13

INTELLIGENT ROBOT PREHENSION

Thang N. Nguyen
IBM United States
Systems Management Consulting
Washington Systems Center
Gaithersburg, Maryland 20879

Harry E. Stephanou
Center for Advanced Technology
in Automation and Robotics
Rensselaer Polytechnic Institute
Troy, New York 12180

Abstract

The intriguing ease with which humans perform grasping and manipulation activities has triggered new investigations to provide robots with prehensile capability for complex tasks in unstructured environment. These investigations resulted in a number of schemes ranging from high-level, AI-oriented, distributed, symbolic schemes to low-level, contact-based, numeric schemes. Current numeric schemes, however, are limited to tip prehension (i.e. intentional grasping by the fingertips), while symbolic schemes have been investigated rather separately from the former and the results of the linking between symbolic and numeric schemes are rather modest. This chapter deals with an intelligent, integrated symbolic-numeric scheme for dextrous manipulation, using a topological approach. In this chapter, we introduce (i) a grasp-based, topological model of multifingered robot hands, with its associated reasoning scheme called topological reasoning and (ii) discuss its application to dextrous *grasps in the framework of an intelligent robot prehension system. Our topological model for multifingered robot hands is capable of describing an arbitrary multifingered hand posture with associated topological algorithms for grasp selection and regrasping. We show that using topological reasoning, both the hand posture and the hand functionality can be derived from symbolic, high level task requirements and object attributes, and can be transformed into numeric, low level, joint space variables. A discussion on simulation results is given.*

1. INTRODUCTION

Grippers and special-purpose end-effectors are adequate for simple operations in structured environments (in which task, object identity, location and orientation are well defined), but have limited capabilities in performing complex, versatile tasks in unstructured environments. Some limitations of grippers are well known, for example, they are unable to adapt to a broad range of object shapes, and they are unable to perform small displacements without moving the entire arm. The use of a *Remote Control Center* device from Draper labobratory placed between the arm and the hand has somewhat solved the problem of producing small motions, but it is not sufficient to provide both the moving and grasping functions (Mason and Salisbury, 1985). With special-purpose tools, one can solve a larger variety of tasks, but one still needs to change tools, as well as control programs, for different tasks. Consequently, in the past decade, *multifingered robot hand* designs and the associated *hand control* algorithms for grasping and manipulation have received a considerable amount of attention in research laboratories.

A multifingered hand has sufficient functional richness to permit dextrous manipulation. However, hand control as a problem of simultaneously controlling multiple fingers, each finger with some degrees of freedom, is extremenly complex. Indeed, in the case of a simple gripper attached to a 3-dof wrist mounted on a 3-dof robot manipulator, the control problem is commonly expressed as a mapping of a six-dof space into the space of $R^3 \times SO(3)$ where R^3 is the space of gripper positions, and $SO(3)$ is its three dimensional space of orientations. In the case of a multifingered hand, the control problem is expressed as a mapping from R^n, where n is the total number of dofs of the hand, into the space of $P = SE_1(3) \times \cdots \times SE_k(3)$ where each $SE_i(3)$ is a Euclidian space isomorphic to $R^3 \times SO(3)$. This shows how complex the control problem may become.

Generally speaking, to explore the applicability of higher dexterity presumably provided by multifingered robot hands, both (i) numeric control schemes have been developed and (ii) AI-oriented schemes have been investigated. Numeric control schemes for dextrous manipulation are contact-based, i.e. based on the modeling of contacts between the robot hands and the objects (Salisbury, 1982). These numeric schemes are generally fairly adequate, although they suffer from high computational complexity due to, for example, finger coupling. They are, furthermore, unable to handle task descriptions specified ambiguously in symbolic terms.

On the other hand, most AI-oriented, symbolic control schemes are based on the observations of human hand functionalities and activities, and are task-oriented. They have not been too sucessful in automatically generating a variety of different grasps, from task and object constraints, that can be described in analytical terms for use by numeric control schemes (Cutkosky, 1989). The need for intelligent, integrated control schemes has also been recognized but has been progressing modestly, due possibly to the lack of uniform grasp representations for both symbolic and numeric schemes.

At the symbolic level, faced with an infinite number of possible grasp selections for a given task, researchers have attempted to reduce the set of feasible grasps into a small, finite set of discrete, common grasps (Lyons 1985, Iberall 1987, Cutkosky 1989). Most associated symbolic reasoning algorithms are based on those grasp models. These algorithms rely on anthropomorphic (Cutkosky et al. 1986), neurophysiological (Arbib et al. 1983), or behaviorial (Tomovic et al. 1987) approaches. These approaches have their root in a study of natural prehension by Napier (1956), a medical surgeon, supplemented by other medical researchers such as Landsmeer (1962), Tubiana (1981), Harrison (1981), Kapandji (1981).

Topological approach, using some form of topology and abstract geometry in robotics, has appeared sporadically in the literature since 1983. In fact, Gottlieb (1986) has speculated that point-set topology and topological invariants may give insight into practical robotic problems such as singularity avoidance. Recently, Baker (1990) has reinforced Gottlieb's speculations by identifying a number of additional robot manipulator problems that could be investigated using a topological approach.

Previously, Schwartz and Shahir (1983) have applied semialgebraic geometry to the *piano movers' problem* in finding a continuous motion from a given initial position to a desired final position of a robot manipulator by considering static properties of real semialgebraic sets. Tannenbaum and Yomdin (1987) have gone beyond static properties of such sets by investigating algebraic morphisms between them. They have defined areas of *bad positions* of the robot manipulators by identifying critical values of certain maps.

An account for applications of topological techniques including, for example, (i) work by Lozano-Perez (1983) on manipulator configuration space, and (ii) work by Hopcroft et al. (1986) on motion in contact (which may be extended to multifingered hands), is given in Schwartz and Shahir (1988). These separate investigations of topology and abstract geometry have not yet been formulated in a unified topological and geometrical view of robotic control problems. Neither have they been applied to the control of multifingered robot hands.

Our work is motivated by (i) the intriguing ease with which humans perform grasping and manipulation activities, (ii) the richness of topology not yet fully explored, and (iii) the limitations of existing grasp models in response to numerous applications of dextrous manipulation in areas such as flexible manufacturing environments, space or underwater exploration, contaminated areas, and other unstructured environments. These applications require the development and deployment of intelligent prehension systems. Such systems should be capable of (i) taking a symbolic task description and translating it into an act of grasping and manipulation described in numeric joint variables, as well as adapting to environmental changes, and (ii) performing dextrous manipulation that involves tip prehension, palm prehension and a combination of both.

The chaper is organized as follows. In section 2, we review a number of representative models of grasps that have been suggested for integrated schemes, and discuss their limitations. In section 3, we present a novel, topological model of dextrous grasps. In section 4, we introduce our topological reasoning scheme based on the model. In section 5, we apply the model and reasoning scheme to the solutions of some basic dextrous problems, and in section 6, discuss the simulation results.

2. MODELS OF GRASPS FOR INTEGRATED CONTROL

At the symbolic level, the models of grasps include: (i) Lyons (1985) 's model as a set of three simple grasps: encompass grasp, precision grasp and lateral grasp, (ii) Iberall (1987) 's model that consists of three categories based on force-opposability: pad, side, and palm oppositions, and (iii) Cutkosky (1989) 's model as a tree-like hierarchy of grasp types which are described in terms of relations between task requirements and object geometry. At the numeric level, the model that has been frequently used is the one devised by Salisbury (1982) which lead to the design of a three-finger Stanford/JPL hand. There are no apparent, simple connections between symbolic and numeric control schemes. The reasons are two-fold: (i) there is no uniform representation of hand postures that links grasp models at symbolic level and those at numeric level, and (ii) little distinction is made between hand posture and hand functionality. Both functionality and posture are described by the same terms e.g. power grip, precision grip, etc. For example, the terms *power* and *precision* (Napier, 1956) have been used in a dynamic as well as in a static sense in the same way that flexion and extension have been used to describe both posture and movement. In reality, the dynamics of grasping produces a particular grip and the static concept indicates the initial/final state of grasping (Landsmeer, 1962). Although not explicitly modeled, Cutkosky (1989) has implied the concept of hand functionality in his description of grasp types. It does not, however, clearly differentiate *what a grip is to perform* from *what it is*. In other words, it does not explicitly differentiate hand functionality from hand posture.

The above models share two other common drawbacks. First, the finite nature of existing discrete grasp models limits the selection of available grasps. Second, a set of large and different combined requirements (from a variety of tasks and different objects) are mapped into the same finite and relatively small set of discrete grip types, as a classification problem, thus leading to the loss of detailed information necessary for numeric control.

Another reason is that symbolic control is task-oriented, while numeric control is mostly contact-based. For an integrated control scheme to be functionally unified, we need a grasp-based model. Furthermore, since motions (forces) at the numeric level are expressed in terms of mappings that are continuous and differentiable (Li and Sastry, 1989), we should also formulate the set of grasps at the symbolic level as a continuous and differentiable set. This dual requirement has led us to the development of a novel, topological model of multifingered hands, as detailed in our previous papers (Nguyen and Stephanou 1989, 1990a). In the following section, we present a computational model of prehensility based on the topological model of multifingered hands.

3. A TOPOLOGICAL MODEL OF MULTIFINGERED HANDS

This section presents a novel, topological model of multifingered hands. The grasp-based, topological model is represented as a collection of topological and geometric spaces described at various levels of detail, with topological transformations and geometric congruences defined between those spaces,

Our *topological* model of multifingered hands (posture and functionality), (Nguyen and Stephanou, 1989, 1990a), is based on two groups of intuitive concepts: (i) **postural concepts** which consist of: a *geometric polyhedron* representing an arbitrary hand posture, and a point-set, *topological polyhedron* bounded by terminal postures of a given hand, representing the set of all possible hand postures, and (ii) **functional concepts** which consist of a concept of *hand subconfiguration* representing an aggregation of fingers to achieve some intended grasp (since not all the digits are always involved in a grasp), and a concept of *contact subconfiguration* representing a collection of topological primitives that are common to both the hand posture and the object (in an act of grasping and manipulation). These concepts lead to the formulation of postulates and definitions crutial to the development of our model of computation.

3.1. Hand posture

Postulate 3.1: Topological representation of a set of all postures. *For a k-finger hand, $k \geq 2$, the set of all hand postures is bounded by four terminal postures, and therefore forms a topological tetrahedron T.*

As a point set, this topological tetrahedron is the highest level of abstraction in all representations of hand postures.

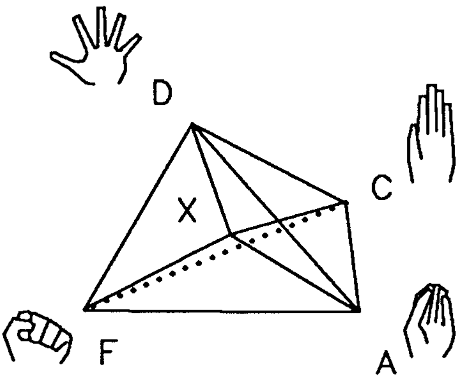

Figure 1: Topological tetrahedron of hand postures

Postulate 3.2: Geometric representation of a hand posture. *An arbitrary k-finger hand posture, $k \geq 2$, is represented as a d-dimensional geometric polyhedron G. A geometric hand posture G is said to be convex if $x \in G$ and $y \in G$ imply $(x,y) \subset G$.*

In general, a geometric polyhedron is not necessarily convex. Since most activities with a hand involve convex postures, and since concave polyhedra may be decomposed into convex ones, we assume that all polyhedral configurations of interest are convex. G can be represented by a set of $(d-1)$ dimensional polyhedra, called simplexes (faces), which are represented by a $(d-2)$-polyhedra (edges), and so on until $d = 0$, or equivalently, a set of vertices of the original d dimensional polyhedron. Topologically, these elements (face, edge, and vertex) are topological primitives called *simplexes* in combinatorial topology (Pontryagyn, 1952).

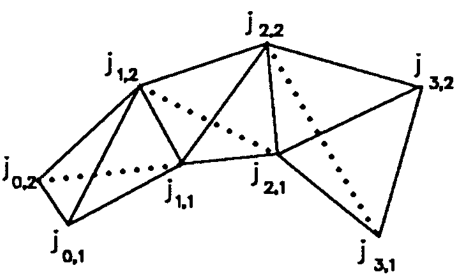

Figure 2: Two-finger hand posture as collection of simplexes

Thus, a hand posture may be decomposed into simpler forms, called *simplexes* of smaller dimensions, which adjoin one another in some describable fashion (i.e. properly-situated). We have shown that (Nguyen and Stephanou, 1989, 1990a), a geometric hand posture is the union of a collection of properly-situated connected sets which are simplexes. Conversely, a hand posture is a geometric polyhedron which can be decomposed into simplexes. The decomposition scheme is called a *geometric complex* K which is, equivalently, a collection of simplexes. The 0-simplexes of a complex K are the joints of the digits, the 1-simplexes are the links, the 2-simplexes are the patches, and the 3-simplexes are the geometric tetrahedra. An example of a two-finger hand is shown in Fig. 2.

In summary, for an arbitrary hand posture, the following representations are equivalent: (i) a geometric polyhedron for the entire hand which is then subdivided into properly-situated geometric tetrahedra, (ii) a collection of properly-situated triangular patches representing the dorsum of the hand, (iii) a set of properly-situated chains representing the fingers, and (iv) a set of vertices representing the joints of the digits.

These simple intuitive concepts give rise to the use of (i) point-set topology techniques for approximating an arbitrary hand posture using barycentric coordinates, and (ii) combinatorial topology techniques, for determining a hand posture at different levels of geometric and topological details (e.g. hand level, finger level, joint level) in Cartesian space and in joint space. In Nguyen and Stephanou (1990a, 1990b), we have shown that lower dimension simplexes can be derived from higher dimension simplexes by the boundary theorems (Pontryagin, 1952).

3.2. Hand functionality

Definition 3.1: Set of hand subconfigurations. *A k-finger hand is represented as a set S of mutually exclusive and exhaustive digit singletons. A subset of S is called a hand subconfiguration. The set of all possible subconfigurations is the power set of S.*

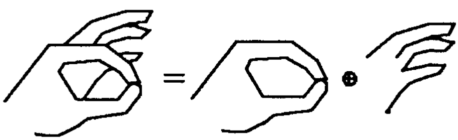

Figure 3: Hand subconfiguration

Since a hand subconfiguration is defined as an aggregation of digits or groups of digits, the simplest subconfiguration has one finger, and the most complete subconfiguration is the entire hand. Conversely, a hand posture may be composed by topological concatenations of two or more subconfigurations. Consequently, a complex task to be performed by an entire hand may be decomposed into simpler tasks to be concurrently performed by subconfigurations in a distributed and coordinated manner.

To characterize hand functionality in terms of contacts, we use the notion of *contact subconfiguration* introduced in Nguyen and Stephanou (1990a). We briefly recall here our discussion on hand subconfiguration, contact configuration and contact subconfiguration.

Definition 3.2: Set of contact subconfigurations. *In grasping, the hand is in contact with the grasped object via a set of contacts, called contact configuration. A contact configuration is then the intersection of two geometric complexes: (i) one complex representing the hand posture, and (ii) the other representing the graspable object.*

In terms of motion, each contact reduces the freedom of motion of the moving object. In terms of force, each contact is described by a mapping between the force exerted by the finger at the contact, and the resultant force and torque at some fixed base. The effect of contact configuration or of any of its subsets (group of contacts) involved in a grip is described in (Salisbury, 1982) as somewhere between a 0-dof and a 6-dof mobility resulting from a set of wrenches (twists) applied at the groups of contacts.

We call a subset of all contacts that produce the same effect (e.g. forces of same direction and amplitude) a *contact subconfiguration.* In other words, a contact subconfiguration is a subset of contacts that are *functionally equivalent.* For example, a lateral pinch which consists of a thumb in contact with the object, and a set of remaining four fingers (of a human hand) in planar convergent posture (Nguyen and Stephanou, 1989) has two contact subconfigurations: (i) the first is the one produced by the thumb, which has a relatively small contact area, and (ii) the other is the group of contacts produced by the four fingers, which has a comparatively large contact area. Thus, any given contact configuration can be decomposed into functionally equivalent contact subconfigurations. To be functionally equivalent, each individual contact of the contact subconfiguration must be of the same nature (friction, frictionless), and of the same type (point, line, surface, soft finger).

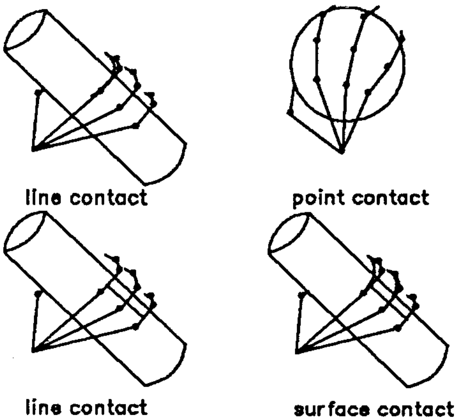

Figure 4: Contact subconfigurations

Consider a cupped hand posture shown in Fig. 5. The same hand posture may be used in three different handling schemes: (i) as a thumb-index precision grip in the left figure, (ii) as a two-finger precision grip in the middle figure, and (iii) as a palm-finger power grip in the right figure.

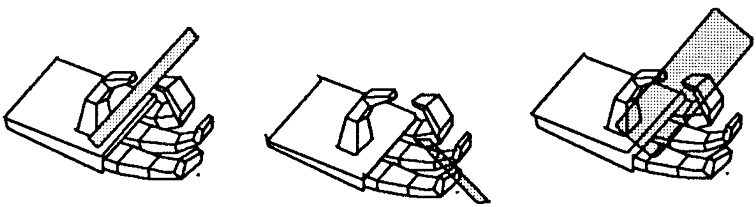

Figure 5: Posture and functionality

Although the hand postures are the same in three grasps, their hand functionality differs greatly. The difference between them are expressed in terms of hand subconfigurations, contact configurations and contact subconfigurations involved. Since contact configurations are actually the intersections of two geometric complexes, one representing a given hand posture, and the other representing the given object, a contact configuration is unique for a given grasp. Furthermore, each constituent contact subconfiguration can be described as a collection of simplexes (point, line, surface).

Our representation of a contact between a hand and a grasped object is a modified representation of Laugier (1989), i.e. $CONTACT(H,O) = [((s_1, t_1, type1,), \ldots ,(s_n, t_n, typeN))]$, where s_i and t_i are the intersections between the simplexes of a hand posture (that are in contact with the object) and those representing the faces, edges or vertices of the grasped object, and *type* indicates a point contact, line contact or surface contact. Thus, a contact subconfiguration is a subset of $CONTACT(H,O)$. This modified representation has several advantages:

1. it can represent contacts in the sense of Salisbury (1982) (point, line and surface contacts) with attributes describing the contact coordinate frames, or other pertinent information about contacts,
2. it can represent large area contacts such as those between the palm (and/or a group of digits) and the object, since it is the intersection of simplexes of the hand posture (patches, chains, joints) and the simplexes of the object (foaces, edges, vertices).
3. it takes advantage of numerous existing algorithms dealing with the intersection of convex bodies for the determination of actual contact configurations,
4. it may be robustly used in a reasoning about *anticipated* contacts for hand preshaping or grasp planning.

We introduce the following definitions:

Definition 3.3: A **functionality pair** *is a pair of contact subconfigurations aimed at either (i) maintaining an equilibrium* ($\sum forces = 0, \sum moments = 0$), *(ii) stability (ability to resist disturbances) of a grasp object, or (iii) allowing it to have some mobility (e.g. ability to rotate) in some predetermined directions.*

Definition 3.4: A **prehensility pair** *is a pair of hand components and object (palm, digits, object parts) that are in a contact subconfiguration. It is called type-I when the contact subconfiguration is a point contact, type-II when it is a line contact, and type-III when it is a surface contact*

The notion of *functionality pair* may be thought of as a generalization of the notion of kinematic pair of Reuleaux (1963). The notion of *prehensility pair* may be thought of as a generalization of the notion of prehensility of Napier (1956). Thus functionality pair denotes *what should occur* and prehensility pair denotes *how it occurs.* We show in the following paragraphs that the functionality pairs and prehensility pairs are necessary and sufficient for characterizing hand functionality and hand postures.

At the numeric level, contacts are used to classify kinematic pairs, into *lower pairs or higher pairs* by Reuleaux (1963), i.e. pairs of rigid bodies which constrain each other's motion. In a lower pair, the contact between the two elements of the pair occurs continuously at all points on the surface of the contact which must therefore have the same form in both elements. In a higher pair, contact between the elements takes places along lines or points of contact (Gelsey, 1987). When the hand is in a power grip with the object, it is generally true that the hand contacts the grasped object over a large surface due to the presence of the palm. When the hand is in a precision grip, the contacts are more of a fingertip type. We associate the type of contact between hand and object to a functionality pair: for large surface contact area (specified by the *size* of contact), we have a power-functionality pair (or power-pair for short) and for small contact area, we have a precision-functionality pair (or precision-pair). In other words, the type of functionality pair is determined by the size of the constituent contact subconfigurations.

Characterizing intended contacts in terms of functionality pairs is not sufficient for the determination of grip types. Indeed, in a support grip, both types of functionality pairs exist: power-pair in platform grip and precision-pair in hook grip. Napier's notions of prehensility and opposability and the derived notion of force-opposition by Iberall (1987) may be extended in these cases. When force opposition is between between digits or between digits and palm, (prehensility pair type I or II), then we are dealing with functionality pairs for power and precision patterns. When force-opposition is between the hand (digits or palm) and the object (prehensility pair type III), then we are dealing with supportability. Thus force-opposability may be considered as a special case of *prehensility pair*. Different types of prehensility pairs have been introduced in Nguyen and Stephanou (1990a). Except for the case of terminal postures where the postures and functionalities are clearly indicated, (i) in a power grip, a power-pair is required, (ii) in a precision grip, a precision-pair is required, and finally (iii) in a support grip, either a power pair or a precision pair is applicable, regardless of the posture involved (power, precision or support-oriented),

The following additional examples illustrate the different types of functionalities of a grip.

- A hammer grip indicates both a *power posture* (e.g a fist configuration) and a *power-functionality pair* (e.g. the act of squeezing by the digits and the opposing palm, digits and palm are two elements of the pair).
- A hook grip consists of a *power posture* and a *precision-functionality pair* (e.g the act of carrying a heavy suitcase by a set of flexed fingers: the suitcase and the groups of fingers are two elements of the precision pair).

- A platform grip consists of a *support posture* (e.g. flat hand) and a *power-functionality pair* (e.g. the act of supporting a large plate).
- A tripod grip similarly consists of a *precision posture* and a *precision-functionality pair* (.e.g. thumb and two fingers placed in opposition).
- A thumb-index pinch seen as compression by the thumb and the index in tip-to-tip, pad-to-pad or pad-to-side configurations consists of a *precision posture* and a combination of *power-functionality pair* and *precision-functionality pair.*

In summary, we state that:

- *Hand functionality* is subdivided into: (i) *prehensile functionality* i.e. the ability to perform a grasping or manipulative task, and (ii) *quasi-prehensile functionality* i.e. the ability to perform a support function (such as in hook grasp or platform grasp) by a given multifingered hand on a given object according to some given task specifications.
- All (prehensile and quasi-prehensile) task functionalities fall into one of three types: (i) supportability, (ii) graspability, and (iii) manipulability. The corresponding hand functionalities are (i) support (e.g. in a platform grip or hook grip), (ii) power (e.g. in a palmar grip), and (ii) precision (e.g. in a two-finger pinch). In an act performed by the hand onto an object, these functionalities result in three types of prehensility, respectively: (i) palm-prehensility, (ii) tip-prehensility, and (iii) quasi-prehensility.
- Task functionality may be transformed into prehensility and opposability, and hand functionality can be described in terms of pairs: functionality pairs and prehensility pairs, which relate to opposability and prehensility at the symbolic level, and to kinematic pairs at the numeric level. The basis for hand functionality determination is a topological reasoning about contact configurations detailed in Nguyen and Stephanou (1990b).

4. TOPOLOGICAL REASONING ABOUT DEXTROUS GRASPS

Cutkosky (1989) has argued that it is the task requirements and object geometry combined that determine the required grip. More specifically, he has derived the set of feasible grips from the intersection of three sets of constraints arising from: (i) the task (e.g. forces and motions that must be imparted) (ii) the grasped object (e.g. object shape) and (iii) the hand (e.g. maximum grasping forces and maximum finger opening). In other words, Cutkosky's reasoning process involves (i) a reasoning about tasks, (ii) a reasoning about objects, and (iii) a reasoning about grasps, in the determination of a suitable grasp. This general formulation has also been used by other authors on their grasp models (e.g. Iberall 1987).

Our three-step approach summarized below appears, on the surface, to be similar to Cutkosky's approach, however there is a basic difference: the reasoning about grasps is replaced by a reasoning about contacts. Furthermore, there are three additional differences between our underlying reasoning scheme, called *topological reasoning*, and others: (i) it is based on topological properties of tasks, objects and grasps, (ii) it avoids the cross-product operation by mapping these properties into the topological contact domain (step 3), and (iii) most importantly, the underlying algorithms and knowledge bases are developed based on the topological model (in contrast with ad hoc rules).

- Step 1: Reasoning about tasks.
  a. Task constraint analysis.
- Step 2: Reasoning about objects.
  b. Graspable configuration analysis.
- Step 3: Reasoning about contacts.
  c. Determination of contact configuration.
  d. Selection of contact subconfiguration.

Basically, topological reasoning consists of a set of algorithms used to derive a suitable grip given a symbolic task description and a graspable object. The derived grip is described in terms of posture and functionality (Nguyen and Stephanou, 1989, 1990a). The set of algorithms (i) can derive hand posture for palm prehension or complex combinations of palm/tip prehension, (ii) reduces the complexity and size of the inverse kinematics solution space, if inverse kinematics is used (as in the common cases of tip prehension, where a hand posture is subdivided into one, two, or a maximum number of three-finger hand subconfigurations), and (iii) guides the development of rules representing the necessary prehensility knowledge.

We consider our topological reasoning as an extension of *geometric reasoning*, the latter being a form of representation and reasoning about geometry (Kapur and Mundy, 1988). Current algorithms for robotic applications in general, or for multfingered hands in particular largely rely on geometric concepts. We have shown in earlier papers that these geometric concepts lead to the topological concepts introduced in our topological model. The topological model also helps reduce difficulties in the acquisition of prehensility knowledge.

Since the three (topological) reasoning processes (i.e. reasoning about tasks, objects, and grasps using their topological properties) involve contact (anticipated or actual, or contact avoidance), the key element of topological reasoning about hand posture and hand functionality is *reasoning about contact subconfigurations.* Reasoning about contacts is *topological* for the simple reason that contact configuration is defined as the intersection of (topological) hand posture simplexes (i.e. vertices, chains, patches) and (topological) object simplexes (i.e. vertices, edges and faces). This reasoning about contacts is simply a scheme for finding and describing these intersections (which are themselves simplexes), given a symbolic task.

4.1. Postural transformations & topological reasoning about hand postures

In Nguyen and Stephanou (1990a, 1990b), we have presented a group of topological algorithms for (i) a polyhedral approximation of a power grasp (palm prehension), and (ii) a barycentric approximation of a precision grasp (tip prehension). The details of those algorithms may be found in Nguyen and Stephanou (1990a, 1990b).

- Step 4: Determination of hand subconfiguration posture.
  - e. Approximation of barycentric coordinates (tip prehension)
  - f. Polyhedral approximation of a hand posture (palm prehension)

The output of this set of algorithms is a *geometric polyhedron*, with associated *barycentric coordinates*, describing the hand posture to achieve a set of contacts on the graspable objects as dictated by some basic functional requirements of the task.

The above set of algorithms is *posture*-oriented, i.e. it has not taken into account explicitly the functional aspects of a grasp. In the next section, we present a topological reasoning scheme for the determination of hand functionality.

4.2. Functional transformations & topological reasoning about hand functionality

As discussed previously, the combined task-object functionality may be analyzed in terms of functionality pairs, and prehensility pairs. In other words, the problem of determination of hand functionality becomes the problem of determination of possible functionality pairs and prehensility pairs as required by the task to be performed on the object. Furthermore, as explained in section 3.2, functionality pair and prehensility pair are notions parallel to (i) Reuleaux's notions of kinematic pairs, and (ii) Napier's notions of prehensility and opposability. This parallelism plays a key element in the derivation of low-level, numeric kinematic pairs from hand functionality and task functionality.

Commonly, in the problem of hand functionality determination, ambiguity occurs when the functional characteristic (power, precision or support) of the grip is not clearly indicated, or when there is more than one functionality associated with a posture. Even in the case where the posture clearly indicates power, precision or support patterns, the hand functionality may not be unique. Indeed, when a hand assumes a certain posture, there exist a number of tasks that it may handle with that posture. For example, a flat hand may perform a support function or a push/pull function with the palm, a cutting-like function with the edge of the palm, a squeeze function between the edges of the fingers, etc. Conversely, given a certain task, there exists a multiplicity of grips that can be used to perform the same task depending on the purposeful utilization of the object. Similarly, one may hold the same object in different ways depending on the task requirements: with a flat hand (object on the horizontal palm), with a power grip (palm and fingers wrapping around the object), with a precision grip (five fingers in fingertip contact with the object). We recall here the following algorithms for the determination of hand functionality in an act of grasping and in regrasping (change of grasp). These algorithms have been detailed in Nguyen and Stephanou (1990b).

- <u>Step 5: Determination of hand subconfiguration functionality.</u>
  - g. Barycentric subdivision of subconfiguration space
  - h. Detection of functionality pair and prehensility pair
  - i. Mapping of functionality/prehensility pairs into contact space

5. APPLICATIONS TO DEXTROUS GRASP SYNTHESIS

Dextrous manipulation required in a task may be viewed as a repeated sequence of grasping and regrasping acts. In grasping, a hand is in contact with a graspable object via a set of contacts which constitutes a contact configuration. Regrasping may be roughly defined as a process involving a change of grasp. Regrasping is necessary particularly when the environment is unstructured, initial grasps generated (by using some intelligent scheme) need to be refined after the first contact, the locations of contacts are easily disturbed due to slippage or rolling, or the objects themselves are not rigid.

Regrasping consists of change of position (re-positioning), or change of force (force adjustments) with or without a change of position.

At the symbolic level, the problem of grasping has been commonly investigated as a problem of grasp selection (Cutkosky, 1989) based on task functionality and object geometry. The problem of regrasping (change of grasp) as the result of manipulation has been modestly addressed. At the numeric level, dextrous manipulation has been analyzed in terms of motion (twist systems), or in terms of forces (wrench systems) imparted to the object via contacts. There is, however, no systematic procedure on how to determine these screw systems from a symbolic task description.

In this chapter, we are concerned with how the computational model is used in the general framework of an *intelligent robot prehension* scheme. The scheme is called *prehension* scheme because it deals with three basic elements of prehension (Harrison, 1981): an intent (the task), perception, and the mechanism of grasping and manipulation. The form of perception discussed here is called *perception by memory*, a term coined by psychologists (Yeap, 1988) to indicate a recall of perceived things previously stored in memory. We implement the concept of perception by memory through the use of prototypes (task, object and grasp prototypes). The *intelligent* aspect is associated with the topological reasoning algorithms discussed in section 4. Whether it involves a grasping or a regrasping task, dextrous robot hand activity then consists of two processes: (i) a postural transformation for the derivation of hand posture (configuration or subconfiguration), and (ii) a functional transformation for the derivation of hand functionality to achieve a set of intended contact configurations.

Since these transformations have been discussed in details elsewhere (Nguyen and Stephanou, 1990a, 1990b), we restrict our discussion to their implementational characteristics.

The basic requirement is that our system accept symbolic task descriptions to produce joint-level parameters for controlling dextrous manipulation. Internally, representational requirements include: (i) topological representations for task and object descriptions, hand postures, contact configurations and prehensility knowledge, and (ii) a common structure for capturing and accessing topological, geometrical, functional and behavioral properties of prototypes.

There are three types of prototypes: task prototypes, objects prototypes and hand subconfiguration prototypes. These prototypes are considered as typical situations or events that describe classes of tasks, objects and hand shapes whose elements are treated more or less equivalently. All prototypes have attributes that describe their structure (topological and geometrical), function, and behavior. A structural description of a prototype consists of (i) the individual components that characterize it, and (ii) their interconnections. A functional description reveals the purpose of the structural component or connection in producing the behavior expected from the task execution. A behavioral description describes the potential events that may occur. The more detailed the description of these prototypes, the better the basis for discrimination/similarity between a given instance I (of task, object, and/or hand posture) and a prototype P.

A high-level task is a task expressed in general terms such as *build, assemble*, etc. Such task may require dextrous manipulation. A high-level prehensile task may be decomposed into subtasks, e.g. *grasp, screw*. For each subtask, we consider two types of tasks: (i) those that produce no motion to the object, or *type-I* tasks, and (ii) those that impart motion to it, or *type-II* tasks. One example of type-I tasks is a *holding* task. Type-II tasks include *shaking, twisting, turning, moving*, etc. Thus, type-II tasks may follow and/or coexist with a type-I. A low-level or primitive task is one expressed in terms of forces and/or primitive motions (e.g. translation, rotation). Type-I and type-II tasks may also be decomposed into low-level tasks. In general, there is no unique way to arrange the typed-tasks in sequence. In other words, there are many ways to perform a high-level task.

From the above discussion, we use three levels of task specifications:

1. high-level (abstract) task such as *build, assemble*;
2. subtask (typed-task) such as *preshape, enclose, hold*;
3. primitive tasks such as *translation, rotation.*

These levels form a tree structure associated with the high-level task. Task functionality may be defined as a set of specifications that describe *what to do.* Just as a high-level task is commonly decomposed into smaller tasks, task functionality of the high-level task may be decomposed into a collection of functionalities of subtask (typed-task) and primitive tasks at each of the nodes of the task tree-structure.

To describe a high-level task, we introduce here the notion of a *task map.* Initially, the task map is in the form of a general skeleton (list of subtasks or a tree of subtasks). A *raw task map* is a task map that is initialized. The raw map is filled in with the forementioned functional details and constraints. The raw map then takes the form of a *full task map*. The structure of both the raw map and the full map of each primitive task are the same: all the characteristics are organized in four categories: (i) geometrical, (ii) topological, (iii) functional, and (iv) behavioral. Each primitive task is described by this list of attributes.

The full map describes the *what to do*, not the *how to do it* associated with a task. The *what to do* is unique for a given typed-task or primitive task, but the *how to do* is not. For example, in a type-I task such as *hold* there is one unique functional requirement, i.e. stability, regarless of *how to hold*. Although there is more than one possible way of *how to hold* a given object, all such possibilibities satisfy a single functional requirement: equilibrium. The *how to do* is derived from a reasoning process described in section 4, using the algorithms detailed in Nguyen and Stephanou (1990a, 1990b).

Task prototypes describe generic tasks. A task prototype is described by a task map memtioned earlier. A *full task map* is a full-blown map that contains attributes grouped into four categories: topological, geometrical, functional and behavioral. These categories are needed for the topological reasoning scheme described in section 4. The subprocess of filling out detailed information in a task map from task specifications is called *reasoning about tasks*.

Similarly, we assume the existence of *object prototypes* in this system. To model the object in its workspace such that its topology and geometry are readily available, the boundary representation (BR) method is appropriate. The BR contains both topological and geometric information. The topological information describes the connectivity between vertices, edges, and faces of the object. The geometrical information includes vertex coordinates, and equations for edges and faces. Transformation matrices are attached to these elements for the computation of object locations with respect to a fixed base (Stevenson, 1987). When regular objects are in BR format, more complex objects may be formed from these regular objects by using a constructive solid geometry (CSG) representation in a hydrid CSG/BR representation (Requicha and Voelcker 1985, Stevenson 1987). The subprocess of filling out detailed information in a full task map from object specification is called *reasoning about objects*.

The BR representation is also suitable for a hand posture since it is viewed as a 3D geometric polyhedron, which is a collection of connected geometric tetrahedra as discussed in section 3. Furthermore, the topological point-set of all hand postures is represented as a tetrahedron. As a result, both topological and geometrical representations of the hand (the entire set or an individual configuration) may use the same data structure. The sub-process of using detailed information in a full task map to derive hand postures is called *reasoning about hand postures.*

A functional block diagram is given below.

Block I in Fig. 6 transforms a symbolic prehensile task description and graspable object specifications into a list of task-object attributes. The attributes describe the topological and geometrical structure, behavioral and functional characteristics of the task and of the object. Block II in Fig. 6 consists of (i) a composite mapping which derives and extracts two sets of grip attributes: posture-oriented grip attributes (i.e. what is the required hand shape), and functionality-oriented grip attributes (i.e. what to do with the hand shape), (ii) a posture mapping for processing posture-oriented attributes, and (iii) a functionality mapping for processing functionality-oriented attributes. In block III, there are two processes: (i) geometric transformation, and (ii) contact mapping. These processes together perform geometric reasoning about, and the joint-space computation of, hand postures. Functionality mapping, contact mapping, topological and geometric mapping have been outlined in section 4, and detailed in Nguyen and Stephanou (1990a, 1990b).

Our topological reasoning scheme consists of processes in blocks I and II of the functional block diagram. The numerical computation consists of processes in block III (Fig. 6).

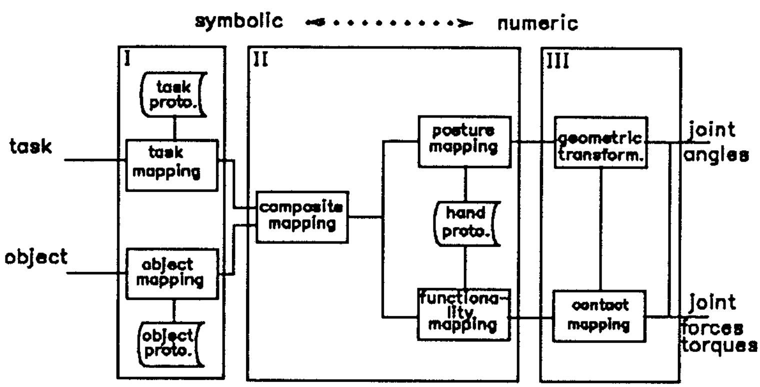

Figure 6: Functional block diagram

In the following, we briefly discuss the applicability of an intelligent robot prehension scheme to areas such as agricultural harvesting (e.g. fruit picking) or hazardous environement, space repair. Such robotic applications differ much from common applications found in manufacturing assembly, where the environment is totally unstructured, graspable objects are different in size (although they may be of the same shape), their locations are easily disturbed (although stable) upon contacts, and the objects themselves are not rigid objects (requiring small, distributed forces to be exerted via multiple contacts). The task of fruit picking, i.e. a combination of grasping the fruit, and disconnecting it from its stem by either pulling, breaking or otherwise, along with different accesses to the fruit (e.g. from the bottom or from any sideway) requires higher level of dexterity than those provided by grippers and special-purpose end-effectors. Other requirements include for example higher adaptability and compliance, ease of regrasping, ability to apply small forces and fine motions.

If the picking task (for example, picking oranges) involves natural hands, one may expect that a different grip (number of fingers, hand posture and orientation) is used for each picking, depending upon the size, the location of the orange and its surroundings. But most likely, an orange is picked by one of the following basic patterns: (i) holding the orange by a precision grip and slightly pulling along or turning the hand about its axis to disconnect it from the stem, or (ii) holding it in a power (spherical) grip and breaking it from the stem by shaking the hand back and forth, or (iii) holding the orange with the palm and possibly placing the thumb and the index finger on the stem and pulling its from the connected branch. Although the described basic patterns seem to be simple enough that few generic grasp types are sufficient, one may notice that these patterns differ largely in their functionalities, for example precision versus power (in the sense of Napier, 1956), and that at grasp execution time, adjustments or changes in grasps may be needed, for example, a change from one grip (e.g. precision) to another (e.g. power) when the first one fails. The anticipated grip also provides the ability to hold the orange stably by slightly adjusting the small forces exerted via contacts with the orange and/or by changing the grip while holding the orange, according to some changed external force requirements (e.g. gravity force). Furthermore, a multifingered hand has a definite advantage over a gripper in that, in any picking mode, (i.e. power or precision), the fingers are sort of wrapping around the orange as a container (or frame), thus allowing the orange to fall into the interior volumeric space formed by the palm and the fingers, and therefore prevent the orange from slipping out of the hand.

6. SIMULATION RESULTS AND DISCUSSIONS

6.1. Task and object input specifications

The task selected for this study is a HOLD task. Three task requirements are used: *stability, force closure, and connectivity.* Stability means the ability of the hand to resist external disturbances, i.e. to keep the object in stable equilibrium. Force closure (Cutkosky, 1989), indicates the conditions to be satisfied by the forces/moments applied via the anticipated contacts without breaking the contacts, and finally, connectivity (Salisbury, 1982) indicates the number of task degree-of-freedom's of the object relative to the hand. A *high* stability, *low* force-closure, and *low* connectivity *hold* task on a graspable object intuitively implies a *zero-dof task mobility and stable* hold.

The regular object selected is of cylindrical shape. For such a shape, the topological simplexes of the object consists of two edges, one cylinder surface, and two circular disks.

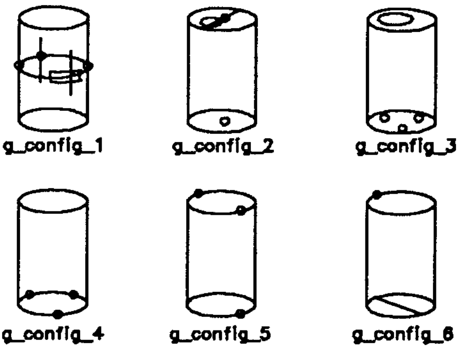

Figure 7: Graspable configurations

The set of all possible graspable configurations include (Nguyen and Stephanou, 1990b): (i) cylindrical graspable configurations (g_config_1), (ii) circular graspable configurations (g_config_2, and g_config_3) and (iii) graspable configurations by the edges (g_config_4, g_config_5, g_config_6). Each graspable configurations may involve any combination of the following three types of contacts: point, line, surface, i.e. point-to-point, point-to-line, point-to-surface, line-to-line, line-to-surface, and surface-to-surface contact. For a grasp, the types of contact are defined as follows. (Note that our definitions of contact types are for the whole hand, and are different from contact types defined and modeled by Salisbury (1985). Ours may be considered as macroscopic definitions with respect to Salisbury's definitions. Our definitions complement those defined by Salisbury, and are necessary for palm prehension, or palm/tip prehension).

We have a *point contact* when the contact is between a fingertip and the object. We have a *line contact* when (i) multiple contacts occur between a finger and the object (on successive finger links), or (ii) a set of single contacts, each of which is between a finger link and the object. We have a *surface contact* when there are more than one line contact between the hand and the object. A precision grasp, in the sense of Napier (1956), most likely occurred in tip prehension, involves a set of point contacts by the fingertips. A power grasp or support grasp, most likely occurring in palm prehension, involves then a combination of point, line and surface contacts. The determination of graspable configuration of a given cylindrical object then involves the selection of one of the multiplicity of available combinations of anticipated contact configurations, out of the six different graspable configurations described above.

Topological reasoning for the determination of a suitable grasp (posture and functionality) consists of a systematic execution of the algorithms introduced in section 4. In this section, we report and evaluate the results of our case study on grasp selection. The grasp selection involves the following process:

1. reasoning about task (HOLD),
2. reasoning about object (CYLINDER),
3. reasoning about anticipated contact configurations, and
4. determination of hand subconfiguration posture and functionality.

6.2. Simulation results

The following results were obtained for the cases below.

CASE 1: POWER GRASP

```
Inputs:
  Task constraints:  High Stability,
                     Low Force Closure,
                     Low Connectivity.
  Object dimensions: Diameter of cylinder: 0.84
                     Height of cylinder:   0.95
  Finger link length: 0.25
Outputs:
  Recommended grasp: THUMB_INDEX SUBCONFIGURATION,
                     Line contact,
                     Fingers have multiple contacts,
                       Palm-prehensility type I,
                       Power pair,
                       Line contact (thumb),
                       Line contact (index),
                       Graspable configuration 1 (cylindrical)
                       Hand opening larger than .84.
  Hand posture (polyhedral approximation):
    Number of iterations: 3.
      First iteration   : 0.97
      Second iteration  : 0.58
      Third iteration   : 0.24
```

Interpretation of input specifications, results and discussion

The object is smaller than the hand size (its diameter and height are normalized with respect to hand size, and are both less than 1). The recommended subconfiguration is a thumb-index grasp, the overall posture is of type I palm prehensility (Nguyen and Stephanou, 1990a), and the overall functionality is *power pair*. The grasp should occur along the cylindrical face of the object, with a hand opening between the thumb and the index larger than .84, the size of the cylinder's diameter. Furthermore, the anticipated contacts should be of line contact type, while each finger is in line contact with the object. This is a power grasp in the sense of Napier (1956).

The approximation of hand posture using polyhedral approximation technique required three iterations. In the first iteration, the edge of the hexagon covering the circular cross-section is computed and is equal to .97. This edge is much larger than the finger link which is .25 (the finger has four links and each link is then $1/4 = .25$). On the second iteration, the hexagon becomes a octagon with edge length equals to .58. Therefore another iteration is necessary. The decagon has an edge length that is equal to .24 which is smaller than the finger link, i.e. The joint angles are the angles between adjacent edges of the decagon.

The following rules have been fired:

CASE 1: EXPLANATIONS:

```
Statement: If (task = HOLD) then dof = 0;
Statement: Switch (Stability)
           Case (high):
            Switch (Force_closure)
             Case (low):
              Switch (Connectivity):
                Case (low):
                Sub_config  = thumb_2_fingers;
                Contact     = line;
                Conflict    = Force_closure;
Statement: If Stability(high) and Force_closure(low)
                and Contact(point)
              then Contact(line);
Statement: If Force_closure(low) and Connectivity(low)
              then substract_sub_config();

Statement: If (obj_diam < 1 and obj_hi < 1)
              then
              switch (Stability)
                   case (high):
                   switch (Connectivity)
                        case (low):
                        g_config_1;
Statement: If (thumb-index and contact(line))
              then
              line(thumb);
              line(index);
              power_pair;
```

We illustrated the results of a precision grasp and a support grasp without further explanations.

CASE 2: PRECISION GRASP

```
Inputs:
 Task constraints:  Medium Stability,
                    Medium Force Closure,
                    High Connectivity.
 Object dimensions: Diameter of cylinder: 0.81
                    Height of cylinder:   0.90
Outputs:
 Recommended grasp: THUMB_INDEX SUNCONFIGURATION,
                    Point-to-point contact
                    Fingers have single contacts,
                    Tip-prehensility type II,
                    Precision pair,
                    Point contact (thumb),
                    Point contact (index),
                    Graspable configuration 1,
                    Hand opening larger than .81.
```

CASE 3: SUPPORT GRASP

```
Inputs:
 Task constraints:  Medium Stability,
                    Medium Force Closure,
                    Low Connectivity.
 Object dimensions: Diameter of cylinder: 1.22
                    Height of cylinder:   2.10
Outputs:
 Recommended grasp: THUMB_AND_THREE_FINGERS,
                    Line_to_Surface contact
                    Fingers have multiple contacts,
                    Supportability type I,
                    Power pair,
                    Line contact (thumb),
                    Surface contact (finger),
                    Graspable configuration 1.
```

7. CONCLUDING REMARKS

The topological model described in section 3 consists of a structural (hand posture) model of multifingered hands, and a functional model of hand functionality. Collectively, they constitute a *computational model for multi-fingered robot prehension.*

We have detailed a reasoning procedure for deriving hand posture and hand functionality from (i) symbolic task specifications described by a task map, and translated into contact configuration specifications, and (ii) object specifications described in terms of graspable configurations (also translated into contact configurations). Our reasoning scheme avoids the common approach to grasp synthesis which relies on the cross-product of task attributes and object attributes.

We have introduced the notion of contact subconfigurations as a group of contacts, and as a subset of contact configurations, thus (i) describing a contact configuration as a hierarchy of contacts (i.e. configurations, sub-configurations, contacts), and (ii) facilitating the description of power-oriented grasps, support grasps, and precision-grasp. Again, all contacts are described in topological terms.

Describing a grasp in terms of contact configurations, contact subconfigurations, and individual contacts facilitates the determination of grasps at the numeric level, which is contact-based. Describing a hand functionality in terms of functionality pairs (task-oriented) and prehensility pairs (object-relevant) facilitates the description of the task and the object as a collection of grasp-relevant attributes.

The overall design of an intelligent robot prehension scheme has been described. The structural and functional design aspects are basically directed by the topological model of prehension. The design serves as a prototype system for further investigation of more efficient prehension algorithms. The data structure introduced here is uniform in that the same structure is applicable for representing objects, hands, and contacts between them. The classification of data into four categories, namely topological, geometrical, functional and behavioral forces designers to think about the nature of each attribute and to accurately specify the data in each category. It also guides the formulation of rules for the processing of categorized facts, helps to analyze prehensile tasks for capturing prehensility knowledge, provides a natural way to link to geometric reasoning. Future work includes an extension to a sensor-based control scheme that gives rise to the application of *evidential reasoning* to dextrous manipulation.

8. REFERENCES

1. Arbib M. A, Iberall T. and Lyons D. (1983), *Coordinated control program for movements of the hand,* Coins Technical Report, University of Massachusetts.
2. Baker D. R. (1990), "Some topological problems in robotics", in *The Mathematical Intelligencer*, Vol. 12, No. 1, pp. 66-76.
3. Cutkosky M. R. and Wright P. K. (1986), "Manufacturing grips and correlation with the design of robotic hands", *Proc. IEEE Int. Conf. Robotics and Automation*, pp.1533-1539.
4. Cutkosky M. R. (1989), "On grasp choice, grasp models and the design of hands for manufacturing tasks", *IEEE Trans. Robotics and Automation,* Vol. 5, No. 3.
5. Gelsey A. (1987), "Automated reasoning about machine geometry and kinematics", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 182-187.
6. Gottlieb D. H. (1986), "Robots and topology", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 1689-1691.
7. Harrison S. H. (1981), "The functional relationship of the thumb to the fingers", in *The Hand*, Tubiana R. (ed.), Vol. 1, Sanders, pp. 481-487.
8. Hopcroft J. and Wilfong G. (1986), "Motion of objects in contact", *Int. J. Robotics Research*, Vol. 4, No. 4, pp. 32-46.
9. Iberall T. (1987), "The nature of human prehension: Three dextrous hands in one", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 396-401.
10. Kapandji I. A. (1981), "Biomechanics of the thumb", in *The Hand*, Tubiana R. (ed.), Sanders, pp.404-422.
11. Kapur D. and Mundy J. L. (1988), *Geometric reasoning and artificial intelligence* , Artificial Intelligence, vol. 37, pp.1-11.
12. Landsmeer J. M. (1962), "Power grip and precision handling", *Ann. Rheum. Deceases*, Vol. 22, pp. 164-170.
13. Laugier C. (1989), "Planning fine motion strategies by reasoning in contact space", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 653-659.
14. Li Z. and Sastry S (1987), "Task-oriented optimum grasping by multifingered robot hands", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 389-394.
15. Lozano-Perez T. (1983), "Spacial planning: A configuration space approach," *IEEE Trans. Computers,* Vol. 32, No. 3, pp. 108-120.
16. Lyons D. M. (1985), "A simple set of grasps for a dextrous hand", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 588-594.
17. Napier J. R. (1956), "The prehensile movement of the human hand", *J. Bone Joint Surgery*, Vol. 38B, pp. 902-913.
18. Nguyen T. N. and Stephanou H. E. (1989), "A topological model of multifingered prehension", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 446-451.

19. Nguyen T. N. and Stephanou H. E. (1990a), "A topological algorithm for continuous grasp planning", *Proc. IEEE Int. Conf. Robotics and Automation.*
20. Nguyen T. N. and Stephanou H. E. (1990b), "Distributed intelligent control of multifingered robot hands", *Proc. ISCIE Japan-USA Symposium on Flexible Automation*, Kyoto, Japan, pp. 1099-1106.
21. Pontryagin L. S., (1952), *Foundations of Combinatorial Topology*, Graylock Press, 1952.
22. Requicha A. A. G. and Voelcker H. B. (1985), "Boolean operations in solid modeling: boundary evaluation and merging algorithms", *Proc. of the IEEE,* Vol. 73, No. 1, pp. 30-44.
23. Reuleaux F. (1963), *The kinematics of machinery*, Dover Pubs.
24. Salisbury J. K. (1982), *Kinematic and force analysis of articulated hands*, PhD thesis, Stanford University, Dept. Mech. Engineering.
25. Schwartz J. T. and Shahir M. (1983), "On the piano movers problem II: General techniques for computing topological properties of real algebraic manifolds", in *Advanced Applied Mathematics*, Vol. 4, pp. 298-351.
26. Schwartz J. T. and Shahir M. (1988), "A survey of motion planning and related geometric algorithms", in *Artificial Intelligence*, Vol. 37, pp. 157-169.
27. Stevenson C. N. (1987), "Model-based programming and control of robot manipulators", *IEEE Computer,* August, pp. 76-84.
28. Tannenbaum A. and Yomdin Y. (1987), "Robotic manipulators and the geometry of real semialgebraic sets", *IEEE J. Robotics and Automation*, Vol. RA-3, No. 4, pp. 83-89.
29. Tomovic R., Bekey G. A. and Karplus W. J. (1987), "A strategy for grasp synthesis with multifingered robot hands", *Proc. IEEE Int. Conf. Robotics and Automation*, pp. 83-89.
30. Tubiana R. (1981), "The architecture and function of the hand", in *The Hand*, Tubiana R. (ed.), Vol. 1, Sanders, pp. 19-93.
31. Yeap W. K. (1988), "Towards a computational of cognitive maps", *Artificial Intelligence,* Vol. 34, pp. 297-360.