Recent Developments in Electronics and Communication Systems
KVS Ramachandra Murthy et al. (Eds.)

doi:10.3233/ATDE221264

Autonomous Robot Navigation Using Fuzzy Inference Based Dynamic Window Approach

Meghana KALAGA[1] and Surekha PANEERSELVAM
Department of Electrical and Electronics Engineering,
Amrita School of Engineering, Bengaluru, Amrita Vishwa Vidyapeetham, Bengaluru, India

Abstract. The autonomous navigation of a mobile robot in an unknown environment depends on the robot's ability to perceive and adapt to that environment. The dynamic window approach (DWA) is a local planning algorithm for obstacle avoidance that considers the motion dynamics of the robot while steering to avoid collisions. The DWA generates a window with all the velocities that are reachable in a time interval and then evaluates all the velocities with an objective function to select the best velocity. In this paper, a novel hybrid algorithm is proposed based on DWA, fuzzy inference system (FIS) and Kalman filter. The proposed work discusses the implementation of DWA algorithm along with FIS to tune the weights of the DWA objective function. Also, the algorithm is extended to incorporate a Kalman filter for state estimation of dynamic obstacles in the environment.

Keywords. Robot navigation, Obstacle avoidance, Dynamic window approach, Fuzzy logic, Kalman Filter

1. Introduction

Autonomous mobile robots in environments such as large warehouses increase the efficiency of the entire picking and stocking process. By completely automating the process, the productivity of the warehouse workers is also improved. The robots first calculate the global path based starting and target points and then incorporates a collision avoidance strategy to avoid any unknown obstacles such as warehouse workers or other robots.

The autonomous robots have various characteristics such as perception, localization, path planning, and motion planning [1]. Perception is the ability of a robot to recognize and understand its surroundings. Simultaneous localization and mapping techniques builds and updates the map of the surroundings when the robot is in motion and also keeps track of the robot’s location in a global co-ordinate system [2]. Path planning algorithms generates a path between the source and target once a map is established [3].

Global path planning (GPP) algorithms have pre-existing information about the environment, obstacle locations, and target position. [4]. Local path planning (LPP) algorithms will navigate through the environment by actively avoiding obstacles until it reaches the goal position [5]. Bug algorithms consider the robot to be a point in space

[1] Meghana Kalaga, E-mail: bl.en.p2ebs20006@bl.students.amrita.edu

mounted with a sensor which detects the boundary of the obstacle [6]. AI-based algorithms have a great ability to handle the uncertainty present in the environment but require large training data [7][8].

The DWA is an obstacle avoidance algorithm that considers robot dynamics and constraints such as limited velocities and accelerations [9]. The DWA was improved as a global DWA to avoid trap conditions with an additional function that calculates the distance from the goal from the robot [10]. Another DWA model with linear regression was proposed that estimates the energy consumed to reach the target [11]. The parameter tuning in the objective function of DWA was achieved using reinforcement learning (RL) [12].

The proposed work aims to address the limitations of constant parameter assignment to the DWA objective function. The parameter tuning is obtained by a simple fuzzy inference system [13] [14]. The FIS in the proposed system takes two inputs, shortest distance to the nearest obstacle and the target azimuth, and generates the parameters to be used in objective function [15]. The proposed work also aims to incorporate the Kalman filter to predict the moving obstacles in the environment [16].

2. Proposed System

The proposed system given in Figure 1 comprises three modules to carry out different functionalities. First, the dynamic window containing the set of admissible velocities is obtained. Then the weights of the scoring functions of DWA are tuned using a fuzzy inference system. A Kalman filter is used to sense the moving obstacles in the unknown environment and eliminate the trajectories that will result in a collision.

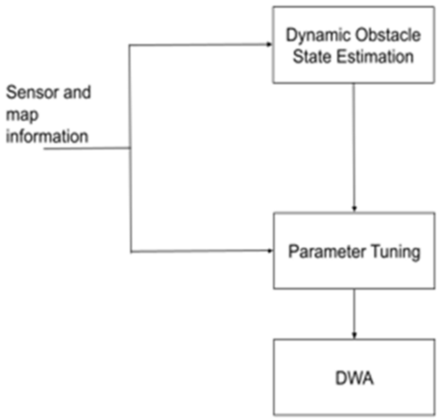


Figure 1. Proposed Model

2.1. Dynamic Window Approach

The DWA is a motion planning algorithm designed for high-speed vehicles. This algorithm relies on the speed control of the robot to avoid any obstacles, unlike position control used by other collision avoidance algorithms. DWA considers dynamic constraints such as linear and angular acceleration limitations of the robot while generating a command to steer the robot.

The motion equations of the robot (ground vehicle) that can move in x and y directions are given by equations (1) and (2):

$$x(tn) = x(0) + \int_0^{tn} v(t) * cos(\theta(t)).dt \quad (1)$$

$$y(tn) = y(0) + \int_0^{tn} v(t) * sin(\theta(t)).dt \quad (2)$$

Where *$(x(t_n), y(t_n))$* is the robot's location co-ordinate at time t_n. *$\theta(t)$* is the robot's orientation angle relative to the target. *v* is the robot's velocity. A dynamic window with set of velocities is obtained with these equations. For each velocity pair in the dynamic window, a circular trajectory is projected and the objective function cost is calculated using (3).

$$E(v, \omega) = \alpha * heading(v, \omega) + \beta * distance(v, \omega) + \gamma * velocity(v, \omega) \quad (3)$$

Where *heading*, *distance* and *velocity* are scoring functions. α, β, γ are the weights of the scoring functions. The trajectory that results in highest cost of the objective function is selected for navigation.

2.2. Fuzzy Inference System

A Mamdani FIS with triangular membership function for input and output is implemented to tune the weights of the scoring functions in DWA objective function in real-time. The system takes two inputs, distance to obstacle and orientation angle and generates three outputs for α, β and γ. The rule set for parameters is given in Table 1, Table 2 and Table 3.

Table 1. Rule set for α

	Distance to obstacle		
Orientation angle	Poor	Average	Good
Poor	LOW	LOW	LOW
Average	LOW	MID	MID
Good	MID	MID	HI

Table 2. Rule set for β

	Distance to obstacle		
Orientation angle	Poor	Average	Good
Poor	HI	MID	LOW
Average	HI	MID	LOW
Good	HI	MID	LOW

Table 3. Rule set for γ

	Distance to obstacle		
Orientation angle	Poor	Average	Good
Poor	LOW	HI	HI
Average	LOW	MID	HI
Good	LOW	LOW	HI

2.3. Kalman Filter for dynamic obstacles

Kalman filter or particle filter uses a sequence of measurements that are taken over a period of time and produces an output estimate of unknown variables. This filter considers environmental noise and other uncertainties also while estimating the output.

In the proposed work, the Kalman filter is implemented to predict the state which contains the position and velocity estimates of the dynamic objects in the environment. The estimated position of the dynamic objects is then used to eliminate the trajectories from the dynamic window set that will result in a collision.

3. Results and Discussion

The system is implemented in python with a 2-dimensional map size of 10mx10m. The minimum and maximum linear velocities are taken as 0m/s and 2m/s respectively. The minimum rotational velocity is taken as -0.7rad/s and the maximum rotational velocity is taken as 0.7rad/s. The maximum linear and rotational accelerations are taken as 0.2m/s2 and 0.5rad/s2. The robot is assumed to be circular with a radius of 1m. The configuration is given in Table 4.

Table 4. Configuration of implemented model

Maximum Linear Velocity	2m/s
Minimum Linear Velocity	0m/s
Maximum rotational Velocity	0.7rad/s
Minimum rotational Velocity	-0.7rad/s2
Time Step	3sec
Maximum Acceleration	0.2m/s2
Maximum angular acceleration	0.5rad/s2

3.1. Dynamic window approach

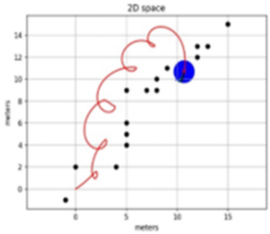
The DWA algorithm was implemented with parameter values as 0.8, 0.1 and 0.1 for α, β and γ respectively. The environment is assumed to be containing only static obstacles. The location of the obstacles is unknown. The path taken by robot is given in the Figure 2(a). As the parameters are constant in the classic DWA, the influence of speed, obstacles and the target are also constant throughout the navigation.

3.2. Fuzzy Inference System

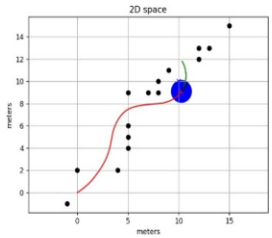
The fuzzy inference system in the proposed model takes two inputs shortest distance to obstacle and heading angle. The parameters heavily depend on values of these two. The response time of proposed fuzzy system is given by few milliseconds. The path taken by proposed system with FIS-DWA is shortened due to selection of an efficient trajectory at each step. The result of FISDWA is demonstrated in the Figure 2(b).

3.3. Kalman Filter

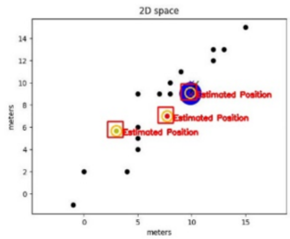
To the classic DWA, Kalman filter is implemented to predict the state of the objects in the environment. The demonstration of estimated position of obstacles and the robot is given in Figure 2(c). The Figure 3 demonstrates the estimated values of position and velocities for moving obstacles in the environment as well as the position estimate for the robot. Once the estimated position of the obstacles is obtained from Kalman filter, they are compared against the list of predicted trajectories of the robot obtained by DWA.



(a) Implementation of classic DWA



(b) Implementation FIS with DWA



(c) Implementation of Kalman Filter.

Figure 2. Implementation of DWA.

```
********The estimated state of obstacle1 is*********
Position (m): (2.95, 5.76)
velocity (m/s):  1
```

(a) Estimation of obstacle 1

```
********The estimated state of obstacle2 is*********
Position (m): (7.52, 7.14)
velocity (m/s):  1
```

(b) Estimation of obstacle 2

```
********The estimated state of Robot is*********
Position (m): (9.76, 9.19)
```

(c) Estimation of robot

Figure 3. State Estimation with Kalman Filter

The classic DWA without tuning of the parameters takes longer time to reach the destination. The proposed system tunes the influence of different scoring functions on the robot. Thus, the robot adapts more to the surroundings by increasing or decreasing its speed as well as the orientation. The proposed system responds faster and resulting in less travel time as given in Table 5. The Kalman Filter reduces the number of trajectories that need to be evaluated by objective function as given in Table 6.

Table 5. Time taken to complete the path

	D(m)	T(sec)	V(m/s)
DWA	20.13	20.56	0.98
Proposed FIS-DWA	14.97	15.53	0.95

Table 6. Number of trajsectories calculated

	Case 1	Case 2
DWA	31	31
DWA with Kalman	29	26

4. Conclusion

The DWA considers the dynamic constraints of the robot to steer its direction, which is essential in high-speed applications. The proposed model with FIS tunes the parameters of DWA objective function in real time. The time taken to complete the path is reduced as the robot picks a more efficient trajectory quickly. Kalman filter is implemented to predict and eliminate the trajectories of these moving objects in the present in the environment. The algorithm can be further improved for low power and low memory applications by aiming at reducing hardware and software overhead.

References

[1] Injarapu AS, Gawre SK. A survey of autonomous mobile robot path planning approaches. In2017 International conference on recent innovations in signal processing and embedded systems (RISE) 2017 Oct 27 (pp. 624-628). IEEE.
[2] Deepika N, Variyar VS. Obstacle classification and detection for vision-based navigation for autonomous driving. In2017 International Conference on Advances in Computing, Communications and Informatics (ICACCI) 2017 Sep 13 (pp. 2092-2097). IEEE.
[3] Varma NP, Vivek A, Pandi VR. Target tracking, path planning and obstacle avoidance by intelligent robot. In2017 International Conference on Technological Advancements in Power and Energy (TAP Energy) 2017 Dec 21 (pp. 1-6). IEEE.
[4] Cheng C, Sha Q, He B, Li G. Path planning and obstacle avoidance for AUV: A review. Ocean Engineering. 2021 Sep 1; 235:109355.
[5] Oroko JA, Nyakoe GN. Obstacle avoidance and path planning schemes for autonomous navigation of a mobile robot: a review. InProceedings of the Sustainable Research and Innovation Conference 2022 Jun 19 (pp. 314-318).
[6] Das SK, Roy K, Pandey T, Kumar A, Dutta AK, Debnath SK. Modified Critical Point–A Bug Algorithm for Path Planning and Obstacle Avoiding of Mobile Robot. In2020 International Conference on Communication and Signal Processing (ICCSP) 2020 Jul 28 (pp. 351-356). IEEE.
[7] Nair RS, Supriya P. Robotic path planning using recurrent neural networks. In2020 11th International Conference on Computing, Communication and Networking Technologies (ICCCNT) 2020 Jul 1 (pp. 1-5). IEEE.
[8] Sangeetha V, Krishankumar R, Ravichandran KS, Kar S. Energy-efficient green ant colony optimization for path planning in dynamic 3D environments. Soft Computing. 2021 Mar;25(6):4749-69.
[9] Fox D, Burgard W, Thrun S. The dynamic window approach to collision avoidance. IEEE Robotics & Automation Magazine. 1997 Mar;4(1):23-33.
[10] Maroti A, Szaloki D, Kiss D, Tevesz G. Investigation of dynamic window-based navigation algorithms on a real robot. In2013 IEEE 11th international symposium on applied machine intelligence and informatics (SAMI) 2013 Jan (pp. 95-100). IEEE.
[11] Henkel C, Bubeck A, Xu W. Energy efficient dynamic window approach for local path planning in mobile service robotics. IFAC-PapersOnLine. 2016 Jan 1;49(15):32-7.
[12] Chang L, Shan L, Jiang C, Dai Y. Reinforcement based mobile robot path planning with improved dynamic window approach in unknown environment. Autonomous Robots. 2021 Jan;45(1):51-76.
[13] Paneerselvam S. Role of AI and Bio-Inspired Computing in Decision Making. Internet of Things for Industry 4.0. 2020:115-36.
[14] Sangeetha V, Ravichandran KS. A Modified Fuzzy A* Based Inference System for Path Planning in an Unknown Environment. In2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI) 2018 May 11 (pp. 181-186). IEEE.
[15] S. Sumathi, L. Kumar, A., and P. Surekha, Computational Intelligence Paradigms for Optimization Problems Using MATLAB/SIMULINK. CRC Press, 2015.
[16] Taylor LE, Mirdanies M, Saputra RP. Optimized object tracking technique using Kalman filter. arXiv preprint arXiv:2103.05467. 2021 Mar 7.