

Article

Precision Denavit–Hartenberg Parameter Calibration for Industrial Robots Using a Laser Tracker System and Intelligent Optimization Approaches

Mojtaba A. Khanesar , Minrui Yan, Mohammed Isa , Samanta Piano and David T. Branson

Faculty of Engineering, University of Nottingham, Nottingham NG7 2RD, UK

* Correspondence: mojtaba.ahmadihkhanesar@nottingham.ac.uk; Tel.: +44-(0)-791-8778402

Abstract: Precision object handling and manipulation require the accurate positioning of industrial robots. A common practice for performing end effector positioning is to read joint angles and use industrial robot forward kinematics (FKs). However, industrial robot FKs rely on the robot Denavit–Hartenberg (DH) parameter values, which include uncertainties. Sources of uncertainty associated with industrial robot FKs include mechanical wear, manufacturing and assembly tolerances, and robot calibration errors. It is therefore necessary to increase the accuracy of DH parameter values to reduce the impact of uncertainties on industrial robot FKs. In this paper, we use differential evolution, particle swarm optimization, an artificial bee colony, and a gravitational search algorithm to calibrate industrial robot DH parameters. A laser tracker system, Leica AT960-MR, is utilized to register accurate positional measurements. The nominal accuracy of this non-contact metrology equipment is less than 3 $\mu\text{m}/\text{m}$. Metaheuristic optimization approaches such as differential evolution, particle swarm optimization, an artificial bee colony and a gravitational search algorithm are used as optimization methods to perform the calibration using laser tracker position data. It is observed that, using the proposed approach with an artificial bee colony optimization algorithm, the accuracy of industrial robot FKs in terms of mean absolute errors of static and near-static motion over all three dimensions for the test data decreases from its measured value of 75.4 μm to 60.1 μm (a 20.3% improvement).

Keywords: positional accuracy; industrial robots; collaborative robots; Denavit–Hartenberg parameter calibration; forward kinematic calibration; laser tracker system; artificial intelligence for optimization



Citation: Khanesar, M.A.; Yan, M.; Isa, M.; Piano, S.; Branson, D.T. Precision Denavit–Hartenberg Parameter Calibration for Industrial Robots Using a Laser Tracker System and Intelligent Optimization

Approaches. *Sensors* **2023**, *23*, 5368. <https://doi.org/10.3390/s23125368>

Academic Editor: Andrey V. Savkin

Received: 10 April 2023

Revised: 18 May 2023

Accepted: 30 May 2023

Published: 6 June 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Industrial robots are vital pieces of equipment for completing a modern manufacturing process. An industrial robot integrates several advanced technologies including mechanics, electronics, interfaces, and control in its structure [1]. To perform precision positioning [2], assembly, peg-in-hole [3–7], object handling, and manipulation on a factory floor [8], it is required to integrate accurate robot forward kinematics (FK) into its control methodology [9–11]. There exist some uncertainties in industrial robot FKs, which decrease its operational accuracy. Zero offset [12], irregularities in industrial robot geometry, robot manufacturing tolerances, assembly tolerances, structural deformations, and environmental factors are major sources of uncertainty in industrial robot FKs. Therefore, to increase the precision of an industrial robot FK, it is required to perform a calibration procedure [13,14]. The FK calibration is the procedure of improving the precision of the industrial robot Denavit–Hartenberg (DH) parameters to improve the robot FK model to have better positional accuracies [15,16]. The calibration of industrial robots is performed in multiple levels.

Three different calibration levels have already been identified for industrial robots [17]. The first calibration level (level I) is the one associated with robot joint angle encoders. The purpose of calibration in this level is to find a correcting relationship between the industrial robot joint angle readings from the installed joint angle transducer and the actual joint

angle values. The second level (level II) of calibration is the one associated with the FK model, which is mainly due to parameter uncertainties and industrial robot geometry. The purpose of the calibration at this level is to estimate the true FK parameters as well as to determine the basic kinematic geometry [17]. Non-kinematic calibration is the calibration at level III, which is mostly carried out to compensate for robot structural flexibilities as well as joint compliance, link compliance, friction, and clearance. The calibration is performed at level II in this paper; it is required to process real-time data from industrial robots and form independent measurement equipment to improve positioning precision. The heterogeneous information gathered from multiple measurement systems increases the perception capability of the overall calibration system. Industrial robot FK calibration can be performed using parametric robot calibration and non-parametric robot calibration.

The majority of industrial robot FK errors are due to structural parameter measurement errors [18,19]. There are two types of industrial robot calibration categories: parametric robot calibration and non-parametric robot calibration. Non-parametric robot calibration techniques include bilinear and fuzzy interpolation methods as well as neural network approaches. The 3D position calibration using non-parametric approaches with different non-contact metrology equipment such as the Leica SMART310 laser tracker, Leica AT960, and Leica AT960-MR has already been performed for a PA10 robot arm, IRB1410, and a collaborative industrial robot, respectively [20–22]. A similar approach is used in [15,20] for the calibration purpose of a Hyundai HH800 robot, a heavy duty industrial robot, using a laser tracker system. Deep neural networks have been used to calibrate a KUKA KR500, an SRA166 using an API Radian laser tracker, and a laser tracker manufactured by IHI Scube, respectively [23,24]. Motivated by the fact that parametric calibration algorithms do not necessitate the use of a highly complex system such as a neural network to obtain the 3D position of robots, in this paper, parametric calibration algorithms are used to have an estimation of the real physical parameters of industrial robots. Furthermore, using the parametric approaches, it is possible to come up with a more compact and less complex type of mathematical equation. To perform the parametric calibration, it is formulated as an optimization algorithm which is solved using four different optimization algorithms from the three main optimization categories of evolutionary optimization algorithms, namely, swarm intelligence and physic-inspired optimization algorithms.

Evolutionary optimization, swarm intelligence, and physic-inspired optimization algorithms are the three major intelligent optimization algorithms preferred for the case when the cost function is not defined in an explicit form [25]. In the case when the function to be optimized is non-differentiable and/or not explicitly defined, intelligent optimization approaches may be more preferable than classical optimization methods. Moreover, meta-heuristic approaches benefit from multiple start points and information exchange between the solutions, which makes them overcome the local minima problem. While evolutionary optimization algorithms rely on Darwin's natural selection principles [26], swarm intelligent [27,28] techniques imitate the chaotic and unpredictable motion behavior of species such as birds and fish in a swarm to perform optimization. An artificial bee colony (ABC) is another strong optimization approach which imitates the behavior of bees within their colony, and hence, they belong to swarm intelligence approaches. On the other hand, physic-inspired optimization algorithms use models of physical laws such as gravitational forces [29], coulomb forces [30], and quantum physics principles [31] to perform optimization. Inspired by the well-established results obtained from differential evolution [32] (DE), particle swarm optimization (PSO) [27,28], ABC [33,34], and gravitational search algorithms (GSA) [13], these four algorithms are used in this paper to perform industrial robot DH parameter optimization. Although the three main categories of optimization differ substantially in their metaphor, in this paper, they are used for the optimization purpose of a single cost function. A comparison is made between the results obtained from each of these optimization algorithms. Using a diverse range of intelligent optimization approaches makes the comparison results more reliable.

To perform the calibration of industrial robots, four optimization algorithms of DE, PSO, ABC, and GSA are utilized. The experimental setup in this paper consists of a laser tracker and collaborative robot. The laser tracker used in this study is a Leica AT960, a contactless metrology device capable of performing the 3D absolute position measurement of its retroreflector using laser interferometry principles. The positional precision of Leica AT960 is $3\ \mu\text{m}/\text{m}$ (https://www.hexagonmi.com/-/media/Hexagon%20MI%20Legacy/m1/metrology/general/brochures/Leica%20AT960%20brochure_en.ashx (accessed on 1 May 2022)), and its maximum measurement volume is up to 40 m. The absolute position measurements from the laser tracker are used to estimate the robot DH parameters. The joint angle readings from the industrial robot are used within its FK to calculate the industrial robot position. However, the positions calculated from the joint angles of the robot contain uncertainty because of the uncertainty in the industrial robot DH parameters. A cost function is defined based on the error between the FK robot positions and the measured position from a laser tracker. The data points selected for this experiment correspond to the static and near static motion of the industrial robot. Data are split to train and test to validate the accuracy of the results. The cost function is then optimized using the four optimization algorithms of DE, PSO, ABC, and GSA. DE belongs to the evolutionary optimization category of optimization, PSO and ABC belong to the swarm intelligence optimization category, and GSA is a physics-inspired optimization algorithm. The results of these optimizations give the precise DH parameters of the industrial robot. The comparison made between the results obtained using these algorithms reveals that ABC outperforms DE, PSO, and GSA in terms of the mean square value of the calibration error. It is further observed that, by using ABC, it is possible to decrease the mean absolute error of industrial robot positions for the test data from 75.4 to 60.1 μm , which is a 20.3% improvement in accuracy. The results reveal that ABC results in a superior performance compared to DE, PSO, and GSA for industrial robot DH parameter estimation purposes.

This paper is organized as follows: in Section 2, the overall methodology including an industrial robot FK, intelligent optimization methods, and the proposed calibration approach are introduced. The experiment setup used for measurements is presented in Section 3. Experimental results are presented in Section 4. Section 5 concludes the paper.

2. Methodology

The overall calibration process, which implements metaheuristic optimization approaches to perform calibration, is presented in this section (see Figure 1). Each element in this flowchart is presented in detail in the next few sections. Although joint angle readings from an industrial robot can be used to find its end effector position, uncertainties associated with industrial robot DH parameters, obtained from its geometry, result in positional inaccuracies. The manufacturing tolerance leads to differences in DH parameter values from one robot to another of a similar build. In this paper, a laser tracker system, Leica AT960-MR, is utilized to increase the accuracy of DH parameter values.

2.1. Data Preprocessing

This laser tracker is capable of measuring accurate non-contact robotic positions at a maximum distance of 60 m. In this work, the laser tracker is used within 2.8 m from the robot base. It uses laser interferometry principles and internal precise encoders for positioning. The 3D position readings from the robot are used for DH parameter calibrations using metaheuristic optimization approaches. The position reading from the industrial robot is conducted at a sampling frequency of 125 Hz, and the sampling frequency of the laser tracker system is 10 Hz. To perform calibration, it is required to synchronize the measurements from the industrial robot joint angle encoders and resample them at the same time instances as those for the laser tracker system. It is further required to extract samples occurring at a linear velocity of less than 2 mm/s using the nominal DH parameters of UR5 to have a quasi-static calibration for the robot. Overall, we have 209 data points which are split into train and test data using a 70/30 ratio. The resulting synchronized

data values are then used with a metaheuristic optimization approach to calibrate the DH parameters of industrial robots. The details of the overall calibration approach are presented in this section.

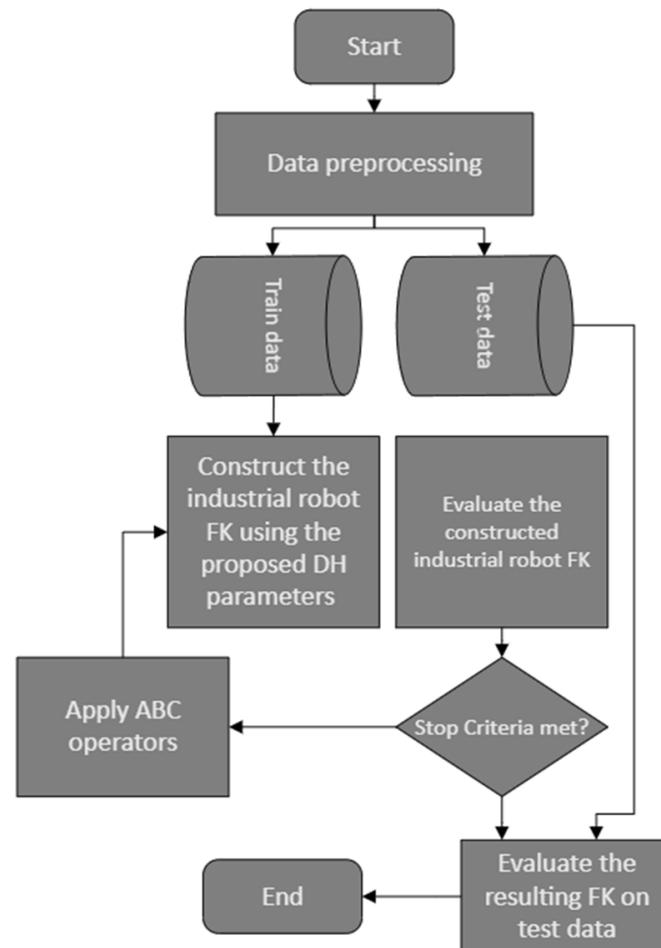


Figure 1. Overall calibration flowchart.

2.2. FK Model of UR5

The mathematical function operating on industrial robot joint angles to find the Cartesian coordinates of an industrial robot within a 3D workspace is called FKs. The link transformation matrix from the link $i - 1$ to the link i using the DH parameters of the robot, as in Table 1, depends on the corresponding joint angle of the industrial robot and its other DH parameters [35,36] (see Figure 2).

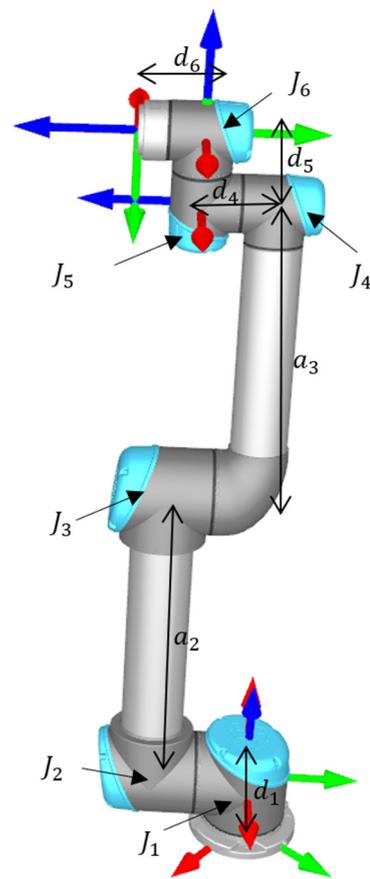
$${}_{i-1}T_i = \begin{bmatrix} cq_i & -\alpha_i sq_i & s\alpha_i sq_i & a_i cq_i \\ sq_i & \alpha_i cq_i & -s\alpha_i cq_i & a_i sq_i \\ 0 & s\alpha_i & c\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where $q_i/s, i = 1, \dots, 6$ represents the joint angle i , $\alpha_i/s, i = 1, \dots, 6$, $a_i/s, i = 1, \dots, 6$ and $d_i, i = 1, \dots, 6$ represents the other DH parameters of the robot. Furthermore, $cq_i, sq_i, c\alpha_i$, and $s\alpha_i, i = 1, \dots, 6$ represent $\cos(q_i), \sin(q_i), \cos\alpha_i$, and $\sin(\alpha_i), i = 1, \dots, 6$, respectively. Other than the DH parameters of the industrial robot in Figure 1, the labels $J_i, i = 1, \dots, 6$ represent the six joints of the industrial robot. Overall, the robot transformation matrix in the robot base coordinates is obtained as follows [37].

$$T_e = {}_0T = {}_0T_1 T_1 T_2 T_2 T_3 T_3 T_4 T_4 T_5 T_5 T_6 T_6 T \quad (2)$$

Table 1. The DH parameters of the 6DOF robot.

Link	q	d	a	α
1	q_1	d_1	0	$\pi/2$
2	q_2	0	a_2	0
3	q_3	0	a_3	0
4	q_4	d_4	0	$\pi/2$
5	q_5	d_5	0	$-\pi/2$
6	q_6	d_6	0	0

**Figure 2.** Typical industrial robot (UR5) schematics ($d_i, i = 1, 4, 5, 6$ and $a_i, i = 2, 3$ parameters refer to the DH parameters of the UR5).

The end effector positions in all three dimensions in the industrial robot coordinate are obtained as follows.

$$\begin{aligned}
 x_{rr} = & d_4s_1 + a_2c_1c_2 + d_6c_5s_1 + a_3c_1c_2c_3 - a_3c_1s_2s_3 + d_5c_1c_2c_3s_4 \\
 & + d_5c_1c_2s_3c_4 + d_5s_1s_2c_3c_4 - d_5c_1s_2s_3s_4 - d_6c_1c_2c_3c_4s_5 + d_6c_1c_2s_3s_4s_5 \\
 & + d_6c_1s_2c_3s_4s_5 + d_6c_1s_2s_3c_4s_5
 \end{aligned} \quad (3)$$

$$\begin{aligned}
 y_{rr} = & a_2s_1c_2 - d_6c_1c_5 - d_4c_1 + a_3s_1c_2c_3 - a_3s_1s_2s_3 + d_5s_1c_2c_3s_4 \\
 & + d_5s_1c_2s_3c_4 + d_5s_1s_2c_3c_4 - d_5s_1s_2s_3s_4 - d_6s_1c_2c_3c_4s_5 + d_6s_1c_2s_3s_4s_5 \\
 & + d_6s_1s_2c_3s_4s_5 + d_6s_1s_2s_3c_4s_5
 \end{aligned} \quad (4)$$

$$\begin{aligned}
 z_{rr} = & d_1 + a_2s_2 + a_3c_2s_3 + a_3s_2c_3 - d_5c_2c_3c_4 - d_5c_2s_3s_4 + d_5c_2s_3s_4 \\
 & + d_5s_2c_3c_4 + d_5s_2s_3c_4 - d_6c_2c_3s_4s_5 - d_6c_2s_3c_4s_5 - d_6s_2c_3c_4s_5 \\
 & + d_6s_2s_3s_4s_5
 \end{aligned} \quad (5)$$

Although the values of the FK parameters are unknown and will be estimated, their numerical values, according to the robot manufacturer, are as follows (<https://www.universal-robots.com/articles/ur/application-installation/dh-parameters-for-calculations-of-kinematics-and-dynamics/> (accessed on 1 May 2022)).

$$d_2 = d_3 = 0, \text{ and } a_1 = a_4 = a_5 = a_6 = 0 \quad (6)$$

$$\begin{aligned} d_1 &= 0.08916m, a_2 = -0.425m, a_3 = -0.392m, \\ d_4 &= 0.1092m, d_5 = 0.0947m, d_6 = 0.0823 + d \end{aligned} \quad (7)$$

where d is the distance between the center of the retroreflector and the center of the robot end-effector, which is measured as equal to $0.1695m$. Furthermore, to conduct the calibration, the end-effector of the robot is considered facing down, with its TCP axis-rotation vector being equal to $(\pi \ 0 \ 0)$.

2.3. Formulating the Estimation Problem as a Cost Function

It is required to define the industrial robot DH parameter estimation problem as a cost function to be optimized. This is a function of the industrial robot DH parameter and returns a quadratic function of the industrial robot FK error. Using each set of DH parameters given by each of the three optimization algorithms, DE, PSO, and GSA, the industrial robot FK is constructed. The positions given by the industrial robot FK are within its coordinate, which is the fixed coordinate attached to its base. The change in the coordinate is required to have the industrial robot positions in the laser tracker coordinates.

$$\begin{bmatrix} x_{rl} \\ y_{rl} \\ z_{rl} \\ 1 \end{bmatrix} = Tr_{rl} \begin{bmatrix} x_{rr} \\ y_{rr} \\ z_{rr} \\ 1 \end{bmatrix} \quad (8)$$

where x_{rl} , y_{rl} , and z_{rl} are the robot end-effector positions using a laser tracker in a laser tracker coordinate, and Tr_{rl} is the transformation matrix from the robot base coordinate system to the coordinate system of the laser tracker. The transformation matrix Tr_{rl} can be easily calculated using a least squares algorithm [38]. Using the calculated transformation matrix Tr_{rl} , the end effector positions in laser tracker coordinates are calculated as follows.

$$\begin{bmatrix} x'_{rl} \\ y'_{rl} \\ z'_{rl} \\ 1 \end{bmatrix} = Tr_{rl} \begin{bmatrix} x_{rr} \\ y_{rr} \\ z_{rr} \\ 1 \end{bmatrix} \quad (9)$$

where x'_{rl} , y'_{rl} , and z'_{rl} represent the robot end effector position in the x , y , and z axes using robot joint encoders in laser tracker coordinates. The position errors (er) can be calculated for each point as the difference between the robot end effector position using joint encoder data in the laser coordinate and the robot end effector measurements from the laser tracker.

$$er = \sqrt{(x'_{rl} - x_{rl})^2 + (y'_{rl} - y_{rl})^2 + (z'_{rl} - z_{rl})^2} \quad (10)$$

The overall mean value of position errors for all measured points associated with each of these industrial robot FKs needs to be calculated using (11). The mean absolute position errors over all measured points are used to construct the cost function for each object in the intelligent optimization algorithm.

$$C = \frac{1}{N_m} \sum_{i=1}^{N_m} er_i \quad (11)$$

where N_m is the total number of measured points. The metaheuristic optimization procedure is then used to find the optimal values of the industrial robot DH parameters.

2.4. Metaheuristic Optimization Algorithms

Metaheuristic approaches are generally designed to solve complex optimization problems where the cost function is not explicitly given and/or suffers from multiple local minima. In this paper, four metaheuristic optimization algorithms of DE, PSO, GSA, and ABC are briefly explained, and they are used to estimate industrial robots' DH parameters.

2.4.1. Differential Evolution

Evolutionary algorithms are frequently used as metaheuristic optimization approaches to solve complex optimization problems. These algorithms are inspired by the natural selection principles and the Darwinian theory of evolution. Individual members of the evolutionary algorithm represent solutions for the optimization problem. In each generation of these algorithms, operators such as crossover, mutation, and selection are applied to the individuals to generate the next generation. The selection operator selects the individuals for crossover or mutations using their fitness function and some random operators. Crossover is an operator that is applied to two or more selected individuals to generate new generations as their mathematical combination. Mutation, on the other hand, is an operator that acts as a single individual to generate a new individual. The mutation operator avoids premature convergence by mutating individuals.

Candidate solutions are presented by a six-dimensional vector $X_{DE}^i = (d_1^i, d_2^i, d_3^i, d_4^i, d_5^i, d_6^i)$, $i = 1, \dots, N$. Individuals representing solutions to the optimization problem need to be uniformly distributed between the permeable minimum and permeable maximum value of the solutions within each dimension. The mutation operator acts on the best individual by adding a term using two other randomly selected individuals from the population as follows [32]:

$$V_{DE}^i(t) = X_{DE,best}(t) + F \left(X_{DE}^{r_1^i}(t) - X_{DE}^{r_2^i}(t) \right) \quad (12)$$

where V_{DE}^i is the mutant vector, $X_{DE,best}$ is the best individual member of DE, and the parameter F is the scale factor, which is selected as equal to 0.5 in this paper. Furthermore, t refers to the number of generations. The indices r_1^i and r_2^i are mutually exclusive integers randomly generated within the range $[1, N]$, where N is the total number of individuals.

After performing the mutation operation, the crossover operation is applied to each individual and its mutant one [32].

$$U_{DE}^{i,j}(t) = \begin{cases} V_{DE}^{i,j}(t) & \text{if } \text{rand}_j[0,1] \leq CR \\ X_{DE}^{i,j}(t) & \text{otherwise} \end{cases}, j = 1, \dots, d \quad (13)$$

where $X_{DE}^{i,j}(t)$, $U_{DE}^{i,j}(t)$, and $V_{DE}^{i,j}(t)$ are the j -th element of $X_{DE}^i(t)$, $U_{DE}^i(t)$, and $V_{DE}^i(t)$, respectively. U_{DE}^i is the result of the crossover. The selection operator performs a comparison between the cost function associated with X_{DE}^i and U_{DE}^i , and the result of the selection procedure is used to perform the crossover and mutation to generate the next generation [32].

$$X_{DE}^i(t+1) = \begin{cases} U_{DE}^i(t) & \text{if } f(U_{DE}^i(t)) \leq f(X_{DE}^i(t)) \\ X_{DE}^i(t) & \text{otherwise} \end{cases}, j = 1, \dots, d \quad (14)$$

The mutation, crossover, and selection steps are repeated iteratively until the optimization termination condition of the maximum number of iterations is met. The maximum number of iterations considered in this paper is equal to 300.

2.4.2. Particle Swarm Optimization

The particle swarm optimization algorithm is based on swarm intelligence. This algorithm is developed by observing the behavior of fish and birds in a swarm [28]. The solutions to the optimization problem are presented as a position vector in a swarm. The velocity vector determines the change in the swarm position vector for the next iteration. The updates in the velocity vector are performed using the best personal experience of each

swarm member and the best experience over the whole swarm. A term preserving the inertia of movement exists in the velocity vector update equation to add more exploration features to the swarm. To apply PSO to calibrate industrial robot FKs, the position vector in PSO is associated with the DH parameters as follows [28]:

$$X_{PSO}^i = (d_1^i, a_2^i, a_3^i, d_4^i, d_5^i, d_6^i) \quad (15)$$

where X^i refers to the solutions within PSO and i refers to the i -th particle within the swarm.

The positions in the next generation of PSO using its current position vector and velocity vector are updated as follows [28]:

$$V_{PSO}^i(t+1) = wV_{PSO}^i(t) + r_1c_1(pbest_i(t) - X_{PSO}^i(t)) + r_2c_2(gbest_i(t) - X_{PSO}^i(t)), i = 1, \dots, N \quad (16)$$

$$X_{PSO}^i(t+1) = X_{PSO}^i(t) + V_{PSO}^i(t) \quad (17)$$

where t refers to the current iteration, $pbest_i(t)$ presents the personal best experience of the i -th particle, $gbest_i(t)$ represents the overall best experience within the swarm, $0 < c_1, c_2$ are the two positive constants, and r_1, r_2 are two uniform and random numbers from the interval of $[0, 1]$. The parameter c_1 is the coefficient associated with the best personal experiment of the particles in the swarm, and the parameter c_2 is the coefficient associated with the best global experiment of the particles within the swarm. The parameter w is the inertia weight, which makes the swarm follow its previous search direction. The stability criteria for PSO require the following condition to be valid for its parameters [39,40].

$$c_1 + c_2 < \frac{4(1 - 2w + w^2)}{1 + w}, 0 < w \quad (18)$$

It is further observed in [41] that while a large value for w improves exploration, a small value guarantees a good exploitation capability for PSO.

2.4.3. Artificial Bee Colony

The ABC algorithm is designed to imitate the principle of foraging, which explains how bees work together to search to find the best hive and maximize the colony yield [33]. In an ABC algorithm, employed bees, onlookers, and scout bees are the three groups of bees which exchange information to perform optimization [34]. The main step of this algorithm is summarized as follows.

1. initialize the population as $x_i, i = 1, \dots, SN$
2. calculate the fitness associated with each member of the population
3. repeat the following loop:
 - a. produce a new set of industrial robot DH parameters as the solutions for the optimization problem using the employed bee using $v_{ij} = x_{ij} + \varphi_{ij}(x_{ij} - x_{kj})$, $k \in 1, \dots, SN, j \in 1, \dots, 6$, where $\varphi_{ij} \in [0, 1]$ is a uniform random number
 - b. calculate the fitness function associated with each solution $f(x_i)$
 - c. for each solution, calculate its selection probability value as follows:

$$p_i = 0.04 \frac{f(x_i)}{\max_{i=1}^{SN} f(x_i)} + 0.96$$

- d. produce the new solutions v_i for the onlookers from the solutions x_i selected depending on p_i and evaluate them
- e. apply a greedy selection process for onlookers

- f. find possible abandoned food sources for scouts and replace them with a new food source using $x_{ij} = x_{i,min} + r_j(x_{i,max} - x_{i,min})$, where $r_j \in [0, 1]$ is a uniform and random number
- g. compare the best solution in this iteration with the overall best solution and replace it, if necessary
- h. if the maximum number of iterations (T_{max}) is achieved, stop; otherwise, continue the loop

2.4.4. Gravitational Search Algorithm

Other than the swarm intelligence and evolutionary optimization algorithms, there exists a third class of intelligent optimization approaches inspired by physics. GSA is an algorithm in this category inspired by the Newtonian gravitational forces between objects. In this algorithm, each solution is defined as a position of an object. The object positions are updated using velocity and acceleration vectors. Each object in this optimization algorithm benefits from a mass property. The larger mass value is assigned to the solution with a better cost function. The acceleration vector at each iteration is updated such that the objects with smaller weights are accelerated towards the ones with larger weights. This makes the objects with a smaller mass value scan the solution space towards the objects with a larger mass value. After a few iterations, all solutions converge to the object with a larger mass value, which will conclude the optimization. The mathematical description of the algorithm is explained within the next paragraphs.

Each object in this algorithm benefits from several properties of mass, positions, velocity, and accelerations. Solutions in a six-dimensional solution space are represented by object positions.

$$X_{GSA}^i = (d_1^i, a_2^i, a_3^i, d_4^i, d_5^i, d_6^i), \quad i = 1, \dots, N \quad (19)$$

The mass value corresponding to the i th particle at iteration number t is called the non-normalized mass value and is represented by $m_i(t)$ [13].

$$m_{GSA}^i(t) = \frac{f(X_{GSA}^i) - f_{worst}(t)}{f_{best}(t) - f_{worst}(t)} \quad (20)$$

where $f_{worst}(t)$ is the overall worst fitness function value and $f_{best}(t)$ represents the overall best fitness function value. Therefore, $m_{GSA}^i(t)$ satisfies $m_{GSA}^i(t) \in [0, 1]$, with the mass value corresponding to the best solution being equal to *one* and the mass value corresponding to the worst solution being equal to *zero*. The parameters $f_{worst}(t)$ and $f_{best}(t)$ are updated at every iteration as follows.

$$f_{worst}(t) = \max\{f(X_{GSA}^i(t))\}_{i=1,\dots,N} \quad (21)$$

$$f_{best}(t) = \min\{f(X_{GSA}^i(t))\}_{i=1,\dots,N} \quad (22)$$

where x_i^j is the j th component of the particle position and N is the total number of particles. Each particle mass is updated and normalized at t th [13].

$$M_{GSA}^i(t) = \frac{m_{GSA}^i(t)}{\sum_{k=1}^N m_{GSA}^k(t)} \quad (23)$$

The overall gravitational force ($F_{GSA}^i(t)$) acting on the i th particle is calculated using the gravitational law of force [13].

$$F_{GSA}^i(t) = \sum_{j \in \{1, \dots, k_b\}} r_j G(t) \frac{M_{GSA}^j(t) M_{GSA}^i(t) (X_{GSA}^j(t) - X_{GSA}^i(t))}{\|X_{GSA}^i(t) - X_{GSA}^j(t)\|^{r_p + \varepsilon}} \quad (24)$$

where k_b is the number of selected best solutions, $\|\cdot\|$ represents the Euclidean norm, ε is a small value added to prevent singularity, r_p is the power value for the Euclidean distance between the two particles, $G(t)$ is the gravitational constant, and $r_j \in [0, 1]$ is a uniform and random value. The gravitational constant is updated at each iteration using the following equation [13].

$$G(t) = G_0 \exp\left(-\beta \frac{t}{t_{max}}\right) \quad (25)$$

where G_0 has a constant real value and t_{max} is the maximum value of the iterations of the algorithm. The acceleration term for each object is calculated according to Newton's second law of motion by dividing the applied force to the i th mass by its mass value [13].

$$A_{GSA}^i(t) = \frac{F_{GSA}^i(t)}{M_{GSA}^i(t)} = \sum_{j \in \{1, \dots, k_b\}} r_j G(t) \frac{M_{GSA}^j(t) (X_{GSA}^j(t) - X_{GSA}^i(t))}{\|X_{GSA}^i(t) - X_{GSA}^j(t)\|^{r_p + \varepsilon}} \quad (26)$$

where $A_{GSA}^i(t) \in R^d$ is the d -dimensional acceleration of the particles. The velocity value corresponding to each object is updated using the acceleration term and velocity vector [13].

$$X_{GSA}^i(t+1) = X_{GSA}^i(t) + V_{GSA}^i(t+1) \quad (27)$$

3. Hardware Setup for the Experiment

The hardware setup for performing the experiment is explained under this section. To perform the calibration test, we need the industrial robot and independent calibration equipment, as shown in Figure 3. A detailed explanation of the hardware required to perform the calibration is explained in this section.

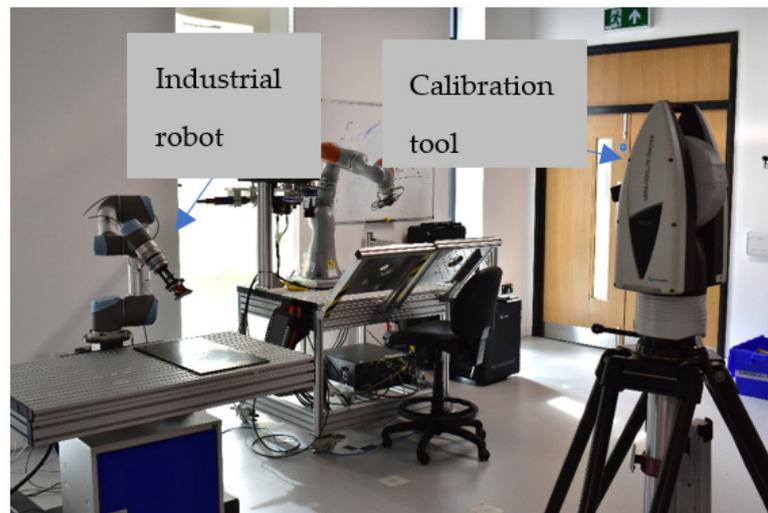


Figure 3. UR5 and the laser tracker.

3.1. Industrial Robot: UR5

The Universal Robots—UR5 is a six-degrees-of-freedom (DOF) industrial robot capable of handling a 5 kg load with an angular velocity of $180^\circ/\text{s}$ and an angular acceleration of $180^\circ/\text{s}^2$. The total reach of this robot without its grippers is 850 mm. Joint angle encoders

and current measurements are performed in each joint. Joint angle measurements are used as the input to the robot FK to estimate its positions and orientations. However, the nominal position repeatability of UR5 is 0.1 mm. To collect joint angle values from UR5, its main controller is connected to a PC via LAN connectivity and a hub. The PC used for this experiment operates under Linux 18.04 OS, and the software interface for the robot is provided by ROS Melodic. The ROS driver that provides the connectivity is available through a GitHub webpage (https://github.com/UniversalRobots/Universal_Robots_ROS_Driver (accessed on 9 April 2023)). This ROS driver publishes a range of rostopics including joint angle values, joint angular velocities, and motor currents over time @125 Hz. In total, 38 waypoints are programmed for the UR5, which travels them linearly in 600 s. Figure 4 illustrates the robot joint angle values over time for all six joints of UR5. The position data gathered from the robot are resampled @10 Hz to match the laser tracker frequency.

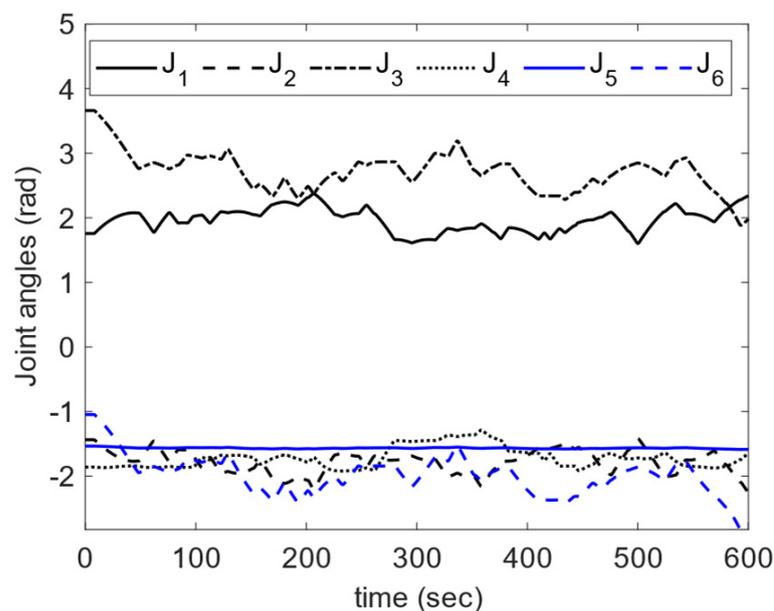


Figure 4. UR5 joint angle trends.

3.2. Laser Tracker System

The laser tracker used in this experiment is AT960-MR, which is a portable dynamic 6DOF laser measurement system manufactured by Hexagon metrology GMBH, Wetzlar, Germany. This metrology equipment benefits from a single-class II laser source and is an IEC certified IP54 which guarantees ingress protection against dust and other contaminants for the unit. It benefits from Wifi connectivity, and in this work, it is set up to collect 10 data samples per second. Distance measurements are performed by using a retroreflector as the measurement target mounted on the UR5 end effector. This type of measurement system is a widely used one for precisely inspecting critical distances, locations, and surfaces [42] (see Figure 5). The target for the laser tracker is a precision Leica 1.5" red ring reflector detectable through the laser tracker at a maximum distance of 60 m. In this work, the laser tracker is used within 2.8 m from the robot base. The reflector used in this experiment is using the principle of the corner cube. To reflect the beam, three plane mirrors at right angles to one another are used. The measurement point is the center of the reflector. To perform measurements, it is required to have a clear line of sight between the laser source and the retroreflector as the laser target.

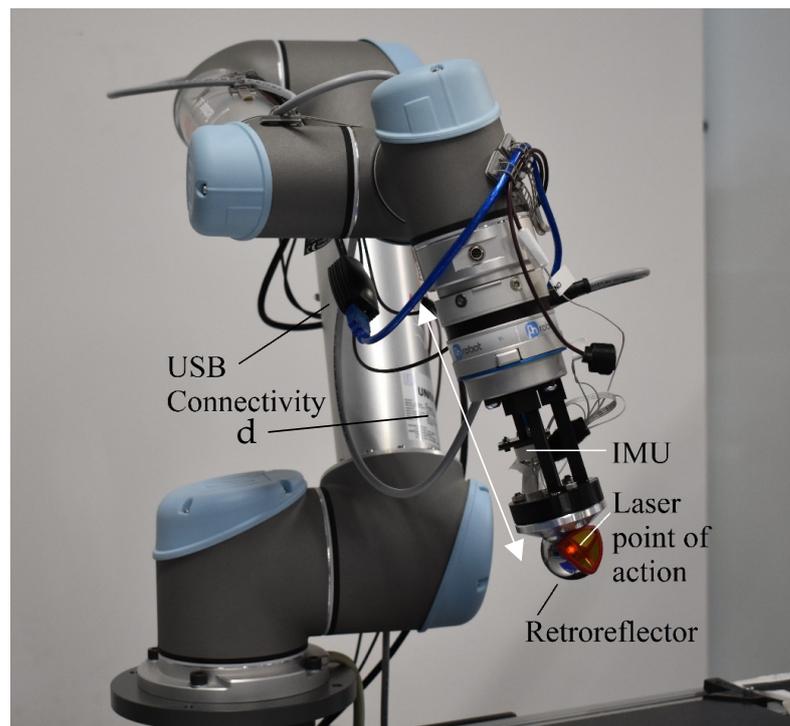


Figure 5. UR5 with a retroreflector mounted on it as the target for the laser tracker.

Further specifications and environmental conditions of the laser tracker are presented in Table 2.

Table 2. Measuring equipment characteristics and specifications.

Environmental Working Conditions	IP54: The IEC-Certified Sealed Unit Guarantees Ingress Protection against Dust and Other Contaminants
Operating temperature	Wide operating temperature range of -15 to 45 degrees Celsius
Temperature compensation	MeteoStation: Integrated environmental unit monitors conditions including temperature, pressure, and humidity to compensate for changes.
ISO certification	ISO 17025
Connectivity	Wi-Fi and LAN
Detector features	Red ring reflector— $1.5''$ radius: $19.05 \text{ mm} \pm 0.0025 \text{ mm}$, centering of optics: $< \pm 0.003 \text{ mm}$, ball roundness: $\leq 0.003 \text{ mm}$, acceptance angle: $\pm 30^\circ$, weight: 170 gr
Data output rate	Measurement rate of up to 1000 points per s
Distance accuracy	40 m in diameter and a 6DoF measuring volume of up to 20 m
Laser safety	Laser class 2
Distance to robot base origin	2.8 m

4. Experimental Results

4.1. Data Gathering

To perform level II calibration on the industrial robot using the laser tracker system, the data flow graph as demonstrated in Figure 1 is followed. While data are collected from the laser tracker using Wi-Fi, the LAN network is used for collecting data from UR5. The ROS melodic package under Linux 18.04 is used to collect joint data including joint angle values, joint angular velocities, and joint efforts from UR5 @ $\sim 125 \text{ Hz}$. The laser tracker data are collected using Spatial Analyzer 2019 software under a Windows 10 operating system (see Figure 6). It is required to shift, synchronize, and resample joint angle data for UR5

to match laser tracker data samples. Furthermore, to extract a near static data sample, the angular velocities less than 2 mm/s are considered. The total number of samples satisfying this angular velocity constraint is 209 points, from which 70% is used for training and the rest is used for testing purposes. The ROS melodic package under Linux 18.04 is used to collect joint data including joint angle values, joint angular velocities, and joint efforts from UR5 @ ~125 Hz. The laser tracker data are collected using Spatial Analyzer software under the Windows 10 operating system (see Figure 7). The connectivity required to gather the data from the laser tracker using SA and the data from UR5 using ROS melodic are illustrated in Figure 8.

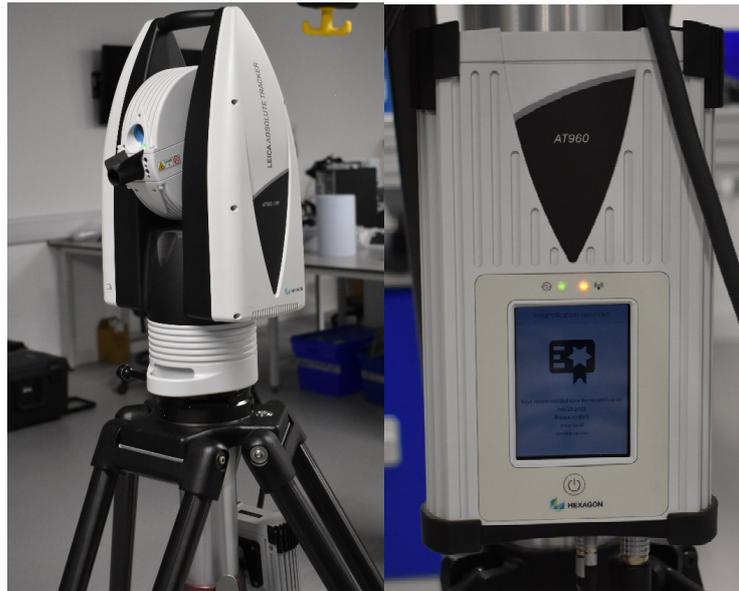


Figure 6. Laser tracker and its controller.

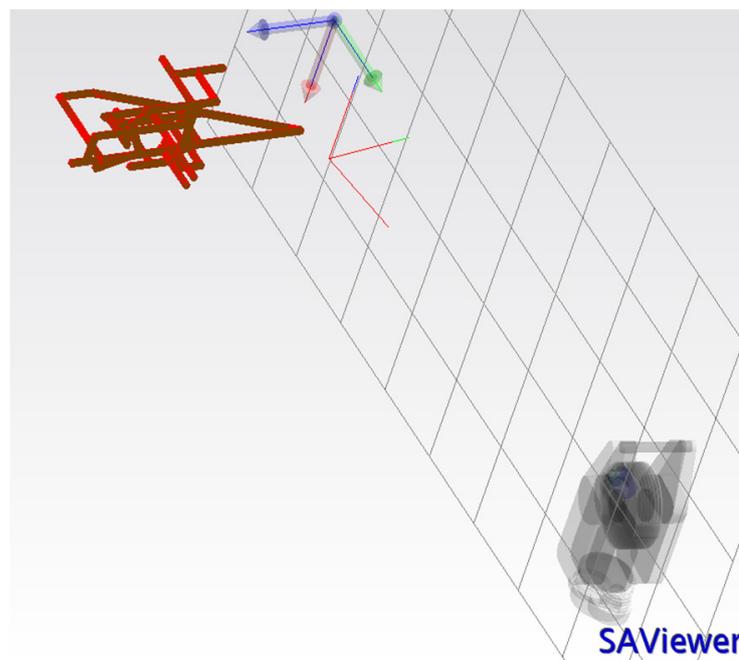


Figure 7. Screenshot of the points measured by the laser tracker system in Spatial Analyzer 2019 software.

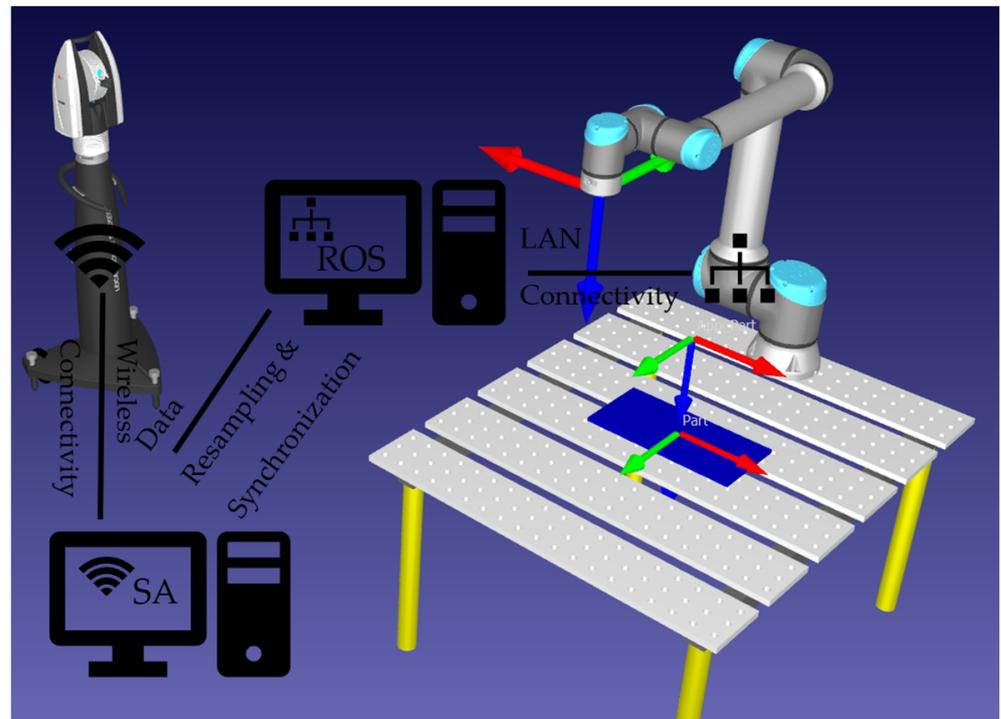


Figure 8. Connectivity required to perform the experiment.

4.2. Performance Measurement

There exist different performance indexes for validating the accuracy performance of models. In this paper, the mean absolute error (MAE) and standard deviation are used for performance measurement evaluation. These performance indexes are defined mathematically as follows.

$$MAE_i = \frac{1}{N} \sum_{k=1}^N |A_{ik} - F_{ik}|, i = x, y, z \quad (28)$$

$$\sigma_i = \sqrt{\frac{1}{N} \sum_{k=1}^N (A_{ik} - F_{ik})^2}, i = x, y, z \quad (29)$$

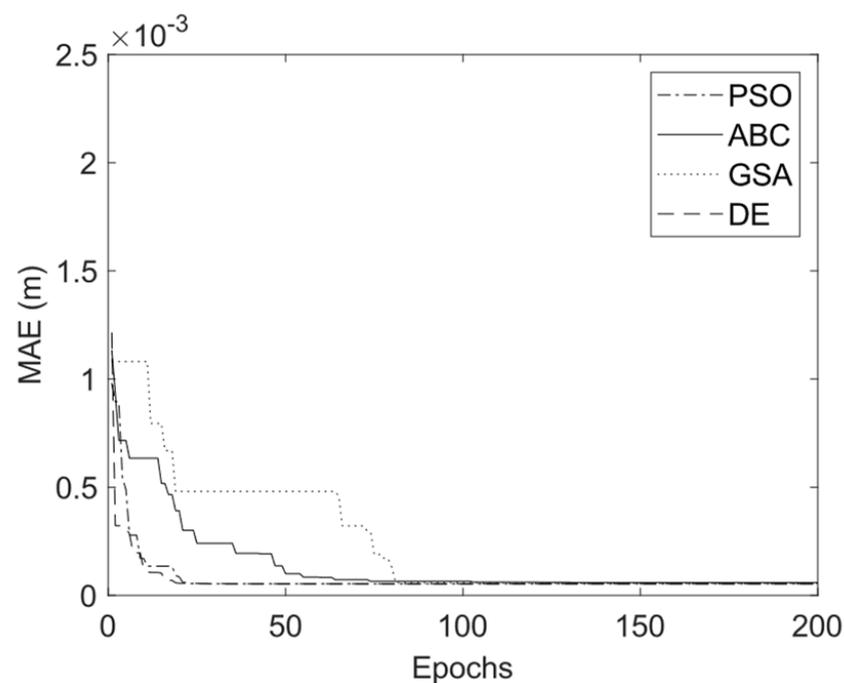
where N is the number of samples, σ_i presents the standard deviations in each dimension, A_{ik} represents the data measured by the laser tracker in each of the three positional dimensions, and F_{ik} represents the measurements by the FK of the robot in laser tracker coordinates in each of the three positional dimensions: x , y , and z .

4.3. Results

The parameters chosen for the three intelligent optimization algorithms of DE, PSO, GSA, and ABC are provided in Table 3. To have a better comparison, the population size for all three algorithms and the number of iterations are kept the same. Figure 9 demonstrates the convergence graph for all these optimization algorithms versus the iteration number. It is observed from the figure that DE converges faster than the other optimization algorithms of ABC, GSA, and PSO.

Table 3. Optimization parameters.

Algorithm	Parameter	Value
DE	F	0.55
	CR	2
	N	150
	$max\ generation$	300
GSA	r_p	1
	ϵ	2.22×10^{-16}
	β	20
	k_b	2
	t_{max}	300
ABC	N	150
	S_N	150
	T_{max}	300

**Figure 9.** Convergence trend for PSO, ABC, GSA, and DE.

The calibration results for the four optimization algorithms of DE, GSA, PSO, and ABC are demonstrated in Figures 10–12. As can be seen from these figures, the calibrated positions obtained through the four optimization algorithms are closer to the 3D precise measurements performed by the laser tracker in this experiment. The numerical values presented in Table 4 demonstrate the improvement made using the proposed calibration method. The MAE values associated with the calibrated FK using ABC in all three dimensions are reduced with respect to the uncalibrated version. The mean absolute value of error is reduced from 75.4 to 60.1 μm for the calibrated FK using ABC, which is a 20.3% improvement. It is further observed that the standard deviation of error is reduced from 100.6 to 76.1 μm . The improvement for the variance of error is 24.4%. The error trend associated with the industrial robot FK tuned by ABC as well as that of the original industrial robot FK are depicted in Figures 13–15. As can be seen from the figures, error

values for the calibrated FK are closer to zero as compared to the uncalibrated version in all three dimensions.

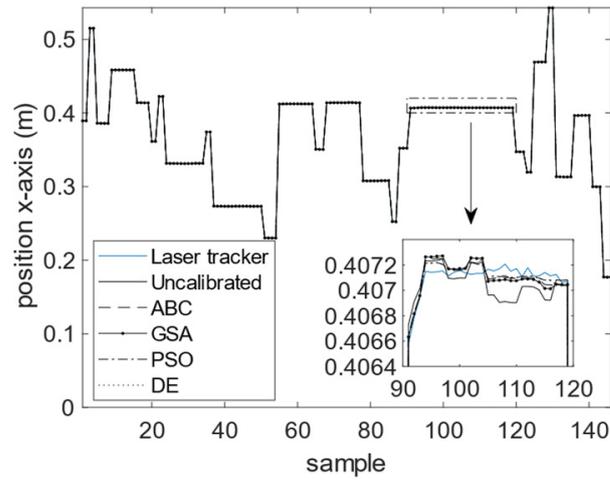


Figure 10. Calibration results in the x -dimension.

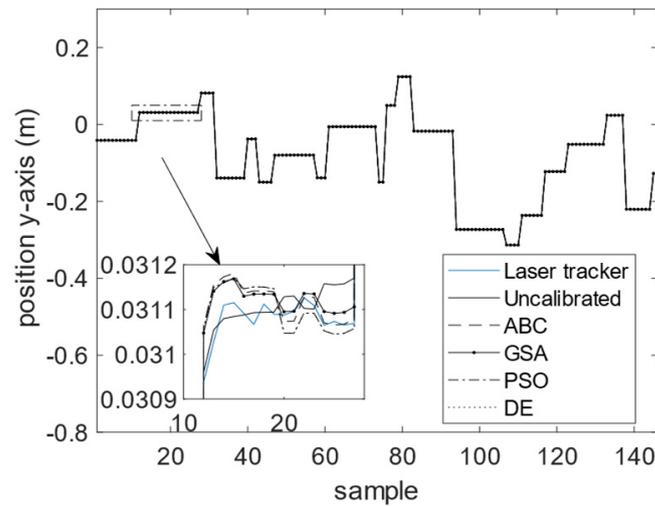


Figure 11. Calibration results in the y -dimension.

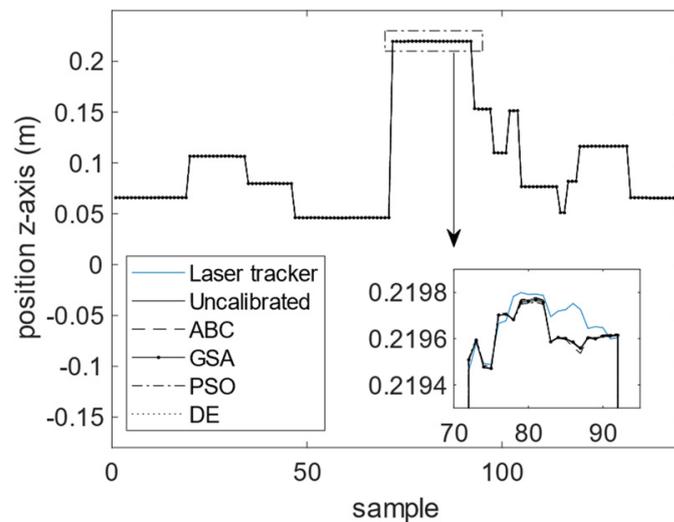
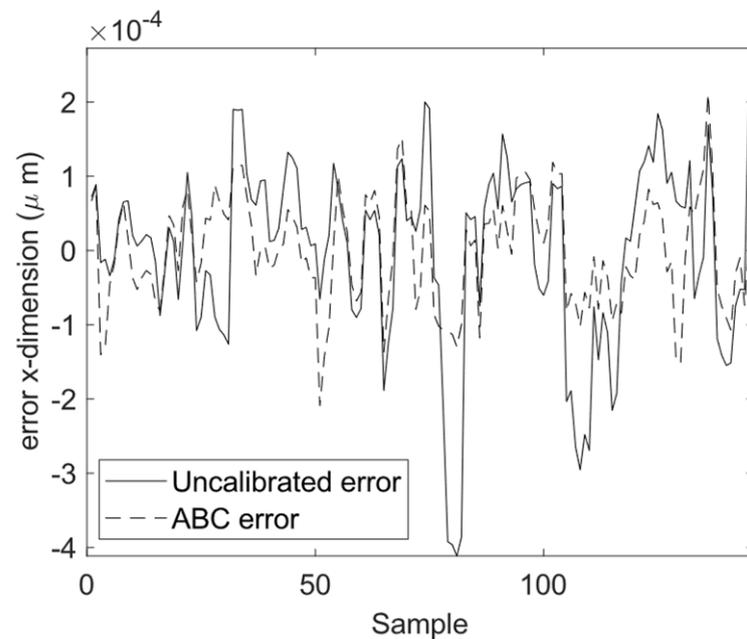


Figure 12. Calibration results in the z -dimension.

Table 4. Mean absolute error values.

Performance Indexes		σ_i		MAE	
		Train	Test	Train	Test
Uncalibrated (μm)	X	125.5	117.8	95.9	90.3
	Y	94.3	105.1	64.0	77.5
	Z	64.2	73.5	50.7	58.4
	3D	97.9	100.6	70.2	75.4
Calibrated (μm) Using ABC	X	74.5	80.0	62.0	64.7
	Y	75.3	73.9	53.2	56.7
	Z	64.2	74.2	50.4	58.9
	3D	71.5	76.1	55.2	60.1
Calibrated (μm) Using GSA	X	79.9	86.0	65.8	71.1
	Y	70.9	66.8	51.3	52.6
	Z	64.1	74.2	50.7	59.0
	3D	71.9	76.1	56.0	60.9
Calibrated (μm) Using PSO	X	72.3	80.0	59.5	64.5
	Y	72.8	76.8	50.8	60.2
	Z	63.4	74.6	50.1	59.9
	3D	69.6	77.2	53.5	61.5
Calibrated (μm) Using DE	X	72.3	80.0	59.5	64.5
	Y	72.7	76.7	50.8	60.2
	Z	63.4	74.6	50.1	59.9
	3D	69.6	77.2	53.5	61.5

**Figure 13.** Calibration error in the x-dimension.

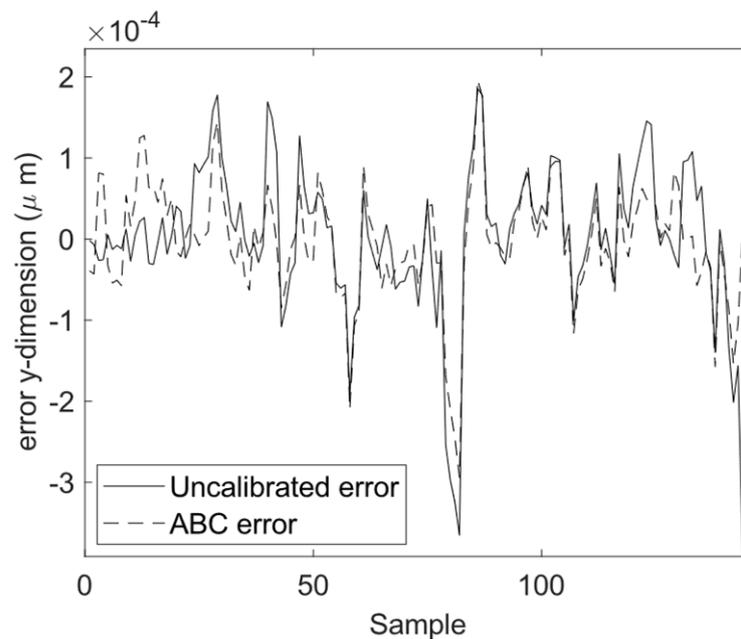


Figure 14. Calibration error in the y -dimension.

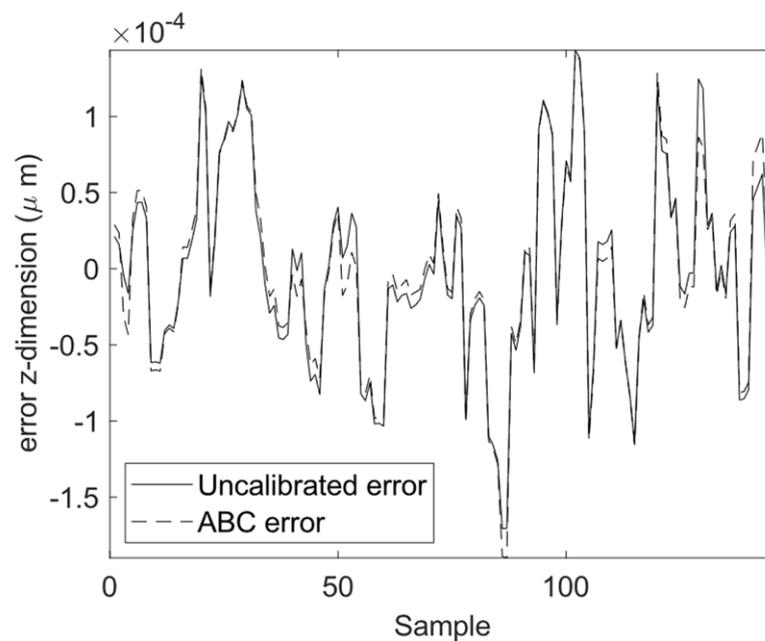


Figure 15. Calibration error in the z -dimension.

5. Conclusions

The sources of uncertainty in industrial robot FKs include manufacturing and assembly tolerances, dimension measurement errors, and environmental conditions. This paper deals with identifying more precise robot DH parameters to increase the positional accuracies of industrial robots. The proposed approach is a parametric calibration approach using a laser tracker. Therefore, not only is the proposed approach a precise approach for positioning robots, but the parameter values obtained using this approach also have physical meanings. To perform the calibration, a laser tracker system which is a non-contact metrology device is used. The laser tracker used in this experiment is a Leica AT960, with precision up to $3 \mu\text{m}/\text{m}$, performing measurements 2.8 m away from the robot base. The industrial robot used in this experiment is a UR5, an industrial robot manufactured by Universal Robots.

Using the four optimization algorithms of DE, PSO, ABC, and GSA, the tuning of industrial robot DH parameters is performed. It is observed that ABC outperforms DE, PSO, and GSA in terms of increasing the precision of robots. It is further observed that using the calibration approach proposed in this paper benefits from the ABC optimization algorithm results in decreasing the mean absolute value of error for the test data from 75.4 to 60.1 μm , which is a 20.3% improvement.

6. Future Works

In this paper, it is shown that the more accurate DH parameters can improve the accuracy of industrial robot FKs. The further accuracy improvement of industrial robots in future work requires online precision position feedback from measurement equipment and advanced computational inverse kinematic algorithms such as damped least squares. To increase the accuracy and convergence speed of the damped least squares algorithm, the DH parameter values obtained in this paper will be used. Furthermore, since the continuous monitoring of calibration and its validity over time is required, a mechanism for calibration monitoring is introduced to keep the position error within a valid interval. Such calibration monitoring system requires the continuous inspection and evaluation of the robot movements and positioning. This proposed approach will be implemented on more industrial robots to test the efficacy, performance, and implementability of the proposed methodology over a large number of industrial robots. The generalization of the proposed approach over a large number of industrial robots makes it possible to have statistical analysis of it and investigate the expected performance improvement using this approach in a more general form. Moreover, the calibration of industrial robots when the robot is operated at high speeds will also be considered. Contact approaches for industrial robot calibration [43,44] are usually lower-cost approaches for estimating more precise DH parameters of industrial robots. Motion sensors are another type of sensor which have already contributed to the precise calibration of industrial robots [45,46]. These types of sensors may perform orientation calibration with higher performance. The fusion of contact sensors for position measurements and inertia measurement sensors for orientation calibration would be considered in a future study.

Author Contributions: Conceptualization M.A.K., M.Y., M.I., S.P. and D.T.B.; Data curation: M.A.K.; methodology, M.A.K.; software, M.A.K.; validation, M.A.K., formal analysis, M.A.K.; investigation, M.A.K.; resources, M.A.K.; writing—original draft preparation, M.A.K. Writing—review and editing, M.A.K., M.Y., M.I., S.P. and D.T.B.; visualization, M.A.K. supervision, S.P. and D.T.B.; project administration, S.P. and D.T.B.; funding acquisition, S.P. and D.T.B. All authors have read and agreed to the published version of the manuscript.

Funding: This work is primarily funded and supported by the Engineering and Physical Sciences Research Council (EPSRC) under grant number: EP/T023805/1—High-accuracy robotic system for precise object manipulation (HARISOM). The authors would also like to thank the UKRI Research England Development (RED) Fund for partial funding this work via the Midlands Centre for Data Driven Metrology. The APC for this article is covered by M.A.K.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data would be available upon request on a personal contact with the corresponding author at the email address: mojtaba.ahmadielhkhanesar@nottingham.ac.uk.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Li, Z.; Li, S.; Luo, X. An overview of calibration technology of industrial robots. *IEEE/CAA J. Autom. Sin.* **2021**, *8*, 23–36. [[CrossRef](#)]
2. Szybicki, D.; Obal, P.; Kurc, K.; Gierlak, P. Programming of Industrial Robots Using a Laser Tracker. *Sensors* **2022**, *22*, 6464. [[CrossRef](#)] [[PubMed](#)]

3. Kim, Y.-L.; Song, H.-C.; Song, J.-B. Hole detection algorithm for chamferless square peg-in-hole based on shape recognition using F/T sensor. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 425–432. [[CrossRef](#)]
4. Luo, R.C.; Chang, A.; Li, C. A novel peg-in-hole approach based on geometrical analysis for inclined uncertainty. In Proceedings of the 2017 IEEE International Conference on Advanced Intelligent Mechatronics (AIM), Munich, Germany, 3–7 July 2017; IEEE: Piscataway, NJ, USA; pp. 891–896.
5. Zhang, X.; Zheng, Y.; Ota, J.; Huang, Y. Peg-in-hole assembly based on two-phase scheme and f/t sensor for dual-arm robot. *Sensors* **2017**, *17*, 2004. [[CrossRef](#)] [[PubMed](#)]
6. Park, H.; Park, J.; Lee, D.-H.; Park, J.-H.; Baeg, M.-H.; Bae, J.-H. Compliance-based robotic peg-in-hole assembly strategy without force feedback. *IEEE Trans. Ind. Electron.* **2017**, *64*, 6299–6309. [[CrossRef](#)]
7. Li, R.; Qiao, H. Condition and strategy analysis for assembly based on attractive region in environment. *IEEE/ASME Trans. Mechatron.* **2017**, *22*, 2218–2228. [[CrossRef](#)]
8. Farhadi, A.; Lee, S.K.; Hinchy, E.P.; O’Dowd, N.P.; McCarthy, C.T. The Development of a Digital Twin Framework for an Industrial Robotic Drilling Process. *Sensors* **2022**, *22*, 7232. [[CrossRef](#)] [[PubMed](#)]
9. Khanesar, M.A.; Branson, D. Robust Sliding Mode Fuzzy Control of Industrial Robots Using an Extended Kalman Filter Inverse Kinematic Solver. *Energies* **2022**, *15*, 1876. [[CrossRef](#)]
10. Olabi, A.; Damak, M.; Bearee, R.; Gibaru, O.; Leleu, S. Improving the accuracy of industrial robots by offline compensation of joints errors. In Proceedings of the 2012 IEEE International Conference on Industrial Technology, Athens, Greece, 19–21 March 2012; IEEE: Piscataway, NJ, USA; pp. 492–497.
11. Li, R.; Qiao, H. A survey of methods and strategies for high-precision robotic grasping and assembly tasks—Some new trends. *IEEE/ASME Trans. Mechatron.* **2019**, *24*, 2718–2732. [[CrossRef](#)]
12. Chen, H.; Fuhlbrigge, T.; Choi, S.; Wang, J.; Li, X. Practical industrial robot zero offset calibration. In Proceedings of the 2008 IEEE International Conference on Automation Science and Engineering, Arlington, VA, USA, 23–26 August 2008; IEEE: Piscataway, NJ, USA; pp. 516–521.
13. Joubair, A.; Bonev, I.A. Kinematic calibration of a six-axis serial robot using distance and sphere constraints. *Int. J. Adv. Manuf. Technol.* **2015**, *77*, 515–523. [[CrossRef](#)]
14. Kana, S.; Gurnani, J.; Ramanathan, V.; Turlapati, S.H.; Ariffin, M.Z.; Campolo, D. Fast kinematic re-calibration for industrial robot arms. *Sensors* **2022**, *22*, 2295. [[CrossRef](#)] [[PubMed](#)]
15. Nguyen, H.-N.; Zhou, J.; Kang, H.-J. A calibration method for enhancing robot accuracy through integration of an extended Kalman filter algorithm and an artificial neural network. *Neurocomputing* **2015**, *151*, 996–1005. [[CrossRef](#)]
16. Gao, G.; Li, Y.; Liu, F.; Han, S. Kinematic calibration of industrial robots based on distance information using a hybrid identification method. *Complexity* **2021**, *2021*, 8874226. [[CrossRef](#)]
17. Roth, Z.; Mooring, B.; Ravani, B. An overview of robot calibration. *IEEE J. Robot. Autom.* **1987**, *3*, 377–385. [[CrossRef](#)]
18. Judd, R.; Knasinski, A. A technique to calibrate industrial robots with experimental verification. In Proceedings of the 1987 IEEE International Conference on Robotics and Automation, Raleigh, NC, USA, 31 March–3 April 1987; IEEE: Piscataway, NJ, USA; Volume 4, pp. 351–357.
19. Xuan, J.-Q.; Xu, S.-H. Review on kinematics calibration technology of serial robots. *Int. J. Precis. Eng. Manuf.* **2014**, *15*, 1759–1774.
20. Aoyagi, S.; Kohama, A.; Nakata, Y.; Hayano, Y.; Suzuki, M. Improvement of robot accuracy by calibrating kinematic model using a laser tracking system-compensation of non-geometric errors using neural networks and selection of optimal measuring points using genetic algorithm. In Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems, Taipei, Taiwan, 18–22 October 2010; IEEE: Piscataway, NJ, USA; pp. 5660–5665.
21. Bai, M.; Zhang, M.; Zhang, H.; Li, M.; Zhao, J.; Chen, Z. Calibration Method Based on Models and Least-Squares Support Vector Regression Enhancing Robot Position Accuracy. *IEEE Access* **2021**, *9*, 136060–136070. [[CrossRef](#)]
22. Duong, Q.K.; Trang, T.T.; Pham, T.L. Robot Control Using Alternative Trajectories Based on Inverse Errors in the Workspace. *J. Robot.* **2021**, *2021*, 9995787. [[CrossRef](#)]
23. Hu, J.; Hua, F.; Tian, W. Robot positioning error compensation method based on deep neural network. *J. Phys. Conf. Ser.* **2020**, *1487*, 012045. [[CrossRef](#)]
24. Kato, D.; Yoshitugu, K.; Maeda, N.; Hirogaki, T.; Aoyama, E.; Takahashi, K. Finding Features of Positioning Error for Large Industrial Robots Based on Convolutional Neural Network. In Proceedings of the International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, Online, 17–19 August 2021; American Society of Mechanical Engineers: New York, NY, USA; Volume 85376, p. V002T02A022.
25. Biswas, A.; Mishra, K.; Tiwari, S.; Misra, A. Physics-inspired optimization algorithms: A survey. *J. Optim.* **2013**, *2013*, 438152. [[CrossRef](#)]
26. Gen, M.; Cheng, R. *Genetic Algorithms and Engineering Optimization*; John Wiley & Sons: Hoboken, NJ, USA, 1999.
27. Eberhart, R.C.; Shi, Y.; Kennedy, J. *Swarm Intelligence*; Elsevier: Amsterdam, The Netherlands, 2001.
28. Yang, X.-S.; Deb, S.; Zhao, Y.-X.; Fong, S.; He, X. Swarm intelligence: Past, present and future. *Soft Comput.* **2018**, *22*, 5923–5933. [[CrossRef](#)]
29. Rashedi, E.; Nezamabadi-Pour, H.; Saryazdi, S. GSA: A gravitational search algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [[CrossRef](#)]
30. Kaveh, A.; Talatahari, S. A novel heuristic optimization method: Charged system search. *Acta Mech.* **2010**, *213*, 267–289. [[CrossRef](#)]

31. Sun, J.; Xu, W.; Feng, B. A global search strategy of quantum-behaved particle swarm optimization. In Proceedings of the IEEE Conference on Cybernetics and Intelligent Systems, Singapore, 1–3 December 2004; IEEE: Piscataway, NJ, USA; Volume 1, pp. 111–116.
32. Mallipeddi, R.; Suganthan, P.N. Differential evolution algorithm with ensemble of parameters and mutation and crossover strategies. In Proceedings of the International Conference on Swarm, Evolutionary, and Memetic Computing, Chennai, India, 16–18 December 2010; Springer: Berlin/Heidelberg, Germany; pp. 71–78.
33. Karaboga, D.; Basturk, B. On the performance of artificial bee colony (ABC) algorithm. *Appl. Soft Comput.* **2008**, *8*, 687–697. [[CrossRef](#)]
34. Karaboga, D.; Gorkemli, B.; Ozturk, C.; Karaboga, N. A comprehensive survey: Artificial bee colony (ABC) algorithm and applications. *Artif. Intell. Rev.* **2014**, *42*, 21–57. [[CrossRef](#)]
35. Kufieta, K. Force Estimation in Robotic Manipulators: Modeling, Simulation and Experiments. Master's Thesis, Department of Engineering Cybernetics NTNU Norwegian University of Science and Technology, Trondheim, Norway, 2014.
36. Sun, J.-D.; Cao, G.-Z.; Li, W.-B.; Liang, Y.-X.; Huang, S.-D. Analytical inverse kinematic solution using the DH method for a 6-DOF robot. In Proceedings of the 2017 14th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Jeju, Republic of Korea, 28 June–1 July 2017; IEEE: Piscataway, NJ, USA; pp. 714–716.
37. Khanesar, M.A.; Yan, M.; Syam, W.P.; Piano, S.; Leach, R.K.; Branson, D.T. A Neural Network Separation Approach for the Inclusion of Static Friction in Nonlinear Static Models of Industrial Robots. *IEEE/ASME Trans. Mechatron.* **2023**, 1–11. [[CrossRef](#)]
38. Fotiou, A.; Kaltsikis, C. Computationally efficient methods and solutions with least squares similarity transformation models. Retrieved Jan. **2016**, 2, 2018.
39. Dong, W.Y.; Zhang, R.R. Order-3 stability analysis of particle swarm optimization. *Inf. Sci.* **2019**, *503*, 508–520. [[CrossRef](#)]
40. Kadirkamanathan, V.; Selvarajah, K.; Fleming, P.J. Stability analysis of the particle dynamics in particle swarm optimizer. *IEEE Trans. Evol. Comput.* **2006**, *10*, 245–255. [[CrossRef](#)]
41. Wang, F.; Zhang, H.; Zhou, A. A particle swarm optimization algorithm for mixed-variable optimization problems. *Swarm Evol. Comput.* **2021**, *60*, 100808. [[CrossRef](#)]
42. Kyle, S. Operational features of the Leica laser tracker. In Proceedings of the IEE Seminar Business Improvement through Measurement, Birmingham, UK, 8 June 1999.
43. Zhu, G.; Cai, M.; Wang, Z.; Li, L.; Zhao, J.; Zhou, B. A new calibration method for a dynamic coordinate system in a robotic blade grinding and polishing system based on the six-point limit principle. *Robot. Comput.-Integr. Manuf.* **2023**, *83*, 102561. [[CrossRef](#)]
44. Dehghani, M.; McKenzie, R.A.; Irani, R.A.; Ahmadi, M. Robot-mounted sensing and local calibration for high-accuracy manufacturing. *Robot. Comput.-Integr. Manuf.* **2023**, *79*, 102429. [[CrossRef](#)]
45. Botero-Valencia, J.; Marquez-Viloria, D.; Castano-Londono, L.; Morantes-Guzmán, L. A low-cost platform based on a robotic arm for parameters estimation of Inertial Measurement Units. *Measurement* **2017**, *110*, 257–262. [[CrossRef](#)]
46. Mourtzis, D. Simulation in the design and operation of manufacturing systems: State of the art and new trends. *Int. J. Prod. Res.* **2020**, *58*, 1927–1949. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.