An Improved Ant-Driven Approach to Navigation and Map Building

Conference Paper · June 2017

DOI: 10.1007/978-3-319-61824-1_33

4 authors:

Chaomin Luo
Mississippi State University

Furao Shen
Nanjing University

Hongwei Mo

Zhenzhong Chu
Shanghai Maritime University

An Improved Ant-Driven Approach to Navigation and Map Building

Chaomin Luo[1(✉)], Furao Shen[2], Hongwei Mo[3], and Zhenzhong Chu[4]

[1] Department of Electrical and Computer Engineering, University of Detroit Mercy, Michigan, USA
luoch@udmercy.edu

[2] Department of Computer Science and Technology, Nanjing University, Nanjing, China

[3] Automation College, Harbin Engineering University, Harbin, China

[4] College of Information Engineering, Shanghai Maritime University, Shanghai, China

Abstract. An improved ant-type approach, ant colony optimization (ACO) model, integrated with a heading direction scheme (HDS) to real-time collision-free navigation and mapping of an autonomous robot is proposed in this paper. The developed HDS-based ACO model for concurrent map building and safety-aware navigation is capable of remedying the shortcoming of risky distance from obstacles in combination with the Dynamic Window Approach (DWA) algorithm as a local navigator. Its effectiveness and efficiency of the developed real-time hybrid map building and safety-aware navigation of an autonomous robot have been successfully validated by simulated experiments and comparison studies.

Keywords: ACO · Motion planning · HDS · DWA · Local navigation · Grid-based map

1 Introduction

An improved ant-driven model integrated with a Dynamic Window Approach (DWA) algorithm to real-time collision-free navigation and mapping with safety consideration is proposed in this paper. Concurrent navigation and map building of an autonomous robot under unknown environments is one of the challenges in robotics motion planning.

There have been a large number of models proposed for autonomous robot navigation with obstacle avoidance such as fuzzy logic method [8], neural networks [6,9,10], machine leaning [7], and Ant Colony Optimization (ACO) [1,5], etc. Vasak and Hvizdos [8] developed a robot navigation method deployed RFID tags with the sonar through a fuzzy logic model.

Neural networks (NNs) approach plays an increasingly crucial role on robot mapping and navigation [6,9,10]. NNs algorithm aims to navigate an autonomous robot in light of sensor information in association with a fuzzy logic

Y. Tan et al. (Eds.): ICSI 2017, Part I, LNCS 10385, pp. 301–309, 2017.
DOI: 10.1007/978-3-319-61824-1_33

controller [8]. Yi *et al.* [10] applied a bio-inspired neural network model to the 3D path planning and task assignment under uncertain circumstances. However, the map building and local navigator have not been integrated yet. Yang and Luo [9] proposed a new bio-inspired neuro-dynamics model for robot complete coverage motion planning under dynamically varying environments. However, the proposed model lacks the local reactive navigation and map building components under unknown environments [6]. Mo *et al.* proposed a machine-learning based Imitation Learning (IL) algorithm for robot navigation [7]. Luo *et al.* [5] proposed an ACO-based multi-goal navigation and mapping approach of an intelligent robot. However, the proposed model lacks local navigation.

In this paper, a two-level hybrid real-time mapping and navigation model of an autonomous robot is proposed. Top level is an improved ACO approach that plans a global trajectory. Bottom level utilizes DWA-based algorithm in light of the sensor information to direct a robot locally to autonomously traverse from one waypoint to another. In addition, in order to generate safer, more reasonable collision-free trajectories, novel heuristic algorithms are designed to optimize the trajectory. The developed real-time ACO model for concurrent map building and safety-aware navigation is capable of remedying the shortcoming of trajectories generated with risky distance from obstacles in combination with the DWA algorithm as a local navigator.

2 The HDS-Based ACO Algorithm

Ants in the ACO algorithm as intelligent agents in the navigation, which traverse from one waypoint to another directed by pheromone trails with an a *priori* available heuristic information. The field of "ant-like algorithms" creates models derived from the behavior observation of real ants, in which these models as a source of biological inspiration are utilized to design novel algorithms for the solution of optimizations.

Ant pheromone strength $\tau_{ij}(t)$, is a sort of numerical information defined with each arc (i, j) that is updated in the ACO algorithm, in which t is the iteration counter. The agent is initially arranged in a waypoint. At each iteration stage, a probabilistic action choice rule is applied to an agent, k (an autonomous mobile robot could be an agent). The probability $p_{ij}^k(t)$ of an agent k, currently at waypoint i, which moves to waypoint j at the tth iteration of the algorithm, is obtained as follows in Eq. (1) [2].

$$p_{ij}^k(t) = \frac{[\tau_{ij}(t)]^\alpha \times [\vartheta_{ij}]^\beta}{\sum_{l\in\aleph_i^k} [\tau_{il}(t)]^\alpha \times [\vartheta_{il}]^\beta} \quad \text{if} \quad j \in \aleph_i^k\,, \tag{1}$$

where $\aleph_i^k$ is the feasible adjacent waypoint of the agent k, the set of waypoints which the agent k has not yet visited. The $\vartheta_{ij} = 1/d_{ij}$ is an a *priori* available heuristic value, and d_{ij} is the distance between two waypoints. Parameter α represents importance factor of the pheromone matching a classical stochastic greedy algorithm. Parameter β is an importance factor of the heuristics function. Parameters α and β determine the relative influence of the pheromone trail and the heuristic information [2].

At each iteration stage, $\triangle\tau_{ij}^k(t)$, the amount of pheromone agent k leaves on the arcs it has visited is dynamically updated by decreasing the pheromone strength on all arcs by a constant factor before enabling each agent to supplement pheromone on the arcs. The pheromone strength τ_{ij} is dynamically updated as follows in Eq. (2) when $0 < \rho < 1$.

$$\begin{cases} \tau_{ij}(t+1) & = (1-\rho)\cdot\tau_{ij}(t) + \triangle\tau_{ij} \\ \triangle\tau_{ij} & = \sum_{k=1}^{n} \triangle\tau_{ij}^k \end{cases} \tag{2}$$

Algorithm 1. ACO algorithm for Navigation with HDS

```
1: procedure ACO WITH HDS(x, y)      ▷ Find out the trajectory (x, y coordinates)
2:     Set parameters, initialize pheromone trails
3:     while (termination condition not met) do                 ▷ reach the goal?
4:         ConstructSolutions
5:         ApplyLocalSearch
6:         HeadingDirectionScheme
7:         UpdateTrails
8:     end while
9:     return The trajectory (x, y coordinates)                 ▷ reach the goal!
10: end procedure
```

The amount of pheromone $\triangle\tau_{ij}^k(t)$ in ant cycle system mode is defined as [2]:

$$\triangle\tau_{ij}^k(t) = \begin{cases} \frac{Q}{L^k(t)} & \text{if arc } (i,j) \text{ is used by the robot } k, \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

$L^k(t)$ is the length of the kth robot's tour in this paper. The ACO algorithm for motion planning with a heading direction scheme is summarized as Algorithm 1. Note that there is a heading direction scheme (HDS) heuristic embedded in this ACO algorithm. It will be depicted in the section later.

3 Map Building and DWA-Driven Local Navigation

The proposed robot navigation system is comprised of two layers. One is an ACO global path planner whereas the other is a DWA-based local navigator. Both effectiveness and efficiency motivate ACO to be adapted to navigation and mapping as the global plan planner. The objective of the local navigator is to generate velocity commands for the autonomous mobile robot to move towards a target. In order to assure the obstacle avoidance, Fox *et al.* [3] first successfully proposed a DWA algorithm method for the local navigation. The DWA is an obstacle-avoidance approach with synchro-drives, which considers the constraints imposed by limited velocities and accelerations, derived directly from the motion dynamics of synchro-drive autonomous mobile robots.

Unlike vector field histogram (VFH) method, the DWA considers the dynamic and kinematic constraints of a mobile robot [3]. In a nutshell, only a short time interval is periodically taken into account when calculating the next steering command to decrement the tremendous complexity of the robot motion planning. Let $x(t)$ and $y(t)$ be the coordinate at time t of a mobile robot in a global coordinate system. Let $\theta(t)$ denote the heading direction (orientation). Let the triplet $x, y, \theta(t)$ denote the kinematic configuration of the robot. The translational velocity and rotational velocity of the robot at time t are described by $v(t)$, and $\omega(t)$, respectively. The robot is assumed to move with a constant velocity (v, ω) during each control loop, in which the pair (v, ω) is considered admissible, if the robot is able to stop before it reaches the closest obstacle on the corresponding curvature. Let the term $d(v, \omega)$ represent the distance to the closest obstacle on the corresponding curvature with regard to a velocity (v, ω).

The maximal admissible velocity, over a given curvature, depends really on the distance to the next obstacle over the curvature. The accelerations are denoted to be $\dot{v}_d$ and $\dot{\omega}_d$ for breakage [3]. The set V_a of admissible velocities is expressed as (4). Therefore, the robot is allowed to stop with obstacle avoidance as the V_a is the set of velocities (v, ω) in Fig. 1(a).

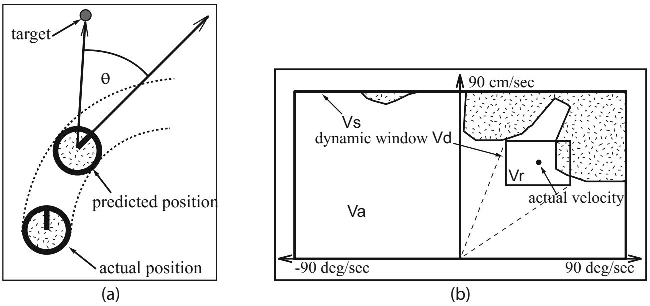

Fig. 1. The illustration of DWA algorithm (a) angle the robot's direction to the target; (b) dynamic window (redrawn from [3]).

$$V_a = \{v, \omega \mid v \leq \sqrt{2 \cdot d(v, \omega \cdot \dot{v}_d)} \wedge \omega \leq \sqrt{2 \cdot d(v, \omega \cdot \dot{\omega}_d)}\} \tag{4}$$

The dynamic window is centered on the actual velocity. All curvatures outside the dynamic window are not reachable in the next time interval, which are not taken into account for the obstacle avoidance. At the time interval t, the accelerations $\dot{v}$ and $\dot{\omega}$ will be exerted. Let (v_a, ω_a) be the actual velocity, and then the dynamic window V_d is described as (5)

$$V_d = \{(v, \omega) \mid v \in [v_a - \dot{v} \cdot t, v_a + \dot{v} \cdot t] \wedge \omega \in [\omega_a - \dot{\omega} \cdot t, \omega_a + \dot{\omega} \cdot t]\} \tag{5}$$

The area V_r within the dynamic window is generated through the restrictions imposed on the search space for the velocities depicted in Fig. 3, in which V_s is assumed to be the space of possible velocities. The area V_r is defined as the intersection of the restricted areas as expressed in Eq. (6) (see Fig. 1(b)).

$$V_r = V_a \cap V_d \cap V_s \tag{6}$$

After the generated search space V_r is determined, a velocity is selected from V_r. The maximum of the objective function $G(v, \omega)$ is calculated over V_r as Eq. (7) to integrate the criteria target heading, velocity and clearance.

$$G(v, \omega) = \rho\,[\alpha \cdot H(v, \omega) + \beta \cdot D(v, \omega) + \gamma \cdot Vel(v, \omega)], \tag{7}$$

where ρ is a factor that smooths the weighted sum of three elements of $G(v, \omega)$. The target heading function, represented by $H(v, \omega)$ is utilized to measure the alignment of the robot with the target direction, calculated by $180 - \theta$, where θ is the angle between the target point and the heading direction of a mobile robot shown in Fig. 1(b). The function velocity $Vel(v, \omega)$ evaluates the movement of the robot on the corresponding route that is a projection on the translational velocity v. The function $D(v, \omega)$ depicts the distance to the nearest obstacle intersected with the curvature.

A $2D$ grid-based map filled with equally-sized cells, marked as either occupied or free, is constructed as the mobile robot moves. The initial value is zero, which indicates that the cell is neither occupied nor unoccupied. Concurrent map building and navigation are the essence of successful navigation under unknown environments.

4 Heading Direction Motion Based Navigation

A heading direction motion planning scheme, called heading direction scheme (HDS), is developed to acquire a more reasonable and safer trajectory in the vicinity of obstacles. The fundamental principle of this scheme is to adjust the heading direction of an autonomous robot to traverse along a safe route while carrying out the ACO navigation algorithm (illustrated in Fig. 2).

The HDS scheme is designed to check the next motion of the robot. In the vicinity of obstacles, the robot is especially guided to next cell horizontally or vertically, instead of, diagonally. By means of the environmental information provided by the LIDAR sensors, the adjacent cells to obstacles on the built map are to be *known* thus accessible. For instance, an agent (ant) is located at the center of the cells next to the eight adjacent cells including free space, and obstacles shown in Fig. 2. The next position the robot will move to is likely ***a, b, c*** or ***d***. Based on the defined HDS scheme, in this kind of structure, positions indicated by red cells ***d***, is unfavorable. Therefore, it is directed to the accessible cells such as ***a, b*** and ***c*** in Fig. 2. The robot shall not move to the positions located in the cells that are not preferable to be entered. Consequently, safer and more reasonable trajectory is generated in light of both the ACO algorithm and HDS scheme. A robot initiates from the initial position ***S*** to the target ***T*** in a test scenario of workspace shown in produced safe and reasonable trajectory is illustrated in Fig. 3(a). Based on the developed heading direction scheme, the produced safe and reasonable trajectory is illustrated in Fig. 3(a).

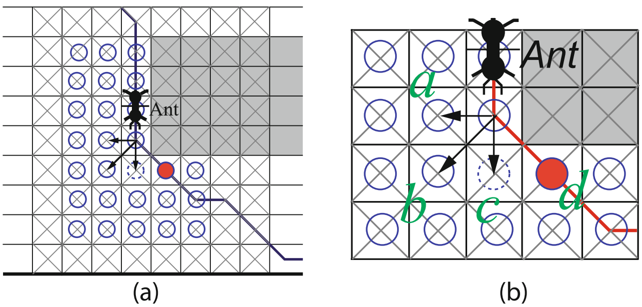

Fig. 2. Developed HDS scheme (a) the generated motion; (b) HDS in the cell structure. (Color figure online)

5 Simulation and Comparison Studies

5.1 The Hybrid Model in a Bar-Like Environment

The developed hybrid approach is applied to a test scenario with a bar-like obstacle. The developed HDS scheme associated with the ACO and DWA navigation directs an autonomous robot in the test scenario exactly identical as the one in Fig. 5 of [1]. It is now illustrated in Fig. 3(b), in order to compare our model with theirs. An ACO approach to the robot path planning in search of the final destination is developed by Chia *et al.* [1].

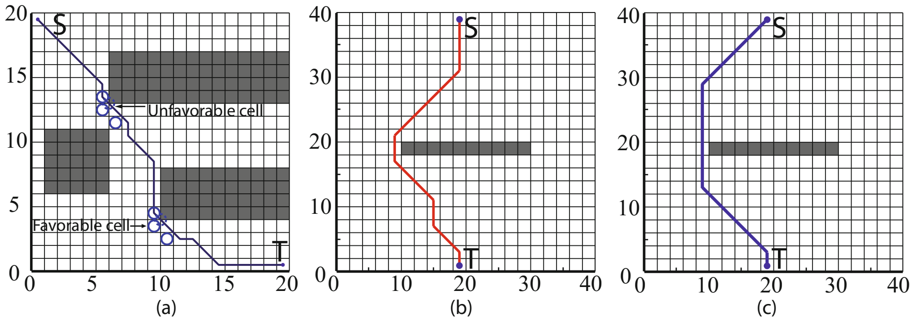

Fig. 3. Illustration of the heading direction motion scheme (a); Planned trajectory in a bar-like test scenario (b) by Chia's model of Fig. 5 (redrawn from [1]); (c) by our proposed model.

The workspace is defined with a size of 40×40, topologically organized as a cell-based map. The parameters of the ACO algorithm are selected as follows: $\alpha = 1$; $\rho = 0.3$ and $\beta = 5$. Initially, the initial position is located at $S(19, 39)$ and the robot drives toward the target at $T(19, 1)$. The trajectory planned by the model of Chia *et al.* [1] is illustrated in Fig. 3(b), whereas the trajectory produced through our ACO algorithm with HDS scheme is shown in Fig. 3(c).

It is found that the generated trajectory has the safer distance from the obstacles shown in Fig. 3 in comparison with the one generated by the model of Chia *et al.* [1]. The trajectory length, and number of turns as well as steps to complete the navigation mission are summarized in Table 1 Although the trajectory length and steps for both models are equal, with regard of the number of turns, ours is significantly better than theirs.

Table 1. Comparison of path length, steps and turns

Model	Length	Steps	Turns
Chia et al.'s model	23.14	19	6
Ours with HDS	23.14	19	3

5.2 The Hybrid Model in a Room-Like Environment

The proposed model is applied to a test scenario with populated obstacles in comparison with an identical case as Fig. 1 of [4] which is now illustrated in Fig. 4(a). Garcia *et al.* developed a hybrid model consisting of an ACO algorithm and a fuzzy logic approach [4], in which the decision-making of navigation is impacted by the distance between the source and target nodes.

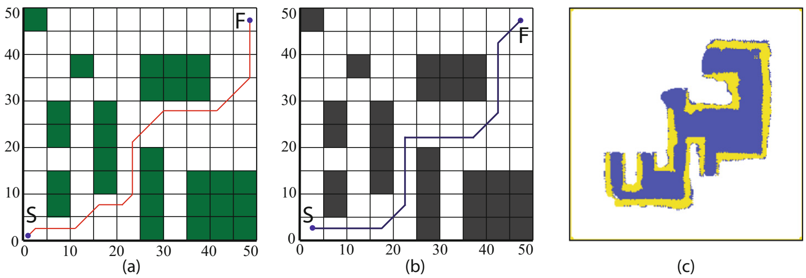

Fig. 4. Planned trajectory in a room-like test scenario (a) by Garcia *et al.*'s model of Fig. 1 (redrawn from [4]); (b) by our proposed model; (c) mapping and navigation by our proposed model.

The test scenario is a 50×50 topologically organized workspace with a grid-based map. The parameters of the ACO are chosen same as the case above with $S(1, 1)$ and $F(49, 49)$. The navigation of robot via Garcia *et al.*'s model is shown in Fig. 4(a) while the trajectory generated through our model is depicted in Fig. 4(b). The trajectory length and number of turns, steps by the proposed model and the model of Garcia *et al.* were calculated in Table 2. It reveals that the length of the trajectory by the developed ACO model is 5.17% shorter than that of their model. The number of turns of the proposed model is only 2/3 of that of their model, *i.e.*, 33.33% better than that of theirs. Additionally, the number of steps to direct the robot from the initial point to the target by the proposed ACO with HDS model is 31.82% less than that of their model. With ACO and DWA based concurrent navigation and mapping from the initial position $S(1,1)$ to the final position $F(49,49)$, The final map built when the robot achieves the final point is shown in Fig. 4(c).

Table 2. Comparison of path length, steps and turns

Model	Length	Steps	Turns
Garcia *et al.*'s model	87	22	9
Ours with HDS	82.5	15	6

6 Conclusion

A real-time ant-driven model for map building and safety-aware navigation is developed to remedy the shortcoming of trajectories generated with risky distance from obstacles in combination with the Dynamic Window Approach algorithm as a local navigator. An efficient heading-enabled ACO algorithm is created for the real-time concurrent mapping and navigation. Its effectiveness and efficiency of the developed model have been successfully validated by simulated experiments and comparison studies.

References

1. Chia, S.H., Su, K.L., Guo, J.H., Chung, C.Y.: Ant colony system based mobile robot path planning. In: Proceedings of 2010 Fourth International Conference on Genetic and Evolutionary Computing, pp. 210–213 (2010)
2. Dorigo, M., Stutzle, T.: Ant Colony Optimization. The MIT Press, San Francisco (2004)
3. Fox, D., Burgard, W., Thrun, S.: The dynamic window approach to collision avoidance. IEEE Robot. Autom. Mag. **4**(1), 23–33 (1997)
4. Garcia, M.A.P., Montiel, O., Castillo, O., Lveda, R.S., Melin, P.: Path planning for autonomous mobile robot navigation with ant colony optimization and fuzzy cost function evaluation. Appl. Soft Comput. **9**(1), 1102–1110 (2009)
5. Luo, C., Mo, H., Shen, F., Zhao, W.: Multi-goal motion planning of an autonomous robot in unknown environments by an ant colony optimization approach. In: Proceeings of the Sixth International Conference on Swarm Intelligence, pp. 519–527 (2016)
6. Luo, C., Yang, S.X.: A bioinspired neural network for real-time concurrent map building and complete coverage robot navigation in unknown environments. IEEE Trans. Neural Netw. **19**(7), 1279–1298 (2008)

7. Mo, H., Luo, C., Liu, K.: Robot indoor navigation based on computer vision and machine learning. In: Proceedings of the Sixth International Conference on Swarm Intelligence, pp. 528–534 (2016)
8. Vasak, J., Hvizdos, J.: Vehicle navigation by fuzzy cognitive maps using sonar and RFID technologies. In: Proceedings of IEEE 14th International Symposium on Applied Machine Intelligence and Informatics, pp. 75–80 (2016)
9. Yang, S.X., Luo, C.: A neural network approach to complete coverage path planning. IEEE Trans. Syst. Man Cybern. **34**(1), 718–725 (2004)
10. Yi, X., Zhu, A., Yang, S.X., Luo, C.: A bio-inspired approach to task assignment of swarm robots in 3-D dynamic environments. IEEE Trans. Cybern. **47**(4), 974–983 (2017)