ON REPLANNING FOR ASSEMBLY TASKS USING ROBOTS IN THE PRESENCE OF UNCERTAINTIES

Jing Xiao
Richard A. Volz

I gratefully and fondly dedicate this article to Dick, who introduced me to robotics and to the important problem of robot assembly in the presence of uncertainty. This article is my second paper on ICRA. The concept of replanning based on contact formations has inspired substantial work on autonomous robot assembly. There are many novel and important assembly problems yet to be tackled by robots. I will have a new collaboration with GM on that front – the research continues.

April 3, 2010

Reprinted from PROCEEDINGS OF THE 1989 IEEE INTERNATIONAL CONFERENCE ON ROBOTICS AND AUTOMATION
Scottsdale, Arizona, May 14-19, 1989

On Replanning for Assembly Tasks Using Robots in the Presence of Uncertainties *

Jing Xiao Richard A. Volz †

Robot Systems Division, CRIM
University of Michigan
Ann Arbor, MI 48109-2110

Abstract

High-precision assembly tasks cannot be successfully done by robots without taking into account the uncertainties that can cause failure of robot motion. We address this problem by planning robot motions at two levels: *Nominal Planning*, which assumes no uncertainty, and *Dynamic Replanning*, to deal with uncertainties that would cause nominal plans to fail. This paper introduces a replanning approach based on knowledge of contacts among assembly parts. It consists of *Patch Planning* to resolve the case when a commanded robot motion prematurely stops at a contact other than those planned, and *Motion Strategy Planning*, to regulate robot motions in order to guarantee the eventual success of a task. A task-independent strategy for patch-plan generation based upon concepts of *contact planes* and *abstract obstacles* is developed. By extending the results of [1], the paper also shows how to apply motion strategies so that, under proper design and motion constraints, the replanning can be guaranteed to succeed. The ultimate goal of this research is to develop a unified, systematic method to enable automatic robot assembly.

1 Introduction

A key issue in automatic generation of robot programs is how to make robots operate reliably in the presence of uncertainties (such as mechanical, control, and sensor errors). For tasks with high-precision requirements, such as assembly tasks, those errors are often significant enough to cause failures of a nominal robot program. Although the problem has attracted considerable attention[4][6] [7][8][5], as we briefly reviewed in [1], none of the previous approaches fully solves the problem. We believe that there is no unconditional solution for the problem, since it is impossible to completely eliminate uncertainties. The uncertainty handling for robot assembly must be a dynamic process involving sensory information and general knowledge of contacts among the parts being assembled, and its success can only be guaranteed if certain constraints on the nominal design parameters, tolerances, and sensor error parameters are enforced. Moreover, we realize that the problem can be simplified by a two-level planning process, *Nominal Planning* (without taking into account the effect of uncertainties) and *Dynamic Replanning* (to handle the effect of uncertainties). We thus introduce an approach which assumes a nominal motion plan for ideal situation, and focuses on replanning utilizing local (and often compliant) motion to correct run-time errors and to resume a nominal plan that has been prematurely terminated.

The approach is based on knowledge of contacts with position/orientation and force/torque sensings. In particular, it uses a concept called *contact formations*[2] rather than the conventional *locations* to describe and classify contacts. The unique characteristics of contact formations which emphasize the topological aspect of contacts, while also carrying spatial and physical meanings, lays a basis for automatic verification of a contact using sensory information. Desai[3] has developed such a verification system to detect contact formations.

The dynamic replanning system mainly consists of two parts:

1. A *patch-planner* generates a nominal patch-plan in real-time to release the held object from the contact which stopped its motion prematurely, and to move the object to a point from which the nominal plan can continue.

2. A *motion strategy planner* determines the actual motion algorithms to be used in the control strategy for both the nominal plan and the nominal patch-plan in the presence of uncertainties.

These two planners are complemented by a *motion-regulator* which determines the actual parameters for a given control strategy and manages its operation. A *plan controller* is used to plug in patch-plans into the system whenever an unexpected contact formation is detected.

In this paper, we extend the concept of contact formations by defining the concept of *contact planes* and *abstract obstacles*, based upon which we present a general patch-planning strategy. By extending our previous work on design and motion constraints[1], we also present general motion strategy planning and show that the constraints derived in [1] also applies to the general case, and that with some additional constraints, the replanning can be guaranteed to succeed in spite of system uncertainties.

2 Error Model and Notation

Assuming position/orientation and force/torque sensors in the system, we established a model of uncertainties in [1], of which, the following are used in this paper:

- position sensing uncertainty ϵ_p,
- orientation sensing uncertainty ϵ_o,
- force sensing uncertainty ϵ_f,
- linear velocity uncertainty ϵ_v.

*This research was partly supported by the Air Force Office of Scientific Research under contract number F33615-85-C-5105.

†The author's current address is: Dept. of Computer Science, Texas A&M University, College Station, Texas 77843-3112.

In this model, each ϵ-type sensing uncertainty is defined as the maximum possible difference in magnitude between a sensed parameter and the actual one. ϵ_o is defined as the maximum possible angular difference in magnitude between a vector in a sensed orientation frame and the corresponding vector in the actual orientation frame. ϵ_v is defined as the maximum difference in magnitude between a commanded velocity and the actual one.

With the error model established, a system parameter or variable may have up to three types of values: *the desired (or derived) value*, *the sensed value*, and *the actual value*. It is useful to introduce some notation to distinguish the different types. Let X denote a variable or parameter. In the rest of the paper, we will use different superscripts of X to indicate its different types of values and to distinguish the values from the symbol:

- X – the symbol or concept of X,
- X^d – the desired (or derived) value of X,
- X^s – the sensed value of X,
- X^a – the actual value of X.

Note that some system parameters/variables may not have all types of values. For instance, a goal position of a peg-in-hole task is a position on the bottom of the hole, which can have a *sensed* value and an *actual* value, but not a *desired* value since the hole exists there and the *desired* is really the *actual*. On the other hand, a *sensed* velocity is meaningless.

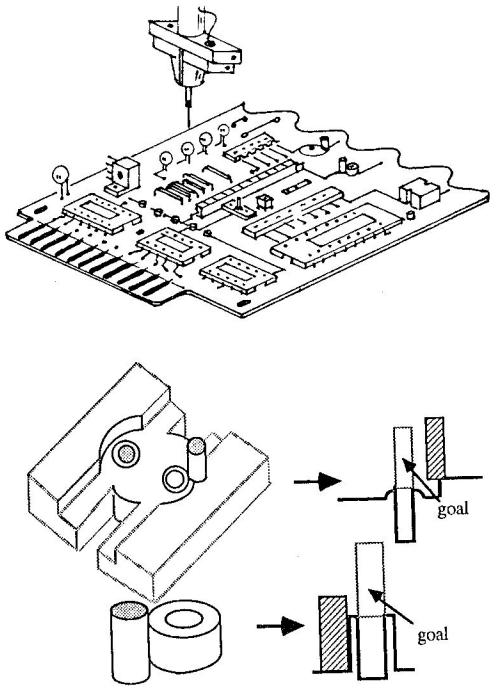

Figure 1: High-precision assemblies and failures

3 Patch-Planning

3.1 Problem Analysis and General Assumptions

Fig. 1 shows a typical environment of high-precision assemblies and some failures that often occur. The problem is that due to mechanical, control, and sensing uncertainties, the part being moved often hits some wrong place of the fixed part and the motion fails. We view fixtures and portions of the fixed part other than the goal area as obstacles; in general, the problem becomes *how to resolve a collision between a moving object and an obstacle (appearing in the form of a surface, an edge, a vertex or combinations thereof) and resume the nominal plan of the object.* The problem is different from that of obstacle avoidance in gross motion planning in both scope and concept; the latter is rather a nominal planning problem dealing with *avoiding obstacles*, while the former must cope with accidental collisions. The problem is also different from a typical navigation problem in the sense that the obstacles are all known beforehand, while only collisions with the obstacles cannot be predicted.

It is often sufficient to view the robot gripper and the object being gripped as one piece in robot part-mating planning, and it is often the case that in high-precision assembly environment,

- the dimensions of the parts are much smaller than the size of the gripper (Fig. 1), and the gripper only holds one end of the moving part, which is the portion not to be assembled with the fixed part;
- the gripper holds the moving part in such a way that the "approach" axis of the gripper frame points to the direction of mating the moving part to the fixed part.

Thus, we can assume a pillar shaped *Gripper-Object Model* (GOM)

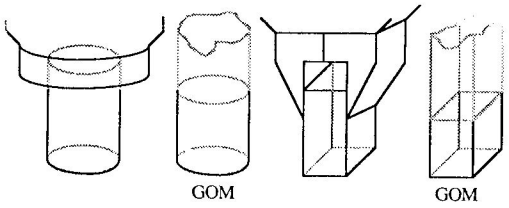

Figure 2: Gripper-Object Model

as shown in Fig. 2, where the effect of gripping is simplified by the portion of the object not being gripped extending to infinity from the gripped end; the axis of the GOM goes through the center of the gripper and along the "approach" axis of the gripper frame. With this model, contacts are restricted to those between an obstacle and *the portion of the object not being gripped.* Due to the open end of the GOM, even fewer contacts between the object and the environment are possible.

For simplicity, we make the following general assumptions:

1. the GOM is solid and convex;
2. obstacles are solid;
3. there is a pre-determined nominal plan for the GOM, which does not take into account the effect of uncertainties;
4. the nominal plan assumes a fixed orientation of the GOM.

By the fourth assumption, the nominal motion plan is translational only. Note that in this paper, the patch-planning will also be focused

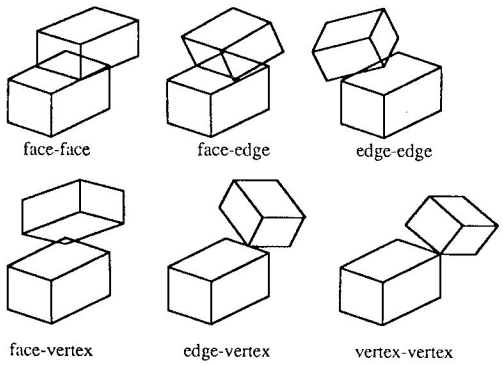

Figure 3: Elemental Contacts

only on generating translational patch-motions. However, we will address the extension of patch-planning to include rotations in Section 5.

In the rest of the section, we first briefly review the concept of *Contact Formations*[2] and the verification of contact formations[2][3], for the general replanning is based on knowledge of contacts; then we introduce a concept of *Contact Planes* and use it to define *Abstract Obstacles*, based upon which we present a patch-planning strategy.

3.2 Contact Formations and Verification

Desai first introduced the topological concept of *a Contact Formation*(CF) [2] as a set of *elemental contacts*(Fig. 3) of the objects involved. An elemental contact is a contact between exactly two topological elements. Obviously, there are only a finite number of possible contact formations between two solid objects of finite surfaces. Since a contact can also be viewed via the spatial relationship between the two objects involved, a contact formation can also be defined *spatially*, as a set of all relative locations (of the objects involved) which make the identical contact consisting of the same set of topological elements. Besides the topological and spatial features of a contact, there are also physical features characterized by the force and moment equilibrium equations. Using *all* the three kinds of features of contact formations to detect contacts can greatly reduce uncertainties. Desai[3] has developed a verification system for contact formations based on this idea, using position/orientation and force/moment sensing.

3.3 Contact Planes and Abstract Obstacles

A contact formation can be viewed as defining a constraint hypersurface in the six dimensional configuration space. However, this can be simplified by a concept called *physical contact planes* under the general assumptions, and the fact that elemental contacts of type vertex-edge, vertex-vertex, and edge-edge with two edges sharing one tangent line at contact point(s) have probability zero of occurring *alone* in reality and can be discarded[1].

Definition 1: A *physical contact plane* (PCP) of an elemental contact is a directed plane tangent to a face involved in face-face, face-edge, or face-vertex types of contacts, or tangent to the two edges in a (crossing) edge-edge contact, with its normal pointing towards the object (Fig. 4).

[1]In the rest of the paper, the term *an elemental contact* will mean one of those *face-face, face-edge, face-vertex, or crossing edge-edge* contacts between a *convex* object and an obstacle, according to which, the meaning of *contact formations* also changes.

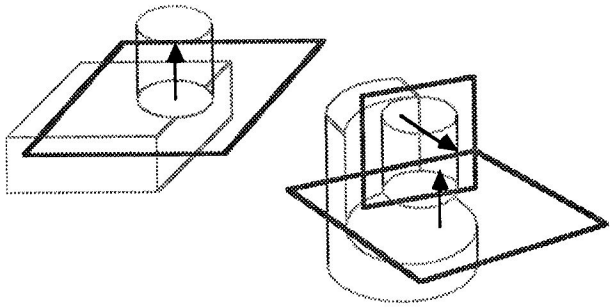

Figure 4: Some physical contact planes

Since we assume *convex* object (or *convex* GOM), it can be proven that 1) the object is wholly on one side of a PCP, and 2) for a contact formation involves more than one PCPs, the PCPs form a concave surface surrounding the object, i.e., the normals of any two PCPs converge at some point (on the side of the object).

The straight forward information that can be sensed about a contact consists of 1) the approximate location of the reference frame of the object, or simply, *the location of the object*, 2) the contact formation (detected by the verifier[3]), which provides an index to the geometric elements (of the obstacle and the object) involved in the contact, and from which, the *normals of the PCPs* can be easily obtained[2]. On the other hand, it is not straight forward to obtain the *positions of the PCPs*, implying that it is not convenient to use PCPs directly in replanning. Therefore, we introduce a concept of *contact planes*, determined completely by the straight forward sensory information.

Definition 2: A *contact plane* (CP) corresponding to a PCP is a directed plane through the position of the object frame (not the obstacle's), with its normal being the normal of the PCP.

Obviously, if a contact formation CF involves more than one CP, then all CPs involved in CF intersect at the position of the object (frame).

We can define *abstract obstacles* in terms of CPs.

Definition 3: An *abstract obstacle* corresponding to a contact formation CF is one of the following, depending on the number of CPs involved in CF:

- a single CP;
- a concave surface formed by two CPs, such that, the intersection line of the two CPs goes through the position of the object, and there exist normals of the two CPs that converge to a point;
- a polyhedral cone of CPs, such that, its vertex is the position of the object, and the convergence points of the normals of any two CPs are inside the cone.

When a CF involves two parallel CPs (with normals pointing to each other), the corresponding abstract obstacle can still be viewed as a concave surface with an infinitesimal distance assumed between the two CPs. Fig. 5 shows some abstract obstacles.

Definition 4: A *boundary point* **P** *of an abstract obstacle* AO is a point on a contact plane C of AO, at which the object is still in contact, and any open neighborhood of **P** on C includes points at which either the object is free, or it encounters a new contact plane other than those of AO. The *boundary of an abstract obstacle* AO is the set of all boundary points of AO.

[2]As discussed in Section 3.1, we assume *known and fixed* obstacles such that each obstacle has a pre-stored database of geometric, spatial, and physical informat

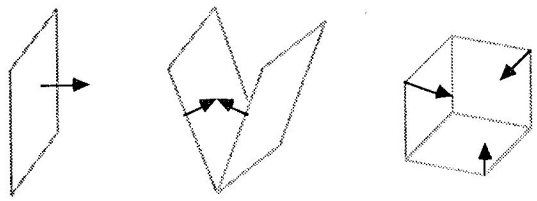

Figure 5: Abstract obstacles

3.4 Straight-Line Nominal Path and Problem Formation

We can generally view a nominal path as consisting of straight-line (positional) path segments, with the first segment starting from the initial position (of the object), the intersection points between segments being the intermediate goal positions, and the last segment ending at one of the ultimate goal positions; however, it is often sufficient to only focus replanning on the last straight-line segment. Hence, we define the *nominal path* of our concern as the following.

Definition 5: A *straight-line nominal path* PA_n is parallel to the axis of the GOM and pointing to (or ending at) a goal position $\mathbf{P}_{ng}$ of the GOM, such that, by translating along PA_n with the finite end of GOM pointing to $\mathbf{P}_{ng}$, the GOM will reach $\mathbf{P}_{ng}$ in ideal case (i.e., with no presence of errors). The starting point of PA_n is denoted as $\mathbf{P}_{ns}$, such that, the distance vector $\mathbf{d}_n$ of PA_n is $\mathbf{d}_n = \mathbf{P}_{ng} - \mathbf{P}_{ns}$.

Clearly, only a sensed axis of the GOM and a pre-sensed goal $\mathbf{P}^s_{ng}$ can be obtained, which determine the sensed PA^s_n. The distance vector $\mathbf{d}^s_n$ is then: $\mathbf{d}^s_n = \mathbf{P}^s_{ng} - \mathbf{P}^d_{ns}$, where $\mathbf{P}^d_{ns}$ is the desired starting position of PA^s_n. Initially, the GOM should be moved to $\mathbf{P}^d_{ns}$, but due to uncertainties, the GOM may stop at some point other than $\mathbf{P}^d_{ns}$ which we denote as $\mathbf{P}^a_{ns}$ with sensed value $\mathbf{P}^s_{ns}$. *The replanning we consider starts from that point.* A *nominal motion of the GOM* can be defined as: the straight-line motion of GOM towards $\mathbf{P}^s_{ng}$. The *patch-planning*, then, is to find a nominal patch-path [3] for the GOM to resolve the contact when it accidentally hits an obstacle due to uncertainties, and to move towards $\mathbf{P}^s_{ng}$ again, or at least, move back to PA^s_n. The *patch motion of the GOM*, thus, can be defined as the motion of the GOM along a patch-path.

3.5 Patch-Plan Generation

A patch-plan is needed when the motion of GOM is terminated because the GOM is in contact with some obstacle. Let AO be the abstract obstacle of the contact formation which the GOM is involved in; let $\mathbf{P}_c$ be the position of the GOM. Let $\mathbf{d}^s_{cn} = \mathbf{P}^s_{ng} - \mathbf{P}^s_c$ denote the sensed straight-line path for the GOM to move to $\mathbf{P}^s_{ng}$. the patch-planner first determines what type of patch-plan is needed: if it can be sure that the actual AO^a does not block $\mathbf{d}^s_{cn}$, the patch-plan is simply to command the GOM to move along $\mathbf{d}^s_{cn}$ towards $\mathbf{P}^s_{ng}$; otherwise, the patch-plan should consist of two parts, one to free the GOM from the blocking of AO^a, and one to lead the GOM back to the nominal path. In order to reduce the dimensionality and the associated uncertainties of the robot motion, the first part of the patch-motion is prefered to be guided by the contact planes of AO, i.e., compliantly[9][10] sliding the object being moved along physical contact planes. Therefore, a patch-path may have compliant path segment preceding a guarded path segment.

[3] We call a patch-path *nominal* (or *sensed*) because the patch-path can only be generated based upon nominal design and sensed information.

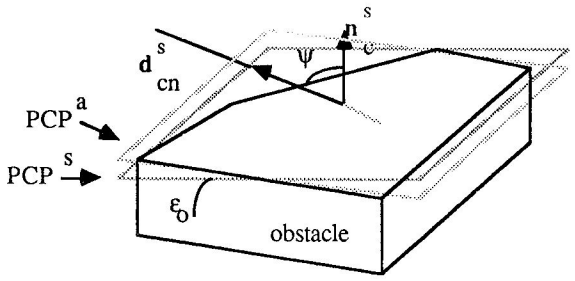

Figure 6: $\mathbf{d}^s_{cn}$ does not point through PCP^a

3.5.1 Pre-condition for Compliant Path

Let C be a contact plane of AO with its sensed normal denoted as $\mathbf{n}^s_C$. Let ψ be the angle between $\mathbf{n}^s_C$ and $\mathbf{d}^s_{cn}$:

$$\psi = \arccos \frac{(\mathbf{n}^s_C \cdot \mathbf{d}^s_{cn})}{\|\mathbf{d}^s_{cn}\|}$$

$(0 \leq \psi \leq \pi)$. Taking into account the orientation sensing uncertainty ϵ_o in $\mathbf{n}^s_C$, if

$$\psi + \epsilon_o < \frac{\pi}{2}, \tag{1}$$

it can be certain that $\mathbf{d}^s_{cn}$ does not *point through* the actual physical-contact plane PCP^a of C even in the worst case (Fig. 6). If for *all* contact planes of AO, equation (1) is satisfied, then the patch-path can simply be $\mathbf{d}^s_{cn}$; otherwise, a compliant path segment is needed to resolve the blocking of AO.

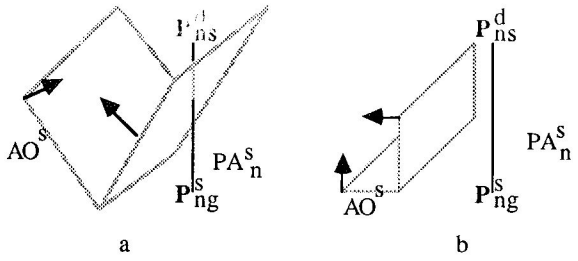

Figure 7: AO^s and PA^s_n

3.5.2 Compliant Path Generation

A compliant path can be generated based on the relationship between AO^s and PA^s_n. Let L^s_n be the line PA^s_n lies on. If 1) $\exists C^s$ of AO^s intersecting with L^s_n, and 2) the intersection point is on AO^s (Fig. 7a), then we say: AO^s *intersects with* PA^s_n; otherwise, AO^s *does not intersect with* PA^s_n(Fig. 7b). By the definitions of GOM and PA_n, it is easy to observe that

1. the abstract obstacles that GOM can possibly encounter are those, such that, all contact planes of which satisfy: $\mathbf{n}^s_C \cdot \mathbf{d}^s_n \leq 0$, where $\mathbf{n}^s_C$ is the sensed normal of contact plane C; thus, if an AO^s intersects with PA^s_n, there must be a unique intersection point (Fig. 7a), and if an AO^s does not intersect with PA^s_n, at least one of its contact planes C^s is parallel to PA^s_n (Fig. 7b);

2. in concept, the opposite direction $-\mathbf{d}_n$ of PA_n points to an *open region* free of obstacles, i.e., translating the GOM along $-\mathbf{d}_n$ (which points to the infinite end of GOM) is always possible (within certain scope); thus in practice, commanding the

GOM to translate along $-\mathbf{d}_n^s$ carries the least risk of collision.

Based on the above observations, the following routine generates a compliant path, which has: a goal $\mathbf{P}_{int}^d$ (if AO^s intersects with PA_n^s), a direction $\mathbf{d}_c^s$, and a set of constraint contact planes S_C (containing one or two contact planes).

1. Input AO^s as a set of contact plane normals $\{\mathbf{n}_{C_1}^s, \mathbf{n}_{C_2}^s, ..., \mathbf{n}_{C_i}^s, ..., \mathbf{n}_{C_n}^s\}$;
2. Input PA_n^s as $\{\mathbf{P}_{ns}^d, \mathbf{P}_{ng}^s\}$;
3. Input $\mathbf{P}_c^s$;
4. $\mathbf{d}_n^s \Leftarrow \mathbf{P}_{ng}^s - \mathbf{P}_{ns}^d$;
5. If AO^s intersects with PA_n^s at $\mathbf{P}_{int}^d$ of contact plane(s) C_i^s (...), goto 6, else goto 8;
6. $\mathbf{d}_c^s \Leftarrow \mathbf{P}_{int}^d - \mathbf{P}_c^s$, $S_C \Leftarrow \{\mathbf{n}_{C_i}^s(...)\}$;
7. Output $\mathbf{P}_{int}^d$, goto 9;
8. $\mathbf{d}_c^s \Leftarrow (-\mathbf{d}_n^s)$, Find the set S_C of the contact planes which are parallel to PA_n^s;
9. Output $\mathbf{d}_c^s$, and S_C, end.

Note that $\mathbf{P}_{int}^d$ can be easily derived. If contact plane C_i^s of AO^s is not parallel to PA_n^s, by solving plane and line equations, $\mathbf{P}_{int_i}^d$ can be determined. If the vector $\mathbf{P}_{int_i}^d - \mathbf{P}_c^s$ does not point through any other contact planes of AO^s, $\mathbf{P}_{int}^d \leftarrow \mathbf{P}_{int_i}^d$.

By the above routine, a compliant path is generated for the GOM to move closer to the nominal path *compliantly* if possible (i.e., AO^s intersects with PA_n^s), or, to move along the least risky direction (i.e., $-\mathbf{d}_n^s$) *compliantly* under some motion strategy and motion constraints (see Section 4), until the contact is broken or new contact occurs (sensed by some force/moment guards). In the latter case, the above strategy may again be used, depending on whether the new abstract obstacle (of the new contact formation detected) blocks the new straight-line path towards the goal $\mathbf{P}_{ng}^s$. In the former case, however, a guarded motion leading the object back to the nominal path should be generated (Section 3.5.3).

It should be noticed that in the definition of AO^s *intersects with* PA_n^s (at the beginning of this subsection), the boundary of AO was not taken into account. In fact, if the boundary of AO is considered, a bounded AO^s should *not* intersect with PA_n^s (i.e., PA_n^s should not penetrate through AO^s *within* its boundary), for, otherwise, compliant *translations* cannot release the GOM from the block of AO by moving it to PA_n^s, and compliant rotations should be applied, which are beyond the focus of this paper. The issue, however, will be addressed as future research in Section 5. As for the above routine, a non-zero $\mathbf{d}_c^s$ is always returned as long as PA_n^s does not penetrate through AO^s *within* its boundary.

3.5.3 Guarded Path Generation

When the object is released from a contact at position $\mathbf{P}_f$ by compliant motions, as sensed and stopped by some force/moment guards, very likely, it still has not reached PA_n^s. Thus, an additional patch-path for guarded motions to make the object back to PA_n^s is required. The following routine generates a straight-line guarded patch-path PA_g^s, which either points to the goal $\mathbf{P}_{ng}^s$ or an intermediate goal $\mathbf{P}_{mg}^d$ on PA_n^s:

1. Input PA_n^s as $\{\mathbf{P}_{ns}^d, \mathbf{P}_{ng}^s\}$;
2. Input $\mathbf{P}_f^s$;
3. $\mathbf{d}_n^s \Leftarrow \mathbf{P}_{ng}^s - \mathbf{P}_{ns}^d$, $\mathbf{d}_{fn}^s \Leftarrow \mathbf{P}_{ng}^s - \mathbf{P}_f^s$;
4. If $\angle(\mathbf{d}_{fn}^s, \mathbf{d}_n^s) < \theta_v + \epsilon_o$, obtain $\mathbf{P}_{mg}^d$ on PA_n^s, based on $\angle[(\mathbf{P}_{mg}^d - \mathbf{P}_f^s), \mathbf{d}_n^s] = \theta_v + \epsilon_o$, else goto 6;
5. $PA_g^s \Leftarrow \{\mathbf{P}_f^s, \mathbf{P}_{mg}^d\}$, goto 7;
6. $PA_g^s \Leftarrow \{\mathbf{P}_f^s, \mathbf{P}_{ng}^s\}$;
7. Output PA_g^s, end.

Note that θ_v is defined by equation (3) in Section 4.1, which describes the directional uncertainty of velocity. By this routine, if $\angle(\mathbf{d}_{fn}^s, \mathbf{d}_n^s) < \theta_v + \epsilon_o$, the GOM is commanded to move to $\mathbf{P}_{mg}^d$, not directly to $\mathbf{P}_{ng}^s$, in order to avoid the GOM to be trapped into the old contact again because of uncertainties. Fig. 8 shows a worst case and how moving the GOM to $\mathbf{P}_{mg}^d$ will avoid the GOM vertically go back to the old contact. Once the intermediate goal $\mathbf{P}_{mg}^d$ is generated, the part of PA_n^s from $\mathbf{P}_{mg}^d$ to $\mathbf{P}_{ng}^s$ can be viewed as a new nominal path segment PA_{n1}^s, along which the replanning can be applied again.

3.5.4 Summary

In this section, we presented the automatic process of generating patch-plans, which starts when the GOM collides with an obstacle. Since the patch-plan generation is guided by the *current* contact formation (or abstract obstacle) and other *local* sensory information, the strategy only guarantees to generate a patch-plan which can resolve the *current* contact and lead the object towards the nominal path; it can not guarantee that the patch-path generated is potentially clear of obstacles. Moreover, even if the patch-path generated is a clear path,

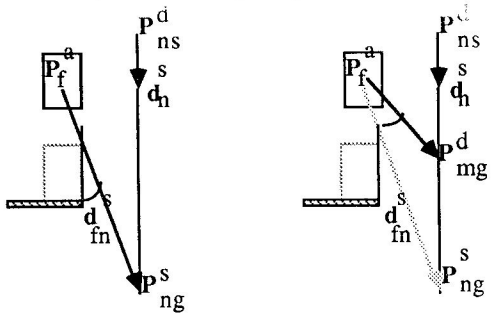

Figure 8: Worst case study

the object may still collide with some obstacle while it deviates from the path due to robot motion uncertainties. Thus, it is always possible for the object to undergo consecutive patch-plannings before it can go back to the nominal path. Then, of course, the object may hit something again while *trying* to follow the nominal path, and again the whole patch-planning process should be activated.

4 Constraints and Motion Strategy Planning

In order to guarantee the eventual success of a task, it is necessary to study how to actually execute the nominal plan or a nominal patch plan in the presence of sensor and robot motion errors, so that the object will be able to reach its goal and never enter an indefinite loop of repeating the same contacts. Thus, proper motion strategies under some motion constraints must be developed, along with the derivation of proper design constraints on the nominal and error parameters of

the system. The necessity of imposing design constraints lies in the fact that uncertainties can never be eliminated entirely.

There are two basic motion strategies. A *single-motion* strategy is to move the object towards some goal in one motion step. Ideally, when the motion is terminated, the object should be at the desired goal. Practically, this may never occur due to uncertainties, and the object can be significantly far away from the desired goal. A *multiple-motion* strategy uses feedback information to reduce the sensing and motion uncertainties so that the object can be close enough to its goal. Namely, this strategy is to move the object in multiple motion steps in such a way that after each motion step, the object is always closer to its goal. The termination condition of each motion step can be determined by some motion constraints on the desired distance of the motion step. The termination condition of the whole process can be determined by testing whether the object has reached some sensing uncertain region or has met some other sensing requirements (such as the force/moment guards).

Based on the nature of the paths given, both motion strategies are used in replanning. The single-motion strategy is used either when a given compliant path is constrained by two contact planes, or a given compliant path does not have an explicit goal, which occurs when the AO^s the GOM is involved with does not intersect with PA^s_n (see Section 3.5.2). The multiple-motion strategy is applied otherwise. In our previous paper [1], we applied the multiple-motion strategies to peg-in-hole tasks, and developed design and motion constraints (on the nominal and error parameters of the system) in order to guarantee the success of the tasks. In the following subsections, we will show that the constraints derived in [1] can be applied to the more general case here, under which and with some additional constraints, the replanning can be successful.

4.1 Guarded Motions to $\mathbf{P}_{ng}$

As defined in Section 3.4, guarded nominal motions are straight-line motions directly heading to the goal $\mathbf{P}^s_{ng}$. However, some guarded patch-motions also directly head to the goal $\mathbf{P}^s_{ng}$, as in the cases 1) when the AO in which the GOM is involved does not block $\mathbf{d}^s_{cn}$ (Section 3.5.1), and 2) when $\angle(\mathbf{d}^s_{fn}, \mathbf{d}^s_n) \geq \theta_v + \epsilon_o$ (Section 3.5.3), and thus can be treated in the same way as the nominal motions.

Without losing generality, let $\mathbf{P}_i$ be the position of the GOM, and $\mathbf{d} = \mathbf{P}_{ng} - \mathbf{P}_i$ be the distance vector from the GOM to the goal with sensed value $\mathbf{d}^s$ and actual value $\mathbf{d}^a$. The multiple-motion strategy is to command the GOM to move along $\mathbf{d}^s$ for certain distance l, such that when the motion stops, the new d^a is smaller than the previous one; then the GOM is commanded to move along the new $\mathbf{d}^s$ for some new distance l, and when the motion stops, d^a is even smaller, and so on. the whole process stops either prematurely when the GOM hits some obstacle, or when some final termination condition is satisfied. The following discusses and presents the termination condition for each motion step (i.e., the motion constraint on l) and the final termination condition, along with the derivation of design constraints.

Let v^d be the desired motion speed, and suppose that only position sensing is used to guide the robot motion. Then, as discussed in [1], the actual direction of motion can deviate from the ideal direction of motion $\mathbf{d}^a$ by $\theta + \theta_v$ in the worst case (Fig. 9), where

$$\sin(\theta) = \frac{2\epsilon_p}{d^s}, \tag{2}$$

and

$$\sin(\theta_v) = \frac{\epsilon_v}{v^d}. \tag{3}$$

In order for the object to be closer to $\mathbf{P}^a_{ng}$ after the motion, as derived

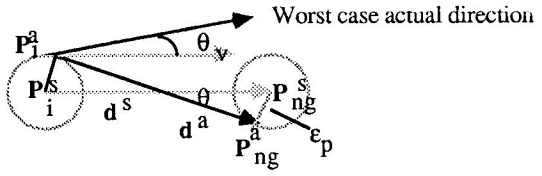

Figure 9: The worst case actual direction

in [1], the design constraint,

$$v^d > 2\epsilon_v,$$

which is equivalent to

$$\theta_v < \frac{\pi}{6}, \tag{4}$$

and the design constraint

$$0 \leq \theta_v + \theta_p < \frac{\pi}{2}, \tag{5}$$

must be satisfied, where θ_p denotes the maximum value that θ is allowed to have, i.e., $\theta \leq \theta_p$, and the motion constraint

$$\frac{\epsilon_v}{v^d} l_{max} < l < (1 - \frac{\epsilon_v}{v^d}) l_{max} \tag{6}$$

must be imposed, where

$$l_{max} = (d^s - 2\epsilon_p)\cos(\arcsin(\frac{2\epsilon_p}{d^s}) + \theta_v), \tag{7}$$

and

$$d^s \geq \frac{2\epsilon_p}{\sin(\theta_p)}. \tag{8}$$

Note that unlike in [1], the effect of parameters v_r and ω_t is ignored in equation (7), since we assume fixed orientation of the GOM. Further, the l_{max} in this paper is half the value we set in [1]. This is to enable the GOM to move closer to the goal *all the time*.

If position sensing is used to command the motion of the object, the violation of inequality (8), that is,

$$d^s < \frac{2\epsilon_p}{\sin(\theta_p)}, \tag{9}$$

is the final termination condition of the multiple-motion process. The reason is that when (9) holds, or $\theta > \theta_p$, d^a is bounded by $0 \leq d^a \leq d^s + 2\epsilon_p < d_p$, where

$$d_p = \frac{2\epsilon_p}{\sin(\theta_p)} + 2\epsilon_p, \tag{10}$$

and the actual direction of motion may deviate so much from the commanded direction of motion, that no matter what value l is chosen, the motion may not lead the GOM closer to $\mathbf{P}^a_{ng}$. d_p can be denoted as *the minimum distance allowed by position sensing between the actual position of the object and the actual goal position.* We denote the region inside the sphere about $\mathbf{P}^a_{ng}$ with radius d_p as *the position sensing uncertain region.* Obviously, given the position sensing uncertainty ϵ_p, the value of d_p can be determined by a given θ_p. θ_p, however, should be chosen based on the design constraints (4) and (5).

In order for the mating task to be successful under position sensing, design constraints relating d_p and some parameters reflecting the geometric tolerance and clearance of the task should be enforced as discussed in [1]. In this more general paper, we are only concerned with moving the object until (9) is satisfied, i.e., moving the object only as close to its goal as position sensing can guide it.

4.2 Guarded Motions to $\mathbf{P}_{mg}$

A guarded patch-path segment may point to an intermediate goal $\mathbf{P}^d_{mg}$, and the patch-motions should head to $\mathbf{P}^d_{mg}$ instead of $\mathbf{P}^s_{ng}$. Moving to $\mathbf{P}^d_{mg}$ is different from moving to $\mathbf{P}^s_{ng}$, since for the former, $\mathbf{P}^d_{mg}$ is the desired destination, while for the latter, the desired destination is not $\mathbf{P}^s_{ng}$ but $\mathbf{P}^a_{ng}$. As the result, the termination conditions for motions towards $\mathbf{P}^d_{mg}$ are *less* restricting. When the multiple-motion strategy is applied to motions towards $\mathbf{P}^d_{mg}$, with design constraints derived in Section 4.1 imposed, it can be shown that, although the motion constraint (or the termination condition for each motion step) can still be expressed by (6), l_{max} should be changed to the following:

$$l_{max} = (d^s - \epsilon_p)\cos(\arcsin(\frac{\epsilon_p}{d^s}) + \theta_v), \tag{11}$$

where d^s is the (sensed) distance from the GOM to $\mathbf{P}^d_{mg}$, which satisfies

$$d^s \geq \frac{\epsilon_p}{\sin(\theta_p)}.$$

Accordingly, the termination condition for the whole process becomes

$$d^s < \frac{\epsilon_p}{\sin(\theta_p)}. \tag{12}$$

d_p, the minimum distance allowed by position sensing, becomes

$$d_p = \frac{\epsilon_p}{\sin(\theta_p)} + \epsilon_p, \tag{13}$$

which is the radius of the position sensing uncertain region about $\mathbf{P}^d_{mg}$.

Note that the multiple-motion strategy should also be used when the GOM is initially moved to $\mathbf{P}^d_{ns}$ (the desired starting position on PA^s_n). Clearly, $\mathbf{P}^d_{ns}$ can be viewed as the *first* intermediate goal $\mathbf{P}^d_{mg}$. Thus, the actual starting position $\mathbf{P}^a_{ns}$ of the GOM is within the position sensing uncertain area about $\mathbf{P}^d_{ns}$, which radius d_p expressed by (13).

4.3 Compliant Motions

As introduced at the beginning of Section 4, the single-motion strategy is used if a given compliant path is constrained by two contact planes, or a given compliant path does not provide an explicit goal. Otherwise, the multiple-motion strategy should be applied, with the goal set to $\mathbf{P}^d_{int}$ (i.e., the intersection point between AO^s and PA^s_n). Clearly, if $\mathbf{P}^d_{int} = \mathbf{P}^s_{ng}$, the termination conditions presented in Section 4.1 should be applied; otherwise, the termination conditions presented in Section 4.2 should be applied, since $\mathbf{P}^d_{int}$ can be virtually viewed as another $\mathbf{P}^d_{mg}$. Note that the compliant motion process is expected to be stopped by force/moment guards *before* the GOM reaches $\mathbf{P}^d_{int}$ (as explained in Section 3.5.2). Moveover, additional design and motion constraints need to be imposed to ensure *compliance*.

Let $\mathbf{n}^s_C$ be the sensed normal of a contact plane. let λ_f be the angle between $\mathbf{n}^s_C$ and the direction of the sensed force $\mathbf{F}^s$ imposed on the object such that

$$\lambda_f = \arctan(|\frac{F^s_{\|}}{F^s_{\perp}}|),$$

where $F^s_{\|}$ indicates the force component parallel to the sensed contact plane and $F^s_{\perp}$ indicates the force component perpendicular to the sensed contact plane (Fig. 10). Let μ be the friction coefficient of the materials in contact. Using the friction cone model[7] (Fig. 11), and taking into account the sensing uncertainty ϵ_o in $\mathbf{n}_C$, and ϵ_f in

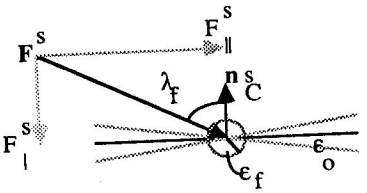

Figure 10: The force for compliant motions

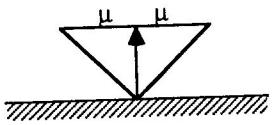

Figure 11: The friction cone

$\mathbf{F}^s$, we see that if

$$\lambda_f - \epsilon_o - arcsin(\frac{\epsilon_f}{\|\mathbf{F}^s\|}) > \arctan(\mu), \tag{14}$$

sticking will not occur, and if

$$\lambda_f + \epsilon_o + arcsin(\frac{\epsilon_f}{\|\mathbf{F}^s\|}) < \frac{\pi}{2}, \tag{15}$$

the compliance can be maintained. Define θ_f to be the following:

$$\theta_f = arcsin(\frac{\epsilon_f}{F^s_{min}}), \tag{16}$$

where $F^s_{min}(> \epsilon_f)$ is the minimum sensed magnitude of the force. Then, combining (14) and (15) with (16), the following motion constraint

$$\arctan(\mu) + \epsilon_o + \theta_f < \lambda_f < \frac{\pi}{2} - \epsilon_o - \theta_f \tag{17}$$

must be enforced to *maintain* compliant motion with the design constraint:

$$\arctan(\mu) + 2\epsilon_o + 2\theta_f < \frac{\pi}{2} \tag{18}$$

on μ, ϵ_o, and θ_f satisfied. With the generalized damper model[11], a compliant motion can be implemented under the motion constraint (17).

4.4 Global Success of the Replanning

In previous sections, we described how to generate patch-paths, and how to apply proper motion strategies with design and motion constraints on the motions of the GOM, so that it always moves closer to its goal. Since the discussions were given from only a *local* view, we still, however, face the global question of whether the GOM will be trapped into an indefinite loop of repeating some old contacts and never reach $\mathbf{P}_{ng}$. Clearly, if there were no obstacles, the GOM can be guaranteed to reach the position sensing uncertain region U_p of $\mathbf{P}^a_{ng}$, if only position sensing is used to guide motions[4]. Since there are obstacles to stop a planned motion process prematurely and sometimes force the GOM to move further away from $\mathbf{P}^a_{ng}$ to (compliantly) get rid of the blocking, the global success of the replanning can be guaranteed only when more constraints are enforced.

For the convenience of discussion, we set a (x, y, z) Cartesian coordinates system at the sensed goal $\mathbf{P}^s_{ng}$ with the z axis pointing to $\mathbf{P}^d_{ns}$, and a (ρ, ϕ) polar coordinates system at $\mathbf{P}^s_{ng}$ on the xy plane (Fig. 12). Since we assume that the real world obstacles are fixed (Section 3.1), we can view them as a single obstacle described by a connected region Obs in the space. Let Obs_{xy} denote the projection of the Obs on xy plane. Let $Z_{max}(x, y)$ be a function defined on

[4] In[1], we also discussed how to use force/moment sensing to compensate position sensing.

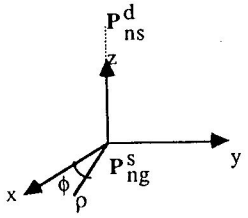

Figure 12: Coordinates systems

Obs_{xy}, such that, $Z_{max}(x,y)$ is the z component of the "highest" point $P(x,y,z_{max}) \in Obs$. It can be proven that the replanning (i.e., patch-planning and motion strategy planning) can always lead the GOM to the U_p of $\mathbf{P}_{ng}$ under the design and motion constraints derived, if Obs (which takes into account the modeling and sensing errors) satisfies *one* of the following geometric and spatial constraints:

1. Obs is a solid of revolution about z axis;
2. $Z_{max}(x,y)$ is a non-decreasing function of ρ along line $\phi = \alpha$, where $\tan(\alpha) = y/x$;
3. Obs can be partitioned into such regions that each region satisfies either one of the above two conditions.

It is easy to see that the obstacles in many typical problems, such as those shown in Fig. 1 satisfy the condition within the working space. Thus, the success of the replanning can be guaranteed for those cases.

5 Future Research and Conclusions

The entire replanning strategy presented is independent of specific tasks, and its success can be guaranteed under certain constraints.

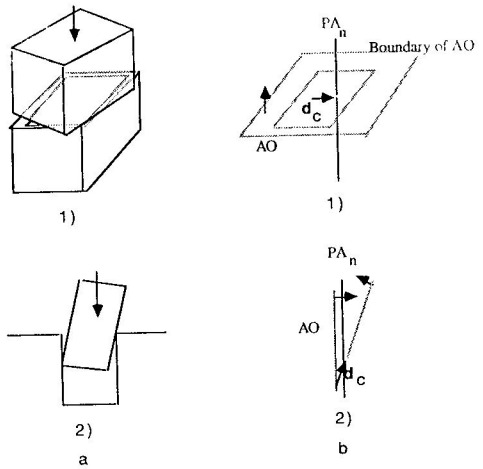

Figure 13: Failures that need rotational replanning

However, the current strategy is limited to translational motions only and incapable of handling failures mainly due to orientational errors (Fig. 13a). Rotational replanning are needed for those cases. Interestingly, it can be generally shown that the abstract obstacle involved in such a failure always intersects with the nominal path PA_n *within its boundary* (Fig 13b), which is a property that might be used for the patch-planner to decide if a rotational patch-path is needed. Thus, we can possibly extend the replanning strategy to including rotations in the next step. We also plan to investigate special applications involving non-convex gripper-object models in the future.

References

[1] J. Xiao, and R. Volz, 1988(April). Design and motion constraints of part-mating planning in the presence of uncertainties. *Proc. IEEE Int. Conf. Robotics and Automation.*

[2] R. Volz, J. Xiao, and R. Desai, 1988(July). Contact formations and design constraints: A new basis for the automatic generation of robot programs, *NATO ARW: CAD Based Programming for Sensor Based Robots.*

[3] R. Desai, 1987(December). On fine motion in mechanical assembly in presence of uncertainty. *Ph.D. Dissertation*, Department of Mechanical Engineering, the University of Michigan.

[4] R. A. Brooks, 1982. Symbolic error analysis and robot planning. *Int. J. Robotics Res.* 1(4):29.

[5] B. R. Donald, 1988. Planning multi-step error detection and recovery strategies. *Proc. IEEE Int. Conf. Robotics and Automation.*

[6] B. Dufay, and J. C. Latombe, 1983(Aug.). An approach to automatic robot programming based on inductive learning. *1st International Symposium on Robotics Research.*

[7] M. Erdmann, 1986. Using backprojections for fine motion planning with uncertainty. *Int. J. Robotics Res.*

[8] T. Lozano-Pérez, M. T. Mason, and R. H. Taylor, 1984. Automatic synthesis of fine-motion strategies for robot. *Int. J. Robotics Res.* 3(1):3.

[9] M. T. Mason, 1982. Compliant motion. *Robot Motion: Planning and Control*, M.Brady et al. Eds.

[10] M. H. Raibert, and J. J. Graig, 1981. Hybrid position/force control of manipulators. *Robot Motion: Planning and Control*, M.Brady et al. Eds.

[11] D. E. Whitney, 1977(June). Force feedback control of manipulator fine motions. *J. Dynamic Sys., Measurement, and Control.* 98:91.