Review of Research in Dexterous Manipulation

Technical Report

Project: DPI2008-02647

Authors:
Juan Antonio Corrales Ramón
Fernando Torres Medina
1.- Introduction

The main goal of this technical report is enumerate the main research papers which have been developed in the field of dexterous manipulation. In particular, this review will focus on the description of planning methods which permit to reconfigure an object (i.e. change its position and orientation) with movements of the fingers of a multi-fingered hand.

First of all, the next section enumerates the main papers which describe in a general way the problem of dexterous manipulation. Next, in the third section of this report, a list of different planning strategies to solve this problem is presented. Each important research paper is described in a different subsection and its main elements are summarized.

2.- The Dexterous Manipulation Problem

2.1.- Dextrous Manipulation by Rolling and Finger Gaiting [1]

There are three types of manipulation tasks for multifingered hand systems:

- **Object Manipulation**: Achieve the desired object configuration without regard for contact configurations.
- **Grasp Adjustment**: Obtain the desired contact configurations without regard for the object configuration.
- **Dexterous Manipulation**: Achieve the goal configuration for the object and contacts.

**Finger Gaiting**: Periodic sequence of finger relocations to form a new grasp consisted of fingers lying within their workspace limits.

To implement finger gaiting with three fingers, two fingers need to form a FC grasp. Then a necessary condition for using gaiting with three fingers is that at least one of the grasping fingers can form a force closure grasp with other two fingers of the hand. Clearly if neither of the grasping fingers can form a FC grasp with the third finger, then none of them can be lifted and rolling must be used to reevaluate the finger(s).
They identify two finger gaiting primitives:

- **Finger Rewind:** In finger rewind, an additional finger F3 is used when F1 and F2 reach their limits. Therefore, the finger F3 is used with each finger F1 and F2 in order to relocate each finger inside its workspace. The basic procedure is to rewind the limiting fingers back to their workspace.

- **Finger substitution:** In finger substitution an additional finger F3 substitutes other finger which has reached its limits. The limiting finger is lifted up and becomes free finger and will be no need to rewind it back.

Simulation with a sphere and a particularization of force closure constraints has been performed. The general framework was applied to the problem of dexterous manipulation of a sphere with 3 hemi-spherical fingertips.

### 2.2.- On Motion Planning for Dexterous Manipulation, Part I: The Problem Formation [2]

In the study of multifingered robot hands, the procedure of adjusting grasp configurations without risk of dropping the object is called dexterous manipulation. In order to perform such a manipulation, the robot hand has to rely on rolling contact, sliding contact as well as on finger relocation. Since rolling constraints are in general non-holonomic, this makes a hand manipulation system a non-integrable system.

The configuration space $P$ of the hand manipulation system is:

$$P = SE_O(3) \times SE_1(3) \times \ldots \times SE_k(3)$$

(1)

Where a configuration $z \in P$ is of the form $=(g_0, g_1, \ldots, g_k)$, with $(g_0 \in SE_O(3))$ (object configuration) and $(g_i \in SE_i(3))$ (finger $i$ configuration). A point $z$ in $P$ is also called a **grasp configuration**.

Let $M = SE_1(3) \times \ldots \times SE_k(3)$ denote the configuration space of the fingers. Since finger motion can be controlled by actuators located at the finger joints, $M$ is also called the **control space**.
Problem Statement: Given two grasp configurations $z_1, z_2 \in P$ construct an admissible piece-wise continuous curve $\gamma(t), t \in [0, t_f]$, in $M$ such that $\gamma$ can be lifted to a curve $\alpha$ in $P$, i.e., $\pi(\alpha) = \gamma$ and $\alpha$ connects $z_1$ to $z_2$, i.e., $\alpha(0) = z_1$ and $\alpha(t_f) = z_2$.

Let $(v_x^i, v_y^i, v_z^i)$ and $(w_x^i, w_y^i, w_z^i)$ denote the components of the instantaneous contact velocity of finger $i$ relative to the object.

Definition: They say that finger $i$ contacts the object by:

a) Fixed point of contact if $v_x^i = v_y^i = v_z^i = 0$ and $w_x^i = w_y^i = w_z^i = 0$.

b) Rolling contact if $v_x^i = v_y^i = v_z^i = 0$ and $w_x^i = 0$.

c) Sliding contact if $v_x^i = 0$ and $w_x^i = w_y^i = w_z^i = 0$.

2.2.1.- Classification of Motion Constraints

In this section, they classify the set of basic constraints on finger motion in $M$ according to two categories: (i) constraints for collision avoidance and (ii) constraints by the kinematic structures of the fingers.

Constraints for Collision Avoidance

During the course of manipulation, collisions between links of all $k$-fingers should be prevented. Collision between finger $i$ and finger $j$ can be prevented if and only if:

$$d(F_i, F_j) > 0$$

(2)

Where $F_i$ and $F_j$ represent the fingers configurations and function $d$ returns the minimum distance between them.

The subspace of $M$ where finger $i$ is collision free with finger $j$ is denoted by $d(F_i, F_j)^{-1}((0, \infty))$ and the constraint subspace for collision avoidance of all $k$-fingers is the intersection:

$$\bigcap_{i<j} d(F_i, F_j)^{-1}((0, \infty)) \subset M$$

(3)
Constraints by Finger Kinematics
The second type of basic constraint is the constraint due to the finger kinematic structures. Since the last link is connected to the hand palm by \( n_i \) links, the set of reachable configurations by the finger is a compact sub-manifold \( Q_i \subset SE(3) \). The subspace of \( M \) where finger kinematic constraints are satisfied is the product:

\[
(Q_1 \times ... \times Q_k) \subset M
\]

Finally, the subspace of \( M \) where all the constraints discussed in this section are satisfied is given by:

\[
Q_s = \left\{ \bigcap_{i<j} d\left(F_i, F_j\right)^{-1}\{(0,\infty)\} \subset M \right\} \cap \{(Q_1 \times ... \times Q_k) \subset M \}
\]

2.2.2.- Manipulation Modes
They will decompose dexterous manipulation into the following manipulation modes: (i) coordinated manipulation, (ii) rolling motion, (iii) sliding motion and (iv) finger relocation. The time interval it takes to reach the initial state to the final state is divided into the union of successive subintervals such that at each subinterval the finger motion is in one of the manipulation modes.

The first manipulation modes correspond to three different types of contact types between the object and the finger:

- **Coordinated manipulation** is performed by fixed contacts: \( (w_x^i = w_y^i = 0) \).
- **Rolling motion** is performed by rolling contacts: \( w_x^i = 0 \).
- **Sliding motion** is performed by sliding contacts: \( \nu_x^i = 0 \) and \( w_x^i = w_y^i = w_z^i = 0 \).

Not only the contact points should fulfill the previous conditions but they should also form a grasp for all the steps of the fingers’ trajectories.

In a **finger relocation** mode, a group of \( m \) fingers \( (1 \leq m \leq k) \) are allowed to break contacts with the objects and they will be positioned at other locations, provided that the set of contact points by the remaining fingers still forms a grasp.
3.- Dexterous Manipulation Strategies for Object Reconfiguration

3.1.- Planning Quasi-Static Fingertip Manipulations for Reconfiguring Objects [3]

3.1.1.- The Reconfiguration Problem

A simple statement of the reconfiguration planning problem is: starting from a given initial grasp of an object $O$ (defined by the configurations of $O$ and of the fingertips), find feasible trajectories (motions and contact forces) for the fingertips to move $O$ to a desired final configuration. They describe a motion planner that solves the reconfiguration problem.

The problem is to find a sequence of pure rolling and/or pure sliding motions of fingers $F_i$’s performing a quasi-static reconfiguration of $O$ while satisfying the contact kinematics and equilibrium of the system, avoiding collisions between the $F_i$’s and making sure each lies in its workspace.

For illustration, consider the reconfiguration of an ellipsoid by three simplified (spherical) fingertips.

Assumptions: For simplicity, they require that all fingers $F_i$’s are always moving in contact with $O$ and they do not allow finger relocation by breaking contacts. They also assume that the $F_i$’s and $O$ are smooth, convex and rigid, resulting in point contacts. Furthermore, they restrict the contacts to be either pure rolling or pure sliding. They consider quasistatic analysis in predicting the motion of the system since the system moves at low velocities and the dynamic effects can be neglected.

A two-level planning scheme has been developed. The first level (or global planner) is a grid search algorithm that globally explores the C-space of $O$. It expands a tree of subgoals (intermediate configurations of $O$) through which $O$ can be possibly manipulated. The second level (or local planner) operates locally and plans for feasible quasistatic motions of the entire fingertips-object system between couples of adjacent subgoals. It makes use of a
fine discretization of time and solves, at each instant, the inverse problem with the velocity of \( O \) provided by the global level.

### 3.1.2. The Global Planner

The global level (an A* type search) searches a graph \( G \) over a discrete representation of \( CS_O \) (configuration space of the object) defined by a collection, \( CC \), of 6-dimensional regular and non-overlapping cells, and it uses the local level to find feasible motions between adjacent cells. \( CC \) is defined on \( TS \times RS \), where \( TS \subseteq \mathbb{R}^3 \) is a finite domain of the positions of \( O \) (of the order of the size of the hand) and \( RS \) stands for the domain of the yaw/roll/pitch angles parameterizing locally the orientation of \( O \).

The global planner starts from the initial object configuration \( g_{O,\text{start}} \) and generates a set of configurations (subgoals) located in cells adjacent to \( g_{O,\text{start}} \) by using canonical paths, computed in the absence of any constraints. Afterwards, a subsequent subgoal \( g_{O,\text{current}} \) is chosen according to a cost function based on the distance to the goal configuration. This subgoal is processed by the local planner in order to verify if it is feasible according to the manipulation constraints. If the subgoal is feasible, it will be added to the final path. If the subgoal is not feasible, the following subgoal adjacent to the previous node of the graph will be evaluated.

### 3.1.3. The Local Planner

They now present the local planning algorithm that checks if a selected subgoal \( g_{O,\text{current}} \) is reachable through dexterous motions from a configuration \( g_{O,\text{previous}} \) reached in the parent cell in \( G \).

The local planner selects randomly the contact mode (pure rolling contact or pure sliding contact) for each contact point and the corresponding contact velocities from the following set:
The contact velocity vector $\dot{q}_r$, also known as relative controls, is process by the instantaneous motion function. The instantaneous inverse problem involves: given a velocity vector $\dot{q}_O$ of $O$ and the relative controls $\dot{q}_r$ at the contacts, compute the resulting collision-free and reachable grasp and the contact forces to be applied on $O$ to keep it in equilibrium. The instantaneous motion routine performs the following steps:

- It computes the coordinates of the new contact points by applying $\dot{q}_r$ to the Montana’s equations (kinematics of contact).
- It computes the new object configuration $g_O$ by moving $O$ by $\dot{q}_O$.
- It computes the new finger configuration $g_{F_i}$ by taking into account that the fingertips must touch the object in the new contact points.
- It determines if the grasp is is reachable by the fingers and is collision-free.
- It computes the contact forces by solving a linear programming problem which minimizes the instantaneous power and considers the equilibrium and the contact constraints (pure rolling or pure sliding).

### 3.2.- Global Planning for Dexterous Reorientation of Rigid Objects: Finger Tracking with Rolling and Sliding [4]

It presents a two-level planner similar to the planner described in the previous section. Nevertheless, this planner is based in finger tracking. Finger tracking consists, at each instant, of fixing three fingertips with regard to the palm and pushing the object through the motion of the fourth fingertip. They extend the original finger tracking of Rus to smooth objects, incorporate contact kinematics and adapt it to quasi-static manipulation motions, which accounts for friction and reachability of fingertips.
An important simplification can be considered for the manipulation of convex polyhedron: If the positions of the fingertips are known and located on three faces whose normals are not coplanar, a local diffeomorphism (correspondence) can be defined, at the current orientation, between the object space $SE(3)$ and $SO(3)$. This means that for any orientation in the neighborhood of the current orientation, there is a unique object position corresponding to it in the workspace. This important result enables us to locally reduce the search for a valid movement of the object to a search in a three-dimensional space.

Let $O$ be a convex polyhedron manipulated by $d + 1$ spherical fingertips $F_i$ (with $d = 2, 3$). They assume that at each instant, only one fingertip can be moved on the surface of $O$ when the $d$ other are fixed. Let $g$ be a given grasp (including the orientation $R_o$ of $O$). If a local diffeomorphism can be defined at $R_o$, then finger tracking (combined with reassigning the moving fingertip and adjusting the set of fixed fingertips) can be used to build a collection made of a finite number of charts, $S_j$, which is locally diffeomorphic to $CS(d)$. In other words, any configuration of the object can be obtained with finger tracking and changing the moving fingertip.

The manipulation planner is identical to the one developed in the previous section except in the fact that the local planner has been modified in order to perform finger tracking. In particular, not only the contact velocities are generated randomly but also the selection of the moving finger. In addition, they constrain, at each instant, two fingertips (a fixed one and the moving one) to be rolling without sliding. In particular, they constrain the moving finger to be purely rolling, and they allow twisting at the fixed rolling fingertip. For the other fixed fingertips, they do not impose any specific contact mode. These new constraints are integrated in the local planner.

The Montana’s equations have been extended for polyhedral objects in order to allow fingertip motion between adjacent faces. The developed method is applied to polyhedral and smooth objects.

3.3.1.- Problem Formulation

To our opinion, the drawbacks of existing methods are of two kinds. Firstly, many methods compute the object trajectory first and then the trajectories of the fingers. As the object trajectory depends strongly on the accessibility domains of the fingers, such methods may not find a solution in many situations. The other drawback is that some methods explore the configuration space at a too low level, having more a control approach than a motion planning approach. As the configuration space dimension of a dexterous manipulation system is particularly big, this leads to huge computation times. Therefore the associate shown examples are always very simplistic (sphere or egg-shaped object small reorientation). The method they propose in this paper aims to solve these weaknesses taking into account the particular structure of the configuration space.

They introduce a fundamental subspace called Grasp Subspace $k$ ($GS_k$), that is the subspace of all the configurations $q$ with $k$ grasping fingers and $(n - k)$ independent fingers. It is not always possible to link two configurations belonging to the same $GS_k$, whatever path type is chosen. It is then necessary to use a path in $GS_{k+1}$ or in $GS_{k-1} \setminus GS_k$ ($GS_{k-1}$ without $GS_k$). The need for subspace change is equivalent to the need for finger gaiting (i.e. to perform finger relocation).

**Constraints:** Among all existing stability criteria, they choose the force closure one that is certainly the most commonly used. Another important constraint concerns the kinematic of contacts. Indeed, they assume that the movement of the object is induced by the movement of the fingers and that the contacts between the object and the fingertips cannot slide on the object surface. This lead us to introduce two fundamental local paths, each one corresponding to an elementary manipulation subtask: regrasping path and transfer path. During a regrasping path, the object is maintained immobile and some fingers move to change the grasp, while during a transfer path, the object is moved but the grasp (contact points) remains unchanged. The goal of the Dexterous Manipulation Planning (DMP) is
then to find a sequence of transfer and regrasping paths that will link two given configurations both belonging to $GS_n$ while ensuring grasp stability (i.e. force closure) all along.

### 3.3.2. Proposed Planning Method

To find a feasible path linking two configurations in $GS_n$, one needs to explore the system $C$-space with an exploration algorithm. Let us take the example of a 3D four-fingered hand to illustrate the search space structure and our exploration algorithm. If $GS_4$ has multiple connected components because of one or more obstacles (obstacles can be joint limits, grasp instabilities or collisions between bodies), paths in $GS_4$ are not enough. One needs to use paths in $GS_3 \setminus GS_4$ subspace. These paths are regrasping paths (used with transfer paths).

The principle of our DMP technique is to explore $GS_4$ with paths inside this subspace, in order to build a graph. If $GS_4$ has multiple connected components, the graph exploring $GS_4$ will have multiple components too and paths in the $GS_4^i, i \in [1, 4]$ will be used to merge them (transfer-regrasping paths). The obtained grasp is the **dexterous manipulation graph**.

The manipulation graph is built alternating two steps:

- $GS_4$ exploring: The goal of this function is to build a graph in $GS_4$ in order to capture this subspace topology. Exploring $GS_4$ in such a way is a motion planning problem for a system containing several closed kinematic loops. One needs to generate configurations verifying chain closures. To solve this problem, they use RLG (Random Loop Generator) algorithm. Each chain is divided into an active part (the object) and a passive one (the fingers). The active part configuration is randomly chosen in the accessibility domain of the passive part. The passive part is calculated using inverse geometric models. The grasp stability (force closure property) is checked for every generated configuration along a path. The function tries to link the nodes of the graph (configurations in $GS_4$) with linear paths in $GS_4$. Along such paths, the object moves while the fingers are sliding on the surface of the object, in the same
time. One just needs to randomly choose points on the object surface and to compute continuous shortest paths on this surface, linking two of these points. A solution is to approximate the surface by a polyhedron and to use a geodesic computation algorithm which find the shortest path between two given object surface points.

- **Merging its connected components with transfer-regrasping paths**: This function tries to merge two different connected components of the manipulation graph using transfer-regrasping paths. The transfer path goal is to bring the object configuration from its initial to its goal configuration. The goal of the regrasping path is then to bring the hand to its final configuration:
  
  - **Transfer Path Computation**: Transfer paths are object movements realized by rolling fingertips on its surface. To compute them, knowing the object trajectory, they need to find finger movements satisfying the constraints of rolling contacts. Once a contact is known and verifies the appropriate geometric constraints (it must belong to both object and fingertip surfaces, the normal vectors of these two surfaces must have the same direction and the contact point must be reachable by the finger), it must satisfy two kinematic constraints: the relative velocity of the object and the fingertip at contact point must be null and the contact point velocity must conform with the associate finger kinematic model. Different methods exist to compute trajectories verifying these constraints. However, they all require a surface parameterization of both object and fingertip. Instead they chose to integrate the constraints numerically. Finger velocities are computed so as to ensure the nullity of the relative velocity of the two contacting bodies, while the two surfaces are constrained to not inter-penetrate (surface distance is computed using polyhedron collision detection techniques).

  - **Resgrasping path computation**: To compute the regrasping paths in a $G^\text{f}_{S}$, a collision free trajectory for the free finger has to be planned. This is simply done using the RRT (Rapidly-exploring Random Tree) method.

As in [3, 4], the planner algorithm consists of two main parts: the global and the local planner.

The task of the global planner is to search for the nominal path in the configuration space among static obstacles between the initial and the desired position and orientation of the object. A graph search algorithm (such as A*) is used to generate points (subgoals) in the object’s quantized configuration space, the local planner tries to find admissible motion trajectories of the fingertips in contact with the object between the subgoals.

The local planner includes four parts: (a) the kinematics solution (computes the trajectories); (b) the force computing algorithm (solves for the contact forces) based on LP (Linear Programming) and LMI (Linear Matrix Inequalities); (c) the generation of the relative velocities between the fingertips and the object; (d) collision detection algorithm. The relative velocities are chosen automatically by a Simulated Annealing algorithm. The energy function of the Simulated Annealing algorithm has several components: (a) No collision of fingers; (b) force equilibrium (applied on the object); (c) the absolute sum of contact forces (in order to minimize the energy); (d) minimize the number of relative velocity changes.

3.5.- On Fitted Stratified and Semi-Stratified Geometric Manipulation Planning with Fingertip Relocations [7]

They assume throughout the paper that the contact points are fixed, i.e., that sliding and rolling are not permitted. At the same time, they allow finger relocations from a point of the surface onto another one. Our paper approaches the manipulation problem from the point of view of open loop control which leads to a motion planning problem (MPP) with constraints.

Stratified motion planning [8] offers a general approach for the system whose (smooth) equations may change in the configuration space. This method combines the object manipulation and the grasp adjustment tasks into a unified dextrous manipulation problem.
The key element of the method is to divide the configuration space into smooth submanifolds (strata) where, in each stratum, a different smooth nonlinear system is valid. The method compounds these systems into a smooth common system later called the bottom stratified extended system where, with some restrictions, smooth motion planning (MP) can be applied. Stratified MPA (MP Algorithm) delivers a unified dexterous manipulation concept to solve this manipulation task. However, the approach has some drawbacks, where (sometimes very hard) symbolic computational difficulties are the most noteworthy [8]. In addition, it is also hard to interpret the resulting trajectory.

Two new manipulation algorithms are proposed by adopting stratified MPA but their foundation is a simple fictitious (fitted) system that reduces the complexity of the computations to almost pure numerical procedures. They also let one interpret the state trajectories easily from the results in the configuration space.

The first proposed method based on the philosophy of stratified motion planning [7] uses a special fictitious system called the fitted system. The special parameterization of the fitted system yields simple vector fields where one also can easily check any system’s property (e.g. stratified controllability). This method is able to carry out a dexterous manipulation in a restricted workspace of fingertips. However, the method does not ensure automatically force closure stability and finger collisions.

The second proposed method is a semi-stratified motion planning on a fitted system using task decomposition [7]. Beside stratified MP, this semi-stratified motion planning includes also a strategy for systematic finger relocations. The finger relocations are based on a subsegment generation procedure, which provides suitable chosen reference fingertip positions to a desired object motion. Hence, the method restricts the fingertip positions (i.e., arbitrary fingertip positions on the object cannot be achieved, causing dexterity in manipulation to fail), hence it aims primarily at object manipulation. In return, it is able to guarantee force closure stability and collision avoidance. Additionally, it provides a greater degree of freedom in finger relocation than fitted stratified manipulation because it allows any trajectory in the free space for the fingertips.
The discussion of singularities is not a subject of this paper, hence one may insert the methods below into a global manipulation planning where singularities, collision avoidance, and force closure are investigated in a higher level [6].

The control methods presented for smooth systems show a major challenge in the general MPP because different strata are described by different equations of motion. The systems defined on the separated strata are typically not controllable. The idea of stratified control is to define a common state space where all the vector fields can be considered from all the strata. This usually is associated with the bottom stratum because the typical initial and final configurations lie in this stratum.

(Stratified control concept.) Consider a finger gaiting system with two fingers. Using the convention of the notation for strata, S0 symbolizes the total configuration space, and S1 stands for the stratum where finger 1 is in contact with the object. Similarly, S2 represents the stratum where finger 2 contacts the object. The most important stratum is the bottom stratum S12 where both fingers touch the object. Let the initial and the desired final points lie in the bottom stratum (in accordance with a typical manipulation problem). If one uses the fixed contact points model then the system will not be controllable in the bottom stratum S12. In other words, one cannot solve a general manipulation task on S12 since every fingertip must be fixed to the surface in this stratum. However, the whole system can be stratified controllable and manipulatable if one puts into use the vector fields from the higher strata allowing the system to move along an extra direction (in higher stratum). This results in the physical epiphenomenon of finger gaiting. The manipulation process is composed of a sequence of flows where vector field g1,1 moves the system off from S12 onto S1 (finger 2 disconnects the object), vector field g2,1 moves the system off from S12 onto S2 (finger 1 disconnects the object), g1,2 is defined on stratum S1, and g2,2 is defined on stratum S2.

The main complication appeared in earlier stratified MPA [8] when the Hall coordinates are computed. It proceeds via evaluation of recursive integrations. If the vector fields of the bottom stratified system have symbolically complex vector fields, the integrand becomes complicated. However, this is not the case for the fitted stratified system because the vector
fields do not contain symbolic variables. It makes the computations very easy, almost entirely numerical.

Unfortunately, stratified manipulation does not necessarily keep the fingers directly in their own workspace for arbitrary manipulation. The semi-stratified manipulation planning in the next section is devoted to handling this problem.

They propose a decomposed manipulation concept for an object with smooth surfaces that combines fitted stratified motion planning with unconstrained motion planning. In this context, unconstrained motion means that a finger can move in the free space between two points independently of the object. The motivation to use unconstrained MP beside stratified manipulation may be useful for more reasons. One of them is to give a greater degree of freedom for finger relocations in manipulation planning. Second, it may be desired to dispose of one part of the complex computations appearing for instance in computation of Hall basis. Additionally, they would like to keep the fingers in their own workspaces independent of the object orientation.

At first, one fixes the contact points to the object and moves the object along the reference object trajectory. Before one of the fingertips is about to leave its quadrant (workspace), they record the object and fingertip configurations. This configuration will be the desired final configuration for the point to point fitted stratified MP. This configuration is reached by a sequence of flows (35). At this point, they insert four unconstrained finger relocations that “pull back” the fingertips into the plane $z = 0$ where $x = y$ and $x = -y$. The fingers are relocated in order, i.e., only one of them moves back into the contact $z = 0$, $x = y$ or $x = -y$ at a time, the other three remain on the surface. After this, the following finger is relocated onto another contact point and the other three ones remain in contact with the object. While the fingertip is being relocated, it breaks the contact, moves in the free space and establishes a new contact in the above mentioned location. It can be shown [8] that this kind of motion of the fingertips is able to ensure force closure stability. As a matter of fact, the unconstrained finger relocations are new features in fitted semi-stratified MP since it did not appear in fitted stratified MP.
It is worth concluding that fitted semi-stratified manipulation with its more global considerations (finger relocation strategy) belongs rather to a slightly higher level in the manipulation than the fitted stratified approach. It comes from the insertion of an extra phase consisting of four unconstrained finger relocations. These unconstrained finger relocations were separated from the fitted stratified manipulation. They substitute systematically one part (namely the finger relocation part) of the fitted stratified approach. As a result, one may ensure force closure stability, collision avoidance. However, one should choose the reference contact points in a restricted way (finger relocation when fingers are about to leave their workspaces). It results that the fitted semi-stratified manipulation accomplishes rather an object manipulation than a dexterous manipulation.

The unconstrained finger relocations in the semi-stratified approach realize a second phase in the manipulation after the fitted stratified manipulation is executed. It accomplishes a finger relocation algorithm with a greater freedom than the stratified approach. The finger relocations necessitate extra computations but they increase the degree of freedom in the fingertip motions. Namely, a finger can be relocated along an arbitrary path above the object while a fitted stratified approach should move the fingertip along only one certain direction above the object at a time.

Their other intention is to extend the method for object manipulation where the object surface is not smooth but has edges.

3.6.- Sampling-based Finger Gaits Planning for Multifingered Robotic Hand [9]

Two approaches: Firstly, Hybrid Automaton (“Hybrid Automaton: A Better Model of Finger Gaits” and “A Kinematic Model of Finger Gaits by Multifingered Hand as Hybrid Automaton”); Secondly, include RRT based planner to find suitable finger substitution (“Finger Gaits Planning for Multifingered Manipulation” and “Sampling-Based Finger Gaits Planning for Multifingered Robotic Hand”).

When a large-scale manipulation is performed, some manipulation requirements, for example, the workspace limitation or the grasping force requirement, sometimes make it impossible for the hand to complete the manipulation task even with rolling and sliding
motions of the fingers. When such situation arises, the hand has to change its grasp status and relocate some fingers in order to continue the manipulation. A strategy composing of both the manipulation of the object and the relocation of the fingers is called a finger gait.

In order to manipulate the object toward its desired configuration without failing any grasping requirement, a robotic hand will sequentially change its grasp status during a finger gait. These changes of grasp status divide the entire manipulation into a sequence of submanipulations corresponding to a sequence of time intervals. On the other hand, each individual submanipulation is still continuous within its corresponding time interval. For this reason, a model with both discrete and continuous characteristics is in need to represent a manipulation with finger gaits. A hybrid automaton in such a dynamical system that interacts both continuous-time and discrete-event dynamics. It describes the evolution of a set of discrete and continuous variables.

They make the following simplifying assumptions:

- The PCWF (Point Contact with Friction) is used between the fingertip and the object. Each fingertip is modeled as an isolated point, and there is no rolling and sliding motions at all points of contacts.
- The object's boundary is a smooth surface.
- No collision between the fingers and the object is considered.

As described in its refined strategy, two types of submanipulations get involved in a manipulation with finger substitution:

1. **Manipulation mode**, denoted by $q_m$: In this mode, $f$ is used as the free finger and stays at a prescribed configuration without any motion. In the meantime, the other fingers are used as grasping fingers to manipulate the object toward its desired configuration.

2. **Substitution mode**, denoted by $q_s$: In this mode, $f$ is used as the free finger to move off or on the object to execute the substitution. The other fingers are used as grasping fingers to maintain the stable grasp and hold the object’s configuration.

In either mode of submanipulation, only one of the fingers is used as the free finger. Therefore, a set of $2k$ discrete variables $Q$ which describes all modes of submanipulations in a $k$-fingered finger substitution gait.
The continuous variables $X$ needed to represent a finger substitution gait are the position/orientation of the object and the positions of all the fingers. The state of the hybrid automaton is composed by a collection of discrete variables $Q$ and a collection of continuous variables $X$.

Constrains which are considered in the domains of the variables of the gait automaton:

- Workspace constraint: No finger runs out of its workspace. This is one of the main reasons that finger gaits are needed in a large-scale manipulation.
- Force-Closure constraint: One basic requirement for a stable grasp is that the hand must be able to balance any external wrench applied to the object.
- Contact constraint: All the fingers must satisfy that there are over the surface of the object.

Since the object is manipulated by different grasping fingers in different modes of submanipulations, evolutions of the continuous states in each mode are different:

- Manipulation mode: In the manipulation mode $q_m$, the object is manipulated by the grasping fingers with a desired velocity. The velocity of the free finger is zero since it holds its position and waits for the substitution.
- Substitution mode: In the substitution mode $q_s$, the grasping fingers maintain a stable grasp and wait for the free finger to execute the substitution.

The evolutions of the discrete states are represented as a set of edges of the finger substitution automaton. All possible switches between different modes of submanipulations form these edges. Three possible switch situations can happen:

- Case 1 (switch from manipulation mode to substitution mode): one of the grasping fingers reaches the configuration to switch, and the free finger can form a force closure grasp with the other grasping fingers.
- Case 2 (switch from substitution mode to another substitution mode): the free finger contact the object and form a force closure grasp with other grasping fingers except the one which will be substituted.
- Case 3 (switch from substitution mode to manipulation mode): the free finger was relocated into a pre-described safe area.
With the hybrid automaton model, they do the simulation using the three planar fingers manipulating a planar circular object with finger substitution. Each finger has 2 degrees of freedom and the reachable workspace of each finger is considered as the workspace limit. Using the hybrid automaton representation, the finger planning problem is formulated into a standard motion planning problem over a hybrid configuration space with both continuous and discrete evolutions. With the rapidly-exploring random tree (RRT) structure, a finger gait planner is developed to quickly search the entire hybrid state space for a feasible gait strategy to accomplish the manipulation task. A refined sampling method, a more efficient exploring strategy and a novel hybrid metric are developed to improve the performance of the RRT based planner.

By extending the basic construction strategy of RRT with the consideration of the hybrid characteristic of the problem, they developed a specific RRT based planner for autonomously planning a feasible finger substitution gait. Since no sliding and rolling motion of contacts is considered in the entire finger gait as introduced in Assumption 2, the grasp G can be fully determined by the continuous state X. In other words, in an individual discrete mode q, any continuous evolution of the hybrid automaton for a finite time interval with such a G is constrained in a subspace M. This submanifold M is a projection of the original configuration manifold S restricted by the task constraints. For this reason, not every Srand generated using random sampling method in classic RRT is reachable by the random tree until new sub-manifolds are explored. Then they propose a new sampling strategy in the gait planner.

As in any other RRT based planner, to find the nearest neighbor of a random sample, qnear, and select the appropriate control input to extend the tree from it, a distance function between two different hybrid states has to be defined. To both efficiently and exhaustively explore the hybrid nature of the finger substitution automaton, they propose a new distance function dH defined over its hybrid state space, which is a complicated integration of the discrete distance dQ : SG ×SG →R and the continuous distance dX : SG ×SG →R.
4.- References


