Programmed Compliance for Error Corrective Assembly

MICHAEL A. PESHKIN, MEMBER, IEEE

Abstract—Suppose that before an assembly task commences, we can specify at will the manipulator's response to assembly forces by providing a single compliance (or damping) matrix to be used for the duration of the operation. Can we choose the matrix elements so that the force that characterizes every possible error condition maps into a motion that reduces it? If so, we are assured that the operation will evolve toward decreased errors and eventual success. In this paper, we describe a framework and a method of synthesis of an error-corrective matrix.

I. Background and Motivation

IT IS WIDELY recognized that for robots to perform tasks involving the contact of rigid objects, some form of compliance,[1] either of the robot or of the objects, is required. Under position control and in the absence of such compliance, the presence of even the slightest error in the positions of the objects can produce unbounded contact forces and, therefore, failure of the operation.

Depending on the task being performed and the geometry of the objects in contact, different forms of compliance are needed—compliance should be specific to the task. Mason [22] formalizes this idea, traces its origins in the literature, and describes a method of creating a compliance matrix *consistent* with a given task. The term *consistent* is introduced here in order to contrast it with *error corrective*, below. By consistent we mean that the compliance of the manipulator is such that unbounded forces do not develop. Roughly speaking, this requires that the manipulator's stiff (position-controlled) axes not be perpendicular to rigid surfaces in the environment. Recent work on the specification of compliance for consistency with a task may be found in [6] and [19].

The compliance of a manipulator has a profound effect on the evolution of a robotic assembly operation. Depending upon its compliance a manipulator's response to errors may drive the system toward or away from proper mating of the parts. Even if a compliance matrix consistent with a task is in use (i.e. unbounded forces do not develop), the operation will not necessarily evolve toward decreased errors. We will call a form of compliance *error-corrective* if it causes an

operation to evolve towards decreased errors. Fig. 1 shows an example in which a compliance that is consistent with a task is distinctly different from one that is both consistent and error corrective.

Since Mason's work, several distinct approaches to the automated planning of contact and assembly tasks force control[2] have developed:

In Lozano-Perez, Mason, and Taylor (LMT) fine motion planning [4] [11], [18], [21], the manipulator is given *a priori* a particular form of compliance (usually a generalized damper with a diagonal damping matrix; see Section I-D). The goal of LMT planning is to design a nominal motion plan. The nominal motion plan is a sequence of robot motions (usually given as velocities), where each one is executed until motion ceases; from there, the next motion commences. It is sometimes possible to design a sequence of motions that proceeds inevitably despite the initial errors to a final stopping point at which the operation (perhaps the mating of two parts) is completed.

Error detection and recovery (EDR) [8], [9] shares with LMT fine motion planning the same paradigm of a given (usually diagonal) compliance and a synthesized (generally multistep) nominal motion plan. The objective of the nominal motion plan is now broadened so that a plan that inevitably leads either to task completion *or to termination at a recognizable failure configuration* is acceptable. If motion terminates at other than task completion, a secondary nominal motion plan, which commences from that point, is invoked. EDR plans, like LMT plans, are fully formed off line.

In replanning [7], [28], both the compliance and the nominal motion plan are given *a priori*. The nominal motion plan, in particular, is just the one that would be created without any consideration of errors and is unlikely to involve multiple sequential sliding motions. If motion terminates at other than task completion, and the resulting configuration is identifiable by force sensors or other sensors, an on-line replanner is invoked. The replanner computes the motion needed to bring the workpiece back into a configuration such that the nominal motion plan can proceed.

The above works are approaches to the planning of *generic* assembly tasks, as is this paper. One particular assembly task (peg-into-chamfered-hole) has, due to its importance, been heavily studied. Draper Lab's analysis of peg-into-hole led to

[1] We use the generic meaning of the term "compliance," which is the relationship of a manipulator's motion to the forces imposed on it. (Compliance has a specific meaning as well, namely, the matrix characterizing spring-like behavior; see Section I-D). In this paper, we are designing an accommodation matrix, which is the inverse of a damping matrix.

[2] By "force control," we include all forms of control in which the response of the manipulator to forces it encounters is intentionally controlled. Among these are free joints, hybrid control, compliance control, stiffness control, impedance control, and many generalizations and extensions.
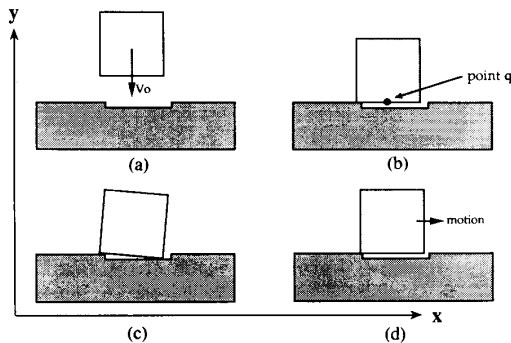
Fig. 1. Simple example of a task in which compliance consistent with the task will indeed prevent unbounded forces from developing but will not cause error reduction and eventual success of the mating operation: (a) Block is to be placed into a shallow depression or detent in a rigid surface. Since the tasks's rigid axis is the y axis, Mason's work would suggest that we make the manipulator soft (force controlled) along the y direction and stiff (position controlled) along the x axis and the θ (rotational) axis, thus making it consistent with the mechanical constraints imposed by the task. Forces due to contact, as shown in (b), are in the $+y$ direction, and there is a clockwise torque about point q. The compliance of the manipulator allows some "give" in the y direction and therefore, the block comes to rest on the lip of the detent. If we let the θ axis also be soft, the outcome is (c), in which the block rests on the lip and bottom of the detent. The evolution is shown in (d) if we have arranged the compliance of the manipulator for error reduction as well; in particular we include an off-diagonal matrix element that maps torque into horizontal displacement. Clockwise torque maps into $+x$ motion and counterclockwise into $-x$ motion. Thus, when the corner of the block comes into contact with the left edge, the torque generated results in a displacement, which moves the block toward proper mating. A compliance that maps torques into translations in the way described above may seem counterintuitive (see also Section I-D). Intuition tells us that sliding toward proper mating will not occur unless the environment surface is steeply sloped toward the detent (i.e., has chamfered edges). Our intuition, however, corresponds to a very limited class of compliances. If we allow ourselves freedom in choosing a compliance matrix, we can obtain the surprising behavior above. This behavior in no way violates the "friction cone" construction from classical mechanics. Classical mechanics says that imposed forces within the friction cone result in sticking (no motion). Here, in contrast, we do not impose a force and observe the motion that occurs. Rather we observe a force and then compute and impose a motion.

development of the remote-center compliance (RCC) wrist. RCC [27] can be implemented as a physical wrist instead of a control loop involving force sensor and computer. In RCC, a simple nominal motion is assumed in which errors of peg orientation are ignored. It is the compliance of the grasped peg, and the forces that arise due to errors, which cause the peg to self-align with the hole.

Another line of work, which results in what are sometimes described as *skeleton strategies* for particular tasks, again uses multistep motion plans, which are sometimes combined with nondiagonal compliances. Strategies for round pegs were considered in [17] and [24]. More recent work considers strategies for nonround pegs [5], [25].

In contrast to off-line planning of a motion plan as in the above examples, or the off-line design of the manipulator's compliance (as in this work), one might also consider an on-line learning of the forces that characterize errors and the proper motion response to them [14], [26].

A. Relation to this Work

Programmed compliance, the subject of this paper, differs from LMT, EDR, and replanning in that we assume we are

given a nominal motion plan, which is presumably a simple (one-step) one that takes no account of possible errors. We then design the compliance (here a damping matrix) so that the plan will succeed despite errors. In LMT, EDR, and replanning, it is the compliance that is given and the motion plan that is to be designed. Design of the compliance matrix for error-correction is the idea behind the development of the RCC wrist, and several others have addressed this idea as well (e.g., [10], [24]). The present work is, to our knowledge, the first attempt to describe a method of synthesis of an error-corrective compliance, given a description of a task. We do not claim to solve the problem but to pose it and to suggest one approach to it.

We should also note at this point that this paper is limited in scope to quasistatic assembly, i.e., to assembly speeds low enough that inertial effects of the robot and workpiece are small compared with frictional effects. Thus, we only create a single, dynamics-independent real-valued accommodation (inverse-damping) matrix in contrast with Hogan's work on impedance control [15].

In the opposite (high-speed) regime, the inertia matrix becomes the dominant factor in the evolution of an operation. There has been recent work [1] on achieving inertia matrices that are appropriate for some individual tasks, particularly high-speed peg-in-hole insertion.

B. Objective and Summary

The main points of this paper are as follows:

1) We observe that in many tasks, it is possible for any one of several qualitatively distinct contact configurations between grasped workpiece and environment to arise, depending on small initial errors. It is necessary to plan the compliance of a manipulator so that it is consistent with *every one* of the contact configurations because we do not know in advance which ones may arise. We seek a compliance matrix that is to be employed for the entire duration of an assembly task; we do not consider changing the compliance of the manipulator during a task. Our requirements on the compliance matrix are therefore quite severe; it must simultaneously consistent with each of several possible contact configurations.

2) We suggest that it is not enough for the compliance of a manipulator to be consistent with a contact task in the sense of the work of Mason and others (i.e., preventing the occurrence of unbounded forces). For many tasks, especially assembly tasks, we need the compliance to be error corrective as well. We ask that the compliance of the manipulator have the effect of converting the forces that arise from misalignments of the workpiece into corrective motions. In Section II, we formalize these two desired properties of the manipulator's compliance.

3) We develop a set of sufficient conditions for our desired properties, where all are of a single mathematical form well suited to synthesis. We demonstrate an optimization method that efficiently finds a matrix satisfying the sufficient conditions if one exists and, if not, finds one which comes close. In the latter case, it is possible to verify whether the original desired properties are, in fact, satisfied.

4) We demonstrate the synthesis of error-corrective compliance matrices for three example assembly tasks. We find

it encouraging that the result for the peg-into-chamfered-hole task is consistent with the RCC solution.

C. Notation

The following are examples of the notations that will be used in this paper:

t scalars
$\mathcal{A}$ matrices
F, V vectors
f unit vectors.

Our examples are planar. In these examples, vectors have three components (x, y, θ), where x is horizontal, y is vertical, and θ is rotation about the z axis (i.e., counterclockwise rotation in the plane of the paper.)

By "forces," we mean forces and torques—a six vector in a three-dimensional world or a three-vector in a planar world. Similarly, by "displacements," we mean displacements and rotations.

D. The Damper Model and the Accommodation Matrix

The function that maps the forces applied to a manipulator into the motion that ensues due to those forces may take many forms. For instance, the function may state that displacement of the manipulator from some nominal position X_o (which could be changing with time) is proportional to the applied force

$$X - X_0 = CF. \tag{1}$$

This is Hooke's law, which represents the behavior of an ideal spring, where C is the compliance matrix. An object held by a manipulator may have six degrees of freedom; three translational and three rotational. For such an object, (1) may be understood in vector form with X and X_o representing locations in six-dimensional configuration space, F a six-dimensional vector of forces and torques, and C a six-by-six matrix.

Equation (1) is often a good approximation for the natural compliance of mechanical systems composed of solids. We need not, however, restrict ourselves to natural compliance. If a manipulator is equipped with a force/torque sensor, we can perform arbitrary computation on those measurements and arrive at a motion, which the manipulator then executes. Thus, any compliance function may be achieved. This is called *active compliance*. If we implement the control law given in (1), we have simulated an ideal spring by using active compliance.

As a practical matter, the control law we implement must be evaluated rapidly; therefore, there is a premium on simple control laws. The one we choose in this paper is simple, has been studied extensively in the past, and is usually called the generalized damper:

$$V = V_o + \mathcal{A}F. \tag{2}$$

Control law (2) states that in the absence of any perceived forces F, the manipulator is to follow a nominal trajectory given by velocity V_o. In the presence of forces, the matrix $\mathcal{A}$ times the six-vector of forces F specifies a term $\mathcal{A}F$, which is simply added to the nominal velocity V_o to yield the velocity
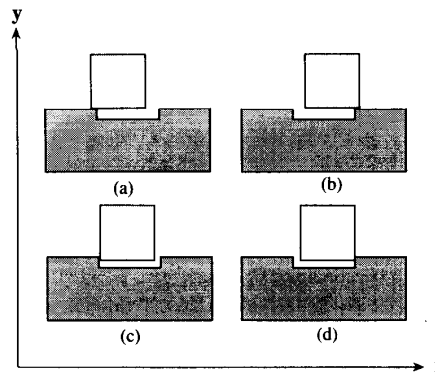

Fig. 2. Qualitatively distinct contact configurations, which may result during the block-into-detent assembly task. The nominal assembly trajectory V_o is $-y$ (down, not shown). Each contact configuration corresponds to a deviation of the movable part (the block) from its properly mated configuration. Contact configurations in which the edge of the block has come into contact with the left or right lip of the detent are shown in (a) and (b). In (c) and (d), the block is in contact with the left or right inside wall of the detent.

V, which is to be performed by the manipulator.[3] $\mathcal{A}$ is an *accommodation matrix*, which is the inverse of a damping matrix, and characterizes a generalized damper.

We have further reason to restrict ourselves to the design of an *accommodation matrix*, i.e., to a *linear* relationship of assembly force to consequent velocity. It may be possible to design a passive mechanical wrist that can assume any (or a large range of) accommodation matrices. The range of accommodation matrices that could be assumed by such a wrist is considered in [12].

E. Contact Configurations

Our approach to the creation of an error-corrective matrix for a given assembly operation proceeds from the identification of a set of contact configurations as shown in Fig. 2. In this example, the assembly operation being performed is, again, the placement of a block into a shallow detent in a fixed chassis. (Note that the RCC wrist would not be able to perform this operation because there are no chamfers). The nominal assembly trajectory of the manipulator V_o is straight down. In this paper, we do not address the selection of V_o.

We should note that specification of a complete set of contact configurations depends on more geometric information than is available in a CAD model of the parts alone. In particular, we must know how great the initial configurational errors are. As an extreme example, if large horizontal initial errors

---

[3] For example, to compare the motion of a manipulator under control laws (1) versus (2), suppose that the manipulator is moving steadily in the $-z$ direction, and we give it a push in the $+x$ direction. If C (or $\mathcal{A}$) is a diagonal matrix with positive elements, in either (1) or (2), the manipulator will respond by deflecting in the $+x$ direction. After we cease pushing, a manipulator following law (1) will spring back to its original path and proceed as if it had not been disturbed. A manipulator following law (2) will resume its original direction of motion $(-z)$ but will not move back to its original path; its displacement in the $+x$ direction is permanent.

With a diagonal matrix C (or $\mathcal{A}$), the motion of the manipulator under either (1) or (2) is intuitive; it moves in the direction in which it is pushed. However in selecting a matrix, we have considerable freedom, and highly nonintuitive behaviors may result. For instance, we can, by making the proper element of C (or $\mathcal{A}$) nonzero, cause forces in the $+x$ direction to map into motions in the $+y$ direction—or even into a rotation about the z axis if so desired.

are possible, it may happen that the block will miss the detent entirely. In the selection of contact configurations for Fig. 2, we have assumed that orientational errors are negligible and that translational errors are limited to about 2 mm.

Specification of the nominal trajectory V_o and the contact configurations is our starting point in this paper. We do not address the problem of extracting the contact configurations from geometric models, nor is it clear how difficult this task would be. Donald's error detection and recovery regions (EDR regions) [8], [9] and Desai's contact formations [7] contain much the same information as our contact configurations. Other related work includes [3], [4], and [11].

II. FORMAL PROPERTIES OF THE ACCOMMODATION MATRIX FOR ERROR-CORRECTIVE MANIPULATION

We pose the problem of creating an appropriate error-corrective accommodation matrix for a manipulator as that of realizing two desired properties of the manipulator's motion during an operation. We will express these in terms of the contact configurations.

Property 1: "Bounded forces"—For every contact configuration, $\mathcal{C}$ must be such that the contact forces that arise between mating parts are bounded.

Property 1 is necessary to prevent damage to the manipulator and/or workpiece. This is akin to the condition justified and studied by Mason [22]. Mason translated the bounded force condition into requirements on an accommodation matrix and showed how an accommodation matrix that guaranteed bounded forces for a given contact configuration could be found. Our work differs in that we now consider several distinct contact configurations for which the bounded forces condition must apply simultaneously.

For example, in Fig. 2(a), $\mathcal{C}F$ (the product of the accommodation matrix $\mathcal{C}$ with the forces created by contact) must cause the lower left corner of the block to have a velocity component in the surface-normal direction $+y$ (upward) to avoid unbounded forces.

Property 2: "Error-reduction"—For every contact configuration, $\mathcal{C}$ must be such that the magnitude of the displacement from proper mating E is reduced.

For example, in Fig. 2(a), $\mathcal{C}F$ must have a component in the $+x$ (rightward) direction to make progress toward proper mating.

For the moment we leave unspecified the metric $\mathbb{M}$ by which the magnitude of the displacement from proper mating E is to be measured. This is a nontrivial detail because the error vector E has both translational and orientational components (e.g., "centimeters" and "degrees"). The $\mathbb{L}^2$ metric (in which we square and add the components) has mixed units and makes no sense. See [20] for further discussion of suitable unit- and frame-invariant metrics.

Fortunately, if there is any metric $\mathbb{M}$ for which we can show that an accommodation matrix we synthesize results in decreasing $|E|$ for all contact configurations, we are assured that the operation must terminate at proper assembly. Our choice of metric is important: if we make a poor choice, we will not be able to synthesize an accommodation matrix $\mathcal{C}$ for which the distance to proper mating $|E|$ is guaranteed to

decrease. Yet, our choice of metric need not pass any formal test of nonarbitrariness.

III. A METHOD OF SYNTHESIS AND OPTIMIZATION

In this section, we describe a synthesis and optimization method. It is by no means the only one that might be used to create a matrix $\mathcal{C}$ satisfying the "bounded forces" and "error reduction" properties above, and some of its shortcomings will be discussed in Section V.

We will divide the "error reduction" property into two new conditions on $\mathcal{C}$ so that the new conditions and the "bounded forces" property will have a common mathematical form—one that is convenient for synthesis. We will show how $\mathcal{C}$ may be efficiently created so that it closely satisfies the whole set of conditions.

For each contact configuration, we determine the forces that we may expect the manipulator to perceive. It turns out that we only need the direction and not the magnitude of these characteristic forces. The direction is easily accessible to us from the familiar friction-cone construction. Given the direction of a characteristic force $\hat{f}_i$, we can express all of the conditions in the form $\hat{u}_i \cdot \mathcal{C}\hat{f}_i = t_i$, where $\hat{u}_i$ is a unit vector available to us from the geometry of the contact configurations (for instances $\hat{u}_i$ may be a surface normal), and t_i is a scalar.

Generally it turns out that the number m of conditions on $\mathcal{C}$

$$\hat{u}_i \cdot \mathcal{C}\hat{f}_i = t_i \qquad i = 1 \cdots m. \tag{3}$$

exceeds the number of elements[4] in the matrix $\mathcal{C}$. It is therefore not generally possible to find an accommodation matrix $\mathcal{C}$ that simultaneously satisfies all of the conditions exactly. All is not lost because the values t_i are not critical. The scalar t_i corresponds to an inverse stiffness. In the literature on hybrid control, for instance, there are "soft" axes and "stiff" axes, represented by stiffnesses k_{soft} and k_{stiff}, but the exact values of k_{soft} and k_{stiff} are not critical. Therefore, we expect that although typical values of the t_i can be given, if it is not possible to form a accommodation matrix with those precise values, one with slightly different values will also be acceptable.

We can use the freedom to adjust the inverse stiffnesses t_i to allow us to form an accommodation matrix $\mathcal{C}$ that almost satisfies the whole set of requirements, even when there is none that exactly satisfies them. We can then return to the "bounded forces" and "error reduction" properties to check that the matrix $\mathcal{C}$ that is synthesized does indeed satisfy them.

A. Predicting the Forces that Characterize a Contact Configuration

To design an accommodation matrix $\mathcal{C}$ that causes a desired motion in a particular situation we must be able to predict the contact force F exerted on the manipulator in that situation. It turns out that if we know the direction of the force F, we can calculate its magnitude, and this will be done immediately below. In the case of one-point contacts, we do know something about the direction of the force F, namely, that it must
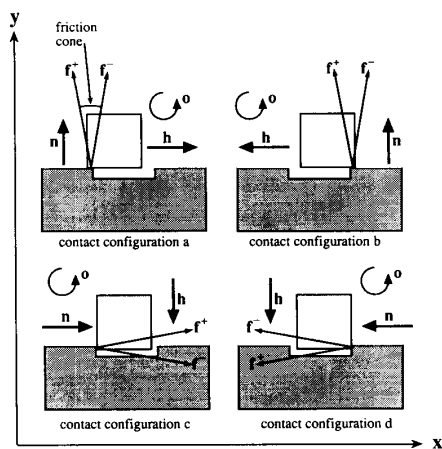
---

[4] Nine in two-space or 36 in three-space.

Fig. 3. Here, we have added to each of the qualitatively distinct contact configurations of Fig. 2 the surface normal n, home h, and orthogonal o vectors (there is only one orthogonal vector in planar examples). The friction cones, the extremes of which are the vectors $f^{\pm}$, are also shown.

lie within the friction cone as shown in Fig. 3. Further, if we know that the direction of motion of the block in Fig. 3 will be to the right ($+x$), we may predict that F will lie on the left extreme of the friction cone (f^{+}), as is shown. Contacts at more than one point are considered in Section III-A-3.

We could build our accommodation matrix α using the assumption that F will lie on the left extreme of the friction cone in Fig. 3(a). However, given the present state of the art in implementing force control, the assumption of a monotonic $+x$ motion, which will keep F on the left extreme, seems questionable at best. Instead we will assume only that F lies within the friction cone and ask that no matter where in the friction cone F lies, the "bounded forces" and "error reduction" properties are satisfied. Our conservative assumption about the direction of F contributes to a robustness of the resulting accommodation matrix α and, hopefully, relative insensitivity to the accuracy of a physical implementation of the control law $V = V_o + \alpha F$.

1) Expressing the "Bounded Forces" Property:
Property 1 requires that the motion of the manipulator must be such that the contact forces that arise between mating parts are bounded. To find the implications of this requirement for α, we evaluate the magnitude of the force F, which arises in a particular contact configuration.

First, we express F as a product of its magnitude and direction: $F = |F|\hat{f}$. Let $\hat{n}$ be the surface normal, as is shown in Fig. 3. Since the velocity of the block V can have no component in the direction $\hat{n}$, we have $\hat{n} \cdot V = 0$. (Some care must be taken in performing dot products among vectors of mixed distances and angles [2], [16]).

Combining the above equations with our control law $V = V_o + \alpha F$, we can solve for the magnitude of the force

$$|F| = -\frac{\hat{n} \cdot V_o}{\hat{n} \cdot \alpha f}. \tag{4}$$

The scalar $(\hat{n} \cdot \alpha f)^{-1}$ represents the stiffness of the manipulator to interactions with a particular surface. To guarantee that unbounded contact forces do not arise due to velocities that

bring the parts together, no matter what the sources of these velocities, it is only necessary to require a finite (noninfinite) stiffness. **Property 1** becomes the following:

Property 1: "Bounded forces" $\Rightarrow n \cdot \alpha f \approx t_1$. (5)

A reasonable value for the scalar stiffness $1/t_1$ depends on the task being performed, the materials involved, the nominal velocity V_o chosen, and other factors. We will presume that although a user can select a typical value for t_1, he or she will not feel very strongly about the exact value. For the rest of this paper, it is important only to notice that t_1 is positive, and its value is user selectable.

Our method for creating α (Section III-B) will take advantage of the freedom to adjust t_1. **Property 2** will result in a similar parameter (t_2) being introduced.

2) Expressing the "Error Reduction" Property:
Property 2 requires that the motion of the manipulator reduce the distance from the manipulator's present configuration to a properly mated configuration as measured by some metric M. We will use as our metric M a modified L^2 metric in which the angular components (in radians) are multiplied by a characteristic dimension of the assembly task (e.g., the width of the grasped object), and then, the components are squared and summed.[5] This metric is physically justifiable though still arbitrary. Again, we are content with a matrix α if it satisfies **Property 2** for any metric; therefore, we may indeed choose M arbitrarily.

To express **Property 2** in a form similar to (5) above, let us now identify a "home" unit vector $\hat{h}$ for each contact configuration. We choose $\hat{h}$ to be the direction consistent with sliding constraints in which the distance from proper mating is most steeply decreasing as measured by our metric M.

With $\hat{h}$ and $\hat{n}$ identified, there are several[6] other vectors o_i orthogonal to both $\hat{h}$ and $\hat{n}$. Fig. 3 shows examples of $\hat{h}, \hat{n}$, and $\hat{o}$. We can now write sufficient conditions for **Property 2** (error reduction) as

$$\hat{h} \cdot V > 0 \tag{7a}$$

and

$$\hat{o}_i \cdot V = 0 \ \forall i. \tag{7b}$$

These conditions ask that there be motion in the "home" direction (7a) and that there not be motion in orthogonal directions (7b). The required velocity at which errors are reduced in (7a) is not critical, but for definiteness, let us request that $\hat{h} \cdot V \approx |V_o|$, i.e., we wish the manipulator to move toward proper mating at approximately the full nominal velocity.

Using the control law $V = V_o + \alpha F$, our requirement $\hat{h} \cdot V \approx |V_o|$, and $\hat{n} \cdot V = 0$, we find

$$\hat{h} \cdot \alpha f = (\hat{h} \cdot V_o - |V_o|)\frac{\hat{n} \cdot \alpha f}{\hat{n} \cdot V_o}. \tag{8}$$

---

[5] $M(x, y, z, \theta_x, \theta_y, \theta_z) = (x^2 + y^2 + z^2) + w^2(\theta_x^2 + \theta_y^2 + \theta_z^2)$, where w is a characteristic length of the task.

[6] One in 2D, or four in 3D.

Note that $\hat{n} \cdot V_o$ is negative.[7] V_o is our nominal assembly velocity, and we have already required that $\hat{n} \cdot \mathbf{G}f \approx t_1$ for some user-selectable parameter t_1. The right side of (8) can be evaluated, and we will denote it by t_2, which is another positive constant. We may now write **Property 2** as the following:

Property 2: "Error reduction" $\Rightarrow$

$$\hat{h} \cdot \mathbf{G}f = t_2, \text{ and} \qquad (9a)$$

$$\hat{o}_i \cdot \mathbf{G}f = 0 \ \forall i. \qquad (9b)$$

For example, consider contact configuration a in Fig. 3. The nominal direction $V_o = -y$, the surface normal $\hat{n} = y$, and the home direction $\hat{h} = +x$. The "bounded forces" property is thus $\hat{y} \cdot \mathbf{G}f = t_1$. In place of the "error-reduction" property, condition (9a) gives us $\hat{x} \cdot \mathbf{G}f = t_2$. Condition (9b) applies only to the one remaining degree of freedom θ and is $\hat{\theta} \cdot \mathbf{G}f = 0$, i.e., $\mathbf{G}f$ should have no rotational component.

3) Extension to Contact Configurations with More than One Point of Contact: If there are two or more points of contact between workpiece and environment, the total force felt by the manipulator is the sum of the forces at the individual contacts. For simplicity, we will work here with only two contacts; the analysis extends trivially to any number. Denote these individual forces F_1, F_2. The motion of the manipulator is still given by the control law $V = V_o + \mathbf{G}F$, which can now be written $V = V_o + \mathbf{G}F_1 + \mathbf{G}F_2$. We can break each force into a product of its magnitude with its direction $F_i = |F_i|\hat{f}_i$. We also have one condition of the form $V \cdot \hat{n}_i = 0$ for each contact. We can therefore construct a set of equations from which the magnitude of the forces can be extracted:

$$\hat{n}_1 \cdot V_o + |F_1|\hat{n}_1 \cdot \mathbf{G}f_1 + |F_2|\hat{n}_1 \cdot \mathbf{G}f_2 = 0$$

$$\hat{n}_2 \cdot V_o + |F_1|\hat{n}_2 \cdot \mathbf{G}f_1 + |F_2|\hat{n}_2 \cdot \mathbf{G}f_2 = 0 \qquad (10a)$$

which can be written as

$$\begin{pmatrix} \hat{n}_1 \cdot \mathbf{G}f_1 & \hat{n}_1 \cdot \mathbf{G}f_2 \\ \hat{n}_2 \cdot \mathbf{G}f_1 & \hat{n}_2 \cdot \mathbf{G}f_2 \end{pmatrix} \begin{pmatrix} |F_1| \\ |F_2| \end{pmatrix} = - \begin{pmatrix} \hat{n}_1 \cdot V_o \\ \hat{n}_2 \cdot V_o \end{pmatrix}. \qquad (10b)$$

As in Section III-A-1, we may identify the determinant of the two-by-two matrix as the inverse stiffness of the manipulator. **Property 1** (bounded forces) therefore becomes

$$\det \begin{pmatrix} \hat{n}_1 \cdot \mathbf{G}f_1 & \hat{n}_1 \cdot \mathbf{G}f_2 \\ \hat{n}_2 \cdot \mathbf{G}f_1 & \hat{n}_2 \cdot \mathbf{G}f_2 \end{pmatrix} \approx t_1^2. \qquad (11a)$$

For the general case of m contacts, where $\{\hat{n}\}$ represents the m surface normals and $\{\hat{f}\}$ represents the m force directions, we may write this condition compactly (if opaquely) as an outer product

$$\det(\{\hat{n}\} \otimes \mathbf{G}\{\hat{f}\}) \approx t_1^m. \qquad (11b)$$

It is useful to notice that if a contact configuration with more than one point of contact can occur, it is also possible for

[7] If $\hat{n} \cdot V_o \approx 0$, it will be best to impose no requirement on $\hat{h} \cdot \mathbf{G}f$. In this case, if $\hat{h} \cdot V_o > 0$, the nominal trajectory V_o itself causes reduction of error, and no adjustment of $\mathbf{G}$ will increase or decrease the rate much. If $\hat{h} \cdot V_o < 0$, the situation is hopeless.

the points of contact to occur individually as contact configurations. The "bounded forces" property applied to these individual contact configurations involve the terms $\hat{n}_i \cdot \mathbf{G}f_i$ that appear on the diagonal in (11a). Since we have already required that $\hat{n}_i \cdot \mathbf{G}f_i \approx t_1$, one way to satisfy (11a) is simply to require that $\hat{n}_i \cdot \mathbf{G}f_j \approx 0$ when $i \neq j$, i.e., that all the off-diagonal terms be zero. This observation will be useful in the following section because it allows us to reduce all of the properties on all of the contact configurations to a single mathematical form.

B. Forming an Accommodation Matrix

All of our conditions on the accommodation matrix $\mathbf{G}$ have now been reduced to numerous conditions of the form

$$\hat{u}_i \cdot \mathbf{G}f_i = t_i \qquad (14)$$

where $\hat{f}_i$ is a force to be expected, $\hat{u}_i$ is a direction vector determined from the geometry of the contact configuration, and t_i is a target value for the dot product. For some conditions, the target value t is zero. For others, t is a user-selectable positive value whose choice depends on the task at hand, materials, etc., and indeed is somewhat arbitrary. Selection of these t_i is guided by (4) and (8).

Recall that f in equations of the form (14) represents any force direction that could arise in a particular contact configuration. f must necessarily lie within the friction cone, but we are reluctant to presume it to be any better localized than that. Let the two extremes of the friction cone be denoted $f^{\pm}$. Then, any force f that arises can be written as a positive linear combination of f^+ and f^-. It will not be possible to design an accommodation matrix $\mathbf{G}$ for which $\hat{u} \cdot \mathbf{G}f^+ = \hat{u} \cdot \mathbf{G}f^- = t$. However, our target value t was not critical, and perhaps, we can design an accommodation matrix $\mathbf{G}$ for which $\hat{u} \cdot \mathbf{G}f^+ = t^+, \hat{u} \cdot \mathbf{G}f^- = t^-$, with both t^+ and t^- acceptably close to the target t. If so, we are assured that the value of $\hat{u} \cdot \mathbf{G}f$ will be acceptably close to t for any f that may arise.

1) Least-Squares Optimization: For each of several contact configurations, we consider two forces $f^{\pm}$, and for each force, we have several conditions of the form $\hat{u}_i \cdot \mathbf{G}\hat{f}_i = t_i$. Let m be the total number of such equations—clearly, m may be large. The $\hat{u}_i, \hat{f}_i$, and t_i are all given; it is $\mathbf{G}$ we wish to create. $\mathbf{G}$ is a matrix consisting of n real numbers. In our two-dimensional examples $n = 9$; in three dimensions, we would have $n = 36$. The usual case is that there is no $\mathbf{G}$, which satisfies all m equations.

We next described a way of finding the best $\mathbf{G}$ in the "least-squares" sense; the $\mathbf{G}$ that minimizes $\sum_{i=1}^{m}(\hat{u}_i \cdot \mathbf{G}\hat{f}_i - t_i)^2$. Solving for this $\mathbf{G}$ is computationally quick even for $m = 200$, $n = 36$.

How are we to interpret the resulting $\mathbf{G}$? A satisfactory $\mathbf{G}$ is one for which the "bounded forces" and "error reduction" properties hold; therefore, that motion can only terminate at proper mating. Our m conditions of the form $\hat{u}_i \cdot \mathbf{G}\hat{f}_i = t_i$ are stronger (sufficient) conditions; therefore, if an $\mathbf{G}$ satisfies them all exactly, that $\mathbf{G}$ is satisfactory.

Further, of the $\{t_i\}$, some were zero (see (9b)), and others

were arbitrary positive values (see (5) or (9a)). If the $\mathbf{a}$ we create satisfies the (9b)-type conditions and gets any positive values for the (5) and (9a)-type conditions, that $\mathbf{a}$ is also satisfactory but for a different choice of the arbitrary stiffnesses. Even if the $\mathbf{a}$ we create does not satisfy the (9b)-type conditions, it may still be satisfactory, but then, we must check the "bounded forces" and "error reduction" properties explicitly.

We now abandon the matrix structure and meaning of $\mathbf{a}$ and simply consider it to be a set of n real-valued variables (here $n = 9$) for which we desire a solution satisfying m equations. To maintain a compact notation, we will form these n variables into a n-vector $\mathbf{a}^{so}$, which, here, is a 9 vector. $\mathbf{a}$ is "strung-out" by rows and displayed as a column vector. Explicitly

$$\mathbf{a} = \begin{pmatrix} \mathbf{a}_{xx} & \mathbf{a}_{xy} & \mathbf{a}_{x\theta} \\ \mathbf{a}_{yx} & \mathbf{a}_{yy} & \mathbf{a}_{y\theta} \\ \mathbf{a}_{\theta x} & \mathbf{a}_{\theta y} & \mathbf{a}_{\theta\theta} \end{pmatrix} \Rightarrow \mathbf{a}^{so} = \begin{pmatrix} \mathbf{a}_{xx} \\ \mathbf{a}_{xy} \\ \mathbf{a}_{x\theta} \\ \mathbf{a}_{yx} \\ \mathbf{a}_{yy} \\ \mathbf{a}_{y\theta} \\ \mathbf{a}_{\theta x} \\ \mathbf{a}_{\theta y} \\ \mathbf{a}_{\theta\theta} \end{pmatrix}. \quad (17)$$

Having formed this unintuitive construction, an equation of the form $\hat{u} \cdot \mathbf{a} f = t$, which is explicitly

$$(u_x, \quad u_y, \quad u_\theta) \cdot \begin{pmatrix} \mathbf{a}_{xx} & \mathbf{a}_{xy} & \mathbf{a}_{x\theta} \\ \mathbf{a}_{yx} & \mathbf{a}_{yy} & \mathbf{a}_{y\theta} \\ \mathbf{a}_{\theta x} & \mathbf{a}_{\theta y} & \mathbf{a}_{\theta\theta} \end{pmatrix} \begin{pmatrix} f_x \\ f_y \\ f_\theta \end{pmatrix} = t \quad (19)$$

can be written as

$$(u_x f_x, \quad u_x f_y, \quad u_x f_\theta, \quad u_y f_x, \quad u_y f_y, \quad u_y f_\theta, \quad u_\theta f_x,$$
$$u_\theta f_y, \quad u_\theta f_\theta) \cdot \mathbf{a}^{so} = t. \quad (20)$$

Each of the m equations of form $\hat{u} \cdot \mathbf{a} f = t$ can be written as above for $i = 1 \cdots m$ so that we have an enormous matrix equation

$$\begin{pmatrix} u_x^1 f_x^1 & u_x^1 f_y^1 & u_x^1 f_\theta^1 & u_y^1 f_x^1 & u_y^1 f_y^1 & u_y^1 f_\theta^1 & u_\theta^1 f_x^1 & u_\theta^1 f_y^1 & u_\theta^1 f_\theta^1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ u_x^m f_x^m & u_x^m f_y^m & u_x^m f_\theta^m & u_y^m f_x^m & u_y^m f_y^m & u_y^m f_\theta^m & u_\theta^m f_x^m & u_\theta^m f_y^m & u_\theta^m f_\theta^m \end{pmatrix} \cdot \mathbf{a}^{so} = \begin{pmatrix} t^1 \\ \cdot \\ \cdot \\ t^m \end{pmatrix}. \quad (21)$$

We will name the great m row by n column matrix on the left G and the m-high column vector of target values t. In these terms, the problem of finding a accommodation matrix $\mathbf{a}$, which satisfies our m conditions, is reduced to finding a n-vector $\mathbf{a}^{so}$ that satisfies

$$\mathsf{G} \cdot \mathbf{a}^{so} = t \quad (22)$$

Generally, there is no $\mathbf{a}^{so}$ that satisfies (22) exactly. However it is easy to check that

$$\|\mathsf{G} \cdot \mathbf{a}^{so} - t\|^2 = \sum_{i=1}^{m} (\hat{u}_i \cdot \mathbf{a} f_i - t_i)^2. \quad (23)$$

The right side is exactly the sum we wish to minimize—the one that defines the "best" $\mathbf{a}$. There is a good and fast way of finding the $\mathbf{a}^{so}$ that minimizes the left side. Using the singular value decomposition (described particularly clearly in [23]), we can form an n-by-m pseudo-inverse[8] G^* of the m-by-n matrix G for which the value of $\mathbf{a}^{so}$ given by

$$\mathbf{a}^{so} = \mathsf{G}^* t \quad (25)$$

minimizes $\|\mathsf{G}\mathbf{a}^{so} - t\|^2$ over all possible $\mathbf{a}^{so}$. If (fortuitously) there is a $\mathbf{a}^{so}$ that satisfies (22) exactly, then (25) finds it. In this case, the sum of squares is zero.

G contains all the geometry of the contact configurations and forces that arise in assembly. t contains our preferences for the user-selectable parameters t_i. If we pick a target vector of parameters t, we can easily compute the accommodation matrix $\mathbf{a}$, which comes closest to achieving those parameters using $\mathbf{a}^{so} = \mathsf{G}^* t$. Usually, we cannot have that set of parameters; the closest available set is $t^\sim = \mathsf{G}\mathbf{a}^{so} = \mathsf{G}\mathsf{G}^* t$.

To see how well $\mathbf{a}$ achieves our preferred parameters, we compare t to $t^\sim$. Once the m-by-m matrix $\mathsf{G}\mathsf{G}^*$ has been formed, we can try out different target vectors t with little overhead.

As mentioned above, if $t^\sim$ has zeros in the same components that t has zeros and is positive in all other components, then $\mathbf{a}$ possesses the "bounded forces" and "error reduction" properties. The only consequence of not satisfying all of the conditions exactly is that the stiffnesses are not exactly as requested.

If $t^\sim$ does not match t in sign, component by component, then to verify that $\mathbf{a}$ possesses the "bounded forces" and "error reduction" properties, we must compute $V = V_o + \mathbf{a}F$ for every contact configuration and explicitly check that the properties are satisfied.

IV. EXAMPLES

In this section, we set up (by hand) the contact configurations and the normal, home, and orthogonal vectors for some sample planar assembly tasks. To create the accommo-

dation matrix $\mathbf{a}$ that achieves the desired "bounded forces" and "error reduction" properties we proposed in Section III, we use the stronger conditions and the optimization procedure described in Section III. When even the stronger conditions

[8] There are other pseudo inverses. Computation of the SVD pseudoinverse is numerically robust, and the SVD pseudoinverse possesses the needed least-squares property.

can be satisfied (or satisfied for a slightly different set of stiffnesses), we are done. This occurs in the first two examples. In the third example, we must verify that the desired properties are in fact satisfied for all contact configurations.

A. Place Block into Chamferless Detent

Here, a block is to be placed into a shallow detent in a surface (Fig. 1). We wish to create a accommodation matrix so that the torque, which arises if the edge of the block hits the lip of the detent, is mapped by the accommodation matrix into a horizontal motion of the block towards the center of the detent. This motion is counterintuitive since the detent has no chamfers, but it is not impossible.

It is necessary to choose a coordinate system: an origin about which torques are measured and rotations are performed. The choice of origin is of absolutely no account; in fact, if we create an accommodation matrix for origin q, it is trivial to convert it to an accommodation matrix for origin p simply by

$$
\mathbf{G}_p = \begin{pmatrix} 1 & 0 & q_y - p_y \\ 0 & 1 & p_x - q_x \\ 0 & 0 & 1 \end{pmatrix} \mathbf{G}_q \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ q_y - p_y & p_x - q_x & 1 \end{pmatrix}.
$$

$$(26)$$

For a given force in the real world, the accommodation matrix $\mathbf{G}_p$ at origin p produces exactly the same motion of the manipulator as would accommodation matrix $\mathbf{G}_q$ at origin q. Inconsequential as it is, we must choose an origin; therefore, we choose it at q, as is shown in Fig. 1.

The four possible contact configurations as shown in Fig. 3. It should be noted that implicit in the specification of only these contact configurations is information about the maximum possible magnitudes of the robot's initial error. If, for instance, the possible horizontal translational error were much greater, the block could miss the detent altogether. We presume, not unrealistically, that errors of orientation are minimal.

For each of the four contact configurations, there are two extremes of the friction cone. Translating the "bounded forces" and "error reduction" properties into three stronger conditions on each of the eight force-and-configuration pairs, we find a total of 24 conditions on the accommodation matrix $\mathbf{G}$ to be synthesized, and each is of the form $\hat{u}_i \cdot \mathbf{G} f_i = t_i$.

For the user-selectable target inverse stiffnesses t_i, we have used the value 1 or 0 as appropriate from (5), (9a), or (9b). We find that it is not possible to satisfy all the conditions exactly. The accommodation matrix that most closely fulfills the conditions is

$$
\mathbf{G}_q = \begin{pmatrix} 2.91 & 0.00 & -1.94 \\ 0.00 & 1.03 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{pmatrix}.
$$

$$(27)$$

Evaluating $t^\sim = \mathbf{GG}^* t$, which is the avaialble set of inverse stiffnesses t_i nearest to our target set, we find all values within 30% of target. This is surely good enough—it would be very

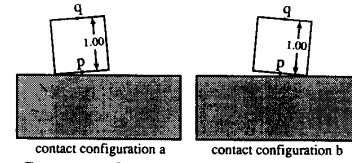contact configuration a   contact configuration b

Fig. 4.   Contact configurations for the put-block-down-gently task.

strange if stiffnesses more precise than this would ever be needed. The (9b)-type conditions $(u_i \cdot \mathbf{G} f_i = 0)$ are satisfied exactly.

For the task, the accommodation matrix that is synthesized cannot be described as a center of accommodation, no matter what choice of origin we use. The result is relatively insensitive to the coefficient of friction assumed, which we took as 0.25.

B. Put a Block Down Gently

Here, a block is simply to placed on a surface. There are only two contact configurations, which are shown in Fig. 4; either corner may touch first, and we would like the behavior of the manipulator to be such that upon contact, the block rotates into alignment with the surface without scraping along the surface.

Property 1 requires bounded forces. **Property 2** requires that the block rotate clockwise or counterclockwise as appropriate. The (9b)-type conditions $(\hat{u}_i \cdot \mathbf{G} f_i = 0)$ require that there be no horizontal movement of the corner that makes contact first: no "skidding." Two contact configurations, by two extremes of the friction cone, by three conditions each, yield 12 conditions of the form $\hat{u}_i \cdot \mathbf{G} f_i = t_i$. Using the origin q as shown in Fig. 4 (at the top center of the block), and again using 1 or 0 as target values for all the user-selectable inverse-stiffnesses, we find that there is an accommodation matrix, which satisfies all the conditions exactly:

$$
\mathbf{G}_q = \begin{pmatrix} 2.06 & 0.00 & -2.06 \\ 0.00 & 0.52 & 0.00 \\ -2.06 & 0.00 & 2.06 \end{pmatrix}.
$$

$$(28)$$

It can easily be verified from (26) that this accommodation matrix is equivalent to a diagonal matrix at the bottom center of the block, meaning that our synthesis has suggested that in this case, a "center of compliance" description is appropriate—with center of accommodation at the bottom center of the block (point p).

The equivalent accommodation matrix, at p, is

$$
\mathbf{G}_p = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.52 & 0.00 \\ 0.00 & 0.00 & 2.06 \end{pmatrix}.
$$

$$(29)$$

It is relatively easy to imagine the operation of an accommodation matrix that is diagonal, and the reader may check that a center of accommodation at p will give the proper response to contact at either corner.

C. Peg into Chamfered Hole Revisited

Here, we find ten contact configurations, some of which are two-point contacts. Five of these are shown in Fig. 5, and the other five are reflections about a vertical axis.

It turns out that there is no accommodation matrix $\mathcal{C}$ that simultaneously gives the desired stiffness in every contact configuration. In fact, there is no way to satisfy all the (9b)-type conditions $\hat{u}_i \cdot \mathcal{C}f_i = 0$ even if we allow any positive value for t_i in the (5) and (9a)-type conditions $\hat{u}_i \cdot \mathcal{C}f_i = t_i$.

The accommodation matrix that comes closest in the least-squares sense is

$$\mathcal{C}_q = \begin{pmatrix} 0.66 & 0.00 & 0.19 \\ 0.00 & 1.57 & 0.00 \\ 0.20 & 0.00 & 0.35 \end{pmatrix} \tag{30}$$

based on an origin at q as shown in Fig. 5.

Interesting, we find that $\mathcal{C}_q$ is in agreement with the results in [27], which are incorporated in the RCC device. Specifically, if (following [27]) we let ℓ_2 be the insertion depth at which two-point (jamming) contact may first occur, the accommodation matrix $\mathcal{C}$, which our optimization produces, can be described as a center of accommodation at a point p and a distance $\ell_2/2$ above the tip of the peg on axis. Rewriting $\mathcal{C}$ for origin at point p, we have

$$\mathcal{C}_p = \begin{pmatrix} 0.55 & 0.00 & 0.01 \\ 0.00 & 1.57 & 0.00 \\ 0.02 & 0.00 & 0.35 \end{pmatrix} \tag{31}$$

which is (almost) diagonal.

V. DISCUSSION

In our view, the main contribution of this paper is to pose the problem of synthesizing an error-corrective compliance for an entire task. To the best of our knowledge, this is the first attempt at a method of synthesis.

The underlying idea (common to other works in this field) is that the forces that arise during an assembly task often contain enough information to figure out how the operation has gone wrong. Forces characterize the errors in assembly tasks in a more direct way than other sensory information (e.g., vision). The question then becomes how best to interpret and act on this force information. Other researchers have (implicitly or explicitly) interpreted this information as well, and a summary of those ideas is given in Section I.

Our approach may be distinguished from Mason's in that Mason synthesizes a accommodation matrix that is consistent with a task, whereas we synthesize one that is also error-corrective (terms defined in Section I-B). It may be conceptually contrasted with most other fine motion planning work in that we assume the nominal motion plan to be given and synthesize the accommodation matrix of the manipulator so that assembly proceeds to completion. Most other workers have assumed the compliance of the manipulator to be given and have synthesized a motion plan. We draw inspiration from
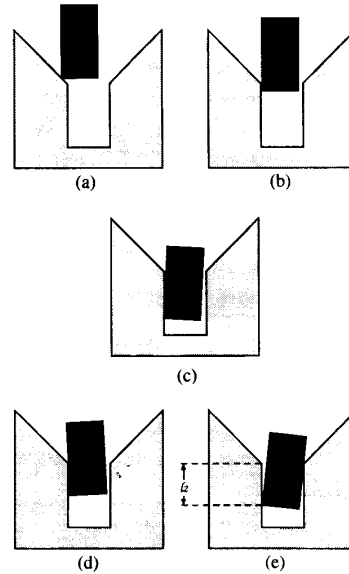

Fig. 5. Contact configurations for the peg-into-chamfered-hole task. We consider these contact configurations that are shown and five, which are not shown, are these reflected about a vertical axis. The nominal assembly trajectory V_o is straight down ("south"). One of the configurations (e) involves two points of contact; the rest are single-point contact configurations. To avoid confusion, the normal $\hat{n}$, home $\hat{h}$, and orthogonal $\hat{o}_i$ directions are not shown in the figure. In configuration (a), $\hat{n}$ is northeast, $\hat{h}$ is southeast, and $\hat{o}$ is rotation. In configurations (b), (c), and (d), $\hat{n}$ is east, and $\hat{o}$ is rotation. The $\hat{h}$ direction is not needed because south is colinear with V_o. In configuration (e) $\hat{h}$ is counterclockwise rotation.

error-corrective compliances that have been created by others for a few particular tasks, most notably the RCC. Our work differs from these in that we attempt a method of synthesis and not a solution of an individual task.

This work has several major limitations:

1) We have assumed that a finite and complete set of qualitatively distinct contact configurations has been (and can be) generated and presented to us. These configurations, together with a nominal assembly trajectory, which would succeed in the absence of error, are the starting point of our synthesis. Although finding these configurations does not seem to be difficult to do by hand for simple examples, we do not know of a formal definition of "qualitatively distinct" or of a method of generating such configurations.

2) Our "error-reduction" property requires strictly decreasing errors for all contact configurations as measured by some metric. We have chosen a particular metric and therefore might not recognize as valid an accommodation matrix that decreases errors according to some other metric.[9] Further, one might imagine compliances that guarantee eventual success but do not achieve it monotonically by any metric.

3) The optimization method described in Section III does not directly use even the one particular metric we have chosen (and by which we verify the resulting $\mathcal{C}$ matrix). Rather, it optimizes adherence to a set of stronger conditions, which, if they are all satisfied would also guarantee strictly decreasing errors. These stronger conditions are not always satisfied

---

[9] Much work in C space seems to be dogged by the current lack of sensible metrics.

so we do not necessarily find an accommodation matrix that guarantees strictly decreasing errors by our metric even when one exists—we are not quite optimizing the right thing.

Though our examples were two dimensional, most of the development applies immediately to three dimensions. An exception is our bounding of the friction cone by two vectors (Section III-B). It is possible to generalize the friction cone to three dimensions [11], [13]. It might also be that providing a set of qualitatively distinct contact configurations (the input to the method of synthesis described here) may be much more difficult in three dimensions than in two.

ACKNOWLEDGMENT

REFERENCES

[1] H. Asada and Y. Kakumoto, "The dynamic RCC hand for high-speed assembly," in Proc. 1988 IEEE Int. Conf. Robotics Automat. (Philadelphia) 1988, IEEE Press.

[2] R. S. Ball, A Treatise on the Theory of Screws. Cambridge, UK: Cambridge University Press 1900.

[3] R. A. Brooks, "Symbolic error analysis and robot planning," Int. J. Robotics Res., vol. 1, no. 4, 1982.

[4] S. J. Buckley, "Planning and teaching compliant motion strategies," Ph.D. thesis, Massachusetts Inst. Technol., Cambridge, 1987.

[5] M. E. Caine, "Chamferless assembly of rectangular parts in two and three dimensions," M.A. thesis, Dept. Mech. Eng., Massachusetts Inst. Tech., Cambridge, 1985.

[6] J. De Schutter and H. Van Brussel, "A formalism for specifying compliant motion tasks," Int. J. Robotics Res., vol. 7, no. 4, 1988.

[7] R. S. Desai, "On Fine Motion in Mechanical Assembly in Presence of Uncertainty," Ph.D. thesis, Univ. Michigan, Ann Arbor, 1988.

[8] B. R. Donald, "Robot motion planning with uncertainty in the geometric models of the robot and environment: A formal framework for error detection and recovery," in Proc. 1986 IEEE Int. Conf. Robotics Automat. (San Francisco), 1986, IEEE Press.

[9] B. R. Donald, "A geometric approach to error detection and recovery for robot motion planning with uncertainty," Artificial Intell., 1988.

[10] S. H. Drake, "The use of compliance in a robot assembly system," in Proc. IFAC Symp. Inform. Contr. Problems Manufact. Technol. (Tokyo), 1977.

[11] M. A. Erdmann, "On motion planning with uncertainty," M.S. thesis, Massachusetts Inst. Technology, Cambridge, 1984.

[12] A. Goswami, M. Peshkin, and J. E. Colgate, "Passive robotics: An exploration of mechanical computation," in Proc. 1990 IEEE Conf. Robotics Automat. (Cincinnatti), 1990, IEEE Press.

[13] S. Goyal and A. Ruina, "Planar sliding with dry friction: Limit surface and moment function," to be published.

[14] G. Hirzinger and K. Landzettel, "Sensory feedback structures for robots with supervised learning," in Proc. 1985 Int. Conf. Robotics Automat. (St. Louis), pp. 627–631, IEEE Computer Society Press.

[15] N. Hogan, "Impedance control: An approach to manipulation," ASME J. Dynamic Syst. Meas. Contr., vol. 107, pp. 1–24, 1985.

[16] K. H. Hunt, Kinematic Geometry of Mechanisms. Oxford: Clarendon, 1978.

[17] H. Inoue, "Force feedback in precise assembly tasks," in Artificial Intelligence: An MIT Perspective (Brown, Ed.). Cambridge, MA: MIT Press, 1979.

[18] J. C. laTombe, Preprint, 1989.

[19] C. S. G. Lee and E. S. H. Hou, "Automatic generation and synthesis of c-frames for mechanical parts in an insertion task," IEEE J. Robotics Automat., vol. 4, no. 3, p. 287, 1988.

[20] H. Lipkin, "Invariant properties of the pseudo-inverse in robotics," in Proc. NSF Design Manufact. Syst. Conf. (Tempe, AZ) 1990.

[21] T. Lozano-Perez, M. T. Mason, and R. H. Taylor, "Automatic synthesis of fine-motion strategies for robots," Int. J. Robotics Res., vol. 3, no. 1, pp. 3–24, 1984.

[22] M. M. Mason, "Compliance and force control for computer controlled manipulators," IEEE Trans. Syst. Man Cybern., vol. SMC-11, no. 6, 1981.

[23] W. Press, B. Flannery, S. Teukolsky, and W. Vetterling, Numerical Recipes, The Art of Scientific Computing. Cambridge, UK: Cambridge University Press, 1985.

[24] S. Simunovic, "Force information in assembly processes," in Proc. Fifth Int. Symp. Ind. Robotics (Chicago Soc. Manufact. Eng.), 1975.

[25] D. R. Strip, "Insertions using geometric analysis and hybrid force-position control: Method and analysis," in Proc. 1988 IEEE Int. Conf. Robotics Automat. (Philadelphia), pp. 1744–1751, 1988, IEEE Comput. Soc. Press.

[26] E. Vaaler and W. Seering, "Automated assembly with systems having significant manipulator and parts location errors," in Proc. 1987 IEEE Int. Conf. Robotics Automat. (Raleigh, NC), pp. 1357–1360, 1987, IEEE Comput. Soc. Press.

[27] D. E. Whitney, "Quasi-static assembly of compliantly supported rigid parts," ASME J. Dynamic Syst. Meas. Contr., vol. 104, pp. 65–77, 1982.

[28] J. Xiao and R. A. Volz, "A replanning approach towards uncertainty handling in robot assembly," in Proc. 3rd IEEE 1988 Int. Symp. Intell. Contr. (Arlington, VA), 1988.

Michael A. Peshkin (M'86) received the Bachelor's degree from the University of Chicago, IL, the Master's degree from Cornell University, Ithaca, NY and the Ph.D. degree from Carnegie-Mellon University, Pittsburgh, PA, all in physics.

At Carnegie-Mellon, he worked in CMU's Robotics Institute. He is presently an Assistant Professor of Mechanical Engineering at Northwestern University, Evanston, IL. His current research interests in robotics are task planning, force control, programmable passive devices, and electroheological fluids. Other interests include neural networks and gain growth phenomena.

Dr. Peshkin is a 1988 NSF Presidential Young Investigator. He is the author of 19 technical publications and a recent book Planning Robotic Manipulation Strategies (Prentice-Hall, 1990). He is a member of ASME.

PROGRAMMED COMPLIANCE FOR ERROR-CORRECTIVE ASSEMBLY

Michael A. Peshkin[1]
Mechanical Engineering Dept.
Northwestern University
Evanston, IL 60208

ABSTRACT:

Suppose that before an assembly task commences we can specify at will the manipulator's response to assembly forces, by providing a single compliance (or damping) matrix to be used for the duration of the operation. Can we choose the matrix elements so that the force which characterizes every possible error condition maps into a motion which reduces it? If so we are assured that the operation will evolve toward decreased errors and eventual success.   In this paper we describe a framework and a method of synthesis of an error-corrective matrix.

1.1 BACKGROUND AND MOTIVATION

It is widely recognized that for robots to perform tasks involving the contact of rigid objects some form of compliance[2], either of the robot or of the objects, is required. Under position-control and in the absence of such compliance, the presence of even the slightest error in the positions of the objects can produce unbounded contact forces and therefore failure of the operation.

Depending upon the task being performed and the geometry of the objects in contact, different forms of compliance are needed -- compliance should be specific to the task. Mason [Mason, 1981 #37] formalizes this idea, traces its origins in the literature, and describes a method of creating a compliance matrix *consistent* with a given task. The term *consistent* is introduced here in order to contrast it with *error-corrective,* below.   By *consistent* we mean that the compliance of the manipulator is such that unbounded forces do not develop. Roughly speaking, this requires that the manipulator's stiff (position controlled) axes not be perpendicular to rigid surfaces in the environment. Recent work on the specification of compliance for consistency with a task may be found in [De Schutter, 1988 #40] [Lee, 1988 #7].

---

[1] This work was supported by NSF grant DMC-8857854 and Northwestern University. Author's electronic mail address is peshkin@eecs.nwu.edu

[2] We use the generic meaning of the term "compliance" – the relationship of a manipulator's motion to the forces imposed on it. (Compliance has a specific meaning as well, namely the matrix characterizing spring-like behavior – see section 1.5)   In this paper we are designing an accommodation matrix, the inverse of a damping matrix.

The compliance of a manipulator has a profound effect on the evolution of a robotic assembly operation. Depending upon its compliance a manipulator's response to errors may drive the system toward or away from proper mating of the parts.   Even if a compliance matrix *consistent* with a task is in use (i.e. unbounded forces do not develop), the operation will not necessarily evolve toward decreased errors. We will call a form of compliance *error-corrective* if it causes an operation to evolve toward decreased errors.   Figure 1 shows an example in which a compliance consistent with a task is distinctly different from one which is both consistent and error-corrective.
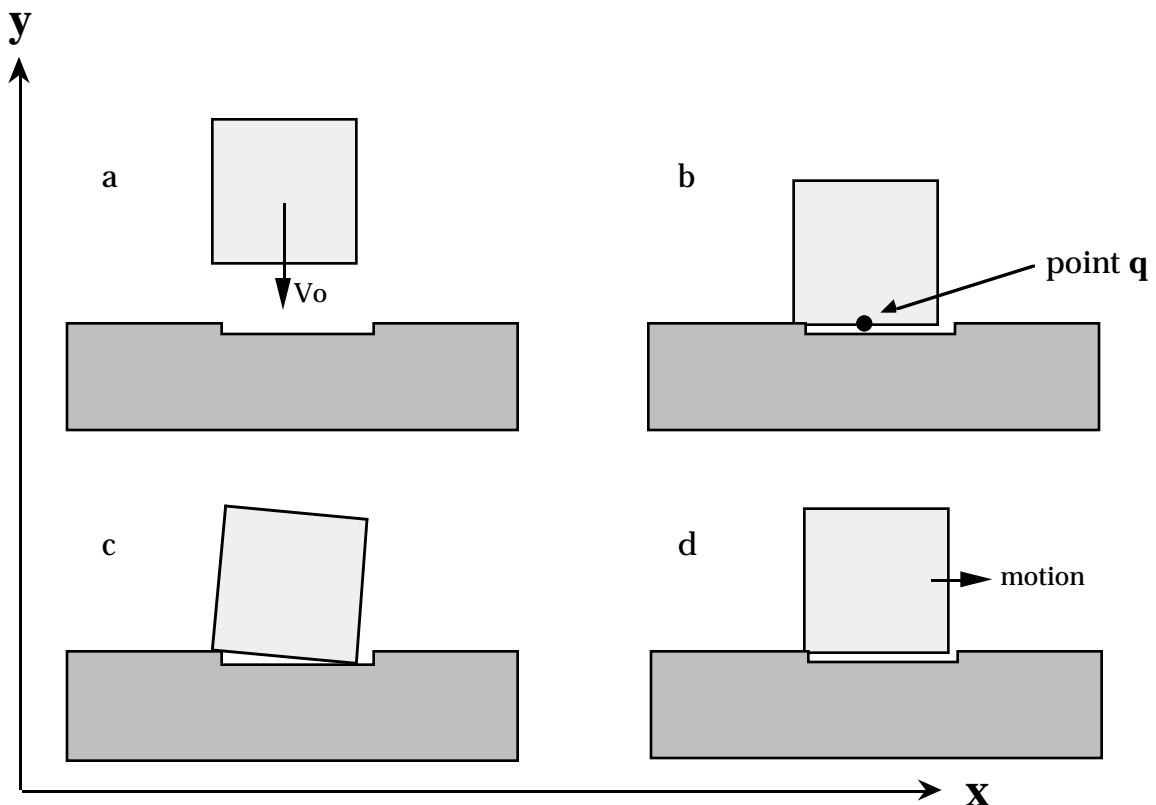

Figure 1

Insert figure 1 here.   Caption: A simple example of a task in which compliance *consistent* with the task will indeed prevent unbounded forces from developing, but will not cause *error-reduction* and eventual success of the mating operation.

In (a), a block is to be placed into a shallow depression or detent in a rigid surface. Since the task's rigid axis is the **y** axis, Mason's work would suggest that we make the manipulator soft (force controlled) along the **y** direction and stiff (position controlled)

along the **x** axis and the θ (rotational) axis - thus *consistent* with the mechanical constraints imposed by the task.

Forces due to contact as shown in (b) are in the +**y** direction, and there is a clockwise torque about point q.   The compliance of the manipulator allows some "give" in the **y** direction, and so the block comes to rest on the lip of the detent. If we let the θ axis be soft also, the outcome is (c), in which the block rests on the lip and bottom of the detent.

(d) shows the evolution if we have arranged the compliance of the manipulator for *error-reduction* as well: in particular we include an off-diagonal matrix element which maps *torque* into *horizontal displacement*. Clockwise torque maps into +**x** motion, and counterclockwise into –**x** motion.   Thus when the corner of the block comes into contact with the left edge, the torque generated results in a displacement which moves the block toward proper mating.

A compliance which maps torques into translations in the way described above may seem counterintuitive (see also section 1.5).   Intuition tells us that sliding toward proper mating will not occur unless the environment surface is steeply sloped toward the detent (i.e. has chamfered edges.)   Our intuition, however, corresponds to a very limited class of compliances.   If we allow ourselves freedom in choosing a compliance matrix we can obtain the surprising behavior above. This behavior in no way violates the "friction cone" construction from classical mechanics. Classical mechanics says that imposed forces within the friction cone result in sticking (no motion.)   Here, in contrast, we do not impose a force and observe the motion which occurs. Rather we observe a force, then compute and impose a motion.

Since Mason's work, several distinct approaches to the automated planning of contact and assembly tasks under force control[3] have developed:

In **Lozano-Perez, Mason & Taylor (LMT) fine motion planning** [Lozano-Perez, 1984 #45; Erdmann, 1984 #49; Buckley, 1987 #50; LaTombe, 1989 #61] the manipulator is given *a priori* a particular form of compliance (usually a generalized damper with a diagonal damping matrix; see section 1.5).   The goal of LMT planning is to design a nominal motion plan.   The nominal motion plan is a sequence of robot motions (usually given as velocities) each one of which is executed until motion ceases; from there the next motion commences.   It is sometimes possible to design a sequence of motions which proceeds inevitably despite the initial errors to a final stopping point at which the operation (perhaps the mating of two parts) is completed.

Error detection and recovery (EDR) [Donald, 1986 #42; Donald, 1988 #43] shares with LMT fine motion planning the same paradigm of a given (usually diagonal) compliance and a synthesized (generally multi-step) nominal motion plan.   The objective of the nominal motion plan is now broadened so that a plan which inevitably leads either to task completion, *or to termination at a recognizable failure*

---

[3] By "force control" we include all forms of control in which the response of the manipulator to forces it encounters is intentionally controlled. Among these are free-joints, hybrid control, compliance control, stiffness control, impedance control, and many generalizations and extensions.

configuration, is acceptable. If motion terminates at other than task completion, a secondary nominal motion plan which commences from that point is invoked. EDR plans, like LMT plans, are fully formed off-line.

In **replanning** [Xiao, 1988 #44][Desai, 1988], both the compliance and the nominal motion plan are given *a priori.* The nominal motion plan, in particular, is just the one that would be created without any consideration of errors, and so is unlikely to involve multiple sequential sliding motions. If motion terminates at other than task completion, and the resulting configuration is identifiable by force sensors or other sensors, an on-line replanner is invoked. The replanner computes the motion needed to bring the workpiece back into a configuration such that the nominal motion plan can proceed.

The above works are approaches to the planning of *generic* assembly tasks, as is this paper. One *particular* assembly task, peg-into-chamfered-hole, has due to its importance been heavily studied: Draper Lab's analysis of peg-into-hole led to development of the **RCC wrist** (remote-center compliance). RCC [Whitney, 1982 #38] can be implemented as a physical wrist, instead of a control loop involving force sensor and computer.   In RCC a simple nominal motion is assumed, in which errors of peg orientation are ignored. It is the compliance of the grasped peg, and the forces which arise due to errors, which cause the peg to self-align with the hole.

Another line of work, which results in what are sometimes described as **skeleton strategies** for particular tasks, again uses multi-step motion plans, sometimes combined with non-diagonal compliances. Strategies for round pegs were considered in [Inoue, 1979 #64; Simunovic, 1975 #53].   More recent work considers strategies for non-round pegs [Strip, 1988 #63; Caine, 1985 #62].

In contrast to off-line planning of a motion plan as in the above examples, or the off-line design of the manipulator's compliance as in this work, one might also consider an on-line **learning** of the forces which characterize errors and the proper motion response to them [Vaaler, 1987 #65][Hirzinger, 1985 #66].

1.2 RELATION TO THIS WORK

Programmed compliance, the subject of this paper, differs from LMT, EDR, and replanning in that we assume we are *given* a nominal motion plan, presumably a simple (one-step) one, which takes no account of possible errors.   We then *design* the compliance (here a damping matrix) so that the plan will succeed despite errors. In LMT, EDR, and replanning it is the compliance which is given and the motion plan which is to be designed.

Design of the compliance matrix for error-correction is the idea behind the development of the RCC wrist, and several others have addressed this idea also, (e.g. [Drake, 1977 #54; Simunovic, 1975 #53])   The present work is, to our knowledge, the first attempt to describe a method of synthesis of an error-corrective compliance,

given a description of a task.   We do not claim to solve the problem, but to pose it and to suggest one approach to it.

We should also note at this point that this paper is limited in scope to quasistatic assembly, i.e. to assembly speeds low enough that inertial effects of the robot and workpiece are small compared to frictional effects. Thus we only create a single, dynamics-independent real-valued accommodation (inverse-damping) matrix, in contrast to the work of Hogan on impedance control [Hogan, 1985 #20].

In the opposite (high-speed) regime the inertia matrix becomes the dominant factor in the evolution of an operation. There has been recent work [Asada, 1988 #41] on achieving inertia matrices appropriate for some individual tasks, particularly high-speed peg-in-hole insertion.

1.3 OBJECTIVE AND SUMMARY

The main points of the paper are these:

1. We observe that in many tasks it is possible for any one of several qualitatively distinct contact configurations between grasped workpiece and environment to arise, depending on small initial errors.   It is necessary to plan the compliance of a manipulator so that it is consistent with <u>every</u> <u>one</u> of the contact configurations, as we do not know in advance which ones may arise. We seek a compliance matrix which is to be employed for the entire duration of an assembly task; we do not here consider changing the compliance of the manipulator during a task. Our requirements on the compliance matrix are therefore quite severe: it must be simultaneously consistent with each of several possible contact configurations.

2. We suggest that it is not enough for the compliance of a manipulator to be *consistent* with a contact task, in the sense of the work of Mason and others, (i.e. preventing the occurrence of unbounded forces.) For many tasks, especially assembly tasks, we need the compliance to be *error-corrective* as well. We ask that the compliance of the manipulator have the effect of converting the forces which arise from misalignments of the workpiece into corrective motions. In section 2 we formalize these two desired properties of the manipulator's compliance.

3. We develop a set of sufficient conditions for our desired properties, all of a single mathematical form well suited to synthesis. We demonstrate an optimization method which efficiently finds a matrix satisfying the sufficient conditions if one exists, and if not finds one which comes close.   In the latter case it is possible to verify whether the original desired properties are in fact satisfied.

4. We demonstrate the synthesis of error-corrective compliance matrices for three example assembly tasks.   We find it encouraging that the result for the peg-into-chamfered-hole task is consistent with the remote-center-compliance (RCC) solution.

1.4 NOTATION

t   scalars are in light type
A   matrices are bold capital script
F, V   vectors are bold capitals Roman
f   unit vectors are bold lower case

Our examples are planar. In these examples vectors have three components **(x, y, θ)** where **x** is horizontal, **y** is vertical, and θ is rotation about the **z** axis (i.e. counter-clockwise rotation in the plane of the paper.)

By "forces" we mean forces and torques - a six vector in a three-dimensional world, or a three-vector in a planar world.   Similarly, by "displacements" we mean displacements and rotations.

1.5 THE DAMPER MODEL AND THE ACCOMMODATION MATRIX

The function which maps the forces applied to a manipulator into the motion which ensues due to those forces may take many forms. For instance, the function may state that displacement of the manipulator from some nominal position **X$_0$** (which could be changing with time) is proportional to the applied force:

$$\mathbf{X} - \mathbf{X_0} = C\mathbf{F} \tag{1}$$

This is Hooke's law, representing the behavior of an ideal spring, where C is the compliance matrix.

An object held by a manipulator may have six degrees of freedom: three translational and three rotational. For such an object, equation (1) may be understood in vector form with **X** and **X$_0$** representing locations in six-dimensional configuration space, **F** a six-dimensional vector of forces and torques, and C a six-by-six matrix.

Equation (1) is often a good approximation for the natural compliance of mechanical systems composed of solids.   We need not, however, restrict ourselves to natural compliance. If a manipulator is equipped with a force/torque sensor, we can perform arbitrary computation on those measurements and arrive at a motion, which the manipulator then executes. Thus any compliance function may be achieved. This is called *active compliance*.   If we implement the control law given in (1), we have simulated an ideal spring by using active compliance.

As a practical matter the control law we implement must be evaluated rapidly, so there is a premium on simple control laws. The one we choose in this paper is

simple, has been studied extensively in the past, and is usually called the *generalized damper:*

$$\mathbf{V} = \mathbf{V_o} + A\,\mathbf{F} \tag{2}$$

Control law (2) states that in the absence of any perceived forces $\mathbf{F}$, the manipulator is to follow a nominal trajectory given by velocity $\mathbf{V_o}$. In the presence of forces, the matrix A times the six-vector of forces $\mathbf{F}$ specifies a term $A\mathbf{F}$ which is simply added to the nominal velocity $\mathbf{V_o}$ to yield the velocity $\mathbf{V}$ which the manipulator is to perform[4]. A is an *accommodation matrix*, the inverse of a damping matrix, and characterizes a generalized damper.

We have a further reason to restrict ourselves to the design of an accommodation *matrix* – i.e. to a *linear* relationship of assembly force to consequent velocity. It may be possible to design a passive mechanical wrist which can assume any (or a large range of) accommodation matrices. The range of accommodation matrices which could be assumed by such a wrist is considered in [Goswami, 1990 #97].

1.6   CONTACT CONFIGURATIONS

Our approach to the creation of an error-corrective matrix for a given assembly operation proceeds from the identification of a set of *contact configurations* as shown in Figure 2. In this example the assembly operation being performed is again the placement of a block into a shallow detent in a fixed chassis.   (Note that the RCC wrist would not be able to perform this operation as there are no chamfers.) The nominal assembly trajectory of the manipulator, $\mathbf{V_o}$, is straight down.   In this paper we do not address the selection of $\mathbf{V_o}$.

---

[4] For example, to compare the motion of a manipulator under control laws (1) vs. (2), suppose that the manipulator is moving steadily in the -$\mathbf{z}$ direction, and we give it a push in the +$\mathbf{x}$ direction. If C (or A) is a diagonal matrix with positive elements, in either (1) or (2) the manipulator will respond by deflecting in the +$\mathbf{x}$ direction. After we cease pushing, a manipulator following law (1) will spring back to its original path and proceed as if it had not been disturbed. A manipulator following law (2) will resume its original <u>direction of motion</u> (-$\mathbf{z}$) but will not move back to its original path: its displacement in the +$\mathbf{x}$ direction is permanent.

With a diagonal matrix C (or A) the motion of the manipulator under either (1) or (2) is intuitive: it moves in the direction pushed. However in selecting a matrix we have considerable freedom, and highly non-intuitive behaviors may result.   For instance we can, by making the proper element of C (or A) non-zero, cause forces in the +$\mathbf{x}$ direction to map into motions in the +$\mathbf{y}$ direction – or even into a rotation about the $\mathbf{z}$ axis if so desired.
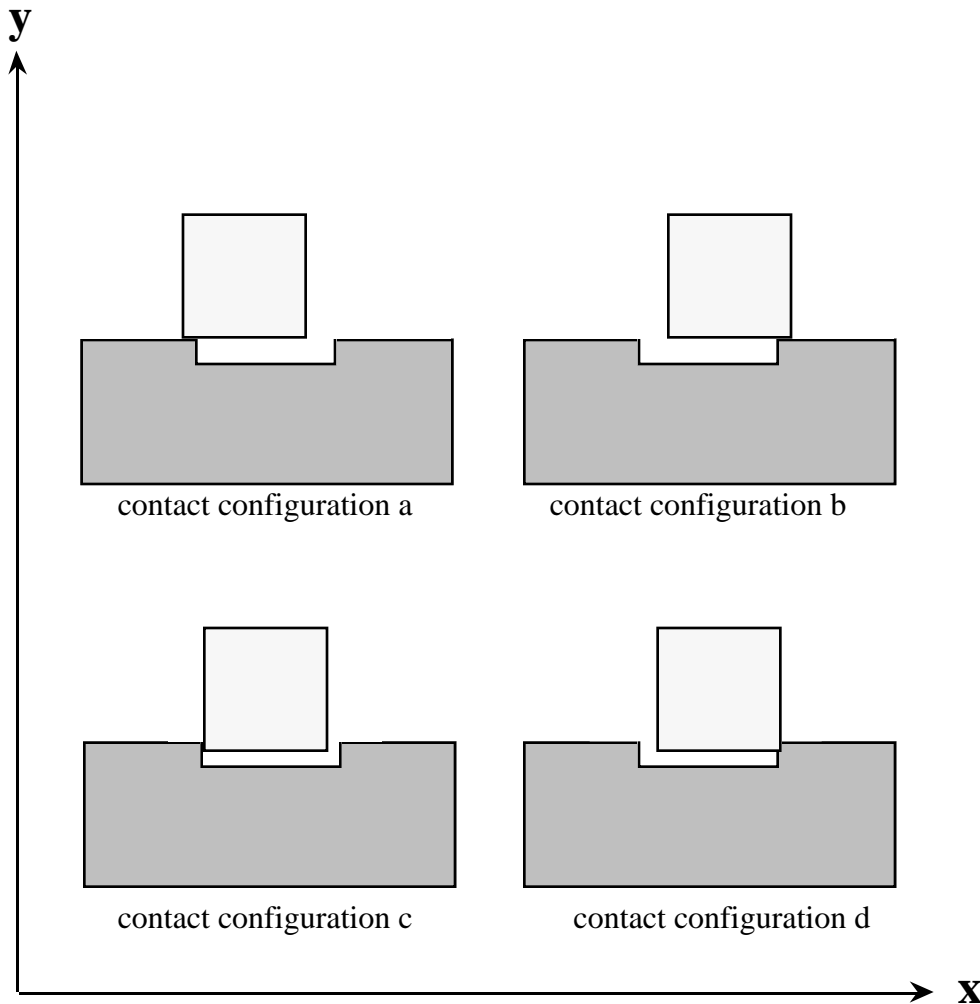
Figure 2

Insert figure 2 here. Caption: Qualitatively distinct contact configurations which may result during the block-into-detent assembly task.   The nominal assembly trajectory $\mathbf{V_o}$ is -$\mathbf{y}$ (down, not shown). Each contact configuration corresponds to a deviation of the movable part (the block) from its properly mated configuration. Sections a and b show contact configurations in which the edge of the block has come into contact with the left or right lip of the detent. In sections c and d the block is in contact with the left or right inside wall of the detent.

We should note that specification of a complete set of contact configurations depends on more geometric information than is available in a CAD model of the parts alone. In particular we must know how great the initial configurational errors are. As an extreme example, if large horizontal initial errors are possible it may

happen that the block will miss the detent entirely. In the selection of contact configurations for Figure 2, we have assumed that orientational errors are negligible and that translational errors are limited to about 2mm.

Specification of the nominal trajectory $\mathbf{V_o}$ and the contact configurations is our starting point in this paper.   We do not here address the problem of extracting the contact configurations from geometric models, nor is it clear how difficult this task would be.   Donald's *error detection and recovery regions* (EDR regions) [Donald, 1986 #42; Donald, 1988 #43] and Desai & Volz' *contact formations* [Desai, 1988] contain much the same information as our contact configurations.   Other related work includes [Brooks, 1982 #56; Buckley, 1987 #50; Erdmann, 1984 #49].

2.1 FORMAL PROPERTIES OF THE ACCOMMODATION MATRIX FOR ERROR-CORRECTIVE MANIPULATION

We pose the problem of creating an appropriate error-corrective accommodation matrix for a manipulator as that of realizing two desired properties of the manipulator's motion during an operation. These we will express in terms of the contact configurations.

Property 1: "Bounded forces":

For every contact configuration, A must be such that the contact forces which arise between mating parts are bounded.

Property 1 is necessary to prevent damage to the manipulator and/or workpiece. This is akin to the condition justified and studied by Mason [Mason, 1981 #37]. Mason translated the bounded force condition into requirements on an accommodation matrix, and showed how an accommodation matrix could be found which guaranteed bounded forces for a given contact configuration.   Our work differs in that we now consider several distinct contact configurations to all of which the bounded forces condition must apply simultaneously.

> For example, in figure 2a, $A\mathbf{F}$ (the product of the accommodation matrix A with the forces created by contact) must cause the lower left corner of the block to have a velocity component in the surface-normal direction $+\mathbf{y}$ (upward) to avoid unbounded forces.

Property 2: "Error-reduction":

For every contact configuration, A must be such that the magnitude of the displacement from proper mating $\mathbf{E}$ is reduced.

> For example, in figure 2a, $A\mathbf{F}$ must have a component in the $+\mathbf{x}$ (rightward) direction to make progress toward proper mating.

For the moment we leave unspecified the metric $\mathbb{M}$ by which the "magnitude of the displacement from proper mating **E**" is to be measured.   This is a non-trivial detail because the error vector **E** has both translational and orientational components (e.g. "centimeters" and "degrees").   The $\mathbb{L}^2$ metric (in which we square and add the components) has mixed units and makes no sense.   See [Lipkin, 1990 #99] for further discussion of suitable unit- and frame-invariant metrics.

Fortunately, if there is *any* metric $\mathbb{M}$ for which we can show that an accommodation matrix we synthesize results in decreasing $|\mathbf{E}|$ for all contact configurations, we are assured that the operation must terminate at proper assembly.   Our choice of metric is important: if we make a poor choice we will not be able to synthesize an accommodation matrix A for which the distance to proper mating $|\mathbf{E}|$ is guaranteed to decrease.   Yet our choice of metric need not pass any formal test of non-arbitrariness.

3.1 A METHOD OF SYNTHESIS AND OPTIMIZATION

In this section we describe a synthesis and optimization method.   It is by no means the only one which might be used to create a matrix A satisfying the "bounded forces" and "error reduction" properties above, and some of its shortcomings will be discussed in section 5.

We will divide the "error reduction" property into two new conditions on A so that the new conditions and the "bounded forces" property will have a common mathematical form – one that is convenient for synthesis. We will show how A may be efficiently created so that it closely satisfies the whole set of conditions.

For each contact configuration we determine the forces which we may expect the manipulator to perceive.   It turns out that we only need the direction, and not the magnitude, of these characteristic forces. The direction is easily accessible to us from the familiar friction-cone construction. Given the direction of a characteristic force $\mathbf{f_i}$, we can express all of the conditions in the form $\mathbf{u_i} \cdot A\mathbf{f_i} = t_i$, where $\mathbf{u_i}$ is a unit vector available to us from the geometry of the contact configurations (for instance $\mathbf{u_i}$ may be a surface normal), and t_i is a scalar.

Generally it turns out that the number m of conditions on A:

$$\mathbf{u_i} \cdot A\mathbf{f_i} = t_i \qquad i = 1...m \qquad\qquad (3)$$

exceeds the number of elements[5] in the matrix A. It is therefore not generally possible to find a accommodation matrix A which simultaneously satisfies all of the

---

[5] Nine in two-space or thirty-six in three-space.

conditions *exactly.*   All is not lost, because the values t_i are not critical.   The scalar t_i corresponds to an inverse stiffness. In the literature on hybrid control, for instance, there are "soft" axes and "stiff" axes, represented by stiffnesses k_{soft} and k_{stiff}, but the exact values of k_{soft} and k_{stiff} are not critical. So too here we expect that while typical values of the t_i can be given, if it is not possible to form a accommodation matrix with those precise values, one with slightly different values will also be acceptable.

We can use the freedom to adjust the inverse stiffnesses t_i to allow us to form a accommodation matrix A which *almost* satisfies the whole set of requirements, even when there is none that exactly satisfies them.   We can then return to the "bounded forces" and "error reduction" properties to check that the matrix A synthesized does indeed satisfy them.

3.2 PREDICTING THE FORCES WHICH CHARACTERIZE A CONTACT CONFIGURATION.

To design a accommodation matrix A which causes a desired motion in a particular situation we must be able to predict the contact force **F** exerted on the manipulator in that situation. It turns out that if we know the direction of the force **F**, we can calculate its magnitude, and this will be done immediately below. In the case of one-point contacts, we do know something about the direction of the force **F**, namely that it must lie within the friction cone as shown in Figure 3. Further, if we know that the direction of motion of the block in Figure 3 will be to the right (+**x**), we may predict that **F** will lie on the left extreme of the friction cone (**f**+) as shown. Contacts at more than one point are considered in section 3.2.3.
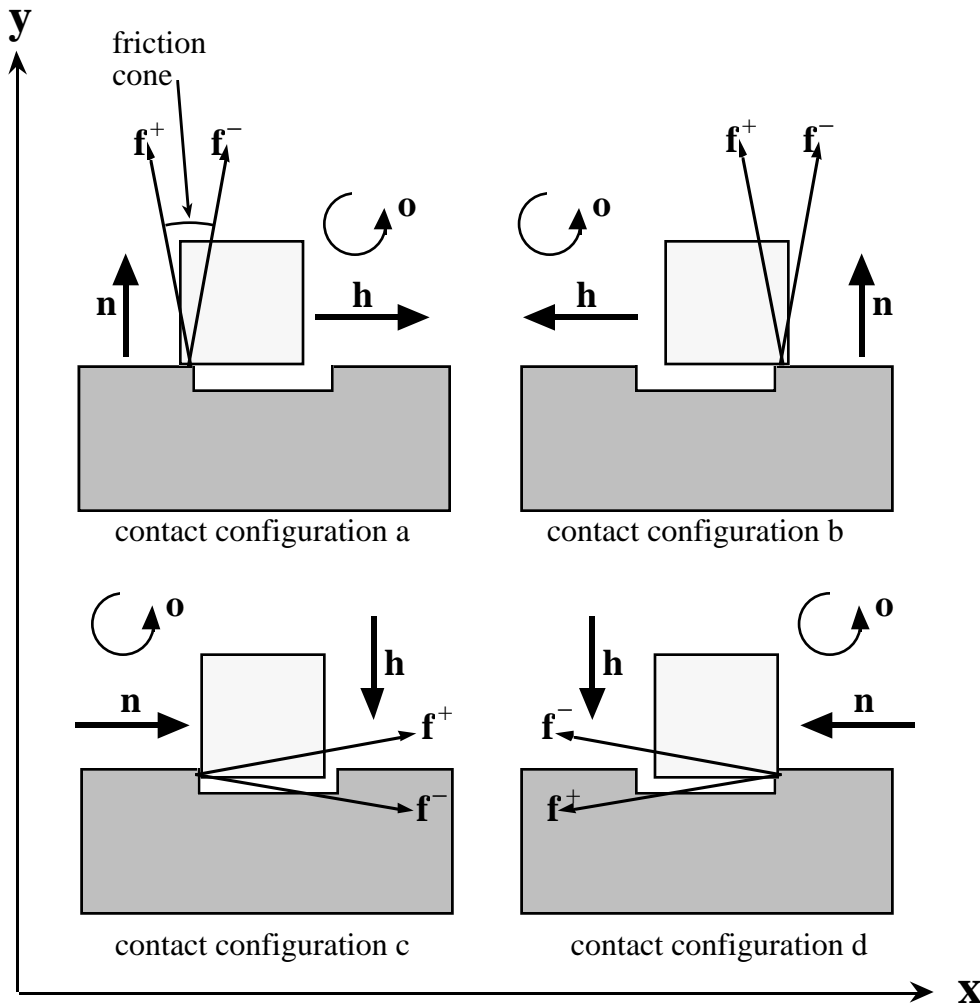
Figure 3

Insert figure 3 here. Caption: Here we have added to each of the qualitatively distinct contact configurations of figure 2 the surface normal (**n**), home (**h**), and orthogonal (**o**) vectors (there is only one orthogonal vector in planar examples.)   Also shown are the friction cones, the extremes of which are the vectors $\mathbf{f}^{\pm}$.

We could build our accommodation matrix A using the assumption that **F** will lie on the left extreme of the friction cone in Figure 3a. However, given the present state of the art in implementing force control, the assumption of a monotonic +**x** motion which will keep **F** on the left extreme seems questionable at best. Instead we will assume only that **F** lies within the friction cone, and ask that no matter where in the friction cone **F** lies, the "bounded forces" and "error reduction" properties are satisfied. Our conservative assumption about the direction of **F** contributes to a robustness of the resulting accommodation matrix A and, hopefully, relative

insensitivity to the accuracy of a physical implementation of the control law
$\mathbf{V} = \mathbf{V_o} + A\mathbf{F}$.

3.2.1   EXPRESSING THE "BOUNDED FORCES" PROPERTY

Property 1 requires that the motion of the manipulator must be such that the contact forces which arise between mating parts are bounded. To find the implications of this requirement for A, we evaluate the magnitude of the force $\mathbf{F}$ which arises in a particular contact configuration.

First, we express $\mathbf{F}$ as a product of its magnitude and direction: $\mathbf{F} = | F | \mathbf{f}$. Let $\mathbf{n}$ be the surface normal as shown in the Figure 3. Since the velocity of the block $\mathbf{V}$ can have no component in the direction $\mathbf{n}$, we have $\mathbf{n} \cdot \mathbf{V} = 0$.   (Some care must be taken in performing dot products among vectors of mixed distances and angles [Ball, 1900 #102; Hunt, 1978 #103])

Combining the above equations with our control law $\mathbf{V} = \mathbf{V_o} + A\mathbf{F}$ we can solve for the magnitude of the force

$$| F | = -\frac{\mathbf{n} \cdot \mathbf{V_o}}{\mathbf{n} \cdot A\mathbf{f}} \tag{4}$$

The scalar $(\mathbf{n} \cdot A\mathbf{f})^{-1}$ represents the <u>stiffness</u> of the manipulator to interactions with a particular surface. To guarantee that unbounded contact forces do not arise due to velocities which bring the parts together, no matter what the source of these velocities, it is only necessary to require a finite (non-infinite) stiffness. Property 1 becomes

Property 1: "Bounded forces" $\Rightarrow$ $\mathbf{n} \cdot A\mathbf{f} \approx t_1$ $\qquad\qquad\qquad\qquad$ (5)

A reasonable value for the scalar stiffness $1/t_1$ depends on the task being performed, the materials involved, the nominal velocity $\mathbf{V_o}$ chosen, and other factors.   We will presume that while a user can select a typical value for t_1, he or she will not feel very strongly about the exact value.   For the rest of this paper it is important only to notice that t_1 is positive, and its value user-selectable.

Our method for creating A (section 3.3) will take advantage of the freedom to adjust t_1. Property 2 will result in a similar parameter (t_2) being introduced.

3.2.2   EXPRESSING THE "ERROR REDUCTION" PROPERTY

Property 2 requires that the motion of the manipulator reduce the distance from the manipulator's present configuration to a properly mated configuration, as measured by some metric $\mathbb{M}$. We will use as our metric $\mathbb{M}$ a modified $\mathbb{L}^2$ metric in which the

angular components (in radians) are multiplied by a characteristic dimension of the assembly task (e.g. the width of the grasped object) and then the components are squared and summed[6].   This metric is physically justifiable though still arbitrary. Again, we are content with a matrix A if it satisfies property 2 for *any* metric so we may indeed choose $\mathbb{M}$ arbitrarily.

To express property 2 in a form similar to (5) above, let us now identify a "home" unit vector **h** for each contact configuration. We choose **h** to be the direction consistent with sliding constraints in which the distance from proper mating is most steeply decreasing, as measured by our metric $\mathbb{M}$.

With **h** and **n** identified, there are several[7] other vectors $\mathbf{o_i}$ orthogonal to both **h** and **n**. Figure 3 shows examples of **h**, **n**, and **o**.

We can now write *sufficient* conditions for property 2 (error-reduction) as:

$$\mathbf{h} \cdot \mathbf{V} > 0, \qquad \text{and} \tag{7a}$$

$$\mathbf{o_i} \cdot \mathbf{V} = 0 \quad \forall_i \tag{7b}$$

These conditions ask that there be motion in the **h** "home" direction (7a), and that there not be motion in orthogonal directions (7b).   The required velocity at which errors are reduced in (7a) is not critical, but for definiteness let us request that $\mathbf{h} \cdot \mathbf{V} \approx |\mathbf{V_o}|$, i.e. we wish the manipulator to move toward proper mating at approximately the full nominal velocity.

Using the control law $\mathbf{V} = \mathbf{V_o} + A\mathbf{F}$, our requirement $\mathbf{h} \cdot \mathbf{V} \approx |\mathbf{V_o}|$, and $\mathbf{n} \cdot \mathbf{V} = 0$, we find

$$\mathbf{h} \cdot A\mathbf{f} \approx (\mathbf{h} \cdot \mathbf{V_o} - |\mathbf{V_o}|) \; \frac{\mathbf{n} \cdot A\mathbf{f}}{\mathbf{n} \cdot \mathbf{V_o}} \tag{8}$$

Note that $\mathbf{n} \cdot \mathbf{V_o}$ is negative[8]. $\mathbf{V_o}$ is our nominal assembly velocity and we have already required that $\mathbf{n} \cdot A\mathbf{f} \approx t_1$ for some user-selectable parameter t_1. The right side

---

[6] $\mathbb{M}(x, y, z, \theta_x, \theta_y, \theta_z) = (x^2 + y^2 + z^2) + w^2 (\theta_x{}^2 + \theta_x{}^2 + \theta_z{}^2)$, where w is a characteristic length of the task.

[7] One in 2-d, or four in 3-d.

[8] If $\mathbf{n} \cdot \mathbf{V_o} \approx 0$ it will be best to impose no requirement on $\mathbf{h} \cdot A\mathbf{f}$.   In this case if $\mathbf{h} \cdot \mathbf{V_o} > 0$ the nominal trajectory $\mathbf{V_o}$ itself causes reduction of error, and no adjustment of A will increase or decrease the rate much. If $\mathbf{h} \cdot \mathbf{V_o} < 0$ the situation is hopeless.

of (8) can be evaluated and we will denote it t_2, another positive constant. We may now write property 2 as

Property 2: "Error Reduction" $\Rightarrow$ $\mathbf{h} \cdot A\mathbf{f} = t_2$,   *and*   (9a)

$\mathbf{o_i} \cdot A\mathbf{f} = 0$ $\forall_i$ (9b)

> For example, consider contact configuration a in Figure 3. The nominal direction $\mathbf{V_o} = -\mathbf{y}$, the surface normal $\mathbf{n} = \mathbf{y}$, and the home direction $\mathbf{h} = +\mathbf{x}$.   The "bounded forces" property is thus $\mathbf{y} \cdot A\mathbf{f} = t_1$.   In place of the "error-reduction" property, condition (9a) gives us $\mathbf{x} \cdot A\mathbf{f} = t_2$. Condition (9b) applies only to the one remaining degree of freedom θ, and is $\theta \cdot A\mathbf{f} = 0$, i.e. $A\mathbf{f}$ should have no rotational component.

3.2.3 EXTENSION TO CONTACT CONFIGURATIONS WITH MORE THAN ONE POINT OF CONTACT

If there are two or more points of contact between workpiece and environment, the total force felt by the manipulator is the sum of the forces at the individual contacts. For simplicity we will work here with only two contacts; the analysis extends trivially to any number.   Denote these individual forces $\mathbf{F_1}$, $\mathbf{F_2}$. The motion of the manipulator is still given by the control law $\mathbf{V} = \mathbf{V_o} + A\mathbf{F}$, which can now be written $\mathbf{V} = \mathbf{V_o} + A\mathbf{F_1} + A\mathbf{F_2}$. We can break each force into a product of its magnitude with its direction $\mathbf{F_i} = |\mathbf{F_i}| \ \mathbf{f_i}$. We also have one condition of the form $\mathbf{V} \cdot \mathbf{n_i} = 0$ for each contact.   We can therefore construct a set of equations from which the magnitude of the forces can be extracted:

$$\mathbf{n_1} \cdot \mathbf{V_o} + |\mathbf{F_1}| \ \mathbf{n_1} \cdot A\mathbf{f_1} + |\mathbf{F_2}| \ \mathbf{n_1} \cdot A\mathbf{f_2} = 0 \qquad (10a)$$
$$\mathbf{n_2} \cdot \mathbf{V_o} + |\mathbf{F_1}| \ \mathbf{n_2} \cdot A\mathbf{f_1} + |\mathbf{F_2}| \ \mathbf{n_2} \cdot A\mathbf{f_2} = 0$$

which can be written as

$$\begin{pmatrix} \mathbf{n_1} \cdot A\mathbf{f_1} & \mathbf{n_1} \cdot A\mathbf{f_2} \\ \mathbf{n_2} \cdot A\mathbf{f_1} & \mathbf{n_2} \cdot A\mathbf{f_2} \end{pmatrix} \begin{pmatrix} |\mathbf{F_1}| \\ |\mathbf{F_2}| \end{pmatrix} = - \begin{pmatrix} \mathbf{n_1} \cdot \mathbf{V_o} \\ \mathbf{n_2} \cdot \mathbf{V_o} \end{pmatrix} \qquad (10b)$$

As in section 3.2.1 we may identify the determinant of the two-by-two matrix as the inverse stiffness of the manipulator.   Property 1 (bounded forces) therefore becomes

$$\det \begin{pmatrix} \mathbf{n_1} \cdot A\mathbf{f_1} & \mathbf{n_1} \cdot A\mathbf{f_2} \\ \mathbf{n_2} \cdot A\mathbf{f_1} & \mathbf{n_2} \cdot A\mathbf{f_2} \end{pmatrix} \approx t_1{}^2 \qquad (11a)$$

For the general case of *m* contacts where { $\mathbf{n}$ } represents the *m* surface normals and { $\mathbf{f}$ } represents the *m* force directions we may write this condition compactly (if opaquely) as an outer product

$$\det\left(\{\mathbf{n}\} \otimes A \{\mathbf{f}\}\right) \approx t_1{}^m \tag{11b}$$

It is useful to notice that if a contact configuration with more than one point of contact can occur, it is also possible for the points of contact to occur individually as contact configurations. The "bounded forces" property applied to these individual contact configurations involve the terms $\mathbf{n}_i \cdot A\mathbf{f}_i$ that appear on the diagonal in (11a). Since we have already required that $\mathbf{n}_i \cdot A\mathbf{f}_i \approx t_1$, one way to satisfy (11a) is simply to require that $\mathbf{n}_i \cdot A\mathbf{f}_j \approx 0$ when $i \neq j$, i.e. that all the off-diagonal terms be zero. This observation will be useful in the following section, because it allows us to reduce all of the properties on all of the contact configurations to a single mathematical form.

3.3 FORMING AN ACCOMMODATION MATRIX

All of our conditions on the accommodation matrix A have now been reduced to numerous conditions of the form

$$\mathbf{u}_i \cdot A\mathbf{f}_i = t_i \tag{14}$$

where $\mathbf{f}_i$ is a force to be expected, $\mathbf{a}_i$ is a direction vector determined from the geometry of the contact configuration, and t_i is a target value for the dot product. For some conditions the target value t is zero. For others t is a user-selectable positive value whose choice depends on the task at hand, materials, etc, and indeed is somewhat arbitrary. Selection of these t_i is guided by equations (4) and (8).

Recall that $\mathbf{f}$ in equations of the form (14) represents <u>any</u> force direction which could arise in a particular contact configuration. $\mathbf{f}$ must necessarily lie within the friction cone, but we are reluctant to presume it any better localized than that.   Let the two extremes of the friction cone be denoted $\mathbf{f}^{\pm}$. Then any force $\mathbf{f}$ that arises can be written as a positive linear combination of $\mathbf{f}^+$ and $\mathbf{f}^-$.   It will not be possible to design a accommodation matrix A for which $\mathbf{u} \cdot A\mathbf{f}^+ = \mathbf{u} \cdot A\mathbf{f}^- = t$.   However our target value t was not critical, and perhaps we can design a accommodation matrix A for which $\mathbf{u} \cdot A\mathbf{f}^+ = t^+$, $\mathbf{u} \cdot A\mathbf{f}^- = t^-$, with both t^+ and t^- acceptably close to the target t.   If so we are assured that the value of $\mathbf{u} \cdot A\mathbf{f}$ will be acceptably close to t for any $\mathbf{f}$ which may arise.

3.3.1 LEAST-SQUARES OPTIMIZATION

For each of several contact configurations we consider two forces $\mathbf{f}^{\pm}$, and for each force we have several conditions of the form $\mathbf{u}_i \cdot A\mathbf{f}_i = t_i$. Let m be the total number of such equations -- clearly m may be large. The $\mathbf{u}_i$, $\mathbf{f}_i$, and t_i are all given; it is A we wish to create. A is a matrix consisting of n real numbers. In our two-

dimensional examples $n=9$; in three dimensions we would have $n=36$.   The usual case is that there is no A which satisfies all m equations.

We describe next a way of finding the best A in the "least-squares" sense: the A which minimizes $\sum_{i=1}^{m} (\mathbf{u_i} \cdot A\mathbf{f_i} - t_i)^2$.   Solving for this A is computationally quick even for $m=200$, $n=36$.

How are we to interpret the resulting A? A satisfactory A is one for which the "bounded forces" and "error reduction" properties hold, so that motion can only terminate at proper mating.   Our m conditions of the form $\mathbf{u_i} \cdot A\mathbf{f_i} = t_i$ are stronger (sufficient) conditions, so if an A satisfies them all exactly, that A is satisfactory.

Further, of the { t_i }, some were zero (equation 9b), and others were arbitrary positive values (equations (5) or (9a)) .   If the A we create satisfies the (9b)-type conditions and gets *any* positive values for the (5) and (9a)-type conditions, that A is also satisfactory, but for a different choice of the arbitrary stiffnesses.   Even if the A we create does *not* satisfy the (9b)-type conditions, it may *still* be satisfactory, but then we must check the "bounded forces" and "error reduction" properties explicitly.

We now abandon the matrix structure and meaning of A, and simply consider it to be a set of n real-valued variables (here $n=9$) for which we desire a solution satisfying m equations.   To maintain a compact notation we will form these n variables into a n-vector A^{so}, here a 9-vector. A is "strung-out" by rows and displayed as a column vector.   Explicitly,

$$A = \begin{pmatrix} A_{\mathrm{xx}} & A_{\mathrm{xy}} & A_{\mathrm{x\theta}} \\ A_{\mathrm{yx}} & A_{\mathrm{yy}} & A_{\mathrm{y\theta}} \\ A_{\mathrm{\theta x}} & A_{\mathrm{\theta y}} & A_{\mathrm{\theta\theta}} \end{pmatrix} \implies A^{\mathrm{so}} = \begin{pmatrix} A_{\mathrm{xx}} \\ A_{\mathrm{xy}} \\ A_{\mathrm{x\theta}} \\ A_{\mathrm{yx}} \\ A_{\mathrm{yy}} \\ A_{\mathrm{y\theta}} \\ A_{\mathrm{\theta x}} \\ A_{\mathrm{\theta y}} \\ A_{\mathrm{\theta\theta}} \end{pmatrix} \tag{17}$$

Having formed this unintuitive construction, an equation of the form $\mathbf{u} \cdot A\mathbf{f} = t$ which is explicitly

$$\left(\begin{array}{ccc} u_x & u_y & u_\theta \end{array} \right) \cdot \left(\begin{array}{ccc} A_{xx} & A_{xy} & A_{x\theta} \\ A_{yx} & A_{yy} & A_{y\theta} \\ A_{\theta x} & A_{\theta y} & A_{\theta\theta} \end{array} \right) \left(\begin{array}{c} f_x \\ f_y \\ f_\theta \end{array} \right) = t \tag{19}$$

can be written as:

$$\left(u_x f_x \;\; u_x f_y \;\; u_x f_\theta \;\; u_y f_x \;\; u_y f_y \;\; u_y f_\theta \;\; u_\theta f_x \;\; u_\theta f_y \;\; u_\theta f_\theta \right) \cdot A^{so} = t \tag{20}$$

Each of the m equations of form $\mathbf{u} \cdot A\mathbf{f} = t$ can be written as above, for $i = 1 \dots m$, so that we have an enormous matrix equation:

$$\left(\begin{array}{ccccccccc} u_x^1 f_x^1 & u_x^1 f_y^1 & u_x^1 f_\theta^1 & u_y^1 f_x^1 & u_y^1 f_y^1 & u_y^1 f_\theta^1 & u_\theta^1 f_x^1 & u_\theta^1 f_y^1 & u_\theta^1 f_\theta^1 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ u_x^m f_x^m & u_x^m f_y^m & u_x^m f_\theta^m & u_y^m f_x^m & u_y^m f_y^m & u_y^m f_\theta^m & u_\theta^m f_x^m & u_\theta^m f_y^m & u_\theta^m f_\theta^m \end{array} \right) \cdot A^{so} = \left(\begin{array}{c} t^1 \\ \cdot \\ \cdot \\ t^m \end{array} \right) \tag{21}$$

We will name the great m-row by n-column matrix on the left G, and the m-high column vector of target values $\mathbf{t}$.   In these terms the problem of finding a accommodation matrix A which satisfies our m conditions is reduced to finding a n-vector A^{so} which satisfies

$$G \cdot A^{so} = \mathbf{t} \tag{22}$$

Generally there is no A^{so} which satisfies (22) exactly. However it is easy to check that

$$\| G \cdot A^{so} - \mathbf{t} \|^2 = \sum_{i=1}^{m} (\mathbf{u_i} \cdot A\mathbf{f_i} - t_i)^2 \tag{23}$$

The right side is exactly the sum we wish to minimize – the one that defines the "best" A.   There is a good and fast way of finding the A^{so} which minimizes the left side. Using the singular value decomposition (described particularly clearly in [Press, 1985 #39]) we can form an n-by-m pseudo-inverse[9] G^* of the m-by-n matrix G for which the value of A^{so} given by

_____

[9] There are other pseudo-inverses. Computation of the SVD pseudo-inverse is numerically robust, and the SVD pseudo-inverse possesses the needed least-squares property.

$$A^{\text{so}} = G^{*}\, \mathbf{t} \tag{25}$$

minimizes $\| G\, A^{\text{so}} - \mathbf{t} \|^{2}$ over all possible A^{so}. If (fortuitously) there is a A^{so} which satisfies (22) exactly then (25) finds it. In this case the sum-of-squares is zero.

G contains all the <u>geometry</u> of the contact configurations and forces which arise in assembly. **t** contains our <u>preferences</u> for the user-selectable parameters t_i. If we pick a target vector of parameters **t**, we can easily compute the accommodation matrix A which comes closest to achieving those parameters using $A^{\text{so}} = G^{*}\, \mathbf{t}$. Usually we can't have that set of parameters; the closest available set is $\mathbf{t}^{\sim} = G\, A^{\text{so}} = G\, G^{*}\, \mathbf{t}$.

To see how well A achieves our preferred parameters, we compare **t** to $\mathbf{t}^{\sim}$. Once the m-by-m matrix $G\, G^{*}$ has been formed, we can try out different target vectors **t** with little overhead.

As mentioned above if $\mathbf{t}^{\sim}$ has zeros in the same components that **t** has zeros, and is positive in all other components, then A possesses the "bounded forces" and "error reduction" properties.   The only consequence of not satisfying all of the conditions exactly is that the stiffnesses are not exactly as requested.

If $\mathbf{t}^{\sim}$ does not match **t** in sign, component by component, then to verify that A possesses the "bounded forces" and "error reduction" properties we must compute $\mathbf{V} = \mathbf{V_o} + A\, \mathbf{F}$ for every contact configuration and explicitly check that the properties are satisfied.

4.1 EXAMPLES

In this section we set up (by hand) the contact configurations and the normal, home, and orthogonal vectors for some sample planar assembly tasks. To create the accommodation matrix A which achieves the desired "bounded forces" and "error-reduction" properties we proposed in section 2, we use the stronger conditions and the optimization procedure described in section 3.   When even the stronger conditions can be satisfied (or satisfied for a slightly different set of stiffnesses) we are done. This occurs in the first two examples. In the third example we must verify that the desired properties are in fact satisfied for all contact configurations.

4.2 PLACE BLOCK INTO CHAMFERLESS DETENT

Here a block is to be placed into a shallow detent in a surface (Figure 1).   We wish to create a accommodation matrix so that the torque which arises if the edge of the block hits the lip of the detent is mapped by the accommodation matrix into a

horizontal motion of the block towards the center of the detent.   This motion is counterintuitive since the detent has no chamfers, but it is not impossible.

It is necessary to choose a coordinate system: an origin about which torques are measured and rotations are performed. The choice of origin is of absolutely no account; in fact if we create a accommodation matrix for origin $\mathbf{q}$, it is trivial to convert it to a accommodation matrix for origin $\mathbf{p}$ simply by

$$
A_{\mathbf{p}} = \begin{pmatrix} 1 & 0 & q_y - p_y \\ 0 & 1 & p_x - q_x \\ 0 & 0 & 1 \end{pmatrix} A_{\mathbf{q}} \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ q_y - p_y & p_x - q_x & 1 \end{pmatrix} \tag{26}
$$

For a given force in the real world, the accommodation matrix A_{p} at origin $\mathbf{p}$ produces exactly the same motion of the manipulator as would accommodation matrix A_{q} at origin $\mathbf{q}$. Inconsequential as it is, we must choose an origin, so we choose it at $\mathbf{q}$ as shown in Figure 1.

The four possible contact configurations are shown in Figure 3. It should be noted that implicit in the specification of only these contact configurations is information about the maximum possible magnitudes of the robot's initial error. If, for instance, the possible horizontal translational error were much greater, the block could miss the detent altogether. We presume, not unrealistically, that errors of orientation are minimal.

For each of the four contact configurations there are two extremes of the friction cone.   Translating the "bounded forces" and "error-reduction" properties into three stronger conditions on each of the eight force-and-configuration pairs, we find a total of 24 conditions on the accommodation matrix A to be synthesized, each of the form $\mathbf{u}_i \cdot A\mathbf{f}_i = t_i$.

For the user selectable target inverse-stiffnesses t_i we have used the value 1 or 0 as appropriate from equations (5), (9a) or (9b).   We find that it is not possible to satisfy all the conditions exactly. The accommodation matrix which most closely fulfills the conditions is

$$
A_{\mathbf{q}} = \begin{pmatrix} 2.91 & 0.00 & -1.94 \\ 0.00 & 1.03 & 0.00 \\ 0.00 & 0.00 & 0.00 \end{pmatrix} \tag{27}
$$

Evaluating $\mathbf{t}^{\sim} = G\, G^{*}\, \mathbf{t}$,   the available set of inverse-stiffnesses t_i nearest to our target set, we find all values within 30% of target. This is surely good enough - it would be very strange if stiffnesses more precise than this would ever be needed.   The (9b)-type conditions ($\mathbf{u}_i \cdot A\mathbf{f}_i = 0$) are satisfied exactly.

For this task the accommodation matrix synthesized cannot be described as a center of accommodation, no matter what choice of origin we use.   The result is relatively insensitive to the coefficient of friction assumed, which we took as .25.

4.3 PUT A BLOCK DOWN GENTLY

Here a block is simply to be placed on a surface. There are only two contact configurations, shown in Figure 4: either corner may touch first, and we would like the behavior of the manipulator to be such that upon contact the block rotates into alignment with the surface, without scraping along the surface.

q

1.00

p

contact configuration a
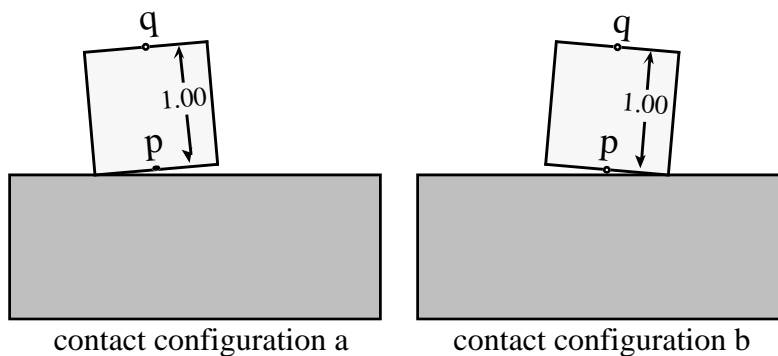
q

1.00

p

contact configuration b

Figure 4
Insert figure 4 here. Caption: Contact configurations for the put-block-down-gently
task.

Property 1 requires bounded forces.   Property 2 requires that the block rotate clockwise or counterclockwise as appropriate.   The (9b)-type conditions ($\mathbf{u_i} \cdot A\mathbf{f_i} = 0$) require that there be no horizontal movement of the corner which makes contact first: no "skidding".   Two contact configurations, by two extremes of the friction cone, by three conditions each, yield twelve conditions of the form $\mathbf{u_i} \cdot A\mathbf{f_i} = t_i$.

Using the origin $\mathbf{q}$ as shown in Figure 4 (at the top center of the block), and again using 1 or 0 as target values for all the user-selectable inverse-stiffnesses, we find that there is an accommodation matrix which satisfies all the conditions exactly:

$$A_{\mathbf{q}} = \begin{pmatrix} 2.06 & 0.00 & -2.06 \\ 0.00 & 0.52 & 0.00 \\ -2.06 & 0.00 & 2.06 \end{pmatrix} \tag{28}$$

It can easily be verified from equation (26) that this accommodation matrix is equivalent to a diagonal matrix at the bottom center of the block, meaning that our synthesis has suggested that in this case a "center of compliance" description is appropriate – with center of accommodation at the bottom center of the block (point **p**).

The equivalent accommodation matrix there is

$$A_{\mathbf{p}} = \begin{pmatrix} 0.00 & 0.00 & 0.00 \\ 0.00 & 0.52 & 0.00 \\ 0.00 & 0.00 & 2.06 \end{pmatrix} \tag{29}$$

It is relatively easy to imagine the operation of an accommodation matrix which is diagonal, and the reader may check that a center of accommodation at **p** will give the proper response to contact at either corner.

4.4 PEG INTO CHAMFERED HOLE REVISITED

Here we find 10 contact configurations, some of which are two-point contacts.   Five of these are shown in figure 5, and the other five are reflections about a vertical axis.
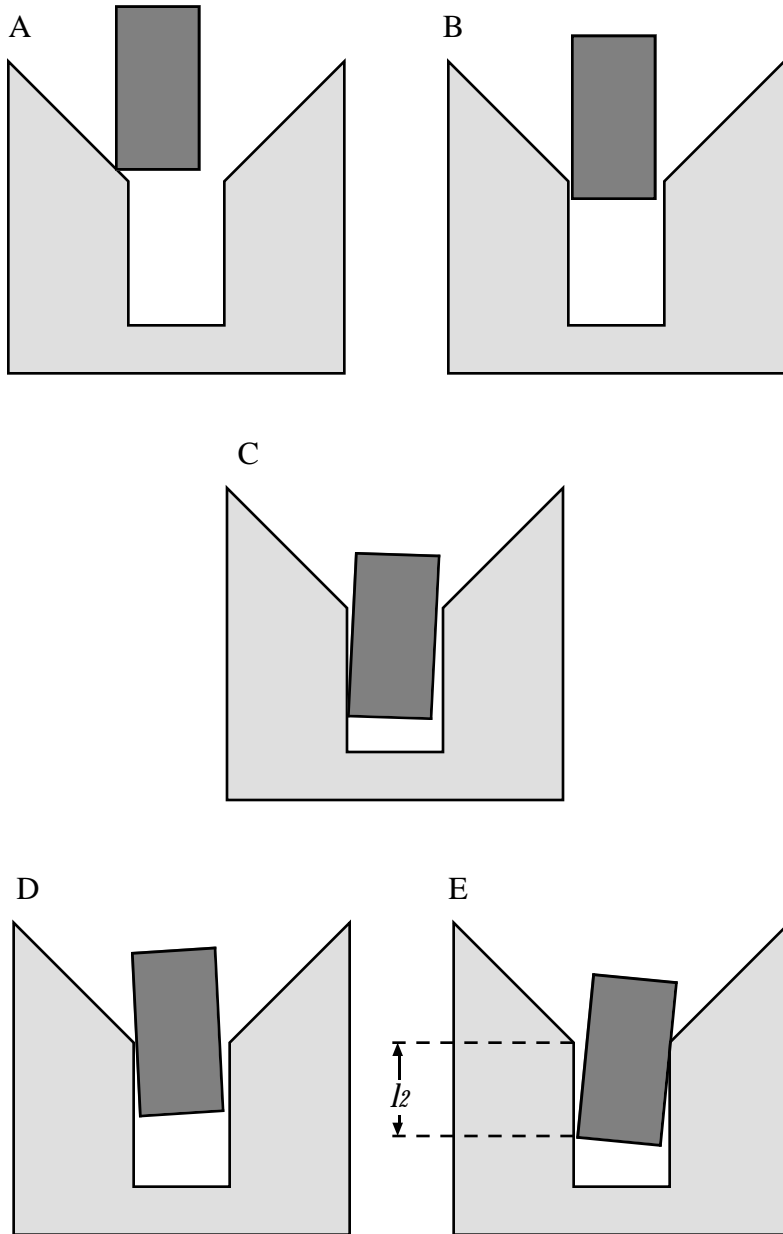
Figure 5

<u>*Insert figure 5 here. Caption:*</u>   Contact configurations for the peg-into-chamfered-hole task.   We consider these contact configurations shown, and five (not shown) which are these reflected about a vertical axis.   The nominal assembly trajectory V_o is straight down ("south"). One of the configurations (E) involves two points of contact; the rest

are single point contact configurations. To avoid confusion the normal (**n**), home (**h**), and orthogonal (**o$_i$**) directions are not shown in the figure. In configuration A, **n** is northeast **h** is southeast, and **o** is rotation. In configurations B, C, and D, **n** is east and **o** is rotation. The **h** direction is not needed because south is colinear with **V$_O$**. In configuration E, **h** is counterclockwise rotation.

It turns out that there is no accommodation matrix A which simultaneously gives the desired stiffnesses in every contact configuration.   In fact there is no way to satisfy all the (9b)-type conditions $\mathbf{u}_i \cdot A\mathbf{f}_i = 0$ even if we allow any positive value for t$_i$ in the (5) and (9a)-type conditions $\mathbf{u}_i \cdot A\mathbf{f}_i = $ t$_i$.

The accommodation matrix which comes closest in the least-squares sense is

$$A_\mathbf{q} = \begin{pmatrix} 0.66 & 0.00 & 0.19 \\ 0.00 & 1.57 & 0.00 \\ 0.20 & 0.00 & 0.35 \end{pmatrix} \tag{30}$$

based on an origin at **q** as shown in figure 5.

Interestingly, we find that $A_\mathbf{q}$ is in agreement with [Whitney, 1982 #38] results which are incorporated in the RCC device. Specifically, if (following [Whitney, 1982 #38]) we let l_2 be the insertion depth at which two point (jamming) contact may first occur, the accommodation matrix A which our optimization produces can be described as a center of accommodation at a point **p** a distance $l_2/2$ above the tip of the peg, on-axis. Rewriting A for origin at point **p** we have

$$A_\mathbf{p} = \begin{pmatrix} 0.55 & 0.00 & 0.01 \\ 0.00 & 1.57 & 0.00 \\ 0.02 & 0.00 & 0.35 \end{pmatrix} \tag{31}$$

which is (almost) diagonal.

DISCUSSION

In our view the main contribution of this paper is to pose the problem of synthesizing an error-corrective compliance for an entire task.   To the best of our knowledge this is the first attempt at a method of synthesis.

The underlying idea (common to other works in this field) is that the forces which arise during an assembly task often contain enough information to figure out how the operation has gone wrong. Forces characterize the errors in assembly tasks in a more direct way than other sensors, (e.g. vision). The question then becomes how best to interpret and act on this force information. Others researchers have (implicitly or explicitly) interpreted this information too, and a summary of those ideas in given in section 1.

Our approach may be distinguished from Mason's in that Mason synthesizes a accommodation matrix *consistent* with a task, where we synthesize one which is also *error-corrective* (terms defined in section 1.1).   It may be conceptually contrasted with most other fine motion planning work in that we assume the nominal motion plan to be given, and synthesize the accommodation matrix of the manipulator so that assembly proceeds to completion.   Most other workers have assumed the compliance of the manipulator to be given, and have synthesized a motion plan. We draw inspiration from error-corrective compliances which have been created by others for a few particular tasks, most notably the RCC. Our work differs from these in that we attempt a method of synthesis, not a solution for an individual task.

This work has several major limitations:

1. We have assumed that a finite and complete set of qualitatively distinct contact configurations has been (and can be) generated and presented to us. These configurations, together with a nominal assembly trajectory which would succeed in the absence of error, are the starting point of our synthesis.   While finding these configurations does not seem to be difficult to do by hand for simple examples, we do not know of a formal definition of "qualitatively distinct" or of a method of generating such configurations.

2. Our "error-reduction" property (section 2.1) requires strictly decreasing errors for all contact configurations as measured by *any* metric.   We have chosen a *particular* metric, and therefore might not recognize as valid an accommodation matrix which decreases errors according to some other metric[10]. Further, one might imagine compliances which do guarantee eventual success, but do not achieve it monotonically by any metric.

3. The optimization method described in section 3 does not directly use even the one particular metric we have chosen (and by which we verify the resulting A matrix).   Rather it optimizes adherence to a set of stronger conditions which, *if they are all satisfied*, would also guarantee strictly decreasing errors.   These stronger conditions are *not* always satisfied, so we do not necessarily find an accommodation matrix which guarantees strictly decreasing errors by our metric even when one exists – we are optimizing not quite the right thing.

Though our examples were two dimensional, most of the development applies immediately to three dimensions. An exception is our bounding of the friction cone by two vectors (section 3.3). It is possible to generalize the friction cone to three dimensions [Goyal, 1990 #101; Erdmann, 1984 #49].   It might also be that providing a set of qualitatively distinct contact configurations (the input to the method of

---

[10] Much work in C-space seems to be dogged by the current lack of sensible metrics.

synthesis described here) may be much more difficult in three dimensions than in two.

ACKNOWLEDGMENTS

REFERENCES