Proceedings of the 6th RSI
International Conference on Robotics and Mechatronics (IcRoM 2018)
October 23-25, 2018, Tehran, Iran

Optimal path tracking of a quadrotor in the presence of obstacle using the League Championship Algorithm

Ehsan Abbasi, Mostafa Ghayour, Mohammad Danesh, Mohammad Hussein Yoosefian
Department of Mechanical Engineering
Isfahan University of Technology
Isfahan, Iran
ehsan.abbasi@me.iut.ac.ir

***Abstract*— Nowadays, cost reduction and optimization has attracted attention of researchers and scientists in many fields. Optimal path tracking for vehicles, which reduces fuel consumption, as well as the distance traversed by the vehicle, has mostly caught attention. Besides, Unmanned Arial Vehicles are in the spotlight because of their great performance in doing a vast majority of jobs, especially those which are perilous for human. One of the most prevailing issues in this field is optimal path tracking. Although different algorithms are presented and despite their acceptable performance, none of them has focused on reducing energy and distance. In this paper a new optimization algorithm is utilized for the latter optimizations. This algorithm is called the Championship Algorithm by its Iranian developer. Combining it with the path tracking problem of the unmanned aerial vehicles, it is shown that its performance is much better than one of the most important existing algorithms (GA). Optimal trajectory tracking problem is also solved to reduce fuel and energy consumption.**

Keywords— Quadrotor, Path optimization, Path tracking, League championship algorithm

I. INTRODUCTION

An unmanned aerial vehicle (UAV) refers to any aircraft which doesn't require pilot. Quadrotor is one known UAV that can be utilized in various cases. Simple structure and uncomplicated rules of motion compared to helicopters are two main reasons of its distinction. In recent years, study on UAV's tracking algorithms has become a hot topic to researchers. Finding a safe path in perilous flying space is an indisputable necessity for quadrotor in order to do its assigned tasks. Thus it's urgent to study tracking algorithms in order to provide a safe path for quadrotor among obstacles, so it can move from start point to final point while avoiding obstacles. Besides, the provided path should be optimized to reduce the operation time, communication interruption and fuel consumption. One of the very first research conducted in path tracking, is presented in [1].

Several applicable research in quadrotor control are provided in former studies. One of these emerged methods is the real time substitution of the final point position in several stations sequenced in the path. One prevailing strategy is PD controller. Many scholars used PD controller in path tracking [2]. Some other works done in this field are sliding mode control, feedback linearization method and neural network strategy [3-5]. Nevertheless, other methods in trajectory tracking are described such as the adaptive control applied in [6] and two step trailing and obstacle avoidance based on polygon used in [7]. In other works a guidance law is emerged in which a virtual goal is defined and the quadrotor attempts to track the goal while keeping its distance fixed from this virtual goal [8-9]. Although the simulation results indicates the highly resistance of the controller against disturbances, yet it uses a large number of parameters that are needed to be set. Likewise, a direct adaptive controller is provided in [10] to track the path, but it also utilizes a virtual goal. A new nonlinear guidance logic for route tracking is proposed in [11] that has better performance in curved path compared to the prior linear methods. Nonetheless, none of the mentioned works concentrated on optimizing the tracking time.

Nowadays, using genetic algorithm to optimize path tracking is in spotlight. As an example, optimal trajectory tracking based on genetic algorithm is discussed for various mobile robots in [12-14]. However, none of these research use dynamic equations of the robot, but they deal with trajectory planning, itself.

There are several solutions for tracking problems like Cell decomposition, Road map and Potential field which are described in [15-17]. Nearly all of these solutions are on spatial configuration basis and show miniscule robust and adaptive behavior. The A^* algorithm [18] is one the alternative ways of optimal tracking in grid decomposed maps used in mobile robots. The D^*algorithm [19] is another approach which is very similar to the previous one and could be used for optimal tracking in the presence of uncertainties. In [20], another method is described that uses genetic algorithm to track in two dimensional spaces. In recent years, several research are conducted in optimal tracking for UAVs like optimal algorithm and quadrotor control using a simple PID structure which is presented in [21]. Also another method is presented in [22] wherein an arbitrary path for quadrotor is optimized.

The objective of this work is to use a new algorithm called the League Championship Algorithm (LCA) which is provided for optimizing nonlinear continues functions. This algorithm is used to optimize trajectory of a quadrotor and then is compared to genetic algorithm, and its better performance is declared.

The paper is structured as follows. The concept of the LCA and some of its details are presented in first section. Then, quadrotor dynamics and equations of motion are presented. the UAV tracking problem and particularly the optimal tracking problem is defined and discussed then. In the next section this problem is solved using LCA algorithm and finally, a comparison between LCA algorithm and genetic algorithm is provided.

978-1-7281-0127-9/18/$31.00 ©2018 IEEE
Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

II. LEAGUE CHAMPIONSHIP ALGORITHM (LCA)

League Championship Algorithm (LCA) is a stochastic population based algorithm for continues global optimization that imitates a championship milieu in which virtual teams play in an artificial league for several weeks. According to the league's schedule in each week, a number of individuals group and play in pairs and their game result is determined in terms of win or loss (or tie) and according to the playing strength (fitness value) along with the formation (solution) is developed by each team. Modeling match analysis, each team devises the necessary changes in its formation (a new solution formation) for next week rivalries and the championship continues for several seasons (stop condition). The algorithm also includes an end season transfer of players modeling. This extra module is used to speed up the convergence of the algorithm.

A. League Championship Algorithm steps

Some characteristics of the regular championship milieu are mentioned in the following.

Rule 1. It is more likely that a team with better playing strength wins the game. The term "playing strength" refers to ability of one team to beat another team [23].

Rule 2. The outcome of a game is not foretellable given known the teams playing strength perfectly. [23]

Rule 3. The probability that team *i* beats team *j* is assumed equal from both teams point of view. [23]

Rule 4. The outcome of the game is only win or loss. Tie outcome is not considered in the basic version of LCA [23].

Rule 5. When team *i* beats team *j*, any strength helped team *i* to win has a dual weakness caused team *j* to lose. In other words, any weakness is a lack of a particular strength. An implicit implication of this rule is that while the match outcome is imputed to chance, teams may not believe it technically. [23]

Rule 6. Teams only focus on their upcoming match without regards of the other future matches. Formation settings are dine just based on the previous week events. [23]

The algorithm steps are presented as flowchart in Fig 1. Practically, a representation should be chosen for population individuals. In LCA, a team formation (solution) is determined with a $1 \times n$ vector (n is number of problem parameters or variables). Each parameter is assigned to a specific player and illustrates the value of the corresponding problem variable. It can be conceived that a change in a parameter value indicates a change in the player's job in the new formation. Let $f(X = (x_1.x_2.\dots.x_n))$ be a numerical function of n variable that should be minimized in decision space that is defined in R^n. A formation (solution) for team *i* at week *j* could be represented by $X_i^t = (x_{i1}^t.x_{i2}^t.\dots.x_{in}^t)$ with $f(X_i^t)$ as its function. Back to the algorithm, this value is the playing strength of the team with X_i^t formation. The best experienced formation by team *i* until week *t* is determined by $B_i^t = (b_{i1}^t.b_{i2}^t.\dots.b_{in}^t)$ that is correspondent to best playing strength. To find B_i^t a selection is employed based on the $f(B_i^{t-1})$ and $f(X_i^t)$ values. For more detailed information on how the league is scheduled, how to determine the winner or loser and how to set up a new formation, see [23].

1) Generating the league schedule

The first step to simulate a championship environment is to schedule matches in each season. For a league consisting *L* teams, there will be $L(L-1)/2$ matches in round robin tournament. Imagine a league with $L=8$ teams, a number is assigned to each team and they will play in pairs in the first week (Fig 2a). Fig 2a demonstrates that in the first week, team 1 plays team 8, 2 plays 7, 3 plays 6 and 4 plays 5. In the second week, team 1 is fixed and the other teams rotates clockwise (Fig 2b). In this week, 1 plays 7, 8 plays 6 and so forth. For the third week, again all teams rotate excluding team 1 which is fixed, so team 1 plays 6, 7 plays 5 and so on (Fig 2c). This process continues until getting to the initial state. The last week schedule is shown in Fig 2d. If *L* is odd, a dummy is added. In each week, the opponent of the dummy team does not play and gets rest [23]. The championship goes on for *S* serial seasons, which means it continues for $S(L-1)$ weeks. In order to implement LCA, the same schedule for all *S* seasons is used.

2) Determining the winner and loser

Because in an ideal league milieu the uninvited effects are avoided, one can assume a linear relationship between a team's playing strength and the result of its game. So commensurate to its playing strength, one team may have the chance to win and it is elicited from the second rule. Using the playing strength of each team, the game winner or loser is determined stochastically, considering this circumstance that the win chance for a team is proportional to its degree of fit. The degree of fit for each team is proportional to its playing strength and is measured by its distance from an ideal reference point.

Assuming teams *i* and *j* play in week *t*, with their formation of X_i^t and X_j^t and playing strengths of $f(X_i^t)$ and $f(X_j^t)$ respectively, let p_i^t describes the chance of team *i* to win against team *j* at week *t* (p_j^t is defined for team j likewise). Furthermore, let $\hat{f}$ to be an ideal value (e.g. a lower bound of the optimal value). From rule 1 it could be written:

$$\frac{f(X_i^t) - \hat{f}}{f(X_j^t) - \hat{f}} = \frac{p_i^t}{p_j^t} \tag{1}$$

Equation (1) shows that the expected chance of team *i* to win is commensurate to the difference between its ongoing playing strength and the ideal playing strength related to the ideal team. In equation (1) it is assumed that a better team can comply with more factors of an ideal team. Since teams are evaluated according to their distance from a shared ideal reference point, the distance ratio can determine the winning portion for each team. From rule 3 it can also be written:

$$p_i^t + p_j^t = 1 \tag{2}$$

Combining equations (1) and (2) will result in:

$$p_i^t = \frac{f(X_j^t) - \hat{f}}{f(X_j^t) + f(X_i^t) - 2\hat{f}} \tag{3}$$

In order to determine the winner and loser, a random number in [0,1] is chosen. If the number is less than or equal to p_i^t team *i* is the winner and team *j* loses, otherwise *j* wins and *i* loses. This procedure of determining the winner and loser comes from rules 2 and 4.

If $f(X_i^t)$ be in the vicinity of $f(X_j^t)$, then p_i^t can be closed to 0.5. Furthermore if $f(X_j^t) \gg f(X_i^t)$, then p_i^t is closed to 1. Since there is no value available for $\hat{f}$, the best function found so far is used.

$$\hat{f}^i = \min_{i=1.\dots.L} \{f(B_i^t)\} \tag{4}$$

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

3) Generating a new team formation

In LCA, the artificial analysis of a team's former performance in week t is considered as internal evaluation (strengths/weaknesses), while that of rival is treated as external evaluation (opportunities/threats). Modeling an artificial analysis for team i (in order to set up a new formation for week $t+1$), if it had won (lost) j at week t, then one can assume that this win (loss) directly pertains to its strength (weakness) or according to rule 5 it is due to weakness (strength) of team j. Fig 3 (left branch) shows this internal evaluation for team i as flowchart. Based on the league schedule at week $t+1$, team i plays with team l in its next match. If team l had won (lost) its competition with team k at week t, that success (failure) along with its formation is, maybe, a threat (opportunity) for team i. apparently, such a success (failure) is determined by means of some strengths (weaknesses). Concentrating on strengths (weaknesses) of team l, it could be found a way to avoid possible threats (to use the possible opportunities). Based on rule 5, one can focus on weaknesses (strengths) of team k instead. Fig 3 (right branch) indicates this artificial external evaluation for team i.

Based on the last week events, the possible actions for team i derives from the artificial analysis. For instance, if team i and team l had won and lost their rivalries respectively, it seems logical for team i to concentrate on its own strengths which led it to win its last match. Simultaneously, it should focus on weaknesses that brought the loss for team l in the last week. These weaknesses are perhaps opportunities for team i. The mentioned analysis is carried out for each team in week $t+1$ until a suitable formation is obtained for each team at the last week.

Based on the last week events, the possible actions for team i derives from the artificial analysis. For instance, if team i and team l had won and lost their rivalries respectively, it seems logical for team i to concentrate on its own strengths which led it to win its last match. Simultaneously, it should focus on weaknesses that brought the loss for team l in the last week. These weaknesses are perhaps opportunities for team i. The mentioned analysis is carried out for each team in week $t+1$ until a suitable formation is obtained for each team at the last week.

X_i^t, X_j^t and X_k^t are formations of team i, j and k respectively at week t. Let $X_k^t - X_i^t$ to be the gap between playing strategies of team i and team k, sensed via focusing on team k strengths. This situation is used whenever team k has won team l and it's reasonable for team i to approximate its formation to that of team k, so it may be able to beat team l, too. Also $X_i^t - X_k^t$ is defined when focusing on weaknesses of team k. $X_i^t - X_j^t$ and $X_j^t - X_i^t$ are defined in similar manner.

Considering the fact that teams normally paly based their best current formation while priming necessary changes provided by match analysis, the new formation $X_i^{t+1} = (x_{i1}^{t+1}. x_{i2}^{t+1}. \ldots . x_{in}^{t+1})$ of team i at week $t+1$ is determined through following equations.

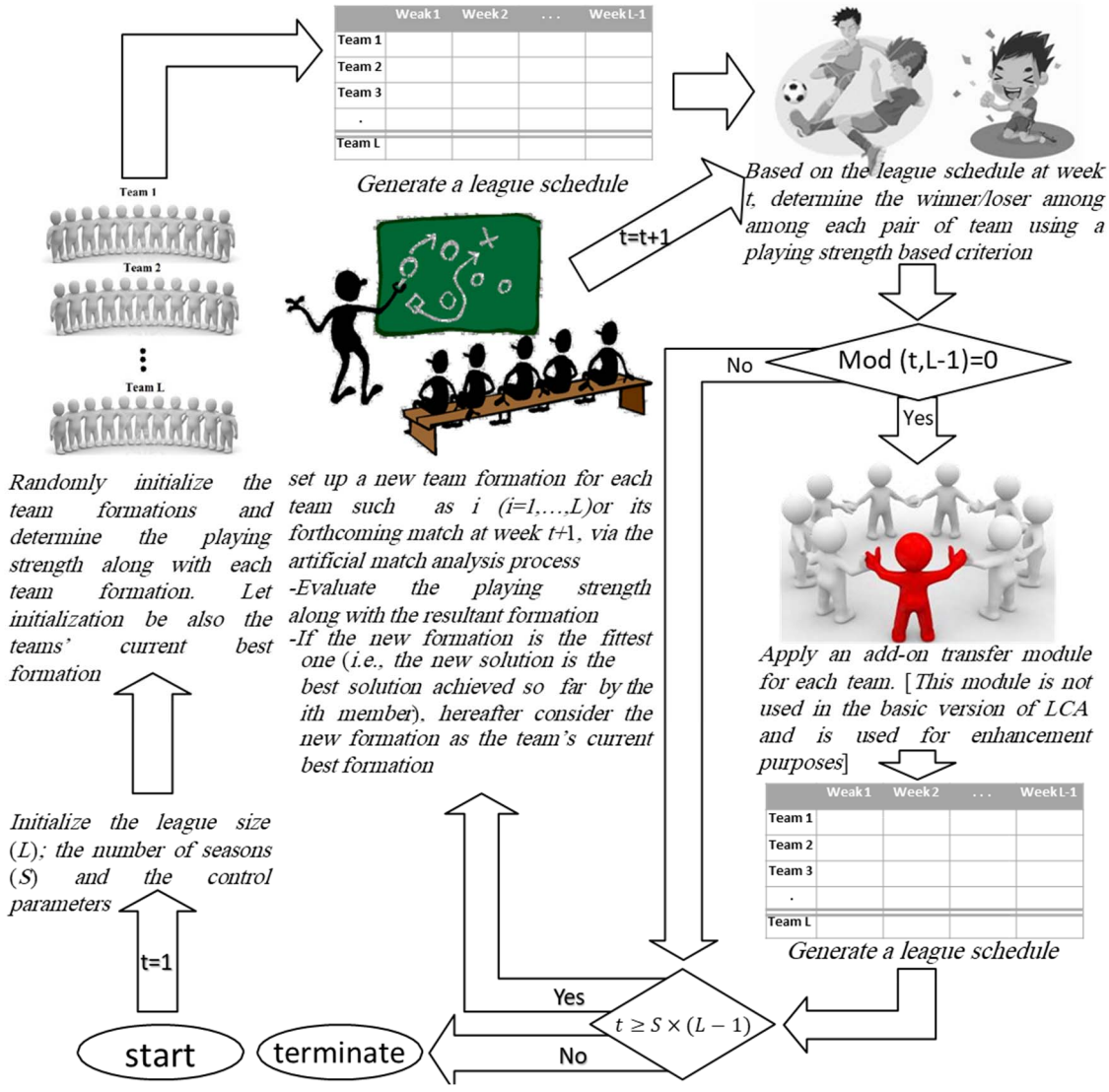


Fig. 1. Flowchart of LCA [23]

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

If both teams *i* and *j* had won:

$$x_{id}^{t+1} = b_{id}^{t} + y_{id}^{t}\left(\Psi_1 r_{1id}(x_{id}^{t} - x_{kd}^{t}) + \Psi_1 r_{2id}(x_{id}^{t} - x_{jd}^{t})\right) \quad \forall d = 1.\ldots.n \tag{5}$$

If team *i* had won but team *l* had lost:

$$x_{id}^{t+1} = b_{id}^{t} + y_{id}^{t}\left(\Psi_2 r_{1id}(x_{kd}^{t} - x_{id}^{t}) + \Psi_1 r_{2id}(x_{id}^{t} - x_{jd}^{t})\right) \quad \forall d = 1.\ldots.n \tag{6}$$

If team *i* had lost but team *j* had won:

$$x_{id}^{t+1} = b_{id}^{t} + y_{id}^{t}\left(\Psi_1 r_{1id}(x_{id}^{t} - x_{kd}^{t}) + \Psi_2 r_{2id}(x_{jd}^{t} - x_{id}^{t})\right) \quad \forall d = 1.\ldots.n \tag{7}$$

If both teams *i* and *j* had lost:

$$x_{id}^{t+1} = b_{id}^{t} + y_{id}^{t}\left(\Psi_2 r_{1id}(x_{id}^{t} - x_{kd}^{t}) + \Psi_2 r_{2id}(x_{id}^{t} - x_{jd}^{t})\right) \quad \forall d = 1.\ldots.n \tag{8}$$

In above equations, *d* is index for variables or dimensions, r_{1id} and r_{2id} are random numbers in [0,1]. Ψ_1 and Ψ_2 are coefficients used for proportion of "approach" and "retreat" components. Note that the difference sign in the parentheses causes to hasten toward the winner or retreat from loser.

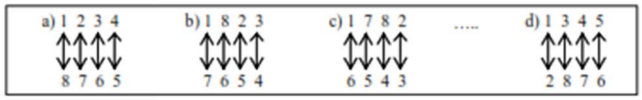


Fig. 2. Time schedule of league [23]

In equations (5) to (8), y_{id}^{t} is binary alter variable that indicates whether x_{id}^{t+1} is different from b_{id}^{t} or not. $y_{id}^{t} = 1$ means they are different and $y_{id}^{t} = 0$ says they are the same. Let $Y_i^t = (y_{i1}^{t}.y_{i2}^{t}.\ldots.y_{in}^{t})$ to be a change binary vector that the number of its ones is equal to q_i^t. Normally coaches don't change all or many facets of their team. Generally a small number of changes are recommended. Likewise, it is rational that the number of changes in B_i^t (i.e. the value of q_i^t) be infinitesimal. Using a truncated geometric probability, the number of changes is simulated. With truncated geometric distribution, the number of changes can be set more dynamically emphasizing on smaller rate of changes. Equation (9) is to determine the random number of changes in B_i^t to acquire a new formation X_i^{t+1}:

$$q_i^t = \left[\frac{\ln(1-(1-(1-p_c)^{n-q_0+1})r)}{\ln(1-p_c)}\right] + q_0 - 1 \quad : q_i^t \epsilon \{q_0.\ q_0+1\ldots.n\} \tag{9}$$

wherein, r is a random number between 0 and 1 and $p_c < 1, p_c \neq 0$ is a control parameter. q_0 is the minimum changes resulted from artificial match analysis. It is assumed that the number of changes in best formation is at least one. Generally, p_c is defined as success probability in truncated geometric distribution. The bigger amount of p_c, the smaller number of changes is advised (for more details please see [23]). After simulating the number of changes by equation (9), q_i^t is selected randomly via B_i^t elements and their value changes according to equations (5) to (8).

III. QUADROTOR DYNAMIC AND CONTROL

A. Quadrotor's equations of motion

In this section, quadrotor's equations of motion, which are discussed in corresponding former works, are provided. The dynamic model is as follows. Please see [9] for more details.

$$\begin{aligned}
\ddot{x} &= (\cos\phi\sin\theta\cos\psi + \sin\phi\sin\psi)\frac{u_1}{m}\\
\ddot{y} &= (\cos\phi\sin\theta\sin\psi - \sin\phi\cos\psi)\frac{u_1}{m}\\
\ddot{z} &= -g + (\cos\phi\cos\theta)\frac{u_1}{m}\\
\dot{p} &= qr\left(\frac{I_y - I_z}{I_x}\right) + \frac{I_R}{I_x}qg(u) + \frac{u_2}{I_x}\\
\dot{q} &= pr\left(\frac{I_z - I_x}{I_y}\right) - \frac{I_R}{I_y}pg(u) + \frac{u_3}{I_y}\\
\dot{r} &= pq\left(\frac{I_x - I_y}{I_z}\right) + \frac{1}{I_z}u_4\\
\dot{\phi} &= p + qsin\,\phi\,tan\,\theta + r\,cos\,\phi\,tan\,\theta\\
\dot{\theta} &= q\,cos\,\phi - r\,sin\,\phi\\
\dot{\psi} &= \frac{sin\,\phi}{cos\,\theta}q + \frac{cos\,\phi}{cos\,\theta}r
\end{aligned} \tag{10}$$

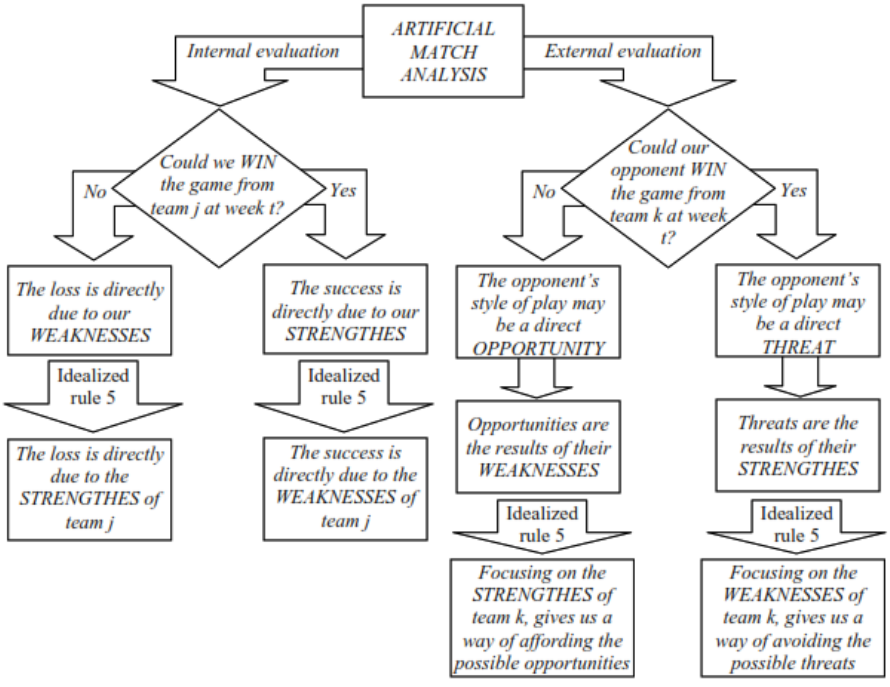


Fig. 3. Game analysis in LCA [23]

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

where $X^T = [x \quad y \quad z]$ is the position vector of the quadrotor in the inertial frame and $\Omega^T = [p \quad q \quad r]$ is angular velocity of the quadrotor in the body frame. ϕ, θ and ψ are Euler angles of the quadrotor.

B. Quadrotor trajectory control

According to three first equations of (10), knowing linear acceleration parameters ($\ddot{x}$،$\ddot{y}$،$\ddot{z}$) and yaw (ψ) angle, will give us roll (ϕ) and pith (θ) angles. Therefore, 2 control loops are used to control quadrotor; the external loop and the internal loop. The external loop controls the position of the quadrotor, while determining the desired roll and pitch angles. The internal loop, on the other hand, controls the orientation and tries to approximate the actual roll and pitch angles two desired ones. As outputs from the external control loop, the linear acceleration parameters are determined based on the position error via the following equations:

$$\begin{aligned} u_x &= \ddot{x} = Kp_x(x_d - x) + Kd_x \frac{d}{dt}(x_d - x) \\ u_y &= \ddot{y} = Kp_y(y_d - y) + Kd_y \frac{d}{dt}(y_d - y) \\ u_z &= \ddot{z} = Kp_z(z_d - z) + Kd_z \frac{d}{dt}(z_d - z) \end{aligned} \tag{11}$$

where $Kp_x, Kd_x, Kp_y, Kd_y, Kp_z, Kd_z$ are constant parameters of the PD controller of the external loop. Combining equations (10) and (11) and yaw angle feedback, roll and pitch angles are computed by these equations:

$$\begin{aligned} \theta_{des} &= atan\left(\frac{u_x \cos\psi + u_y \sin\psi}{u_z + g}\right) \\ \phi_{des} &= atan\left(\frac{u_x sin\psi - u_y cos\psi}{\frac{u_x cos\psi + u_y sin\psi}{sin\theta}}\right) \\ \psi_{des} &= 0 \end{aligned} \tag{12}$$

Now these angles are sent to the inner control loop and it will try to approach the angles to these values by changing the speeds of the motors. u_2, u_3 and u_4 control ϕ, θ and ψ respectively.

$$\begin{aligned} U_1 &= \frac{m(g - u_z)}{\cos\theta \cos\phi} \\ u_2 &= Kp_2(\phi_d - \phi) + Kd_2 \frac{d}{dt}(\phi_d - \phi) \\ u_3 &= Kp_3(\theta_d - \theta) + Kd_3 \frac{d}{dt}(\theta_d - \theta) \\ u_4 &= Kp_4(\psi_d - \psi) + Kd_4 \frac{d}{dt}(\psi_d - \psi) \end{aligned} \tag{13}$$

where $Kp_2, Kd_2, Kp_3, Kd_3, Kp_4, Kd_4$ are constant parameters of the PD controller of the internal loop. Fig. 4 shows the diagram block of the quadrotor control for a better understanding of presented relationships.

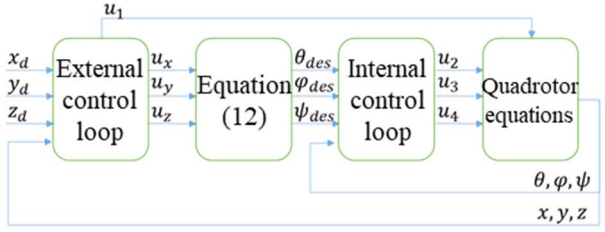


Fig. 4. Block diagram of quadrotor controller

IV. Results

In this section, the implementation of the League Championship Algorithm is discussed to find an optimal trajectory for a quadrotor. So it will make us able, through this optimal path and the algorithm, optimize a particular function.

At first, league teams for the championship are determined. In this research, each team has 11 players just like football league teams have. These 11 players are representatives of a particular path from start point to the final point, so each player in a team is a point in the path. It is supposed that quadrotor starts its motion from $\{x, y, z\}^T = \{0,0,0\}^T$ (start point) and stops at $\{x, y, z\}^T = \{5,4,4\}^T$ (final point). During the algorithm all teams have two fixed players in addition to other 11 players, one for start point $\{0,0,0\}^T$ and the other for the final point $\{5,4,4\}^T$, still other 11 players will change in the best way. It is considered 20 teams from European leagues and initialized these teams haphazardly. Initial paths for 3 teams are shown in Fig 5.

Two cases are considered in this optimal tracking problem. In the first case there is not any obstacle in the path. In contrast, the second case is defined in the presence of obstacle. In this research, the cost function consists of two different variables and it is tried to find an optimal path which optimizes these two variables. The first variable is the amount of energy used by quadrotor and the second variable is the distance traversed by quadrotor. The amount of used energy is calculated through quadrotor input values, u_1, u_2, u_3, u_4that are presented in equation (10). Therefore, minimizing the cost function, quadrotor inputs will be minimized and so the energy used by quadrotor. Besides, quadrotor traverse the path in 12 steps. In the first step, it goes to the point assigned to the first player. Then it approaches to the second point corresponding to the second player. This procedure goes on until quadrotor arrives at the final point. The distance of all steps are summed to calculate total distance. Now the cost function is acquired by summing the two cost variables.In order to validate the algorithm and evaluate its performance, it is reasonable to solve the same problem using another valid and known algorithm. Genetic algorithm is chosen to do this analogy. In other words, genetic algorithm is used to optimize the path for the same quadrotor. Like in the LCA, 11 bits population in genetic algorithm is used. As same as LCA, 20 people are used as primary population. All parameters of these two algorithms are represented in Table 1 for a complete comparison. As from Table 1 could be understood, in LCA, the cost function value, with less number of iterations compared to the genetic algorithm, is smaller than that of genetic algorithm. Drawing distinction between Fig 6 and Fig 7, it is axiomatic that the paths obtained from LCA, shown in Fig 6, are considerably better and more reliable than those of genetic algorithm, which are presented in Fig 7.

With that in mind, the second case is considered to evaluate the performance of the LCA. In this case, obstacle is present in the path. Let the obstacle be a sphere with radius of 1 meter. Let the center of sphere be (2,1.5,1.5), which is in the way of quadrotor to the final point. Fig 8 exhibits the two optimal paths, one obtained from LCA after 200 seasons, and the other obtained from genetic algorithm after 3800 iterations. As it could be understood

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

from Fig 8 and Table 2, the LCA optimizes quadrotor path in the presence of the obstacle much better than the genetic algorithm. According to the presented figures and the comparison table, it is obvious that the League Championship Algorithm did excellent in finding an optimal path for a quadrotor. Therefore, it could be used in optimizing other functions and other optimization problems.

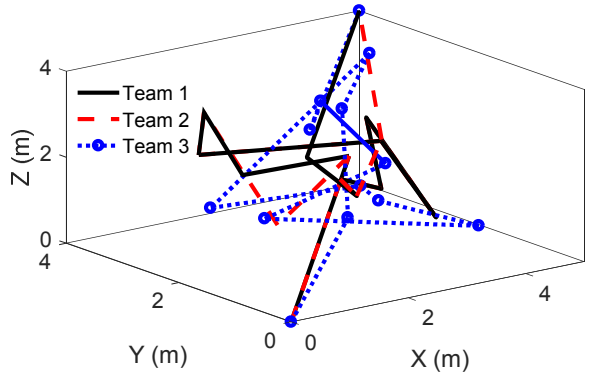


Fig. 5. Random initial paths for three different teams

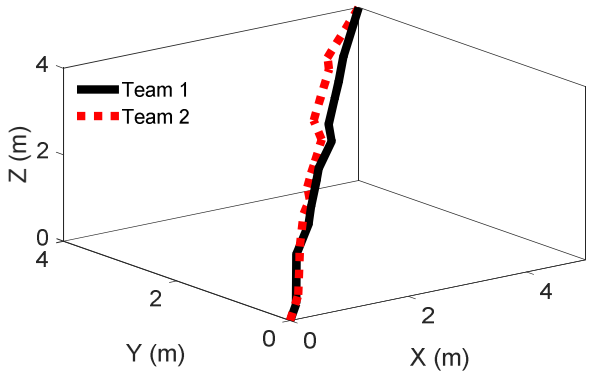


Fig. 6. Final paths for two different teams after 200 seasons

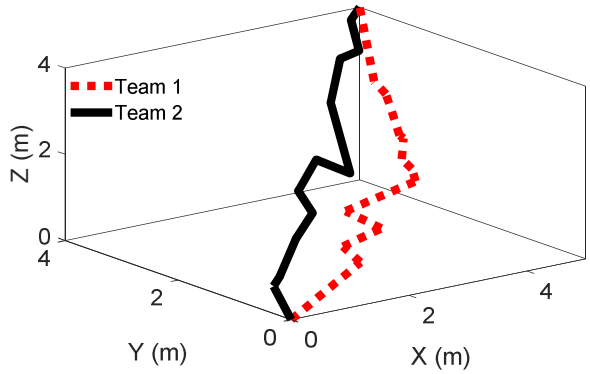


Fig. 7. Optimal paths for two different people of genetic algorithm after 4000 iteration

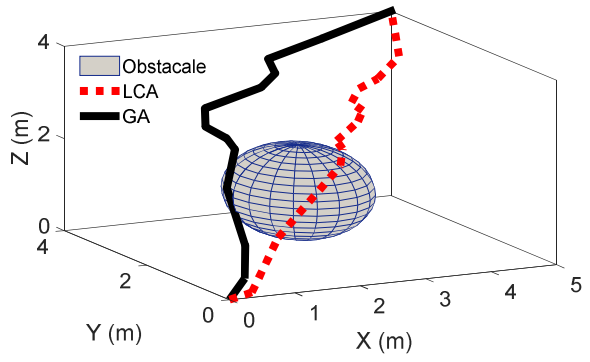


Fig. 8. Optimal paths obtained from LCA and GA with presence of obstacle

TABLE I. COMPARISON BETWEEN GA AND LCA FOR OPTIMAL PATH

	Genetic algorithm	***LCA***
Initial population	20	20
Number of bits for each population	11	11
Number of iteration	4000	3800
Quadrature cost function in the initial path	1.55×10^7	1.82×10^7
Quadrature cost function after iterations	4.58×10^6	1.61×10^6

TABLE II. COMPARISON BETWEEN GA AND LCA FOR CHOOSING OPTIMAL PATH WITH EXISTENCE OF OBSTACLE IN THE PATH

	Genetic algorithm	***LCA***
Initial population	20	20
Number of bits for each population	11	11
Number of iteration	4000	3800
Quadrature cost function in the initial path	1.81×10^7	1.75×10^7
Quadrature cost function after iterations	3.94×10^6	1.64×10^6

V. CONCLUSION

In this paper, one of the most important issues in UAVs field contemplated; trajectory optimization in order to reduce the fuel consumption as well as the distance traversed by the vehicle. Combining the League Championship Algorithm, which is emerged by an Iranian scholar, with optimal path tracking problem of UAVs, it concluded that its performance is extraordinary, when drawing distinction between that and genetic algorithm, one of the prevailing algorithms in optimization problems. Better performance of the LCA than the genetic algorithm showed in Tables 1 and 2, and Figs 7, 8 and 9. When the obstacle was not present, genetic algorithm caused the cost function value to be nearly 9 percent of its initial value, while this percentage for LCA was about 30 percent. In the presence of the obstacle, genetic algorithm diminished the cost function value to 22 percent of its initial value, but using LCA, this value reduced to about 10 percent of its initial value.

REFERENCES

[1] D. J. Cook, Adding Intelligence To Robot arm Path Planning Using A Graph-match Analogical Reasoning System., *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, 1992, pp. 657-662.

[2] B. Y. N. Michael, D. Mellinger, Q. Lindsey, V. Kumar, The GRASP Multiple Micro-UAV Test Bed, *IEEE Robtics & Automation Magazine*, vol. 17, no. 3, pp. 56-65, 2010.

[3] M. Doakhan, M. Kabganian, R. nadafi, A. K. Eigoli, Trajectory Tracking of a Quadrotor for Obstacle Avoidance Using Super-Twisting Sliding Mode Controller and Observer. *Modares Mechanical Engineering*, Vol. 17, no.8, p.p 333-342, 2017(in Persian).

[4] E. Abbasi, M. Ghayour and M. Danesh, "Virtual Leader-Follower Formation Control of Multi Quadrotors by using Feedback Linearization Controller," *2017 5th RSI International Conference on Robotics and Mechatronics (ICRoM)*, Tehran, 2017, pp. 614-619.

[5] L. Doitsidis, K. P. Valavanis, A Framework for Fuzzy Logic Based UAV Navigation and Control, *IEEE International Conference on Robotics and Automation*, pp. 4041-4046, April, 2004.

[6] A. Mottahedi, A. Akbarzadeh Kalat. Adaptive robust sliding mode control of quadrotor in the presence of wind disturbance. *Modares Mechanical Engineering*, vol. 16, no. 12, p.p 95-102, 2016 (in persian).

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.

[7] S. A. Bortoff, Path planning for UAVs *American Control Conference, 2000. Proceedings of the 2000*, Chicago, IL, USA, USA, 2000.

[8] N. Regina, M. Zanzi, 2D Tracking and Over-ight of a Target by Means of a Non-linear Guidance Law for UAV, *IEEE Aerospace conference*, pp. 1-11, Mar. 2009.

[9] E. Abbasi, Quadrotor UAV Guidence for Ground Moving Target Tracking, *Journal of Advances in Computer Engineering and Technology*, vol. 2, no. 1, pp. 37-44, 2016.

[10] Y. Morel, A. Leonessa, Direct Adaptive Tracking Control of Quadrotor Aerial Vehicles, *Dynamic Systems and Control*, Parts A and B, vol. 2006, pp. 155-161, 2006.

[11] S. Park, J. Deyst, J. P. How, Performance and Lyapunov Stability of a Nonlinear Path Following Guidance Method, *Journal of Guidance, Control, and Dynamics*, Vol. 30, No. 6, pp. 1718-1728, 2007.

[12] A. Tuncer, M. Yildirim, Dynamic path planning of mobile robots with improved genetic algorithm, *In Computers & Electrical Engineering*, Volume 38, Issue 6, Pages 1564-1572, 2012.

[13] O. Castillo, L. Trujillo, P. Melin, Multiple Objective Genetic Algorithms for Path-planning Optimization in Autonomous Mobile Robots, *Soft Computing*, Volume 11, Issue 3, pp 269–279, 2007.

[14] I. AL-Taharwa, A. Sheta, M. Al-Weshah, A Mobile Robot Path Planning Using Genetic Algorithm in Static Environment, *Journal of Computer Science*, vol. 4, no. 4, pp. 341-344, 2008.

[15] A. Khan, N. Iram, H. Zulfiqur, On Complete Coverage Path Planning Algorithms for Non-holonomic Mobile Robots: Survey and Challenges.,. *Journal of Information Science & Engineering,* Vol. 33 Issue 1, p.p 101-121, 2017.

[16] J. C. Latombe, Robot Motion Planning. *1 st ed, Boston, UK, MA: Kluwer Academic Publishers (Published by Springer),* 1991.

[17] T. Lozano-Perez and M. Wesely, An algorithm for planning collision-Free paths among polyhedral obstacles. *Commun. of the ACM [Online]*, 22(10), pp. 560-570, 1979.

[18] S. Russell, P. Norving, Artificial Intrlligence: A Modern Approach, *Prentice Hall*, pp. 97-104, 2003.

[19] A. Stentz, Optimal and efficient path planning for partially-known environments, *Proceedings of the 1994 IEEE International Conference on Robotics and Automation*, San Diego, CA, 1994, pp. 3310-3317 vol.4.

[20] J. Tu, S. X. Yang, Genetic algorithm based path planning for a mobile robot, *2003 IEEE International Conference on Robotics and Automation (Cat. No.03CH37422)*, 2003, pp. 1221-1226 vol.1.

[21] J. Fan, Optimal Path Planning and Control of Quadrotor Unmanned Aerial Vehicle for Area Coverage, *The University of Toledo Thesis*, 2014.

[22] C. Gebhardt, B. Hepp, T. N¨ageli, S. Stev˘si´c, O. Hilliges, Airways: Optimization-Based Planning of Quadrotor Trajectories according to High-Level User Goals, *CHI Conference*, 2016.

[23] A. H. Kashan. An Efficient Algorithm for Constrained Global Optimization and Application to Mechanical Engineering Design: League Championship Algorithm (LCA). *Computer-Aided Design*, vol. 43, pp. 1769-1792, 2011.

[24] E. Abbasi, M. Mahjoob, R. Yazdanpanah. Controlling of Quadrotor UAV using a Fuzzy System for Tuning the PID Gains in Hovering Mode. *International Conference on Advances in Control Engineering* (ACE 2013), Istanbul, Turkey, 2013.

Authorized licensed use limited to: Northeastern University. Downloaded on October 08,2026 at 18:16:17 UTC from IEEE Xplore. Restrictions apply.