2019 27th European Signal Processing Conference (EUSIPCO)

A Modelling Approach to Generate Representative UAV Trajectories Using PSO

Babak Salamat
Chair of Embedded Communication Systems
Alpen-Adria-Universität Klagenfurt
Klagenfurt, 9020, Austria
babaksa@edu.aau.at

Andrea M. Tonello
Chair of Embedded Communication Systems
Alpen-Adria-Universität Klagenfurt
Klagenfurt, 9020, Austria
andrea.tonello@aau.at

***Abstract*—We propose a trajectory generation algorithm (STGA) that represents realistically and stochastically trajectories followed by unmanned air vehicles (UAVs), in particular quadrotors UAVs. It is meant to be a tool for testing localization, state estimation and control algorithms. We propose to firstly model a number of representative flight scenarios. For each scenario, stochastic trajectories are generated. They follow a parametric non-linear model whose parameters are determined using a multi-objective evolutionary optimization method called particle swarm optimization (PSO). Numerical results are reported to verify feasibility in comparison to pure random unconstrained trajectory algorithm.**

***Index Terms*—Stochastic trajectory generation, unmanned air vehicles (UAVs), particle swarm optimization.**

I. Introduction

Electrically powered autonomous air vehicles are getting high attention for several applications. In particular, quadrotors offer mechanical simplicity and maneuverability. A key component is the ability to track the position of the UAV and more, in general, its full state (orientation, velocity, acceleration, and attitude). Often, the literature on radio localization considers the UAV to be a point mass and focuses only on estimating the position of the UAV. However, to enable the development of control algorithms [1], [2] full state is needed. Not only the position and the evolution of the position (the center of gravity of the UAV) is relevant, but also the orientation of the UAV (the pitch, roll, and yaw angles) must be considered when we want to test the inertial navigation system algorithms or sensor fusion algorithms [3]–[5]. To the authors knowledge, there is no recognized simulator that is able to generate statistically representative trajectories.

In this paper, a non-linear model and its evolution in realistic representative scenarios is proposed. In each scenario, stochastic trajectories are generated under certain mechanical and environmental constraints that can be imposed with the given initial state (IS) and desired state (DS) in three-dimensional coordinates. Indeed, having a point mass moving randomly in a three-dimensional space is not enough if a realistic flight scenario and UAV dynamics have to be considered. In particular, there exist several methods to generate trajectories, including B-splines [6], polynomial [1] and machine learning approaches [7]. In order to provide a method that can represent the variability of the ensemble of flight trajectories followed by a UAV, it is of interest to follow a stochastic approach. In this respect, the Confined Area Random Aerial Trajectory Emulator (CARATE) proposed in [8], stochastically generates a 3D path obtained from a variable length previous history of the trajectory and a tunable set of random variables. CARATE is specifically designed to emulate stochastic trajectories with a limited flight area. However, it does not consider the dynamics of the UAV and it is not optimized to cope with harsh environmental constraints. In this paper, we tackle the limitations above and we propose a stochastic trajectory generation algorithm (STGA) that: (1) takes into account the dynamics of a quadrotor UAV, and (2) copes with environmental constraints. The algorithm is derived from a general framework where trajectories are obtained from the solution of a multi objective optimization problem solved via particle swarm optimization. Considering several performance indices for our framework provides more flexibility and adaptability to represent the expected behavior of the trajectories in ways that are difficult to express otherwise. To validate the proposed method, three representative scenarios are defined:

- Hovering
- Take-off and landing
- Representative aggressive trajectory.

The rest of the paper is organized as follows. In Section I, a non-linear model and its evolution is given. The three representative scenarios are described in Section II. Section III presents the trajectory realization methodology and simulation results. Finally, the conclusions then follow.

II. UAV Non-linear Model

To begin with, we consider the non-linear model of a quadrotor UAV described in [6] whose dynamics equations read as follows

$$
\begin{aligned}
\ddot{x} &= u_{1x} \\
\ddot{y} &= u_{1y} \\
\ddot{z} &= u_1(\cos(\theta)\cos(\psi)) - g \\
\ddot{\psi} &= u_2 + \dot{\phi}\dot{\theta}a_1 \\
\ddot{\theta} &= u_3 + \dot{\phi}\dot{\psi}a_2 \\
\ddot{\phi} &= u_4 + \dot{\psi}\dot{\theta}a_3
\end{aligned} \tag{1}
$$

where $a_1 = \left(\frac{I_y - I_z}{I_x}\right)$, $a_2 = \left(\frac{I_z - I_x}{I_y}\right)$, $a_3 = \left(\frac{I_x - I_y}{I_z}\right)$ and $u_{1x} = u_1(\cos(\phi)\sin(\theta)\cos(\psi) + \sin(\phi)\sin(\psi))$ and $u_{1y} = u_1(\sin(\phi)\sin(\theta)\cos(\psi) - \cos(\phi)\sin(\psi))$. $\boldsymbol{\xi} = [x, y, z]^T$ is the position of the quadrotor helicopter center of gravity in the inertial frame and $\boldsymbol{\Theta} = [\phi, \theta, \psi]^T$ is the attitude (roll, pitch and yaw). I_i $(i = x, y, z)$ are the moments of inertia along the x, y and z directions. u_1 is the total force generated by the four rotors and directly related to the altitude in the z direction. u_2, u_3 and u_4 are related to the yaw, pitch, and roll motion respectively. The quantities $\boldsymbol{u} = [u_1, u_2, u_3, u_4]^T$ are the control inputs (control signals) of our dynamical system model.

A. Simplified Model

By assuming small angles ϕ and θ and a constant yaw angle ψ, for instance $\psi = 0$, (1) can be written as

$$
\begin{aligned}
\ddot{x} &= u_1\theta \\
\ddot{y} &= -u_1\phi\theta \\
\ddot{z} &= u_1 - g \\
\ddot{\psi} &= u_2 \\
\ddot{\theta} &= u_3 \\
\ddot{\phi} &= u_4.
\end{aligned} \tag{2}
$$

It can be seen that the dynamical system model (2) fulfills the flatness property [9]. Therefore, it is possible to express all quantities of the stochastic trajectory generation algorithm (STGA) as a function of the time evolution of only four parameters $\boldsymbol{F}(t) = (x(t), y(t), z(t), \phi(t))$ and their time derivatives.

III. Modeling Three Representative Flight Scenarios

The idea of developing a stochastic trajectory generation algorithm can appear a formidable task since the UAV can follow a different number of trajectories and be in different flying environments. Therefore, we propose to consider certain scenarios that can be representative of realistic flight missions. In each scenario, the reference trajectory evolution $\boldsymbol{F}(t)$ is represented with piecewise smooth polynomials of order n over m time intervals with limits $\{t_0, t_1, ..., t_m\}$. One element

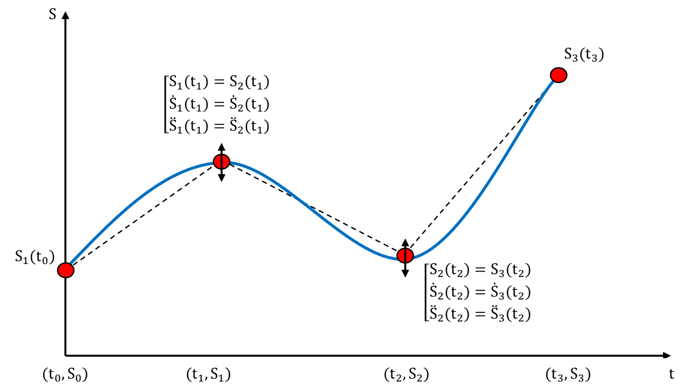

Fig. 1. Evolution of the piecewise continuous polynomial trajectory. Red circles are the waypoints.

out of $\boldsymbol{F}(t)$ of the piecewise continuous trajectory can be written as:

$$
S(t) = \begin{cases}
\sum_{i=0}^{n} \alpha_{1i} t^i & t_0 \le t \le t_1 \\
\sum_{i=0}^{n} \alpha_{2i} t^i & t_1 \le t \le t_2 \\
\vdots & \vdots \\
\sum_{i=0}^{n} \alpha_{m-1i} t^i & t_{m-1} \le t \le t_m
\end{cases} \tag{3}
$$

where α_{ji} is the i^{th} order polynomial coefficient of the j^{th} segment. We assume that we know the position, velocity and acceleration at the start mission and at the end mission. Our dynamical system model (2) is a second order system which means that the control inputs are directly related to the accelerations. Therefore, we need to specify the minimum acceleration trajectory (realistic trajectory) with multiple intermediate points (waypoints). Each one of the waypoints is defined with coordinates in the search space, which are $\{x(t), y(t), z(t), \phi(t)\}$ and each of them is generated stochastically as a uniform random variable ranging between a lower and an upper limit. The waypoints are distributed over time by the user. This configuration describes a stochastic behaviour for $\boldsymbol{F}(t)$. Indeed, by changing the location of the waypoints in the search space, a new trajectory can be realized (see Fig. 1). According to Fig. 1, we highlight that there must be continuity between segments. We impose that the velocity and acceleration of a given first segment match with the beginning of the adjacent segment. This is done for all segments, so that we obtain $4m$ constraints to determine the coefficients of the piecewise polynomial where in our setup $m = 3$.

In the following, we describe three significant and representative flying scenarios: hovering, take-off and landing with the associated trajectories.

A. Scenario 1: Hovering

Hovering is modeled by a vertical altitude with zero roll, pitch, and yaw angles. One can assume that in hovering condition, $u_1 \approx g$ in the x and y directions, therefore, (2) becomes

$$\begin{aligned} \ddot{x} &= g\theta \\ \ddot{y} &= -g\phi\theta \\ \ddot{z} &= u_1 - g \\ \ddot{\psi} &= u_2 \\ \ddot{\theta} &= u_3 \\ \ddot{\phi} &= u_4. \end{aligned} \tag{4}$$

For this scenario a third order polynomial (cubic-spline) with two waypoints is considered. More details on hovering can be found in the next section.

B. Scenario 2: Take-off and Landing

In all kind of UAVs, take-off and landing is important. The challenge in modeling this scenario is the tuning of the parameters of the cubic-spline in a way that: (1) the velocity profiles of the trajectories during take-off and landing are zero (rest-to-rest maneuver), (2) the absolute value of the velocity is less than $4m/s$, (3) the Euler angles $(\phi(t), \theta(t), \psi(t))$ during take-off and landing are zero (rest-to-rest manoeuvre), and (4) the UAV reaches the specific altitude z during the flight time. Clearly, other constraints can be defined. For simplicity, the surface of the ground is assumed to be horizontal. The length and duration of the trajectory between take-off and landing can be made variable. Further details about modeling this scenario are reported in the next section.

C. Scenario 3: Representative Aggressive Trajectory

The third considered scenario highlights the ability of our algorithm to model trajectories that comprise obstacle avoidance procedures. Static obstacles with different size can be defined in the search space. In particular, we consider four obstacles. Finding a safe and short trajectory in the search space with obstacles becomes a challenging problem in path planning. On the other hand, according to the configuration of our UAV, the feasibility of these trajectories is also significant.

IV. Trajectory Realization and Numerical Generator

In this section, firstly, the definition of the trajectory candidates will be given. Then, the constraints of the control inputs generated by UAV will be discussed. Finally, examples of realized trajectories are reported.

A. Trajectory Generation

As discussed previously, the trajectory evolution is $\boldsymbol{F}(t) = (x(t), y(t), z(t), \phi(t))$ with control inputs $\boldsymbol{u}(t) = [u_1(t), u_2(t), u_3(t), u_4(t)]^T$. Also, the dynamical system model of a UAV and environmental constraints add mathematical constraints to the generation of feasible UAV trajectories.

In the following the constraints will be defined. The dynamical system model (2) is a second-order system which means we have to apply inputs which are algebraically related to the accelerations. Therefore, our problem is to determine the optimal coefficients of the piecewise polynomial α_{ji} under particular constraints. The cost function and the boundary constraints for our scenarios are defined as follows

TABLE I
Setup parameters of the quadrotor helicopter

Parameters	Symbol	Value	Unit
Mass	m	0.65	Kg
Inertia around x axis	I_{xx}	0.07582	$Kg \bullet m^2$
Inertia around y axis	I_{yy}	0.07582	$Kg \bullet m^2$
Inertia around z axis	I_{zz}	0.1457924	$Kg \bullet m^2$
Gravitational acceleration	g	9.806	m/s^2
Lagrange multiplier	β	1000	-

$$\begin{aligned} &\min_{S(t)} \left[\int_{t_0}^{t_1} \ddot{S}^2 + ... + \int_{t_{m-1}}^{t_m} \ddot{S}^2 \right] dt \\ &subject\ to \\ &u_k^{min} \le u_k^t \le u_k^{max} \quad k = (1, ..., 4). \end{aligned} \tag{5}$$

The constraint in (5) can be defined through a violation function in the following form:

$$\upsilon_{u_k} = \max\left(\frac{u_k}{|u_{limit}|} - 1, 0 \right), \quad k = (1, ..., 4). \tag{6}$$

The violation function (6) is used in the following relaxed optimization problem

$$\min_{S(t)} \left(\left[\int_{t_0}^{t_1} \ddot{S}^2 + ... + \int_{t_{m-1}}^{t_m} \ddot{S}^2 \right] dt \right) \left(1 + \beta < \upsilon_{u_k} > \right) \tag{7}$$

where $< . >$ is the mean operator and β is the Lagrange multiplier. See [7] for an elegant extension of (7) to a collision-free trajectory.

Basically, we propose to solve (7) by combining an evolutionary algorithm to evolve possible trajectory candidates and third order piecewise polynomials to obtain a specific trajectory given a number of waypoints $\{(t_0, S_0), ..., (t_m, S_m)\} \subset \mathbb{R} \times \mathbb{R}$ in the search space. More in detail each one of the waypoints is defined with coordinates in the search space, which are $(x_w(t_i), y_w(t_i), z_w(t_i), \phi_w(t_i))$ $i = 0, ..., m$ and each of them is generated stochastically as a uniform random variable ranging between a lower and an upper limit. This configuration enables a stochastic behaviour of $\boldsymbol{F}(t)$. Finally, we keep evolving the trajectories by generating waypoints through the particle swarm optimization (PSO) technique such that the cost function of the PSO in (7) is minimized to yield the global best of the particles [10].

B. Numerical Generator

We report a summary of model parameters to be used to generate trajectories in the three considered scenarios. The initial parameters are fixed as shown in Table I. Furthermore, in each representative scenario the start point, end point and a number of waypoints are defined by the user. The details of each scenario are given in Section II.

TABLE II
SETUP PARAMETERS OF THE QUADROTOR HELICOPTER

Scenarios	Waypoints	Swarm Size	Iterations	Execution Time (s)
Scenario 1	2	10	30	2.25
Scenario 2	2	10	30	3.21
Scenario 3	3	5	40	4.4587

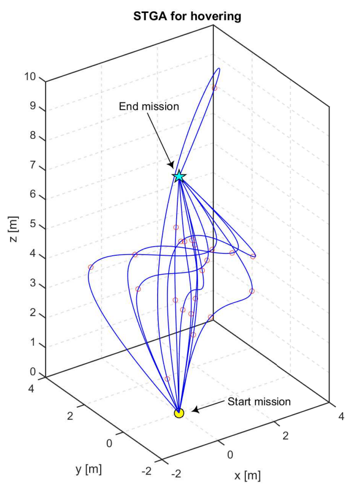

Fig. 2. Stochastic realization of trajectories for hovering. Red circles show the position of waypoints.

1) Hovering: Fig. 2 shows a number of realizations of possible trajectories generated by STGA. The initial and final position can be chosen by the user. In this specific realization the initial position of our UAV is $\boldsymbol{F}_{initial} = (0, 0, 0, 0)$ in meters. The final position is $\boldsymbol{F}_{final} = (0, 0, 8, 0)$. The third order polynomial is considered for this scenario.

2) Take-off and landing: In Fig. 3, we report an example of trajectory belonging to the scenario 2. In particular, the starting

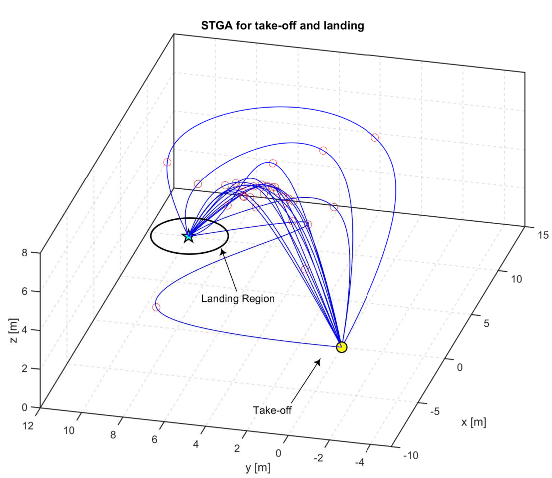

Fig. 3. Stochastic realization of trajectories for take-off and landing.

position is $\boldsymbol{F}_{initial} = (0, 0, 0, 0)$ and the landing region is $\boldsymbol{F}_{final} = (10, 10, 0, 0)$. The desired altitude during the flight is satisfied and the value is $5m$. Fig. 4 represents the inertial frame for the velocity and acceleration components which for the velocity is less than $4\frac{m}{s}$ according to our constraint definition in scenario two and the total flight time is $12s$.

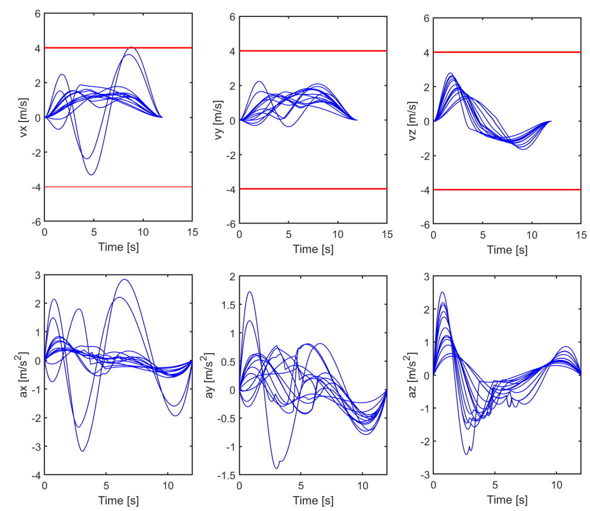

Fig. 4. trajectories of velocity and acceleration for take-off and landing.

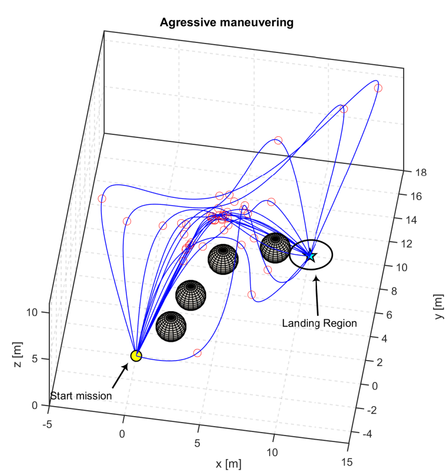

Fig. 5. Stochastic complex maneuvering trajectories in 3D space.

3) Representative aggressive trajectory: In Fig. 5, the search space is populated with four generic spherical objects that acts as obstacles for the UAV and thus hinder their path. These obstacles can be either positioned arbitrarily by the user or can be randomly generated by the algorithm. In this specific case they were generated by the user. A stochastic realization as a set of trajectories is shown in Fig. 5. The start point is $\boldsymbol{F}_{initial} = (0, 0, 0, 0)$ and the end point is $\boldsymbol{F}_{final} = (10, 10, 0, 0)$. Simulation results for the velocities and accelerations are shown in Fig. 6. All these trajectories are feasible and avoid obstacles. Our approach is able to generate feasible trajectories also in cluttered environments. Furthermore, the initial and final velocities and accelerations are zero (rest to rest maneuver) that is important for take-off and landing.

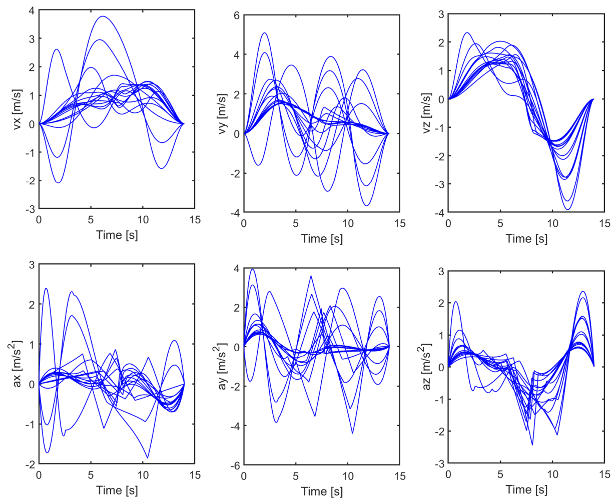

Fig. 6. Evolution of velocity and acceleration for aggressive maneuvering.

4) STGA vs CARATE: As already mentioned before, to illustrate the effectiveness of the proposed method, we consider the baseline CARATE methodology [8] to obtain a path in 3D space. The total time of simulation is $100s$. As shown in Fig. 7 and Fig. 8, the results for the proposed method show that it generates smooth trajectories and fulfills the constraints. Despite the ability of CARATE to generate 3D trajectories, the obtained dynamics are not feasible since for instance the acceleration is too high. Furthermore, CARATE treats the UAV as a point mass, therefore it does not allow to fully model the UAV state evolution.

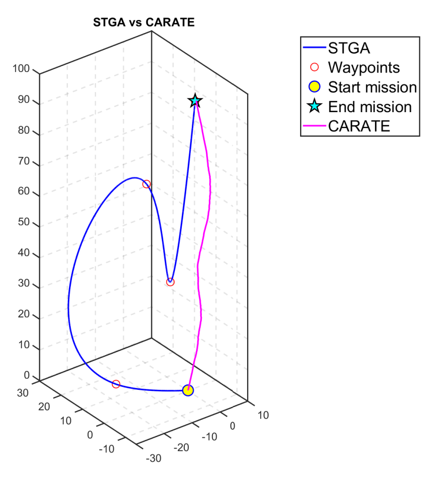

Fig. 7. STGA vs CARATE [8].

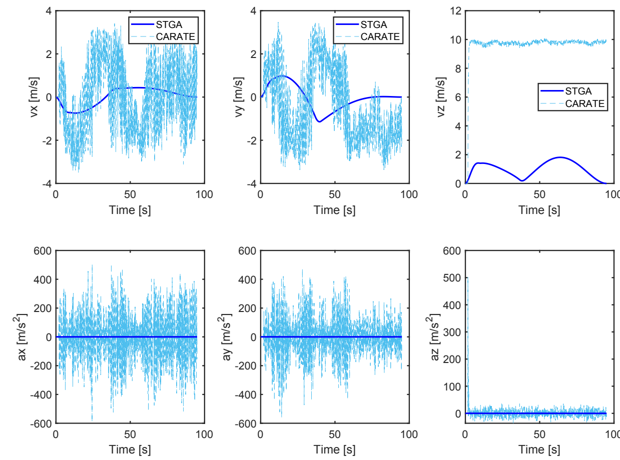

Fig. 8. Translation of velocities and acceleration for STGA vs CARATE.

V. Conclusion

We have proposed a stochastic trajectory generator in the domain of UAVs that generates representative trajectories followed by UAVs. It enables simulation and testing for instance of state estimation and localization algorithms. The method allows not only to simulate the position of the UAV but also it provides the orientation (roll, pitch and yaw) of the UAV. This is also important when inertial navigation system (INS) algorithms have to be tested.

References

[1] B. Salamat and A. M. Tonello, “Novel trajectory generation and adaptive evolutionary feedback controller for quadrotors,” in *2018 IEEE Aerospace Conference Big Sky, Montana, USA*, March 2018.

[2] ——, “Adaptive nonlinear PID control for a quadrotor UAV using particle swarm optimization,” in *2019 IEEE Aerospace Conference Big Sky, Montana, USA*, March 2019.

[3] A. Nemra and N. Aouf, “Robust INS/GPS sensor fusion for UAV localization using SDRE nonlinear filtering,” *IEEE Sensors Journal*, vol. 10, no. 4, pp. 789–798, April 2010.

[4] A. Tonello and V. Pathuri Bhuvana, “Exploiting antenna arrays for position tracking of unmanned aerial vehicles,” in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, March 2018.

[5] J. Zhang, Y. Wu, W. Liu, and X. Chen, “Novel approach to position and orientation estimation in vision-based UAV navigation,” *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 687–700, April 2010.

[6] B. Salamat and A. M. Tonello, “Stochastic trajectory generation using particle swarm optimization for quadrotor unmanned aerial vehicles (uavs),” *Aerospace*, vol. 4, no. 2, 2017. [Online]. Available: http://www.mdpi.com/2226-4310/4/2/27

[7] B. Salamat and A. M. Tonello, “A generalized multi-objective framework for UAV mission planning,” in *2019 IEEE Aerospace Conference Big Sky, Montana, USA*, March 2019.

[8] A. Papaiz and A. M. Tonello, “Azimuth and Elevation Dynamic Tracking of UAVs via 3-Axial ULA and Particle Filtering,” *International Journal of Aerospace Engineering*, vol. 2016, pp. 1–9, September 2016.

[9] D. J. Cooke, “Optimal trajectory planning and LQR control for a quadrotor UAV,” in *UKACC International Conference on Control*, 2006.

[10] J. Kennedy, “The particle swarm: social adaptation of knowledge,” in *Proceedings of 1997 IEEE International Conference on Evolutionary Computation (ICEC ’97)*, April 1997.