Path Planning of the Manipulator Arm FANUC Based on Soft Computing Techniques

Loubna Bouhalassa, Laredj Benchikh, Zoubir Ahmed-Foitih, Kamel Bouzgou

Path Planning of the Manipulator Arm FANUC Based on Soft Computing Techniques

L. Bouhalassa[1], L. Benchikh[2], Z. Ahmed-Foitih[3], K. Bouzgou[4],

Abstract *–In this paper, direct and inverse geometric models for a 6 degrees of freedom manipulator robot arm are developed, and a set of homogeneous matrices are generated by Denavit-Hartenberg formalism. Moreover, a path planning method based on soft computing techniques is presented, which consists of using the neural network to model the end-effector workspace, and then determining the optimal trajectory to reach a desired position. The optimization of the trajectory depends on the minimization of the cost function, defined by the sum of two energies. The first one is the collision penalty (E_c) generated by each obstacle shape; the second one is the trajectory length penalty (E_l). Four steps are considered; at first, the configuration of the robot workspace where any assumption is taken into account. The robot is assimilated to a particle that moves inside a small two-dimensional space, and the obstacle is a polygon. Second, the Multilayer Perceptron Neural Network (MLP neural network) is used for the robot workspace modeling. Model block inputs are robot's previous position values and the outputs are robot's following position values. Then, the proposed approach is applied to optimize the cost function and to determine the end-effector trajectory in case of obstacles. The obtained result is a smooth trajectory, which represents robot motion. Finally, the presented approach is validated on a FANUC robot arm in a virtual environment, where the simulation results show this approach efficiency.*

Keywords*: Artificial Intelligence, Neural Network, Path Planning, Manipulator arm, Geometric Modeling, Kinematic Modeling, FANUC Robot.*

Nomenclature

${}^0A_E$	Orientation matrix of frame R_E/R
C_j^k	Degree of collision of the j-th point with obstacle k
dj	Link offset along previous z to the common normal
e_r	Error
E	Translation matrix of the coordinate system
E_T	Total energy
E_c	Energy of the collision
E_l	Energy of the length f the trajectory
f	Activation function
p	Number of subdivision
IM_m	Input of neural hidden layer
IT	Input of neural output layer
k	Number of obstacles
l_j	Distance between two points
N	Number of the points in the trajectory
Nbr	Total number of obstacles vertices
OT	Output of neural output layer
OM_m	Output of neural hidden layer
$P(x,y)$	Point
$[P_x\ P_y\ P_z]^T$	End-effector position vector

q_i	Joint variable of the i-the link
Ri	Link length of the common normal
SM_m	Bias of the hidden layer
ST	Bias of the output layer
T	Parameter appropriate for the neural network
${}^oT_k$	Transformation matrix
w_l, w_c	Weights of the length, collision
$w_{x,m}; w_{y,m}$	Weights of the input layer
x, y	Coordinates of the point (input layer)
X_{min}, Y_{min}	Minimum X, Y coordinates
X_{max}, Y_{max}	Maximum X, Y coordinates
x_j, y_j	Coordinates of the j-th point
αi	Twist angle about common normal, from old z axis to new z axis
θi	Joint angle about z, from old x to new x

I. Introduction

The past decade has seen a renewed importance in the automatic path planning of robots without collision in the environment and it has been the subject of a very important number of searches. Several techniques of planning have been proposed in literature. However, there is not a general method to solve a planning problem of robots. The path planning is the determination of the trajectory, which allows the robot to move from the start position to the target without collision with any obstacles along the trajectory. The optimization of the function cost is expressed by two terms; the first one is the distance crossed by the robot between two position points (start and target). The Second one is the time or the energy optimization necessary for the execution of the manipulator displacement.

The path planning of the end-effector can be defined as a sequence of translation and rotation from the starting position to the desired one, while avoiding static and dynamic obstacles in the workspace of the manipulator [1]-[2].

Therefore, several path planning methods in the literature are presented; the most frequently used is a decomposition method cited in[3]-[4], when the configuration of the robot arm is defined and the free spaces of that configuration can be divided into small regions called cells.

Robots designed to assist the human operator require various features such as stability, human-machine interface, obstacle avoidance, and path planning. Thanks to extensive research on conventional techniques, path planning offers a new development path. Various trajectory planning techniques have been compared for real application in an internal environment in the recent works, as shown by Galceran [5], Pol [6],Bharadwa [7] and neural networks have been shown to be reliable alternatives to traditional methods by Li in [8], and Ren and al in [9].

The objective of the path planning approach is to provide a collision-free path to reach the desired position of the robot end-effector in optimal conditions. The potential field method is presented in [10]-[11]-[12]. Therefore, authors in [13] have used a combination of parallel navigation and potential fields guidance methods and then performance metrics have been improved for a mobile robot rendezvous. In addition, The sub-goal method is presented in [14]-[15], sampling-based methods in [16], neural network has been mentioned in [17], Distance Wave Transform can be found in [18], a Start Algorithm is tested in [19], where a D start algorithm is presented by authors in [20]. However, a modified A start algorithm is presented in [21].

Two types of trajectory planning are mostly used in practice, i.e., the path planning and the planning of movement of robot presented in [22]-[23].Therefore, in this work, the first type of Path planning will be introduced.

The classic methods of the most used path planning are the visibility graph algorithm and the artificial potential field algorithm. However, the former lacks flexibility and the latter is prone to suffer from difficulties with local minima [17].

Since manipulator robots are used in the industry where t he environment is very inaccessible and obstacles are des cribed in 3D, these classic methods cannot provide reliab le short time answers [24].

Recently, the neural network is considered an effective way to generate a trajectory for a manipulator robot particularly [25]. Using neural networks for the control of robot manipulators has attracted much attention and various related schemes and methods have been proposed and investigated [26]-[27].

In this paper, a neural networks-based approach is introduced for a robot path planning by avoiding collisions. The developed algorithm is used to find a shortest safe trajectory for the robot and to optimize the function cost. The efficiency of the proposed neural network in solving the path planning problems is clearly demonstrated through a simulation applied on an optimal trajectory for the FANUC robot end-effector [28].

This paper is structured as follows. Section II describes the FANUC LR MATE 200iB modelling. Section III presents the two developed algorithms, first one for optimal trajectory extraction, and the second one for subdivision method. In Section IV, simulation tests are performed to validate the proposed approach. Lastly, conclusions and future work are drawn in section V.

II. FANUC LR MATE 200iB Modelling

In order to show the efficiency of the proposed approach, the versatile FANUC LR MATE 200iBcan be used. This manipulator is compact and it is a modular construction, and it is a tabletop robot that can be used for a variety of applications. The 200iB robot is electric servo-driven and has a 6-Dof revolute joints and with many different mounting capabilities, and it has the ability to flip over backward for a larger work envelope and it provides the maximum flexibility.

Additional benefits include joint velocity that can reach up to 480 degrees/sec, the end-effector built on the wrist design, fail-safe brakes on the second and third axis, and an integral internally mounted solenoid valve pack. The brushless AC servomotors and the harmonic drives on all the axes need minimal maintenance. In addition, there are sealed bearings and drives, and internally installed cables and additional safety tooling services [29].

Moreover, FANUC manipulator arm is the most used in the manufacturing tasks and in academic research, due to the kinematic of the wrist that is a RRR type and its three revolute joints with intersecting axes, equivalent to a ball

socket. In [21] the FANUC 200iC is used by authors to define singularity regions and studying the Jacobian matrix determinant. In addition, the recursive Newton-Euler method and the vector multiplication method are used for the Jacobian matrix generation and validation.

Whereas, in [30], authors use the same manipulator arm, they determinate singularity regions by using a numerical method and they validate the results in the virtual reality environment developed in the VRML and Matlab interface.

In [31], the complete forward kinematic model using analytical and numerical approaches is given. The manipulator arm consists of a FANUC Robotics 6-Dof robotic arm LR MATE 200iC. Calculations are extracted and all the results are checked and compared with the software data generated by the robot.

In this paper, the FANUC LR MATE 200iB is used. The structure and the architecture of this manipulator are similar to the 200iC and the results of the proposed method and simulations can be tested in the real-time in the Lab. Figure 1 shows FANUC LR MATE 200iB robot dimensions and its workspace.

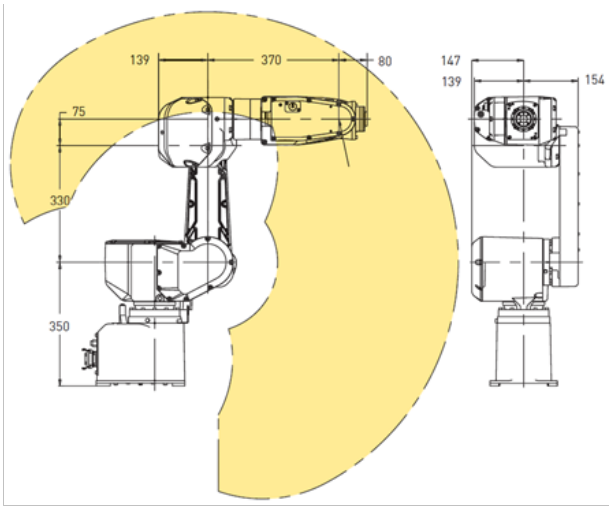


Figure1:FANUCLR MATE 200iB robotdimensions and its workspace

With

$d_2 = 150mm$, $d_3 = 250mm$, $d_4 = 75mm$, $r_4 = 290mm$, $r_6 = 80mm$

From a methodological viewpoint, firstly, the z_j axes will be placed on the joint axes, and then the x_j axes, geometric parameters of the robot are determined. Therefore, frames placement of the robot arm is shown in Figure 2, [32]-[33].

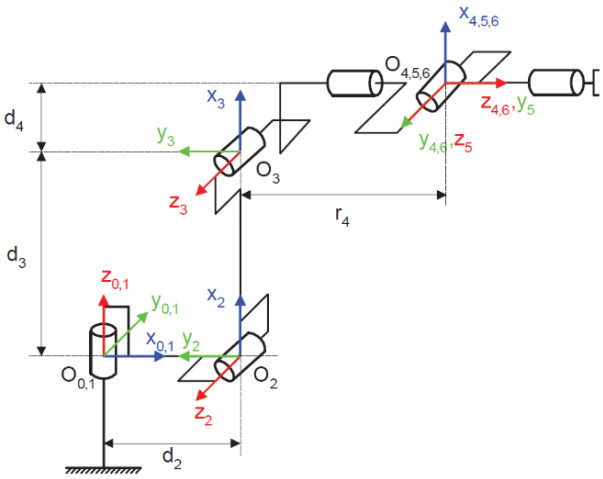


Figure2: FANUC robot architecture

Axes 4,5 and 6 are concurrent, the orientation of the end-effector is presented, and it has not affected its position. For that effect, E matrix that represents the attitude of the attached tool with respect to the end-effector frame can be defined, where a r value relative to the tool length is added. The passing from z_1 to $z_2$is done with a $\frac{\pi}{2}$ rotation angle, around x_1 axis, therefore, θ_2 becomes $\theta_2 + \frac{\pi}{2}$. The passing from $x_1 \rightarrow x_2$ is done with $\frac{\pi}{2}$, around Z_2 axis. For more details, [1] should be studied.

The coordinate axes for the robot allow the Denavit-Hartenberg parameters presented in table I below to be obtained.

TABLE I: Denavit-Hartenberg parameters of the FANUC robot

Joint i	α_i	d_i	θ_i	r_i
1	*0*	*0*	θ_1	*0*
2	*90*	d_2	$\theta_2+\pi/2$	*0*
3	*0*	d_3	θ_3	*0*
4	*90*	d_4	θ_4	r_4
5	*–90*	*0*	θ_5	*0*
6	*90*	*0*	θ_6	r_6

The sequence of the link coordinates assigned by the Denavit-Hartenberg convention is again transformed from the coordinate frame i to (i–1) where i is the joint, then using homogeneous coordinate transformation matrix given in the next subsection.

II.1. Geometric model of the robot

The homogeneous transformation matrices are defined as

$$
{}^{0}T_{1}=\begin{bmatrix} c\theta_1 & -s\theta_1 & 0 & 0\\ s\theta_1 & c\theta_1 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^{1}T_{2}=\begin{bmatrix} s\theta_2 & -c\theta_2 & 0 & d_2\\ 0 & 0 & -1 & 0\\ c\theta_2 & s\theta_2 & 0 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{2}T_{3}=\begin{bmatrix} c\theta_3 & -s\theta_3 & 0 & d_3\\ s\theta_3 & c\theta_3 & 0 & 0\\ 0 & 0 & 1 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^{3}T_{4}=\begin{bmatrix} c\theta_4 & s\theta_4 & 0 & d_4\\ 0 & 0 & -1 & -r_4\\ -s\theta_4 & c\theta_4 & 0 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
{}^{4}T_{5}=\begin{bmatrix} c\theta_5 & -s\theta_5 & 0 & 0\\ 0 & 0 & 1 & 0\\ -s\theta_5 & -c\theta_5 & 0 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
{}^{5}T_{6}=\begin{bmatrix} c\theta_6 & s\theta_6 & 0 & 0\\ 0 & 0 & -1 & 0\\ -s\theta_6 & c\theta_6 & 0 & 0\\ 0 & 0 & 0 & 1 \end{bmatrix}
$$

$$
E=\begin{bmatrix} 1 & 0 & 0 & 0\\ 0 & 1 & 0 & 0\\ 0 & 0 & 1 & r_6\\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (1)
$$

With $c_i=\cos(\theta_i)$ and $s_i=\sin(\theta_i)$ and $E={}^{0}T_{E}$,is the transformation matrix between the attached end-effector frame with respect to the reference frame.

II.2.1 Direct geometric model

The direct geometric model (DGM) is the set of relations that express the position of the end-effector [34], i.e. the operational coordinates of the robot, according to its joint coordinates. In the case of a simple open chain, it can be represented by the matrix transformation ${}^{0}T_{k}$ defined as the following equation

$$
{}^{0}T_{k}=\prod_{i=1}^{k}{}^{i-1}T_{i}\,q_i \qquad (2)
$$

The transformation matrix from R_0 frame to R_6frame can be written as the successive multiplication of all the elementary matrices for each axe from the manipulator base to the end-effector attached frame; it can be written as

$$
{}^{0}T_{6}={}^{0}T_{1}\,{}^{1}T_{2}\,{}^{2}T_{3}\,{}^{3}T_{4}\,{}^{4}T_{5}\,{}^{5}T_{6} \qquad (3)
$$

If ${}^{f}T_{E}={}^{0}T_{6}\times E$, where E is the transformation matrix from the tool frame to the end-effector frame, thus ${}^{f}T_{E}$ can be written as follows:

$$
{}^{0}T_{6}=\begin{bmatrix} s_x & n_x & a_x & P_x\\ s_y & n_y & a_y & P_y\\ s_z & n_z & a_z & P_z\\ 0 & 0 & 0 & 1 \end{bmatrix}=\begin{bmatrix} & {}^{0}A_{6} & & P\\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (4)
$$

After computing and identifying the terms of the two matrices of the equation (1) and (2), the following expression can be defined

$$
{}^{f}T_{E}(1:3,4)=T(1:3,4)
$$

$$
\begin{cases}
P_x=c_1d_2+r_6c_5c_1c_{(2-3)}+s_5s_1s_4-c_4c_1s_{(2-3)}\\
-d_4c_1s_{(2-3)}+r_4c_1c_{(2-3)}-c_1s_2d_3\\
P_y=d_2s_1-d_4s_1s_{(2-3)}+\\
r_6\left[c_5s_1c_{(2-3)}-s_5\left(c_4s_1s_{(2-3)}\right)\right]\\
+r_4s_1s_{(2-3)}-d_3s_1s_2\\
P_z=c_2d_3+d_4c_{(2-3)}+r_4s_{(2-3)}+r_6c_5s_{(2-3)}\\
+c_4s_5c_{(2-3)}
\end{cases} \qquad (5)
$$

II.2.2 Inverse kinematic model

The inverse problem is to calculate the joint coordinates corresponding to a given situation of the end-effector [19]. When it exists, the form that gives all the possible solutions constitutes the one called the inverse kinematic model (IKM).

Three calculating methods can be defined

- The Paul's method.
- The Pieper's method.
- The General method of Raghavan& Roth.

In this case, Pieper's method is suitable for FANUC robot [33]-[34].

- **Calculation of** $\theta_1,\theta_2,\theta_3$

The desired position and the orientation of the end-effector can be written as a homogenous transformation matrix; it can be defined as follows:

$$
U_0=\begin{bmatrix} & & & p_x\\ & {}^{0}A_{E} & & p_y\\ & & & p_z\\ 0 & 0 & 0 & 1 \end{bmatrix} \qquad (6)
$$

Where ${}^{0}A_{E}$ is the (3x3) direction cosine matrix that describes the orientation of the end-effector attached frame and the last column of that matrix presents its position expressed in the arm base frame. U_0 is defined as an elementary matrix multiplication and a matrix that defines the attitude of the attached tool to the end-effector frame. It can be written as

$$
U_0={}^{0}T_{6}.E\Rightarrow U_0.E^{-1}={}^{0}T_{6}
$$

$$
{}^{1}T_{0}.U_0.\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^{T}={}^{1}T_{6}.\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^{T}
$$

$$
{}^{1}T_{6}.\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^{T}={}^{1}T_{4}.\begin{bmatrix}0 & 0 & 0 & 1\end{bmatrix}^{T}
$$

The position of the end-effector is defined just with three first angles. Furthermore, the angles of the wrist give the

end-effector orientation; only using a numerical method under the Matlab Software, the set of the following equations can be presented

$$\begin{cases} c_1 P_x - r_6 c_1 + P_y s_1 = d_2 - d_3 s_2 - d_4 s_{(2-3)} + r_4 c_{(2-3)} \\ r_6 s_1 + c_1 P_y - P_x s_1 = 0 \\ P_z = c_2 d_3 + d_4 c_{(2-3)} + r_4 s_{(2-3)} \end{cases} \quad (7)$$

Where $c_{(2-3)} = cos(\theta_2 + \theta_3), s_{(2-3)} = sin(\theta_2 + \theta_3)$

By using the 2nd expression of (7), θ_1 can be computed with a terms simplification of

$$-s_1 P_x - s_1 r_6 + c_1 P_y = 0$$

Thus, the first angle θ_1 can be written as follows

$$\begin{cases} \theta_1 = ATAN2\left(P_y, P_x - r6\right) \\ \theta_1 = \theta_1 + \pi \end{cases} \quad (8)$$

ATAN2 is the Matlab define function that gives the Argument Arctangent for angle. From a 1st equality of (7), the following equation can be written

$$c_1 P_x - r_6 c_1 + P_y s_1 - d_2 = A$$

Thus, the 1st and 3rdexpression become

$$\begin{cases} A = -d_3 s_2 - d_4 s_{(2-3)} + r_4 c_{(2-3)} \\ P_z = c_2 d_3 + d_4 c_{(2-3)} + r_4 s_{(2-3)} \end{cases} \quad (9)$$

From the 1st expression of (9), $sin(\theta_2)$ is computed as

$$s_2 = \frac{r_4 c_{(2-3)} - d_4 s_{(2-3)} - A}{d_3} \quad (10)$$

In addition, from the 2nd expression of (7), $cos(\theta_2)$ is obtained and can be written as

$$c_2 = \frac{P_z - d_4 c_{(2-3)} + r_4 s_{(2-3)}}{d_3} \quad (11)$$

Therefore,

$$\begin{cases} \theta_2 = a\tan 2(s_2, c_2) \\ \theta_2 = a\tan 2(s_2, c_2) + \frac{\pi}{2} \end{cases}$$

In order to compute θ_3, the following equation can be used

$$\begin{cases} d_3^2 s_2^2 = r_4^2 c_{(2-3)}^2 + d_4^2 s_{(2-3)}^2 - 25 r_4 d_4 c_{(2-3)} s_{(2-3)} + A^2 \\ -2Ar_4 c_{(2-3)} + 2Ad_4 s_{(2-3)} \\ d_3^2 s_2^2 = P_z^2 + d_4^2 c_{(2-3)}^2 - 2P_z d_4 c_{(2-3)} + r_4^2 s_{(2-3)}^2 \\ +2P_z r_4 s_{(2-3)} + 2r_4 d_4 c_{(2-3)} s_{(2-3)} \\ d_3^2 = c_{(2-3)}^2 r_4^2 + d_4^2 + s_{(2-3)}^2 r_4^2 + d_4 + c_{(2-3)} \\ -2Ar_4 - 2P_z d_4 + s_{(2-3)}\ 2Ad_4 - 2P_z r_4 + P_z^2 \end{cases} \quad (12)$$

The simplified expressions can be defined for a simpler computation

$$\begin{cases} X = -2Ar_4 - 2P_z d_4 \\ Y = 2Ad_4 - 2P_z r_4 \\ H = r_4^2 + d_4^2 + P_z^2 - d_3^2 + A^2 \end{cases}$$

In (12), the following is replaced:

$$Xc_{(2-3)} + Ys_{(2-3)} + H = 0$$

The equation (According to $c_{(2-3)}$) of the second degree admits two real solutions when $s_{(2-3)}$=1+ $c_{(2-3)}^2$and, if $\Delta \geq 0$ with:

$$\Delta = \left(2XH\right)^2 - 4\left(X^2 + Y^2\right)\left(H^2 - Y^2\right)$$

Thus

$$\begin{cases} c_{(2-3)} = \dfrac{-2XH \pm \Delta}{2\left(X^2 + Y^2\right)} \\ s_{(2-3)} = 1 + c_{(2-3)}^2 \end{cases}$$

Substituting these results in equations (10) and (11), the values of the second and the third joint angle can be found and written as

$$\begin{cases} \left\{ \theta_3 = ATAN2\left(s_{(2-3)}, c_{(2-3)} - \theta_2\right) \right. \\ \theta_3 = \theta_3 + \dfrac{\pi}{2} \end{cases} \quad (13)$$

The wrist angle values $\theta_4, \theta_5, \theta_6$ will be computed when the previous step is finished. Therefore, the values of the first three joint angles are found, and $\theta_1, \theta_2, \theta_3$, can be substituted in the three according transformation matrix. Furthermore, ${}^0T_3$ matrix is known and the following expression can be written:

$$U_0 = {}^0T_6 \Rightarrow {}^3T_0 U_0 = {}^3T_6$$

$${}^4T_3\ {}^3T_0 U_0 = {}^4T_6 = \begin{bmatrix} & {}^4A_6 & & P \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_6 U_0 = \begin{bmatrix} V_{11} & V_{12} & V_{13} & V_{14} \\ V_{21} & V_{22} & V_{23} & V_{24} \\ V_{31} & V_{32} & V_{33} & V_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$M = {}^4T_3\,{}^3T_0U_0$$

$$M = \begin{bmatrix} c_4V_{13}+s_4V_{33} & -c_4V_{12}-s_4V_{32} & c_4V_{11}+s_4V_{31} \\ c_4V_{33}-s_4V_{31} & -c_4V_{32}+s_4V_{12} & c_4V_{31}-s_4V_{11} \\ -V_{23} & V_{22} & -V_{21} \end{bmatrix}$$

$${}^4A_6 = \begin{bmatrix} c_5c_6 & -c_5s_6 & s_5 \\ s_6 & c_6 & 0 \\ -c_6s_5 & s_5s_6 & c_5 \end{bmatrix}$$

$${}^4A_6(2,3) = 0; M(2,3) = c_4V_{31} - s_4V_{11}$$

By identifying, the following expression can be written

$$c_4V_{31} = s_4V_{11} \Rightarrow \frac{s_4}{c_4} = \frac{V_{33}}{V_{11}}$$

Therefore, the value of θ_4 can be computed as follows

$$\begin{cases} \theta_4 = ATAN2(V_{31}, V_{11}) \\ \theta_4 = \theta_4 + \pi \end{cases} \quad (14)$$

$$\begin{cases} M(3,3) = -V_{21} \\ {}^4A_6(3,3) = c_5 \\ M(1,3) = c_4V_{11} + s_4V_{31} \\ {}^4A_6(1,3) = s_5 \end{cases}$$

$$\theta_5 = ATAN2(M(1,3), M(3,3)) \quad (15)$$

$$M(2,1) = c_4V_{33} - s_4V_{12}$$

$${}^4A_6(2,2) = c_6$$

Thus, the last joint value that consists of the rotation of the wrist can be computed and written as

$$\frac{s_6}{c_6} = \frac{M(2,1)}{M(2,2)}$$

$$\theta_6 = ATAN2(M(2,1), M(2,2))$$

III. MLP Neural Network approach

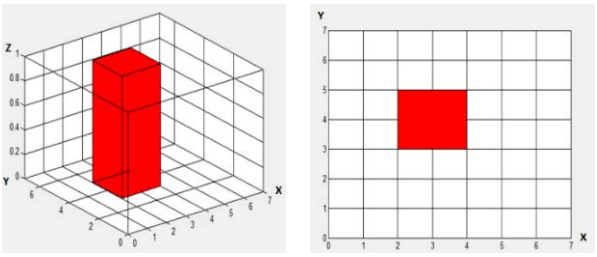
As a highly parallel distributed system, neural network provides a great possibility to solve the problem of high real-time requirements of robot system. Moreover, this system is applied to intelligent autonomous mobile robot navigation and path planning [35]. The developed method to solve the path planning problems uses the neural network for modeling the robot environment and for defining the trajectory that the robot can follow without any collision with obstacles, where the energy function is used. The sum of two energies, the collision energy function (E_c) generated by each obstacle shape and the length trajectory energy function (E_l), is used as the optimization cost function.

The development of the proposed method is as follows. Firstly, a configuration of the robot environment is made; after that, the path planning and the extraction of the trajectory by the Multilayer Perceptron Neural network will be started. Finally, the FANUCend-effector control is programmed.

III.1. *Configuration of the environment*

The manipulator arm is supposed to move in the determinate workspace, and it is described below as

- The robot is assimilated to a particle that moves in a limited two-dimensional space (2D).
- The static obstacle in the workspace is described as polygons (Figures 3.a. and 3.b.).



a b

Figure3: 3-D configuration of the obstacle and its projection in the (x,y) plan

III.2. *Planning and extraction of the robot trajectory*

The Multilayer Perceptron Neural Network (MLP neural network) is used in this paper and it is depicted in Figure 4. The MLP neural network is a feed-forward layered network of artificial neurons, where the data circulates in one way, from the input layer to the output layer. It contains three layers: input, hidden and output. The layers are connected by synaptic weights. The MLP neural network learning process adapts the connections weights in order to obtain a minimal difference between the network output and the desired output. Several algorithms are used for the learning step, but the most used is called back-propagation (BP) algorithm. It consists of four stages: initializing weights, feed forward, back propagation of errors and weight update.

Each obstacle is described by the Multilayer Perceptron Neural Network, and it is represented by a set of linear inequalities, so that the points in the obstacle will satisfy all the inequalities limits.

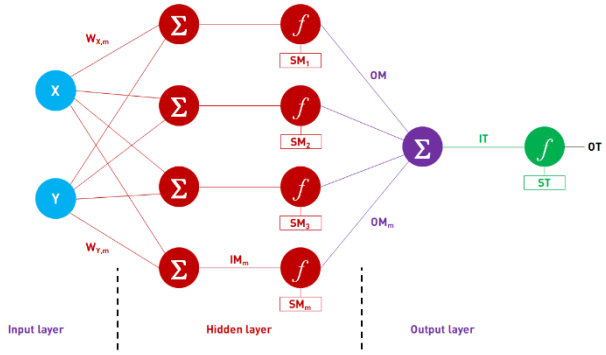


Figure 4: MLP neural network model

Two nodes of the input layer represent coordinates x and y of the given path point. The number of neurons in the hidden layer is equal to the number of obstacle vertices. Connection weight coefficients of the input layer and the hidden layer are equal to the x and y side of the inequality. The coefficient of each node in the hidden layer is equal to the constant term in the corresponding inequality. The connection weight from the hidden layer to the top layer is 1, and the bias of the top node is taken as the number of obstacle vertices decreased by 0.5. The following equation describes this neural network model:

$$\begin{cases} OT = f\left(IT\right) \\ IT = \sum_{m=1}^{M} OM_m + ST \\ OM_m = f\left(IM_m\right) \\ IM_m = w_{x,m}x \ + w_{y,m}y \ + SM_m \end{cases} \tag{16}$$

The sigmoid function is selected as the neuron activation function [36] shown as follows:

$$f = \frac{1}{1+exp(-x/T)} \tag{17}$$

Figure 5 shows the description of the square shape as an obstacle

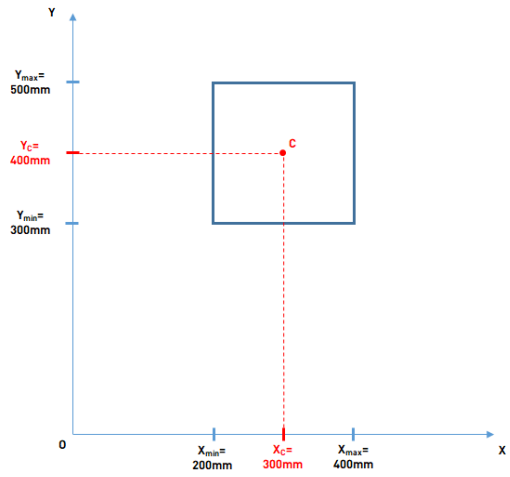


Figure5:Obstacle vertices coordinates(C point)

The point P(x,y) inside the obstacle can be describedwith the following equation

$$\begin{cases} X_{min} < x < X_{max} \\ Y_{min} < y < Y_{max} \end{cases} \Rightarrow \begin{cases} x - X_{min} > 0 \\ x - X_{max} < 0 \\ y - Y_{min} > 0 \\ y - Y_{max} < 0 \end{cases}$$
$$\Rightarrow \begin{cases} x - X_{min} > 0 \\ -x + X_{max} > 0 \\ y - Y_{min} > 0 \\ -y + Y_{max} > 0 \end{cases} \tag{18}$$

The biases of the hidden layer are:

$$\begin{cases} SM_1 = -X_{min} \\ SM_2 = X_{max} \\ SM_3 = -Y_{min} \\ SM_4 = Y_{max} \end{cases} \tag{19}$$

The weights from the input layer to the hidden layer are introduced as

$$\begin{cases} w_{x,1} = 1, w_{y,1} = 0 \\ w_{x,2} = -1, w_{y,2} = 0 \\ w_{x,3} = 0, w_{y,3} = 1 \\ w_{x,4} = 0, w_{y,4} = -1 \end{cases} \tag{20}$$

From equations (19) and (20), IM_m is expressed as

$$\begin{cases} IM_1 = w_{x,1}x \ + w_{y,1}y \ + SM_1 = x - X_{min} \\ IM_2 = w_{x,2}x \ + w_{y,2}y \ + SM_2 = -x + X_{max} \\ IM_3 = w_{x,3}x \ + w_{y,3}y \ + SM_3 = y - Y_{min} \\ IM_4 = w_{x,4}x \ + w_{y,4}y \ + SM_4 = -y + Y_{max} \end{cases} \tag{21}$$

The neural network model corresponding to the example is shown in Figure 6.

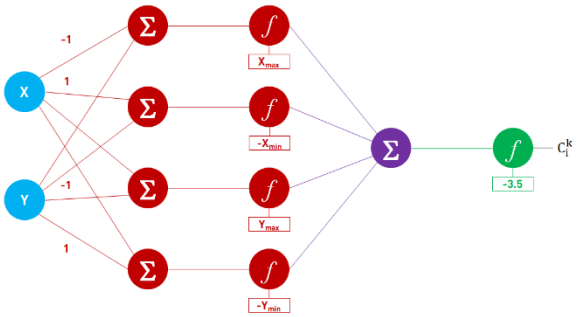


Figure 6: MLP neural network environment model

The collision penalty function of a path is defined as the sum of the collision penalty functions of each path point, and the collision penalty function of the point is obtained by representing each obstacle in its neural network. The neural network model is active when the output value is equal to 1, and if the point P(x,y) is inside the obstacle, otherwise, the output is equal to 0.

The optimization of the trajectory depends on the minimization ofthe cost function, defined by the following energy function

$$E_T = w_l E_l + w_c E_c \tag{22}$$

Where $w_l + w_c = 1$; E_l is the linked energy to the trajectory and it is written as

$$E_l = \sum_{j=1}^{N} l_j^2 = \left(\left(x_j - x_{j-1}\right)^2 + \left(y_j - y_{j-1}\right)^2\right) \tag{23}$$

E_c is given by the following expression

$$E_c = \sum_{j=1}^{N}\sum_{k=1}^{k} C_j^k \tag{24}$$

Minimizing equation (22), the optimal trajectory can be found between two positions (start and goal) without collision and with any obstacles. Thus, it is a simple time derivative of E_T , written as

$$\begin{aligned}\frac{dE_T}{dt} &= \sum_{j=1}^{N}\left[w_l\left(\frac{\partial l_j^2}{\partial x_j}+\frac{\partial l_{j-1}^2}{\partial x_j}\right)+w_c\sum_{k=1}^{k}\frac{\partial C_j^k}{\partial x_j}\right]+\dot{x_j}\\ &+\sum_{j=1}^{N}\left[w_l\left(\frac{\partial l_j^2}{\partial y_j}+\frac{\partial l_{j-1}^2}{\partial y_j}\right)+w_c\sum_{k=1}^{k}\frac{\partial C_j^k}{\partial y_j}\right]+\dot{y_j}\end{aligned} \tag{25}$$

If

$$\begin{cases}\dot{x_j} = -\eta\left[w_l\left(\dfrac{\partial l_j^2}{\partial x_j}+\dfrac{\partial l_{j-1}^2}{\partial x_j}\right)+w_c\displaystyle\sum_{k=1}^{k}\frac{\partial C_j^k}{\partial x_j}\right]\\ \dot{y_j} = -\eta\left[w_l\left(\dfrac{\partial l_j^2}{\partial y_j}+\dfrac{\partial l_{j-1}^2}{\partial y_j}\right)+w_c\displaystyle\sum_{k=1}^{k}\frac{\partial C_j^k}{\partial y_j}\right]\end{cases} \tag{26}$$

Where η is an adaptation gain, equation (25) can be rewritten as:

$$\frac{dE_T}{dt} = -\frac{1}{\eta}\sum_{j=1}^{N}\left(\dot{x_j}^2+\dot{y_j}^2\right)<0 \tag{27}$$

When $\dot{x_j}\to 0$ and $\dot{y_j}\to 0$, E_T converges to its minimum. In other words, when all the points of the trajectory stop moving, there is no collision and the trajectory is the optimal one.

Finally, equation (26) becomes

$$\begin{aligned}\dot{x_j} &= -2\eta w_l\left(2x_j - x_{j-1}+x_{j+1}\right)\\ &-\eta w_c\sum_{k=1}^{k} f'\left(IT_j^k\right)\left(\sum_{m=1}^{M} f'\left(\left(IM_m\right)_j^k\right)w_{xm}^k\right)\\ \dot{y_j} &= -2\eta w_l\left(2y_j - y_{j-1}+y_{j+1}\right)\\ &-\eta w_c\sum_{k=1}^{k} f'\left(IT_j^k\right)\left(\sum_{m=1}^{M} f'\left(\left(IM_m\right)_j^k\right)w_{xm}^k\right)\end{aligned} \tag{28}$$

Where f' is given by the following expression

$$f'(.) = \frac{1}{T} f(.)\left[1-f(.)\right] \tag{29}$$

The first member of equation (28) in the right is for the path length optimization, and the second one, is for the obstacle avoidance.

The extraction of the optimal trajectory is summarized by the following algorithm

Algorithm 1: Optimal trajectory extraction

Start

1: Set start and goal positions of the robot $(x_1,y_1);(x_N,y_N)$ respectively.

2: The coordinates of the points of the initial trajectory (straight line) are described by the following equation:

$$\begin{cases}x_j = x_0 + j\left(x_N - x_1\right)/\left(N-1\right)\\ y_j = \left(y_N - y_1\right)\left(x_j - x_1\right)/\left(x_N - x_1\right)+y_0\end{cases} \tag{30}$$

3: For the points inside the obstacle, the calculation is performed according to equation (28).

4: For the points outside the obstacle, the calculation is continued only to minimize the length of the path, using the following equation:

$$\begin{cases}\dot{x_j} = -\eta w_l\left(2x_j - x_{j-1} - x_{j+1}\right)\\ \dot{y_j} = -\eta w_l\left(2y_j - y_{j-1} - y_{j+1}\right)\end{cases} \tag{31}$$

5: Test the convergence; equation (27) is used as a condition for termination of the calculation.

End

The disadvantage of this method is the computation time, which depends on the number of the points of the initial trajectory. Increasing the number of the points also increases the computation time. In order to solve this problem, the subdivision method is used[23].The principle of this method can be written as the following algorithm

Algorithm 2: The subdivision method

Start

1: Set start and goal positions of the robot $(x_1,y_1);(x_N,y_N)$ respectively.

2: Set mid-point (x_c,y_c) between the start and goal positions.

3: For the mid-point (x_c,y_c) inside the obstacle, the calculation is performed according to the equation (28).

4: Adding a new point in between every consecutive pair of points, the new points are outside obstacle, the calculation is continued using the equation (31).

5: Testing the convergence.

End

For the fifth step of algorithm (2), the following equation can be used

$$e_r \langle error/(1+0.25*p) \tag{32}$$

Where

$$error = \sum_{j=1}^{N}\sqrt{f\left(IT_j\right)^2/N} \tag{33}$$

For p=6, trajectory with 65 points is given.

IV. Results

This paper aims to validate the path planning applied to the 6-Dof robot arm regarding different works as the 7-Dof redundant robot arm used in [38]. Furthermore, this work uses a 6-Dof robot with the development of an inverse kinematics model based on a numerical approach and a neural network method with several layers to cover all the possible cases and the singular positions.

Several works will focus on the shortest path problem of manipulator path planning of 2-Dof robot arm without taking into account the best way to reach a desired position [39]. In this case, the approach presented in this paper deals with several improves of the path planning, with respect to the short time, the best trajectory and the avoidance of singular positions.

Obstacles are usually avoided in the case of manipulator robots when viewing the obstacle in a 3-D environment. The approach presented in this paper will decrease and it deals a 3-D space to the 2-D plane space [40]-[41].

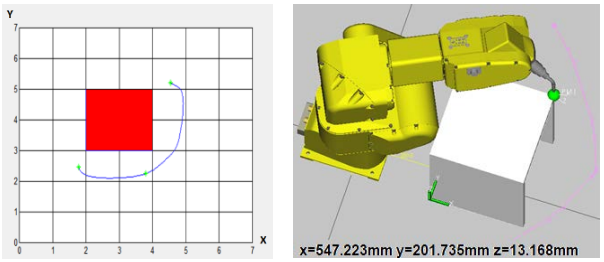
A manipulator arm used is an industrial robot. Its main task is the manufacturing and it is necessary to avoid objects and tools when realizing trajectory missions.

Unlike other systems, from 2-Dof to 5-Dof manipulators used in laboratories for academic research, the presented approach has been implemented on an industrial robot according to restrictive technical specifications.

The main contribution of this work is based on a simple approach implemented in specific dimensions, which can be tested, with any robot in all the dimensions.

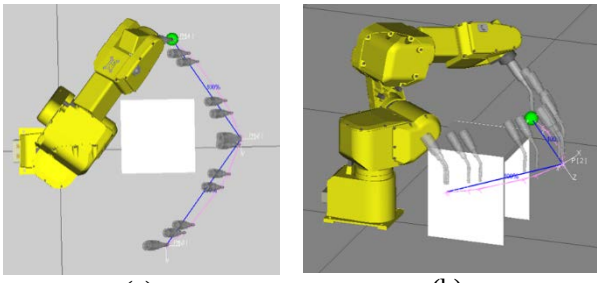
In this section, simulation results obtained by applying the proposed path planning method based on Multilayer Perceptron Neural Network are presented. The first graphical interface is realized by Matlab software [37].Figures 7.a, 7.b show different steps of the execution program. The robot moves inside a small two-dimensional space, the obstacle start and target positions and path planning are marked as red square depicted in Figures 3.a and 3.b, the green points and a smooth blue curve in the Figure 7, respectively. The proposed algorithm is a significant improvement over the classical methods (the visibility graph algorithm and the artificial potential field algorithm), because it finds an optimal path without being trapped in the local minima [17] and the calculation time is reasonably fast. The algorithm can solve the navigation problem in very complex environment and even with static or dynamic obstacles [24]; any obstacle of rectangular, circular, or triangular shape can be represented by the neural network. For the description of the MLP neural network parameters, the algorithm uses only Cartesian coordinates of the obstacles vertices. The effectiveness of the proposed algorithm depends on the value of (T,w_c,w_l,e_r). In this simulation, the number of the points is 65and the path is successfully generated. Figure 7(c) shows that the robot can move from the starting to the target positions without collision with the obstacle. The safe robot distance is respected. The coordinates of the points are recorded in two excel files (vecteurx.xlsx and vecteury.xlsx).

The second graphical interface is realized by Roboguide-FANUC software [42], for the simulation and the control of the virtual FANUC robot with 6-Dof in a 2D environment. The results of the proposed algorithm have been successfully applied to the robot end-effector. The system can read automatically the Excel (vecteurx.xlsx and vecteury.xlsx) files, and then it will be converted into the Inverse kinematic program. Figures 8.a. and 8.b. represent the sequences of the path planning of the FANUC robot with MLP neural networks from the starting point to the target point.



(a) (b)

Figure 7: Different steps of the program execution



(a) (b)

Figure 8: Virtual prototype of the proposed approach

V. Conclusion:

In this paper, a new path planning method based on Multilayer Perceptron Neural Network is proposed. The architecture of the neural network model is constructed from equations that define a point inside the obstacle, and establish a relationship between the degree of collision and the out-put of the model. This model has given the desired solution to solve the path planning problems. The proposed method has been validated by several simulation tests applied to the FANUC robot. The simulation results are verified by two research laboratories (L.E.P.E.S.A-Algeria, IBISC-France); both have demonstrated the feasibility and the effectiveness of the presented method to solve the path planning problem. The path obtained is optimal (the shortest trajectory) with a safe distance between the robot and the obstacle. The obtained results have encouraged authors to deepen the study and think about 3D path planning in the future, by adding an additional input for the z dimension to the obstacle description on MLP neural network.

REFERENCES

[1] BadrElKari, Hassan Ayad, Abdeljalil El Kari, Mostafa Mjahed, A New Approach of Fusion Behavior-Based Fuzzy Control for Mobile Robot Navigation,*International Review of Automatic Control, Vol.10, N°1,2017*

[2] F. Bouchiba, W.Nouibet, Neuro-Fuzzy Navigation of a Mobile Robot in a Unknown Environment ,*International Review of Automatic Control, Vol.8, N°3, pp. 220-227,2015.*

[3] Rosell Jan, and Pedro Inguez, Path planning using harmonic functions and probabilistic cell decomposition, *Proceedings of the 2005 IEEE International Conference on Robotics and Automation, IEEE 2005.*

[4] Šeda, Miloš, Roadmap methods vs. cell decomposition in robot motion planning, *Proceedings of the 6th WSEAS international conference on signal processing, robotics and automation. Word Scientific and Engineering Academy and Society (WSEAS), 2007.*

[5] E.Galceran and M.Carreras, "A survey on coverage path planning for robotics'', *Rob.Auton. Syst.,vol.61, n°.12, pp.1258-1276,2013.*

[6] R.S.Pol and M.Murugan, "A review on indoor human aware autonomous mobile robot navigation through a dynamic environment survey of different path planning algorithm and methods'', *2015 Int.Conf.Ind.Instrum.Control.ICIC 2015, n°.July, pp. 1339-1344,2015.*

[7] H.Bharadwaja, Vinodh Kumar, "Comparative study of neural networks in path planning for catering robots'', *Procedia Computer Science 133 (2018),pp.417-423,2018.*

[8] C.Li and J.Zhang, "Application of Artificial Neural Network Based on Q-learning for Mobile Robot Path planning'', Sci.Technol., pp.978-982,2006.

[9] L.Ren, W.Wang, and Z.Du, "Intelligent Obstacle Avoidance Control Strategy for Wheeled Mobile Robot'', *ICROS-SICE Int, Jt.Conf.2009, pp.3199-3204, 2009.*

[10] FA Cosío, MA Padilla Castañeda, Autonomous robot navigation using adaptive potential fields, *Mathematical and computer modelling 40.9-10 (2004):1141-1156.*

[11] García, Néstor, Jan Rossel, and RaúlSuárez, Motion planning bu demonstration with human-likeness evaluation for dual-arm robots, *IEEE Transactions on Systems, Man, and Cybernetics: Systems (2017).*

[12] Sami Haddadin, Rico Belder, and AlinAlbu-Schäffer, Dynamic motion planning for robots in partially unknown environments, *IFAC Proceedings Volumes 44.1 (2011): 6842-6850.*

[13] Friudenberg, Patrick, and Scott Koziol, Mobile robot rendezvous using potential fields combined with parallel navigation,*IEEE Acess 6 (2018): 16948-16957.*

[14] R.Salgado, et al., Motivational engine with autonomous sub-goal identification for the Multilevel Darwinist Brain, *Biologically Inspired Cognitive Architectures 17 (2016):1-11.*

[15] Praveena, Pragathi, et al., Unser-guided offline synthesis of robot arm motion from 6-dof paths:*2019 International Conference on Robotics and Automation (ICRA), IEEE, 2019.*

[16] Ichter, Brian, James Harrison, and Marco Payone, Learning sampling distributions for robot motion planning, *2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE, 2018.*

[17] Xin, D.U., Chen Hua-hua, and Gu Wei-kang, "Neural network and genetic algorithm based global path planning in a static environment'', *Journal of Zhejiang University-Science A 6.6 (2005):549-554.*

[18] Zelinsky, Alexander, "A unified approach to planning, sensing and navigation for mobile robots'', *Experimental Robotics III. Springer, Berlin, Heidelberg, 1994, pp. 444-455.*

[19] Warren, Charles W., "Fast path planning using modified A*method'', *[1993] Proceedings IEEE International Conference on Robotics and Automation, IEEE, 1993.*

[20] Kritikou, Georgia, Nikos Lamprianidis, and Nikos Aspragathos, "A Modified Cooperative A* Algorithm for the Simultaneous Motion of Multiple Microparts on a "Smart platform'' with Electrostatic Fields'', *Micromachines 9.11 (2018):548.*

[21] Abderrahmane, M. S., et al., "Study and validation of singularities for a FANUC LR MATE 200iC robot", *IEEE International Conference on Electro/Information Technology*. IEEE, 2014.

[22] Jin, Long, Shuai Li, Jiguo Yu, and Jinbo He. "Robot manipulator control using neural networks: A survey." Neurocomputing 285 (2018): 23-34.

[23] L. Bouhalassa, Magister dissertation, *path planning of the manipulator arm based on the neural network and the genetic algorithm*, Dept. Elect. Eng., University of Science and Technology-Algeria, November 2010.

[24] FarhadShamsfakhr, Bahram SadeghiBigham, A neural network approach to navigation of a mobile robot and obstacle avoidance in dynamic and unknown environments, *Turkish Journal of Electrical Engineering & Computer Sciences, 25:1629-1642, 2017*

[25] NgaT.T.Vu, Do D.Bui, and HieuT.Tran, Artificial neural network based path planning of excavator arm, *International Journal of Mechanical Engineering and Robotics Research,Vol.8, N°1, January 2019*

[26] Long Jin, ShuaiLi,JiguoYu,JinboHe,Robot manipulator control using neural networks: A survey,*Neurocomputing, Volume 285:pp.23-34,2018*

[27] OuamriBachir, Ahmed-FoitihZoubir, Adaptive Neuro-fuzzy Inference System Based Control of Puma 600 Robot Manipulator, *International Journal Of Electrical and computer Engineering (IJECE), Vol.2, N°1. pp.90-97, February 2012.*

[28] SwadhinSambit Das, DayalR.Parhi, SuranjanMohanty,insight of a six layered neural network along with other al techniques for path planning strategy of a robot,*Emerging trends in engineering, Science and Manufacturing (ETESM), 2018.*

[29] FANUC LR MATE 200iB Retrieved from (http://www.FANUC.eu/robots).

[30] K. Bouzgou, Z.AFoitih, Workspace analysis and geometric modeling of 6 dofFANUCLR MATE 200ic robot, *Procedia-Social and Behavioral Sciences 182 (2015): 703-709.*

[31] Constantin, Daniel, et al., "Forward kinematic analysis of an industrial robot'', *Proceedings of the International Conference on Mechanical Engineering (ME 2015), 2015.*

[32] W.Khalil, E.Dombre, *modeling, identification and control of robots* (2°édition, 1999).

[33] K. Bouzgou,Z.Ahmed-Foitih, Geometric modeling and singularity of 6d of FANUC 200iC robot, IEEE Transactions on Innovative Computing Technology (INTECH), pp. 208–214, Luton, UK, October 2014.

[34] B. Ibari,K. Bouzgou, Z. Ahmed-Foitih and L. Benchikh, Design and Manipulation in Augmented Realty of 200 iC Robot,*Journal of intelligent computing, Vol. 6, (Issue 3):84-91,September 2015*

[35] Y Tong,J Liu,Y Liu and Z Ju, Improved neural network 3D space obstacle avoidance algorithm for mobile robot. *International Conference on Intelligent Robotics,2019.*

[36] B. Karlik, A.V.Olgac , Performance Analysis of Various Activation Functions in Generalized MLP Architectures of Neural Network,*International Journal of Artificial Intelligence and Expert, Vol.1,N°4:111-122,February 2011.*

[37] Ahmed R.J.Almusawi, L.CanaDülger, and sadettinKapucu, A new artificial neural network approach in solving inverse kinematics of robotic arm (denso VP6242),Computational intelligence and neuroscience,2016.

[38] Klanke, S., Lebedev, D., Haschke, R., Steil, J., & Ritter, H. (2006, October). Dynamic path planning for a 7-dof robot arm. In *2006 IEEE/RSJ international conference on intelligent robots and systems* (pp. 3879-3884). IEEE.

[39] Li, Yan, and Keping Liu. "Two-degree-of-freedom manipulator path planning based on zeroing neural network." In *MATEC Web of Conferences*, vol. 309, p. 04005. EDP Sciences, 2020.

[40] Cheng, M. H., Huang, P. L., Chu, H. C., & McKenzie Jr, E. A. (2019, November). Motion Estimation and Path Planning for Assistive Robotic Devices. In *ASME International Mechanical Engineering Congress and Exposition* (Vol. 59407, p. V003T04A073). American Society of MechanicalEngineers.

[41] Vu, Nga TT, Do D. Bui, and Hieu T. Tran. "Artificial Neural Network Based Path Planning of Excavator Arm." *International Journal of Mechanical Engineering and Robotics Research* 8, no. 1 (2019).

[42] FANUC Roboguide, Retrieved from (http://www.fanuc.eu).

Authors' information

[1,3,4]Laboratory of Power Electronics, Solar Energy and Automation (LEPESA), Department of Electronics, University of Science and Technology, Oran-Algeria.

[2,4]Laboratory of Computer Science, Integrative Biology and Complex Systems (IBISC),University of Evry Val d'Essence-France.

BouhalassaLoubna, born on 03 September 1982 Oran, Algeria, received the Dipl-Ing degree in Electronics and the Magister degree in Automatic, Robotic, Productive (ARP), both from de University of Science and Technology Of Oran, Algeria, in 2006 and 2010, respectively, currently I prepare the Ph.D.dissertation in Automatic, Robotic, Productive (ARP) at the same university.

E-mail: bouhalassa.loubna@gmail.com.

DrBenchikh Laredj, Professor assistant at Evry Val d'Essonne University, Campus Paris-Saclay (France). Main research focus Human-Robot Interfaces, Intelligent Robots, Aerial Robots, Robotics and Automation. FANUC Robotics expert with 20 years experience on Research and Development of Industrial Projects and author of 3 patents.

E-mail: laredj.benchikh@univ-evry.fr

Ahmed-FoitihZoubir,Has received the Engineering Degree in Electronics in 1980, the Magister in Automatic in 1988, and Ph.D. degree in 2004, from de University of Sciences and Technology Of Oran USTO (Oran, Algeria). Since 1982 he is a lecturer and researcher member of the Electronics Department (U.S.T.O). Since 2012, he is Professor at the USTO Oran and head team Teleoperation, Automation and Supervisory of laboratory of power electronic, solar energy and automation, team which is implied in several French and European projects. He is currently a Full Professor with the University of Sciences and Technology of Oran. His current research interests include Robotics, Computer automation, control of industrial processes and supervision, Identification of process, soft computing.

E-mail: zoubir.foitih@univ-usto.dz

Kamel Bouzgou,received his master's degree at the electronics department of University of Sciences and Technology of Oran Mohamed-Boudiaf, Algeria and started his Ph.D in 2014 in the same university. Since 2017 he joined IBISC Laboratory and performing his second PhD studies in at University Paris-Saclay, Evry Val d'Essonne University working towards for a new project. His research interests include the aerial manipulation, design and control of mechanical systems, nonlinear and hybrid systems modeling and applications.

E-mail: kamel.bouzgou@univ-evry.fr