applied sciences

MDPI

Article

A Workspace-Analysis-Based Genetic Algorithm for Solving Inverse Kinematics of a Multi-Fingered Anthropomorphic Hand

Chun-Tse Lee and Jen-Yuan (James) Chang *

Department of Power Mechanical Engineering, National Tsing Hua University, Hsinchu 30013, Taiwan; ctlee@gapp.nthu.edu.tw
* Correspondence: jychang@pme.nthu.edu.tw; Tel.: +886-3-574-2498

Abstract: Although the solution of inverse kinematics for a serial redundant manipulator has been widely researched, many algorithms still seem limited in dealing with complex geometries, including multi-finger anthropomorphic hands. In this paper, the inverse kinematic problems of multiple fingers are an aggregate problem when the target points of fingers are given. The fingers are concatenated to the same wrist and the objective is to find a solution for the wrist and two fingers simultaneously. To achieve this goal, a modified immigration genetic algorithm based on workspace analysis is developed and validated. To reduce unnecessary computation of the immigration genetic algorithm, which arises from an inappropriate inverse kinematic request, a database of the two fingers' workspace is generated using the Monte Carlo method to examine the feasibility of inverse kinematic request. Furthermore, the estimation algorithm provides an optimal set of wrist angles for the immigration genetic algorithm to complete the remaining computation. The results reveal that the algorithm can be terminated immediately even when the inverse kinematic request is out of the workspace. In addition, a distribution of population in each generation illustrates that the optimized wrist angles provide a better initial condition, which significantly improves the convergence of the immigration genetic algorithm.

Keywords: inverse kinematics; genetic algorithm; workspace analysis; multi-fingered anthropomorphic hand

check for updates

Citation: Lee, C.-T.; Chang, J.-Y. A Workspace-Analysis-Based Genetic Algorithm for Solving Inverse Kinematics of a Multi-Fingered Anthropomorphic Hand. *Appl. Sci.* **2021**, *11*, 2668. https://doi.org/10.3390/app11062668

Academic Editor: Alessandro Gasparetto

Received: 24 February 2021
Accepted: 15 March 2021
Published: 17 March 2021

1. Introduction

The solution of inverse kinematics (IK) is one of the most critical and elemental issues in robotics. Many related research areas such as motion planning, robotic grasping, manipulation, and manufacturing are all involved in IK. The IK of a manipulator is used to find the map between the joint coordinate (θ) and the Cartesian coordinates (x, y, z), where θ represents the joint angles of each joint in the manipulator and (x, y, z) represent the position of the manipulator's end-effector. While most IK solutions are performed on simple manipulators, several studies focus on redundant manipulators. For a redundant robot, in addition to the basic accurate positioning, different requirements such as computation time, robustness, and minimum displacement are considered suboptimal conditions. One of these complex redundant manipulators is the multi-fingered anthropomorphic robotic hand, whose joints of fingers move simultaneously and two wrist joints are deemed as the base of the hand. To obtain the IK solution of the anthropomorphic hand, a robust algorithm is essential to solve this multiple end-effector problem.

Traditionally, there are three kinds of methods used to acquire the IK solutions of a redundant manipulator, namely, the algebraic [1,2], geometric [3], and iterative methods [4–7]. Each of them has its own advantages and disadvantages. For example, the algebraic method can derive a closed-form solution from a Denavit and Hatenberg (D–H) table [8] and, thus, offers efficiency in computation. However, the algebraic method cannot guarantee a closed-form solution for all configurations of manipulators. On the other hand, the geometric method can obtain a closed-form solution, while it is usually limited in specific configurations such as sphere–revolute–sphere manipulators. Likewise, the iterative algorithm utilizes local linearization to get a linear trajectory, and it is suitable for most manipulators. Nevertheless, the iterative method may encounter a singularity problem in which it can accidentally converge to the same solution if the same initial point is given. All of these traditional methods have one point in common, i.e., high computational efficiency, but they all inevitably suffer from certain restrictions. Alternatively, thanks to the advancement of computer hardware, compute-intensive algorithms can come in handy. For these reasons, although the computation is intensive, researchers have focused on using biologically inspired optimization to solve the IK problem.

Recently, many papers associated with neural networks [9–13] and evolutionary algorithms [14–18] have been published to allow obtaining the IK solution of redundant manipulators with some convincing results. KöKer [19] presented a hybrid approach using neural networks and the genetic algorithm (GA) to solve the IK problem under the consideration of error minimization in the end-effector. Three Elman neural networks were applied to create the floating-point initial population for the GA. The simulation experiments were performed on a six-axis serial robot and showed a critical high precision. Tabandeh et al. [20] presented an adaptive niching strategy to acquire several possible solutions to IK problem. By using the niching genetic algorithm accompanied by the quasi-Newton algorithm, the precision and resolution of the simulation results were improved dramatically. A sequential mutation genetic algorithm combined with an extreme-learning machine was presented by Zhou et al. [21]. In their study, the extreme-learning machine first computed the preliminary IK solution. With the use of the simple GA, the refined solution was optimized. The grasping experiments were then conducted on the 6 degree of freedoms (DOFs) Stanford MT-ARM robotic manipulator, from which an improvement in the computational time without reducing the accuracy of IK solutions was demonstrated. Momani et al. [17] provided a continuous genetic algorithm to solve the IK problems of a 3R planar manipulator. The continuous genetic algorithm operators used in the initialization phase, crossover, and mutation were designed to smoothen the joint space while the Cartesian path was kept accurate.

Although the aforementioned algorithms made a great contribution to solving the IK of redundant robots, they focused only on single kinematic chain redundant manipulators which are widely used in industry. For instance, the most common applications using GA to solve the IK problem are 6-DOFs manipulators, although there are closed-form solutions already; this can also be found in 7-DOFs manipulators, for which GA is quite applicable because it includes one redundant DOF in the joint space when the end-effector moves in the Cartesian coordinate system. Given that previous research did not consider the inverse kinematic problem of multiple manipulators such as robotic hands and dual-arm robots, further investigation on the IK problem of multiple kinematic chain manipulators is crucial. The other drawback of GA is its ignorance of an unreachable target. Specifically, GA is implemented regardless of feasibility, which is a fatal flow in solving the IK problem of a dual-arm robot. Without checking the viability in advance, it is frequently found that, when two target points are given to each arm, only one arm can reach the target point while the other one fails because of the limited range of motion. Therefore, feasibility analysis beforehand is indispensable.

In this paper, a workspace-analysis-based immigration genetic algorithm (IGA) is proposed to solve the inverse kinematics of a multi-fingered anthropomorphic hand. The architecture of the proposed algorithm is divided into two stages, namely, the offline stage and the online stage. In the offline stage, workspaces of the exemplary index finger and the thumb are established by the Monte Carlo method and stored in a database for the feasibility check. In the online stage, a feasibility estimation function examines whether the IK request is out of the workspace or not. If the request is unreachable, the algorithm is suspended immediately. Conversely, with a reachable request, the feasibility function provides a better initial condition for the IGA to get high-precision IK solutions. This paper aims to solve the IK problem of multiple chain manipulators and to provide a mechanism that can avoid unnecessary computation if the target point is inappropriate. Meanwhile, the proposed mechanism can offer a superior range to variables, which improves the convergence of the IGA. The performance of this algorithm is validated through simulations, which include both reachable and unreachable IK requests. A population distribution during the evolutionary process is conducive to the understanding of the improvement in convergence.

The remainder of this paper is organized as follows: in Section 2, the configuration of the anthropomorphic hand is described, and the IK problem is defined. In Section 3, the establishment of the workspace and mechanism to check the feasibility is explained. Section 4 introduces the architecture of the proposed algorithm and illustrates details of each operator in IGA, while Section 5 presents the simulation results of the proposed algorithm applied to several IK requests. Lastly, the contributions of this paper are summarized in the conclusion.

2. Problem Formulation

In this section, the geometric description of the anthropomorphic hand and the scenario of the motion task are discussed. The anthropomorphic hand is designed according to the human hand anatomy. In this anthropomorphic hand, the wrist mechanism is designed to possess two degrees of freedom in order to mimic the motion of a human's wrist. For fingers, the thumb and index finger each has three joints, namely, the metacarpophalangeal (MCP), the proximal interphalangeal (PIP), and the distal interphalangeal (DIP) joints, as shown in Figure 1. Among these three, the PIP and DIP joints of a human finger are anatomically coupled by a tendon. As a result, to mimic a human finger's anatomy, the design of a four-bar linkage mechanism to create coupled motion by a ratio of 1:1 is adopted in this research, whereby θ_{t3} is equal to θ_{t4} and θ_{i3} is equal to θ_{i4}, where subscript t stands for thumb and i stands for index finger. As many dexterous gestures and tasks are executed by the thumb and index finger, such as picking up small items and handling a key, the kinematics of the thumb and index finger of the anthropomorphic hand is the main focus of this paper. The other fingers, i.e., the middle finger, ring finger, and little finger, could also be taken into consideration for the IK problem by using the same procedure presented in this study. However, to fix ideas and to convey the concept clearly, the remainder of this paper uses only the thumb and index finger as examples for simplification.

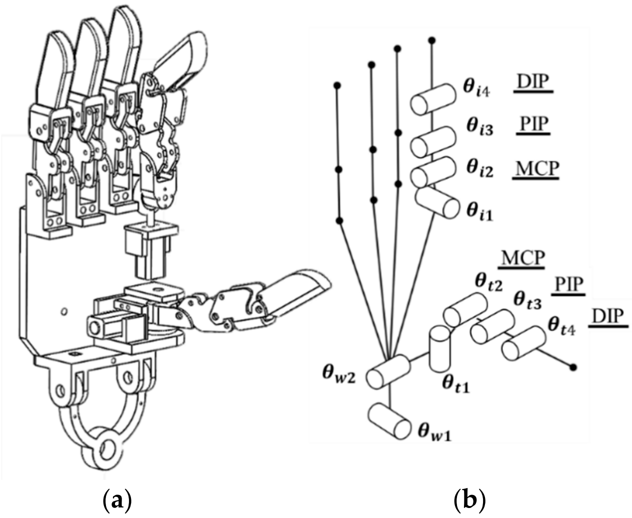


Figure 1. Schematics of (**a**) the design and (**b**) the kinematic structure of the model anthropomorphic hand.

In this study, the target points of the index finger and the thumb are given at the same time as the inverse kinematic requests and the associated task space is restricted only to a translation of posture. Because both the index finger and the thumb are connected to the wrist, it is evident that the robotic hand has multiple serial kinematic chains, and each of them is dependent. In other words, the tips of both the index finger and the thumb comprise the same angle θ_{w1} and θ_{w2}, where the subscript w represents the wrist. Mechanically, the inverse kinematics of the two fingers are highly coupled.

With the problem described according to Cartesian coordinates, an IGA is then used to solve this highly nonlinear and coupled inverse kinematics. However, from theory, an IGA always returns a minimum value that it has computed even if the target point is out of the workspace. To avoid this ambiguous situation, a procedure to generate the workspace is proposed, and a technique to check the feasibility to assist the computation of the inverse kinematics is discussed in the next section.

3. Workspace Analysis

3.1. Forward Kinematics

Through kinematic analysis, the relationship between the position of the end-effector and the angles of each joint can be easily defined for a robot. The kinematics can be conducted using two analyses, namely, forward kinematics analysis and inverse kinematics analysis. The forward kinematics analysis allows one to obtain orientation and translation of the end-effector on the basis of the given angles of joints. Therefore, the workspace of the fingers is easily acquired using the forward kinematics analysis.

Before analyzing kinematics, the problem must be formulated using a mathematic tool. Denavit and Hatenberg in 1995 introduced the convention for two spatial coordinates [8]. From this convention, a matrix, which contains the information on the translation and rotation, can be obtained. To access the values of the D–H parameters, the coordinate of each joint must be defined first. The long axis of each cylinder, as shown in Figure 1, represents the rotational direction of each joint for the index finger and the thumb. Tables 1 and 2 give the geometric parameters of the index finger and the thumb, respectively. The forward kinematic model of the index finger or thumb is given through D–H parameter convention, and the transformation between adjacent i-th and $(i-1)$-th joint coordinate systems can be written in the form of the following homogeneous transformation matrix:

$$ {}^{i-1}T_i(\theta_i) = \begin{bmatrix} cos\theta_i & -sin\theta_i cos\alpha_i & sin\theta_i sin\alpha_i & a_i cos\theta_i \\ sin\theta_i & cos\theta_i cos\alpha_i & -cos\theta_i sin\alpha_i & a_i sin\theta_i \\ 0 & sin\alpha_i & cos\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{1} $$

where α_i, a_i, θ_i, and θ_i are D–H parameters. From Tables 1 and 2, one can observe that each finger has six joints from the base to its fingertip. Starting from joint 0 to joint 5, the homogeneous transformation matrix ${}^0T_5$ can be written as

$$ {}^0T_5 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5 = \begin{bmatrix} n_x & o_x & a_x & P_x \\ n_y & o_y & a_y & P_y \\ n_z & o_z & a_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{2} $$

where $(P_x \; P_y \; P_z)^T$ is the position vector of the fingertip measured from the base coordinate system. By giving many sets of joint angles, the workspace of the finger can be collected through the forward kinematics analysis.

3.2. Database Collection

The workspace of the robot mathematically represents all possible solutions of locations that the robot's end-effector can reach in space. To ensure that the finger of the model anthropomorphic hand can reach the target points, careful analysis of its workspace is crucial.

Table 1. Denavit and Hatenberg (D–H) table of the index finger.

Joint i	α_i (°)	a_i (mm)	θ_i	d_i (mm)	Range
0	90°	33	θ_{w1}	0	−60° to + 60°
1	−90°	93	θ_{w2}	38	−60° to + 90°
2	90°	15.5	θ_{i1}	−4	−45° to 0°
3	0	20	θ_{i2}	0	0° to + 90°
4	0	25	θ_{i3}	0	0° to + 90°
5	0	36	θ_{i4}	0	0° to + 90°

Table 2. D–H table of the thumb.

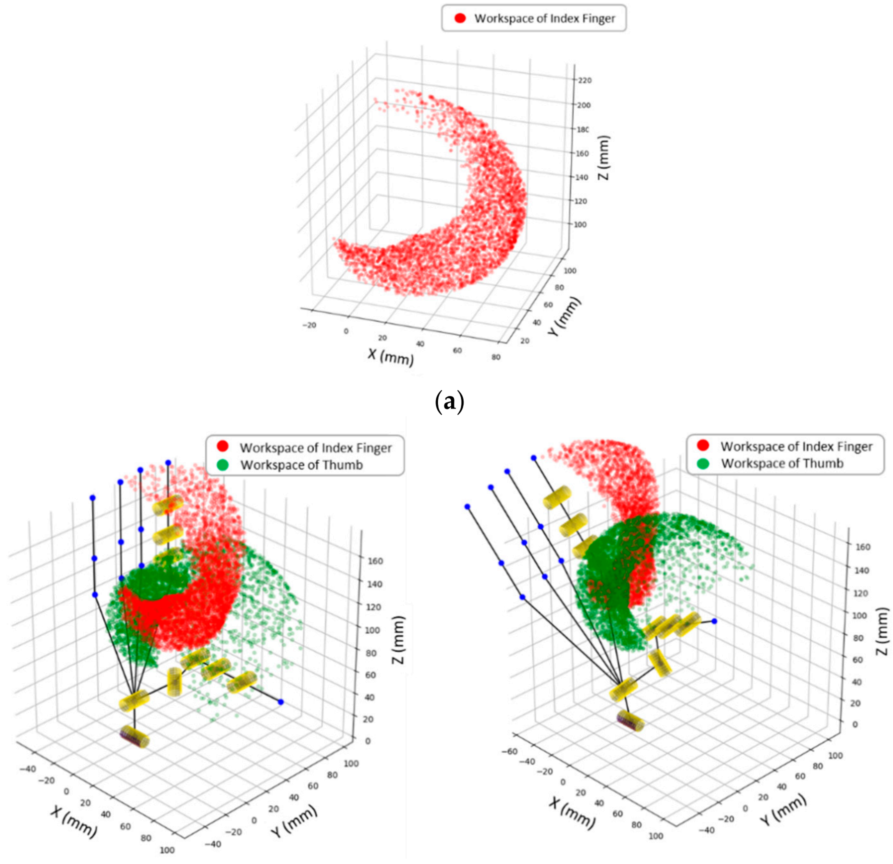
Joint i	α_i (°)	a_i (mm)	θ_i	d_i (mm)	Range
0	90°	33	θ_{w1}	0	−60° to + 60°
1	90°	1.8	θ_{w2}	38	−60° to + 90°
2	−90°	21	θ_{t1}	31	−30° to + 90°
3	0	20	θ_{t2}	0	−90° to 0°
4	0	25	θ_{t3}	0	−90° to 0°
5	0	36	θ_{t4}	0	−90° to 0°

The conventional workspace analysis is to find the range of the locations according to all joints. For example, the dual-arm robot investigated by Kang et al. [22] has two workspaces to indicate the motion range of each arm. In this dual-arm robot, the two arms are connected to the same fixed frame, and the joints of the left arm are independent of the right arm's joints. Consequently, we can simply use the workspace to check whether one arm can reach the target point.

Unlike the dual-arm robot, the index finger and the thumb of the anthropomorphic hand are connected to a moving palm. As they share the same degrees of freedom of the wrist, the workspaces of two fingers are definitely dependent and coupled. Even if the two target points of the index finger and the thumb are all in their workspaces, with the traditional method, the calculated angle of the wrist may show different value at each point, which implies that there may exist mechanical interference and that the index finger and thumb cannot reach their target points at the same time. To resolve this problem, the workspace analysis below is proposed to make the workspace of the thumb independent of the workspace of the index finger in this research.

Unlike the traditional method to build a workspace, we propose the workspace analysis under the condition that the angle of the wrist is separated from the angle of the fingers. The angle of the wrist is denoted as $\vec{q_w} = [\theta_{w1},\ \theta_{w2}]^T$. The angles of the two fingers are denoted as $\vec{q_{f1}} = [\theta_{t1},\ \theta_{t2},\ \theta_{t3},\ \theta_{t4}]^T$ and $\vec{q_{f2}} = [\theta_{i1},\ \theta_{i2},\ \theta_{i3}, \theta_{i4}]^T$, with the subscript f standing for finger. Once the angle of the wrist is decided, the workspaces of the index finger and the thumb are determined by substituting sets of $\vec{q_{f1}}$ and $\vec{q_{f2}}$ into Equation (2) to generate the cloud of points, as shown in Figure 2a, in which the red dots represent the possible positions of the fingertip of the index finger. The workspace of the index finger for a given set of wrist angles is denoted as $I\left(\vec{q_w}\right)$ and that of the thumb is denoted as $T\left(\vec{q_w}\right)$.

The Monte Carlo method, a statistical method to solve problems by random sampling, was adopted to analyze the workspace. A considerable number of particles were used to approach the real shape of the workspace. Algorithm 1 shows the details for generating the workspace of the index finger and the thumb for a given set of wrist angle. Figure 2b,c illustrate the point cloud of the index finger (red dots) and the thumb (green dots) under different conditions of wrist angle.



(**a**)

(**b**) (**c**)

Figure 2. Representation of three-dimensional (3D) finger workspace: (**a**) point cloud of the index finger's tip generated by the Monte Carlo Method; (**b**) workspace of the index finger's tip and the thumb's tip at wrist angle $(\theta_{w1}, \theta_{w2}) = (0^{\circ}, 0^{\circ})$; (**c**) workspace of the index finger's tip and the thumb's tip at the wrist angle $(\theta_{w1}, \theta_{w2}) = (12^{\circ}, 30^{\circ})$.

Since the purpose of workspace analysis was to roughly examine whether the target point is near the finger's workspace or not, the number of sampling points of each finger in a certain configuration of wrist was intentionally set to be low. As a result, the sub-workspaces $I\left(\vec{q_w}\right)$ and $T\left(\vec{q_w}\right)$ were established every 10 degrees in wrist angle to cover the whole workspace. Using this algorithm, the database for two fingers was established and the data were sorted by the angle of the wrist. With this database, a further estimation function was created to check the feasibility of the IK request.

Algorithm 1. Algorithm for constructing the workspace of fingers

```
Purpose: To obtain the workspace of the fingers at certain wrist angle
Input: $\overrightarrow{q}_w, \overrightarrow{q}_{f,range}$
Output: $S$
    $S = \varnothing, I\left(\overrightarrow{q_w}\right) = \varnothing, T\left(\overrightarrow{q_w}\right) = \varnothing$
    Choose sample size $N_{sample}$
    for $i \in \left\{1, 2, \ldots, N_{sample}\right\}$ do
        Obtain $\overrightarrow{q_{f1}}$ from randomly sampling $\overrightarrow{q}_{f,range}$
        Obtain $\overrightarrow{q_{f2}}$ from randomly sampling $\overrightarrow{q}_{f,range}$
        $P_T = {}^0T_5\left(\overrightarrow{q_w}, \overrightarrow{q_{f1}}\right)$
        $P_I = {}^0T_5\left(\overrightarrow{q_w}, \overrightarrow{q_{f2}}\right)$
        $add\ P_T\ to\ T\left(\overrightarrow{q_w}\right)$
        $add\ P_I\ to\ I\left(\overrightarrow{q_w}\right)$
    end for
    $add\ T\left(\overrightarrow{q_w}\right),\ I\left(\overrightarrow{q_w}\right)\ to\ S$
    return $S$
```

3.3. Feasibility Check

Now, we have two databases for the index finger and the thumb, and each of them has many point clouds to indicate the workspace of the finger's tip according to the angle of the wrist. The feasibility of the motion command can be determined by computing the nearest distance between the target point and the point in the workspace. This distance is defined as

$$\kappa(S, p) = \underset{x_i \in S}{min} \|p - x_i\|, \tag{3}$$

where S is the point cloud of the finger's tip which is stored in the database, and p is the target point. This function can be achieved by many algorithms such as the k-nearest neighbor algorithm which returns the nearest point and the distance. Because the point cloud is not concentrated enough, a threshold value is needed as the basis for proximity to the workspace. Using Equation (3) and the database of reachable area, the coupling phenomenon can be solved by the following equation:

$$Dist = min\left[\kappa\left(T\left(\overrightarrow{q_w}\right),\ p_1\right) + \kappa\left(I\left(\overrightarrow{q_w}\right),\ p_2\right)\right], \tag{4}$$

where $\kappa(a, b)$ denotes the operation with the k-nearest neighbor algorithm for entities a and b, and p_1 and p_2 are the target points for the thumb and the index finger, respectively. Please note that the angle set of the wrist $\overrightarrow{q_w}$ is intentionally set to be the same in the database of the thumb and the index finger so that the inconsistency between the wrist angles of two fingers can be avoided. The result of Equation (4) is used as a basis for performing the IGA to prevent unnecessary computations. Moreover, the database is sorted in the order of the angle of the wrist for quick search and access to the corresponding optimized angles of the wrist. This optimized angle set of the wrist $\overrightarrow{q_w}^*$ is further provided to the IGA through the estimation function below as the initial guess to accelerate the computation and to offer better convergence. Equation (5) defines the estimation function.

$$\theta_{w1}^*, \theta_{w2}^* = \underset{\overrightarrow{q_w}}{\arg}\ min\left[\kappa\left(T\left(\overrightarrow{q_w}\right), p_1\right) + \kappa\left(I\left(\overrightarrow{q_w}\right), p_2\right)\right]. \tag{5}$$

4. Algorithm

The proposed hybrid genetic algorithm mainly includes two steps:

1. Examining the feasibility of the motion command on the basis of the workspace and providing the angle set of the wrist if it is reachable.
2. Using IGA to solve the inverse kinematics and finding joint values of two fingers.

A schematic illustration of the proposed algorithm is depicted in Figure 3, and a flowchart of the proposed hybrid GA is shown in Figure 4. A detailed explanation of each step is addressed in the sections below.

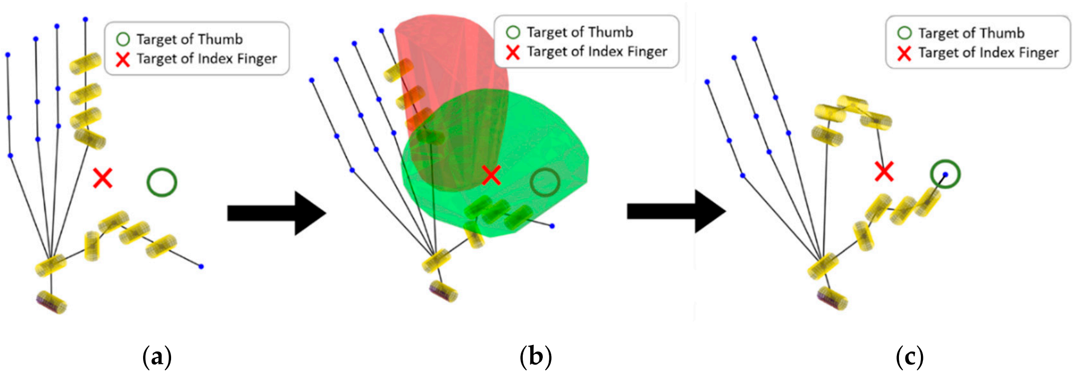


Figure 3. Process of the proposed algorithm: (**a**) assign inverse kinematics (IK) request; (**b**) feasibility check and generate wrist angle range; (**c**) perform immigration genetic algorithm (IGA) to obtain precise solution.

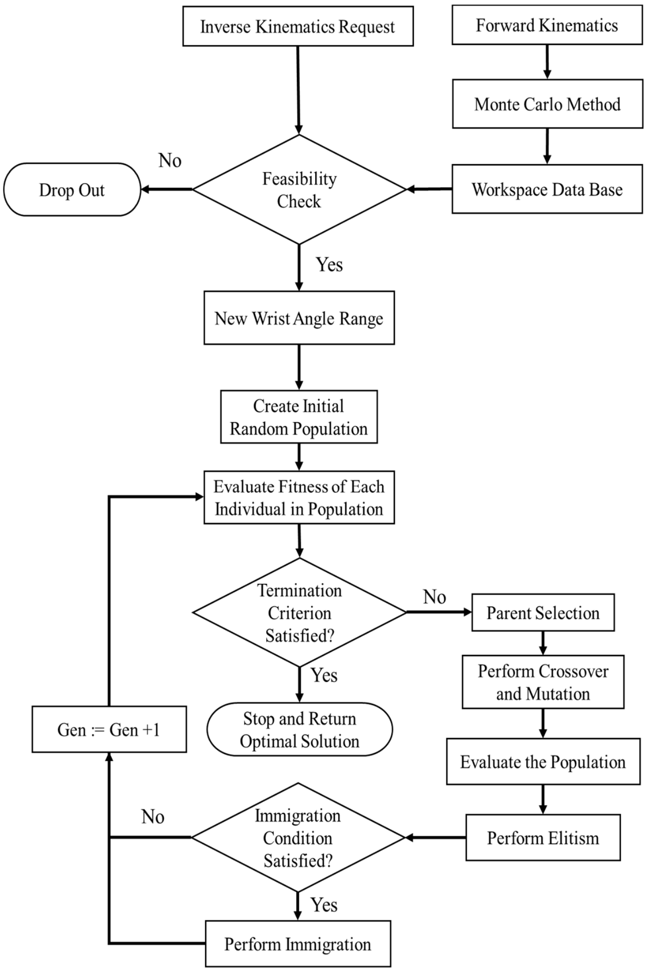

Inverse Kinematics Request
Forward Kinematics
Monte Carlo Method
No
Drop Out
Feasibility Check
Workspace Data Base
Yes
New Wrist Angle Range
Create Initial Random Population
Evaluate Fitness of Each Individual in Population
Termination Criterion Satisfied?
No
Parent Selection
Perform Crossover and Mutation
Yes
Gen := Gen +1
Stop and Return Optimal Solution
Evaluate the Population
No
Immigration Condition Satisfied?
Perform Elitism
Yes
Perform Immigration

Figure 4. Flowchart of the proposed algorithm.

4.1. Feasibility Check and Feedforward of Wrist Angle

Utilizing the techniques elaborated in Section 3, in which collection and establishment of the database are conducted, the feasibility check is performed when the inverse kinematics request is assigned. The value obtained from Equation (4) is compared to the heuristic threshold value to decide whether the motion command should be dropped out or not. The heuristic threshold value and the fitness value of termination are the sum of the distances between target points and end-effectors of the two fingers. Basically, the heuristic threshold value is set to be a bit bigger than the fitness value of termination. Provided that the heuristic threshold value is as low as termination value, the feasibility check will frequently fail. On condition that the heuristic threshold value is much larger than the fitness value of termination, the fitness value will not converge to termination 1. In this research, the fitness value of termination is set to be 1 and the heuristic threshold value is set to be 5.

On the assumption that the motion command passes the feasibility check, Equation (5) yields an optimal wrist angle set $\overset{\rightharpoonup}{q_w}^{*}$ according to the aforementioned workspace database. The new wrist angle range provided to the IGA is, therefore, generated in the form of $\overset{\rightharpoonup}{q_w}^{*} \pm \theta_w$. The value θ_w is associated with the intervals of the wrist angle used in creating the workspace database. A wider interval denotes a larger value of θ_w. In this paper, θ_w was set to be 10° to cover the adjacent workspace. For example, the original range of θ_{w1} was $-60°$ to +60° and that of θ_{w2} was $-60°$ to +90°. Supposing that the $\overset{\rightharpoonup}{q_w}^{*}$ provided by the feasibility estimation function is $[0°,\ 12°]$, then the new range of θ_{w1} becomes $-10°$ to +10° and that of θ_{w2} becomes +2° to 22°. Note that, if the new angle range exceeds the original angle limit, the boundary shall be set to the original one.

4.2. Chromosome Coding

In an IGA, a chromosome is the individual in a population, and it consists of many genes to show the individual differences. In this case, the chromosome represents a solution to the IK problem, while a series of genes constitutes a chromosome. The population contains many chromosomes and acts as a set of numerous possible IK solutions. To begin with the IGA, it is necessary to encode the joint angle configurations as the genes for each chromosome. The genetic encoding format is a binary string representation for each joint. By concatenating the binary string of each joint, the chromosome is built and provided for further operations. Figure 5 shows the binary encoding structure of the chromosome.

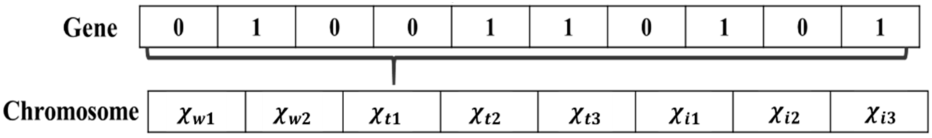

Gene
0 1 0 0 1 1 0 1 0 1
Chromosome
χ_{w1} χ_{w2} χ_{t1} χ_{t2} χ_{t3} χ_{i1} χ_{i2} χ_{i3}

Figure 5. Binary encoding structure.

The decoding process can help a joint angle recover from a chromosome. Using binary–decimal conversion, the bit value is used to calculate the exact joint angle and to ensure that the joint variable is within the bounded joint space, which is mathematically written as

$$\theta_i = \theta_{i_{min}} + \frac{\left(\sum_{j=1}^{n} \chi_i[j] * 2^j\right)}{2^n} * \left(\theta_{i_{max}} - \theta_{i_{min}}\right), \tag{6}$$

where $\theta_{i_{min}}$ is the lowest value for a certain joint, and $\theta_{i_{max}}$ is the highest one. It is hard for iterative algorithms based on the Jacobian matrix to avoid exceeding the joint limits. Through this encoding mechanism, the resulting angles computed by the IGA always fall in the range of the joint limit and, thus, prevent unpredictable errors.

4.3. Evaluation and Selection

Better chromosomes are chosen from the population in order to reproduce the offspring during the process of the IGA. To evaluate a chromosome, a fitness function, which is a quantitative measure, is adopted to measure how satisfactory a chromosome is. In this study, the task-space is only within the position. As a result, the fitness function is defined as

$$Error_p = \|p_1 - x_t\| + \|p_2 - x_i\|, \tag{7}$$

$$Error_d = \max_{k=1,\ldots,8} \theta_{k,i} - \theta_{k,f}, \tag{8}$$

$$Fitness = Error_p + K \cdot Error_d, \tag{9}$$

where p_1 is the target point for the thumb, x_t is the tip of the thumb, p_2 is the target point for the index finger, x_t is the tip of the index finger, $\theta_{k,i}$ is the initial angle of each joint before computing the IGA, $\theta_{k,f}$ is the angle of the joint contained in the chromosome, and K is the weight of the second criterion (infinity-norm term).

The position of x_t is calculated using the forward kinematics according to the angle set contained in the chromosome. Equation (7) computes the distance in Cartesian space to indicate how far the tips of fingers are from the target points and, thus, sums the distances to be a criterion for the fitness function. Equation (8) takes the maximum absolute value within the angle differences between the initial angle and the angle generated by the IGA. Equation (8) is derived from the concept of the infinity norm and is used to guarantee the smoothness of the motion. The fitness function is minimized by the IGA, which means that the tips of two fingers are set to reach the target points as closely as possible, with each joint having about the same angular difference.

On the other hand, to maintain the quality and diversity of the population, the selection operator is performed to select superior chromosomes for breeding of the next population. Supposing that the chosen chromosomes are always the best individuals, the diversity of the IGA will be narrow. Hence, the k-tournament selection operator is used to choose which chromosomes should reproduce new offspring. The k-tournament selection randomly withdraws k chromosomes from the population and picks the top two chromosomes for the crossover operator. The advantage of the k-tournament selection is that it is unnecessary to sort the entire population to get the top two chromosomes, thereby reducing computational resources. In addition, the top two chromosomes may be poor ones, which keeps the population diverse and eliminates a local minimum.

4.4. Crossover and Mutation

In this step, the new chromosomes are produced by genetic operations, such as a crossover and a mutation. The crossover operation is adopted to create two offspring from two parent chromosomes. The parents swap with the partial section of genes and then generate two children. Here, k-point crossover is used. The k-point crossover first selects k crossover points to make the k + 1 interval in one chromosome and then changes the section of genes every two intervals to form two offspring.

The crossover operation is followed by the mutation operation. Mutation is utilized to change some genes of the children chromosomes. The mutation rate is defined as the reciprocal of the chromosome's length. The mutation operation is implemented on each gene and flips it with the probability of the mutation rate. Because of these two genetic operations, the diversity of the IGA is highly promoted; simultaneously, it becomes more possible to get close to the global minimum.

4.5. Elitism

Elitism is a mechanism that guarantees the best chromosome will survive in every generation. Elitism can accelerate the convergence of the IGA but slightly decrease the diversity of the population. In this study, the feasibility check function was used to provide an optimal wrist angle set that narrows down the search space of the IGA. Therefore, the possibility of being trapped in local minimum was lower, thus validating elitism as a good strategy to be adopted.

4.6. Immigration

An immigration strategy can be performed periodically to maintain the diversity level of the genetic algorithm [23]. The best individuals immigrate to the new population to replace the worst ones at defined intervals, which is called "structured memory immigration operator." In this research, the immigration operator was set to occur every three generations in order to obtain better optimal solutions. Half of the population was replaced with the best individuals of the previous generation in the same quantity. As mentioned, we can ensure that the solution to the IK problem is near the best individuals of the population through the feasibility check. In this case, the introduced immigration operator is more suitable for the population than a random immigration operator. As a result, the tournament selection can have a better chance of selecting better chromosomes and maintaining the diversity.

5. Results and Discussion

In this section, the performance of the proposed hybrid GA is validated by examination through parametric numerical simulations. The simple GA and the IGA are also performed to compare the results. To highlight the capability of determining the feasibility, both appropriate and improper IK requests are given to different algorithms for computing IK solutions. The detailed parameters used in the illustrated cases through numerical simulations with the proposed algorithm are shown in Table 3.

Table 3. Parameters used in the proposed hybrid genetic algorithm. GA, genetic algorithm.

Parameter	Value
GA type	Generational
Representation	Binary
Chromosome length	80
Population number	500
Selection	10-tournament
Crossover	2-point
Crossover rate	0.8
Mutation	Bit-flip
Mutation rate	1/L
Survivor	1000
Termination	3 s or fitness lower than 0.6

Among the parameters in Table 3, the fitness value of termination dominates the computation time. As exhibited in Table 4, when the termination value increases, it takes less time to compute, while the accuracy of the solution is sacrificed. It also compares the average execution time of the simple genetic algorithm and the performance of the proposed hybrid GA, which shows that the proposed hybrid GA is two times faster than the simple GA. Additionally, it is known that the selection of the fitness value of termination depends on the application scenario. Thus, the termination condition applied in different tasks was varied accordingly.

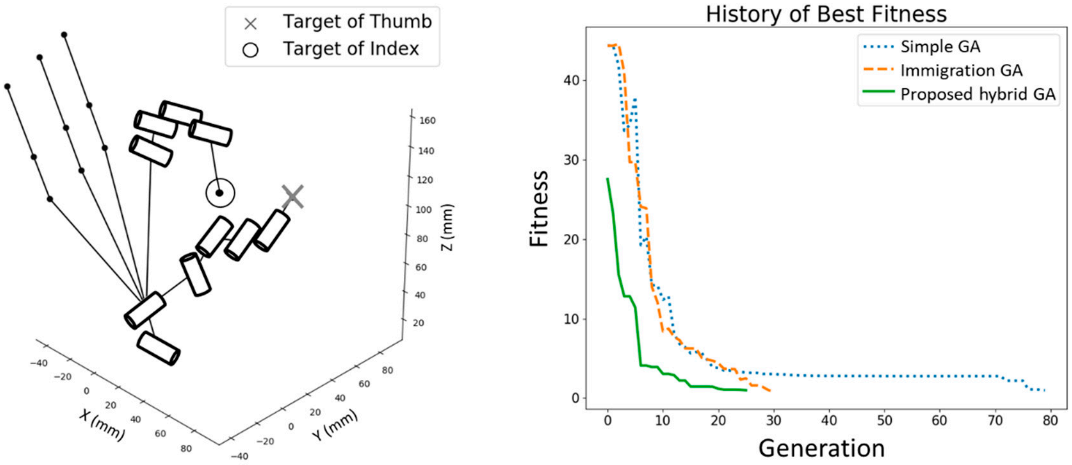
Table 4. Effect of termination value on the convergence speed of GA.

	Termination Fitness Value	Average Number of Generation	Average Execution Time (s)
Simple GA Proposed hybrid GA	1	52 24	4.41 2.06
Simple GA Proposed hybrid GA	5	18 10	1.68 0.72
Simple GA Proposed hybrid GA	10	7 4	0.63 0.35

In calculating the inverse kinematics, two points were assigned as the target points of the index finger and thumb. Different algorithms were performed to solve the inverse kinematics, and the best fitness values were recorded for the graphical representation to analyze the performance of each algorithm. Table 5 lists the conditions and results for three illustrated cases. The illustrated case #1 and case #2 were those with reachable target points, whereas an unreachable target point was assigned in the illustrated case #3. In the illustrated case #1, the target fingertip point sets were (50, 0, 130) and (75, 30, 125) for the index finger and the thumb, respectively. As shown in Figure 6a, the two target points were in the workspace of two fingers and they passed the feasibility check. Additionally, the corresponding angle set of the wrist $(\theta_{w1},\ \theta_{w2}) = (12^\circ,\ 15^\circ)$ was provided to the hybrid GA as the initial condition. By using the solutions computed by the GA, the tips of the index finger and the thumb reached their target points. The simple GA and IGA were also performed to solve the same IK problem as a comparison. As shown in Figure 6b, the proposed hybrid GA was observed to have a lower fitness value than the other GAs in the beginning. The proposed hybrid GA was found to obtain the solution for the inverse kinematic problem faster than the other GAs with fewer generations.

Table 5. Results of illustrated cases performed by the proposed algorithm.

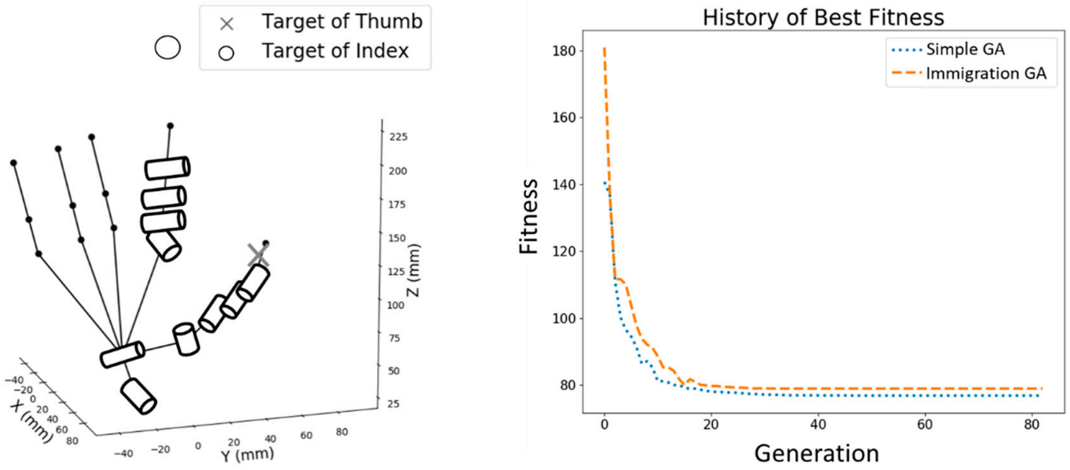
Case	Target Points (mm)	Feasibility	Feedforward Wrist Angle	Solutions of the Proposed Algorithm		Error Sum of Two Fingers (mm)
#1	Index finger: (50, 0, 130) Thumb: (75, 30, 125)	Pass	$\theta_{w1} : 12^\circ$ $\theta_{w2} : -15^\circ$	θ_{w1} θ_{w2} θ_{t1} θ_{t2} θ_{t3} θ_{i1} θ_{i2} θ_{i3}	14.4981° −5.0000° 14.9817° −8.4396° −21.9780° −45.0001° 55.2968° 45.0110°	0.5795
#2	Index finger: (45, 52, 172) Thumb: (81, 60, 111)	Pass	$\theta_{w1} : 0^\circ$ $\theta_{w2} : 0^\circ$	θ_{w1} θ_{w2} θ_{t1} θ_{t2} θ_{t3} θ_{i1} θ_{i2} θ_{i3}	0.2515° −4.9987° 15.0110° −19.6923° −8.8571° −19.6374° 14.5494° 41.4726°	0.5875
#3	Index finger: (50, 0, 300) Thumb: (60, 50, 125)	Drop out	Not Computed	Not Computed		Not Computed



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

Figure 6. (**a**) Schematic illustration showing the solution resulting from an appropriate motion command and (**b**) the corresponding fitness performance of the illustrated case #1.

In the illustrated case #3, the target fingertip point sets were (50, 0, 300) and (60, 50, 125) for the index finger and thumb, respectively. As shown in Figure 7a, the target point of the index finger was obviously out of the workspace. In this circumstance, as shown in Figure 7b, the proposed hybrid GA immediately ceased the inappropriate motion command to prevent unnecessary computation. However, the other GAs still conducted computation until meeting certain termination criteria. As a result, Figure 7b shows high fitness values of the simple GA and IGA, while the proposed hybrid GA successfully stopped computing. The hybrid GA, thus, was successful in avoiding the unnecessary computation for a given inappropriate IK request.



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

Figure 7. (**a**) Schematic illustration showing the solution resulting from an inappropriate motion command and (**b**) the corresponding fitness performance of the illustrated case #3.

Results in Table 6 demonstrate the distribution of population by exemplary generations during the evolution process in the illustrated case #1 for the simple GA and the proposed hybrid GA. The blue dots represent all the thumbs' tips of the population in one generation, whereas the red dots serve as those of the index finger. The corresponding cross marks stand for the target point or the answer of the IK problem. In the initial generation, the population distribution of the simple GA was scattered more widely than that of the proposed hybrid GA. Because of the feedforward wrist angle, the proposed hybrid GA started with a smaller searching scope. When the generation increased, the proposed hybrid GA had more chances to get close to the target points and, therefore, converged faster than the simple GA.

To surpass the performance of the other GA, the workspace of two fingers plays an important role in the proposed hybrid GA. Initially, the workspace analysis can provide the nearest distance between the target point and the workspace. As a result, the feasibility of the IK request can be checked accordingly. Second, the workspace was pre-established and stored in the order of the wrist angle. Consequently, the workspace analysis gives a feedforward wrist angle to the hybrid GA. The hybrid GA then reduces the variable range of the wrist angle to offer a superior initial condition. In general, the proposed hybrid GA has better convergence than the other GA and can effectively terminate the computation when an improper IK request is given.

Table 6. Comparison of population distribution at different generations in the illustrated case #1.

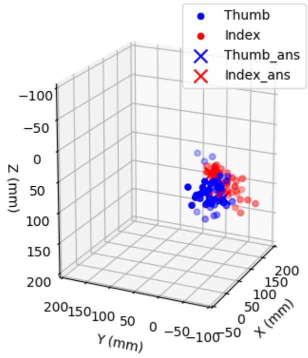
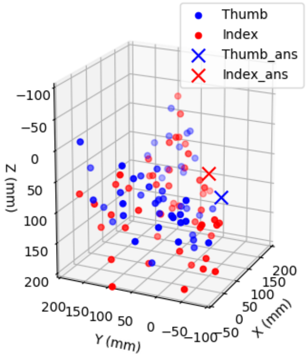
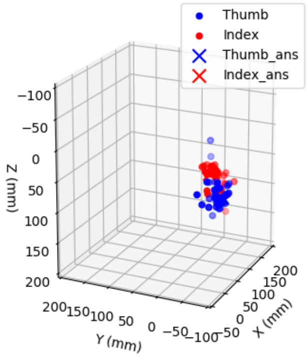
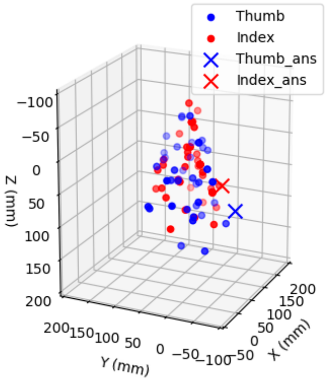
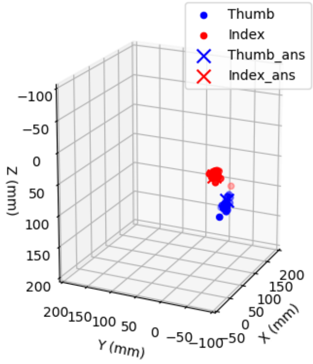
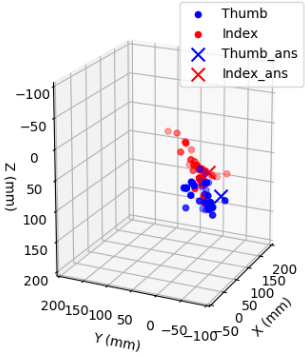
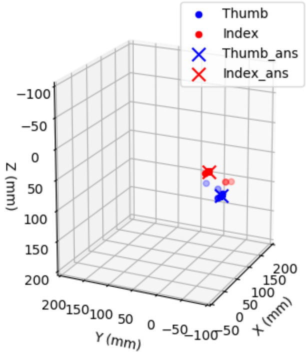
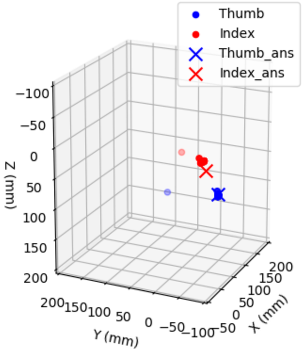
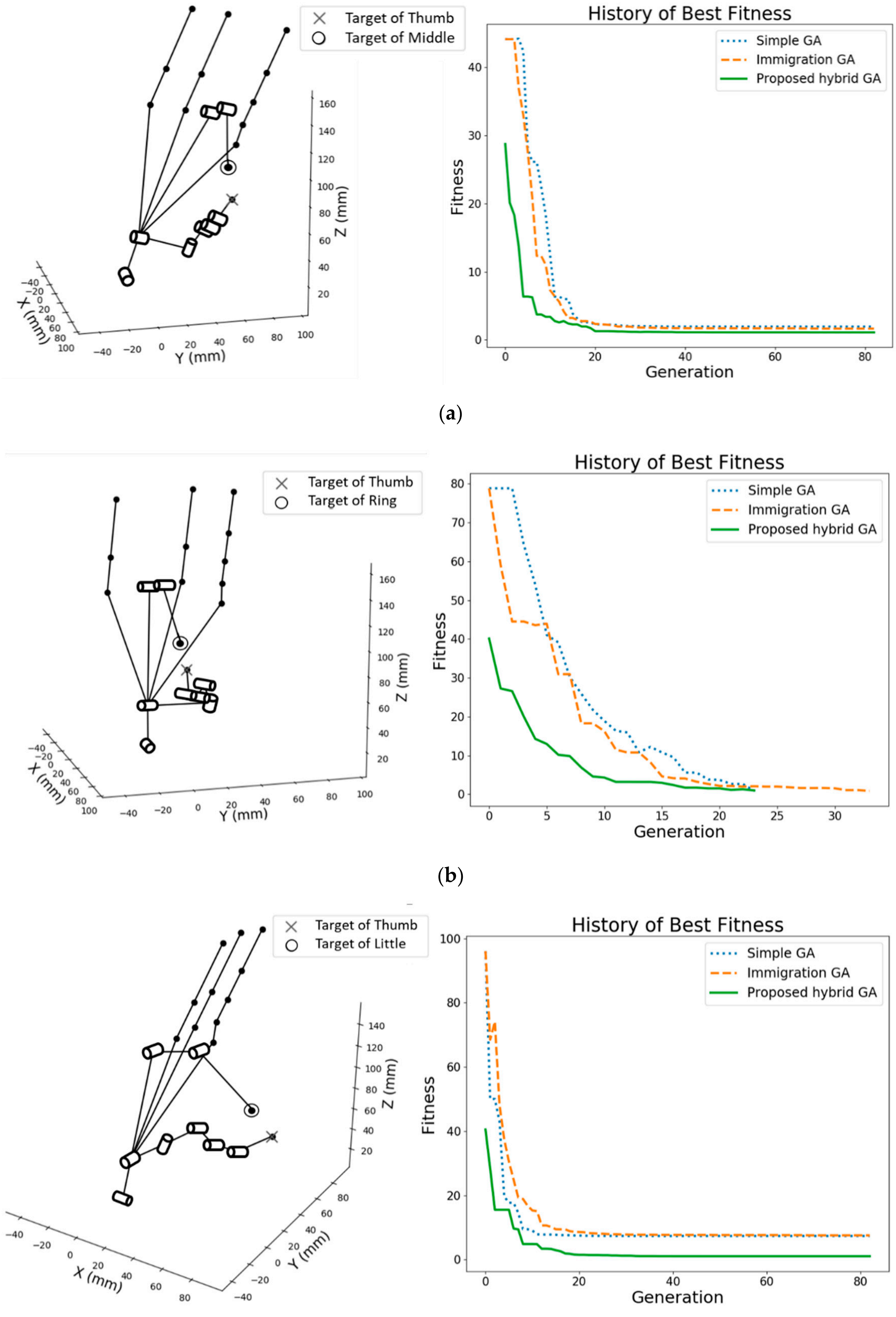
	Proposed Hybrid GA	**Simple GA**
Generation No. 1		
Generation No. 2		
Generation No. 5		

Table 6. *Cont.*

	Proposed Hybrid GA	**Simple GA**
Generation No. 20		

In addition to the interaction between a thumb and an index finger, the proposed algorithm can solve the IK problem for a thumb and the other three fingers. Numerical simulations were performed to solve the IK under all three scenarios, namely, the IK of the thumb and middle finger, the IK of the thumb and ring finger, and the IK of the thumb and little finger. Similarly, the same procedures were adopted in these IK problems, including building the database of the workspace, checking the feasibility, and providing the feedforward wrist angle. Figure 8 shows a schematic illustration of the IK solutions for the thumb and the other three fingers. The diagram of the best fitness value over generations indicates that the proposed algorithm outperformed the simple GA and the immigration GA. Through these cases, it was confirmed that the proposed algorithm is capable of solving the IK problem of a multi-fingered robotic hand.



(**a**)

(**b**)

(**c**)

Figure 8. Schematic illustration showing the solution of the IK problem with (**a**) thumb and middle finger, (**b**) thumb and ring finger, and (**c**) thumb and little finger, as well as the corresponding fitness performance.

6. Conclusions

In this study, a workspace-analysis-based immigration genetic algorithm was presented to solve the inverse kinematics of a multi-fingered hand. In the offline stage, the kinematic model of fingers is built to perform forward kinematics, from which the workspace database of the fingers can be established using the Monte Carlo method. In the online stage, an estimation function is used to check the feasibility of the target point whenever an IK request is assigned. If the target point is within the workspace, the estimation function then provides an optimal set of wrist angles for the IGA to compute the solution; contrarily, the estimation function terminates further calculation. This estimation function enables the searching range to be narrowed to the neighborhood of the solution. Therefore, the distribution of the population is less widespread, and the IGA can have better initial condition and greater convergence. Numerical simulations were conducted to verify the performance of the proposed hybrid GA. For the unreachable targets, the estimation function could efficiently abort the entire computation. For the reachable targets, the proposed algorithm took fewer iterations to obtain accurate results than the simple GA and the IGA. As a whole, this study presented an algorithm to solve the multi-fingered inverse kinematics, which enables the tips of the index finger and the thumb to move to their target points concurrently; likewise, the same approach can be applied to the other fingers. The results can also be applied to the IK problem of a dual-arm robot that has degree of freedoms in the waist. The proposed method is believed to have the potential to be employed in many industrial robots.

Author Contributions: Conceptualization, methodology, software, formal analysis, visualization, and writing—original draft preparation, C.-T.L.; conceptualization, writing—review and editing, supervision, project administration, and funding acquisition, J.-Y.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Ministry of Science and Technology of Taiwan, grant number MOST 109-2218-E-007-024.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Jiang, L.; Sun, B.; Gu, H. A Power Series Based Inverse-Kinematics Solution of Humanoid Robot Finger with Coupled Joints. *Int. J. Hum. Robot.* **2018**, *15*, 1850016. [CrossRef]
2. Manocha, D.; Canny, J.F. Efficient inverse kinematics for general 6R manipulators. *IEEE Trans. Robot. Autom.* **1994**, *10*, 648–657. [CrossRef]
3. Lee, C.G. Robot arm kinematics, dynamics, and control. *Computer* **1982**, *15*, 62–80. [CrossRef]
4. Kelemen, M.; Virgala, I.; Lipták, T.; Miková, L'.; Filakovský, F.; Bulej, V. A novel approach for a inverse kinematics solution of a redundant manipulator. *Appl. Sci.* **2018**, *8*, 2229. [CrossRef]
5. Bensalah, C.; Gonzalez-Quijano, J.; Hendrich, N.; Abderrahim, M. Anthropomorphic robotics hand inverse kinematics using estimated svd in an extended sdls approach. In Proceedings of the 2013 16th International Conference on Advanced Robotics (ICAR), Montevideo, Uruguay, 25–29 November 2013; pp. 1–7.
6. Tchon, K. Optimal Extended Jacobian Inverse Kinematics Algorithms for Robotic Manipulators. *IEEE Trans. Robot.* **2008**, *24*, 1440–1445. [CrossRef]
7. Buss, S.R. Introduction to inverse kinematics with jacobian transpose, pseudoinverse and damped least squares methods. *IEEE J. Robot. Autom.* **2004**, *17*, 16.
8. Hartenberg, R.S.; Denavit, J. A Kinematic Notation for Lower Pair Mechanisms Based on Matrices. *J. Appl. Mech.* **1955**, *77*, 215–221.
9. Ren, H.; Ben-Tzvi, P. Learning inverse kinematics and dynamics of a robotic manipulator using generative adversarial networks. *Robot. Auton. Syst.* **2020**, *124*, 103386. [CrossRef]
10. Kim, B.-H. An adaptive neural network learning-based solution for the inverse kinematics of humanoid fingers. *Int. J. Adv. Robot. Syst.* **2014**, *11*, 3. [CrossRef]

11. Al-Gallaf, E. Multi-fingered robot hand optimal task force distribution: Neural inverse kinematics approach. *Robot. Auton. Syst.* **2006**, *54*, 34–51. [CrossRef]
12. Mayorga, R.V.; Sanongboon, P. Inverse kinematics and geometrically bounded singularities prevention of redundant manipulators: An Artificial Neural Network approach. *Robot. Auton. Syst.* **2005**, *53*, 164–176. [CrossRef]
13. El-Sherbiny, A.; Elhosseini, M.A.; Haikal, A.Y. A comparative study of soft computing methods to solve inverse kinematics problem. *Ain Shams Eng. J.* **2018**, *9*, 2535–2548. [CrossRef]
14. 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]
15. Starke, S.; Hendrich, N.; Magg, S.; Zhang, J. An efficient hybridization of genetic algorithms and particle swarm optimization for inverse kinematics. In Proceedings of the 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), Qingdao, China, 3–7 December 2016; pp. 1782–1789.
16. Durmuş, B.; Temurtaş, H.; Gün, A. An inverse kinematics solution using particle swarm optimization. In Proceedings of the International Advanced Technologies Symposium, Elazığ, Turkey, 16–18 May 2011; pp. 193–197.
17. Momani, S.; Abo-Hammour, Z.S.; Alsmadi, O.M. Solution of inverse kinematics problem using genetic algorithms. *Appl. Math. Inf. Sci.* **2016**, *10*, 225. [CrossRef]
18. Ahmad, M.; Kumar, N.; Kumari, R. A Hybrid Genetic Algorithm Approach to Solve Inverse Kinematics of A Mechanical Manipulator. *Int. J. Sci. Technol. Res.* **2019**, *8*, 1777–1782.
19. KöKer, R. A genetic algorithm approach to a neural-network-based inverse kinematics solution of robotic manipulators based on error minimization. *Inf. Sci.* **2013**, *222*, 528–543. [CrossRef]
20. Tabandeh, S.; Melek, W.W.; Clark, C.M. An adaptive niching genetic algorithm approach for generating multiple solutions of serial manipulator inverse kinematics with applications to modular robots. *Robotica* **2010**, *28*, 493–507. [CrossRef]
21. Zhou, Z.; Guo, H.; Wang, Y.; Zhu, Z.; Wu, J.; Liu, X. Inverse kinematics solution for robotic manipulator based on extreme learning machine and sequential mutation genetic algorithm. *Int. J. Adv. Robot. Syst.* **2018**, 15. [CrossRef]
22. Shengzheng, K.; Wu, H.; Yao, L.; Li, D. Coordinated workspace analysis and trajectory planning of redundant dual-arm robot. In Proceedings of the 2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI), Xi'an, China, 19–22 August 2016; pp. 178–183.
23. Tajani, C.; Abdoun, O.; Lahjouji, A.I. Genetic algorithm adopting immigration operator to solve the asymmetric traveling salesman problem. *Int. J. Pure Appl. Math.* **2017**, *115*, 801–812. [CrossRef]