Structural Synthesis of Hands for Grasping and Manipulation Tasks

Ali Tamimi, Alba Perez-Gracia, and Martin Pucheta

Abstract In the kinematic synthesis of multi-fingered robotic hands for a specific task, the selection of the hand topology is an important step. Considerable research efforts have been directed to the structural synthesis of hand topologies for satisfying grasping and manipulation metrics such as mobility and force closure. In this work, we develop a structural synthesis, isomorphism-free enumeration method that combines the solvability for rigid-body guidance with the grasping and manipulation metrics, for general hands with a tree structure. An algorithmic implementation of the methodology is presented and illustrated with validation examples.

Key words: Multi-fingered robotic hands, Dexterous grasping, Structural synthesis, Spatial dimensional synthesis.

1 Introduction

Multi-fingered robotic hands are mechanical linkages where a common set of links spans a number of serial chains, designed for grasping and manipulation tasks. Traditionally, a robotic hand consists of a single link, or palm, spanning several subchains, which are the fingers. This definition can be extended to consider a common set of links and joints spanning the finger chains, possibly in several stages.

Ali Tamimi · Alba Perez-Gracia
Department of Mechanical Engineering
Idaho State University, Pocatello, ID, USA
e-mail: tamiali@isu.edu, perealba@isu.edu

Martin Pucheta
Centro de Investigación en Informática para la Ingeniería (CIII) and CONICET
F.R. Córdoba - Universidad Tecnológica Nacional, Córdoba, Argentina
e-mail: mpucheta@frc.utn.edu.ar

The structural or type synthesis of multi-fingered hands seeks to enumerate all possible topologies for a desired quality of the hand. Most of the previous work focuses on enumerating the topologies according to the mobility of the hand. Salisbury and Roth [4] studied the type synthesis of three-fingered hands with a single palm and no wrist. They defined the degrees of freedom of the finger-object contact to synthesize all the topologies with full mobility and non-positive locked-joint mobility between palm and object. Based on this work, Lee and Tsai [2] undertook the structural synthesis of multi-fingered hands without wrist, with a single palm and identical type of finger contacts, with 3 to 7 fingers, to present an enumeration of *feasible* kinematic structures of mechanical hands.

Tischler, Samuel, and Hunt worked on the type synthesis for robotic hands with emphasis on the creation of a minimal-isomorphism list of kinematic chains [7]. They also considered a positive mobility between the ground and the grasped object and the connectivity between fingertips and the grasped object as selection criteria [8]. Their work imposes full-cycle mobility and restricts the results to full six-dof of mobility and point contact with friction for the finger-object contact. This contact is modelled as a spherical pair, realized as a 3R serial chain.

More recently, Özgür *et al.* [3] used the structural analogy between the palm-fingers-object system and a parallel robot consisting on a base-limbs-platform system (as was recognized before by [8] and [2]). They adapted the procedures developed by Gogu [1] for parallel robot manipulators and worked on the structural synthesis of robotic hands for given values of dexterity, mobility, overconstraint, and redundancy.

All this previous work in structural synthesis of hands is focused on mobility and related metrics for grasping and manipulation of hands.

This paper presents a method for the structural synthesis of general hands (allowing multiple branchings of the tree topology) for grasping, mobility, and free motion of the fingertips. This method combines the checking for solvability for the rigid-body guidance dimensional synthesis problem [6] with the computation of a desired mobility and force closure for the hand-object system, for a given number of fingertips. The method generates an isomorphism-free list of structural solutions with a labelling approach which can be considered similar to [7].

2 Hand, fingertip contacts, and hand-object representations

A multi-fingered hand is defined as a multi-body system with a common body - the wrist, which is a fundamental part of the hand manipulation- spanning several branches and ending in multiple end-effectors [5]. The kinematic chain of a multi-fingered hand has a tree topology that can be represented as rooted a tree graph [9], with the root vertex being fixed with respect to a reference system, see Fig. 1(b).

A more general hand also has several palms arbitrarily branched and can be called a multi-fingered, multi-palm hand. A *palm* is an intermediate link whose degree is ternary or above. A *branch* of the hand is defined as the series of joints connecting the root node to one of the end-effectors, or fingertip. They are the main elements whose motion or contact with the environment is being defined by the task, see Fig. 1(a). Hereafter, the *tree* of the hand will refer to the contracted tree of the hand obtained by replacing the binary links between two higher order links by an edge labeled with their connectivity, which is equivalent to the number of 1-DOF joints between them; see for example Fig. 1(c). Open hands with a hybrid topology can also be transformed into a contracted tree topology, adequate to perform its dimensional synthesis, by removing the internal loops [6].

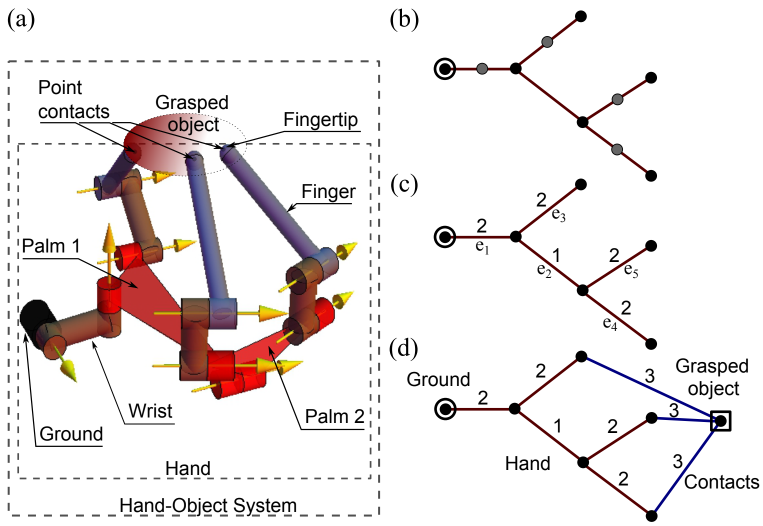


Fig. 1 A hand-object interaction and its graph representation. (a) A multi-fingered hand with 1-DOF revolute joints and 3 spherical fingertips. (b) The tree associated to the hand. Vertices filled in black are the root (circled), the palms, and the fingertip links; vertices filled in grey are binary links. (c) Contraction replaces the string of vertices with a unique edge labeled with its number of 1-DOF joints: 9 edges are contracted to 5 edges. (d) The contracted tree is connected by blue edges from each fingertip, each one labelled with the type (DOF) of contact, to the grasped object (squared vertex), to form the hand-object graph.

Two arrays are defined for the tree topology with n vertices and $e = n-1$ edges, which capture the incidence and adjacency properties as well as information of the edges. They are the *parent-pointer array* **p** and the *joint array* **j**. A labelling of the graph edges from 1 to e is assumed for the entries of both arrays. The parent-pointer array implements the parent-pointer representation of the tree. The first edges incident at the root vertex take the value zero.

The relative motion allowed between each fingertip and the object can geometrically be classified as proposed by Salisbury and Roth [4]. The connectivity c of the grasped object relative to the fingertip, which is denoted as the degrees of freedom at the contact, can take any value from 0 (rigidly attached) to 6 (no contact). The Table 1 summarizes the description of the contacts [4, 8].

Table 1 Contact types between a fingertip and a grasped object.

Degrees of freedom	Description
6	Free link (without contact)
5	Point contact without friction
4	Line contact without friction
3	Point contact with friction or plane contact without friction
2	Area of contact with friction (*Soft finger* [4])
1	Plane contact with friction
0	Rigid attachment to object

This work uses these contact types to extend the representation of the hand to the representation of the hand-object system. When an object is grasped, a loop is created in the graph of the hand-object system for any two fingers in contact with the object.

An additional *fingertip array* **c** that contains the type of contact between the fingertip and the object. Figure 1(d) depicts the graph of the tree topology of the hand grasping an object with all fingertips in contact. In this case, the corresponding parent-pointer, joint, and fingertip arrays are respectively $\mathbf{p} = \{0,1,1,2,2\}$, $\mathbf{j} = \{2,1,2,2,2\}$, and $\mathbf{c} = \{3,3,3\}$.

For dimensional synthesis purposes, the general Chevychev-Grübler-Kutzbach mobility criteria for the hand-object system [4] is preferred to the more accurate methods developed by [1] and used in [3], because the information on the relative positions of the axes is not available; assuming a general position of the axes is appropriate in the general design problem where no geometric constraint on the unknown axes is prescribed. Once the degree-of-freedom of the fingertips are defined, the grasping and manipulation tasks for a body with a known shape can be defined to dimensionally size each of the feasible hand topologies found by the following algorithm.

3 Type Synthesis Algorithm for Free-finger and Object-contact Tasks

The goal is to find all hand topologies that can be paired with the task for dimensional synthesis, given a set of user-defined restrictions. User-defined inputs are the number of positions of the task m, the number of end-effectors or branches b, the range $[e_{\min}, e_{\max}]$ for the total number of edges of the graph e, the types of allowed fingertips **c** (from Table 1), and the desired mobility M. The output is the set of topologies that (i) meet the solvability criterion subject to these requirements, and (ii) meet the constraints related to the mobility of any root to end-effector subgraph.

3.1 Mobility and Solvability Conditions

Given the tree of the hand T and its mobility M, any root-to-end-effector subgraph T_{sub} must satisfy a non-positive mobility M' if joints are locked and a mobility greater or equal than that of the overall tree.

$$\forall T_{\text{sub}} \in T : \qquad \begin{cases} M'(T_{\text{sub}}) \leq 0 \\ M(T_{\text{sub}}) \geq M. \end{cases} \tag{1}$$

The solvability criterion for the dimensional synthesis of a tree topology T passing through a number of m positions is the formula $m = s(T)$ proposed by Simo-Serra and Perez-Gracia [6], which also requires the analysis for each subtree T_{sub} of the graph. The tree is solvable iff

$$s(T_{\text{sub}}) \geq m \qquad \forall T_{\text{sub}} \in T \tag{2}$$

3.2 Variety

Tischler and Hunt [8] define the variety of a graph as the difference between its full mobility M and the minimum mobility of a subgraph containing a loop or set of loops, M_{min}, that is, $V = M - M_{min}$.

For the reduced and compacted tree graphs of the hand, all the loops contain the vertex corresponding to the grasped object. Imposing that the graphs have variety $V = 0$ ensures that the object has the desired degrees of freedom and that the locked-joints mobility is non-positive. This condition is imposed by identifying and checking the subgraphs created along the tree graph, starting at the root. Let the ternary or above vertices (palms) be labeled as p_i, and the subgraph starting at p_i in which all previous edges and vertices have been eliminated be T_{p_i}. The following condition is imposed:

$$M(T_{p_i}) \geq M, \quad i = 1, \ldots, p, \tag{3}$$

where p is the total number of palms in the hand.

3.3 Algorithmic Implementation

The algorithm is divided in three main steps. In Step 1, the algorithm searches all possible topologies which satisfy user inputs. Then, Step 2 checks the solvability of candidate topologies and keeps only the topologies that are solvable. Finally, the mobility of the solvable candidates is computed in Step 3 and those topologies that satisfy the user inputs are presented as final answers. The method is described in Algorithm 1.

Algorithm 1 Type Synthesis Algorithm for Free-finger and Object-contact Tasks

(1) Find all the possible topologies.
Inputs: number of positions (m), number of branches (b), number of edges (e)
Outputs: Parent Pointer Array and Joints Array.

(1.1) Find parent pointer array (p). Parent pointer array must have length of e and b branches.

(1.2) Find joint array. For each parent pointer array in step 1.1, construct all possible joint arrays which meet the input criteria

(2) Solvability check. For each pair of parent pointer array and joint array found in step 1, calculates the number of positions for the exact kinematic synthesis. If the number of positions obtained for the kinematic task of all subtrees is greater or equal than the number of positions for the overall tree, the tree is solvable.

(2.1) Find all root to end effectors subgraphs. A graph with b branches has $2^b - 1$ subgraphs. Calculate m for all subgraphs and compare with m for the overall tree.

(2.2) Remove common edges. Common edges are the edges which are contained in all branches. In this step, an algorithm finds all common edges and removes them.

(2.3) Change root to one of end effectors. When the root of the graph is changed the value of the parent pointer array and joint array should be updated. The algorithm updates them in two steps. There is a path between the previous root and the new root.

- First, the parent-pointer value of the edges that are connected to this path is updated.
- Second, the parent-pointer value for the edges which are in the path is updated.
- Other edges which are not in the path or does not connect to the path do not need to be updated because the parents of them did not change.

(2.4) Iterate steps 2.1 to 2.3. This part will be stopped when only two end-effectors remain.

(3) Check Mobility. The output of the step 2 are the possible topologies. In this step the algorithm verifies that the mobility of the topology is equal to that defined as input when the grasping loops are created adding the fingertip contact array **c** to the graph.

(3.1) Remove unused part and calculate mobility. Since some part of rigid body may not participate in the grasping process, the algorithm removes them. For finding the used part, the algorithm finds all the edges that are in the branches from root to the end-effectors which contribute in grasping. The other edges are unused and the value of -1 is assigned to each corresponding element of parent pointer array and joint array. Then, calculate mobility for the resulting topology. If it equals to the user input, it is one of possible solutions.

(3.2) Find Mobility for subgraphs. Using the algorithm proposed in step 2.1, find all the root to end-effector subgraphs and calculate mobility (M) and locked joint mobility (M') for them.

(3.3) Remove common edges (Palms). Using the algorithm proposed in step 2.2 remove palms.

(3.4) Calculate Mobility for the graph of part 3.3.
(3.5) Iterate step 3.3 and 3.4 until there is no common edge.

(3.6) Internal checks. If all the subgraphs fulfill the two following conditions, the topology is one of the solutions.

- $M'_{subgraph} \leq 0$
- $M_{subgraph} \geq M$

4 Results

The table below shows a binary hand in which the calculations are detailed for the overall mobility and in-palm mobility for different palms along the depth of the tree, removing first the wrist and then the depth-1 palm. For clarity, the solvability of this hand is calculated separately.

Table 2 Mobility calculations for a binary hand with four fingertips.

Topology	Parameter	Symbol	Value
$\mathbf{p}=\{0,1,1,2,2,3,3\}$ $\mathbf{j}=\{2,1,1,2,4,2,4\}$	Number of task positions	m	9
	Number of branches (fingertips)	b	4
	number of edges	e	7
	Type of fingertip contact	$\mathbf{c}$	$\{2,2,2,2\}$
	Mobility	M	6
	locked-joints mobility	M	-10
Subgraph 1 Remove Wrist $\mathbf{p}=\{0,0,1,1,2,2\}$ $\mathbf{j}=\{1,1,2,4,2,4\}$	Number of branches (fingertips)	b	4
	number of edges	e	6
	Type of fingertip contact	$\mathbf{c}$	$\{2,2,2,2\}$
	Mobility	M	4
	locked-joints mobility	M	-10
Subgraph 2 Remove Palm 1 $\mathbf{p}$=\{0,0\} $\mathbf{j}$=\{2,4\}	Number of branches (fingertips)	b	2
	number of edges	e	2
	Type of fingertip contact	$\mathbf{c}$	$\{2,2\}$
	Mobility	M	4
	locked-joints mobility	M	-2

For comparison, the input used in the type synthesis example of Tischler and Hunt [8] is used here in the first example below. For the second example, we compare the output to the results of Salisbury and Roth [4] but using soft fingers instead of pointy fingers with friction. The number of positions for the synthesis is chosen so that the number of joints in the hand candidates is similar to those in the references used.The input values for both examples are shown in Table 3.

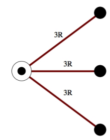
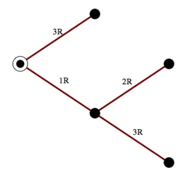
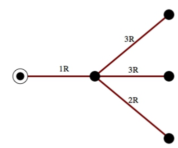
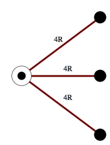
Table 3 Input values for the example

Parameter	Symbol	Example 1	Example 2
Number of task positions	m	5	9
Number of branches (fingertips)	b	3	3
Minimum and maximum number of edges	e	$(2,4)$	$(2,5)$
Type of fingertip contact	$\mathbf{c}$	$\{3,3,3\}$	$\{2,2,2\}$
Mobility	M	6	≥ 6

For the first example, the algorithm constructed 95 hand topologies and 10 of them were solvable. Out of those 10, only 3 topologies fulfilled the mobility requirements, that is, having $M = 6$ at the object with negative locked-joints mobility. The 3 topologies are shown in Table 4. Out of these topologies, two of them have a

1-dof wrist, which means that they have in-palm mobility equal to 5. The no-wristed hand obtained is the same that was obtained in the example from [8].

Table 4 Resulting topologies suited for the tasks of Examples 1 and 2

Example	Parent-pointer array	Joint array	Tree graph
Example 1	$\{0,0,0\}$	$\{3,3,3\}$	
	$\{0,0,1,1\}$	$\{1,3,2,3\}$	
	$\{0,1,1,1\}$	$\{1,2,3,3\}$	
Example 2	$\{0,0,0\}$	$\{4,4,4\}$	

For the second example, 295 topologies are compatible with the rigid-body guidance task, out of which 78 are solvable. However only one topology, the one corresponding to three 4-dof fingers and no wrist, has the required mobility $M = 6$ without being constrained by any subgraph, and negative locked-joints mobility. This topology corresponds to the solution chosen in [4]. In this case several other topologies had the required overall mobility, but the additional constraint of having the same or higher in-palm mobility from any palm discarded those other topologies.

The results clearly show that the obtained hand topologies are general. Salisbury and Roth as well as Lee and Tsai procedures leads to hands with serial chain fingers and a unique palm without wrist. Özgür methodology leads to serial and complex (chains with loops) parallel hand topologies analogous to parallel robots. Tischler *et al.* procedures have complex fingers with hybrid kinematic chains and produce topologies similar to the ones produced here for the case with a unique palm without wrist. Additionally, the tree topologies used here can be dimensioned through exact dimensional synthesis and when connected to the grasped object have serial, parallel, and hybrid topologies given more, or eventually new, design alternatives compared to those obtained in previous research.

The current algorithm also allows flexibility on where and when define the mobility. The current implementation imposes the same or higher mobility at each palm as that of the overall hand, but that can be modified to make some of the palms as grasping-only, for instance, while having different degrees of dexterity depending on the palm and fingers involved.

5 Conclusions

In this work, a structural synthesis procedure for general multi-fingered hands has been presented. The methodology considers the solvability of the hand for rigid-body guidance, and the mobility and locked-joints mobility when grasping an object, including the selection of the fingertips involved in the grasping and manipulation action. This procedure yields an isomorphism-free enumeration for compacted and reduced tree graphs. The presented examples show the adequacy of the methodology as a first step in the selection of a hand structure for a given general task that may include free-finger motion, grasping, and manipulation of the grasped objects.

Acknowledgements The first and second authors' work is supported by the National Science Foundation under Grant No. 1208385. The content is solely the authors' responsibility. The third author acknowledges to Argentinean projects, CONICET PIP 2011-01105, ANPCyT PICT 2013-2894, and UTN PID 3935.

References

1. Gogu, G.: Structural Synthesis of Parallel Robots. Part 1: Methodology. Solid Mechanics and Its Applications. Springer, Netherlands (2008)
2. Lee, J., Tsai, L.: Structural synthesis of multi-fingered hands. Journal of Mechanical Design **124**, 272–276 (2002)
3. Ozgur, E., Gogu, G., Mezouar, Y.: Structural synthesis of dexterous hands. In: Intelligent Robots and Systems (IROS) Conference (2014)
4. Salisbury, J., Roth, B.: Kinematic and force analysis of articulated mechanical hands. Journal of Mechanisms, Transmissions and Automation in Design **105(1)**, 35–41 (1983)
5. Selig, J.M.: Geometric Fundamentals of Robotics (Monographs in Computer Science). SpringerVerlag (2004)
6. Simo-Serra, E., Perez-Gracia, A.: Kinematic synthesis using tree topologies. Mechanism and Machine Theory **72 C**, 94–113 (2014)
7. Tischler, C., Samuel, A., Hunt, K.: Kinematic chains for robot hands - i. orderly number-synthesis. Mechanism and Machine Theory **30(8)**, 1193–1215 (1995)
8. Tischler, C., Samuel, A., Hunt, K.: Kinematic chains for robot hands - ii. kinematic constraints, classification, connectivity and actuation. Mechanism and Machine Theory **30(8)**, 1217–1239 (1995)
9. Tsai, L.W.: Mechanism Design: Enumeration of Kinematic Structures According to Function. CRC Press, Boca Raton (2001)