Robotica (2022), **40**, pp. 329–347
doi:10.1017/S0263574721000588

CAMBRIDGE UNIVERSITY PRESS

RESEARCH ARTICLE

Agoraphilic navigation algorithm in dynamic environment with obstacles motion tracking and prediction

H. S. Hewawasam* M. Yousef Ibrahim, Gayan Kahandawa and T. A. Choudhury

School of Science, Engineering and Information Technology, Federation University Australia, Gippsland, Australia
*Corresponding author. Email: h.hewawasam@federation.edu.au

Received: 18 March 2020; **Revised:** 19 April 2021; **Accepted:** 20 April 2021; **First published online:** 28 May 2021

Keywords: agoraphilic algorithm, path planning, object tracking, mobile robot navigation, artificial potential field, moving obstacles

Abstract

This paper presents a new algorithm to navigate robots in dynamically cluttered environments. The proposed algorithm uses basic concepts of space attraction (hence the term Agoraphilic) to navigate robots through dynamic obstacles. The new algorithm in this paper is an advanced development of the original Agoraphilic navigation algorithm that was only able to navigate robots in static environments. The Agoraphilic algorithm does not look for obstacles (problems) to avoid but rather for a free space (solutions) to follow. Therefore, it is also described as an optimistic navigation algorithm. This algorithm uses only one attractive force created by the available free space. The free-space concept allows the Agoraphilic algorithm to overcome inherited challenges of general navigation algorithms. However, the original Agoraphilic algorithm has the limitation in navigating robots only in static, not in dynamic environments. The presented algorithm was developed to address this limitation of the original Agoraphilic algorithm. The new algorithm uses a developed object tracking module to identify the time-varying free spaces by tracking moving obstacles. The capacity of the algorithm was further strengthened by the new prediction module. Future space prediction allowed the algorithm to make decisions considering future growing/diminishing free spaces. This paper also includes a bench-marking study of the new algorithm compared with a recently published APF-based algorithm under a similar operating environment. Furthermore, the algorithm was validated based on experimental tests and simulation tests.

1. Introduction

Robots play an important role in manufacturing, process automation, surveillance, transportation, underground mining, and rescue missions, etc. There are situations where humans cannot reach targets due to danger or terrain conditions. Therefore, mobile robots can be very useful in these types of situations. However, robots path planning and navigation entails a great challenge [1]. This has sparked the interest of many researchers to develop algorithms and frameworks for mobile robot navigation and path planning.

The objective of a path planning algorithm is maneuvering a robot either to reach a predetermined goal or to follow a predetermined trajectory as closely as possible in a collision-free safe motion [2]. The intricacy of the task depends on the robot's environment. A number of researchers have studied the problem of path planning, and a number of path planning methodologies have been proposed. Some of the classical methodologies which can be found in the literature are: (i) Roadmap [3], (ii) Artificial Potential Fields [4, 5], (iii) Cell Decomposition [6], and (iv) Mathematical Programming [7]. Examples of some heuristic methodologies are: (i) Probabilistic Roadmaps [8], (ii) Stigmergy [9], (iii) Rapidly Exploring Random Trees [10], (iv) Tabu Search methods [11], (v) Ant Colony Optimization [12], (vi) Particle Swarm Optimization [13], (vii) Neural Networks [14], and (viii) Fuzzy logic [15].

© The Author(s), 2021. Published by Cambridge University Press.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

In 1985, Khatib [16] introduced Artificial Potential Field (APF) concept. In this concept, robot is modeled as a particle in the space, while goal and obstacles are modeled as charged particles. The goal creates an attractive force on the robot, and obstacles create repulsive forces. The robot is pushed and pulled by these attractive and repulsive forces until it reaches the goal [17]. Many researchers were attracted by APF-based methods because of its simplicity [18, 19]. However, that method has several well-documented key problems [2] such as:

1. The field may contain local minima, especially when there are many obstacles in the environment. This can trap the robot, as a gradient-descent algorithm cannot escape from a local minimum.
2. The robot may find itself unable to pass through small openings such as through doors.
3. Robot may exhibit oscillations when moving through corridors.
4. The robot may be guided away from the goal by a moving obstacle which creates a moving local minimum.

Those well-documented challenges have motivated researchers to find solutions to improve the APF algorithm. These have included offsetting local minima by introducing virtual objects [2]; Vector field histogram techniques [20]; Dynamic window approach [21, 22]; Rule-based Methods [23, 24]; Rules learning techniques [25]; etc.

In addition, Yamashita [18] has proposed a methodology of addressing local minima problem in APF and Sfeir et al. [19] have developed a methodology to reduce oscillations in navigation. Attempts were also made to APF-based methodology to be capable of navigating robots in dynamic environments and minimize its drawbacks [26, 27]. Montiel et al. [26] have proposed a methodology based on bacterial potential field. In this methodology, animal motion attributes have been integrated with APF concepts [27] to minimize the drawbacks of APF algorithm.

All the above APF-based methods used an attractive force created by the goal as well as repulsive forces created by obstacles to navigate a mobile robot toward the goal. A novel optimistic navigation algorithm was introduced by Ibrahim and McFetridge [28] which is known as Agoraphilic Algorithmİ is fundamentally different from the APF method. This is a space-attraction algorithm which generates its forces by available free space. Therefore, this algorithm uses only one attractive force to navigate the robot toward the goal instead of having attractive and repulsive forces. This concept has allowed the Agoraphilic algorithm to overcome most of the inherent drawbacks of APF algorithm [2, 28]. The original Agoraphilic algorithm managed to navigate the robots successfully in static environments. However, it remains to have limitation in environments cluttered with moving obstacles.

Recent surveys on navigation algorithms show that there are only a few navigation algorithms which can navigate robots in dynamic environments compared to static environments [29, 30]. According to the survey conducted by Patle et al. [30], less than 15% of navigation algorithms are capable of navigating robots under dynamic environments (about 200 research papers have been used in this study). Most of the navigation algorithms fail to navigate robots in dynamic environments because they do not efficiently take the velocities of moving obstacles into decision-making. This is the main limiting factor of the original Agoraphilic algorithm in static environments.

Jaradat et al. [31] propose a navigation algorithm which is capable of navigating robots under dynamic environments. This algorithm uses Q-learning technique to build the algorithm. The algorithm needed a limited number of training before using it in an unknown environment. However, the proposed algorithm in ref. [31] does not have its own tracking system. Therefore, it assumes that velocity and position of moving objects are completely known. Transferable belief method is used in ref. [32]. The proposed system is also capable of navigating robots under dynamic environments. The algorithm has its own system to identify dynamic obstacles. Furthermore, this algorithm detects traps and uses virtual targets to resolve local minima issues. Xin et al. [33] also discuss about a navigation algorithm capable of navigating robots in dynamic environments. This algorithm has been developed based on the dynamic window approach. The algorithm in ref. [33] considers the velocities of moving objects when making

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

decisions. Fuzzy-wind Driven Optimization algorithm is used in ref. [34] to develop a path planning algorithm capable of navigating robots in static and dynamic environments. This uses Fuzzy Logic controller with Mandani approach to handle the high uncertainties in dynamic environments. However, the proposed algorithm does not consider velocities of moving obstacles for decision-making which leads to issues in cluttered environments.

The main motivation of this research work is to address the limitation of the original Agoraphilic algorithm which is applicable only to static environment. Hence, the newly developed algorithm ANADE was developed to overcome this limitation while addressing the common issues of other navigation algorithms capable of navigating robots under dynamic environments. The presented algorithm in this paper introduces the novel free-space attraction concept in dynamic environment. The novel algorithm pulls the robot toward the predicted growing free spaces leading to the goal through current free spaces. This paper presents the development of the algorithm and obtained results.

The paper is arranged in the following order: Section 2 describes original Agoraphilic algorithm and its challenges. Section 3 discusses the development of the Agoraphilic navigation algorithm in dynamic Environment. Section 4 presents the investigation results, algorithm bench-marking, and comparison with a modified APF method. The final section discusses the conclusions and remarks of the new navigation scheme.

2. Original Agoraphilic algorithm and challenges

The original Agoraphilic algorithm was a fundamentally different concept from APF-based algorithms as it utilizes Free-Space Force (FSF) concept [28, 35]. These forces are attractive forces created by the free space in robots surrounding environment [35]. These FSFs are shaped toward the goal using a force shaping technique in order to produce one single force that drives the robot toward the goal. Four fundamental stages can be identified in the original Agoraphilic algorithm when used in static environments [35]:

1. Creation of the Free-Space Histogram (FSH).
2. Calculation of the FSFs.
3. Generation of the force shaping coefficients.
4. Summation of scaled force vectors.

However, this algorithm is incapable of navigating robots under dynamic environments. It makes decisions based only on the current location of obstacle. The algorithm does not have a proper way to adapt the effects from moving obstacles to make important decisions in dynamic environments. Therefore, the proposed algorithm in this paper has only taken the free-space concept from the original algorithm and developed the novel algorithm which is capable of navigating robots in static as well as in dynamic environments.

The new algorithm proposed in this paper introduced the novel concept of navigating robots toward the future growing free spaces leading to the goal through current free spaces. A new tracking module is used to accurately estimate the current states (position and velocity) of moving obstacles. It enables the Agoraphilic algorithm to be capable of navigating robots in simple dynamic environments [17]. However, tracking alone is inadequate for complex environments. Agoraphilic algorithm with only object tracking failed to avoid collisions when used in a complex environment cluttered with dynamic obstacles obstructing the Robots path at the same time. Therefore, an algorithm of predicting future objects positions has been developed. The future positions are mapped with respect to global axis system and call them, Future Global Maps (FGM). Main sub-modules run in the prediction module to generate future driving forces. The future driving forces generated from the novel FGM concept influence the Robots current driving force to navigate the robot successfully in the dynamic environments.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

The detailed development of the new navigation algorithm and the obtained experimental results are discussed in the following sections.

3. Agoraphilic Navigation Algorithm under Dynamic Environment (ANADE)

In order to enable a mobile robot to be navigated autonomously in dynamic environment; tracking and position prediction of moving objects had to be developed. The proposed ANADE was also developed while keeping the basic characteristics of the original Agoraphilic algorithm. ANADE consists of the following seven main modules:

1. Object tracking module.
2. Prediction module.
3. FSH generation module.
4. FSF generation module.
5. Force shaping module.
6. Instantaneous driving force component (Fc) generation module.
7. Instantaneous driving force component weighting module.

These modules are repeatedly used in the proposed algorithm as shown schematically in Fig. 1. Initially, the environment sensory data are fed into the tracking module. Consequently, the tracking module creates two outputs: (i) a Current Global Map (CGM) and (ii) states of moving obstacles (position and velocity). Subsequently, the navigation process is divided into the following two main parts, Fig. 1:

1. Instantaneous driving force component generation for the CMG (Fc_0).
2. Prediction: Instantaneous driving force components generation for the FGMs (Fc_1, Fc_2,..., Fc_N).

Those two parts of the navigation process can be further explained as follows:

(1) Instantaneous driving force component (Fc_0) generation for the Current Global Map (CGM): In this process, the generated CGM data are fed into the FSH generation module. The generated FSH is subsequently used by a FSF generation module to create the set of FSFs corresponding to the CGM. The generated FSFs are focused toward the goal using the force shaping module. Then, the shaped forces are fed into an instantaneous driving force component generation module. This module produces one single resultant force (F_{c1}), Fig. 1.

(2) Prediction: Instantaneous driving force components (Fc_1, Fc_2,..., Fc_N) generation for the Future Global Maps (FGMs): In this process, the second output of the tracking module, the current states of moving obstacles, is fed into the prediction module which outputs N number of FGM, where N is the final iteration of prediction. Each of the FGM becomes an input of a FSH generation module. This process returns N number of FSHs. The generated FSHs are fed into N different FSF generation modules. Outputs of FSF generation modules are send through force shaping modules to get FSFs focus toward the goal. Then, the N number of shaped FSFs is fed into N different instantaneous driving force component generation modules to produce instantaneous driving force components (Fc_1, Fc_2,..., Fc_N). Each of the Fc corresponds to each of the FGMs as shown in Fig. 1.

Finally, all the instantaneous driving force components (Fc_0, Fc_1,..., Fc_N) related to the CGM and FGMs are fed through the instantaneous driving force component weighting module to get the final instantaneous driving force, Fig. 1. This instantaneous driving force becomes the actual driving force of the robot in the current iteration.

This whole process occurs in each iteration until the robot reaches its goal. The development and contents of each of the above-mentioned modules are explained in the following subsections.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

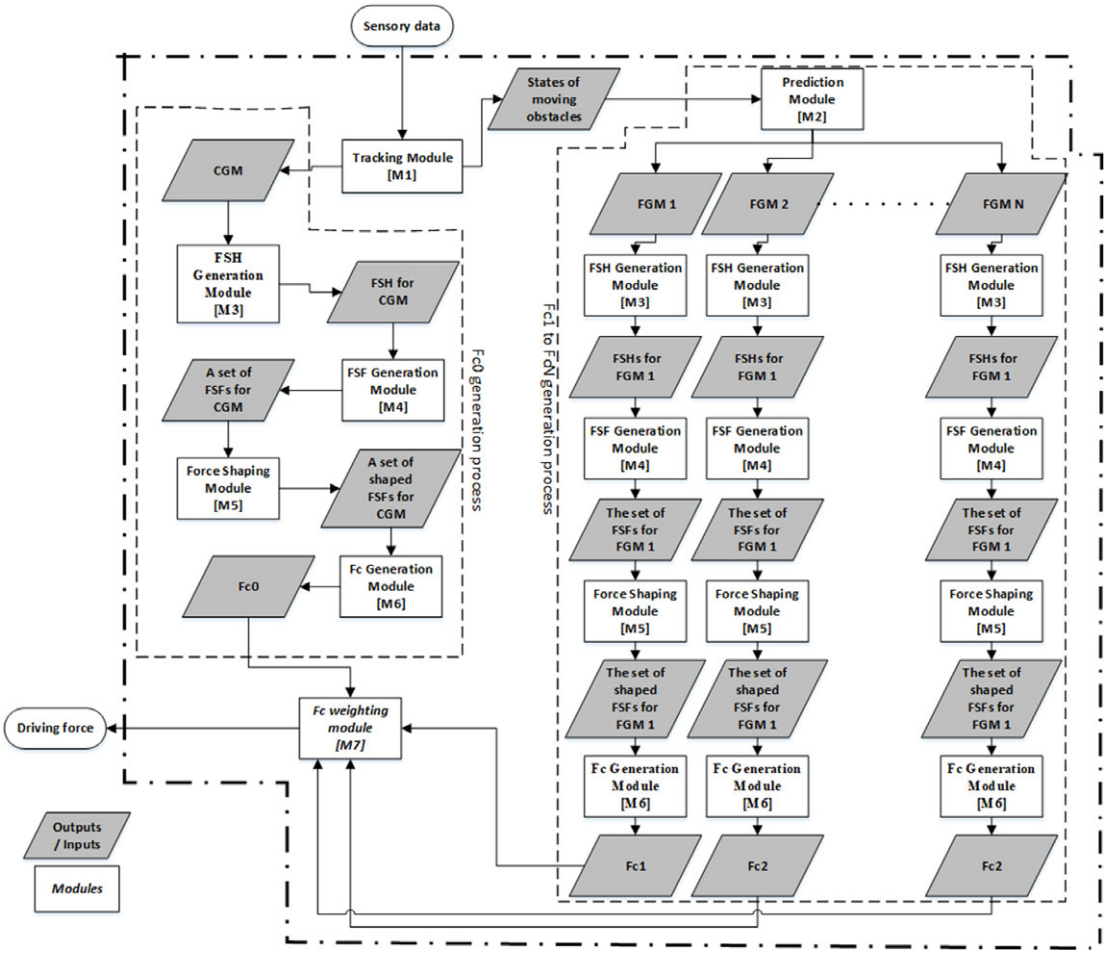


Figure 1. *Schematic diagram of Robot's driving force generation of the new Agoraphilic Algorithm.*

3.1. Pseudocode

Step 1: Start the algorithm
Step 2: Sensory data are taken and fed into the tracking module
Step 3: The tracking module estimates the location and velocity of moving objects
Step 4: Based on the current locations of moving obstacles (one output of tacking module) and locations of static obstacles (from sensory module), the CGM is generated.
Step 5: Based on the estimated locations and velocities of moving obstacles, location of static obstacles and robots states (location and velocity), surrounding the robot in future iterations, are predicted by prediction module. The predicted environments are mapped and called FGMs. After step 5, there is one CGM and multiple FGMs. Each of these maps goes through steps 6–9
Step 6: The free-space passages in a global map are identified by the FSH generation module. Also converted the global map into a FSH, Fig. 2.
Step 7: The generated FSH is converted to a set of FSFs by the FSF generation module.
Step 8: The force shaping module focuses the FSFs toward the goal.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

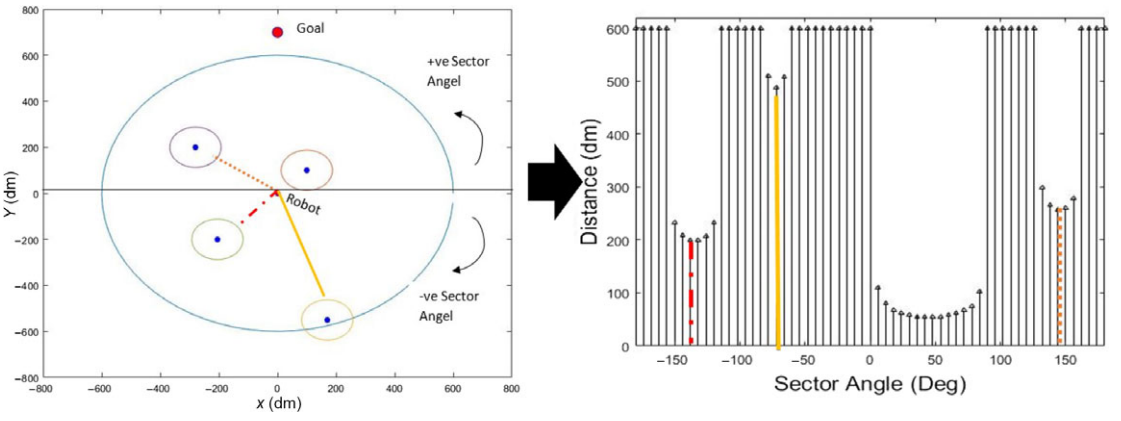


Figure 2. *FSH for a simple environment.*

Step 9: The set of shaped forces is fed into the instantaneous driving force component generation module. This module produces one single force for the particular global map known as instantaneous driving force component (Fc, n).

Step 10: All the instantaneous driving force components ($Fc, 1, Fc, 2, Fc, N$) are fed into the instantaneous driving force component weighing module. This creates the instantaneous driving force for the current ideation. This is the robots actual driving force.

Step 11: If robot has reached the goal, go to step 12. If not go to step 2.

Step 12: Stop

3.2. Object tracking module

The function of the tracking module is to identify the time-varying free space by tracking moving obstacles. This module uses sensory data as its input and produces two outputs. One of the outputs is a map which carries information about positions of moving obstacles in the current iteration. This map is the CGM. The CGM is processed to produce a current FSH. The second output gives current estimated states (position and velocity) of moving obstacles. This output is used by the prediction module to predict the future positions of moving obstacles.

The tracking algorithm in this research uses two models: a process model and a measurement model. Kalman Filter (KF) is used to combine these models to estimate the states of moving obstacles. The main reason for using KF is its simplicity and accuracy in tracking slow manoeuvring objects compared with other methodologies [36, 37]. The estimated positions of moving obstacles are combined with positions of static obstacles to generate the CGM [38].

3.3. Prediction module

The main advantage of the prediction module can be realized when multiple moving obstacles challenge the robot at the same time (further discussed in Section 4). The estimation of moving obstacles' future position helps the navigation algorithm to identify future growing, or diminishing, free spaces.

Prediction module takes sensory data and information about current states of moving obstacles from the tracking module. Prediction module uses this information in future iterations and develops FGM. These global maps are the outputs of prediction module (if the prediction is done to $t+n$ iteration, there will be $n-1$ FGMs). Future states (position and velocity) of moving obstacles are predicted using the prediction model described in Eq. (1). The predictions are updated in the next iteration, and new predictions will be made according to updated state.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

The FGM at iteration $N\dot{\text{I}}$ is generated by combining positions of moving obstacles in the Nth iteration with positions of static obstacles. The obstacles position information is then transferred to a global coordinate system. These FGMs are also used for the future FSH creation.

$$\begin{bmatrix} x(t+n) \\ y(t+n) \\ dx(t+n)/dt \\ dy(t+n)/dt \end{bmatrix} = \begin{bmatrix} 1 & 0 & nT & 0 \\ 0 & 1 & 0 & nT \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x(t) \\ y(t) \\ dx(t)/dt \\ dy(t)/dt \end{bmatrix} + \begin{bmatrix} 1/2(nT)^2 \\ 1/2(nT)^2 \\ nT \\ nT \end{bmatrix} \begin{bmatrix} d^2x(t)/dt^2 \\ d^2y(t)/dt^2 \\ d^2x(t)/dt^2 \\ d^2y(t)/dt^2 \end{bmatrix} \quad (1)$$

$x(t) \stackrel{\text{def}}{=}$ position in x direction
$y(t) \stackrel{\text{def}}{=}$ position in y direction
$T \stackrel{\text{def}}{=}$ sample period
$n \stackrel{\text{def}}{=}$ sample number

3.4. Free-Space histogram generation module

FSH generation module takes a global map as its input and creates a robot-cantred polar map. This polar map is known as FSH [35]. This represents the space profile of the Robots surrounding environment, as shown in Fig. 2.

In this module, the occupied cells by obstacles of the global map are identified. Pre-determined safety boundaries are applied to the occupied cells using an enlargement technique [38]. Distances to points of safety boundaries from the centre of the robot are calculated. The global map is then divided into K neighboring sectors. In each sector, the corresponding minimum distance (d_k) to safety boundary is considered. Using the calculated d_k values, FSH is generated, Fig. 2.

The CGM and all the FGMs are sent through a FSH generation module to create current FSHs and future FSHs [38].

3.5. Free-Space force generation module

FSF generation module takes a FSH (current FSH or future FSH) as its input. FSF generation module creates an initial force (F_k) for each sector k of the robot's surrounding environment. Generally, the initial force for a particular sector is directly proportional to normalized sector distance (Ed_k), Eq. (2). However, if the sector distance (Ed_k) is greater than the predetermined $Ed_{\max}$ value, an initial force for the corresponding sector is taken as u_k [35]. These initial forces are attractive forces toward the open space in the general direction of the goal. These initial forces are also known as FSFs, Fig. 3.

CGM, as well as all the FGMs, is passed through FSF generation modules to generate individual sets of FSFs for the CGM and for all the FGMs. If we assume that prediction is made for N iterations, then there will be one CGM and N FGMs.

$$\vec{F}_k = (d_k/d_{\max})^2.\vec{u}_k \quad (2)$$

Where:
$\vec{u}_k \stackrel{\text{def}}{=} [\cos(\theta_k) \sin(\theta_k)]$

3.6. Force shaping module

The force shaping module takes a set of FSFs as its input and focuses them directly to the goal. It uses a weighing system to modify the FSFs. In this process, forces pointing toward the goal get higher weight, while FSFs pointing away from the goal get lesser weights.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

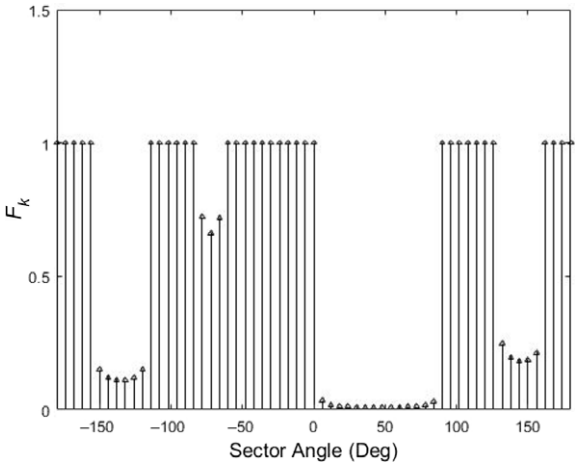


Figure 3. *A set of FSFs created by FSF generation module.*

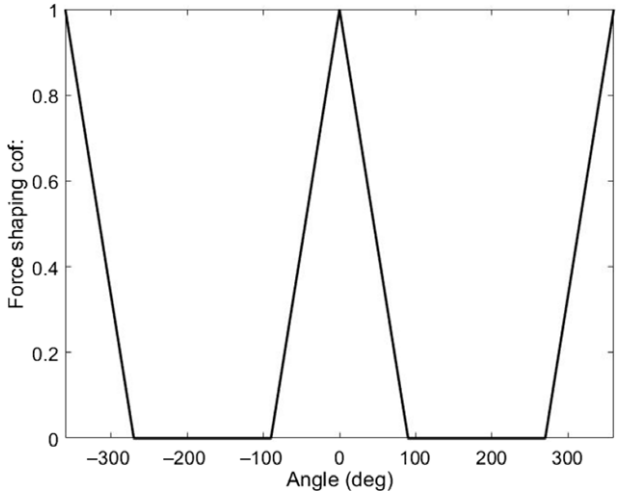


Figure 4. *Force-shaping coefficient.*

Each initial force (F_k) has a relevant force shaping coefficient (δ) which is determined by Eq. (3). A force shaping coefficient can be varied from zero to one, Fig. 4 (becomes one when corresponding Fk directly points to the goal).

All the sets of FSFs related to FGMs and the set of FSFs related to CGM are processed through the force shaping module to point them toward the goal.

$$\delta(\theta) = \begin{cases} \frac{1}{90}\theta + 1, & \text{if } -90 \le \theta \le 0; \\ \frac{-1}{90}\theta + 1, & \text{if } 0 \le \theta \le 90; \\ \frac{1}{90}\theta - 3, & \text{if } 270 \le \theta \le 360; \\ \frac{-1}{90}\theta - 3, & \text{if } -270 \le \theta \le 360; \\ 0, & \text{otherwise.} \end{cases} \tag{3}$$

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

3.7. Instantaneous driving force component generation module

Instantaneous driving force component (F_c) generation module takes a set of the shaped FSFs as its input and generates an instantaneous driving force component (F_c), Eq. (4). In Eq. (4), F_1 to F_K are a set of FSFs and the corresponding force shaping coefficients are denoted by δ_1 to δ_K.

The instantaneous driving force component is derived by taking the vector summation of shaped FSFs and multiplying by a constant scalar (λ) which is defined according to Eq. (5). This keeps the upper bound of F_c to be one which is expected by the instantaneous driving force component weighting module.

In Eq. (5), θ is the angle difference between two sectors. $F_k\text{Sin}(k\theta)$ gives the force component of the F_k (initial force of sector k) in the direction of the goal (the sector on the robot-goal line gets $k\theta$ = 90). When the robot's surrounding is free of obstacles, all initial forces (F_k) become u_k. Therefore, the instantaneous driving force component becomes its maximum ($Fc_{\max}$). To keep the upper bound of instantaneous driving force component (Fc) to be one, λ is defined as Eq. (5).

$$\vec{Fc} = \lambda(\delta_1\vec{F}_1 + \delta_2\vec{F}_2 + \ldots + \delta_K\vec{F}_K) \tag{4}$$

Fc$\Rightarrow Fc_{\max}$ when $F_k \Rightarrow u_k$

Therefore,

$$\vec{Fc}_{\max} = \lambda \int_0^{\pi} \vec{u}_k.\sin(\theta)d\theta$$

$$\vec{Fc}_{max} = \lambda \sum_{k=1}^{K} \vec{u}_k.\ \sin\ (k\theta)$$

but, $Fc_{\max}$ is defined as

$$\vec{Fc}_{\max} = 1$$

Therefore,

$$\lambda = \frac{1}{\sum_{k=1}^{K} \vec{u}_k.\ \sin\ (k\theta)} \tag{5}$$

During one sample, an instantaneous driving force component is produced for CGM and N instantaneous driving force components are produced for FGMs.

3.8. Instantaneous driving force component weighting module and final robots driving force

This module takes all the instantaneous driving forces related to all the FGMs and for the CGM as its inputs. This module uses a weighting system to scale the instantaneous driving forces according to the accuracy of prediction module (highest weight of one is given to the instantaneous driving force corresponding to CGM). The weighted average of instantaneous driving forces is taken as the final driving force of the current iteration which drives the robot.

Performance of the new algorithm was tested by conducting simulation tests under different conditions.

4. Performance testing of the new algorithm

The new proposed algorithm was tested and bench-marked against an APF-based algorithm in a similar dynamic environment. Furthermore, Agoraphilic Navigation Algorithm under Dynamic Environment (ANADE) was tested with and without the prediction module to test the importance of the prediction

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

Table 1. *Object Tracking Module test results.*

Method	Maximum error (cm)	Error variant (cm^2)
Without tracking (measured)	14.63	25.87
With tracking (estimated)	2.57	0.83

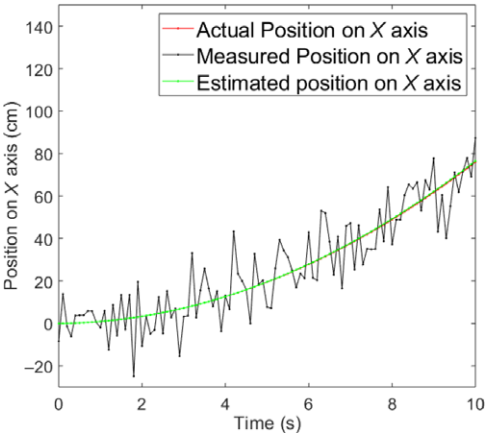


Figure 5. *The recorded actual locations, measured locations by sensory system and estimated locations by the tracking module of a moving object in a single axis.*

module. In this context, two experimental results will be discussed. In addition, the tracking module was isolated and tested to find out its effectiveness.

4.1. Test results for tracking module

In this experiment, a slow maneuvering object was modeled with random acceleration noise. A sensor was modeled with a random noise to measure the position of moving obstacles.

Under the above-mentioned conditions, position of the slow maneuvering object was measured using the modeled sensory system. Also, the same object was tracked using the object tracking module [36]. The object was tracked for 10 s with a sampling frequency of 10 Hz. Three object position data were obtained to test the accuracy of tracking module. They are:

- Actual position of the moving object
- Measured position from sensor
- Estimated position from the tracking module

The recorded data on a single axis (to keep the comparison simple, instead of measuring the actual position on 2D plane positions only on x axis was recorded) are plotted and shown in Fig. 5. The tracking module had a maximum position estimation error of 2.57 cm. The recorded error variance of the testing was 0.83 cm^2, Table 1. An experiment conducted in 2D plane is shown in Fig. 6. In this experiment, initial states of moving object was unknown. At the beginning, tracking module took 2 s to estimate the location of moving object accurately. After that based on tracking module's information, the prediction module predicted the path of moving object with minimal error.

4.2. Test results for ANADE algorithm with and without prediction

In this sub-section, two simulation experiments and one real experiment are discussed. Those experiments were designed to test the algorithm and verify the importance of new prediction module.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

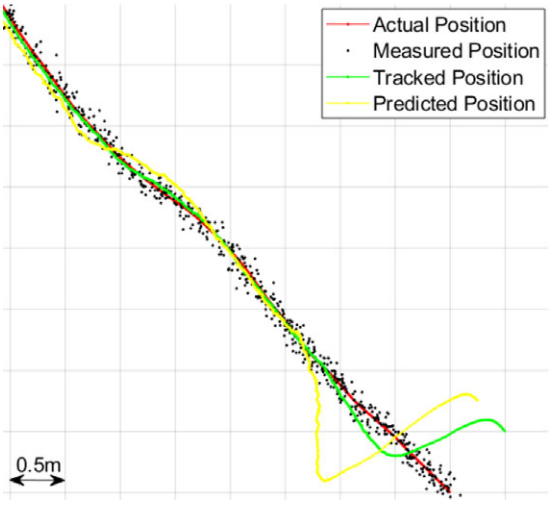


Figure 6. *The recorded actual locations, measured locations by sensory system, estimated locations by the tracking module and predicted locations by the prediction module of a moving object in a 2D plane.*

Each of the two simulation experiments was conducted with prediction and without prediction under exactly same dynamic environments. To conduct the simulation experiments, a special simulation platform was developed using MATLAB. The developed simulation platform allows the user to create and locate static obstacles, goal and moving obstacles in a desired 2D environment. Velocities and accelerations of moving obstacles can be adjusted as needed. Furthermore, user can add random noises to acceleration of moving obstacles. The developed simulation platform also modeled a sensory system to identify the locations of obstacles. The user can vary the noise level of the sensory system. Therefore, the developed simulation platform was capable of generating random dynamic environments.

In the practical experiment, a TurtleBot3 waffle pi (TB3 waffle pi) research robot platform was used for the experimental validation. This is a Robot Operating System (ROS) standard platform. Tb3 waffle pi consists of 360 LiDAR sensor, signal board computer (Robots PC) runs with Ubuntu, OpenCR 32-bit ARM controller, Bluetooth module, and Rasberry Pi camera. The LiDAR sensor was used to capture the Robots environment. The sensor data are passed to a remote PC via the robot PC using a Wi-Fi network. The main algorithm is programmed in the remote PC. The algorithm processes the sensory data and sends appropriate motion commands to the Robots PC. A conducted real-world experiment is also presented in this sub-section (experiment 3).

4.2.1. Experiment 1: Navigation in dynamic environment with three moving obstacles

This experiment was conducted under dynamic environment with three moving obstacles (MOs) and three static obstacles. In this experiment, MOs challenged the robot independently. The experimental conditions are summarized in Table 2.

Figure 7 shows the robots path under those conditions without and with the developed prediction algorithm for obstacles future positions.

(1) Without prediction module: In Area3_2, the robot has engaged with moving obstacle MO2 which moved toward the x-direction. Initially, the robot moved further to $(+x, +y)$ direction because the instantaneous free space in this direction is larger. As a result of this, the robot has moved close to MO2. To avoid the collision, the robot had to stop (Fig. 8) and takes a sharp turn.

(2) With prediction module: In contrast to the above, the algorithm with prediction of MO positions has maneuvered the robot toward the $(-x, +y)$ direction initially. As a result of this decision, the robot

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

Table 2. *Experimental conditions of Experiment 1.*

Object	Initial position (x, y) (cm)	Initial velocity (x, y) (cm/s)
Goal	$(-200, 950)$	(0,0) stationary
Robot	$(-250, -700)$	(0,0)
Moving obstacle 1 (MO1)	$(-500, -200)$	(1.0, 0.6)
Moving obstacle 2 (MO2)	$(-550, 100)$	(1.9, 0.4)
Moving obstacle 3 (MO3)	(250, 200)	$(-5.0, 1.9)$

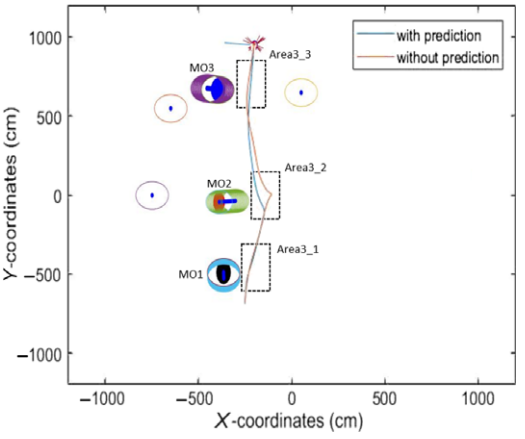


Figure 7. *The recorded paths of the robot with and without the prediction module in experiment 1 (with multiple moving obstacles).*

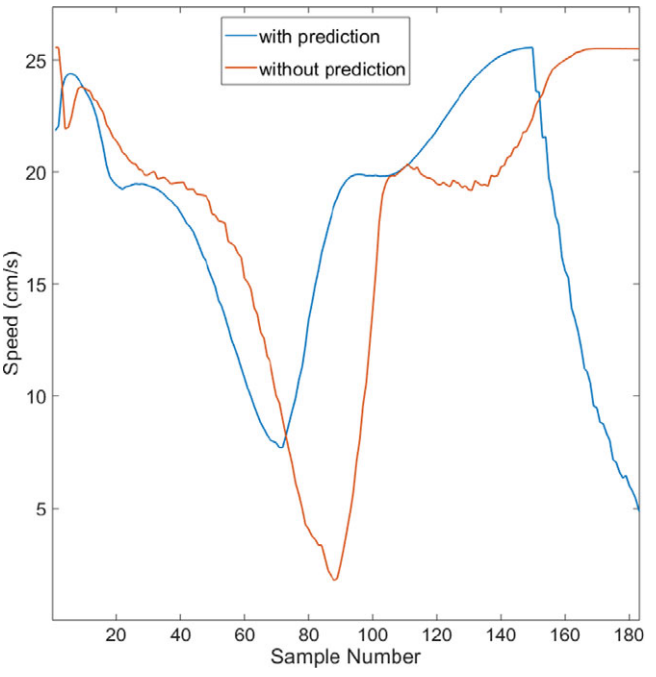


Figure 8. *The recorded robot's speed variations in dynamic environment (experiment 1) with and without the prediction module.*

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

Table 3. *Comparative results.*

Parameters	Without prediction module	With prediction module
Path length (cm)	2456	1666
Time taken (s)	185	165
Avg. speed (cm/s)	13	10
Minimum speed (cm/s)	0	5.7
Max speed (cm/s)	23.5	23.5

Table 4. *Experimental conditions of Experiment 2.*

Object	Initial position (x, y) (cm)	Initial velocity (x, y) (cm/s)
Goal	$(-200, 950)$	(0,0) stationary
Robot	$(-150, -1000)$	(0,0)
Moving obstacle 1 (MO1)	$(-500, -200)$	(1.5, 1.5)
Moving obstacle 2 (MO2)	$(-550, 100)$	(4.3,0.4)
Moving obstacle 3 (MO3)	(250, 350)	$(-2.0, -0.8)$

could pass MO2 without stopping and sharp turns. In Area3_3, the robot with prediction went toward the goal while increasing the speed without changing the direction as it has the knowledge about the future free space.

During this task, the robot with prediction has taken 165 s and traveled around 1666 cm. On the other hand, the robot without prediction has taken 185 s and traveled 2456 cm.

Key Findings:

- The developed algorithm for prediction has allowed the robot to reach the goal quicker by maintaining smooth speed variations (Fig. 8) and taking shorter paths.
- The developed algorithm for prediction has allowed the robot to make prior decisions when it engages with MOs.
- The developed algorithm for prediction has avoided the robot from getting sharp turns.
- The developed algorithm for prediction of MOs position has significantly reduced the path length of the Agoraphilic navigation algorithm in dynamic environment.

From comparative results shown in Table 3, it can be seen that ANADE with prediction module has reached the goal 10% earlier using a 32% shorter path. During the experiment, ANADE with prediction has never stopped the robot in contrast with Agoraphilic algorithm without prediction.

4.2.2. Experiment 2: Dynamic environment with cluttered moving obstacles

This experiment was conducted under conditions mentioned in Table 4 with three static obstacles. The aim of the experiment was testing the behaviour of algorithms in dynamically cluttered complex environments with many MOs challenging the robot at the same time (Fig. 9).

During this experiment, it was observed that at around 75th iteration, two passages (Passage_1, Passage_2) were created by MOs.

Key observations at the 75th iteration ($t = 75$ s):

- Observation 1: Passage_1 has more free space compared to Passage_2
- Observation 2: Free space of passage_1 is increasing, and free space of Passage_2 is decreasing (Fig. 10)

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

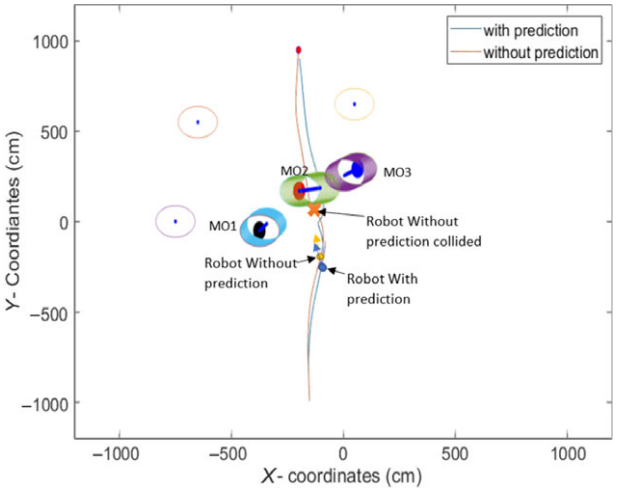


Figure 9. *Effect of prediction on mobile robot path with three moving obstacles challenging the robot at the same time.*

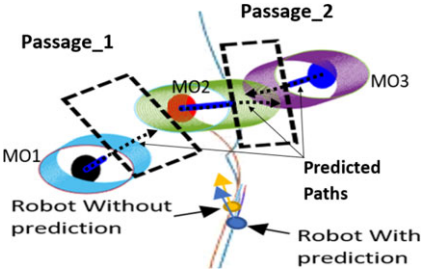


Figure 10. *Robots surrounding at $t = 75$ s with dynamically cluttered environment.*

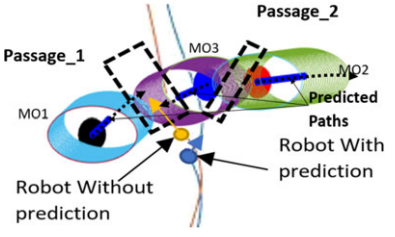


Figure 11. *Robots surrounding at $t = 110$ s with dynamically cluttered environment.*

At around 75th iteration, algorithms with prediction and without prediction have pulled the robot toward the Passage_1. Algorithm without prediction chose this option because of observation 1. On the other hand, Algorithm with prediction chose this option because of the 2nd observation.

At around 110th iteration ($t = 110$ s), states of Passage_1 and Passage_2 have changed,
Key observations at the 110th iteration ($t = 110$ s):

- Observation 3: Passage_1 has more free space compared to Passage_2
- Observation 4: Free space of passage_1 is decreasing, and free space of Passage_2 is increasing (Fig. 11)

At the 110th iteration, algorithm without prediction has kept pulling the robot toward Passage_1 because of observation 3. However, algorithm with prediction chose to change the robot direction and

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

Table 5. *Experimental conditions of Experiment 3.*

Object	Initial position (x, y) (cm)	Initial velocity (x, y) (cm/s)
Goal	(300,0)	(0,0) stationary
Robot	(0,0)	(0,0)
Moving obstacle (MO)	(100,0)	(2, -1)

Table 6. *Results of Experiment 3.*

Parameters	Value
Path length (cm)	475
Time taken (s)	95
Avg. speed (cm/s)	5
Minimum speed (cm/s)	0
Max speed (cm/s)	7

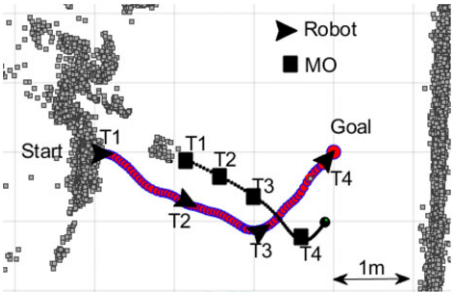


Figure 12. *Robots path observed in experiment 3. Locations of robot and moving obstacle (MO) are shown at four different time-instants (T1-T4).*

move toward the Passage_2, although it has less free space at that moment. This decision was made because of future position prediction of the MO observation 4. As a result of these decisions, the robot controlled without prediction trapped in Passage_1 and collided with MO3. On the other hand, due to the decision made at 110^{th} iteration, the robot with prediction could navigate safely through Passage 2 to the goal.

Key Finding:

- Prediction helps the robot to identify future free spaces.
- In dynamic cluttered environments, the prediction methodology guides the robot away from traps.

4.2.3. Experiment 3: Random real-world experiment

This real-world experiment was conducted under dynamic environment with one MO and one static obstacle. The experimental conditions are summarized in Table 5.

Figure 12 shows the robot's path under those conditions without and with the developed prediction algorithm for obstacles future positions. In Fig. 12, at time instance T1, robot and the MO started moving. The robot detected the static obstacle and turned toward -y direction. At T2, Robots tracking module estimated the MO velocity. With the help of the FGMs (prediction), algorithm realized that the robot can overtake the MO safely in future iterations. Therefore, robot kept moving in the same direction with its maximum speed. At T3, robot changed its direction toward the goal and safely passed the MO. In this experiment, robot could successfully reach the goal without any collisions. During this task, the robot traveled 475 cm in 95 s, Table 6

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

Table 7. *Experimental conditions of algorithm bench-marking.*

Object	Initial position (x, y) (dm)	Initial velocity (x, y) (dm/s)
Goal	$(-100, 100)$	(0,0) stationary
Robot	(0,0)	(0,0)
Moving obstacle 1 (MOB1)	(15, 18)	$(0, -2)$
Moving obstacle 2 (MOB2)	(25, 15)	$(-2, 0)$

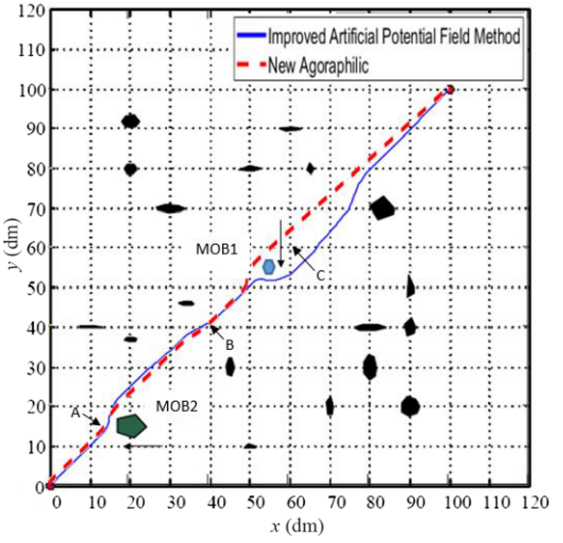


Figure 13. *Comparison of navigation paths under Improved Artificial Potential Field Method and the New Agoraphlic Algorithm(ANADE).*

Key Finding:

- The novel algorithm is capable of navigating robots under dynamic environments in real world.
- In the real-world experiments, robot's maximum speed had to limit to 7 cm/s due to hardware constrain.

4.3. Algorithm bench-marking

This experiment was conducted to compare the ANADE with a recently published Improved Artificial Potential Field Method (IAPFM) [4]. In order to have a valid comparison, we had to adopt the experimental setup used in that method. The experimental environment was modeled according to Table 7.

The developed ANADE algorithm and the IAPFM discussed in ref. [4] were compared under same environment conditions. An experimental setup was constructed to remodel the dynamic environment and the robot path reported in ref. [4], Fig. 13.

The results show that both algorithms were able to navigate robots to the goal safely. It was observed in sections A and B that both algorithms made similar decisions, Fig. 13. However, in section C, robot with ANADE has made a different decision compared to IAPFM. In section C, the robot controlled by IAPFM attempts to avoid the MO1 by moving toward $+x$ direction. In this attempt, the robot could pass MO1 in the front of the path of the obstacle. Because of this decision, the robot was followed by MO1 and has kept pushing it away from the goal (In IAPFM, repulsive forces created by MO1 push the robot

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

toward "y direction). This decision was costly as it has increased the length of robot's path and also there is a risk of collision with moving in front of the obstacle. On the other hand, the robot with ANADE has taken completely different approach to handle this challenge. ANADE has steered the robot to move against the MO1 (in $+y$ direction) and pass MO1. This has allowed the robot to avoid MOB1 in a shorter path and avoided the risk of collision. This human like behaviour has helped the robot to reach its goal faster using a shorter path.

In this experiment, the robot with ANADE has used 10% shorter path compared to the robot with IAPFM. Furthermore, it can be identified that ANADE has made 40% less direction changes.

5. Conclusion

This paper presents a research conducted on developing a new mobile robot navigation algorithm. As demonstrated in the paper, the developed algorithm was successfully proven to navigate mobile robots in complex dynamic environments. ANADE is a FSF-based optimistic algorithm. It uses only attractive forces created by free space in the robots surrounding environment. The new algorithm always drives the robot through the growing free spaces leading to the goal. This new proposed methodology was able to overcome the main limitation of the other algorithms limited to static environments.

Furthermore, the comparative study of the proposed algorithm with and without prediction demonstrated the criticality of the developed prediction module for robots' safe navigation in dynamic environment. The experiment conducted with three MOs has also showed that future space prediction reduced the path distance by 38% and reduced traveling time by 10%.

In addition, the paper also presented a bench-marking study of the developed algorithm against a recently published APF-based navigation algorithm. The research results showed that ANADE has used 10% shorter path with around 40% less direction changes compared to the APF-based algorithm, under the same environment and conditions.

Future works will focus on improving the instantaneous driving force component weighting module. Also, the new algorithm will be tested in the real world in order to further tune the tracking and predication modules for real-world implementations.

References

[1] M. P. Garcia, O. Montie, O. Castillo, R. Sepúlveda and P. Melin, "Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation," *Appl. Soft Comput.* **9**(3), 1102–1110 (2009).

[2] Y. Ibrahim and A. Fernandes, "Study on Mobile Robot Navigation Techniques," *Proceedings of the 2004 IEEE International Conference on Industrial Technology, 2004. IEEE ICIT'04* (2004) pp. 230–236.

[3] H. Akbaripour and E. Masehian, "Semi-lazy probabilistic roadmap: A parameter-tuned, resilient and robust path planning method for manipulator robots," *Int. J. Adv. Manuf. Technol.* **89**(5), 1401–1430 (2017).

[4] W. Siming, Z. Tiantian and L. Weijie, "Mobile Robot Path Planning Based on Improved Artificial Potential Field Method," *Proceedings of the 2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)* (2018) pp. 29–33.

[5] P. Y. Zhang, T. S. Lü and L. B. Song, "Soccer robot path planning based on the artificial potential field approach with simulated annealing; Soccer robots; Soccer robots," *Robotica* **22**(5), 563 (2004).

[6] M. Kloetzer, C. Mahulea and R. Gonzalez, "Optimizing Cell Decomposition Path Planning for Mobile Robots Using Different Metrics," *Proceedings of the 19th International Conference on System Theory, Control and Computing (ICSTCC)* (2015) pp. 565–570.

[7] T. Schouwenaars, B. D. Moor, E. Feron and J. How, "Mixed Integer Programming for Multi-Vehicle Path Planning," *Proceedings of the 2001 European Control Conference (ECC)* (2001) pp. 2603–2608.

[8] J. D. Marble and K. E. Bekris, "Asymptotically Near-Optimal Is Good Enough for Motion Planning," **In:** *Robotics Research: The 15th International Symposium ISRR* (H. I. Christensen and O. Khatib, eds.) (Springer International Publishing, Cham, 2017) pp. 419–436.

[9] R. Johansson and A. Saffiotti, "Navigating by Stigmergy: A Realization on an RFID Floor for Minimalistic Robots," *Proceedings of the 2009 IEEE International Conference on Robotics and Automation* (2009) pp. 245–252.

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

[10] C. Moon and W. Chung, "Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree," *IEEE Trans. Indus. Electr.* **62**(2), 1080–1090 (2015).
[11] W. Khaksar, M. Khaksar, O. Motlagh and T. Hong, "Sampling-based tabu search approach for online path planning," *Adv. Rob.* **26**(8–9), 1013–1034 (2012).
[12] Y. Mingxin, W. Sun'an, W. Canyang and L. Kunpeng. "Hybrid ant colony and immune network algorithm based on improved APF for optimal motion planning," *Robotica* **28**(6), 833 (2010).
[13] H. Mo and L. Xu, "Research of biogeography particle swarm optimization for robot path planning," *Neurocomputing* **148**(1), 91–99 (2015).
[14] K. Pandey and D. Parhi, "Trajectory planning and the target search by the mobile robot in an environment using a behavior-based neural network approach," *Robotica* **38**(9), 1627–1641 (2020).
[15] A. Alomari, W. Phillips, N. Aslam and F. Comeau, "Dynamic fuzzy-logic based path planning for mobility-assisted localization in wireless sensor networks," *Sensors* **17**(8), (2017). https://www.mdpi.com/1424-8220/17/8/1904
[16] O. Khatib, "Real-Time Obstacle Avoidance for Manipulators and Mobile Robots," *Proceedings of the 1985 IEEE International Conference on Robotics and Automation* (1985) pp. 500–505.
[17] H. S. Hewawasam, M. Y. Ibrahim, G. Kahandawa and T. A. Choudhury, "Development and Bench-marking of Agoraphilic Navigation Algorithm in Dynamic Environment," *Proceedings of the 2019 IEEE International Symposium on Industrial Electronics* (2019).
[18] G. Li, A. Yamashita, H. Asama and Y. Tamura, "An Efficient Improved Artificial Potential Field Based Regression Search Method for Robot Path Planning," *Proceedings of the 2012 IEEE International Conference on Mechatronics and Automation* (2012) pp. 1227–1232.
[19] J. Sfeir, M. Saad and H. Saliah-Hassane, "An Improved Artificial Potential Field Approach to Real-Time Mobile Robot Path Planning in an Unknown Environment," *Proceedings of the 2011 IEEE International Symposium on Robotic and Sensors Environments (ROSE)* (2011) pp. 208–213.
[20] A. Babinec, C. Duchon, D. Trung, M. Dekan Z. Mikulová and L. Jurisica, "Vector Field Histogram* with look-ahead tree extension dependent on time variable environment," *Trans. Inst. Meas. Control* **40**(4), 1250–1264 (2018).
[21] P. Saranrittichai, N. Niparnan and A. Sudsang "Robust Local Obstacle Avoidance for Mobile Robot Based on Dynamic Window Approach," *Proceedings of the 2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology* (2013) pp. 1–4.
[22] B. H. Eriksen, M. Breivik, K. Y. Pettersen and M. S. Wiig "A Modified Dynamic Window Algorithm for Horizontal Collision Avoidance for AUVs," *Proceedings of the 2016 IEEE Conference on Control Applications (CCA)* (2016) pp. 499–506.
[23] C. Liu, S. Lee, S. Varnhagen and H. E. Tseng "Path Planning for Autonomous Vehicles Using Model Predictive Contro," *Proceedings of the 2017 IEEE Intelligent Vehicles Symposium (IV)* (2017) pp. 174–179.
[24] C. Son, "Intelligent rule-based sequence planning algorithm with fuzzy optimization for robot manipulation tasks in partially dynamic environments," *Inf. Sci.* **342**(1), 209–221 (2016).
[25] B. Babic, N. Nesic and Z. Miljkovic, "A review of automated feature recognition with rule-based pattern recognition," *Comput. Ind.* **59**(4), 321–337 (2008).
[26] O. Montiel, U. Orozco-Rosas and R. Sepuĺveda, "Path planning for mobile robots using Bacterial Potential Field for avoiding static and dynamic obstacles," *Expert Syst. Appl.* **42**(12), 5177–5191 (2015).
[27] B. Kovaćs, G. Szayer, F. Tajti, M. Burdelis and P. Korondi, "A novel potential field method for path planning of mobile robots by adapting animal motion attributes," *Rob. Auto. Syst.* **82**(C), 24–34 (2016).
[28] M. Y. Ibrahim and L. McFetridge, "The Agoraphilic Algorithm: A New Optimistic Approach for Mobile Robot Navigation," *Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics. Proceedings (Cat. No.01TH8556)* (2001) pp. 1334–1339.
[29] M. G. Mohanan and A. Salgoankar, "A survey of robotic motion planning in dynamic environments," *Rob. Auto. Syst.* **100**, 171–185 (2018). doi: 10.1016/j.robot.2017.10.011
[30] B. K. Patle, G. Babu, A. Pandey, D. R. K. Parhi and A. Jagadeesh, "A review: On path planning strategies for navigation of mobile robot," *Def. Technol.* **15**(4), 582–606 (2019).
[31] M. Abdel, M. Al-Rousan and L. Quadan, "Reinforcement based mobile robot navigation in dynamic environment," *Rob. Comput. Integr. Manuf.* **27**(1), 135–149 (2011).
[32] W. Yaonan, Y. Yimin, Y. Xiaofang, Z. Yi, Z. Yuanli, Y. Feng and T. Lei, "Autonomous mobile robot navigation system designed in dynamic environment based on transferable belief model," *Measurement* **44**(8), 1389–1405 (2011).
[33] J. Xin, X. Jiao, Y. Yang and D. Liui, "Visual Navigation for Mobile Robot with Kinect Camera in Dynamic Environment," *2016 35th Chinese Control Conference (CCC)* (2016) pp. 4757–4764.
[34] A. Pandey and D. R. Parhi, "Optimum path planning of mobile robot in unknown static and dynamic environments using Fuzzy-Wind Driven Optimization algorithm," *Def. Technol.* **13**(1), 47–58 (2016).
[35] L. McFetridge and M. Y. Ibrahim, "A new methodology of mobile robot navigation: The agoraphilic algorithm," *Rob. Comput. Integr. Manuf.* **25**(3), 545–551 (2009).

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press

[36] H. S. Hewawasam, M. Y. Ibrahim, G. Kahandawa and T. A. Choudhury, "Comparative Study on Object Tracking Algorithms for Mobile Robot Navigation in GPS-Denied Environment," *Proceedings of the 2019 IEEE International Conference on Industrial Technology* (2019).
[37] A. Elnagar, "Prediction of Moving Objects in Dynamic Environments Using Kalman Filters," *Proceedings of the 2001 IEEE International Symposium on Computational Intelligence in Robotics and Automation (Cat. No.01EX515)* (2001) pp. 414–419.
[38] H. S. Hewawasam, M. Y. Ibrahim, G. Kahandawa and T. A. Choudhury, "Agoraphilic Navigation Algorithm in Dynamic Environment with and without Prediction of Moving Objects Location," *Proceedings of the 2019 IEEE International Symposium on Industrial Electronics Society* (2019).

Cite this article: H. S. Hewawasam, M. Y. Ibrahim, G. Kahandawa and T. A. Choudhury (2022). "Agoraphilic navigation algorithm in dynamic environment with obstacles motion tracking and prediction", Robotica **40**, 329–347. https://doi.org/10.1017/S0263574721000588

https://doi.org/10.1017/S0263574721000588 Published online by Cambridge University Press