JAST, Vol. 12, No. 2, pp. 19-33

JAST
Journal of Aerospace Science and Technology

Automatic Tuning of a Behavior-Based Guidance Algorithm for Formation Flight of Quadrotors

E. Zibaei[1], M.A. Amiri Atashgah[2]*, A. Kalhor[3]

1,2Faculty of New Sciences and Technologies, University of Tehran, Tehran, Iran

3. School of Electrical Engineering, University of Tehran, Tehran, Iran

* Postal Code: 14395-1561 Tehran, Iran,

atashgah@ut.ac.ir

This paper presents a tuned behavior-based guidance algorithm for formation flight of quadrotors. The behavior-based approach provides the basis for the simultaneous realization of different behaviors such as leader following and obstacle avoidance for a group of agents; in our case they are quadcopters. In this paper optimization techniques are utilized to tune the parameters of a behavior-based guidance algorithm; to compromise between safety, trajectory optimality, and control effort during the formation flight. The tuning is formulated as a constraint optimization problem where the penalty function method is used to secure the safe passage of quadrotors around an obstacle. The guidance subsystem is integrated with a consistent dynamic inversion controller to realize a smooth maneuver of the quadrotors along desired trajectories. For more, MATLAB/Simulink is used as the programming platform. The effectiveness of the tuning method is verified, based on the performance of the closed-loop system in the presence of an overall navigation system uncertainties and actuator lags.

Keywords: Formation Flight; Behavior-based Algorithm; Obstacle Avoidance; Constraint Optimization

IST OFSYMBOLS

Superscripts

n	Inertial/navigation frame
b	Body frame

Subscripts

a	Actual
az	Avoidance zone
d	Desired or differential
dz	Danger zone
D	Deviation
G	Penalty function
hover	Hovering mode
lf	Leader following
o	Obstacle
oa	Obstacle avoidance
ob	Obstacle
p	Proportional
sat	Saturation
sd	Safe distance
tot	Total
U	Control effort
0	Initial

Symbols

b	Lift coefficient
C	Coefficient
d	Drag coefficient
$\mathbf{D}$	Relative position vector
e	Error
g	Gravitational acceleration
G	Penalty function
I_{xx} , I_{yy} , I_{zz}	Moment of inertia around the principal axis
J_r	Rotor's moment of inertia

1. M. Sc.
2. Associate Professor (Corresponding Author)
3. Assistant Professor

J	Cost function
K	Coefficient
l	Moment arm
m	Mass of quadrotor
p , q , r	Rotational speed in the body frame
$\mathbf{R}$	Position vector
U_1	Thrust in z_b direction
U_2	Moment around x_b
U_3	Moment around y_b
U_4	Moment around z_b
x , y , z	Position in an inertial frame
θ	Pitch
τ	The approximate time constant of the rotor
μ_x , μ_y , μ_z	Virtual accelerations
ϕ	Roll
ψ	Yaw
$\Omega_{1,2,3,4}$	Rotors' rotational speed
Ω_r	Resultant of rotors' rotational speed

INTRODUCTION

Formation flight is a kind of multi-agent mission in which agents maintain their distance and orientation towards each other during the mission. In nature, birds perform a V-shaped formation flight to reduce the consumed energy during the flight. Some research has been conducted for the utilization of this aerodynamic phenomenon in the formation flight of a fixed-wing aircraft [1], [2]. Autonomous formation flight was initially proposed for spacecraft and major approaches like leader-follower, virtual structure, and behavior-based were evolved in this field [3]. The formation flight of aerial robots has many applications, making it a popular research topic in recent years. For example, [4] proposed a swarm of quadrotors, to extend the coverage of available Wi-Fi networks over adisaster area. In another research, [5] proposed a framework of path planning and control for a formation of quadrotors to lift objects which cannot be lifted by a single quadrotor. Two properties of quadrotors, namely, high maneuverability and the ability to hover, make them suitable platforms for formation flight missions. High maneuverability lets quadrotors perform obstacle avoidance and collision avoidance maneuvers easily which is critical in the formation flight. Moreover, hovering capability is a very useful property to form desired patterns in the air by multiple robots.

In the formation flight, a flight control framework is needed to generate proper trajectories for robots and enable them to track the trajectories. In this paper, a layered architecture consisting of four layers is proposed (Fig. 1) for the flight control framework. In the first layer, a primary trajectory is produced to move the group from one point to another. The primary trajectory has not to consider obstacles in the path because the next layer will take charge of it. In the second layer, the guidance subsystem considers the desired offsets from the primary trajectory and the safe distance from the obstacles, to produce a reference input for the control subsystem. The third layer is a control subsystem, which tracks the generated trajectory. Finally, in the fourth layer, the actuators of each robot transform the control command signals to proper forces and moments.

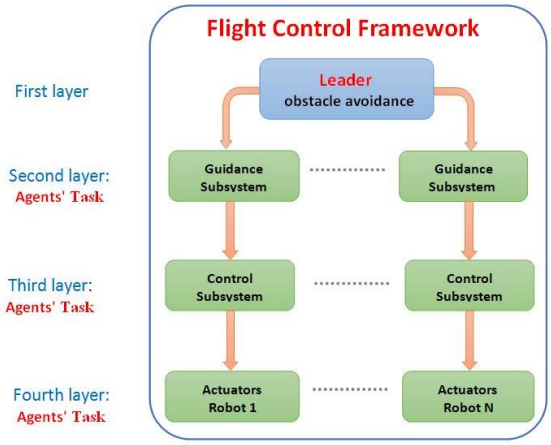

Figure 1. Architecture of the proposed flight control framework for N robots.

Among four layers of a typical flight control framework, the guidance subsystem is the most important part of formation flight missions. Until now, leader-follower, virtual structure, and behavior-based algorithms are the three major approaches that have been developed to guide robots through the formation flight. In the leader-follower approach, a leader robot tracks a primary trajectory and follower robots try to keep their distance and orientation

to the leader robot. This approach has a simple architecture; therefore, many control techniques such as PID [6], Sliding Mode [6], [7], LQR [8], Lyapunov [9] and neural networks [8] have been implemented through it. In the virtual structure approach, the formation is modeled as a rigid body, which leads to aconstant distance of robots to each other during the flight. High accuracy of this approach led to some research on using it for the formation flight of fixed-wing UAVs [10] and spacecraft [11].

In the behavior-based approach, different behaviors such as formation keeping, leader following, obstacle avoidance, and collision avoidance are designated for every robot. Based on different situations that a robot encounters, the guidance subsystem produces a resultant vector of those behaviors. In this approach, every behavior has a weighting factor that can be tuned to increase or decrease its impact on the produced resultant. The formation flight of rotary aerial robots by a behavior-based approach was firstly proposed in [12] by realizing a safe passage of the formation over horizontal obstacles. [13] modeled the group of quadrotors as a large-scale interconnected linear system to determine the boundaries of the behavior-based weighting factors, in which the closed-loop system remains stable. In [14], optimum values for the weighting factors of a behavior-based algorithm are proposed by making a compromise between formation keeping and leader following behaviors.

The formation control was traditionally considered as a problem which integrates leader following and formation keeping behaviors [14], [15]. Recently, there is a shift in attitudes toward this problem, by considering the obstacle avoidance in the formation control. For example, the effects of sensors [16] or communication delays [17] are investigated while the agents have to keep their formation and pass around obstacles simultaneously. Obstacle avoidance makes the formation control problem more complicated. In a behavior-based approach, if the weighting factors of behaviors are not tuned properly, the agents may collide with the obstacles or get far away from their group reference. As a result, an effective tuning algorithm is needed to satisfy both optimality and safety of the trajectories. In this paper, optimization techniques are utilized to tune the parameters of a behavior-based guidance subsystem to reach smooth trajectories around the obstacles. Reference deviation and control effort are significantly reduced while the formation is maintained and obstacles are safely avoided.

More on the prescribed citation, some domestic research is cited as follows. Ashrafi et al. in [18] presented the subject of leader-follower formation control of unmanned aerial vehicles (UAVs) using a PID-fuzzy method. Moreover, they used a nonlinear inverse dynamic control method along with the cooperative guidance system. Another activity on cooperative guidance and control is [19], in which the authors presented a behavioral guidance and control system for a flying robot on the leader-follower concept by incorporation of energy-based collision and obstacle avoidance algorithms. Lastly, Mohammadi et al. [20] conducted a research on formation control and path tracking for a group of quadrotors to carry a suspended load.

The contributions of this work can be categorized into three aspects. First, a layered architecture is proposed for the flight control framework which provides all major requisites of the formation flight of quadrotors in an efficient way. The behavior-based guidance subsystem and dynamic inversion control subsystem are compatible with each other and can move the quadrotors smoothly along predefined trajectories. Second, the tuning of the behavior-based parameters is formulated as a constraint optimization problem and solved using the penalty function method. This technique allows minimizing the reference deviation and control effort while the safety distance to the obstacle is maintained. Third, the tuning is done based on the performance of the closed-loop system consisting of guidance and control subsystems, flight dynamics of quadrotors, navigation system uncertainties, and rotor lags. As a result, the tuned parameters are practical for future real implementations.

The rest of this paper is organized as follows. In section two, the flight dynamics of quadrotors and a control subsystem based on dynamic inversion method are presented. The behavior-based guidance module is presented in section three. Formulating the tuning as a constraint optimization problem is presented in section four. In section five, the flight control framework is validated by numerical

simulations. Finally, the conclusion of the paper is presented in section six.

DYNAMIC MODEL AND CONTROL SUBSYSTEM

To design the control subsystem, the equations of motion of the robot should be derived at first. The most important factors in a dynamic model of a quadrotor are aerodynamic forces and moments on the blades and gyroscopic effects of the body and rotors. Quadrotors usually move at relatively slow velocities; which makes body aerodynamic drag, blade flapping, and airflow perturbation small enough to be ignored. The related coordinate frames of the vehicle, including body and inertial/navigation frames, in a six-degree-of-freedom (6DoF), and rigid dynamic model are depicted in Figure 2.

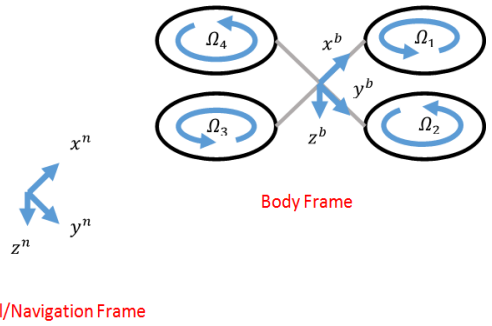

Figure 2. Portray of reference frames in the 6DoF dynamic model.

6DoF Dynamic Model

To increase numerical effectiveness and realization simplicity, the translational dynamics model of a quadrotor is derived in the inertial/navigation frame and rotational dynamics are derived in the body frame. Consequently, the 6DoF dynamical model of quadrotor [21] can be expressed as:

$$\begin{aligned}
\ddot{x} &= \frac{-U_1}{m}(\cos\psi\sin\theta\cos\phi+\sin\psi\sin\phi)\\
\ddot{y} &= \frac{-U_1}{m}(\sin\psi\sin\theta\cos\phi-\cos\psi\sin\phi)\\
\ddot{z} &= g-\frac{U_1}{m}(\cos\theta\cos\phi)\\
\dot{p} &= qr(\frac{I_{yy}-I_{zz}}{I_{xx}})+\frac{J_r q\Omega_r}{I_{xx}}+\frac{U_2}{I_{xx}}\\
\dot{q} &= pr(\frac{I_{zz}-I_{xx}}{I_{yy}})-\frac{J_r p\Omega_r}{I_{yy}}+\frac{U_3}{I_{yy}}\\
\dot{r} &= qp(\frac{I_{xx}-I_{yy}}{I_{zz}})+\frac{U_4}{I_{zz}}
\end{aligned} \tag{1}$$

In which, $\boldsymbol{R}=[x,y,z]$ is the quadrotor position in the inertial/navigation frame. Rotational velocity components expressed in the body frame are denoted by $[p,q,r]$. Euler angles are denoted by $\boldsymbol{\Theta}=[\varphi,\theta,\psi]$. $[I_{xx},I_{yy},I_{zz}]$ are principal moments of inertia. J_r and Ω_r are rotor inertia and residual propeller angular speed, respectively. Furthermore, U_1 denotes astotal thrust, and U_2 to U_4 are moments produced by actuators around the body axes. Using a simple aerodynamic model, U_1 to U_4 and Ω_r can be computed as noted below:

$$\begin{aligned}
U_1 &= b(\Omega_1^2+\Omega_2^2+\Omega_3^2+\Omega_4^2)\\
U_2 &= bl(-\Omega_2^2+\Omega_4^2)\\
U_3 &= bl(\Omega_1^2-\Omega_3^2)\\
U_4 &= d(-\Omega_1^2+\Omega_2^2-\Omega_3^2+\Omega_4^2)\\
\Omega_r &= -\Omega_1+\Omega_2-\Omega_3+\Omega_4
\end{aligned} \tag{2}$$

Where Ω_1 to Ω_4 are angular velocities of four rotors and l is the moment arm. b and d are the aerodynamic lift and drag coefficients respectively, which are functions of propeller surface characteristics, propeller radius, and air density. An important issue about modeling of rotors is the lag, which conveys a negative impact on the stability of the quadrotor. Rotors' lag is due to mechanical and electrical factors, which can be modeled as a second-order transfer function. Provided that time constant of mechanical lag is much more than the electrical one, the lag can be approximated by a first-order transfer function as:

$$\Omega(s)=\frac{1}{\tau s+1} \tag{3}$$

Where τ is the approximate time constant of the rotor. As motor and propeller size increase, time constant of the rotor increases too.

Because in most cases there is no need to perform aggressive maneuvers in the formation flight, small angles assumption is used to simplify rotational equations of motion. The small-angle assumption for θ and φ leads to:

$$\begin{bmatrix}p\\q\\r\end{bmatrix}=\begin{bmatrix}1&0&-\sin\theta\\0&\cos\phi&\sin\phi\cos\theta\\0&-\sin\phi&\cos\phi\cos\theta\end{bmatrix}\begin{bmatrix}\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix}\approx\begin{bmatrix}\dot{\phi}\\\dot{\theta}\\\dot{\psi}\end{bmatrix} \tag{4}$$

which means that rotational equations can be written as:

$$\ddot{\phi} \approx \dot{\theta}\dot{\psi}(\frac{I_{yy}-I_{zz}}{I_{xx}}) + \frac{J_r\dot{\theta}\Omega_r}{I_{xx}} + \frac{U_2}{I_{xx}}$$
$$\ddot{\theta} \approx \dot{\phi}\dot{\psi}(\frac{I_{zz}-I_{xx}}{I_{yy}}) - \frac{J_r\dot{\phi}\Omega_r}{I_{yy}} + \frac{U_3}{I_{yy}} \quad (5)$$
$$\ddot{\psi} \approx \dot{\theta}\dot{\phi}(\frac{I_{xx}-I_{yy}}{I_{zz}}) + \frac{U_4}{I_{zz}}$$

Control Method.

As stated formerly, the control subsystem has to track the produced trajectory of the guidance subsystem. It is evident from equations 1 and 5 that quadrotor has independent control inputs for $[\varphi\ \theta\ \psi\ z]$axis, unlike $[x\ y]$axis. In formation flight, like many other missions, a position reference consisting of$[x_d y_d z_d \psi_d]$should be tracked. Therefore, the underactuated dynamics of the quadrotor should be dealt with. To design the control subsystem for a quadrotor, it is common to use a higher frequency of rotational dynamics compared to translational dynamics. In this way, the controller will attainan inner-outer loop configuration. In the dynamic inversion approach, three virtual accelerations are produced at first. Then, the desired thrust and the desired attitudes$[\varphi_d \theta_d]$ are computed to reach the mentioned virtual accelerations. In this approach, φ and θ are controlled automatically. Virtual accelerations are calculated based on the translational dynamics of the quadrotor as:

$$\mu_x = \frac{-U_1}{m}(\cos\psi_d \sin\theta_d \cos\phi_d + \sin\psi_d \sin\phi_d)$$
$$\mu_y = \frac{-U_1}{m}(\sin\psi_d \sin\theta_d \cos\phi_d - \cos\psi_d \sin\phi_d) \quad (6)$$
$$\mu_z = -\frac{U_1}{m}(\cos\theta_d \cos\phi_d) + \mathrm{g}$$

Next, thrust, needed to reach these accelerations, is computed as:

$$U_1 = m\sqrt{\mu_x^2 + \mu_y^2 + (\mathrm{g} - \mu_z^2)} \quad (7)$$

Finally the desired φ and θ can be determined by:

$$\phi_d = \arcsin(-\mathrm{m}\frac{\mu_x \sin\psi_d - \mu_y \cos\psi_d}{U_1})$$
$$\theta_d = \arctan(-\frac{\mu_x \cos\psi_d + \mu_y \sin\psi_d}{g - \mu_z}) \quad (8)$$

Stability analysis and convergence of this approach are studied in [22], where equations 7 and 8 have been derived without any limiting assumption. However, no aggressive maneuvers are needed in the context of this paper. As a result, the small-angle condition for φ and θ can be used in deriving the desired thrust and the desired attitudes. Accordingly, virtual acceleration equations are modified as noted below:

$$\mu_{x_{new}} = \frac{U_{1_{new}}}{m}[\sin\phi_d \sin\psi_d + \cos\phi_d \sin\theta_d \cos\psi_d]$$
$$\approx \frac{U_{1_{new}}}{m}[\phi_d \sin\psi_d + \theta_d \cos\psi_d]$$
$$\mu_{y_{new}} = \frac{U_{1_{new}}}{m}[\sin\phi_d \cos\psi_d - \cos\phi_d \sin\theta_d \sin\psi_d] \quad (9)$$
$$\approx \frac{U_{1_{new}}}{m}[\phi_d \cos\psi_d - \theta_d \sin\psi_d]$$
$$\mu_{z_{new}} = -\frac{U_{1_{new}}}{m(\cos\phi_d \cos\theta_d)} + g$$
$$\approx -\frac{U_{1_{new}}}{m} + g$$

Now the new thrust is computed based on the required vertical acceleration:

$$U_{1_{new}} = m(g - \mu_z) \quad (10)$$

Finally, the new desired φ and θ can be computed as follow:

$$\phi_{d_{new}} = \frac{m}{U_{1_{new}}}[-\mu_{x_{new}} \sin\psi_d + \mu_{y_{new}} \cos\psi_d]$$
$$\theta_{d_{new}} = \frac{m}{U_{1_{new}}}[-\mu_{x_{new}} \cos\psi_d - \mu_{y_{new}} \sin\psi_d] \quad (11)$$

To guarantee the realization of a small-angle assumption, it is needed to enforce a saturation constraint on φ and θ. This prevents quadrotors from aggressive maneuvers during the maneuvers.

$$\phi_{d_{new}} = \begin{cases} \phi_{\max} & \phi_{\max} < \phi_d \\ \phi_d & -\phi_{\max} < \phi_d < \phi_{\max} \\ -\phi_{\max} & \phi_d < -\phi_{\max} \end{cases}$$
$$\theta_{d_{new}} = \begin{cases} \theta_{\max} & \theta_{\max} < \theta_d \\ \theta_d & -\theta_{\max} < \theta_d < \theta_{\max} \\ -\theta_{\max} & \theta_d < -\theta_{\max} \end{cases} \quad (12)$$

Dynamic Inversion method transformed the control design problem to the problem of controlling six state variables. PD technique is appropriate to produce control law, because of its simplicity and effectiveness. Thus, moments and virtual accelerations can be computed as:

$$\mu_{x_{new}} = K_{px}e_x + K_{dx}\dot{e}_x$$
$$\mu_{y_{new}} = K_{py}e_y + K_{dy}\dot{e}_y$$
$$\mu_{z_{new}} = K_{pz}e_z + K_{dz}\dot{e}_z$$
$$U_2 = K_{p\phi}e_\phi + K_{d\phi}\dot{e}_\phi \quad (13)$$
$$U_3 = K_{p\theta}e_\theta + K_{d\theta}\dot{e}_\theta$$
$$U_4 = K_{p\psi}e_\psi + K_{d\psi}\dot{e}_\psi$$

where e is the error in the relevant state, and K_p and K_d are proportional and derivative coefficients, respectively. To summarize, a MIMO system is decomposed into six SISO systems by the dynamic inversion approach and then the SISO systems are controlled with PD technique. Since the concentration of this paper is on the guidance subsystem, an existing controller for quadrotors is utilized, which is sufficient to track the produced trajectories of the guidance subsystem. It will be demonstrated that the guidance subsystem adapts to the proposed control subsystem through the proposed tuning procedure.

Guidance Subsystem

The guidance subsystem produces an appropriate reference for the control subsystem. This reference is computed based on mission specifications, dynamical limitations, and control subsystem capabilities. In this research, the behavior-based approach is adopted in the design of the guidance subsystem. The guidance subsystem directs a group of quadrotors to their destination besides keeping them in a specific formation and preventing them from colliding to obstacles or each other. In this paper, leader following and obstacle avoidance behaviors are implemented directly; however, collision avoidance and formation keeping behaviors are implemented indirectly. Accordingly, the total reference for the i[th] quadrotor $\boldsymbol{R}_{tot,i}$, is generated by adding up three vectors as noted below:

$$\boldsymbol{R}_{tot,i} = \boldsymbol{R}_{lf,i} + \boldsymbol{R}_{oa,i} + \boldsymbol{R}_{a,i} \quad (14)$$

Where $\boldsymbol{R}_{lf,i}$ is the leader following reference, $\boldsymbol{R}_{oa,i}$ isthe obstacle avoidance reference and $\boldsymbol{R}_{a,i}$ is the actual position vector of the I[th] quadrotor, where all are resolved in inertial/navigation frame.

Leader Following Behavior

In this behavior, the leader can be another robot or a virtual moving target that other robots have to keep their distance towards it. Different hierarchical strategies can be proposed to implement the leader following behavior. In our strategy, there is only one leader which is a virtual moving target that all quadrotors have to keep their relative position regarding it (Figure 3). A virtual moving target will move along a primary trajectory that is produced without considering the obstacles in the path. If all quadrotors keep their desired distance to the primary trajectory accurately, the formation keeping and collision avoidance will be realized indirectly. It should be noted that the obstacle avoidance will be resolved by introducing a new behavior in the next subsection. Reference vector produced by leader following behavior for I[th] quadrotor will benoted as below:

$$\boldsymbol{R}_{lf,i} = K_{lf,i}(\boldsymbol{D}_{d,i} - \boldsymbol{D}_{a,i}) \quad (15)$$

Where

$$\boldsymbol{D}_{a,i} = \boldsymbol{R}_{a,i} - \boldsymbol{R}_{vmt}$$
$$\boldsymbol{D}_{d,i} = \boldsymbol{R}_{d,i} - \boldsymbol{R}_{vmt} \quad (16)$$

In the above, $K_{lf,i}$is the leader following gain for the i[th] quadrotor,$\boldsymbol{D}_{d,i}$is the desired relative position between the I[th] quadrotor and virtual moving target,$\boldsymbol{D}_{a,i}$is the actual relative position between the I[th] quadrotor and virtual moving target,$\boldsymbol{R}_{d,i}$is the desired position vector of the I[th] quadrotor and $\boldsymbol{R}_{vmt}$is the position vector of the virtual moving target, all of them in the inertial/navigation frame.

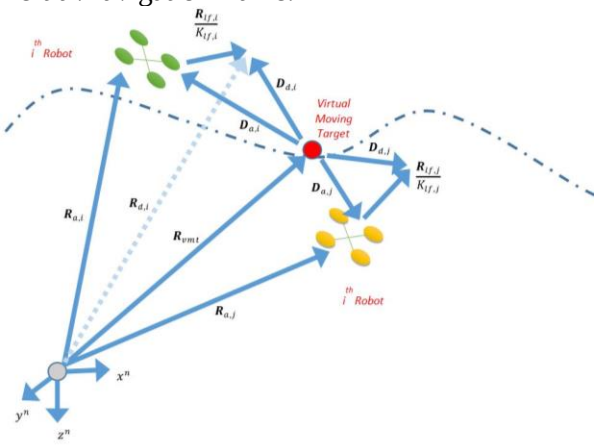

Figure 3. The strategy for implementation of the leader following behavior.

The guidance subsystem has to be kept simple enough to make the tuning problem easier. The proposed strategy for the leader following behavior, in which all quadrotors keep their relative positions regarding a virtual moving target, keeps the number of optimization variables small enough for the optimization process.

Obstacle Avoidance Behavior

To perform a safe formation flight, an obstacle avoidance behavior is defined which will be executed in parallel with the leader following behavior. It is assumed that the position of the

obstacle is known and quadrotors can compute their relative position to the obstacle during the mission. To prevent quadrotors from entering the danger zone, an avoidance zone is considered around the obstacle in which obstacle avoidance behavior will be active. As it is shown in Figure 4, the obstacle avoidance behavior will produce a reference vector, radially outwards of the obstacle. Reference vector produced by obstacle avoidance behavior is defined so that its magnitude increases gradually as quadrotor gets closer to the obstacle:

$$\boldsymbol{R}_{oa,i}=\begin{cases} K_{oa,i}\left(\frac{R_{az}}{\|\boldsymbol{D}_{o,i}\|}-1\right)\frac{\boldsymbol{D}_{o,i}}{\|\boldsymbol{D}_{o,i}\|} & for \ \ \|\boldsymbol{D}_{o,i}\|<R_{az} \\ 0 \quad otherwise \end{cases} \qquad (17)$$

Where

$$\boldsymbol{D}_{o,i}=\boldsymbol{R}_{a,i}-\boldsymbol{R}_{o,i} \qquad (18)$$

In which $\boldsymbol{R}_{oa,i}$ is the reference vector of obstacle avoidance behavior for the ith quadrotor, $K_{oa,i}$ is the gain of obstacle avoidance behavior for the ith quadrotor, $\boldsymbol{D}_{o,i}$ is the relative position between ith quadrotor and cylindrical obstacle center, $\boldsymbol{R}_{o,i}$ is the position vector of the cylindrical obstacle center axis and R_{az} is the avoidance zone radius. To compensate for different velocities in obstacle avoidance maneuver, R_{az} is defined as a function of quadrotor velocity as noted below:

$$R_{az}=R_{az_{\min}}+K_{az}\|\dot{\boldsymbol{R}}_{a,i}\| \qquad (19)$$

Where K_{az} is the gain and $R_{az_{min}}$ is a minimum magnitude for avoidance zone radius. It should be stated that the quad diameter effects in obstacle avoidance can be included in the safety regions. The other point is that during the obstacle avoidance process, the robots give up the formation and follow the resolution path separately, while they are considering collision avoidance between themselves. As a result, as long as they perform individually, the formation radius is not required during the obstacle avoidance phase.

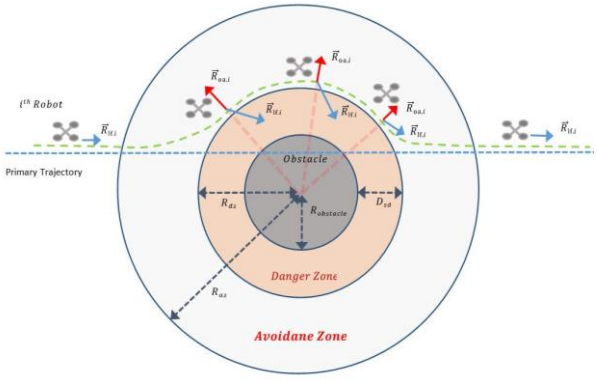

Figure 4. Interaction of obstacle avoidance and leader following behaviors during passage around the obstacle.

Reference Generation

In this work, slow maneuvers in the control design process are assumed. Thus, it is necessary to limit the aggressiveness of the produced trajectory to prevent control subsystem saturation. Consequently, a threshold is set to limit the final reference. The saturation function is defined as:

$$\boldsymbol{R}'_{\boldsymbol{tot},i}=\min\{\|\boldsymbol{R}_{\boldsymbol{tot},i}\|,R_{\max}\}\frac{\boldsymbol{R}_{\mathrm{tot},i}}{\|\boldsymbol{R}_{\mathrm{tot},i}\|} \qquad (20)$$

Where $\boldsymbol{R}'_{tot,i}$ is the final reference and R_{max} is the maximum allowable magnitude for final reference. To improve performance, R_{max} is defined dynamically as:

$$R_{\max}=R_{\min}+K_{sat}\|\dot{\boldsymbol{R}}_{a,i}\| \qquad (21)$$

in which K_{sat} is again and R_{min} is a minimum for R_{max}.

In the following, the block diagram of the designed flight control framework is shown in Fig. 5. As stated earlier, the navigation system uncertainties and rotor lags will be included in simulations to reach more realistic tuned parameters. Until this time, the guidance and control subsystems were designed. In the next section, the optimization process for tuning the behavior-based parameters is described to achieve the optimal and safe formation flight.

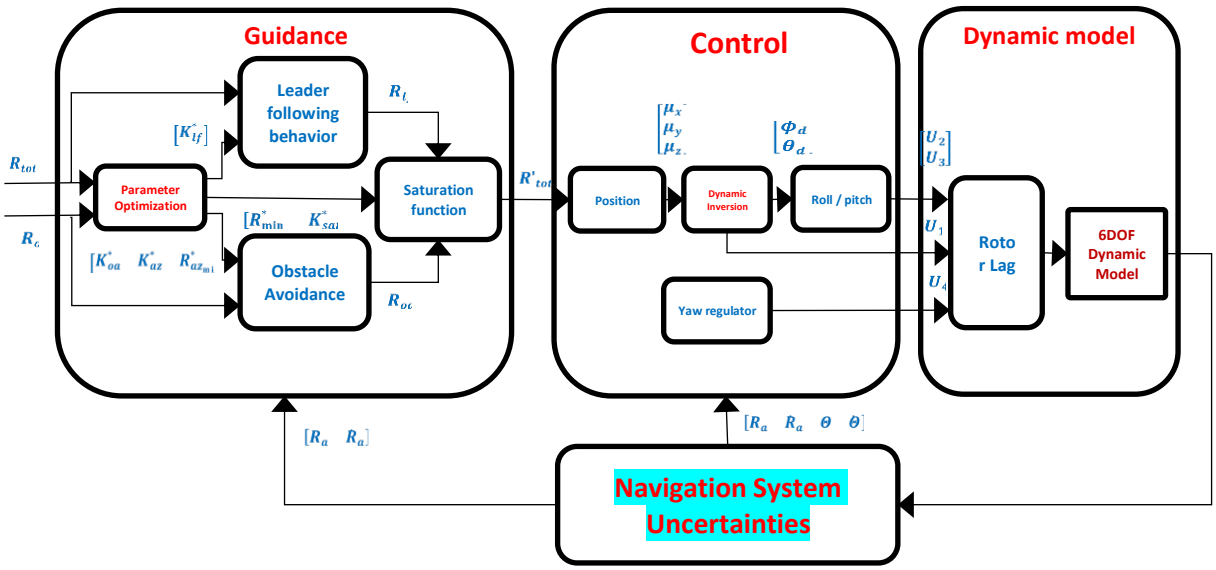

Figure 5. Block diagram of the closed-loop system including guidance and control subsystems.

TUNING OF GUIDANCE PARAMETERS

In this section, the tuning of the behavior-based algorithm to minimize reference deviation and control effort, besides maintaining a safe distance from obstacles, is described. The optimization problem includes three different objectives; leader following (reference deviation), obstacle avoidance (safe distance constraint), and control effort. Quadrotors will keep their relative position from a predefined trajectory which leads to a specific formation without the necessity of definition of a separate formation keeping behavior. Moreover, the trajectories are defined in different altitudes, which prevents the collision of quadrotors. In fact, by the indirect realization of formation keeping and collision avoidance behaviors, the optimization problem is simplified. By considering the proposed behavior-based algorithm, six independent parameters $[K_{lf} K_{oa} R_{min} K_{sat} R_{az} K_{az}]$emerge.

Magnitudes of these parameters have major effects on the flight control framework performance. Manual tuning of these parameters to realize optimal and safe formation flight is very challenging. This is because of the nonlinear dynamics of quadrotors and the opposing role of behaviors in the guidance subsystem. To tune the mentioned parameters, an optimization algorithm with acceptable accuracy and speed is proposed. Maintaining a safe distance to the obstacle makes the optimization problem a constraint one. There are two main approaches to solve a constraint optimization problem. In the first approach, the constraint optimization problem is converted to an unconstraint one using techniques like penalty functions or Lagrange multiplier, and then it is solved using conventional unconstrained optimization methods. In the second approach, the optimization problem in the presence of the constraints is solved; all together with methods like SLP and SQP. The goal of this paper is to tune the performance of a flight control framework; and detailed optimization analysis is beyond the current research contributions. Thus, the first approach is utilized. Indeed, the Penalty function method is used to convert the optimization problem to an unconstraint one and then the univariate search method is used to solve the unconstraint optimization problem without the need for derivative evaluation.

The first step to solve the optimization problem is to define an appropriate cost function. The primary cost function for minimization of reference deviation and control effort can be defined as:

$$J_0 = \int (\mathrm{K}_D \,\|\, \mathbf{D}_{a,i} \,\|) + K_U \,\|\, \mathbf{U} - \mathbf{U}_{hover} \,\|)\, \mathrm{dt} \qquad (22)$$

Where $K_D$is the reference deviation coefficient, $K_U$is the control effort coefficient and$\boldsymbol{U}$is the input vector. $\boldsymbol{U}_{hover}$is the input vector in hovering mode, where moments are zero, and the thrust force equals the weight of the quadrotor. After defining the cost function, the optimization algorithm has to be determined. The univariate search method is among non-gradient optimization approaches that do

not require derivative evaluation. As depicted in Figure 6, in this method an initial guess vector is made and every element of the optimization vector is changed, separately, by a unit mesh. The direction of the largest decrease in the cost function is chosen, and the new optimization vector is constructed. Then, the process is repeated for the new optimization vector. If none of the directions leads to a decrease in the cost function, then the mesh size is made half and the process is repeated for the new optimization vector. The process moves forward until the relative change in mesh size or cost function reaches lower values than the specified one. To import the safe distance constraint into the cost function, an exterior penalty function is defined as noted below:

$$G = \frac{\max(0, R_{dz} - \min(\| \boldsymbol{D}_{o,i} \|))}{R_{dz}} \quad (23)$$

when the quadrotor violates the safe distance constraint, the G function starts to increase. The equation is divided by the danger zone radius to bind the range of Gin [0, 1]. The danger zone is computed as follows:

$$R_{dz} = R_{ob} + D_{sd} \quad (24)$$

in which$R_{ob}$is the cylindrical obstacle radius and D_{sd} is a desired safe distance from the obstacle surface. It should be stated that we have assumed the multi-rotors already know the dimensions and size of the obstacle, which affects the warning/danger zone of the collision with the obstacle. As a result, the final cost function will be as follows:

$$J_1 = \int (\mathrm{K}_D \| \mathbf{D}_{a,i} \|) + K_U \| \mathbf{U} - \mathbf{U}_{hover} \|) \mathrm{dt} + \mathrm{K}_G \times G^2 \quad (25)$$

WhereinK_G is the penalty function coefficient, which should be gradually increased during optimization iterations. The optimization process will start with an infeasible guess, which in our case, means that the quadrotor collides with the obstacle on a straight line. While optimization proceeds, guidance parameters are tuned so that the traveled trajectory of quadrotor moves toward the danger zone circumference. The optimization continues till the minimum distance of the quadrotor from the obstacle during the mission converges to the danger zone radius; which makes G tend to zero. IfK_G is kept fixed during the iterations, the optimization will be stuck in an infeasible region. Accordingly, $K_G$is defined so that it increases during the optimization process. Although many functions in the literature have been proposed for the K_G, here the following heuristic function works appropriately:

$$K_G = \frac{C}{G} \quad (26)$$

Where C is a constant value. This function makes K_G grow gradually during optimization and tend to infinity when algorithm converges. Gradual improvement of the trajectories during optimization iterations are presented in the first scenario of simulations. Moreover, the effect of coefficients of reference deviation and control effort is studied in this scenario.

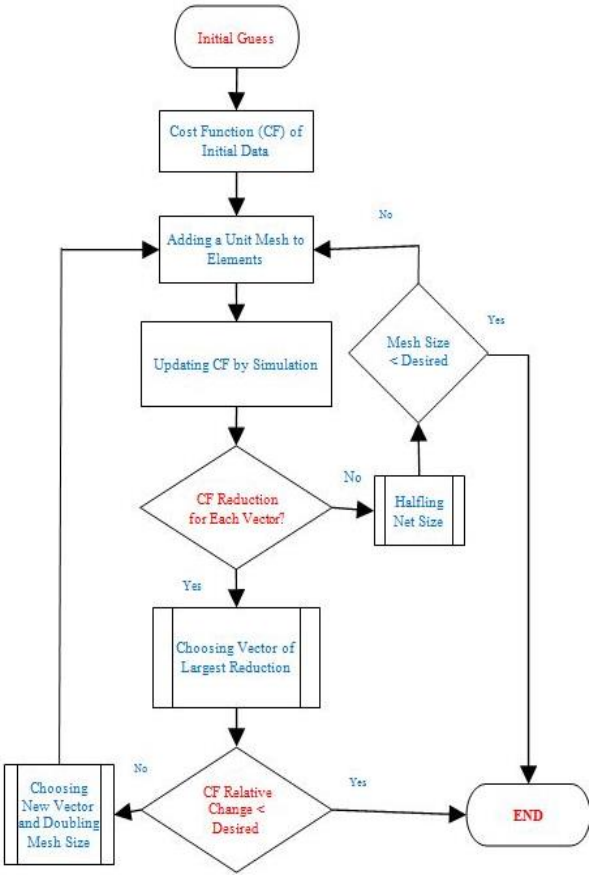

Figure6. Flowchart oftheoptimization process.

Simulation Results

In the current research, MATLAB/Simulink is exploited to simulate the performance of the closed-loop system [23]. For consistency, most of the structural and aerodynamic parameters of the simulation platform, in Table 1, are obtained from the OS4 platform [24].

Table 1. Quadrotor aerodynamic and structural parameters.

Parameter	Magnitude
I_{xx}	$7.5\times10^{-3}\ kgm^2$
I_{yy}	$7.5\times10^{-3}\ kgm^2$
I_{zz}	$1.3\times10^{-2}\ kgm^2$
m	$1.65\ kg$
J_r	$6\times10^{-5}\ kgm^2$
l	$0.23\ m$
b	$3.13\times10^{-5}\ Ns^2$
d	$7.5\times10^{-7}\ Nms^2$

A quadrotor is a naturally unstable flying vehicle that does not use any control surfaces and only relies on its rotors angular velocities to produce required forces and moments. As a result, the rotors lag has a major impact on the stability of the quadrotor. The time constant of rotors is taken as 0.05 sec. [25] which is a moderate magnitude for the rotors of the quadrotor. Furthermore, taking into account the most important destabilizing factors in the simulations, the tuned parameters become more practical. In this regard, as stated before, the overall navigation system uncertainties are considered. The value of errors depends on the type of sensors, filters, and mechanization methods. For the case of simplicity, by considering a control-type class of sensors, a zero-mean white Gaussian noise (ZMWGN) with 0.1 std has been added to the position and velocity signals, and the attitude measurements have been corrupted by a ZMWGN with 0.01 std.

Two scenarios are designed to test the proposed flight control framework. In the first scenario, the tuning procedure of the guidance subsystem is analyzed for one quadrotor. The capability of the optimization algorithm to differentiate multiple objectives (reference deviation and control effort) is discussed in this scenario. In the second scenario, the formation flight of three quadrotors is investigated. Optimality and safety of the proposed flight control framework to perform the formation flight of quadrotors is studied in this scenario.

First Scenario: Single Flight.

In this scenario, a quadrotor should pass around a cylindrical obstacle safely and optimally. Optimization convergence and the effect of different objectives in the cost function is studied in this scenario. A straight trajectory as the primary trajectory is defined which travels through the danger zone. The optimization algorithm, not only adjusts guidance parameters to minimize reference deviation and control effort, but also prevents violation of the safe distance constraint.

According to the previous discussions, optimization variables are:

$$[K_{lf}K_{oa}R_{min}K_{sat}R_{az}K_{az}]$$

The virtual moving target is $R_{vmt}(t) = [3t, 0,0]$where quadrotor is required to keep its relative position from it with an offset of $\boldsymbol{D}_d = [0, 0 ,0]$. Initially, quadrotor is at$\boldsymbol{R}_{a0} = [0, 0,0]$ and there is a cylindrical obstacle, the central axis of which is at$\boldsymbol{R}_o = [120,5]$ and has a 15m radius. The safe distance of the quadrotor from the obstacle surface is taken as 5m. Consequently, the radius of the danger zone around the obstacle will be 20 meters. In the first analysis of the current scenario, the convergence of the optimization algorithm is studied. The optimization algorithm is started by an initial guess [0.1, 0.1, 1, 0, 20, 0], in which the robot goes along the predefined trajectory while collision with an obstacle is definite. As optimization goes forward, guidance parameters are adjusted to push the traveled trajectory outside the danger zone. Optimization will continue until the minimum distance of quadrotor to the obstacle during the mission equals to the danger zone radius. In Figure 7, the shrinking of cost function during the optimization iterations is depicted. Moreover, Figure 7 reveals the tuned set of guidance parameters.

Convergence Analysis.

In the current work, optimization computations are conducted employing a PC based on a Core i5 Intel processor including 4 GB of RAM. Hence, the presented optimization time in Table 3 can be significantly reduced by using a more powerful system. The simulation scenario has been executed for 193 times to compute the cost function. However, run times can be reduced by exploitation of more advanced optimization techniques. While optimization runs forward,

the minimum distance of the quadrotor to the obstacle increases until the safe distance constraint is satisfied. This is shown in Figure 8, where in primary iterations, quadrotor will collide with the obstacle but in the last iterations, the safe distance constraint is granted. It is clear from Figure 9 that the tuning process has reduced reference deviation before and after the obstacle avoidance maneuver. Moreover, reference deviation has reached its peak, when the quadrotor passes by the obstacle.

It is noticeable from Figure 10 that the safe distance constraint is satisfied in the last iteration. Moreover, the rate of change of the distance between the quadrotor and the obstacle is straighter in the last iteration which implies the accuracy of the last set of tuned variables in the tracking of the predefined trajectory. It should be noted that the cost function is calculated based on the traveled trajectory and not relying on the produced one. The guidance subsystem generates a trajectory but the traveled trajectory in the simulation is used to calculate the cost function. It is shown in Figure 11 that the traveled trajectory differs from the produced trajectory specifically when the quadrotor passes through the danger zone. This is because of the dynamics of the vehicle, where direction changing takes time.

Table 2. Optimization values in different iterations.

Step	K_G	$\min(\|\boldsymbol{D}_{o,i}\|)$ (m)	**Cost Function**	**Function Count**	**Elapsed Time (s)**
0	1444	6.15	949	1	**1**
3	3351	14.03	707	7	**13**
9	6077	16.71	634	20	**41**
29	9646	17.93	482	84	**166**
51	860840	19.99	416	193	**359**

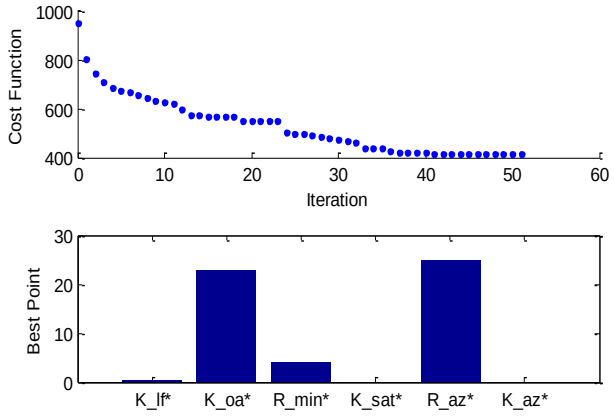

Figure 7. Cost function history through the optimization process (up), and the tuned parameters (down).

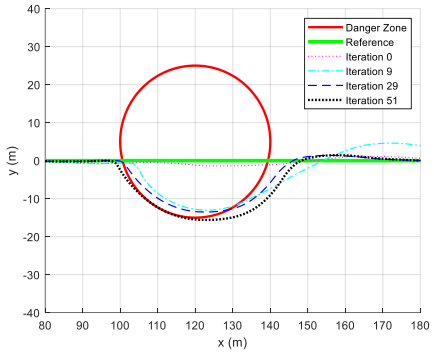

Figure. 8. Traveled path around the obstacle through optimization iterations.

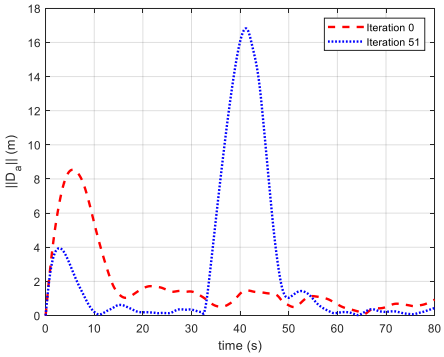

Figure 9. Reference deviation in the first and last iteration.

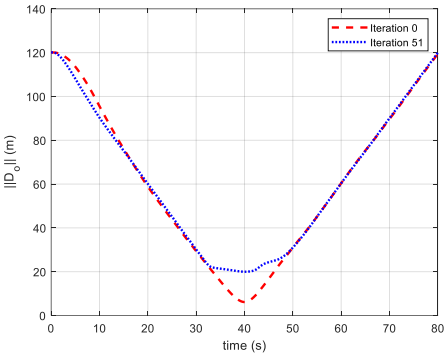

Figure 10. Distance between the quadrotor and the obstacle central axis in the first and last iteration.

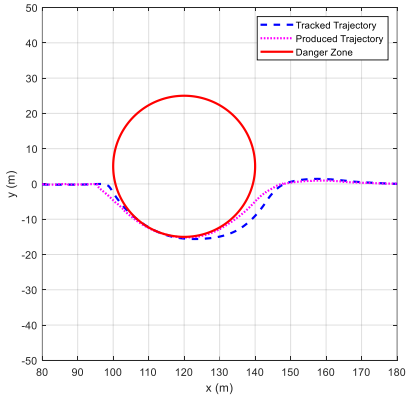

Figure 11. The traveled and produced trajectories for the last iteration of optimization.

As stated before, Figure 8 demonstrates the effect of the solution of the optimization process in the performance of the aerial robot. In initial steps, due to using low gain values (a non-optimal solution), the agents are not capable of resolving the obstacle avoidance, and collide with the obstacle or pass through the danger zones.

Cost function Analysis

At this step, the effect of coefficients of the cost function in the optimization procedure is investigated. The second scenario is optimized for four different magnitudes of control effort coefficient K_U, which are taken as:

$K_{U1} = [1\ 250\ 250\ 250]$

$K_{U2} = [1\ 500\ 500\ 500]$

$K_{U3} = [1\ 750\ 750\ 750]$

$K_{U4} = [1\ 1000\ 1000\ 1000]$

The first element of K_U is kept fixed at 1, because the quadrotor must have as little altitude change as possible to realize collision avoidance behavior indirectly. So the first element in K_U which is related to the thrust has not been changed. The other three elements have been changed to see their effects. The 2D view of the optimized traveled trajectory for different control effort coefficients is shown in Figure 12. The main difference appears when the quadrotor needs to return to the initial trajectory. In that phase, increased K_U makes quadrotor to experience anovershoot along the initial trajectory and converge to it later. This fact implies that the vehicle has used its control inputs less. So the optimization algorithm can effectively separate reference deviation and control effort terms.

The tuned guidance parameters are shown for different K_U in Figure 13. As K_U increases, K_{lf}^*, K_{oa}^* and R_{min}^* decrease while K_{sat}^* and R_{az}^* remain roughly constant. Meanwhile K_{az}^* is kept zero by tuning algorithm in the first 3 conditions and it raises to 1 which leads to a triple increase in R_{az}^*. An approximate trend can be seen in the optimal variables in Figure 13. Since K_{sat}^* and K_{az}^* are useful in aggressive maneuvers where the velocity of robot changes significantly, here they have not changed considerably. The control input history is displayed in Figure 14. As K_U increases, usage of U_1, U_2 and U_3 is reduced significantly, but the usage of U_4 is increased. Although the first element of K_U is kept fixed in its four cases, limitation of U_2 and U_3 has led to less usage of U_1 during the maneuvers. Smaller order of magnitude of U_4 in comparison to other control variables, makes algorithm rely more on it when K_U increases. Integral of reference deviation and control effort terms is presented in Figure 15, which indicates that the optimization algorithm is capable of differentiating the two competing objectives.

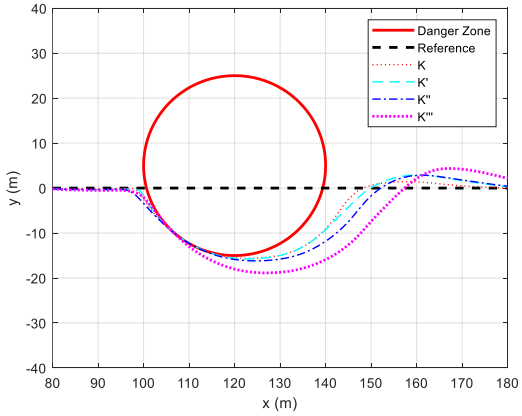

Figure 12. Traveled trajectories for different coefficients of control effort term after tuning.

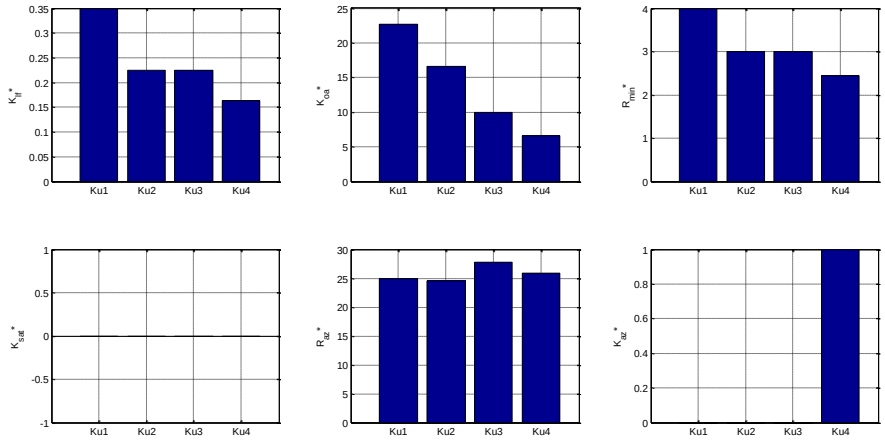

Figure 13. Tuned guidance parameters for differentcoefficients of control effort term.

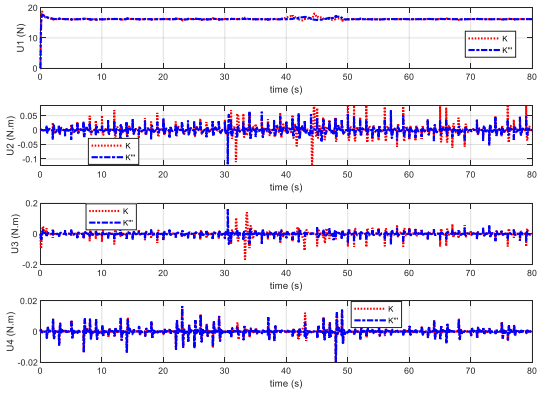

Figure 14. Control inputs after tuning for the highest and lowest values of the control effort coefficient.

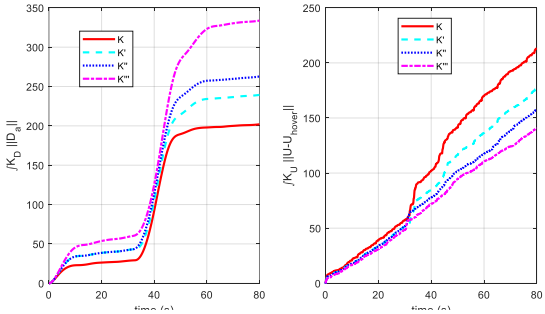

Figure 15. Integral of reference deviation (left) and control effort (right) for different values of control effort coefficient.

5.2. Second Scenario: Formation Flight. In this scenario, a formation flight of three quadrotors is simulated to evaluate flight control framework performance. In this scenario, quadrotors track a sinusoidal trajectory while they keep their relative position to the reference, during flights at different altitudes. The predefined trajectory is taken as $\boldsymbol{R}_{vmt} = [3t, 25(1-\cos(0.1t)), 0]$ where the initial position of quadrotors are $\boldsymbol{R}_{a0,1} = [0,-10,0]$, $\boldsymbol{R}_{a0,2} = [0,0,5]$ and $\boldsymbol{R}_{a0,3} = [0,10,10]$, respectively. Quadrotors should keep their relative position to the virtual moving target as $\boldsymbol{D}_{d,1} = [0,-10,0]$, $\boldsymbol{D}_{d,2} = [0,0,5]$ and $\boldsymbol{D}_{d,3} = [0,10,10]$, respectively. Likewise, a cylindrical obstacle with its central axis on $\boldsymbol{R}_o = [120,41]$ is included in the scenario. Due to the 15m radius of obstacle and 5m safe distance of quadrotor to the obstacle surface, the minimum distance of quadrotor to the obstacle central axis has to be 20m. Due to different situations that quadrotors encounter during this scenario, the tuning algorithm produces a different set of parameters for every quadrotor. The final performance of formation flight in the second scenario is exhibited in 2D and 3D views (Figure 16).

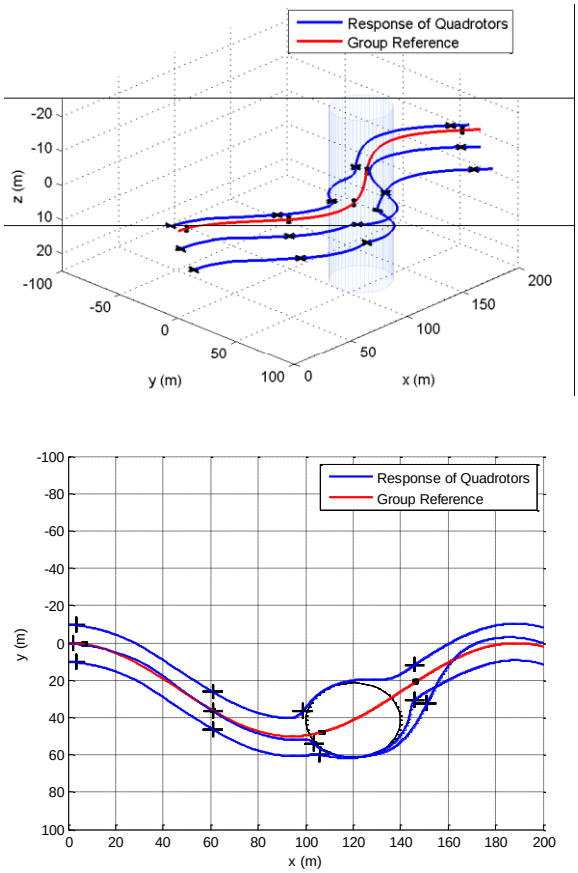

Figure 16. Formation flight of three quadrotors while passing around an obstaclein 3D (up) and 2D (down).

It is noticeable from Figure 16 that all three quadrotors keep a minimum possible distance from the obstacle during the mission, which implies optimality and safety of the performed mission. because quadrotors number two and three both go to the right of the obstacle, there is a collision chance which is reduced to zero by enforcing different flying altitudes to them. This is done to establish collision avoidance behavior indirectly. In Figure17, altitudes of the three quadrotors during formation flight is shown. The flight control framework is capable of keeping quadrotors altitudes even in lateral maneuvers. Quadrotor number two has gone a more aggressive lateral maneuver, which

has made its altitude change more than the two other quadrotors. The maximum height loss of these quadrotors during the mission, are 2, 10, and 4 percent of the height difference between every two of them. Furthermore, the pairwise distance between quadrotors during formation flight is presented in Figure 18. As the first quadrotor passes over the left of the obstacle, its distance from the two other quadrotors remained safe. Although quadrotor numbers two and three both pass over the right side of the obstacle, their minimum distance becomes 7m which remains completely safe.

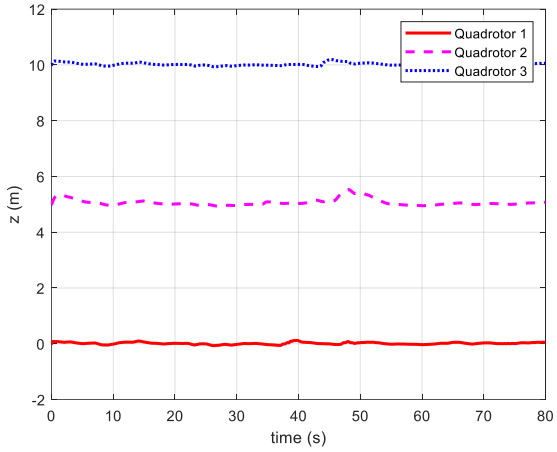

Figure 17. Altitudes of three quadrotors during the formation flight.

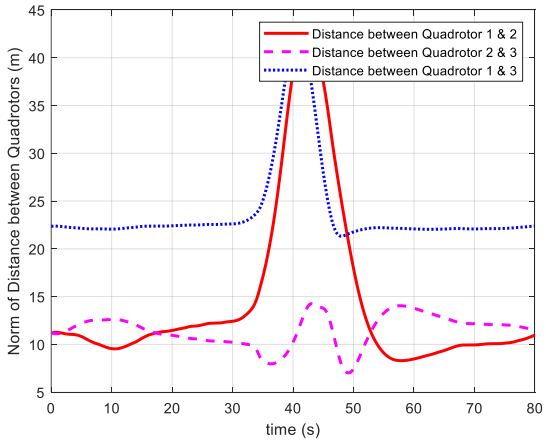

Figure 18. Pairwise distance between quadrotors during the formation flight.

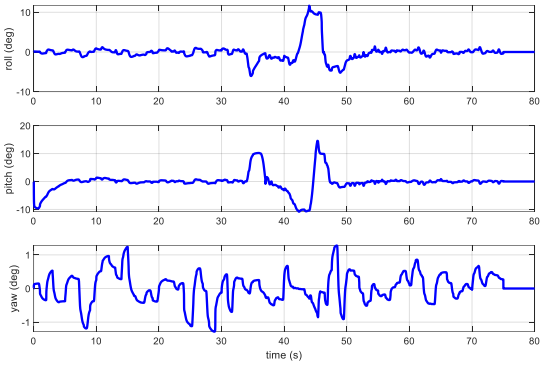

Figure 19. Time history of Euler angles.

To ensure that the Euler angles remain in their linearization limit, we illustrate them in Figure 19. As it can be seen, roll and pitch angles, which are used in the linearization process of the dynamic inversion control, stay below 10 degrees.

CONCLUSION

In this article, the performance of a flight control framework for the formation flight of quadrotors is tuned by using optimization techniques. In the proposed layered flight control framework, parallel primary trajectories at different altitudes are produced for robots. As a result, collision avoidance and formation keeping behaviors will be automatically realized if those initial trajectories are followed precisely via leader following behavior. Subsequently, the guidance module decides on the dominance of the leader following and obstacle avoidance behaviors by its tuned parameters. The penalty function method is used to tune those parameters to make the flight path tangent to the danger zone. By this method, reference deviation and control effort are minimized while quadrotors do not enter the danger zone. Furthermore, the defined cost function can separate reference deviation and control effort objectives effectively during the tuning procedure. The simulation results exhibited the improved performance of the proposed flight control framework in accurate maneuvering and the formation of quadrotors around the obstacles. To enhance the current work, one can include the “coordination

delay" between the agents of the group, further on the current engine lags.

REFERENCES

[1] R. J. Ray, B. R. Cobleigh, M. J. Vachon, and C. StJohn, "Flight test techniques used to evaluate performance benefits during formation flight," in *NASA Conference publication*, 2002.

[2] Z. A. Bangash, R. P. Sanchez, A. Ahmed, and M. J. Khan, "Aerodynamics of formation flight," *J. Aircr.*, vol. 43, no. 4, pp. 907–912, 2006.

[3] D. P. Scharf, F. Y. Hadaegh, and S. R. Ploen, "A survey of spacecraft formation flying guidance and control (part I): Guidance," in *Proceedings of the American control conference*, 2003, vol. 2, pp. 1733–1739.

[4] M. S. Alvissalim *et al.*, "Swarm quadrotor robots for telecommunication network coverage area expansion in disaster area," in *SICE Annual Conference (SICE), 2012 Proceedings of*, 2012, pp. 2256–2261.

[5] J. Fink, N. Michael, S. Kim, and V. Kumar, "Planning and control for cooperative manipulation and transportation with aerial robots," *Int. J. Rob. Res.*, vol. 30, no. 3, pp. 324–334, 2011.

[6] M. Fadhil *et al.*, "Circular Leader-Follower Formation Control of Quad-Rotor Aerial Vehicles," vol. 25, no. 1, 2013.

[7] D. A. Mercado, R. Castro, and R. Lozano, "Quadrotors flight formation control using a leader-follower approach," in *Control Conference (ECC), 2013 European*, 2013, pp. 3858–3863.

[8] F. Rinaldi, S. Chiesa, and F. Quagliotti, "Linear quadratic control for quadrotors UAVs dynamics and formation flight," *J. Intell. Robot. Syst.*, vol. 70, no. 1–4, pp. 203–220, 2013.

[9] V. Roldão, R. Cunha, D. Cabecinhas, C. Silvestre, and P. Oliveira, "A novel leader-following strategy applied to formations of quadrotors," in *Control Conference (ECC), 2013 European*, 2013, pp. 1817–1822.

[10] C. K. Peterson and J. Barton, "Virtual structure formations of cooperating UAVs using wind-compensation command generation and generalized velocity obstacles," in *Aerospace Conference, 2015 IEEE*, 2015, pp. 1–7.

[11] W. Ren and R. Beard, "Decentralized scheme for spacecraft formation flying via the virtual structure approach," *J. Guid. Control. Dyn.*, vol. 27, no. 1, pp. 73–82, 2004.

[12] T. Paul, T. R. Krogstad, and J. T. Gravdahl, "Modelling of UAV formation flight using 3D potential field," *Simul. Model. Pract. Theory*, vol. 16, no. 9, pp. 1453–1462, 2008.

[13] L. Garcia-Delgado, A. Dzul, V. Santibáñez, and M. Llama, "Quad-rotors formation based on potential functions with obstacle avoidance," *IET Control Theory Appl.*, vol. 6, no. 12, pp. 1787–1802, 2012.

[14] S. Kim and Y. Kim, "Three dimensional optimum controller for multiple UAV formation flight using behavior-based decentralized approach," in *Control, Automation and Systems, 2007. ICCAS'07. International Conference on*, 2007, pp. 1387–1392.

[15] J. Ghommam, H. Mehrjerdi, and M. Saad, "Coordinated path-following control for a group of mobile robots with velocity recovery," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, vol. 224, no. 8, pp. 995–1006, 2010.

[16] A. Fujimori, H. Kubota, N. Shibata, and Y. Tezuka, "Leader–follower formation control with obstacle avoidance using sonar-equipped mobile robots," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, vol. 228, no. 5, pp. 303–315, 2014.

[17] Y. Li, J. Gao, X. Su, and J. Zhao, "Cooperation control of multiple miniature robots in unknown obstacle environment," *Proc. Inst. Mech. Eng. Part I J. Syst. Control Eng.*, p. 0959651814560422, 2014.

[18] A. Ashrafi, M. Mortazavi, A. Askari, and A. Gholami, "Leader-follower formation control oF UAVs by PID-fuzzy," *AST J.*, vol. 3, pp. 29–40, 2017.

[19] E. Zibaei and M. A. Amiri Atashgah, "A Behavior-Based Approach To Simultaneous Realization of Leader-Following and Obstacle-Avoidance Behaviours for A Flying Robot," *Sharif Journals, Mech. Eng.*, vol. 34, no. 3, pp. 73–85, 2018.

[20] A. Mohammadi, E. Abbasi, M. Ghayour, and M. Danesh, "Formation Control and Path Tracking for a Group of Quadrotors to Carry Out a Suspended Load," *Modares Mech. Eng.*, vol. 19, no. 4, 2019.

[21] A. Nagaty, S. Saeedi, C. Thibault, M. Seto, and H. Li, "Control and navigation framework for quadrotor helicopters," *J. Intell. Robot. Syst.*, vol. 70, no. 1–4, pp. 1–12, 2013.

[22] F. Kendoul, I. Fantoni, and R. Lozano, "Asymptotic stability of hierarchical inner-outer loop-based flight controllers," in *Proceedings of the 17th IFAC world congress*, 2008, pp. 1741–1746.

[23] C. Balas, "Modelling and linear control of a quadrotor," *Cranf. Unicersity, MSc Thesis*, vol. 2007, 2006.

[24] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying." École Polytechnique federale de Lausanne, 2007.

[25] N. Michael, D. Mellinger, Q. Lindsey, and V. Kumar, "The grasp multiple micro-uav testbed," *Robot. Autom. Mag. IEEE*, vol. 17, no. 3, pp. 56–65, 2010.