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

Internal Report: 1991-01

Non-Contact Compliant Robot Motion: Dynamic Behavior and Application to Feature Localization[1]

Montano L., Sagüés C.

If you want to cite this report, please use the following reference instead:
Non-Contact Compliant Robot Motion: Dynamic Behavior and Application to Feature Localization, Montano L., Sagüés C., *IMACS Symposium Modelling and Control of Technological Systems*, pages 457-464, May 1991.

[1] This work has been partially supported by CICYT, Grant PA-86-0028

NON-CONTACT COMPLIANT ROBOT MOTIONS: DYNAMIC BEHAVIOUR AND APPLICATION TO FEATURE LOCALIZATION

IMACS SYMPOSIUM MCTS

Montano L., Sagüés C.
Departamento de Ingeniería Eléctrica e Informática
C. P. S. – Universidad de Zaragoza
María de Luna, 3 – 50015 ZARAGOZA (SPAIN)

Robot performance is enhanced when its end-effector is provided with range sensors and its controller with the capability of achieving compliant motions using sensorial information. Based on a robot-sensor model obtained by identification techniques, the dynamic behaviour and stability of the system are studied as a function of the motion control parameters. This allows us to choose the most suitable parameters for each task. Applications of these kind of motions to acquisition of object geometric features 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], [11], [13]. One of the main goals of multisensor data fusion is to perform object recognition and localization tasks [2].

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 and six infrared proximity sensors on the end-effector [7]. Two micro-cameras and two proximity laser sensors are being mounted on the robot hand.

In this paper, we focus our attention on the proximity sensors. We consider that using *guarded and compliant motions* based on proximity sensors [10], enhances the performance of robotic systems in some tasks. We use them locate individual geometric features (vertexes, edges and planar surfaces), by means of data-driven active sensing strategies.

Several compliant motion control schemes have been proposed in the literature [9], [14]. Our control system is based on the idea of *generalized damping* [4], extending its application to non-contact motions. In §2 the control scheme is presented. Control is achieved by correcting the nominal velocities, programmed on the robot controller, based on information obtained from the proximity sensors.

The control parameters depend on the task and on the proximity sensors layout. Aditionally the control parameters are constrained by system stability requirements and desired dynamic behaviour.

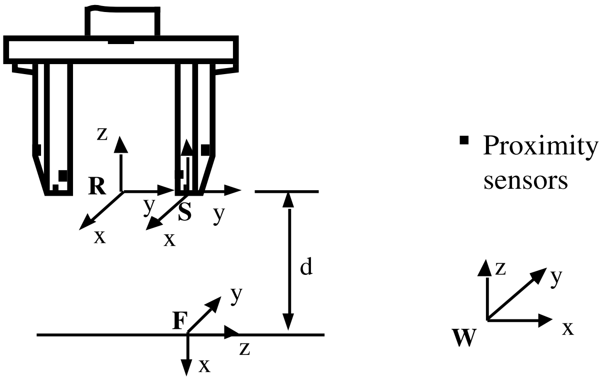


Figure 1: Proximity sensors on the robot gripper

In order to carry out an analytical study of the system behaviour, we use identification techniques [3], to obtain a model of the open loop system which comprises the robot, its position controllers and the proximity sensors. The methods and criteria used in the identification are exposed in §2.

In §3 we analyze the theoretical stability limits and in §4 we study the dynamic behaviour of the closed loop system as a function of the control parameters. This allows us to choose the most suitable values for each task.

As an application, in §5 we present a method to estimate the location of some object geometric features using active sensing strategies based on proximity sensors which make use of compliant and guarded motions. Robot positioning and sensing errors are explicitly considered in the estimation, representing them by means of probabilistic models.

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 (three on each finger), located as follows: two 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*, achieved with the aid of APRIL language primitives, and *motion control*, performed by a control system using sensorial information.

In relation to task specification, the adopted approach is partially based on Mason's theory [4]. According to it, we must define: the *compliance frame*; *compliant* and *non compliant degrees of freedom*; nominal *path* and/or *velocity*; distance *setpoint* vectors and motion *end conditions* [5].

The adopted control scheme for guarded and compliant motions is based on the idea of *generalized damping*, extending its application to non-contact motions. Control is achieved by correcting nominal velocities programmed on the robot MK-II controller, as a function of the information obtained from proximity sensors.

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

$$\boldsymbol{\delta}\dot{\boldsymbol{X}}_c = \mathbf{C}\cdot(\mathbf{d}_s - \mathbf{d}_c)\ ;\ \dot{\boldsymbol{X}}_c = \dot{\boldsymbol{X}}_s + \boldsymbol{\delta}\dot{\boldsymbol{X}}_c \tag{1}$$

where each term is defined as follows:

$\boldsymbol{\delta}\ \dot{\boldsymbol{X}}_c = (\delta\dot{x}, \delta\dot{y}, \delta\dot{z}, \delta\dot{\theta}_x, \delta\dot{\theta}_y, \delta\dot{\theta}_z)^T$ correction velocity in the compliance frame
$\mathbf{C}$ = correction matrix (control parameter)
$\mathbf{d}_s = (d^s_{RF}, d^s_{LF}, d^s_{RC}, d^s_{LC}, d^s_{RE}, d^s_{LE})T$: distance setpoint
$\mathbf{d}_c = (d_{RF}, d_{LF}, d_{RC}, d_{LC}, d_{RE}, d_{LE})^T$: measured distance in the compliance frame
$\dot{\boldsymbol{X}}_c = (\dot{x}, \dot{y}, \dot{z}, \dot{\theta}_x, \dot{\theta}_y, \dot{\theta}_z)^T$ corrected velocity
$\dot{\boldsymbol{X}}_s = (\dot{x}^s, \dot{y}^s, \dot{z}^s, \dot{\theta}^s_x, \dot{\theta}^s_y, \dot{\theta}^s_z)^T$: velocity setpoint

Figure 2 shows the damping control scheme for non-contact compliance. Using sensorial information, a digital controller computes the velocity correction in each sampling period, and the PUMA robot controller adds (integrates) it to the previous corrections.

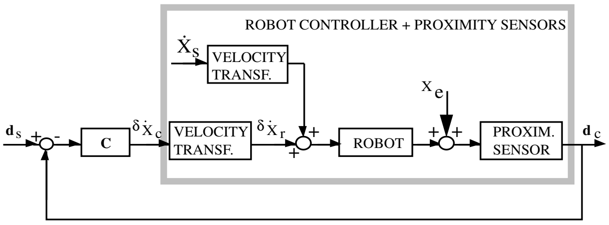


Figure 2: Damping control for non-contact compliance

The selection of the elements of **C** matrix depends on the task and on the proximity sensors layout. That is, we must associate the velocity correction in the compliance frame with a specific set of sensors, suitable for the completion of the task. Additionally, the **C** matrix influences the dynamic behaviour and the stability of the system. Below, this influence is analyzed.

3 IDENTIFICATION OF THE DISTANCE PERCEPTION SYSTEM

With the aim of making an analytical study of the transient response and the stability of the control system as a function of the control parameters (**C** matrix) we have obtained its mathematical model by identification techniques.

The open loop plant to be identified is remarked in Figure 2. It includes the robot controller (motion generation, coordinate transformation and joint servoing) and the proximity sensors. We restrict our study to tasks in which only frontal sensors are involved (for example, approaching to a surface to estimate its orientation).

Some statements about the system model

Robots are systems with non-linear, coupled, and time-varying parameters. However, non-contact motions in the refered tasks are normally accomplished at low speeds. This allows us 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.

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

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

where:

$\mathbf{y}(k) = (\delta d_{RF}(k), \delta d_{LF}(k))^T$ are the distance desviations from the distance setpoint for the two frontal proximity sensors,

$\mathbf{u}(k) = (\delta \dot{z}(k), \delta \dot{\theta}_x(k))^T$ are the velocity corrections along the z and around the x axes of the tool frame (Figure 1),

$\mathbf{e}(k)$ represents white noise.

$\mathbf{M}(q)$ and $\mathbf{N}(q)$ are the transfer matrices:

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

Therefore, $M_{11}(q)$, $M_{12}(q)$ and $N_1(q)$ are the transfer functions to identify.

Model structures

We have used parametric identification techniques to find the transfer matrices and we have chosen the general SISO model structure:

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

where

$$\begin{aligned} A(q) &= 1 + a_1 q^{-1} + \cdots + a_{na} q^{-na} \\ B(q) &= b_1 + b_2 q^{-1} + \cdots + b_{nb} q^{-nb+1} \\ F(q) &= 1 + f_1 q^{-1} + \cdots + f_{nf} q^{-nf} \\ E(q) &= 1 + e_1 q^{-1} + \cdots + e_{ne} q^{-ne} \\ D(q) &= 1 + d_1 q^{-1} + \cdots + e_{nd} q^{-nd} \end{aligned}$$

from which we have tried different particular cases: ARX, ARMAX, and Box-Jenkins structures [3].

The assesment of a model determined by identification is based on the output prediction error. The lower the prediction error, the better the model is. We develope the predictor for the Box-Jenkins structure ($A(q) = 1$), because it is the most general between the tested. The predictor can be written as a linear regression model:

$$\hat{y}(k \mid \boldsymbol{\theta}) = \boldsymbol{\varphi}^T(k, \boldsymbol{\theta}) \cdot \boldsymbol{\theta} \tag{5}$$

Naming

$$\begin{aligned} w(k) &\triangleq \tfrac{B(q)}{F(q)}\, q^{-nk}\, u(k) \\ v(k) &\triangleq \tfrac{E(q)}{D(q)}\, e(k) = A(q)y(q) - w(k) \end{aligned}$$

and being $\varepsilon(k \mid \boldsymbol{\theta}) = y(k) - \hat{y}(k \mid \boldsymbol{\theta}) = \frac{D(q)}{E(q)}\, v(k)$, the vectors $\boldsymbol{\varphi}$ and $\boldsymbol{\theta}$ in (5) are:

$$\begin{aligned} \boldsymbol{\varphi}(k, \boldsymbol{\theta}) = \quad & [u(k - nk), \cdots, u(k - nk - nb + 1), \\ & -w(k-1, \theta), \cdots, -w(k - nf, \theta), \\ & \varepsilon(k-1, \theta), \cdots, \varepsilon(k - ne, \theta), \\ & -v(k-1, \theta), \cdots, -v(k - nd, \theta)]^T \end{aligned}$$

$$\boldsymbol{\theta} = [b_1, \cdots, b_{nb}, f_1, \cdots, f_{nf}, e_1, \cdots, e_{ne}, d_1, \cdots, d_{nd}]^T$$

The ARX model is obtained from (4) making $F(q) = E(q) = D(q) = 1$ and the ARMAX model making $F(q) = D(q) = 1$.

Identification methods considered

Among the classical identification methods we have considered 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 were larger than those obtained with the least squares technique. Therefore, we focus our study on the least squares techniques.

In the LS methods the parameter estimate, $\hat{\boldsymbol{\theta}}_N^{LS}$ is obtained by minimizing the loss function:

$$V_N(\boldsymbol{\theta}, \boldsymbol{Z}^N) = \frac{1}{N} \sum_{k=1}^{N} [y(k) - \boldsymbol{\varphi}^T(k, \boldsymbol{\theta})\boldsymbol{\theta}]^2 \tag{6}$$

from which, the parameter estimate is:

$$\hat{\boldsymbol{\theta}}_N^{LS} = \left[\sum_{k=1}^{N} \boldsymbol{\varphi}\ \boldsymbol{\varphi}^T \right]^{-1} \sum_{k=1}^{N} \boldsymbol{\varphi} y(k) \tag{7}$$

Model selection criteria

In order to identify the system we have generated a random binary sequence from a normal distribution. Its amplitude has been scaled according to system limits (maximum position increment allowed by the robot servos and maximum range measured by the proximity sensors).

Using input and output data, we have tested the model structures presented above. To select the best model order in each structure, the Akaike final prediction-error FPE (as defined in [3]) and a cross-validation using the function V_N with a new data set (different from the one used to compute the model) have been applied.

As these criteria do not suffice to decide on a definitive model order, we compare Bode plots and simulated outputs between the models and the real system. Additionally, a cross-correlation test between errors and inputs is made. From the derived results we choose the simplest order which fits real behaviour.

Experimental results

To carry out the experiments we have generated sequences of 1000 input data. We have found that a larger sequence does not produce significant changes in the identified parameters. The input sequences have been independently applied to $\delta\dot{z}$ and $\delta\dot{\theta}_x$, in two robot workspace locations, one near and other far from its base. The sampling period is fixed by the PUMA controller, $T = 28$ msec. As it will be shown, this period is clearly lower than the settling time of the system.

The analysis of the results from the FPE and V_N test functions leads to a first selection of polynomial orders for the three structures. Figure 3 shows the loss-function V_N for different delays (nk) and different orders (nf) in the Box-Jenkins structure.

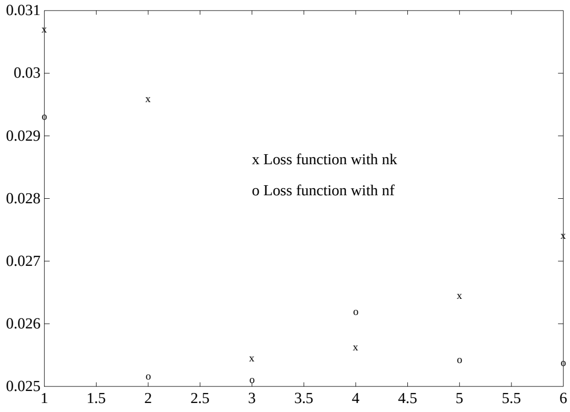


Figure 3: V_N function for several number of delays and several number of poles

Comparing the simulations of the system model outputs with similar orders

(Figure 4), we decide to reject the ARX structure, which does not describe the properties of the disturbance term. Since the Box-Jenkins and ARMAX structures have given similar results, we choose the second one, because of its simplicity.

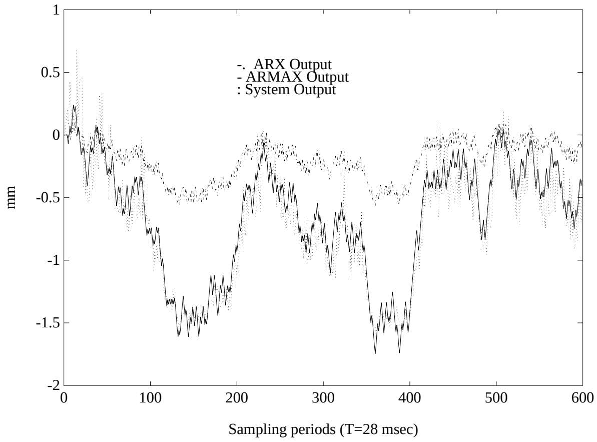


Figure 4: Output of ARX and ARMAX structures with $nk = 3$, $nb = 3$, $nc = 2$, $na = 3$

With the aid of Bode plots and cross-correlation functions between input and prediction errors, we select the best order for each polynomial of the ARMAX structure. Figure 5 shows one of the Bode plots corresponding to the model with two different number of zeros, against real system output.

Final models

From the previous analysis, we obtain the M_{11} and M_{12} transfer functions of the equations (3) for two extreme locations in the robot workspace. In all transfer functions, the number of time delays (nk) and poles (na) is 3.

Table 1 shows the polinomial parameters and their standard deviations for the two inputs in the two robot locations. M_{11} corresponds to $\delta\dot{z}$ input and M_{12} to $\delta\dot{\theta}_x$ input. Due to the integrator of the PUMA controller, the $A(q)$ polynomial obtained has a root on the unit circle.

We must consider an additional time delay due to the computation of the velocity correction in our controller. Therefore, the general expressions of the open loop transfer functions are:

$$\begin{aligned} M_{11}(q) &= \frac{b_1 + b_2 q^{-1}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}} \cdot q^{-4} \\ M_{12}(q) &= \frac{b_1 + b_2 q^{-1} + b_3 q^{-2}}{1 + a_1 q^{-1} + a_2 q^{-2} + a_3 q^{-3}} \cdot q^{-4} \end{aligned} \tag{8}$$

where a_i and b_i are as given in Table 1.

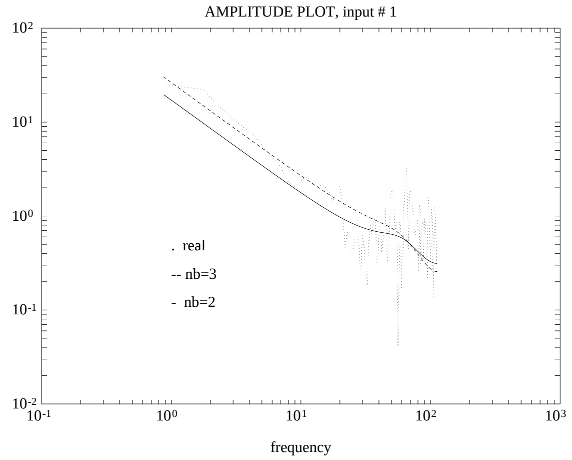


Figure 5: Real and model Bode plots with $nb = 2$ or 3, $nk = na = 3$ and $nc = 2$

4 CLOSED LOOP DYNAMIC BEHAVIOUR

From the open loop system model obtained in §3, we study the system stability as a function of the control parameters when using the generalized damping scheme presented in Figure 2. These parameters are the elements of the **C** matrix, and they represent the gains in the robot velocity correction from the distance information.

The velocity corrections along the z and around the x axes from the distance error of the two frontal sensors are related by **C**

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

which leads to a coupled multivariable system.

To simplify 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 the theoretical stability limits, because the poles of both transfer matrices $\mathbf{M}(q)$ and $\mathbf{N}(q)$ are equals.

		M_{11}	M_{11}	M_{12}	M_{12}
		far	near	far	near
a_1	P	-1.0334	-1.9676	-1.6817	-1.0544
	sd	0.0474	0.0572	0.0587	0.0928
a_2	P	-0.0001	1.2004	0.6250	0.1980
	sd	0.0707	0.0831	0.0954	0.1011
a_3	P	0.0343	-0.2326	0.0573	-0.1433
	sd	0.0303	0.0287	0.0385	-0.026
b_1	P	0.1583	0.1801	0.0683	-0.0062
	sd	0.0308	0.0561	0.0192	0.0664
b_2	P	0.7787	0.0809	-0.2889	-0.0994
	sd	0.0377	0.0763	0.0306	-0.0889
b_3	P	0	0	0.0389	-0.6237
	sd	0	0	0.0291	0.0862

Table 1: Polinomial parameters estimate

State space model

The state and output equations of the system are expressed as:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{F}\cdot\mathbf{x}(k) + \mathbf{G}\cdot\mathbf{u}(k) \\ \mathbf{y}(k) &= \mathbf{h}\cdot\mathbf{x}(k) \end{aligned} \tag{9}$$

where $\mathbf{x}(k)$ is the state vector, $\mathbf{y}(k)$ is the output vector that is the measured distance $(\delta d_{RF}, \delta d_{LF})^T$, and $\mathbf{u}(k)$ is the input vector, made up by the velocity corrections $(\delta\dot{z}, \delta\dot{\theta}_x)^T$.

Transforming the transfer matrices model (2) into a state space representation we obtain the $\mathbf{F}$, $\mathbf{G}$ and $\mathbf{h}$ matrices:

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

where their traslational and rotational parts are decoupled. $\mathbf{F}_\alpha$ and $\mathbf{G}_\alpha$ ($\alpha = z, \theta$) can be expressed in a general way as:

$$
\begin{aligned}
&\mathbf{F}_\alpha = \begin{pmatrix} -a_1 & -a_2 & -a_3 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \\
&\mathbf{G}_\alpha = (1\ 0\ 0\ 0\ 0\ 0)^T \\
&\mathbf{h}_z = \begin{pmatrix} 0 & 0 & 0 & b_1 & b_2 & 0 \\ 0 & 0 & 0 & b_1 & b_2 & 0 \end{pmatrix} \\
&\mathbf{h}_\theta = \begin{pmatrix} 0 & 0 & 0 & b_1 & b_2 & b_3 \\ 0 & 0 & 0 & -b_1 & -b_2 & -b_3 \end{pmatrix}
\end{aligned}
\tag{10}
$$

The a_i and b_i parameters are given in Table 1, for near and far robot locations.

Stability analysis

From the state space equations deduced above, we undertake here the closed loop stability analysis for the damping scheme presented in §2.

Being $\mathbf{y}_s(k)$ and $\mathbf{y}(k)$ the setpoint and the measured distances, the velocity correction $\mathbf{u}(k)$ (input to the system) is computed as:

$$\mathbf{u}(k) = \mathbf{C}(\mathbf{y}_s(k) - \mathbf{y}(k)) \tag{11}$$

Replacing (11) into (9), we reach to:

$$
\begin{aligned}
\mathbf{x}(k+1) &= \mathbf{F}\mathbf{x}(k) + \mathbf{G}\mathbf{C}(\mathbf{y}_s(k) - \mathbf{y}(k)) = \\
&= (\mathbf{F} - \mathbf{G}\mathbf{C}\mathbf{h})\mathbf{x}(k) + \mathbf{G}\mathbf{C}\mathbf{y}_s(k)
\end{aligned}
\tag{12}
$$

If we define $\mathbf{F}^* \overset{\triangle}{=} \mathbf{F} - \mathbf{G}\mathbf{C}\mathbf{h}$, the bounds of the parameters which make the system stable are computed from the eigenvalues of $\mathbf{F}^*$. This matrix can be obtained from $\mathbf{F}, \mathbf{G}$ and $\mathbf{h}$ given in (10) to arrive to the decoupled system:

$$\mathbf{F}^* = \begin{pmatrix} \mathbf{F}_z^* & \mathbf{0} \\ \mathbf{0} & \mathbf{F}_\theta^* \end{pmatrix} \tag{13}$$

The eigenvalues of $\mathbf{F}^*$ are the eigenvalues of $\mathbf{F}_z^*$ and $\mathbf{F}_\theta^*$. The characteristic equations of both subsystems are, respectively:

$$
\begin{aligned}
&(z^5 + a_1 z^4 + a_2 z^3 + a_3 z^2 + 2c_1 b_1 z + 2c_1 b_2)z = 0 \\
&z^6 + a_1 z^5 + a_2 z^4 + a_3 z^3 - 2c_2 b_1 z^2 - 2c_2 b_2 z - 2c_2 b_3 = 0
\end{aligned}
\tag{14}
$$

As can be seen in (14), the stability of $\mathbf{F}_z^*$ only depends on c_1 and that of $\mathbf{F}_\theta^*$ on c_2. Solving equations (14) we obtain an estimate of limit values of the c_i parameters which render the system stable:

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

Figure 6 represents the root-locus for the near location, when c_1 and c_2 change. A similar plot is obtained for the far location.

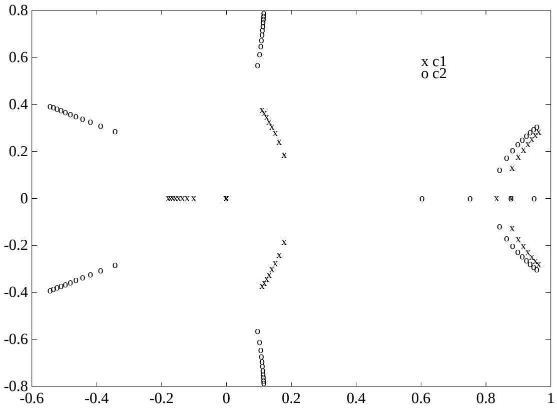


Figure 6: Root-locus for near location

Dynamic behaviour

No overshooting should arise when carrying out compliant motions. We analyze now the values of $\mathbf{C}$ values for which the system is overdamped or underdamped. As can be seen in Figure 6, the system behaviour is dominated by the poles with positive real part.

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 dominant poles always are complex. 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 allowed values of c_1 and c_2 to assure the desired 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 7, model and real system step responses are drawn. The chosen parameters are those which lead to the lowest settling time, without overshooting. 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 settling time response point of view, but it must be taken into account for the stability limits.

An improvement in the transient response can be obtained modifying the

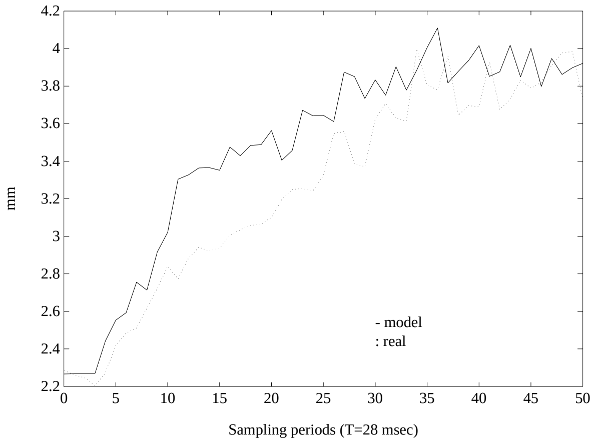


Figure 7: Real system and model behaviour ($c_1 = 0.031$, $c_2 = 0.05$): for far location

controller. We are considering to implement PI or PID controllers in our real-time multiprocessor system, removing the integrator of the PUMA controller. It will render higher stability and lower settling time.

5 FEATURE ACQUISITION USING COMPLIANT MOTIONS

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

Models of the features as well as of the sensors are required. The acquired information has an inherent uncertainty which must also be modeled. We have chosen a probabilistic model to represent it.

We consider polyhedral objects composed of vertexes, edges and planar surfaces. We use a frame attached to each feature [12] to express its location with respect to a world reference.

From here on, we express the generic location of frame F in the world reference W by a vector ${}^W\mathbf{x}_F = (p_x, p_y, p_z, \psi, \theta, \phi)^T$, where (p_x, p_y, p_z) is the origin of the frame and (ψ, θ, ϕ) are the orientation parameters. We choose the Yaw-Pitch-Roll as the orientation angles.

To represent the uncertainty we use an incremental transformation ${}^F\mathbf{e} \triangleq (\delta p_x, \delta p_y, \delta p_z, \delta\psi, \delta\theta, \delta\phi)$ [8], associated to a feature frame F. The true frame location is obtained as:

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

where ${}^W\hat{\mathbf{x}}_F \triangleq E\{{}^W\mathbf{x}_F\} = (\hat{p}_x,\ \hat{p}_y,\ \hat{p}_z,\ \hat{\psi},\ \hat{\theta},\ \hat{\phi})^T$ is the estimated value of

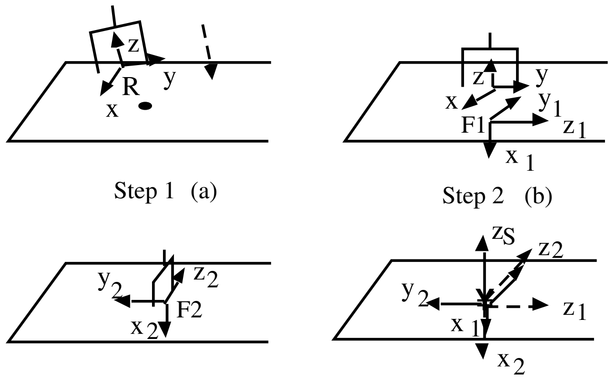
${}^{W}\mathbf{x}_F$, ${}^{F}\mathbf{e}$ and ${}^{W}\mathbf{e}$ are incremental transformations in F and W frames, ${}^{F}J_W$ the transformation jacobian defined in [8], and $\oplus$ represents the composition of transformations, when they are represented as location vectors [11].

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 $Cov({}^{F}\mathbf{e})$.

We propose data-driven strategies for feature acquisition with proximity sensors. We suppose that we have a priori data about the object contours obtained from the vision system. This information is used to drive the robot towards the feature we are going to observe. The approach trajectory is performed using non-contact compliant and guarded motions [10], which make it easier to sense features accomplishing the motion near unknown surfaces, while collisions with objects are avoided. We will outline below the strategies to locate a plane and an edge.

Locating a plane.

The basic strategy to calculate a frame associated with a surface proceeds in the following way (Figure 8):



Step 1 (a)

Step 2 (b)

Step 3 (c)

Step 4 (d)

Figure 8: Sequence to sense a plane

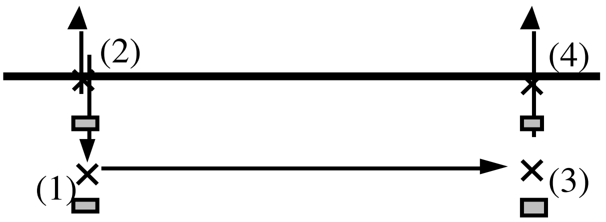


Figure 9: Sensing an edge

1. Approach towards the surface geometric center, using a compliant motion with a given distance setpoint. (Figure 8a).

2. When motion is finished, define a frame (${}^{W}\mathbf{x}_{F1}$) whose x axis in a parallel direction to the sensed surface. We choose this frame to have the z axis along the direction joining the measured points by the two sensors and its origin in the middle of both points (Figure 8b).

3. Rotate the end-effector by 90 degrees around z axis of the tool reference, using a compliant motion. As in step 2, store the frame (${}^{W}\mathbf{x}_{F2}$) attached to a parallel direction to the sensed surface (Figure 8c).

4. Obtain the frame (${}^{W}\mathbf{x}_{S}$) attached to the planar surface by rotating ${}^{W}\mathbf{x}_{F2}$ 90 degrees (R) around the z axis of ${}^{W}\mathbf{x}_{F1}$ frame (Figure 8d).

Following the above steps we compute ${}^{W}\mathbf{x}_{S}$ as:

$${}^{W}\mathbf{x}_{S} = ({}^{W}\hat{\mathbf{x}}_{F1} \oplus R \oplus {}^{F1}\hat{\mathbf{x}}_{F2}) \oplus {}^{F2}\mathbf{e}^{F1} = {}^{W}\hat{\mathbf{x}}_{S} \oplus {}^{S}\mathbf{e}$$

being ${}^{F1}\hat{\mathbf{x}}_{F2} = {}^{F1}\hat{\mathbf{x}}_{W} \oplus {}^{W}\hat{\mathbf{x}}_{F2}$ and ${}^{S}\mathbf{e} = {}^{F2}J_{F1}{}^{F1}\mathbf{e} + {}^{F2}\mathbf{e}$, where ${}^{F2}J_{F1}$ is the transformation Jacobian between $F1$ and $F2$ frames.

The estimate error of the plane frame (${}^{S}\mathbf{e}$) is function of the ${}^{F1}\mathbf{e}$ and ${}^{F2}\mathbf{e}$ errors, which are due to the associated uncertainty to active sensing of the $F1$ and $F2$ frames.

Locating an edge.

To locate an edge we sense two points on it. We choose them as close as possible to the known ends of the edge, because location uncertainty of these points has less influence on the location uncertainty of the edge. The proposed strategy to find an edge proceeds in the following way (Figure 9):

1. Approach towards the geometric center of the surface using a compliant motion with a given distance setpoint.

2. Position the robot end-effector normal to the plane containing the edge. To perform it, we use the motions described to detect a plane.

3. By using a compliant motion, approach to a point located near to one end of the edge (P1). Its location error is ${}^{P1}\mathbf{e}_{p}$.

4. Perform a guarded motion, in a direction an normal to the edge, monitoring with one sensor the measured distance to the plane. A point is stored (P2) when a sudden distance change is detected. Its location error is ${}^{P2}\mathbf{e}_{p}$.

5. Return to the initial point P1.

6. Start a compliant motion towards the other end of the edge (P3).

7. By means of a similar motion than 4, locate the second point (P4).

Detection of the two points suffice to locate an edge. We obtain the edge frame (${}^{W}\mathbf{x}_E$) from both points as:

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

being

$$ {}^{W}\hat{\mathbf{x}}_E = (\hat{p}_{x1}, \hat{p}_{y1}, \hat{p}_{z1}, 0, atan2\left(\frac{\hat{p}_{z1} - \hat{p}_{z2}}{\hat{p}_{x2} - \hat{p}_{x1}}\right),\ atan2\left(\frac{\hat{p}_{y2} - \hat{p}_{y1}}{\hat{p}_{x2} - \hat{p}_{x1}}\right))^T $$

where P1 position is $(\hat{p}_{x1},\ \hat{p}_{y1},\ \hat{p}_{z1})$ and P2 position is $(\hat{p}_{x2},\ \hat{p}_{y2},\ \hat{p}_{z2})$. The error ${}^{E}\mathbf{e}$ is a function of ${}^{P1}\mathbf{e}_p$ and ${}^{P2}\mathbf{e}_p$.

The feature location errors are characterized by its mean ($\hat{\mathbf{e}} = 0$) and its covariance matrix ($Cov(\mathbf{e})$). More details about these methods and their application to object recognition and localization can be found in [6].

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 embebbed in a multisensorial robotic system. Control is based on the idea of generalized damping.

With the aim of finding the most suitable control scheme 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 in any workspace location close to the object. With the selected values, the settling time in a far location from the robot base is larger than it would be obtained if we had set the maximum computed. However, it is not a significant difference. In order to obtain a good and rapid response when the robot is far from objects, the controller takes the largest values of c_1 and c_2 to assure the desired velocity.

Besides the application of the compliant motions to tasks such as surface following or gripper centering, these kinds of motions is applied to acquisition of geometric features, which are utilized in a constraint-based recognition and localization process. We have presented some active sensing strategies to observe features such as edges and planar surfaces, taking into account both robot location and proximity sensor errors.

ACKNOWLEDGEMENTS

This work has been partially supported by CICYT, Grant PA-86-0028

References

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

[2] W. E. L. Grimson and T. Lozano-Pérez. Model-based recognition and localization from sparse range or tactile data. *Int. J. Robotics Research*, 3(3):3–35, 1984.

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

[4] 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.

[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] L. Montano and C. Sagüés. Active sensing using proximity sensors for object recognition and localization. Research report, Departamento de Ingenieria Eléctrica e Informática, Universidad de Zaragoza, 1990.

[7] L. Montano, J.D. Tardós, C. Sagües, and J. Neira. Entorno de programación y control para un sistema robótico multisensorial. In 1^{er} *Congreso de la Asociación Española de Robótica*, pages 365–375, Zaragoza-Spain, Noviembre 1989.

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

[9] M. H. Raibert and J.J. Craig. Hibrid position/force control of manipulators. *Journal of Dynamic Systems, Measurement, and Control*, pages 126–133, 1981.

[10] 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.

[11] 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.

[12] 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.

[13] J.D. Tardós and M. Silva. Multisensor object recognition using a hypothesis-verification scheme. In *Proc. Int. Workshop on Sensorial Integration for Industrial Robots*, pages 265–272, Zaragoza,Spain, November 1989.

[14] Y. Xu and R.P. Paul. On position compensation and force control stability of a robot with a compliant wrist. In *IEEE Int. Conf. on Robotics and Automation*, pages 1173–1184, 1988.