Check for updates

A* Path Planning Algorithm Optimization via Unscented Kalman Filter in Dynamic Environments

Jiazhe Wang

Department of AI Robotics, Sejong University, Seoul, South Korea
wangjz@sju.ac.kr

Abstract. The fast advancement of robot technology in recent years has drawn a lot of attention to path planning algorithms in dynamic situations. The classic A* path planning algorithm faces numerous hurdles because of its limited adaptation in dynamic contexts. These challenges include poor real-time performance, a high risk of planning failure, and problems with the heuristic function's adaptability. This study proposes a method that combines the A* algorithm with the Unscented Kalman Filter (UKF). This method uses UKF for state estimation and obtaining dynamic obstacle predictions, providing prior information for path planning, and ultimately enabling the A* algorithm to reduce the expansion of unnecessary nodes. Simulation experiments show that compared with the original A* algorithm, the improved UKF-A* performs better in planning time and failure rate, reducing planning time by 10%–30% and lowering the failure rate. Preliminary results show that this method has great potential. The optimization approach put forth in this study, which combines path planning algorithms with sensor filtering techniques (UKF), expands the field of conventional path planning research by providing a fresh viewpoint on how to handle path planning difficulties in complex dynamic environments.

Keywords: Path Planning, A* algorithm, UKF, Dynamic obstacles, Obstacle avoidance

1 Introduction

The tremendous advancements in automation and artificial intelligence have drawn more attention to the study of intelligent robot path planning in recent years. Nonetheless, the primary objective of robot research is to identify the most precise and efficient navigation technique in intricate dynamic situations. Finding the best route from a starting point to a destination while avoiding dynamic obstacles effectively and preserving path quality in dynamic situations is the main goal of path planning.

Many path planning algorithms fall into one of three main categories: traditional algorithms (like the A* algorithm [1], Dijkstra algorithm, D* algorithm [2], Rapidly-exploring Random Tree (RRT) [3], etc.), intelligent algorithms (like the Particle Swarm Optimization algorithm, Ant Colony algorithm, Reinforcement Learning,

© The Author(s) 2026
S. Zhang (ed.), *Proceedings of the 2025 International Conference on Electronics, Electrical and Grid Technology (ICEEGT 2025)*, Advances in Engineering Research 292,
https://doi.org/10.2991/978-94-6463-986-5_35

etc.), or a combination of traditional and intelligent algorithms. Traditional path planning algorithms have the advantage of low computational complexity, but in complex dynamic environments, they often fall into local optima and have poor stability. In order to improve the performance of path planning in complex dynamic environments, researchers have introduced state estimation and prediction methods to enhance the stability and real-time performance of traditional algorithms in dynamic environments. Kalman filter [4-7] and Model Predictive Control (MPC) [2, 8, 9] both undertake the tasks of state estimation and prediction. Due to their superior estimation performance in nonlinear systems, Unscented Kalman Filter (UKF) has received considerable attention in recent years [10-12]. Currently, the research on directly integrating UKF into path planning algorithms is relatively scarce, and it mainly belongs to exploratory nature.

This paper proposes an improved A* path planning algorithm combined with UKF (UKF-A*), in order to make up for the insufficient path planning performance of traditional algorithms in complex dynamic environments. In order to provide the A* algorithm with prior information about obstacles and improve the path planning efficiency of the A* algorithm in dynamic environments, this algorithm chooses to use UKF for the state estimation and prediction of dynamic obstacles.

2 Methodology

2.1 Principle of A* Algorithm

Dijkstra's algorithm and greedy search techniques were combined to create the A* algorithm, a traditional heuristic search technique. It estimates the cost of traveling from the current node to the target node using a heuristic function that is developed from Dijkstra's algorithm. The heuristic function improves path seeking performance by efficiently limiting the search area. The heuristic function typically uses either the Manhattan or Euclidean distances to determine the distance between two nodes.

Typically, the A* algorithm employs either a four-neighborhood or an eight-neighborhood expansion method to constantly extend nodes from the initial point towards the target position. To allow path backtracking after the target node is located, the parent nodes of each node are recorded during expansion. The algorithm backtracks from the destination to the starting point after arriving at the target node in order to trace the best route.

Robotic path planning makes considerable use of the A* algorithm. because of its simple logic and simplicity of implementation. It uses a cost estimating function to choose and enlarge nodes. The cost estimating function's formula can be written as follows:

$$f(n) = g(n) + h(n) \tag{1}$$

Where:
$f(n)$: The total cost of node n
$g(n)$: The lowest actual cost from the starting position to node n

$h(n)$: The approximate distance cost between node n and the target node

The following is a breakdown of the A* algorithm's specific steps:

① Initialization: Determine the initial node's total cost $f(n)$) and add it to the open list.

② Path search: Choose the node with the shortest $f(n)$value from the open list to be the current node, then check to see if it is the target node. Path planning is finished if it is. If not, add the current node's neighbors to the open list by expanding them and updating their $f(n)$, $g(n)$ and $h(n)$values appropriately.

③ Path generation: The path planning is finished when the node is expanded to the target location, and the entire path is created by going back to its parent nodes.

The A* method evaluates the cost estimation function $f(m)$ in order to find the best course of action, however it is prone to local optimum and can also produce a less-than-ideal route.

2.2 Principle of UKF

In nonlinear contexts, UKF, a nonlinear state estimation method, outperforms the Extended Kalman Filter (EKF) and Particle Filter (PF) in terms of accuracy and stability [13, 14]. The unscented transformation is the core idea of UKF. The unscented transformation can be understood as not approximating the nonlinear function itself, but instead approximating the probability distribution (mean μ and variance σ^2) of the nonlinear function directly using sigma points, which are a set of deterministic samples used to approximate the posterior probability distribution of the state.

UKF selects sigma points from the Gaussian distribution of the current state and then recalculates the covariance through nonlinear function mapping. This method can obtain a distribution shape closer to the original one compared with EKF, resulting in more accurate prediction and estimation.

The UKF forecasts the mean and covariance of the system state at the next time step by using a process model, just like the linear Kalman Filter does.

Below is the main function introduction of UKF:

Consider a nonlinear system:

State equation:

$$x_k = f(x_{k-1}, w_{k-1}) + w_{k-1}, \quad w_{k-1} \sim N(0, Q) \tag{2}$$

Observation equation:

$$z_k = h(x_k) + v_k, \quad v_k \sim N(0, R) \tag{3}$$

Where:

x_k: system state vector

w_k: input control vector

z_k: observation vector

Q: process noise covariance

Q: observation noise covariance

The specific process of UKF can be divided into the following steps:

① Initialization: Initialization is completed through the system ' s initial state estimation and covariance, and some hyperparameters are set to control the distribution of sigma points;

② Generation of sigma points: Assuming the state dimension is $n, 2n+1$ sigma points will be generated, which are located in the positive and negative directions of the mean;

③ Prediction: The sigma points are substituted into the system model to complete the prediction, obtaining the predicted state value and covariance;

④ Observation: The predicted sigma points are then substituted into the prediction model to obtain the predicted observation value and covariance;

⑤ Update: The Kalman gain is calculated using the previously obtained predicted values and observation values, and the state estimation and covariance are updated to complete one calculation.

2.3 Optimization Method

The UKF-A* method's main idea is to combine state estimation utilizing sensor filtering with the route planning algorithm in order to enhance the A* algorithm's path planning capacity in complicated dynamic situations. Three parts comprise the unique research process: simulation and comparison, state assessment of dynamic obstacles, and filter selection.

① Filter Selection

In the path planning process, the positions of dynamic obstacles are full of uncertainty, which can seriously affect the path quality and planning efficiency. To improve the algorithm's perception and prediction capability of dynamic obstacles, the method of incorporating sensor filtering algorithms is adopted for optimization. First, a suitable sensor filtering algorithm is selected, and comparisons are made among EKF, UKF, and PF. EKF estimates through first-order Taylor expansion, which has low computational complexity, but large error and low accuracy in nonlinear problems; PF estimates through a particle swarm approach, and the number of particles greatly affects the prediction. If there are enough particles, very accurate state estimation can be achieved, but the computational complexity increases and efficiency decreases; UKF estimates the state through the unscented transformation method. Sigma point sampling strikes a fair compromise between accuracy and processing economy while avoiding the issue of computing the Jacobian matrix. Following a thorough comparison, UKF exhibits superior estimating accuracy, efficiency, and stability. Consequently, UKF is chosen as the model for state estimate in this study.

② Dynamic Obstacle State Estimation

The objective of this study is to utilize UKF to provide prior information to the A* algorithm, in order to predict the state of dynamic obstacles. By controlling the node expansion of the A* algorithm and reducing unnecessary node expansions, these methods help the algorithm avoid issues such as path correction and re-computation of node costs when encountering dynamic obstacles. This method can significantly improve the stability and computational efficiency of the algorithm.

③ Comparison and Simulation.

In order to assess the UKF-A* algorithm's performance, this study thoroughly contrasted the A* and UKF-A* algorithms in MATLAB by varying the map size, the number of static obstacles, and the number of dynamic obstacles. The average journey length, planning time, and failure rate are among the comparison and evaluation metrics.

This technology is commonly used in unmanned aerial vehicles [4], autonomous vehicles [2, 9], and robot navigation, among others.

2.4 Simulation Environment

This study uses a two-dimensional grid map for simulation in MATLAB. The map size is adjustable, initially set to 100×100 cells, with each cell representing either a passable area or an obstacle. Two types of obstacles are set on the map: static obstacles and dynamic obstacles. The positions of static obstacles are randomly distributed, and their quantity can be adjusted, such as 500, 2000, 8000, etc., to simulate static building environments of different complexity levels. Dynamic obstacles have initial positions and velocities, and are set to follow a constant velocity linear motion model. Dynamic obstacles are designed to move in both horizontal and vertical directions, and will bounce when encountering the map boundaries, thereby keeping them within the map size. The number of dynamic obstacles can also be adjusted, such as 50, 200, 2000, etc. Fixed starting and target points are set on the map. Simulations are conducted based on the above settings. Figure 1 shows an example of path planning using the A* and UKF-A* algorithms on a two-dimensional grid map.

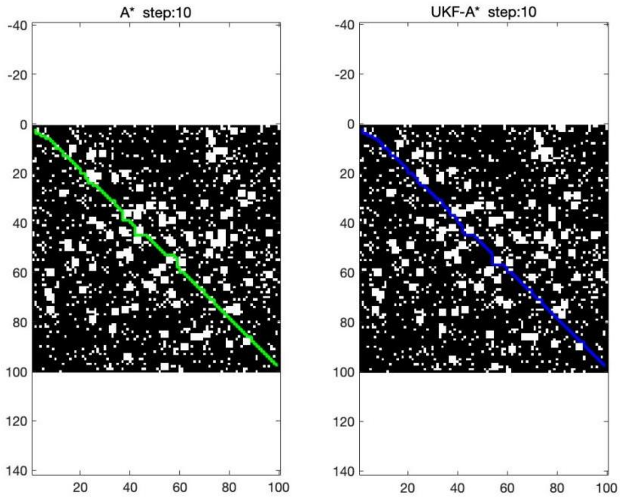

Fig. 1. Example of path planning using A* and UKF-A* algorithms on a two-dimensional grid map

3 Research on the Combination of UKF and A* Algorithm

Papers [10-12] all use Kalman filtering for state prediction of dynamic obstacles, thereby improving the path planning algorithm's responsiveness to dynamic obstacles. In addition, simulations show that the UKF-A* algorithm is significantly better than the original A* algorithm in terms of planning time and success rate. Table 1 and Table 2 show the comparison of average planning time and failure rate, respectively, proving the advantage of UKF in improving path planning efficiency.

This study compares the A* and UKF-A* algorithms in six aspects: map size (grid map resolution), number of static/dynamic obstacles, average path length (the shorter the better), average planning time (ms/time, the lower the better), failure rate (%), and planning time improvement rate (%).

In terms of average path length, Table 1 presents the average path lengths of A* and UKF-A* under different parameters. The results show that UKF-A* performs similarly to the A* algorithm on most two-dimensional grid maps, and even shows slight advantages. For example, in a 200×200 map with 8000 static obstacles and 1000 dynamic obstacles, the average path length planned by UKF-A* is 398.4, slightly shorter than A*'s 399.6. In medium and small-scale maps such as 100×100 and 50×50, the average path lengths of the two algorithms remain almost the same. This indicates that UKF-A* can maintain a path accuracy close to that of A* while achieving more effective dynamic obstacle avoidance, without compromising path quality.

Table 1. The average path length of A* and UKF-A* under different parameters.

Map Size	Representative Configuration	A* Avg Path Length	UKF-A* Avg Path Length	Comparison
200×200	Static: 8000, Dynamic: 1000	399.6	398.4	UKF-A* slightly shorter
100×100	Static: 2800, Dynamic: 200	219.6	219	UKF-A* slightly shorter
50×50	Static: 500, Dynamic: 20	94	94	Same

In terms of planning time, Table 2 presents the planning time of A* and UKF-A* under different parameters. The results show that UKF-A* shows higher planning efficiency. By introducing UKF to predict the state distribution of dynamic obstacles, UKF-A* significantly reduces unnecessary node expansion in multiple scenarios, thereby accelerating node search speed and reducing planning time. For example, in a 100×100 map with 3000 static obstacles and 200 dynamic obstacles, the planning time of UKF-A* is reduced from A*'s 14.82ms to 11.35ms, an improvement of 23%; in a 50×50 map with 700 static obstacles and 50 dynamic obstacles, the planning speed of UKF-A* is even improved by 31%. Even in larger-scale and more obstacle-dense scenarios (such as a 200×200 map with 8000 static obstacles and 2000 dynamic obstacles), UKF-A* can still achieve a slight optimization in planning time. These results reflect the stability and efficiency of UKF-A* in complex environments.

Table 2. The planning time of A* and UKF-A* under different parameters.

Map Size	Configuration	A* Time (ms)	UKF-A* Time (ms)	Improvement
200×200	Static: 8000, Dynamic: 2000	16.45	16.06	Slightly faster
100×100	Static: 3000, Dynamic: 200	14.82	11.35	23% faster
50×50	Static: 700, Dynamic: 50	5.79	3.99	31.07% faster

In addition, Table 3 presents the failure rate of A* and UKF-A* under different parameters. The results show that UKF-A* also shows significant advantages in robustness (failure rate). When facing a large number of dynamic obstacles, the A* algorithm is prone to planning failures, especially in high-density obstacle environments where the failure rate can reach 100%; while UKF-A*, with the help of state estimation and trajectory prediction, greatly improves the planning success rate. In a 200×200 map with 2000 dynamic obstacles, the failure rate of UKF-A* can be reduced to 70%; in medium-scale scenarios such as a 100×100 map with 200 dynamic obstacles, UKF-A* even achieves 100% successful planning. This high success rate in complex dynamic environments indicates that UKF-A* possesses stronger environmental adaptability and stability.

Table 3. The failure rate of A* and UKF-A* under different parameters.

Configuration	A* Failure Rate	UKF-A* Failure Rate
200×200， Dynamic: 2000	100%	70%
100×100， Dynamic: 200	10%	0%
50×50， Dynamic: 50	20%	20%

Overall, compared with the A* algorithm, the overall performance of the UKF-A* algorithm is more outstanding. Firstly, these two algorithms perform equally well in terms of path accuracy; in different scenarios, the length of the paths generated by UKF-A* is similar to that of the A* algorithm, and in some cases even shorter. This indicates that adding a prediction mechanism for dynamic obstacles does not reduce the quality of the path. Secondly, UKF-A* clearly has a speed advantage in planning. By using UKF for state prediction and dynamic obstacle assessment to reduce the expansion of invalid nodes, the planning efficiency has increased by 10% to 30%. More importantly, when dealing with dynamic obstacles, UKF-A* shows stronger adaptability and flexibility. The A* algorithm usually fails in path planning in high-density obstacle situations, while UKF-A* uses state prediction to ensure system stability and significantly reduces the failure rate.

In summary, UKF-A* is suitable for real-world application scenarios that have high requirements for obstacle avoidance capabilities and system stability, as it not only achieves comparable path accuracy to the A* algorithm, but also demonstrates significant advantages in real-time performance and reliability in dynamic environments.

4 Limitations

Even while the UKF-based optimization approach for the A* path planning algorithm works well in intricate dynamic situations, it still has a lot of problems. First off, the UKF-A* technique necessitates that UKF forecast the condition of dynamic impediments, which will raise the program's processing complexity and unavoidably cause major delays in situations requiring fast motion. Second, the accuracy of UKF's prediction outputs is greatly influenced by the values of the model parameters (process noise covariance, measurement noise covariance, etc.). But because sensor noise in real-world settings is unpredictable, employing the wrong parameters may result in poorer prediction accuracy, which will ultimately have a big effect on the algorithm's planning efficiency. This study also performed simulations utilizing a two-dimensional grid map for this technology to confirm its viability. Since this method hasn't been evaluated in a real-world setting yet, more research is necessary to determine its true performance and impact.

To overcome these constraints, future studies will concentrate on the following topics. First, an adaptable dynamic UKF [11] can be included to address the issue of UKF parameter setting and adapt to additional scenarios, or lightweight UKF can be taken into consideration to lower the algorithm's computing complexity. Second, to improve the algorithm's planning capabilities in intricate dynamic circumstances, the UKF model can be combined with other real-time path planning methods (such artificial potential field method, dynamic window technology, etc.). Lastly, to further assess the application performance of our method in real-world scenarios, tests can be carried out in real-world settings with real sensor data and robots.

5 Conclusion

This study investigated the combination of UKF and A* algorithms to improve the A* algorithm's path planning ability in dynamic situations. Adopting the suggested UKF-A* algorithm can greatly increase path planning's resilience and efficiency. By anticipating the state (location and speed) of dynamic obstacles, this approach can gather information beforehand, modify the path's search direction beforehand, and minimize needless node growth. Additionally, simulation findings show that the UKF-A* method outperforms the original A* algorithm in terms of planning time and success rate, with improved stability, especially when obstacles are widely spaced.

This study has practical application value. It has enhanced the efficiency and stability of the robot's path planning in complex dynamic environments. In complex scenarios with high-density obstacles, such as large-scale warehouse storage, intelligent vehicle navigation, autonomous delivery robots, and forest ground patrol, the application prospect of UKF-A* is enormous.

References

1. D. Xiang, H. Lin, J. Ouyang & D. Huang, “Combined improved A* and greedy algorithm for path planning of multi‐objective mobile robot,” Sci. Rep., **12**, 13273 (2022). DOI: 10.1038/s41598-022-17684-0.
2. D. Osmankovic, A. Tahirovic & G. Magnani, “All terrain vehicle path planning based on D* lite and MPC based planning paradigm in discrete space,” in Proc. 2017 IEEE Int. Conf. Advanced Intelligent Mechatronics (AIM), Munich, Germany, 2017, pp. 3. DOI : 10.1109/AIM.2017.8014039.
3. H. Yan & X. Fu, “Dynamic Path Planning of UAV Based on KF‐RRT Algorithm,” in Proc. 2023 IEEE 12th Data Driven Control and Learning Systems Conf. (DDCLS), Xiangtan, China, 2023, pp. 1348–1352. DOI: 10.1109/DDCLS58216.2023.10166695.
4. Z. Wu, J. Li, J. Zuo & S. Li, “Path Planning of UAVs Based on Collision Probability and Kalman Filter,” IEEE Access, vol. **6**, pp. 34237–34245, 2018. DOI: 10.1109/ACCESS.2018.2817648.
5. J. Mei, S. L. Brunton & J. N. Kutz, “Mobile Sensor Path Planning for Kalman Filter Spatiotemporal Estimation,” Sensors, 24, 3727 (2024). DOI: 10.48550/arXiv.2212.08280.
6. J.‐Y. Wang, L.‐X. Xu & Y.‐L. Wang, “Enhanced Hybrid A* Path Planning for Unmanned Surface vessels Using Integrated Kalman Filtering,” in Proc. 2023 IEEE 2nd Industrial Electronics Society Annual On‐Line Conf. (ONCON), South Carolina, USA, 2023, pp. 1–6. DOI: 10.1109/ONCON60463.2023.10430756.
7. F. Marino & G. Guglieri, “Beyond Static Obstacles: Integrating Kalman Filter with Reinforcement Learning for Drone Navigation,” Aerospace, 11, 395 (2024). DOI: 10.3390/aerospace11050395.
8. X. Li, S. Yu, X.‐z. Gao, Y. Yan & Y. Zhao, “Path planning and obstacle avoidance control of UUV based on an enhanced A* algorithm and MPC in dynamic environment,” Ocean Eng., 302, 117584 (2024). ISSN 0029‐8018. DOI: 10.1016/j.oceaneng.2024.117584.
9. Z. Elmi & S. Elmi, “Autonomous Vehicle Path Planning Using MPC and APF,” in Motion Planning, IntechOpen, Jan. 26, 2022. DOI: 10.5772/intechopen.99628 .
10. H. Liu, B. Liang, W. Xu, Q. Mu & J. Yu, “Motion prediction for spinning target and autonomous capture path planning method,” Jilin Univ. J. Eng. Sci., 44(3), 757–764 (2024).
11. Y. Gao, S. Sun & D. He, “A local path planning method based on collision prediction in dynamic environments,” China Mech. Eng., 21, 6 (2009).
12. Z. Song, S. Ji, W. Hu, Z. Ren & L. Wang, “3D path planning for an amphibious robot based on an improved PSO algorithm,” Sci. Technol. Innovation, (004), 000 (2024).
13. Y. Nan, Q. Jian‐Hua, Z. Xue‐Qiong & W. Hong‐Chang, “Dynamic Path Planning Based on Improved Particle Filter Optimisation for Patrol Robots,” in Proc. 2022 7th Asia Conf. Power Electr. Eng. (ACPEE), Hangzhou, China, 2022, pp. 1898–1903. DOI: 10.1109/ACPEE53904.2022.9784034.
14. Y. Gao, S. Sun & D. He, “Global Path Planning of Mobile Robot Based on Particle Filter,” in Proc. 2009 WRI World Congress on Computer Science and Information Engineering, Los Angeles, CA, USA, 2009, pp. 205–209. DOI: 10.1109/CSIE.2009.834

Open Access This chapter is licensed under the terms of the Creative Commons Attribution-NonCommercial 4.0 International License (http://creativecommons.org/licenses/by-nc/4.0/), which permits any noncommercial use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.