# Leader Follower Tracking with Obstacle Avoidance using Circular Paths Algorithm

**Abdulmuttalib T. Rashid**
**Electrical Engineering Department**
**University of Basrah**
**Basrah, Iraq**

***Abstract-*** **This paper deals with a new algorithm called circular paths for leader follower tracking with obstacle avoidance using. In leader follower tracking, one robot acts as a leader with defined motion and the other robot acts as a follower which position itself in accordance with the position and orientation of the leader. The leader movement is dependent on an assigned trajectory and the follower movement is dependent on the circular paths algorithm. In each step, this algorithm constructs a circular path using three points represented by the next step position of the leader robot, the last step position and the current step position of the follower robot. The next position of the follower robot lays on the circumstance of the circular path and the orientation is represented by the tangent line to this circular path at this next position of the follower robot. When an obstacle intersect any circular path for the follower robot, then this path must be replaced by another circular path construct from the two positions of the follower robot and the leader position is replaced by the tangent point to the obstacle. Simulation results illustrate the soundness of this algorithm.**

**Keywords**: **Leader follower tracking; Obstacle avoidance; Apollonius' problem; circular paths algorithm.**

## I. INTRODUCTION

In recent years, the investigation of multi-robot formation using leader follower tracking has been studied by a lot of researchers. In this formation a leader robot creates a path to guide other follower robots. The multi-robot systems have many benefits over single robot [1]. These systems can perform several kinds of jobs more professionally than a single robot. Also, they can provide robustness and fault tolerance properties which they are necessary in military applications [2] and rescue missions [3]. Multi-robot system has lower cost of building than the cost of building a big and complex robot. Multi-Robot formation develop rapidly because of its applications in many areas, such as cooperative surveillance [4], teleportation structure for multi-mobile robots [5]. The formation is considered as a rigid body and develop the motion planning from the dynamics [6] and the others change some control laws to maintain an indicated formation [7, 8]. The leader-following formation navigation is used to solve the problem in the existing methods to assigning the follower robots to move in a rigid formation [9] by oriented the follower robots to move into the leader’s path. However, this motion may lead the follower robots to collide with each other due to quick turns.

The follower robot needs to update its location with respect to the leader to complete the tracking task. GPS can be used to achieve the localization but this not fit for the tracking task because they may lose the satellite signal. Sensors, such as cameras and laser range finders, can be chosen to be better methods for tracking under both indoor and outdoor situations [10–12]. Camera vision can produce more information than the laser range finders scanning with fewer cost, so that it has been used for path planning and tracking [13–15]. Camera vision-based leader-follower location estimating method has been designed for tracking of multi-robot system [16, 17]. This method was effected by lighting surroundings and was distance-limited. An effective method for solving those problems was to use a marker to recognize the leader and guess the leader-follower location using color information of the markers [18].

Leader follower tracking can be achieved by using voronoi diagram which constructed from the leader and obstacles locations. Many methods are employed a voronoi diagram to build up a path planning with obstacle avoidance in both global [19], and local environment [20]. Also voronoi diagram is implemented with a smoothness path planning and better obstacle avoidance by using iterative enhancement method [21]. The drawback of the voronoi diagram is that, when the robot follows a linear path it must be to stop at each end and restart movement again. This causes additional waste of robot power. There are several works treating the follower path of a disc. In these works the visibility graph extended to a tangent graph, by assume circular arcs on the boundary of the obstacles and joint tangents of the circular arcs [22], [23]. All these works need the value of the radius R of the disc, and hence a rebuilding is necessary for a disc of a different radius. The Voronoi diagram is not needs the value of the radius of disc, because its shortest distance to the obstacles is larger than the radius of the disc [24].

This paper proposes an algorithm named the circular paths for leader follower tracking with obstacle avoidance. This algorithm constructs a circular path for follower movement using three points represented by the next step position of the leader robot, the last step position and the current step position of the follower robot. When an obstacle intersect any circular path for the follower robot, then this path must be replaced by another circular path represent by two position of the follower robot and the tangent point to the obstacle. The rest of the paper is organized as follows. Section 2 describes the circular paths algorithm. In Sections 3, circular paths algorithm has been tested in three scenarios. Finally Section 4 draws the conclusions of the paper.

## II. CIRCULAR PATHS ALGORITHM

In this section, a circular paths algorithm for leader follower tracking with obstacle avoidance is introduced. This algorithm is used to predict the trajectory of the follower robot dependent on the trajectory of the leader one. It takes into account the observed location of the leader robot to follow it and the observed locations of the obstacles to avoid collisions with them. In this algorithm the follower robot path consist of several circular trajectories each one is obtained from three points represented by the future location of leader robot, last and current locations of follower robot. At first, the distance (L) between leader and follower robots is computed and at each step of movement the length of the movement step is increases by some factor (l) as the distance between leader and follower is greater than the half of L, else its decreases by the same factor l.

In any step, when an obstacle intersect the current circular movement path of the fo llower robot, then this path must be replaced by another circular trajectory constructed from both last and current locations of the follower robot and the tangent point to the collide obstacle. The investigation of the circular paths algorithm requires the following steps:

Step 1: sensing the initial parameters: position of leader robot $(P_l(x_l^1, y_1^1))$, position of follower robot $(P_f(x_f^1, y_f^1))$, position of leader source$((x_{S1}, y_{S1}))$, position of leader target $((x_{T1}, y_{T1}))$, radiuses of robots *(r)*, directions of leader robot and follower robot $(\theta_l, \theta_f)$.

Step 2: The next position of follower robot: To compute the next position of follower robot we need to draw a straight line between the initial positions of the two robots as shown in Fig.1. The new position of the follower robot lies on this line.

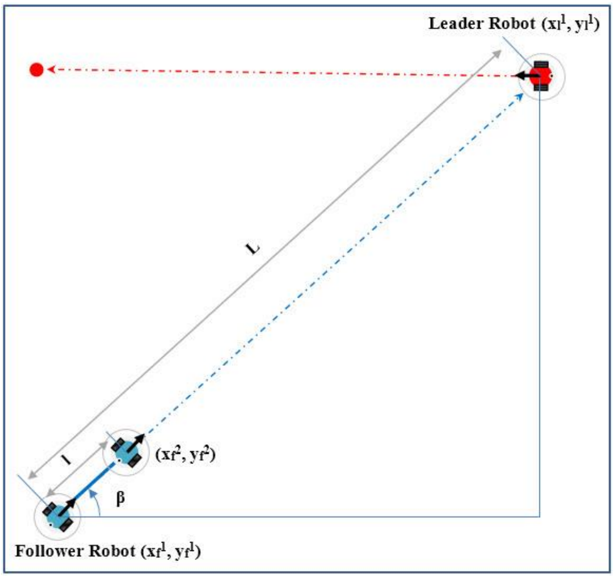



Fig. 1. Prediction of the straight line trajectory.

According to Fig.1 the next position of the follower robot is computed as follows:

1. Compute the distance L between the initial position of the leader robot and the initial position of the follower robot using equation 1.

$$L = \sqrt{\left(y_l{}^1 - y_f{}^1\right)^2 + \left(x_l{}^1 - x_f{}^1\right)^2} \qquad (1)$$

2. Compute the orientation $\beta$ of the strait line between current locations of leader and follower robots.

$$\beta = \tan^{-1}\left(\left(y_l{}^1 - y_f{}^1\right)/\left(x_l{}^1 - x_f{}^1\right)\right) \qquad (2)$$

3. Since the distance between the diameters of current and next position of the follower robot is equal to the diameter of the robot, then this distance is computed by using equation 3.

$$l = 4 * R \qquad (3)$$

where *R* is the radius of each robot.

4. The next position of the follower robot is computed using equation 4.

$$x_f{}^2 = x_f{}^1 + l \times \cos\beta$$
$$y_f{}^2 = y_f{}^1 + l \times \sin\beta \qquad (4)$$

where $(x_f^2, y_f^2)$ is the next position of the follower robot.

Step 3: Drawing the first circular trajectory to the follower robot: In this step we use the next position of leader robot and the last and current positions of the follower robot to draw the first circular trajectory as shown in Fig.2. The next position of the follower robot lies on this circular trajectory and computed as follows:

1. According to Fig. 2 we compute the next position of the leader robot using equation 5.

$$x_l{}^2 = x_l{}^1 + l$$
$$y_l{}^2 = y_l{}^1 \qquad (5)$$

where all the future positions of leader robot place on the same horizontal line.

2. The circular trajectory can be drawn using three points: the next position of leader robot $(x_l^2, y_l^2)$, the last position for the follower robot $(x_f^1, y_f^1)$, and the current position for the follower robot $(x_f^2, y_f^2)$.

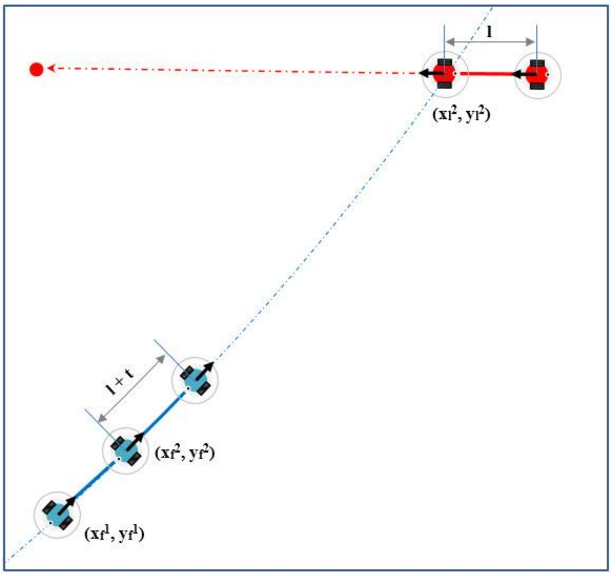



Fig. 2. Drawing the first circular trajectory to the follower robot.

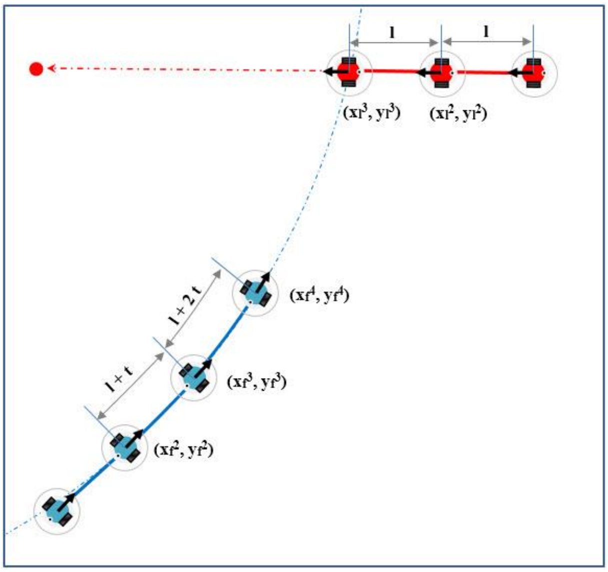



Fig. 3. Drawing the second circular trajectory to the follower robot.

3. Compute the distance $d_{22}$ between the next position of the leader robot $(x_l^2, y_l^2)$ and the next position of the follower robot $(x_f^2, y_f^2)$ using equation 6.

$$d_{22} = \sqrt{\left(y_l{}^2 - y_f{}^2\right)^2 + \left(x_l{}^2 - x_f{}^2\right)^2} \qquad (6)$$

4. Since the distance $d_{22}$ is greater than the distance $L/2$ then the third position of the follower robot must far about $l + t$, where t is a constant chosen in a way to make the maximum speed of robot within its kinematic. At each new step the new position of follower robot is far from last position by $l + t$, $l + 2t$, … since the distance $d_{22}$ is greater than the distance $L/2$ elsewhere the distances is decreases by $l - t$, $l - 2t$, … until reach to the original distance $l$.

Step 4: Computing the coordinate access of the follower robot: At beginning, the procedure used in last step is repeated to calculate the next position of the follower robot $(x_l^3, y_l^3)$ as shown in Fig.3, drawing the next circular trajectory using three points: the next position of leader robot $(x_l^3, y_l^3)$, the last position for the follower robot $(x_f^2, y_f^2)$, and the current position for the follower robot $(x_f^3, y_f^3)$, and computing the distance $d_{33}$ between the next position of the leader robot $(x_l^3, y_l^3)$ and the current position of the follower robot $(x_f^3, y_f^3)$ using equation 6. Now if the distance $d_{33}$ is greater than the distance $L/2$ then the new position of follower robot is far from last position by $l + 2t$ as shown in Fig. 3.

To compute the coordinate access of the new position of the follower robot we need to know the coordinate access to the center point and radius of the circular trajectory. This process can be achieved as follows:

1. As shown in Fig. 4, the circular trajectory which passes through the points: $(x_l^3, y_l^3)$, $(x_f^2, y_f^2)$, and $(x_f^3, y_f^3)$ has the following equation:

$$(x - x_C{}^3)^2 + (y - y_C{}^3)^2 = R_3{}^2 \qquad (7)$$

where $(x_C^3, y_C^3)$ is the center point axes of the circular trajectory and $R_3$ is the radius.

2. Since the three points lie on this trajectory, their coordinates will satisfy the following equation.

$$\begin{aligned} \left(x_f{}^2 - x_C{}^3\right)^2 + \left(y_f{}^2 - y_C{}^3\right)^2 &= R_3{}^2 \\ \left(x_f{}^3 - x_C{}^3\right)^2 + \left(y_f{}^3 - y_C{}^3\right)^2 &= R_3{}^2 \qquad (8) \\ (x_l{}^3 - x_C{}^3)^2 + (y_l{}^3 - y_C{}^3)^2 &= R_3{}^2 \end{aligned}$$

3. By solving these three equations we can obtained the coordinate access $(x_C^3, y_C^3)$ of the center point to the circular trajectory as shown in equation 9 and equation 10 [25].

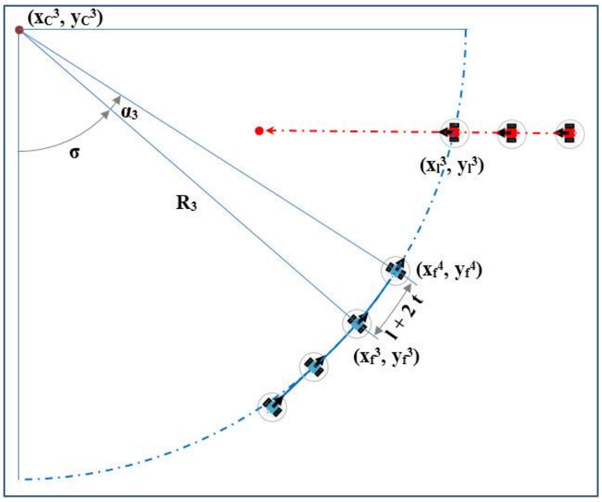



Fig. 4. Computing the coordinate access to the follower robot at the second circular trajectory.

$$x_C{}^3 = \frac{\begin{bmatrix} x_f{}^2+y_f{}^2 & y_f{}^2 & 1 \\ x_f{}^3+y_f{}^3 & y_f{}^3 & 1 \\ x_l{}^3+y_l{}^3 & y_l{}^3 & 1 \end{bmatrix}}{\begin{bmatrix} x_f{}^2 & y_f{}^2 & 1 \\ x_f{}^3 & y_f{}^3 & 1 \\ x_l{}^3 & y_l{}^3 & 1 \end{bmatrix}} \qquad (9)$$

$$y_C{}^3 = \frac{\begin{bmatrix} x_f{}^2 & x_f{}^2+y_f{}^2 & 1 \\ x_f{}^3 & x_f{}^3+y_f{}^3 & 1 \\ x_l{}^3 & x_l{}^3+y_l{}^3 & 1 \end{bmatrix}}{\begin{bmatrix} x_f{}^2 & y_f{}^2 & 1 \\ x_f{}^3 & y_f{}^3 & 1 \\ x_l{}^3 & y_l{}^3 & 1 \end{bmatrix}} \qquad (10)$$

4. Compute the radius $R_3$ of the circular trajectory which represented by the distance between the center of this trajectory and one of the three points lies on it is circumference.

$$R_3 = \sqrt{\left(x_C{}^3 - x_f{}^2\right)^2 + \left(y_C{}^3 - y_f{}^2\right)^2} \qquad (11)$$

5. Compute the angle $\sigma$ which represent the orientation of the line between center point and current location of follower robot using equation 12.

$$\sigma = \tan^{-1}\left(\frac{y_f{}^3 - y_C{}^3}{x_f{}^3 - x_C{}^3}\right) \qquad (12)$$

6. Compute the angle $\alpha_3$ between the current and the new locations of the follower robot using equation 13.

$$\alpha_3 = \frac{l+2\,t}{R_3} \qquad (13)$$

7. According to Fig. 4 the coordinate access of new location of the follower robot is computed using equation 14.

$$x_f{}^4 = x_C{}^3 + R_3 \ \sin(\sigma + \alpha_3)$$
$$y_f{}^4 = y_C{}^3 + R_3 \ \cos(\sigma + \alpha_3) \qquad (14)$$

Step 5: Obstacle avoidance for the follower robot: when an obstacle intersect the circular trajectory between the follower and the leader robots then we must choose another circular path to avoid the collision. There are two possible paths to solve this problem:

(a) The first circular path to avoid the collision: This path is chosen when the follower robot is far from the obstacle to produces a smooth and safe trajectory as shown in Fig. 5. The construction of this path is dependent on the last and current positions of the follower robot (($x_f^n$, $y_f^n$) and ($x_f^{n+1}$, $y_f^{n+1}$) as well as a touch point to the obstacle.

(b) The second circular path to avoid the collision: This path is chosen when the follower robot is behind the obstacle to produces a smooth trajectory and to prevent the follower robot to be far from the leader robot as shown in Fig. 6. The construction of this path is dependent on the current positions of the follower and the leader robot (($x_f^{n+1}$, $y_f^{n+1}$) and ($x_l^{n+1}$, $y_l^{n+1}$) as well as the touch point to the obstacle.

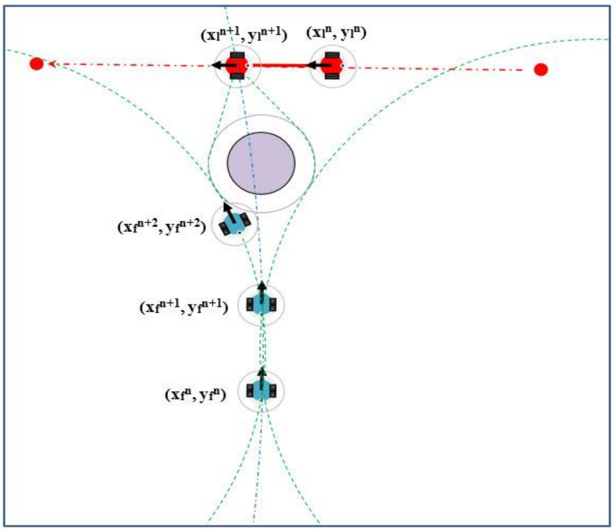



Fig. 5. The first circular path to avoid the collision with obstacle.

Both the two methods based on drawing a circular path passes through two points and touch point to circle. These types of methods are called the Apollonius' problem with two points and a circle (PPC) which is one type of the Apollonius' Problems [26]. The method is investigated by drawing any circle that passes through the given points, as example A and B and intersects with the given circle in two point, say, C and D. Let E be the intersection of line AB and line CD. Draw EF and EG as tangents from E to the given circle. Circles pass through point ABF and ABG solve the problem. The implementation of the PPC method is as follows:

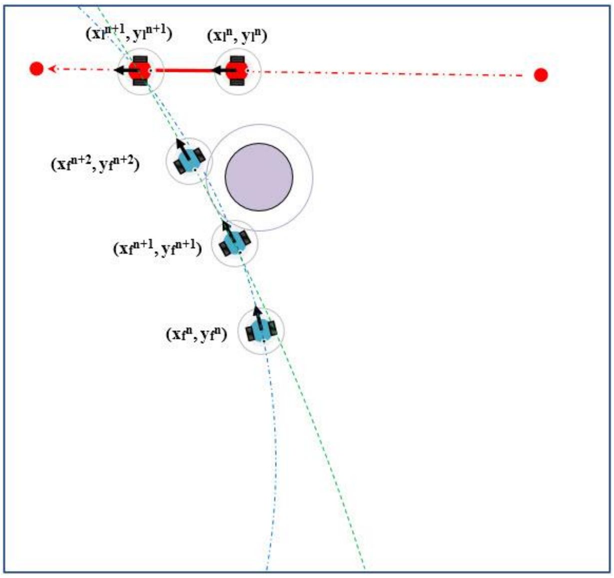



Fig. 6. The second circular path to avoid the collision with obstacle.

1. The solving to the Apollonius' problem with two points and a circle is started by drawing any circle that passes through the given points A and B and intersects with the given circular obstacle in two points C and D as shown in Fig.7.a. For easiness, the drawing circle is chosen to pass through the center point O of the obstacle. Now we can calculate the coordinate access to the center point ($x_P$, $y_P$) of the drawing circle that pass through the points: ($x_f^3$, $y_f^3$), ($x_f^4$, $y_f^4$) and ($x_O$, $y_O$), using the equations 15 and 16.

$$x_P = \frac{\begin{bmatrix} x_f{}^3+y_f{}^3 & y_f{}^{32} & 1 \\ x_f{}^4+y_f{}^4 & y_f{}^4 & 1 \\ x_O+y_O & y_O & 1 \end{bmatrix}}{\begin{bmatrix} x_f{}^3 & y_f{}^3 & 1 \\ x_f{}^4 & y_f{}^4 & 1 \\ x_O & y_O & 1 \end{bmatrix}} \quad (15)$$

$$y_P = \frac{\begin{bmatrix} x_f{}^3 & x_f{}^3+y_f{}^3 & 1 \\ x_f{}^4 & x_f{}^4+y_f{}^4 & 1 \\ x_O & x_O+y_O & 1 \end{bmatrix}}{\begin{bmatrix} x_f{}^3 & y_f{}^3 & 1 \\ x_f{}^4 & y_f{}^4 & 1 \\ x_O & y_O & 1 \end{bmatrix}} \quad (16)$$

The next step in the Apollonius' problem with two points and a circle is to draw a straight line between the points C and D and straight line between the points A and B as shown in Fig.8. Then the coordinate access of the intersection point E between these two lines are computed using equation 24.

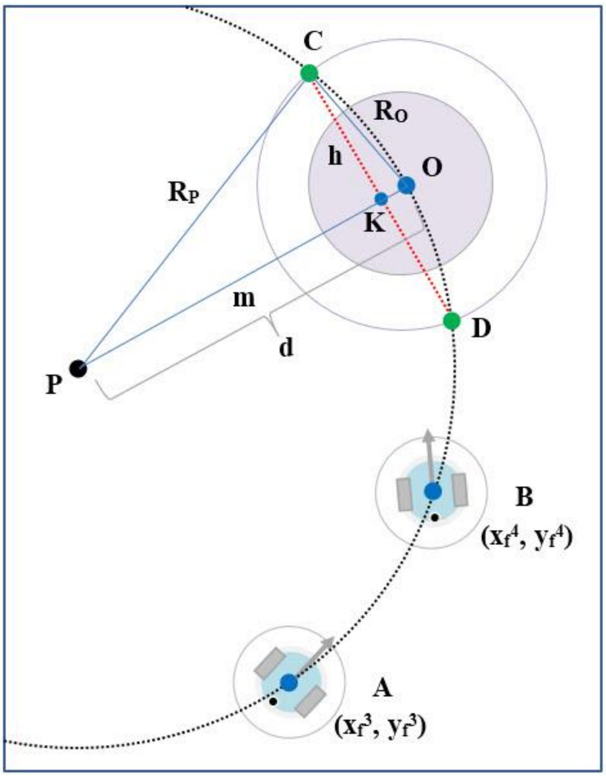



Fig. 7. Determining the intersection points between the drawing circle and the circular obstacle.

$$x_E = \frac{(x_C y_D - y_C x_D)(x_A - x_B) - (x_C - x_D)(x_A y_B - y_A x_B)}{(x_C - x_D)(y_A - y_B) - (y_C - y_D)(x_A - x_B)}$$

$$(24)$$

$$y_E = \frac{(x_C y_D - y_C x_D)(y_A - y_B) - (y_C - y_D)(x_A y_B - y_A x_B)}{(x_C - x_D)(y_A - y_B) - (y_C - y_D)(x_A - x_B)}$$

Also, the radius $R_P$ of the drawing circle that connect the point *($x_P$, $y_P$)* to the point *($x_f^3$, $y_f^3$)* is computed by using equation 17.

$$R_P = \sqrt{\left(x_P - x_f{}^3\right)^2 + \left(y_P - y_f{}^3\right)^2} \quad (17)$$

2. To calculate the coordinate access of the two intersection points between the drawing circle and circular obstacle we must drive some parameters as shown in Fig.7. The derivations of the expression to calculate such point is omitted since it is based on quite simple geometrical constructions.

$$d = \sqrt{(x_P - x_O)^2 + (y_P - y_O)^2} \quad (18)$$

$$m = \frac{R_P{}^2 - R_O{}^2 + d^2}{2*d} \quad (19)$$

since the drawing circle passes through the center of the obstacle then $d = R_P$

$$h = \sqrt{{R_P}^2 - m^2} \tag{20}$$

$$x_K = x_P + m * \frac{x_O - x_P}{R_P}$$
$$y_K = y_P - m * \frac{y_P - y_O}{R_P} \tag{21}$$

this equation is written with respect to the original point of the screen which lies at the upper left side.

3. The coordinate access for the intersection points between the drawing circle and circular obstacle is computed using equations 22 and 23. For point C:

$$x_C = x_K - h * \frac{y_P - y_O}{R_P}$$
$$y_C = y_K - h * \frac{x_O - x_P}{R_P} \tag{22}$$

and for point D:

$$x_D = x_K + h * \frac{y_P - y_O}{R_P}$$
$$y_D = y_K + h * \frac{x_O - x_P}{R_P} \tag{23}$$

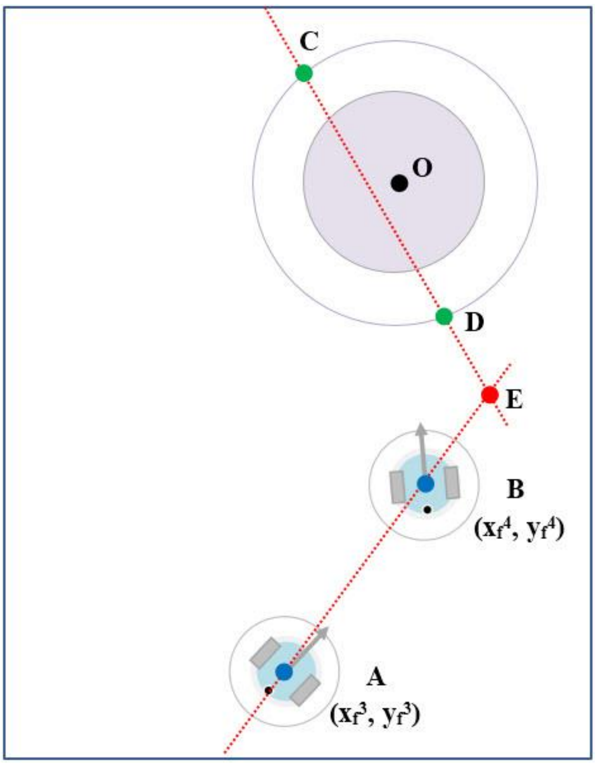



Fig. 8. Computing the intersection point E between CD line and AB line.

4. The Apollonius' problem with two points and a circle is completed by drawing two tangent lines from the intersection point E to the circular obstacle as shown in Fig.9. These two tangent lines produce two tangent points F and G with the circular obstacle. To compute the coordinate access of these two tangent points we treated them as two intersection points between two circles represented by the circular obstacle and a circle constructed from the tangent lines as radiuses in circle with center point E. The procedure used to compute the coordinate access of the intersection points F and G in Fig.9 is similar to the procedure used in computes the coordinate access of the intersection points C and D in Fig.7.

$$d_1 = \sqrt{(y_E - y_O)^2 + (x_E - x_O)^2} \tag{25}$$

$$R_E = \sqrt{(d_1)^2 - (R_O)^2} \tag{26}$$

The coordinate access of the points F and G is computed by using the equations 19 – 23.

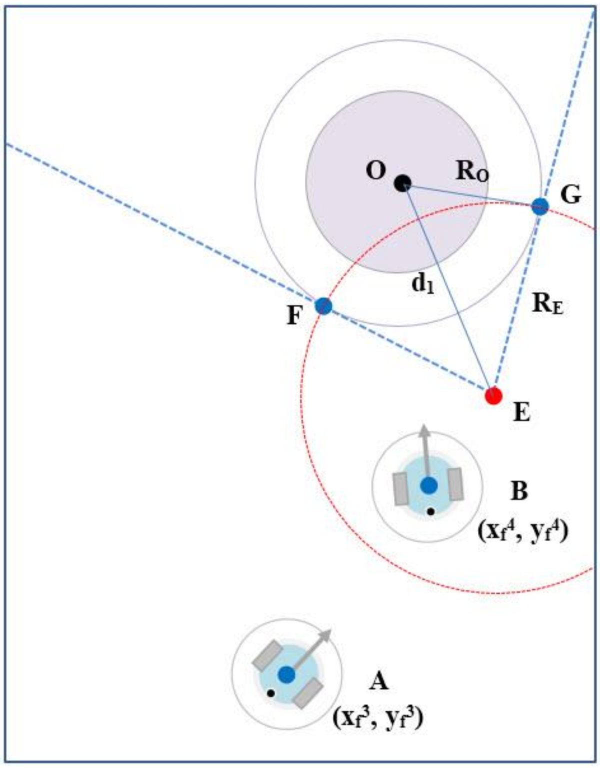



Fig. 9. Computing the coordinate access of the tangent points F and G.

5. The final step in the Apollonius' problem with two points and a circle is to draw the two circles that pass through the points A and B and tangent to the circular obstacle. The first circuit is drawn by three points represented by A, B and F and the second circuit is drawn by the three points A, B and G as shown in Fig. 10. The coordinate access of the center point to each circle is computed in similar manner to the circle drawn in Fig. 4 using equations 9 and 10.

Fig. 11 shows the applied of the Apollonius' problem with two points and a circle to avoid the collision with the circular obstacle. The shortest path is chosen with path has center point ($x_C^4$, $y_C^4$) and starting point ($x_f^4$, $y_f^4$) and ending point ($x_f^5$, $y_f^5$). The procedure used to compute the position of ending point is similar to the procedure used to compute the ending point ($x_f^4$, $y_f^4$) in Fig.4 using equations 12 – 14.

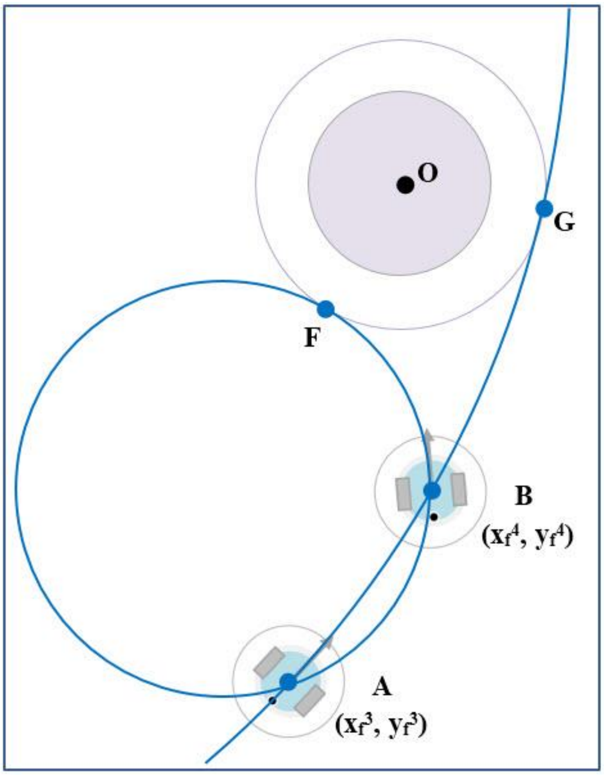



Fig. 10. Drawing the two circles that pass through two points and tough the circular obstacle.

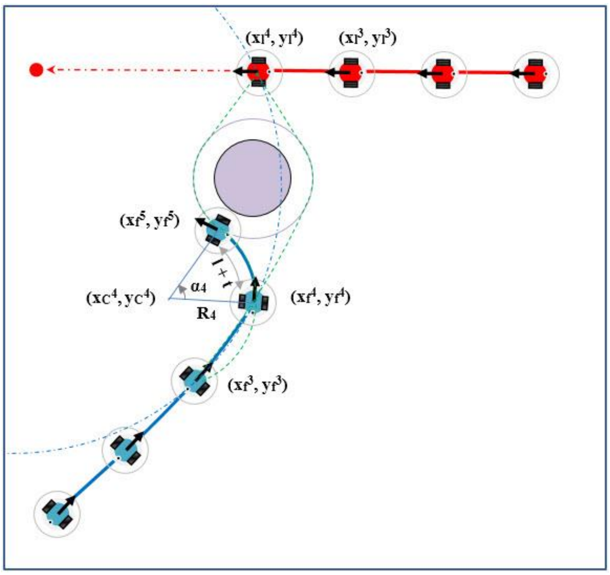



Fig. 11. Circular trajectory to the follower robot with obstacle avoidance.

Step 6: Obstacle avoidance with the second circular path: Also in this step, the circular trajectory among the last and current position of the follower robot ($x_f^4$, $y_f^4$) and ($x_f^5$, $y_f^5$) and the next position to the leader robot ($x_l^5$, $y_l^5$) still collide the circular obstacle. Since the follower robot is near to obstacle so that we choose the second circular path to avoid the collision as shown in Fig. 12. The procedure used to compute the next position to the follower robot ($x_f^6$, $y_f^6$) is similar to the procedure used to compute the position ($x_f^5$, $y_f^5$) to the follower robot in last step. Also, this procedure dependent on the Apollonius' problem with two points and a circle to avoid the collision.

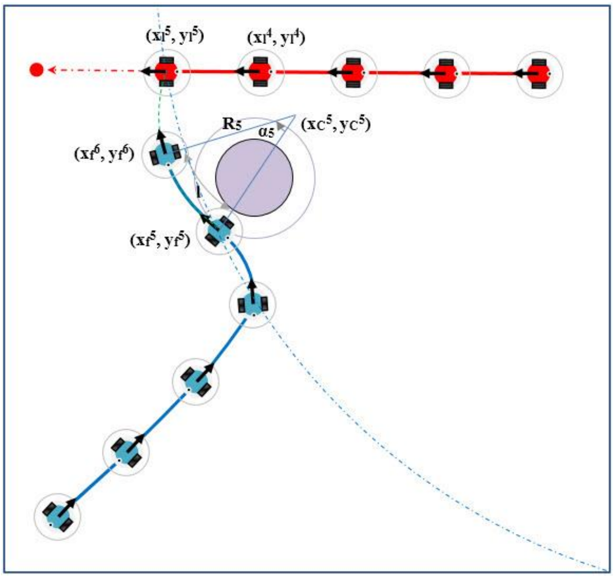



Fig. 12. The second circular trajectory to the follower robot to avoid the collision.

Step 7: Complete the movement of the follower robot toward the leader robot trajectory: The follower robot track the leader robot movement with the same steps of procedure as shown in Fig. 13. The circular trajectory at this step is constructed by the last position ($x_f^5$, $y_f^5$) and the current position ($x_f^6$, $y_f^6$) of the follower robot and the next position ($x_l^6$, $y_l^6$) of the leader robot. The circular trajectory obtained has no collision in this step, so that the follower robot use the same procedure in step 4 to compute the next position ($x_f^7$, $y_f^7$) of movement.

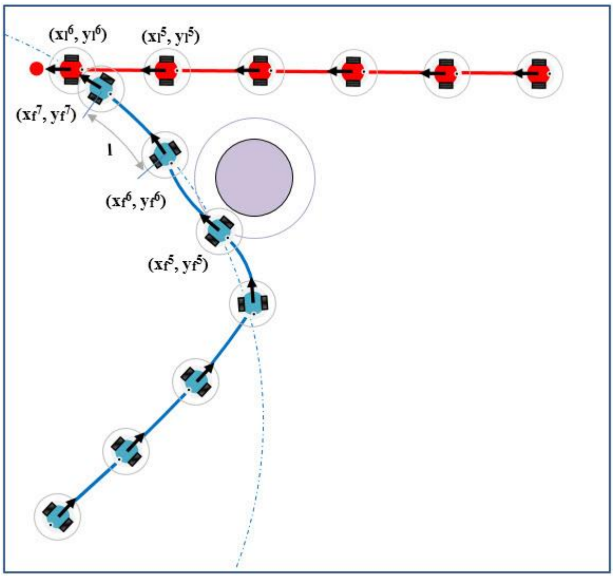



Fig. 13. Complete the movement of the follower robot toward the leader robot.

## III. SIMULATION RESULTS

The new leader follower tracking algorithm is simulated with two robots using Visual basic programming language 12 and tested in Windows environment using an Intel core i5 CPU 2.53 GHz processor. The introduced algorithm has been tested on three different scenarios:

– Linear tracking scenario. In this scenario the leader robot moves on a straight line path and the follower robot track this trajectory. The environment used in this scenario has no obstacles.

– Circular tracking scenario. In this case the leader robot moves on a circular path and also the environment has no obstacles.

– Linear tracking scenario with obstacle. In this scenario the leader robot moves also, on a straight line but the environment has some circular obstacles and the follower robot track the movement of the leader and avoid the collision.

For all scenarios we discuss the circular paths algorithm with two cases: the first one is simulated with steps have different lengths dependent on the position of the follower robot with respect to the leader one and the other is simulated with the same algorithm but with equal length for all steps of movement. For a measureable analysis of the circular paths algorithm, we used the following performance metrics:

1. Leader robot tracking (LRT): this metric is used to measure the tracking occurrence between leader and follower robots with respect to the steps of the leader movement.
2. Leader path tracking (LPT): this metric is used to measure the tracking occurrence between the leader robot path and the follower robot with respect to the steps of leader movement.

Fig. 14 shows the linear tracking scenario with equal length of the steps for the follower robot. From Fig. 14.a, b, c and d we note that all the steps length of the follower robot have the same length. The linear tracking scenario is shown in Fig. 15, where the steps of movement for follower robot is increases sequentially until the distance between leader and follower robots reach to the half of the original distance at the beginning of the movement. When the distance is became less than the half of the original distance then the steps of movements are decreases sequentially until reach to the original length of the step. From Fig. 15.a, b, c and d we notice that the steps of movement of the follower robot have different lengths.

Relating to the number of steps the follower robot with incremental movement length has better performance than the equal movement to track the leader robot. Fig. 16 shows that the tracking between leader and follower robot is occurred at eleventh step at equal steps movement for follower robot where it is occurred at fifth step at increment movement steps. Also the tracking between the trajectory of the leader robot and the follower robot has better performance when the follower robot moves in increment steps as shown in Fig. 17. The zero pixel for the distance between leader path and the follower robot means that the tracking is occurred 100% and the positive value means the tracking is at one side and the negative value is at the other side. Also, Fig. 17 shows that the tracking is occurred at the ninth step of movement with increment distance steps and the tracking is occurred at the thirteenth step of movement when this movement occurred in equal distance steps.

The second scenario (circular tracking scenario) is implemented with circular trajectory to the leader robot. Fig. 18 shows the implementation of the scenario for the follower robot that has equal movement for the steps lengths. Fig. 18.a, b, c, d, e and f show the movement steps of follower robot to reach to the leader one. The second scenario also repeated for the circular tracking scenario but with different length of the movement steps. Also these steps at beginning are increase until the distance between leader and follower robots reach to the half of the original distance and after that these steps start to decrease. Fig. 19.a, b, c, d, e and f show the movement steps of follower robot to reach to the leader one.

The circular tracking scenario shows that the leader robot tracking and the leader path tracking metrics have less performance than the linear tracking scenario, that is mean the follower robot needs more steps to reach to the leader robot and to track the path of this robot. For equal distances steps for the follower robot we notice that this robot needs about 16 steps to reach to the leader robot where in increment distances steps the follower robot needs about 8 steps to reach to the leader one as shown in Fig. 20. The follower robot is track the circular trajectory of the leader robot at 18 steps when the movement steps are equal in distances as shown in Fig.21 and at 10 steps when the movement steps have different values.

The third simulation is implemented with environment has several circular obstacles and leader robot moves only in straight line. This simulation is repeated for both equal and increment length of distances for each step of movement.

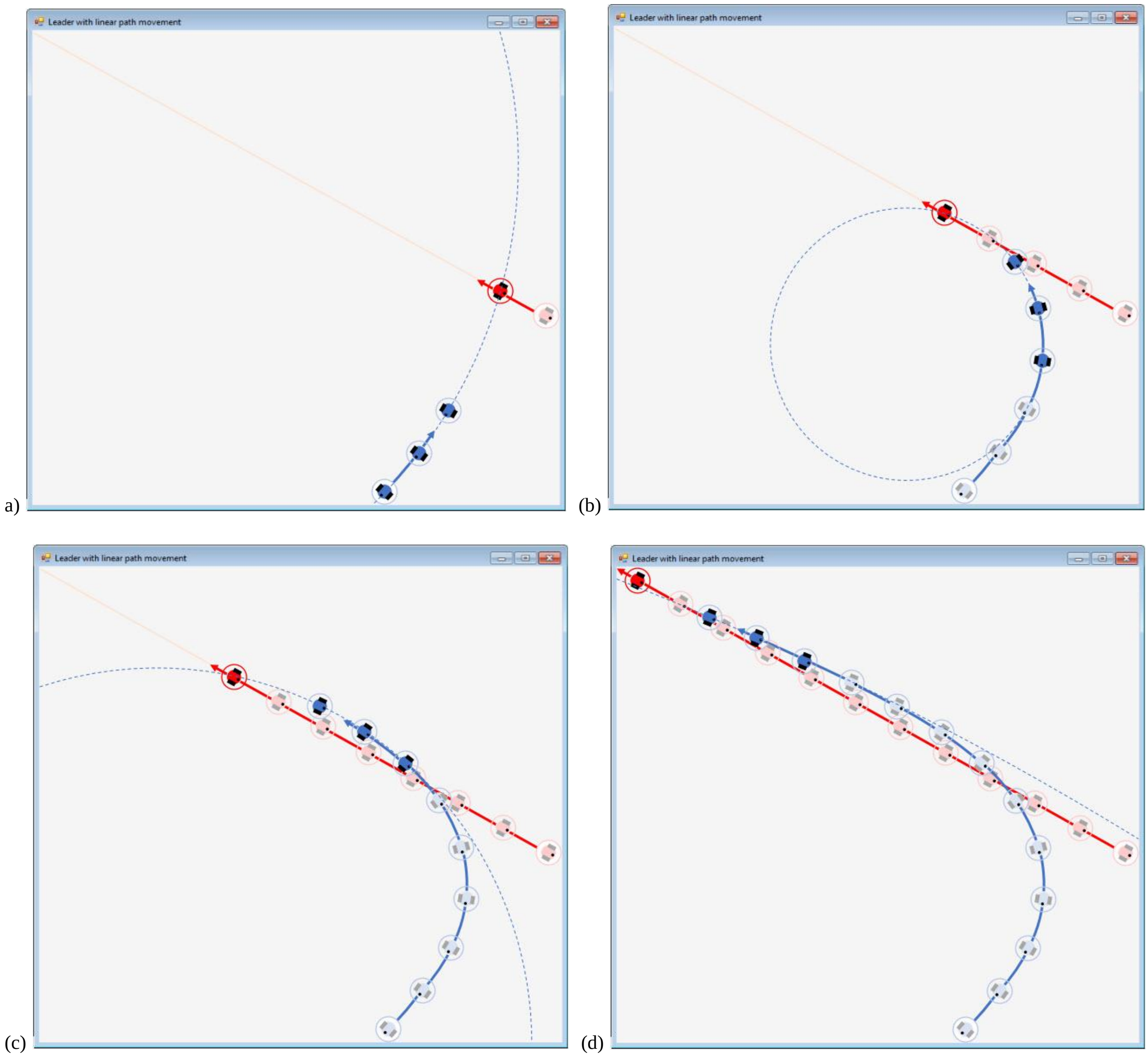



Fig. 14. Leader follower tracking using circular paths algorithm with equal steps in different positions of a straight movement line: (a) step = 1; (b) step = 4; (c) step = 7; (d) step = 10.

Fig. 22 shows the simulation of the movement of the follower robot in environment has 16 circular obstacles with uniform distributed. This robot track the straight line movement of the leader robot with equal movement steps. The same simulation is repeated in Fig. 23 but with increment distance steps for the follower robot.

The third simulation shows that the circular paths algorithm produces safe and smooth trajectory to the follower robot when it moves to track the leader robot. The follower robot with incremental movement distance has better performance than the equal movement to track the leader robot with obstacle avoidance. 10 steps required to the follower robot to track the leader when steps have equal distances and 5 steps required when steps have incremental distances as shown in Fig. 24. Also, for the case when the follower robot is track the path of the leader one we found that the steps with incremental distances for the follower movement have better performance than the equal movement steps as shown in Fig.25. 12 steps required for equal movement steps and 8 steps required for the increment movement steps.

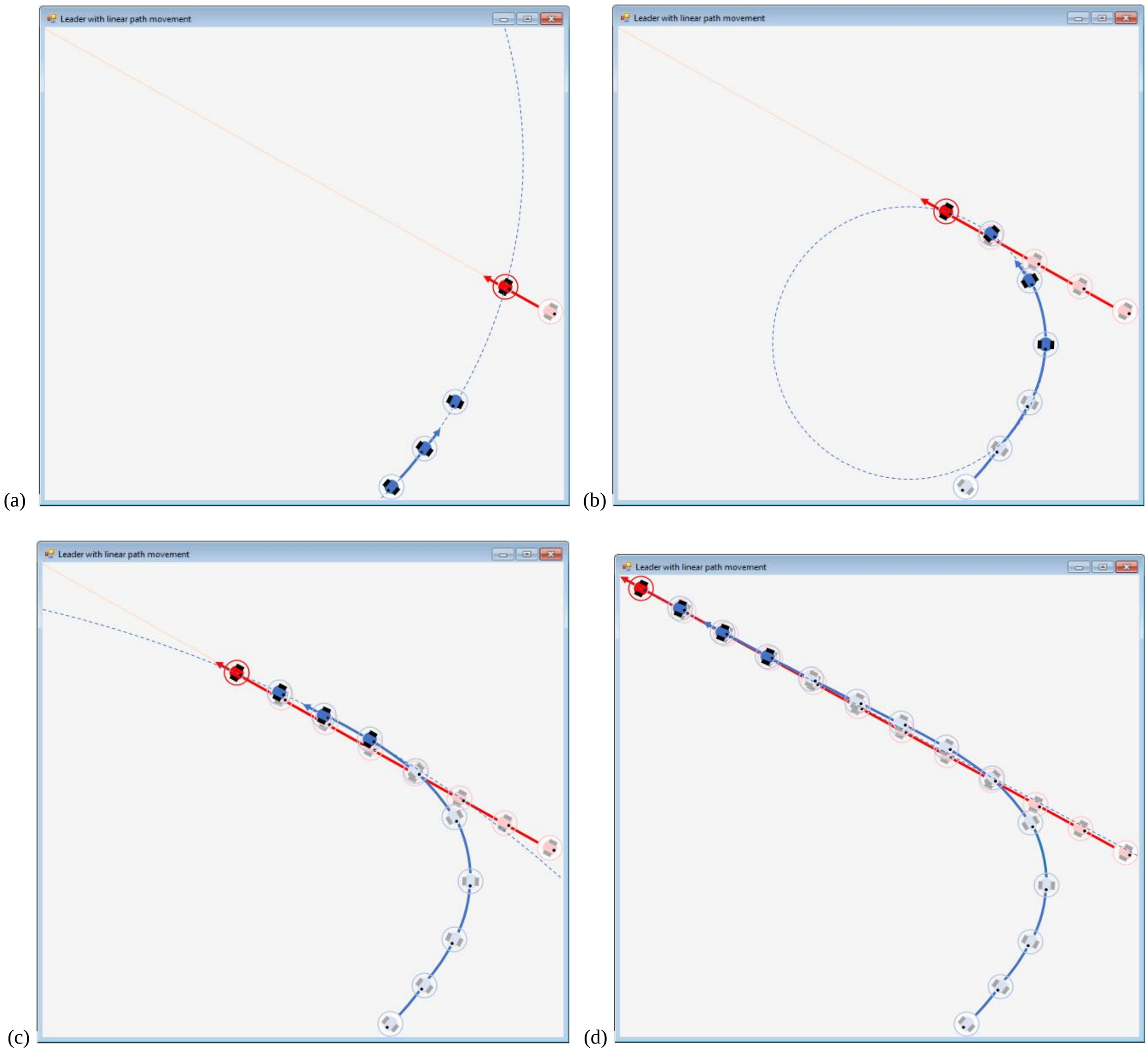



Fig. 15. Leader follower tracking using circular paths algorithm with incremental steps in different positions of a straight movement line: (a) step = 1; (b) step = 4; (c) step = 7; (d) step = 10.

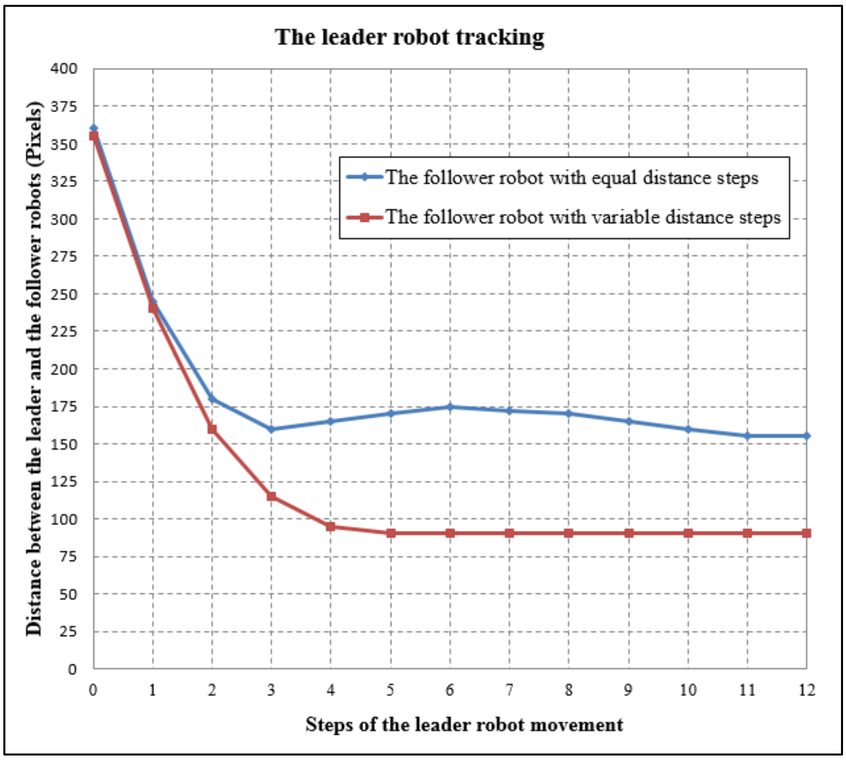
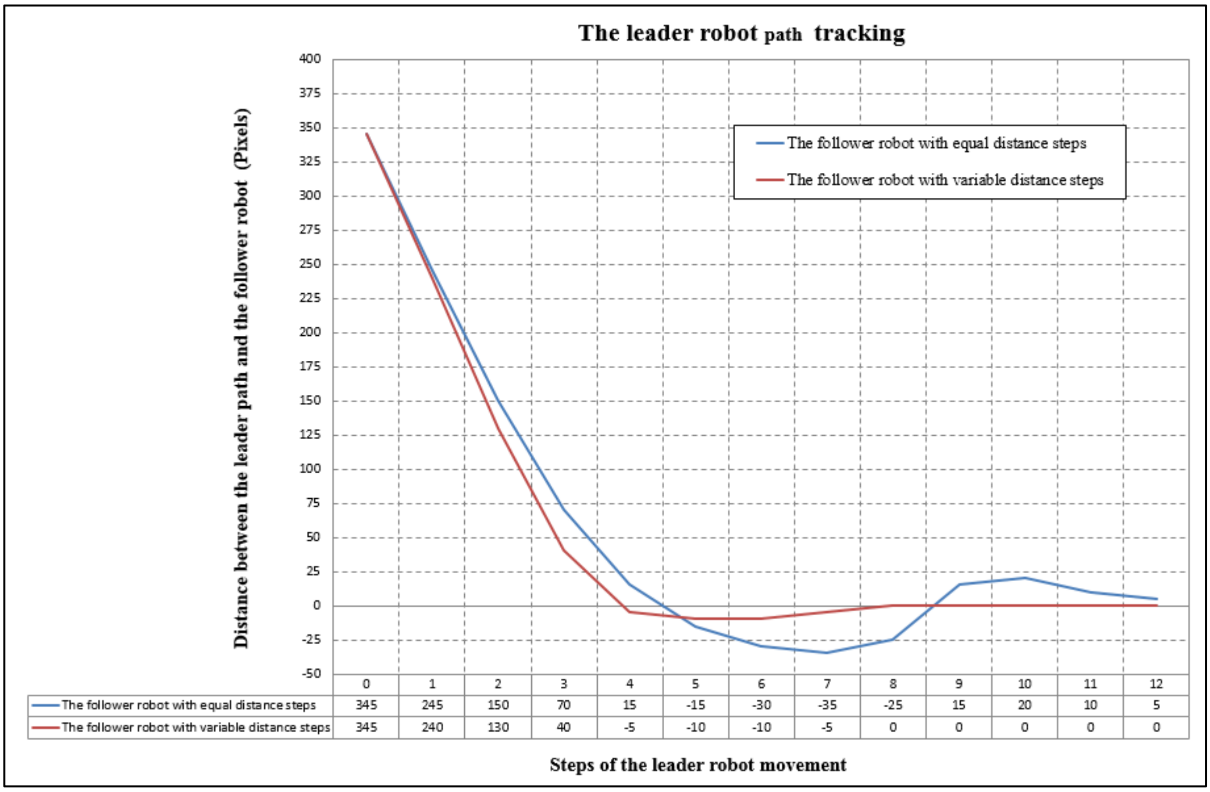



Fig. 16. Illustration of the leader robot tracking metric on a straight line trajectory when the follower robot move on equal and increment steps.




| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| The follower robot with equal distance steps | 345 | 245 | 150 | 70 | 15 | -15 | -30 | -35 | -25 | 15 | 20 | 10 | 5 |
| The follower robot with variable distance steps | 345 | 240 | 130 | 40 | -5 | -10 | -10 | -5 | 0 | 0 | 0 | 0 | 0 |

Fig. 17. Illustration of the leader path tracking metric on a straight line trajectory when the follower robot move on equal and increment steps.

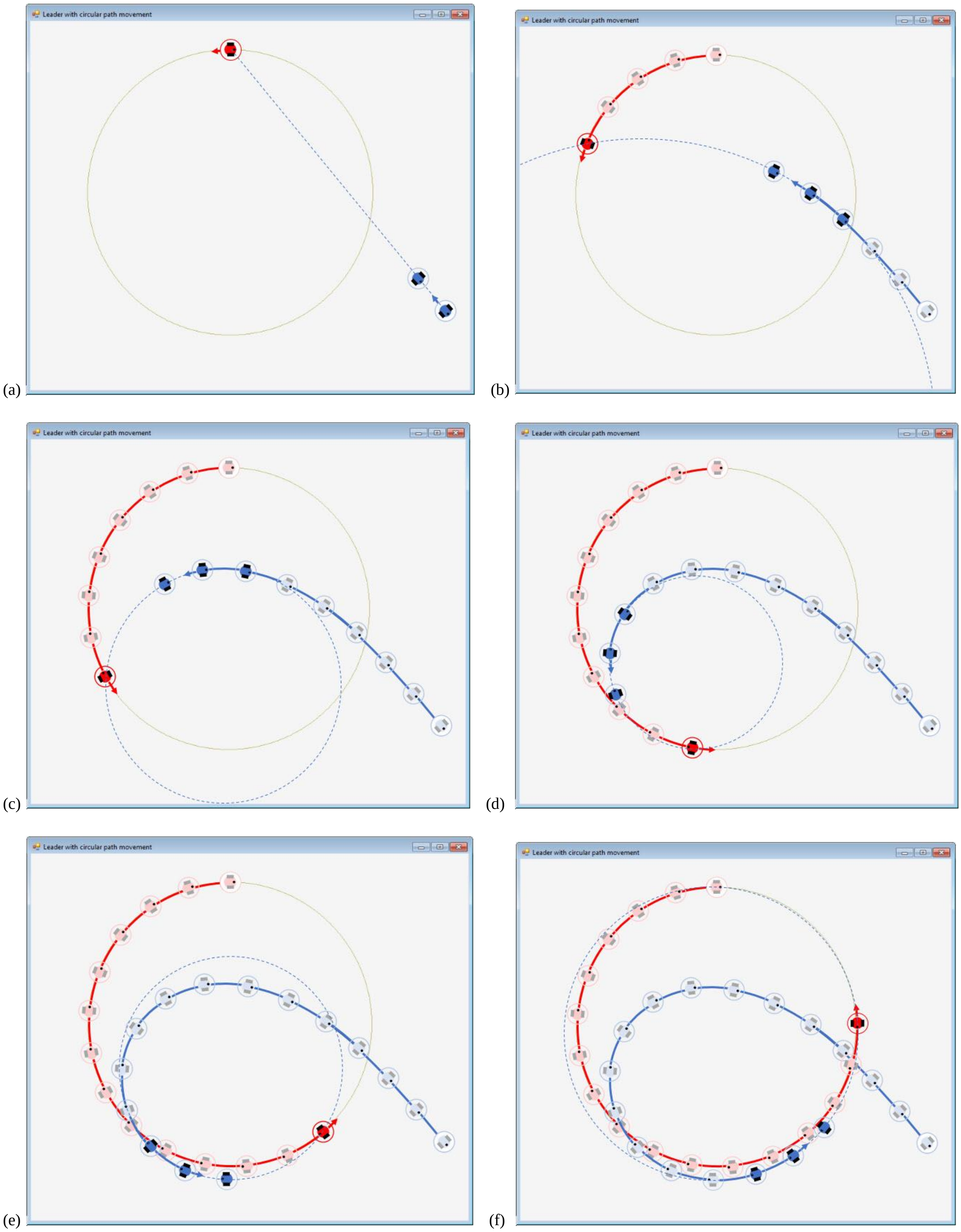



Fig. 18. Leader follower tracking using circular paths algorithm with equal steps in different positions of a circular movement path: (a) step = 0; (b) step = 4; (c) step = 7; (d) step = 10; (e) step = 13; (f) step = 16.

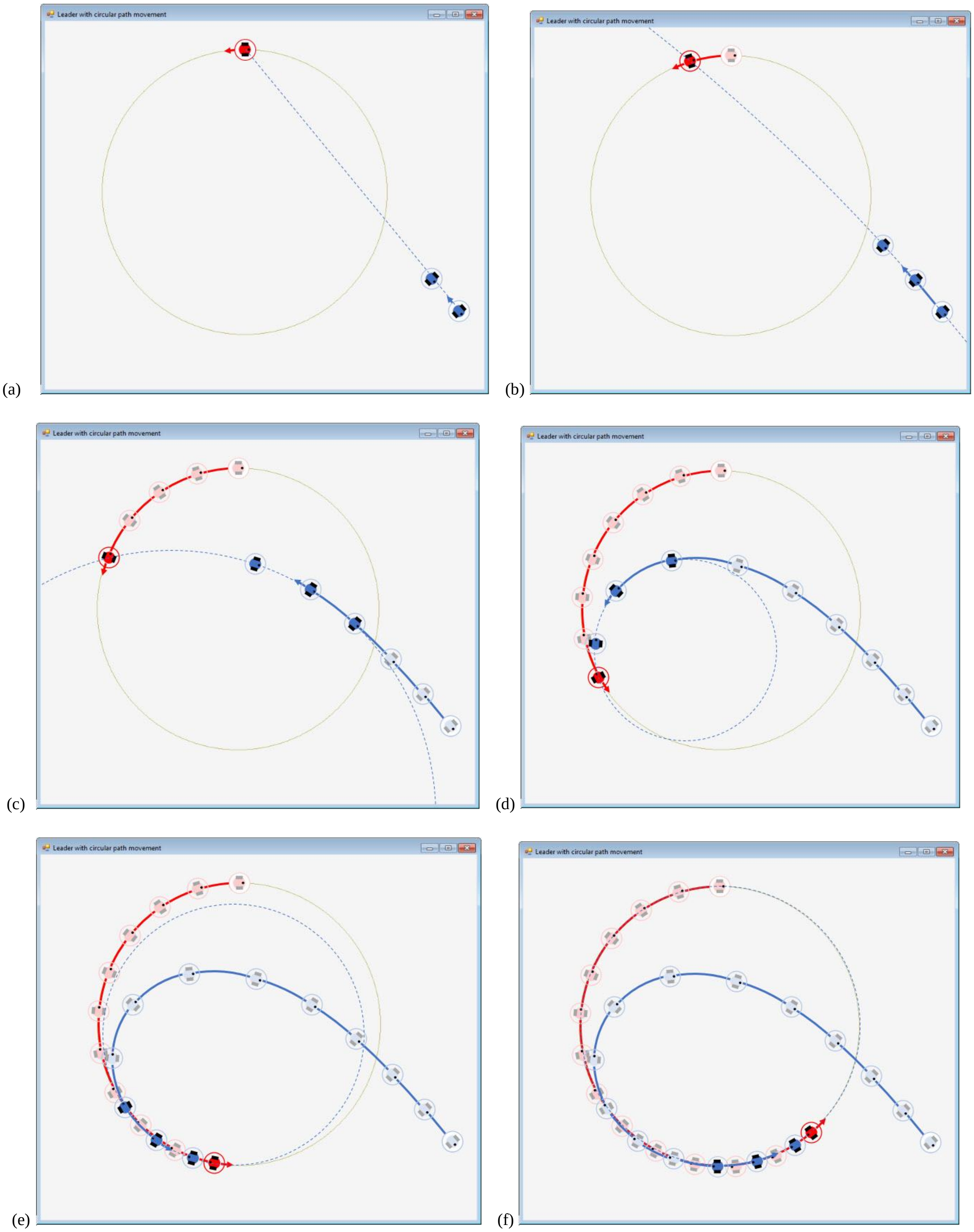



Fig. 19. Leader follower tracking using circular paths algorithm with incremental steps in different positions of a circular movement path: (a) step = 0; (b) step = 1; (c) step = 4; (d) step = 7; (e) step = 10; (f) step = 13.

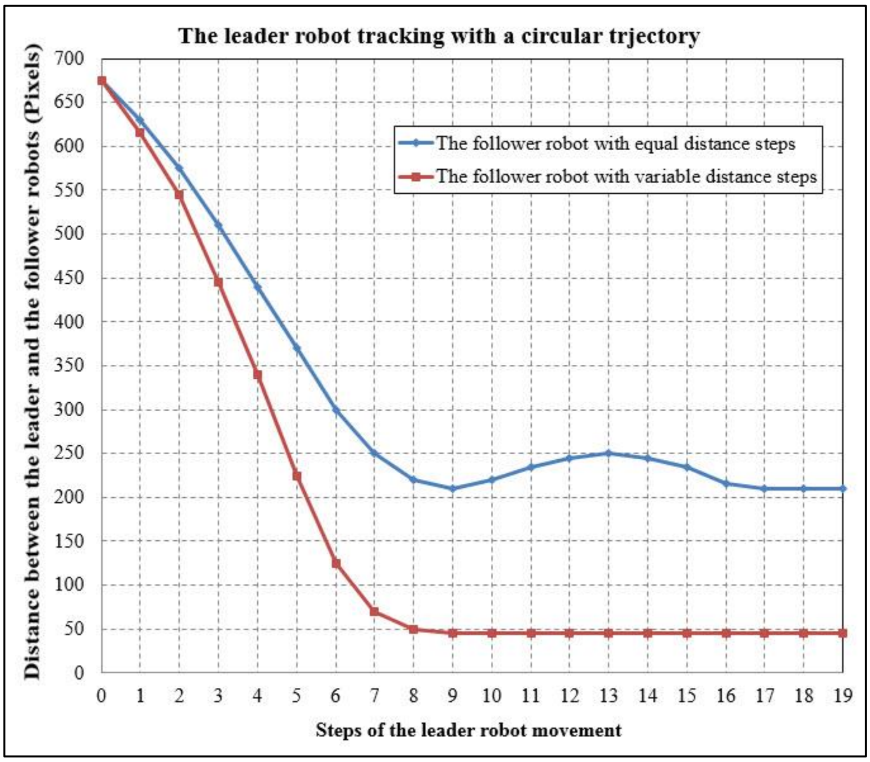



Fig. 20. The tracking occurrence between leader and follower robots with respect to equal circular movement steps.

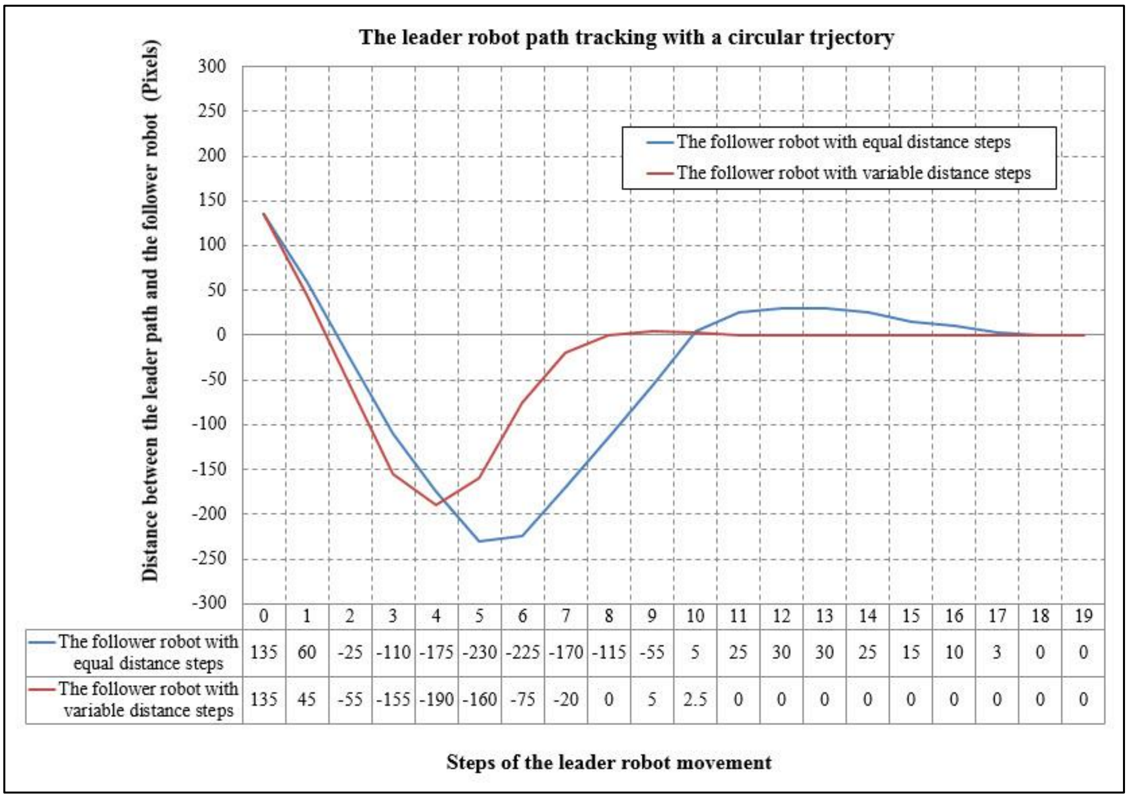



Fig. 21. The tracking occurrence between leader and follower robots with respect to incremental circular movement steps.

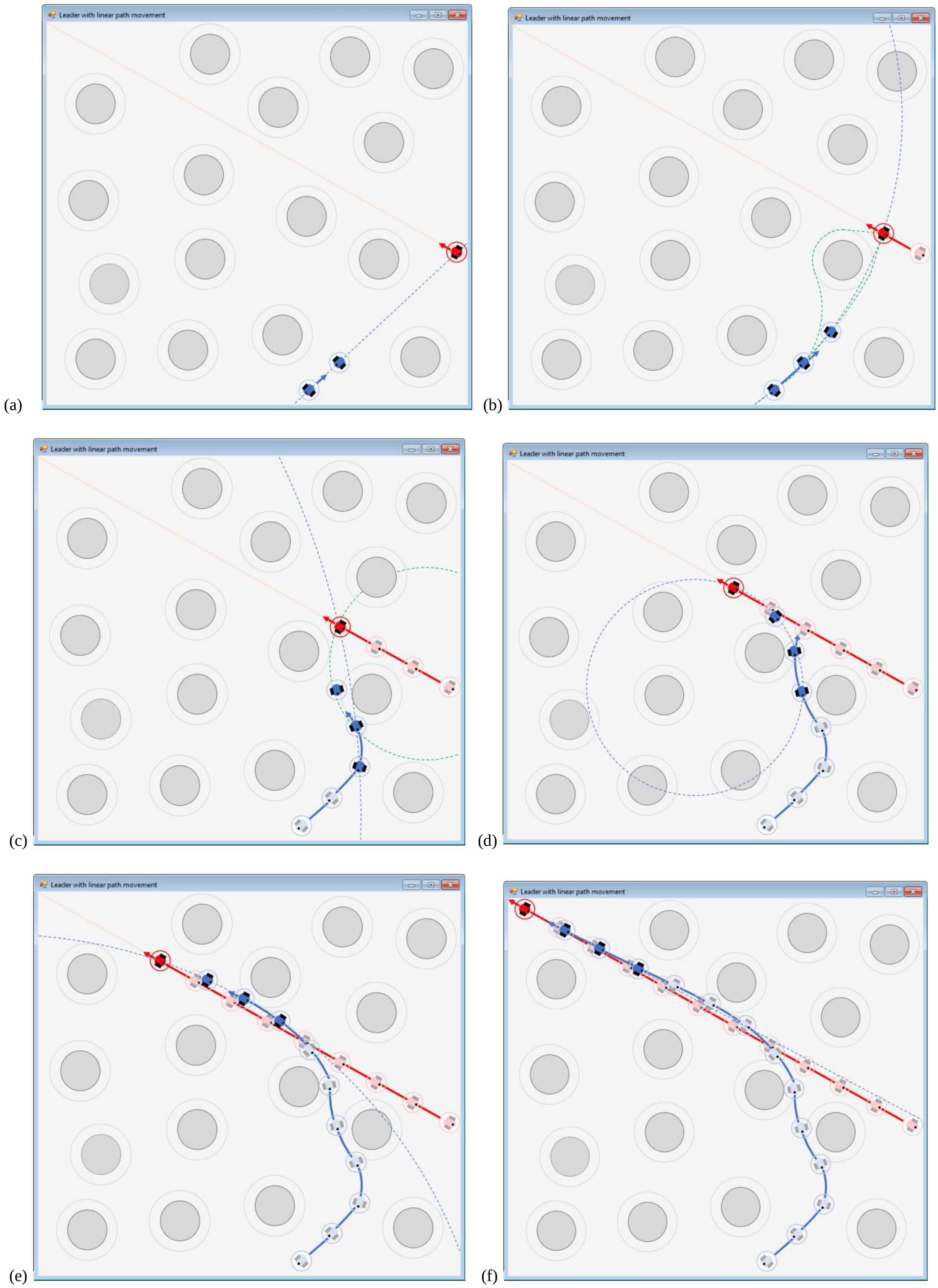



Fig. 22. Leader follower tracking with obstacles avoidance using circular paths algorithm with circular trajectory equal steps at different positions of steps: (a) step = 0; (b) step = 1; (c) step = 3; (d) step = 5; (e) step = 8; (f) step = 11.

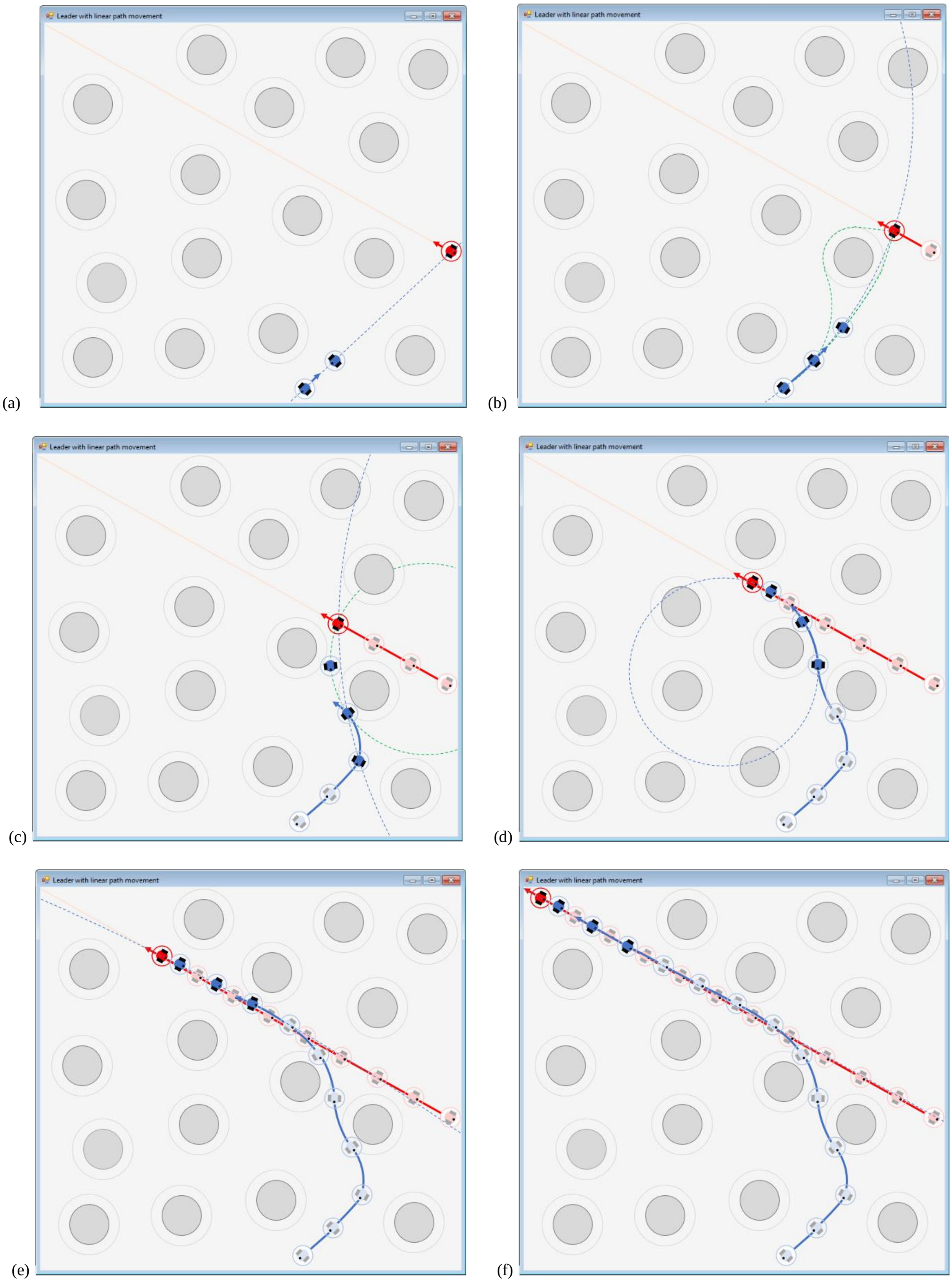



Fig. 23. Leader follower tracking using circular paths algorithm with circular trajectory equal steps at different positions of steps: (a) step = 0; (b) step = 1; (c) step = 3; (d) step = 5; (e) step = 7; (f) step = 8.

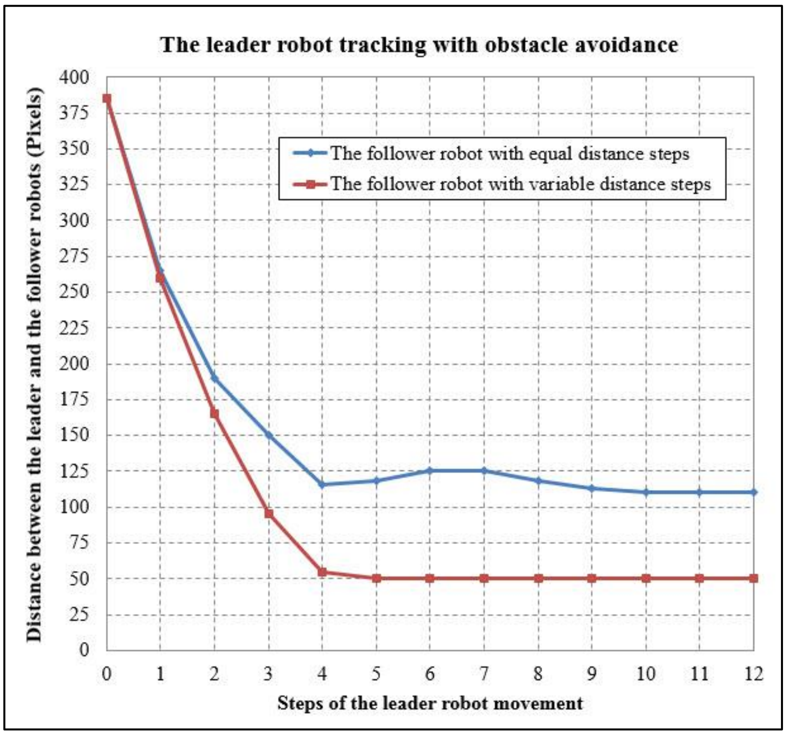



Fig. 24. Illustration of the leader robot tracking with obstacle avoidance on a straight line trajectory when the follower robot move on equal and increment steps.

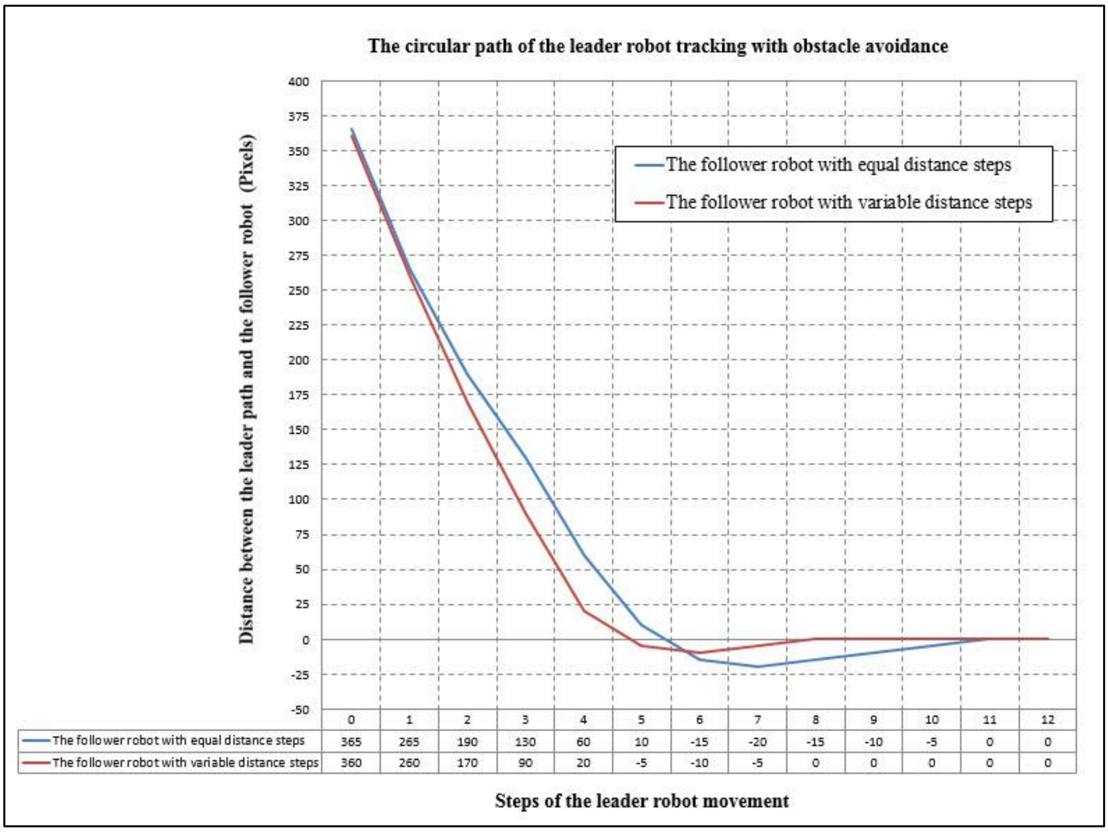



Fig. 25. Illustration of the leader path tracking with obstacle avoidance on a straight line trajectory when the follower robot move on equal and increment steps.

## IV. CONCLUSION

In this paper, the algorithm of circular paths is introduced as a new method to generate a smooth and collision free trajectory using the leader follower tracking mechanism. This algorithm is designed to produce a path consists of several elements each one has a circular trajectory to a void the collisions with obstacles and to give a safe and smooth paths to the follower robot. This algorithm is tested in different types of leader paths with equal and increment step length and in environment has several obstacles. Two type of metrics are tested in these simulations: leader robot tracking and leader path tracking. The simulation results show that all the paths generated by our algorithm are smooth and collision free but the performance of the increment step length is better than using equal step length and for all cases. The two metrics tested in these simulation have better results when the leader robot path is a straight line than the circular trajectory and also have better result for increment step length than the equal step length. Also, in environment has several obstacles we found the trajectory generated by our algorithm is collision free, smooth and has shortest distance when the steps of movement are incremental.

The circular paths algorithm is designed to work in a two-dimensional plane. In future work, it can be extended to work with three-dimensional space. Future work also suggests to extending this algorithm to work with environment has polygon obstacles.

## V. REFERENCES

[1] J. Liu and J. Wu. “Multi-Agent Robotic Systems”, the CRC Press, 2001.

[2] P. Santana, J. Barata, H. Cruz, A.Mestre, J. Lisboa, and L. Flores, “ A multi-robot system for landmine detection” In Emerging Technologies and Factory Automation, 10th IEEE Conference on 2005.

[3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, “Collaborative Multi-Robot Exploration” In Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), 2000.

[4] E. Luis and N. Serment, “ Modularity in small distributed robots” Proceedings of the SPIE conference on Sensor Fusion and Decentralized Control in Robotic Systems II, 1999.

[5] T. Nakatuka, “Teleportation System for Multiple Mobile Robots with Autonomous Collision Avoidance” 36th SICE Symposium on Intelligent system, 2009.

[6] J. Desai, “Controlling formations of multiple mobile robots”, In *Proc. IEEE Int. Conf. Robot. Automat.* pp. 2864-2869, 1998.

[7] R. Fierroet, “Cooperative control of robot formations” Submitted to IEEE/RSJ Int. Conf. Intell. Robots and Syst., IROS 2001, 2002.

[8] T. Ikeda, “Formation Control of Multiple Non holonomic Mobile Robots” , IEEJ Trans.IA, vol. 124, no. 8, pp. 814-819, 2004.

[9] K. Sakurama, “ Leader-following formation navigation for multiple robots with collision avoidance” In Proc. of the 17th IFAC World Congress, Seoul, Korea, 2008.

[10] F. Espinosa, C. Santos, M. M. Romera, D. Pizarro, F. Valdés, J. Dongil, “Laser Scanner Fusion Based on a Discrete Extended Kalman Filter for Robotic Platooning Guidance” Sensors, vol. 11, pp. 8339–8357, 2011.

[11] T. Ahamed, L. Tian, T. Takigawa, and Y. Zhang, “ Development of Auto-Hitching Navigation System for Farm Implements using Laser Range Finder” Trans. ASABE 2009, vol. 52, pp. 1793–1803. 2009.

[12] A. Gou, M. Akira, and N. Noguchi, “Study on a Straight Follower Control Algorithm based on a Laser Scanner” J. Jpn. Soc.Agric. vol. 67, pp. 65–71, 2005.

[13] S. Han, Q. Zhang, B. Ni, and J. F. Reid, “A guidance directrix approach to vision-based vehicle guidance system”, Comput. Electron. Agric. vol. 43, pp. 179–195, 2004.

[14] J. Courbon, Y. Mezouar, N. Guenard, and P. Martinet, “Vision-based navigation of unmanned aerial vehicles”, Control Eng. Pract. vol. 18, pp. 789–799, 2010.

[15] F. Caballero, L. Merino, J. Ferruz, and A. Ollero, “Vision-based odometry and SLAM for medium and high altitude flying UAVs”, J. Intell. Rob. Syst. Theory Appl. vol. 54, pp. 137–161, 2009.

[16] H. Hasegawa, T. Takigawa, M. Koike, A. Yoda, and N. Sakai, “Studies on Visual Recognition of an Agricultural Autonomous Tractor-Detection of the Field State by Image Processing”, Jpn. J. Farm Work Res. vol. 35, pp. 141–147, 2000.

[17] S. K. Kannan, E.N. Johnson, Y. Watanabe, and R. Sattigeri, “Vision-Based Tracking of Uncooperative Targets”, Int. J. Aerosp. Eng., 2011.

[18] T. Krajnik, M. Nitsche, J. Faigl, P. Vanek, M. Saska, L. Preucil, T. Duckett, and M. Mejail, “A practical multirobot localization system”, J. Intell. Rob. Syst. vol. 76, pp. 539–562, 2014.

[19] P. Bhattacharya and M. L. Grvrilova, ”Voronoi diagram in optimal path planning”, in 4th IEEE International Symposium on Voronoi Diagrams in Science and Engineering, pp.38-47, 2007.

[20] S. Mohammadi and N. Hazar, ”A Voronoi-Based Reactive Approach for Mobile Robot Navigation”, Advances in Computer Science and Engineering, Springer Berlin Heidelberg, Vol.6, pp.901-904, 2009.

[21] P. Bhattacharya and M. Gavrilova, “Roadmap-Based Path Planning- Using the Voronoi diagram for a clearance-based shortest path,” IEEE Robotics and Automation Magazine, vol. 15, no. 2, pp. 58–66, 2008.

[22] Y. H. Liu and S. Arimoto, “Proposal of tangent graph and extended tangent graph for path planning of mobile

robots," in Proc. IEEE Inr. Con. Robot. Automat., pp. 312-317, 1991.
[23] Y. Liu and S. Arimoto, "Finding the Shortest Path of a Disc Among Polygonal Obstacles Using a Radius-Independent Graph," IEEE Transactions on Robotics and Automation. vol. 11, pp. 682-691, 1995.
[24] 0. Takahashi and R. J. Schiling, "Motion planning in a plane using generalized Voronoi diagrams," IEEE Trans. Robot. Automat., vol. 6, no. 2, pp. 143-150, 1989.
[24] D. pedoe, "A Mathematical View (Spectrum) Paperback ",The Mathematical Association of America; 2 edition, 1997.
[25] N. A. Court, "College Geometry: An Introduction to the Modern Geometry of the Triangle and the Circle", second edition, Barnes and Noble, 1952; Dover reprint, 2007.