AUTOMATIC FINE-MOTION PLANNING BASED ON POSITION/FORCE STATES

R. Suárez and L. Basáñez

Institut de Cibernètica (CSIC-UPC), Barcelona, Spain

Abstract – This paper outlines a new approach to fine motion planning in presence of uncertainty. The concept of position/force space is introduced, and the uncertainty sources briefly reviewed. With these elements a new model of the task is proposed using position/force states, and based on them, the procedure to obtain the plan and to execute it, is described. The method has no theoretical constraints on the number or type of the degrees of freedom, and can be used with different position/force control types.

Keywords – Robots; assembling; automation; fine motion; planning.

INTRODUCTION

The Problem

Assembly tasks with robots requires performing fine motions when the pieces are close up or in physical contact. One of the main problems in dealing with fine motion planning and execution is the uncertainty inherent in real world object dimensions and positions, in sensory information, and in robot positioning. This uncertainty seriously affects the performance of some types of assembly tasks using robots, unless a properly planned strategy is used. Decreasing the amount of uncertainty largely increases the cost of the system, and anyway technical limitations always remain.

This paper outlines a new approach to fine motion planning in presence of uncertainty. A new model of the task based on position/force states is proposed, and the procedure to obtain and to execute the plan is described. The method has no theoretical constraints on the number or type of the degrees of freedom *(dof.)*, or on the position/force control type.

The proposed method forms part of an automatic cell programming and monitoring system for assembly tasks presently under development at the Institute of Cybernetics of Barcelona (Basañez and others, 1988a, 1988b).

Previous Work

Significant contributions dealing with the problem of automatic fine motion planning have been reported.

Mason (1981) provides a way to determine natural and artificial constraints in order to use hybrid control. Dufay (1984) proposes an automatic planning method, dealing with uncertainty, in which, after multiple task executions during a training phase, an induction phase supplies the general plan. Lee (1985) suggests a technique based on two-dimensional cut diagrams to generate the compliance vector for hybrid control. Turk (1985) proposes a fine motion planning algorithm based on geometric states (regions) and assuming damping control.

*This work was partially supported by Fundación Ramón Areces under the project SEPETER

Lozano Perez (1984) proposes a formal approach to the synthesis of compliant motion strategies. It is based on the concept of *pre-image* obtained from the task geometric information, goal positions and commanded velocities. Erdmann (1984) suggests a method for planning motions in presence of uncertainty, based on the concepts of *pre-image* and *back-projection* and, on the same line, Buckley (1987) presents an interactive system to build a compliant motion strategy, and a planner capable of dealing with simple problems.

TASK POSITION AND FORCE MODEL

Configuration Space

Describing the position of a rigid object requires the specification of all its degrees of freedom, both translations and rotations. This can be done by mean of a set of independent parameters, called its *configuration*. The number n of independent parameters is equal to the number of degrees of freedom the object has. The n-dimensional space defined by these parameters is called *Configuration space (C-space)* (Lozano Perez, 1983).

Therefore, the position of a rigid object in the real world is represented by a point in the C-space. This means that the problem of manipulating a rigid body in the real world can be translated into the problem of manipulating a point in the C-space. When there are obstacles in the environment, the object is only free to move in some ranges of its degrees of freedom; thus, only a subspace of the C-space represents valid configurations in which there is no collision with the obstacles. This subspace is called *free space*. The boundary between the free space and the subspace of invalid configurations is represented by hypersurfaces called *C-surfaces*.

The main advantage of transforming the problem of moving a real object among real obstacles into the problem of moving a point among transformed obstacles is that motion constraints appear explicitly, and it is easier to deal with them. However, computing the exact C-space for a high dimensional problem may be a hard work, specially if rotations are involved.

Position/Force Space

The $2n$-dimensional *Position/Force Space (PF-space)* will be defined as an extension of the n-dimensional C-space, by attaching to each point of the C-space the n-dimensional static reaction force that appear when the object becomes in contact with the obstacles.

In the case of rigid object and rigid obstacles, it is clear that there can only be finite non-zero forces in configurations corresponding to C-surfaces, while zero force will be attached to free space configurations and arbitrarily large force will be associated to invalid configurations. In the case of elastic object and obstacles, the valid points of PF-space will depend on the elastic properties, but the above definition also applies. In this paper, we only deal with the rigid case.

Using an appropriate reference system and in static situation, reaction forces will be normal to C-surfaces (Erdmann, 1984), thus force parameter values will depend on C-surfaces.

The set of all points in PF-space whose projections into C-space belong to a C-surface will be called *Contact Subspace (CS)*, and the set of points of PF-space whose projections belong to free space in C-space will be called *PF-free space*.

UNCERTAINTY

Uncertainty Sources

The sources of uncertainty can be categorized into three major groups:

a) *Geometric tolerances in the object dimensions*. All industrial manufactured mechanical parts have tolerances in their dimensions. Three main tolerance specifications can be mentioned: size tolerance, form tolerance and relative features position tolerance (Requicha, 1983).

b) *Inaccuracy in object position*. Relative position between the object to be inserted and the place where it must be inserted could be generated by three causes: inaccuracy in the absolute location of objects in the environment, inaccuracy in the object position in the robot gripper, and inaccuracy in the robot positioning (Brooks, 1982; Day, 1988).

c) *Inaccuracy in forces measurement*. Inaccuracy in forces measurement is due to the limited resolution of force sensors and also to their physical location in the system, which sometimes implies complex transformations to obtain the resultant (e.g. sensors in robot joints).

Forces measurements are often used to decrease position uncertainty. Nevertheless, the need of estimating friction coefficients to reduce orientation uncertainty seriously limits this method.

Groups a) and b) affect position parameters, while group c) affects force parameters.

Uncertainty Model

The uncertainty in the rigid object position relative to the environment is equivalent to the uncertainty in the position parameters of C-space and PF-space. In the same way, the uncertainty in the determination of reaction forces in the real world is equivalent to the uncertainty in the force parameters of PF-space. Therefore, we will consider and model uncertainty directly in the PF-space.

All the uncertainty sources mentioned above must be taken into account together to obtain the *worst case uncertainty values* in both position and force parameters. With these values it is possible to construct uncertainty envelopes giving rise to uncertainty regions in PF-space. This means that the actual static location of a certain point in the PF-space can be any other inside the uncertainty region attached to that point. The form of these uncertainty regions depends on the type of parameters chosen to specify position and reaction forces in PF-space, and also on the desired model of uncertainty. This uncertainty model is based on those described by Requicha (1983) and Benhabib (1987).

Once the uncertainty in the PF-space has been defined, is easy to obtain the subspace of PF-space containing possible contact points in presence of uncertainty. This subspace will be called *Uncertain Contact Subspace (UCS)*, and can be obtained by making the union of the uncertainty regions associated to each point of CS. The "expansion" of CS to UCS implies the diminution of the PF-free space and of the subspace of invalid configurations.

POSITION/FORCE STATES

Position/Force States Definition

Moving through the diminished PF-free space has no risk of collision despite uncertainty, so that movements in this subset can be considered and treated as *gross motion*. On the contrary, moving through UCS, it is not possible to know with precision where a collision will occur and then, purely position control is not adequate in this subspace. Movements and actions tending to solve the task in UCS is what we consider as *fine motion*.

In our approach, UCS is partitioned into subspaces called *Position/Force States (PF-states)*. PF-states must satisfy two properties to avoid ambiguous situations:

a) The union of all PF-states is equal to UCS.

b) Any two PF-states are disjoint.

There are no more restrictions in PF-states selection, but some criteria are necessary in order to do an appropriate partition of UCS into a useful set of states. Assuming rigid objects, the criteria to partition UCS could be:

a) PF-states projections from PF-space over C-space, called *states position projections (SP)* must be disjoints or completely equivalents.

b) Different PF-states with equal SP must have different ranges of force directions.

c) Each PF-states will be associated with some geometric features of the C-space (vertices, edges and faces).

Contiguous PF-states Graph

PF-states will be represented in a *contiguity graph (CGraph)*, in which the nodes are PF-states and the links connect the contiguous ones. Two PF-states with a frontier dimension less than (2n-1) are not considered as contiguous. This CGraph is a useful representation tool for the planning work.

STATE TRANSITION OPERATORS

Definition

A *State Transition Operator (T)* is a command for the robot control system in order to produce the transition from a PF-state to another contiguous one. The form of T depends on the type of the robot control system and on the PF-states definition.

A robot position/force control system can accept both position and force commands, depending on the control scheme adopted and the desired behaviour (Suárez, 1988). PF-states approach can be used to determine possible natural and artificial constraints in task geometry for *hybrid control* (Mason, 1981; Raibert, 1981), as well as the direction of movement for *stiffness* (Salisbury, 1980) or *damping control* (Whitney, 1977).

In this paper, a robot control system working in damping control mode is assumed, therefore T must be a commanded velocity.

Determination of T

Determining T is equivalent to obtain the motion direction parameters and, after that, fix the module of the commanded velocity. Because more than one T can exist between two contiguous states they will be grouped into *State Transition Operators Sets (TS)*. Under this assumption, the rules for automatically obtaining TS directions for the two degrees of freedom problem are given in Appendix A. Directions will be represented by unitary vectors δ, and those permitting sliding on an edge are considered as valid ones to change the state.

FINE-MOTION PLANNING AND EXECUTION

Figure 1 shows a flow chart with the steps of the planning and execution phases. On-line decision work is intended to be reduce to a minimum, in order to allow major operation velocity.

After constructing the PF-states and knowing the procedure to obtaining the associated operators, the next steps must be followed.

PF-states Sequence

A sequence of contiguous PF-states, linking initial and goal PF-states, will be established using any search strategy in CGraph. The initial PF-state can be determined by sensory information, and the goal PF-state can be easily obtained from the final desired conditions. Different criteria can guide the search through CGraph (e.g. minimum PF-states number in the sequence, minimum PF-states number with non-zero force,...).

Operators Sequence

Once a PF-states sequence has been selected, the set of operators (TS) to pass from one state to the following in the sequence must be determined. It is possible to select only a subset of all feasibles state transition operators according to different criteria (e.g. higher directions range or minimum number of possible successor states).

Filtered Operators Sequence

Two consecutive TS may have a partially coincident range of directions, so they can be intersected and then replaced by the intersection result. This operation will be done beginning with the first TS and ending when the last one is reached or when the intersection becomes null. When the procedure ends with the last TS, the intersection set of directions is the goal TS that can solve the task directly. On the contrary, if a null intersection results, the procedure must be reinitialized taking as the first TS the last one considered. This means that to solve the task, a command change must be done when the corresponding PF-state of the sequence is reached. Others criteria can be simultaneously taken into account, e.g. if some states must be specially avoided, the operators that consider them as possible successor states can be discarded.

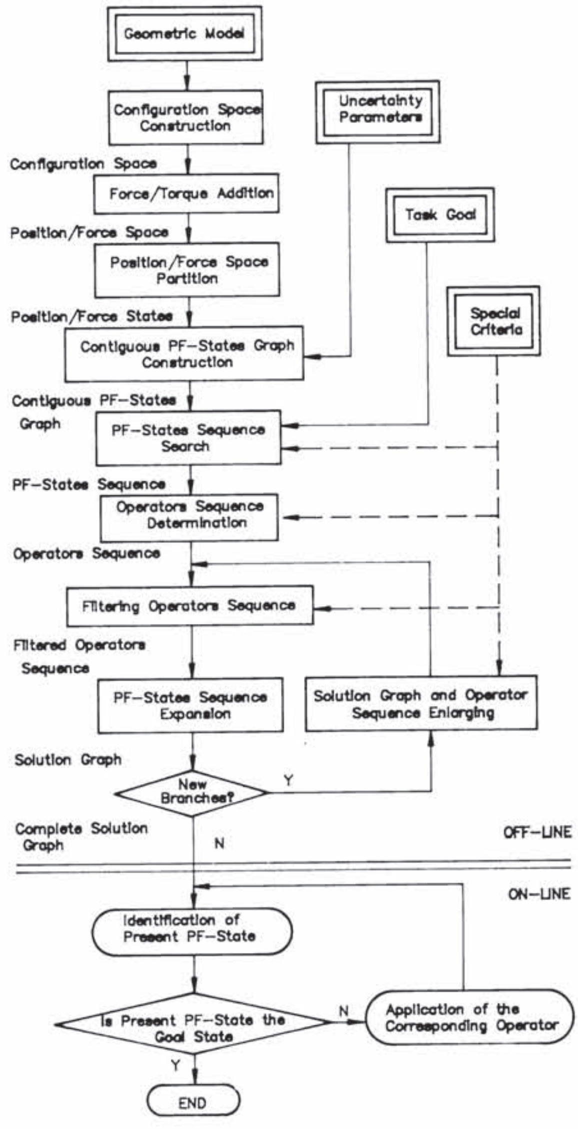

Fig. 1. Planning and executing process flow chart.

Branch PF-states Sequence Expansion

After filtering the operators sequence, the resulting TS may allow for transition to others PF-states besides those of the desired sequence. These PF-states may occur during the task execution so that they must be considered in the plan. This is done taking them as initial states and executing the planning procedure again (PF-states sequence, TS sequence, filtered TS sequence, and branch PF-states sequence expansion). When no more branches appear, the expansion has finished with the result of a directed subgraph of CGraph called *complete solution graph (SOLGraph)*.

SOLGraph may have closed loops of two differents types: *pseudo closed loops* and *repetitive closed loops*. The first ones are those in which the union of their associated TS sequence does not have directions with opposite components (more than 90° between them); this means that although the loops in the SOLGraph actually exist, during plan execution they will be automatically solved by transition operators, or even they will not appear. The second type of loops are those in which the union of their associated TS sequence have directions with opposite components; these loops really give rise to vicious circles in the plan execution and must be monitored during on-line work. If they actually appear, another plan must be executed, taking as initial PF-state one of those in the loop and following a different strategy. If possible, repetitive closed loops must be avoided in the plan.

Task Execution

SOLGraph has the necessary information to guide the execution: PF-states that may appear and TS to go through them. The plan execution consists of identifying the present PF-state from sensory information and applying an operator T from the proper TS until a new PF-state is detected, repeating this process until the goal PF-state is reached. When a repetitive closed loop is expected, the actual PF-state sequence must be monitored in order to detect and avoid vicious circles.

If an unexpected PF-state appears (e.g by accident or any external action) the task must be re-planned from this PF-state. Special error recovering strategies can be formulated to avoid re-planning all the sequence.

SIMPLE CASE EXAMPLE

Suppose the simple task of positioning a square block in a corner (Fig. 2), considering only two translational degrees of freedom. Taking point A of the block as reference, the resulting C-space is shown in Fig. 3. The PF-space is 4-dimensional, so it is illustrated in Fig. 4 as two projections of dimension 3, representing forces by module M and phase Φ. Supposing the worst uncertainty values as U_x, U_y, U_M and U_Φ, the PF-space in presence of uncertainty is shown in Fig. 5. The partition of UCS of PF-space into PF-states is described in Fig. 6, and the resulting CGraph is shown in Fig. 7.

Suppose that the initial PF-state, reached by gross motion, is *a0*. The final desired state is *abAB*, thus, a sequence of PF-states that solve the task is: *a0*, *ab0* and *abAB*.

The operators to follow that PF-state sequence, selected with the criterium of minimun target PF-states number, and the initial filtered result are shown in Fig. 8. Branching PF-states sequence with this resulted TS gives the directed graph in Fig. 9. Repeating the planning procedure from states *aA*, *abA*, and *abB* (search a sequence of states and operators, filtering, and branching) the remaining operators on Fig. 10 are found. The SOLGraph of Fig. 11 is obtained with the attached TS of Fig. 12. So, the task will be solved despite uncertainty using any T from TS showed in Fig. 12. The reduction of the initial filtered TS in Fig. 8 to that in Fig. 12, is the result of intersecting it with the operators to pass from *aA* to *abA*, and from *abA* and *abB* to *abAB* (Fig. 10).

CONCLUSIONS

A new automatic fine-motion planner, specially oriented to insertion tasks in presence of uncertainty, has been proposed. The plan, including uncertainty both in position and force parameters, is elaborated from a special model of the task based on position/force states and a set of operators to change from one state to another. The output is a set of commands for the robot control system that ensures the success in the assembly execution. The approach has no constraints in the number or type of the degrees of freedom, and can be used with different position/force control types.

REFERENCES

Basañez L., R. Kelley, M. Moed, and C. Torras (1988a). A least-commitment approach to intelligent robotic assembly. *Proc. of the 1988 IEEE Int. Conf. on Robotics and Automation*, Pennsylvania, USA, 1318-1319.

Basañez, L., C. Torras, A. Sanfeliu, and J. Ilari (1988b). Automatic cell programming and monitoring through the cooperative interplay of operation specialists. *Proc. of the Second Int. Symposium on Robotics and Manufacturing*, New Mexico, USA, 1067-1074.

Benhabib, B., R. Fenton, and A. Goldenberg (1987). Computer-aided joint error analysis of robots. *IEEE Journal of Robotics and Automation, RA-3* (4), 317-322.

Brooks, R.A. (1982). Symbolic error analysis and robot planning. *The Int. Journal of Robotics Research, 1* (4), 29-68.

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

Day, C. (1988). Robot accuracy issues and methods of improvement. *Robotics Today, 1* (1), 1-9.

Dufay, B., and J. Latombe (1984). An approach to automatic robot programming based on inductive learning. In M. Brady and P. Paul (Ed.), *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 (Master Thesis)*

Lee, C.S.G., and D. Huang (1985). A geometric approach to deriving position/force trajectory in fine motion. *Proc. of the 1985 IEEE Int. Conf. on Robotics and Automation*, St. Louis, USA, 691-697,

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.

Mason, M. T. (1981). Compliance and force control for computer controlled manipulators. *IEEE Trans. on Systems, Man, and Cybernetics, SMC-11* (6), 418-432.

Raibert, M.H., and J.J. Craig (1981). Hybrid position/force control of manipulators. *Trans. of ASME Journal of Dyn. Syst., Meas., and Control, 102*, 126-133.

Requicha, A.A. (1983). Toward a theory of geometric tolerancing. *The Int. Journal of Robotics Research, 4* (2), 45-60.

Salisbury, J.K. (1980). Active stiffness control of a manipulator in Cartesian coordinates. *Proc. of the 19th IEEE Conf. on Decision and Control*, 95-100.

Suárez, R. (1988). Control de fuerza en robótica. *Automática e Instrumentación, 185*, 221-235.

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 1977), 91-97.

APPENDIX A: OPERATORS DETERMINATION

In this Appendix, the rules for automatically obtaining TS directions for the two degrees of freedom problem are given. The following nomenclature will be used:

E : PF-state with attached force.

$\overline{E}$: PF-state with null attached force.

SP : state position projection of E or $\overline{E}$.

n : normal vector to the frontier between the SP of the present PF-state and any contiguous SP, external to the former.

Δd : set of unitary vectors with the directions of the attached forces of E.

d : unitary vector belonging to Δd.

δ : unitary vector representing the operator direction.

$(\ \cdot\ )$: dot product.

$(\ \times\ )$: cross product.

and the following subindices:

p : corresponding to the present PF-state.

s : corresponding to any non-present PF-state with $SP_s = SP_p$.

c : corresponding to any PF-state with SP_c contiguous to SP_p.

Two different cases are initially possible, depending on the present PF-state reaction force:

<u>**Case A.**</u> Present PF-state $\overline{E}_p$

1. δ may produce the transition from $\overline{E}_p$ to $\overline{E}_c$ if it satisfies $(\delta \cdot n) \geq 0$. This condition also applies to transitions to PF-free space.
2. δ may produce the transition from $\overline{E}_p$ to E_s if it satisfies $(\delta \cdot d_s) \leq 0$.

<u>**Case B.**</u> Present PF-state E_p.

Two cases must then be contemplated:

Case B1. E_p is associated only with edges.

1. δ may produce the transition from E_p to $\overline{E}_s$ (contact lost) if it satisfies $(\delta \cdot d_p) \geq 0$.
2. δ may produce the transition from E_p to E_c with $\Delta d_c \cup \Delta d_p \neq \phi$, if it satisfies $(\delta \cdot d_p) \leq 0$ and $(\delta \cdot d_c) \geq 0$.
3. δ may produce jamming in E_p if it satisfies $(-\delta) \in \Delta d_p$.

Case B2. E_p is associated with a vertex. A vertex generates three non-null force states, two (E_p and E_s) are associated with the normal directions to the edges that intersect in the vertex, and the third state (E_s') is associated with the range of directions between those of the others two.

If E_p is associated with a convex vertex the following transitions are possible:

1. δ may produce the transition from E_p to $\overline{E}_s$ (contact lost) if it satisfies $(\delta \cdot d_p) \geq 0$ or $(\delta \cdot d_s) \geq 0$.
2. Idem **B1**-2
3. Idem **B1**-3
4. δ may produce the transition from E_p to E_s if it satisfies $(\delta \cdot l) \geq 0$ and $(\delta \cdot d_s) \leq 0$, where l is a unitary vector that satisfy $(d_p \cdot l) = 0$ and $(d_s \cdot l) \leq 0$.

In the convex vertex case, E_s' is an unstable state, thus no operator to reach it is considered.

If E_p is associated with a concave vertex the following transitions are possible:

1. Idem **B1**-1
2. Idem **B1**-2
3. Idem **B1**-3
4. δ may produce the transition from E_p to E_s if it satisfies $(\delta \cdot k) \geq 0$ and $(\delta \cdot d_s) \leq 0$, where k is a unitary vector that satisfy $(d_p \cdot k) \geq 0$ and $(d_s \cdot k) = 0$.
5. δ may produce the transition from E_p to E_s' if it satisfies $(-\delta) \in \Delta d_s$ or $(-\delta) \in \Delta d_p$ or $((-d_p \times \delta) \cdot (-d_s \times \delta)) \leq 0$.

Table 1 summarizes the operator conditions to change PF-state.

<u>TABLE 1 Operator conditions for the transition between PF-states in the 2 dof. case</u>

<table>
<tr><th colspan="3">To / From</th><th>E_p</th><th>$\overline{E}_s$</th><th>E_s</th><th>E_s'</th><th>$\overline{E}_c$</th><th>E_c</th></tr>
<tr><td colspan="3">$\overline{E}_p$</td><td>n.a.</td><td>n.a.</td><td>$(\delta \cdot d_s) \leq 0$</td><td rowspan="2">n.a.</td><td>$(\delta \cdot n) \geq 0$</td><td>n.a.</td></tr>
<tr><td rowspan="3">E_p</td><td colspan="2">Edges</td><td rowspan="3">$(-\delta) \in \Delta d_p$</td><td rowspan="2">$(\delta \cdot d_p) \geq 0$</td><td>n.a.</td><td rowspan="3">n.a.</td><td rowspan="3">$(\delta \cdot d_p) \leq 0$ and $(\delta \cdot d_c) \geq 0$
with
$\Delta d_c \cup \Delta d_p \neq \phi$</td></tr>
<tr><td rowspan="2">Vertex</td><td>Concave</td><td>$(\delta \cdot k) \geq 0$ and $(\delta \cdot d_s) \leq 0$; with $(d_p \cdot k) \geq 0$ and $(d_s \cdot k) = 0$</td><td>$(-\delta) \in \Delta d_s$ or $(-\delta) \in \Delta d_p$ or $((-d_p \times \delta) \cdot (-d_s \times \delta)) \leq 0$</td></tr>
<tr><td>Convex</td><td>$(\delta \cdot d_p) \geq 0$ or $(\delta \cdot d_s) \geq 0$;</td><td>$(\delta \cdot l) \geq 0$ and $(\delta \cdot d_s) \leq 0$; with $(d_p \cdot l) = 0$ and $(d_s \cdot l) \leq 0$</td><td>n.a.</td></tr>
</table>

n.a. : non applicable

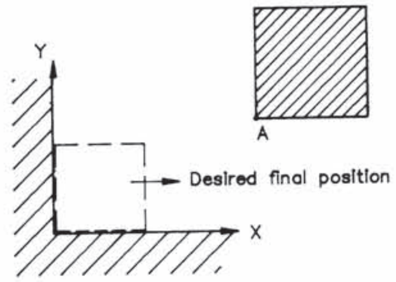

Fig. 2. Real description of the task.

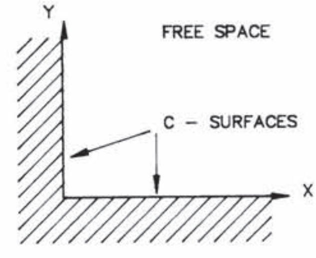

Fig. 3. Configuration space (C-space).

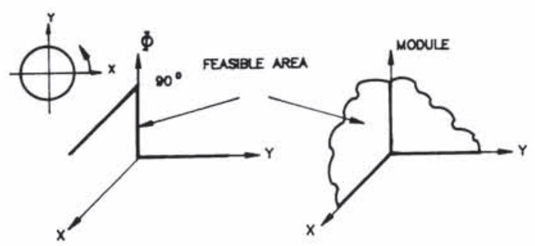

Fig. 4. Position/Force space (PF-space).

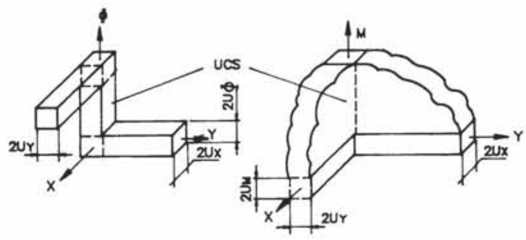

Fig. 5. PF-space with uncertainty.

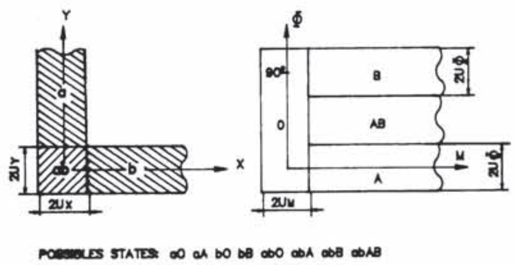

Fig. 6. Position/Force states description (PF-states).

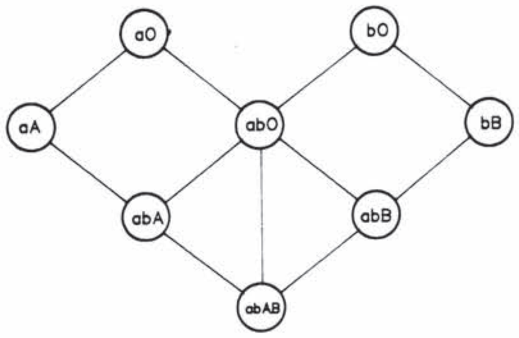

Fig. 7. Contiguous PF-states Graph (CGraph).

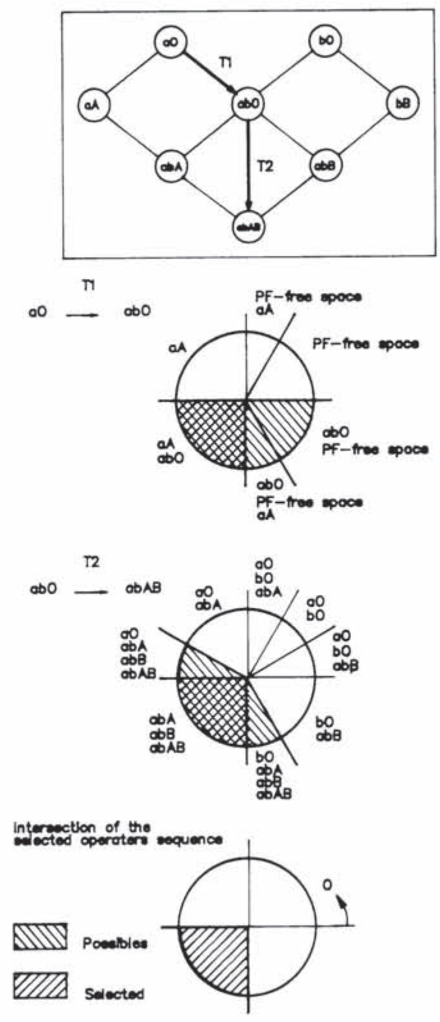

Fig. 8. PF-states sequence and corresponding operators, and operators filtering result.

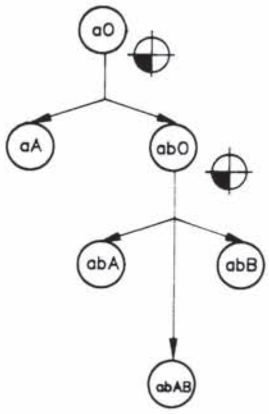

Fig. 9. Branched initial PF-states sequence.

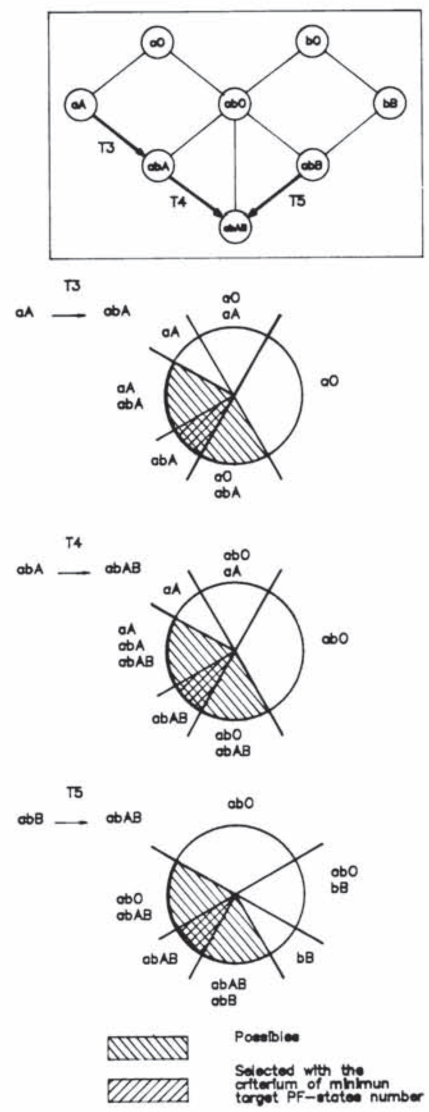

Fig. 10. Operators to pass from states *abA* and *abB* to *abAB*, and from *abB* to *abAB*.

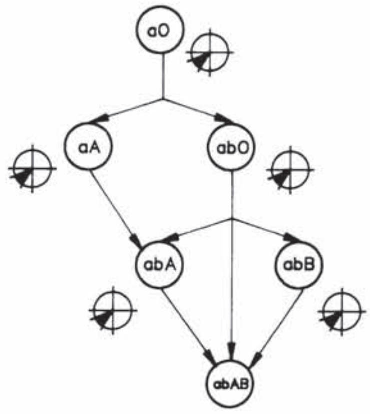

Fig. 11. Complete Solution Graph (SOLGraph).

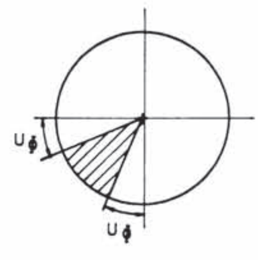

Fig. 12. Directions of the solution operators set.