Improved Artificial Potential Field Method for Motion-Planning of Autonomous Vehicles

Pulkit Paliwal
Department of Electrical Engineering
Indian Institute of Technology, Bombay
Mumbai, India
20d100021@iitb.ac.in

***Abstract*—Motion Planning is one of the primary tasks that needs to be taken care of for autonomous vehicles to function reliably. Given an environment to navigate, the vehicles must be able to navigate towards the desired goal location without colliding with any obstacles. Furthermore, one would want the vehicles to take the shortest route possible, in order to minimize the fuel consumption, and thus reduce the cost of operation. To find these optimal paths, several algorithms have been introduced over time. One such algorithm is the Artificial Potential Field Method, which considers an artificial field around the obstacles and goal, and by moving along the force given by the gradient of the resultant field, an reasonably optimal path is obtained. This paper discusses a variation of the Artificial Potential Field Method for path planning of autonomous vehicles which generates shorter paths than the ones generated using the original algorithm in a wide variety of environments. The simulation results presented reflect that the proposed algorithm reduces the path length by roughly 10% as compared to the path generated by the classical potential field method.**

***Index Terms*—Autonomous, Potential Field, Motion Planning, Path Optimization**

I. Introduction

The introduction of autonomous vehicles has proven to be of great advantage to the human race. They have not only led to a significant reduction in effort required in driving, but have proven to make roads much more safer. Autonomous vehicles have also shown to have improved energy consumption, thus making them more environment-friendly [1]- [3]. A major task that needs to be taken care of during the motion of the autonomous vehicles is that they should move on a collision-free path. In general, the vehicle's on-board computer has information about the environment it is navigating in. To minimize energy consumption, it is also required that these paths on which the vehicle moves are as short as possible.

This has given rise to the field of motion planning, wherein the research is focused on developing fast algorithms that generate reasonably optimal paths for the vehicle to move on. The authors of [4] surveyed a wide variety of motion planning techniques. These techniques include are but not limited to graph-search methods (such as A* [5], D* Lite [6]), sampling-based planning (such as RRT [7]) and Function Optimization [8]. A frequently used path planning technique was introduced by Khatib in [9], which is popularly called as Artificial Potential Field Method. This paper proposes a variation of the Artificial Potential Field Method, which can produce significantly shorter paths in densely obstructed environments with a minor trade-off of slightly increased computation time. The paper first presents the classical and a slightly modified version of the classical artificial potential field (hereon referred to as APF) method in Section II. It then presents the proposed APF variation strategy along with its pseudocode in Section III. Finally the simulation results are presented in Section IV, with some concluding remarks in Section V.

II. Classical Artificial Potential Field Method

The APF method was introduced in [9] by Khatib, in the year 1986. The technique involves assuming the obstacles and the goal to have their own "potential fields", which vary with the distance between the vehicle and the obstacles and the goal respectively. The obstacles are assumed to be generating repulsive fields (with potential $U_{repulsive}$), while the fields generated by the goal are taken to be attractive (with potential $U_{attractive}$). Mathematically the fields can be represented as follows:

$$U_{attractive}(x) = \frac{1}{2}k_{attractive}(x_{v,goal})^2 \tag{1}$$

$$U_{repulsive}(x) = \frac{1}{2}k_{repulsive}(\frac{1}{x_{v,obstacle}} - \frac{1}{x_o})^2 \tag{2}$$

where, $(x_{v,goal})$ represents the distance between the vehicle and the goal, and $(x_{v,obstacle})$ represents the distance between the vehicle and the obstacle. The quantity x_o represents the distance between the obstacle and the vehicle from and beyond which $U_{repulsive}(x)$ falls to zero. $k_{attractive}$ and $k_{repulsive}$ are the attractive and the repulsive field constants, respectively. The forces given by these potentials are then summed up using vector addition, giving,

$$F(x) = |\vec{F}_{attractive}(x) + \vec{F}_{repulsive}(x)| \tag{3}$$

where,

$$\begin{aligned} F_{attractive}(x) &= -\nabla U_{attractive}(x) \\ \implies F_{attractive}(x) &= k_{attractive}x_{v,goal} \end{aligned} \tag{4}$$

directed towards the goal and

$$\begin{aligned} F_{repulsive}(x) = -\nabla U_{repulsive}(x) = \\ k_{repulsive}(\frac{1}{x_{v,obstacle}} - \frac{1}{x_o})\frac{\nabla x_{v,obstacle}}{x^2{}_{v,obstacle}} \end{aligned} \tag{5}$$

directed away from the obstacle, whenever $x_{v,obstacle} < x_o$ and 0 otherwise. The algorithm directs that at any point in the region where the vehicle is moving, it should move along the direction of the net force experienced by it at that position. Since the goal attracts the vehicle, and the obstacles repel the vehicle, it becomes intuitively clear that the vehicle will move towards the goal. However, it is possible that the goal has multiple obstacles surrounding it, which would increase the repulsion to an extent that it becomes impossible for the vehicle to reach the goal. Moreover, it is possible that the forces at a certain position all cancel out, i.e., the net potential attains a local minimum, and the vehicle gets "stuck" in that position, beyond which the algorithm is unable to plan a path.

To tackle these problems, as well as the problem of dynamic obstacles, the authors of [10] introduced a modified APF method. In this method the authors first modify the repulsive potential function (for $x_{v,obstacle} < x_o$, 0 otherwise) as follows,

$$U_{repulsive}(x) = \frac{1}{2}k_{repulsive}\left(\frac{1}{x_{v,obstacle}} - \frac{1}{x_o}\right)^2 x_{v,goal}^n \quad (6)$$

From this we have,

$$F_{repulsive}(x) = F_{1,repulsive}(x) + F_{2,repulsive}(x) \quad (7)$$

where,

$$F_{1,repulsive}(x) = k_{repuslive}\left(\frac{1}{x_{v,obstacle}} - \frac{1}{x_o}\right)\frac{x_{v,goal}^2}{x^n{}_{v,obstacle}} \quad (8)$$

and

$$F_{2,repulsive}(x) = \frac{n}{2}k_{repuslive}\left(\frac{1}{x_{v,obstacle}} - \frac{1}{x_o}\right)^2\frac{x_{v,goal}^{n-1}}{x^2{}_{v,obstacle}} \quad (9)$$

This helps to combat the goal non-reachable with obstacles nearby (GNRON) problem. One can see that the repulsive function value now depends on the distance between the goal the vehicle as well, and reduces the repulsive force (for $n \geq 2$) experienced near the goal significantly as compared to the classical potential function. Fig. 1. shows the path planned by APF method (repulsive potential function from equation (6) is used with $n = 2$).

The paper further modifies the algorithm by proposing an RHG (Regular Hexagon-Guided) Method to tackle the local minima problem. It also modifies the potential function to accommodate for dynamic obstacles by adding a relative velocity dependent potential to the repulsive potential function. This causes the repulsive potential to be non-zero only when the vehicle is moving towards the obstacle, and within distance x_o of the obstacle.

III. Proposed APF Method

This paper proposes a variant of the classical APF method, which not only accounts for GNRON Problem as in [10], but also produces shorter paths than the classical APF method

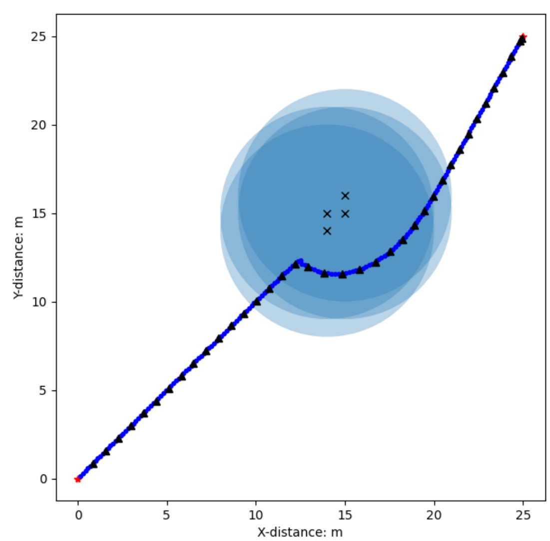

Fig. 1. Path planned with APF method (repulsive potential as in equation (6), $n = 2$). The blue circles represent the region around the obstacles where the repulsive potential is non-zero.

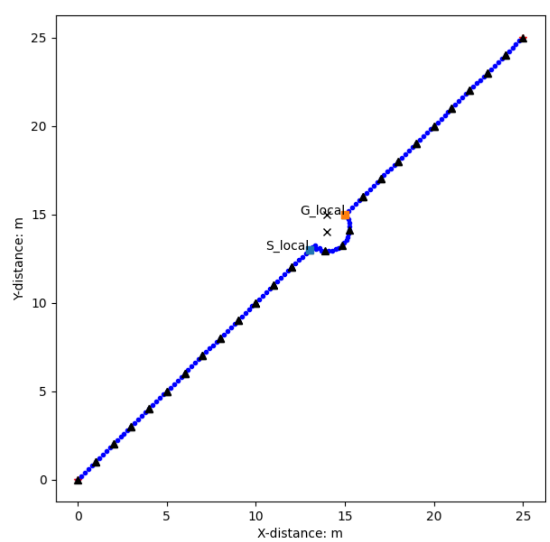

Fig. 2. Planning path for navigating an environment consisting of a single obstruction on the line joining the start and goal points. X indicates obstacles, while the triangles and blue line indicate the planned path.

(having repulsive potential as in equation (6) with $n = 2$). The strategy draws parallels from [11], where the authors improved the classical A* search algorithm by planning for small sections around obstacles blocking the straight line joining the goal and the start position.

The classical method using repulsive potential as in equation (6) with $n = 2$ will plan a route as shown in Fig. 1. The proposed strategy, instead of planning from start to goal using the APF method, first obtains a straight line joining thee start and the goal points. The equation of this line is given by,

$$\frac{y - y_{goal}}{y_{goal} - y_{start}} = \frac{x - x_{goal}}{x_{goal} - x_{start}} \tag{10}$$

where, (x_{start}, y_{start}) denotes the starting point and (x_{goal}, y_{goal}) denotes the goal.

The idea here is to move on the straight line whenever possible, avoid the obstacles on the straight line by moving on a deviated path computed using the APF method, and return to the line as soon as possible. Thus we need a set of local starting and local goal points around these obstacles, and obtain the deviated paths using these local start and goal points as the input to the APF algorithm. These points can be computed by obtaining the points of intersection of a circle of given radius r centered at the obstacles on the line. Path planned using this algorithm for a single obstacle on the line case is shown in Fig. 2.

Another improvement that the proposed strategy uses is to ”collectively avoid” the obstacles on the straight line if they are in close vicinity of each other. Thus for a set of consecutive obstacles on line within ϵ distance of each other, the collective avoidance is implemented by planning the deviated path using the APF algorithm with local start as the local start of the point from this set closest to the start and local goal point as local goal of the point from this set closest to the goal (as in Fig. 3.). The set deviated path segments is then obtained by computing the paths using APF method with (local start, local goal) sets as inputs to the algorithm. Then by merging the segments of the straight which were

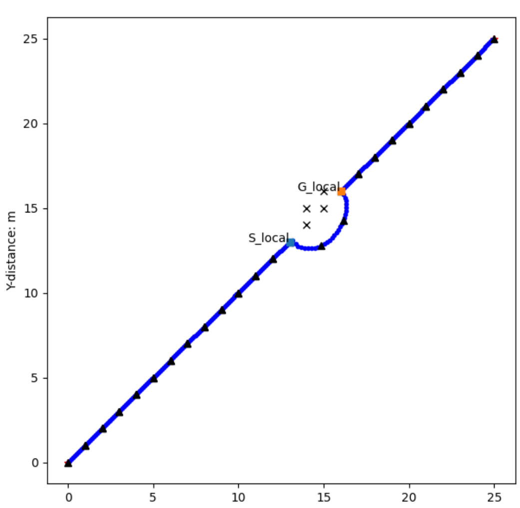

Fig. 3. Planning for4multiple obstacles on the shortest path case

obstacle free with the deviated paths, a final obstacle free path is obtained.

This will result in a new path planned which is shorter than the path planned by the classical APF method using repulsive potential as in equation (6). The local start and end points are obtained by an intersection of the circle of desired radius centered at the obstacle with the straight line joining the goal and the starting points.

Algorithm 1 Improved APF Algorithm

def $localstartgoal(p1, p2, obstacle)$
$z, w \leftarrow$ Intersection points of circle of radius r centered at obstacle with line passing through $p1$ and $p2$, z closer to start point
return z,w
def $join(point1, point2)$
return Points on line joining point1 and point2
def $apf(point1, point2, Obstacles, x_o)$
return path planned from point1 to point 2 using APF method with repulsion potential as in equation (6) and $n = 2$
def $Reorder(AvoidanceList, Start, Goal)$
return Ordered $AvoidanceList$ i.e., points closer to the start are at the beginning of the list and the points closer to the goal are at the end of the list
def $improvedApf(start, goal, obstacles, x_o)$
$Path \leftarrow []$
$AvoidanceList \leftarrow []$
for O in Obstacles **do**
  if O lies on $StraightLine$ **then**
    $AvoidanceList \leftarrow AvoidanceList \cup O$
  end if
end for
$Reorder(AvoidanceList, Start, Goal)$
$Index = 0$
$LocalStart \leftarrow []$
$LocalGoal \leftarrow []$
while $Index < len(AvoidanceList)$ **do**
  $IndexOfNextObs = Index$
  while $IndexOfNextObs + 1 < len(AvoidanceList)$ and $AvoidanceList[IndexOfNextObs + 1]$ is within ϵ distance of $AvoidanceList[IndexOfNextObs]$ **do**
    $IndexOfNextObs = IndexOfNextObs + 1$
  end while
  $obs1 \leftarrow AvoidanceList[Index]$
  $obsN \leftarrow AvoidanceList[IndexOfNextObs]$
  $z_1, w_1 \leftarrow localstartgoal(Start, Goal, obs_1)$
  $z_N, w_N \leftarrow localstartgoal(Start, Goal, obs_N)$
  $LocalStart \leftarrow z_1$
  $LocalGoal \leftarrow w_N$
  $Index = IndexOfNextObs + 1$
  $Path \leftarrow join(Start, LocalStart[0])$
  $i = 0$
end while

while $i < len(LocalStart) - 1$ **do**
$Path \leftarrow Path \cup\ apf(LocalStart[i],\ LocalGoal[i],\ Obstacles,\ d_o)$
if $i < len(LocalStart)$ **then**
$Path \leftarrow Path \cup\ join(LocalGoal[i],\ LocalStart[i+1])$
else
$Path \leftarrow Path \cup\ join(LocalGoal[i],\ Goal)$
end if
$i = i + 1$
end while
return $Path$

IV. Simulations and Results

The claim regarding the better performance of the proposed improved APF algorithm has been verified by performing simulations on multiple experimental environments. The proposed algorithm is compared with the classical APF method (having repulsive potential function from equation (6) with $n = 2$).

A. Simulation Environment

The vehicle is permitted to move anywhere in the space, i.e. the region does not have any boundary. The obstacles are placed at various points across the grid, and for the purpose of the simulation, considered to be point sized. The vehicle as well, been considered to be a point object. One can however, account for the size of the vehicle by appropriately adding more point obstacles beyond the boundary of the actual obstacle. This would increase the apparent size of the obstacle and the vehicle would have some margin even if the vehicle passed close to the apparent boundary. The starting position is $(0, 0)$ and the goal position is (25,25).

B. Parameters

The value of x_o for the both the algorithms is taken to be 6 meters. The values for $k_{attractive}$ and $k_{repulsive}$ are 1 and 0.8 respectively. The vehicle is taken to have arrived at its goal if it is within 0.2 meters of its goal position. The radius of the intersecting circle has been taken to be $\sqrt{2}$ meters.

C. Results

Although the algorithm was tested over a wide variety of environments having different obstacle positions, three such simulations have been presented, which show that the proposed algorithm indeed generates a more optimal path than the classical APF method (having repulsive potential function from equation (6) with $n = 2$). The two algorithms were run on the same environment and their results have been compared. Simulation-I is shown in Fig. 4 and Fig. 5., Simulation-II in Fig. 6 and Fig. 7 and Simulation-III is shown in Fig.8 and Fig. 9. The results for each simulation are summarized in Tables I, II and III respectively. Table IV gives an overview of the promising results of the proposed algorithm.

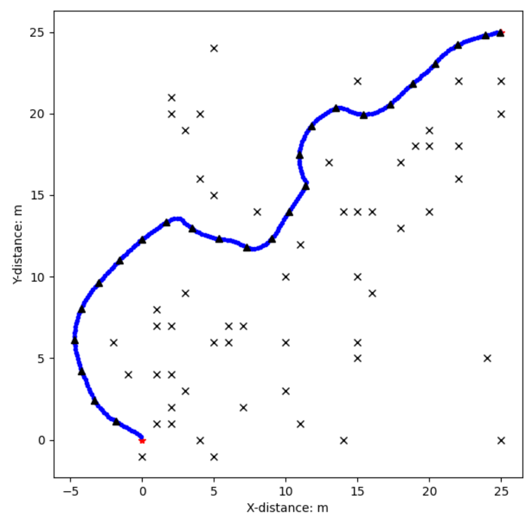

Fig. 4. Simulation-I: Path Generated by the classical APF method (having repulsive potential function from equation (6) with $n = 2$). The crosses indicate the obstacles. The blue line and black triangles represent points on the path planned by the algorithm.

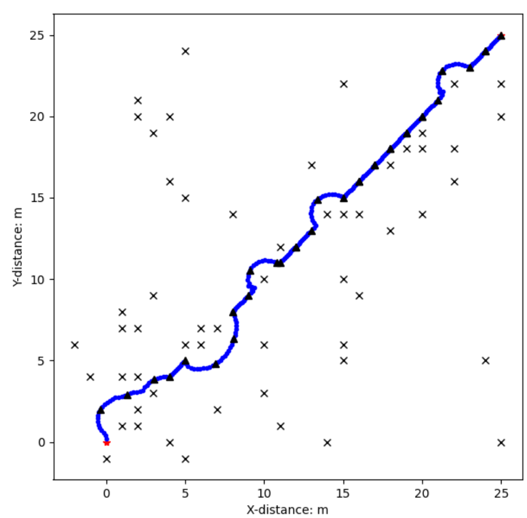

Fig. 5. Simulation-I: Path generated by the proposed algorithm. The crosses indicate the obstacles. The blue line and black triangles represent points on the path planned by the algorithm.

TABLE I
Results for Simulation-I

Algorithm	Time Taken (s)	Path Length (m)
APF	0.05509	49.00
Proposed Algorithm	0.07241	40.77

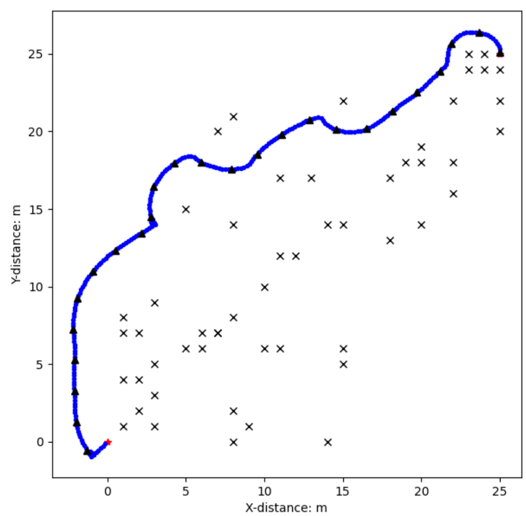

Fig. 6. Simulation-II: Path Generated by the classical APF method (having repulsive potential function from equation (6) with $n = 2$)). The crosses indicate the obstacles. The blue line and black triangles represent points on the path planned by the algorithm. triangles represent points on the path planned by the algorithm.

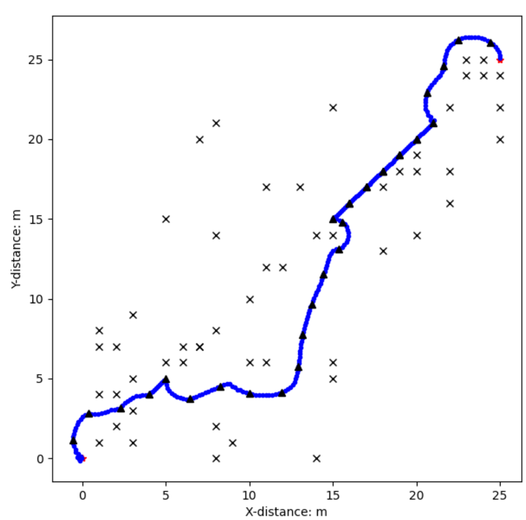

Fig. 7. Simulation-II: Path generated by the proposed algorithm. The crosses indicate the obstacles, while the circles indicate the potential fields around the obstacles. The blue line and black triangles represent points on the path planned by the algorithm.

TABLE II
RESULTS FOR SIMULATION-II

Algorithm	Time Taken (s)	Path Length (m)
APF	0.068087	50.00
Proposed Algorithm	0.082405	47.50

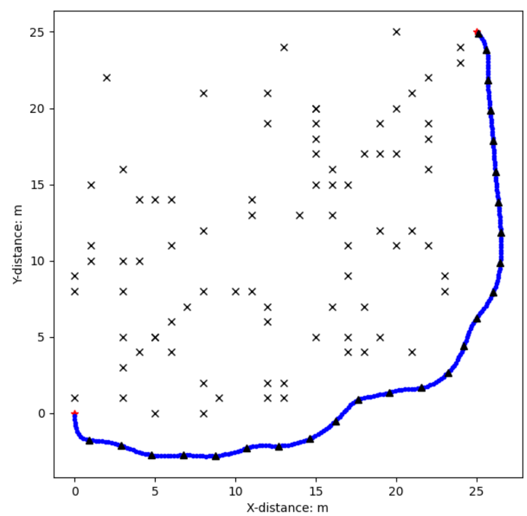

Fig. 8. Simulation-III: Path Generated by the classical APF method (having repulsive potential function from equation (6) with $n = 2$)). The crosses indicate the obstacles. The blue line and black triangles represent points on the path planned by the algorithm.

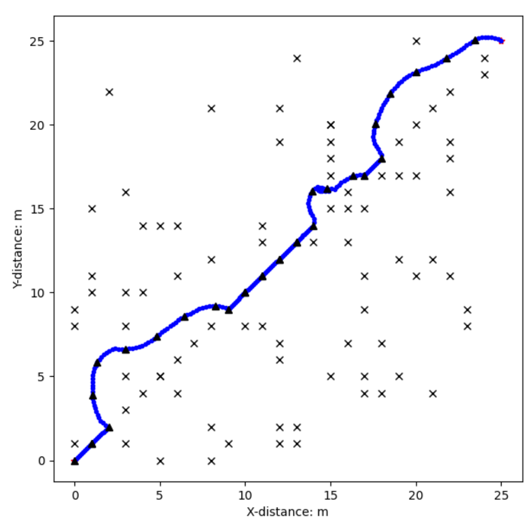

Fig. 9. Simulation-III: Path generated by the proposed algorithm. The crosses indicate the obstacles. The blue line and black triangles represent points on the path planned by the algorithm.

TABLE III
RESULTS FOR SIMULATION-III

Algorithm	Time Taken (s)	Path Length (m)
APF	0.087709	49.20
Proposed Algorithm	0.085937	42.11

TABLE IV
SUMMARY OF SIMULATION RESULTS

Simulation	% change in Path Length	% change in computation time
I	-16.79%	31.43%
II	-5%	21%
III	-14.41%	-2.02%

V. CONCLUSION AND FUTURE SCOPE

An improved variant of the artificial potential field method for motion planning of autonomous vehicles was introduced in this paper. The simulation results show the superiority of the improved algorithm in obtaining shorter paths as compared to the classical APF algorithm. The APF algorithm used in the proposed method also accommodates for the GNRON problem. There are certain cases in which the proposed algorithm might ”get stuck”. This happens when the resultant force on the vehicle is zero (i.e., the potential attains a local minima) at that position, it may fail to generate a path beyond that position. The RFG method mentioned in Section-II can be extended to be applied in the APF algorithm used in the proposed algorithm to improve the local minima problem. Future scope of improvement also includes accounting for dynamic environments.

However, the results of the currently proposed algorithm are very promising. Although it might be said in general that the improved path length comes at the cost of increased computational time, there can be cases such as the one shown in Simulation-III, where, the proposed method not only results in a shorter path length, but the time needed for computing this path is lesser than that needed for computing a path using the classical APF algorithm (with repulsive potential function as in equation (6) and $n = 2$). The methodology used is simple and easy to understand and potentially lays a foundation for the development of a variety of variations of existing motion planning algorithms in the future.

REFERENCES

[1] S. Rafael, L. P. Correia, D. Lopes, J. Bandeira, M. C. Coelho, M. Andrade, C. Borrego, and A. I. Miranda, “Autonomous vehicles opportunities for cities air quality,” Sci. Total Environ., vol. 712, Apr. 2020, Art. no. 136546.

[2] A. Vahidi, A. Sciarretta, ”Energy Saving Potentials of Connected and Automated Vehicles,” Transportation Research Part C: Emerging Technologies, vol. 95., Sep. 2018, doi:10.1016/j.trc.2018.09.001.

[3] Coppola, Riccardo and Morisio, Maurizio, ”Connected Car: Technologies, Issues, Future Trends,” in ACM Comput. Surv., vol. 49, issue 3, no. 46, pp. 1-36, September 2017, doi: 10.1145/2971482

[4] D. González, J. Pérez, V. Milanés and F. Nashashibi, ”A Review of Motion Planning Techniques for Automated Vehicles,” in IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 4, pp. 1135-1145, April 2016, doi: 10.1109/TITS.2015.2498841.

[5] P. E. Hart, N. J. Nilsson and B. Raphael, ”A Formal Basis for the Heuristic Determination of Minimum Cost Paths,” in IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100-107, July 1968, doi: 10.1109/TSSC.1968.300136.

[6] S. Koenig, M. Likhachev, ”D*Lite,” Eighteenth National Conference on Artificial Intelligence, American Association for Artificial Intelligence, USA (2002), pp. 476-483

[7] S. M. LaValle , J. J. Kuffner, “Randomized kinodynamic planning,” Int. J. Robot. Res., vol. 20, no. 5, pp. 378–400, 2001.

[8] D. Kogan and R. Murray, “Optimization-based navigation for the darpa grand challenge,” in Proc. CDC, 2006, pp. 1–6.

[9] O. Khatib, ”The Potential Field Approach And Operational Space Formulation In Robot Control,” Adaptive and Learning Systems. Springer, Boston, MA., 1986, pp. 367–377. doi:10.1007/978-1-4757-1895-9_26

[10] X. Fan, Y. Guo, H. Liu, B. Wei, W. Lyu, ”Improved Artificial Potential Field Method Applied for AUV Path Planning,” in Mathematical Problems in Engineering, Volume 2020, Article ID 6523158, 21 pages, doi:10.1155/2020/6523158

[11] C. Ju, Q. Luo and X. Yan, ”Path Planning Using an Improved A-star Algorithm,” 2020 11th International Conference on Prognostics and System Health Management (PHM-2020 Jinan), 2020, pp. 23-26, doi: 10.1109/PHM-Jinan48558.2020.00012.