International Journal on
“Technical and Physical Problems of Engineering”
(IJTPE)
Published by International Organization of IOTPE

ISSN 2077-3528
IJTPE Journal
www.iotpe.com
ijtpe@iotpe.com

March 2022 Issue 50 Volume 14 Number 1 Pages 78-88

INTELLIGENT MODELING AND SIMULATION OF THE INVERSE KINEMATICS REDUNDANT 3-DOF COOPERATIVE USING SOLIDWORKS AND MATLAB/SIMMECHANICS

A. Bahani M.E. Ech Chhibat H. Samri H.A. El Attar

Laboratory of Modeling and Simulation of Intelligent Industrial Systems (M2S2I), Mechanical Engineering Department, ENSET, Hassan II University, Mohammedia, Morocco
abdeer.bahani@gmail.com, echchhibate@gmail.com, samrih127@gmail.com, aitelattar.hicham@gmail.com

Abstract- The problem of redundant industrial manipulator robots has always resided in obtaining the desired position and orientation in the operational space linked to the end-effector. For this reason, many researchers have intervened in this problem by trying to obtain the inverse kinematics of the robots by using several solutions. As these types of redundant robots offer the possibility to bypass obstacles, their inverse kinematics is very complicated and difficult to solve mathematically. The aim of this work is to replace conventional modeling with intelligent modeling based on an ANFIS (Adaptive neuro-fuzzy inference system) controller enriched with CAD (computer-aided design) data, after uploading to Matlab/SimMechanics. for that, we will detail in a first time the mathematical model of a robot with 3-DOF (3 degrees of freedom). In a second time, we proceed to the simulation on the two software to validate the mathematical model and to build an ANFIS controller of the kinematic and geometrical inverse model of the manipulator robot. To finally test this artificial ANFIS controller and validate it. The verification of the results obtained by the two software will let us evaluate the validity of the model thus created and come up with better conclusions.

Keywords: Robot Manipulator, Intelligent Control, ANFIS Controller, Intelligent Controller, CAD, Modeling, 3-DOF.

1. INTRODUCTION

Industrial robots are still widely used throughout the continuous progression of industrial automation. To replace humans with more complicated tasks, several manipulators are needed to work together. This improves the time, efficiency, and accuracy of industrial robots, but it also introduces a lot of problems related to trajectory planning and cooperative control, especially in the case of redundant robots.

Redundant manipulators offer a number of advantages. One of these is the multiplicity of solutions to the joint manipulator variables for a single end-effector position. This allows the redundant robot to avoid obstacles and singularities. Therefore, this type of robot is an excellent candidate for optimization and obstacle avoidance techniques.

In this paper, we try to study the cooperation of two industrial robots with 3-DOF under the constraint of the non-slip part to grasp. in this regard, we study the state of the art concerning geometric, kinematic, and dynamic modeling, in the case of the cooperation of manipulator robots. Then we will address the global coordination control in the operational space of a multi-robot system manipulating an object in space.

Indeed, we have used the two software SolidWorks and MATLAB/SimMecanics not only to validate the mathematical aspect but also to simulate the movement of the two redundant manipulator robots in the case of learning and verification of ANFIS controller. This method could be easily generalized to two identical 3-DOF robots and, consequently, to serial robots (Scara, Puma, etc.).

2. RELATIVE RESEARCH

In our current work, we concentrate on the development of a specific method for modeling the two redundant robot manipulators by applying the MATLAB/SimMechanics simulation technique, this modeling will allow us to use its results in the learning of the ANFIS controller. This technique has not yet been reported by other researchers. At least the studies conducted by [1, 2], except that they are based on analytical methods for learning the ANFIS controller.

Indeed, the intelligent controllers are already used by several researchers but the database was enriched by pure mathematical modeling which increases the duration of the training.

3. PROBLEM MODELING

3.1. Preliminary

In a first step, we take the modeling of two 3-DOF planar robots as shown in Figure 1 below:

3.2. Assumptions

In order to facilitate the formulation of the kinematic problem, we have made the following assumptions [3, 4].

- both robots are identical and redundant
- The center point of the tool (TCP) of both robots is connected by a point link.
- The common object is unreformable.
- The forces applied to the object ensure that the object does not slip with respect to TCP.

4. MODELLING BY SOLIDWORKS/MATLAB/SIMMECHANICS

SolidWorks is a modeling software used to design parts or objects through an assembly using parametric features. The geometry of the model is determined from the dimensions of each part. With the help of this CAD modeling software, the two redundant robots and the manipulated object are modeled. The mates applied to the parts during their assembly are translated by the SimMecanics scheme as shown in Figure 2. The selected design dimensions are listed in Table 1 (Appendix 1).

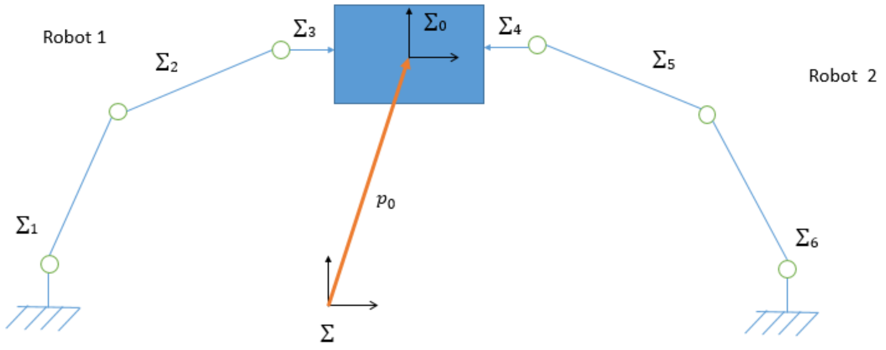


Figure 1. Modeling diagram of two cooperative robots with 3-DOF

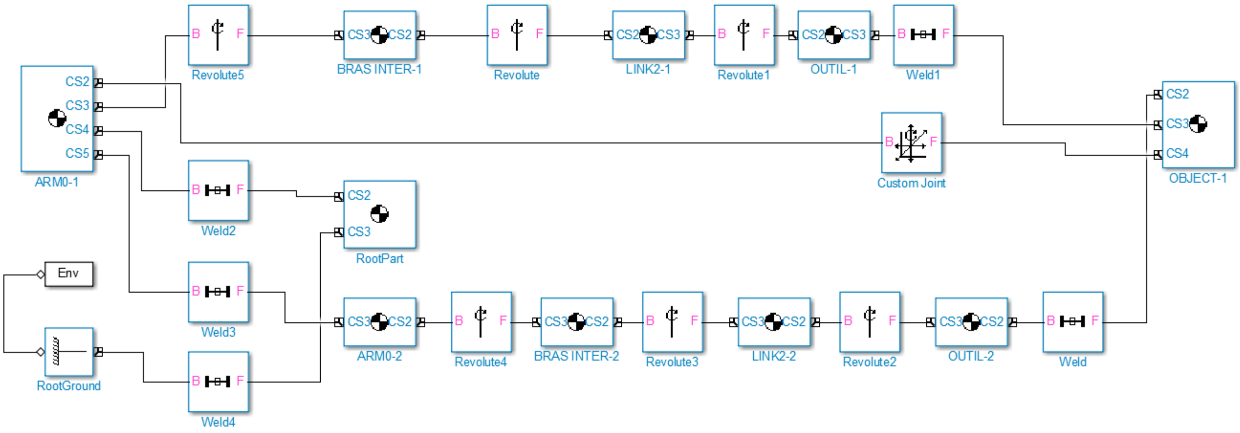


Figure 2. Diagram of two manipulators on Matlab/SimMecanics

5. GEOMETRIC AND KINEMATIC MODEL OF REDUNDANT ROBOTS

5.1. Forward Kinematics

First, we take the modeling of two 3-DOF flat robots as shown in Figure 3.

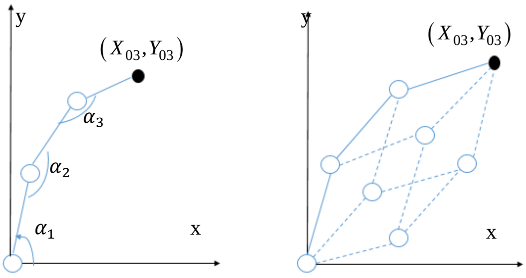


Figure 3. The different possible postures of 3-DOF robots in the same position and orientation of the TCP [5]

The Denavit-Hartenberg convention, in its original version, is often applied to a simple open-chain robot. This convention can be summarized as shown in Table 1 (Appendix 1).

In kinematics, the rigid transformation matrix is used to describe the transformation from a coordinate system associated with a manipulator's arm to a subsequent coordinate system from another arm.

In the general case, the transformation from one coordinate system to another is described by the following transformation matrix [6]:

$$A_{i,i+1} = \begin{bmatrix} \cos\alpha & -\sin\alpha\cos\beta & \sin\beta\sin\alpha & L_i\cos\alpha \\ \sin\alpha & \cos\alpha\cos\beta & -\sin\beta\cos\alpha & L_i\cos\alpha \\ 0 & \sin\beta & \cos\alpha & a_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

So, for our case:

$$A_{0,3}=\begin{bmatrix} C_{123} & -S_{123} & 0 & L_1C_1+L_2C_{12}+L_3C_{123} \\ S_{123} & C_{123} & 0 & L_1S_1+L_2S_{12}+L_3S_{123} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2)$$

$$C_{123}=\cos(\alpha_i+\alpha_j+\alpha_k) \quad (3)$$

$$S_{123}=\sin(\alpha_i+\alpha_j+\alpha_k) \quad (4)$$

The coordinates of TCP in the operational space are given in terms of the generalized coordinates [6]:

$$\begin{cases} X_{03}=L_1C_1+L_2C_{12}+L_3C_{123} \\ Y_{03}=L_1S_1+L_2S_{12}+L_3S_{123} \\ \phi=\alpha_1+\alpha_2+\alpha_3 \end{cases} \quad (5)$$

where, X_{03} and Y_{03} describes the position of TCP and ϕ describes the orientation of TCP [6]. For the planar 3R manipulator shown in Figure 4, the direct kinematics equations are given in Equations (6) and (8). To obtain the angles of the first robot manipulators:

$$\alpha_2=\pm\cos^{-1}\left(\frac{X_{03}{}^2+Y_{03}{}^2-L_1{}^2-L_2{}^2}{2L_1L_2}\right) \quad (6)$$

$$\alpha_1=A\tan 2(Y_{03},X_{03})-A\tan 2(k_2,k_1) \quad (7)$$

where, $k_2=L_2S_2$, $k_1=L_1+L_2C_2$ and finally:

$$\alpha_3=\phi-(\alpha_1+\alpha_2) \quad (8)$$

To simplify the studies, we take for this application always $\phi=0$.

5.2. Inverse Kinematics

The procedure used to calculate the coordinates of the joints as a function of the coordinates in operational space is called inverse kinematics. Basically, this procedure is based on the resolution of a set of formulas. However, these equations are, in general, non-linear and very difficult to solve. The inverse kinematic analysis becomes very complex and sometimes even impossible to find, especially in the case of redundant manipulators where uniqueness is not guaranteed. Sometimes there is not a unique set of joint coordinates for the given coordinates of the end effector.

For each given position and orientation of TCP, there are two different ways or postures, each corresponding to a different value of the other (elbow-up or down). This change of posture offered by the redundant manipulator will allow it to bypass the obstacles as shown in Figure 4.

The ambiguity of ordering a robot to do a specific trajectory lies in the variety of choice between the two possible configurations. sometimes this variety is not given if the limits of the joints are such that a posture cannot be realized.

5.3. Velocity Studies

The monitoring of the TCP trajectory during the manipulator movement is very fundamental in robotics not only to avoid obstacles during the movement but also to counteract the singularities of operation. Let's take the example when the manipulator has to maintain its TCP in a prefixed orientation while moving along the trajectory.

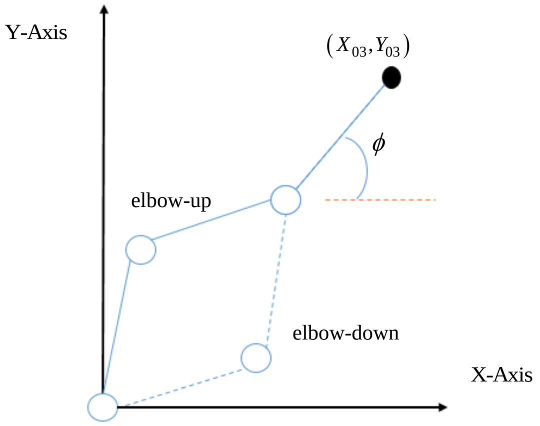


Figure 4. The two possible configurations for the same position and orientation (elbow-up and down) [5]

This constraint is often encountered in welding tasks or in general in machining. In fact, it is necessary to program the speed of the end effector or the TCP tool during these movements. As the robot controller controls the joint coordinates, one must be able to determine the joint speeds to make the desired TCP speeds. This control requires a more extensive kinematic study, one based not only on position or coordinate changes but also on the speeds or rate of coordinate changes.

Let's take again the redundant 3-DOF manipulator robot and derive Equation (1) with respect to time, we can then find equations that allow us to determine the different speeds [6].

$$\begin{cases} \dot{X}_{03}=-L_1\dot{\alpha}_1S_1-L_2(\dot{\alpha}_1+\dot{\alpha}_2)S_{12}-L_3(\dot{\alpha}_1+\dot{\alpha}_2+\dot{\alpha}_3)S_{123} \\ \dot{Y}_{03}=L_1\dot{\alpha}_1C_1+L_2(\dot{\alpha}_1+\dot{\alpha}_2)C_{12}+L_3(\dot{\alpha}_1+\dot{\alpha}_2+\dot{\alpha}_3)C_{123} \\ \dot{\phi}=\dot{\alpha}_1+\dot{\alpha}_2+\dot{\alpha}_3 \end{cases} \quad (9)$$

The Jacobian matrix is written in the following form:

$$\begin{pmatrix} \dot{X}_{03} \\ \dot{Y}_{03} \\ \dot{\phi} \end{pmatrix}=\begin{pmatrix} -L_1S_1-L_2S_{12}-L_3S_{123} & -L_2S_{12}-L_3S_{123} & -L_3S_{123} \\ L_1C_1+L_2C_{12}+L_3C_{123} & L_2C_{12}+L_3C_{123} & L_3C_{123} \\ 1 & 1 & 1 \end{pmatrix}\begin{pmatrix} \dot{\alpha}_1 \\ \dot{\alpha}_2 \\ \dot{\alpha}_3 \end{pmatrix} \quad (10)$$

We note that J the Jacobian matrix. This one describes the rate of TCP coordinates change with respect to the articular coordinates:

$$J=\begin{pmatrix} -L_1S_1-L_2S_{12}-L_3S_{123} & -L_2S_{12}-L_3S_{123} & -L_3S_{123} \\ L_1C_1+L_2C_{12}+L_3C_{123} & L_2C_{12}+L_3C_{123} & L_3C_{123} \\ 1 & 1 & 1 \end{pmatrix} \quad (11)$$

The non-singularity of the Jacobian matrix requires that its determinant is non-zero. If this condition is true, the matrix becomes invertible, which will allow us, as a consequence, to obtain the expression of the joint velocities as a function of the speed of TCP:

$$\dot{q}=J^{-1}\dot{P} \quad (12)$$

By determining the expression of the determinant J, we could know the conditions where the Jacobian matrix becomes irreversible and singular.

$$|J|=L_1L_2S_2 \quad (13)$$

The singularity is present when α_2 is equal to 0 or 180 degrees. This posture indicates that the elbow is fully extended or completely bent, in fact, this posture cannot be done physically on many robots. That's why it was necessary to avoid this configuration, except for this configuration robot had the ability to perform all desired speeds with the desired position and orientation of TCP.

6. COOPERATIVE ROBOTS CONTROL

6.1. Intelligent Control of Cooperative Robots

The control based on artificial intelligence represents a very vast and active research field thanks to the advantages that these tools offer especially for the nonlinear systems, it is based on the exploitation of these capacities of learning, approximation, and optimization that characterize these tools among which we cite the artificial neural networks (ANN), and the fuzzy logic-based controllers.

We are interested in the design of an ANFIS controller that will replace the conventional controller (Figure 5) this controller will be able to work in cooperation with another.

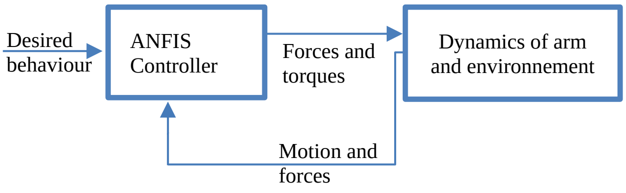


Figure 5. Block diagram of the intelligent robot controller *i*

6.2. Adaptive Network Fuzzy Inference Systems (ANFIS)

By combining the concepts of fuzzy logic and neural networks, ANFIS is improved and made usable in the 1990's by Jang [7]. Thus, ANFIS automatically improves the learning and adaptation capacity [8] by combining the concepts of fuzzy logic and neural networks in order to form an intelligent and hybrid system.

The modeling and forecasting of technological systems are done by these hybrid systems and adaptive techniques [7], [9], [10], [11] in order to develop a method and processes that learn information.

6.3. Architecture of ANFIS

The structure of ANFIS is composed of five layers, as shown in Figure 6. To simplify understanding, the following system has two inputs and a single output.

- Layer 1. Membership degree generation:

In the first layer, the function of each node *i* of is given by the following Equation (6):

$$\begin{cases} O_i^1 = \mu_{Ai}(x); \quad i=1,2 \\ O_i^1 = \mu_{Bi}(y); \quad i=1,2 \end{cases} \qquad (14)$$

where, μ_A and μ_B are fuzzy membership function (MF).

- Layer 2. Generation of the rule weight *i*:

The activation strength of each node of a rule is computed by multiplication:

$$O_i^2 = W_i = \mu_{Ai}(x)\mu_{Bi}(y); \quad i=1,2 \qquad (15)$$

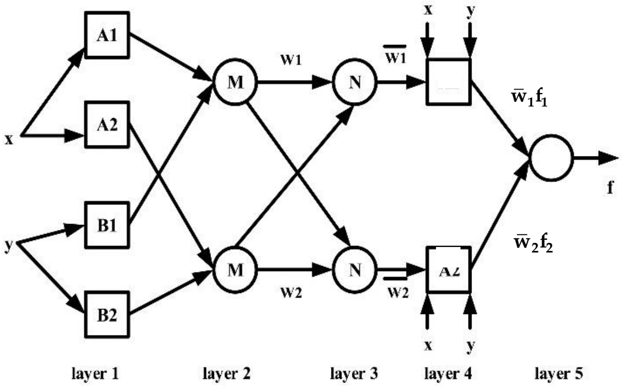


Figure 6. The different layers of ANFIS

- Layer 3. Normalisation of rule weights *i*:

The computation of the *i*th node of this layer by the ratio of the activation strength of the *i*th rule to the total activation strength of the other rules:

$$O_i^3 = \bar{W}_i = \frac{W_i}{W_1 + W_2}; \quad i=1,2 \qquad (16)$$

where, W_i denotes the activation force.

- Layer 4. Calculating rule output:

In this layer 4, each node *i* is a squared and having the following function:

$$O_i^4 = \bar{W}_i f_i = \bar{W}_i (p_i x + q_i y + r_i) \quad ; \quad i=1,2 \qquad (17)$$

where, $\bar{W}_i$ is the output of layer 3, and the settable parameters of the output of the rule *i* are designated by (p_i, q_i, r_i).

- Layer 5. Summarize all entries from layer "4":

In this layer, single node computes global output as summation of all input signals as expression (18).

$$O_i^5 = \sum_i \bar{W}_i f_i = \sum_{i=1}^{2} W_i f_i \Big/ \sum_i W_i \quad ; \quad i=1,2 \qquad (18)$$

The output *y* in Figure 6 is expressed as follows [11, 12]:

$$f_i = (\bar{W}_1 x) p_1 + (\bar{W}_1 y) q_1 + (\bar{W}_2 x) p_2 + (\bar{W}_2 y) q_2 + \bar{W}_2 r_2 \quad (19)$$

Neuro-fuzzy systems combine fuzzy logic and neural networks in order to be very efficient and useful for the identification of complex non-linear systems. This is why we chose to use the adaptive neuro-fuzzy inference systems ANFIS, in fact ANFIS can take any form of modeling. this of a multilayer neural network has five layers. each of these layers corresponds to the completion of a step of a fuzzy inference system of Takagi Sugeno type (FIS) [13].

The flowchart of the ANFIS procedure is shown in Figure 7.

7. SIMMECHANICS SIMULATION AND KINEMATICS STUDY

7.1. Simulation Methodology

SimMechanics [14] is a multi-body simulation environment for multi-physical and electromechanical systems. It can model links and constraints between mechanical parts while representing them by blocks.

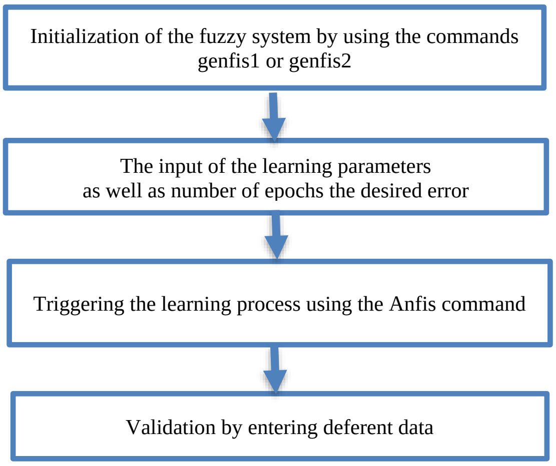


Figure 7. ANFIS Procedure

Based on CAD data, SimMechanics processes and manipulates the equations of the global mechanical system. Its strong point is the ability to manipulate the structure, optimize the mechanical characteristics and finally produce relevant results.

After modeling the CAD model by SolidWorks on the platform, it is imported into SimMechanics. The kinematics or even the dynamics of the system is visualized using a 3D animation automatically generated in MATLAB/Mechanics Explorer.

7.2. Kinematic Study by SimMechanics

The simulation is done both on Matlab/SimMechanics and Solidworks to simulate the kinematics of the object to manipulate and another simulation on Matlab/ SimMechanics for the control of the cooperative robots.

we present the training diagram of our ANFIS controller after having registered the joint coordinates in the workspace of MATLAB Figure 9.

In this paragraph we propose the model of manipulator robots and we consider the planar problem and the two identical robots are planar manipulators with 3-DOF as shown in Figure 9 below. The dimensions are of each link is $L_1 = 0.225$ m, $L_2 = 0.15$ m and $L_3 = 0.16$ m.

The direct kinematic equation from Chapter 4 is now used, or the workspace simulation results are obtained by manipulating the object using a "joint actuator" on SimMecanics. The data samples produced for training the neural network models considered are shown in Table 2 (Appendix 2).

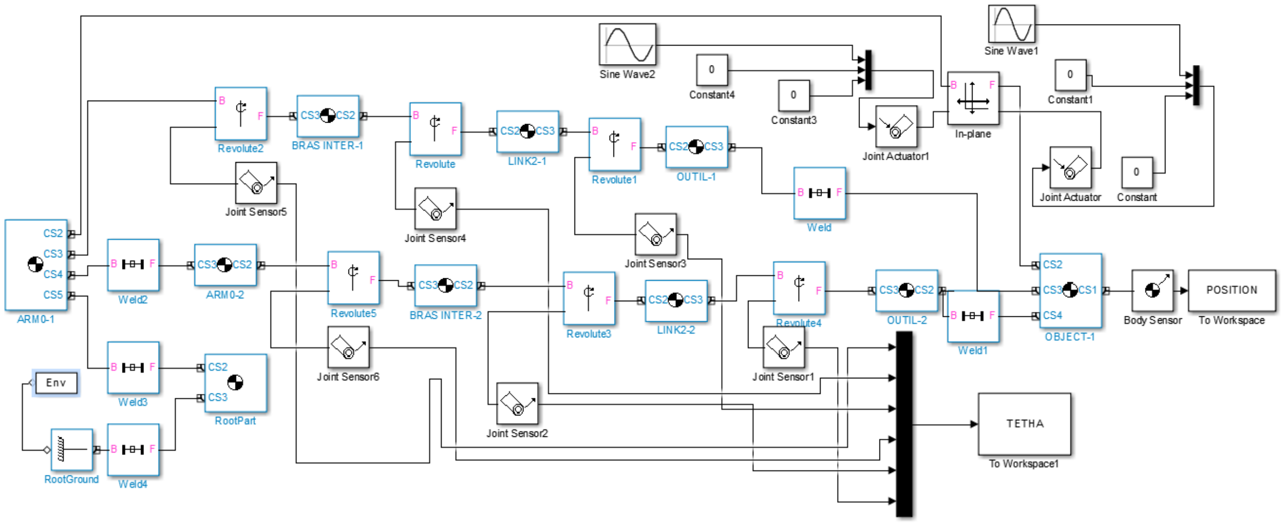


Figure 8. Diagram of the two manipulators on Matlab/SimMecanics in the learning situation

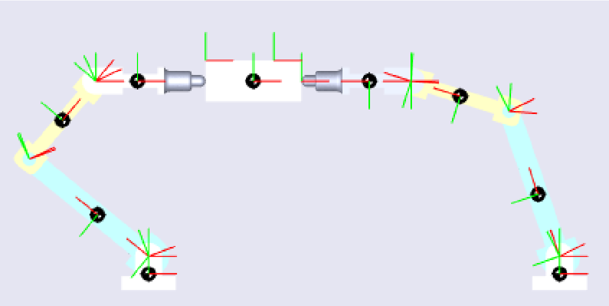


Figure 9. The 3-DOF robots modeled in MATLAB/Simulink

The learning process is based on the database drawn from "Workspace", an extract of which is shown in Table 2 (Appendix 2).

7.3. ANFIS Controller Implementation's

The combination of two controllers: fuzzy logic and neural network make a new hybrid controller called ANFIS controller. This combination improves the control characteristics of the machine compared to using a single controller. In order to implement this process in MATLAB in the MATLAB command window, we type "anfisedit". The ANFIS GUI consists of the basic steps below [12]:
Load Data: it consists in loading our recently saved data from the workspace.

- Generate FIS: by evaluating the number of clusters in the dataset, we can generate this FIS model

The structure of ANFIS can be observed by opening the "Structure" option as can be seen in Figures 10 and 11.

Training and validation of the FIS: In this step the FIS model is trained by repeating the pre-programmed epochs until the number of these epochs is reached and the learning error is satisfied.

The FIS is a very well-known and cardinal computer tool. This computer tool allows to process the fuzzy If-Then rules and to adjust these parameters in order to find an optimal solution. Indeed, an adaptive network is an artificial neural network (ANN) multilayer in which the outputs depend on the parameters of the adaptive and adjustable nodes. parameters of the adaptive and adjustable nodes. This adjustment of the parameters is done in order to respect the specified error [13, 17, 18].

The FIS test is performed by selecting "Test Now". We then use this database in order to learn our ANFIS controller which will then be injected into the inverse model in Figure 12.

7.4. Geometric and Velocity Analysis

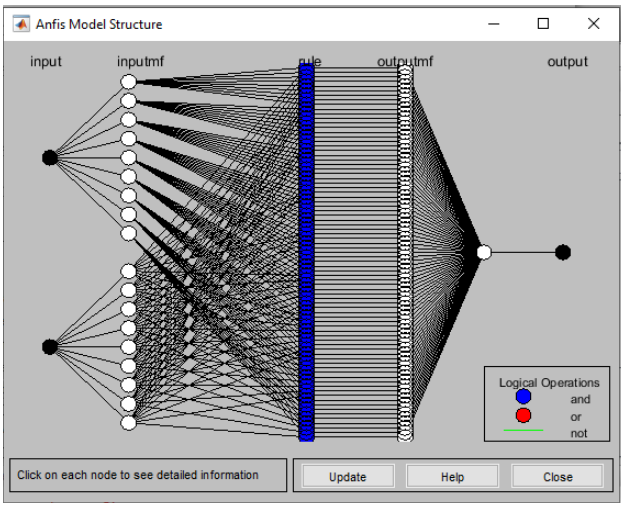
The simulation of the kinematics is done on Matlab/ SimMechanics for the control of the cooperative robots, we will be interested in the following movement of common-object:

- Translations along the x and y-axis are shown in Figures 13 and 15.
- Circular translation of a circle of diameter 100 mm is shown in Figure 17.

After having trained the controllers of each joint we can then control our robot by choosing a trajectory and velocity and we then compare this trajectory with those made using the fuzzy controller are shown in Figures 14, 16 and 19.

a. Translation of the common object along the x-axis.

b. Translation of the common object along the y axis.



Click on each node to see detailed information

Figure 10. ANFIS model structure for each joint coordinate *i*

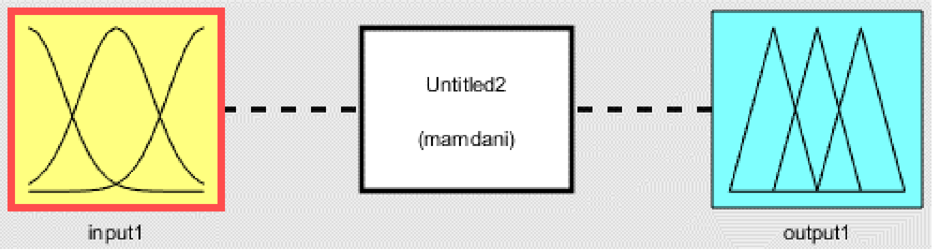


Figure 11. Relation between input-output in FIS editor [13]

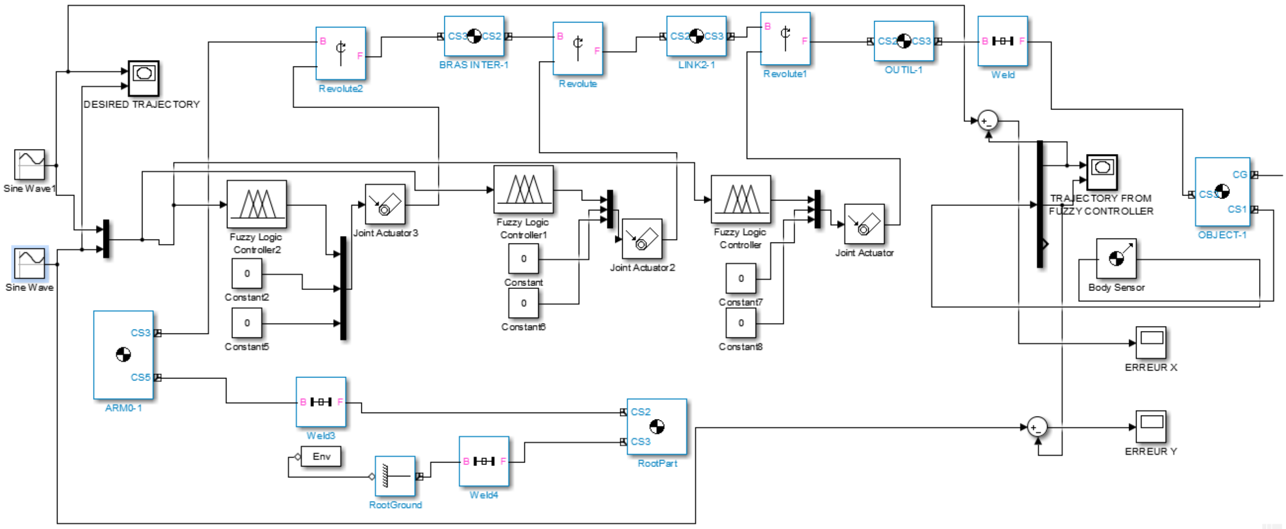
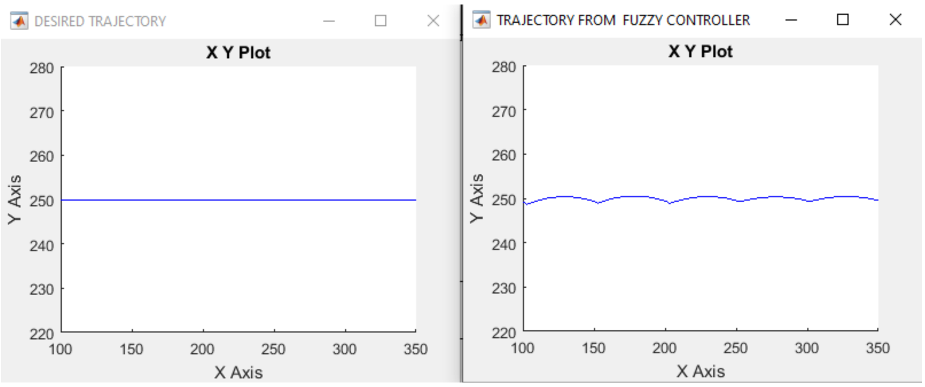
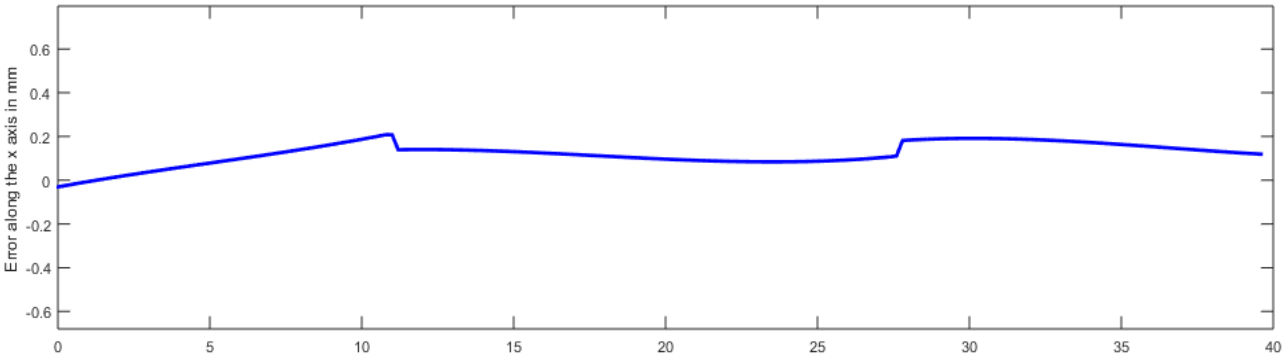


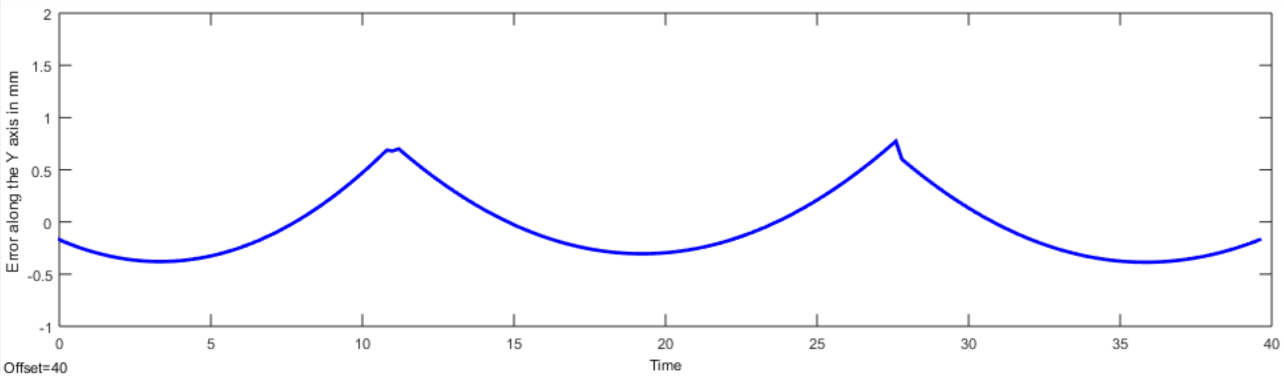
Figure 12. Diagram of the one manipulator on Matlab/SimMecanics in the inverse kinematics situation



(a) The desired position (b) The real position



(c) Position error, center of inertia of the manipulated object along the X-axis



(d) Position error, center of inertia of the manipulated object along the Y-axis

Figure 13. Simulation results translation along the X-Axis: (a) The desired position, (b) the real position, (c) Position error along X-axis, (d) Position error along Y-axis

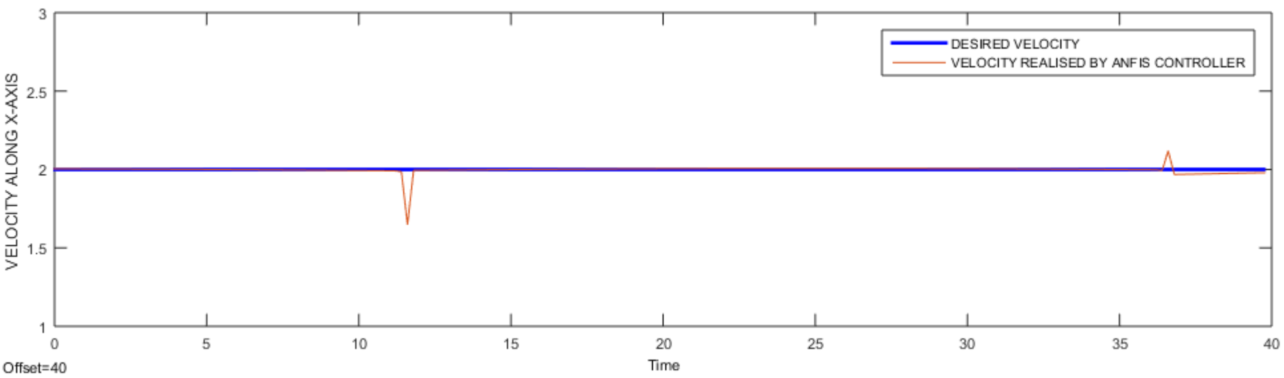
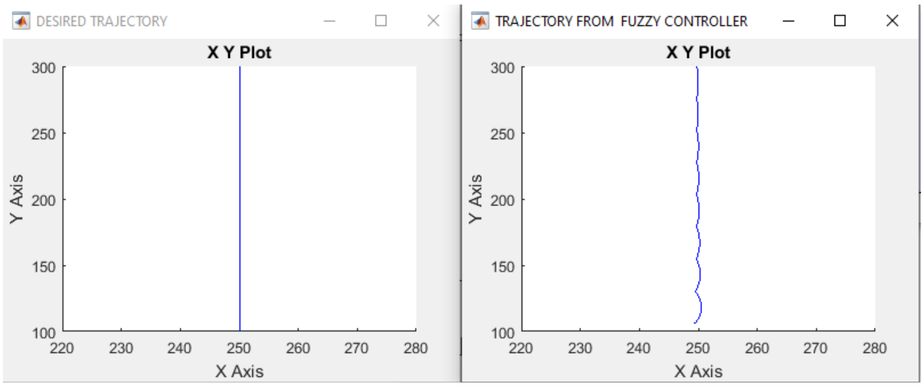
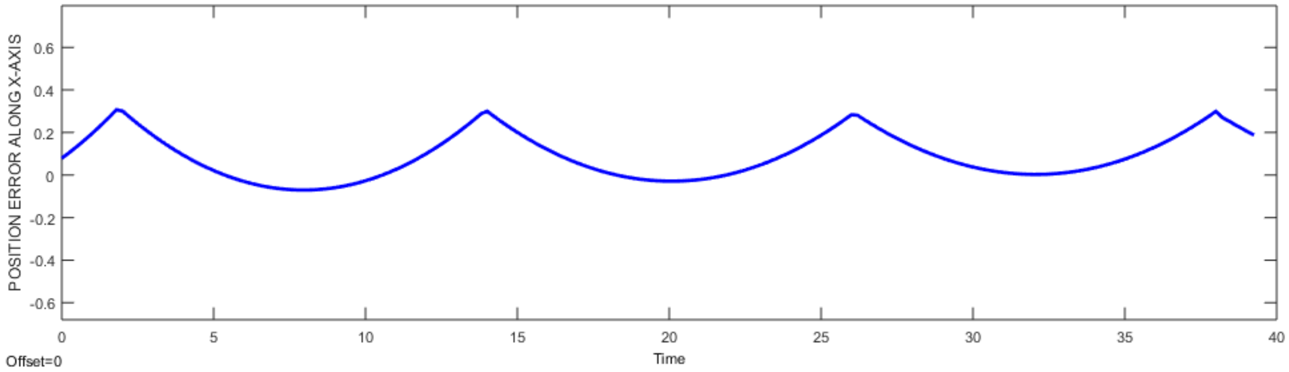


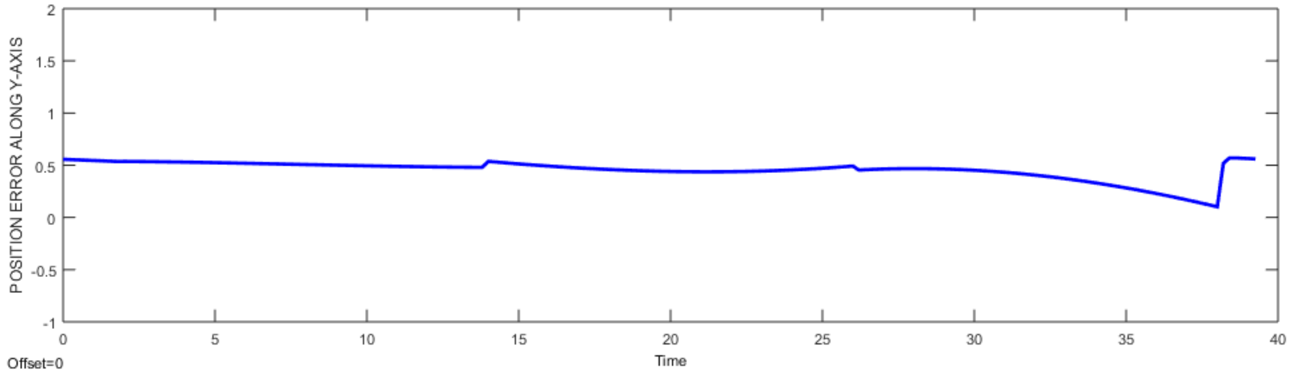
Figure 14. Velocity desired and the velocity generated by ANFIS controller vs time plot in mm/s



(a) The desired position (b) The real position



(c) Position error, center of inertia of the manipulated object along the X-axis in mm



(d) Position error, center of inertia of the manipulated object, along the Y-axis in mm

Figure 15. Simulation results translation along the Y-Axis in mm

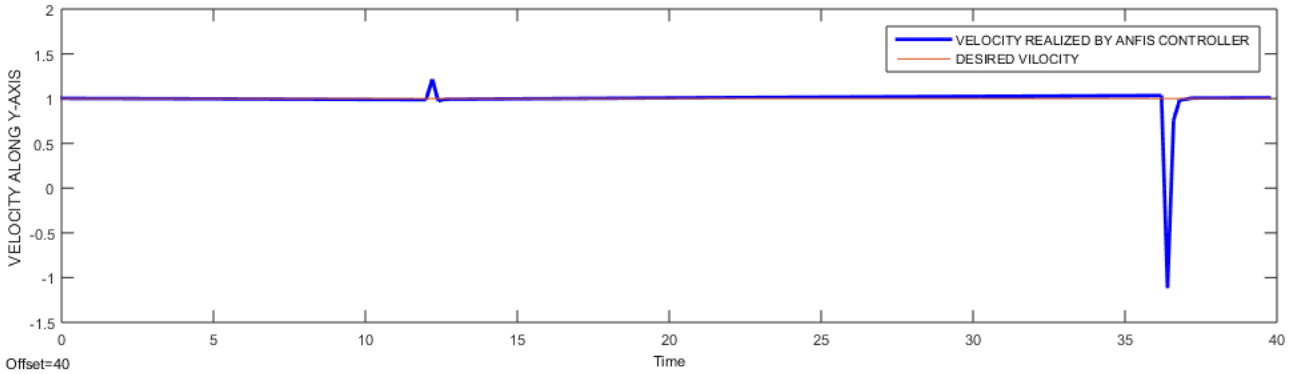
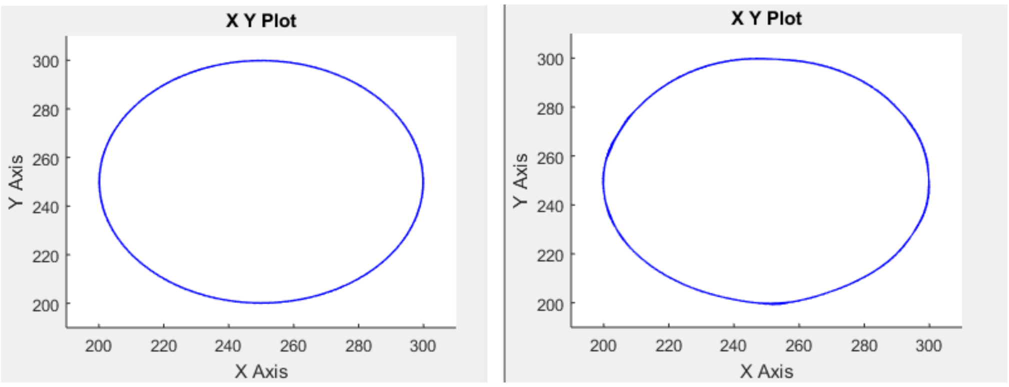
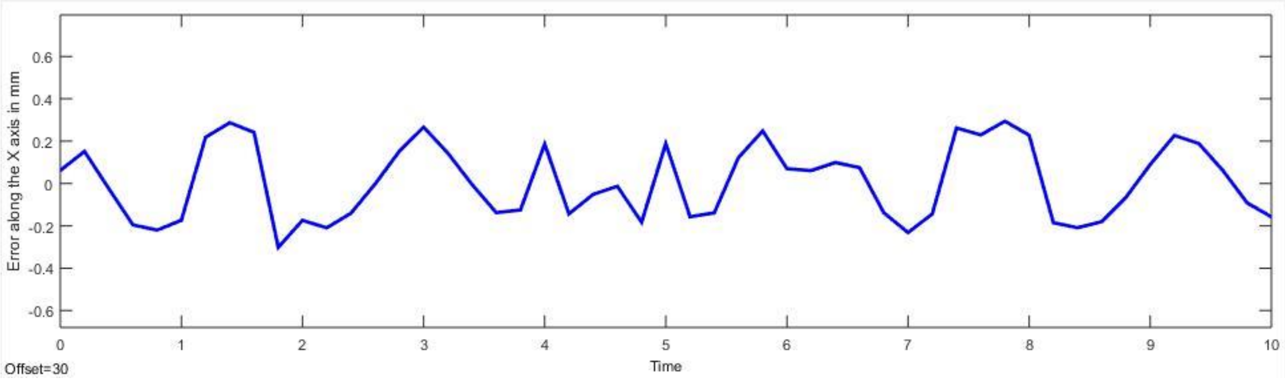


Figure 16. Velocity desired and the velocity generated by ANFIS controller vs time plot in mm/s

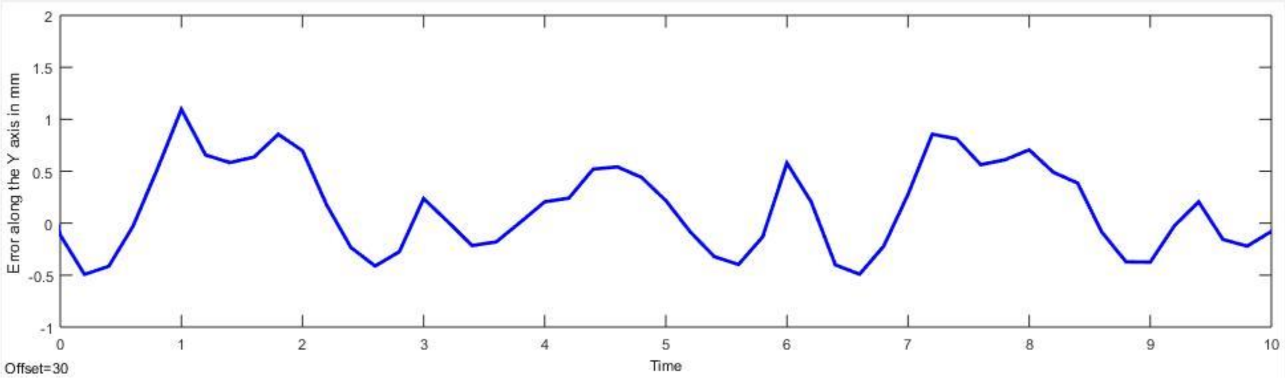


(a) The desired position

(b) The real position



(c) Position error, center of inertia of the manipulated object, along the X-axis



(d) Position error, center of inertia of the manipulated object, along the Y-axis

Figure 17. Simulation results of circle translation
diameter 100 mm

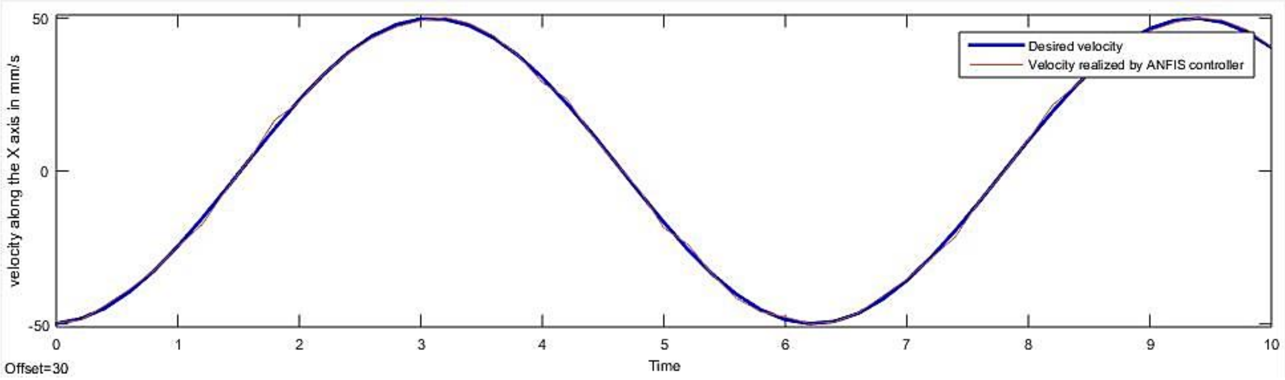


Figure 18. Velocity desired and the velocity along the X-axis generated by ANFIS controller vs time plot in mm/s

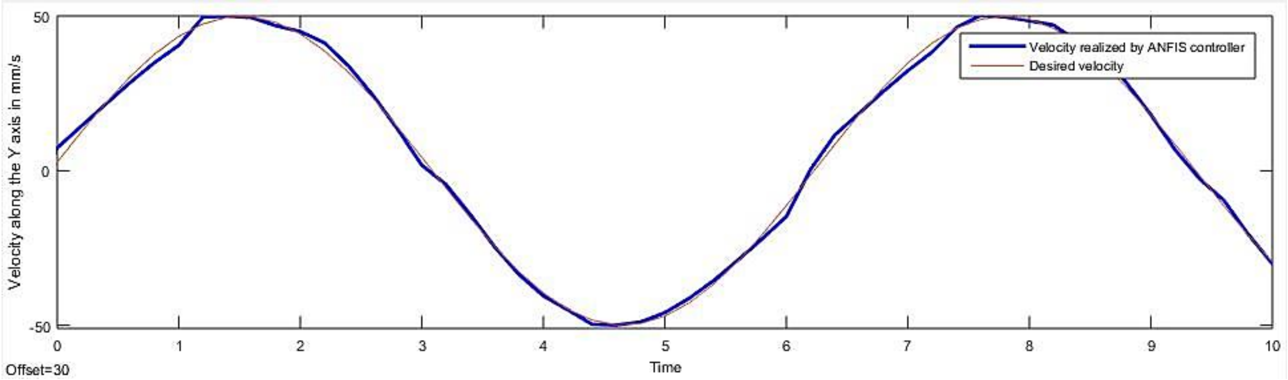


Figure 19. Velocity desired and the velocity along the Y-axis generated by ANFIS controller vs time plot in mm/s

The simulation data indicate that the ANFIS controller can reproduce desired trajectories even with a drive that does not exceed 100 epochs. The position error committed by this controller is about 0.6 mm as shown in the Figures, which validates this method.

Concerning the speed realized by the ANFIS controller, we notice that it is almost the same as the desired one with an error that does not exceed 0.05 mm/s for the two linear movements

The error between the desired speed and the one obtained using the ANFIS controller is negligible so that we notice that both robots follow the desired speed of the common object faithfully which validates this type of intelligent controller.

8. CONCLUSIONS

This article describes the first try to model and simulate the two redundant and cooperative manipulator robots using the SolidWorks CAD model and MATLAB/SimMechanics software. This modeling is based on a block representation of the robot CAD model which makes it easier for us to manipulate and obtain the results.

A new kinematic model was developed. The simulation of the inverse kinematics performed by the ANFIS bay controller was performed in the first-generation MATLAB/SimMechanics software by converting the 3D CAD model from the SolidWorks environment. then, the results of this simulation are used for the design and the training of the controller thus realized in MATLAB.

So finally, the mathematical complexity of the inverse kinematics especially for redundant manipulator robots becomes very accessible by this method which is entirely based on a combination of CAD model design and the creation of an intelligent controller. The results of this work show the validity of this method.

APPENDICES

Appendix 1. The D-H Parameter Values

Table 1. The DH settings for various links

LINK	a_i	a_i	d_i	θ_i
1	L_1	0	0	a_1
2	L_2	0	0	a_2
3	L_3	0	0	a_3

Appendix 2. Extract of the Database for the First Robot

Table 2. Extract of the database for training ANFIS controllers (from the workspace of Matlab/SimMechanics)

P_0		a_1	a_2	a_3
X_0	Y_0			
252,871	106,812	-6,951948	-59,38130	66,333258
252,871	106,812	-6,951948	-59,38130	66,333258
272,837	106,832	-14,921076	-56,73688	71,657958
292,6048	106,852	-21,707850	-53,14814	74,855994
311,9750	106,872	27,4126304	-48,85107	76,263703
330,7546	106,892	-32,282327	-44,12080	76,403136
348,7561	106,912	-36,520875	-39,17836	75,699241
365,7994	106,932	-40,270248	-34,19365	74,463904

NOMENCLATURES

1. Acronyms

DOF	Degree of freedom.
ANFIS	Adaptive neuro-fuzzy inference system
D-H	Denavit Hardenberg
CAD	Computer-Aided Design
3D	Tree dimension
ANN	Artificial neural network
FIS	Fuzzy Inference System
TCP	Tool center point
MF	Membership Function

2. Symbols / Parameters

Σ_0 : The common-object coordinate system

S_0 : Common-object

Σ_i : The arm coordinate system of each robot *i*

P_0 : Vector position of center of inertia of common object

P_i : Vector the position of center of inertia of each link *i*

Ω_0 : Vector of orientation common-object

J : Jacobian matrix

$|J|$: Determinant of Jacobian matrix

α_i : Angle of *i* and *i*+1 links for the robot 1

β_i : Angle of *i* and *i*+1 links for the robot 2

ϕ: Angle that link 3 makes with respect to the horizontal

X_{03}, Y_{03} : The TCP coordinates with respect to frame 0

X_0, Y_0 : The coordinates of the center of inertia of the common object with respect to frame 0

L_i : The Link *i* length.

REFERENCES

[1] P. Jha, B.B. Biswal, “Inverse Kinematic Solution of 5R Manipulator Using ANN and ANFIS”, International Journal of Robotics and Automation (IJRA) Vol. 4, No. 2, June 2015.

[2] P. Jha, B.B. Biswal, O.P. Sahu, “Inverse Kinematic Solution of Robot Manipulator Using Hybrid Neural Network”, International Journal of Materials Science and Engineering, Vol. 3, No. 1, pp. 31-38, March 2015.

[3] H. Kawazaki, A. dabdatif, “Desentralised Coordinated Control of Multiple Robot Arms”, IFAC Robot Control, Wroclaw, Poland, 2003.

[4] S. Ivanov, Z. Meleshkova, L. Ivanova, “Calculation and Optimization of Industrial Robots Motion”, 26th Conference of Open Innovations Association (FRUCT) pp. 115-123, April 2020.

[5] T. Ogawa, H. Kanada, “Solution for Ill-Posed Inverse Kinematics of Robot Arm by Network Inversion”, Journal of Robotics, Special Issue, Vol. 2010, Article ID 870923, p. 9, 2010.

[6] S.E. Ivanov, T. Zudilova, T. Voitiuk, L.I. Ivanova, “Mathematical Modeling of the Dynamics of 3-DOF RobotManipulator with Software Control”, The 9th International Young Scientist Conference on Computational Science, pp. 311-319, YSC 2020.

[7] A. Ghosal, “Kinematics of Serial Manipulators”, “Handbook of Manufacturing Engineering and Technology”, Department of Mechanical Engineering, Indian Institute of Science, pp. 1777-1808, Bangalore, September 2014.

[8] A. Talli. L. Kotturshettar, “Forward Kinematic Analysis, Simulation & Workspace Tracing of Anthropomorphic Robot Manipulator by Using MSC. ADAMS”, International Journal of Innovative Research in Science Engineering and Technology, Issue 4, Vol. 1, January 2015.

[9] J.S.R. Jang, “Adaptive Network Based Fuzzy Inference Systems”, IEEE Transactions on Systems Man and Cybernetics, pp. 665-685, 1993.

[10] C.T. Lin, C.S.G. Lee, “Neural Fuzzy Systems: A Neuro-Fuzzy Synergism to Intelligent Systems”, Prentice Hall, Upper Saddle River, Har/Dskt edition, May 1996.

[11] V.A. Constantin, “Fuzzy Logic and Neuro-Fuzzy Applications Explained”, Prentice Hall, Englewood Cliffs, New Jersey, 1995.

[12] K.V. Rop, D.B.O. Konditi, H.A. Ouma, S.M. Musyoki, “Parameter Optimization in Design of a Rectangular Microstrip Patch Antenna Using Adaptive Neuro-Fuzzy Inference System Technique”, International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 12, Vol. 4, No. 3, pp. 16-23, September 2012.

[13] A. Rezaeifar, R. Effatnejad, A. Dehghani Tafti, “Turbine Dynamics Study Control using Adaptive Neuro-Fuzzy Inference System (ANFIS)”, International Journal on Technical and Physical Problems of Engineering (IJTPE), Issue 17, Vol. 5, No. 4, pp. 81-87, December 2013.

[14] V. Fedak, F. Durovsky, R. Uveges, “Analysis of Robotic System Motion in SimMechanics and MATLAB GUI Environment in MATLAB Applications for the Practical Engineer”, 3rd ed., K. Bennett (Ed.), pp. 565-581, Rijeka, Croatia, 2014.

[15] H. Idadoub, M. Kourchi, M. Ajaamoum, D. Yousfi, A. Rachdy “Comparison and Experimental Validation of Three Photovoltaic Models of Four Technology Types” Technical and Physical Problems of Engineering (IJTPE), Issue 41, Vol. 11, No. 4, pp. 1-10, December 2019.

[16] Y.J. Wang P.C. Hsu, “Modeling of Solar Cells and Modules Using Piecewise Linear Parallel Branches”, IET Renewable Power Generation, Issue 3, Vol. 5, pp. 215-222, May 2011.

[17] J.S.R. Jang, “Neuro-Fuzzy Modelling for Dynamic System Identification”, Computer Science Department, Tsing Hua University, Hsinchu, Taiwan, 1998.

[18] M.N. Cirstea, A. Dinu, J.G. Khor, M. McCormick, “Neural and Fuzzy Logic Control of Drives and Power Systems”, Elsevier Ltd., Newnes, 2002.

BIOGRAPHIES

Abderrahim Bahani was born in Zagora, Morocco, in 1989. He received the B.Sc. and M.Sc. degrees all in Mechanical Engineering from University of Mohammed V, Rabat, Morocco in 2018 and 2020, respectively. He also received an external aggregation degree in mechanical engineering in 2015. Currently, he is a lecturer of Mechanical Engineering at preparatory classes for engineering schools, Mohammedia, Morocco. His research interests are in the area of mechanical modeling, robotics, and artificial intelligence.

El Houssine Ech Chhibat was born in Morocco, 1967. He is an Associate Professor and Doctor of Mechanical Engineering. He has the Accreditation to Direct Research (AD R) in the laboratory of signals, distributed systems and artificial intelligence at Higher Normal School of Technical Education, Hassan II University, Casablanca, Morocco. He is interested in maintenance, reliability, robotics and industry 4.0.

Hassan Samri was born in Casablanca, Morocco, in 1970. He received the B.Tech. in 1989, the DESA in Physical Sciences and Doctorate degree from Mohammed V University, Rabat, Morocco in 2006 and 2015, respectively. Currently, he is Professor of Mechanical Engineering at ENSET, Mohammedia, Morocco. He is the Head of FI-GMSI Engineers (Mechanical Engineering of Industrial Systems). His research interests focus on mechanical modeling, robotics, and artificial intelligence

Hicham Ait El Attar was born in Beni Mellal, Morocco in 1986. He received the Master degree in industrial systems engineering from University of Evry Val d'Essone, France in 2012. Currently, he is a lecturer of robotic with “Makermind” and a lecturer of computer vision at ESITH high school, Casablanca, Morocco. He is interested in computer vision, robotics and industry 4.0.