

8-25-2023

Path Planning of Mobile Robots Based on Memristor Reinforcement Learning in Dynamic Environment

Hailan Yang

School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China,
yhailan163@163.com

Yongqiang Qi

School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China,
qiyongqiang@163.com

Baolei Wu

School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China

Dan Rong

School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the [Artificial Intelligence and Robotics Commons](#), [Computer Engineering Commons](#), [Numerical Analysis and Scientific Computing Commons](#), [Operations Research](#), [Systems Engineering and Industrial Engineering Commons](#), and the [Systems Science Commons](#)

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

Path Planning of Mobile Robots Based on Memristor Reinforcement Learning in Dynamic Environment

Abstract

Abstract: In order to solve the path planning problem of mobile robots in dynamic environment, two-layer path planning algorithm based on improved ant colony algorithm and MA-DQN algorithm is proposed. *Static global path planning is accomplished by ant colony algorithm that improved the probabilistic transfer function and the pheromone updating principle; the traditional DQN algorithm structure is improved by using the memristor as the synaptic structure of neural network, and then completed the local dynamic obstacle avoidance of the mobile robot. The path planning mechanism is switched according to whether there are dynamic obstacles within the sensing range of the mobile robot, so as to completed the path planning task in the dynamic environment.* The simulation results show that the algorithm can effectively plan a feasible path for mobile robots in a dynamic environment in real time.

Keywords

dynamic environment, (deep q-network)DQN, memristor, in-memory computing, path planning

Recommended Citation

Yang Hailan, Qi Yongqiang, Wu Baolei, et al. Path Planning of Mobile Robots Based on Memristor Reinforcement Learning in Dynamic Environment[J]. Journal of System Simulation, 2023, 35(7): 1619-1633.

动态环境下基于忆阻强化学习的移动机器人路径规划

杨海兰¹, 祁永强^{1*}, 吴保磊², 荣丹¹, 洪妙英¹, 王军³

(1. 中国矿业大学 数学学院, 江苏 徐州 221116; 2. 中国矿业大学, 计算机科学与技术学院, 江苏 徐州 221116;
3. 中国矿业大学 信息与控制工程学院, 江苏 徐州 221116)

摘要: 为解决动态环境下的移动机器人路径规划问题, 提出基于改进蚁群算法和基于忆阻器阵列的DQN(deep q-network)算法的双层路径规划算法。通过改进了概率转移函数和信息素更新原则的蚁群算法完成静态全局路径规划; 利用忆阻器“存算一体”的特性, 将其作为神经网络的突触结构, 改进了传统DQN算法结构, 完成移动机器人的局部动态避障。根据移动机器人感知范围内是否有动态障碍物来切换路径规划机制, 完成动态环境下的路径规划任务。仿真结果表明该算法有效可行, 能在动态环境中为移动机器人实时规划出可行路径。

关键词: 动态环境; DQN(deep q-network); 忆阻器; 存算一体; 路径规划

中图分类号: TP242; TP391 文献标志码: A 文章编号: 1004-731X(2023)07-1619-15

DOI: 10.16182/j.issn1004731x.joss.22-0334

引用格式: 杨海兰, 祁永强, 吴保磊, 等. 动态环境下基于忆阻强化学习的移动机器人路径规划[J]. 系统仿真学报, 2023, 35(7): 1619-1633.

Reference format: Yang Hailan, Qi Yongqiang, Wu Baolei, et al. Path Planning of Mobile Robots Based on Memristor Reinforcement Learning in Dynamic Environment[J]. Journal of System Simulation, 2023, 35(7): 1619-1633.

Path Planning of Mobile Robots Based on Memristor Reinforcement Learning in Dynamic Environment

Yang Hailan¹, Qi Yongqiang^{1*}, Wu Baolei², Rong Dan¹, Hong Miaoying¹, Wang Jun³

(1. School of Mathematics, China University of Mining and Technology, Xuzhou 221116, China;

2. School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China;

3. School of Information and Control Engineering, China University of Mining and Technology, Xuzhou 221116, China)

Abstract: In order to solve the path planning problem of mobile robots in dynamic environment, two-layer path planning algorithm based on improved ant colony algorithm and MA-DQN algorithm is proposed. *Static global path planning is accomplished by ant colony algorithm that improved the probabilistic transfer function and the pheromone updating principle; the traditional DQN algorithm structure is improved by using the memristor as the synaptic structure of neural network, and then completed the local dynamic obstacle avoidance of the mobile robot. The path planning mechanism is switched according to whether there are dynamic obstacles within the sensing range of the mobile robot, so as to completed the path planning task in the dynamic environment.* The simulation results show that the algorithm can effectively plan a feasible path for mobile robots in a dynamic environment in real time.

Keywords: dynamic environment; (deep q-network)DQN; memristor; in-memory computing; path planning

收稿日期: 2022-04-11 修回日期: 2022-07-07

基金项目: 国家自然科学基金(61304088); 中央高校基本科研专项基金(2013QNA37); 中国博士后科学基金(2015M581886); 非结构化环境混合感知(2020ZDPY0217); 中国矿业大学实验室开放基金(2020SYKF42); 中国矿业大学未来杰出人才助力计划(2022WLJCRCZL134)

第一作者: 杨海兰(1999-), 女, 硕士生, 研究方向为智能机器人控制。E-mail: yhailan163@163.com

通讯作者: 祁永强(1980-), 男, 副教授, 博士, 研究方向为航天器控制和智能机器人集群控制。E-mail: qiyongqiang@163.com

0 引言

随着智能化技术的快速发展, 移动机器人已广泛应用于货物搬运、智慧生产、智能生活以及异常环境探测等领域^[1-2]。移动机器人路径规划是指为移动机器人规划出一条从起点到终点安全无碰撞的最优或次优路径。采用较好的机器人路径规划技术, 可以提高移动机器人的工作效率、减少机器人的工作损耗^[3]。

针对移动机器人路径规划问题, 国内外学者做了大量工作并提出很多解决方法。如传统的 A* 算法^[4]、人工势场法^[5]、快速扩展随机树算法^[6]、蚁群算法、遗传算法等。传统算法在处理路径规划问题时, 原理简单易行、安全性高。如蚁群算法在二维静态环境中收敛速度快, 鲁棒性强。但蚁群算法不适合解决动态环境下的路径规划问题^[7]; 遗传算法具有较强的全局寻优能力, 但运算速度慢, 容易产生含有障碍物的无效路径^[8]。随着人工智能技术的发展, 深度学习^[9]和强化学习^[10]等学习算法成为路径规划问题研究的热点。

基于强化学习的路径规划方法以环境反馈作为输入, 并通过不断与环境交互、试错来寻找最优路径^[11-12]。Q-learning 算法是强化学习中一种代表性算法。在简单环境下, 移动机器人路径规划的 Q 值常以表格的形式存储计算。然而, 在动态环境中, 动作-状态维数较多, 使用表格的方式进行 Q 值计算时会导致“维数灾难”问题^[13]。基于此, Mnih 教授提出将强化学习和神经网络结合, 利用神经网络作为 Q 函数拟合器可以有效解决动态环境下的“维数灾难”问题^[14]。

深度 Q 网络(deep q-network, DQN)将神经网络作为 Q 函数拟合器, 但由于传统半导体元件实现的神经网络复杂度高, 且受冯诺伊曼结构的限制。1971 年, 文献[15]从对称性角度提出第四种基本元件——忆阻器。忆阻器具有尺寸小、能耗低和“存算一体”的特性, 可用来解决传统神经

网络电路复杂且受冯诺依曼结构限制的问题。在此基础上, 文献[16]提出将忆阻器和多层神经网络结合实现对 MNIST 数据的高效分类。文献[17]提出具有更高训练效率的忆阻器阵列和强化学习结合方案。文献[18]提出一种基于多层忆阻脉冲神经网络的强化学习算法, 并验证了其算法的有效性。但目前利用忆阻器阵列来提高路径规划效率的研究很少。

受上述研究启发, 本文提出一种基于忆阻器阵列的深度 Q 网络(memristor array deep DQN, MA-DQN), 以解决动态环境下移动机器人的路径规划问题。首先利用改进的蚁群算法完成全局静态路径规划; 再将忆阻器作为神经网络的突触结构, 通过 MA-DQN 算法对动态障碍物进行碰撞规避; 最后根据移动机器人局部环境是否有动态障碍物来切换路径规划机制并完成路径规划。通过在静态环境中设置动态障碍物的仿真实验, 验证了算法的有效性和可行性。

1 基本模型和算法

1.1 忆阻器及其原理

忆阻器是表示磁通量与电荷关系的电路器件, 它是一种具有记忆功能的非线性电阻。2008 年, 美国惠普实验室发明忆阻器实物^[19]。忆阻器模型如图 1 所示。

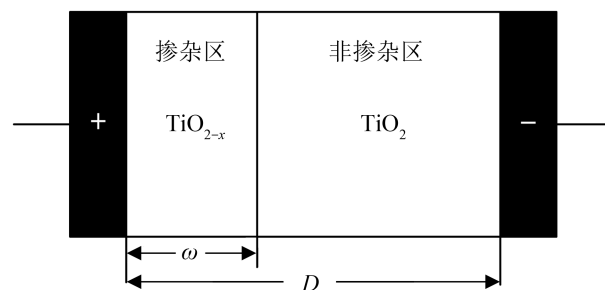


图 1 惠普忆阻器模型
Fig. 1 Model of HP memristor

图 1 中, D 表示整个忆阻器器件的长度。忆阻器掺杂区由 TiO_{2-x} 构成, 其长度为 ω , 非掺杂区由

TiO₂ 构成, 长度为 $D-\omega$ 。HP 忆阻器数学模型表示为

$$V(t) = \left[R_{\text{on}} \frac{\omega(t)}{D} + R_{\text{off}} \left(1 - \frac{\omega(t)}{D} \right) \right] I(t) \quad (1)$$

$$\frac{d\omega(t)}{dt} = \mu_v \frac{R_{\text{on}}}{D^2} i(t) \quad (2)$$

$$R(t) = R_{\text{on}} \frac{\omega(t)}{D} + R_{\text{off}} \left(1 - \frac{\omega(t)}{D} \right) \quad (3)$$

$$G(t) = \frac{1}{R(t)} \quad (4)$$

式中: $V(t)$ 为 t 时刻的电压; R_{on} 为忆阻器的最小阻值; R_{off} 为忆阻器的最大阻值; $I(t)$ 为 t 时刻的电流; $\omega(t)$ 为 t 时刻掺杂区的宽度; μ_v 为离子在均匀场中的运动漂移率; $R(t)$ 为 t 时刻忆阻器的阻值; $G(t)$ 为 t 时刻忆阻器电导值。

忆阻器掺杂区和非掺杂区的分界面在漂移到忆阻器边界时, 漂移速度会变慢。为了更好地描述忆阻器的这种非线性特征, 文献[20]引入 Joglaker 窗函数来表示忆阻器的非线性特性,

$$f(x) = 1 - (2x - 1)^{2p} \quad (5)$$

式中: x 为掺杂区在整个忆阻器元件中所占的比值, 即 ω/D , 其取值范围为 $0 \leq x \leq 1$; p 为窗函数非线性程度; $f(x)$ 为窗函数。

本文采用的模型^[21]表示为

$$R(x) = R_{\text{on}} x + R_{\text{off}} (1 - x) \quad (6)$$

$$\frac{dx}{dt} = \begin{cases} \frac{\mu_v R_{\text{on}}}{D^2} i(t) f(x), & |V| > V_{\text{th}} \\ 0, & \text{其他} \end{cases} \quad (7)$$

式中: V_{th} 为阈值电压。

1.2 忆阻神经网络

忆阻器具有“存算一体”、尺寸小、能耗低的特性, 可以高密度地集成在交叉阵列上。图2所示是一个忆阻器交叉阵列, 字线和位线由忆阻器交叉相连。由基尔霍夫定律可知, 当在字线1和字线2上分别输入电压 V_1 和 V_2 , 经过忆阻器 G_1 和 G_2 后, 位线输出为电导乘电压得到的两个电流 I_1 和 I_2 之和, 这样就实现了一次乘加操作。

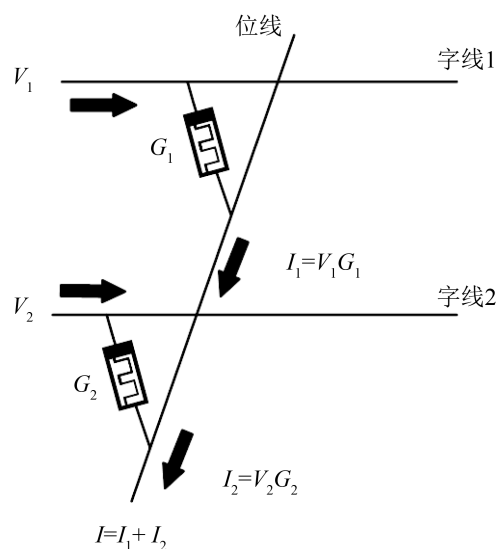


图2 忆阻器交叉阵列
Fig. 2 Memristor cross array

由于神经网络的计算方式普遍为点积运算, 这种计算方式与忆阻器交叉阵列乘加方式高度相似。因此, 本文利用由忆阻器交叉阵列构建的神经网络来替代传统神经网络。这种架构方式在计算时不需要数据在存储器件和计算器件之间来回搬移, 可有效提高神经网络的训练效率。

1.3 Q-learning 算法

本文采用强化学习中基于价值迭代 Q-learning 算法。图3所示为 Q-learning 算法和环境的交互过程。



图3 Q-learning 交互过程
Fig. 3 Process of Q-Learning

Q-learning 算法是对 $Q(s_t, a_t)$ 进行迭代, 初始化 Q 值为 0 或者随机数值, 移动机器人在当前状态 s_t 选择动作 a_t 时, 环境会转换到新状态 s_{t+1} 并反馈相应的奖励 r_t 。然后通过即时奖励 r_t 对 Q 值表进

行更新，一直重复迭代直到 Q 值表收敛。 $Q(s_t, a_t)$ 的计算公式为

$$Q(s_t, a_t) = r_t + \gamma \sum_{s_{t+1}} P_{a_t}(s_t, s_{t+1}) \max_a Q(s_{t+1}, a) \quad (8)$$

式中： $Q(s_t, a_t)$ 为在状态 s_t 采取动作 a_t 能够获得的期望收益； r_t 为 t 时刻的即时奖励； $P_{a_t}(s_t, s_{t+1})$ 为在当前状态 s_t 和动作 a_t 确定的情况下转移到状态 s_{t+1} 的概率； $\max_a Q(s_{t+1}, a)$ 为在状态 s_{t+1} 时采取不同的动作所能得到的最大奖励值； γ 为折扣因子，表示未来奖励对移动机器人当前动作的影响程度。

$Q(s_t, a_t)$ 的更新公式为

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha [r_t + \gamma \max_a Q(s_{t+1}, a) - Q(s_t, a_t)] \quad (9)$$

式中： α 为学习率，其取值范围为 $0 \leq \alpha \leq 1$ 。

2 动态环境建模

动态环境建模是解决动态环境下机器人路径规划问题的首要任务，也是研究的难点和热点。文献[22]在静态栅格地图中加入动态障碍物，并规定动态障碍物的移动速率和方向，利用改进的蚁群算法完成了动态避障。文献[23]在栅格地图的基础上设置未知的动态障碍物，通过传感器感知动态障碍物状态，并完成动态障碍物避障。受此启发，本文针对移动机器人的动态环境，采用栅格法对全局静态环境进行建模，如图4所示。

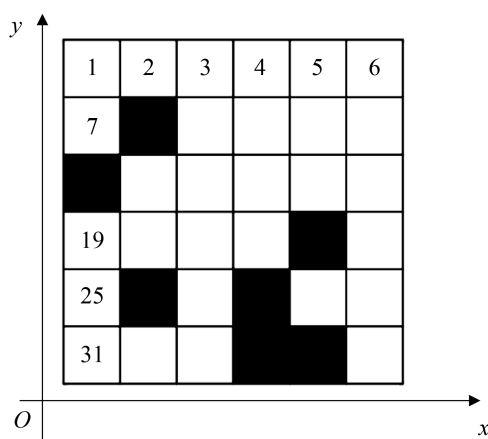


图4 栅格法环境建模图

Fig. 4 Diagram of grid method environment modeling

图4中，左下角为坐标原点，横轴为 x 、纵轴为 y ，依次对栅格地图中的栅格从上到下、从左到右进行编号 $1, 2, 3, \dots, n$ ，每个栅格都有对应的编号和坐标。 W 为栅格地图的宽度，每个小栅格的宽度为1。 $W \times W$ 的栅格地图中第 x 行、第 y 列的栅格记为 (x, y) ，栅格序号为 i ，则栅格序号 i 与坐标 (x, y) 之间的对应关系为

$$\begin{cases} x_i = \text{mod}(i/W) - 0.5 \\ y_i = W + 0.5 - \text{ceil}(i, W) \end{cases} \quad (10)$$

式中： $\text{mod}()$ 为取余函数； $\text{ceil}()$ 为向上取整函数。

DQN算法在解决路径规划问题时，移动机器人需要大量的时间和计算量来获得所需经验，算法前期的学习速度慢。蚁群算法在处理二维静态环境路径规划问题时具有较强的鲁棒性，但在动态环境中，蚁群算法易于陷入局部最优且收敛速度慢。因此，本文利用改进的蚁群算法在静态全局环境中规划出路径信息作为先验信息，并结合MA-DQN算法完成局部动态避障。针对环境中的静态和动态障碍物作出如下假设。

(1) 移动机器人被视为质点，移动方向为上、下、左、右。移动机器人已知起点、终点及静态障碍物信息。

静态障碍物集合为

$$O_s = \{s_1, s_2, \dots, s_{N_s}\}$$

式中： s_i 为第 i 个静态障碍物在栅格图中的序号； N_s 为静态障碍物的数量。

(2) 移动机器人可自由遍历的空间地图 G 由节点 C_f 和路径 ε 构成，定义为

$$G = (C_f, \varepsilon)$$

式中： $C_f = \{c_1, c_2, \dots, c_{N_f}\}$ 为移动机器人可自由遍历的 N_f 个节点的集合， ε 为可遍历空间中的路径。 $e_{ij} = (c_i, c_j) \in \varepsilon$ 表示地图中节点 c_i 和 c_j 之间的无障碍路径。

(3) 蚁群算法在全局静态环境中规划出的最优路径记为

$$g^* = \{c_{\text{start}}, g_1, \dots, c_{\text{goal}}\}$$

式中: c_{start} 为起点, c_{goal} 为终点。对 $\forall k, g(k) \in C_f$ 。

(4) 机器人在行进过程中, 不断感知外界环境, 得到所处环境的局部视野图, 其大小为 $W_l \cdot W_r$ 。

机器人在 t 时刻的局部观测值为

$$P_t = \{p_t^f, p_t^s, p_t^m, p_t^g\}$$

式中: p_t^f, p_t^s, p_t^m 及 p_t^g 分别为该视野内可自由遍历的栅格、静态障碍物、动态障碍物及改进的蚁群算法路径信息。

(5) t 时刻 N 个动态障碍物的集合为

$$O_m(t) = \{m_1(t), m_2(t), \dots, m_N(t)\}$$

式中: $m_i(t)$ 为 t 时刻动态障碍物 m_i 在栅格图中的序号。

动态障碍物的移动满足以下基本条件:

$$\begin{cases} m_i(t) \in C_f \\ (m_i(t), m_i(t+1)) \in \varepsilon \\ m_i(t) \neq m_j(t) \end{cases} \quad (11)$$

本文设置动态障碍物针对某一固定方向进行往返运动或原地等待, 但每个障碍物可选择 $v=1$ 或 $v=2$ 速度行走。在 t 时刻, 动态障碍物 m_i 的速度为 v_i 。动态障碍物在行进过程中可能会发生碰撞, 为此本文设计如下冲突消解原则:

情况 1: 相邻时间节点的两个动态障碍物 m_i 和 m_j (其中 $i \neq j$) 不能互换位置。即

$$m_j(t+1) \neq m_i(t), m_i(t) = m_j(t+1)。$$

情况 2: 若动态障碍物在某一时刻 t 停止移动 $m_i(t) = m_i(t+1)$, 则该动态障碍物不能占用终点位置。即

$$m_i(t+1) \neq c_{\text{goal}}, m_i(t) = m_i(t+1)。$$

情况 3: 同一行或同一列动态障碍物的相向冲突表示为

$$\begin{cases} m_i(t) \neq m_j(t) \\ m_i(t+1) = m_j(t+1) \\ \cos(v_i, v_j) = -1 \end{cases} \quad (12)$$

式中: $\cos(v_i, v_j)$ 为动态障碍物 m_i 和动态障碍物 m_j 速度夹角的余弦值。如图 5 所示为动态障碍物的相向冲突图。其中, 圆圈表示冲突位置, 箭头为

动态障碍物运动方向。为了避免动态障碍物碰撞, 本文设置序号优先原则的冲突消解原则, 表示为

$$\begin{cases} m_i(t), m_i(t) > m_j(t) \\ m_j(t), m_i(t) < m_j(t) \end{cases} \quad (13)$$

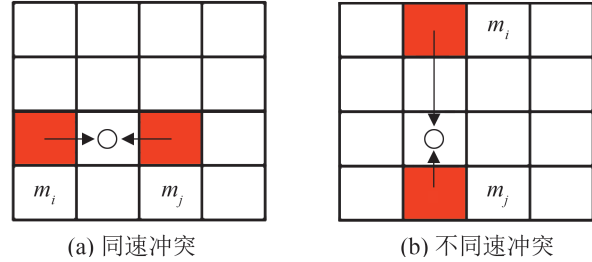


图 5 动态障碍物相向冲突

Fig. 5 Diagram of dynamic obstacles colliding in opposite directions

图 5(a) 中, 动态障碍物的移动速度相同, 图 5(b) 中, 障碍物 m_i 的移动速率是 m_j 的 2 倍。根据式 (12) 可判断动态障碍物发生相向冲突, 则依据式 (13), 选择 m_j 移动到冲突点位置, m_i 反向移动到其他位置或者原地等待。

情况 4: 不同行不同列动态障碍物的侧向冲突表示为

$$\begin{cases} m_i(t) \neq m_j(t) \\ m_i(t+1) = m_j(t+1) \\ \cos(v_i, v_j) = 0 \end{cases} \quad (14)$$

图 6 所示为动态障碍物的侧向冲突图。动态障碍物根据式 (14) 判断发生侧向冲突, 并依据公式 (13) 避免冲突, 选择 m_j 移动到冲突点位置, m_i 反向移动到其他位置或者原地等待。

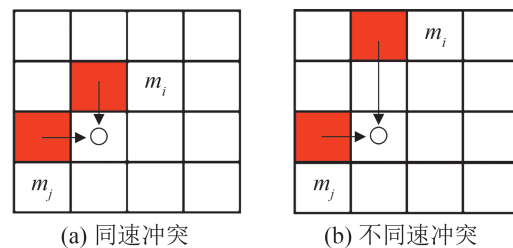


图 6 动态障碍物侧向冲突

Fig. 6 Diagram of dynamic obstacle colliding in lateral directions

情况 5: 同一行或同一列动态障碍物的同向冲突表示为

$$\begin{cases} m_i(t) \neq m_j(t) \\ m_i(t+1) = m_j(t+1) \\ \cos(v_i, v_j) = 1 \end{cases} \quad (15)$$

图 7 所示为动态障碍物的同向冲突示意图。为了避免动态障碍物碰撞, 后方动态障碍物减速排队。图 7 中后方障碍物 m_j 原地等待或减速移动, m_i 移动到冲突点。

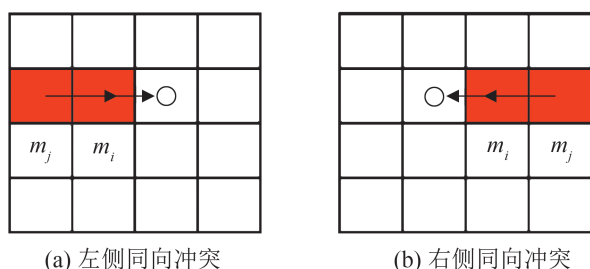


图 7 动态障碍物同向冲突

Fig. 7 Diagram of dynamic obstacles in the same direction

情况 6: 动态障碍物匀速移动但前方障碍物忽然停止移动, 如图 8 所示。移动障碍物 m_i 和 m_j 均以速度 v 移动, 但前方障碍物 m_i 忽然停止移动 $m_i(t) = m_i(t+1)$, 则障碍物 m_j 也应原地等待, 即 $m_j(t) = m_j(t+1)$, $m_i(t) = m_i(t+1)$ 。

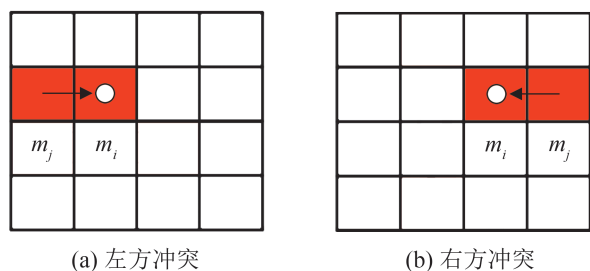


图 8 动态障碍物停止移动

Fig. 8 Diagram of dynamic obstacle stops moving

3 动态环境下基于改进的蚁群算法和 MA-DQN 算法的多层路径规划

本文主要研究环境中同时存在已知静态障碍物和未知动态障碍物情况下的路径规划, 采用静态全局规划和动态局部避障相结合的思想, 完成

动态环境下路径规划任务, 具体思路如下:

步骤 1: 根据改进的蚁群算法完成静态全局环境的路径规划任务, 并将此路径信息作为全局规划路径。

步骤 2: 移动机器人在沿全局规划路径行进过程中, 通过传感器获取周围局部实时环境信息, 当传感器感知范围内存在动态障碍物时, 采用 MA-DQN 算法进行局部动态避障。移动机器人传感范围内仍有动态障碍物, 则重复步骤 2, 继续使用 MA-DQN 算法进行局部动态避障, 直到传感器感知范围内没有动态障碍物为止; 若移动机器人的局部感知范围内无动态障碍物, 则转移到步骤 3。

步骤 3: 移动机器人在完成局部动态避障之后, 可能已经偏离初始全局规划路径。因此, 移动机器人采用改进的蚁群算法将起点位置转换为当前移动机器人所在位置, 重新进行全局静态环境下的路径规划, 更新静态环境下的路径。

步骤 4: 移动机器人沿更新后的全局规划路径寻找目标点。若移动机器人行进过程中遇到动态障碍物, 则返回步骤 2; 若移动机器人到达目标点, 路径规划任务结束。

3.1 改进蚁群算法的静态全局路径规划

传统蚁群算法在解决路径规划问题时, 具有鲁棒性强、全局寻优能力强等优点, 但也存在收敛速度慢, 易于陷入局部最优, 拐弯次数多等缺点。本文主要针对移动机器人收敛速度慢, 拐弯次数多等问题做出如下改进。

(1) 改进启发函数及转移概率

本文在传统启发函数的基础上, 引入下一节点 j 到目标点 c_{goal} 的欧式距离 $d_{j,goal}$ 来改进启发函数, 表示为

$$\eta_{ij} = \frac{1}{d_{ij} + d_{j,goal}} \quad (16)$$

式中: η_{ij} 为蚁群算法的启发函数; d_{ij} 为当前节点 i 到下一节点 j 的欧式距离。为了避免蚁群算法规划

出的路径拐弯次数过多, 降低移动机器人工作效率。本文在当前节点*i*选择下一节点*j*的概率时, 引入从起点到下一节点*j*的转弯次数*r*, 概率转移函数表示为

$$p_{ij}^k(t) = \begin{cases} \frac{[\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta \left[\frac{1}{r}\right]^\kappa}{\sum_{j \in allow_k} [\tau_{ij}(t)]^\alpha [\eta_{ij}(t)]^\beta \left[\frac{1}{r}\right]^\kappa}, & j \in allow_k \\ 0, & otherwise \end{cases} \quad (17)$$

式中: $p_{ij}^k(t)$ 为在*t*时刻蚂蚁*k*从节点*i*转移到节点*j*的概率; $allow_k$ 为下一节点*j*可达的所有节点的集合; τ_{ij} 为节点*i*到节点*j*之间信息素的大小; α 为信息素启发因子; β 为距离启发因子; κ 为转弯启发因子。

(2) 改进信息素更新原则

传统蚁群算法在信息素更新时, 考虑全局信息素更新, 即从起点到终点的信息素。而本文由于每次使用蚁群算法更新路径时, 起点位置都不相同。因此本文中的信息素更新只发生在局部动态规划后移动机器人的当前位置到目标点的路径上。改进蚁群算法流程图如图9所示。

3.2 MA-DQN 算法的局部动态路径规划

MA-DQN算法是将忆阻器交叉阵列和DQN算法结合, 来提高移动机器人动态避障的效率。MA-DQN算法解决移动机器人局部动态路径规划的主要思路如下: 首先, 采用异位训练的方式训练MA-DQN; 其次, 神经网络的权重训练收敛之后, 将权重导入忆阻器交叉阵列中; 最后, 利用忆阻器“存算一体”特性, 快速完成移动机器人的路径规划任务。

忆阻神经网络模型由卷积层结合两层全连接层构成, 训练方式采用异位训练的方法。忆阻神经网络作为*Q*函数的逼近器, 输入为机器人的局部环境信息, 输出为与环境交互得到的*Q*值。

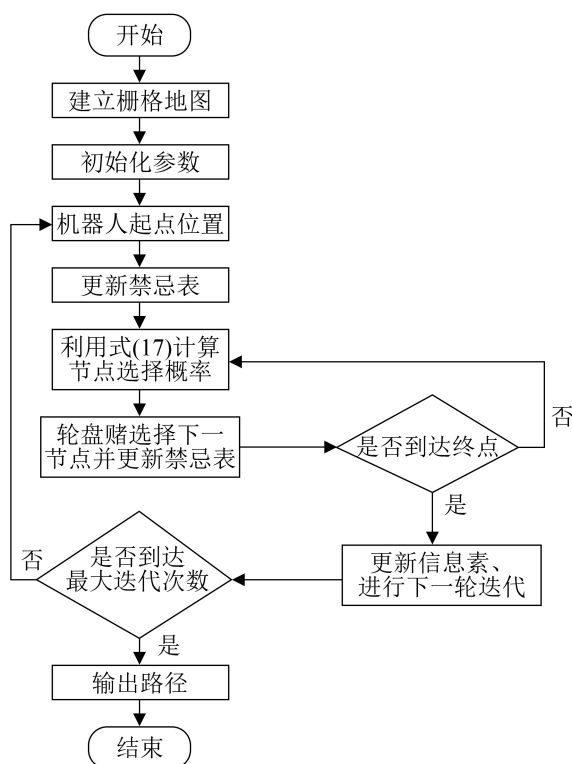


图9 改进蚁群算法流程图
Fig. 9 Flowchart of improved ant colony algorithm

移动机器人通过传感器可以得到大小为 $W_1=4$ 的局部环境信息, 将局部环境信息映射为一个 4×4 的矩阵信息输入MA-DQN算法。

由于输入样本前后之间具有很强的相关性, DQN算法直接从连续的数据中学习, 会导致神经网络学习效率低下。MA-DQN算法引入了经验池来降低样本之间的高度相关性, 使样本满足独立同分布, 以提高算法的鲁棒性。下面阐述MA-DQN算法的异位训练过程。

移动机器人在每次选择动作时, 以 ϵ 的概率随机探索环境, 以 $1-\epsilon$ 的概率通过MA-DQN算法和环境交互, 依据*Q*值选择状态*s_t*下对应的动作*a_t*。移动机器人选择动作*a_t*并得到新的状态*s_{t+1}*及奖励*r_t*, 并将(*s_t*, *a_t*, *r_t*, *s_{t+1}*)添加到经验池中。

MA-DQN算法需要构造合理的奖励函数*r_t*, 使得机器人能够规避动态障碍物并到达目标节点。为了有效地进行动态规划, 本文在构造奖励函数时, 考虑了由蚁群算法规划出来的静态路径信息,

表示为

$$r_t = \begin{cases} 0.01, & c_r(t+1) \in C_f \cup g^* \\ 0, & c_r(t+1) \in (C_f \setminus g^*) \cup (C_f \setminus O_m) \\ -1, & c_r(t+1) \in O_s \cup O_m \\ 1, & c_r(t+1) = C_{goal} \end{cases} \quad (18)$$

式中： $c_r(t+1)$ 为移动机器人在 t 时刻采取动作 a_t 后到达的位置； g^* 为在静态环境下利用蚁群算法得到的全局规划路径信息； O_s 和 O_m 分别为环境中的静态障碍物及动态障碍物信息； C_f 为移动机器人运动环境中可自由遍历的空间。

MA-DQN算法在每次训练时会随机从经验池中提取小部分样本 mini-batch，提取样本形式为 (s_j, a_j, r_j, s_{j+1}) ，其中， $j \leq t$ (t 表示当前时间)。MA-DQN算法将提取出的 mini-batch 样本进行前向传递。其状态-动作价值为

$$y_j = \begin{cases} r, & \text{在 } j+1 \text{ 步时本轮迭代结束} \\ r_j + \gamma \max_{a_{j+1}} Q(s_{j+1}, a_{j+1}, \theta), & \text{其他} \end{cases} \quad (19)$$

损失函数 L 表示为

$$L = \sum_{j \in \text{mini-batch}} (y_j - Q(s_j, a_j, \theta))^2 \quad (20)$$

通过误差反向传播过程来训练神经网络，直到移动机器人到达目标点，训练结束。MA-DQN算法训练的伪代码如下：

初始化经验池 D ，容量为 N

初始化忆阻神经网络权重和电导

初始化学率 α 、折扣因子 γ 、探索因子 ϵ

for episode = 1, 2, ..., M **do**

for $t = 1, 2, \dots, T$ **do**

 产生一个0~1之间的随机数 p

if $p < \epsilon$

 以概率 ϵ 从基本的动作上、下、左、右中随机选择一个

else

 选择动作 $\max_a Q(s_t, a)$

end

 执行动作 a_t ，得到奖励 r_t ，以及新的状态 s_{t+1}

将数据样本 (s_t, a_t, r_t, s_{t+1}) 存入 D 中

从经验池中随机采取样本 (s_j, a_j, r_j, s_{j+1})

$$y_j = \begin{cases} r_j, & \text{样本是本轮迭代的最后一步} \\ r_j + \gamma \max_{a_{j+1}} Q(s_{j+1}, a_{j+1}, \theta), & \text{其他} \end{cases}$$

利用式(20)计算误差

利用误差反向传播计算权重的误差梯度并更新权重

end for

end for

MA-DQN算法通过异位训练得到神经网络的权重，将权重映射到忆阻器阵列的电导中。MA-DQN算法将 4×4 的局部环境的矩阵信息展开成 16×1 的输入向量，并将输入向量转化为对应的脉冲电压输入忆阻器阵列中。

由于忆阻器的电导只能为正数，因此本文采用1T1R结构的忆阻器阵列，以方便表示负权重，即每个权重由一对差分忆阻器来表示，具体结构如图10所示。

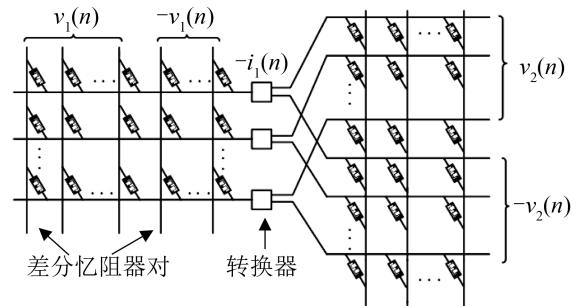


图 10 忆阻器阵列实现神经网络结构图

Fig. 10 Diagram of memristor array implementation neural network structure

图10中，第一层忆阻神经网络的输入电压由输入向量转化而来，下一层的输入电压是由上一层的输出电流经放大器转换而来。模拟加权在忆阻器阵列中实施，第 l 层神经元的输入电压向量 V_l 和输出电流向量 I_l 之间的关系为

$$I_l = W_l V_l \quad (21)$$

$$I_j^l = \sum_{i=1}^n w_{ij}^l V_i^l = \sum_{i=1}^n [G_{ij}^+ \cdot V_i^l + G_{ij}^- \cdot V_i^l] \quad (22)$$

式中: W_l 为第 l 层的权值矩阵; I_j^l 为第 l 层神经网络中第 j 行的输出电流, w_{ij}^l 为第 l 层神经元中第 i 行第 j 列权重; G_{ij}^+ , G_{ij}^- 分别为正、负忆阻器阵列中第 i 行第 j 列的电导值; V_i^l 为第 l 层神经元中第 i 行的输入电压。由式(22)可推出传统神经网络权重和忆阻器阵列中各忆阻器电导之间的对应关系为

$$w_{ij} = G_{ij}^+ - G_{ij}^- \quad (23)$$

依据忆阻器阵列的结构特征, 本文将输入向量复制, 产生 32 个模拟电压, 来对应负权重。输出神经元为动作对应的 4 个神经元。

为了将权重映射为忆阻器阵列的电导值, 将神经网络的权值矩阵 W 分为两个矩阵 W^+ 和 W^- , 其中, W^+ 的元素是将 W 中非负位置的权值存储到 W^+ 对应位置, 其他位置元素存储为 0, W^- 的元素是将 W 中负位置的权值绝对值存储到 W^- 对应位置, 其他位置元素存储为 0。依据本文中的电路结构, 忆阻器突触与权重对应编码方案为^[24]

$$G_{ij}^+ = \frac{G_{\max} - G_{\min}}{\max |W|} W_{ij}^+ + G_{\min} \quad (24)$$

$$G_{ij}^- = \frac{G_{\max} - G_{\min}}{\max |W|} W_{ij}^- + G_{\min} \quad (25)$$

式中: W_{ij}^+ 为矩阵 W^+ 中第 i 行第 j 列权重, W_{ij}^- 为矩阵 W^- 中第 i 行第 j 列权重, G_{\max} 和 G_{\min} 分别为忆阻器的最大电导值及最小电导值。偏置通过软件进行计算。

通过这种对应关系, 神经网络训练好的权值可以输入到忆阻器阵列中, 加速神经网络的计算, 完成移动机器人的路径规划任务。MA-DQN 算法实现路径规划具体流程如图 11 所示。

4 实验仿真

为验证本文提出算法在处理动态环境中路径规划问题中的可行性。本文算法运行环境为 CPU 2.50 GHz, i7 处理器, 编程环境为 Matlab2021a。

首先对静态全局环境下的蚁群算法进行仿真验证, 本文改进的蚁群算法各项基本参数如表 1 所示。

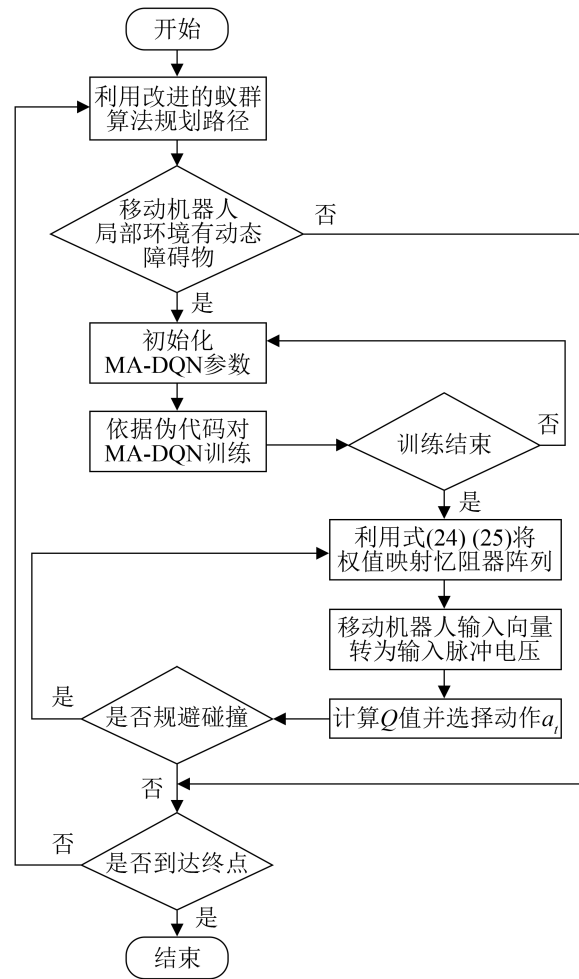


图 11 MA-DQN 实现路径规划流程图

Fig. 11 Flowchart of MA-DQA algorithm path planning

表 1 改进的蚁群算法参数

Table 1 Improved ant colony algorithm parameters

参数	数值
蚂蚁数目 m	50
最大迭代次数 N_{\max}	100
信息素启发因子 α	1
距离启发因子 β	7
拐弯启发因子 κ	1
挥发系数 ρ	0.5
信息素强度 Q_1	10

本文机器人运动环境设置为 16×16 的栅格环境, 并定义起点位置为 (0.5, 15.5), 目标点位置为 (15.5, 0.5)。通过实验仿真, 在全局静态环境下, 利用改进的蚁群算法及传统蚁群算法得到的初始路径对比图及迭代收敛对比如图 12 所示。

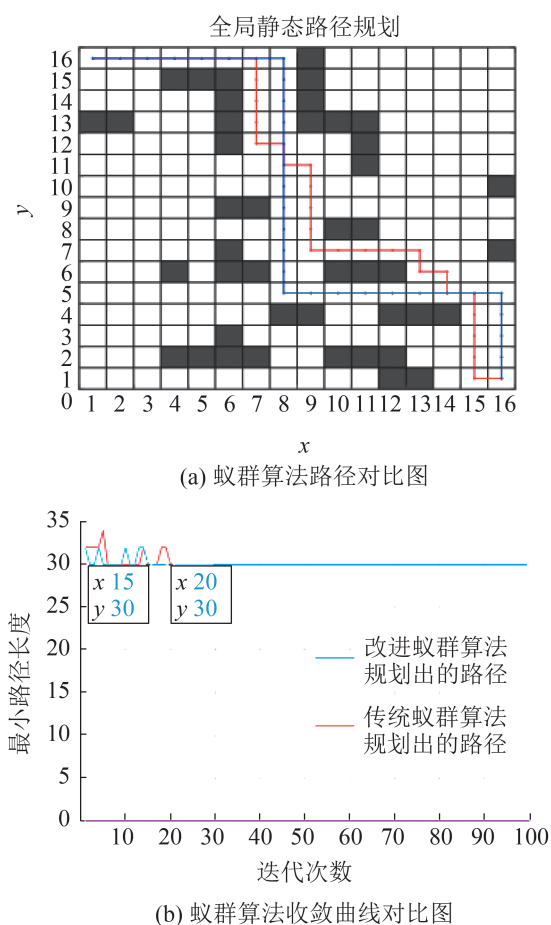


图 12 蚁群算法路径规划及收敛曲线对比图
Fig. 12 Flowchart of MA-DQA algorithm path planning

图 12 中，由于移动机器人的移动方向为上下左右，因此两种方法得到的路径长度相等。改进的蚁群算法迭代 15 次就能得到最优路径，而传统蚁群算法需要迭代 20 次，因此改进的蚁群算法相比于传统蚁群算法收敛速度更快。同时，改进的蚁群算法规划出的路径拐点更少，更有利于提高机器人的移动效率。

为了验证本文提出的算法能够有效进行动态避障，移动机器人在上述 16×16 的栅格环境中进行移动，设置起点位置为(0.5, 15.5)，终点位置为(15.5, 0.5)。基于 MA-DQN 算法进行局部动态避障，本文首先对单个忆阻器进行实验仿真，单个忆阻器模型的参数如表 2 所示。

当忆阻器中输入电压 $v = \sin(2\pi ft)$ 时，可以得

到相应的输出电流，如图 13 所示。此时，忆阻器的伏安特性曲线出现“8”字型回线，表明本文所建立的忆阻器模型满足忆阻器的基本特性。

表 2 忆阻器参数
Table 2 Memristor parameters

参数	数值
最小阻值 R_{on}/Ω	100
最大阻值 R_{off}/Ω	2 000
忆阻器初值 x_0	0.1
线性漂移系数 $\mu_v / (\text{m}^2 \cdot \text{s}^{-1} \cdot \text{V}^{-1})$	10^{-14}
忆阻器长度 D/nm	10
电压阈值 V_{th}/V	0.1
窗函数系数 p	5
频率 f/Hz	6

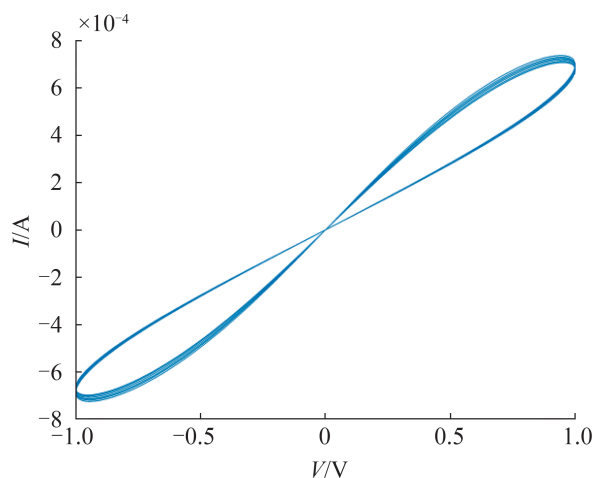


图 13 忆阻器的伏安特性曲线
Fig. 13 Voltammetric characteristic curve of memristor

忆阻器在神经网络中能够作为突触，主要是由于忆阻器的“存算一体”特性不仅能够保存神经网络的权重，而且还能够模拟突触的特性进行计算。通过对忆阻器模型的介绍可知，忆阻器阻值状态会根据施加在其两端的电压而发生改变。忆阻器阻值变化与施加在其两端电压的大小与极性有关。在 MATLAB 环境中给忆阻器施加脉冲信号如图 14(a) 所示，忆阻器阻值变化如图 14(b) 所示。

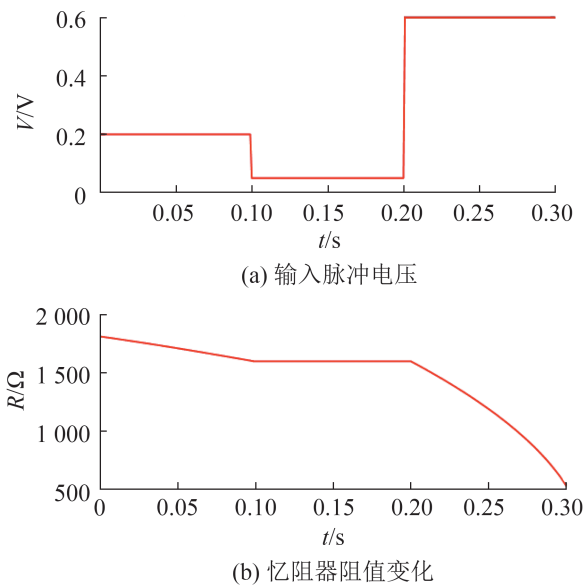


图 14 不同电压下忆阻器阻值的变化
Fig. 14 Change in resistance of memristor at different voltages

由表 2 可知, 本文忆阻器阈值为 0.1 V, 当输入电压大于 0.1 V 时, 忆阻器阻值发生改变, 当输入电压小于 0.1 V 时, 忆阻器阻值恒定不变。通过图 14 可知, 在 0~0.1 s 内输入大小为 0.2 V 的电压, 忆阻器阻值降低; 在 0.1~0.2 s 内, 输入大小为 0.05 V 的电压, 忆阻器阻值恒定不变; 在 0.2~0.3 s 内, 输入 0.6 V 的电压, 忆阻器阻值继续发生改变。这证明忆阻器模型具有“存算一体”的特性。

本文的仿真设计了局部环境矩阵到脉冲的产生单元, 脉冲产生单元的主要作用是将移动机器人的局部环境矩阵信息映射为电压脉冲序列, 进而输入到 MA-DQN 模型中去。环境信息中的无障碍区域用 0 来表示并映射为 0.2 V; 静态障碍物用 1 表示并映射为 0.4 V; 动态障碍物用 2 表示并映射为 0.6 V; 蚁群算法规划出的全局路径信息用 3 表示并映射为 0.8 V, 脉冲时间均为 1 ms。

本文利用一个 2×2 的环境地图来演示脉冲电压的生成过程, 如图 15 所示。2×2 的栅格环境中白色表示可自由遍历的栅格, 黑色表示静态障碍物, 黄色表示改进的蚁群算法规划出的路径信息, 红色表示动态障碍物。此时, 移动机器人的输入向量可表示为 [0, 1, 2, 3]^T, 则输入忆阻器阵列中的脉冲电压

大小为 [0.2, 0.4, 0.6, 0.8, -0.2, -0.4, -0.6, -0.8]^T, 每个电压的输入时间均为 1 ms, 如图 16 所示。

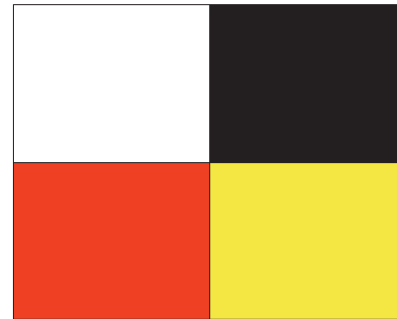


图 15 局部环境信息
Fig. 15 Graph of local environment information

在上述静态全局环境中, 本文设置两个动态障碍物 m_1 和 m_2 来验证算法的有效性。其中, 动态障碍物 m_1 的起点为 (3.5, 15.5), 移动方向为左, 移动速度为 1; 动态障碍物 m_2 的起点为 (2.5, 0.5), 移动方向为上, 移动速度为 1。MA-DQN 算法的参数如表 3 所示。

图 17 中橙色虚线表示移动机器人在静态环境下利用改进蚁群算法得到的全局规划路径, 蓝色路线表示移动机器人实际移动的路线。在 t_1 时刻, 移动机器人视野内存在动态障碍物 m_1 , 则移动机器人使用 MA-DQN 算法进行动态避障, 规避动态障碍物的路径如图 17(b) 中的蓝色路线所示。在移动机器人规避动态障碍物 m_1 的过程中, 已经偏离蚁群算法规划出的路线。

在 t_2 时刻, 移动机器人规避动态障碍物 m_1 过程中到达图 17(b) 所示蓝色路线尾端位置。此时, 移动障碍物 m_2 出现在移动机器人感知范围内, 因此移动机器人仍使用 MA-DQN 算法完成局部动态避障。

在 t_3 时刻, 移动机器人到达图 17(c) 所示位置。此时, 移动机器人的感知范围内已经没有动态障碍物, 则切换路径规划机制, 利用改进的蚁群算法进行全局静态环境下的路径规划。改进的蚁群算法规划出的全局静态环境下的规划路径如图 17(c) 中橙色虚线所示。

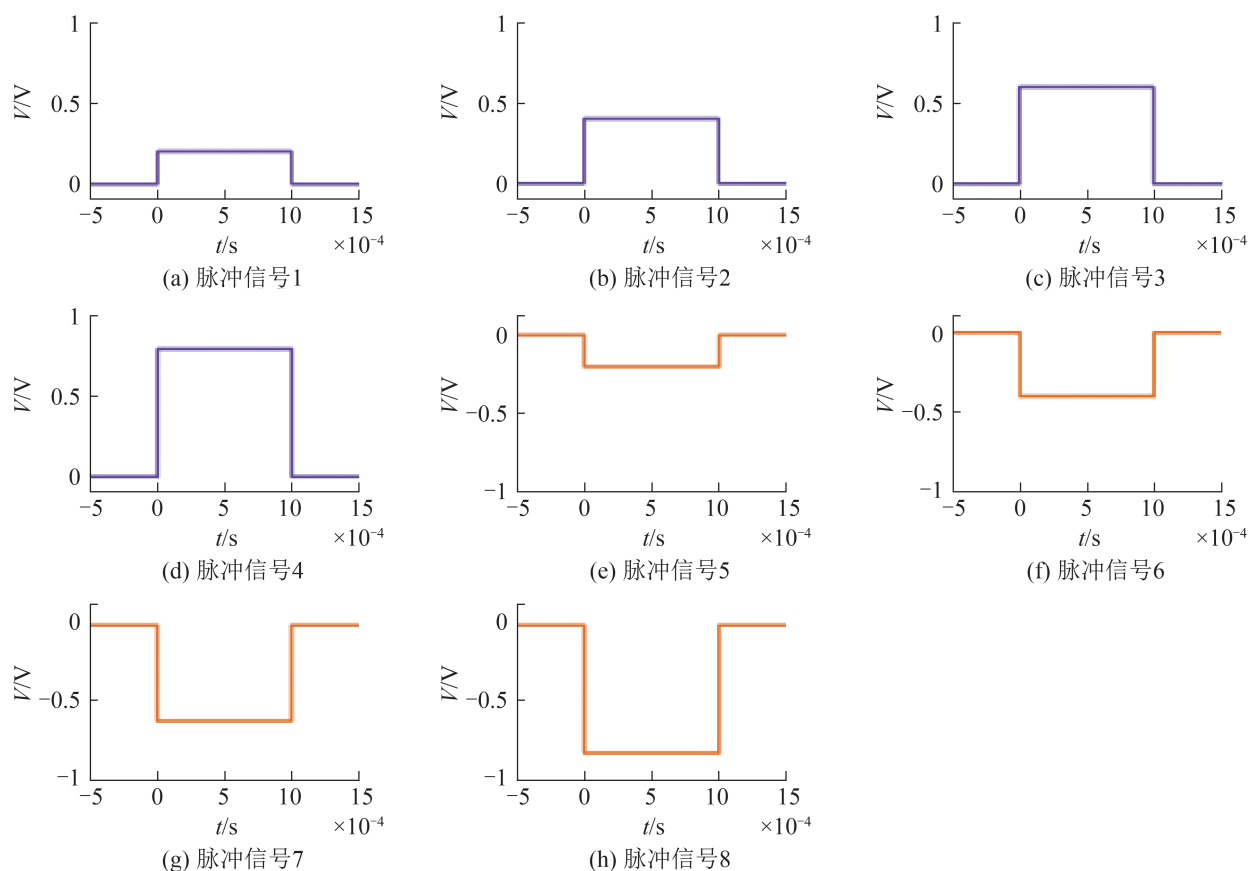


图16 局部环境信息到脉冲产生单元

Fig. 16 Local environmental information to the pulse generation unit

表3 MA-DQN参数
Table 3 MA-DQN parameters

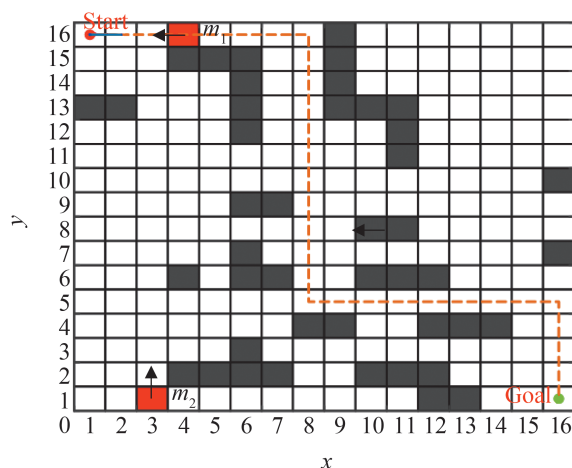
参数	数值
学习率 α	0.01
折扣因子 γ	0.9
探索因子 ϵ	0.1
训练次数 episode	1000

在 t_4 时刻，移动机器人到达图17(d)所示位置。此时移动机器人感知范围内没有动态障碍物，移动机器人一直沿改进的蚁群算法规划的路径行进。但动态障碍物发生如图6所示的侧向冲突，则为了避免动态障碍物冲突。根据式(13)序号优先原则，动态障碍物选择 m_2 移动到冲突点位置， m_1 原地等待。

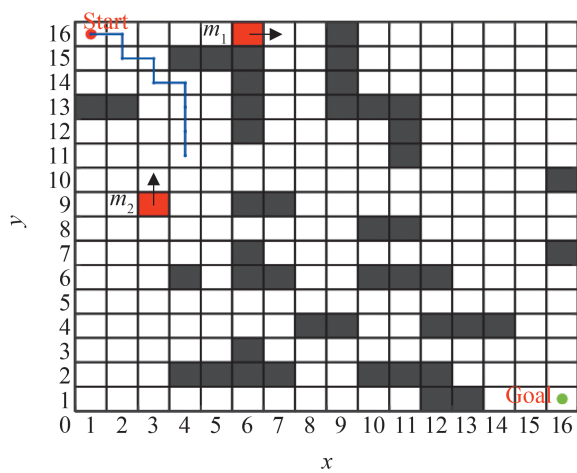
在 t_4 时刻之后，移动机器人未感知到动态障碍物，因此始终沿改进的蚁群算法规划路径行走至目标点位置，如图17(e)所示。

为了观察移动机器人在多个 episode 中的学习

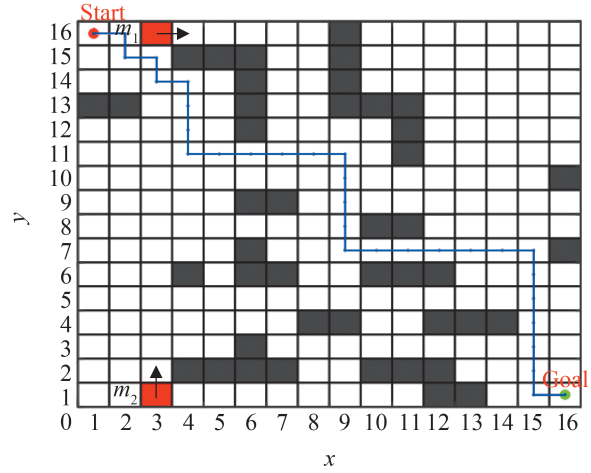
状况，通过 MA-DQN 算法迭代输出学习过程中的移动步数，如图18所示。可以看出，在算法迭代初期，移动机器人移动步数较多，算法不稳定，但 MA-DQN 算法后期移动步数趋于稳定，最终收敛。这也证明了算法可有效进行动态避障。



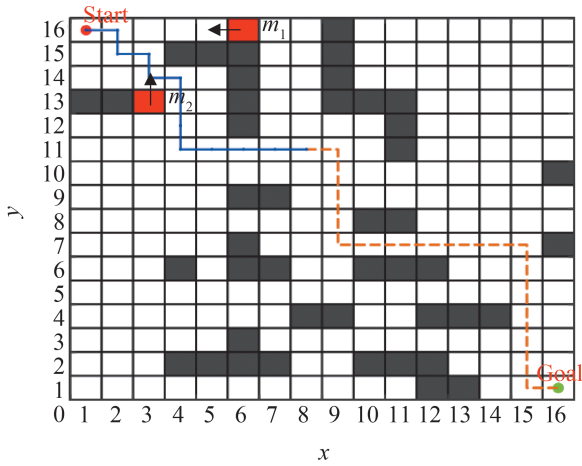
(a) 移动机器人检测到动态障碍物 m_1



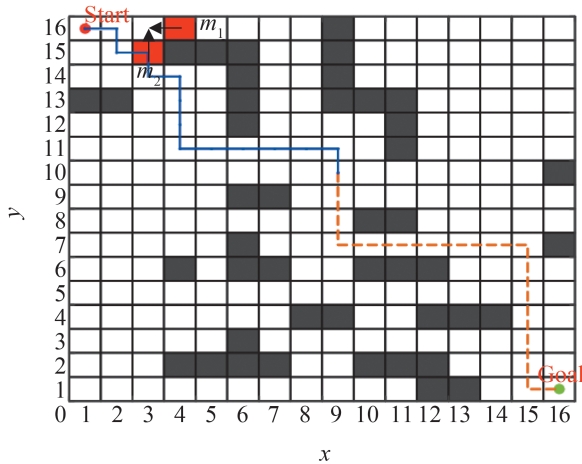
(b) 移动机器人检测到动态障碍物 m_2



(e) 动态环境下路径规划



(c) 移动机器人感知范围无动态障碍物



(d) 动态障碍物碰撞规避

图 17 动态环境路径规划图

Fig. 17 Diagram of dynamic environment path plan

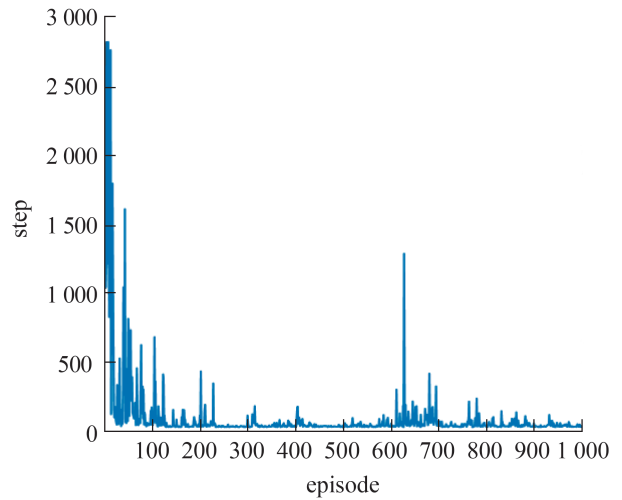


图 18 MA-DQN 算法迭代收敛图

Fig. 18 Iterative graph of MA-DQN algorithm

综上所述, 本文所提出的算法可以在动态环境下规划出一条通往目标点的路径, 证明了本文基于改进蚁群算法和 MA-DQN 算法的双层路径规划算法的有效性。

5 结论

针对动态环境下移动机器人的路径规划问题, 本文提出了基于改进蚁群算法和 MA-DQN 算法的

双层路径规划算法。采用改进信息素更新规则及概率转移函数的蚁群算法完成静态环境下的路径规划。同时设计了将忆阻器阵列和DQN算法结合的方案,提高了神经网络的效率。根据移动机器人感知范围内是否有动态障碍物制定了实时路径规划的切换机制,提高了移动机器人的工作效率。仿真结果表明,本文所提出的算法可有效规避不同速度的动态障碍物,有效、实时地完成动态环境下的路径规划任务。

但本文的MA-DQN算法仅针对方向已知、速度未知的动态障碍物进行有效避障。下一步将考虑更为复杂的动态障碍物进行避障及路径规划。

参考文献:

- [1] 林彬, 韩光辉, 宋晨晨, 等. 基于辐射扫描算法的机器人路径规划与仿真[J]. 系统仿真学报, 2021, 33(1): 84-90.
Lin Bin, Han Guanghui, Song Chenchen, et al. Traversal Path Planning and Simulation of Robot Based on Radiation Scanning[J]. Journal of System Simulation, 2021, 33(1): 84-90.
- [2] 朱大奇, 朱婷婷, 颜明重. 基于改进神经网络的多AUV全覆盖路径规划[J]. 系统仿真学报, 2020, 32(8): 1505-1514.
Zhu Daqi, Zhu Tingting, Yan Mingzhong. Multi-AUV Complete Coverage Path Planning Based on Improved Neural Network[J]. Journal of System Simulation, 2020, 32(8): 1505-1514.
- [3] 白天翔, 王帅, 沈震, 等. 平行机器人与平行无人系统: 框架、结构、过程、平台及其应用[J]. 自动化学报, 2017, 43(2): 161-175.
Bai Tianxiang, Wang Shuai, Shen Zhen, et al. Parallel Robotics and Parallel Unmanned Systems: Framework, Structure, Process, Platform and Applications[J]. Acta Automatica Sinica, 2017, 43(2): 161-175.
- [4] 吴鹏, 桑成军, 陆忠华, 等. 基于改进A*算法的移动机器人路径规划研究[J]. 计算机工程与应用, 2019, 55(21): 226-232, 269.
Wu Peng, Sang Chengjun, Lu Zhonghua, et al. Research on Mobile Robot Path Planning Based on Improved A* Algorithm[J]. Computer Engineering and Applications, 2019, 55(21): 226-232, 269.
- [5] 陈继清, 谭成志, 莫荣现, 等. 基于人工势场的A*算法的移动机器人路径规划[J]. 计算机科学, 2021, 48(11): 327-333.
Chen Jiqing, Tan Chengzhi, Mo Rongxian, et al. Path Planning of Mobile Robot with A* Algorithm Based on Artificial Potential Field[J]. Computer Science, 2021, 48(11): 327-333.
- [6] 阮晓钢, 周静, 张晶晶, 等. 基于子目标搜索的机器人目标导向RRT路径规划算法[J]. 控制与决策, 2020, 35(10): 2543-2548.
Ruan Xiaogang, Zhou Jing, Zhang Jingjing, et al. Robot Goal Guide RRT Path Planning Based on Sub-target Search[J]. Control and Decision, 2020, 35(10): 2543-2548.
- [7] 张毅, 李奎, 黄超. 基于改进蚁群算法的二维码移动机器人路径规划方法[J]. 重庆邮电大学学报(自然科学版), 2021, 33(3): 491-497.
Zhang Yi, Li Kui, Huang Chao. Path Planning Method for Two-dimensional Code Mobile Robot Based on Improved Ant Colony Algorithm[J]. Journal of Chongqing University of Posts and Telecommunications (Natural Science Edition), 2021, 33(3): 491-497.
- [8] Xin Junfeng, Zhong Jiabao, Yang Fengru, et al. An Improved Genetic Algorithm for Path-planning of Unmanned Surface Vehicle[J]. Sensors, 2019, 19(11): 2640.
- [9] 王霄汉, 张霖, 任磊, 等. 基于强化学习的车间调度问题研究简述[J]. 系统仿真学报, 2021, 33(12): 2782-2791.
Wang Xiaohan, Zhang Lin, Ren Lei, et al. Brief Review on Applying Reinforcement Learning to Job Shop Scheduling Problems[J]. Journal of System Simulation, 2021, 33(12): 2782-2791.
- [10] Gao Junli, Ye Weijie, Guo Jing, et al. Deep Reinforcement Learning for Indoor Mobile Robot Path Planning[J]. Sensors, 2020, 20(19): 5493.
- [11] 王毅然, 经小川, 田涛, 等. 基于强化学习的多Agent路径规划方法研究[J]. 计算机应用与软件, 2019, 36(8): 165-171.
Wang Yiran, Jing Xiaochuan, Tian Tao, et al. Multi-agent Path Planning Based on Reinforcement Learning[J]. Computer Applications and Software, 2019, 36(8): 165-171.
- [12] Watkins C J C H, Dayan P. Q-learning[J]. Machine Learning, 1992, 8(3): 279-292.
- [13] Mao Chao, Shen Zuojun. A Reinforcement Learning Framework for the Adaptive Routing Problem in Stochastic Time-dependent Network[J]. Transportation Research Part C: Emerging Technologies, 2018, 93: 179-197.
- [14] Mnih V, Kavukcuoglu K, Silver D, et al. Human-level Control Through Deep Reinforcement Learning[J]. Nature, 2015, 518(7540): 529-533.
- [15] Chua L. Memristor-the Missing Circuit Element[J]. IEEE

- Transactions on Circuit Theory, 1971, 18(5): 507-519.
- [16] Li Can, Belkin D, Li Yunning, et al. Efficient and Self-adaptive in-situ Learning in Multilayer Memristor Neural Networks[J]. Nature Communications, 2018, 9(1): 2385.
- [17] Wang Zhongrui, Li Can, Song Wenhao, et al. Reinforcement Learning With Analogue Memristor Arrays[J]. Nature Electronics, 2019, 2(3): 115-124.
- [18] 张耀中, 胡小方, 周跃, 等. 基于多层忆阻脉冲神经网络的强化学习及应用[J]. 自动化学报, 2019, 45(8): 1536-1547.
- Zhang Yaozhong, Hu Xiaofang, Zhou Yue, et al. A Novel Reinforcement Learning Algorithm Based on Multilayer Memristive Spiking Neural Network With Applications[J]. Acta Automatica Sinica, 2019, 45(8): 1536-1547.
- [19] Strukov D B, Snider G S, Stewart D R, et al. The Missing Memristor Found[J]. Nature, 2008, 453(7191): 80-83.
- [20] Joglekar Y N, Wolf S J. The Elusive Memristor: Properties of Basic Electrical Circuits[J]. European Journal of Physics, 2009, 30(4): 661.
- [21] 段书凯, 胡小方, 王丽丹, 等. 忆阻器阻变随机存取存储器及其在信息存储中的应用[J]. 中国科学(信息科学), 2012, 42(6): 754-769.
- Duan Shukai, Hu Xiaofang, Wang Lidan, et al. Memristor-based RRAM With Applications[J]. Scientia Sinica(Informationis), 2012, 42(6): 754-769.
- [22] 王雷, 石鑫. 基于改进蚁群算法的移动机器人动态路径规划[J]. 南京理工大学学报, 2019, 43(6): 700-707.
- Wang Lei, Shi Xin. Dynamic Path Planning of Mobile Robot Based on Improved Ant Colony Algorithm[J]. Journal of Nanjing University of Science and Technology, 2019, 43(6): 700-707.
- [23] Wang Binyu, Liu Zhe, Li Qingbiao, et al. Mobile Robot Path Planning in Dynamic Environments Through Globally Guided Reinforcement Learning[J]. IEEE Robotics and Automation Letters, 2020, 5(4): 6892-6939.
- [24] 胡飞, 尤志强, 刘鹏, 等. 基于忆阻器交叉阵列的卷积神经网络电路设计[J]. 计算机研究与发展, 2018, 55(5): 1097-1107.
- Hu Fei, You Zhiqiang, Liu Peng, et al. Circuit Design of Convolutional Neural Network Based on Memristor Crossbar Arrays[J]. Journal of Computer Research and Development, 2018, 55(5): 1097-1107.