

Article

Integration of DE Algorithm with PDC-APF for Enhancement of Contour Path Planning of a Universal Robot

Issraa Jwad Kazim ^{1,2,*} , Yuegang Tan ¹ and Layth Qaseer ³ 

¹ Department of Mechanical Engineering, School of Mechanical and Electronic Engineering, Wuhan University of Technology, Wuhan 430070, China; ygtan@whut.edu.cn

² Computer Technology Engineering, AL-Mustafa University College, Baghdad 10067, Iraq

³ City College of Technology, The City University of New York, Brooklyn, NY 11201, USA; Lqaseer@citytech.cuny.edu

* Correspondence: israa.cet@almustafauniversity.edu.iq

Abstract: In the robotic engineering field, the main target, especially in industry, manufacturing, and surgical operations, is reaching the optimal performance of manipulators. The purpose of this paper is to quantify the contour tracking performance of collaborative universal manipulator robot (UR5) by setting the gain of position domain controller. In order to improve and enhance the track of manipulator in experimental applications we utilize differential evolution (DE) optimization, using MATLAB toolbox with an applied robot operating system (ROS). The adopted current approach does not only optimize the gain of position domain controller but also prevent collisions by detecting a “border crossing” without turning off the manipulator and allowing the automation agent to be on the scene, coexisting in harmonic mode and avoiding collisions. This requires the implementation of an algorithm that detects an obstacle to avoid anticipated collisions. For this purpose, the adopted algorithm uses the DE algorithm to modify the artificial potential field (APF). The results of this paper present that on one hand, meta-heuristic optimization algorithm features give the best performance indices for linear and non-linear contours, and on the other hand, DE algorithm features give good modification to APF to generate collision free contour path planning.

Keywords: optimization; position domain control; contour tracking; artificial potential field; differential evolution; universal manipulator robot



Citation: Kazim, I.J.; Tan, Y.; Qaseer, L. Integration of DE Algorithm with PDC-APF for Enhancement of Contour Path Planning of a Universal Robot. *Appl. Sci.* **2021**, *11*, 6532. <https://doi.org/10.3390/app11146532>

Academic Editor: Alessandro Gasparetto

Received: 4 June 2021
Accepted: 1 July 2021
Published: 16 July 2021

Publisher's Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



Copyright: © 2021 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

A manipulator robot is the most useful and popular robot type, so has been acquiring importance in recent years [1]. The manipulator robot has been increasingly used more and more, especially in industrial and manufacturing applications like packaging, welding, etc. It also is associated with all life fields, especially in personal assistance where it works instead of a human hand. The manipulator must have good accuracy and high precision [2]. There are numerous methods to solve the trajectory generation problem, and many key methods were studied such as cell decomposition [3], mathematical program [4], skeleton [5] and boundary following methods. In order to design the performance of planning, more than one planning approach, like the APF decomposition method on aircraft [6] needs to be utilized. Since the introduction of the proportional-integral-derivative (PID) controllers, many algorithms were used such as trial-and-error, Ziegler-Nichols, pole placement and D-partitioning methods. All of these algorithms have been adopted for tuned PID controllers on the elastic joints [7]. In addition, all of these algorithms have been used in frequency response of linear time invariant systems. These algorithms require a good knowledge of the system, and therefore, many meta-heuristic optimization approaches have been used to gain tuned controllers for linear and nonlinear robotic manipulator systems. There are many optimization algorithms such as differential evolution (DE), genetic algorithms (GA) and particle swarm optimization (PSO) which have been used for

PID tuning of robotic systems. At present, the DE algorithm gives better results for the PID-PDC controller in linear and nonlinear contour tracking [8]. The previous researches concluded that the meta-heuristic algorithms produce better results than traditional tuning techniques (in time domains), to tune the position domain PID controller. The principal work of the position domain controller (PDC) was presented for PD and PID controllers [9]. The principal idea of the position domain controller is to describe the system in slave motions and master motions. Master motions were considered as reference to the slave motion instead of time. The optimization method is defined for the configuration space and is based on an obstacle-dependent Gaussian model predictive control to avoid obstacles [10], and to move to the goal point by APF which is velocity dependent [11]. The APF needs to be developed by many algorithms such as proximity maneuvering, formation-flying, large angle slew maneuvers, and small spacecraft formation [12]. To build an efficient path for the manipulator robot, it is important to define the frame transformation and solve the inverse kinematics. In inverse kinematics, all joint variables of the manipulator are considered as a function of goal configuration of the end-effector. A basic principle of APF is best described when adding an attractive potential field with a repulsive potential field as a local minimum formation to trap a maneuvering object in many configurations [13]. The basic idea of the APF algorithm for manipulator path planning was to consider the manipulator movement while considering obstacles as bodies [14]. The famous applications of the collision-free trajectory lie in unmanned aerial vehicles (UAVs) which used a dynamic artificial potential field (D-APF) [15], industrial manipulator robots, and robotics in medical applications.

There are a lot of research studies regarding the manipulator to perform the trajectory with obstacle avoidance. To avoid obstacles, external sensors or technology must be added to the robot [16]. Spatial mapping was proposed for nonlinear and linear distances [17]. In the APF, realize real-time avoidance path planning was enhanced against a randomly moving obstacle. The bidirectional APF method was proposed for obstacle avoidance [18]. In [19], the APF method was enhanced to obtain collision-free paths in a 3D space by using sliding mode techniques. Numerous studies were performed to analyze the trajectories with obstacle avoidance made by the robot. Previous studies showed there are many methods for the controlled robot and deal with it because it has been proven useful in the required field. The objectives of this paper are as follows:

1. To enhance linear and nonlinear contour tracking gain of PID by way of a differential evolution algorithm (DE) to tune the gain of the PID controller and show that DE gives better results than traditional methods especially in non-linear contour tracking systems.
2. Three fitness functions are used for contour tracking to find the best performance function in linear and non-linear tracking, these are integrated absolute error (IAE), integrated square error (ISE) and mean and standard deviation of absolute error (MSDAE).

The artificial potential field (APF) algorithm is enhanced by utilizing a differential evolution algorithm (DE) in order to avoid obstacles in trajectory planning until arriving the target position, i.e., the APF was modified to create a collision-free trajectory by DE. Hence the APF algorithm is presented for the generation of the collision-free trajectory of UR5 manipulator. The improvements in APF will be used to optimize the repulsion constants for obstacles and improve the attraction constant of the endpoint which is directly related to the strength of the force resulting from the APF. In this paper, a six-DOF articulated robot is taken as the object to study the optimization of contour tracking and enhanced obstacle avoidance. The research consists of two parts: The first part studies the differential evolution improvement of an artificial potential field method, which lays the foundation for the optimization of contour tracking and enhanced obstacle avoidance. In the second part, methods of optimizing contour tracking and enhancing path planning of obstacle avoidance robot using improved algorithms are studied. The contribution and novelty of this paper is to use DE algorithm to modify the APF approach and tune the PDC-

PID controller at same time by applying it on the UR5 manipulator robot to control the (linear–nonlinear) contour tracking with obstacle avoidance. The results show that these algorithms are effective.

2. Optimization Methodology

2.1. Contour Tracking Control

In this section, we describe the parameters and the concept of controlling the position domain controller in order to control contour tracking with the UR5 manipulator robot. In this paper, we use the position domain controller because gain tuning is very important in order to obtain good performances for a given controller. Contour tracking performance is mainly determined by the selected control gains of a position domain PID controller.

2.1.1. Position Domain Mapping with Relative Derivative

To convert the slave dynamic motions from time domain into the position domain, the relation must be developed in terms of master and slave motions in the position domain. This relation is represented by the relative derivative of the master motion (q_m) and slave motion (q_{si}).

To explain master/slave is a model of communication for hardware devices where one device has a unidirectional control over one or more devices. This is often used in the electronic hardware space where one device acts as the controller, whereas the other devices are the ones being controlled. In short, one is the master, and the others are slaves to be controlled by the master.

$$q'_{si} = \frac{dq_{si}}{dq_m} = \frac{q_{si}}{q_m} \tag{1}$$

where q_i is the relative position velocity, which represents “the ratio of the slave motion velocity to the master motion velocity”, and it also explains the procedure of motion relation between slave motions and master motions. The relative acceleration of the slave motion which is the second relative derivative of Equation (1) is given by:

$$q''_{si} = \frac{dq'_{si}}{dq_m} \tag{2}$$

According to Equations (1) and (2), in order to convert the slave velocity and acceleration from the time to the position domain, the following equations are considered:

$$\begin{cases} q_{si} = q_m q'_{si} \\ q_{si} = q''_{si} (q_m)^2 + q_m q'_{si} \end{cases} \tag{3}$$

Equation (3) describes the relation between the relative and the absolute motions, i.e., to transform the absolute velocities and accelerations in the time domain to relative velocities and accelerations in the position domain. The above equation transforms a dynamic system from time-domain to position-domain [20,21].

2.1.2. PDC-PID Control

A PDC-PID controller is considered as the time domain representation of the PID controller which is given by:

$$\tau_{si}(q_m) = k_{pi}e_{si}(q_m) + k_{Di}e'_{si}(q_m) + k_{Ii} \int_0^{q_m} e_{si}(s)ds \tag{4}$$

where τ_{si} the track of slave motions and k_{Ii}, k_{Di}, k_{pi} the control gains of slave motions.

The slave motion tracking errors are given by:

$$tracking\ error = \begin{cases} e_s(qm) = q_{sd}(qm) - q_s(qm) \\ e'_s(qm) = q'_{sd}(qm) - q'_s(qm) \end{cases} \tag{5}$$

Note that master motions are still being controlled by the traditional PID controller in the time domain, so the DE algorithm is adopted in contour tracking to get an optimum gain control of the PDC-PID controller.

2.1.3. Contour

For the optimization steps to be more efficient, the controller is tuned to two kinds of contours, linear (straight-line) contour and nonlinear (circular) contour for the tracking of manipulator robot. The linear contour is faster than the circular contour with a coverage of bigger space. The characteristics of the two kinds of contour are presented in Table 1.

Table 1. Contour information.

Contour Type	Circular Contour	Linear Contour
Starting Point (m)	(0.5,0, pi/3)	(0.49,0.49, pi/3)
Duration (s)	9	2
Maximum Joint Speed (rad/s)	2.291	1.280
Ending Point (m)	(0.5,0, pi/3)	(0.69,0.68, pi/3)

2.1.4. Fitness Functions

Fitness function is required in the optimization process to transform the contour tracking error into a single arithmetic value for optimization process. In the present work, three types of fitness functions (ISE, IAE, and MSDAE) are adopted and respectively given by:

$$IAE = \int_0^{\infty} |e_c| dt \quad (6)$$

$$ISE = \int_0^{\infty} e_c^2 dt \quad (7)$$

$$MSDAE = |\bar{e}_c| + \sigma|e_c| + \max(|e_c|) \quad (8)$$

where e_c is the contour tracking error, $|\bar{e}_c|$ is the mean value of contour tracking error, $\sigma|e_c|$ is the standard deviation of contour tracking error, and $\max(|e_c|)$ is the maximum value of contour tracking error.

2.2. Artificial Potential Field

The artificial potential field method was firstly defined in [22]. The research was initially applied to the manipulator and mobile robots with static and moving obstacles and target points. In the APF method, a force field is assumed to have a great impact on the planned trajectory of a manipulator. The obstacles were placed on the planned path of the manipulator and experiencing repulsive forces with the manipulator while the target was sending an attractive force to the manipulator. In the APF method, the manipulator's joint moves in a force field. As its name implies, the APF method produces a vector field of forces $F(x)$ derived from the potential (scalar) field $U(x)$ [23]. In the beginning, description of the trajectory path planning by the APF forces will be given, then the APF method will be improved by a differential evaluation algorithm since APF parameters are often adopted empirically, making the process expensive. With the use of algorithms of DE, an improvement in the potential field algorithm is achieved to generate free trajectory collisions. The DE algorithms are conducted for a population of 30 initial individuals (particles) with 50 iterations (generations) with the trajectory of APF using DE showing the smoothest path. It was also possible to check the average error position between the desired end position and the last trajectory position generated by the APF method improved by DE. When comparing computational cost, adopting the DE algorithm gave less execution time regarding parameters calculations of the APF algorithm. Advantages of this hybrid

algorithm (DE-APF) can be summarized as improvement of the APF algorithm, since its parameters are obtained in optimized form, low computational cost and precision of the points of the trajectories reached.

Trajectory Path Planning by APF

The potential function $U(\theta)$ resulting from the artificial field is defined as a linear combination of a repulsion potential field $U_{rep}(\theta)$ which pushes the manipulator away from obstacles, and an attraction potential field $U_{atr}(\theta)$ which pulls the manipulator to the target position [24] as given by the following equation:

$$U(\theta) = U_{rep}(\theta) + U_{atr}(\theta) \quad (9)$$

Thus, considering the presence of a single obstacle, the coordinates of the obstacle position are defined as $\theta_{obs} = (\theta_{obs1}, \theta_{obs2})$ and the target coordinates of target position are defined by $\theta_{target} = (\theta_{target1}, \theta_{target2})$ whereas the position of robot center is located at $\theta = (\theta_1, \theta_2)$. The vector field of forces $F(\theta)$ is derived from a scalar potential field $U(\theta)$ according to the following equation [25].

$$F(\theta) = \nabla|U(\theta)| \quad (10)$$

where $F(\theta)$ is the resultant force which is also a linear combination of two components; $F_{atr}(\theta)$ is the attractive force that guides the robot's center to the target, and $F_{rep}(\theta)$ is a force that causes an artificial repulsion from the obstacle surface produced by $U_{rep}(\theta)$ as given by the following equation

$$F(\theta) = F_{rep}(\theta) + F_{atr}(\theta) \quad (11)$$

Therefore, the APF algorithm deals with a repulsive force from the obstacles and an attractive force to the target. The trajectory generation using potential fields is a method that can be used in global offline planning, where the robot environment is already defined, or in local online planning where the environment is unknown. The presence of obstacles are being detected by sensors mounted on the robot [26]. The repulsion force is first calculated from the starting position of the robot by definition and then recalculated as the robot moves to the final position. It is a way of ensuring that the inertia of movement will be overcome and also that there will be no local minimum in the starting position. The repulsion potential field of the starting position is represented by the following equation [27]:

$$U_{rep}(x, y) = \left| \frac{k_r}{\sqrt{(y - y_0)^2 + (x - x_0)^2}} \right| \quad (12)$$

where k_r is the repulsion intensity parameter in the initial position, which defines the intensity of the repulsion field at the starting position, x and y are reference points for the APF and x_0 and y_0 are the coordinates of the starting position of the robot in the planned trajectory. The attraction force is first calculated from the final position of the robot by definition and then recalculated as the robot moves from the starting to the final position. The potential field of attraction for the final position is represented by the following equation [27]:

$$U_{atr}(x, y) = \frac{1}{2}k_a \left| (y - y_f)^2 + (x - x_f)^2 \right| \quad (13)$$

where k_a is a parameter of the intensity of attraction which defines the intensity of the attraction field at the final position, x and y are the coordinates of reference position for the APF and x_f and y_f are the coordinates of the final position of the generated trajectory. The repulsion force due the obstacle is first calculated from the position of the obstacle by definition and then recalculated as the robot moves from the starting to the final position.

The equation that represents repulsion potential field due to the presence of an obstacle is given by [27]:

$$U_0(x, y) = \left| \frac{k_0}{\sqrt{(y - y_{0i})^2 + (x - x_{0i})^2}} \right| \tag{14}$$

where k_0 is a parameter that controls the intensity of the field potential to repel obstacles, x and y are coordinates of reference position for the APF and x_{0i} and y_{0i} are the coordinates of the position of the obstacle’s center.

2.3. Methodology of Optimization Process in Trajectory Planning and Contour Tracking

The DE optimization method works as robust alternative meta-heuristic algorithm. DE algorithm consists of four steps, the first is the initialization step, the second is the mutation step, the third is the crossover step and the fourth is the selection step. In the first step initialization of population NP with D as a parameter is carried out. Each individual is symbolized as $x_{i,g} = [x_{i,j}^1, x_{i,j}^2, \dots, x_{i,j}^D]$, where $i = [1, 2, \dots, NP]$, $g = [1, 2, \dots, g_{max}]$ and g_{max} is the maximum generation and X_{max} , X_{min} are the upper and lower search bounds imposed on the population. The initialization of the i -th individual is generated according to the following equation:

$$x_{i,0}^j = x_{min}^j + rand(0, 1) * (X_{max} - X_{min}) \tag{15}$$

In the second step, mutation of each individual is generated and hence a new mutated vector is produced. There are many strategies to mutate the individual, in this study “DE/rand/1” is adopted such that:

$$v_{i,g} = x_{r1,g} + F * (x_{r2,g} - x_{r3,g}) \tag{16}$$

where $v_{i,g}$ is the i -th element in the mutated vector. F is a scaling factor which lies between 0 and 1. The indices r_1 , r_2 and r_3 are randomly selected from i . After the mutation step, the crossover step is performed in order to generate the trial vector. In this step, the new mutated vector (which is the target vector) should have the same properties of the trial vector. There are two basic crossover schemes of DE which are the binomial recombination and the exponential recombination [28]. In this study, binomial recombination is used in DE algorithm as performed in (17).

$$u_{ij,k+1} = \begin{cases} w_{ij,k} & \text{if } l > CR \text{ AND } j \neq R \\ v_{ij,k+1} & \text{if } l \leq CR \text{ OR } j = R \end{cases}, \quad j = 1, \dots, D \tag{17}$$

where CR is the crossover probability, j_{rand} is a randomly selected index to confirm that trial vector has at least one property of the mutated vector. Finally, in the selection step, three fitness functions are individually used to determine whether the trial vector is converted into a target vector. The selection operation is characterized in (18):

$$x_{i,g+1} = \begin{bmatrix} u_{i,g} & \text{if } f(u_{i,g}) \leq f(x_{i,g}) \\ x_{i,g} & \text{if } f(x_{i,g}) < f(u_{i,g}) \end{bmatrix} \tag{18}$$

where $f(x)$ is a fitness function.

Trajectory planning can be treated as an optimization problem, that is, calculating the global minimum of U , from the initial configuration $\theta_{initial}$. One of the simplest methods used to solve this problem is through the gradient. In this study, the negative gradient of U can be considered as a force acting on the robot; that is, it is directly related to the movement of the joints [29]. In general, the U field is an additional field in which one part constitutes the attraction, and the other part constitutes the repulsion, as given in (10). The resultant force of artificial potential field that should be applied to the robot is given in (11).

The real action of the potential field for the planar manipulator is obtained by the sum of each force in its respective direction [27].

$$F(x, y) = F_{xya}(x, y) + F_{xyo}(x, y) + F_{xyr}(x, y) \quad (19)$$

Equation (19) is used as an evaluation function to perform the simulation of the DE algorithm. This equation is minimized to find the best strength of the potential field from which the values of the parameters k_r , k_a and k_o have to be determined. For the collision-free trajectory of the UR5 manipulator, cartesian coordinate system is adopted. The generated path must avoid collision with circular obstacles having equal size. The scenario of the simulation is formed by the start point of the robot, the obstacle, and the end point of the robot. An important point regarding the methodology adopted lies in the fact that from the techniques used, the obstacle presented here has the shape of a circle hence reducing the generation of local minimums while creating the path as indicated in [30], although the system is capable of analyzing obstacles of other various shapes like triangular, rectangular and trapezoidal but with the generation of more local minimums in path planning.

2.4. Parameters of Optimization Process

Table 2 introduces the values of parameters which affect the sampling, optimization and frequency response of PDC-PID controller, these are fixed at minimum values to optimize the computation time. In addition, Table 3 introduces parameters of the optimization methodology. It is important to mention that these optimization parameters and the DE optimization parameters given in Tables 2 and 3 are specifically restricted to the position domain controller (PDC-PID) and artificial potential field (APF) also, below the pseudocode of the DE algorithm to be clearly the procedure work of DE to modify APF and PDC-PID shown in Algorithm 1.

The pseudocode of the DE algorithm is given below:

Algorithm 1 DE algorithm for GAIN OF PDC-PID AND APF

```

1: function  $x = \text{DE}(\text{CR}; \text{NP}; \text{F}; \text{CR}; \text{range}; f; \text{gain});$ 
2:    $x \quad \text{rand}(\text{NP}; \text{range});$ 
3:    $\text{fit}_x \quad f(x);$ 
4:   as long as stopping criterion is not satisfied do;
5:     for  $i = 1$  to NP do;
6:        $v_i; \text{G} + 1$  mutation ( $\text{G}; x_i; \text{F}; \text{gain}$ );
7:        $u_i; \text{G} + 1$  crossing ( $\text{G}; \text{G} + 1; x_i; v_i; \text{CR}; \text{gain}$ );
8:       order to
9:        $\text{fit}_u \quad f(u);$ 
10:      for  $i = 1$  to NP do
11:        if  $\text{fit}_u(i) > \text{fit}_x(i)$  then;
12:           $x_i; \text{G} + 1 \quad u_i; \text{G} + 1;$ 
13:        if not
14:           $x_i; \text{G} + 1 \quad x_i; \text{G};$ 
15:        end if
16:      end to
17:    end while

```

Table 2. Optimization parameters.

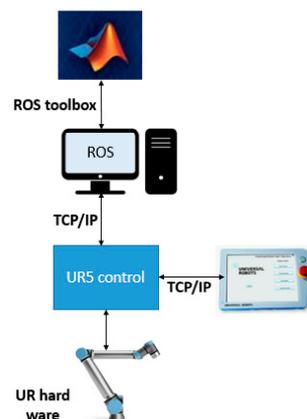
Feasible Bounds of Gain	0–0.0001
Master Motion Sampling Frequency	99 [Hz]
Maximum Iterations Allowed	29

Table 3. DE algorithm parameter values.

Parameters	Value
Population size	30 individuals
Number of generations	50
Probability of crossing (CR)	0.9
magnetic moment	1 erg/G = 1 emu
Step size to be taken in the difference (F)	0.5

3. Results

The experimental results of optimization process were achieved through optimizing linear and non-linear contour tracking using three different fitness functions in DE algorithm for position domain gain optimization and modified APF which was presented in the previous section. To obtain enhanced path and quantified tracking of the manipulator robot, the presentation of the proposed controller has been assessed through a 6-DOF universal manipulator robot UR5 shown in Figure 1. Experimentally, the Robotics System Toolbox of MATLAB simulation with the ROS platform was used to connect with the manipulator robot UR5. The experimental results presented here show the improvement in path planning and contour tracking through the application of DE algorithm into both the APF algorithm and the PDC-PID controller for the UR5 manipulator. In Figure 2. The overall communication model to control UR5.

**Figure 1.** Communicating the UR5 robot with the computer software.**Figure 2.** The overall communication model to control UR5.

In the first part, optimization was achieved for the position domain controller (PID) of manipulator UR5 by meta-heuristic algorithm. The main goal is minimizing the contour error level of the manipulator. Three different types of fitness functions (ISE, IAE, and MSDAE) were adopted to calculate the optimization efficiency in linear and nonlinear contour tracking to complete the optimization process.

The optimization method of the six-DOF UR5 manipulator includes connecting the base joint to the ground for master motion which is controlled using PID controller, while other joints are controlled by the PDC-PID controller for slave motions. Gains of the control process were optimized for linear and non-linear contour tracking, with three types of different fitness functions given in Tables 4 and 5. The DE optimization algorithm was successful to produce gains at the specified bounds.

Table 4. Linear contour gain.

Optimization Algorithm	Fitness Functions	k_p	k_D	k_I
DE	ISE	diag{100,4785,155}	diag{90,10.000,375}	diag{90,10.000,96}
	IAE	diag{9766,5676,89}	diag{8443,10.000,1}	diag{10.000,90,10}
	MSDAE	diag{5436,90,490}	diag{5849,7671,221}	diag{5673,2351,87}

Table 5. Non-linear contour gain.

Optimization Algorithm	Fitness Functions	k_p	k_D	k_I
DE	ISE	diag{9371,6643,9850}	diag{10,000,1405,2720}	diag{4361,183,255}
	IAE	diag{90,10.000,8805}	diag{100,1062,1110}	diag{6330,750,352}
	MSDAE	diag{90,8769,9876}	diag{4078,1516,2600}	diag{1202,127,90}

Figures 3 and 4 show the values of three fitness functions (ISE, IAE and MSDAE) versus the number of iterations for the DE optimization algorithm in linear and non-linear contour tracking respectively. Figure 5 shows the contour error (e_c) produced in the linear and non-linear contour tracking using the three fitness functions described earlier. It was observed that the DE algorithm using ISE fitness function resulted in minimum contour error in linear tracking while the IAE fitness function showed minimum contour error in nonlinear contour tracking, however, the three fitness functions gave good optimization results.

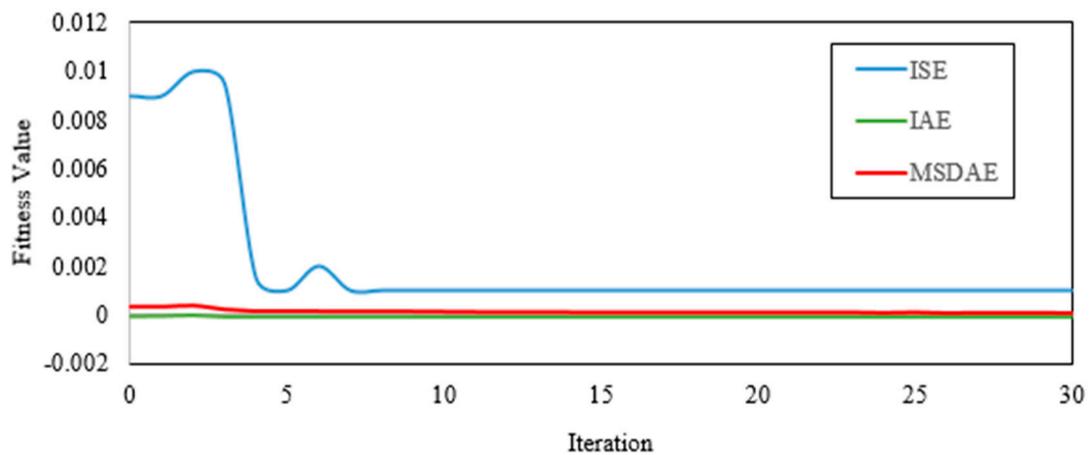


Figure 3. Fitness functions vs. iteration number for linear contour.

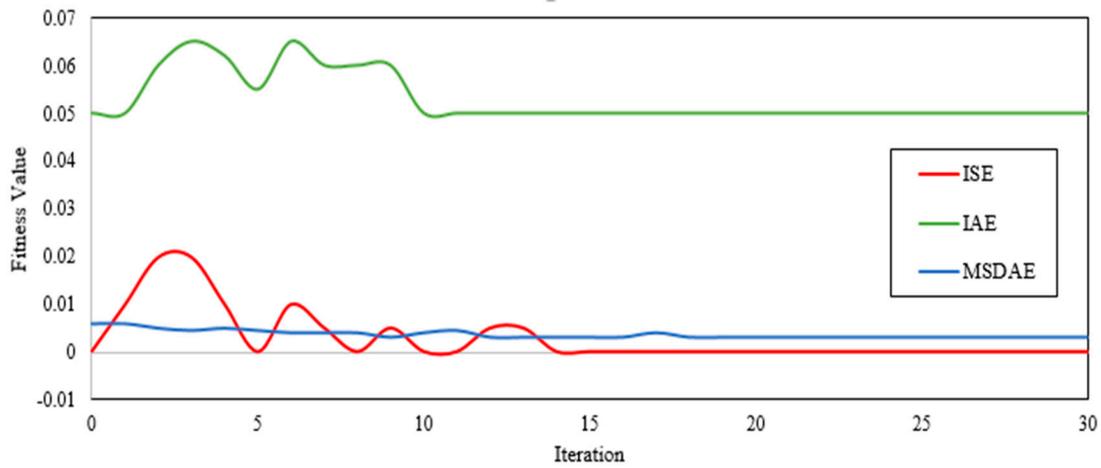


Figure 4. Fitness functions vs. iteration number for non-linear contour.

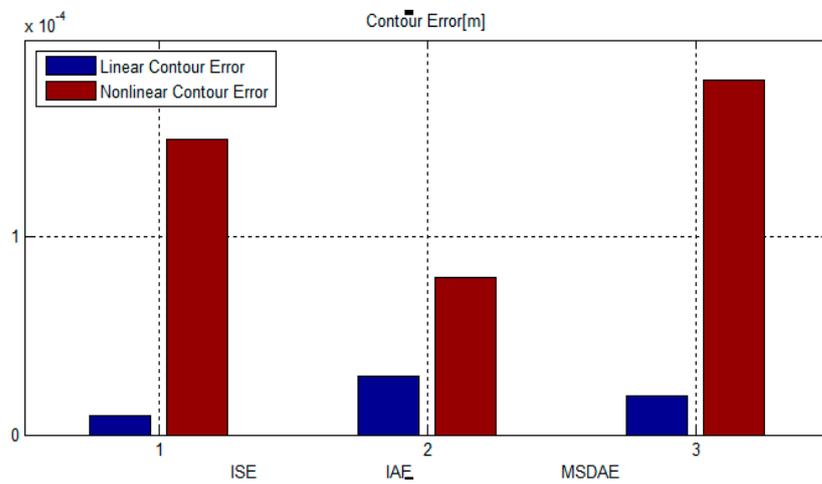


Figure 5. Contour error produced in linear and non-linear contour tracking using gain optimization through fitness functions.

The contour errors of DE optimization for linear and non-linear contours are shown in Figures 6 and 7.

In this section of the results, we find the optimal contour performance for linear and nonlinear contours. This aim was completed by calculating the optimal fitness function for linear and nonlinear contours, and after this, we calculated the error contour when applied to three functions in linear and nonlinear contour functions. Therefore, the result of this section is to obtain optimal contour tracking of the UR5 manipulator robot.

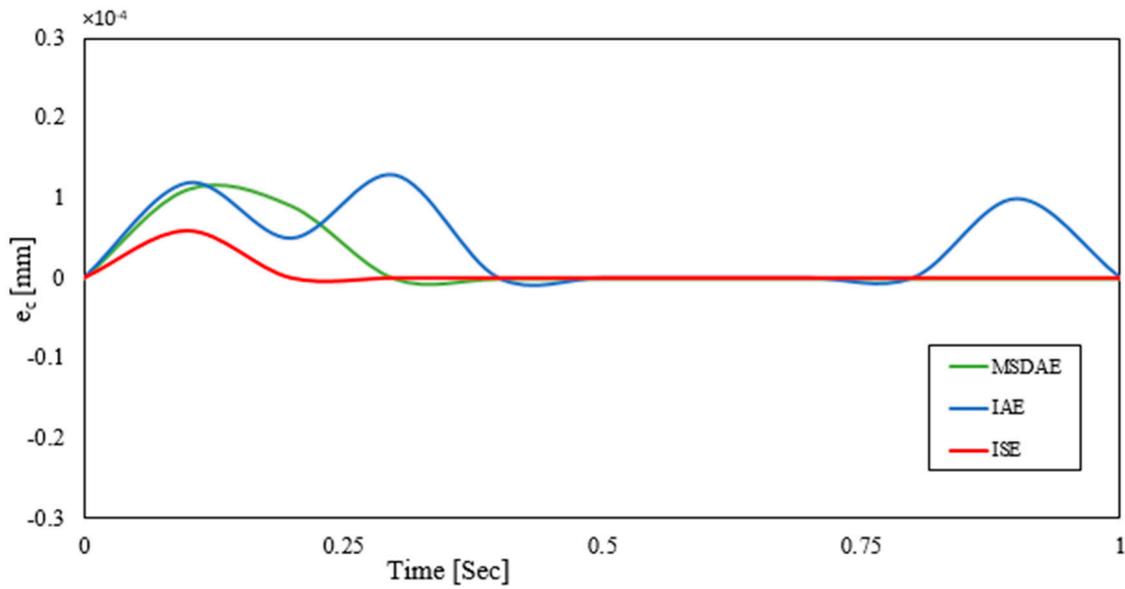


Figure 6. Contour performance of linear tracking.

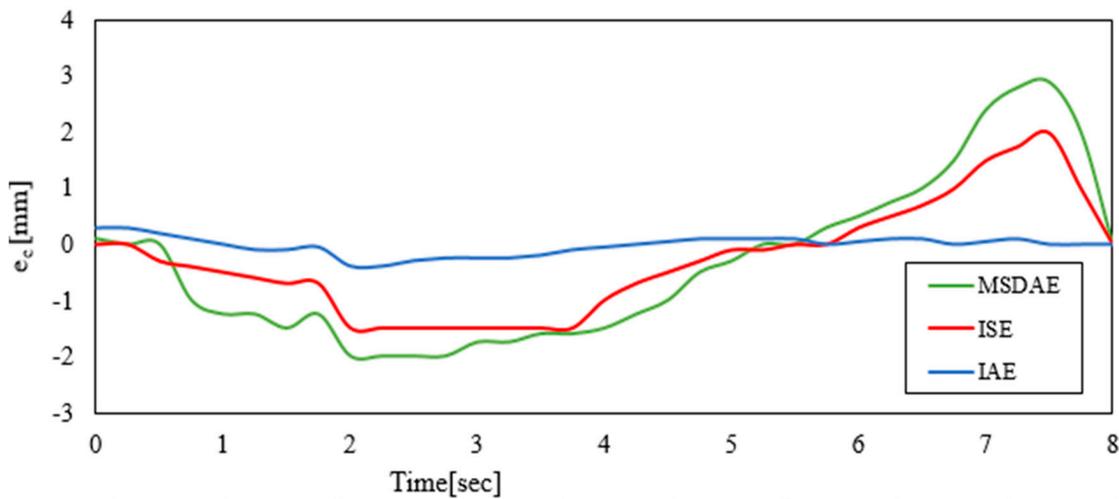


Figure 7. Contour performance of non-linear tracking.

Table 6 shows that the optimization algorithm produced similar mean torques T_m for Joint 1 of the UR5 manipulator using the three fitness functions while other joints produced different values. The DE algorithm in linear tracking produced the smallest contour error with the smallest mean torque required. The same argument applies for non-linear contour tracking, in which, the smallest contour error produced the smallest torque except the mean torque in Joint 2. Figure 8 shows a comparison between linear and non-linear tracking for mean torque required using the three fitness functions.

Table 6. Mean torques and contour errors.

DE Algorithm	Fitness Function	Linear Contour							Non-Linear Contour						
		e_c	T_{m1}	T_{m2}	T_{m3}	T_{m4}	T_{m5}	T_{m6}	e_c	T_{m1}	T_{m2}	T_{m3}	T_{m4}	T_{m5}	T_{m6}
	ISE	3.72×10^{-6}	0.94	0.73	0.63	0.7	0.9	0.56	2.32×10^6	5.56	1.43	0.79	1.3	5.4	0.7
	IAE	8.18×10^{-6}	0.56	0.78	0.78	0.7	0.5	0.7	3.23×10^{-6}	5.53	1.74	1.33	1.6	4.5	0.9
	MSDAE	4.87×10^{-6}	9.46	0.73	0.6	0.67	0.8	0.5	1.29×10^{-6}	5.43	1.30	0.84	1.0	0.37	1.6

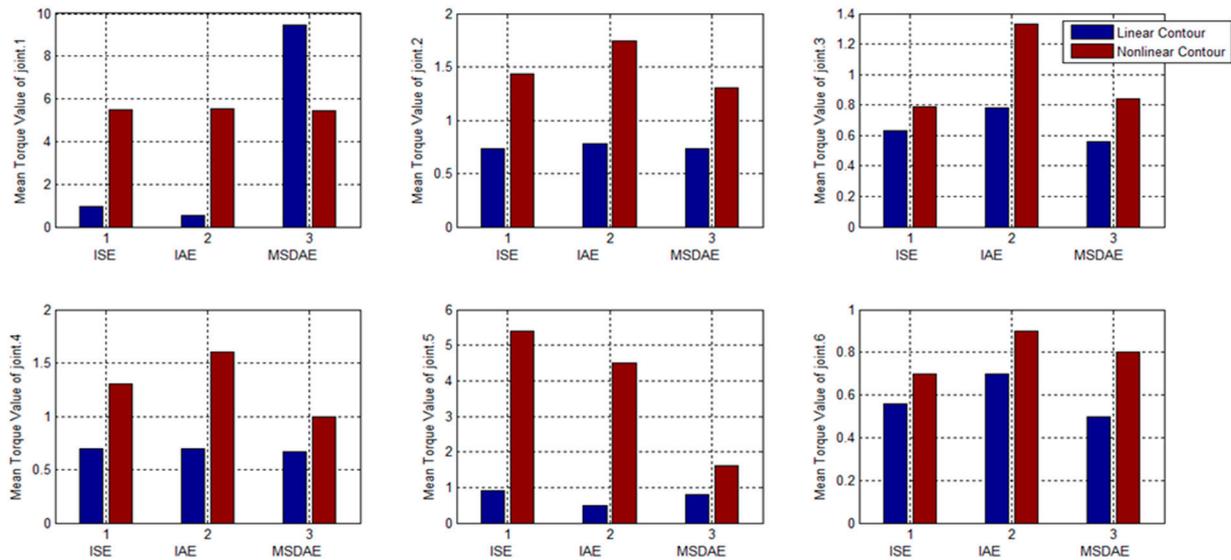


Figure 8. Mean torques required in linear and non-linear tracking.

Finally, the effect of position domain control on Joints 2 and 3 needs to be explored. The reason of choosing these two joints is that they have big variations in joint angle compared to other joints. Tests were conducted using two configurations. The first configuration assumed for joints angles is represented by $[-80, -45, -55, -80, 260, -94]$ and the second configuration is represented by $[35, -105, -75, 90, -125, 110]$. Figure 9a,b shows the step response to the first and second robot configurations for Joint 2 where PID refers to the design controller and dif. refers to the difference in joint angle between the optimum angle and the angle produced using the PID controller.

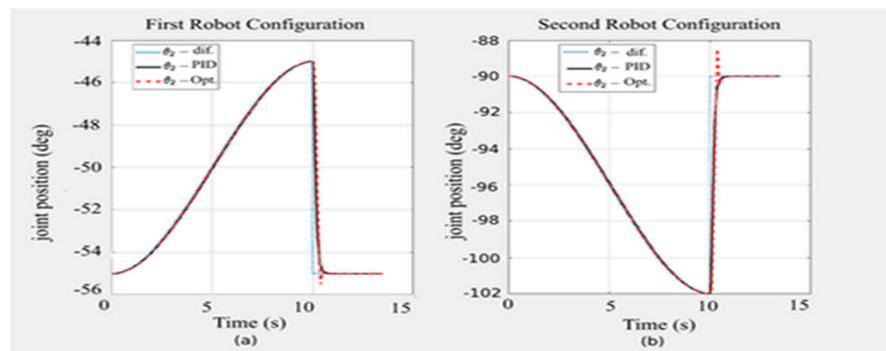


Figure 9. Step response of the UR5 robot for the two configurations of Joint 2.

Figure 10a,b show the step response to the first and second robot configurations for Joint 3.

The second section of the result is to improve the APF by DE algorithm to obtain the contour tracking with obstacle avoidance.

In the step response of the PDC-PID controller, there was no overshoot in contrast to other types of controllers. This is true only in linear contour tracking, while there was no overshoot observed in both linear and non-linear contour tracking when the DE algorithm is applied to improve the performance of PDC-PID. This result shows the advantage of using DE optimization algorithm in linear and non-linear contour tracking.

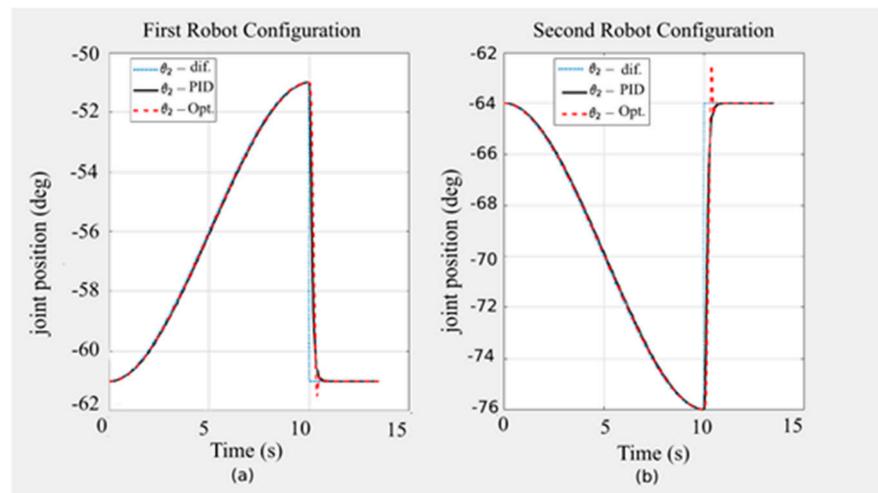


Figure 10. Step response of the UR5 robot for the two configurations of Joint 3.

APF parameters are often adopted empirically which makes the process expensive but with the application of DE algorithms, an improvement in the performance of the APF algorithm is achieved due to the generation of collision-free trajectories. APF-DE simulations were presented for a population of 30 individuals with 50 iterations. The APF algorithm with DE trajectory showed the smoothest trajectory. It was also possible to verify the average position error between the last required position of the path generated by the APF-DE and the final position. In terms of computational cost, the DE algorithm showed shorter execution time for the calculation of the parameters of APF algorithm. Advantages of these simulations can be attributed to the improvement of the APF algorithm, because the parameters are now being generated in optimized way and the precision of the points of the trajectories reached. Figure 11 shows the effect of DE algorithm for 30 individuals and 50 generations where the best individual and the population average are also given. Table 7 presents the intensity parameter values of the potential field computed through the locations of obstacles in the simulation using DE algorithm for 30 individuals and 50 generations. Figure 12 shows the lines and trajectory of the APF together with the collision-free path, in the Cartesian space for the simulation performed using the aforementioned parameters.

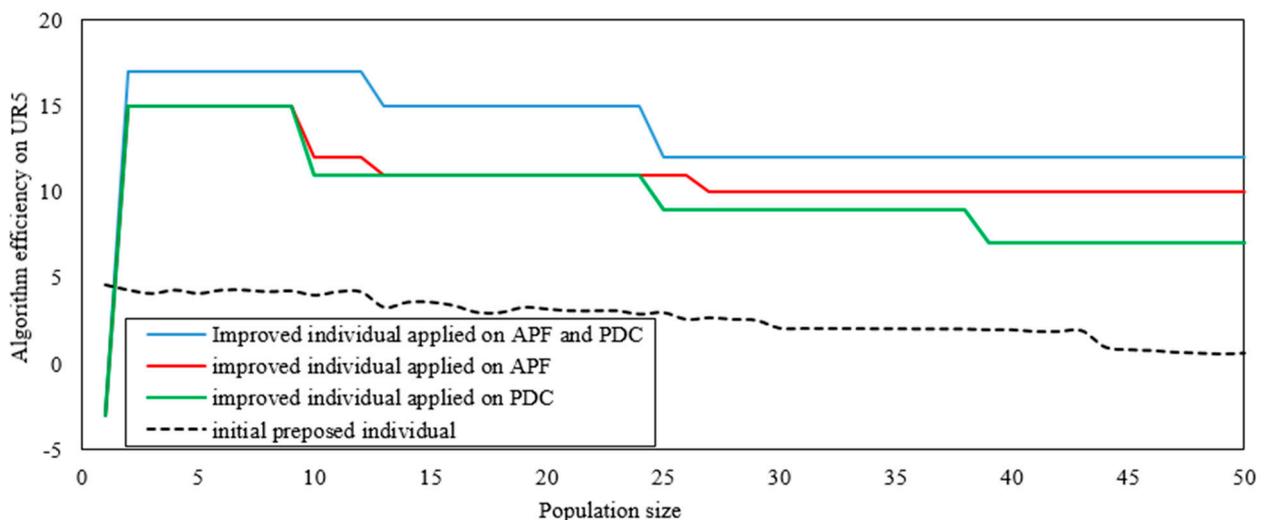


Figure 11. DE effecting with 30 individuals and 50 generations.

Table 7. Parameter values and intensities obtained by the DE.

Parameter	k_r	k_a	k_o
Intensity	1829	4230	2389

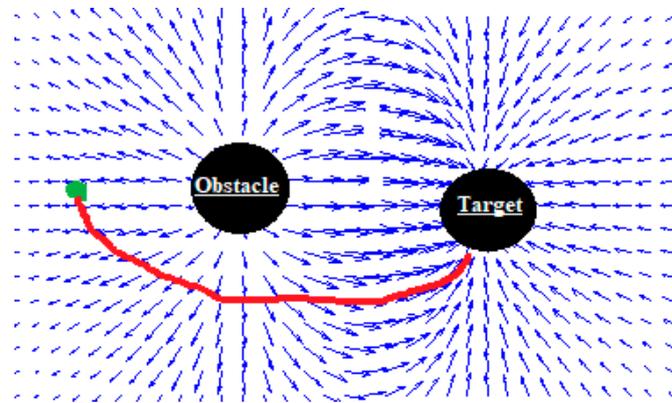


Figure 12. Trajectory and collision-free path performed by the APF for the parameters in Table 6.

Figure 13 shows the potential surface obtained from the DE-APF scene in Figure 12.

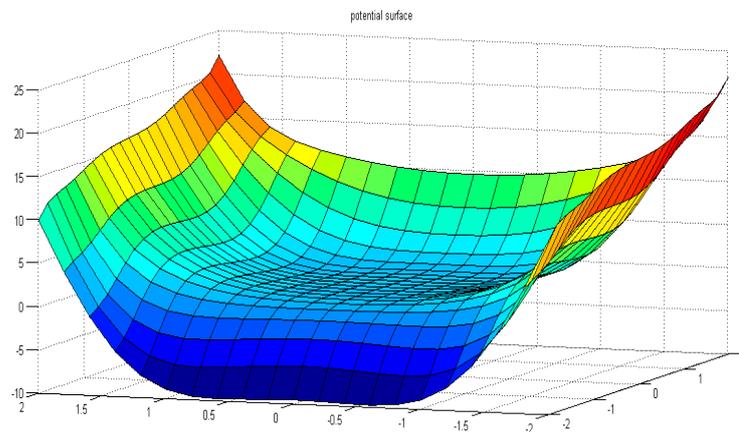


Figure 13. Potential surface obtained from the DE-APF scene in Figure 12.

Finally, Figure 14 represents the trajectories of UR5 manipulator in three cases, the first case is with PDC-PID controller to obtain circular contour, the second case is with APF to obtain collision free-path and the third case is with PDC-APF modified by DE which combines the two merits of the above two cases. Figure 15 shows the trajectory of UR5 manipulator in these three cases, when it moves from the starting position and avoids the obstacle in the contour path to reach the final position. In the first case, the manipulator moved from the starting to the final position but the distance between obstacle and manipulator was larger than in the second case and the distance in the second case was larger than the in third case which means that manipulator is following the desired contour path.

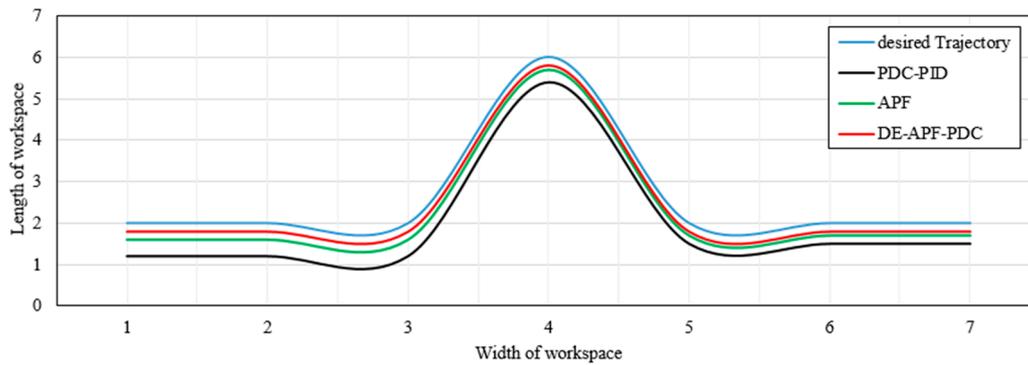


Figure 14. Trajectory of UR5 manipulator.

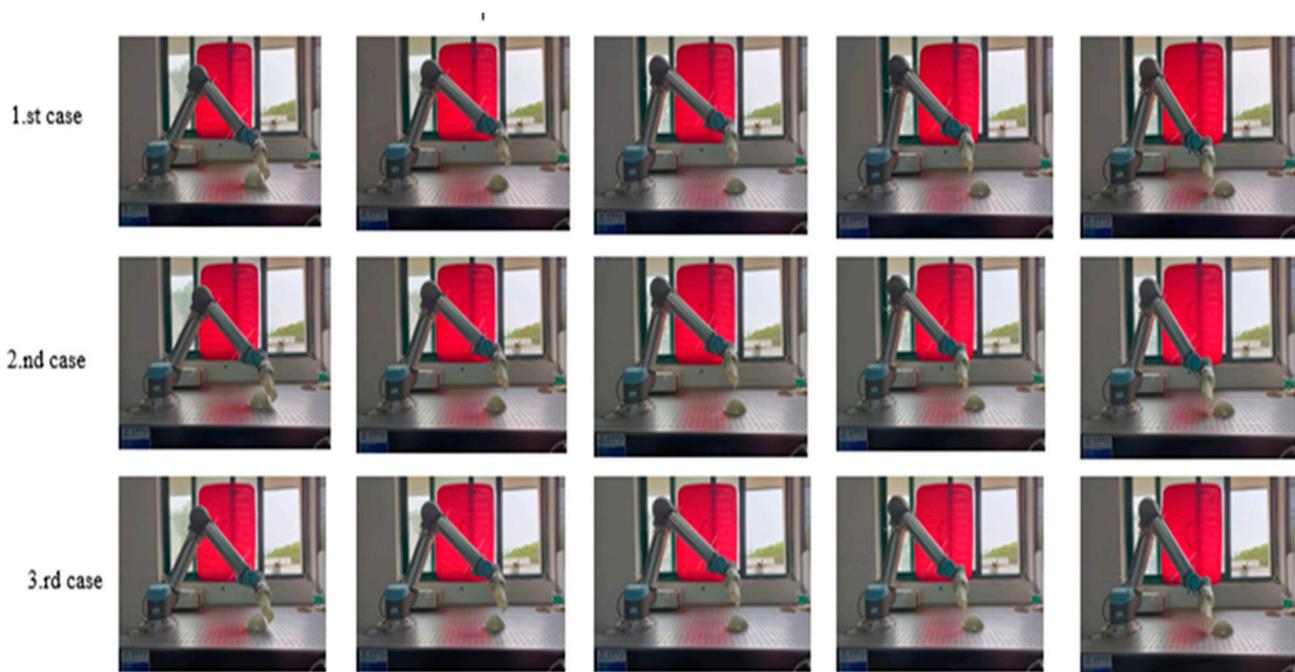


Figure 15. Trajectory of UR5 manipulator in three cases (PDC-PID controller, APF and with PDC-APF modified by DE).

As noted, to compare this paper with Ref. [8], we showed that the DE algorithm enhances the parameters of the APF approach and control on the PDC-PID controller. This was completed by helping a fitness function. The result of this study gave linear and nonlinear contour tracking with obstacle avoidance to the UR5 manipulator robot.

4. Conclusions

The novelty of this paper was that the DE algorithm can be used to modify APF and enhance the gain of position domain PDC-PID controller in the UR5 manipulator robot to minimize the contour tracking error and enhance the path of the UR5 robot. Three different fitness functions (ISE, IAE, and MSDAE) were used to determine optimization efficiency. The DE-ISE method resulted in minimum contour error in linear contour tracking. Minimum contour error in non-linear contour tracking is produced using the DE-IAE method. It has been concluded that the DE algorithm revealed good optimization in linear and nonlinear contours since the application of DE algorithm made it more flexible to avoid the local minimum. It is important to mention here that these optimization methods for gain tuning of the position domain controller can be extended to other controller designs made for contour tracking which suffered from manipulator problems. On the

other hand, the improved APF method with DE presented low computational cost allowing applications in dynamic environments for collision-free trajectories. The artificial potential field method requires adjustments of several parameters so that the generated path is satisfactory. As shown in this work, through the proper selection of the parameters k_a , k_r and k_o it is possible to reach the position and optimize the path through a minimum distance. Therefore, we conclude that in all of the evaluations carried out, the DE algorithm presented good results regarding generation of a smooth contour trajectory and better computational cost. In future work, there are many ideas to explore. One of them would be to apply the fuzzy decision on the APF approach to avoid the obstacle in online working, and modify the APF approach by connecting it with the Internet.

Author Contributions: Formal analysis I.J.K., L.Q.; Investigation, I.J.K.; Methodology, I.J.K. and L.Q.; Supervision Y.T., and L.Q.; Writing—original draft, I.J.K. and Y.T.; Writing—review L.Q. editing, Y.T. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by the Open Research Foundation of Advanced Innovation Center for Intelligent Robotics and Systems under Grant Number 2019IRS18, and the Fundamental Research Funds for the Central Universities of China under Grant Number 2021IVA019.

Data Availability Statement: Not applicable.

Acknowledgments: To my husband Omer W. Taha, my mother, my family, and to Ruiya Li who have helped me in all my research studies. We also thank all the anonymous reviewers for their insightful comments.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Li, C.; Jiang, H.; Ma, S.; Jiang, S.; Li, Y. Automatic Parking path planning and tracking control research for intelligent vehicles. *Appl. Sci.* **2020**, *10*, 9100. [\[CrossRef\]](#)
2. Siciliano, B.; Lorenzo, S.; Villani, L.; Orilo, G. *Robotics: Modelling, Planning and Control*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2010.
3. Bishop, R.H. *The Mechatronics Handbook*; CRC Press: Boca Raton, FL, USA, 2002.
4. Kim, Y.; Yoon, W.C. Generating task-oriented interactions of service robots. *IEEE Trans. Syst. Man Cybern. Syst.* **2014**, *44*, 981–994. [\[CrossRef\]](#)
5. Henrich, D. Fast Motion Planning by Parallel Processing—A Review. *J. Intell. Robot. Syst.* **1997**, *20*, 45–69. [\[CrossRef\]](#)
6. Lee, K.; Choi, D.; Kim, D. Incorporation of potential fields and motion primitives for the collision avoidance of unmanned aircraft. *Appl. Sci.* **2021**, *11*, 3103. [\[CrossRef\]](#)
7. Alvarez-Ramirez, J.; Cervantes, I. PID regulation of robot manipulators with elastic joints. *Asian J. Control.* **2003**, *5*, 32–38. [\[CrossRef\]](#)
8. Ouyang, P.; Pano, V. Comparative study of DE, PSO and GA for position domain PID controller tuning. *Algorithms* **2015**, *8*, 697–711. [\[CrossRef\]](#)
9. Cominos, P.; Munro, N. PID controllers: Recent tuning methods and design to specification. *IEE Proc. Control. Theory Appl.* **2002**, *149*, 46–53. [\[CrossRef\]](#)
10. Pae, D.S.; Kim, G.H.; Kang, T.K.; Lim, M.T. Path planning based on obstacle-dependent gaussian model predictive control for autonomous driving. *Appl. Sci.* **2021**, *11*, 3703. [\[CrossRef\]](#)
11. Kachitvichyanukul, V. Comparison of Three Evolutionary Algorithms: GA, PSO, and DE. *Ind. Eng. Manag. Syst.* **2012**, *11*, 215–223. [\[CrossRef\]](#)
12. Zappulla, R.; Park, H.; Virgili-Llop, J.; Romano, M. Real-Time Autonomous Spacecraft Proximity Maneuvers and Docking Using an Adaptive Artificial Potential Field Approach. *IEEE Trans. Control. Syst. Technol.* **2019**, *27*, 2598–2605. [\[CrossRef\]](#)
13. Barraquand, J.; Langlois, B.; Latombe, J.C. Numerical Potential Field Techniques for Robot Path Planning. *IEEE Trans. Syst. Man Cybern.* **1992**, *22*, 224–241. [\[CrossRef\]](#)
14. Park, C.; Kee, S.C. Online local path planning on the campus environment for autonomous driving considering road constraints and multiple obstacles. *Appl. Sci.* **2021**, *11*, 3909. [\[CrossRef\]](#)
15. Ge, S.S.; Cui, Y.J. Dynamic motion planning for mobile robots using potential field method. *Auton. Robot.* **2002**, *13*, 207–222. [\[CrossRef\]](#)
16. Roger, A.B.; McInnes, C.R. Safety constrained free-flyer path planning at the international space station. *J. Guid. Control. Dyn.* **2000**, *23*, 971–979. [\[CrossRef\]](#)

17. Galceran, E.; Eustice, R.M.; Olson, E. Toward integrated motion planning and control using potential fields and torque-based steering actuation for autonomous driving. In Proceedings of the IEEE Intelligent Vehicles Symposium, Seoul, Korea, 28 June–1 July 2015; pp. 304–309. [[CrossRef](#)]
18. McIntyre, D.; Naeem, W.; Xu, X. Cooperative obstacle avoidance using bidirectional artificial potential fields. In Proceedings of the 2016 UKACC 11th International Conference on Control (CONTROL), Belfast, UK, 31 August–2 September 2016. [[CrossRef](#)]
19. Bloise, N.; Capello, E.; Dentis, M.; Punta, E. Obstacle avoidance with potential field applied to a rendezvous maneuver. *Appl. Sci.* **2017**, *7*, 1042. [[CrossRef](#)]
20. Ouyang, P.R.; Dam, T.; Huang, J.; Zhang, W.J. Contour tracking control in position domain. *Mechatronics* **2012**, *22*, 934–944. [[CrossRef](#)]
21. Ouyang, P.R.; Pano, V.; Dam, T. PID position domain control for contour tracking. *Int. J. Syst. Sci.* **2013**, *46*, 111–124. [[CrossRef](#)]
22. Khatib, O. Real-Time Obstacle Avoidance for Manipulators and Mobile Robots. *Int. J. Robot. Res.* **1986**, *5*, 90–98. [[CrossRef](#)]
23. Villani, L.; de Schutter, J. Force control. In *Springer Handbook of Robotics*; Springer: Berlin/Heidelberg, Germany, 2016.
24. Borenstein, J.; Koren, Y. Real-Time Obstacle Avoidance for Fast Mobile Robots. *IEEE Trans. Syst. Man Cybern.* **1989**, *19*, 1179–1187. [[CrossRef](#)]
25. Sepehri, A.; Moghaddam, A.M. A motion planning algorithm for redundant manipulators using rapidly exploring randomized trees and artificial potential fields. *IEEE Access* **2021**, *9*, 26059–26070. [[CrossRef](#)]
26. Scotto, T.J. Book reviews: Book reviews. *Party Politics* **2011**, *17*, 525–527. [[CrossRef](#)]
27. Eberhart, R.; Kennedy, J. A New Optimizer Using Particle Swarm Theory. In Proceedings of the Sixth International Symposium on Micro Machine and Human Science, Nagoya, Japan, 4–6 October 1995. [[CrossRef](#)]
28. Panda, S. Robust coordinated design of multiple and multi-type damping controller using differential evolution algorithm. *Int. J. Electr. Power Energy Syst.* **2011**, *33*, 1018–1030. [[CrossRef](#)]
29. Pires, J.N. Robot Manipulators and Control Systems. In *Industrial Robots Programming*; Springer: Berlin/Heidelberg, Germany, 2007.
30. Volpe, R.; Khosla, P. Manipulator Control with Superquadric Artificial Potential Functions: Theory and Experiments. *IEEE Trans. Syst. Man Cybern.* **1990**, *20*, 1423–1436. [[CrossRef](#)]