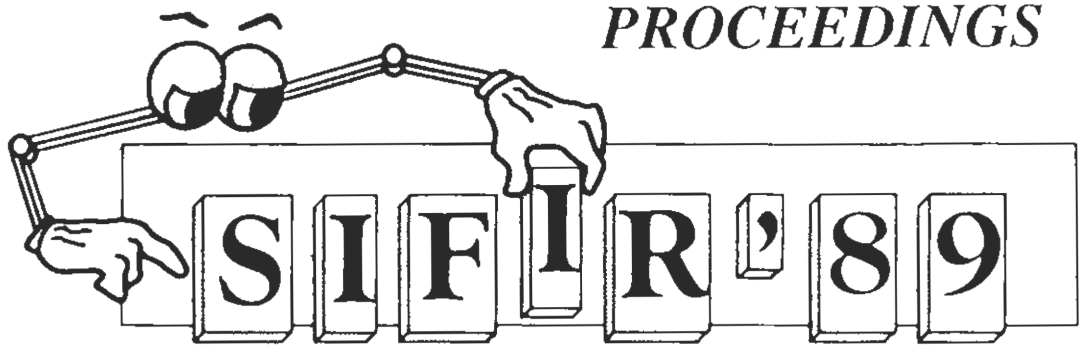
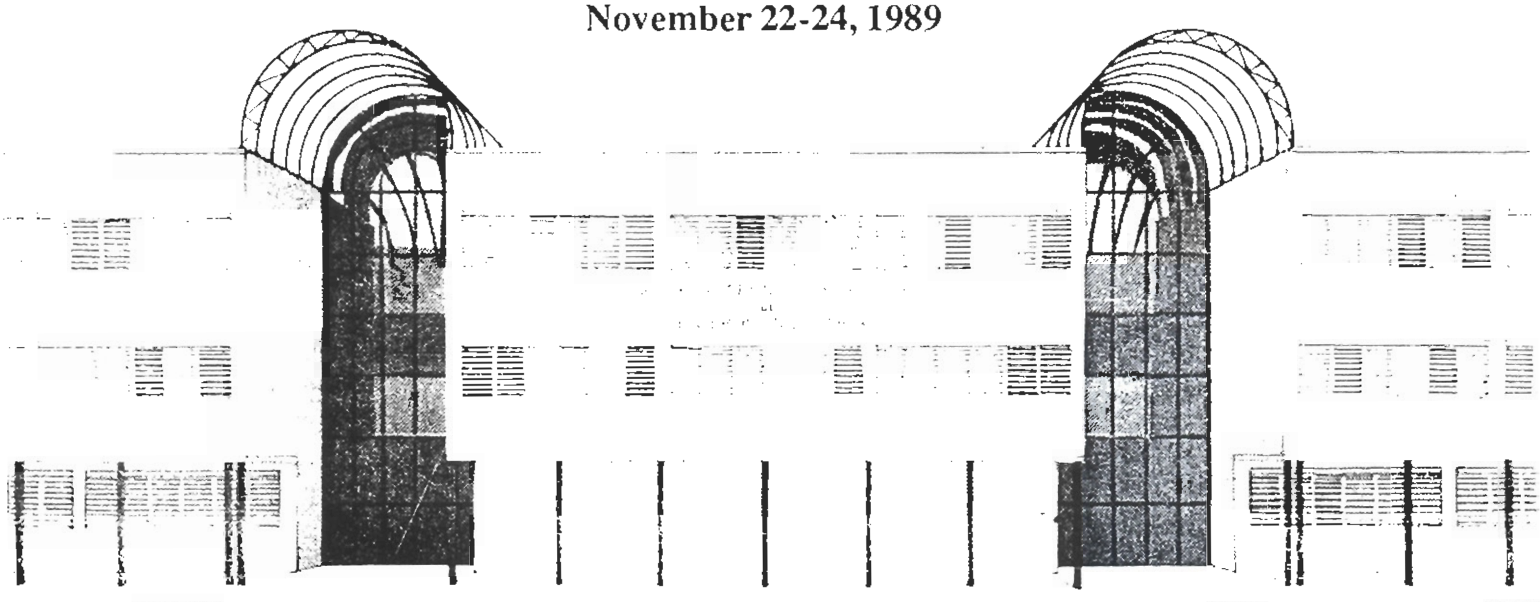
PROCEEDINGS



INTERNATIONAL WORKSHOP ON SENSORIAL INTEGRATION FOR INDUSTRIAL ROBOTS:
Architectures & Applications

ZARAGOZA, SPAIN
November 22-24, 1989



UNIVERSIDAD DE ZARAGOZA
DEPARTAMENTO DE INGENIERIA ELECTRICA E INFORMATICA

GUARDED AND COMPLIANT MOTIONS USING FORCE AND PROXIMITY SENSORS

Sagüés C., Montano L., Neira J.

Dpto. Ingeniería Eléctrica e Informática
Grupo Ingeniería de Sistemas e Informática
E.T.S.I.I. - Universidad de Zaragoza
C/ María de Luna, 3
50015-ZARAGOZA

ABSTRACT

This paper presents a definition and implementation of guarded and compliant motions, using force/torque and infrared proximity sensors. We establish a parallelism between contact and non-contact tasks, in specification and control. To clarify the ideas presented, an example of contact and non-contact tasks is discussed. It shows the task specification issues and a cualitative choice of the control parameters.

Implementation has been carried out in a programming and control system, which is being developed in our laboratory.

1. INTRODUCTION

The automation of manufacturing and assembly tasks, in which robot motion is constrained by the environment geometry, requires the control of contact forces. Several papers have dealt with the analysis of this kind of tasks [4], [11], and a number of motion control schemes have been proposed [8], [9], [10].

Up to now, non-contact tasks have been seldom discussed in literature [1]. These can be useful in surface inspection, centering before grasping, as an aid to object detection and recognition processes, etc.

We consider that guarded and compliant motion analysis and implementation can be extended to non-contact tasks, treating them in a similar way. In such cases, the distance to objects must be controlled.

This paper presents a definition and implementation of guarded and compliant motions, establishing a parallelism between contact and non-contact tasks, in specification and control.

Task specification must be made taking into account the task itself, which objects are involved and how these constrain the motion of the robot. Motion control in contact tasks has been implemented using a generalized damping scheme [3], including force setpoints definition. This scheme has been adapted to non-contact tasks.

The ideas presented here have been implemented in a programming and control system (named APRIL), which is being developed in our laboratory. It is designed to control and manage a multisensorial robotic system [6], constituted by a PUMA 560-MKII robot and force/torque, proximity and 2D-vision sensors.

Section 2 gives a brief presentation of the programming system and the force and proximity sensors used in motion control. Section 3 discusses task specification and control. Section 4 considers an example of two tasks: one with contact and the other whithout. Section 5 describes the computing architecture used.

2. CONTROL AND PROGRAMMING ENVIRONMENT

In this section we present a general description of the system in which guarded and compliant motions discussed here, have been implemented.

2.1. Control and programming system APRIL

APRIL (A Programming system for Robots of IndustriaL type), actually being implemented [5], [6], has been designed to control and coordinate a multisensor robotic system. Its main objectives are:

- to provide an evolved textual explicit programming language (robot level)
- to coordinate the robot and the different sensors that constitute the system
- to support an "intelligent" robotic subsystem (task level)

Figure 1 shows the general frame in which APRIL has been conceived. As a robot programming language, APRIL is considered of the explicit textual level. It has been developed as an extension of a general purpose language (MODULA-2). Therefore, the system provides a set of libraries, which include primitives for robot and sensor control, data types (objects, transformations, etc) and functions to manipulate them. A dynamic world model, based on the idea of *object*, is defined. This explicit representation of objects

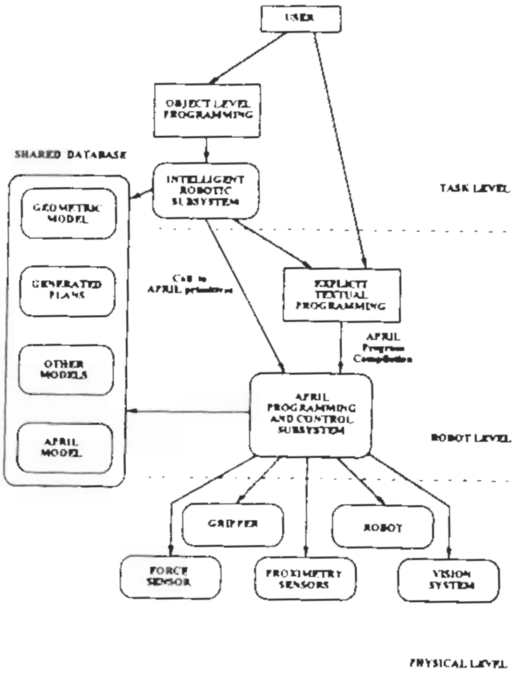


Figure 1. General system description.

allows the incorporation of referencial, geometric or perceptual models.

The subjacent idea of its design is to make the robotic system more flexible, in the sense of it being able to adapt to different tasks and to partially unknown environments. Great importance has been given to sensor control and coordination, and to establish mechanisms to interpret and use sensorial information in real time.

Furthermore, guarded and compliant motions have been explicitly considered, supplying primitives to execute tasks involving these motions. To acomplish them, force/torque and proximity sensors are used. They are presented below.

2.2. Sensors

Force/Torque Sensor

Two force/torque sensors have been designed: one to be located in the robot wrist and the other on the work table [2]. At this moment, experimentation has been carried out with the first one; the other one is under construction.

The wrist sensor has a Malta Cross shape (Fig. 2a). It is made of an aluminium-magnesium alloy (AlMg5). This material has low specific weight and hysteresis and high linearity. Six pairs of extensiometric gauges are adhered to the cross radius in order to measure the deformation of the structure when it is subjected to forces and torques. The two gauges, one on each side of the radius, are connected to a Wheastone bridge, so that their deformations are complementary.

When the sensor is at rest, the bridge is balanced. Deformations induce a change in the gauges resistance, which can be measured by means of the unbalance tension of the bridge. An amplification device has been set on a small card on the same sensor, connected to the output of the bridge. An analogic filter and an A/D conversion card completes the acquisition set.

The six gauge pairs allow to uniquely calculate the six components (three forces and three torques) of a force vector associated to the sensor reference. The relationship between the acting forces on the robot wrist ($F = (f_x\ f_y\ f_z\ m_x\ m_y\ m_z)^T$) and the signals detected on the gauges ($S = (s_1\ s_2\ s_3\ s_4\ s_5\ s_6)^T$) can be expressed in matricial form as follows:

$$F = K \cdot S$$

The K matrix is called calibration matrix and it is obtained experimentally. The maximum forces that the sensor can stand are $f_x = f_y = f_z = 6$ kg, $m_x = m_y = 40$ kg.cm and $m_z = 20$ kg.cm. The maximum experimental errors that have been detected are 20 gr. in force and 100 gr.cm in torque.

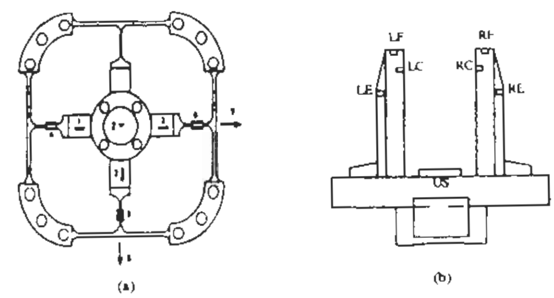


Figure 2. (a) Force/Torque sensor; (b) Proximity sensor.

Proximity Sensors

Two types of proximity sensors are used: ***ultrasound*** and *infrared*. The ultrasound sensor, capable of detecting objects in front of it, has been installed on the base of the gripper. It is an electrostatic Polaroid sensor (a metal sheet which vibrates when being attracted or repelled by an alternate electric field) and its operation is based on the measurement of the time elapsed between the emission and the detection of ultrasonie pulses, which are reflected by the object to be detected.

This sensor has a 1.8 mt. effective range. Since the ultrasound generation phenomenon is reversible, the elements used for emission are also used for detection, determining a dead zone (30 cm. in our case) in which the sensor is unable to measure distances.

Each of the six infrared sensors is composed of a modulated infrared radiation emitter, and a phototransistor that detects part of the luminous intensity reflected by the object. When a luminous ray strikes a surface, it is reflected more or less intensely, depending on the ray reflection coefficient and the surface orientation with respect to it. These factors influence the detection process in such a way, that a previous calibration is needed to obtain a measurement curve for each type of surface that the sensor is required to detect. Up to 4 cm. distances can be detected with these sensors.

The physical layout of the six infrared sensors on the gripper is represented in Fig. 2b. The sensors are named RF, LF, RC, LC, RE, LE (right and left: front, center and external).

3. CONTACT AND NON-CONTACT GUARDED AND COMPLIANT MOTIONS

In most cases, guarded and compliant motions have been related to force sensors [3], [8], [9], [10], [11], [13]. This paper extends them to non-contact guarded and compliant motions, based on infrared proximity sensorial information.

We present motions with force or proximity sensors, distinguishing two kinds of issues:

- *task specification*, achieved with the aid of language primitives
- *motion control*, carried out by a control system using external sensorial information

A great similarity may be established between contact and non-contact motions from the task specification and motion control points of view. In the sections that follow, both types of motions will be treated at the same time.

3.1. Task specification

In this respect, the APRIL programming system approach is partially based on Mason's theory [4]. According to it, in the task specification we must define:

- *compliance frame*. This task-depending frame may be defined fixed to the global or to the end effector reference systems. Examples of both are presented in [4], [11]. The compliance frame origin is most adequately located at the *compliance centre* [4]. Mobile compliance frame definition with respect to the global and the end effector reference system has not been considered here.
- *compliant* and *non compliant degrees of freedom* (d.o.f.). These are defined in the compliance frame. In the first, motion must comply to the environment. In the second, the nominal motion must be achieved.
- nominal *path* and/or *velocity*.
- force or distance *setpoint* vectors.
- motion *end conditions* [6]. Some of these are: to reach the target position or orientation in one or more non compliant d.o.f., to detect a given force or distance in one compliant d.o.f., etc.

These considerations will be discussed by means of an example in section 4.

3.2. Motion control

Several compliant motion control schemes have been proposed in the literature. In [12] the state of the art is presented. In [14], the implementation and the computational cost of some of these schemes are shown.

The APRIL control system for guarded and compliant motions is based on the idea of ***generalized damping*** [3], [12], extending its application to non-contact motions [1]. Control is achieved by correcting the nominal velocities, programmed on the robot MK-II controller, as a function of the information obtained from force or proximity sensors.

Mathematically, damping control can be expressed in the following way:

- for contact compliance:

$$\Delta\dot{X}_c = B^{-1}\cdot(F_d-F_c) = K\cdot(F_d-F_c) \tag{1}$$

- for non-contact compliance:

$$\Delta\dot{X}_c = C\cdot(D_d-D_c) \tag{2}$$

- for both:

$$\dot{X} = \dot{X}_d + \Delta\dot{X}_c \tag{3}$$

where each term is defined as follows:

B: damping matrix

$K=B^{-1}$: gain matrix

C: correction matrix (explained below)

$F_d=(f_x^d, f_y^d, f_z^d, m_x^d, m_y^d, m_z^d)^T$: force setpoint

$F_c = (f_x, f_y, f_z, m_x, m_y, m_z)^T$: measured force in the compliance frame

$D_d= (d_{RF}^d, d_{LF}^d, d_{RC}^d, d_{LC}^d, d_{RE}^d, d_{LE}^d)^T$: distance setpoint

$D_c= (d_{RF}, d_{LF}, d_{RC}, d_{LC}, d_{RE}, d_{LE})^T$: measured distance in the compliance frame

$\dot{X}_d = (\dot{x}^d,\dot{y}^d,\dot{z}^d,\dot{\theta}_x^d,\dot{\theta}_y^d,\dot{\theta}_z^d)^T$: velocity setpoint

$\Delta\dot{X}_c= (\Delta\dot{x},\Delta\dot{y},\Delta\dot{z},\Delta\dot{\theta}_x,\Delta\dot{\theta}_y,\Delta\dot{\theta}_z)^T$: correction velocity in the compliance frame

$\dot{X}= (\dot{x},\dot{y},\dot{z},\dot{\theta}_x,\dot{\theta}_y,\dot{\theta}_z)^T$: corrected velocity

Figure 3 shows the damping control schemes for contact and non-contact compliance. Usually, the sensing frame does not match the task frame. Likewise, motion correction can be given in another frame. Therefore, a force and velocity transformation between frames must be done [7]. Taking this into account, equations (1), (2) and (3) can be rewritten as:

$$\Delta\dot{X}_r = {}^rQ_c\cdot K\cdot(F_d-{}^cM_s\cdot F_s)$$

$$\Delta\dot{X}_r = {}^rQ_c\cdot C\cdot(D_d-D_c)$$

$$\dot{X}_r = \dot{X}_{dr} + \Delta\dot{X}_r$$

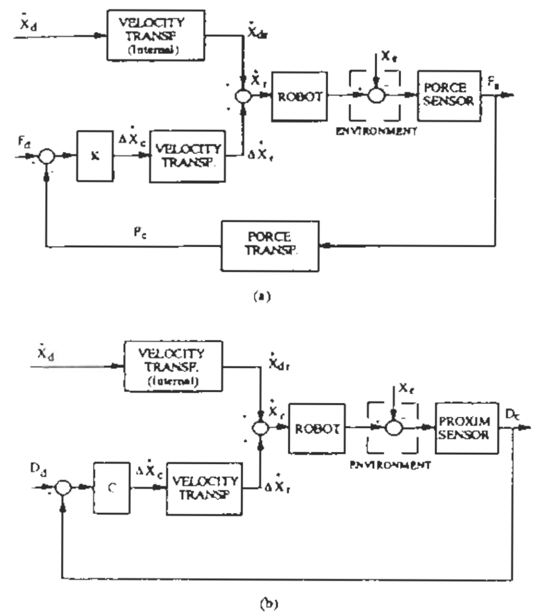


Figure 3. Damping control: (a) Contact compliance; (b) Non-contact compliance.

where:

${}^cM_s$: force transformation between sensing and compliance frames

${}^rQ_c$: velocity transformation between compliance and robot frames

$\dot{X}_{dr}$: transformed setpoint velocity to the robot frame

It must be pointed out that using a control scheme based on the idea of *generalized stiffness* [3], can cause malfunctions. For example, in contour following tasks with contact, deviations due to obstacles (ΔX_c) imply the appearance and maintenance of forces greater than the force setpoint ($F_c = F_d - S \cdot \Delta X_c$, S being the stiffness matrix). On the other hand, a loss of contact can become permanent if a null force setpoint is defined, or it can cause overshooting if the force setpoint is non-null. Similar behaviour can be observed in non-contact tasks.

About K and C matrices

Although K and C are control parameters, they are closely related to the specification task. If the compliance frame is properly associated to the task, matrix $K=B^{-1}$ can be chosen diagonal:

$$K=diag(k_{11},k_{22},k_{33},k_{44},k_{55},k_{66})$$

Its elements show the motion damping on the reference system axis. Large values of k_{ii} decrease damping, while small values increase it (nominal velocity will be kept). The system stability depends on the K coefficients. When working with rigid objets, large values of k_{ii} can instabilize the system [12].

The selection of the elements of matrix C 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.

It should be pointed out that, unlike the K matrix, the C matrix is independent of the object rigidity. Therefore, the choice of C matrix is usually easier.

Remarks on guarded and compliant motions

Sometimes, a compliant motion is preceded by a guarded motion. For example, the insertion of a peg in a hole is started by a guarded motion from the free space to a contact position. Then, the compliant motion is executed. However, with the damping control scheme presented, no explicit consideration of the guarded motion is necessary. The system itself controls the motion from the beginning, if the velocity, force/distance setpoints and the K/C matrices are properly chosen.

In a contact compliant motion, the approach movement (considered as a guarded motion) is defined by the velocity and force setpoints. That is:

$$\dot{X} = \dot{X}_d + K \cdot F_d$$

By selecting an adequate F_d, we can control the direction of the approach motion to the contact position. This direction may be different from the motion direction setpoint, represented by $\dot{X}_d$.

Likewise, for non-contact compliance, the direction of the approach motion can be controlled by the C matrix and the distance error:

$$\dot{X} = \dot{X}_d + C \cdot (D_d - D_c)$$

To clarify these ideas, some examples will be given in the next section.

4. EXAMPLE OF COMPLIANT MOTION TASKS

Compliant motions with contact appear in machining, deburring, polishing, grinding and assembly tasks. In the same way, compliant motions without contact are useful in inspection tasks, obstacle avoidance, as well as to center the end effector before grasping.

This section presents a contour following example. It is analyzed as a contact task, applying a constant force on the z axis, and as a non-contact task (Fig. 4). We show the similitude between both, in task specification and control.

The specifications required are:

- *Compliance frame* definition. It is intimately related to the task. When using force sensors, it is located on the tip of the peg. If proximity sensors are used, it is defined on the end of the gripper. In both cases, it corresponds with the compliance centre and it is fixed with respect to the end effector.

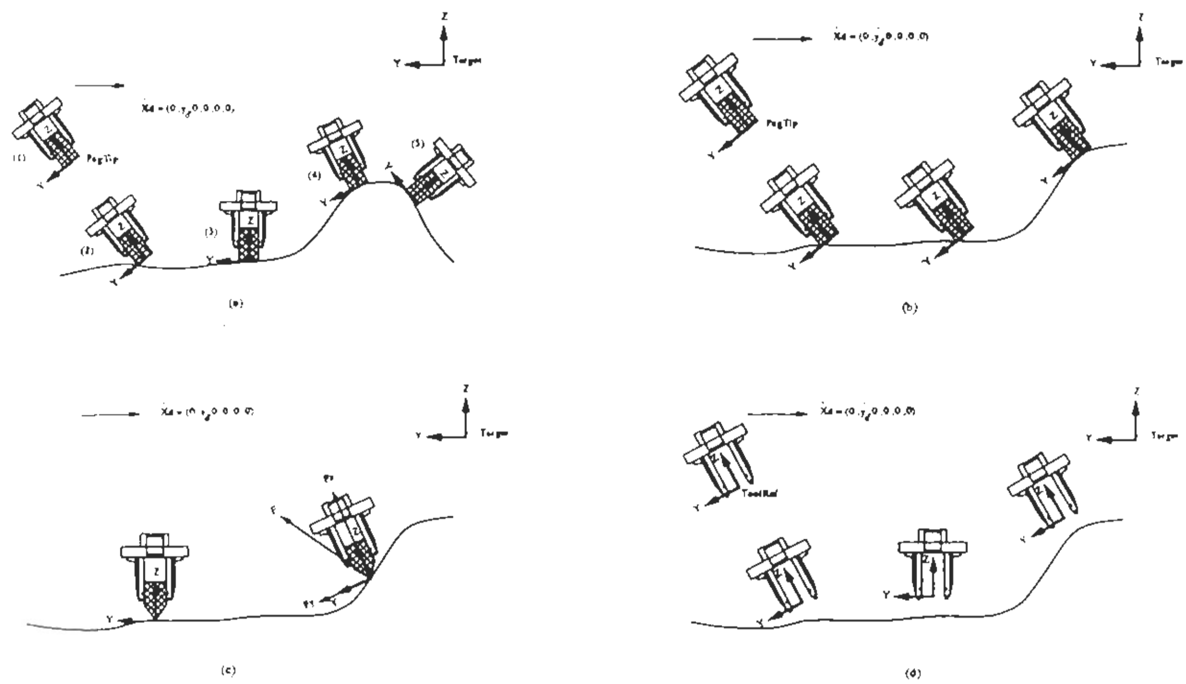


Figure 4. Contour following example: (a) z-θ_x contact compliance; (b) z contact compliance; (c) contact compliance with a sharp peg; (d) non-contact compliance.

- Definition of *compliant and non compliant d.o.f.*. Considering a path on the y-z plane, movement must comply on the z axis and around the x axis (Fig. 4a), while nominal movement must be maintained on the rest. The gain matrix has the form:

$$K = \mathrm{diag}\left(k_{11}^{s}\ k_{22}^{s}\ k_{33}^{l}\ k_{44}^{l}\ k_{55}^{s}\ k_{66}^{s}\right)$$

where k_{ii}^{s} represents a small value and k_{ii}^{l} a large one.

The appearance of torques around the x axis cause the reorientation of the peg (due to k_{44}^{l}) in a direction normal to the surface. If a small k_{44} is selected, the peg will remain in contact, but the orientation will not be modified according to the normal of the surface (Fig. 4b).

Having defined the compliance frame on the tip of a sharp peg, if we wish to follow a surface with the peg normal to it, the matrix cannot be diagonal. The reason is that torques capable of correcting the orientation do not appear. For example, in Fig. 4c, k_{42} must be less than zero, so that $\Delta\theta_x = k_{42} \cdot f_y$.

In a similar way, in a compliant movement without contact (Fig. 4d), the compliance controlled axis are z and θ_x. The matrix C elements selection is intimately related to the proximity sensors layout on the end effector. Since the frontal proximity sensors (RF, LF) are used for this task, the C matrix can be written as follows:

$$C = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ c_{31} & c_{32} & 0 & 0 & 0 & 0 \\ c_{41} & c_{42} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$c_{31} = c_{32}$

$c_{41} = -c_{42}$ c_{ij} being positive values.

The value of c_{ij} must be chosen according to the desired correction velocity on each axis (z, θ_x) as a function of the distance errors.

- *Velocity setpoint* selection. In both cases if we desire the motion to be made in the y-z plane, the velocity setpoint vector should be (Fig. 4):

$$\dot{X}_d = (0, \dot{y}^d, 0, 0, 0, 0)$$

- *Force/distance setpoint* selection. In the contact compliance motion case, we have to select the forces and torques to be applied to the objects. In our example these are:

$$F_d = \left(0, 0, f_z^d, 0, 0, 0\right)$$

The null forces could be left undefined, but if we wish the compliant motion to begin from free space, as explained in section 3, it is mandatory to make them equal to zero. Otherwise, velocity in the non-approach direction would be affected by the force setpoint. In our example, the approach velocity will be $\dot{z} = \Delta\dot{z} = k_{33} \cdot f_z^d$

Likewise, the control scheme with force setpoint avoids loss of contact during the change of surface orientation (Fig. 4a, step 5).

In a non-contact motion, the distance vector elements are not related to the selected compliance frame, but to the proximity sensor layout on the end effector. The setpoint vector will be:

$$D_d = (d, d, Und, Und, Und, Und)^T$$

where:

d desired distance of the Left Front (LF) and Right Front (RF) proximity sensors to the surface during motion.

Und shows that the related sensor value is undefined.

- *End condition* definition. In this task, to reach the target situation in the y axis (non compliant), can be selected as the end condition.

The APRIL specification for both tasks is shown in Table 1.

```
...
ForceRef (PegTip);
Speed (10.0);
Fd:- ForceDef (0,0,fzd,0,0,0)
CompMove(ForceSensor,PegTip,Target,Fd,Reached(Y));
.
                        (a)

...
ProxDef (ToolRef);
Speed (10.0);
Dd:=DistanceDef (d,d,Und,Und,Und,Und);
CompMove (ProxSensor,ToolRef,Target,Dd,Reached(Y));
..
                        (b)
```

Table 1. APRIL contour following especification: (a) Contact, (b) Non-contact.

Sometimes contact and non-contact motions can appear in the same task. For example, in the insertion of a peg in a hole, we can initially use non-contact compliant motion to position the peg close to the hole, and afterwards, a contact compliant motion is executed.

5. SYSTEM ARCHITECTURE

Concerning its robotic architecture, the system is constituted by a PUMA 560 robot, and force/torque, proximity and vision sensors.

The APRIL computing architecture is shown in Fig. 5. It is composed of a VAX 11/750, an MK-II robot controller and a real time multiprocessor system that we have named OSIRIS. The VAX is the main control unit, where the programming system resides. OSIRIS acts as a slave system of the VAX, controlling sensor operation and taking charge of real time sensorial information processing. It is constituted by a VME bus, in which several 68000 microprocessor-based cards have been installed. Each sensor has an associated processor, rendering possible the simultaneous use of them. Programs in OSIRIS have been codified in C.

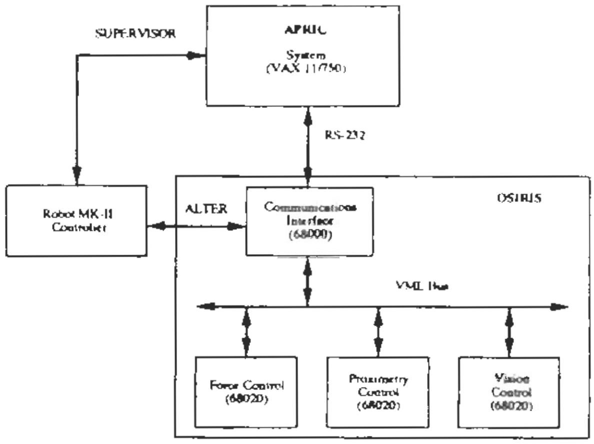


Figure 5. System architecture.

The VAX is communicated with the MK-II controller by a serial line (SUPERVISOR) through which it sends movement orders and receives information about the state of the robot. Communication with OSIRIS is achieved using another serial line, that allows it to order perception operations and to receive sensorial information.

A third serial line (ALTER) communicates OSIRIS directly with the robot controller. This line makes compliant motions possible, allowing trajectory modification to be accomplished in real time. Through this line it is also possible to stop the robot movement when the end condition is verified, or extreme situations are detected (loss of surface contact, overforces, etc).

6.- CONCLUSIONS

Guarded and compliant motions for contact and non-contact tasks have been presented. Both kinds of tasks are stated as similar problems.

To control these motions, we have implemented schemes based on the idea of generalized damping, including force/distance setpoints. Control system parameters are closely related to task specification. As the control parameters in non-contact motions are independent on the object rigidity, its selection is easier than in contact motions.

It must be pointed out that task specification is not considered here from the fine motion high level strategies generation point of view. Our approach is based on control of compliant motions, using a closed loop scheme with force or distance feedback. In many complex tasks it is necessary to adopt a more elaborated strategy, to carry out a correct execution. This objective will be undertaken in future works. Also, in the APRIL system, we will consider object detection and recognition strategies, based on the studied motions.

ACKNOWLEDGEMENTS

This work has been partially supported by CICYT, Grant nº PA-86-0028.

REFERENCES

1. Balek D.J., Kelley R.B.: "Using Gripper Mounted Infrared Proximity Sensors for Robot Feedback Control". *IEEE International Conference on Robotics and Automation*, 1985, pp 282-287.

2. Lladó J. et al: "Análisis experimental de un transductor de esfuerzos". *Revista de Robótica*, nº 33, Sept-Oct 1988.

3. Mason M.T. Compliance and Force Control for Computer Controlled Manipulators. *Robot Motion: Planning and Control*, (Brady M. et al. editors), MIT Press, 1982, pp. 374-404.

4. Mason M.T.: "Compliant Motion". *Robot Motion: Planning and Control*, (Brady et al. editors), MIT Press, 1982, pp. 305-322,.

5. Montano L.: "APRIL: Un sistema evolucionado de programación de robots. *Tesis Doctoral*, Universidad de Zaragoza, 1987.

6. Montano L., Tardós J.D., Sagüés C., Neira J.: "Entorno de programación y control para un sistema robótico multisensorial". *1er Congreso de A.E.R.*, Zaragoza, Nov. 1989.

7. Paul R.P.: "Robot Manipulators: Mathematics, Programming and Control". *The MIT Press*, Cambridge, Mass, 1981.

8. Paul R.P., Zhang H.: "Hybrid Control of Robot Manipulators". *TR-EE 84-27*, School of Electrical Engineering, Purdue University, 1984.

9. Raibert M.H., Craig J.J.: "Hybrid Position Force Control of Manipulators". *Journal of Dynamic Systems Measurement and Control*, Vol. 102, 1981, pp. 126-133.

10. Salisbury J.K.: "Active Stiffness Control of a Manipulator in Cartesian Coordinates". *19th IEEE Conference on Decision and Control*, Alburquerque, New Mexico. 1980, pp. 95-100.

11. De Schutter J., Van Brussel H.: "Compliant Robot Motion I. A Formalism for Specifying Compliant Motion Tasks". *The Int. Journal of Rob. Research*, Vol. 7, nº 4, 1988, pp. 3-17.

12. Whitney D.E.: "Historical Perspective and State of the Art in Robot Force Control". *The Int. Journal of Rob. Research*, Vol. 6, nº 1, 1987, pp. 3-14.

13. Xu Y., Paul R.P.: "On Position Compensation and Force Control Stability of a Robot with a Compliant wrist". *IEEE International Conference on Robotics and Automation*, April 1988, pp. 1173-1184.

14. Zhang Y., Paul R.P.: "Robot Manipulator Control and Computational Cost". *MS-CIS-88-10*, GRASP.LAB, 1988.