Adaptive neural network (ANN) for visual servoing: the mimetic approach

Mirjana Bonković, Mojmil Cecić, Vladan Papić

***Abstract*—** In this paper we present a model free hybrid visual servoing system. The "model free" term refers to the system with the unknown kinematics model that has to be estimated on-line, while "hybrid" specifies the visual controller architecture. The proposed system has a conventional Jacobian estimation part necessary for control output generation and it is supplemented with an additional adaptive neural network (ANN). It is shown that ANN could be used to improve the visual servoing performances of the conventional visual servoing controller, as well as to enable the mimetic control of the robot which dynamics differs from the robot which it mimics.

***Keywords*—**Jacobian estimation, Mimetic control, Model-free control, Robotics, Visual servoing

INTRODUCTION

Numerous advances in robotics have been inspired by the biological systems. Necessity for improvements has been recognized due to lack of sensory capabilities in robotic systems which make them unable to cope with the challenges such as unknown and changing workspace, undefined location, calibration errors and so on. Well known facts which claim, that vision is the most powerful sense in humans and that using vision humans manipulate its environment, result with response to this challenge after which the visual servoing was born. It emerges naturally from our own human experience and from observing other living beings which are able to execute complicated tasks thanks to their sometimes primitive visual systems [1]. Visual servoing (VS) is now a mature subject which currently hosts many different research lines such as image processing, computer vision algorithms, real time control, robot modeling, linear and non-linear control theory, etc. VS aims to control a robot through artificial vision in a way as to manipulate the environment, just as humans do. Using the context of technical system vocabulary, VS has been defined as using visual data within the control loop [1], enabling visual-motor (hand-eye) coordination.

On the other side researches from many scientific disciplines are designing artificial neural networks to solve the variety problems in pattern recognition, prediction, optimization, associative memory, and control [2]. As the visual servoing problem tackles almost all of mentioned areas, it seems natural to use the artificial NN for the problem solving. There are numerous examples from research theory and practice which applied the mentioned approach. The drawback of many neural schemes to tackle visual-motor control problem is that of a long training period [2]. In this article we suggest an approach using EMRAN-RBF which is able to learn this visual-motor coordination on-line. Consequently, conventional (model-based or model-free) control of a single robot manipulator has been improved and/or, as it is shown in our simulations it can be used for mimetic visual servoing in which EMRAN-RBF appropriately translate conventional visually servoed robot movements causing the similar behavior of another robot. For the clarity reasons, the next section of the paper briefly described all the important terms used in this paper, specify the state of the art and what our contribution are comparing to the previous approaches. In Section 3 the control algorithms have been presented, while Section 4 give an overview of the simulated system characteristics. Section 5 presents the simulation results, and finaly, Section 6 concludes the paper.

Manuscript Received April 17, 2007;
Revised received October 21, 2007

M. Bonković is with Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Rudjera Boškovića bb, 21000 Split, Croatia; e-mail: mirjana@fesb.hr

M. Cecić is with Faculty of Electrical Engineering, Mechanical Engineering and Naval Architecture, University of Split, Rudjera Boškovića bb, 21000 Split, Croatia; e-mail: mcecic@fesb.hr

V. Papić is with Faculty of Natural Science, Mathematics and Kinesiology, University of Split, Teslina 12, 21000 Split, Croatia, vpapic@pmfst.hr

STATE OF-THE-ART

The main goal of the visual servoing is to move the robot tip (or mobile robot) to a certain pose with respect to particular objects or features in images. Based on the error signal domain, two types of visual servoing system could be defined: image based visual servoing (IBVS) and position based visual servoing [3]. The first one assumes that the error is defined in 3D (task space) coordinates, while IBVS is based on the error which is defined in terms of image features. The specification of an image-based visual servo task involves determining an appropriate error function *f*, that yields f=0 when the task is achieved [4]. Visual servoing problem could be formulated as a nonlinear least squares problem in which the goal function F is define as:

$$F = \frac{1}{2} f(\theta,t)^T f(\theta,t) \quad (1)$$

where, $f(\theta,t)$ is an appropriate function, based on which the error has been calculated and could be expressed as:

$$\Delta f = J(q) * \Delta q \quad (2)$$

In (2) J is the Jacobian matrix, which relates the rate of change in the image space with the rate of change in the task space. Conventional approaches assume that the Jacobian could be identified analytically based on the camera calibration parameters, depth estimation, and the number of features parameters related to number of degrees of freedom the robot has to be controlled [3]. Also, it can be taken into account that visual servoing algorithms have been independent of the hardware types of configuration (robot and camera). Such approach is model free visual servoing [5,6] which we have also partially used in this paper. We formulate the visual servoing problem as a nonlinear least squares problem solved by a quasi-Newton method using Broyden Jacobian estimation. Such system is supplemented with adaptive neural network (ANN) EMRAN-RBF [7] to enable mimicking of the robot movements, as we show in this paper. It is also possible to use the same ANN to improve the accuracy in the task solving (i.e. approaching, tracking). The approach used in this paper slightly differs from the other approaches with neural networks [8,9]. Typical example has been presented in [9] where the hybrid neural control scheme has been proposed for the problem solving. The problem has been viewed as a calibration problem for which the authors propose three ways in which the problems could be solved: model-based, model-free and hybrid approach. The first one is related with combined model of the manipulator and camera that can be used to compute the joint space coordinates given the camera coordinate. The second, model-free approach, assume that a learning paradigm is adapted using on-line data to learn the required mapping. The last one, hybrid approach, treats the model generated in the first approach as an approximate model. The authors [9] are using this model to construct the learning paradigm and save considerable time. Then, the learning paradigm is fine-tuned by choosing selected workspace regions where the error is expected to be pronounced, thereby improving the accuracy of the model-based approach to that of the model-free approach. In this paper model-free approach assume that the system Jacobian has been defined using numerical estimation technique and after that, the real robot joint values of the first robot have been used as an input to the ANN controller which, using also a visual signal from the camera adopt the input values to the appropriate changes in the system. The paper also shows that the same NN could be used to improve the visual servoing tasks of the conventional controller and explains under which circumstances it can be useful.

III THE CONTROL SCHEME

In this paper we are interested in mimetic robot visual control in a fixed camera configuration. Fig.1. shows the structure of the visual servo systems used in this paper. Here, so called image-based visual servoing is considered, in which the error signal that is measured directly in the image, is mapped to the robot actuators' command input. Visual controller has two separated parts. The first one is the conventional visual servo controller and the second one is adaptive NN. In the remainder of this section we briefly present the main characteristics of those two most important parts of the system.

A. *Conventional visual controller*

In our earlier paper [10], as well as in [11,12], the control law for conventional visual controller has been developed minutely. Here, for the clarity reasons, we reply that the visual controller is constructed in order to determine the joint velocities $\dot{q}$ as:

$$\dot{q} = J^{+} K e \quad (3)$$

where J^{+}, K, and e are the pseudo inverse of the Jacobian matrix J that relates joint coordinates with image features, control gain, and the error signal that is obtained by comparing the desired and current image feature parameters, respectively.

The relation between joint coordinates and image features is given by (2). The same relation could be rewritten using derivatives (4)

$$\dot{f}_1 = J \dot{q}_1 \quad (4)$$

where J is a compound of a robot and image Jacobian. If the expression (3) is multiplied by J then we get

$$J \dot{q}_1 = J J^{+} K e_1 \quad (5)$$

that after rearrangement finally yields decoupled closed loop dynamics of first order (6).

$$\dot{f}_1 + K f_1 = K f_1^{d} \quad (6)$$

However, the compound Jacobian J depends on the system calibration parameters that are hard to obtain accurately in practical applications. In the proposed visual servoing scheme, the Jacobian J is obtained by the estimation process. Various estimation scheme have been exhaustively studied [11,12] and have shown various degrees of successfulness in performing various tasks. One of the standard schemes is algorithm based on the Broyden estimation technique which can be used for on-line estimation of the Jacobian matrix.

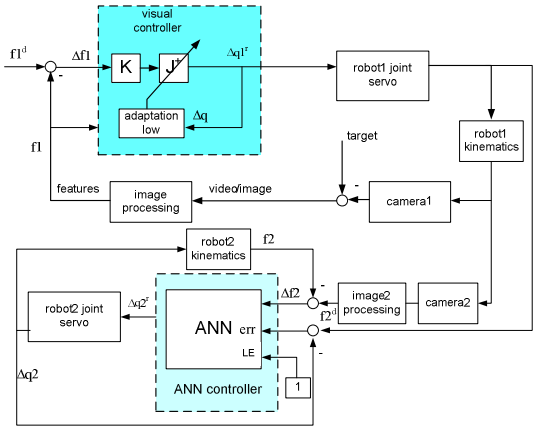


Fig. 1. Mimetic visual servoing block diagram.

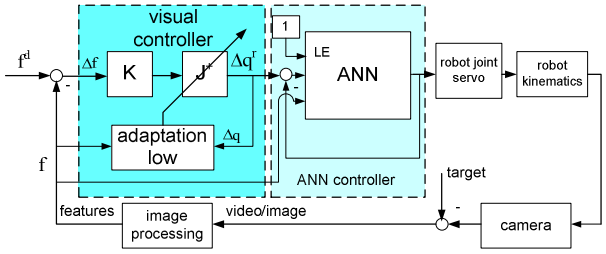


Fig. 2. Single robot visual servoing block diagram

Consequently, the update equation of its estimate $\hat{J}$ is given by (7),

$$\hat{J} = J + \eta\left(\Delta f_1 - J\Delta q_1\right)\Delta q_1^{\,T}\left(\Delta q_1^{\,T}\Delta q_1\right)^{-1} \tag{7}$$

where the adaptation constant η is introduced in order to maintain the convergence overcoming the noise problems of the Broyden method [13, 14]. In this paper, we propose to use the additional ANN controller to adapt the visual control task achievement in a way that the robot follows the robot tip of another one or to improve the visual task achievements of a conventional visual controller.

B. ANN for visual mimicking

The ANN we use has been based on the network from library [7]. Differently from existing neural network collections and toolboxes, in this library, a neural network is strictly treated as a dynamic system with its inputs, outputs and states, and the "dynamic" of the approximation process is therefore considered as an essential part of this "system". In addition, emphasis is given to the approximation result rather than to the approximation process. Therefore, it seems reasonably to use the network together with a conventional estimator which approximates the process and after that ANN directs the results to fit into the approximated process. The Neural Network is represented as Simulink block with its inputs and outputs. Inputs to the block are:

- The input vector to the NN
- The error between the real output and the network approximation
- A logic signal that enables/disables the learning.

Outputs from the block are:

- The value of the approximated function for the current point in the input space
- All the «states» of the network, namely the weights and all parameters that change during the learning process.

By means of the supplied interface, the user can easily set the network parameters that usually remain constant within a specific simulation such as, for example, the number of inputs to the network, the learning rate or the sample time.
For visual mimicking purpose we have used EMRAN-RBF neural network. The details are given in [1]. For the clarity reasons, we present here the most important parts. The EMRAN-RBF is a variation of the standard MRAN (Minimal Resource Allocating Network) [15]. The RAN itself emerges in order to avoid the dimensionality problems generated by the standard RBFNs, proposed a sequential learning technique for RBSNs. The RAN network has proven to be suitable for online modelling of non-stationary processes. The RAN learning algorithm proceeds as follows [7]:

At each sampling instant, if all of the following 3 criteria are met one unit is added:

Current estimation error criteria, error must be bigger than a threshold:

$$e(k) = y(k) - \hat{y}(k) \geq E_1 \tag{8}$$

Novelty criteria, the nearest center distance must be bigger than a threshold:

$$\inf_{j=1}^{M}\left\|x(k) - \mu_j(k)\right\| \geq E_2 \tag{9}$$

Windowed mean error criteria, windowed mean error must be bigger than a threshold:

$$\frac{1}{T}\sum_{i=0}^{T}\left[y(k-T+i) - \hat{y}(k-T+i)\right] \geq E_3 \tag{10}$$

This new neuron is initialized with the following center, variance and weight respectively:

$$\mu_{M+1}(k) = x(k) \tag{11}$$

$$\sigma_{M+1} = \lambda \inf_{j=1}^{M} \left\| x(k) - \mu_j(k) \right\| \quad (12)$$

$$w_{M+1}(k) = e(k) = y(k) - \hat{y}(k) \quad (13)$$

where λ is a constant called «overlapping factor».

If one (or more) of the criteria is not satisfied, the vector $\theta(k)$ containing the tuning parameters of the RBF-NN is updated using the following relationship:

$$\theta(k+1) = \theta(k) - \eta \left. \frac{\partial \hat{y}(k)}{\partial \theta(k)} \right|_{(k)} \cdot e(k) \quad (14)$$

where e(k) is the prediction error and η is the learning rate and $\theta(k)$ is the vector of parameters to be updated.

Based on the described characteristics, in EMRAN-RBFNs, the growing and pruning mechanism remains unchanged, while the parameters are updated following a "winner takes all" strategy. In practice, only the parameters of the most activated neuron are updated, while all other are unchanged. This strategy, (named Extended MRAN, or EMRAN) implies a significant reduction of the number of parameters to be updated online, and for this reason it is particularly suitable for online applications. More details on EMRAN-RBF could be found in [7]. We have found it well suited for the visual mimicking purpose when using together with conventional visual servoing controller. Consequently, visual task could be performed remotely using a camera which "see" the robot and a projection of the "robot master" which also visually performs the task, or as simulation shows, the EMRAN-RBF could be used to improve the visual task goals achievements.

IV Simulations

A. The system

The simulated system is presented in Fig.1. and Fig.2. During simulations the task has been performed using 2DOF planar manipulator with two revolute joints and a camera that can provide position information of the robot tip and the target in the robot workplace. The robot direct kinematics is given by the following equations,

$$x = L \begin{bmatrix} \cos(q_1) + \cos(q_1 + q_2) \\ \sin(q_1) + \sin(q_1 + q_2) \end{bmatrix} \quad (15)$$

where q_1, q_2 are the robot joint angles and x is a vector of robot tip coordinates in the Cartesian world coordinate frame (Fig.3). $l_1 = l_2 = L = 0.4m$ is the length of the robot single link. Translation and rotation of the camera frame with respect to the robot world base frame is given by the RPY homogenous transformation matrix Rc (16). It is rotated around y-axis for 135°, and translated for 1.2 m in both, y and z direction respectively.

$$R_C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -0.707 & -0.707 & 1.2 \\ 0 & 0.707 & -0.707 & 1.2 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

A block named “robot servo” in Fig. 2. represents the robot system dynamics which includes motor, current and velocity-loop dynamics for joints. It has been modeled with the first order open loop transfer function as:

$$G(s)=100/(s+100) \quad (17)$$

which means that the velocity-loop is very fast with respect to the sampling interval (T_{camera}). For the mimicking task, we use the robots with the same dynamics, although the algorithm performs well for robots with different dynamics. The input velocity error has been saturated according to robot specification with limit=0.5 rad/s. Visual feedback gain has been set to K=5. The “robot servo” itself represents an open loop system, due the direct feedback from joints has been used as input in visual servo controller (Fig.1.and 2.) for Jacobian update $\hat{J}$ calculation.

B. Simulation results

In this paper, the image processing node generates the target point applied in the visual task definition within the image. When the first robot tip reached the target, the target point was moved to another position in order to

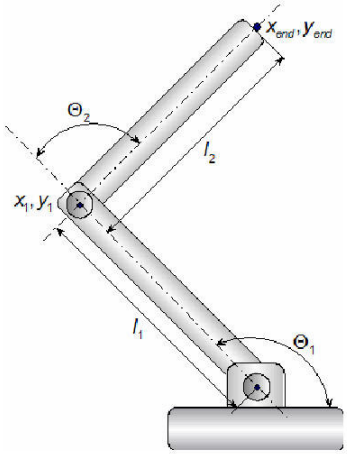


Fig. 3. Planar 2DOF parallel manipulator

provide traveling of the robot tip through the whole robot work plane. The position of the target point determined corners of a characteristic trajectory in the image plane. The projection of the target positions on the robot workplane is

depicted by Fig.4. The initial robot tip position is marked with "0" and the corresponding robot joint angles have the following values: $q_1 = 30^\circ, q_2 = 120^\circ$. The initial target position is marked with "0", and the referent positions are marked "1", "2", "3" and "4". For the first robot task, the target point positions were generated in the following order: "0"-"1"-"2"-"3"-"4"-"1"-"0".The control algorithm has been implemented in SIMULINK model using the appropriate S function. For reference trajectory, marked with points "0"-"1"-"2"-"3"-"4" (Fig. 4.), the rectangle has been chosen with the upper left corner (380,340) and the down right corner (130,150), expressed in the image coordinates. A target start position has been the same as the robot tip start position and it has been moved during simulations with constant speed (measured in pixel/s). The trajectory rectangle has X_{max}=*250 pixel* and Y_{max}= *190pixel* width and height, respectively. The trajectory rectangle has the start point T_{start}=(*x_end0*, *y_end0*)=(246, 236) and *T_camera=0.033 s* has been used in simulations as camera refresh rate (measured in s). Along the curves "1"-"2" and "4"-"3" the y component of the speed has been set to zero. The robot tip starts from the point where target is positioned and marked in Figure 4. as "0". It is worth to notice that all simulations have been performed under the geometrical noise, which is generated through truncation of image pixels value of the robot tip position, which is a normal procedure in IBVS.

We have started our simulations using only one robot performing the described task using Broyden estimation method for conventional visual controller Jacobian estimation with constant $\eta = 0.15$.

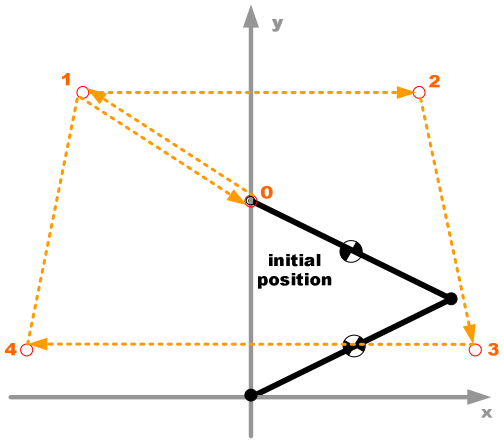
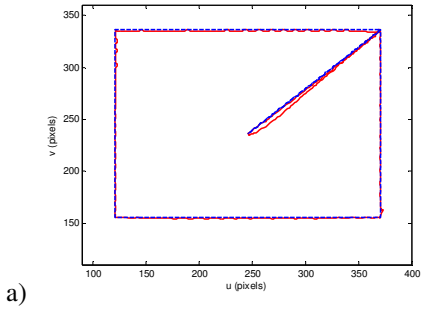
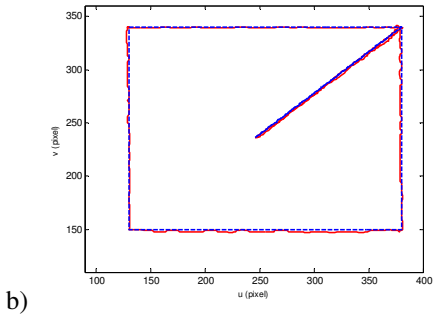


Fig. 4. Target movement

The robot tip traces presented in Fig.5.a. follows the desired curve. Small deviations appear while robot tip tracks the trajectory between points marked as "4"-"1" and "1"-"0". We have study more deeply the mentioned deviations in our previous work and suggest how they can be improved with other Jacobian estimation techniques [12]. In this paper we intentionally choose average performances conventional visual controller to emphasize the improvement effects caused by ANNs. Fig.5.b. shows the same trajectory traces when EMRAN-RBF neural network has been added to conventional visual controller according to scheme presented in Fig.2.



a)



b)

Fig. 5. The task in which the first robot tip has been sequentially moved through the specified points. The image of the reference curve (solid line) and robot tip curve (dashed line) have been presented for:
a) conventional visual controller only
b) added EMRAN-RBF neural network

Neural network corrects the deviations causing the robot tip trajectories almost perfectly follows the desired trajectory. The EMRAN-RBF neural network parameter has been setup through appropriate Simulink block interface (Fig.6). We have found out that the most important parameters are learning rates and sample time, which are [0,175438 0,175438 0,175438] and 0,033, respectively. The rest of the parameters are shown in Fig.6. It is worth to emphasize that improvement of the tracking characteristics is not always possible with an ANN. In the case of more complicated tracking tasks, during which the robot end effector traces more complex curves such as one presented in Fig.9., it is hard to adjust the parameters of the ANN to achieve better results. The more appropriate is to use visual controller with overall better performances for a specific tracking task [11].

After the desired tracking characteristic of the first robot had been achieved, we have proceed with the simulations which include two robots connected with neural network according to scheme in Fig.1. Such scheme enables mimetic behavior of the second robot. In our simulations we have use one camera for visual servoing of the first robot and one camera for the second one. Consequently, the robots could be positioned not in the desired points,

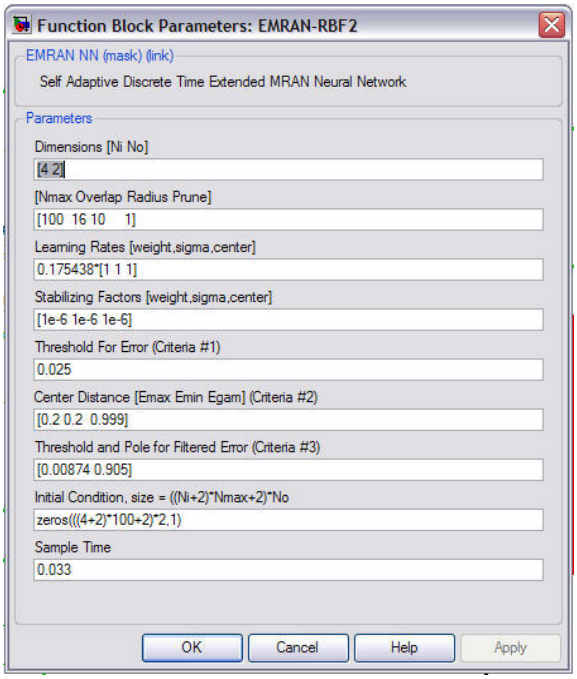


Fig.6. Parameter setup block for standard visual controller improvement

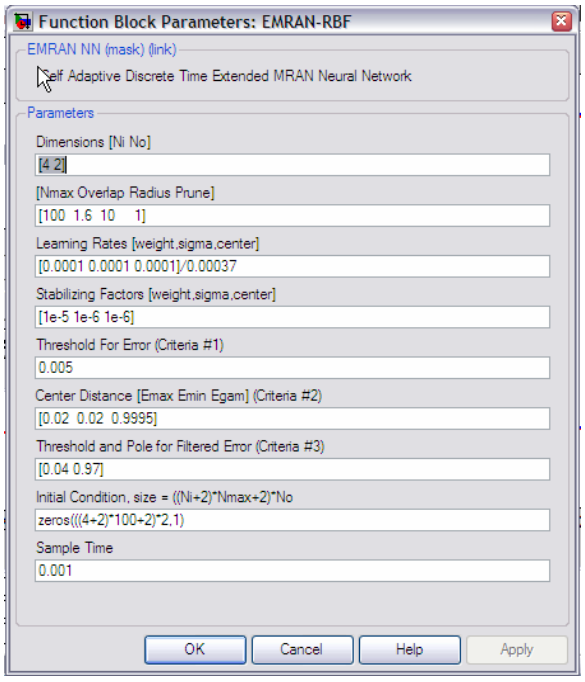


Fig.7. Parameter setup block for mimicking task

but in the points projected from appropriate camera optical center positions.

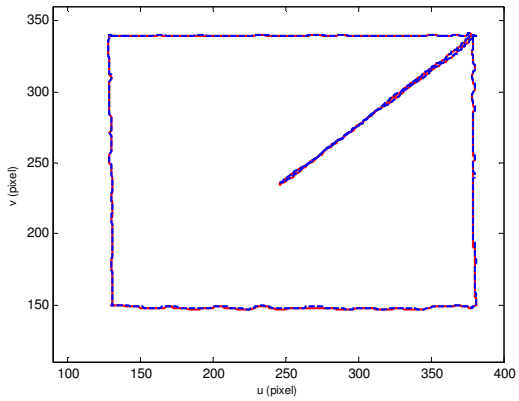


Fig. 8. First robot tip trajectory (solid line-red), which is a referent one for the mimetic movements of the second robot (dashed line-blue).

For correct positioning, at least two cameras have to be used, but under such conditions the real experiment has to be planned more carefully due the both robots' end effector would reach the same space positions. The neural network parameters for mimetic trajectory are [0,27 0,27 0,27] for learning rates and 0,001 for sample time (Fig.7). The results of the robots' end effector movements are presented in Fig. 8. Simulations show that the second robot perfectly follows the tip of the first one as the tracking trajectories overlap perfectly, as Fig.8. indicates. The same conclusion has been confirmed through the second simulation task, which is presented in Fig.9. The tracking trajectory has been more complex spreading the whole robot workspace. The robot end effector starting point has bas been the same as before: $T_{start}=(x\_end0, y\_end0)=(246, 236)$. The robot tip has to pass through the points "0"-"1"-"2"-"3"-"0"-"4"-"5"-"6"-"0"-"7"-"8"-"9"-"0"-"10"-"11"-"12" and finally return back to point marked as "0" again. The simulation results are shown in Fig.10. The both robot tip traces overlapped each other along the whole trajectory, which confirms the conclusion that the EMRAN-RBF neural network could be used for mimicking tasks. It is worth to notice that the same results appear even the robots do not have the same dynamics. Such achievements have given us the idea that the whole system could be used for muscle training under the therapy in which healthy part of the body trains its symmetrical parts forcing the achievement of the same visual goals.

CONCLUSION

The image based visual servoing paradigm represents the challenge in the visual controller design due to numerous unknowns present in the system. Such systems have been usefull in an unstructured envinronment and for well defined industrial tasks as well. Typicaly, such systems are designed as model and calibration free visual servo system in which various numerical methods could be used for Jacobian estimation.

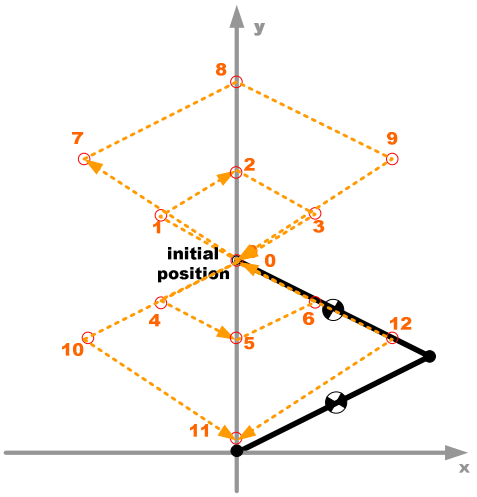


Fig.9. More complex target movement

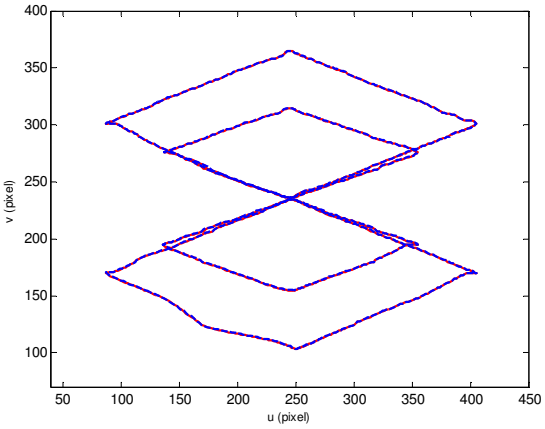


Fig.10. More complex overlapped trajectories of the robots' end effector. The first robot tip trajectory (solid line-red) is a referent one for the mimetic movements of the second robot (dashed line-blue).

In this paper we have shown that average quality numerical solutions could be improved with EMRAN-RBF neural network for the simple and well defined tracking tasks. Moreover, the same neural network is able to transfer the visual servoing goals to the other robot which has the same configuration, but different dynamics. Consequently, the other robot sucessfully mimic the first one in achieving the visual goals. Real application for such type of control cover the broad range of human activities, such as telepresence, and rehabilitation therapy.

ACKNOWLEDGMENT

This research has been partially supported by the project no. 177-0232006-1662, „Computer vision in identification of sport activities kinematics“, funded by the Croatian Ministry of Science, Education and Sports.

REFERENCES

[1] Perez Cisneros M. A., “Intelligent model structures in visual servoing”, *PhD, University of Manchester, Institute for Science and Technology*, 2004.

[2] S. Hutchinson, G. D. Hager, P. Corke, “A Tutorial on Visual Servo Control”, IEEE Trans. On Robotics and Automation, Vol.12, No.5, Oct 1996.

[3] Anil K. Jain, Jianchang Mao, Mohiuddin K. M., “Artificial Neural Networka: A Tutorial“, Computer, March 1996 , pp. 31-44.

[4] P. I. Corke, “Visual control of robots, high performance visual servoing”, *John Wiley & Sons Inc.*, 1996.

[5] M. Jägersand, R. Nelson, “On-line Estimation of Visual-Motor Models using Active Vision”, *Proc. ARPA Image Understanding Workshop 1996.*

[6] J. A. Piepmeier, G. V. McMurray, H. Lipkin “Uncalibrated Dynamic Visual Servoing”, *IEEE Trans. On Robotics and Automation*, Vol.20, No.1, pp. 143-147, February 2004.

[7] Campa G., Fravolini M. L., Napolitano M.: A library of Adaptive netural networks for control purposes, *http://www.mathworks.com/matlabcentral/fileexchange*

[8] Behera L., Kirubanandan N., “A Hybrid Neural Control Scheme for Visual-Motor Coordination”, *IEEE Control Systems*, Vol.19, Num.4., August 1999.

[9] Klobučar R., Pačnik G., Šafarić R., Uncalibrated visual servo control for 2 DOF parallel manipulator with neural network, *Proceedings of 16th Int. Workshop on Robotics in Alpe-Adria-Danube Region-RAAD 2007.*, Ljubljana, June 7-9, 2007.

[10] Bonković M., Hace A., Jezernik K., ”A new method for uncalibrated visual servoing”, *Proc. of AMC*, Istanbul, Turkey, pp.624-629., 2006.

[11] Bonkovic M., Hace A., Bovan S., Jezernik K., “Iterative solution paradigms for uncalibrated visual servoing”, *Proceedings of 16th Int. Workshop on Robotics in Alpe-Adria-Danube Region-RAAD 2007.*, Ljubljana, June 7-9, 2007.

[12] Bonković M., Hace A., Cecić M., Modified Broyden Method for Noise Visual Servoing, *Proc. Of EUROSIM 2007.*, Ljubljana, September, 2007.

[13] C.G. Broyden, "A class of methods for solving nonlinear simultaneous equations", *Mathematics of Computation 19*, pp. 577-593, 1965.

[14] J.J. More, J.A.Trangenstein. ”On the Global Convergence of Broyden’s Method”, Mathematics of Computation, Volume 30, Number 135, July 1976, pp. 523-540.

[15] Lu Y., Sundararajan N., Saratchandran P., “Analysis of Minimal Radial Basis Function Network Algorithm for Real-time identification of nonlinear dynamic systems“, *IEE Proceedings on Control Theory and Application 2000;* Vol.4, no. 147, pp.476.