Dpto. de Informática e Ingeniería de Sistemas
Universidad de Zaragoza
C/ María de Luna num. 1
E-50018 Zaragoza
Spain

Internal Report: 1992-02

Dynamic behaviour analysis of non-contact compliant robot motions: Application to edge and plane localization

Sagüés C., Montano L.

If you want to cite this report, please use the following reference instead:
Dynamic behaviour analysis of non-contact compliant robot motions: Application to edge and plane localization, Montano L., Sagüés C., *Robotic and Flexible Manufacturing Systems. Elsevier Science Publishers*, ISBN 0 444 89705 4, pages 57-66, 1992.

This work has been supported in part by project ROB91-0949 from the Comisión Interministerial de Ciencia y Tecnología of Spain and project IT-50/91 from the Diputación General de Aragón.

Dynamic Behaviour Analysis of Non-Contact Compliant Robot Motions: Application to Edge and Plane Localization

Montano L. and Sagüés C.

Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza,
María de Luna 3, 50015 ZARAGOZA, SPAIN.

Abstract

Robot performance in tasks such as contour following or gripper centering is enhanced when its end-effector is provided with range sensors and its controller with the capability of performing non-contact compliant motions. Based on a model obtained by identification techniques, dynamic behaviour and stability of a system composed by a robot with range sensors are studied. This allows to choose the most suitable parameters when carrying out tasks with compliant motions. Applications of these kind of motions to acquisition of edges and planes to locate and recognize objects in the scene are presented.

1 INTRODUCTION

Robots are increasingly required to operate in partially unknown and uncertain workspaces. To improve robot performance in these environments, multisensorial systems are being used. Integration techniques have been proposed to fuse the information obtained from different sources. Some examples can be found in [1], [2], [3]. One of the main goals of multisensor data fusion is to perform object recognition and localization [4].

We have designed and implemented a multisensorial robotic system, named APRIL [5], using a PUMA 560 robot, a 2D vision system, a force/torque sensor-in-wrist, six infrared proximity sensors and two laser sensors on the end-effector. A microcamera is being mounted on the robot hand.

In this work, we focus the attention on the infrared proximity sensors. We consider that the use of *non-contact guarded and compliant motions* based on proximity sensors [6] allows to increase the capabilities of the robot in tasks such as inspection or feature perception for localization and recognition.

Several compliant motion control schemes have been proposed in the literature [7]. Our control system is based on the idea of *generalized damping* [8], extending its application to non-contact motions. In §2 the control scheme is presented. The control parameters depend on the task and on the proximity sensors layout and they are constrained by system stability requirements and the desired dynamic behaviour. In order to carry out an analytical study of the system behaviour, we use identification techniques to obtain a

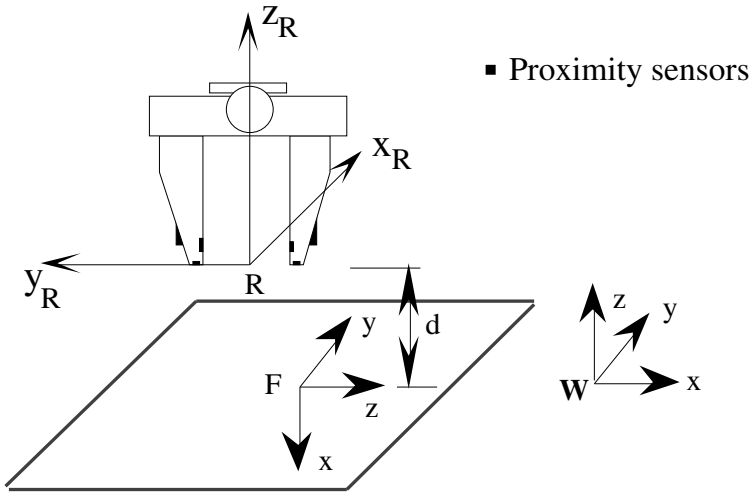

Fig. 1. Proximity sensors on the robot gripper

model of the open loop system which includes the robot, its position controller and the proximity sensors. The methods and criteria used in the identification are exposed in §3. In §4 we analyze the theoretical stability limits and the dynamic behaviour of the closed loop system model, as a function of the control parameters. This allows to choose the most suitable values to achieve the desired motions.
As application, in §5 we present a method to estimate the location of edges and planar surfaces, integrating information provided by a vision system and by proximity sensors, with active sensing strategies which make use of compliant and guarded motions. Perception errors are explicitly considered in the estimation, representing them by means of a probabilistic model.

2 COMPLIANT MOTION SPECIFICATION AND CONTROL

Non-contact compliant motions based on infrared proximity sensorial information, have been implemented in the APRIL robot programming system. We have mounted six sensors, two located on the front of the fingers (RF, LF), two on the internal face (RC, LC) and two on the external face (RE, LE) as shown in Figure 1.
In carrying out compliant motions, we consider two kinds of issues: *task specification*, programmed with the aid of APRIL language primitives, and *motion control*, performed by a control system.
In relation to task specification, the adopted approach is partially based on Mason's theory [8]. According to it, we define: the *compliance frame* (in this case, tool frame R in Figure 1); the *compliant* and *non-compliant degrees of freedom*; the nominal *path* and/or *velocity*; the distance *setpoint* vectors and the motion *end conditions* [6].
As said above, the adopted control scheme for compliant motions is based on the idea of *generalized damping*, extending its application to non-contact motions. Control is achieved by correcting the nominal velocities programmed on the robot controller, as a function of the information obtained from the proximity sensors.

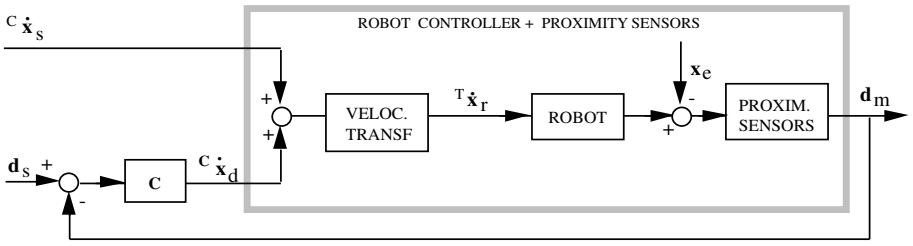

Fig. 2. Control scheme for non-contact compliance

Control for non-contact motions can be mathematically expressed in the following way:

$$ {}^{C}\dot{\mathbf{x}}_d = \mathbf{C}\ (\mathbf{d}_s - \mathbf{d}_m)\ ;\quad {}^{T}\dot{\mathbf{x}}_r = {}^{T}\mathbf{Q}_C\ ({}^{C}\dot{\mathbf{x}}_s + {}^{C}\dot{\mathbf{x}}_d) \tag{1} $$

where each term is defined as follows:

${}^{C}\dot{\mathbf{x}}_d = (\dot{x}_d, \dot{y}_d, \dot{z}_d, \dot{\theta}_{xd}, \dot{\theta}_{yd}, \dot{\theta}_{zd})^T$ distance correction velocity in the compliance frame

$\mathbf{d}_s = (d^s_{RF}, d^s_{LF}, d^s_{RC}, d^s_{LC}, d^s_{RE}, d^s_{LE})^T$ setpoint distance

$\mathbf{d}_m = (d^m_{RF}, d^m_{LF}, d^m_{RC}, d^m_{LC}, d^m_{RE}, d^m_{LE})^T$ measured distance

${}^{T}\dot{\mathbf{x}}_r = (\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z)^T$ corrected velocity in tool frame

${}^{C}\dot{\mathbf{x}}_s = (\dot{x}^s, \dot{y}^s, \dot{z}^s, \dot{\theta}^s_x, \dot{\theta}^s_y, \dot{\theta}^s_z)^T$ velocity setpoint in compliance frame

$\mathbf{C}$ is the correction matrix (control parameter) and ${}^{T}\mathbf{Q}_C$ is the velocity transformation between tool and compliance frames. Figure 2 shows the control scheme for non-contact compliance.

The $\mathbf{C}$ matrix depends on the task and on the proximity sensors layout. We associate the velocity correction in the compliance frame (${}^{C}\dot{\mathbf{x}}_d$) with a specific set of sensors, suitable for the completion of the task. Additionally, the $\mathbf{C}$ matrix influences on the dynamic behaviour and the stability of the system. Below, this influence is analyzed.

3 IDENTIFICATION OF THE DISTANCE PERCEPTION SYSTEM

A open loop model of the plant remarked in Figure 2 has been obtained by identification techniques. It includes the robot controller (motion generation, coordinate transformation and joint servoing) and the proximity sensors. We restrict the study here to tasks in which only frontal sensors are involved (for example, approaching to a surface to estimate its orientation).

Robots are non-linear systems, with coupled and time-varying parameters. However, non-contact motions in the refered tasks are normally accomplished at low speed. This allows to use a time-invariant linear model as an approximation to the real system. Parameter variation which appears when the robot is working close to, or far from its base, is considered in the further analysis by obtaining models in both extreme positions.

The model adopted can be written in a general way as:

$$ \mathbf{y}(k) = \mathbf{M}(q)\ \mathbf{u}(k) + \mathbf{N}(q)\ \mathbf{e}(k) \tag{2} $$

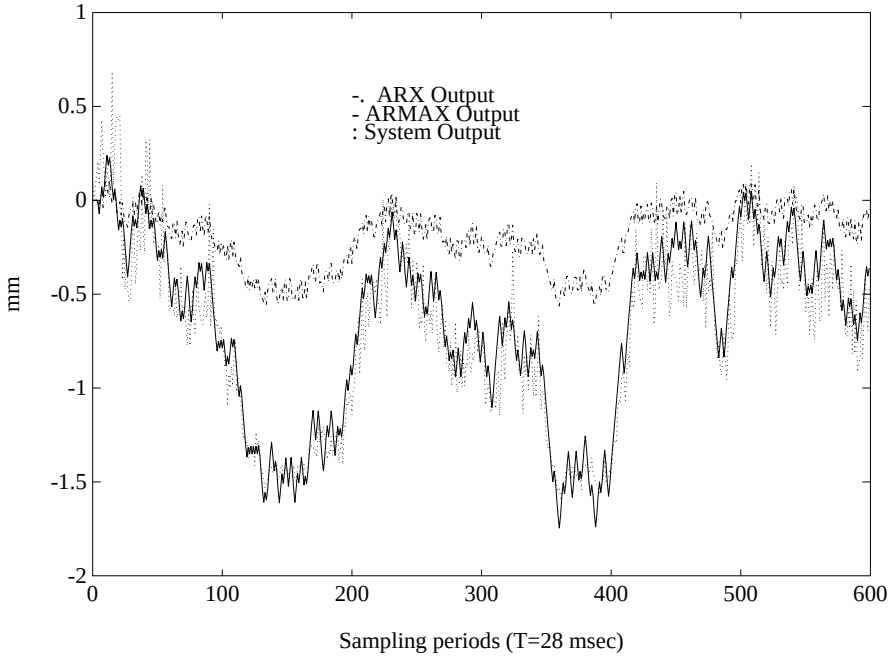

Fig. 3. Output of System, ARX and ARMAX structures

where:
$\mathbf{y}(k) = (d_{RF}(k), d_{LF}(k))^T$ are the distances measured by the two frontal sensors
$\mathbf{u}(k) = (\dot{z}_d(k), \dot{\theta}_{xd}(k))^T$ are the corrections along the z and around the x axes
$\mathbf{e}(k)$ represents white noise.
$\mathbf{M}(q)$ and $\mathbf{N}(q)$ are the transfer matrices:

$$\mathbf{M}(q) = \begin{pmatrix} M_{11}(q) & M_{12}(q) \\ M_{11}(q) & -M_{12}(q) \end{pmatrix} \qquad \mathbf{N}(q) = \begin{pmatrix} N_1(q) \\ N_1(q) \end{pmatrix} \tag{3}$$

We have used parametric identification techniques to find the transfer matrices and we have chosen the general SISO model for the transfer functions in (3):

$$A(q)\ y(k) = \frac{B(q)}{F(q)}\ q^{-nk}\ u(k) + \frac{E(q)}{D(q)}\ e(k) \tag{4}$$

from which we have tried different particular structures: ARX, ARMAX, and Box-Jenkins [9].
Among the classical identification methods, we consider the least-squares criterion (LS), based on the minimization of the prediction error, and the instrumental variable method (IV), based on the idea of prediction errors being uncorrelated with past data. The prediction errors obtained using IV technique are larger than those obtained with the LS technique. Therefore, we only take into account the results obtained by this technique.
In order to identify the system we generate a random binary sequence from a normal distribution. To select the best model order in each structure, the Akaike final prediction error, FPE, and a cross-validation using a loss function V_N (both defined in [9]) with

	a_1	a_2	a_3	b_1	b_2	b_3	e_1	e_2
M_{11} far	-1.0334	-0.0001	0.0343	0.1583	0.7787	0	0.5158	0.1747
M_{11} near	-1.9676	1.2004	-0.2326	0.1801	0.0809	0	-1.6043	0.6343
M_{12} far	-1.6817	0.6250	0.0573	0.0683	-0.2889	0.0389	-1.2525	0.3373
M_{12} near	-1.0544	0.1980	-0.1433	-0.0062	-0.0994	-0.6237	-0.9191	0.0448

Table 1
Parameters estimate of polinomials A(q), B(q) and E(q)

a new data set different from the one used to compute the model have been applied. The analysis of the results from the FPE and V_N functions leads to a first selection of polynomial orders for the three structures.
Comparing the simulations of the system model outputs with similar orders leads to choose the ARMAX structure (Figure 3), being $F(q) = D(q) = 1$ in equation (4).
As these criteria do not suffice to decide on a definitive model order, Bode plots, comparison of simulated outputs between the models and the real system and a cross-correlation test between errors and inputs are made. From the derived results we choose the simplest order which fits well with real behaviour.
Considering an additional time delay due to the computation of the velocity correction from the distance error in the controller, the general expression of the open loop transfer functions is:

$$M_{1j}(q) = \frac{b_1 + b_2 q^{-1} + b_3 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}} \; q^{-4} \tag{5}$$

being $j = 1$ or 2, and a_i and b_i are given in Table 1.

4 CLOSED LOOP DYNAMIC BEHAVIOUR

From the open loop system model obtained above, we study the system stability as a function of the control parameters when using the scheme presented in Figure 2. These parameters are the elements of the **C** matrix, which represent the gains of the robot velocity correction from the distance information.

4.1 State space model

Velocity corrections along the z and around the x axes and the distance error (difference between setpoint and measured distances) of the two frontal sensors are related by **C** (equation 1), which is expressed as:

$$\mathbf{C} = \begin{pmatrix} c_1 & c_1 \\ -c_2 & c_2 \end{pmatrix}$$

To carry out the dynamic behaviour analysis, we transform the representation of the system into a state space model. From the model obtained in §3 we only take the terms of system dynamics, omitting the noise terms. As we have chosen an ARMAX structure,

this simplification does not influence in the theoretical stability limits, because the poles in both transfer matrices $\mathbf{M}(q)$ and $\mathbf{N}(q)$ are equals.
The state and output equations of the system are expressed as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\ \mathbf{x}(k) + \mathbf{G}\ \mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{H}\ \mathbf{x}(k) \end{aligned} \tag{6}$$

$\mathbf{y}(k)$ and $\mathbf{u}(k)$ have been defined in equation (2).
Transforming the transfer matrices model (equation 2) into a state space representation we obtain:

$$\mathbf{F} = \begin{pmatrix} \mathbf{F}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_\theta \end{pmatrix}; \quad \mathbf{G} = \begin{pmatrix} \mathbf{G}_z & \mathbf{0} \\ \mathbf{0} & \mathbf{G}_\theta \end{pmatrix}; \quad \mathbf{H} = (\mathbf{H}_z \ \ \mathbf{H}_\theta)$$

where the traslational and rotational parts are decoupled.

4.2 Stability analysis and dynamic behaviour

If we name $\mathbf{y}_s(k)$ the setpoint distance, the velocity correction $\mathbf{u}(k)$, input to the system, is computed as:

$$\mathbf{u}(k) = \mathbf{C}\ (\mathbf{y}_s(k) - \mathbf{y}(k)) \tag{7}$$

Replacing (7) into (6), we reach to:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\ \mathbf{x}(k) + \mathbf{G}\ \mathbf{C}\ (\mathbf{y}_s(k) - \mathbf{y}(k)) = \\ &= (\mathbf{F} - \mathbf{G}\ \mathbf{C}\ \mathbf{H})\mathbf{x}(k) + \mathbf{G}\ \mathbf{C}\ \mathbf{y}_s(k) \end{aligned} \tag{8}$$

If we define $\mathbf{F}^* \stackrel{\triangle}{=} \mathbf{F} - \mathbf{G}\ \mathbf{C}\ \mathbf{H}$, the bounds of the parameters which make the system stable are computed from the eigenvalues of $\mathbf{F}^*$. This matrix is:

$$\mathbf{F}^* = \begin{pmatrix} \mathbf{F}_z^* & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_\theta^* \end{pmatrix} \tag{9}$$

and the characteristic equations of the decoupled traslational and rotational subsystems are, respectively:

$$\begin{aligned} &(z^5 + a_1 z^4 + a_2 z^3 + a_3 z^2 + 2c_1 b_1 z + 2c_1 b_2) z = 0 \\ &z^6 + a_1 z^5 + a_2 z^4 + a_3 z^3 - 2c_2 b_1 z^2 - 2c_2 b_2 z - 2c_2 b_3 = 0 \end{aligned} \tag{10}$$

As can be seen in (10), the stability of the translational subsystem ($\mathbf{F}_z^*$) only depends on c_1 and that of the rotational subsystem ($\mathbf{F}_\theta^*$) on c_2. Solving equations (10), we obtain an

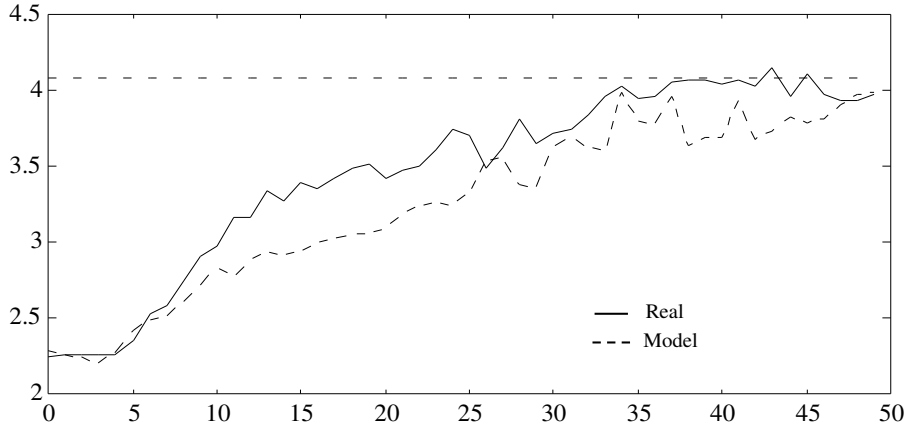

Fig. 4. Real system and model behaviour ($c_1 = 0.031$, $c_2 = 0.05$) for far location

estimate of limit values of the c_i parameters which render the system stable:

$$\mathbf{C}_{far} = \begin{pmatrix} 0.178 & 0.178 \\ -0.196 & 0.196 \end{pmatrix} \qquad \mathbf{C}_{near} = \begin{pmatrix} 0.164 & 0.164 \\ -0.224 & 0.224 \end{pmatrix}$$

No overshooting should arise when carrying out compliant motions. We analyze by means of the root-locus method the values of $\mathbf{C}$ for which the system is either overdamped or underdamped.

In the traslational subsystem, there is a c_1 value for which the dominant poles turn from real to complex, and therefore the subsystem turns from overdamped to underdamped. The estimated values which make the system overdamped are $c_1 < 0.041$ for the far location and $c_1 < 0.031$ for the near location. For the rotational subsystem, the values of c_2 for which the system exhibits a clearly overdamped behaviour are $c_2 < 0.12$ for a near location and $c_2 < 0.05$ for a far location.

In order to accomplish a good and rapid response when the robot end-effector is far from the object surface, the controller takes the largest control values ($c_1 = 1.5$ and $c_2 = 1$) to assure the maximum velocity. When the robot moves close to the objects, the controller commute them to the most conservative values ($c_1 = 0.031,\ c_2 = 0.05$), which render an overdamped behaviour.

In Figure 4, model and real system step responses are drawn. A negligible difference is obtained in the settling time ($t_s \simeq 0.98$ sec.) between simulated and real response at near and far locations. This confirms that, with the robot low speed hypothesis, changes in parameters are negligible from the point of view of settling time, but it must be taken into account for the stability limits.

5 EDGE AND PLANE LOCALIZATION USING COMPLIANT MOTIONS

We use the compliant motions presented above to sense geometric features which are utilized in a geometric recognition and localization process.

We consider polyhedral objects composed of vertices, edges and planar surfaces. A frame attached to each feature [10] expresses its location with respect to a world reference. The information acquired by the sensors has an inherent uncertainty, which we have

represented by a probabilistic model.
From here on, we express a generic location of frame F in the world reference W by a location vector ${}^W\mathbf{x}_F = (p_x, p_y, p_z, \psi_x, \theta_y, \phi_z)^T$, where (p_x, p_y, p_z) is the origin of the frame and $(\psi_x, \theta_y, \phi_z)$ are the orientation parameters (Yaw-Pitch-Roll). To represent the uncertainty we use an incremental transformation ${}^F\mathbf{e} \triangleq (\delta p_x, \delta p_y, \delta p_z, \delta\psi_x, \delta\theta_y, \delta\phi_z)^T$ [11], associated to the feature frame.
The real frame location is obtained as:

$$ {}^W\mathbf{x}_F = {}^W\hat{\mathbf{x}}_F \oplus {}^F\mathbf{e} $$

where ${}^W\hat{\mathbf{x}}_F \triangleq E\{{}^W\mathbf{x}_F\} = (\hat{p}_x, \hat{p}_y, \hat{p}_z, \hat{\psi}_x, \hat{\theta}_y, \hat{\phi}_z)^T$ is the estimated value of ${}^W\mathbf{x}_F$ and $\oplus$ represents the composition of transformations [2], when they are represented as location vectors.
Assuming the hypothesis of Gaussian white noise, the location uncertainty is characterized by its estimated value, ${}^F\hat{\mathbf{e}} = E\{{}^F\mathbf{e}\} = 0$, and its covariance matrix $Cov({}^F\mathbf{e})$. Thus, ${}^W\mathbf{x}_F$ is completely characterized by ${}^W\hat{\mathbf{x}}_F$ and $Cov({}^F\mathbf{e})$.
To carry out data-driven strategies for feature acquisition with proximity sensors, previous data about the object contours obtained from the vision system are supposed. This information is used to drive the robot towards the feature the system is going to observe. The approach trajectory is performed using non-contact compliant motions, which make easier to sense features accomplishing the motion near surfaces whose location is partially unknown, while collisions with objects are avoided. We outline below the strategies to locate a plane and an edge.

5.1 Locating a plane.

The basic strategy to compute a frame associated to a surface proceeds as follows. Robot approaches towards a point on the surface using a compliant motion with a given distance setpoint. When motion is finished, a frame (${}^W\mathbf{x}_{F1}$) with its x axis parallel to the sensed surface is stored. Rotating the end-effector 90 degrees around z axis of the tool reference using a compliant motion, a second frame (${}^W\mathbf{x}_{F2}$) is stored. Finally, the frame ${}^W\mathbf{x}_S$ attached to the planar surface is obtained by composing a 90 degrees rotation ($\mathbf{R}$) of ${}^W\mathbf{x}_{F2}$ around the z axis of ${}^W\mathbf{x}_{F1}$ frame.
The computation of ${}^W\mathbf{x}_S$ can be expressed as:

$$ {}^W\mathbf{x}_S = {}^W\mathbf{x}_{F1} \oplus \mathbf{R} \oplus {}^{F1}\mathbf{x}_{F2} = {}^W\hat{\mathbf{x}}_{F1} \oplus {}^{F1}\mathbf{e} \oplus \mathbf{R} \ominus {}^{F1}\mathbf{e} \oplus {}^{F1}\hat{\mathbf{x}}_W \oplus {}^W\hat{\mathbf{x}}_{F2} \oplus {}^{F2}\mathbf{e} = {}^W\hat{\mathbf{x}}_S \oplus {}^S\mathbf{e} $$

being ${}^S\mathbf{e} = ({}^{F2}J_{F1} \;\; J_R - {}^{F2}J_{F1})\; {}^{F1}\mathbf{e} + {}^{F2}\mathbf{e}$, where ${}^{F2}J_{F1}$ is the transformation Jacobian between $F1$ and $F2$ frames, J_R is the jacobian of $\mathbf{R}$ transformation and $\ominus$ represents the inversion operator of a transformation.
The estimate error of the plane frame (${}^S\mathbf{e}$) is a function of the ${}^{F1}\mathbf{e}$ and ${}^{F2}\mathbf{e}$ errors, which are due to the associated uncertainty in the location of $F1$ and $F2$ frames.

5.2 Locating an edge.

To locate an edge the system sense two points on it. These are chosen as close as possible to the hypothesized tips of the edge, because larger the distance between points, lower the location uncertainty of the edge is.

The proposed strategy to find an edge proceeds as follows. Robot approaches towards a point on the surface using a compliant motion with a given distance setpoint and the end-effector is positioned normal to the plane containing the edge, by using the motions described to detect a plane. With a compliant motion, robot approaches towards a point located near to one end of the edge. Performing a guarded motion in a direction normal to the edge and monitoring the measured distance with one sensor, a point is stored (P1). In a similar way a point (P2) close to the other tip, is detected.
Frame ${}^{W}\mathbf{x}_E$ is computed from both sensed points as:

$$ {}^{W}\mathbf{x}_E = {}^{W}\hat{\mathbf{x}}_E \oplus {}^{E}\mathbf{e} $$

being

$$ {}^{W}\mathbf{x}_E = f({}^{W}\mathbf{x}_{P1}, {}^{W}\mathbf{x}_{P2}) = \left(\frac{p_{x1}+p_{x2}}{2} \ , \ \frac{p_{y1}+p_{y2}}{2} \ , \ \frac{p_{z1}+p_{z2}}{2} \ , 0 \ , \ \alpha \ , \ \beta \right)^{T} \qquad (11) $$

where:

$$ \alpha = atan2 \ \left((p_{z1} - p_{z2}) \ , \sqrt{(p_{x2}-p_{x1})^2 + (p_{y2}-p_{y1})^2} \right) $$

$$ \beta = atan2 \ \left((p_{y2} - p_{y1}) \ , (p_{x2} - p_{x1}) \right) $$

in which $(p_{x1}, \ p_{y1}, \ p_{z1})$ and $(p_{x2}, \ p_{y2}, \ p_{z2})$ are P1 and P2 positions. The error ${}^{E}\mathbf{e}$ is obtained as a function of errors of the sensed points, ${}^{P1}\mathbf{e}$ and ${}^{P2}\mathbf{e}$.
More details about these methods and their application to object recognition and localization can be found in [12] and [13].

6 CONCLUSION

Compliant motions based on proximity sensors can enhance the performance of a robotic system in some tasks. We have presented a non-contact compliant motion control scheme, embedded in a multisensorial robotic system. Control is based on the idea of generalized damping, classically applied to compliance with force sensors.
With the aim of finding the most suitable control parameters for each task, we have studied the stability limits and the dynamic behaviour as a function of the control parameters. As the system parameters vary with the robot location, when it is moving close to object surfaces the controller takes the most conservative values which give an overdamped response. With the selected values, the settling time in a location far from the robot base is larger than it would have obtained if we had set the maximum values computed for each position. However, there is not a significant difference.
Non-contact compliant motions can be applied to tasks such as surface following or gripper centering. In this work we have presented its application to localization of geometric features, used in a geometric recognition and localization process. We have outlined some active sensing strategies to observe edges and planar surfaces, taking into account the measurement errors in the perception system that leads to location uncertainty of the features.

We are actually implementing the presented techniques in the multisensorial system. Some properties of geometric features of the objects (angle between edges, distance between planes,...) are computed with uncertainty from the location uncertainty of the features [13]. These properties are used, in a constraint-based recognition process, to reduce the computational complexity when matching features in the model and features in the scene.

ACKNOWLEDGEMENTS
This work has been supported in part by project ROB91-0949 from the Comisión Interministerial de Ciencia y Tecnología of Spain and project IT-50/91 from the Diputación General de Aragón.

References

[1] H.F Durrant-Whyte. *Integration, Coordination and Control of Multi-Sensor Robot Systems.* Kluwer Academic Pub., Massachusetts, 1988.

[2] R. Smith, M. Self, and P. Cheeseman. Estimating uncertain spatial relationships in robotics. In J.F. Lemmer and L.N. Kanal, editors, *Uncertainty in Artificial Intelligence 2*, pages 435–461. Elsevier Science Pub., 1988.

[3] J.D. Tardós. *Integración Multisensorial para Reconocimiento y Localización de Objetos en Robótica.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, Febrero 1991.

[4] W.E.L. Grimson. *Object Recognition by Computer: The Role of Geometric Constraints.* The MIT Press, Massachusetts, 1990.

[5] L. Montano. *APRIL: Un Sistema Evolucionado de Programación de Robots.* PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, Universidad de Zaragoza, Zaragoza, Spain, September 1987.

[6] C. Sagüés, L. Montano, and J. Neira. Guarded and compliant motions using force and proximity sensors. In *Int. Workshop on Sensorial Integration for Industrial Robots*, pages 274–280, Zaragoza-Spain, 1989.

[7] D.E. Whitney. Historical perspective and state of the art in robot force control. *The Int. Journal of Robotics Research*, 6(1):3–14, Spring 1987.

[8] M. Mason. Compliance and force control for computer controlled manipulators. In M. Brady et al., editor, *Robot Motion: Planning and Control*, pages 373–404. The MIT Press, 1982.

[9] L. Ljung. *System Identification: Theory for the User.* Prentice-Hall, New Jersey, 1987.

[10] J.D. Tardós. Representing partial and uncertain sensorial information using the theory of symmetries. In *IEEE International Conference on Robotics and Automation*, Nice, France, May 1992. (To appear).

[11] R.P. Paul. *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, Mass., 1981.

[12] C. Sagüés and L. Montano. Active sensing strategies with non-contact compliant motions for constraint-based recognition. In *IFAC/IFIP/IMACS-Symposium on Robot Control SYROCO'91*, pages 295–300, Vienna, Austria, September 1991.

[13] C. Sagüés. *Percepción Activa en Reconocimiento Geométrico Basado en Restriccciones*. PhD thesis, Dpto. de Ingeniería Eléctrica e Informática, University of Zaragoza, Spain, Enero 1992.