

doi: 10.7690/bgzdh.2022.04.018

ROS/Gazebo 在 SLAM 算法评估中的应用

丁林祥¹, 陶卫军²

(1. 东南大学成贤学院机械与电气工程学院, 南京 210088; 2. 南京理工大学机械工程学院, 南京 210018)

摘要: 为解决对即时定位与地图构建 (simultaneous location and mapping, SLAM) 算法进行评估时难以获得精确的机器人实际运动轨迹和环境参数的问题, 使用 ROS/Gazebo 仿真软件进行评估。通过 ROS/Gazebo 软件建立配置有多种传感器的机器人模型和参数已知的仿真环境, 让机器人模型在仿真环境中运行并收集传感器信息以形成数据集, 使用该方法对 2 种 2D 激光 SLAM 算法进行评估分析。结果表明, 该方法可有效解决上述问题。

关键词: SLAM; ROS; Gazebo; gmapping; hector_slam

中图分类号: TP391.9 **文献标志码:** A

Application of ROS/Gazebo in the Evaluation of SLAM Algorithm

Ding Linxiang¹, Tao Weijun²

(1. School of Mechanical and Electrical Engineering, Southeast University
Chengxian College, Nanjing 210088, China;

2. School of Mechanical Engineering, Nanjing University of Science and Technology, Nanjing 210018, China)

Abstract: The ROS/Gazebo simulation software is used to evaluate the simultaneous location and mapping (SLAM) algorithm in order to solve the problem that it is difficult to obtain the accurate actual trajectory and environmental parameters of the robot. The ROS/Gazebo software was used to establish a simulation environment with known parameters and a robot model equipped with a variety of sensors. The robot model was run in the simulation environment and the sensor information was collected to form a data set. The method was used to evaluate and analyze two 2D laser SLAM algorithms. The results show that this method can effectively solve the above problems.

Keywords: SLAM; ROS; Gazebo; gmapping; hector_slam

0 引言

机器人系统被广泛运用在救灾抢险、太空探索、军事、医疗和教育等领域。在这些应用中, 机器人都需要具备与环境进行交互的能力。如果环境是已知的, 机器人可以通过传感器确定其在环境中的位置; 如果环境未知或是变化的, 那么机器人就需要运用即时定位与地图构建 (SLAM) 技术来同步定位与构建环境地图^[1-2]。

SLAM 最早由 SMITH 等^[2]1988 年提出。从 20 世纪 80 年代概率理论的广泛应用开始, SLAM 的发展也进入了快车道。概率方法和 SLAM 的结合取得了很大突破, 出现了以卡尔曼滤波、粒子滤波、最大似然估计为基础的研究方法^[3-6]。在 SLAM 算法后续的发展过程中, 出现了许多开源的算法, 有的基于激光雷达, 有的基于相机。随着 SLAM 算法的发展, 各类 SLAM 算法均表现出了优秀性能: 基于激光的 SLAM 算法在静态小场景下更能发挥其稳定性, 有着稳定的测距性能, 受光照强度的影响较小,

操作简单^[7-8]; 基于视觉的 SLAM 算法在纹理细节丰富的环境中可快速地识别环境中的纹理信息, 能应对复杂多变的动态环境^[1, 9-12]。

在工程应用中, 通常要根据实际应用场景对不同的 SLAM 算法进行评估分析。将 SLAM 系统产生的机器人运动轨迹和环境地图数据与机器人实际运动轨迹和真实的环境地图参数进行对比^[13-14]; 但要获得精确的机器人实际运动轨迹和实际环境参数非常困难, 往往需要通过其他昂贵的传感器进行测量。另一方面, 也可使用其他人发布的数据集进行算法评估, 但这些数据集通常是通过具备特定传感器的机器人获得的, 无法用于评估使用其他传感器数据的 SLAM 算法。例如, gmapping 算法需要里程计数据和激光雷达数据, 而 hector_slam 算法只需要激光雷达数据且对激光雷达的性能要求较高, 即适用于 hector_slam 的数据集不适用于 gmapping。笔者提出使用 ROS/Gazebo 软件产生数据集来评估 SLAM 算法的方法, 并对 2 种 2D 激光 SLAM 算法——gmapping、hector_slam 进行评估分析。

收稿日期: 2022-01-26; 修回日期: 2022-02-28

基金项目: 2021 年度东南大学成贤学院青年教师科研发展基金项目 (z0024)

作者简介: 丁林祥 (1992—), 男, 湖北人, 硕士, 从事 ROS、SLAM、机器人等相关研究。E-mail: 1178516007@qq.com。

1 基础知识

1.1 SLAM 问题描述

SLAM 问题可以通过如下概率进行描述^[15]:

$$p(X_{1:t}, m|u_{1:t}, Z_{1:t}) \tag{1}$$

即根据移动机器人从起始时刻 1 到时刻 t 获得的传感器数据 $Z_{1:t}$ 以及一系列控制数据 $u_{1:t}$, 同时对地图信息 m 和机器人位姿 $X_{1:t}$ 进行估计。

1.2 gmapping 算法

gmapping (也称 FastSLAM) 基于 RBPF (rao-blackwellized particle filters)。RBPF 是一种有效解决同时定位于地图构建的算法, 将机器人定位和建图 2 个过程分离^[16-17]。RBPF 的关键是根据观测数据 $Z_{1:t}$ 和里程数据 $u_{0:t}$ 来估计机器人处于潜在位姿 $X_{1:t}$ 下的后验概率 $p(X_{1:t}|u_{0:t}, Z_{1:t})$, 然后利用此概率来计算地图和位姿的联合后验概率:

$$p(x_{1:t}, m|u_{0:t}, z_{1:t}) = p(m|x_{1:t}, z_{1:t})p(x_{1:t}|u_{0:t}, z_{1:t}) \tag{2}$$

该概率可被有效地求出, 因为地图的概率 $p(m|X_{1:t}, Z_{1:t})$ 可以由 $X_{1:t}$ 和 $Z_{1:t}$ 计算得出。

为估计位姿的后验概率 $p(X_{1:t}|u_{0:t}, Z_{1:t})$, RBPF 采用粒子滤波方法, 使每个粒子样本都对应一个独立地图。每个地图都根据对应粒子的环境观测数据 $Z_{1:t}$ 和位姿 $X_{1:t}$ 创建。由于机器人位姿的演变是根据机器人的运动信息, 所以建议分布设为与里程运动模型相等。

RBPF 存在因粒子数较多而造成计算量和内存消耗较大以及频繁执行重采样而造成粒子退化问题, 因此 gmapping 在 RBPF 的基础上改进建议分布和选择性重采样以减少粒子数和防止粒子退化。

1.3 hector_slam 算法

hector_slam 使用高斯牛顿方法, 不需要里程计数据, 只根据激光数据便可实现 SLAM 功能。该算法的一般处理过程: 获取概率栅格地图、Gauss—Newton 法扫描匹配、使用多分辨率地图避免陷入局部极小值^[15-18]。具体流程如下:

1) 获取概率栅格地图。

栅格地图被分为有限个栅格:

$$m = \sum_i m_i \tag{3}$$

估计每一个栅格 m_i 在 t 时刻的状态:

$$p(m_i|z_{1:t}, x_{1:t}) = p(m_i|z_{1:t-1}, x_{1:t-1})p(m_i|z_t, x_t) \tag{4}$$

式中: $z_{1:t}$ 为时刻 1 到时刻 t 的观测数据; $x_{1:t}$ 为时刻 1 到时刻 t 机器人的位姿数据。对于都被标记为占

据的栅格, 可能对应被占据的概率数值会有些许不同。

2) Gauss—Newton 法扫描匹配。

在 t 时刻, 将获取到新的激光数据与 $t-1$ 时刻的栅格地图进行匹配, 如果 t 时刻所有激光点都能变换到被占据的栅格中, 则匹配成功。

为描述新的激光点与栅格地图中被占据栅格的对应程度, 其取值范围为 0~1, 采用双线性插值的方法。

对于变换到栅格中的一个激光点, 希望匹配程度 $M(Pm)$ 趋于 1, 构造了目标函数:

$$\varepsilon^* = \arg \min \sum_{i=1}^n [1 - M(S_i(\varepsilon))]^2 \tag{5}$$

其中 $S_i(\varepsilon)$ 为 t 时刻第 i 个激光点变换到栅格地图的坐标。

3) 使用多分辨率地图避免陷入局部极小值。

由于该算法是基于梯度上升的, 有可能陷入局部极小值。通过使用多个不同分辨率的地图来避免陷入局部最小值。

2 使用 ROS/Gazebo 对 SLAM 算法进行评估

2.1 Gazebo

如图 1 所示, Gazebo 具备强大的动力学仿真能力, 可以访问、调用多个高性能物理引擎。在 Gazebo 中可配置各种传感器, 产生激光雷达、2D/3D 摄像机、Kinect 类型传感器以及接触式传感器等多种传感器数据。通过 Gazebo 可为机器人、传感器和环境控制开发定制插件。同时, Gazebo 提供多种机器人的模型, 包括 PR2、Pioneer2 DX、iRobot Create 和 TurtleBot 等, 也可使用 URDF 来自己创建模型。

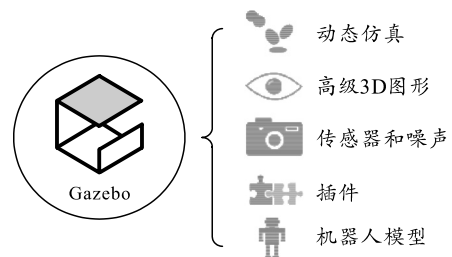


图 1 Gazebo 的特点

2.2 仿真环境和机器人配置

机器人系统仿真器被广泛应用于机器人的运动仿真以及机器人控制算法研究。Gazebo 作为一种强大的机器人仿真工具, 通常搭载在机器人操作系统 (robot operating system, ROS) 中。通过使用 Gazebo, 可建立参数确定的仿真环境以及搭载包括里程计、

激光雷达、IMU 和摄像头等各种传感器的机器人模型。

本文中用于 2 种 2D 激光 SLAM 算法评估分析的仿真环境如图 2 所示, 环境外围是 $10\text{ m}\times 10\text{ m}$ 的围墙, 内部是一个外切圆直径 6 m 的正六边形的墙壁。两墙壁之间的走廊是机器人搜集传感器数据以形成数据集的区间。

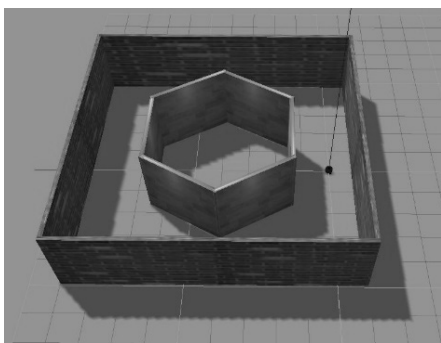


图 2 Gazebo 仿真环境

URDF (unified robot description format) 是 ROS/Gazebo 中机器人模型的描述格式, 包含对机器人刚体外观、物理属性、关节类型等方面的描述。笔者使用 URDF 建立了一个差分驱动的三轮机器人模型如图 3 所示, 底盘上装有 1 个万向轮和 2 个驱动轮, 驱动轮上装配有里程编码器。以同样方法建立激光雷达 rplidar A1 的模型, 并装配到机器人上。

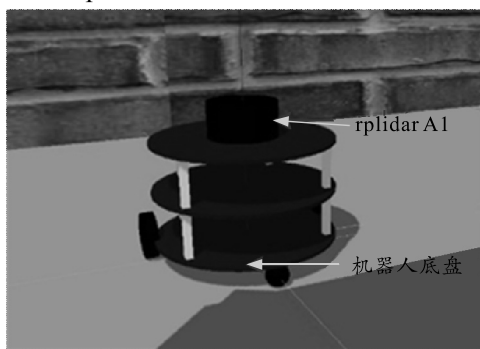


图 3 机器人模型

为能在 Gazebo 中进行仿真, 为上述 URDF 模型添加了 Gazebo 属性, 包括对机器人进行差分控制的 libgazebo_ros_diff_drive.so 插件。在加载差速控制器的过程中, 需要配置一些重要的参数:

<robotNamespace>: 机器人的命名空间。

<wheelSeparation>和<wheelDiameter>: 机器人模型的相关尺寸, 用于计算差速参数。

<wheelAcceleration>: 车轮转动的加速度。

<commandTopic>: 控制器订阅的速度控制指令, 命名为 cmd_vel。

<odometryFrame>: 里程计数据的参考坐标系,

命名为 odom。

载入 Gazebo 中的机器人模型中加入 libgazebo_ros_diff_drive.so 插件后, Gazebo 会订阅/cmd_vel 话题通过发布/cmd_vel 话题就可以控制机器人模型在 Gazebo 中运动。

通过建立一个节点 robot_teleop 接受键盘指令并发布/cmd_vel 话题控制 Gazebo 中的机器人运动以收集传感器数据获得数据集。/cmd_vel 话题中的消息类型为 geometry_msgs/Twist。该消息类型包括线速度 geometry_msgs/Vector3 liner 和角速度 geometry_msgs/Vector3 angular 信息。

笔者使用激光雷达模型对 SLAMTEC 公司的 rplidar A1 进行模拟, ROS 中有相应的功能驱动包, 只需根据实际 rplidar A1 的性能参数对功能包进行配置就可在 Gazebo 中对 rplidar A1 进行模拟。

激光雷达的传感器类型是 ray, 结合 rplidar A1 产品手册配置 ray 中的雷达参数: 360° 检测范围、单圈 360 个采样点、 5.5 Hz 采样频率, 最远 6 m 检测范围等。最后使用<plugin>标签加载激光雷达的插件 libgazebo_ros_laser.so, 所发布的激光雷达话题为 “/scan”。

2.3 获取数据集

由于 gmapping 需要用到里程计数据和激光雷达数据而 hector_slam 只需要激光雷达数据; 因此, 通过建立的节点 robot_teleop 控制机器人, 在图 2 仿真环境中对 2 种 SLAM 算法分别收集了 4 组数据集。机器人在仿真环境中移动的平均线速度保持在 0.2 m/s 左右, 平均角速度在 0.5 rad/s 左右。

数据集采集的系统主要配置如下: 1) Intel(R) Core(TM) i5-7200 CPU @ 2.50 GHz 2.71 GHz; 2) 8 GB DDR4 RAM; 3) 256 GB SSD M.2 PCIe 固态硬盘; 4) 操作系统为 Ubuntu 16.04 LTS OS 并安装有 ROS Kinetic。

机器人在 Gazebo 仿真环境中运行的实际轨迹由 libros_p3d.so 插件获得。通过读取机器人在 world 框架下的位置坐标得到机器人在 Gazebo 仿真环境下的真实坐标。

数据集保存在 ROS 中的 bag 文件中, 包括如下主题的数据:

1) /clock: 用来发布仿真时间;

2) /cmd_vel: robot_teleop 节点用来发布速度的话题来对机器人进行控制;

3) /odom: 机器人发布的里程计数据;

- 4) /scan: 机器人发布的激光雷达数据;
- 5) /tf: 发布系统中的坐标变换;
- 6) /ground_truth: 发布机器人在 Gazebo 仿真环境下的真实坐标。

2.4 评估分析

在 Gazebo 仿真环境中, 通过 robot_teleop 控制机器人模型运动, 收集机器人的真实位置信息 /ground_truth 以及机器人上传感器里程计数据 /odom 和激光雷达数据 /scan。上述数据集保存在 ROS bag 中, /odom 和 /scan 数据将用于 gmapping 算法计算机器人的估计轨迹 1, /scan 数据用于 hector_slam 算法计算机器人的估计轨迹 2。2 个估计轨迹与机器人模型在 Gazebo 仿真环境中的实际轨迹 /ground_truth 进行对比即可对 2 种 SLAM 算法进行评估。流程如图 4 所示。

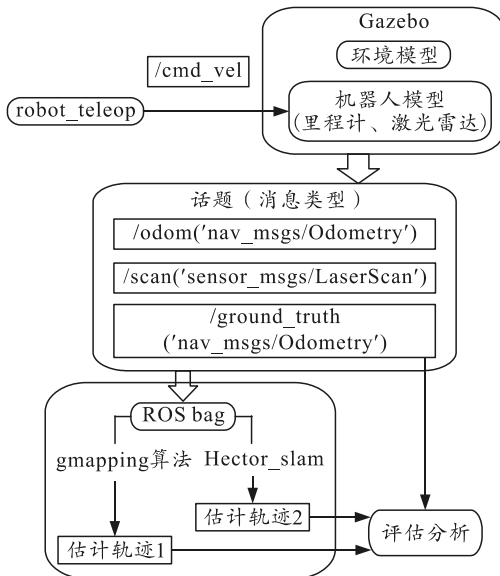


图 4 使用 ROS/gazebo 对 SLAM 算法进行评估分析的流程

对于 2 种 SLAM 算法 (gmapping 和 hector_slam) 分别使用了 4 个数据集, gmapping 编号分别为 gmapping00、gmapping01、gmapping02 和 gmapping03, hector_slam 编号分别为 hector00、hector01、hector02 和 hector03。

图 5(a) 是使用数据集 gmapping00 时, gmapping 算法估计机器人的位置 (gmapping00) 与机器人实际位置 (ground_truth) 的对比, 图 5(b) 是使用数据集 hector00 时, hector_slam 算法估计机器人位置 (hector00) 与实际位置 (ground_truth) 的对比。

通过绝对轨迹误差 (absolute trajectory error, ATE) 来衡量算法定位的误差, k 时刻的 ATE 定义为:

$$e_k = |e_k - g_k| \quad (6)$$

其中 e_k 和 g_k 分别表示机器人位置的估计值和真实值。

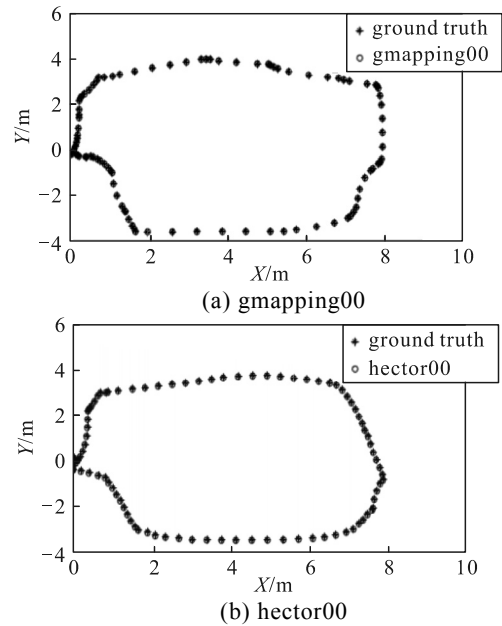
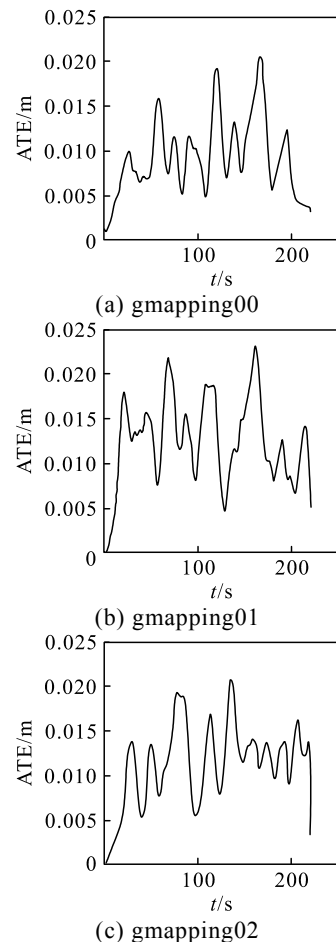
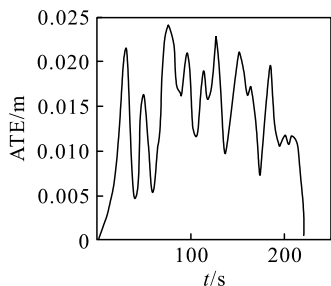


图 5 gmapping 和 hector_slam 中机器人估计轨迹与真实轨迹 (选取 gmapping00 和 hector00 数据集)

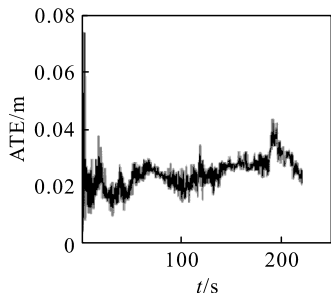
gmapping 算法和 hector_slam 算法分别处理 4 个数据集产生的 ATE 分别如图 6、7 所示。



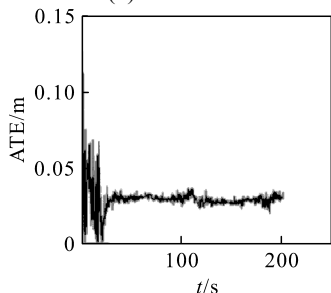


(d) gmapping03

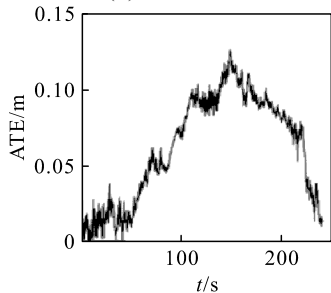
图 6 gmapping 中 ATE 随时间的变化



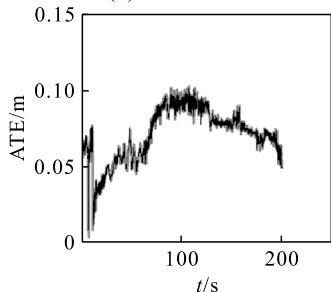
(a) hector00



(b) hector01



(c) hector02



(d) hector03

图 7 hector_slam 中 ATE 随时间的变化

gmapping 算法估计机器人位置的 ATE 平均值为 0.012 m 且最大不超过 0.023 m, 而 hector_slam 算法估计机器人位置的 ATE 平均值为 0.049 m, 可

见在图 2 所示的仿真环境中且传感器配置一样的情况下, gmapping 算法估计机器人位置的精度要高于 hector_slam 算法, 此结果与文献[19]相符。同时, gmapping 算法估计机器人位置的 ATE 都在平均值附近小幅波动, 而 hector_slam 算法估计机器人位置中 hector00 和 hector01 的 ATE 在 0.029 m 上下浮动, 而 hector02 和 hector03 的 ATE 最高可达到 0.132 m, 可见在图 2 所示的仿真环境中且传感器配置一样的情况下, gmapping 算法估计机器人位置的稳定性要高于 hector_slam 算法, 此结果与文献[19]相符。

3 结束语

笔者提出使用 ROS/Gazebo 仿真软件对 SLAM 算法进行评估的方法。评估过程包括: 创建虚拟的环境和机器人模型、配置机器人上的传感器、收集数据集用于 SLAM 算法以及计算 ATE 对算法进行评估。通过该方法, 一方面可以配置机器人模型上的传感器得到需要的数据集; 另一方面可以容易地获得机器人的真实位置和环境的真实参数而无需使用其他传感器进行测量。笔者对 2 种 2D 激光 SLAM 算法进行了评估分析, 验证了该方法的有效性。

参考文献:

- [1] 白云汉. 基于 SLAM 算法和深度神经网络的语义地图构建研究[J]. 计算机应用与软件, 2018(1): 183-190.
- [2] SMITH R C, CHEESEMAN P. On the representation and estimation of spatial uncertainty[J]. The International Journal of Robotics Research, 1986, 5(4): 56-68.
- [3] 周宁亚, 黄友锐. 基于狮群优化的 Fast SLAM 算法[J]. 计算机应用与软件, 2020(7): 206-211.
- [4] 马争光, 赵永国. 激光和视觉融合 SLAM 方法研究综述[J]. 计算机测量与控制, 2019, 27(3): 1-6.
- [5] TORRES-GONZÁLEZ A, MARTINEZ J R, OLLERO A. Range-only SLAM for robot-sensor network cooperation[J]. Autonomous Robots, 2018, 42(3): 649-663.
- [6] 丁林祥, 陶卫军. 未知环境下室内移动机器人定位导航设计与实现[J]. 兵工自动化, 2018, 233(3): 15-20.
- [7] 危双丰, 庞帆, 刘振彬, 等. 基于激光雷达的同时定位与地图构建方法综述[J]. 计算机应用研究, 2020, 37(2): 13-18.
- [8] ZHOU B, HE Y, QIAN K, et al. S4-SLAM: A real-time 3D LIDAR SLAM system for ground/watersurface multi-scene outdoor applications[J]. Autonomous Robots, 2021, 45(3): 77-98.
- [9] ZHANG C, HUANG T, ZHANG R, et al. PLD-SLAM: A new RGB-D SLAM method with point and line features for indoor dynamic scene[J]. International Journal of

- Geo-Information, 2021, 10(3): 163.
- [10] MINGYANG, GENG, SUNING, et al. Unsupervised learning-based depth estimation-aided visual SLAM approach[J]. Circuits Systems and Signal Processing, 2020, 39(2): 543-570.
- [11] GUCLU O, CAN A B. Integrating global and local image features for enhanced loop closure detection in RGB-D SLAM systems[J]. The Visual Computer, 2020, 36(5): 1271-1290.
- [12] 叶刚, 周庆隆. 多机器人系统中基于环境侧信息的移动任务调度算法[J]. 计算机应用与软件, 2019, 36(3): 295-300.
- [13] GAO J, YAN K, BING H. SLAM estimation method for uncertain model noise parameters[J]. Cluster Computing, 2018(7): 1-10.
- [14] 李海丰, 胡遵河, 陈新伟. PLP-SLAM: 基于点、线、面特征融合的视觉 SLAM 方法[J]. 机器人, 2017, 39(2): 214-220, 229.
- *****
- (上接第 86 页)
- [9] 林志强, 樊斌斌, 王磊. 基于模糊聚类分析的相控阵雷达效能评估[J]. 电子信息对抗技术, 2020, 35(3): 64-67.
- [10] 雷志良, 秦开兵, 许明, 等. 基于 AHP-云模型的雷达对抗装备组网作战效能评估[J]. 舰船电子对抗, 2014, 37(6): 77-82.
- [11] 邓堃, 吴静, 柳世考. 基于云理论的雷达组网作战效能评估[J]. 火力与指挥控制, 2007(11): 81-84.
- [12] 戚宗锋, 王华兵, 李建勋. 基于深度学习的雷达侦察系统作战能力评估方法[J]. 指挥控制与仿真, 2020, 42(2): 59-64.
- [13] 李妮, 李玉红, 龚光红, 等. 基于深度学习的体系作战效能智能评估及优化[J]. 系统仿真学报, 2020, 32(8): 1425-1435.
- [14] 刘春生, 王斌, 卢义成. 基于 Netica 的机载雷达侦察效能评估[J]. 现代雷达, 2017, 39(3): 8-12, 16.
- [15] 李健, 刘海滨, 胡笛. 基于生成对抗网络和深度神经网络的武器系统效能评估[J]. 计算机应用与软件, 2020, 37(2): 253-258.
- [16] 宋贵宝, 宋佳明, 周荣基. 基于作战环-可拓云模型的反舰导弹装备体系作战效能评估[J]. 兵工自动化, 2020, 39(8): 4-9.
- [17] 朱宝蓼, 朱荣昌, 熊笑非. 作战飞机效能评估[M]. 2 版. 北京: 航空工业出版社, 2006: 101-106.
- [18] 邓雪, 李家铭, 曾浩健, 等. 层次分析法权重计算方法分析及其应用研究[J]. 数学的实践与认识, 2012, 42(7): 93-100.
- [19] 许树伯. 层次分析原理[M]. 天津: 天津大学出版社, 1988: 6-13.
- [20] 王锦国, 周志芳, 袁永生. 可拓评价方法在环境质量综合评价中的应用[J]. 河海大学学报(自然科学版), 2002(1): 15-18.
- [21] 蔡文. 可拓学概述[J]. 系统工程理论与实践, 1998(1): 77-85.
- [22] 李德毅, 刘常昱. 论正态云模型的普适性[J]. 中国工程科学, 2004(8): 28-34.
- [23] 叶琼, 李绍稳, 张友华, 等. 云模型及应用综述[J]. 计算机工程与设计, 2011, 32(12): 4198-4201.
- [24] 周红波, 李照顺, 谢佑波. 基于云模型的系统综合效能评估方法[J]. 火力与指挥控制, 2017, 42(8): 61-63, 67.
- [25] 李如琦, 苏浩益. 基于可拓云理论的电能质量综合评估模型[J]. 电力系统自动化, 2012, 36(1): 66-70.
- [26] 王晓玲, 戴林瀚, 吕鹏, 等. 基于 DSR-可拓云的渗流安全综合评价研究[J]. 天津大学学报(自然科学与工程技术版), 2019, 52(1): 52-61.
- [27] 高松, 滕克难, 陈健, 等. 基于 MCM-可拓云的装备体系效能评估[J]. 兵器装备工程学报, 2020, 41(2): 20-24.