ENHANCED HYBRID PATH PLANNING ALGORITHM BASED ON APF AND A-STAR

Ahmed S. Abdel-Rahman[1], Shady Zahran[2], Basem E. Elnaghi[1] , S.F. Nafea[1]

1 Dept. of Electrical Engineering, Suez Canal University, Ismailia, Egypt – (ahmed.Soliman, basem_elhady sheriff_kamel)@eng.suez.edu.eg
2 Computer Science, College of Computing & Information Technology, AAST Cairo Egypt- shady.zahran1@adj.aast.edu

KEYWORDS: Path-Planning, Obstacle Avoidance, Mobile Robot, Navigation, APF, A-Star.

ABSTRACT:

The use of robots had increased widely in various fields, such as air transport systems, search and rescue, and agriculture, necessitating the need for path planning and obstacle avoidance systems to ensure safe and autonomous navigation toward the intended goal. Many Path-planning techniques are used to guide the mobile robot toward its goal with an optimized path and time. Among these techniques, the Artificial Potential Field (APF) algorithm stands out as one of the effective approaches, capable of operating in both static and dynamic environments to achieve optimal path planning. The APF is built by fusing two forces that attract the robot toward a goal location and repulsive forces that repel the robot away from the obstacle. Despite its effectiveness, the APF algorithm faces a major challenge, which is falling into a local-minimum issue. This paper presents a hybrid approach that combines the modified APF algorithm global optimization capabilities with the A-Star path-planning technique's real-time adaptability to overcome traditional APF local minimum issues. The transition back and forth between the two algorithms where carried out by a manager that can determine the adequate algorithm to be used instantaneously. Several experiments are presented to demonstrate the hybrid algorithm's effectiveness in various environments. The results show that the enhanced APF reach an optimal path to goal 50% faster compared to A-star, and managed to get out of local minimum compared to traditional APF and find a path shorter than A-star.

1. INTRODUCTION

Path planning is a fundamental problem in robotics. Path planning aims to find a safe and efficient path for a robot to move from one point to another in an environment that may contain obstacles. Two of the most well-known path-planning algorithms are the A* algorithm and the artificial potential field (APF) algorithm. A* algorithm is a pathfinding algorithm that is used to find the shortest path between two points on a graph. It was first described by Peter Hart, Nils Nilsson, and Bertram Raphael of Stanford Research Institute (now SRI International) in 1968(Hart et al., 1968). The algorithm uses a heuristic function to estimate the cost of the cheapest path from the start node to the goal node. The heuristic function is an estimate of the remaining distance between the current node and the goal node. The algorithm then expands the node with the lowest estimated cost, until it reaches the goal node. A* algorithm is widely used in various domains for pathfinding and optimization problems. It has applications in robotics, video games, route planning, logistics, and artificial intelligence. In robotics, A* helps robots navigate obstacles and find optimal paths. In video games, A* is used to find the shortest path between two points on a map("A* search algorithm - Wikipedia," n.d.). In route planning, A* is used to find the shortest path between two points on a map. In logistics, A* is used to optimize delivery routes. The A* algorithm can solve very complex problems as it can find the shortest path through the search space using the heuristic function, this search algorithm expands fewer search trees and provides optimal results faster. The main drawback of the A* algorithm is memory requirement as it keeps all generated nodes in the memory, so it is not practical for various large-scale problems. However, the APF algorithm is a path-planning algorithm used in robotics that was first introduced by Oussama Khatib in 1986(Khatib, 1985). The algorithm is based on the idea of creating a virtual potential field around obstacles and guiding the robot toward the goal by minimizing the potential energy. The artificial potential field algorithm has been examined by several researchers for various applications such as mobile robots, wheelchairs(Matsuura et al., 2022), underwater vehicles(Cheng et al., 2015; Zhu et al., 2021), humanoid robots(Zhong et al., 2015), walking robots(Igarashi and Kakikura, 2004), planetary rovers(Sancho-Pradel and Saaj, 2014), autonomous sailboats(Plumet et al., 2013), and bio-fish(Iswanto et al., 2019). The artificial potential field algorithm has some advantages such as it is simple to implement and computationally efficient. It is also robust to sensor noise and can handle dynamic environments. The artificial potential field algorithm has some limitations such as it can get stuck in local minima and maxima. Also, it is difficult to tune the parameters for complex environments(Koren et al., n.d.). The fusion of A* and APF algorithms aims to combine the strengths of both algorithms. The fusion of A* and APF algorithms has been a topic of research in robotics for many years. Several different methods have been proposed, and the performance of these methods has been evaluated. There are several different ways to fuse A* and APF algorithms. (Ju et al., 2020) proposed a path-planning algorithm that fuses A* and APF algorithms. Their algorithm first uses A* to generate an initial path, which is then refined by the APF algorithm. They showed that their algorithm can generate shorter paths and higher success rates than A* or APF alone. This can lead to more efficient and robust path-planning algorithms. (Chen et al., 2021) propose a three-neighbor search A* algorithm combined with artificial potential fields to optimize the path-planning problem of mobile robots. The algorithm integrates and improves the partial artificial potential field and the A* algorithm to address irregular obstacles in the forward direction. The artificial potential field guides the mobile robot to move forward quickly. The A* algorithm of the three-neighbor search method performs accurate obstacle avoidance. The current pose vector of the mobile robot is constructed during obstacle avoidance, the search range is narrowed to less than three neighbors, and repeated searches are avoided. To evaluate their algorithm in the matrix laboratory environment, grid maps with different obstacle ratios are compared with the A* algorithm. (Liu et al., 2022) proposed a fusion algorithm that uses A* to find an initial path and then uses APF to smooth the path. The paper also proposed a method for incorporating obstacle avoidance into the path-planning algorithm. This paper proposes a novel method for fusing A* and APF algorithms. Our method uses a combination of A* and APF to find a safe and efficient path for a robot to move from one point to another in an environment that may contain obstacles. We evaluate the performance of our method using several simulation scenarios that incorporate different obstacles.

2. METHODOLOGY

The focus of this paper is to address the main limitation of the traditional Artificial Potential Field (APF) algorithm in path planning, which is the tendency to fall into a local minimum and fail to reach the intended goal. To overcome this challenge, an enhanced APF approach incorporates the Modified APF algorithm, the Fusion Manager Algorithm, and the Modified A-Star algorithm to achieve optimized path planning.

The modification sequence of the Artificial Potential Field (APF) starts with Fusion Manager which serves as the central coordinating component of the system. That contains the start point and the goal point. Fusion Manager starts by sending the start point and the goal point to the modified APF. The modified APF algorithm returns two types of responses to the Fusion Manager. The first case: is when the algorithm reaches its aim, and returns an optimized full path to the goal. In the second case if it encountered a local-minimum point, so, the modified APF returned to the Fusion Manager with the local-minimum point and the optimized path to this local-minimum point. The Fusion Manager in this 2nd case triggers the A-Star algorithm which starts searching for the goal or the point just outside the local-minimum border. The A-Star algorithm returns to Fusion Manager with the goal or a point on the local-minimum border whichever is nearest. In case the A-Star reaches the goal point, the Fusion Manager concatenates the path returned from the modified APF with the path returned from A-Star. In case the A-Star reaches the local-minimum border, The Fusion Manager calls the modified APF again to start from the local-minimum border point as a starting point to reach the goal.

2.1 Modified APF

This section illustrates the modification in APF, which used traditional APF with three sequential steps needed for each point in the path:

1- Check if the current point equals the Goal point.
2- Check oscillation.
3- Check Local-minimum.

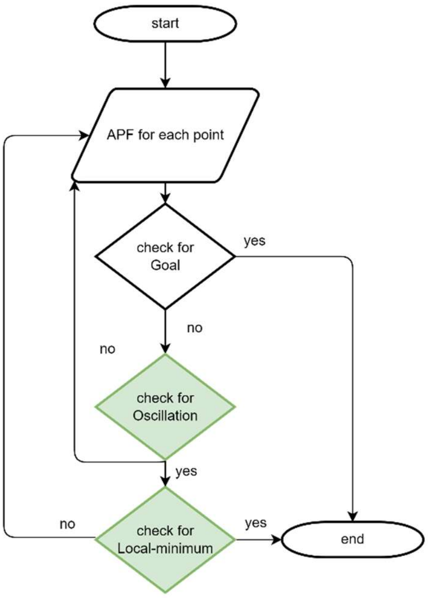

Figure 1. APF Modification Flowchart

Fig 1 illustrates the APF Modification Flowchart. According to the executed experiments, it's observed that when the local minimum occurs, there are 2 criteria happens, oscillation and Go back steps (moving in a direction opposite to the intended goal). As shown in Fig 2 the local minimum happens after the oscillation as a first step then go back. That's why there are 2 steps that were added to the traditional APF.

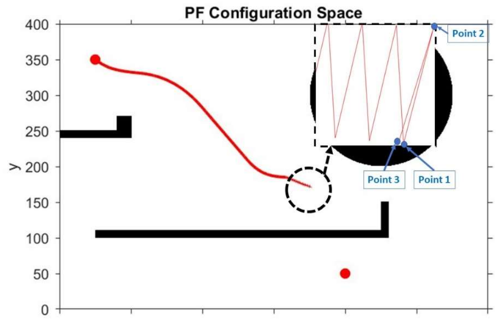

Figure 2. Local-minimum Detection

The oscillation check algorithm is based on the Trigonometry calculation. This step is used to detect the behavior of the APF algorithm in case there is oscillation by drawing a triangle using the last 3 points.

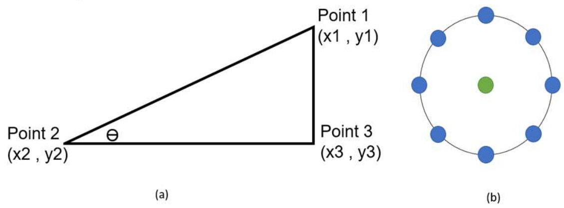

Figure 3. a) Last 3 Points in Path b) 8 points surrounding the current point to get the mean of the field.

This triangle is based on the last 3 points on the Path as shown in Fig 3. a) First, calculate the length of each side of this triangle using Euclidean Distance eq. (1).

$$length = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad (1)$$

where $length$ = distance between two points
x, y = image coordinates

Algorithm 1 Check Oscillation	
1:	Initialize oscillation parameter with a false value Then start calc all triangle sides length
2:	a = sqrt((x3 - x1)^2+ (y3 - y1)^2)
3:	b = sqrt((x2 - x1)^2+ (y2 - y1)^2)
4:	c = sqrt((x3-x2)^2+ (y2- y3)^2)
5:	ϴ = cos -1 ((b^2 + c^2 - a^2)/(2*b*c))
6:	IF (ϴ < 35) then
7:	Oscillation = True
8:	End if

After calculating the length of three sides, start calculating Theta angle with eq. (2)

$$b^2 = a^2 + c^2 - 2ac * \cos(B) \quad (2)$$

Where a, b, c = triangle sides
B = angle between side a and side c

The angle value between the 2 lines describes the path linearity, a bigger value close to 180 degrees means a straight line. A small angle value means a hesitated path. For now, the angle value can be adapted experimentally. In this case, the Angle limit used in this test was 35 degrees. If the angle between those 3 points is less than 35 degrees, that means the path is hesitating around a line and consumes more distance and more time so this Algorithm will return true, to notify the modified APF the existence of oscillation and calculate the mean value of point the last 2 points and then move to the next step in the flowchart which is **check local-minimum**.
This check local-minimum step checks eight points around the current point as shown in Fig 3.b), to get the field values of the surroundings and then calculates the mean value of those eight points if the normalization value of the mean is less than the positive 1 and greater than negative 1, that means this point is the in equilibrium point in the field, so APF cannot solve this situation because it's a local minimum. So, the modified APF returns to the fusion manager with the local minimum point value and the path to the local minimum, then the fusion manager will use A-Star Algorithm to overcome this issue.

2.2 A-Star check criteria

Fig 4 illustrates the A-Star flow chart. A-Star Algorithm starts a normal search for the goal point using Approximation Heuristics (Manhattan Distance which was tested), After completing a search and before starting to Construct the route, A-Star Algorithm checks each point on the route if this point is out of the local-minimum area or not using Algorithm 2. While checking the route, if there is a point outside the local-minimum area, A-Star Algorithm will stop and then send this point to the fusion manager, if not A-Star Algorithm will continue to reach the goal.

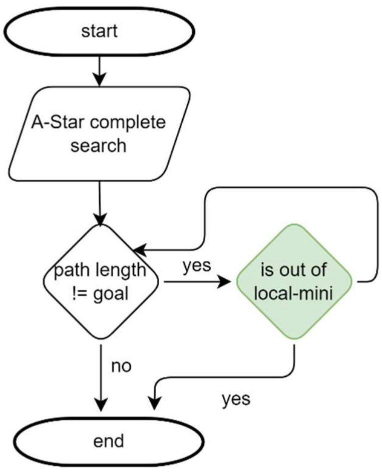

Figure 4. A-Star modification flowchart

This step will start to measure the direction of the field at this point and then compare this direction with the goal direction. if the measured field direction is not toward the goal, this means it is still inside the local minimum. While if the measured field direction points towards the goal point, so now we are outside the local minimum. Once we are outside the local minimum the A-Star modification stops and then returns the current point as a local minimum border point.

Algorithm 2 Check if out of the local-minimum area

```
Initialize the out_of_local_mini parameter with a false value.
Initialize the direction matrix 3*3  by zero
Set center of direction matrix = 1
set y-axis cell leads to the goal  = 2
set x-axis cell leads to the goal  = 2
set the goal cell  = 3
get field direction for this point
IF (field direction == goal direction) then
      out_of_local_mini = True
End if
```

2.3 Fusion Manager Criteria

This is the main part, which switches from APF to A-Star and returns to APF if needed. Fig 5 illustrates the Fusion Manager flow chart which starts the modified APF Algorithm until reaches the goal or returns the local minimum point. In case APF reaches the goal the Fusion Manager uses the modified APF path, in case the modified APF detects the local minimum point, the Fusion Manager swaps to A-Star to reach the goal instead of APF or reach the local minimum border. In case A-Star reaches the goal Fusion Manager will concatenate the first APF route with the A-Star route and then return this route as one path.

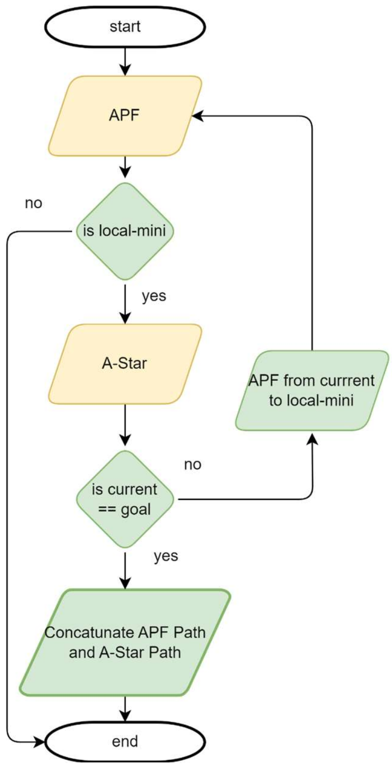

Figure 5. Fusion Manager Flowchart

In case A-Star returns the local minimum border point, the Fusion Manager will reduce the repulsive force first and then, use this point as a current point and the current point as a goal to find the optimized path by the modified APF. Then the second stage is to use the local minimum border as a current point again and the goal point by the modified APF to find the next path, then increase the repulsive force again, the aim of changing the repulsive force is to reduce the path length Fig 6 shows the optimized path (due to changing the repulsive force) and Fig 7 shows the non optimized path.

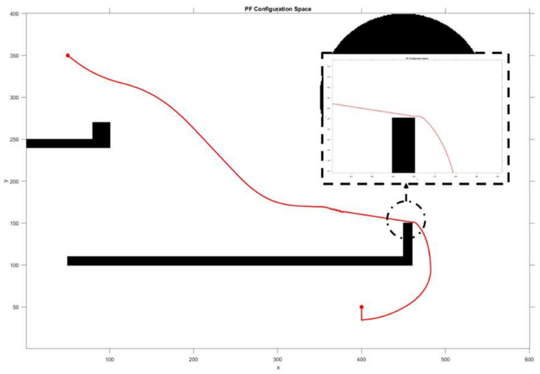

Figure 6. Optimized path

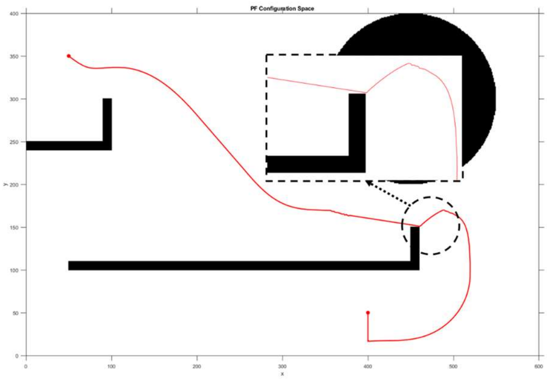

Figure 7. Non-optimized path

3. EXPERIMENTS AND RESULTS

This section discusses the experiments carried out to verify the proposed approach and compares the result with the traditional APF and the traditional A-Star algorithms, in two different scenarios.

3.1 Case I

In this case, the start point will be at (50,300) and the goal point will be at (400,50). The simulation will start with the traditional **A-Star Algorithm**. Fig.8 illustrates the search area painted with red color and the path with blue color. Table 1 illustrates the total time consumed by A-Satr Algorithm to reach the goal. the total time was 1.163 S and the path Length was 929 units.

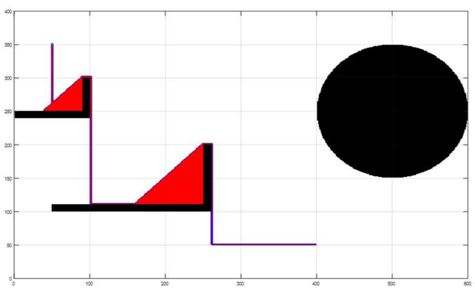

Figure 8. A-Star Search Algorithm

Function Name	Calls	Total Time	distance
AStarGrid	1	1.163 s	929

Table 1. A-Star Time Timetable.

The second Algorithm will be the traditional APF, and the third one is the modified APF, with the same start and end points. Fig 9 shows the Attractive forces, Fig 10 shows the repulsive forces, Fig 11 and Fig 12 illustrate the path for the traditional APF and the modified APF. It can be noticed that the same path and approximately the same time are achieved during the absence of the local minimum. The experiment results shown in Table 2 and Table 3 reveals that the total time consumed by the traditional APF is 0.032 Second, and the total time consumed by the modified Algorithm is 0.046 Second. and the distance is 669.7 units in both

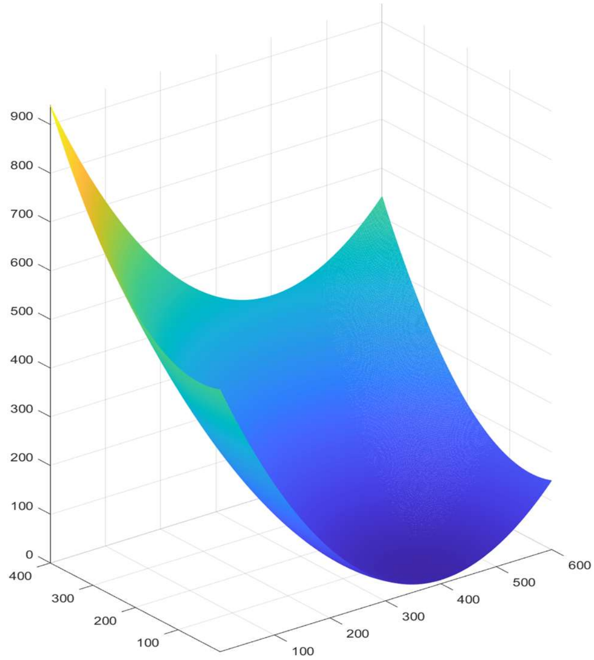

Figure 9. Attractive force

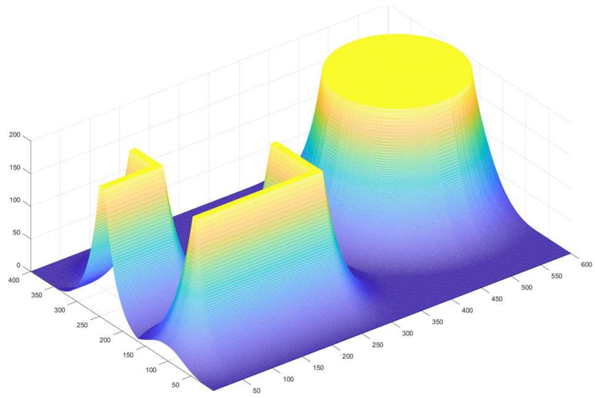

Figure 10. Repulsive Force

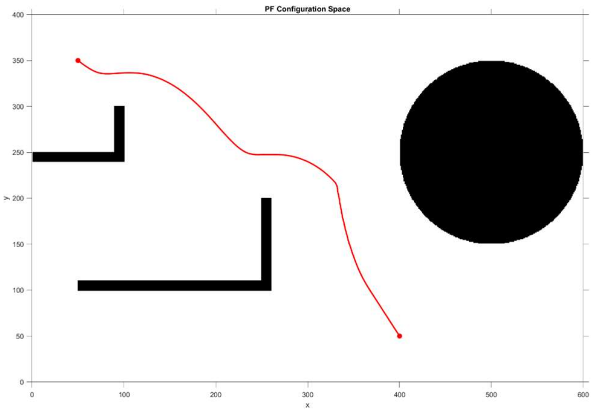

Figure 11. final Path with Obstacles

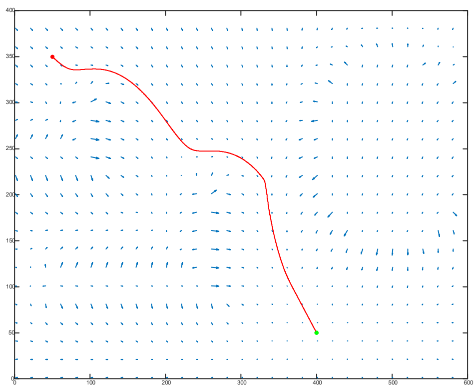

Figure 12. Total field

Function Name	Calls	Total Time	distance
Traditional Potential Field Script	1	0.032 s	669.7 unit

Table 2. APF Timetable

The third test in case I will be the fusion between APF and A-Star

Function Name	Calls	Total Time	distance
Modified Potential Field Script	1	0.046 s	669.7 unit

Table 3. Traditional APF Timetable

3.2 Case II

During this experiment, some changes were made to the scene by adding obstacles to introduce a local minimum test case. Also, the three algorithms will be tested to find the Goal. First: A-Star Algorithm, second APF only then APF fused with A-Star.
Fig 13 shows the search Area with red color, and the final path with blue color, and Table 4 illustrates the consumed time by A-Star to reach the goal of 2.084 sec.. Fig 14 shows the Attractive forces, Fig 15 shows the repulsive forces. Fig.16 proves that APF only failed to find the goal due to falling in a local minimum.

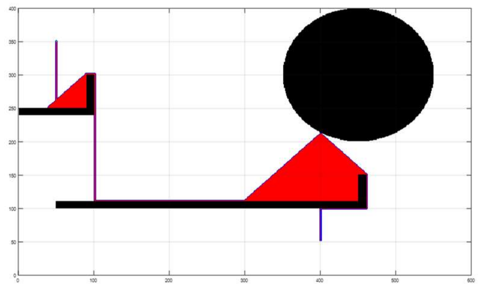

Figure 13. A-Star Search Area with Path

Function Name	Calls	Total Time	distance
AStar	1	2.084 s	951 unit

Table 4. A-Star Timetable

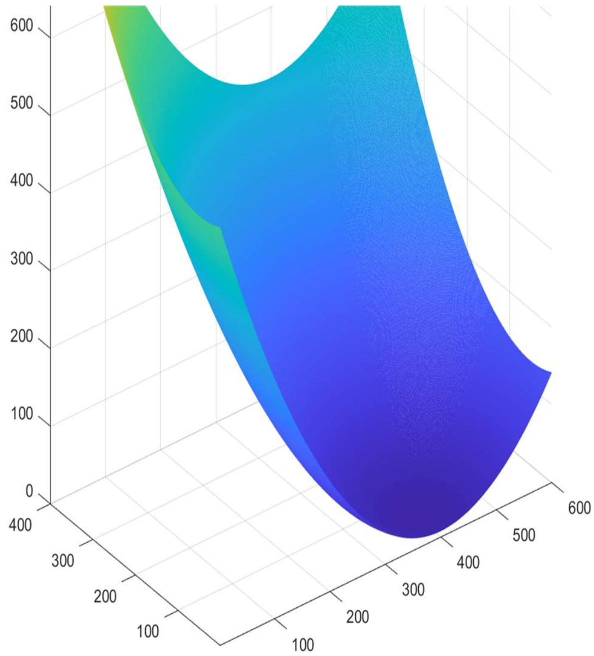

Figure 14. Attractive force

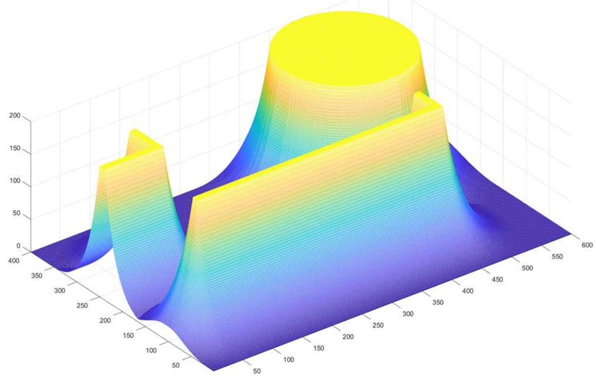

Figure 15. Repulsive Force

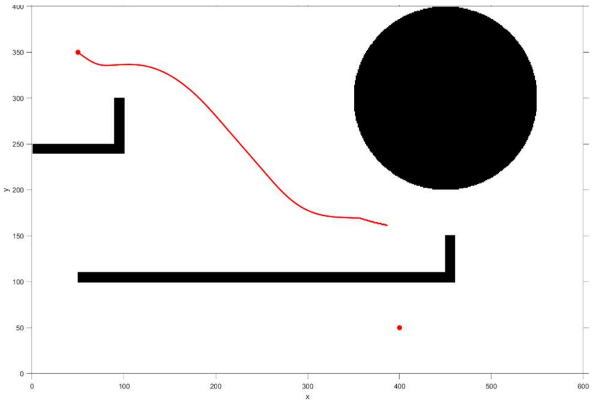

Figure 16. Path with Traditional APF Algorithm

This is the second experiment to reach the goal with the modified APF. The modified APF starts with APF which detects the local minimum. Then the modified APF swap to A-Star to find the goal or the local minimum border. During A-Star starts, it finds the local minimum border first. Fig 17 illustrates the generated path by the A-Star algorithm, which consume more time and more distance than expected, so the modified APF uses the final point at this path as a current point and the local minimum point as a goal to find the optimized path. the two paths (the first path start from the start point to the local minimum point which is generated with APF, and the second path starts from the local minimum point to the local minimum border point which is generated with A-Star) are merged to generate one path. The modified APF starts again from the local minimum border point to the goal. The modified APF reaches another local minimum point. The A-Star algorithm starts again but it reaches the goal this time. Fig 18 shows the A-Star path. The modified APF merges the A-Star path with the previous path to create the total path which is illustrated in Fig 20. The modified APF consumed 1.621 sec to find the optimized path. The total distance planned by the modified APF was 748.4 units.

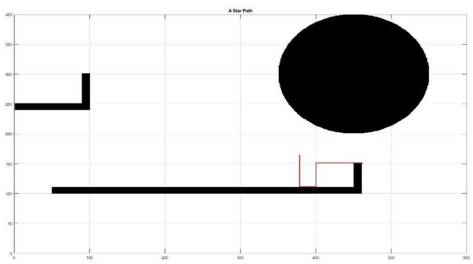

Figure 17. First A-Star Path

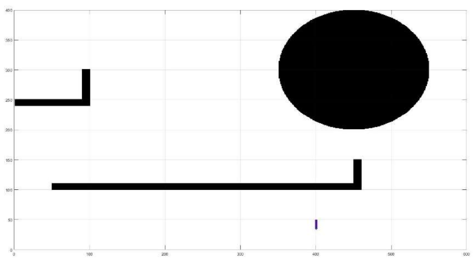

Figure 18. Second A-Star Path

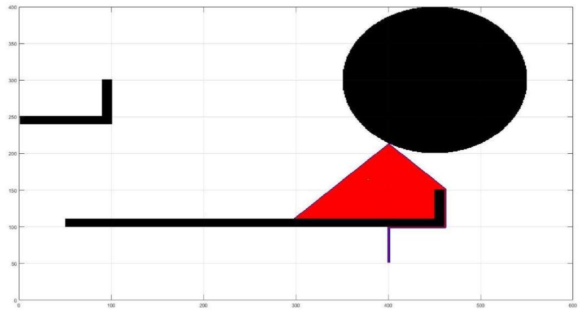

Figure 19. Search Area with A-Star

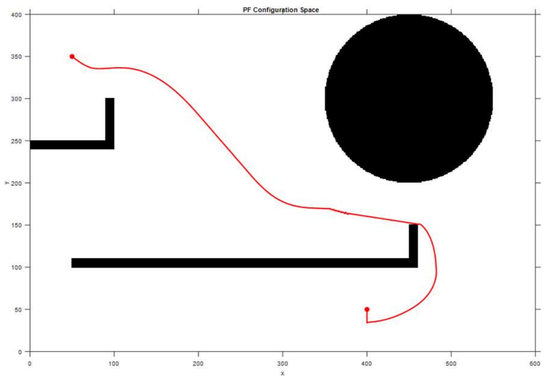

Figure 20. Total Path with the Modified APF

Function Name	Calls	Total Time	distance
Modified Potential Field Script	1	1.621 s	748.4 unit

Table 5. Modified APF Timetable

4. CONCLUSION

The focus of this paper is to address the main limitation of the traditional Artificial Potential Field (APF) algorithm in path planning, which is the tendency to fall into a local minimum and fail to reach the intended goal. To overcome this challenge, the paper proposes an enhanced APF approach that incorporates a Modified APF algorithm, a Fusion Manager Algorithm, and a Modified A-Star algorithm to achieve an optimized path planning algorithm. To verify the proposed enhanced APF approach two experiments were carried out, each experiment was assessed with traditional APF, traditional A-star, and the proposed enhanced APF approach. The experiments revealed the superiority of the proposed approach to overcome the limitation of traditional APF as it managed to get out of the local minimum, also better timing when compared with traditional A-Star. The proposed approach managed to reach an optimized path to the goal in 1.62 sec with a path length of 748 units, while the A-star required 2.04 sec with a path length of 929 units, finaly the APF did not manage to reach the goal due to falling in a local minimum.

	Algorithm	Time	distance
Case I	A-Star	1.163 s	929 unit
	the Traditional APF	0.032 s	669.7 unit
	the Modified APF	0.046 s	669.7 unit
Case II	A-Star	2.084 s	951 unit
	the Traditional APF	-	-
	the Modified APF	1.621 S	748.4 unit

Table 6 Final result conclusion

5. REFERENCES

P. E. Hart, N. J. Nilsson, and B. Raphael, "A Formal Basis for the Heuristic Determination of Minimum Cost Paths," *IEEE Transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.

"A* search algorithm - Wikipedia." https://en.wikipedia.org/wiki/A*_search_algorithm (accessed Jul. 10, 2023).

Khatib, "REAL-TIME OBSTACLE AVOIDANCE FOR M.ANIPULATORS AND MQUIEE ROBOTS," 1985, doi: 10.1109/ROBOT.1985.1087247.

H. Matsuura, K. Nonaka, and K. Sekiguchi, "Model Predictive Obstacle Avoidance Control for an Electric Wheelchair in Indoor Environments Using Artificial Potential Field Method," *2022 IEEE/SICE International Symposium on System Integration, SII 2022*, pp. 19–24, 2022, doi: 10.1109/SII52469.2022.9708755.

J. Zhu, S. Zhao, and R. Zhao, "Path Planning for Autonomous Underwater Vehicle Based on Artificial Potential Field and Modified RRT," *2021 International Conference on Computer, Control and Robotics, ICCCR 2021*, pp. 21–25, Jan. 2021, doi: 10.1109/ICCCR49711.2021.9349402.

C. Cheng, D. Zhu, B. Sun, Z. Chu, J. Nie, and S. Zhang, "Path planning for autonomous underwater vehicle based on artificial potential field and velocity synthesis," *Canadian Conference on Electrical and Computer Engineering*, vol. 2015-June, no. June, pp. 717–721, Jun. 2015, doi: 10.1109/CCECE.2015.7129363.

Q. Zhong, J. Zhao, and C. Tong, "Tracking for humanoid robot based on Kinect," *Proceedings - 2014 International Conference on Mechatronics and Control, ICMC 2014*, pp. 1191–1194, Aug. 2015, doi: 10.1109/ICMC.2014.7231741.

H. Igarashi and M. Kakikura, "Path and posture planning for walking robots by artificial potential field method," *Proc IEEE Int Conf Robot Autom*, vol. 2004, no. 3, pp. 2165–2170, 2004, doi: 10.1109/ROBOT.2004.1307383.

D. L. Sancho-Pradel and C. M. Saaj, "Assessment of Artificial Potential Field methods for navigation of planetary rovers," *2009 European Control Conference, ECC 2009*, pp. 3027–3032, Mar. 2014, doi: 10.23919/ECC.2009.7074869.

F. Plumet, H. Saoud, and M. D. Hua, "Line following for an autonomous sailboat using potential fields method," *OCEANS 2013 MTS/IEEE Bergen: The Challenges of the Northern Dimension*, 2013, doi: 10.1109/OCEANS-BERGEN.2013.6607961.

Iswanto, A. Ma'arif, O. Wahyunggoro, and A. I. Cahyadi, "Artificial Potential Field Algorithm Implementation for Quadrotor Path Planning," *International Journal of Advanced Computer Science and Applications*, vol. 10, no. 8, pp. 575–585, 2019, doi: 10.14569/IJACSA.2019.0100876.

by Y. Koren, S. Member, and J. Borenstein, "Potential Field Methods and Their Inherent Limitations for Mobile Robot Navigation," pp. 1398–1404.

C. Ju, Q. Luo, and X. Yan, "Path Planning Using Artificial Potential Field Method and A-star Fusion Algorithm," *2020 Global Reliability and Prognostics and Health Management, PHM-Shanghai 2020*, Oct. 2020, doi: 10.1109/PHM-SHANGHAI49105.2020.9280929.

J. Chen, C. Tan, R. Mo, H. Zhang, G. Cai, and H. Li, "Research on path planning of three-neighbor search A* algorithm combined with artificial potential field," *Int J Adv Robot Syst*, vol. 18, no. 3, Jun. 2021, doi: 10.1177/17298814211026449/ASSET/IMAGES/LARGE/10.1177_17298814211026449-FIG16.JPEG.

L. Liu, B. Wang, and H. Xu, "Research on Path-Planning Algorithm Integrating Optimization A-Star Algorithm and Artificial Potential Field Method," *Electronics 2022, Vol. 11, Page 3660*, vol. 11, no. 22, p. 3660, Nov. 2022, doi: 10.3390/ELECTRONICS11223660.

https://doi.org/10.5194/isprs-archives-XLVIII-1-W2-2023-867-2023