

# Path Planning and Obstacles Avoidance in Dynamic Workspace Using Polygon Shape Tangents Algorithm

Duaa Ahmed Ramadhan\*, Aday Al-Mayyahi, Mofeed Turky Rashid  
Electrical Engineering Department, University of Basrah, Basrah, Iraq

## Correspondence

\* Duaa Ahmed Ramadhan  
Electrical Engineering Department,  
University of Basrah, Basrah, Iraq  
Email: duaa.eng.coe1992@gmail.com

## Abstract

*This paper presents the design of a path planning system in an environment that contains a set of static and dynamic polygon obstacles localized randomly. In this paper, an algorithm so-called (Polygon shape tangents algorithm) is proposed to move a mobile robot from a source point to a destination point with no collision with surrounding obstacles using the visibility binary tree algorithm. The methodology of this algorithm is based on predicting the steps of a robot trajectory from the source to the destination point. The polygon shapes tangent algorithm is compared with the virtual circles' tangents algorithm for different numbers of static and dynamic polygon obstacles for the time of arrival and the length of the path to the target. The obtained result shows that the used algorithm has better performance than the other algorithms and gets less time of arrival and shortest path with free collision.*

**KEYWORDS:** Mobile robot, Path planning, Polygon obstacle, Obstacle avoidance.

## I. INTRODUCTION

Path planning is defined as a way of moving a robot from the starting point to the ending point with no collision, less time arrival, and shortest path [1]. Path planning has a wide number of applications in computer games, logistics robotics [2-3], family services, and military applications [4-6]. Also, path planning of mobile robots is one of the most challenges in robotics because it is applied in different applications such as routing transportation, organizing allocation of machines in factories, controlling robots, intelligent agents [7]. There are many types of the algorithm used to construct path planning system. Also, path planning algorithms can be classified into two types: static and dynamic path planning algorithms according to the environment in which the robot will navigate. In dynamic path planning, the robot uses information that records it from its sensors to detect the current position and direction of moving objects in the environment. There are a wide number of algorithms used to solve this problem of dynamic path planning and these algorithms can be classified according to their respective issues such as path safety, path length, smooth path, runtime, accuracy, stability, control, future prediction (uncertainty), and computation cost. Also, path planning algorithms are classified into two types: online and offline path planning. On-line path planning is proper for a partially known environment with a static target since the robot uses its sensors to collect information about the environment but not

all can be solved moving target problem [8-10]. Also, online path planning depends on the current information of an environment to select the current optimal path [11]. Off-line path planning is not proper in the large dynamic environment and moving target problem handling because it consumes time since the complete path from starting point to the ending point is computed at the start position [12-13]. Algorithms for off-line path planning are converted to incremental algorithms to be more efficient. Incremental algorithms are suitable for a partially known environment but do not solve moving target problems. Section II of this paper describes the path planning algorithm with static obstacle avoidance. Section III describes the path planning algorithm with dynamic obstacle avoidance. Section IV describes the simulation results. Section V describes the conclusion.

## II. PATH PLANNING ALGORITHM WITH STATIC OBSTACLE AVOIDANCE

In this section, a path planning system will be designed in an environment contains a set of static polygon obstacles by using a differential drive mobile robot. The method of the algorithm is based on obstacle avoidance by drawing paths in contact with obstacles to configure a tree of paths using a visibility binary algorithm [14]. The shortest path is chosen of this tree for moving a robot to the target point.



This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.

© 2021 The Authors. Published by Iraqi Journal for Electrical and Electronic Engineering by College of Engineering, University of Basrah.

A. Compute the centroid of each obstacle

In this section, it will represent every obstacle with a circle contains all of the polygon obstacle heads and then find a center of this circle to explore if this obstacle is in the way of a robot or not. For this purpose, the algorithm is chosen is the minimum bounding circle algorithm [15-16], this algorithm depends on using Chan's algorithm in Fig.1 and Fig.2.

```

Chan's Algorithm (p)
1. For t=1; 2; ...
2. Let m=min(2^(2^t), n)
3. Let r=ceil(n/m)
4. Divide stage
5. For k=1 to r d
6. For i=1 to m do
7. Compute sub hull P (m) using Graham's scan and store the
   vertices in an ordered array S in CCW oriented.
8. End
9. Jarvis stage
10. Find leftmost vertex S0 from all sub hulls that resulted from
   Graham scan algorithm
11. Compute final convex hull ch (h) using Jarvis March algorithm
12. End
    
```

Fig. 1: Chan's algorithm.

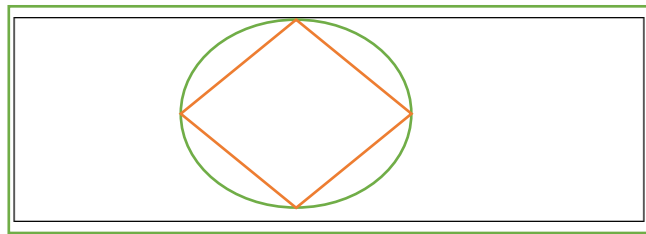


Fig. 2: Minimum circle bounding the convex polygon.

B. Binary tree paths construction algorithm

In this section, it will construct a tree of paths for a robot according to the following steps:

Step 1: Direct path to the target: In this step, it will draw a direct path for moving the robot from the source point to the destination point as in Fig. 3.

Step 2: Testing the collision with an obstacle: In this step, it will test an intersection of one of the obstacles with the path that drawing from the robot to the target. At first, it will determine the distance between every obstacle and path that connects between robot and target using the following relations [14] where a is the slope, b is constant and h is the distance between a point on an obstacle and the direct path connect the robot with a target as in Fig. 4.

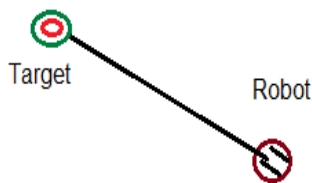


Fig. 3: Direct path.

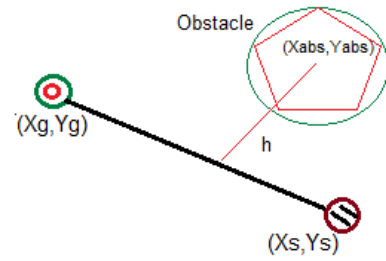


Fig. 4: Detection the collision with the obstacle.

$$a = \frac{(y_g - y_s)}{(x_g - x_s)} \quad (1)$$

$$b = -1 \quad (2)$$

$$c = y_s - a * x_s \quad (3)$$

$$h = \frac{|a * x_{abs} + b * y_{abs} + c|}{\sqrt{a^2 + b^2}} \quad (4)$$

Step 3: Choose the nearest collision: In this step, the obstacle that having a minimum value of distance about robot will be the first obstacle that intercepts the path of the robot so choose obstacles that far away from the path that connects between robot and target with the shortest distance from the radius of the circle that surrounding obstacle.

The following relation determined the distance between obstacle and robot:

$$Dis = \sqrt{(x_s - x_{abs})^2 + (y_s - y_{abs})^2} \quad (5)$$

Step 4: Estimate the outer and inner tangent points: In this step, deflect the direction of the robot to prevent collision between it and the obstacle, it will determine the outer and inner tangent points for tangent contacts from the robot to the obstacle. Estimation of tangent points by the following steps:

1. Determine the distance between every head in polygon obstacle and the direct path from robot to target as in Fig. 5.
2. Vertices of the polygon obstacle that has the largest distance for the path between robot and target represent tangent points.

Step 5: Draw the outer and inner tangent lines: In this step, it will draw lines that contact with obstacle from the robot as in Fig. 6.

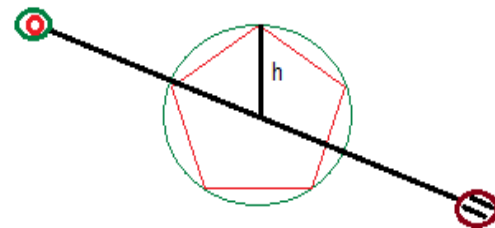


Fig. 5: Estimate the two tangent points.

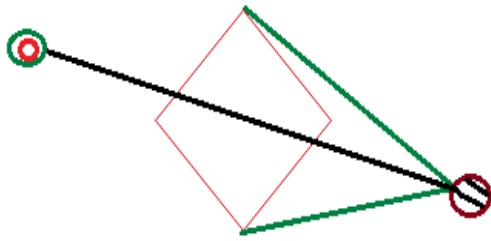


Fig. 6: The two tangent lines.

**C. Optimal paths estimation**

In this part, using an algorithm called visibility binary search algorithm [14] as in Fig.8 to choose the shortest path from a set of paths drawn from robot to target.

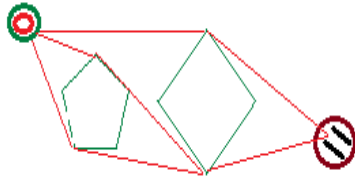


Fig. 7: Complete binary tree construction.

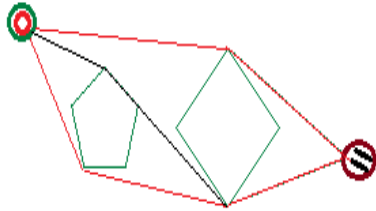


Fig.8: Optimal path from robot to the target.

**III. PATH PLANNING ALGORITHM WITH DYNAMIC OBSTACLE AVOIDANCE**

This section of research will design path planning system in an environment contains some peoples work as dynamic obstacles. The method of the algorithm is based on predicting the steps of the robot trajectory from source to target point. Each person step (dynamic obstacle) is computed as well as it is the direction to estimate the intersection point with the line drawing from the source to the target points. This process helps in estimate the number of steps that people required to reach the intersection point [18-20]. If the robot position (chosen from the steps of the robot trajectory) is equal to the people position that means the collision occurs. This problem is solved by delay the movement of the robot one step by repeating the first position of the robot in the array of the robot trajectory steps. The following steps show the procedure used to avoid the collision with the dynamic obstacles.

**A. Compute the step length of each dynamic obstacle**

In this part of the procedures, the location of each dynamic obstacle in two successive locations is computed as shown in Fig. 9. The step length L in Pixel is computed by using the following equation:

$$L = \sqrt{(x_{ob}^2 - x_{ob}^1)^2 + (y_{ob}^2 - y_{ob}^1)^2} \quad (6)$$

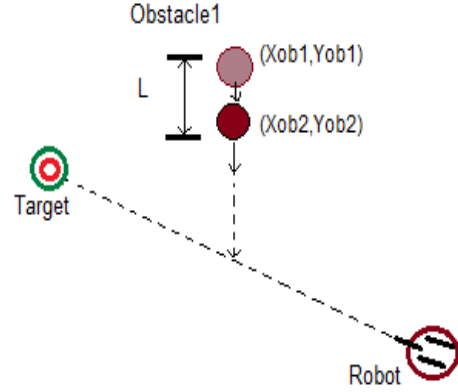


Fig. 9: Computing step length of each dynamic obstacle.

**B. Compute the step length of the robot**

In this part of the procedures, the location of the robot in two successive locations is computed as shown in Fig. 10. The step length D in Pixel is computed by using the following equation:

$$D = \sqrt{(x_R^2 - x_R^1)^2 + (y_R^2 - y_R^1)^2} \quad (7)$$

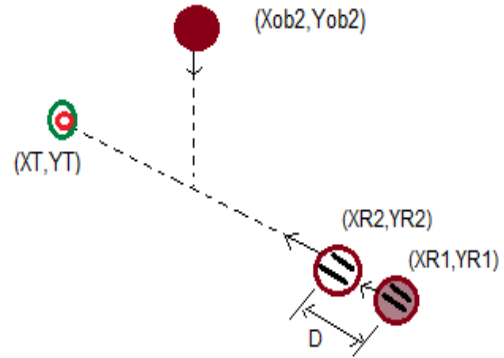


Fig. 10: Computing step length of the robot.

**C. Compute the intersection point for the obstacle movement path with the robot path**

In this part of the procedures, the intersection point (x1, y1) as shown in Fig. 11 is computed according to the following equations:

$$x_1 = \frac{(x_R^1 y_R^2 - y_R^1 x_R^2)(x_{ob}^1 - x_{ob}^2) - (x_R^1 - x_R^2)(x_{ob}^1 y_{ob}^2 - y_{ob}^1 x_{ob}^2)}{(x_R^1 - x_R^2)(y_{ob}^1 - y_{ob}^2) - (y_R^1 - y_R^2)(x_{ob}^1 - x_{ob}^2)} \quad (8)$$

$$y_1 = \frac{(x_R^1 y_R^2 - y_R^1 x_R^2)(y_{ob}^1 - y_{ob}^2) - (y_R^1 - y_R^2)(x_{ob}^1 y_{ob}^2 - y_{ob}^1 x_{ob}^2)}{(x_R^1 - x_R^2)(y_{ob}^1 - y_{ob}^2) - (y_R^1 - y_R^2)(x_{ob}^1 - x_{ob}^2)} \quad (9)$$

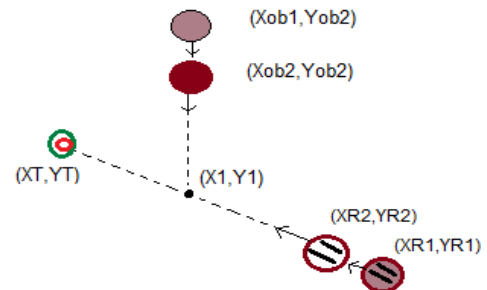


Fig. 11: Computing the intersection point.

#### D. Estimate the number of the steps for the dynamic obstacle to reaching the intersection point

In this part of the procedures, the number of steps to reach the intersection point is computed according to the following steps:

Step1: Compute distance  $L_{ob1}$  between the dynamic obstacles  $i$  and the intersection point (Fig. 12) by using the following equation:

$$L_{ob1} = \sqrt{(x_1 - x_{ob2})^2 + (y_1 - y_{ob2})^2} \quad (10)$$

Where  $L_{ob1}$  is the distance between the dynamic obstacle 1 and the intersection point.

Step 2: The number of steps  $obs_{step}$  that are required to reach the intersection point is computed using the following equation:

$$obs_{step} = \frac{L_{ob1}}{L} \quad (11)$$

Where  $L$  is the step length in pixels.

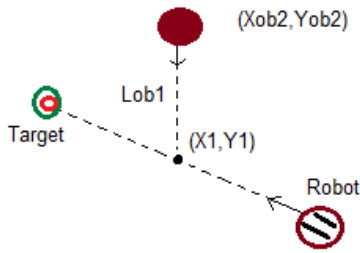


Fig. 12: Computing the distance between the dynamic obstacle and intersection point.

#### E. Estimate the position of the steps of the robot movement

In this part of the procedures, the position of the step of the robot from the source point to the target point is estimated according to the following steps:

Step 1: Compute the orientation of the direct path from robot location to the target using the following equation:

$$\phi = \tan^{-1} (y_T - y_R^2) / (x_T - x_R^2) \quad (12)$$

Step 2: For the first position  $i=1$  to the last position of robot movement, estimate the coordinate axis of the robot using the following equations:

$$x_R^{i+1} = x_R^i + D \cos \phi \quad (13)$$

$$y_R^{i+1} = y_R^i + D \sin \phi \quad (14)$$

#### F. Computing the free collision trajectory to the robot

In this part of the procedure, the free collision trajectory is estimated by applying a delay to the starting movement of the robot. This process is achieved by using the following steps:

Step 1: Starting from the nearest intersection point to the robot location choose the robot location  $i$  which has the same number of steps movement of the dynamic obstacle. Compute the distance  $D_{stR}$  between the intersection point and the robot location using the following equation:

$$D_{stR} = \sqrt{(x_1 - x_{Ri})^2 + (y_1 - y_{Ri})^2} \quad (15)$$

Step 2: If the distance is less than the robot diameter then applying one-step delay by repeats the first robot location and repeat the process from the first step.

Step 3: Repeat steps 1 and 2 until the robot reaches the target point. The resulting array of robot locations represents the free collision robot trajectory.

## IV. SIMULATION RESULTS

The new algorithm (Polygon shape tangent algorithm) is simulated to investigate the path planning of mobile robots using visual basic programming language and tested in a workspace environment using an Intel core i5. This algorithm is applied in an environment that contains static polygon obstacles and dynamic obstacles such as peoples distributed randomly. Fig. 13 (a-f) represents the screenshots of the simulation at different time steps for the polygon shapes tangent algorithm. Fig. 14 (a-f) represents the screenshots of the simulation at different time steps for the virtual circles tangent algorithm. Fig. 15 represents path planning with different numbers of static and dynamic obstacles using polygon shapes tangent algorithm where (a-c) represents two static obstacles and (2,4,6) dynamic obstacles while (d-f) represents four static obstacles and (2,4,6) dynamic obstacles. Fig. 16 represents the path planning with different numbers of static and dynamic obstacles using the virtual circles tangent algorithm where (a-c) represent two static obstacles and (2,4,6) dynamic obstacles while (d-f) represent four static obstacle and (2,4,6) dynamic obstacles. The main goal of this simulation is to show the relation between the number of obstacles and the time of arrival and the length of the path from the source point to the destination point with no collision.

Fig. 17 and Fig. 18 show the comparison between the number of static and dynamic obstacles and the time of arrival between the virtual circles tangent algorithm and the polygon shape tangent algorithm. As the number of obstacles increases in the environment, the time of arrival increase for both algorithms. Fig. 19 and Fig. 20 show the second comparison between the number of static and dynamic obstacles and the length of the path to the target between virtual circles tangent algorithm and polygon shape tangent algorithm. As the number of obstacles increases in the environment, the length of the path for both algorithms is also increased. The polygon shape tangent algorithm is produced the best performance than the virtual circles tangent algorithm.

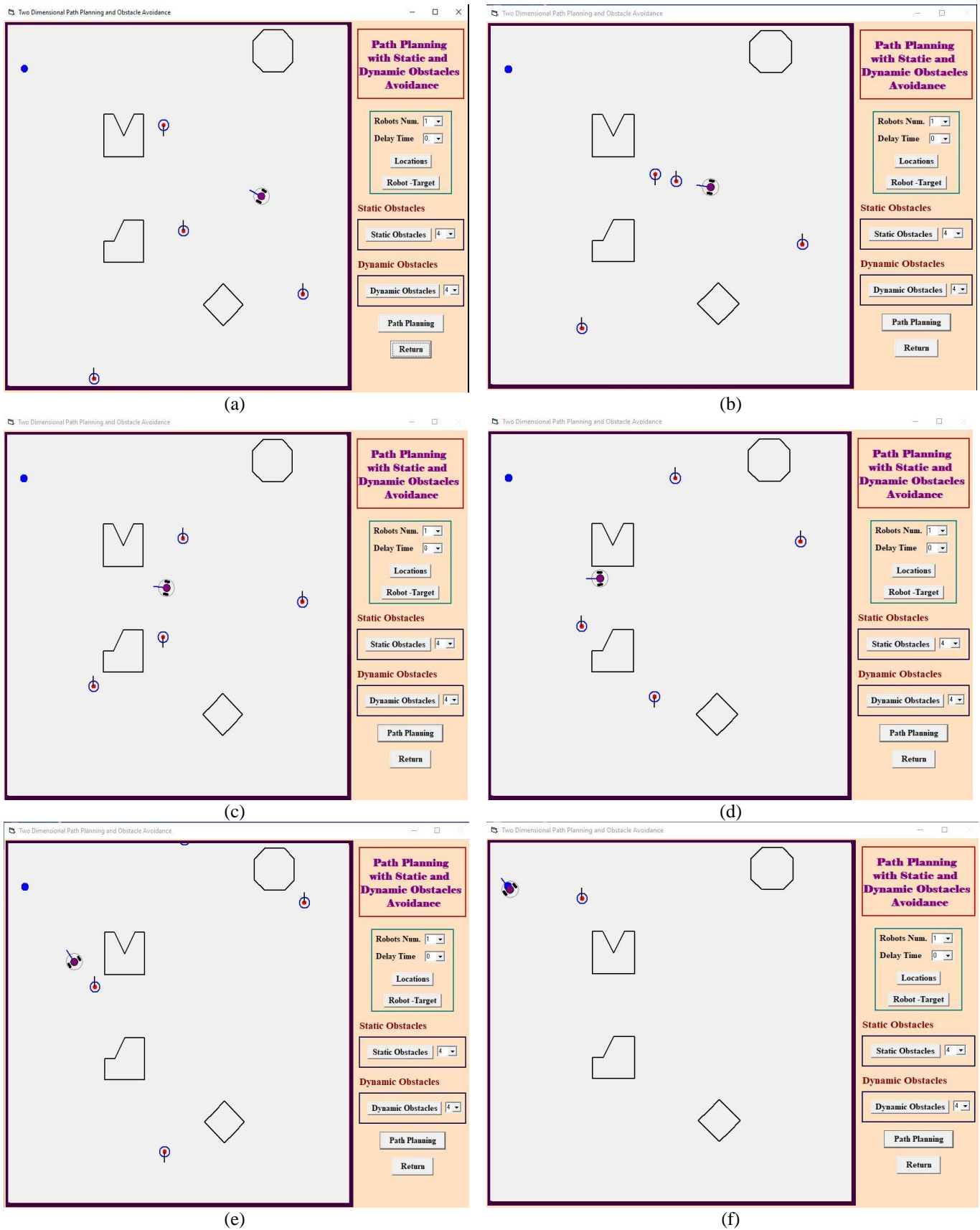


Fig.13: The polygon shape tangent algorithm. (a-f) Screenshots at different time-step trajectory planning.



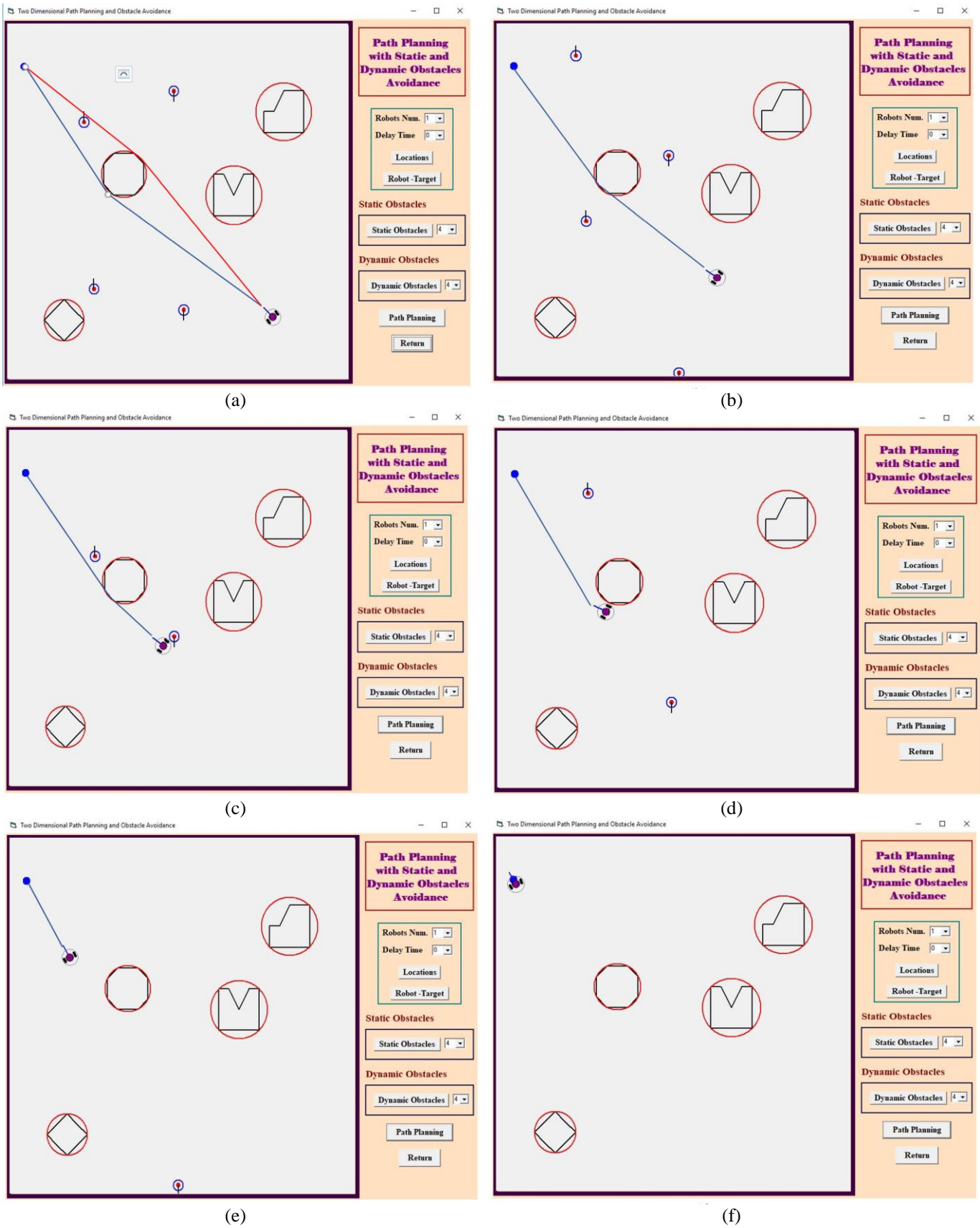


Fig.14: The virtual circles tangent algorithm. (a-f) Screenshots at different time-step trajectory planning.

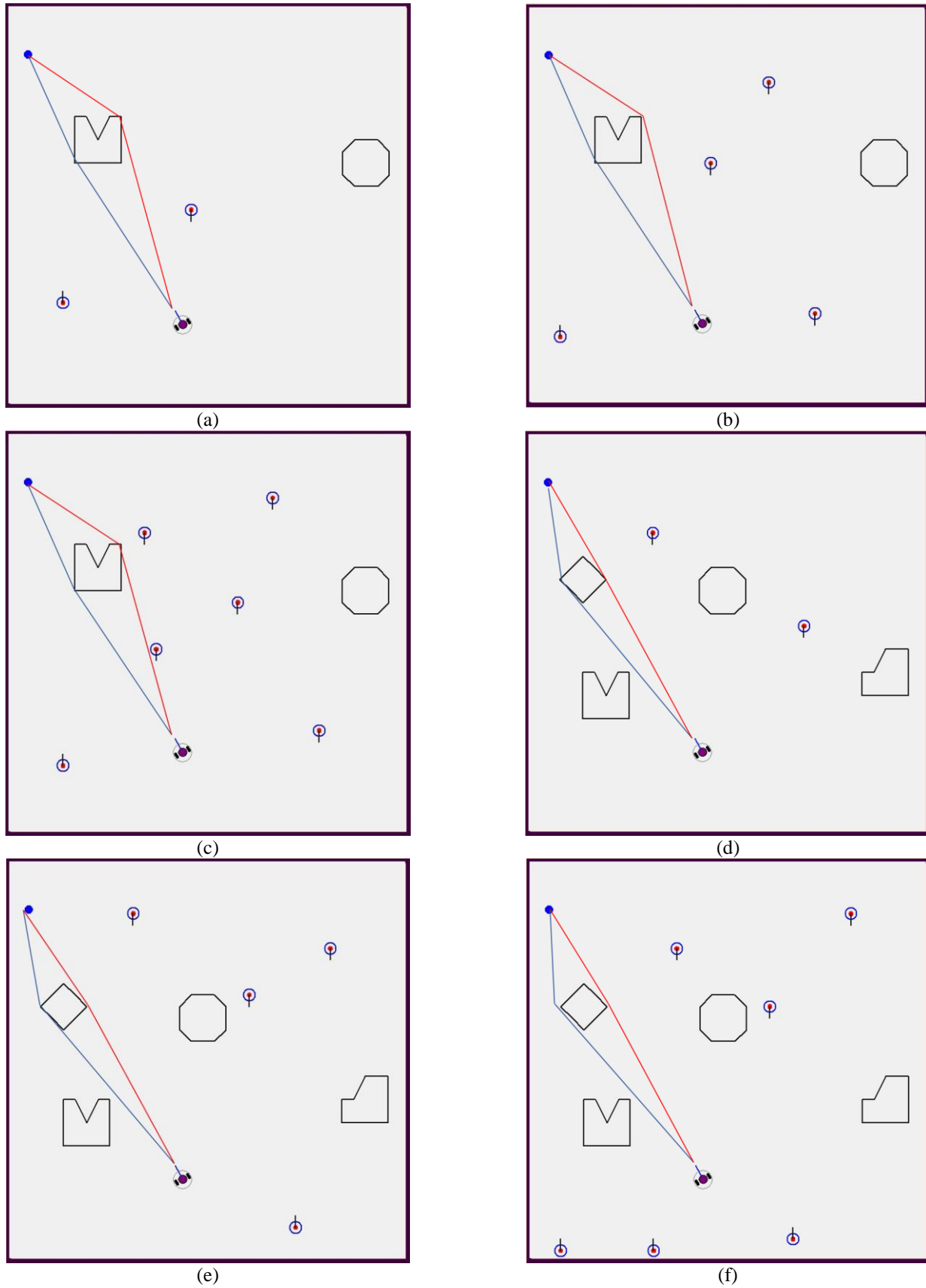


Fig.15: The polygon shape tangent algorithm with different numbers of static and dynamic obstacles.

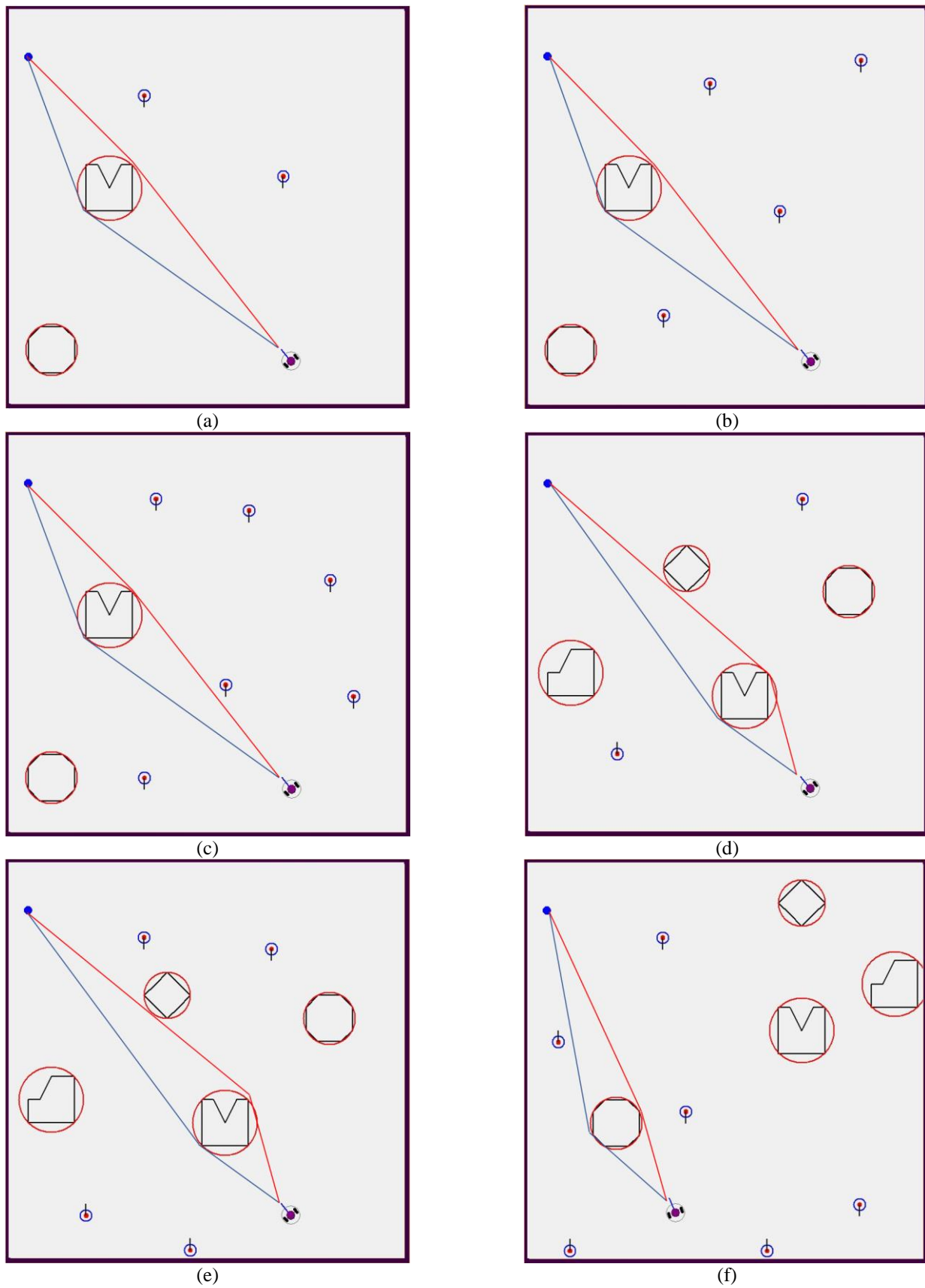


Fig.16: The virtual circle tangent algorithm with different numbers of static and dynamic obstacles.



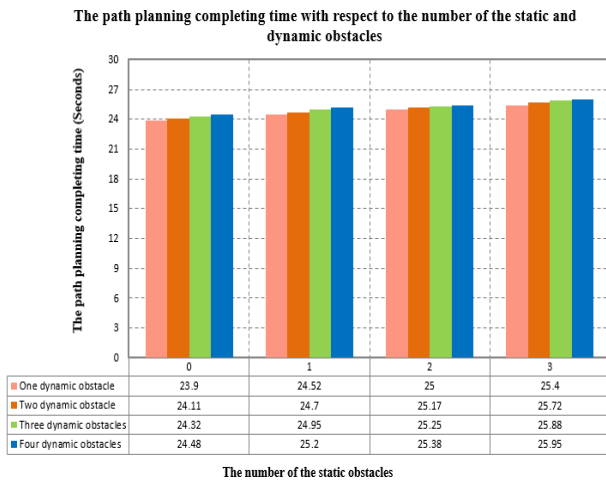


Fig. 17: Comparison of the time of arrival for a different number of static and dynamic obstacles using the virtual circles tangent algorithm.

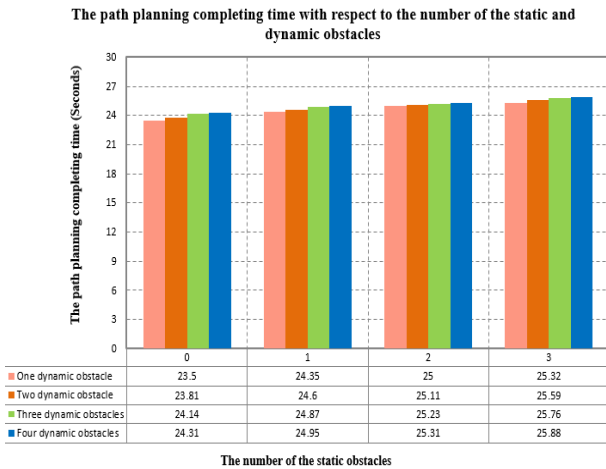


Fig. 18: Comparison of the time of arrival for different numbers of static and dynamic obstacles using the polygon shape tangent algorithm.

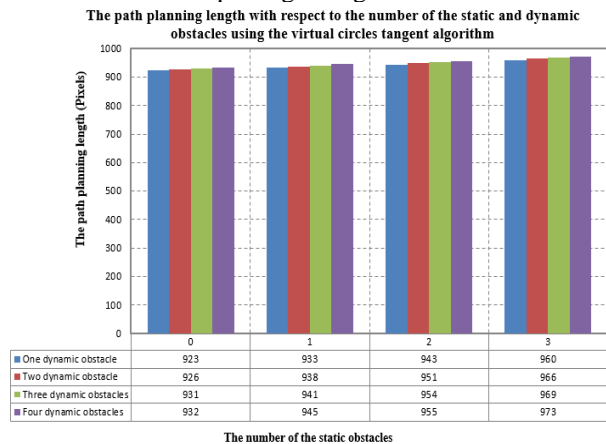


Fig. 19: Comparison of the length of the path for a different number of static and dynamic obstacles using the virtual circles tangent algorithm.

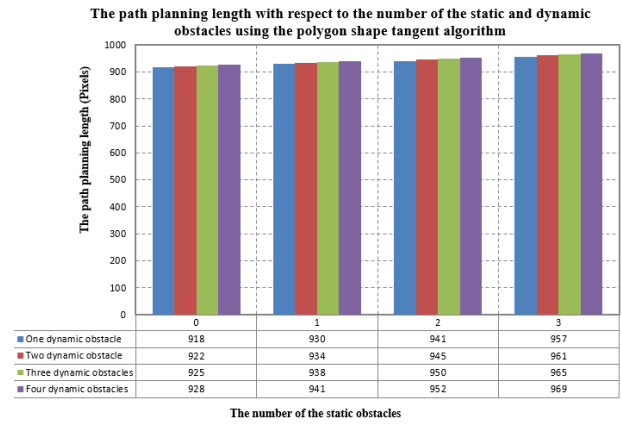


Fig. 20 Comparison of the length of the path for a different number of static and dynamic obstacles using the polygon shape tangent algorithm.

### V. CONCLUSIONS

In this paper, a two-dimensional path planning system is designed in an environment that contains several numbers of static polygon obstacles and several dynamic polygon obstacles such as peoples localized and distributed randomly. This system is designed to get an algorithm that has good performance for using it to help the robot to move from source point to destination point such as in factories of amazon where each robot in these factories is considered a dynamic obstacle to the other robot. The method of this algorithm depends on predicting the step of each dynamic obstacle. This system determines the steps of the robot and the steps of obstacles, if the number of the steps of the robot is equal to the number of the steps of an obstacle then it will lead to a collision. So this system is designed to help the robot to reach to target with less time of arrival and with free collision. Polygon shapes tangent algorithm and virtual circles tangent algorithm are used with different time steps trajectory planning. Also used polygon shapes tangent algorithm and virtual circles tangent algorithm with different numbers of static and dynamic obstacles and record results for time of arrival and length of the path to the target and gets when the number of obstacles increases, time of arrival is also increased and length of the path to the target is increased too.

### FUTURE WORK

1. The hardware system implementation can be suggested for future work because the introduced algorithm was implemented in simulation.
2. The algorithm in this paper is implemented in a two-dimensional environment, so the extension of this algorithm to three-dimensional can be suggested for future work.

### ACKNOWLEDGMENT

I would like to introduce my deep gratefulness to the soul of my previous supervisor Assist. Prof. Dr. Abdulmutallab Turkey Rashid who with deep sadness died due to Covid-19. I sincerely appreciate his great efforts that helped me to reach this stage of my master project.

### CONFLICT OF INTEREST

The authors have no conflict of relevant interest to this article.

### REFERENCES

- [1] Z. Y. Ibrahim, A. T. Rashid, and A. F. Marhoon, "Prediction-Based Path Planning with Obstacle Avoidance in Dynamic Target Environment," *Basrah J. Eng. Sci.*, vol. 16, no. 2, pp. 48–60, 2016.
- [2] B. Bonet and H. Geffner, "Planning as heuristic search," *Artif. Intell.*, vol. 129, no. 1–2, pp. 5–33, 2001.
- [3] J. van den Berg, R. Shah, A. Huang, and K. Goldberg, "ANA\*: Anytime Nonparametric A\*," in *Proceedings of the Twenty-Fifth AAAI Conference on Artificial Intelligence*, 2011, vol. 25, no. 1, pp. 105–111.
- [4] P.-Y. P. Ting-Kai Wang, Quan Dang, "Path Planning Approach in Unknown Environment," *Int. J. Autom. Comput.*, vol. 7, no. 3, pp. 310–316, 2010.
- [5] Enric Galceran and M. Carreras, "A survey on coverage path planning for robotics," *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1258–1276, 2013.
- [6] M. K. JHA, G. A. KARRI, and M.-W. KANG, "A military path planning algorithm using visualization and dynamic GIS," in *Proceedings of the 4th WSEAS international conference on Computer engineering and applications*, Cambridge, Stevens Point, Wisconsin, 2010, pp. 188–193.
- [7] K. F. T. SH, K. W, Z. N, and A. SA, "A Review on Motion Planning and Obstacle Avoidance Approaches in Dynamic Environments," *Adv. Robot. Autom.*, vol. 4, no. 2, 2015.
- [8] A. A. Ali, A. T. Rashid, M. Frasca, and L. Fortuna, "An algorithm for multi-robot collision-free navigation based on shortest distance," *Robotics and Autonomous Systems*, vol. 75, Part B, pp. 119–128, 2016.
- [9] Z. Y. Ibrahim, A. T. Rashid, and A. F. Marhoon, "Path planning algorithm for mobile robot navigation in a dynamic environment based on motion prediction and tangency graph," *IEEE First International Conference on Recent Trends of Engineering Science and Sustainability*, January 2018.
- [10] C. Undeger and F. Polat, "Real-time edge follow A real-time path search approach," *IEEE Trans. Syst. Man, Cybern. Part C (Applications Rev.)*, vol. 37, no. 5, pp. 860–872, 2007.
- [11] M. S. Alam, M. U. Rafique, and M. U. Khan, "Mobile Robot Path Planning in Static Environments using Particle Swarm Optimization," *Int. J. Comput. Sci. Electron. Eng.*, vol. 3, no. 3, pp. 253–257, 2015.
- [12] S. Koenig and M. Likhachev, "Fast replanning for navigation in unknown terrain," *IEEE Trans. Robot.*, vol. 21, no. 3, pp. 354–363, 2005.
- [13] A. Stentz, "The Focussed D\* Algorithm for Real-Time Replanning," *Int. Jt. Conf. Artif. Intell.*, vol. 95, pp. 1652–1659, 1995.
- [14] A. T. Rashid, A. A. Ali, M. Frasca, and L. Fortuna, "Path planning with obstacle avoidance based on visibility binary tree algorithm," *Rob. Auton. Syst.*, vol. 61, no. 12, pp. 1440–1449, 2013.
- [15] [Z. Y. Ibrahim, A. T. Rashid, and A. F. Marhoon, "An algorithm for Path planning with polygon obstacles avoidance based on the virtual circle tangents," *Iraqi J. Electr. Electron. Eng.*, vol. 12, no. 2, 2016.
- [16] I. S. A. Al-Forati and A. Rashid, "Design and Implementation an Indoor Robot Localization System Using Minimum Bounded Circle Algorithm," in *2019 8th International Conference on Modeling Simulation and Applied Optimization (ICMSAO)*, 2019, pp. 1–6.
- [17] A. T. Rashid, A. A. Ali, M. Frasca, and L. Fortuna, "Multi-robot collision-free navigation based on reciprocal orientation," *Rob. Auton. Syst.*, vol. 60, no. 10, pp. 1221–230, 2012.
- [18] A. A. Aldair and A. Al-Mayyahi, "Maze Maneuvering and Colored Object Tracking for Differential Drive Mobile Robot," *Iraqi J. Electr. Electron. Eng.*, vol. 15, no. 1, pp. 47–52, 2019.
- [19] B. N. Abdul Samed, A. A. Aldair, and A. Al-Mayyahi, "Robust Trajectory Tracking Control and Obstacles Avoidance Algorithm for Quadrotor Unmanned Aerial Vehicle," *J. Electr. Eng. Technol.*, vol. 15, no. 2, pp. 855–868, 2020.
- [20] A. Al-Mayyahi and W. Wang, "Fuzzy inference approach for autonomous ground vehicle navigation in a dynamic environment," in *2014 IEEE International Conference on Control System, Computing and Engineering (ICCSCE 2014)*, 2014, pp. 29–34.