COMPLIANT-MOTION PLANNING AND EXECUTION FOR ROBOTIC ASSEMBLY[*]

Jan Rosell **Luis Basañez** **Raúl Suárez**

Institut d'Organització i Control de Sistemes Industrials (UPC), Diagonal 647, 08028 Barcelona, SPAIN
Phone: +34 (93) 4016653, Fax: +34 (93) 4016605, e-mails: {rosell,basanez,suarez}@ioc.upc.es

Abstract: This paper presents a method for the planning and execution of compliant motions within the scope of a two-phase fine-motion planner for the performance of planar assembly tasks with robots. Algorithms are provided to find a nominal solution path in both free and contact configuration space which is feasible in spite of the uncertainties affecting the task. Compliant-motion commands based on the generalized damping control mode are synthesized to follow this path, allowing to maintain a constant bounded force.

1 Introduction and overview

Fine-motion planning usually gives rise to an active-compliance strategy that describes geometric trajectories as a function of the current actual situation during the task execution. Three main approaches have been presented following this research line: a) the LMT approach [5] describes the synthesis of compliant motions as the backchaining of preimages from the goal region to the initial region, the preimage for a given velocity command being the set of configurations that guarantee that the goal is reachable and recognizable, taking into account uncertainty in sensing and control; b) two-phase planners first generate a nominal plan assuming no uncertainty, and then consider uncertainty and replan the steps of the path pruning possible errors (e.g. [10]); c) contact-space approaches represent the task as a graph of contact states and synthesize a plan by searching in this graph, considering the uncertainty in the states definition and in the state transition operators (e.g. [9]).

This paper presents a method for the planning and execution of compliant motions (velocities are modified according to reaction forces), which is used in a two-phase fine-motion planner for planar assembly tasks (two degrees of freedom of translation and one of rotation). Section 2 introduces the basic motion planning phase which analyzes all the geometric constraints of a planar assembly task and uses them to obtain an exact cell partition of the free configurations in configuration space ($\mathcal{C}_{free}$). A graph representation of this partition allows the search for a nominal solution path using graph searching techniques. Section 3 introduces the motion synthesis phase, which evaluates the nominal solution path in free configuration space taking into account all the uncertainties affecting the task. When necessary, the arcs of the path are patched in contact configuration space ($\mathcal{C}_{contact}$), where the paths are found analogously to those in $\mathcal{C}_{free}$. The synthesis of motion commands is performed in Section 4. The generalized damping control mode is assumed and compliant-motion commands are synthesized with two components, one devoted to follow the solution path previously found, and the other devoted to maintain the contact taking into account the effect of friction. Task execution issues are tackled in Section 5, where task execution experiments are reported. Finally, Section 6 concludes the work.

2 Basic Motion planning

Let $\mathcal{A}$ and $\mathcal{B}$ be two polygons describing a manipulated object and an static object, respectively. Let $\{W\}$ and $\{T\}$ be the reference frames attached to the workspace and to the manipulated object $\mathcal{A}$, respectively. $\{T\}$ has the origin at the manipulated object reference point, and an orientation ϕ with respect to $\{W\}$. Each vertex of $\mathcal{A}$ is described with respect to $\{T\}$ by a vector $\vec{h}$, with module h and orientation γ. The vertices of $\mathcal{B}$ are described with respect to $\{W\}$ by their coordinates x and y.

Two types of basic contacts can take place: an edge of $\mathcal{A}$ against a vertex of $\mathcal{B}$ (Type-A) and a vertex of $\mathcal{A}$ against an edge of $\mathcal{B}$ (Type-B). The following

[*]This work was partially supported by the CICYT projects TAP96-0868 and TAP98-0471.

subsections analyze the geometric constraints of contact situations involving different combinations of basic contacts, considered only those imposed by the edges and vertices of the basic contacts involved.

2.1 One Basic Contact

Let us define:

ψ_W, ψ_T: the orientation of the normal to the contact edge with respect to $\{W\}$ and $\{T\}$, respectively.

d_W, d_T: the signed distances between the straight line that supports the contact edge and the origins of $\{W\}$ and $\{T\}$, respectively. If (x_e, y_e) is a point of the contact edge, then:

$$\begin{aligned} d_W &= x_e \cos\psi_W + y_e \sin\psi_W \qquad (1)\\ d_T &= x_e \cos\psi_T + y_e \sin\psi_T \qquad (2) \end{aligned}$$

A $\mathcal{C}$-face is defined as the set of contact configurations involving only one basic contact, and a $\mathcal{C}'$-face its parametrized projection into the xy-plane [7, 8]. The $\mathcal{C}'$-face represents the contact positions for each possible contact orientation. For a given orientation ϕ the $\mathcal{C}$-face is a segment with the following features:

- Its supporting line is

$$x \cos\psi_W + y \sin\psi_W = D \qquad (3)$$

where, for type-A basic contacts

$$\begin{aligned} D &= x_v \cos\psi_W + y_v \sin\psi_W + d_T \qquad (4)\\ \psi_W &= \psi_T + \phi + \pi \qquad (5) \end{aligned}$$

(x_v, y_v) being the coordinates of the contact vertex; and for type-B basic contacts

$$D = h_v \cos(\psi_W + \pi - \gamma_v - \phi) + d_W \qquad (6)$$

h_v and γ_v being the module and orientation of the vector defining the contact vertex, respectively, and ψ_W being independent of ϕ.

- Each extreme is on a circumference:

$$\begin{aligned} x &= x_v + h_v \cos(\pi + \phi + \gamma_v)\\ y &= y_v + h_v \sin(\pi + \phi + \gamma_v) \end{aligned}$$

where (x_v, y_v) and $(h_v \cos\gamma_v, h_v \sin\gamma_v)$ are, respectively, the contact vertex and a vertex of the contact edge for a type-A basic contact, and vice versa for a type-B basic contact.

The range of nominal contact orientations is determined by the parallelism condition between the contact edge and the edges adjacent to the contact vertex, which is given by:

$$\psi_W^{sta} = \phi + \psi_T^{mob} + \pi \qquad (7)$$

where ψ_T^{mob} and ψ_W^{sta} represent, respectively, the orientation of the contact edge and the orientation of an adjacent edge of the contact vertex for a type-A basic contact, and vice versa for a type-B basic contact.

2.2 Two Basic contacts

Let a $\mathcal{C}$-edge be the set of contact configurations for a contact situation involving two basic contacts i and j, and a $\mathcal{C}'$-edge its parametrized projection into the xy-plane [7, 8]. The $\mathcal{C}'$-edge is an arc of a curve obtained from the system of equations of the supporting lines of each basic contact given by equation (3). The solution for the $\mathcal{C}'$-edge is:

(a) If $\sin(\psi_{Wj} - \psi_{Wi}) \neq 0$:

$$\begin{aligned} x &= \frac{D_i \sin(\psi_{Wj}) - D_j \sin(\psi_{Wi})}{\sin(\psi_{Wj} - \psi_{Wi})}\\ y &= -\frac{D_i \cos(\psi_{Wj}) - D_j \cos(\psi_{Wi})}{\sin(\psi_{Wj} - \psi_{Wi})} \qquad (8) \end{aligned}$$

(b) Otherwise it is the straight line:

$$x \cos(\psi_{Wi}) + y \sin(\psi_{Wi}) = D_i \qquad (9)$$

for the orientation that satisfies $|D_i| = |D_j|$.

In case (a), each extreme of the $\mathcal{C}'$-edge occurs for a value of the orientation such that one of the contacts satisfies any of the following two constraints:

- *orientation constraint*: the contact edge is parallel to an adjacent edge of the contact vertex.
- *finite length constraint*: the contact vertex coincides with a vertex of the contact edge.

The expression of these values, as well as the value of the orientation in case (b), can be found in [8].

2.3 Three Basic contacts

Let a $\mathcal{C}$-vertex be the contact configuration of a three basic contact situation, involving contacts i, j and k. Then, if $\sin(\psi_{Wj} - \psi_{Wk}) \neq 0$, $\sin(\psi_{Wk} - \psi_{Wi}) \neq 0$ and $\sin(\psi_{Wi} - \psi_{Wj}) \neq 0$ the orientation ϕ of the $\mathcal{C}$-vertex satisfies:

$$\begin{aligned} D_i \sin(\psi_{Wj} - \psi_{Wk}) + D_j \sin(\psi_{Wk} - \psi_{Wi}) +&\\ D_k \sin(\psi_{Wi} - \psi_{Wj}) = 0 &\qquad (10) \end{aligned}$$

otherwise, it is the unique orientation where one of the contact situation involving only two of the contacts occurs. This condition is obtained by solving the system of equations describing the geometric constraints of the $\mathcal{C}'$-edge of contacts i and j and that of the $\mathcal{C}'$-face of contact k. Equation (10) is a second order equation when all basic contacts are of the same type, or a fourth order equation, otherwise [8].

For more than three basic contacts, the corresponding $\mathcal{C}$-vertex coincides with the $\mathcal{C}$-vertex of any subset of three non-redundant basic contacts. These situation is tackled in the following subsection.

2.4 Considering all the Constraints

There can be other constraints than those imposed by the edges and vertices of the involved basic contacts, either due to concave objects or due to the existence of several static objects. The $\mathcal{C}$-space considering these additional constraints can be built by first generating, as described above, the $\mathcal{C}$-faces, the $\mathcal{C}$-edges and the $\mathcal{C}$-vertices, in this order, and by pruning then these sets considering all the constraints:

$\mathcal{C}$-vertex prunning: Eliminate the $\mathcal{C}$-vertices that correspond to configurations that produce an overlapping of the objects, and merge those $\mathcal{C}$-vertices that correspond to the same contact configuration (i.e. a contact configuration involving more than three basic contacts).

$\mathcal{C}$-edge prunning: Divide each $\mathcal{C}$-edge into segments, depending on the number of $\mathcal{C}$-vertices where it is involved, and validate each segment. Eliminate those $\mathcal{C}$-edges without valid segments.

$\mathcal{C}$-face prunning: Divide each $\mathcal{C}$-face into patches, depending on the $\mathcal{C}$-edges where it is involved, and validate each patch. Eliminate those $\mathcal{C}$-faces without valid patches.

The detailed algorithms can be found in [8]. As an example Figure 1 shows the $\mathcal{C}$-space of the assembly task of Figure 2.

2.5 $\mathcal{C}$-space partition

Let $e_1(\phi)$, $e_2(\phi)$ and $e_3(\phi)$ be the configurations of three $\mathcal{C}$-edges of the $\mathcal{C}$-space for a given orientation ϕ. A *$\mathcal{C}$-prism* is defined as the set of configurations $c \in \mathcal{C}_{free}$ that satisfy:

$$c = \alpha e_1(\phi) + \beta e_2(\phi) + \gamma e_3(\phi) \tag{11}$$

with

$$\begin{aligned} \alpha, \beta, \gamma &\in [0,1] \\ \alpha + \beta + \gamma &= 1 \\ \phi &\in [\phi_{bottom}, \phi_{top}] \end{aligned} \tag{12}$$

$[\phi_{bottom}, \phi_{top}]$ being the range of orientations were the three $\mathcal{C}$-edges simultaneously exist and c does not belong to any other $\mathcal{C}$-edge.

The $\mathcal{C}$-prisms are a partition of $\mathcal{C}_{free}$, since they are disjoint regions whose union is $\mathcal{C}_{free}$. Figure 2 shows a section of the $\mathcal{C}$-space for a given orientation, where each triangle represents the section of a $\mathcal{C}$-prism. This exact cell partition is based on [1].

2.6 Planning

Let a $\mathcal{C}$-node be the middle configuration of the border between two $\mathcal{C}$-prisms, and let a $\mathcal{C}$-arc be an arc of curve within a $\mathcal{C}$-prism connecting two $\mathcal{C}$-nodes defined as follows. The configuration c with orientation ϕ of a $\mathcal{C}$-arc between two $\mathcal{C}$-nodes n_i and n_g satisfies:

$$\overrightarrow{ce_1} = \alpha \, \overrightarrow{e_2e_1} + \beta \, \overrightarrow{e_3e_1} \tag{13}$$

with

$$\begin{aligned} \alpha &= \alpha_i + (\alpha_g - \alpha_i)\frac{\phi - \phi_i}{\phi_g - \phi_i} \\ \beta &= \beta_i + (\beta_g - \beta_i)\frac{\phi - \phi_i}{\phi_g - \phi_i} \end{aligned} \tag{14}$$

where ϕ_g and ϕ_i are the orientations of n_g and n_i, respectively, and α_i, α_g, β_i and β_g are determined by (13) for the values ϕ_g and ϕ_i.

A graph is created whose nodes represent $\mathcal{C}$-nodes and whose arcs represent $\mathcal{C}$-arcs. Then, given a goal node, an initial node and a cost function, the Dijkstra algorithm is applied in order to compute the path of minimum cost. The cost of the arcs is set equal to the length of the $\mathcal{C}$-arcs, but some policies can be defined to modify the cost associated to some given arcs in order to guide the search.

The partition and graph representation of $\mathcal{C}_{contact}$ is obtained in a similar way, and then planning in $\mathcal{C}_{contact}$ is done as in $\mathcal{C}_{free}$.

3 Motion Synthesis

3.1 Contact uncertainty analysis

Once a nominal solution path has been found, its feasibility has to be evaluated considering all the uncertainties affecting the task. Contact uncertainty analysis is not in the scope of this paper, but the main ideas are briefly sketched in this Section. The thorough contact uncertainty analysis can be found in [8].

Modelling and sensing uncertainties have been taken into account, which include: a) manufacturing tolerances, b) imprecision in the positioning of the static objects, c) imprecision in the positioning of the manipulated object in the robot gripper, and d) imprecision in the position and orientation of the robot

Uncertainty is handled by associating a Contact Configuration Domain U^S to the current measured robot configuration c_o for each contact situation S, and maintaining the nominal $\mathcal{C}$-space. Then, a contact situation S is compatible with c_o if the intersection of U^S and the set of the corresponding nominal contact configurations is not empty.

Force measurements can help to solve some ambiguous contact situations [2]. A Generalized Force Domain is defined for each contact situation considering the uncertainties that affect the possible directions of the reaction force. Then, a contact situation is compatible with a measured force if the uncertainty region due to imprecision in the force measurements intersects the corresponding Generalized Force Domains. These tests are done using the dual representation of forces [3].

3.2 Path evaluation

In order to evaluate the $\mathcal{C}$-arcs of the path, a discrete number of configuration of each $\mathcal{C}$-arc is classified as follows. A configuration:

- is *uniquely identifiable* if it is assigned to only one contact situation by the contact identification algorithm.
- is *compliant* at a contact situation for a given commanded velocity if for any possible (contact) realization of this configuration, the resulting motion direction moves the manipulated object towards the nominal path [8].

Then, each configuration is classified as:

- *Compliant*: if a contact can take place due to uncertainties and the manipulated object is compliant at it.
- *Guarded*: if it is not compliant but uniquely identifiable.
- *Ambiguous*: if it is not compliant and nor uniquely identifiable.

And each arc is classified into:

- *Compliant*: if all its configurations are compliant.
- *Guarded*: if none of its configurations are ambiguous, and at least one is guarded.
- *Ambiguous*: if there is at least one ambiguous configuration.

A compliant $\mathcal{C}$-arc can be traversed even if the contact situation changes. A guarded $\mathcal{C}$-arc terminates when a new contact situation occurs; then, an error patch plan must be issued in order to recover from the contact situation. An ambiguous $\mathcal{C}$-arc should not be traversed.

3.3 Path synthesis

Given two configurations, either in $\mathcal{C}_{free}$or $\mathcal{C}_{contact}$, the following algorithm is used to find the optimum non-ambiguous path if it exists, or the path with less ambiguous $\mathcal{C}$-arcs.

```
Find-path(C-space ){
    find nominal path
    DO{
        evaluate all the arcs of the nominal path
        IF no arc is ambiguous then RETURN
        ELSE{
            eliminate the ambiguous arcs
            find nominal path
        }
    }until no nominal path exists
    RETURN the path with less ambiguous arcs
}
```

The solution path is synthesized using the following algorithm, which patches the path in $\mathcal{C}_{contact}$ whenever a solution in $\mathcal{C}_{free}$ is not feasible.

```
Path-Synthesis( ){
    C-space construction
    Find-path ($\mathcal{C}_{free}$)
    IF the path is no ambiguous RETURN
    ELSE{
        FOR all the sets of ambiguous arcs{
            Find-path ($\mathcal{C}_{contact}$)
        }
    }
}
```

As an example, figure 3 shows a final solution path which includes $\mathcal{C}$-arcs in both $\mathcal{C}_{free}$ and $\mathcal{C}_{contact}$.

4 Command motion synthesis

The generalized damping control mode is assumed. The velocity commands sent to the robot are computed from two velocity components, $\vec{v_t}$ and $\vec{v_f}$ (Figure 4).

The $\vec{v_t}$ component tries to follow the nominal $\mathcal{C}$-arcs of the solution path (either in $\mathcal{C}_{free}$ or $\mathcal{C}_{contact}$). It is computed as the tangent direction of the $\mathcal{C}$-arc at c_o. The expression of the current $\mathcal{C}$-arc of the solution path being traversed is iteratively updated by substituting n_i by the current measured configuration c_o in equations (13) and (14).

The compliant component, $\vec{v_f}$, has as aim to maintain a constant bounded force during motion in $\mathcal{C}_{contact}$. Given a desired reaction force $\vec{F_d}$, a force control loop with a PID controller is used to generate $\vec{v_f}$ as follows. The input of the PID controller is the force error between the desired reaction force and the actual measured reaction force. The output, multiplied by the predefined accommodation matrix, is the compliant component $\vec{v_f}$.

The desired reaction force $\vec{F_d}$ is computed as follows. Given a contact situation involving one basic contact, if the commanded velocity points towards the $\mathcal{C}$-face, a reaction force arises within the generalized friction cone [4]. If the mobile object moves along an instantaneous direction of motion over the tangent plane, the reaction force lies on one of the edges of the generalized friction cone:

$$\begin{aligned}\vec{e}^{\,-} &= (n_x - \mu n_y,\ n_y + \mu n_x,\\ &\qquad [r_y(n_x - \mu n_y) - r_x(n_y + \mu n_x)]/\rho)\\ \vec{e}^{\,+} &= (n_x + \mu n_y,\ n_y - \mu n_x,\\ &\qquad [r_y(n_x + \mu n_y) - r_x(n_y - \mu n_x)]/\rho) \quad (15)\end{aligned}$$

where (n_x, n_y) and (r_x, r_y) are the normal to the contact edge and the vector from the contact vertex to the manipulated object reference frame, respectively, defined in the workspace reference frame, μ is the friction coefficient, and ρ the radius of gyration of the manipulated object. The proposed $\vec{F_d}$ is in the direction of one of the edges of the generalized friction cone, with a given desired module.

For contact situations involving two basic contacts, $\vec{F_d}$ is in the direction of a linear combination of the corresponding edges of the generalized friction cones:

$$\vec{F_d} = \frac{\beta_i \vec{e_i} + \beta_j \vec{e_j}}{\mid \beta_i \vec{e_i} + \beta_j \vec{e_j} \mid} F_d \text{ with } \beta_i, \beta_j > 0 \qquad (16)$$

5 Task Execution

The assembly task used as example was executed by a Staübli RX-90 robot equiped with a JR3 force sensor. A force control loop with a PID controller has been designed to achieve a good performance in the maintenance of a constant bounded force. The PID parameters were $K_P = 1266.0$, $K_D = 920.6$ and $K_I = 20.6$, and the sampling time was 32 ms. The reaction force module of a type-B contact motion involving translation and rotation is shown in Figure 5. Figure 6 shows snapshots of the real task execution that follows the solution path of Figure 3.

6 Conclusions

The paper has presented the development and implementation of a new fine-motion planner based on a two-phase approach for the performance of planar assembly tasks with robots, as a previous step towards its extension to assembly tasks in the space. The potential application of this research is the automation of complex assembly tasks (i.e. which require several motions to be performed), where the clearance between the objects to be assembled is small with respect to the manufacturing tolerances and uncertainties affecting the task.

References

[1] Avnaim F., Boissonnat J. D., and B. Faverjon, "A practical exact motion planning algorithm for polygonal objects amidst polygonal obstacles", *Proc. of the 1988 IEEE ICRA*, pp. 1656-1660.

[2] Basañez L., Suárez R. and J. Rosell, "Contact situations from Observed Reaction forces in Assembly with Uncertainty", *Proc. of the 13th IFAC World Congress, 1996*, Vol A, pp. 331-336.

[3] Brost R. C. and M. Mason, "Graphical Analysis of Planar Rigid-Body Dynamics with Multiple Frictional Contacts, *Fifth International Symposium of Robotics Research, 1989.*

[4] Erdmann M., "On a representation of Friction in Configuration Space", *The International Journal of Robotics Research*, 13 (3), pp. 240-271, 1994.

[5] T. Lozano-Perez, M.T. Mason and R.H. Taylor, "Automatic Synthesis of Fine-Motion Strategies" *The International Journal of Robotics Research*, Vol.3, No.1, pp.3-24, 1984.

[6] Rosell J., Basañez L. and R. Suárez "Determining Compliant Motions for Planar Assembly Tasks in the Presence of Friction" *Proc. of the 1997 Int. Conf. on Intelligent Robots and Systems*, pp. 946-951.

[7] Rosell J., Basañez L. and R. Suárez, "Embedding Rotations in Translational Configuration Space" *Proc. of the 1997 IEEE ICRA*, pp. 2825-2830.

[8] Rosell J. "Fine-motion planning for robotic assembly under modelling and sensing uncertainties" Ph.D. Thesis, Polythecnical University of Catalonia, 1998.

[9] Suárez R., Basañez L. and Rosell J. Using Configuration and Force Sensing in Assembly Task Planning and Execution *Proc. of the 1995 IEEE Int. Symposium on Assembly Task Planning*, pp.273-279.

[10] J. Xiao and R. Volz, "On Replanning for Assembly Tasks Using Robots in the Presence of Uncertainties", *Proc. of the 1989 IEEE ICRA*, pp. 638-645.

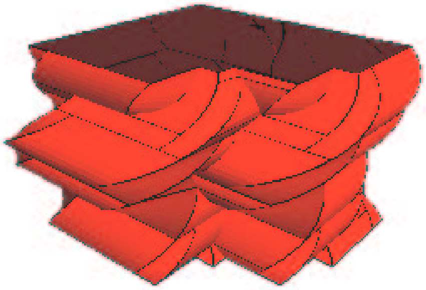

Figure 1: *$\mathcal{C}$-space of the assembly task.*

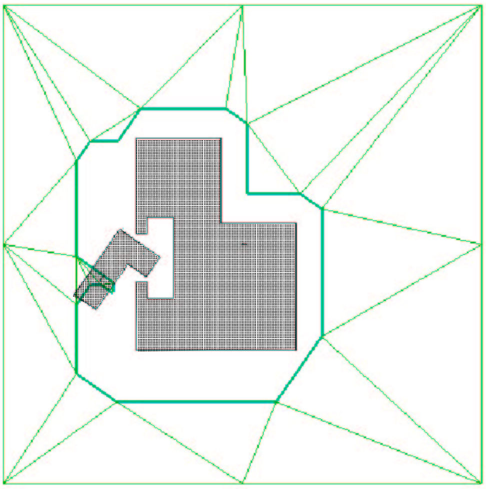

Figure 2: *Slice of $\mathcal{C}$-space showing the sections of the $\mathcal{C}$-prisms that partition $\mathcal{C}_{free}$.*

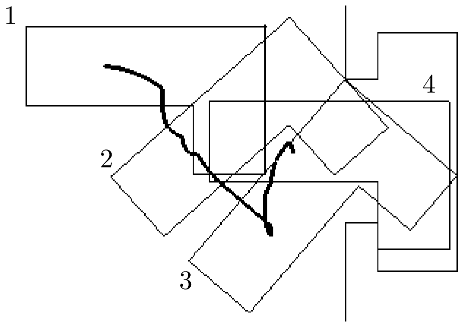

Figure 3: *Solutionpath in $\mathcal{C}_{free}$(from 1 to 2) and $\mathcal{C}_{contact}$(from 2 to 4).*

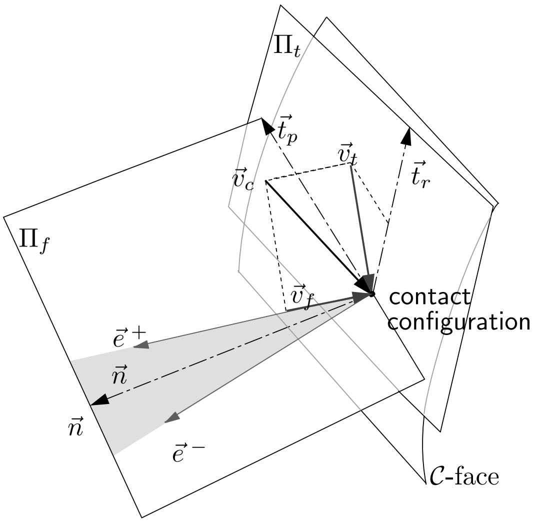

Figure 4: *Commanded velocity decomposition.*

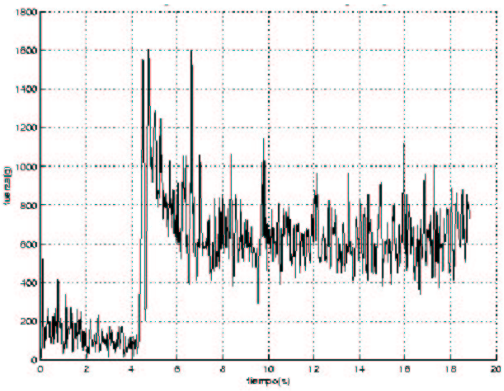

Figure 5: *Reaction force module during a type-B contact motion involving translation and rotation.*

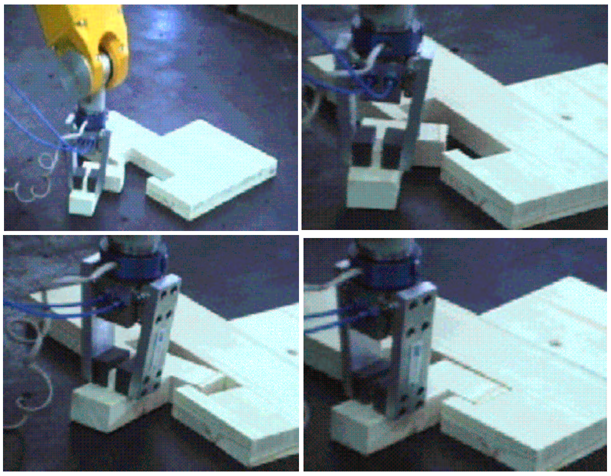

Figure 6: *Snapshot of the real task execution.*