digital

PARIS RESEARCH LABORATORY

34

A penalty function method for constrained motion planning

September 1993

Pierre Ferbach
Jérôme Barraquand

34

A penalty function method for constrained motion planning

Pierre Ferbach
Jérôme Barraquand

September 1993

Publication Notes

The authors can be contacted at the following addresses:

Pierre Ferbach	Jerome Barraquand
Ecole Nationale Supérieure	Digital Equipment Corporation
des Techniques Avancées	Paris Research Laboratory
32 Bd Victor	85 Avenue Victor Hugo
75015, Paris, France	92500 Rueil-Malmaison, France
`ferbach@ensta.fr`	`barraquand@prl.dec.com`

Abstract

The path planning problem, i.e. the geometrical problem of finding a collision-free path between two given configurations of a robot moving among obstacles, has been studied by many authors in recent years. The interest in *constrained motion planning* is more recent. Constrained motion planning consists in finding motion sequences for robotic systems whose free space in not an open subset of the configuration space. In particular, *manipulation planning* is an important instance of the general constrained motion planning problem. It consists in planning the motions of a system including robots and movable bodies that are constrained by grasping relationships. In a manipulation planning problem, the dimension of the free space may be dynamically changing along the solution path.

In this report, we establish necessary and sufficient conditions under which grasping constraints are holonomic. Then we present a systematic approach to motion planning in the presence of grasping constraints deriving from this theory. Its principle is to replace a constrained problem by a convergent series of less constrained subproblems increasingly *penalizing* motions that do not satisfy the constraints. Each subproblem is solved using a standard path planner. We use the method of Variational Dynamic Programming for solving the subproblems. We report several experiments in manipulation planning with multiple redundant robots and multiple moving objects in configuration spaces having up to 12 degrees of freedom (DOF). The implemented planner has solved manipulation planning problems of unprecedented complexity.

Résumé

Le problème de la planification de trajectoire, i.e. the problème géométrique consistant à trouver des chemins sans collisions entre deux configurations d'un robot en présence d'obstacles, a été largement étudié ces dernières années. L'intérêt pour les problèmes de planification *sous contraintes* est plus récent. La planification de mouvement sous contraintes consiste à trouver des séquences de mouvements pour des systèmes robotisés dont l'espace libre n'est pas un sous espace ouvert de l'espace des configurations. En particulier, le problème de la planification de manipulation est une instance importante du problème général de la planification sous contraintes. Il consiste à planifier les mouvements d'un système incluant des robots et des objets à manipuler liés entre eux par des contraintes de préhension. Dans un problème de manipulation, la dimension de l'espace libre peut changer dynamiquement le long du chemin solution.

Dans ce rapport, nous établissons les conditions nécessaires et suffisantes sous lesquelles une contrainte de préhension est holonome. Nous présentons ensuite une approche systématique pour la planification sous contraintes de préhension dérivant de cette théorie. Son principe est de remplacer un problème sous contraintes par une suite convergeante de sous-problèmes moins contraints pénalisant de façon croissante les mouvements qui ne satisfont pas les contraintes. Chaque sous-problème est résolu grâce à un planificateur de trajectoire classique. Nous utilisons la méthode de Programmation Dynamique Variationnelle pour résoudre les sous-problèmes. Des simulations effectuées sur plusieurs problèmes de manipulation sont présentées dans ce rapport, incluant plusieurs robots redondants et plusieurs objets à manipuler dans des espaces ayant jusqu'à 12 degres de liberté. Le planificateur peut résoudre des problèmes de manipulation d'une complexité sans précédent.

Keywords

Path Planning, Constrained Motion Planning, Manipulation Planning, Grasping, Nonholonomic Constraints, Robotics, Computer Aided Design, Computer Graphics, Penalty Function Method

Contents

1 Introduction

We present a systematic approach to *constrained motion planning* for robotic systems with many degrees of freedom (DOF). Constrained motion planning consists in finding motion sequences for robotic systems whose free space in not an open subset of the configuration space. In general, constraints in motion planning problems can be classified into two categories:

- *Path-independent constraints*, i.e. constraints that only depend on the current configuration of the robot. Path-independent constraints are often called *holonomic constraints*.
- *Path-dependent constraints*, that depend on more than a single configuration along the path. Path-dependent constraints are generally called *nonholonomic* constraints. An important class of path-dependent constraints is that of constraints on the first derivative (velocity) that are not integrable.

In particular, *manipulation planning* is an important instance of the general constrained motion planning problem. Given an environment containing robots, stationary obstacles, and movable bodies, the manipulation problem consists in finding a sequence of free robot motions, grasping and ungrasping operations, to reach a given state from a given initial state in the joint configuration space of all robots and movable bodies. In a manipulation planning problem, the dimension of the free space may be dynamically changing along the solution path.

We first develop a theory of manipulation planning. We show that in general a grasping constraint is nonholonomic, since it involves the first derivative of the path followed by the movable objects. We characterize conditions under which grasping constraints can be made holonomic. Loosely stated, a grasping constraint is holonomic if and only if the set of stable configurations for the movable object is a series of disconnected points called *docking positions*.

Then, we present an implemented planner derived from this theory. This planner is capable of planning manipulation tasks when the number of docking positions is finite. This simplification makes the problem *holonomic*. The principle of our approach is to replace the original equality-constrained problem by a convergent series of more and more difficult inequality-constrained planning problems with open free spaces increasingly *penalizing* motions that do not satisfy the grasping constraints. We call this approach a *penalty function* method. In other words, grasping constraints are handled by our planner in a *progressive* fashion. We first solve the problem assuming that the movable objects can move without being grasped by a robot. Then, this path is used as an input for a series of more and more difficult problems forcing the objects to get closer and closer to the robots in order to move. Each subproblem is solved using a standard path planner. We use the method of Variational Dynamic Programming (VDP) described in Barraquand and Ferbach 1993 [2] for solving the subproblems. However, in theory, any other variational planner could be used instead of VDP. We call the overall method *Progressive Variational Dynamic Programming* (PVDP).

The planner has successfully solved manipulation planning problems of unprecedented complexity. We report several manipulation planning experiments for systems with up to 12 DOF.

Initially, this penalty function approach was developed in Barraquand and Ferbach 1993 [2] as a variant of the VDP method for solving difficult instances of the basic path planning problem in open free space. Indeed, given any variational path planning method, one can replace the collision avoidance constraints in a classical path planning problem by a convergent series of simpler constraints increasingly penalizing motions that do not satisfy collision avoidance constraints. The resulting planner is less general in theory than the original VDP planner, since it uses problem-specific heuristics to guide the search. On the other hand, it is dramatically faster. In fact, it can solve some problems in a time comparable to that of potential-field based methods (see Barraquand and Ferbach 1993 [2]).

This report is organized as follows. In Section 2, we relate our contribution to previous work in motion planning. In Section 3, we develop a theory of manipulation planning, and characterize in particular conditions under which grasping constraints can be made holonomic. In Section 4 we describe the general principle underlying the penalty function approach to manipulation planning in the presence of holonomic grasping constraints. We also describe our implemented planner PVDP. In Section 5, we present experimental results illustrating the capabilities of the implemented planner. In section 6, we discuss current limitations and possible generalizations of the PVDP approach to manipulation planning. We also review possible applications of the penalty function method to other constrained motion planning problems. Section 7 is the conclusion.

2 Relation to other work

The path planning problem, i.e. the geometrical problem of finding a collision-free path between two given configurations of a robot moving among obstacles, has been studied by many authors in recent years (Latombe 1990 [8]).

The interest in constrained motion planning is more recent in the robotics literature. The problem of planning the path of a convex polygonal robot translating in a two-dimensional polygonal workspace in the presence of multiple convex polygonal movable objects is addressed in Wilfong 1988 [12]. The general manipulation problem is described in a series of papers from Alami, Laumond, and Simeon (e.g. Alami Simeon and Laumond 1989 [1], Laumond and Alami 1989 [10]).

To the best of our knowledge, although nonholonomic rolling constraints in motion planning have been investigated by several authors (e.g. Laumond 1986 [9] and subsequent papers, Li and Canny 1989 [11], Barraquand and Latombe 1989 [4], 1993 [6]), the nonholonomic nature of grasping constraints in manipulation tasks has never been investigated to date.

An implemented algorithm for manipulation task planning with a 2 DOF robot grasping a single object at a time and several 2 DOF bodies translating in the plane is presented in Alami Simeon and Laumond 1989 [1]. The planner has two components: a classical path planner, and a manipulation task planner (MTP). The MTP plans a sequence a robot motions, grasping and ungrasping operations, and transfer motions (i.e. motions of the robot together with a grasped object). The approach is in practice limited to non-redundant robots with few DOF, and requires an exhaustive exploration of the robot's configuration space. Koga and Latombe 1992 [7] present several implemented planners solving various dual-arm manipulation planning problems of increasing difficulty. They use and extend the framework of Alami Simeon and Laumond 1989 [1]. The planner is again the combination of a path planner and a manipulation task planner. For problems with many degrees of freedom, the path planner used is the potential field based planner RPP (Barraquand and Latombe 1991 [5]).

Our approach to manipulation task planning is fundamentally different. We do not decompose the problem into a sequence of robot motions and manipulation tasks. Our planner is not a combination of a path planner and a manipulation task planner. Instead, we simply consider the whole manipulation problem as a special instance of the basic path planning problem in the joint configuration space of the robot and the movable objects. The major advantage of this approach is to avoid the artificial decoupling between motion planning and task planning. As a consequence, PVDP can solve manipulation planning problems of unprecedented complexity.

3 Manipulation tasks, grasping, and nonholonomy

3.1 The manipulation planning problem

We consider one or more robots $R_1, \ldots, R_l$ with respective configuration spaces $\mathcal{C}^{R_1}, \ldots, \mathcal{C}^{R_l}$, one or more movable objects $M_1, \ldots, M_s$ with respective configuration spaces $\mathcal{C}^{M_1}, \ldots, \mathcal{C}^{M_s}$, evolving in a workspace $\mathcal{W}$ populated by stationary obstacles. We assume that for any object $A \in \{R_1, \ldots, R_l, M_1, \ldots, M_s\}$, the manifold $\mathcal{C}^A$ is bounded (hence compact, since it is a closed subset of the Euclidean space) and connected. The joint configuration space of the robots is denoted by $\mathcal{C}^{robots} = \prod_{i=1}^{l} \mathcal{C}^{R_i}$. Similarly, the joint configuration space of the movable objects in denoted by $\mathcal{C}^{obj} = \prod_{j=1}^{s} \mathcal{C}^{M_j}$. The joint configuration space of the robots and movable objects is simply the cartesian product $\mathcal{C} = \mathcal{C}^{robots} \times \mathcal{C}^{obj}$. We denote its dimension, i.e. the total number of degrees of freedom of all the robots and movable objects, by n.

For any robot or movable object $A \in \{R_1, \ldots, R_l, M_1, \ldots, M_s\}$, we can define the *projection* that maps any configuration for all robots and objects $\mathbf{q} = \left(\mathbf{q}^{R_1}, \ldots, \mathbf{q}^{R_l}, \mathbf{q}^{M_1}, \ldots, \mathbf{q}^{M_s}\right) \in \mathcal{C}$ to the corresponding configuration $\mathbf{q}^A \in \mathcal{C}^A$ of A.

$$\begin{array}{rcccc} X & : & \mathcal{C} & \longrightarrow & \mathcal{C}^A \\ & & \mathbf{q} & \longmapsto & \pi^A(\mathbf{q}) = \mathbf{q}^A \end{array}$$

We assume that an appropriate distance metric $d(\mathbf{q}, \mathbf{q}')$ has been defined over $\mathcal{C}$. The collision avoidance space, i.e. the set of configurations $\mathbf{q} \in \mathcal{C}$ such that robots, movable objects, and stationary obstacles do not collide with each other in denoted by $\mathcal{C}_{avoid}$. We assume that $\mathcal{C}_{avoid}$ can be represented as

$$\mathcal{C}_{avoid} = \{\mathbf{q} \in \mathcal{C}, \quad g_{avoid}(\mathbf{q}) < 0\}$$

where g_{avoid} is an appropriate function defined over $\mathcal{C}$. For example, g_{avoid} could be the opposite of the minimum distance between any robot or object with other robots, objects, or with obstacles. We see that the collision avoidance constraint is an *inequality constraint* in configuration space. $\mathcal{C}_{avoid}$ is an open subset of $\mathcal{C}$.

Let $\mathbf{q}_{init}$ be the initial configuration of the system, and $\mathbf{q}_{goal}$ its desired configuration. The problem of manipulation planning consists in finding a sequence of free robot motions, grasping and ungrasping operations, to reach $\mathbf{q}_{goal}$ from $\mathbf{q}_{init}$ in the joint configuration space $\mathcal{C}$ of the robot and all movable bodies.

This sequence can be simply represented by a path γ in $\mathcal{C}$ joining $\gamma(0) = \mathbf{q}_{init}$ and $\gamma(1) = \mathbf{q}_{goal}$. As in any classical path planning problem, the path γ must not cross colliding configurations, i.e. we must have:

$$\forall t \in [0, 1], \quad \gamma(t) \in \mathcal{C}_{avoid}$$

However, there are two other kinds of constraints that must also be satisfied by γ: grasping constraints, and stability constraints. These constraints are due to external forces (e.g. gravitational forces) applied to the movable objects. First, the movable objects can only move when they are grasped by some robot. Hence, the forbidden paths γ are not only those for which the robots or the movable objects hit the stationary obstacles or collide with each other, but also all those for which the movable objects are moving without being grasped by a robot. Second, movable objects can be ungrasped by robots only within the subset of stable configurations, i.e. the subset of configurations where exterior forces sum to zero.

The formalisation of these notions is the object of the next subsection.

3.2 Grasping and stability constraints

We first introduce the notion of grasping constraint by a simple example. More complex examples are presented in Section (5). Let us consider a 2-dimensional workspace $\mathcal{W}$, with a single serial robot manipulator R with $n_{robot} = 2$ revolute joints a_1 and a_2 as illustrated in Figure (1). We assume that the robot base is fixed at the position (x_0, y_0) in $\mathcal{W}$. The coordinates (x_{eff}, y_{eff}) of the end-effector satisfy the *forward kinematic equations*:

$$x_{eff}(a_1, a_2) = x_0 + L_1 \cos a_1 + L_2 \cos a_2$$

$$y_{eff}(a_1, a_2) = y_0 + L_1 \sin a_1 + L_2 \sin a_2$$

where L_1 and L_2 are the respective lengths of the two robot arms. A configuration of R is a couple $\mathbf{q}^R = (a_1, a_2)$. We assume that the workspace is populated by a single movable disk M with $n_{obj} = 2$ DOF translating in the plane, and whose configuration is defined by the coordinates of its center $\mathbf{q}^M = (x_m, y_m)$. The total dimension of the joint configuration space is $n = n_{robot} + n_{obj} = 4$. We define a grasp between R and M as the set G_{grasp} of configurations $\mathbf{q} = (a_1, a_2, x_m, y_m)$ of the whole system verifying:

$$g^1_{grasp}(a_1, a_2, x_m, y_m) = x_{eff} - x_m = 0$$

and

$$g^2_{grasp}(a_1, a_2, x_m, y_m) = y_{eff} - y_m = 0$$

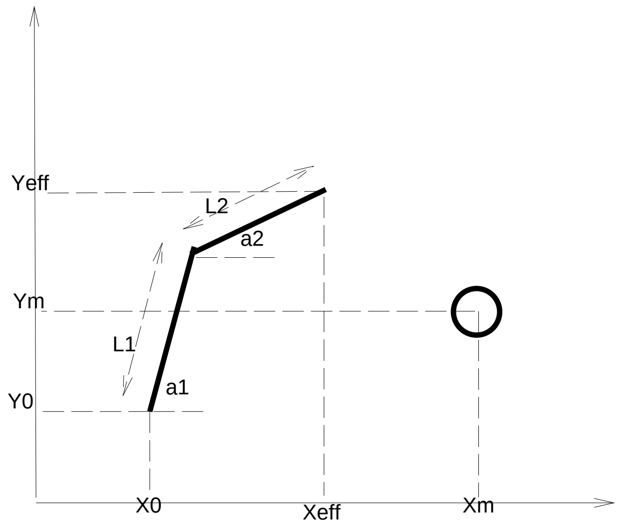

Figure 1: A simple two-dimensional grasping problem in a two-dimensional configuration space

The number h of coordinates of the grasping constraint $g_{grasp} = \left(g^1_{grasp}, g^2_{grasp}\right)^T$ is $h = 2$. We say that g_{grasp} is a $h = 2$-dimensional grasping constraint. Hence, we see that G_{grasp} is a $n - h = 2$-dimensional submanifold of $\mathcal{C}$. The dimension of a typical grasping constraint for a classical industrial robot with $n_{robot} = 6$ DOF manipulating a solid object with $n_{obj} = 6$ DOF in a three dimensional configuration space is $h = 6$. A robot with n_{robot} DOF designed for satisfying grasping constraints of dimension $h = n_{obj}$, i.e. for grasping objects with n_{obj} DOF, is said to be non-redundant iff $n_{robot} = h = n_{obj}$. The robot is said to be redundant iff $n_{robot} > h = n_{obj}$.

More generally, given a robot R_i and an object M_j, an h-dimensional grasping constraint g^{R_i,M_j}_{grasp} is represented by the $n - h$ dimensional set of valid grasping configurations G_{R_i,M_j}:

$$G_{R_i,M_j} = \{\mathbf{q} \in \mathcal{C}, \quad g^{R_i,M_j}_{grasp}\left(\pi^{R_i}(\mathbf{q}), \pi^{M_j}(\mathbf{q})\right) = 0\}$$

where g^{R_i,M_j}_{grasp} is a continuous mapping from $\mathcal{C}^{R_i} \times \mathcal{C}^{M_j}$ onto R^h.

$$\begin{array}{rcll} g^{R_i,M_j}_{grasp} & : & \mathcal{C}^{R_i} \times \mathcal{C}^{M_j} \quad \longrightarrow & R^h \\ & & \left(\mathbf{q}^{R_i}, \mathbf{q}^{M_j}\right) \quad \longmapsto & g^{R_i,M_j}_{grasp}\left(\mathbf{q}^{R_i}, \mathbf{q}^{M_j}\right) = \left(g^{R_i,M_j\,1}_{grasp}\left(\mathbf{q}^{R_i}, \mathbf{q}^{M_j}\right), \ldots, g^{R_i,M_j\,h}_{grasp}\left(\mathbf{q}^{R_i}, \mathbf{q}^{M_j}\right)\right)^T \end{array}$$

In general, a robot can grasp a given object in several different ways. For example, if the robot at hand has two arms, and the object to be manipulated is a long bar, there are two possible ways of grasping the long bar using both arms at both extremities of the bar, that correspond to the possibility of swapping the two arms. Such problems are called *dual arm* manipulation planning problems. An example of such problem is presented in Section (5). Also, any finite set of robots can be viewed as another robot. Hence, the above formalism applies to complex problems such as multifingered manipulation planning. We take the set G_{R_i,M_j} as the set of all possible valid grasping configurations, regardless of the way the object is being grasped.

A manipulation path satisfies the grasping constraints iff for any movable object M_j, the object is either stationary or grasped by some robot R_i. Formally:

$$\forall t \in [0,1], \quad \left(\exists i \in [1,l], \; g_{grasp}^{R_i,M_j}(\pi^{R_i}(\gamma(t)), \pi^{M_j}(\gamma(t))) = 0\right) \bigvee \left(\frac{d}{dt}\pi^{M_j}(\gamma(t)) = 0\right) \quad (1)$$

We see that a grasping constraint may be *nonholonomic*, since it involves the first derivative of the path followed by the movable object. We will formally prove this intuitive fact in the next subsection. Before, we introduce a simplified concept of stability.

An object M_j can be ungrasped by some robot only if its configuration $\mathbf{q}^{M_j}$ is stable with respect to exterior forces, e.g. with respect to the gravitational force and reaction forces from stationary obstacles. It may be that the configuration of other objects influences the stability of M_j. Here we only consider absolute stability, i.e. stability in the absence of other objects. Hence, the set of stable configurations of M_j is a well-defined subset of $\mathcal{C}^{M_j}$. It is denoted by $\mathrm{STABLE}(M_j)$. We assume that $\mathrm{STABLE}(M_j)$ is a closed set.

3.3 Nonholonomy of grasping constraints

3.3.1 An introductory example

We first consider the example above, and then turn to the general case. Applying formula (1) to the example of Figure (1), we get:

$$\forall t \in [0,1], \quad \left(x_{eff}(t) - x_m(t) = y_{eff}(t) - y_m(t)) = 0\right) \bigvee \left(\frac{dx_m}{dt}(t) = \frac{dy_m}{dt}(t) = 0\right) \quad (2)$$

If the above two-dimensional constraint were holonomic, we could *integrate* it, i.e. we could find a couple of real-valued functions F_1 and F_2 such that a path $\gamma(t) = (a_1(t), a_2(t), x_m(t), y_m(t))$ satisfies the constraint iff

$$\forall t \in [0,1], \quad F_1(a_1(t), a_2(t), x_m(t), y_m(t)) = F_2(a_1(t), a_2(t), x_m(t), y_m(t)) = 0$$

If we define the real-valued function $F = \sqrt{F_1^2 + F_2^2} = ||(F_1, F_2)^T||_{R^2}$, the above constraint can be rewritten:

$$\forall t \in [0,1], \quad F(a_1(t), a_2(t), x_m(t), y_m(t)) = 0$$

We consider an arbitrary object location (x_m^0, y_m^0) reachable by the robot, i.e. such that there exists (a_1^0, a_2^0) verifying $x_{eff}(a_1^0, a_2^0) = x_m^0$ and $y_{eff}(a_1^0, a_2^0) = y_m^0$. We write $(x_m^0, y_m^0) \in \mathrm{REACH}(M)$. We consider a path starting at $a_1(0) = a_1^0, a_2(0) = a_2^0, x_m(0) = x_m^0, y_m(0) = y_m^0$ consisting of immediately ungrasping the object and bringing the robot to an arbitrary other location $a_1^1 = a_1(1), a_2^1 = a_2(1)$. Along such a path, the object will not move since it must satisfy (2). Hence, we have $x_m(1) = x_m^0$ and $y_m(1) = y_m^0$. By holonomy, we must have at $t = 1$:

$$F(a_1^1, a_2^1, x_m^0, y_m^0) = 0;$$

We have just shown that for any reachable couple (x_m^0, y_m^0), the function F takes the value 0 no matter the value taken by its first two variables a_1^1 and a_2^1:

$$\forall (a_1^1, a_2^1) \in \mathcal{C}^R,\ \ \forall (x_m^0, y_m^0) \in \mathrm{REACH}(M),\quad F(a_1^1, a_2^1, x_m^0, y_m^0) = 0$$

Hence, F is identically null over the set of reachable configurations. Now, given any configuration $(a_1^1, a_2^1, x_m^0, y_m^0)$ such that (x_m^0, y_m^0) is reachable, we take any other reachable configuration (x_m^1, y_m^1) different from (x_m^0, y_m^0), and we can consider the path:

$$\forall t \in [0,1],\ \ a_1(t) = a_1^1,\ \ a_2(t) = a_2^1,\ \ x_m(t) = x_m^0 + t(x_m^1 - x_m^0),\ \ y_m(t) = y_m^0 + t(y_m^1 - y_m^0)$$

Along this path, F is identically null by the above result. But this path obviously does not satisfy constraint (2). Hence, we have a contradiction. The constraint (2) cannot be holonomic.

3.3.2 General case

We now turn to the general case. Before stating the result, we first define formally the notion of reachable configuration for a movable object. Given an object M_j, the configuration $\mathbf{q}^{M_j}$ is *reachable* and we write $\mathbf{q}^{M_j} \in \mathrm{REACH}(M_j)$ iff there exists a robot R_i that can grasp M_j at $\mathbf{q}^{M_j}$. The set of reachable configurations $\mathrm{REACH}(M_j)$ can be defined as follows.

$$\mathrm{REACH}(M_j) = \{\mathbf{q}^{M_j} \in \mathcal{C}^{M_j},\ \ \exists i \in [1,l],\ \exists \mathbf{q}^{R_i} \in \mathcal{C}^{R_i},\ \ g_{grasp}^{R_i,M_j}(\mathbf{q}^{R_i}, \mathbf{q}^{M_j}) = 0\}$$

Clearly, $\mathrm{REACH}(M_j)$ is closed, since the constraints $g_{grasp}^{R_i,M_j}$ are continuous and $\mathcal{C}$ is compact. We call *docking position* an object configuration that is both stable and reachable, and we write

$$\mathrm{DOCK}(M_j) = \mathrm{STABLE}(M_j) \cap \mathrm{REACH}(M_j)$$

By the above results, we see that $\mathrm{DOCK}(M_j)$ is a closed bounded subset of $\mathcal{C}^{M_j}$. Hence, $\mathrm{DOCK}(M_j)$ is compact. We will prove the following result.

Characterization of holonomic grasps

We consider the grasping constraint (1) for an object M_j. The following two statements are equivalent.

i) All pathwise-connected components of the set of stable reachable object positions $DOCK(M_j) = STABLE(M_j) \cap REACH(M_j)$ are singletons.

ii) The constraint is holonomic

In other words, a grasping constraint is holonomic iff there is no path composed only of stable reachable configurations connecting two different stable reachable configurations.

Proof of $ii) => i)$.

We first prove $ii) => i)$ by generalizing the argument developed above for the example in Figure (1).

Assume the constraint (1) is holonomic. We consider a function F mapping $\mathcal{C}$ onto R such that (1) is equivalent to:

$$\forall t \in [0,1], \;\; F(\gamma(t)) = 0$$

We take an arbitrary $\mathbf{q}_0^{M_j} \in \mathrm{DOCK}(M_j)$. Since $\mathbf{q}_0^{M_j}$ is reachable, there exists a robot R_i and a configuration $\mathbf{q}_0^{R_i}$ such that

$$g_{grasp}^{R_i,M_j}(\mathbf{q}_0^{R_i}, \mathbf{q}_0^{M_j}) = 0$$

Without loss of generality, we can consider all other objects and robots as static, and limit our study to paths in $\mathcal{C}^{R_i} \times \mathcal{C}^{M_j}$. We define the path γ starting at $\gamma(0) = (\mathbf{q}_0^{R_i}, \mathbf{q}_0^{M_j})$, consisting of ungrasping M_j and bringing R_i to an arbitrary other location $\mathbf{q}_1^{R_i}$. This path is valid since $\mathbf{q}_0^{M_j}$ is a stable configuration. By the grasping constraint (1) we must have $\gamma(1) = (\mathbf{q}_1^{R_i}, \mathbf{q}_0^{M_j})$. By holonomy, we have $F(\mathbf{q}_1^{R_i}, \mathbf{q}_0^{M_j}) = 0$. Since $\mathcal{C}^{R_i}$ is connected, this is true for any couple $(\mathbf{q}_1^{R_i}, \mathbf{q}_0^{M_j}) \in \mathcal{C}^{R_i} \times \mathrm{DOCK}(M_j)$. We can write:

$$F(\mathcal{C}^{R_i} \times \mathrm{DOCK}(M_j)) = 0 \tag{3}$$

Now, we take any other object configuration $\mathbf{q}_1^{M_j} \in \mathrm{DOCK}(M_j)$. We will show that there can be no path δ in $\mathrm{DOCK}(M_j)$ linking $\delta(0) = \mathbf{q}_0^{M_j}$ and $\delta(1) = \mathbf{q}_1^{M_j}$. This will in turn prove $i)$. Indeed, if such a path exists, we can define the path $\gamma(t) = (\mathbf{q}^{R_i}(t), \mathbf{q}^{M_j}(t))$ in $\mathcal{C}^{R_i} \times \mathcal{C}^{M_j}$:

$$\forall t \in [0,1], \;\; \mathbf{q}^{R_i}(t) = \mathbf{q}_1^{R_i}, \;\; \mathbf{q}^{M_j}(t) = \delta(t)$$

By (3) and $ii)$, this path γ satisfies the grasping constraint. But clearly γ cannot satisfy (1) since δ joins two different configurations. Hence δ cannot exist, and we get the desired result.

Proof of $i) => ii)$.

Reciprocally, let us assume $i)$. We define the following real-valued function over $\mathcal{C}$:

$$F(\mathbf{q}) = \min\left(\min_{i \in [1,l]} \left(||g_{grasp}^{R_i,M_j}(\pi^{R_i}(\mathbf{q}), \pi^{M_j}(\mathbf{q}))|| \right), \min_{D \in \mathrm{DOCK}(M_j)} \left(d(\pi^{M_j}(\mathbf{q}), D) \right) \right) \tag{4}$$

Since $\mathrm{DOCK}(M_j)$ is a compact set, the minimum $\min_{D \in \mathrm{DOCK}(M_j)} \left(d(\pi^{M_j}(\mathbf{q}), D) \right)$ is well-defined by the Bolzano-Weierstrass theorem. Hence, $F(\mathbf{q})$ is a well-defined function.

We will show that the grasping constraint (1) is equivalent to the following holonomic constraint:

$$\forall t \in [0,1], \;\; F(\gamma(t)) = 0 \tag{5}$$

Indeed, the above holonomic constraint can be rewritten:

$$\forall t \in [0,1], \;\; \left(\exists i \in [1,l], \;\; g_{grasp}^{R_i,M_j}(\pi^{R_i}(\gamma(t)), \pi^{M_j}(\gamma(t))) = 0\right) \;\; \bigvee \\ \left(\exists D(t) \in \mathrm{DOCK}(M_j), \;\; \pi^{M_j}(\gamma(t)) = D(t)\right)$$

Hence, it suffices to show the following implication for any interval $[t_1, t_2] \subset [0, 1]$

$$\Big(\forall t \in [t_1, t_2], \exists D(t) \in \mathrm{DOCK}(M_j),\ \ \pi^{M_j}(\gamma(t)) = D(t)\Big)$$
$$=> \quad \Big(\forall t \in [t_1, t_2],\ \ \tfrac{d}{dt}\pi^{M_j}(\gamma(t)) = 0\Big)$$

But, since $\mathrm{DOCK}(M_j)$ verifies $i)$, the path $D(t)$ is necessarily constant:

$$\forall t \in [t_1, t_2],\ \ \pi^{M_j}(\gamma(t)) = D(t) = D$$

Deriving the above identity with respect to t, we get the desired result.

3.4 Docking positions and holonomic grasp planning

In this paper, we consider only manipulation planning problems for which the grasping constraints are holonomic. From the above result, this implies that we impose to the set of stable reachable configurations (i.e. docking positions) to be composed of pathwise-connected components containing a single element. In practice, we will restrict ourselves to problems for which a movable object M_j is only stable at a *finite* number of docking positions $\forall u \in [1, d_j], D_j^u \in \mathcal{C}^{M_j}$, unless it is grasped by a robot. Formally:

$$\mathrm{DOCK}(M_j) = \{D_j^1, \ldots, D_j^{d_j}\}$$

We call free space and denote by $\mathcal{C}_{free}$ the subset of $\mathcal{C}_{avoid}$ satisfying the (holonomic) grasping constraints. We see that in general, $\mathcal{C}_{free}$ is not an open subset of $\mathcal{C}$, since the grasping constraints are equality constraints. The dimension of $\mathcal{C}_{free}$ can dynamically change along the solution path. This may happen when a robot grasps an object (reduction of the dimension) or ungrasps it (increase in the dimension). Hence, a standard path planner only capable of planning motions in open free spaces of constant dimension cannot be used for solving the manipulation planning problem. In the next section, we introduce a new paradigm for circumventing this limitation of path planning techniques.

4 A penalty function method for holonomic manipulation planning

In this section, we present a new paradigm for solving motion planning problems in the presence of holonomic constraints. This approach applies in particular to manipulation planning problems when the set of docking positions for movable objects is finite and prespecified as part of the planning problem.

4.1 An introductory example

We first illustrate the approach on a manipulation planning example using the setup of Figure (1). We assume that initially the robot R is at the configuration $(a_1^{init}, a_2^{init}) = (a_1^0, a_2^0)$, and that the object M is located at an arbitrary reachable configuration $\left(x_m^{init}, y_m^{init}\right) = \left(x_m^0, y_m^0\right)$. We define the goal configuration for the robot as $\left(a_1^{goal}, a_2^{goal}\right) = \left(a_1^0, a_2^0\right)$, and the goal configuration for the object as $\left(x_m^{goal}, y_m^{goal}\right) = \left(x_m^1, y_m^1\right) \neq \left(x_m^0, y_m^0\right)$. We see that the task assigned to the robot is simply a pick and place operation consisting of moving M to another location. Hence, in this case, it is easy to imagine how the problem should be addressed. First, the robot must move its end effector towards the object and grasp it. Second, both the robot and the grasped object must move in order to bring the end-effector (hence the object) to the goal configuration. Third, the robot must ungrasp the object and get back to its initial configuration. The above manipulation problem can be decomposed into a sequence of three basic motion planning problems in open free space. One may argue that it is not necessary to develop planners capable of dealing with grasping constraints, since manipulation problems can be decomposed into sequences of classical path planning problems without grasping constraints. However, the task of planning the sequence of motions, called manipulation task planning, is often much more complex than in the above pick and place problem. For example, there can be more than one object to manipulate, and the geometry of the workspace may imply that different objects interact with each other. As another example, the object to manipulate can be a long bar that must be grasped at the same time by two robot arms. In such a case, the robot arms may have to switch their respective grasps at the two ends of the bar. Examples of such problems will be presented in Section (5). For those more difficult problems, the task of planning the sequence of motions, grasping, and ungrasping operations may become a hard problem in itself. Then, a manipulation planner should be the combination of two planners: 1) a classical path planner in open free space computing motions between grasping and ungrasping operations 2) a manipulation task planner computing the sequence of grasps. This is the way the problem has been addressed in previous work (see Section (2)).

Here, we take an opposite approach, and attempt to solve the whole manipulation planning at once. This enables us to deal with intricated interactions between geometric and task planning constraints that cannot be taken into account by other planners decoupling the two problems.

We first define two docking positions for the object M, namely its initial and goal positions.

$$\mathrm{DOCK}(M) = \{\left(x_m^0, y_m^0\right), \left(x_m^1, y_m^1\right)\}$$

In a more difficult problem, we might have to define intermediate docking positions. Examples of such intermediate docking positions are given in Section (5). This raises the problem of defining these docking positions as part of the planning task. Although the issue is still open in the general case, we briefly discuss it in Section (6).

Following the results of the previous section, the grasping constraint can be replaced by the following scalar holonomic constraint:

$$\forall t \in [0,1], \quad F\big(a_1(t), a_2(t), x_m(t), y_m(t)\big) = 0$$

with

$$F(a_1, a_2, x_m, y_m) = \min\left[\sqrt{\left(x_{eff} - x_m\right)^2 + \left(y_{eff} - y_m\right)^2}, \min_{u \in \{1,2\}} \sqrt{\left(x_m - x_m^u\right)^2 + \left(y_m - y_m^u\right)^2}\right]$$

We define a decreasing sequence of positive numbers $\epsilon_k, k > 0$ converging towards 0. For example, we can choose $\epsilon_k = 1/k$. We replace the pick and place problem P defined above by the sequence of problems P_k for which the grasping constraint is replaced by:

$$\forall t \in [0, 1], \quad F\big(a_1(t), a_2(t), x_m(t), y_m(t)\big) < \epsilon_k$$

We see that the free space for problem P_k is an open subset of the configuration space. Hence, we can use a standard path planner to solve P_k.

The idea underlying the penalty function method is that, since the difficulty of solving P_k increases with k, it is better to first solve problem P_1, and then use the solution path of problem P_1 as a heuristic to solve problem P_2. In turn, the solution hopefully obtained of problem P_2 can be used as an input heuristic for solving problem P_3, and so on until the grasping constraint is satisfied up to a prespecified accuracy $\epsilon_{k_{max}} = 1/k_{max}$.

Hence, the penalty function method will work well if one can devise a path planner that makes efficient use of the solution path of problem P_{k-1} in order to solve problem P_k. Such a path planner is called a *variational* path planner. Variational planning is discussed in the next subsection.

4.2 Variational planning

Variational path planning consists of iteratively improving an initial path possibly colliding with obstacles (or any other inequality constraint) in order to obtain a collision free path.

We use the method of Variational Dynamic Programming (VDP) developed by Barraquand and Ferbach 1993 [2]. VDP consists in perturbing at each iteration the current path by performing a dynamic programming search around the current path in a m-dimensional submanifold of the n-dimensional configuration space $\mathcal{C}$. In practice, m is chosen equal to 2,3, or 4, in order to make the dynamic programming search tractable. The m-dimensional submanifold is an arbitrarily chosen ruled surface containing the current path. This surface is quantized into a m-dimensional grid of configurations. Then, the grid is searched using Dijkstra's algorithm with an additive cost function proportional to the number of configurations colliding with obstacles. Hence, the new path obtained is guaranteed to collide obstacles less than the previous path. Then, the operation is repeated until a free path is found. See Barraquand and Ferbach 1993 [2] for more detail on the VDP algorithm.

Experiments reported in Section 5 show that VDP is well suited to using the output solution of problem P_{k-1} in order to solve efficiently problem P_k. However, other planners capable of planning paths in high dimensional configuration spaces could be adapted in theory to imbed the same feature. For example, the RPP planner described in Barraquand and Latombe 1991 [5] is not a variational planner in its original form, but could be adapted in the following fashion.

Given a solution path γ_{k-1} of problem P_{k-1}, and since the solution of problem P_k should not differ significantly from γ_{k-1}, one could estimate the maximum distance d_{max} over all $t \in [0, 1]$ between configuration $\gamma_{k-1}(t)$ and the corresponding configuration $\gamma_k(t)$ on a solution path for problem P_k. Then, γ_{k-1} could be used to include the additional heuristic constraint on the free configuration space $\mathcal{C}^k_{free}$ of problem P_k:

$$\forall \mathbf{q} \in \mathcal{C}^k_{free}, \quad \exists t \in [0,1], d\big(\mathbf{q}, \gamma_{k-1}(t)\big) < d_{max}$$

This additional heuristic constraint would force the path planner RPP to limit its search for a solution path γ_k of problem P_k to a "tubular neighborhood" of the previous solution path γ_{k-1}. We plan to investigate this alternative in future research.

4.3 General case

We consider the general manipulation planning problem of Section (3). We assume a finite number of docking positions for the objects have been prespecified as part of the planning problem. From the results of subsection (3.3), and more specifically from formula (4), we see that the grasping constraints for object M_j can be expressed:

$$\forall t \in [0,1], \quad F_j\big(\gamma(t)\big) = 0$$

with

$$F_j(\mathbf{q}) = \min\left(\min_{i \in [1,l]}\Big(||g^{R_i,M_j}_{grasp}(\pi^{R_i}(\mathbf{q}), \pi^{M_j}(\mathbf{q}))||\Big), \min_{D \in \mathrm{DOCK}(M_j)}\Big(d\big(\pi^{M_j}(\mathbf{q}), D\big)\Big)\right)$$

For each object M_j, we can choose a positive decreasing sequence of numbers ϵ^j_k converging towards zero, an replace our original manipulation planning problem P by a sequence of problems P_k using the partially relaxed grasping constraints:

$$\forall j \in [1,s], \quad \forall t \in [0,1], \quad F_j\big(\gamma_k(t)\big) < \epsilon^j_k$$

The penalty function method consists in first solving the simpler problem P_1 as a standard path planning problem in open free space using a variational path planner such as VDP, to obtain a first path γ_1. Then, for any $k > 1$, problem P_k is solved using as input to the variational planner the solution γ_{k-1} of the previous problem P_{k-1}. The algorithm is stopped when the numbers $E_k = \big(\epsilon^1_k, \ldots, \epsilon^s_k\big)$ are all smaller than a prespecified tolerance value $\epsilon_{k_{max}}$. The overall method is called *Progressive Variational Dynamic Programming* (PVDP), since the penalties on the grasping constraints are applied in a progressive fashion.

The vector sequence $\big(E_k\big)_{k>0}$ is called ϵ-strategy. Both the rate of convergence towards 0 and the relative values for a given k of the various numbers $\epsilon^1_k, \ldots, \epsilon^s_k$ may influence the overall computation time and even the nature of the solution found by the planner.

More complex ϵ-strategies can be devised. Indeed, many different functions F_j can be used to represent the same holonomic grasping constraints applying to the object M_j. For example, when the dimension h of a given grasp between robot R_i and object M_j is higher than 1, one may choose any other equivalent norm instead of the standard L^2 Euclidean norm to compute the number $||g_{grasp}^{R_i,M_j}||$. One may find it preferable to use a weighted L^2 norm instead of the standard norm:

$$||(g_1,\ldots,g_h)^T||_{L^2}^{weighted} = \sqrt{\sum_{i=1}^{h} \alpha_i g_i^2}$$

where $\alpha_1,\ldots,\alpha_h$ are adequately defined positive numbers depending on the type of grasp considered. Alternatively, one may use the weighted L^∞ norm:

$$||(g_1,\ldots,g_h)^T||_{L^\infty}^{weighted} = \max_{i\in[1,h]} \alpha_i |g_i|$$

Also, any other additional heuristic constraints can be added to improve the progressiveness in the difficulty of the sequence of problems P_k when facing a particularly complex manipulation problem. In general we call ϵ-strategy the whole set of empirical parameters that can be used to define the holonomic constraints F_j and their rate of convergence towards 0. Examples of efficient problem-specific strategies are presented in Section (5).

5 Experimental results

We have implemented PVDP in a program written in C, running on a DEC3000-500 Alpha AXP workstation. We have experimented with PVDP using a variety of manipulation planning problems. The most significant of these experiments are described below.

5.1 10-DOF non-serial manipulator robot grasping a 2-DOF disk.

We consider the planar non-serial manipulator robot depicted in Figure 2, which includes three prismatic joints (telescopic links) and seven revolute joints. The task assigned to this robot is a simple pick and place operation consisting in grasping the disk in the lower right corner of the workspace, bringing it to the lower left corner, and then returning to its initial configuration. The total number of degrees of freedom for the whole problem is 12. Figure 2 illustrates a manipulation plan found by PVDP.

In this example, the robot is said to have grasped the disk when the following conditions are satisfied:

- the center M of the disk coincides with the middle $R = \frac{E_1+E_2}{2}$ of the two end-effectors E_1 and E_2 of the robot.
- the distance $||E_1E_2||$ between the two end effectors E_1 and E_2 is equal to the diameter D of the disk.

There are two admissible docking positions M_1 and M_2 for the disk, namely its initial and goal configurations.

The grasping constraint $\forall t, F(\gamma(t)) = 0$ is replaced in the approximating problem P_ϵ by the constraint $\forall t, F(\gamma(t)) < \epsilon$ with the following expression for $F(\mathbf{q})$.

$$F(\mathbf{q}) = \min\left(\max(||E_1E_2|| - D, ||RM||), \min_{i\in\{1,2\}} ||MM_i||\right)$$

In other words, in problem P_ϵ, either the disk is at distance less than ϵ of a docking position, or if satisfies *both* conditions $||E_1E_2|| < D + \epsilon$ *and* $||RM|| < \epsilon$.

We denote by F_1E_1 the segment representing one of the two terminal fingers of the robot, F_2E_2 the other one. In addition to the ϵ-strategy defined above, we use in problem P_ϵ the following heuristic constraint, systematically satisfied in the original definition of the grasp. If M is at distance longer than ϵ from both M_1 and M_2, it must always lie in the domain of the workspace bounded by 1) the two straight lines prolongating the two robot fingers, i.e. the lines passing through F_1, E_1 and F_2, E_2, and 2) the line F_1F_2. This additional heuristic constraint enhances the progressiveness in the difficulty of problem P_ϵ.

The initial value of ϵ is one fourth of the size of the workspace. Then, it is decreased at each iteration of the penalty function method by 0.001, i.e. 0.1% of the size of the workspace. The tolerance value was set to $\epsilon_{k_{max}} = 0.006$, i.e 0.6% of the workspace. The path was computed in about half an hour.

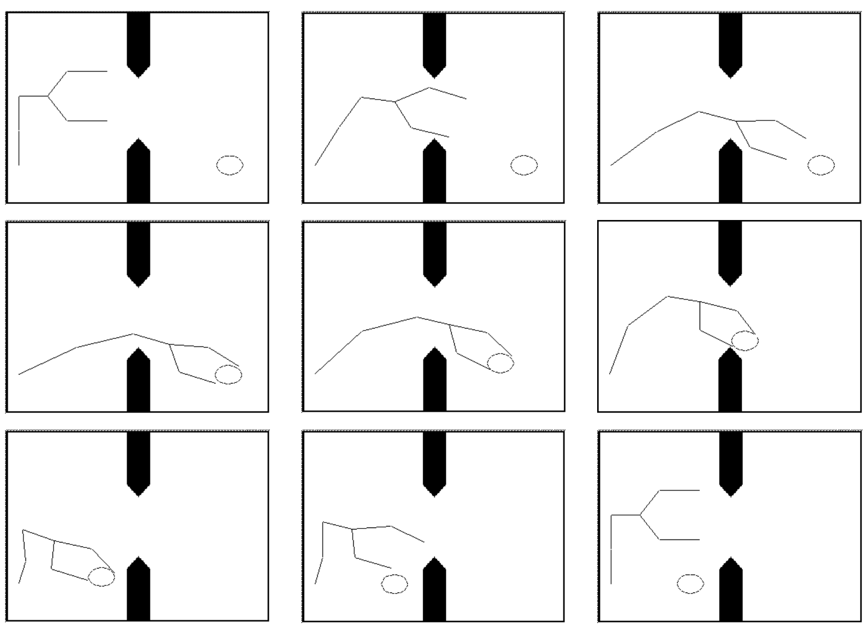

Figure 2: A pick and place operation using a 10-DOF non-serial manipulator

5.2 3-DOF serial manipulator robot manipulating two 2-DOF disks.

We consider the planar serial manipulator robot depicted in Figure 3, which includes two revolute joints and one telescopic link. The robot is composed of 1) a single telescopic arm, and 2) an end effector that can rotate at the end of the telescopic arm. The task assigned to this robot is to move both disks from the right-hand bucket to the left-hand bucket on the bottom of the workspace. Additionally, the two disks must be put in the left-hand bucket in a last-in first-out order, i.e. the disk on the top of the right-hand bucket at the initial configuration must be on the top of the left-hand bucket at the final configuration. Hence, the robot must first move the top disk to an intermediate docking position, then put the disk at the bottom of the right-hand bucket into the left-hand bucket, and finally bring the top disk from its intermediate position to the top of the left-hand bucket.

The total number of degrees of freedom for the whole problem is 7. Figure 3 illustrates a manipulation plan found by PVDP.

Let E_1 and E_2 be the two points at the extremities of the two fingers on the end-effector. Let M be the center of any one of the two disks. Let D be the diameter of the disks. In this example, the robot is said to have grasped a disk iff the distances E_1M and E_2M are both equal to $D/2$.

There are three docking positions for the top disk: 1) the initial position, 2) the goal position, 3) the intermediate docking position on top of the central obstacle separating the two buckets (see figure (3)). There are two docking positions for the bottom disk: 1) the initial position, 2) the goal position.

The grasping constraints in the partially relaxed problems P_ϵ were defined in a fashion similar to that of the previous example. The total computation time in the example shown in figure (3) was 45 minutes. The planner PVDP was also capable of solving the same problem using a randomly selected intermediate docking position on top of the obstacles, instead of the prespecified position in the center (Figure 4).

5.3 Two 3-DOF manipulator arms manipulating a 3-DOF bar

We consider the dual-arm manipulating planning problem depicted in Figure 5. There are two identical 3-DOF arms. Each arm has two links. The first link has fixed length, and can rotate around the base. The second link is telescopic and has two degrees of freedom, one revolute and one prismatic. The bar has 3-DOF. The bar can only move when it is grasped by both arms. The task assigned to the two arms is to move the bar from the right side to the left side of the workspace. Because of the presence of an obstacle, the two arms must swap their respective grasps before reaching the goal. The total number of degrees of freedom for the whole problem is 9. Figure 5 illustrates a manipulation plan found by PVDP. The path was computed in about 40 minutes.

The definitions of the grasping constraints and of the approximating problems P_ϵ are similar to that of the previous examples. Three docking positions are allowed for the bar: its initial

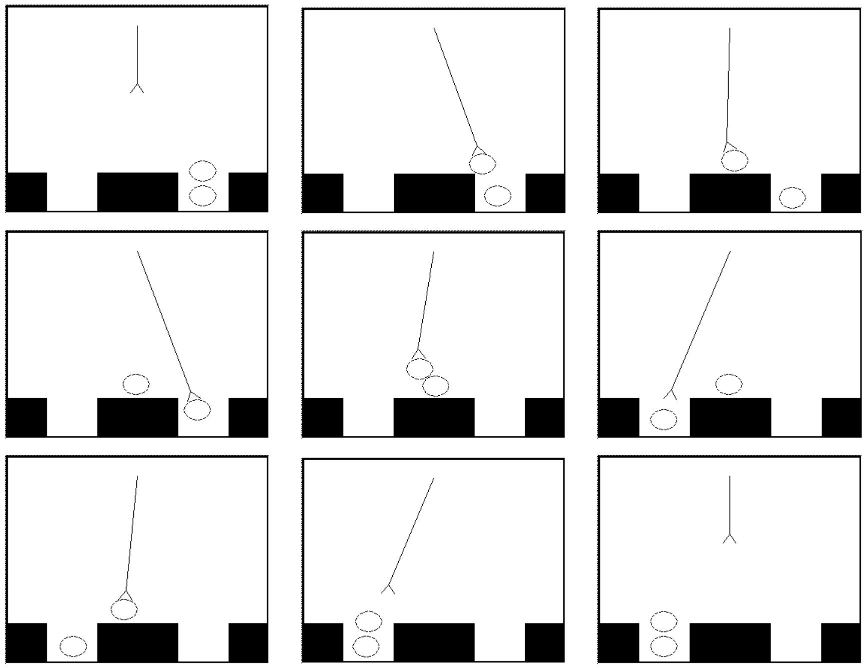

Figure 3: A 3-DOF arm manipulating two disks

position, its goal position, and an intermediate vertical position in the middle of the workspace. The specification of this intermediate position is critical for the success of the manipulation plan. We have chosen this intermediate docking position in an ad-hoc fashion. If this intermediate position had been randomly chosen in the workspace, the planner would have most probably failed to find a manipulation plan. This example demonstrates that the choice of intermediate docking positions is a serious limitation of our manipulation planner in its current implementation. In Section (6), we briefly discuss how intermediate docking positions could be dynamically determined by the planner itself in a less ad-hoc fashion.

5.4 Three robots manipulating a 2-DOF disk

We consider the planar manipulation planning problem depicted in Figure 6. The robot on the lower left side is a kind of piston that can extend vertically. It has 1 DOF, which corresponds to the prismatic extension link. It can grasp the 2-DOF disk on the middle of the horizontal shelf. Another robot with 1 DOF on the upper left corner can only rotate around its base. The third manipulator is composed of one telescopic arm, another arm with fixed length and two revolute joints. This manipulator has 3 DOF. The task assigned to these manipulators is to move the 2-DOF disk from the right side to the left side of the picture. At the end the disk is placed on the piston in the lower left corner.

The total number of degrees of freedom for the whole problem is 7. Figure 6 illustrates a

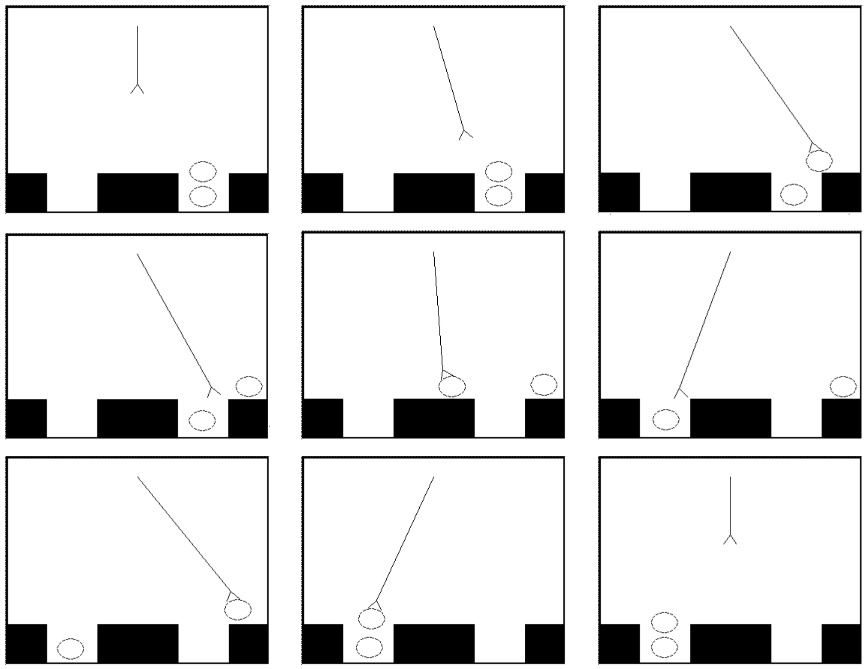

Figure 4: The same problem with a randomly chosen intermediate docking position

manipulation plan found by PVDP in 13 minutes.

This example shows a solution where several manipulators have to cooperate with each other: one robot picks the disk and brings it to a second manipulator that gives the object to a third one. The solution has been found without using any docking position or predefining the positions where the manipulators transmit the disk between each other.

6 Discussion

6.1 Current limitations and possible extensions

6.1.1 Automatic selection of docking positions

The examples presented above show that the penalty function method is capable of dealing with manipulation planning problems of unprecedented complexity. However, the method has a severe limitation in its current form: an appropriate set of docking positions must be chosen for each manipulation problem. Often, a simple and natural choice consists in considering only docking positions that correspond to the initial or goal configurations for the movable objects. But in some more complex problems, intermediate docking positions must be prespecified in a somewhat ad-hoc fashion. We believe this limitation of the current planning algorithm could

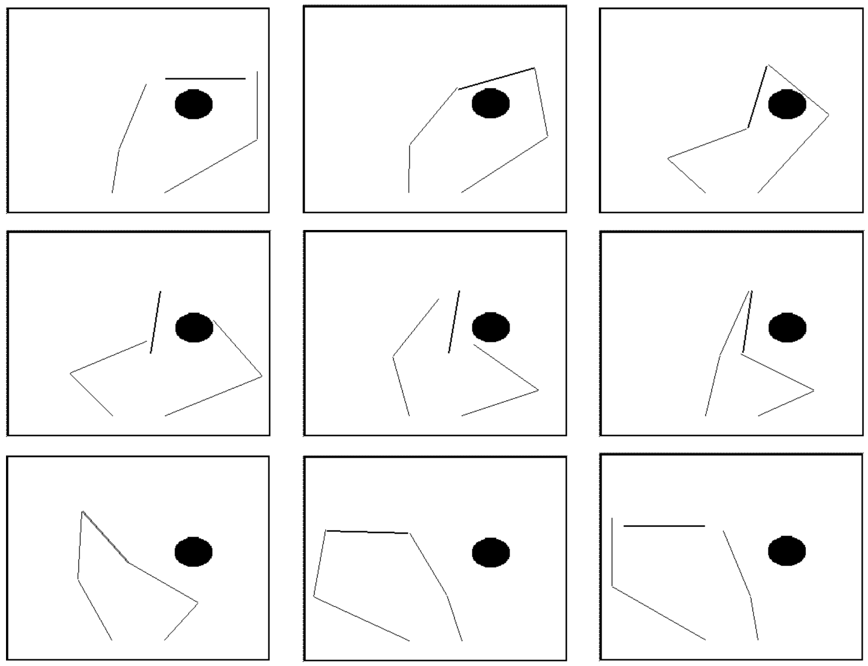

Figure 5: A dual arm manipulation planning problem

be removed by defining the set of docking positions dynamically while the manipulation plan is constructed. Indeed, one could define the intermediate docking positions in the following way. Using the notations of Subsection (4.3), and given an object M_j, we can consider the grasping function:

$$G_j(\mathbf{q}) = \min_{i \in [1,l]} ||g_{grasp}^{R_i,M_j}(\pi^{R_i}(\mathbf{q}), \pi^{M_j}(\mathbf{q}))||$$

Then, we can subdivide the current solution path γ_k of problem P_k into a sequence $0 = t_0 < t_1 < \ldots < t_{2r} < t_{2r+1} = 1$ verifying:

$$\forall p \in [0, r], \forall t \in [t_{2p}, t_{2p+1}], G_j(\gamma_k(t)) \geq \epsilon_k^j$$

and

$$\forall p \in [0, r-1], \forall t \in]t_{2p+1}, t_{2p+2}[, G_j(\gamma_k(t)) < \epsilon_k^j$$

In other words, we can subdivide the current path into subintervals where the object is successively close (i.e. at distance less than ϵ_k^j) to some robot end-effector, of far from any robot's end-effector. Then, for each subinterval $[t_{2p}, t_{2p+1}]$ for which the object is far from all robots, we can define dynamically the intermediate docking position in the middle of the subinterval:

$$D_p^j = \pi^{M_j}(\gamma_k(\frac{t_{2p} + t_{2p+1}}{2}))$$

The $r + 1$ intermediate docking positions $D_0^j, \ldots, D_r^j$ then defined can be used as the new set of docking positions for problem P_{k+1}. We plan to investigate this idea in future research.

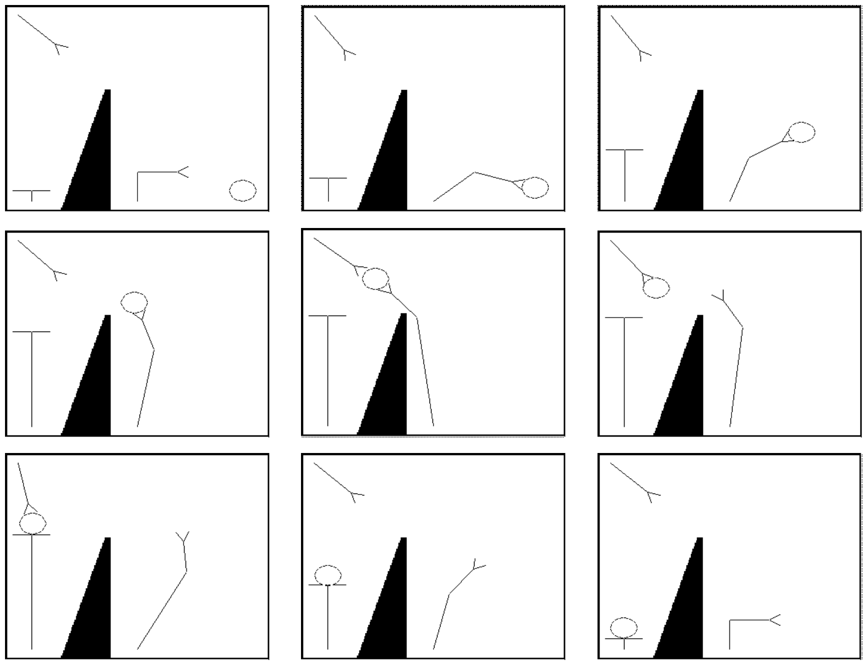

Figure 6: A pick and place operation with three cooperating robots

6.1.2 Computation of efficient ϵ-strategies

Besides the problem of defining intermediate docking positions, another important issue is to define an appropriate ϵ-strategy. Although we have found that the planner is relatively robust with regards to the choice of the ϵ-strategy, future research is needed to reduce the number of empirical parameters associated with the definition of the ϵ-strategies. Devising efficient computational techniques for automatically selecting the appropriate strategies is an open problem.

6.1.3 Combining the penalty function method with other path planners

Finally, although the method of Variational Dynamic Programming has proven efficient for solving the subproblems P_k, other variational planners could be used in theory to solve the same problem. In Subsection (4.2), we have briefly described how standard planners such as RPP capable of dealing with many degrees of freedom could possibly be adapted for this purpose. We believe this idea is also worth investigating.

6.2 Other applications of the penalty function method

6.2.1 Application to standard path planning problems

Initially, this penalty function approach was developed in Barraquand and Ferbach 1993 [2] as a variant of the VDP method for solving difficult instances of the basic path planning problem in open free space. Indeed, given any variational path planning method, one can replace the collision avoidance constraints in a classical path planning problem by a convergent series of simpler constraints increasingly penalizing motions that do not satisfy collision avoidance constraints. The resulting planner is less general in theory than the original VDP planner, since it uses problem-specific heuristics to guide the search. On the other hand, it is dramatically faster. In fact, it can solve some problems in a time comparable to that of potential-field based methods (see Barraquand and Ferbach 1993 [2] for more detail).

6.2.2 Applications in assembly planning

More generally, the penalty function method can be used to represent *any* kind of holonomic constraints in motion planning. This opens a broad range of new possibilities. In particular, most constraints in assembly planning problems are holonomic equality constraints. Hence, an assembly planning problem can in theory be approximated by a sequence of simpler problems P_ϵ where the different objects to be assembled are allowed to overlap with each other by a distance not greater than ϵ. Although we do not have investigated the application of the penalty function method to assembly planning, we feel that this line of research is particularly promising.

7 Conclusion

This report described a new approach to motion planning with holonomic constraints, which essentially consists of replacing a constrained problem by a convergent series of less constrained subproblems increasingly penalizing the motions that do not satisfy the constraints. Each subproblem is solved using a variational path planner.

We have applied the approach to manipulation planning problems in the presence of holonomic grasping constraints. To this end, we have characterized the conditions under which grasping constraints can be made holonomic. In practice, a grasping constraint on a movable object is holonomic if the number of stable reachable configurations is finite.

This approach has been implemented in a program, called PVDP, which was run successfully on several difficult manipulation planning problems. We used the method of Variational Dynamic Programming to solve the subproblems, although other variational planners could be used in theory.

The penalty function method could be applied to any other kind of holonomic constraint. A very promising line of research is the application of this method to assembly planning problems.

References

1. Alami, R. , Simeon, T., and Laumond J.P. "A Geometrical Approach to Planning Manipulation Tasks: The Case of Discrete Placements and Grasps." *Robotics Research 5*, Miura and Arimoto, eds., MIT Press, pp 453-459, 1989.

2. Barraquand, J., and Ferbach, P. *Path Planning through Variational Dynamic Programming* Research Report 33, Paris Research Laboratory, Digital Equipment Corp., September 1993.

3. Barraquand, J., Langlois, B. and Latombe, J.C. "Numerical Potential Field Techniques for Robot Path Planning." *IEEE Transactions on Systems, Man, and Cybernetics*, vol 22-2, 1992.

4. Barraquand, J. and Latombe, J.C. "On Nonholonomic Mobile Robots and Optimal Maneuvering." *Revue d'Intelligence Artificielle*. Hermes. 3(2):77-103, 1989.

5. Barraquand, J. and Latombe, J.C. "Robot Motion Planning: A Distributed Representation Approach." *International Journal of Robotics Research*, MIT Press, 10-6, 1991.

6. Barraquand, J. and Latombe, J.C. "NonHolonomic MultiBody Mobile Robots: Controllability and Motion Planning in the Presence of Obstacles." *Algorithmica*, vol 10, number 2/3/4, 1993.

7. Koga, Y. and Latombe, J.C. "Experiments in Dual-Arm Manipulation Planning". *Proceedings of the IEEE International Conference on Robotics and Automation*, Nice, France, pp 2238-2245, May 1992.

8. Latombe, J.C. *Robot Motion Planning*. Kluwer Academic Publishers, Boston, 1990.

9. Laumond, J.P. "Feasible Trajectories for Mobile Robots with Kinematic and Environment Constraints." *Proc. International Conference on Intelligent Autonomous Systems*, Amsterdam (1986), 346-354.

10. Laumond, J.P. and Alami, R. *A Geometrical Approach to Planning Manipulation Tasks in Robotics*. Technical Report 89-261, LAAS, Toulouse, France, 1989.

11. Li, Z. and Canny, J.F. *Robot Motion Planning with Nonholonomic Constraints*. Memo UCB/ERL M89/13, Electronics Research Laboratory, University of California, Berkeley, February 1989.

12. Wilfong, G. "Motion Planning in the Presence of Movable Obstacles." *Proceedings of the 4th ACM Symposium of Computational Geometry*, pp 279-288, 1988.

PRL Research Reports

The following documents may be ordered by regular mail from:

Librarian – Research Reports
Digital Equipment Corporation
Paris Research Laboratory
85, avenue Victor Hugo
92563 Rueil-Malmaison Cedex
France.

It is also possible to obtain them by electronic mail. For more information, send a message whose subject line is `help` to `doc-server@prl.dec.com` or, from within Digital, to `decprl::doc-server`.

Research Report 1: *Incremental Computation of Planar Maps*. Michel Gangnet, Jean-Claude Hervé, Thierry Pudet, and Jean-Manuel Van Thong. May 1989.

Research Report 2: *BigNum: A Portable and Efficient Package for Arbitrary-Precision Arithmetic*. Bernard Serpette, Jean Vuillemin, and Jean-Claude Hervé. May 1989.

Research Report 3: *Introduction to Programmable Active Memories*. Patrice Bertin, Didier Roncin, and Jean Vuillemin. June 1989.

Research Report 4: *Compiling Pattern Matching by Term Decomposition*. Laurence Puel and Ascánder Suárez. January 1990.

Research Report 5: *The WAM: A (Real) Tutorial*. Hassan Aït-Kaci. January 1990.†

Research Report 6: *Binary Periodic Synchronizing Sequences*. Marcin Skubiszewski. May 1991.

Research Report 7: *The Siphon: Managing Distant Replicated Repositories*. Francis J. Prusker and Edward P. Wobber. May 1991.

Research Report 8: *Constructive Logics. Part I: A Tutorial on Proof Systems and Typed λ-Calculi*. Jean Gallier. May 1991.

Research Report 9: *Constructive Logics. Part II: Linear Logic and Proof Nets*. Jean Gallier. May 1991.

Research Report 10: *Pattern Matching in Order-Sorted Languages*. Delia Kesner. May 1991.

† This report is no longer available from PRL. A revised version has now appeared as a book: "Hassan Aït-Kaci, *Warren's Abstract Machine: A Tutorial Reconstruction. MIT Press, Cambridge, MA (1991).*"

Research Report 11: *Towards a Meaning of LIFE*. Hassan Aït-Kaci and Andreas Podelski. June 1991 (Revised, October 1992).

Research Report 12: *Residuation and Guarded Rules for Constraint Logic Programming*. Gert Smolka. June 1991.

Research Report 13: *Functions as Passive Constraints in LIFE*. Hassan Aït-Kaci and Andreas Podelski. June 1991 (Revised, November 1992).

Research Report 14: *Automatic Motion Planning for Complex Articulated Bodies*. Jérôme Barraquand. June 1991.

Research Report 15: *A Hardware Implementation of Pure Esterel*. Gérard Berry. July 1991.

Research Report 16: *Contribution à la Résolution Numérique des Équations de Laplace et de la Chaleur*. Jean Vuillemin. February 1992.

Research Report 17: *Inferring Graphical Constraints with Rockit*. Solange Karsenty, James A. Landay, and Chris Weikart. March 1992.

Research Report 18: *Abstract Interpretation by Dynamic Partitioning*. François Bourdoncle. March 1992.

Research Report 19: *Measuring System Performance with Reprogrammable Hardware*. Mark Shand. August 1992.

Research Report 20: *A Feature Constraint System for Logic Programming with Entailment*. Hassan Aït-Kaci, Andreas Podelski, and Gert Smolka. November 1992.

Research Report 21: *The Genericity Theorem and the Notion of Parametricity in the Polymorphic λ-calculus*. Giuseppe Longo, Kathleen Milsted, and Sergei Soloviev. December 1992.

Research Report 22: *Sémantiques des langages impératifs d'ordre supérieur et interprétation abstraite*. François Bourdoncle. January 1993.

Research Report 23: *Dessin à main levée et courbes de Bézier : comparaison des algorithmes de subdivision, modélisation des épaisseurs variables*. Thierry Pudet. January 1993.

Research Report 24: *Programmable Active Memories: a Performance Assessment*. Patrice Bertin, Didier Roncin, and Jean Vuillemin. March 1993.

Research Report 25: *On Circuits and Numbers*. Jean Vuillemin. April 1993.

Research Report 26: *Numerical Valuation of High Dimensional Multivariate European Securities*. Jérôme Barraquand. March 1993.

Research Report 27: *A Database Interface for Complex Objects*. Marcel Holsheimer, Rolf A. de By, and Hassan Aït-Kaci. March 1993.

Research Report 28: *Feature Automata and Sets of Feature Trees*. Joachim Niehren and Andreas Podelski. March 1993.

Research Report 29: *Real Time Fitting of Pressure Brushstrokes*. Thierry Pudet. March 1993.

Research Report 30: *Rollit: An Application Builder*. Solange Karsenty and Chris Weikart. April 1993.

Research Report 31: *Label-Selective λ-Calculus*. Hassan Aït-Kaci and Jacques Garrigue. May 1993.

Research Report 32: *Order-Sorted Feature Theory Unification*. Hassan Aït-Kaci, Andreas Podelski, and Seth Copen Goldstein. May 1993.

Research Report 33: *Path Planning through Variational Dynamic Programming*. Jérôme Barraquand and Pierre Ferbach. September 1993.

Research Report 34: *A penalty function method for constrained motion planning*. Pierre Ferbach and Jérôme Barraquand. September 1993.

Research Report 35: *The Typed Polymorphic Label-Selective λ-Calculus*. Jacques Garrigue and Hassan Aït-Kaci. October 1993.

34

A penalty function method for constrained motion planning

Pierre Ferbach and Jérôme Barraquand

digital

PARIS RESEARCH LABORATORY

85, Avenue Victor Hugo
92563 RUEIL MALMAISON CEDEX
FRANCE