Design and Implementation of an Intelligent Predictive Controller towards Vision-based Application

Rasit Koker[1] Server Firat[2] Mesut Kahriman[3]

[1] Sakarya University Engineering Faculty, Department of Computer Engineering 54187
Sakarya - TURKEY, rkoker@sakarya.edu.tr
[2] Sakarya University, Vocational High School, Electrical Department 54187
Sakarya - TURKEY
[3] Fatih Technical High School Electronics Department
Sakarya - TURKEY

Abstract

Recently, automation systems have taken a great deal of attention due to the advances in computer technology and expanding demand for such techniques in the robotics. Additionally, the usage of artificial intelligence has also begun to be used in the solution of many engineering problems. Artificial neural networks (ANN) are a widely used intelligent technique on this way. In this study, an intelligent predictive controller was designed for 3- joint robotic manipulator towards moving object capture. Generalized predictive control (GPC) algorithm and Elman neural network were used in the controller design. The GPC algorithm, which belongs to a class of digital control methods and known as Model Based Predictive Control, require long computational time and can result in a poor control performance in robot control. To overcome this problem, an Elman network controller was designed towards vision based application. The used algorithms and their results were given with their results in this paper.

1. INTRODUCTION

In recent years, due to the cost of growing raw materials, workmanship, energy and growing up competition environment, much effort has been directed to automating various production activities in industry by applying machine vision technology. The machine vision technology has begun to be developed rapidly, parallel to the development on the computer technology. Robots has also begun to be used with vision systems, such as an object picking up system using a manipulator and a vision system including a camera, a capture card and a software. Furthermore, artificial intelligence techniques have taken great deal of attention in the solution of many engineering problems [1].

Generalized Predictive Control (GPC), introduced by Clarke and his co-workers [2] in 1987, belongs to a class of digital control methods called Model-Based Predictive Control (MBPC). MBPC techniques have been analyzed and implemented successfully in process control industries since the end of the 1970's and continue to be used because they can systematically take into account real plant constraints in real-time. GPC is known to control non-minimum phase plants, open-loop unstable plants and plants with variable or unknown dead time. It is also robust with respect to modelling errors, over and under parameterisation, and sensor noise. GPC has been originally developed with linear plant predictor models, which leads to a formulation that can be solved analytically, but it has also been proposed to be extended for the control of non-linear systems [3].

In this study, firstly the images are captured from the work area. These images are processed to separate the object image from background and, then computed the centroid of the object. This position information is used in the simulations as a target position. The simulation software includes all dynamics and kinematics model of robotic manipulators. GPC algorithm, image processing algorithms and simulation results are presented in this paper. On the other hand, the designed Elman controller is presented.

2. IMAGE PROCESSING

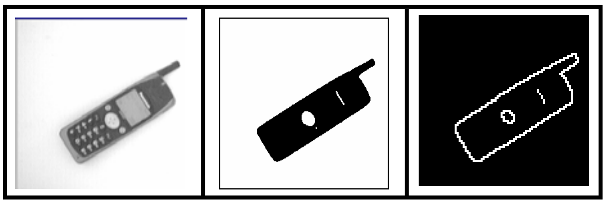
In this stage the related image processing algorithms are given. In prepared software, low and intermediate level image processing algorithms are used, and it is aimed to compute the position and speed of the object.

2.1 LOW-LEVEL IMAGE PROCESSING

An experimental captured image is a 256 shade 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 moments that are used to compute centroid of the object, 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 thresholding method for more uniformity, better shape of the object in the binary image and short processing time [4][5].

2.2 INTERMEDIATE LEVEL IMAGE PROCESSING

Edge detection can be categorized in intermediate level processing. This stage of the system aims to obtain edge map of image to use in the centroid computation. An edge is the place where there is a more or less abrupt change in gray level or colour. In an image most of the information lies on the boundaries between different regions. Using edge map of images decreases the processing time in image processing systems. Among the large number of edge detection algorithms, Sobel is used due to its popularity on computational simplicity [6]. The samples of processed object images are given in Figure.1.



(a) (b) (c)

Figure:1. Samples of processed images
a) Raw image
b) Thresholded image
c) Edge map of the image

2.3 CALCULATION OF THE CENTROID

The applications of moments provide 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. The height of the object is assumed known in this study.

$$m_{ij} = \sum_{x}\sum_{y} x^{i}.y^{j}.f(x,y) \qquad (1)$$

Where,

x ,y : Pixel coordinates
f(x,y) : Pixel brightness

Zero- and first order moments can be given as;

$$m_{00} = \sum_x \sum_y f(x,y) \qquad (2)$$

$$m_{10} = \sum_x \sum_y x.f(x,y) \qquad (3)$$

$$m_{01} = \sum_x \sum_y y.f(x,y) \qquad (4)$$

The equation (2) 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 [7]. The centroid can be expressed in terms of moments as

$$x' = \frac{m_{10}}{m_{00}} \qquad (5) \qquad\qquad y' = \frac{m_{01}}{m_{00}} \qquad (6)$$

3. THE DYNAMIC AND KINEMATICS MODEL OF THE ROBOT

A three-joint robot model shown in Figure 2 has been used in the simulation. The simulation software includes dynamics and kinematics equations for the given robot model. As it is well known, the dynamics of a robot arm [8] can be expressed by the following general set of equations given in eq. (7):

$$\sum_j d_{kj}(q)\ddot{q}_j + \sum_{i,j} c_{ijk}(q)\,\dot{q}_i\dot{q}_j + f_k(q) = \tau_k \qquad (7)$$

k = 1, . . . n, i = 1, . . ., n, j = 1, . . ., n and where;

q_j	jth generalized coordinate
q	generalized coordinate vector
τ_k	kth generalized force
n	number of joints
d_{kj}	inertial coefficients
c_{ijk}	centrifugal and Coriolis coefficients
f_k	loading item due to gravity

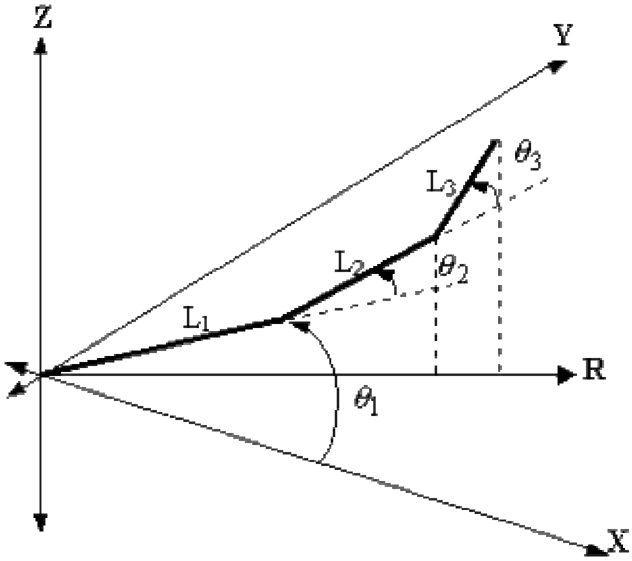


Figure 2. The manipulator model used in this study

4. THE CONTROLLER DESIGN

In this section, it is firstly aimed to present the design of the neural network, which is working online to produce applied torque values for each joint in the system. A Back-Propagation neural network that uses gradient descent error learning algorithm with sigmoid activation function is used to model GPC algorithm. The data prepared during the realization of traditional GPC algorithm have been used to train neural networks for designed vision system. A cubic trajectory is used in the simulations. The training process has been implemented only for some areas in the work volume of robot. To design a controller generalized for whole area of work volume of robot is a very time consuming study due to the training difficulty. Elman neural network is used to model the GPC controller in this study. The reason of the selection of Elman neural networks among several neural network architectures is that the presence of feedback loops has a profound impact on the learning capability of the network.

4.1 DATA PREPARATION

To train the neural network, a training set has been prepared by using the results of implementation of GPC. The manipulator has been controlled for different trajectories to generate the data for the training set. The designed neural network has 12 inputs and 3 outputs. To obtain the torque value at time “t” as an output, the torque values at time (t-1), (t-2), and y and $y_{references}$ at time (t-1) are used in input stage as 12 elements. These data has been generated using GPC controller for different trajectories. These trajectories have been selected uniformly to model the GPC control. In the preparation of training set, care has also been taken of payload variations. Payload variations are taken between 0 gram and 10000 gram. Due to the characteristic feature of sigmoid activation function used in the training of back propagation error-learning algorithm is used.

4.2 TRAINING OF THE ELMAN NETWORK

In the off-line training of the back propagation neural network, 10000 input and output vector sets are generated using simulation software. 9000 of these are used as learning set, and others are used in test. A neural network toolbox is used to train the network [9].

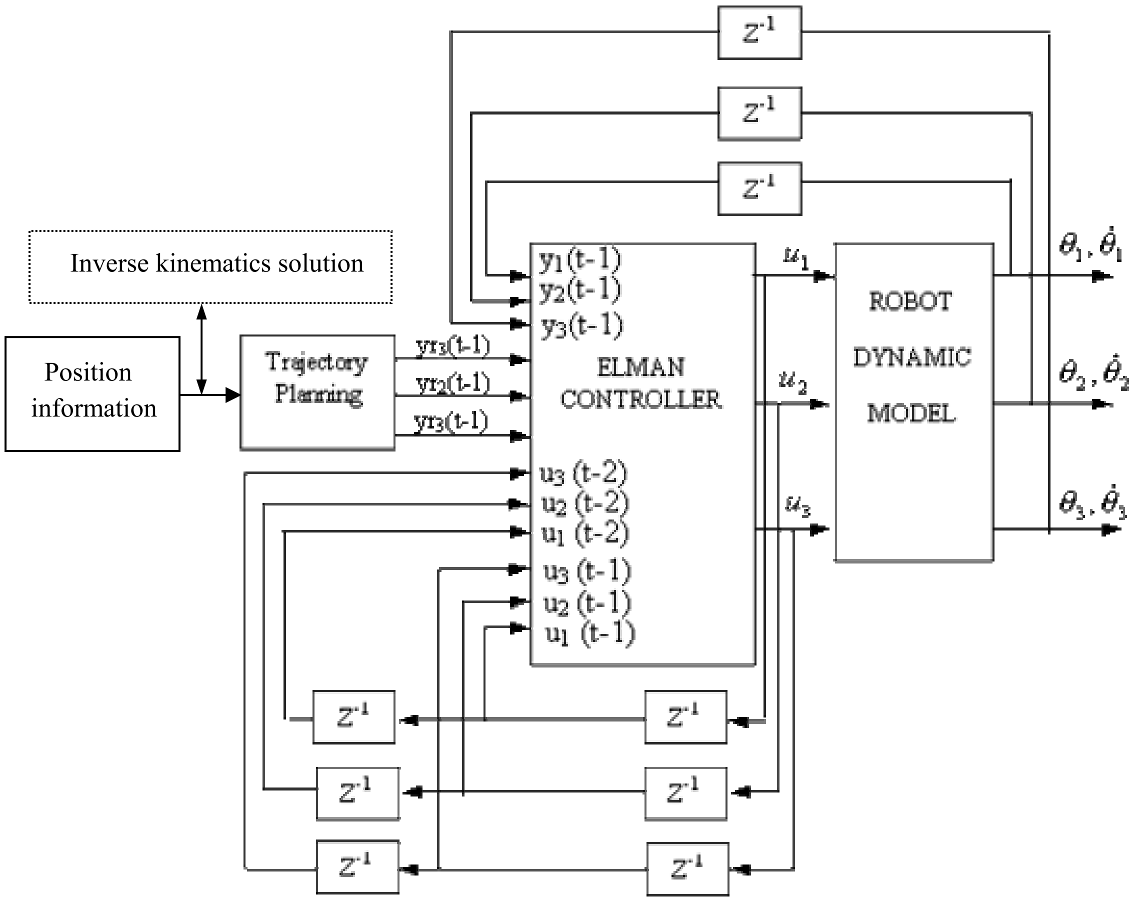


Figure 3. The block diagram of the trajectory control implemented by using the designed Elman controller for vision based application

The training process has been completed approximately in 2.250.000 iterations. After the off-line neural network training is finished, the neural network, which works online, is coded with obtained synaptic weights as seen in Figure 3. The neural network includes 18 neurons in the hidden layer, and it has been tried to obtain the neural network controller with the least number of perceptron in the hidden layer. The sigmoid activation function is used as seen in the Figure 4.

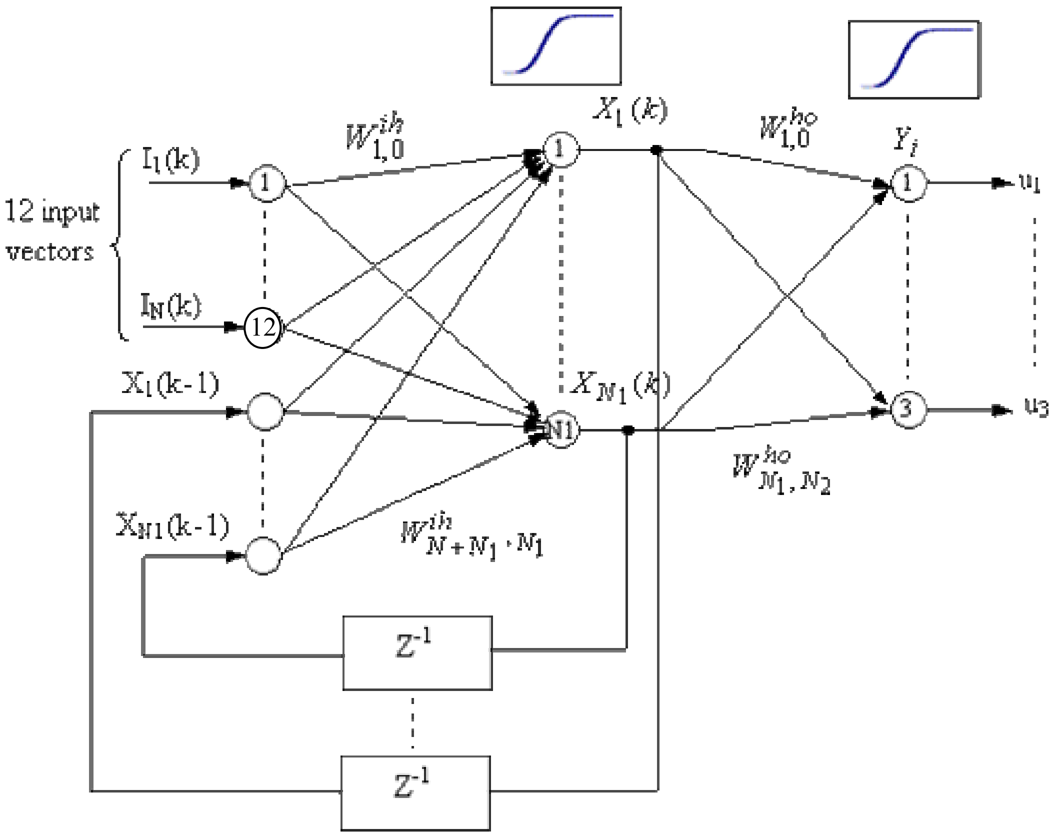
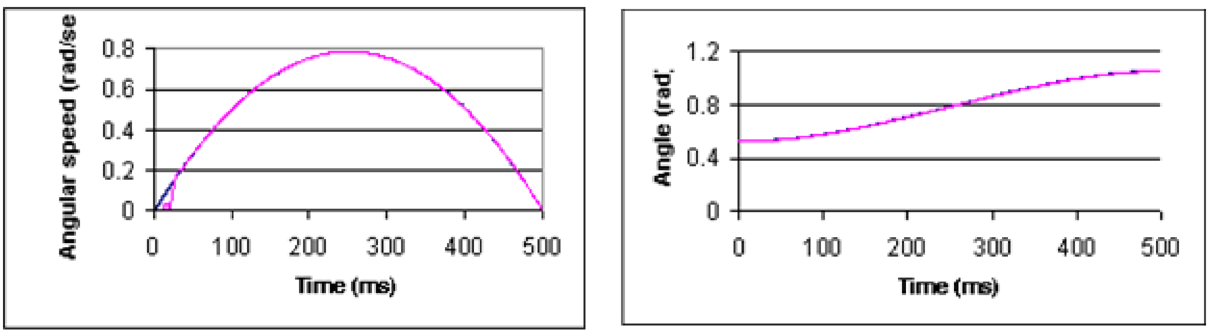


Figure 4. The Elman neural network topology used in the study

5. SIMULATION RESULTS AND DISCUSSION

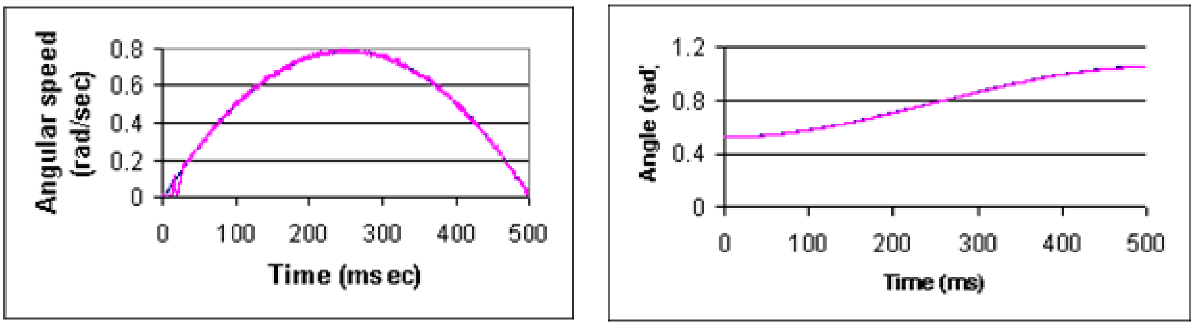
In the simulation studies, it is observed that the training process is too much important for the accuracy of the obtained Elman controller. Generalization is also made through the reproduction of a situation that was absent from the training set. On the other hand, the system is more successful for the trajectory, which is in training set than one, which is not in training set. In Figure 5, the obtained results from GPC algorithm are given. In Figure 6, the result is taken from the trajectory that is not in training set, to demonstrate the generalization of the designed Elman controller.



(a) (b)

Figure 5. a) A sample reference and obtained speed curve using GPC controller

b) A sample reference and obtained position curve using GPC controller



(a) (b)

Figure 6. a) A sample reference and obtained speed curve using Elman controller
b) A sample reference and obtained position curve using Elman controller

From the viewpoint of simulation time, the performance of generalized predictive control algorithm has taken 650 ms, on the other hand designed Elman controller has taken only 62 ms. The Elman network was obtained with the least number of neurons in the hidden layer. This reduction in process time will reduce the need of hardware equipments.

Furthermore, the illumination is also important due to the reflectance and shadow. The illumination is done using two light sources to obtain image without shadow and reflectance. It is observed that noise reduction removed some of these noises.

6. CONCLUSIONS

From the simulation studies, it was seen that the robotic manipulators could be modelled using artificial neural networks. The usage of the Elman network reduced the process time. Normally, the GPC algorithm takes much process time because of the highly mathematical computations. On the other hand, image-processing algorithms have many matrixes. Therefore, the reduction in process time on the control system using Elman network makes the proposed control algorithm more popular in this study. On the other hand, the need of hardware equipments was reduced for a real time system.

REFERENCES

[1] R. Köker, C. Öz, A. Ferikoğlu, "Development of a Vision Based Object Classification System For an Industrial Robotic Manipulator", 8th IEEE Intern. Conf. on Electr, Circuits and Syst. Malta, 2001.
[2] D.W. Clarke, C. Mohtadi, P.S. Tuffs, "Generalized predictive control – Part I: The Basic Algorithm", Automatica, Vol. 23, pp.137-148, 1987.
[3] D. Soloway, P. Haley, "Neural/Generalized Predictive Control,a Newton-Raphson Implementation", Proc. of The Eleven IEEE Int. Symp. on Intel. Cont., pp. 277-282, 1996.
[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] P.A. Laplante, and A.D. Stoyenko, "Real Time Imaging", IEEE Press, 1996, p.231.
[7] G. J. Awcock, and R. Thomas, "Applied Image Processing", McGraw Hill, Inc., 1996, p. 148.
[8] L. Acosta, G.N. Marichal, J.J. Rodrigo, A. Hamilton, J.A. Mendez, "A robotic system based on neural network controllers", Artificial Intelligence in Engineering, 13 (1999) 393-398.
[9] R. Koker, A. Ferikoğlu, "Model-Based Intelligent Control of a 3-Joint Robotic Manipulator: A Simulation Study Using Artificial Neural Networks", Lecture Notes In Computer Sciences, Vol. 3280, pp. 31-40, 2004.