

An Intelligent Inference System for Robot Hand Optimal Grasp Preshaping

Cabbar Veysel Baysal*

*Department of Electrical & Electronics Engineering, Cukurova University
Balcali, Adana, 01330, Turkey*

Aydan M. Erkmen

*Department of Electrical & Electronics Engineering, Middle East Technical University
Ankara, 06530, Turkey
E-mail: aydan@metu.edu.tr
www.metu.edu.tr*

Received: 23-10-2009

Accepted: 02-06-2010

Abstract

This paper presents a novel Intelligent Inference System (IIS) for the determination of an optimum preshape for multifingered robot hand grasping, given object under a manipulation task. The IIS is formed as hybrid agent architecture, by the synthesis of object properties, manipulation task characteristics, grasp space partitioning, low-level kinematical analysis, evaluation of contact wrench patterns via fuzzy approximate reasoning and ANN structure for incremental learning. The IIS is implemented in software with a robot hand simulation.

Keywords: Robot Hand; Dexterous Manipulation; Grasp Preshaping; Preshape Estimation; Intelligent Inference System.

1. Introduction

Robot grasping is one of the major research areas of robotics since it is required for many applications, from handling hazardous material to surgical operations. A review of robot grasping is given by Bicchi and Kumar.¹ One of the main problems in robot grasping is the dexterous manipulation of a multifingered robot hand. An overview of dexterous manipulation is given by Okamura and Cutkosky.²

Dexterous Manipulation (DM) can be decomposed into individual phases such as preshaping (grasp planning and approach), grasping and manipulation.^{1, 2} However, a human like dexterous manipulation requires phases of grasp planning, approach and grasping to be analyzed and handled all together. In the literature related to dexterous manipulation of a robot hand, one of the major problems that remains to be solved is the determination of an appropriate preshape for grasping, given an object coupled to a predefined manipulation task. In order to have a successful grasp execution, the robot control system should initiate grasp as the continuum of a successful grasp planning and approach

of the preshaping phase. Grasp planning and grasp preparation works in literature are generally grouped into two major approaches: grasp synthesis (mechanical or analytical grasp based) and heuristic grasp planning approaches (physiological or knowledge based).

The grasp synthesis approach is mathematically based and seeks to develop computational algorithms for the low-level control of the hand. Low-level control includes computation of finger contact positions, controlling finger joints through finger force and torque. These approaches assume well-defined models and precise sensory information. Shimoga³ provides a survey and overview of grasp synthesis techniques including the algorithms used. Many of the grasp synthesis methods deal with finding optimal contact points to satisfy a force closure property of the possible grasp using linear and nonlinear optimization techniques, such as Li and Sastry's work.⁴ Coelho and Grupen⁵ proposed a robust control for an online grasp synthesis to achieve statically stable grasp configurations given instantaneous contact information. Arimoto⁶ implemented an intelligent control scheme for grasping and manipulation of an object by multi-

* Department of Electrical and Electronics Engineering, Cukurova University Balcali 01330 Adana TURKEY
cvbaysal@yahoo.com ; cvbaysal@cu.edu.tr;

fingered robot hands with fingers inducing rolling contacts with the object in 2D. In addition, grasp synthesis methods based on the concepts of wrench spaces were analyzed in the literature in terms of the set of all wrenches that can be applied to the object through grasp contact, such as Grasp Wrench Space (GWS) proposed by Ferrari and Canny⁷, Object Wrench Space (OWS) introduced by Pollard⁸ and Task Wrench Space (TWS) developed by Borst and Fischer⁹. These methods utilize the convex hull of task wrenches in order to match task and object in terms of wrench cones and ellipsoids, which yielded a scalar value for grasp matching between task and object and some grasp quality measures. However, these methods do not give a direct measure for an arbitrary grasp and do not link the high-level task to low-level grasp kinematics. Synthesis methods are in general, computationally intensive and do not yield effective results for mapping of high-level task, object and low-level grasp kinematics.

The other group of works include heuristic grasp planning approaches, which are knowledge based, and utilize hand preshaping of human grasps. In general, a combination of the object geometry and task properties is the main factor for determining the grasp. In early phase of robotic grasps, many researchers have studied human grasp preshapes for various types of tasks. Cutkosky¹⁰ illustrates a grasp taxonomy for hand preshape matching for manufacturing tasks. Iberall¹¹ gives a task oriented approach to hand preshaping and VF concept. In a work by Kang and Ikeuchi¹², the robotic system observes a human while performing the task and then classifies the human grasp and transfers to the manipulator. Further heuristic methods are developed using fuzzy expert systems to represent human experience for grasp preshaping, such as in the work by Bowers and Lumia¹³, a heuristic fuzzy expert system to grasp non-modelled objects using vision based noisy data. In addition, Gorce and Rezzoug¹⁴ propose a heuristic approach for hand posture configuration using neural networks for inverse kinematics and reinforcement learning in order to optimize hand position and orientation without direct combination of task and object. Miller et al.¹⁵ introduced a method for grasp planning using shape primitives, by demonstrating grasp simulations. Berenson et al.¹⁶ performed a work on grasp planning in complex scenes using grasp analysis, manipulation planning techniques and grasp databases. In a recent

work, Ciocarlie et.al¹⁷ analyzed low dimensional grasp postures for grasp synthesis. Among these methods, heuristic grasp planning methods are efficient for grasp posture formation utilizing hand preshapes however they are lack of low-level kinematics for execution of the grasp.

Neither of two approaches alone constitutes a global solution method for grasp preparation and execution for dexterous manipulation. Hence, another approach is necessary to utilize from heuristic planning and grasp synthesis approaches, to form an intelligent inference architecture. Such an approach can combine low-level kinematics from grasp synthesis approach and preshape formation from heuristic planning approach to complement each other.

In this paper, we use a novel approach to deduce a successful preshape for an optimal initialization of the grasping task, considering the problem of preshaping and grasp kinematics as a sole continuum. It involves grasp planning and grasp synthesis using the attributes coming from each grasp phases. It is foreseen that preshape estimation must be planned virtually in a decision control unit, before any actual movement. This perspective is somewhat similar to physiological grasp situations in human infants and primates, observed by Arbib and Oztop.¹⁸ Further details of our approach are illustrated in the methodology section.

The remainder of the paper is organized as follows. Section 2 presents our methodology. Section 3 illustrates the structure our Intelligent Inference System model and the computer implementation for the model developed. Section 4 contains the implementation results. Finally, Section 5 gives the conclusion of the paper.

2. Methodology

In our preshape estimation methodology, given in figure 1, first, the grasping task is analyzed and decomposed into primitives of axial translational and axial rotational displacements, as done in some of the previous works, such as Michelman's experiments.¹⁹ Object pose and shape are also analyzed to select approaching and grasp characteristics, by using object 3D model. In addition, hand grasp space is partitioned by the Approach Plane method to combine task primitives and hand posture. Task primitives are interpreted as spatial displacements represented by

wrenches in a partitioned grasp space. Moreover, using screw systems properties in existing robot hand contact

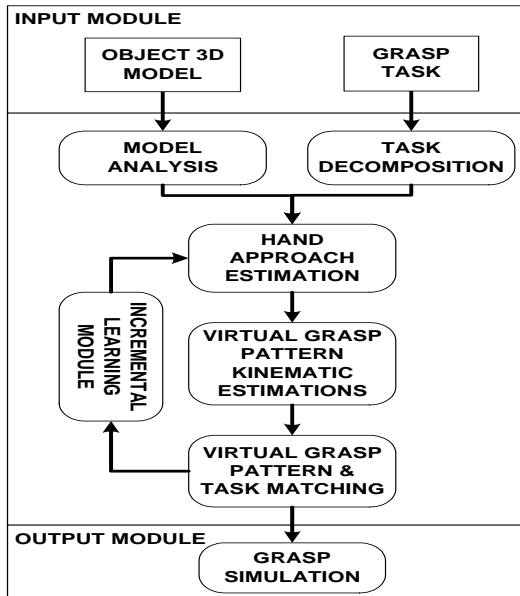


Figure 1. Preshape Estimation Methodology

models, representation pattern of the virtual grasps in wrench form is obtained by estimating kinematical effects of the possible virtual grasps on object 3D model. Utilizing task primitives and virtual grasp patterns in wrench form; a method for high-level task mapping to low-level kinematical actions is developed. We also introduce screw system based grasp measures, which are stability index and manipulability index. These measures support our inference system in justification and inferencing stages.

The inference system uses the all estimated attributes as inputs, some of which are in forms of Fuzzy Sets. Using approximate reasoning based on fuzzy logic, we evaluate the matching of the desired task and possible grasp patterns, all represented in wrench form. Learning is achieved incrementally by means of developed ANN structure. All above techniques are integrated into an Intelligent Inference System. In the following sub-sections, we present detailed explanations for individual parts of our method.

2.1. Object modelling and analysis

In robot hand grasping, grasp synthesis necessitates object representation or a model of object, due to nature of the manipulation problem. In grasp simulation environments, 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.

We use 3D object models in forms of vertex and wireframe solid models of objects to be manipulated, shown in figure 2. In the implementation, the maximum object size is 4U, where U is size of our simulator hand middle finger phalange size.

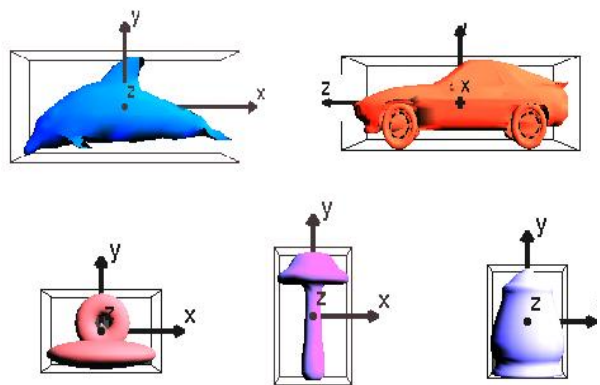


Figure 2. Example Object Models for Workspace

The object models are created using 3D simulation programs, such as 3DCanvas, and stored in disk files as wavefront object format. Using the disk files, possible finger contact points, contact surface normals are obtained by performing object model data analysis. The analysis also produces object attribute sets 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 global coordinate frame is centered on the estimated geometrical center of the object, using set of vertex coordinates of the object model, given in eq. (1) & (2). In this work, it is referenced as **COG** and its coordinate axes are collinear with width, height and depth axes of the object model.

$$V = \bigcup_{i=1}^n v(x_i, y_i, z_i) \quad ; \text{COG} = (m_x, m_y, m_z) \quad (1)$$

$$\begin{pmatrix} m_x \\ m_y \\ m_z \end{pmatrix} = \frac{1}{2} * \begin{pmatrix} (\min_{i=1}^n (x_i) + \max_{j=1}^n (x_j)) \\ (\min_{i=1}^n (y_i) + \max_{j=1}^n (y_j)) \\ (\min_{i=1}^n (z_i) + \max_{j=1}^n (z_j)) \end{pmatrix} \quad (2)$$

2.2. Task representation and grasp space partitioning

In our perspective, for a task to be performed on a graspable object, there must be a link between a given high-level task description and a given low-level object description. Furthermore, the dextrous manipulation task needs to be parameterized by spatial displacements of object after being securely grasped. In our preshape estimation, the task is analyzed and decomposed into primitives of axial translational and axial rotational displacements. Thus, high-level task information is translated into low-level grasp kinematics. Each primitive manipulation function performs a single translation or rotation. To perform arbitrary translations and rotations, the primitive functions must be sequenced. According to Chasle’s theorem, every spatial displacement is the composition of rotation about an axis and translation along the same axis. Spatial displacements can be represented by using Screw Systems.^{20,21,22} We adapt screw systems representations and basically define task as:

$$Q=[T;R]; T=(Tx,Ty,Tz); R=(Rx,Ry,Rz) \tag{3}$$

In eq. (3), *T* represents the translational displacement and *R* the pure rotational displacement, with respect to origin of coordinate frame, COG. The wrench is represented as :

$$W=[f;m] \tag{4}$$

The wrench is dual to *Q* where *f* is the net force and *m* is the pure moment on a body about its origin. Example tasks represented in wrench form are given in Table 1 as follows:

Table 1 Task Decompositions into Wrench Form.

Task Q1: Translations Along Approach & Opposition Axes, Rotation Around Opposition Axis							
Approach Axis	Opposition Axis	Tx	Ty	Tz	Rx	Ry	Rz
X	Y	1.00	1.00	0.1	0.05	0.8	0.05
Task Q2: Translations Along Approach & Opposition Axes, Rotation Around Approach Axis							
Approach Axis	Opposition Axis	Tx	Ty	Tz	Rx	Ry	Rz
X	Y	1.00	1.00	0.1	0.8	0.05	0.05

In table 1, for translational displacements, 1.00 indicates full existence of displacement along an axis and 0.1 non existence. For rotational displacements, 0.8 indicates full existence of rotation around an axis here as 0.05 indicates non existence. The magnitude values are determined during test simulations.

The approach, opposition and transverse axes for a preshaping hand are illustrated in figure 3. In our system, to achieve a task described in terms of translational and rotational primitives, the hand needs to have an approach and opposition axes representation. The decomposed task determines the set of congruent axes of the object and the hand, as shown in figure 3.

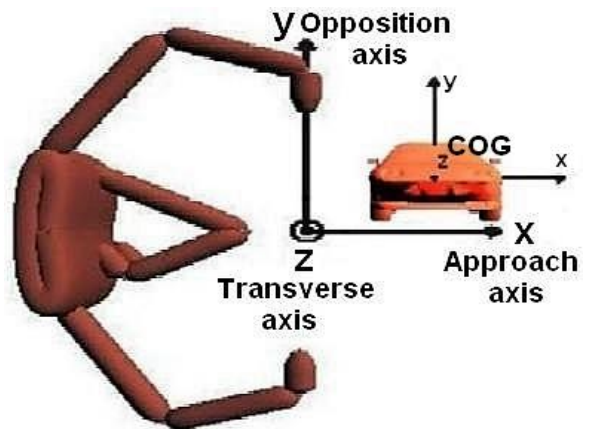


Figure 3. Approach of Hand towards an Object Model

In addition, for precision type grasps, we foresee that the thumb and the forefinger are in opposite direction and there is one contact per finger. This is expressed in our sample task space in terms of categorization of the axes as hand approach planes.

Our further study resulted in the categorization of the hand approach and opposition axes as hand Approach Planes. A 12-Plane categoric scheme that can encompass all possible cases of grasping approach and covering our primitive task space is given in figure 4. In this scheme, horizontal axis for a plane corresponds to the approach axis for the hand and vertical axis to opposition axis. In figure 4, planes are also labeled with plane index numbers, such as 05,11.

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 4.

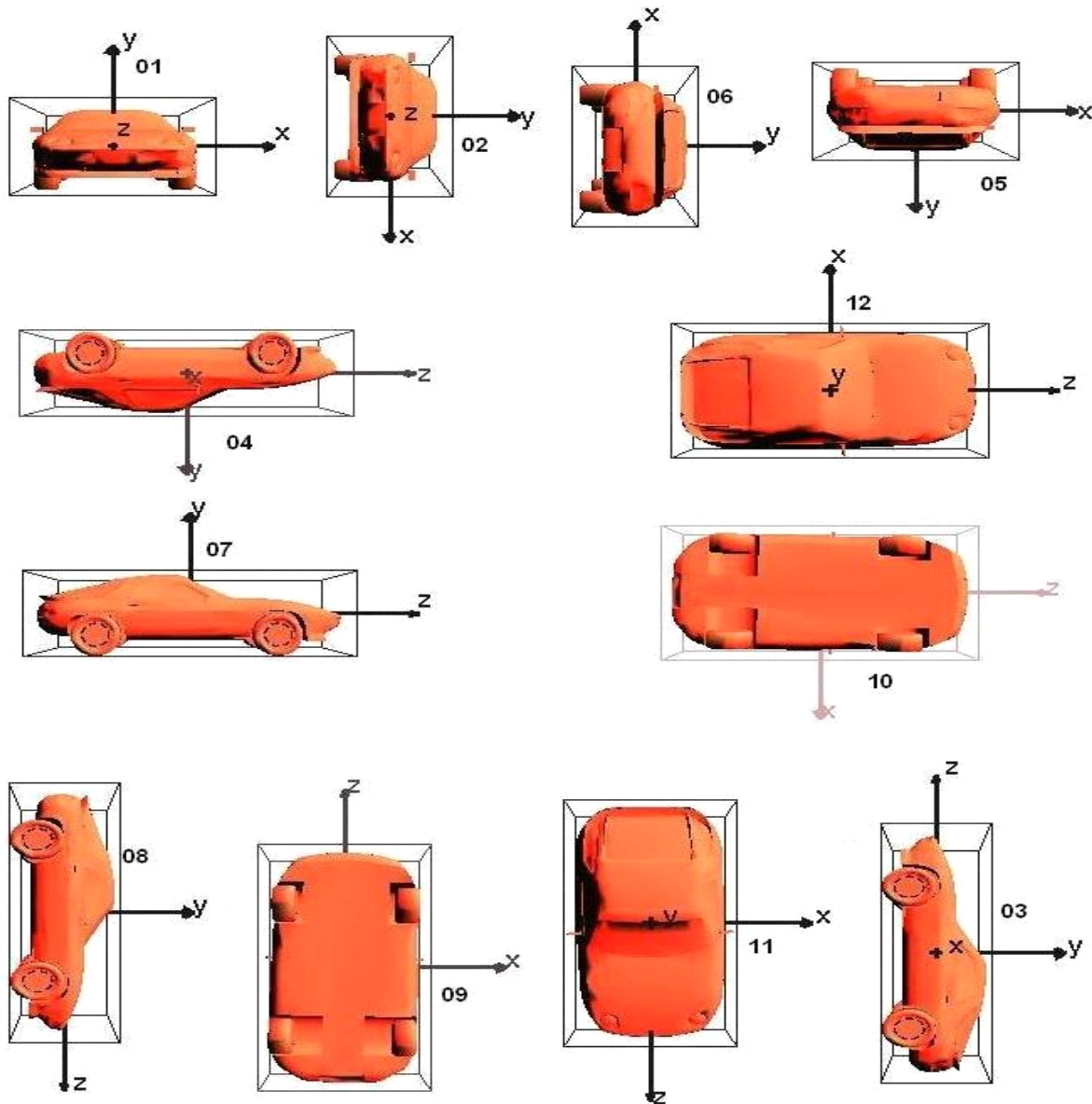


Figure 4. Hand Approach Planes Corresponding to Primitive Task Space.

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. 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 figure 3, the transverse axis of preshape corresponds to the Z axis . 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. Feedforward Neural Networks are widely used for categoric classifications.²³ Hence, we estimate the approach plane and the relevant approach value of robot hand, by a Multilayer Neural Network.

In our work, we implement a 3-layer MLP (Input, Output and a 45-node Hidden Layer) as ANN, in which nonlinear activation function $f(\cdot)$ is logarithmic sigmoid, and a Back Propagation algorithm is used for training. We would like to explain structure of input and

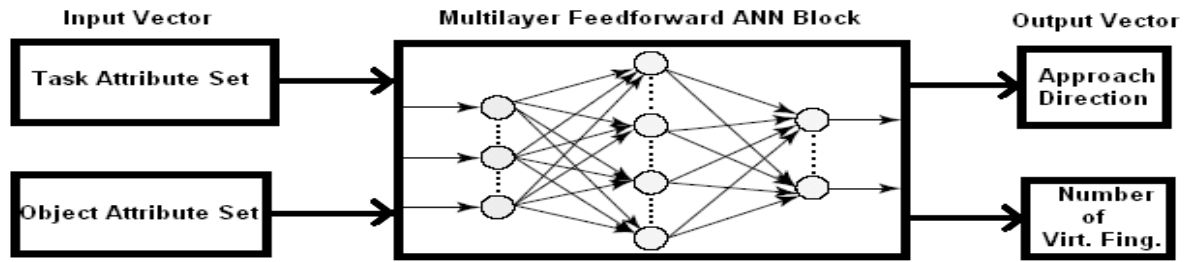


Figure 5. Structure of ANN for Hand Approach Plane Estimation.

output vectors of the ANN used, given in figure 5. The input vector to ANN is composed of task represented in wrench form to represent a spatial displacement as well as object model attributes, such as size attributes, number of vertices, number of mesh triangles of its modeling. The output 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.

Input Vector of ANN, IV={ Task ; Object } (5)

$T = \{ T_x, T_y, T_z, R_x, R_y, R_z \}$;

$O = \{ W, H, D, \#Vert, \#V.Norm, \#Faces, \#Triang \}$

Output Vector: OV={PI ; VF } (6)

$PI = \{ Y_0, Y_1, Y_2, Y_3, Y_4, Y_5, Y_6 \}$; $VF = \{ Y_7, Y_8 \}$

Table 2 Input Vector of ANN.

Task Attributes	Tx,y,z: Translat. Motion along x,y,z axes.
	Rx,y,z: Pure Rotations around x,y,z axes.
Object Attributes	W: Width, H: Height, D: Depth
	#Vert: Number of vertices in object model.
	#V.Norm: Number of vertex normals,
	#Faces: Number of faces in object model.
	#Triang: Number of triangles used in object model representation.

Table 3 Output Vector of ANN.

Plane Index	Y_0, Y_1, Y_2, Y_3 : Binary representation for approach plane index number. E.g. 1001=10
Fuzzy Approach Value	Y_4, Y_5, Y_6 : Binary representation for fuzzy approach value . E.g. 01=Rwd_H
Virtual Finger	Y_7, Y_8 : Binary number of virtual fingers to be involved

Table 4 Training Indicators of ANN.

EMS=2.5e-3		EMS=1.0e-3		EMS=5.0e-4	
Epoch	Train. Time	Epoch	Train. Time	Epoch	Train. Time
8424	72 sec.	15864	85 sec	28032	145 sec
6960	66 sec	13272	75 sec	30240	170 sec
9432	89 sec	17832	105 sec	38184	210 sec

2.3. Kinematic estimations for virtual grasp patterns

The kinematic effects of contacts on a body can be analyzed using constraint analysis methods. From kinematic perspective, a multifingered robot hand is basically a parallel manipulator mechanism composed of serial chained links operating in parallel.^{24,25} 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. Salisbury²⁶ developed contact models for robotic grasping using screw systems. Mason and Salisbury²⁷ analyzed the contact kinematics of robot hand based on contact models using screws. In our work, we use Salisbury's²⁶ soft finger contact model which has been less worked on. Salisbury²⁶ 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 as indicated by K_a, K_n, K_o and M_n .

We implement our virtual grasp using the mentioned contact model and demonstrate it on a case, in order to estimate kinematic effects of the virtual grasp. Given the approach plane, the number of fingers and the approach value, we determine contact locations by

making collision test on object 3D model to verify that our fingers land on contact points and assign contact parameters to each virtual contact, as shown in figure 7.

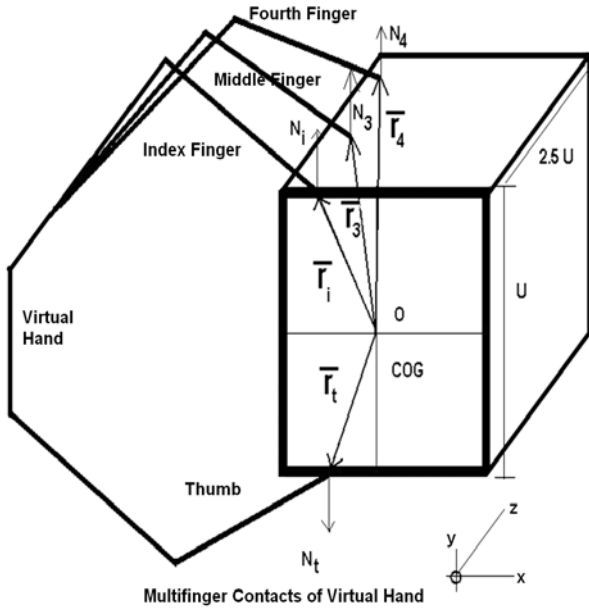


Figure 7. Virtual Grasp with Contact Parameters.

We also calculate the contact position vectors r_i with respect to the COG (Object Geometric Center) of the object which is estimated in object analysis module. The contact position vector r_i is calculated using the results from collision tests for the contacts and stored. The position vector is defined as:

$$r_i = (x_i, y_i, z_i) \tag{7}$$

Indexes are r_1 for thumb, r_2 for index, r_3 for middle finger and r_4 for fourth finger, where C_i denotes the contacts and W_i denotes contact wrench.

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

$$W_i = [f_i; m_i] \tag{8}$$

where f_i denotes the force applied at the contact and m_i denotes the moment existed at the contact. From the figure 7, 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 C_2, C_3, C_4 which are indicated by position vectors r_i in figure 7 are same. For illustration purposes, we write wrenches only for

thumb contact C_1 and index finger contact C_2 . However, in the implementation all contact wrenches are individually estimated. If we rewrite W_1 and W_2 in this form, for the figure 7, we obtain below couples:

$$W_1 = \begin{bmatrix} 1 & -1 & 1 & ; & 0 & -1 & 0 \end{bmatrix} \tag{9}$$

$$W_2 = \begin{bmatrix} 1 & 1 & 1 & ; & 0 & 1 & 0 \end{bmatrix} \tag{10}$$

In order to exemplify the contact wrenches, we use unity magnitudes in (9) & (10). When we include frictional coefficients in wrenches, we obtain following wrench equations (11) & (12).

$$W_1 = \begin{bmatrix} K_a & -K_n & K_o & ; & 0 & -M_n & 0 \end{bmatrix} \tag{11}$$

$$W_2 = \begin{bmatrix} K_a & K_n & K_o & ; & 0 & M_n & 0 \end{bmatrix} \tag{12}$$

The values for coefficients are determined during simulations and their values are discussed in the results section. After having calculated contact position vectors and couples for each contact, we use Poinot'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. 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 eq. (13) and (14).

$$W_o(k) = \begin{bmatrix} f_o \\ m_o \end{bmatrix} = \begin{bmatrix} \sum f_i \\ \sum_j \frac{m_j}{r_i} + \sum_i r_i x f_i \end{bmatrix} \tag{13}$$

$$W_{net} = \sum_k W_o(k) \tag{14}$$

where, in our case for each finger, $W_o(k)$ 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 C_i , and r_i the contact position

vectors from the COG, given in figure 7 (x) operator is the cross product operator. In our case, the *i* index could range from 1 to 3, *k* ranges from 1 to number of fingers and *j* is 1 for our finger model. The Resultant Net Wrench from all fingers are obtained direct sum of projected couples. We use resultant net wrench W_{net} , as the representation pattern for a proposed grasp by avoiding complexity and computational burdens of the Minkowski sums and convex hull methods used in the other works.

As the grasp performance measures, we developed new measures which are “Stability Index” and “Manipulability Index” using the results obtained kinematic analysis of virtual grasp, by utilizing the properties of screw systems. We use Wrench Union (WU), the union of $W_o(k)$, to represent Grasp Wrench Space. We also use Twist Intersection (TI) to represent Grasp Twist Space and utilize the reciprocity theorem between twists and wrenches. The reciprocal twist space of a (WU), if exists, represents hidden degrees of freedom which indicates a degree of instability in the (WU).

Then, we define “Stability Index” and “Manipulability Index” as follows:

The Stability Index:

$$Sta = rank(WU) - rank(TI), \text{ valued in } [0..6] \quad (13)$$

The Manipulability Index :

$$Manu = rank(WU), \text{ valued in } [0..6] \quad (14)$$

In order to find the reciprocal space twist, we make use of (WU) of the grasp and the algorithm developed by Adams and Whitney²⁸ to calculate reciprocal twist space and hidden degrees of freedom under constraints.

2.4. Task and virtual grasp pattern matching using fuzzy logic

In the previous sections, we have formulated a wrench representation of the task as a spatial displacement *Q* and interpreted the grasp net wrench W_{net} on the object as a pattern representing the virtual grasp. Hence, the matching degree between these two patterns can be analyzed in order to infer about suitability of the task *Q* and proposed preshape executed on object model and represented by a virtual grasp pattern W_{net} , represented in (15) & (16).

$$Q = \begin{bmatrix} T_x & T_y & T_z & R_x & R_y & R_z \\ q_1 & q_2 & q_3 & q_4 & q_5 & q_6 \end{bmatrix} \quad (15)$$

$$W_{net} = \begin{bmatrix} T_x & T_y & T_z & R_x & R_y & R_z \\ w_1 & w_2 & w_3 & w_4 & w_5 & w_6 \end{bmatrix} \quad (16)$$

Although there are several possible methods for matching patterns represented in vector space notation, such as distance based methods, we have chosen similarity based methods to be able to infer in symbolic level. In addition, to avoid possible errors which could originate from estimations, we use approximate reasoning methods. To develop the pattern matching, we inspected fuzzy pattern matching, especially fuzzy similarity measures. In this work, approximate reasoning based on fuzzy logic is selected for task and virtual grasp mapping to avoid possible uncertainties coming from sensitivity to virtual contact parameters and possible computational errors. A comparative survey of fuzzy similarity methods are given by Yeung et al.²⁹ Among fuzzy similarity methods, “Fuzzy Approaching Degree Index” is the most convenient for symbolic level inference using fuzzy vectors. Given two Fuzzy sets or vectors of dimension *N*, as *A* and *B*, defined on Universe *U*:

$$A = \left\{ \frac{\mu_{a1}}{x_1}, \frac{\mu_{a2}}{x_2}, \frac{\mu_{a3}}{x_3}, \dots, \frac{\mu_{an}}{x_n} \right\} \quad (17)$$

$$B = \left\{ \frac{\mu_{b1}}{x_1}, \frac{\mu_{b2}}{x_2}, \frac{\mu_{b3}}{x_3}, \dots, \frac{\mu_{bn}}{x_n} \right\} \quad (18)$$

“Fuzzy Approaching Degree Index” which is derived by Wang³⁰, measures similarity between two fuzzy sets or vectors. ADI(A,B) yields a scalar value between 0 and 1.

$$ADI(A, B) = (A \bullet B) \wedge \overline{(A \oplus B)} \quad (19)$$

$$(A \bullet B) = \bigvee_i^n (a_i \wedge b_i) \quad (20)$$

$$(A \oplus B) = \bigwedge_i^n (a_i \vee b_i) \quad (21)$$

In order to use “Fuzzy Approaching Degree Index”, we need to fuzzify our patterns which are in wrench form. For fuzzification of wrench patterns, we have implemented different membership functions, such as triangle, gaussian and trapezoid forms. We observed that trapezoid membership function yields better results during implementation phase.

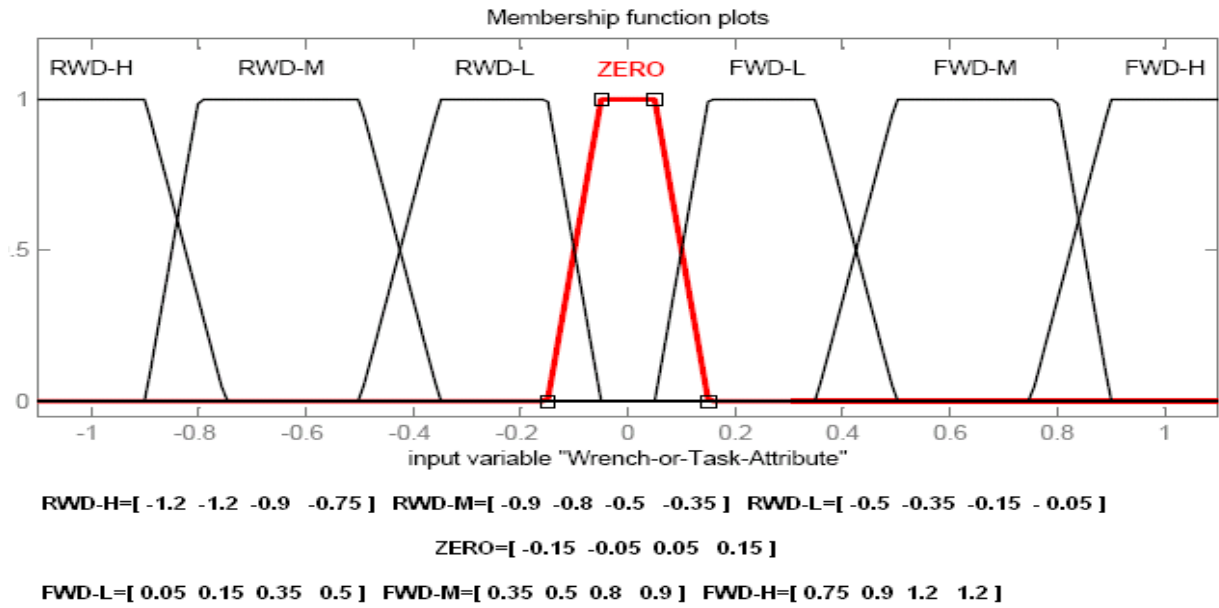


Figure 8. Fuzzy Membership Functions for Grasp Wrench or Tasks.

The membership function with 7 fuzzy sets used is indicated in figure 8 . We have 7 fuzzy sets named as RWD-H, RWD-M, RWD-L, Zero, FWD-L,FWD-M,FWD-H, with descriptions given in table 5. We have used 7 sets to have a moderate level of granularity and symmetry on both positive and negative side. A fewer number of fuzzy sets yields a coarse estimations and a higher number of fuzzy sets yields more granulation but higher cost of computations and slower controller.

Table 5 Explanation of Fuzzy Sets.

Fuzzy Set	Description
RWD-H	A high quantity in negative direction
RWD-M	A medium quantity in negative direction
RWD-L	A low quantity in negative direction
Zero	A zero quantity
FWD-H	A low quantity in positive direction
FWD-M	A medium quantity in positive direction
FWD-L	A high quantity in positive direction

The wrench patterns are fuzzified using m-Fuzzy Vectors or Fuzzy Matrix of MxN. Hence our wrenches are 6 dimensional vectors, we have 6-Fuzzy Vectors to represent a wrench. When we apply ADI operator on our pattern we obtain 6 output 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.

$$TFVADI(Q, W_{net}) \triangleq \sum_{i=1}^M ADI(q_i, w_i) \quad (22)$$

where Q represents the task and W_{net} represents the resultant wrench of grasp. We use TFVADI given in (22) as a matching criteria between task pattern Q and virtual grasp pattern W to infer about suitability of task and estimated grasp preshape.

3. System Architecture

3.1. Overview of the intelligent inference system

We integrate our novel methodic approaches which are individually different in nature, into an Intelligent Inference System (IIS) . The IIS is formed by the synthesis of object properties, manipulation task characteristics, virtual kinematic analysis and evaluation of contact wrench patterns via fuzzy approximate reasoning and developed ANN structure for incremental learning. The IIS takes task and object model as inputs and yields a resultant grasp preshape selecting a virtual approach to the predefined object by analysis of possible virtual grasps with kinematic estimations and grasp quality measures. In order to reach our preshape determination goal, we design IIS as Hybrid Intelligent Inference System (HIIS) .

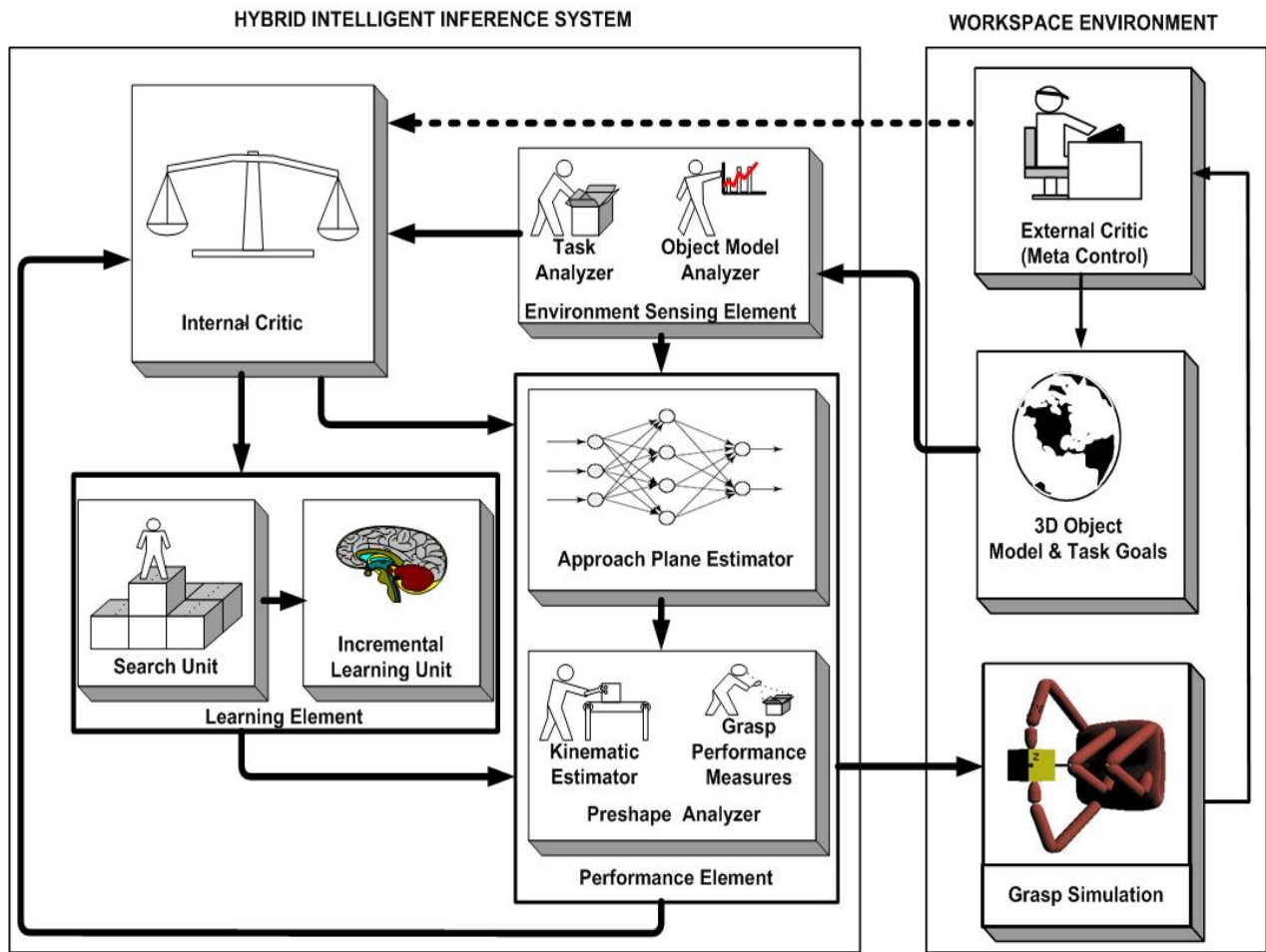


Figure 9. Overview of the Hybrid Intelligent Inference System Architecture

The design model is compatible with Hybrid Intelligent Agent Structure^{31,32}, in the aspect of existence of individual units, such as performance element, learning element, internal critic unit, environment sensing element and in their interaction. In figure 9, overall block representation of HIIS is given.

In the architecture of figure 9, we have environment items which are external critic as a meta control, objects to be grasped as represented 3D models, task actions to be performed as grasp and grasp simulation to indicate result of the HIIS to external critic. The external critic is the human operator in our case. Grasp simulation is the end effector. The grasp performed by the grasp simulation is limited to a quasistatic precision fingertip grasp and positions of the hand and object are known at each phase. The external critic adjusts the parameters of

the internal critic and it inputs the task actions via user interface of the controller and also selects the object 3D model via user interface. Internal units of our controller are explained in the following sections with their functionality.

3.2. Environment sensing element

“Environment Sensing Element” (ESE) 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. Task Q is converted to low-level grasp kinematics in wrench form.

3.3. Performance element

In our design, “Performance Element” (PE) is the execution unit which takes inputs from ESE in terms of task attributes and object attributes. It is composed of two successive parts: Approach Estimator (AE) which is composed of an ANN and Preshape Analyzer (PA) which estimates the virtual grasp kinematic estimations on object 3D model. The PA performs grasp kinematic estimations and grasp performance measures according to the outputs of AE that is given in terms of hand virtual approach towards object. For a selected Approach Plane, which determines approach and opposition axes of virtual grasp with respect to object geometry, 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. In the estimation, virtual grasp formation is done using collision based exact 3D geometric contact detection algorithms such as “Ray Triangle Intersection” method to determine possible contact positions.³³ In this way, we test existence of a contact surface on the object for a given approach. After virtual contacts are determined, PA estimates W_{net} representing the proposed virtual grasp for the given Approach Plane and Approach value, with grasp measures.

3.4. Internal critic

“Internal Critic” (IC) has a role to decide about performance of the execution of the HIIS rather than deciding for the performance of Learning Element (LE) as stated in its general definition. For our case, it decides about execution performance of the HIIS Controller via reinforcement, indicating success as reward and failure as penalty. This results in determination of the execution mode of the HIIS. It uses fuzzy approximate reasoning for criticism. It uses TFVADI to decide about matching degree of task and resultant grasp preshape, using the various thresholds. Indeed, some of the execution modes involves activation of learning element by internal critic. The internal critic takes inputs from PE, ESE and inputs from External Critic. It send its output to PE, LE as reinforcement and to Meta Control as report. The PE and LE depending on the HIIS execution mode, gets reinforcement from IC and using the Multistate Decision algorithm for execution both synchronously, they either decide to run at a mode or to stop.

3.5. Learning element

“Learning Element” (LE) is composed of a “Search Unit” to find a proper coverage of workspace and Incremental Learning Unit (the ANN of PE in supervised training mode). The input to LE is from internal critic and environment sensing element. Output from LE is sent to PE, to internal critic and to external critic as report. Learning of a new objects occur as ‘Incremental Learning’. Initially, the ANN is basically trained and training data is composed of only for a simple cube. The other objects are introduced by “Meta Control” to the controller one by one with some sample tasks from our task space representation. When learning element is activated, first, for the given object and task, the search unit runs and finds the corresponding “Best Possible Grasp”. This information is added to training data for ANN and IO training is performed in supervised learning mode. It recurses the PE after IO training, to test its learning performance by internal critic. Hence, learning performance also tested by reinforcement. With this structure, learning of the overall HIIS is an “Incremental Learning” which is a hybrid combination of supervised and reinforcement learning.

3.6. Execution modes of the HIIS

Execution of the HIIS is very similar to a Multistate Decision Control. The HIIS is a Four State Controller. States are successive. The flow diagram for execution modes for HIIS are given figure 10.

Execution Mode 0: This is the normal execution of PE with Internal Critic using Execution Threshold. The HIIS 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 HIIS send its output to Grasp Simulation, it stops and waits next command from Meta Control. If a deadlock is reset by External Critic, the 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 HIIS send its output to Grasp

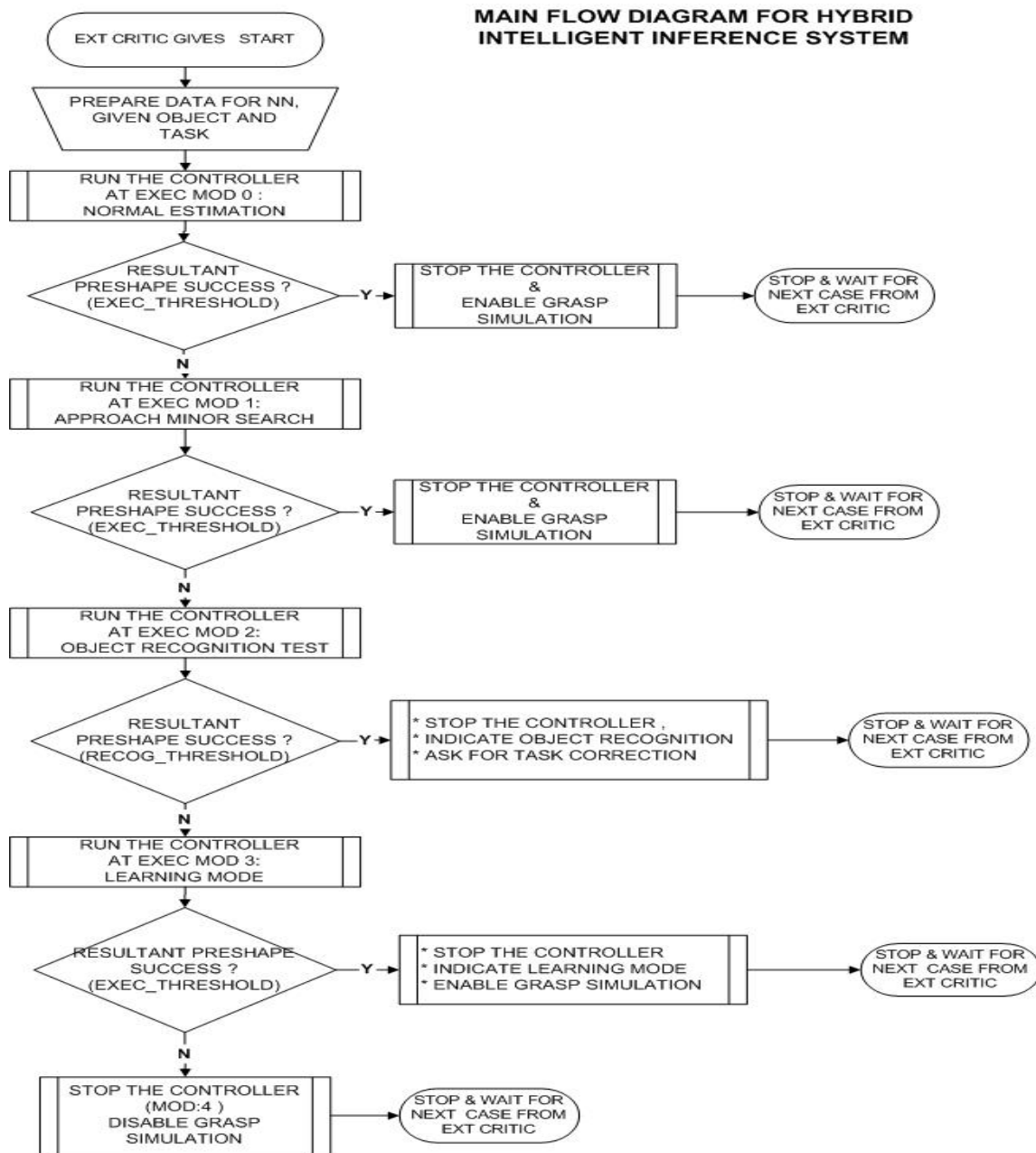


Figure 10. HIIS Execution Flow Diagram

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 HIIS 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 HIIS stops and reports to Meta Control recognition 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 used for training new object and new task to the HIIS . After Object is reconized as new Object, 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 this way, both a new object and a possible new task is taught to controller. In order to test LE learning performance, the HIIS executes itself at mode 0 and if not successful executes at mode 1 and after all, if it fails, controller reports a deadlock and stops.

4. Results and Discussions

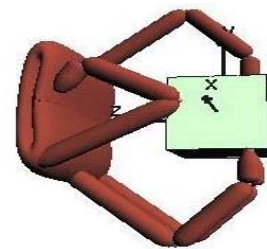
The designed HIIS architecture is implemented as a software package. In our work, we have used 3D object models in forms of vertex and wireframe solid models of objects to be manipulated. In order to demonstrate the performance of HIIS in a better environment, we have also implemented a basic Robot Hand Simulation toolbox only with kinematics. The Hand is a 4 fingered 16 DOF hand model which is similar to UTAH-MIT Hand. 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 the implementation, we assume that hand base frame can move around object within any path or without any obstacle avoidance. The all software source code is written in C++ programming language and using OpenGL Libraries on a PC environment.

We have tested HIIS with certain categories of tasks and objects. We have investigated learning capability, parametric sensitivity and estimation performance of the HIIS as well as its robustness. In the result figures, desired task Q is represented in wrench form. Output values for the HIIS are given according to each result category such as Approach Plane #, approach value, TFVADI and Execution Mode Index as well as the simulated grasp. In some cases, Wnet is also given to justify the performance. In the following sections, we present our results with our basic comments.

4.1. Results for testing performance robustness of HIIS using pre-trained object

In this category of results, we demonstrate robustness of the HIIS performance to, variations in task, using pre-trained object. In the results given in

figure 11, as case #1, for the given basic training object, we request the task Q as a translational motion along Y axis with a rotation around Y axis, as well as translation along Z axis, all according to COG. However, there are also added disturbance items in rotational terms for X and Z rotations. When we inspect the results given in the figure, we could see that The HIIS yields Approach Plane as #7 and Approach Value as 0.1. It means that robot hand will approach along Z axis while opposition axis is Y and estimated virtual grasp pattern matched to desired task with TFVADI value of 4.09 /6.00. The execution mode is 0 which indicates controller run in the mode remembering both object and task category. A successful grasp simulation is the final result. In addition, response of HIIS is satisfactory to tolerate disturbances in task which do not exist in our training data set. Task disturbances are tolerated for a trained object with success, giving the correct approach plane, approach and indication of proper execution mode. In case #2, we have changed the task Q as a translational motion along X & Y axes and with a rotation around X axis. HIIS gives us Approach Plane as #6 and Approach Value as 0.1. It means that robot hand will



#	Task Q			Approach Plane/ Value	TFVADI	Controller Execution Mode
	Tx Rx	Ty Ry	Tz Rz			
1	0.09 0.20	0.96 0.83	0.93 0.24	Plane: 7 Value: 0.1	4.09	0
2	1.48 0.84	1.28 0.05	0.05 0.05	Plane: 6 Value: 0.04	3.6	1
3	1.04 0.79	0.92 0.94	1.11 0.65	Plane: 5 Value: 0.01	5.31	2

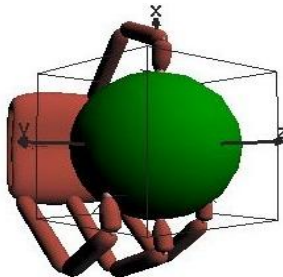
Figure 11. Testing Controller Robustness

approach along Y axis while opposition axis is X and estimated virtual grasp pattern matched to desired task with TFVADI value of 3.6 /6.00. The execution mode is 1 which indicates controller run in the local search

mode along approach axis but remembering both object and task category. In case #3, we ask the HIIS an almost impossible task with this object geometry which is prismatic. The task as a translational motions along X,Y,Z axes; with rotations around X,Y,Z axes. As indicated by the results, the HIIS recognized the object with TFVADI 5.31 and asked for task correction in execution mode 2. In this case, we see that HIIS recognized the trained object under variations of task attributes. The responses of the HIIS to disturbances in tasks and to irrelevant tasks indicates the robustness of the HIIS.

4.2. Results with training for primitive objects

In these group of results, we aim to test learning performance of the HIIS for relatively simple objects



#	Task Q			Approach Plane / Value	TFVADI	HIIS Execution Mode
	Tx Rx	Ty Ry	Tz Rz			
1	1.31	0.76	0.11	Plane: 5	3.78	3
	0.86	0.07	0.07	Value: 0.13		
2	1.05	0.13	0.96	Plane: 10	3.6	1
	0.84	0.05	0.05	Value: 0.02		
3	1.03	0.05	1.15	Plane: 9	4.0	0
	0.05	0.05	0.91	Value: 0.01		

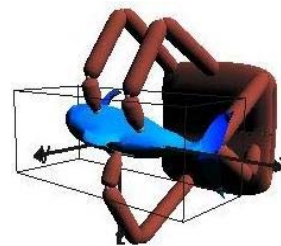
Figure 12 Testing HIIS learning with a simple object

,using previously trained tasks. In the results of figure 12, for the given basic tasks, we ask the HIIS to recognize a new primitive object and ask to learn it. As case # 1, the task, a pre-trained task, is given as translational motions along X, Y axes; with rotation around X axis. The HIIS recognized new object and learned it successfully, in execution mode 3 with TFVADI 3.78. In case #2, we would like to test execution performance of the HIIS after learning the object, for another basic task. The task is given as

translational motions along X and Z axes; with rotation around X axis. The HIIS performed execution of the task at exec mode 1, with TFVADI 3.6 indicating that object remembered and a suitable grasp for task is yielded. As case #3, we would like to test execution performance of the HIIS, for another basic task. The task is given as translational motions along X and Z axes with rotation around, Z axis. As seen from grasp simulation and preshape results, the HIIS performed for another basic task, at execution mode 0, demonstrating successful learning of the object, with TFVADI 4.0.

4.3. Results with training for complicated objects

In this category, we test learning performance of the HIIS with relatively difficult objects.



#	Task Q			Approach Plane / Value	TFVADI	HIIS Execution Mode
	Tx Rx	Ty Ry	Tz Rz			
1	1.08	-1.0	0.05	Plane: 11	5.3	3
	0.05	-0.8	0.05	Value: 0.01		
2	1.01	0.97	0.05	Plane: 5	4.0	1
	0.05	0.85	0.05	Value: 0.22		
3	0.75	0.1	0.96	Plane: 7	3.0	1
	0.07	0.1	0.76	Value: -0.03		

Figure 13 Testing HIIS Learning with a complex object

In the results given at figure 13 for case #1, for the given basic task, we ask the HIIS 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 around -Y axis, a pre-trained task. As seen from grasp simulation and preshape results, the HIIS recognized new object and learned it successfully. The execution mode 3 indicates the learning of new object and it occurred with a relatively high matching degree TFVADI 5.89 due to given a very optimal task which yields our conclusion that complicated objects should be trained with compatible tasks. As case #2, we

ask the HIIS a possible but relatively difficult task as translational motions along X & Y axes and with a medium valued rotation around Y axis. As seen from results, the HIIS 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 local search. As case #3, we ask the HIIS a possible but relatively difficult task as translational motions along X & Z axes; with a rotation around X axis. Successful Grasp simulation results indicate that the HIIS performed correctly but with a local search of contact places along approach axis.

4.4. The HIIS performance with variations in contact model parameters

In the following section, we give performance results with variations in contact frictional coefficients. In virtual grasp case, these coefficient are only effective for grasp matching. The Task is arbitrarily chosen from our sample task space. The other parameters remain unchanged. The parameters, Ka,Ko,Kn and Mn are used to represent soft finger contact wrench coefficients, as in actual implementations of soft finger by Howe and Cutkosky³⁴. 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 Murakami³⁵. In our models we take friction coefficient as $\mu=1.0$.

The results given in figure 14, we present our best found values highlighted with bold typeface. It is important to mention that, variations in contact friction parameters, do not effect plane index and approach determination, but may effect approximate reasoning part of HIIS, as it would yield different wrench magnitudes for a given task and object.

4.5. The HIIS performance with variations in threshold parameters

The HIIS is a multistage decision system and transition between stages are determined by the internal critic by using a relevant threshold. In our threshold parameters analysis, indicated by Figure 15, we observed that the system is dependent on thresholds and especially “Estimation Threshold” which effected HIIS overall performance since it is the major switching element for HIIS execution modes and acts in combination with other thresholds.

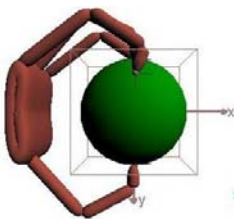
		Task Q=[1.0 1.0 0.1 0.05 0.8 0.05] Recog. Threshold=5.0 Estim. Threshold= 3.0 Exec. Mode=0 Approach Plane / Value Plane= 5 Value= 0.01				
		#	Contact Coefficient Parameter Sets	TFVADI	W _n Tx Ty Tz Rx Ry Rz	
1	Ka=0.3 Ko=0.2 Kn=1.0 Mn=0.9	3. 1	0.86 -0.03	1.1 0.96	-0.22 0.06	
2	Ka=0.45 Ko=0.3 Kn=1.0 Mn=0.8	3. 71	1.3 0.06	1.1 0.86	-0.27 0.11	
3	Ka=0.55 Ko=0.35 Kn=1.0 Mn=0.75	3. 0	2.1 0.38	1.2 0.71	-0.49 0.2	

Figure 14 Sensitivity to Contact Coefficient Parameters.

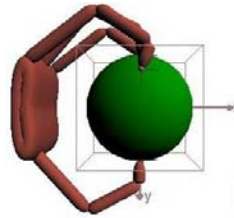
		Task Q=[1.0 1.0 0.1 0.05 0.8 0.05] Recog. Threshold=5.0 Estim. Threshold= 3.0				
		Estimation Threshold	Approach Plane/ Value	TFVADI	HIIS Execution Mode	W _n Tx Ty Tz Rx Ry Rz
3.0	Plane: 7 Value: -0.1	3.25	0	0.27 -0.11	1.0 0.82	2.7 1.2
3.5	Plane: 7 Value: 0.09	3.77	1	0.27 -0.2	1.1 0.88	0.36 -0.61
3.9	Plane: 7 Value: 0.08	5.74	2	0.27 -0.18	1.1 0.87	1.3 0.05
4.2	Plane: 7 Value: 0.08	5.74	2	0.27 -0.18	1.1 0.87	1.3 0.05

Figure 15 Sensitivity to Variations in Thresholds.

4.6. The HIIS performance with variations in fuzzy membership functions

In the figure 16, we present the analysis of the HIIS under variations of fuzzy memberships, one membership at each step. By making parameter variations in Fuzzy Membership Functions either narrowing or widening about 20-30 % of the trapezoidal membership functions one at a time, we reached the following conclusion about the effect of the MFs to HIIS. As it could be seen from figure 16, MF variations mainly effective for ZERO MF, for the other MF variations, even there is small change in TFVADI values, overall grasp preshape is almost same.

This is an indication of that the HIIS uses a stable approximate reasoning that is not effected much by parameterization due to fuzzy logic attribute matching technique.

4.7. The Results with Odd Cases

In the following results given in figure 17 and figure 18, eventhough HIIS estimated a preshape for the given task and object, however hand simulation kinematics indicates that these preshapes are not possible with hand constraints using inverse kinematics and the fingers are retracted. The amount of such results are quite low around 15 % overall 500 experiments.

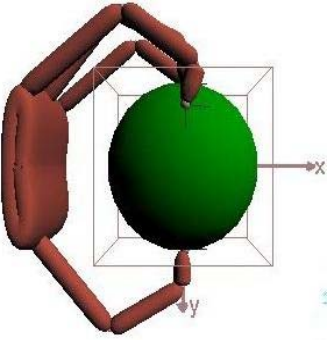
	Task Q=[1.0 1.0 0.1 0.05 0.8 0.05] Recognition Threshold=5.0 Estimation Threshold= 3.0 Default Fuzzy Set Membership Functions (Trapezoid) Rwd-H=[-1.2, -1.2, -0.9, -0.75] Rwd-M=[-0.9, -0.8, -0.5, -0.35] Rwd-L=[-0.5, -0.35, -0.15, -0.05] Zero=[-0.15, -0.05, 0.05, 0.15] Fwd-L=[0.05, 0.15, 0.35, 0.5] Fwd-M=[0.35, 0.5, 0.8, 0.9] Fwd-H=[0.75, 0.9, 1.2, 1.2]					
	Changed Fuzzy Membership Function	Approach Plane/ Value	TFVADI	HIIS Execution Mode	W _n	
Zero*=[-0.08, -0.03, 0.03, 0.08]	Plane: 5 Value: 0.15	6.0	2	Tx Rx	Ty Ry	Tz Rz
Zero*=[-0.15, -0.08, 0.08, 0.15]	Plane: 5 Value: 0.01	3.98	0	-0.48 -1.0	0.82 0.5	-0.2 0.03
Rwd-L*=[-0.4, -0.25, -0.15, -0.08] Fwd-L*=[0.08, 0.15, 0.25, 0.4]	Plane: 5 Value: 0.01	3.98	0	1.3 -0.06	1.1 0.86	-0.27 0.11
Rwd-L*=[-0.6, -0.4, -0.15, -0.05] Fwd-L*=[0.05, 0.15, 0.4, 0.6]	Plane: 5 Value: 0.01	3.98	0	1.3 -0.06	1.1 0.86	-0.27 0.11
Rwd-M*=[-0.9, -0.7, -0.55, -0.35] Fwd-M*=[0.35, 0.55, 0.7, 0.9]	Plane: 5 Value: 0.01	3.88	0	1.3 -0.06	1.1 0.86	-0.27 0.11
Rwd-M*=[-0.95, -0.85, -0.45, -0.35] Fwd-M*=[0.35, 0.45, 0.85, 0.95]	Plane: 5 Value: 0.01	4.48	0	1.3 -0.06	1.1 0.86	-0.27 0.11
Rwd-H*=[-1.2, -1.2, -0.9, -0.85] Fwd-H*=[0.85, 0.9, 1.2, 1.2]	Plane: 5 Value: 0.01	3.98	0	1.3 -0.06	1.1 0.86	-0.27 0.11

Figure 16 Sensitivity to Variations in Fuzzy Membership Functions.

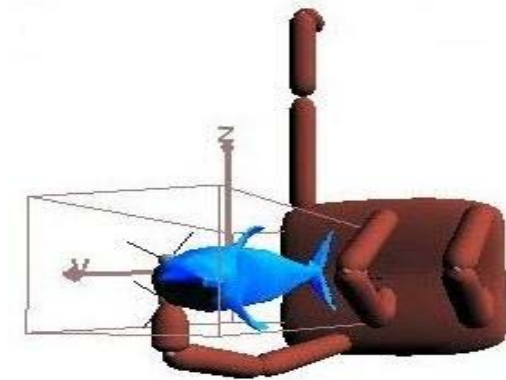


Figure 17 Results with Odd Cases-A

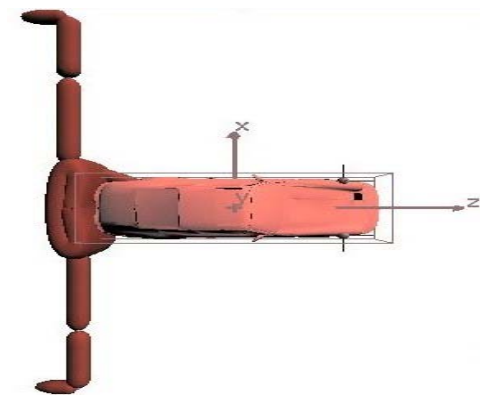


Figure 18 Results with Odd Cases-B

5. Conclusion

In this work, a Hybrid Intelligent Inference System (HIIS) for optimal preshaping and grasping of robot hand is designed and the model is implemented in a software simulation environment. The HIIS model is developed such that it includes grasp planning coming from task requirements, approaching and grasping which is related to object, its pose and environment, as a sole continuum controller. The HIIS uses attributes coming from each grasp phases to deduce a successful preshape for a good initialization of a subsequent grasp task. Object pose, shape, together with environment constraints are also analyzed to select approaching and grasp characteristics. It 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 developed using screw systems.

The results show us that our proposed method combining high level task and low level kinematics for preshape estimation works as an intelligent system. The designed HIIS was implemented using 3D computer simulation environment which demonstrated us that it could be implemented in a real hardware environment. HIIS executes attributes in symbolic forms, which can be easily integrated with a higher level robot control mechanisms. The HIIS for preshaping demonstrated satisfactory results for enough robustness, incremental learning of new objects and tasks. It also handled some uncertainty in tasks. The hand grasp simulation results are encouraging to work further in this approach, especially to implement HIIS in a real hardware environment.

6. References

1. Bicchi A. and Kumar V., Robotic Grasping and Contact: A Review, in *Proc. IEEE ICRA2000*, USA.
2. Okamura A.M. and Cutkosky M.R., An overview of dexterous manipulation, in *Proc. IEEE ICRA2000*, USA.
3. Shimoga K.B., Robot grasp synthesis algorithms: A survey, *The IJRR*, 15(3): 230-266, 1996.
4. Li Z. and Sastry S.S., Task oriented optimal grasping by multifingered robot hands, *IEEE Journal of Robotics and Automation*, Vol. 4, No. 1., 1988.
5. Coelho J.A. and Grupen R.A., Online Grasp Synthesis, in *Proc. IEEE ICRA1996*, USA.
6. Arimoto S., Intelligent Control of Multi-fingered Hands, *Annual Review in Control*, 2004.
7. Ferrari C. and Canny J., Planning optimal grasps, in *Proc. IEEE ICRA1992*, France.
8. Pollard N.S., Closure and Quality Equivalence for Efficient Synthesis of Grasp from Examples, *The IJRR*, 23(6):595-613, 2004.
9. Borst Ch., Fischer M. & Hirzinger G., Grasp Planning: How to Choose a Suitable Task Wrench Space, in *Proc. IEEE ICRA 2004*, USA.
10. Cutkosky M.R., On grasp choice grasp models and the design of hands for manufacturing tasks, *IEEE Trans. Robotics and Automation*, Vol. 5, No. 3., 1989.
11. Iberall T., The Nature of Human Prehension: Three Dexterous Hands in One, in *Proc. IEEE ICRA1987*, USA.

12. Kang S.B. and Ikeuchi K., Toward automatic robot instruction from perception-mapping human grasps to manipulator grasps, *IEEE Trans. Robotics and Automation.*, Vol. 13, 1997.
13. Bowers D.L. and Lumia R., Manipulation of Unmodeled Objects Using Intelligent Grasping Schemes, *IEEE Trans. Fuzzy Systems*, 11(3), 2003.
14. Gorce P. and Rezzoug N., Grasping Posture Learning with Noisy Sensing Information for a Large Scale of Multifingered Robotic Systems, *Journal of Robotic Systems*, 22 (12) ,pp.711-724, 2005.
15. Miller A. T. et al, Automatic grasp planning using shape primitives, in *Proc. IEEE ICRA2003*, USA.
16. Berenson D., Diankov R., et.al., Grasp Planning in Complex Scenes , in *Proc Humanoids 2007*.
17. Ciocarlie M.T and Allen P.K., Hand Posture Subspaces for Dexterous Manipulation, *The IJRR*, 28(7):851–867, 2009.
18. Oztop E., Bradley N.S. and Arbib M. A., Infant Grasp Learning: Computational Model, *Exp. Brain Research* 158:480-503, Springer-Verlag, 2004.
19. Michelman P., Precision Object Manipulation with a Multifingered Robot Hand, *IEEE Trans. Robotics and Automation*, Vol. 14, No.1, 1998.
20. Duffy J., *Statics and Kinematics with Applications to Robotics*, Cambridge University Press, NY, 1996.
21. Waldron K.J., The Constraint Analysis of Mechanisms, *The Journal of Mechanisms*, Vol. 1, pp 101-114., 1966.
22. Roth B., Screws, Motors, and Wrenches that Cannot be Bought in a Hardware Store, *Robotics Research*, 1984.
23. Haykin S., *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, NJ, 1999.
24. Murray R., Li Z. and Sastry S., *A Mathematical Introduction to Robot Manipulation*, CRC Press, 1994.
25. Konkar R., Incremental Kinematic Analysis and Symbolic Synthesis of Mechanisms, *PhD Thesis, Dept. of Mech. Engineering, Stanford University*, 1993.
26. Salisburt J.K., Kinematic and Force Analysis of Articulated Hands, *PhD Thesis, Dept. of Mech. Engineering, Stanford University*, 1982.
27. Mason M. T. and Salisbury J. K. , *Robot Hands and the Mechanics of Manipulation*, MIT Press, 1985.
28. Adams J.D and Whitney D.E., Application of Screw Theory to Constraint Analysis of Mechanical Assemblies Joined by Features, *Journal of Mechanical Design, Trans. of ASME*, vol 123, pp26-32, March 2001.
29. Yeung D.S and Tsang E.C.C, A Comparative Study on Similarity Based Fuzzy Reasoning Methods, *IEEE Trans. on SMC Part B: Cybernetics*, vol 27 No 2, 1997.
30. Wang P., *Approaching Degree Method in Fuzzy Sets Theory and Its Applications*, Science and Technology Press, Shanghai, 1983.
31. Jacobsen H.A., A Generic Architecture for Hybrid Intelligent Systems , in *Proc IEEE-FUZZ 98* , 1998,USA.
32. Abraham A., Hybrid Intelligent Systems Design-A Review of a Decade Research, *Technical Report, School of CIT, Monash University*, 2000 , Australia.
33. Moller T. and Trumbore B., Fast, Minimum Storage Ray-Triangle Intersection, *Journal of Graphics Tools* 2(1), 21-28 (1997).
34. Howe R.D. and Cutkosky M.R, Practical Force-Motion Models for Sliding Manipulation, *The IJRR*, 15 (6) :557-572, 1996, USA.
35. Murakami K. and Hasegawa T., Tactile Sensing of Edge Direction of an Object with a Soft Fingertip Contact, in *Proc IEEE ICRA2005*, Spain, pp 319-325.