DESIGN AND IMPLEMENTATION OF A VISION BASED CONTROL SYSTEM TOWARDS MOVING OBJECT CAPTURE FOR A 3-JOINT ROBOT

Raşit Köker[1], Hüseyin Ekiz[2], Ali Fuat Boz[2]

Abstract

Robot control systems are designed for different purposes such as picking and placing, painting, welding, and carrying. Industrial robots use vision systems to perform their tasks. Due to the growing cost of row materials, workmanship, energy, and growing up competition environment, manufacturers are forced to produce cheaper and high quality productions. This results new needs for automation techniques. In this paper, Generalized Predictive Controller (GPC) is designed to control a 3-joint robotic manipulator towards moving object capture. A Carima model is used in the controller. The system consists of a camera, a capture card, and software that include all the dynamics and kinematics of the robotic manipulator. Vision system, used in the implementation, also includes a conveyor system to analyze moving object images. At the end, the results obtained from the simulations are given.

Index terms—*Robot Vision, Robot Control, Carima model*

I. INTRODUCTION

Recently, automated picking and placing systems has received a great deal of attention in many industrial robotics applications. Likewise, much effort has been directed to automating various production activities in industry by applying machine vision technology. Moving object capture on a conveyor may be given as an example of an industrial machine vision application.

Robots have also been using extensively to perform the task of picking and placing. A robot can be defined as a multi degree freedom open loop chain of mechanical linkages and joints. These mechanisms driven by actuators are capable of moving an object in space from initial to final locations or along prescribed trajectories. Tracking accuracy is of prime concern in assembly task. Hence, a robot controller should be able to achieve fast and at the same time accurate tracking and positioning control of the end effector. The dynamic equations of a mechanical manipulator are highly nonlinear and complex. To overcome these difficulties, several advanced adaptive control techniques for dynamic control of mechanical manipulators have been proposed. GPC is chosen in this study. An efficient application of GPC algorithm towards moving object capture is presented in this study [1][2].

Robot control and vision simulation software are synchronized on online studies. Open-GL is used to observe the moving object images. It also gives the possibility of getting references easily related to the robot and conveyor. In this paper, GPC design using a Carima model is firstly given. Then, necessary algorithms for vision and control system are explained, and presented with their results. One of the most important limitations in this study is that the third dimension of object is accepted as known. The object position information is just obtained 2-D by using image-processing algorithms, and then it is joined with known height (z) information. In this study, it is assumed that robotic manipulator has a vacuum end effector.

II. GENERALIZED PREDICTIVE CONTROL ALGORITHM

Basically, GPC is the algorithm predicting y (t) output throughout finite horizon and, control signals string u (t) is computed using the minimization of a quadratic performans criterion according to the some assumptions interested in the control signals that implements appropriate response. The main point of the GPC is in the assumptions made about the future control trajectory. It can be designed as a SISO (Single Input Single Output) controller or MIMO (Multi Input Multi Output) controller. In this study GPC is designed as a MIMO controller. To design the controller, the discrete time CARIMA model is used in Eq. (1). A block diagram of the joint trajectory control for a robot that uses MIMO Generalized Predictive Control is given in Fig. (1) [3].

$$A(z^{-1}).y(t) = B(z^{-1}).u(t-1).e(t) \qquad (1)$$

A. A CARIMA Model For Manipulator Motion

A CARIMA model is suggested for the motivation of (2) for the motion of a robotic manipulator. As is well known, the parameters in such an assumed model can be estimated on-line using recursive equations. They can be obtained by minimizing the sum of the squared errors. Firstly, the resulting equations for the recursive least-square estimation algorithm (RLS) will be reviewed here and secondly, the algorithm for a predictive controller will be presented using the model with estimated parameters.

[1] SAKARYA UNIVERSITY, Engineering Faculty, Computer Engineering Department, 54187 ADAPAZARI - TURKEY (rkoker@sakarya.edu.tr)
[2] SAKARYA UNIVERSITY, Technical Education Faculty Electronics Education Department, 54187 ADAPAZARI – TURKEY
(ekiz@sakarya.edu.tr - afboz@sakarya.edu.tr)

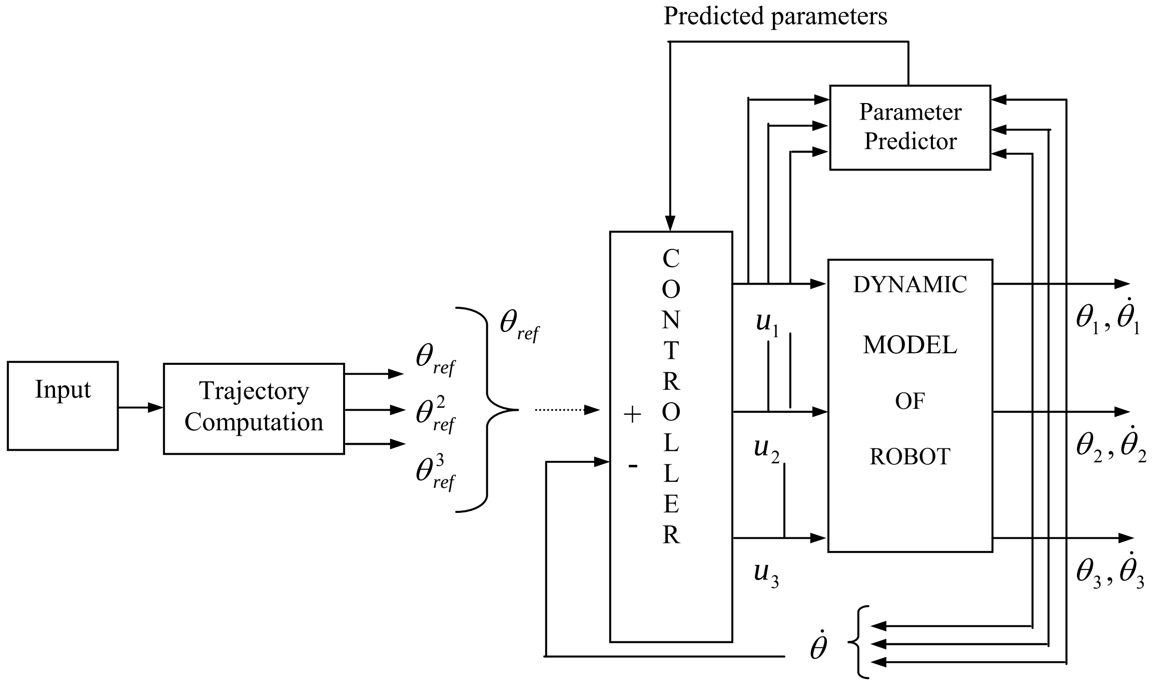


Figure : 1 A block diagram of the joint trajectory control for a robot that uses MIMO Generalized Predictive Control.

y (t) = a_0 + A_1.y (t-1) + A_2. y (t-2) + B_1.u (t-1) +

+ B_2 . u (t-2) + e (t) (2)

The CARIMA model is assumed to have the same number of inputs and outputs. The CARIMA model will be written in the following general form:

$$y(t) = A(q^{-1}).y(t) + B(q^{-1}).u(t-1) + e(t) \quad (3)$$

where the m-dimensional output vector y and the input vector u have as the ith component the output y_i and the output u_i of joint i, respectively, and $i = 1,2,....,m$. The equation error vector e(t) in the CARIMA model is considered to be of the form:

$$e(t) = \xi(t)/\Delta \quad (4)$$

Where Δ is the differencing operator $1 - q^{-1}$ and $\xi(t)$ is an uncorrelated random sequence. The argument q^{-1} is a backward shift operator. The (mxm) matrices A and B are polynomials defined by:

$$A(q^{-1}) = A_1.q^{-1} + ... + A_{n_a} q^{-n_a}$$

$$B(q^{-1}) = B_0 + B_1.q^{-1} + ... + B_{n_b} q^{-n_b} \quad (5)$$

To estimate the parameters in (3), the matrix Θ and the vector Φ are defined:

$$\Theta = \left[A_1,.....A_{n_a}; B_0,.....B_{n_b}\right]^T = \left[\Theta_1,....\Theta_m\right] \quad (6)$$

Where the superscript denotes the transposition. Then

For i = 1,2,....,m.

$$\Theta_1 = \left[a_{11}^1,....,a_{1m}^1, a_{11}^2,....,a_{1m}^2,....; b_{11}^0,....,b_{1m}^0,....b_{11}^{n_b},....b_{1m}^{nb}\right]^T$$

$$\Phi[(t-1)] = \left[y^T(t-1),....,y^T(t-n_a); u^T(t-1),...,u^T(t-n_b)\right]^T \quad (7)$$

Equation (3) can be written as follows [1]:

$$y(t) = \Theta.\Phi^T(t-1) + e(t) \quad (8)$$

III. IMAGE PROCESSING

The related image processing algorithms are given in this section. In prepared software, care is taken of time due to its importance in the system to preserve the practicality.

A. Low-Level Image Processing

An experimental captured image is a 256-colored gray image. In prepared software, initial image size is reduced down to 256x256 pixels to decrease the processing time. A median filter is used to eliminate the undesirable effects due to the noise and other effects. Because of the sensitivity of moment invariants to the noise, filtering is important in this system. The illumination is provided to have an object image without shadow and reflection using two light sources. To automate the thresholding operation, we have used the method of Optimal Thresholding by Minimizing Within-Group Variance. This method is a reasonably good threshoulding method for more uniformity, better shape of the object in the binary image and short processing time [4][5][6].

B. Intermediate Level Image Processing

Edge detection can be categorized in intermediate level processing. This stage of the system aims to extract data for the high-level image processing. Among the large number of edge detection algorithms, Sobel is used due to its popularity on computational simplicity [7]. The centroid and moment invariants are computed by using edge map of image in different angles of the objects. The samples of processed object images are given in figure 2.

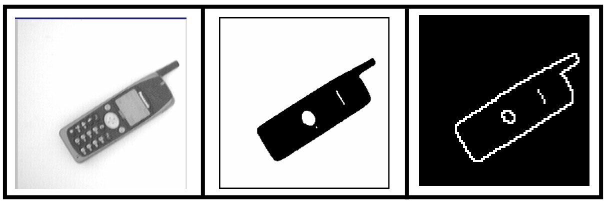


Figure : 2 Samples of processed images

C. Calculation of the centroid

The application of moments provides a method of describing the properties of an object in terms of its area, position, orientation and other precisely defined parameters. The basic equation defining the moment of an object is given as below.

$$m_{ij} = \sum_x \sum_y x^i . y^j . f(x,y) \qquad (9)$$

Where,

x, y : Pixel coordinates

$f(x,y)$: Pixel brightness

Zero- and first order moments can be given as;

$$m_{00} = \sum_x \sum_y f(x,y) \qquad (10)$$

$$m_{10} = \sum_x \sum_y x.f(x,y) \qquad (11)$$

$$m_{01} = \sum_x \sum_y y.f(x,y) \qquad (12)$$

The equation (10) known as first-order moment (m_{00}) can be used to compute the object area in the binary image. The 'centroid' is a good parameter for specifying the location of an object. It is the point having coordinates x', y' such that the sum of the square of the distance from it to all other points within the object is a minimum [8][9]. The centroid can be expressed in terms of moments as;

$$x' = \frac{m_{10}}{m_{00}} \quad (13) \qquad y' = \frac{m_{01}}{m_{00}} \quad (14)$$

IV. HARDWARE AND SOFTWARE IMPLEMENTATION

The system is experienced on a moving conveyor. A view from simulation study is given in figure 3. The work area is selected on the conveyor-moving surface. At the beginning of the conveyor we have captured images sequentially with intervals of 200 ms. By using these captured images we have computed the area of the object. We have compared the area changes to understand if the whole object image is ready to be captured or not in the work area. When the whole object is at the work area we have captured the image to be processed. Firstly, centroid is computed to have information about object position. The velocity of the conveyor is computed by using centroid changes. Using this velocity we have calculated the position of object after a certain time approximately at the end of the conveyor. This position is an appointment point with robot end-effector. The z dimensions of objects have written to a file, and taken by using the prepared software. The robot control software should be coded to get these position information of object to use in solving inverse kinematics problem.

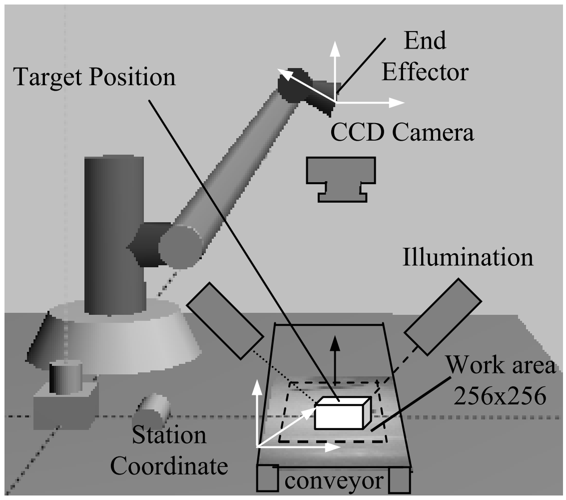


Figure : 3 A view from the system towards moving object

The system has been implemented by using a Pentium III-750 computer. All processes have taken approximately 160 ms. Median filtering and edge detection takes the much time 110 ms. and 50 ms, respectively in real time experiment. A TV-capture card, which has a BT878 chipset, is used to capture image frames. All image-processing algorithms with real time capture process are implemented in Delphi Programming Language. To capture image frames we have used the drivers of the capture card (DLL File). Thus, TV-capture card is on-line controlled by prepared software.

The parameters of manipulator used in the simulations are shown in table 1, and its model is given in figure 5. The simulations were done using computed object centroids (x,y) via image processing algorithms. The third dimension z is accepted known for the object. Obtained (x,y,z) coordinates is the final position of end effector.

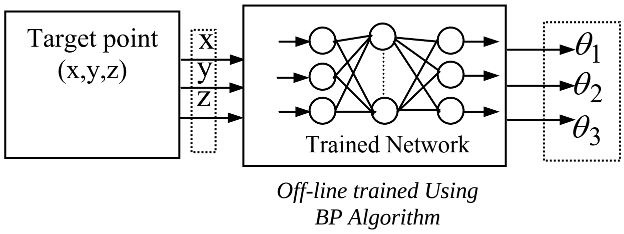


Figure: 4 The inverse kinematics solution scheme.

The object position is used as target position. GPC is applied to take the end effector to target position. The (x,y,z) point is firstly transformed to the angular position (θ_1, θ_2, θ_3). This is known as inverse kinematics solution. In this study, we have done inverse kinematics solution using neural network, and the solution method is explained below [10][11].

A Back-Propagation Neural Network with sigmoidal activation function is used to solve inverse kinematics problem. The schematic diagram of the system is given in Figure 4. Firstly, some points in the work volume of manipulator are taken to use in the cubic path planning to generate the (θ_1, θ_2, θ_3) joint angles according to different (x,y,z) cartesian coordinates. These values were recorded in a file to form the learning set of the neural network. We obtained the angles (θ_1, θ_2, θ_3) from a certain cubic by sampling 6000 for orbit trajectory a given job. 5000 of these data was used in training of neural network, and other 1000 were used in testing. The training process is completed until the error is acceptable. It has been completed approximately in 3.000.000 iterations. The neural network is designed including 40 perceptrons in hidden layer, and 3 perceptrons in input and 3 perceptrons in output layer. Learning rate (η) and the momentum rate (α) are experimentally chosen as 0.3 and 0.8, respectively. The number of perceptrons in hidden layer is also found experimentally. Error at the end of the learning is 0.000121 for training set.

Artificial neural networks based inverse kinematics solution is especially more useful for robot models, which has redundant configuration. In this study, this solution is just tested for out robot model, and by using back propagation neural networks.

i	Joint # 1	Joint # 2	Joint # 3	Units
$\mathbf{m_i}$	13.1339	10.3320	6.4443	kg.
$\mathbf{a_i}$	0.1588	0.445	0.10	m.
α_i	$\pi/2$	0.0	$\pi/2$	Radian
$\mathbf{x}_i^*$	-0.0493	-0.1618	0.0	m.
$\mathbf{z}_i^*$	0.0	0.0	0.2718	m.
$\mathbf{k}_{i11}$	5.6064	3.929	82.0644	m^2x10^{-3}
$\mathbf{k}_{i22}$	8.9196	47.8064	81.9353	m^2x10^{-3}
$\mathbf{k}_{i33}$	13.2387	45.4838	1.400	m^2x10^{-3}

Table: 1 The parameters of chosen robot model [12][13]

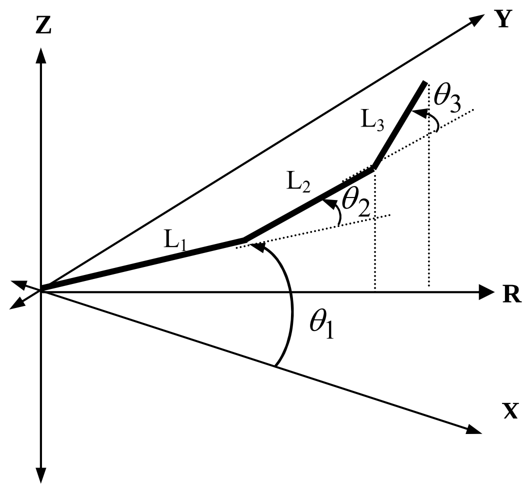
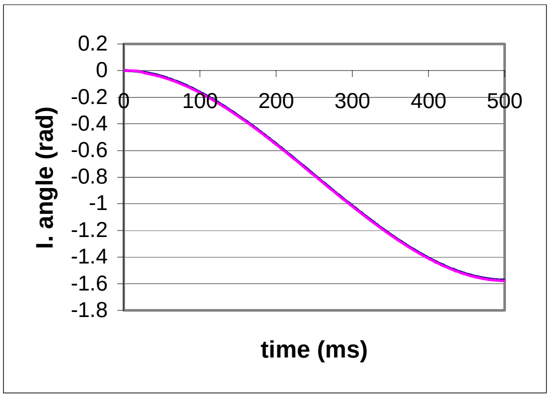
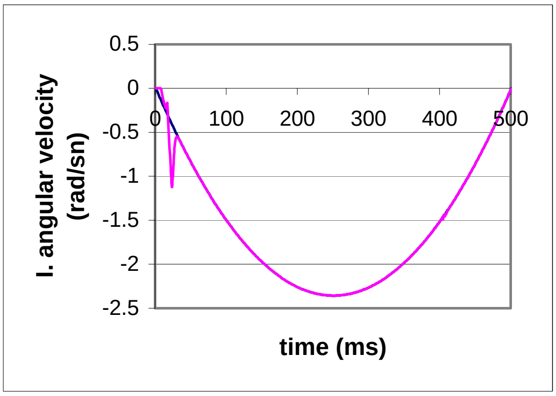
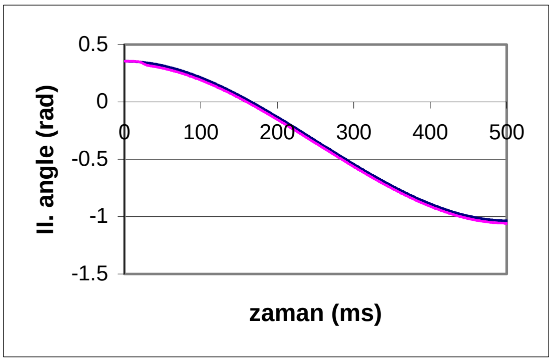


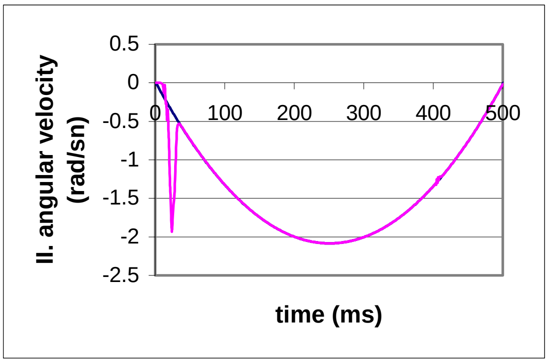
Figure: 5 The manipulator model used in this study



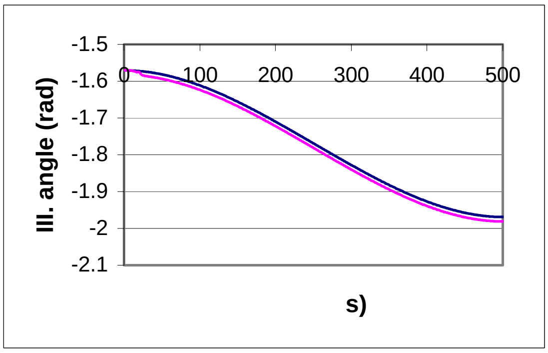


(a) **(d)**





(b) **(e)**



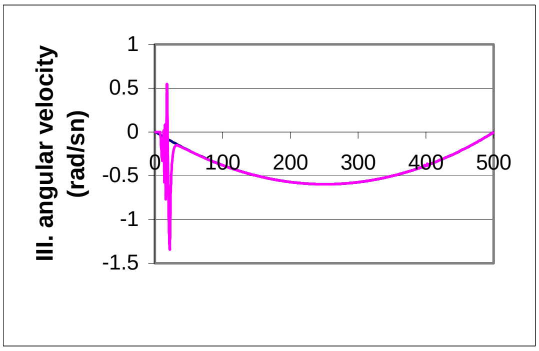


Figure: 6 The Simulation Results of GPC algorithm

(a) (b) (c) Obtained joint positions and reference positions
(d) (e) (f) Obtained joint speeds and reference speeds

V. RESULTS AND DISCUSSION

CARIMA difference equations are introduced to model the motion of the joints in a robotic manipulator. The inputs in the model are the motor torques, and the outputs are selected as the velocities of the manipulator joints.

A MIMO GPC controller is presented in this paper with its efficient application based on a vision based industrial application. The used Generalized Predictive Control Scheme does not require a priori knowledge of the physical parameters of the manipulator or its configuration.

The obtained speed and position results in simulation-based studies have been given in figure 6, graphically. It was observed that some of given difficult trajectories have been followed by robotic manipulator with satisfying results.

It has been observed that there are some disturbances at the beginning of the control of the speeds in graphics. But these disturbances are controlled in future steps. So that, the system can be improved with taking these unwanted disturbances at the beginning under control as a future study.

It should be taken care at limitations of the study as stated earlier. The first one is the height (z) of object is accepted known. Other one, it is assumed that robotic manipulator has a vacuum end effector.

VI. REFERENCES

[1] C. Özsoy, and R. Kazan, "Self-Tuning Predictive Controllers For Robotic Manipulators", 2nd Project on Cim and Robotics Applications, September 1991, BELGRAD.

[2] C. H. Liu, "A Comparison of Controller Design and Simulation For an Industrial Manipulator", IEEE Transaction on Industrial Electronics, Vol. IE-33, No 1, February 1986.

[3] R. Köker, Cemil Öz, and R. Kazan, "Vision Based Robot Control Using Generalized Predictive Control", Intern. Conf. on Electrics and Electronics Engineering ELECO'01, Bursa, 7-11 Kasım 2001, TURKEY.

[4] E. Çokal, K. Özdemir, A. Alansal, and A. Erden, "Development of a Vision Based System For Mobile Robots", 7th Int. Mach. Design and Production Conference, Turkey, Sept. 11-13 1996, pp. 595-604.

[5] K.S. Fu, R.C. Gonzalez, and C.S.G. Lee, Robotics, McGraw-Hill, 1987, p. 362.

[6] Sahoo P. K., Soltani S., "A Survey of Thresholding Techniques", Computer Vision Graphics and Image Processing, 41, 1988, pp.233-260.

[7] P.A. Laplante, and A.D. Stoyenko, Real Time Imaging, IEEE Press, 1996, p.231.

[8] G. J. Awcock, and R. Thomas, Applied Image Processing, McGraw Hill, Inc., 1996, p. 148.

[9] G. I. Chicu and J. Hwang, "A Neural Network-Based Stochastic Active Contour Model (NNS-SNAKE) for Contour Finding of Distinct Features", IEEE Transactions on Image Processing, Vol.4, No.10, October 1995, pp. 1407-1416.

[10] R. Köker, C. Öz, A. Ferikoğlu, "Development of A Vision Based Object Classification System For A Robotic Manipulator", 8th IEEE International Conference on Electronics, Circuits and Systems, 2-5 EYLÜL 2001, Malta.

[11] R. Köker, C. Öz, and Tarık Çakar, "A Study of Neural Network Based Inverse Kinematics Solution For a 3-joint Robot", IMS 2001, 30-31 Augst 2001, Sakarya, Turkey.

[12] D. W. Clarke, C. Mohtadi, and P. S. Tuffs, "Generalized Predictive Control: A New Robust Self-Tuning Algorithm in Landau I. D., and L. Dugards (Eds.). Commande Adaptive-Aspects Pratiques et Theories, pp. 209-228, Masson, Paris, 1997.

[13] R. Köker, "Model Based Intelligent Control Of 3-Joınt Robotic Manipulator With Machine Vision System", Ph.D. Thesis, Sakarya University, TURKEY (in Turkish).