Heuristic approaches in robot path planning: A survey

Thi Thoa Mac [a,b,*], Cosmin Copot [a], Duc Trung Tran [b], Robin De Keyser [a]

[a] *Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 914, 9052 Zwijnaarde, Belgium*
[b] *School of Mechanical Engineering, Hanoi University of Science and Technology, Dai Co Viet street 1, Hanoi, Viet Nam*

HIGHLIGHTS

- Autonomous navigation of a robot is a promising research domain due to its extensive applications.
- This survey concentrates on heuristic-based algorithms in robot path planning which are comprised of neural network, fuzzy logic, nature inspired algorithms and hybrid algorithms.
- The strengths and drawbacks of each algorithm are discussed and future outline is provided.

ARTICLE INFO

ABSTRACT

Autonomous navigation of a robot is a promising research domain due to its extensive applications. The navigation consists of four essential requirements known as perception, localization, cognition and path planning, and motion control in which path planning is the most important and interesting part. The proposed path planning techniques are classified into two main categories: classical methods and heuristic methods. The classical methods consist of cell decomposition, potential field method, subgoal network and road map. The approaches are simple; however, they commonly consume expensive computation and may possibly fail when the robot confronts with uncertainty. This survey concentrates on heuristic-based algorithms in robot path planning which are comprised of neural network, fuzzy logic, nature-inspired algorithms and hybrid algorithms. In addition, potential field method is also considered due to the good results. The strengths and drawbacks of each algorithm are discussed and future outline is provided.

© 2016 Elsevier B.V. All rights reserved.

1. Introduction

Autonomous navigation is one of the most important requirements of an intelligent vehicle. Robot navigation is a designed process toward a target position while avoiding obstacles. There are four basic components of this process as shown in Fig. 1 [1]: (i) *perception*, the robot uses its sensors to extract meaningful information; (ii) *localization*, the robot determines its location in the working space; (iii) *cognition and path planning*, the robot decides how to steer to achieve its goal; (iv) *motion control*, the robot regulates its motion to accomplish the desired trajectory. Path planning of a robot can be considered as a sequence of translation and rotation from starting position to the destination while avoiding ob-

stacles in its working environment. There are two suggested techniques covering all approaches in robot path planning: (i) *global path planning* or off-line path planning and (ii) *local path planning* or on-line path planning [2,3]. A global path planner usually generates a low-resolution high-level path based on a known environmental map or its current and past perceptive information of the environment. The method is valuable of producing an optimized path; however, it is inadequate reacting to unknown or dynamic obstacles. On the other hand, local path planning algorithm does not need *a priori* information of the environment. It usually gives a high-resolution low-level path only over a fragment of global path based on information from on-board sensors. It works effectively in dynamic environments. The method is inefficient when the target is long distance away or the environment is cluttered. Normally, the combination of both methods is advised to enhance their advantages and eliminates some of their weaknesses [4–6]. The robot path planning problem can be divided into classical methods and heuristic methods [7,8] as shown in Fig. 2.

The most important classical methods consist of cell decomposition method (CD), potential field method (PFM), subgoal method

---

* Corresponding author at: Thi Thoa Mac, Department of Electrical Energy, Systems and Automation, Ghent University, Technologiepark 914, 9052 Zwijnaarde, Belgium.

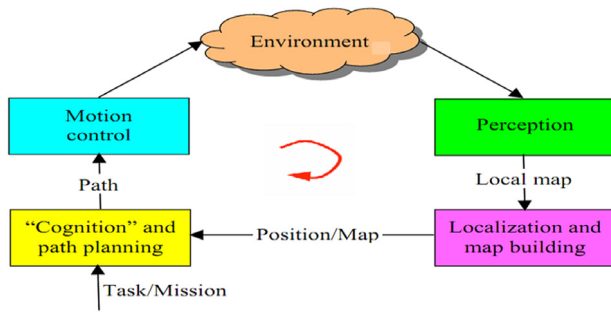*E-mail address:* thoa.macthi@ugent.be (T.T. Mac).

Fig. 1. Robot navigation structure [1].

(SG) and sampling-based methods. In Cell Decomposition method, the free space of the robot's configuration is divided into small regions called cells. The goal is to provide a collision-free path to reach the target. The applications of robot path planning based on this approach can be found in [9,10]. In potential field method, the obstacles and the goal are assigned repulsive and attractive forces, respectively, so that the robot is able to move toward the target while pushing away from obstacles [11]. A new formula of repelling potential is performed in the interest of reducing oscillations and avoiding conflicts when obstacles locate near a target [12]. Instead of single robot, the method is also extended successfully to navigate multi robots to perform complex tasks [13,14]. D.H. Kim proposed a new framework to escape from a local minimum location of robot path based on PFM [15,16]. To solve path planning problem in dynamics environments, modifications on classical PFM are introduced in [17,18]. Subgoal method uses a list of reachable configurations from the starting position to the goal position while avoiding all obstacles. The subgoal method applications for robot navigation are presented in [19–21].

Planning schemes based on sampling-based motion planning (SBP) algorithms have received considerable attention due to their capability in complex and/or time critical real world planning problems. Arguably, the most persuasive SBPs to date include probabilistic road-map (PRM) and rapidly-exploring random trees (RRT) [22]. Although the idea of connecting points sampled randomly is fundamental in both approaches, these two methods are different in the manner that they construct a graph connecting the points [23]. A comprehensive survey of work in SBP is given in [24]. The PRM algorithm has been recorded to implement well in high-dimensional state spaces. The PRM is created by curves or straight lines that enable the robot to go anywhere in its free space. The two well-known road-map methods, namely, visibility graph (VG) and Voronoi diagram (VD) have achieved very good results with dramatically different types of roads. A visibility graph is a graph that comes as close as possible to obstacles. As a result, the shortest path is found by applying this method; however, the path touches obstacles at the vertices or edges and thus is dangerous for the robot. Contrary, Voronoi diagram creates a road that tends to maximize the distance between the robot and the obstacles. Therefore, the solution paths based on Voronoi diagram are not optimal with respect to path length.

The advantage of this method is that only a limited number of sensors is used in the robot navigation task. Path planning of a robot swarm using road-map technique is proposed in [25–27]. Several improvements are proposed by [28,29]. RRT has received a considerable amount of attention, because of its computational efficiency and effectiveness and its ability to find a feasible motion plan relatively quickly, even in high-dimensional space [30,31]. In [32,33], the navigation approach consists of four separate modules: localization, path planning, path execution and obstacle avoidance; obstacle avoidance is proposed for autonomous urban service mobile robots. To avoid obstacles, the authors combine a local planner with a slightly modified dynamic window method. The local planner is implemented using RRT. RRT explores a robot working space by incrementally building a tree, creating new branches by generating points randomly and linking them to the closest point for which an obstacle-free path is obtained. A problem in RRT is that it produces a path with many branches in the workspace by using the randomized technique. To overcome this problem, a novel path planning approach for a mobile robot in dynamic and cluttered environments with kinodynamic constraints is presented in [34]. The algorithm called Heuristic Arrival Time Field biased Random Tree (HeAT-RT) that takes advantage of the high-exploration ability of a randomized tree is combined with an arrival time field and heuristics to achieve the path optimality, safety, and applicability to the real robot. Instead of choosing a random point from the entire workspace like the basic RRT algorithm, they select a random point using the bias from the arrival time field so that the tree grows in a favorable direction toward the target. The kinodynamic RRT*, an incremental sampling-based approach for asymptotically optimal motion planning for robots with linear dynamics is introduced in [35].

The ability of SBP to provide valid paths for constrained high dimensional problems is advantageous. Despite the hit-or-miss sampling approach being the core of the SBP's effective strategy, it leads to the inclusion of many redundant maneuvers in the obtained path. In [36], a modification of the termination condition is proposed in a way such that the SBP keeps running to iteratively converge the path cost. The solution convergence remains an unanswered problem; until it is proven that given infinity runtime RRT will not achieve an optimal path [37]. Recently, a family of optimal SBP, RRT*, PRM* and RRG* are introduced to guarantee asymptotic optimality. Despite their effectiveness, they provide no theoretical guarantees for reaching an optimal solution [24].

Many efforts have been made to apply classical approaches onto real-time motion planning [38–42]. Incremental algorithms to update distance maps, Voronoi diagrams, and configuration-space collision maps are presented in [38]. The representations are initialized by using a given grid map or point cloud. For efficient on-line applications, only update cells that are affected by changes in the environment are updated. Therefore, these algorithms can be used in real-world scenarios with unexpected or moving obstacles. Another practical approach to solve the limitations of the road-map based mobile robot path planner in a home environment is introduced in [39]. The proposed
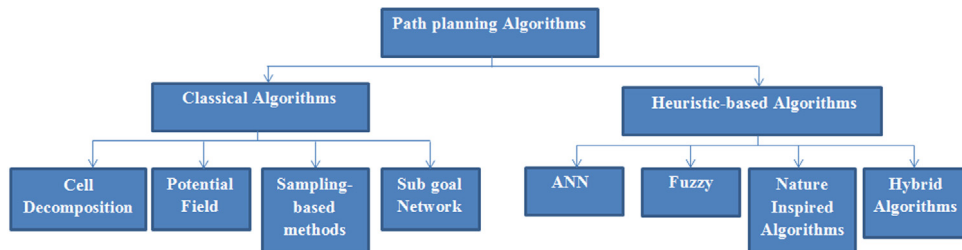

Fig. 2. The classification of robot path planning algorithms.

Table 1
Nature inspired methods in chronological order [65].

| Year	Nature inspired methods	Year	Nature inspired methods
2012	Keill Herd	2002	Estimation of distribution algorithm
2010	Bath algorithm	1995	Particle swarm optimization
2010	Artificial bee algorithm	1992	Ant colony optimization
2009	Cuckoo search algorithm	1989	Tabu search
2009	Gravitational search algorithm	1983	Simulated annealing
2007	Firefly algorithm	1979	Cultural algorithms
2007	Intelligent water drops	1975	Genetic algorithms
2005	Harmony search algorithm	1966	Evolutionary programming
2005	Honey bee algorithm	1965	Evolution strategies
2002	Bacterial foraging algorithm		
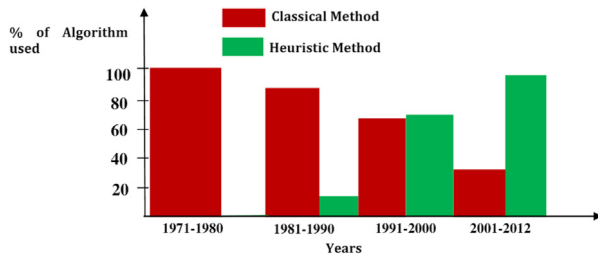
Fig. 3. Application of classical and heuristic algorithms [8].

approach incrementally constructs a hierarchical road-map which has a multi-layered structure using low cost sonar sensors. Motion planners based on RRT algorithms for multi-agent system, omni-directional robots and an autonomous vehicle in an urban environment are, respectively, proposed in [40–42]. However, classical approaches do not produce optimal paths and tend to be locked in some local minima. Moreover, some of them may not provide the suitable solution in the presence of multiple obstacles or in dynamic environments. In order to avoid the inefficiency of classic methods, heuristic approaches are employed.

Heuristic methods analyzed in this paper consist of neural network (NN), fuzzy logic (FL) and nature inspired methods (NIM) and hybrid algorithms. Neural networks have been applied successfully in many robot path planning applications due to their advantages such as nonlinear mapping, learning ability, and parallel processing [43–51]. In fuzzy logic algorithm, robot navigation is based on a set of IF–THEN rules. In [52–60], different approaches that are based on fuzzy logic are introduced to solve robot path planning problems. Methods of integrating fuzzy inference systems and neural networks to achieve ability of human thinking (fuzzy logic) and the ability of learning (neural networks) are studied in [61–64]. Other approaches are inspired by biological behaviors becoming more and more popular in robot path planning applications. A chronological order of the most popular nature inspired methods is provided in Table 1 [65]. Some widely known algorithms inspired by biology behaviors have been successfully applied in robot path planning, for instance, Genetic Algorithm (GA) [66–72], Particle Swarm Optimization (PSO) [73–78], and Ant Colony Optimization (ACO) [79–85].

Classical methods dominated the field of robot path planning before 2000 but have much lower percentage of usage after that year. It is also found that heuristic approaches are more and more popular in robot navigation field compared to classical techniques as shown in Fig. 3. Therefore, this study concentrates on heuristic-based algorithms for robot path planning (Section 2). In addition, one of the most attractive classical methods named potential field method is also considered due to the good results (Section 3). Their advantages and drawbacks are discussed, then the applications are analyzed. The discussion, conclusion of the survey and future outline of robot path planning are provided in Section 4.

2. Heuristic-based algorithms

In this section, the study covers neural network (NN), fuzzy logic (FL), some of the most common nature-inspired algorithms which are GA, PSO, ACO in the robot path planning applications. The positive and negative aspects of the algorithms are available in each subsection. In the beginning of subsections GA, PSO, ACO, a brief introduction of the basic algorithm is provided. This description is considered to be necessary in order to comprehend the use of the specific method for robot path planning performances. The section is organized into four parts. Neural network (NN) is presented in 2.1 and fuzzy logic (FL) is shown in 2.2. After that, the hybrid algorithm integrating neural network and fuzzy logic (NN–FL) technique is introduced in 2.3. Nature-inspired algorithms lastly are analyzed in 2.4. Other hybrid algorithms are integrated in each subsection.

2.1. Neural network (NN)

Recently, neural network has been developed in many robot path planning applications [86,43,87,44–51]. NN is employed to model complex relationships between inputs and outputs in the robot navigation task.

The implementation of NN in robot navigation is categorized into three categories: (i) *interpreting the sensory data*, (ii) *obstacle avoidance*, and (iii) *path planning* [86]. S.X. Yang and M. Meng [43] have proposed robot path planning using a neural network in non-static environment. Two testing scenarios are surveyed: a well-known beam robot competition micro mouse maze and an environment of both moving target and moving obstacles. The study shows that the robot travels on a continuous smooth route to catch the moving target while avoiding moving obstacles. The combination of the Lyapunov stability theory and qualitative analysis guarantees the system stability. However, the size of mobile robot, moving obstacles and target are not considered in the simulations. Moreover, the robot and moving obstacles are assumed in the same constant velocity so that this methodology might confront some difficulties when executed in real systems. A hierarchical architecture which constructs internal models of a robot environment for goal-oriented navigation by an imitation learning process is proposed in [87]. This method is based on the Reservoir Computing paradigm for training Recurrent Neural Networks (RNN). It consists of two randomly generated RNNs, one for modeling the localization ability and one for learning the navigation skill. The experimental results on a simulated robot indicate that the trained system can localize the robot in large unknown environments and navigate successfully to the desired goals.

In a different sight of view, D. Janglova [44] constructed collision-free path based on two neural networks: principal component analysis (PCA) and a multilayer perceptron (MLP), in Fig. 4(a) and (b), respectively. PCA is an unsupervised linear procedure which finds uncorrelated features from the inputs. It is a data reduction method which converts the input data to a few
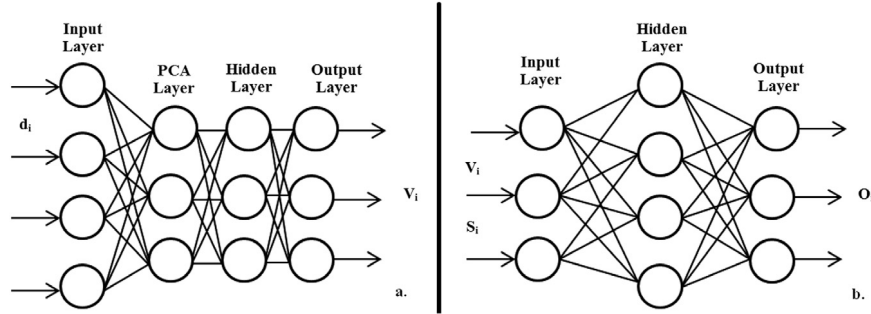
Fig. 4. Topology of PCA and MLP network [44].

principal components. The number of principal components is a compromise between training efficiency and accuracy results. The PCA network learns by generalized Hebbian rule. In the beginning, PCA network is performed to decide the safe space (outputs-V_i) using data from ultrasound range finder (inputs-d_i). Hence, the robot has sufficient information about its distances to all objects in the workspace. The knowledge about free segments, V_i are obtained as the outputs of the first neural network. Those values together with the goal segments S_i are used as inputs of the second neural network. The outputs of the second neural network are suitable directions O_i of the robot which will be sent to the controller. The network includes 18 input neurons (9 value V_i and 9 value S_i), 20 hidden neurons and 9 output neurons O_i. The advantages of this approach are easy to implement and approximate any inputs/outputs map. However, the disadvantages are a low speed and a huge amount of samples of training process.

S.H. Dezfoulian et al. [45] inherited the study done by [44] for determining a free space in a robot working environment. The algorithm is performed on three robots using different range sensors. Although there are slightly differences in the journeys, all three mobile robots can successfully navigate from the starting location to the goal location. Hence, it promises to extend the algorithm to real system using low cost sensors. However, the main drawback of the algorithm is the large amount of gathering training samples (3000 training patterns) for the robots in different scenarios.

Furthermore, M.K. Singh et al. [46–48] designed a neural network algorithm that enables robot to move safely in an unknown environment with static/dynamic obstacles. The inputs of the proposed neural network consist of left, right, and front obstacle distance (LOD, FOD, ROD perspectively); the angle between the robot and the target (TA) and the output is robot steering angle as depicted in Fig. 5. The information is received from an array of sensors. A four-layer neural network is applied to solve the optimization problem of the path planning. Compared with [45], the neural network is trained by presenting only 200 patterns typical scenarios. Therefore, that algorithm dramatically reduces gathering training samples as well as computational time.

M.A. Sagban et al. nominated a new neural network approach based on reactive navigation algorithm in unstructured indoor environments [49]. The method integrated off-line learning and on-line learning stages to reach optimum performance. In [50], a NN back-propagation algorithm is applied to train the robot on-line so that the robot is capable to avoid moving obstacles. Expanding the technique in [49] for formation navigation of multiple mobile robots, an adaptive NN control is proposed in [51]. The algorithm is based on Lyapunov function, graph theory and PSO algorithm. First, the method generates successive path for the leader according to current obstacles perceived. Then, follower robots use that information to perform successfully their navigations. This method conducts quite good results both in simulations and in real experiments. Table 2 condenses the above reviewed researches.
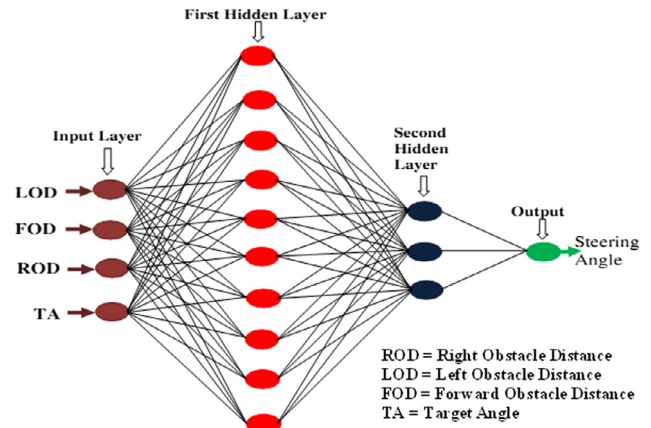

Fig. 5. Neural Network structure for robot path planning [46].

In a summary, NN has been widely used to implement motion planner of autonomous robot. However, NN has some drawbacks. For instance, a neural network is usually time consuming. Further, the learning algorithm may not be able to guarantee the convergence to an optimal solution. Therefore, the integrated methods are more appropriate for addressing the robot navigation problem in real-world applications.

2.2. Fuzzy logic technique

Beside neural networks, fuzzy logic is considered for expressing the subjective uncertainties in human mind. Human has a noticing ability to perform navigation tasks without any exact measurements or computations. It is highly desirable to mimic this ability to develop autonomous robot navigation strategies [52]. In fuzzy logic algorithm, the decision making of robot navigation can be phrased by a set of IF–THEN rules. To easily implement, the navigation problem breaks into simpler tasks and behaviors.

H. Chang and T. Jin introduced a fuzzy inference model to solve mobile robot path planning problem [53]. The locations of the goal/obstacles and current speed of the robot in unknown dynamic environments are perceived from sensors. In that model, three major navigation goals: target orientation (Seek Goal), obstacle avoidance (Avoid Obstacle) and rotation movement (Maintain Heading) are included in a cost function to find optimal steering angle θ. The mobile robot navigates intelligently by varying the weights of the cost functions depending on the environment (Fig. 6).

In [54], V.M. Peri et al. have solved multi-robots navigation problem by using a fuzzy logic controller and a Petri net. The fuzzy rules navigate the robot in the working space according to obstacles distribution or targets position. Each controller of the mobile robot enable it not only to avoid obstacles in a cluttered environment but also to avoid other mobile robots. Hence, the

Table 2
Application of NN for robot path planning.

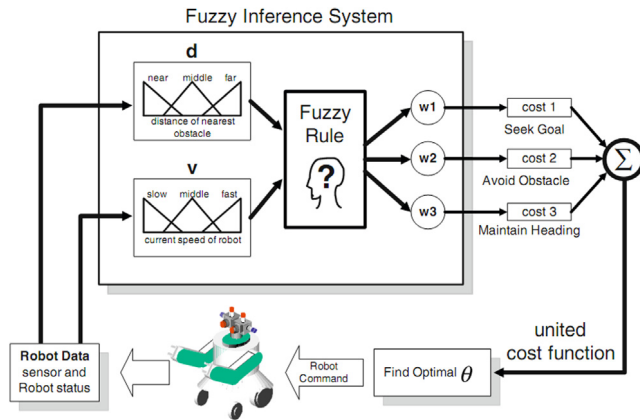| Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
Neural	S.X. Yang [43]	N	Rectangle	Y	Y	N	Y	N	Y	N	2000
	E. A. Antonelo [87]	N	Arbitrary	Y	N	Y	N	N	Y	N	2010
	D. Janglova[44]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2004
	S.H. Dezfoulian [45]	N	Arbitrary	Y	N	Y	N	N	Y	N	2013
	M.K. Singh [46] [47]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2009
	M.K. Singh [48]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2011
	M. Sagban [49]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2012
	I. Engedy [50]	Y	With markers	Y	Y	Y	N	Y	Y	Y	2009
NN/PSO	X. Chen [51]	Y	Circle	Y	Y	Y	Y	Y	Y	Y	2006

Fig. 6. Structure of fuzzy inference system for robot path planning [53].

robot can perform its task and incorporate with other robots by using collision prevention rules implemented. In [55], J.H. Lilly suggested different approach utilizing both negative fuzzy rules and traditional positive rules. In which, negative fuzzy rules determine movements to be avoided rather than performed. A positive rules base is stated to drive the robot to the goal in the absence of obstacles while a negative rules base is activated in the presence of obstacles. As a result, fewer rules than using solely positive rules are applied on the obstacle avoidance controller.

A fuzzy logic system with 48 fuzzy rules is designed in [88], which comprises of three behaviors: target seeking, obstacle avoidance and barrier following. A combination of multiple sensors is equipped to sense the obstacles near the robot, the target location and the current robot speed. Thus, the mobile robot is able to preferably "see" its environment and avoid static and dynamic obstacles autonomously. It can generate reasonable paths toward the target in various scenarios without suffering from the "symmetric indecision" and "dead cycle" problem. A novel terrain traversability analysis method for mobile robot navigation is presented in [89]. The robot builds a real-time map and analyzes the terrain traversability of its working environment. Using fuzzy approach, the robot can handle uncertainties in the laser range finder data mounted on the mobile robot. The robot extracts the roughness and slope from an elevation map created from point cloud measurement data. Those values are inputted into the fuzzy system and the traversability is calculated based on the fuzzy rule and the defuzzifier. This traversability value is transformed into a risk value which is expressed in the form of a vector field histogram (VFH) to avoid obstacles. The comparison of this method and the conventional method showed the reliability of the proposed method in spite of the limited sensor data acquisition area.

The robot navigation method using fuzzy logic technique and stereo vision based path planning in a complex unknown environment is introduced in [56–58]. With a purpose of increasing the autonomy in cluttered environments, an ordered hierarchical architecture based on fuzzy reasoning is suggested [56]. The robustness, accuracy, adaptability and efficiency of the proposed system are tested in various scenarios in which the obstacles are distributed randomly. The fuzzy logic rules are expected to robustly lead the robot toward the goal position without collision of dynamic obstacles.

The navigation of an autonomous mobile robot in an unknown environment with obstacles is considered by G. Mester [59]. In their approach, the distance and the orientation between robot and obstacle, target are taken into account. A robot system integrating technique, dead reckoning, self-localization and environment recognition, has been proposed by T. Lee and C. Wu [60]. In their approach, membership functions and fuzzy rules are designed based on genetic algorithm. Nevertheless, the method only solves the direction problem without considering velocity. In addition, genetic algorithm may not be the best choice for generation of rule base for robot navigation so that the robot might fail to find suitable path in some scenarios. The uses of fuzzy logic in robot path planning of the reviewed studies are summarized in Table 3.

As mentioned earlier, fuzzy logic has the power of simulating the human thinking represented by linguistic variables and knowledge base represented by if–then rules. However, it has difficulty in selecting the most suitable rules and membership functions. On the other site, neural network is able to learn by updating synaptic weights of the network. However, this representation of knowledge as synaptic weights cannot be acquired by human reasoning. These matters together suggest a method of integrating fuzzy inference system and neural network to achieve both advantages of similarity to human thinking (fuzzy logic) and the ability to learn (neural networks) which will be introduced in 2.3. Furthermore, fuzzy logic can be applied to solve the problem of the parameter adaptation with some natural-inspired algorithms for optimal robot path planning which will be discussed in some parts of Section 2.4.

2.3. Neuro-Fuzzy technique

The robot path planning performances in [46] are improved in [90,91] where neuro-fuzzy technique is used instead of sole neural technique. With the same inputs as treated in [46], the technique performs a neural network as a preprocessor of a fuzzy logic controller. The compared performances between the two techniques demonstrate the prominence of neuro-fuzzy system. The algorithm is illustrated in Fig. 7.

The integrated methods also have been proposed in [61–64]. A method named Prune-able fuzzy adaptive resonance theory neural architecture (PAFARTNA) is adopted in [61]. The mobile robot is able to automatically navigate in dynamic environments with a degree of optimality due to its capacity of global path planning using a world model that is dynamically learned and updated on-line. In [62], it is suggested an adaptive five layer neuro-fuzzy network which consists of a fuzzy controller with 48 fuzzy rules and a learning adaptation model. Two learning algorithms are

Table 3
Application of FL for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
Fuzzy	H. Chang [53]	Y	Corridor	Y	N	Y	N	N	Y	Y	2013
	V.M. Peri [54]	N	Arbitrary	Y	Y	Y	N	N	Y	Y	2005
	J.H. Lilly [55]	N	Circle	Y	N	Y	N	N	Y	N	2007
	A. Zhu [88]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2004
	Y. Tanaka [89]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2015
	A. Foudil [56]	N	Arbitrary	Y	Y	Y	N	Y	Y	Y	2014
	L. Li [57]	N	Arbitrary	Y	N	Y	N	N	N	Y	2009
	C.H. Chao [58]	N	Arbitrary	Y	N	Y	N	N	N	Y	2009
	G. Mester [59]	N	Arbitrary	Y	N	Y	N	Y	Y	N	2008
Fuzzy–GA	T. Lee [60]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2003

Fig. 7. Neuro-Fuzzy structure for robot path planning [90].
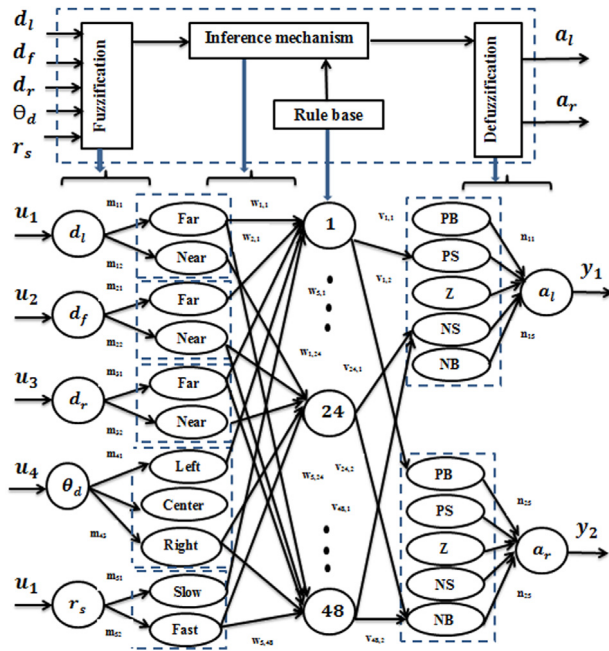
Fig. 8. Neuron–Fuzzy structure for robot path planning [62].

designed to adjust the parameters of the membership functions and automatically overcome a problem of redundant fuzzy rules. As a sequence, a reasonable trajectory is generated for a mobile robot in a dynamic environment with both moving target and moving obstacles. As can be seen in Fig. 8, the inputs are the distances between the robot and obstacles in the left, the front and the right, d_l, d_f, d_r; the target direction θ_d; the current speed of the robot r_s. Next, the input membership variables are denoted
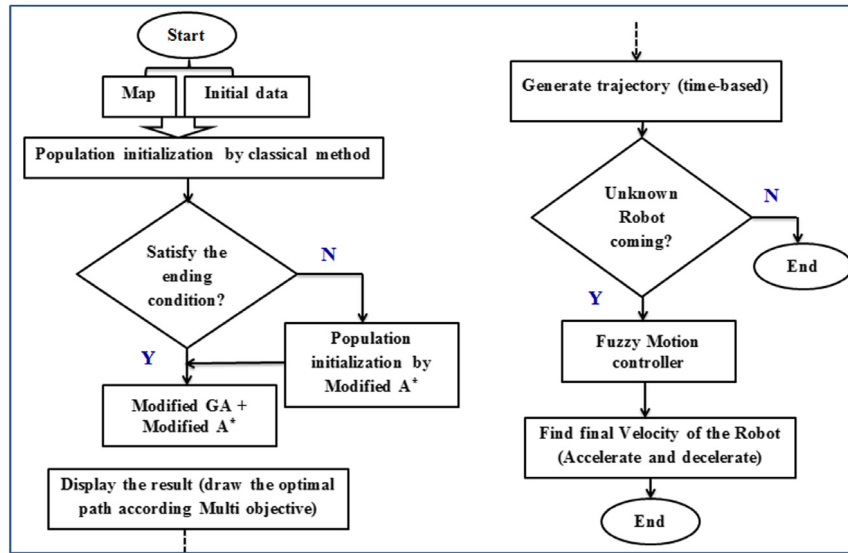
in the second layer. Then, the rule base is referred in the third layer. After that, the output membership variables are designed in the fourth layer. In the end, the accelerations of left and right wheels, a_l and a_r, are chosen as outputs. The inputs d_l, d_f, d_r are expressed by only two linguistic variables: Near and Far. Similarly, θ_d is expressed by three linguistic variables: Left, Center and Right; the current speed r_s is expressed by two linguistic variables: Fast and Slow. The good results prove the efficiency of the algorithm. However, the proposed approach requests much effort for tuning the parameters.

The performances of different integrated approaches (NN-tuned FL, GA–NN–FL, GA-tuned adaptive network-based fuzzy inference system (Anfis)) are compared among themselves and with those of three other approaches (default behavior, manually constructed FL, potential field method) through computer stimulations of a car like robot in a dynamic environment [64]. The approach based on default behavior is quite simple, instead of using motion planner, the robot maintains a default rule with maximum acceleration, zero deviation at each time step and the collision between the robot and the most critical obstacle is avoided based on the predicted position technique. In the manually constructed FL approach, Mamdani type fuzzy logic controller is applied while in NN-tuned FL, GA–NN–FL approaches, a back-propagation algorithm and GA are used to optimize FL, respectively. In GA-tuned Anfis approach, GA is also applied to find optimal solution for Anfis. The performance of the approach based on default behavior is the worst. Moreover, the integrated approaches show better performance than the other approaches in most of the studied scenarios. Potential field method is the fastest of all methods however less adaptive than the integrated approaches. It also indicates that the performance of the neuro-fuzzy approaches depends on the training data, it will face some difficulties when the testing scenarios are much different from the training scenarios.

Table 4
Application of Neuron–Fuzzy technique for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
Neuro-fuzzy	D.R. Parhi [90]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2008
	S.K. Pradhan [91]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2006
PAFARTNA	R. Araújo [61]	Y	Arbitrary	Y	Y	Y	N	N	Y	Y	2006
Neuro-fuzzy	A. Zhu [62]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2009
	M.M. Joshi [63]	N	Arbitrary	Y	N	Y	N	Y	Y	N	2011
Neuro-fuzzy, GA-Anfis, PFM	N.B. Hui [64]	Y	Circle	N	Y	Y	N	Y	Y	N	2006

Fig. 9. Robot path planning by GA–Fuzzy-A* approach [68].

Normally, the integrated approaches based on neural and fuzzy provide much better results than individual techniques. The comparisons in [90,91,63,66] show this advantage; however, the integrated methods usually have a time consuming problem. The sum up of application on Neuro-Fuzzy technique of robot path planning in the subsection is presented in Table 4.

2.4. Natural inspired algorithms

Recently, robot navigation techniques inspired by biology behaviors, known as bio-mimetic algorithms, have obtained much more attention. The following sub-sections introduce the applications of robot path planning using three members of them: GA, PSO and ACO. The choice of these three natural inspired methods is made because there are many studies which demonstrated the good performances for robot path planning in the literature.

2.4.1. Genetic Algorithms (GA)

GA is an optimization tool based on the natural genetics that takes advantages from procedures such as natural selection, crossover, and mutation. GA has great potentiality to solve the combination optimization problems. The pseudo code of GA path planning is provided in Algorithm 1 [66].

Various studies have been executed based on GA in robot path planning domain. In [67], that issue is solved by applying a knowledge based genetic algorithm (problem-specific genetic algorithm) instead of the standard GA. The algorithm is designed with both domain knowledge and small-scale local search. The proposed method is suitable to both static and dynamic environments. This algorithm is extended in [92] for multiple mobile robots in dynamic environments. When the robot working space is simple or does not change very fast, this approach provides a near optimal solution from the current location to the target. In a very complicated fast-dynamic environment, a feasible path is always produced. To generate a feasible collision free path from the starting point to the goal point when the mobile robot is trapped in an acute "U" or "V" shaped obstacle or the mobile robot encounters dynamic obstacles, a GA based dynamic path planning algorithm (DPPA) is proposed in [93]. In addition, this algorithm is goal oriented and thus reduces unnecessary search time. The performances in both simulation and real-time implementation are investigated and evaluated by checking the ability to generate a feasible path to avoid acute obstacles/dynamic obstacles and to find the shortest path. In [68], B.K Oleiwi et al. presented a hybrid approach based on GA, a modified search A* algorithm and fuzzy logic. A* algorithm tends to minimize route cost on the graph, as in road map approach or on a regular grid. It evaluates the goodness of each node or cell by combining two metrics which are distances between robot and the start/goal positions. As a result, multi objective optimization of free-collision path is generated. The approach obtains a smooth trajectory with an associated minimum energy cost by using cubic spline. First, the modified GA and a modified search A* algorithm are applied to find the optimal path. After that, global optimal trajectory is treated as input for fuzzy motion controller to regenerate a time based trajectory. When unknown obstacles appear in the robot path, the fuzzy controller will decrease the speed of the robot. The performances are very impressive in the environment with dynamic obstacles. A description of proposed approach is depicted in Fig. 9.

Table 5
Application of GA technique for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
GA	Y. Hu [67]	N	U shape, Rectangle	Y	Y	Y	N	N	Y	N	2004
	S.X. Yang [92]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2006
GA based DPPA	S.C. Yun [93]	N	Parallelepiped	Y	Y	Y	N	N	Y	Y	2011
GA–Fuzzy, A*	B.K. Oleiwi [68]	Y	Circle	Y	Y	Y	N	N	Y	N	2014
GA–Fuzzy	B. Karim [69]	N	Point	N	N	N	Y	Y	Y	N	2013
GA–PSO	H.C. Huang [70]	N	U shape, Rectangle	Y	N	Y	N	N	Y	N	2011
GA–Fuzzy	S.M.R. Farshchi [71]	Y	Cylinder	Y	Y	Y	N	N	Y	N	2011
Adaptive GA	A.K. Karami [72]	N	Trap shape	Y	N	Y	N	N	Y	N	2015

In order to track the moving target, a method based on fuzzy logic and genetic algorithm is presented [69]. An optimal path of autonomous robot is delivered by applying hybrid meta-heuristic GA–PSO algorithm [70]. The combination of GA and PSO proposes new solution by executing crossover and mutation operators. As a result, the premature convergence in conventional GA and PSO algorithm is avoided.

The novel genetic–fuzzy algorithm has been applied to generate a robot path in an unknown environment [71]. In which, the genetic algorithm is served to find the optimal path in an alien dynamic environment. A full consideration of three elements: the collision avoidance path, the shortest distance and smoothness of the path is taken into account in the fitness function. Moreover, a dual-population concept on diversifying the population is also introduced to improve the performances. In order to overcome the local trap problem and to avoid premature convergence, A.H. Karami et al. designed an adaptive selection operator instead of conventional one in GA based on feedback information of the search process [72]. In the beginning, an initial population is generated by a random-based method. Then, the created paths are improved in each generation of the genetic algorithm. Therefore, the robot path planning quality is significantly improved. The applications of GA algorithm of robot path planning of above studies are shown in Table 5.

In summary, the main imperfection of the GA approach in robot path planning field is that it is not feasible in dynamic environments. Since GA operates in a grid map and does not control the population diversity, a premature convergence might happen. Therefore, GA is integrated with other algorithms such as fuzzy, PSO, etc. to achieve better results in robot path planning applications.

2.4.2. Particle Swarm Optimization (PSO)

Similar with GA, PSO performs a random population and uses objective function to evaluate particles. However, crossover and mutation operation as in genetic algorithm disappear in PSO. Particles update themselves with the internal velocity based on particles' self-experiences with their social experiences. As PSO takes real numbers as particles, it is very convenient compared to GA, which needs to be changed to binary encoding.

Initially, PSO is inspired by social behavior of bird flocking or fish schooling. This algorithm is a population based stochastic optimization technique. Basically, PSO is initialized with a set of random solutions and then updated each generation based on optimal schema. Then, the global optimum is achieved by changing the collection of particles in a search space toward a promising area. In each iteration, the particles will update their locations according to Eqs. (1) and (2) until reaching the target location (global optimum) [74]. The moving is affected by a fitness function that evaluates the quality of each solution. Fig. 10 in the left hand side depicts the position update in PSO particle j in iteration i.

$$prtpos_j^{i+1} = prtpos_j^i + prtvel_j^{i+1} \tag{1}$$

Algorithm 1 : GA path planner pseudo-code

Generate randomly the initial population (set of feasible paths) using the greedy approach based on Euclidean distance heuristic
  while (generation number<max generation number) **do**
    Fitness function
    Elitist selection
    Rank selection
  repeat
    choose randomly two paths (parents) from current generation
    if (random number generated<crossover rate) **then**
      if (the parents have common cells) **then**
        perform the crossover
        move the resulting paths to the next generation
      else
        choose other two parents
      end if
    else
      move the parents to the next generation
    end if
  until (next generation size< max population size)
  for each path in the next generation **do**
    if (random number generated <mutation rate) **then**
      choose randomly a cell and replace it
      if (the resulting path feasible) **then**
        replace the old path with the new one
      else
        choose randomly two cells C_1 and C_2 from the path
        remove all the cells between C_1 and C_2
        connect C_1 and C_2 by using the greedy approach based on Euclidean distance heuristic
      end if
    end if
  end for
  end while

$$prtvel_j^{i+1} = prtvel_j^i + c_1 r_1 (pbest_j^i - prtpos_j^i) \\ + c_2 r_2 (gbest^i - prtpos_j^i) \tag{2}$$

$$prtvel_j^{i+1} = prtvel_j^{i-1} + c_1 r_1 (prtpos_j^i - pworst_j^i) \\ + c_2 r_2 (prtpos_j^i - gworst^i) \tag{3}$$

where:

$prtpos_j^{i+1}$ is the position of the particle j in iteration $i + 1$;
$prtvel_j^{i+1}$ is the velocity of the particle j in iteration $i + 1$;
$pbest_j^i$ is the best position of the particle j in iteration i;
$gbest^i$ is the best position in the swarm till iteration i;
$pworst_j^i$ is the worst position of the particle j in iteration i;
$gworst^i$ is the worst position in the swarm till iteration i;
c_1, c_2 are factors balancing the effect of self-knowledge and social knowledge;
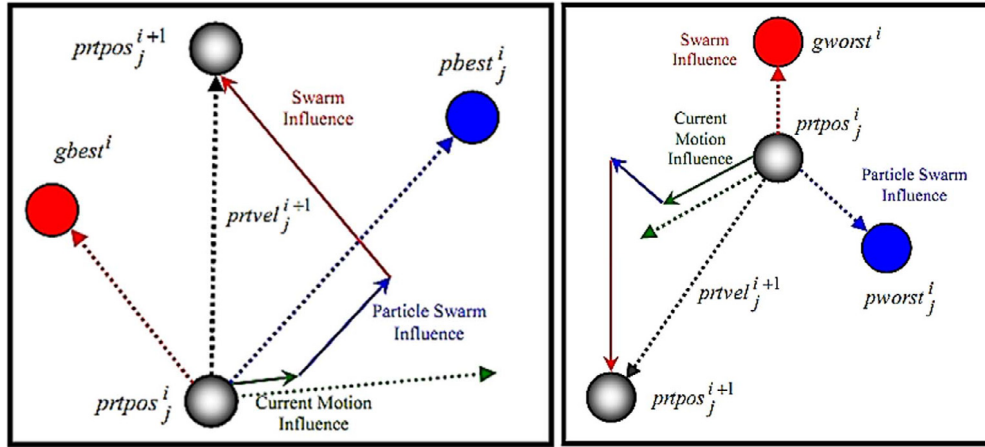r_1, r_2 are random numbers between 0 and 1.

Fig. 10. Depiction of a particles position update in PSO (left) and NPSO (right) [74].

Table 6
Application of PSO technique for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
PSO/NPSO	E. Masehian [74]	N	Arbitrary	Y	N	Y	N	N	Y	Y	2010
PSO	Y. Zhang [75]	N	Arbitrary	Y	N	Y	N	N	Y	N	2013
	Y. Hao [94]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2007
	Y. Wang [95]	Y	Circle	Y	Y	Y	N	N	Y	N	2009
PSO–Fuzzy Lyapunov	K.D. Sharma [76]	N	Rectangle	Y	N	Y	N	N	Y	N	2006
Stochastic PSO	C. Xin [77]	N	Various	Y	N	Y	N	N	Y	N	2006
Fuzzy, Darwinian PSO	M.S. Couceiro [78]	N	Arbitrary	Y	Y	Y	N	Y	Y	Y	2012

First, the Negative PSO (NPSO) is presented in [73] and then, it is employed for robot motion planning in [74]. In the opposite direction of PSO, each particle in NPSO updates its position according to its own previous worst solution and its group's previous worst solution to find the optimal value as in Fig. 10-right hand side. The procedure is applied to avoid the worst positions based on similar formula of the regular PSO. In this research, the performance of PSO and NPSO is compared in robot path planning.

In [75], Y. Zhang et al. presented a multi-objective robot path planning algorithm based on PSO in uncertain environment. The objective function consists of the risk degree and the distance of path. Hence, the path planning problem is considered as a constrained bi-objective optimization problem with uncertain coefficients. A polar coordination PSO (PPSO) for the robot path planning in dynamic environments is presented in [94]. The algorithm decomposes the task into a global phase and local planning phase. It can search for the global optimal path depended on static obstacles information. Then, an on-line real-time path planning strategy is adopted to avoid moving obstacles by predicting their future positions. In [95], a path planning for a group of robots is formulated as an optimal problem with constraints on obstacle avoidance and V-shape formation in the dynamic environment. The fitness function is defined by minimizing the trajectory of the group while keeping the V-shape formation.

Many variations of PSO have been proposed by extensive researches since its inception. K.D. Sharma et al. proposed a novel idea for vision-based robot navigation using stable adaptive fuzzy tracking controllers and PSO-based hybrid methodologies [76]. GA and PSO approaches have carried out for robot navigation in unknown environments [77]. For the same purpose, a Darwinian particle swarm optimization algorithm is applied to escape from local optimal solutions [78]. This one is expanded from the origin PSO by adding a natural selection operation. The experimental

---

Algorithm 2 : PSO pseudo-code

while maximum iterations or minimum error criteria is not attained **do**
   for each particle **do**
     Initialize particle
   end for
   for each particle **do**
     Calculate the fitness value
      if the fitness value is better than the best fitness value in history (pbest)
        Set current value as the new pbest
      end if
   end for
for each particle **do**
   Find in the particle neighborhood the particle with the best fitness (gbest)
   Calculate particle velocity $prtvel_{ij}$ according to the velocity equation (2)
   Apply the velocity constriction
   Update the particle position $prtpos_{ij}$ according to the position equation (1)
   Apply the position constriction
end for
end while

---

result demonstrates an effective convergence as compared to the traditional algorithms in real system executions and simulated trials. Also, the distribution of target locations, does not greatly affect the adaptive algorithm performance. The summary of PSO algorithm for robot path planning in this part is presented in Table 6 and the pseudo code of PSO is provided in Algorithm 2.

However, most of the proposed PSO approaches were not reliable in real-time robot applications, special in dynamic
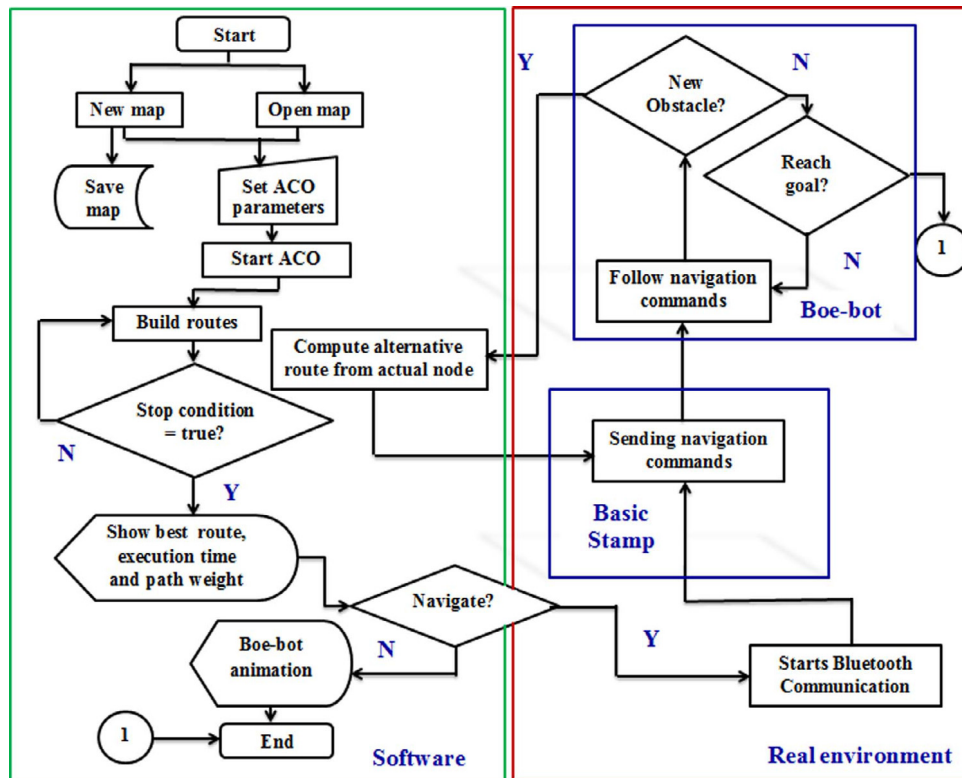
Fig. 11. Robot navigation based on Fuzzy–ACO algorithm [83].

environments. It might be that these researches mainly solve problems by updating the velocity parameters aiming to attain more effective and faster convergence.

2.4.3. Ant Colony Optimization (ACO)

Like PSO algorithm, ACO is a data clustering algorithm by implementing swarm behavior. ACO is originated from natural ant colony behavior. At the essence of the behavior, the interacted communication between the ants enables them to find the shortest path between their nest and the food sources. The optimal route is obtained through evaluation of the amount of pheromones deposited by ants on the paths. This characteristic of real ant colonies is inherited in ACO algorithm to solve discrete optimization problems. Consequently, the ACO is more suitable for problems where source and destination are clearly predefined and specific. The main procedure of the ant colony algorithm is as follows:

1. Generate ant (or ants).
2. Loop for each ant (until complete scheduling of tasks)
3. Deposit pheromone on visited states.
4. Daemon activities.
5. Evaporate pheromone.

An algorithm for solving multi-goal planning problems in the presence of obstacles is presented in [79]. In this study, the ant colony optimization is combined with a sampling-based point to point path planning algorithm. To evaluate the performance, a quantitative comparison with two existing sampling-based algorithms has been made. The ACO algorithm, offering a compromise between solution quality and speed, is the superior choice given the physical parameters for the robot planning. In [80], a method for solving the SLAM-Problem by applying an optimization technique based on the Ant Colony Optimization meta-heuristic is introduced. A tree-like data structure is generated, in which the path from the root to a leaf represents a metrical map estimation. An (almost) optimal path is found by employing a ACO.

In [81], X. Chen et al. proposed a two-stage ACO model which has ability to overcome the main inconsistency problem between premature convergence and the optimal path. In [82]-[83], an optimal path planning method is introduced based on Ant Colony Optimization Meta-Heuristic (ACO-MH). The robot is considered as a point so that it occupies an exact cell in the discrete representation of working environment. The length of the path and the difficulty for the navigation are taken into account in the cost function that is evaluated by a fuzzy logic system. The algorithm has the adaptive capability of changing in the environment so that it can perform a global robot path planning with dynamic obstacles. The mobile robot is navigated on-line as the flow chart in Fig. 11.

In [84], the searching range and the speed of algorithm are significantly improved by choosing the initial pheromone distribution. The combination of potential field method (PFM) and ACO is proposed for the problem of robot planning in [85]. In the hybrid algorithm, deterministic planning is established by applying PFM, then ACO is used for searching optimal route. The results indicate that the probability of the optimal solution is dramatically increased. The sum up of this subsection is presented in Table 7.

In conclusion, traditional ACO involves some shortcomings, for example, requiring a long time to reach to optimal value if the size of the problem is large. The solutions are created blindly in the starting of evolution phase since the pheromone concentrations of all members are equally initialized. As a consequence, it takes a long time to find a better path from a great number of smoother paths.

3. Classical method—potential field method

As mentioned in Section 1, classical methods no longer provide suitable and effective results for mobile robot navigation in an unknown and dynamic environment compared to heuristic techniques. Hence in this section, only Potential Field Method (PFM)

Table 7
Application of ACO technique for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
ACO	B. Englot [79]	N	Rectangle	Y	N	Y	N	N	Y	N	2011
	R. Iser [80]	N	Various	Y	N	Y	N	N	Y	N	2010
	X. Chen [81]	N	Various	Y	N	Y	N	N	Y	N	2013
ACO–Fuzzy	M.A.P. Garcia [82,83]	N	Rectangle	Y	N	Y	N	N	Y	Y	2007 2009
Improved ACO	J. Bai [84]	N	Circle	Y	N	Y	N	N	Y	N	2011
PFM–ACO	D. Zhao [85]	N	Rectangle	Y	N	Y	N	Y	Y	N	2006

is presented due to its advantages of combination with different heuristics methods. In addition, the modified potential methods also provide quite good performances in non-static environment.

Potential field method is broadly applied for real time collision-free path planning in mobile robot domain. In this method, the robot is considered as a particle steering under the influence of an potential field, which is determined by obstacles and the goal. The goal is supposed to have attractive potential force while the obstacles generate the repulsive potential forces. Therefore, the potential field will pull robot toward the goal and push it far away from obstacles (Fig. 12).

The PFM is attractive due to its mathematical simplicity and efficiency. However, the basic PFM algorithm is simply based on the distance between the robot and obstacles/target in a static environment. A navigation system for a family of indoor monitor mobile robots is presented in [97]. The robot detects its working space using a 2D laser range finder. The hybrid localization method integrates an odeometry localization method, a straight line matching method, and a corner matching method. Then, a weight average method is implemented to obtain the optimal pose according to the error model. PFM is used to plan a collision free path that satisfies certain optimization criteria. Recently, several proposed potential field methods applied in a dynamic environment with both of moving target and moving obstacles are studied. In order to avoid moving obstacles, the relative positions and relative velocities of robot and the obstacles are taken into account in the repulsive potential function [96,98–101]. Similarly, the attractive potential function is modified so that this value increases as the relative distance or relative velocity between robot and target becomes larger [96] as in Eq. (4):

$$U(p, v) = U_{att}(p, v) + U_{rep}(p, v)$$

$$U_{att}(p, v) = \alpha_p \|p_{tar}(t) - p(t)\|^m + n\alpha_v \|v_{tar}(t) - v(t)\|^n$$

$$U_{rep}(p, v) = \begin{cases} 0, & \text{if } \rho_s(p, p_{obs}) - \rho_m(v_{RO}) \geq \rho_0 \text{ or } v_{RO} \leq 0 \\ \eta \left(\dfrac{1}{\rho_s(p, p_{obs}) - \rho_m(v_{RO})} - \dfrac{1}{\rho_0} \right), & \\ & \text{if } \rho_s(p, p_{obs}) - \rho_m(v_{RO}) < \rho_0 \text{ or } v_{RO} > 0 \\ \text{not defined}, & \\ & \text{if } \rho_s(p, p_{obs}) - \rho_m(v_{RO}) < 0 \text{ or } v_{RO} > 0 \end{cases} \quad (4)$$

where

$p_{tar}(t)$ and $p(t)$ are the position of the target and the position of the robot at time t;

$v_{tar}(t)$ and $v(t)$ are the velocity of the target and the velocity of the robot at time t;

ρ_0 is the positive constant describing the influence range of the obstacle;

$\rho_s(p, p_{obs})$ is shortest distance between the robot and the body of the obstacle;

η, α_p, α_v, m and n are positive tuning constants;

$\rho_m(v_{RO})$ is the distance traveled by the robot before its velocity reduces to zero.

In [96], the vectors for defining the repulsive force and the attractive force are depicted in Fig. 12. The proposed attraction force F_{att} and repulsion force F_{rep} are negative gradient of respective attraction $U_{(att)}(q)$ and repulsion $U_{rep}(q)$ potential field functions. The formulas are

$$F(p, v) = F_{att}(p, v) + F_{rep}(p, v)$$

$$F_{att}(p, v) = m\alpha_p \|p_{tar}(t) - p(t)\|^{(m-1)} n_{RT}$$
$$+ n\alpha_v \|v_{tar}(t) - v(t)\|^{(m-1)} n_{RT}$$

$$F_{rep(p,v)} = \begin{cases} 0, & \text{if } \rho_s(p, p_{obs}) - \rho_m(v_{RO}) \geq \rho_0 \text{ or } v_{RO} \leq 0 \\ F_{rep1} + F_{rep2}, & \text{if } 0 < (\rho_s(p, p_{obs}) \\ & - \rho_m(v_{RO})) < \rho_0 \text{ or } v_{RO} > 0 \\ \text{not defined}, & \\ & \text{if } \rho_s(p, p_{obs}) - \rho_m(v_{RO}) < 0 \text{ or } v_{RO} > 0. \end{cases} \quad (5)$$

The functions F_{rep1} and F_{rep2} are defined in Eq. (6).

$$F_{rep1} = \frac{-\eta}{(\rho_s(p, p_{obs}) - \rho_m(v_{RO}))^2} \left(1 + \frac{v_{RO}}{a_{max}} \right) n_{RO}$$

$$F_{rep2} = \frac{\eta v_{RO} v_{RO\perp}}{\rho_s(p, p_{obs}) a_{max}(\rho_s(p, p_{obs}) - \rho_m(v_{RO}))^2} n_{RO\perp} \quad (6)$$

$$\rho_m(v_{RO}) = \frac{(v_{RO}(t))^2}{2a_{max}}$$

where

n_{RT} is the unit vector pointing from the robot to the target;
a_{max} is the maximum deceleration of robot;
v_{RO} is the relative velocity of the robot with respect to the obstacle in the direction from the robot to the obstacle;
$v_{RO\perp}$ is the velocity component perpendicular to v_{RO};
n_{RO} and $n_{RO\perp}$ are the unit vectors in their directions.

The vectors for defining the new repulsive potential are presented in Fig. 13.

In [98,99], modified PFM are extended to track a moving target. Furthermore, in [100], the implementation elements, such as maximum linear speed and angular speed of the robot, are also contemplated. Those approaches make assurance that the robot follows the moving target while avoiding moving obstacles. An integrated method named path-guided Artificial Potential Field (APF) with Stochastic Reachable Sets (Path-Guided APF-SR) including sampling-based technique and APF is proposed in [102]. The method has low computation cost and works flexible with any path generating collisions in very crowded environments with various moving obstacles. The sampling-based technique is used to identify a collision-free path with respect to static obstacles. Then, APF is applied to safely navigate through moving obstacles, using the path as an attractive intermediate goal bias. In addition, the authors incorporate a repulsive potential field for each moving obstacle based on pre-computed stochastic reachable (SR) sets. The proposed method is tested in environments with 300 stochastic moving obstacles that either are free of static obstacles or have static obstacles in the shape of a "bug trap" or narrow corridors.

Table 8
Application of PFM for robot path planning.

| Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
PFM	F. Tan [97]	N	Various	Y	N	Y	N	N	N	Y	2010
	S.S. Ge [96]	N	Point	N	Y	N	Y	Y	Y	N	2002
New PFM	L. Huang [98]	N	Circle	Y	N	N	Y	Y	Y	N	2009
	L. Huang [99]	N	Circle	Y	N	N	Y	Y	Y	N	2012
	L. Valbuena [100]	Y	Cube	Y	N	Y	N	N	Y	Y	2012
Path-Guided APF-SR	H.T. Chiang [102]	N	Square	Y	Y	Y	N	N	Y	N	2015
GA–Fuzzy, GA–NN, PFM	N. B. Hui [101]	Y	Circle	N	Y	N	Y	Y	Y	Y	2009
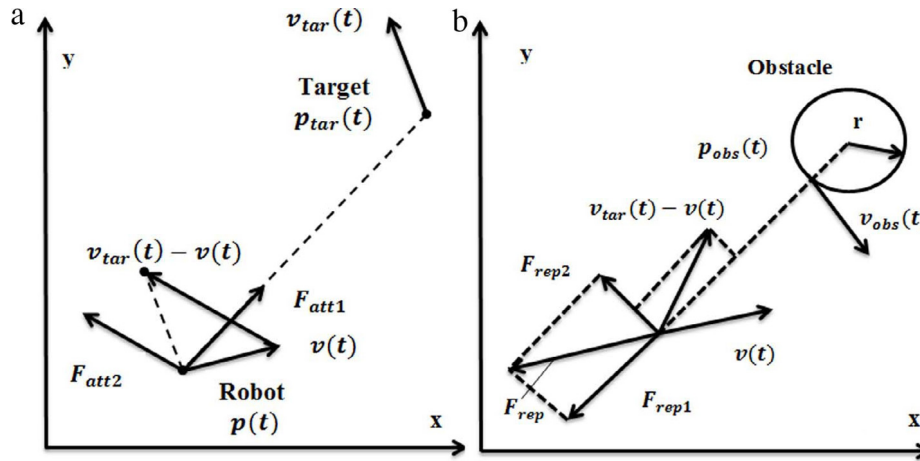
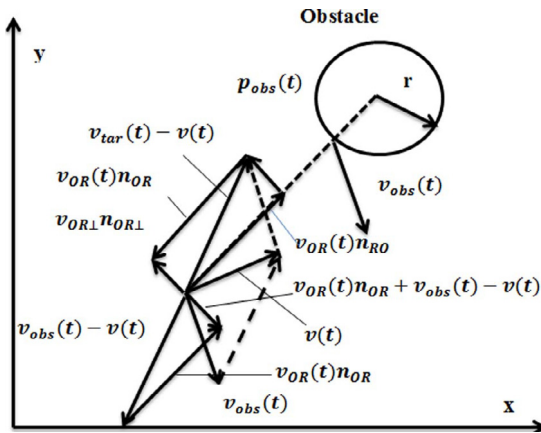**Fig. 12.** The attractive force (a) and repulsive force (b) in 2D space [96].

Fig. 13. The Vectors for defining the repulsive potential [96].

The obtained results show that the method avoids local minima and has a very high success rate (over 90%) in highly dynamic environments.

Comparative performances of various robot path planning schemes have employed in [101]. Hybrid algorithms including genetic–fuzzy, genetic–neural systems and a conventional PFM have been examined. An attempt is made to identify the best method in terms of traveling time, robustness, adaptability, reliability. The time consumption of those hybrid algorithms are observed to be quite small. Hence, they are qualified for on-line implementations. However, PFM is found to be less adaptive than genetic–fuzzy, genetic–neural approach. The uses of PFM for robot path planning presented in this subsection are summarized in Table 8(a).

To sum up, although PFM has a simple structure and easy to implement, it might fail in some scenarios where obstacles are located close to each other. It is necessary to utilize modified potential field method as suggested [96,98–101] or integrated PFM with other intelligent techniques.

4. Discussion and conclusion

The robot path planning problem is divided into classical methods and heuristics methods. The classical methods are easy to implement. Therefore, they are preferred in many real-time motion planning applications and may obtain good results shown by RRT, PFM, PRM. However, classical approaches often require precise information about the robot's working environment thus more accurate sensors should be used in real-time applications. Recently, a family of optimal SBP such as RRT*, RRM*, and RRG* are adopted to guarantee asymptotic optimality. Although they work very effectively, they provide no theoretical guarantees for reaching an optimal solution [24]. Compared to classical methods, heuristic methods are considered more "intelligent" and more "advanced" as they can adapt to both uncertain and incomplete information in constantly changing environments (i.e. neural network, fuzzy logic methods) and achieve optimal solutions (nature-inspired methods). Thus, they are applied on autonomous navigation with complex tasks. However, their serious drawbacks are the necessary learning phase and the high computational time. In many applications, the computational capacity of the robot controller unit is limited; thus, the approaches with lower computational cost such as RRT and PFM are better choices. With ever-increasing computer power, this concern is only concentrated on small and fast robots.

In the last 15 years, significant progress toward the development of robots path planning has been made. This survey has reviewed the current state of the art of heuristic approaches and potential field method in robot path planning. Heuristic methods analyzed in this paper consist of neural network, fuzzy logic and nature-inspired methods (genetic algorithm, particle swarm optimization and ant colony optimization). The applications of the reviewed studies are summarized in Table 9 and in Fig. 14.

Table 9
Application of NN–FL–GA–PSO–ACO–PFM for robot path planning.

Approach	Authors	Kinematic model	Obstacle shape	Static obstacle	Dynamic obstacle	Static target(s)	Dynamics target(s)	Adapted velocities	Simulation	Real system	Year
	S.X. Yang [43]	N	Rectangle	Y	Y	N	Y	N	Y	N	2000
	E. A. Antonelo [87]	N	Arbitrary	Y	N	Y	N	N	Y	N	2010
	D. Janglova[44]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2004
Neural	S.H. Dezfoulian [45]	N	Arbitrary	Y	N	Y	N	N	Y	N	2013
	M.K. Singh [46] [47]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2009
	M.K. Singh [48]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2011
	M. Sagban [49]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2012
	I. Engedy [50]	Y	With markers	Y	Y	Y	N	Y	Y	Y	2009
NN/PSO	X. Chen [51]	Y	Circle	Y	Y	Y	Y	Y	Y	Y	2006
	H. Chang [53]	Y	Corridor	Y	N	Y	N	N	Y	Y	2013
	V.M. Peri [54]	N	Arbitrary	Y	Y	Y	N	N	Y	Y	2005
	J.H. Lilly [55]	N	Circle	Y	N	Y	N	N	Y	N	2007
	A. Zhu [88]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2004
Fuzzy	Y. Tanaka [89]	Y	Arbitrary	Y	N	Y	N	N	Y	Y	2015
	A. Foudil [56]	N	Arbitrary	Y	Y	Y	N	Y	Y	Y	2014
	L. Li [57]	N	Arbitrary	Y	N	Y	N	N	N	Y	2009
	C.H. Chao[58]	N	Arbitrary	Y	N	Y	N	N	N	Y	2009
	G. Mester [59]	N	Arbitrary	Y	N	Y	N	Y	Y	N	2008
Fuzzy–GA	T. Lee [60]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2003
Neuro-fuzzy	D.R. Parhi [90]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2008
	S.K. Pradhan [91]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2006
PAFARTNA	R. Araújo [61]	Y	Arbitrary	Y	Y	Y	N	N	Y	Y	2006
Neuro-fuzzy	A. Zhu [62]	N	Arbitrary	Y	N	Y	N	Y	Y	Y	2009
	M.M. Joshi [63]	N	Arbitrary	Y	N	Y	N	Y	Y	N	2011
Neuro-fuzzy, GA-Anfis, PFM	N.B. Hui [64]	Y	Circle	N	Y	Y	N	Y	Y	N	2006
GA	Y. Hu [67]	N	U shape, Rectangle	Y	N	Y	N	N	Y	N	2004
	S.X. Yang [92]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2006
GA based DPPA	S.C. Yun [93]	N	Parallelepiped	Y	Y	Y	N	N	Y	Y	2011
GA–fuzzy, A*	B.K Oleiwi [68]	Y	Circle	Y	Y	Y	N	N	Y	N	2014
GA–fuzzy	B. Karim [69]	N	Point	N	N	N	Y	Y	Y	N	2013
GA–PSO	H.C. Huang [70]	N	U shape, Rectangle	Y	N	Y	N	N	Y	N	2011
GA–fuzzy	S.M.R. Farshchi [71]	Y	Cylinder	Y	Y	Y	N	N	Y	N	2011
GA	A.K. Karami [72]	N	Trap shape	Y	N	Y	N	N	Y	N	2015
PSO/NPSO	E. Masehian [74]	N	Arbitrary	Y	N	Y	N	N	Y	Y	2010
PSO	Y. Zhang [75]	N	Arbitrary	Y	N	Y	N	N	Y	N	2013
	Y. Hao, [94]	N	Arbitrary	Y	Y	Y	N	N	Y	N	2007
	Y. Wang [95]	Y	Circle	Y	Y	Y	N	N	Y	N	2009
PSO–Fuzzy, Lyapunov	K.D. Sharma [76]	N	Rectangle	Y	N	Y	N	N	Y	N	2006
Stochastic PSO	C. Xin [77]	N	Various	Y	N	Y	N	N	Y	N	2006
Fuzzy, Darwinian, PSO	M.S. Couceiro [78]	N	Arbitrary	Y	Y	Y	N	Y	Y	Y	2012
ACO	B. Englot [79]	N	Rectangle	Y	N	Y	N	N	Y	N	2011
	R. Iser [80]	N	Various	Y	N	Y	N	N	Y	N	2010
	X. Chen [81]	N	Various	Y	N	Y	N	N	Y	N	2013
ACO–Fuzzy	M.A.P. Garcia [82] [83]	N	Rectangle	Y	N	Y	N	N	Y	Y	2007
											2009
Improved ACO	J. Bai [84]	N	Circle	Y	N	Y	N	N	Y	N	2011
PFM–ACO	D. Zhao [85]	N	Rectangle U shape	Y	N	Y	N	Y	Y	N	2006
PFM	F. Tan [97]	N	Various	Y	N	Y	N	N	N	Y	2010
	S.S. Ge [96]	N	Point	N	Y	N	Y	Y	Y	N	2002
New PF	L. Huang [98]	N	Circle	Y	N	N	Y	Y	Y	N	2009
	L. Huang [99]	N	Circle	Y	N	N	Y	Y	Y	N	2012
	L. Valbuena [100]	Y	Cube	Y	N	Y	N	N	Y	Y	2012
Path-Guided APF-SR	H.T. Chiang [102]	N	Square	Y	Y	Y	N	N	Y	N	2015
Genetic–fuzzy Genetic–NN, PFM	N. B. Hui [101]	Y	Circle	N	Y	N	Y	Y	Y	Y	2009

As mentioned before, each method has some advantages and disadvantages. Neural network has been applied successfully in many robot path planning due to the nonlinear mapping, learning ability, and parallel processing characteristic. However, it is usually time consuming since it often has a large number of parameters that have to be adjusted. It also requires an enough variety of training databases with a possible large learning cost. Another drawback of NN is that it is impossible to interpret the functionality as it is a black-box. It is also difficult to determine the number of layers and the number of neurons. A popular paradigm of learning named supervised learning involves modification of the neural network weights by applying a set of labeled training samples. Each training sample comprises a unique input signal and a corresponding desired response. Then the weights of the neural network are modified to minimize the difference between the desired response and the actual response. In addition, NNs have a build-in ability to adapt their weights to change in the surrounding environment. This means NN can be retrained to deal with minor changes in the operating environmental conditions. Moreover, it is robust to disturbances. Fuzzy logic has power of simulating the human thinking represented by linguistic variables and knowledge-based if–then rules but it has difficulty in selecting the most suitable rules and membership functions. One approach to design a fuzzy logic system is to try to imitate the actions of an experienced operator. It is claimed that this provides good results. A positive aspect is that the fuzzy law is expressed in terms of linguistic variables which is simple and transparent. Fuzzy logic, as a mathematical tool to handle uncertainties, has been applied to systems that are hard to precisely define. The integrated approaches based on neural and fuzzy provide much better results than the individual techniques since they achieve both advantages of similarity to human thinking (fuzzy logic) and the ability to learn (neural networks). As a guide-line for implementing high efficient neuro-fuzzy robot path planning systems they should have characteristics such as fast learning, online adaptability, and self-adjustment to changing environments. The data acquisition and pre-processing of input training data are important factors for neuro-fuzzy systems. Nature-inspired algorithms have great potentialities to solve the optimization problems. Optimality criteria such as minimum distance traveled, minimum time and minimum control effort are considered for path planning problems. However, they are not reliable in real-time applications of robot path in dynamic environments. Therefore, they are often integrated with other methods such as classical algorithms, fuzzy and neural approaches to achieve better results.

Many studies have been implemented in robot path planning but very few of them have reported convincing experimental results. There is a gap between theoretical work and experimental results, although a great amount of theoretical work has been validated in simulations and off-line processing. Most current research activities provide effective solutions to robot path planning in static environments. Current robot path planning algorithms have not yet reached the level of robustness and reliability required for real-world applications. Path planning in alien dynamic environments with moving obstacles and moving target is still a challenging aspect for researchers.

Therefore, it is clear that integrated methods achieve better performances than individual ones despite the time consuming problem. In [51], the combination of neural network and particle swarm optimization is applied successfully for smooth formation navigation of multiple robots to avoid moving obstacles. The fuzzy logic is integrated with neural network [90,91,62–64], genetic algorithm [68,69,71], particle swarm optimization [70,76,78], ant colony optimization[82,83] to achieve better results. The modified potential field method can solve robot navigation problem in dynamic environment [98] but it is not optimal. An impressive
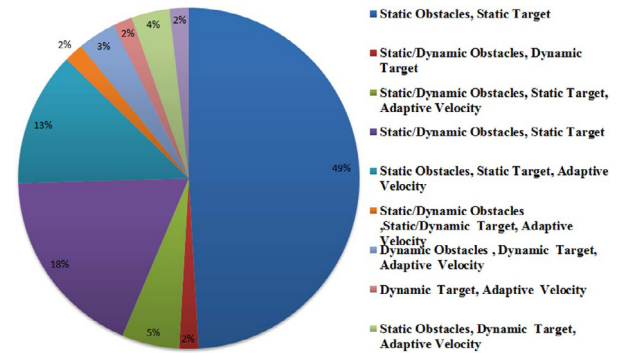

Fig. 14. The path planning applications in the mentioned researches in 2000–2015.

comparisons among genetic–fuzzy, genetic–neural and a conventional potential field methods with moving obstacles in terms of robustness, adaptability, goal reaching capability and repeatability are presented [101]. In the future, the integration of fuzzy logic and neural network, PSO/GA/ACO new potential method RRT, RRT*, RRM*, and RRG* will be promising approaches in robot path planning domain. Those approaches are highly suggested for prospective researches.

References

[1] R. Siegwart, I.R. Nourbakhsh, Introduction to Autonomous Mobile Robot, Massachusetts Institute of Technology Press, Cambridge, USA, 2004.
[2] Y.Q. Qin, D.B. Sun, N. Li, Y.G. Cen, Path planning for mobile robot using the particle swarm optimization with mutation operator, in: International Conference on Machine learning and Cybernetics, Shanghai, China, 2004, pp. 2473–2478.
[3] P. Raja, S. Pugazhenthi, Optimal path planning of mobile robots: A review, Int. J. Phys. Sci. 7 (9) (2012) 1314–1320.
[4] H. Zhang, J. Butzke, M. Likhachev, Combining global and local planning with guarantees on completeness, in: IEEE International Conference on Robotics and Automation, USA, 2012, pp. 4500–4506.
[5] L.C. Wang, L.S. Yong, M.H. Ang Jr., Hybrid of global path planning and local navigation implemented on a mobile robot in indoor environment, in: IEEE International Symposium on Intelligent Control, Canada, 2001, pp. 821–826.
[6] Z. Bi, Y. Yimin, Y. Wei, Hierarchical path planning approach for mobile robot navigation under the dynamic environment, in: IEEE Conference on Industrial Informatics, Daejeon, Korea, 2008, pp. 372–376.
[7] E. Masehian, D. Sedighizadeh, Classic and heuristic approaches in robot motion planning–a chronological review, in: Proceeding of World Academy of Science, Engineering and Technology, Vol. 23, 2007, pp. 101–106.
[8] P.K. Mohanty, D.R. Parhi, Controlling the motion of an autonomous mobile robot using various techniques: a review, J. Adv. Mech. Eng. (2013) 24–39.
[9] J. Rosell, P. Iniguez, Path planning using harmonic functions and probabilistic cell decomposition, in: International Conference on Robotics and Automation, ICRA, Barcelona, Spain, 2005, pp. 1803–1808.
[10] M. Seda, Roadmap method vs. cell decomposition in robot motion planning, in: International Conference on Signal Processing, Robotics and Automation, Greeks, 2007, pp. 127–132.
[11] F.A. Cosio, M.A.P. Castaneda, Autonomous robot navigation using adaptive potential fields, Math. Comput. Modelling 40 (2004) 1141–1156.
[12] J. Sfei, M. Saad, H.S. Hassane, An improved artificial potential field approach to real-time mobile robot path planning in an unknown environment, in: IEEE International Symposium on Robotic and Sensors Environments, Montreal, QC, 2011, pp. 208–213.
[13] S.K. Pradhan, D.R. Parhi, A.K. Panda, R.K. Behera, Potential eld method to navigate several mobile robots, Appl. Intell. (2006) 321–333.
[14] Y. Kang, M. Lee, C. Kim, S. Yoon, C. Noh, A study of cluster robots line formatted navigation using potential field method, in: IEEE International Conference on Mechatronics and Automation, Beijing, China, 2011, pp. 1723–1728.
[15] D.H. Kim, Escaping route method for a trap situation in local path planning, Int. J. Control Autom. Syst. (2009) 495–500.
[16] D.H. Kim, S. Shin, Local path planning using a new artificial potential function configuration and its analytical design guidelines, Adv. Robot. 20 (2006) 115–135.
[17] L. Xin, Y. Yin, C.J. Lin, A new potential field method for mobile robot path planning in the dynamic environment, Asian J. Control 11 (2) (2009) 214–225.
[18] Q. Zhang, S. Yue, Q. Yin, Y. Zha, Dynamic obstacle-avoiding path planning for robots based on modified potential field method, Intell. Comput. Theor. Technol. 7996 (2013) 332–342.
[19] N.N. Singh, A. Chatterjee, A. Chatterjee, A. Rakshit, A two-layered subgoal based mobile robot navigation algorithm with vision system and IR sensors, in: Emerging Research in Artificial Intelligence and Computational Intelligence, in: Communications in Computer and Information Science, vol. 237, 2011, pp. 325–334.

[20] H. Liu, W. Wan, H. Zha, A dynamic subgoal path planner for unpredictable environments, in: IEEE International Conference on Robotics and Automation, ICRA, Anchorage, AK, 2010, pp. 994–1001.
[21] S. Candido, Y.T. Kim, S. Hutchinson, An improved hierarchical motion planner for humanoid robots, in: IEEE-RAS International Conference on Humanoid Robots, Daejeon, Korea, 2008, pp. 654–661.
[22] J. Lee, O. Kwon, L. Zhang, S.E. Yoon, A selective retraction-based RRT planner for various environments, IEEE Trans. Robot. 30 (4) (2014) 1002–1011.
[23] S. Karaman, E. Frazzoli, Sampling-based algorithms for optimal motion planning, Int. J. Robot. Res. (IJRR) 30 (7) (2011) 846–894.
[24] M. Elbanhawi, M. Simic, Sampling-based robot motion planning: A review, IEEE Access 2 (2014) 56–77.
[25] Z. Yan, N. Jouandeau, A. Cherif, ACS-PRM: Adaptive cross sampling based probabilistic road-map for multi-robot motion planning, in: Intelligent Autonomous Systems 12, in: Advances in Intelligent Systems and Computing, vol. 193, 2013, pp. 843–851.
[26] M.T. Rantanen, M. Juhola, A configuration deactivation algorithm for boosting probabilistic road-map planning of robots, Int. J. Autom. Comput. 9 (2) (2012) 155–164.
[27] A.N. Nazif, A. Davoodi, P. Pasquier, Multi-agent area coverage using a single query road-map: A swarm intelligence approach, in: Advances in Practical Multi-Agent Systems, in: Studies in Computational Intelligence, vol. 325, 2011, pp. 95–112.
[28] D. Hsu, J.C. Latombe, H. Kurniawati, On the probabilistic foundations of probabilistic roadmap planning, Int. J. Robot. Res. (IJRR) 25 (7) (2006) 627–643.
[29] A.L. Ladd, L. Kavraki, Measure theoretic analysis of probabilistic path planning, IEEE Trans. Robot. Autom. 20 (2) (2004) 229–242.
[30] C. Moon, W. Chung, Kinodynamic planner dual-tree RRT (DT-RRT) for two-wheeled mobile robots using the rapidly exploring random tree, IEEE Trans. Ind. Electron. 62 (2) (2015) 1080–1090.
[31] S. Karaman, M.W. Walter, A. Perez, E. Frazzoli, S. Teller, Anytime motion planning using the RRT*, in: IEEE International Conference on Robotics and Automation, ICRA, Shanghai, China, 2011, pp. 1478–1483.
[32] A.C. Murtra, E. Trulls, O. Sandoval, J. Perez-Ibarz, D. Vasquez, J.M. Mirats-Tur, M. Ferrer, A. Sanfeliu, Autonomous navigation for urban service mobile robots, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Taipei, Taiwan, 2010, pp. 4141–4146.
[33] E. Trulls, A. Corominas Murtra, J. Pe rez-Ibarz, G. Ferrer, Autonomous navigation for mobile service robots in urban pedestrian environments, J. Field Robot. 28 (3) (2011) 329–354.
[34] I. Ardiyanto, J. Miura, Real-time navigation using randomized kinodynamic planning with arrival time field, Robot. Auton. Syst. 60 (2012) 1579–1591.
[35] D.J. Webb, J. Berg, Kinodynamic RRT*: Asymptotically optimal motion planning for robots with linear dynamics, in: IEEE International Conference on Robotics and Automation, ICRA, Karlsruhe, Germany, 2013, pp. 5054–5061.
[36] S.M. LaValle, J.J. Kuffner, Randomized kinodynamic planning, Int. J. Robot. Res. (IJRR) 20 (5) (2001) 378–400.
[37] S. Karaman, E. Frazzoli, Optimal kinodynamic motion planning using incremental sampling-based methods, in: Proc. 49th IEEE CDC, Dec. 2010, pp. 7681–7687.
[38] B. Lau, C. Sprunk, W. Burgard, Efficient grid-based spatial representations for robot navigation in dynamic environments, Robot. Auton. Syst. 61 (2013) 1116–1130.
[39] B. Park, J. Choi, W.K. Chung, An efficient mobile robot path planning using hierarchical roadmap representation in indoor environment, in: IEEE International Conference on Robotics and Automation, ICRA, Minnesota, USA, 2012, pp. 180–186.
[40] V.R. Desaraju, J.P. How, Decentralized path planning for multi-agent teams in complex environments using rapidly-exploring random trees, in: IEEE International Conference on Robotics and Automation, ICRA, Shanghai, China, 2011, pp. 4956–4961.
[41] G.S. Aoude, B.D. Luders, D.S. Levine, J.P. How, Threat–aware path planning in uncertain urban environments, in: IEEE International Conference on Robotics and Automation, ICRA, Shanghai, China, 2011, pp. 6058–6063.
[42] S. Chamberland, E. Beaudry, L. Clavien, F. Kabanza, F. Michaud, M. Lauriay, Motion planning for an omnidirectional robot with steering constraints, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Taipei, Taiwan, 2010, pp. 4305–4310.
[43] S.X. Yang, M. Meng, An efficient neural network approach to dynamic robot motion planning, Neural Netw. 138 (2000) 143–148.
[44] D. Janglova, Neural networks in mobile robot motion, Int. J. Adv. Rob. Syst. 1 (2004) 15–22.
[45] S.H. Dezfoulian, D. Wu, I.S. Ahmad, A generalized neural network approach to mobile robot navigation and obstacle avoidance, in: Intelligent Autonomous Systems 12, in: AISC, vol. 193, 2013, pp. 25–42.
[46] M.K. Singh, D.R. Parhi, Intelligent neuron controller for navigation of mobile robot, in: International Conference on Advances in Computing, Communication and Control, 2009, pp. 123–128.
[47] D.R. Parh, M.K. Singh, Real-time navigational control of mobile robots using an artificial neural network, Proc. Inst. Mech. Eng. C 223 (7) (2009) 1713–1725.
[48] M.K. Singh, D.R. Parhi, Path optimisation of a mobile robot using an artificial neural network controller, Int. J. Syst. Sci. 42 (1) (2011) 107–120.
[49] M.A. Sagban, R. Dhaouadi, Neural-based navigation of a differential-drive mobile robot, in: Control Automation Robotics and Vision, ICARCV, 2012, pp. 353–358.
[50] I. Engedy, G. Horvath, Artificial neural network based mobile robot navigation, in: IEEE International Symposium on Intelligent Signal Processing, Budapest, Hungary, 2009, pp. 241–246.
[51] X. Chen, Y. Li, Smooth formation navigation of multiple mobile robots for avoiding moving obstacles, Int. J. Control Autom. Syst. 4 (4) (2006) 466–479.
[52] T.S. Hong, D. Nakhaeinia, B. Karasfi, Application of fuzzy logic in mobile robot navigation, in: Fuzzy Logic - Controls, Concepts, Theories and Applications, 2012, pp. 21–36.
[53] H. Chang, T. Jin, Command fusion based fuzzy controller design for moving obstacle avoidance of mobile robot, in: Future Information Communication Technology and Applications, in: Lecture Notes in Electrical Engineering, 2013, pp. 905–913.
[54] V.M. Peri, D. Simon, Fuzzy logic control for an autonomous robot, in: Fuzzy Information Processing Society, Annual Meeting of the North American, 2005, pp. 337–342.
[55] J.H. Lilly, Evolution of a negative-rule fuzzy obstacle avoidance controller for an autonomous vehicle, IEEE Trans. Fuzzy Syst. 15 (2007) 718–728.
[56] A. Foudil, F. Mohammed, E. Muhammed, H. Ramdane, A.M. Khalid, A. Mansour, M. Hassan, A hierarchical fuzzy control design for indoor mobile robot, Int. J. Adv. Rob. Syst. (2014) 1–16.
[57] L. Li, M. Zhang, L. Guo, Y. Zhao, Stereo vision based obstacle avoidance path-planning for cross-country intelligent vehicle, in: IEEE International Conference on Fuzzy Systems and Knowledge Discovery, Tianjin, China, 2009, pp. 463–467.
[58] C.H. Chao, B. Hsueh, M. Hsiao, S. Tsai, T. Li, Fuzzy target tracking and obstacle avoidance of mobile robots with a stereo vision system, Int. J. Fuzzy Syst. 11 (3) (2009) 183–191.
[59] G. Mester, Obstacle avoidance and velocity control of mobile robots, in: International Symposium on Intelligent Systems and Interpretation, Subotica, Serbia, 2008, pp. 1–5.
[60] T. Lee, C. Wu, Fuzzy motion planning of mobile robots in unknown environments, J. Intell. Robot. Syst. (2003) 177–191.
[61] R. Arajo, Prune-able fuzzy ART neural architecture for robot map learning and navigation in dynamic environments, IEEE Trans. Neural Netw. 17 (5) (2006) 1235–1249.
[62] A. Zhu, S.X. Yang, An adaptive neuro-fuzzy controller for robot navigation, in: Recent Advances in Intelligent Control Systems, 2009, pp. 277–307.
[63] M.M. Joshi, M.A. Zaveri, Reactive navigation of autonomous mobile robot using neuron-fuzzy System, Int. J. Robot. Autom. (IJRA) 2 (3) (2011) 128–145.
[64] N.B. Hui, V. Mahendar, D.K. Pratihar, Time-optimal, collision-free navigation of a car-like mobile robot using neuron-fuzzy approaches, Fuzzy Sets and Systems 157 (2006) 2171–2204.
[65] F. Valdez, P. Melin, O. Castillo, A survey on nature-inspired optimization algorithms with fuzzy logic for dynamic parameter adaptation, Expert Syst. Appl. 41 (2014) 6459–6466.
[66] M. Alajlan, A. Koubaa, I. Chaari, H. Bennaceur, A. Ammar, Global path planning for mobile robots in large-scale grid environments using genetic algorithms, in: IEEE International Conference on Individual and Collective Behaviors in Robotics, Sousse, Tunisia, 2013, pp. 1–8.
[67] Y. Hu, S.X. Yang, A knowledge based genetic algorithm for path planning of a mobile robot, in: IEEE International Conference on Robotics and Automation, ICRA, New Orleans, LA, USA, 2004, pp. 4350–4355.
[68] B.K. Oleiwi, R. Jarrah, H. Roth, B.I. Kazem, Multi objective optimization of trajectory planning of non-holonomic mobile robot in dynamic environment using enhanced GA by fuzzy motion control and A*, in: Neural Networks and Artificial Intelligence, in: Communications in Computer and Information Science, vol. 440, 2014, pp. 34–49.
[69] B. Karim, Q. Zhu, Genetic fuzzy Logic control technique for a mobile robot tracking a moving target, Int. J. Comput. Sci. Issues 10 (1) (2013) 607–613.
[70] H.C. Huang, C.C. Tsai, Global path planning for autonomous robot navigation using hybrid meta heuristic GA–PSO algorithm, in: IEEE SICE Annual Conference, SICE, Tokyo, Japan, 2011, pp. 1338–1343.
[71] S.M.R. Farshchi, S.A.N. Hoseini, F. Mohammadi, A novel implementation of G-fuzzy logic controller algorithm on mobile robot motion planning problem, Comput. Inf. Sci. 4 (2) (2011) 102–114.
[72] A.K. Karami, M. Hasanzadeh, An adaptive genetic algorithm for robot motion planning in 2D complex environments, Comput. Electr. Eng. (2015) 1–13.
[73] C. Yang, D. Simon, A new particle swarm optimization technique, in: IEEE International Conference on Systems Engineering, 2005, pp. 164–169.
[74] E. Masehian, D. Sedighizadeh, Multi–objective PSO and NPSO based algorithms for robot path planning, Adv. Electr. Comput. Eng. 10 (4) (2010) 69–76.
[75] Y. Zhang, D. Gong, J. Zhang, Robot path planning in uncertain environment using multi-objective particle swarm optimization, Neuron Comput. 103 (2013) 172–185.
[76] K.D. Sharma, A. Chatterjee, A. Rakshit, A PSO–Lyapunov hybrid stable adaptive fuzzy tracking control approach for vision-based robot navigation, IEEE Trans. Instrum. Meas. 61 (2012) 1908–1914.
[77] C. Xin, L. Yangmin, Smooth path planning of a mobile robot using stochastic particle swarm optimization, IEEE Mechatronics Autom. (2006) 1722–1727.
[78] M.S. Couceiro, J.A.T. Machado, R.P. Rocha, N.M.F. Ferreira, A fuzzified systematic adjustment of the robotic Darwinian PSO, Robot. Auton. Syst. 60 (2012) 1625–1639.
[79] B. Englot, F. Hover, Multi-goal feasible path planning using ant colony optimization, in: IEEE International Conference on Robotics and Automation, ICRA, Shanghai, China, 2011, pp. 2255–2260.

[80] R. Iser, F.M. Wahl, AntSLAM: Global map optimization using swarm intelligence, in: International Conference on Robotics and Automation, ICRA, Alaska, USA, 2010, pp. 265–272.

[81] X. Chen, Y. Kong, X. Fang, Q. Wu, A fast two–stage ACO algorithm for robotic path planning, Neural Comput. Appl. 22 (2013) 313–319.

[82] M.A.P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, Optimal path planning for autonomous mobile robot navigation using ant colony optimization and a fuzzy cost function evaluation, Adv. Soft Comput. 41 (2007) 790–798.

[83] M.A.P. Garcia, O. Montiel, O. Castillo, R. Sepulveda, P. Melin, Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation, Appl. Soft Comput. 9 (2009) 1102–1110.

[84] J. Bai, L. Chen, H. Jin, R. Chen, H. Mao, Robot path planning based on random expansion of ant colony optimization, Lect. Notes Electr. Eng. 125 (2012) 141–146.

[85] D. Zhao, J. Yi, Robot planning with artificial potential field guided ant colony optimization algorithm, Lecture Notes in Comput. Sci. 4222 (2006) 222–231.

[86] A.M. Zou, Z.G. Hou, S.Y. Fu, M. Tan, Neural Networks for Mobile Robot Navigation: A Survey, Springer–Verlag, Berlin, Heidelberg, 2006, pp. 1218–1226.

[87] E.A. Antonelo, B. Schrauwen, Supervised learning of internal models for autonomous goal-oriented robot navigation using reservoir computing, in: IEEE International Conference on Robotics and Automation, ICRA, Alaska, USA, 2010, pp. 2959–2964.

[88] A. Zhu, S.X. Yang, A fuzzy logic approach to reactive navigation of behavior-based mobile robots, in: IEEE International Conference on Robotics and Automation, ICRA, New Orleans, LA, 2004, pp. 5045–5050.

[89] Y. Tanaka, Y. Ji, A. Yamashita, H. Asama, Fuzzy based traversability analysis for a mobile robot on rough terrain, in: IEEE International Conference on Robotics and Automation, ICRA, Washington, USA, 2015, pp. 3965–3970.

[90] D.R. Parhi, Neuro-fuzzy Navigation Technique for Control of Mobile Robots, In Tech, 2008, pp. 409–432.

[91] S.K. Pradhan, D.R. Parhi, A.K. Panda, Neuro-fuzzy technique for navigation of multiple mobile robots, Fuzzy Optim. Dec. Mak. 5 (2006) 255–288.

[92] S.X. Yang, Y. Hu, M.Q.H. Meng, Knowledge based GA for path planning of multiple mobile robots in dynamic environments, in: IEEE Conference on Robotics, Automation and Mechatronics, Bangkok, Thailand, 2006, pp. 1–6.

[93] S.C. Yun, S. Parasuraman, V. Ganapathy, Dynamic path planning algorithm in mobile robot navigation, in: IEEE Symposium on Industrial Electronics and Applications, Langkawi, Malaysia, 2011, pp. 364–369.

[94] Y. Hao, W. Zu, Y. Zhao, Real-time obstacle avoidance method based on polar coordination particle swarm optimization in dynamic environment, in: 2nd IEEE Conference on Industrial Electronics and Applications, Harbin, China, 2007, pp. 1612–1617.

[95] Y. Wang, P. Chen, Y. Jin, Trajectory planning for an unmanned ground vehicle group using augmented particle swarm optimization in a dynamic environment, in: IEEE International Conference on Systems, Man, and Cybernetics, San Antonio, TX, USA, 2009, pp. 4341–4346.

[96] S.S. Ge, Y.J. Cui, Dynamic motion planning for mobile robots using potential field method, Auton. Robots 13 (2002) 207–222.

[97] F. Tan, J. Yang, J. Huang, T. Jia, W. Chen, J. Wang, A navigation system for family indoor monitor mobile robot, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS, Taipei, Taiwan, 2010, pp. 5978–5983.

[98] L. Huang, A potential field approach for controlling a mobile robot to track a moving target, in: IEEE International Symposium on Intelligent Control, Singapore, Singapore, 2007, pp. 65–70.

[99] L. Huang, Velocity planning for a mobile robot to track a moving target a potential field approach, Robot. Auton. Syst. 57 (2009) 55–63.

[100] L. Valbuena, H.G. Tanner, Hybrid potential field based control of differential drive mobile robots, J. Intell. Robot. Syst. 68 (2012) 307–322.

[101] N.B. Hui, D.K. Pratihar, A comparative study on some navigation schemes of a real robot tackling moving obstacles, Robot. Comput.-Integr. Manuf. 25 (2009) 810–828.

[102] H. Chiang, N. Malone, K. Lesser, M. Oishi, L. Tapia, Path–guided artificial potential fields with stochastic reachable sets for motion planning in highly dynamic environments, in: IEEE International Conference on Robotics and Automation, ICRA, Washington, USA, 2015, pp. 2347–2354.

Thi Thoa Mac received the B.Eng. degree in mechatronics engineering from Hanoi University of Science and Technology, Vietnam, in 2006. She obtained her master degree in 2009 at National Taiwan University of Science and Technology in Faculty of Engineering, Taiwan. Now, she is a joint Ph.D. student between Hanoi University of Science and Technology, Vietnam and Ghent University, Belgium. Her interests include fuzzy logic control, neural networks, path planning and optimization control for autonomous robot.

Cosmin Copot received his M.Sc. and M.E. degrees in systems engineering from Gheorghe Asachi Technical University of Iasi, Romania, in 2007 and 2008, respectively. In 2011, he received Ph.D. degree from the same university on control techniques for visual servoing systems. He is currently a post-doctoral researcher at Ghent University. His research interests include robotics, visual servoing systems, identification and control.

Duc Trung Tran received his doctor degree from Kharkop University, Ukraine in 1983. He is currently a professor with Hanoi University of Science and Technology, Vietnam. His research interests include optimization, Fuzzy logic control, neural networks, and hedge algebras control.

Robin De Keyser received the M.Sc. degree in electro-mechanical engineering and the Ph.D. degree in control engineering from Ghent University, Ghent, Belgium, in 1974 and 1980, respectively. He is currently a full professor of control engineering with the Faculty of Engineering, Ghent University. He is the author or co-author of over 300 publications in journals, books, and conference proceedings. He has acted as an external review expert in several European Commission Research Programs and is one of the pioneers who produced the original concepts of predictive control during the 1980s. The research is application-driven, with many pilot implementations in technical and nontechnical systems, amongst others in the chemical, steel, marine, mechatronic, semiconductor, power electronics, and biomedical spheres. His current research interests include model predictive control, auto-tuning and adaptive control, modeling and simulation, and system identification.