Nat. Volatiles & Essent. Oils, 2021; 8(5): 1726 - 1737

Robot Path Planning System in Populated Dynamic Environment using Dynamic Wave Expansion Neural Network Techniques

[1]Divyendu Kumar Mishra, [2]Dr. Mohd Ashraf, [3]Dr. Anandakumar Haldorai, [4]Dr. Md. Zair Hussain, [5]Dr.Ram Subbiah, [6]Mr. Shanavas TN

1 Department of Computer Science and Engineering, Faculty of engineering, VBS Purvanchal University Jaunpur, Uttar Pradesh, India. divyendu01mishra@gmail.com

2 Associate Professor, Computer Science & Engineering, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). ashraf.saifee@gmail.com

3 Professor (Associate), Department of Computer science and engineering, Sri Eshwar College of Engineering, Coimbatore, Tamil Nadu, India 641202. anandakumar.psgtech@gmail.com

4 Associate Professor, Information Technology, School of Technology, Maulana Azad National Urdu University, Hyderabad (TS). mdzairhussain@gmail.com

5 Associate Professor, Mechanical engineering, Gokaraju Rangaraju institute of Engineering and Technology, Hyderabad ram4msrm@gmail.com

6 Associate Professor, Dept. of Electrical and Electronics Eng., TKM College of Engineering, Karicode, Kollam 691005, Kerala. shanavastn@tkmcee.ac.in

ABSTRACT

In heavily crowded dynamic situations robot navigation faces numerous problems. The path planning approaches for navigation robots in dense situations are presented in this work. In terms of planning and temperament, the route planning of independent mobile robots in the navigation architecture is separated into local path planning and global path planning. About the scope, preparation, and capacity for the implementation, we have presented in this article a neural network model that implements a dynamic version of the route length transformation process (used a stationary domain for path planning). With this new version of path generation extremely dynamic environment is possible. The neural network operates in discrete time, is locally linked, and hence is extremely fast. The planning procedure does not need any early findings of the state of the world. The neural-activity landscape, which creates a dynamically updated topographic map across a distribution representing the robot's arrangement space, is used to generate paths. The network dynamics ensure local adaptations and give stringent criteria for selecting a robot's future route step. With a L_1standards, planned pathways are likely to be optimum, because of these principles. The efficiency of the suggested model is demonstrated by simulating the results inside a set of tests for diverse dynamical conditions.

Keywords: Dynamic Wave Expansion Neural Network, Dynamic Environment, Path planning.

1. INTRODUCTION

Security, verification/acknowledgment, monitoring, process automation, and weapons operations are now possible for agents [1]. This progress has contributed to the advancement of robots with a particular architecture worthy of gathering and analyzing data for a variety of purposes [2, 3], which includes autonomous navigation [4]. In case, if these are not understood or clear, then plan globally or locally for robot navigation [5]. In a situation where planning is domestic and the situation is unpredictable, that method is termed as "reactive" that occurs [6].

The main areas of mobile robot autonomous navigation include behavior decisions, manipulation control and the perception of information. Path planning is the basis of mobile robot navigation and control [7, 8]. The aim of robot route planning is to follow a path that links the present and the destination locations. A smooth route must correspond to the features of mobile robots and the path must be collision-free [9].

Among the most difficult aspects of developing smart robotic systems is giving robots the capacity to plan and travel independently. This capability is especially important for robots that work in dynamic settings with unexpected and abrupt changes. Even when dynamic data changes by the robot's sensing device, the planning process must adjust the path correspondingly. Actual locations that feature human contacts, such as museums, stores, or homes, are good examples. In most cases, a robot's path must be secure (i.e. explosion-free), optimum or proximate to ideal, and natural, i.e. the robot shouldn't get misplaced and wander far from its objective in a complicated setting. When the surroundings are dynamic (i.e. barriers in addition to the goal move), there are two possibilities. When barrier motions are known ahead of time and dynamics, robots couldn't be taken into account (in this case of free-flying items), these issues are simplified to a stationary situation by inserting the time dimension to the planned space (driving obstacle to turn out to be stationary in the new space) [10]. [11-13] are examples of techniques that care for the limitations on the robot dynamics while planning. There are other limited techniques with the most difficult situation, where barrier placements or trajectories are not known before. Hurdles are identified locally throughout the robotic arm and dynamically integrated into the path creation process in this scenario, which increasing applications methods with reorganizing computationally intensive. When the robot hits an obstruction in [14], for example, the whole course is reconfigured from the beginning. Graph-search algorithms developed to use knowledge from past searches to speed up reestablishment [15, 16].

A Dynamic wave expansion neural network (DWENN) is just a new form of a neural network able to produce dynamic range possibilities for real-time navigation in a time probably surroundings, as described in this study. Everything in the above-mentioned sorts of route optimization problems can be solved using this paradigm. The DWENN algorithm's fundamental concept is to arrange wave multiplication similarly to how waves on liquid propagate through a fallen stone. The network's neurons are organized in a lattice that is consistently discredited. When waves of brain activity are generated repeatedly and originate out of the target region a scalar potential field is created. Every succeeding wave carries revised distance knowledge from the goal in addition to raises the power of lattice nodes, causing neurons further away (from goal) to collect higher activity levels. On a certain occasion when a place is not achieved by the actual wavefront, it is said to be untravellable for a given robot.

The following is a breakdown of the paper's structure. The suggested algorithm's main concept is described in section 2 along with a codification of the problem. The neural network model is shown in Section 3 of the paper. Section 4 describes simulation findings, while section 5 concludes with a commentary.

2. BACKGROUND

The route planning may be divided into global and local route planning depending on the frequency of environmental information accessible throughout the whole planning stage [17]. The planning of paths may take many shapes. The methods of path planning may usually be divided into four categories depending on specific methodologies and techniques: template matching, artificial field potential, map generation, and artificial intelligence [18]. Each path planning algorithm has a best-case summary and its own set of restrictions. Mobile robots' present path planning is largely reliant on their surroundings. Aside from the constraints of classical path planning; robots are unable to finish their learning and judgment in complicated settings, which is a major roadblock for the field's growth [19]. As a result, it's critical to design a path planning approach that isn't overly reliant on the surroundings and can fit in fast to changing circumstances.

The Deep Q-Learning Network (DQN) is the main source of functional loss and is a technique of environmental modelling and calculation. The approach evaluated for minimising functional losses using a gradient descent methodology to finish the route plan is the gradient descent technique. Different sampling data are required for education and experience to provide higher generalization capability in the neural network, but an overly big data sample can lengthen the training period [21, 22]. Deep Reinforcement Learning (DRL) has gotten more attention as just a machine learning approach, and it's being used more and more in robot route planning [23]. The agent learns by trial and mistake as it explores the surroundings. The robot acquires information through exploring an area and learning through trial and error. A DRL method offers obvious advantages in route planning and calls for less environmental information in the first place [24, 25].

Fuzzy control is based on human experiences, and its set of principles is hard to adapt for complicated real-world situations. Gharajeh and Jond presented a technique based on a hybrid GPS-ANFIS system [26]. It comprises GPS-based control for worldwide robot navigation towards the target and an ANFIS regulator for localized collision avoidance movement. To adapt to the unpredictable environment, ANFIS incorporates a neural network into a fuzzy system. The suggested algorithm's viability in discrete settings is tested in this study. The disadvantages of neural networks are their extensive training period and slower convergence. As such, Liu et al. developed an optimised particle swarm training neural network approach [27]. This approach was nonetheless intended to optimise speed and resolution rather than complications. Zhang et al. developed and trained a novel deep-set coevolutionary dual-branch neural network (DB-CNN) that enhanced the convergence rate, drawing both from global and local information [28]. Zhang et al. designed and built a novel deeply coevolutionary double branching neural network (DB-CNN) that improves calculation time by extracting the features globally and locally [29]. It is a method to design a worldwide route. Sung et al. developed and evaluated the neural network route planner by utilising two different offline path planning methods. Offline neural network training needs a very large number of data samples which make data gathering difficult and impede self-learning.

3. SYSTEM MODEL

The field waves of brain function are dispersed throughout the neuron linked with the target area in the DWENN network. Then a recent generation develops at each point in time, containing data relating to the distance to the target, i.e. it spreads to a greater activity value for neurons connected to distant locations (Figure 1). Addition and certain binary controls are utilised in the parameter-free updating rule and include whole-value calculation. Consequently, the spread of activity is computationally efficient and enables real-time planning.

A neuron at each step "doubles" the action of a neighbour who is: (i.) directly at the target neuron, (ii) not at the hurdle, and who has (iii) the active neuron (i.e. the neighbour has a preferred activity value) and (iv) the actual neuron (i.e. the current neighbour altered his level of activity in the previous iteration). If the bulk of criteria (i)–(iv) for a neighbour is fulfilled, the associated weight changes to one, while the other neighbours related change weight (or stay) nil.

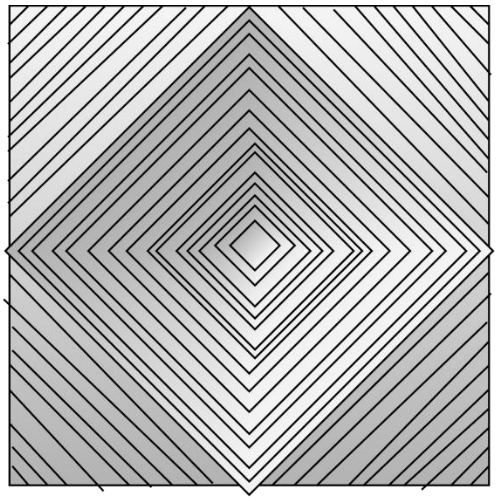

Figure 1: Wave propagation

The activity of each neuron forms a scalar potential field with a minimum positive value already in the target area (in the potential field the zero states are forbidden).The activity of each neuron forms a scalar potential field with a minimum positive value already in the target area (in the potential field the zero states are forbidden).The goal attracts the robot worldwide, and it begins to move immediately as the first wavefront and reaches its beginning point. It can only migrate to the nearby place from where it is inherited from the activity. This makes robot path stages safe, as well as the path, is more likely to be theL_1- optimal.

3.1 NETWORK DYNAMICS

Each neuron i is linked to its set by using the definitions $s_i = \{i_n, \ldots., i_n\}$we presume an unpredictable and (potentially) fixed count for the neighbors and the neuron model is shown in Figure 2.

A discrete-time dynamical system can be regarded as the DWENN model. At time $t + 1$, the activity of neuron i will be determined by the current activities of its neighbors (the vector$\vec{x} = \left(x_{i_1}(t), \ldots, x_{i_n}(t)\right)$). It will be influenced by its neighbor' actions as well as its activity at the precedent time step$\vec{x}_d = \left(x_i(t-1), x_{i_1}(t-1), \ldots\ldots, x_{i_n}(t-1)\right)$. Therefore, if $x_i > 0$, neuron i is active; else, it is inert. All neuron's activity levels and connected weights are set to zero at the start. Let $i * (t)$be the target neuron's index at time t.

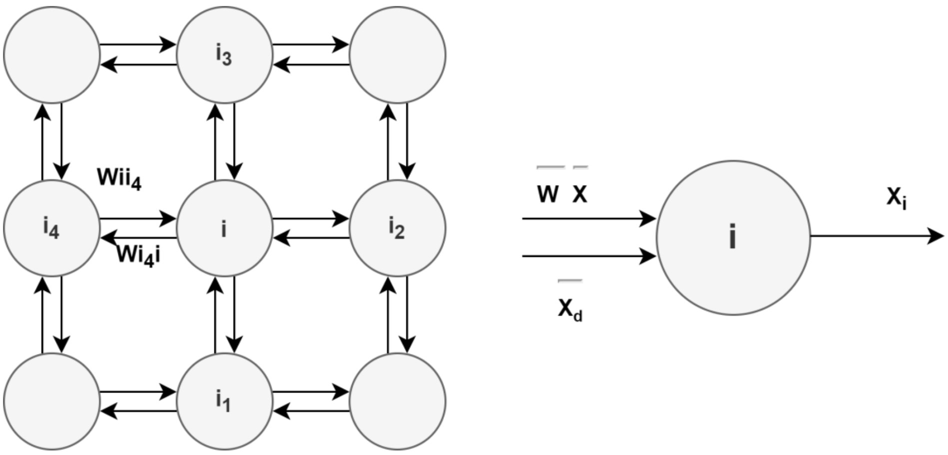

Figure 2: Network Architecture.

There are three types of neurons, each with its own set of dynamics:

I. For a new target $(i = i * (t))$
$x_i(t+1) = 1.$ (1)

II. For the immediate surroundings$(i \in s_i * (t))$
$$x_i(t+1) = \begin{cases} x_i(t) + 1, & \text{if } i_*(t+1) = i_*(t) \\ 2, & \text{otherwise} \end{cases} \quad (2)$$

III. For the rest of the neurons
$x_i(t+1) = \sum_{j \in x} w_{ij}(t)(x_j\,(t) + 2).$ (3)

The corresponding connection weights are defined by updating the activity level of neuron i

$$w_{ij}(t+1) = \begin{cases} \delta_{jk}, & \text{if k} \in s_i\ is\ the\ first\ neuron, \\ & for\ which\ (a) - (d) below\ hold, \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

A Kronecker symbol is δ_{jk}, and needs of neuron k are as follows:

(a) where k is not a hindrance.,

(b) $x_k(t) > 0$, i.e. The component of the action wavefront and contains data about transformation in the work environment,

(c) $x_k(t) \neq x_k(t-1)$, i.e. It is updated with the latest information,

(d) if $\left(x_i(t) + x_i(t-1)\right) > 0$, then $x_k(t) < x_i(t)$ must satisfy, i.e. Its present location is closest to the objective than that of the robots.

It's worth noting that the neighbors of neuron i are taken into account in terms of some predetermined ordering. Equation (1) guarantees that its neuron is at the (possibly motion) target does have the lowest action rate within the neural field throughout all times. Weights update rule (4) stipulates that every neuron has not more than a single connection weight $w_{ij^*} = 1$, where $j^*$is chosen according to rules (a)–(d), and $w_{ij} = 0$ for everyone else $j \in s_i$.

Its weight $w_{ik}(t+1)$ is equivalent to one; brain activity flow that is selected is determined by the rules (a)–(d). The weight of W_{ik} (t+1), equaling to just one if the neuron k has altered its status in the previous step rule (c), if it is active at the time t (rule(b)), is not correlated with a hurdle (rule(a)) and if the neuron I has a reduced level of active activity, that is due to the global distribution of potential, neuron k is actually nearest to the target neuron (d).

DWENN's dynamics Equations (1)–(4) may be used to derive the following characteristics right away:

Property 1: The neurons turn active with the value$x_i(t_a) = (2t_a - 1)$ if the initial wavefront hits the neuron only at time step within a stationary environment.

Property 2: If neuron i has a positive weight, its weight reflects the orientation of the inclined potential field: $w_{ij} > 0 \Rightarrow x_i > x_j$;

Property 3: The doubling number n of network repetitions limits the level of activity of neuron i:$x_i(t) \leq n$;

Property 4: That when an active neuron i becomes inactive in the next time step, it will remain inactive. Indeed, since$x_i(t-1) > 0 \; and \; x_i(t) = 0$ then $x_k(t) < x_i(t)$is consistently false, the condition (d) in (4) is likewise false, and ∀ $j \in s_i : w_{ij} = 0$, and, therefore, $x_i(t+1) = 0$.

In addition, if a target is static:

Property 5: If an activated neuron stays alive at every time step, its activity is raised by one:

$x_i(t) > 0 \Rightarrow x_i(t+c) = x_i(t) + c$;

Property 6: When neuron i and neuron j both changes active at the same moment, then

$t_j > t_{i,}$then$x_j(t) > x_i(t)$ for all $t \geq t_j > t_i$.

If $w_{ij} > 0$, The next route step of the robot is now in a direction proposed by the single non-zero weight of the robot, i.e., the neuron j representing the configuration of the neuron I configuration (or when j becomes the target neuron). If there is no other method to continue the action, a step-back action is also an option. According to Property 2, the robot's route steps are taken in the potential field in a gradient descent direction. As a result, the resultant path is usually optimum in terms of L_1 metrics.

The inclinations to produce routes consisting of straight lines are caused by the pre-selected and constant count of the neighbor. As a result, it is sensible to question the same neighbors from where the activity is the precedent time step was derived first because the likelihood of a fresh wave arriving in this direction is higher.

3.2 IN THE DYNAMIC ENVIRONMENT, OBSTACLE AVOIDANCE, AND GLOBAL ADAPTATION

Because activated neurons can momentarily turn inactive to convey dynamic transformation in dynamic surroundings, a rapid rearrangement of a potential field is feasible (Property 4). In condition (d), this key attribute is encoded (4). Therefore, an inactive neuron may begin to spread an inhibitory wave. This is shown in Figure 3a, showing the initial placement of two barriers.

The neuron beyond the gate (where the black edge is shown) is solely active over the gate, but all its close neighbors are more actively active. As a result, whenever these gates are shut, this neuron goes dormant (Property 4). This causes an inhibitory wave to also be generated, which inhibits neurons beyond the gate one by one until the fresh activity is conveyed by a wavefront crossing through the newest gate location.

The very same inhibitory process allows the robot to dodge dynamic impediments in its path in such a natural manner. It is shown in Figure 3 for a dynamic barrier that appears in time t_k, m route steps ahead of the robot. Because all of its neighbors (save the barrier neuron) has higher activity, condition (d) in (4) is incorrect for them. The neuron i which is between the robot as well as the barrier, in particular adjacent to the obstruction, goes inactive at the time$t_k + 1$. This is comparable to the above-mentioned 'gate' scenario.

According to property 4, the neuron I is sluggish at time, t-k.+2, and one of the neighbors is also inactive. Therefore, two inactive neurons occur at least twice immediately after the barrier arises. It moves far away from the target along the direction of wave head growth. If the robot's position corresponds with any of these inactive neurons following $m/2$steps, the robot halts and pauses at least another time step before continuing navigation at the time $t_k + \frac{m}{2} + 2$ if a fresh wavefront has arrived at its location (Figure 3b). The next stage on the robot track is the number of neurons in the vicinity. Figure 3c illustrates the final route of the robot according to the number of neighbors in Figure 3a.

This inhibitory process distinguishes our model from others like the traditional resistive grid, which requires many samples to overcome local maxima and develops into a solution. Throughout these cycles, oscillatory movements of the robots that are locally directed, are recorded, resulting in artificial, far from optimal 'Hither and Thither' pathways.

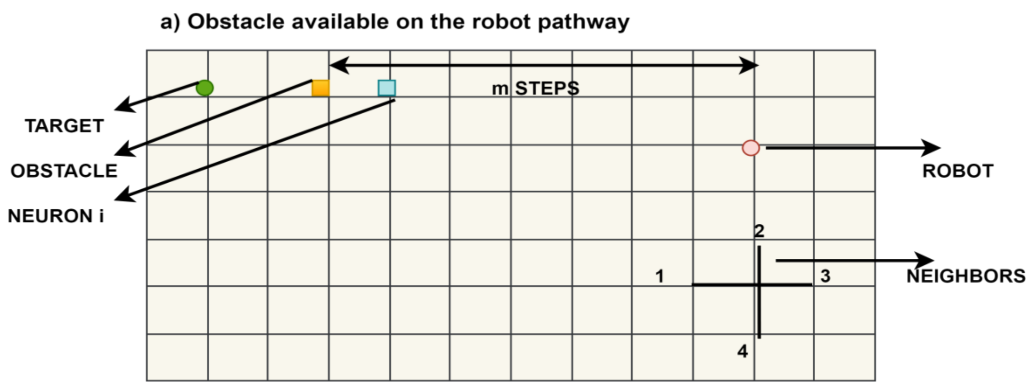

Figure 3a: Obstacle avoidance at $t = t_k$.

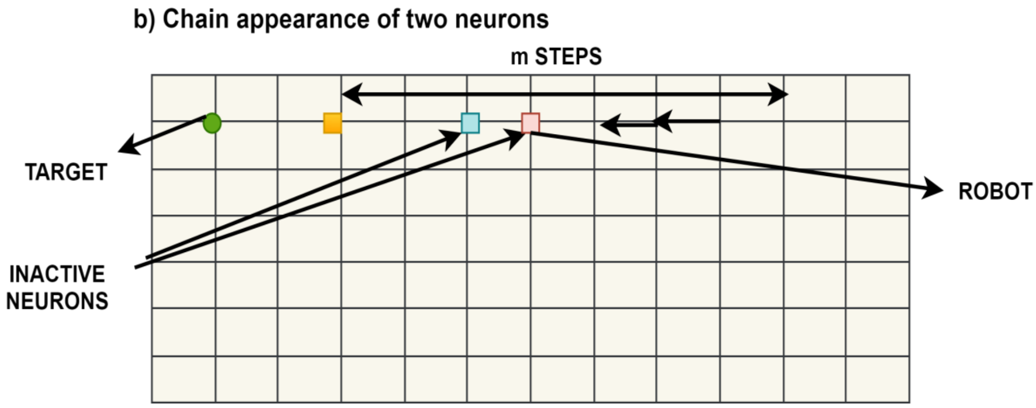

Figure 3b: Obstacle avoidance at $t = t_k + \frac{m}{2}$.

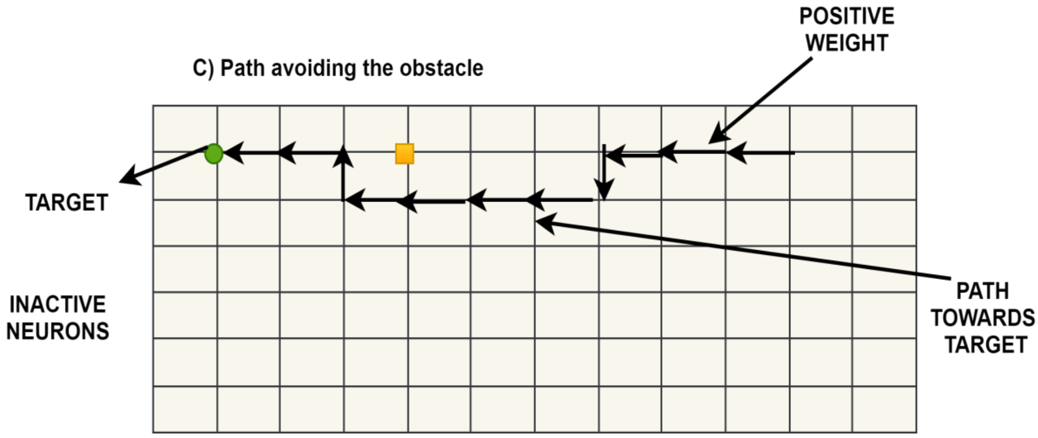

Figure 3c: Obstacle avoidance at $t > t_k + \frac{m}{2} + 1$.

4. EXPERIMENTAL ANALYSIS

We show simulation findings for several sorts of dynamical environmental changes in this section. All of the tests were conducted for a point robot inside a 2D workspace to show the dynamic nature of the proposed method more clearly, although this does not limit the model's broad application. Graphs depict stationary obstacles in the workplace are given in a µ - plots light grey tone, whereas dynamical obstacles are colored black. Continuous curves are used to illustrate the robot's paths. Black squares that come out of nowhere in the workspace represent obstacles that arise at random. The letters SP and TP stand for start and target positions, respectively.

Above the discretized workspace model, we employed a network field of 3721(61×61) neurons for our studies. The workspace's boundaries are regarded as impediments. We picked the labeled adjacent neurons for a deadline extension. As a result, the robot prefers to travel horizontally first wherever possible. The workplace is crowded with static impediments in an open gate condition. The dynamic obstacle begins to travel in the direction of the goal once the robot has taken 50 route steps. It comes to a halt at the spot, leaving a tiny gate open. At each time step, the robot must navigate through the gate while avoiding 20 random barriers that emerge in the workspace.

The starting configuration for this test case in a closed gate condition is quite identical to the "open gate" scenario. The robot goes over the same stages as the human. The moving impediment, on the other hand, shuts the gate before the robot can travel through. The robot subsequently responds to the changes in the surroundings in a dynamic manner. The ultimate path and the activity landscape at the time of achieving the goal are represented.

“Freezing up” is a term used to describe a situation in which Obstacles that changes throughout time: When the robot takes its initial first step throughout this model, the dynamic obstacles begin to travel in the direction of the arrows. Once the obstacles are frozen the robot has already taken 20 path steps, as illustrated in Figure 4. After then, this activity landscape swiftly adjusts to the state of the environment. This resultant activity landscape depicts the reflect organization of the stationary workspace at the time of arriving at the objective.

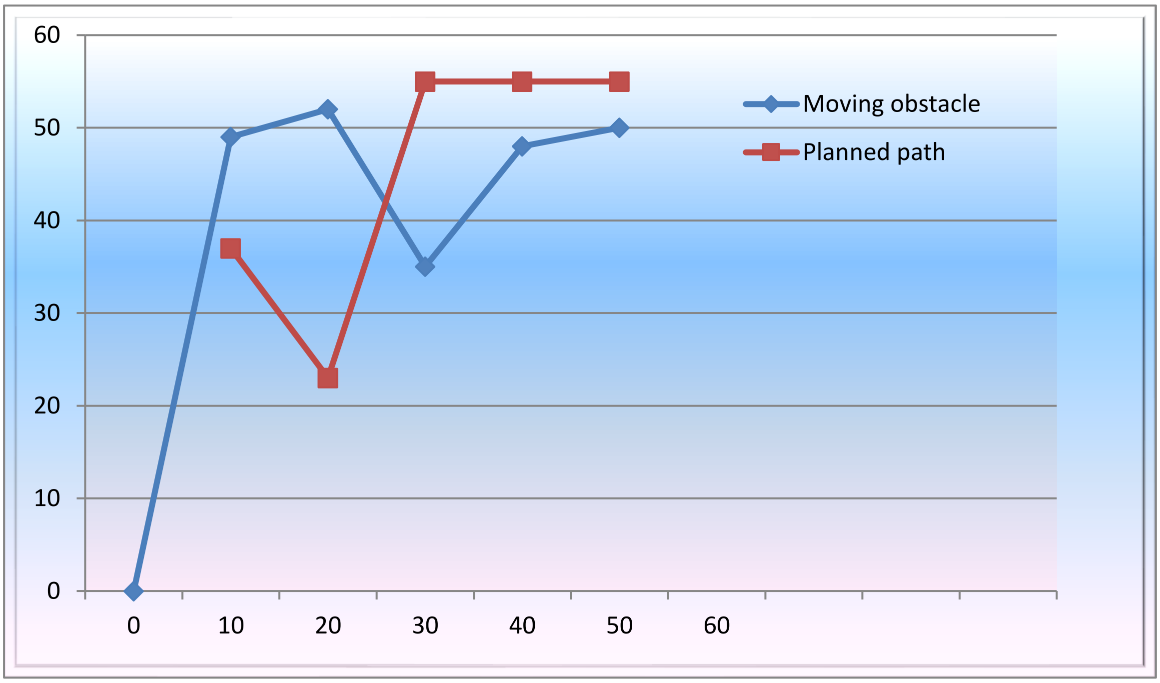

Figure 4: Freezing up a dynamic obstacle.

“Warming up” is a term used to describe the process of preparing for Obstacles that change throughout time: In this scenario, the robot's starting and target position are the same as in the previous one. The workspace barriers appear in the places illustrated in Figure 5. The robot begins to move, and after five route steps, the obstacles begin to drift in the directions indicated by the arrows. Here the robot dynamically and successfully approaches the target in a difficult circumstance.

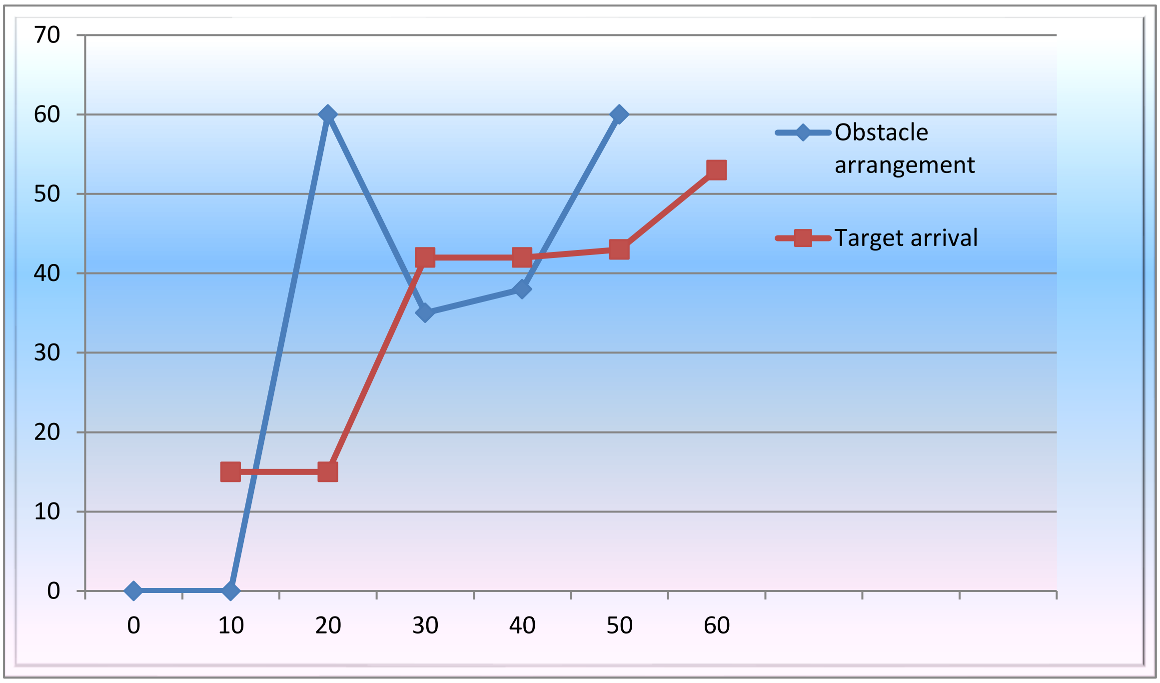

Figure 5: Warming up a dynamic obstacle.

With this model, we did a series of tests with randomly occurring obstacles in sizes 20, 150, and 250. In a L_1 metric, these models examples clearly show the path's potential to be optimum. With these amounts of barriers, the path length steadily rises. The aim in this example is to attain a goal in a setting with fixed obstacles. The robot is said to identify an impediment only when it is directly in front of it. Obstacle positions are considered free throughout the paths design procedure. The described method, which starts from scratch every time, has a new obstacle that could be identified by the robot. This suggested solution dynamically incorporates found impediments into the path creation process. Further, allow the robot to travel in real-time rather than having to wait for the entire course to be preplanned. The Robot detects the resulting paths and the hurdles in the traversal.

5. CONCLUSION

We've proposed a new model of neural network that can calculate a dynamic distance transform (dynamic grid potentials), which might be beneficial for route planning in an altering environment. With a wave expansion process and a set of principles for detecting the next suitable route step, robots are combined effectively in the suggested neural network dynamics. The brain activity landscape evolves and compensates for environmental modification due to local connections within neurons and regular stimulation also at the target neuron. This ensures that the potentials of the grid are properly formed. The target point is consistently at the potential field's lowest value, attracting the robot to a goal. The production of a grid potential is a quick operation where the network is extremely parallel and locally linked (every neuron in this networked field updates its states every time).

These proposed neural network dynamics has been evaluated in the circumstances of autonomous navigation and investigation are conducted on a variety of complicated dynamic environmental changes, such as obstacle emergence, vanish, and drift, rejection of random hurdles, and utilizing the workspace. It has demonstrated both the ability to adjust quickly to dynamic changes and, in the exclusion of the latter, the ability to quickly stabilize the activity. In such aL_1metric, the planned pathways are safe and have a propensity to be optimum. Due to the rapid dynamical update of the potential field, the robot explores actively rather than waiting for "great opportunities" in the environment. As a result, the suggested method may be thought of as a balance between a proclivity for path ideally and an active and mobile response to environmental alterations. The method provided here might be used to design paths for mobile autonomous systems as well as robotic manipulators.

REFERENCE

1. Praveen Kalla et al. "Coordinate Reference Frame Technique for Robotic Planar Path Planning". In: Materials Today: Proceedings 5.9, Part 3 (2018). Materials Processing and characterization,16th – 18th March 2018, pp. 19073–19079. issn: 2214- 7853. doi: 10.1016/j.matpr.2018.06. 260. url: http://www.sciencedirect.com/science/article/pii/ S221478531831383X.
2. Chaymaa Lamini, Said Benhlima, and Ali Elbekri. "Genetic Algorithm Based Approach for Autonomous Mobile Robot Path Planning". In: Procedia Computer Science 127 (2018). PRO-CEEDINGS OF THE FIRST INTERNATIONAL CONFERENCE ON INTELLIGENT COMPUTING IN DATA SCIENCES, ICDS2017, pp. 180–189. issn: 1877-0509. doi:10.1016/j.procs.2018.01.113. url: http://www.sciencedirect.com/science/article/pii/ S187705091830125X.
3. Thi Thoa Mac et al. "Heuristic approaches in robot path planning: A survey". In: Robotics and Autonomous Systems 86 (2016), pp. 13–28. issn: 0921-8890. doi: 10.1016/j.robot.2016.08. 001. url: http://www.sciencedirect.com/science/article/pii/ S0921889015300671.
4. Azzeddine Bakdi et al. "Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control". In: Robotics and Autonomous Systems 89 (2017), pp. 95–109. issn: 0921-8890. doi: 10.1016/j.robot.2016.12.008. url: http://www.sciencedirect.com/science/article/pii/ S0921889016302512.
5. Farhad Bayat, Sepideh Najafinia, and Morteza Aliyari. "Mobile robots path planning: Electrostatic potential field approach". In: Expert Systems with Applications 100 (2018), pp. 68–78. issn: 0957-4174. doi: 10.1016/j.eswa.2018.01.050. url: http://www.sciencedirect.com/science/article/pii/ S0957417418300630.
6. Julien Moreau et al. "Reactive path planning in intersection for autonomous vehicle". In: IFAC-PapersOnLine 52.5 (2019). 9th IFAC Symposium on Advances in Automotive Control AAC 2019, pp. 109–114. issn: 2405-8963. doi: 10.1016/j.ifacol.2019.09.018. url: http: //www.sciencedirect.com/science/article/pii/ S2405896319306378.
7. Ghosh, S., Panigrahi, P. K., and Parhi, D. R. (2017). Analysis of FPA and BA meta-heuristic controllers for optimal path planning of mobile robot in cluttered environment. IET Sci. Measure. Technol. 11, 817–828. doi: 10.1049/iet-smt.2016.0273.
8. Orozco-Rosas, U., Montiel, O., and Sepúlveda, R. (2019). Mobile robot path planning using membrane evolutionary artificial potential field. Appl. Soft Comp. 77, 236–251. doi: 10.1016/j.asoc.2019.01.036.
9. Han, J., and Seo, Y. (2017). Mobile robot path planning with surrounding point set and path improvement. Appl. Soft Comp. 57, 35–47. doi: 10.1016/j.asoc.2017.03.035.
10. Latombe, J. C. (1991). Robot motion planning. Boston, MA: Kluwer Academic Publishers.
11. Fiorini, P., & Shiller, Z. (1998). Motion planning in dynamic environments using velocity obstacles. International Journal of Robotics Research, 17, 760–772.

12. Suresh, P., Rajesh, K.B. and Sivasubramonia Pillai, T.V. and Jaroszewicz, Z., "Effect of annular obstruction and numerical aperture in the focal region of high NA objective lens," Elsevier – Optics Communication, Vol – 318, 2014, pp. 137–141.
13. Hsu, D., Kindel, R., Latombe, J. C., & Rocks, S. (2002). Randomized kinodynamic motion planning with moving obstacles. International Journal of Robotics Research, 21, 233–255.
14. Koenig, S., & Likhachev, M. (2002). Incremental A*. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), Advances in neural information processing systems 14. Cambridge, MA: MIT Press.
15. Suresh, P., Thilagavathi. R., Gokulakrishnan, K., Rajesh, K.B. and Pillai, T.V.S., "Focusing properties of a 4Pi configuration system under the illumination of double ring shaped LG11 beam," Springer – Optical and Quantum Electronics, 1-6, 2014
16. Li, G., and Chou, W. (2018). Path planning for mobile robot using self-adaptive learning particle swarm optimization. Sci. China Inform. Sci. 61, 052204–052213. doi: 10.1007/s11432-016-9115-2.
17. Suresh, P., Mariyal, C., Rajesh, K.B. and Pillai, T.V.S., "Polarization effect of cylindrical vector beam in high numerical aperture lens Axicon systems," Elsevier – Optik, Vol. 124, 2013. pp. 1632-1636
18. Bakdi, A., Hentout, A., Boutami, H., Maoudj, A., Hachour, O., and Bouzouia, B. (2017). Optimal path planning and execution for mobile robots using genetic algorithm and adaptive fuzzy-logic control. Robot. Autonomous Syst. 89, 95–109. doi: 10.1016/j.robot.2016.12.008.
19. Ohnishi, S., Uchibe, E., Nakanishi, K., and Ishii, S. (2019). Constrained Deep Q-learning gradually approaching ordinary Q-learning. Front. Neurorobot. 13, 103–112. doi: 10.3389/fnbot.2019.00103.
20. Shen, C.-,w., Ho, J.-,t., Ly, P. T. M., and Kuo, T.-,c. (2019a). Behavioural intentions of using virtual reality in learning: perspectives of acceptance of information technology and learning style. Virtual Reality 23, 313–324. doi: 10.1007/s10055–018-0348–1.
21. Sung, I., Choi, B., and Nielsen, P. (2020). On the training of a neural network for online path planning with offline path planning algorithms. Int. J. Inform. Manage. 102142–102150. doi: 10.1016/j.ijinfomgt.2020.102142.
22. Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: a brief survey. IEEE Signal Process. Magazine 34, 26–38. doi: 10.1109/MSP.2017.2743240.
23. Wulfmeier, M., Rao, D., Wang, D. Z., Ondruska, P., and Posner, I. (2017). Large-scale cost function learning for path planning using deep inverse reinforcement learning. Int. J. Robot. Res. 36, 1073–1087. doi: 10.1177/0278364917722396.
24. Suresh, P., Mariyal, C., Gokulakrishnan, K., Rajesh, K.B. and Pillai, T.V.S., "Tight Focusing of Circularly Polarized Beam over high NA Lens Axicon with a Diffractive Optical Element," Elsevier – Optik, vol. 124. Issue 20, 2013. pp. 4389-4392
25. M. S. Gharajeh and H. B. Jond, "Hybrid global positioning system-adaptive neuro-fuzzy inference system based autonomous mobile robot navigation," Robotics and Autonomous Systems, vol. 134, Article ID 103669, 2020.
26. X. Liu, D. Zhang, J. Zhang et al., "A path planning method based on the particle swarm optimization trained fuzzy neural network algorithm," Cluster Computing, vol. 2021, pp. 1–15, 2021.
27. Z. Jiang, Y. Xia, and G. Shen, "A novel learning-based global path planning algorithm for planetary rovers," Neuro-computing, vol. 361, pp. 69–76, 2019.
28. I. Sung, B. Choi, and P. Nielsen, "On the training of a neural network for online path planning with offline path planning algorithms," International Journal of Information Management, vol. 57, Article ID 102142, 2021.