

# Empirical Control Strategy for Learning Industrial Robot

Today's industrial robot systems intensively include external sensors like cameras used for identification of objects in the working environment of industrial robot. Including cameras in the system of an industrial robot, the control problem of such learning industrial robot is set. Using empirical control strategy based on application of artificial neural networks system, the learning industrial robot can realize adaptive behaviour in the sense of flexible adjustment to changes in the working environment. Unlike natural systems which could learn on the basis of experience, artificial systems are thought to be unable to do so for a long time. However, the concept of empirical control realizes the ability of machine learning on the basis of experience. This paper aims to show that it is possible to realize the empirical control strategy for learning industrial robot using camera and system of artificial neural networks. Results obtained by the system of neural nets have shown that the robot can move the end-effector to the desired location of the object, even in the case where the location differs slightly from the learned patterns.

**Keywords:** Learning industrial robot, Robot vision system, Empirical control strategy, Artificial neural networks

**Zoran Miljković**

Associate Professor

**Bojan Babić**

Professor

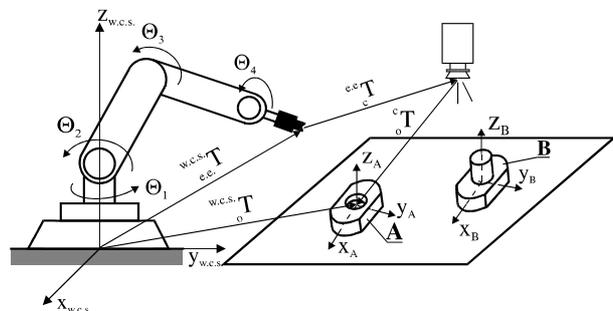
University of Belgrade  
Faculty of Mechanical Engineering

## 1. INTRODUCTION

Robot vision systems are available from major robot vendors that are highly integrated with the robot's programming system. Capability ranges from simple binary image processing to more complex edge and feature based systems capable of handling overlapped parts [1,2]. The feature in common with all of these systems is that they are static. Robots that incorporate vision system are usually designed for task level programming, and such systems are generally hierarchical. The highest level is capable of reasoning about the task. So, visual servoing is no more than the use of vision at the lowest level, with simple image processing to provide reactive or reflexive behaviour. The task in visual servoing is to control the pose (position and orientation) of the robot's end-effector, using information from camera (features), extracted from the image. The camera may be fixed or mounted on the robot's end-effector. If the camera is mounted on the robot's end-effector, there exists a constant relationship between the pose of the camera and the pose of the end-effector. The image of the object is a function of the relative pose between the camera and the object. The distance between the camera and object of interest is referred to as depth or range.

Some relevant poses for experimental system of the learning industrial robot used in this paper are shown in Fig. 1. Experimental robot vision system is created to carry out the assembly task. This experimental system consists of three major components: IBM PENTIUM PC platform equipped with a data acquisition card

(„AVIATOR” - FAST), „Lynxarm” anthropomorphic robot with four degrees of freedom, and „Sony” CCD-TR512E camera. Thus, camera captures a two-dimensional image including objects *A* and *B*. After objects identification and determining of their poses, robot has to fulfill the assembly task, which considers grasping of the object *A* and placing the same object on the object *B*.



**Figure 1. Experimental robot vision system**

External sensors (cameras) are used to obtain the location of objects. The use of vision to acquire the location of objects requires robust recognition algorithms [1,2,4]. The camera captures a two-dimensional image, from which the vision processing software must extract image features. These features are compared to models of the objects to identify the object of interest, and the location of robot's grasp points. Visual servoing approaches based on neural networks, and general learning algorithms, have been used to achieve robot hand-eye coordination [2,11,16]. Camera observes objects and the robot within the workspace, and robot vision system can learn the relationship between robot joint angles and 3D pose of the end-effector. Such systems require training, but the need for complex analytic relationships between image features

Received: April 2007, Accepted: June 2007

Correspondence to: Zoran Miljković and Bojan Babić  
University of Belgrade, Faculty of Mechanical Engineering,  
Kraljice Marije 16, 11120 Belgrade 35, Serbia  
E-mail: zmiljkovic@mas.bg.ac.yu bbabic@mas.bg.ac.yu

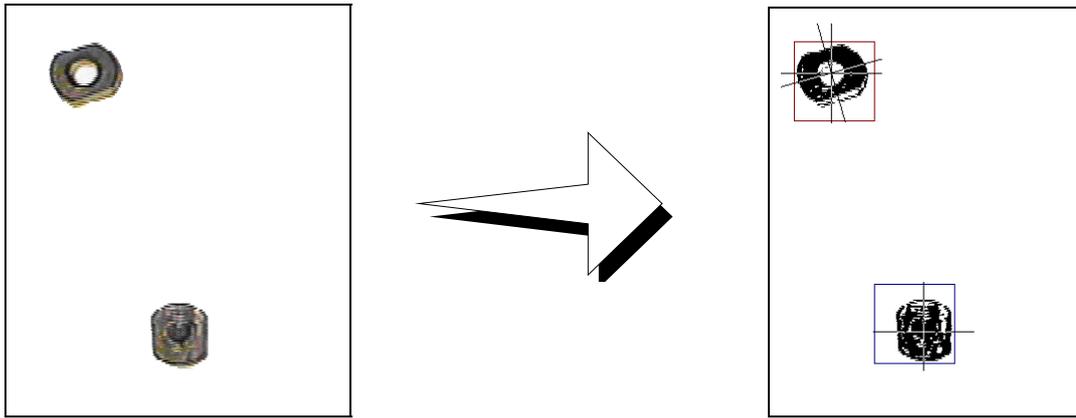


Figure 2. Binary images of objects of interest

and joint angles is eliminated. So, this paper presents empirical control strategy for learning robot based on the prototype artificial neural network system developed to perform pattern recognition, i.e. identification of objects, and to determine the robot joint angles. Artificial neural network system consists of two neural nets: ART-1 which is used for identification of objects, and feedforward which is used to determine the robot joint angles.

## 2. IMAGE PRE-PROCESSING

The image obtained by the camera has a resolution of 640 by 480 pixels and  $2^{24}$  colours, and a buffer store of 300 Kbytes per image. Information about colour is stored as well as intensity information. This image is changed in two steps. The first step considers transformation of the image into grey-scale, and the second one transforms the image into binary image consisting of pixels that are either black or white. To obtain a good binary image, a high contrast between the object and its background is established. The camera is placed so as to point straight down to get the best view of the object.

Today, most systems capture grey-scale information and, if a binary image is required, they generate that image by comparing it pixel by pixel to a threshold value [1,2]. This is an advantage of grey-scale vision as it allows the binary threshold to be easily adjusted. The advantage of this technique is that the threshold level can be set to discriminate between the object in the foreground and the background. Threshold level in our case is usually set to 128 (maximum 255). Placing the threshold in the valley produces the closest match to a binary vision with a good lighting. Based upon this procedure, the filtration of the image is done as follows:

```

For i = 0 To 639
  For j = 0 To 479
    k = BackInGrey(aNiz1(j, i))
    If k >= nThresholdLevel Then
      aNiz2(j, i) = 1
    Else
      aNiz2(j, i) = 0
    End If
  Next j
Next i,

```

where aNiz1 is 2D matrix with the colour, and aNiz2 is 2D matrix with black/white region.

The next step in the image pre-processing procedure defined the subfield of the image, including part *A* and part *B*. The binary images of parts *A* and *B* are obtained in a subfield of 80 by 80 pixels, as shown in Fig. 2. Now, the vision system wants to recognize objects of a known shape in any position and orientation in the image. Also, the vision system wants to obtain their positions and orientations. Features that capture this information are the *centroid*, the *moments*, and *angle*  $\theta$ . To determine these data, the following equations are used:

AREA:

$$A = \sum_{x=l}^n \sum_{y=k}^m B(x, y), \quad B(x, y) = \begin{cases} 1, & aNiz2(x, y) = 1 \\ 0, & aNiz2(x, y) = 0 \end{cases} \quad (1)$$

CENTROID:

$$K_x^C = \frac{1}{A} \cdot \sum_{x=l}^n \sum_{y=k}^m x \cdot B(x, y),$$

$$K_y^C = \frac{1}{A} \cdot \sum_{x=l}^n \sum_{y=k}^m y \cdot B(x, y) \quad (2)$$

MOMENTS OF INERTIA:

$$I_x = \frac{1}{A} \cdot \sum_{x=l}^n \sum_{y=k}^m (y - K_y^C)^2 \cdot B(x, y) \quad (3)$$

$$I_y = \frac{1}{A} \cdot \sum_{x=l}^n \sum_{y=k}^m (x - K_x^C)^2 \cdot B(x, y) \quad (4)$$

$$I_{xy} = \frac{1}{A} \cdot \sum_{x=l}^n \sum_{y=k}^m (x - K_x^C)(y - K_y^C) \cdot B(x, y) \quad (5)$$

ORIENTATION:

$$\Theta = \frac{1}{2} \arctg \frac{2 \cdot I_{xy}}{I_y - I_x} \quad (6)$$

These data are used as input values of the feedforward neural network discussed in the following sections of the paper. Binary images of objects *A* and *B* obtained by image pre-processing present input vectors for two ART-1 neural networks (for each object

separately), which carry out the identification of objects  $A$  and  $B$ .

### 3. IMAGE UNDERSTANDING USING ART-1 SIMULATOR

Artificial Neural Network ART-1 (Adaptive Resonance Theory neural network paradigm ART-1) was introduced by Carpenter and Grossberg (1987) on the basis of the idea of coding and competitive learning [6]. The mechanism of recurrent connections between the competitive and the input layer is used in ART-1 ANN for the retention of old when learning new information. The architecture of ART-1 neural network is given in Fig. 3. Two main ART-1 neural network subsystems are the attentional subsystem and orienting subsystem. The attentional subsystem includes  $F_1$  and  $F_2$  layers which by activation of their neurons (nodes) create ANN associative conditions in a short duration (Short Term Memory - STM) for each input pattern.

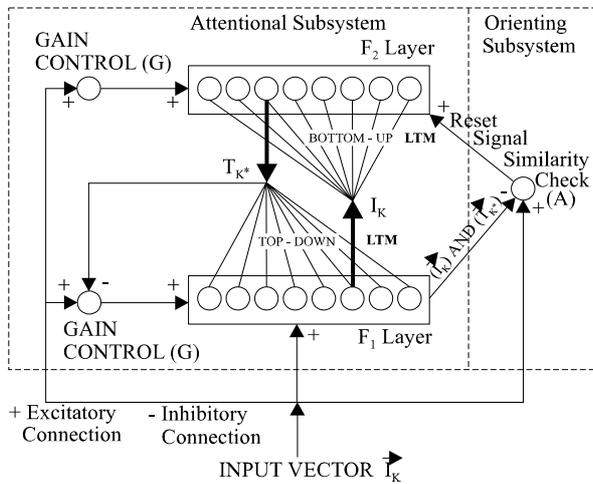


Figure 3. ART-1 System diagram

The weights associated with bottom-up and top-down connections between  $F_1$  and  $F_2$  are called Long Term Memory (LTM). These weights are the encoded information that remains a part of the network for an extended period. The orienting subsystem is needed to determine whether a pattern is familiar and well represented in the LTM or if a new class needs to be created for an unfamiliar pattern.

For each neuron in  $F_1$  layer we have three possible input signals: input pattern ( $I_k$ ), vector gain control signal ( $G$ ) and the pattern created from  $F_2$  layer ( $T_{k^*}$ ) and two output signals. The neuron in the  $F_1$  layer becomes active when at least two, out of three possible, input signals are active ("2/3 rule"). As far as  $F_2$  layer neuron is concerned there exists similar condition of input and output signals.

Input vector  $I_k$  is given in a binary form and the number of neurons in  $F_1$  layer usually coincides with the input pattern dimensions. The connections between  $F_1$  and  $F_2$  layers are given through weight vector  $W_j = [w_{1,j}, w_{2,j}, \dots, w_{p,j}]$ . The  $F_2$  layer also forms a representative or an exemplar pattern

$T_j = [t_{1,j}, t_{2,j}, \dots, t_{p,j}]$ . When an input vector  $I_k = [i_1, i_2, \dots, i_p]$  is presented at the  $F_1$  layer of the ART-1 network, the gain vector  $G$  is initially set to  $G = [1, 1, \dots, 1]$ . The output neurons compete next with one another to respond to the input vector  $I_k$ , and the output neuron  $k^*$  which has the closest weight vector to the input vector identified as:

$$W_{k^*} \cdot I_k = \max_{j \in 1, \dots, q} (W_j \cdot I_k) \quad (7)$$

Dot product is used as the metric to identify the weight vector closest to the input vector. After identifying the output neuron  $k^*$ ,  $T_{k^*}$  is fed back into the input layer, and if any of the  $T_{k^*}$  components is '1', the gain vector  $G$  is then set to  $G = [0, 0, \dots, 0]$ . By the "2/3 rule" the output of the  $F_1$  layer is then  $(I_k)$  AND  $(T_{k^*})$ , a new vector, whose elements are obtained by applying the logical AND on the corresponding elements of the two vectors, giving the following estimate of the similarity between  $T_{k^*}$  and  $I_k$ :

$$\text{similarity} = \frac{\text{number of 1s in } (I_k) \text{ AND } (T_{k^*})}{\text{number of 1s in } I_k} \quad (8)$$

This similarity measure is compared with a prespecified threshold called the vigilance parameter  $\rho$ . If the computed similarity measure is greater than  $\rho$ , then the stored representative pattern associated with the output neuron  $k^*$  is changed to  $(I_k)$  AND  $(T_{k^*})$ . The  $W_{k^*}$  is also changed to:

$$w_{k^*,i} = \frac{L \cdot c_i}{L-1 + \sum_{i=1, \dots, p} c_i} \quad (9)$$

where  $C_i$  is the  $i$ -th component of  $(I_k)$  AND  $(T_{k^*})$ , and  $L$  is the constant (usually set to 2). If the similarity is not greater than the prespecified vigilance parameter, then the output neuron with the next highest  $W_j \cdot I_k$  is selected, and the same procedure is repeated.

So, ART-1 neural network paradigm is selected for identification of objects [8,10,12,14]. This architecture of neural network has the capability of updating its memory with any new input pattern without corrupting the existing information. This is especially important in automated assembly process where the product designs change with time [8,10]. In a typical assembly process the components are usually well defined, and the images can be obtained fairly noise free. It is also necessary to learn new components quickly to reduce set-up time. The fast learning algorithm of ART-1 neural network is used, because applications like robot assembly have crisp data sets and require immediate learning [10,12].

This paper aims to show that it is possible to identify objects using binary input vectors which represent images captured by camera. **ART-1 Simulator** is

developed for this purpose [14]. The software for running the system is developed in Visual Basic [18].

The parameters of the ART-1 structure are in accordance with images and learning procedure. **ART-1 Simulator** sets all parameters for learning process very easily. Learning process of the ART-1 neural network is adopted for identification of objects. This is because the machine vision operations include the ART-1 neural network simulations for pattern recognition. Input vector consisting of 1s and 0s represents the image. In the learning process of the ART-1 net, the number of neurons in the comparison layer, representing input binary vector might be very large. The maximum number of processing nodes required at any instant is at the comparison layer and equals 10.000. The maximum width  $W$  and maximum height  $H$  for any pattern in the image are assigned the value of 500 pixels. In this paper, the resolution of each input binary vector is reduced by a factor of 6.25 from  $W \times H$  and of 250.000 pixels to a grid of 80 by 80 pixels, i.e. of 6400 pixels. The binary matrix of 80 x 80 is then presented to the ART-1 neural network paradigm to determine the object identity.

The parameters of the **ART-1 Simulator** structure used are: 6400 comparison layer neurons (F1 layer), initially 19 recognition layer neurons (F2 layer), and vigilance parameter  $\rho=1.0$ . The bottom-up weights  $Z_{ji}$ , and the top-down weights  $Z_{ij}$  within the network are initialised by other parameters:  $A1=1$ ,  $B1=1.5$ ,  $C1=5$ ,  $D1=0.9$ ,  $L=3$ . During the image understanding phase, **ART-1 Simulator** used for pattern recognition and identification of objects took several different numbers of iterations, considering the most similar input vector, for which ART-1 net is trained, in accordance with required vigilance  $\rho$ , as shown in Fig. 4.

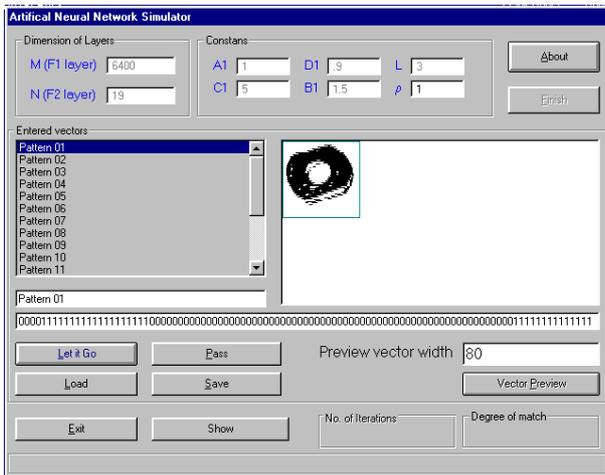


Figure 4. Working environment of the ART-1 Simulator

#### 4. EMPIRICAL CONTROL STRATEGY

*Learning robots* [7] are able to carry out adaptive behaviour based on experience (in a given environment), without the man-operator to take part in it, above all thanking to *empirical control* [8]. This statement has to be verified, which means a scientific challenge presented in this paper. However, the scientific goal is connected to carrying out the ability of „*empirical machine-robot*” to learn, i.e. to be able to

develop behaviour on the basis of its own experience. The ability of „*empirical machine*” is in generating such autonomous behaviour which allows searching for special relations within its own environment as well as its implementation into its future behaviour.

*Empirical control strategy* objectively has to realize three steps in order to accomplish necessary behaviour using its own ability of machine learning, and they are:

- STEP 1:* To produce certain behaviour under certain conditions.
- STEP 2:* To measure (by testing) whether that behaviour is carried out.
- STEP 3:* To produce the behaviour that has the highest probability of successful realization of the task according to given conditions.

On the basis of these three steps, the *empirical control algorithm* (Alg. 1), defined by four rules, is realized for learning industrial robot. These four simple general rules of *empirical control algorithm* create the growth, evolution, i.e. successful development of all *empirical systems*, including the learning industrial robot.

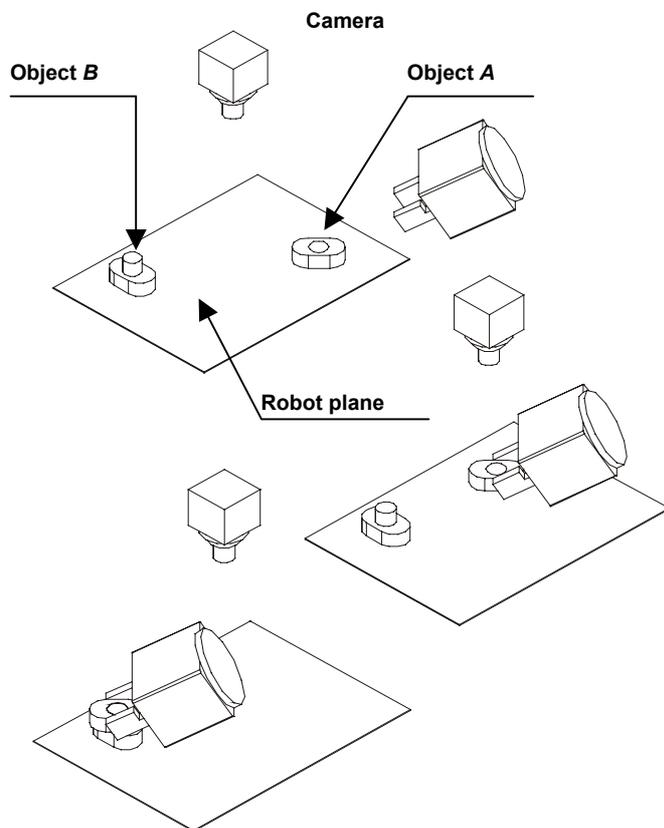
However, within conventional systems of *visual servoing* of industrial robot, the application of *empirical control algorithm* is complex, because it is already complicated enough to impose relative position and orientation of robot end-effector related to the object in the environment. The concept of hierarchical intelligent control [13,15,16,17] on the basis of empirical control algorithm presented in this paper excludes particular nonlinear transformations, because this robot control strategy is based on the backpropagation (BP) learning ability of the artificial neural network (ANN).

Alg. 1 *Empirical control algorithm*

- Rule1:* The empirical control system must select the right output (after expected delay) having the highest level of reliability, according to its memory, for a specific given input.
- Rule2:* If the output selected can be carried out, memorized level of reliability of that output for that given input must be increased, so that the probability of later successful selecting that output for that input has the same increasing trend.
- Rule3:* If the output selected for a specific given input cannot be fulfilled (because it is inhibited, restricted, or something similar interfered with by the environment or some other outside influence such as a „teacher-trainer” in training, or by some internal signals, actuators, or structure of its own), the level of reliability which is memorized must be decreased, so that the probability of the controller later successful selecting that output for that input is increased.
- Rule4:* If some other new output is carried out, memorized level of reliability of that output for that given input must be increased, so the probability of later successful selecting that output for that given input is increased.

Within the set up control strategy, position and orientation of the object in relation to the robot world coordinate system is not known. However, through the robot learning the desired relative pose of the robot end-effector in relation to the object is well known (Fig. 5). Relations between the object data obtained from the recognition system and robot joint angles for the desired pose of the robot end-effector are extremely nonlinear. Developed empirical control system is self-organized in solving this nonlinearity in such a way where it uses the abilities of artificial neural networks (ART-1 and BP nets) which can overcome the problem of nonlinear correlations through the learning process [5]. So, the realized intelligent control system directly integrates visual data information of the object into servo-control system of the robot. The described empirical control algorithm supports hierarchical intelligent robot control through development of artificial neural network system (ANNS) for object recognition as well as for sensor-motor coordination in approaching to the recognized object and its manipulation. Fig. 5 shows the robotic assembly process with object *A* and *B*, i.e. manipulation of object *A*. The robot usually carries out this process in three phases:

- Phase 1:* Robot gripper approach to the recognized object *A*.
- Phase 2:* Grasping the object *A*.
- Phase 3:* Moving the object *A* in order to solve assembly task.



Learning industrial robot must carry out the necessary relative pose of the gripper in relation to the object *A*, which can be in a totally arbitrary position and orientation. This means that if the robot end-effector is in desired relative pose in relation to the object *A* after phase 1, phase 2 and phase 3 of object manipulation, the assembly process will be done easily and successfully, because small position errors can be tolerated and sometimes compensated for by using the system with adaptive behaviour. Owing to visual feedback and machine learning on the basis of artificial neural network system, robot end-effector can take the final pose in relation to the recognized object *A*, i.e. *B*. According to this observation, the experimental system is realized which confirms this statement for the physical model of anthropomorphic robot called *Don Kihot* [16,17] (Fig. 5).

#### 4.1 Empirical control strategy based on feedforward neural network

In situations where the structure of the robot and the geometrical features of the object are exactly known, the position and orientation of the end-effector and the object can be calculated by a geometric method. However, the accuracy of the geometric method largely depends upon camera and image memory resolutions and lens linearity, but also upon camera system parameters, such as focal length and image center offsets [2,11]. In the established approach presented in this paper, the amount of movement of the

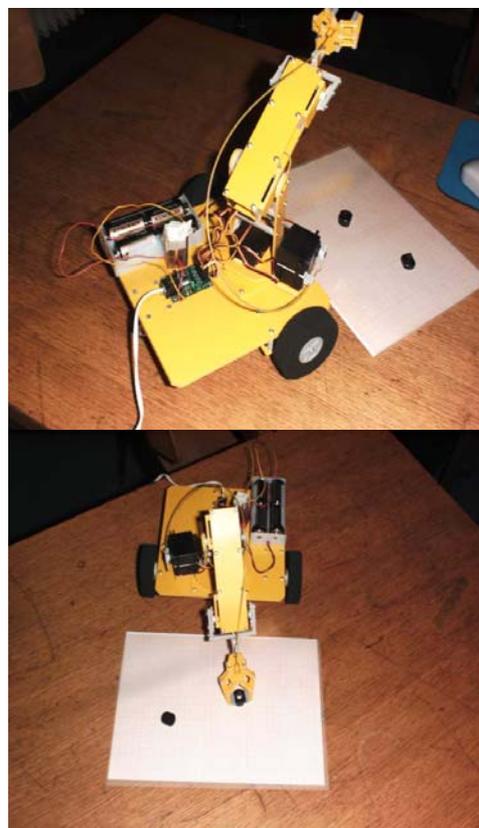


Figure 5. Relative pose of the robot end-effector in relation to the object in order to solve robotic assembly task

robot is determined using image data and system of neural networks. The proposed and described empirical control strategy is based on two mechanisms, as shown in Fig. 6. One is feedforward controller, and another is a neuro-vision feedback controller based on the feedforward neural network.

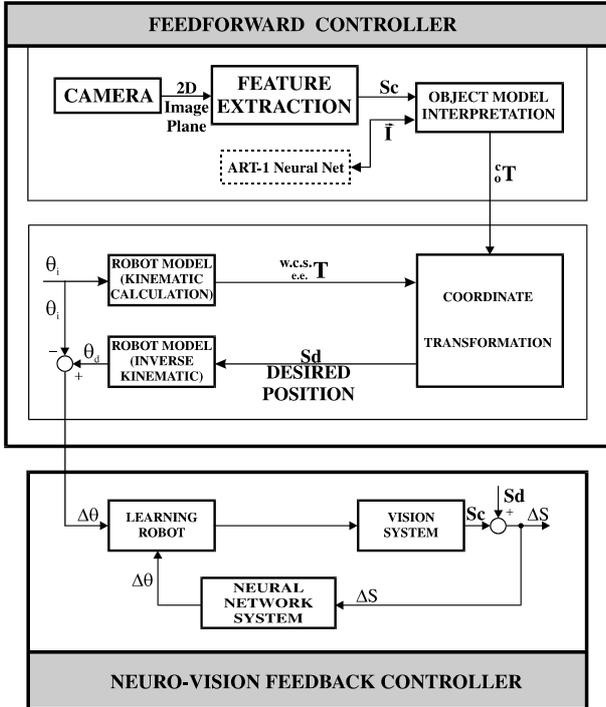


Figure 6. Neuro-vision control block diagram

*Feedforward controller* is based on the camera system, robot model, and object model. The camera system is used as a recognition observer. The relative position between the robot's end-effector and object in its environment (Fig. 5) can be decomposed into some non-linear transformations.

The co-ordinate transformations are needed in order to determine the control inputs to the joints angles from visual data. Thus, the transformation matrix from the object frame to the camera frame is represented in terms of the homogeneous co-ordinate  ${}^c_o T$ , and the position and orientation of the end-effector can be represented by the homogeneous co-ordinate  ${}^{w.c.s.}_{e.e.} T$ . The position and orientation of the object with respect to the world co-ordinate system (w.c.s.) is calculated by the homogeneous co-ordinate  ${}^{w.c.s.}_o T$ , as follows:

$${}^{w.c.s.}_o T = {}^{w.c.s.}_{e.e.} T \cdot {}^{e.e.}_c T \cdot {}^c_o T \quad (10)$$

where  ${}^{e.e.}_c T$  is the transformation matrix from the camera co-ordinate system to the end-effector.

The desired position and orientation of the end-effector with respect to the world co-ordinate system is calculated as follows:

$${}^{w.c.s.}_{e.e.} T = {}^{w.c.s.}_o T \cdot {}^o_o T^{-1} \cdot {}^{e.e.}_c T^{-1} \quad (11)$$

After all of these co-ordinate transformations, inverse kinematics is solved to determine the joint

angles  $\Theta_i$  of the robot from the desired position and the desired orientation of the robot's end-effector. However, control  $\Delta\Theta$  to the robot generated by computing the difference between the desired joint angles  $\Theta_d$  and the current joint angles  $\Theta_{cur}$ . In the present paper, a control strategy is proposed to control the robot without any coordinate transformation, which needs a lot of calculation. So, the control strategy for our robot is based on learning capabilities of the feedforward neural net.

*Feedforward neural net* is developed to determine the robot joint angles. This neural net has a capability to learn to map a set of input patterns (location of object) in a set of output patterns (robot joint angles). The learning algorithm used in this non-linear mapping is backpropagation [3]. The learning process is off-line. A neural network learns the area in the neighborhood of the object for adjusting object location misalignment. The non-linear relation between the image data and the control signals for the changes in the joint angles is learned by feedforward neural net. Developed **BPnet**-software [16,17], based on *backpropagation learning algorithm* [3], is used as a neural simulator. Simulation program package **BPnet** uses a gradient search technique to minimize error function, called generalized delta rule [5]. **BPnet**-software is developed in Visual Basic [18], and its working environment is shown in Fig. 7.

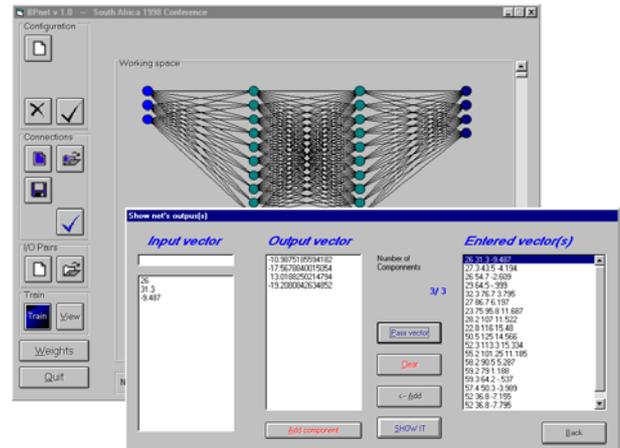


Figure 7. Working environment of the BPnet

## 4.2 BPnet-software and experimental results

Experimental results for the anthropomorphic robot called *Don Kihot* (Fig. 5), with four degrees of freedom, are shown in this section. Two four-layered feedforward neural networks (3x10x10x4) for part *A*, and (2x10x10x4) for part *B*, are used in simulation. One neural net maps the three inputs characterizing the position and orientation of the object *A* onto the four outputs, which are the robot joint angles used for grasping of the object *A*. Another neural net maps the two inputs characterizing the position of the object *B* onto the four outputs, which are the robot joint angles used for execution of assembly process by placing object *A* on the object *B*. Positions and orientations of the object *A*, and positions of the object *B* are given in

Table 1.

	Object A			Object B	
	$K_x^C$	$K_y^C$	$\theta$	$K_x^C$	$K_y^C$
1	26	31.3	-9.5	53.7	119.5
2	27.3	43.5	-4.2	49.5	122
3	29	64.5	-1	52.5	109
...					
18	52	36.8	-8	28.2	76.6
19	48.7	20.5	-12.2	21.6	56

Table 2.

	Object A				Object B			
	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$
1	-11	-18	13	-20	12.5	-22.5	3	-19
2	-9	-17	14	-20	14	-21	7.5	-15
3	-2	-17	12.5	-19.5	10	-23	3.5	-16.5
...								
18	-10	-23.5	11.5	-4.5	0	-11	16	-20
19	-13.5	-23	13	-3.5	-6	-7	20	-19

Table 3.

	Object A								Object B							
	Desired values				Final values				Desired values				Final values			
	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$
1	-11	-18	13	-20	-11	<u>-17.5</u>	13	<u>-19</u>	12.5	-22.5	3	-19	<u>13</u>	<u>-23</u>	3	<u>-18.5</u>
2	-9	-17	14	-20	<u>-8.5</u>	-17	14	<u>-19.5</u>	14	-21	7.5	-15	<u>13</u>	-21	17.5	<u>-15.5</u>
3	-2	-17	12.5	-19.5	<u>-3</u>	<u>-17.5</u>	<u>13</u>	-19.5	10	-23	3.5	-16.5	<u>10.5</u>	<u>-22.5</u>	<u>2.5</u>	-16.5
...																
18	-10	-23.5	11.5	-4.5	-10	-23.5	<u>12</u>	-4.5	0	-11	16	-20	<u>-0.5</u>	<u>-10.5</u>	<u>16.5</u>	-20
19	-13.5	-23	13	-3.5	<u>-13</u>	-23	<u>12.5</u>	<u>-4</u>	-6	-7	20	-19	-6	<u>-8</u>	20	-19

Table 4.

	Object A			Object B	
	$K_x^C$	$K_y^C$	$\theta$	$K_x^C$	$K_y^C$
1	43	94	12.5	21.6	56
2	45	56	-5	52.5	109
3	44	72	0	57.9	106.9

Table 5.

	Object A								Object B							
	Expected values				Final values				Expected values				Final values			
	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$	$\Theta_1$	$\Theta_2$	$\Theta_3$	$\Theta_4$
1	6	-18	15	-10.5	7	-21.5	10.5	-15	-6	-7	20	-19	-6	-8.5	20	-19
2	-5.5	-19.5	13	-10.5	-7	-23	12.5	-7	10	-23	3.5	-16.5	10.5	-23.5	2.5	-16.5
3	0	-19.5	13	-10.5	-1.5	-23.5	8	-14	9.5	-25	0	-16	9	-24.5	1	-16

Table 1 (after using image pre-processing procedure, described in section 2). Desired values of the robot joint angles for successfully assembly process with objects A and B are given in Table 2.

Experimental results obtained by **BPnet**-software, present experimental results for learned patterns. After 20000 iterations in the learning process, the final errors obtained by moving the robot using four-layered neural network are about  $1^\circ$ . The accuracy of this feedforward neural net is 99.6%. The final output vectors are given in Table 3 (robot joint angles which are different from desired ones are marked and underlined). It is observed that the feedforward neural network lets the robot approach the object. The end-effector can move to the neighbourhood of the desired position and orientation of objects A and B.

For experimental process with not-learned patterns, the object location is slightly changed (Table 4). Several experimental results for not-learned patterns are given in this paper too (Table 5). The final errors in that case are larger ( $5^\circ$ ), compared with the ones for the learned patterns. Thus, maximum absolute error for simulation process with not-learned patterns is  $5^\circ$ , and

relative error is **5.5%**. However, the robot's end-effector can move to the neighbourhood of the desired not-learned patterns, and to execute the assembly task too. Because the structure of the feedforward net is suited to parallel processing, the execution time using **BPnet** -software is very fast.

## 5. CONCLUSION

In the present paper, the control strategy of a learning industrial robot with visual sensor is described. The control strategy is based on the empirical control algorithm, as well as on the system of two artificial neural networks: ART-1, and feedforward. The artificial neural network system organizes itself for a robot configuration through a learning process. Identification of objects is based on ART-1 neural network. Non-linear mapping between the image data of objects and the control signals for the changes in the joint angles is learned by feedforward neural network. The proposed empirical control strategy is effective because the generalization ability of the neural networks assures control robustness and adaptability in the event of slightly changed object location.

## REFERENCES

- [1] Milutinovic, D.S., Milacic, V.R.: A Model-Based Vision System Using a Small Computer, *Annals of the CIRP*, Vol.36/1, pp.327-330, 1987.
- [2] Corke, P.I.: Visual Control of Robot Manipulators - a review, *Visual Servoing World Scientific*, pp.1-31, 1993.
- [3] Werbos, P.: *Beyond regression: New tools for prediction and analysis in the behavioral sciences*, Ph.D. dissertation, Committee on Applied Mathematics, Harvard University, Cambridge, USA, 1974.
- [4] McKerrow, P.J.: *Introduction to Robotics*, Addison-Wesley Publishing Company, 1991.
- [5] Freeman, J.A. and Skapura, D.M.: *Neural Networks, Algorithms, Applications and Programming Techniques*, Addison-Wesley Pub.Co., 1992.
- [6] Carpenter, G.A, Grossberg, S.: A Massively Parallel Architecture for a Self-Organizing Neural Pattern Recognition Machine, *Computer Vision, Graphics and Image Processing*, 37, 54-115, Academic Press, Inc., 1987.
- [7] Walter Van de Velde (editor): *Toward Learning Robots*, MIT Press, Special Issues of Robotics and Autonomous Systems, 1993.
- [8] Brown, R.A.: *Machines That Learn*, Oxford University Press, 1994.
- [9] Dagli, C.H.: Artificial Neural Networks in Intelligent Manufacturing, *Proceedings of the 12<sup>th</sup> International Conference on Production Research*, Lappeenanta, pp.127-134, Finland, August 1993.
- [10] Dagli, C.H.: *Artificial Neural Networks for Intelligent Manufacturing*, Chapman & Hall, 1994.
- [11] Hashimoto, H., Kubota, T., Sato, M., Harashima, F.: Visual Control of Robotic Manipulator Based on Neural Networks, *IEEE Transactions on Industrial Electronics*, Vol. 39, No. 6, pp. 490-496, 1992.
- [12] Miljkovic, Z.: Application of ART-1 Neural Network for Pattern Recognition in Robotics (invited paper), *Proceedings of the International A.M.S.E. Conference: Communications, Signals and Systems-CSS96*, Vol.1., pp. 235-238, Brno, Czech Rep., 1996.
- [13] Miljkovic, Z.: Hierarchical Intelligent Robot Control Based on Artificial Neural Network System, *Journal Mathematical Modelling and Scientific Computing (ISSN 1067-0688)*, Vol.8 No. 1-2, pp. 331-336, Principia Scientia, Printed in USA, 1997.
- [14] Miljkovic, Z., Lazarevic, I.: ART-1 Simulator for Identification of Objects in Robotics, *Proceedings of the International A.M.S.E. Conference on Contribution of Cognition to Modelling-CCM'98*, pp.5.48-5.51, Lyon-Villeurbanne, France, 1998.
- [15] Miljkovic, Z., Lazarevic, I.: Control Strategy for Learning Industrial Robot Based on Artificial Neural Network System, *Proceedings of the International Conference on Systems, Signals, Control, Computers – SSCC'98*, Vol.3, pp.124-128, Durban-South Africa, September 1998.
- [16] Miljkovic, Z.: Hierarchical Intelligent Control of Learning Robot Using Camera and System of Artificial Neural Networks, *International Journal of Applied Computer Science (ISSN 1507-0360)*, Special Issue: Selected Applications of Artificial Intelligence, Vol.8 No.2, pp.79-97, Poland, 2000.
- [17] Miljkovic, Z.: Empirical Control System for Robots That Learn, *Proceedings of the 1<sup>st</sup> International Conference on Manufacturing Engineering (ICMEN 2002) and EUREKA Partnering Event*, pp. 759-768, Greece, 2002.
- [18] *Visual Basic 5.0*, Enterprise edition, Microsoft Corporation, 1997.

---

### ЕМПИРИЈСКА УПРАВЉАЧКА СТРАТЕГИЈА ЗА ИНДУСТРИЈСКИ РОБОТ КОЈИ УЧИ

Зоран Миљковић, Бојан Бабић

Данашњи системи индустријског робота интензивно укључују спољашње сензоре као што су камере које се користе за идентификацију објеката у радном окружењу индустријског робота. Укључивањем спољашњих сензора-камера проблем управљања индустријским роботом који учи постаје значајно изражен. Коришћењем емпиријске управљачке стратегије, базиране на систему вештачких неуронских мрежа, индустријски робот који учи може да оствари адаптивно понашање у погледу флексибилног прилагођавања променама у радном окружењу. Поред природних система који могу да уче на бази искуства, за вештачке системе се у дужем периоду говорило да то нису у стању да остваре. Овај рад има за циљ да покаже да је могуће остварити емпиријску управљачку стратегију за индустријски робот који учи, коришћењем камере и система вештачких неуронских мрежа. Резултати добијени коришћењем система неуронских мрежа показали су да хватач робота може да дође у захтевани положај у односу на објекат хватања, чак и у случају када је тај положај различит од научених примера.