

## OBJECT MANIPULATION WITH DEXTROUS ROBOTIC HAND USING SIMULATED ANNEALING

Gábor VASS \* Shahram PAYANDEH\*\* Béla LANTOS \*

\*Budapest University of Technology and Economics  
Department of Control Engineering and Information Technology  
H-1117 Budapest, Pázmány Péter sétány 1/D. Hungary

\*Simon Fraser University,  
Experimental Robotics Laboratory, School of Engineering Science  
8888 University Drive, Burnaby, British Columbia, Canada V5A 1S6

Submitted: November 10, 1999

### Abstract

An algorithm of object manipulation with multifingered dextrous hand is presented. The manipulation task (called object re-configuration problem) is stated as the following: given an initial grasp of the object find the motion's trajectories of the fingertips to move the object to the desired configuration. The forces exerted at the contact points by the fingertips are determined first to ensure a stable grasp, then to manipulate the object. The objective of this paper is to develop a framework of a manipulation planner in the presence of obstacle constraints between the initial and goal configuration of the object. In this approach, fingertips can loose and regain contact with the object at some other planned next location. The controller devises a strategy where the environmental constraints (e.g. obstacles) can also be used as passive contact surfaces supporting the manipulation of the object. The algorithm can also be applied to the design of object reconfiguration with multiple agents including cooperating robots.

*Keywords:* object reconfiguration, dextrous hand, simulated annealing.

### 1. Introduction

The fingertips of a dextrous robotic hand can be treated as manipulating agents with motion constraints, thus object manipulation algorithms with multiple agents can be applied to dextrous robotic hands [1]. Recently numerous object manipulation algorithms have been developed. An algorithm for four-finger equilibrium and force-closure grasps of polyhedral objects is proposed by [3], another algorithm for dextrous grasping force optimization is given by [4]. The topic of object manipulation with agents or robotic hands can be classified by the mode of the relative motion of the agents and the object. Pushing, rolling and sliding are examples of relative motions which can be produced by the agents with respect to the object. In this paper only pure rolling and pure sliding relative motions between the fingertips and the object and finger relocation are assumed. The planner allows breaking contact between a fingertip and the object due to finger relocation. The relative motion between the fingertips and the object determines the whole motion of the system,

if the object's trajectories are known. We use simulated annealing algorithm (SA) [6], at the local planner level to choose these relative velocities. The purpose of using SA is to avoid local minima of the motion planning objective function which exist because of the various constraints required for defining feasible motions and forces. SA is a general-purpose optimization tool which can be easily used with any constraints in the searching space. Motion planning can be treated as an optimization problem, therefore SA is useful if the energy function of the searching space has many minima. Effective applications of SA are used in different fields of robotics. A near time-optimal inspection-task-sequence planning for two cooperative industrial robots is outlined in [5]. The object to be observed is moved by one of the robots, while the other handles the inspection tool. The objective of the task-sequence planning is to find a series of near time-optimal final configurations for two arms where the inspection operations are undertaken for segment (link) motions and find a near time-optimal task sequence of inspection points. The task-sequence planning is formulated as a variation of the traveling salesman problem (see [6]), and SA is used to find a near time-optimal route.

The proposed algorithm can be used for example to move a known shaped object to a different position and orientation with a robotic hand or with multiple robotic arms. The organization of the paper is as follows: in section 2 we give the background of the problem. In section 3 the overview of the planning system is presented, section 4 describes the use of SA with the relative velocity matrix. In section 5 simulation results are presented.

## 2. Background of the Kinematic Modelling and Constraints

The relative motion of two contacting rigid and piecewise smooth bodies is solved by [2] by describing the motion of the contact point on the surface of the bodies. The relative velocities between the contacting bodies, and the description of the contacting surfaces are required to solve the equations of the contact motion. The motion constraints are defined in the following: **Maintaining contact between the object and the fingertips** – it is given by the solution of the differential equations of contact motion. **Equilibrium condition** – in case of quasi-static motion the resultant force applied on the object has to be zero. **Rolling constraint** – in case of the relative motion between the fingertips and the object is rolling or stationary, the relative linear ( $v_i$ ) and angular velocities ( $\omega_i$ ) are:  $v_{i_x} = v_{i_y} = v_{i_z} = \omega_{i_z} = 0$  where  $z$  is the direction of the surface normal. The contact force has to be inside of the cone of friction. **Sliding constraint** – if the relative motion is pure sliding ( $\omega_{i_x}, \omega_{i_y}, \omega_{i_z} = 0$ ) the contact force has to be outside of the cone of friction, and the tangential component of the grasping force at a given contact point should lie in the plane given by  $v_{i_x}$  and  $v_{i_y}$  and has to have the same direction. **No-collision** between the fingers and object, and between fingers and obstacles, and **workspace condition** – e.g. the contact points have to be in the reachable space of the fingers of the robotic hand. **Relocation condition** – the relocating finger does not apply any

contact force on the surface of the object during breaking contact and relocation.

### 3. Overview of the Multi-Finger Planning System

The planner algorithm consists of two main parts: the global and the local planner (Fig. 1). 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 nodes of the graph are the subgoals and an arc is a nominal path between two neighboring subgoals. The global planner also decides on the usage of additional static contact surfaces.

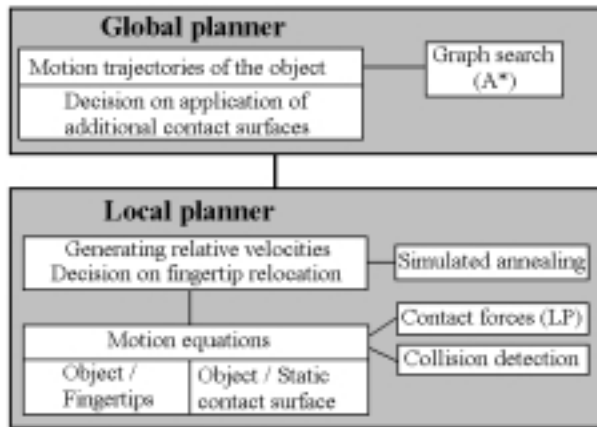


Fig. 1. Planning flowchart of the system.

The local planner includes four parts: a) the kinematics solution (computes the trajectories), b) the force computing algorithm (solves for the contact forces) and c) the generation of the relative velocities between the fingertips and the object using a d) collision detection algorithm. The motion of fingertips is dependent on the relative velocities between the fingertips and the object chosen by the local planner algorithm. Breaking contact due to finger relocation is applied for example if the contact forces are large compared to other configurations. Breaking contact is determined by the planner, the decision of breaking contact is based on an objective function which shows how appropriate the object and finger configurations are. In case if two fingertips are close to each other and the force equilibrium (applied on the object) can be maintained only by one, then one of them is unnecessary so it can break the contact and relocate itself to increase the value of the objective function of the grasp. For object relocation an additional finger is used: the algorithm

searches for a new contact point for the additional fingertip, meanwhile one of the other fingertips breaks the contact. The later finger will not participate in the grasp until another fingertip breaks the contact. The relocation algorithm searches for the optimal new contact position for the fingertip, the objective function of relocation is defined as the sum of contact forces applied on the object by all the fingertips. A strategy for relocating is the following: the relocation algorithm scans the local coordinates of the object (Gauss frame), for each potential fingertip position the sum of contact forces applied by all the fingertips is computed as if the relocating fingertip would be at the potential contact point. In every potential contact point, collision detection should be performed. The potential contact point with the smallest relocation function value is selected as the new contact point for the relocating fingertip.

The forces at the contact points are resolved by the solution of a linear program (LP). The aim of this task is to comply with the constraints and minimize the contact forces. The constraints of the LP task are the motion constraints (described in the previous section): the equilibrium constraint and the rolling and sliding constraints. In the LP task we minimize the sum of the normal components of the contact forces subject to the motion constraints. Any solution of forces that satisfies the constraints is appropriate because the object's movement is already defined by the constraints and the geometry. The advantage of minimizing the normal component of the contact forces is to save in total energy and to spare the object if it is delicate.

#### 4. Simulated Annealing Algorithm as a Local Planner

Consider the complete motion sequence of the object and the fingers between two subgoals. The motion of the fingertips (thus the motion of the fingers) is parameterized by relative linear and angular velocities. In *Fig. 2* the first configuration and the object's position and orientation at the last configuration are given by the global planner. The in-between frames are filled up by the local planner. The object is held by three fingers: the orientation of the object and fingertips are marked by a normal-direction radial line. An additional static contact surface is used, so one finger may break the contact, the remaining two can maintain the force equilibrium with the help of the contact surface.

Using SA algorithm we search for appropriate relative velocity parameters. The SA method chooses from a large number of sequences using a relative velocity matrix, and accepts the best of those (which is near optimal). For example let the interval between two subgoals ( $t_k$ ) be divided into  $n$  sub-intervals ( $\Delta t = t_k/n$ ). For every  $\Delta t$  interval we would like to generate relative velocities for all fingertips. Consider the complete motion, there are  $n * k$  parameters, where  $k$  is the number of fingers, hence the complete motion can be parameterized by a matrix of the relative velocities denoted by  $\mathbf{V} \in \mathbb{R}^{n \times k}$ . The relative velocities  $v_{ij}$  are chosen automatically by the SA algorithm which tries to optimize the  $n * k$  size matrix of the relative velocities considering given constraints (shown in section 2). Let the  $u_j$  variable

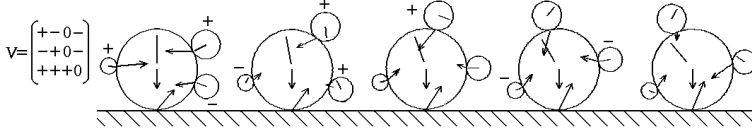


Fig. 2. A motion sequence, parameterized by the relative velocities (rolling clockwise and counterclockwise in this example, denoted by  $-$  and  $+$  respectively). Matrix  $\mathbf{V}$  contains the velocity parameters: rows and columns are describing fingers and configurations respectively.

be defined as the relative velocity of the  $i^{\text{th}}$  fingertip in the  $j^{\text{th}}$  interval. The values of the variables can be quantized (not necessary), for example  $u_j \in \{0, Relocation, v_{x_{max}}, -v_{x_{max}}, v_{y_{max}}, -v_{y_{max}}, \omega_{x_{max}}, -\omega_{x_{max}}, \omega_{y_{max}}, -\omega_{y_{max}}\}$ .

The energy function of the SA algorithm has several components (e.g. Eq. 1): the first and the second components are defined to be: a) **No collision** of fingers, b) **force equilibrium** (applied on the object). Other components can be heuristic, for example: c) **the absolute sum of contact forces** for all fingertips in all  $\Delta t$  intervals has to be small. This is useful if the object is delicate or if the objective is to minimize the energy. Another component can d) **minimize the number of relative velocity changes**. An example of the energy function can be written as:

$$E(\mathbf{V}) = K_c \sum_{j=1}^n c_j + K_e \sum_{j=1}^n e_j + K_f \sum_{i=1}^k \sum_{j=1}^n |f_{ij}| + K_g \sum_{i=1}^k g_i. \quad (1)$$

The first sum of (1) is the penalty due to collision (the number of collisions multiplied by a  $K_c$  constant), the second is the penalty due to no force equilibrium. The third part is the absolute sum of the contact forces, the fourth is the number of velocity changes of the fingertips. The definitions of the variables and constants are:

$E$  is the energy function we want to minimize,  $\mathbf{V}$  contains the relative velocities of the fingertips (parameters), the variables of the system,  $K_c, K_e, K_f, K_g$  are positive weight constants for collision, equilibrium, forces and velocity changes, respectively;  $n$  and  $k$  denote the number of intervals between two subgoals and the number of fingers, respectively.  $c_j$  equals 0 if there is no collision in the  $j^{\text{th}}$  interval and 1 otherwise,  $e_j$  equals 0 if there is force equilibrium in the  $j^{\text{th}}$  interval and 1 otherwise.  $f_{ij}$  is the contact force applied by the  $i^{\text{th}}$  fingertip at the  $j^{\text{th}}$  interval.  $g_i$  is the number of relative velocity change of the  $i^{\text{th}}$  fingertip between the two subgoals.  $K_c \approx K_e, K_c, K_e \gg K_f, K_g$  because no collision and equilibrium are more important than small contact forces and a small number of velocity changes. The force component of the energy function uses the force computing LP (described in section 3), which is called for every configuration in every iteration.

## 5. Discussion, Simulation Results

In order to demonstrate the applicability of the developed method to the design of object reconfiguration with mobile agents including cooperating robots, in the simulation three and four fingered hands were implemented having six degree of freedom for each finger. The model of each finger satisfies the joint formula and Denavit-Hartenberg form of the Puma 560 robotic arm. Hence the fingers of the palm of the dextrous hand can also be substituted by cooperating robots on fixed or mobile platforms (see *Fig. 3*).



*Fig. 3.* A robotic hand grasping an object.

The nature of the objective function of the optimization problem is highly dependent on the constraints. The advantage of using SA is to avoid local minima of the motion planning objective function. If there is no local minimum then simply downhill search can be applied instead of SA algorithm which is equivalent to applying an SA algorithm with very small temperature. In most of the cases there are many local minima, which can be shown by executing downhill searches from different random generated initial  $\mathbf{V}$  matrices. In all of our simulations the energy function has had local minima which further justified the advantages of using SA. In *Fig. 4* an example for finger relocation is shown at the local level. During the finger relocation there are always three fingers in contact with the object. In the figure only the last segment of each finger (spheres) is shown.

Computing trajectories for the fingertips with the SA based algorithm affects near-optimal paths, optimized by arbitrary parameters (for example small contact forces). The speed of the computation is low, for example computing 30 sub-intervals on a PC with 4000 SA iterations takes one hour, so it cannot be used real-time yet. Finger relocation, collision detection and contact force generation are the most time consuming parts of the planning system. The method has a number of parameters which need to be fine tuned.

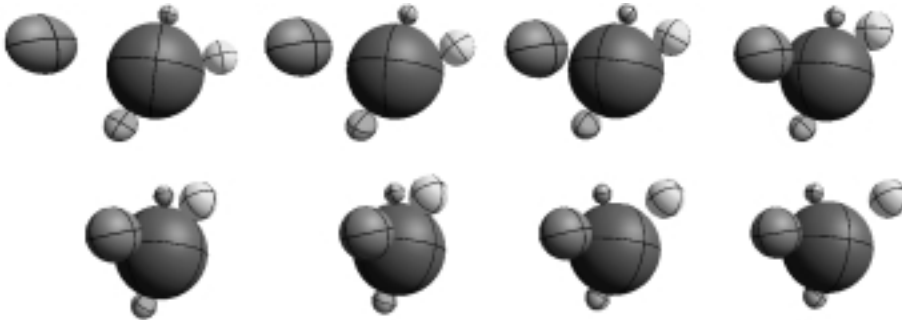


Fig. 4. Finger relocation

## 6. Conclusion

An algorithm has been presented for the object re-configuration problem if the task is to find appropriate trajectories for the fingertips to move the object to a desired position and orientation. The motion of the fingertips relative to the object can be rolling, sliding, and breaking contact due to finger relocation. Our contribution is an SA based relative velocity generator of the motion planner algorithm at the local planner level. Using a relative velocity matrix (which defines the motion sequence) with an SA based algorithm the motion can be optimized by various factors for example small contact forces. The algorithm can also be applied for object reconfiguration by mobile agents including cooperating robots.

## Acknowledgement

Support for the research is provided by the Natural Sciences and Engineering Research Council of CANADA (NSERC) grant No. 611205 and the Hungarian National Research Programs under grant No. FKFP 0417/1997 and OTKA T 029072.

## References

- [1] VASS, G.–PAYANDEH, S.–LANTOS, B.: Application of Simulated Annealing for Object Manipulation with Multiple Agents, *Proceedings of the Second IASTED International Conference, Control and Applications, Banff, Canada, 1999*, pp. 566–571.
- [2] MONTANA, D.N: The Kinematics of Contact and Grasp, *International Journal of Robotics Research*, **7** (3), 1988, pp. 17–31.
- [3] PONCE, J.–SULLIVAN, S.–SUDSANG, A.–BOISSONNAT, J-D.–MERLET, J. P.: On Computing Four-finger Equilibrium and Force-closure Grasps of Polyhedral Objects, *International Journal of Robotics Research*, **16** (1), 1997, pp. 11–35.

- [4] BUSS, M.–FAYBUSOVICH, L.–MOORE, J. B.: Dikin-type Algorithms for Dextrous Grasping Force Optimization, *International Journal of Robotics Research*, **17** (8), 1998, pp. 831–839.
- [5] CAO, B.–DODDS, G.I.–IRWIN, G. W.: A Practical Approach to Near Time-optimal Inspection-task-sequence Planning for two Cooperative Industrial Robot Arms, *International Journal of Robotics Research*, **17** (8) , 1998, pp. 858–67.
- [6] ANSARI N.–HOU, E.: *Computational Intelligence for Optimization*, Kluwer Academic Publishers , 1997, pp. 47–82.