T.C.

ALTINBAS UNIVERSITY

Institute of Graduate Studies

Electrical and Computer Engineering

DESIGN OF OPTIMAL TRAJECTORY PLANNING FOR ROBOT ARM USING MODIFIED SWARM ALGORITHM

Salam Ghanim NAJEEB

Master of Science

Supervisor

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Istanbul, 2021

DESIGN OF OPTIMAL TRAJECTORY PLANNING FOR ROBOT ARM USING MODIFIED SWARM ALGORITHM

by

Salam Ghanim NAJEEB

Electrical and Computer Engineering

Submitted to the Institute of Graduate Studies

in partial fulfillment of the requirements for the degree of

Master of Science

ALTINBAŞ UNIVERSITY

2021

The thesis titled “DESIGN OF OPTIMAL TRAJECTORY PLANNING FOR ROBOT ARM USING MODIFIED SWARM ALGORITHM” prepared and presented by “Salam Ghanim NAJEEB” was accepted as a Master of Science Thesis in Electrical and Computer Engineering.

____

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Supervisor

Thesis Defense Jury Members:

Asst. Prof. Dr. Abdullahi Abdu IBRAHIM	School of Engineering and Architecture, Altinbas University	__________________
Asst. Prof. Dr. Mesut ÇEVIK	School of Engineering and Architecture, Altinbas University	__________________
Asst. Prof. Dr. Sewale Musadaq TAHA	Faculty of Communication, Beykent University	__________________

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

Approval Date of Institute of Graduate Studies: ____/____/____

I hereby declare that all information in this document has been obtained and presented in accordance with academic rules and ethical conduct. I also declare that, as required by these rules and conduct, I have fully cited and referenced all material and results that are not original to this work.

Salam Ghanim NAJEEB

DEDICATION

First of all, I would like to thank my family **(**Dad and Mom**)** for their unyielding support and encouragement over the years. I am forever indebted to my parents, who always kept me in their prayers. Not only did they provide me with the best opportunities, but they also created a protected environment to help me succeed. The values my family instilled in me have always guided me. I hope completing my thesis will give them some of the happiness and comfort they really deserve. else I am very grateful to (who become rest in peace) for supporting me in all my life and especially during my studies.

I am also very grateful for the patience and effort that went into improving my writing and presentation skills to reach the high standards that he set, and I thank my friends and brothers who provided me with the moral support who stood with me throughout my master's degree my family, the source of happiness, and the first supporter (Dr.Riyam, Dr.Reeham, Dr. Noor Al-Huda, Mustafa) also, I would like to thank my teacher, Dalal Al-Sayegh, A. Mohammad Kazem, Dr. Ali Al-Issa, My big family Al Najeeb, Al Lafta, and dear Uncle Rahim Al Zerijawi, (My wonderful Friends Al Khazraji , Al-Fahham, Al-Budairi, Al Awade, Al-Salihi, Al-Sayegh, Al-Hasani, Al Dhalami, Al Aiaajiby, Al-Musawy)

My Colleagues

Mr. Barzan, Dr. Marwa Al-Majidi, Dr. Nour Al-Huda, Noha Walid, Abdullah AL-AWADI, Hussam Alhasany, Hussein Ali AL-KARAAWI, Ali Tarimish, Muhaimin Fayez, Ahmed Mahdi, Nawar AL-RUBAIE, Alaa Saad, Muhammad Amin, Mustafa Fadel, Hussein Muhammad, Amir Imad, Al-Mustafa AL-GHURIRI. Yusuf

ACKNOWLEDGEMENTS

I might want to thank my administrators: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM Please let me express my profound feeling of appreciation and gratefulness to both of you for the information: direction and unrestricted help you have given me. I want you to enjoy all that life has to offer and further achievements and accomplishments throughout your life and thanks to the discussion committee Asst. Prof. Dr. Mesut ÇEVIK and Asst. Prof. Dr. Sewale Musadaq TAHA I would also like to thank Asst. Prof. Dr. Sefer KURNAZ .

ABSTRACT

DESIGN OF OPTIMAL TRAJECTORY PLANNING FOR ROBOT ARM USING MODIFIED SWARM ALGORITHM

Salam Ghanim NAJEEB,

M.Sc., Electrical and Computer Engineering, Altınbaş University,

Supervisor: Asst. Prof. Dr. Abdullahi Abdu IBRAHIM

Date: 09/2021

Pages: 52

Much attention has been paid to improve, achieve the efficient performance of robot arm and to cope with the unlimited needs and demands of the worldwide industrial revolution that necessitate the availability of high productivity and accuracy. The finding of an optimal path possesses a significant role in guiding and attaining the accurate robot arm movements.

In the current work, the optimal path and trajectory planning of two-link robot arm with 2-DOF in 2-D static known environment has been analyzed by proposing the Particle Swarm Optimization algorithm (PSO) and cubic polynomial equations.

The primary function of modified PSO method is to generate the free-collision shortest path through searching within the free Cartesian space. The simulation results illustrate the efficiency of proposed methods performance in finding and determining the optimal path and trajectory even in various degree of environmental complexity.

Moreover, the experimental work has been tested for practical physical validation in order to check the quality of the planned path practically and to analyze the difficulties of optimal practical path planning practically.

Keywords*:* Robot arm, path planning, PSO algorithm

TABLE OF CONTENTS

Pages

LIST OF TABLES

Pages

LIST OF FIGURES

Pages

1. INTRODUCTION

1.1 BACKGROUND

A robot arm is a mechanical articulated manipulator that possesses an electronic mechanism for automatically controlled and reprogrammable, multipurpose arm programmable in three or more axes for the performance of a variety of tasks. A robot arm is either stationary in place or mobile which is extensively utilized in the industrial automation applications and has many other specialized applications. The re-programmability of robots is the computer brain that can be considered as the key element in granting and promoting robot utility and adaptability [1, 2, 3].

A robotic system is composed of many components that operate in the integrated reactive process namely: the arm, external power supply, end-of-arm tooling, external and internal sensors, computer interface, and control computer. In addition to programmed software, which is integrated with strong gearing systems and very accurate models. Moreover, the methods of programming and controlling of the robot have a significant role in impact on robot performance and subsequent range of applications [3]. Figure (1.1) illustrates components of a robotic system.

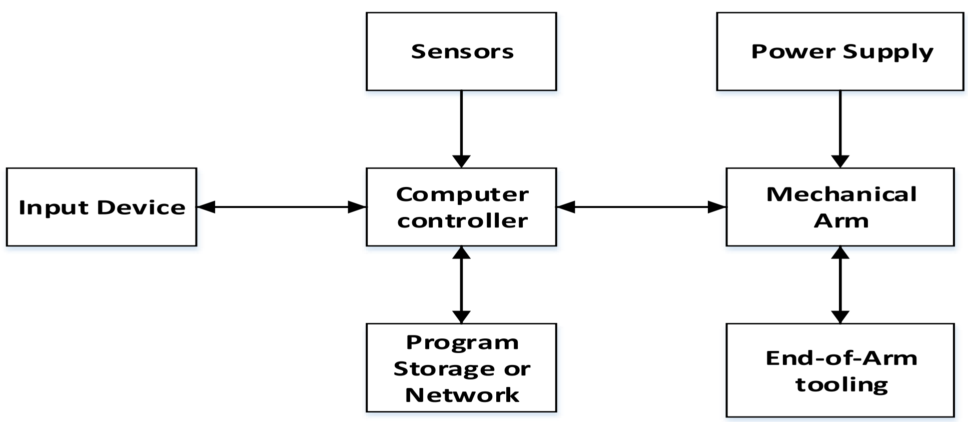


Figure 1.1: Components of a Robotic System [3].

A robot arm is composed of an assembly of multiple strong rigid segments with powerful motors. The rigid sections consist of links that interconnected through hinges or joints to form a kinematic chain. Joints or hinges represent the connections between two consecutive links that allow a relative motion to be rotary or linear. In the case of rotary joints (R), a revolute joint allows the restricted relative motion to be in a rotation state, and displacements are called joint angles. Whereas a prismatic joint (P) allows relative displacement between links to be translated and called joint offset. These joints are usually equipped with position sensors to measure relative position of consecutive links. Moreover, End-effector is rigid device attached to the arm at the free end of the links chain, and to interact with the environment for performing specific tasks depending on the intended application of the robot [2, 3, 4].

The study of robot arms encompasses control of robot and determining the positions and orientations of the several segments of arm. The complexities of control and accuracy of positions of robots may be better achieved by supplementing human capabilities with computer power. Moreover, control of an industrial robot should take into account nonlinearities, uncertainties, and external perturbations in the design of control laws [5, 6].

Robot arms are classified into several types according to various criteria, as power source, application area, geometry or method of control. Depending on power source, robots can be powered by electricity, hydraulic, or pneumatic. However, based on application area, robots are mostly categorized as assembly and non-assembly robots. Moreover, robots can be assorted kinematic ally into five geometric types based on the way in which the first three joints of the arm are actuated. The geometric types include an articulated (RRR), spherical (RRP), SCARA (RRP), cylindrical (RPP), or Cartesian (PPP). Furthermore, depending on the control method of arms, robots can be categorized into servo and non-servo robots. The non-servo robot is basically an open-loop device where its motion is confined to predetermined mechanical stops, and it is utilized primarily for transferring materials, whereas the Servo robot relies on the closed-loop computer control in determining its motion, therefore, it has the capability of being a multifunctional and reprogrammable device [3].

There is another motion classification of robot depending on the methods (ways) that are used by the controller for guiding the end-effector. The point-to-point robot is the simplest kind of robot, which can generate a discrete series of points. However, in the point-to-point robots, the path of end-effector in between taught points is not controlled. On the contrary, in continuous path robot, the robot end-effector can be taught to control and follow the complete path of the end-effector as well as its velocity and/or acceleration. Such classification has a significant role in determining which robot is convenient to used according to specifically given tasks [3].

1.2 APPLICATION OF A ROBOT ARM

A robot arm is typically outfitted for several industrial applications in order to carry out particular tasks repeatedly based upon predetermined movements and specifically located objects. Therefore, a robot arm can be operated and applied in [7, 8, 9]:

i. The most dangerous tasks for humans is purifying the master circulating pump housing in a nuclear power plant and retrieving suspicious objects without endangering humans.
ii. Stressful tasks that require intensive labor done by humans such as welding, material handling, thermal spraying and digging trenches.
iii. Pick and place task in food, electronics and consumer goods industry.
iv. Medical applications such as surgery, health care, pharmaceutical industry and laboratories.
v. Agricultural applications such as drilling and growing crops.
vi. Construction tasks or handling a damaged/radioactive building.
vii. Engineering applications, manufacturing military and space instruments.
viii. Tasks that require accessing unevenly placed parts.
ix. Tasks that simulate human behavior.
x. Building cars.

1.3 ROBOT ARM PATH PLANNING

Path planning is an essential task and the most crucial element in the robot operation that is useful primarily to find a set of continuous motions for moving the robot, beginning at the initial configuration and ending at the goal configuration. The main concern of path planning is upgrading and guiding the arm movement by determining only the kinematic (geometric) specifications of positions and orientations of the robot motion regardless of the dynamics of the robots [11,12].

The complexity of the robot environment is one of the very important characteristics of robots. Moreover, in environments that comprise static or moving obstacles including human workers and other robots, path planning is subjected to the constraints in finding the possible path, provided that no point in the path collides with any obstacle and the constructed path satisfies specific optimization criteria. Therefore, the path planning problem has been labeled as an optimization problem [11, 12].

The essential function of path planning is to create a path that satisfies the criteria and aims of motion including averting obstacles collision, reducing time interval, decreasing the path traveling cost and energy safety in addition to path smoothness. The shorter path with minimum traveling cost is the most typical case that is rapidly executed and required less energy. A path may be optimal or near-optimal unless path planning has satisfied the entire or most of the motion criteria. Nevertheless, the local optimum, complexity of computations and adaptability are the main difficulties that face the robot path planning problem [13, 14].

Robot Path Planning can be categorized according to search criteria including type of planning, environment information and execution time as shown in figure (1.2). Moreover, there are four main strategies that can be adopted in robot path planning, based on any available information regarding robot environment, its position and configuration, namely global or local planning, static or dynamic planning, off-line or on-line planning and complete or incomplete planning [11].

In the static path planning problem, the obstacles positions in the robot environment are fixed along the planning period, and the whole information of recognized robot environment is previously available. Therefore, the robot motions are planned depending on fixed existent information. While in the dynamic path planning problem, the information related to the dynamic environment of robot that contains the moving obstacles is partially or entirely recognized. Moreover, the partially used information is referred uniquely to the visible parts of the environment at the current robot motion. In this case, the robot re-plans a path based on updated information that was obtained by using different sensing devices until the given goal is achieved [15].

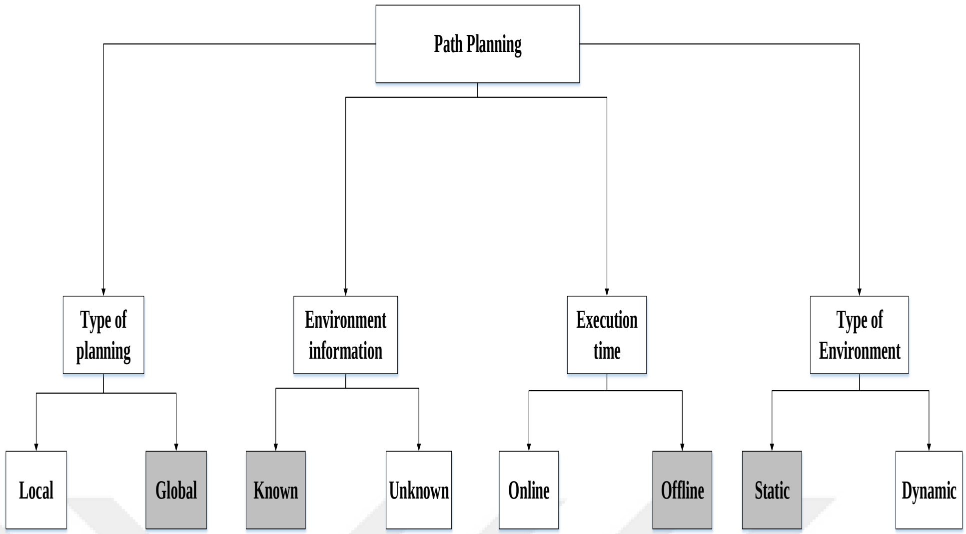


Figure 1.2: Path Planning Classification.

Meanwhile, the local and global path planning rely on the algorithm and availability of environment information. In the case of global path planning, the whole information related to the positions of stationary obstacles and the dynamic obstacles trajectories is a priori to the algorithm based on map, cells, grid or etc. Accordingly, the algorithm computes a complete path from the start point to the goal point in advance of robot following the planned motions. In contrast, the local path planning algorithm depends only on the information obtainable at the immediate robot movement for deciding the command of the next motion. Moreover, the algorithm needs the capability to build a map of the environment and to generate a new path in response to environmental changes, that essentially represent an iterative operation of moving to a new position, sensing the environment changes, updating the map and planning subsequent motion [14, 16, 17, 18].

Ideally, it is desirable that the path is generated online at the motion of the robot, due to the needs of the robot for capability to react to environmental changes. But there are many inherent difficulties that restrict robot motion from being computed online and sufficiently fast, namely the high dimensionality of the search space, the geometric nature of obstacles, the cost function to be optimized as well as the robot's kinematic and dynamic model [19].

In addition, the motion can be planned offline or online according to a global or a partial view of the environment respectively. In off-line planning, the entire path is generated prior to the robot motion, and it may be globally optimal unless the information on environment is fully known. Whereas path in the on-line planning is generated incrementally during motion and it is locally optimal at best. The online path planning is done initially offline and this depends on any available information about environment [19, 20].

However, the off-line planning has many challenges that has to achieve: the optimality in global planning, completeness of solution and complexity of computations. While the challenges in online planning that has to achieve are: the optimality or how far a solution from the optimal solution is, guaranteed completeness of solution, complexity of computation at every move [19].

The most significant category of robot path planning is determining the configuration of robot workspace where the robot path is computed. Moreover, the path may be constructed in either joint space or Cartesian space. In the joint space, the path is particularly identified for each distinct joint of robot while the Cartesian position is only predicted at the start and target positions. In contrast, the Cartesian path has the characteristic of being easily specified, and the robot motion is entirely identified. Since the path is not constructed in joint space, singularity should be taken into consideration in which the path is not located, or is close to it. Nonetheless, the crucial requirement of the real time path planning in Cartesian space is transforming the motion from Cartesian to joint coordinates. Accordingly, the velocity Jacobian is utilized to achieve Joint motion. [21].

1.4 LITERATURE SURVEY

Wenfu Xu, et al. (2008) [39] have proposed the point-to-point path planning of free-floating robot in Cartesian space. The free-floating robot with six-DOF. The path planning method based on the particle swarm optimization was proposed to compute the Cartesian path by integrating the differential kinematic equations. Initially, the polynomial functions have been applied to parameterize a smooth joint robot motion. Subsequently, the PSO would generate the Cartesian path by searching the parameters to be determined in Cartesian space. The experimental results showed efficacy of the PSO algorithm in verifying the Cartesian path of the free-floating robot.

Tsai C., et al. (2011) [40] presented a new combination method to solve the Inverse Kinematics (IK) problem and trajectory planning for a seven-DOF robotic arm. The PSO and GA were utilized to be combined by taking the characteristic of better population diversity and inheriting premature convergence. The developed hybrid method was supposed to generate a set of seven joint angles to manipulate based on the result of forward kinematic that was derived by Denavit-Hartenberg parameters, to move the arm from the start configuration to the end configuration.

The collision-free continuous trajectory of arm has been computed by utilizing the B-spline smoothing technique that connect the via-points generated by hybrid PSO-GA. The experimental results have shown that the proposed method of combining PSO-GA and B-spline were effective to generate a set of seven joint angles for arm moving from initial to desired configurations.

Guo J., et al. (2012) [41] Proposed the Cultural based PSO (CBPSO) algorithm to find the collision-free trajectory for redundant robot arms in the presence of fixed obstacles. The joint space was utilized in the proposed method to avoid Cartesian and singularity problems. The Quadrinomial and quintic polynomials have been used to describe the segment of the trajectory. In addition, PSO algorithm was applied to provide the intermediate via points of the trajectory. the proposed algorithm intended to find an acceptable collision free trajectory.

Rokbani and Alimi (2013) [42] have applied the Inverse Kinematics PSO (IKPSO) to find the inverse kinematics of robot arm. The objective was to generate the trajectory path by forward Kinematics depending on the joint variables that obtained from IK-PSO.

Yang Yueqiang and Tang Zhiwei (2014) [43] have dealt with the design of the path for welding arm. Particle swarm algorithm with quantum behavior were proposed to generate the path of welding robot by performing the weight code of particles based on the DELTA potential. The path planning problem of welding a robot was considered as a TSP problem which was transformed into a continuous optimization problem in order to expand the search scope of the algorithm to the space of the entire feasible solutions. The QPSO algorithm adopted this space for determining the path of the sequence of welding based on the weight of spots while passed by a certain particle. Experimental results showed that QPSO was more efficient in solving a sequence of welding arm path and specifically TSP problem.

Jeong-Jung Kim (2015) [44] has presented the optimization technique for trajectory optimization based on PSO algorithm to robot arm. The robot arm with three links and its trajectory has been planned in a three-dimension space with certain constraints. The PSO algorithm has been utilized to provide optimization arm trajectory by global searching. The PSO algorithm was combined with the Normalized Step Cost (NSC) concept in order to initialize its particles, in addition to developing the PSO method of reusing previously optimized parameters for improving its convergence in trajectory planning. Each of the new optimized parameters has been stored with corresponding vectors of NSC to select the identical parameters with the query NSC vector when they occur as a constraint violation. From experimental results, the modified PSO based on NSC in initializing its particles has successfully optimized a trajectory while satisfying the constraints.

1.5 WORK OBJECTIVES

The primary aim of this thesis is to demonstrate the optimal robot arm path in order to guide the movement of robot arm from start to goal configurations without collision with obstacles in the static environment by developing the swarm algorithms based on a hybrid optimization algorithm**.**

1.6 THESIS ORGANIZATION

This thesis is divided into five chapters including the present chapter. They are as follows:

Chapter Two: Includes the robot arms modeling and presenting the geometric problems with Cartesian paths and briefly reviews the swarm algorithms and path planning technique.

Chapter Three: Presents the proposed swarm algorithms of robot arm path planning.

Chapter Four: Illustrates the simulation work of two revolute links robot arm.

Chapter five: Presents the conclusions of this thesis and some suggestions for future work.

2. ROBOT KINEMATICS AND ALGORITHMS

2.1 INTRODUCTION

Robot arm is an extremely essential technology in the industrial world. Since industrial robots have various types of tasks that have to be efficiently achieved, the industrial robots should possess the primary functional requirements of being highly productive, of high quality and adaptability. However, this variety and complexity of tasks give rise to many problems such as finding the Inverse kinematic, forward kinematic, robot path planning and robot trajectory planning**.**

In reference to a manipulator of the robot, kinematics is concerned with description the analytical relation of the joint positions with position and orientation of the end-effector. Moreover, the path planning is considered as a very significant aspect in robotics, and it has been received great attention in studies over the past decades. Robotics, artificial intelligence, and control are the main important areas where the path planning is used. Searching for a free collision trajectory for driving a robot from the initial point to the goal point under various limitations is the common problem that faces the path planning**.**

This chapter describes the main concepts of robot arm kinematic and geometric problems with Cartesian paths, as well as path planning problem based on optimization techniques for the two-links robot arm that is the main part of the industrial manipulator and can be planned in the Cartesian space with 2-DOF.

2.2 ROBOT ARM KINEMATICS MODELING

Kinematics is the geometry of motion that deals with the object by analytically studying the motion mechanism regardless the forces and torques that give rise to it. The construction of the appropriate form of kinematics models is considered as a very important issue for studying the robot mechanism behavior [3,45, 46].

The robot kinematic problems can be categorized into two types: forward and inverse kinematics problems.

2.2.1 Forward Kinematics of Manipulators

The forward kinematics problem is considered as an essential issue for studying and analyzing the mechanical operation and guidance of the robot. The forward kinematics is a static geometrical problem that allocates the position of a robot end-effector and its orientation with the values provision for the robot joint variables. Particularly, the forward kinematic problem is concerned with determining the corresponding robot end-effector position and orientation for a specified set of joint angles [45].

Figure (2.1) demonstrates the 2-DOF robot arm having l_1 and l_2 as their link lengths and θ_1 and θ_2 as joint angles with x and y as task coordinates with p_x and p_y as robot end-effector position.

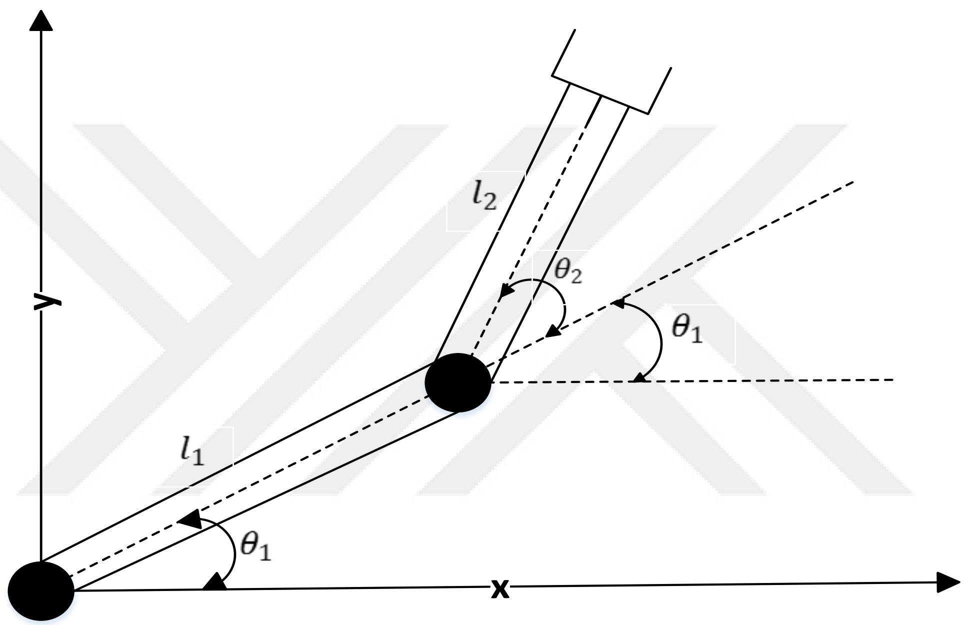


Figure 2.1: 2-DOF Manipulator.

The forward kinematic equations are [3]:

$$p_x = l_1 \cos\theta_1 + l_2 \cos(\theta_1 + \theta_2) \qquad (2.1)$$

$$p_y = l_1 \sin\theta_1 + l_2 \sin(\theta_1 + \theta_2) \qquad (2.2)$$

2.2.2 Inverse Kinematics of Manipulators

The inverse kinematics problem is concerned with the identification of the whole probable and suitable sets of robot joint variables that would realize the determined positions and orientations of the robot's end-effector. In contrast to the forward problem, the inverse kinematics problem would not specify constantly a unique solution. Specifically, one end-effector position is reachable with various configurations [47, 48, 49].

The solution techniques for the inverse kinematics problem can be classified into analytical and numerical methods. Moreover, the analytical method is subdivided into two approaches:

geometric and algebraic solutions. The geometric solution approach is utilized in case of simple robot structures such as 2-DOF robot arm with revolute joints. The essential function of Geometric solution approach depends on decomposing the robot locative geometry into many plane geometry problems [11].

Consequently, the following equations of geometric solution approach are applied to calculate θ_1 and θ_2 of the 2-DOF planar robot arm. Consider the diagram of Figure (2.2). Using the law of cosines that the angle is given by:

$$cos\,\theta_2 = \frac{p_x^2+p_y^2-l_1^2-l_2^2}{2l_1l_2} =: D \qquad (2.3)$$

Since, the sin θ_2 is

$$\sin\theta_2 = \mp\sqrt{1-D^2} \qquad (2.4)$$

Accordingly, the two probable solutions to θ_2 can be attained by:

$$\theta_2 = \text{Atan2}\,\frac{\mp\sqrt{1-D^2}}{D} \qquad (2.5)$$

Subsequently, $\theta_1$can be determined by:

$$\theta_1 = \text{Atan2}\left({}^{y}/_{x}\right) - \tan^{-1}\frac{l_2\sin\theta_2}{l_1+l_2\cos\theta_2} \qquad (2.6)$$

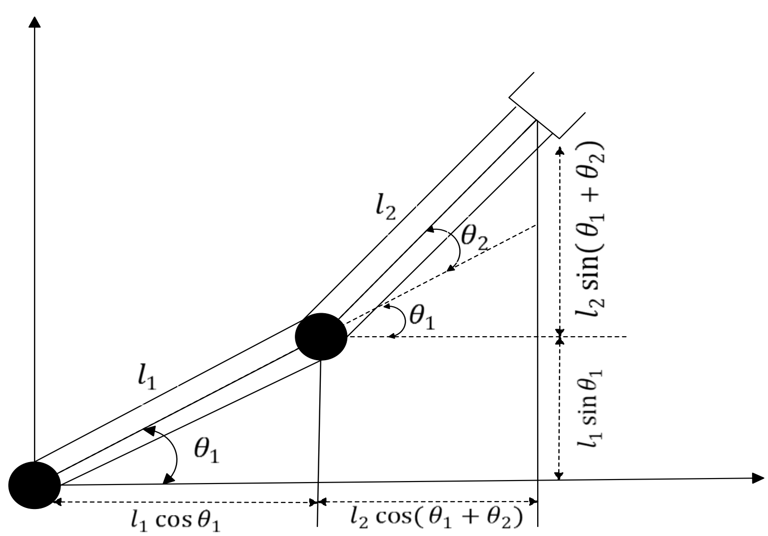


Figure 2.2: Solving The Inverse Kinematics Based on Trigonometry.

The major characteristic of this approach is the ability to retrieve the two elbow-up and elbow-down solutions by selecting both negative and positive signs in the equations [3, 50].

Based on the configuration of a two-links robot arm with two revolute joints, three possible solutions to the inverse kinematics problem essentially are identified. In the first case, precisely one solution might be identified in the situation where the robot is completely stretched out to attain the point. Whereas in the second case, two solutions might be captured if a given (p_x, p_y) coordinate point is within the reach of the robot as illustrated in Figure 2.3, while in the third case, no solutions might be identified in the condition where a given (p_x, p_y) coordinate point is out of robot reach [3].

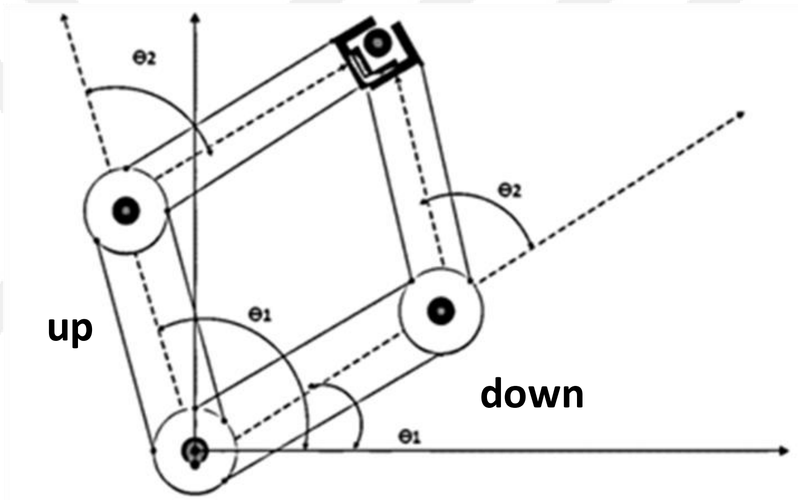


Figure 2.3: Multiple Inverse Kinematic Solutions.

2.3 GEOMETRIC PROBLEMS WITH CARTESIAN PATHS

Since the path can be constructed in the Cartesian and the joint spaces, Cartesian paths are faced with a variety of problems related to robot workspace and singularities [45].

i. Problem 1: Intermediate Points Unreachable.

Despite the fact that the initial and goal positions of the robot end-effector lay in the robot workspace, it is highly probable that not the entire intermediate points on a straight line connecting the two positions are placed in the robot workspace and subsequently unreached by a robot end-effector. As shown in Figure 2.4

This problem arises in the presence of differences between robot links lengths.

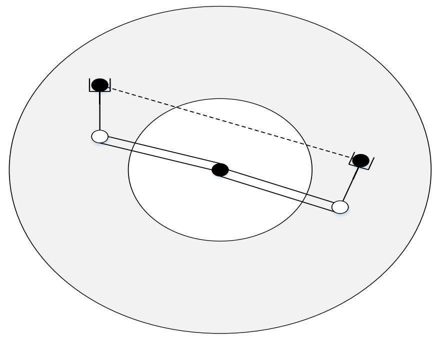


Figure 2.4: Cartesian-Path Problem of Type 1.

ii. Problem 2: High Joint Rates near Singularity.

There are areas in the robot workspace where it is difficult to select fixed joint rates that would achieve the required speed of the robot end-effector in Cartesian space. Moreover, several straight-line paths constructed in Cartesian space are impossibly performed by the robot due to the fact that one or more joint velocities might increase toward infinity as Figure (2.5).

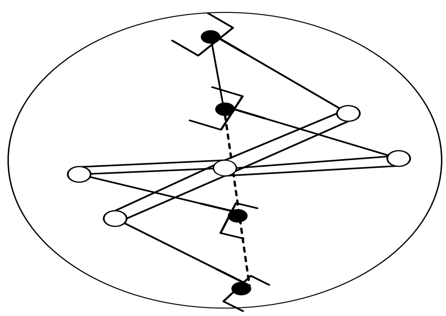


Figure 2.5: Cartesian-Path Problem of Type 2.

iii. Problem 3: Start and Goal Reachable in Different Solutions.

A problem emerges in the case where a robot cannot reach the final desired position in the same physical solution that a robot can reach the initial position.

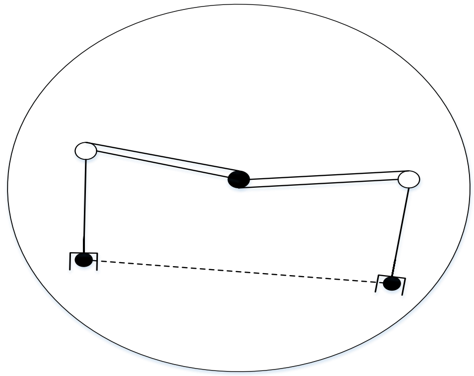


Figure 2.6: Cartesian-Path Problem of Type 3.

In Figure 2.6, not all the points of the path can be reached by a robot in the same physical solution.

The problems of difficulties of Cartesian paths can be solved by supporting both the joint-space and Cartesian-space path generation [45].

2.4 PARTICLE SWARM OPTIMIZATION

PSO is a technique of metaheuristic optimization, which operates on a base or a rule that simulates the social behavior of birds flocking. The PSO procedure is based on an evolutionary computation, which has been advanced, by Kennedy and Eberhart [63, 64].

Simplicity, rapid convergence, power and imposed few tuned parameters are the main features that differentiate it as superior to the other complex optimization algorithms. Consequently, PSO is utilized to efficiently solve the optimization problems, and has been applied in numerous robotic applications including: Robot Path Planning, mobile robot navigation, Swarm Robotic Applications and Robot Manipulators applications [65,66].

In a PSO system, the individual swarm that flying within the search space called particles, and each particle is considered as a candidate optimal solution for an optimization problem. Every particle is formulated as an object, which possesses many features that can be assigned the following codes: the current particle position (pos_i), the current particle velocity (vel_i), the particle personal best position ($pbest_i$) which indicate the best position a particle can achieve, the particle global best position ($gbest_i$) which indicates the best position achieved by the whole particles [67, 68].

The particle position is basically impacted by both the particle personal best position and the particle global best position, where the best personal position is demonstrated in the following updated equation:

$$pbest_i(t+1) = \begin{cases} pbest_i(t) & if\ f\big(pos_i(t+1)\big) \geq f\big(pbest_i(t)\big) \\ pos_i(t+1) & if\ f\big(pos_i(t+1)\big) < f\big(pbest_i(t)\big) \end{cases} \quad (2.13)$$

Where t represents the time step, and symbol f is used to denote the objective function that is being minimized [65, 67].

Moreover, the updated equation of the particle global best position is demonstrated as follows:

$$gbest(t) \in \{pbest_0, pbest_1, \dots, pbest_s\} = \min\{f(pbest_0(t)), f(pbest_1(t)), \dots, f(pbest_s(t))\} \quad (2.14)$$

Various PSO models have been evolved depending on the topology of the neighborhood, namely lbest and gbest. In the condition where the whole swarm represent the particle neighborhood, the $gbest$ PSO is the resulted algorithm, and the global best position of the particle is considered as the best position in the neighborhood. While the *lbe*st PSO algorithm can commonly be assigned when smaller neighborhoods are used, and the best particle is tracked by each particle, which is gained by the particle local topological neighborhood [65, 67, 68].

The PSO algorithm runs on the following principle: at first, a group of particles is initialized with randomly distributed position within the workspace. Furthermore, the velocity for these particles is also assigned randomly. Each particle possesses a memory, which serves as a position vector to store all the best previously visited positions, in addition to the fitness function value that is improved over time. Along the iterations, both global and local best values are updated. Subsequently, particle velocity and positions are updated accordingly. Moreover, the particles are converged towards either the personal best vector ($pbest_{ij}$) or the swarm's best vector ($gbest_{ij}$), while the position change rate of particle basically depends on its own velocity [68].

The $vel_{i,j}^{t}$ vectors of particle i are modified according to equation (2.15)

$$vel_{i,j}^{t+1} = w \times vel_{i,j}^{t} + c_1 \times r_{1,j} \times \left(pbest_{ij} - pos_{ij}\right) + c_2 \times r_{2,j} \times \left(gbest_{ij} - pos_{ij}\right) \quad (2.15)$$

Where w is the inertia weight that controls the particle momentum, c_1 and c_2 are the cognitive coefficients (c_1+c_2<=4), and $r_{1,j}$ and $r_{2,j}$ are random real numbers between [0, 1]. However, the value of $vel_{i,j}^{t}$is set between the range [-vel_{max}, vel_{max}] to ensure that there is no possibility of particles going away from the work space. When the search space is specified with the bounds [-pos_{max}, pos_{max}], the value of $vel$is clamped as $vel_{max} = k * pos_{max},$ where $0.1 \leq k \leq 1.0$.

Equation 2.15 comprises three segments, to be specific:

i. The inertia weight (w) works as a prior velocities memory of particle for controlling the effect of the past velocities. Where a large inertia weight and a small inertia weight support the exploration and exploitation respectively. Moreover, it decreases from (w_{max} = 0.9 to w_{min} = 0.4) over the whole run according to the following equation (2.16)

$$w = w_{max} - \frac{w_{max} - w_{min}}{\text{iteration}_{max}} * iteration\_num \qquad (2.16)$$

ii. The cognitive segment $(pbest_{ij} - pos_{ij})$ indicates individual experience of the particle that guides to the best solution.

iii. The social segment$(gbest_{ij} - pos_{ij})$ indicates the experience of entire particles that guides towards the best solution.

The position vectors of each particle is updated using the modified velocity vector for that particle by the following equation (2.17)

$$pos_{ij}^{t+1} = pos_{ij}^{t} + vel_{ij}^{t+1} \qquad (2.17)$$

This iterative process is terminated according to the following stopping criteria:

iv. Detecting the successful global optimal solution with desired fitness.

v. Achieving the specific iterations number.

vi. Velocity adjustment is close to zero.

As soon as one of the terminal conditions is achieved, the best position and best velocity are evaluated as the best optimal solutions found [68, 69, 70].

The basic PSO algorithm is illustrated in Figure (2.10).

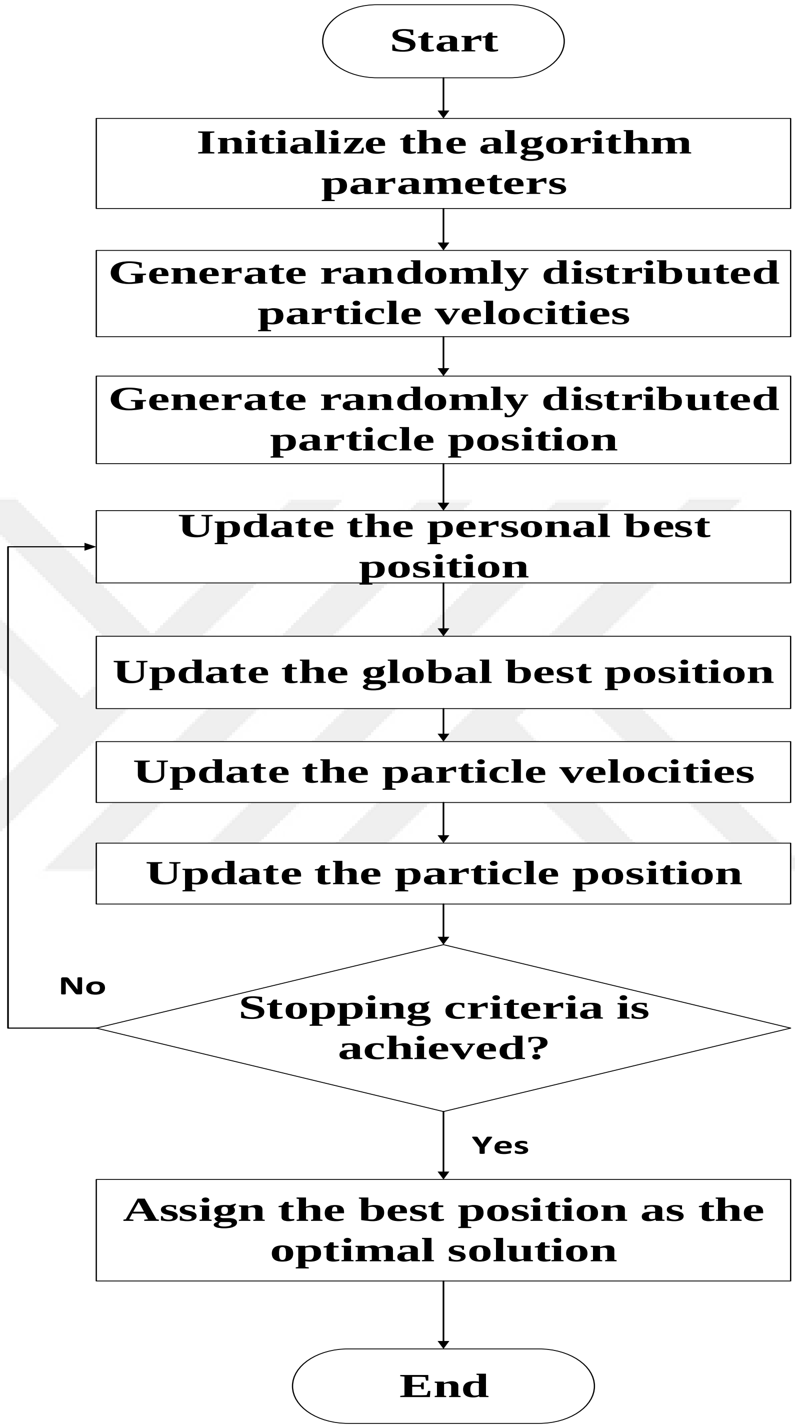


Figure 2.7: Basic PSO Flowchart.

2.5 POINT TO POINT TRAJECTORIES PLANNING

Cubic Polynomial Trajectories technique is one of the essential techniques that generates a smooth trajectory at a specific time. Cubic polynomial trajectory is specified by position in addition to velocity to generate smooth curve, so that it has four constraints, which are as initial and final for each one of configurations and velocities [49, 71, 72].

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \qquad (2.18)$$

According to equation (2.18) at time t_0, the initial position and velocity equations are:

$$q(t_0) = q_s \qquad (2.19)$$

$$\dot{q}(t_0) = v_s \qquad (2.20)$$

$(q_s \; and \; v_s)$ represent the initial position and velocity respectively. While at time $\mathrm{t_f}$ the final position and velocity equations are

$$q(t_f) = q_g \qquad (2.21)$$

$$\dot{q}(t_f) = v_g \qquad (2.22)$$

In equations (2.21) and (2.22), $(q_g \; and \; v_g)$ are representing the target position and velocity respectively. Depending on appropriate number of derivatives for equation (2.18), the following equations have been obtained:

$$q_0 = a_0 + a_1 t_0 + a_2 t_0^2 + a_3 t_0^3 \qquad (2.23)$$

$$v_0 = a_1 + 2a_2 t_0 + 3a_3 t_0^2 \qquad (2.24)$$

$$q_f = a_0 + a_1 t_f + a_2 t_f^2 + a_3 t_f^3 \qquad (2.25)$$

$$v_f = a_1 + 2a_2 t_f + 3a_3 t_f^2 \qquad (2.26)$$

As a formulated equation above, all parameters that should be computed are the joint angle of each initial, intermediate and goal points, and joint angular velocity of initial, intermediate and goal points.

The result of implementing cubic polynomial trajectory equations is a smooth trajectory that starts from the initial point, passes through intermediate points and terminates at the goal points [49, 71, 72].

3. PROPOSED ROBOT PATH PLANNING

3.1 INTRODUCTION

The main goal for attaining an autonomous robot arm is the planning for a collision-free path. The design of the path planning method is essentially the predominant concerned issue in robotics that aimed at finding a possible path moves Two-link robot arm from specific initial to the target configurations, provided that no point in the motion trajectory nor on the links of the arm collide with any obstacle, and ensures an optimal shortest path.

However, the optimality of the planned path is guaranteed when the path satisfies the aims of motion in terms of averting obstacles collision, reducing time interval, decreasing the path traveling cost and realizing the kinematics constraints.

The PSO algorithm and cubic polynomial equations hybridization explain in detail for off-line global optimal path and trajectory planning for the Two-link robot arm in a static environment. Moreover, the principle of the proposed improvement methods is depending on the implementation of the basic metaheuristic algorithms with some modification or hybridization.

3.2 OPTIMAL PATH PLANNING METHOD

Improvement of PSO based metaheuristic algorithm is proposed to optimize the path planning in the current study. Simplicity, rapid convergence, power and imposing of few tuned parameters are the main characteristics that encourage the use of PSO in robot arm path planning design. However, PSO algorithm suffers from the problem-dependent-performance problem that results from parameter settings of the algorithm, while utilization of different parameter settings can give rise to high-performance variances. Consequently, this problem can be efficiently solved by proposing hybridization, namely the PSO algorithm and cubic polynomial equations.

The objective of the proposed PSO method is to find an optimal path that satisfies motion constraints and moves the arm from an initial (start) configuration to a final (goal) configuration by avoiding all obstacles.

The proposed modification consists of five stages. Figure (3.1) demonstrates the stages of the proposed improvement of two-link robot arm path planning based on PSO algorithm.

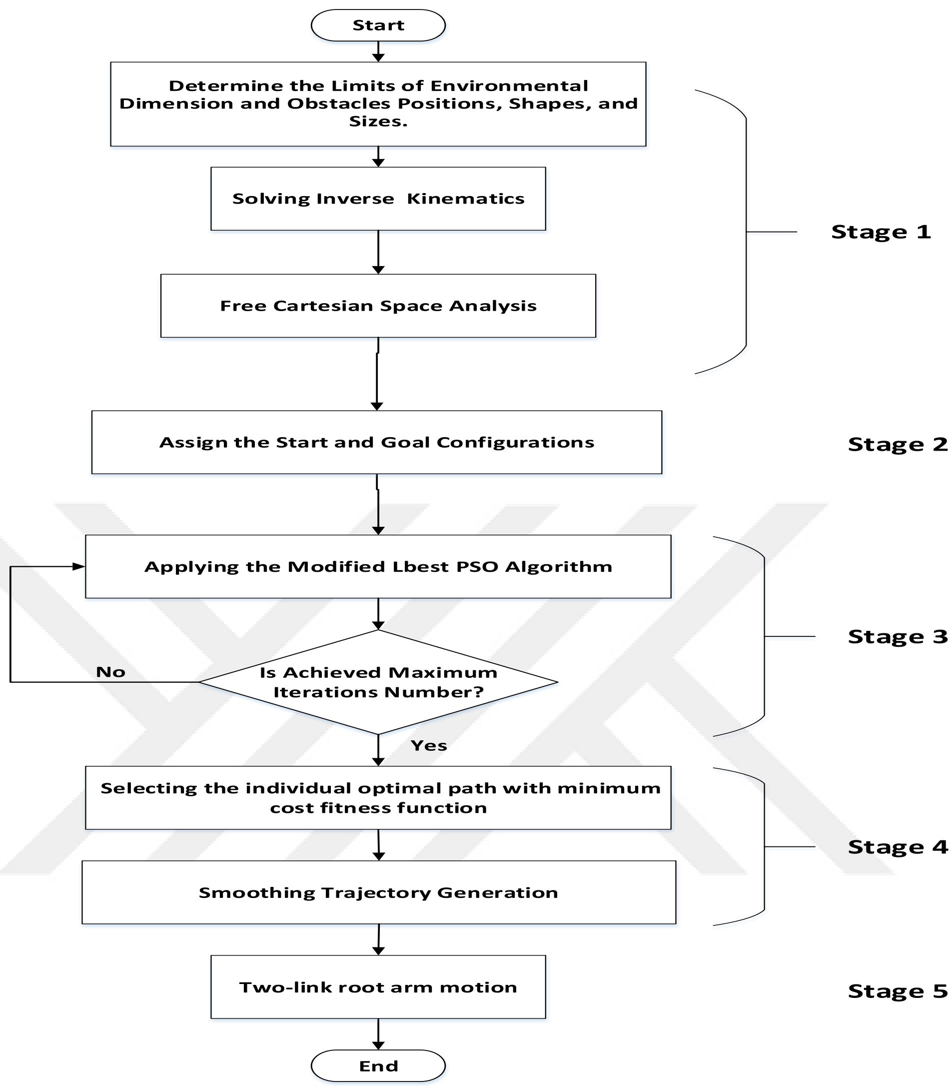


Figure 3.1: The General Proposed Flow Chart of Improvement of Two-Link Robot Arm Path Planning Based on Lbest PSO Algorithm.

The main five stages of the proposed method are:

1-Robot Arm Environment Selection

The essential stage of design the proposed system of robot arm planning depending on D* based ACO algorithm is to select and analyze the robot arm environment. The current stage comprises two phases, namely definition of robot arm environment followed by the free Cartesian space analysis.

In the first phase, the proposed system is initialized by selecting a robot arm environment, where D* based ACO algorithm and cubic polynomial equations plan an optimal shortest path and the smooth trajectory respectively. The proposed system preserves a set of different robot arm environments and each one is associated with specific definition of the workspace dimension measurement as well as obstacles positions, shapes and sizes, length of robot arm links and joint limitation.

In the second phase, the free Cartesian space is analyzed according to the process of free Cartesian space analysis in order to detect the free space and guarantee a collision-free path planning accordingly.

2-Star and goal Selection

The start and goal points are selected from the free Cartesian space. However, prior to planning the robot path, the start and goal points should be examined and ensured within the free Cartesian space. Otherwise, there will be no task for a path and trajectory planning.

3- Implementation of Proposed method PSO Algorithm

To solve the path planning problems for Two-Link arm, the modified PSO algorithm is put forwards to generate a free -collision path. The PSO algorithm is locally searched through workspace (free Cartesian space) to find an optimal arm path from its start to goal configurations according to several steps.

In the primary step, the start node with an initial velocity that is equal to zero is considered as the currently tested node. Then the current node is expanded to search for next tested nodes that are tested and analyzed by applying the velocity equation of PSO algorithm. The expansion process operates by determining the eight-connected neighborhood nodes (neighbor particles) that would be preserved in the expansion list. However, only the nodes belong to the free Cartesian space are preserved in the expansion list in order to provide a guarantee of collision-free path planning. Moreover, the checking function of free-collision elements is used for testing the pertinence of the node to the free Cartesian space.

Subsequently, the velocity equation (3.1) of the proposed Lbest PSO algorithm operates as the probabilistic testing function to find the next node of optimal arm path by computing and analyzing the fitness function of each node in the expansion list.

$$vel_{ij}^{t+1} = w \times vel_{ij}^{t} + c_1 \times R_1 \times \left(pbest_{ij} - pos_{ij}\right) + c_2 \times R_2 \times \left(gbest_{ij} - pos_{ij}\right) \qquad (3.1)$$

The coefficients R_1 and R_2 [0, 1] have a significant role in guiding the solution either towards the node with minimum D* cost or toward the node with minimum Euclidean distance. In addition to coefficients c_1 and c_2 that provide improved performance for analysis. Then only one node (particle) with minimum cost and high probability is selected by updating a position equation to be at the current node in the proposed path.

In the case of a current node has non-tested neighborhood nodes on the expansion list, then the current node is considered as dead point and the proposed *Lbest* PSO algorithm has to be returned to the last node in the path and another node has to be selected. This case appears when one or more of the eight nodes belong to the passing path while the rest are out of the free Cartesian space. However, all dead points are preserved in the dead point list to be avoided in the next iteration.

While in the next step, the new velocity for each particle (potential solution) is set depending on its prior velocity vel_{ij}^{t}. The lbest PSO is repeatedly generating and testing eight-connected neighborhood nodes for current node until reaching the goal point. In the proposed Lbest PSO algorithm, the constructed path can be considered as the candidate optimal path.

This process is iterated until a specific number of iterations are achieved or a minimum cost is obtained. Finally, the optimal path is formed from entire iterations via selecting the best candidate solution with the minimum cost function which can be calculated by equation (3.16). Depending on the number of the grid, the optimal path may be more than one path. Each path has the same cost but differs in some middle points.

$$\text{Cost}=\sum_{i=1}^{m} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (3.2)$$

Where, m is the number of particles that is equal to the number of nodes along the path, $(x_i\ , y_i)$ represents the current position of i^{th} particle.

4-Smooth Trajectory Planning Design

In the fourth stage of the proposed method, the smooth trajectory planning design is suggested to use cubic polynomial equations for providing smooth trajectory at the specific time $(\mathrm{t}_f - t_0)$. The proposed trajectory planning can generate a smooth trajectory depending on generated path that is planned by the proposed PSO method.

The proposed smooth trajectory planning method initializes with the definition of the number of intermediate via points as well as interval time $(\mathrm{t}_f - t_0)$. Then the proposed method randomly selects positions of intermediate via points from planned path of PSO method with respect to the number of these points, in addition to initial and goal configurations. Accordingly, the cubic polynomial equation is applied to describe the corresponding smooth trajectory by generating segments that connect the initial, intermediate via points, and goal points at the specified time.

As soon as the smoothing trajectory is defined, the collision checking function is applied to check whether all points of the trajectory are within the free Cartesian space. In case the value of verification function is true, then the trajectory is defined as the smoothing trajectory of the proposed method. Otherwise, the proposed method iteratively selects another random intermediate via point from *Lbset* PSO path and generates the corresponding trajectory until achieving the accepted one.

The cost function of optimal smooth trajectory that is constructed from the entire proposed method can be calculated by equation (3.3).

$$\text{Trajectory- Cost} = \sum_{i=1}^{m} \sqrt{(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \qquad (3.3)$$

Where, m is the number of configuration, which is equal to the amount of specific time along the trajectory, (x_i , y_i) represents the current position of i^{th} points of trajectory.

5- Two-Link Robot Arm Movement

In the fifth stage of the proposed *Lbest* PSO method, the two-link robot arm moves by following smooth trajectory path.

In this stage, the inverse kinematics is utilized to convert the points of trajectory path to corresponding joint's variables that would be used to move the two-link arm from initial (start) to target (goal) points.

4. SIMULATE RESULTS

4.1 INTRODUCTION

A developed simulation of Robot Arm Path Planning Application is proposed in this chapter that is tailored specifically to find an optimal path and move the robot arm with 2-DOF in the 2-D static environment.

The application simulates the implementation of the three proposed approaches on a variant complexity of 2-D static robot arm environments.

The simulation improvement approaches are implemented and tested in MATLAB program 2019b.

The proposed methods are tested with 2-D various difficulties suggested maps that have a limit from (-50 to 50) cm for both x-y dimensions and contain various shapes of static obstacles.

The specification of the first suggested environment is that composed of a variety of obstacles and equaled length of arm links. Whereas the suggested length of the robot links for both link1 and link2 is equal to 25cm and the suggested joint limit is $0 \leq \theta_1 \leq 360$ and$-180 \leq \theta_2 \leq 180$.

The specification of the second suggested environment is that composed of a variety of obstacles and different lengths of arm links. Whereas the suggested length of the first robot link is equal to 20cm while the suggested length of the second robot link is equal to 30cm and the suggested joint limit is $0 \leq \theta_1 \leq 360$ and$-180 \leq \theta_2 \leq 180$.

The specification of the third suggested environment is that composed of a variety of obstacles and different lengths of arm links. Whereas the suggested length of the first robot link is equal to 35cm while the suggested length of the first robot link is equal to 15cm and the suggested joint limit is $0 \leq \theta_1 \leq 360$ and$-180 \leq \theta_2 \leq 180$.

The specification of the fourth suggested environment is that composed of a variety of obstacles and equal lengths of arm links, but the joints limit of the robot arm is restricted with the range $0 \leq \theta_1 \leq 180$ and$-90 \leq \theta_2 \leq 90$.

In the first run of the proposed method with the first suggested environment, where the robot has to move in elbow down solution from start configuration (28, 39) to the goal configuration (15, -42) with the optimal path.

Figures (4.1-4.3) illustrate the generated path from proposed method of PSO algorithm and trajectory path that has been obtained from applying cubic polynomial equations, change of parameters with the interval time 10s of run 1 and execution time of run 1 respectively.

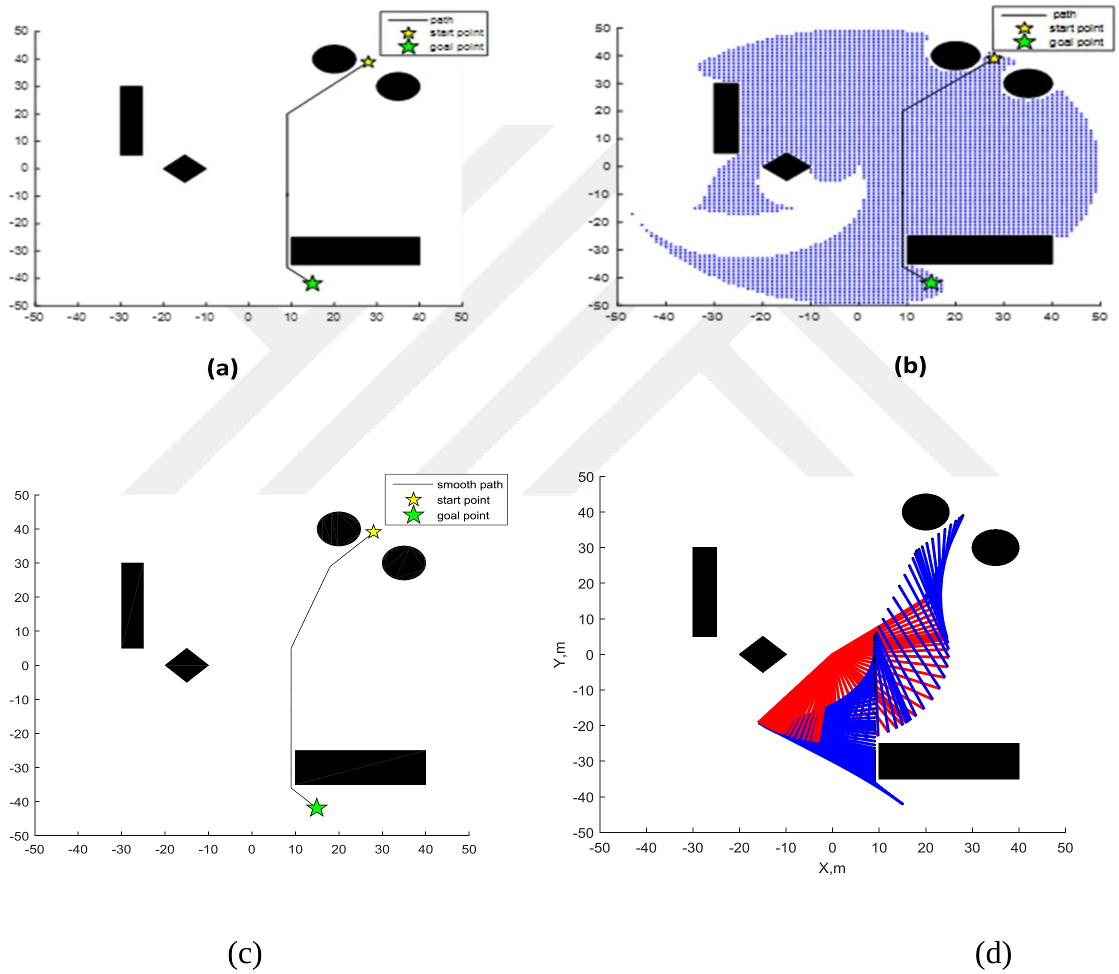
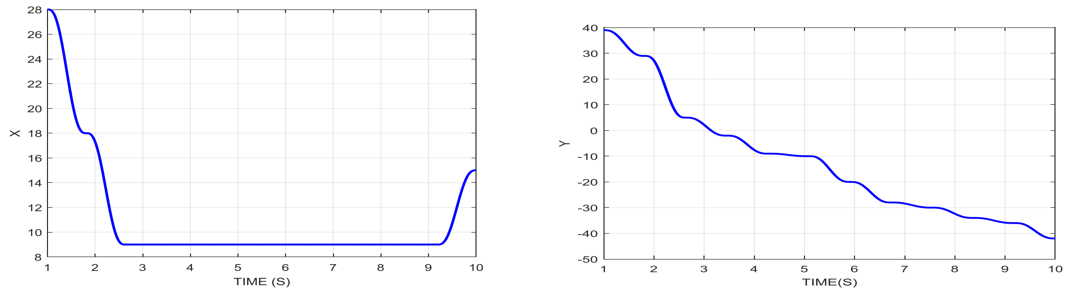
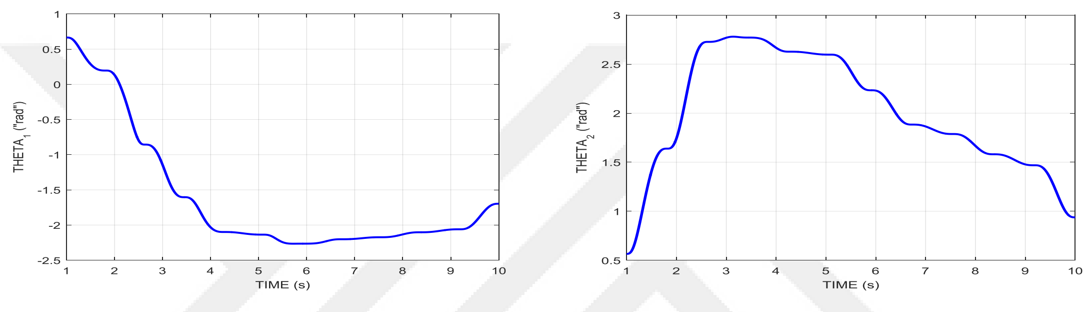


Figure 4.1: Test Run of Modified D* Based PSO Method in the First Environment, (a) Generated PSO Path, (b) the Generated Path within Elbow Down Space, (c) the Smooth Trajectory, (d) the Two-link Robot Arm Follows the Path.



(a)



(b)

Figure 4.2: Change of Parameters with the Interval Time of Run in First Environment, (A) Change of X and Y, (B) Change of $\theta_1$and θ_2 .

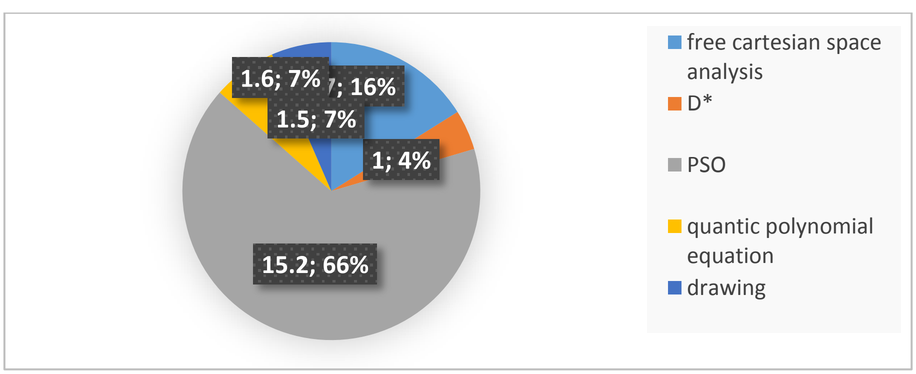


Figure 4.3: Execution Time of Run of Proposed D* Based PSO in First Environment.

In the second run of the proposed method with the first suggested environment. Where the robot has to move in elbow up solution from start configuration (30, 38) to the goal configuration (-36, 13) with the optimal path. Moreover, the suggested length of the robot links for both link1 and link2 is equal to 25cm and the suggested joint limit is $0 \leq \theta_1 \leq 360$ and $0 \leq \theta_2 \leq 360$.

The figure (4.4) illustrate the generated path from proposed method of PSO algorithm and trajectory path that has been obtained from applying cubic polynomial equations.

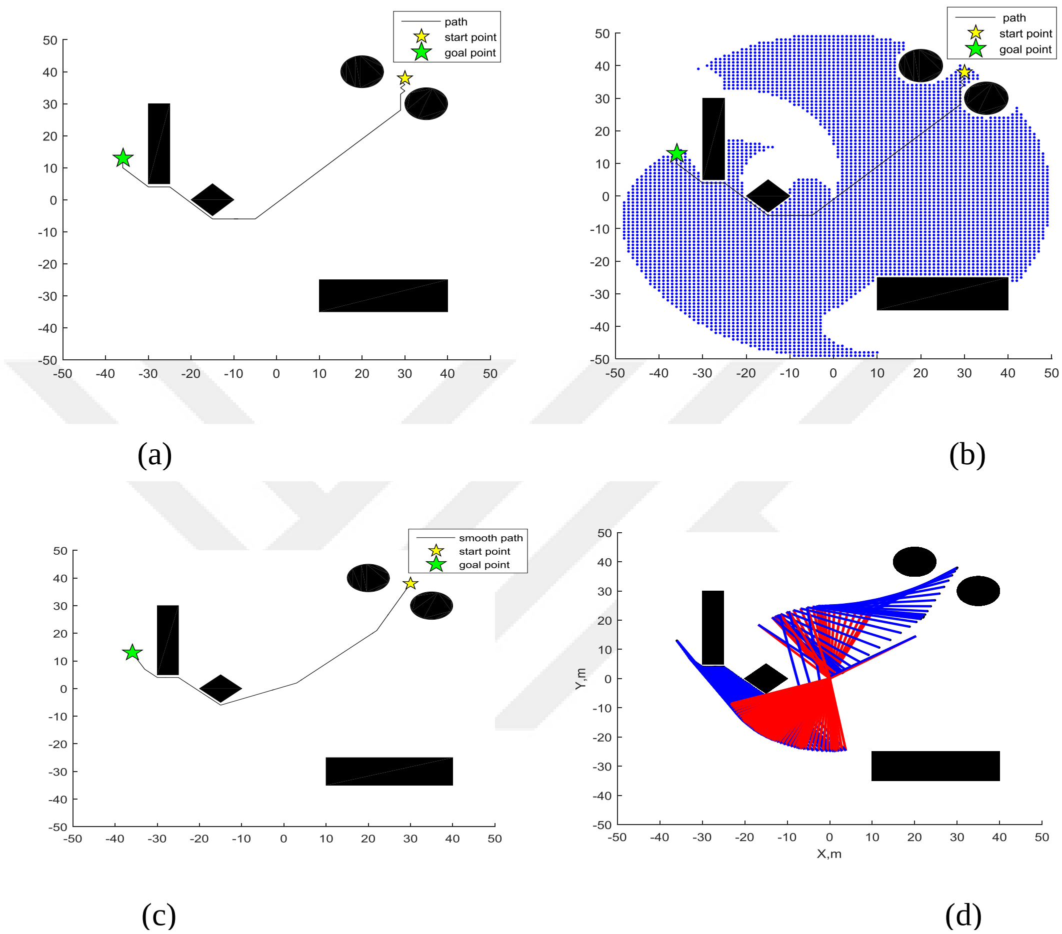


Figure 4.4: Two-link robot arm environment of run 1, (a) generated PSO path, (b) the generated path within elbow up space, (c) the smooth trajectory, (d) the Two-link robot arm follows the path.

Modeling of the proposed method shows influence and role the coefficients R_1 and R_2 in guiding the selection of a next point by PSO algorithm. Furthermore, the cubic polynomial equations have a significant role in smoothing the produced oscillation and as shown in the figure (4.5). Moreover, the four obstacles were successfully avoided.

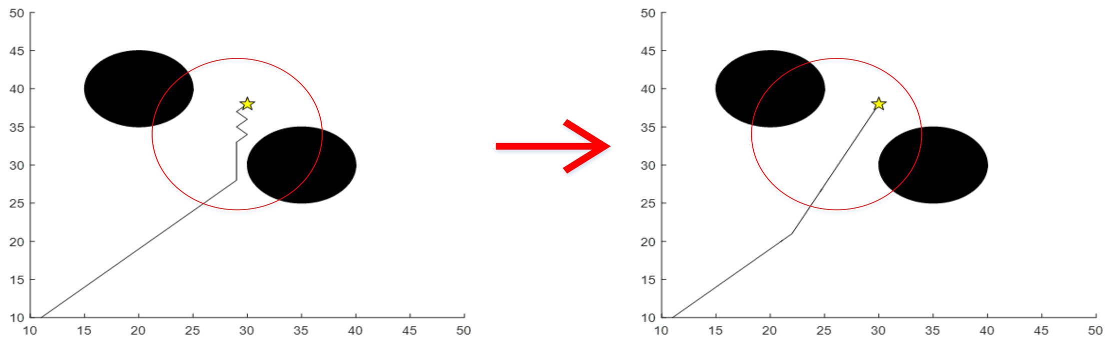
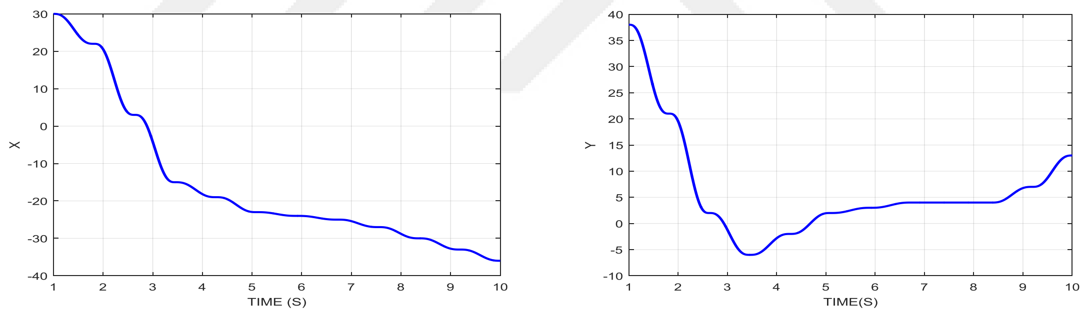
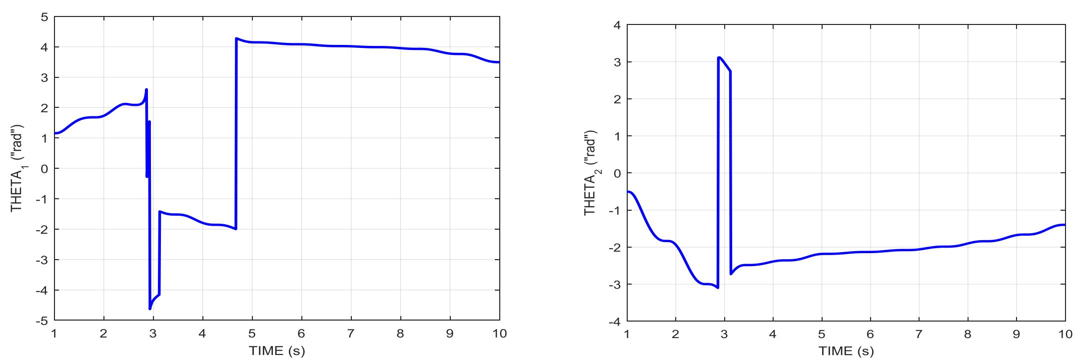


Figure 4.5: influence of coefficients R_1 and R_2 in guiding the selection of a next point and role of quintic polynomial equations**.**

The figures (4.6-4.7) demonstrate the change of parameters with the interval time of first run of the first environment. The result of changing θ_1 and θ_2 of the modeling illustrate the problem of high joint rates near the singularity, which is occupied at the middle points of the trajectory where the robot arm has to follow it, and the execution time of second run of first environment.



(a)



(b)

Figure 4.6: change of parameters with the interval time of run 2 of second environment, (a) change of x and y, (b) change of $\theta_1$andθ_2.

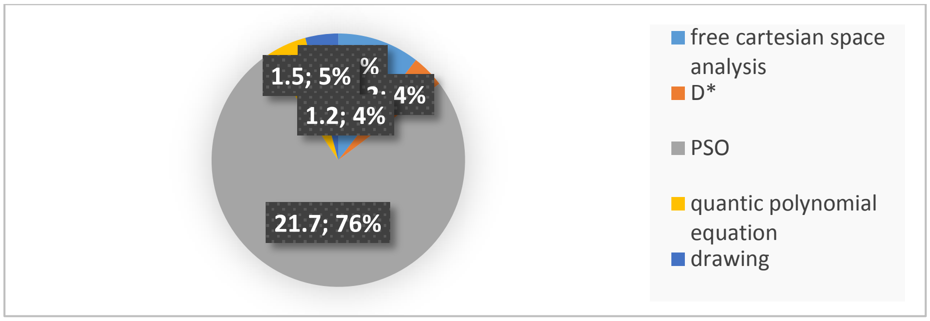


Figure 4.7: Execution Time of Second Run of First Environment of Proposed Based PSO.

5. EXPERIMENTAL IMPLEMENTATION

5.1 INTRODUCTION

This chapter describes the experimental work of the path construction for modified version of Lynx motion AL5D robot arm and focuses on the proposed method of the optimization path using PSO algorithm and Cubic parametric equations. The collision checking and best neighbor selection are the objective functions used to avoid obstacles and select a minimum distance node respectively. Moreover, construction of the free Cartesian space is considered as an essential process to guarantee a free collision path planning. The proposed optimization algorithms are capable of finding a shortest free collision path that moves the Two-link robot arm from the start to the goal configurations and can be considered as the off-line path planning methods. However, the off-line path planning has complete knowledge of the generated path before the Two-link robot arm starts moving.

5.2 EXPERIMENTAL IMPLEMENTATION

The aim of the experimental part in this work is to develop a practical experimental physical setup for optimal path validation using and optimal path planning methods for the Two-link robot arm with 2-DOF in the known static environment. During the experiment, the movement of the Two-link robot arm from the initial point to the goal point should be absolutely known in advance. In this regard, the proposed methods are to globally off-line find an optimal path planning and trajectory generation using swarm optimization techniques and trajectory generating equations as well as to get the corresponding coordinates. The coordinates of generated trajectory have been converted into the joints variables using the inverse kinematics, where these joints variables are converted into Pulse Width Modulation (PWM) and transmitted to each servo motors that attached to the link of the robot arm.

The main concept of the experimental work depends on path planning, trajectory generation, computing the joints variables of the Two-link robot arm and converting them into PWM using MATLAB program. The serial servo controller (SSC-32) is used to drive each motor of the Two-link robot arm. However, the SSC-32 is connected to the PC via serial port to transfer the PWMs from MATLAB program to the servo motor. Figures (5.1 and 5.2) show the modified version of Lynx motion of AL5D robot arm and the full experimental work respectively. Dimensions of AL5D robot arm listed in Appendix A.

Figure 5.1: The Modified AL5D Robot Arm.

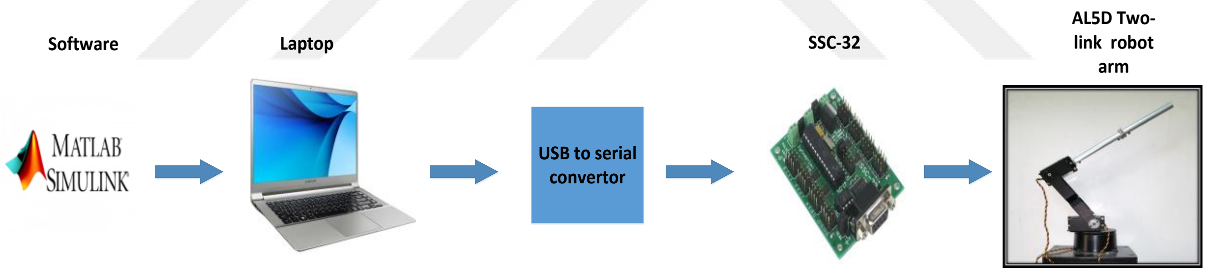


Figure 5.2: Full Experimental Work.

5.2.1 Overall System

Practically, the movement of a Two-link robot arm is directly based on the signals or torque control commands given to the servo motors. However, the output variable of the suggested systems is a direct motor command signal (τ_j) to every robot arm link. The motor command has a bipolar-range (+/-) to drive the robot link in positive or negative changing steps.

Moreover, the joint angle is converted into the motor command signal (τ_j) which is sent to the joint motor of the corresponding AL5A robotic link in order to move it. All the input and output variables take positive and negative values, therefore provide information about the magnitude as well as the sign which related to every relative left or right link, where j=1,2.

The part of the motor control commands is generated depending on the values of parameters (joint angle) and these parameters can be produced by the system of arm movements planning.

This arises the need for changing the structure of this system to adapt it for use with the experimental robot arm. The aim is to add a new unit that generates control commands to the servo motors. The following equations are derived from the actual hardware system:

$$\tau_1 = 0.0027778 * {\theta_1}^2 + 7.9167 * \theta_1 + 650 \qquad (5.1)$$

$$\tau_2 = 0.000621728 * {\theta_2}^2 - 9.2222 * \theta_2 + 1625 \qquad (5.2)$$

Equation (5.1 and 5.2) are linear equations that convert the input (angle) into (PWM), which is τ_j.

5.2.2 Dimensions and Specs of AL5D Robot Arm

Table 5.1: dimensions and specs.

Name of parameter	Constant value
The length of the arm from the first motor to the other. From Shoulder to Elbow (L1), cm.	9.5
The length of the arm from the second motor to the third. From Elbow to Wrist (L2), cm.	10.5
The length of the arm from third engine to the tip of the arm. From Wrist to the top (L3), cm.	8.5
Height (arm parked): approx., cm.	15.24
Height (reaching up): approx., cm.	35.56
Median forward reach: approx., cm.	1A.605
Gripper opening, cm.	3.175
Alternate gripper opening, cm.	A.7625
Lift weight (arm extended): approx., g.	113.39808
Weight, g.	595.33992
Range of motion per axis, degree.	180°

Figure 5.3: Shows robot manipulator details.

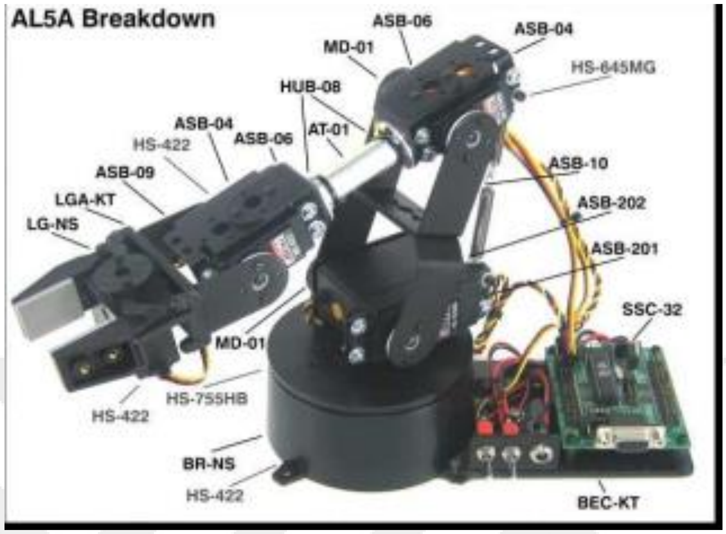


Figure 5.3: Robot manipulator details.

II) Electronic circuit description of Lynx motion SSC-32

SSC-32 Hardware Information: Figure (5.4) shows SSC-32.

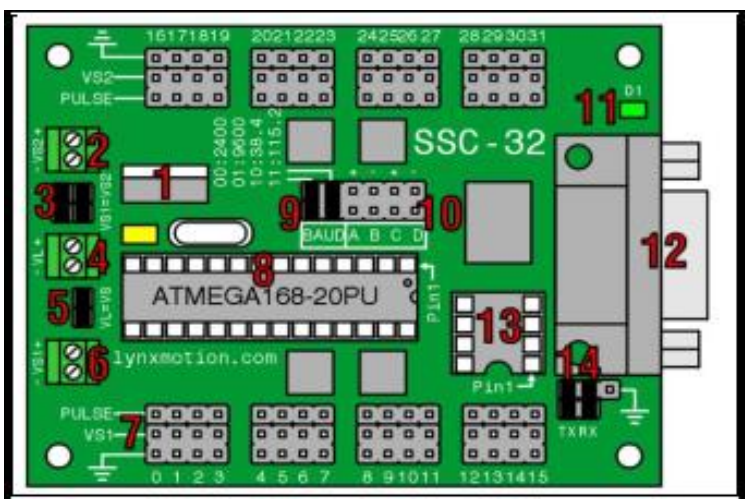


Figure 5.4: SSC-32 [76].

i. The Low Dropout regulator will provide 5vdc out with as little as 5.5vdc coming in.

ii. This terminal connects power to servo channels 16 through 31. Apply A.8vdc to 6.0vdc for most analog or digital servos.

iii. These jumpers are used to connect VS1 to VS2.

iv. This is the Logic Voltage, or VL.

v. This jumper allows powering the microcontroller and support circuitry from the servo power supply.

vi. This terminal connects power to servo channels 16 through 31. Apply A.8vdc to 6.0vdc for most analog or digital servos.

vii. This is where the servos or other output devices, is connected.

viii. This is where the Atmel IC chip goes.

ix. The two BAUD inputs allow configuring the baud rate.

x. The ABCD inputs have both static and latching support.

xi. This is the Processor Good LED.

xii. Simply plug.

xiii. This is an 8-pin EEPROM socket. The EEPROM is supported by the 2.01GP firmware.

xiv. This is the TTL serial port or DB9 serial port enable.

5.3 EXPERIMENTAL IMPLEMENTATION OF OPTIMAL PATH PLANNING

The proposed method of PSO algorithm is implemented to find an optimal path that drives the experimental robot arm from the start configuration to the goal configuration without collision. Moreover, the proposed method is tested on 2-D suggested environment for illustrating the operation of the proposed method and demonstrating detailed results for finding the optimal trajectory.

Figure (5.5) demonstrates the corresponding simulated results of tested the particle real experimental robot arm by the proposed method of D* based PSO algorithm from start configuration (32,30) with the joints angles (θ_1=24.8, θ_2 =27.4) to the goal configuration (-42, -10) with the joints angles (θ_1=170, θ_2 =34.8) and trajectory cost equal to 100.6332.

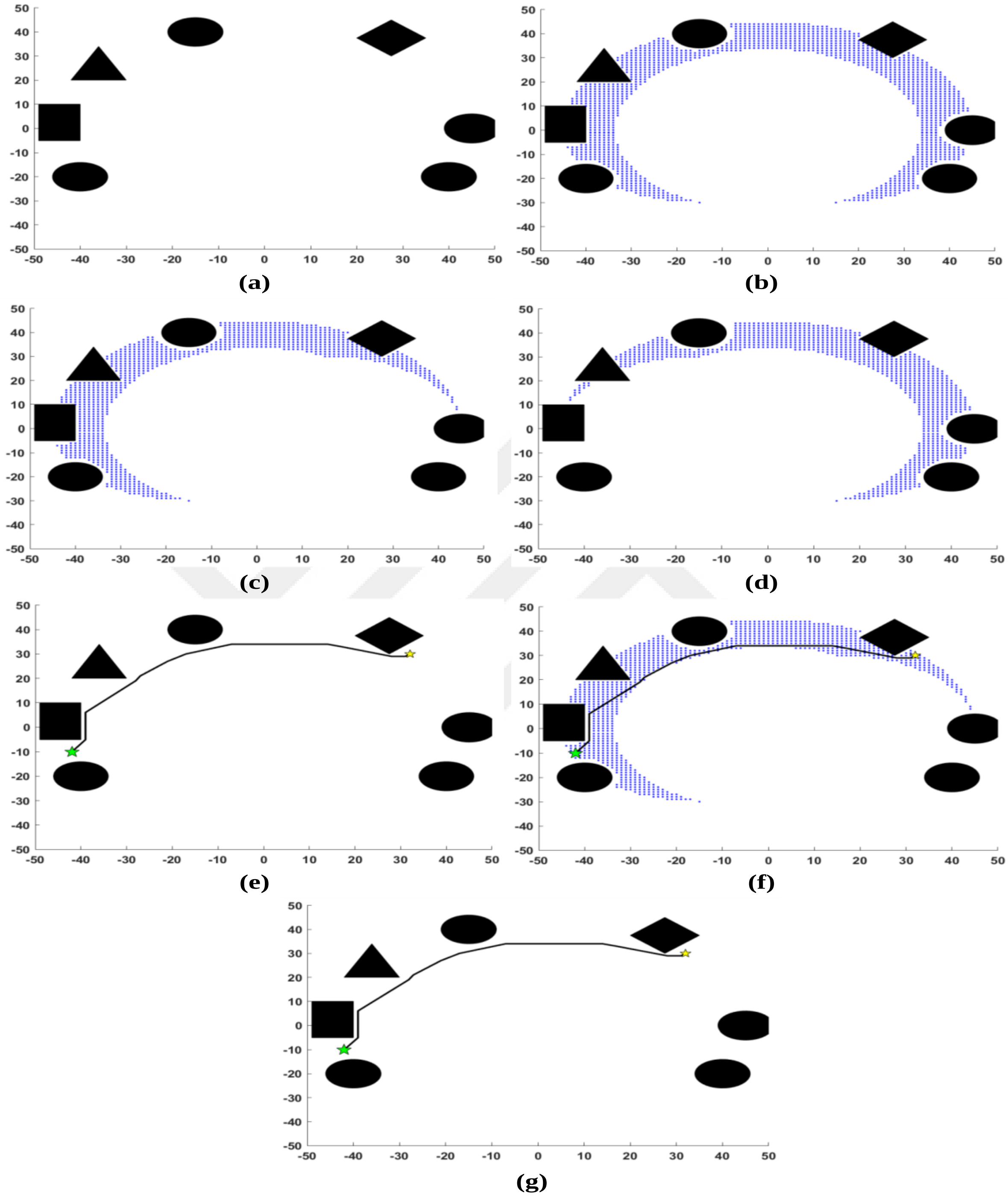


Figure 5.5: Experimental Robot Arm Analysis, (a) Suggested Environment, (b) Free Cartesian Space, (c) Free Elbow Down Space, (d) Free Up Down Space, (e) Generated D* Based PSO Path, (f) the Generated Path within Elbow Down Space, (g) the Smoothing Trajectory.

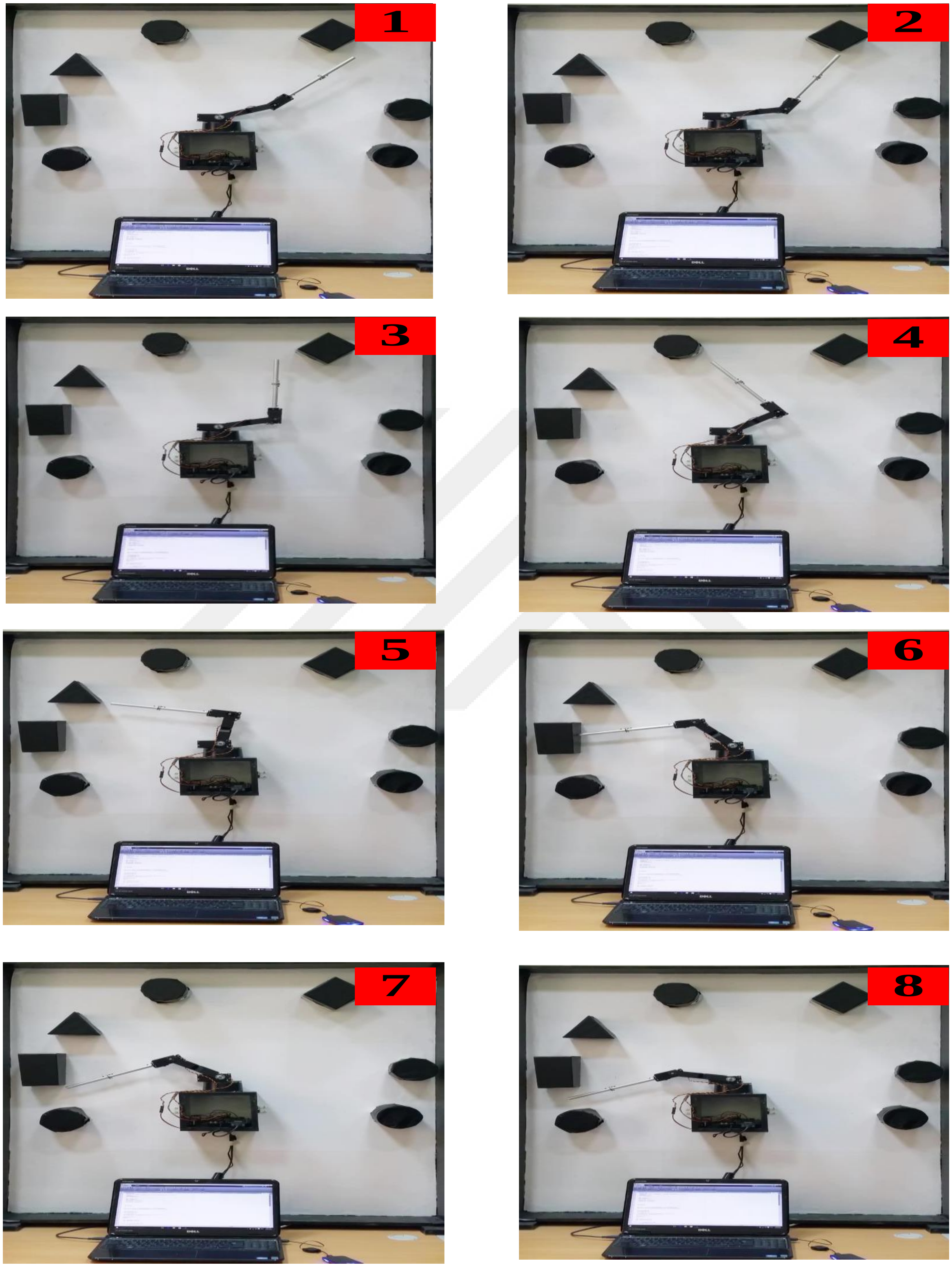


Figure 5.6: Result of Particle Real Experimental Robot Arm Sequences Movements.

Figure (5.7) shows the results carried out in practical and known static environment.

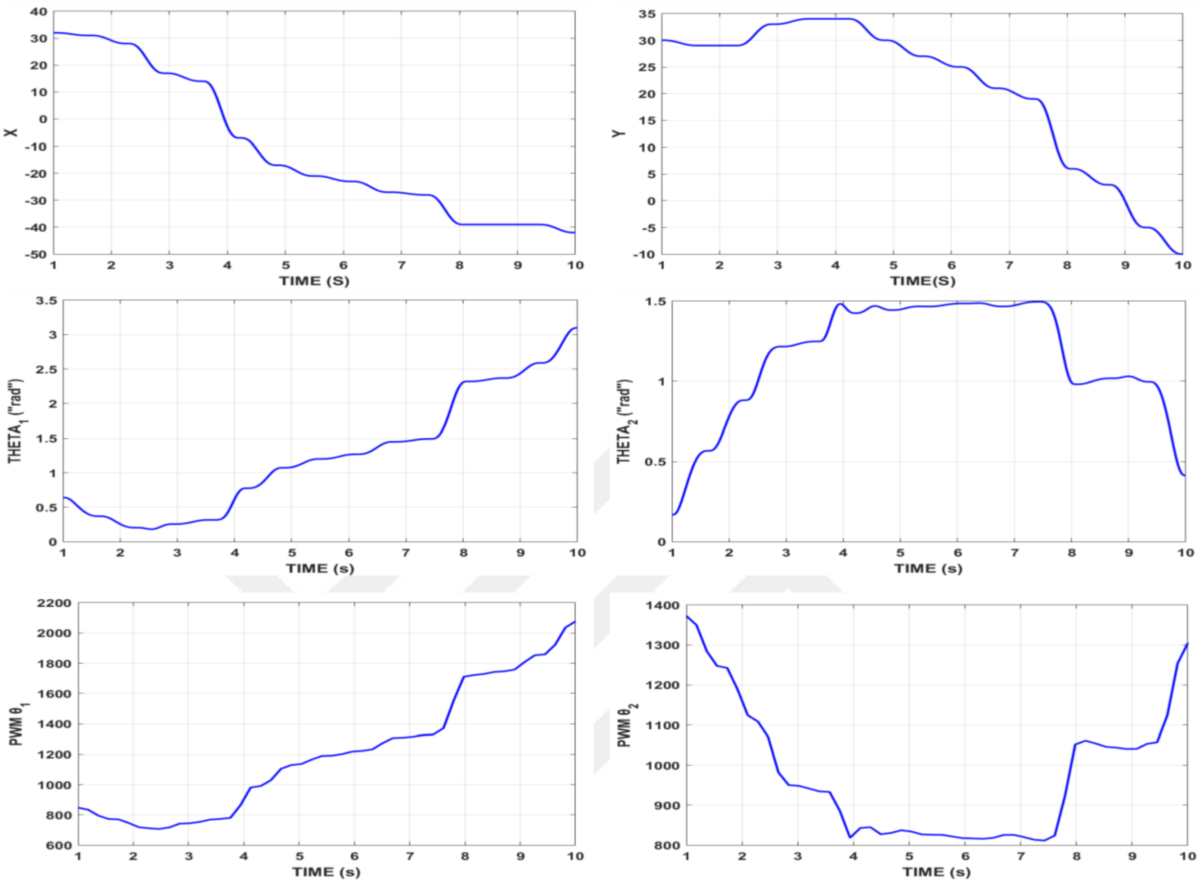


Figure 5.7: The Results of Practical Movements.

Consequently, the proposed method of PSO algorithm provides perfect results since the robot has moved following the shortest trajectory by avoiding the obstacles and reached the goal configuration with the relatively very small error in terms of path planning.

6. CONCLUSION AND FUTURE WORKS

6.1 CONCLUSIONS

a) The hybridization method of PSO and cubic polynomial equations shows clearly an enhancement in the smoothness of the generated trajectory whose intermediate via-points are randomly selected from the planned shortest path.

b) The simulation results of the hybridization method of PSO and cubic parametric equations have illustrated the selection influences of the intermediate via-points number on the execution time and number of iteration. However, in the case where a small number of intermediate via points is selected, the rapid generation of the trajectories with less iteration number was demonstrated.

c) In the experimental work, the implementation results of three proposed methods were applied practically on a modified version for Lynx motion AL5D robot arm for path planning with some errors that can be attributed to the servo-motor control resolution.

d) The robot arm successfully followed the shortest path and avoided the obstacles three proposed methods in the off-line modelling and practical results.

6.2 SUGGESTIONS FOR FUTURE WORK

The suggestions for future work can be summarized on the basis of the following ideas:

a) Design of path planning based on dynamic constraints depending on proposed PSO method for on-line locally path planning.

b) Enhancement of the hybridization method of PSO and cubic parametric equations by developing the selection criteria of intermediate via points depend on intelligent strategy.

c) Further future work is to maximize the robot arm flexibility by increasing the number of robot arm degree of freedom.

REFRENCE

[1] Kelly R., Santibanez V. and Loria A., "Control of Robot Manipulators in Joint Space", Springer-Berlin, Heidelberg, New York, Hong Kong, London, Milan, Paris, Tokyo, pp. 3-4, 2005.

[2] L-W. Tsai. "Robot Analysis: The Mechanics of Serial and Parallel Manipulators". NY, John Wiley & Sons, Inc, 1999.

[3] Spong M. W., Seth Hutchinson, and Vidyasagar M., "Robot Modeling and Control", 1st Edition. John Wiley & Sons, Inc. New York / Chichester / Weinheim / Brisbane / Singapore / Toronto. pp. 5-6, 76, 2006.

[4] W. W. Melek. "ME 547: Robot Manipulators: Kinematics, Dynamics, and Control". Waterloo, University of Waterloo, 2010.

[5] Kermiche S., Larbi S. M., and Abbassi H. A., " Fuzzy Logic Control of Robot Manipulator in the Presence of Fixed Obstacle", The International Arab Journal of Information Technology, Vol. 4, No.1, pp. 26-32, January 2007.

[6] Amer A. F. , Sallam E. A. , and Elawady W. M. , " Adaptive Fuzzy Sliding Mode Control using Supervisory Fuzzy Control for 3 DOF Planar Robot Manipulators", Robotics and Autonomous Systems, Vol. 59, Issues 7-8, pp. 543-554, , July-August, 2011.

[7] Abo-Hammour Z. S., Mirza N. M., Mirza S. M., and Arif M., "Cartesian Path Generation of Robot Manipulators using Continuous Genetic Algorithms", Robotics and Autonomous Systems Vol. 41, No. 4, pp. 179- 223,2002.

[8] Mechatronics: Applications. In: Polytechnic Institute of NYU 2010 http:// mechatronics.poly.edu/ Accessed on 2/8/2016.

[9] https://en.wikibooks.org/wiki/Robotics/Types_of_Robots/Arms accessed on 2/8/2016.

[10] S. H. Tang, W. Khaksar, N. B. Ismail and M. K. A. Ariffin, " A Review on Robot Motion Planning Approaches", (2012).

[11] Sam Cubero," Industrial Robotics Theory, Modelling and Control", Advanced Robotic Systems International, 2007.

[12] Sangita S.," Optimization of Robot Motion Planning using Ant Colony Optimization", Department of Mechanical Engineering National Institute of Technology Rourkela, Orissa, India,2011.

[13] Ahmed Elshamli," Mobile Robots Path Planning Optimization in Static and Dynamic Environments", University of Guelph, 2004.

[14] Nadia S., Lana J.," Optimal Path Planning for Intelligent Mobile Robot Navigation Using Modified Particle Swarm Optimization", International Journal of Engineering and Advanced Technology (IJEAT) ISSN: 2249 – 8958, Volume-2, Issue-4, April 2013.

[15] Hwang, Y.K., Ahuja, N., "Gross Motion Planning – A Survey", ACM Computing Surveys, volume 24, issue 3, pp. 219-291, September 1992.

[16] Buniyamin N., Wan W.A.J., Sariff N., Mohamad Z., " A Simple Local Path Planning Algorithm for Autonomous Mobile Robots ", International Journal Of Systems Applications, Engineering & Development Issue 2, Volume 5, 2011.

[17] B. Browning, D. Polani, A. Bonarini, and K. Yoshida, "Local Multiresolution Path Planning", Robot Soccer World Cup VII, pp. 332-343, Springer, 2004.

[18] Buniyamin N., Sariff N., Wan Ngah W.A.J., Mohamad Z. , "Robot global path planning overview and a variation of ant colony system algorithm" , INTERNATIONAL JOURNAL OF MATHEMATICS AND COMPUTERS IN SIMULATION, Issue 1, Volume 5, 2011.

[19] Giuseppe C., Fernando G., "Motion and Operation Planning of Robotic Systems", Springer, Vol. 29, 2015.

[20] Fabrizio Padula1 and Veronique Perdereau2, "An On-Line Path Planner for Industrial Manipulators", International Journal of Advanced Robotic Systems, 2013.

[21] ATA, A. A.(2007),"Optimal Trajectory Planning of Manipulators",Journal of Engineering Science and Technology, Vol. 2.

[22] Steven M. LaValle,"PLANNING ALGORITHMS", University of Illinois,2006.

[23] John F. Canny ," Complexity of Robot Motion Planning", MIT Press, 1988.

[24] Serdar K., "Serial and Parallel Robot Manipulators – Kinematics, Dynamics, Control and Optimization", Janeza Trdine, Rijeka, Croatia, 2012.

[25] Akash D., R. B. Mishra, A. K. Jha ," Task Time Optimization of a Robot Manipulator using Artificial Neural Network and Genetic Algorithm", International Journal of Computer Applications (0975 – 8887) Volume 51– No.13, August 2012.

[26] Devendra P. Garg, Manish Kumar, "Optimization techniques applied to multiple manipulators for path planning and torque minimization", Engineering Applications of Artificial Intelligence, 2002.

[27] M. Stilman, "Global manipulation planning in robot joint space with task constraints," Robotics, IEEE Transactions on, vol. 26, no. 3, pp. 576–584, 2010.

[28] Cruz-Bernal," Meta-Heuristic Optimization Techniques and Its Applications in Robotics", Intech, 2013.

[29]Jorge Stephen J. Wright, "Numerical Optimization", Springer-Verlag New York, 1999.

[30]M. Kalakrishnan, S. Chitta, E. Theodorou, P. Pastor, and S. Schaal, "STOMP: Stochastic trajectory optimization for motion planning," in Proceedings of IEEE International Conference on Robotics and Automation, pp. 4569–4574, 2011.

[31] Marco Dorigo, Gianni Di Caro and Luca M. Gambardella, "Ant Algorithms for Discrete Optimization", Artificial Life Volume 5, Number 2, 1999.

[32] Michael Brand, Michael Masuda, Nicole Wehner and Xiao-Hua Yu, "Ant Colony Optimization Algorithm for Robot Path Planning", IEEE International Conference on Computer Design and Applications (ICCDA 2010), 2010.

[33] J. Kennedy and R. Medes, "Population Structures and Particle Swarm Performance", In Proceedings of the IEEE Congress on Evolutionary Computation, Hawaii, USA, 2002.

[34] Zargar, A. N. & Javadi, H,"Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target", Third UKSim European Symposium on Computer Modeling and Simulation, Vol. , No., pp.60-65,2009.

[35] S. G. Yue, Henrich, "Trajectory Planning In Joint Space For Flexible Robot With Kinematics Redundancy", IASTED International Conference, Austria, 2001.

[36] Frank Lingelbach," Path Planning For Mobile Manipulation Using Probabilistic Cell Decomposition", IEEE/RSL International Conference On Intelligent Robots And Systems, Sendai, Japan, 2004.

[37] R. E. Meligy, A. M. Bassiuny, E. M. Bakr and A. A. Tantawy, "A Feasible Minimum-Time Trajectory Of Robot Manipulator", Proceedings Of The 9th International Symposium On Mechatronics And Its Applications (Isma13), Jordan, 2013.

[38] Babak R., Javad M., Hasan K., Gholam D. And Ali O.," Robot Manipulator Path Planning Based On Intelligent Multiresolution Potential Field", International Journal of U- And E-Service, Science and Technology, Vol.8, No.1, Pp.11-26, 2015.

[39] Mohd Murtadha Mohamad, Nicholas K. Taylor, and Matthew W. Dunnigan," Articulated Robot Motion Planning Using Ant Colony Optimisation", 3rd International IEEE Conference Intelligent Systems, September 2006.

[40] WANG J., GUO M., LI L., SUN S., GU S.," Collision-free Path Planning of Dual-arm Robots Based on Improved Ant Colony Algorithm", IEEE, 2009.

[41] Zhang H., Li L.," An Improved Ant Colony for Servo Mechanical Arm Path Planning",International Conference on Intelligent Systems Research and Mechatronics Engineering (ISRME), 2015.

[42] Dongsheng Z., Lan Wang and Qiang Zhang," Obstacle avoidance planning of space manipulator end-eff ector based on improved ant colony algorithm ", Springer Plus, 2016.

[43] Wenfu Xu, Cheng Li, Bin Liang, Yu Liu and Yangsheng Xu," The Cartesian Path Planning of Free-floating Space Robot using Particle Swarm Optimization ", International Journal of Advanced Robotic Systems, Vol. 5, No. 3 , 2008.

[44] Hsu-Chih Huang and Ching-Chih Tsai, "Global Path Planning for Autonomous Robot Navigation Using Hybrid Metaheuristic GA-PSO Algorithm", SICE Annual Conference, 2011.

[45] Chao and Liu," Modelling And Optimization Approach For Trajectory Planning of Three Freedom Planar Manipulators", Journal of Pure and Applied Mathematics: Vol. 7, No.1, pp. 1-20, 2012.

[46] Rokbani and Alimi,"Inverse Kinematics Using Particle Swarm Optimization, A Statistical Analysis", Procedia Engineering, Vol. 64, 2013.

[47] Yang Yueqiang, Tang Zhiwei,"The Study of Path Planning of Welding Manipulator Based on Improved QPSO ", Sixth International Conference on Measuring Technology and Mechatronics Automation, 2014.

[48] Jeong-Jung Kim," Trajectory Optimization with Particle Swarm Optimization for Manipulator Motion Planning", IEEE Transactions on Industrial Informatics, 2015.

[49] Craig, J. J., Introduction to Robotics, Addison-Wesley, 1989.

[50] Wesam Mohammed," Solution of Inverse Kinematics for SCARA Manipulator Using Adaptive Neuro-Fuzzy Network", IJSC, Vol.2, No.4, 2011.

[51] Richard M., Zexiang Li and S. Shankar, "Mathematical Introduction to Robotic Manipulation", 1994.

[52] M. O’Malley, "Introduction to Robotics, Inverse Manipulator Kinematics", International Journal on Soft Computing (IJSC) Vol.2, No.4, 2011.

[53] Carl D. III Crane. By Joseph Duffy (2009), "Kinematic Analysis of Robot Manipulators", Cambridge University Press.

[54] Serdar Kiiqiik Zafer Bingul, The Inverse Kinematics Solutions of Industrial Robot Manipulators, IEEE, 2004.

[55] Anthony Stentz, " The D* Algorithm for Real-Time Planning of Optimal Traverses", CML-RI-TR-94-37, The Robotics Institute, Carnegie Mellon University, Pittsburgh, Pennsylvania, 1994.

[56] Howie Choset, " Robotic Motion Planning: A* and D* Search", Lecture notes of Robotics Institute 16-735, Carnegie Mellon University; www.cs.cmu.edu/~motionplanning.

[57] Nosrati M., Karimi R. and Hasanvand H.," Investigation of the * (Star) Search Algorithms: Characteristics, Methods and Approaches", World Applied Programming, Vol. 2, No.4, 2012.

[58] M. Dorigo, luca M. G., " Ant Colony system: A Cooperative learning approach to the Travelling Salesman Problem", IEEE transaction on evolutionary computation, Vol. 1, No. 1, 1997.

[59] A. Colorni, M. Dorigo and V. Maniezzo," Distributed optimization by ant colonies", Proceedings of ECAL 91, European Conference on Artificial Life, Elsevier Publishing, Amsterdam,1991.

[60] Narendra P. and Sanjeev S.," Robot Path Planning using Swarm Intelligence: A Survey", International Journal of Computer Applications, Volume 83 – No 12, 2013.

[61] Song-Hiang Chia , Kuo-Lan Su, Jr-Hung Guo and Cheng-Yun Chung, "Ant Colony System Based Mobile Robot Path Planning", Fourth International Conference on Genetic and Evolutionary Computing, IEEE computer society, 2010.

[62] Marco Dorigo, Thomas Stutzle," Ant Colony Optimization", MIT Press Cambridge, Massachusetts, London, England, 2004.

[63] El-ghazali Talbi, " Metaheuristics: from design to implementation", John Wiley and Sons, 2009.

[64] Marco Dorigo, Mauro Birattari, and Thomas St¨utzle" Ant Colony Optimization Artificial Ants as a Computational Intelligence Technique", IRIDIA – TECHNICAL REPORT SERIES, 2006.

[65] Shi and R. Eberhart. "A Modified Particle Swarm Optimizer". In Proceedings of the IEEE International Conference on Evolutionary Computation, Piscataway, New Jersey, 1998.

[66] Shahab A., M. Usman Rafique, and M. Umer Khan," Mobile Robot Path Planning in Static Environments using Particle Swarm Optimization", International Journal of Computer Science and Electronics Engineering (IJCSEE), Vol. 3, 2015.

[67] Mahamed G. H. Omran, "Particle Swarm Optimization Methods for Pattern Recognition and Image Processing", PhD Thesis. Department of Computer Science, University of Pretoria, South Africa, 2004.

[68] Li, J. & Xiao, X," Multi- Swarm and Multi- Best particle swarm optimization algorithm", Intelligent Control and Automation, 2008.

[69] F. Van den Bergh "An Analysis of Particle Swarm Optimizers", PhD Thesis. Department of Computer Science, University of Pretoria, South Africa, 2002.

[70] J. Kennedy and R. Medes, "Population Structures and Particle Swarm Performance", In Proceedings of the IEEE Congress on Evolutionary Computation, Hawaii, USA, 2002.

[71] Sreenivasulu R., “Simulation of Desired End Point Trajectory for A 2-Dof Planar Manipulator”, International Journal of Advanced Scientific and Technical Research, Vol. 5: 688-696, 2012.

[72] Tangpattanakul P., Meesomboon A., and Artrit P., “Optimal Trajectory of Robot

Manipulator Using Harmony Search Algorithms, Recent Advances In Harmony Search Algorithm Studies in Computational Intelligence”, Volume 270: pp. 23-36, Springer-Verlag Berlin Heidelberg, 2010.

[73] Michael E. Mortenson," Geometric Modeling", John Wiledy, New York, 1996.