

**A GEOMETRIC APPROACH TO DERIVING
POSITION/FORCE TRAJECTORY IN FINE MOTION***

C. S. G. Lee

D. Huang

Department of Electrical Engineering and Computer Science

The University of Michigan

Ann Arbor, Michigan 48109-1109

February 1985

CENTER FOR RESEARCH ON INTEGRATED MANUFACTURING

Robot Systems Division

COLLEGE OF ENGINEERING

THE UNIVERSITY OF MICHIGAN

ANN ARBOR, MICHIGAN 48109-1109

*This work was supported in part by the AFOSR Grant F49620-82-C-0089. Any opinions, findings, and conclusions or recommendations expressed in this article are those of the authors and do not necessarily reflect the views of the funding agency.

TABLE OF CONTENTS

1. Introduction	3
2. T-C and C-PF Mapping Concepts	6
3. Geometrical Knowledge Base (GKB)	13
4. Algorithms of the T-C and C-PF Mappings	16
4.1. T-C Mapping	17
4.1.1. Deriving nominal position trajectory $P(t)$	17
4.1.2. Algorithm for Freedom Decomposition Function $S(t)$	23
4.2. C-PF Mapping	25
5. Conclusions	28
6. References	29

ABSTRACT

The synthesis of position/force trajectories for compliant motion is an important task for the control of industrial robots in assembly tasks. The position and orientation of the manipulator can be specified by a set of six linear independent constraints. Some are caused by contact with other objects (natural position constraints) and the others are applied artificially (artificial position constraints). Since the uncertainties and the modeling errors make the exact natural position constraints incomputable, a set of contact forces (artificial reaction-force constraints) is servoed to implicitly reflect their effects. Two mappings, $T-C$ and $C-PF$, are defined which derive the nominal position trajectory $P(t)$ and freedom decomposition function $S(t)$ from the attributes of a given task, and then the desired artificial position and reaction-force constraints. In the $T-C$ mapping, for an insertion task, the convex operand shrinks to a line and the three 2-D cut-diagrams of the concave operand are derived and expanded correspondingly. The nominal position trajectory is determined from these expanded cut-diagrams. Three 2-D trapezoids are defined to model the uncertainties. The intersections between the trapezoids and the cut-diagrams are checked to determine the values of freedom decomposition function $S(t)$. Consequently, in the $C-PF$ mapping the time sequences of the artificial position constraints and artificial reaction-force constraints are derived from the nominal position trajectory and the freedom decomposition function.

1. Introduction

Most manipulator tasks will likely require some form of compliant motion control which is defined as the ability to modify the manipulator motion based on sensed contact forces during the motion. Compliant motion is a common phenomenon in robot tasks and occurs when the position of a manipulator is constrained by the task geometry. Previous work on active compliant motion control focused on controlling the manipulator to follow the desired position/force trajectories as closely as possible and can be divided into two categories: hybrid position/force control [7-9] and explicit feedback control [5,11].

The hybrid position/force control consists of separate position and force control loops. For a given task, the input to the controller is the desired position and force trajectory set points in the C-frame[†]. The robot motions in the C-frame are partitioned into position and force constraint subsets and the controller servos each degree of motion freedom, position or force, by a feedback loop. Unlike the hybrid position/force control, the explicit feedback control approach uses a force control law which relates the deviation of the hand position or velocity to the required force applied to the hand. Both methods require correct synthesis of the position and/or force trajectory which will lead the

[†] C-frame: Compliance frame [8-9] is an orthogonal coordinate system in the Cartesian space. The frame is so chosen that the task freedoms are defined to be translation along and rotation about each of the three principal axes.

robot to complete its assembly task. This paper addresses the problem of generating position/force trajectory for the hybrid position/force control scheme.

Current manipulator trajectory planning methods focused on position trajectories [1,5,6]. For compliant motion control, a force trajectory is required as well. Mason [7] developed a formal approach to derive velocity and force constraints from a C-surface[‡], not the partitioned position and force trajectory set points for the hybrid position/force controller. Since the C-surface is task dependent, his method is not completely general.

For any robot task no matter whether it is a gross motion or compliant motion, there exists an *ideal nominal position trajectory* which must be followed by the manipulator to complete the task. This trajectory can be viewed as a set of six linear independent constraints to eliminate the six degrees of freedom in a six-dimensional configuration space which consists of all possible positions and orientations of the manipulator. These position constraints can be partitioned into two subsets of constraints. They are *natural position constraints* which occur due to contacting with other objects and *artificial position constraints* which are applied artificially. Due to uncertainties and errors present in the modeling of the objects and the manipulator, obtaining the natural position constraints exactly may be impossible. However, natural position constraints can be indirectly represented by *artificial reaction-force constraints* because contact is always associated with reaction force. Therefore, the task of designing a

[‡] C-surface is defined on a C-frame. It is a task configuration which allows only partial position freedoms. Along its tangent is the positional freedom and along its normal is the force freedom [7].

position trajectory which will lead the robot to complete an assembly task can be decomposed into defining two subsets of constraints, artificial position constraints and artificial reaction-force constraints. Their combined effect to the controller should be equivalent to that of the ideal nominal position trajectory.

This paper extends Lozano-Perez's shrink-expand approach [6] and combines it with a 2-D geometric cut-diagram approach to derive a 3-D position/force trajectory for the hybrid position/force control scheme. The problem of automatic synthesis of position/force trajectory can be expressed as a mapping $T-PF$ (task to position/force mapping) which in turn consists of two submappings. They are $T-C$ submapping (task to configuration space mapping) which derives a nominal position trajectory $P(t)$ and a freedom decomposition function $S(t)$ from the attributes of a given task and $C-PF$ submapping (configuration space to position/force trajectory mapping) which derives the artificial position constraints and the artificial reaction-force constraints from the $P(t)$ and $S(t)$.

In the $T-C$ mapping, for an insertion task, an algorithm similar to Lozano-Perez's shrink-expand approach is used to shrink the convex object grasped by the manipulator into a line with one of its end points at the origin of its body attached frame. This origin is called the *nominal center*. Then, the 3-D nominal trajectory planning is done on the three 2-D cut-diagrams of the other relevant object. The uncertainties cause a larger region of possible contact. These uncertainties are represented by three 2-D trapezoids. By checking the intersections between the trapezoids and the three 2-D cut-diagrams, a piecewise

constant vector function $S(t)$ is determined which represents the partition of the six position constraints into artificial position constraints and natural position constraints. $S(t)$ as well as the nominal position trajectory $P(t)$ serve as inputs to the $C-PF$ submapping which is responsible for determining the artificial position and reaction-force constraints. In the $C-PF$ submapping, a time sequence of every element of the artificial position constraints and artificial reaction-force constraints is determined according to the corresponding element of the freedom decomposition function $S(t)$. The outcome of this mapping is a time sequence of artificial position constraints $P_a(t)$ and artificial reaction-force constraints $F_a(t)$ which are ready to serve as reference inputs to the position/force servo loops.

In the next section, the concepts of $T-C$ and $C-PF$ mappings are defined and discussed. A geometrical knowledge base which provides the geometric information of the objects in the workspace to the $T-C$ mapping is briefly described in Section 3. Then the algorithms of $T-C$ and $C-PF$ mappings are described in Section 4, followed by conclusions in Section 5.

2. T-C and C-PF Mapping Concepts

In this section we formulate the problem of generating position/force trajectory for the hybrid controller by two mappings ($T-C$ and $C-PF$ mappings) and introduce some related concepts.

A configuration space is a generalized coordinate system used to specify certain entity. Therefore the position and orientation of an arbitrary object in a

3-D space can be specified by a six-tuple $P = (p_x, p_y, p_z, q_x, q_y, q_z)$ in a 6-D configuration space \mathbf{C} , where $p_x, p_y,$ and p_z are displacement along the $X, Y,$ and Z axes, respectively, of the base frame and $q_x, q_y,$ and q_z are angular displacement about the the principal axes of a frame whose axes are parallel to the base frame and whose origin is at (p_x, p_y, p_z) . In other words, the six orthogonal constraints specify the position and orientation of an object in a 3-D space.

Similarly forces and torques in the configuration space \mathbf{C} in the base frame can be represented by a six-tuple $F = (f_x, f_y, f_z, \tau_x, \tau_y, \tau_z)$ where $f_x, f_y,$ and f_z are the forces along the $X, Y,$ and Z axes, respectively, of the base frame and $\tau_x, \tau_y,$ and τ_z are the torques about the principal axes of a frame whose axes are parallel to the base frame and whose origin is at (p_x, p_y, p_z) . All these possible six-tuples form another configuration space, called force configuration space \mathbf{C}_f . However, it is not independent and is associated with the configuration space \mathbf{C} .

The following definitions are important in discussing the $T-C$ and $C-PF$ mappings.

Position freedom in configuration space: A degree of configuration space is said to possess position freedom if the motion along this degree (i.e. the motion along or about the corresponding principal axis of the base frame) is free.

Reaction-force freedom in configuration space: A degree of configuration space is said to possess reaction-force freedom if the reaction-force along this degree

(i.e. the reaction-force along or the reaction-torque about the corresponding principal axis of the base frame) can take any value within some assumed domain.

Natural position constraint: A position freedom of degree in configuration space of an object under consideration will be eliminated by contact with other objects, or in other words, constrained by the task geometry. This constraint is called natural position constraint on the object, while the object is said to be natural position constrained on the degree of configuration space.

Artificial position constraint: A constraint artificially applied to a degree of configuration space of an object under consideration to eliminate the position freedom of that degree. The object is said to be artificial position constrained on the degree of configuration space.

Natural reaction-force constraint: A reaction-force freedom of configuration space of an object under consideration will be eliminated by absence of contact on that degree with other objects. The corresponding reaction force can not take any arbitrary value besides zero or certain value of friction. This constraint is called the natural reaction-force constraint on the object, while the object is said to be natural reaction-force constrained on that degree of configuration space.

Artificial reaction-force constraint: A constraint artificially applied to a degree of configuration space of an object under consideration to eliminate the reaction-force freedom of that degree. The object is said to be artificial

reaction-force constrained on the degree of configuration space.

For every robot task, there exists a time sequence of six orthogonal constraints in the configuration space, which specifies the positions and orientations where the manipulator must be if the task is successfully executed. This time sequence of constraints is called *ideal nominal position trajectory*. If we could derive such trajectory by some algorithms and we had an ideal controller which could drive the manipulator without any error, then the robot could perform any assembly task as precisely as possible. Unfortunately, with uncertainties and modeling errors, it is impossible to derive such an ideal nominal position trajectory from a set of uncertain data. What we can do is to derive a time sequence of six constraints in the configuration space based on the available uncertain data to approximate the ideal nominal position trajectory. The trajectory resulted from this approximation is called *nominal position trajectory*.

At any time t , during the execution of a robot task, some of these required six position constraints are applied by the task geometry. They are the natural position constraints. Generally, the number of linear independent natural position constraints is less than six. Several artificial position constraints should be selected to constrain the remaining degrees of position freedom. The total number of natural position constraints, denoted as N_c , and the total number of artificial position constraints, denoted as N_a , satisfy the following relations:

$$0 \leq N_c \leq 6 \quad \text{and} \quad N_a = 6 - N_c$$

When $N_c = 0$, it implies gross motion; while when $N_c = 6$, it implies that the

grasped object is not movable. In an ideal case, these artificial and natural position constraints should form the ideal nominal position trajectory and serve as reference input to the robot controller. Since artificial position constraints are designed by the operator and are known exactly in advance, they can serve as the reference input to the controller easily. However, natural position constraints exist naturally and can be detected only with certain degree of uncertainty, which disqualifies them to be the reference input. This is the reason why the ideal nominal position trajectory can not be derived exactly. A solution to this difficulty is to indirectly servo the natural position constraints. Since the natural position constraints are caused by contact with other objects and the contact is always associated with reaction force, the reaction force along the degree of configuration space, which is natural position constrained, can indirectly provide natural position constraints to the controller. The desired reaction force is the artificial reaction-force constraint.

The artificial position constraints can be extracted from the nominal position trajectory which is an approximation to the ideal nominal position trajectory. And the rest of the nominal position trajectory can be considered as an approximation of the natural position constraints and can be mapped to the desired artificial reaction-force constraints by a proper mapping. Since the same contact can generate different contact forces, the relation from natural position constraints to artificial reaction-force constraints is of one-to-many. There is a class of sets of artificial reaction-force constraints which can serve as an indirect input to replace the natural position constraints. This multiplicity makes it

possible to substitute the artificial reaction-force constraints derived from the nominal position trajectory for those from the ideal nominal position trajectory. In most cases, force control is used as an indirect position control to make the actual position trajectory of the manipulator to approach the incomputable ideal nominal position trajectory.

Let us now formulate the above concepts by two mappings. Let

\mathbf{C} = a six-dimensional configuration space, based on a Cartesian frame, B-frame for short.

$P(t)$ = a point in \mathbf{C} at time t . It is a six-tuple $(p_x(t), p_y(t), p_z(t), q_x(t), q_y(t), q_z(t))$.

\mathbf{S} = the set of all possible $S(t)$ which is a freedom decomposition function, $S(t) \in \mathbf{S}$.

\mathbf{T} = the set of all possible T which is a task, $T \in \mathbf{T}$.

\mathbf{C}_{np} = a subspace of \mathbf{C} , which consists of all those degrees in \mathbf{C} which are natural position constrained with the rest of the degrees having zero elements, e.g. $\mathbf{C}_{np} = \{(p_x, p_y, 0, q_x, q_y, 0) \mid p_x, p_y, q_x, q_y \in R\}$.

$P_n(t)$ = a point in \mathbf{C}_{np} , which represents the natural position constraints.

$\mathbf{C}_{ap} = (\mathbf{C}_{np})^\perp$, the orthogonal complement of \mathbf{C}_{np} . It consists of all those degrees in \mathbf{C} which are artificial position constrained with the rest of the degrees having zero elements, e.g.

$\mathbf{C}_{ap} = \{(0, 0, p_z, 0, 0, q_z) \mid p_z, q_z \in R\}$.

$P_a(t)$ = a point in \mathbf{C}_{ap} , which represents the artificial position constraints.

C_f = a six-dimensional force configuration space associated with C .

$F(t)$ = a point in C_f at time t . It is a six-tuple
 $(f_x(t), f_y(t), f_z(t), \tau_x(t), \tau_y(t), \tau_z(t))$.

C_{af} = a subspace of C_f consisting of all those degrees which are artificial reaction-force constrained with the rest of the degrees having zero elements, e.g. $C_{af} = \{(0, 0, f_z, 0, 0, \tau_z) \mid f_z, \tau_z \in R\}$.

The problem of automatic synthesis of position/force trajectory can be expressed as a mapping T - PF :

$$T-PF: T \rightarrow C_{ap} \times C_{af}$$

It consists of two submappings. The first is called T - C submapping which derives a nominal position trajectory $P(t)$ (not the ideal nominal trajectory, which can not be derived exactly) and a freedom decomposition function $S(t)$ from the attributes of a given task. The second is the C - PF submapping which derives the artificial position constraints $P_a(t)$ and the artificial reaction-force constraints $F_a(t)$ from $P(t)$ and $S(t)$. That is:

$$T-PF = C-PF \circ T-C$$

$$T-C: T \rightarrow C \times S$$

$$C-PF: C \times S \rightarrow C_{ap} \times C_{af}$$

where \circ denotes the left composition of functions. Notice that $P(t)$ can be uniquely represented as a sum of two points $P_n(t) \in C_{np}$ and $P_a(t) \in C_{ap}$ because $(C_{np})^\perp = C_{ap}$ in the space C . Obviously, $P_n(t)$ is the natural position

- (1) The combinational operators used in this GKB are $+^*$ and $-^*$ which are the regularized set union and difference, respectively, of the sets of points in Cartesian space, that is,

$$A +^* B = \text{closure}(\text{interior}(A \cup B))$$

$$A -^* B = \text{closure}(\text{interior}(A - B))$$

where A, B are sets of connected points in a Cartesian space, $\text{closure}(A)$ is the closure of the set A and $\text{interior}(A)$ is the interior of the set A .

- (2) Every solid feature of an object is prefixed by a $+^*$, while every hollow feature is prefixed by a $-^*$.
- (3) All the primitive solids are convex. Therefore an instance of a primitive solid or a set of them connected by $+^*$'s represents a feature itself if it is a solid one, or the solid one which has the same boundary as the feature if it is a hollow one (see Figure 1).

When a task command, which is represented by the five attributes, enters the geometrical knowledge base, it will provide the following data for the $T-PF$ mapping:

- (1) operand = a subset of features in the object with one of the features being the feature-operand and with the rest being those directly connected with the feature-operand.
- (2) B-frame = the frame which is the base frame of the configuration space C in the given task. Generally it is the body attached frame of the feature.

operand which is fixed during the execution of the task and is represented by a 4×4 HTM H_{c_w} relating the B-frame to the global reference frame.

- (3) $P(0)$ = the initial state of the body attached frame of the operand to be moved with respect to the B-frame.
- (4) $P(T)$ = the goal state of the body attached frame of the operand to be moved with respect to the B-frame.
- (5) $H_{ow_1}(0)$ and $H_{ow_2}(0)$ = the HTMs which represent the body attached frames of the operands.

All the objects present in the assembly workstation are constructed as a graph. Each object or its feature is linked to another through the lower pair relation or some other relations, if they exist. The GKB will be modified immediately after the completion of a subtask. This involves the modification of attachment relations and the various homogeneous transformation matrices of the objects or features involved.

4. Algorithms of the T-C and C-PF Mappings

This section describes algorithms for the $T-C$ and $C-PF$ mappings based on the concepts developed in the previous sections. However only cylindrical and prismatic insertions are dealt with here because they form a majority of assembly tasks. Figure 2 gives an example of a peg-hole insertion task.

constraints given by the task geometry and $P_a(t)$ is the corresponding artificial position constraints. Furthermore, every point $P_n(t)$ in C_{np} has an associated subset $F_a(t) \subseteq C_{af}$. Each element $F_a(t) \in F_a(t)$ specifies a set of reaction forces (torques) along the degrees which are natural position constrained in the configuration space.

3. Geometrical Knowledge Base (GKB)

In this paper, a given task is represented by the following attributes:

1. Feature-Operand = key names of the features relevant in the task.
2. Grasp position = a homogeneous transformation matrix H_{ho} relating the hand frame to the body attached frame of the grasped operand.
3. Goal state = $P(T)$, a six-dimensional vector in the configuration space. Some of its elements can be "don't care." In this way, the goal state actually expands to a subset in the configuration space C and no longer a single point.
4. Attachment type = relation between two feature-operands, it may be one of the lower pairs: cylindrical, prismatic, planar, screw, spherical, and revolute.
5. Specification = a miscellaneous part, which may specify the speed of the hand or magnitude of contact force or torque, etc.

Other relevant information about the task and its environment are supplied by the GKB.

The "geometrical knowledge" are represented by the constructive solid geometry (CSG) scheme associated with primitive instancing. In this representation scheme, every object is represented by a tree. The root is the key name of the object (parts) and a pointer linking to a homogeneous transformation matrix (HTM) relating its local coordinate frame to the global reference coordinate frame. The second level non-terminal nodes are the key names of the feature classification (e.g. holes, cubes, etc). The third level non-terminal nodes are the key names of the features of the object (e.g. hole A, slot B, etc) and a pointer linking to a HTM relating the body attached frame of the feature to the object's local coordinate frame. The fourth level non-terminal nodes, if exist, are the combinational operators[†]. The terminal nodes are the instances of the primitive solids. They represent the features of the object by themselves or through combinational operators. A particular instance of the primitive solid is indicated by the key name of the solid and a tuple of specified parameters. The location of the body attached frame will be defined according to these rules:

- (1) The frame is always defined on the central line of the feature with one of the principal axes parallel to the central line.
- (2) For all concave features, there is a "mouth." The Z-axis of the frame should pass through the mouth.

The combination of the instances of primitive solids follows the following rules:

[†] Combinational operators are the regularized Boolean operators operating on primitive solids [10].

4.1. T-C Mapping

The objective of the $T-C$ mapping is to derive nominal position trajectory $P(t)$ and freedom-decomposition function $S(t)$ from the attributes of a given task. In an insertion task, there are two feature-operands, one is solid and the other is hollow. In the following discussion, we name the operand which contains the solid feature-operand as *operand1* and the other which contains the hollow one as *operand2*. Furthermore, we assume that *operand1* is grasped by the manipulator and is to be moved in completing the task, while *operand2* is fixed in the workspace. The body attached frame of the hollow feature operand is selected as the B-frame. If these assumptions can not hold true for some insertion tasks, then appropriate transformations can be found to transform these tasks into their equivalent counterparts which satisfy the assumptions.

4.1.1. Deriving nominal position trajectory $P(t)$

The 3-D nominal position trajectory $P(t)$ is obtained by three coherent but separate steps. First, the three 2-D cut-diagrams of *operand2*, G_{xz} , G_{yz} , and G_{xy} , on the $X-Z$, $Y-Z$, and $X-Y$ planes, respectively, of its body attached frame H_{ow2} are computed. Second, *operand1* shrinks to a line L (with one of the principal axes of its body attached frame parallel to the Z -axis of H_{ow2} at the goal state), while G_{xz} , G_{yz} , and G_{xy} expand correspondingly to GE_{xz} , GE_{yz} , and GE_{xy} , respectively. Third, a 3-D nominal position trajectory is planned on each $U-V$ plane ($uv = xz, yz, xy$) with the criterion

that L_{uv} representing operand1 will not intersect with GE_{uv} representing operand2.

Step 1: Find the cut-diagrams of the concave operand. Due to the representation of objects in the GKB by the CSG scheme, the task of computing the cut-diagrams of operand2 in its $X-Z$, $Y-Z$, and $X-Y$ planes can be decomposed into computing the cut-diagrams of its component primitive solids by divide-and-conquer approach. The final result is obtained by recombining these sub-cut-diagrams using the same CSG regularized operators. The divide-and-conquer approach greatly reduces the computational complexities of computing the the three cut-diagrams of operand2 and can be described by the procedure CUT_DIA.

procedure CUT_DIA (operand2)

If (operand2 = single instance of a primitive solid)

then

$G_{xz}, G_{yz}, G_{xy} \leftarrow$ cut diagrams of operand2 on XZ, YZ, XY planes, respectively.

else

divide operand2 into $s1 \text{ op } s2$, where $s1$ and $s2$ are sub-objects with almost equal number of instances of primitive solids.

$G_{xz}, G_{yz}, G_{xy} \leftarrow$ CUT_DIA ($s1$) op CUT_DIA ($s2$)

end_if

return (G_{xz}, G_{yz}, G_{xy})

end_procedure

where $op \in \{ +^*, -^* \}$

The correctness of the divide-and-conquer procedure can be proved by induc-

tion (see Appendix A). The resultant cut-diagrams are denoted as:

G_{uv} = the cut-diagrams of the operand2 on the $U-V$ plane

$$= c_{uv} B_1 op_1 c_{uv} B_2 op_2 \cdots op_{n-1} c_{uv} B_n$$

where B_m = instance of primitive solids, $m = 1, 2, \cdots, n$, $op_i \in \{ +^* , -^* \}$,

and $c_{uv} B_m$ = the cut-diagrams of the instance B_m on the $U-V$ plane,

represented by a set of parametric function pairs:

$$\left\{ \begin{array}{l} (u_1 = f_{11}(t_1) , v_1 = f_{12}(t_1)) \\ (u_2 = f_{21}(t_2) , v_2 = f_{22}(t_2)) \\ \dots \\ (u_K = f_{K1}(t_K) , v_K = f_{K2}(t_K)) \end{array} \right.$$

where $uv = zz , yz , xy$. The f_{ij} , $i = 1, 2, \cdots, K; j = 1, 2$ are so selected that when one point travels along the curve with increasing t_i the interior of B_m lies on the left of the point.

Step 2: Shrink-expand on operand1 and the cut-diagrams. In this step, the operand1 is shrunk into a line which is coincident with one of the principal axes of its body attached frame $H_{ow1}(T)$ and parallel to the Z -axis of H_{ow2} . The line L is then projected onto the $X-Z$, $Y-Z$ and $X-Y$ planes of H_{ow2} as L_{zz} , L_{yz} , and L_{xy} , respectively. Then, the cut-diagrams of operand2, G_{zz} , G_{yz} , and G_{xy} expand correspondingly to GE_{zz} , GE_{yz} , and GE_{xy} , respectively. Like operand2, operand1 may consist of several instances of primitive solids combined by regularized set operators. Hence it can be represented as

$$operand1 = A_1 op_1 A_2 op_2 \cdots op_{m-1} A_m$$

and the shrinking of operand1 can be decomposed into the shrinking of A_i 's too.

Consequently, the expansion of operand2 can be decomposed into m sub-expansion of G_{zz} , G_{yz} , and G_{xy} corresponding to the shrinking of A_i 's, $i = 1, 2, \dots, m$ so that the computational complexities can be greatly reduced. The final expanded cut-diagrams corresponding to operand1, denoted as GE_{zz} , GE_{yz} , and GE_{xy} , are defined in the following assertion:

Assertion 1.

Assume $operand1 = A_1 op_1 A_2 op_2 \dots op_{m-1} A_m$

Then $GE_{uv} = \bigcup_{i=1}^m GE_{iuv}$; $uv = zz, yz, xy$

where GE_{iuv} is the expanded G_{uv} corresponding to A_i , $uv = zz, yz, xy$. The proof of Assertion 1 can be found in Appendix B. Based on above discussion, we define a divide and conquer procedure SHR_EXP to implement the shrink-expand task with details omitted.

procedure SHR_EXP (G_{uv} , operand1)

If (operand1 = instance of a primitive solid)

then

$GE_{uv} \leftarrow$ expansion of G_{uv} according to the shrinking of operand1;

else

divide operand1 into $C_1 op C_2$, where C_1 and C_2 are sub-
lids with almost equal number of instances primitive solids.

$GE_{uv} \leftarrow SHR_EXP(G_{uv}, C_1) \cup SHR_EXP(G_{uv}, C_2)$

end_if

return GE_{uv}

end_procedure

Step 3: Find nominal position trajectory. In this step the nominal position trajectory $P(t) = (p_x(t), p_y(t), p_z(t), q_x(t), q_y(t), q_z(t))$ defined with respect to the frame H_{ow2} is to be obtained from the cut_diagrams GE_{xz} , GE_{yz} , and GE_{xy} . These cut-diagrams provide the information of natural position constraints. Deriving $P(t)$ from these cut-diagrams consists of four phases:

- (1) Define a set of position nodes, $\{N_1, N_2, \dots, N_T\}$, on a 3-D Euclidean space as well as their projections on the $X-Z$, $Y-Z$, and $X-Y$ planes, $\{N_{1uv}, N_{2uv}, \dots, N_{Tuv}\}$, $uv = xy, yz, xz$ such that for all N_{iuv} : $N_{iuv} \notin GE_{uv}$ and N_i is visible[†] to both N_{i-1} and N_{i+1} , $i = 2, \dots, T-1$, and $N_1 = \text{origin}(H_{ow1}(0))$, $N_T = \text{origin}(H_{ow1}(T))$; where $\text{origin}(H) =$ the origin of frame H in the B-frame.
- (2) Define a set of straight lines, $\{(N_1, N_2), (N_2, N_3), \dots, (N_{T-1}, N_T)\}$, which forms the nominal position trajectory of the nominal center of operand1. This trajectory is the position part $(p_x(t), p_y(t), p_z(t))$ of the $P(t)$.
- (3) Define a set of orientation nodes corresponding to the position nodes, $\{M_1, M_2, \dots, M_T\}$, where $M_i = (q_x(t), q_y(t), q_z(t))$. These orientation nodes are defined in such a way that with the nominal center at $(p_x(t), p_y(t), p_z(t))$, the line L_{xz} , L_{yz} , and L_{xy} will not intersect with the

[†] N_i is said to be visible to N_{i-1} if there exists a straight line from N_{i-1} to N_i which will not overlap with GE_{uv} .

GE_{xz} , GE_{yz} , and GE_{xy} , respectively, and with $M_1 = (q_x(0), q_y(0), q_z(0))$, the initial orientation of operand1 with respect to the B-frame (H_{ow2}) and $M_T = (q_x(T), q_y(T), q_z(T))$, the orientation of the goal state with respect to the frame H_{ow2} .

- (4) Define a set of straight lines in angular motion, $\{(M_1, M_2), (M_2, M_3), \dots, (M_{T-1}, M_T)\}$, which forms the orientation part of the nominal position trajectory $P(t), (q_x(t), q_y(t), q_z(t))$.

In this paper we do not consider the collision between operand1 and other obstacles besides operand2. Under this assumption, we select only three nodes for each insertion task: $N_1 = (p_x(0), p_y(0), p_z(0))$, initial position of the nominal center with respect to the B-frame; $N_2 = (p_x(t_1), p_y(t_1), p_z(t_1))$, a point on the Z-axis of the frame H_{ow2} which is visible to N_1 ; and $N_T = (p_x(T), p_y(T), p_z(T))$, the goal state of the nominal center with respect to the frame H_{ow2} . It is desired to have $(q_x(t), q_y(t), q_z(t))$ reach its goal state $(q_x(T), q_y(T), q_z(T))$ before the nominal center reaches N_2 . The resultant nominal position trajectory can be expressed as a piecewise linear vector function, $P(t) = (p_x(t), p_y(t), p_z(t), q_x(t), q_y(t), q_z(t))$, where

$$p_i(t) = \begin{cases} p_i(0) + \frac{t}{t_1} (p_i(t_1) - p_i(0)) & ; N_1 \rightarrow N_2 \\ p_i(t_1) + \frac{t-t_1}{T-t_1} (p_i(T) - p_i(t_1)) & ; N_2 \rightarrow N_T \end{cases}$$

$$q_i(t) = \begin{cases} q_i(0) + \frac{t}{t_1} (q_i(T) - q_i(0)) & ; M_1 \rightarrow M_2 \\ q_i(T) & ; M_2 \rightarrow M_T \end{cases}$$

where $i = x, y, z$.

4.1.2. Algorithm for Freedom Decomposition Function S(t)

Strictly, if the line L_{uv} which represents operand1 on the $U-V$ plane of the frame H_{ov2} is on the boundary of GE_{uv} (i.e. touch with one point or whole line, but not intersect with), then operand1 is under natural position constraints at some degree of the configuration space. Even in some region where L_{uv} does not contact with GE_{uv} , the natural position constraints may still exist. This is due to the uncertainties which will bring operand1 and operand2 in contact if the free space between operand1 and operand2 is not big enough to tolerate the uncertainties. These uncertainties will cause operand1 to deviate from $P(t)$. On the $U-V$ plane, the result is that the line L_{uv} will deviate from its nominal position and orientation. In order to handle these uncertainties, we use a ball B_{uvr} to represent the uncertainty of the position of the nominal center and a cone $C_{uv\theta}$ to represent the uncertainty of the orientation of L_{uv} on the $U-V$ plane. They are combined into an uncertainty trapezoid, T_{uvrs} .

Let $\Phi_0 = (u_0, v_0)$ be the nominal position of the nominal center and $\Phi = (u, v)$ be the position of the nominal center. Then the uncertainty ball of the nominal center is defined to be

$$B_{uvr}[\Phi_0] = \{ \Phi \in U-V \text{ plane: } d_{\infty}(\Phi, \Phi_0) \leq r \}$$

where r is the radius of the ball and $d_{\infty}(\Phi, \Phi_0)$ is a metric function defined on the $U-V$ plane,

$$d_{\infty}(\Phi, \Phi_0) = \max\{ |u-u_0|, |v-v_0| \}$$

Let Ψ_0 be the nominal angular displacement of L_{uv} on the $U-V$ plane and Ψ be the angular displacement of L_{uv} . Then the uncertainty cone of the orientation of L_{uv} is defined to be

$$C_{uv\theta}[\Psi_0] = \{ \Psi \in [0, 2\pi): d_1(\Psi, \Psi_0) \leq \theta \}$$

where θ is a small positive angle and

$$d_1(\Psi, \Psi_0) = | \Psi - \Psi_0 |$$

The uncertainty trapezoid $T_{uvrs}[\Phi_0, \Psi_0]$ is the combination of $B_{uvr}[\Phi_0]$ and $C_{uv\theta}[\Psi_0]$ and is given as

$$T_{uvrs}[\Phi_0, \Psi_0] = \{ L_{uv}(\Phi, \Psi): \Phi \in B_{uvr}[\Phi_0] \text{ and } \Psi \in C_{uv\theta}[\Psi_0] \}$$

where $L_{uv}(\Phi, \Psi)$ represents the line L_{uv} with its nominal center at Φ and angular displacement of Ψ . Figure 3 gives illustrations of $B_{uvr}[\Phi_0]$, $C_{uv\theta}[\Psi_0]$, and $T_{uvrs}[\Phi_0, \Psi_0]$.

Obviously when $L_{uv}(\Phi_0, \Psi_0)$ moves along the nominal position trajectory, some $L_{uv}(\Phi, \Psi) \in T_{uvrs}[\Phi_0, \Psi_0]$ might contact, or even intersect, with GE_{uv} . Therefore operand1 is natural position constrained at some degree of configuration space. The freedom decomposition function $S(t)$ reflects such occurrence of natural position constraints and is defined as

$S(t) = (s_1(t), s_2(t), s_3(t), s_4(t), s_5(t), s_6(t))$, where

$$s_i(t) = \begin{cases} -1, & \text{case 1} \\ 0, & \text{case 2} \\ 1, & \text{case 3} \\ 2, & \text{case 4} \end{cases}$$

where:

case 1 = natural position constraint occurs on the direction against the i^{th} degree of the configuration space,

case 2 = no natural position constraint occurs on the i^{th} degree of the configuration space,

case 3 = natural position constraint occurs on the direction along the i^{th} degree of the configuration space,

case 4 = natural position constraint occurs at both directions along and against the i^{th} degree of the configuration space.

The freedom decomposition function $S(t)$ is determined by checking the intersections between the both edges of T_{uv} and GE_{uv} with $uv = zz, yz$, and xy planes. Then $s_i(t)$'s are assigned with proper values according to the intersection checking. The resultant $S(t)$ is a piecewise constant function which reflects the different subtask periods of the given task, e.g. gross motion subtask, compliant motion subtask, etc.

4.2. C-PF Mapping

The *C-PF* mapping is to derive the artificial position constraints and the artificial reaction-force constraints from the nominal position trajectory $P(t)$

and freedom decomposition function $S(t)$. These constraints serve as the input to the hybrid controller. In general, the configuration space is represented as $C = \{(p_x, p_y, p_z, q_x, q_y, q_z) \mid p_x, p_y, p_z, q_x, q_y, q_z \in R\}$ and C_{np} , C_{ap} , and C_{af} vary with the task. For instance, in the insertion task shown in Figure 2, the C_{np} , C_{ap} , and C_{af} subspaces are represented as:

$$C_{np} = \begin{cases} \{(0, 0, 0, 0, 0, 0)\} & ; t \in [0, t_1] \\ \{(p_x, p_y, 0, q_x, q_y, 0) \mid p_x, p_y, q_x, q_y \in R\} & ; t \in [t_1, T] \end{cases}$$

$$C_{ap} = \begin{cases} C & ; t \in [0, t_1] \\ \{(0, 0, p_z, 0, 0, q_z) \mid p_z, q_z \in R\} & ; t \in [t_1, T] \end{cases}$$

$$C_{af} = \begin{cases} \{(0, 0, 0, 0, 0, 0)\} & ; t \in [0, t_1] \\ \{(f_x, f_y, 0, \tau_x, \tau_y, 0) \mid f_x, f_y, \tau_x, \tau_y \in R\} & ; t \in [t_1, T] \end{cases}$$

At any time t , a point $P^*(t)$ on the ideal nominal position trajectory is the sum of the two points $P_n^*(t) \in C_{np}$ and $P_a^*(t) \in C_{ap}$. Hence, the ideal nominal position trajectory $P^*(t)$ can be uniquely represented as the sum of the time sequences of artificial position constraints $P_a^*(t)$ and natural position constraints $P_n^*(t)$, $P^*(t) = P_a^*(t) + P_n^*(t)$.

The first step of C -PF mapping is to derive the time sequence of constraints, $P_a(t) \in C_{ap}$, from the nominal position trajectory $P(t)$. Since the artificial position constraints are artificially selected, it will not be affected by uncertainties. Therefore $P_a(t)$ extracted from $P(t)$ is equivalent to $P_a^*(t)$,

$$P_a(t) = (\Gamma(s_1(t), p_x(t)), \Gamma(s_2(t), p_y(t)), \Gamma(s_3(t), p_z(t)), \\ \Gamma(s_4(t), q_x(t)), \Gamma(s_5(t), q_y(t)), \Gamma(s_6(t), q_z(t)))$$

where:

$$\Gamma(s(t), p(t)) = \begin{cases} p(t) & t \in [t_1, t_2) \text{ and } s(t) \equiv 0 \\ 0 & \text{otherwise} \end{cases}$$

and $s(t)$ is an element of $S(t)$ and $p(t)$ is the corresponding element of $P(t)$.

The next step of $C-PF$ mapping is to derive the time sequence of artificial reaction-force constraints $F_a(t)$. Since the relation from the natural position constraints to artificial reaction-force constraints is of one-to-many, $F_a(t)$ actually is only the function of freedom decomposition function $S(t)$. Therefore $F_a(t)$ can implicitly represent the natural position constraints part of the ideal nominal position trajectory and defined to be

$$F_a(t) = (\Omega(s_1(t)), \Omega(s_2(t)), \Omega(s_3(t)), \Omega(s_4(t)), \Omega(s_5(t)), \Omega(s_6(t)))$$

where:

$$\Omega(s_i(t)) = \begin{cases} \pm f_i & ; t \in [t_1, t_2) \text{ and } s_i(t) \equiv 2 \\ -f_i & ; t \in [t_1, t_2) \text{ and } s_i(t) \equiv 1 \\ +f_i & ; t \in [t_1, t_2) \text{ and } s_i(t) \equiv -1 \\ 0 & ; t \in [t_1, t_2) \text{ and } s_i(t) \equiv 0 \end{cases}$$

and f_i is the desired contact force either given by the task attributes or an arbitrary small positive number.

The final outcome of the $T-PF$ mapping serves as the reference input to the hybrid position/force controller with the artificial position constraints served by the position feedback loops and artificial reaction-force constraints served by the force feedback loops.

5. Conclusions

In this paper, the concepts of compliant motion is developed and discussed. Based on the concepts, Lozano-Perez's shrink-expand approach is combined with a 2-D cut-diagram approach to derive the nominal position trajectory from which the artificial position and artificial reaction-force constraints are derived to serve as the reference inputs to the hybrid position/force controller. The synthesis of position/force trajectories for compliant motion was realized by two mappings, $T-C$ and $C-PF$, which are defined to derive the nominal position trajectory $P(t)$ and freedom decomposition function $S(t)$ from the attributes of a given task and then the time sequence of the artificial position constraints and artificial reaction-force constraints from the $P(t)$ and $S(t)$.

6. References

1. Lee, C. S. G., Lee, B. H., "Planning of Straight Line Manipulator Trajectory in Cartesian Space with Torque Constraint," *Proc. 23rd IEEE Conf. on Decision and Control*, Las Vegas, Nevada, Dec. 1984, pp. 1603-1609.
2. Lee, C. S. G., Smith, R., "Pattern Recognition Methods for Force Recognition in Insertion Processes," *Proc. 1984 American Control Conf.*, San Diego, June, 1984, pp 39-44.
3. Lee, C. S. G., "Robot Arm Kinematics, Dynamics, and Control," *IEEE Computer*, vol. 15, no. 12, December 1982, pp 62-80.
4. C. S. Lin, P. R. Chang, and J. Y. S. Luh, "Formulation and Optimization of Cubic Polynomial Joint Trajectories for Industrial Robots," *IEEE Trans. Auto. Contr.*, Vol. AC-28, No. 12, December 1983, pp. 1066-1073.
5. Lozano-Perez, T., Mason, M., and Taylor, R., "Automatic Synthesis of Fine-Motion Strategies for Robots," *The International J. of Robotics Res.*, vol. 3, no. 1, Feb. 1984.
6. Lozano-Perez, T., "Spatial Planning: A Configuration Space Approach," *IEEE Trans. Computers*, vol. 32, no.2, Feb. 1983, pp.108-120.
7. Mason, M. T., "Compliance and Force Control for Computer-Controlled Manipulators," *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-11, no. 6, 1981, pp. 418-432.

8. Paul, R., Shimano, B., "Compliance and Control," *Proc. Joint Auto. Contr. Conf.*, Purdue University, July 1976, pp 694-699.
9. Raibert, M. H., Craig, J. J., "Hybrid Position/Force Control of Manipulators," *Trans. ASME, J. Dynamic Syst., Meas., Contr.*, vol. 102, June 1981, pp 126-133.
10. Requicha, A., "Representations for Rigid Solids: Theory, Methods, and Systems," *Computer Surveys*, vol. 12, no. 4, Dec. 1980, pp 437-464.
11. Salisbury, J., "Active Stiffness Control of Manipulator in Cartesian Coordinates," *Proc. 19th IEEE Conf. on Decision and Control*, 1980, pp. 95-100.
12. Wesley, M. A., "Construction and Use of Geometric Models," in *Computer Aided Design, Modeling, System Engineering, CAD-Systems* Chapter 2, edited by J. Encarnacao, Springer Verlag, 1980, pp. 79-136.

Appendix A

The geometrical knowledge about an operand, as described in Section 3, is represented as follows:

O = any operand,

F or F_i = any feature,

A or A_i = any instance of a primitive solid,

$op_i \in \{ +', -' \}$,

where $+'$ and $-'$ are regularized set union and difference, respectively.

Then

$$\begin{aligned} O &= F \text{ or } O = F_1 op_1 F_2 op_2 \cdots op_{m-1} F_{m-1} \\ F &= A \text{ or } F = A_1 op_1 A_2 op_2 \cdots op_{k-1} A_{k-1} \end{aligned} \tag{A-1}$$

Since A is always convex, a hollow feature can be represented as

$$-'(A +' A +' \cdots +' A)$$

while a solid feature can be represented as

$$+'(A +' A +' \cdots +' A).$$

Noting the following identities,

$$A_1 +' (A_2 +' A_3) \equiv A_1 +' A_2 +' A_3$$

$$A_1 -' (A_2 +' A_3) \equiv A_1 -' A_2 -' A_3$$

$$+' A \equiv A$$

Eq. A-1 can be rewritten as

$$O = A \text{ or } O = A_1 op_1 A_2 op_2 \cdots op_{n-1} A_n \quad (\text{A-2})$$

Assertion 2

For any arbitrary 3-D operand O represented as in Eq. A-2, the 2-D cut-diagrams, G_{uv} , $uv = zz, yz$, and xy planes of H_{ow2} , can be decomposed as

$$G_{uv} = c_{uv} A_1 op_1 c_{uv} A_2 op_2 \cdots op_{n-1} c_{uv} A_n \quad (\text{A-3})$$

where $c_{uv} A_i$ is the cut-diagram of an instance of the primitive solid A_i at the $U-V$ plane.

proof:

Assertion 2 can be proved by induction.

1. basic step:

Assume $O = A_1 +^{\circ} A_2$

$O, A_1, A_2 \subseteq U$ where U is a 3-D Cartesian space.

$\forall \Phi \in U$:

$$\Phi \in O \iff \Phi \in A_1 +^{\circ} A_2$$

$$\iff \Phi \in C_i(A_1 \cup A_2) \text{ where } C_i(\cdot) \text{ represents closure(interior(\cdot))}$$

$$\iff (\Phi \in A_1 \cup A_2) \text{ and } (\Phi \notin ((A_1 \cup A_2) - C_i(A_1 \cup A_2)))$$

$$\iff (\Phi \in A_1 \text{ or } \Phi \in A_2) \text{ and } (\Phi \notin ((A_1 \cup A_2) - C_i(A_1 \cup A_2)))$$

(A-4)

Now, if we restrict $\Phi \in \mathbf{P}$ to a 2-D plane in \mathbf{U} , then,

$$\Phi \in \mathbf{P} \text{ and } \Phi \in O \implies \Phi \in cpO$$

$$\Phi \in \mathbf{P} \text{ and } \Phi \in A_1 \implies \Phi \in cpA_1$$

$$\Phi \in \mathbf{P} \text{ and } \Phi \in A_2 \implies \Phi \in cpA_2$$

where cpO , cpA_1 , and cpA_2 are the cut-diagrams of O , A_1 , and A_2 on the plane \mathbf{P} , respectively.

Then A-4 becomes

$$\forall \Phi \in \mathbf{P}: \Phi \in cpO \iff (\Phi \in cpA_1 \text{ or } \Phi \in cpA_2) \text{ and} \\ (x \notin ((cpA_1 \cup cpA_2) - C_i(cpA_1 \cup cpA_2)))$$

$$\iff \Phi \in cpA_1 +^* cpA_2$$

$$\implies cpO = cpA_1 +^* cpA_2$$

Similarly, we can prove

$$O = A_1 -^* A_2 \implies cpO = cpA_1 -^* cpA_2$$

Therefore

$$O = A_1 op_1 A_2 \implies cpO = cpA_1 op_1 cpA_2$$

where $op \in \{+^*, -^*\}$

2). Assumption step:

Assume

$$\begin{aligned}
O &= A_1 \text{ op}_1 A_2 \text{ op}_2 \cdots \text{ op}_{n-2} A_{n-1} \\
\implies cpO &= cpA_1 \text{ op}_1 cpA_2 \text{ op}_2 \cdots \text{ op}_{n-2} A_{n-1}
\end{aligned}$$

holds true.

3). Induction step:

$$\begin{aligned}
O &= A_1 \text{ op}_1 A_2 \text{ op}_2 \cdots \text{ op}_{n-2} A_{n-1} \text{ op}_{n-1} A_n \\
&= (A_1 \text{ op}_1 A_2 \text{ op}_2 \cdots \text{ op}_{n-2} A_{n-1}) \text{ op}_{n-1} A_n \\
&= B \text{ op}_{n-1} A_n \\
\implies cpO &= cpB \text{ op}_{n-1} cpA_n \\
\implies cpO &= cp(A_1 \text{ op}_1 A_2 \text{ op}_2 \cdots \text{ op}_{n-2} A_{n-1}) \text{ op}_{n-1} cpA_n \\
\implies cpO &= cpA_1 \text{ op}_1 cpA_2 \text{ op}_2 \cdots \text{ op}_{n-2} cpA_{n-1} \text{ op}_{n-1} cpA_n
\end{aligned}$$

(A-5)

Since \mathbf{P} is any arbitrary 2-D plane in the 3-D Cartesian space, the induction A-5 holds true on the $X-Z$, $Y-Z$, and $X-Y$ planes of the body attached frame H_{ow2} .

Q.E.D.

Consequently, the correctness of the divide-and-conquer procedure CUT_DIA described in Section 4 is the direct result of the Assertion 2.

Appendix B

Assertion 1 in Section 4 is restated here for convenience.

Assertion 1

$$\text{Assume } \textit{operand1} = A_1 \textit{op}_1 A_2 \textit{op}_2 \cdots \textit{op}_{m-1} A_m \quad (\text{B-1})$$

$$\text{Then } GE_{uv} = \bigcup_{i=1}^m GE_{iuv} ; uv = xy, yz, zx$$

where GE_{iuv} is the expanded G_{uv} corresponding to A_i , $uv = zx, yz$ and xy planes of H_{ov2} , and $\textit{op}_i = +^\dagger$.

Proof:

1. First we prove that $\Phi \in GE_{uv} \implies \Phi \in \bigcup_{i=1}^m GE_{iuv}$, where $\Phi = (u, v)$ is a point in the $U-V$ plane.

$\Phi \in GE_{uv}$ means that if *operand1* is put at a position such that L_{uv} passes through Φ , then *operand1* will collide with *operand2* on the $U-V$ plane. The collision point at least belongs to one of the A_i 's. Therefore $\Phi \in GE_{juv}$, $j \in \{1, 2, \dots, m\}$ so that

$$\Phi \in GE_{uv} \implies \Phi \in \bigcup_{i=1}^m GE_{iuv}. \quad (\text{B-2})$$

2. Next we need to prove that $\Phi \in \bigcup_{i=1}^m GE_{iuv} \implies \Phi \in GE_{uv}$

[†] The object which *operand1* belongs to might have some hollow features. However, in an insertion task, the feature-operand of *operand1* and its direct neighbors are solid ones. Therefore

$\Phi \in \bigcup_{i=1}^m GE_{iuv}$ means that Φ belongs to at least one GE_{iuv} . We denote it as $GE_{j_{uv}}$, corresponding to A_j . Therefore, if L_{uv} passes through $\Phi = (u, v)$, then A_j will collide with operand2 on the $U-V$ plane and so will operand1, since A_j is a solid part of operand1. Therefore

$$\Phi \in \bigcup_{i=1}^m GE_{iuv} \implies \Phi \in GE_{uv} \quad (\text{B-3})$$

From B-2 and B-3, we conclude that

$$GE_{uv} = \bigcup_{i=1}^m GE_{iuv}, \quad uv = xz, yz, xy.$$

Q.E.D.

the combinational operators in Eq. B-1 are all + 's.

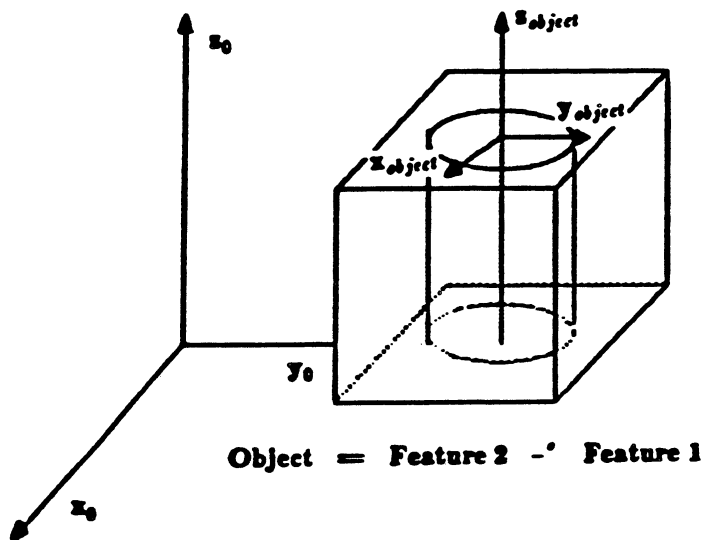
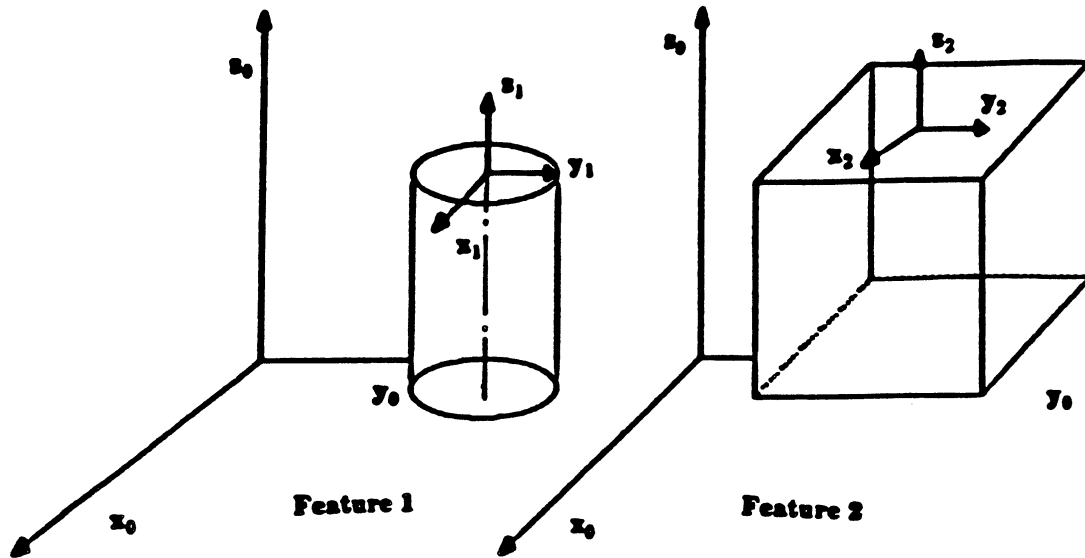


Figure 1. An object which is the Combination of Solid and Hollow Features

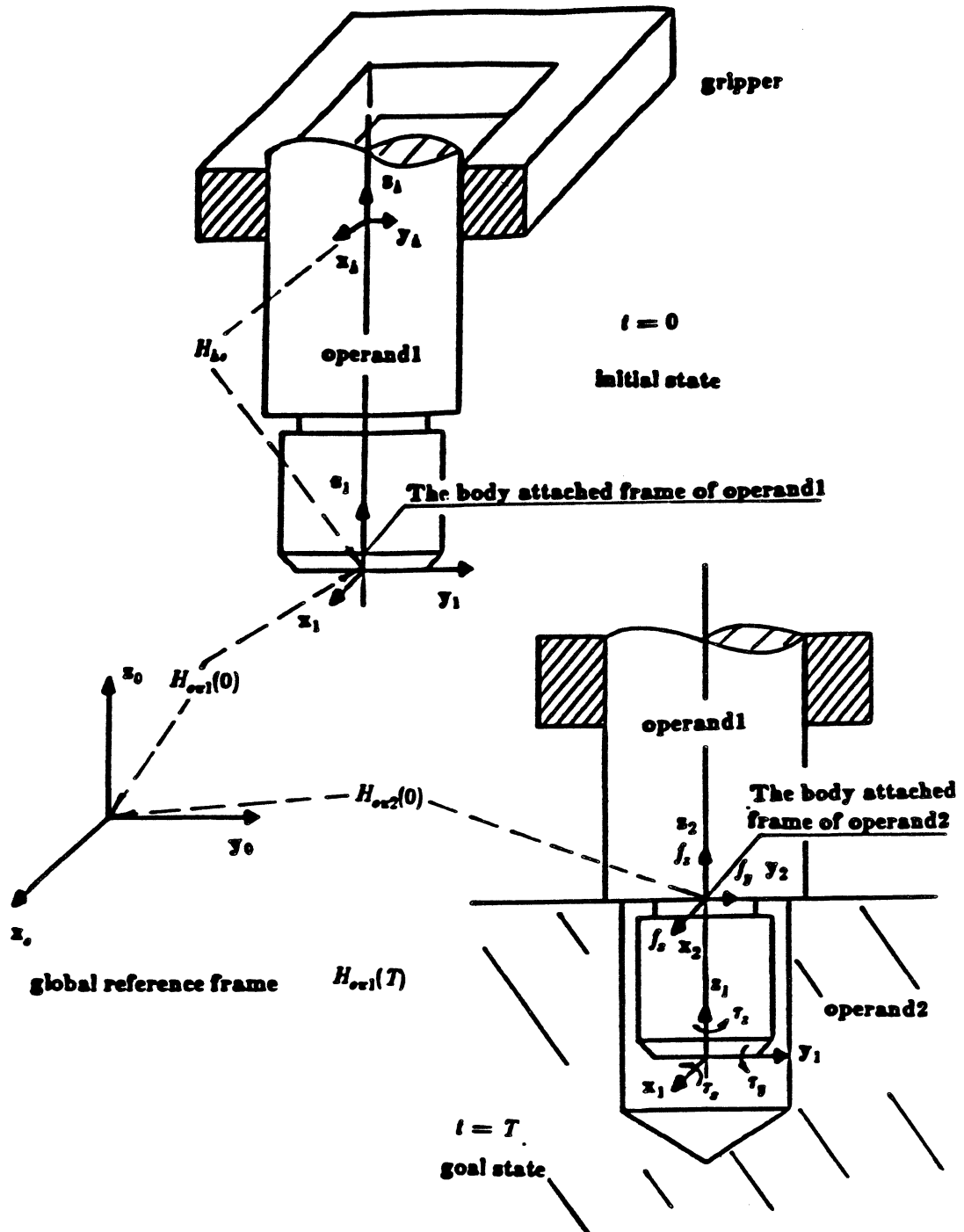


Figure 2. An Example of Peg-hole Insertion

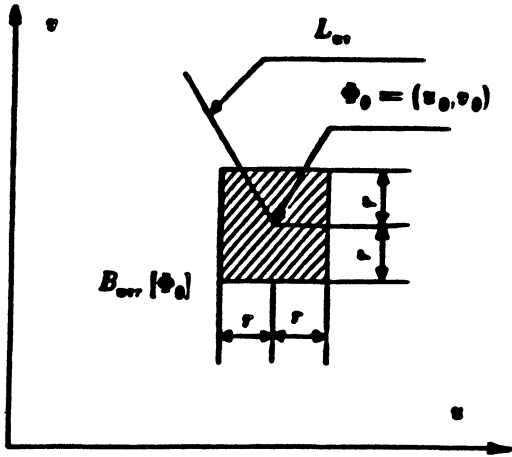


Figure 3(a). The ball $B_{uv}[\Phi_0]$

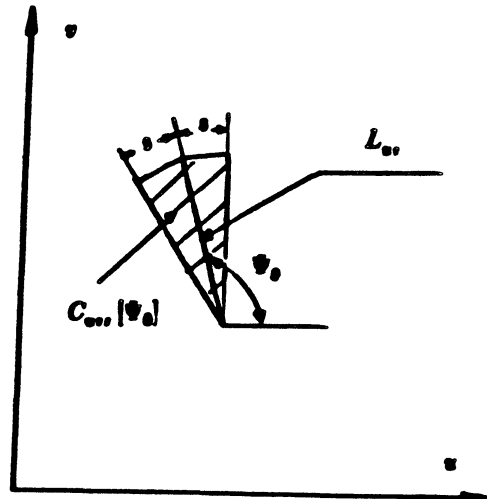


Figure 3(b). The cone $C_{uv}[\Psi_0]$

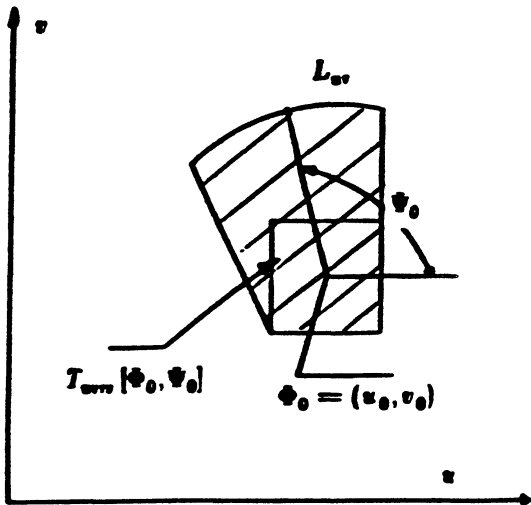


Figure 3(c). The trapezoid $T_{uv}[\Phi_0, \Psi_0]$

Figure 3. (a) The ball $B_{uv}[\Phi_0]$.
 (b) The cone $C_{uv}[\Psi_0]$.
 (c) The uncertainty trapezoid $T_{uv}[\Phi_0, \Psi_0]$.

UNIVERSITY OF MICHIGAN



3 9015 03023 0315