

# Path Planning: An Analytical Comparison of A\* Algorithm Fusion DWA and Ant Colony Algorithm Fusion DWA

Ernuo Zhang

School of Mechatronics, Harbin Institute of Technology, Harbin, China

120L033103@stu.hit.edu.cn

**Abstract.** In recent decades, the robotics sector has grown significantly, and the automation of industrial production and daily life has become a shared objective. In recent years, self-driving cars have emerged around the globe, but the majority are still in the research phase and very few are in use in developed regions, not only because of their cost, but also because of their safety and efficiency for path planning and the speed of avoidance in the face of dynamic obstacles, which has been a concern. In this work, multiple path planning algorithms, including the A\* method, the dynamic window technique, and the ant colony optimization (ACO) algorithm, will be empirically explored. Initially, the fundamental algorithm will be enhanced, and then the enhanced algorithms will be fused, resulting in a considerable increase in responsiveness and computing efficiency. The study will then attempt to combine the A\* algorithm and the ACO algorithm with the dynamic window method and analyse the pros and cons of the two fusion algorithms for path planning in various situations. The A\* algorithm fused with the dynamic window method is, on average, more suited for complex situations than the ACO algorithm fused with the dynamic window technique.

**Keywords:** Path planning, DWA, A\* algorithm, ACO algorithm.

## 1. Introduction

Robots, which vary from basic industrial robots to autonomous ones for difficult missions like space exploration, increasingly help or even completely replace human work in numerous fields. Robots may be used in a variety of industries and for a wide range of jobs because to their high adaptability and flexibility [1]. The public's attention has steadily been drawn to artificial intelligence technology as a result of the fast advancement of computer technology; autonomous driving technology is an essential application of artificial intelligence technology. The main goal of autonomous driving technology is to allow cars to drive autonomously, reduce driver fatigue and improve the safety and comfort of car driving. For those about self-driving robots, studying their path-planning approaches is a very worth exploring field at the moment. Today, there are many approaches to path planning in existing research. The objective of robot 3D (three-dimensional) route planning is to discover an optimum and collision-free path in a 3D workspace while taking kinematic limitations into consideration (including geometric, physical, and temporal constraints). Path planning, unlike motion planning, which must take into account dynamics, aims to discover a kinematically optimum path in the shortest amount of time and to characterise the surroundings exhaustively [2]. Algorithms for 3D path planning include the visibility graph, which connects the visible vertices of a polyhedron, random-exploring algorithms such as rapidly exploring random tree and Probabilistic Road Map, optimal search algorithms (like Dijkstra's algorithm, A\*, and D\*), and bioinspired planning algorithms [3-7].

The need of efficient path-planning algorithms for current autonomous underwater vehicles cannot be overstated. Conventional path-planning algorithms in artificial intelligence were not designed to handle huge, continuous, and subject to currents landscapes. We provide a novel Fast Marching (FM)-based strategy for resolving the following issues. To effectively extract a 2-D continuous route from a discrete description of the environment, the FM\* approach is first developed. Second, we account for underwater currents by anisotropically expanding the original FM approach. Finally, the vehicle's turning radius is added as a constraint on the optimal route curvature for both isotropic and anisotropic media. Ultimately, it is shown that a multiresolution method may speed the whole path-planning

process [8]. Safety, dependability, and regulatory systems still require improvement. For instance, Uber caused the first self-driving pedestrian fatality accident, while Tesla has been involved in a number of tragic accidents. This was a response to the information security risks of autonomous driving, which are still at the root of a number of security challenges concerning data security, industrial security, national traffic security, and geographic information security, among other things. As a result, autonomous driving and its path planning still face several obstacles. First of all, there are no rules or regulations in place, and highly autonomous commercial cars are subject to roadside restrictions. Second, there is a lack of coordination and co-ordination between the development of self-driving vehicles and the construction of transportation infrastructure, and the promotion of intelligent road construction has to be increased. The requirements of autonomous driving are vastly different from those for installing and maintaining traffic infrastructure and traffic signs. On a technological level, big data, cloud platforms, high-precision maps, and automotive industry chain resources are still in their infancy. Automotive intelligent technology is still in the development iteration phase. Furthermore, the 32- or 64-line mechanical rotating LIDAR that the majority of firms utilize in their autonomous driving prototypes is expensive and does not adhere to vehicle rules.

To address the shortcomings of autonomous vehicles in path planning, several algorithms for path planning in terms of their responsiveness and different responses to different types of environment (e.g. rotation angle, angular speed of running lines) will be discussed in this paper. On the other hand, the best route can only be determined using conventional path planning algorithms in a situation where the position of barriers is fixed. Yet, due to the prevalence of robots in dynamic settings like banks, hospitals, and shopping centers, conventional path planning algorithms can quickly collide with risks like dynamic obstructions. Although there are several fusion algorithms available today to address the problem of dynamic obstacle avoidance, it is currently difficult to determine which fusion method is the most effective one in the same environment, in terms of time and path.

As a result, this essay is prepared to contrast the two fusion algorithms, A\* fusion DWA and Ant Colony fusion DWA, and determine which approach offers the best overall solution.

## 2. Methodology

### 2.1. A\* Algorithm Fusion DWA

#### 2.1.1 The Classical A-STAR Algorithm

The A\* algorithm, suited for demand forecasting since it not only guarantees the finding of the best path in a centralized system, but also boosts efficiency through systematic search. The A\* technique for 2D matrix maps uses an evaluation function to choose the next node to visit [9].

$$f(n) = g(n) + h(n) \quad (1)$$

It is the expense of a route from the origin to the cluster at hand, denoted by  $g(n)$ .; due to the uncertainty of the path induced by barriers that are not yet known, the cost of travelling from the present node to the endpoint is denoted by the heuristic function  $h(n)$ . The A\* algorithm's heuristic function for calculating distance incorporates Manhattan distance and Euclidean distance:

$$h(n) = |x_n - x_g| + |y_n - y_g| \quad (2)$$

$$h(n) = \sqrt{(x_n - x_g)^2 + (y_n - y_g)^2} \quad (3)$$

Where the coordinates of the target point G are  $(x_g, y_g)$  and the coordinates of the current node are  $(x_n, y_n)$ . The current node's position in the graph is indicated by the brackets.

The representation of the Manhattan distance can be found in equation (2), and the portrayal of the Euclidean distance can be found in equation (3). In this particular piece of work, the Geometric representation of distance is put to use.

### 2.1.2 Improved A-STAR Algorithm

The another searching cluster in the conventional A\* algorithm, which is founded on the width first approach, can only be found inside the present node's immediate eight-degrees of separation, by expanding the search direction with the rise in the amount of scan levels, the conventional multilateral search direction is enhanced to the five-side-length investigation, as presented in this study. This is accomplished while maintaining the initial eight directions for search intact [9]. Here is how to write down the connection:

$$\begin{cases} D_1 = 8 \\ D_n = 8 \times \left[ 1 + \frac{n(n-1)}{2} \right] \quad (n \geq 1, n \in \mathbb{Z}) \end{cases} \quad (4)$$

In order to improve the efficacy of the algorithm and account for the increased number of nodes that need to be processed in the OPEN LIST as a result of the multi-lateral search, a procedure that removes redundant nodes has been developed. This procedure ensures that the algorithm is accurate and that it can process information at a high rate. These are the procedures you'll take:

Stage 1: Researchers will utilise A as the first node, and the third node as the evaluation starting node.

Stage 2: Based on the fact that each single grid configuration in the 2D grid image is completely distinct, researchers determine whether or not there are any obstacles in the path that connects the first and third nodes.

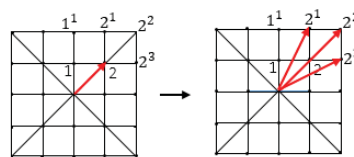
Stage 3: If there are no obstructions in the path, this point to the fact that the second node is unnecessary; otherwise, the second node is retained for use as a turning node.

Stage 4: The node that was determined to be a stopping point is used as the new beginning point, and the evaluation of the network continues with the node after it.

Stage 5: The procedure of evaluating each component should be repeated until the conclusion is reached.

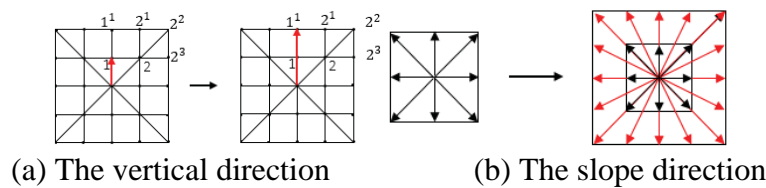
### 2.1.3 Expanding the Direction of Exploration

Because it is unable to turn consistently, the robot will either swing erratically or become stuck if it encounters a particularly thick concentration of obstructions. The methods shown in Fig. 1 will be implemented in this chapter to broaden the scope of inquiry, with the goals of making the route less bumpy and reducing the robot's capacity for steering [10].



**Figure 1.** Expanding exploration direction [10]

Specific steps: Based on Fig. 2, if node 1 is a vacant node, we should continue examining node 11. (a). As indicated in Fig. 2, if node 2 is a vacant node, then nodes 21, 22, and 23 should be explored sequentially (b).



**Figure 2.** Exploration steps of the improved A\* algorithm [10]

### 2.1.4 A Fusion Algorithm

The A\* algorithm, originally developed by Google, is used here in its enhanced form to do global route planning. An enhanced DWA algorithm uses the final destination on the global route as a sub-target point, sending the vehicle in a linear progression towards every successive destination.

This article adds a smoothing component to the assessment function so that the route can be made even smoother, and the stability of the robot's movements can be improved. The smooth point is determined to be the sub-target point that is located in greatest proximity to the point that is currently being targeted by the robot. This allows the robot's speed to be optimised as it moves closer to the smooth point. The assessment function of the enhanced fusion algorithm can be seen in formula (5), which reads as follows:

$$G(v, w) = \sigma(\alpha \cdot \text{head}(v, w) + \beta \cdot \text{dist}(v, w) + \gamma \cdot \text{velocity}(v, w) + \text{head}_{\text{smooth}}(v, w)) \quad (5)$$

Where  $\text{head}_{\text{smooth}}(v, w)$  shows the angle difference between the sleek point and the trajectory when the robot arrives at the conclusion of the route at speed  $(v, w)$ .

## 2.2. Ant Colony Algorithm Fusion DWA

Iterations are a fundamental building block of the Ant Colony Optimization (ACO), was the first swarm intelligence-based algorithm. During each repetition, a certain number of ants build comprehensive solutions by making use of heuristic information and the experiences that have been gathered from earlier populations of ants. The pheromone trail, which is distributed on the various components that make up a solution, is used to provide a graphical representation of the experiences that have been accumulated. Depending on the nature of the problem that needs to be solved, the pheromone could be distributed on either the connections or the components of a solution. The following is a description of the process that is used to change the pheromone regulation [11].

### 2.2.1 Algorithm Design

Ant colony search is utilised to tackle multi-stage continuous decision problems with an exponential and logarithmic target value.

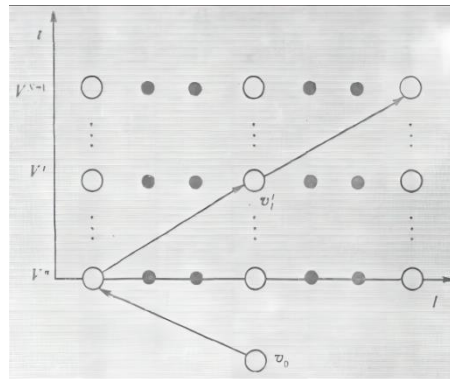
The multi-stage continuous decision problem with additivity and monotonicity starts by discretizing the decision variable  $u(t)$  at each stage into a series of discrete lattice points  $u_1 v(t) \in U(t)$ ,  $1=1, 2, \dots, q_r$  within its allowable decision set.,  $q_r$  (let each dimension of the  $r$ -dimensional decision vector be uniformly discretized into  $q$  lattice points), which is called the discretized decision variable, and is denoted as the stage discretized decision variable; the set of discrete lattice points of all decision variables at stage  $t$  is denoted as  $U_d(t) \in U(t)$ , which is called the discretized allowed decision set. Then the discrete decision variables  $u_1(t)$  at each stage of each stage are mapped one-to-one with the nodes in the construction diagram 1, it can have

$$\varphi: u_1(t) \in U_d(t) \leftrightarrow v_1^t \in V_1 \quad (6)$$

From the above mapping a hierarchical construction diagram can be derived as shown in Fig. 3.

The nodes at each horizontal level in Fig. 1 are defined as decision levels, corresponding to the set of discretization-tolerant decisions at a certain stage, and  $V_t$  is the set of nodes at the  $t$ th level. In this hierarchical graph, the ants originate from the original node and repeatedly select a node in the previous level to form a path  $\{v_0, v_n(0), \dots, v_n(N-1)\}$ ,  $n(t)$  is the number of nodes chosen at stage  $t$ , and the path corresponds to a feasible solution. Since the ants choose the decision values stage by stage to construct a feasible solution to the problem, it is only necessary to define the connection between adjacent layers, i.e.  $(v_i^t, v_j^{t+1})$ , and define the connection cost as

$$\text{Cost}(v_i^t, v_j^{t+1}) = [J(t+1) - J(t)] + P(t+1) = \begin{cases} \infty & \text{if } P(t+1) = \infty \\ L[x(t+1), u(t+1), t+1] + P(t+1) & \text{if } P(t+1) = 0 \end{cases} \quad (7)$$



**Figure 3.** The layer structure [12]

If the criterion for the decentralisation allowance of the decision collection at all levels is  $qr$ , then the complete amount of nodes in each layer of the layer structure is  $r$ , and the scale of layer structures increases indexically with  $r$ . This can be seen in Fig. 3. The number of computations required to construct a solution is also enumerated, and at the same time, the ants in each layer are required to select all of the nodes in the layer below them.

### 2.2.2 Search Window Generated and Updated

For the multi-stage continuous decision-making problem involving high-dimensional decision variables, the scale of the phase-by-phase decentralised permissive decision collection will "indicate expansion," making it challenging for ants to make decisions in each phase of the decentralised permission decision collection [11]. As a result, the decision variables of each stage are no longer all mapped as nodes in the layered structural diagram, but only according to a specific optimization rule, the node mapping is performed from the various  $w$  values that allow decision variable  $u(t)$  to be extracted from the decision set  $U(t)$ . The search window for ants in the permitted decision set  $U(t)$  is  $U_s(t)$ , which is made of this  $w$  decision variable, and the number of  $U_s(t)$   $w$  components equals the search width. In each phase, the ants will only search the path on the  $w$  nodes of the  $U_s(t)$  search window.

All links that point to a particular node in the structure diagram of the ants' search are referred to as upstream connections to that node. More information on a node's upstream connection increases the likelihood that it will be chosen by the ants to form a new solution. Since the information rate also reflects the benefits and drawbacks of the solution that comprises the corresponding decision value, this also indicates that the selection of this decision value at this stage is "promising". For continuous decision variables, the search window can be updated using real-number coding inheritance.

Determine the adjustment value of each node (i.e., its corresponding decision value) in the search window after multiple myrtle search iterations  $U_s(t)$ . On the basis of the preceding analysis, node  $G$ 's adjustment value can be defined as the average quantity of data on its upstream connection.

$$G[u_1(t)] = G(v_1^t) = \frac{1}{N_v} \sum_{v \in r^{t-1}} \tau(v, v_1^t) \quad (8)$$

After several iterations of ants, the genetic operation adjusts the search window of ants at various stages according to the information distribution on the connection, on the one hand, so that the ants avoid comparing all of the decision values at each stage when constructing, but preference in the decision areas of each stage according to the formula (8) and the principle of genetic optimization to improve search efficiency; and, after several iterations of ants, the genetic operation adjusts the search window of ants at the final stage according to the information distribution.

### 2.2.3 Measuring Steps

(1) The beginning of the process. In accordance with the formula, the process of determining the possible decision value for each level constitutes a search window and is regarded as a population of genetically optimised individuals.

(2) The researchers match the value of the choice variable in the search window to the structure of the Search layer's relevant layer nodes.

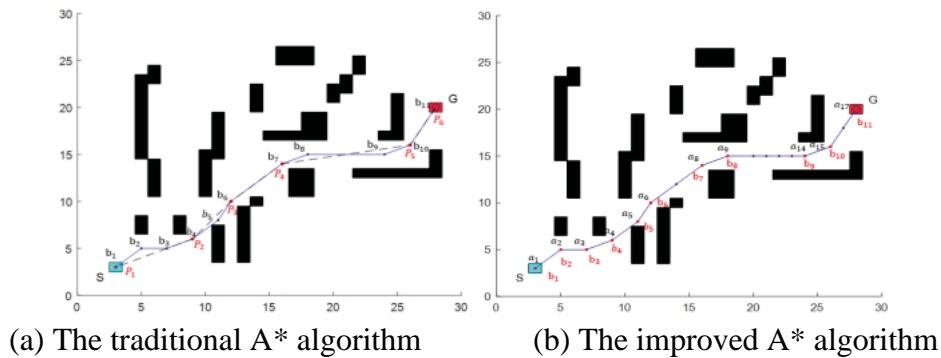
(3) In their quest for ants, studies utilized many search strategies.

The ants group comes up with workable solutions by using layered structures and then changes the quantity of information on each relationship in the layered structure based on how well it matches the requirements.

(4) Investigators perform a variety of genetic methods in order to keep the search window updated at all times.

### 3. Result

#### 3.1. A\* Algorithm Fusion DWA



**Figure 4.** Path planning with different algorithm [10]

Fig. 4 and Table 1 both display the obtained outcomes. As seen in Fig. 4, the updated algorithm eliminates the issue of the robot making sharp, repeated spins over a small area. As compared to Table 1, the optimised method reduces the number of turns by 33.33 percent and the angle by 53.33 percent.

**Table 1.** Analysis of the efficacy of various A\* algorithms

Algorithm	Turning points		Turning curve		Route Distance
	quantity	reduce /%	curve(°)	reduce/%	
Traditional A*algorithm	9	---	4050	---	32.63
Improved A* algorithm	6	33.33%	1890	53.33%	32.27

Table 2 displays the outcomes of a simulation using the revised procedure. As can be seen in Table 2, the total number of route sites is decreased by 64.7% after optimization, while the amount of significant turning points is dropped by 55.6%.

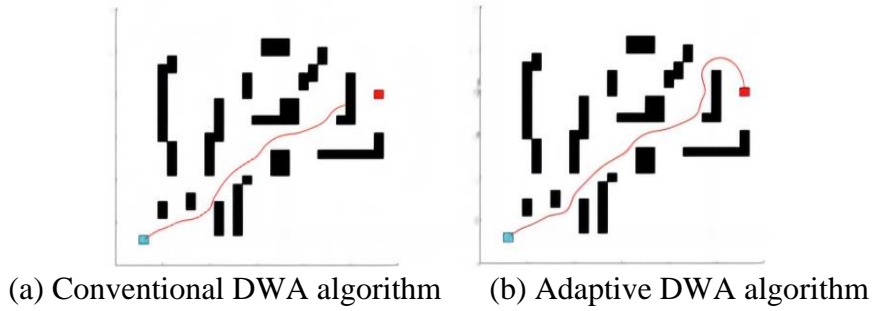
**Table 2.** Analysis of the efficacy of various A\* algorithms

Algorithm	Path point		Turning points		Route Distance
	quantity	reduce	quantity	reduce	
Conventional A* algorithm	17	---	9	---	32.7
Improved A* algorithm	6	64.7%	4	55.6%	32.0

**Table 3.** DWA algorithm parameters

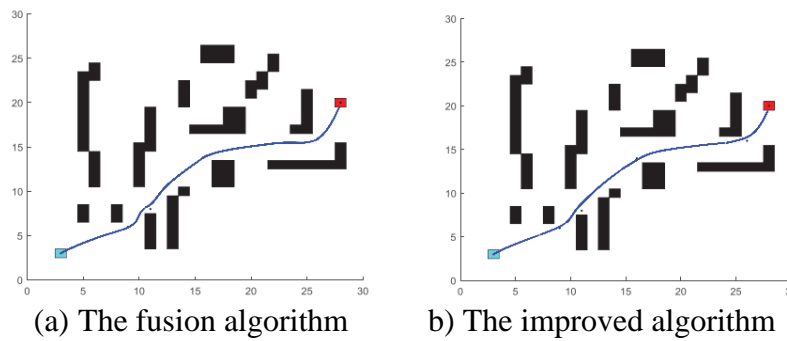
Linear speed resolution	Angular speed resolution	Forward simulation time	Analog time resolution	$\alpha$	$\beta$	$\gamma$
0.023m/s	2.56°/s	1.70 s	0.090 s	0.38	0.11	0.48

As shown in Table 3, in contrast to the conventional DWA algorithm, the adaptive DWA algorithm slows movement in advance and modifies the path it takes through confined spaces by reversing the direction in which it travels. The algorithm manages to sidestep the concave obstruction hazards and arrive at a route that is less dangerous.



**Figure 5.** The results of several algorithm simulations

The purpose of the computer tests was to verify the efficacy of the improved fusing method. Fig.6 and Table 4 display the obtained outcomes.



**Figure 6.** The results of several algorithm simulations

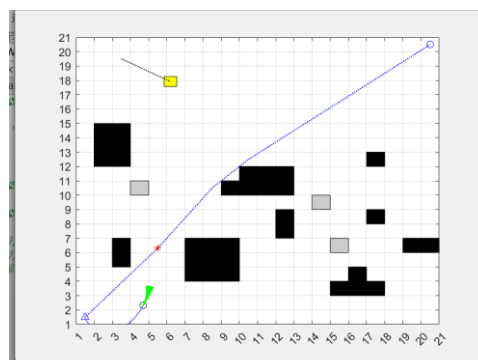
**Table 4.** Evaluation of algorithmic efficiency in relation to route length

Algorithm	Route Distance
Fusion algorithm	32.79
Smoothed fusion algorithm	31.66

It is clear from looking at Fig. 5 and Fig. 6 that the fusing algorithm plots the route with the shortest distance between the starting point and the destination point. This eliminates the issue of the robot having to move around the outside of the obstruction. By maximising the speed at which the robot travels as it approaches the sub-target point, the enhanced fusing algorithm is able to produce a route that is less bumpy. As can be seen in Table 4, the enhanced fusion algorithm has a significantly shorter route length than its predecessor.

### 3.2. Ant Colony Algorithm Fusion DWA

Fig. 7 depicts an environment constructed by the investigators using Matlab code to represent barriers as "1"; the black squares indicate static obstacles, while the yellow squares represent dynamic obstacles. By capturing the trajectory of running code, the shortest path is determined, as shown in Fig. 8. In addition, a curve of changes in line speed, angle speed, and gesture angle, similar to Fig. 9 and Fig. 10, is created by the code compilation when the robot is operating.



**Figure 7.** Creating an environment

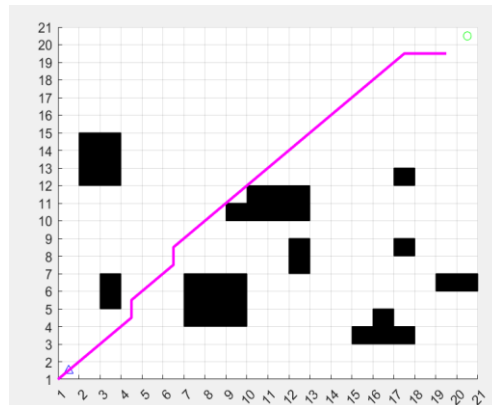


Figure 8. Shortest path

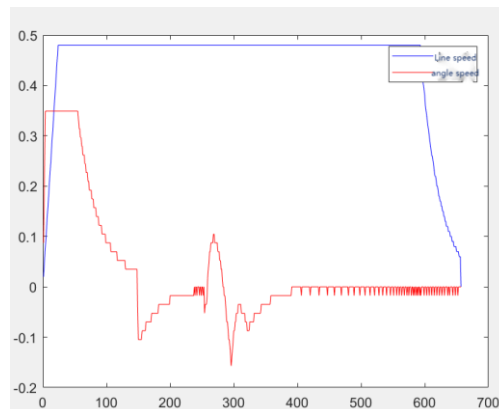


Figure 9. Line speed and angle speed

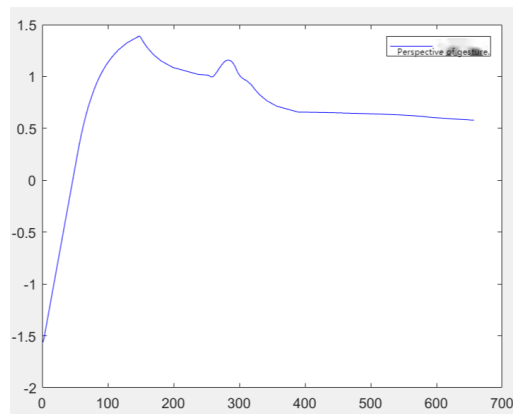


Figure 10. Perspective of gesture

To fine-tune the dynamic characteristics of the robot in its natural surroundings, a unique way of generating dynamic environments is proposed. On unfamiliar terrain, the robot employs sensors mounted on its body to learn about its surroundings and generate an universal position and velocity for dynamic obstacle avoidance based on an upgraded ant colony algorithm [13]. Researchers are considering the dissimilarity between the planned route and the robot's real needs, as well as the limited flexibility of conventional ACO in such settings. To counteract the ant's early myopia, we suggest a non-uniform initial pheromone approach. The ant's path-finding performance in complicated settings is then maximised by employing a smoothing heuristic function with corner suppression and an enhanced retraction strategy. Pheromones are layered in order to solve the ACO's flaws. In addition, they introduce a route smoothing technique to better accommodate the robot's mobility needs in a grid setting.

In order to overcome the conventional DWA's shortcomings in navigation and dynamic obstacle avoidance, investigators employ IDWA to build a robot model constrained by kinematics, and then

use the global route plotted by IACO as the robot's navigational reference. Afterwards, the sample window and evaluation function are modified to improve the robot's route tracking, dynamic obstacle avoidance, and motion stability. Based on the results of the studies, it is evident that the proposed method enhances the robot's dynamic obstacle avoidance in both basic and complex static environments.

#### 4. Discussion

Using a combination of global and local planning, this article addresses the issue of robot path planning in complicated environments and suggests solutions to the drawbacks of both approaches. Simulation tests further confirm the efficacy of the refined algorithm. After examining the experimental data, it could be easy to determine that the linear velocity of the ACO fusion DWA is more consistent during face-to-face encounters than the linear velocity of the A\* fusion DWA. Beginning at the starting point, the linear velocity of the ACO fusion DWA accelerates to a certain speed before beginning a uniform velocity and then decelerating near the focus. On the other hand, the A\* fusion DWA algorithm has a point in the process where the linear velocity approaches zero. This point can be found anywhere in the process. On the basis of the experimental procedure, investigators can make the educated guess that this zero point is the instant when the vehicle comes into contact with the dynamic obstruction and comes to a momentary halt.

In the experiments that were performed back-to-back, the results were even more obvious. When the cart finally caught up with the dynamic obstacle, it stopped, and it was impossible for the cart to reach the target point because there was no data curve produced. Using the ACO fusion DWA algorithm, the vehicle will alter its course to steer clear of the moving obstruction the instant it comes into contact with it, and will continue on its regular journey until it reaches its destination. Finally, let's take a look at how both of these things act in complicated settings. Following our experimental debugging and data selection, both algorithms were able to handle the dynamic obstruction avoidance problem effectively, and the fundamental paths exhibited behaviour that was analogous when placed in the same environment and travelled at the same speed. The two have an inclination angle that is practically identical to one another, which is further evidence that the trajectories are practically equivalent. During the journey, however, researchers discovered that the rotational velocity varied in a certain way. The angular velocity of the ACO tends to be very close to zero at a certain instant, whereas the angular velocity of the DWA is relatively stable. This demonstrates that the DWA algorithm has a faster response time in comparison to the ACO and is able to quickly adjust the attitude angle in order to avoid obstacles when they are encountered.

#### 5. Conclusion

In conclusion, in the straightforward environment of a straight line, an obstruction emerges directly in front of A\* and it is unable to determine whether it should avoid it from the left or the right, whereas the ant colony avoids it directly and continues on its way to its destination. In order for A\* to have a distinct advantage over the ant colony and be able to make route adjustments to the subsequent node, which is more flexible, the route needs to be altered in complicated environments in order to create the shortest path possible. On the other hand, due to the fact that it is determined in terms of objective points, the ant colony moves at a rather sluggish pace. This indicates that A\* is better suited for environments that are more complicated, while ACO is better suited for environments that are more open and straightforward.

#### References

- [1] Gasparetto, A., Boscariol, P., Lanzutti, A., Vidoni, R. (2015). Path Planning and Trajectory Planning Algorithms: A General Overview. In: Carbone, G., Gomez-Bravo, F. (eds) Motion and Operation Planning of Robotic Systems. Mechanisms and Machine Science, vol 29. Springer, Cham.

- [2] Liang Yang, Juntong Qi, Dalei Song, Jizhong Xiao, Jianda Han, Yong Xia, "Survey of Robot 3D Path Planning Algorithms", *Journal of Control Science and Engineering*, vol. 2016, Article ID 7426913, 22 pages, 2016.
- [3] S. Flemming, C.-H. Anders la, and B. Morten, *Configuration Space and Visibility Graph Generation from Geometric Workspaces for UAVs*, Book Section 4, American Institute of Aeronautics and Astronautics, 2011.
- [4] S. M. LaValle, "Rapidly-exploring random trees a new tool for path planning," Tech. Rep. 98-11, Computer Science Department, Iowa State University, Ames, Iowa, USA.
- [5] L. E. Kavraki, P. Svestka, J. C. Latombe, and M. H. Overmars, "Probabilistic roadmaps for path planning in high dimensional configuration spaces," *IEEE Transactions on Robotics and Automation*, vol. 12, no. 4, pp. 566–580, 1996.
- [6] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, pp. 269–271, 1959.
- [7] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proceedings of the IEEE International Conference on Robotics and Automation*, pp. 3310–3317, San Diego, Calif, USA, May 1994.
- [8] C. Petres, Y. Pailhas, P. Patron, Y. Petillot, J. Evans and D. Lane, "Path Planning for Autonomous Underwater Vehicles," in *IEEE Transactions on Robotics*, vol. 23, no. 2, pp. 331-341, April 2007.
- [9] X. Li, X. Hu, Z. Wang and Z. Du, "Path Planning Based on Combaion of Improved A-STAR Algorithm and DWA Algorithm," *2020 2nd International Conference on Artificial Intelligence and Advanced Manufacture (AIAM)*, Manchester, United Kingdom, 2020, pp. 99-103.
- [10] H. Gao, Z. Ma and Y. Zhao, "A Fusion Approach for Mobile Robot Path Planning Based on Improved A\* Algorithm and Adaptive Dynamic Window Approach," *2021 IEEE 4th International Conference on Electronics Technology (ICET)*, Chengdu, China, 2021, pp. 882-886.
- [11] W. Deng, J. Xu and H. Zhao, "An Improved Ant Colony Optimization Algorithm Based on Hybrid Strategies for Scheduling Problem," in *IEEE Access*, vol. 7, pp. 20281-20292, 2019.
- [12] Wen Y, Wu T J. Dynamic widow search of ant colny optimiazation fbr complex multi-stage decision problems, *Proceedings of the 2003 IEEE International Conference on Systems, man and Cybernetics*, 2003, 4091-4097.
- [13] Yang, L.; Fu, L.; Li, P.; Mao, J.; Guo, N. An Effective Dynamic Path Planning Approach for Mobile Robots Based on Ant Colony Fusion Dynamic Windows. *Machines* 2022, 10, 50.