The Inverse Kinematics Evaluation of 6-DOF Robots in Cooperative Tasks Using Virtual Modeling Design and Artificial Intelligence Tools

Abderrahim Bahani, Moulay El houssine Ech-Chhibat, Hassan Samri, and Hicham Ait Elattar

Hassan II University of Casablanca, Mohammedia, Morocco; Emails: Echchhibate@gmail.com; samrih127@gmail.com; Aitelattar.hicham@gmail.com
*Correspondence: abdeer.bahani@gmail.com

***Abstract*—This work aims at evaluating the inverse kinematics of a two-robot cooperative system using Matlab/SimMechanics-based simulations and artificial intelligence tools, namely the Levenberg-Marquardt (LM) optimization method. The artificial neural networks (ANN) thus constructed will replace the controllers of the six degrees of freedom (6-DOF) cooperative robots. Therefore, the entire cooperative system was designed in SolidWorks, taking into account all the dimensions necessary for kinematic modeling, then converted into Matlab/SimMechanics, and thanks to the manipulation of the model in this software, we will be able to extract the articulatory and operational data of the cooperative system in its workspace. The kinematic database of the robotic system is built in Matlab in order to train the ANN and implement it in Matlab/SimMechanics. Lastly, a test is performed in a collaborative task to evaluate the intelligent control error. The results obtained can be applied to and tested for the kinematic control of two real ABB IRB 120 cooperative robots.**

***Keywords*—inverse kinematic, cooperative system, Matlab/SimMechanics, Artificial Neuronal Network, Levenberg Marquardt, 6-DOF, Robot IRB 120**

I. Introduction

In modern robotics, the topic of multi-robot cooperation is very broad and covers a wide range of domains and applications where a group of robots can cooperate to perform a common task. Cooperative industrial robotics has several applications, including transportation, autonomous military systems, and industrial manufacturing. Fig. 1 depicts the use of cooperative robotics in a welding operation.

When two or more industrial manipulators are used to grasp or manipulate an object, the motion trajectory and then the internal force of the manipulated object must be controlled [1]. In this paper, we attempt to track a cooperative robot's trajectory using not only simulation and optimization software but also artificial intelligence tools. To reach to at very convincing conclusions, we first investigate the kinematics of the cooperative robotic system, taking into account the state of the art in geometric and kinematic modeling of a single 6-DOF robot. We then created the kinematic database using the SimMechanics model, and finally, using the Matlab library, we could build an intelligent controller that would be validated by the SimMechanics simulation.

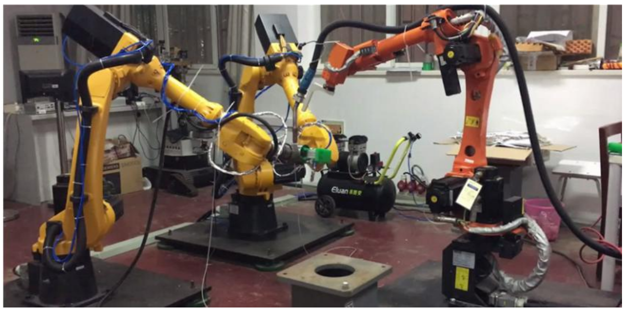

Figure 1. Example of multi-robot cooperation in a welding task [1].

II. Related Works

The issue of cooperative path planning in a multi-robot system without planning tools is covered in research [1] and [2]. Other studies, like [3] investigate cooperative item manipulation tasks for two-armed robots. The sliding mode controller, based on the hyperbolic tangent function, is used in these tasks.

The study [4] proposes a methodology for developing a web-based graphical user interface for motion simulations of the industrial robot PUMA 560. The same interface can then be connected to a real system using the web-based communication architecture to control the real PUMA 560 robot remotely. This methodology is undeniably effective for educational purposes, but it lacks artificial intelligence tools for controlling robots.

In this paper, we focus mainly on the development of a new model of cooperative robots manipulating a common object. This model is designed to overcome the problem of inverse kinematics of cooperative manipulator arms because this type of work involves several geometric, kinematic, and dynamic constraints. The method consists of modeling with CAD tools, followed by simulation with MATLAB/SimMechanics, a technique previously introduced by [5]. The method of setting up a trajectory

Manuscript received October 13 2022; revised November 18 2022; accepted December 28 2022.

doi: 10.18178/ijmerr.12.2.121-130

for a cooperative system is given in [6], which suggests a method based on the closed kinematic chain model. The velocity and acceleration constraints caused by the kinematic closure in the case where two robots grasp a common object are deduced in [7] and [8]. In addition to the analytic modeling, we have tried to control our cooperative system with the help of artificial networks, a method used by [9]. The database of these networks is derived from CAD computer modeling after being uploaded to SimMechanics. Certainly, the research on inverse kinematics of robots has already been done [10], but they did not take into consideration the tool orientation other than when they built the database of networks by the conventional method based on the analytical approach. In contrast to our work based on the reunion from the "Workspace" following a Matlab/SimMechanics simulation.

The development of an artificial intelligence approach to the task of manipulator robot trajectory planning has already been discussed in [11]. While [12] is concerned with the matrix method of a pseudo-inverse neural network with a correct Jacobian applied to the control of a robot's inverse kinematics.

The cooperative system is supposed to be composed of two identical robots; that's why the configuration of only one robot will be enough to describe the cooperative system.

III. Kinematics of the IRB 120 Robot

The multi-purpose industrial robot, IRB-120, has been introduced by ABB. This 6-DOF robotic manipulator has all the capabilities and advanced design features of ABB's large robot but is very lightweight and cost-effective. It has a mass of only 25 kg, an accessibility of 580 mm, and the ability to reach 112 mm under its own base. It has a rotating arm structure, a spherical wrist structure, and a structure similar to that of the PUMA-560 robot. The important aspects of the IRB-120 robot and the range of variation of each joint axis are shown in Fig. 2.

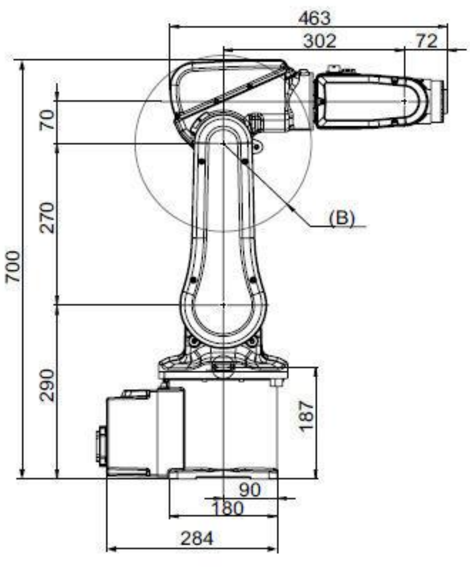

Figure 2. Structure of the manipulator robot IRB-120 [13].

A. Geometric Model

The computation of the orientation and position of the robot end is called a "direct kinematic analysis". In order to model a direct kinematic analysis for the robot, there is generally a methodology to follow. First, number the links and joints and attach local coordinate reference systems to each link. Then establish the Denavit-Hartenberg (D-H) parameters for each link. Direct kinematic analysis of the IRB-120 robot manipulator (6-DOF) Fig. 3 shows the D-H parameters for the IRB-120. The D-H for the IRB-120 listed in Table I are ai (link length), αi (link twist), di (link offset), and θi (joint angle).

TABLE I. The D-H Parameters of ABB IRB 120.

Joint i	D-H *parameters*			
	θi/(°)	di/mm	ai/mm	αi/(°)
1	θ1	290	0	-90
2	θ2 – (pi/2)	0	270	0
3	θ3	0	70	-pi/2
4	θ4	302	0	pi/2
5	θ5	0	0	-pi/2
6	θ6 +pi	72	0	0

Fig. 3 is a schematic view of the coordinate systems, joints, joint angles, articulations, and centers of mass of the manipulator robot IRB-120. Since all joints of the IRB-120 robot are rotating, the robot mechanism is 6R (rotating).

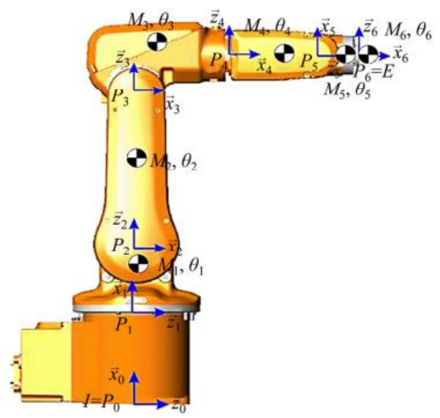

Figure 3. The centers of mass and coordination of an IRB 120 [13].

Given the parameters in Table I, the transformation matrices for the links of the robot are obtained as follows:

$$T_{i,i+1} = \begin{bmatrix} \cos\theta_i & -\sin\theta_i.\cos\alpha_i & \sin\theta_i.\sin\alpha_i & L_i.\cos\theta_i \\ \sin\theta_i & \cos\theta_i.\cos\alpha_i & -\sin\theta_i.\cos\alpha_i & L_i.\sin\theta_i \\ 0 & \sin\beta & \cos\alpha & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

The transformation matrices for six axes are given in Eq. (2) :

$$\begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = T_{0,1}.T_{1,2}.T_{2,3}.T_{3,4}.T_{4,5}.T_{5,6} \quad (2)$$

where n (normal), o (orientation), and a (approach) are orientation elements, and P (position) is a position element relative to the reference frame [13]. The elements of the matrix on the left-hand side of Eq. (2) are given in the following equations:

$$n_x = -S_6(C_4S_1 + S_4(C_1C_2C_3 - C_1S_2S_3)) - C_6(C_5(S_1S_4 - C_4(C_1C_2C_3 - C_1S_2S_3)) + S_5(C_1C_2S_3 + C_1C_2S_2)) \tag{3}$$

$$n_y = S_6(C_1C_4 + S_4(S_1S_2S_3 - C_2C_3S_1)) + C_6(C_5(C_1S_4 - C_4(S_1S_2S_3 - C_2C_3S_1)) - S_5(C_2S_1S_3 + C_3S_1S_2)) \tag{4}$$

$$n_z = C_6(S_5(C_2C_3 - S_2S_3) + C_4C_5(C_2S_3 + C_3S_2)) - S_3S_6(C_2S_3 + C_3S_2) \tag{5}$$

$$o_x = S_6(C_5(S_1S_4 - C_4(C_1C_2C_3 - C_1S_2S_3)) - S_5(C_1C_2S_3 + C_1C_3S_2)) - C_6(C_2S_1 + S_4(C_1C_2C_3 + C_1S_2S_3)) \tag{6}$$

$$o_y = C_6(C_1C_4 + S_4(S_1S_2S_3 - C_2C_3S_1)) - S_6(C_5(C_1S_4 - C_4(S_1S_2S_3 - C_2C_3S_1)) - S_5(C_2S_1S_3 + C_3S_1S_2)) \tag{7}$$

$$o_z = -S_6(S_5(C_2C_3 - S_2S_3) + C_4C_5(C_2S_3 + C_3S_2)) - C_6S_4(C_2S_3 + C_3S_2) \tag{8}$$

$$a_x = S_5(S_1S_4 - C_4(C_1C_2C_3 - C_1S_3S_2)) - C_5(C_1C_2S_3 + C_3C_1S_2 - C_1C_3S_2) \tag{9}$$

$$a_y = -S_5(C_1S_4 - C_4(S_1S_2S_3 - C_1S_3C_2)) - C_5(S_1C_2S_3 + C_3C_1S_2 - S_1C_3S_2) \tag{10}$$

$$a_z = C_5(C_2C_3 - S_3S_2) - S_5C_4(S_3C_2 + C_3S_2) \tag{11}$$

$$P_x = d_6(S_3(S_1S_4 - C_4(C_1C_2C_3 - C_1S_3S_2)) - C_5(C_1C_2S_3 + C_3C_1S_2)) - d_4(C_1C_2S_3 + C_1C_3S_2) + a_2C_1C_2 + a_3C_1C_2C_3 - a_3C_1S_2S_3 \tag{12}$$

$$P_y = -d_6(S_5(C_1S_4 - C_4(S_1S_2S_3 - S_1C_3C_2)) + C_5(S_1C_2S_3 + C_3S_1S_2)) - d_4(S_1C_2S_3 + S_1C_3S_2) + a_2S_1C_2 + a_3S_1C_2C_3 - a_3S_1S_2S_3 \tag{13}$$

$$P_z = d_1 + d_6(C_5(C_3C_2 - S_3S_2) - C_4S_5(C_2S_3 + C_3S_2)) + d_4(C_2C_3 + S_3S_2) + a_2S_2 + a_3C_2S_3 + a_3C_3S_2 \tag{14}$$

Where, S_i and C_i are the representatives of $\sin(\theta_i)$,and $\cos(\theta_i)$ respectively.

B. *Inverse Kinematic Analysis of the IRB-120 6 DOF Manipulator Robot*

The existence of a solution to the equation X = f(q) is conditioned by the fact that the end effector evolves in the reachable domain. This domain is defined, on the one hand, by the dimensional limitations of the mechanical elements forming the vector X and, on the other hand, by possible structural limitations. The resolution of the inverse kinematics of this robot leads us to:

For θ_1 , the two solutions are:

$$\begin{cases} \theta_{1.1} = a\tan(y_p, x_p) \\ \theta_{1.2} = a\tan((-y_p, -x_p) \end{cases} \tag{15}$$

For θ_3 , the two solutions are:

$$\begin{cases} \theta_{3.1} = a\tan 2(t, \sqrt{1-t^2}) + \beta \\ \theta_{3.2} = a\tan 2(t, -\sqrt{1-t^2}) + \beta \end{cases} \tag{16}$$

With :

$$\begin{cases} \sin(\theta_3 - \beta) = t \\ \beta = \ atan2(70.302) \end{cases} \tag{17}$$

calculation of θ_2 :

$$\theta_2 = a\tan 2(q_1q_3 + q_4q_2, q_2q_3 + q_1q_4) \tag{18}$$

with

$$q_1 = \ -302S_3 + \ 70C_3 + 270 \tag{19}$$

$$q_2 = \ 302C_3 + \ 70S_3 \tag{20}$$

$$q_4 = \ -z_p \tag{21}$$

calculation of θ_5 :

$$\theta_{5.1} = a\tan 2(\sqrt{1- \ (-a_zS_{23} + (C_1a_x + S_1a_y)C_{23})^2}, -a_zS_{23} + (C_1a_x + S_1a_y)C_{23}) \tag{22}$$

$$\theta_{5.1} = a\tan 2(-\sqrt{1- \ (-a_zS_{23} + (C_1a_x + S_1a_y)C_{23})^2}, -a_zS_{23} + (C_1a_x + S_1a_y)C_{23}) \tag{23}$$

For θ_4 and θ_6

$$\theta_4 = a\tan 2\left(\frac{-S_1a_x + C_1a_y}{S_5}, \frac{-(C_1a_x + S_1a_y)S_{23} - a_zC_{23}}{S_5}\right) \tag{24}$$

$$\theta_6 = a\tan 2\left(\frac{-S_{23}a_x + (C_1o_x + S_1o_y)C_{23}}{S_5}, \frac{S_{23}n_z + (C_1n_x + S_1n_y)C_{23}}{S_5}\right) \tag{25}$$

Where, $C_{23} = \cos(\theta_2 + \theta_3)$ and $S_{23} = \sin(\theta_2 + \theta_3)$ are defined contractually.

IV. Kinematic Analysis of Two Cooperative Robots

A. *Preliminaries*

When serial robots manipulate an object in common, they form a closed chain, which makes the multi-robot system an overdriven or redundant system, since the effective degrees of freedom (ne) are greater than the strict degrees of freedom (nt). This ability enhances the dexterity of the mechanism, which offers advantages such as the ability to avoid joint restrictions, singularities, and obstructions in the work area, but the mechanism also makes trajectory planning and kinematic control difficult [14].

B. Combinative Kinematics Model of the Dual-Arm Robots

The multi-robot system that manipulates a common-object has a different combinatorial kinematic model than the traditional multi-degree-of-freedom serial manipulators. The diagram shown in Fig. 4 of the two-arm cooperative system is adopted.

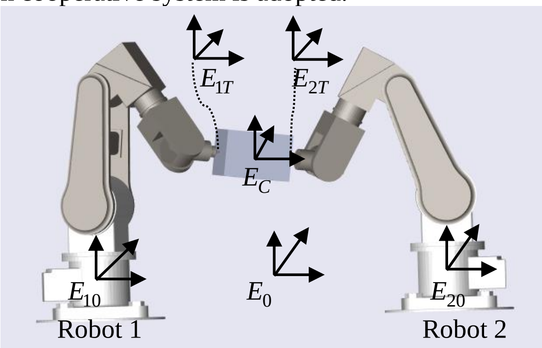

Figure 4. Cooperating robot arms carrying a rigid object.

The kinematic closure requirements and the pose constraint between the two-robot system's effectors impose the following structural relationships [3].

$$\begin{cases} T_{0,C} = T_{0,10}.T_{10,1T}.T_{1T,C} \\ T_{0,C} = T_{0,20}.T_{20,2T}.T_{2T,C} \end{cases} \tag{26}$$

Where

- $T_{0,C}$ denotes the pose transformation matrix of the object with respect to the world coordinate system E_0.
- $T_{0,10}$ and $T_{0,20}$ denotes the pose transformation matrices of the base coordinate systems of robot 1 and robot 2 with respect to the world coordinate system, respectively.
- $T_{10,1T}$ and $T_{20,1T}$ denotes the pose transformation matrix of the end effectors (TCP) with respect to the base coordinate system, respectively.
- $T_{1T,C}$ and $T_{2T,C}$ denote the pose transformation matrix of the object with respect to the coordinate systems of the two-armed robot end effectors, respectively.

The kinematics equations for the manipulators can therefore be determined as follows:

$$\begin{cases} T_{10,1T} = T_{10,0}.T_{0,C}.T_{C,1T} \\ T_{20,2T} = T_{20,0}.T_{0,C}.T_{C,2T} \end{cases} \tag{27}$$

We assume that the system has two 6-DOF-style manipulator arms. The transformation matrix is expressed as follows:

$$T_{0,1T} = \begin{bmatrix} R_{0,1T} & p_{0,1T} \\ 0 & 0 \end{bmatrix} \tag{28}$$

Where $R_{0,1T} \in \mathbb{R}^{3\times3}$ and $p_{0,1T} \in \mathbb{R}^{3\times1}$ are a rotation matrix and a position vector, for robot 1, of the end-effector frame E_{1T} relative to the base frame E_0, respectively.

The orientation vector of the end-effector frame E_{1T} relative to the base frame E_0 ; $\phi_{01T} \in \mathbb{R}^{3\times1}$ can be calculated by this rotation matrix.

These matrix equations can result in six equations that represent the system's kinematic closure [14].

$$p_{0,C} = p_{0,1T} + p_{1T,C} = p_{0,2T} + p_{2T,C} \tag{29}$$

$$\phi_{0,C} = \phi_{01,T} + \phi_{1T,C} = \phi_{0,2T} + \phi_{2T,C} \tag{30}$$

With :

- $p_{1T,C} \in \mathbb{R}^{3\times1}$ and $\phi_{1T,C} \in \mathbb{R}^{3\times1}$ represents the object frame's E_C position and orientation with respect to the first arm's end-effector frame E_{1T}, respectively.
- $p_{0,C} \in \mathbb{R}^{3\times1}$ and $\phi_{0,C} \in \mathbb{R}^{3\times1}$ are the common object's position and orientation vectors with respect to the basic framework E_0.

The vector can be used to express the position of arm 1's end effector; $x_1 = \left[p_{01}{}^T, \phi_{01}{}^T\right]^T \in \mathbb{R}^{6\times1}$. Similarly, the vector $x_2 = \left[p_{02}{}^T, \phi_{02}{}^T\right]^T \in \mathbb{R}^{6\times1}$ described arm 2, so we can join the two expressions by $x = \left[x_1{}^T, x_2{}^T\right]^T \in \mathbb{R}^{12\times1}$.

And by putting $q = \left[q_1{}^T, q_2{}^T\right]^T \in \mathbb{R}^{12\times1}$ is the two-armed robots' joint position vector. The following relationship can be written thanks to the kinematic analysis:

$$\dot{x} = J(q)\dot{q} \tag{31}$$

Deriving the above equation with respect to time t we can obtain:

$$\ddot{x} = J(q)\ddot{q} + \dot{J}(q)q \tag{32}$$

The arm robots' joint acceleration's control vector can be determined as follows:

$$\ddot{q} = \left[J(q)\right]^{-1}(\ddot{x} - \dot{J}(q)\dot{q}) \tag{33}$$

C. Cooperative Control Strategy for the Robots

When the two robots handle a rigid object in common, the task of path planning becomes very complicated because the object and the two end effectors form a closed set [14]. This is why we insert a certain stiffness at the level of the tool-object contact, as shown in the following Fig. 5 [15].

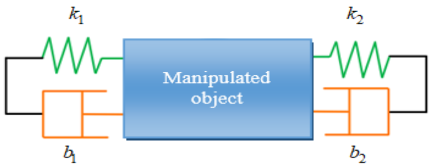

Figure 5.The tool-object contact modeling

D. Simplification in Our Case

Finding the values of the joint variables in relation to the center of a common object in the pose is the definition of the inverse kinematics problem for cooperative manipulators.

In our case, we coincide the global reference E_0 with the reference of the first robot. in this situation the equations become:

$$\begin{cases} T_{0,C} = T_{10,1T} \cdot T_{1T,C} \\ T_{10,1T} = T_{0,C} \cdot T_{C,1T} \end{cases} \tag{34}$$

Furthermore, we assume that the object is fixed with respect to the end effector. This corresponds to a stiffening of the two final object-effector contacts ($k_1 \sim \infty$ and $k_2 \sim \infty$) as well as a, the b-damping ($b_1 \sim \infty$ and $b_2 \sim \infty$), all to avoid error messages during the simulation.

V. Robots Cooperative Design and Controlling Tool in Matlab

Today, simulation and CAD modeling tools are essential for the construction of a robotic system. Among these tools, we find of all the modeling software SolidWorks, which is used to design, simulate, manage data, and even evaluate the impact of products on the environment. We use this software to design the two cooperative robots to build a real model; we are obliged to respect the dimensions and the mechanics of each arm, and then we insert the object to be manipulated. The following Fig. 6 shows the proposed cooperative system, which consists of the two manipulator robots in the Matlab/SimMechanics environment.

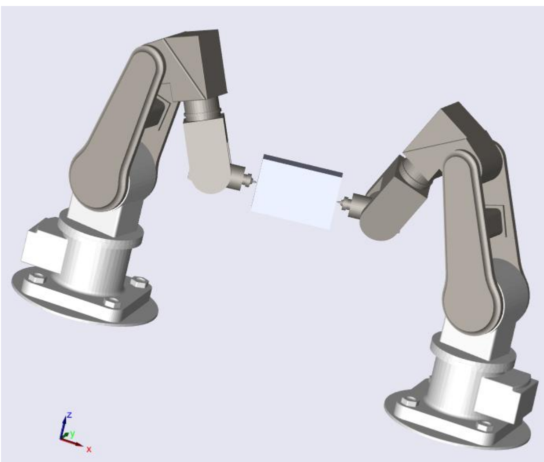

Figure 6.Virtual model of the dual-arm system

A. Synchronization of SolidWorks Software with MATLAB/SimMechanics

SolidWorks is a CAD tool that offers the possibility to synchronize the designed model with other software, not only for design but also with calculation or control software such as MATLAB. All Design elements, including the mechanical and geometric characteristics of the structure built using SolidWorks' assembly function, are synchronized [5].

B. SimMechanics Module

The generation-first model is then obtained by synchronizing SolidWorks and Matlab. This model is presented as an XML file that contains the models and the characteristic parameters of our cooperative system. Once generated, this file will be able to be read in Matlab/ Simulink, as shown in Fig. 7.

C. Model Manipulation and Data Collection for Neural Networks

The opening of the XML file in Matlab allows us to manipulate our system by deleting or adding elements from the Simscape library [16]. We proceed to manipulate the object by adding motor commands to the file, which allows us to move the object in all directions and orientations while maintaining object-arm contact in order to collect the articular coordinates of each cooperative robot. Fig. 7 illustrates the collection strategy for articular coordinates based on object coordinates:

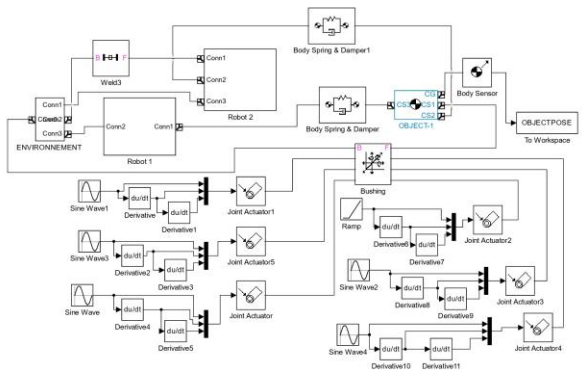

Figure 7. The Simulink diagram for the sampling collection.

VI. Proposed Artificial Neural Network Design

A. Preliminaries

Artificial neuronal networks are methodical computing systems that process information without tiring of the structure of the cervix.

Hard-limit features, linear features, and sigmoid features are all common. The output of the activation function can be connected to the input of another neuron. The weights and the biases are formed to minimize the errors between the desired outputs and the actual outputs [17], as illustrated in Fig. 8.

Training is a learning process to update weights and biases. Site weight and bias play an important role in artificial neuronal networks. The weights determine the position in the entry space, and without bias, the entries are forced to go through the origin of the entry space. the origin of the entry space [18, 20, 21].

$$a = F(\sum_{j=1}^{n} w_j p_j + b) \tag{35}$$

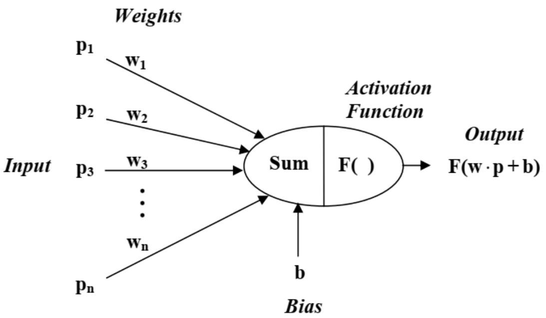

Figure 8. A fundamental neuron's structure

B. Levenberg Marquardt (LM)

The Levenberg Marquardt method (LM) is a method of optimization or non-linear solution of the least squares. This is the quickest method of retropropagation training. Levenberg [22] and later Marquardt [23] proposed a Gauss-Newton method to depreciate.

C. Network Architecture and Training

The feedforward MLP network is employed to solve the inverse kinematics of each ABB robot. The learning technique employed is based on supervised batches of target values provided to the network by Matlab/SimMechanics and predictions (outputs) are then generated by the network based on the regression value of a well-trained network a successfully trained network [24,25].

MATLAB/neural network toolbox is used for training, validation, and testing. Fig. 9 shows a block diagram for traditional ANN and its model as follows:

$$[Q_i] = \begin{bmatrix} \theta_{i1} \\ \theta_{i2} \\ \theta_{i3} \\ \theta_{i4} \\ \theta_{i5} \\ \theta_{i6} \end{bmatrix} = ANN\_Net(X, Y, Z, R_x, R_y, R_z) \tag{36}$$

With X, Y, Z, R_x, R_y, R_z are the coordinates of the manipulated object with respect to the global reference frame.

$[Q_i]$: is each cooperative robot's joint coordinate vector.

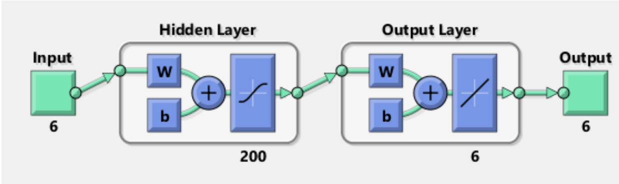

Figure 9. Neural network architecture.

Levenberg-Marquardt is therefore used its reputation for ensuring rapid convergence of the learning error [20, 23–27]. For all the 1000 samples collected, 150 are used for validation and 150 for testing the neural network. The 700 of the remaining samples are used for the actual training of the neural network.

The following Figs. 10 and 11 show how each arm's neural network's performance has changed over time:

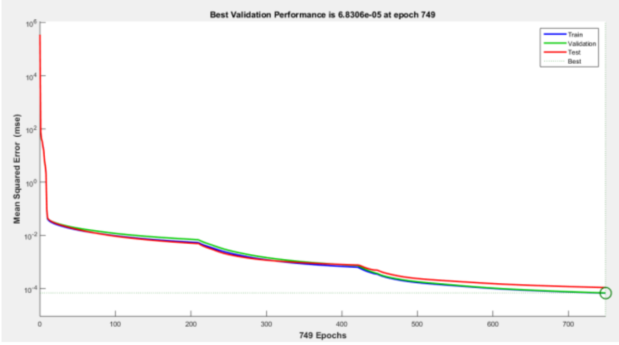

Figure 10. The performance of the neural network for robot 1.

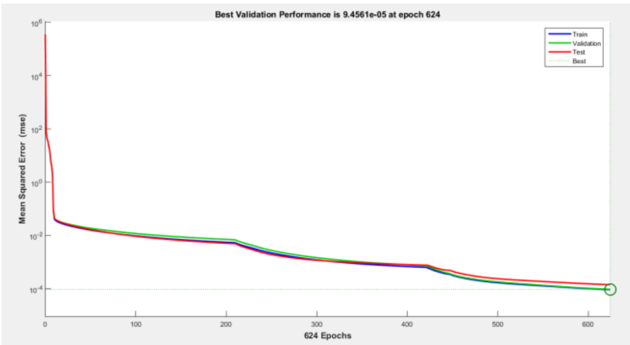

Figure 11. The performance of the neural network for robot 2.

VII. Simulation and Results

It would be wise to compute a solution for the inverse kinematics of the multi-robot system in order to control the handling error because, in general, the desired movement of the manipulated object is performed in cartesian coordinates while the movement of the two manipulator robots is controlled from the joint coordinates.

We import the two neural networks into SimMechanics for testing when they have finished their training. The SimMechanics model of our cooperative system, including the two neural networks, is shown in the schematic in Fig. 12 below.

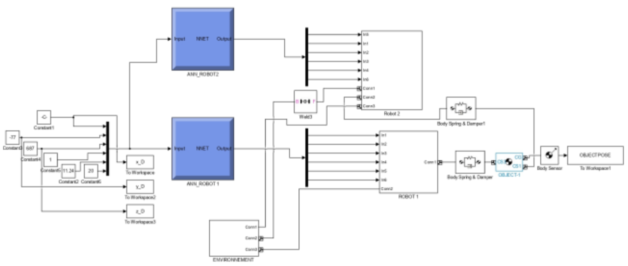

Figure 12. Internal blocks in Simulink of the cooperative system controlled by intelligent controller

A. Evaluation of the Trajectory by the Neural Network

Obviously, for this reason, a sample of the coordinates of the object manipulated in the workspace (X, Y, Z, R_X , R_Y , R_Z) is introduced in the network. This set of data will correspond to known values of the joint coordinates delivered by ANN; the same values would be compared as calculated from the previous section or given directly by Matlab/SimMechanics. Table III below shows some samples of these simulation results.

TABLE II. ERROR CALCULATION FOR PREDICTED JOINT ANGLES

Input (mm)	Robots joint angles	Real output in degrees	desired output in degree	Error %
$\begin{bmatrix} X \\ Y \\ Z \\ R_X \\ R_Y \\ R_Z \end{bmatrix} = \begin{bmatrix} 747.6 \\ -77.0 \\ 687 \\ 1 \\ 11.24 \\ 20 \end{bmatrix}$	θ_{11}	−22.77	−22.46	1.36
	θ_{12}	11.52	11.90	3.29
	θ_{13}	38.02	37.84	0.47
	θ_{14}	57.57	59.58	3.49
	θ_{15}	52,78	55.11	4.41
	θ_{16}	1	0.981	1.9
	θ_{21}	−4.30	−4.41	2.55
	θ_{22}	−17.96	−18.97	6.18
	θ_{23}	48.95	47.96	2.02
	θ_{24}	−24.65	−24.12	2.15
	θ_{25}	79.52	78.48	1.3
	θ_{26}	1	0.986	1.4
$\begin{bmatrix} X \\ Y \\ Z \\ R_X \\ R_Y \\ R_Z \end{bmatrix} = \begin{bmatrix} 936.3 \\ -189.2 \\ 685 \\ 0 \\ 7.78 \\ -6.1 \end{bmatrix}$	θ_{11}	−14.09	−14,37	1.98
	θ_{12}	27,77	27,56	0.75
	θ_{13}	17,45	17,79	1.94
	θ_{14}	12,81	12,16	5.07
	θ_{15}	38,10	40,58	6.5
	θ_{16}	1	0.98	2
	θ_{21}	42.55	41,88	1.57
	θ_{22}	−4.69	−4,98	6.18
	θ_{23}	63.94	63.73	0.32
	θ_{24}	48.52	47.30	2.51
	θ_{25}	83.53	82.01	1.81
	θ_{26}	1	0.97	3

Several readings can be taken to calculate the position/orientation errors. The average position/orientation error of the object manipulated by the cooperative system in a given position/orientation is shown in Table III below:

TABLE III. THE PROPOSED METHOD'S PERFORMANCE FOR A ROBOTIC SYSTEM.

Desired position/orientation	Real position/orientation	Parameters	Error(%)
$\begin{bmatrix} 630 \\ -150 \\ 670 \\ 20 \\ 10 \\ 15 \end{bmatrix}$	$\begin{bmatrix} 631.05 \\ -151.12 \\ 671.17 \\ 19.75 \\ 9.86 \\ 15.33 \end{bmatrix}$	X	0.16
		Y	0.75
		Z	0.17
		R_X	1.25
		R_Y	1.4
		R_Z	2.2

B. Trajectoir Prediction by ANN

After constructing each robot manipulator's ANN network, experiments are carried out, and the controlled solid is moved (without orientation; Rx=Ry=Rz=0) along a helical track within the workspace. The Simulink library creates the trajectory by monitoring the following equations:

$$O(t) = Amp \times Sin(Freq \times t + Phase) + Bias \quad (37)$$

So in order to program the helix equation, we proceed to the following parametric equation:

$$\left\{ \begin{array}{c} x(t) = 100 \times \sin(t) + 700 \\ y(t) = 100 \times \sin(t + \pi / 2) + 100 \\ z(t) = 3t + 600 \end{array} \right\} \quad (38)$$

In this simulation, the object to be manipulated must follow this parametric equation, which gives the desired helix built from the previous equations and real-designed thanks to the ANN network as shown in Fig. 13.

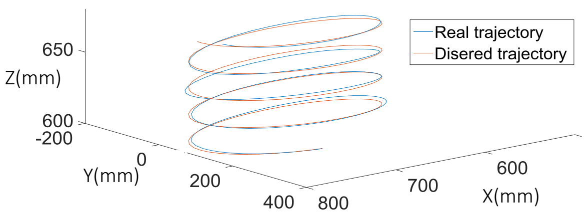

Figure 13. Trajectory configuration generated by two robots in cooperative task with ANN

The position errors along the helical path are represented in the following Fig. 14:

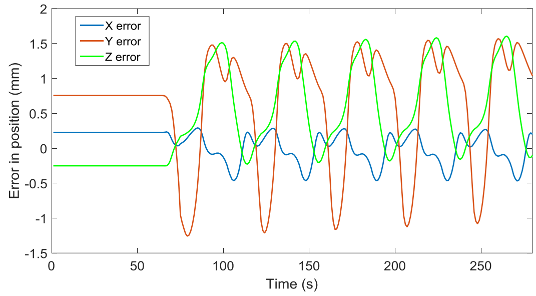

Figure 14. ANN motion error in x, y, and z direction

Another simulation in the (x, y) plane was run to see how well the two robots on the plane could be tracked.

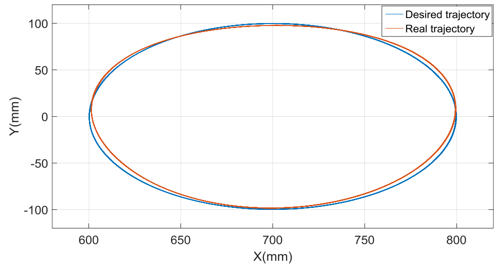

Figure 15. The comparison results of the desired and the actual trajectory of the common object using ANN

To evaluate the cooperative system with the ML control approach on a planar trajectory, we ran this simulation and obtained many images, which are shown in Fig. 16 on the following page.

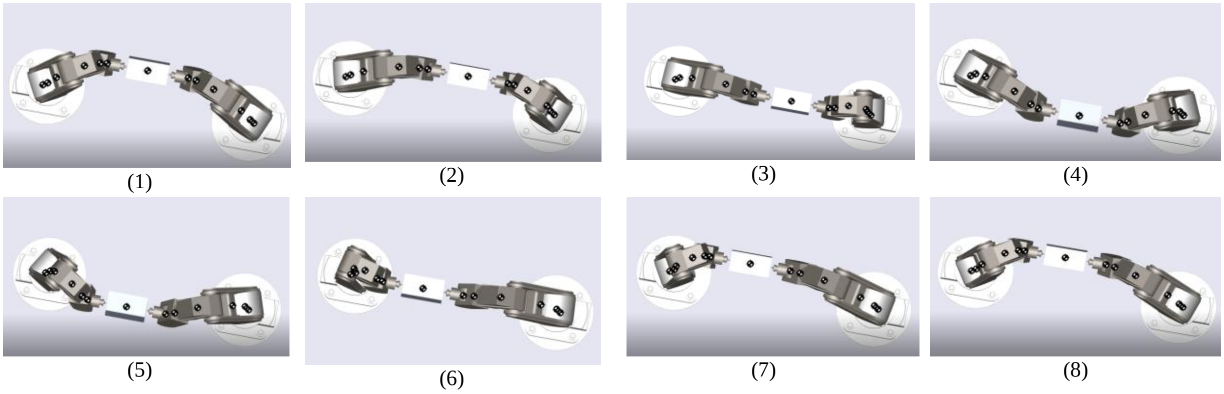

Figure 16. The simulation snapshots for the object manipulated along an elliptical trajectory

We have only tested the position of the manipulated object for the time being, but keep in mind that our ANN network can always predict the orientations of the object, as demonstrated in numerous works [28–32], as shown in Table IV below. The following two Figs. (17 and 18) show the effectiveness of our proposed model, which is concerned with tracking the object's orientation around the Y and Z axes.

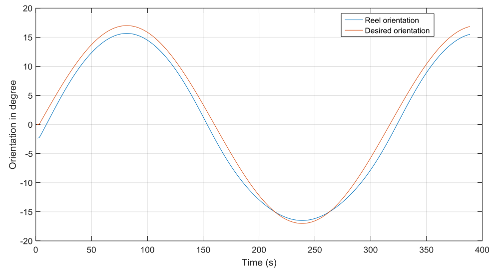

Figure 17. The comparison results of the desired and the actual orientation around the Z axis of the common object using ANN.

TABLE IV. SYSTEM PERFORMANCE COMPARISON BETWEEN THIS STUDY AND OTHERS.

Study	System	DOF	Method	MSE	Erreurs
Proposed	Two ABB IRB 120	12	ANN	$6.83e^{-5}$	X=0.16% Y=0.75% Z=0.17% R_X =1.25% R_Y =1.4% R_Z =2.2%
Almusawi et al ,2016[9]	DENSO VP6242	6	ANN	$3.3\ e^{-8}$	X=0.17% Y=0.36% Z=0.12% R_X , R_Y , R_Z are not done
Luv et all,2014 [31]	PUMA 560	6	ANN	1.217	X=6.42% Y=4.90% Z=2.92% R_X , R_Y , R_Z are not done
Hasan et all,2010[32]	FANUC M-710i robot	6	ANN	~1	X=3.34% Y=6.72% Z=0.35% R_X , R_Y , R_Z are not done

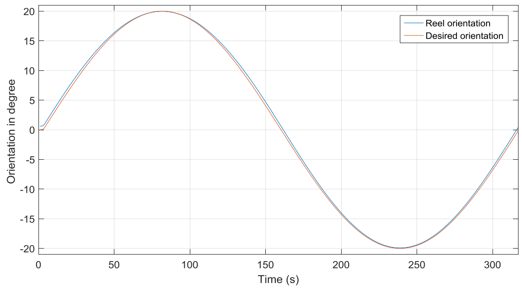

Figure 18. The comparison results of the desired and the actual orientation around the Y axis of the common object using ANN.

As shown in Fig. 13, when the two manipulators follow the object trajectory, the angles of the joints run smoothly. At the same time, the positional errors for a complex trajectory are between-2 and 2 mm, as shown in Fig. 14. Fig. 15 for an elliptical trajectory in the (x,y) plane confirms these errors.

For the capacity of the ANN control for the rotations around the z and y axes, the errors are very small, as shown in Figs. 17 and 18.

We attempt to develop a control strategy based on artificial intelligence tools, specifically the LM method, in this paper. This control will be able to cooperatively manipulate an object with two 6-DOF manipulators.

The two-armed robots' combinatorial kinematic geometric model is established, and the intelligent controller is designed and validated using SimMechanics simulation. The simulation results show that the proposed model is effective.

VIII. Conclusion

The ANN approach proposed in this paper has proven to be an effective method for 6-axis cooperative robot control, offering advantages and gains over other traditional geometric, iterative, or algebraic methods. From the coordinates of a common-object in the workspace, the ANN developed in this work can confidently predict the inverse kinematic solution of a cooperative system consisting of two 6-axis manipulator arms. With high accuracy, this prediction includes not only the position of the manipulated object but also its orientation. This method is extremely useful for cooperative robot path/trajectory planning, particularly for computer-aided cooperative robots.

Author Contributions

Abderrahim Bahani conducted the research and collected the data to write the article. El houssine Ech-Chhibat directed the research while Hassan Samri developed the computational work and finally Hicham Ait Elattar realized the CAD design and the text in English.

Conflict of Interests

The authors declare no conflict of interest.

References

[1] B. Zhou, L. Xu, Z. Meng, and X. Dai, "Kinematic cooperated welding trajectory planning for master-slave," in *Proc. the 2016 35th Control Conference*, vol. Chengdu, no. China, pp. 6369–6374, 27–29 July 2016.

[2] J. Xu, et al, "Advances in multi-robotic welding techniques: A review," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 3, March 2020.

[3] J. Xuefei Liu, et al, "Dual-arm coordinated control strategy based on modified," *Sensors*, p. 21, 2021. https://doi.org/10.3390/s21144653.

[4] U. Dey and S. K. Cheruvu, "A web-based integrated GUI for 3D modeling, kinematic study, and control of robotic manipulators," *Comput Appl Eng Educ.*, 2020, pp. 1–13. https://doi.org/10.1002/cae.22282

[5] R. Benotsmane, et al, "Simulation and trajectory optimization of collaborating robots," *Academic Journal of Manufacturing Engineering*, vol. 18, no. 4, pp. 191-197, 2020.

[6] Y. Gan, et al, "Multi-robot trajectory planning and position/force," *Applied Science*, 5 March 2019.

[7] D. Manocha et al, "Efficient inverse kinematics for general," *IEEE Transactions on Robotics and Automation*, vol. 10, pp. 648-657, Oct. 1994.

[8] B. P. Chiacchio and J. Coppolaand Curatella, "Task-oriented motion planning for multi-arm robotic systems," *Robot, Comput.nteg.manuf*, vol. 28, p. 2012, 2012.

[9] A. R. J. Almusawi et al , "A new artificial neural network approach in solving inverse," *Computational Intelligence and Neuroscience*, vol. 2016, p. 10, 2016.

[10] B. Daya, et al, "Applying neural network architecture for inverse kinematics problem in robotics," *Journal of Software Engineering and Applications*, vol. doi:10.4236/jsea.2010.33028, March 2010.

[11] Z. Han. Ang et al, "Development of an artificial intelligent approach in adapting the characteristic of polynomial trajectory planning for robot manipulator," *International Journal of Mechanical Engineering and Robotics Research*, vol. 9, no. 3, March 2020.

[12] Z. Han. Ang et al, "Proper Jacobian pseudo inverse neural network matrix method applied to robot inverse kinematics controlling," *International Journal of Mechanical Engineering and Robotics Research*, vol. 5, no. 2, April 2016.

[13] M. H. Barhaghtalab, V. Meigoli, and al, "Dynamic analysis, simulation, and control of a 6-DOF IRB-120 robot," *J. Cent. South Univ*, vol. 25, p. 2219−2244, 2018.

[14] A. L. Gustavo M. Freitas, "kinematic control of constrained robotic systems," *Revista Controle & Automaçăo*, vol. 22, pp. 559-571, 2011.

[15] J. Luh and Y. Zheng, "Constrained relations between two coordinated industrial robots for motion," *Int. J. Robot. Res.*, vol. 6, p. 60–70, 1987.

[16] Y. Zheng and J. Luh, "Control of two coordinated robots in motion," in *Proc. of the 1985 24th IEEE*, Vols. Fort Lauderdale, FL, no. USA, pp. 1761–1766, December 1985.

[17] K. Levenberg, "A method for the solution of certain non-linear problems in," *Quart. Appl. Math. 2*, pp. 164-168, 1944.

[18] D. Marquardt, "An algorithm for least-squares estimation of nonlinear," *SIAM J. Appl. Math. 11*, pp. 431-441, 1963.

[19] Al-Alaoui M.A. a,d al, "A cloning approach to classsifier training," *IEEE Transactions on Systems, Man and Cybernetics – Part A: Systems*, vol. 32, no. 6, 2002.

[20] C. J. Alba E, "Training neural networks with GA hybrid

algorithms," *Lecture Notes in Computer Science,* vol. 3102, no. Berlin / Heidelberg, pp. 852-863, 2004.

[21] B. M. H. M. Demuth H, "Neural network toolbox 6," *User's Guide*, vol. Inc, 2008.

[22] F. L., "An implementation of the levenberg-marquardt algorithm," *Eidgenossische Technische Hochschule Zurich,* 1996.

[23] M. M. Hagan MT, "Training feedforward networks with the marquardt algorithm," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, 1994.

[24] I. S. Wilamowski BM, "An algorithm for fast convergence in training neural networks," in *Proc. the International Joint*, 2001, pp. 1778-1782.

[25] A. V. Duka, "Neural network based inverse kinematics solution for trajectory," *Procedia Technology* , pp. 20–27, 2014.

[26] W. A. Wolovich and H. Elliott, "A computational technique for inverse kinematics," in *Proc. 23rd IEEE Conference on Decision and Control*, vol. 23, Dec 1984, pp. 1359-1363.

[27] C. R. Houck, J. Joines, and M. Kay, "A genetic algorithm for function optimization: A MATLAB implementation," *ACM Transactions on Mathematical Software*, 1996.

[28] C. W. Wampler, "Manipulator inverse kinematic solutions based on vector formulations and damped least-squares methods," *IEEE Trans. on Syst., Man, Cyber.*, vol. 16, pp. 93-101, 1986

[29] L. Wei, H. Wang, Y. Li, "A new solution for inverse kinematics of manipulator based on neural network," *International Conference on Machine Learning and Cybernetics*, vol. 2, no. 2-5, pp. 1201–1203, Nov. 2003.

[30] R. V. Mayorga, P. Sanongboon, "Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An artificial neural network approach," *Robotics and Autonomous Systems*, vol. 53, pp. 164-176, 2005.

[31] A. T. Hasan, N. Ismail, A. M. S. Hamouda, I. Aris, M. H. Marhaban, and H. M. A. A. Al-Assadi, "Artificial neural networkbased kinematics Jacobian solution for serial manipulator passing through singular configurations," *Advances in Engineering Software*, vol. 41, no. 2, pp. 359–367, 2010.

[32] A. Luv, A. Kush, and J. Ruth, "Use of artificial neural networks for the development of an inverse kinematic solution and visual identification of singularity zone(s)," *Procedia CIRP*, vol. 17, pp. 812–817, 2014.

Abderrahim Bahani was born in Morocco in 1989. He received the B.Sc. and M.Sc. degrees in mechanical engineering from the 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 in 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 a B.Tech. in 1989, the DESA in Physical Sciences, and a 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 a Master's degree in industrial systems engineering from the University of Evry Val d'Essone, France, in 2012. Currently, he is a lecturer of robotics 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.