Minimum Distance and Minimum Time Optimal Path Planning with Bioinspired Machine Learning Algorithms for Faulty Unmanned Air Vehicles

Onder Tutsoy[a,*], Davood Asadi[b], Karim Ahmadi[c], Seyed Yaser Nabavi-Chashmi[d], Jamshed Iqbal[e]

Abstract- Unmanned air vehicles operate in highly dynamic and unknown environments where they can encounter unexpected and unseen failures. In the presence of emergencies, autonomous unmanned air vehicles should be able to land at a minimum distance or minimum time. Impaired unmanned air vehicles define actuator failures and this impairment changes their unstable and uncertain dynamics; henceforth, path planning algorithms must be adaptive and model-free. In addition, path planning optimization problems must consider the unavoidable actuator saturations, kinematic and dynamic constraints for successful real-time applications. Therefore, this paper develops 3D path planning algorithms for quadrotors with parametric uncertainties and various constraints. In this respect, this paper constructs a multi-dimensional particle swarm optimization and a multi-dimensional genetic algorithm to plan paths for translational, rotational, and Euler angles and generates the corresponding control signals. The algorithms are assessed and compared both in the simulation and experimental environments. Results show that the multi-dimensional genetic algorithm produces shorter minimum distance and minimum time paths under the constraints. The real-time experiments prove that the quadrotor exactly follows the produced path utilizing the available maximum rotor speeds.

***Index Terms*— Actuator failures, impaired quadrotors, meta-heuristic algorithms, path planning, unmanned air vehicles.**

I. Introduction

Unmanned air vehicles (UAVs) have been primarily spurred by the military demands, but its application rapidly diversified into the areas including environment, trade and filming industry. Popularity of the UAVs mainly stems from their autonomy, lower costs, longer endurances, and networking capabilities. However, they are prone to a number of failures mostly occurring in the sensors, actuators, and mechanical parts. During failures, they have to land the most suitable area at minimum distance or at minimum time. Since they operate in unseen and complex environments under a variety of unstructured uncertainties, their autonomous and safe landing to the ground are crucial. Thus, the developed minimum distance and minimum time path planning algorithms for the UAVs should cover these properties:

- Model free to compensate the varying UAV dynamics,
- Adaptive to consider the instantly changing requirements,
- Constrained to incorporate the actuator, kinematic and dynamic limitations,
- Multi-dimensional to move in 3D space,
- Robust to alleviate the uncertainties.

In this paper, constrained multi-dimensional minimum distance and minimum time path planning algorithms for the impaired quadrotors are developed by considering the internal and external uncertainties. Modified meta-heuristic optimization algorithms, which are the sub-class of the Artificial Intelligence (AI) algorithms, plan the desired paths by optimizing the minimum distance and minimum time cost functions. Since the considered quadrotors are impaired due to partial actuator failures, the path planning solutions utilize the whole available rotor speeds.

Trajectory generation and path planning methods differ from each other where the key dissimilarity is the trajectory generation methods necessaire full certain system dynamics [1]. The trajectory generation methods are called as the mathematical model-based methods that regulate the motions by considering the amount of the control actions and the system states [2]. The overall mathematical model-based methods are Mixed Integer Linear Program (MILP), Mixed Integer Quadratic Program (MIQP), and the optimal control. In terms of the MILP, Song et al. generated trajectories for a group of UAVs by minimizing the total travelled distances [3]. This optimization algorithm considers the initial locations of the UAVs, the positions of the other UAVs, and their battery levels. Recently, Watanabe and Mukai generated trajectories with the MILP to guide a recognized wheelchair [4]. This research assumes that the position and velocity of the UAV, obstacles, and the map of the environment are available in advance. With respect to the MIQP, it is the quadratic cost function version of the MILP. Letizia et al. generated smooth polynomial UAV trajectories that satisfy the dynamic and kinematic constraints [5]. In this work, uncertainties and perturbations are modelled and added in the trajectory optimization problem. With regard to the path planning with the optimal control, Adhikari and Ruiter designed an optimal control-based trajectory generation and autonomous collision avoidance algorithm for the fixed-wing UAVs [6]. The optimal controller aims the UAVs to fly low among the obstacles and closely follow the desired target to capture accurate data. Since the mathematical model-based approaches require iterative solutions, they result in high computational costs. In addition, their solutions possibly diverge in the presence of the unseen and unmodelled uncertainties.

Decomposition graph-based methods plan the paths by dividing the environment into cells in order to discriminate the

[a]Onder Tutsoy is with Adana Alparslan Turkes Science and Technology University, Adana, Turkey, otutsoy@atu.edu.tr (Corresponding author)
[b]Davood Hendoustani is with Adana Alparslan Turkes Science and Technology University, Adana, Turkey, dasadihendoustani@atu.edu.tr
[c]Karim Ahmadi is with Adana Alparslan Turkes Science and Technology University, Adana, Turkey, kadastgerdi@atu.edu.tr
[d]Seid Yaser Nabavi is with Adana Alparslan Turkes Science and Technology University, Adana, Turkey, synabavi@atu.edu.tr
[e]Jamshed Iqbal is with University of Hull, Hull, UK, j.iqbal@hull.ac.uk

obstacles and free spaces [7]. Ischuk and Lichachev modified Dijkstra algorithm to plan an optimal path for the UAVs [8]. The algorithm uses infrared navigation data to create cells and assign edge weights based on the lengths of the cells. Mac et al. employed Dijkstra algorithm to determine collision-free UAV paths from a graph divided by the triangular decomposition [9]. The results show that the processing time of the algorithm increases significantly if the environment is complex. Another decomposition graph-based method is the A* algorithm which implements the best-search procedure; henceforth, unlike the Dijkstra algorithm it can handle the path planning problems in large environments with less computational costs. Sun et al. proposed a cooperative path planning with A* algorithm [10]. This algorithm initially explores the spaces randomly to create a map of the corresponding environment. A further decomposition graph-based method is the D* algorithm that also senses the obstacles and updates the weights of the cell edges. Celestine et al. implemented interfered fluid dynamical system and Bezier curve to plan 3D UAV paths [11]. Because all these approaches are search-based and ignore the UAV constraints, the real time results do not possibly coincide with the simulation results. Moreover, optimality of the solutions cannot be discussed due to lack of a corresponding cost function.

A similar path planning is the sampling-based approach which forms a path by randomly mapping the environment. It essentially connects the free cells with the nearby cells to create a path between the initial and target states [12]. Primatesta et al. presented kinodynamic constrained UAV motion planning algorithm leveraging the distinctive properties of a Rapidly-exploring Random Tree (RRT) and model predictive control [13]. Schmid et al. proposed RRT*-inspired online informative path planning algorithm which eliminates the local minima and the sub-optimal path convergence problems [14]. Madridano et al. developed a 3D probabilistic roadmap approach that can plan multiple trajectories in case of emergencies [15]. Sampling-based methods are not quite complete since they can fail to connect the initial states to target states because of the insufficient search abilities.

Bioinspired methods including the Neural Networks (NN) and evolutionary algorithms aim to solve complex objective problems without modelling the environment like the mathematical methods. Duan and Huang trained the NN by the imperialist competitive algorithm for global UAV path planning [16]. The comparative results with the artificial bee colony showed that the NN can reduce the uncertainty and also avoid the local minima. Sung et al. constructed a NN trained with a set of off-line UAV data which are collected from the desired trajectories [17]. Since the NN approaches can approximate any unknown functions, they attempted to incorporate the environmental uncertainties within the NN. Nikkolos et al. applied a similar off-line approach to train a differential evolution meta-heuristic algorithm that produces 2D UAV paths [18]. To collect data, the UAVs are launched from a number of locations and their initial states to terminal states were recorded. Li et al. implemented a genetic algorithm for the UAV global path planning and utilized a local rolling optimization approach to update the path continuously [19]. They processed the ground images attained from the aerial vision systems for the path planning. Even though these bioinspired methods are able to solve challenging optimization problems, their solutions are not unique and can fail in real-time applications since the dynamic, kinematic, and actuator constraints are not incorporated in the optimization problems.

Minimum time optimal path planning algorithms have been developed for a wheeled autonomous vehicle [20], for a computer numerical control (CNC) machines [21], for an industrial robot [22], and for transportation of a quantum particle [23]. However, for the best of the authors' knowledge, it is considered for the UAVs only in one research [24] where the path planning problem is reduced to the identification of the switching points among the candidates rather than autonomously planning the desired paths. Even though the minimum distance or the shortest path planning algorithms for the UAVs are ubiquities in the literature, it has not been considered for the impaired drones, for our best knowledge. Based on these stated gaps, the key contributions of the paper can be expressed as:

1. Develops minimum distance and minimum time path planning algorithms for the impaired uncertain UAVs.
2. Generates 3D paths with two modified meta-heuristic algorithms and compare the results.
3. Produces translational, rotational, and Euler angles paths where the controller can use whichever is desired.
4. Determines the maximum control signals in terms of the available impaired UAV rotor speeds.
5. Incorporates the actuator, kinematic, and dynamic constraints in the optimization problems.
6. Performs real-time experiments, analyses and compares the results extensively.

In the rest of the paper, Section II reviews the quadrotor model and introduces the cost functions, Section III constructs the multi-dimensional meta-heuristic algorithms, Section IV analysis the results, and finally Section V summarizes the paper and expresses the future works.

II. Quadrotor Model and Cost Functions

The proposed trajectory planning algorithms in this paper utilize the insights of the quadrotor model and plans the paths in simulation environment for the real-time applications. Therefore, this section initially reviews the quadrotor model and then introduces the constrained minimum distance and minimum time cost functions.

A. Quadrotor Model

The quadrotors are constructed with two crossed sticks, four mounted propellers and four motors as in Figure 1.

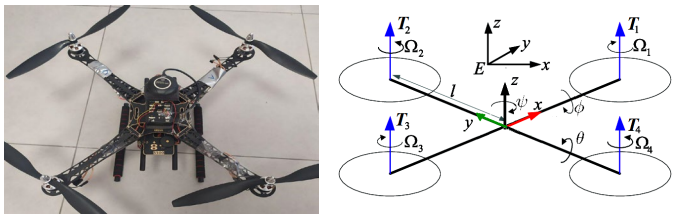

Figure 1: Schematic view of the S500 quadrotor.

In Figure 1, x, y, z are the translational position states, ϕ, θ, ψ are the Euler angles, $\Omega_i \;\; i = 1,...,4$ are the rotational velocities, $T_i \;\; i = 1,...,4$ are the produced motor torques. The quadrotor model is utilized to ensure that the generated trajectories exactly reflect these properties:

- The dynamics constraints of the quadrotor,
- The kinematic constraints of the quadrotor,
- The motor velocity constraints of the quadrotor,
- The non-linear coupling effects among its sub-models,
- The parametric and non-parametric uncertainties,

Using the quadrotor model to generate trajectories in simulation environments allows us to train the intrinsically model-free algorithms in simulation environments without damaging the unstable and uncertain quadrotors. However, in real time applications, due to its model free properties, the proposed algorithms should be able:

- Modify the planned paths in simulations,
- Incorporate the un-modelled dynamics,
- Consider the unseen uncertainties,

The next sub-section reviews the translational model of the quadrotor.

1) *The Translational Model*

The quadcopter translates itself from the initial states with the translational model given by

$$\ddot{x} = \left(\sin\psi \sin\phi + \cos\psi \sin\theta \cos\phi\right)\frac{u_z}{m} + f_x^{drag} \tag{1}$$

$$\ddot{y} = \left(-\cos\psi \sin\phi + \sin\psi \sin\theta \cos\phi\right)\frac{u_z}{m} + f_y^{drag} \tag{2}$$

$$\ddot{z} = -g + \left(\cos\theta \cos\phi\right)\frac{u_z}{m} + f_z^{drag} \tag{3}$$

where u_z is the control signal, m is the mass, g is the gravity force and the drag forces are expressed as

$$f_x^{drag} = -\frac{k_x}{m}\dot{x}, \quad f_y^{drag} = -\frac{k_y}{m}\dot{y}, \quad f_z^{drag} = -\frac{k_z}{m}\dot{z} \tag{4}$$

where k_x, k_y, k_z are the translational drag forces. One can summarize the properties of the translational model as

- It is non-linear due to sine and cosine functions,
- It is coupled since the same Euler angles are shared,
- It is constrained because the sine, cosine functions and certain limits on u_z control signal.

The next sub-section provides the rotational model of the quadrotor.

2) *The Rotational Model*

The quadrotor must rotate in order to translate. Therefore, this paper also generates $\dot{p}$, $\dot{q}$, $\dot{r}$ rotational velocities expressed as

$$\dot{p} = a_p qr + b_p u_\varphi + \tau_p \tag{5}$$

$$\dot{q} = a_q pr + b_q u_\theta + \tau_q \tag{6}$$

$$\dot{r} = a_r pq + b_r u_\psi + \tau_r \tag{7}$$

where a_p, a_q, and a_r are the rotational parameters given by

$$a_p = \frac{I_y - I_z}{I_x}, \quad a_q = \frac{I_z - I_x}{I_y}, \quad a_r = \frac{I_x - I_y}{I_z} \tag{8}$$

where I_x, I_y, I_z are the moment of inertias around the x, y, z-axes, respectively. The b_p, b_q, and b_r parameters in Equations (5) to (7) are

$$b_p = \frac{1}{I_x}, \quad b_q = \frac{1}{I_y}, \quad b_r = \frac{1}{I_z} \tag{9}$$

The τ_p, τ_q, and τ_r forces in Equations (5) to (7) are

$$\tau_p = \tau_p^{drag} + \tau_p^{gyro} + \tau_p^{wind} \tag{10}$$

$$\tau_q = \tau_q^{drag} + \tau_q^{gyro} + \tau_q^{wind} \tag{11}$$

$$\tau_r = \tau_r^{drag} + \tau_q^{wind} \tag{12}$$

The parametric drag forces are given by

$$\tau_p^{drag} = -\frac{k_\varphi}{I_x}p^2, \quad \tau_q^{drag} = -\frac{k_\theta}{I_y}q^2, \quad \tau_p^{drag} = -\frac{k_\psi}{I_z}r^2 \tag{13}$$

where k_φ, k_θ, k_ψ are the rotational drag constants. The gyroscopic forces in Equations (10) to (12) are formulated as

$$\tau_p^{gyro} = \frac{J_T}{I_x}q\Omega, \quad \tau_q^{gyro} = -\frac{J_T}{I_y}p\Omega \tag{14}$$

where J_T is the total rotational moment of inertia and Ω is the total rotational speeds defined as

$$\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \tag{15}$$

The inputs $u_z, u_\varphi, u_\theta, u_\psi$ in Equations (3),(5),(6) and (7) are

$$u_z = b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \tag{16}$$

$$u_\varphi = bl\left(\Omega_2^2 - \Omega_4^2\right) \tag{17}$$

$$u_\theta = bl\left(\Omega_3^2 - \Omega_1^2\right) \tag{18}$$

$$u_\psi = d\left(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2\right) \tag{19}$$

where b is the thrust parameter, l is the moment arm, and d is the drag parameter. The Euler angles, presented next, are essential parts of the translational trajectories in Equations (1) to (3) and the rotational trajectories in Equations (5) to (7).

3) *The Euler Angles*

The Euler angles constructs the desired translational and rotational trajectories. The corresponding Euler angles are

$$\dot{\varphi} = p + q\sin\varphi\tan\theta + r\cos\varphi\tan\theta \quad (20)$$

$$\dot{\theta} = q\cos\varphi - r\sin\varphi \quad (21)$$

$$\dot{\psi} = \frac{1}{\cos\theta}\left[q\sin\varphi + r\cos\varphi\right] \quad (22)$$

Properties of the quadrotor Euler angles can be highlighted as:

- They are directly coupled with the $\dot{p}, \dot{q}, \dot{r}$ rotational velocities,
- They are indirectly coupled with the $\ddot{x}, \ddot{y}, \ddot{z}$ translational accelerations.

The next sub-section introduces the cost functions.

B. Cost Functions

To enhance the autonomy of the UAVs, they should be equipped with various self-optimization approaches. This section introduces the minimum distance and minimum time cost functions that optimize the quadrotor paths.

1) *Minimum Distance Cost Function*

The quadrotors can require to land off the available ground at the minimum distance. The corresponding cost function is

$$\min \quad e_d = \int_{t=0}^{t_f} \left\|\mathbf{p}_d - \mathbf{p}_t(t)\right\|_1 dt$$

subject to (23)

$$\mathbf{a}_e^{\min} \le \mathbf{a}_e(t) \le \mathbf{a}_e^{\max},\ \mathbf{p}_t^{\min} \le \mathbf{p}_t(t) \le \mathbf{p}_t^{\max},\ 0 \le \dot{\mathbf{p}}_t(t) \le \dot{\mathbf{p}}_t^{\max}$$
$$\mathbf{r}_t^{\min} \le \mathbf{r}_t(t) \le \mathbf{r}_t^{\max},\ \ \mathbf{u}^{\min} \le \mathbf{u} \le \mathbf{u}^{\max}$$

where e_d is minimum distance cost, t_f is finite time, $\mathbf{p}_d$ is terminal translational position, $\mathbf{p}_t(t) = \left[x(t) \quad y(t) \quad z(t)\right]^T$ is instant translational position in Equations (1) to (3), $\dot{\mathbf{p}}_t(t)$ is instant translational velocity, $\mathbf{a}_e(t) = \left[\varphi(t) \quad \theta(t) \quad \psi(t)\right]^T$ is Euler angles in Equations (20) to (22), $\mathbf{r}_r(t) = \left[p(t) \quad q(t) \quad r(t)\right]^T$ is instant rotational positions, $\mathbf{u} = \left[u_z \quad u_\varphi \quad u_\theta \quad u_\psi\right]^T$ are the input signals in Equations (16) to (19), $(.)^{\min}$ and $(.)^{\max}$ are the minimum and maximum values. It is important to note that even though these constrains seem static, in case of an actuator failure, the resulting rotational velocities, so that the translational states and the control signal constraints vary accordingly. For instance, in the presence of the second actuator failure, the constraints on the input signals in Equations (16) to (19) are re-determined based on its available speed. Therefore, the constraints on the rotational speeds in Equations (5) to (7) and translational states in Equations (1) to (3) are re-shaped since they are direct function of the input signals. One can summarize the key properties of the cost function in Equation (23) as

- It has vector 1-norm $\|.\|_1$, hence a scalar value represents the overall performance of the 3-dimensional translational trajectory space.
- Since t_f is initially unknown, instant cost value which is the minimum of all the available costs, is utilized.

The next sub-section introduces the minimum time cost function.

2) *Minimum Time Cost Function*

In case of emergencies, the quadrotors are required to land off at minimum time. The minimum time cost function is

$$\min \quad T = \int_{t=0}^{t_f} 1 dt$$

subject to (24)

$$\mathbf{a}_e^{\min} \le \mathbf{a}_e(t) \le \mathbf{a}_e^{\max},\ \mathbf{p}_t^{\min} \le \mathbf{p}_t(t) \le \mathbf{p}_t^{\max},\ 0 \le \dot{\mathbf{p}}_t(t) \le \dot{\mathbf{p}}_t^{\max}$$
$$\mathbf{r}_t^{\min} \le \mathbf{r}_t(t) \le \mathbf{r}_t^{\max},\ \ \mathbf{u}^{\min} \le \mathbf{u} \le \mathbf{u}^{\max}$$

The cost function in Equation (24) is a time optimal problem for a third order non-linear system with non-linear constraints. Therefore, its direct solution is not straightforward. To ease the solution, re-formulate the minimum time cost function as

- Minimum Time = Minimum Distance/Maximum Velocity

Hence the minimum time problem becomes

$$\min \quad T = \int_{t=0}^{t_f} \frac{\left\|\mathbf{p}_d - \mathbf{p}_t(t)\right\|_1}{\left\|\dot{\mathbf{p}}_t^{\max}(t)\right\|_1} dt$$

subject to (25)

$$\mathbf{a}_e^{\min} \le \mathbf{a}_e(t) \le \mathbf{a}_e^{\max},\ \mathbf{p}_t^{\min} \le \mathbf{p}_t(t) \le \mathbf{p}_t^{\max},\ 0 \le \dot{\mathbf{p}}_t(t) \le \dot{\mathbf{p}}_t^{\max}$$
$$\mathbf{r}_t^{\min} \le \mathbf{r}_t(t) \le \mathbf{r}_t^{\max},\ \ \mathbf{u}^{\min} \le \mathbf{u} \le \mathbf{u}^{\max}$$

In Equation (25) $\dot{\mathbf{p}}_t^{\max}(t)$ does not necessarily imply that all the motors must rotate at their maximum speeds. To satisfy certain translational movements, faulty motors must generate the required rotational speeds. The next section introduces the multi-dimensional meta-heuristic algorithms which plan the minimum distance and minimum time paths.

III. Multi-Dimensional Meta-Heuristic Algorithms

This section briefly introduces the modified Multi-dimensional Particle Swarm Optimization (M-PSO) and Multi-Dimensional Genetic Algorithm (M-GA) that optimize the minimum distance and minimum time path planning cost functions.

A. Quadrotor Path Planning with Constrained M-PSO

The PSO algorithm focuses on optimizing a cost function by selecting a number of members among a population. In this paper, M-PSO iteratively choses the optimum Euler angles that reduces the minimum distance cost function in Equation (23) and the minimum time cost function in Equation (25). The path planning process is feedforward; hence, back movement of the quadrotor is hindered.

The PSO algorithm updates the Euler angles by a rate expressed as

$$\mathbf{v}(t+\Delta t) = \eta^v \mathbf{v}(t) + r_1\eta^p\left(\mathbf{p}(t) - a_e(t)\right) + r_2\eta^g\left(\mathbf{g}(t) - a_e(t)\right) \quad (26)$$

where $\mathbf{p}(t)$ and $\mathbf{g}(t)$ are the local and global best Euler angle solutions among its constrained ones $\mathbf{a}_e^{\min} < \mathbf{a}_e^t < \mathbf{a}_e^{\max}$, respectively, η^v, η^p and η^g are the corresponding learning parameters, r_1 and r_2 are the uniformly distributed random exploration noises. The update rule for the Euler angles is

$$\mathbf{a}_e(t+\Delta t) = \mathbf{a}_e(t) + \mathbf{v}(t+\Delta t) \tag{27}$$

The best local $\mathbf{p}(t)$ and global $\mathbf{g}(t)$ solutions are attained depending on the minimum values of the cost functions in Equations (23) and (25). At the end of each epoch, the best local Euler angle solutions are assigned as the global best Euler angles solution. The global best Euler angles solutions are used to calculate the translational trajectories in Equations (1) to (3) and the rotational trajectories in Equations (20) to (22). Algorithm 1 summarizes the path planning with the M-PSO.

Algorithm 1: Pseudocode of M-PSO path planning algorithm.

Input:

The parameters η^v, η^p, η^g in Equation (26) and η^d discount,

The limits $v^{\max} = 0.2\left(a_e^{\max} - a_e^{\min}\right)$, $v^{\min} = -v^{\max}$,

The initial and target translational trajectories $\mathbf{p}(1)$ and $\mathbf{p}_d$,

The upper and lower limits of the constraints in Equation (24),

Output:

Optimized quadrotor states $\mathbf{a}_e(t)$, $\mathbf{p}(t)$, $\mathbf{r}_t(t)$,

for $t = 1$ **to** *trajectory length*

for $j = 1$ **to** *repeat length*

for $k = 2$ **to** *search length*

1. Generate a random Euler angle population $\mathbf{a}_e(k)$.

$$\mathbf{a}_e(k) = unifrnd\left(\mathbf{a}_e^{\max}, \mathbf{a}_e^{\min}, 3\right) \tag{28}$$

2. Determine the update rate $\mathbf{v}(k)$ by Equation (26).
3. Update the Euler angles $\mathbf{a}_e(k)$ by Equation (27).
4. Determine the control signal $u_z(k)$ in Equation (16).

$$u_z(k) = sign\left(\mathbf{K} * \mathbf{p}(k-1)\right) \tag{29}$$

5. Apply the control signal constraints on $u_z(k)$
6. Obtain translational states $\mathbf{p}(k)$ in Equations (1) to (3)
7. Apply the translational position constraints on $\mathbf{p}(k)$.
8. Calculated the costs in Equations (23) or (25).
9. Update the local best solutions as:

if current cost $e(k)$ **is less than** ***personal best*** e_p

$$\mathbf{a}_e^p = \mathbf{a}_e(k), \mathbf{p}^p = \mathbf{p}(k), \mathbf{u}^p = \mathbf{u}_z(k),\ e_p = e(k)$$

end if

10. Update the global best solution as:

if personal best e_p **is less than** ***global best*** e_g

$$\mathbf{a}_e^g = \mathbf{a}_e^p, \mathbf{p}^g = \mathbf{p}^p, u_z^g = u_z^p,\ e_g = e_p$$

end if

end for

11. Update the learning parameter $\eta^v = \eta^v.\eta^d$ in (26).
12. Initialize optimization with the global best solutions.

end for

13. Save the global best trajectory solutions

$$\mathbf{a}_e(t) = \mathbf{a}_e^g, \mathbf{p}(t) = \mathbf{p}^g, u_z(t) = u_z^p$$

14. Obtain the rotational trajectory $\mathbf{r}_t(t)$ by Equations (20) to (22).

end for

The next sub-section presents the quadrotor path planning with the constrained M-GA.

B. Quadrotor Path Planning with the Constrained M-GA

The GA is inspired by the natural selection process of the springs among a population. The whole population represents the possible candidate solutions for the cost function.

Algorithm 2: Pseudocode of M-GA path planning algorithm.

Input:

The selection η^s, crossover η^c, mutation η^m parameters,

The limits $v^{\max} = 0.2\left(a_e^{\max} - a_e^{\min}\right)$ and $v^{\min} = -v^{\max}$.

The initial and target translational trajectories $\mathbf{p}(1)$ and $\mathbf{p}_d$,

The upper and lower limits of the constraints in Equation (24),

Output:

Optimized quadrotor states $\mathbf{a}_e(t)$, $\mathbf{p}(t)$, $\mathbf{r}_t(t)$,

Initialization:

Initialize the cost e, cumulative sum $c(1) = 0$, parent $\mathbf{a}_e^p$,

for $t = 1$ **to** *trajectory length*

for $k = 2$ **to** *search length* k^l

1. Generate a random Euler angle population $\mathbf{a}_e(k)$.
2. Determine the selection probabilities $\eta^p = e^{-\left(\eta^s e^m\right)}$

where the mean cost is $e^m = \frac{1}{k^l}\sum_{i=1}^{k^l} e(i)$.

3. Perform the crossover process with the half population.

for $j = 2$ **to** $k^l / 2$

4. Select the parents $\mathbf{a}_e^{p_1}$ with Roulette Wheel.

$$r = rand * \sum_{i=1}^{k^l} \eta^p(i),\ c(j) = c(j-1) + \eta^p(j) \quad (30)$$

If $r \le c(j)$ ***then*** $\mathbf{a}_e^{p,1} = \mathbf{a}_e^{p,1} + 1$ ***end***

5. Repeat Equation (30) for $\mathbf{a}_e^{p,2}$.
6. Process crossover with $\eta^c = unifrnd\left(-\eta^c, 1+\eta^c, k^l\right)$ as

$$\mathbf{a}_e^{c,1} = \eta^c .* \mathbf{a}_e^{p,1} + \left(1-\eta^c\right) .* \mathbf{a}_e^{p,2} \quad (31)$$

end for

7. Perform the mutation with the $\mathbf{a}_e^c = \left[\mathbf{a}_e^{c,1}; \mathbf{a}_e^{c,2}\right]$.

for $l = 2$ **to** k^l

8. Process mutation $\mathbf{a}_e^m(l) = \mathbf{a}_e^c(f) + \eta^s * randn(3)$ with

$$f = rand(3) \le \eta^t \quad (32)$$

9. Check for the Euler angle bounds.
10. Determine the control signal $u_z(l)$ in Equation (16).
11. Apply the control signal constraints on $u_z(l)$.
12. Obtain the translational states $\mathbf{p}(l)$ in Equations (1) to (3).
13. Apply the translational position constraints on $\mathbf{p}(l)$.
14. Calculated the costs in Equations (23) or (25).
15. Update the local best solutions as:

if current cost $e(l)$ **is less than** ***personal best*** e_p

$$\mathbf{a}_e^p = \mathbf{a}_e(l), \mathbf{p}^p = \mathbf{p}(l), u^p = u_z(l),\ e_p = e(l)$$

end if

end for

16. Save the personal best e^p as global best e^g.
17. Initialize optimization with the global best solutions.

end for

18. Save the global best trajectory solutions

$$\mathbf{a}_e(t) = \mathbf{a}_e^g, \mathbf{p}(t) = \mathbf{p}^q, u_z(t) = u_z^p$$

19. Obtain the rotational trajectory $\mathbf{r}_t(t)$ by Equation (20) and (22).

end for

The next section provides the results and analyses them extensively.

IV. Results

This section provides the parameters of the quadrotor, meta-heuristic algorithms and then analyses the results.

A. Parameters of the Quadrotor and the Algorithms

Table 1: Parameters of the S500 quadrotor UAV.

Parameter	Description	Value
m	Mass	1.7 kg
b	Thrust parameter	4.1×10^{-7} N/rpm^2
d	Drag parameter	8×10^{-9}Nm/rpm^2
l	Moment arm	0.243m
I_x	Moment inertia about x-axis	0.0213kg.m^2
I_y	Moment inertia about y-axis	0.0221kg.m^2
I_z	Moment inertia about z-axis	0.028kg.m^2
k_x, k_y	Translational drag coefficients	5.5×10^{-4}N/m/s
k_z	Translational drag coefficients	6.3×10^{-4}N/m/s
k_φ, k_θ	Rotational drag coefficients	5.5×10^{-4}N/m/s
k_ψ	Rotational drag coefficient	6.3×10^{-4}N/rad/s
J_t	Total moment inertia	6.8×10^{-4}kg.m^2
a_p	Parameter of the p-sub-model	-0.2770
a_q	Parameter of the q-sub-model	0.3032
a_r	Parameter of the r-sub-model	-0.0286

The M-PSO algorithm parameters η^v, η^p, η^g are 1.45, 2.99, 2.99, respectively. The M-GA algorithm parameters η^s, η^c, η^m are 0.1, 0.02, 0.1, respectively.

B. Minimum Distance Path Planning

Figure 2 presents the minimum distance quadrotor translational position planning with the M-PSO and M-GA meta-heuristic optimization algorithms.

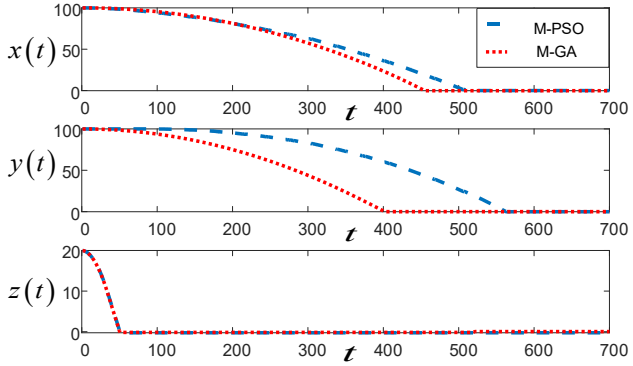

Figure 2: Minimum distance translational positions.

Initial translational positions of the quadrotor are 100, 100, 20 for the $x(t)$, $y(t)$ and $z(t)$, respectively, and the aim of the optimization algorithm is to produce target minimum distance trajectories for its landing. As can be seen from Figure 2, the M-GA meta-heuristic optimization algorithm manages to generate smaller $x(t)$ and $y(t)$ translational positions for the quadrotor under the dynamic, kinematic and actuator constraints. However, both optimization algorithms plan exactly the same $z(t)$ translational position that converges the target faster than the other translational positions. The proposed minimum distance algorithms in this paper are also able to produce translational velocities for the quadrotor as in Figure 3.

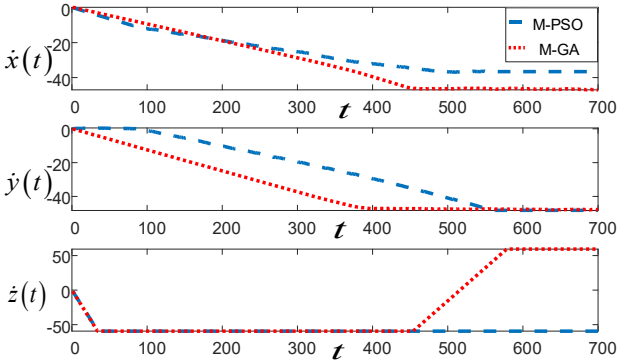

Figure 3: Minimum distance translational velocities.

The M-GA meta-heuristic optimization algorithm produces larger translational velocities which yield smaller translational positions as illustrated by Figure 2. Even though all the generated target trajectories in Figure 3 have similar characters, the $\dot{z}(t)$ translational velocity produced by the M-GA algorithm converges to 50 m/s, the one produced by the M-PSO algorithm converges to -50 m/s, which can occur because of the reference point changes. This paper also plans the minimum time trajectories discussed in next sub-section.

C. *Minimum Time Trajectory Planning*

Figure 4 illustrates the minimum time quadrotor translational positions planned with the M-PSO and M-GA meta-heuristic optimization algorithms.

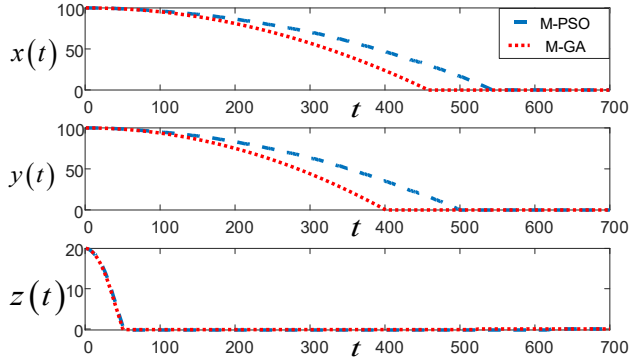

Figure 4: Minimum time translational positions.

Similar to the minimum distance translational positions and velocities provided in Section V, the M-GA meta-heuristic optimization algorithm plans the optimum trajectories in a smaller time. In addition, the overall characters of the trajectories are identical except the time as addressed in Section V. It is important to note that the planned paths are produced under the constraints introduced in Section II.B and relaxing or hardening them yield different paths. The proposed algorithms can also consider the obstacle avoidance problems by just adding a 3D position constraint in the costs given by Equations (23) and (25) if the positions of the obstacles are available. Since detecting the positions of the obstacles in real-time applications require a camera or a sensor, this paper does not directly address the obstacle avoidance problem. The minimum time trajectory planning algorithm produces the corresponding translation velocities given in Figure 5.

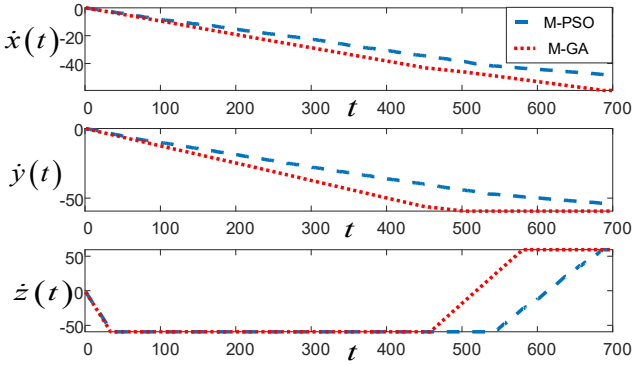

Figure 5: Minimum time translational velocities.

In contradistinction to minimum distance translation velocities in Figure 3, the minimum time translational $\dot{z}(t)$ velocities have consistent characters. The proposed trajectory planning algorithms in this paper can generate the corresponding Euler angles and rotational positions presented in next sub-section.

D. *Euler Angles and Rotational Trajectories*

Figure 6 shows the minimum time quadrotor Euler angles.

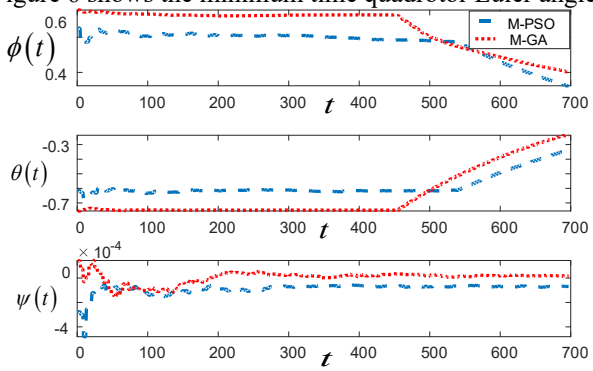

Figure 6: Minimum time Euler angle trajectories.

The corresponding Euler angles for the given target translational positions are selected by the M-PSO and M-GA meta-heuristic optimization algorithms. As can be seen from Figure 6, the M-PSO algorithm initially have slightly larger fluctuations, but later both algorithms produce similar Euler angles for the quadrotors. However, the M-GA algorithm requires shorter times to plan the target trajectories by producing quite larger Euler angles. Similar to the Euler angles, the proposed algorithm in this paper requires the minimum time to plan the rotational position trajectories as presented in Figure 7.

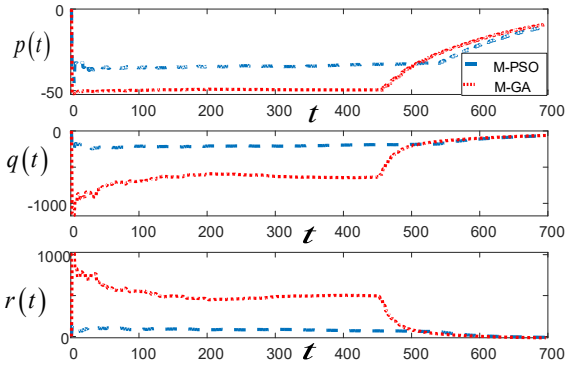

Figure 7: Minimum time rotational position trajectories.

Since the quadrotors must rotate to translate, this paper also generates the rotational trajectories that satisfy the desired translational trajectories. From Figure 7, similar comments such as the M-GA algorithm produces larger rotations to generate faster landing can be made. It is also noticeable that the large rotations yield larger initial overshoots. This is expected since the minimum time trajectory requires aggressive rotational actions. Next sub-section compares the minimum distance and minimum time trajectory planning results with the M-PSO and M-GA meta-heuristic algorithms.

E. Comparison Results

Figure 8 compares the optimized minimum distance and minimum time trajectory costs with the M-PSO and M-GA meta-heuristic algorithms.

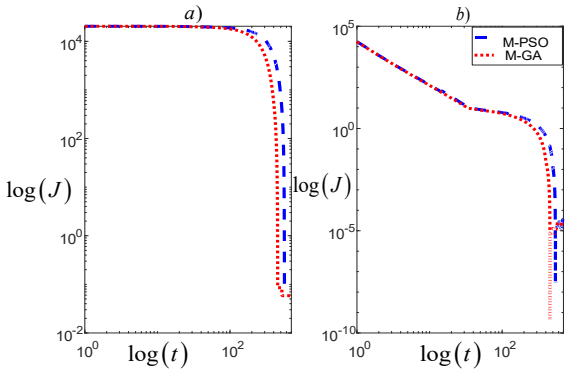

Figure 8: Optimized costs with the, a) minimum distance, b) minimum time algorithms.

As can be seen from Figure 8, the M-PSO algorithm necessaries larger time to learn the optimum trajectories for both the minimum distance and minimum time algorithms. While the M-GA algorithm requires 4.55 and 4.57 seconds for the minimum distance and minimum time paths optimization, the M-PSO algorithm spends 5.71 and 5.42 seconds for the minimum distance and minimum time path optimization. It is clear that the minimum time costs starts reduction quickly because it also considers the inverse of the maximum velocity as in Equation (25). Together with the optimized costs, convergence times can be compared as in Figure 9.

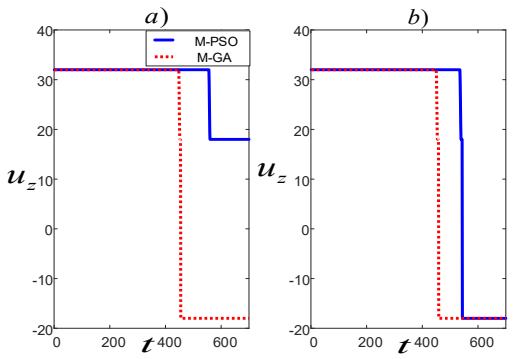

Figure 9: Constrained control signals a) minimum distance, b) minimum time algorithms.

It is clear from Figure 9 that both algorithms generate maximum control signals to drive the quadrotor as desired. As the target trajectories are approached, the control signals switch and converge to either negative or positive minimums. With this property, both approaches exhibit bang-bang control character as expected.

F. Minimum Time Trajectories for Impaired Quadrotor

Figure 10 shows the minimum time translational trajectories planned with the M-GA algorithm for the impaired quadrotor.

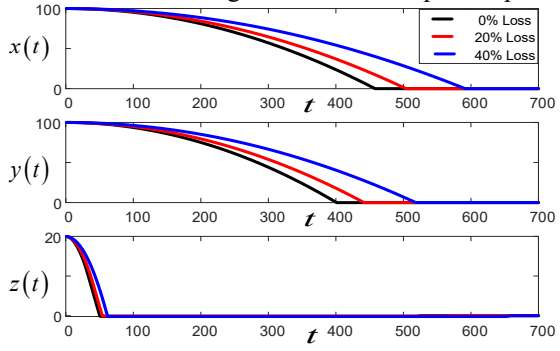

Figure 10: Minimum time translational trajectory planning for the impaired quadrotor.

When the actuator failure occurs, the rotational speeds of the actuators decrease. Therefore, the minimum time trajectory generation algorithm plans new trajectories which require larger times to land on the ground as can be seen from Figure 10. Together with the trajectories, the minimum time trajectory generation algorithm re-constructs the control signal as in Figure 11.

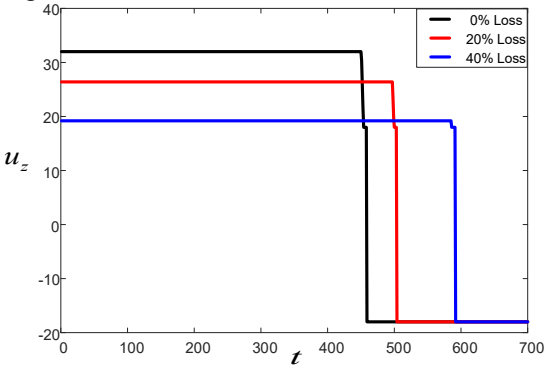

Figure 11: Minimum time control signals generated by the M-GA algorithm for the impaired quadrotor.

The maximum values of the control signal u_z are 32, 26.2, 19.2 when the actuators lose 0%, 20%, and 40% of their rotational speeds. The control signals have larger durations at their maximum values with the increasing actuator failures as illustrated in Figure 11. Next sub-section provides the real-time experimental results.

G. Real-Time Experimental Results

This part of the paper presents the experimental results obtained from the real-time testbed. The testbed is designed to follow the three-dimensional rotation of the frame. For this purpose, as shown in Figure 12, a testbed equipped with Pixhawk autopilot and S500 quadrotor frame has been developed.

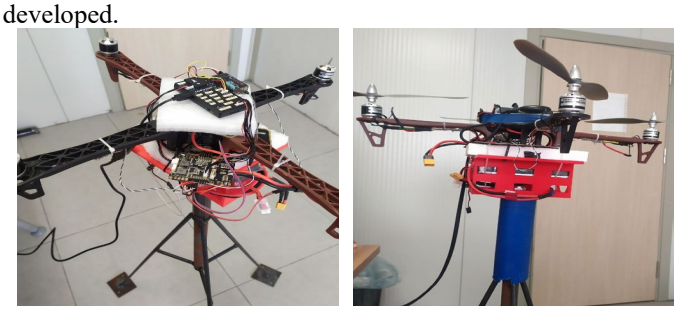

Figure 12: S500 quadrotor and the Pixhawk autopilot.

Figure 13 shows the real-time rotational states in the presence of 15% actuator 1 and actuator 2 failures. As can be seen from Figure 13, the testbed can follow the target Euler angle trajectories generated by optimizing the minimum time cost function in Equation (25). It is important to note that since the testbed is fixed, it generates extra state constraints and disturbances which cause bounded fluctuations around the target trajectories.

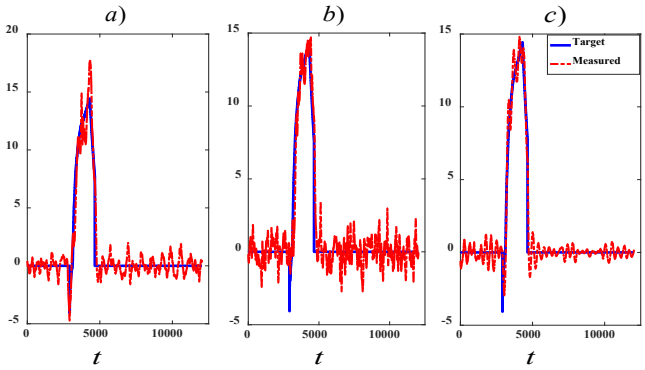

Figure 13: Real time experimental results with 15% actuator 1 and 2 fault, a) $\phi(t)$, b) $\theta(t)$, c) $\psi(t)$.

A further real-time experiment is performed when the actuator 1 has 30% fault and the generated Euler trajectories are shown in Figure 14. It is clear that the generated target path for $\phi(t)$ and $\theta(t)$ are closely followed, but target path for $\psi(t)$ has a convergence after a long oscillatory transient. This is possibly due to the fixed quadrotor frame which is not directly considered in the simulation-based training.

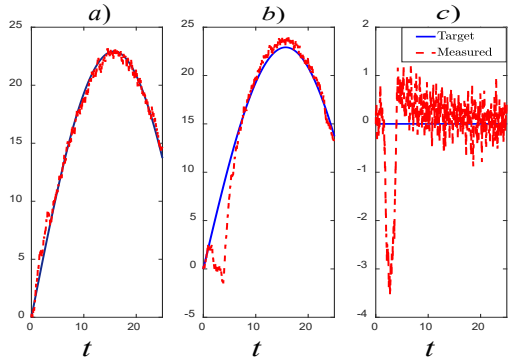

Figure 14: Real time experimental results with 30% actuator 1 fault, a) $\phi(t)$, b) $\theta(t)$, c) $\psi(t)$.

Next section summarizes the paper and states the future research directions.

V. Conclusion and Future Works

This paper developed multi-dimensional minimum distance and minimum time path planning algorithms for the quadrotors impaired because of the actuator failures. Constrained and multi-dimensional M-PSO and M-GA meta-heuristic optimization algorithms optimized the cost functions by selecting the best quadrotor Euler angles. The specified Euler angles were utilized to plan the corresponding translational and rotational positions and velocities. The results showed that both meta-heuristic algorithms plan the required quadrotor paths. However, the M-GA algorithm managed to plan the shortest quadrotor distances and times. It was also demonstrated that the designed algorithms properly handled the quadrotor impairments and modified the planned paths based on the amount of the failures. Simulation and real time experiments were performed under equal conditions and it was showed that the quadrotor followed the simulation paths closely as desired. In the future, these algorithms will be enhanced to coordinate multiple quadrotors by considering the locations of the other UAVs, their battery levels, obstacles and the specified UAV missions.

Funding

This research is supported by the Scientific and Technological Research Council of Turkey (TBİTAK) under 1002 program, with project number [122E641].

Authors' biographies

Dr. Onder Tutsoy is an Associate Professor at ATU specialized in robotics, control, artificial intelligence and object recognition. He did his M.Sc. and Ph.D. in Electrical and Electronic Engineering at The University of Manchester, UK.

Dr. Davood is an Assistant Professor at ATU, Department of Aerospace Engineering, Adana, Turkey. He received a BSc and MSc in Aerospace Engineering from Sharif University in 2009, and a Ph.D. in 2014, from Amirkabir University of Technology.

Dr. Yaser Nabavi is a Postdoctoral researcher in Smart Flight System Lab. in ATU, Turkey. He received his BSc, MSc, and Ph.D. from Sharif University of Technology respectively in 1999, 2001, and 2017 in Aerospace engineering.

Dr. Karim Ahmdi is working as a Postdoctoral researcher in Smart Flight system Lab. in ATU, Department of Aerospace Engineering, Adana, Turkey. He has been graduated from Tehran K. H. Toosi University in M.S and Azad university science and research branch in Ph.D.

Dr. Jamshed Iqbal is working as a Senior Lecturer at the University of Hull, UK where he is also looking after BEng/MEng Robotics and AI program. With over two decades of professional experience, he is a Senior Fellow of Advance HE, UK.

VI. References

1. Madridano, Á., Al-Kaff, A., Martín, D. and Escalera, A., 2021. Trajectory planning for multi-robot systems: Methods and applications. *Expert Sys. with App.*, 173, p.114660.
2. Zuo, Y., Tharmarasa, R., Jassemi-Zargani, R., Kashyap, N., Thiyagalingam, J. and Kirubarajan, T., 2020. MILP Formulation for Aircraft Path Planning in Persistent Surveillance. *IEEE Tran. on Aerospace and Electronic Sys.*, 56(5), pp.3796-3811.
3. Song, J., Song, T., Seo, D., Jin, D. and Kim, J., 2017. Social Big Data Analysis of Information Spread and Perceived Infection Risk During the 2015 Middle East Respiratory Syndrome Outbreak in South Korea. *Cyb., Behavior, and Social Networking*, 20(1), pp.22-29.
4. Watanabe, S. and Mukai, M., 2021. Optimal trajectory generation of a drone for wheelchair tracking using mixed-integer programming. *Art. Life and Rob*, 27(1), pp.159-164.
5. Letizia, N., Salamat, B. and Tonello, A., 2021. A Novel Recursive Smooth Trajectory Generation Method for UAV. *IEEE Trans. on Robotics*, 37(5), pp.1792-1805.
6. Adhikari, M. and de Ruiter, A., 2020. Online Feasible Trajectory Generation for Collision Avoidance in Fixed-Wing Unmanned Aerial Vehicles. *Journal of Guidance, Control, and Dynamics*, 43(6), pp.1201-1209.
7. Guo, X., Ji, M., Zhao, Z., Wen, D. and Zhang, W., 2020. Global path planning and multi-objective path control for unmanned surface vehicle based on modified particle swarm optimization. *Ocean Engineering*, p.107693.
8. Ischuk, I., 2021. Modeling of the Optimal Flight Route of UAV Based on Infrared Video Navigation Data Based on the Upgraded Dijkstra Algorithm. *J. of Siberian Federal University. Engineering Technologies*, 14(7), pp.788-802.
9. Mac, T., Copot, C., Tran, D. and Keyser, R., 2017. A hierarchical global path planning approach for mobile robots based on multi-objective particle swarm optimization. *Applied Soft Computing*, 59, pp.68-76.
10. Sun, G., Zhou, R., Di, B., Dong, Z. and Wang, Y., 2019. A Novel Cooperative Path Planning for Multi-robot Persistent Coverage with Obstacles and Coverage Period Constraints. *Sensors*, 19(9), p.1994.
11. Celestini, D., Primatesta, S., Copello, E., 2022. Trajectory Planning for UAVs Based on Interferred Fluid Dynamical Systems and Bezier Curve. *IEEE Rob. and Aut.*, 7(4), pp.9620-9626.
12. Cryan, M., 2006. Michael Mitzenmacher and Eli Upfal. Probability and computing: Randomized algorithms and probabilistic analysis. Cambridge Uni. Press, Cambridge, *Bulletin of Symbolic Logic*, 12(2), pp.304-308.
13. Primatesta, S., Pogliona, A., Guglieri, G., and Rizzo, A., 2021. Model Predictive Sample-based Motion Planning for Unmanned Aircraft Systems. *Int. Con. on Unmanned Aircraft Systems*, pp.1-9.
14. Schmid, L., Pantic, M., Khanna, R., Ott, L., Siegwart, R. and Nieto, J., 2020. An Efficient Sampling-Based Method for Path Planning in Unknown Environments. *IEEE Robotics and Automation Letters*, 5(2), pp.1500-1507.
15. Madridano, Á., Al-Kaff, A., Martín, D. and de la Escalera, a., 2020. 3D Trajectory Planning Method for UAVs Swarm in Building Emergencies. *Sensors*, 20(3), p.642.
16. Duan, H. and Huang, L., 2014. Imperialist competitive algorithm optimized artificial neural networks for UCAV global path planning. *Neurocomputing*, 125, pp.166-171.
17. Sung, I., Choi, B. and Nielsen, P., 2021. On the training of a neural network for online path planning with offline path planning algorithms. *Int. J. of Inf. Man.*, 57, p.102142.
18. Nikolos, I.K., E.S. Zografos, and A.N. Brintaki, *UAV path planning using evolutionary algorithms*, in *Innovations in intelligent machines-1*. 2007, Springer. p. 77-111.
19. Li, J., Deng, G., Luo, C., Lin, Q., Yan, Q. and Ming, Z., 2016. A Hybrid Path Planning Method in Unmanned Air/Ground Vehicle Cooperative Systems. *IEEE Trans. on Vehicular Technology*, 65(12), pp.9585-9596.
20. Consolini, L., Locatelli, M., Minari, A. and Piazzi, A., 2017. An optimal complexity algorithm for minimum-time velocity planning. *Systems Control Letters*, 103, pp.50-57.
21. Zhang, Q., Li, S. and Guo, J., 2012. Smooth time-optimal tool trajectory generation for CNC manufacturing systems. *J. of Manufacturing Systems*, 31(3), pp.280-287.
22. Kim, J., Kim, S., Kim, S. and Kim, D., 2010. A practical approach for minimum-time trajectory planning for industrial robots. *Industrial Robot: An Int. J.*, 37(1), pp.51-61.
23. Stefanatos, D. and Li, J., 2014. Minimum-Time Quantum Transport With Bounded Trap Velocity. *IEEE Tran. on Automatic Control*, 59(3), pp.733-738..
24. Techy, L. and Woolsey, C., 2009. Minimum-Time Path Planning for UAV in Steady Uniform Winds. *Journal of Guidance, Control, and Dynamics*, 32(6), pp.1736-1746.