7th International Machine Design and Production Conference
September 11-13, 1996, METU, ANKARA, TURKEY

ON-LINE TRAJECTORY PLANNING AND CONTROL FOR ROBOTIC MANIPULATORS

Ali KIRECCI, Sadettin KAPUCU and Mehmet TOPALBEKIROGLU
University of Gaziantep, Mechanical Engineering Department, GAZIANTEP

ABSTRACT

This paper includes an experimental study of on-line trajectory planning and trajectory control for a hydraulic robot. In order to plan a trajectory the necessary precision points that describe the motion of the end-effector must be specified by the user. However, for real time trajectory planning and control, all the information about the manipulator and its environment have to be sensed by computer integrated sensory equipment. Sensory equipment such as encoder, tachometers, potentiometers and others can provide a limited information about the situation of a manipulator, but, vision provides an extensive information about its working environment.

In this study, the image of the arbitrarily placed red spots, which represent the precision points on the path of the manipulator in the Cartesian co-ordinate system are captured by a video camera. The information obtained from image processing and the original position of the end-effector are the vital information for trajectory planning. Trajectories are generated in real time by cubic spline functions and self-tuning adaptive control, which is one of the advanced control method, is applied to the robot for tracking of example trajectories.

1 INTRODUCTION

The activity of generating motion curves while satisfying the precision points by defining time sequences of configurations of the end-effector of a manipulator, feasibility checks and adaptation of the motion before implementation is referred to as motion design or trajectory planning. As the trajectory is executed, the tip of the end-effector traces a curve and changes its orientation.

There are two different approaches for trajectory planning: off-line and on-line planning. Off-line planning is carried out outside the manipulator environment and makes use of extensive software usually including computer graphics. The developments of trajectories take place without access to the manipulator itself and without recourse to real time operation [1]. On-line trajectory planning refers to the determination of the history of a motion by means of on-board sensory equipment and then the generation and execution of the motion in real time. That means, for a specific task, the constraints for the trajectory including constraints for obstacles along the path and constraints for co-ordination with other machines will be determined be means of sensory equipment, then the optimum trajectory can be generated and executed in a real time. Such planning requires highly sensitive sensory equipment, high-speed computers and artificial intelligence capabilities.

A video camera is one of the vital sensory equipment which will be used for on line robot control. It produces images of the workspace of a manipulator. Image processing provides a robotic system with extensive information about its working environment. Location of each pixel on the screen and color definitions describe not only the external appearance of the object in view, but also its location and angular orientation as well as all the other objects. Different objects may be expected to have different colors, therefore the most efficient way to identify some definite objects in an image is to seek for some definite colors at definite sizes. Similarly, theinformation which describes the motion of the end effector of a manipulator can be obtained from image processing.

The motion curves of manipulators are generally produced by time based mathematical functions, such as polynomial functions, spline functions, cam motions, rational functions and others. However, polynomial and spline functions have the most suitable forms since they lend themselves naturally to the solution of the types of problems involving arbitrary constraints [2]. They give continuous motion curves (i.e. position, velocity and acceleration). They are explicit functions of time, have a unique solution and strong sign-regularity. Especially, polynomial functions are easy to specify boundary conditions for any derivative of the function, such as velocity and acceleration, however, these functions may be produce inadequate or unexpected curves for some cases [3].

The control scheme for robotics system is trajectory tracking for which some control techniques are applied to obtain acceptable results from a system. Conventional control methods, such as computed torque, PD or PID, are effective for some simple processes at low speeds. However, there are several cases where precise tracing of a trajectory under different payloads require more advanced control technique to ensure stability of the process. Adaptive control is one of the important method of advanced control. It can be divided into many groups, however, two of them can be used for practical applications. These are self tuning adaptive control and model reference adaptive control. Self tuning adaptive control can be further divided into two sub-groups as implicit and explicit self tuning adaptive control. The main idea of adaptive control is based on continuous estimation of the parameters of the controller choosing a time-series difference equation model which describes the input-output measurements in the system. Since the system dynamic is assumed unknown, the parameters in such model can be computed recursively on-line at each sampling instant.

Adaptive control is not a new idea, however, the complexity of its algorithm was the most important difficulty for practical and theoretical study for a long time. Kalman [4] has proposed the first adaptive self tuning algorithm. Astrom and Wittenmark [5] have combined the recursive least-squares parameter estimation method with a minimal variance controller to calculate the parameters of the controller directly. Cegrell and Hedqvist in 1975 and Borisson and Syding in 1976 successfully applied the first self-tuning regulator [6, 7]. In the mean time, many other self-tuning adaptive control studies have been reported [8-11]. Nevertheless, nearly all the applications of self-tuning adaptive control in robotics remain in computer simulation studies.

2 SYSTEM HARDWARE

The robot has a spherical manipulator. Each joint of the robot is driven by a hydraulic actuator controlled by electro-hydraulic servo valves and these valves are also controlled by a 25 MHz PC computer. The supply of hydraulic pressure to the servo valves is provided by 22 kW hydraulic power unit capable of supplying up to 95 *lit* / min at 0.12 MP. The main parts of the robot are:

1) Pulnix Tmc-74 (NTCS) color video camera.
2) Raster Ops video color board 364.
3) Apple Machintosh fxI computer.
4) Hydraulic actuated spherical manipulator.
5) (I/O) interface card.

3 DETERMINATION OF CO-ORDINATES OF PATCHES

It is commonly accepted that a complete description of a point in a color image requires the specification of three mutually independent parameters [12, 13], that are so called primary colors (red "R", green "G", and blue "B"). Images can be analyzed by different standard methods which are Colorimetric [14], Nevatia transformation [15], and NTSC [16] color systems. In this work NTCS color system is used.

4.1 NTSC Color System

The three tristimulus values red (R), green (G), and blue (B) are transformed into another set of tristimulus values denoted by Y, I, and Q, which are related to R, G, and B by:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.596 & -0.272 & -0.322 \\ 0.211 & -0.523 & -0.312 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (1)$$

The Y component is called the *luminance component* since it roughly reflects the luminance. It is primarily responsible for the perception of the brightness of a color image and can be used to form a black and white image. The I and Q components are called the *chrominance components*, and they are primarily responsible for the perception of the hue and saturation of a pixel. In the NTSC color system, threshold values T' and T" for the hue and saturation can be arbitrarily defined for a certain color and while searching for it. The centroid of color patch primitives are used as matching strategies. In order to define the co-ordinates of the color spots in a short time color discontinuity criterion is used for the image processing.

4. TRAJECTORY PLANNING

The activity of converting the description of a desired task to a trajectory by defining time sequences of configurations of the end-effector of a manipulator between start and final positions is referred to as trajectory planning.

There are many mathematical functions can be used to describe a trajectory, however, cubic spline function is used for the generation of motion curves in this study. Because, its curves are more predictable in comparison to curves of polynomial interpolation due to their low degree.

4.1 Cubic Splines

The generation of interpolation spline curves is a useful and powerful tool in computer-aided design. Several methods have been developed to control the shape of the cubic interpolation such as those in [17-19].

A procedure to generate a series of cubic polynomials that pass through a set of (1,2...n) data points and has continuity of slope and curvature is described below. Equations for an interval which lie between two points (t_i, y_i) and (t_{i+1}, y_{i+1}) are:

$$y = a_i(t-t_i)^3 + b_i(t-t_i)^2 + c_i(t-t_i) + d_i \qquad \text{(i=0,1,2,3...n)} \tag{2}$$

Taking the first and second derivative of this equation gives:

$$\dot{y} = 3a_i(t-t_i)^2 + 2b_i(t-t_i) + c_i \tag{3}$$

$$\ddot{y} = 6a_i(t-t_i) + 2b_i \tag{4}$$

Equating the function to the two data points and substituting h_i for $(t-t_i)$ gives:

$$y_i = d_i \tag{5}$$

$$y_{i+1} = a_i h_i^3 + b_i h_i^2 + c_i \tag{6}$$

$$\dot{y}_i = c_i \tag{7}$$

$$\dot{y}_{i+1} = 3a_i h_i^2 + b_i h_i + c_i \tag{8}$$

$$\ddot{y}_i = 2b_i \tag{9}$$

$$\ddot{y}_{i+1} = 6a_i h_i + 2b_i \tag{10}$$

In order to satisfy the interpolation and provide continuity for the first and second derivatives at the ends of the interval, it is most convenient to express the coefficients a_i, b_i, c_i and d_i by means of the given function values y_i, y_{i+1} and the unknown second derivatives $\ddot{y}_i, \ddot{y}_{i+1}$ at the two ends of the interval $t_i - t_{i+1}$, so that:

$$a_i = \frac{1}{6h_i}(\ddot{y}_{i+1} + \ddot{y}_i) \tag{11}$$

$$b_i = \frac{1}{2}\ddot{y}_i \tag{12}$$

$$c_i = \frac{1}{h_i}(\ddot{y}_{i+1} - \ddot{y}_i) - \frac{1}{6}h_i(\ddot{y}_{i+1} + \ddot{y}_i) \tag{13}$$

$$d_i = y_i \tag{14}$$

We can invoke the condition that the slope of the two functions which join at (t_i, y_i) must be equal. The slope of the function for the i^{th} interval at the left end is:

$$\dot{y}_i = c_i \tag{15}$$

In the previous interval from t_{i-1} to t_i the slope at its right end will be:

$$\dot{y}_i = 3a_{i-1} - h_{i-1}^2 + 2b_{i-1}h_{i-1} + c_{i-1} \tag{16}$$

Equating equations (15) and (16) and substituting for a_i, b_i, c_i and d_i gives:

$$\dot{y}_i = \frac{y_{i+1} - y_i}{h_i} - \frac{2h_i\ddot{y}_i + h_i\ddot{y}_{i+1}}{6} = 3\frac{\ddot{y}_i - \ddot{y}_{i-1}}{6h_{i-1}}h_{i-1}^2 + \ddot{y}_{i-1}h_{i-1} + \frac{y_i - y_{i-1}}{h_{i-1}} - \frac{2h_i\ddot{y}_{i-1} + h_{i-1}\ddot{y}_i}{6} \tag{17}$$

Simplifying the equation we obtain:

$$h_{i-1}\ddot{y}_{i-1} + (2h_{i-1} + 2h_i)\ddot{y}_{i+1} + h_i\ddot{y}_{i+1} = 6\left[\frac{y_{i+1} - y_i}{h_i} - \frac{y_i - y_{i-1}}{h_{i-1}}\right] \tag{18}$$

Equation 18 applies at each interval point (from $i = 2$ to $i = n-1$). This gives $(n-2)$ equations and "n" unknown values of $\ddot{y}_i$. We can get two additional equations for the end points of the whole curve involving $\ddot{y}_1$ and $\ddot{y}_n$. Arbitrary values may be specified but for a natural spline they should be equal to zero.

5 EXPLICIT SELF-TUNING ADAPTIVE CONTROL

The structure of explicit self-tuning adaptive control loop is shown in Figure 1, where the parameters of the system model are directly identified.

5.1 System Model

In order to implement the explicit self-tuning adaptive control, a digitized single-input/single-output (SISO) model is required to represent the input-output measurements of the system. Under a stochastic measurement environment with uncertainties, such as random disturbances etc., a linear discrete time-series of an Auto regressive Moving Average Exogenous (ARMAX) model is used to represent the system under the control as shown below.

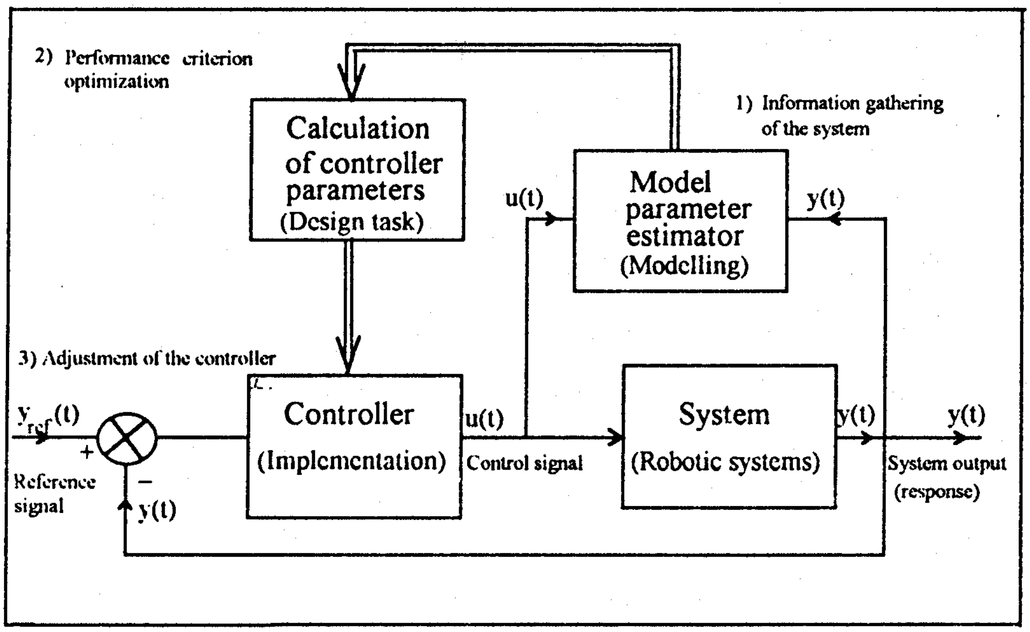

Figure 1 Self-tuning adaptive controller structure.

$$A(z^{-1})y(t) = z^{-d}B(z^{-1})u(t) + C(z^{-1})e(t) \tag{19}$$

$$A(z^{-1}) = 1 + a_1 z^{-1} + a_2 z^{-2} +, \dots, + a_{n_a} z^{-n_a}$$

$$B(z^{-1}) = b_0 + b_1 z^{-1} +, \dots, + b_{n_b} z^{-n_b}, \quad b_0 \neq 0$$

$$C(z^{-1}) = 1 + c_1 z^{-1} +, \dots, + c_{n_c} z^{-n_c}$$

Where n_a, n_b and n_c is the order of $A(z^{-1}), B(z^{-1})$ and $C(z^{-1})$ polynomials respectively.

5.2 Parameter Estimation

Parameter estimation is the most important part of the self-tuning algorithm. The main idea is to update continuously the parameters in the digital system model. Several powerful methods to estimate model parameters have been described in the literature [20-22]. These are least square method, recursive least square method, instrumental variable, recursive extended least squares method and U-D factorization method and etc.

The unknown parameters of model in Equation (19) are estimated recursively by on-line U-D factorization method. When these estimated parameters are replaced with the unknown parameters the following model equation can be obtain:

$$\hat{A}(z^{-1})y(t) = z^{-1}\hat{B}(z^{-1})u(t) + e(t) \tag{20}$$

The difference equation model with known numerical values for the parameters is next used to design the controller.

5.3 Controller Design Based on Minimum of Performance Criterion

The explicit self-tuning adaptive controller is designed so as to make the output of the system track the desired trajectory by minimizing a performance criterion while the energy associated with the input is kept at a minimum. These goals are achieved by choosing the following convenient quadratic function which is the variance of generalized cost function in the position error and energy of the control input appropriately weighted [11]:

$$J = E\left\{\left[y(t+1) - y_{ref}(t+1)\right]^2 + \lambda u^2(t) \middle| \Sigma(t)\right\} \tag{21}$$

Where J refers the performance index, $y_{ref}(t)$ is the desired trajectory of the system in discrete points. λ is the weighting factor which can be pre specified as a non negative number. The weighting factor can be selected to adjust the trade off between the tracking accuracy and the magnitude of the control signal. $E\{|\Sigma(t)\}$ indicates a conditional expectation operation when the system outputs with past values $y(j)$, $j \le t$ and the past control inputs $u(j)$, $j < t$ indicated by $\Sigma(t)$ are given. If the equation $e(t) \equiv 0$, then the expectation operation is not needed.

To minimize performance's index with respect to control input, the expression of $y(t+1)$ is required in terms of $y(j)$. By estimating model parameters for $y(t+1)$, the output can be written as:

$$y(t+1) = -\hat{a}_1 y(t) -,\ldots,-\hat{a}_{n_a} y(t+1-n_a) + \hat{b}_0 u(t) +,\ldots,+\hat{b}_{n_b} u(t-n_b) + e(t+1) \tag{22}$$

The last term in Equation (22) is future disturbance that can not be predicted and an estimated system output can be written as:

$$y(t+1|t) = -\hat{a}_1 y(t) -,\ldots,-\hat{a}_{n_a} y(t+1-n_a) + \hat{b}_0 u(t) +,\ldots,+\hat{b}_{n_b} u(t-n_b) \tag{23}$$

Where $y(t+1|t)$ is a one-step ahead predictor of $y(t+1)$. Under this consideration, the $y(t+1|t)$ is used instead of $y(t+1)$ in the performance index.

$$J = \left[y(t+1|t) - y_{ref}(t+1)\right]^2 + \lambda u^2(t) \tag{24}$$

Substituting Equation (23) into (24) gives

$$J = \left[-\hat{a}_1 y(t) -,\ldots,-\hat{a}_{n_a} y(t+1-n_a) + \hat{b}_0 u(t) +,\ldots,+\hat{b}_{n_b} u(t-n_b) - y_{ref}(t+1)\right]^2 + \lambda u^2(t) \tag{25}$$

To minimize the performance index, its derivative with respect to the control signal at time t is set equal to zero. Finally, the admissible control input is obtained from Equation (25) as:

$$u(t) = \frac{\hat{b}_0}{\hat{b}_0^{\,2} + \lambda}\left[y_{ref}(t+1) + \hat{a}_1 y(t) +,\ldots,+\hat{a}_{n_a} y(t+1-n_a) - \hat{b}_1 u(t-1) -,\ldots,-\hat{b}_{n_b} u(t-n_b)\right] \tag{26}$$

6 IMPLEMENTATION

For the experimental study, some color spots are placed arbitrarily in two dimensional vertical plane in Cartesian co-ordinate system. The image of working plane is first grabbed by a video camera when the program is executed, as shown in Figure 2. Then, the co-ordinates of these precision points are determined by image processing. See Figure 3, where "o" and "+" represent the co-ordinates of the spots obtained from the image processing and direct measurements respectively. The trajectory is generated on-line using the information obtained from the image processing and this trajectory is used to control the hydraulic robot.

The results taken from the hydraulic robot are given in Figures 4.a-g. Figures 4a-b show the input output curves for first and the second joints of the manipulator when the system is controlled with a conventional control method, whereas Figures 4.c and d are taken when explicit self tuning control method is used. The variation of parameters for the system model at the revolute and prismatic joints are given in Figures 4e and f. Finally, the desired trajectory and the response of the manipulator at the end-effector level are shown in Figure 5g.

Several examples are tested on the robot. It is observed that co-ordinates of the precision points can be determined approximately %2 error and the images can be analyzed in a reasonable time (nearly 8 seconds). It is also seen that a robot can be controlled more efficiently when self tuning adaptive is applied as shown in Figure 4 c and d.

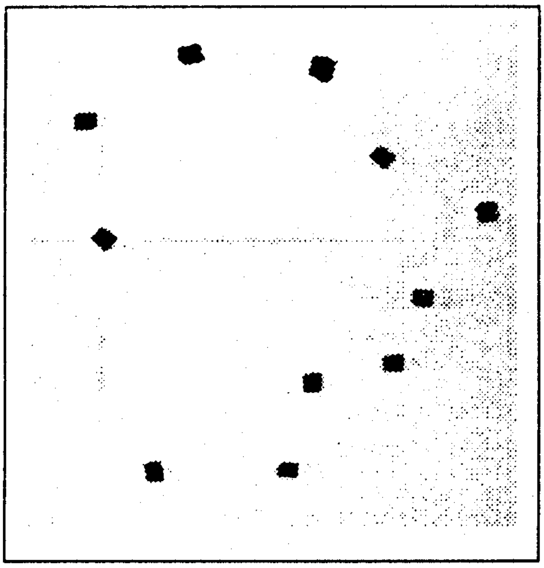

Figure 2. Positions of the spots in the image taken from the video camera.

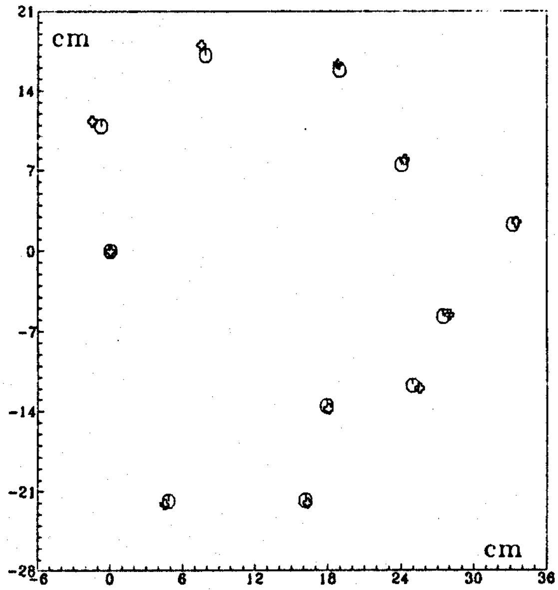

Figure 3. Actual and computed co-ordinates of spots.

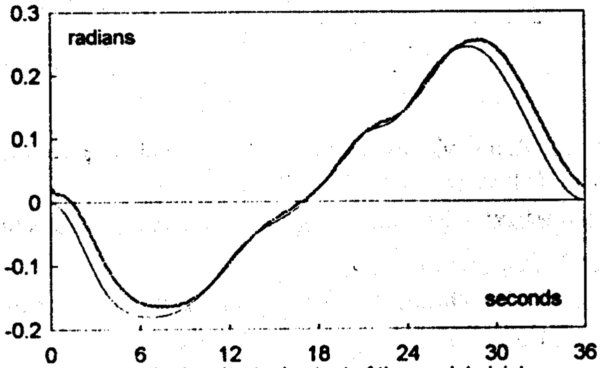

Figure 4.a. Desired and actual output of the revolute joint with conventional control.

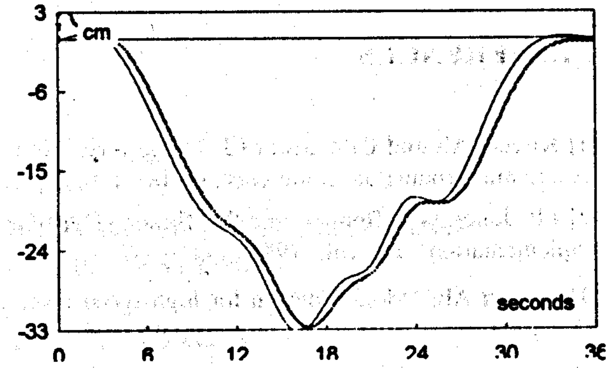

Figure 4.b. Desired and actual output of the prismatic joint with conventional control

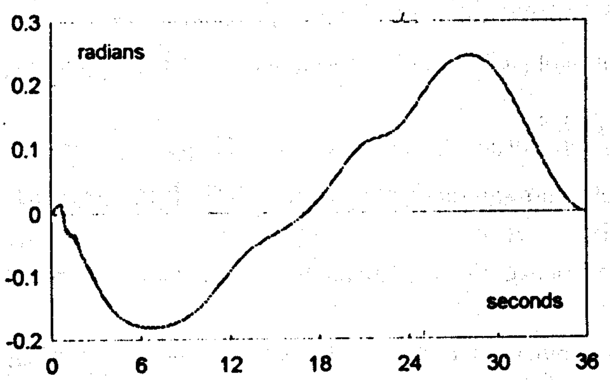

Figure 4.c. Desired and actual output of the revoulte joint with explicit self tuning.

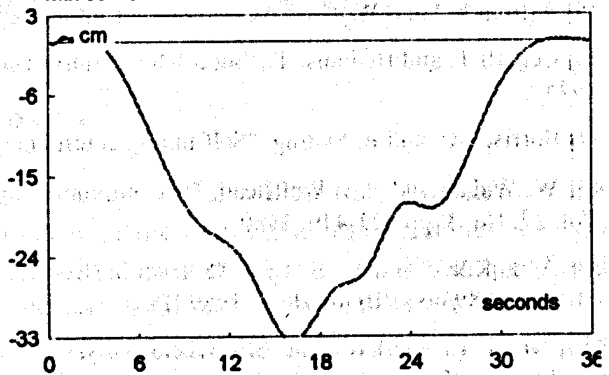

Figure 4.d. Desired and actual output of the prismatic joint with explicit self tuning.

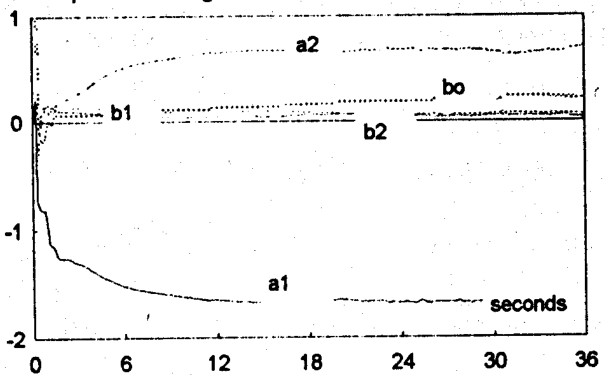

Figure 4.e. System parameters for the revoulte joint.

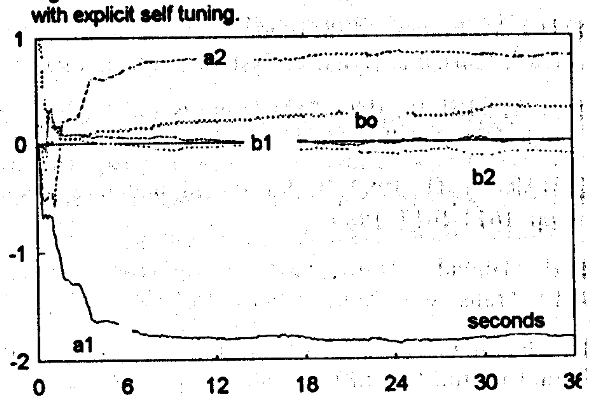

Figure 4.f. System parameters for the prismatic joint.

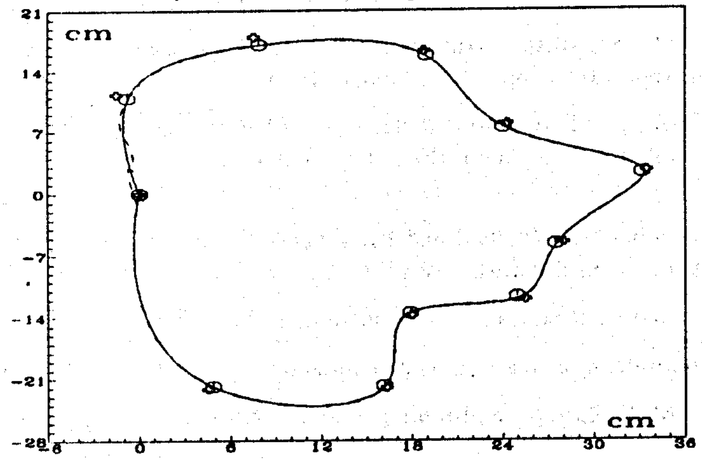

Figure 4.g. Desired output and actual ouput at the end-effector level.

7 REFERENCES

[1] Kirecci Ali and Gilmartin.M.J, "Trajectory Planning for Robot Manipulators", 6th international Machine Design and Production Conference, pp 181-190, September, METU, ANKARA, TURKEY, 1994.

[2] J.R. Jones, G.T. Rooney, and P.J. Fisher, "Dynamically Smoother Motion for Industrial Robots, Design and Implementation", IMechE, 1984.

[3] Kirecci Ali, "Motion design for high-speed machines.", PhD. Thesis, Liverpool Jonh Moores University, UK., 1993.

[4] Kalman R. E., "Design of self-optimizing control system.", Transactions ASME, Vol. 13, pp 59-75, 1958.

[5] Astrom K.J and Wittenmark B., "On self-tuning regulators.", Automatica, Vol. 9, pp 189-199, 1973.

[6] Cegrell T. and Hedquist T.,"Successful adaptive control of paper machines",Automatica, Vol. 11, pp 53-59, 1975.

[7] Borrison U. and R. Syding, "Self-tuning control of an ore crusher.", Automatica, Vol. 12, pp 1-7, 1976.

[8] W. Wahab and P.E. Wellstead, "Self-tuning control of a robot manipulator arm.", INT. J. PROD. RES., Vol. 27, No. 3, pp 423-449, 1989.

[9] A. J. Koivo and K. S. Lee., "Self-tuning control of planar two-link manipulator with non-rigid arm." CH2750-8/89/0000/1030$01.00, 1989 IEEE

[10] M. A. El-Sharkawi and Siri Weerasooriya., "Development and implementation of self-tuning tracking controller for Dc motors.", IEEE Transactions on Energy Conversion, Vol. 5, No. 1, March 1990.

[11] J. M. Finney, M. S. Bloor and G. S. Gill, "A pole-assignment controller for an electrohydraulic cylinder drive.", Journal of Dynamic Systems, measurement, and control ASME, Vol. 107, pp 145-150, June 1985.

[12] Wandell, B., 1987. "The Syntesis and Analysis of Color Images", IEEE Trans. Pattern Anal. Machine Intell., Vol. PAMI-9, No.1.

[13] Healey, G., 1989. "Color discrimination by computer", IEEE Trans. Syst. Man. Cybern., Vol. 19, No. 6, pp. 1613-1617, 1989.

[14] Dhond, U.R., and Aggarwal, J.K., 1989. "Structure from Stereo-A Review." IEEE Trans. Syst., Man. Cybern., Vol.19, No. 6, pp. 1489-1509.

[15] Nevatia, R., 1977."A Color Edge Detector and Its Use in Scene Segmentation" IEEE Trans. Syst. Man. Cybern., Vol. SMC-7, No. 11, pp. 820-826.

[16] Lim.J.S.,1990. Two-Dim. Signal and Image Processing, Prentice Hall, Englewood Cliffs New Jersey.

[17] Y. Fletcher and D.F. McAllister, Automatic Tension Adjustment for Interpolatory Splines, IEEE Computer Graphics and Applications, pp. 10-17, January 1990.

[18] M.K. Jouaneh, Z. Wang and D.A. Dornfeld, Trajectory Planning for Co-ordinated Motion of Robot and a Positioning Table: Part 1-Path Specification, IEEE Transactions on Robotics and Automation, V.6, N.6, pp. 735-744, 1990.

[19] T.A. Foley, Interpolations with Interval and Point Tension Controls Using Cubic Weighted V-Splines, ACM Transactions on Mathematical Software, V.13, N.1, pp. 68-96, 1987.

[20] Torsten Soderstrom and Peter Stoica., "System identification.", 1988.

[21] M. Vidyasagar, "Controller system synthesis-a factorization approach.", MIT Press, 1985.

[22] P. E. Wellstead and M. B. Zarrop, "Self-tuning systems control and signal processing.", John Wiley and Sons, 1991.