Cumhuriyet Üniversitesi Fen Fakültesi
Fen Bilimleri Dergisi (CFD), Cilt:36, No: 3 Özel Sayı (2015)
ISSN: 1300-1949

Cumhuriyet University Faculty of Science
Science Journal (CSJ), Vol. 36, No: 3 Special Issue (2015)
ISSN: 1300-1949

A New Obstacle Avoidance Algorithm for Quad rotors Group in the presence of Dynamic and Static Obstacles

Mohammad REZAEİ[1,*], Hossein BOLANDİ[1], Fateme JAMALDOOST[1],

SeyedMajid SMAİLZADEH[1]

[1]*Iran University of Science and Technology, Narmak, Tehran, Iran*

Received: 01.02.2015; Accepted: 05.05.2015

Abstract. In this paper a new geometric-based collision avoidance scheme is presented for a group of quad rotor in environment with static and dynamic obstacles. The objective is to consider the full dynamics of the system to design obstacle avoidance controllers for the group of quad rotors. We introduce a method for both moving and non-moving obstacles. The proposed algorithm guide group toward the target in the path which is tangent to obstacle defined safe circle and optimize trajectory for minimized path to target. Due to simulation results, it is independent of the number of agents and applicable for static and dynamic obstacles. Simulation results are presented to validate the designed algorithm.

Keywords: Obstacle Avoidance, flying robot, Geometric Approach, dynamic obstacle, Path planning

1. INTRODUCTION

In the recent years, using of robots, specially flying robot has been interested for many researchers. Quad rotor is a type of flying robot which is very applicable due to simplicity, small size and the ability of Vertical Take-off and landing (VTOL) [1]. Since we need the outdoor application, it is very interesting for researchers from obstacle avoidance point of view.

The ability to detect and avoid obstacles in real time is very important for any practical application of flying robot. Therefore, a number of solutions have been proposed for this problem. Most of these solutions demand a heavy computational load, which are difficult to implement. In this paper we introduce a novel method of obstacle avoidance for a flying robot or group of flying robots relying on low cost on-board sensors, and involving a reasonable level of calculations, so that it can be used in real time control applications.

Overview

In this section we discus about some existing obstacle avoidance in brief.

THE BUG ALGORITHMS

The first and simplest obstacle avoidance algorithm is called "bug algorithm" [1]. In the bug algorithm, when an obstacle is encountered, the robot fully circles the object to find the point

* Corresponding author. *Email address*: *rezaei_mde@yahoo.com*

Special Issue: The Second National Conference on Applied Research in Science and Technology

with the shortest distance to the goal, and then leaves the boundary of the obstacle from this point [2, 3]. In the modified bug algorithm, called 'bug2" , the robot leaves obstacle as soon as it intersects the line segment that connects the start point and the goal.

The simplicity is major advantage of bug algorithms but have some disadvantages such as:

- Do not consider the actual kinematics of the robot.
- Sensor noise seriously affects the overall performance of the robot because bugs algorithms only consider the most recent sensor readings.
- Algorithm speed is low.

THE POTENTIAL FIELD ALGORITHM

This algorithm assumes that the robot is in the potential fields that attract it towards the goal, or reject it away from the obstacles. The actual path is determined by the resultant of these forces [4, 5]. This algorithm still doesn't solve the drawbacks of the bug algorithms, and is more difficult to use in real time applications.

THE VECTOR FIELD HISTOGRAM ALGORITHM

The Vector Field Histogram (VFH) creates a polar histogram of several recent sensor readings to compute the probability of real obstacle in direction of sensor readings. The polar histogram is used to identify all the passages large enough to allow the robot to pass through. The particular path is select by the evaluation of a cost function, defined for each passage and the passage with the minimum cost is selected. The algorithm has better robustness to sensor noise but still has a considerable computation load [6-8].

THE BUBBLE BAND ALGORITHM

The algorithm defines a "bubble" containing the maximum available free space around the robot, which can be travelled in any direction without collision [9]. The size of the bubble is determined by model of the robot and the range between robot and obstacle provided by the sensors.

2. NEW OBSTACLE AVOIDANCE ALGORİTHM

One of the most important problems in flying robots is the obstacle avoidance issue due to existence of obstacles especially in outdoor environment. To do this, robot must detect the obstacles and avoid from contact the obstacles by applying control commands. In this case we have three goals:

- The short path to target.
- Avoiding the obstacles without contact.
- Speed and maneuver limitations.

STATIC OBSTACLE [10, 11]

Suppose a group of quad rotors are passing towards predefined trajectory (for example in triangular formation). Leader is located at point A with coordination $(x_L, y_L)$and the followers are placed at the vertexes of a triangle with d_{jo}, d_{ko}, l_{ko} sides. Each quad rotor has a safe circle with radius r_s. Note, the algorithm must control the path so neither obstacle nor other robot does not enter this safe circle. In addition we define a larger circle with radius r_t. Algorithm should always survey the region inside this circle and make proper decisions to avoid contact to obstacles existing in this region. Suppose that leader has found an obstacle at point B at coordination (x_o, y_o) and radius r_o.

Indeed this obstacle is located on the safe circle border of the group. There will be two options: group may pass the obstacle from right or left side to avoid contact. Leader shifts its path toward one of points C1 or C2. So we should calculate C1 & C2 with coordination(x_{c1}, y_{c1}), (x_{c2}, y_{c2})so those followers do not contact the obstacle. To do this, we choose points D1 & D2 so that satisfy following conditions: if group is placed on the corners of triangle and triangle is located in the right direction, the safe circle which is placed in obstacle side, be tangent with obstacle while leader is placed on one of points C1 or C2. With these conditions, we should first calculate coordination of points D1 & D2. Then, being specified the length of triangle sides, coordination of C1 and C2 points can be also obtained.

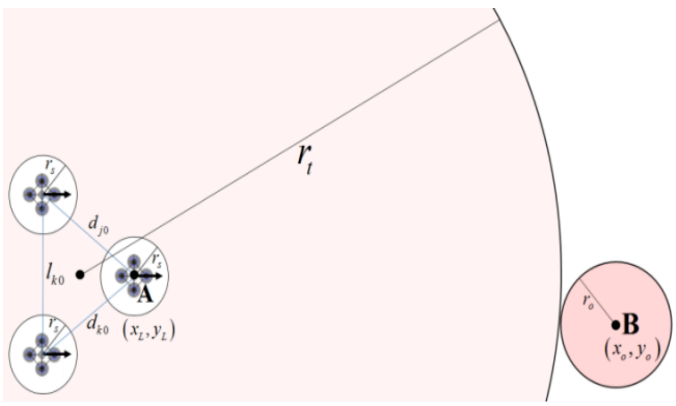

Figure 1. Obstacle detection.

To calculate coordination of D1 &D2 (x_{d2}, y_{d2}), we reconfigure above figure as follows. In above figure, r is equal to the (x_{d1}, y_{d1}) distance between points B and D1 or D2 and is obtained from the following equation:

$$r = r_o + r_s + \frac{l_{k0}}{2} \tag{1}$$

Applying the relations of the above system, coordination of D1 & D2 can be extracted. We know that the line connected points A & D are expressed in coordination system of (x, y) as followers:

$$y = \frac{y_d - y_L}{x_d - x_L}(x - x_L) + y_L \tag{2}$$

Derivation of above equation yields for line slope (m_{line}):

$$m_{line} = \frac{y_d - y_L}{x_d - x_L} \tag{3}$$

On the other hand, we know that the equation of circle in figure 7 is expressed as follows:

$$y = \pm\sqrt{r^2 - (x - x_o)^2} + y_o \tag{4}$$

Derivation of above equation yields for slope (m_{line}) as follows:

$$m_{circle} = \pm\frac{-2(x - x_o)}{\sqrt{r^2 - (x - x_o)^2}} \tag{5}$$

To satisfy problem conditions, circle slope at point (x_d, y_d) must be equal to line. So we can write as follows:

$$\pm\frac{-2(x_d - x_o)}{\sqrt{r^2 - (x_d - x_o)^2}} = \frac{y_d - y_L}{x_d - x_L} \tag{6}$$

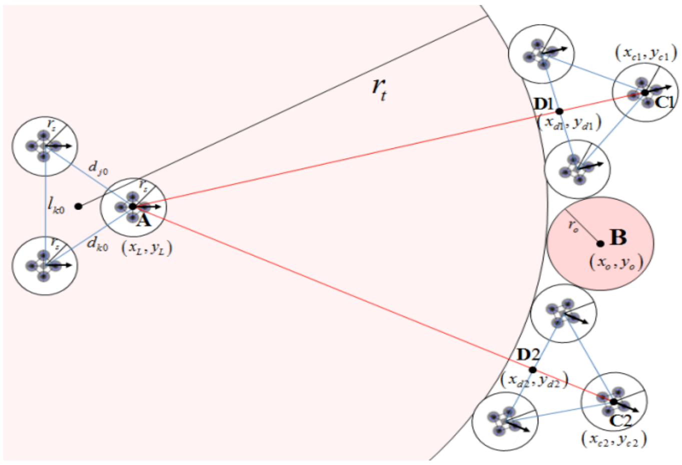

Figure 2. Safe circles.

On the other hand, the point (x_d, y_d) is placed on circle border, so we have:

$$y_d = \pm\sqrt{r^2 - (x_d - x_o)^2} + y_o \tag{7}$$

So we have:

$$\begin{cases} \pm\dfrac{-2(x_d - x_o)}{\sqrt{r^2 - (x_d - x_o)^2}} = \dfrac{y_d - y_L}{x_d - x_L} \\ y_d = \pm\sqrt{r^2 - (x_d - x_o)^2} + y_o \end{cases} \tag{8}$$

D1 & D2 are obtained by solving the (8). Rewriting the equation, we present error function for estimation of proper point as follows:

$$error = \pm\frac{-2(x_d - x_o)}{\sqrt{r^2 - (x_d - x_o)^2}} - \frac{y_d - y_L}{x_d - x_L} \tag{9}$$

$$(x_o - r) \le x_d \le (x_o + r), y_d = \pm\sqrt{r^2 - (x_d - x_o)^2} + y_o$$

If group are locate at the above or under the circle, when we consider y as a dependant variable, we should pay attention derivate at the circle tangent point, inclines to infinity. So, the upper equation is valid if groups are locating inside the grey region in following figure.

To identify grey region we use equation of l1 and l2:

$$\begin{cases} l_1 : y = -x + x_o + y_o \\ l_2 : y = x - x_o + y_o \end{cases} \tag{10}$$

Regarding the equation, if leader is placed in one of following regions, we use equation (9):

$$\begin{cases} x_L \le x_o \quad , \quad x - x_o + y_o \le y_L \le -x + x_o + y_o \\ x_L > x_o \quad , \quad x - x_o + y_o \ge y_L \ge -x + x_o + y_o \end{cases} \tag{11}$$

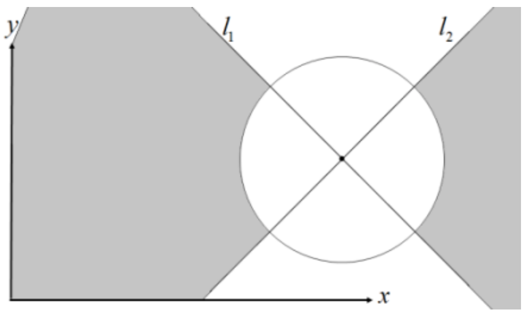

Figure 3. Geometric representation of obstacle avoidance.

If leader is not placed in mentioned region, we use following equation to find tangent point. In this equation x and y are exchanged:

$$error = \pm\frac{-2(y_d - y_o)}{\sqrt{r^2-(y_d-y_o)^2}} - \frac{x_d - x_L}{y_d - y_L} \tag{12}$$

$$(y_o - r) \le y_d \le (y_o + r), x_d = \pm\sqrt{r^2-(y_d-y_o)^2} + x_o$$

After calculation of D1 & D2, we can calculate C1 & C2 from following equations:

$$\begin{cases} y_{c1} = y_d + \sqrt{d_{k0}^2 - (0.5l_{k0})^2}\,\dfrac{(y_d - y_L)}{\sqrt{(y_d-y_L)^2+(x_d-x_L)^2}} \\ y_{c2} = y_d + \sqrt{d_{j0}^2 - (0.5l_{k0})^2}\,\dfrac{(y_d - y_L)}{\sqrt{(y_d-y_L)^2+(x_d-x_L)^2}} \end{cases} \tag{13}$$

Now with coordination of C1 & C2, leader can determine proper path to avoid contact to obstacle and can simply choose the point which concludes to shorter path by calculating distance of C1 & C2 relative to the predefined trajectory. A 3D view of simulated environment is illustrated in following figure. As can be seen, obstacles are distributed as spheres at different heights and positions.

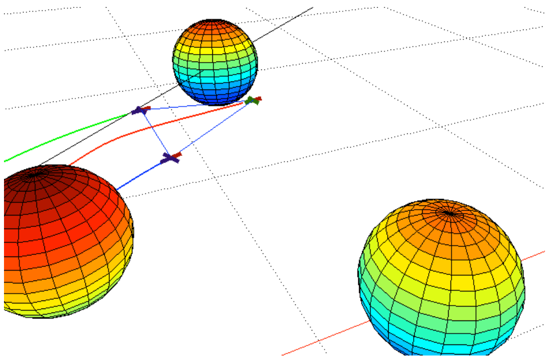

Figure 4. Simulation in 3d space and static obstacles.

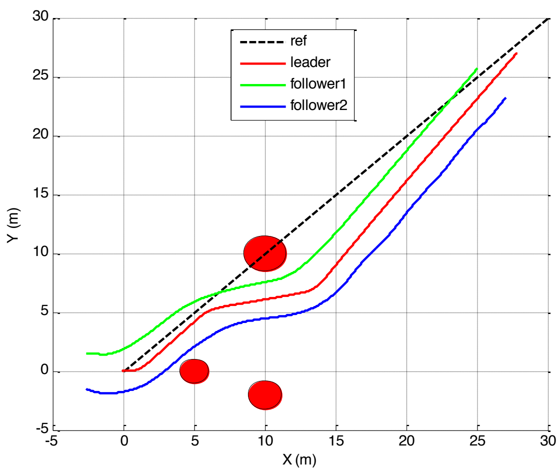

Figure 5. Simulation in 2d space and static obstacles.

The travelled path is illustrated in figures 6 and 7.

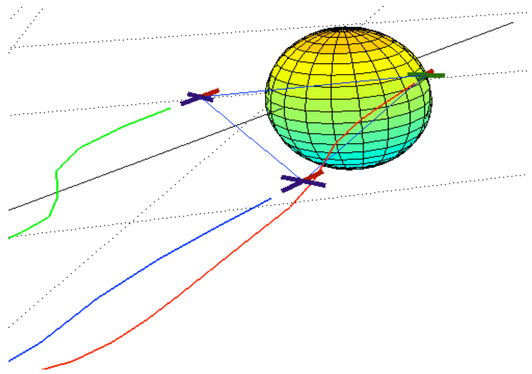

Figure 6. Change in obstacle surface (3d).

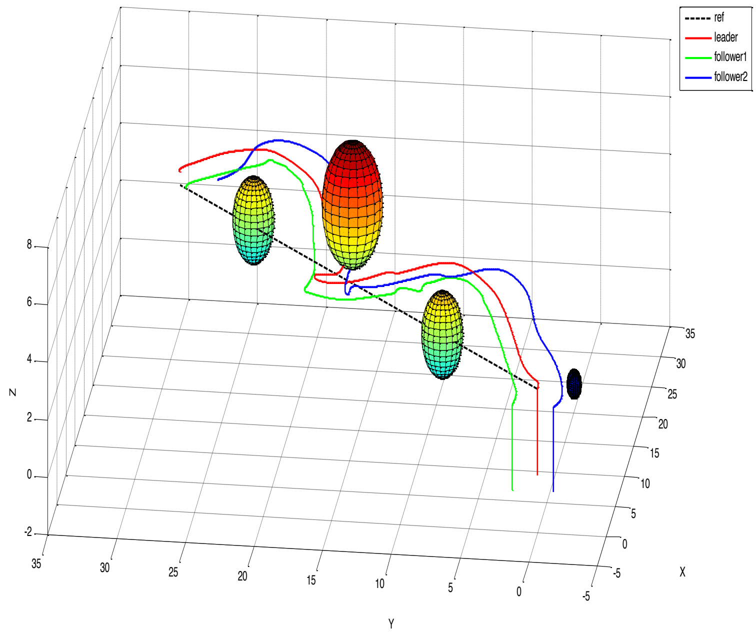

Figure 7. 3d path.

The travelled path in 3D space for a group of six agents and static obstacles is illustrated in figures 8, 9, 10 and 11.

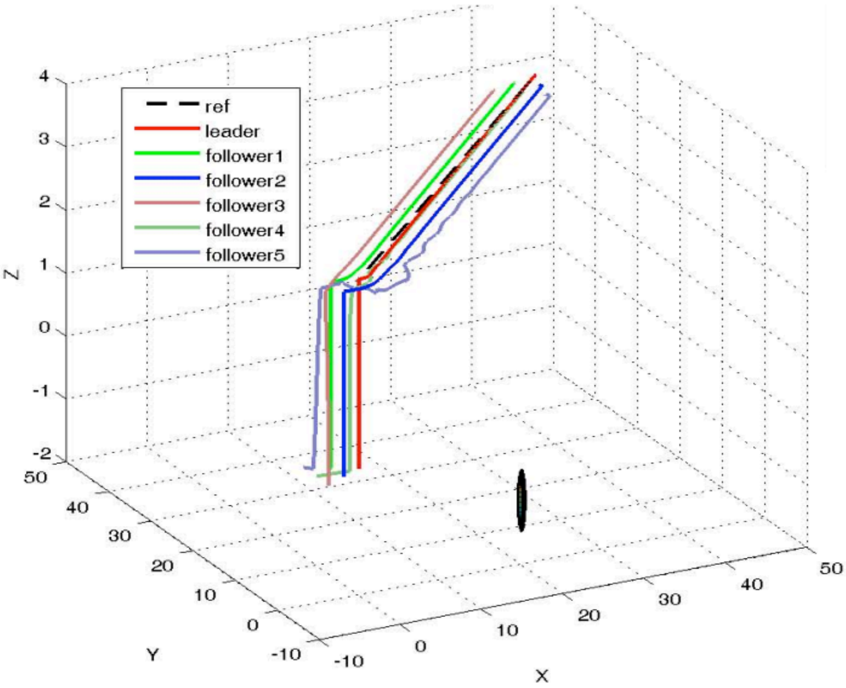

Figure 8. 3d path of 6 quadrotors group.

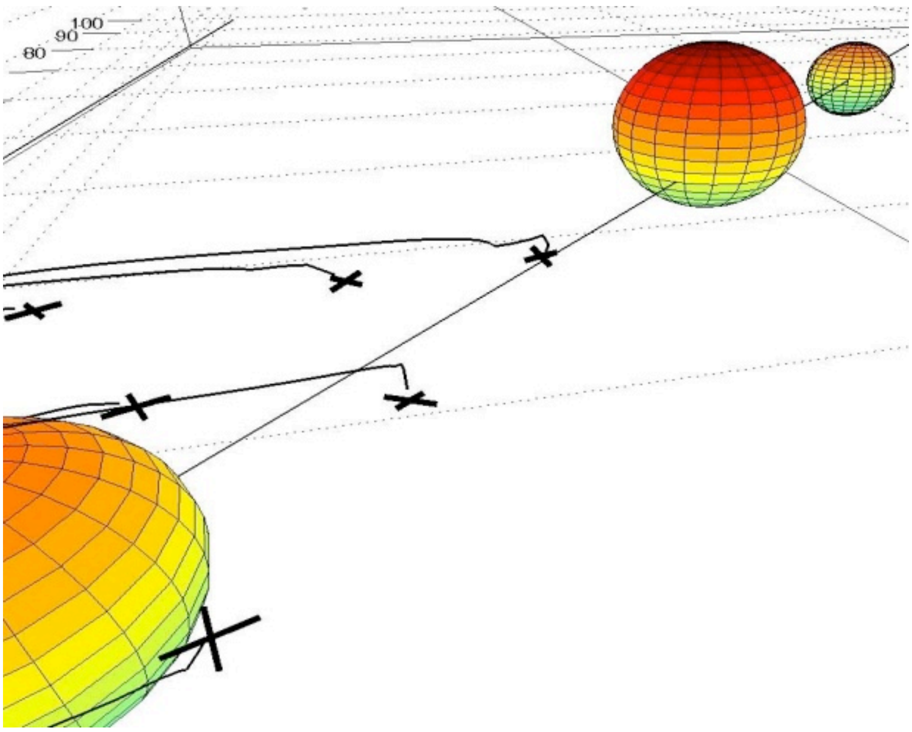

Figure 9. Groupof six quadrotorand 3 static obstacles(first obstacle).

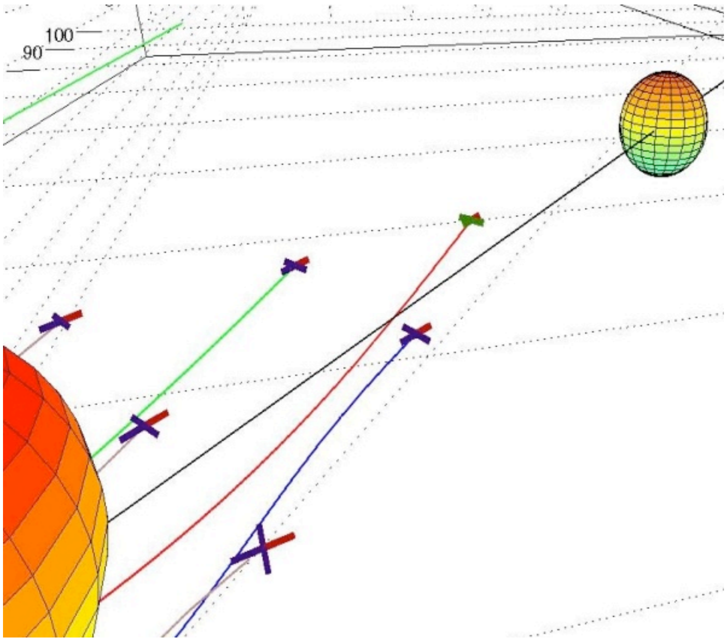

Figure 10. Groupof six quadrotor(2nd obstacle).

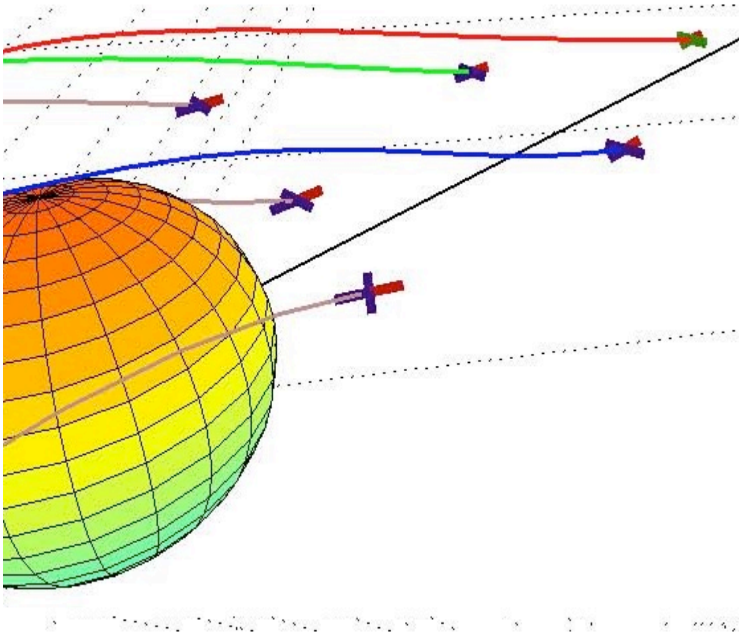

Figure 11. Groupof six quadrotor (3rd obstacle).

DYNAMIC OBSTACLE

Most of obstacle avoidance algorithms are applicable for static obstacles. We present new method for both static and dynamic obstacles. Suppose a vector from quad rotors safe circle tangent to safe circle of dynamic obstacle and calculate distances and orientations of quad rotors and obstacles, in every step time (Instantaneous Vector). The initial position of quad rotor(x0,y0,z0,rq), target point(xt,yt,zt) and obstacle position(xobs,yobs,zobs, robs) are illustrated in figure 12.

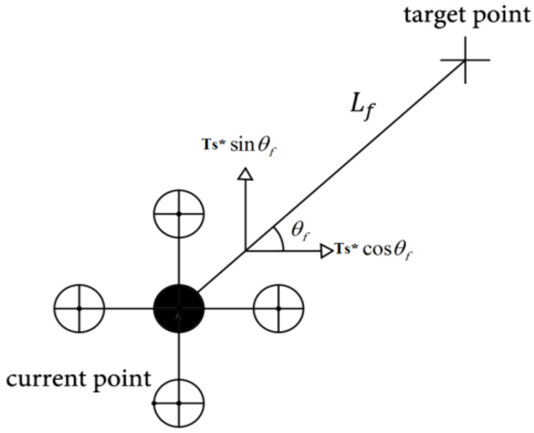

Figure 12. Positions of quadrotor, obstacles and target without obstacle.

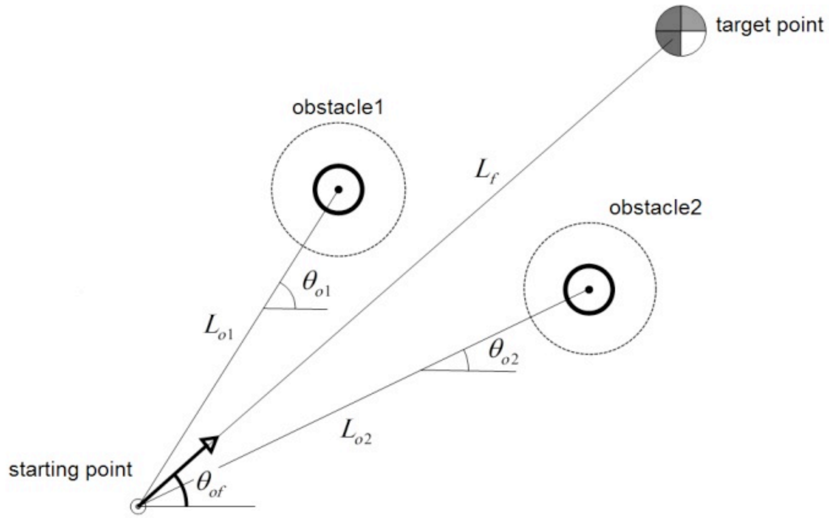

Figure 13. Positions of quadrotor, obstacles and target point with obstacles.

If there are not any obstacles we have:

$$L_f = \sqrt{(x - x_t)^2 + (y - y_t)^2 + (z - z_t)^2} \tag{14}$$

$$\theta_f = \tan^{-1}(\frac{y - y_t}{x - x_t}) \tag{15}$$

$$\gamma_f = \tan^{-1}(\frac{z - z_t}{\sqrt{(x - x_t)^2 + (y - y_t)^2}}) \tag{16}$$

$$x_S = V_r * T_s * \cos(\theta_f)\cos(\gamma_f) \tag{17}$$

$$y_S = V_r * T_s * \sin(\theta_f)\cos(\gamma_f) \tag{18}$$

$$z_S = V_r * T_s * \sin(\gamma_f) \tag{19}$$

V_r is robot speed and T_s is step size of algorithm. By updating the position of robot we have:

$$x_{new} = x_0 + x_S \tag{20}$$

$$y_{new} = y_0 + y_S \tag{21}$$

$$z_{new} = z_0 + z_S \tag{22}$$

If there are static and dynamic obstacles we have:

$$L_{obs} = \sqrt{(x - x_{obs})^2 + (y - y_{obs})^2 + (z - z_{obs})^2} \tag{23}$$

$$\theta_{obs} = \tan^{-1}\left(\frac{y - y_{obs}}{x - x_{obs}}\right) \tag{24}$$

$$\gamma_{obs} = \tan^{-1}\left(\frac{z - z_{obs}}{\sqrt{(x - x_{obs})^2 + (y - y_{obs})^2}}\right) \tag{25}$$

Lobs is distance from robot to obstacle, θ obs is angle of Lobs vector with x axis and γ_{obs} is angle of Lobs vector with z axis as shown in figure 14. Figure 15 shows robot maneuver for obstacle avoidance.

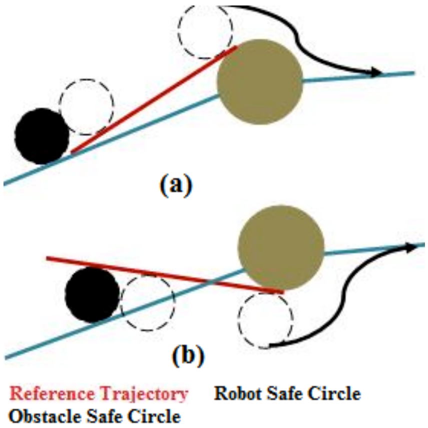

Figure 14. Robot maneuver.

Figure 23 shows tangent vector and its angle in mobile path:

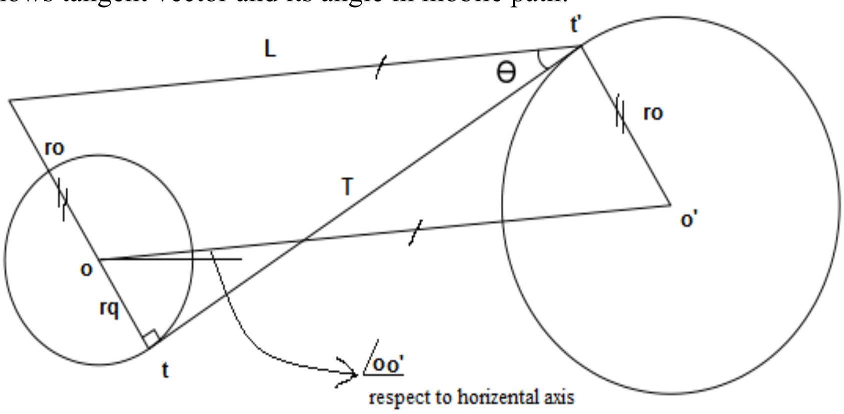

Figure 15. Geometry of robot and obstacle.

T is tangent line between robot and obstacle safe circle. θ_T is angle of tangent line respect to oo' line. We consider distance limit to oo' line and define θ_T .

$$T = \sqrt{|oo'|^2 - (r_{obs} + r_q)^2} \tag{26}$$

$$\theta_T = \tan^{-1}(\frac{r_{obs} + r_q}{T}) \tag{27}$$

So we have below equations((a) or (b) selection is based to obstacle position:

$$\begin{aligned}
&f\ \angle oo' < 0 \\
&\quad \theta_T' = \theta_T + \angle oo' \quad (a) \\
&elseif\ \angle oo' >= 0 \\
&\quad \theta_T' = -\theta_T + \angle oo' \quad (b) \\
&end
\end{aligned} \tag{28}$$

We use equations (26), (27) and (28) for 2d state. In 3d state we have:

$$\gamma_T = \tan^{-1}(\frac{r_{obs} + r_q}{T}) \tag{29}$$

In 3d space there are four tangent lines which must be select based on obstacle and robot positions.

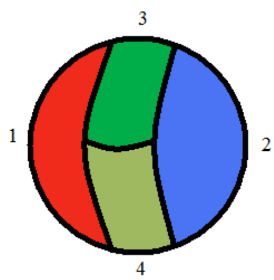

Figure 16. Possible regions.

region 1:

$$\begin{aligned}
&T = \sqrt{|oo'|^2 - (r_{obs} + r_q)^2} \\
&\theta_T' = -\tan^{-1}(\frac{r_{obs} + r_q}{T}) + \angle oo' = -\tan^{-1}(\frac{r_{obs} + r_q}{T}) + \theta_{obs} \\
&\gamma_T' = \gamma_{obs}
\end{aligned} \tag{30}$$

region 2:

$$\begin{aligned}
&T = \sqrt{|oo'|^2 - (r_{obs} + r_q)^2} \\
&\theta_T' = \tan^{-1}(\frac{r_{obs} + r_q}{T}) + \angle oo' = \tan^{-1}(\frac{r_{obs} + r_q}{T}) + \theta_{obs} \\
&\gamma_T' = \gamma_{obs}
\end{aligned} \tag{31}$$

region3:

$$T = \sqrt{|oo'|^2 - (r_{obs} + r_q)^2} \qquad (32)$$
$$\theta_T = \theta_{obs}$$
$$\gamma_T' = \gamma_T + \gamma_{obs}$$

region 4:

$$T = \sqrt{|oo'|^2 - (r_{obs} + r_q)^2} \qquad (33)$$
$$\theta_T = \theta_{obs}$$
$$\gamma_T' = \gamma_T + \gamma_{obs}$$

The trajectory in 3D space with3static and one dynamic obstacle are illustrated in figures 17, 18, 19 and 20.

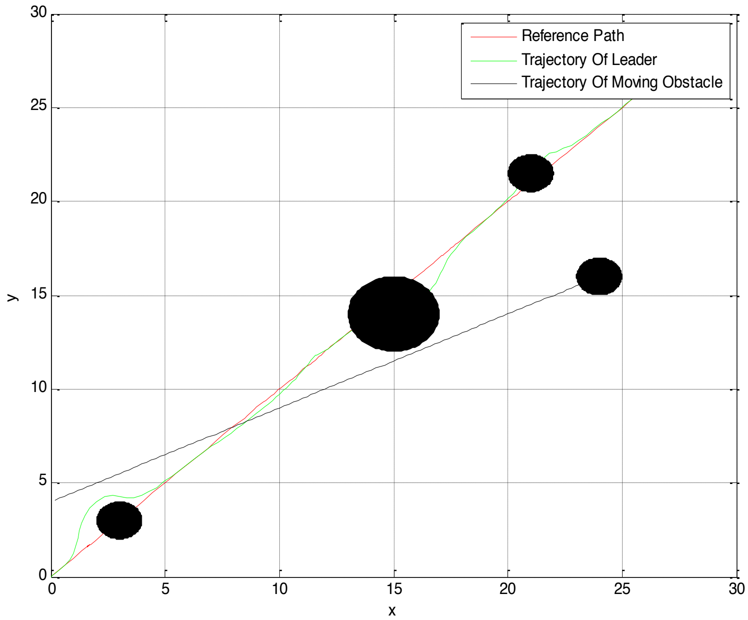

Figure 17. Trajectory in x-y plane(black line is the moving obstacle path)

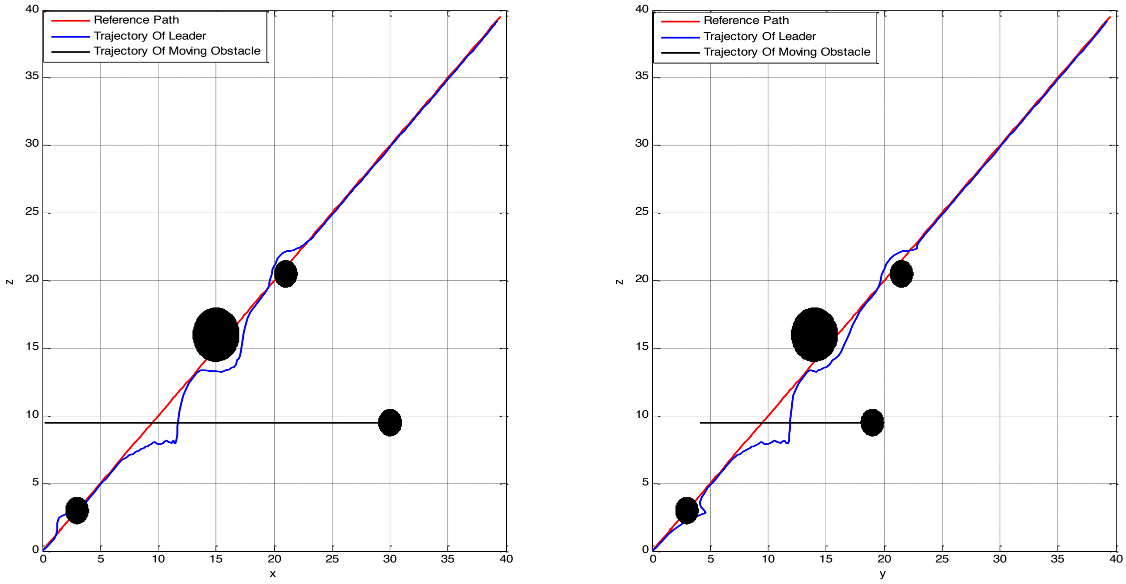

Figure 18. Trajectory in x-z and y-z planes(black line is the moving obstacle path).

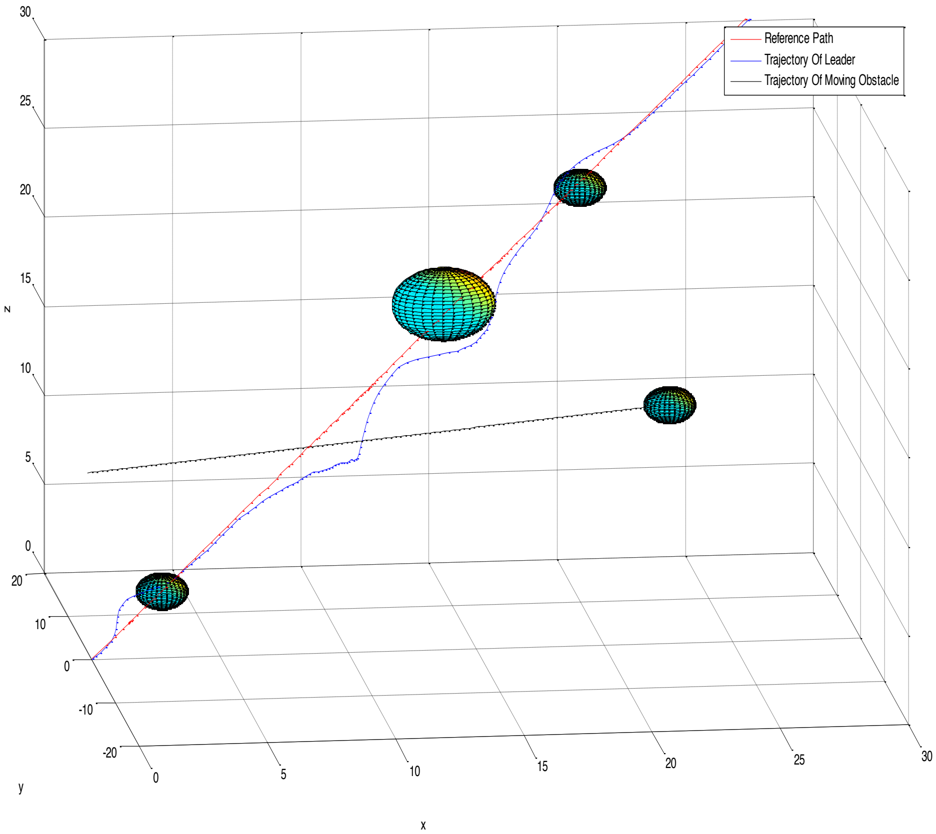

Figure 19. 3d trajectory and 3 static and 1 dynamic obstacle avoidance (black line is the moving obstacle path).

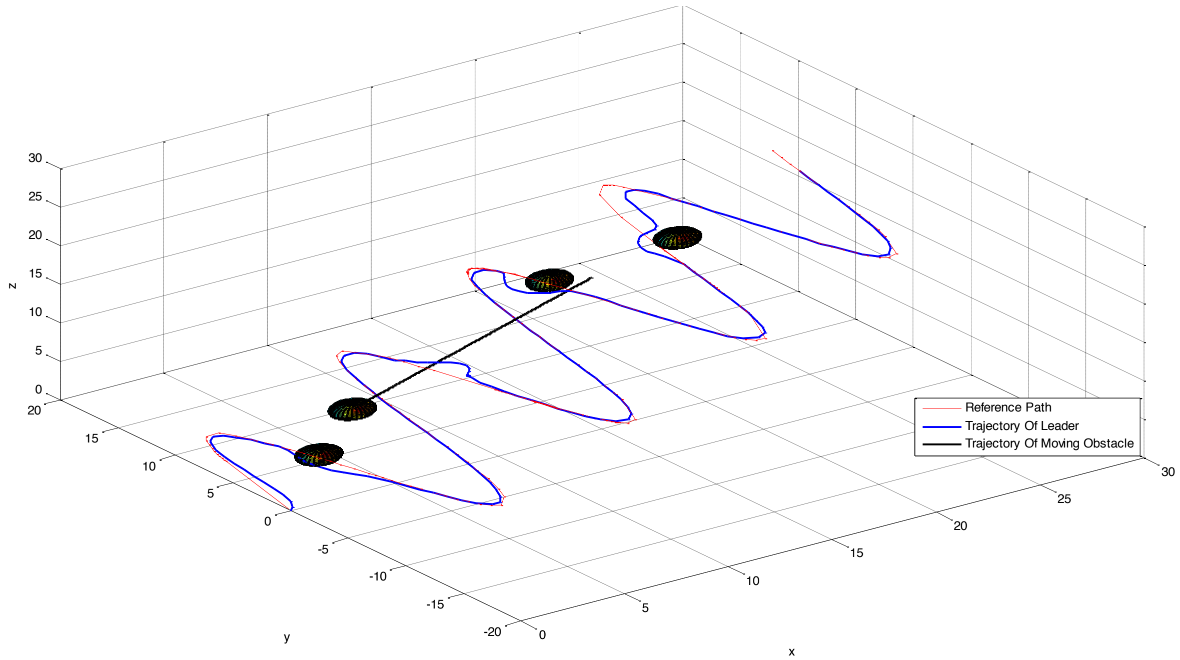

Figure 20. 3d trajectory and 3 static and 1 dynamic obstacle avoidance in sinusoidal path (black line is the moving obstacle path).

3. CONCLUSION

In this paper we introduce a new obstacle avoidance algorithm which is use for dynamic and static obstacles. As the results show that the presented algorithm can control the group in designed trajectory for avoiding obstacles and keeping formation. The algorithm uses the sensor information which is measurable by onboard sensors. By applying the algorithm momentarily, there are no differences between static or dynamic obstacles. This is the main contribution of algorithm. The algorithm is independent to the number of agent and it is stable in whole path to target. The introduced method of relying on low calculations, so that it can be easily used in real time control applications with microcontroller based quadrotors. Simulation results show the algorithm performance and ability.

REFERENCES

[1] Lumelsky, V.and Skewis, T.(1990),"Incorporating Range Sensing in the Robot Navigation Function," IEEETransactions on Systems, Man, and Cybernetics, 20: pp. 1058–1068.
[2] Lumelsky, V. and Stepanov, A. (1990), "Path-Planning Strategies for a Point Mobile Automaton Moving Amidst Unknown Obstacles of Arbitrary Shape," Autonomous Robot Vehicles,Spinger-Verlag.
[3] Kamon, I. and Rivlin, E. and Rimon E. (1996),"A New Range-Sensor Based Globally Convergent Navigation Algorithm for Mobile Robots," IEEE International Conference on Robotics and Automation.
[4] Khatib, O. (1985)," Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," IEEE International Conference on Robotics and Automation,;pp. 25-28: 500- 505.

[5] Koren, Y. and Borenstein, J. (1988)," High-Speed Obstacle Avoidance for Mobile Robotics,". Proceedings of the IEEE Symposium on Intelligent Control, pp.382-384.
[6] Borenstein, J. and Koren, Y. (1991)," The Vector Field Histogram – Fast Obstacle Avoidance for Mobile Robots," IEEE Journal of Robotics and Automation, pp. 278–288.
[7] Ulrich, I. and Borenstein, J. (1998)," VFH: Reliable Obstacle Avoidance for Fast Mobile Robots," Proceedings of the International Conference on Robotics and Automation.
[8] Ulrich, I. and Borenstein, J. (2000)," VFH: Local Obstacle Avoidance with Look-Ahead Verification," Proceedings of the IEEE International Conference on Robotics and Automation, pp. 24–28.
[9] Khatib, O. and Quinlan, S. (1993)," Elastic Bands: Connecting, Path Planning and Control," Proceedings of IEEE International Conference on Robotics and Automation.
[10] Bolandi, H. and Rezaei, M. and Smailzadeh, S. M. and Khoorany, V. (2013)," New Algorithm of Formation and Obstacle Avoidance Simultaneous Control for Quadrotorgroupe," Electronics Industries Quarterly Vol.4 No.4 Winter, pp. 47-57
[11] Bolandi, H. and Rezaei, M. and Mohsenipour, R. and Smailzadeh, S. M and Nemati, H. (2013),"Attitude Control of a Quadrotor with Optimized PIDcontroller," Intelligent Control and Automation, pp. 342-349.