Adaptive Neuro-Fuzzy Inference System for Kinematics Solutions of Redundant Robots

Mihai Crenganis, Radu Breaz, Gabriel Racz, Octavian Bologa

MIE department, Engineering Faculty

Lucian Blaga University of Sibiu

Sibiu, Romania

mihai.crenganis@ulbsibiu.ro

***Abstract*—This written paper presents aspects concerning the implementation of the Adaptive Neuro-Fuzzy Inference System (ANFIS) in the resolution of a redundant serial robot kinematics. The kinematics solutions are divided into two categories: direct kinematics solutions and inverse kinematics solutions. To be able to control a robot the most important solutions are the ones for the inverse kinematics since one knows the position and the final orientation of the end effector and needs to determine the relative displacement or movements into the robot couplings. To obtain the optimal solutions for the inverse kinematics of a redundant robot the mathematical equations were based onto the redundancy circle method. The ANFIS model is used in order to determine the robot elbow position onto the redundancy circle so the robot will able to avoid different obstacles.**

Index Terms - ANFIS; redundant robot; kinematics;

I. INTRODUCTION

The robots, with their structure and functions are a class of systems that synthesizes elements from a number of top technical fields. In fact, through its abilities a robot imitate or substitute the functions of locomotion, manipulation and intellect of man. A redundant robotic arm with multiple degrees of freedom (DOF) shows a great potential because it gains a vast working space and a high mobility and therefore it can be used in numerous technical fields. Nevertheless by being a redundant robot it can also be used in crowed places. The redundancy allows it to avoid obstacles when is being manipulated. It is obviously, then, that the robot is a highly complex system, described by sophisticated mathematical models and is defined by nonlinear differential systems of equations with variable parameters, including a large number of input/output variables [1], [2].

The basic function of the robot is represented by its movement in space, so the kinematics of the mechanical structure will represent the starting point in the driving and the control of the robot.

With the development of robotics and industrial robots various mechanical systems that mimic living systems and the human body have been developed. The general structure of robots is highly dependent of the utility and the purpose for which they are built. [1] The method proposed for resolving the redundancy problem of the robot presented in this paper uses the fuzzy logic toolbox from MATLAB®. Since this type of robot is redundant and has seven degrees of mobility with one in excess of six possible movements in the 3D Euclidian space we impose additional conditions to solve the system of equations [3].

Firstly the paper describes the structure and the kinematical model and the equations of the robotic arm, then it is highlighted the model of the ANFIS approach for the redundancy solution. Finally the results of the simulations are presented for the numerical analysis and then the conclusions are expressed. The fundamental problems of a robotic arm are the computation of the forward and inverse kinematics, also we can say the mappings between joint space and Cartesian task space. In robot control applications, the inverse kinematics is more important because the trajectories and the path-planning are usually specified in terms of Cartesian task space coordinates [4], [5].

Solving the inverse kinematics (IK) problem is more complex than the forward kinematics (FK) since the inverse kinematics determines a nonlinear system of equations which is more demanding [6], [7].

Many solutions for the redundancy problem have been presented by others, and the most significant are: the generalized inverse Jacobian matrix, the pseudoinverse $J^{\dagger}=J^T(J\cdot J^T)^{-1}$ are widely applied for redundant robots [9], were J represents the Jacobian matrix. The main disadvantage of this method is that pseudoinverse often leads the robot into singularities. Another widely used method for redundancy resolution is the inertia weighted pseudoinverse $JI^{\dagger}=M^{-1}J^T(J\cdot M^{-1}\cdot J^T)^{-1}$, proposed by [5] and this method is based on the fact that the energy consumption can be minimized using the inertia matrix M as the weighting matrix [8], [9], [10].

Moreover, most numerical solutions rely on the calculation of the inverse Jacobian matrix to solve the inverse kinematics problem. In this case because the robot arm is redundant the matrix is not square [n x n], it is of type [m x n]. Therefore, the kinematics redundancy problem is usually resolved by adding enough external constraints in the solution procedure. We are using the ANFIS model to compute the position of the robot elbow and to place it on the redundancy circle in such way to avoid collision with different obstacles [11], [12].

In this case using ANFIS for the solution of complex nonlinear systems is very useful and easy to implement.

However, in some cases, the designed seemed to involve some blind spots in the fuzzy rules.

II. Kinematics Solutions

In Fig. 1 the structural scheme of the seven DOF robotic arm is shown. To resolve the inverse kinematics problem for a 7 DOF manipulator we have modeled a robot manipulator in SolidWorks and then based on this structure we have developed the kinematical equations. The robotic arm was then manufactured and assembled based on the CAD parts and drawings.

If we attach to each element *i*, $(i = 0 \ldots 7)$ of the robotic arm, one fixed coordinate system *ki*, then we can express the homogeneous transfer matrices *Ai* which characterize the relative movements between each element of the mechanic structure [7], [9], [10], [13], [14].

If we know the relative parameters θi and the homogeneous transfer matrix form between two elements, or the homogeneous transfer matrix between the coordinate systems attached to each element, we can determine the total transfer matrix between the system of coordinate k_7 and k_0 [15], [17].

$$H_{07} = A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot A_5 \cdot A_6 \cdot A_7 \tag{1}$$

Where:

A_1 represents the rotation between the base and the first element of the robotic arm or between k_0 and k_1 coordinate systems, with the angle θ_1, A_2 represents the rotation between the first and the second element of the robotic arm or between

k_1 and k_2 coordinate systems, with the angle θ_2, A_3 represents the rotation between the second and the third element of the robotic arm or between k_2 and k_3, with the angle θ_3, A_4 represents the rotation between the third and the forth element of the robotic arm or between k_3 and k_4, with the angle θ_4, A_5 represents the rotation between the forth and the fifth element of the robotic arm or between k_4 and k_5, with the angle θ_5, A_6 ,represents the rotation between the fifth and the sixth element of the robotic arm or between k_5 and k_6, with the angle θ_6, A_7 ,represents the rotation between the sixth and the seventh element of the robotic arm or between k_6 and k_7 with the angle θ_7.

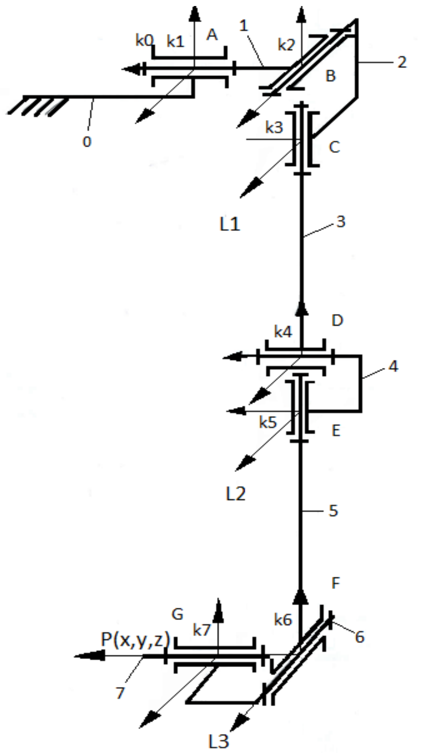

Fig. 1. The kinematic scheme of the robotic arm

We will note the elements of H_{07}, *(ai,j)*. To resolve the inverse kinematics problem we need to start from the fact that we know the position and the orientation of the end effector X, Y, Z, φx, φy, φz, in reference to the fixed coordinate system *k0* and we need to determine the relative positions between robot elements [16], [17], [21].

Therefore we wish to determine the relative parameters θi between elements which represent the rotation in the kinematical couplings. In this case $i = (0 \ \ldots 7)$ [6], [7].

Since this type of robot has a degree of mobility in excess of six possible in the 3D Euclidian space we imposed additional conditions to solve the system of equations. We have considered the angle $\theta 1$ known. In our recent studies this angle was characterized by a fixed value and so the simulations were limited.

The main steps in our research to solve the inverse kinematics problem are:

Creating the transfer matrix which characterizes the end effector position and orientation in reference to a fixed coordinate system k_0, based on the absolute parameters *X, Y, Z, φx, φy, φz*.

The wrist position x_{im}, y_{im}, z_{im} will then be determined. After we have computed the wrist position and orientation we will determine the relative parameters θi. We will consider these notations: *P (X, Y,* and *Z)* is the end effector position in reference to the fixed coordinate system; the *k7* coordinate system is attached to this section. L1 is the arm length, L2 the length of the forearm, L3 the length from *P (X, Y, Z)* to the wrist.

The transfer matrix which expresses the position and orientation of the end effector in reference to a fixed coordinate system k0 is composed. The transfer matrix is denoted H07 and consists of the product of all homogeneous transfer matrices that characterize the end effectors position and orientation in 3D space in reference to a fixed system of coordinates,

$$H_{07} = T_x \cdot T_y \cdot T_z \cdot R_{\varphi x} \cdot R_{\varphi y} \cdot R_{\varphi z} \,. \tag{2}$$

Where: *Tx* represents the translation with *X* dimension along the *OX* axis of the k_0 coordinate system, *Ty* represents the translation with *Y* dimension along the *OY* axis of the *k0* coordinate system, *Tz* represents the translation with *Z* dimension along the *OZ* axis of the *k0* coordinate system, *Rφx* represents the rotation with *φx* degrees around the *OX* axis of the *k0* coordinate system, *Rφy* represents the rotation with *φy* degrees around the *OY* axis of the k_0 coordinate system, *Rφz* represents the rotation with *φz* degrees around the *OZ* axis of the k_0 coordinate system.

The matrix that determines the position of the hand wrist is shown below:

$$H_{im} = H_{07} \cdot (0 \quad -L3 \quad 0 \quad 1)^T . \quad (3)$$

We will note:

$$(a_{14} \cdot a_{24} \cdot a_{34} \cdot a_{44})^T = H_{im} \cdot (0 \quad 0 \quad 0 \quad 1)^T . \quad (4)$$

The distance between the hand wrist position and k0 is:

$$l = \sqrt{a_{14}{}^2 + a_{24}{}^2 + a_{34}{}^2} . \quad (5)$$

The angle θ4 is determined using the expression:

$$\theta_4 = \pi \pm \arccos\left(\frac{L1^2 + L2^2 - p^2}{2 \cdot L1 \cdot L2}\right). \quad (6)$$

To determine the other angles, we have used the redundancy circle method. We know by fact that even if the hand wrist is fixed the elbow joint manages to describe a circular trajectory around the line segment from O to B, like shown in Fig. 2. This is the most efficient method for solving the inverse kinematics problem of a serial and redundant robotic arm.

The next objective is to determine the position and orientation of the elbow joint. To do that we need to use the Roll, Pitch and Yaw angles starting from k0 coordinate system. In this case we have used α, β, θ. The last one is the most important because this angle will be used to resolve the entire inverse kinematics of the robotic arm. To be short, the input data to the inverse kinematics equations are the position and orientation of the end effector *X, Y, Z, Rφx, Rφy, Rφz* and the *θ* angle. The *θ* angle is zero when the OBE triangle is perpendicular onto YOZ plane.

After we have determined the distance from the wrist to the shoulder l, we can compute the area Ar and height h of the OBE triangle, using the Heron's formula. Starting from k0 we can reach the elbow position using some simple homogenous transformations like:

$$He = R_x(\alpha) \cdot R_y(\beta) \cdot T_z(a) \cdot R_z(\theta) \cdot T_y(h) \quad (7)$$

Where:

He, represents the entire homogenous transfer matrix that characterizes the position and orientation of the elbow joint;

α, β, θ, represent the Roll, Pitch and Yaw angles, the orientation of the elbow joint;

a, represents the projection of L1 onto l and it is determined using Pitagora's formula in the OCE triangle;

h, is the height of the OCE triangle and OBE triangle.

From *He* matrix we can determine:

$$\theta 1 = \mathrm{atan2}(He(1,1), He(3,1)). \quad (8)$$

Knowing that:

$$He = A_1 \cdot A_2 \cdot A_3 \cdot A_4 , \quad (9)$$

We have the following system of equations:

$$A_1 \cdot A_2 \cdot A_3 \cdot A_4 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} a_{i,4} \\ 1 \end{pmatrix}. \quad (10)$$

If we multiply to the right with the inverse matrix of A_1 it results the following system of equations:

$$A_2 \cdot A_3 \cdot A_4 \cdot \begin{pmatrix} 0 \\ 1 \end{pmatrix} = \begin{pmatrix} b_{i,4} \\ 1 \end{pmatrix}, \quad (11)$$

Because θ_4 is known it results from the first equation of the system (11):

$$\theta_3 = \arccos\left(\frac{b_{14}}{L2 \cdot \sin(\theta_4)}\right). \quad (12)$$

θ_2 results from the second equation of the system.

To determine the angles θ_5, θ_6, θ_7, we consider the following system:

$$A_5 \cdot A_6 \cdot A_7 = A_4^{-1} \cdot A_3^{-1} \cdot A_2^{-1} \cdot A_1^{-1} \cdot H_{im} \quad (13)$$

Also we will note:

$$A_4^{-1} \cdot A_3^{-1} \cdot A_2^{-1} \cdot A_1^{-1} \cdot H_{im} = \left(m_{i,j}\right) \quad (14)$$

From the equality of the two matrices the following results:

$$\begin{cases} \theta_6 = \arcsin(m_{32}) \\ \theta_7 = -\arctan 2(m_{31}, m_{33}) \\ \theta_5 = -\arctan 2(m_{12}, m_{22}) \end{cases} \quad (15)$$

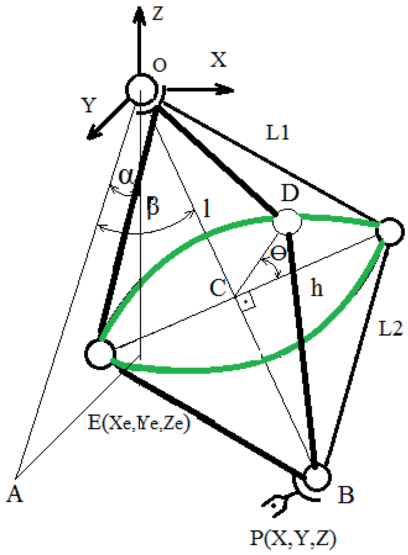

Fig. 2. Redundancy circle representation

III. Testing The Mathematical Equations

To test the mathematical equations presented above and to obtain numerical results for the kinematics analysis, the mathematical equations were written as “m-code” using MATLAB® and then the code was implemented in Simulink®. The implementation was done using an Embedded MATLAB Function block, see Fig. 3. The proposed equations were tested using a virtual 3D model of the robotic arm and the reached positions and orientations were the same as the ones used as input data for the inverse kinematics. The MATLAB Function block allows adding MATLAB functions to Simulink models for deployment to desktop and embedded processors. This capability is useful for coding algorithms that are better stated in the textual language of the MATLAB software than in the graphical language of the Simulink product. From the MATLAB Function block, one can generate readable, efficient, and compact C/C++ code for deployment to desktop and embedded applications [18], [19].

Fig. 4 shows the virtual 3D model of the robotic arm realized in Simulink-SimMechanics, used for visualization of the angle θ and the relative position between the elements of the robotic arm.

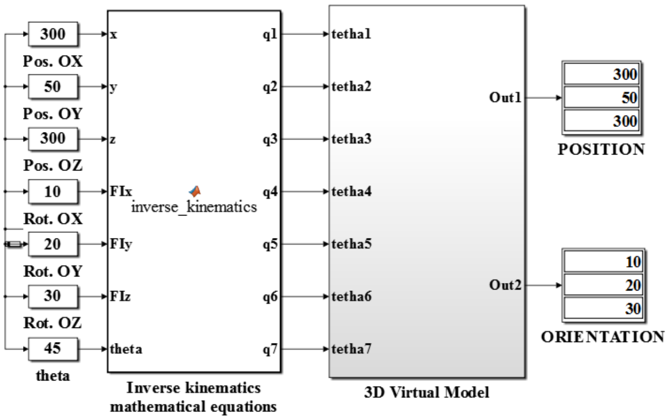

Fig. 3. Block model of the inverse kinematics resolution.

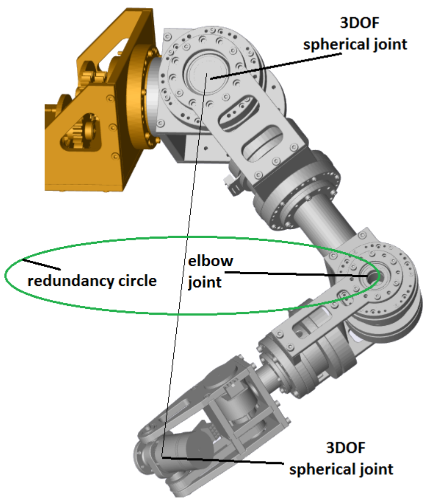

Fig. 4. 3D virtual model of the robot used for visualization.

IV. ANFIS Aproach for Redundancy Resolution

As we have presented above the entire kinematics analysis is based on the redundancy circle method. The redundancy circle method is based on the fact that even if the end effector of the robotic arm has reached a desired position the elbow joint keeps the ability to move. Not only has that but the elbow still described a circular trajectory. Using the Fuzzy logic and ANFIS models we are trying to limit the movement of the elbow and fix it in order to force the robotic arm to adapt a certain position in space and to be able to avoid obstacles or to have a collision free trajectory. To impose a certain position for the robot elbow we need determine the θ angle of the redundancy circle. In this case θ angle must have a known law of variation. This law of variation will be determined using ANFIS. We have used Adaptive Neuro-Fuzzy method because it is conceptually easy to understand and easy to implement. The mathematical concepts behind the model are very simple and have a more intuitive approach without a very high complexity. First of all by using ANFIS we can benefit of its great flexibility. In this case instead of using a steady value of the angle θ, like we have used before in our earlier simulations, this type of control allows us to benefit of a much higher mobility and working space of the robotic arm. ANFIS technique is best applied when we do not have a predetermined model structure based on characteristics of variables in our system. Sometimes we can't create a simple fuzzy logic model because we do not know how the membership functions should look like. The adaptive neuro fuzzy method works similar to neural networks and this technique provides a method for the fuzzy modeling procedure to learn information about a data set [20].

In our research we have tried to test the mathematical equations with a steady vale of the angle θ. We have thought that the robotic arm having 7 DOF and a infinity of solutions any value for the angle θ might be possible. In this case the movements of the robotic arm were limited because the manipulator was reaching into singularities.

After we have tested the mathematical equations of the inverse kinematics with a constant value of the angle θ, Fig. 3. We have noticed that that the most important inputs for the model, if we impose a trajectory for the end effector at a certain distance from base onto the X axis, are the variation of the Y coordinate and Z coordinate of the position of the end effector.

The first step in our research of implementing ANFIS method was to test the working space of the robotic arm. In order to do so we have created a m-file in MATLAB based on the direct kinematics of the robotic arm. The working space is created by varying the relative position between the robotic arm elements. The displacement between two consecutive elements was limited to 180 degrees. The workspace equations are presented in Fig. 5 and Fig. 6 presents the workspace of the robotic arm. After having this collection of input-output data that we would like to use for modeling we can now create the ANFIS scenario.

The next step is to train and test Sugeno type fuzzy systems using the ANFIS Editor GUI from MATLAB. To create a FIS

```
for theta1=-90:10:180
    for theta2=0:10:180
        for theta3=0:10:180
            for theta4=0:10:180
                for theta5=0
                    for theta6=0
                        for theta7=0
%  direct kinematics
H07=A1*A2*A3*A4*A5*A6*A7;

% position of the End Effector
X=[H07(1,4), H07(2,4), H07(3,4)];

% workspace
WS(idx,:)=X(:,:);

% value of the theta1
Z(idx,:)=[theta1];
idx=idx+1;

                        end
                    end
                end
            end
        end
    end
end
% matrix for training [X, Z, theta1]
A_training=[WS(:,1) WS(:,3) Z(:,1)];

% matrix for training [X, Z, theta1]
%A_testing=[WS(:,1) WS(:,3) Z(:,1)];
```

Fig. 5 Workspace equations written in "m-code"

(fuzzy inference system) we must begin by loading a training data set that contains the desired input/output data of the system to be modeled. This data set is obtained from the above mentioned and created workspace [20]. This action adjusts the membership function parameters and displays the error plots. Next the FIS model structure must be specified. We have two inputs and one output. For each input we imposed six membership functions of triangular shape type and the FIS type is Sugeno as mentioned above. In Fig. 7 the ANFIS information are presented and then Fig. 8 shows the ANFIS structure.

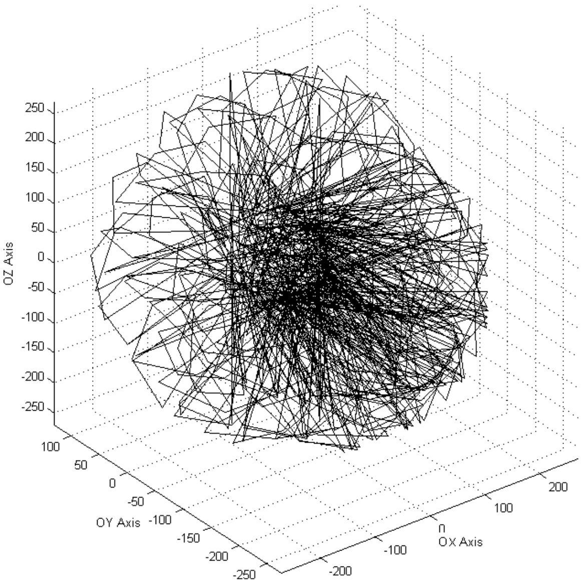

Fig. 6 Workspace of the robotic arm

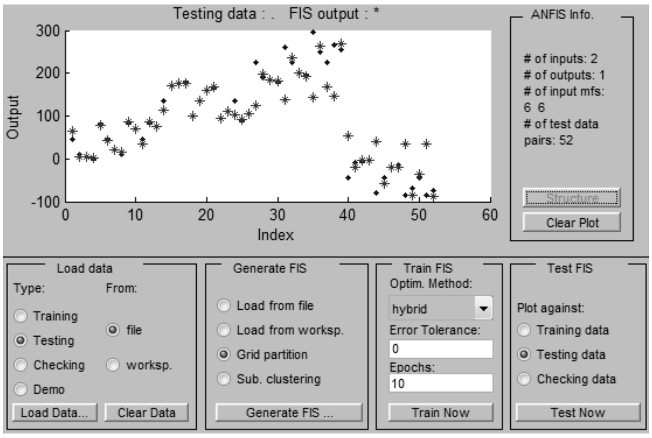

Fig. 7 ANFIS model

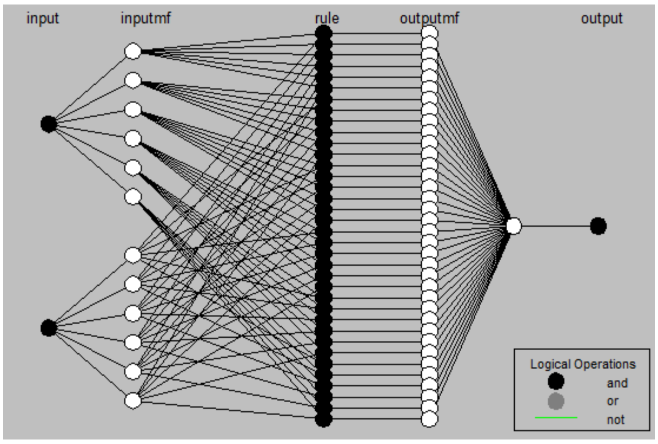

Fig. 8 ANFIS structure

After specifying the FIS structure and the training is complete, we have validated the model using a different set of data from the one used before to train the FIS. This set of data was obtained by running again the workspace equations but with different input values for the theta angles. The FIS model was then exported into MATLAB workspace. To use the created FIS model, a Fuzzy block was used in the Simulink model as shown in Fig. 9.

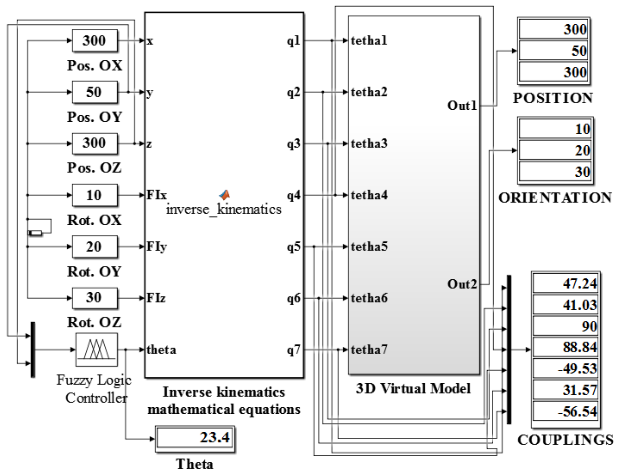

Fig. 9. Block model of the inverse kinematics resolution with ANFIS.

Here the .fis file was imported from workspace. The FIS model can anytime be edited using a simple command in the MATLAB command window and then a fuzzy editor GUI allows you to edit the highest level features of the fuzzy inference system.

In Fig. 10 the Surface Viewer is presented. The Surface Viewer displays a roadmap of the Whole Fuzzy Inference Process. It is based on the fuzzy inference diagram described in the previous section. This represents a very useful tool for modifying and changing the ANFIS Structure.

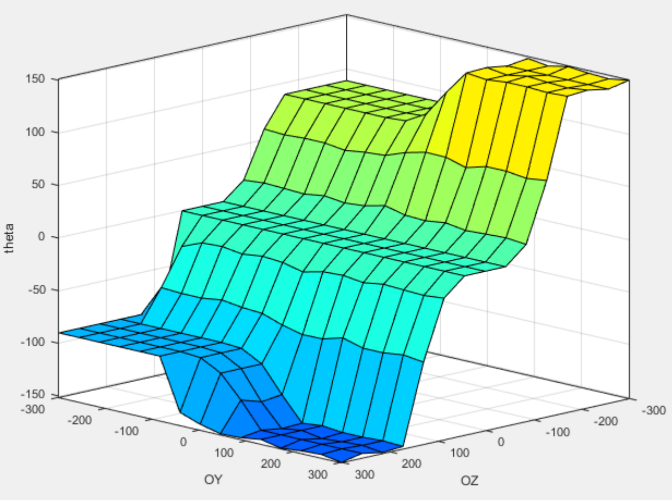

Fig. 10. ANFIS surface viewer.

V. Conclusions

The chosen analytical method for solving inverse kinematics problem was effective, accurate and easy to implement.

This 7 DOF robotic arm shows a great working space, highly mobility and we are hoping to use this additional degree of freedom in complex applications.

We have tried tuning the ANFIS model, using different range divisions for the inputs and output variation and different number of Membership Functions, also different types of Membership Functions but the results were very similar.

Using the Fuzzy logic and ANFIS models we have tried to limit the movement of the elbow and fix it in order to force the robotic arm to adapt a certain position in space and to be able to avoid obstacles or to have a collision free trajectory.

We have used Adaptive Neuro-Fuzzy method because it is conceptually easy to understand and easy to implement. The mathematical concepts behind the model are very simple and have a more intuitive approach without a very high complexity. First of all by using ANFIS we can benefit of its great flexibility.

The only problem remains the singularity point when θ angle should be -180 degrees or 180 degrees, the extreme values of the θ interval of variation.

References

[1] L. Tian and C. L. Collins, ”Motion planning for redundant manipulators using a floating point genetic algorithm,” Journal of Intelligent Robots and Systems, pp. 138-143, 1995.

[2] J. Craig, “Introduction to robotics”, Reading, MA: Addison-Wesley, 1989.

[3] J. Angeles, “Fundamentals of Robotic Mechanical Systems: Theory, Methods, and Algorithms”, 3rd Edition, Springer- Verlag, ISBN-10: 0387945407, New York, 2006.

[4] G. Antonelli, G. Indiveri and S. Chiaverini, “Prioritized closed- loop inverse kinematic algorithms for redundant robotic systems with velocity saturations,” Proceedings of IEEE/RSJ Int. Conf. on Intelligent Robots and Systems, pp. 5892–5897, ISBN978- 1- 4244- 3803- 7, St. Louis, USA, Oct 2009.

[5] C. Klein, C. H. Hung, “Review of pseudoinverse control for use with kinematically redundant manipulators,” IEEE Trans. on Systems, Man, and Cybernetics, Vol. 13, pp. 245–250, ISSN 0018- 9472.

[6] B. Siciliano and G. Bastin, “Theory of Robot Control", Springer- Verlag, ISBN- 10: 3540760547, London, U.K., 1996.

[7] L. Sciavicco and B. Siciliano, “A solution algorithm to the inverse kinematic problem for redundant manipulators,” IEEE J. of Robotics and Automation, Vol. 4, No. 4, pp. 303–310, ISSN 0882- 4967, 1998.

[8] C. Carignan, “Trajectory optimization for kinematically redundant arms,” Journal of Robotic Systems, Vol. 6, No. 5, pp. 221–248, ISSN 0741- 2223, 1993.

[9] D., Pîslă, N., Plintea, and C., Vaida, “Kinematic Modeling and Workspace Generation for a New Parallel Robot Used in Minimally Invasive Surgery”, Advances In Robot Kinematics: Analysis And Design Part 7, pp. 459-468, 2008.

[10] T. Deepak, “Real-Time Inverse Kinematics of the Human Arm”, Presence. Vol. 5. No.4., pp. 393-401 by the Massachusetts Institute of Technology, .1996.

[11] D. E. Whitney, “Resolved motion rate control of manipulator and human prosthesis,” IEEE Trans. Man- Machine Systems, Vol. 10, No. 2, pp. 47–53, ISSN 0536- 1540.

[12] A. Liegeois, “Automatic supervisory control of the configuration and behavior of multibody mechanisms,” IEEE Trans. Systems, Man, and Cybernetics, Vol. 7, No. 12, pp. 868–871, ISSN 0018- 9472, 1977.

[13] L. Sciavicco and B. Siciliano, “A solution algorithm to the inverse kinematic problem for redundant manipulators,” IEEE J. of Robotics and Automation, Vol. 4, No. 4, pp. 303–310, ISSN 0882- 4967, 1988.

[14] C. Klein, C. Huang, “Review on pseudo inverse control for use with kinematically redundant manipulators,” IEEE Transactions on Systems, Man and Cybernetics, vol. 13, no. 3, pp. 245–250, 1983.

[15] W. A. Wolovich and H. Elliott, “A computational technique for inverse kinematics,” in IEEE Conference on Decision and Control, vol. 23, pp. 1359–1363, 1984.

[16] S. Klanke, D. Lebedev, R. Haschke, ”Dynamic Path Planning for a 7-DOF Robot Arm” Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp. 3879-3884, 2006.

[17] B. Siciliano, „Handbook of Robotics-Springer”-Kinematics, pp. 9-31.

[18] MathWorks Suport-Fuzzy Logic.

[19] MathWorks Suport-m-function.

[20] D. Howerd, „Application of Fuzzy Logic for the Solution of Inverse Kinematics and Hierarchical Controls of Robotic Manipulators” Journal of Intelligent and Robotic Systems 23: pp. 217–247, 1998.

[21] R. Balan, V. Maties, S. Stan, “Motion Control Using On-Line Simulation and Rule Based Control,” IEEE ISIE 2005, June 20-23, 2005, Dubrovnik, Croatia, pag. 167-172.