problem, the classical controllers would not fit to our case. The desired controller must be adaptive and should have learning capacity from experiments to expand its functionality incrementally. It should also be adapt itself to optimize its performance. After all above, we conclude that we need an intelligent control architecture to reach our goal.

Before going into design stages, we would like to review available intelligent control architectures. In literature review , we observe that the intelligent control structures, in general , implemented by using Artificial Neural Networks, Fuzzy Logic and Control Methods Evolutionary Computational Methods and different combinations of these techniques.

In these approaches, ANN are used for their capabilities of adaptation, learning and approximation. Fuzzy Logic methods are used for approximate reasoning. Evolutionary Computational methods are used for global optimization as derivative free optimization.

Jacobsen (1998) presents a classficiation scheme for intelligent systems based fuzzy and neural methods. Figure 3.1 illustrates his broad classificiation. His classification is also expanded by Abraham (2000), in his review of intelligent systems. Abraham (2000) provides a broad review of all mentioned methods, as well as different combinations of the methods. Abraham (2001) also presented a review of Neuro-Fuzzy Systems.

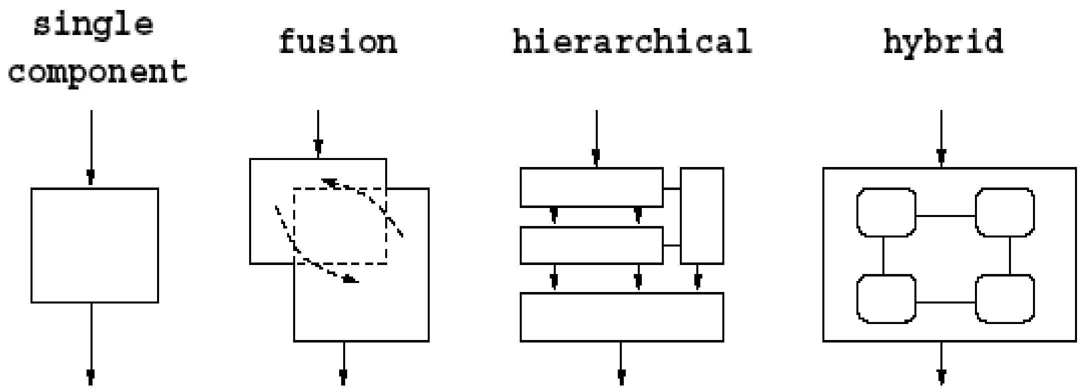


Figure 3.1. Different Categories of Intelligent Systems (Jacobsen, 1998)

According to the classification by Jacobsen,

- **The Single Component System** class contains systems based solely on one technique. It contains the "puristic" approaches, such as plain fuzzy control, TSK-control or Multi-Layered Perceptron based approaches. These systems realize a mapping from an input space to an output space.
- **The Fusion Based System** class includes systems which combine different tecniques into one single computational model. Neuro-Fuzzy Systems such as ANFIS , NEFCON are well knwn examples of this class. These systems includes multi-stage evaluation scheme and do not perform strategic planning or self assesment.

- **The Hierarchical System** class comprises more architecturally complex systems. Its instances are build in a hierarchical fashion, associating a different functionality with each layer. The correct functioning of the system depends on the correct operation of all layers. Error from a low layer is propagated up thru the hierarchy effecting the system performance.
- **The Hybrid System** class contains approaches that include different techniques side by side basis and focus on their interaction in the problem solving task. With this interaction, it becomes possible to integrate alternative techniques and exploit their mutuality. By using this class, conceptual view of the agent allows us to abstract from individual techniques and focus on global system behavior, as well as study of the individual contribution of ecah component. This class provides greater potential for solving tasks including learning ,classification and control. ARIC and GARIC are well known examples of this class.

After review of the classification, we observe that a hybrid system architecture is the class that would best fit to our requirements for intelligent control architecture, due to nature of our problem and requirements for integration of a multi-stage grasp action and integration of different sub-systems. In our case, we would like to integrate task decomposition ,object modeling and grasp kinematics and we would like to use approximate reasoning.

In nature these subjects to integrate are different individual items that can not be combined into a single computational model. Therefore, integration can be done by interaction of different methods in a common frame. In order to have an adaptive system , we could not use a hierarchical structure. As a result, we could implement our architecture in an agent based hybrid system. The oveview of agent architecture structure is given in figure 3.2 .

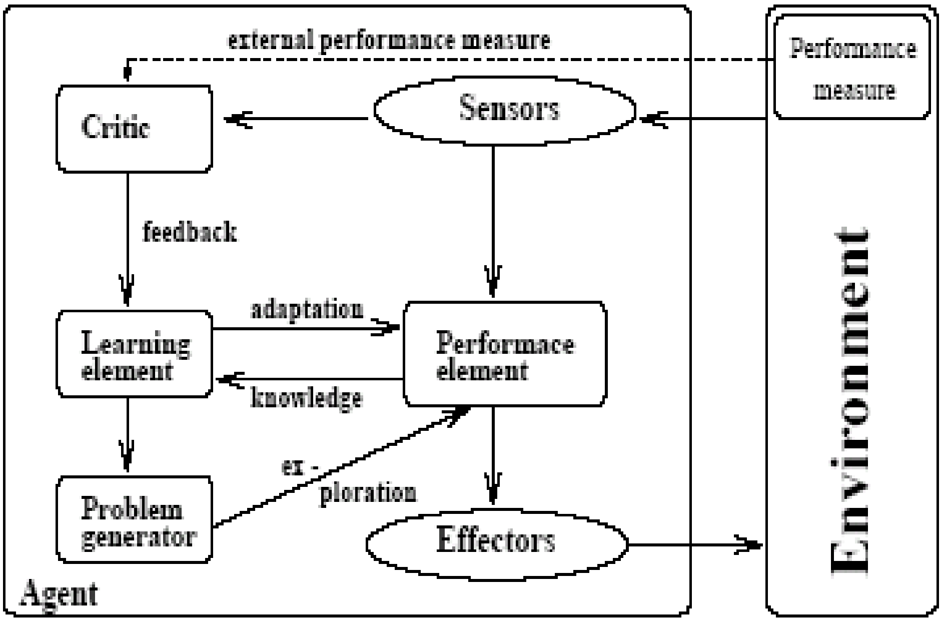


Figure 3.2. Overview of Agent Architecture (Russel & Norvig ,1995)

We would like to give a brief descriptions of components of the agent.

- **The Environment** constitutes the problem task and it is described by a state vector percieved by the agent thru the sensors and influenced by it thru its effectors.
- **The Performance Element (PE)** is the actual control unit mapping environment states to actions or results.
- **The Learning Element (LE)** updates the knowledge represented in the PE in order to optimize the agent's performance with respect to a performance measure used by internal or external critic . It has access to the environment state.
- **The Critic** faces the problem of transforming an external reinforcement signal into an internal one. It generates the reinforcements as rewards or penalties. Internal reinforcement is immediate and more informative. It indicates for each action taken whether it was beneficial or not.

- **The Problem Generator** has a role to contribute to the exploration of the problem space. It proposes different actions or results which might lead to the discovery of new and better solutions.

In general agent strcuture based on reinforcement learning technique. In order to decide about suitability of this technique, we would like to review Learning Methods very briefly. In machine Learning, there are three basic categories of learning methods. Supervised Learning , Unsupervised Learning and Reinforcement Learning. Supervised Learning method uses the training data that includes not only the inputs but also the corresponding output values . In Reinforcement Learning, each decision made by the system affects the subsequent decisions. After each decision, environment provides a real valued reward. The agent's goal is to choose seqeunces of actions to maximize its long term reward. This method is different from Supervised Learning where each decision is completely independent of others. A comprehensive survey of Reinforcement is given by Kaelbling and Littman (1996). In Unsupervised Learning, a collection of objects are given to the system and it is asked to construct a model to explain the observed properties of the objects. No teacher provides output or rewards.

Another categorization of Learning systems is given by Jouffe (1998) in his work which combines reinforcement learning and fuzzy inference systems as FACL and FQL . According to Jouffe, Learning Methods are characterized by the information source used for learning and can be classified with respect to the degree of information of the source. Hence, four families exist.

- **Direct Teacher** , which provides, at each time step, the correct control action to be applied by the learner. Typically, these learning methods are based on an input-output set of training data, on which we have to minimize errors between the teacher's actions and the learner's.
- **Distal Teacher** , which does not give the correct actions, but the desired effects on the process. As the error term is not directly exploitable by the learner, an inverse model is classically used.

- **Performance Measure** , which indicates the quality of a learner on a set of states. This kind of learning method is generally associated with Evolutionary Methods and Learning Classifier Systems.
- **Critic** , which gives rewards and punishments with respect to the statesreached by the learner, but does not provide correct actions nor trajectories.

In addition to above categorization, Dorigo and Bersini (1994) gives a comparison of Q-Learning and Classifier systems.

Our Controller Architecture:

After finishing the review of intelligent control systems, we would like to explain our controller architecture which we have designed. In our architecture , we used basically an agent structure but there are minor diversifications from the standart agent architecture. Our model is compatible with agent structure in aspect of existence of individual units such as performance element, learning element , internal crictic unit, environment sensing element and problem generator and is ocmpatible in their interaction. In figure 3.3, overall block representation of our controller is given.

However, we have an External Critic unit as a meta control and our end effector is not inside the modular architecture, it is considered as an environment model. There are also differences in the implementation of the individual modules in our structure. That is to say, we used existing methodologies to build up the modules our own controller architecture.

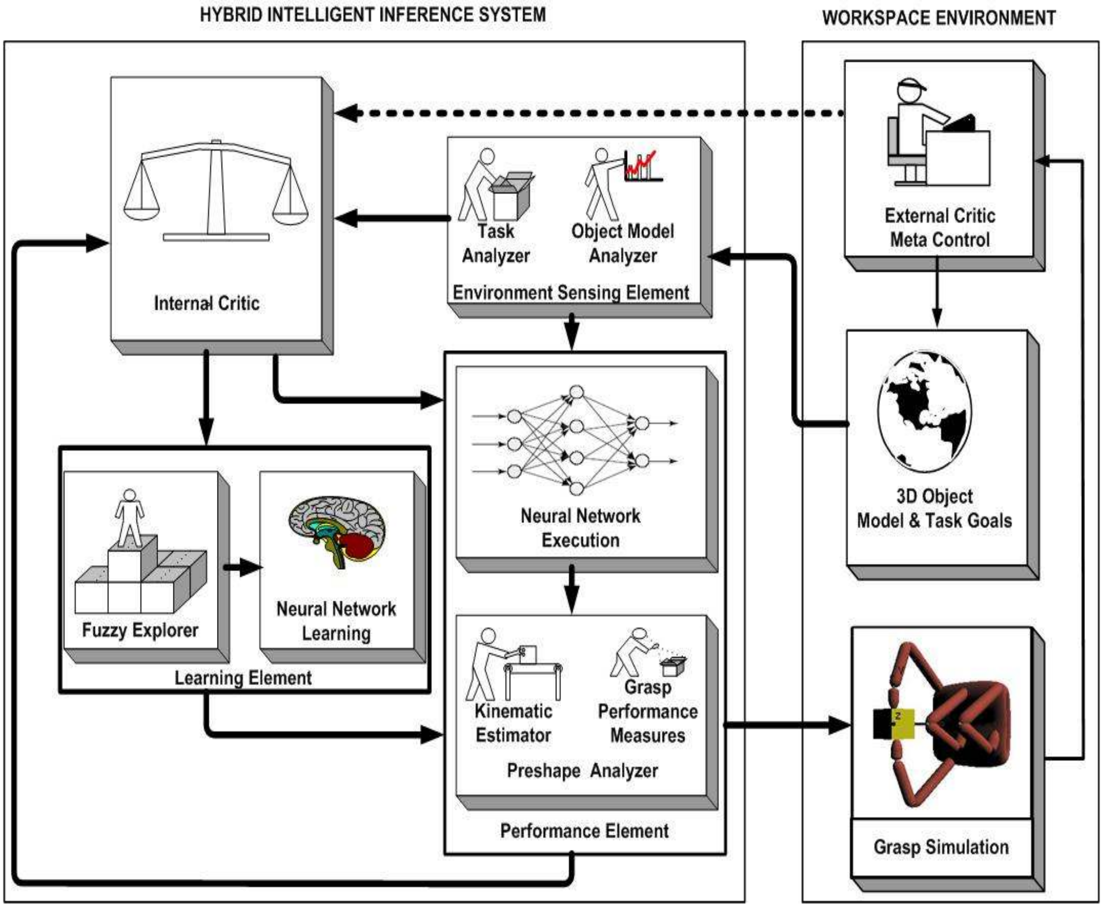


Figure 3.3. Overview of the Intelligent Grasp Preshape Controller Architecture .

In our architecture, we have environment items which are External Critic as a Meta Control , Objects to be grasped as represented 3D models and Task Actions to be performed as grasp, Grasp Simulation to indicate result of the controller to Extermal Critic. The External Critic is the human operator in our case. Grasp simulation is the end effector. The External Critic adjusts the parameters of the Internal Critic , especially Threshold values, and it inputs the Task actions via user interface of the Controller and also selects the object 3D model via user interface which will be explained in Computer Implementation chapter.

We would like explain the internal units of the Hybrid Intelligent Controller Architecture. In our controller, the internal units are as follows with their functionality.

- **Environment Sensing Element:** This unit gets the command from External Critic for the task and object model input and prepares it for the other parts of the controller. Object model is transferred to memory from a disc file for further processing, with describing feature set attribute estimation, such as dimensions, number of triangles etc. This is just an interface unit. Task is transferred to wrench structure.
- **Performance Element:** As its formal definition states this is the actual controller who takes inputs from environment sensing element in terms of task attributes and object attributes. It is composed of two successive parts: ANN as executor which was described in section 3.10 and Preshape Analyzer which estimates the virtual grasp and contact kinematic estimations on object model according to the outputs of ANN given in terms of approach plane and approach value. It performs contact determination using Ray Plane Intersection explained before. The Preshape Analyzer also performs opposition space check and estimations of grasp performance measures , Stability, Manipulability and Reciprocal Twist Space Rank etc..Finally , Preshape analyzer estimates a resultant wrench represting the virtual grasp for the given Approach Plane and Approach value, with grasp measures. The details for Preshape Analyzer were given in previous sections from 3.3 to 3.8.
- **Internal Critic :** In our implementation, Internal Critic has a role to decide about performance of the execution of the Intelligent Controller rather than deciding for the performance of LE as stated in its general definition. We would like to use the this term for our case because it decides about execution performance of the Intelligent Controller via via reinforcement, indicating success as reward and failure as penalty. This results in determination of the execution mode of the Controller. It uses approximate reasoning for critism. It uses fuzzy similarities to decide about mathing degree of task and resultant grasp preshape, using the various thresholds. Indeed some of the execution modes involves activation of Learning element which we explain in next paragraphs, Internal Critic also contributes to learning feature of the Intelligent Controller, as Incremental Learning. The Internal Critic takes

inputs from Performance element, Environment Sensing element and inputs from External Critic. It send its output to Performance Element, Learning Element as reinforcement and to Meta Control as report. The Performance Element and Learning Element depending on the Controller Execution mode, gets reinforcement from Internal Critic and using the Multistate Decision algorithm for execution both synchronously, they either decide to run at a mode or to stop. The Multistate Decision algorithm is explained at Execution Part for the Intelligent Controller, in next sections.

- **Learning Element:** Learning Element is composed of Fuzzy Explore to find a proper coverage of workspace and IO training of ANN in supervies mode. The input to Learning Element is from Internal Critic, environment Sensins Element . Output from LE is sent to PE and Internal Critic and External Critic as report. In Execution modes 0 and 1 LE stands and waits. When, Execution Mode 3 is decided, First Fuzzy Explorer with the given Object and Task is run as Beam Search and the Best Possible Grasp is found corresponding to Task and Object. This information is added to IO training for ANN and IO training which is performed in Supervised Learning. It recurses the PE after IO training, to test its Learning Performance by Internal Critic . Hence, Learning Performance also tested by reinforcement.

Learning of a new Objects occur as Incremental Learning. Initially , ANN is IO trained with only a basic object. Training data is initially composed of only for basic object. In our implementation we used a simple cube as basic trainimng object. The other objects are introduced by Meta Control to the system one by one with some possible tasks from our task space representation. With this structure Learning of the Overall Controller is an Incremental Learning Method which is also a hybrid combination of Supervised and Reinforcement Learning.

Execution of the Intelligent Controller:

Execution of the Intelligent Controller is a Multistate Decision Control.

Algortihms for executions are given in Computer Implementation Chapter 4. The Intelligent Controller is a Four State controller. States are successive.

The Four states for Intelligent Controller are as follows:

Execution Mode 0: This is the normal execution of PE with Internal Critic using Execution Threshold. The Controller start for cold startup at this mode, when PE is reinforced as failure by Internal Critic, it ends with transition to next mode , Execution Mode 1. In case of success reinforcement from Internal Critic, the Controller send its output to Grasp Simulation , it stops and waits next command from Meta Control. If deadlock is reset by External Critic, execution again starts at this mode. LE is at stop in this mode .

Execution Mode 1: This is the execution of PE with Internal Critic using Execution Threshold, but it includes a local search within estimated Approach Plane in terms of Approach Value. When the PE is reinforced as fail by Internal Critic, it ends with next mode , Execution Mode 2. In case of success reinforcement from Internal Critic, the Controller send its output to Grasp Simulation , it stops and waits next command from Meta Control. LE is at stop in this mode .

Execution Mode 2: This is the execution of PE with Internal critic, but using Recognition Threshold to test acknowledge of the object by the Controller itself . When the PE is reinforced as failure by Internal Critic, it ends with next mode , Execution Mode 3. In case of success reinforcement from Internal Critic, the Controller stops and reports to Meta Control recogniton of object and also asks for Task Correction. It waits next command from Meta Control. Grasp simulation is not performed at this mode. LE is at stop in this mode .

Execution Mode 3: This is the execution of PE and LE with Internal Critic using Execution Threshold . This mode is training new object to the Controller. After Object is reconized as new Object then, LE uses its Fuzzy Explorer to search all planes and approaches to find best possible task as grasp wrench pattern which is most similar to the desired task by Meta Control. Object attributes and Best Possible Task is used in IO training of ANN , in Supervised Learning . The new IO set is added to training data files and IO training for ANN is performed. In order to test LE learning performance, the Controller executes itself at mode 0 and if not succesful exectues at mode 1. After this if it fails controller reports deadlock and stops.

To summarize the execution of Multistate Decision Controller, we briefly restate it. The Controller starts to execute at initial state and performs execution at this state and sends its output first to internal critic, if critic send reinforcement as success, then controller PE sends its output to Grasp simulation for execution in normal execution modes. Meta Control gets report from PE and Internal Critic and the current state is completed, then Intelligent Controller waits its new execution command from Meta Control. In case Internal Critic indicates a reinforcement as failure, The Intelligent Controller advances its execution state from current to next higher. In case of failures , state advancements repeated. If in all 4 stage, failure occurs, The controller goes to total failure state (deadlock) and stops , by reporting deadlock to External Critic. The total failure state can only be reset to initial state State transitions are performed according to reinforcements send by Internal Critic, as success or failures.

3.2. Object Modelling

In robotic manipulations, the task applications require a representation for environment and object ,by means of visual feedback. For Dexterous Manipulation ,there are numerous vision controlled approaches. Namiki (2003), recently, presents a fast robot hand grasping using a vision based control.

Thus, grasp synthesis, necessitates object representation or a model of object , due to nature of the vision-servoed manipulation problem . In the early works, object representation was done via visual information about object that was acquired from 2D images by using image processing and low level feature extraction techniques. These techniques were useful upto a certain extent. However, interpretation of extracted features requires extensive analyses because the features are primitive attributes, such as edge corner ,roundness etc and interpretation is higher level. Combination of these low level features via symbolic processing generally leads to higher level perception. However, it may still be very difficult to infer for these fused features ,the object and task.

In vision systems , there are other approaches yet, one of which is obtaining and analyzing the 3D model of the object. The acqusition of a 3D model is based on a variety of methods. Among these, the projective geometry and multiple 2D images are used to construct a 3D model , whereas in some other approaches, this construction is also performed together with a 2D low level processing.

Stenstrom and Connoly (1992), have implemented a method constructing object models from multiple images by low level image processing. In their work, range images and camera images are combined using image processing algorithms to find wireframe representation of object model.

Reed and Allen (1999), Stamos and Allen (2002), use and extend this approach for recovery of large scale scenes and objects, again using the combination of range and camera images. Figure 3.4 indicates the algorithm used in their work. In figure 3.5, one of their results is presented.

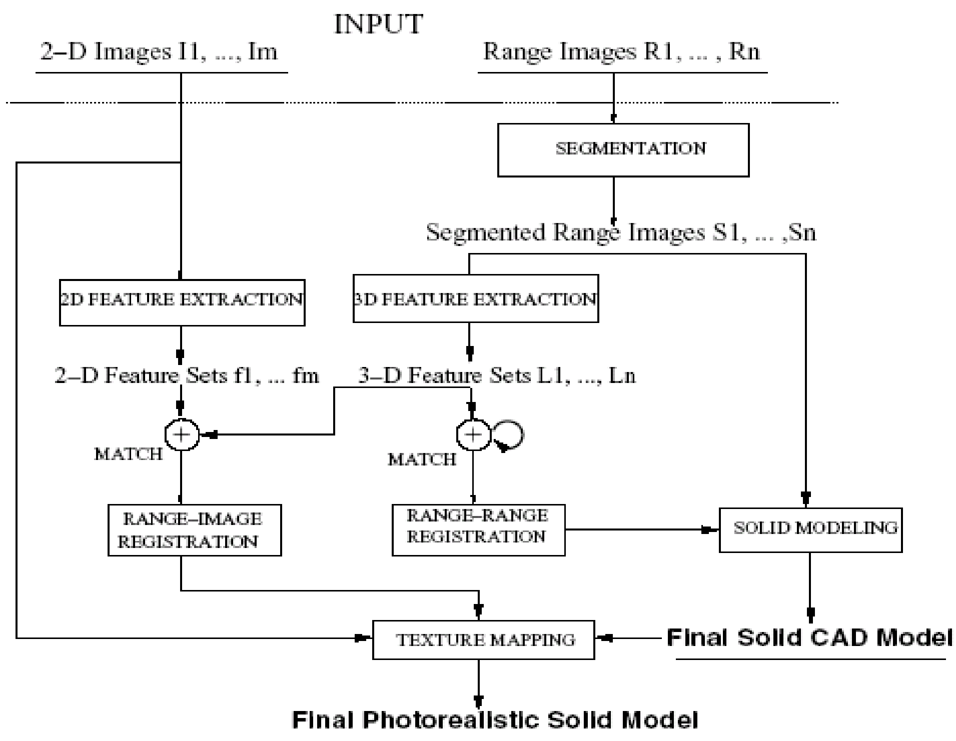


Figure 3.4. Algorithm for Obtaining Object Model (Stamos and Allen, 2002)

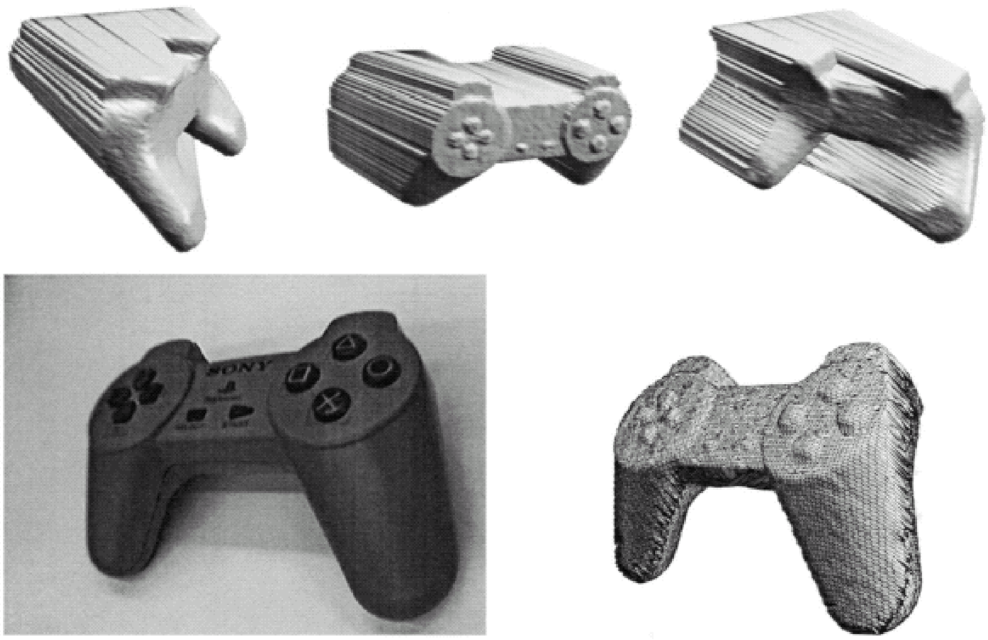


Figure 3.5. Some Results for Obtaining Object Model (Stamos and Allen,2002)

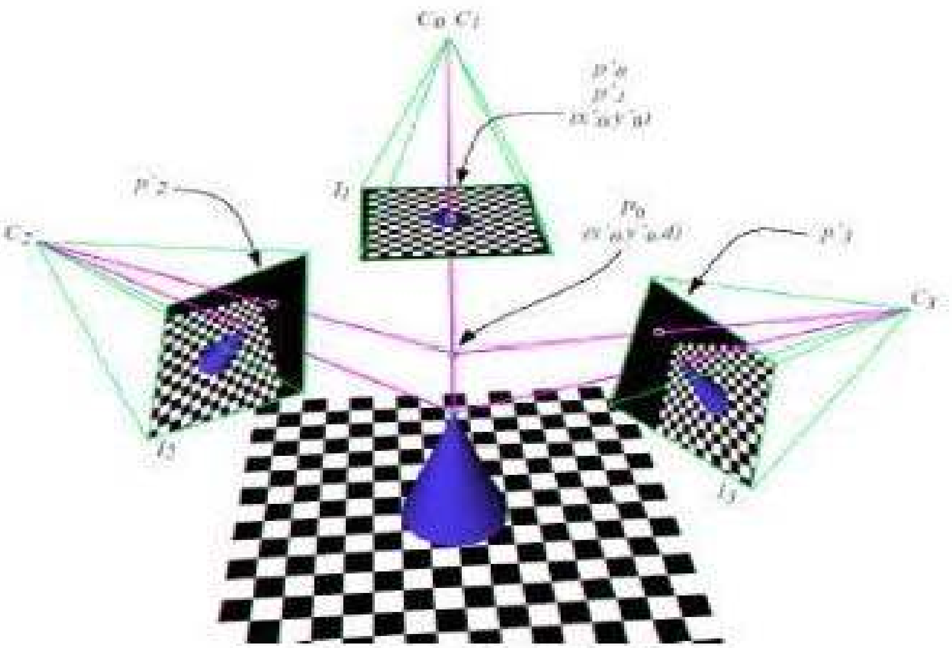


Figure 3.6. Projecting Multiple Images for Object 3D Modeling (Leung and Lovell 2003)

Figure 3.6 , shows the use of multiple projection images to obtain a 3D model. Leung and Lovell (2003) expand their work for obtaining 3D model from video frames, which could be quite useful in mobile robotic applications.

In our work, we do not implement a Object model acquisition, but we use the resultant forms of 3D object models. In our work , we use 3D object models in forms of vertex and wireframe solid models of objects to be manipulated.

The Objects for the workspace are basic 3D objects which are created by well known 3D editors ,such as 3D Studio and stored in files. During controller execution, objects are accessed from data files. This object workspace can be augmented as necessary for later stages.

Obtained 3D object sets are used for object information in terms of size , such as width, height, depth, geometric center and also topological properties of model , number of vertices ,faces together with triangles of the mesh in the wireframe model.

The length of finger link of robot hand middle finger is used as a unit size U to scale object dimensions with respect to robot hand finger size. By doing so, we relate object sizes to the robot hand size that will be need to grasp it. If the object size is extremely big, then object decomposition techniques can be used to separate the big object into its graspable parts sized appropriately to the robot hand size.

3.3. Our Definitions and Assumptions for Workspace

Object Model : An object model defines the geometry and surface of the grasped object. This model can be defined as a primitive shape (prism, cylinder, etc.), grid, mesh, or polygonal structure. Object Model provides a geometrical description of the surfaces of the object which is used in determining the accessible areas for each finger.

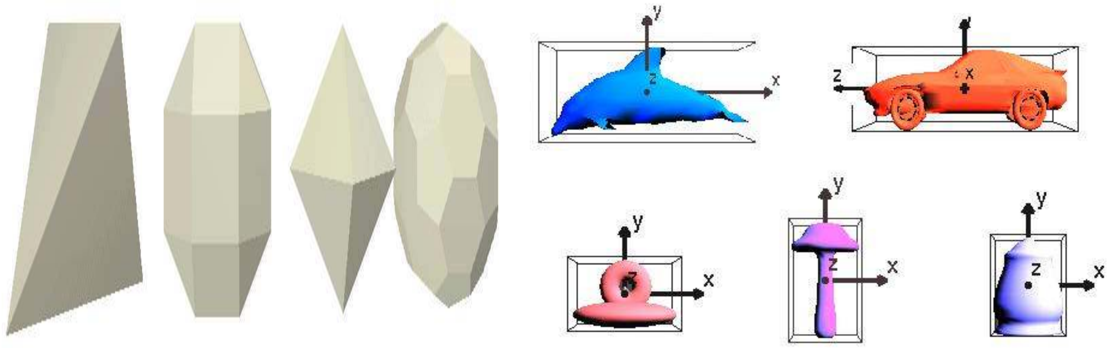


Figure 3.7. Example Object Models for Workspace

The object models are created by 3D CAD construction formats by external softwares. We use Simple rigid body and convex object models. The maximum object size should fit into a hypercube of size 4U, where U is size of our virtual hand middle finger phalange size. This assumption is based on the hand geometry, since the hand can reach this maximum size of objects with its fingertips. Figure 3.7 presents an example set for user created object models.

Hand Constraints : An analytical model of the hand set the hand constraints where the primary constraints are defined by the forward and inverse kinematics of the fingers of the hand, and by rules for collision detection of fingers. The hand model is used in conjunction with the object model and environment constraints to determine the intersections of the finger workspaces with the object surface. This defines the accessible areas of the object surface for each finger. We assume in our work that anthropomorphic hands are used.

Therefore, we also assume finger gaits will be in accordance to each other to avoid collisions. In our Virtual Hand Model, we also assume that for any proposed preshape, there exists all joint torque availability. In practice this assumption has a limited validity.

Environment Constraints: The Environment constraints typically include workspace of the manipulator to which the hand is attached, the location of the object in that workspace and the location of any obstacles in the workspace. In our Virtual Case, we do not have any environment constraints.

Assumptions for Kinematic Estimation:

The assumptions are used for the kinematic estimation part as below:

- The object and finger links are rigid bodies.
- The Contacts are fixed. No rolling or slip occurs at the contact points.
- Models of the object and the hand exist and are predefined.
- Contacts between the fingers and object are modeled as soft finger contacts with friction.
- Relative positions of the hand and object are known at each phase.
- Grasp is limited to a static and / or quasistatic, precision (fingertip) grasp.

3.4. Task Analysis Combined with Object Geometry

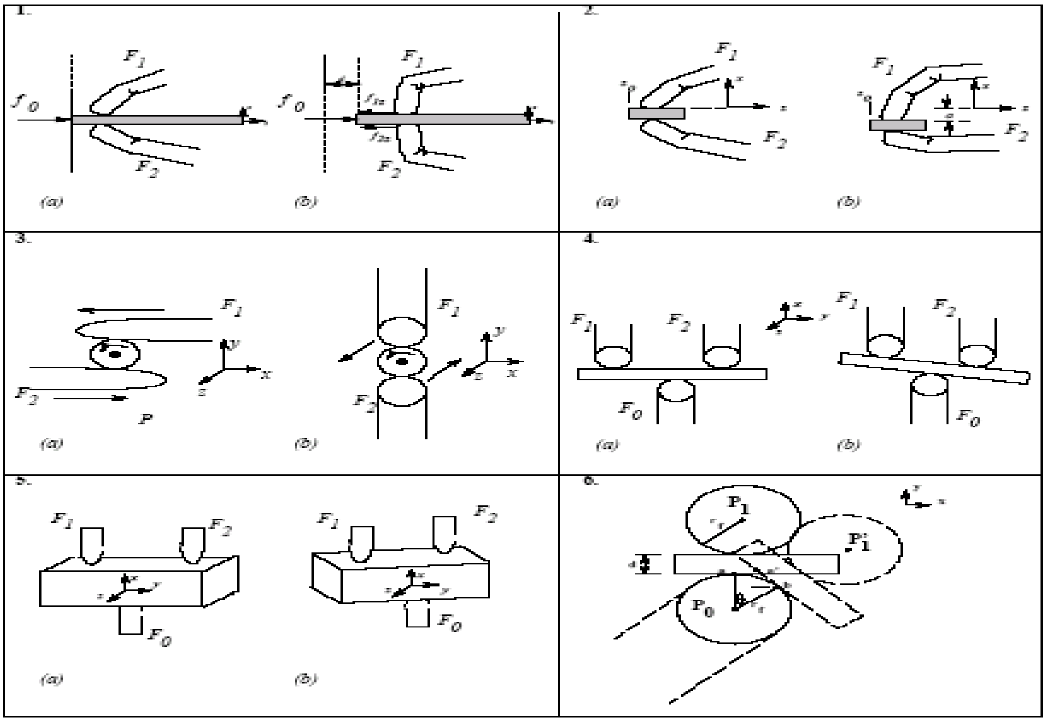
In chapter 2, we have noticed that task and object is related by means of some heuristic rules or by means of taxonomies, outcome of observations or experiments. We have already given Cutkosky's Taxonomy (1989) in the review section of chapter 2. We would like to restate heuristic rules from observations of Napier (1956) , which we utilized in our task analysis. According to Napier,

- The hand opening must be larger than the size of the object surface containing anticipated contact points.
- For a precision grip preshape, the thumb and the forefinger are in opposite direction and there is one contact per finger.
- During preshaping, the task determines the set of congruent axis of the object and the hand.

- If a graspable part of an object is larger than the width of the hand (measured along the transverse axis), the preshape involves the all available fingers. If it is small relative to the hand width, it involves fewer fingers. If it is small relative to the finger size, the preshape involves only two fingers, one of which is the thumb.
- For a task to be performed on a graspable object, there must be commonalities (properties that satisfy both task and object) established between task and object information. This set of commonalities constitutes the link between a given high-level task description and a given low-level object description. Thus, high-level task information must e translated into low level grasp kinematics.

Precision Manipulations are experimentally analyzed by Michelman (1998), using a UTAH-MIT robot hand, which has 4 fingers and a total 16 DOFs. According to his results, some manipulations are subject to object geometry under manipulation.

Hence , he categorized the precision tasks that could be performed with a UTAH-MIT hand. The main categories is given at fig. 3.8 .He also grouped his results into a taxonomy. The taxonomy is given at fig. 3.9. He also noted that , for a rigid body to move along a straight line vector P, the finger contact points must move in parallel to P.



The basic manipulations:
(1)-(2) translations toward and parallel to palm;
(3) Log-rolling; (4) Twiddle; (5) Pivot; (6) 'Y'-roll

Figure 3.8. Michelman's Basic Manipulations (Michelman, 1998)

Michelman, concluded from his experiments that:

- Each primitive manipulation function performs a single translation or rotation.
- To perform arbitrary translations and rotations, the primitive functions must be combined either by merging or by sequencing them.

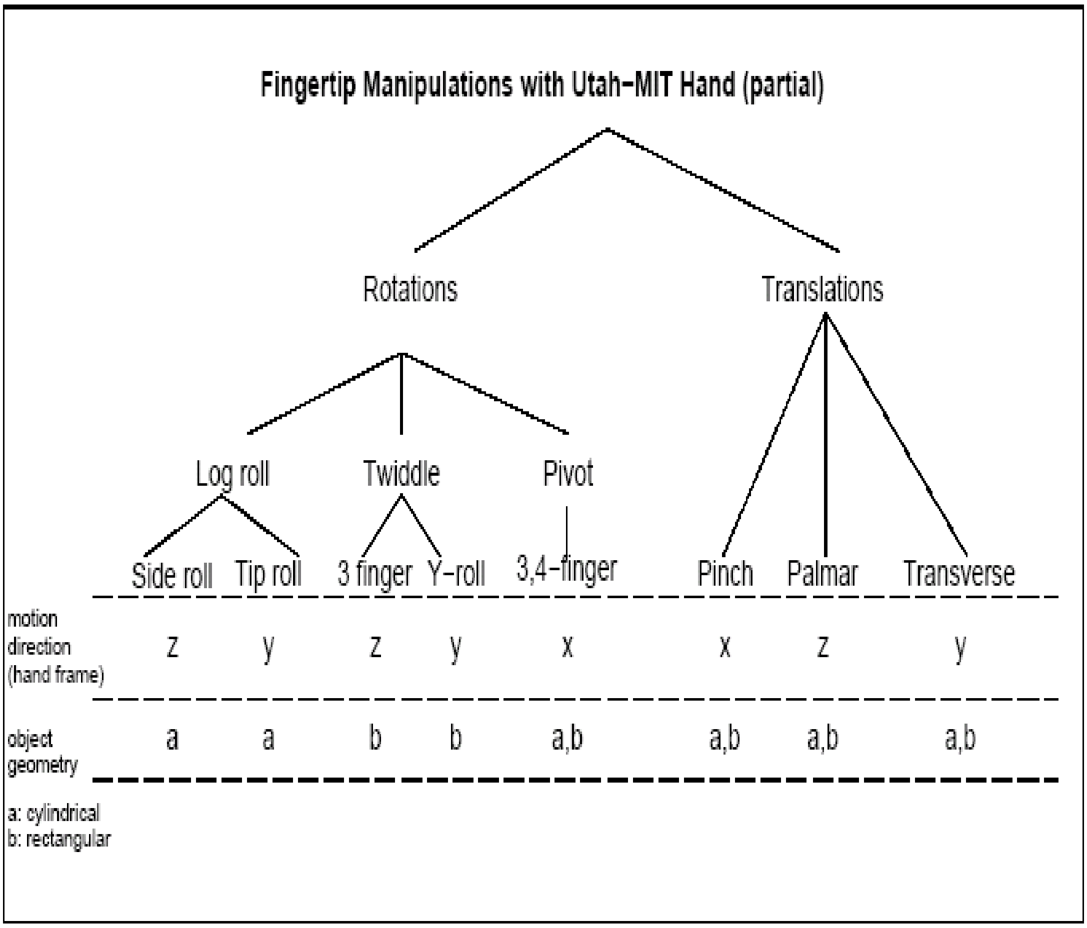


Figure 3.9. Michelman's Fingertip Manipulations Taxonomy (Michelman, 1998)

In addition, Arbib (1981) also made some observations for grasp preshaping. According to his observations,

- The Reach and Grasp movements occur in parallel in the hand preshapes for the grasp while the reach is in progress. The reach is concerned with the object position. The grasp is concerned with the object characteristics,i.e. shape and orientation.
- For each grasp type, the preshape defines a volume between the fingers called the Approach Volume. The entrance to this volume is via a surface formed by joining the fingertips of the preshaped hand by straight lines. The approach axis for the grasp to be along the normal to this area through its centroid. The hand approaches the object along its approach axis.

Inspired from all previous observations and experiments mentinod till here, in our perspective, task is a dextrous manipulation primitive task which is basically a spatial displacement of object after securely grasp.

According to Chasles theorem , every spatial displacement is the composition of a rotation about some axis and a translation along the same axis. Mason (1987) gives the theoretical background for this theorem in his textbook. Hence , from robotic manipulation viewpoint, we can represent a primitive task as the decomposition of a translation and a rotation. Spatial displacements can also be represented by using Screw Systems. Ball(1900) introduced Screw Systems and followers such as Duffy (1996), Hunt(1978) ,Roth (1984) , and Waldron(1966) expanded the use of Screw Systems.

We define **Q=[T;R]** as our task representation. Hence in our representation task has two main parts: Translation and Rotation. In this representation, **T=(Tx,Ty,Tz)** represents vector for the translational displacement and **R=(Rx,Ry,Rz)** represents the pure rotational displacement, with respect to origin of coordinate frame. A couple representation of two parts constitutes the overall spatial displacement defining our manipulation task and is similar to Wrench representation in Screw Systems.

Wrench representation has also two components. The first one is the force component that represents the net force vector on a body refrenced to a frame and the second one is the moment part that represents the pure moment on a body about origin of the frame. When magnitudes of both force and moment are unity , then wrench and task representation forms are identical.

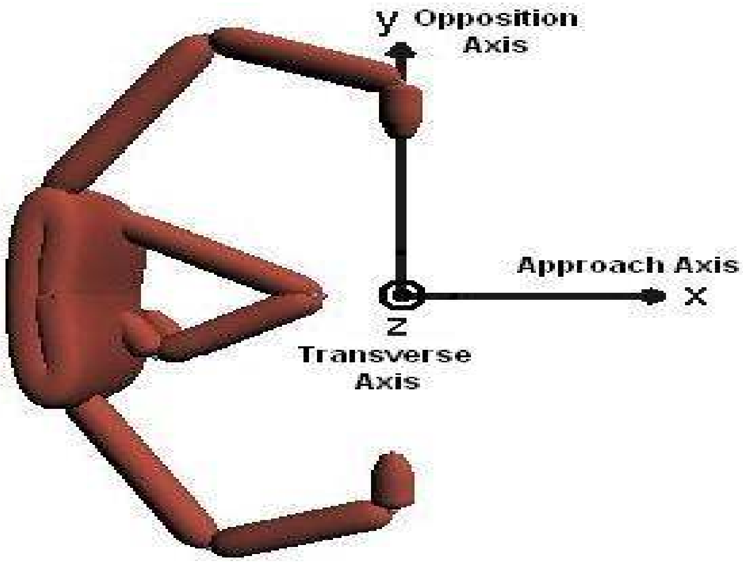


Figure 3.10.Principal Axes of Coordinate Frame labeled with Grasp Terms.

In figure 3.10, we identify the principal axes of our global coordinate frame, which is centered on the object geometrical center as object cooordinate frame, according to the grasp terminology. In this identification, the correspondance between our frame axes and Michelman's representation are as follows: Approach Axis to corresponds to the Palmar Axis and Opposition Axis to the Pinch Axis, in Michelman's application. The difference comes from the reference coordinate frames. In his work , he bases computations upon a hand coordinate frame. In our work, we base on a global coordinate frame centered on the geometrical center of the object, COG.

In order to develop the sample task space for our implementation we use our definition and frames as defined **Q=[T;R]** and take task primitives from Michelman's work . For instance, translational tasks of Michelman can be expressed using our task definitions as pure translational displacement and zero rotation **Q=[T;0]**. Similarly, pivoting task around a principle axis can be expressed as zero translational displacement and pure rotation **Q=[0;R]**.

Using this analogy, we constructed our sample task space in a mapping by using the primitive tasks from Michelman . In figure 3.9, sample task space is given.

SAMPLE TASK SPACE DESCRIBED IN WRENCH FORM

TASK T1: TRANSLATIONS ALONG APPROACH & OPPOSITION AXES, ROTATION AROUND OPPOSITION AXIS (PIVOTING)

IDX	TASK SAMPLE WRENCHES							
	APPR. AXIS	OPP. AXIS	Tx	Ty	Tz	Rx	Ry	Rz
T1_1	X (W)	Z (D)	1	0,1	1	0,05	0,05	0,8
T1_2		Y (H)	1	1	0,1	0,05	0,8	0,05
T1_3	Y (H)	X (W)	1	1	0,1	0,8	0,05	0,05
T1_4		Z (D)	0,1	1	1	0,05	0,05	0,8
T1_5	Z (D)	X (W)	1	0,1	1	0,8	0,05	0,05
T1_6		Y (H)	0,1	1	1	0,05	0,8	0,05

TASK T2: TRANSLATIONS ALONG APPROACH & OPPOSITION AXES, ROTATION AROUND APPROACH AXIS (TWIDDLE)

IDX	TASK SAMPLE WRENCHES							
	APPR. AXIS	OPP. AXIS	Tx	Ty	Tz	Rx	Ry	Rz
T2_1	X (W)	Z (D)	1	0,1	1	0,8	0,05	0,05
T2_2		Y (H)	1	1	0,1	0,8	0,05	0,05
T2_3	Y (H)	X (W)	1	1	0,1	0,05	0,8	0,05
T2_4		Z (D)	0,1	1	1	0,05	0,05	0,8
T2_5	Z (D)	X (W)	1	0,1	1	0,8	0,05	0,05
T2_6		Y (H)	0,1	1	1	0,05	0,8	0,05

TASK T3: TRANSLATION ALONG APPROACH AXIS
ROTATION AROUND APPROACH AXIS (TWIDDLE)

IDX	TASK SAMPLE WRENCHES							
	APPR. AXIS	OPP. AXIS	Tx	Ty	Tz	Rx	Ry	Rz
T3_1	X (W)	Z (D)	1	0,1	0,1	0,8	0,05	0,05
T3_2		Y (H)	1	0,1	0,1	0,8	0,05	0,05
T3_3	Y (H)	X (W)	0,1	1	0,1	0,05	0,8	0,05
T3_4		Z (D)	0,1	1	0,1	0,05	0,8	0,05
T3_5	Z (D)	X (W)	0,1	0,1	1	0,05	0,05	0,8
T3_6		Y (H)	0,1	0,1	1	0,05	0,05	0,8

TASK T4: TRANSLATIONS ALONG APPROACH , OPPOSITION AXES ,
ROTATIONS AROUND OPP. & APPR. AXES (PIVOTING & TWIDDLE)

IDX	TASK SAMPLE WRENCHES							
	APPR. AXIS	OPP. AXIS	Tx	Ty	Tz	Rx	Ry	Rz
T4_1	X (W)	Z (D)	1	0,1	1	0,8	0,05	0,8
T4_2		Y (H)	1	1	0,1	0,8	0,8	0,05
T4_3	Y (H)	X (W)	1	1	0,1	0,8	0,8	0,05
T4_4		Z (D)	0,1	1	1	0,05	0,8	0,8
T4_5	Z (D)	X (W)	0,1	0,1	1	0,05	0,05	0,8
T4_6		Y (H)	0,1	0,1	1	0,05	0,05	0,8

NOTE: The Wrench Values for Rotation Axis must have same signs of their relevant Translational Axis: Both T and R axis should have same sign.

Figure 3.11. Task Space Described in Wrench Form using Task Primitives.

In the task sample space shown at fig. 3.11, we formed basically 4 task categories described by descriptions given at fig 3.11. In these categories, there are also sub-categories. It is also possible to have magnitues in negative values ; however, the relevant T axis and R axis components of wrench should have same sign , both either positive or negative.

To achieve, a task that is described in our work,in terms of Translational and Rotational components, the hand needs to have an approach and opposition axes representation, which is also expressed in our sample task space. That is to say the approach for the hand is only partially determined from the task representation as it was observed by Napier and Arbib. A further study of this item resulted in the categorization of the axes as hand Approach Planes.

In general for a 3D object, without any constraint in the object space, it is possible to approach from many different directions and relevant view planes. For an asymmetric object, each approach direction leads to a different contact positions and grasp contact patterns. Hence, different contact spaces have different kinematic effects. In practice, due to constraints and obstacles, approach directions may easily be reduced to a few number . Our preshaped hand approaches to object from left to right in a certain direction with a controller that uses, the size information of length, width and depth, relative to that Approach Direction.

In a 12-Plane categoric scheme that can encompass all possible cases of grasping approach and covering our primitive task space in figure 3.12. In this scheme, horizontal axis for a plane corresponds to the approach axis for the hand and vertical axis to opposition axis.

We then deduce a relation between task and object geometry, in terms of Approach Planes. However, this relation is not found to be a one-to-one mapping. In other words, object geometry and contact kinematics yields complementary parts of the relation between task and object. To explain further, as an observation from sample task space, for instance task items T1_1 and T2_1 as shown in figure 3.11, tasks can have common approach and opposition axes, although these tasks are different. This commonality is the only a partial relation, that does not suffice for complete guidance for the reach of hand. The reach of hand requires an established proper matching of an Approach Plane and task. In order to have this matching ,we have decomposed task space using 12-Plane categoric scheme as shown in figure 3.12.

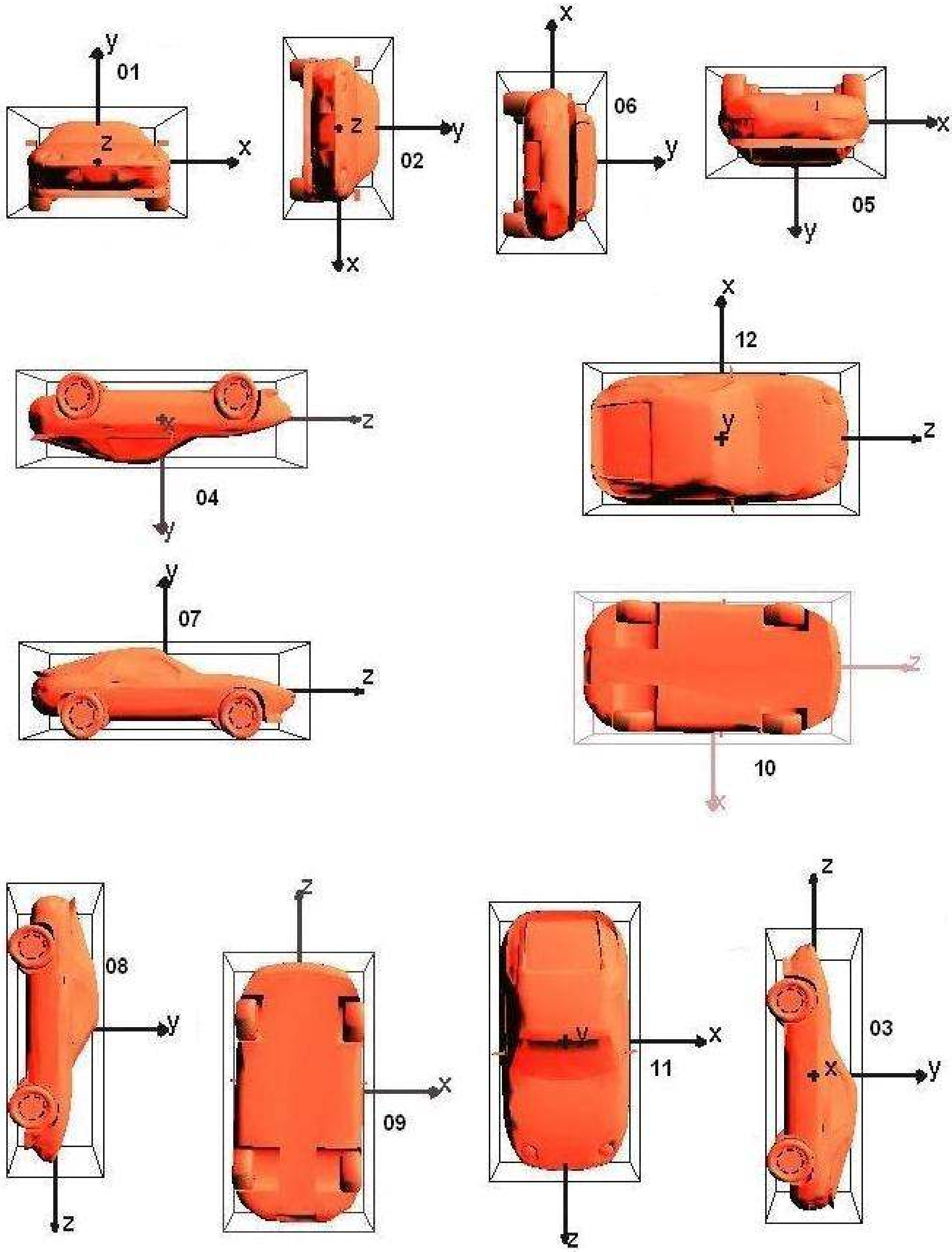


Figure 3.12. Hand Approach Planes Corresponding to the Task Space.

In addition, by means of 12 Approach Plane categorization, we partition the total grasp space, for the approach of robot hand. The preshape execution should then adopt one of these planes. At this point, a question can be asked that why planes are in alignment with pricipal axes and why not to have any arbitrary approach planes. The answer for this question is that principal axes forms the basis planes that span the whole grasp space. This analogy has standing points coming from our task decomposition into primitive displacements along principal axes. It is also a point of discussion that why number of planes is 12. The reason for that, it is possible to cover all cases of approach, with 12 basis planes, formed by variations in approach and opposition axis.

For an Approach Plane identified by a plane index number, the grasp axes match to object geometric dimensions, especially transverse axis of the Approach Plane is used for the determination of the number of fingers to be involved in preshaping ; hence it is also used in contact kinematics analysis. For instance, for Approach Plane no 1 in fig 3.12, the transverse axis of preshape corresponds to the Z axis . This is illustrated in figure 3.13

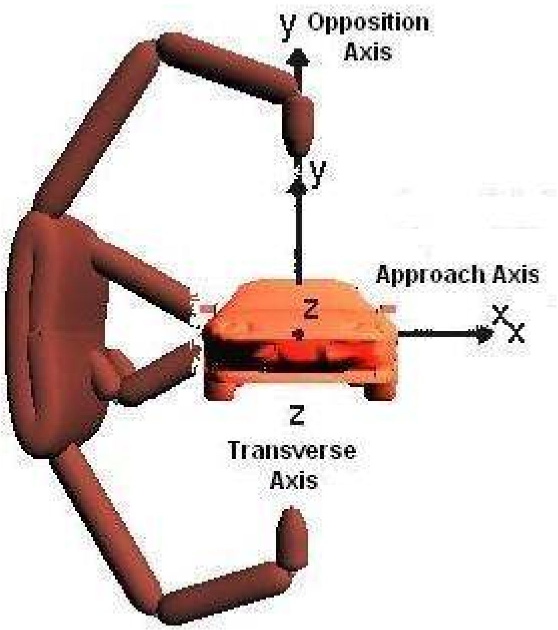


Figure 3.13. Hand Approaching to Object along Approach Plane 1.

The number of fingers to be involved in a preshape approaching along Plane no 1, is determined using the object size information along transverse axis, that is depth size of object in this case. This is as a result of Napier's heuristics. For each Approach Plane which gives a correspondance between an object dimension size and transversal axis of preshape , use of this heuristic approach yields the number of fingers to be involved in the preshape for that approach plane.

In our study, thumb is always involved in preshape, the number of forefingers involved is determined by a rule based algorithm. In the algorithm , while the number of fingers are determined, the position of thumb is also determined. In order to determine number of fingers to be involved we obey the following rule based algorithm, given in next page.

R1-If (Object Size Corresponding to Grasp Tranverse axis < 2U size) then

{

- Only Forefinger is Index finger and Thumb is opposing Index finger.
- # VF=1. (Number of Virtual Fore Fingers).
- The contact locations are aligned at the center , concerning transversal axis and other coordinates determined by further analyses.

}

R2-If (Object Size Corresp. to Grasp Tranverse axis >= 2U size and <4U size) then

{

- Indexfinger and Midfinger involved. Thumb is opposing midcenter between forefingers.
- # VF=2. (Number of Virtual Fore Fingers).
- The contact locations are aligned along transversal axis . keeping thumb at center, and other coordinates determined by further analyses.

}

R3-If (Object Size Corresponding to Grasp Tranverse axis >= 4U size)
then

{

- Indexfinger, Midfinger and Ringfinger involved. Thumb is opposing Midfinger at center between forefingers.
- # VF=3. (Number of Virtual Fore Fingers).
- The contact locations are aligned along transversal axis . keeping thumb and midfinger at center, and other coordinates determined by further analyses.

}

Where , the Hand Model Base transversal size is 5U. In our workframe, the maximum object dimension is 6U.

3.5. Determination of Contact Points on Object Model by Collision Detection

After having determined the Approach Plane or axes for an approach with number of fingers to be involved in grasp, in order to make kinematic contact analysis, the determination of contact points becomes a necessary action to be taken. In order to determine contact locations on the object model, we need to make an analysis on the geometric representation of object model, which is composed of polygonal surfaces. This analysis requires estimation of posssible landing points for fingertips. But, for an object which is only represented by polygonal geometric representation in terms of vertices and triangles, we cannot make pre estimations for contact locations and cannot deduce about contact surface properties, for a given approach direction.

Lin M.C. et al, made a survey about collision detection between geometric models. In her survey, Lin (1998) states that the goal of collision detection (or contact determination) is to automatically report a geometrical contact when it is about to occur or has actually occured. The geometric models may be polygonal objects , splines or algebraic surfaces. Collision detection also enables simulation based design and spatial reasoning . In the survey, the state of the art tecqniques were mentioned as well as well known methods for polygonal objects. Hence, Such a determination can only be realistic with a collision detection of fingertips on object surface in simulation environment , as it would be in an actual implementation.

As we stated in our workspace assumptions and object modeling sections, our geometric representation uses polygonal methods and specifically our models are vertex based triangles. Hence, for collision detection analysis to determine contact surface with this representation, one of the most simple and efficient methods is the Ray Triangle Intersection. This method, uses so called “Barycentric approach” , which can be found at Shirley et al (2003). Another optimized Ray Triangle Intersection Method can be found at Möller et al. work (1997).

In our implementation, we use “Ray Triangle Intersection” method to determine possible contact positions as follows:

For a selected Approach Plane, which determines approach and opposition axes of grasp with respect to object geomerty, and for a given approach value which is the coordinates of fingertips along the approach axis, we make a search of possible surfaces to land upon. We make this search, by using Ray Triangle Intersection methods. For each finger to be involved in grasp, we define a Ray originating from fingertip coordinate value at maximum opening value of 4U , in parallel to the opposition axis . After having determined Ray parameters and properties according to the Barrycentric Method, a search runs on the object model in the memory using intersection technique, in a way that all fingers involved are tested simultaneously for each of triangles. If intersection occurs, then coordinates of the contact point and surface normal for the collided traingle is determined and stored for the related finger and approach value. That is to say , information stored is composed of the collision point, a cartesian point **P=(x,y,z)** and three vertices forming the collided polygon (triangle) **P0,P1,P3**, which are also cartesian points. In this way , we test existence of a contact surface on the object for a given approach.

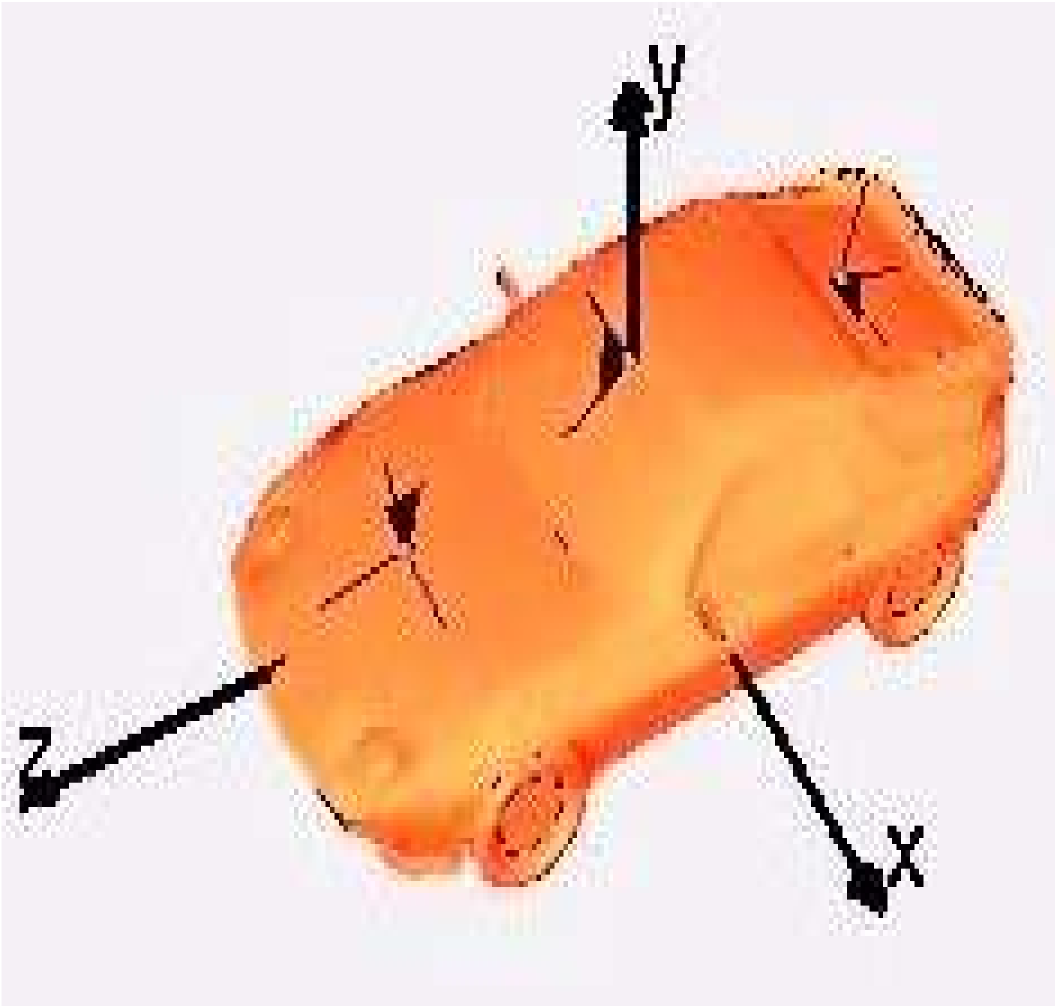


Figure 3.14. Triangles Intersected by Test Rays on Object Model Surface.

Figure 3.14 ilustrates results of our collsion test on a surface of model object, as shown dark triangles and surface normals. Eventhough there exists better collision detection algortihms I-Collide, V-Collide given in Lin M. et al, we prefer to use Ray Triangle technique. The popular methods make search in terms of distance between vertices of colliding objects, hence require more computational power for our case .

3.6. Estimations for Kinematic Analysis of Virtual Grasp

The kinematic effects of contacts on a body can be analyzed using constraint analysis methods. Hunt (1978) presented geometrical methods for kinematic analysis of mechanisms. Duffy (1996) analyzed kinematics of parallel manipulators. Waldron (1966), worked on constraint analysis of mechanisms, using screws. So called "Parallel Law" of mechanisms states that "The motions that a body is capable of performing while under the action of multiple wrenches of contact are the logical intersection of the motions it can perform under the action of each wrench acting alone, with the others physically disconnected". Ohvoworiole (1980) and Roth (1984) developed kinematic analysis methods using screw systems. Konkar (1993) in his thesis, performed incremental kinematic analysis using screw systems for synthesis of mechanisms. Figure 3.15 illustrates screw system representation of different parallel and serial kinematic mechanisms .

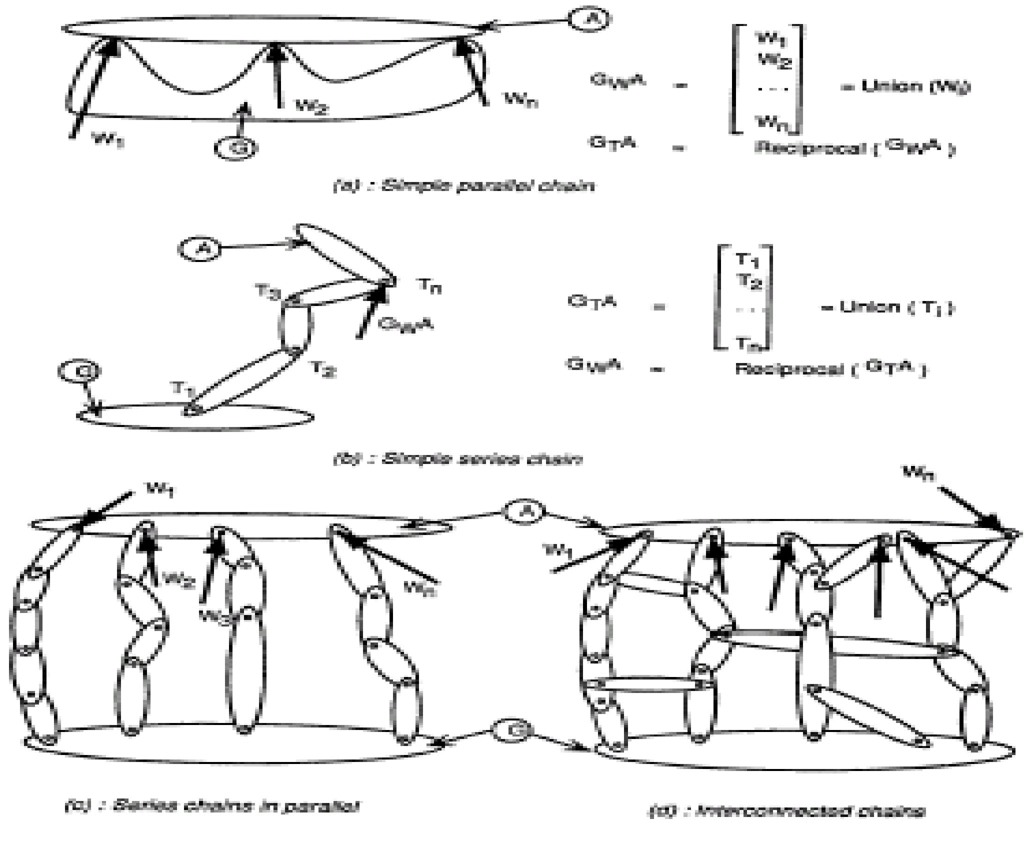


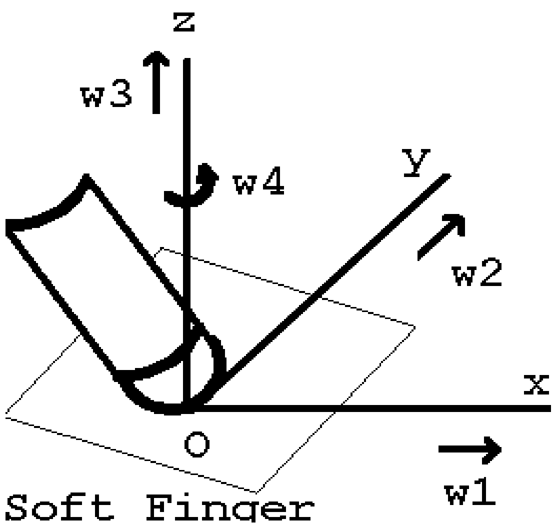
Figure 3.15. Screw System Representation of Kinematic Mechanisms (Konkar,1993)

In figure 3.15, for a parallel manipulator a representation of contact kinematics is given in terms of Wrench Union and Twist Intersection which we later use to develop certain measures. In addition, Adams and Whitney (2001) based on Konkar's algorithm, extends the constraint analysis methods based on screw systems.

From kinematic perspective, a multfingered robot hand is basically a parallel manipulator mechanism composed of serial chained links operating in parallel . Therefore, application of screw system methods for parallel manipulators can be extended to robot hand kinematics. In addition, robot hand contacts on a body can be analyzed as constraining contacts, as mentioned in survey sections. Salisbury (1982) developed contact models for robotic grasping using screw systems. Mason and Salisbury (1985) analyzed the contact kinematics of robot hand based on contact models using screws. Lipkin and Patterson (1988) proposed a method for control of robotic manipulator , using screw representation.

Unlike the other works in literature based on point contact with friction model , in our work, we use Salisbury's soft finger contact model which has been less worked on . The finger contact models were analyzed in detail by Akella (1993) in his thesis. We would like restate Salisbury's contact model at this point.

Figure 3.15 indicates the contact model of Salisbury's Soft finger, where contact plane normal is along z axis.



w1=(1,0,0,0,0,0)
w2=(0,1,0,0,0,0)
w3=(0,0,1,0,0,0)
w4=(0,0,0,0,0,1)
t1 =(1,0,0,0,0,0)
t2 =(0,1,0,0,0,0)

Figure 3.16. Salisbury's Soft Finger Contact Model

In figure 3.16 , $\mathbf{w_i}$ indicates the contact wrenches in ray coordinates of the screw systems and $\mathbf{t_j}$ indicates the twists in axis coordinates of the screw systems. Twists are not shown on figure yet but , they are included in our analysis for our controller.

Salisbury in his model of soft finger contact, gives unit wrenches and twists; however the practical implementation of this model concerns Coulomb friction force. Therefore, wrenches due to frictional components, i.e. tangential forces and moment around contact normal, are scaled with some factors, using following equation 3.6.1 , from Ciocarlie et al (2005).

$$f_x^2 + f_y^2 + \frac{\tau_z^2}{e_z^2} \leq \mu^2 f_n^2 \qquad (3.6.1)$$

In the equation, $\mathbf{f_x}$ corresponds to $\mathbf{w_1}$, $\mathbf{f_y}$ to $\mathbf{w_2}$ and $\mathbf{f_n}$ to $\mathbf{w_3}$. In addition scaled torque term $\boldsymbol{T}$ corresponds to $\mathbf{w_4}$.

After having analysis of our contact model, we now implement our virtual grasp using the mentioned contact model , to estimate kinematic effects of the virtual grasp. We demonstrate our implementation on a case. Given the approach plane , the number of fingers and the approach value, after we determined contact locations with existence test as in previous section, we assume that our fingers land on contact points.

We assign contact parameters to each virtual contact, shown fig 3.17. We also calculate the contact position vectors $\mathbf{r_i}$ with respect to the **COG** (Object Geometric Center) of the object which was previously estimated. The contact position vector $\mathbf{r_i}$ is calculated using the results from collison tests for the contacts and stored.

$$\mathbf{r_i = (x_i, y_i, z_i)} \qquad (3.6.2)$$

where the position vector indexes are $\mathbf{r_1}$ for thumb , $\mathbf{r_2}$ for index, $\mathbf{r_3}$ for middle finger and $\mathbf{r_4}$ for fourth finger. $\mathbf{C_i}$ denotes the contacts and $\mathbf{W_i}$ denotes contact wrench.

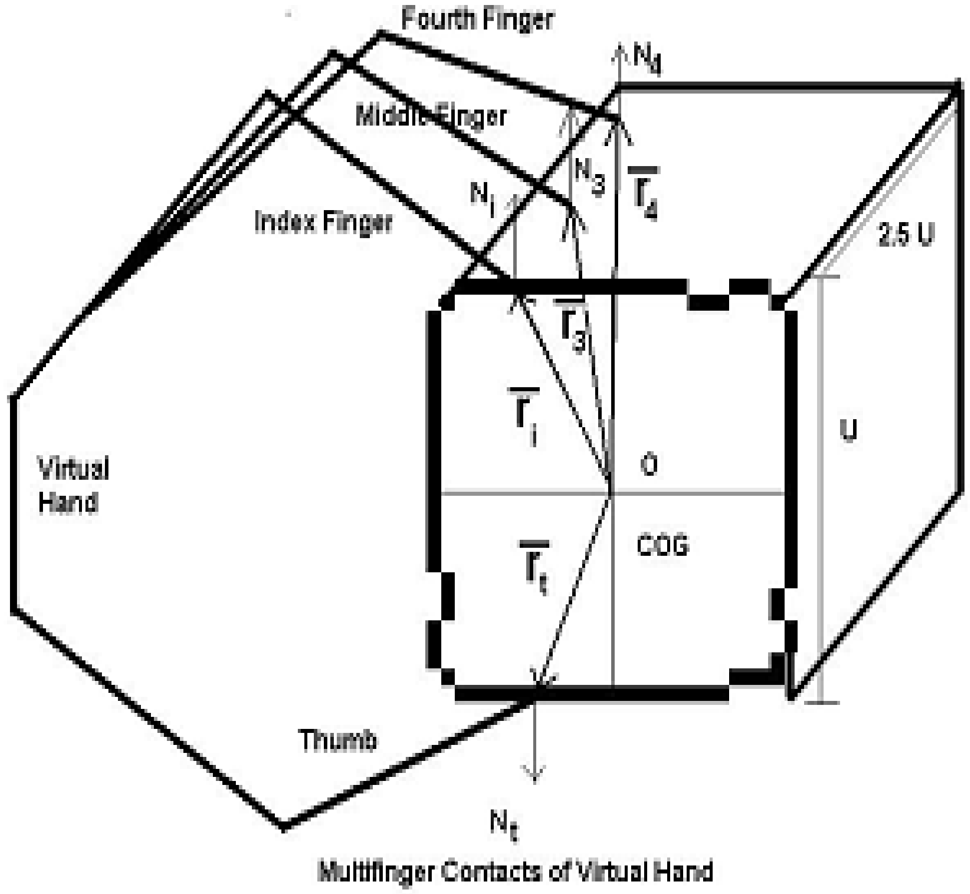


Figure 3.17.Virtual Four Finger Contacts with Normals and Position Vectors.

For the all possible contacts, the contact wrenches are estimated in two forms: one as a contact wrench matrix and other as a force and a pure couple. These forms are stored for further use. Here from the figure 3.17, we notice that all contact surface normals are along opposition axis (or –opposition axis for thumb). Since, the contact normals are in the same direction, and the contact wrenches for contacts $\mathbf{C_2,C_3,C_4}$, which are indicated by position vectors $\mathbf{r_i}$ in fig 3.17 are same therefore, we write only for index finger contact $\mathbf{C_{2,}}$. In general, all contact wrenches are indiviudally estimated.

The contact wrenches can also be written in screw system couple form in ray coordinates. That is to say a couple is formed as:

$$W_i = [f_i ; m_i] \tag{3.6.3}$$

where $\boldsymbol{f_i}$ denotes the force applied at the contact and $\boldsymbol{m_i}$ denotes the moment existed at the contact.

If we rewrite W_1 and W_2 in this form , for the fig 3.17, we obtain below couples:

$$W_1 = \begin{bmatrix} 1 & -1 & 1 & ; & 0 & -1 & 0 \end{bmatrix} \quad W_2 = \begin{bmatrix} 1 & 1 & 1 & ; & 0 & 1 & 0 \end{bmatrix} \tag{3.6.4}$$

but when we include frictional coeffients in wrench eqn (3.6.1),

$$W_1 = \begin{bmatrix} K_a & -K_n & K_o & ; & 0 & -M_n & 0 \end{bmatrix} W_2 = \begin{bmatrix} K_a & K_n & K_o & ; & 0 & M_n & 0 \end{bmatrix} \tag{3.6.5}$$

The coefficients are determined experimentally and their values are discussed in the performance analysis chapter. For illustration purposes, we use unity magnitudes.

The other representation of contact wrench is in matrix for each contact is therefore written as below:

$$W_1 = \begin{array}{c} \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \end{array} \begin{array}{l} \leftarrow w_1 \\ \leftarrow w_2 \\ \leftarrow w_3 \\ \leftarrow w_4 \end{array} \quad W_2 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 \end{bmatrix} \begin{array}{l} \leftarrow w_1 \\ \leftarrow w_2 \\ \leftarrow w_3 \\ \leftarrow w_4 \end{array} \tag{3.6.6}$$

where $\mathbf{W_2=W_3=W_4}$, only for fig 3.17 illustration.

After having calculated contact position vectors and couples for each contact , we use Poinsot's theorem of Screw Theory, to move all contact wrenches to the **COG**. By doing so we obtain resultant net wrench on the object in couple form.

<u>The Poinsot Theorem :</u>

A couple in screw quantities, can be projected to a point whose position vector lies from that point towards the couple screw axis, using the following transformation given in equation 3.6.7 .

$$W_o = \begin{bmatrix} \overline{f_o} \\ \overline{m_o} \end{bmatrix} = \begin{bmatrix} \sum_i \overline{f_i} \\ \sum_j \overline{m_j} + \sum_i \overline{r_i}\, x\, \overline{f_i} \end{bmatrix}$$

$$= \begin{bmatrix} \sum_{i=1} \overline{f_i} \\ \sum_{j=1} \overline{m_j} + \end{bmatrix} + \begin{bmatrix} 0 \\ \sum_{\iota=1}^{3} \overline{r_k}\, x\, \overline{f_i} \end{bmatrix} \qquad (3.6.7)$$

Where, in our case for each finger, W_o denotes the projected wrench to **COG** , and f_i denotes the forces applied at the contact and m_j denotes the moments existed at the contact $\mathbf{C_i}$, and $\mathbf{r_i}$ the contact position vectors from the **COG ,** given in fig 3.17. **(x)** operator is the cross product operator. The **i** index ranges from 1 to 3 for force but since we have only one contact per finger, **k** is 1 for position vector. In our case, also **j** is 1 for our finger model.

The Resultant Net Wrench from all fingers are obtained direct sum of projected couples .

$$W_n = \sum_k W_{ok} \qquad (3.6.8)$$

We use resultant net wrench , as pattern for indication of grasp, which we explain at section 3.8

In order to futher explain , we give an example for calculations for resultant wrench. We use figure 3.17 for the calculations given in this example.

Example 1. The object is a rectangle as shown in figure 3.17.

Rectangle has Estimated Size: X=1.0769U, Y=3.846U, Z=2.385U

We start to calculate the contact coordinates using our previous assumptions of being 0.5U away from near corner and near edges. We use previously derived r_i and we substitute object size info into these derivations.

We find out following contact position vectors.

r_1= (-0.538U, 1.423U, 0.712U) for thumb

r_2= (0.538U, 1.423U, 0.712U) for index finger

r_3= (0.538U, 1.423U, -0.288U) for mid. finger

r_4= (0.538U, 1.423U, -1.288U) for fourth finger

If we rewrite W_1 and W_2 in this form , for the fig 3.16, we obtain below couples:

$$W_1 = \begin{bmatrix} 1 & -1 & 1 & ; & 0 & -1 & 0 \end{bmatrix} \qquad W_2 = \begin{bmatrix} 1 & 1 & 1 & ; & 0 & 1 & 0 \end{bmatrix}$$

We substitute the values into the equations. We assume U=1. W_{oi} indicates the resultant net wrench for finger i, projected to COG.

$\sum_{\iota=1}^{3} \overline{\boldsymbol{r}_k} \, \boldsymbol{x} \, \overline{\boldsymbol{f}_i} =$ r_1 x f_{i_total} =(-0.538, 1.423, 0.712) x (-1, 1,1) = (0.711, -0.174, 0.885)

$W_{o1} = W_1 +$ r_1 x f_{i_total} = [-1,1,1,-1,0,0]+[0,0,0,0.711,-0.174,0.885]

W_{o1} =[-1,1,1,-0.289,-0.174,0.885]

r_2 x f_{i_total} =(0.538, 1.423, 0.712) x (1, 1,1) = (0.711, 0.174, -0.885)

$W_{o2} = W_2 +$ r_2 x f_{i_total} = [1,1,1,1,0,0]+[0,0,0,0.711,0.174,-0.885]

W_{o2} =[1,1,1,1.711,0.174,-0.885]

r_3 x f_{i_total} =(0.538, 1.423, -0.288) x (1, 1,1) = (1.711, -0.826, -0.885)

$W_{o3} = W_3 +$ r_3 x f_{i_total} = [1,1,1,1,0,0]+[0,0,0,1.711,-0.826,-0.885]

W_{o3} =[1,1,1,2.711,-0.826,-0.885]

r_4 x f_{i_total} =(0.538, 1.423, -1.288) x (1, 1,1) = (2.711, -1.826, -0.885)

$W_{o4} = W_4 +$ r_4 x f_{i_total} = [1,1,1,1,0,0]+[0,0,0,2.711,-1.826,-0.885]

W_{o4} =[1,1,1,3.711,-1.826,-0.885]

$\boldsymbol{W_n = W_{o1} + W_{o2} + W_{o3} + W_{o4}}$ **= [2,4,4,7.844,-2.652,-1.77]**

By using matrix notation for each contact, we obtain Wrench Union **(WU)** of the Virtual Grasp, by stacking on top wrench matrices of individual contacts, which is given Adams and Whitney (2001) , which is given in equation 3.6.9 .

$$WU = \begin{bmatrix} \underline{\underline{W\ 1}} \\ \underline{\underline{W\ 2}} \\ \underline{\underline{W\ 3}} \\ W\ 4 \end{bmatrix} \tag{3.6.9}$$

We use wrench Union form , for the estimation of grasp performance measures, which we explain at the section 3.7 .

3.7.Grasp Performance Measures

For the analysis of grasps performance measures existing in the literature based on task attributes. Among them, Stability and Manipulability are widely analyzed and evaluated. In our work, we will also utilize these performance measures. Here we would like to restate definitions of Stability and Manipulability, which are available in the literature works given in chapter 2 .

Stability : It is a measure of the tendency of the object to return to its original position after disturbances. If there is a disturbance, equilibrium can be restored in stable cases. When higher stability is required in the task, control is required over large distrubances occuring at more locations on the object. As a task attribute, we can use the analytical definition of stability which will be stated later on.

Manipulability : It is the ability to impart motions to the object while keeeping fingers in contact with the object. This maps directly into the analytical measure of manipulability. This definition is similar to a connectivity measure but it also takes into account the approach orientation in which connectivity of the grasp should be maximized.

Our Interpretation of Stability and Manipulability :

In our case, we define grasp perfomance measures using the results obtained kinematic analysis of virtual grasp, by utilizing the properties of screw systems. We use Wrench Union to represent Wrench Space and Twist Intersection to represent Twist Space for the Grasp. Before going further, we would like restate reciprocity theorem between twists and wrenches.

The reciprocity theorem states that a wrench is reciprocal to a twist if their reciprocal product is zero and this means, the wrench can not produce any work or motion in the direction of the twist.

That is to say, the resultant wrench has nothing to overcome the degrees of freedom represented by its reciprocal space twist.

$$W = [f;m]^t \quad T = [\omega;v]^t \; ; \tag{3.7.1}$$

The reciprocal product:

$$W \otimes T = [f \bullet v + m \bullet \omega] \tag{3.7.2}$$

In order to find the reciprocal space twist, we make use of Wrench Union of the grasp . Adams and Whitney (2001) developed an algoritm ,which is initially used Konkar (1993), to calculate recipcrocal twist space and degrees of freedom under constraints. In our implementation we also used this method.

In addition, Kim and Chung (2003) presented an analytical formulation of reciprocal screws. They also analyzed application of reciprocal screws into robotic applications.

In general depending on the number of contacts, this Union Wrench Space Matrix WU which is mx6, where m is the number of contacts.

$$WU = \begin{bmatrix} \underline{\underline{W\ 1}} \\ \underline{\underline{W\ 2}} \\ \underline{\underline{-\ -\ -}} \\ Wm \end{bmatrix} \tag{3.7.3}$$

A grasp by a multifingered hand is said to be stable if **WU** has rank of 6. The manipulability is the rank of resultant net wrench space , represented by $\mathbf{W_n}$, projected to COG for each candidate virtual grasp.

In the stability, if the rank of the WU , is lower than 6, we need to analyze the reciprocal of WU which is a twist space T. The reciprocal space representation gives us the remaining degrees of freedom, which can be used by internal forces or disturbances , even exist under constraints of grasp. Therefore, we find reciprocal twist space by using reciprocity theorem of screw systems.

To find out the twist space,we use Konkar's algorithm which we restate briefly .

1- Form WU by stacking all contact wrench matrices
2- Apply Row Reduced Echelon Form to WU
3- Use Gauss-Jordan Elimination to find Null Space of RREF WU.
4- The Null Space ,if exists, is the Reciprocal Twist Space.

Hence,

The Stability is :

$$\textbf{Sta=rank(WU)-rank(T) or Sta = (6-Null(WU)} \qquad (3.7.4)$$

and its value varies between 0 to 6.

The Manipulability is:

$$\textbf{Manu=rank}(\mathbf{W_n}) \qquad (3.7.5)$$

and this varies between 0 to 6.

In our case, a grasp by a multifingered hand is said to be completely stable if WU has rank of 6 . If the grasp also has a manupulability of 6 ,then this form corresponds to force closure beacuse grasp can resist any arbitrary generalized force to be applied and can keep its original state.

3.8. Conclusion About The Resultant Wrench Representing Grasp Pattern.

In order to infer about which preshape will contribute the execution of the desired task, we developed a grasp respresentation pattern which is the Resultant Wrench applied by the robot hand on the object. In our analysis, we use virtual grasp meaning that, analysis is carried out on a 3D object model , in a simulation environment. The use of grasp wrench as a representation pattern is "Our Novel Method" and it is introduced first time in this thesis.

In classical kinematics, forward and reverse analyses of parallel manipulators are performed from a perpective for the manipulator itself.

This is also extended in constraint analysis and compliance and stiffness mappings which are widely used in literature. As we stated before, Duffy(1996), Hunt(1978) gives detailed forward and reverse analyses of parallel manipulators, analyzed for manipulator. Ciblak and Lipkin (1998) performed analyses about stiffness and compliance mappings using screw systems for parallel manipulators. In addition, Adams et al (2001) presented kinematic constraint analysis using reciprocal screw systems.

From the classical perspective, when a parallel n-limb manipulator exerts contacts on a rigid body, the resultant motion capability of the body is represented by resultant twist space which we already mentioned as reciprocal space of applied contact wrenches. And, the resultant contact wrenches are accepted as resisting force and moment exerted on body by the manipulator. This perspective and analysis methods are true and valid if object is under grasp and no post-grasp action is analyzed.

However, in dexterous manipulation, post grasp mobility is desired and it is also a requirement for the dexterous manipulation, without loosing the grasp. If we try to use twist space to analyze the post grasp mobility, we notice that for a stable grasp with a soft finger contact model, there is no hidden degrees freedom left and twist space is null. Therefore, the analysis for post grasp case must be carried out from another perspective. In case, at initial grasp with full rank 6 of WU, the resultant twist is null and hence deducing from this information, the object is fully constrained and not possible to move. Even, stiffness mapping yields results for very small changes.

However, when we look from the perspective of post grasp motion, resultant wrench for the object under grasp indicates possible post grasp motion capabilities which is not a direct indication but in a one-to one realation form which we will explain and use.

In order to explain the relation from post grasp motion perspective, we start to make forward kinematic analysis of hand and the grasped object together as n_parallel manipulator acting on a rigid body. As a result of the forward kinematic analysis, the resultant force is the net force acting on the body and the resultant moment is also net moment on the body. After having estimated the resultant force and moment ,which is wrench, we assume that object under grasp can be analyzed as if a free body in space under control of the resultant wrench. The net wrench on the free body contributes its momentum both in linear and angular terms. That is to say, body would have a motion along the direction of the force applied on and a rotoation around an axis along which a moment exists. In other words, the wrench contributes the motion of the body after wrench applied, in terms of determination of the direction of the motion. This can be tought as contribution to spatial displacement of the object after grasp. To extend, the resultant wrench is also a screw but not for the representation of spatial displacement space, but it could be transformed into spatial displacement form using a relation matrix. For our case of pattern representation, we assume the relation matrix is an identity matrix of 6, $\mathbf{I_6}$, in order to have one-to one relation between spatial displacement screw and the resultant wrench , for a correct mapping of translational and rotational components.

In other words, we can directly take the resultant wrench as a pattern representing post grasp mobility of a performed grasp. Remembering the fact that our task representation is also spatial displacement screw , we can compare relational effects of the resultant wrench after grasp to desired task. After grasping, the resultant wrench is in contribution to post grasp motion of the object which is a controlled motion .

3.9.Approximate Reasoning Using Fuzzy Logic

In the previous sections, we have formulated a wrench representation of the task as a spatial displacement and interpreted the resultant net wrench on the object as a pattern representing the virtual grasp. It is possible deduce that these two representations are semantically similar because they are represented by the same known classes, based on our conclusion at section 3.8. Hence, the matching degree between these two patterns can be analyzed in order to infer about suitability of the task and proposed preshape which is executed with a virtual grasp on object model. Therefore, we can utilize the matching degree, as a criteria for reasoning about task and preshape suitability.

Based on our conlusion given in section 3.8, we can foresee that the matching degree analysis can be handled as a pattern matching problem where matching between a sample and data are represented by a number of known classes . Pattern classes are represented by symbolic attributes or numerical representations, such as vectors.

For patterns represented in vector space notation, different forms of Norms, such as Euclidean Norms, can be defined as metrics to represent distance based matching. On the other hand, patterns represented symbolic attributes, similarity measures are defined, using attributes. Santini and Jain (1999) gives definitions on similarity measures, in their work .

In this work , we make use of set theoretic similarity as a similarity measure , such as : Given two sets A and B defined at universe U. The similarity between A,B is given by eqn 3.9.1

$$s(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|} \qquad (3.9.1)$$

where $|\cdot\cdot|$ denotes cardinality of the set.

One can think that matching analysis can be performed by methods like L_x Norms because our patterns are represented in vector notation . However, referring to our conlusion at section 3.8 again, we remind the fact that the task is a representation for desired spatial displacement of object and grasp wrench is an indirect representation for controlled motion of object under grasp. The grasp wrench is only a relation to possible spatial displacement of object, although it is expressed in the common form with the task . Hence, we should look for some symbolic representation of our patterns so that we can utilize this relation. Hence, we should use similarity based matching methods. In addition, to avoid possible errors which could originate from estimations, approximations, we need approximate reasoning methods to infer about matching degree of our patterns.

In order to fullfill our requirements for symbolic representation, we decide to use fuzzy set representation of our patterns. By using fuzzy logic, it is both possible to use relations and tolerate errors. We assume an implicit fuzzy relation matrix which is an identity matrix, to express relation between the spatial displacement of object and the grasp wrench. Therefore, we can use the fuzzy sets obtained for task and fuzzy sets obtained for grasp wrench , to analyze pattern matching. Yeung and Tsang (1997) gives a comparative survey on fuzzy similarity methods. Yoshikawa and Nishimura(1999) evaluated a comparative analysis on various mathematical similarity indices between fuzzy sets , by carrying certain tests. Among these methods, most remarkable ones are “Fuzzy Distance Based Similarity” and “Approaching Degree Index Evaluation”, according to our rough categorization of the matching methods. Next, we will explain fuzzy vectors which we would use in fuzzfication of our wrench patterns.

A Universe U can be partitioned by nonintersecting features ,represented by N fuzzy sets with Membership values . Any variable x can be represented in U by fuzzy sets. The fuzzy set with Membership values, including all features, representing variable x is called a “Fuzzy Vector”. Fuzzy Vector for x is shown by F(x).

$$F(x) = \{\mu_1, \mu_2, \ldots\ldots\ldots \ldots \mu_N\} \qquad (3.9.2)$$

Similarly , a vector V represented in M dimensional space, having common feature space , can also be represented by Fuzzy Matrix of MxN, where each row of the matrix is a Fuzzy Vector for a dimensional element in V. Our wrenches are screws which are vectors in general are fuzzified using m-Fuzzy Vector or Fuzzy Matrix of MxN. Wrenches are 6 dimensional vectors, so using the following fuzzy set indicated in figure 3.18, we obtain 6-Fuzzy Vectors to represent a wrench, in figure 3.19.

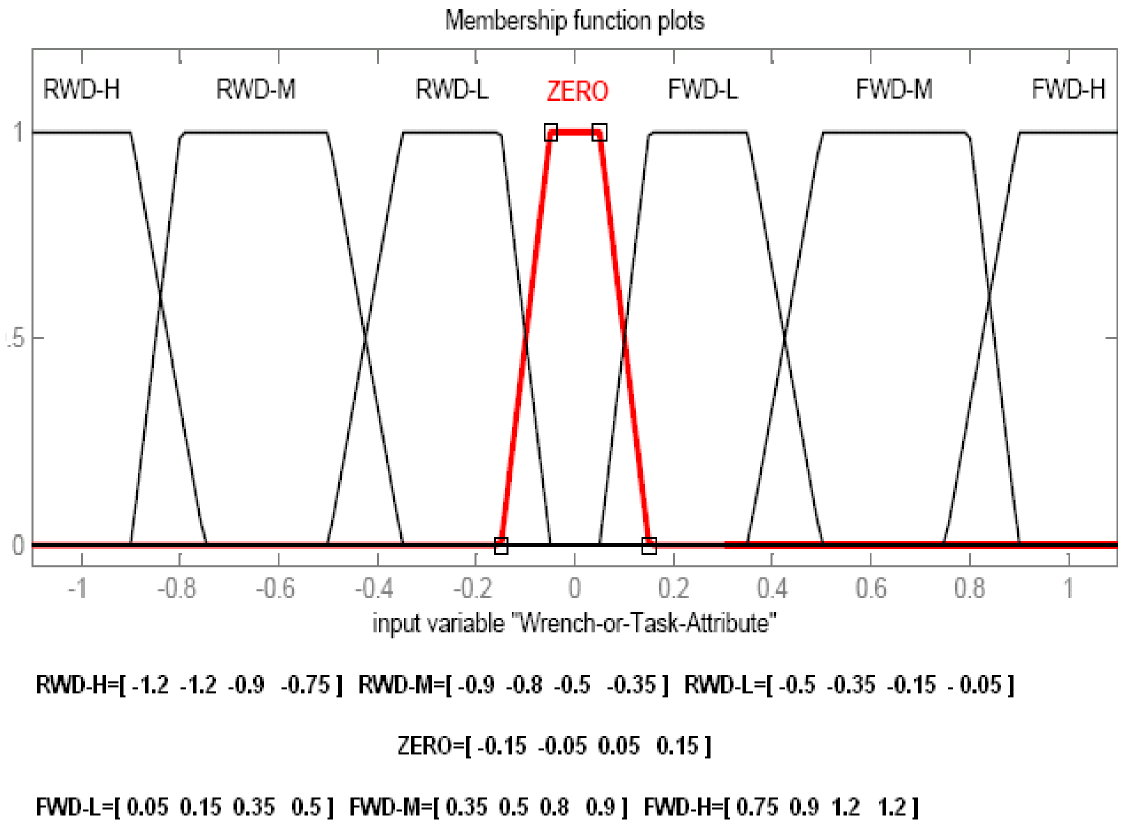
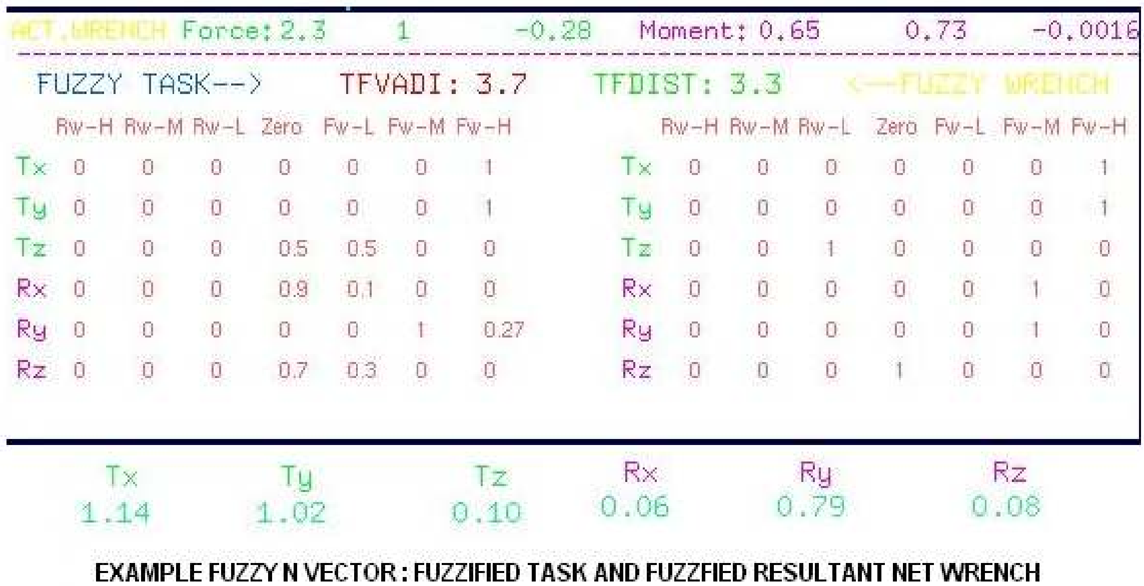


Figure 3.18. Fuzzy Set for Grasp Wrench or Tasks.


ACT.WRENCH Force: 2.3 1 -0.28 Moment: 0.65 0.73 -0.0016

FUZZY TASK--> TFVADI: 3.7 TFDIST: 3.3 <--FUZZY WRENCH

	Rw-H	Rw-M	Rw-L	Zero	Fw-L	Fw-M	Fw-H
Tx	0	0	0	0	0	0	1
Ty	0	0	0	0	0	0	1
Tz	0	0	0	0.5	0.5	0	0
Rx	0	0	0	0.9	0.1	0	0
Ry	0	0	0	0	0	1	0.27
Rz	0	0	0	0.7	0.3	0	0

	Rw-H	Rw-M	Rw-L	Zero	Fw-L	Fw-M	Fw-H
Tx	0	0	0	0	0	0	1
Ty	0	0	0	0	0	0	1
Tz	0	0	1	0	0	0	0
Rx	0	0	0	0	0	1	0
Ry	0	0	0	0	0	1	0
Rz	0	0	0	1	0	0	0

Tx	Ty	Tz	Rx	Ry	Rz
1.14	1.02	0.10	0.06	0.79	0.08

EXAMPLE FUZZY N VECTOR : FUZZIFIED TASK AND FUZZFIED RESULTANT NET WRENCH

Figure 3.19. Fuzzified Grasp Wrench and Fuzzifed Tasks, as 6-Fuzzy Vectors.

Given two Fuzzy sets or vectors of dimension N, A ,B, defined on Universe U:

$$A=\left\{\frac{\mu_{a1}}{x_1},\frac{\mu_{a2}}{x_2},\frac{\mu_{a3}}{x_3},.......,\frac{\mu_{an}}{x_n}\right\};\ B=\left\{\frac{\mu_{b1}}{x_1},\frac{\mu_{b2}}{x_2},\frac{\mu_{b3}}{x_3},.......,\frac{\mu_{bn}}{x_n}\right\} \qquad (3.9.3)$$

Fuzzy Distance Based Similarity, is defines as:

$$D_2(A,B)\ \underline{\underline{\Delta}}\sqrt{\sum_{i=1}^{n}\left|\mu_A(x_i)-\mu_B(x_i)\right|^2} \qquad (3.9.4)$$

which is Fuzzy Euclidean Metric, L_2 Norm.

Since our pattern is a 6-Fuzzy Vectors, when we apply D operator on our pattern , we obtain 6 values , one for each dimension. In order to combine these values into a general index , we define Total Fuzzy Distance Between M-Fuzzy Vectors.

$$TFDist\ (Q,W_n)\ \underline{\underline{\Delta}}\sum_{i=1}^{M}D_2(q_i,w_i) \qquad (3.9.5)$$

where Q represents the task and Wn represents the resultant wrench of grasp.

$$Q=\{\overline{\overline{\overset{T_x,T_y,T_z,R_x,R_y,R_z}{q_1,q_2,q_3,q_4,q_5,q_6}}}\}\quad W_n=\{\overline{\overline{\overset{T_x,T_y,T_z,R_x,R_y,R_z}{w_1,w_2,w_3,w_4,w_5,w_6}}}\} \qquad (3.9.6)$$

In the literature, Wang (1983), derived the “Approaching Degree Index” which measures exactness or similarity between two fuzzy sets or vectors. ADI(A,B) gives a scalar between 0 and 1. Ross(1995) gives examples for evaluation of ADI.

$$ADI\ (A,B)\underline{\underline{\Delta}}(A\bullet B)\wedge(\overline{A\oplus B}) \qquad (3.9.7)$$

$$(A\bullet B)\equiv(A\bullet B\ )=\bigvee_{i}^{n}\ (a_i\wedge b_i) \qquad \text{, Fuzzy Inner Product} \qquad (3.9.8)$$

$$(A\oplus B)\equiv(A\oplus B\ )=\bigwedge_{i}^{n}\ (a_i\vee b_i) \qquad \text{,Fuzzy Outer Product} \qquad (3.9.9)$$

.

Using Fuzzy Algebra and De Morgan Properties,

$$ADI(T,W) = (T \bullet W) \wedge \overline{(T \oplus W)} = (T \bullet W\ ) \wedge (\overline{T} \bullet \overline{W}\ ) \qquad (3.9.10)$$

Since our pattern is a 6-Fuzzy Vectors, when we apply ADI operator on our pattern , we obtain 6 values , one for each dimension. In order to combine these values into a general index , we define Total Fuzzy Vector Approaching Degree Index between M-Fuzzy Vectors.

$$TFVADI\ \ (Q, W_n)\ \underline{\underline{\Delta}} \sum_{i=1}^{M} ADI\ (q_i, w_i) \qquad (3.9.11)$$

where Q represents the task and Wn represents the resultant wrench of grasp.
In order to further explain TFVADI, we interpret the figure 3.19 .
In the figure , wrench is given as W=[2.3 1 –0.28 0.65 0.73 –0.016]
and task is given as Q=[1.14 1.02 0.10 0.06 0.79 0.05]. The figure also gives the 6-Fuzzy Vectors for each screw as labeled. Using the equations 3.9.11 and 3.9.10 successively, TFVADI is found as 3.7. similarly TDIST as 3.3 using equation 3.9.5

In our intelligent controller structure implementation , we get results for approach value as symbolic fuzzy set attributes. However ,in virtual grasp execution, we need numerical values , hence defuzzification is required to obtain crisp numerical values. We use the same Fuzzy Set given in figure 3.9.1 for defuzzification as follows. For an approach value expressed in fuzzy set attribute index, we find and use the corresponding membership function related to fuzzy set attribute. For instance, if approach is given as Rwd-M, then MF for Rwd-H is used. Then, we estimate the numerical value mapping to the center of the membership function which is a Trapezoid upper level. In this way we obtained a crips numerical representation for fuzzy set attribute.

As a result, for our pattern matching problem of the task and the grasp wrench, we have introduced and enhanced two similarty measures as matching methods to be used in conjunction. The use of the methods are implemented in our Intelligent Controller Structure.

3.10.Use of Artificial Neural Network for Preshape Classification .

After the analysis of task and kinematics of virtual grasp, one can conclude that determination of approach plane could be performed by means of heuristic searches such as “Hill Climbing” or “Beam Search” methods, using the approximate reasoning criteria that we developed at section 3.9. Details of the herustic search methods could be found at textbook of Kubat et al (1996) . Heuristic searches could also be performed by an Artificial Neural Networks, acting as a classifier. In the balance of the this thesis work we estimate the approach plane and the relevant approach value by a Multilayer Neural Network, given the task attributes and object properties as input. In literature, Multilayer Neural Networks such as Multi Layer Perceptrons have been widely used as classifier. Brown (1996) gives a tutorial about Multi Layered Perceptron. In general, Haykin (1994), in his textbook, presents foundations of Neural Networks. According to literature, for a classification work with N-nonlinearly seperable classes, a Multi Layered Perceptron Network is required at least 1-hidden layer , an input and an output layer with nonlinear activation functions.

In figure 3.20, the structure of an Artificial Neuron and a Multilayered Artificial Neural Network (ANN) is given. In our case, we implement a 3-layer MLP (Input, Ouput and a Hidden Layer) as ANN, in which nonlinear activation function f(.) is logaritmhic sigmoid, and a Back Propagation algorithm is used for training. An example for the structure of Input and Output vectors are illustrated in figure 3.21.

At this point, I would like to explain structure of input and output vectors of the ANN in figure 3.21. The input vector to ANN is composed of by two parts. In the first part, we use task represented as wrench system formed to represent a spatial displacement equipped with desired stability and manupulability values. In the second part, we use object model attributes, such as size attributes, number of vertices, number of mesh triangles of its modeling.

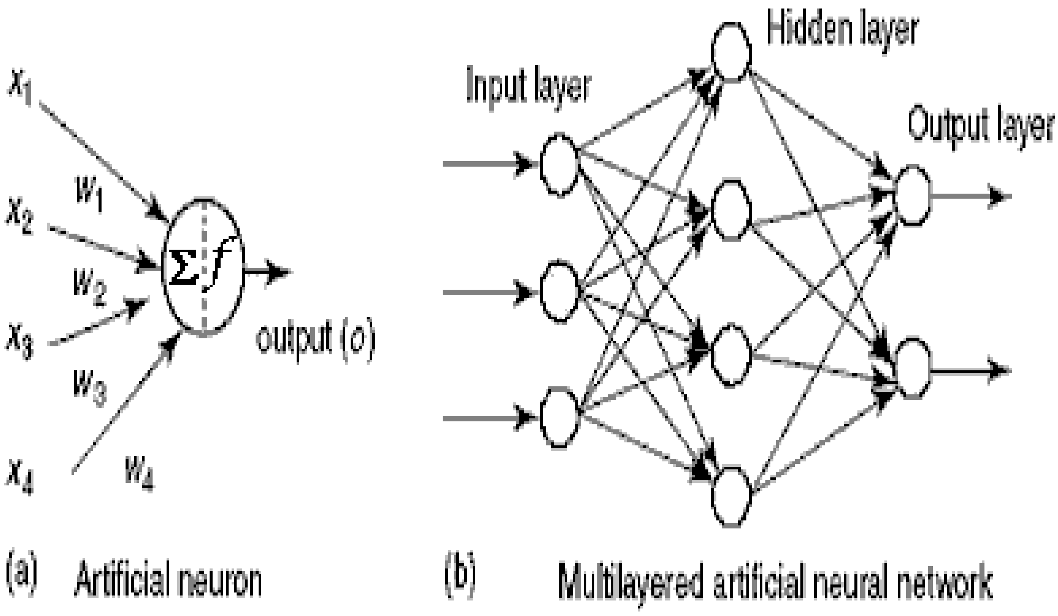


Figure 3.20 Artificial Neuron and Multilayered Artificial Neural Network.

NEURAL NETWORK INPUT VECTOR STRUCTURE

TASK ATTRIBUTES OBJ:CUBE OBJECT ATTRIBUTES

X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
Tx	Ty	Tz	Rx	Ry	Rz	Sta	Manu	Width	Height	Depth	# Vert.	# Vn	# Faces	# Triang.
0,900	-1,000	0,100	0,050	-0,790	0,091	1,000	0,833	0,333	0,333	0,333	0,667	0,500	1,000	0,012
-1,000	0,900	-0,100	-0,810	0,051	0,110	1,000	0,833	0,333	0,333	0,333	0,667	0,500	1,000	0,012
0,300	0,900	-1,000	0,090	0,049	-0,790	1,000	0,833	0,333	0,333	0,333	0,667	0,500	1,000	0,012
-0,100	-1,000	0,900	-0,050	-0,790	0,051	1,000	0,667	0,333	0,333	0,333	0,667	0,500	1,000	0,012

NEURAL NETWORK OUTPUT VECTOR STRUCTURE

Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
0,00	0,00	0,00	1,00	0,00	1,00	1,00	1,00	0,00
0,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	0,00
0,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00	0,00
0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	0,00

Figure 3.21 The Structure of Input and Output Vectors of ANN.

The ouput vector of the ANN is a binary output vector composed of approach plane number , fuzzy set attribute index for approach value and estimation for number of fingers to be involved in grasp. Remember that, in our implementation, the number of finger is determined by the rule based heuristic according to object size determined with respect to transversal axis of the grasp preshape . Eventhough we do not use it for current implementation, we have included the number of finger estimation in Neural Network because we think that it might be useful for a possible future extension of our work. In figure 3.22, we give the decomposition of the structure of the output vector.

PLANE INDEX

Y0	Y1	Y2	Y3	PLANE
0,00	0,00	0,00	0,00	NA
0,00	0,00	0,00	1,00	1
0,00	0,00	1,00	0,00	2
0,00	0,00	1,00	1,00	3
0,00	1,00	0,00	0,00	4
0,00	1,00	0,00	1,00	5
0,00	1,00	1,00	0,00	6
0,00	1,00	1,00	1,00	7
1,00	0,00	0,00	0,00	8
1,00	0,00	0,00	1,00	9
1,00	0,00	1,00	0,00	10
1,00	0,00	1,00	1,00	11
1,00	1,00	0,00	0,00	12

DISTANCE FROM COG

Y4	Y5	Y6	FQ	IDX
0,00	0,00	0,00	NA	NA
0,00	0,00	1,00	Rw-H	1
0,00	1,00	0,00	Rw-M	2
0,00	1,00	1,00	Rw-L	3
1,00	0,00	0,00	ZERO	4
1,00	0,00	1,00	Fw-L	5
1,00	1,00	0,00	Fw-M	6
1,00	1,00	1,00	Fw-H	7

OF FINGERS

Y7	Y8	#VF
0,00	0,00	NA
0,00	1,00	1,00
1,00	0,00	2,00
1,00	1,00	3,00

Figure 3.22 Decomposition of The Structure of Output Vector of ANN.

The training data and other parameters are given in appendix A of the thesis. According to our parametric analysis of ANN, we found that 45 hidden node is the optimum value for hidden layer. The details of various training sets and surface analysis results for optimation of the ANN architecture presented in appendix A. The ANN is integrated into our Intelligent Controller, as we illustrated in section 3.1.

4.COMPUTER IMPLEMENTATION

4.1 Overview of Software Implementation

The designed controller architecture is implemented in a software environment of 3D graphics. The software source code is written in C++ programming language and using OpenGL Libraries, under Windows XP. The resultant controller is a Win32 application. The controller uses data files which are previously formed by 3D Graphics editors such as 3D Canvas in our case, and exported to in Wavefront object format, for regeneration of object models. Figure 4.1 is a screen snapshot from our Controller.

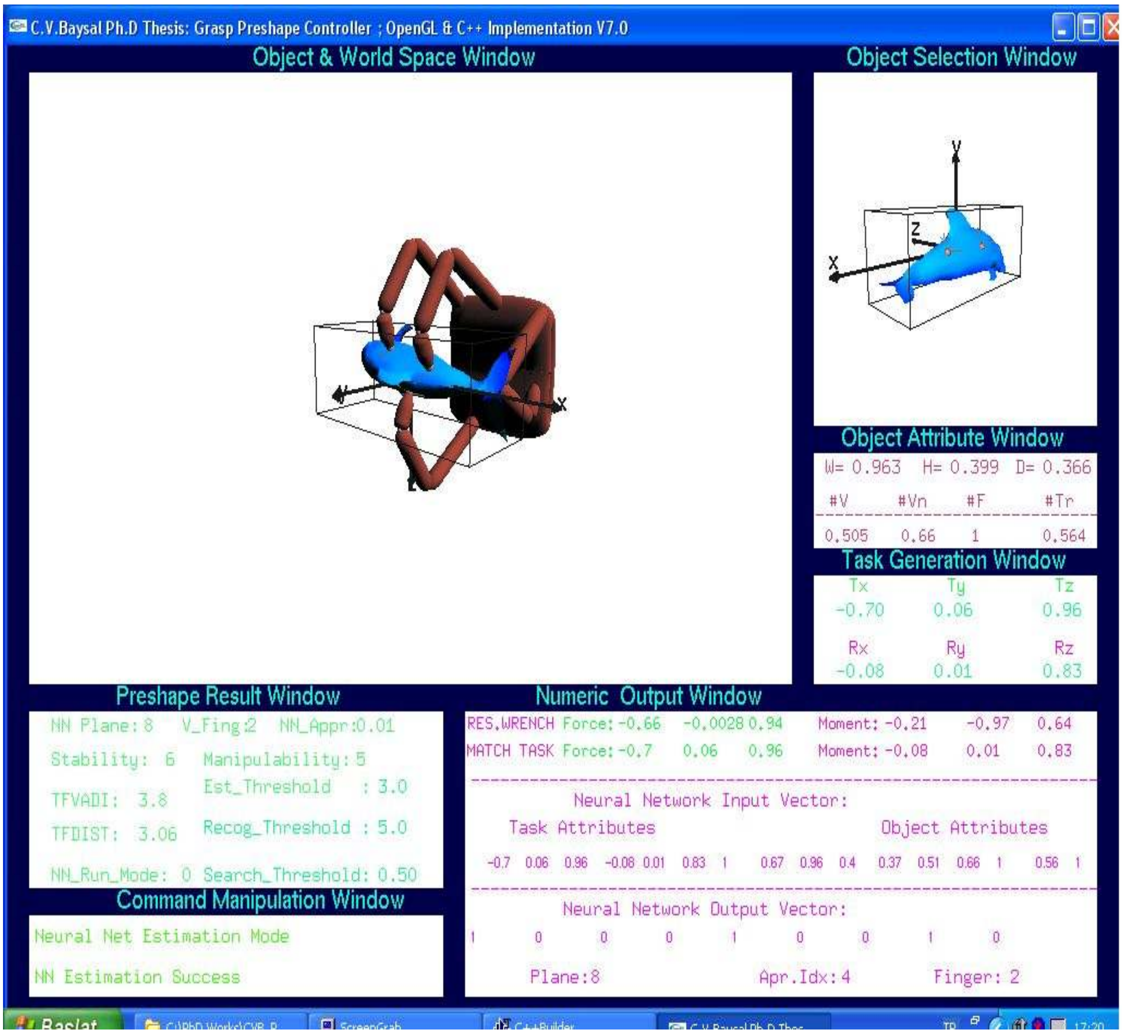


Figure 4.1. The Screen Snapshot from Our Controller Software.

I would like explain the screen snapshoot of our controller to give an idea about user interface. As seen from the figure, there are dedicated sub windows such as Command Manipulation, Preshape Result, Task Generation, Numeric Output and Object Selection. Object selection is done by the user via pull down menu using mouse. Task Generation is done by manual adjustment of each translational and rotational components by using mouse. Command Manipulation window is used to initialize , to select the modes , and to start–stop the controller execution. Numeric Result window updates its display format according to controller execution mode. Preshape Result window is the main result window, displaying resultant preshape in terms of Approach Plane no , Approach Value, Number of Fingers, Stability and Manipulability values, Properties for Reciprocal Twist Space and Thresholds for crictics. The changes to Thresholds are performed by mouse rolls in this window .

The virtual contact and grasp formation is done using collision based exact 3D geometric contact detection algorithms. Therefore, virtual finger contact points, contact surface normals are real 3D information which obtained from object model data analysis for our controller . The object is stored in a dynamic data structure after file access for processing, which is a linked list. The software implementation is made quite less dependent on operating system; hence it is portable to different platforms since both C++ and OpenGL are portable to many platforms such as Linux, Unix, Sun etc.

4.2 Software Implementation of Controller Parts

The Controller Software has following program parts : Main Program , OpenGL , ObjectModel Acquisition, General Mathematical Functions , Reciprocal Twist Space Analyzer, Hand Simulation , Neural Network Implementation, Expert Preshape Executor, Fuzzification and Defuzzification routines. In order to give an idea about code we developed, we state that our own created program code size is 117 A4 pages. All routines run at the shell of GLUT API. .Figure 4.2 shows these parts as blocks. We explain their functions next paragraph.

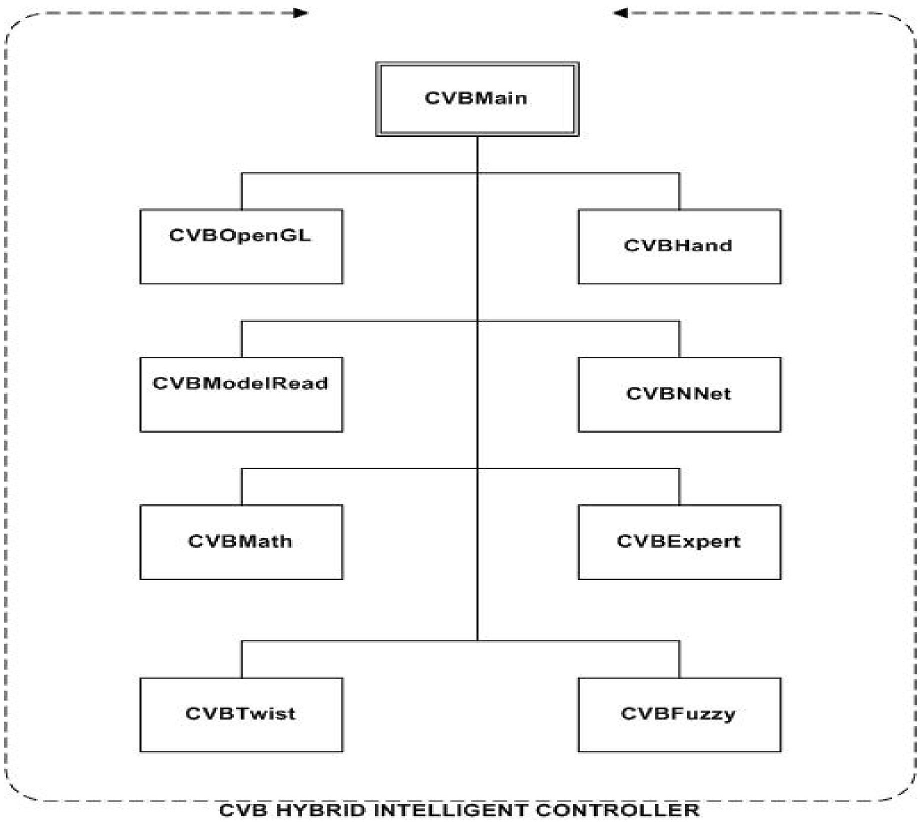


Figure 4.2. The Controller Software Blocks.

The blocks shown in figure 4.2 are C++ programs routines. The “Main Program” is the routine responsible for User Interface and GLUT API shell and it is also used as integration of all routines. “OpenGL” routine is for some specific OpenGL function for visualization. “ModelRead” routine is used to model acquisition of object from disk file and partially implements environment sensing element to estimate object attributes. “Math” routine is the general mathematical routine where all common functions used by other routines are implemented such as norms, poinsot projection, vector and matrix algebra, wrench estimations etc. “Twist” routine is used for implementation of reciprocal space and grasp performance measures. “Hand” routine is derived from our own Hand simulation toolbox and used to demonstrate kinematic execution of the result from controller for Meta Control. “ Neural Net” routine is based on our own Neural Network Development Toolbox and used for NN execution and Learning Modes.

"Expert" routine is the one part of the Performance Element for determining number of fingers , VF, and "Dynamic Depth" values for transversal axis size estimation as well as checking for "Opposition Space Occurrence" .This routine acts as Kinematic estimator as shown in our Controller architecture. Rule based estimations are performed by this routine so it is called expert.

"Fuzzy " routine is used for fuzzification and defuzzification as well as implementation of approximate reasoning by Fuzzy Logic via similarity measures. The Blocks which are possible to implement on MATLAB are also cross checked using the implementation results from MATLAB built in Libraries such as Reciprocal Twist Space, Fuzzy Logic and Neural Network routines.

In addition to above routines for the Controller implementation, we also developed code for some auxillary programs as our toolbox which are Neural Network Design and Analysis Toolbox , Hand Simulator to analyze Forward and Inverse Kinematics of Hand Simulation, Object Task Grasp Workspace Analyzer. The core routines from these toolboxes are also utilized in Controller Software. We would like explain here utilization of Neural Network Toolbox software, of which ,screen snapshot is given at figure 4.3.

The Neural Net Toolbox is used for initial construction of Neural Network structure. We preffered an external tool because some features of toolbox are only required when the parametric estimations for Neural Network design performed. Hence, IO training, normal execution routines are only implemented in the controller software. The Neural Net routine of the controller share same training data files and access to weight storage files constructed by toolbox. By using this toolbox , we designed the Neural Network and performed a parametric analysis to find optimal parameters for Neural Network in our implementation. We also checked our results in MATLAB.

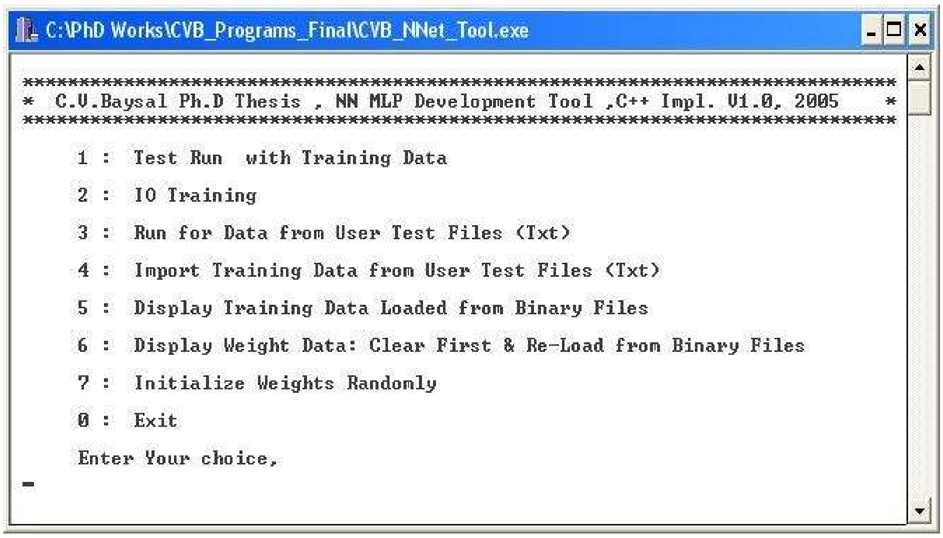


Figure 4.3. The Developed Neural Netwok Toolbox .

OpenGL routines are utilized from GLU, GLUT,GLM,GLAUX precompiled toolboxes. These routines provides, basic IO and interfacing function utilizing OpenGL low level codes available in the OpenGL.dll , which is exist in the Windows XP operating system or Linux OS versions. The details of OpenGL programming can be further referenced in OpenGL Manuals by Silicon Graphics Inc, namely “OpenGL Redbook” (1990) , “OpenGL Bluebook” (1996). Object Model Acqusition codes are based on Web OpenGL tutorials from Nate Robbins at OpenGL web site. The Fuzzy Logic and Neural Network codes are based on available C++ codes in many Fuzzy Packages and textbook of Welstead (1994) .

4.3 Algortihms as Flow Diagrams of Controller Software

Algorithms for the Controller Execution are introduced in this section. The Multistate Decision algorithm is given first and algortithms for execution states are given. Finally, the algortihm for Fuzzy Explorer is given .

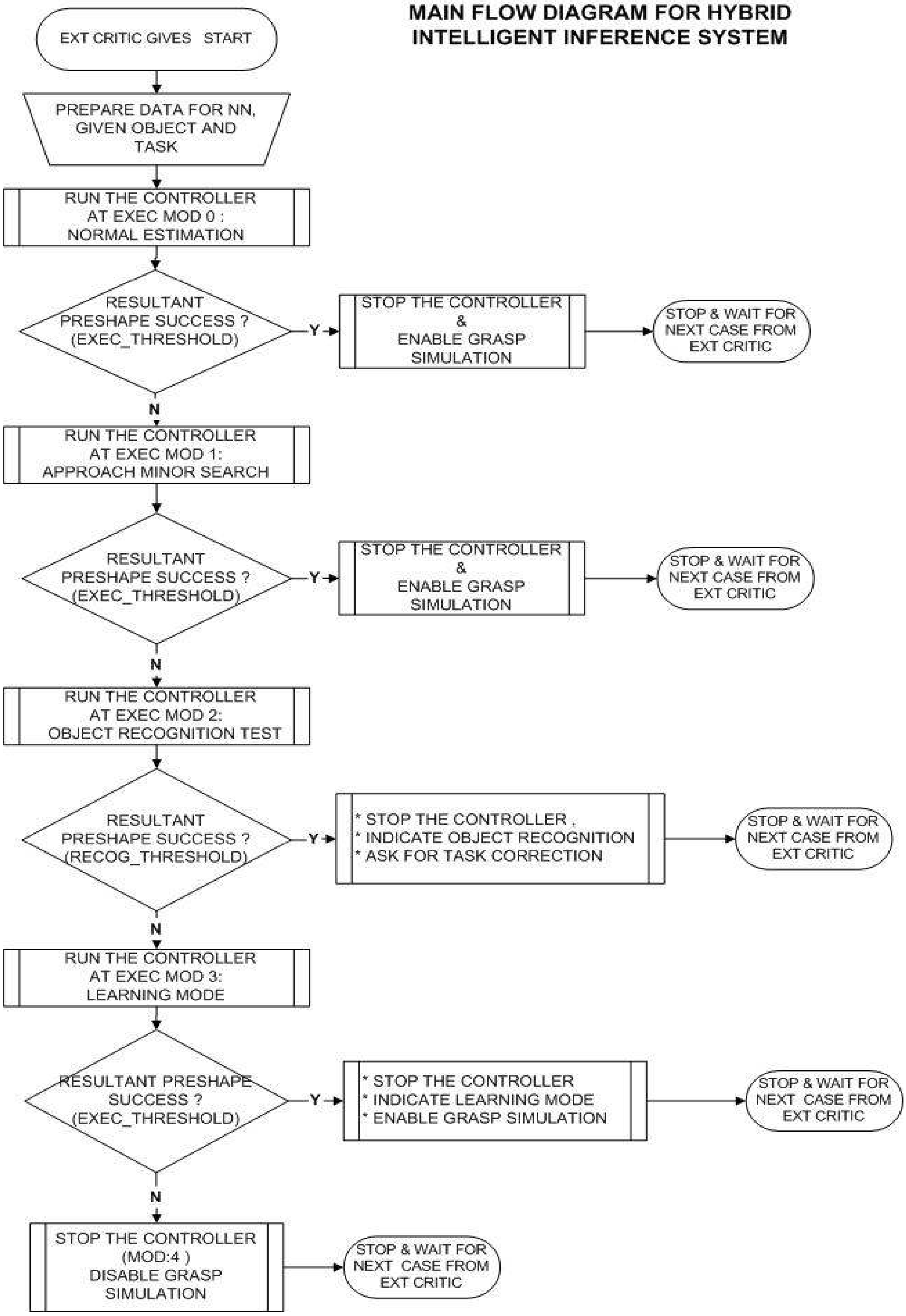


Figure 4.4. The Multistate Decision Main Algorithm .

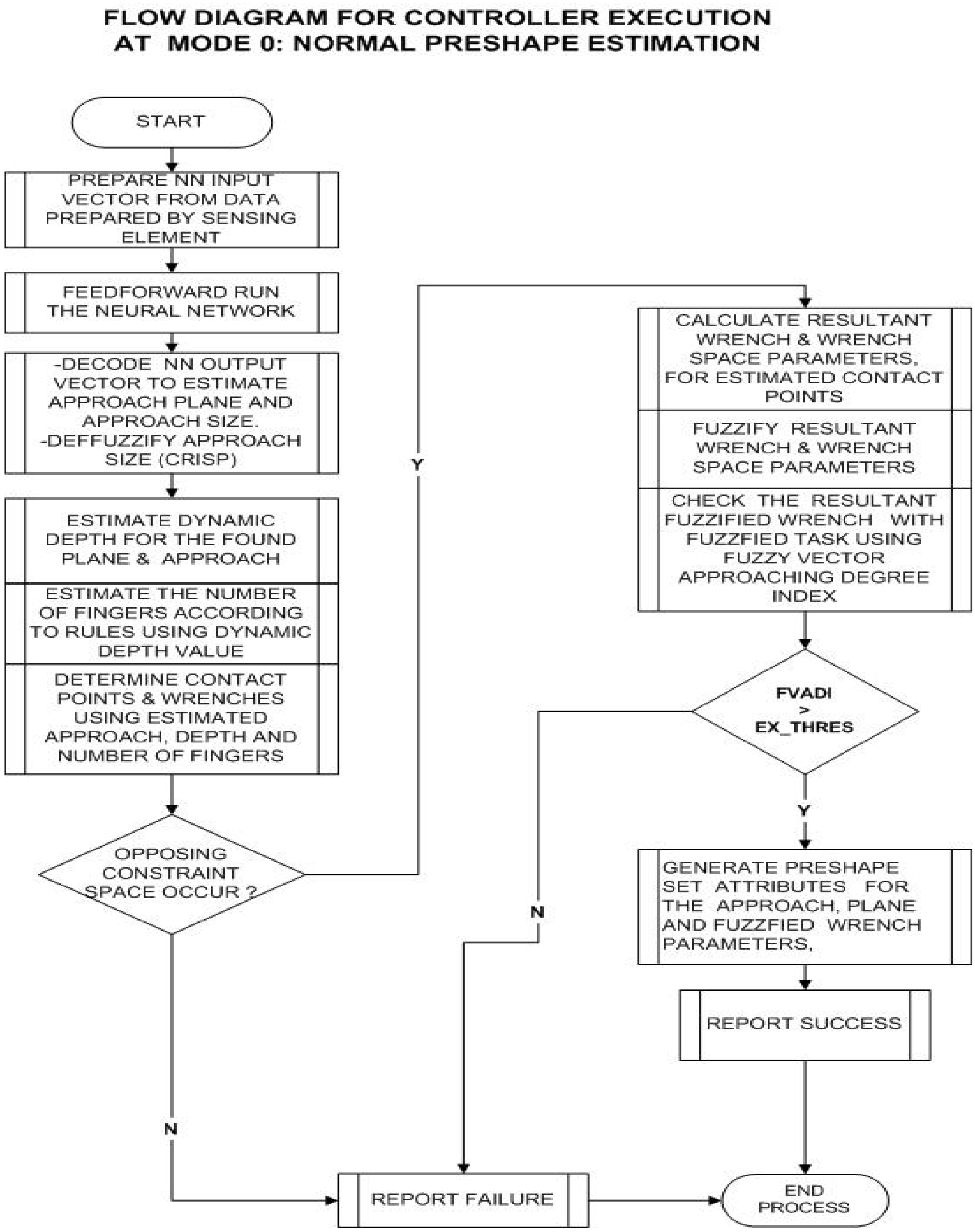


Figure 4.5. The Algorithm for the Controller at Execution Mode 0.

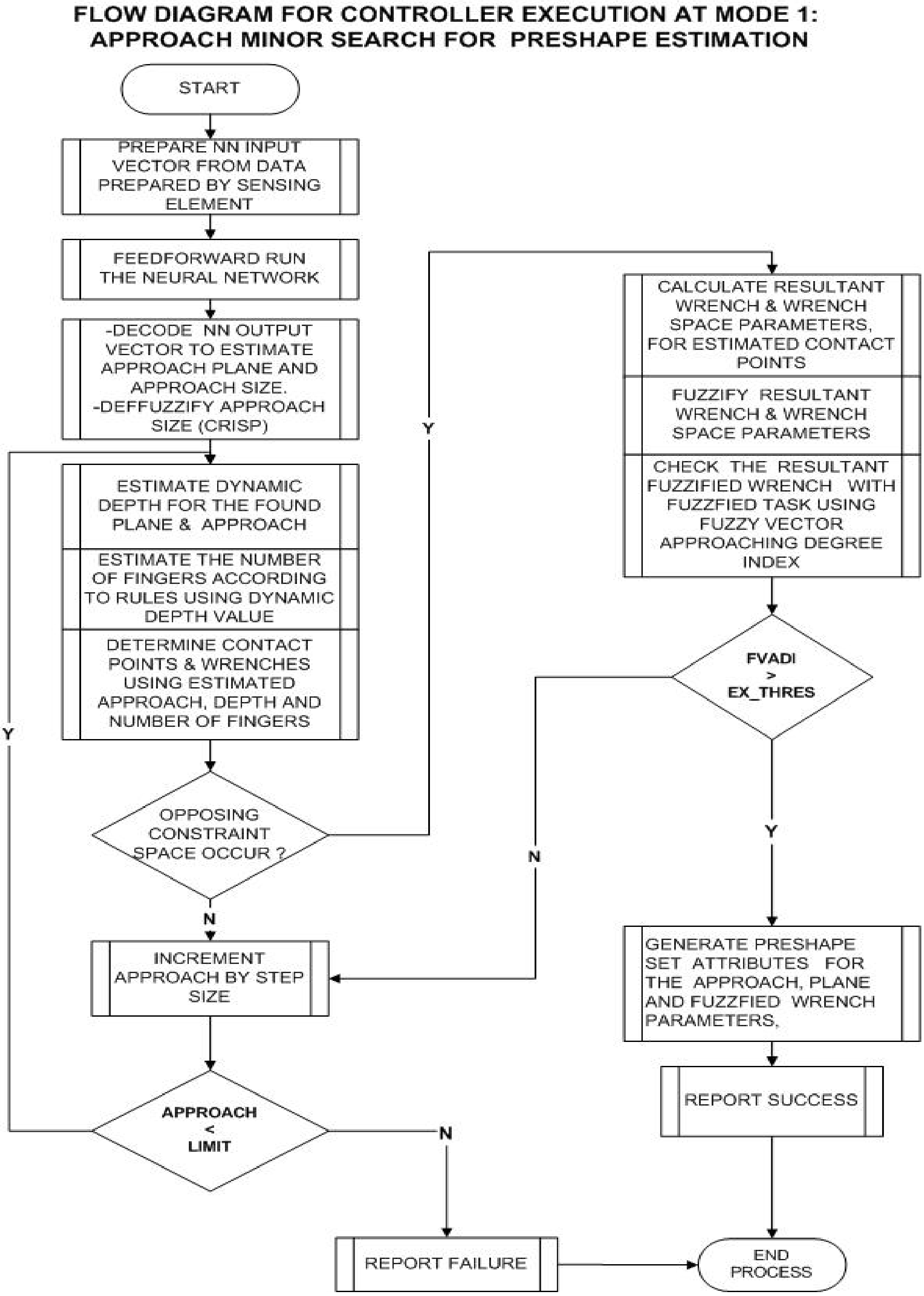


Figure 4.6 .The Algorithm for the Controller at Execution Mode 1.

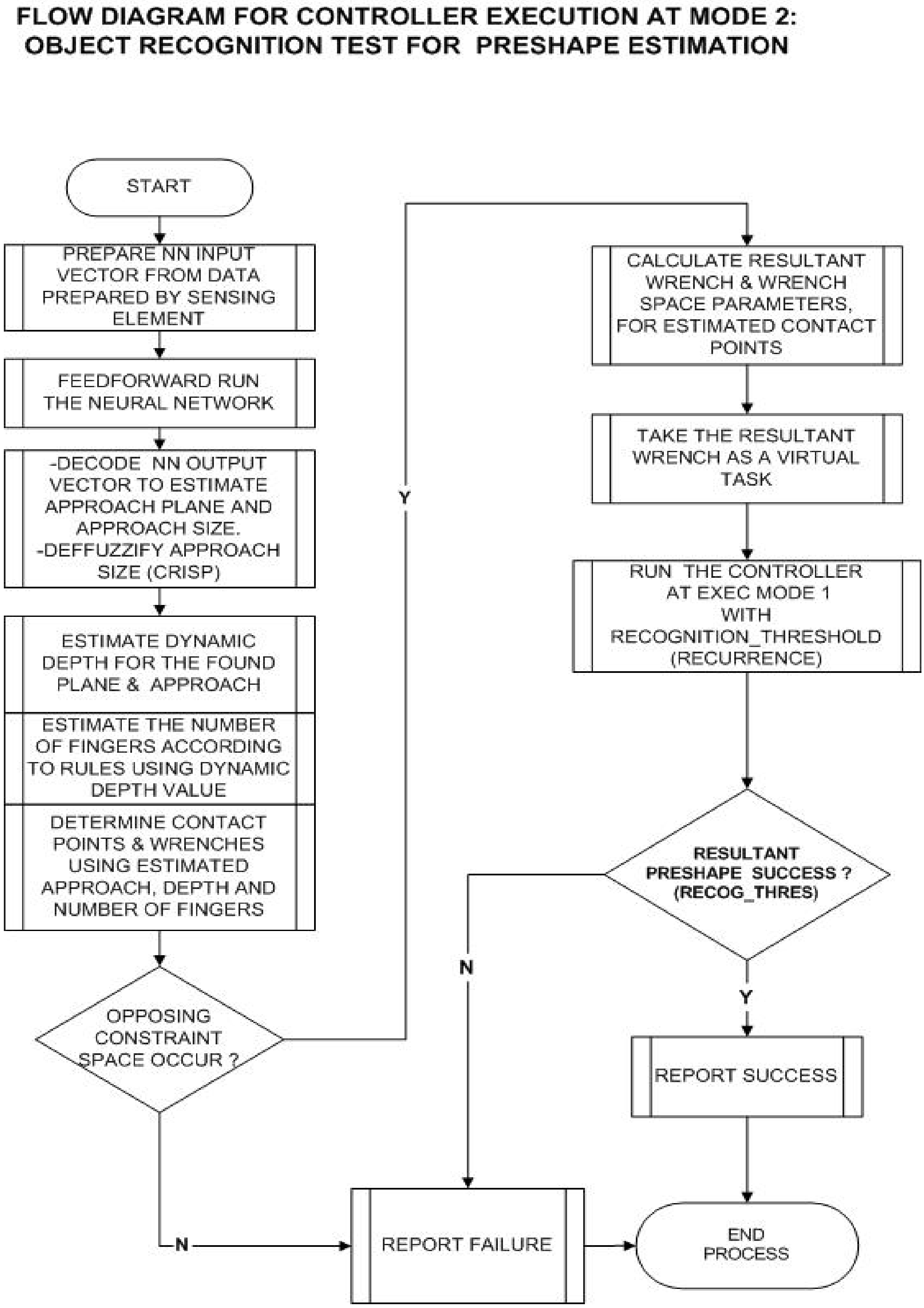


Figure 4.7. The Algorithm for the Controller at Execution Mode 2.

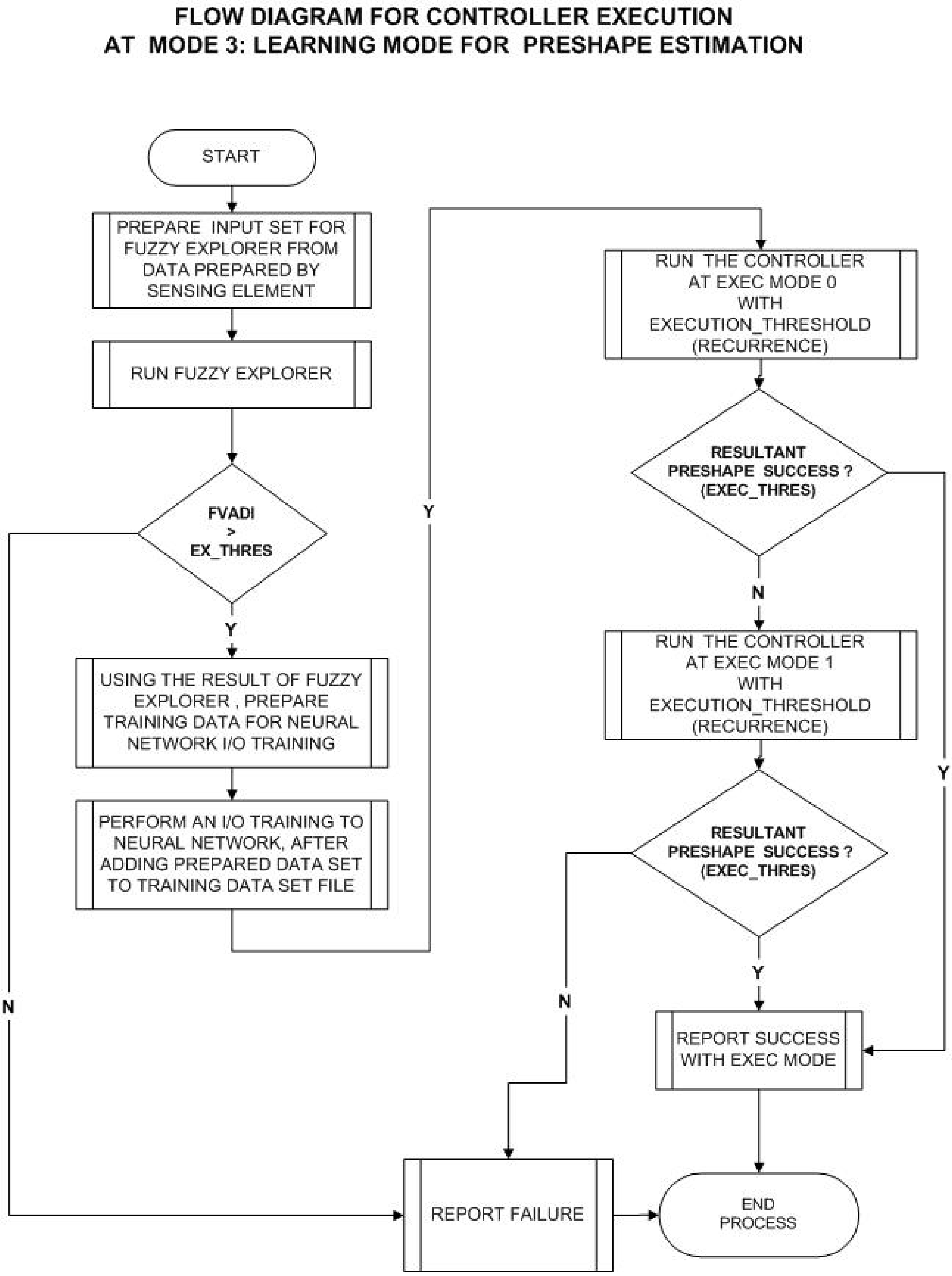


Figure 4.8.The Algorithm for the Controller at Execution Mode 3.

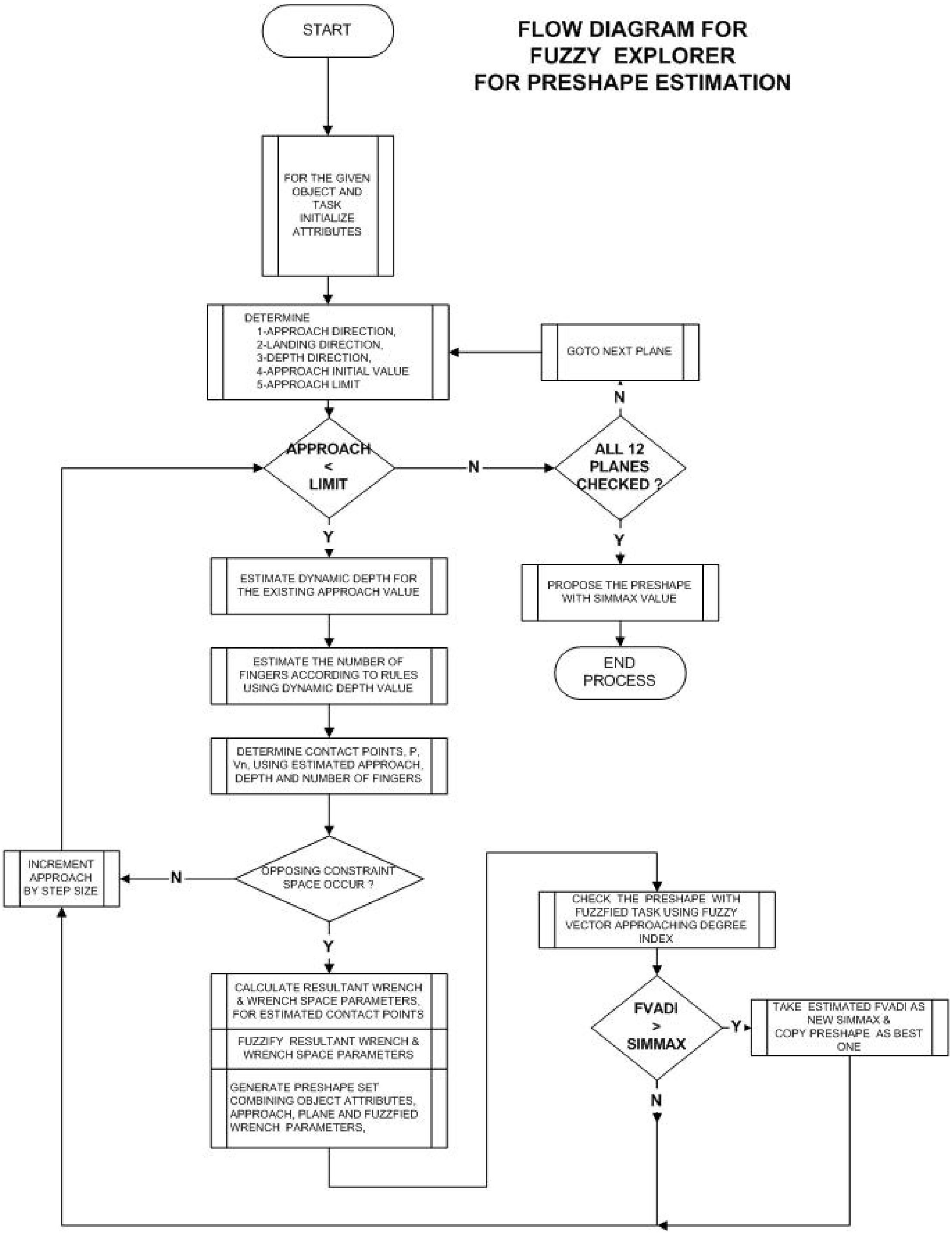


Figure 4.8. The Algorithm for the Fuzzy Logic Explorer .

4.4. Hand Simulation.

In order to demonstrate the performance of our controller in a better environment, in a way, to improve our controller, we have implemented a basic Robot Hand Simulation toolbox only with kinematics, in OpenGL and C++ software . Our aim for this simulation is to visualize the end effector selecting a preshape and grasping a given object and task defined by an External Critic. In this work , we did not aimed to make a perfect Hand Simulation which is subject of another study. In literature there are works for simulation of virtual environments. In this thesis work , The Hand is a 4 fingered 16 DOF hand model which is similar to UTAH-MIT Hand. Each finger is implemented as planar 3 link manipulator (2 link manipulator with an end effector) with respect to hand base frame, wrist. Hand Base frame adjusts its position according to the approach plane index. Input for the hand simulator is the approach plane and the estimated contact locations. In the hand simulation, we test exact collision of fingertip with the estimated contact surface. In our implementation, we assume that hand base frame can move around object within any path or without any obstacle avoidance .

Hand Direct Kinematics is implemented using Denavit Hartenberg method and Inverse Kinematics is implemented by using Geometrical Approach. The Hand Constraints are similar to Denavit -Hartenberg notation. The details of these methods can be found in Craig (1986) and Fu, Gonzales, Lee(1987) robotic textbooks.

The hand simulation is tested by checking Joint Cooridinates and Constraint limits, for its efficiency and accuracy. Although, it basically gives satisfactory results, however, it still needs improvements. Since, the main goal of this thesis is to implement the Preshape Controller, not to implement Hand Simulation. Figure 4.9 is a screen snapshot from Hand simulation tool to demonstrate given IK parameters, the Simulation performs accordingly by giving Kinematic Joint Parameters.

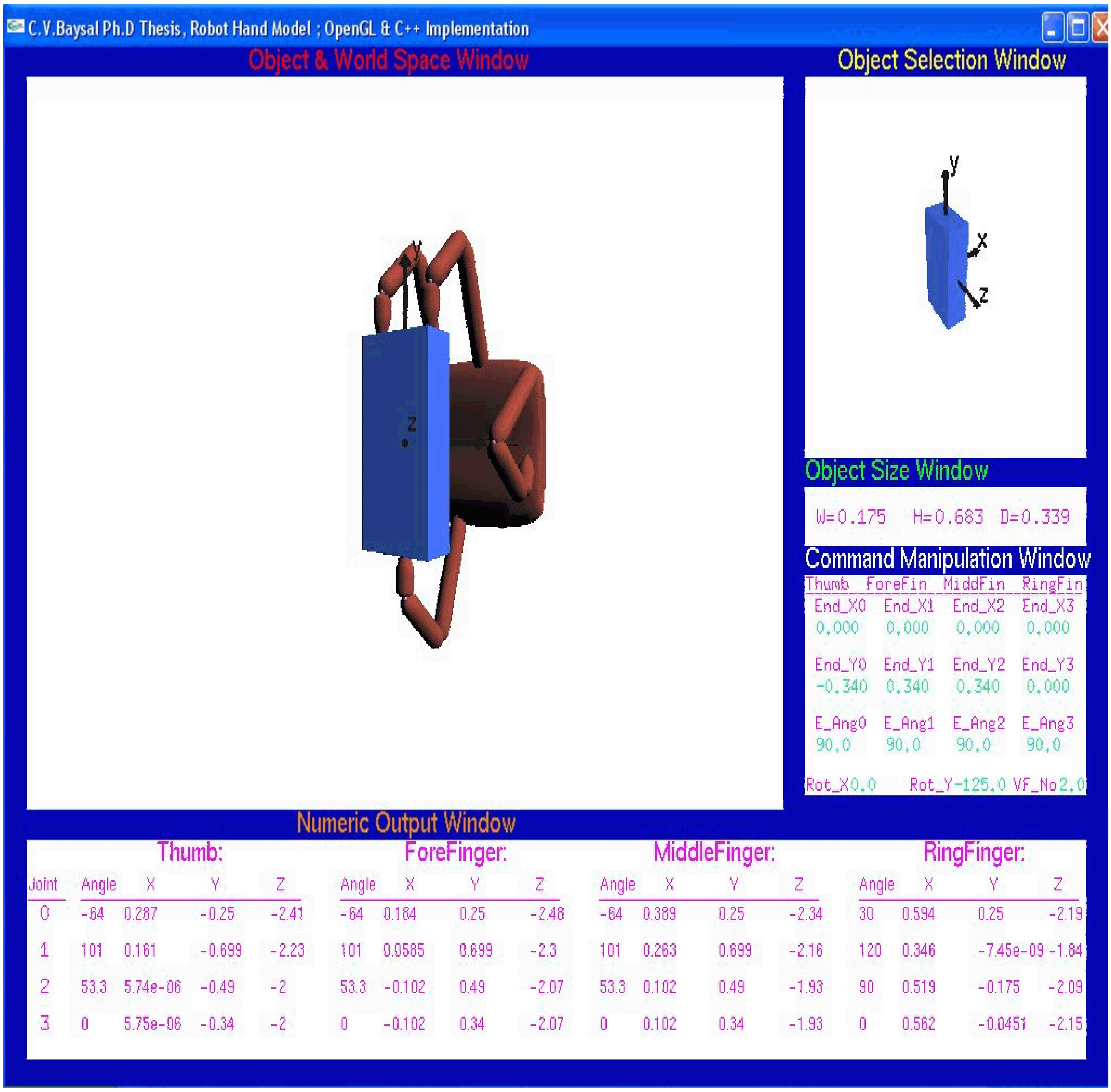


Figure 4.9.The Screen Snapshot from Our Hand Simulation Software Tool.

5.RESULTS AND DISCUSSIONS

In this work, the results of the controller are outputs to Meta Control as both in terms of Grasp Preshapes and Grasp Simulation. We tested our Intelligent Controller with certain categories of Tasks and Objects. We tested Learning and Estimation Performance of the Controller as well as its Robustness. In the following sections , we present our results with our basic comments. In order to better express our results, we first explain the components of the result figures that we will present through the rest of this chapter.

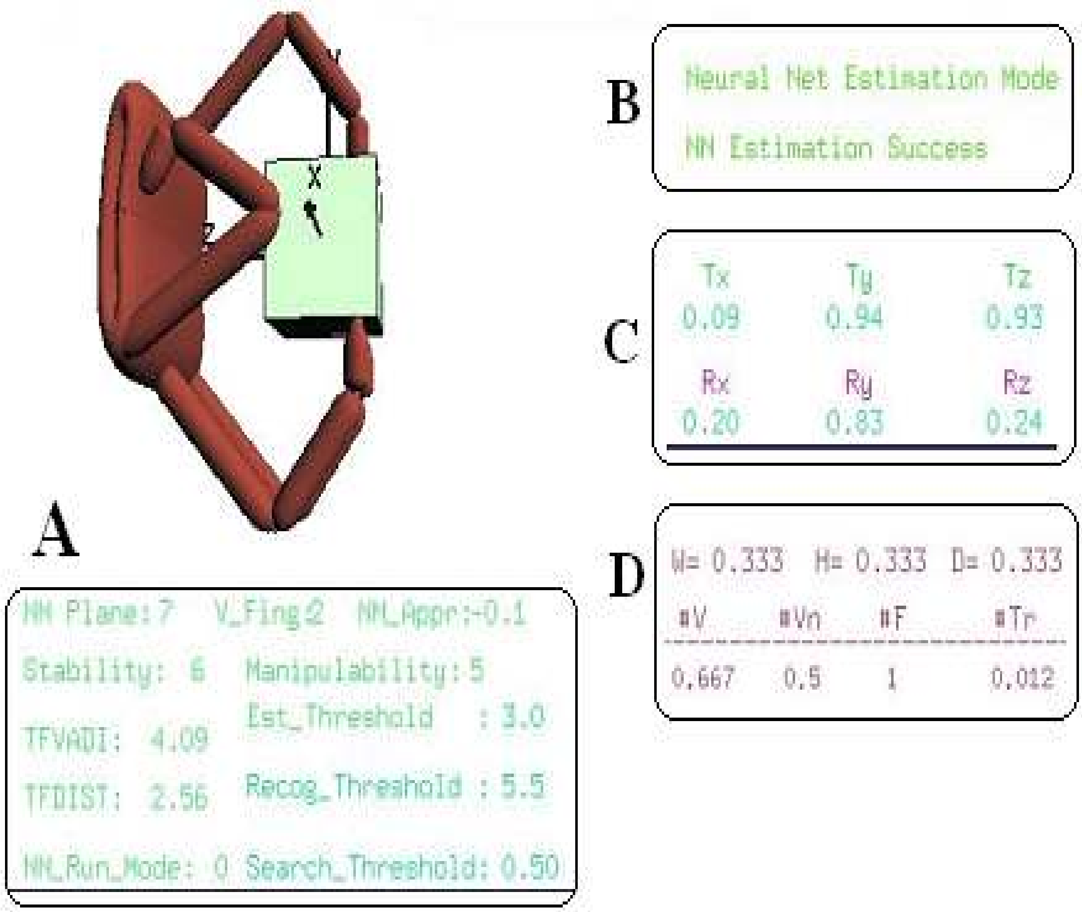


Figure 5.1. Example Result for Explanation of Components

In figure 5.1 , we grouped components of the figure as A,B,C and D groups where the theoretical foundation in these groups are introduced in ch3.

In group A , we could see Approach Plane Index as NN_Plane: 7 , Number of forefingers involved in grasp as V_Fing:2 and Approach Value as NN_Apr: -0.1., described in section 3.4-3.6. In addition, we could also see Stability: 6 and Manipulability:5 values which derived in section 3.7. TFVADI has a value of 4.09 and TFDIST has a value of 2.56 ,which are also defined in section 3.9. In group A, it is also indicated execution mode of the controller as NN_Run_Mode:0, which is given in section 4.3. We also see Estimation Threshold for Execution as Est_Threshold:3.0, Recognition Threshold for Object Recognition as Recog_Threshold: 5.5 and Search Threshold for Minor Search Modes as 0.5 indicating that from preliminary estimated contact point search is limited to 0.5 times relevant object size for that approach plane.

In group B, we could see reporting output form as text message for user to interpret result easily. The first line indicates the Controller Run status and the Second line indicates execution mode of the controller.

In group C, we present the task decomposed as spatial displacements. Upper line indicates translational components and lower line indicates rotational components as it is defined in section 3.4.

In group D, we could see the Object Model attributes, W as Width , H as Height and D as Depth sizes of the model. #V indicates number of vertices in the model, #Vn indicates number of vertex normals in the model, #F indicates number of faces in the model and #Tr indicates number of triangles in the model as described in section 3.2.

As we want to interpret verbally our example result, we could say that for the given task in group C and for the object attributes given in group D, our controller yielded preshape in terms of Approach Plane 7, meaning approach along Z axis and opposition is along Y axis. The number of forefingers included in the preshape is given as 2 and also thumb, fingertips estimated contact points are –0.1 unit away from the Geometric Center of the Object Model along Z axis, with predicted stability of 6 and manipulability of 5 , indicating one principal motion is impossible with this preshape. The result is successfull because TFVADI is higher than Est_Threshold.

5.1.Results with Pre-Trained Object

In these group of results , we aim to test Robustness of the controller performance to, variations in task , using pre-trained object.

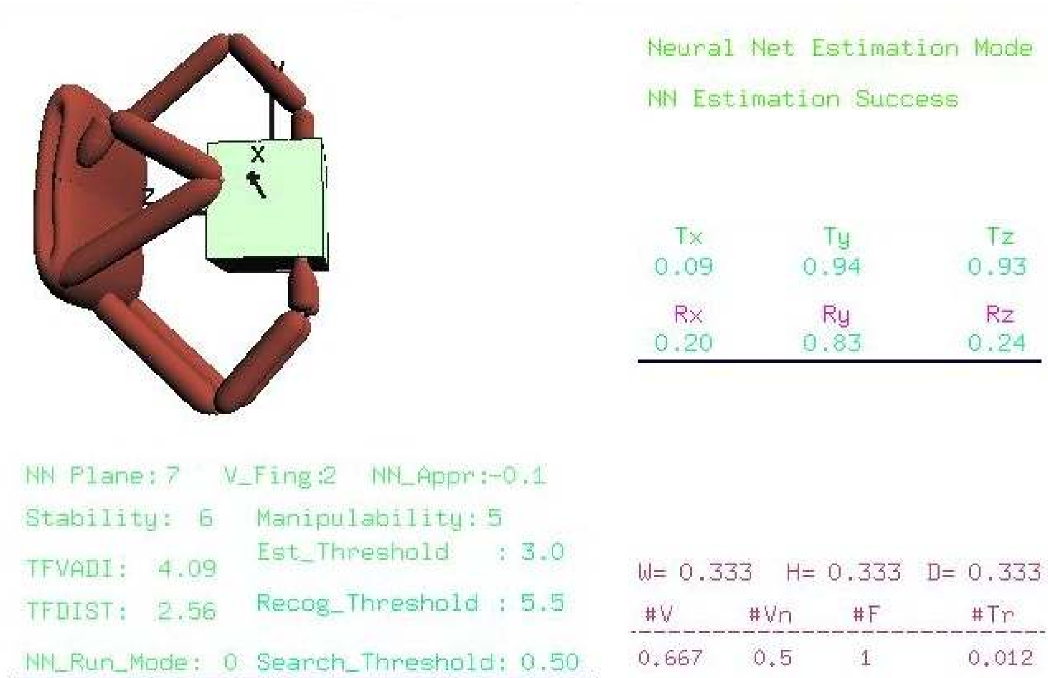


Figure 5.2. Result for Task with Disturbance applied to Pre-trained Object.

In the result above, for the given basic training object, we ask the task as a translational motion along Y axis with a rotation along Y axis as well as translation along Z axis . However, there are also added disturbance items in rotational terms for X and Z rotations . We would like to see response of our controller to tolerate disturbances which do not exist in our training data set. As seen from grasp simulation and Preshape results, controller tolerated such a disturbance for a trained object with success, giving the approach plane , approach and indication of proper execution mode 0, given in figure 5.2 group A.

Figure 5.3.Result for a Known Task applied to Pre-trained Object.

In the result above, for the given basic training object, we ask the controller a known task as a translational motion along X axis with a rotation along X axis as well as translation along Y axis . We would like to see response of our controller to perform the resultant preshape for the pretraining object . As seen from Grasp simulation and Preshape results, the Controller yields a right preshape estimation mathcing degree index TFVADI 3.6, higher than Est_Threshold; in execution mode 1, meaning that this category of task not trained well.

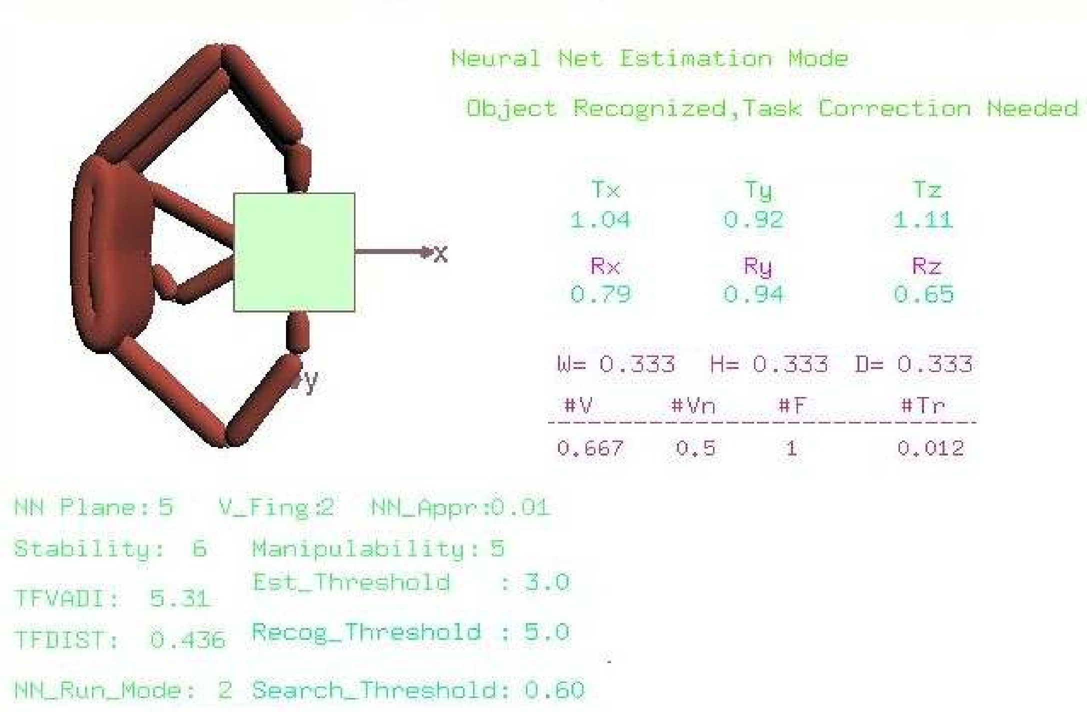


Figure 5.4.Result for a Very Difficult Task applied to Pre-trained Object.

In the result above, for the given basic training object, we ask the controller an almost impossible task with this object geometry which is prismatic . The task as translational motions along X,Y,Z axes with rotations along X,Y,Z axes. We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller recognized the object and asked for task correction with TFVADI 5.31 which is compared to Recog_Threshold this time. This is indication of proper execution mode 2.

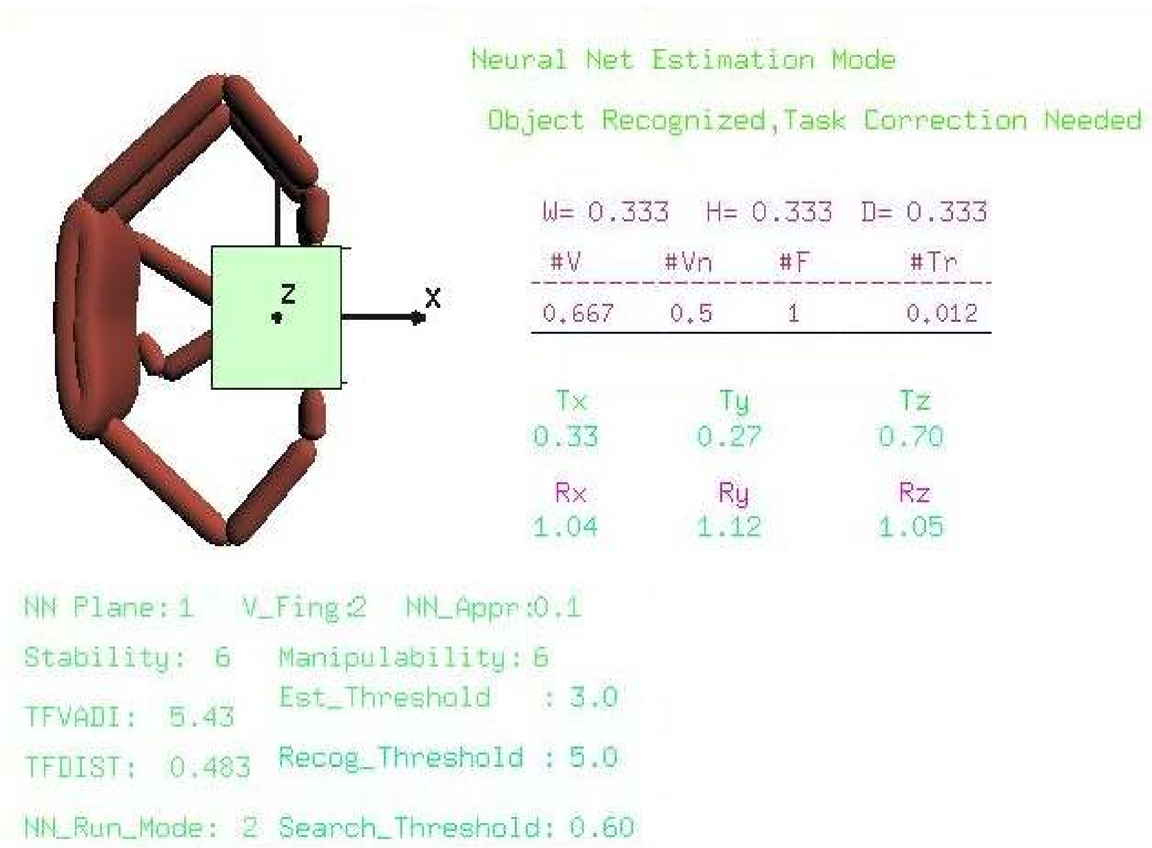


Figure 5.5.Result for Another Very Difficult Task applied to Pre-trained Object.

In the result above, for the given basic training object, we ask the controller an almost impossible task with this object geometry which is prismatic . The task as a translational motions along Z axes with rotations along X,Y,Z axes. We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller recognized the object with TFVADI 5.43 and asked for task correction. This is also indication of proper execution mode 2. In this result we see that our controller recognized the trained object under variations of task attributes.

5.2.Results with Training for Primitive Objects

In these group of results , we aim to test Learning Performance of the controller for relatively simple objects ,using trained tasks.

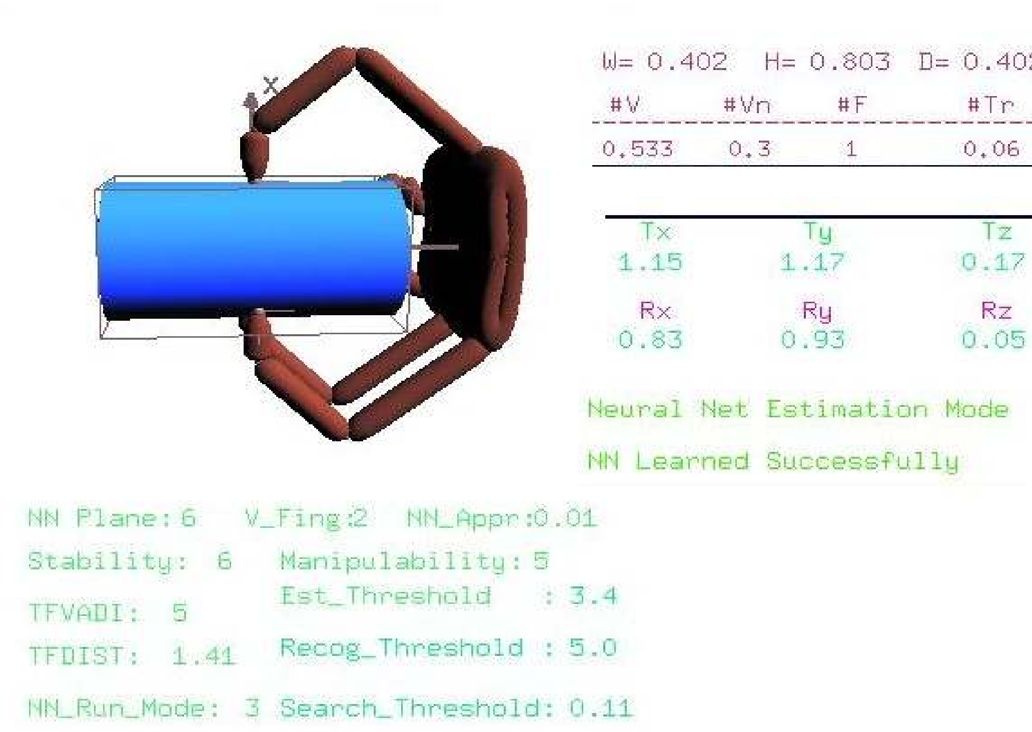


Figure 5.6.Result for a Normal Task applied to a new Primitive Object,Cylinder.

In the result above, for the given basic task, we ask the controller to recognize a new primitive object, Cylinder and ask to learn it. The task is given as translational motions along X,Y axes with rotation along X,Y axes. We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully. This also indicates successful self recurrence of controller at exec mode 0 after learning, for self test resulting with a TFVADI 5.0.

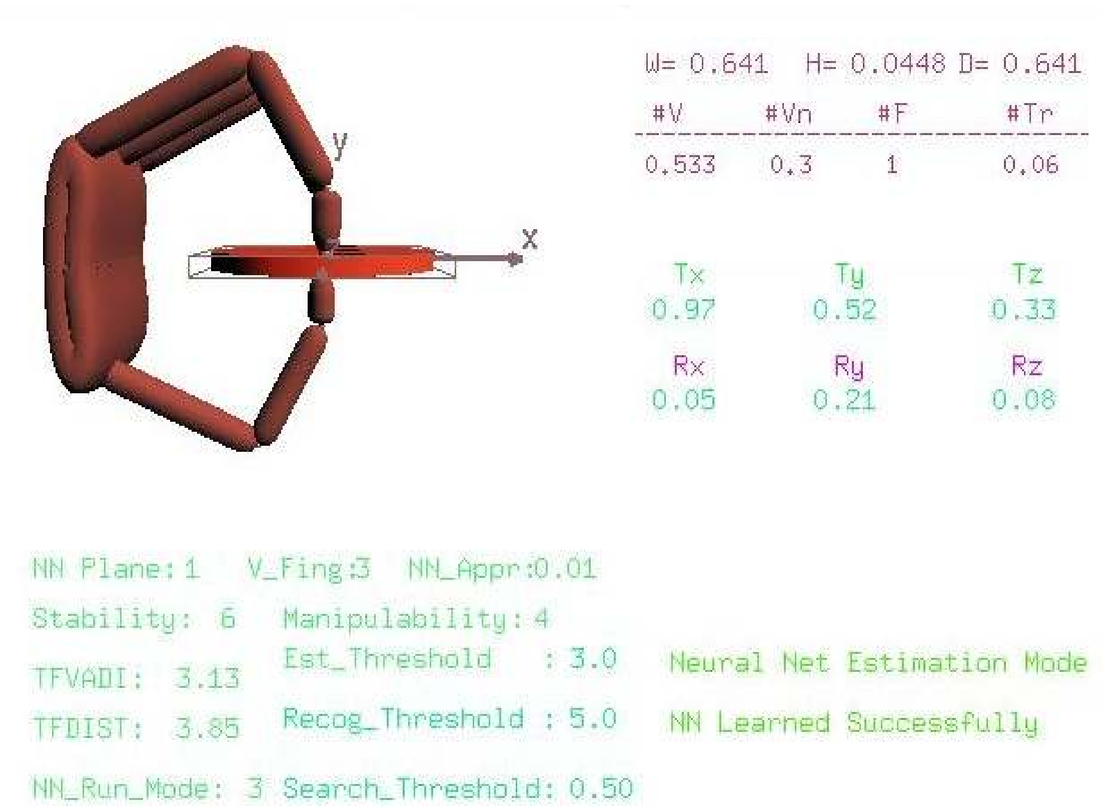


Figure 5.7.Result for a Normal Task applied to a new Primitive Object, Disc.

In the result above, for the given basic task, we ask the controller to recognize a new primitive object , Disc and ask to learn it. The task is given as translational motions along X,Y,Z axes with rotation along Y axes. We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully with TFVADI 3.13 as compared to Est_Threshold. This also indicates successful self recurrence of controller at exec mode 0 after learning, for self test.

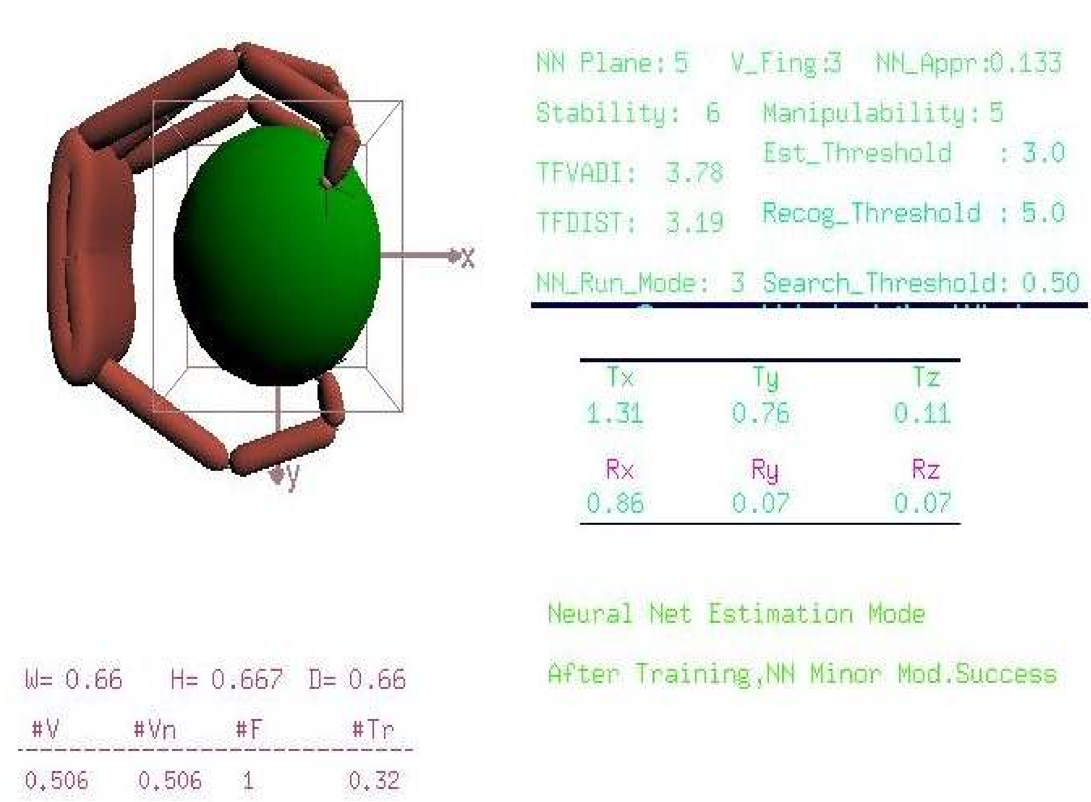


Figure 5.8.Result for a Normal Task applied to a new Primitive Object, Sphere.

In the result above, for the given basic task, we ask the controller to recognize a new primitive object and ask to learn it. The task is given as translational motions along X,Y axes with rotation along X axes. We would like to see response of our controller. As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully. However, the controller recognized the object in exec mode 1 with TFVADI 3.78. This also indicates successful self recurrence of controller at exec mode 1 after learning, for self test.

5.3.Results with Training for Complicated Objects

In this category we test Learning performance of the controller with relatively difficult objects.

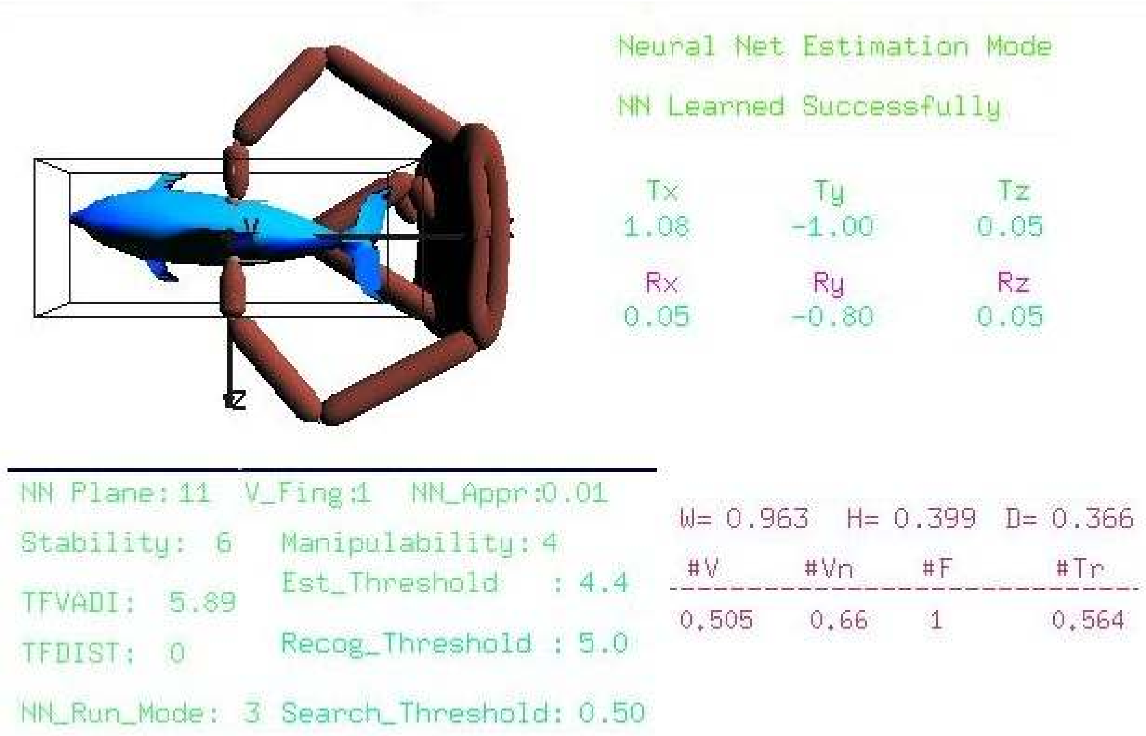


Figure 5.9.Result for a Normal Task applied to a new Object, Dolphin.

In the result above, for the given basic task, we ask the controller to recognize a new complicated object, Dolphin and ask to learn it. The task is given as translational motions along X, -Y axes with rotation along ,-Y axes. We would like to see response of our controller. As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully. The learning occured with a relatively high matching degree TFVADI 5.89 due to given a very optimal task which yields our conlusion that complicated objects should be trained with compatible tasks. This also indicates successful self recurrance of controller at exec mode 0 after learning, for self test.

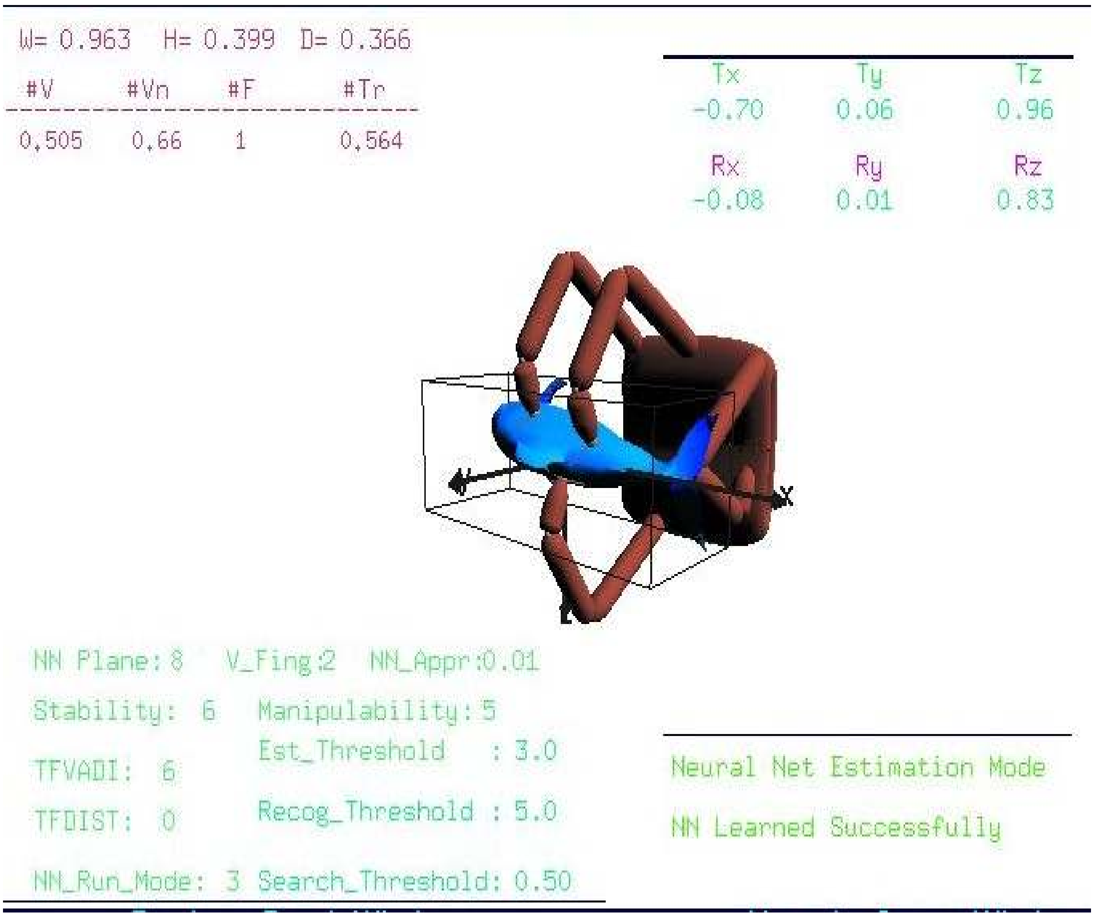


Figure 5.10. Result for Another Normal Task applied to the Dolphin.

In the result above, for the given basic task, we ask the controller to recognize a new complicated object, Dolphin and ask to learn it. The task is given as translational motions along -X, Z axis with rotation along ,Z axes. We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully with another task. This also indicates successful self recurrance of controller at exec mode 0 after learning, for self test. This result is from same object with another task. Notice that TFVADI is 6 due to contributions from previous example learnings.

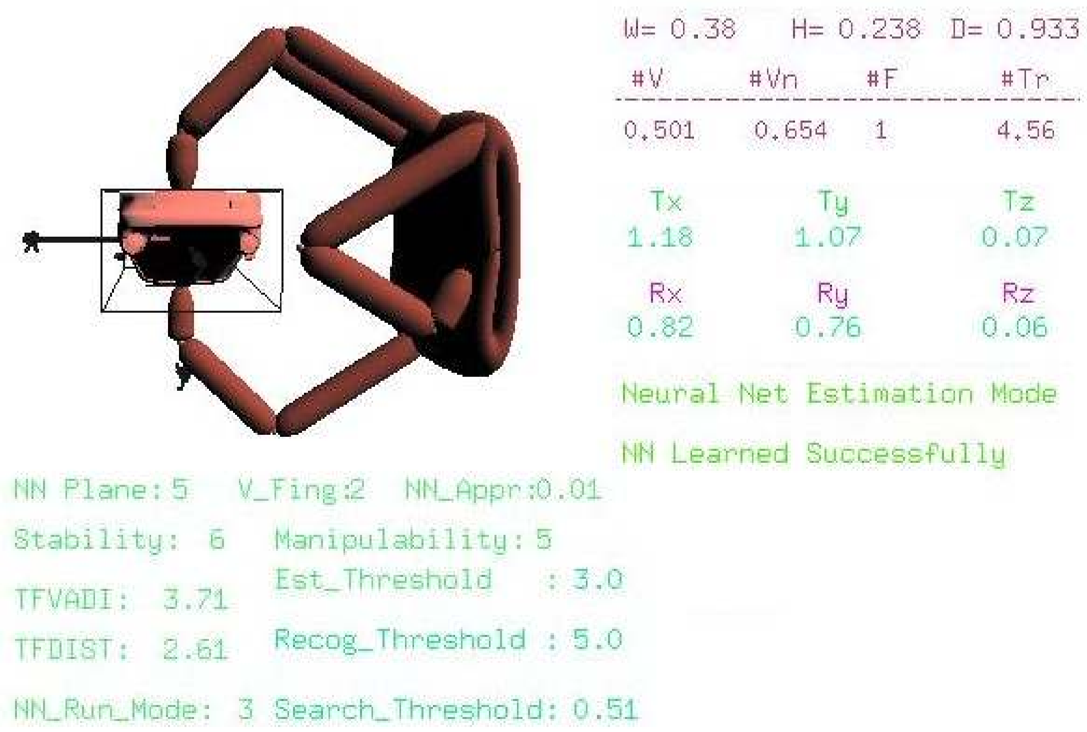


Figure 5.11.Result for a Normal Task applied to a new Object, ToyCar.

In the result above, for the given basic task, we ask the controller to recognize a new complicated object, ToyCar and ask to learn it. The task is given as translational motions along X, Y axes with rotation along ,X,Y axes. We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller recognized new object and learned it successfully with TFVADI 3.71. This also indicates successful self recurrence of controller at exec mode 0 after learning, for self test.

5.4.Results with Primitive Objects After Learning

In this category, we test Execution Performance of the controller with relatively simple objects.

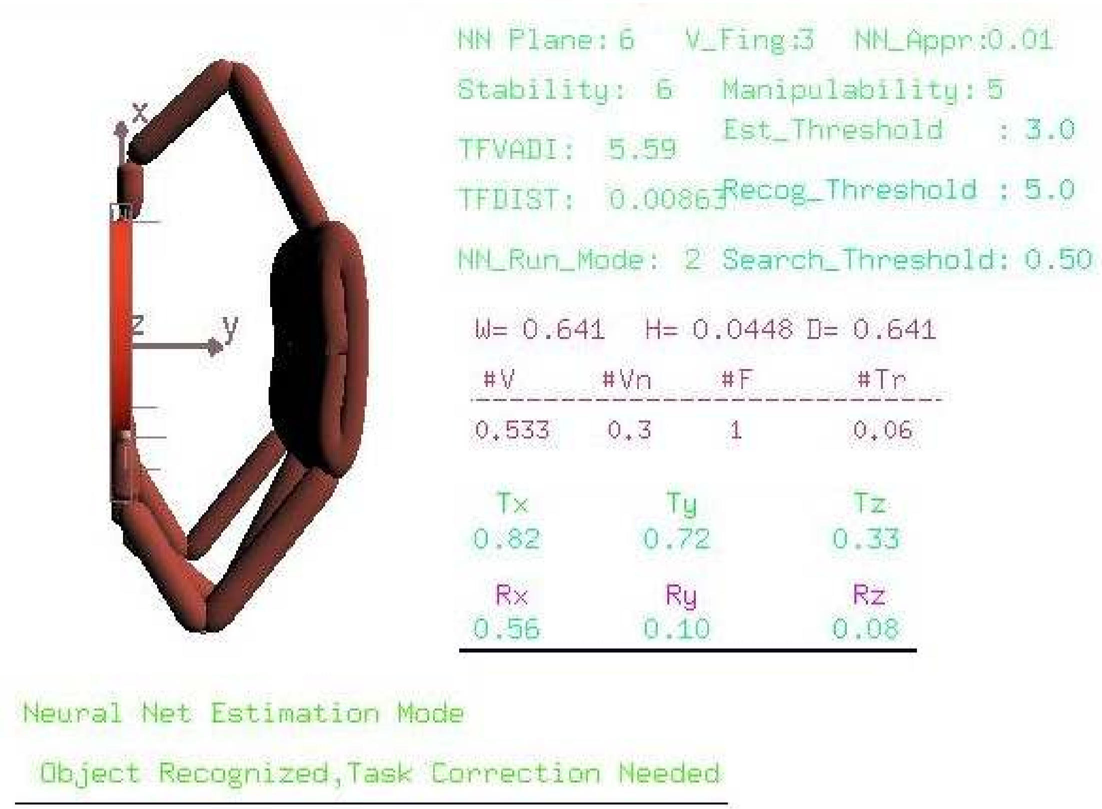


Figure 5.12. Result for a Difficult Task applied to Primitive Object After Training.

In the result above, for the given primitive object after training, we ask the controller an difficult but possible task with this object geometry which is disc and thin . The task as a translational motions along X,Y axes with rotations along X axes. We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller recognized the object and asked for task correction. This is indication of proper execution mode 2.

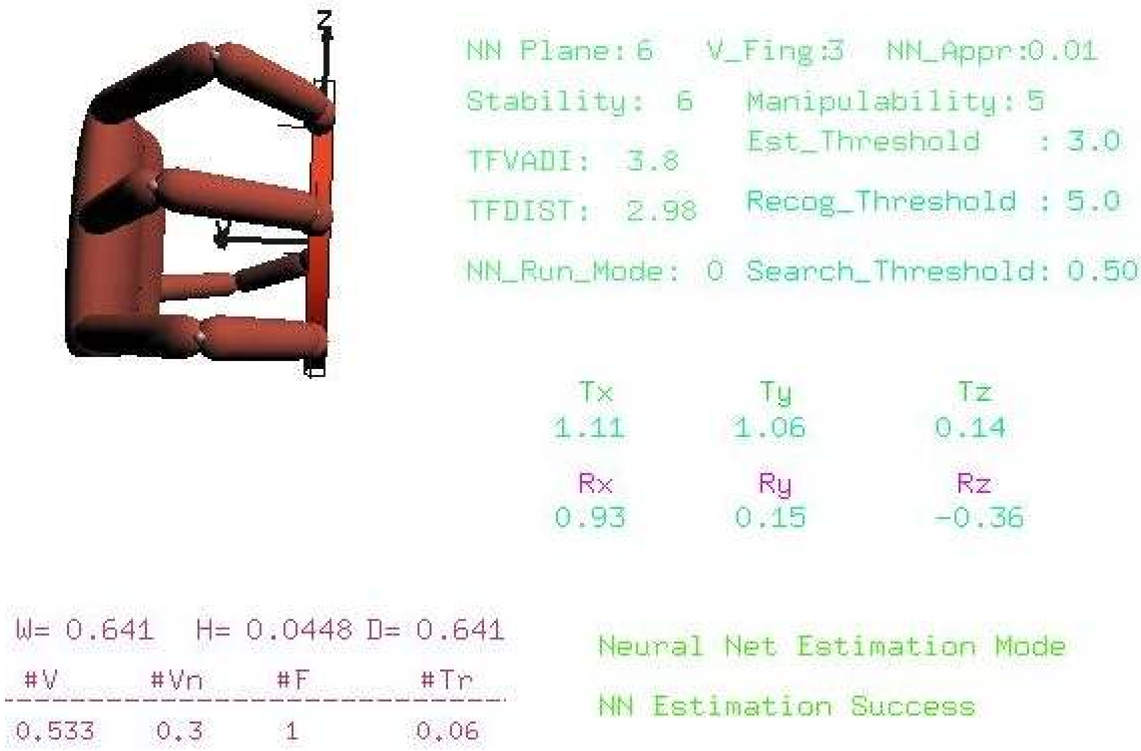


Figure 5.13.Result for a Task with Disturbance ,to primitive Object After Training.

In the result above, for the given primitive object after training, we ask the task as a translational motion along X,Y axess with arotation along X axis as well as disturbance rotation along -Z axis . We would like to see response of our controller to tolerate disturbances which do not exist in our training data set. As seen from grasp simulation and Preshape results, controller tolerated such a disturbance for a trained object with success, giving right plane and approach. It is also indication of proper execution mode 0.

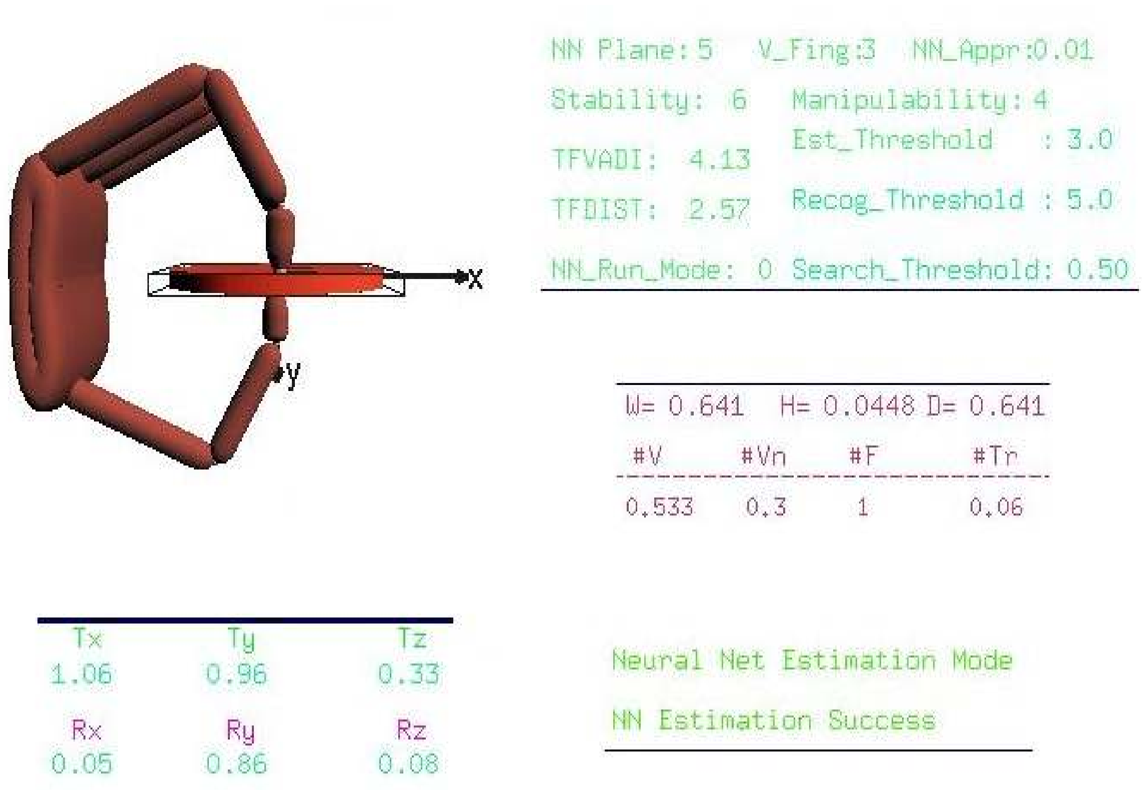


Figure 5.14.Result for another Task with Disturbance ,to primitive Object After Training.

In the result above, for the given primitive object after training, we ask the task as translational motion along X,Y axes with a rotation along X axis as well as disturbance translation along Z axis . We would like to see response of our controller to tolerate disturbances which do not exist in our training data set. As seen from grasp simulation and Preshape results, controller tolerated such a disturbance for a trained object with success, giving right plane and approach. It is also indication of proper execution mode 0.

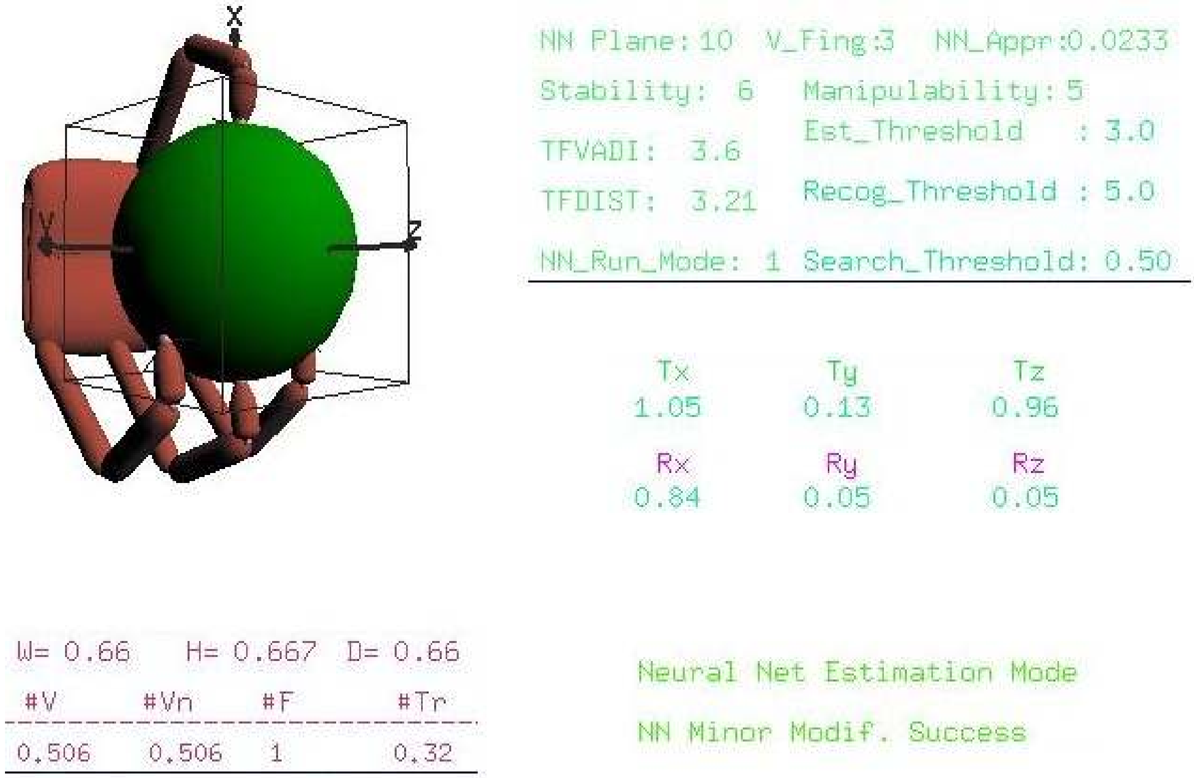


Figure 5.15.Result for a Normal Task applied to a new Primitive Object, Sphere, After Training.

In the result above, for the given basic task, we would like to execution performance of the Controller . The task is given as translational motions along X axis with rotation along , X axes. We would like to see response of our controller. As seen from Grasp simulation and Preshape results, the Controller performed execution of a modified task at exec mode 1, with TFVADI 3.6.

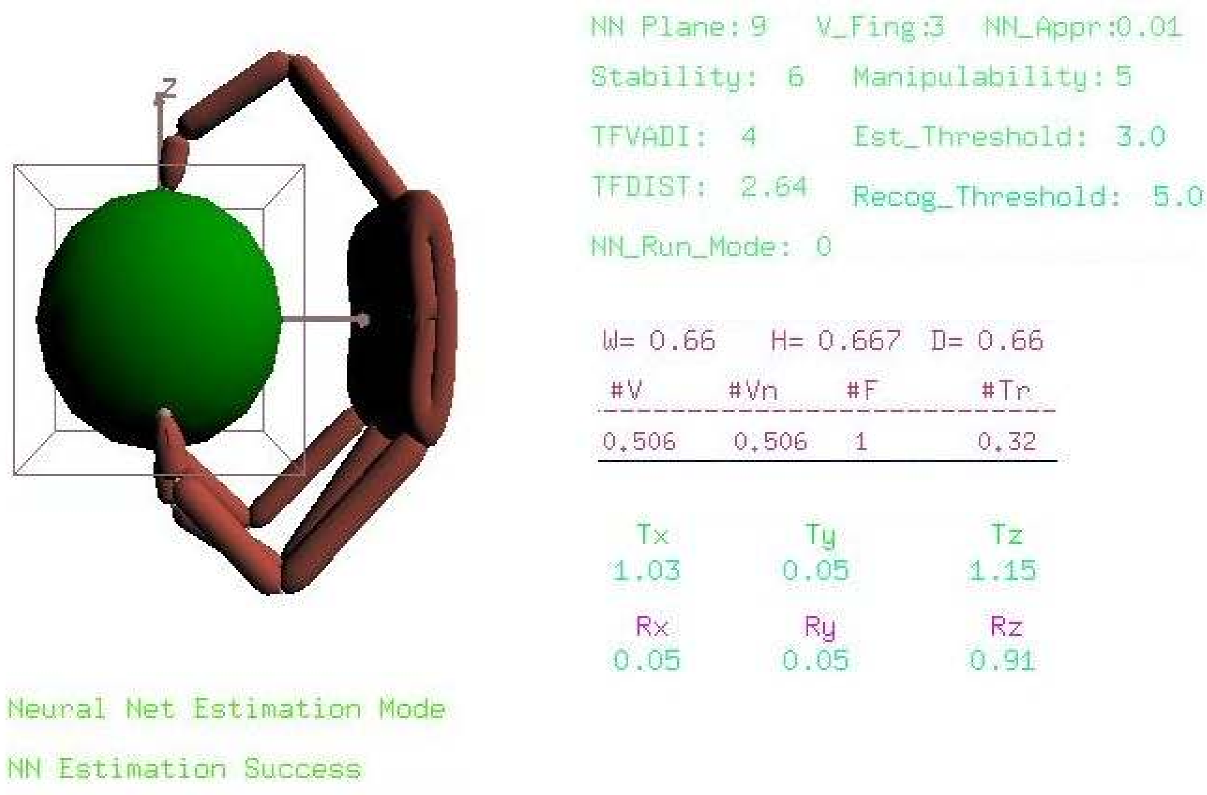


Figure 5.16.Result for a Normal Task applied to Primitive Object, Sphere After Training.

In the result above, for the given basic task, we would like to execution performance of the Controller . The task is given as translational motions along Z axis with rotation along , Z axes. We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller performed execution of a modified task at exec mode 1.

5.5.Results with Complicated Objects After Learning

In this category we test Execution performance of the controller with relatively difficult objects.

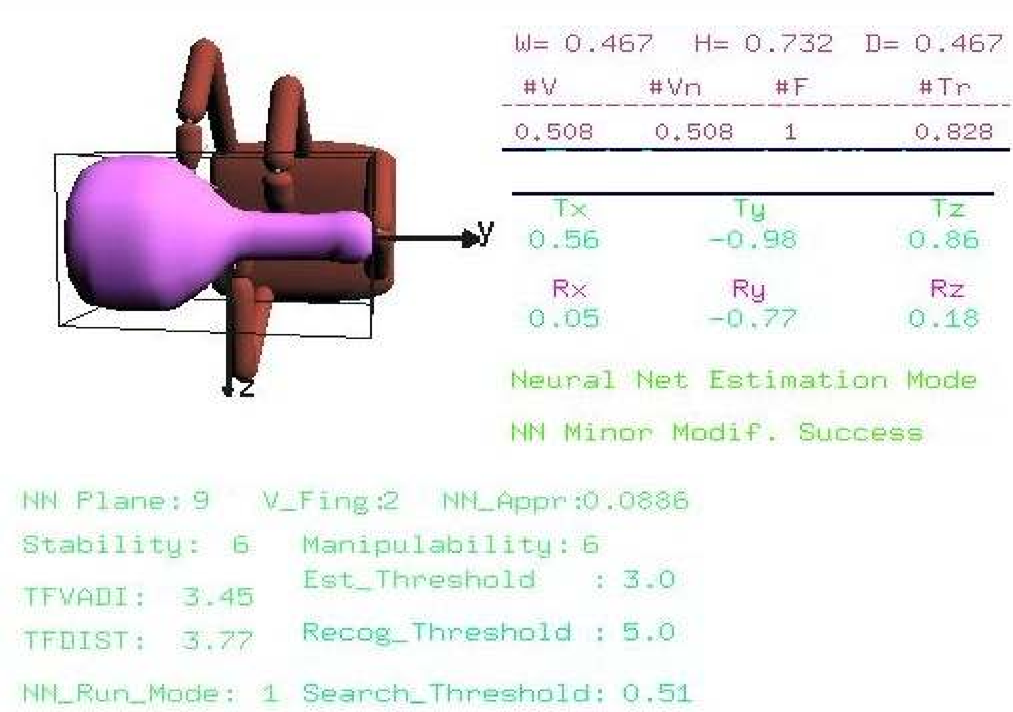


Figure 5.17.Result for a Possible Task applied to Complicated Object,After Training.

In the result above, for the given complicated object after training, we ask the controller a possible but relatively difficult task as translational motions along X,-Y,Z axes with rotation along –Y axis . We would like to see response of our controller. As seen from Grasp simulation and Preshape results, the Controller performed right but with a minor modification with TFVADI 3.45 which is satisfactory since Est_threshold is 3.0.

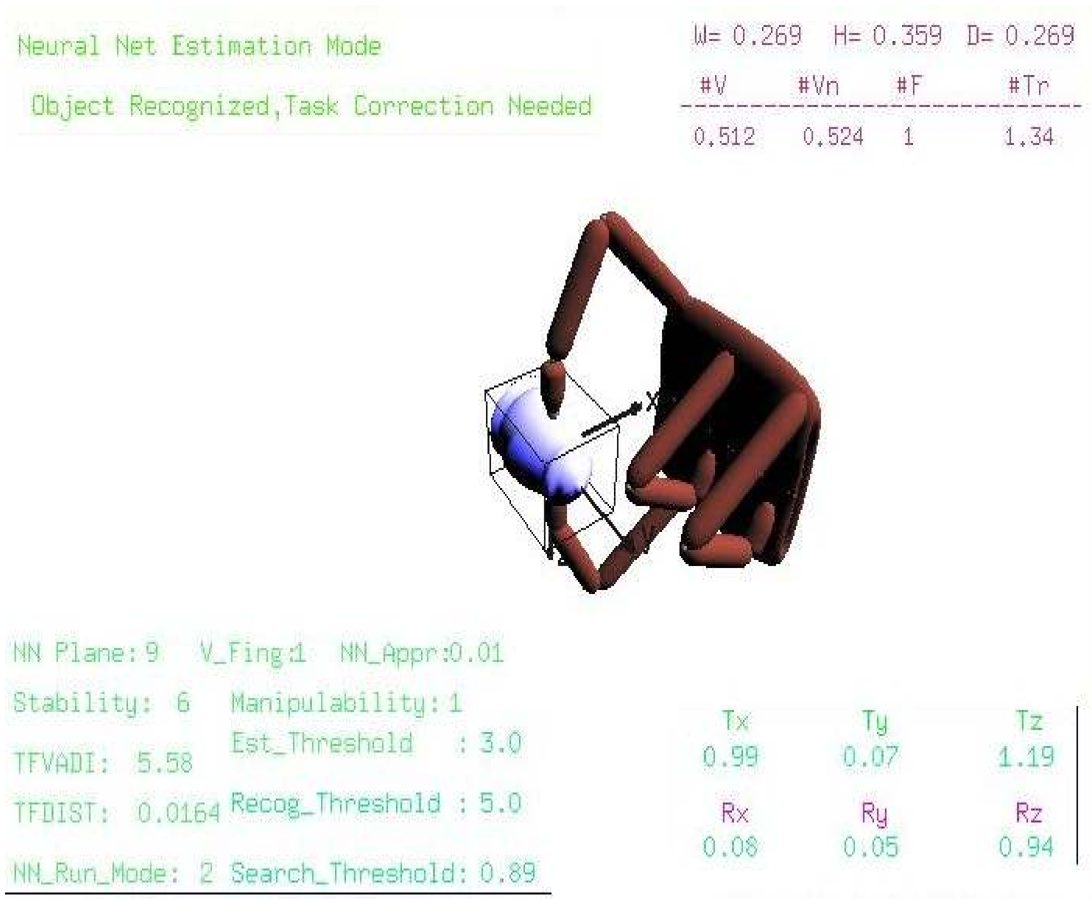


Figure 5.18.Result for a Non Applicable Task for a Complicated Object.

In the result above, for the given complicated object , on which due to its geometry is only VF 1 grasps are possible . However, we asked a grasp which could require min VF 2. We would like to see response of our controller . Hence, our controller, response to this unfeasible task requirement by recognizing the trained object and asking for the correct task to Meta Control. Task is as seen , Translation along Z and X axis, combined with a rotation around Z axis..

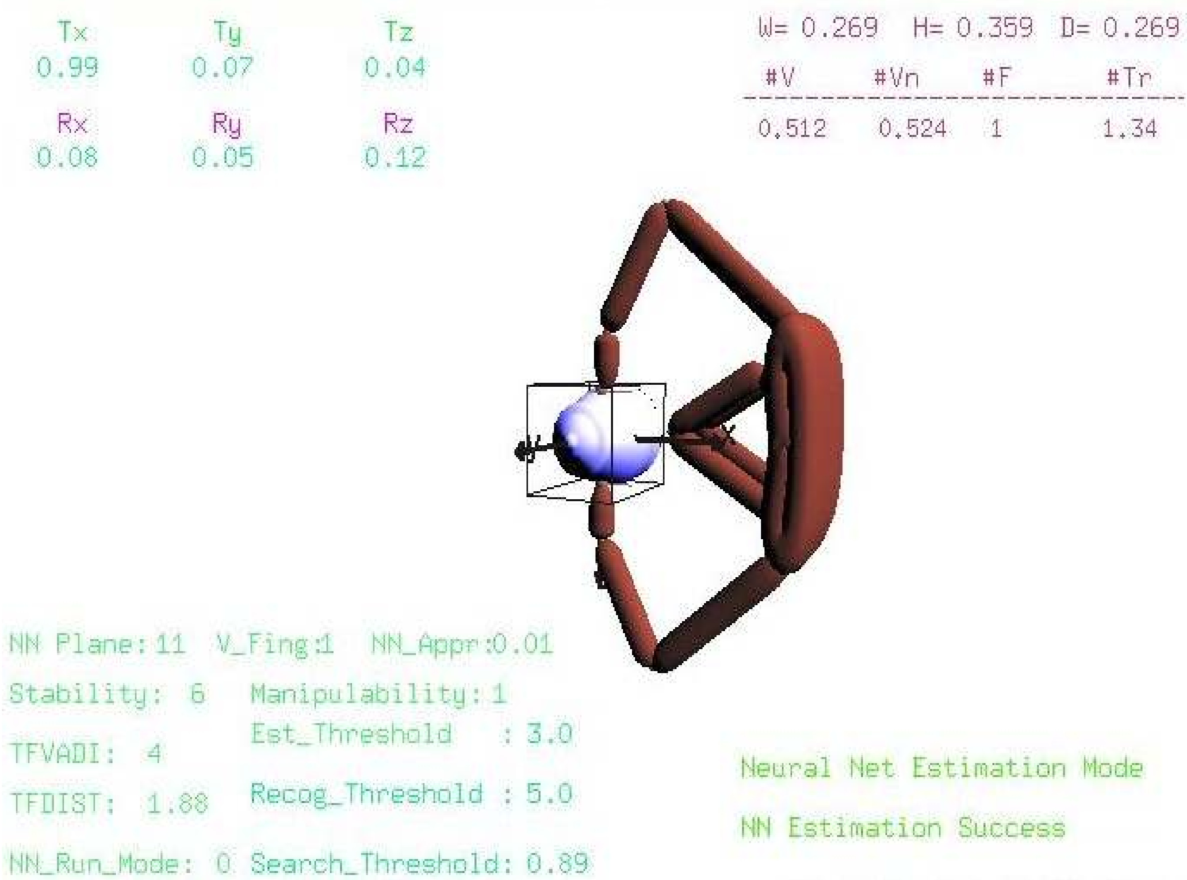


Figure 5.19.Result for a Corrected Task applied to Pre-trained Object.

In the result above, for the given basic training object, we ask the controller a known task as a translational motion along X axis without any rotation . We would like to see response of our controller. As seen from Grasp simulation and Preshape results, the Controller yielded a suitable preshape , when the task is corrected , with TFVADI 4.

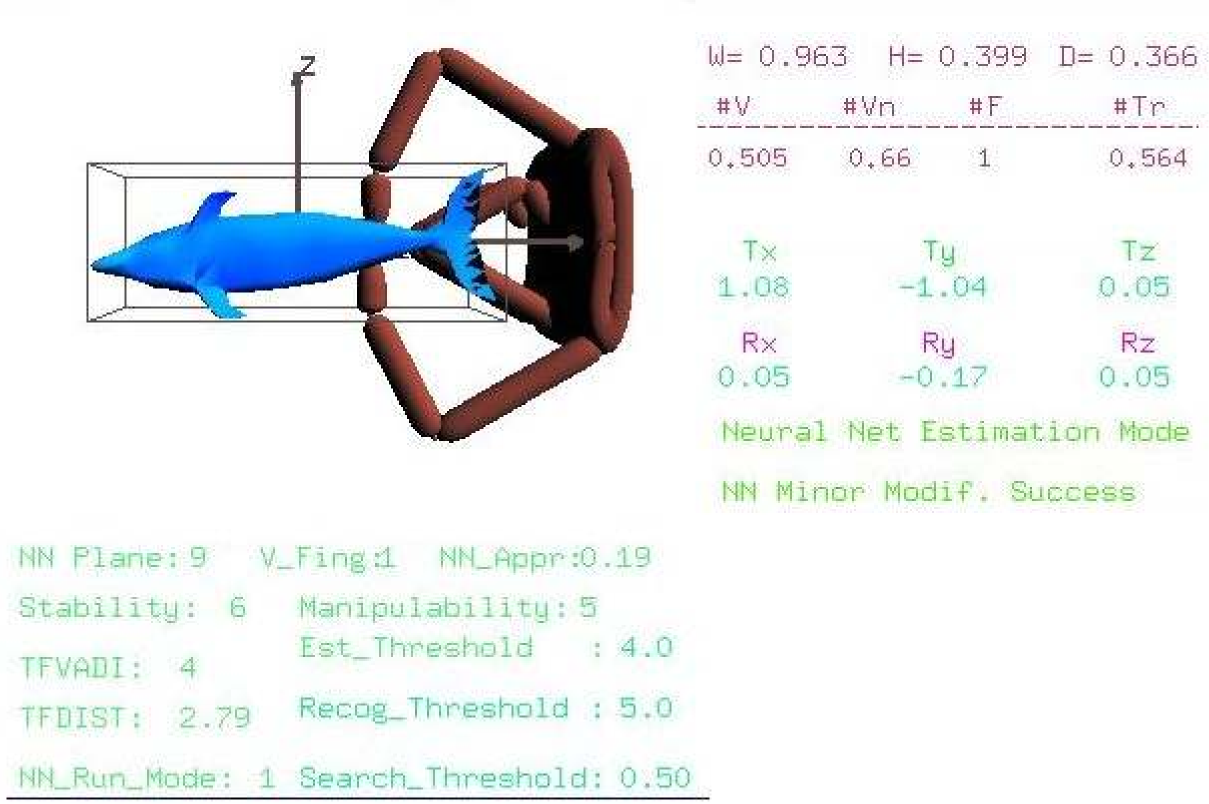


Figure 5.20.Result for a Possible Task applied to Complicated Object,After Training.

In the result above, for the given complicated object after training, we ask the controller a possible but relatively difficult task as translational motions along X,-Y, axes with low rotation along –Y axis . We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller performed right but with a minor modification. It is seen from the figure that matching degree for this case is TFVADI :4 , which is obtained by means of search threshold use.

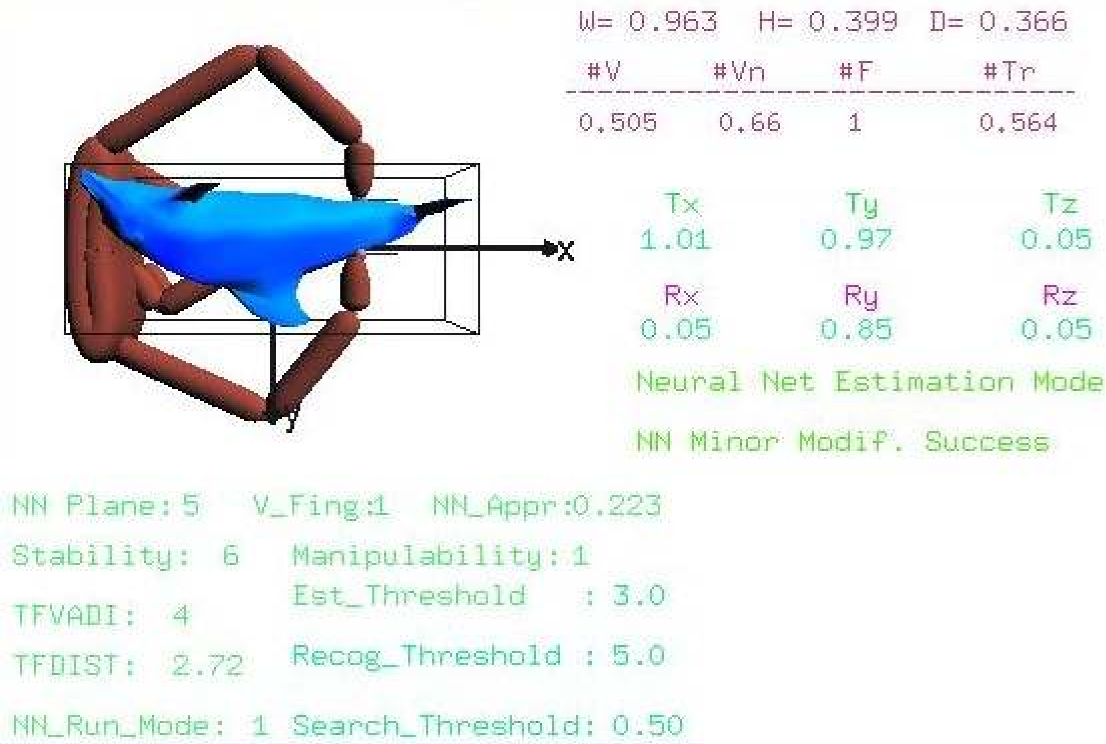


Figure 5.21.Result for Another Possible Task applied to Complicated Object, After Training.

In the result above, for the given complicated object after training, we ask the controller a possible but relatively difficult task as translational motions along X,Y, axes with high rotation along Y axis a new task as compared to previous result. We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller performed right but with a minor modification. It is seen from the figure that mathcing degree for this case is TFVADI :4 , which is obtained by means of search threshold use.

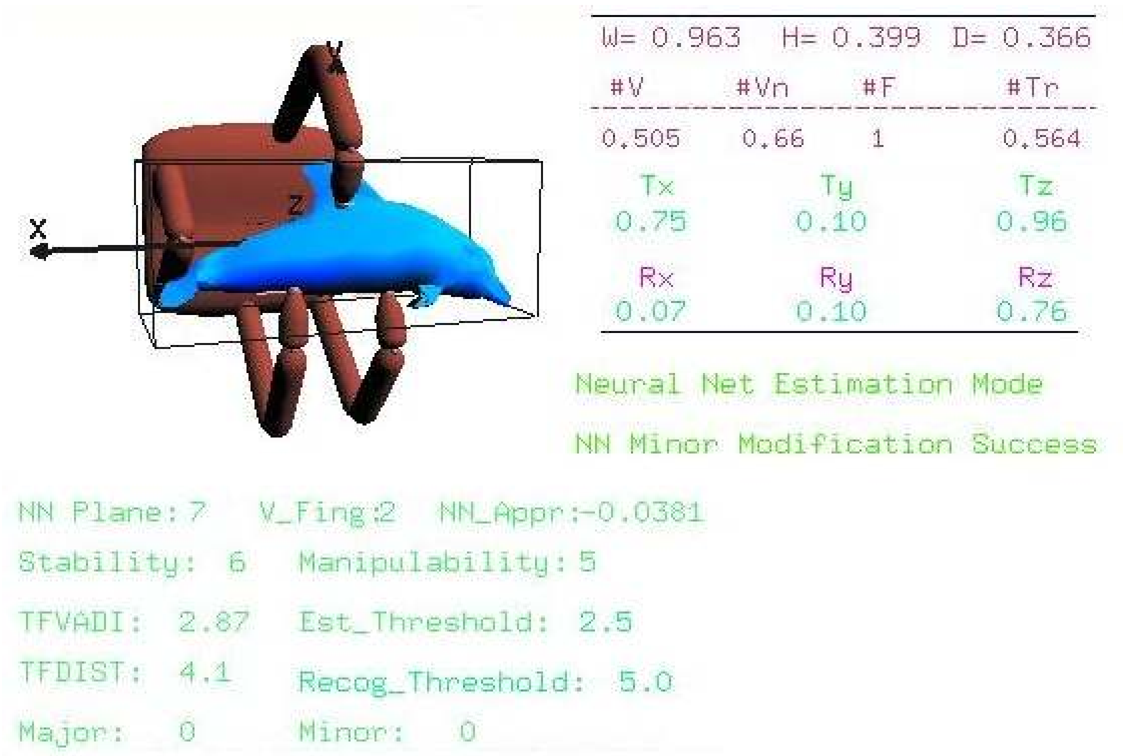


Figure 5.22 Result for Another Possible Task applied to Object Dolphin ,After Training.

In the result above, for the given complicated object after training, we ask the controller a possible but relatively difficult task as translational motions along X,Y, axes with high rotation along Y axis . We would like to see response of our controller . As seen from Grasp simulation and Preshape results, the Controller performed right but with a minor modification. In other words, for a task other than used in learning the object, controller demonstrated that it has learned the object and task mappings for this object but with a very low TFVADI 2.87. We accept this result because Est_threshold is 2.5 in this case.

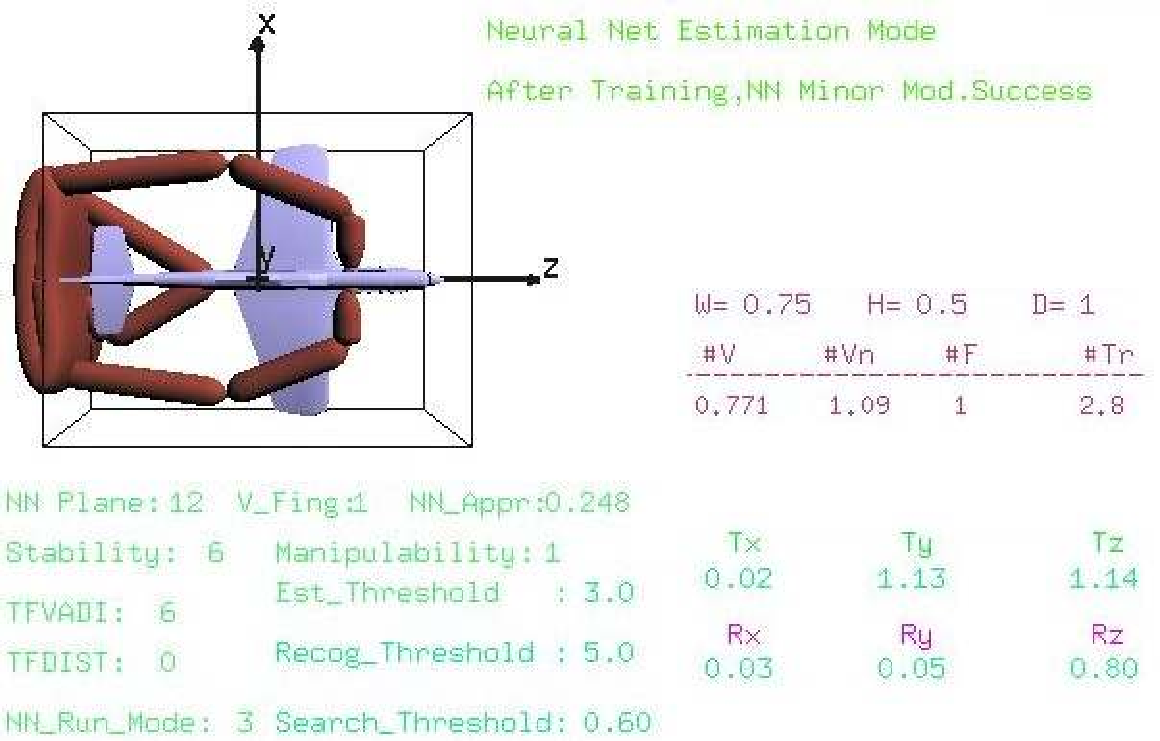


Figure 5.23.Result for a Normal Task applied to trained Complicated Object, ToyPlane.

In the result above, for the given a complicated training object after training, we ask the controller a known task as a translational motion along Y,Z axes with a rotation along Z axis . We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller performed right but with an extra training indicating that object is not properly introduced in the one run training. The minor modification also indicates that.

Figure 5.24.Result for a Normal Task applied to trained Complicated Object, ToyCar.

In the result above, for the given a complicated training object after training, we ask the controller a known task as a translational motion along Y,Z axes with a rotation along Z axis . We would like to see response of our controller to perform an expected behaviour. As seen from Grasp simulation and Preshape results, the Controller performed right but with an extra training indicating that object is not properly introduced in the one run training. The minor modification also indicates that.

5.6.Results with Odd Cases

In this group, we present improper results which did not yield in grasp simulation results successfuly. Fingers could not landed is taken to an initial position by the controller to indicate failure visually .

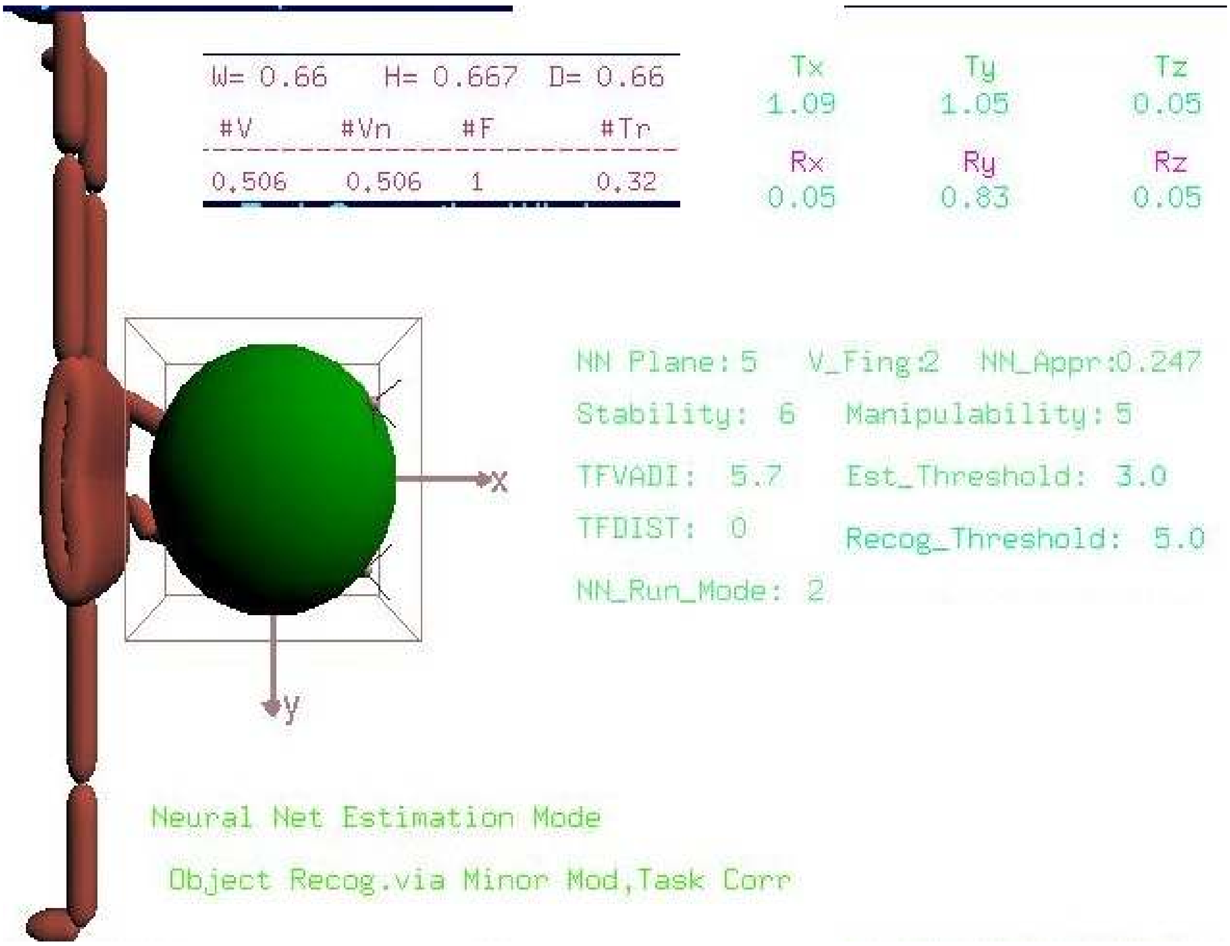


Figure 5.25.Result for a Unreachable Contact landing Points with a Primitive Object.

In figure 5.25 , we observe that the controller recognized object but due to the hand simulation kinematics reach check condition given in Craig (1985), it could not land on a surface. In order to indicate this situation, in our hand simulation we implemented "Reach Check" condition and to demonstrate existence of non reachability condition, we defined a state for finger position to visualize that. Hence we classify these results as odd cases.

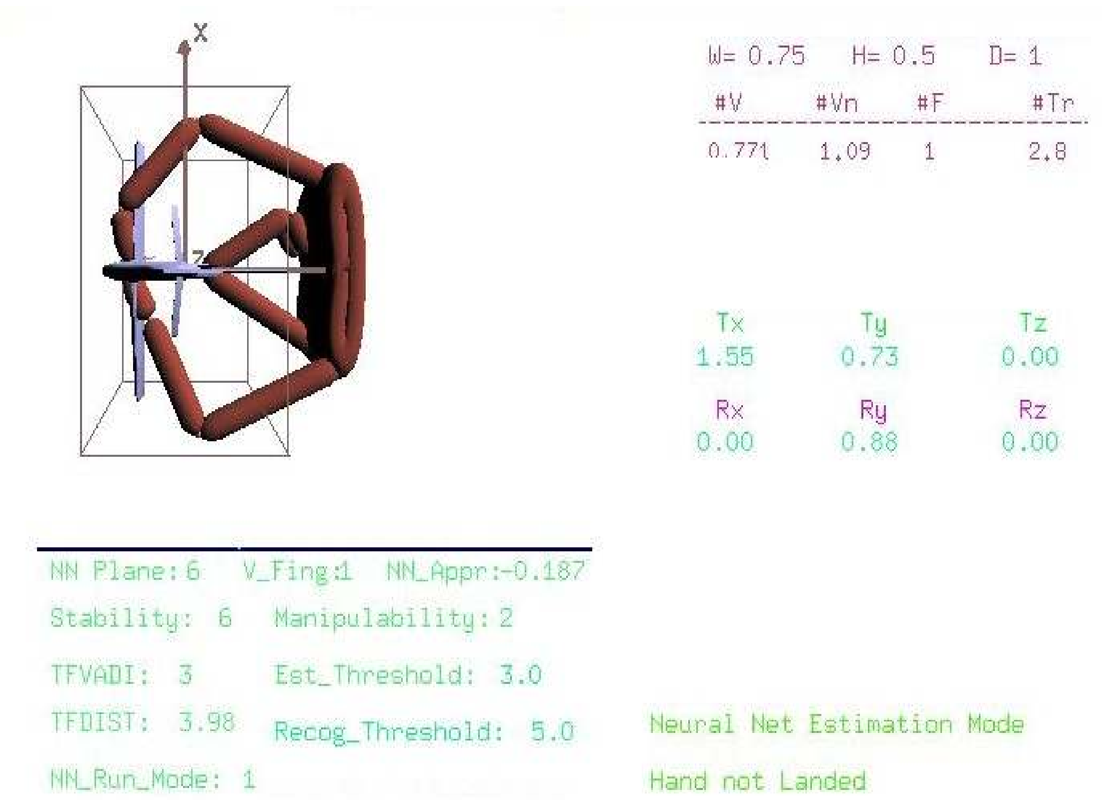


Figure 5.26. Result for a Unreachable Contact landing Points,for Complicated Object ToyPlane .

In the figure 5.26, we accept the result as odd case because our controller noticed that it could not reach the estimated contact landing points due to hand simulation kinematics reach control as stated in previous example. In addtion, we also observe, unwanted collision of fingers to object surfaces. Hence, we classify this result as an odd case.

5.7. Discussions

In our implementation, we have tested the Intelligent Controller with certain categories of Tasks and Objects , for Estimation and Learning Performance of the Controller as well as its Robustness. We present our further discussions as follows .

While we test the controller with the Pre-Trained Object , in section 5.1, we observed that the controller remembers the Pre-Trained Object in any case and has capability to tolerate variations in the task up to 10-15 % translational components and 15-20 % in rotational components. In other words, we deduce that after a proper training with a given object, using this tolerance capacity to some extent, the controller could perform some composite tasks at once, instead of a sequential execution of individual task primitives. In addition, we also observed that robustness of our controller is satisfactory to compensate disturbances in task and recognition capabilities are well enough to detect non applicable tasks. The controller demonstrated an easy learning of primitive objects not paying much attention on the task , shown in section 5.2. However, for the complicated objects, we noticed that giving a known task when a new object is presented, yields better learning performance. Furthermore, the estimation performance of the controller for primitive objects is also satisfactory and the objects presented are successfully learnt by the Intelligent Controller as discussed in section 5.4. The execution and learning performances of the Controller for complicated objects are also satisfactory and have been discussed within the context of results of section 5.5 . However, we also had a few odd case for our framework that we included in section 5.6.

As compared to the grasp synthesis controllers done previous works in the literature, the Controller is more versatile and has advantageous features coming from the developed architecture. For instance previous works in the literature were completely based on rules and taxonomies and carried these characteristics as weakness in handling disturbances and it was not possible to teach new objects in those approaches. In our work, due to our Incremental Learning feature, we could train new objects while still obeying general assumptions of stability and manupulability.

Our work is also superior to the some existing works which only indicates how the techniques could be implemented as demonstrations. However, our approach is a control approach designed and implemented with a modular architecture that can handle uncertainities as disturbances , vagueness and new unseen cases by its learning capability. Our work yields with actual and practically implementable results.

6. TECHNICAL PERFORMANCE ANALYSIS

In our work, after having results with certain classes, we also investigated performance of the Preshape Controller when parametric changes occur. In this chapter, we present the technical performance analysis of Preshape Controller under those parameter changes. As it could be remembered from our design given in Chapter 3, parametric variations were classfied under following categories.

- Contact Model Parameters
- Fuzzy Logic Membership Functions
- Neural Network Parameters
- Thresholds for Critics
- Pretraining Object Model

In our analysis we start with the above mentioned order . In order to investigate the response of the controller to parametric variations, we implemented some tests. In these tests, we aim to demonstrate the sensitivity of the controller to various parameters. In general, sensitivity analyses are represented by classical plots, however in our case, we have to demonstrate it using results from the controller modules, due to its decentralized nature. In the following sections we present the analysis.

6.1.The Controller Performance with Variations in Contact Model Parameters

In kinematic estimations section 3.6, we mentioned parameters for contact wrenches such as Ka,Ko,Kn and Mn . These parameters are used to represent softfinger contact wrench coefficients. In Salisbury's contact model, wrenches have unity values for coefficients. However, in actual implementations of soft finger by Howe and Cutkoksy (1996) , the wrenches are scaled with some coefficients, as given in equation (3.6.1) . In our work , we design our contact model as compatible with the existing contact models and we utilize Howe and Cutkosky's work in determination of these parameters as well as softfinger experiments of Murakami (2003).

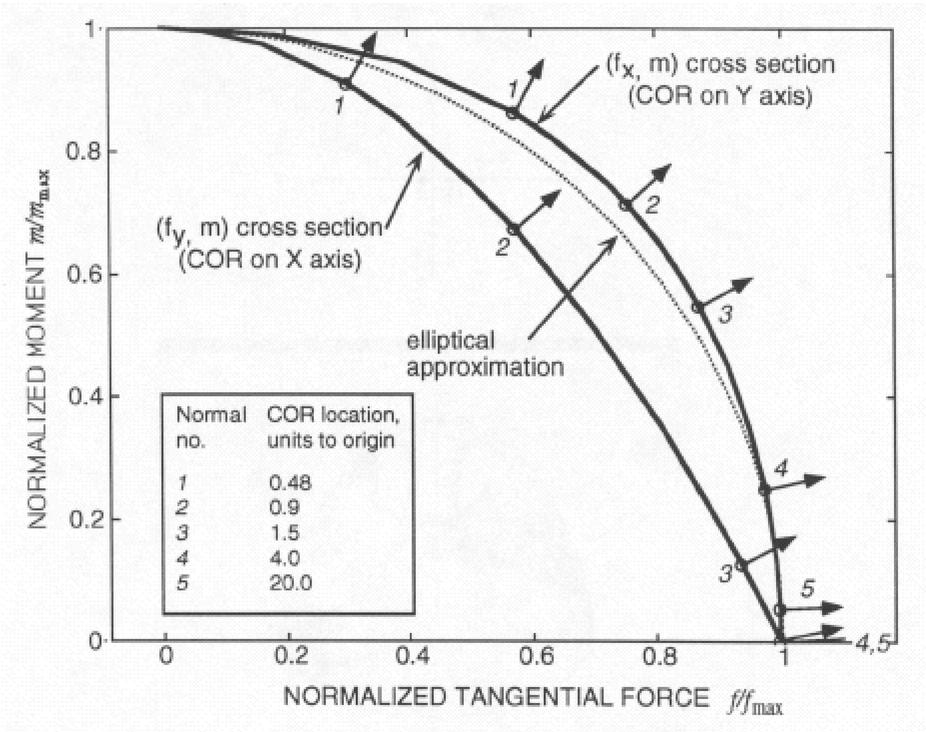


Figure 6.1.Ratio Between Tangential Force and Moment for Soft Finger Contact Model (Howe & Cutkosky 1996)

Figure 6.1 indicates , soft finger friction coefficent ratios based on Limit Surface approach and eqn 3.6.1. Since in our object model, contact surfaces are triangular planes, surface normals could be taken as contact normals, hence, this correspond to normal 1 category of figure 6.1. We also choose COR (center of rotation for contact) location as 0.2. After an inspection of the figure 6.1, we infer about values for Ka, Ko and Mn, where Kn=1.0 value represents normal force coefficient, as a natural result of our contact model. We select Mn=0.8 and therefore the corresponding values of Ka and Ko, becomes 0.5 and 0.3 each ,where Mn is moment coefficient , Ka is frictional force coefficient along approach axis and Ko is coefficient along transversal axis with respect to hand grasp approach. In eqn 3.6.1, the friction coeffient μ is also expressed. In order to determine friction coefficent μ, we take Murakami's experiments with various soft fingertip materials.

Figure 6.2 indicates that for a stable grasp with softfingertips, friction coefficient must be higher than 0.9. Therefore in our models we take friction coefficient as μ=1.0.

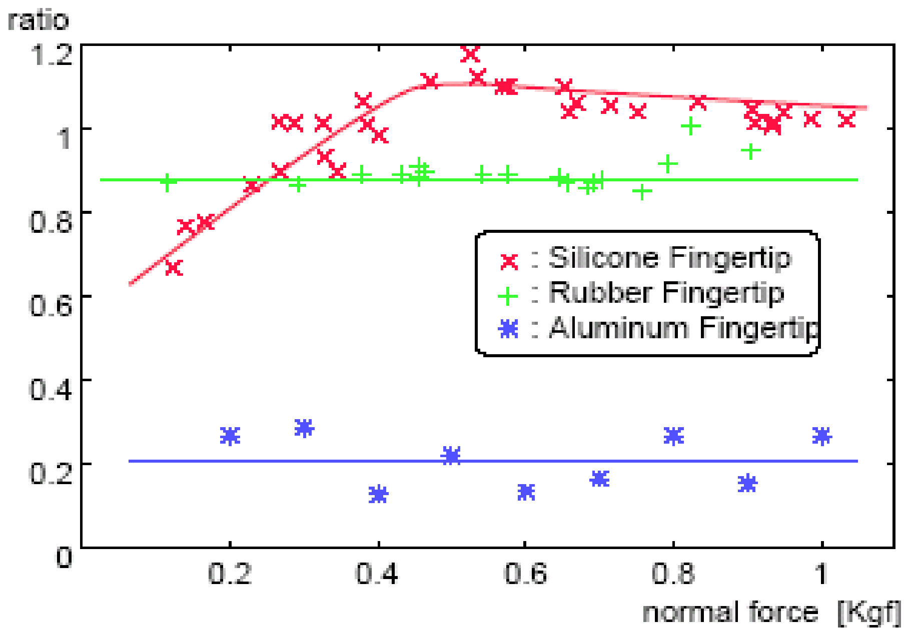


Figure 6.2.Ratio Between Tangential and Normal Force , for Various Soft Finger Contact Materials (Murakami & Hasegawa 2003)

Eventhough our implementation is preshape decision based a virtual grasp using soft finger contact model, which does not correlate itself to an actual hardware grasp execution, , we will analyze sensitivity of the controller in this implementation with respect to contact frictional coefficients. This could guide the reader for an eventual goal of our controller implementation in a particular robot hand hardware. In the following series of figures, we give performance results with various contact frictional coefficients. In virtual grasp case, these coefficent are only effective for grasp matching. In the following tests, we have chosen “Sphere” object to observe contact surface varitions which is directly related to contact normals. The Task is arbitrarily chosen from our sample task space. The other parameters remain unchanged. These parameters are given in fig 6.3 where result 6.1.1 show the normal parameter settings that we begin with.

Result 6.1.1 Contact Coefficients: Ka=0.45, Ko=0.25, Kn=1, Mn=0.8 , μ=1.0

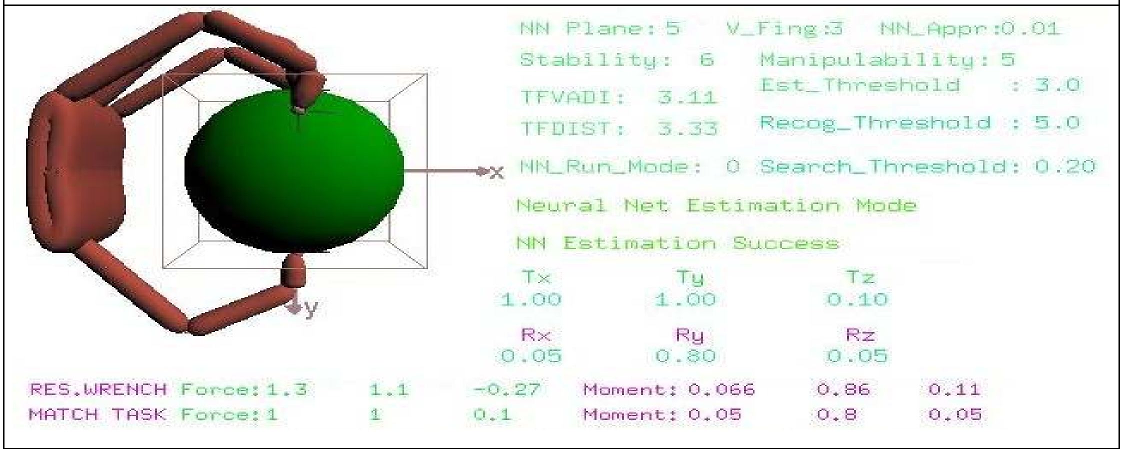


Figure 6.3.Result with Contact Coefficient Set #1

In Result 6.1.1 given in fig 6.3, we present our normal values. Our results in normal executions are performed using these values. In the result , we see that task and grasp wrench magnitudes are at almost equal levels eventhough TFVADI is 3.11 just slightly higher than Est_Threshold.

Result 6.1.2 Contact Coefficients: Ka=0.2, Ko=0.12, Kn=1, Mn=0.95 , μ=1.0

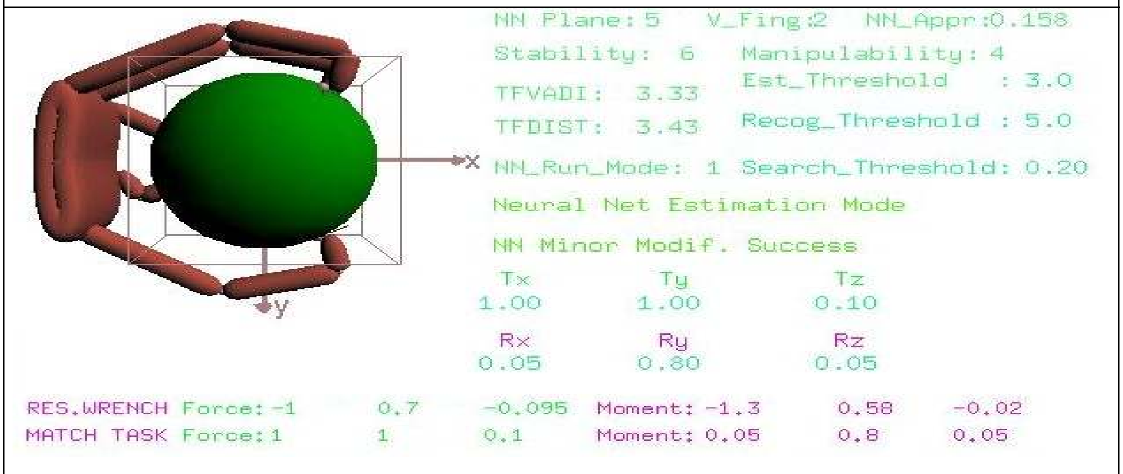


Figure 6.4.Result with Contact Coefficient Set #2

In Result 6.1.2 , we reduce tangential force components Ka and Ko drastically, and slightly increased Mn from 0.8 to 0.95 as shown in fig 6.4. Eventhough controller output a grasp , but, the result is not satisfactory in terms of Task and Grasp Matching. Resultant Wrench is no more relevant to Task eventhough TFVADI is 3.33 which is higher than the one in result 6.1.1.

Result 6.1.3 Contact Coefficients: Ka=0.3, Ko=0.2, Kn=1, Mn=0.9 , μ=1.0

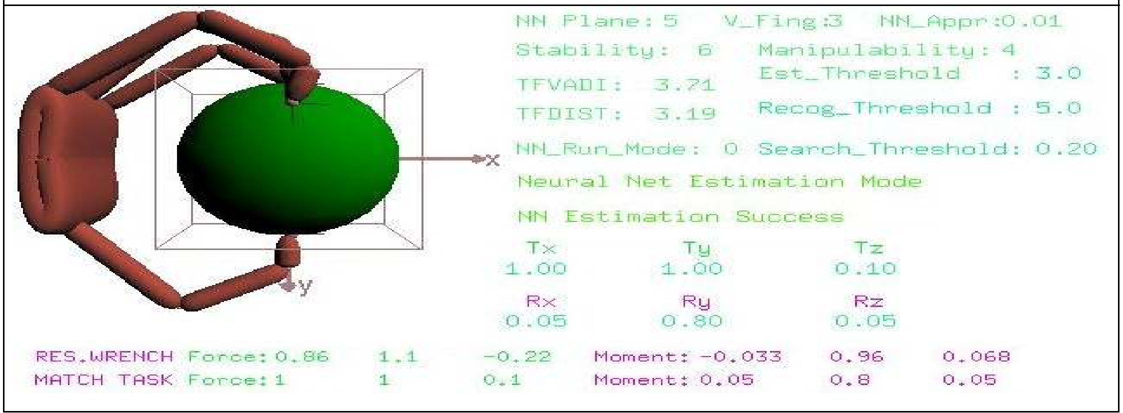


Figure 6.5.Result with Contact Coefficient Set #3

In Result 6.1.3 , we reduce tangential force components by half almost, and slightly increase Mn from 0.8 to 0.9 as shown in fig 6.5. Eventhough controller output a grasp , but, the result is not much satisfactory in terms of Task and Grasp Matching. Resultant Wrench is less relevant to Task eventhough TFVADI is 3.71 which is higher than the one in result 6.1.1. But, this result is better than Result 6.1.2.

Result 6.1.4 Contact Coefficients: Ka=0.4, Ko=0.25, Kn=1, Mn=0.85 , μ=1.0

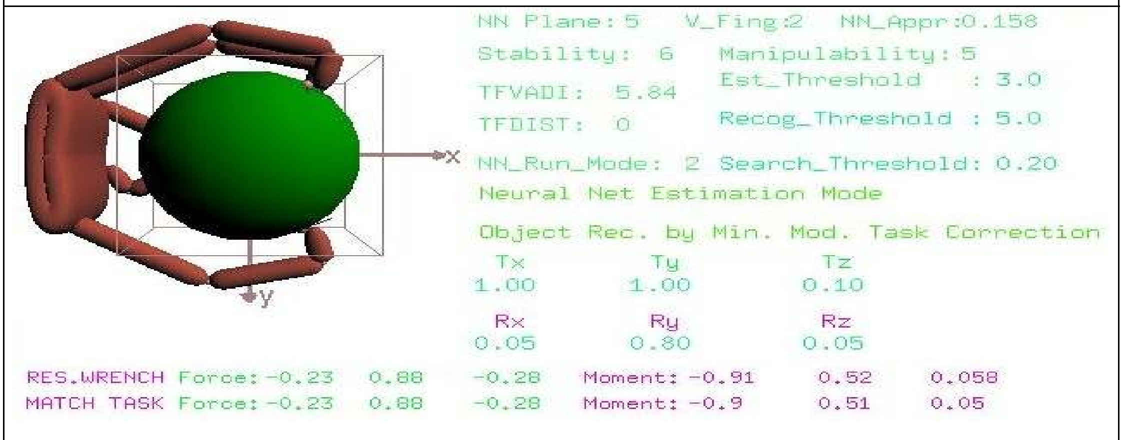


Figure 6.6.Result with Contact Coefficient Set #4

In Result 6.1.4 , we observe a unstable characteristics in this range of coefficifients. The controller did not give a successful grasp , and the result is in terms of object recogition which is occurs in case of a improper task. However, task is unchanged but variation in wrench representation become too much and approximate reasoning of the controller yield very low Est_Threshold and the main output come as recognized object .

Result 6.1.5 Contact Coefficients: Ka=0.5, Ko=0.35, Kn=1, Mn=0.78 , μ=1.0

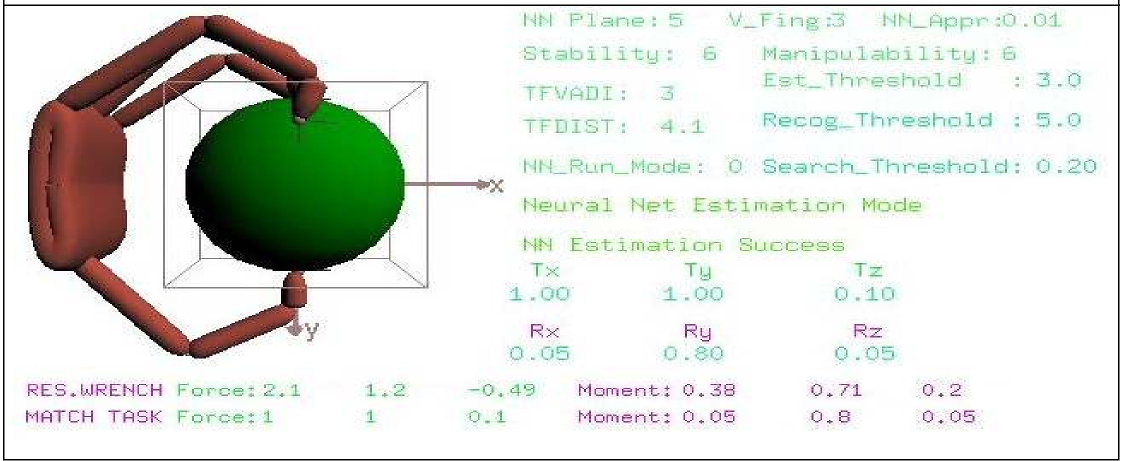


Figure 6.7 Result with Contact Coefficient Set #5

In Result 6.1.5 , we increased tangential force components above our normal values, and slightly decreased Mn from 0.8 to 0.78. The Controller gives a grasp , and the result is satisfactory as Task and Grasp Matching. Resultant Wrench is highly relevant to Task. TFVADI is 3 for this case, it is low due to disturbance components in Tz and Rx coming from increased tangential force coefficients.

Result 6.1.6 Contact Coefficients: Ka=0.6, Ko=0.45, Kn=1, Mn=0.66 , μ=1.0

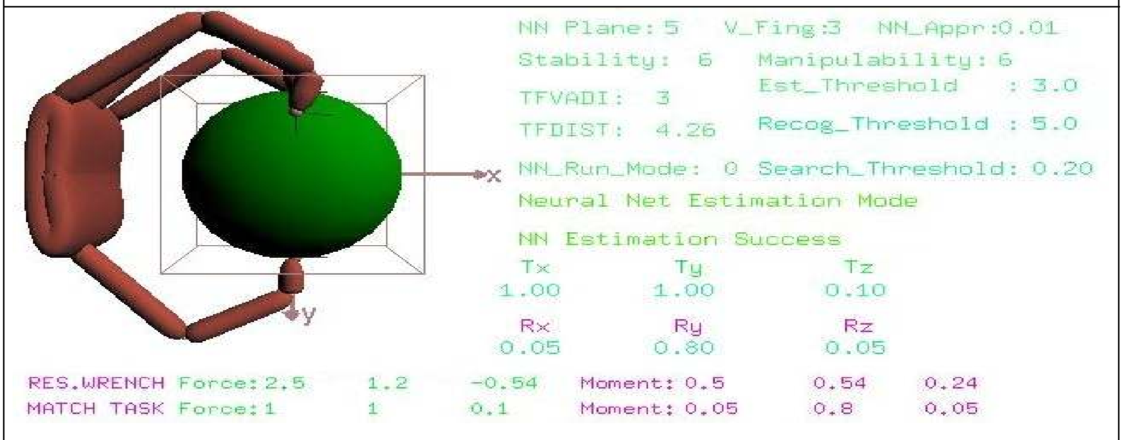


Figure 6.8 Result with Contact Coefficient Set #6

In Result 6.1.6 , we increased tangential force components almost 1.4 times of normal values, and decreased Mn from 0.8 to 0.66. The Controller gives a grasp , and the result is satisfactory as Task and Grasp Matching. But Resultant Wrench is less relevant to Task than previous result and distortion observed due the same reason in result 6.1.6.

Result 6.1.7 Contact Coefficients: Ka=0.7, Ko=0.6, Kn=1, Mn=0.4 , μ=1.0

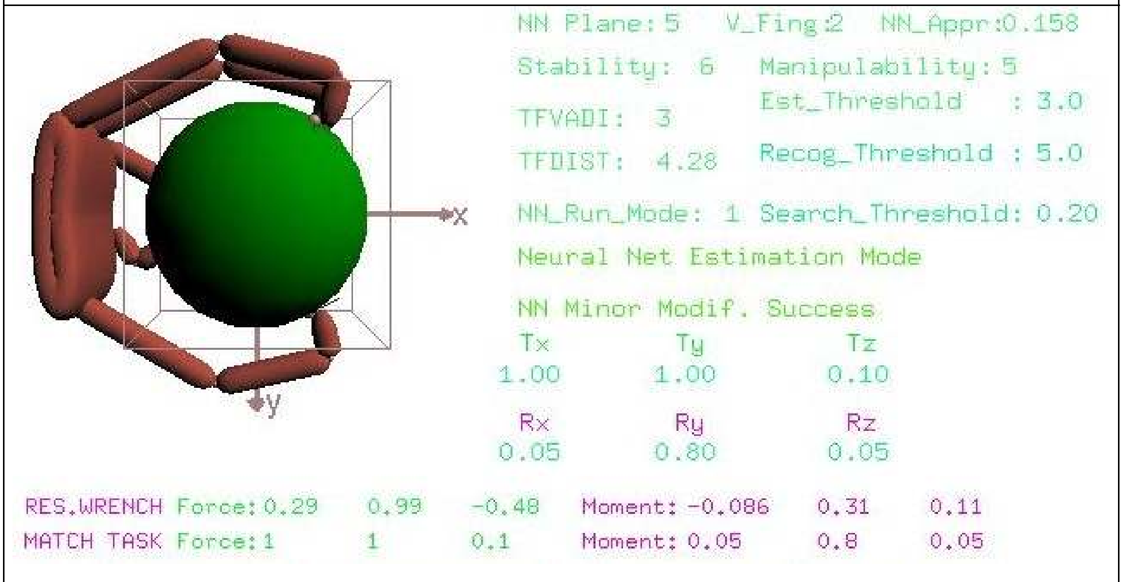


Figure 6.9.Result with Contact Coefficient Set #7

In Result 6.1.7 , we increase tangential force components almost 2 times of normal values, and decreased Mn from 0.8 to 0.4. The Controller gives a grasp , but the result is not satisfactory as Task and Grasp Matching. But Resultant Wrench is highly irrelevant to Task than previous result and obviously a big distortion observed. It is due to disturbance components in Tz and Rz and also reduction in Ry component , coming from increased tangential force coefficients and decreased Mn.
This set constitutes an extereme case.

To conclude about our experiments in analyze effects of contact friction coefficients, we could say that our parameter selection given in figure 6.3 as result 6.1.1, yielded best performance in terms of pattern matching between task and virtual grasp wrench. It is also important to to mention that, variations in contact friction parameters, do not effect plane index and approach determination, but may effect approximate reasoning part of our controller, as it would yield different wrench magnitudes for a given task and object. However, effect of these changes are tolerable by our controller local search mode, execution mode 1 up to 20-30 % variations.

6.2.The Controller Performance with Variations in Fuzzy Logic Membership Functions

The Critic in our Controller makes use of Approximate Reasoning by fuzzy logic. Therefore, it is critical to analyze the effects of fuzzy logic parameter variations on the overall performance of the controller. Mainly, the fuzzy model of the controller uses fuzzification based on Fuzzy set given in section 3.9. In other words, important parameters are there that of the fuzzy membership functions. In the following figures, we present the analysis of the controller under variations of fuzzy memberships, one membership at each step. Again , the Object is Sphere and the task is chosen as the previous one, but we added fuzzified task into figures to represent the effects of variations in membership functions. In our work, both task and grasp wrench use the same Fuzzy Sets.

Result 6.2.1 with Fuzzy MF Changed as Set#1 , Zero=[-0.08 –0.03 0.03 0.08]

Figure 6.10.Result with Fuzzy MF Zero Changed as Set#1

In above result 6.2.1, Narrowing Zero MF (top layer from 0.05 to 0.03 for trapezoids) caused the controller to ouput results in unstable mode, which is not expected and not seen in previous results .

Due to this narrowed zero MF approximately 35 %, the controller could not yield a successful result. This is mainly due to reason that a narrow zero MF, evaluates wrench components of the low distortion values as a contribution to representation and the structure of the wrench becomes away from our representation for generalization. In other words our controller becomes more sensitive to disturbances and estimation errors by means of variation in approximate reasoning structure.

Result 6.2.2 with Fuzzy MF Changed as Set#2 ,Zero=[-0.15 –0.08 0.08 0.15]

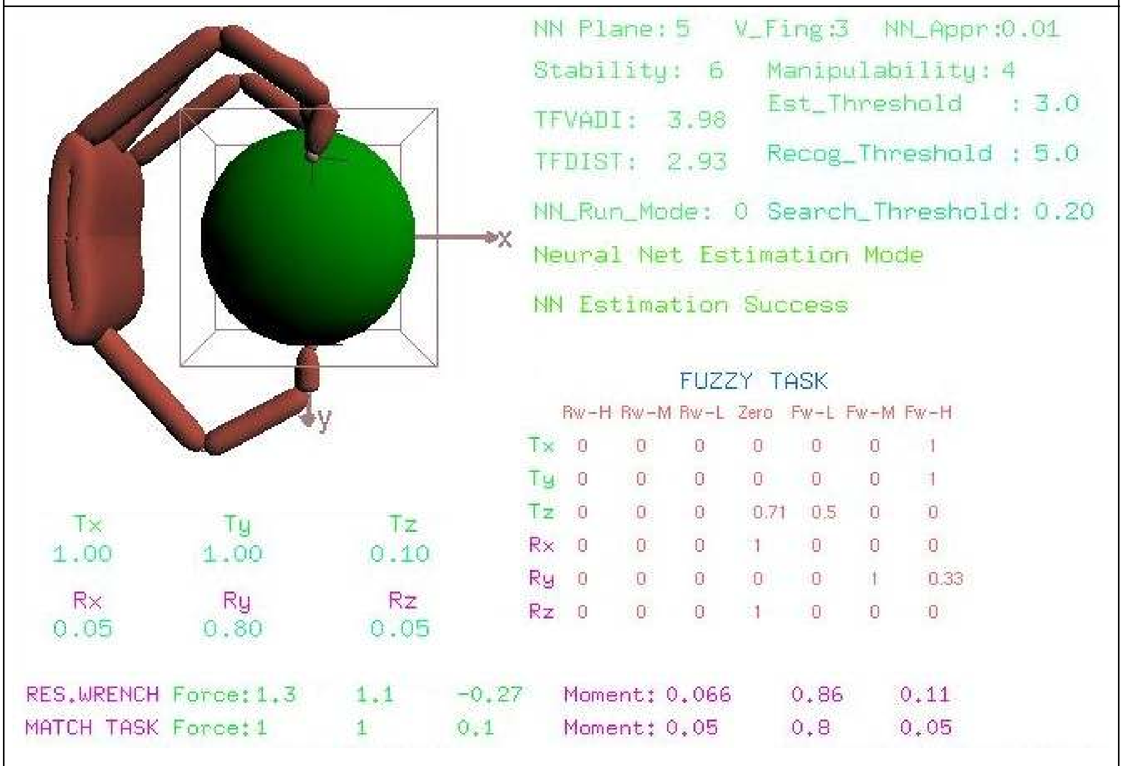


Figure 6.11.Result with Fuzzy MF Zero Change with Set#2

In above result, we Widening Zero MF (top layer from 0.05 to 0.08) results in a stable mode. The reason for that is same reason explained in narrowing case. The result is as normally expected.

Result 6.2.3 Fuzzy MF Changed Rwd-L=[-0.4 –0.25 -0.15 -0.08],Fwd-L symmet.

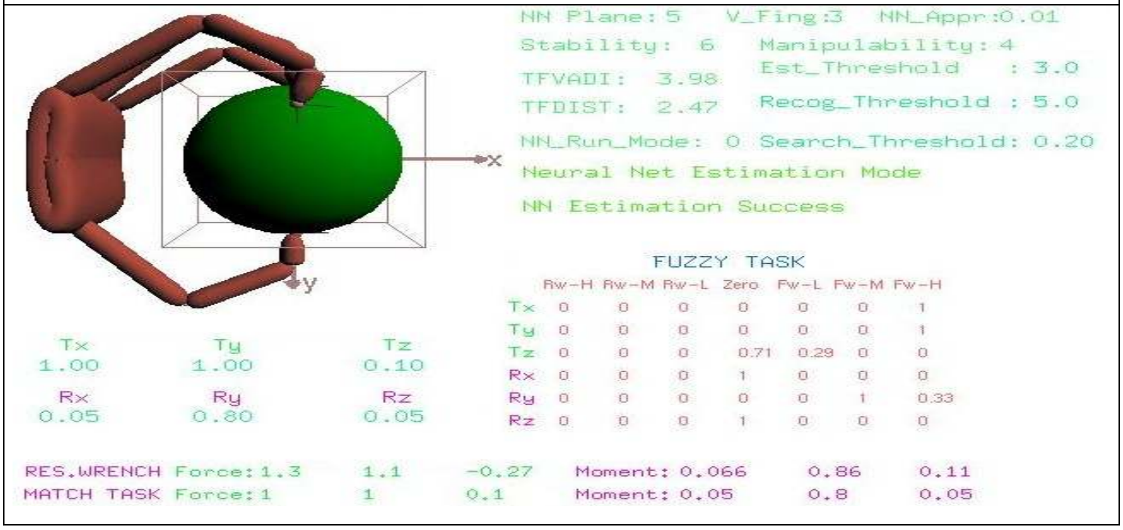


Figure 6.12.Result with Fuzzy MF Rw-L and Fw-L Changed with Set#3

In above result, Narrowing xx-L MF results in stable mode, effects can be seen in fuzzified task of the figure as change in MF value ,but no variation in wrench as a result.

Result 6.2.4 Fuzzy MF Changed Rwd-L=[-0.6 –0.4 -0.15 -0.05],Fwd-L symmet.

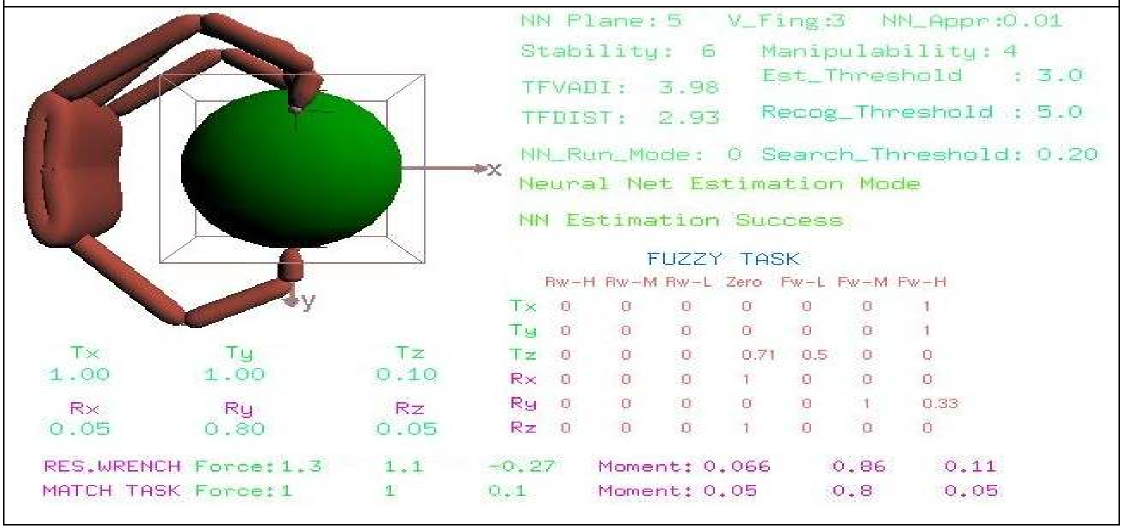


Figure 6.13.Result with Fuzzy MF Rw-L and Fw-L Changed with Set#4

In above result, Widening xx-L MF results in stable mode, effects can be seen in fuzzified task of the figure as change in MF value, but no variation in wrench as a result, and no variation in TFVADI , 3.98.

Result 6.2.5 Fuzzy MF Changed Rwd-M=[-0.9 –0.7 -0.55 -0.35],Fwd-M symmet.

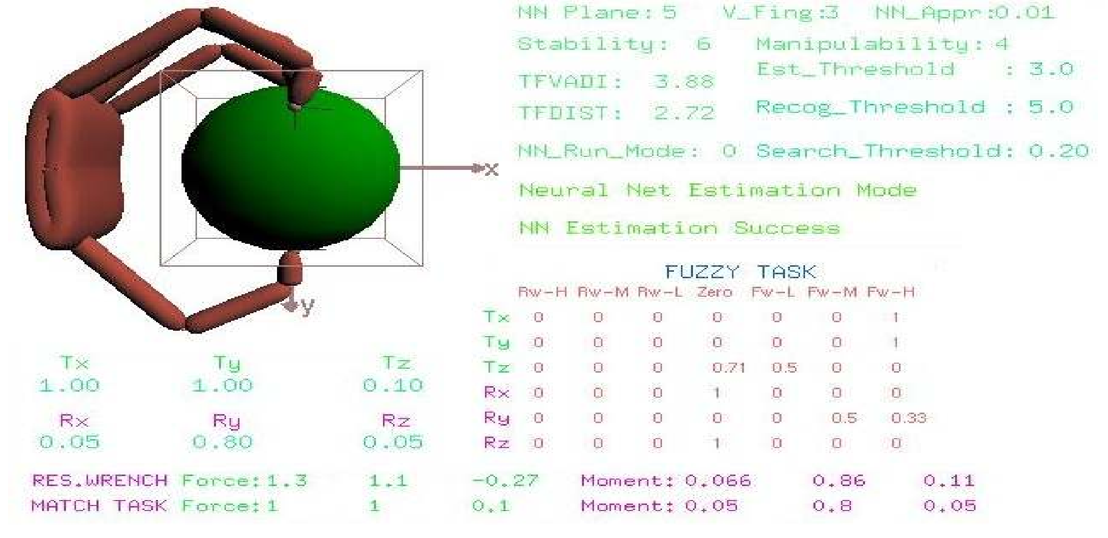


Figure 6.14.Result with Fuzzy MF Rw-M and Fw-M Changed with Set#5

In above result, Narrowing xx-M MF results in stable mode, effects can be seen in fuzzified task of the figure as change in MF value, but no variation in wrench as a result, and a negligible variation in TFVADI , 3.88.

Result 6.2.6 Fuzzy MF Change Rwd-M=[-0.95 –0.85 -0.45 -0.35],Fwd-M symmet.

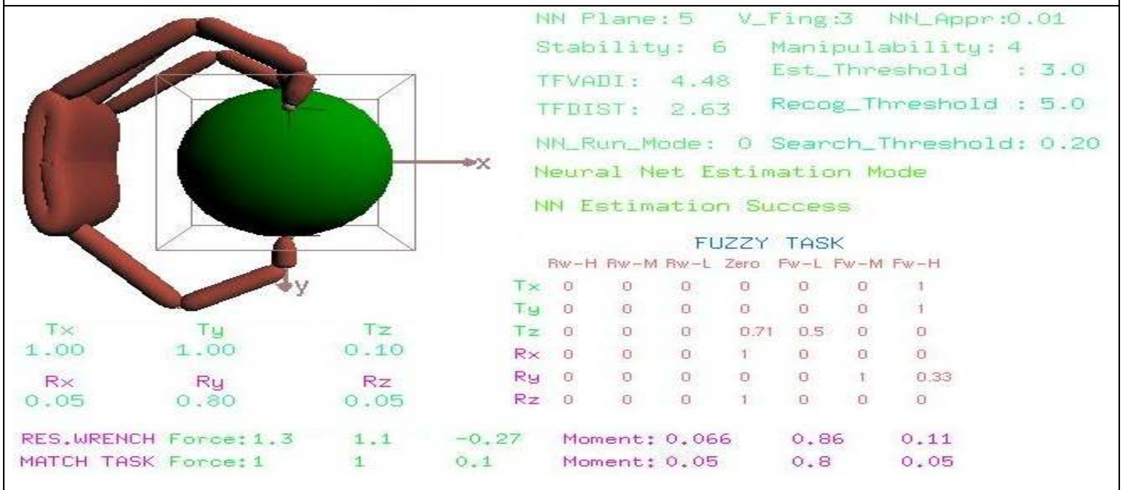


Figure 6.15.Result with Fuzzy MF Rw-M and Fw-M Changed with Set#6

In above result, Widened xx-M MF results in stable mode, effects can be seen in fuzzified task of the figure as change in MF value, but no variation in wrench as a result, and a little variation in TFVADI , 4.48.

Result 6.2.7 Fuzzy MF Changed Rwd-H=[-1.2 –1.2 -0.9 -085],Fwd-H symmet.

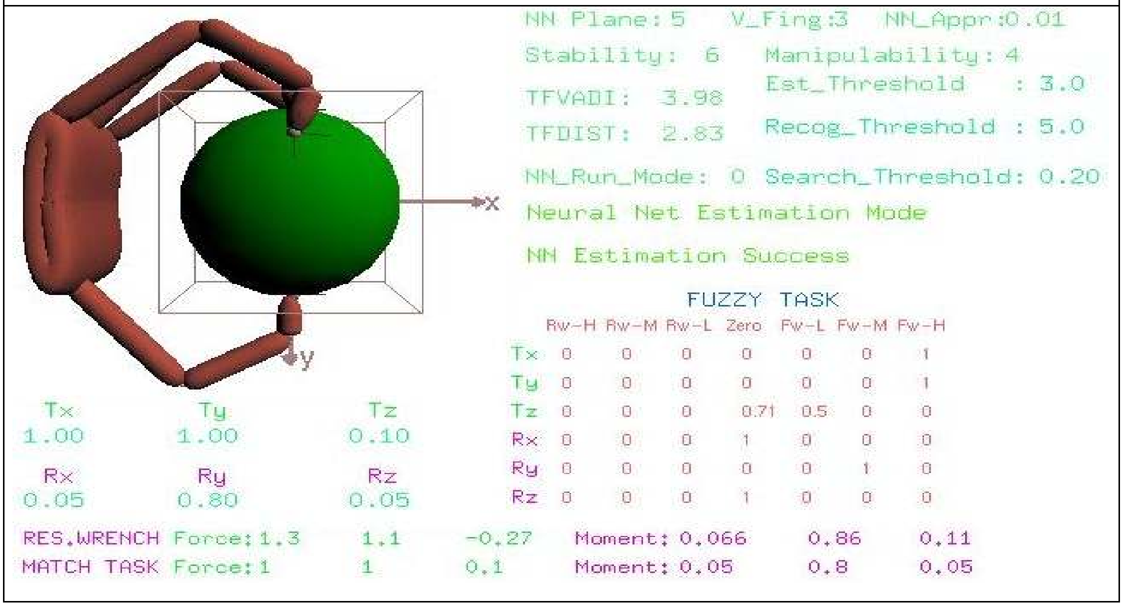


Figure 6.16 .Result with Fuzzy MF Rw-H and Fw-H Changed with Set#7

In above result, Narrowing xx-H MF results in stable mode, normally expected.

After parameter variations in Fuzzy Membership Functions either narrowing or widening about 20-30 % of the trapezodial membership functions one at a time, we reached the following conclusion about the effect of the MF to our controller. As it could be seen from figures, MF variations mainly effective for ZERO MF , for the other varitions, even there is small change in TFVADI values between 3.88 to 4.4 , however overall grasp preshape is same for other MF . This is an indication of that the controller uses approximate reasoning that is not effected much by parametrization due to fuzzy logic attribute matching technique. As we also observe that grasp wrench values do not change becasue approach plane and approach value is independently determined by ANN, meanwhile TFVADI slightly varies due to changes in MF from 3.8 to 4.4. These variations do not effect the result of the controller using the threshold values given. In general, when the contact parameters changed and/or task representation also changed the MF needs to be optimized for a this new structure to have a proper mapping. It is also necessary to optimize final hardware structure based on experimental results. This optimization will demonsrate the actual sensitiviy of the system on MF.

6.3.The Controller Performance with Variations in Neural Network Parameters

The Controller is based on ANN structure for its execution. Therefore, ANN parameters are very important for the overall performance of the controller. The sensitivity of the controller with respect to parameters of ANN is also important. Howevever, the estimations for the optimal parameters for ANN was performed mainly in the design phase of the controller in our work. The runtime performance of the controller is dependent on ANN parameters in the learning mode , that is execution mode 3. Learning rate parameter is the basic parameter. In order to analyze the effects of ANN parameters in the learning mode of controller execution, we carried out some experiments, however, we could simply observe that controller is more dependent on thresholds than ANN learning rates. We could not observe an obvious effect of Learning rates. In experiments, we could only observe that after some limit of thresholds, controller goes into Learning mode but if it is tested with these thresholds, it generally fail. However, if thresholds are slightly reduced, the controller shows a successful performance. This situtation is shown in figure 6.17. and figure 6.18. The only observable effect of this parameter is at pretraining phase by the number of epoch which is given in detail in Appendix A.

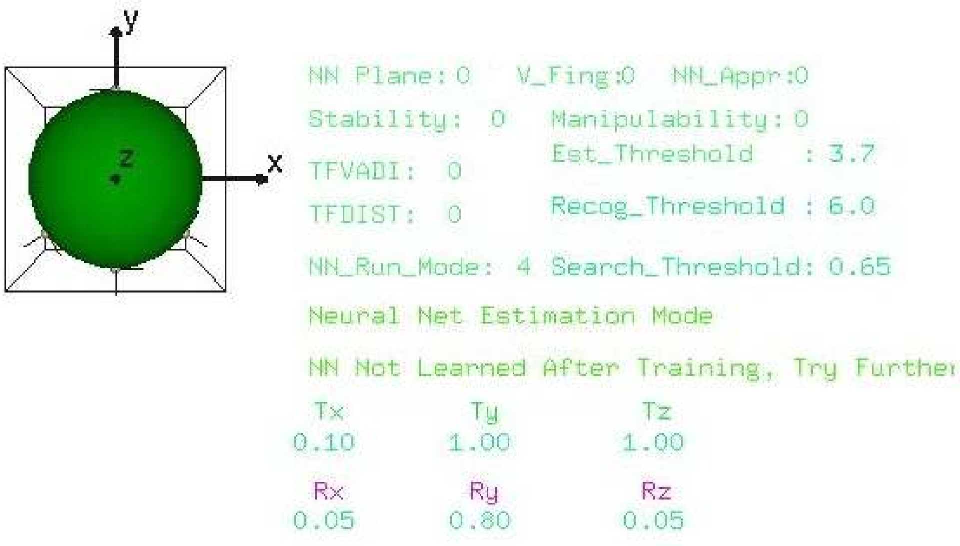


Figure 6.17.ANN After Training Indicating Failure

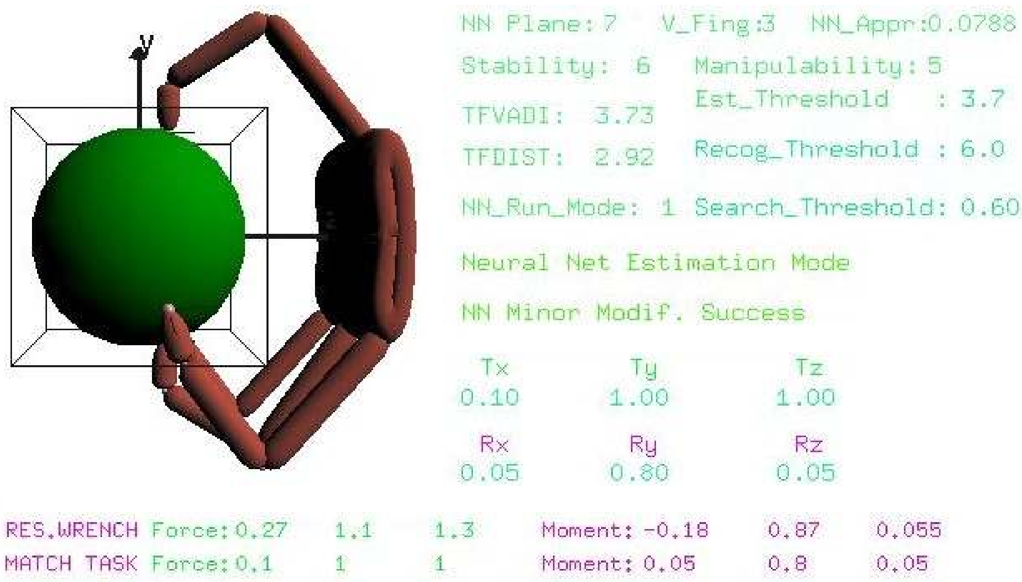


Figure 6.18.ANN After Training Indicating Success with Small Threshold Change.

In order to give an idea about our parameter optimization of ANN, we present the following figure 6.18. The figure indicates the optimal surface for hidden nodes and learning rates wit respect to number of epochs in pretraining phase. The details of the pretraining data is given in Appendix A.

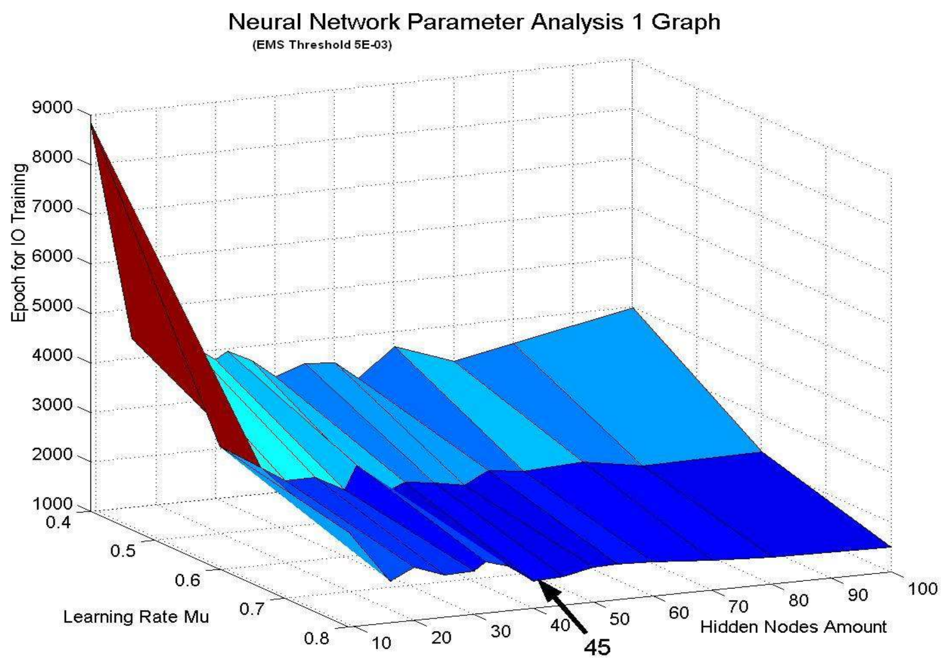


Figure 6.19.Surface Plot for ANN Optimal Parameter Analysis .

6.4.The Controller Performance with Variations in Thresholds for Critics .

The Controller is a multistage decision controller and transition between stages are determined by the internal critic by using a relevant threshold. Therefore, analysis of the effects of the thresholds is an important point for our performance analysis. In this section ,we analyze the effects of each threshold type one by one. In some controller stage or execution mode, only a single threshold is effective whereas in some other stage , a combination of thresholds is effective. First, we start with the Estimation Threshold. Again in this test, we select Sphere as test object for geometric reasons and the task is another arbitrary task from our task space.

Estimation Threshold Effects:

Effects of the Variations in Estimation Threshold could be observed in the following results.

Figure 6.20.Result for Estimation Threshold Effects Set #1.

We start with a basic limit value 3.0 as Estimation Threshold. We take this value since in our task representation we represent task with 3 nonzero wrench item. In figure 6.10 , we observe that our controller decides normal execution mode without any local search. The task and grasp wrench indicates the compliance between them so result is successful.

Figure 6.21.Result for Estimation Threshold Effects Set #2.

In this set , we increased Estimation Threshold from 3.0 to 3.5 and we observe that to satisfy that our controller runs at execution mode 1 which means a local approach search is carried in the decided approach plane. In this mode, Search Threshold is also effective in combination to Estimation Threshold but we narrowed the search limit up to 0.52.

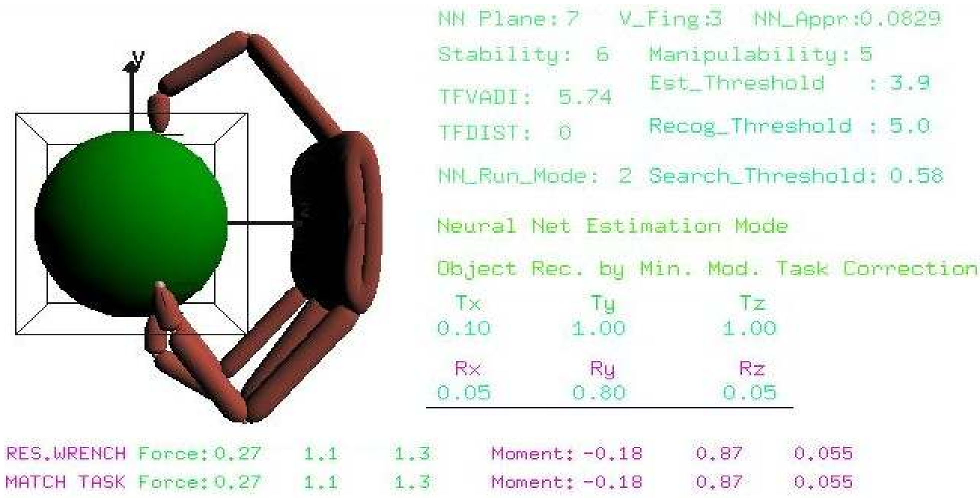


Figure 6.22. Result for Estimation Threshold Effects Set #3.

As we further increase Estimation Threshold to 3.9, as in figure 6.12, our controller decides that the required task does not match the object grasp with this value and it switches to execution mode 2 which means object recognition. We observe from fig 6.12 that object recognition is done with a very high compliance.

In this mode, Search Threshold is also effective in combination to Recognition Threshold becasue Estimation threshold initiated transfer to other stage and it lost its effectiveness.

Figure 6.23. Result for Estimation Threshold Effects Set #4.

Search Threshold Effects:

Search Threshold is the limit for local approach search in the given approach plane . When the threshold increased, it decreases the approach search limit.

Effects of the Variations in Search Threshold could be observed in the following results. The Controller is at execution mode 1 and the test starts with a narrow search limit, 0.55.

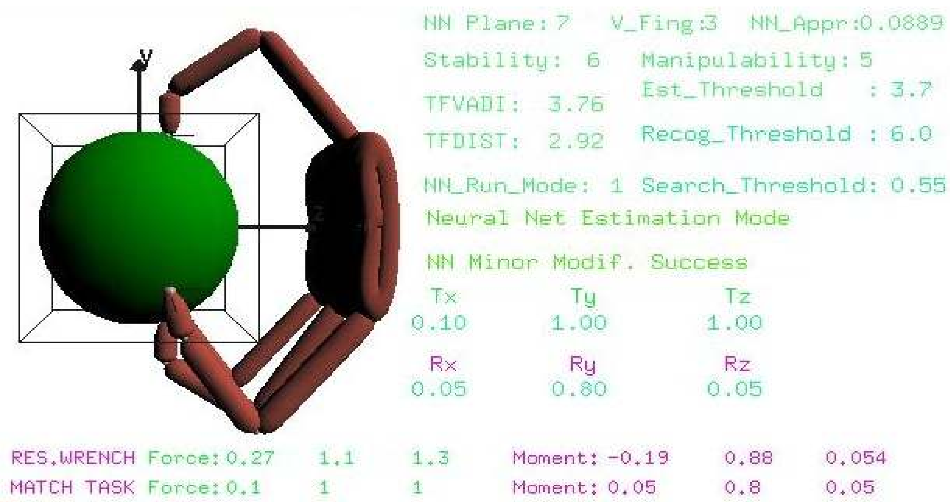


Figure 6.24. Result for Search Threshold Effects Set #1.

Figure 6.25. Result for Search Threshold Effects Set #2.

Search Threshold is increased slightly from 0.55 to 0.5 . Eventhough, the Controller yielded successful preshape with same matching degree TFVADI 3.77 but changes occured in grasp wrench structure as distortions like it was in contact parameters.

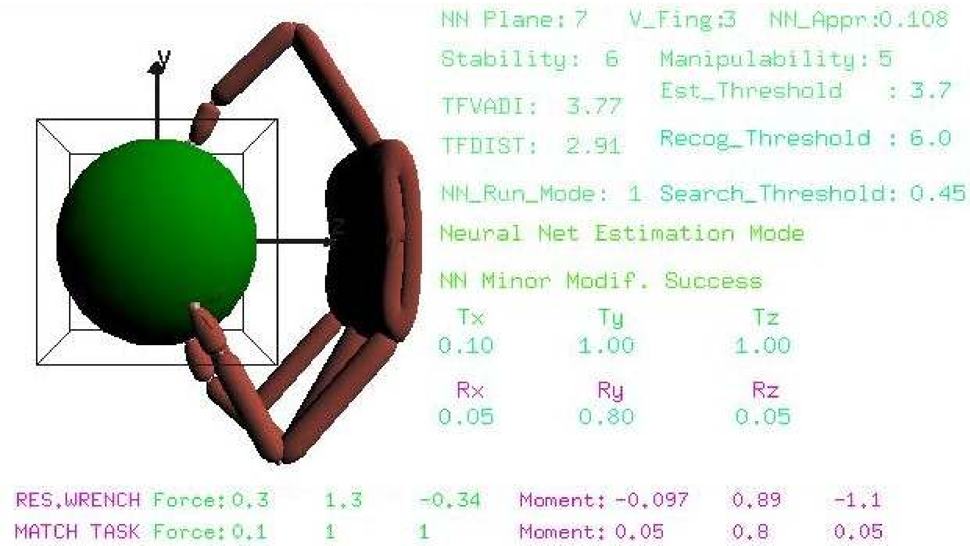


Figure 6.26. Result for Search Threshold Effects Set #3

Search Threshold is increased slightly from 0.5 to 0.45 . Eventhough, the Controller yielded successful preshape with same matching degree TFVADI 3.77 but changes occured in grasp wrench structure as distortions like it was in contact parameters.

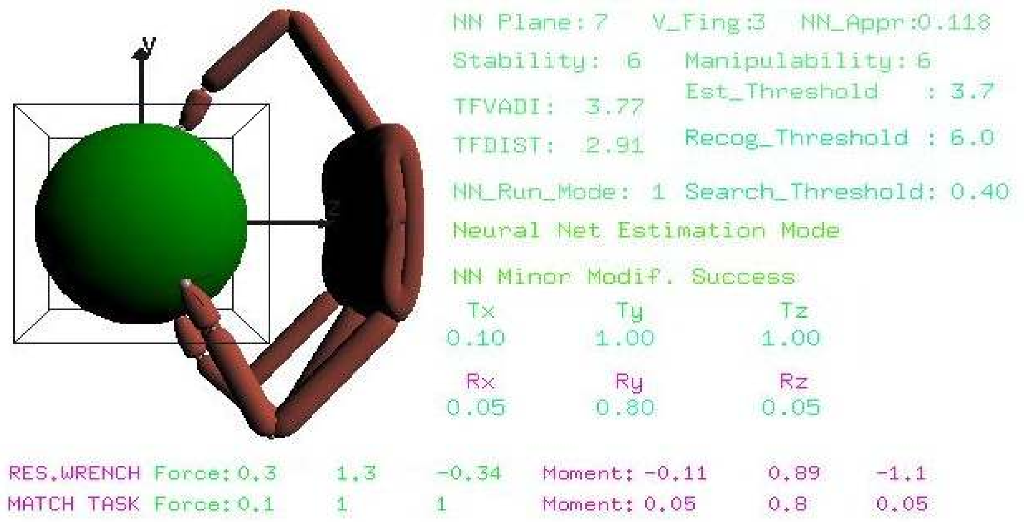


Figure 6.27. Result for Search Threshold Effects Set #4

Search Threshold is increased slightly from 0.45 to 0.4 which effected the approach value. Eventhough, the Controller yielded successful preshape with same matching degree TFVADI 3.77 but negligable changes occured in grasp wrench structure as distortions.

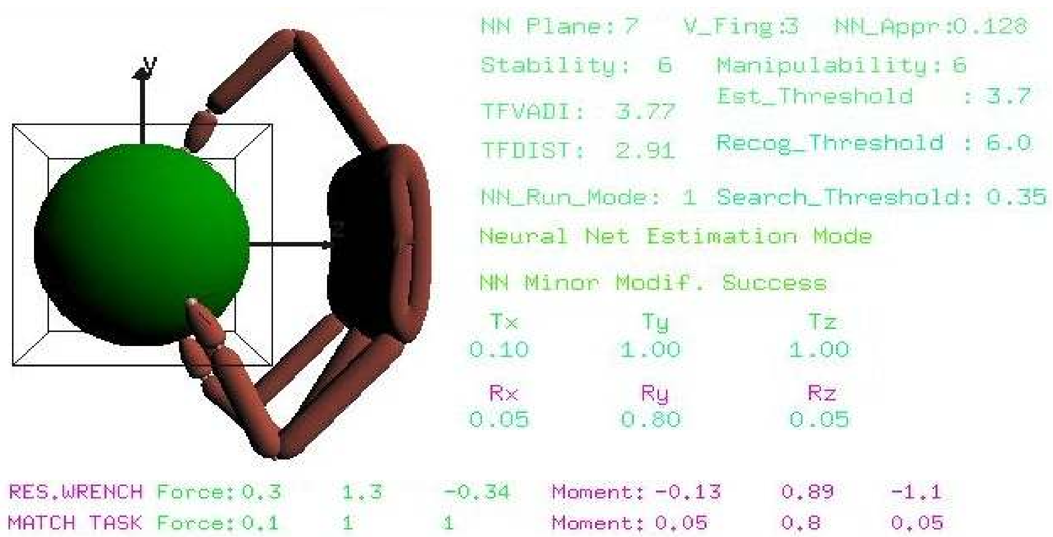


Figure 6.28.Result for Search Threshold Effects Set #5

Search Threshold is increased slightly from 0.4 to 0.35 . Eventhough, the Controller yielded successful preshape with same matching degree TFVADI 3.77 but negligable changes occured in grasp wrench structure as distortions .

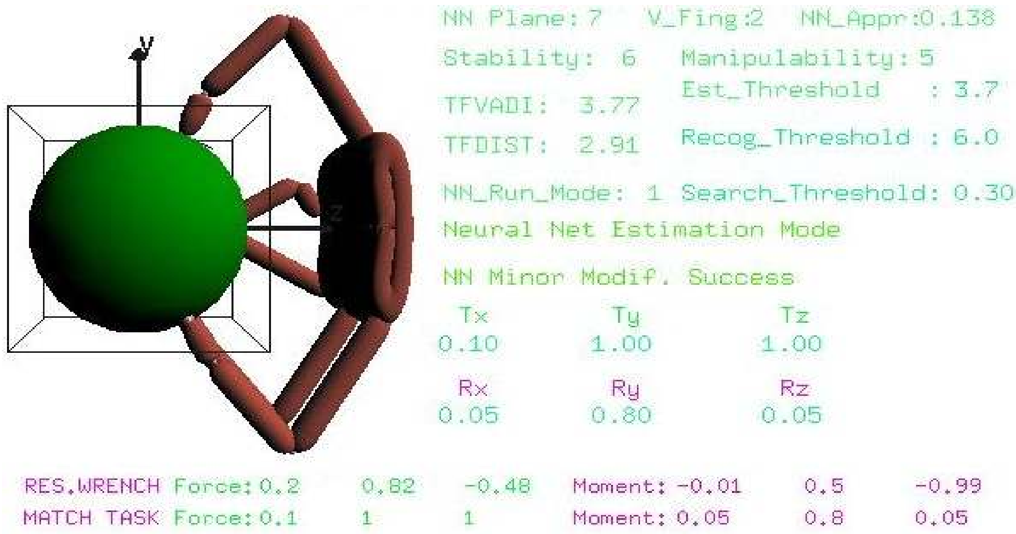


Figure 6.29.Result for Search Threshold Effects Set #6

Search Threshold is increased slightly from 0.35 to 0.3 . Due to dynamic depth estimations, number of finger has changed from 3 to 2 hence, resultant grasp wrench also changed . Eventhough, the Controller yielded successful preshape with same matching degree but minor changes occured in grasp wrench structure.

Figure 6.30.Result for Search Threshold Effects Set #7

Search Threshold is increased slightly from 0.3 to 0.25 . Eventhough, the Controller yielded successful preshape with same matching degree but minor changes occured in grasp wrench structure, as shown in figure above.

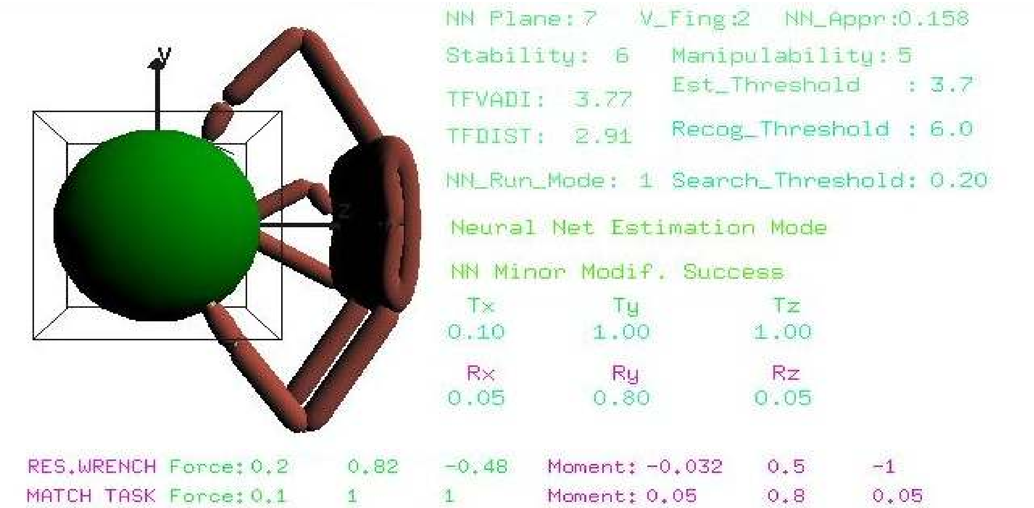


Figure 6.31.Result for Search Threshold Effects Set #8

Search Threshold is increased slightly from 0.3 to 0.25 . Eventhough, the Controller yielded successful preshape with same matching degree but minor changes occured in grasp wrench structure, as shown in figure above.

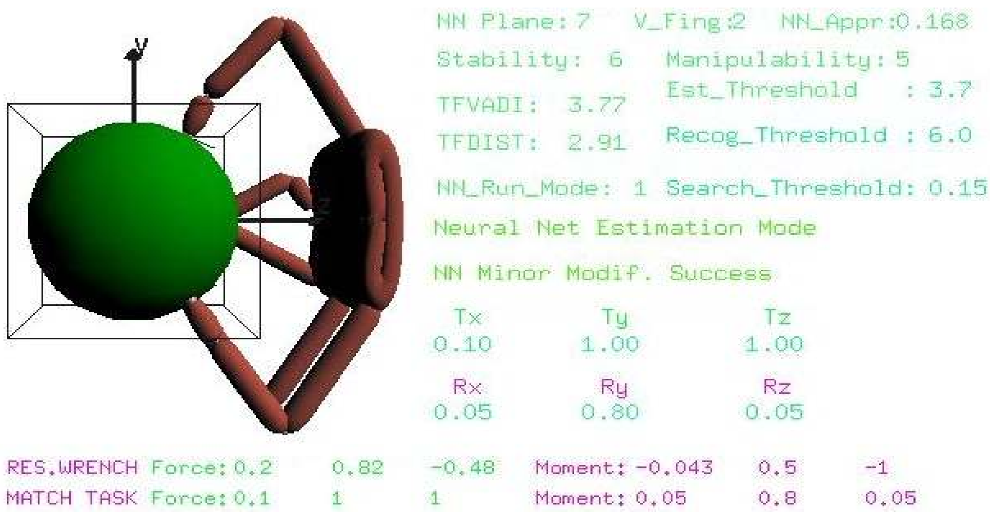


Figure 6.32 Result for Search Threshold Effects Set #9

Search Threshold is increased slightly from 0.3 to 0.25 . Eventhough, the Controller yielded successful preshape with same matching degree but minor changes occured in grasp wrench structure, as shown in figure above.

Recognition Threshold Effects:

Recognition Threshold is the limit for recogontion of an object to understand whether object previously introduced to controller or not. It is executed with a applicable task on the object under test. When the threshold is increased, a higher degree of matching for object recognition is asked from controller. Effects of the Variations in Recognition Threshold could be observed in the following results. The Controller is at execution mode 2 and the test started with a limit 5.5. Effects of Variations in Recognition Threshold could be observed with the following results.

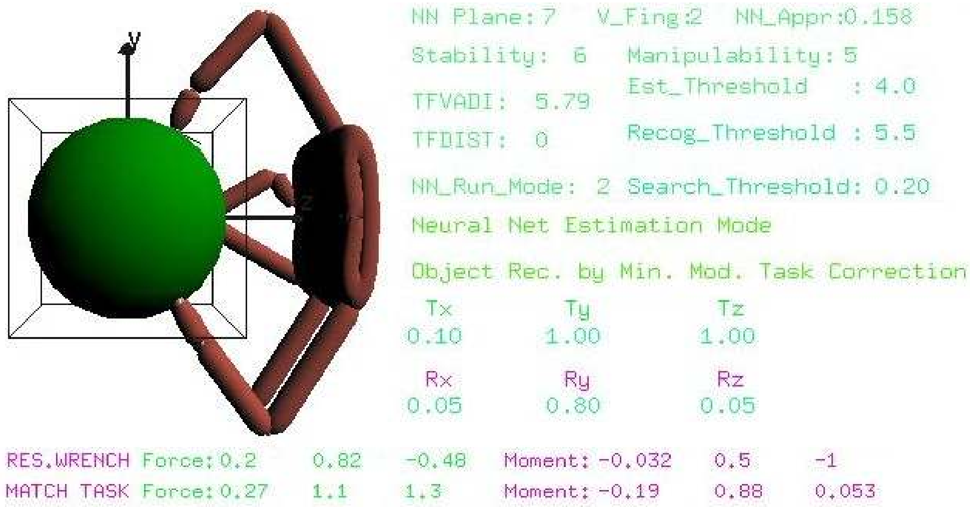


Figure 6.33.Result for Recognition Threshold Effects Set #1

Recognition Threshold 5.5 and Search Threshold 0.2 .

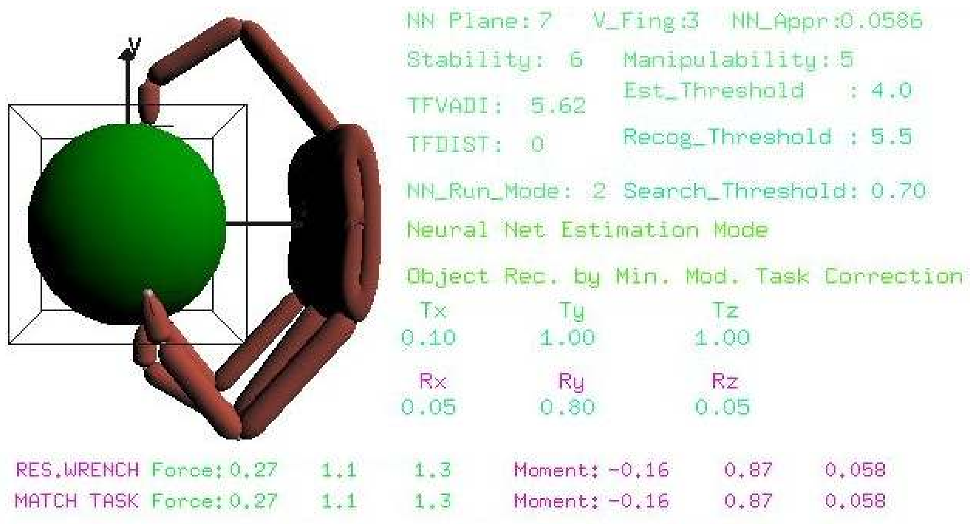


Figure 6.34.Result for Recognition Threshold Effects Set #2

Recognition Threshold 5.5 and Search Threshold 0.7 .

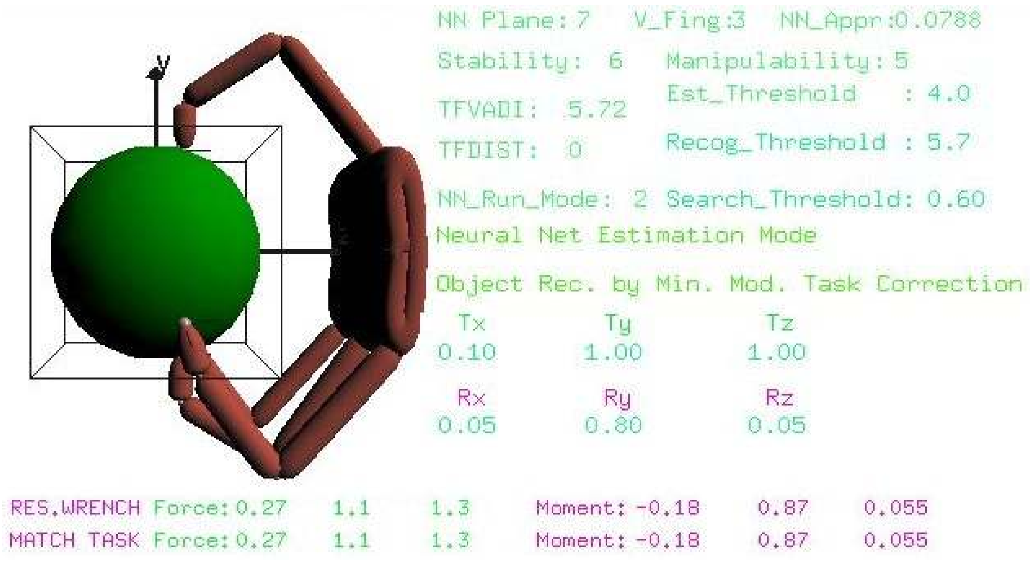


Figure 6.35.Result for Recognition Threshold Effects Set #3

Recognition Threshold 5.7 and Search Threshold 0.6 . Normal case

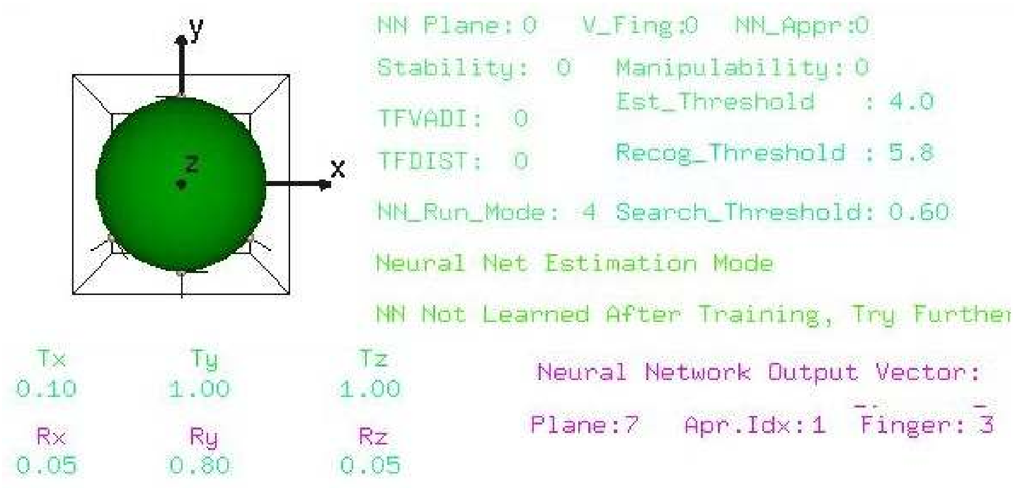


Figure 6.36.Result for Recognition Threshold Effects Set #4

Recognition Threshold 5.8 and Search Threshold 0.6 . Failure case due to high threshold value. It is not possible to reach this value with this object and task.

In Recognition threshold, for values starting from 5.5, in fig 6.23, we gradually increased up to 5.7 in fig 6.26 as well as minor modification in search threshold . In these experiments, we observed that multi-stage decision mechanism is dependent on thresholds very much. In a very small range of 5.5 to 5.7, the controller changes its response. Therefore, it may be better to use recognition threshold around 5 to 5.5. And this range becomes sufficien for our object framework. For the other cases of implementation, other values may be to experimentally found.

In our threshold analysis, we observed that the system is most dependent on thresholds and especially search threshold effected controller overall performance since it acts in combination with other thresholds. In order to find optimum values for thresholds, it is necessary to perform a certain number of experiments for each implementation itself and threshold are not a generalizable.

6.5 The Controller Performance with Different Pretraining Object Model

In order to analyze performance of controller as response to changes in Pretraining Object, we selected Sphere from our primitive object set. After this selection using our workplane analysis tool, we develeoped Pretraining data. With this data, we observed following cases:

Pretraining Phase: In this phase Controller Neural Network structure is trained using same number of hidden nodes, 45 and learning rates 0.6 and 0.5 for hidden and output layers successively. At this phase, in training trials , we noticed that Epoch increased from 9500-11000 interval to 33800-34000 interval, which indicates that learning propagation thru same network become more difficult with this set and data used for training is not highly representative for task space classification.

After Training Phase: In this phase, we tested the learning performance of the controller with training object and data. Following figures 6.5.1, fig 6.5.2 and fig 6.5.3 indicates the learning performance of the controller. As it could be seen from figures, the controller did not remember even its pretraining object . It also did not learned cylinder object. After 2 more trainings, it eventually recognized the sphere.

To conclude about this variation, as we might remember we partition the task space via pretraining data, however, data obtained from sphere model does not represent this partion sufficiently. Therefore our choice for selection of cube as pretraining object yields better performance.

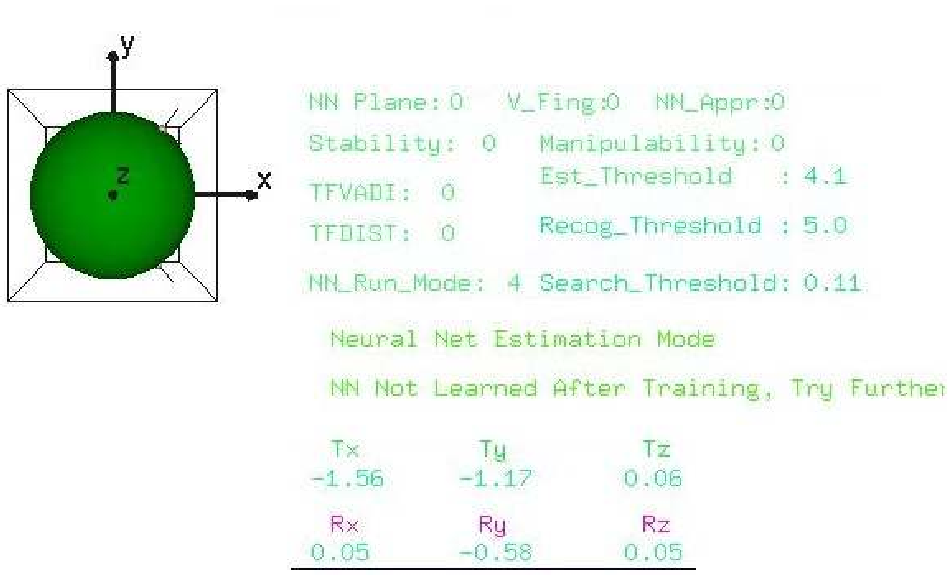


Figure 6.37.Sphere Tested after Pretraining with Its Own Training Data.

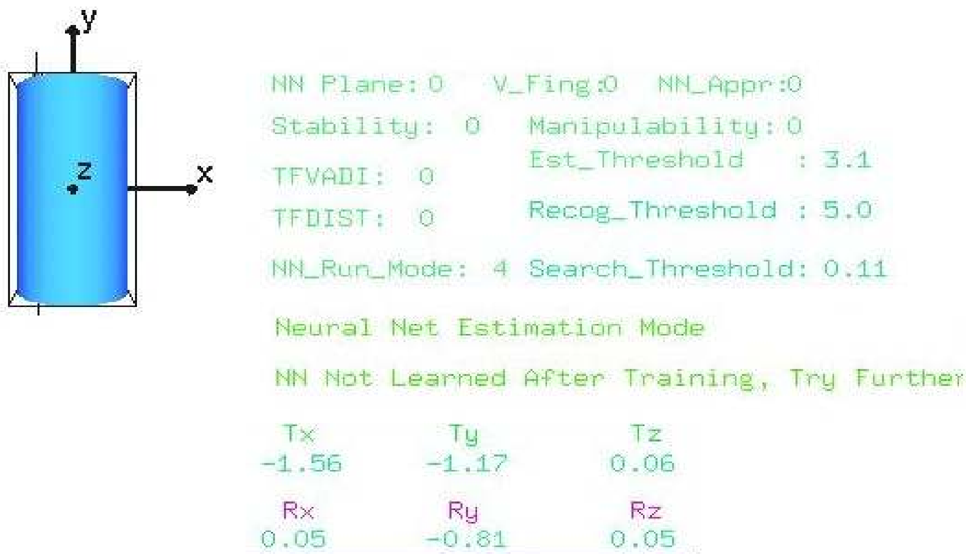


Figure 6.38.Cylinder Tested after Pretraining with a Sample from Pretraining Set.

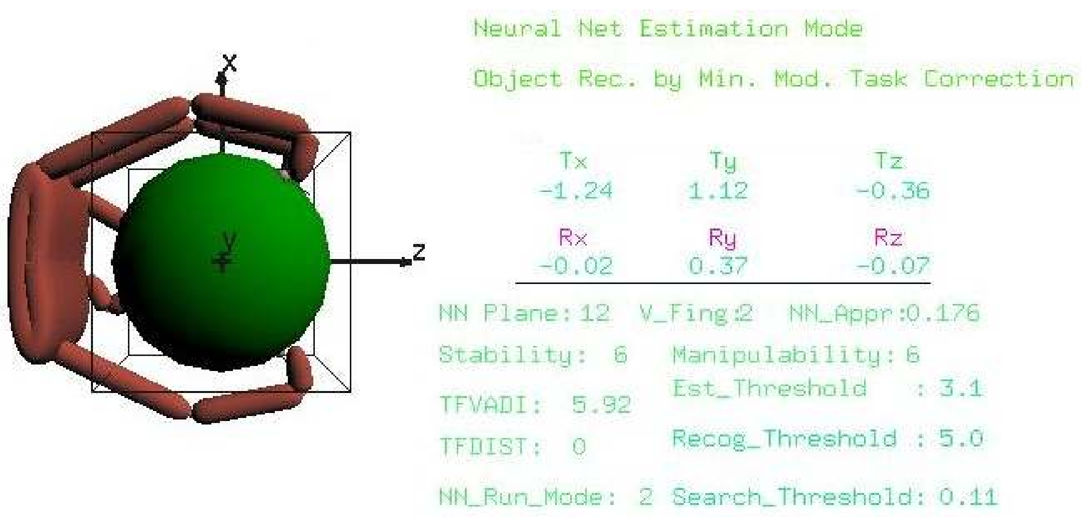


Figure 6.39.Object Recognition, Sphere Tested with a Sample Pretraining Data .

6.6 Summary for the Controller Performance analysis

As a general conclusion about performance of our system, we could state that, our model due to its nature is much more dependent on the Threshold Values rather than other parameters such as ANN or Fuzzy MF , which is not as expected. Dependency to contact coefficients is confined to some values found in previous experimental works. We perform limited parametric test for Fuzzy MF and Fuzzy MF parameters could be further tested for very extreme case ,too .Therefore, for an optimization work, experiments should extend into these parameters in more detail. Since the scope of our work at implementation phase is very broad, the detailed analysis of the parameters needed to be extended in further studies.

7.CONCLUSIONS AND FUTURE WORKS

7.1 Conclusions

In this thesis, for a multifingered robot hand, a new model of grasp preshaping controller is designed and the model is implemented in a software simulation environment. Our Preshape Controller involves grasp planning and grasp synthesis.

Our Preshape Controller use attributes coming from each grasp phases to deduce a successful preshape for a good initiation of a subsequent grasp task. To this end, we have combined task, 3D object model and the grasp kinematics for grasp planning and preshape selection phase. That is to say, we interpret task as a spatial displacement in wrench form and partition grasp space into approach planes. We use existing contact models, to estimate kinematic effects of the possible virtual grasps on object 3D model, to obtain representation pattern of the grasp , in wrench form. We also introduce new symbolic grasp measures which are stability and manipulability. These measures are in justification and inferencing stages supporting our inference system. Using approximate reasoning based on fuzzy logic, we evaluate the matching of the desired task and possible grasp postures. ANN structure is used as for increcemntal learning element . And finally, we integrate all of these methods in to the modular architecture of an Intelligent Controller.. We have tested our Intelligent Controller with a basic grasp simulation for its performance.

A main advantageous point for our controller is that, it takes into account the interpretation of Grasp synthesis in two major components, Preshaping Control and Grasping Initialization. We concentrate in Preshaping Control in our work and analyze Grasping as a Virtual Grasp. The designed controller is implemented using 3D computer simulation environment which enables the controller could be implemented in real hardware environment.

Our Preshape Controller executes attributes in symbolic forms, which can be easily integrate a higher level robot control mechanisms. The Intelligent Preshape

Controller, also handled some uncertainity forms of contacts, collisions, exact grasp analysis estimation errors in simulation.

7.2 Future Works

After stating the contributions made and conclusions about the thesis work, we would like to express possible future extensions of the our work as follows:

Main extension item of this work will be hardware based implementation of the Intelligent Controller, to perform online synthesis of Grasp Preshape. This is critically important point to make the final conclusion about the performance and efficiency of the method that we proposed and implemented in simulation due to facilities available.

As a practising engineer, based on my commissioning experience on field applications of the projects that I have designed in Industrial Automation, I could say that it is expected to tune up software model and assumptions concerning the model and environments during hardware experiments. In practical engineering , there is a well known statement that “Nothing beats the Experience”. By the experience , we mean actual hardware experiments based on a model which is developed using scientific methods, not just trials and errors. Even before hardware implementation , I can foresee that there will be some optimization process for the controller, and it will be determined by the hardware architecture and other environmental conditions. Since The Intelligent Controller implemented in this thesis work, has run in a simulation environment which is parameterized to adapt itself, hence it could be interpreted in an actual hardware case without major changes. We expect some modificiation in the structure and methods. Mainly , there will be additional modules such as for vision based 3D model acquisition units and combination units for interfacing the hand low level kinematic and dynamic force control and tactile sensor fusion.

In addition, hardware collision detection mechanism by tactile sensors , needs to be accompanied which will also be an additional criteria for grasp preshape execution. Finally, some of the software code may be deleted which becomes unnecssary in hardware implementation such as grasp simulation.

Other issues for the implementation of the controller in simulation environment can be stated as follows for further improvements:

First of all, we have a few restrictions on our object models , for instance, necessity of being convex, represented by outer shell surface and size limitations. In order to overcome these limitations, geometric object decomposition techniques could be added in modelling part and assumptions could be updated accordingly.

An additional point is that hand grasp simulation could be improved to represent a more realistic simulation. In our case as we stated before, grasp simulation is just for demonstrating applicability of the controller. However, it is possible extend simulation by adding whole workspace collision detection mechanisms and by adding hand dynamics and object model dynamics. After that, it could be also possible to combine dynamics of grasp performed on object model to grasp synthesis.

Moreover, hand simulation could be integrated by a robot manipulator simulation to resemble more to a hardware case. Using this integration, manipulator kinematics, dynamics and path planning with obstacle avoidance concepts could also be analyzed and might be involved in grasp synthesis.

Finally, the grasp synthesis and preshape estimation is still an open and challenging subject for future research in dexterous robotics, with different possible approaches.

REFERENCES

ABRAHAM Ajith, " Neuro-Fuzzy Systems : State-of-the-Art Modeling Techniques", LNCS 2084, Springer-Verlag, June 2001, pp 269-276, Germany.

ABRAHAM Ajith, " Hybrid Intelligent Systems Design-A Review of a Decade Research", Technical Report, School of CIT, Monash University, 2000 , Australia.

ADAMS J.D and WHITNEY, D.E.,"Application of Screw Theory to Constraint Analyis of Mechanical Assemblies Joined by Features", Journal of Mechanical Design, Trans. of ASME, vol 123, pp26-32, March 2001.

AKELLA Prasad N., " Contact Mechanism and the Dynamics of Manipulation", PhD Thesis, Dept. of Mech. Engineering, Stanford University, 1993.

ARBIB, M.A., 1981, "Perceptual structures and distributed motor control ", Handbook of Physiology-The nervous system. II. Motor control , In V.B. Brooks Ed., pp. 1449-1480 , American Physiological Society.

BALL, Sir Robert Stawell, 1900, "A Treatise on the Theory of Screws", Cambridge University Press, reprint 2000, Cambridge, U.K.

BORST Ch., FISCHER M. and HIRZINGER G., "A Fast and Robust Grasp Planner for Arbitrary 3D Objects", Proceedings of the 1999 IEEE International Conference on Robotics and Automation, Detroit, Michigan, 1999, pp. 1533--1539.

BORST Ch., FISCHER M. and HIRZINGER G., "Calculating Hand Configurations for Precision and Pinch Grasps", Proceedings of the 1999 IEEE/RSJ IROS 2002 International Conference on Robots and Systems, EPFL, Lausanne ,October 2002, pp. 1553--1559.

BORST Ch., FISCHER M. and HIRZINGER G., "Grasp Planning: How to Choose a Suitable Task Wrench Space",, Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2004, New Orleans, LA,USA, pp 319-325.

BROWN Mark, “Layered Perceptron Networks and the Error Back Propagation Algorithm”, Technical Report, DECS, University of Southampton, UK, 1996.

BICCHI A. and KUMAR V. “Robotic Grasping and Contact: A Review” , In Proc. IEEE ICRA2000, , San Francisco, CA.

CIBLAK Namik, “ Analysis of Cartesian Stiffness and Compliance with Applications”, Ph.D thesis, Dept. of Mech. Engineering, Georgia Institute of Technology, 1998.

CIOCARLIE M. , MILLER A. and ALLEN P., “Grasp Analysis Using Deformable Fingers”, Proceedings of the IEEE/RSJ IROS 2005 International Conference on Robots & Systems, 2005, Canada.

CHRISTEL Marianne I. and FRAGASZY Dorothy ,” Manual Function in Cebus apella, Digital Mobility,Preshaping, and Endurance in Repetitive Grasping “**,** International Journal of Primatology, Vol. 21, No. 4, pp 697-716, 2000.

CRAIG, John J.,1986, “ Introduction to Robotics”, Addison-Wesley Inc. ,USA

CUTKOSKY Mark R.,1985, Robotic Grasping and Fine Manipulation, Kluwer Academic Publishers, Massachusetts.

CUTKOSKY Mark.R. and Paul K. Wright, ``Modeling Manufacturing Grips and Correlation with the Design of Robotic Hands," Proceedings of the IEEE International Conference on Robotics and Automation, 1986, pp. 1533--1539.

CUTKOSKY Mark.R., “ On grasp choice grasp models and the design of hands for manufacturing tasks ” , IEEE Trans. Robotics and Automation, Vol. 5, No. 3., 1989.

DUFFY Joseph, 1996, “Statics and Kinematics with Applications to Robotics”, Cambridge University Press, NY. USA .

FERRARI C. and CANNY J., “Planning optimal grasps”, In Proc. IEEE ICRA, May 1992, Nice France, pp. 2290-2295.

FU K.S., GONZALES R.C. , LEE C.S.G.,1987, “Robotics: Control,Sensing,Vision and Intelligence”, McGraw-Hill, Singapure.

GRUPEN Roderic A. and HENDERSON Thomas C. , ``A Survey of Dexterous Manipulation," Unpublished Manuscript, Department of Computer Science, The University of Utah, 1987.

GRUPEN Roderic A. ,1998, " A Developmental Organization for Robot Behaviour", Technical Report, UMass, Amherst,USA.

HANAFUSA H. and ASADA H., ``Stable Prehension by a Robot Hand with Elastic Fingers," Proceedings of the 7th International Symposium on Industrial Robots, Tokyo, 1977, pp. 361--368, (also appears in Brady et. al.(Ed.), Robot Motion, pp. 323--335).

HAYKIN Simon, 1994, "Neural Networks-A Comprehensive Foundation", Macmiilan College Publishing.

HOWE R. D. and CUTKOSKY M.R, "Practical Force-Motion Models for Sliding Manipulation", International Journal of Robotics Research, 15(6:557-572, December 1996, USA.

HSU P., LI Z., and SASTRY S., 1988, "On Grasping and Coordinated Manipulation by a Multifingered Robot Hand", Proc. of 1988 Int. Conf. on Robotics and Automation, pp. 384-389.

HUNT Kenneth H., 1978, "Kinematic Geometry of Mechanisms", Oxford Unviersity Press, Oxford.

HYDE James M. , " A Phase Management Framework Event-Driven Dextrous Manipulation", PhD Thesis, Dept. of Mech. Engineering, Stanford University, 1995.

IBERALL Thea, ``The Nature of Human Prehension: Three Dextrous Hands in One," Proceedings of the 1987 IEEE International Conference on Robotics and Automation, 1987, pp. 396--401.

JACOBSEN Hans A., " A Generic Architecture for Hybrid Intelligent Systems", Proc. of the IEEE-FUZZ 98 International Conference on Fuzzy Systems, 1998, pp. 100--106., Alaska , USA.

JOUFFE Lionel, " Fuzzy Inference System Learning by Reinforcement Methods" The, IEEE Trans. Systems Man and Cybernetics Part C: Applications, vol 28 No 3, August 1998.

KAELBLING Leslie and LITTMAN Michael, " Reinforcement Learning : A Survey", Journal of Artificial Intelligence Research 4 (1996) :237-285, Morgan-Kaufman Publishers.

KERR J.and ROTH B.. “Analysis of multifingered hands”, International Journal of Robotics Research, 4(4):3-17, Winter 1986.

KIM Doik and CHUNG Wan Kyun, ”Analytic Formulation of Reciprocal Screws and Its Application to Nonredundant Robot Manipulators”, Transactions of ASME, vol 125, pp158-164, March 2003.

KONKAR Ranjit., “ Incremental Kinematic Analysis and Symbolic Synthesis of Mechanisms”, PhD Thesis, Dept. of Mech. Engineering, Stanford University, 1993.

KUBAT Miroslav, BRATKO Ivan and MICHALSKI R., 1996, “Machine Learning and Data Mining: Methods and Applications”, John Wiley & Sons, 1996.

LEUNG Carlos and LOVELL Brian, “ 3D Reconstruction through Segmentation of Multi-View Image Sequences” , Proc. Of APRS Workshop on Digital Image Computing, pp. 87-92, 2003, Australia.

LI Z., SASTRY S.S. , “ Task oriented optimal grasping by multifingered robot hands ”, IEEE Journal of. Robotics and Automation, Vol. 4, No. 1., 1988.

LIN Ming C. And GOTTSCHALK Stefan, “Collision detection between geometric models: a survey”, Proc. of IMA Conf. On Mathematics of Surfaces, 1998, pp 37-56.

LIPKIN & PATTERSON, “Hybrid Twist and Wrench Control for a Robotic Manipulator”, Trans. ASME,Vol 110, June 1988.

LIU H., IBERALL T., BEKEY G.A. , “ The multidimensional quality of task requirements for dextrous robot hand control ” , IEEE -Control Systems Magazine, pp452-457., 1989.

LYONS D.M , “ A simple set of grasps for a dextrous hand ”, Proc. IEEE Int. Conf. Robotics & Automation, 1985, pp.588-594.

MASON Matthew T. and SALISBURY J. Kenneth , 1985, Robot Hands and the Mechanics of Manipulation, MIT Press.

MICHELMAN P., “Precision Object Manipulation with a Multifingered Robot Hand”, IEEE Trans. Robotics and Automation, Vol. 14, No.1, 1998.

MILLER Andrew T. and Allen Peter K. , "Examples of 3D Grasp Quality Computations", In Proc. of the 1999 IEEE International Conference on Robotics and Automation, pages 1240-1246, 1999.

MIRTICH Brian and CANNY John ." Easily computable optimum grasps in 2-D and 3-D" . In Proc. of the 1994 IEEE International Conferenceon Robotics and Automation, pages 739-747, San Diego, CA, May 1994.

MISHRA Budd, SCHWARTZ J.T. and SHARIR M. , ``On the Existence and Synthesis of Multifinger Positive Grips,'' Algorithmica, Special Issue: Robotics, Vol. 2, No. 4, November, November, 1987, pp. 541--558.

MISHRA B., SILVER N., "Some Discussion of static Gripping and Its Stability", IEEE Trans. Systems, Man and Cybernetics,Vol .19,No.4, 1989.

MÖLLER Tomas and TRUMBORE Ben, " Fast, Minimum storage Ray / Triangle Intersection", Journal of Graphics Tools, 2(21)21-28,1997.

MURAKAMI Kouji and HASEGAWA Tsutomu, "Novel Fingertip Equipped with soft Skin and Hard Nail for Dexterous Multifingered Robotic Manipulation", Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2003, Sept. 2003, Taiwan, pp 708-713.

MURAKAMI Kouji and HASEGAWA Tsutomu, "Tactile Sensing of Edge Direction of an Object with a Soft Fingertip Contact", Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2005, April 2005,Barcelona, Spain,, pp 319-325.

MURRAY R., LI Z. and SASTRY S., 1994, "An Mathematical Introduction to RobotManipulation", CRC Press, Inc.

NAMIKI Akio and ISHIKAWA Masatoshi, "Robotic Catching Using a Direct Mapping from Visual Information to Motor Command", Proceedings of the IEEE International Conference on Robotics and Automation, ICRA 2003, Sept. 2003, Taiwan, pp 2400-2405.

NAPIER J. R. (1956), " The prehensile movement of humand hand ", J. Bone Joint Surgery, Vol. 38B, pp.902-913.

NGUYEN Van-Duc, ``Constructing Stable Grasps in 3D," Proceedings of the 1987 IEEE International Conference on Robotics and Automation, 1987, pp. 234--239.

NGUYEN T.N. and STEPHANOU H.E., "A continuous model of robot hand preshaping " , Proc. IEEE Int. Conf. Robotics & Automation. 1987, pp83-89.

OHWOVORIOLE M.S. , An Extension of Screw theory and its Applications to the Automation of Industrial Assemblies, Stanford Artificial Intelligence Laboratory Memo AIM-338, April, 1980.

OKAMURA A. M., N. Smaby, et al. (2000). "An overview of dexterous manipulation." IEEE International Conference on Robotics and Automation. San Francisco, CA. 255-262.

OZTOP E. ,BRADLEY N.S. and ARBIB M. A., " Infant Grasp Learning: A Computational Model", Exp. Brain Res. (2004) 158:480-503, Springer-Verlag, 2004.

________, OpenGL Programming Guide,1990, "The RedBook", Silicon Graphics Inc., USA.

________,OpenGL Utility Toolkit Programming Interface,1996,"GLUT API V3", Silicon Graphics Inc. , USA.

POLLARD N. S., "Parallel Methods for Synthesizing Whole-Hand Grasps from Generalized Prototypes.", PhD thesis, Department of Electrical Engineering and Computer Science, Massachusetts Institute of Technology, 1994.

POLLARD N. S., "Closure and Quality Equivalence for Efficient Synthesis of Grasp from Examples", The International Journal of Robotics Research, Vol 23, No 6, pp595-613, June 2004.

REED Michael K. And ALLEN Peter K., " 3D Modeling from Range Imagery", Image and Vision Computing, vol 17: 99-111, 1999, Elsevier Science, B.V.

ROSS, Timothy J. ,1995, "Fuzzy Logic with Engineering Applications", McGraw-Hill, New York, USA.

ROTH Bernard, "Screws, Motors, and Wrenches that Cannot be Bought in a Hardware Store", In Robotics Research, Chapter 8, pp 679-693, 1984.

RUSSEL S. and NORVIG P., 1995, ”A Modern Approach to Artificial Intelligence”, Prentice-Hall, Englewood Cliffs, NJ ,1995 .

SALISBURY J.K., “ Kinematic and Force Analysis of Articulated Hands”, PhD thesis, Dept. of Mech. Engineering, Stanford University, 1982.

SANTINI Simon and JAIN Ramesh, “Similarity Measures”, The IEEE Trans. on Pattern Analysis and Machine Intelligence , 21(9):871-883, 1999.

SHIMOGA K.B. ,”Robot grasp synthesis algorithms: A survey”, International Journal of Robotics Research, 15(3):230-266, June 1996.

SHIRLEY Peter and MORLEY Keith, 2003, “Realistic Ray Tracing”, 2nd Eds, A.K. Peters Publishing, pp 206-298, USA.

SILVER Naomi and MISHRA Budd, ``Issues in Dextrous Manipulation: Grasping, Compliance, and Control," Robotics Report No. 141, Courant Institute, NYU, New York, February 1988.

SCHWARTZ J.T, ``Problems and Perspectives in Robotics," Ed. J.W. de Bakker and M. Hazewinkel and J.K. Lenstra, Mathematics and Computer Science Proceedings of the CWI Symposium November 1983, Amsterdam, Elsevier Science Publishing Company, 1986, pp. 66--87.

STAMOS Ioannis and ALLEN Peter K., “3D Model Construction Using Range and Image Data”, Proc. of the IEEE Conference on Computer Vision and Pattern Recognition, CVPR2000, pp. 531-536, South Carolina, 2000, USA.

STAMOS Ioannis and ALLEN Peter K., “Geometry and Texture Recovery of Scenes of Large Scale”, Computer Vision and Image Understanding, No 88 ,pp. 94-118, Elsevier Science, 2002, USA.

STENSTROM J. Ross and CONNOLY C. Ian, “Constructing Object Models from Multiple Images”, International Journal of Computer Vision, vol. 7, no 9, pp. 185-212, 1992.

TAYLOR C.L. and SCHWARTZ R.J., ``The Anatomy of Mechanics of the Human Hand," Artificial Limbs, Vol. 2, May, 1955, pp. 22--35.

TSOUKALAS L.H. and UNRIG R.E, 1997, “Fuzzy and Neural Approaches in Engineering”, John Wiley and Sons Inc.,Canada.

TSUNEO Yoshikawa and KIYOSHI Nagai, ``Manipulating and Grasping Forces in Manipulation by Multi-fingered Hands," Proceedings of the 1987 IEEE International Conference on Robotics and Automation, IEEE Computer Society Press, Los Angeles, March, 1987, pp. 1998--2004.

WALDRON, Kenneth J., "The Constraint Analysis of Mechanisms." , The Journal of Mechanisms, Vol. 1, pp 101-114. Great Britain: Pergamon Press, 1966.

WANG P., 1983, "Approaching Degree Method", in Fuzzy Sets Theory and Its Applications, Science and Technology Press, Shangai, P.R.C.

WELSTEAD Stephen ,1994, "Fuzzy Logic and Neural Network Applications in C++ ", John Wiley and Sons Inc., Canada.

WRIGHT Paul Kenneth and DAVID Alan, 1988, Manufacturing Intelligence, Addison-Wesley Publishing Company, Inc., 1988.

YEUNG D.S and TSANG E.C.C, " A Comparative Study on Similarity Based Fuzzy Reasoning Methods", The, IEEE Trans. on Systems Man and Cybernetics Part B: Cybernetics, vol 27 No 2, April 1997.

YOSHIKAWA Ayumi and NISHIMURA Takeshi, " Relationship between Subjective Degree of Similarity and Some Similarity Indices of Fuzzy Sets", Proc. of the IEEE International Conference on Fuzzy Systems, 1999.

ZAEH Michael F., EGERMEIER Hans, PETZOLD Bernd and SPITZWEG Michael, "Dexterous Object Manipulation in a Phsyics Based Virtual Environment", Proc. of the IEEE-MechRob 2004 conf on Mechatronics and Robotics 2004, pp 1340-1344, Aachen Germany.

CURRICULUM VITAE

Personal Details

- Name: Cabbar Veysel BAYSAL
- Marital Status: Married.
- Nationality: Turkish
- Date of Birth and Place: 21-02-1969, CEYHAN ADANA
- e-mail: cvbaysal@yahoo.com

Professional Experience

2003 – 2006 CJSC INTERCORN Dnipropetrovsk ,UKRAINE
Corn Processing Industries for Starch and Corn Syrups.
Project E& I Coordinator

2001 - 2003 TAT NİŞASTA A.Ş. ADANA
Corn Processing Industries for Starch and Corn Syrups.
Energy & Maintenance Manager.

1998 - 2001 Çukurova University. ADANA
Department of Electrical & Electronics Engineering.
Lecturer.

1995 - 1998 Amylum Nişasta. ADANA
Corn Processing Industries for Starch and Corn Syrups.
Electrical & Automation Chief Engineer, Energy Manager.

1993 - 1995 UNILEVER FOOD. ÇORLU/ADANA
Electrical & Insturmentation Manager Trainee / Manager.

1990 - 1993 T.Ş.F.A.Ş-EMAF. ANKARA
Türkiye Şeker Fab. A.Ş. Elektromekanik Aygıtlar Fab
R & D Engineer.

Education

1999 - 2006 Çukurova University **ADANA**
Ph.D in Electrical & Electronics Eng. , Specific Area: Robotics

1993 - 1990 M.E.T.U **ANKARA**
MSc. in Electrical & Electronics Eng.

1986 - 1990 M.E.T.U **ANKARA**
BSc. in Electrical & Electronics Eng.

1985 - 1986 M.E.T.U **ANKARA**
English Prep. School

- Honors degree

Publications

- "A Computer Impementation of Fractal Based Inference Network for Sensorimotor Robot Hand Preshaping" , IEEE IROS92 Int. Conf. Proc.,1 ,202, Raleigh, NC USA,1992.
- "Preshape Induced Grasp Controller for Dexterous Manipulation with a Multifingered Robot Hand", IEEE ICM04 Int. Conf. Proc., , Istanbul, TURKEY,2004.

Memberships

- IEEE, The Institute of Electrical And Electronics Eng. USA, since 1993
- TMMOB-EMO, Turkish Chamber of Electrical Engineers, since 1990
- METU Alumni Society, since 1993

APPENDIX A:

DETAILS FOR DESIGN AND ANALYSIS OF THE NEURAL NETWORK

In this section, we present the details for the design of ANN and parametric analysis to optimize the ANN structure, in figures. In order to give a clear picture of what we have done, we briefly give information about our design and optimization process for the ANN structure, which is given in section 3.10.

First, we explain that how we obtained the Training Data for Neural Network. During our work, in order to reach our goals, while we analyze our workframe for kinematic estimations, we developed a stand-alone software program as an auxillary tool. With this software, with a user selected object, it is possible select an approach plane and possible to give a value for the approach. After the selection , the tool performs the kinematic estimations and outputs the resultant wrench, WU etc., based on our analysis parameters. Hence, we collected results with data for 12-plane and various approaches, to form Training Data for different objects which eventually to be used in Pretraining Phase. The sample sets are presented in the following pages with the figure captions.

Another issue we would like to explain is that how we estimate ANN parameters such as Learning Rate, Number of Hidden Nodes, EMS Thresholds. After forming general structure of the ANN, using the Pretraining data we obtained, we started to perform experiments with different sets of ANN parameters. At the end ,we have collected responses of ANN with different parameter sets. The details of these analysis are also given at the following pages indicated with captions. As a result of the tabulated forms we obtained a surface plot to find an optimum set for the ANN parameters. These plots are also given in the following pages with the figure captions.

Finally, based on these details, we constructed ANN module of our controller designed specifically.

NEURAL NETWORK TRAINING DATA SET-1 FOR REC. BOX OBJECT

	NEURAL NETWORK TRAINING INPUT SET															NEURAL NETWORK TRAINING OUTPUT SET								
	TASK ATTRIBUTES								OBJECT ATTRIBUTES							PLANE IDX				APR IDX			#VF	
NN_IDX	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
NO	Tx	Ty	Tz	Rx	Ry	Rz	Sta	Manu	Width	Height	Depth	# Vert.	# Vn	# Faces	# Triang.									
1	0,450	-1,000	0.150	0.150	-0,290	-0.006	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	0,00	1,00	1,00	0.00	0,00	1,00	0,00
2	0,450	-1,000	0.150	0.150	-0,310	-0.110	1,000	1,000	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	0,00	1,00	1,00	0.00	0,00	1,00	0,00
3	-1.000	0,450	-0,160	-0,260	0,042	-0.260	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	1,00	0,00	0,00	1.00	0,00	1,00	0,00
4	-1.000	0,450	-0,160	-0,330	0,042	0,220	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	1,00	0,00	1,00	0.00	1,00	1,00	0,00
5	0,000	0,300	0.000	0.000	0,051	0,000	1,000	0,167	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	1,00	1,00	0,00	1.00	1,00	0,00	1,00
6	0,000	0,300	0.000	0.000	0,051	0,000	1,000	0,167	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	0,00	1,00	1,00	1,00	0.00	1,00	0,00	1,00
7	0,000	0,000	0.300	0.000	0,000	0,100	1,000	0,333	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	0,00	0,00	1,00	0.00	0,00	0,00	1,00
8	0,000	0,000	0.300	0.000	0,000	0,100	1,000	0,333	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	0,00	0,00	1,00	0.00	0,00	0,00	1,00
9	0,450	1,000	-0,150	0.150	0,290	0,015	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	0,00	1,00	1,00	0.00	0,00	1,00	0,00
10	0,450	1,000	-0,150	0.150	0,310	0,097	1,000	1,000	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	0,00	1,00	1,00	0.00	0,00	1,00	0,00
11	1,000	0,450	0.160	0.270	0,040	0,170	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	1,00	0,00	0,00	1.00	1,00	1,00	0,00
12	1,000	0,450	0.160	0.330	0,040	-0.200	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	1,00	0,00	1,00	0.00	1,00	1,00	0,00
13	0,000	0,000	0.300	0.000	0,000	0,100	1,000	0,333	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	1,00	1,00	0,00	1.00	1,00	0,00	1,00
14	0,000	0,000	0.300	0.000	0,000	0,100	1,000	0,333	0,175	0,683	0.339	0,667	0,500	1,000	0,012	0.00	1,00	1,00	1,00	1,00	0.00	1,00	0,00	1,00
15	0,000	0,300	0.000	0.000	0,051	0,000	1,000	0,167	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	0,00	0,00	0,00	1.00	0,00	0,00	1,00
16	0,000	0,300	0.000	0.000	0,051	0,000	1,000	0,167	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	0,00	0,00	1,00	0.00	1,00	0,00	1,00
17	0,600	0,300	2.000	0.100	0,021	0,590	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	0,00	1,00	1,00	0.00	0,00	1,00	1,00
18	0,600	0,300	2.000	0.100	-0,200	0,620	1,000	1,000	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	0,00	1,00	1,00	0.00	0,00	1,00	1,00
19	2,000	-0,300	0.620	0.570	-0,170	0,057	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	1,00	0,00	0,00	1.00	1,00	1,00	1,00
20	2,000	-0,300	0.620	0.650	0,350	0,057	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	1,00	0,00	1,00	0.00	1,00	1,00	1,00
21	0,600	-0,300	-2,000	0.100	-0,069	-0.580	1,000	1,000	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	1,00	1,00	1,00	0.00	0,00	1,00	1,00
22	0,600	-0,300	-2,000	0.100	0,150	-0.620	1,000	1,000	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	0,00	1,00	1,00	1,00	0.00	0,00	1,00	1,00
23	-2.000	0,300	0.580	-0,560	0,260	0,047	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	1,00	0,00	0,00	0,00	1.00	1,00	1,00	1,00
24	-2.000	0,300	0.580	-0,650	-0,350	0,046	1,000	0,833	0,175	0,683	0.339	0,667	0,500	1,000	0,012	1.00	1,00	0,00	0,00	1,00	0.00	1,00	1,00	1,00

NEURAL NETWORK TRAINING DATA SET-2 FOR REC. BOX OBJECT

NEURAL NETWORK TRAINING INPUT SET

	TASK ATTRIBUTES								OBJECT ATTRIBUTES						
NN_IDX	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14
NO	Tx	Ty	Tz	Rx	Ry	Rz	Sta	Manu	Width	Height	Depth	# Vert.	# Vn	# Faces	# Triang.
1	-1.000	0,300	0.310	-0,790	-0.130	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
2	-1.000	0,300	0.310	-0,820	-0.230	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
3	1,500	-0,310	-0,720	0,081	-0.230	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
4	1,500	-0,310	-0,870	0,081	0,250	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
5	1,000	0,000	0.000	0,100	0,000	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
6	1,000	0,000	0.000	0,100	0,000	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
7	0,000	1,000	0.000	0,000	0,200	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
8	0,000	1,000	0.000	0,000	0,200	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
9	1,000	-0,300	0.310	0,790	0,130	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
10	1,000	-0,300	0.310	0,810	0,220	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
11	1,500	0,310	0.740	0,079	0,150	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
12	1,500	0,310	0.860	0,079	-0.230	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
13	0,000	1,000	0.000	0,000	0,200	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
14	0,000	1,000	0.000	0,000	0,200	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
15	1,000	0,000	0.000	0,100	0,000	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
16	1,000	0,000	0.000	0,100	0,000	1,000	0.333	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
17	0,600	2,000	0.200	-0,099	1,600	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
18	0,600	2,000	0.200	-0,310	1,600	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
19	-0.600	2,000	1.500	-0,110	0,120	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
20	-0.600	2,000	1.700	0,410	0,120	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
21	-0.600	-2,000	0.200	0,050	-1.600	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
22	-0.600	-2,000	0.200	0,270	-1.600	1,000	1.000	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
23	0,600	2,000	-1,500	0,200	0,090	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012
24	0,600	2,000	-1,700	-0,410	0,088	1,000	0.833	0,175	0.683	0,339	0,667	0,500	1.000	0,012	0,012

NEURAL NETWORK TRAINING OUTPUT SET

NN_IDX NO	PLANE IDX				APR IDX			#VF	
	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
1	0,00	0.00	0,00	1,00	1.00	0,00	0,00	1,00	0.00
2	0,00	0.00	0,00	1,00	1.00	0,00	0,00	1,00	0.00
3	0,00	0.00	1,00	0,00	0.00	1,00	0,00	1,00	0.00
4	0,00	0.00	1,00	0,00	1.00	0,00	1,00	1,00	0.00
5	0,00	0.00	1,00	1,00	0.00	1,00	1,00	0,00	1.00
6	0,00	0.00	1,00	1,00	1.00	0,00	1,00	0,00	1.00
7	0,00	1.00	0,00	0,00	1.00	0,00	0,00	0,00	1.00
8	0,00	1.00	0,00	0,00	1.00	0,00	0,00	0,00	1.00
9	0,00	1.00	0,00	1,00	1.00	0,00	0,00	1,00	0.00
10	0,00	1.00	0,00	1,00	1.00	0,00	0,00	1,00	0.00
11	0,00	1.00	1,00	0,00	0.00	1,00	1,00	1,00	0.00
12	0,00	1.00	1,00	0,00	1.00	0,00	1,00	1,00	0.00
13	0,00	1.00	1,00	1,00	0.00	1,00	1,00	0,00	1.00
14	0,00	1.00	1,00	1,00	1.00	0,00	1,00	0,00	1.00
15	1,00	0.00	0,00	0,00	0.00	1,00	0,00	0,00	1.00
16	1,00	0.00	0,00	0,00	1.00	0,00	1,00	0,00	1.00
17	1,00	0.00	0,00	1,00	1.00	0,00	0,00	1,00	1.00
18	1,00	0.00	0,00	1,00	1.00	0,00	0,00	1,00	1.00
19	1,00	0.00	1,00	0,00	0.00	1,00	1,00	1,00	1.00
20	1,00	0.00	1,00	0,00	1.00	0,00	1,00	1,00	1.00
21	1,00	0.00	1,00	1,00	1.00	0,00	0,00	1,00	1.00
22	1,00	0.00	1,00	1,00	1.00	0,00	0,00	1,00	1.00
23	1,00	1.00	0,00	0,00	0.00	1,00	1,00	1,00	1.00
24	1,00	1.00	0,00	0,00	1.00	0,00	1,00	1,00	1.00

NEURAL NETWORK TRAINING DATA SET FOR CUBE OBJECT

NEURAL NETWORK TRAINING INPUT SET

NN_IDX	TASK ATTRIBUTES X0	X1	X2	X3	X4	X5	X6	X7	OBJECT ATTRIBUTES X8	X9	X10	X11	X12	X13	X14
NO	Tx	Ty	Tz	Rx	Ry	Rz	Sta	Manu	Width	Height	Depth	# Vert.	# Vn	# Faces	# Triang.
1	0,900	-1,000	0,100	0,050	-0,790	0.091	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
2	0,900	-1,000	0,100	0,050	-0,810	-0,200	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
3	-1,000	0.900	-0,100	-0.790	0,051	-0,005	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
4	-1,000	0.900	-0,100	-0.810	0,051	0.110	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
5	0,300	0.900	-1,000	0,090	0,049	-0,790	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
6	0,100	0.900	-1,000	-0.190	0,049	-0,810	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
7	-0,100	-1,000	0,900	-0.050	-0,790	0.051	1,000	0,667	0.333	0,333	0,333	0,667	0,500	1,000	0,012
8	-0,100	-1,000	0,900	0,190	-0,810	0.051	1,000	0,333	0.333	0,333	0,333	0,667	0,500	1,000	0,012
9	0,900	1.000	-0,100	0,051	0,790	-0,005	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
10	0,900	1.000	-0,100	0,051	0,810	0.190	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
11	1,000	0.900	0,100	0,790	0,049	0.050	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
12	1,000	0.900	0,100	0,810	0,049	-0,190	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
13	0,100	1.000	0,900	0,050	0,790	0.049	1,000	0,500	0.333	0,333	0,333	0,667	0,500	1,000	0,012
14	0,100	1.000	0,900	-0.190	0,810	0.049	1,000	0,333	0.333	0,333	0,333	0,667	0,500	1,000	0,012
15	-0,100	0.900	1,000	-0.050	0,051	0.790	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
16	-0,100	0.900	1,000	0,150	0,051	0.810	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
17	0,900	0.100	1,000	0,049	0,091	0.790	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
18	0,900	0.100	1,000	0,049	-0,150	0.810	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
19	1,000	-0,100	0,900	0,790	-0,095	0.051	1,000	0,500	0.333	0,333	0,333	0,667	0,500	1,000	0,012
20	1,000	-0,100	0,900	0,810	0,150	0.051	1,000	0,333	0.333	0,333	0,333	0,667	0,500	1,000	0,012
21	0,900	-0,100	-1,000	0,051	-0,091	-0,790	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
22	0,900	-0,100	-1,000	0,051	0,150	-0,810	1,000	0,833	0.333	0,333	0,333	0,667	0,500	1,000	0,012
23	-1,000	0.100	0,900	-0.790	0,051	0.049	1,000	0,500	0.333	0,333	0,333	0,667	0,500	1,000	0,012
24	-1,000	0.100	0,900	-0.810	-0,150	0.049	1,000	0,333	0.333	0,333	0,333	0,667	0,500	1,000	0,012

NEURAL NETWORK TRAINING OUTPUT SET

NN_IDX NO	PLANE IDX Y0	Y1	Y2	Y3	APR IDX Y4	Y5	Y6	#VF Y7	Y8
1	0,00	0,00	0,00	1,00	0,00	1,00	1,00	1,00	0.00
2	0,00	0,00	0,00	1,00	1,00	0,00	1,00	1,00	0.00
3	0,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	0.00
4	0,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	0.00
5	0,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00	0.00
6	0,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	0.00
7	0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	0.00
8	0,00	1,00	0,00	0,00	1,00	0,00	1,00	1,00	0.00
9	0,00	1,00	0,00	1,00	1,00	0,00	0,00	1,00	0.00
10	0,00	1,00	0,00	1,00	1,00	0,00	0,00	1,00	0.00
11	0,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	0.00
12	0,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	0.00
13	0,00	1,00	1,00	1,00	0,00	1,00	1,00	1,00	0.00
14	0,00	1,00	1,00	1,00	1,00	0,00	1,00	1,00	0.00
15	1,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	0.00
16	1,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00	0.00
17	1,00	0,00	0,00	1,00	1,00	0,00	0,00	1,00	0.00
18	1,00	0,00	0,00	1,00	1,00	0,00	0,00	1,00	0.00
19	1,00	0,00	1,00	0,00	0,00	1,00	1,00	1,00	0.00
20	1,00	0,00	1,00	0,00	1,00	0,00	1,00	1,00	0.00
21	1,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00	0.00
22	1,00	0,00	1,00	1,00	1,00	0,00	1,00	1,00	0.00
23	1,00	1,00	0,00	0,00	1,00	0,00	0,00	1,00	0.00
24	1,00	1,00	0,00	0,00	1,00	0,00	0,00	1,00	0.00

NEURAL NETWORK TRAINING DATA SET FOR SPHERE OBJECT

	NEURAL NETWORK TRAINING INPUT SET															NEURAL NETWORK TRAINING OUTPUT SET								
	TASK ATTRIBUTES								OBJECT ATTRIBUTES							PLANE IDX				APR IDX			#VF	
NN_IDX	X0	X1	X2	X3	X4	X5	X6	X7	X8	X9	X10	X11	X12	X13	X14	Y0	Y1	Y2	Y3	Y4	Y5	Y6	Y7	Y8
NO	Tx	Ty	Tz	Rx	Ry	Rz	Sta	Manu	Width	Height	Depth	# Vert.	# Vn	# Faces	# Triang.									
1	2,900	-0,150	-0,045	0,860	-0,053	-0,058	1,000	0,500	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	0,00	1,00	0,00	1,00	0,00	1,00	1,00
2	-1,800	-1,200	0,180	-0,500	-0,340	-0,022	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	0,00	1,00	1,00	0,00	0,00	1,00	1,00
3	-0,960	1,700	-0,530	-0,250	0,510	-0,096	1,000	1,000	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	1,00	0,00	0,00	1,00	0,00	1,00	1,00
4	-1,100	-0,530	-0,560	-0,390	-0,150	-0,014	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00
5	-0,190	1,700	-1,100	-0,110	0,510	-0,340	1,000	1,000	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	1,00	1,00	0,00	1,00	0,00	1,00	1,00
6	-0,220	-0,530	-1,200	-0,180	-0,150	-0,300	1,000	1,000	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	0,00	1,00	1,00	0,00	1,00	1,00	1,00	1,00
7	-0,160	-1,000	1,200	0,031	-0,310	0,380	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00
8	-0,360	-1,100	-1,200	-0,039	-0,360	-0,350	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00
9	2,300	0,950	-0,330	0,720	0,280	0,065	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	0,00	1,00	0,00	1,00	0,00	1,00	1,00
10	-1,200	1,100	-0,360	-0,350	0,360	-0,041	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	0,00	1,00	1,00	0,00	0,00	1,00	1,00
11	0,910	2,600	-0,130	0,300	0,820	-0,180	1,000	1,000	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	1,00	0,00	0,00	1,00	0,00	1,00	1,00
12	1,200	-0,530	-0,220	0,290	-0,150	-0,210	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00
13	0,028	1,000	2,300	-0,120	0,320	0,710	1,000	0,666	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	1,00	1,00	0,00	1,00	0,00	1,00	1,00
14	0,005	1,200	-1,200	-0,096	0,320	-0,350	1,000	0,666	0,660	0,667	0,660	0,506	0,506	1,000	0,320	0,00	1,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00
15	-0,530	1,700	0,960	-0,016	0,520	0,280	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00
16	-0,560	-0,530	1,100	-0,110	-0,140	0,350	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	0,00	0,00	0,00	1,00	1,00	1,00	1,00
17	2,400	0,180	1,000	0,730	-0,023	0,300	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	0,00	1,00	0,00	1,00	0,00	1,00	1,00
18	-1,300	0,180	1,200	-0,370	-0,074	0,350	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	0,00	1,00	1,00	0,00	0,00	1,00	1,00
19	1,000	-0,180	2,400	0,310	0,026	0,730	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	1,00	0,00	0,00	1,00	0,00	1,00	1,00
20	1,200	-0,180	-0,550	0,350	0,053	-0,150	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	1,00	0,00	1,00	0,00	0,00	1,00	1,00
21	2,400	-0,180	-1,000	0,730	0,026	-0,310	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	1,00	1,00	0,00	1,00	0,00	1,00	1,00
22	-1,300	-0,180	-1,200	-0,370	0,078	-0,350	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	0,00	1,00	1,00	1,00	0,00	0,00	1,00	1,00
23	-1,000	0,180	2,400	-0,310	-0,033	0,730	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	1,00	0,00	0,00	0,00	1,00	0,00	1,00	1,00
24	-1,200	0,180	-1,300	-0,350	-0,074	-0,370	1,000	0,833	0,660	0,667	0,660	0,506	0,506	1,000	0,320	1,00	1,00	0,00	0,00	1,00	0,00	0,00	1,00	1,00

NEURAL NETWORK PARAMETER ANALYSIS-1

Mu_O=0.3, Mu_H=0.4, Learning Rate				Mu_O=0.5, Mu_H=0.6, Learning Rate				Mu_O=0.7, Mu_H=0.8, Learning Rate			
EMS Threshold =1e-03		EMS Threshold =5e-03		EMS Threshold =1e-03		EMS Threshold =5e-03		EMS Threshold =1e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch	Hidden Nodes	Epoch	Hidden Nodes	Epoch	Hidden Nodes	Epoch	Hidden Nodes	Epoch
9	18228	9	8876	9	14343	9	3479	9	10836	9	2919
16	13426	16	4390	16	6524	16	2954	16	5740	16	1848
20	14392	20	4347	20	7294	20	2653	20	5327	20	2079
25	9212	25	4221	25	6216	25	2653	25	5019	25	1855
30	10682	30	3794	30	6692	30	2338	30	4704	30	1869
32	11186	32	3941	32	6321	32	2821	32	4382	32	2037
36	9002	36	3703	36	6041	36	$\boxed{2170}$	36	4347	36	1897
40	10318	40	3331	40	5971	40	2373	40	4487	40	$\boxed{1554}$
45	9204	45	3535	45	$\boxed{5474}$	45	2303	45	4704	45	1589
50	9576	50	3486	50	5658	50	2191	50	5145	50	1708
54	9674	54	$\boxed{3129}$	54	6132	54	2436	54	3927	54	1715
60	$\boxed{8718}$	60	3689	60	5579	60	2359	60	4403	60	1687
70	8893	70	3283	70	5220	70	2415	70	3885	70	1603
80	9254	80	3507	80	5319	80	2221	80	3948	80	1547
100	9548	100	3982	100	6209	100	2251	100	3647	100	1498

NOTE : THIS ANALYSIS WAS PERFORMED WITH RECT. BOX OBJECT TRAINING DATA WHICH WAS OBTAINED USING CONTACT COEFF. Ka=0.15 Ko=0.15 Mn=0.5

NN PARAMETER ANALYSIS-2

Mu_O=0.3, Mu_H=0.4, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	31632	36	6888
45	22872	45	10632
60	18576	60	13752

Mu_O=0.5, Mu_H=0.6, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	17616	36	4032
45	11664	45	5424
60	14496	60	5952

Mu_O=0.7, Mu_H=0.8, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	13128	36	11088
45	5664	45	4416
60	3624	60	3600

NOTE : ANALYSIS WITH CUBE OBJECT TRAINING DATA

OBTAINED USING CONTACT COEFF. Ka=0.5 Ko=0.3 Mn=0.8

NN PARAMETER ANALYSIS-3

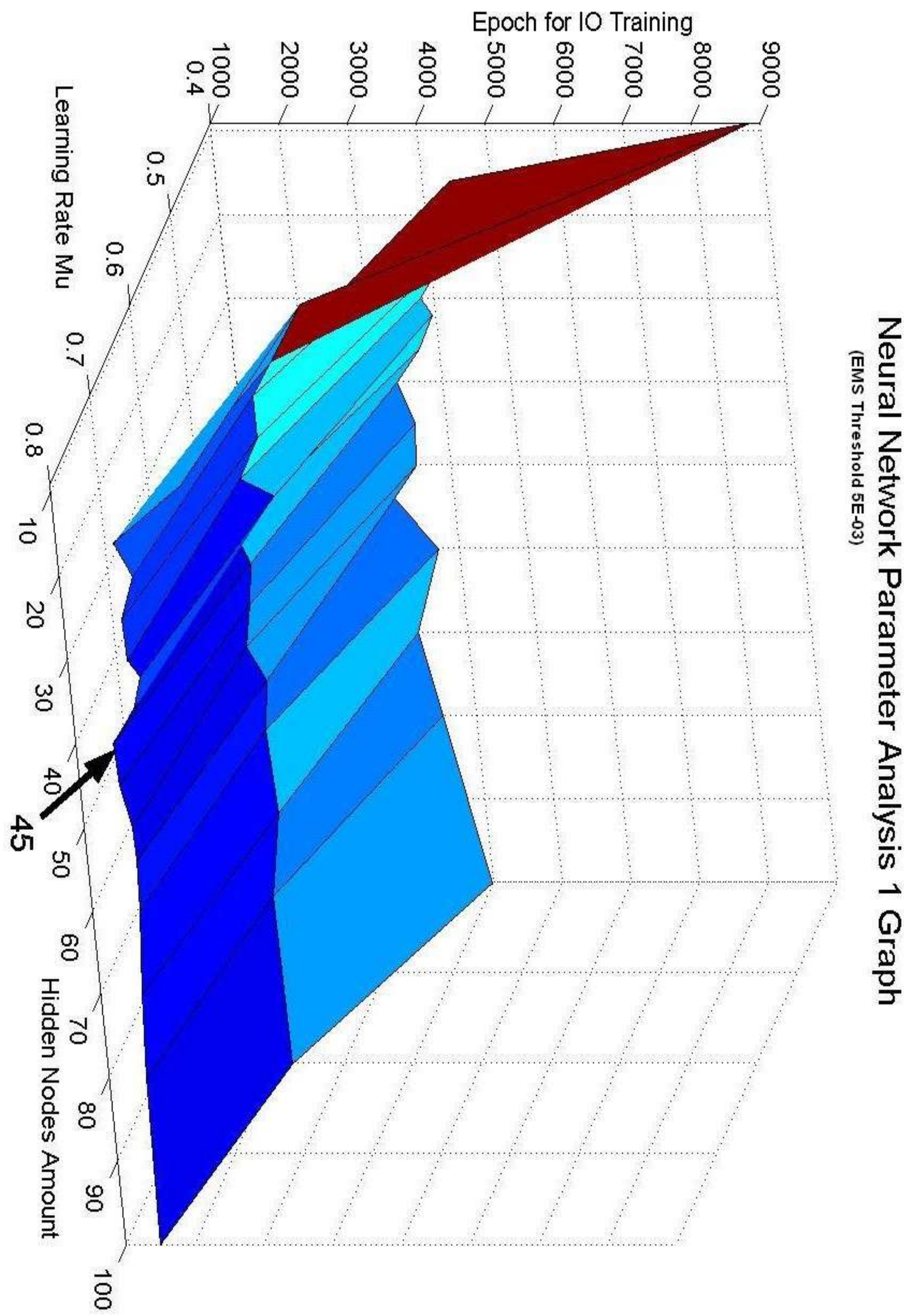
Mu_O=0.3, Mu_H=0.4, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	72096	36	32304
45	53376	45	23208
60	53664	60	29904

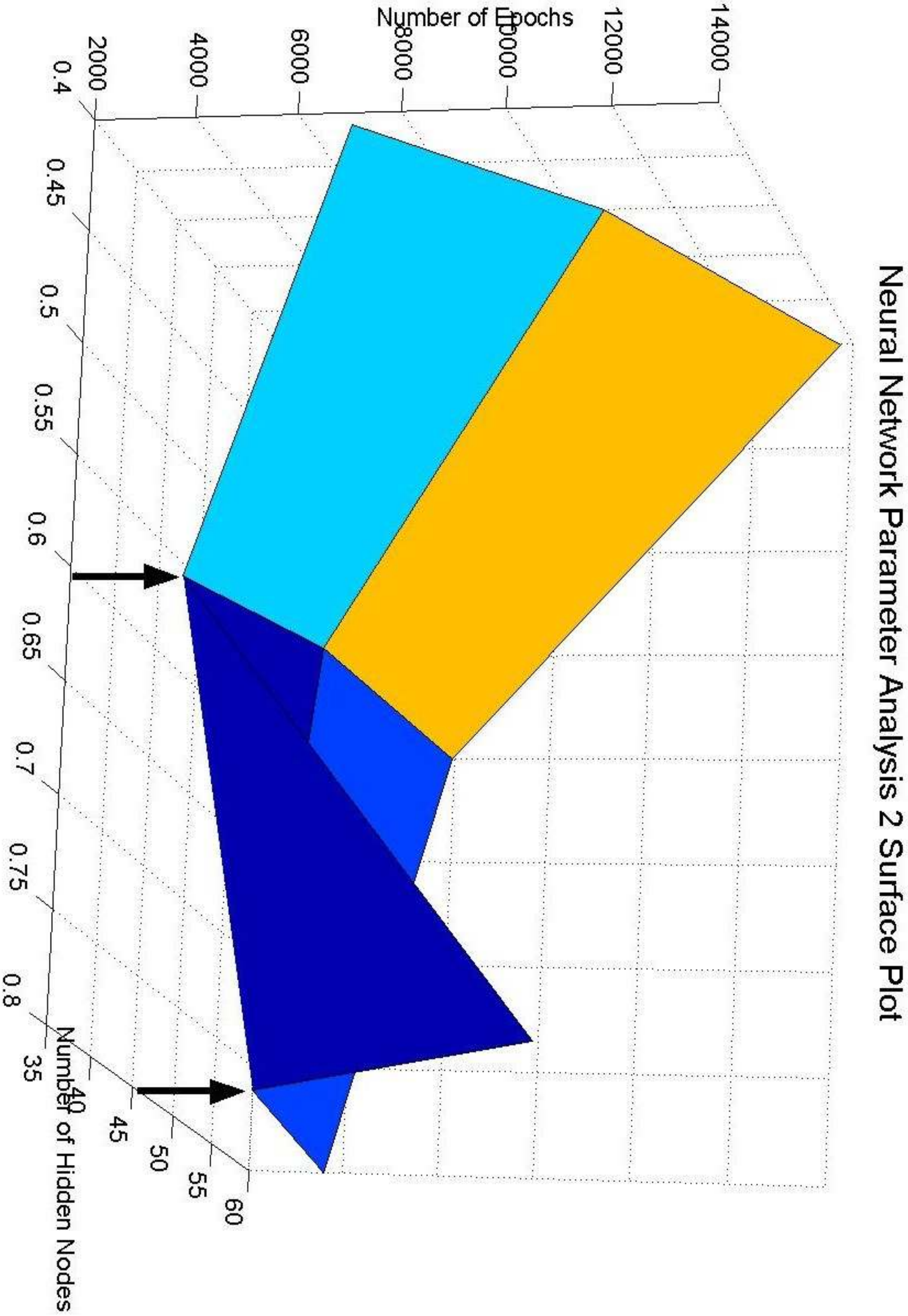
Mu_O=0.5, Mu_H=0.6, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	62212	36	33744
45	55128	45	16632
60	26520	60	17328

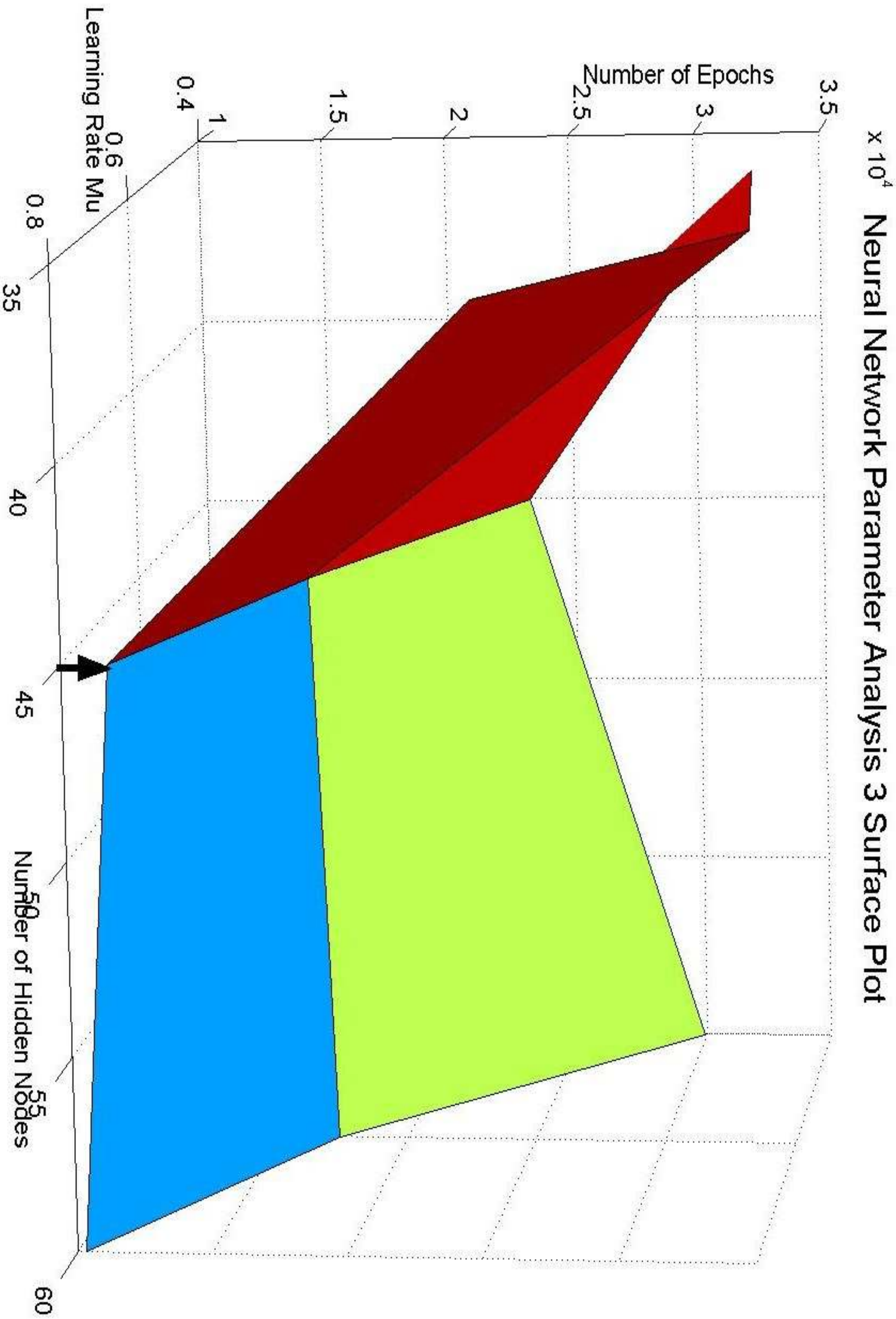
Mu_O=0.7, Mu_H=0.8, Learning Rate			
EMS Threshold =2.5e-03		EMS Threshold =5e-03	
Hidden Nodes	Epoch	Hidden Nodes	Epoch
36	28344	36	25320
45	29976	45	11711
60	35276	60	10296

NOTE : ANALYSIS WITH RECT. BOX OBJECT TRAINING DAT.

OBTAINED USING CONTACT COEFF. Ka=0.5 Ko=0.3 Mn=0.8


Neural Network Parameter Analysis 1 Graph
(EMS Threshold 5E-03)
Epoch for IO Training
1000
2000
3000
4000
5000
6000
7000
8000
9000
Learning Rate Mu
0.4
0.5
0.6
0.7
0.8
Hidden Nodes Amount
10
20
30
40
50
60
70
80
90
100
45


Neural Network Parameter Analysis 2 Surface Plot
Number of Epochs
2000
4000
6000
8000
10000
12000
14000
0.4
0.45
0.5
0.55
0.6
0.65
0.7
0.75
0.8
35
40
45
50
55
60
Number of Hidden Nodes


Neural Network Parameter Analysis 3 Surface Plot
x 10^4
Number of Epochs
1
1.5
2
2.5
3
3.5
Learning Rate Mu
0.4
0.6
0.8
Number of Hidden Nodes
35
40
45
50
55
60