Multi-Layered Optimal Navigation System For Quadrotors UAV

Journal:	*Aircraft Engineering and Aerospace Technology*
Manuscript ID	AEAT-12-2018-0313.R1
Manuscript Type:	Research Paper
Keywords:	Quadrotors, UAV, Optimization, Control, Trajectory generation and Differential Flatness

Multi-Layered Optimal Navigation System For Quadrotors UAV

Abstract

Purpose – This paper proposes a new multi-layered optimal navigation system that, jointly, optimizes the energy consumption, improves the robustness and raises the performance of a quadrotors unmanned aerial vehicle (UAV).

Design/methodology/approach – The proposed system is designed as a multi-layered system. First, the control architecture layer links the input and the output space via quaternion-based differential flatness equations. Then, the trajectory generation layer determines the optimal reference path and avoids obstacles to secure the UAV from any collisions. Finally, the control layer allows the quadrotors to track the generated path and guaranties the stability using a double loop non-linear optimal backstepping controller (OBS).

Findings – All the obtained results are confirmed using several scenarios in different situations to proof the accuracy, energy optimization and the robustness of the designed system.

Practical implications – The proposed controllers are easily implementable on-board and are computationally efficient.

Originality/value – The originality of this research is the design of a multi-layered optimal navigation system for quadrotors UAV. The proposed control architecture presents a direct relation between the states and their derivatives, which then simplifies the trajectory generation problem. Furthermore, the derived differentially flat equations allow optimizations to occur within the output space as opposed to the control space. This is beneficial because constraints such as obstacle avoidance occur in the output space; hence the computation time for constraint handling is reduced. For the optimal backstepping controller, the novelty is that all the controller parameters are derived using the Multi-Objective Genetic Algorithm (MO-GA) that optimizes all the quadrotors state's cost functions jointly.

Keywords Quadrotors; UAV; Optimization; Control; Trajectory generation and Differential Flatness.

Paper type Research paper

Introduction

In the last several years the UAVs design has quickly developed to meet the exigencies of many types of critical missions such as surveillance, fire protection and search & rescue (SAR). Nowadays fling robots are giving a great support to humans for both military and civil applications.

A quadrotors is a rotorcraft capable of hover, forward flight and vertical takeoff landing; it is emerging as a fundamental research and application platform at present with flexibility, adaptability, and ease of construction. As drawbacks, the nonlinear, underactuated dynamic system and the high energy consumption can be mentioned.

Before the control issue, modeling the quadrotors dynamics is necessary. For this purpose the Newton-Euler approach is often used (Kamel, M. et al. (2015); Liu, Z. X. et al. (2015)) due to its simplicity, but it presents a singularity whenever the pitch angle$\theta = \pm\frac{\pi}{2}$. As an alternative of this approach the most recent researches (CHOUTRI, K. et al. (2018); Carino, J. et al. (2015)) converge over the modeling with quaternions instead of the Euler angles, to eliminate the gimball lock phenomena, and simplify the modeling algebra. A survey of modeling and identification of quadrotor UAV can be found in Zhang, X.et al. (2014).

Quadrotors control and attitude stability have attracted the researchers from several institutions, many works have been published. Wang, S. et al. (2012, July) proposes an attitude estimation and control method based on kalman filter. Huo, X. et al. (2014) and Djamel, K. et al. (2016).studied the attitude stabilization control by using backstepping approach, while others such as Liu, H. et al. (2015) investigates a new quaternion-based robust attitude control method for uncertain parameters of quadrotors.

In fact aerial robots are often required to follow a desired path therefore; trajectory tracking is the most important part in a navigation system. Many published papers dealing with the trajectory generation and tracking problem can be cited, Kehlenbeck, A. G. (2014) treats the aggressive trajectory tracking using a quaternion-based control technique. In Sun, et al. (2015).a nonlinear adaptive trajectory tracking control with parametric uncertainty is proposed. Other controllers was also used such as PID (Tanveer et al. (2013)), linear quadratic LQR algorithm (Pena, M. et al. (2012)), sliding mode variable structure control (Larbi, M. A. et al. (2013)), backstepping (Basri, M. et al. (2014)), optimal control (Bouzid, Y. et al. (2017)), and even intelligent control using neural networks and fuzzy logic (Yang, Y. et al. (2016)). A comparison of the different linear and non-linear controllers using the quaternion approach can be found in Chovancová, A. et al. (2016).

Overall all the above cited works aim to improve the robustness, the performances, and energy optimization, but they may fail to raise them jointly. Therefore, this latter issue implies to consider all the possible constraints as for instance: obstacles in the workspace, the effect of wind disturbances, a possible engine failure and the power limitations. Moreover, the relation between the input and the output space is given via a differential flatness approach that presents a direct relation between the states and their derivatives, which then simplifies the trajectory generation problem. Furthermore, a new Multi-Objective Genetic Algorithm (MO-GA) gain-scheduling approach is introduced.

This is beneficial because fuzzy andneural network approximation-based (Yang, Y. et al. (2016)) from a side, and Gravitational Search Algorithm (Mohd, A. et al. (2015)) from another side, may fail to optimize all the state's cost functions jointly, hence the controller performances are reduced.

System design

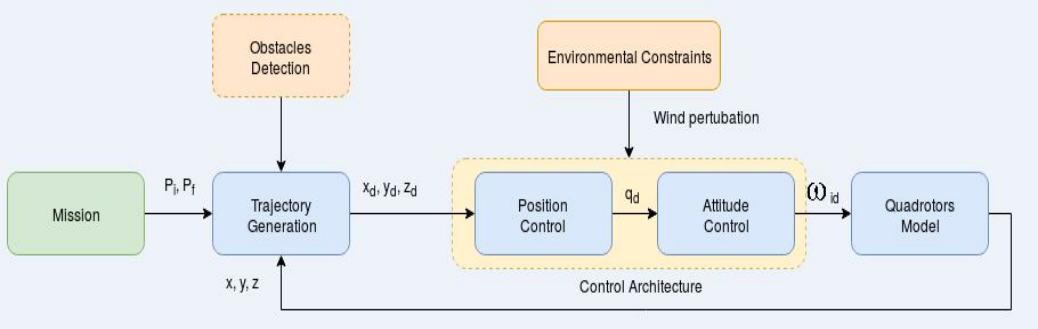
As cited before, the proposed system is designed using a multi-layered architecture. As shown in Figure 1. There are three main levels:

1- *Optimal Trajectory Generation:* This layer is concerned by the generation of the quadrotors trajectory. An optimal path is generated between the departure point P_i and arrival pointP_f. All the physical limitations as wheel as the obstacles are considered as constrains in the generations algorithm. This layer provides the second layer by the desired path (x_d, y_d, z_d) in order to be tracked.

2- *Optimal Controller*: This layer is designed with an optimal controller able to track the desired bath generated by the first layer. A non-linear Optimal Backstepping (OBS) controller allows the quadrotors to track the generated trajectory with a high accuracy and a minimum of energy. The OBS controller parameters are derived using the Multi-Objective Genetic Algorithms (MO-GA) which optimize the energy in the outputs control cost functions.

3- *The Control Architecture:* The quadrotors control architecture is based on a double loop control strategy, an inner loop for the attitude control and outer loop for the position control. An optimization over the control space is reached by using a differential flatness- quaternion based equations.

Figure 1 Multi-layered Optimal Navigation System



Quadrotors modeling

Quadrotors has many advantages over the other UAVs in terms of maneuverability, motion control and cost. Nevertheless, due to the singularities and the nonlinearities of the dynamic model, its modeling becomes quite a challenge.

Dynamic model

Let $E_i\{x_i,y_i,z_i\}$ denotes the inertial frame fixed with the earth while $E_b\{x_b,y_b,z_b\}$ denotes the body frame attached to the quadrotors body. The vector $\Theta = [\varphi\theta\psi]^T$ describes the orientation of the body frame with respect to the inertial frame called the Euler angles. The relation between the angular speeds vector $[pqr]^T$ and Euler angles derivatives is:

$$\begin{bmatrix}\dot{\varphi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix} = \begin{bmatrix}1 & \sin\varphi\tan\theta & \cos\varphi\tan\theta\\ 0 & \cos\varphi & -\sin\varphi\\ 0 & \sin\varphi\sec\theta & \cos\varphi\sec\theta\end{bmatrix}\begin{bmatrix}p\\ q\\ r\end{bmatrix} \quad (1)$$

The problem of singularity emerges whenever $\theta = \pm\frac{\pi}{2}$;

To avoid this problem quaternions are often used. The full dynamic model of a quadrotor using Newton-Euler equations with quaternions is described as follows:

$$\ddot{p} = q \otimes \frac{T}{m} \otimes q^{*} + \bar{g} \quad (2)$$

$$\dot{q} = \frac{1}{2} q \otimes \omega \quad (3)$$

$$\ddot{\omega} = J^{-1}(\tau - \omega \times J\omega) \quad (4)$$

Where $p \in \mathrm{R}^3$and $\dot{p} \in \mathrm{R}^3$are the position and velocity vectors with respect to the inertial frame, Tdefines the thrust vector generated by the quadrotors motors, m and $\bar{g}$ represent the vehicle's mass and gravity vector, respectively, q describes the quaternion that represents the vehicle orientation with respect to the inertial frame, J introduces the inertia matrix with respect to the body-fixed frame and $\tau = \tau_u + \tau_{ext}$, where τ_u and τ_{ext} are the input and external torques respectively, applied on the aerial vehicle in the body-fixed frame.

The relationships between the input torques and forces is:

$$U = \begin{bmatrix}T\\ \tau_{u_x}\\ \tau_{u_y}\\ \tau_{u_z}\end{bmatrix} = \begin{bmatrix}\sum_{i=1}^{4} k_i{\omega_i}^2\\ l(k_1{\omega_1}^2 - k_2{\omega_2}^2 - k_3{\omega_3}^2 + k_4{\omega_4}^2)\\ l(k_1{\omega_1}^2 + k_2{\omega_2}^2 - k_3{\omega_3}^2 - k_4{\omega_4}^2)\\ \sum_{i=1}^{4}\tau_i(-1)^2\end{bmatrix} \quad (5)$$

Where $k_i{\omega_i}^2$ defines the thrust of the propeller of motor $i$with respect to its angular velocity ω_i , l is the distance from the center of mass to the motor axis of action and $\tau_i$denotes the torque of motor i. Finally, a possible conversion from Euler angles to quaternion can be performed using the following equation:

$$q_d = \begin{bmatrix} \cos\frac{\varphi_d}{2}\cos\frac{\theta_d}{2} \\ \sin\frac{\varphi_d}{2}\cos\frac{\theta_d}{2} \\ \cos\frac{\varphi_d}{2}\sin\frac{\theta_d}{2} \\ -\sin\frac{\varphi_d}{2}\sin\frac{\theta_d}{2} \end{bmatrix} \quad (6)$$

Table 1 presents all the parameters adopted to the quadrotors model used in the simulation.

Table 1 Quadrotors Parameters

Parameter	Value	Unit
I_x	0.00080	$kg.m^2$
I_y	0.00080	$kg.m^2$
I_z	0.0014	$kg.m^2$
l	0.125	m
M	0.26	Kg

Trajectory tracking and control

The proposed control architecture is based on a double loop control strategy, an inner loop for the attitude control and outer loop for the position control. An optimization over the control space is reached by using a differential flatness- quaternion based equations.

Control Architecture

Quadrotors can be considered as a differential system with 4 at outputs, such that: $\mathcal{X} = [x,y,z,\psi]^T$. By assuming zero yaw angle for simplicity, then the mapping from the at outputs to the position, velocity, and acceleration of the quadrotors is:

$$[x,y,z]^T = [\mathcal{X}_1, \mathcal{X}_2, \mathcal{X}_3]^T$$

$$[\dot{x},\dot{y},\dot{z}]^T = [\dot{\mathcal{X}}_1, \dot{\mathcal{X}}_2, \dot{\mathcal{X}}_3]^T \quad (7)$$

A quaternion orientation can be formulated as a rotation about some axis $\hat{n}$, as shown in Eqn (8):

$$q = \begin{bmatrix} \cos\left(\frac{\theta}{2}\right) \\ \hat{n}\sin\left(\frac{\theta}{2}\right) \end{bmatrix} \tag{8}$$

The normalized body frame thrust vector is always $[0;0;1]^T$ and the normalized inertial frame thrust vector is defined in Eqn(9).

$$\hat{F}^I = \frac{1}{\sqrt{\ddot{x}^2 + \ddot{y}^2 + (\ddot{z} - g)^2}} \begin{bmatrix} \ddot{x} \\ \ddot{y} \\ \ddot{z} - g \end{bmatrix} \tag{9}$$

The rotation vector $\hat{n}$ is solved for in Eqn(10)

$$\hat{F}^B . \hat{F}^I = ||\hat{F}^I|| \, ||\hat{F}^B|| \cos(\theta) = \cos(\theta)$$

$$\hat{F}^B \times \hat{F}^I = ||\hat{F}^I|| \, ||\hat{F}^B|| \sin(\theta\hat{n}) = \sin(\theta\hat{n})$$

$$\hat{n} = \frac{\hat{F}^B \times \hat{F}^I}{\sin(\theta)} = \frac{\hat{F}^B \times \hat{F}^I}{\sqrt{1 - \cos^2(\theta)}} = \frac{\hat{F}^B \times \hat{F}^I}{\sqrt{1 - (\hat{F}^{BT} . \hat{F}^I)^2}} \tag{10}$$

Furthermore, the following equations can be derived:

$$\cos\left(\frac{\theta}{2}\right) = \sqrt{\frac{1}{2}(1 + \cos\theta)} = \sqrt{\frac{1}{2}(1 + \hat{F}^{BT} . \hat{F}^I)} \tag{11}$$

$$\sin\left(\frac{\theta}{2}\right) = \sqrt{\frac{1}{2}(1 - \cos\theta)} = \sqrt{\frac{1}{2}(1 - \hat{F}^{BT} . \hat{F}^I)} \tag{12}$$

By substituting Eqn(11) and Eqn(12) into Eqn(8), resulting in the quaternion rotation without a yaw correction $\tilde{q}$. The final quaternion$q$ can then be obtained:

$$\tilde{q} = \frac{1}{\sqrt{2(1 + \hat{F}^{BT} . \hat{F}^I)}} \begin{bmatrix} 1 + \hat{F}^{BT} . \hat{F}^I \\ \hat{F}^B \times \hat{F}^I \end{bmatrix} \tag{13}$$

$$q = \tilde{q} \otimes \begin{bmatrix} \cos\left(\frac{\psi}{2}\right) \\ 0 \\ 0 \\ \sin\left(\frac{\psi}{2}\right) \end{bmatrix} \tag{14}$$

Controller Design

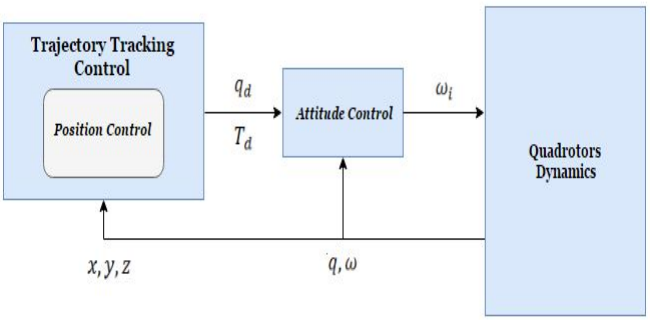
The trajectory tracking controller consists of two parts, an inner control loop namely attitude controller and an outer one namely the position controller as depicted in Figure 2. The outputs of the attitude controller are the desired

angular speeds. The position controller generates the desired position quaternion value $q_d$and the desired trust T_d for the attitude controller.

The Lyapunov function V_A used to design a backstepping attitude controller is given by Eqn(15), where q_{e0} is the quaternion error and e_2 denotes angular velocity error $\eta_d - \eta$.

$$V_A = |q_{e0}| + \frac{1}{2}{e_2}^T e_2 \tag{15}$$

Figure 2 Block diagram of the proposed control structure



The derivative of the Lyapunov function V_A is expressed in Eqn(16).

$$\dot{V_A} = -\frac{1}{2} sign(q_{e0}) {q^T}_{e13} \eta_d + e_2(\dot{\eta}_d - \dot{\eta}) \tag{16}$$

Eqn(17) shows the desired derivative of the Lyapunov function $\dot{V_A}$, which is negative as long as c_{1A} is a positive constant and matrix c_{2A} is a positive definite matrix.

$$\dot{V_A} = -\frac{1}{2} c_{1A} {q^T}_{e13} q_{e13} - {e_2}^T c_{2A} e_2 \tag{17}$$

Assuming the virtual control η_d is expressed by Eqn(18) then the control law τ is expressed by formula Eqn(19):

$$\eta_d = sign(q_{e0}) c_{1A} q_{e13} \tag{18}$$

$$\tau = (c_{2A} e_2 + \dot{\eta}_d) I_q + \tau_{ext} \tag{19}$$

The Lyapunov function V_P used to design the trajectory tracking backstepping controller is given by Eqn(20), where $e_1 = p_d - p$ is the position error and $e_2 = v_d - \dot{p}$ is the velocity error.

$$V_P = \frac{1}{2}{e_1}^T e_1 + \frac{1}{2}{e_2}^T e_2 \tag{20}$$

Given the desired velocity as Eqn(21) and the desired derivative of the Lyapunov function $\dot{V}_P$ as Eqn(22) that is negative as long as c_{1P} and c_{2P} are positive definite matrices, then the control law U is derived from Eqn(23), where $\ddot{p}$ is substituted for Eqn(2).

$$v_d = c_{1P}e_1 + \dot{p}_d \tag{21}$$

$$\dot{V}_P = -e_1{}^T c_{1P} e_1 - e_2{}^T c_{2P} e_2 \tag{22}$$

$$\ddot{p} = e_1\left(I - c_{1P}{}^2\right) + e_2(c_{1P} + c_{2P}) + \ddot{p}_d \tag{23}$$

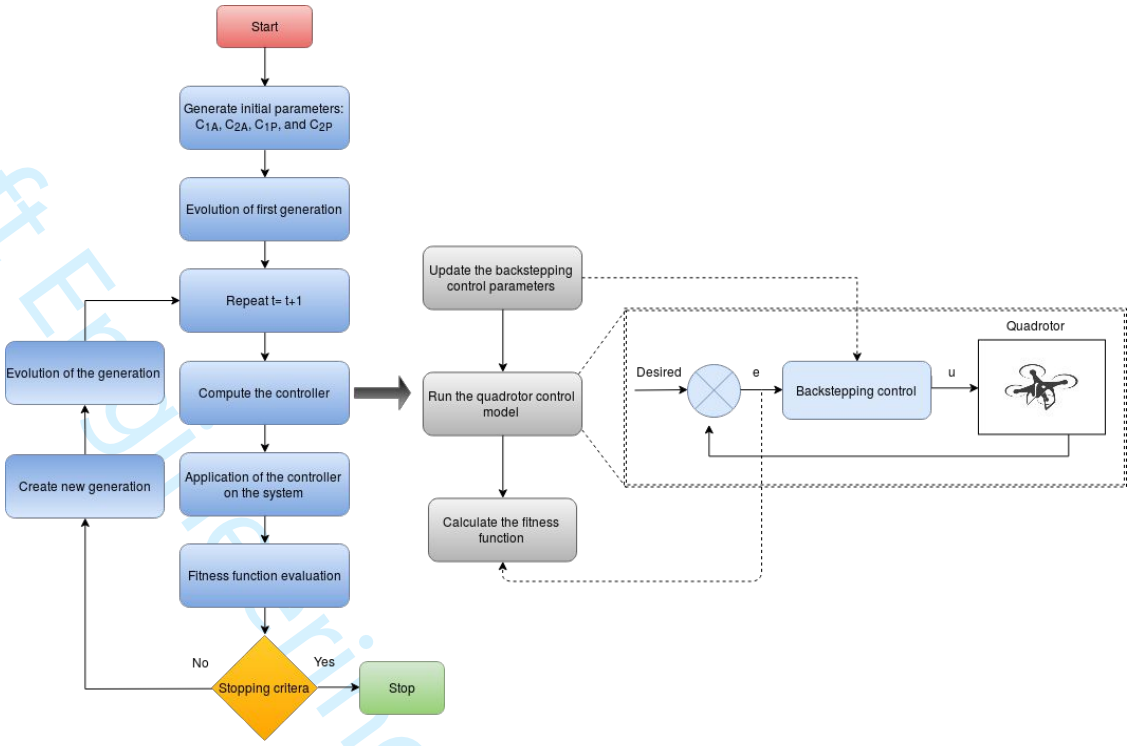
MO-GA Optimization

In the present study, The MO-GA is utilized offline to determine the backstepping controller parameters. Thus all the control parameters are selected simultaneously so that each subsystem is asymptotically stable. The fitness functions defined in Eqn(24) allow optimizing not only the tracking errors due to steady state errors and the overshoot but also the consumed energy:

$$\begin{cases} f_q = \dfrac{1}{t_f - t_i}\displaystyle\int_{t_i}^{t_f} (q - q_d)^T (q - q_d)dt \\ f_\eta = \dfrac{1}{t_f - t_i}\displaystyle\int_{t_i}^{t_f} (\eta - \eta_d)^T (\eta - \eta_d)dt \\ f_p = \dfrac{1}{t_f - t_i}\displaystyle\int_{t_i}^{t_f} (p - p_d)^T (p - p_d)dt \\ f_v = \dfrac{1}{t_f - t_i}\displaystyle\int_{t_i}^{t_f} (v - v_d)^T (v - v_d)dt \end{cases} \tag{24}$$

With f_q,f_η,f_p and $f_v$are the quaternion, the angular speed, the position and the linear speed fitness functions respectively. $t_i$andt_f denote the initial and the final instants respectively.

Figure 3 MO-GA Optimization Algorithm



As illustrated in Figure 3 the performance of the controller varies according to adjusted parameters. The coefficients matrices c_{1A},c_{2A},$c_{1P}$$and$$c_{2P}$ are the control parameters related to f_q,f_η,f_p and f_v respectively that need to be positive to satisfy stability criteria. The integral absolute error (IAE) is utilized to judge the performance of the controller, So that the optimization algorithm as long as it achieves a desired IAE. The IAE criterion is used because it is simple to implement and widely adopted to evaluate the dynamic performance of the control system. The index IAE is expressed as: $E = \int_0^t |e(t)|dt$.

Our controller is designed respecting several constraints namely, bounded inputs, bounded rate of velocities and some mathematical singularities. The obtained control parameters are shown on Table 2

Table 2 Control Parameters

Controller	c1	c2
Attitude	[0.8 0.8 0.5]	[0.1 0.1 1.2]
Position	[0.2 0.2 3]	[0.1 0.1 1.5]

Trajectory optimization and obstacles avoidance

From the differentially flat equations the problem is to allow optimizations to occur within the output space as opposed to the control space. This is beneficial because constraints such as obstacle avoidance occur in the output space, hence the computation time for constraint handling is reduced. The problem is posed as follows:

$$\min \Phi \quad \text{for } t \in [0, T]$$

$$y(t)$$

$$s.t.\ c_y(y) \leq 0$$

$$x_0 - h_1(y(0)) = 0$$

$$y_T - y(T) = 0 \tag{25}$$

Where the inequality constraints are now expressed as a function of the output $c_y(y)$ and the state is now a function of the output obtained from the differential flatness$h_1(y)$.

The objective function, Φ, is a quantitative measure of the optimality of the trajectory, which, can be approximated by a measure of the running costs. Assuming running costs are proportional to average velocity then the objective function can be defined as:

$$\Phi = \frac{1}{T}\int_0^T \sqrt{(P_1 \dot{x}^2 + P_2 \dot{y}^2 + P_3 \dot{z}^2)}dt \tag{26}$$

Where P_1, P_2, P_3 are weighting factors.

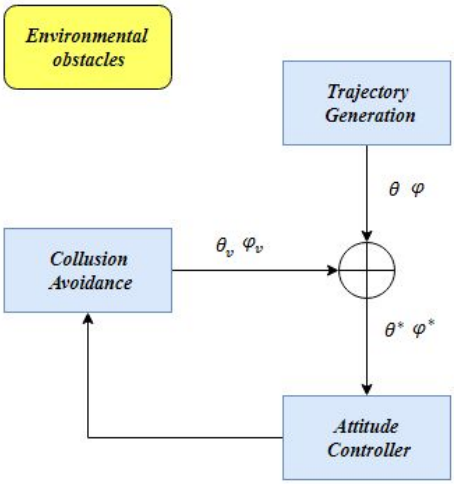
To prevent UAVs from destroying themselves and surrounding objects (other vehicles, personnel, and infrastructures), a collision avoidance mechanism should be generally incorporated into the control design. As illustrated in Figure 4, the primary idea of this technique is to modify the roll and the pitch angles while keeping the separation between the UAV and the obstacle. The essence of this technique can also be mathematically expressed as follows:

$$\theta^* = \theta + \theta_v$$

$$\varphi^* = \varphi + \varphi_v \tag{27}$$

Where θ^* and φ^* denote the ultimately desired roll and pitch angles after modification, θ_v and φ_v represent the reactive obstacle avoidance terms for roll and pitch angles of each UAV.

Figure 4 Obstacle avoidance algorithm



Simulations results

In this section the simulation results related to the quadrotors optimal trajectory generation and control discussed in the other sections is shown.

The controller was simulated at a rate of 200 Hz which makes it suitable for a real implementation. All the necessary limitations over the actuators and energy consumption were taking into consideration. For all the simulation cases both controllers (LQR and OBS) are applied and compared. Five scenarios for different situations of the quadrotors drone have carried out as follow:

1- *Scenario1:* The quadrotors starts from an initial position and maintain its position in another points until the battery is discharged.

2- *Scenario2:* In this case the quadrotors is tracking a circular path in the presence of an external wind gust disturbance.

3- *Scenario3:* For this scenario the quadrotors is facing a sudden engine failure for a few seconds.

4- *Scenario4:* During this case the quadrotors is scanning a long range area at a constant altitude.

5- *Scenario5:* The quadrotors UAV is facing a circular obstacle so it had to avoid it then continue the desired mission.

Scenario 1

As mentioned before, in this case the quadrotors from a point $P_0(x,y,z) = (0,0,0)$ and travel to anothor point $P_1(x,y,z) = (1,1,1)$ and hold it position until the batterie is discharged. This first scienario was chossen to test the optimal trajectory generation and the energy optimization.

Figure 5 Trajectory for scenario 1

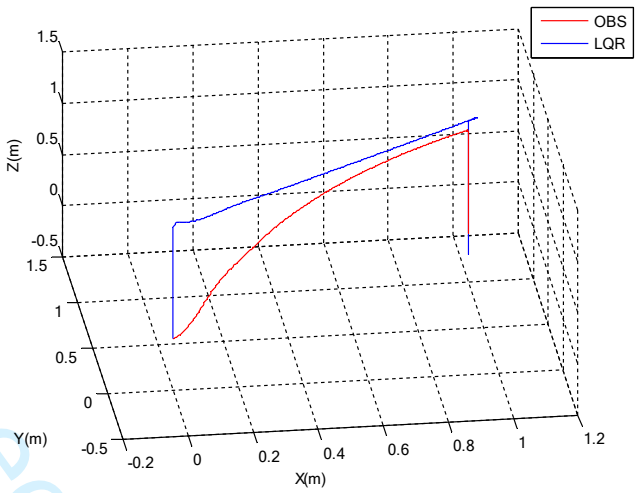
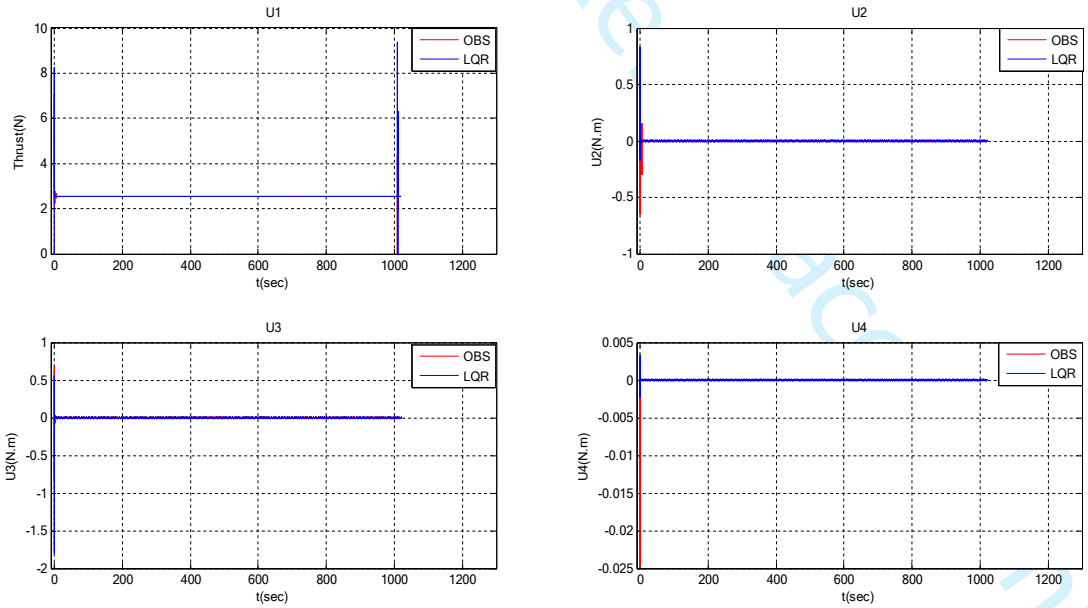


Figure 5 shows the obtained results for the trajectory generation of the LQR and the OBS controller where it is clear that the OBS controller is following a more optimal 3D path and presents a more accurate control when comparing to the LQR controller (overshoots in the X-Y plan). For the emergency landing due to the battery discharging the LQR controller is more energy saving technique due to the low magnitude in the control inputs (see Figure 6) but both controllers were able to execute a safe emergency landing, and prevent the quadrotors from the destruction.

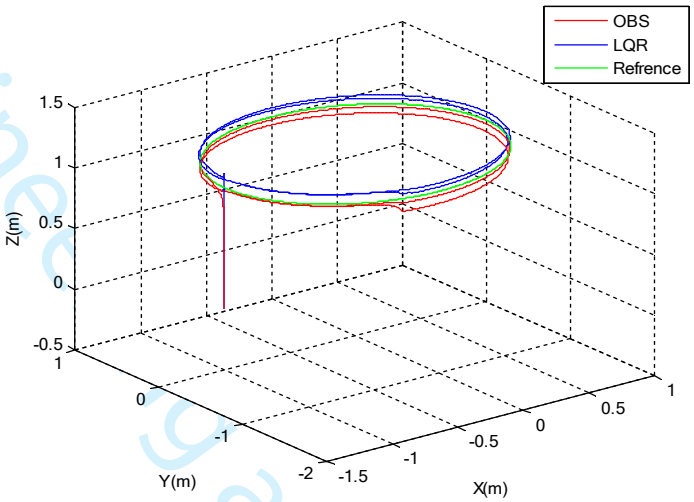
Figure 6 Control Inputs for scenario 1



Scenario 2

In this scenario the quadrotors UAV is sacked to follow a circular path of a 1m diameter and at 1m of altitude in the presence of a wind gust perturbation. This kind of scenarios allows testing the robustness of the controllers. The obtained results are shown in Figure 7.

Figure 7 Trajectory Tracking in the presence of a wind gust

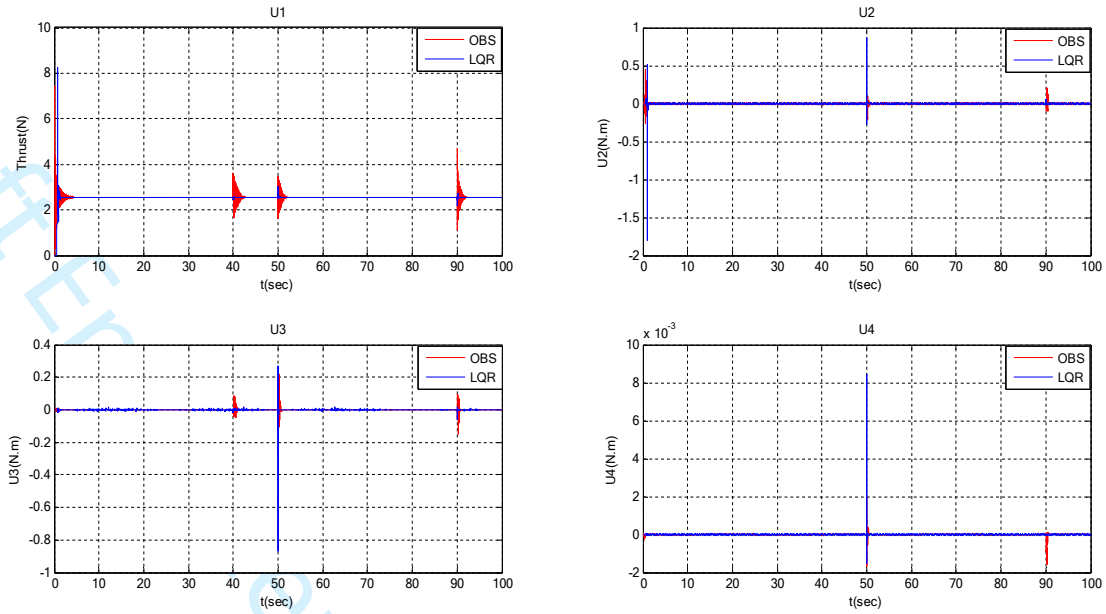


In this case the quadrotors is facing a wind gust. The perturbation is applicated twice a time for a 40 sec during the simulation, a recovery period of 10 sec is given to the quadrotrs to stabilize.

Figure 7 introduces the results obtained during the trajectory tracking in the presence of the wind gust perturbation. The LQR and OBS controller were able to stabilize the quadrotors and accomplish the missions but with a better accuracy for the OBS controller in the X-Y plan and a good one of the LQR controller in the altitude hold.

For the control inputs shown in Figure 8 the OBS controller presents a more aggressive maneuvers in each time the perturbation starts of finish but with a less magnitude when compared to the LQR control inputs.

Figure 8 Control Inputs for scenario 2



Scenario 3

This scenario is dedicated to simulate the attitude stability of the quadrotors after a sudden engine failure of the number 01 engine. The fault starts from the 15 sec and still for 3 sec, the UAV is supposed to be at 10m of altitude. For the scenario the LQR controller was not able to execute this task, and the quadrotors was not able the recover its altitude so it crashes.

All the results are using only the OBS controller. Figure 9 indicates the altitude response of the quadrotors, while Figure 10 shows the control inputs respectively.

From the obtained results it can be noticed that the quadrotors was able to recover its altitude and stabilize after the sudden engine failure even with some lose of the altitude (go down to less than 6 m),

As mentioned before the OBS controller was the only controller to execute this task, this was due to its ability of big attitude degrees tracking (up to 1 rad) using an aggressive control inputs with high rate changes (Figure 10) that cannot be generated using a linear LQR controller.

Figure 9 Altitude response for scenario 3

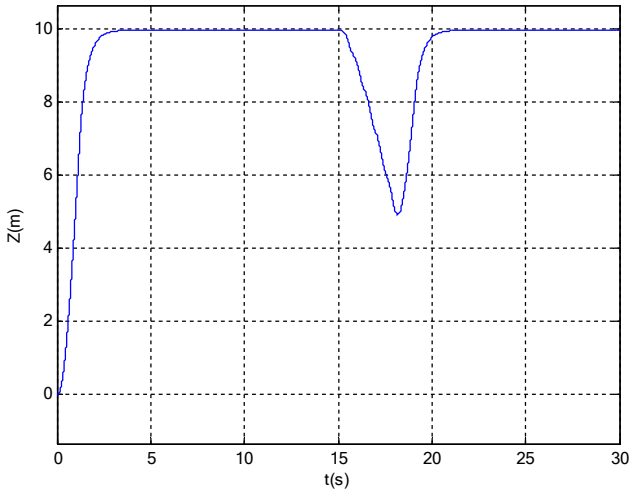
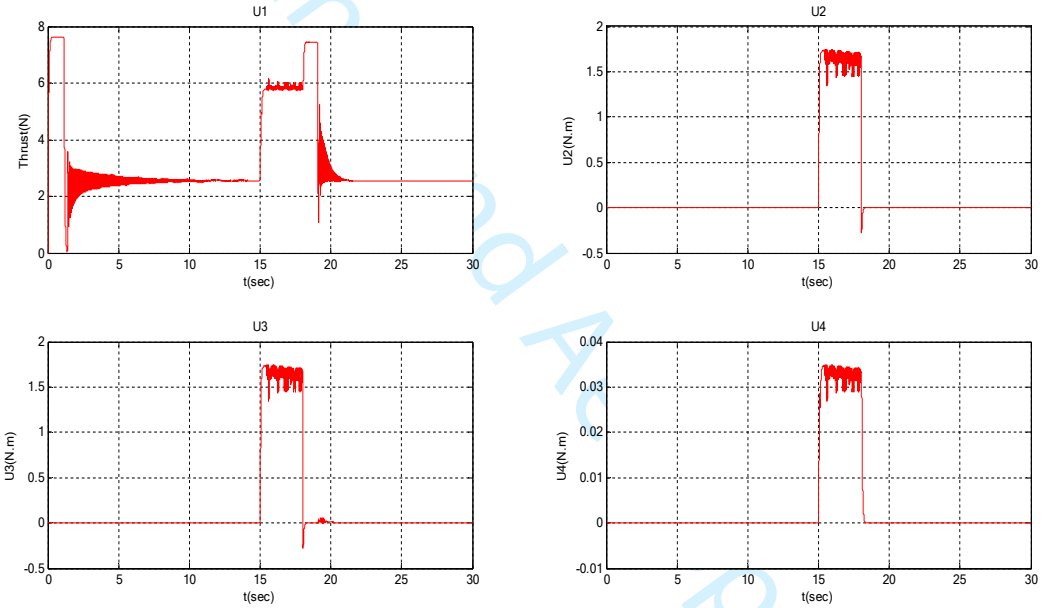


Figure 10 Control Input for scenario 3



Scenario 4

For this scenario the quadrotors is scanning a large area by tracking a rectangular shape (1000 m * 1000 m) at 10m of altitude, the goal behind this scenario is to test the energy optimization of the used controllers during the trajectory tracking. Figure shows the obtained 3D trajectory tracking.

Figure 11 Trajectory tracking for scenario 4

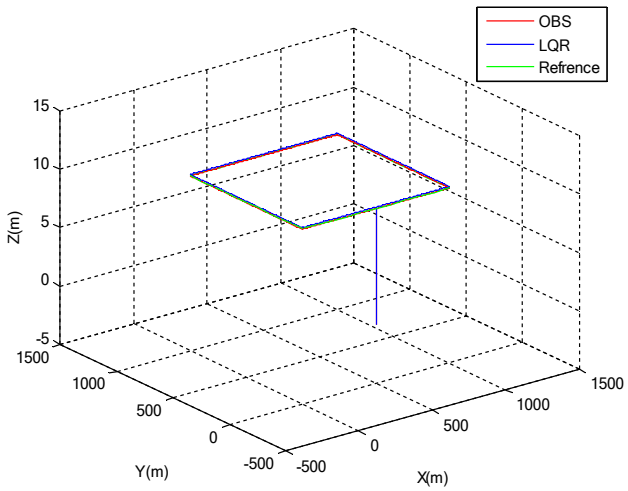
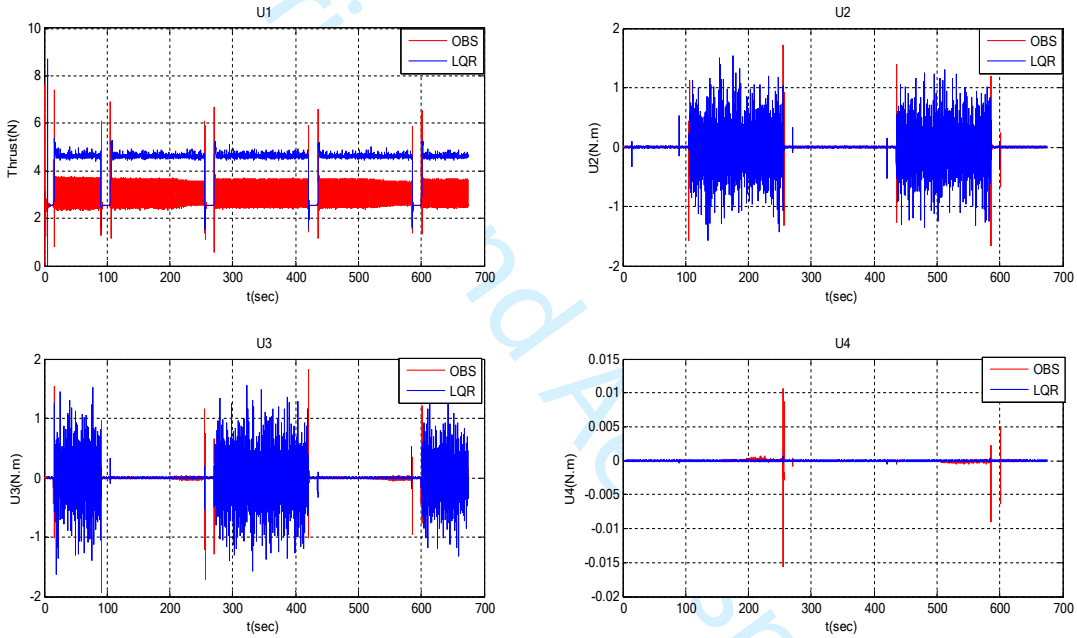


Figure 12 Control Inputs for scenario 4



It is clear that the OBS controller presents a more accurate results compared to the LQR controller, this is due to the high attitude degree (up to 1 rad) used by the OBS controller during the curving movements, but with less control inputs magnitude (about 3N) during rectilinear movements (See Figure 12).

The total energy consumption is estimated to be less for the OBS controller compared to the LQR controller.

Scenario 5

For this last scenario the quadrotors is facing a circular obstacle after starting from an initial point P_0 $(x,y) = (50,35)$ and traveling to anothor point $P_1(x,y) = (51,65)$ in the horizontal plan.

It is important to mention that the LQR controller was not able to execute this task, so all the results are using only the OBS controller.

From Figure 13 it can be noticed that the quadrotors was able to track an optimal trajectory to reach the desired destination, and avoid the collision with the obstacle.

Figure 14 presents the obtained control inputs. Those results reflect the high performance of the OBS controller to aviod the obtacle with high acuraccy and a minimum of energy during a short time.

Figure 13 Obstacle avoidance

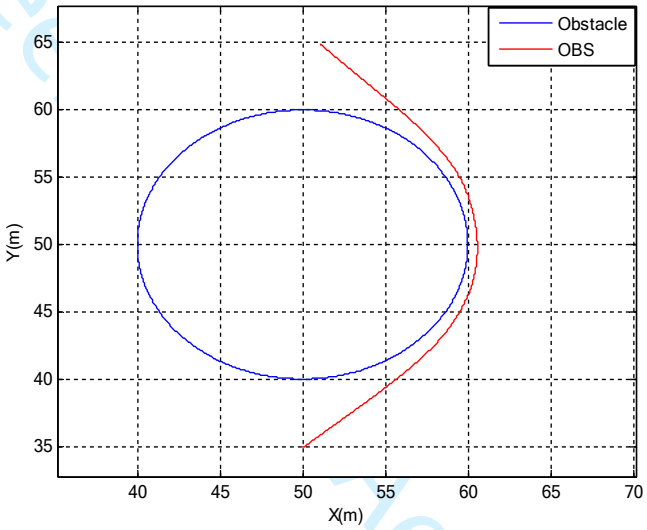
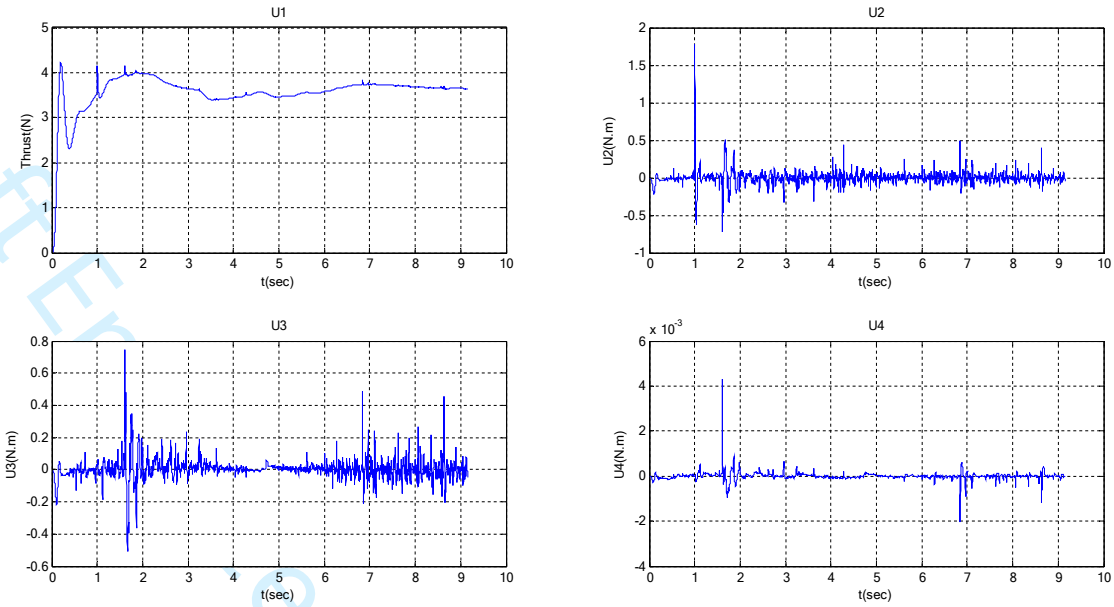


Figure 14 Control Inputs for scenario 5



Control strategies comparison

In this section a comparison between the different control strategies is made, the obtained results are shown in Table 3.

Table 3 Control strategies comparison

Scenario/Controller	Fitness Function (IAE)	Battery Consumption (%)	Mission Time (sec)
Scenario 1			
LQR	0.0867	100	1020
OBS	0.0234	100	1020
Scenario 2			
LQR	0.1215	26.08	100
OBS	0.0718	24.22	100
Scenario 3			
LQR	-	-	-
OBS	-	8.4	30
Scenario 4			

LQR	0.1026	35.74	675
OBS	0.0145	33.40	675
Scenario 5			
LQR	-	-	-
OBS	-	5.6	9.26

From Table 3 it is clear that the OBS controller is the only controller that was able to execute all the proposed scenarios (The LQR controller was not able to avoid the obstacle and to recover the stability after an engine failure).

During all the scenarios the energy optimization of the controllers is improved, since the consumed energy is estimated to be optimal compared to the mission time (The quadrotors is able to still for 17 min at a hovering movement before the emergency landing – mission 1 -) spatially for mission 4 where the quadrotors was able to scan a large area with only 35 % of the battery energy.

For the controllers tracking accuracy, the OBS is clearly the most accurate controller with a very interesting IAE coefficients (less than 0.1 during all the scenarios) even when the UAV is facing a wind perturbation (0.1215 for LQR and 0.0718 for OBS) which proof the controllers robustness.

Conclusion

In this paper the problem of trajectory generation and control was investigated using a new multi-layered optimal navigation system. The proposed approach has, jointly, optimizes the energy consumption, improves the robustness and raises the performance of a quadrotors unmanned aerial vehicle (UAV).

First, the control architecture was designed based on a double loop control strategy, an optimization over the control space was reached using a differential flatness- quaternion based equations. Then, the optimal reference path was generated, and tracked using a non-linear optimal backstepping controller (OBS). Finally, many scenarios were proposed, all the obtained results are judged to be satisfactory. Optimization, accuracy and robustness of the designed system were demonstrated.

The next step for this research will be the implementation of these algorithms in a real quadrotors. Some real application scan be also considered.

References

Basri, M., Ariffanan, M., Danapalasingam, K. A., & Husain, A. R. (2014). Design and optimization of backstepping controller for an underactuated autonomous quadrotor unmanned aerial vehicle. *Transactions of FAMENA*, *38*(3), 27-44.

Bouzid, Y., Siguerdidjane, H., Bestaoui, Y., & Zareb, M. (2017). Energy based 3D autopilot for VTOL UAV under guidance & navigation constraints. *Journal of Intelligent & Robotic Systems*, *87*(2), 341-362.

Carino, J., Abaunza, H., & Castillo, P. (2015, June). Quadrotor quaternion control. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on* (pp. 825-831). IEEE.

CHOUTRI, K., LAGHA, M., DALA, L., & LIPATOV, M. (2018, October). Quadrotors UAVs Swarming Control Under Leader-Followers Formation. In *2018 22nd International Conference on System Theory, Control and Computing (ICSTCC)* (pp. 794-799). IEEE.

Chovancová, A., Fico, T., Hubinský, P., & Duchoň, F. (2016). Comparison of various quaternion-based control methods applied to quadrotor with disturbance observer and position estimator. *Robotics and Autonomous Systems*, *79*, 87-98.

Cowling, I. D., Whidborne, J. F., & Cooke, A. K. (2006, August). Optimal trajectory planning and LQR control for a quadrotor UAV. In *International Conference on Control*.

Djamel, K., Abdellah, M., & Benallegue, A. (2016). Attitude optimal backstepping controller based quaternion for a UAV. *Mathematical Problems in Engineering*, *2016*.

Huo, X., Huo, M., & Karimi, H. R. (2014). Attitude stabilization control of a quadrotor UAV by using backstepping approach. *Mathematical Problems in Engineering*, *2014*.

Kamel, M., Alexis, K., Achtelik, M., & Siegwart, R. (2015, September). Fast nonlinear model predictive control for multicopter attitude tracking on SO (3). In *Control Applications (CCA), 2015 IEEE Conference on* (pp. 1160-1166). IEEE.

Kehlenbeck, A. G. (2014). Quaternion-based control for aggressive trajectory tracking with a micro-quadrotor UAV(Doctoral dissertation).

Larbi, M. A., Meguenni, K. Z., Meddahi, Y., & Litim, M. (2013). Nonlinear observer and backstepping control of quadrotor unmanned aerial vehicle. *International Review of Aerospace Engineering (IREASE)*, *6*(5), 233-242.

Liu, H., Wang, X., & Zhong, Y. (2015). Quaternion-based robust attitude control for uncertain robotic quadrotors. *IEEE Transactions on Industrial Informatics*, *11*(2), 406-415.

Liu, Z. X., Yu, X., Yuan, C., & Zhang, Y. M. (2015, June). Leader-follower formation control of unmanned aerial vehicles with fault tolerant and collision avoidance capabilities. In *Unmanned Aircraft Systems (ICUAS), 2015 International Conference on* (pp. 1025-1030). IEEE.

Mohd Ariffanan Mohd Basri Abdul Rashid Husain Kumeresan A. Danapalasingam , (2015),"GSA-based optimal backstepping controller with a fuzzy compensator for robust control of an autonomous quadrotor UAV", Aircraft Engineering and Aerospace Technology: An International Journal, Vol. 87 Iss 5 pp. 493 – 505.

Pena, M., Vivas, E., & Rodriguez, C. (2012). Simulation of the Quadrotor controlled with LQR with integral effect. In *ABCM Symposium Series in Mechatronics* (Vol. 5, No. 1, pp. 390-399).

Sun, L., & Zuo, Z. (2015). Nonlinear adaptive trajectory tracking control for a quad-rotor with parametric uncertainty. *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering*, *229*(9), 1709-1721.

Tanveer, M.H., Ahmed, S.F., Hazry, D., Warsi, F.A. and Joyo, M.K. (2013), "Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions", American Journal of Applied Sciences, Vol. 10 No. 8, pp. 819-831

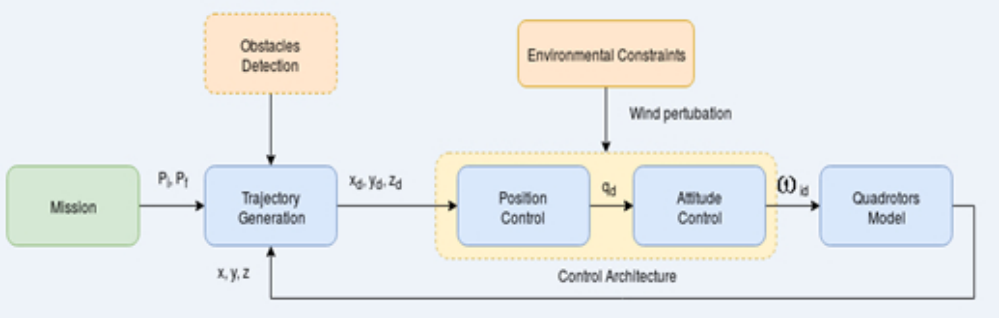
Wang, S., & Yang, Y. (2012, July). Quadrotor aircraft attitude estimation and control based on kalman filter. In *Control Conference (CCC), 2012 31st Chinese* (pp. 5634-5639). IEEE.

Yang, Y., & Yan, Y. (2016). Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control. *Aerospace Science and Technology*, *54*, 208-217.

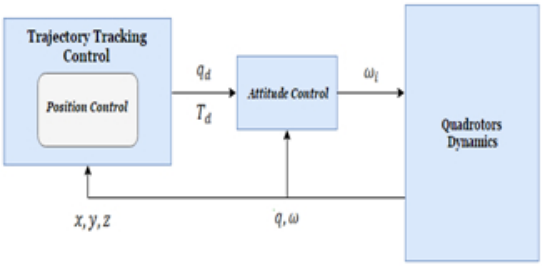
Yang, Y., & Yan, Y. (2016). Neural network approximation-based nonsingular terminal sliding mode control for trajectory tracking of robotic airships. *Aerospace Science and Technology*, *54*, 192-197.

Yang, Y. N., Wu, J., & Zheng, W. (2012). Trajectory tracking for an autonomous airship using fuzzy adaptive sliding mode control. *Journal of Zhejiang University SCIENCE C*, *13*(7), 534-543.

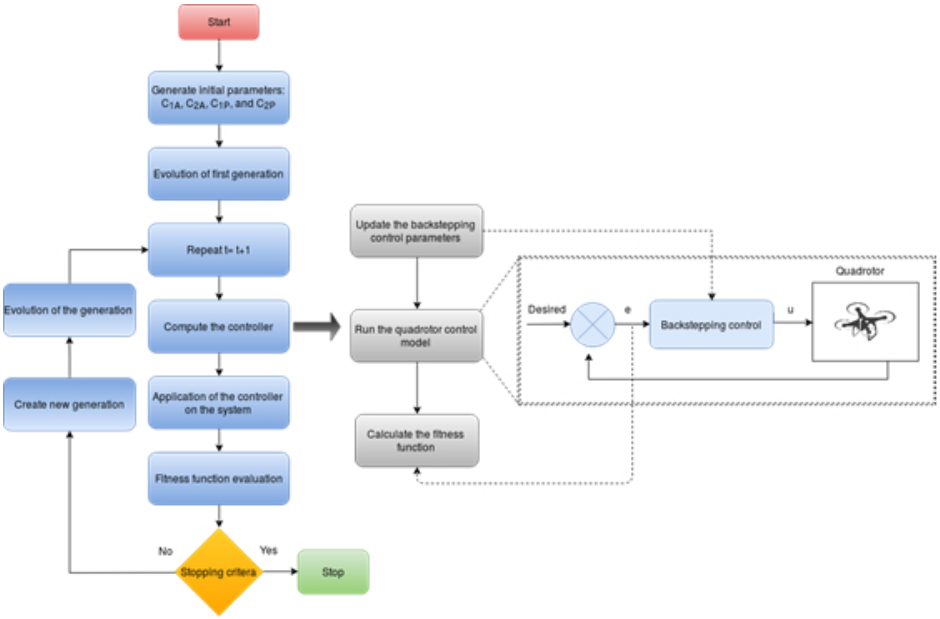
Zhang, X., Li, X., Wang, K., & Lu, Y. (2014). A survey of modelling and identification of quadrotor robot. In *Abstract and Applied Analysis* (Vol. 2014). Hindawi.



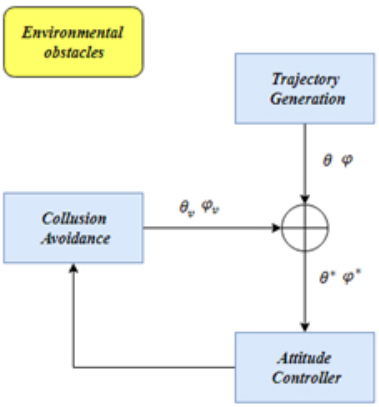
Multi-layer Optimal Navigation System



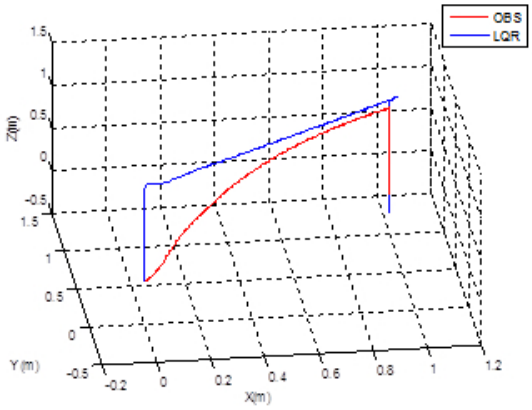
Block diagram of the proposed control structure



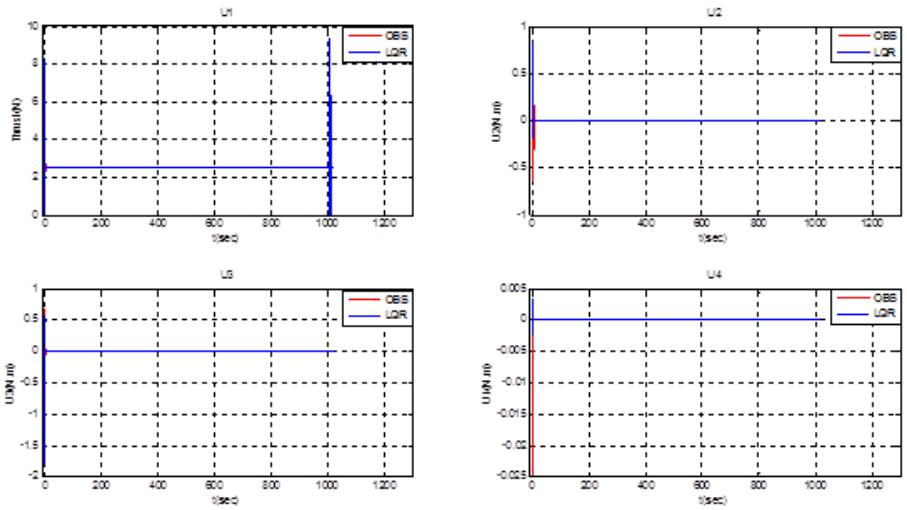
MO-GA Optimization Algorithms



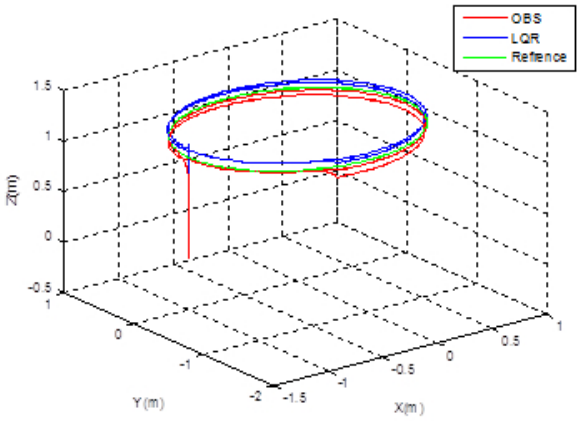
Obstacle avoidance algorithm



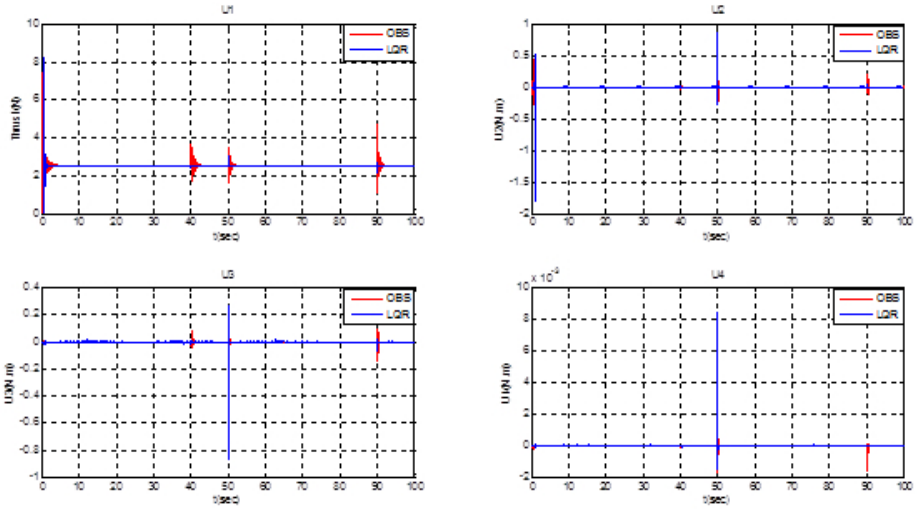
Trajectory for scenario 1



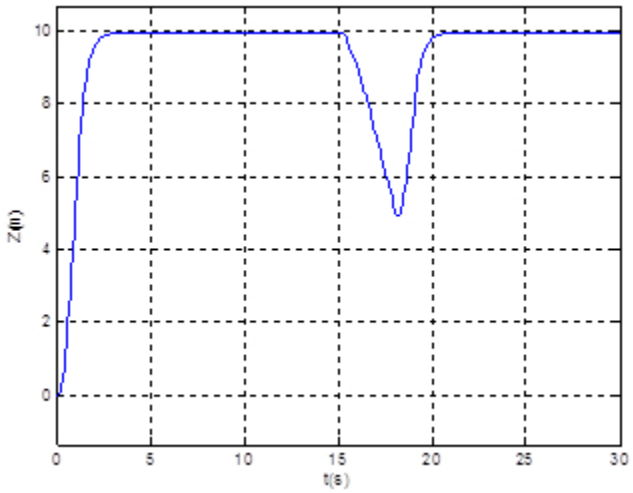
Trajectory for scenario 1



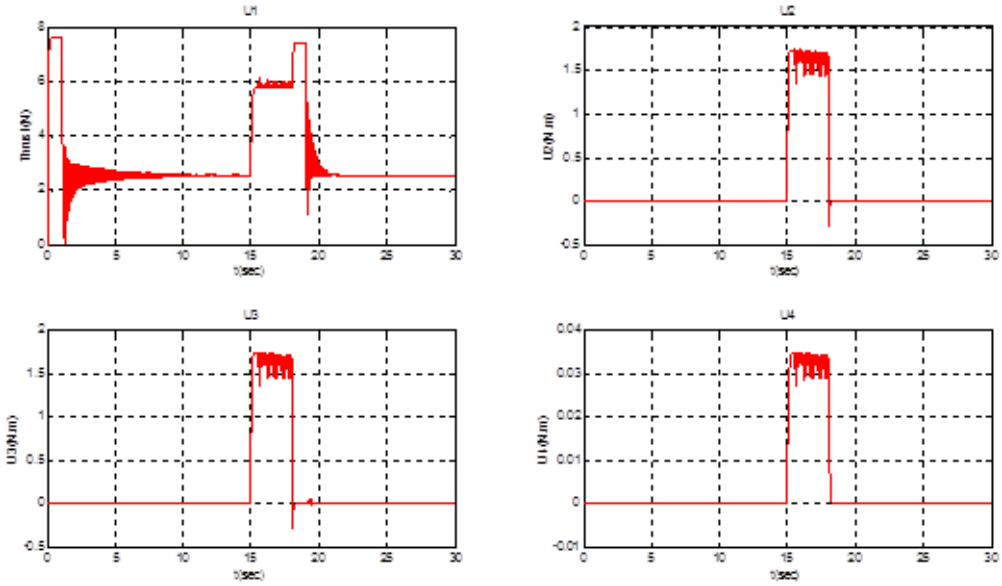
Trajectory Tracking in the presence of a wind gust



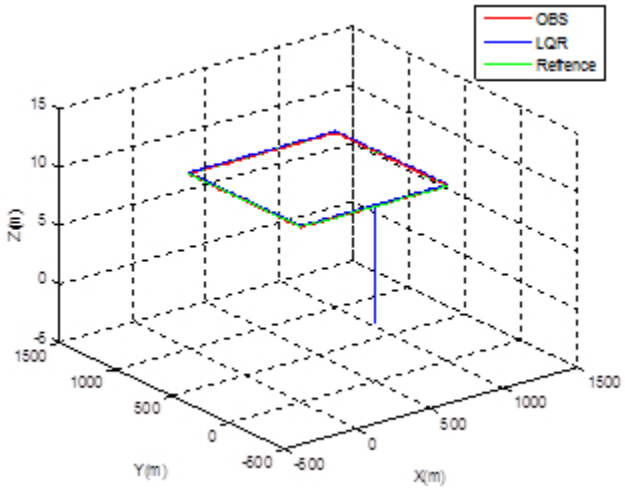
Control Inputs for scenario 2



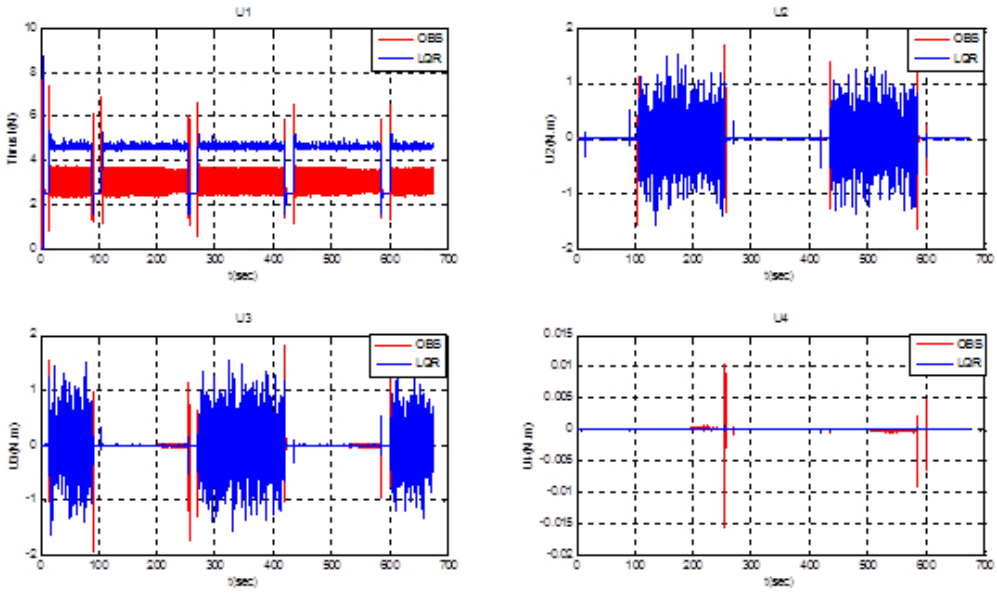
Altitude response for scenario 3



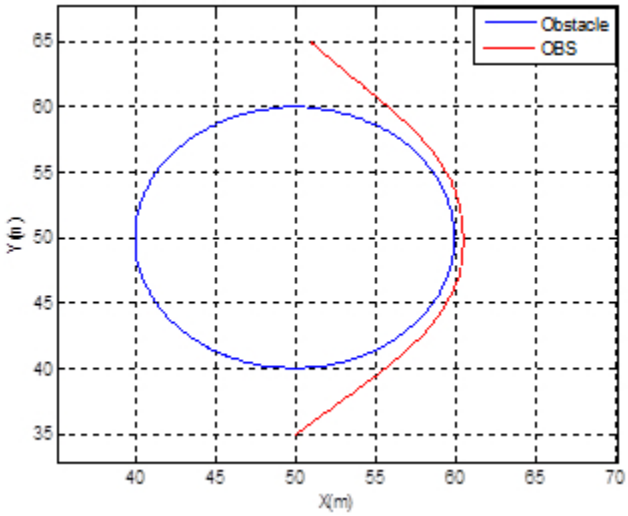
Control Input for scenario 3



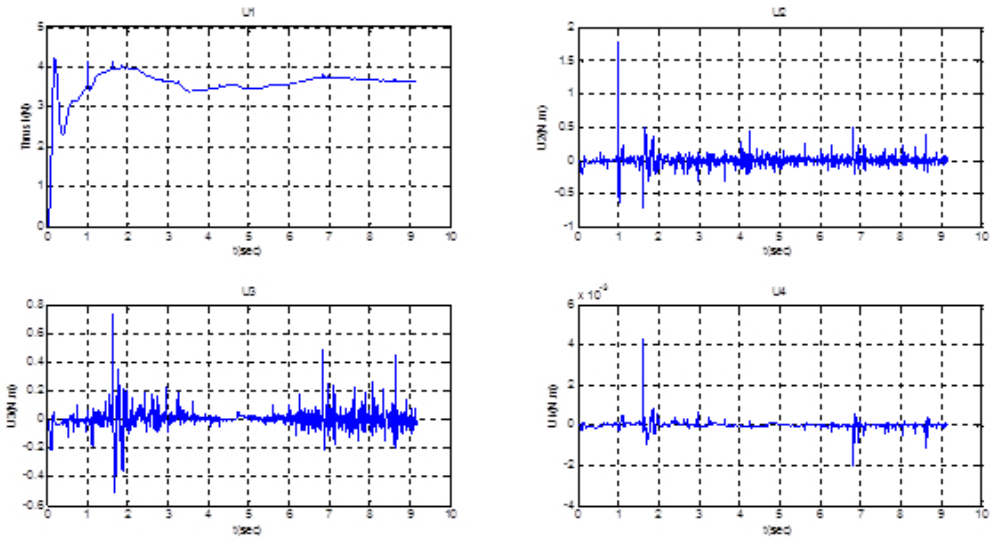
Trajectory tracking for scenario 4



Control Inputs for scenario 4



Obstacle avoidance



Control Inputs for scenario 5

Parameter	Value	Unit
I_x	0.00080	kg.m^2
I_y	0.00080	kg.m^2
I_z	0.0014	kg.m^2
l	0.125	m
M	0.26	Kg

Quadrotors Parameters

Controller	c1	c2
Attitude	[0.8 0.8 0.5]	[0.1 0.1 1.2]
Position	[0.2 0.2 3]	[0.1 0.1 1.5]

Control Parameters

Scenario/Controller	Fitness Function (IAE)	Battery Consumption (%)	Mission Time (sec)
Scenario 1			
LQR	0.0867	100	1020
OBS	0.0234	100	1020
Scenario 2			
LQR	0.1215	26.08	100
OBS	0.0718	24.22	100
Scenario 3			
LQR	-	-	-
OBS	-	8.4	30
Scenario 4			
LQR	0.1026	35.74	675
OBS	0.0145	33.40	675
Scenario 5			
LQR	-	-	-
OBS	-	5.6	9.26

Control strategies comparison