Topological reasoning about dextrous grasps

by T. N. Nguyen
H. E. Stephanou

The need, in robot manipulation, for higher levels of dexterity and versatility than those provided by grippers and by special-purpose end-effectors has prompted much research effort during the last decade on the design and control of multifingered hands. Most work on multifingered robot hands has dealt with low-level, numeric control, commonly based on screw theory and tools drawn from line geometry, differential geometry, kinematics, and dynamics. Current numeric, contact-based schemes, however, are limited to tip prehension (intentional grasping by the fingertips). The intriguing ease with which humans perform grasping and manipulation activities has concurrently triggered new investigations to provide robots with humanlike, prehensile capability for complex tasks in unstructured environments. These investigations have resulted in numerous AI-oriented, task-directed, distributed, symbolic schemes that have been conducted essentially independently. Efforts to link symbolic and numeric schemes have been undertaken, but the results have been rather modest. This paper deals with an intelligent, integrated symbolic-numeric scheme for dextrous manipulation, using a topological approach. In this paper, we introduce a reasoning scheme called topological reasoning that is used in conjunction with a grasp-based, topological model for uniform representations of multifingered robot hands at different levels of detail (e.g., whole hand, finger, joint), and discuss its application to dextrous manipulation (grasp selection and regrasping). We show that using topological reasoning, both hand posture and 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. Furthermore, the reasoning scheme is applicable not only to tip prehension, but also to palm prehension and any combination of the two.

1. Introduction

A multifingered robot hand, apparently, has sufficient functional richness to permit some level of dextrous manipulation more complex than that provided by traditional grippers and special-purpose end-effectors.

However, hand control as a problem of simultaneously controlling multiple fingers, each finger with several degrees of freedom (dof), causes serious computational difficulties. 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 6-dof space into the space $R^3 \times SO(3)$, where R^3 is the space of gripper positions and $SO(3)$ is its three-dimensional space of orientations. For a k-finger 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 Euclidean space isomorphic to $R^3 \times SO(3)$. This shows how complex the control problem can become, geometrically and analytically.

In this paper, we introduce a topological approach to the formulation and solving of dextrous manipulation problems. The idea of using some form of topology and abstract geometry in robotics is not new, having appeared sporadically in the literature since 1983. In fact, Gottlieb [1] speculated that point-set topology and topological invariants may give insight into practical robotic problems such as singularity avoidance. Recently, Baker [2] 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 [3] applied semi-algebraic geometry to the piano mover's problem of finding a continuous motion from a given initial position to a desired final position of a robot manipulator by considering the static properties of real semi-algebraic sets. Tannenbaum and Yondin [4] went beyond the static properties of such sets by investigating algebraic morphisms between them. They have defined areas of *bad positions* of the robot manipulators by identifying the critical values of certain maps.

An account of applications of topology-relevant techniques to robot-manipulator problems is given in [5]. This includes, for example, work by Lozano-Perez [6] on manipulator configuration space, and work by Hopcroft and Wilfong [7] on motion in contact. These separate investigations of topology and abstract geometry have not yet been brought together in a unified topological and geometrical view of robotic control problems. Neither have they been applied to the control of multifingered robot hands.

Our topological model and topological reasoning scheme are based on two intuitive, simple observations.

First, an arbitrary hand posture (i.e., a spatial arrangement of fingers) appears to form a geometric polyhedron. This geometric polyhedron may be represented by its interior volume, its surface boundary, its edges (i.e., the links of the fingers), or its vertices (i.e., the finger joints). The elements (i.e., vertex, edge, ...) are topological primitives, called *simplexes* in combinatorial topology [8]. Thus, a hand posture is a collection of these simplexes (called a *geometric complex*) connected in some fashion. This first and simple intuitive notion raises an important question: *Would combinatorial topology provide a basic framework into which the geometric, topological description of hand postures at different levels of details (e.g., whole-hand level, finger level, joint level) could be included for hand posture representation and computation, in Cartesian space and in joint spaces?*

Second, from observations of a human hand, the set of all hand postures appears to be bounded by a number of special postures (called *terminal postures*), such as a flat hand or a fist. This set may be considered as a point-set topological polyhedron with vertices representing the terminal postures. When an arbitrary hand posture is viewed as a point X of a topological polyhedron, the question is, *Can an arbitrary posture be measured with respect to those represented by the vertices of a topological polyhedron using a barycentric coordinate system?* Another related question is, *When a point X moves along a certain path on or inside the topological polyhedron, can the changing posture of the hand be predicted?* In other words, *Can techniques drawn from point-set topology and algebraic topology be applied to solve some basic multifingered hand problems such as 1) determination of grasp, 2) change of grasps, and/or 3) continuous grasp planning for prehensile activities?*

This paper suggests some answers to the above questions. The paper is organized as follows. In Section 2, we review and discuss some previous work on a number of representative models of grasps that have been suggested for numeric, symbolic, and integrated control schemes. In Section 3, we present a novel computational model for dextrous grasps. In Section 4, we introduce a topological reasoning scheme based on that model. In Section 5, we apply the model and reasoning scheme to the solution of some basic dextrous-grasp problems. A discussion of the simulation results is also included.

2. Models of grasps and integrated control

At the symbolic level, the models of grasps include Lyons' model [9] as a set of the three simple grasps *encompass grasp*, *precision grasp*, and *lateral grasp*; Iberall's model [10], which consists of three categories based on force-opposability—*pad*, *side*, and *palm oppositions*; and Cutkosky's model [11], a *treelike hierarchy of grasp types* which are described in terms of relations between task requirements and object geometry.

The above models share two 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

Method	Theory/Model	Control scheme
Symbolic	Arbib et al. (1983) (Schema theory) Tomovic, Bekey, and Karplus (1987) (Reflex control)	Lyons (1985): (3 grasps: encompass, precision and lateral) Cutkosky (1986): (16 grasps in a treelike taxonomy) Iberall (1987): (3 grasps: pad, side and palm oppositions) Cutkosky (1989): (Knowledge-based) Liu, Iberall, and Bekey (1989): (Knowledge-based)
Numeric	Salisbury (1982) (Grasping theory) (Li, Hsu, and Sastry (1987, 1988) (Optimum grasping theory) Cutkosky (1989) (Grasp measures)	Cutkosky (1985): (Analysis for stability and resistance to slipping) Kobayashi (1985): (Derivation of grasping/manipulation forces) Holzman and McCarthy (1985): (Prediction of contact forces) Kerr (1986): (Overconstrained and underconstrained grasps) Fearing (1986): (Optimum regrasping) Trinkle (1987): (Enveloping grasp) Salisbury (1988): (Whole-arm manipulation) Pettinato and Stephanou (1988): (Tentacle-based grasp) Nguyen V.D. (1988): (Force-closure) Montana (1988): (Kinematics of contact and of grasp) Yoshikawa and Nagai (1988): (Determination and evaluation of grasp)
Integrated	Tomovic, Bekey, and Karplus (1987) Cutkosky (1989) Erkmen and Stephanou (1989) Nguyen and Stephanou (1990)	

Figure 1

Symbolic, numeric, and integrated models and control schemes.

problem, thus leading to loss of the detailed information necessary for numeric control.

Most symbolic reasoning algorithms are based on these discrete-grasp types. They originate from neurophysiological (Arbib et al., *coordinated control scheme* [12]), anthropomorphic (Cutkosky et al., *knowledge-based control* [11, 13]), or behavioral ideas (Tomovic et al., *reflex control scheme* [14]).

At the numeric level, the model that has been frequently used is the one devised by Salisbury [15], which is based on *screw theory* and line geometry, and which has led to a general theory of grasping and manipulation and the design of the three-finger Stanford/JPL hand. Much work on numeric control for grasping and manipulation for multifingered hands centers around *a characterization of contacts between the fingertips and the grasped object* in tip-prehension mode. Associated control algorithms can be further categorized into joint-level, finger-level, hand-level, and contact-level, or into different analysis aspects (e.g., force, motion, constraints, force-closure), as shown in **Figure 1**.

There are no apparent, simple connections between symbolic and numeric control schemes. The reasons are twofold: 1) There is no uniform representation of hand postures that links grasp models at the symbolic level with those at the numeric level; and 2) 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. For example, the terms *power* and *precision* [16] 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 [17]. Although it is not explicitly modeled, Cutkosky [11] has implied the concept of hand functionality in his description of grasp types. His discussion 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.

At the symbolic level, the problem of grasping has commonly been investigated as a problem of grasp selection [11] 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.

Note also 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 [18], we should also formulate the set of grasps at the symbolic level as a

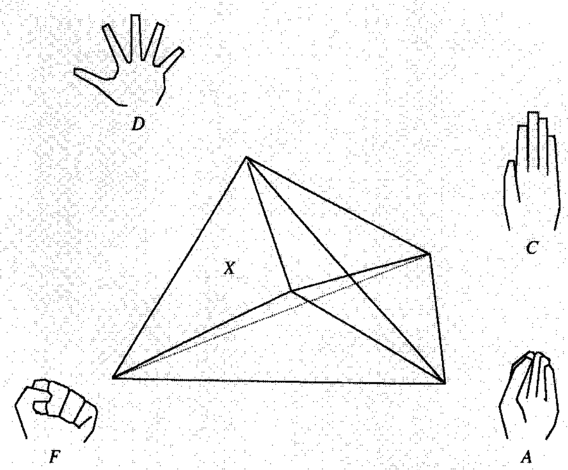

Figure 2

Topological tetrahedron of hand postures.

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 paper [19]. In the following section, we present a computational model of prehensility based on the topological model of multifingered hands.

3. A computational model of prehensility

This section briefly summarizes our computational model of prehensility for multifingered hands. This 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 congruencies defined between those spaces [20].

The *topological* model of multifingered hands (for hand posture and hand functionality) is based on two groups of intuitive concepts: 1) *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 2) *functional concepts* which consist of a *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 *contact subconfiguration* representing a collection of topological primitives that are common (i.e., intersection) to both a hand posture and a grasped object (in an act of grasping and manipulation). These concepts lead to the formulation of postulates and definitions crucial to the development of our computational model.

- *Hand posture*

Postulate 1 Topological representation of a set of all postures. *For a k-finger hand with $k \geq 2$, 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 2**). When the four terminal postures are completely specified (i.e., described in terms of their attributes, dimensions, features, etc.), any arbitrary posture represented by a point X inside or on the boundary of the tetrahedron is uniquely determined by a set of barycentric coordinates of the vertices representing the four terminal postures, i.e.,

$$X = aF + bA + cC + dD,$$

where $a + b + c + d = 1$ with $a, b, c, d \geq 0$ and F, A, C, D are feature vectors of the four terminal postures. For example, the barycentric coordinates of F, A, C, and D are (1, 0, 0, 0), (0, 1, 0, 0), (0, 0, 1, 0), and (0, 0, 0, 1), respectively. Those of the centroid are (1/4, 1/4, 1/4, 1/4).

When two coordinates are equal to zero, the point X representing the corresponding hand posture belongs to the edge connecting the other two vertices. When one barycentric coordinate is equal to zero, X is on the opposite face (triangle) formed by the other three vertices. When none of a, b, c, d are zero, then X is inside the tetrahedron. For each tetrahedron, the four barycentric coordinates a, b, c, d represent only three independent dimensions. The relation $a + b + c + d = 1$ indicates that one variable is dependent on the other three.

If the hand consists of a single finger, the topological tetrahedron is reduced to a topological triangle, since the two configurations C and D now coincide. This triangle represents a point set of all possible finger configurations. A general finger configuration is of the *claw* type.

Postulate 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 $(d - 2)$-polyhedra (edges), and so on until $d = 0$, or, equivalently, a set of vertices of the original d-dimensional polyhedron. These elements (face, edge, and vertex) are topological primitives called *simplexes* in combinatorial topology [8].

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., are properly situated). We have shown [19, 20] that 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 **Figure 3**.

In summary, for an arbitrary hand posture, the following representations are equivalent: 1) a geometric polyhedron for the entire hand, which is then subdivided into properly situated geometric tetrahedra, 2) a collection of properly situated triangular patches representing the dorsum of the hand, 3) a set of properly situated chains representing the fingers, and 4) a set of vertices representing the joints of the digits.

These simple intuitive concepts give rise to the use of point-set topology techniques for approximating an arbitrary hand posture using barycentric coordinates, and 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. It is known that lower-dimension simplexes can be derived from higher-dimension simplexes by applying the boundary theorems [8].

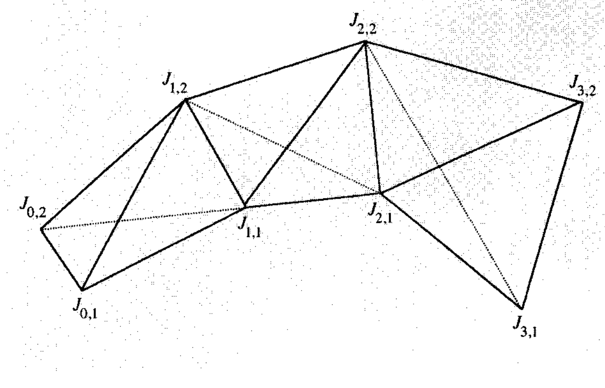

Figure 3

Two-finger hand posture as collection of simplexes.

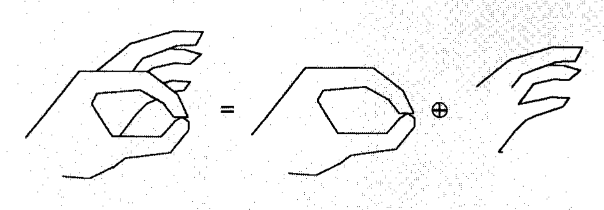

Figure 4

Hand subconfiguration.

- *Hand functionality*

Definition 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.* This power set is a discrete topological space.

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 of the topological concatenation of two or more subconfigurations.

In a fine pinch (**Figure 4**), the five-finger configuration may be decomposed into two subconfigurations: a *thumb-index* subconfiguration and a three-finger *lightly cupped* subconfiguration. Conversely, a configuration s_f is obtained by a *topological concatenation* of two subconfigurations: a two-finger subconfiguration s_2 of the two-finger tetrahedron S_2, and a three-finger subconfiguration s_3 of the three-finger tetrahedron S_3; i.e., $(s_f \in S_5) = (s_2 \in S_2) \oplus (s_3 \in S_3)$, where $\oplus$ is the concatenation operator defined in 2^S, and S_2, S_3, $S_5 \in 2^S$. The concatenation operator and the rules for concatenation are defined in [20]. 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 [21]. We briefly recall here our discussion on hand subconfiguration, contact configuration, and contact subconfiguration.

Definition 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, one complex representing the hand posture and the other representing the graspable object.*

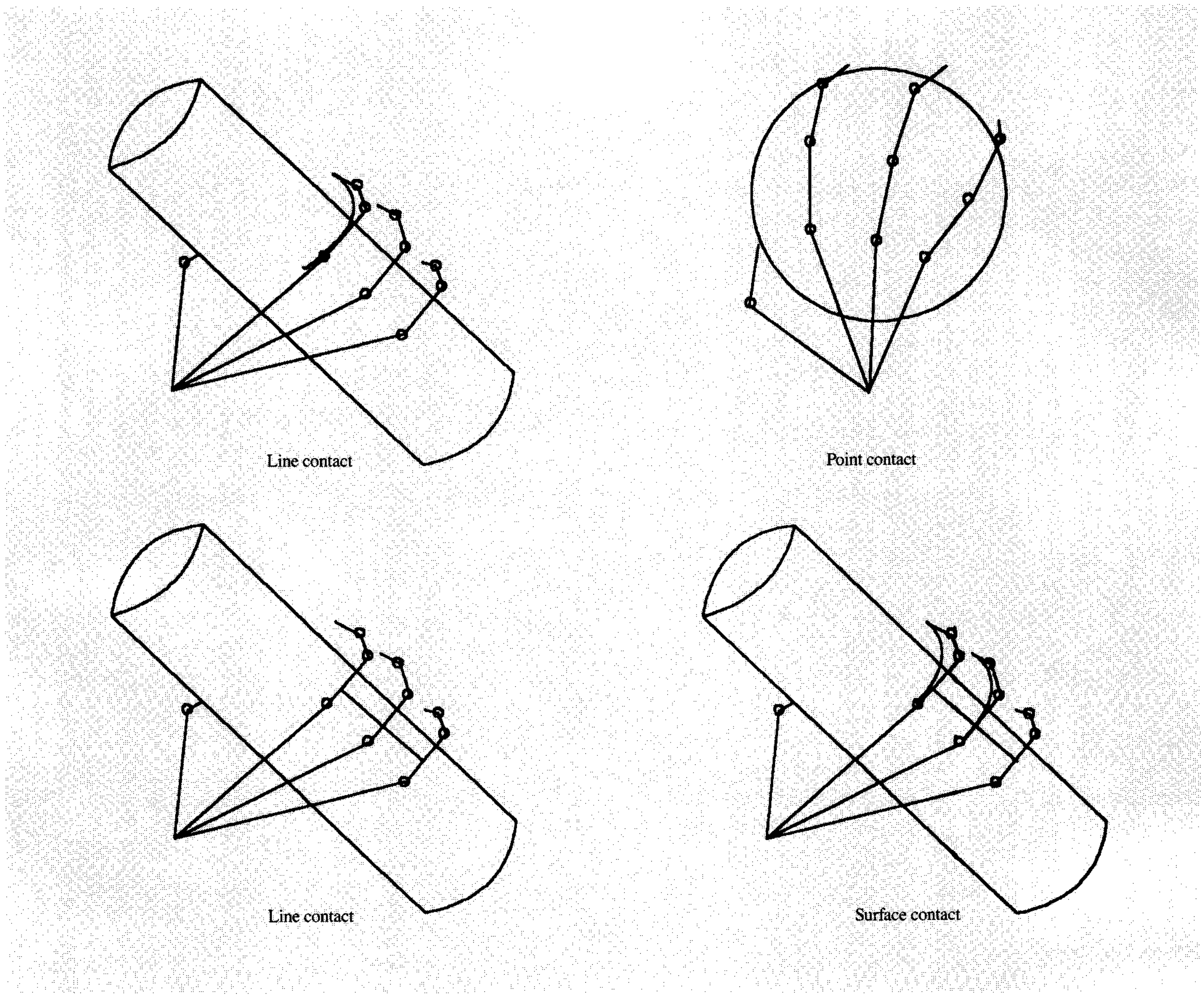

Figure 5

Contact subconfigurations.

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 [15] 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 the subset of all contacts that produce the same effect (e.g., forces of same direction and amplitude) a *contact subconfiguration* (**Figure 5**). In other words, a contact subconfiguration is a subset of contacts that are *functionally equivalent*. For example, a lateral pinch that consists of a thumb in contact with the object and a set of the remaining four fingers (of a human hand) in planar convergent posture [19] has two contact subconfigurations. The first is the one produced by the thumb, which has a relatively small contact area, and 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).

Consider a cupped-hand posture, as shown in **Figure 6**. The same hand posture may be used in three different handling schemes: as a thumb-index precision grip [Figure

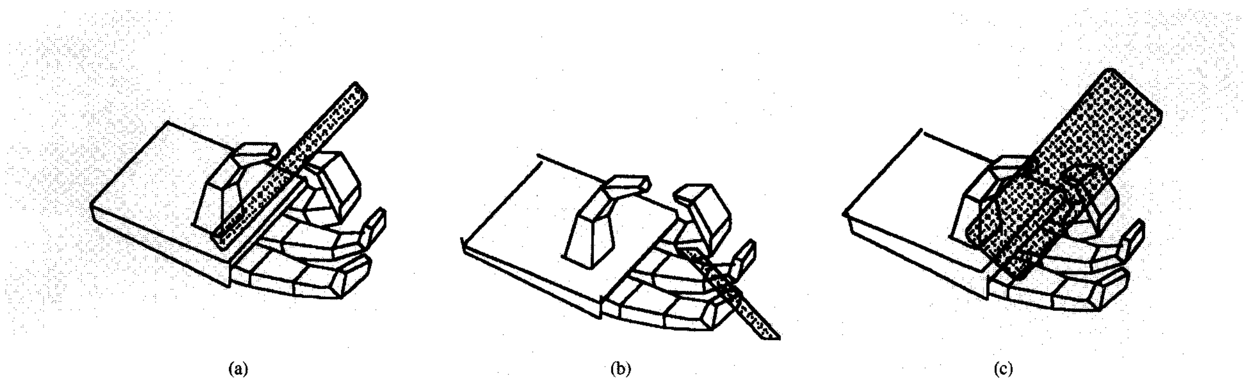

Figure 6

Posture and functionality.

6(a)], as a two-finger precision grip [Figure 6(b)], and as a palm-finger power grip [Figure 6(c)].

Although the hand postures are the same in the three grasps, their hand functionality differs greatly. The differences among them are expressed in terms of the hand subconfigurations, contact configurations, and contact subconfigurations involved. Since a contact configuration is actually the intersection 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 [22], i.e., $CONTACT(H, O) = [((s_1, t_1, type1), \cdots, (s_n, t_n, typeN))]$, 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:

- It can represent contacts in the sense of Salisbury [15], i.e., point, line, and surface contacts, with attributes describing the contact coordinate frames, or other pertinent information about contacts.
- 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, i.e., patches, chains, joints, and the simplexes of the object (i.e., faces, edges, vertices).
- It takes advantage of numerous existing algorithms dealing with the intersection of convex bodies for the determination of actual contact configurations.
- It may be used robustly in reasoning about *anticipated* contacts for hand preshaping or grasp planning.

We introduce the following definitions.

Definition 3 *A* functionality pair *is a pair of contact subconfigurations aimed at either maintaining equilibrium (Σforces = 0, Σmoments = 0), maintaining the stability (ability to resist disturbances) of a grasped object, or allowing the object to have some mobility (e.g., the ability to rotate) in some predetermined directions.*

Definition 4 *A* prehensility pair *is a pair consisting of a group of hand components and a group of object components (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 concept of *functionality pair* may be thought of as a generalization of the concept of *kinematic pair* [23]. The concept of *prehensility pair* may be thought of as a generalization of the concept of *prehensility* [16]. Thus, prehensility pair denotes *what should occur*, and functionality 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 by Reuleaux [23] to classify kinematic pairs into *lower pairs* or *higher pairs*, 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 place along lines or points of contact [24]. 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 because of 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 with a functionality pair; for a large surface-contact area (specified by the *size* of the contact), we have a power-functionality pair (or power-pair, for short), and for a 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 the platform grip and precision-pair in the hook grip. Napier's notions of prehensility and opposability and the derived notion of force-opposition by Iberall [10] may be extended in these cases. When force opposition is between digits or between digits and palm (prehensility pair type I or II), 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), we are dealing with supportability. Thus, Iberall's force-opposability may be considered as a special case of *prehensility pair*. Different types of prehensility pairs have been introduced in [20]. Except for the case of terminal postures, where the postures and functionalities are clearly indicated, 1) in a power grip, a power-pair is required, 2) in a precision grip, a precision-pair is required, and finally, 3) 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 posture) and a *power-functionality pair* (e.g., the act of squeezing by the digits and the opposing palm, where 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, where the suitcase and the groups of fingers are the two elements of the precision-pair).
- A platform grip consists of a *support posture* (e.g., a 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 finger 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 *prehensile functionality*, i.e., the ability to perform a grasping or manipulative task, and *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: 1) supportability, 2) graspability, and 3) manipulability. The corresponding hand functionalities are 1) support (e.g., in a platform grip or hook grip), 2) power (e.g., in a palmar grip), and 3) precision (e.g., in a two-finger pinch). In an act performed by the hand on an object, these functionalities result in three types of prehensility, respectively: 1) palm-prehensility, 2) tip-prehensility, and 3) 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, as detailed in [25].

4. Topological reasoning about dextrous grasps

Several investigations have been conducted to apply to robot hands some capability of reasoning about dextrous prehension. An example is the effort by Cutkosky [11] with GRASP-EXP, an expert system designed for automatic grasp generation. Another example is SUPER-GRIP, designed to generate the best grasp for picking up a rectangular box [26]. Most current expert systems for grasp selection are knowledge-based, with rules derived from observations of human prehension. The rules supposedly express *prehensility knowledge*, i.e., the knowledge required to perform a prehensile task. This prehensility knowledge is used by some reasoning schemes designed for various grasp-synthesis problems. Some authors use deductive schemes [27, 28], some use inductive methods [29], while others use evidential reasoning [30].

Prehensility knowledge is required for both symbolic and numeric controls. At the symbolic level, prehensility knowledge is partially expressed in terms of task attributes such as stability, graspability, manipulability, and supportability. At the numeric level, it is partially expressed as a collection or sequence of primitive actions and constraints, e.g., one or more of the following: 1) equilibrium conditions ($\Sigma forces = 0$ and $\Sigma moments = 0$); 2) insensitivity to disturbances (meeting stability requirements; i.e., net forces are within friction cones determined by contact normal and the cone angle α); 3) primitive translation (expressed in terms of current location, translation direction, and magnitude) along a known axis; and 4) primitive rotation (expressed in terms of current location, axis of rotation, and angle of rotation).

A common approach to reasoning about dextrous grasps is to reason about the tasks, and about the objects separately, and then to merge these two processes into a process of reasoning about task–object combinations to derive a set of grasp-relevant attributes. The attributes obtained are, in turn, used to determine a particular grasp from the small and finite set of available grasps [11]. The merging normally involves the cross product of the task-attribute domain and the object-attribute domain, thus drastically increasing the complexity of the associated grasp-synthesis techniques.

Although our approach to topological reasoning appears, on the surface, to be the same as most current reasoning schemes (i.e., reasoning about tasks, reasoning about objects, and reasoning about grasps), there are two basic differences between our reasoning scheme, called *topological reasoning*, and the others. First, our reasoning is based on topological properties of tasks, objects, and contacts, and second, it avoids the cross-product operation by mapping these properties into the topological contact domain.

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., joints, 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.

Topological reasoning extends the concept of *geometric reasoning*, which is a form of representation and reasoning about geometry [31]. The topological model also facilitates the acquisition of prehensility knowledge. For example, in an assembly task of two component parts, a symbolic prehensile task may involve a simple operation such as placing one part in contact with another. A topologically relevant question may be asked: *What is the anticipated contact configuration between the two parts: point contact, line contact, or surface contact?* Another question may be, *If it is a surface contact, can the surface be identified?* A grasp-relevant question may be, *Is the anticipated contact surface area large enough to require a placement of more than one finger or the palm?* These questions describing some topological and geometrical situations are instrumental in the formulation of appropriate rules for reasoning about hand configurations. These questions and rules are then grouped according to their specific details—for example, into rules for two-finger subconfigurations, rules for the entire hand, or rules for reasoning about anticipated contact subconfigurations.

We now describe how the computational model and the topological reasoning can be used in the general framework of an *intelligent robot prehension* scheme. The scheme is called a *prehension* scheme because it deals with three basic elements of prehension [32]: an intent (a task), perception, and the mechanism of grasping and manipulation. The *intelligent* aspect is associated with the topological reasoning scheme described below. Note that in our scheme, whether dextrous hand activity involves grasping or regrasping, it consists of two processes—a postural transformation for the derivation of hand posture, and a functional transformation for the derivation of hand functionality to achieve a set of intended contact configurations. The form of perception discussed here is called *perception by memory*, a term coined by psychologists [33] to indicate a recall of perceived things previously stored in memory. We implement the concept of perception by memory through the use of prototypes.

There are three types of prototypes: task prototypes, object 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 the individual components that characterize it and their interconnections. A functional description reveals the purpose of the structural component or connection in producing the behavior expected from 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 or similarity between a given instance I (of task, object, and/or hand posture) and a prototype P.

A high-level, prehensile task is a task expressed in general terms such as *build* or *assemble*. A high-level task may be decomposed into subtasks, e.g., *grasp*, *screw*. For each subtask, we consider two types of tasks: those that produce no motion to the object, or *type I* tasks, and 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 precede, follow, or coexist with type I tasks. A low-level or primitive task is one expressed in terms of forces and 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 multiple levels of task specifications:

1. High-level (abstract) tasks such as *build*, *assemble*.
2. Subtasks (typed tasks) such as *preshape*, *enclose*, *hold*.
3. Sub-subtasks, etc.
4. 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 subtasks (typed tasks) and primitive tasks at each of the nodes of the task tree-structure.

To describe a high-level task, following Yeap [33] we present here the idea 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 aforementioned functional details and constraints. The raw map then takes the form of a *full task map*. The structures of both the raw map and the full map of each task are the same: All the characteristics and attributes are organized in four categories: geometrical, topological, functional, and behavioral. Each task (subtask, primitive) 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, regardless of *how to hold*. Although there is more than one possible way of *how to hold* a given object, all such possibilities satisfy a single functional requirement: equilibrium. The *how to do* is derived from our reasoning process using the algorithms detailed in [20, 25].

Task prototypes describe generic tasks. A task prototype is also described by a task map, as mentioned 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. The subprocess of filling out detailed information in a task map from task specifications is called *reasoning about tasks* (task mapping, **Figure 7**).

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 [34]. 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 [34, 35]. The subprocess of filling out detailed information in a full task map from object specification is called *reasoning about objects* (object mapping, Figure 7).

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 may use the same data structure. The subprocess of using detailed information in a full task map to derive hand postures is called *reasoning about hand postures* (block II, Figure 7).

Thus, block I in Figure 7 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 Figure 7 consists of 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); a posture mapping for processing posture-oriented attributes; and a functionality mapping for processing functionality-oriented attributes. In block III, there are two subprocesses—geometric transformation and contact mapping. These processes together perform what is called *reasoning about contact configurations* (block III, Figure 7).

Basically, topological reasoning uses a set of algorithms to derive a suitable grip given a symbolic task description and a graspable object. The set of algorithms can derive hand posture for palm prehension or complex

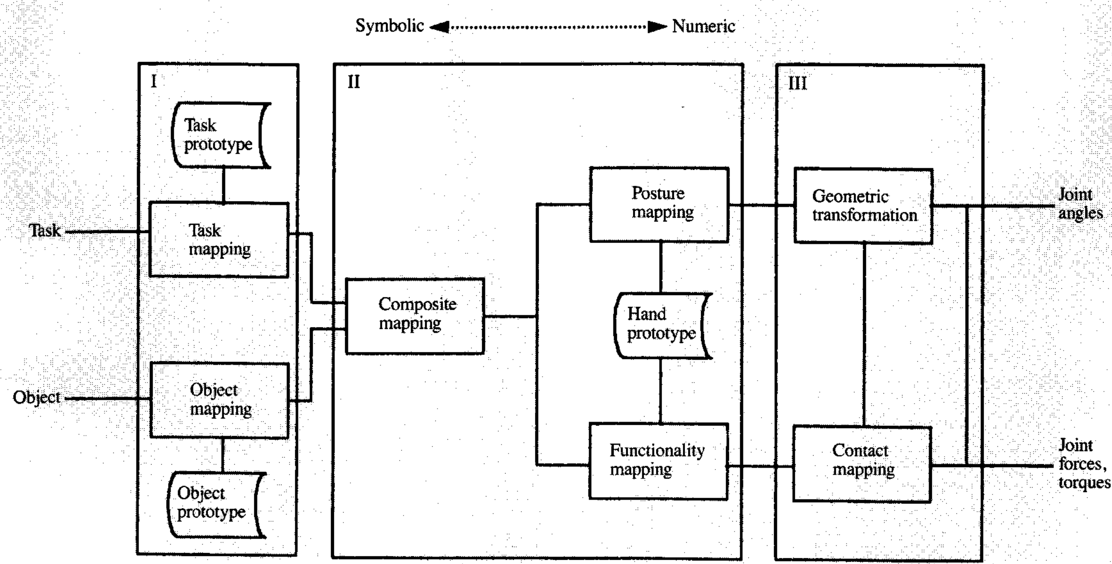

Figure 7

Functional block diagram.

combinations of palm/tip prehension; 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 guides the development of rules representing the necessary prehensility knowledge.

We recall below the set of topological algorithms that have been detailed elsewhere [20, 25], upon which our topological reasoning is built. This set of algorithms aims at providing solutions to the transformation from symbolic task description and specifications to the numeric description of hand posture and functionality.

- *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.
- *Step 4: Determination of hand subconfiguration posture* (postural transformations and topological reasoning about hand postures)
  - e. Approximation of barycentric coordinates (tip prehension).
  - f. Polyhedral approximation of a hand posture (palm prehension).

The details of these two algorithms may be found in [20, 25]. The output obtained from the 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. These algorithms are *posture*-oriented; i.e., this step has not taken into account explicitly the functional aspects of a grasp.

- *Step 5: Determination of hand subconfiguration functionality* (functional transformations and topological reasoning about hand functionality)
  - g. Barycentric subdivision of subconfiguration space.
  - h. Detection of functionality pair and prehensility pair.

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, functionality pair and prehensility pair are notions parallel to Reuleaux's notions of kinematic pairs and Napier's notions of prehensility and opposability. This parallelism plays a key part 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-type function with the edge of the palm, or a squeeze function between the edges of the fingers. 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), or with a precision grip (five fingers in fingertip contact with the object).

- *Step 6: Derivation of joint space variables*
  i. Mapping of functionality/prehensility pairs into contact space.

This process is based on reasoning about contacts in contact space. The mobility of a hand-held object is determined by the nature of contacts, the type of contact, the number of contacts, the contact configurations (i.e., spatial arrangement of contacts), and the contact motion allowed (e.g., sliding, rolling). Contacts may be described in terms of forces and velocities (i.e., in terms of a wrench and twist system) between the multifingered hand and the grasped objects at a body frame of coordinates, at the contacts (in Cartesian coordinates), and at the joint coordinate systems [36, 37].

Since each finger interacts with the grasped object, for each contact between the finger and the object there exists a single closed-loop kinematic chain; i.e., there are six kinematic constraint equations which define the loop: three position and three orientation equations. The only variables in these equations are the degrees of freedom in the loop, i.e., the joint angles. The sets of contacts describing the kinematic relations must be satisfied to allow the wrench (twist) systems at the contacts and at the finger joints to be computed given the required external net force. The kinematic relations describing a hand–object system have been derived by Salisbury (using grip transform [15]), extended by Kerr and Roth (selection of internal forces in overconstrained and underconstrained hand–object system [38]), augmented with stiffness control to resist slippage by Cutkosky and Wright [13], and allowed rolling contact motion with compliance to be monitored by Montana [39] and by Hsu et al. [37] and Li and Sastry [36]. For a closed kinematic chain, one may analyze the hand–object system in terms of transformations between coordinate frames as defined and used by Li and Sastry [36] (coordinate frame of the object body at its center of mass, coordinate frames at each contact point, coordinate frames at each joint, coordinate frame at the palm, and coordinate frame—the fixed base—of the hand system).

The mapping from hand functionality and prehensility pairs to contact space is a computation-intensive process. At the joint and contact level, there are many existing algorithms for grasping and manipulation in tip prehension mode, including algorithms for simple grips with a two-finger planar, frictionless hands (e.g., [40]), and algorithms for the more complex case of a three-soft-finger hand (e.g., [36]). The mathematics of these grips are well documented.

5. Applications to dextrous manipulation

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) must be refined after the first contact, the locations of contacts are easily disturbed because of slippage or rolling, or the objects themselves are not rigid. Regrasping consists of a change of position (repositioning), or a change of force (force adjustment) with or without a change of position. In the following subsections, we discuss the application of our scheme to two problems, grasp selection and regrasping. Simulation results are included for the former case.

- *Grasp selection*

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 [11] indicates the conditions to be satisfied by the forces and moments

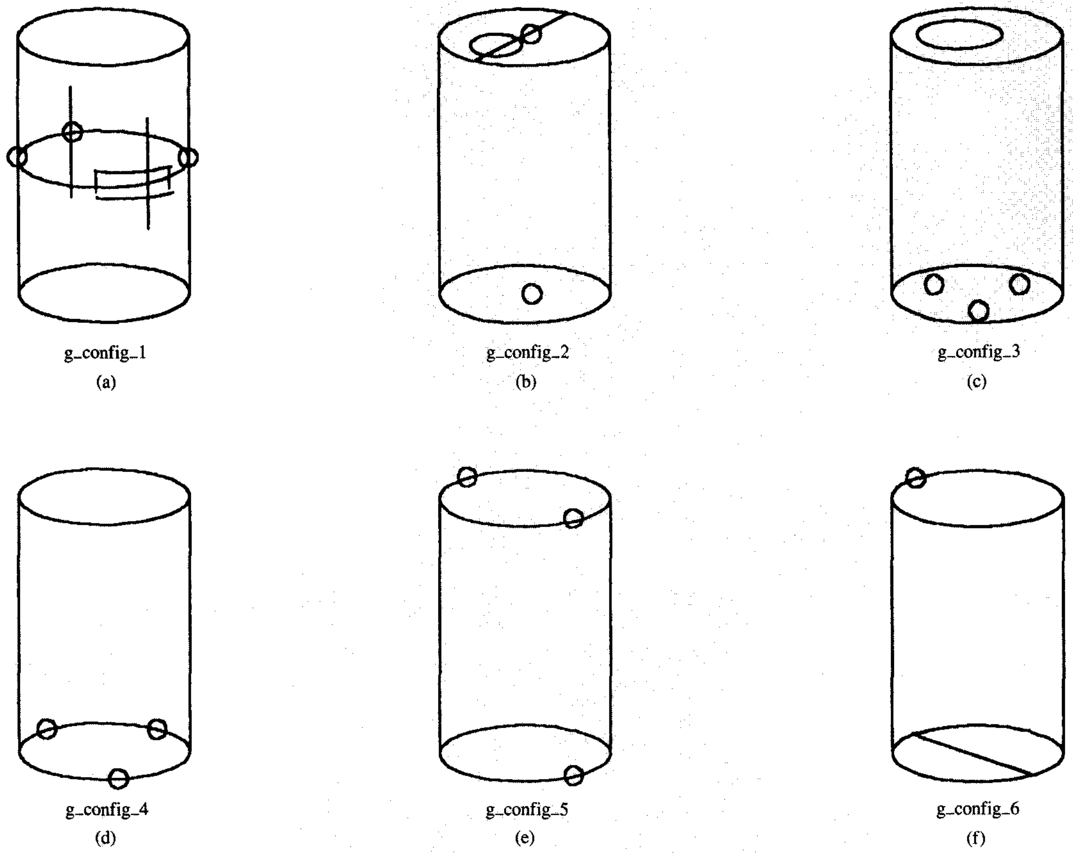

Figure 8

Graspable configurations.

applied via the anticipated contacts without breaking the contacts, and finally, connectivity [15] indicates the number of task degrees of freedom 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 a *stable* hold.

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

The set of all possible graspable configurations (**Figure 8**) includes [25] (a) cylindrical graspable configurations (g _config _1); (b, c) circular graspable configurations (g _config _2 and g _config _3); and (d–f) configurations graspable by the edges (g _config _4, g _config _5, g _config _6). Each graspable configuration may involve any combination of point, line, and surface contacts, 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 the contact types defined and modeled by Salisbury [15]. 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 multiple point contacts occur between a finger and the object (on successive finger links), or when there is a set of point contacts, each of which is between the same finger link and the object. We have a *surface contact* when there is more than one line contact between the hand and the object. A precision grasp in the sense of Napier [16] most likely occurs in tip prehension, which involves a set of point contacts by the fingertips. A power grasp or support

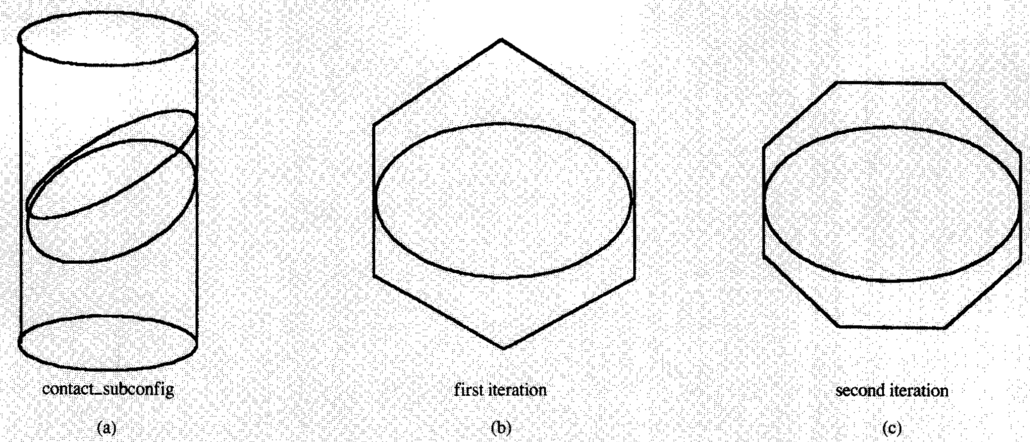

Figure 9

Iterations of hand posture.

grasp, most likely occurring in palm prehension, involves a combination of point, line, and surface contacts. The determination of the 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 described in Section 4. In this subsection, we report and evaluate the results of our case study on grasp selection. The grasp selection is limited to the following processes:

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

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 0.84.
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
In this simulation run, the object is smaller than the hand size (refer to the object dimensions; 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 [19], 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 0.84, the size of the cylinder diameter. Furthermore, the anticipated contacts should be of line contact type, with each finger in line contact with

the object. This is a power grasp in the sense of Napier [16].

The problem is to find a polygonal approximation for each finger, knowing the contact subconfiguration and the wrist position. This is equivalent to the problem of identifying an outer polygonal covering of the intersection between the hand and the object. For example, in the case of graspable configuration 1 (cylindrical face), the contact subconfigurations involved are the elliptic cross sections [**Figure 9(a)**].

The polyhedral approximation comprises, first, the generation of a regular hexagon covering the cross section [**Figure 9(b)**] and then an iterative shortening of the edge of the polygon until it is comparable to the phalange length [**Figure 9(c)**]. The finger shape is the shape of the polygon obtained at the last iteration.

The approximation of hand posture using the 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 0.97. This edge is much larger than the finger link, which is 0.25 (the finger has four links, and each link is then 1/4 = 0.25). On the second iteration, the hexagon becomes an octagon with edge length equal to 0.58. Therefore, another iteration is necessary. The third iteration, a decagon, has an edge length that is equal to 0.24, which is smaller than the finger link, 0.25, and the process stops. The joint angles are the angles between adjacent edges of the decagon. The following rules have been fired:

Reasoning for Case 1

```
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 subtract_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 now illustrate the results of a precision grasp and a support grasp without further explanation.

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
                          SUBCONFIGURATION,
                        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 0.81.
```

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.
```

- *Regrasping tasks*

Regrasping (change of hand posture) exists in three ways: 1) from a subfunctionality with few fingers to one with more fingers, 2) from a subconfiguration of more fingers to one with fewer fingers, and 3) a change of subconfiguration for which the number of fingers remains the same. The first two cases involve *a mapping between complexes*, which is not discussed here. As an example, we elaborate only the third case.

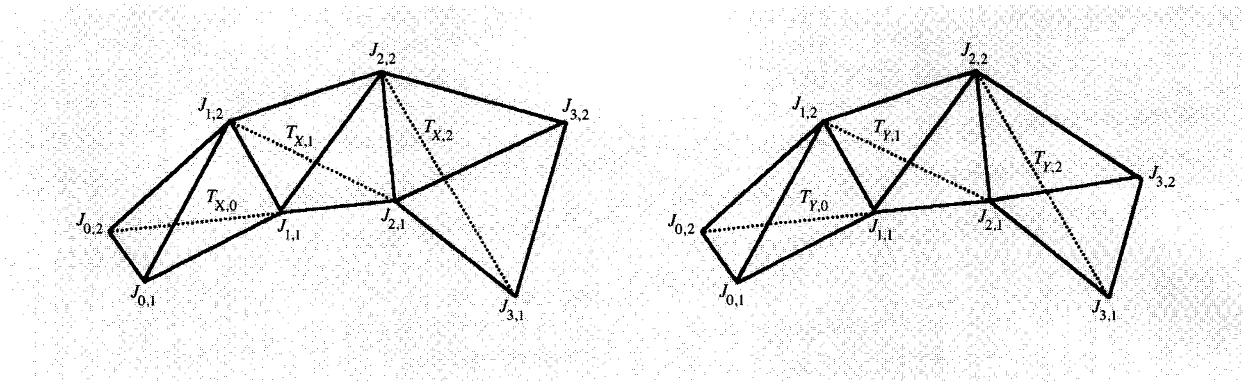

Figure 10

Change in hand subconfigurations.

Traditionally, as a result of changes in hand shape, the grasp matrix must be recomputed analytically. Topologically, the change in hand subconfiguration may involve only some of the constituent topological simplexes of a particular geometric complex representing the current hand posture. Thus, it is possible to partially recompute some simplexes to reflect the change, thereby avoiding the total recomputation of all simplexes.

A topological or geometric change of simplexes may occur either in their connectivity to other simplexes or in their own geometry. A change in connectivity and in subconfiguration geometry is due to two basic motions at some or all of the joints: *abduction/adduction* or *flexion/extension*. A flexion/extension motion changes the joint angle at the joint of two connected links. An abduction/adduction motion, however, changes both the joint angle and the angles between consecutive fingers.

The distinction between a topological and a geometric change is clarified in a rule, called the *rule for simplex identification*: *A subconfiguration change is topological if it involves a continuous mapping of the geometry of some or all of the simplexes. It is geometric if the rule produces a unique simplex geometry for each simplex.*

In other words, the discrimination question (between a topological change and a geometrical change) is this: *Does the continuous transformation vary the geometry of some simplexes while maintaining connectivity?* If the answer is yes, the change (i.e., transformation) is topological; otherwise, it is geometric. In flexion/extension, the change may be seen as a folding/unfolding operation in which an axis through the joint becomes the axis of flexion/extension. The shape of the 2-simplex is preserved, and only its orientation changes. In abduction/adduction at each joint, the angle between fingers changes linearly, $\alpha \in (0, A_{max})$, where A_{max} is the joint limit of the corresponding revolute joint.

Consider now the two thumb–index subconfigurations in **Figure 10**; the one on the left is a fine thumb–index pinch S_X, and the one on the right, S_Y, is derived from S_X from the flexion motion of the distal phalanges at the distal joints (joints $n + 1$). Let K_X and K_Y denote the two subconfiguration complexes corresponding respectively to the two subconfigurations. In terms of the geometric tetrahedron representation, the geometric tetrahedra $T_{X,0}$, $T_{X,1}$, $T_{Y,0}$, and $T_{Y,1}$ remain unchanged. The distal geometric tetrahedron $T_{Y,2}$ may be obtained from $T_{X,2}$ by a rotation (flexion) about the axis $J_{0,2}$, $J_{1,2}$. In other words, one needs only to recompute the complex corresponding to $T_{Y,2}$ instead of the entire complex K_Y.

In a geometric polyhedron representing a hand posture, the scheme described above allows the computation of a topological or geometrical complex representing the hand posture in coordinate frames associated with the vertices (hence, in joint coordinate variables). However, it cannot determine which side of the boundary surface (i.e., the triangular surfaces) is the dorsum of the hand. This difficulty is eliminated by using oriented boundaries, known as *oriented simplexes* in algebraic topology [41].

6. 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 multifingered robot prehension.*

We have detailed a reasoning procedure for deriving hand posture and hand functionality from symbolic task specifications, described by a task map and translated into contact configuration specifications, and object specifications, described in terms of graspable configurations and 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 concept of contact subconfigurations as a group of contacts and as a subset of contact configurations, and used them in describing a contact configuration as a hierarchy of contacts (i.e., configurations, subconfigurations, contacts) and in facilitating the description of power-oriented grasps, support grasps, and precision-grasps. Again, all contacts are specified 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 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 at the symbolic level.

The overall design of an intelligent robot prehension scheme has also 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 (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, and provides a natural way to link to geometric reasoning.

We have presented the application of the topological reasoning scheme to both the problem of grasp selection and the problem of regrasping. We have pointed out numerous advantages of our scheme, one of which is the greatly reduced recomputation of the grasp matrix.

The *continuous* nature of the topological model has not been fully explored in our current work. The topological tetrahedron representing the set of all postures permits a *neighborhood* concept to be defined in the set. Motion planning and path planning of the fingers are then represented as curves inside or on the boundary of the tetrahedron. The one-to-one and bicontinuous nature of mappings among various representational spaces of hand postures, at different levels of detail, allows inverse transformations to be defined. This may be particularly useful in the study of numeric sensing and in extending the current model to active sensing. Our model and reasoning scheme, so far, is used as a straightforward transformation from symbolic to numeric representation spaces. Future work will include an extension to a sensor-based control scheme, which is a perception–action integration that gives rise to the application of *evidential reasoning* to dextrous manipulation.

References

1. D. H. Gottlieb, "Robots and Topology," *Proc. IEEE Int. Conf. Robotics & Automation*, 1986, pp. 1689–1691.
2. D. R. Baker, "Some Topological Problems in Robotics," *Math. Intell.* **12**, No. 1, 66–76 (1990).
3. J. T. Schwartz and M. Shahir, "On the Piano Mover's Problem II: General Techniques for Computing Topological Properties of Real Algebraic Manifolds," *Adv. Appl. Math.* **4**, 298–351 (1983).
4. A. Tannenbaum and Y. Yondin, "Robotic Manipulators and the Geometry of Real Semialgebraic Sets," *IEEE J. Robotics & Automation* **RA-3**, No. 4, 83–89 (1987).
5. J. T. Schwartz and M. Shahir, "A Survey of Motion Planning and Related Geometric Algorithms," *Artif. Intell.* **37**, 157–169 (1988).
6. T. Lozano-Perez, "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Computers* **C-32**, No. 2, 108–120 (1983).
7. J. Hopcroft and G. Wilfong, "Motions of Objects in Contact," *Int. J. Robot. Res.* **4**, No. 4, 32–46 (1986).
8. L. S. Pontryagin, *Foundations of Combinatorial Topology*, Graylock Press, Rochester, NY, 1952.
9. D. M. Lyons, "A Simple Set of Grasps for a Dextrous Hand," *Proc. IEEE Int. Conf. Robotics & Automation*, 1985, pp. 588–594.
10. T. Iberall, "The Nature of Human Prehension: Three Dextrous Hands in One," *Proc. IEEE Int. Conf. Robotics & Automation*, 1987, pp. 396–401.
11. M. R. Cutkosky, "On Grasp Choice, Grasp Models and the Design of Hands for Manufacturing Tasks," *IEEE Trans. Robotics & Automation* **5**, No. 2, 151–165 (1989).
12. M. A. Arbib, T. Iberall, and D. Lyons, "Coordinated Control Programs for Movements of the Hand," *Coins Tech. Report*, University of Massachusetts, Boston, 1983.
13. M. R. Cutkosky and P. K. Wright, "Manufacturing Grips and Correlation with the Design of Robotic Hands," *Proc. IEEE Int. Conf. Robotics & Automation*, 1986, pp. 1533–1539.
14. R. Tomovic, G. A. Bekey, and W. J. Karplus, "A Strategy for Grasp Synthesis with Multifingered Robot Hands," *Proc. IEEE Int. Conf. Robotics & Automation*, 1987, pp. 83–89.
15. J. K. Salisbury, "Kinematic and Force Analysis of Articulated Hands," Ph.D. thesis, Department of Mechanical Engineering, Stanford University, CA, 1982.
16. J. R. Napier, "The Prehensile Movement of the Human Hand," *J. Bone & Joint Surgery* **38B**, 902–913 (1956).
17. J. M. Landsmeer, "Power Grip and Precision Handling," *Ann. Rheum. Diseases* **22**, 164–170 (1962).
18. Z. Li, J. F. Canny, and S. Sastry, "On Motion Planning for Dextrous Manipulation, Part I: The Problem Formulation," *Proc. IEEE Int. Conf. Robotics & Automation*, 1989, pp. 775–780.
19. T. N. Nguyen and H. E. Stephanou, "A Topological Model of Multifingered Prehension," *Proc. IEEE Int. Conf. Robotics & Automation*, 1989, pp. 446–451.
20. T. N. Nguyen and H. E. Stephanou, "A Topological Algorithm for Continuous Grasp Planning," *Proc. IEEE Int. Conf. Robotics & Automation*, 1990, pp. 791–796.
21. T. N. Nguyen and H. E. Stephanou, "A Computational Model of Prehensility and Its Applicability to Dextrous Manipulation," *Proc. IEEE Int. Conf. Robotics & Automation*, 1991, pp. 772–777.
22. C. Laugier, "Planning Fine Motion Strategies by Reasoning in Contact Space," *Proc. IEEE Int. Conf. Robotics & Automation*, 1989, pp. 653–659.

23. F. Reuleaux, *The Kinematics of Machinery*, Dover Publications, New York, 1963.
24. A. Gelsey, "Automated Reasoning About Machine Geometry and Kinematics," *Proc. IEEE Int. Conf. Robotics & Automation*, 1987, pp. 182–187.
25. T. N. Nguyen and H. E. Stephanou, "Distributed Intelligent Control of Multifingered Robot Hands," *Proc. US–Japan (ISCIE & ASME) Symposium on Flexible Automation*, 1990, pp. 1099–1106.
26. M. Pelletier and V. Hayward, "SUPER-GRIP: An Expert System for Grasping Boxes," *Proc. IEEE Int. Conf. Robotics & Automation*, 1989, pp. 510–515.
27. T. Iberall, "Knowledge-Based Prehension: Capturing Human Dexterity," *Proc. IEEE Int. Conf. Robotics & Automation*, 1988, pp. 82–87.
28. H. Liu, T. Iberall, and G. Becky, "The Multi-Dimensional Quality of Task Requirements for Dextrous Robot Hand Control," *Proc. IEEE Int. Conf. Robotics & Automation*, 1989, pp. 452–457.
29. G. B. Dunn and J. Segen, "Automatic Discovery of Robotic Grasp Configurations," *Proc. IEEE Int. Conf. Robotics & Automation*, 1988, pp. 396–401.
30. A. M. Erkmen and H. E. Stephanou, "Preshape Jacobians for Minimum Momentum Grasping," *Proc. IEEE Int. Conf. on Systems, Man & Cybernetics*, 1989, pp. 790–795.
31. D. Kapur and J. L. Mundy, "Geometric Reasoning and Artificial Intelligence: Introduction to the Special Volume," *Artif. Intell.* **37**, 1–11 (1988).
32. S. H. Harrison, "The Functional Relationship of the Thumb to the Fingers," *The Hand*, Vol. 1, R. Tubiana, Ed., W. B. Saunders, Philadelphia, 1981, pp. 481–487.
33. W. K. Yeap, "Towards a Computational Theory of Cognitive Maps," *Artif. Intell.* **34**, 297–360 (1988).
34. C. N. Stevenson, "Model-Based Programming and Control of Robot Manipulators," *IEEE Computer* **20**, 76–84 (1987).
35. A. A. G. Requicha and H. B. Voelcker, "Boolean Operations in Solid Modeling: Boundary Evaluation and Merging Algorithms," *Proc. IEEE* **73**, No. 1, 30–44 (1985).
36. Z. Li and S. Sastry, "Task-Oriented Optimal Grasping by Multifingered Robot Hands," *Proc. IEEE Int. Conf. Robotics & Automation*, 1987, pp. 389–394.
37. P. Hsu, Z. Li, and S. Sastry, "On Grasping and Coordinated Manipulation by a Multifingered Robot Hand," *Proc. IEEE Int. Conf. Robotics & Automation*,- 1988, pp. 384–389.
38. J. Kerr and B. Roth, "Analysis of Multifingered Hand," *Int. J. Robotics Res.* **4**, No. 4, 3–17 (1986).
39. D. J. Montana, "The Kinematics of Contact and Grasp," *Int. J. Robotics Res.* **7**, No. 3, 17–31 (1988).
40. R. Volz, J. Wolter, J. Barber, R. Desai, and A. C. Woo, "Automatic Determination of Gripping Positions," *Languages for Sensor-based Control in Robotics*, NATO ASI Series, Vol. F29, Springer-Verlag, New York, 1987, pp. 481–505.
41. S. Eilenberg, "Algebraic Topology," *Lectures on Modern Mathematics*, Vol. I, T. L. Saaty, Ed., John Wiley & Sons, New York, 1961, pp. 99–114.

Bibliography

M. R. Cutkosky, *Robotic Grasping and Fine Manipulation*, Kluwer Academic Publishers, 1985.

R. S. Fearing, "Simplified Grasping and Manipulation with Dextrous Robot Hands," *IEEE J. Robotics & Automation* **RA-2**, No. 4, 188–195 (1986).

W. Holzmann and J. M. McCarthy, "Computing the Friction Forces Associated with a Three-Fingered Grasp," *IEEE J. Robotics & Automation* **RA-1**, No. 4, 206–210 (1985).

H. Kobayashi, "Control and Geometrical Considerations for an Articulated Hand," *Int. J. Robotics Res.* **4**, No. 1, 3–12 (1985).

V. D. Nguyen, "Constructing Force-Closure Grasps," *Int. J. Robotics Res.* **7**, No. 3, 3–16 (1988).

J. S. Pettinato and H. E. Stephanou, "Manipulability and Stability of a Tentacle Based Robot Manipulator," *Proc. IEEE Int. Conf. Robotics & Automation*, 1988, pp. 458–463.

J. K. Salisbury, W. Townsend, B. Iberman, and D. DiPietro, "Preliminary Design of a Whole-Arm Manipulation System (WAM)," *Proc. IEEE Int. Conf. Robotics & Automation*, 1988, pp. 254–260.

J. C. Trinkle, J. M. Abel, and R. P. Paul, "An Investigation of Frictionless Enveloping Grasping in the Plane," *Int. J. Robotics Res.* **7**, No. 3, 33–51 (1988).

T. Yoshikawa and K. Nagai, "Evaluation and Determination for Multifingered Hands," *Proc. IEEE Int. Conf. Robotics & Automation*, 1988, pp. 245–248.

Received November 20, 1990; accepted for publication March 12, 1992

Thang N. Nguyen *IBM United States, 800 N. Frederick Avenue, Gaithersburg, Maryland 20879 (THANGN at DALVM41B).* Dr. Nguyen received his B.S. in electrical engineering from Laval University in 1966, his M.S. in information and computer science from the Georgia Institute of Technology in 1973, and his Ph.D. in information technology and engineering from George Mason University in 1990. Dr. Nguyen is currently with IBM Systems Management Marketing, where his primary work has been investigating and modeling business knowledge acquisition for automated operations processes. As part of this assignment, he publishes and makes presentations on business-directed automation, integrated business strategies, and IT operations schemes. Prior to this assignment, he worked with IBM Professional Services and provided systems support services to large customers in both MVS operating system and telecommunications areas. Since 1989, he has published actively in the fields of robotics and automation as well as software engineering. His research interests include symbolic–numeric integration for intelligent control of multifingered robot hands, knowledge-data models, business process modeling, business policy (knowledge) acquisition, and reasoning schemes. Dr. Nguyen is a member of the IEEE; he has received three IBM Outstanding Achievement Awards.

Harry E. Stephanou *Rensselaer Polytechnic Institute, Troy, New York 12180 (hes@cat.rpi.edu).* Dr. Stephanou received the Ph.D. degree in electrical engineering from Purdue University in 1976. From 1976 to 1985, he was with the Exxon Production Research Company, where he headed the Systems Research Section, Long-Range Research Division. In 1985, he joined the George Mason University School of Information Technology and Engineering. From 1987 to 1988, he also served as Program Director for Robotics and Machine Intelligence at the National Science Foundation. Since June 1990, he has been at the Rensselaer Polytechnic Institute, where he is a faculty member in the Department of Electrical, Computer and Systems Engineering, and serves as Director of the New York State Center for Advanced Technology in Automation and Robotics. Dr. Stephanou is a Senior Member of the IEEE, Vice-President of the IEEE Society for Robotics and Automation, and Chairman of the IEEE Control Systems Society Technical Committee on Intelligent Control. He is the General Chairman of the 1991 IEEE International Symposium on Intelligent Control. His research interests are in the areas of intelligent robotic systems, multifingered robot hands, sensor fusion, and reasoning under uncertainty.