Planning Motion Compliant to Complex Contact States *

Xuerong Ji, Jing Xiao
Computer Science Department
University of North Carolina - Charlotte
Charlotte, NC 28223, USA
xji@uncc.edu, xiao@uncc.edu

Abstract

Many robotic tasks require compliant motion, but planning such motion poses special challenges not present in collision-free motion planning. One challenge is how to achieve exactness, i.e., how to make sure that a planned path exactly compliant to a desired contact state, especially when the configuration manifold of such a contact state is hard to describe analytically due to high geometrical complexity and/or high dimensionality. We tackle the problem with a hybrid approach of direct computation to exploit contact constraints and randomized planning. In this paper, we describe such a planner for planning motion compliant to a contact formation between two arbitrary polyhedra and present results of implementation.

1 Introduction

Motion compliant to contact occurs on the boundary of configuration space obstacles (C-obstacles) [13], but computing C-obstacles of high dimensions remains a formidable task to date. While there were exact descriptions of C-obstacles for polygons [1, 2], there were only approximations for polyhedra [4, 9]. Hence, it is desirable to explore contact motion planning without explicitly computing C-obstacles [5].

Recently the authors introduced a general divide-and-merge approach for automatically generating a contact state graph between arbitrary polyhedra [16]. Each node in the graph denotes a contact state, described by a *contact formation* (CF) [15] and a representative configuration of the CF, and adjacent states are connected by edges in the graph. With this approach, the problem of contact motion planning is effectively simplified as graph search at high-level for state transitions and motion planning at low-level within one state, i.e., within the set of contact configurations constrained by the *same* CF[1].

It is not trivial to plan a path of configurations constrained by a CF, or a CF-compliant path, if the configuration manifold (i.e., C-surface patch) $\mathcal{C}_{\mathcal{CF}}$ of such a CF is hard to describe analytically due to high geometrical complexity and/or high dimensionality. We tackle the problem with a general approach combining exploitation of the contact constraints of a CF and randomized planning. First, we have developed a strategy to sample configurations satisfying certain constraints of a CF by a hybrid of direct computation based on contact constraints and random sampling to maximize efficiency [7, 8]. In this paper, we describe how to use the sampling results to plan a CF-compliant path of configurations.

The paper is organized as the following. In Section 2, we review the notion of *contact formation* (CF) in terms of *principal contacts* (PC) [15] to characterize topological contacts and define concepts related to configurations constrained by a CF. In Section 3, we describe how to plan a CF-compliant path based on random samples on $\mathcal{C}_{\mathcal{CF}}$ [7, 8] and the paradigm of probabilistic road maps (PRM) [10, 11]. We present some planning results for CFs between polyhedral objects in Section 4 and conclude the paper in Section 5.

2 CF and Related Configurations

In this section, we review the concept of contact formations to characterize topological contact states and the associated constraints, and then define concepts describing the contact configurations constrained by a contact formation.

*This research is supported by the National Science Foundation under grant IIS-9700412.

[1]Note that a general contact motion crossing several contact states consists of segments of motion in each contact state.

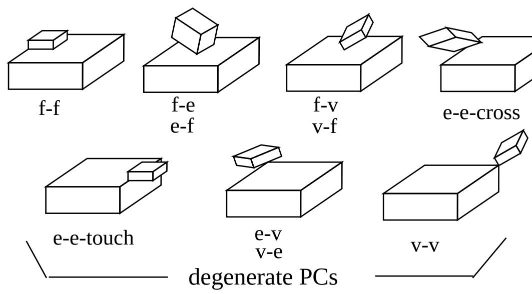

Figure 1: Principal Contacts (PCs)

2.1 PC and CF

Denoting the *boundary elements* of a face as the edges and vertices bounding it, and the boundary elements of an edge as the vertices bounding the edge, A *principal contact* (PC) is defined as the contact between a pair of surface elements (i.e., faces, edges, or vertices) which are not the boundary elements of other contacting surface elements (if there is more than one pair in contact) [15]. There are ten types of PCs between arbitrary polyhedra, as shown in Fig. 1. Each non-degenerate PC is associated with a *contact plane*, defined by a contacting face or the two contacting edges in an e-e-cross PC. Each degenerate PC is characterized as between two convex edge or vertex elements and not being associated with a contact plane[2].

With the notion of PC, an arbitrary contact between arbitrary polyhedra is described as the set of PCs formed, called a *contact formation* (CF).

2.2 Configurations Constrained by a CF

Since a CF consists of a set of PCs, let us first consider the contact configurations constrained by a PC.

As introduced in [16], the *geometrical representation of a PC* (GeoPC), $\mathcal{C}_{PC}$, is the set of configurations (or configuration manifold) which satisfy the contact condition characterized by the PC's topological definition. Such a contact condition implies a *constraint equation* and *constraint inequalities* imposed on the contact configurations in $\mathcal{C}_{PC}$. The constraint equation of a PC describes the fact that the two contacting elements of the PC are on the same point (if the PC is a v-v type, see Fig. 1), the same line (if the PC is a v-e/e-v or an e-e-touch type), or the same plane (if the PC is an f-*/*-f or an e-e-cross type), which can be easily written as shown in [7].

On the other hand, the constraint inequalities of a PC limits the values of every configuration variable to a feasible range that maintains the PC without causing additional contacts or penetrations. The inequality constraints on positional and orientational variables are often inter-dependent and highly dependent on the specific contact geometry so that it is hard to express them analytically. Thus, obtaining configurations in $\mathcal{C}_{PC}$ is not a trivial task. To overcome the difficulty, we find it useful to first find configurations satisfying certain localized constraints of a PC and then eliminate those not in $\mathcal{C}_{PC}$ later. Hence, we introduce the following two concepts.

Definition 1: A configuration C is *PC-compliant* to a PC a-b, where a and b are the two contacting surface elements of objects A and B respectively, iff the following conditions are held:

- C satisfies the constraint equation of the PC,
- In C, $\breve{a} \cap \breve{b} \neq \emptyset$, where $\breve{a}$ and $\breve{b}$ are the *open* set of a and b respectively[3],
- there is no additional contact or penetration between a or any of a's adjacent elements and b or any of b's adjacent elements other than the PC, i.e., there is no *local collision*.

Definition 2: A configuration C is *a feasible PC-compliant* configuration to a PC, iff C is in the GeoPC $\mathcal{C}_{PC}$. If C is PC-compliant but not in $\mathcal{C}_{PC}$, it is *infeasible.*

Fig. 2 shows examples of configurations that are (1) feasible PC-compliant, (2) infeasible PC-compliant, and (3) not PC-compliant, with respect to a v-f PC.

Now we can define similar concepts with respect to a CF. The *geometrical representation of a CF* (GeoCF), $\mathcal{C}_{CF}$, denotes the set of contact configurations which satisfy the contact constraints of every PC in the CF. Obviously, $\mathcal{C}_{CF}$ is $\cap \mathcal{C}_{PC_i}, \forall PC_i \in CF$.

Definition 3: A configuration C is *CF-compliant* to a CF iff it is PC-compliant to every PC in the CF.

Definition 4: A CF-compliant configuration C is *a feasible CF-compliant configuration* to a CF iff C is in the GeoCF of the CF. C is infeasible to the CF otherwise.

[2] Note that a contact between a convex edge/vertex and a concave edge/vertex is regarded not as a single PC but as consisting of a couple non-degenerate PCs.

[3] The open set of a vertex is the vertex itself, the open set of an edge is the edge without boundary vertices, and the open set of a face is the face without boundary vertices and edges.

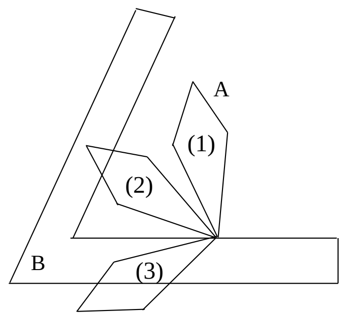

Figure 2: Configurations of A that are (1) feasible PC-compliant, (2) infeasible PC-compliant, and (3) not PC-compliant to a v-f PC

3 CF-Compliant Planning

The problem of CF-compliant planning can be defined as: given a CF, an initial configuration C_i and a goal configuration C_g in GeoCF, find a path of feasible CF-compliant configurations that connect C_i and C_g. We have addressed the problem by extending the PRM [10, 11] approach of randomized planning for collision-free motions to the space of CF-compliant contact configurations.

To plan collision-free paths, the PRM approach builds a random "roadmap" in the configuration space (C-space) of the moving object/robot: a graph consisting of collision-free configurations as nodes and arcs connecting two nodes if there exists a straight-line path of collision-free configurations between them. A randomly generated configuration becomes a node in the graph if it is collision-free, and if a sequence of linearly interpolated configurations between two nodes are all collision-free, an arc is added to connect the two nodes. With such a roadmap, a collision-free path can be found by graph-search (of the map). The approach is efficient because (1) it does not need to compute C-obstacles (which is often too hard) and (2) it takes advantage of randomness to achieve good coverage of the C-space.

To extend the PRM approach to the space of contact configurations constrained by a CF, we have tackled the following special issues:

1. to generate random CF-compliant configurations;
2. to check if a CF-compliant configuration is feasible, i.e., with no other collision than the CF;
3. to check if two feasible CF-compliant configurations can be connected by an arc in the roadmap by "compliant interpolation," i.e., to ensure that a sequence of interpolated configurations are CF-compliant, and if all such configurations are also feasible, the two nodes are connected by an arc.

We have reported efficient strategies to resolve the first issue in [7, 8], we now describe how we resolve the second and the third issues in the rest of the section.

3.1 Feasibility Checking

As defined in Section 2, a CF-compliant configuration is feasible if, at the configuration, there is no *other* collision/contact between the two objects than the PCs of the CF. Otherwise, it is infeasible. To check the feasibility of a given CF-compliant configuration, our strategy is to use a standard collision detection tool to detect all collisions between the two objects at the configuration and then discard those caused by the CF to see if there is any collision left.

Specifically, we started by using RAPID, developed by Lin, Manocha, et al.[12], to detect collisions. RAPID is a robust and efficient software that takes as input the triangulation results of two arbitrary solids, i.e., surface triangles and outputs the triangle pairs which are in collision at a given configuration. We used the triangulation program implemented by Narkhede and Manocha based on [14]. The program does not introduce additional vertices, i.e., Steiner points, to triangulate faces, and it can handle faces with holes.

We then designed an algorithm to distinguish, from the output of RAPID applied to a CF-compliant configuration C_c, if the collision of a pair of triangles is caused by some PC in the CF, or not. Given a triangle pair $< t_A, t_B >$ in collision, where t_A and t_B are triangles of polyhedra A and B respectively, and suppose the CF is $CF_c = \{PC_i | i = 1, ..., n\}$, our algorithm used the conditions summarized in Table 1 [4] to check if $< t_A, t_B >$ is caused by any $PC_i, i = 1, ..., n$. If every pair of colliding triangles are caused by some PC_i in CF_c, than C_c is feasible. Otherwise, it is not feasible.

3.2 Compliant Interpolation

Recall (as introduced in the beginning of this section) that in the PRM approach to planning high-dimensional collision-free motion, a straight-line path between two collision-free nodes (of a roadmap) is approximated by a sequence of straight-line interpolated configurations.

[4]Note that only non-degenerate types are listed since degenerate PCs have zero probability of occurrence.

PC_i type	condition
$f_A - f_B$	t_A and f_A share an edge and t_B and f_B share an edge.
$e_A - f_B$	t_A and e_A share a vertex and t_B and f_B share an edge.
$e_A - e_B - cross$	e_A is an edge of t_A and e_B is an edge of t_B.
$v_A - f_B$	v_A is a vertex of t_A and t_B is on f_B.

Table 1: $< t_A, t_B >$ is caused by a PC_i if the condition w.r.t the PC_i type holds

In planning CF-compliant motion, however, an exactness requirement has to be always satisfied, that is, a path should be compliant to the CF, but a standard straight-line interpolation between two feasible CF-compliant nodes does not necessarily result in such a CF-compliant path even if it exists. Thus, an important issue is how to make sure that the interpolation is compliant to the configuration manifold of the CF, i.e., compliant interpolation, so that if there is a feasible CF-compliant straight-line path[5], it can be found by interpolation, or else no such path exists. We discovered that by properly setting up the moving object's reference coordinate system, the issue can be resolved. Fig. 3 shows an example, where (a) shows a straight-line interpolation that is not CF-compliant, and (b) shows one that is CF-compliant because of proper selection of the coordinate system.

For a single-PC CF, we can show [6] that by choosing the reference coordinate system of the moving object in the following way, if there exists a feasible CF-compliant straight-line path between two feasible CF-compliant configurations C_1 and C_2, than the path will not be missed by straight-line interpolation:

Origin : for a point PC (i.e., v-f/f-v/e-e-cross), at the contact point; for a line PC (i.e., e-f/f-e) or a plane PC (i.e., f-f), at an arbitrary boundary vertex of the contact edge and face respectively;

Z : along the normal of the contact plane;

X : for a line PC, along the contact edge; for a point PC or a plane PC, along an arbitrary vector on the contact plane;

Y : determined by the right-hand rule from **X** and **Z**.

[5]A "CF-compliant straight-line path" means a straight-line path in the space of CF-compliant configurations, which is of lower dimensions than the general C-space. Such a path is feasible if it consists of only feasible CF-compliant configurations.

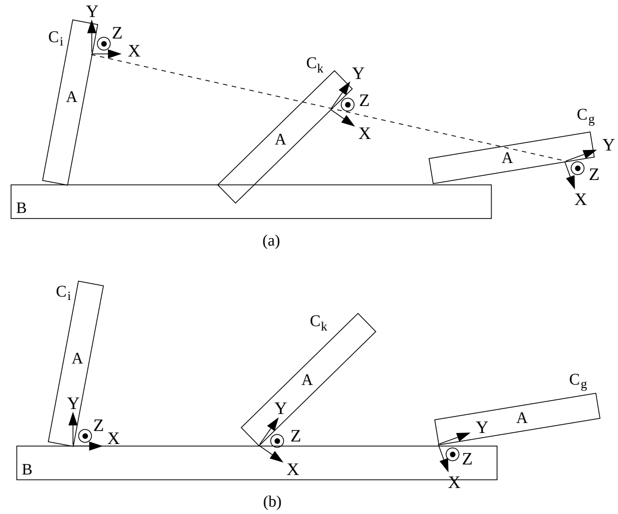

Figure 3: Results of straight-line interpolation depend on the moving object A's frame

In other words, if, using the above reference coordinate system, a configuration generated by straight-line interpolation between C_1 and C_2 is either not CF-compliant or not feasible CF-compliant, then there is surely no feasible CF-compliant straight-line path between C_1 and C_2.

For a two-PC CF $CF_c = \{PC_1, PC_2\}$, our strategy is to achieve compliant interpolation with respect to one PC first, say PC_1, as described above, and then for each interpolated configuration C_{int} which is not PC_2 compliant, use a PC_1-compliant guarded motion to adjust it to be also PC_2 compliant. The PC_1-compliant guarded motion can be either a guarded translation if PC_1 is a f-f PC or a guarded rotation if PC_1 is of other types of PCs. The direction and axis of the guarded translation or rotation can be determined from the contacting elements of the two PCs, and the amount of motion can be efficently calculated based on C_{int} and the coordinates of the contacting elements. The details can be found in [6]. This strategy can be extended to multiple-PC CFs.

4 Some Implementation Results

We have implemented the CF-compliant planner for a CF with one or two PCs. The following examples show feasible CF-compliant paths of moving objects found by the planner under different CFs. Fig. 4, Fig. 5, and Fig. 6 show examples under different single-PC CFs.

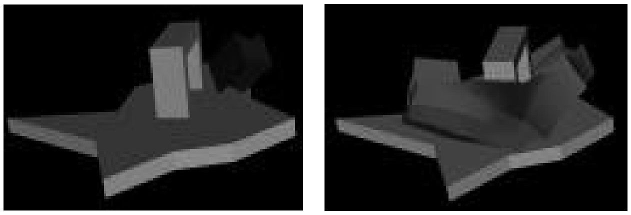

Figure 4: An object's start configuration (left) and motion (right) around the gate while compliant to a {v-f} CF

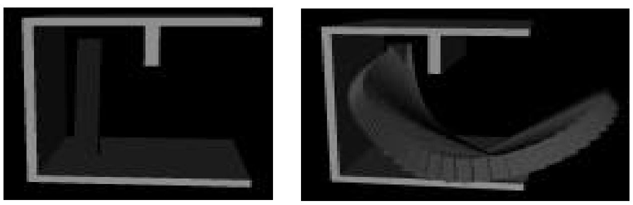

Figure 5: An object's start configuration (left) and motion (right) compliant to an {e-f} CF

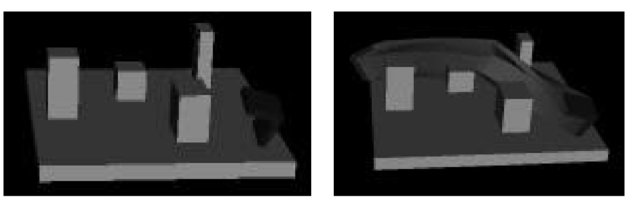

Figure 6: An L-shape object's start configuration (left) and motion (right) compliant to a {v-f} CF

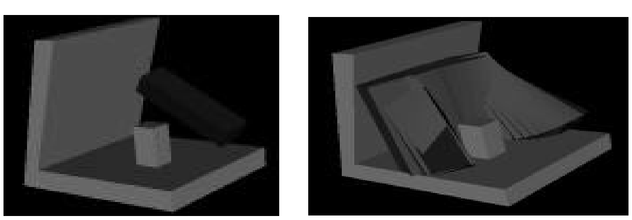

Figure 7: An object's start configuration (left) and motion (right) compliant to a {v-f, v-f} CF

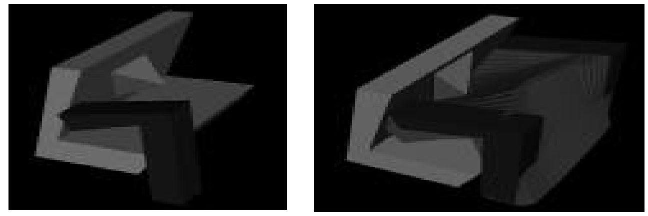

Figure 8: An L-shape object's start configuration (left) and motion (right) compliant to an {e-f, f-e} CF

Fig. 7 and Fig. 8 show examples under different two-PC CFs.

In these examples, random roadmaps consisting of 1,000 to 3,000 configurations are sufficient for finding a path. The running time to construct such a roadmap ranges from 2 to 20 minutes. Once a roadmap is constructed, it can be repeatedly used for fast planning of a path whenever needed, as is characteristic to the PRM approach. The running time to find a path ranges from 1 to 18 seconds. The running times were measured on a SUN Ultra 10 workstation, which is rated at 12.1 SPECint95 and 12.9 SPECfp95.

5 Conclusion

We have introduced an approach to plan motions compliant to a contact state characterized by a CF. We have implemented a CF-compliant planner for CFs between two arbitrary polyhedral objects and consist of one or two PCs. Thus, the degrees of freedom of the CF-compliant motions range from 1D to 5D. The planner combines exploitation of contact constraints and randomized planning to plan motions compliant to contact states with complex configuration manifolds successfully. Such a planner completes the two-level approach to planning general compliant motion [16] (also see Section 1), which is crucial to realizing general compliant motion control [3].

Acknowledgement

We would like to thank Dinesh Manocha for providing codes of RAPID and the triangulation program. We would also like to thank David Johnston for writing the display program.

References

[1] F. Avnaim, J. D. Boissonnat, and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles," *1988 ICRA*, pp. 1656-1660, April 1988.

[2] Randy C. Brost, "Computing Metric and Topological Properties of Configuration- Space Obstacles," *1989 ICRA*, pp. 170-176, May 1989.

[3] J. De Schutter, H. Bruyninckx, S. Dutrè, J. De Geeter, J. Katupitiya, S. Demey, T. Lefebvre, "Estimating First Order Geometric Parameters and Monitoring Contact Transitions During Force Controlled Compliant Motion", *Int. J. Robotics Res.*, 18(12), December 1999.

[4] B. R. Donald, "On Motion Planning with Six Degree of Freedoms: Solving the Intersection Problems in Configuration Space," *1985 ICRA*, pp. 536-541, 1985.

[5] H. Hirukawa, "On Motion Planning of Polyhedra in Contact," *1996 WAFR*, 1996.

[6] X. Ji, "On Generation of Contact State Space and Contact Motion Planning," Ph.D. thesis, University of North Carolina - Charlotte, September 2000.

[7] X. Ji and J. Xiao, "On Random Sampling in Contact Configuration Space," *Workshop on Algorithmic Foundation of Robotics*, March 2000.

[8] X. Ji and J. Xiao, "Towards Random Sampling with Contact Constraints," *2000 ICRA*, San Francisco, April 2000.

[9] L. Joskowicz, R. H. Taylor, "Interference-Free Insertion of a Solid Body Into a Cavity: An Algorithm and a Medical Application," *Int. J. Robotics Res.*, 15(3):211-229, June 1996.

[10] L.E. Kavraki, P. Svestka, J.C. Latombe, and M. Overmars, "Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces," *IEEE Trans. Robotics & Automation*, 12(4):566-580, 1996.

[11] L.E. Kavraki and J.C. Latombe, "Probabilistic Roadmaps for Robot Path Planning," *Practical Motion Planning in Robotics: Current Approaches and Future Directions*, pp. 33-53, 1998.

[12] M. Lin, D. Manocha, and M. Ponamgi, "Fast Algorithms for Penetration and Contact Determination between Non-Convex Polyhedral Models", *IEEE Int. Conf. on Robotics and Automation*, May 1995.

[13] T. Lozano-Pérez, "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Comput.*, C-32(2):108-120, 1983.

[14] R. Seidel, "A Simple and Fast Incremental Randomized Algorithm for Computing Trapezoidal Decompositions and for Triangulating Polygons", *Computational Geometry: Theory and Applications*, 1(1):51-64, 1991.

[15] J. Xiao, "Automatic Determination of Topological Contacts in the Presence of Sensing Uncertainties," *1993 ICRA*, pp. 65-70, May 1993.

[16] J. Xiao and X. Ji, "A Divide-and-Merge Approach to Automatic Generation of Contact States and Contact Motion Planning.", *2000 ICRA*, San Francisco, April 2000.