

Paper:

# Object Reconfiguration with Dextrous Robot Agents

Gábor Vass\*, Béla Lantos\*, and Shahram Payandeh\*\*

\*Department of Control Engineering and Information Technology, Budapest University of Technology and Economics  
H-1117 Magyar Tudosok krt. 2, Budapest, Hungary

E-mail: vassgabor@yahoo.com, lantos@iit.bme.hu

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

E-mail: shahram@cs.sfu.ca

[Received September 22, 2005; accepted October 28, 2005]

**This paper addresses an object manipulation planning algorithm for dextrous robot systems consisting a multifingered hand and a robotic manipulator. A method has been developed for object reconfiguration design. The result is a new algorithm using artificial intelligence based on simulated annealing and A\* search. The upper level of the manipulation system, the global planner generates the motion of the object. The lower level, the local planner deals with the motion of the agents relative to the object and the design of the contact forces. The local planner is based on simulated annealing, thus the the local minima can be avoided in the energy function of the motion with high probability. Application of the algorithm has been discussed for three robot arms.**

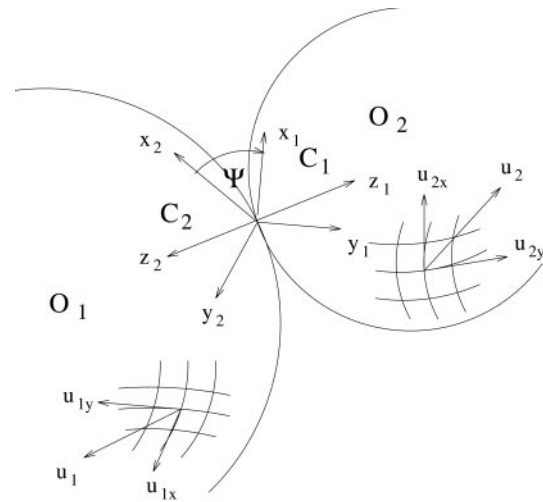
**Keywords:** robot, object reconfiguration, simulated annealing

## 1. Introduction

Object manipulation either by multiple agents or dextrous robotic hand is a center field in robotic research. In this paper a motion planner for object reconfiguration with multiple dextrous agents is proposed.

First the object is required to be located within its environment. Then, the object must be initially grasped by the agents. When the grasp is stable, the manipulation process can be started. The manipulation task (called object reconfiguration problem) is stated as the following: given an initial grasp of the object find the motion's trajectories of the agents to move the object to the desired configuration. In general collision free paths for all agents must be found toward the contact points on the object (pre-grasp configuration) and the grasping and manipulation forces should then be exerted on the object by the agents. These forces are determined first to ensure a stable grasp, then to manipulate the object.

The object manipulation problem can be divided into two sub-problems: the global planner and the local planner (see [2]). The task of the global planner is to search



**Fig. 1.** The  $C_1, C_2$  contact frames.

for nominal path in the configuration space among static obstacles between the initial and the desired position and orientation of the object by generating points (subgoals) in the object's configuration space. The local planner tries to find admissible motion of the agents between subgoals and generates the detailed trajectories of the agents' motion (path).

Numerous object manipulation algorithms have been proposed, one of the most frequent model used in motion planner algorithms is the description of the motion of two contacting object which is given in [4]. **Fig.1** illustrates the contact frames  $C_1$  and  $C_2$  of two known contacting objects ( $O_1, O_2$ ), which move while maintaining the contact with each other.  $\mathbf{u}_1 = (u_{1x}, u_{1y})$  and  $\mathbf{u}_2 = (u_{2x}, u_{2y})$  are 2D Gauss frame coordinates of the 3D surfaces with parameterization:  $(x, y, z) \rightarrow (u_x, u_y)$ .  $\Psi$  denotes the angle between the x-axes of the contact frames of the objects ( $C_1, C_2$ ).

The topic of object manipulation with agents 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 the agents can produce with respect to the object. In this paper only pure rolling and pure sliding relative motions between the agents and the

object are assumed. The two relative motion types have difference in the exerted forces in the contact points, sliding requires contact forces outside of the cone of friction, otherwise the motion will be rolling.

A separate problem within the object manipulation problem is determining the contact forces which assure force closure against disturbances in any possible directions if the contact points have already been found. This problem is a nonlinear programming (NP) problem if the friction cone is not approximated by linear constraints. Since this NP subproblem is a numerically expensive low level subproblem within the object manipulation hence we use first linear approximation of the friction cones allowing the application of linear programming (LP) method for the low level optimization subproblem. The friction cone approximation can be removed by using linear matrix inequality (LMI) method first used by [3] for grasp analysis. LMI technique allows the use of interior point polynomial algorithms in convex programming (see [5]).

Instead of random selection proposed in [2] and [8], simulated annealing algorithm can be used at the local planner level to choose relative velocities for the agents (see [7]). The purpose of using simulated annealing is to avoid local minima of the motion planning objective function which exist when there are various constraints required for defining feasible motions and forces. Simulated annealing 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 Simulated annealing is useful if the energy function of the searching space has many minima.

The objective of this paper is to present a quasi-static Simulated annealing based motion planner which uses a proposed relative velocity matrix to define the motion sequence of the object and the agents. The organization of the paper is as follows: in section 2 the motion constraints are introduced. In section 3 the overview of the planning system is presented, and a solution for removing the friction cone linearization is shown. In section 4 the relative velocity matrix is introduced. Section 5 describes the use of Simulated annealing with the relative velocity matrix. Section 6 discusses the results. Finally the conclusions of the paper are presented in section 7.

## 2. Motion Constraints

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 main contribution of this paper is a definition of a relative velocity generating method for the kinematic model while the motion of the object and the agents is constrained during the manipulation. The motion constraints are defined in the following:

**Maintaining contact between the object and the agents** – it is given by the solution of the differential equa-

tions of contact motion.

**Equilibrium condition** – in case of quasi-static motion the resultant force applied on the object has to be zero. Any non-grasping forces applied on the object are defined as “external force”. The grasp transformation is given as  $G \in R^{6 \times 6n}$  if  $n$  is the number of contact points.  $G$  transforms forces and torques between the frames of contact points  $(C_1, C_2, \dots, C_n)$  to the object frame in case of point contact model. It is used to determine the resultant force applied on the object by the agents. The equilibrium constraint is written as:

$$G[f_1^T, f_2^T, \dots, f_n^T]^T = -f_{O,ext}. \quad \dots \dots (1)$$

$f_1, f_2, \dots, f_n$  denote the contact forces at the contact points' frames  $1, 2, \dots, n$ , for instance in 2D  $f_i = (f_{ix}, f_{iy})$ ;  $f_{O,ext}$  is the resultant of the external forces applied on the object, for example gravity and/or the friction force  $\mu_O mg$  between the object and the plane of motion ( $\mu_O$  denotes the coefficient of friction,  $m$  the mass and  $g$  coefficient of gravity).  $G$  is the transpose of the resulting Jacobian between object origin and contact points.

**Rolling constraint** – for a pure rolling contact, the relative motion between the agents and the object is rolling or stationary. The relative linear ( $v_i$ ) and angular velocities ( $\omega_i$ ) are:

$$v_{ix} = v_{iy} = v_{iz} = \omega_{iz} = 0 \quad \dots \dots \dots (2)$$

where  $z$  is the direction of the surface normal. The contact model is point contact with friction (assuming a Coulomb friction modeling),  $\mu$  is the friction coefficient. The contact force has to be inside of the cone of friction. In the motion planning algorithm this constraint is linearized: instead of cone of friction pyramid of friction is used. This pyramid has four faces given by

$$|f_{ix}| < -\mu f_{iz}/\sqrt{2}, |f_{iy}| < -\mu f_{iz}/\sqrt{2}, f_{iz} < 0. \quad \dots (3)$$

The constraint is linearized in order to be able to use algorithm based on linear programming.

**Sliding constraint** – In case of the relative motion is pure sliding, the spatial relative velocity should satisfy

$$v_{iz} = \omega_{ix} = \omega_{iy} = \omega_{iz} = 0. \quad \dots \dots \dots (4)$$

The contact force has to be outside of the pyramid of friction.

$$|f_{ix}| > -\mu f_{iz}, |f_{iy}| > -\mu f_{iz}, f_{iz} < 0 \quad \dots (5)$$

The tangential component of the grasping force at a given contact point should lie in the plane given by  $v_{ix}$  and  $v_{iy}$  and has to have the same direction and sign ( $\gamma$  is arbitrary):

$$f_{ix} = \gamma v_{ix}, f_{iy} = \gamma v_{iy}, \gamma > 0. \quad \dots \dots \dots (6)$$

**No-collision between any pair of agents** – a collision detection algorithm must be used dealing with the agents, the object, and the obstacles of the environment.

**Workspace condition** – The agents must stay inside their workspace. For example in case of a dextrous hand