RSD-18-87

Design and Motion Constraints of Part-Mating Planning in the Presence of Uncertainties

Jing Xiao
Richard A. Volz

Robot Systems Division
Center for Research on Integrated Manufacturing
The University of Michigan
Ann Arbor, Michigan 48109

September 1987

Center for Research on Integrated Manufacturing

Robot Systems Division
College of Engineering
The University of Michigan
Ann Arbor, Michigan 48109-2110

TABLE OF CONTENTS

Design and Motion Constraints of Part-Mating Planning in the Presence of Uncertainties

Jing Xiao
Richard A. Volz
Robot Systems Division*
Center for Research on Integrated Manufacturing
University of Michigan

September 10, 1987

Abstract

In order to achieve the ultimate goal of automatically generating assembly programs for robots from design information, it is necessary that one be able to devise part mating strategies that will work in spite of sensor, control and manufacturing errors. In general, this is almost certainly unachievable. However, if appropriate design constraints are made, progress can be made. In this paper, we assume a nominal motion plan for the zero error situation, and then devise a simple replanning strategy based upon the availability of force, moment and position sensors to handle errors that may arise during program execution. We develop design constraints relating the parameters of the strategy, parameters of the sensor, control and manufacturing errors, and nominal design parameters. If the constraints are satisfied, the replanning strategy can theoretically be guaranteed to be successful. The constraints are shown to be reasonable in the sense that they do not impose unrealistic conditions on typical designs. Simulation results uphold the theoretical derivations and show empirically that the theoretical constraints can be relaxed somewhat with excellent results still obtained.

1 Introduction

The manual development of robot programs for assembly is a laborious, error prone task. It would be highly desireable to be able to derive them automatically from design information. In order to do so, several important problems must be solved: 1)

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

automatic determination of an assembly plan (sequence of assembly operations), 2) automatic determination of parts presentation and fixturing, 3) gross motion planning, 4) fine motion planning, and 5) automatic generation of error recovery routines. This paper addresses the last of these problems.

It has often been stated that the largest part of any robot assembly program is made up of fixups to handle things that don't quite work as planned. Algorithms for generating these fixups are crucial to successful automatic program generation. Problems in the execution of nominal robot programs arise for the following reasons:

- *mechanical* and *control errors*: robots and all other mechanical devices are only accurate to within certain bounds;
- *sensor errors*: all sensors have limited sensitivity and accuracy;
- *manufacturing tolerances*: different instances of the same model are not exactly identical because of manufacturing limitations.

Often such errors can lead to the failure of a nominal program that would theoretically accomplish a task such as part mating. The automatic generation of robot programs for assembly cannot be successful until some means of either avoiding or correcting these errors is found.

Several approaches towards handling these uncertainties have appeared in the literature. Taylor[11] introduced an error-propagation method to estimate compound errors from the uncertainty bound or tolerance of each individual part involved in a task (including the robot hand). Brooks[1] extended Taylor's method by making the process backward, so that the constraints on some compound errors to make the task succeed can lead to constraints on individual parts. This method, however, suffers from high computational complexity. And it does not provide means to reduce errors dynamically. The inductive learning approach by Dufay and Latombe[3] corrects run-time errors by adding rules into the system as a corrective plan. In this approach, error-handling is not fully automatic, since rules must be provided by human users.

The pre-image approach introduced by Lozano-Pérez, Mason, and Taylor[6], and simplified by Erdmann [4] incorporates the effect of uncertainty directly within one planning phase. The goal is to create motion plans that will avoid errors. However, this approach has unsolved theoretical problems and its applicability to practical cases is questionable. A more practical force control method was developped by Whitney[12]. His remote center compliance device can correct small insertion errors for a peg-in-hole task by applying correct forces to the peg. But it only works when the peg is in or partly in the hole.

All the previous approaches contribute to solve the problem in some ways, but none fully solves it. The problem is so complex that new solutions are still in need. The approach presented in this paper is based upon three hypotheses:

1. The problem cannot be solved in general. Design constraints relating nominal parameters, tolerances and error parameters must be satisfied if success is to be guaranteed.

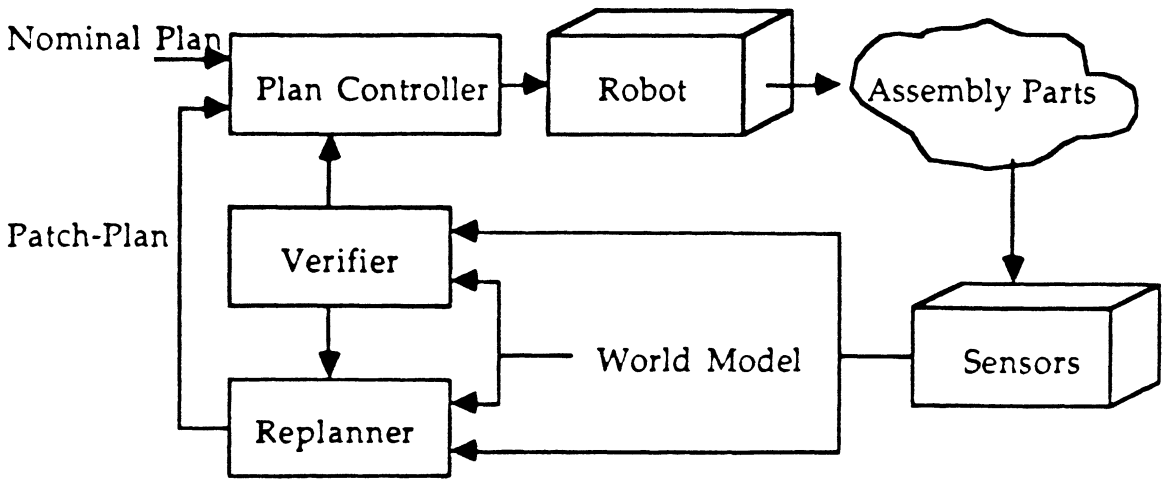

Figure 1: System framework

2. A two phase planning process, an off-line nominal planner and an on-line replanner to correct for run-time errors, simplifies the overall system.

3. Replanning can be based upon knowledge of which surfaces of the parts involved are in contact with which other surfaces.

Desai [2] has introduced the notion of *contact formations* to describe the different topological contacts among the parts being assembled. and has developed a technique for, under certain design constraints, uniquely determining the contact formation in which the parts lie in spite of sensor and control errors. When an error occurs in a nominal plan. an unintended contact occurs among surface elements of the parts involved. While there can, in general. be a large number of contact formations. there are typically only a modest number that are achievable with small deviations from the nominal plan. Thus, one can consider devising replanning strategies based upon the contact formation in which the system completes an attempted move (we presume that there are force/moment guards that stop motions that make unintended contacts). Further. since the actual positions may be presumed to be close to the nominal trajectory. simple replanning strategies, such as straight sliding motions or single axis rotations may be considered without having to worry about obstacle avoidance. In this paper, we develop such simple replanning strategies and determine design and motion constraints, which if satisfied, can be used to guarantee that the replanning strategy will be successful.

2 System Framework and General Restrictions

The system framework relevent to the research is shown in Fig. 1. The system consists of three parts: the verifier, the replanner, and the plan controller. Given the task environment(i.e. the robot, the assembly parts, and the sensors), a nominal plan, and a world model which consists of geometrical and physical descriptions of the task environment, the verifier determines contacts among the assembly parts whenever an unexpected collision occurs, and then the replanner generates patch-plans to correct errors. Each patch-plan is a path connecting the unexpected configuration to one of the expected configurations so that the nominal motion can continue. The plan controller plugs in a patch-plan to the nominal plan whenever an unexpected contact is verified. It acts like a switch controlled by the verifier.

We restrict the replanning to motions in contact[5], for errors are more crucial and more easily identified when contact occurs. Also, the desired goal of most part-mating operations involves contacts among the parts involved. We further assume that compliant motions [8][9][10] are applied wherever necessary.

3 Error Model

For a successful replanning, obviously, it is necessary to establish an appropriate error model which covers all (dominant) static and dynamic (including sensory) errors in a system. We begin by discussing the error model we use for sensory controls and robot motions.

As we know, sensor and robot motion errors (which consist of mechanical and control errors) are major errors in a system. We suppose that only a position/orientation sensor and a force/torque sensor are available. Then, under the general restrictions discussed in the last section, the following system uncertainties must be modeled:

- position/orientation sensor uncertainty;
- robot motion uncertainty (resulting in both position and orientation errors at the destination);
- force/torque sensor uncertainty.

In the following subsections, parameters to model those three kinds of uncertainties are defined respectively.

3.1 Modeling Position/Orientation Sensor Uncertainty

Let $(\mathbf{P}_a, \phi_a)$ denote a location in the system configuration space constrained by the Compliant Surface (C-Surface), where $\mathbf{P}_a$ refers to the position and ϕ_a refers to the orientation. Let $(\mathbf{P}_s, \phi_s)$ denote the corresponding sensed position and orientation on a position/orientation sensor.

Definition 1: ϵ_p is defined as the *positional uncertainty* of a position/orientation sensor, iff $\epsilon_p > 0$, and it has the smallest value satisfying the condition that $\forall \mathbf{P}_a$, $\forall \mathbf{P}_s$, if $\mathbf{P}_a$ is sensed by the position/orientation sensor as $\mathbf{P}_s$, then

$$||\mathbf{P}_s - \mathbf{P}_a|| \leq \epsilon_p.$$

Definition 2: ϵ_o is defined as the *orientational uncertainty* of a position/orientation sensor, iff $\epsilon_o > 0$, and it has the smallest value satisfying the condition that $\forall\phi_a$, $\forall\phi_s$, if ϕ_a is sensed by the position/orientation sensor as ϕ_s, then

$$\|\phi_s - \phi_a\| \leq \epsilon_o.$$

3.2 Modeling Force/Torque Sensor Uncertainty

Let $\mathbf{F}_s$ denote a force sensed by a force/torque sensor. Let $\mathbf{F}_a$ be the actual force.
Definition 3: ϵ_f is defined as the *force uncertainty* of a force/torque sensor , iff $\epsilon_f > 0$, and it has the smallest value satisfying the condition that $\forall\mathbf{F}_a$, $\forall\mathbf{F}_s$, if $\mathbf{F}_a$ is sensed by the position/orientation sensor as $\mathbf{F}_s$, then

$$\|\mathbf{F}_s - \mathbf{F}_a\| \leq \epsilon_f.$$

Let $\mathbf{M}_s$ denote a moment sensed by a force/torque sensor. Let $\mathbf{M}_a$ be the actual moment.
Definition 4: ϵ_m is defined as the *momental uncertainty* of a force/torque sensor , iff $\epsilon_m > 0$, and it has the smallest value satisfying the condition that $\forall\mathbf{M}_a$, $\forall\mathbf{M}_s$, if $\mathbf{M}_a$ is sensed by the position/orientation sensor as $\mathbf{M}_s$, then

$$\|\mathbf{M}_s - \mathbf{M}_a\| \leq \epsilon_m.$$

3.3 Modeling Motion Uncertainty

Let $\mathbf{v}_d$ denote the desired linear velocity of a robot for a desired straight-line translation. Let $\mathbf{v}_a$ denote the actual linear velocity.
Definition 5: ϵ_v is defined as the *linear velocity uncertainty* of a robot , iff $\epsilon_v > 0$, and it has the smallest value satisfying the condition that $\forall\mathbf{v}_d$, $\forall\mathbf{v}_a$,

$$\|\mathbf{v}_d - \mathbf{v}_a\| \leq \epsilon_v.$$

Let ω_d denote the desired angular velocity of a robot for a desired pure rotation. Let ω_a denote the actual angular velocity.
Definition 6: ϵ_ω is defined as the *angular velocity uncertainty* of a robot , iff $\epsilon_\omega > 0$, and it has the smallest value satisfying the condition that $\forall\omega_d$, $\forall\omega_a$,

$$\|\omega_d - \omega_a\| \leq \epsilon_\omega.$$

During a pure translation, a robot should not rotate. However, since there are various errors, such as mechanical, control, and sensory errors, a translation may not be so pure and the robot may rotate slightly. For the same reason, a not-so-pure rotation results in translation of the robot. To model these phenomena, let variable ω_t^a represent the "illegal" angular speed of the robot in an actual translation, and let variable v_r^a represent the "illegal" linear speed of the robot in an actual rotation. The following parameters can be defined.

Definition 7: ω_t is defined as the *translational orientation-uncertainty-rate* of a robot , iff $\omega_t > 0$, and it has the smallest value satisfying the condition that $\forall \omega_t^a$,

$$\|\omega_t^a\| \leq \omega_t.$$

Definition 8: v_r is defined as the *rotational position-uncertainty-rate* of a robot , iff $v_r > 0$, and it has the smallest value satisfying the condition that $\forall v_r^a$,

$$\|v_r^a\| \leq v_r.$$

4 Peg-in-hole Study

Peg-in-Hole tasks are most commonly studied in the area of uncertainty handling. But previous research either does not guarantee the success of the task or is too complex to be practical. Another important issue that hampers the previous progress is that different shaped pegs and holes require seemingly different uncertainty handling strategies. And apparently, the more complex the peg-in-hole structure is, the more complex the strategies should be. In this section, however, we apply our new approach to peg-in-hole tasks by introducing a simple replanning strategy, and discussing proper design and motion constraints necessary to the success of the strategy.

For a peg-in-hole task, it is most possible that the peg is not put properly above the hole before the insertion, so that the peg hits the entry surface of the hole and the task fails. The replanning that is of interest starts from that point. The objective is to move the peg above the hole, with error within the tolerance, so that the insertion can be done.

Next, we state some restrictions or assumptions of the study:

- the replanning motion is compliant, and the Compliant surface (C-surface) is the entry surface of the hole;
- the entry surface and the bottom surface of the peg which contacts the entry surface are planar;
- the bottom surface of the peg fully contacts the entry surface of the hole, so that the task space has only two translational and one rotational degrees of freedom(Fig. 2);
- there are no other objects (i.e. obstacles) except for the peg and the hole.

4.1 Replanning Strategy and Constraints

Let $(\mathbf{P}_a^p, \phi_a^p)$ and $(\mathbf{P}_a^h, \phi_a^h)$ be the locations of the peg and the hole respectively. Let $\mathbf{d}_a = \mathbf{P}_a^h - \mathbf{P}_a^p$, with $d_a = \|\mathbf{d}_a\|$ be the actual distance. Let $\Delta\phi_a = \phi_a^h - \phi_a^p$.

Ideally, since it is assumed that there is no obstacle around, the motion for replanning can be simply a (compliant) translation of the peg in the direction of $\mathbf{d}_a$

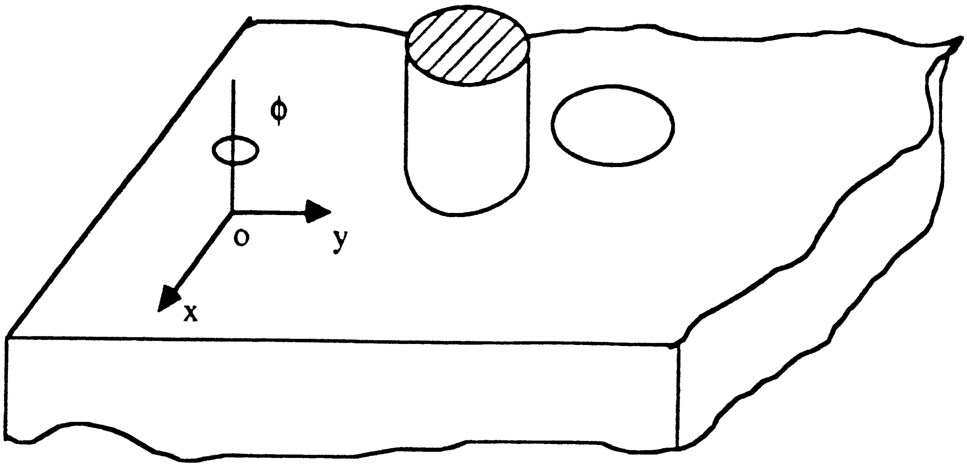

Figure 2: Peg-in-hole task space

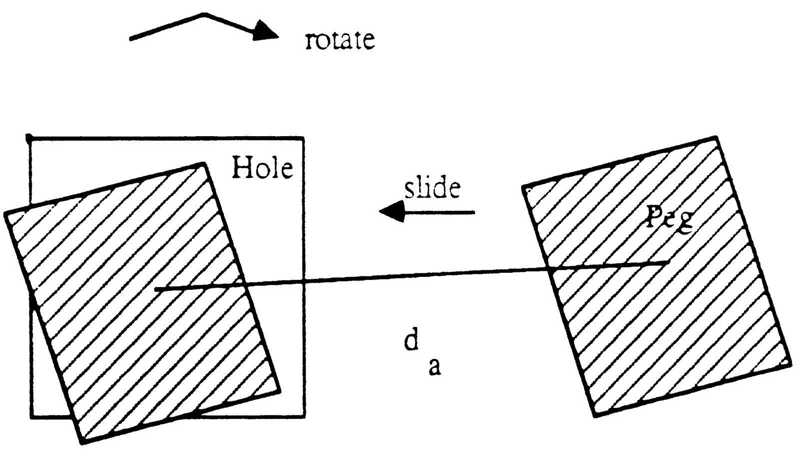

Figure 3: Simple replanning strategy in ideal case

with moving length $l = d_a$, and then a rotation of the peg to make sure that a proper ϕ_a^p is obtained for the peg to enter the hole (Fig. 3). However in a real case, because

- only a *sensed* $\mathbf{d}_s$ for $\mathbf{d}_a$ and a sensed $\Delta\phi_s$ for $\Delta\phi_a$ can be possibly obtained,
- $\mathbf{d}_s$ and $\Delta\phi_s$ are different from $\mathbf{d}_a$ and $\Delta\phi_a$ respectively due to sensory errors,
- there are also motion errors,

that ideal strategy with only *one* translation and *one* rotation can not guarantee the success of replanning. A modified, more realistic strategy must be set up.

In essence. the proposed strategy consists of more than one motion step. Each *desired* motion step is formed by a (compliant) translation along the direction of the sensed $\mathbf{d}_s$, and a rotation to correct the orientation error caused by the translation (if necessary). The distance l of the translation is constrained with respect to both the sensor error in $\mathbf{d}_s$ and the motion error in actual moving, so that after each motion step, d_a always becomes smaller. In this way, the positional replanning can be convergent. After the success of the positional replanning (i.e. after d_a is less than the positional tolerance of the task), the peg is rotated to correct $\Delta\phi_a$. Note that since motion error contributed to $\Delta\phi_a$ is corrected in each motion step, there must be $\Delta\phi_a \leq \epsilon_c$. (recalling that ϵ_o is the uncertainty bound in orientation sensing). If a rotation step can correct $\Delta\phi_a$, while not destroying the success in position, the success of the whole replannning can be guaranteed. For realizing the strategy, developing proper design constraints and motion constraints (on l of each motion step) is fundamental.

In the following subsections, we will discuss constraints for successful replanning in two situations:

- only position/orientation sensing is available;
- in addition to position/orientation sensing, force/torque sensing is also available.

4.2 Using Position/Orientation Sensing only

For convenience, yet without losing generality, we restrict our study to those peg-and-holes with ***regular polygon*** bases. Let r_h be the *nominal* radius of the inner tangent circle of a hole. and r_h^a be an actual radius. Let r_p be the *nominal* radius of the inner tangent circle of the corresponding peg, and r_p^a be an actual radius.

Definition 9: δ_h is defined as the *tolerance* of the hole, iff $\delta_h > 0$, and it has the smallest value satisfying the condition that $\forall r_h^a$,

$$\|r_h - r_h^a\| \leq \delta_h.$$

Definition 10: δ_p is defined as the *tolerance* of the peg, iff $\delta_p > 0$, and it has the smallest value satisfying the condition that $\forall r_p^a$,

$$\|r_p - r_p^a\| \leq \delta_h.$$

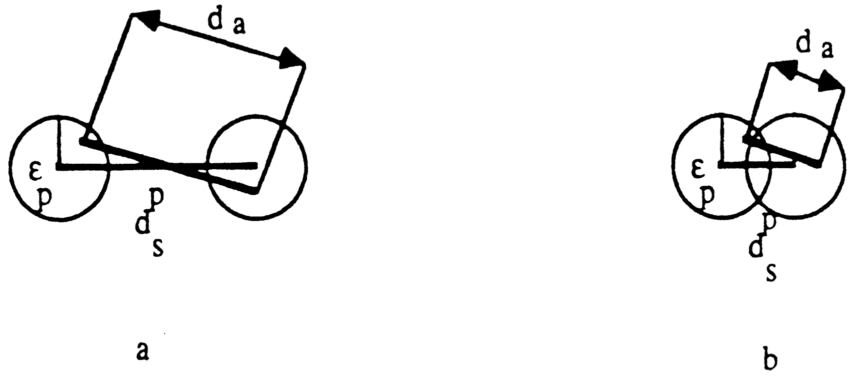

Figure 4: Relationship between $\mathbf{d}_s^p$ and $\mathbf{d}_a$

Definition 11: The *tolerance of the task* is defined as δ, such that

$$\delta = r_h - r_p - \delta_h - \delta_p.$$

Our objective is to seek relations among the position/orientation sensor uncertainty ϵ_p, ϵ_o; the motion uncertainty ϵ_v, ϵ_ω, v_r, ω_t; the (desired) nominal linear speed v_d and angular speed ω_d of the robot; and the task tolerance δ, such that under those relations (or constraints), the success of the replanning strategy (described previously) can be assured.

Let $\mathbf{d}_s^p$ indicates the sensed distance vector, i.e.,

$$\mathbf{d}_s^p = \mathbf{P}_s^h - \mathbf{P}_s^p \tag{1}$$

where ^{p} in d_s^p indicates position sensing, and $d_s^p = \|\mathbf{d}_s^p\|$ indicates the sensed distance. Fig. 4 shows the relationship between $\mathbf{d}_a$ and $\mathbf{d}_s^p$. Obviusly, the position sensor uncertainty ϵ_p may cause $\mathbf{d}_s^p$ to differ from $\mathbf{d}_a$ both in direction and in magnitude. Consider when $d_s^p \leq 2\epsilon_p$ (as shown in Fig. 4a), the direction of $\mathbf{d}_s^p$ can be as bad as the opposite of the direction of $\mathbf{d}_a$. And from the magnitude d_s^p, it can not be assured whether the goal relationship in position between the peg and the hole (i.e. $d_a \leq \delta$) has been achieved. Consequently, the next replanning motion can not be determined. However, in Fig. 5 where $d_s^p > 2\epsilon_p$, the angle

$$\theta = \arcsin\left(\frac{2\epsilon_p}{d_s^p}\right)$$

indicates the difference between the direction of $\mathbf{d}_s^p$ and the direction of $\mathbf{d}_a$ *in worst-case*. (Note that the worst-case direction of motion is obtained by finding the line tangent to circles of radius ϵ_p centered at $\mathbf{P}_s^p$ and $\mathbf{P}_s^h$ respectively). Clearly θ becomes bigger as d_s^p gets smaller. Thus, for a small $d_s^p > 2\epsilon_p$, θ can still be large enough that the direction of $\mathbf{d}_s^p$, namely the commanded direction of the motion, may not result in

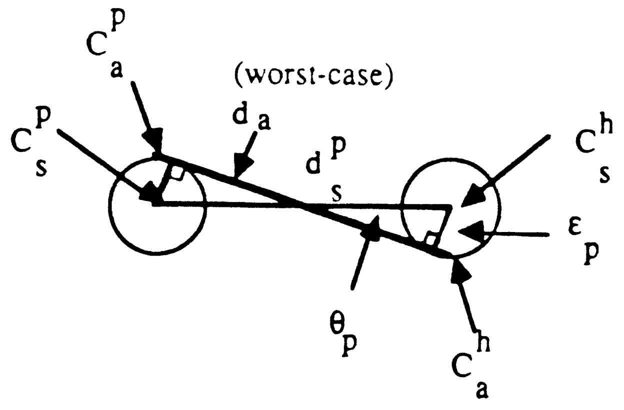

Figure 5: The sensed direction vs. the worst-case actual direction

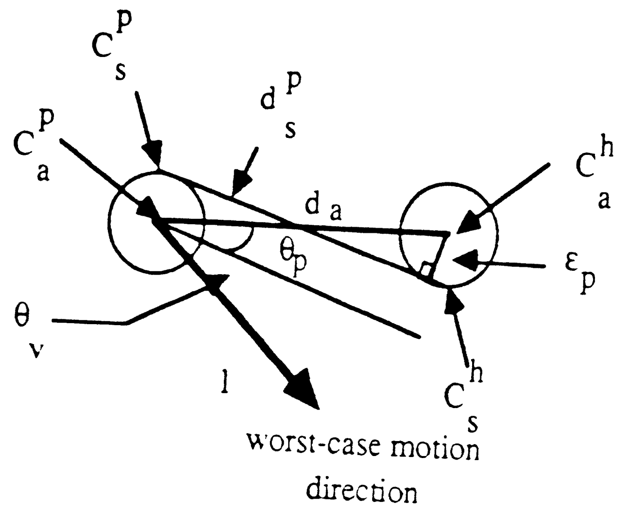

Figure 6: The worst-case direction under position sensing

reducing d_a. Thus, we need to determine a bound, θ_p, on θ such that if $\|\theta\| \leq \theta_p$, the direction of motion can be controlled to those directions that lead to the reduction of d_a. Since when the robot actually moves, its mechanical and control error also results in direction uncertainty, we must have the constraint

$$0 \leq \theta_v + \theta_p < \pi/2 \tag{2}$$

hold(Fig. 6), where

$$sin(\theta_v) = \epsilon_v/v_d \tag{3}$$

(Fig. 7). It is obvious that the determination of θ_p depends on the value of ϵ_v and the nominal speed v_d. We call the parameter θ_p the *maximum angle allowed* between $\mathbf{d}_s^p$ and $\mathbf{d}_a$.

Recalling the relationship between d_a and d_s^p (Fig. 4), when

$$d_s^p < \frac{2\epsilon_p}{\sin(\theta_p)},$$

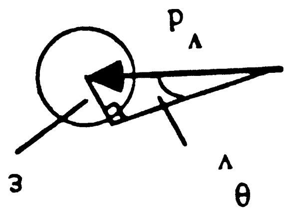

Figure 7: Relationship between ϵ_v and θ_v

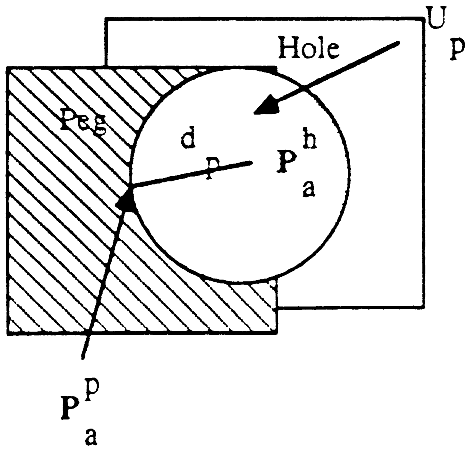

Figure 8: The position sensing uncertain area

d_a might be as small as zero and is bounded by

$$0 \leq d_a \leq d_s^r + 2\epsilon_p < d_p$$

where

$$d_p = \frac{2\epsilon_p}{\sin(\theta_p)} + 2\epsilon_p. \tag{4}$$

Call d_p *the minimum distance allowed* by position sensing. Then the area inside the circle about $\mathbf{P}_a^h$ of radius d_p can be defined as the *position sensing uncertain area*, denoted by U_p as in Fig. 8.

If positional replanning is to be successful, we must require that $d_p \leq \delta$, i.e. the position sensing uncertain area is contained in the goal position area of the peg. However, to also guarantee the success of the orientational replanning, the requirement must be stronger. According to our replanning strategy, the orientational replanning takes place when the positional replanning succeeds, i.e., when $d_a \leq \delta$ and $\Delta\phi_a \leq \epsilon_o$. Thus, the orientational replanning is a simple rotation to correct $\Delta\phi_a$ *until* the peg enters the hole. To accomplish this, we must guarantee that *after* rotating the peg $\Delta\phi_a$, the peg will still be in a goal position. In other words, if $d_a^b (\leq d_p)$ indicates

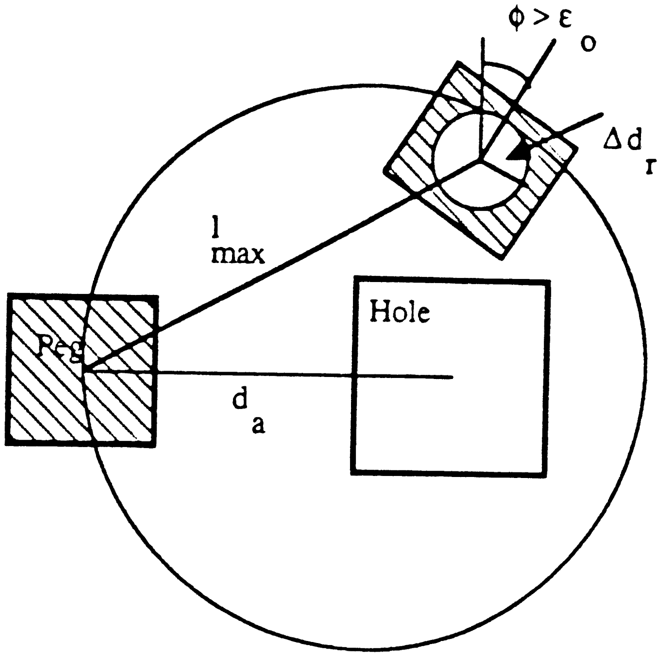

Figure 9: l_{max} and motion constraint

the actual distance between the peg and the hole *before* the rotation, and Δd_r indicates the distance of the illegal translation caused by the rotation, the requirement $d_a^t + \Delta d_r \leq \delta$ must be satisfied. For the requirement to hold, the following design constraint is sufficient:

$$d_p + \frac{v_r \epsilon_o}{\omega_d - \epsilon_\omega} \leq \delta. \tag{5}$$

In order to make the positional replanning convergent, in addition to the design constraint (2), proper motion constraint on the length of the translation step l must be enforced. l must be less than some maximum value l_{max} (as shown in Fig. 9), so that after each motion step (i.e. a translation of distance l and a rotation to correct the orientation error caused by the translation if the error exceeds the orientation sensing uncertainty ϵ_o), the actual distance between the peg and the hole is shorter than that before the motion step. In particular, it can be shown that

$$l_{max} = \frac{2(d_s^r - 2\epsilon_p)(\cos(\arcsin(2\epsilon_p/d_s^r) + \theta_v) - f)}{1 - f^2} \tag{6}$$

where d_s^r is the sensed distance, and

$$f = \frac{v_r \omega_t}{(v_d - \epsilon_v)(\omega_d - \epsilon_\omega)}. \tag{7}$$

Also, for $l < l_{max}$, the following design constraint is necessary:

$$f < \cos(\theta_p + \theta_v). \tag{8}$$

Due to motion error ϵ_v, the *actual* distance of translation l_a may be different from the *desired* distance of translation l. Specifically, we have

$$||l - l_a|| \leq \epsilon_v t$$

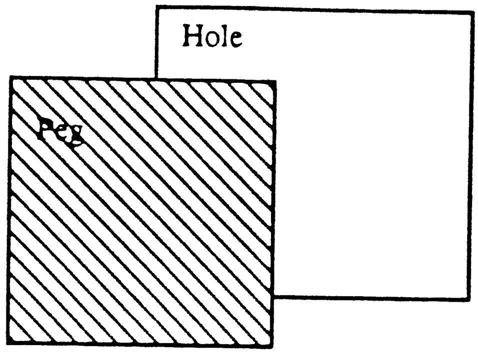

Figure 10: The peg partially overlaps the hole

where $t = l/v_d$ is the desired time of translation, assuming that the timer is perfect. Thus, to guarantee the convergence of the replanning, we must have

$$\epsilon_v t < l < l_{max} - \epsilon_v t$$

which lead to the design constraint

$$v_d > 2\epsilon_v \tag{9}$$

and the motion constraint

$$\frac{\epsilon_v}{v_d} l_{max} < l < (1 - \frac{\epsilon_v}{v_d}) l_{max}. \tag{10}$$

In summary, to guarantee the success of both the positional and the orientational replanning, the design constraints formulated as inequalities (5), (2), (8), and (9), and the motion constraint expressed by (10) have to be inforced.

4.3 Using Force/Torque Sensing to Compensate Position/ Orientation Sensing

4.3.1 General Discussion

As shown previously, the area inside the circle about $\mathbf{P}_a^h$ of radius d_p (see equation 4) is defined as the *position sensing uncertain area*, denoted by U_p as in Fig. 8. When only a position/orientation sensor is used to guide the replanning, the position sensing uncertain area must be contained in the goal position area of the peg, i.e. $d_p \leq \delta$ must be satisfied, so that the replanning can succeed. However, if force/torque sensing is also involved, the constraint $d_p \leq \delta$ can be weakened or even dropped. Consider the situation where $\mathbf{P}_a^p$ is in U_p. The fact that a force/torque sensor can possibly sense a moment when the peg is partially overlapping the hole(Fig. 10) suggests us that even if the peg reaches the position uncertain area, the replanning process may not have to be terminated, since an ineffective position sensing might be compensated by an effective moment sensing to continue guiding the replanning.

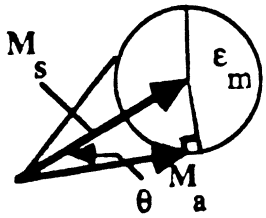

Figure 11: Directional difference between M_s and M_a in worst case

In the following discussion, we will define the moment-sensible area, denoted S_m, as the set of actual position volume of the peg for which moment sensing alone can

1. detect the goal locations;
2. determine a direction of motion along which d_a can be reduced.

Obviously we will then want $S_m \supseteq U_p$. In order to describe S_m more precisely and use the above containment relation to obtain design constraints, we need to introduce two parameters θ_m and d_m, with the idea similar to that behind the introduction of θ_p and d_p.

As an alternative to determining the direction of motion from the position sensor, we can determine a direction of motion from the moment sensed, i.e., the direction of motion would be orthogonal to the projection of $\mathbf{M}_s$ onto the contact plane. Hence, the uncertainty in the commanded direction of motion can be estimated exactly as the directional uncertainty of $\mathbf{M}_s$. Fig. 11 shows the directional difference between $\mathbf{M}_s$ and $\mathbf{M}_a$ in *worst case*, as indicated by the angle θ. The smaller is M_s, the bigger θ can be. Thus in analogy to the introduction of θ_p, we need to introduce a bound θ_m on θ, such that when $||\theta|| \leq \theta_m$, in other words when

$$M_s \geq \frac{\epsilon_m}{\sin\theta_m},$$

$\mathbf{M}_s$ gives a direction of motion along which d_a can be reduced. Also in analogy to the case of position sensing, the design constraint

$$0 \leq \theta_v + \theta_m < \pi/2 \tag{11}$$

where θ_v is expressed by equation (3), is necessary while taking into account the motion error in direction. The determination of θ_m depends on the value of θ_v. We call the parameter θ_m the *maximum angle allowed* between the sensed moment $\mathbf{M}_s$ and the actual moment $\mathbf{M}_a$.

It is easily observed that given a specific environment (i.e. the robot, the solid peg and hole, and the force applied to the peg by the robot, etc), the *relative* location

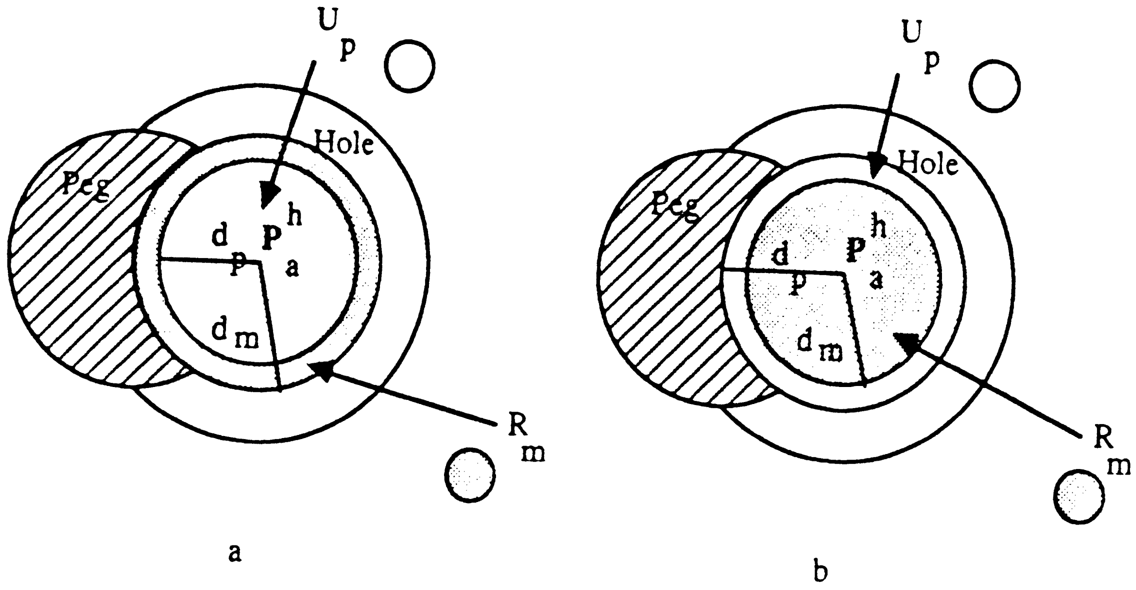

Figure 12: Relationship between U_p and R_m

$(\mathbf{d}_a, \Delta\phi_a)$ determines the actual moment $\mathbf{M}_a$ (applied to the force/torque sensor). The *closer* the peg to the hole, the *bigger* is M_a, hence the *bigger* is the sensed moment M_s, and therefore the *smaller* is the angle θ(Fig. 11). Thus, given θ_m, a parameter d_m can be defined.

Definition 12: d_m, called the *maximum moment-sensing distance allowed* of a peg-in-hole task with a force/torque sensor, has the largest value (> 0) satisfying the condition that $\forall \mathbf{d}_a$, if $d_a > d_m$, then $\exists \Delta\phi_a$, such that at $(\mathbf{d}_a, \Delta\phi_a)$, $M_s < \frac{\epsilon_m}{\sin\theta_m}$, (i.e. one can not guarantee to determine an adquate $\mathbf{d}_s^m$ from the sensed moment). Then $\forall \mathbf{d}_a$ $\forall \Delta\phi_a$ if $d_a \le d_m$, $M_s \ge \frac{\epsilon_m}{\sin\theta_m}$, and it is possible to determine a direction of motion from the moment sensed that guarantee to reduce d_a. d_m is a function of ϵ_m, ϵ_f, θ_m, the sensed force F_s, and the task geometry.

Denote the circular area centered at $\mathbf{P}_a^h$ of radius d_m by R_m. Then clearly $S_m \supseteq R_m$.

As the result of the above definitions and discussions, it is obvious that if $U_p \subseteq R_m$ (as shown in Fig. 12a), then

1. in combination, the position and force/torque sensors can detect the goal locations;

2. in combination, the position and force/torque sensors can determine a direction of motion along which d_a can be reduced;

since the position uncertain area is compensated by the moment-sensible area. Wherease, if $U_p \supset R_m$, since the area $U_p - R_m$ is both position sensing uncertain and moment sensing uncertain(Fig. 12b), adding force/torque sensing can not help in droping the constraint $d_p \le \delta$ to guarantee the success of the replanning. Note that one can increase d_m by increasing the applied force.

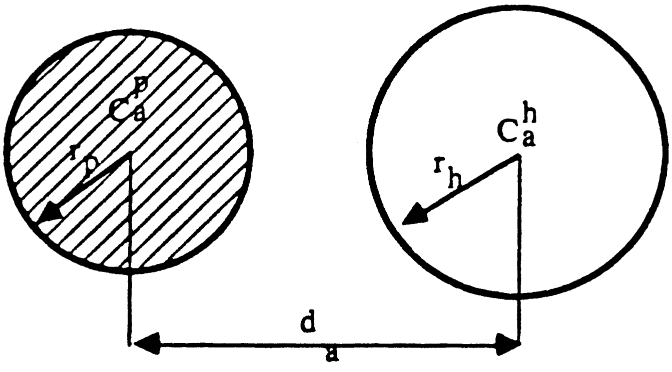

Figure 13: A top view of cylindrical peg and hole

4.3.2 Cylindrical Peg-in-Hole Case Study

In general, it is very difficult to obtain constraints under which the replanning strategy can always be successful, when both position/orientation and force/torque sensings are involved. It is still one of our research topics. However, for a special case: cylindrical peg and hole (CPH) tasks, we have obtained an analysis on the design and motion constraints that shows the applicability of the idea (i.e. using both the position/orientation and the force/torque sensing to guide the replanning).

Consider a cylindrical peg-in-hole task (Fig. 13). First, we need to find out how to determine d_m, so that the constraint

$$d_p \leq d_m \tag{12}$$

(or $U_p \subseteq R_m$ as shown in Fig. 12a) can be explicitly expressed. Then we need to find out how to estimate d_a from force/moment sensing when the position of the peg is within U_p, so that l_{max} and the motion constraint that lead to the reduction of d_a can be formed. Note that for CPH tasks, orientations do not need to be considered due to symmetry. Thus, each replanning step does not involve a rotation, and the whole replanning is positional only. Therefore, orientation-related uncertainties ϵ_ω, ϵ_ω, v_r, and ω_t will not appear in the constraints.

For convenience, let the center of the base of the peg C_a^p be the reference position of the peg, and the center of the base of the hole C_a^h be the reference position of the hole. Let r_p be the *nominal* radius of the peg and δ_p denote the *tolerance* of the peg (defined as in the polygon case). Let r_h be the *nominal* radius of the hole, and δ_h be the *tolerance* of the hole. Then, similar to the regular polygon case (discussed previously), we can define the tolerance δ of a cylindrical peg and hole case:

$$\delta = r_h - r_p - \delta_h - \delta_p.$$

For simplicity, in the following discussions, we will not count for δ_h and δ_p, for it can be seen from the derivations below that the effects of δ_h and δ_p are negligible.

We now use a simple and rather conservative method to determine d_m and to estimate d_a. Let F_s be the value of the sensed force. Let M_s be the value of the

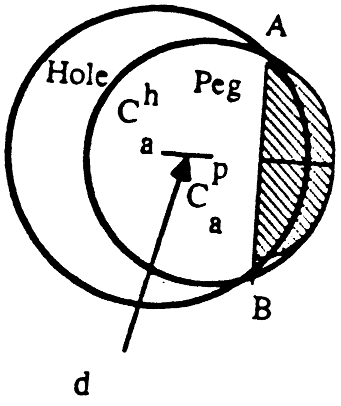

Figure 14: C^p_a is on the left of line AB

sensed moment *about* C^p_a. Consider the situation shown in Fig. 14 where C^p_a is on the left of line AB. Let the distance between C_{AB} (i.e. the center of the line AB) and C^p_a be Δ. Then it can be easily shown that

$$d_a = \sqrt{\delta(r_p + r_h) + \Delta^2} - \Delta. \tag{13}$$

The next step is to estimate Δ from F_s and M_s, so that an estimated value for d_a can be obtained, since we can not get the exact d_a. Note that we always want to *underestimate* d_a in order to get a conservative l_{max} for each translation step l.

Assume that the contact force distribution is symmetric about the line going through the two centers. The real action-point of the contact force should be within the shaded area, and somewhere on the line. Let

$$R_s = \frac{M_s}{F_s}.$$

Then R_s is the *sensed* distance between the action-point and C^p_a. Since the *actual* distance between the action-point and C^p_a must be *greater than or equal to* Δ, taking into account the sensing uncertainties ϵ_f and ϵ_m, the following must hold:

$$\frac{M_s + \epsilon_m}{F_s - \epsilon_f} \geq \Delta.$$

If we use

$$R_{max} = \frac{M_s + \epsilon_m}{F_s - \epsilon_f} \tag{14}$$

to replace Δ in equation (13), then we get d^m_s, such that

$$d^m_s = \sqrt{\delta(r_p + r_h) + R^2_{max}} - R_{max}. \tag{15}$$

Clearly,

$$d^m_s \leq d_a$$

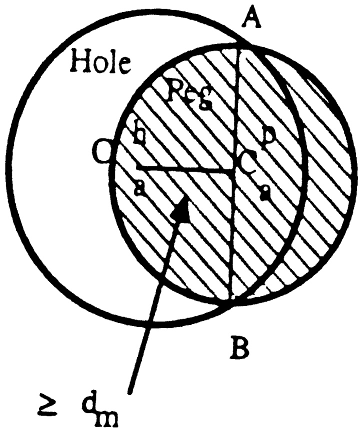

Figure 15: Possible range for d_m

i.e. d^m_s underestimates d_a. Note that it makes sense to use this method to estimate d_a only when $\mathbf{C}^p_a$ is on the left of line AB. Therefore, d_m in this case, should be less than or equal to the distance between $\mathbf{C}^p_a$ and $\mathbf{C}^h_a$ when $\mathbf{C}^p_a = \mathbf{C}_{AB}$ (Fig. 15). More explicitly,

$$d_m \le \sqrt{\delta(r_p + r_h)}.$$

Since it is extremely difficult to obtain the contact force distribution. As the result, the *exact* d_m following definition 12 is almost impossible to be known. Thus, we can only expect to get a close d_m sufficient to the condition that $\forall \mathbf{d}_a$ if $d_a \le d_m$.

$$M_s \ge \frac{\epsilon_m}{\sin\theta_m}, \tag{16}$$

so that it is possible to determine a direction of motion from $\mathbf{M}_s$ that guarantee to reduce d_a. Here we introduce a simple method to determine such a d_m. Since

$$F_a\Delta \le M_a$$

(see Fig. 14 and Fig. 15), and it is sufficient for (16) to hold if

$$M_a - \epsilon_m \ge \frac{\epsilon_m}{\sin\theta_m},$$

thus it is sufficient for (16) to hold if

$$F_a\Delta - \epsilon_m \ge \frac{\epsilon_m}{\sin\theta_m},$$

or

$$\Delta \ge \frac{\epsilon_m}{F_a}(1 + \frac{1}{\sin\theta_m}).$$

From the above inequality and equation (13), we can finally have

$$d_m = \sqrt{\delta(r_h + r_p) + (\frac{\epsilon_m}{F_a}(1 + \frac{1}{\sin\theta_m}))^2} - \frac{\epsilon_m}{F_a}(1 + \frac{1}{\sin\theta_m}). \tag{17}$$

Since we usually only get $\mathbf{F}_s$, replace F_a in (17) by $F_s - \epsilon_f$, we get the alternative d_m:

$$d_m = \sqrt{\delta(r_h + r_p) + (\frac{\epsilon_m}{F_s - \epsilon_f}(1 + \frac{1}{\sin\theta_m}))^2} - \frac{\epsilon_m}{F_s - \epsilon_f}(1 + \frac{1}{\sin\theta_m}). \tag{18}$$

Now the design constraint (12) can be explicitly expressed by (4) and (18). Under this and the other design constraint expressed by inequality (2), when $d_s^p > \frac{2\epsilon_p}{\sin(\theta_p)}$, position sensing is effective. On the other hand, when $d_s^p \leq \frac{2\epsilon_p}{\sin(\theta_p)}$, the robot motion should follow $\mathbf{d}_s^m$ determined by force/moment sensing, subject to the further design constraint (11) holding. The design constraint (9) also has to be enforced so that the motion constraint in the form represented by inequality (10) can hold. It can be easily shown that in CPH case

$$l_{max}^p = 2(d_s^p - 2\epsilon_p)\cos(\theta_v + \arcsin(2\epsilon_p/d_s^p)), \tag{19}$$

where ^{p} indicates sensing by position sensor, and d_s^p is formulated by equation (1), and that,

$$l_{max}^m = 2d_s^m \cos(\theta_v + \arcsin(\epsilon_m/M_s)), \tag{20}$$

where ^{m} indicates sensing by force/moment sensor, and d_s^m is formulated by equation (15). In both equations (19) and (20), θ_v is the angle formulated by equation (3).

In summary, in order for the replanning in CPH case to succeed, the design constraints (12), (9), and

$$0 \leq \arcsin(\epsilon_v/v_d) + \max(\theta_p, \theta_m) < \frac{\pi}{2} \tag{21}$$

(where the value of θ_p and θ_m can be determined by the constraints), and the motion constraint

$$\frac{\epsilon_v}{v_d} l_{max}^p < l < (1 - \frac{\epsilon_v}{v_d}) l_{max}^p \qquad \text{if } d_s^p > \frac{2\epsilon_p}{\sin(\theta_p)} \tag{22}$$

$$\frac{\epsilon_v}{v_d} l_{max}^m < l < (1 - \frac{\epsilon_v}{v_d}) l_{max}^m \qquad \text{otherwise} \tag{23}$$

where l_{max}^p is expressed by equation (19) and l_{max}^m is expressed by equation (20), have to be satisfied.

4.4 Simulation Results

To test the constraints and the replanning strategy, we designed computer programs to simulate the replanning process for both the regular-polygon-based peg-in-hole (RPH) cases and the CPH cases. The computer simulation we have done shows that the design constraints obtained are correct and practically reasonable, and that under the design constraints, the motion constraint is *necessary* for guaranteeing the success and efficiency of the strategy. Furthermore, we observed that *as long as the design and motion constraints are satisfied*,

- the replanning process is not sensitive to changes in the initial relative positions of the peg and the hole,
- the replanning process is not sensitive to changes of the value of ϵ_v over a reasonably large range,
- for RPH tasks, the error effects of v_r, ω_t are negligible,
- for CPH tasks, the introduction of force/moment sensing can guarantee the strategy to be successful with a δ much less than it must be when only position sensing is available.

All observations above show the advantage of the replanning strategy. In particular, the second observation means that the motion error will not significantly affect the replanning strategy. Therefore, a rough over-estimation of ϵ_v could be enough for use of the strategy. The third observation suggests that no rotation to correct the effects of v_r and ω_t will be necessary, which simplifies the strategy. The last observation is most exciting, which shows that using force/moment sensing to compensate position/orientation sensing may greately improve the quality of the replanning process, and it means that more research on this may eventually widens the range of the part-mating tasks that can be automatically done by robots.

Table 1 and Table 2 show some of the simulation results for CPH cases, given the following position sensor and force/torque sensor uncertainties [1]

$$\epsilon_p = 0.025 \ \ \text{mm}$$
$$\epsilon_f = 0.014 \ \ \text{kg}$$
$$\epsilon_m = 0.740\text{mm-kg}.$$

We also choose the commanded linear speed of the robot to be

$$v_d = 20 \text{ mm/s} ,$$

with motion uncertainty

$$\epsilon_v = 1.0 \text{ mm/s} .$$

Here we use

- $MAX\#SS$: Maximum Number of Steps to Success, and
- $MEAN\#SS$: Mean Number of Steps to Success

as parameters to evaluate the replanning strategy, under the condition that the success is guaranteed by all design and motion constraints being satisfied. We also use the parameter $\%SUCCESS$ as the percentage of successful replanning cases when *not all* design constraints are satisfied. In Table 1 and Table 2, the statistical parameters were obtained by running over 100 trials, while $MAX\#SS$ and $MEAN\#SS$ were

[1]The values chosen were based on commercially available sensors.

$\theta_p = 80^\circ \quad d_p = 0.101 \quad \text{mm}$

δ (mm)	$MAX\#SS$	$MEAN\#SS$	$\%SUCCESS$	$d_p \leq \delta$
0.105	6	2	100	satisfied
0.101	7	3	100	
0.095	7	3	67	violated
0.090	8	3	45	

Table 1: Use position sensing only

$\theta_p = \theta_m = 80^\circ \quad d_p = 0.101 \quad \text{mm}$

$r_h = 5.0 \quad \text{mm}$

$F_a = 1.0 \quad \text{kg}$

r_p(mm)	δ(mm)	d_m(mm)	$MAX\#SS$	$MEAN\#SS$	$\%SUCCESS$	$d_p \leq d_m$
4.968	0.032	0.103	4	2	100	satisfied
4.970	0.030	0.097	4	2	100	violated
4.975	0.025	0.081	4	2	99	
4.980	0.020	0.065	5	3	97	
4.985	0.015	0.049	6	3	92	
4.987	0.013	0.043	6	4	85	
4.990	0.010	0.033	8	4	64	

Table 2: Use position and force/torque sensing

among successful trials. The two tables show how the task tolerance δ affects the replanning strategy without and with force/moment sensing respectively. Here d_p was obtained by equation (4), and d_m by (17). Table 1 shows that the design constraint $d_p \leq \delta$ is just necessary since when it is violated, $\%SUCCESS$ decreases sharply. But from Table 2, we see that even when d_p is greater than the computed d_m, the replanning strategy still performs well. That suggests that the selection for d_m could be relaxed a little, and the real case could be more optimistic than what we get theoretically.

5 Conclusions

With the idea of exploring the relationships between tolerances for successfully accomplishing a task and bounds on uncertainties so that an uncertainty handling strategy can guarantee the success, we have developped a replanning motion strategy and the design and motion constraints necessary for the success of the strategy for peg-in-hole tasks. We have also interpreted the constraints and tested the results by simulation. Our next step is to establish a testbed on the puma 560 robot, to further explore the applicability of the strategy.

Acknowledgements

We would like to thank Professor J. R. Barber for his helpful suggestions.

References

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

[2] R. S. Desai, and R. A. Volz, 1986(Sept.). Contact formations — a new approach to local motion in mechanical assembly. IBM Technical Report, Robotics Research Laboratory, Department of Electrical Engineering and Computer Science, University of Michigan.

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

[4] M. Erdmann, 1986. Using backprojections for fine motion planning with uncertainty. *Int. J. Robotics Res.* 5(1):19.

[5] J. Hopcroft, and G. Wilfong, 1986. Motion of objects in contact *Int. J. Robotics Res.* 4(4):32.

[6] 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.

[7] T. Lozano-Pérez, 1983. Task planning. *Robot Motion: Planning and Control*, M.Brady et al. Eds.

[8] M. T. Mason, 1981. Compliance and force control for computer controlled manipulators. *Robot Motion: Planning and Control*, M.Brady et al. Eds.

[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. Hybird position/force control of manipulators. *Robot Motion: Planning and Control*, M.Brady et al. Eds.

[11] R. H. Taylor, 1976. The synthesis of manipulator control programs from task-level specifications. *AIM-282.* Stanford Univ.

[12] D. E. Whitney, 1977. Force feedback control of manipulator fine motions *J. Dynamics Sys. Measurement, and Control.* 98:91.