22nd International Symposium on Industrial Robots

♦ ♦ ♦ ♦ ♦

International Robots & Vision Automation Conference

October 21–24, 1991

Detroit, Michigan USA

Proceedings

Session Seventeen

ROBOT CONTROL TECHNIQUES

International Robots & Vision Automation Conference
and
22nd International Symposium on Industrial Robots

October 21–24, 1991

Assembly with Robots In Presence Of Uncertainty

RAÚL SUÁREZ
and
LUIS BASAÑEZ
Institut de Cibernètica (UPC-CSIC)
Barcelona, SPAIN

ABSTRACT—Automatic assembly of two rigid objects with small clearance and position uncertainty has not been fully solved in robotics. Present solutions (increasing robot precision, improving object design, using passive compliance, using sensors with preprogrammed strategies . . .) do not lead to completely automatic assembly systems. We propose a new methodology to automatically generate an assembly plan based on position and force information, considering several kinds of uncertainty, friction forces and rotational degrees of freedom. Tasks states are defined according to the occurrence of different sets of basic contacts, and state transition operators are established to move through them. Task execution consists in indentifying present state and applying a selected associated operator, until the goal state is reached.

1 Introduction

Nowadays, robots are used in a broad field of applications and robot programming is intended to be done automatically from task level specifications. Despite this, assembly with robots still exhibits some basic problems when uncertainty in robot positioning is comparable to parts matting clearance. This has restrained robot applications in assembly.

The significance of the subject has lead to a great deal of research in the area and several solutions have been proposed. The obvious one consists in improving the precision of the robot and of the objects positioning, but it is expensive and there will be always a limit. Another approach includes some deformable element in the robot end effector to permit *passive compliance*; the typical device is the Remote Center Compliance (RCC). The same effect is tried to be done by *active compliance*, this is, by sensing reaction forces and using an active device (usually the robot itself) to correct the position; the advantage is the possibility of programmable compliance, and the drawback is the need of a force control besides position control. Simultaneously, parts and objects involved in the task have been better designed to simplify the assembly. Finally, assembly strategies are developed by a human operator, but they are task-dependent and frequently difficult to be found. Then, the need of an automatic fine motion planning system comes up. Such a system must be capable of establish a sequence of movements that assure success in assembly tasks despite uncertainty.

There are several works dealing with fine motion planning, but they often consider a particular or excessive simplified problem, or, on the contrary, the proposed solutions are too complicated to be readily applied. There are also a number of interesting works in closed related areas to fine motion planning (e.g. position/force control; movement of a rigid body with several frictional contacts; model, manipulation and propagation of uncertainty) that have contributed with key ideas.

The use of position/force control to compensate uncertainty in assembly tasks was introduced by Inoue (1974). Several approaches to automatic determination of strategies to perform assemblies using position/force control were then proposed. Dufay and Latombe (1984) described a method to generate an insertion strategy from several training task executions. Turk (1985) described a fine motion planning algorithm based on geometric regions and the movements directions to go through them, rotations and geometric uncertainty was not included. Lozano Perez, Mason and Taylor (1984) presented other approach to the synthesis of compliant motion strategies introducing the concept of *pre-image* obtained from the task geometric information, goal position and commanded velocity. In the same line, Erdmann (1984) suggested a method for planning motions in presence of uncertainty, based on the concepts of *pre-image* and *back-projection*, and Buckley (1987) presented an interactive system to build a compliant motion strategy and a planner capable of dealing with simple problems. Other approaches divide the problem in two phases: in the first, a plan is built without considering uncertainty, in the second, uncertainty is added and the plan is properly modified or adapted; such are the approaches of Xiao and Volz (1989) and Gottschlich and Kak (1991).

In this paper we describe a fine motion planner for assembly tasks. It tries to be simple enough, although all kind of uncertainties and friction forces are considered. The planner was developed considering rigid polyhedral objects and planar movements (i.e. two translational and one rotational degrees of freedom). Problems with more degrees of freedom can be theoretically solved by this way, but they become too complicated, as happens with most of the previous approaches.

In section 2 the position and force representation are described, including a description of friction forces for planar movements. Section 3 deals with uncertainty. Uncertainty sources in real world are enumerated and a model is established for each one. Then, these models are used to model uncertainty in Configuration Space and Generalized Force Space. In section 4 the task is modeled as a set of states, defined according to basic contacts between the mobile object and the environment. Uncertainty is considered to determine the information that sensors could give in each state. Section 5 introduce State Transition Operators to move between states, then the planning procedure is described. Section 6 simply describes how the plan is executed. Finally, some conclusions are outlined.

2 Position and Force Representation

2.1 Object Position and Orientation

In order to represent the mobile object position and orientation we make use of the well-known concept of *Configuration Space* (Lozano Perez, 1983). Then, the object degrees of freedom are described by a set of parameters which can be considered as components of a vector $\vec{c}$, called the object *configuration*. The space that vectors $\vec{c}$ define is the Configuration Space, C.

Real obstacles in work space give rise in C to hyper-surfaces called *C-surfaces*, which contains all contact configurations and divide C into two sets of configurations: Ci, the set of impossible configurations due to the physical intersection of the mobile object with an obstacle in the environment, and Cp, the set of possible configurations which is divided again into Cc, the set of contact configurations, and Cf, the set of free configurations in which there is no contact and the object is free to make any movement.

When rotational degrees of freedom are considered, before using the concepts of distance and orthogonality a metric adjust of C becomes necessary. This can be done by multiplying parameters describing orientation angles by a constant, ρ, with units of *length*. For a dynamic analysis, the numerical value of ρ must be the radio of gyration, but it is not relevant for a static analysis.

In order to describe the configuration of an object being moved on a plane by a robot, we choose a reference point fixed to the gripper (e.g. the TCP point) represented by $\vec{p_r}$ in the absolute reference system. The three degrees of freedom (two of translation and one of rotation) will be indicated by p_{rx}, p_{ry} and $\rho\phi_r$. Since the numerical value of ρ is not critical for this analysis, it can be taken as unitary and thus a point of Configuration Space will be directly represented by $[p_{rx} \;\; p_{ry} \;\; \phi_r]^T$.

Polyhedral objects moved on a plane can be represented as polygons, and two kinds of basic contacts are possible between two of them: a vertex of the moving object meets an edge of a static object, called *type-1*, and an edge of the moving object meets a vertex of a static object, called *type-2*. All other contacts can be represented as a combination of these basic contacts. Each type of basic contact has associated a C-surface in the 3-dimensional Configuration Space, also called *type-1* or *type-2*. C-surfaces intersect each other defining portions of them that represent real contact configurations, this is, belong to *Cc*. These portions are called *C-faces* of *Cc*, and their boundaries consist of *C-edges* and *C-vertices* of *Cc*. C-surfaces have been broadly studied in absence of uncertainty (Canny, 1988).

2.2 Reaction Forces and Torques

In a similar way, force and torque components can be represented using a unique vector $\vec{g}$ called *generalized force*. The space that these vectors define is called *Generalized Force Space*, $\mathcal{F}$.

Again, when torques are considered, a metric adjust is necessary to confer to the generalized forces an equivalent behavior to that of real forces in real space. This is done by dividing components representing torques by the constant ρ introduced above. In this way, when a generalized force is applied onto a C-surface of C, the generalized reaction force lies on the C-surface external normal for the unfrictional case, and inside a generalized friction cone for the general frictional case (Erdmann, 1984). For the planar case, real forces are described by $[f_x \;\; f_y]^T$ and torques by τ, then generalized forces are of the form $\vec{g} = [f_x \;\; f_y \;\; f_\tau]^T$ being $f_\tau = \tau/\rho$.

Friction

Real friction forces, $\vec{f_t}$, appear when a contact point moves (or tend to move) while keeping contact. Friction force direction is tangent to the contact surface and the sense is opposite to the movement, and, according to Coulomb's friction law, its module is given by $f_t = \mu f_n$, f_n being the module of the force applied on the contact surface in the normal direction. μ varies inversely to the velocity of the contact point, but it can be approximated by its static value considered as a constant. Thus, the reaction force is always within a friction cone determined by an angle $\theta_c = \arctan\mu$. Object movements that produce a rotation around the contact point does not generated any friction force.

Therefore, the friction cones for generalized and real forces are not equal (Erdmann, 1984). For the case of planar movements, $\mathcal{F}$ is 3-dimensional and the friction cone is given by

$$\vec{n} \pm k\mu\vec{m} \tag{1}$$

where $\vec{n} = [n_x \;\; n_y \;\; n_q]^T$ is the C-surface external normal, $\vec{m} = [-n_y \;\; n_x \;\; (n_x r_x + n_y r_y)/\rho]^T$ with $\vec{r} = [r_x \;\; r_y]^T$ the vector from the contact point to the reference point (for both type-1 and type-2 contacts), and $k \in [0,1]$. The projection of this generalized friction cone over the plane of null torque gives the real forces friction cone. Figure 1 illustrate a generic friction cone for planar movements.

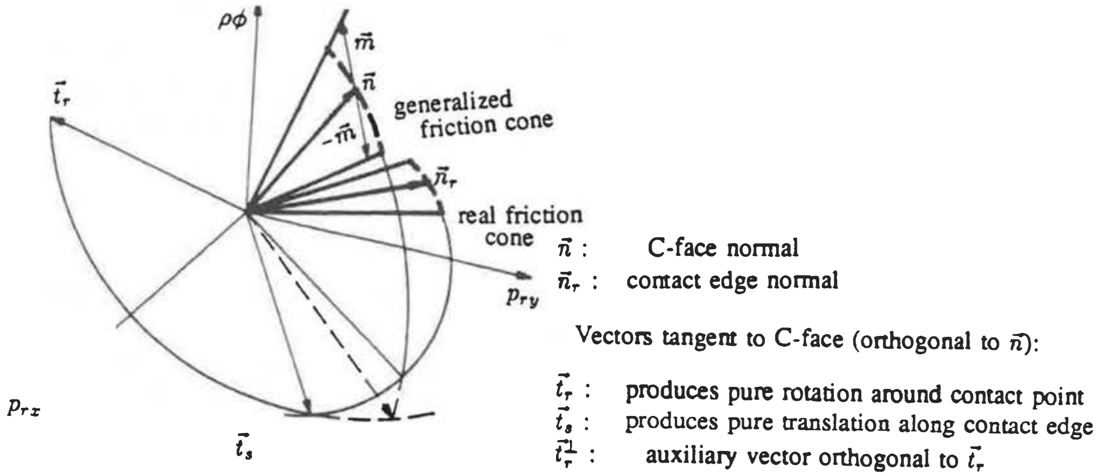

Figure 1: Generalized friction cone for planar movements

3 Uncertainty

Parameters and variables describing objects or objects' behavior in real world are not exactly known. In some robot tasks, it is not enough to deal with *nominal* or *predicted* values, but some specification about their imprecision is necessary. This is the case of robot fine-motion planning, specially when the object in the robot gripper (or the robot itself) may become in contact with other objects in the environment.

We will call *deviation* of a parameter or variable to the difference between the actual and the measured or calculated value, and *uncertainty* of such parameter or variable to the domain containing all possible actual values for an observed one subject to deviations. Deviations are represented by δ and uncertainties by U, both with a subscript representative of the related variable or parameter. Parameters used to describe maximum deviations are represented by ϵ with the same subscript. The subscript o indicates an *observed* value, sensed or calculated. This nomenclature is valid for both scalar and vectorial quantities.

Strictly speaking, uncertainty always arises from a measurement process. So, uncertainty has always sensorial origin, and it is propagated as the measured values are used to calculate another ones (theory of errors). Nevertheless, it is often possible to consider and to model higher level uncertainty sources for some specific purposes. In the following subsection we will enumerate and model the uncertainty sources for the motion planning problem. This includes uncertainty in configuration space, in generalized forces space and in generalized velocity of the mobile object. Uncertainty in configuration space, which is due to several sources, has been described with more details in Basañez and Suárez (1991).

3.1 Uncertainty in Real World

3.1.1 Uncertainty Sources and Associated Models

Uncertainty of the shape and size of objects. Uncertainty of the objects shape and size is due to their manufacturing tolerances. Describing the object boundary by using vectors $\vec{v}$ referred to an object reference point P_g and an axis j that has P_g as origin (Figure 2), objects shape and size uncertainty is expressed through

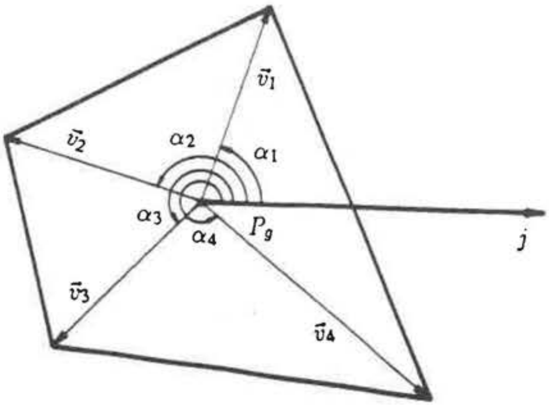

Figure 2: Description of an object using a point P_g and an axis j as reference

the vertex position uncertainty:

$$U_v = \{\vec{v} \mid \|\vec{v} - \vec{v}_o\| \leq \epsilon_v\} \tag{2}$$

Uncertainty of the position of any point of the object boundary can be modeled by a similar expression. It is also assumed that the objects boundary satisfies an additional slow-variation condition.

Uncertainty of the position measurement of a point of a static object. This uncertainty entirely depends on the sensors or measurement system used to locate the point $\vec{a}$, but a generic model can be established as

$$U_m = \{\vec{a} \mid \|\vec{a} - \vec{a}_o\| \leq \epsilon_m\} \tag{3}$$

Uncertainty of the configuration of a static object. Three different situations are possible depending on: 1) the object configuration is known off-line, 2) the object configuration is observed on-line, 3) the object configuration is known because the object has been placed there by the robot. Since the plan is going to be generated off-line and assuming that the object will not be previously manipulated by the robot, we are in the first situation. A typical case is the object positioning by a feeder. Being desirable an uncertainty model independent of any specific feeder, the simplest way of modeling uncertainty is to assume that each object vertex is located inside a circle of radius ϵ_{p_p} centered at the vertex predicted position. From this assumption, it follows that every object point, including the object reference point P_g, is constrained to be inside a circle of the same radius. The object orientation uncertainty is limited to $\epsilon_{\phi_p} = \arcsin(2\epsilon_{p_p}/l_m)$ where l_m is the maximum distance between two vertices; object orientation uncertainty can be ignored to determine position uncertainty of any object point.

Uncertainty of the robot positioning. Robot precision is usually specified by two parameters, representing the maximum distance and angle that position and orientation of the robot end effector can defer from the predicted values. The same parameters can also be chosen to specify position and orientation uncertainty independently of robot configuration. Using $\vec{p}_r$ and ϕ_r to represent the real position and orientation of the end effector, the corresponding uncertainty is

$$U_{p_r} = \{\vec{p}_r \mid \|\vec{p}_r - \vec{p}_{ro}\| \leq \epsilon_{p_r}\} \tag{4}$$

$$U_{\phi_r} = \{\phi_r \mid |\phi_r - \phi_{ro}| \leq \epsilon_{\phi_r}\} \tag{5}$$

Slide of the object in the gripper. Undesired slides can be strongly reduced by using both a proper gripper and a proper grasping strategy. Uncertainties due to other sources can also be reduced in this way. We model the slide of the object in the gripper assuming upper limits for translation, ϵ_{p_d}, and for rotation around the reference point, ϵ_{ϕ_d}.

Uncertainty of the reaction forces and torque measurements. The uncertainty model of reaction forces and torque measurements depends on the type of sensors used and where they are placed. In this work, we will consider a wrist force/torque sensor that gives reaction forces as two orthogonal components and the torque about a predefined point; for simplicity this point will be considered as coincident with the reference point ($\vec{p}_r$) used to build C-surfaces in $\mathcal{C}$. Uncertainty is a function of the sensor precision and can be modeled as,

$$U_{f_x} = \{f_x \mid \|f_x - f_{xo}\| \leq \epsilon_{f_x}\} \quad (6)$$

$$U_{f_y} = \left\{f_y \mid \|f_y - f_{yo}\| \leq \epsilon_{f_y}\right\} \quad (7)$$

$$U_\tau = \{\tau \mid \|\tau - \tau_o\| \leq \epsilon_\tau\} \quad (8)$$

Uncertainty of the mobile object velocity. This uncertainty is due to the robot control system errors. Deviations of the velocity depend on the configuration of the manipulator and also on the magnitude of the commanded velocity. Moreover, linear and angular velocities are, in general, not independent.

We choose an uncertainty model that relates linear and angular velocities, so it is described directly for generalized velocities of the form $\vec{V} = [V_x \;\; V_y \;\; V_\phi]^T = [\dot{p}_{rx} \;\; \dot{p}_{ry} \;\; \rho\dot{\phi}_r]^T$.

Real velocity is bounded to be inside a cone whose axis direction is determined by the nominal velocity direction. Modules of real and nominal velocities satisfy $\|\vec{V}\| - \|\vec{V}_o\| \leq \epsilon_{V_M}$.

3.1.2 Uncertainty of the Absolute Position of any Object Point

In order to predict a possible contact configuration it is necessary to consider the uncertainty of the absolute location of the objects boundary. This can be done by determining the uncertainty of the absolute position of each boundary point. Uncertainty sources are different for a static object lying in the work environment and for a grasped object fixed in the robot gripper.

Static object. Depending on how the position of a point is estimated, the following two cases are possible:
1) Point position computed from a nominal object model and the object configuration. Uncertainty comes from:

- deviations of the object shape and size (δ_v)
- deviation of the object configuration (δ_{p_p}).

With the proposed models both deviations can be added as $\delta_a = \delta_v + \delta_{p_p}$. Then, the maximum deviation is $\epsilon_a = \epsilon_v + \epsilon_{p_p}$.

2) Direct observation of the point position by a sensor external to the robot. Uncertainty comes from the measurement deviation of the observation system (δ_m); then $\delta_a = \delta_m$.

The *absolute position of a point of a static object* that belongs to the edge limited by vertices $\vec{a}_1$ and $\vec{a}_2$, will be given by

$$\vec{a} = \vec{a}_1 + k_a(\vec{a}_2 - \vec{a}_1) = \vec{a}_{1o} + k_a(\vec{a}_{2o} - \vec{a}_{1o}) + \vec{\delta}_a = \begin{bmatrix} a_{1xo} + k_a(a_{2xo} - a_{1xo}) + \delta_a \cos\theta_a \\ a_{1yo} + k_a(a_{2yo} - a_{1yo}) + \delta_a \sin\theta_a \end{bmatrix} \quad (9)$$

where $k_a \in [0,1]$.

Grasped object. The deviations bringing about from the following three sources are to be considered:

- deviations of the object shape and size (δ_v)
- deviations of the position and orientation of the robot end effector (δ_{p_r} and δ_{ϕ_r})
- deviations of the position and orientation of the object in the gripper, which are due to:
  - deviations of the object configuration before it was grasped (δ_{p_p})
  - deviations of the position and orientation of the robot end effector during the grasping (δ_{p_r} and δ_{ϕ_r})
  - slides of the object in the gripper (δ_{p_d} and δ_{ϕ_d})

  The deviation of the position of the object in the gripper can be summarize as $\delta_{p_g} = \delta_{p_r} + \delta_{p_p} + \delta_{p_d}$. Then the maximum deviation is $\epsilon_{p_g} = \epsilon_{p_r} + \epsilon_{p_p} + \epsilon_{p_d}$. Since δ_{ϕ_r} and δ_{ϕ_d} are referred to the same rotation point, the deviation of the orientation of the object in the gripper can be summarized as $\delta_{\phi_g} = \delta_{\phi_r} + \delta_{\phi_d}$, with the maximum value $\epsilon_{\phi_g} = \epsilon_{\phi_r} + \epsilon_{\phi_d}$.

Uncertainties affecting the absolute position of a vertex of a grasped object are shown in Figure 3.

The *absolute position of a point of a grasped object* that belongs to the edge limited by vertices $\vec{b}_1$ and $\vec{b}_2$ will be given by

$$\vec{b} = \vec{b}_1 + k_b(\vec{b}_2 - \vec{b}_1) = \left[\begin{array}{l} p_{r_xo} + h_1\cos(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\cos\theta_{p_rp_gv} + \\ k_b(h_2\cos(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_2) - h_1\cos(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_1)) \\ \\ p_{r_yo} + h_1\sin(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\sin\theta_{p_rp_gv} + \\ k_b(h_2\sin(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_2) - h_1\sin(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_1)) \end{array} \right] \quad (10)$$

where $k_b \in [0,1]$, $\epsilon_{p_rp_gv} = \epsilon_{p_r} + \epsilon_{p_g} + \epsilon_v$, and $\epsilon_{\phi_r\phi_g} = \epsilon_{\phi_r} + \epsilon_{\phi_g}$.

3.2 Uncertainty in the Configuration Space

Type-1 C-surfaces. A grasped object vertex $\vec{b}$ is in contact with a point $\vec{a}$ of a static object edge when the condition $\vec{a} = \vec{b}$ is satisfied. From expression (9) and expression (10) computed for $k_b = 0$, it results

$$\left\{ \begin{array}{l} p_{r_xo} + h_1\cos(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\cos\theta_{p_rp_gv} = a_{1xo} + k_a(a_{2xo} - a_{1xo}) + \delta_a\cos\theta_a \\ p_{r_yo} + h_1\sin(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\sin\theta_{p_rp_gv} = a_{1yo} + k_a(a_{2yo} - a_{1yo}) + \delta_a\sin\theta_a \end{array} \right. \quad (11)$$

By eliminating k_a, a family of type-1 C-surfaces parametrized in the deviations could be obtained.

Type-2 C-surfaces. A point $\vec{b}$ of a grasped object edge is in contact with a vertex $\vec{a}$ of a static object when the condition $\vec{a} = \vec{b}$ is satisfied. From expression (9) computed for $k_a = 0$ and expression (10), it results

$$\left\{ \begin{array}{lcl} a_{1xo} + \delta_a\cos\theta_a & = & p_{r_xo} + h_1\cos(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\cos\theta_{p_rp_gv} + \\ & & k_b(h_2\cos(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_2) - h_1\cos(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_1)) \\ \\ a_{1yo} + \delta_a\sin\theta_a & = & p_{r_yo} + h_1\sin(\phi_{ro} + \phi_{go} + \delta_{\phi_r\phi_g} + \gamma_1) + \delta_{p_rp_gv}\sin\theta_{p_rp_gv} + \\ & & k_b(h_2\sin(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_2) - h_1\sin(\phi_{ro}+\phi_{go}+\delta_{\phi_r\phi_g}+\gamma_1)) \end{array} \right. \quad (12)$$

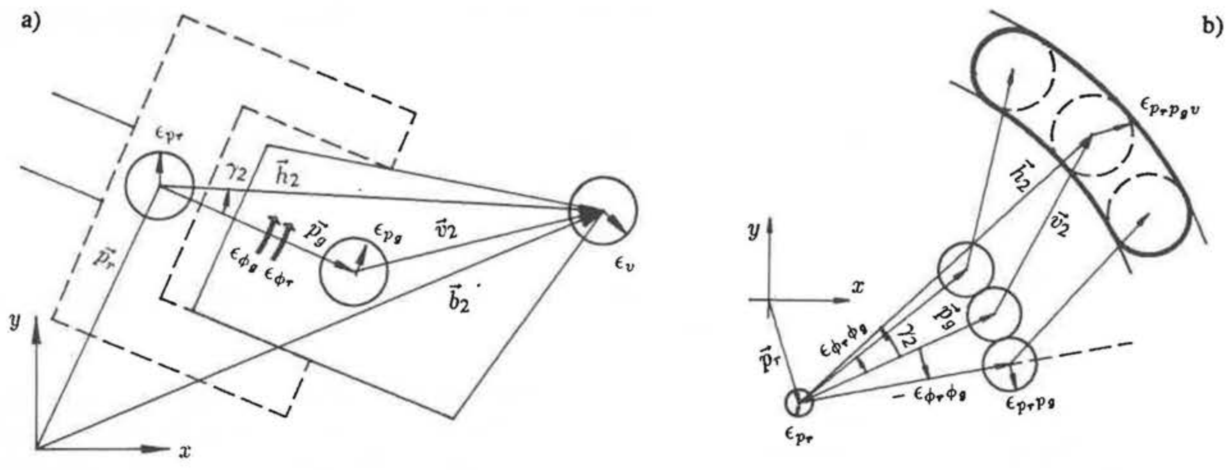

Figure 3: a) Uncertainties affecting the absolute position of a vertex of a grasped object. b) Resulting possible absolute position of the vertex.

By eliminating k_b, a family of type-2 C-surfaces parametrized in the deviations could be obtained.

The families of type-1 and type-2 C-surfaces have an associated family of C-faces, which contain *all possible contact configurations in presence of uncertainty (UCc)*. When planning robot movements using the Configuration Space approach, the set of configurations *UCc* must be considered instead of the nominal set *Cc*.

UCc is made up of the *uncertainty regions CU* defined by each C-face family. Then, *UCc* boundary can be determined from the envelopes of *CU* regions. These envelopes are obtained in Basañez and Suárez (1991). They are made up of patches of surfaces of $\mathcal{C}$ generated by sweeping, along axis ϕ, circumferences and straight lines on planes parallel to the reference plane xy. Figure 4 shows xy cuts of *CU* for a type-1 C-face and the two possible cases in type-2 C-face.

3.3 Uncertainty in Generalized Force Space

Uncertainty of generalized forces is due to deviations in the forces and torques measurements (δ_{f_x}, δ_{f_y} and δ_τ). It must be distinguished between these uncertainties and the possibility that a C-surface normal varies due to uncertainty in $\mathcal{C}$, also changing the possible generalized reaction forces for the corresponding contact.

The model of uncertainty of a generalized force is quite simple and can be directly established from real force/torque uncertainty model. Equations (6) and (7) also hold for the two first components of a generalized force. For the third component, uncertainty specification is obtained dividing by ρ the condition of (8), resulting $|f_\tau - f_{\tau o}| \leq \epsilon_{f_\tau}$ where $\epsilon_{f_\tau} = \epsilon_\tau/\rho$.

The geometric interpretation of generalize force uncertainty is a prism whose size is given by ϵ_{f_x}, ϵ_{f_y} and ϵ_{f_τ}.

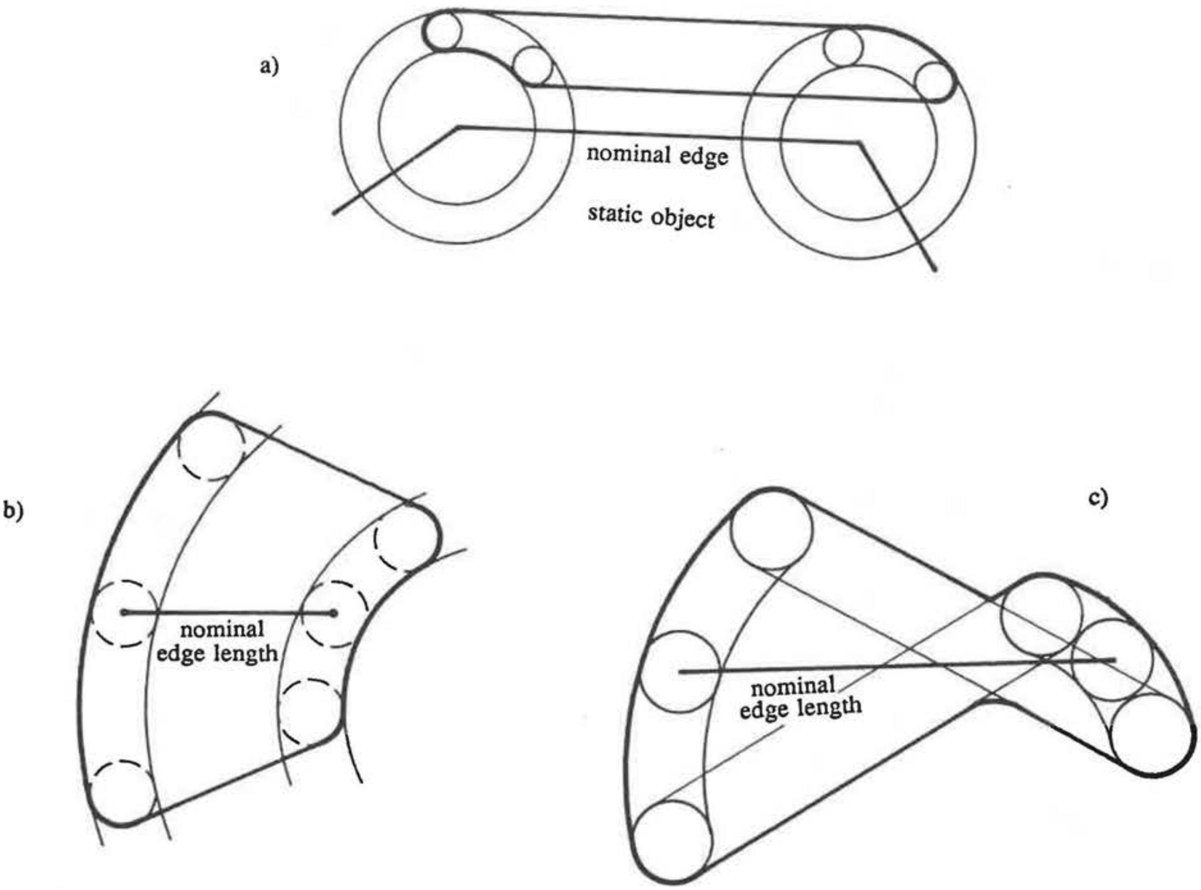

Figure 4: Cut of *CU* for C-faces: a) Type-1. b) Type-2 (case A). c) Type-2 (case B).

4 Task Modeling

4.1 Basic Contacts and Task States

When objects become in contact during an assembly task, it is important to know which basic contacts take place to decide next movement. We will define different task states according to basic contacts occurrence.

A set of edge-vertex basic contacts (each of them type-1 or type-2) is said to be *compatible* if all the contacts can occur simultaneously. Then, for each compatible set $\mathcal{E}$ of basic contacts between the mobile object and the environment, a *task state* E is defined as the set of connected configurations in which all basic contacts of $\mathcal{E}$, and only those, occur.

It must be noted that some particular compatible sets $\mathcal{E}$ can generate more than one state due to the condition of "connected" configurations.

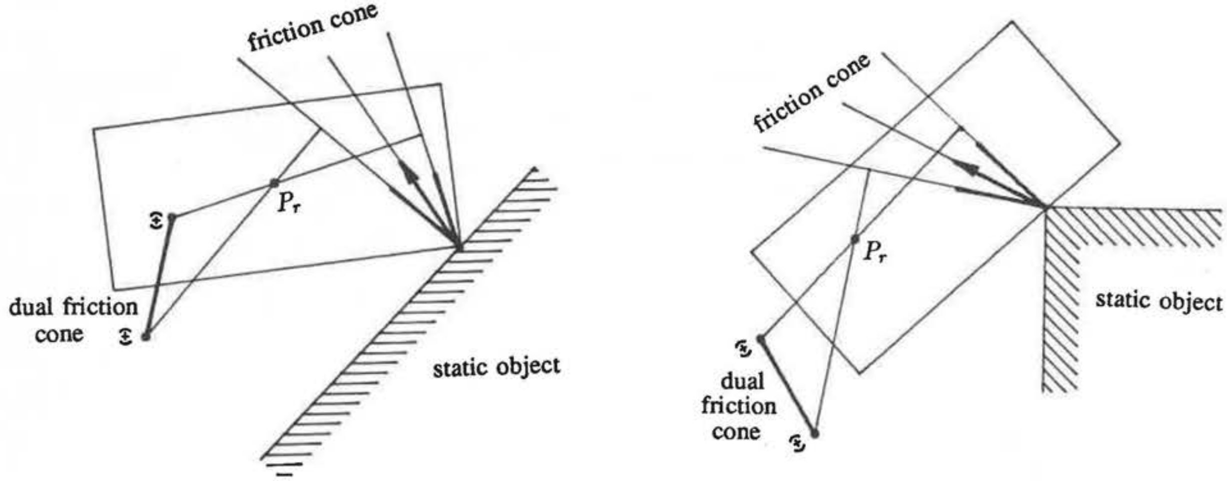

Figure 5: Possible generalized reaction forces for type-1 and type-2 contacts without uncertainty.

States determined from a nominal task model, i.e. without uncertainty, are called *nominal task states*. They have a clear geometric interpretation: each C-face (without the C-edges), each C-edge (without the C-vertices) and each C-vertex of *Cc* conforms a different nominal state.

Nominal states can be represented as nodes of a graph, *N-Graph*, where the arcs link those states that could be consecutive during a nominal task execution.

Generalized Reaction Forces in Nominal States

Generalized reaction forces expected in a nominal state can be obtained as follows. In the case of a nominal state with only one basic contact, possible directions [1] of reaction forces are those in the generalized friction cones computed for the corresponding C-face. For nominal states with more than one basic contact (i.e. vertices and edges configurations) the possible directions of reaction forces are those obtained by a lineal combination, with positive coefficients, of vectors from the generalized friction cone of each contact, computed over the corresponding vertex or edge configurations.

For the planar problem, generalized forces directions can be graphically represented by the method described by Brost and Mason (1989). The line of action of a real force *(force line)*, $ax + by + c = 0$, is represented by the dual point $(a/c, b/c)$, and a sign indicates the sign of the moment produced by the force with respect to the origin of reference. Considering that the origin of reference is coincident with the reference point (P_r) for torque measurements, this representation is equivalent (rotated $\pi/2$ radians) to consider the intersection point of the corresponding generalized force with the plane $f_\tau = 1$ preserving the sign of the component f_τ. Graphically, the dual point of a force line f is placed on a line orthogonal to f and $1/d$ far from the origin, d being the distance between f and the origin.

Figure 5 shows the representation of possible generalized reaction forces for type-1 and type-2 contacts without uncertainty for a given ϕ. If more than one contact take place, possible generalized reaction forces directions are represented by the convex hull of points representing possible forces from each contact.

[1] If no explanation is given, from now on the word *direction* will be used referring to both direction and sense.

4.2 States Realization

In a nominal model -without uncertainty- nominal states are easily recognized or predicted, but this is not always the case in real task executions. Thus, it is necessary to analyze possible states realizations in presence of uncertainty.

We will define the *region of possible realization of a state, R*, as the set of configurations in which that state can occur due to uncertainty.

By using results of section 3, R regions of states with only one basic contact can be obtained from the corresponding CU [2] uncertainty regions built for $\epsilon_{p_r} = 0$ and $\epsilon_{\phi_r} = 0$. R regions of states with more than one basic contact are obtained by intersecting CU regions of C-faces of each basic contact.

<u>Generalized Reaction Forces in States Realization</u>

Since the directions of the external normals to C-surfaces may vary due to uncertainty, the sets of possible reaction forces directions grow. Using the graphic representation, these sets are now limited not only by straight segments but also by some conic curves (Suárez, 1991). They are obtained by considering force directions within the corresponding friction cones and the force lines crossing the uncertainty region of the corresponding points of the object boundary. When more than one contact take place, forces directions are represented by linking with straight segments forces directions of each contact involved.

4.3 States Estimation

During task execution the actual state must be observed and recognized in order to apply a proper robot command. A state may be identified by using configuration and/or force measurements. In what follows we describe the set of configurations and generalized reaction forces that could be measured when a given state occurs.

4.3.1 Configuration Measurements

Uncertainty of the mobile object configuration is introduced due to the configuration measurement process. We will call S to *the set of possible sensed configurations* for each task state.

Again, results of section 3 will be used here. Sets S of the states with only one basic contact are fully equivalent to CU uncertainty regions of the corresponding C-face including all the uncertainties. Sets S of states with more than one basic contact are obtained by intersecting CU regions of C-faces of each basic contact. Thus, if Z_k, $k = 1, ...n$ represents each of the n C-faces, the region S_i of a state E_i is

$$S_i = \cap_k CU_k \,, \qquad \forall k \mid Z_k \supset E_i \tag{13}$$

[2]Strictly speaking, it would be $\overset{\circ}{CU}$, which represent the open set of CU, but, because of continuity, when CU boundary is the frontier between two state realizations, it is not critical to which of them the boundary belongs. For the sake of simplicity we will consider closed sets.

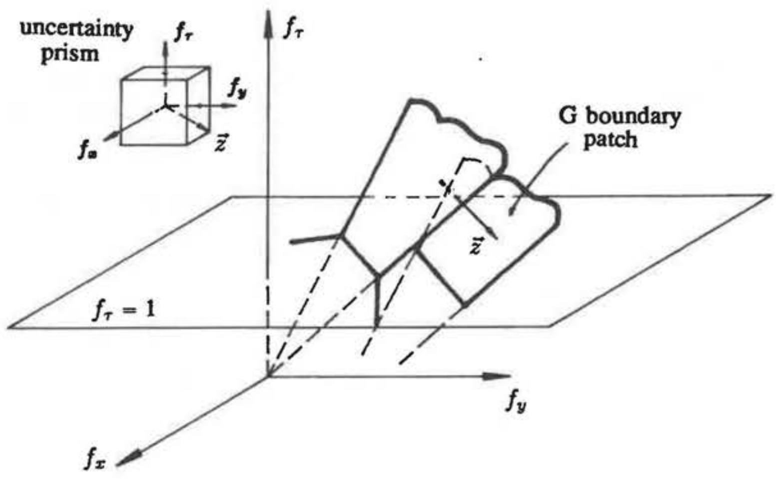

Figure 6: Description of the G boundaries construction.

4.3.2 Generalized Reaction Forces Measurements

We will call G *the set of possible observed generalized reaction forces* that could be sensed during each state occurrence.

Sets G are obtained by adding generalized force uncertainty to the sets of possible generalized reaction forces in the states realization. In order to do it, the boundaries of each of these sets, which consist of planes and conic surfaces in $\mathcal{F}$are obtained. Then, generalized force uncertainty is added; basically, it is equivalent to translate the boundaries of the sets of possible generalized reaction forces in a direction and amount given by the vector, $\vec{z}$, from the center of the uncertainty prism to one of its vertex; this vertex must be the one contained in the same octant as the external normal to the translated boundary (the orientation of the uncertainty prism is fixed) (Figure 6).

5 Fine Motion Planning

5.1 State transition Operators

Once the current state has been identified, during task execution, a proper robot command must be applied to change current state towards the goal one. This implies that each state must have associated *state transition operators*, T, that produce transition to another predicted state. Due to uncertainty, transition to an unique predicted state is unlikely, but a set of possible reachable states can be established.

Operators are in fact robot commands, thus they depend on the type of robot control. Assuming the robot controller is able to work in damping control mode (Whitney, 1977) robot commands are commanded velocities, and so are state transition operators.

The prediction of an object movement with frictional contacts is not a fully solved problem. There are several interesting works on this subject (Erdmann, 1984; Rajan, Burridge and Schwartz, 1987; Brost and Mason, 1989) but none of them considers a general uncertainty model.

Because transition from one state to another depends on the direction of the commanded velocity and not on its module, herein we determine operators T as the set of all movement directions that may permit transition from certain state to each of the contiguous ones, although not all of them will always lead to that transition in the real task execution. The module of T is not relevant here, it must be determined from dynamic and expected forces module considerations.

The set of operators T between any two contiguous states, E_i and E_j, are obtained by computing all movement directions with positive component in the direction normal to the frontier between R_i and R_j; even when $R_i \cap R_j \neq \emptyset$, the normal direction is chosen pointing into R_j from configurations of R_i outside R_j. The obtained set of direction is then enlarged by including velocity uncertainty. The final set may look not too accurate, but the actual solution is bounded to be inside it.

Thus, for each state, a map of transition directions can be obtained, labeling each movement direction (i.e. commanded velocity direction) with the contiguous states that may be reached following it.

5.2 Planning Procedure

With the elements previously introduced and developed, the planning procedure can be divided into the following steps:

1. *Choice of a sequence of states* in N-Graph that links initial and goal state.
2. *Choice of a sequence of state transition operators* that may permit transitions between states in the states sequence from step 1.
3. *Expansion of the sequence of states* from step 1 considering all possible state transitions due to selected operators in step 2. The result will be a subgraph of *N-graph* that we call *E-graph.*
4. *Repetition of steps 1, 2 and 3* considering as initial states those terminal states in E-graph different from the goal.

Step 1. The initial sequence of states can be built using any search algorithm in N-graph. Different criteria can be used to guide this search. Nevertheless, it will probably not succeed exactly during real task execution.

Step 2. A operator T must be selected to advance through the states of the initial states sequence. Again, different criteria can be used to guide the selection (e.g. avoiding certain particular states or minimizing the number of possible reachable states). Moreover, another strategy can be used simultaneously, giving better performance: the set of all operators that may permit the desired transition are selected for each consecutive pair of states, then these sets are intersected to get a common set of operators that will be used instead the original ones. Intersections are done following states sequence, in such a way that if the intersection set becomes null in any point of the sequence, the last non-null intersection set is chosen for the previous transitions and the intersection procedure starts again from that point. By this way, operators to be applied are then selected from the resulting sets, and transitions operators changes can be minimized during task execution.

Step 3. Because each operator T may give rise to a transition to more than one state, those states not selected in the initial sequence must be included as reachables for the selected T, resulting the E-graph. In subsequent applications of this step, E-graph is considered and expanded instead of the initial states sequence.

Step 4. The three previous steps are repeated until there is not any terminal state in E-graph other than the goal one.

The assembly plan is then represented by the final E-graph and the corresponding operators T.

<u>Closed loops in E-graph</u>

Closed loops in the final E-graph appear when the selected operators permit closed sequences of states. Two types of closed loops can be distinguished depending on the associated operators T, we will call *pseudo closed loops* to those in which the associated operators T do not have directions with opposite components (more than 90^o between them), and *repetitive closed loops* to those in which this condition is not verified.

Pseudo closed loops will be automatically broken sooner or later during plan execution, or even they will not appear. Repetitive closed loops may really give rise to vicious circles during plan execution and therefore they must be specially monitored during on-line work. If they actually appear, an alternative plan must be executed beginning from any state of the close loop. If possible, repetitive closed loops must be avoided in the plan.

6 Plan Execution

Plan execution was intended to be simple enough to permit real applications. It consists basically in the identification of the current task state by fusioning configuration and force/torque sensors information and then apply the corresponding operator T until a new state was detected.

Nevertheless, there are some situations that must be on-line considered. Because of uncertainty, when a selected operator T is applied a high generalize reaction force may appear. This means that the commanded velocity makes the mobile object to push against an obstacle, then the object moves too slowly (according to the corresponding damping value) or even it is absolutely quiet in the current state. In this last case, due to damping control, it results $\vec{g} = K\vec{v}$ being $\vec{g}$ the generalized reaction force, K the damping matrix, and $\vec{v}$ the generalized velocity.

These situations are detected from the sensed force/torque information. When reaction forces overcome a prefixed threshold (function of the damping matrix elements) another operator T must be selected. The selection is done by modifying T direction away from the force, this is, by increasing the angle between the sensed generalized force and the applied velocity.

Other particular situations appear when both sensed configuration and generalized reaction force belongs to the corresponding sets of two different states. In this case, the states are indistinguishable with the available information. During task execution, it does not matter which state is taken as current one. Possible transitions from both states have been considered in the plan, thus, applying the corresponding operator T an unambiguous state will be reached.

7 Conclusions

In this paper an automatic fine motion planner to perform assembly tasks in presence of uncertainty was described. The task is modeled as a set of states defined according to the occurrence of different sets of basic contacts. The plan is specified as a graph of states and state transition operators (equivalent to robot commands) to move through them. Plan execution consists in identifying current state and applying the proper operator. A simple on line replanning is necessary when generalized reaction force module exceeds a threshold. All sources of uncertainty that appear in real problems are modeled and considered to determine configurations and generalized forces that can be observed in each state; this information are used to identify current state during task execution.

The planner was developed for movements on a plane. Extension to more degrees of freedom is theoretically possible, but a hard work will be necessary to determine the sensor information (configurations and generalized forces) that can be observed in higher dimensional task states.

ACKNOWLEDGEMENTS

This work was partially supported by *Comisión Interministerial de Ciencia y Tecnología (CICYT)* under the project ROB 89-0287.

REFERENCES

Basañez L. and R. Suárez (1991). Uncertainty Modelling in Configuration Space for Robotic Motion Planning. *Preprints of SYROCO'91* .

Brost R. C. and M. Mason (1989). Graphical Analysis of Planar Rigid-Body Dynamics with Multiple Frictional Contacts. *Fifth Int. Symposium of Robotics Research.*

Buckley, S.J. (1987). Planning and teaching compliant motion strategies. *MIT Art. Intell. Lab. report AI-TR-936 (Ph.D Thesis).*

Canny, J. (1988). The complexity of Robot Motion Planning. MIT Press, Cambridge, Mass.

Dufay, B., and J. Latombe (1984). An approach to automatic robot programming based on inductive learning. *Robotics Research: The First Int. Symposium.* The MIT Press., pp. 97-115.

Erdmann, M. (1984). On motion planning with uncertainty. *MIT Art. Intel. Lab. report AI-TR-810*

Gottschlich S. and A. Kak (1991). Motion Planning for Assembly Mating Operations *Proc. of the 1991 IEEE Int. Conference on Robotics and Automation*, California, USA, 1956-1963.

Inoue H. (1974). Force Feedback in Precise Assembly Task. *MIT Art. Intell. Lab. Memo 308.*

Lozano Perez, T. (1983). Spatial planning: a configuration space approach. *IEEE Trans. on Computers, C-32* (2), 108-120.

Lozano Perez, T., M. Mason, and R. Taylor (1984). Automatic synthesis of fine-motion strategies for robots. *The Int. Journal of Robotics Research, 3* (1), 3-24.

Rajan V.T., R. Burridge and T. Schwartz (1987). Dynamics of a Rigid Body in Frictional Contact with Rigid Walls. *Proc. of the 1991 IEEE Int. Conf. on Robotics and Automation*, North Carolina, USA, 671-677.

R. Suárez (1991). Fuerzas de reacción en contactos básicos con incertidumbre. *Inst. de Cibernética IC-DT-91-03.*

Turk, M.A. (1985). A fine-motion planning algorithm. *Intelligent Robots and Computer Vision - SPIE, 579.*

Whitney, D. (1977). Force feedback control of manipulator fine motions. *Trans. of ASME Journal of Dyn. Syst., Meas., and Control*, June, 91-97.

Xiao J. and R. Volz (1989). On replanning for Assembly Tasks Using Robots in the Presence of Uncertainties. *Proc. of the 1989 IEEE Int. Conf. on Robotics and Automation*, Arizona, USA, 638-645.