Copyright © IFAC Intelligent Components and Instruments
for Control Applications, Buenos Aires, Argentina, 2000

# A CONTROL SYSTEM MODIFICATION IN THE SCORBOT-ER VII ROBOT CONTROLLER

**J. Pérez, S. Backis and M. Anigstein***

*Laboratorio de Robótica. Depto. de Ing. Mecánica y Naval, y de Electrónica.*
*Facultad de Ingeniería – UBA. manigst@fi.uba.ar*

Abstract: The constraints in the orientation of the tool owing to the lack of the 3rd axis of the Scorbot-er VII's wrist and the strategy with which the robot controls the tool's orientation during the cartesian movements, are both analyzed. Non desirable deflections produced during simple robot's tool planar movements are shown, and a modification in the control system design to solve this problem is proposed and implemented. Robot's controller limitations that impose conditions on the performance of the developed control system modification are discussed. The dynamics of the 5th axis is identified and simulation is used to study the new control. *Copyright © 2000 IFAC*

Keywords: Scorbot-er VII robot, non desirable movement, control system, cartesian trajectories, dynamics identification.

## 1. INTRODUCTION

In the Laboratorio de Robótica de la Facultad de Ingeniería de la Universidad de Buenos Aires we are using the Scorbot-er VII Robot for teaching and research activities (Backis *et al.*, 1998). It is a five axis, semi-industrial, training robot. It is being used in a Project about cooperative tasks (Zheng and Luh, 1987; Ramos *et al.*, 1999) with a Gantry type robot for planar movements, built at the Laboratorio de Robótica (Ramos *et al.*, 1997, 1998). In the strategy that will be employed for the coordination it is necessary to perform cartesian movements (Paul, 1981). But the Scorbot-er VII has particular characteristics owing to the lack of an axis in the wrist: the tool's orientation is not being adequately controlled during the linear movements (Backis *et al.*, 1999). For this reason, a modification of the Scorbot-er VII robot's control system was accomplished that solves the problem in the horizontal planar movements which are the only ones that will be executed in the cooperative tasks with the Gantry robot.

The present article is organized as follows. In section 2 the Scorbot-er VII's kinematics is summarized. In section 3 two simple planar movements are performed. In these movements the undesirable behavior of the tool's orientation can be seen. In section 4 we study the causes of the problem. In section 5 a solution is proposed and implemented. In section 6 the dynamics of the 5th axis is

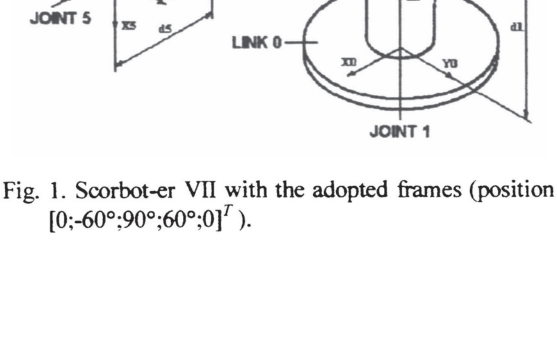


Fig. 1. Scorbot-er VII with the adopted frames (position ≈ $[0;-60°;90°;60°;0]^T$ ).

* This work was supported by the Programación Científica 1998-2000 UBA-SECyT with grant AI02.

identified and simulation is used to examine the new control system.

## 2. SCORBOT-ER VII

### 2.1. Kinematics

Table 1. Kinematics Parameters.

| axis (i) | $a_i$ [mm] | $d_i$ [mm] | $\alpha_i$ [°] | $\theta_i$ [°] |
|---|---|---|---|---|
| 1 | 50 | 358.5 | -90 | Variable |
| 2 | 300 | -35.3 | 0 | Variable |
| 3 | 250 | 0 | 0 | Variable |
| 4 | 0 | 0 | 90 | Variable |
| 5 | 0 | 200 (x) | 0 | Variable |

(x) it depends on the tool used.

Table 1 and Fig. 1 show the kinematics structure of the Scorbot-er VII robot, as well as the links, axes and frames assigned according to the Denavit and Hartenberg method (Paul,1981). With the data of Table 1 it is easy to obtain the matrix transformations relating consecutive frames. As an example, $A_0^5$ relates frame 5 (tool's frame) with frame 0, and its rotation sub-matrix results in:

$$R_0^5 = \begin{bmatrix} c_1 & 0 & -s_1 \\ s_1 & 0 & c_1 \\ 0 & -1 & 0 \end{bmatrix} \begin{bmatrix} c_{234}c_5 & -c_{234}s_5 & s_{234} \\ s_{234}c_5 & -s_{234}s_5 & -c_{234} \\ s_5 & c_5 & 0 \end{bmatrix} \quad (1)$$

The Scorbot-er VII's space is of dimension 5. The cartesian coordinates that it uses are $X = [X\ Y\ Z\ P\ R]^T$ , where $r = [X\ Y\ Z]^T$ is the position vector of the origin of frame 5 projected in frame 0 and

$P$ = *Pitch* = $90^0 - (\theta_2 + \theta_3 + \theta_4)$ = angle between $\bar{z}_4 \equiv \bar{z}_5$ and $\bar{x}_1$ (2)

$R$ = *Roll* = $-\theta_5 + \mathit{Offset}$ = tool's rotation around $\bar{z}_5$ (3)

### 2.2. Linear cartesian movements

In order to perform cartesian movements the basic command is MOVEL, with which the point $O_5$ of the tool describes a straight line between the departure and arrival positions. The interpolation is accomplished over the cartesian coordinates $X = [X\ Y\ Z\ P\ R]^T$.

## 3. EXAMPLES OF PLANAR MOVEMENTS

When the robot is supported with its base as in Fig. 1, in the horizontal planar movements, the rotation of the objects that are being manipulated is performed around an axis parallel with respect to the 1rst. robot's axis. In the following examples, the robot will be always moved preserving its 5th axis parallel with respect to the 1rst. one, and with $\bar{z}_5$ pointing down. Our purpose is to perform cartesian movements (i.e. movements defined in the task space), independent of the robot kinematics. Starting from an initial position (*pos_a* = [3000 -1000 Z -90 $R_1$]) a linear movement until a final one (*pos_b*) will be performed, trying to change only the tool's position (not the orientation). The tool will have a displacement of 500 mm in the Y positive direction. The instructions sequence in the controller language (ACL) may be

SHIFTC *pos_a* BY *y 5000*
MOVEL *pos_a*

In Fig. 2 (a) we show the movement projected on the [X Y] plane. The change of the spatial tool's orientation can be seen. The objective has not been obtained because our purpose was to have no change in the tool's orientation. In order to solve the problem we will teach the final position to the robot, using the TEACH command.

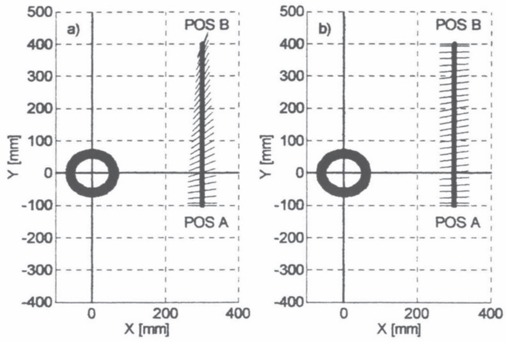


Fig. 2. (a) Linear movement changing only the Y coordinate. (b) Linear movement teaching the final position.

The final position will be *pos_b* = [3000 4000 Z -90 $R_2$] and the movement instruction in this case is simply

MOVEL *pos_b*

In Fig. 2. (b) we show the result. Although in the final position the tool's orientation is correct, during the movement the orientation is not preserved. Then, a curious situation happens: a 5 axis anthropomorphous robot cannot correctly perform a simple movement that can easily be performed with a 4 axis (SCARA or Gantry) robot. Suppose, as an example, that in *pos_a* the tool supports a pin at the correct position and orientation to be inserted in a hole, and *pos_b* would be the final position for the pin being inserted. To accomplish the task the Scorbot-er VII robot simply has to move from pos_a to pos_b in a straight line, and without changing the orientation. As we saw it cannot do it.

## 4. ABOUT THE REASON FOR THE PROBLEM

### 4.1. General cartesian movement

We will analyze the velocity Jacobian of the robot. We compute the kinematics characteristics of the tool with respect to the origin of frame 5 ($O_5$). $\bar{V}_{O_5}$ = velocity of point $O_5$ of link 5 (tool) and $\bar{\omega}_5$ = angular velocity of link 5. To obtain a simple symbolic expression we will compute both vectors projected in frame 4. The vectors $\bar{V}_{O_5}\big|_4$ and $\bar{\omega}_5\big|_4$ are easily obtained (Craig, 1986) and result in

$$\left[ \bar{V}_{O_5}\big|_{\bar{x}_4} \quad \bar{V}_{O_5}\big|_{\bar{y}_4} \quad \bar{V}_{O_5}\big|_{\bar{z}_4} \quad \bar{\omega}_5\big|_{\bar{y}_4} \right]^T =$$

$$\begin{bmatrix} -d_2c_{234} & d_5 + a_3s_4 + a_2s_{34} & d_5 + a_3s_4 & d_5 \\ d_5s_{234} + a_3c_{23} + a_2c_2 + a_1 & 0 & 0 & 0 \\ -d_2s_{234} & -(a_3c_4 + a_2c_{34}) & -a_3c_4 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \\ \dot{\theta}_3 \\ \dot{\theta}_4 \end{bmatrix} \quad (4)$$

$$\bar{\omega}_5\big|_{\bar{x}_4} = -s_{234}\,\dot{\theta}_1 \quad (5)$$

$$\overline{\omega}_5\big|_{\overline{z}_4} = c_{234}\,\overset{\circ}{\theta}_1 + \overset{\circ}{\theta}_5 \quad (6)$$

On the other hand, with the cartesian coordinates of the Scorbot-er VII, the velocity of the tool would result $\overset{\circ}{X} = \begin{bmatrix} \overset{\circ}{X} & \overset{\circ}{Y} & \overset{\circ}{Z} & \overset{\circ}{P} & \overset{\circ}{R} \end{bmatrix}^T$. The relation between both expressions is as follows. The first three components of $\overset{\circ}{X}$ are the projections of $\overline{V}_{O_5}$ in frame 0

$$\overline{V}_{O_5}\big|_0 = \begin{bmatrix} \overset{\circ}{X} & \overset{\circ}{Y} & \overset{\circ}{Z} \end{bmatrix}^T \quad (7)$$

and therefore

$$\begin{bmatrix} \overset{\circ}{X} & \overset{\circ}{Y} & \overset{\circ}{Z} \end{bmatrix}^T = R_0^4\,\overline{V}_{O_5}\big|_4 \quad (8)$$

where $R_0^4$ is the rotation matrix between frames 4 and 0 which are computed using the kinematics parameters of table 1. The fourth component of $\overset{\circ}{X}$ is obtained deriving Eq. 2 and matching it with the fourth component of Eq. 4

$$\overset{\circ}{P} = -\overline{\omega}_5\big|_{\overline{y}_4} \quad (9)$$

Finally according to Eq. 3

$$\overset{\circ}{R} = -\overset{\circ}{\theta}_5 \quad (10)$$

It must be observed that Eq. 4 allows to compute the velocities of the first four axes of the robot for an arbitrary vector $\overset{\circ}{X} = \begin{bmatrix} \overset{\circ}{X} & \overset{\circ}{Y} & \overset{\circ}{Z} & \overset{\circ}{P} & \overset{\circ}{R} \end{bmatrix}^T$. But the $\overset{\circ}{X}$ vector does not fully define the tool's velocity. The $\overline{\omega}_5\big|_{\overline{x}_4}$ component cannot be arbitrarily selected: according to Eq. 5, it is defined by the robot position (through $s_{234}$) and by $\overset{\circ}{\theta}_1$ which results from Eq. 4. This component causes a change in the tool's orientation that cannot be corrected by the controller. This behavior of the Scorbot-er VII is due to the lack of the 3rd axis of the wrist. For this motive the robot cannot accomplish, in general, tasks in which it must preserve the tool's orientation during the motion. Finally Eq. 6 shows that it could always be fitted using $\overset{\circ}{\theta}_5$, an arbitrary desired value for $\overline{\omega}_5\big|_{\overline{z}_4}$, for any $\overset{\circ}{\theta}_1$ which results from Eq. 4.

### *4.2. Linear planar movements revisited*

Afterwards we analyze the examples of the preceding chapter. As we take $P$ = -90, from Eq. 2 it results $(\theta_2 + \theta_3 + \theta_4) = 180^0$ and then $s_{234} = 0$. Therefore, from Eq. 5 vanishes the component of the angular velocity that could not be compensated by the controller ($\overline{\omega}_5\big|_{\overline{x}_4} = 0$). Moreover as $c_{234} = -1$, from Eq. 6 we see that it suffices to set $\overset{\circ}{\theta}_1 = \overset{\circ}{\theta}_5$ in order to preserve the orientation of the tool. Then, the robot is in a particular configuration in which the tool's orientation could be preserved. Nevertheless, the Scorbot-er VII's controller does not do it.

It is easy to understand what is the reason for this behavior. During the linear movements the controller interpolates in the cartesian coordinates $[X \;\; Y \;\; Z \;\; P \;\; R]$ as we did in (2.2) and Eq. 3 shows that $\theta_5$ only depends on $R$. On the other hand $\theta_1$ results from the resolution of the inverse kinematics position (Backis *et al.*, 1998) at each point of the lineal trajectory. Obviously, in general it will be $\overset{\circ}{\theta}_1 \neq \overset{\circ}{\theta}_5$ and therefore the tool's orientation is not preserved. Then in the first example, to move the tool along the linear trajectory, the controller changes the value of $\theta_1$ but it maintains the value of $\theta_5$ because $R$ must be constant. In the second example, $\theta_1$ changes again as in the first one because it only depends on the linear tool's trajectory. On the contrary, $\theta_5$ in this case is not preserved. It changes, but following a polynomial of interpolation that is independent of the linear trajectory and therefore independent of $\theta_1$.

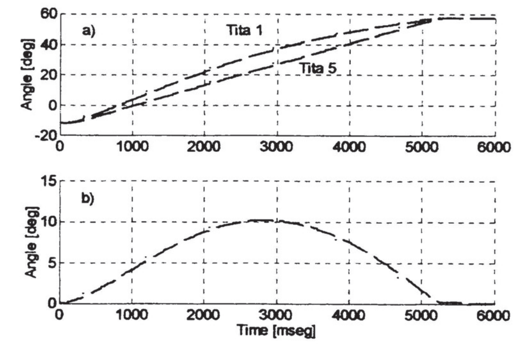


Fig. 3. (a) Evolution of $\theta_1$ and $\theta_5$ in the second example. (b) Difference between $\theta_1$ and $\theta_5$. Error in the tool's orientation.

In Figs. 3 (a) and (b), the evolution of $\theta_1$ and $\theta_5$ and the error in the tool's orientation are shown.

## 5. A SOLUTION TO THE ORIENTATION'S PROBLEM

### *5.1. New coordinate to interpolate*

The 5th cartesian coordinate of the Scorbot-er VII (*Roll*) actually is a joint coordinate (Eq. 3). In the planar movements this coordinate does not define the tool's orientation. The orientation could be obtained from Eqs. 1 and 2. As $P$ = –90 it results in

$$R_0^{\,5} = \begin{bmatrix} -c\theta^P & -s\theta^P & 0 \\ -s\theta^P & c\theta^P & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (11)$$

where

$$\theta^P = \theta_1 - \theta_5 \quad (12)$$

is the angle between the unit vector $\overline{y}_5$ of the tool with the unit vector $\overline{y}_0$ of the robot's base. Nevertheless, when the robot accomplishes cartesian movements, the controller interpolates in $R$ but not in $\theta^P$. To correctly control the orientation it is necessary to interpolate in $\theta^P$, as we will explain immediately.

### *5.2. Control modification*

*Independent control system of the 5th axis.* To resolve the above problem we had to modify the Scorbot-er VII's 5th axis control system. The controller has an option to run user's programs for control of the robot's axes. It allows to read the encoders (ENC [*n*]) and to drive each axis (ANOUT [*n*]). To implement this option the ACL language suggests to use the COFF command which disables the robot's axes control and allows to drive each axis with the ANOUT [*n*] command. But COFF disables all the axes at the same time. As we need the first four axes to continue being robot controlled, this alternative could not be used.

Then, we choose to perform the independent control of the $5^{th}$ axis using one of the cards for auxiliaries axes (we used the $7^{th}$ servo control axis card). The changes that have been performed are the following:

1. The connection of motor 5 was modified in order to drive it, alternatively, from card 5 (robot control) or from card 7 (user control).
2. The cables corresponding to the motor are then exchanged from card 5 to 7 and conversely, through two of the relay outputs that the controller has. In this way it can alternatively perform tasks in which the movement of the $5^{th}$ axis is controlled by the robot, with others in which this axis is controlled by the user's program.
3. The movement that results for the $5^{th}$ axis with the new control, will have in general an important difference with respect to the movement that the robot's controller will try to perform for this axis at the same time. In order that the controller's impact protection soft does not disable the robot movement, the exchange between cards must be accompanied with the vanishing/repositioning of the $5^{th}$ axis servo control constants.

*$5^{th}$ Axis control with an user program.* The user program is responsible for:

- Obtaining the initial and final values of $\theta^P$ coordinate, which actually defines the tool's orientation. The ACL language supplies the axes coordinate of known positions (PVAL *pos axis*). For example, if *pos_a* and *pos_b* are the initial and final positions of a movement, the following sequence of instructions computes the $\theta a^P$ value:

  SET *th_1* = PVAL *pos_a 1*
  SET *th_5* = PVAL *pos_a5* (13)
  SET *th_p* = *th_1* – *th_5*

  The final value $\theta b^P$ is computed likewise.

- The time interpolation of the desired value for the tool's orientation ($\theta^P$ coordinate). To accomplish the interpolation, the controller has the border conditions:

  $\theta a^P$, $\theta b^P$, $\overset{\circ}{\theta a^P} = \overset{\circ}{\theta b^P} = 0$ (in the cartesian movement the Scorbot-er VII always reaches the final position with null velocity). The time ($T$) needed to perform the movement can be obtained from the controller itself (parameter LTA) if it was not included in the movement instruction. In our application to perform cooperative tasks, the Scorbot-er VII receives $T$ through the RS-232 interface from the other robot. We use parabolic interpolation for the joint velocities (Spong and Vidyasagar, 1989 ):

$$\theta^P(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \; ; \; 0 \le t \le T$$
$$a_0 = \theta a^P \; ; \; a_1 = \overset{\circ}{\theta a^P} = 0 \qquad (14)$$
$$a_2 = \frac{3(\theta b^P - \theta a^P)}{T^2} ; a_3 = \frac{2(\theta a^P - \theta b^P)}{T^3}$$

- To control the $5^{th}$ axis movement in order to get the tool's orientation to be as it is desired. The desired value for $\theta_5$ in each sample time results from Eq. 12

$$\theta_5^d(t) = \theta_1(t) - \theta^P(t) \qquad (15)$$

  where $\theta_1(t)$ is read from the encoder of axis 1 (ENC [*1*]) and $\theta^P(t)$ is computed with Eq. 14.

*Control strategy. Servo constants and sample frequency.* The Scorbot-er VII uses an independent control for each axis, with strategy PID + an Offset (*F*) to cancel the Coulomb friction. The original $5^{th}$ axis factory servo constants are

$$\begin{aligned} K_P &= PAR\ 25 = 160 \\ K_D &= PAR\ 45 = 200 \\ K_I &= PAR\ 65 = 1000 \\ F &= PAR\ 85 = 0 \end{aligned} \qquad (16)$$

The sample frequency the controller states that it is used (User's Manual) is 100 Hz. It is also the maximum one that can be utilized from a user's control program (clock tick = 10 ms). To decide the new $5^{th}$ axis control strategy that will be used, we must take care of the following facts.

As in the cooperative tasks with the Gantry robot the $5^{th}$ axis is always at the vertical position, it is not affected by torques due to the tool and load's weight. Also, in the cooperative tasks application, the external disturbances that come from the other robot, neither must be rejected by the control system. Therefore, the integral control will not be used ($K_I$ = 0). Without integral control, the Coulomb friction will cause stationary error in the final position and during the trajectory tracking. To reduce this error, the Coulomb friction was identified and feedforward compensated. We identify different values for each turn direction $F_1$ = 608, $F_2$ = 763 (see section 6) and slightly smaller ones were utilized in the compensation.

The value that must be applied to the DAC of driver 7 (which controls motor 5) is then computed with a PD strategy + the feedforward compensation $F_1/F_2$ . Experimentally we fitted the following proportional and derivative servo constants

$$\begin{aligned} K_P &= 1 \\ K_D &= 1 \end{aligned} \qquad (17)$$

It must be observed that this values are very smaller than the originals (Eq. 16) set for the axis.

In Fig. 4 (a) the evolution of axis 1 and 5 for the second example of chapter 3 is shown, but with the new control. In Fig. 4 (b) the error in the tool's orientation is shown. Although the error in the tool's orientation has notably decreased, the control performance is not as might be expected.

In Fig. 4 (b) it is clear to see an important noise (which would grow with bigger servo constants).

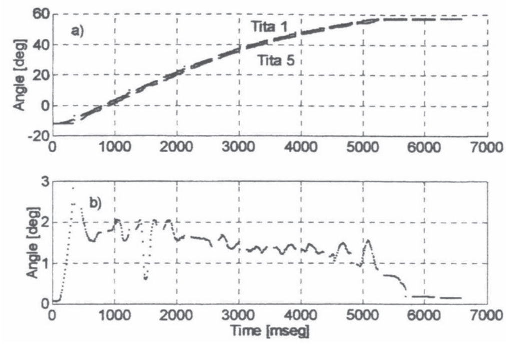


Fig. 4. (a) Evolution of $\theta_1$ and $\theta_5$ for the second example of chapter 3 with the new control. (b) Error in the orientation of the tool with the new control.

The calculation that the user's program performs in real time makes the sample frequency result slightly smaller than 100 Hz.. The average time between samples is 11 ms. but this small increase in the delay cannot justify by itself the smaller servo constants and the poor performance of the control.

To study more carefully what happens during the movement, we did the time record at which the samples were taken. We found periodicals interruptions of 100 ms. It happens that the user's program is neglected. The frequency with which the user's program is not executed increases with the controller's duty. The ACL language has the possibility of assigning priority levels to the user's programs, but even with the highest priority the periodical interruptions remain. This behavior of the ACL puts a limit on its multitasking quality and on the possibility to perform good controls with a user's program. Nevertheless, in our case, for the particular application in which the new control system is used (cooperative movements), the behavior is satisfactory. The non desired tool's deflections are corrected and the noise in the movement is deadened because the object manipulated with the Scorbot-er VII is supported by the other robot at the same time.

### *5.3. Offline programming*

The cartesian positions and movements are used in off-line programming (Regev, 1995), among other reasons, so that the robot is not needed during the programming stage. But with the Scorbot-er VII the off-line programming has the fault that the tool's orientation, as we saw above, cannot be directly fixed with the robot's cartesian coordinates. In the first example of chapter 3 the *R* coordinate does not change when the robot moves from *pos_a* to *pos_b* , although the tool's orientation changes. On the contrary, *R* changes in the second example, but the orientation does not.

For the user it would be convenient that this Scorbot-er VII's behavior was "not visible" during the programming. In order to be so, he must be able to define the positions with the $\theta^P$ coordinate. The implementation of this alternative is simple. It is not necessary to use Eq. 13. On the other hand, if it were necessary to go to the position (*pos_a* for example) with the normal robot interpolation, the value of the *R* (*Ra*) cartesian coordinate would have to be computed. Using Eqs. 3 and 12 results

$$Ra = \theta a_1 - \theta a^P + Offset \tag{18}$$

where the known value is now $\theta a^P$. Also, as $\theta a_1$ does not depend on *Ra* (both are joint coordinates), it is obtained using PVAL *pos_a 1* with any value of *Ra*.

For our application in cooperative movements, the Scorbot-er VII just receive the initial and final values of $\theta^P$, thorough the RS-232 interface from the other robot.

## 6. SIMULATION OF THE NEW CONTROL SYSTEM

As we said in subsection 5.2.3, although the new control system resolves the problem of the undesirable deflections, the control performance is not as might be expected. To analyze this subject we develop a simulator for the 5[th] axis dynamics.

### *6.1. Dynamics identification*

The 5[th] axis will always work at vertical orientation, then it will not be affected by torques due to the tool and load's weight. The axis actuator is a dc servomotor with a harmonic drive speed reducer. We neglect the armature inductance and therefore we use a second order system for the model, that takes into account the robot and the actuator dynamics:

$$U_5 = J\ddot{\theta}_5 + B\dot{\theta}_5 + \mathrm{sgn}(\dot{\theta}_5)F_{1/2} \tag{19}$$

where $U_5$ = motor voltage, $J$ = effective joint inertia, $B$ = effective viscous damping coefficient, $F_1$ = effective clockwise Coulomb friction, $F_2$ = effective counterclockwise Coulomb friction. The inertial parameters $J, B, F_1/F_2$ must be identified. The accelerations in the exciting trajectory must be as high as possible for accurate inertia estimates. The DAC value that can be applied to the motor is in the range $\pm 5000$ millivolts. The experiment was performed driving the 5[th] axis with a predefined voltage varying linearly in the range $\pm 3000$ millivolts (Fig. 5). The exciting trajectory which results with the collected data from the encoder of axis 5 (with periodical interruptions of 100 ms.) can be seen in Fig. 6. We interpolate inside the interruptions preserving the velocity of the trajectory in each extreme point. We also dropped the first 5 and last 5 samples of the trajectory to eliminate the effects of transients. Approximately 1000 data points have been used to estimate the velocities and accelerations (Fig. 7) by finite difference with the formulas $\dot{\theta}_5{}^i = \left(\theta_5{}^{i+1} - \theta_5{}^{i-1}\right)/(2\tau)$, and $\ddot{\theta}_5{}^i = \left(\dot{\theta}_5{}^{i+1} - \dot{\theta}_5{}^{i-1}\right)/(2\tau)$ where $\tau$ is the sampling period (10 ms).

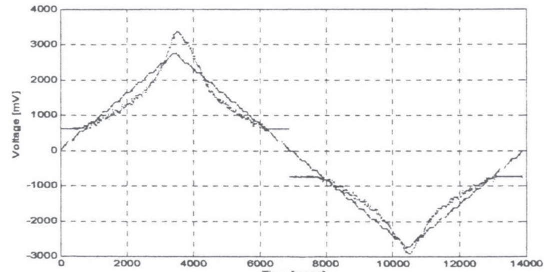


Fig. 5. Exciting and computed voltages.

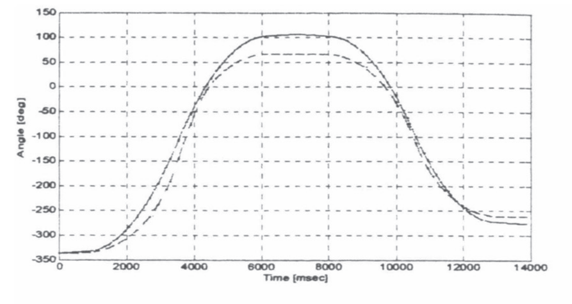


Fig. 6. Exciting (interrupted) and open loop simulated trajectories.

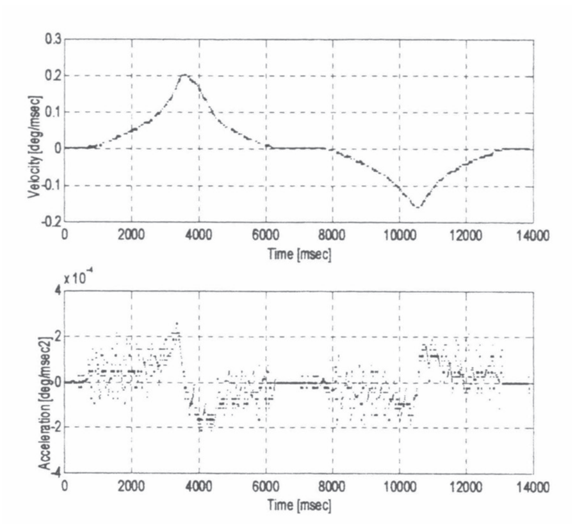


Fig. 7. Estimated velocity and acceleration.

With a least-square algorithm we obtain $J = 545520.51$, $B = 13371.15$, $F_1 = 607.91$, $F_2 = 762.93$.
Statistical characterization of the errors of the estimated parameters has not been attempted yet, at the present stage of the study. To make a first verification we compare in Fig. 5 the exciting voltage, with the computed one using the identified parameters over the generated trajectory. Also, to test our dynamic model, we simulate the open loop control of the 5$^{th}$ axis applying the same voltage that was used in the identification procedure. In Fig. 6 we superpose the simulated trajectory with the real one, and it can be verified that the model works well.

### *6.2. Simulation of the new control for the 5$^{th}$ axis*

Finally, we revisit the second example of chapter 3 with the simulator. Axis 5 is controlled with the same strategy and the same servo constants used in the new control for the robot. Fig. 8 compares $\theta_5^{\,d}(t)$, with the values of $\theta_5(t)$ read from the encoder of the robot, and from the simulation. The measured and simulated trajectories have very slight differences ratifying the preceding results. At the present time we are studying the performance of the new control systems, trying to verify the causes of the noise during the robot's movement.

## 7. CONCLUSION

It has been shown how in simple planar movements, non desired deflections are produced in the Scorbot-ER VII's tool. The causes of this behavior were analyzed. In order to solve the problem, a modification in the system control and in the programming language was studied and implemented (a video to illustrate the robot's behavior before and after the control system modification, will be enclosed). Controller limitations that impose quality performance conditions on the developed control system were discussed. To analyze this problem the dynamic parameters of axis 5 was identified, and dynamic simulation was used to examine the new control system.

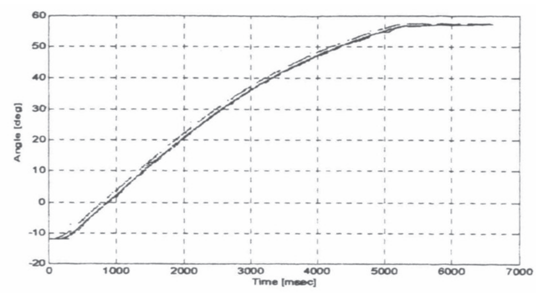


Fig. 8. Comparison of $\theta_5^{\,d}(t)$ and $\theta_5(t)$ from the actual and simulated robot trajectories.

## REFERENCES


Backis S., González P., Pérez J. and M. Anigstein. (1998). Identification of the SCORBOT-ER VII Robot's Trajectory Generator. *XVI Congreso Argentino de Control Automático. AADECA'98.* Vol. 2, pp. 484-489. Buenos Aires.

Backis S., González P., Pérez J. and M. Anigstein. (1999). About the Cartesian Motion of the five axes SCORBOT-ER VII Robot. *VIII RPIC.* Vol. 2, pp. 1-6/7. Mar del Plata, Argentina.

Craig J.C. (1986). *Introduction to Robotics Mechanics and Control.* Addison-Wesley, Reading, Mass.

Paul R. P. (1981). *Robot Manipulators: Mathematics, Programming, and Control.* MIT Press, Cambridge, Mass.

Ramos R., C. Costas, C.S. Kang, D.S. Son and M. Anigstein. (1997). Real Time Distributed Processing. Robotics Controller Application. *RPIC'97.* Vol. 1, pp. 224-229. San Juan, Argentina.

Ramos R., C. Costas, E. Panciera Molanes and M. Anigstein. (1998). Gantry type Robot for Research and Educational Applications. *XVI Congreso Argentino de Control Automático. AADECA'98.* Vol. 1, pp. 267-272. Buenos Aires.

Ramos R., E. Panciera Molanes, C. Costas and M. Anigstein. (1999) Communications and Cooperative Motion Functions between the Gantry-UBA Robot and the SCORBOT-ER VII. *VIII RPIC.* Vol. 2, pp. 7-14/7. Mar del Plata, Argentina.

Regev Y. (1995). The evolution of offline programming. *Industrial Robot.* **22**(3), 3.

Spong W. W. and M. Vidyasagar. (1989). *Robot Dynamics and Control*, J. Wiley & Sons, New York.

Zheng Y.F. and J.Y.S. Luh. (1987). Constrained Relations between Two Coordinated Industrial Robots for Motion Control. *The Int. J. of Robotics Research*, **6**(3), 60-70.