Article

A Multi-Objective Modified PSO for Inverse Kinematics of a 5-DOF Robotic Arm

Nizar Rokbani [1,2], **Bilel Neji** [3,*], **Mohamed Slim** [1], **Seyedali Mirjalili** [4,5] **and Raymond Ghandour** [3]

[1] Institute of Applied Science & Technologies of Sousse, University of Sousse, Sousse 4034, Tunisia; nizar.rokbani@ieee.org (N.R.); medselim@gmail.com (M.S.)

[2] Research Group on Intelligent Machines, National Engineering School of Sfax, University of Sfax, Sfax 3029, Tunisia

[3] College of Engineering and Technology, American University of Middle East, Egaila 54200, Kuwait; raymond.ghandour@aum.edu.kw

[4] Centre for Artificial Intelligence Research and Optimisation, Torrens University, Adelaide 5000, Australia; seyedali.mirjalili@alumni.griffithuni.edu.au

[5] Yonsei Frontier Lab, Yonsei University, Seoul 03722, Korea

[*] Correspondence: bilel.neji@aum.edu.kw

Abstract: In this paper, a new modified particle swarm optimization, m-PSO, is proposed, in which the novelty consists of proposing a fitness-based particle swarm optimization algorithm, PSO, which adapts the particles' behavior rather than the PSO parameters and where particles evolve differently considering their level of optimality. A multi-objective optimization, MO, approach is then built based on m-PSO. In the proposed method, particles with fitness better than the mean local best are only updated toward the global best, while others keep moving in a classical manner. The proposed m-PSO and its multi-objective version MO-m-PSO are then employed to solve the inverse kinematics of a 5-DOF robotic arm which is 3D-printed for educational use. In the MO-m-PSO approach of inverse kinematics, the arm IK problem is formulated as a multi-objective problem searching for an appropriate solution that takes into consideration the end-effector position and orientation with a Pareto front strategy. The IK problem is addressed as the optimization of the end-effector position and orientation based on the forward kinematics model of the systems which is built using the Denavit–Hartenberg approach. Such an approach allows to avoid classical inverse kinematics solvers challenges such as singularities, which may simply harm the existence of an inverse expression. Experimental investigations included the capacity of the proposal to handle random single points in the workspace and also a circular path planning with a specific orientation. The comparative analysis showed that the mono-objective m-PSO is better than the classical PSO, the CSA, and SSA. The multi-objective variants returned accurate results, fair and better solutions compared to multi-objective variants of MO-PSO, MO-JAYA algorithm, and MO-CSA. Even if the proposed method were applied to solve the inverse kinematics of and educational robotics arms for a single point as well as for a geometric shape, it may be transposed to solve related industrial robotized arms withthe only condition of having their forward kinematics model.

Keywords: inverse kinematics; multi-objective optimization approach; PSO; robotics; robotic arm

1. Introduction

Engineering problems are, in general, multi-objective ones, where solutions are based on trade-offs of functionality, ergonomic constraints, financial limitations, and technical awareness. The decision-making process in engineering may be very challenging essentially because of the compromises which should be made between possibilities and constraints. There are two main approaches to handle such problems; the first consists of addressing them as a mono-objective problem made by merging the several controversial sub-objectives into a single one using an old mechanism called the weighted sum. The weighted sum

aggregates the different objectives of a problem into a single one [1–3]. The weighted sum consists of aggregating several mono-objective functions using a moderation coefficient, the weight, which is defined according to the relevance of an objective toward others. The main issue with such a strategy is how to manage those coefficients and how to define the needed aggregation rules. In reality, this approach includes a heuristic aspect, since in reality there are no rules fixing clearly how the coefficients, weights, are attributed. When all objectives obtain the same impact on optimality, we may think about equivalent equal weights, while this is very rare in reality [3], and arguing the choices of a weighting is far from being a simple issue.

Multi-objective optimization tends to find a trade-off of several controversial mono-objectives satisfying a global optimality for a specific need. In Pareto-based multi-objective optimization, MO, a set of mono-objective solutions are generated, then a selection strategy using the Pareto front is used to merge the objectives and to find the solution satisfying several objectives [4].

Alternatively, and using the fuzzy or the rough representations of the controversial objectives, a rough or a fuzzy solution representing a possible trade-off may be defined using rough sets and fuzzy sets, which may be also used for the assessment of global optimality based on mono-objectives results [5].

Pareto multi-objective strategies have been investigated for mechanical system design by Deb et al. since 2005, where authors applied an evolutionary multi-objective optimization to design a leg mechanism based on three objective functions [6]. In a more generic manner, a multi-objective bees algorithm (MOBA) was investigated to solve three different multi-objective engineering problems: a satellite heat pipe design problem, a space truss design, and the pressure vessel design problem. Results showed evidently that the Pareto-based approach is better than the weighted sum method, essentially for the test bed of mechanical engineering problems [7].

In industrial robotics, robotized arms are used in a large panel of applications, essentially in mass production and assembly lines, such as car assembly lines [8]. Robotics arms are also used in food manufacturing, handling operations, and smart logistics chains [9,10]. Robotics arms may be considered as a generic tool of any industrial process; at the only condition that a specific end-effector is available or specially added to the arm, such tools may be used for painting or welding operations [11]. Whatever the industrial application of a robotic arm, a path design is needed. The path design plans the robot motions needed to perform a task [12]. To perform a task, a robot arm needs to achieve a position with respect to a suitable orientation. Such a criterion is mandatory for handling processes where a specific orientation is needed for the end-effector position to be able to grasp objects. In welding processes, the welding tool should achieve a target point with a specific orientation, making the inverse kinematics problem a real multi-objective one [11]. In robotics, inverse problems may also be addressed as multi-objective; among them, inverse dynamics and inverse kinematics, IK. IK is a typical multi-objective problem at any time the focus is not limited to the end-effector position.

For inverse kinematics of robotic arms, the weighted sum is commonly used to merge position and orientation objectives of the end-effectors, and even if the proposals are presented as multi-objective, this approach is definitely closer to mono-objective rather than multi-objective. The authors in [13] proposed a knowledge-based artificial bee, k-ABC, which is an ABC with a compromise between the exploration and exploitation. The k-ABC was applied to solve the inverse kinematics of a 5-degree of freedom, 5-DOF, robotic arm, by merging the error fitness and the orientation into a single mono-objective formulation with application which minimizes the position and the orientation Euclidean distance toward a couple of target values. The proposal showed its effectiveness in drawing a circular path. In [14], a particle swarm optimization inverse kinematics solver, Ik-PSO, is investigated for the 6-DOF industrial KUKA KR 6 r900, with position and orientation objectives, and where the multi-objective strategy was based on a weighted sum of position and orientation errors. A similar approach was proposed with application to a 5-DOF

industrial robot, the Kuka Youbot manipulator [15], where authors used the inertia weight PSO with an exponentially decreasing inertia weight, cognitive, and social coefficients. A swarm approach with behavior moderation based on Gaussian-like distributions was also proposed in [16], where a beta function was introduced to distinguish two types of swarm behaviors: one for search phase and one for exploitation phase with application to inverse kinematics. The beta salp swarm algorithm, beta SSA, was proposed in [16] for general optimization and inverse kinematics. The beta SSA is an SSA where search and explore phases are managed using beta profiles.

Multi-objective approaches of articulated systems and robotic arms are not popular, while they are more effective and more suitable to handle the difference in dimensions between position and orientation. Pena et al. (2016) developed an MO strategy for inverse kinematics for a 6-DOF robotic arm based on The Fast and Elitists Non-Dominated Sorting Genetic Algorithm and the Bacterial Chemotaxis Multi-Objective Algorithm [17]. The multi-objective inverse kinematics, MO-IK, solver was used to optimize a couple of objective functions for position and orientation of the end-effector. A multi-objective PSO (MO-PSO)-based method for inverse kinematics was proposed in [18], where authors used MO-PSO to minimize the error position as well as the angular elongation between initial and final poses. An evolutionary multi-objective approach for inverse kinematics was proposed in [19] for the grasping problem of the humanoid robots; the proposal showed that such an approach can find accurate solutions in real time when compared to classical IK solvers.

In this paper, a new modified behavior PSO variant is proposed, m-PSO, which stands for a fitness-oriented PSO where the swarm particles evolve differently according to their fitness. A multi-objective optimization approach is then proposed based on m-PSO, the MO-M-PSO, which is applied to solve the inverse kinematics of a 5-DOF robotic arm, which is a 3D-printed educational robot. The arm inverse kinematics problem is formulated as a multi-objective problem assessing end-effector position and orientation as two separate objectives. A Pareto front strategy is then used to find out the best compromise between those objectives.

The remainder of this paper is organized as follows: Section 2, presents the multi-objective optimization concepts and challenges, then in Section 3, a multi-objective modified PSO, MO-m-PSO, is presented and detailed. Section 4 is dedicated to experimental investigations, which include a single-point IK solution as well as circular path planning for the 5-DOF robotic arm. The proposed method is compared to recent state-of-the-art contributions in Section 5. Finally, conclusion and perspectives are developed in Section 6.

2. Modified PSO and Multi-Objective Modified PSO

2.1. The Modified Behaviour PSO, m-PSO

The inertia weight particle swarm optimization, PSO, as in [20], is the most used PSO version. PSO is a swarm optimization technique where a swarm of (k) individuals, particles, are moving in search space looking for an optimal solution of a problem. Any particle, x_i^k, represents a possible solution of the problem. The local best, x_l^k, and global best, x_g, are evaluated using fitness function, then particle position updates are performed according to Equations (1) and (2).

$$v_{i+1}^k = w \, v_i^k + c_1 \, rn \, (x_l^k - x_i^k) + c_2 \, rn(x_g - x_i^k) \qquad (1)$$

$$x_{i+1}^k = x_i^+ + v_{i+1}^k \qquad (2)$$

In PSO, particle positions are updated toward the best global and the best local solutions with random moderated steps. The particles are evaluated iteratively according to a fitness function. The processing may be stopped iteratively or when a solution satisfies a given precision. PSO will always return a solution, the best one obtained at the end of its processing.

Several improvements have being proposed since the early PSO variant, essentially to avoid the premature convergence since PSO uses confused search/exploit phases. In [21], the authors proposed to manage the inertia weight according to population diversity. Several adaptive PSO variants were reviewed in [22,23]. Adaptive proposals included inertia weight adaptations, as well as the control of the cognitive and social factors, c_1, and c_2, to ensure convergence. The adaptive capacities of the reviewed method were limited and the majority of survived adaptive versions are as efficient as the inertia weight PSO [23].

Fitness-adaptive PSO are adaptive PSO variants where PSO parameter changes were arranged according to the fitness of the obtained solutions. In [24], the authors proposed to adapt the inertia weight according to the fitness of a given solution and the mean fitness obtained so-far; a maximum inertia weight is applied to particles for which fitness is higher than the mean obtained so far, while a decreasing fitness is used if not. In [25], the authors also proposed to adapt the inertia weight according to the fitness of the particles, while in [26], the focus was on adapting inertia weight to the swarm size in order to better control the search and to ensure the convergence. A fitness-oriented PSO was also proposed in [27], where authors defined speed and aggregation factors, both depending on the particles' fitnesses, then proposed to arrange the inertia weight as a moderated sum of speed and aggregation.

The proposed modified PSO, m-PSO, novelty consists of proposing a fitness-based PSO which adapts the particles' behavior rather than the PSO parameters. It combines a classical inertia weight PSO with social and cognitive search mechanism to a global simplified PSO such as in [28].

In m-PSO, the swarm particles' updates are operated in different manners, some particle updates are performed according to the classical inertia weight PSO, such as in Equations (1) and (2), while some others undergo a global optimality update using Equation (4), which creates a swarm made of two sub-swarms. The particles with low-quality solutions will keep exploring the search space as a classical inertia weight PSO, while a pool of qualitative particles will search to exploit the optimality around interesting solutions around the best global solution. The proposal implements an exploration/exploitation mechanism with an iterative update where the particles may be moved from a sub-group to another according to their evolution. This process is expressed by Equations (3)–(5).

For each particle, belonging to a sub-swarm depends on its fitness and on the mean of best local solutions around. The fitness switching condition, expressed in Equation (3), stands for the following: If a particle is worse than the mean of local bests, $x_{m}ean^l$, then its velocity update follows the classical inertia weight PSO as in the velocity expression of Equation (1), and if not, the velocity update is operated toward the global best only, as can be seen in Equation (2). This specific PSO processing is detailed in Algorithm 1

if $(f(x_i^k)) > (f(x_l^m ean))$

$$v_{i+1}^k = w\, v_i^k + c_1\, rn\, (x_l^k - x_i^k) + c_2\, rn(x_g - x_i^k) \tag{3}$$

else

$$v_{i+1}^k = w\, v_i^k + c_2\, rn(x_g - x_i^k) \tag{4}$$

end if

$$x_{i+1}^k = x_i^+ + v_{i+1}^k \tag{5}$$

Here, the local best stands for a cognitive best and is the best position that the particle finds over the iterative processing, while the global best, x_g, is the best particle position ever found. The stop condition may be iterative or happens when the fitness achieves a prefixed value.

Algorithm 1 The simplified pseudocode of the m-PSO

Begin

Initialize PSO

while (not stop)
for each Particle
Compute *fitness_xi*()
$xli \leftarrow best\_local\ solution\ of\ particle(i)$
$xg \leftarrow best\_swam\ solution$
$xmean \leftarrow means\{xl1, ..., xln\}$
If $(f(x_i^k)) > (f(x_l^m ean))$
$v_{i+1}^k \leftarrow w\ v_i^k + c_1\ rn\ (x_l^k - x_i^k) + c_2\ rn(x_g - x_i^k)$
else
$v_{i+1}^k \leftarrow w\ v_i^k + c_2\ rn(x_g - x_i^k)$
end if
$x_{i+1}^k \leftarrow x_i^+ + v_{i+1}^k$
end for
end while

Return (xg)

end

2.2. Multi-Objective MO-m-PSO for Inverse Kinematics

Engineering problems are, in general, subject to conflicting objectives for which a trade-off decision is needed to fulfill those controversial issues; such problems are typical multi-objective optimization ones. It is possible to formulate the several objectives into a set of mono-objective minimization problems so that a generic formulation of a multi-objective function is addressed as in Equations (6).

$$
\begin{cases}
\min(f_1(\overrightarrow{x}), f_2(\overrightarrow{x}), ..., f_k(\overrightarrow{x})) \\
\qquad\qquad subject\ to: \\
\qquad\qquad \overrightarrow{x} \in X \\
\qquad\qquad f_i(\overrightarrow{x}) \in D_i
\end{cases}
\tag{6}
$$

where (k) is the number of objective functions $(k \geqslant 2)$, $\overrightarrow{x}$ is a vector representing the problem variable, and X is a set of feasible solutions space, it may also represent the search space at large, and D_i is the definition domain of the function $fi()$.

The multiple objective functions that the system is subject to may lead to situations where several possible solutions are acceptable while no one is fully satisfying all objectives. Those possibilities represent the Pareto non-dominated solutions. A non-dominated solution is a solution around which none of the objective functions could be improved without degrading at least one of the other objective functions.

A feasible solution $\overrightarrow{x_1}$ dominates another feasible solution $\overrightarrow{x_2}$ if $\overrightarrow{x_1}$ is no worse (better or equal) than $\overrightarrow{x_2}$ in all objectives, and $\overrightarrow{x_1}$ is strictly better than $\overrightarrow{x_2}$ at least in one objective, which could be expressed mathematically as in Equation (7):

$$
\begin{cases}
\forall\ i \in \{1, 2, ..., k\},\ f_i(\overrightarrow{x_1}) \leqslant f_i(\overrightarrow{x_2}) \\
\exists\ j \in \{1, 2, ..., k\},\ f_j(\overrightarrow{x_1}) < f_j(\overrightarrow{x_2})
\end{cases}
\tag{7}
$$

A Pareto optimal solution is a solution not dominated by any other one. The outcomes of all non-dominated solutions define the Pareto front, P. Any solution belonging to the Pareto front is assumed to be an acceptable solution of the problem. However, the best one is assumed to be the closest to the utopia point, see Figure 1.

The utopia point is a virtual ideal solution, $\overrightarrow{P_u}$, corresponding to the minimum; it may be expressed as in Equation (8).

$$\overrightarrow{P_u} = \left(\min_{\overrightarrow{x} \in X}(f_1(\overrightarrow{x})), \dots, \min_{\overrightarrow{x} \in X}(f_k(\overrightarrow{x})) \right) \tag{8}$$

For a two-objective function, the Pareto front could be visualized as in Figure 1, where dominated solutions are marked in red while non-dominated solutions are in black. The best Pareto optimal solution is the solution on the Pareto front which is the closest to the utopia point, as represented in Figure 1. Note that this representation corresponds to a couple of functions both subject to minimization.

The stop condition could either be set to the maximum achieved iterations or a fixed error value.
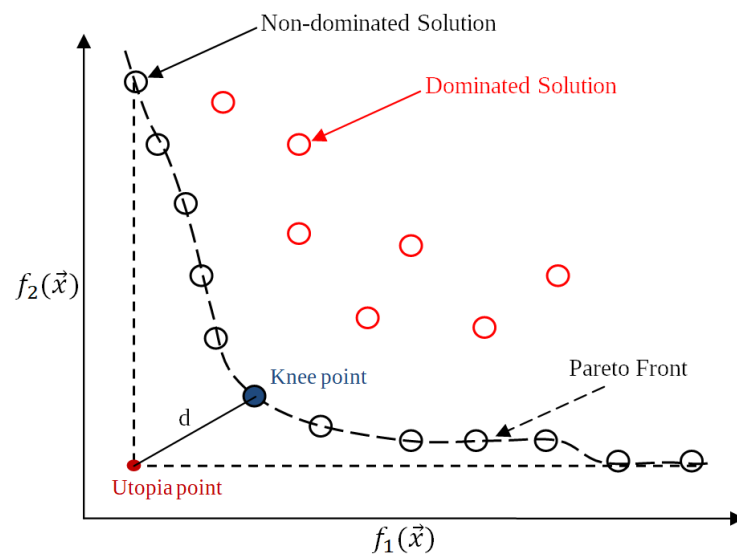

Figure 1. Pareto front of a multi-objective problem with two objective functions.

The multi-objective modified PSO, MO-m-PSO, works as a set of mono-objective m-PSO instances, where each instance is in charge of a single objective, then a Pareto-based approach is used to find a compromise solution by sorting the closest particle to the utopia point of the Pareto front, as illustrated in Figure 1.

3. Multi-Objective Inverse Kinematics Problem Statement

The inverse kinematics, IK, problem consists of computing a set of joint's motions that enable the end-effector of a robot manipulator to reach a specific target point. However, some industrial operations, such as welding, may need also a specific orientation of the end-effector besides its position. For example, the welding operation demands a functional tilt of the welding tools toward the welded surface. Such a constraint needs to be handled by a multi-objective strategy.

Classically, a problem may be solved by retrieving the inverse kinematics model by inverting the forward model, which is simple to build. However, it is not as simple as that, because the inverse model may not be existing at some specific singular points where the forward model is not invertible.

Therefore, the proposed method transforms the IK problem into an optimization consisting of minimizing the end-effector position and orientation based on the forward kinematics model of the systems, which is built using the Denavit–Hartenberg approach. Such an approach allows to avoid classical inverse kinematics solvers challenges including singularities, which may harm the existence of an inverse expression.

The proposal consists of addressing the constraints and needs with mono-objective functions, one for each, and then applying a metaheuristic solver with a Pareto strategy

to obtain the best trade-off among them. For example, when the need is about position and orientation, such as in welding or painting, two main functions are addressed:

- A minimization of the end-effector position toward a target point.
- A minimization of the end-effector orientation toward a fixed one.

3.1. The Forward Modeling of the 5-DOF-Robotic Arm

A 3D-printed robotic arm with 5-DOF excluding the gripper joint, as in Figure 2a, and the detailed kinematic representations and frames decomposition are illustrated in Figure 2b. The first three rotational joints starting from the base are actuated using an MG996R servo-motor, while the two joints of wrist roll and wrist pitch are actuated with a smaller micro-servo, the SG90. The gripper grasping is also controlled by a SG90 servo; however, it is not included in the model. The system is controlled using an Arduino board and a set of H-bridges to avoid any overload when controlling servos. Such a robotic arm has a low cost and can be used for a variety of educational applications. It allows us to understand the main issues of robotics arms' motions pacification, which are similar in industrial arms applications.

Figure 2 represents a 3D rendering of the used 5-DOF arm (Figure 2a) as well as its kinematics scheme (Figure 2b). The Denavit–Hartenberg (DH) convention is used to obtain the forward kinematics expression of the 5-DOF robot manipulator; the DH parameters of the arm are reported in Table 1 and the relative frames displacements and rotations are indicated in Figure 2b.
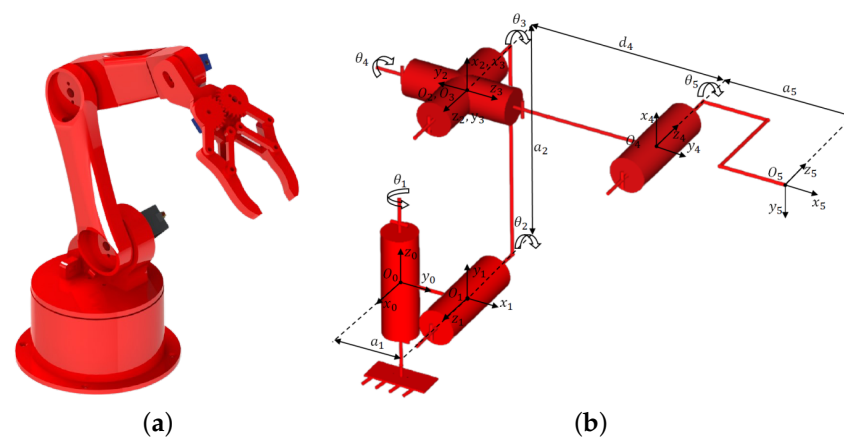

Figure 2. (**a**) A 3D rendering of the 5-DOF robot manipulator. (**b**) Kinematic and frames representations.

Table 1. Denavit–Hartenberg parameters of the 5-DOF robotic arm.

Frame (i)	a_i (cm)	α_i (°)	d_i (cm)	θ_i (°)
1	1.374	90	0	[0, 180]
2	12	0	0	[0, 170]
3	0	90	0	[0, 135]
5	0	90	11.965	[0, 180]
5	9.887	0	0	[0, 180]

3.2. Denavit–Hartenberg Forward Kinematics Model of the Robotic Arm

The Denavit–Hartenberg, DH, convention provides a systematic methodology to express the forward kinematics model of a serial robot, using a generic transformation matrix modeling the position and the orientation between two sequential frames (i) and $(i − 1)$, (T_i^{i-1}) [29,30]. Table 1 reports the DH parameters of the system, where i indicates the joint sequence, and a_i is the distance between O_{i-1} and O_i along x_i. α_i is the angle between z_{i-1} and z_i around x_i. d_i is the distance from x_{i-1} towards x_i along z_{i-1}. θ_i is the angle of joint i which is the angle between x_{i-1} and x_i around z_{i-1}.

The position and orientation of the end-effector of the 5-DOF robotic arm are expressed as a 4×4 homogeneous matrix (T_5^0), which is the result of the multiplication of the above matrices, as in Equation (9).

$$T_5^0 = T_1^0 \cdot T_2^1 \cdot T_3^2 \cdot T_4^3 \cdot T_5^4 = \begin{bmatrix} r_{11} & r_{12} & r_{13} & p_x \\ r_{21} & r_{22} & r_{23} & p_y \\ r_{31} & r_{32} & r_{33} & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{9}$$

Note that (T_i^{i-1}) stands for the homogeneous transformation matrix describing transformations of frame (i) toward frame $(i - 1)$ [30,31].

The terms $r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32}$, and r_{33} are the elements that describe the orientation of the end-effector, while (p_x, p_y, p_z) are the coordinates of the end-effector with respect to the base frame, whose final expression is in Equation (10).

$$\begin{cases} P_x = a5C4C5(C1C2C3 - C1S2S3) + a5C4C5S1 + \\ (a5S5 + d4(C1C2S3 + C1S2C3)) + (a2C1C2 + a1C1) \\ \\ P_y = a5C4C5(S1C2C3 - S1S2S3) - a5C4C5 \cdot C1 \\ \qquad\qquad + (a5S5 + d4(S1C2S3 + S1C2C3)) \\ \qquad\qquad\qquad\qquad + (a2S1C2 + a1S1) \\ \\ P_Z = a5C4C5(S2C3 + C2S3) + \\ (a5S5 + d4(S2S3 - C2C3)) + a2S2 \end{cases} \tag{10}$$

Ci stands for $Cos(\theta_i)$ and Si represesnts $Sin(\theta_i)$. The terms (ai) and (di) are the Denavit–Hartenberg, DH, parameters indicated in Table 1.

This Denavit–Hartenberg kinematic modeling approach allows to retrieve the position of the end-effector as well as the orientations. Orientations needs to be converted to Euler angles which are commonly used in orientation control for robotics. This transformation is detailed in the next paragraph.

3.3. Managing the Orientation of the End-Effector

Euler angles convention (α, β, γ) is a classical standard used in robotics, essentially to specify orientations. The Euler convention, as illustrated in Figure 3, represents orientations as follows:

- Frame 1 rotates first by an angle γ around $\overrightarrow{X_0}$, of frame 0;
- Then Frame 2 rotates of β around $\overrightarrow{Y_0}$;
- Lastly, Frame 3 rotates around $\overrightarrow{Z_0}$ by an angle α.
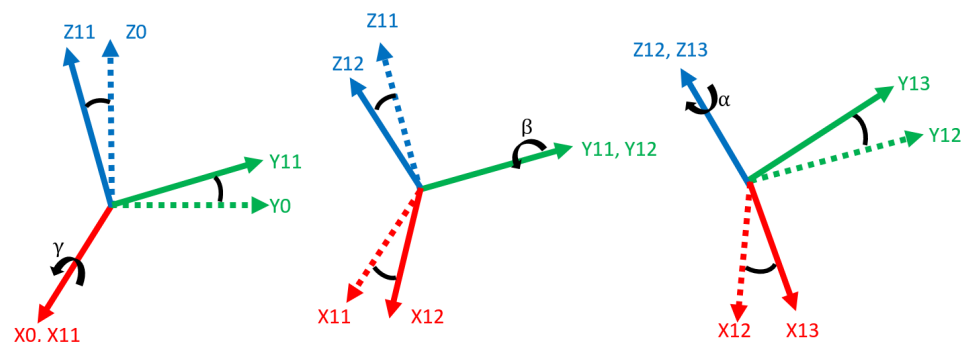

Figure 3. Euler convention rotations $R_X(\gamma)$, $R_Y(\beta)$ and $R_Z(\alpha)$.

The end-effector orientation by be represented by Euler angular convention, which allows a more generic and interpretable representation of the orientation. Euler's rotations toward the (X, Y, Z) base frame are, respectively, (γ, β, α). They may be computed on Equations (11)–(13), terms which are retrieved from the homogeneous transformation matrix detailed in Equation (9).

$$\beta = \arctan_2(-r_{13}, \sqrt{r_{11}^2 + r_{21}^2}) \tag{11}$$

$$\alpha = \arctan_2(\frac{1}{r_{21}}cos(\beta), \frac{1}{r_{11}}cos(\beta)) \tag{12}$$

$$\gamma = \arctan_2(\frac{1}{r_{32}}cos(\beta), \frac{1}{r_{33}}cos(\beta)) \tag{13}$$

where the r_{ij} terms are those appearing in the homogenous transformation matrix of the arm detailed in Equation (9).

Mono-objective optimization of the orientation aims to minimize the Euclidean distance of the generated solution toward a target orientation, as in Equation (14).

$$min f_2(Q) = \frac{|\gamma - \gamma_t| + |\alpha - \alpha_t| + |\beta - \beta_t|}{3} \tag{14}$$

3.4. Inverse Kinematics as a Multi-Objective Problem

Assuming that the joints coordinate is $Q()$, which may be rotations or translations but also angular or linear velocities; that the target position is X_t; and that the forward kinematics model of the system is in Equation (9), the inverse multi-objective inverse kinematics model considering orientation may be formulated as in Equation (15).

$$\text{Find } Q() \text{ satisfying} \begin{cases} min f_1(Q) = min \sqrt{(p_x - x_t)^2 + (p_y - y_t)^2 + (p_z - z_t)^2} \\ min f_2(Q) = min \frac{|\gamma - \gamma_t| + |\alpha - \alpha_t| + |\beta - \beta_t|}{3} \\ \text{Subject to: } Q \in Q_{DS} \end{cases} \tag{15}$$

where $f_1(Q)$ stands for the Euclidean distance of the end-effector position to the target position, which is subject to minimization. $f_2(Q)$ is the Euclidean distance of the end-effector orientation to the target orientation which is also subject to minimization. (x_t, y_t, z_t) are the coordinates of the target position X_t. $P = (p_x, p_y, p_z)$ is the end-effector position for a given joints position (Q). The target orientation $(\alpha_t, \beta_t, \gamma_t)$ represents the desired orientation of the end-effector.

The first objective is related to f_1 and aims to minimize the distance between the target position and the end-effectors' position. The second objective is related to the function f_2, and tends to satisfy a given orientation constraint of the effector. Orientation optimization attempts to minimize the mean of the total angular errors to obtain the closest orientation possible to what is desired.

The multi-objective inverse kinematics solving may be resumed within the key steps presented in Algorithm 2

Note that both mono-objectives PSO processes are set to the same number of particles, similar m-PSO parameters, and the same maximum iterations number. The stop condition could either be set to the maximum achieved iterations or a fixed error value.

Algorithm 2 Simplified pseudocode of Mo-m-PSO for inverse kinematics.

Begin

Fix target position X_t
Fix orientation target O_t
Initialize a MO-m-PSO Swarm
Q() ← random(Q1, Q2, ..., Qn)
$F_K(Q_i)$ ← (forward kinematics model)

while (not stop)
for   each   particle

Compute $(p_x,\ p_y,\ p_z) = F_K(Q_i)$ ▷ System forward kinematics
Compute $O_t\ =\ Euler(Q_i)$ ▷ Euler angles transformation

Execute $m - PSO()$ / subject to
min $f_1()$ as a mono-objective
Execute $m - PSO()$ / subject to
min $f_2()$ as a mono-objective

find Pareto Utopia point of $(f_1(), f_2())$
Select $Best_q$ / satisfying $[\min(f_1(), f_2())]$

end for
end while

Return ($Best_q()$)

end

---

4. Simulation and Results

The modified PSO, m-PSO, and its multi-objective version MO-m-PSO for inverse kinematics were applied to solve the inverse kinematics of an educational robotic arm using four scenarios:

The first test stands for the mono-objective version evaluations; it includes the single target point performances analysis and comparison with related methods.

The second scenario is the classical one-target test, where one target and one orientation is given and MO-m-PSO aims to determine the appropriate angles that assure achieving that orientation and position with as minor an error as possible.

The third scenario consists of generating 100 different random points covering the majority of the workspace space of the robot, then for each point an inverse kinematics solution is returned by MO-m-PSO. To assess the solver quality, the mean and standard deviations of the results are analyzed.

The path tracking test is where a circular trajectory is generated in the workspace of the 5-DOF robot. The circular path is made of 50 points. The role of the inverse kinematics solvers based on MO-m-PSO is to find the best possible solution for each point taking into consideration the desired orientation. Such a scenario may be used in soldering or in the industrial robotized components placement process.

4.1. Mono-Objective, m-PSO, Single Target Point Evaluation

A feasible target point is fixed at the location ($x = 25.144$ cm, $y = 9.573$ cm, $z = 18.092$ cm), which corresponds to an random joints position $q = [30°, 60°, 45°, 60°, 60°]$. Then, the m-PSO mono-objective solver is asked to achieve this position. The test is operated with a swarm of 50 particles, $w = 0.8$ and $c1 = c2 = 1.493$ for the search behavior of the m-PSO, and $c2 = 1.493$ and $w = 0.8$ for the exploitation behavior.

This test was repeated 30 times, and for all tests, the proposal achieved a feasible solution of the problem with position errors ranging from 3.01×10^{-10} to 1.08×10^{-5}. The average error is about 2.01×10^{-6}. A 3D representation of a solution is shown in Figure 4a; its corresponding fitness of up to 500 iterations appears in Figure 4b.

(a) (b)
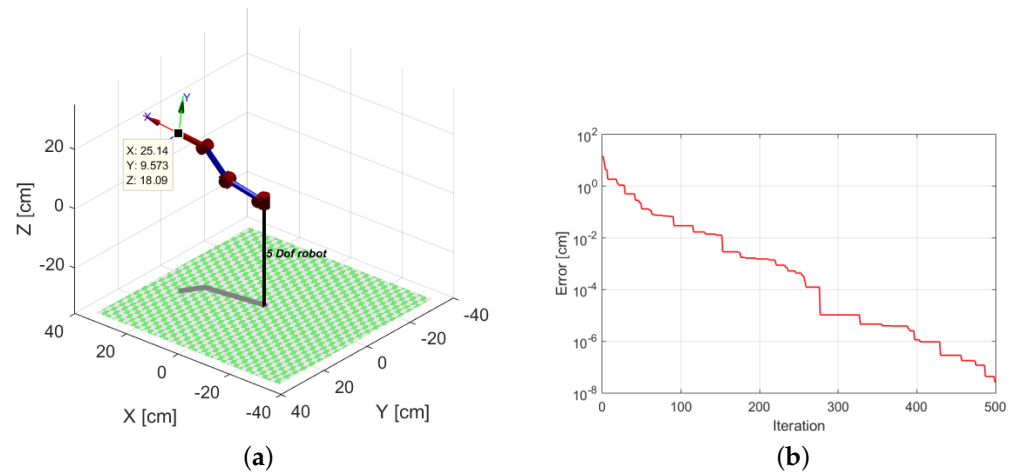
Figure 4. Mono-objective m-PSO for inverse kinematics: (**a**) 3D representation of the best solution using m-PSO, (**b**) m-PSO fitness evolution up to 500 iterations.

It is important to note here that the obtained solutions through the computation, m-PSO, by about iteration 300, are precise enough for the real robotic arm used for a real practice test, since the printed robot resolution has about a 0.5 cm range, which means a quadratic position error of about 1.0×10^{-4}. The 3D representation of the robotic arm is performed using the Corke robotics toolbox [31].

4.2. MO-m-PSO with Position and Orientation Objectives

This test is equivalent to the first one but with an additional orientation constrain. The test concerns the multi-objective variant of m-PSO, MO-m-PSO. The target point ($x = 25.144$ cm, $y = 9.573$ cm, $z = 18.092$ cm) was chosen arbitrary by applying the forward kinematics of a particular configuration $[30°, 60°, 45°, 60°, 60°]$. The desired orientation used in this test was formulated by $[\alpha = 0°, \beta = -30°, \gamma = -23.8°]$. Its forward kinematics homogeneous transformation matrix is as in Equation (16):

$$T = \begin{bmatrix} 0.8660 & 0.2021 & -0.4574 & 25.1441 \\ 0.0000 & 0.9147 & 0.4042 & 9.5735 \\ 0.5000 & -0.3500 & 0.7921 & 18.0927 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (16)$$

According to Equation (16), the resulted orientation is defined by ($\alpha = 0°$, $\beta = -30°$, $\gamma = -23.84°$).

Figure 5a shows the Pareto font for position/orientation objectives which allows to select the optimal solution corresponding to $Q = [30.00°, 59.16°, 44.59°, 54.63°, 57.92°]$, for which the corresponding 3D representation appears in Figure 5b.

The test was repeated over 100 epochs, which allowed us to compute the mean position and orientation errors as well as the standard deviations. Results are presented in Table 2, which shows that the best position error is about 3.81×10^{-10}; this is not technically achievable with the current 5-DOF robotic arm. The best orientation error was about 2.34×10^{-9} which is extremely precise for an educational robotic arm. The means and standard deviations of these errors are also presented in Table 2. The mean of position error is about 0.015, while the mean of orientation error is about 0.098.
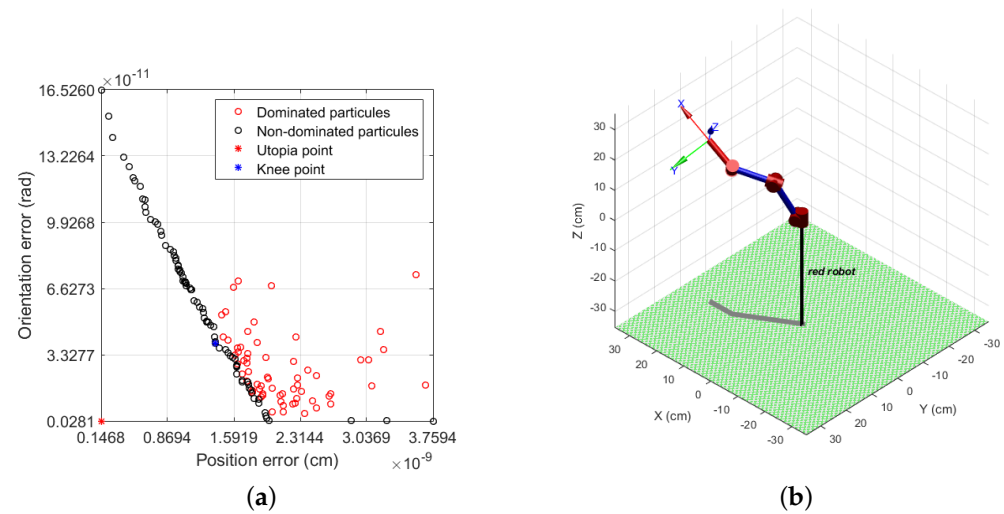
Figure 5. Multi-objective m-PSO for inverse kinematics with position and orientation objectives: (**a**) Pareto front for position error/orientation objectives, (**b**) visualization of robot arm.

Table 2. Statistical results of MO-M-PSO target point with end-effector orientation.

	Best	Worst	Mean	Std
Position error	3.81×10^{-10}	0.0492	0.0150	0.1973
Orientation error (rad)	2.34×10^{-9}	0.6439	0.0918	0.2268

4.3. Multiple Target Points for Statistical Analyses

The test consists of evaluating the capacity of MO-m-PSO to correctly solve a set of 100 random points generated in the workspace of the 5-DOF robotic arm; each point is associated with a random orientation. This test serves for a statistical analysis of the MO-m-PSO in solving the IK problem. The test conditions are the following: 100 search agents and 500 iterations for each target point. Table 3 shows the best, worst, mean, and standard deviation of the obtained solutions.

Table 3. Statistical results of MO-M-PSO for IK problem.

Error	Best	Worst	Mean	Std
Position (cm)	1.88×10^{-9}	2.925	1.450	1.897
Orientation (rad)	1.65×10^{-10}	0.643	0.091	0.126

It is clear that when an orientation constraint is added to the inverse kinematics, the position errors obtained are more important than when the system operates without the orientation constrain, while in real industrial applications, the orientation respect may make a great difference between acceptable and non-acceptable solutions.

Figure 6 shows both the position errors and the orientation errors for the 100 random test points, taking into consideration both the position and the orientation errors. The plots confirmed the results and showed that the orientation objective was achieved with a relative degradation of the position error, as in mono-objective m-PSO.
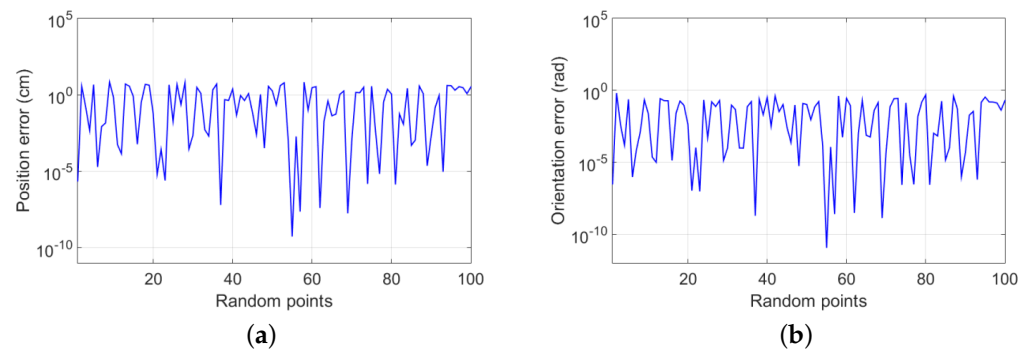
Figure 6. (**a**) Position error of each randomly generated point, (**b**) orientation error of each randomly generated point.

4.4. Multi-Objective m-PSO for Path Tracking with Fixed Orientation Simulation

The path tracking test consists of assessing the robot to track a circular path in a 3D workspace. The path is composed of a set of target points $X_t = (x_t, y_t, z_t)$ belonging to a planar circular path (see Figure 7c) and satisfying Equation (17):

$$\begin{cases} x_t(p) = x_0 \\ y_t(p) = y_0 + r\,cos(\theta_s\,p) \\ z_t(p) = z_0 + r\,sin(\theta_s\,p) \end{cases} \tag{17}$$

where θ_s stands for the angular step, which depends on the number of points (steps) chosen in the trajectory, and the center of the circle position is at $X_o = (x_0, y_0, z_0)$. If the circle is sampled into np points, then the angular step θ_s can be computed using Equation (18).

$$\theta_s = \frac{2\pi}{np} \; [\text{rad}] \tag{18}$$

For this test, the circular path is designed as follows:

The workspace distances are specified in centimeter, cm.
The circle center position is $X_o = (x_0 = 15, y_0 = 0, z_0 = 12)$.
The circle radius is $r = 8$ cm.
The end-effector orientation is fixed so that it is maintained perpendicular to the plan of the circular trajectory, which leads to the following orientation constraints: $(\alpha = 0°, \beta = 0°)$, γ is free.

Simulations showed that the circular path tracking was conducted in a very efficient manner. The mean error obtained over the 100 points was about 0.0325 cm, where the target path is in red while the obtained path is in black. The orientations were respected with high precision since the mean error is about 5.732×10^{-5}. The mean and standard deviation of position error and orientation are reported in Table 4, while Figure 7 shows the relative error obtained at each point.

Desired and obtained path are visible in Figure 7a, while Figure 7b shows the joints' angular positions returned by MO-m-PSO for a circular path made of 50 points. A 3D representation of the arm at point 25 is presented in Figure 7c.

Table 4. Statistical errors analysis of MO-M-PSO for circular path planning.

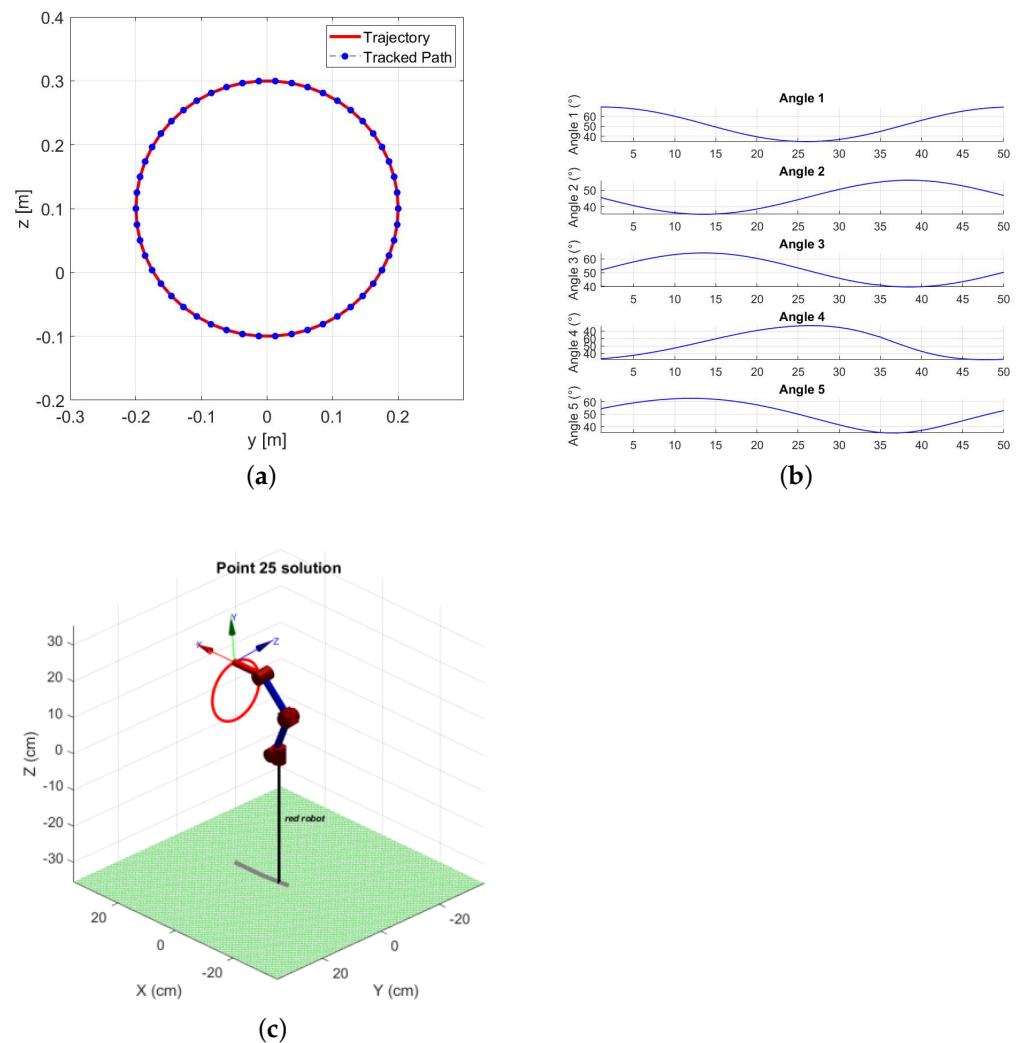|	**Best**	**Worst**	**Mean**	**Std**
Position (cm)	2.31×10^{-6}	0.268	0.032	0.191
Orientation (rad)	1.85×10^{-8}	4.58×10^{-4}	5.73×10^{-5}	0.002

Figure 7. Circular path tracking of 50 points: (**a**) The input trajectory and the tracked trajectory by robotic arm, (**b**) Angle values resulted by MO-m-PSO for each point of the trajectory, (**c**) 3D representation of the robotic arm and the circular path.

5. Nonparametric Statistical Comparatives

To compare the relative performances of the proposed m-PSO to classical PSO and also to related and close techniques, the nonparametric Wilcoxon statistical analysis was used based on the single target point, and using the same configuration for all methods in terms of number of search agents, which were 50, and in terms of maximum iteration, which was fixed at 500. Tests were based on Matlab (2019) simulations, with an AMD A6 processor and 4 GB of RAM.

Wilcoxon nonparametric analysis was conducted based on 30 independent executions of the compared metaheuristics.

5.1. Mono-Objective m-PSO Comparative Analysis

The mean fitness evolutions on the compared mono-objective techniques for 500 iterations are visible in Figure 8a. Results evidently show a different evolution dynamic of the m-PSO metaheuristic, essentially, when compared to classical PSO, SSA, and CSA. The obtained position errors for the 30 tests appear in Figure 8b, and show clearly that the m-PSO is returning a more precise inverse kinematics solution than its concurrent.
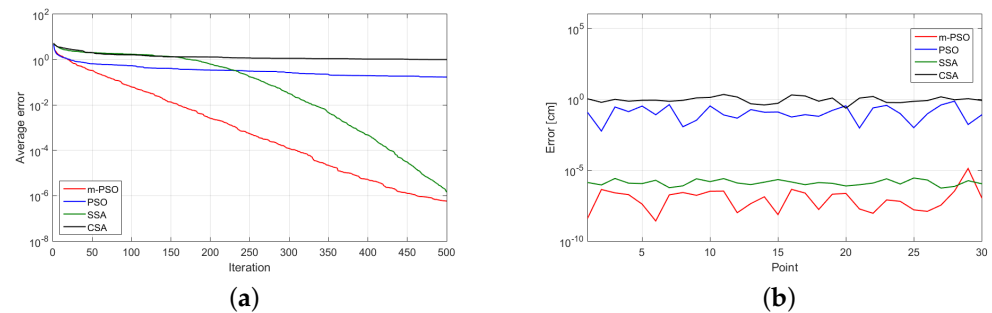
Figure 8. (**a**) Comparative mean fitness evolutions of m-PSO, PSO, CSA, and SSA. (**b**) Comparative position errors over 30 tests of m-PSO, PSO, CSA, and SSA.

These results were confirmed by the Wilcoxon signed-rank test reported in Table 5, where we can see that, globally, the position error obtained based on the m-PSO is better than the classical PSO such as in [32,33], salp swarm algorithm, SSA [34], and crow search algorithm [35].

Table 5. Nonparametric comparatives with PSO, SSA, and CSA.

Data 1	Data 2	R^+	R^-	ρ	Best
m-PSO	PSO	0	465	3.01×10^{-11}	m-PSO
m-PSO	SSA	30	435	5.57×10^{-10}	m-PSO
m-PSO	CSA	0	465	3.01×10^{-11}	m-PSO

5.2. Multi-Objective m-PSO, MO-m-PSO, Comparative Analysis

In order to benchmark the performance of the proposed method compared to existing multi-objective approaches used for inverse kinematics, MO-m-PSO was compared to multi-objective PSO for inverse kinematics, MO-PSO, and in [36], multi-objective CSA [37] and multi-objective JAYA algorithm (MO-JAYA) [38]. Comparatives are based on 30 random points RUNS and using the Wilcoxon nonparametric method. That target positions were set randomly, while the orientation was fixed to 90° for the end-effector. Comparisons showed that the proposal ranking is better than the MO-PSO and the MO-JAYA, while very close to MO-CSA, the worst. Detailed results appear in Table 6, and all tests satisfied the null hypothesis condition with $\rho < 0.05$.

Table 6. Statistical results of MO-M-PSO for IK problem.

Data	R^+	R^-	ρ	Best
MO-m-PSO/MO-PSO	486	778	2.31×10^{-3}	MO-m-PSO
MO-m-PSO/MO-JAYA	195	1069	6.87×10^{-4}	MO-m-PSO
MO-m-PSO/MO-CSA	663	661	1.02×10^{-4}	MO-m-PSO

6. Experimental Validation on a 3D-Printed ARM

The experimental investigation consists of printing, assembling, and controlling a 3D-printed arm using inverse solutions generated by the m-PSO and M0-m-PSO. The 3D-printed arm is for educational use only and serves as an integration prototype. The experimental setup is shown in Figure 9. It includes the following:

- Matlab software;
- An Arduino Uno board;
- The 3D-printed arm.

The inverse kinematics solver runs on Matlab software with a PC powered by i7 Intel processor and a RAM of 12 GB.

This test is conducted using the following steps:

- A target point is fixed by the user, it mimics the simulated random point test, then the proposed metaheuristic solver generates an online inverse solution and generates the joints control using the Arduino pulse wave modulation outputs, PWM.
- The test is video reported, and the Kinovea video analysis software is used to gather the practical joints motions as well as the real target point, as in Figure 9.
- This test is repeated 20 times and the position errors are reported to evaluate the repeatability precision.

Figure 9 shows the target point, which is $X_t = [18, 10, 25]$, in centimeters, and the obtained solution with Kinovea annotations for analysis.
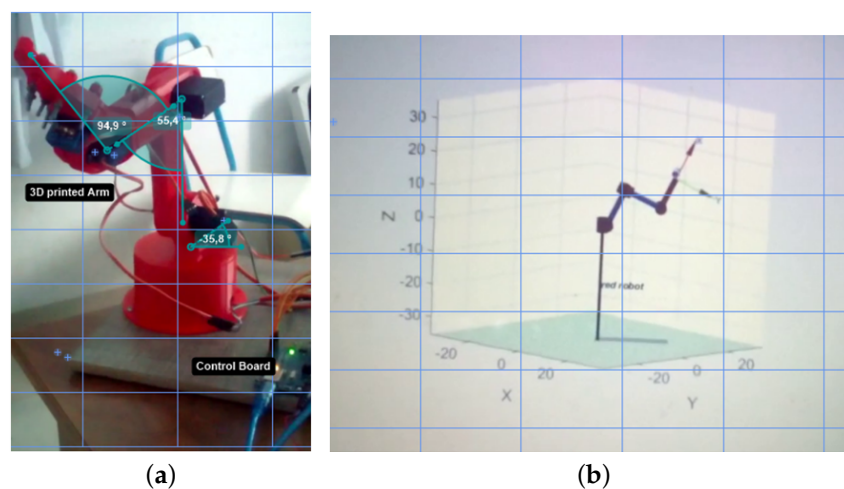

(**a**)   (**b**)

Figure 9. (**a**) 3D arm setup (**b**) Solution simulated in the control interface.

Table 7 presents the mean, minimum, maximum, and standard deviation errors towards the base frame $[X_0, Y_0, Z_0]$.

Table 7. Experimental repeatability test for the target point [18, 10, 25] statistical analysis.

Target Point	Mean	Standard Deviation	Minimum	Maximum
$X = 18$	17.98	0.37360266	17.3	18.5
$Y = 10$	9.885	0.32650461	9.4	10.5
$Z = 25$	24.85	0.29824045	24.3	25.4

Results showed that the computational results of the proposed method are very precise in regards to the experimental results. The limitations of the 3D-printed robotic arm are clear, essentially when based on maximum and minimum real positions. The real target points position errors were of $[-0.7, +0.5]$ toward X, it ranges among $[-0.6, +0.5]$ toward Y, and among $[-0.7, +0.4]$ toward Z. Standard deviations were around 0.3. The precision limitations are essentially due to the limited quality of the 3D-printing mechanical frame, making it very limited for industrial use while being acceptable for educational use, and may help in explaining the gaps that may occur between the computational models and the mechanical system limitations.

7. Conclusions and Future Works

This paper addresses the inverse kinematics problems of a 3D-printed 5-DOF robotics arm for educational use. The paper proposed a behavior-modified PSO, m-PSO, and its multi-objective version (MO-m-PSO) to solve the inverse kinematics problem of a 3D-printed 5-DOF robotic arm, with a focus on the multi-objective approach. The proposed approach adapts the particles' behavior rather than the PSO parameters, and particles evolve differently considering their level of optimality. Mo-m-PSO for inverse kinematics was tested using test scenarios which mimic the industrial needs of robotics arms:

- The simple target point test, which assesses the precision capacities of the inverse kinematics solver as well as its convergence.
- The random target point test, which allows to prove the robustness of the proposal and its capacity to correctly address any inverse solution within its work space.
- The path planning task, which allows to confirm that the proposed method can be used to plan specific inverse solutions belonging to a specific geometric shape; here, a circle.

Results demonstrated that the mono-objective m-PSO as well as the MO-m-PSO are good potential solvers for inverse kinematics; however, the multi-objective Pareto strategy reduced the position precision while satisfying the needed orientation.

Acceptable trade-offs were correctly returned with solutions which are mechanically up to the precision and resolution of the system. Errors were on the order of millimeters for a system acting within the centimeter range.

Nonparametric statistical comparatives demonstrate that the proposed methods are very competitive toward existing state-of-the-art methods.

Future investigations may concern the impact of the heuristic parameters on the level of performances such as in [39]. A focus on multi-objective optimization to cope with dynamical aspects of the robotic arm modeling, which may include torque and global energy reductions of the systems while executing a path, is also to be conducted.

Author Contributions: Formal analysis, N.R., B.N., R.G. and M.S.; Funding acquisition, B.N. and R.G.; Investigation, N.R.; Methodology, N.R., B.N. and M.S.; Project administration, N.R. and B.N.; Resources, S.M.; Validation, N.R., B.N. and R.G.; Writing—original draft, N.R., B.N. and M.S.; Writing—review & editing, B.N., S.M. and R.G. All authors have read and agreed to the published version of the manuscript.

Data Availability Statement: Not applicable.

References

1. Abraham, A.; Jain, L. Evolutionary multiobjective optimization. In *Evolutionary Multiobjective Optimization*; Springer: London, UK, 2005; pp. 1–6.
2. Marler, R.T.; Arora, J.S. The weighted sum method for multi-objective optimization: New insights. *Struct. Multidiscip. Optim.* **2010**, *41*, 853–862. [CrossRef]
3. Yang, Q.; Wang, Z.; Luo, J.; He, Q. Balancing performance between the decision space and the objective space in multimodal multiobjective optimization. *Memetic Comput.* **2021**, *13*, 31–47. [CrossRef]
4. Hou, L.; Zhang, H.; Peng, Y.; Wang, S.; Yao, S.; Li, Z.; Deng, G. An integrated multi-objective optimization method with application to train crashworthiness design. *Struct. Multidiscip. Optim.* **2021**, *63*, 1513–1532. [CrossRef]
5. Karimi, H.; Bahmani, R.; Jadid, S. Stochastic multi-objective optimization to design optimal transactive pricing for dynamic demand response programs: A bi-level fuzzy approach. *Int. J. Electr. Power Energy Syst.* **2021**, *125*, 106487. [CrossRef]
6. Deb, K.; Tiwari, S. Multi-objective optimization of a leg mechanism using genetic algorithms. Engineering Optimization. *Eng. Optim.* **2005**, *37*, 325–350. [CrossRef]
7. Moradi, A.; Mirzakhani Nafchi, A.; Ghanbarzadeh, A. Multi-Objective Optimization of Truss Structures Using Bees Algorithm. *Sci. Iran.* **2015**, *22*, 1789–1800.
8. Benotsmane, R.; Dudás, L.; Kovács, G. Survey on new trends of robotic tools in the automotive industry. In *Vehicle and Automotive Engineering*; Springer: Singapore, 2020; pp. 443–457.
9. Erzincanli, F.; Sharp, J.M. A classification system for robotic food handling. *Food Control.* **1997**, *8*, 191–197. [CrossRef]

10. Dekhne, A.; Hastings, G.; Murnane, J.; Neuhaus, F. Automation in logistics: Big opportunity, bigger uncertainty. *McKinsey Q*, 24 April 2019.

11. Hong, T.S; Ghobakhloo, M.; Khaksar, W. Robotic welding technology. *Compr. Mater. Process.* **2014**, *6*, 77–99.

12. Babu Loganathan, G. Implementing industrial robotics arms for material holding process in industries. *Harbin Gongye Daxue Xuebao J. Harbin Inst. Technol.* **2021**, *9*, 53.

13. El-Sherbiny, A.; Elhosseini, M.A.; Haikal, A.Y. A new ABC variant for solving inverse kinematics problem in 5 DOF robot arm. *Appl. Soft Comput.* **2018**, *73*, 24–38. [CrossRef]

14. Alkayyali, M.; Tutunji, T.A. PSO-based algorithm for inverse kinematics solution of robotic arm manipulators. In Proceedings of the 2019 20th IEEE International Conference on Research and Education in Mechatronics (REM), Wels, Austria, 23–24 May 2019; pp. 1–6.

15. Reyes, S.V.; Gardini, S.P. Inverse kinematics of manipulator robot using a PSO metaheuristic with adaptively exploration. In Proceedings of the 2019 IEEE XXVI International Conference on Electronics, Electrical Engineering and Computing (INTER-CON), Lima, Peru, 12–14 August 2019; pp. 1–4.

16. Rokbani, N.; Mirjalili, S.; Slim, M.; Alimi, A.M. A beta salp swarm algorithm meta-heuristic for inverse kinematics and optimization. *Appl. Intell.* **2022**, *52*, 10493–10518. [CrossRef]

17. Pena, C.A.; Guzmán, M.A.; Cárdenas, P.F. Inverse kinematics of a 6 dof industrial robot manipulator based on bio-inspired multi-objective optimization techniques. In Proceedings of the 2016 IEEE Colombian Conference on Robotics and Automation (CCRA), Bogota, Colombia, 29–30 September 2016; pp. 1–6.

18. Adly, M.A.; Abd-El-Hafiz, S.K. Inverse kinematics using single-and multi-objective particle swarm optimization. In Proceedings of the 2016 28th International Conference on Microelectronics (ICM), Giza, Egypt, 17–20 December 2016; pp. 269–272.

19. Starke, S.; Hendrich, N.; Krupke, D.; Zhang, J. Evolutionary multi-objective inverse kinematics on highly articulated and humanoid robots. In Proceedings of the 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, Canada, 24–28 September 2017; pp. 6959–6966.

20. Shi, Y.; Eberhart, R.C. Empirical study of particle swarm optimization. In Proceedings of the 1999 Congress on Evolutionary Computation-CEC99 (Cat. No. 99TH8406), Washington, DC, USA, 6–9 July 1999; Volume 3, pp. 1945–1950.

21. Zhang, D.X.; Guan, Z.H.; Liu, X.Z. Adaptive particle swarm optimization algorithm with dynamically changing inertia weight. *Control. Decis.* **2008**, *11*, 1253–1257.

22. Han, J.H.; Li, Z.R.; Wei, Z.C. Adaptive particle swarm optimization algorithm and simulation. *J. Syst. Simul.* **2006**, *18*, 2969–2971.

23. Harrison, K.R.; Engelbrecht, A.P.; Ombuki-Berman, B.M. Self-adaptive particle swarm optimization: A review and analysis of convergence. *Swarm Intell.* **2018**, *12*, 187–226. [CrossRef]

24. Liu, B.; Wang, L.; Jin, Y.H.; Tang, F.; Huang, D.X. Improved particle swarm optimization combined with chaos. *Solitons Fractals* **2005**, *25*, 1261–1271. [CrossRef]

25. Nickabadi, A.; Ebadzadeh, M.M.; Safabakhsh, R. A novel particle swarm optimization algorithm with adaptive inertia weight. *Appl. Soft Comput.* **2011**, *11*, 3658–3670. [CrossRef]

26. Dong, C.; Wang, G.; Chen, Z. The inertia weight self-adapting in PSO. In Proceedings of the 2008 IEEE 7th World Congress on Intelligent Control and Automation, Chongqing, China, 25–27 June 2008; pp. 5313–5316.

27. Yang, X.; Yuan, J.; Yuan, J.; Mao, H. A modified particle swarm optimizer with dynamic adaptation. *Appl. Math. Comput.* **2007**, *189*, 1205–1213. [CrossRef]

28. Pedersen, M.E.H.; Chipperfield, A.J. Simplifying particle swarm optimization. *Appl. Soft Comput.* **2010**, *10*, 618–628. [CrossRef]

29. Hayat, A.A.; Chittawadigi, R.G.; Udai, A.D.; Saha, S.K. Identification of Denavit-Hartenberg parameters of an industrial robot. In Proceedings of the Conference on Advances in Robotics, Pune, India, 4–6 July 2013; pp. 1–6.

30. Rocha, C.R.; Tonetto, C.P.; Dias, A. A comparison between the Denavit-Hartenberg and the screw-based methods used in kinematic modeling of robot manipulators. *Robot. Comput. Integr. Manuf.* **2011**, *27*, 723–728. [CrossRef]

31. Corke, P.I. A robotics toolbox for MATLAB. *IEEE Robot. Autom. Mag.* **1996**, *3*, 24–32. [CrossRef]

32. Rokbani, N.; Alimi, A.M. Inverse kinematics using particle swarm optimization, a statistical analysis. *Procedia Eng.* **2013**, *64*, 1602–1611. [CrossRef]

33. Dereli, S.; Köker, R. A meta-heuristic proposal for inverse kinematics solution of 7-DOF serial robotic manipulator: Quantum behaved particle swarm algorithm. *Artif. Intell. Rev.* **2020**, *53*, 949–964. [CrossRef]

34. Mirjalili, S.; Gandomi, A.H.; Mirjalili, S.Z.; Saremi, S.; Faris, H.; Mirjalili, S.M. Salp Swarm Algorithm: A bio-inspired optimizer for engineering design problems. *Adv. Eng. Softw.* **2017**, *114*, 163–191. [CrossRef]

35. Zolghadr-Asli, B.; Bozorg-Haddad, O.; Chu, X. Crow search algorithm (CSA). In *Advanced Optimization by Nature-Inspired Algorithms*; Springer: Singapore, 2018; pp. 143–149.

36. Walha, C.; Bezine, H.; Alimi, A.M. A multi-objective particle swarm optimization approach to Robotic Grasping. In Proceedings of the 2013 IEEE International Conference on Individual and collective behaviors in robotics (ICBR), Sousse, Tunisia, 13–15 December 2017; pp. 120–125.

37. Rizk-Allah, R.M.; Hassanien, A.E.; Slowik, A. Multi-objective orthogonal opposition-based crow search algorithm for large-scale multi-objective optimization. *Neural Comput. Appl.* **2020**, *32*, 13715–137461. [CrossRef]

38. Tran Thien, H.; Van Kien, C.; Anh, H.P.H. Optimized stable gait planning of biped robot using multi-objective evolutionary JAYA algorithm. *Int. J. Adv. Robot. Syst.* **2020**, *17*, 1–13. [CrossRef]
39. Rokbani, N.; Kumar, R.; Alimi, A.M.; Thong, P.H.; Priyadarshini, I.; Nhu, V.H.; Ngo, P.T.T. Impacts of heuristic parameters in PSO inverse kinematics solvers. *Int. J. Nonlinear Sci. Numer. Simul.* **2022**. [CrossRef]