

Collision Free 3D Navigation Algorithms and Their Implementation in an Autonomous Flying Robot

I. N. Ibrahim¹, A. A. Karam², M. A. Al Akkad³

¹ Department of Mechatronics, Kalashnikov Izhevsk State Technical University,
Izhevsk, Russian Federation
E-mail: ibrncfe@gmail.com

² Department of Mechatronics Engineering, University of Tehran,
Tehran, Iran
E-mail: k.maghout@gmail.com

³ Department of Computer Science, Kalashnikov Izhevsk State Technical University,
Izhevsk, Russian Federation
E-mail: aimanakkad@yandex.ru

Received: November 17, 2017

Abstract. In this paper, a method for collision-free three-dimensional autonomous navigation of an underactuated coupled non-holonomic unmanned aerial vehicle (UAV) among obstacles was proposed. This method will be the basis for designing a planner for the UAV trajectory guidance in a 3D cluttered environment. The planner assumes that the cost of flying over an area is independent of the path through which the UAV reaches that area, however this is not always true. Moreover, the path problem is not formulated as a matter of numeric cost minimization to be solved by methods like dynamic programming, which is time-consuming. A dynamic model of six degrees of freedom hexacopter equipped with a robotic arm has been formulated using Newton-Euler's method. Then, the equations of motion of the UAV are derived by including disturbances analysis. The derived dynamic model reflects the real motion of the hexacopter with respect to the earth, which is also characterized by nonlinearity, time variance, underactuation and coupling among the equations' variables. This paper suggests bio-inspired and sample-based algorithms in order to solve and optimize the three-dimensional path-planning problem. A unique real-time obstacle avoidance approach based on artificial potential field concept in addition to an off-line genetic algorithm were investigated.

Keywords: evolutionary algorithm, sample-based algorithm, bio-inspired algorithm, artificial potential field, genetic algorithms, unmanned aerial vehicles, path planning, collision avoidance.

INTRODUCTION

Tsai, et al. [1] proposed a three-dimensional real-time path planning based on rapidly-exploring random tree algorithm (RRT). However, the paths generated are, in general, not optimal due to the existence of redundant waypoints. Bagheran and Alos [2] used genetic algorithm (GA) and particle swarm algorithms to generate the path that should be a sequence of speed rate and angles at discrete times, where the cost function was calculated precisely and 3D maps were generated containing the geographic data accompanied by a digital terrain model and a geographical information system. Two approaches were investigated the artificial

potential field algorithm (APF) and the genetic algorithm (GA). APF algorithms are sorted as sampling-based algorithm because it needs the whole workspace sampling information to escape from local minima [3]. While genetic algorithm [4–5] is the most famous population-based numerical optimization method and originate from mimicking biological behavior in solving problems, Gouda [4] presented a path planning for a manipulator in a 3D environment based on a GA method, where the full and optimized path was formulated as three real chromosomes containing three-real trajectories in a fixed time. Zhang, et al. [5] proposed fuzzy logic control technology to solve the distance between the vessels to solve uncertainty-planning problem of a set of underactuated unmanned vehicles within a given space, moreover, they used GA to optimize the input/output scaling factor of a fuzzy controller. Interesting ideas for path formulation in addition to fuzzy adaptive differential evolution for path planning in a 3D environment was presented in [6–9], where UAV must have some sensing capabilities to operate in a dynamic environment while it is not required in a static environment as every information of the environment will be known beforehand. Wang and Zhang [10] used a fuzzy logic approach as a method to output a threat cost gain in order to enable the UAV escape from a sudden threat quickly, and the final route was calculated by a differential evolution algorithm. Shen, et al. [11] used a model predicting technique in real-time to find an optimal track in a three-dimensional complex environment, especially under the threat of moving targets. Where the weight of each factor of maneuvering performance was calculated by a multi-objective optimization algorithm in the search space. Kurnaz, et al. [12] used tactical air navigation (TACAN) approach and the performance of the fuzzy based controllers is evaluated with time-based diagrams, the simulation studies presented verify that the UAV can follow the predefined trajectories despite the simplicity of the controllers. Kurnaz, et al. [13] evaluated the performance of adaptive neuro-fuzzy inference system (ANFIS) based controllers in relation to the autonomous operation of UAVs, nevertheless, for some flight conditions, the ANFIS type controller has resulted in unstable performance. This has demonstrated that more stable learning algorithms need to be adopted. Vadakkepat, et al. [14] used an APF method combined with a genetic algorithm, to derive a new methodology named evolutionary artificial potential field (EAPF), the proposed method aims to navigate robots situated among moving obstacles based on optimal potential field functions. Khatib [15] proposed APF as a real-time obstacle avoidance approach for manipulators and mobile robots. APF methods have lower computational requirements than local planning approaches and could be extended to moving obstacles by using time-varying and adaptive techniques while running [16]. Wang, et al. [17] extended the range of robot’s vision in order to operate in unknown and uncertain dynamic three-dimensional environments with a number of stationary or moving obstacles. Lee, et al. [18] addressed a new inherent limitation of the potential field methods referred to as symmetrically aligned robot-obstacle-goal (SAROG), which involves critical risk of local minima trap.

PROBLEM STATEMENT

This paper suggests a design of a robot guidance algorithm, which will be used as a planner for UAV navigation with collision avoidance in a 3D cluttered environment. This design is based on bio-inspired and sampling-based algorithms with two types of known 2D-3D environments. This problem of path planning is attributed to the top layer of a robot control process. And it is described as a continuous real-time planning, in a dynamic 3D environment with a number of stationary or moving obstacles, of a non-holonomic under-actuated nonlinear UAV with bounded control input.

We suggest that the vehicle operates in a three-dimension (R^3) space called S as shown in Figure 1 that contains several static and dynamic obstacles Os_i and Od_i . Obstacles can be of any irregular shapes but we assume that obstacles are always inside or covered by a sphere of a known radius r_i and a known center c_i located in obstacle space called $S_{obstacle}$. In addition, there is also a stationary start point S and a target point T located in a three-dimensional free space called S_{free} . Now the objective is to generate trajectories of the path from S to T in S_{free} where the vehicle should always stay. Thus, the path-planning problem is defined by a triplet (S, T, S_{free}) , with the following definitions [3]:

- **Path planning** is a continuous unbreakable process F that achieves $\delta(\tau) \in S_{free}$, for all $\tau \in [0, L]$ where $\delta \subset R^3$ is a function of bounded variation, where $\delta(0) = S$ and $\delta(L) = T$.
- **Optimal path planning** is a process F^* after fulfilling F to find the optimal path δ^* whose cost is $\min(C(\delta_i))$, where δ_i is the set of all feasible paths, and C is the cost function: $\sum_i C(\delta_i) \in R | \delta_i \geq 0$.
- **Path planning augmentation** is the process of finding a continuous curve in the configuration space made up of segments of which each can be a trajectory, and starts at node S and ends at node T , without continuous time consideration, including stops in defined position.
- **Trajectory generation** is the process of taking the solution from the path planning algorithm and determining how to move along the path, considering the Kino dynamic constraints, which can be an element of the path. The trajectory is a set of states that are associated with time, described by a polynomial $X(t)$. Velocities and accelerations are computed by taking derivatives with respect to time.

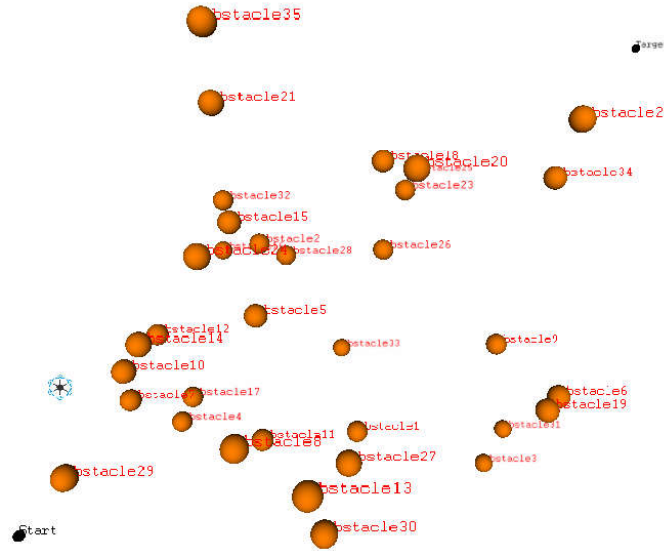


Figure 1. Three-dimension (R^3) environment

METHODOLOGY PATH FORMULATION

In this initial path planning research, it is assumed that all the information about the environment, including the obstacles' areas, is known to the UAV. In our study, the UAV is assumed to be in an environment with different obstacles' areas. The UAV has to avoid the obstacles areas; otherwise, there will be some penalty for paths passing through those areas. The

path-planning problem can also be formulated as a 3D path-planning problem or a 2D path-planning problem, by assuming the constant altitude sometime after takeoff. All three dimensions are considered in this study during the entire flight [4]. The main goals in our method are to produce an optimum 3D path by taking into our consideration real-time computational time, obstacles areas and path distance from S to T while formulating the path. Figure 2 shows Voronoi diagram that was introduced by Shamos and Hoey [45] and developed by Luchnikov et al. [44] for 3D environments. It is used to generate the topological connection. With high number of obstacles the problem turns out sophisticated and here we can see if there is a solution based on the calculation of the Voronoi channel [45]. Figure 2 demonstrates the complexity of the problem for a fixed altitude.

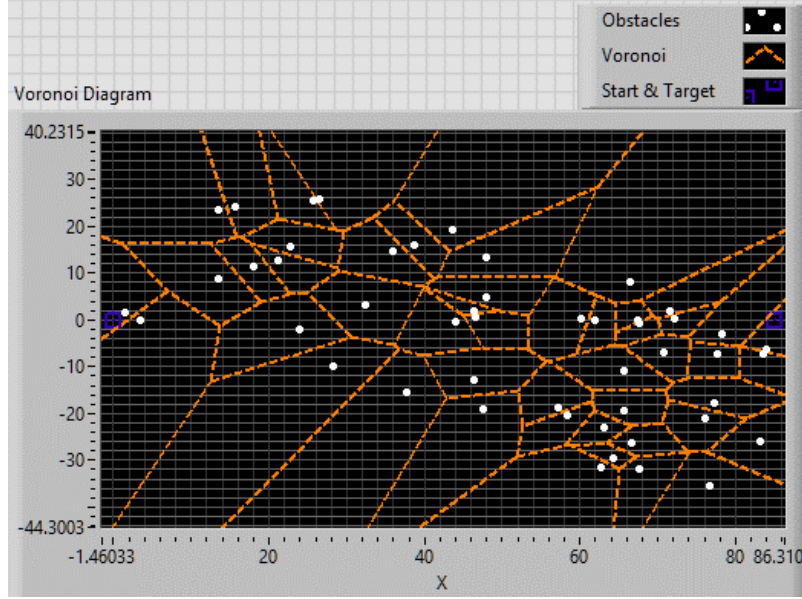


Figure 2. Voronoi diagram of environment

In this paper, two types of algorithms will be investigated: artificial potential field algorithm in the 3D environment and bio-inspired algorithms in the 2D environment after assuming the constant altitude. Firstly, the initial path process is initialized by determining the start point S and the target point T , as shown in Figure 3. There are some obstacles' areas in the task region, which are all presented in the form of a sphere. In this environment, we will apply APF algorithm which works on-line, while the vehicle is in the space, by taking only the coordinates, the center, and the radius of each obstacle. Then we will apply bio-inspired algorithms (genetic and differential evolution algorithms) which works off-line after doing some transformations in the 2D environment.

These simplifications are as follow. First, we connect point S and point T in a straight line, which is considered as the optimum path but contains penalty collisions with obstacles, so we divide this line to equal ST segments that define the accuracy of the planning, draw a vertical line of at each ST segment point. This set of lines can be denoted as $L_1, L_2, \dots, L_k, \dots, L_T$ at each line producing a discrete points collection

$$C_k = \{ \dots, (x(k), y(i-2)), (x(k), y(i-1)), (x(k), y(i)), (x(k), y(i+1)), (x(k), y(i+2)), \dots \},$$

which is called waypoint. We take a discrete waypoint at each line and connect them in sequence form in a flight path P . In this way, the path-planning problem is turning into optimizing the coordinate series to achieve a superior fitness value of the objective function. To ac-

celerate the algorithm’s search speed, we can let ST line be the X -axis and take the coordinate transformation on each discrete point $(x(k), y(k))$ according to formula (1), as shown in Figure 3, where $(-\pi < \theta < \pi)$ is the angle that the original X -axis counterclockwise rotates to parallel ST segment, while (x_s, y_s) represent the coordinates in the original coordinate system [5–6, 9]. Figure 4 shows the problem of path planning after applying the simplifications with fixed radius obstacles:

$$\begin{bmatrix} X' \\ Y' \end{bmatrix} = R_\theta \begin{bmatrix} X \\ Y \end{bmatrix}, \text{ while } R_\theta = \begin{bmatrix} \cos \theta & \sin \theta \\ -\sin \theta & \cos \theta \end{bmatrix}. \quad (1)$$

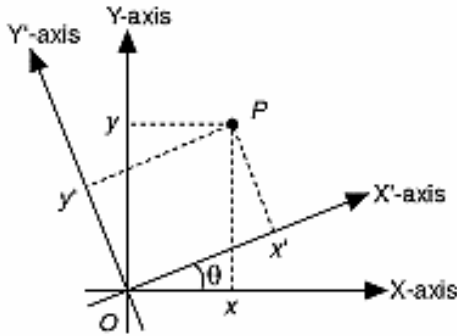


Figure 3. Rotation of a two-dimensional Cartesian coordinate

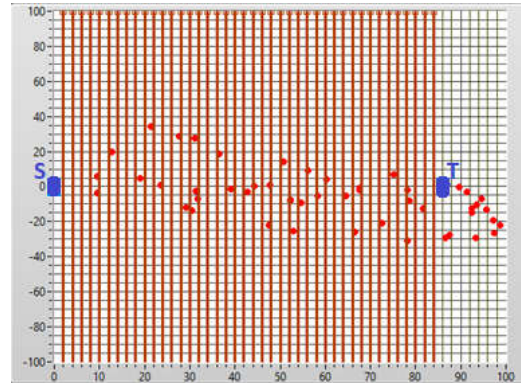


Figure 4. Path planning formulation and coordinate transformation

UAV EQUATIONS OF MOTION

In this paper the flight model equations and UAV physical constraints are similar to [21, 22]. A simplified kinematics model of a UAV flying in a three-dimensional airspace is of interest. So, the UAV is considered as a point in a 3D space, and its translational and angular states at time t are defined in [23–25] as follow:

$$\begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} \end{bmatrix} = \begin{bmatrix} U_x \\ U_y \\ U_z - g \end{bmatrix} + \begin{bmatrix} \frac{F_{dlx}}{m} \\ \frac{F_{dly}}{m} \\ \frac{F_{dlz}}{m} \end{bmatrix}, \text{ where } \begin{cases} U_x = (c\phi c\psi s\theta + s\phi s\psi)u_T / m, \\ U_y = (c\phi s\psi s\theta - s\phi c\psi)u_T / m, \\ U_z = (c\phi c\theta)u_T / m. \end{cases} \quad (2)$$

The inertial frame position of the vehicle is given by vector $\xi = [x \ y \ z]^T$. The angular position of the body frame with respect to the inertial one is usually defined by means of the Euler angles: roll ϕ , pitch θ , and yaw ψ where $c\theta$ is equivalent to $\cos \theta$, also $s\theta$ means $\sin \theta$, and so on, g is the gravity acceleration, m is the mass of aircraft, the disturbance force is the other forces like Coriolis force from the earth, wind, and Euler forces which are considered as disturbances, summarized as F_{DI} in the Earth frame: $F_{DI} = [F_{dlx} \ F_{dly} \ F_{dlz}]_E^T$, blade

rotation is with angular velocity ω . Finally, the total thrust force vector of the aircraft is u_T and it is the sum of the propellers thrust force vectors $\sum_{i=1}^6 T_i$.

OBJECTIVE FUNCTIONS

UAV path planning is generally formulated as a multi-objective optimization problem. The most critical objective is the path length, which can be directly translated to energy and obstacle costs along the path. However, there are researches that take the objectives [2, 6, 27]. In our study, the UAV path planning is considered as a bi-objective optimization problem in the aim of minimization of both energy cost and obstacle cost. The energy cost J_e is directly dependent on the path length P by measuring the distance from a specified waypoint to the target while obstacle cost J_o represents the distance from each center of an obstacle by taking into account the radius of the obstacle, as follows:

$$J_e = \int_0^P w_e dt, \quad J_o = \int_0^P w_o dt. \quad (3)$$

Where w_e , w_o are the weights for UAV for the whole path. This research focuses on two methods for finding a solution and optimization, the artificial potential field and the genetic algorithm methods. The first one is an active classical approach of sampling-based algorithms to reactive collision avoidance. This method depends on three objective functions that cause repelling from obstacles and attraction to the target, in addition to, escaping from local minima trap [18–20]. Then weight functions can be now written in 3D workspace as follows:

$$w_e = \begin{cases} 0, & \text{if UAV at the Target,} \\ \alpha \left[(x_{target} - x_{uav})^2 + (y_{target} - y_{uav})^2 + (z_{target} - z_{uav})^2 \right], & \text{otherwise,} \end{cases} \quad (4)$$

$$w_o = \begin{cases} 0, & \text{if collision free,} \\ \frac{\varepsilon}{\sum_{i=1}^m \left[(x_{obstacle_i} - x_{uav})^2 + (y_{obstacle_i} - y_{uav})^2 + (z_{obstacle_i} - z_{uav})^2 \right]}, & \text{otherwise,} \end{cases} \quad (5)$$

where m is the number of obstacles in the workspace, α and ε are the attractive and repulsive parameters respectively, $[x_{uav} \ y_{uav} \ z_{uav}]^T$ is the UAV coordination's vector, $[x_{target} \ y_{target} \ z_{target}]^T$ is the target coordinate's vector and $[x_{obstacle_i} \ y_{obstacle_i} \ z_{obstacle_i}]^T$ is the i^{th} obstacle coordinate's vector.

While the other algorithm is the genetic algorithm which is the second method discussed in this study, this is one of the bio-inspired algorithms that aims to optimize the problem based on objective cost. Like previous methods, two parameters were also analyzed in this study, distance to the target and to the obstacles. Then weight equations can be written as extended forms for each waypoint i in the vertical line of segment k and n is a number of waypoints in each level k , as follows:

$$J_{ek} = \sum_{i=1}^n D_{eik}^2, \quad (6)$$

$$J_{ok} = \sum_{t=1}^m \sum_{i=1}^n \frac{1}{D_{o_{t,jk}}^2}, \quad (7)$$

$$D_{eik} = \sqrt{(x_{target} - x_{uav_{ik}})^2 + (y_{target} - y_{uav_{ik}})^2}, \quad (8)$$

$$D_{o_{t,jk}} = \sum_{t=1}^m \sqrt{(x_{obstacle_t} - x_{uav_{ik}})^2 + (y_{obstacle_t} - y_{uav_{ik}})^2}. \quad (9)$$

ARTIFICIAL POTENTIAL FIELD METHOD

The philosophy of artificial potential field method APF can be schematically described by the movement of the vehicle in a field of forces. The position to be reached is an attractive pole for the target and the obstacles are repulsive forces for the vehicles [42] as shown in Figure 5. In addition to, an escape force for the critical risk of local minima trap, which is a new inherent limitation of potential field method, which is a symmetrically aligned robot-obstacle-goal (SAROG) [43] as shown in Figure 6. APF method has a lower computational requirement than local planning approaches [40]. These forces are described as follow:

$$F_{att} = \alpha \left[(x_{target} - x_{uav})^2 + (y_{target} - y_{uav})^2 + (z_{target} - z_{uav})^2 \right], \quad (10)$$

$$F_{rep} = \frac{\varepsilon}{\sum_{t=1}^m \left[(x_{obstacle_t} - x_{uav})^2 + (y_{obstacle_t} - y_{uav})^2 + (z_{obstacle_t} - z_{uav})^2 \right]}. \quad (11)$$

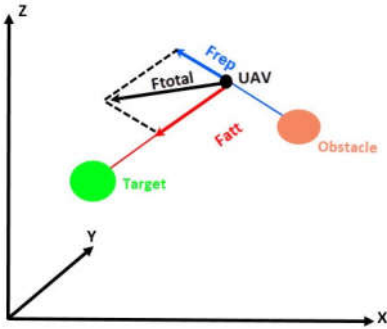


Figure 5. Normal situation of APF

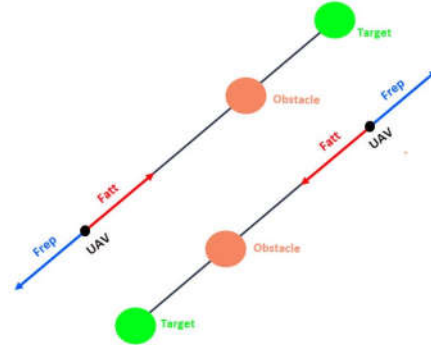


Figure 6. Critical risk of local minima traps

While the escape force was calculated based on the distances of the obstacle from the collision-free area as shown in Figure 7, we take into consideration the UAV's orientation in space:

$$F_{esc} = \begin{cases} 0, & \text{if No_SARGO,} \\ 3D_Cartesian_Coordinate_Rotation \left(\cos^{-1} \left(\frac{r_o + sd}{dv} \right) \right), & \text{if SARGO,} \end{cases} \quad (12)$$

$$\begin{bmatrix} x' \\ y' \\ z' \end{bmatrix} = BCD \begin{bmatrix} x \\ y \\ z \end{bmatrix}, \quad (13)$$

where the 3D Cartesian coordinate rotation is the BCD Euler’s rotation matrices with $(-\pi < \theta < \pi)$ as follow:

$$D = \begin{bmatrix} \cos \varphi & \sin \varphi & 0 \\ -\sin \varphi & \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad C = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \theta & \sin \theta \\ 0 & -\sin \theta & \cos \theta \end{bmatrix}, \quad A = \begin{bmatrix} \cos \psi & \sin \psi & 0 \\ -\sin \psi & \cos \psi & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (14)$$

Figure 8 shows the axes’ order around which to rotate the coordinates.

Also, $\begin{bmatrix} x \\ y \\ z \end{bmatrix}, \begin{bmatrix} x' \\ y' \\ z' \end{bmatrix}$ are the old and new coordinates of the UAV, respectively. This force

applies when SARGO occurs; this is when the UAV, the obstacle, and the target are aligned or when the target and obstacle are closely positioned. This causes oscillation and is called local minima trap. The problem condition is described as:

$$\left[(x_2 - x_1) \hat{X} + (y_2 - y_1) \hat{Y} + (z_2 - z_1) \hat{Z} \right] \times \left[(x_3 - x_1) \hat{X} + (y_3 - y_1) \hat{Y} + (z_3 - z_1) \hat{Z} \right] \leq \vec{c}, \quad (15)$$

where, \vec{c} is the vector that specifies the margin, where the UAV and target are located in regard to the obstacle.

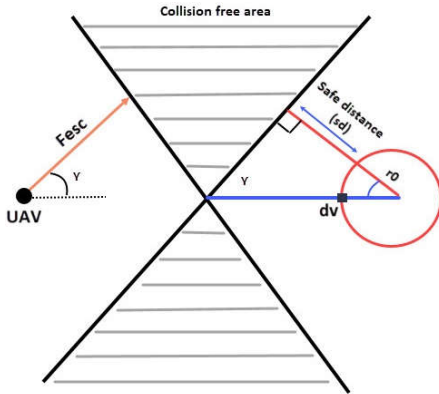


Figure 7. Escape force calculation method

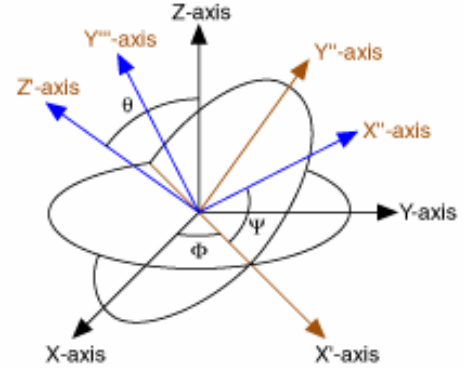


Figure 8. 3D Cartesian coordinate rotation based on Euler’s rotation

GENETIC ALGORITHMS

Bio-inspired algorithms brought in heuristic ideas, and they can excellently deal with complex and dynamic unstructured constraints [3]. It is the most popular population-based optimization method. The basic version of GA defines a cost function to evaluate the potential solutions, which are the optimal waypoints on each k level along the path. This algorithm starts by selecting randomly feasible waypoints as the first generation. Then it takes the environment, dynamic ability, target, and other constraints into consideration, to evaluate the fitness of each individual (waypoint). In the next step, a set of individuals is selected as parents for the next generation according to their fitness. The last step is a mutation and crossover step. The whole process is performed in an iterative way and the process stops when a preset goal is achieved. The best fitness waypoints are marked as the optimal path nodes achieved the shortest path to target and far away from the obstacle for each segment k as follows:

$$J_k = hJ_{ek} + (1-h)J_{ok} = h \sum_{i=1}^n D_{eik}^2 + (1-h) \sum_{t=1}^m \sum_{i=0}^n \frac{1}{D_{o_{t,ik}}^2}. \quad (16)$$

There are five parameters in GA algorithms, namely maximum generation number, length of solution (number of waypoints in this case study), population size, mutation, and crossover. Generally, solutions can evolve further when generation number is increased. The length of solution decides the complexity of the problem while population size, mutation, and crossover alter the performance of GA.

SIMULATION

Processes of testing and simulation were made using LabView software. In first simulation process of APF algorithm we suppose simple scenario as shown in Figure 9, where start point is (0, 0, 0), target point is (0, 15, 40), orange spheres are randomly generated obstacles which parameters are presented in Figure 10. This algorithm is characterized by the ability to work in real time without the need for preprocessing of the map in comparison with other algorithms. However, it requires precise adjustment and adaptation of the parameters of attraction, repelling, and escape $\alpha = 1E^{-6}$, $\varepsilon = 0.8$, $sd = 0.3$, $dv = 4r_o$, in order to get a better and more realistic path. The algorithm suffers from the trap problem that was explained before, when the aircraft is in the same plane with the target and the obstacle and was solved by the escape force. This solution is considered insufficient and requires testing and evaluation to define the extent of effectiveness.

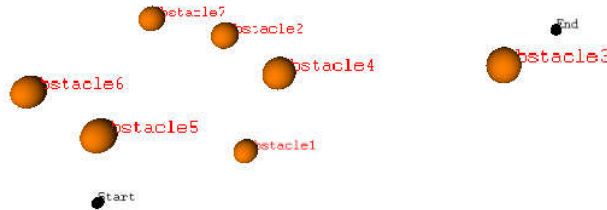


Figure 9. The scenario of testing the APF algorithm in a 3D environment

Obstacles Info.									
Oi	O_E	UAV_O	Radius	Position			SARGO D	SARGO	O_Avoid
0	35	35	1	6.8186	2.5272	0.9582	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
1	33	32	1	8.3815	13.852	9.3312	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
2	29	28	1	31.291	15.424	27.250	1168.67	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
3	32	31	1	16.761	13.777	21.871	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
4	41	41	1	6.8312	10.612	24.328	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
5	45	44	1	1.0124	12.569	21.653	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
6	40	39	1	0.5816	14.359	4.6308	Inf	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>

Figure 10. A diagram illustrating the location and the number of the spherical shapes

Simulation depends on assuming that the aircraft is a point in space regardless of motion constraints represented by motion. That is why the algorithm requires some modifications of adding flying constraints and adding adaptive techniques for automatic modification of the algorithm parameters in order to get quicker decision when overtaking obstacles of constant radius, presumably for all the obstacles, including the value of the distance between the obstacles and the target and between them and the vehicle; also, the SARGO distance that is defined according to the trap condition equation; additionally, indicators that define the trap state and a state that defines overtaking the obstacle. The obtained results of this test are due to pure mathematical operations represented by applying the mathematical vectors techniques in executing the algorithm according to [17–18] as shown in Figure 11. However, with a 3D generalization to find a solution for a multi-target problem represented by keeping away from obstacles and getting near from the target. The vector applied on the vehicle is unfeasible; there are no constraints on it. Implementing a planner for the motion path of an unmanned aerial vehicle requires making modification by imposing constraints on the motion vector along with scaling to transform the vector from pure mathematical to realistic vector. The following figures show the vector values.

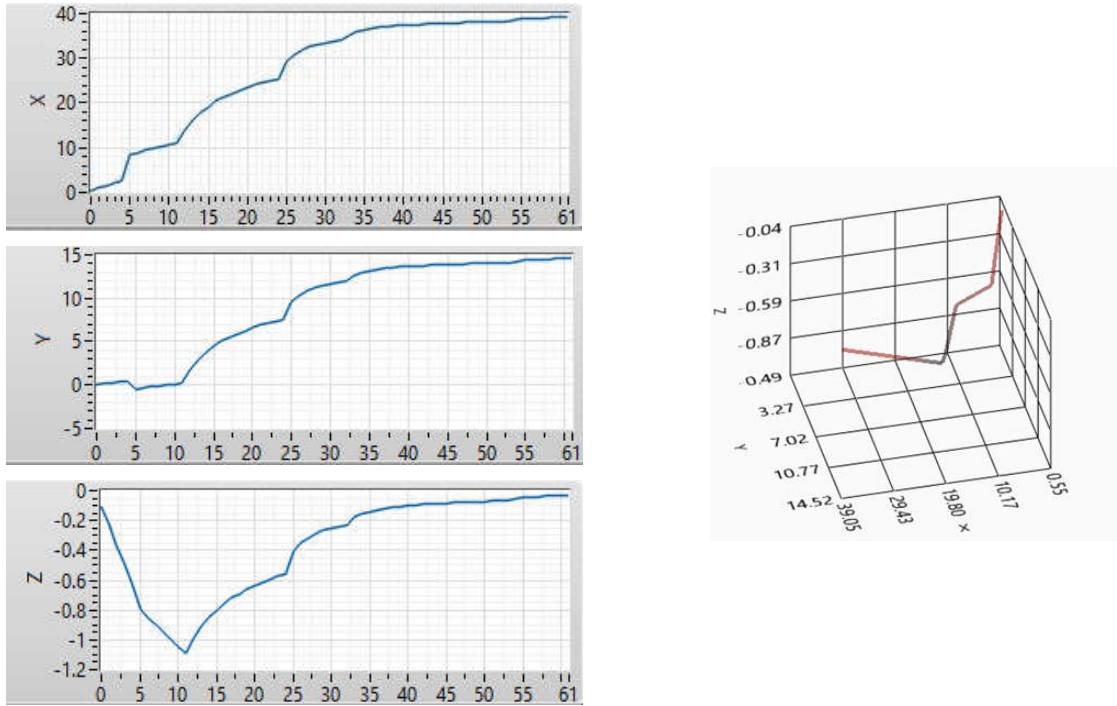


Figure 11. The results of applying APF method and coordinate vectors in 3D view

In the second simulation the GA algorithm was applied as a suggestion of a navigation algorithm with several testing scenarios, which is characterized by being more complex and contains 60 spherical obstacles of fixed diameters, generated randomly. These scenarios require making some prior transformations and maintaining a suitable height. This is to simplify the calculations as mentioned before. This theory does not work instantaneously, requires iterative processing, and consumes processing time. In real world, the shapes of obstacles are not spherical as proposed in previous scenarios. In order to apply the algorithms, it is necessary to make an approximation, which converts the obstacle into a spherical form. The following figures from 12 to 15 illustrates the phases of checking the algorithm with modifying the algorithm parameters in addition to comparing the results.

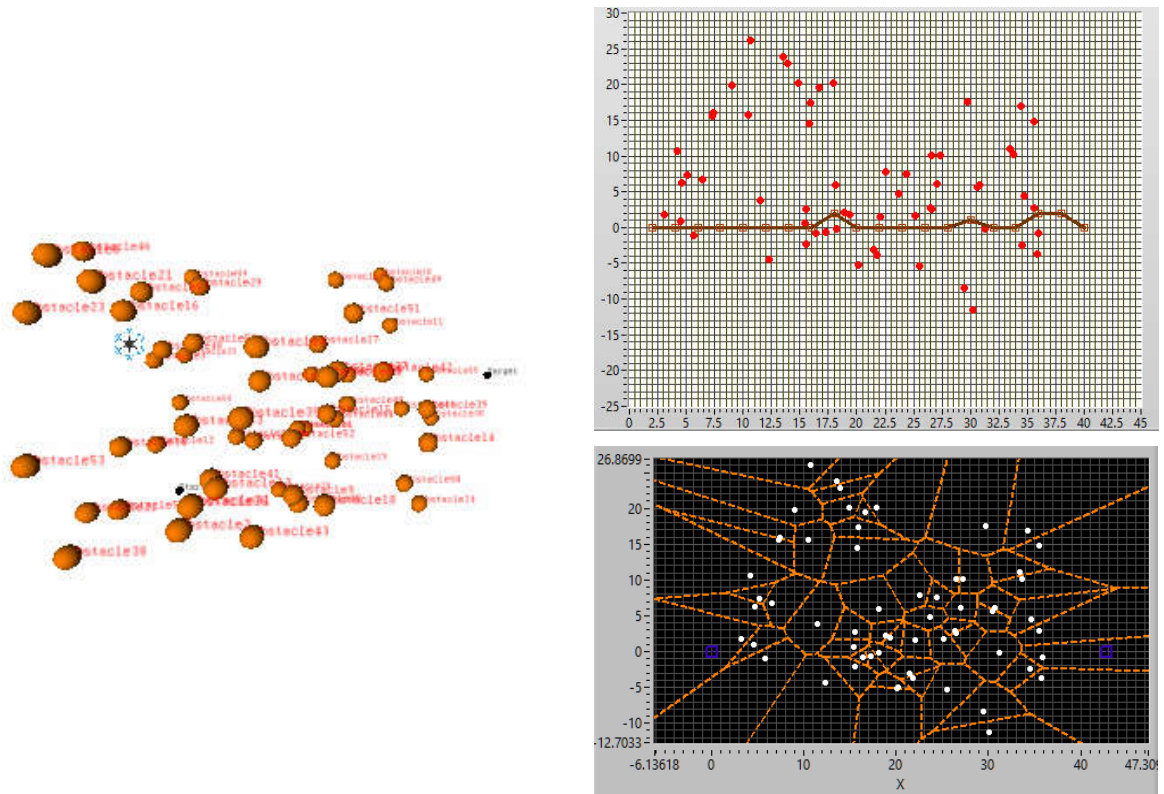


Figure 12. The results of applying the genetic algorithm in 3D environment with 60 obstacles and by using following parameters: segment length $ST = 2$, population size = 12, generation iterations = 12, crossover parameter = 0.9, mutation parameter = 0.03

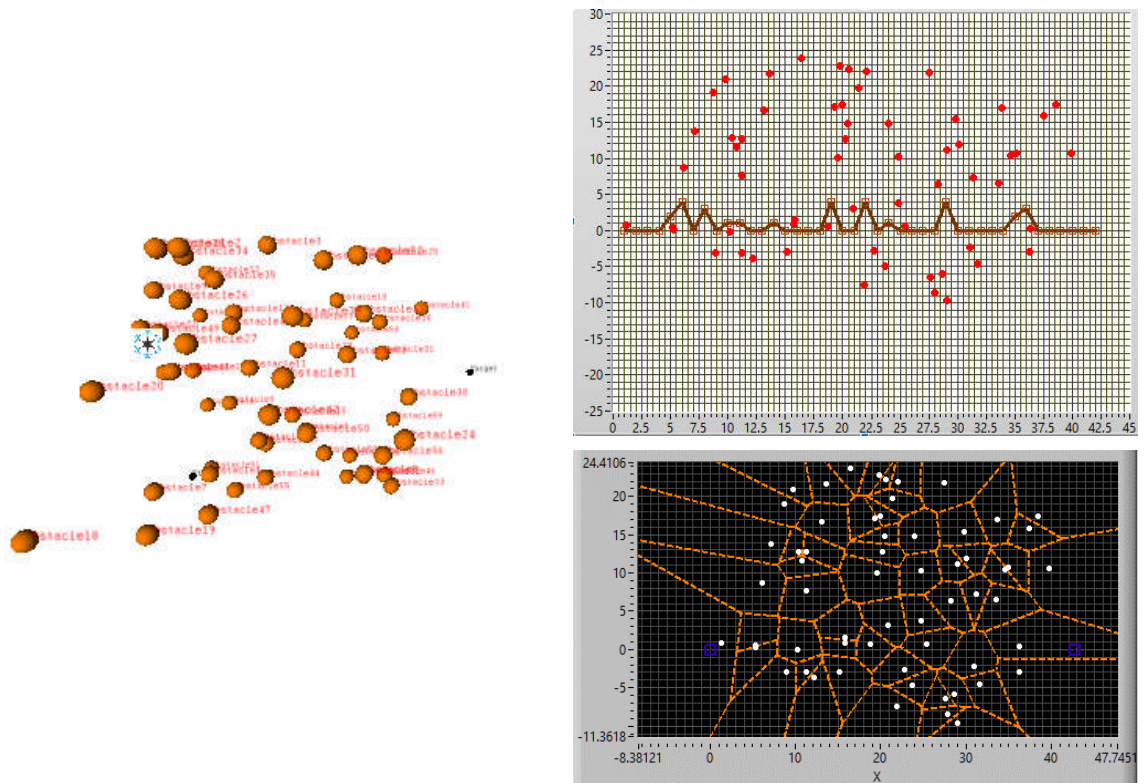


Figure 13. The results of applying the genetic algorithm in 3D environment with 60 obstacles and by using following parameters: segment length $ST = 1$, population size = 12, generation iterations = 12, crossover parameter = 0.9, mutation parameter = 0.03

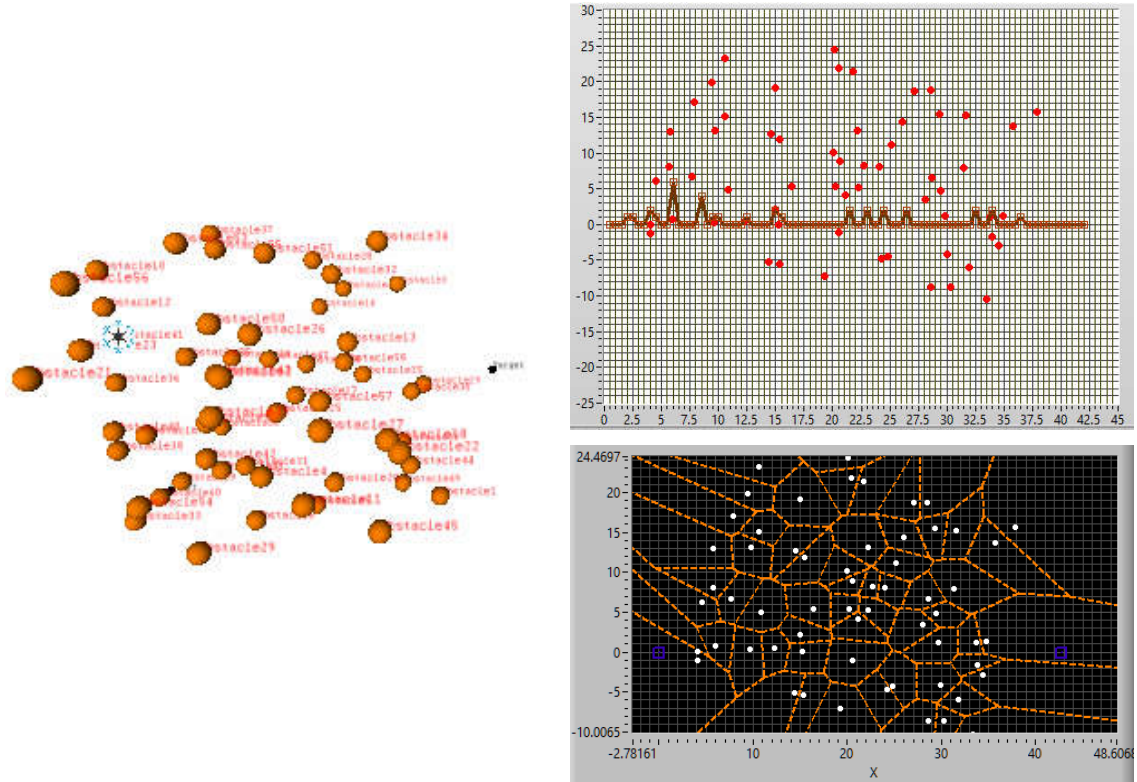


Figure 14. The results of applying the genetic algorithm in 3D environment with 60 obstacles and by using following parameters: segment length $ST = 0.5$, population size = 12, generation iterations = 12, crossover parameter = 0.9, mutation parameter = 0.03

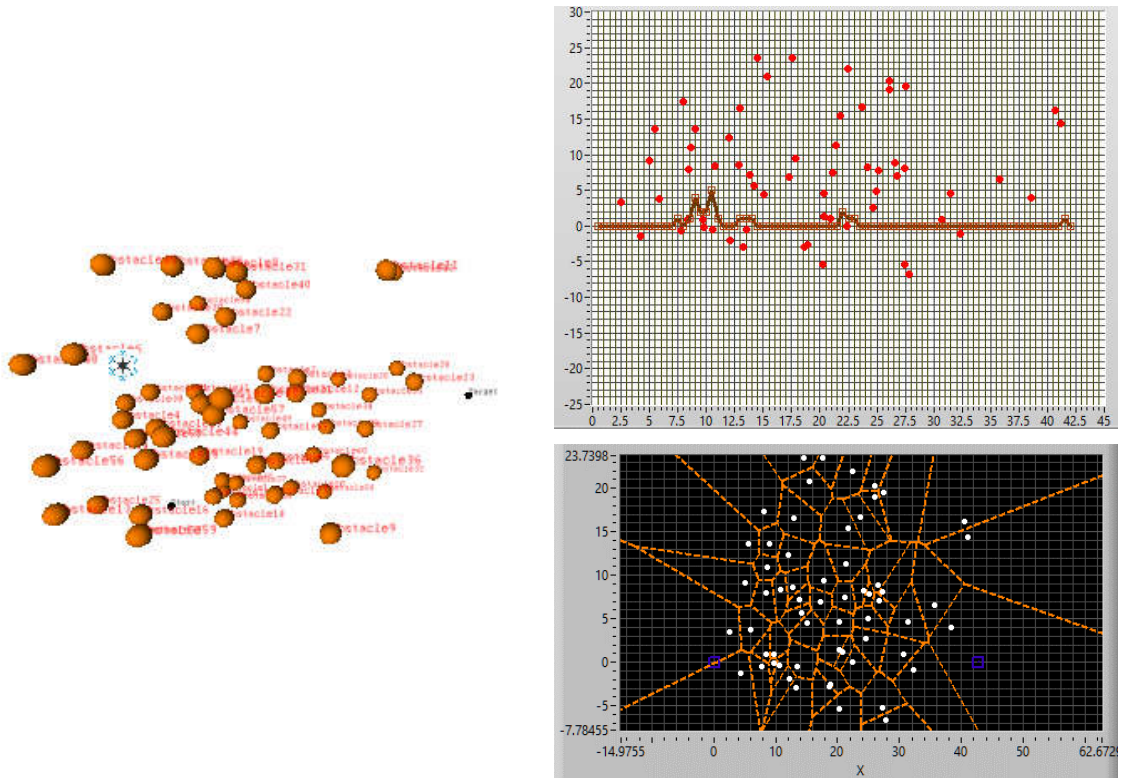


Figure 15. The results of applying the genetic algorithm in 3D environment with 60 obstacles and by using following parameters: segment length $ST = 0.5$, population size = 20, generation iterations = 20, crossover parameter = 0.9, mutation parameter = 0.03

CONCLUSION

This paper offers a comparison between two procedures of 3D navigation to prevent collision of an unmanned vehicle in space. The effectiveness of the two algorithms are defined according to computer simulation, and mathematically it is possible to modify the APF algorithm in order to get improvements in the path by adding constraints on motion and adjusting the parameters continually in order to get path planning that works in a real time. On the other hand, the genetic algorithm is considered ideal; it works in complex dynamic environments and does not require prior information about the search field. Due to the experiments, it also requires the precise adjustment of the parameters in order to get an ideal path, but it consumes processing time because its work is iterative. For applying both algorithms in flying, more detailed investigation is required. The following table 1 shows time complexity, work stages, and the environments as a comparison between the two algorithms.

Table 1. Time complexity and the environments as a comparison between the two algorithms

Method	Time Complexity	S/D Environment	Real-Time
APF Sampling-based Algorithm	$O(n \log(n)) \leq T \leq O(n^2)$	Static and Dynamic obstacles	On-line
GA Bio-inspired Algorithm	$T \geq O(n^2)$	Static and Dynamic obstacles	Off-line

REFERENCES

1. Tsai, Y. J., Lee, C. S., Lin, C. L., & Huang, C. H. (2015). Development of flight path planning for multirotor aerial vehicles. *Aerospace*, 2(2), 171–188. doi:10.3390/aerospace2020171.
2. Bagherian, M., & Alos, A. (2015). 3D UAV trajectory planning using evolutionary algorithms: a comparison study. *The Aeronautical Journal*, 119(1220), 1271–1285. doi:10.1017/S0001924000011246.
3. Yang, L., Qi, J., Song, D., Xiao, J., Han, J., & Xia, Y. (2016). Survey of robot 3D path planning algorithms. *Journal of Control Science and Engineering*, 2016, article ID 7426913, 22 pp. doi:10.1155/2016/7426913.
4. Gouda, B. K. (2006). Optimal robot trajectory planning using evolutionary algorithms (Masters thesis, Cleveland State University, USA). Retrieved from: [https://academic.csuohio.edu/embedded/Publications/Thesis/Gouda Thesis.pdf](https://academic.csuohio.edu/embedded/Publications/Thesis/Gouda%20Thesis.pdf).
5. Zhang, L., Liu, Z., Papavassiliou, C., & Liu, S. (2017). Formation path control method for group coordination based on fuzzy logic control method. *Cluster Computing*, 1–14. doi:10.1007/s10586-017-0924-2.
6. Adhikari, D., Kim, E., & Reza, H. (2017, June). A fuzzy adaptive differential evolution for multi-objective 3D UAV path optimization. In *Evolutionary Computation (CEC), 2017 IEEE Congress on* (pp. 2258–2265). doi:10.1109/CEC.2017.7969578.
7. Kok, K. Y., & Rajendran, P. (2016). Differential-evolution control parameter optimization for unmanned aerial vehicle path planning. *PLoS ONE*, 11(3), e0150558. doi:10.1371/journal.pone.0150558.
8. Zhu, W., & Duan, H. (2014). Chaotic predator–prey biogeography-based optimization approach for UCAV path planning. *Aerospace science and technology*, 32(1), 153–161. doi:10.1016/j.ast.2013.11.003.
9. Xu, C., Duan, H., & Liu, F. (2010). Chaotic artificial bee colony approach to Uninhabited Combat Air Vehicle (UCAV) path planning. *Aerospace Science and Technology*, 14(8), 535–541. doi:10.1016/j.ast.2010.04.008.
10. Wang, Q., & Zhang, J. (2017, July). MPC and TGFC for UAV real-time route planning. In *Control Conference (CCC), 2017 36th Chinese* (pp. 6847–6850). Dalian, China: IEEE. doi:10.23919/ChiCC.2017.8028436.
11. Shen, H., Chen, J., Li, H., & Zhou, Z. (2016, December). Research on real-time flight path planning of UAV based on grey prediction. In *2016 9th International Symposium on Computational Intelligence and Design (ISCID)* (Vol. 1, pp. 62–67). Hangzhou, China: IEEE. doi:10.1109/ISCID.2016.1023.

12. Kurnaz, S., Cetin, O., & Kaynak, O. (2009). Fuzzy logic based approach to design of flight control and navigation tasks for autonomous unmanned aerial vehicles. *Journal of Intelligent and Robotic Systems*, 54(1–3), 229–244. doi:10.1007/s10846-008-9263-0.
13. Kurnaz, S., Cetin, O., & Kaynak, O. (2010). Adaptive neuro-fuzzy inference system based autonomous flight control of unmanned air vehicles. *Expert Systems with Applications*, 37(2), 1229–1234. doi:10.1016/j.eswa.2009.06.009.
14. Vadakkepat, P., Tan, K. C., & Ming-Liang, W. (2000). Evolutionary artificial potential fields and their application in real time robot path planning. In *Proceedings of the 2000 Congress on Evolutionary Computation, 2000* (Vol. 1, pp. 256–263). La Jolla, CA, USA: IEEE. doi:10.1109/CEC.2000.870304.
15. Khatib, O. (1986). Real-time obstacle avoidance for manipulators and mobile robots. *The International Journal of Robotics Research*, 5(1), 90–98. doi:10.1177/027836498600500106.
16. Hoy, M., Matveev, A. S., & Savkin, A. V. (2015). Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica*, 33(3), 463–497. doi:10.1017/S0263574714000289.
17. Wang, C., Savkin, A. V., & Garratt, M. (2016). Collision free navigation of flying robots among moving obstacles. In *2016 35th Chinese Control Conference (CCC)* (pp. 4545–4549). Chengdu, China: IEEE. doi:10.1109/ChiCC.2016.7554058.
18. Lee, J., Nam, Y., & Hong, S. (2010). Random force based algorithm for local minima escape of potential field method. In *2010 11th International Conference on Control Automation Robotics & Vision (ICARCV)* (pp. 827–832). Singapore, Singapore: IEEE. doi:10.1109/ICARCV.2010.5707422.
19. Shamos, M. I., & Hoey, D. (1975). Closest-point problems. In *Proceedings of the 16th Annual Symposium on Foundations of Computer Science* (pp. 151–162). Berkeley, Calif, USA: IEEE. doi:10.1109/SFCS.1975.8.
20. Luchnikov, V. A., Medvedev, N. N., Oger, L., & Troadec, J.-P. (1999). Voronoi-Delaunay analysis of voids in systems of nonspherical particles. *Physical review E*, 59(6), 7205–7212. doi:10.1103/PhysRevE.59.7205.
21. Ibrahim, I. N., Pavol, B., Al Akkad, M. A., & Karam, A. (2017). Navigation control and stability investigation of a hexacopter equipped with an aerial manipulator. In *2017 21st International Conference on Process Control (PC)* (pp. 204–209). Strbske Pleso, Slovakia: IEEE. doi:10.1109/PC.2017.7976214.
22. Ibrahim, I. N., Al Akkad, M. A., & Abramov, I. V. (2017). Attitude and altitude stabilization of a microcopter equipped with a robotic arm. In *2017 International Siberian Conference on Control and Communications (SIBCON)* (pp. 1–8). Astana, Kazakhstan: IEEE. doi:10.1109/SIBCON.2017.7998561.
23. Ibrahim, I. N., Hussin, S., & Abramov, I. V. (2017). Minicopter nonlinear control with the existence of disturbances. In *Instrumentation Engineering, Electronics and Telecommunications – 2016: Proceedings of the International Forum* (pp. 21–33). Izhevsk, Russia: Kalashnikov Izhevsk State Technical University. ISBN: 978-5-7526-0767-7.
24. Ibrahim, I. N., & Al Akkad, M. A. (2017). Studying the disturbances of robotic arm movement in space using the compound-pendulum method. *Bulletin of Kalashnikov ISTU*, 20(2), 156–159. doi:10.22213/2413-1172-2017-2-156-159.
25. Ibrahim, I. N., Al Akkad, M. A., & Sosnovich, E. V. (2017). Optimization of hexacopter stability control with disturbances in its stochastic state space dynamics model. *55th International Scientific Conference on Experimental Stress Analysis 2017 (EAN 2017): Conference Proceedings* (pp. 731–739). Novy Smokovec, Slovakia: Technical University of Kosice. ISBN: 978-805533167-6.