World Academy of Science, Engineering and Technology
International Journal of Aerospace and Mechanical Engineering
Vol:9, No:8, 2015

Three-Dimensional Off-Line Path Planning for Unmanned Aerial Vehicle Using Modified Particle Swarm Optimization

Lana Dalawr Jalal

***Abstract*—**This paper addresses the problem of offline path planning for Unmanned Aerial Vehicles (*UAVs*) in complex three-dimensional environment with obstacles, which is modelled by *3D* Cartesian grid system. Path planning for *UAVs* require the computational intelligence methods to move aerial vehicles along the flight path effectively to target while avoiding obstacles. In this paper Modified Particle Swarm Optimization (*MPSO*) algorithm is applied to generate the optimal collision free *3D* flight path for *UAV*. The simulations results clearly demonstrate effectiveness of the proposed algorithm in guiding *UAV* to the final destination by providing optimal feasible path quickly and effectively.

***Keywords*—**Obstacle Avoidance, Particle Swarm Optimization, Three-Dimensional Path Planning Unmanned Aerial Vehicles.

I. Introduction

PATH planning is one of the most important tasks in intelligent control of an autonomous mobile robot. Path planning is generating a collision-free feasible path between the initial location and the final destination in known or unknown environments with obstacles, and optimizes it with respect to constraint conditions [1]. An algorithm for path planning is said to be off-line if the environment is known and it is said to be on-line if it is capable of modifying a path in response to environmental changes [2].

Path planning using two-dimensional (*2D*) space is widely studied in ground robotics. However, when dealing with Unmanned Aerial Vehicles (*UAVs*), altitude has to be added to the planar movement for maneuvers in space [3]. *UAVs* fly in three-dimensional (*3D*) space, and so path planning needs to be able to produce flyable paths in three dimensions. A path planning algorithm produces one or more flyable paths for the *UAVs*. The path has to be of a specified (usually minimal) length, and, as the *UAV* has limited range, the time spent surveying specific areas should be minimized [4].

The obstacle avoidance problem is closely associated with path planning because the presence of obstacles usually results in the re-planning of paths. Path planning with obstacle avoidance in *3D* space is more complicated. The complication arises because there are infinitely more directions for maneuvers for both the *UAV* and the obstacle. Also, other constraints, especially generating the shortest path length, are more complicated in *3D* space [4].

In path planning problem for high-dimension space, classic methods proved to be inefficient, requiring considerably long time and huge storage memory. Consequently, heuristic methods are developed to solve path planning problem. There are several methods have been applied to path planning problems for mobile object working in *3D* space containing fixed obstacles. Computational intelligence methods, such as Neural Networks, Fuzzy Logic and Evolutionary Algorithms have been successfully used to derive trajectories for guiding mobile robots in known, unknown or partially known environments [5]. Path planning algorithms are required to be coded in software that runs on a processor carried on-board the *UAVs*.

Several researches investigated path planning problem searching for a collision-free shortest flight path connecting the starting and ending points. Reference [1] proposed a genetic algorithm (*GA*) for path planning of an autonomous mobile robot in the *3D* space which is modelled with a grid structure. Reference [6] presented a *3D* off-line path planner for *UAVs* using a multi-objective evolutionary algorithm. Reference [7] reported an Evolutionary Algorithm-based off-line path planner for *UAVs* to calculate a curved path line in a known *3D* space. “There are only a few references to work done on path planning in three dimensions” [4].

The objective of this study is to determine the shortest *3D* flight path for *UAV* to move from a starting to a target position. To resolve the path planning problem for *UAV* in a *3D* environment, the *MPSO* algorithm is proposed. Simulations are provided to validate the effectiveness of the proposed *MPSO*, and path planning for aerial vehicle in *3D* working space with the presence of obstacles is discussed.

In my previous study [8] path planning problem is investigated for mobile robot in *2D* environment occupied by fixed obstacles. The main contribution of this study is the extension of the path planning problem for mobile robot in *2D* to path planning problem for *UAVs* work in a complex *3D* space. Another contribution of this work is to develop and extending *MPSO* algorithm from *2D* to *3D* path planning problem. The extension of *2D* navigation into *3D*, made it possible to apply for *UAV*.

II. Path Planning Constraints

Producing a path between the start and target point is straightforward in the absence of any constraints. In practice, there are various constraints involved in path planning. The two most important constraints for path planning of a *UAV* are that the path must be flyable and safe. Flyable paths meet

Lana. D. Jalal is with the Electrical Department, Engineering Faculty, University of Sulaimani, Sulaimani, Iraq (e-mail: lana.jalal@ univsul.edu.iq).

kinematic or motion constraints and dictate the maneuverability of the *UAVs*. The safety of the *UAV* is achieved by avoiding obstacles that intersect the path. The scenario is shown in Fig. 1 for obstacle avoidance in *3D* space, a simple case of flying through an urban area containing buildings is considered. Each building is square in shape, and the size of each building is different. The buildings are overlapping in this example because the locations of the buildings are generated randomly. The heights and areas of the buildings are also generated randomly [4].

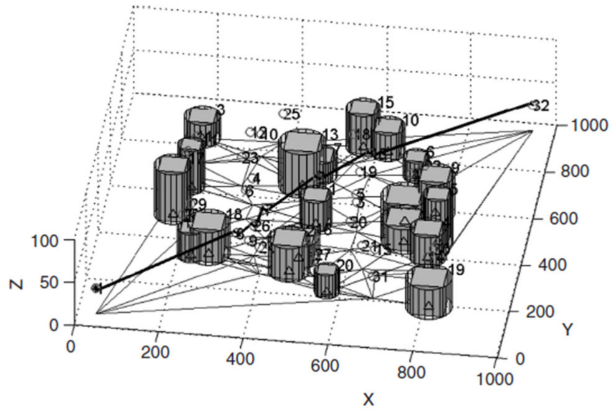

Fig. 1 Obstacle avoidance in *3D* space [4]

III. Path Planning Formulation

The primary aim of path planning is to provide structured mobility, that is, to facilitate moving or flying *UAV* from one location to another. Generally, there will be several location or waypoint on a known environment to visit before reaching the final destination. An optimal path is formed by line segments which are connecting the waypoint falling in working environment [4]. An optimal path *3D* flight path is formed by move the *UAV* from a starting to a target position through the waypoints in working environment.

IV. Modified Particle Swarm Optimization (*MPSO*)

Particle Swarm Optimization (*PSO*) was originally designed and introduced by Eberhart and Kennedy [9]. The *PSO* algorithm is a population based search algorithm based on the simulation of the animal societies, such as birds, ants and bees. Each individual within the swarm is represented by a vector in multidimensional search space. This vector has also one assigned vector which determines the next movement of the particle and is called the velocity vector. The *PSO* algorithm also determines how to update the velocity of a particle. Each particle updates its velocity based on current velocity and the best position it has explored so far; and also based on the global best position explored by swarm. The *PSO* process then is iterated a fixed number of times or until a minimum error based on desired performance index is achieved [9].

Basically the modified particle swarm optimization (*MPSO*), just like the *PSO*, consists of a population of particles that collectively search in the search space for the global optimum. Improvements are mainly trying to address the problem of premature convergence associated with the standard *PSO*. These improvements usually try to solve this problem by increasing the diversity of solutions in the swarm. In the *MPSO* an error factor is modelled to ensure that the *PSO* converges. The *MPSO* try to address another problem which is population may include many infeasible paths which have undesirable effect on the performance of the algorithm. In the *MPSO*, the infeasible paths are not discarded but can be modified to be feasible path. For example, if the particle path falls with an obstacle boundary, it is relocated to a position outside the obstacle. The *MPSO* algorithm is initialized with a number of particles and then searches for optimal. And also, positions and velocities for each particle are randomly initialized. Afterward, the position of each particle is influenced by the best position visited by itself which is referred to as particle best "*pbest*", and the best position in the whole swarm which is referred to as a global best "*gbest*" [8].

To show the feasibility and efficiency of the proposed *MPSO* algorithm, two test environments are present. First, the *MPSO* algorithm is test with one column of obstacles (Fig. 2 (a)) and then it test with more complex example (Fig. 2 (b)). The cubic are represent the *3D* obstacles. An 'obstacle' will be defined as any object in the environment that the *UAV* has to avoid. These obstacles could be objects such a buildings, hills, forests, etc. that will intercept the normal flight path of the *UAV*. The simulation environment was established by defining a rectangular grid of cells, with each cell representing a distinct location.

Simulations were performed on a *PC* with *CPU* Intel Core2 Duo and 2 *GB* of *RAM*, and *MPSO* algorithm is coded as m-files in *MATLAB*™ software version *R*2012. The size of the test environments were set to be 25 ×25 ×25 grid and place obstacles in grid cells, and the start and destination cells of *UAV* were located at (2, 2, 2) and (23, 23, 23), respectively. Obstacles were simulated in *MATLAB* in the space represented by cubic and their locations were set as shown in Fig. 2.

The proposed algorithm introduces a set of waypoints which is denoted as three-dimension (*x*, *y*, and *z*), the *UAV's* flight is require to pass through these points from the initial to the target point. In a sequence of path points $\{S, P_1, \dots\dots, P_{n-1}, D\}$ $(n = 1,2, \dots.)$, *S* and *D* are fixed path points which represent the start and destination points, respectively, and $P_1, \dots\dots, P_{n-1}$ represents intermediate path points. Every path point is uniquely determined by location coordinates (x_i, y_i, z_i) $(i = 0, \dots., n)$ and every two adjacent points are connected with straight line segment. In this way, it can be acquire proper flight path accuracy.

For the proposed algorithm, the simulation parameters are set as follows: number of particles =10, inertia weighing factor $w = 0.02$, The cognitive scaling and social scaling factors are $c_1 = c_2 = 2$.

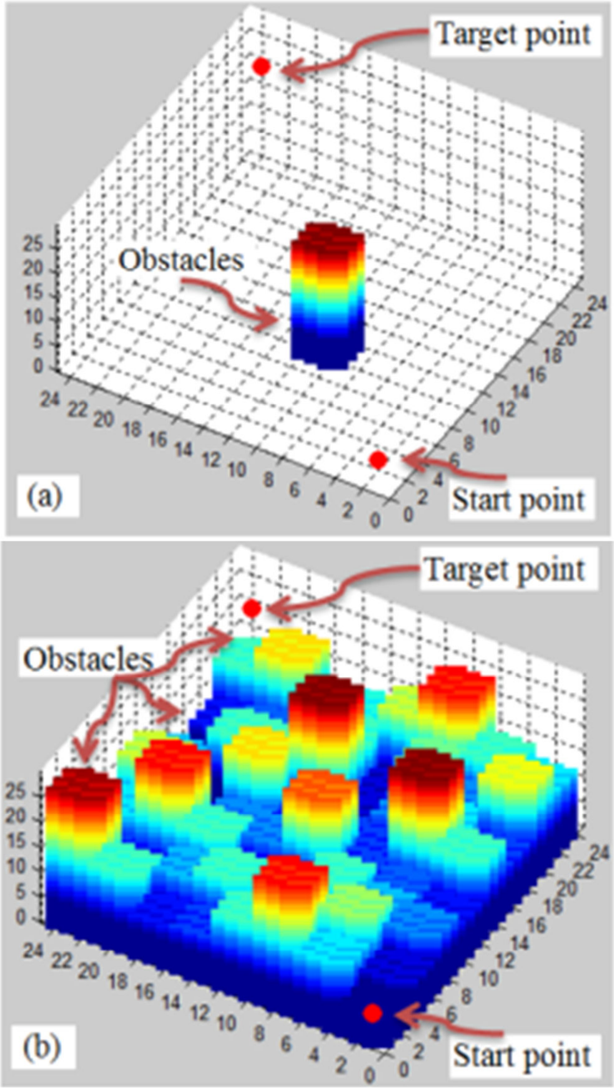

Fig. 2 Test Environments

V. Result and Discussion

A *MPSO* is successfully used for solving the path planning problem in a *3D* environment for *UAV*, assuming that the path is a sequence of cells in a *3D* grid. In this simulation work, the off-line path planner has been tested to search for path lines between columns of obstacles in simulation environments. As the aerial vehicle approached the obstacles, the proposed algorithm generated a path around the obstacles and the aerial vehicle began to track the generated path.

The result of the simulation in Fig. 3 shows the shortest path for *UAV* in the first test environment. An optimal path is formed by line segment which is connecting the global best positions "*gbest*" falling on the grids of the working environment. The *UAV* has to compulsorily pass through the points "*gbest*" which is represented by a black star signs in Fig. 3. As illustrated in the figure, the aerial vehicle moved from initial point (2, 2, 2) to destination position (23, 23, 23) while avoiding column of obstacle. The simulation result demonstrates that the proposed algorithm fits for path planning problems of mobile objects in *3D* environment with fixed obstacles.

The results presented in Fig. 4 show how the *MPSO* algorithm works to find the optimal path for the *UAV*. To visually illustrate how particles update their positions, all particles are considered. The position of a particle is influenced by the best position visited by itself which is referred to as particle best "*pbest*", and the best position in the whole swarm which is referred to as a global best "*gbest*". Swarm motion is graphed on the collared contour maps, the blue stare signs are represent the "*pbest"* and black stare signs are represent the "*gbest"*. The nodes "*gbest*" are waypoints for the *UAVs*. Route planning is accomplished by finding the shortest route through these points.

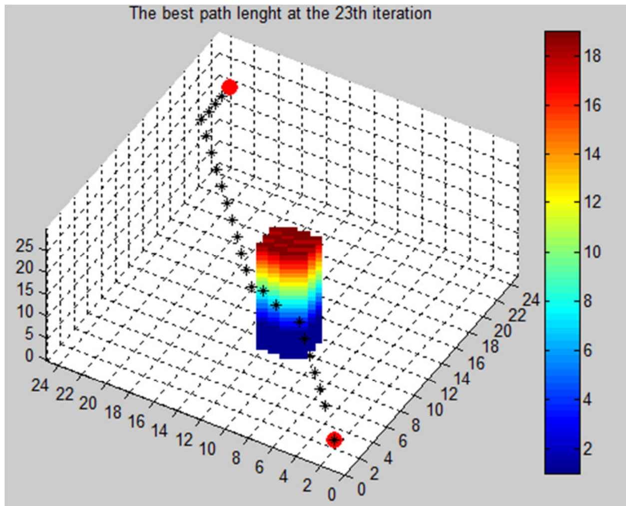

Fig. 3 *3D* flight test results for the first environment

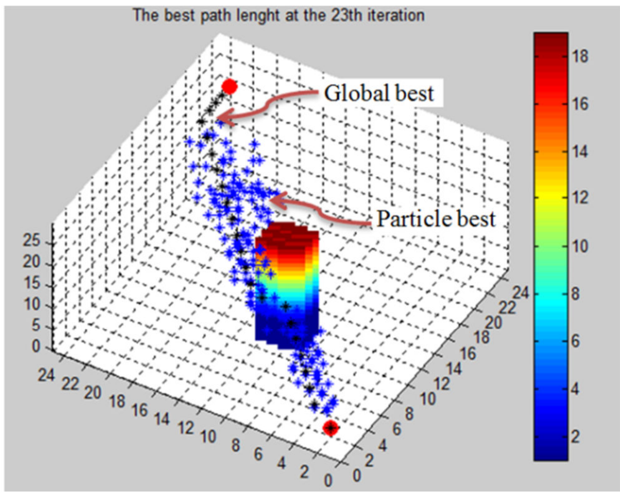

Fig. 4 *3D* flight test results

In our proposed model, the inertia weight has been adjusted adaptively. If the particles of swarm have been improved in previous iteration, it shows that the previous movement of the particles is good and they should continue their pervious movement. If the particles of swarm have been failed, it shows that their previous movement isn't good enough and it is better that these particles don't continue the previous movement.

Fig. 5 shows the predicted path at different time steps. In zero iteration, positions and velocities for each particle are randomly initialized. Local and the global best particles are initialized accordingly. In first iteration, particles move along their resultant velocity vectors to new positions. The randomly initialized velocities of iteration zero change, and particles accelerate toward the global best. In first iteration, particles are continues moving according to its inertia and social acceleration. After 4 iterations, all best particles "*pbest*" are accelerating toward the global best "*gbest*", as shown in Fig. 5. Particles iteratively follow their resultant velocity vectors to new positions, where particles seen flying from random initialization to eventual stagnation at the local minimizer. This is the continuation of the search used to illustrate how positions update in Fig. 5.

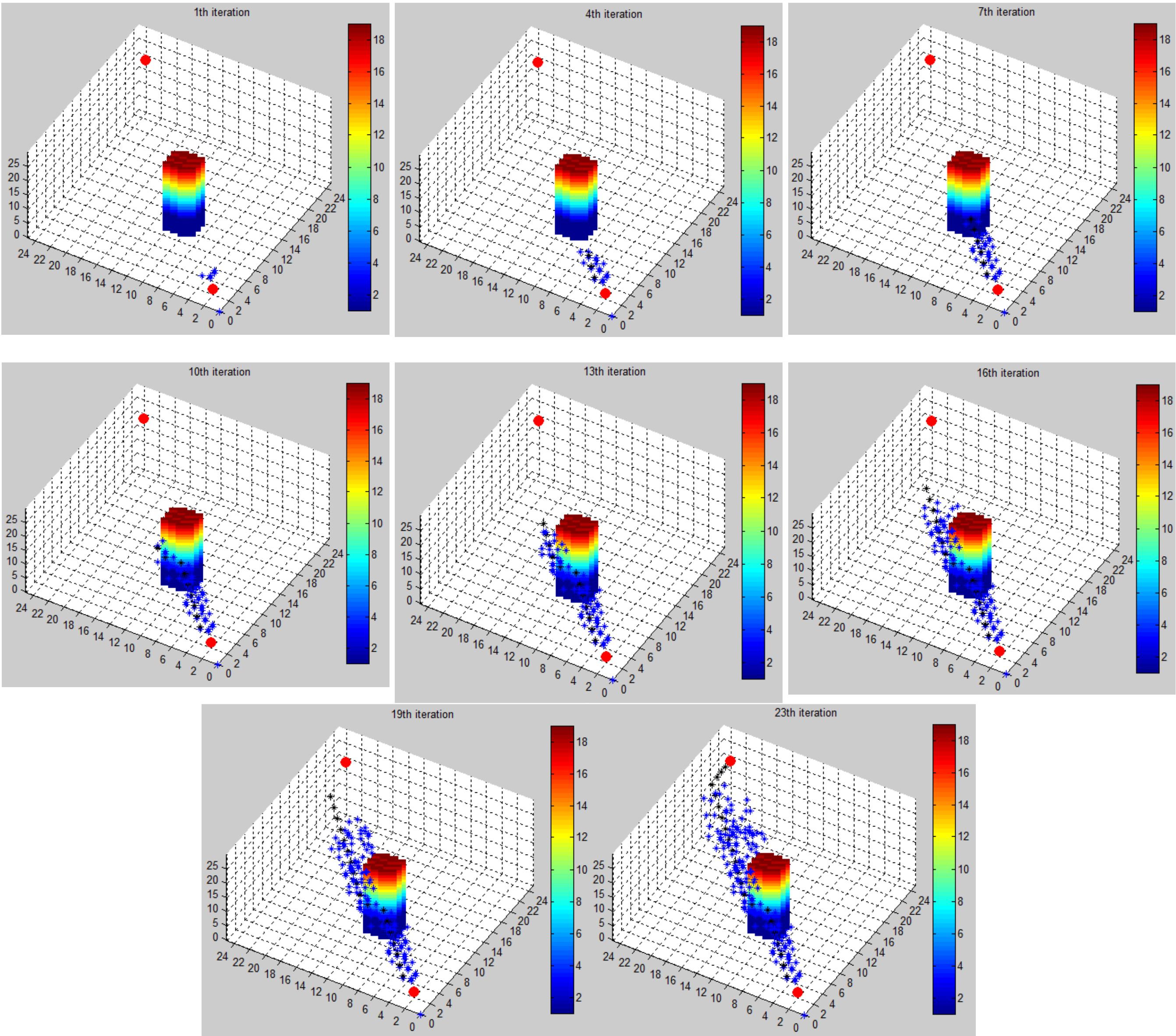

Fig. 5 Position updates of the path evolution

For the second test environment, an efficient path generated using *MPSO* where obstacle positions set up in a complex environment as shown clearly in Fig. 6. The proposed method takes 28 waypoints to reach destination point and it is the shortest path. Consequently, the path generated in this simulation was produce suitable and efficient path for aerial vehicle application.

Finding the optimal path through complex environment is discussed in the previous section. In order to fly along to the planned path, the *UAV* assigned to track the trajectory the path. In trajectory tracking, the *UAV* has to track a point as it moves along the path to complete their mission.

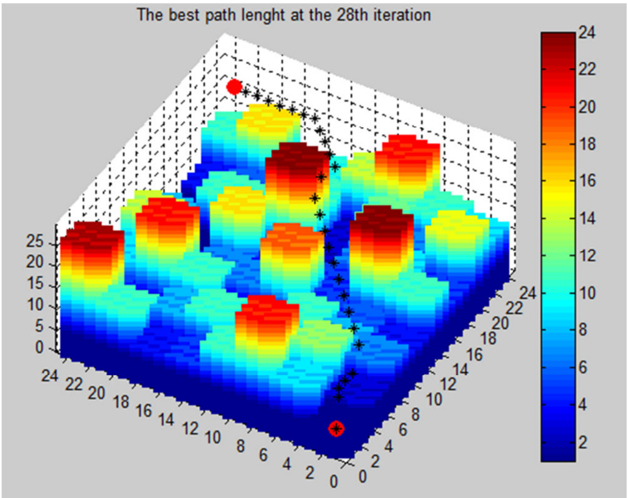

Fig. 6 *3D* flight test results for the second environment

VI. Conclusions

This paper proposed a Modified Particle Swarm Optimization (*MPSO*) algorithm for three-dimensional (*3D*) path planning of Unmanned Aerial Vehicle (*UAV*) in two test environments with fixed obstacles, where the *3D* space was approximated with grid cells in a discrete space and place obstacles in grid cells. The simulation results clearly demonstrate the power of the proposed algorithm in solving the path planning problem by providing an effective path. The resultant path is collision-avoidance with all obstacles.

References

[1] B. Taati, M. Greenspan, K. Gupta, "A dynamic load-balancing parallel search for enumerative robot path planning," Journal of Intelligent and Robotic Systems, pp. 47: 55-85, 2006.

[2] K. Sugihara, J. Smith, "A Genetic Algorithm for 3-D Path Planning of a Mobile Robot,." white paper, Department of Information and Computer Science, University of Hawaii, Manoa, Honolulu, 1999.

[3] F. Fahimi, "Autonomous Robots: Modeling, Path Planning, and Control," Canada: Mechanical Engineering Department University of Alberta, 2008.

[4] A. Tsourdos, B. White, M. Shanmugavel, "Cooperative Path Planning of Unmanned Aerial Vehicles," John Wiley & Sons, West Sussex, UK, 2011.

[5] K.P. Valavanis, "Advances in Unmanned Aerial Vehicles," State of the Art the Road to Autonomy. Springer, 2007.

[6] S. Mittal, K. Deb, "Three-dimensional offline path planning for UAVs using multiobjective evolutionary algorithms," Proceedings of the IEEE Congress on Evolutionary Computation, vol. 7, pp. 3195–3202, 2007.

[7] Y. Nikolos, N. Tsourveloudis, K. P. Valavanis, "Evolutionary Algorithm Based off-line Path Planner for UAV Navigation," Automatika, vol. 42, no. 3-4, pp. 143-150, 2001.

[8] L. D. Jalal, "Modified Particle Swarm Optimization and Modified Genetic Algorithm Approaches for Mobile Robot Navigation - A Comparative Study," University of Sulaimani, MSc thesis, 2013.

[9] A. Lazinica, "Particle swarm optimization," In-Tech, Vienna, Austria, 2009.

Lana Jalal was born in 1982. She graduated from Control and Systems Engineering, University of Technology in 2004. She received her MSc degree in Computer and Automation Engineering in 2013 and she is currently a lecturer in Electrical Engineering Department, Sulaimani University. Her research interests include Robotics and Artificial Intelligence.