

大连理工大学本科毕业论文

基于数字孪生的多机协同搬运监控系统设计

Design of Multi Robot Cooperative Handling Monitoring System Based on Digital Twin

学 院（系）： 计算机科学与技术学院

专 业： 计算机科学与技术

学 生 姓 名： 邢文字

学 号： 201745055

指 导 教 师： 丁男

评 阅 教 师：

完 成 日 期： 2021.06.06

大连理工大学

Dalian University of Technology

原创性声明

本人郑重声明：本人所呈交的毕业设计（论文），是在指导老师的指导下独立进行研究所取得的成果。毕业设计（论文）中凡引用他人已经发表或未发表的成果、数据、观点等，均已明确注明出处。除文中已经注明引用的内容外，不包含任何其他个人或集体已经发表或撰写过的科研成果。对本文的研究成果做出重要贡献的个人和集体，均已在文中以明确方式标明。

本声明的法律责任由本人承担。

作者签名：

日 期：

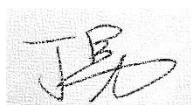
关于使用授权的声明

本人在指导老师指导下所完成的毕业设计（论文）及相关的资料（包括图纸、试验记录、原始数据、实物照片、图片、录音带、设计手稿等），知识产权归属大连理工大学。本人完全了解大连理工大学有关保存、使用毕业设计（论文）的规定，本人授权大连理工大学可以将本毕业设计（论文）的全部或部分内容编入有关数据库进行检索，可以采用任何复制手段保存和汇编本毕业设计（论文）。如果发表相关成果，一定征得指导教师同意，且第一署名单位为大连理工大学。本人离校后使用毕业毕业设计（论文）或与该论文直接相关的学术论文或成果时，第一署名单位仍然为大连理工大学。

论文作者签名：

日 期：

指导老师签名：



日 期：

摘 要

随着数字化水平的加深和复杂度的提高，物联网、智慧城市等领域热度逐渐升高，但也面临了一些挑战，其中就包括物理系统和数字系统之间割裂、交互不充分的问题。数字孪生技术就是解决此问题的一种方法。

本文以多机协同搬运为场景切入，深入研究了多机器人路径规划和协同策略，并针对该场景搭建数字孪生系统，以期设计一种可行的智慧工厂监控方案。主要研究内容如下：

（1）对比和分析了多机协同场景的表示方法，选择双向单通道的栅格地图作为协同搬运任务的工作场景。

（2）针对实际场地设计仿真场景，比较验证了多种多机协同算法。实现了 A* 路径规划算法与动态工作区划分算法相结合的协同策略，测试了该策略的性能。

（3）深入研究数字孪生系统的架构和建模方法，根据数字孪生五维模型进行开发。最终使用 Flask 框架和 SQLite 数据库搭建了数字孪生 Web 平台，能够收集物理场景中的各项数据，实时传输到前端界面进行可视化展示。

（4）研究并实现数字孪生系统对物理实体进行预测和评估的功能。通过 KMeans 聚类算法分析机器人实体的运行状态，根据历史数据评估多机协同搬运的任务效率。

（5）将上述系统部署在实验室的多机协同搬运场景中，使用实体机器人测试系统可行性和有效性。

关键词：多智能体搬运；协同策略；数字孪生系统；状态评估

Design of Multi Robot Cooperative Handling Monitoring System Based on Digital Twin

Abstract

With the deepening of digital level and the improvement of its complexity, the popularity of Internet of things, smart city and other fields is gradually increasing, but it also faces some challenges, including the separation and weak interaction between physical system and digital system. Digital twin technology is a way to solve this problem.

Based on the scene of multi robot cooperative transportation, this paper deeply studies the path planning and cooperative strategy of multi robot, and builds a digital twin system for the scene, in order to design a feasible intelligent factory monitoring scheme. The main research contents are as follows

(1) This paper compares and analyzes the representation methods of multi machine collaborative scene, and selects the two-way single channel grid map as the working scene of collaborative transportation task.

(2) For the actual site design simulation scene, a variety of multi machine cooperation algorithms are compared and verified. A collaborative strategy combining A* path planning algorithm and dynamic workspace partition algorithm is implemented, and the performance of the strategy is tested.

(3) In depth study of digital twin system architecture and modeling methods, according to the digital twin five-dimensional model. Finally, a digital twin web platform is built by using the Flash framework and SQLite database, which can collect various data in the physical scene and transmit them to the front-end interface for visualization.

(4) Research and implement the function of digital twin system to predict and evaluate physical entities. KMeans clustering algorithm is used to analyze the running state of the robot, and the task efficiency of multi machine cooperative transportation is evaluated according to the historical data.

(5) The above-mentioned system is deployed in the multi machine cooperative transportation scene in the laboratory, and the feasibility and effectiveness of the system are tested by using the real robots.

Key Words: MAS Transportation; Collaborative Strategy; Digital Twin System; Condition Assessment

目 录

摘 要.....	I
Abstract	II
1 文献综述.....	1
1.1 课题来源及研究意义.....	1
1.2 国内外研究现状.....	1
1.2.1 数字孪生模型和建模方法.....	1
1.2.2 物理实体的数据采集和传输方法.....	2
1.2.3 数字孪生的信息安全.....	2
1.2.4 数字孪生的应用领域和面临的问题.....	3
1.2.5 多机协同任务的路径规划算法.....	3
1.2.6 多机协同任务的协同策略.....	5
2 系统整体设计.....	7
2.1 系统设计要求.....	7
2.2 系统框架.....	7
2.3 系统实现方法概述.....	8
3 多智能体路径规划和动态区域划分策略.....	10
3.1 任务场景与机器人实体.....	10
3.1.1 地图表示方法.....	10
3.1.2 多机器人协同物理场景的搭建与描述.....	11
3.1.3 机器人平台简介.....	12
3.2 多机器人控制.....	13
3.2.1 集中式多机器人控制法.....	13
3.2.2 单机路径规划算法.....	14
3.3 单项单通道场景协同策略及其问题.....	16
3.3.1 多机协同搬运场景比较.....	16
3.3.2 单项单通道场景的冲突解决.....	18
3.4 双向单通道场景下的动态工作区划分策略.....	18
3.4.1 策略设计思想.....	18
3.4.2 策略具体实现.....	20
3.4.3 性能测试.....	22
3.5 本章小结.....	25

4	数字孪生子系统设计与实现.....	26
4.1	使用 SQLite 框架的数据库	26
4.2	使用 Flask 框架的 Web 服务器.....	27
4.3	前端可视化界面	28
4.4	Ajax 前后端数据交互方法	30
4.5	物理实体状态评估功能	30
4.5.1	使用 KMeans 聚类算法评估运行状态	31
4.5.2	KMeans 聚类算法性能分析	31
4.5.3	整体协同搬运任务效率评估	33
4.6	本章小结	34
结	论	35
参	考 文 献.....	36
修	改记录	38
致	谢	39

1 文献综述

1.1 课题来源及研究意义

随着数字化产业的发展，物联网、智慧城市等概念逐渐升温，并且在世界范围内掀起了一股浪潮。比如“中国制造 2025”、“工业 4.0”等概念的提出，都体现了这种发展趋势。这些技术能够为我们带来全新的生活体验，包括多维度的感知、高效可靠的通信、智能化的交互、精细化的分析管理等。但是，随着数字化水平的加深和复杂度的提高，物联网也面临了一些挑战，其中就包括物理系统和数字系统之间割裂、交互不充分的问题。数字孪生技术就是解决这一问题的一种方法。

数字孪生(Digital Twin)是通过数字化的方式，建立针对物理实体的多维度、多学科、多物理量的动态虚拟模型，来反映物理实体在真实物理场景中的各种属性、行为^[1]。它是实现物理系统向信息空间数字化模型映射的关键技术，能够通过传感器，对物理实体进行建模，借助高性能计算、高速通信等技术，辅以数据分析和仿真模拟，实时地呈现物理系统的实际情况^[2]。

2020 年 4 月，国家发展改革委中央网信办印发的《关于推进“上云用数赋智”行动培育新经济发展实施方案》的通知中数次提及数字孪生的发展需求，将其与大数据、人工智能、云计算、5G、物联网和区块链技术并列，作为数字化转型共性技术和关键技术。作为网络物理空间 CPS(Cyber-Physical Space)建设的重要支撑技术之一，数字孪生已应用在航空航天、智能制造、智慧城市等领域。

本项目在数字化智能工厂的应用需求下，构建多机器人协同搬运任务场景，设计并部署多机协同策略，通过实时、多维地获取物理实体的各项参数来搭建相应的数字孪生模型，将其用于对物理实体的监控和行为预测。

1.2 国内外研究现状

1.2.1 数字孪生模型和建模方法

数字孪生模型与数字孪生的建模方法有很多，包括应用范围较广的通用模型和针对具体场景的、有自身特点模型。刘青等^[3]提出了一种面向多感知模型的建模方法，首先要建立数字孪生初始模型，通过视觉、听觉、动力感知、控制信号等多传感器。其次，要将初始模型逐步匹配到应用场景中，通过物理实体返回的控制数据进行迭代，将实际场景中可能影响到物理实体的各种信息逐步到数字孪生模型中。数字孪生模型与通常的仿真模型不同，它是通过数据驱动的，能够实时地完善仿真模型，从而实现对物理实体

的更加准确的仿真预测^[2]。例如 Mukherjee 等^[4]提到的基于仿真数据库的微内核架构，能够为模型提供足够可靠高效的实时数据支持。陶飞等^[1]提出了数字孪生五维模型，将一个完整的数字孪生模型划分为物理实体、虚拟实体、孪生数据、连接和服务五部分。基于“工业 4.0”智能化场景，还有文献提出了数字孪生三层模型，包括物理实体、数据层、信息处理与优化层^[5]。刘大同等^[6]提出了四层模型，根据数据采集到应用，分为数据保障层、建模计算层、数字孪生功能层和沉浸式体验层。其中，建模计算层充分利用了机器学习、人工智能的方法，挖掘数据间的相互关系和特征，通过嵌入式计算和云服务器来协同完成建模任务。

这些模型虽然描述形式不同，但核心思想较为一致：首先，数字孪生系统包含物理实体和虚拟实体，二者通过可靠的数据传输与储存系统，能够实时地保持在多种维度上的一致性；根据这些数据，辅以人工智能、边缘计算、高性能计算等技术，完成数字孪生模型的服务功能，例如风险评估、行为预测等。

1.2.2 物理实体的数据采集和传输方法

数据采集需要满足多维度和实时性，使用多种类型的传感器进行采集，同时保证数据传输的效率和安全性。刘青等^[3]提供了以下传感器选择的思路：视觉相关技术，使用扫描设备等建立环境模型；使用听觉设备，如超声波检测仪器，建立结构模型和功能模型；最后使用表面触觉体验系统、嗅觉系统等协助模型建立。再具体实现方法上，有文献基于 CPS 架构，在信息传输过程中根据环境数据进行了误差估计，提高了数据的可靠性^[7]；Elattar 等^[8]使用了一种全新的多路径路由选择算法，提高了数据传输的速度；有文献^[9]提出了 Ada-MAC 协议，相较于其他方法，能够有更低的功耗和延时。由于数字孪生需要在真实环境中实时地采集数据，传感器获得的数据具有多源、异构、多尺度、高噪声等特点^[2]，因此需要采用一些数据清洗的方法，比如机器学习、规则约束等。

1.2.3 数字孪生的信息安全

数字孪生系统的信息安全问题非常关键。数字孪生系统遭到攻击可能会引起物理设备本身的损坏，从而加大了安全问题带来的损失。此外，数字孪生模型本身包含了物理实体的参数和历史数据，可能会导致系统的关键技术泄露。

工业自动化和控制系统(IACS)是一个非常宽泛的术语，涵盖了与不同行业中的控制、监视和生产相关的所有内容。近年来，安全性已在研究界乃至整个行业内引起了很多关注。一些重大安全事件，例如 2010 年的 STUXNET 蠕虫，2012 年的 Shamon Saudi Aramco 鱼叉式网络钓鱼攻击和 2014 年的德国钢铁厂攻击，都突显了攻击 IACS 所带来的风险^[10]。这些攻击具有许多不同的类型和起源，它们也突显了增强安全机制和对策的重要性。

除了针对 IACS 的攻击外，现有的物联网和智能设备体系本身也存在这一些风险：传统的设备、分布式设备、云计算平台等本身存在安全漏洞；很多体系结构涉及到物联网、大数据等技术的融合，体系结构越来越复杂，安全问题也会加剧。针对数字孪生安全性问题，有文献^[10]提出了一种数字孪生 IACS 对手模型，并且基于传统的数字孪生模型设计了新的状态复制模型，能够更好的满足安全性要求。Feng M 等^[11]将 CPS 系统中的攻击问题转化为博弈论问题，利用深度强化学习的方法进行求解，从而降低了攻击的影响。Claudio Mandolla 等^[12]结合了区块链技术，对数字孪生系统的各种数据进行保护。

1.2.4 数字孪生的应用领域和面临的问题

数字孪生最早是 2003 年 Grieves M.W.教授在全生命周期管理课程中提出的，称为“镜像空间模型”。由于 GE、西门子等公司的推广，数字孪生技术近年来在工业制造领域同样发展迅速^[6]。例如 IBM 的“智慧地球”概念，雄安新区的“数字孪生城市”，都体现了数字孪生在智慧城市建设方面的应用^[2]。现阶段，除了航空航天领域，数字孪生还被应用于电力、船舶、城市管理、农业、建筑、制造、石油天然气、健康医疗、环境保护等行业。特别是在智能制造领域，数字孪生被认为是一种实现制造信息世界与物理世界交互融合的有效手段。

数字孪生同样面临许多问题。刘青等^[3]指出：现有的数字孪生模型往往针对某一特定领域，各个领域之间难以找到通用的模型；已有的模型中，很少有能够满足高保真、多尺度、多物理场特点的实际项目，这些数字孪生模型不能很好的满足需求；还需要解决数字孪生场景中的各种不确定性问题。有文献^[1]指出，我们需要将数字孪生系统的结果服务化，如何应用移动端 APP、VR、AR 等形式给用户带来沉浸式体验和相应的智能服务，是数字孪生普适应用的一大难题。本项目力求建立高保真、多尺度的针对具体场景的数字孪生模型，通过 Web 监控系统的形式给用户预测、分析等智能化服务。

1.2.5 多机协同任务的路径规划算法

机器人路径规划是智能体控制技术中的热点问题，已经在许多领域得到应用。例如在军事领域，无人机路线规划、导弹回避、智能机器人控制、水下无人航行器自助航行，以及美国“小精灵”项目等；在日常方面，有基于地理信息系统的路径规划，城市智能交通动态路径规划、物流或者外卖路径规划，以及自动导引装置问题等^[13]。路径规划的算法主要包含：模拟退火法、A*算法、Dijkstra 算法、遗传算法、粒子束算法，人工势场法等。路径规划算法也可通过硬件方式进行优化，例如梁晓辉等^[14]利用微电子元件和光学期间，有效提高了路径规划算法在实时系统中的运行速度。路径规划算法可以分为非进化型路径规划和进化型路径规划两种类别。

非进化型路径规划是指通过机械式的方法解决路径规划问题。这种算法的设计思想非常简洁,并且处理问题的效率很高,但是有可能产生非最优路径,并且不能实现自我优化和学习,没有记忆能力。这种算法在处理复杂维度的路径规划问题时往往效果不好。

20 世纪 90 年代初期, M.H.Overmars 提出图搜索概率法(PRM)^[15], PRM 算法是一种基于图搜索的方法,包括学习阶段和查询阶段两个阶段,可以将连续空间转化成离散的空间进行路径搜索,以提高在高维场景问题中的搜索效率。这种方法复杂度较低,建模难度低,但是因为采样点的随机性问题,无法覆盖全部的路径,容易得出非最优解的结果,并且遇到狭窄通道或者障碍集合复杂的情况时效率会明显降低。针对 PRM 的缺点,夏清松等^[16]提出了一种节点增强法来替代原来的结点,利用圆弧来替代折线,能够缩小路径长度并减少节点个数。

另一种典型的非进化型算法是 N.Metropolis 等提出的模拟退火算法(SA),它通过模拟热力学中固体物质的退火过程与一般组合优化问题之间的相似性并结合概率突跳特性,使得局部最优解能概率性地跳出并最终趋于全局最优的模式。模拟退火算法在计算中不一定可以获得最优解,但一定可以得到全局次优解^[17]。随后, Ulungu 提出了多目标模拟退火算法(MOSA),针对传统 SA 算法的缺点, MOSA 能够处理多目标的问题。文献^[18]介绍了多目标退火算法(r MOSA),显著降低了 MOSA 算法计算出结果所需要的时间,提高了算法的效率。张丹露等^[19]将遗传算法和模拟退火算法相结合得到了混合 GA-SA 算法,有着更高的收敛速度和可行性。除了上述算法,非进化型路径规划算法还有人工势场法、A*算法、Dijkstra 算法、Floyd 算法、Voronoi 算法、矢量法、栅格法等。

进化型路径规划算法就是指具有一定的自我学习、自我更新和记忆能力的路径规划算法,能够处理复杂的路径规划问题,但由于计算量大,效率低下,不适合进行实时控制^[13]。

二十世纪九十年代,意大利学者以蚂蚁在自然界中协同工作寻找食物为数学模型提出了蚁群算法。蚁群算法利用正反馈机制,信息共享,并且易于和其他算法结合使用,搜索能力极强。由于蚁群算法不能进行实时地路径搜索,不能在有限的计算次数内得到最优路径。Liana Machado 等^[20]介绍了一种改进的 Ant-Q 蚁群算法,此算法能以较大的概率选择信息素强度最大的路径,从而使得算法获得最优解的概率提高。徐精明等提出了多态蚁群算法,通过对蚁群内不同种类的单位进行工作划分,结合局部与全局搜索,提高了传统蚁群算法的效率^[21]。肖绍等^[22]指出粒子群参数优化的改进蚁群算法,通过全局异步策略和精英策略,减少了粒子群算法在调用蚁群算法时产生结果所需的迭代次数。

另一种典型的算法就是遗传算法(GA),它是 1962 年 John Holland 提出的。遗传算法通过模拟进化论中的自然选择和遗传变异的过程形成的一种搜索算法。遗传算法具有

自适应和学习性的特点,适合处理多个群体中的单位,常用于解决全局搜索问题。由于遗传算法的随机性,它的可靠性表现较差。周松儒等^[23]提出了一种混合遗传算法,将遗传算法、粒子群算法、人工免疫算法结合,这种算法解决了遗传算法只找到局部最优解的问题。2016年,田欣等提出了一种模拟退火算法和遗传算法相结合的新型遗传算法,核心思想是自适应调整,显著提高了算法的寻找最优解的速度^[24]。

此外,非进化型路径规划算法还包括禁忌搜索算法、神经网络算法、粒子群算法等。

1.2.6 多机协同任务的协同策略

多机器人协同工作并不是单机器人的简单叠加,在任务分配、动态避碰等方面都要复杂的多,由于多机器人协同路径规划是 NP-Hard 问题,对该问题的研究尚未成熟^[25]。多机器人协同问题中,最基本的就是碰撞避免问题,主要有两种解决思路。一是基于时间窗的预测方法,在路径规划阶段添加时间维度,使其在路径规划阶段就生成不会与其他移动机器人发生动态碰撞的路径;二是两阶段的方法,即首先由全局控制中心进行路径规划,完成规划后将路径下发给移动机器人,在移动机器人运动的过程中,出现将要碰撞的情况时再进行协调。

曾庆文等^[26]提出了次维扩展的新框架,正常情况下只进行一个机器人的路径规划,仅在机器人可能发生冲突时进行协调策略,从而降低了搜索路径的计算成本。有文献^[27]针对 Automated Guided Vehicle(AGV)间的复杂作用关系,提出自动调整 AGV 优先级的方法,利用交通规则从时间和空间上解决冲突,并对对向冲突导致的死锁问题进行预测与解决。张丹露等^[28]使用动态加权地图,用预约表解决最基本的碰撞问题,用每条边的权重反映路径的拥堵程度,从而在路径规划阶段就避开拥堵区域,减少了发生冲突的可能性。Guruji A K 等^[29]针对 A*算法计算量较大的问题,提出在实际运行的碰撞阶段之前确定启发式函数的值,具有较高的处理速度。上述方法都从不同角度出發,解决了多个机器人之间的碰撞、拥堵与死锁等问题,总体来说,目前并不存在一种绝对占优的多机人协同策略。谢云等^[30]使用了一种多机器人顺序协同策略来解决多机器人物流搬运问题,首先将每个机器人能够执行任务的区域进行限定,上层的控制中心负责判断任务能否由单独的机器人完成,如果是多机协同任务,控制中心会生成顺序执行策略,即一个机器人将任务交给下一个机器人继续完成。有文献^[31]采用统一协调式规划方法中基于优先级的协调策略。控制中心根据机器人的特点和任务属性设置优先级,先对高优先级的机器人进行路径规划,再进行低优先级机器人的路径规划。低优先级机器人会将高优先级的机器人转化为环境地图中的障碍物。这种方法不需要复杂的协同策略,只需要研究单机器人的路径规划问题。

多机器人协同算法的性能和效果与机器人所在的场景直接相关,因此,协同策略需

要将机器人实体、任务类型与任务场景相结合进行统筹设计，目前并没有一种较为普适的方法或模型。

2 系统整体设计

2.1 系统设计要求

根据选题内容，本项目主要设计任务和要求如下：

（1）搭建符合设计场景的数字孪生平台。该平台部署在 Web 上，用于监控、分析和预测多机器人智能工厂场景下机器人的各项参数和行为，提供用户可视化界面。

（2）构建满足多维性和实时性特点的数字孪生模型。能够实时地多维度反应机器人的各项参数，包括姿态参数、传感器读数、位置坐标、任务进度等。

（3）使用数字孪生模型评估和预测物理实体。平台通过收集到的数据，预测智能工厂中机器人即将进行的动作，并根据实际运行的结果给出评估和异常报警。

（4）设计智能工厂场景下的多机协同搬运策略。包括机器人的路径规划、任务调度、冲突避免等方法的探索 and 实现。

（5）将上述数字孪生平台和多机器人实体部署在实验室的多机协同搬运场景中测试运行，评价方法性能和系统稳定性。

2.2 系统框架

根据国内外研究现状，以及主要研究内容分析，本题目可以确定如下的研究方案：首先在实验室现有的实验场地中，实现“机器人——中央控制节点——数字孪生服务器”的信息与数据交互；第二步，明确数字孪生 Web 服务器所需要的传感器数据和状态信息，构建服务器存储物理实体传输来的各项数据；第三步，通过前端设计，多维度实时地呈现物理实体的各项数据，界面合理，具有用户友好性；第四步，研究并部署多机器人协同策略。第五步，完成数字孪生系统预测和分析物理实体行为的功能。

根据研究方案，系统可以自底向上划分成三部分：物理实体、数字孪生服务器、虚拟实体。物理实体是指多机协同搬运任务场景下的机器人和多智能体（MAS）控制器，它会向数字孪生服务器实时发送传感器数据、任务进度、位置坐标、状态参数等数据。数字孪生服务器负责接收和存储数据，利用数据构建数字孪生模型，即虚拟实体，并且为用户提供相应的智能化服务。虚拟实体就是一个与物理实体状态同步、同时运行的数字化模型，通过监控视频、数字栅格地图、传感器数据、参数表等方法呈现物理实体的运行状态。虚拟实体通过 Web 页面呈现。

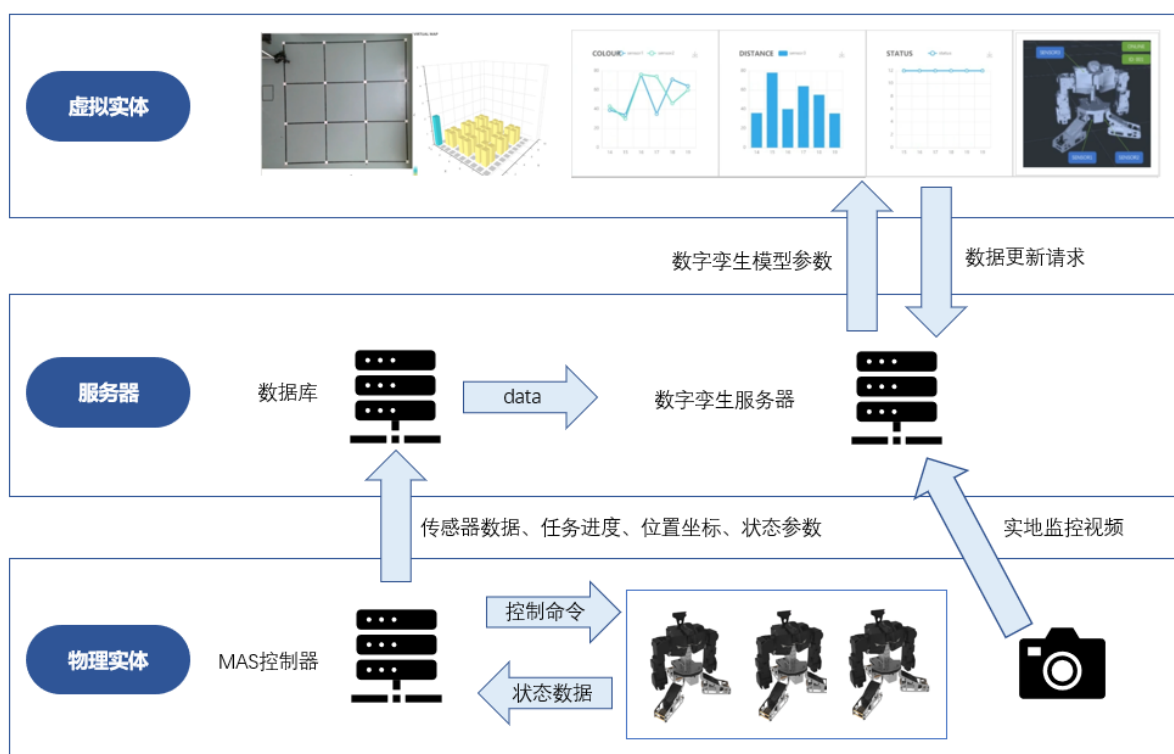


图 2.1 系统整体架构

2.3 系统实现方法概述

具体实现系统所用的技术和方法如下：机器人平台使用博创尚和公司开发的智元素格斗机器人，它具有可移动底盘和多种传感器接口，由树莓派和 Arduino 作为控制系统，能够完成项目所需的各种功能。系统运行时，机器人读取传感器数据，通过 socket 发送给 MAS 控制器，MAS 控制器通过 socket 将生成的控制指令发送给机器人；MAS 控制器运行双向单通道场景下的动态工作区分配算法和 A*单机路径规划算法，生成多机器协同动作指令；MAS 控制器将传感器数据、机器人任务进度、机器人坐标写入 SQLite 框架搭建的数据库；数字孪生服务器使用 Flask 框架搭建，能够提供 Web 服务；使用 jQuery 工具包实现 Ajax（Asynchronous JavaScript And XML）方法，数字孪生服务器接收前端页面发送的 HTML 请求，从数据库取出数据，以 json 格式发送到前端；前端采用 HTML、CSS 和 JavaScript 进行编写，使用 Echart 图表库的折线图、柱状图、3D 图表等接口实现数据可视化功能；将广角摄像头安装在树莓派上，使用 OpenCV 和 Mjpg-Streamer，将视频实时传输到前端页面；数字孪生 Web 界面能够提供物理实体监控视频、机器人颜色传感器和距离传感器数据、多机协同搬运任务进度、机器人 3D 模型、虚拟栅格地图、系统运行健康程度评估等功能。通过 KMeans 聚类算法，将异常数据与正常

数据分离，后续可以根据聚类中心进行状态预测。项目完整的技术栈如图 2.2。

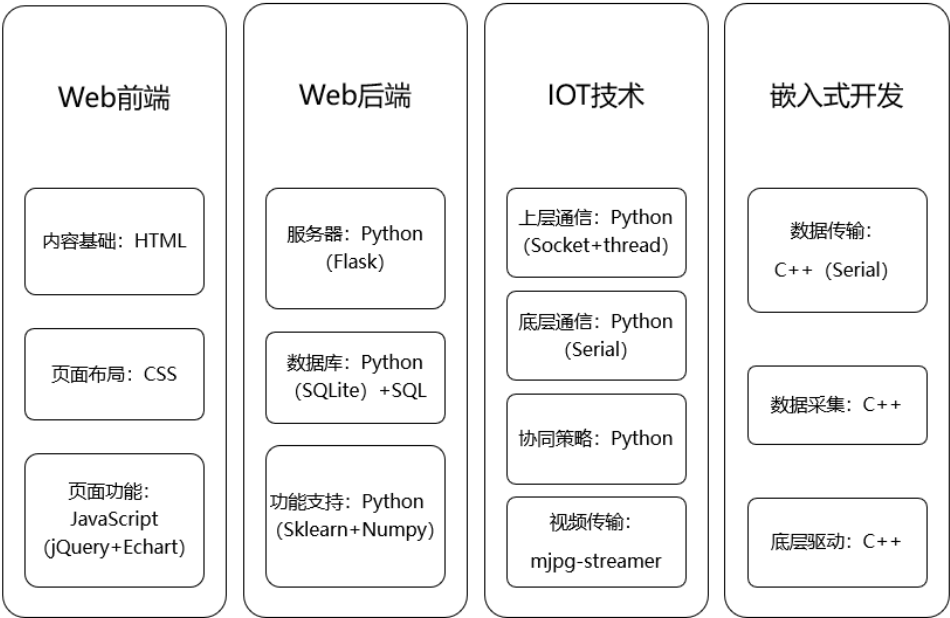


图 2.2 项目技术栈

3 多智能体路径规划和动态区域划分策略

3.1 任务场景与机器人实体

3.1.1 地图表示方法

研究地图表示方法目的是将物理空间向抽象空间转化，并且表示出物理空间中各种客观实体的信息。物理空间就是机器人实际运行的空间，抽象空间就是程序、算法运行的数字环境。地图表示方法主要有栅格法、拓扑法、可视图法三种。栅格法将物理场景转化成不重叠并且相互连接的栅格。可以根据栅格不同分为精确栅格和近似栅格两种类型。精确栅格法是对物理空间的精确映射，一般会保留原空间的形状特征，而近似栅格法使用的栅格形状一般都为矩形^[32]。近似栅格法使用的栅格大小会决定地图表达程度、占用存储空间的容量、路径规划算法的复杂度，因此选择适合场景的栅格大小非常关键。可视图法是用封闭多边形表示场景中的地形，用点表示机器人，通过机器人、目标点和障碍物多边形顶点形成的可视直线来描述的可视图。可视图法表达简洁，占用空间小，并且简单搜索可视图就能得到一条无碰撞的最短路径，效率很高。但是可视图法只适用于较为固定的开放场景，一旦障碍物位置发生改变，就需要重新确定障碍物多边形顶点以及绘制可视图。除此之外，随着场景复杂程度的提高，可视图法的计算复杂度会爆炸增长。拓扑法是指使用点、边来表示地图连通状态的方法，将物理场景转化成合适的结点是拓扑法的关键步骤。拓扑法的优点是占用存储空间小，路径计算比较简单，进行全局规划时表现较好。但拓扑法在使用时依赖机器人定位的精确度，这就需要一些传感器和定位技术的辅助，实际部署是难度较高。

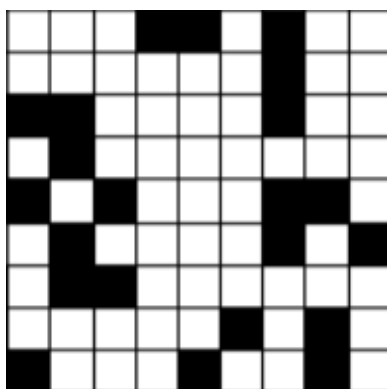


图 3.1 栅格法示意图

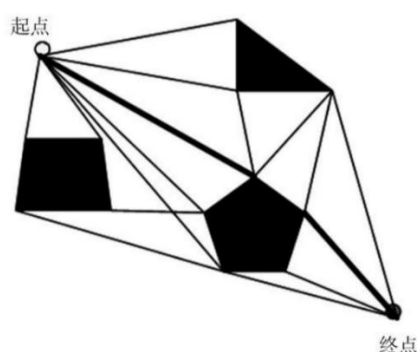


图 3.2 可视图法示意图

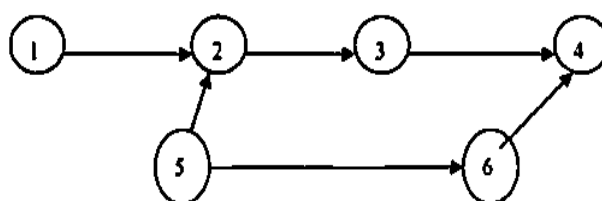


图 3.3 拓扑法示意图

3.1.2 多机器人协同物理场景的搭建与描述

本项目的物理场景基于实验室现有的多机器人实验场地进行搭建。如图 3.4 所示，这是一个单通道（循迹）多机器人物流搬运场景，多个机器人可以在该场景上协同完成随机生成的搬运任务。我使用栅格法建立该场景的抽象模型，白色表示机器人可以通过的栅格，黑色表示障碍物。机器人的任务是：从起点出发，前往随机生成的货物坐标取走货物，到达随机生成的任务终点放下货物。当多个机器人同时执行任务时，会涉及到路径规划算法的设计以及协同策略的部署。场地包括栅格地图、机器人平台若干、用于监控的树莓派和广角摄像机一台。

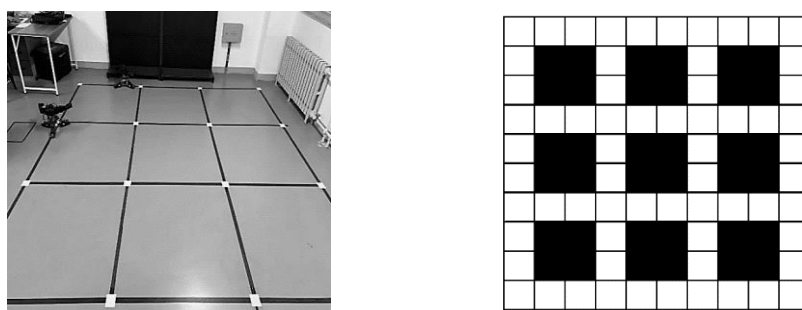


图 3.4 协同任务场景

由于该模型是单通道栅格地图，在进行多机器人协同任务策略设计时，我们首先要考虑路径规划的问题，即如何生成代价最低的多机器人的行进路线；其次是如何处理多机器人协同任务中的路径冲突和相遇碰撞问题。这些问题我将在以下的章节中讨论和解决。

3.1.3 机器人平台简介

本项目的物理实体使用博创尚和公司开发的"智元素"智能机器人开放平台。该平台有以下特点：（1）机器人整体采用模块化设计，下半部分为3轮全向底盘，使用坚固的不锈钢材质，上半部分安装两个4自由度机械臂，手臂可单独使用；（2）有良好的运动性能，全向底盘最大速度 2m/s，具有航向和角速度双闭环的姿态稳定结构；（3）智能化，安装有用于视觉处理的树莓 3B 以及行为处理的 Arduino，底层运动控制由 STM32 完成；（4）可执行动作丰富，有机械臂、带机械离合总线舵机和全向移动底盘，能够适应复杂的任务需求；（5）有丰富的传感器和扩展接口，包括运动系统传感器，霍尔与加速度传感，RJ2 扩展接口等。



图 3.5 机器人实体

通信方面，机器人由树莓派和 Arduino 协同控制。为了适应数字孪生系统数据多维性的要求，在此基础上，还添加了以下传感器：速度传感器、角度传感器、超声波传感器、颜色传感器。其中，速度传感器、角度传感器用于补充机器人运行状态的各项属性；超声波传感器用于测距，防止机器人与障碍物发生碰撞；颜色传感器用于识别场景中的黑色路线与白色交叉路口结点。树莓派负责接收 MAS 控制器发送的指令，向上位机发送机器人的状态信息和各种传感器的读数；Arduino 负责接收树莓派下达的控制信号，读取传感器的数据并上传树莓派，控制底盘的步进电机以移动机器人。树莓派和 Arduino 之间使用串口通信。机器人的各种指令都由 MAS 控制器进行生成和下发，具体内容将在后续段落中介绍。

3.2 多机器人控制

3.2.1 集中式多机器人控制法

常用的多机器人控制方法可以分为集中式控制、分布式控制和混合式控制。其中结构最简单的就是集中式控制法。集中式控制法是指，整个系统有一个统一的上位机完成所有的信息传递和任务调度工作，各个机器人之间没有数据交互。但是控制中心的性能会直接影响到系统整体的性能，需要承担很大的计算量。此外，同时向多个机器人发送和接收数据，需要解决通讯的延时、信道拥塞、线程同步等问题，随着机器人数量的增多，问题会越来越难以解决。分布式控制法是指每一个机器人都是高度独立的个体，可以自行与场景中的其他机器人通信，每个机器人独立完成自己的路径规划。这种方法实现最为复杂。混合控制法又称弱集中式控制法，它是集中式控制与分布式控制的结合。系统中存在一个控制中心，但每个机器人也可以互相通信，控制中心下发任务后，每个机器人独立执行。当遇到冲突问题是，首先有机器人自行尝试解决，如果无法处理，则会上报控制中心进行调度。这种方法结合了两种控制方法的优点，但是实现难度较高。

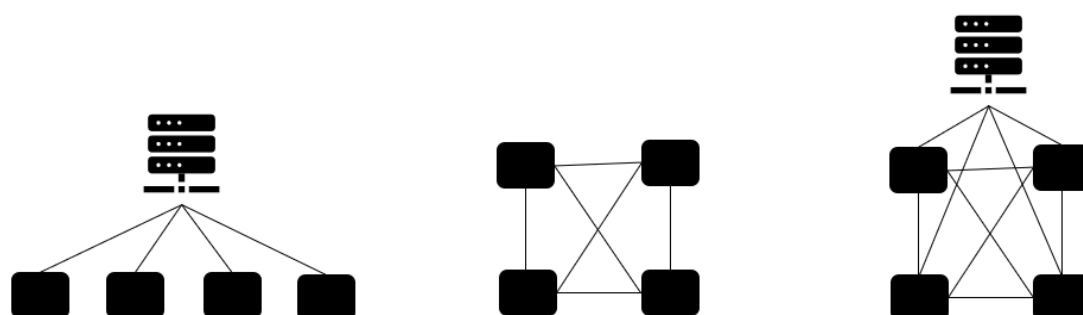


图 3.6 三种多机器人控制方法

我使用集中式控制法的思想搭建了一个 MAS 控制器，它负责的任务是：生成每个机器人的路径、执行协同策略、接收机器人发送的各项数据、将数据处理后上传到数字孪生服务器。通过多线程技术，MAS 控制器一个线程通过 socket 向机器人发送移动指令，另一个线程接收机器人通过 socket 发送的传感器数据，并将这些数据写入数据库。每有一个机器人接入 MAS 控制器，就需要为它开放两个新的线程和两个 socket 端口完成上述工作。我使用 python3 的 threading 库实现多线程并行。Socket 通信就是使用 TCP/IP 协议的进程间的通讯方式，通过绑定 IP 地址和端口号，服务器和客户端就可以互相发送数据包。

3.2.2 单机路径规划算法

MAS 控制器可以生成一个随机的搬运任务，给出机器人的起点和终点，进行最短路径规划，然后将路径转换成机器人的移动指令序列。

栅格地图中最简单的路径规划算法是广度优先搜索，从起点开始，首先遍历起点周围邻接的点，然后再遍历已经遍历过的点的邻接点，直到到达终点。这种算法并不一定能找到最短的路径，并且它认为每个点的代价都是一样的。另一种是 Dijkstra 算法，在该算法中，需要计算每一个点的总移动代价，这个代价包括本身的移动距离，和移动所需要的时间（考虑到某些地形需要转向、减速），并且根据每个点的代价生成一个优先级队列。执行算法时，每次都会选择代价最小的点作为下一个遍历的点，直到找到终点。与之类似的是最佳优先搜索算法，区别就是最佳优先搜索的代价是根据到达终点的代价进行优先级排序的。但是这两种算法在有障碍物的情况下，并不一定能找到最短路径，并不适合我使用的单通道物流搬运场景。A*算法综合了上述算法优先级的思想，它的优先级定义如下：

$$f(n) = g(n) + h(n) \quad (3.1)$$

其中 $f(n)$ 是结点 n 的综合优先级， $g(n)$ 是它到起点的代价， $h(n)$ 是它到终点的预估代价，也就是 A*算法的启发函数。A*算法与上述考虑优先级的算法类似，都是根据优先级队列遍历下一个点。通常选择到达终点的代价作为启发函数。由于单项单通道场景只能允许上下左右四个方向的移动，我选择用曼哈顿距离来衡量距离代价。

$$h(n) = |n.x - goal.x| + |n.y - goal.y| \quad (3.2)$$

其中， n 表示当前所在点， $goal$ 表示终点。除此之外，还需要考虑机器人转向的代价。传统 A*算法仅考虑的路径长度对于节点优先级的影响，不考虑路径的平滑度，即转向的次数。在实际场景中，转弯动作会耗费大量时间，如果不考虑转弯动作，则规划出的路径可能出现虽然路径长度最短，但实际行进时间不是最短的情况。所以，在进行路径规划时，不但应该考虑路径长度，还应考虑转弯次数对实际执行时间的影响。因此，需要对 A*算法的启发函数进行优化。

综上所述，A*算法中，结点 n 的优先级由它到起点的代价、到终点的代价和机器人转向的次数共同决定。这样就能尽可能的获得最短的路径。A*算法的具体流程如图 3.7 所示。

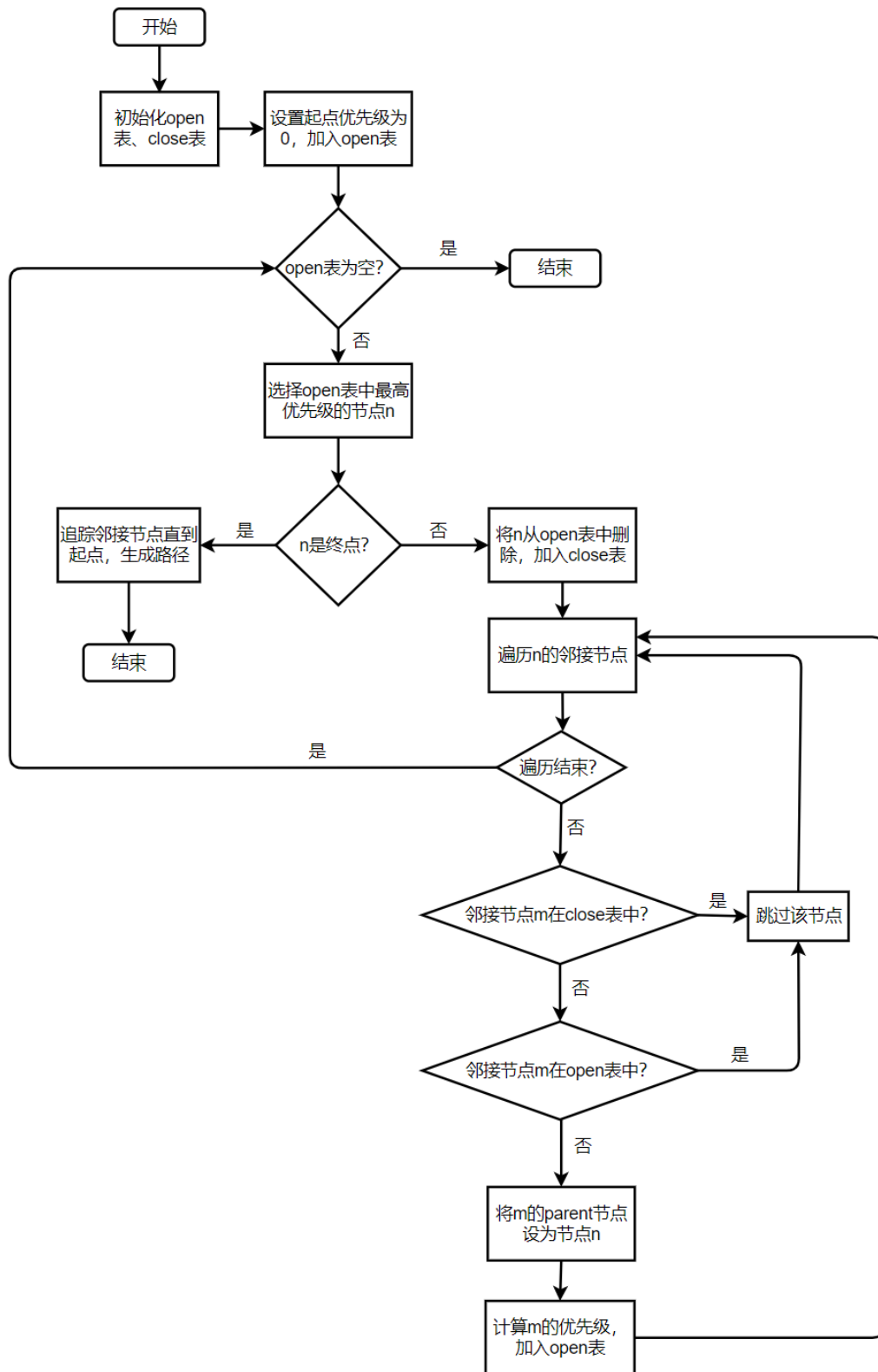


图 3.7 A*算法流程图

具体运行时，MAS 服务器会随机生成起点和终点，以及机器人启动时的朝向。现在给出机器人起点在红色格点、终点在绿色格点的任务示例，具体路径生成的过程如下：

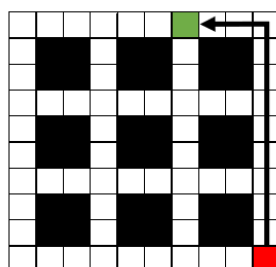


图 3.8 单机路径规划示例

(1) 坐标式路径[(4, 4), (4, 3), (4, 2), (4, 1), (3, 1)]。第一步是根据栅格地图给出的最短路径的坐标点，由起点开始，到终点结束。

(2) 方向式路径['L', 'L', 'L', 'B']。第二步是根据坐标转换成的移动方向序列，包含“前后左右”四个移动方向。

(3) 动作式路径["lssslsb"]。第三步，根据移动方向，转化成机器人的动作指令序列。这些动作指令包括“左转右转”、“前进”以及“向后转”。这个字符串类型的指令序列将会被发送到机器人的树莓派中，通过串口送到 Arduino。Arduino 会根据串口收到的指令，执行机器人的简单动作程序。

3.3 单项单通道场景协同策略及其问题

3.3.1 多机协同搬运场景比较

在栅格地图场景下，主要由以下四种通行规则的定义：单项单通道、双向单通道、单向多通道、双向多通道。单向多通道、双向多通道两种场景可能发生的冲突类型较少，仅使用简单的避障算法就能解决冲突问题，并且单通道更符合实验室场地的循迹场景。因此，重点研究单向单通道和双向单通道场景下多机协同的冲突问题，多通道场景不做详细讨论。

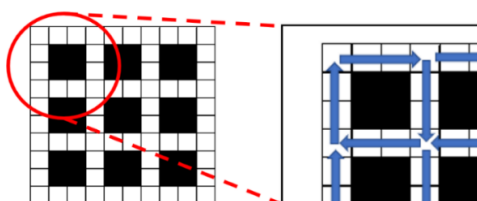


图 3.9 单向单通道通行方向定义

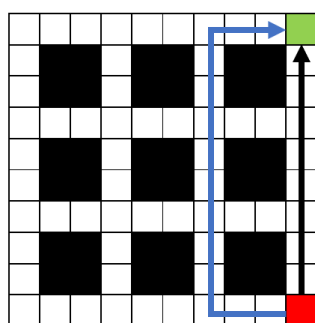


图 3.10 两种场景的路径差异

为了比较单向单通道和双向单通道场景的搬运效率，我随机生成了 10000 个搬运任务，在两种场景下运行相同的路径规划算法，在考虑到路径长度和机器人的转向代价后，可以得到图 3.11 的结果。图 3.11 纵坐标表示搬运的代价（用指令条数表示），横坐标表示的是任务的序号，结果按照搬运时间长短进行排序，曲线和 x 轴围成的面积即为完成任务的总时间。针对相同的搬运任务，双向单通道场景的效率约为单向单通道场景效率的 1.56 倍。由此可见，在该场景中，单项单通道的任务效率是非常低的。

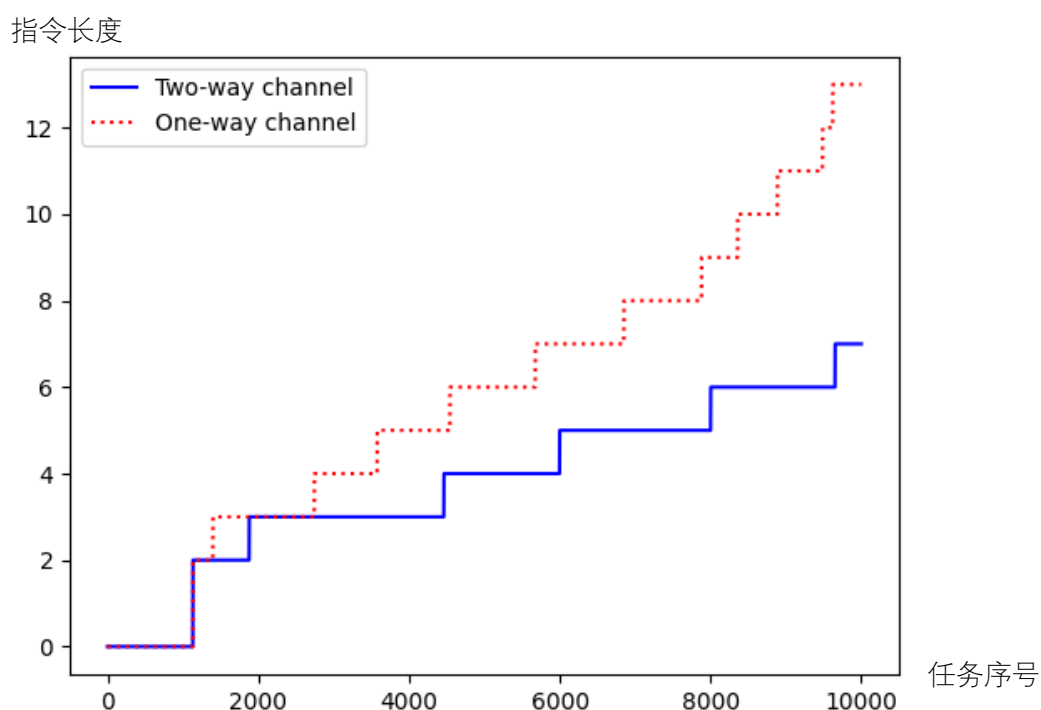


图 3.11 两种单通道场景的任务效率

3.3.2 单项单通道场景的冲突解决

在单向单通道的栅格地图中，多机协同任务常见的冲突问题有两种，分别是同向的追逐问题，交叉路口的冲突问题，如图 3.12 所示。针对单项单通道的冲突有以下解决方法：首先，相同路径上前后两个机器人的追逐问题。由于多机器人并不是以相同的速度做简单的匀速直线运动，机器人可能发生追逐碰撞。这种情况依靠机器人上安装的超声波传感器解决，一旦传感器检测到前方有可能发生碰撞，就会上控制中心发送警报，控制中心会暂停该机器人的动作；第二，在交叉路口的相遇碰撞问题。使用路口结点状态表，结合操作系统资源管理中常用的信号量技术，每个机器人在到达路口结点前需要申请路口资源，使用后再释放资源。如果路口被占用，那么申请资源的机器人将处于挂起状态，暂停行动。通过以上两个简单的策略，可以解决大部分单项单通道场景下可能出现的多机器人协同问题。此处研究单项单通道的冲突解决，主要用于后续与双向单通道场景的协同策略进行对比。

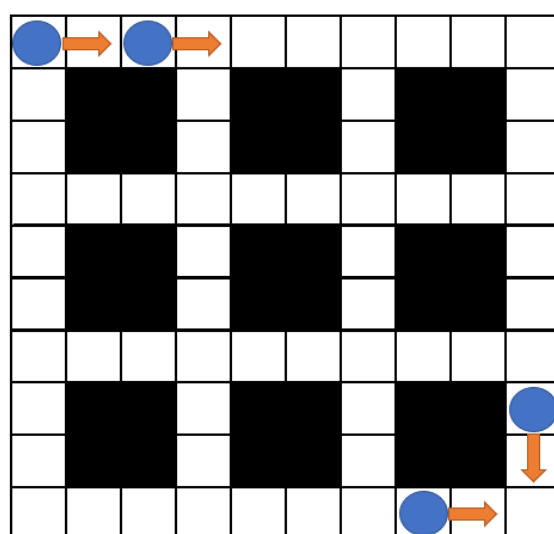


图 3.12 单向单通场景的两种冲突

3.4 双向单通道场景下的动态工作区划分策略

3.4.1 策略设计思想

由于单项单通道场景任务效率低，本项目重点解决双向单通道场景下的多机协同问题，这种场景冲突类型更多，解决方法更为复杂。双向单通道场景同样存在同向的追逐

问题和交叉路口的冲突问题，除此之外，还会出现对向冲突的情况。对向冲突问题是指两个机器人在只允许单一机器人通过的通道中发生的相遇问题。简单的对向冲突解决方法，其一是信号量技术，通过预约路口结点来防止对向冲突的发生；其二是低优先级避让策略，根据二者任务的优先级，优先级低的单位将会才去避让措施。这两种方法都有缺陷。使用信号量技术时，容易发生“死锁”，考虑以下运行场景：机器人 a 在结点 (1, 1) 预约结点 (1, 2) 的使用权，系统检测到 (1, 2) 结点正在被机器人 b 占用，系统将机器人 a 挂起；机器人 b 在结点 (1, 2) 预约结点 (1, 1) 的使用权，系统检测到 (1, 1) 结点正被机器人 a 占用，系统将机器人 b 挂起。这样系统就进入了典型的死锁状态。解决死锁是操作系统领域重要的研究方向，在此不做讨论。另一种低优先级避让策略，可能会出现连锁避让的情况，即低优先级的对象在避让时又触发了和其它机器人的对向冲突，需要再次进行优先级避让计算，从而发生一系列连锁反应，极大降低了整个多机协同系统的工作效率。

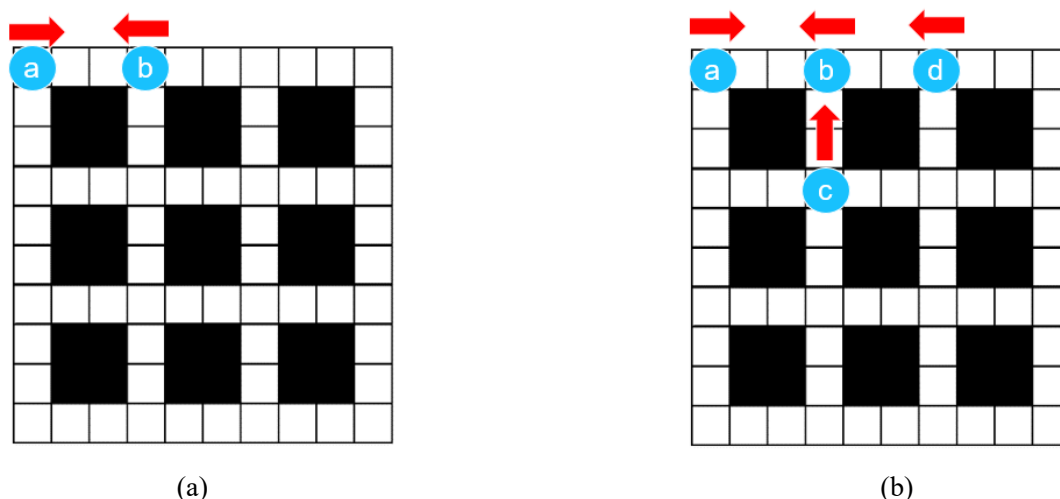


图 3.13 双向单通场景的两种问题

针对上述双向单通道场景下的对向冲突问题，我设计了一种动态工作区划分的方法，能够在多机协同任务正常进行的情况下完全避免对向冲突的发生。张兴国等^[18]提出了一种多机器人顺序协同策略，我使用的动态工作区算法参考了这种思想，即动态划分每个搬运机器人负责的搬运任务区域，由 MAS 控制器判断任务由单一机器人执行还是多个机器人协同执行。进行多机任务时，前一个机器人将货物运送到中转站，后一个机器人去中转站取走货物的方式，继续搬运任务。

3.4.2 策略具体实现

定义工作区：工作区是指 MAS 控制器给机器人指定的进行搬运任务的区域。机器人在接收到工作区划分指令后，只能在工作区范围内的栅格移动，不会与其它工作区的机器人发生任何交互，可以到达中转站（图 3.13 中黄色圆形区域）存放货物。当机器人完成该工作区内的全部搬运任务后，机器人将会等待 MAS 控制器下发新的工作区和搬运任务。根据机器人的数量和实际场景的大小，工作区有以下合适的划分方法：

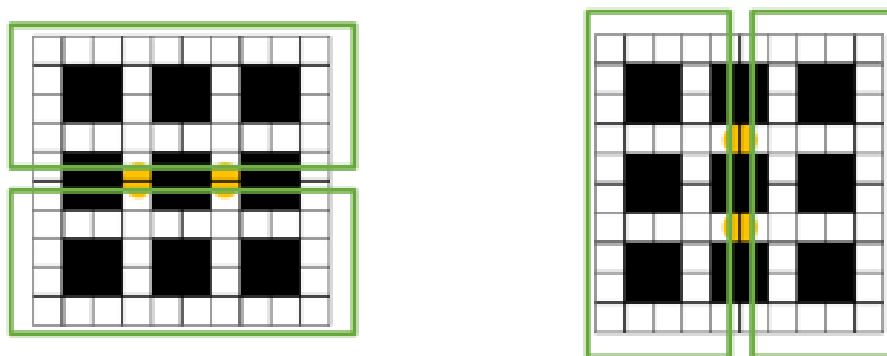


图 3.14 $n=2$ 工作区划分

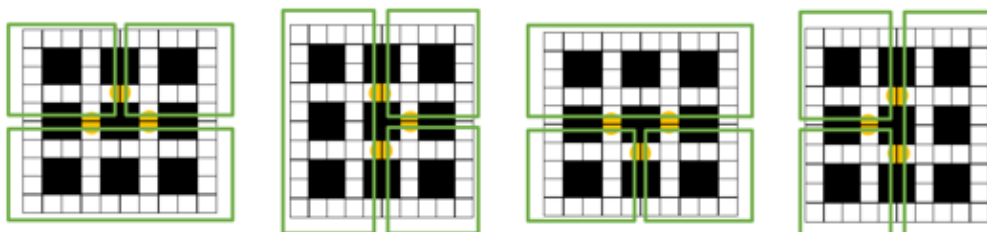


图 3.15 $n=3$ 工作区划分

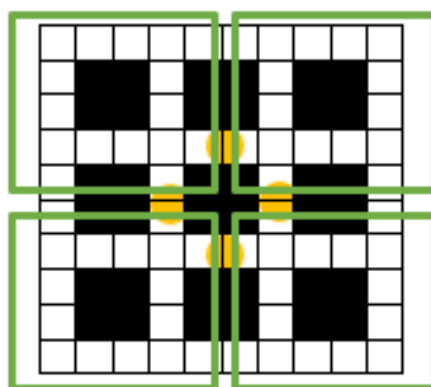


图 3.16 $n=4$ 工作区划分

工作区根据任务的优先级进行划分。以 $n=2$ 时的划分方法为例，我们给每个划分方法定义一个评分标准：由于中转区的存在会增加任务的时间成本，策略应该尽可能减少多机任务，当一个任务能够在某个工作区内独立完成（指起点和终点都在同一个工作区内），这个任务的优先级分数计入这种划分方法的分数。通过比较每一种工作区划分方式的评分，即可选择当前任务需要的方法。划分工作区后，搬运任务将被分为单机任务和多机任务。单机任务由单一机器人在一个工作区内完成。多机任务由多个机器人在多个工作区内完成，这些任务将被重新划分为“起点——中转区、中转区——中转区...中转区——终点”。任务划分后，系统会更新任务的执行顺序，在考虑任务优先级的前提下，使“起点——中转区”的任务先执行，保证后续机器人能够在中转区拿到所需的货物。具体路径如图 3.16 所示。

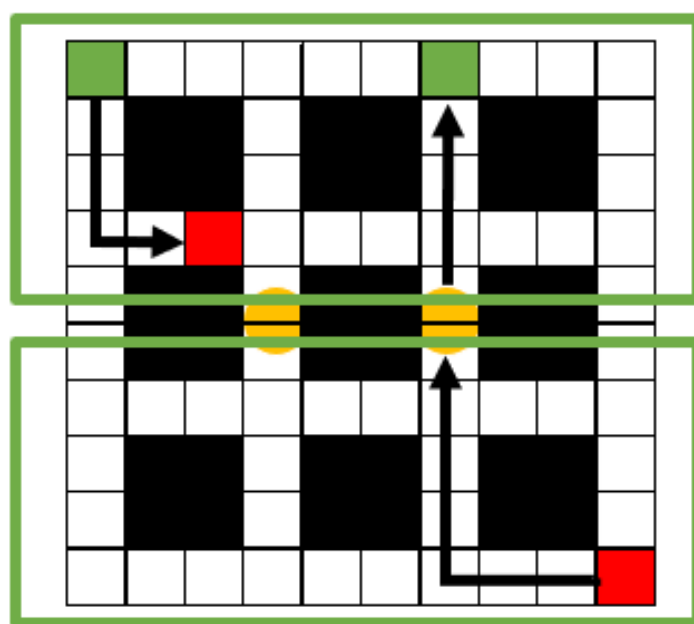


图 3.17 单机任务与多机任务

此处简化了中转区的具体实现方法，中转区内可以任意存放货物，且机器人每次都能恰好从中转区取出自己需要的货物。通过调整多机任务的任务执行顺序，确保“起点——中转区”的任务先执行，这样就能尽可能消除机器人在中转区等待货物到达所需的时间。动态工作区划分策略的具体算法流程图如图 3.17 所示。

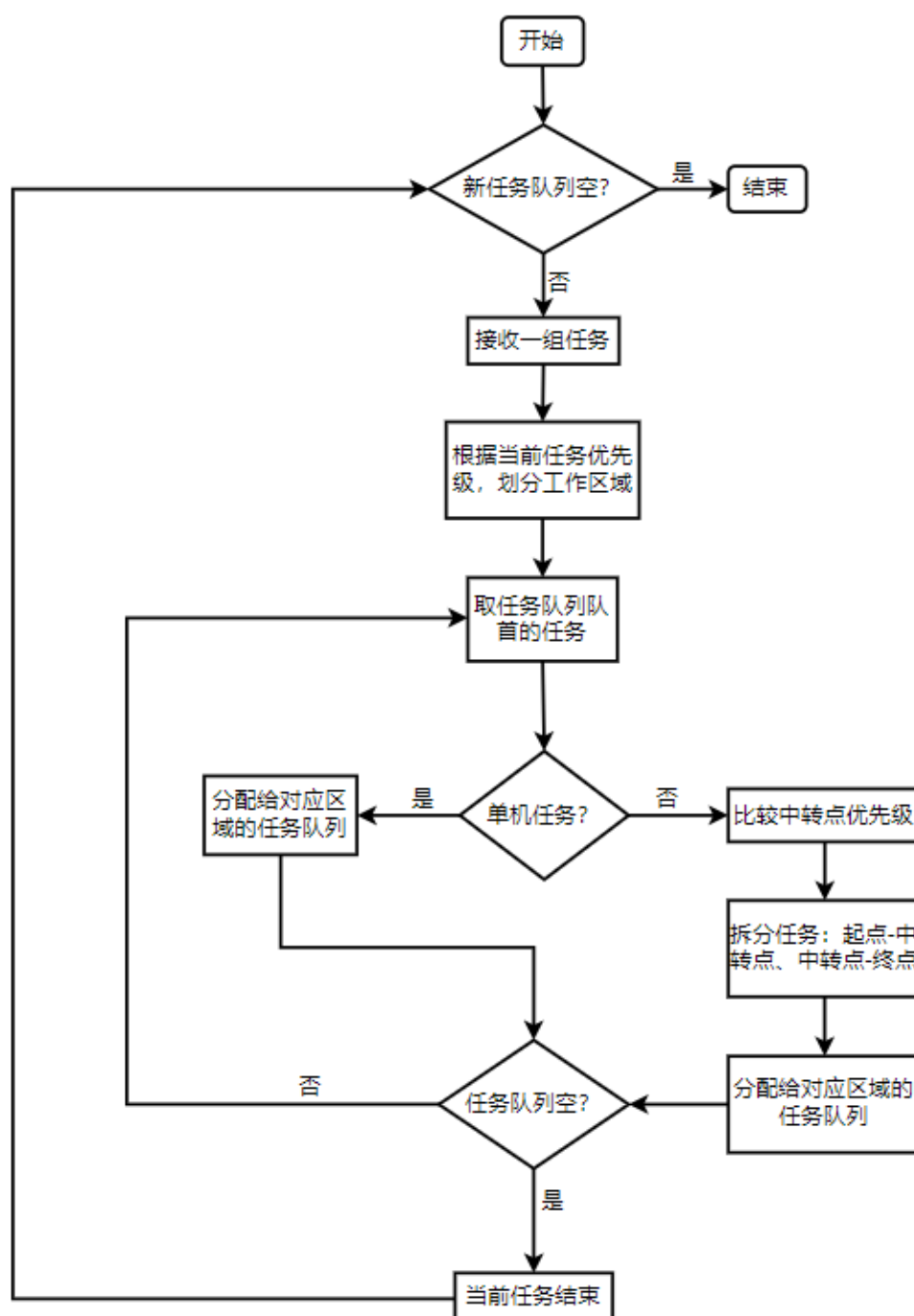


图 3.18 动态工作区划分算法流程图

3.4.3 性能测试

以下实验内容在仿真环境完成。仿真环境运行在 Win10 操作系统下, 使用 PyCharm 构建项目文件。任务场景是 10*10 的栅格地图, 0 表示通道, 1 表示障碍物和边界, 障

碍物和边界与实验室中实际的场景相对应。任务起点和终点采用 random 库随机生成，如果起点和终点位置相同，则舍弃此次任务。

首先，研究双向单通道场景中动态工作区算法的工作区划分次数与算法性能的关系。使用两个机器人协同完成搬运任务。定义性能指标：系统随机生成若干个搬运任务，将其转化成机器人的动作序列，包括执行、转弯，以 2 个机器人完成搬运任务所需动作序列的总时间作为评价指标。图 3.18 中，从左侧到右侧分别是：固定工作区、动态划分 2 次工作区、动态划分 5 次工作区、动态划分 10 次工作区得到的搬运任务总时间消耗。纵坐标表示完成所有任务所需要的总时间（用指令的条数表示）。

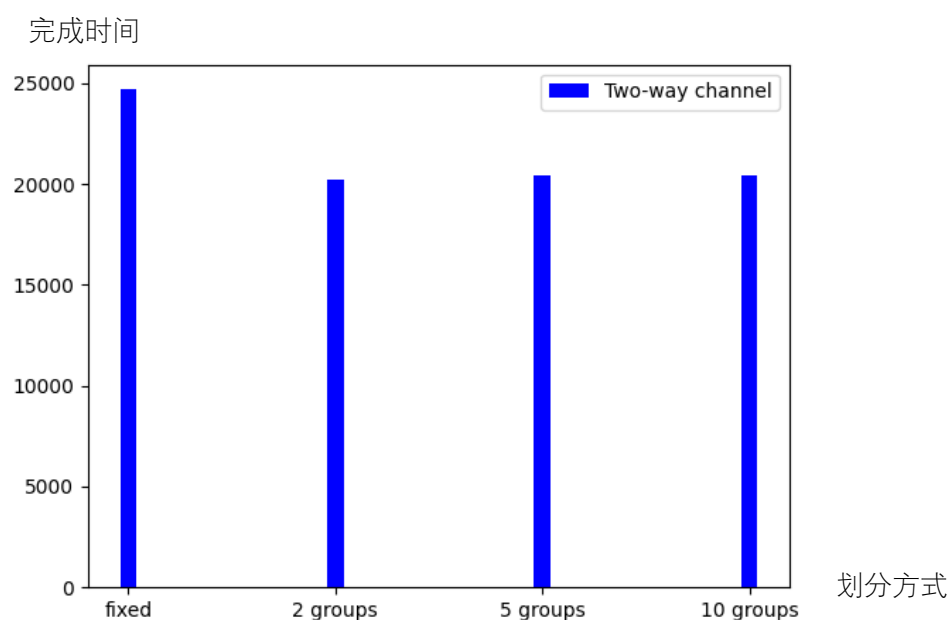


图 3.19 执行 10000 次任务性能比较

根据实验结果我们可以得出初步结论：动态划分工作区与静态固定工作区相比，算法性能有所提升；随着动态工作区划分次数的增加，完成任务总时间代价变化不明显。静态固定工作区可能出现多机任务数量过多的问题，这就会导致任务完成所需的总时间增加。通过动态工作区划分，可以尽可能的保留单机任务，减少多机任务，从而减少多机任务中引入中转区而产生的代价。由于 random 库在生成大量任务时具有随机分布的趋势，虽然工作区划分次数的增加，得到的可能仍是和第一次划分相同的结果，因此在本次实验中并没有呈现明显的性能差异。如果生成的任务具有一定的区域倾向性，才能体现动态工作区多次划分的优势。

接下来比较单向单通道场景和双向单通道场景下多机协同策略的性能。定义性能指标：系统随机生成若干个搬运任务，将其转化成机器人的动作序列，包括执行、转弯，以 2 个机器人完成搬运任务所需动作序列的总时间作为评价指标。在两种场景中，均使用相同的单机路径生成算法，即 3.2 节提到的单机机器人路径规划 A* 算法，测试不同的协同策略。单向单通道场景中，首先假设机器人完全不需要避免碰撞，再逐渐增加解决冲突需要的时间代价；双向单通道场景中，测试固定工作区和 5 次分配动态工作区两种协同策略，由于工作区的独立性，不需要考虑机器人的冲突问题。分别测试任务数量为 500、1000、5000、10000 时的协同搬运情况，得到图 3.19 中的结果。图 3.19 中纵坐标表示完成所有任务所需要的总时间（用指令的条数表示）。

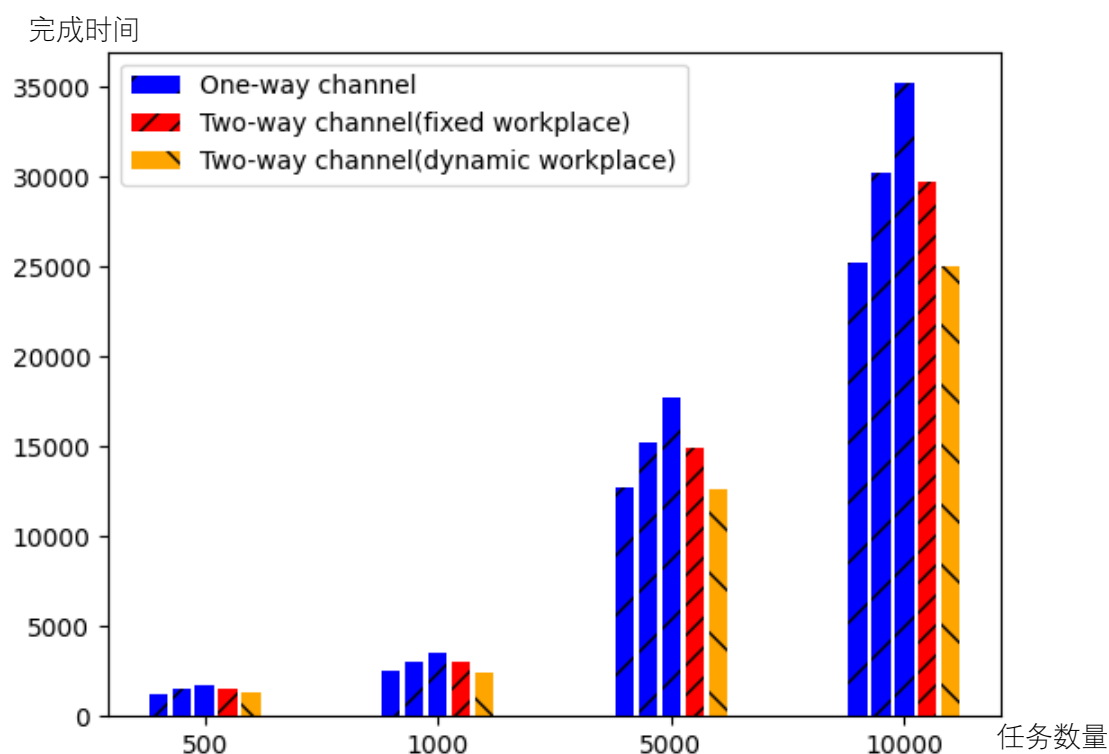


图 3.20 协同策略性能比较

根据实验结论可以得出初步结论：双向单通道场景下动态工作区算法完成任务需要的时间代价，略小于完全不考虑冲突的单项单通道场景的时间代价。根据 3.3.1 中的内容，双向单通道场景的任务效率约为单项单通道场景效率的 1.56 倍，这就导致即使在完全忽视机器人碰撞体积的理想条件下，单向单通道场景依旧没有显著优势。由于双向单

通道场景任务效率高的特点以及完全避免死锁的思想，动态工作区划分策略有一定的可行性。

3.5 本章小结

本项目使用栅格地图表示物理场景。通过比较不同的单通道场景，我选择了任务效率较高的双向单通道。为了解决双向单通道场景中相遇冲突带来的死锁问题，我设计了动态工作区划分策略，完全避免了死锁。经过仿真环境下的测试，这种调度策略并没有显著增加多机协同搬运任务的时间成本，证明了策略的可行性。

4 数字孪生子系统设计 with 实现

使用 Flask、SQLite 和 AJAX 技术实现。前端部分采用常规的 HTML、CSS、JavaScript 进行页面设计，使用 Echart 开源图表库和 jQuery 库来丰富功能。整个项目的架构大致如图 4.1 所示：

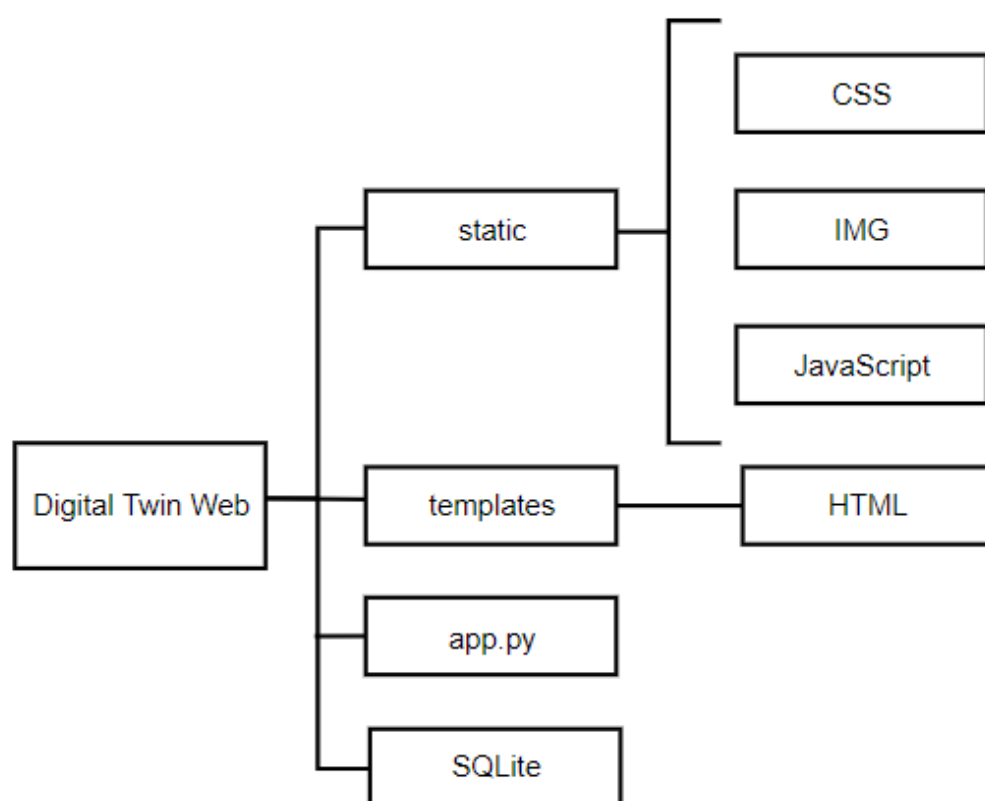


图 4.1 数字孪生子系统项目结构图

4.1 使用 SQLite 框架的数据库

数据库用于存储物理实体上传的各种数据，使用 python3 的 SQLite3 进行搭建。SQLite 是一个可以在当前程序进程内的直接链接的数据库，使用时不需要服务器，不需要配置，它是一个事务性的数据库引擎，特点是轻量级的，足够完成本项目在实验场景下的功能。在项目中使用时，首先使用 connect 方法连接到本地创建的数据库，然后创建一个 cursor 对象，即可使用各种 SQL 语句对数据库进行查询、创建、删除等操作。

本地数据库是一个关系型数据库。关系数据库由实体和实体间的联系构成。实体包括搬运机器人、状态参数表和任务状态表：搬运机器人的属性主键是编号，其他属性是电量、连接状态；状态参数表的主键是时间戳，其他属性是各种传感器数据；任务状态表的主键是时间戳，属性包括任务起点终点、任务进度、机器人坐标。关系包括机器人和状态表之间的“采集”关系，机器人和任务状态之间的“执行”关系。具体数据库 E-R 图如图 4.2 所示。

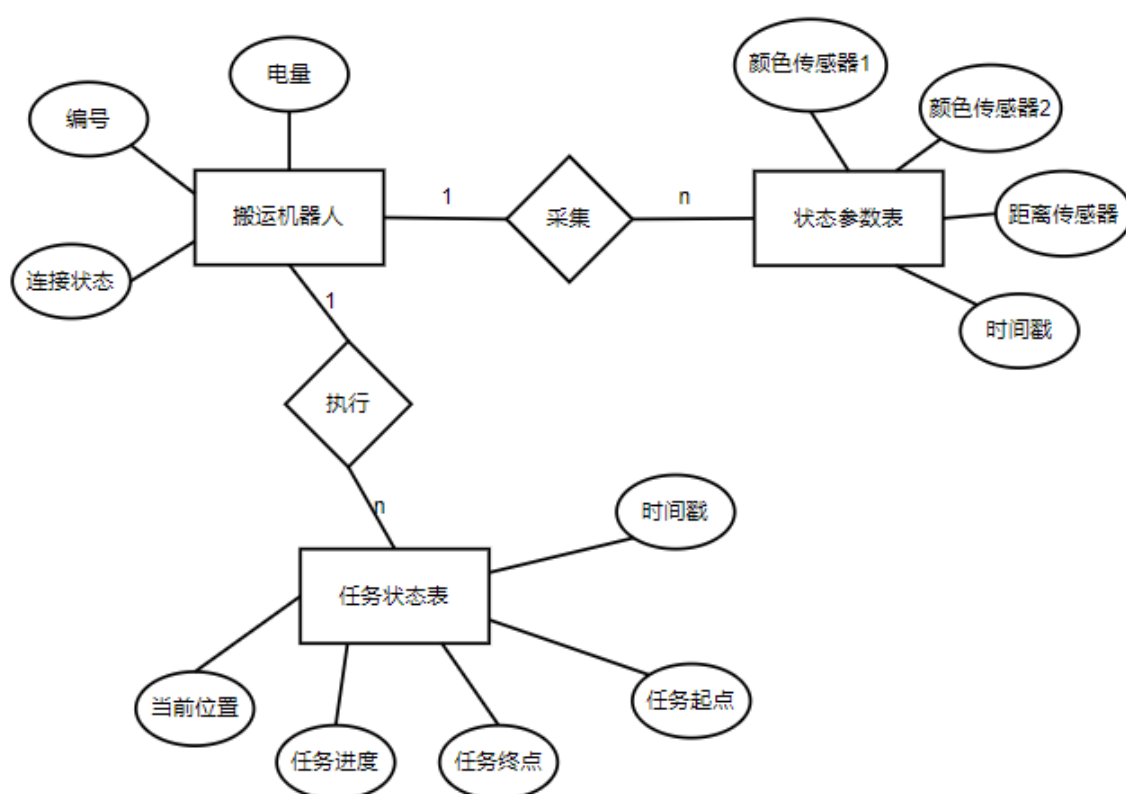


图 4.2 数据库 E-R 图

4.2 使用 Flask 框架的 Web 服务器

Flask 是一个轻量级的 Web 应用程序框架。所谓 Web 框架就是 Web 应用程序开发人员所使用的一个库和模块的集合。使用 Web 框架可以简化网络协议、线程管理等细节问题的处理。Flask 是用 Python 实现的，在同类型的框架中，Flask 代码量相对较小，轻便，易部署，并且有较好的安全性，非常适应 MVC(Model View Controller)开发模式。另一方面，Flask 还有很强的定制性。由于 Python 第三方库的丰富性，使用 Flask 开发

的项目既能保持核心功能简单，同时又能实现功能的丰富与扩展，用户可以实现个性化的网站定制，开发出结构简单但是功能丰富的网页应用。

Flask 基于 Werkzeug WSGI 工具包和 Jinja2 模板引擎。Web Server Gateway Interface (Web 服务器网关接口, WSGI) 就是 Python Web 应用程序的通用开发标准, 它规定了 Web 服务器和 Web 应用程序之间的通信要求。WSGI 定义了两部分功能, 一个是服务器, 另一个是应用程序。服务器在处理 Web 请求时, 会为应用程序提供环境信息及一个回调函数, 应用程序使用回调函数将处理后的结果返回给服务器。Werkzeug 就是一个 WSGI 工具包, 包含了请求、响应对象和实用函数的功能。Jinja2 是仿照 Django 模板的 Python 模板语言。总的来说, Flask 保持应用程序的核心简单且可扩展。

本项目主要完成了 Flask 运行环境搭建、创建 Flask 应用程序、Flask 路由方法以及 Flask 的 HTTP 请求方法。首先, Flask 运行环境是 Win10 系统下的 Virtualenv, 它是一个虚拟的 Python 环境构建器。通过 Flask 构造函数建立应用程序对象, 然后使用 run 方法运行。Flask 路由通过 route 函数实现, 它是一个装饰器, 用于告诉应用程序哪个 URL 应该调用相关的函数。这里我定义了两个路由: 一个是默认路由, 使用 HTTP 协议中的 GET 方法, 用于用户访问时呈现主界面的 HTML 文件; 一个是数据更新路由, 使用 HTTP 协议中的 GET 和 POST 方法, 在收到前端界面的数据更新请求后, 服务器会从数据库中获取最新的数据, 然后以 JSON 数据格式发送到前端界面。将数据库中的数据转化成 JSON 格式, 使用的是 Flask 中的 jsonify 方法。

综上, 我使用 Flask 搭建数字孪生系统的 Web 服务, 能够提供前端页面所需的静态 HTML 文件, 处理前端数据更新请求, 处理数据库中的数据并发送到前端页面。

4.3 前端可视化界面

前端开发主要使用 HTML、CSS、JavaScript 共同进行, HTML 定义了网页的内容, CSS 描述了网页的布局, JavaScript 控制了网页的行为。使用 Echart 开源图表库来完成数据可视化工作。主要显示了以下内容: (1) 传感器数据。包括两个颜色传感器和一个距离传感器。(2) 任务进度。根据任务路径和已经完成的路径, 计算的出的任务进度百分比。(3) 机器人数字结构图。包括机器人整体的三维模型、传感器位置和说明、连接状态、电量等。(4) 监控视频。使用安装广角摄像头的树莓派构建的远程视频传输服务, 基于 Mjpg-Streamer 方法实现监控视频在 web 上的实时访问。(5) 虚拟栅格地图。根据实际物理场景抽象得出的虚拟地图, 能够实时展示机器人的位置。在 Chrome 浏览器中的具体运行效果如图 4.3 所示。

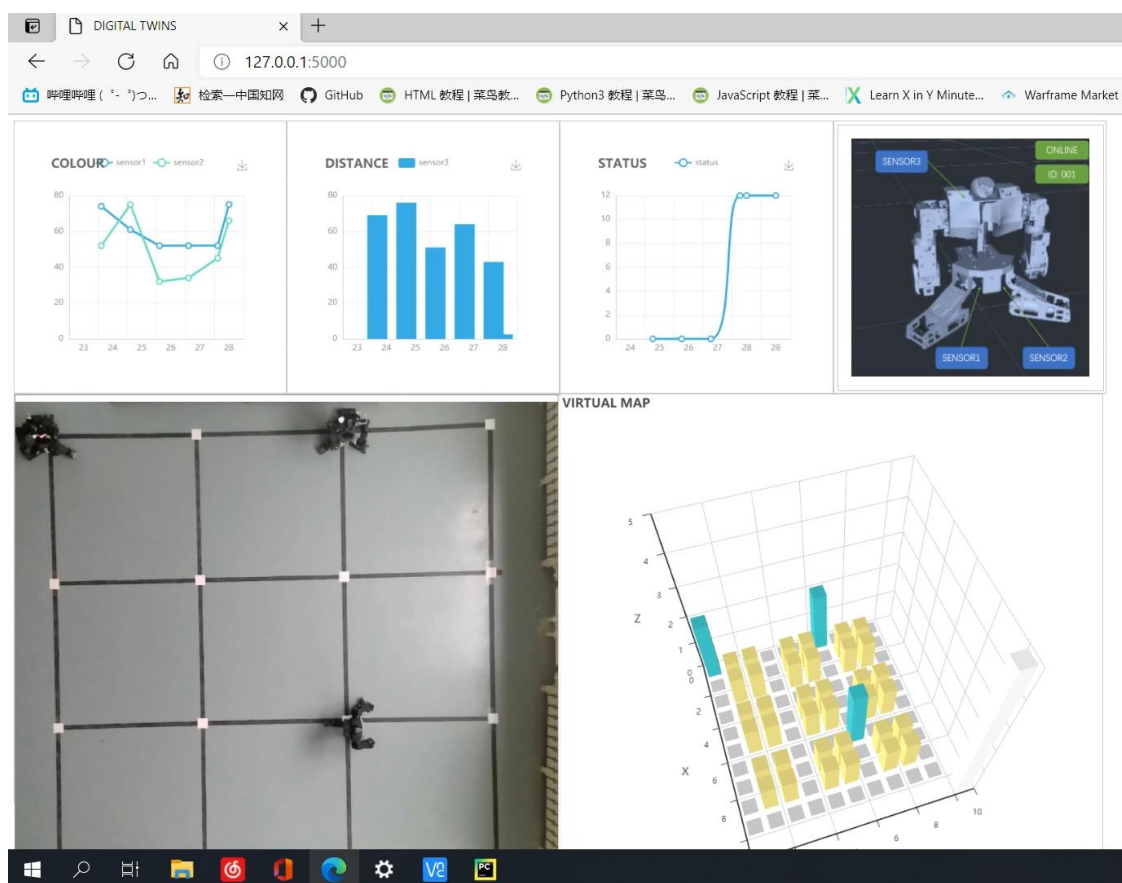


图 4.3 数字孪生前端界面

数字孪生前端页面需要包括折线图、散点图、柱状图、3D 虚拟栅格图等多种图表，以多维度地呈现物理实体的各种属性。同时，为了有更高的用户友好性，图表需要有很好的交互，以方便用户查看所需的数据。此外，由于物理实体一直处于工作状态，孪生实体需要实时呈现大量变化的数据，这对图形引擎的视觉效果和渲染效率提出了很高的要求。综合考虑上述需求和各种图表组件，我选择了 Echart 作为数据可视化界面设计的工具。

ECharts 是使用 JavaScript 实现的开源可视化库，它具有以下优点：（1）图表丰富且搭配自由。ECharts 能够绘制折线图、柱状图、散点图等基础图表，并且为图与图之间的组装和结合提供了支持。还可以通过 GL 等扩展库实现 3D 图表、交通动态图等高级图表。（2）支持多种数据格式。ECharts 通过 dataset 属性实现了二维表，key-value 等多种格式的数据，为开发者的数据转换省去了许多工作。除此之外，多个图可以公用一份数据，不需要为每个图表单独配置。（3）图表的交互性好。ECharts 提供了图例、视觉映射、数据刷选等交互组件，可以对数据进行多维度数据筛取、视图缩放、展示细节

等交互操作。(4)对于动态数据的优化。图表会根据数据的动态变化提供支持。检测到数据变化后, ECharts 根据数据的变化通过合适且流畅的动画进行过渡。

具体实施时, 首先使用 `echarts.init` 方法初始化一个图表对象, 然后使用 `setOption` 方法配置图表的参数。可配置参数包括标题 `title`、图例 `legend`、坐标系 `grid`、坐标轴、颜色主题等。`Series` 对象用于向图表区域插入图表, 需要指明图表的类型和数据集。实现实时更新数据需要传入新的数据并修改 `series` 中对应的数据项, 这里我使用 `Ajax` 实现数据异步更新, 具体方法将在后续段落中指出。完成每个图表的设计和配置后, 使用 `CSS` 完成图表区域划分和定位, 完成数字孪生前端可视化界面的设计工作。

4.4 Ajax 前后端数据交互方法

Ajax 即 Asynchronous JavaScript And XML (异步 JavaScript 和 XML), 使用 Ajax 技术网页应用能够快速地将增量更新呈现在用户界面上, 而不需要重载整个页面, 这使得程序能够更快地回应用户的操作。通常网页更新内容时, 必需刷新整个网页面^[33], 等待浏览器解析出新的 `HTML` 内容时, 用户才能够看到更新后的页面, 这并不能满足数字孪生对于实时性状态呈现的需求。通过 `ajax`, 可以让数字孪生前端实时向服务器发送数据更新请求, 服务器实时更新数据, 这样就能更好地实现物理实体与虚拟实体的状态同步。

本项目使用 `jQuery` 库来完成 Ajax 功能。`jQuery` 是一个快速、简洁的 JavaScript 框架, 它封装 JavaScript 常用的功能代码。通过 `jQuery` 向服务器发送请求, 服务器会把数据库中的数据以 `json` 格式发送到前端。具体实现方法如下。首次加载图表时, 通过 `jQuery` 的 `GET` 方法获取已经在数据库中存放的初始数据。为了实现数据实时更新, 我使用 `setInterval` 每经过一秒自动触发 `POST` 方法, 发送数据更新 `HTTP` 请求到服务器, 从而获得新写入数据库的各项数据。

4.5 物理实体状态评估功能

为了实现数字孪生系统对于物理实体的评估功能, 我采集了每个机器人的传感器数据, 机器人的坐标、朝向、任务进度等状态参数, 这些数据由 `Echart` 图表直观呈现; 实验场地部署了广角摄像头, 它们的视频信号将会实时呈现在页面中, 用于实验场地整体的监控; 此外, 还会展示与实验场地相对应的栅格地图, 上面将实时显示每个机器人的位置。这样就获得了一个具有多维性和实时性特征的数字孪生模型, 服务器就可以根据模型的当前数据和历史数据完成对于系统性能的评估功能。

4.5.1 使用 KMeans 聚类算法评估运行状态

机器人前方安装了两个颜色传感器，可以采集到机器人所在位置的颜色灰度值，机器人通过这两个数据来完成循迹和到达路口结点的动作。实际运行时，传感器需要区分三种颜色：循迹线的灰色、路口结点的白色、实验室地面的灰色，三种颜色对应的灰度值范围如图 4.4 所示：

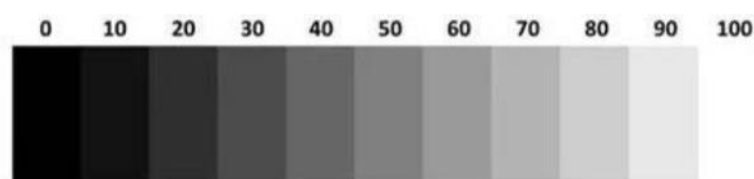


图 4.4 灰度值对应表

可以通过两侧传感器灰度值的差异来反映机器人的动作状态：两侧传感器读数对应白色，表明机器人到达路口结点；两侧传感器读数出现黑色和灰色，表明机器人正常循迹；两侧传感器读数间隔较大地交替出现黑色和灰色，表明机器人正在原地转向；两侧传感器长时间同时读数为灰色，表明机器人循迹异常，可能已经脱离搬运场地。

数字孪生服务器将上述传感器数据通过聚类算法处理，可以用于评估和预测机器人动作状态。我使用 KMeans 聚类算法，这个算法是一种迭代求解的聚类算法，能够将机器人颜色传感器产生的数据点划分成多个聚类，这样就能从聚类结果中选出异常数据集的聚类中心。

4.5.2 KMeans 聚类算法性能分析

我收集了机器人直行、转弯、到达路口结点和走出场地这四种情况时产生的颜色传感器数据，按照数据的分布特点在仿真环境下模拟实际执行任务时产生的数据。仿真环境运行在 Win10 操作系统下，使用 PyCharm 构建项目文件。任务场景是 10*10 的栅格地图，0 表示通道，1 表示障碍物和边界，障碍物和边界与实验室中实际的场景相对应。

以下是根据仿真环境两次不同的搬运任务绘制的机器人颜色传感器数据图 4.5，图中横纵坐标对应的是左右两个传感器的读数。图 4.5 中，圆形点表示正常运行时采集的数据，三角形点是人为制造的异常运行数据。图 4.6 对应的散点图是使用 K=5 的 KMeans 算法处理后的数据，可以看到异常数据基本可以从正常数据中筛选出来（图 4.6 子图 a 中的三角形点、子图 b 中的五角星点为异常数据）。

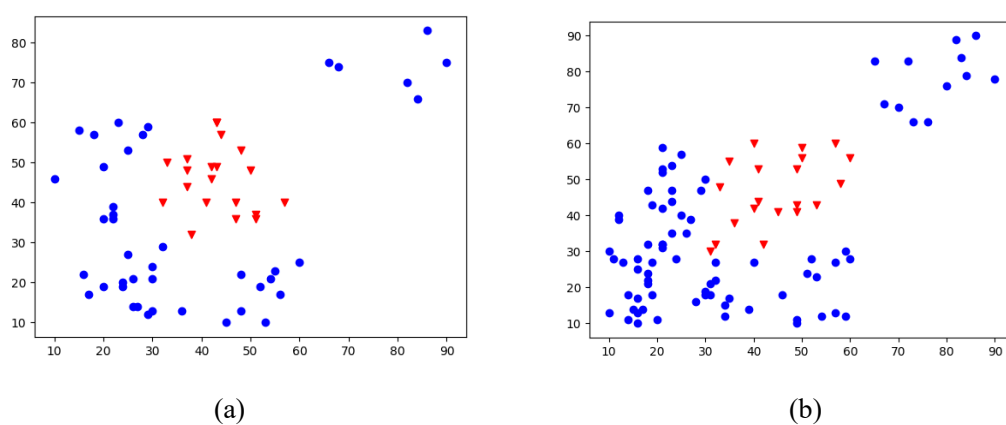


图 4.5 颜色传感器读数散点图

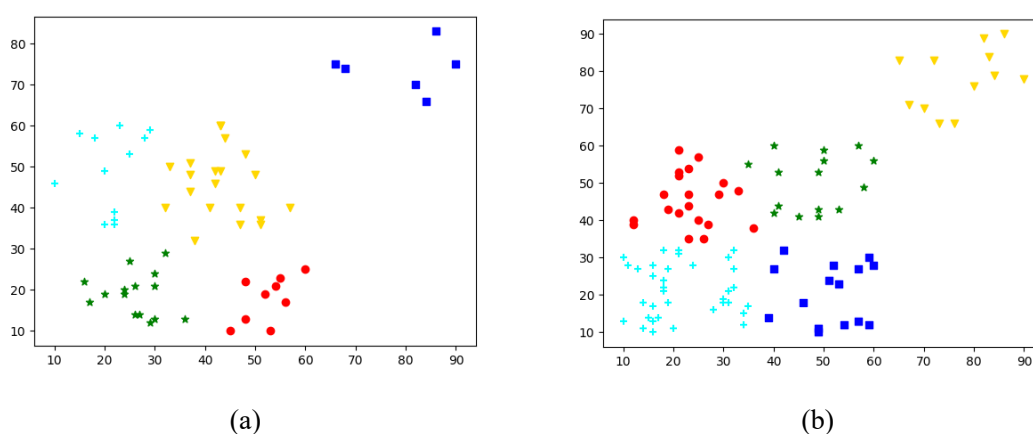


图 4.6 聚类结果

表 4.1 KMeans 聚类算法预测准确率

任务长度	正常数据数	异常数据数	测试数据数	预测准确率
3	36	20	10	0.65
6	78	40	10	0.73
11	120	60	10	0.74

数字孪生服务器通过历史数据得到聚类中心后,就可以给机器人实时上传的数据进行类别判断。根据算法预测的准确率,系统规定如果机器人连续 3 次都处于异常工作状态,服务器就会发出警报,并停止机器人的动作。使用 KMeans 聚类算法的优点是,聚类中心是根据最近的历史数据生成的,能够动态变化。在理想环境中,系统可以将“两个传感器读数都是灰色”作为判断标准,判断机器人处于栅格地图外。此时设置灰色对

应的灰度值为边界即可实现报警。但是，由于传感器的精度、实验室中地面材质、实验室的光照条件等原因，灰度值传感器采集到的数据并不稳定，并不能找到这样一个确定的边界值。使用 KMeans 聚类算法就可以避免这种影响，它是一个不需要大量数据集、不需要长时间训练的聚类算法，根据少量的传感器数据即可动态地得到当前环境下的聚类中心，从而提高了预测结果的可靠性。

4.5.3 整体协同搬运任务效率评估

数字孪生服务器可以通过历史数据和实时数据对比来评估协同搬运任务效率。物理实体每次都会接收相同规模的任务，数字孪生服务器可以统计每次任务完成的时间并计算平均值。当数字孪生系统运行时，前端界面可以呈现当次任务的完成时间与系统平均完成时间，如图 4.7 所示。



图 4.7 任务完成时间评估图

通过实时数据与历史数据的任务完成时间均值进行比较，我们可以直观地找到完成时间较长的任务（如图 4.7 中的任务 44），通过查找执行此搬运任务的数字孪生数据，就可以分析任务效率低的原因，这对于优化多机协同系统的策略和算法有着重要的意义。

4.6 本章小结

本项目搭建了一个数字孪生系统。系统通过收集物理实体的传感器数据、状态参数、任务进度、位置坐标、监控视频等数据建立了虚拟实体模型，使用 Web 可视化界面实时呈现给用户。系统可以利用历史数据和 KMeans 聚类算法对机器人运行状态进行预测。通过统计任务执行时间，系统可以标注出异常任务，方便用户对物理实体进行研究和优化。该系统可以在实验室场景中运行。

结 论

近年来,物联网、智慧城市等领域在高速发展的同时也面临了诸多问题,其中就包括物理系统和数字系统之间割裂、交互不充分。数字孪生技术就是解决此问题的一种方法。数字孪生是通过数字化的方式,建立针对物理实体的多维度、多学科、多物理量的动态虚拟模型,来反映物理实体在真实物理场景中的各种属性、行为等,对于物理系统的评估和维护有着重要的意义。

在现代智能化工厂中,仓储系统中的多机器人协同搬运场景非常普遍。如何解决多机器人冲突问题、提高任务效率是重要的研究方向。本文从该场景切入,深入研究了多机器人路径规划和协同策略,并针对该场景搭建数字孪生系统,设计了一种可行的智慧工厂监控方案。主要成果如下:

(1) 在多机协同问题上,对比和分析了多机协同场景的表示方法,针对单项单通道场景任务效率较低的问题,选择双向单通道场景。为了解决双向单通道场景中的对向冲突死锁问题,设计了 A*路径规划算法与动态工作区划分算法相结合的协同策略,完全避免了死锁问题,并且提高了任务效率。

(2) 在数字孪生方面,经过研究和比较建模方法后选择数字孪生五维模型进行设计。最终使用 Flask 框架和 SQLite 数据库,结合 jQuery、Echart 等工具包完成了数字孪生 Web 平台,该平台具有多维性和实时性的特点,能够收集物理场景中的各项数据,实时传输到前端界面进行可视化展示。

(3) 将上述系统部署在实验室的多机协同搬运场景中。通过收集实体机器人产生的测试数据,完善了数字孪生系统对物理实体进行预测和评估的功能。能够通过 KMeans 聚类算法分析机器人实体的运行状态,并且可以根据历史数据评估多机协同搬运的任务效率。

本文仍然存在一些不足需要改进:

(1) 由于时间原因,没有解决该数字孪生监控系统中的信息安全性问题,系统可能会受到攻击导致机器人收到错误的指令,或是泄露数据库中物理实体的运行数据。

(2) 在进行多机协同策略研究时,没有采用当前较为热门的强化学习算法。强化学习在此类问题上能够得到效果更好、地图泛用性更强的策略。

参 考 文 献

- [1] 陶飞, 刘蔚然, 张萌等. 数字孪生五维模型及十大领域应用[J]. 计算机集成制造系统, 2019, 25(01):1-18.
- [2] 杨林瑶, 陈思远, 王晓等. 数字孪生与平行系统:发展现状、对比及展望[J]. 自动化学报, 2019, 45(11):2001-2031.
- [3] 刘青, 刘滨, 张宸. 数字孪生的新边界——面向多感知的模型构建方法[J]. 河北科技大学学报, 2021, 42(02):1-15.
- [4] T. MUKHERJEE, T. DEBROY. A digital twin for rapid qualification of 3D printed metallic components[J]. Applied Materials Today, 2019, 14:59-65.
- [5] THOMAS H.-J. Uhlemann, Christian Lehmann, Rolf Steinhilper. The Digital Twin: Realizing the Cyber-Physical Production System for Industry 4.0[J]. Procedia CIRP, 2017, 61:335-340.
- [6] 刘大同, 郭凯, 王本宽等. 数字孪生技术综述与展望[J]. 仪器仪表学报, 2018, 39(11):1-10.
- [7] GUO A, YU D, HU Y et al. Design and implementation of data collection system based on CPS model[C]. In: Proceedings of the 2015 International Conference on Computer Science and Mechanical Automation (CSMA). Hangzhou, China: IEEE, 2015.
- [8] ELATTAR M, CAO T, WENDT V et al. Reliable multipath communication approach for internet-based cyber-physical systems[C]. In: Proceedings of the 26th International Symposium on Industrial Electronics (ISIE). Edinburgh, England: IEEE, 2017.
- [9] FENG XIA, LINQIANG WANG, DAQIANG ZHANG, DAOJING HE et al. An adaptive MAC protocol for real-time and reliable communications in medical cyber-physical systems[J]. Telecommunication Systems, 2015, 58(2):125-138.
- [10] CHRISTIAN GEHRMANN, MARTIN GUNNARSSON. A Digital Twin Based Industrial Automation and Control System Security Architecture[J]. Journal of Technology & Science, 2020, 16:527-535.
- [11] FENG M, XU H. Deep reinforcement learning based optimal defense for cyber-physical system in presence of unknown cyberattack[C]. In: Proceedings of the 2017 IEEE Symposium Series on Computational Intelligence (SSCI). Honolulu, USA: IEEE, 2017.
- [12] CLAUDIO MANDOLLA, ANTONIO MESSENI PETRUZZELLI, GIANLUCA PERCOCO et al. Building a digital twin for additive manufacturing through the exploitation of blockchain: A case analysis of the aircraft industry[J]. Computers in Industry, 2019, 109:134-152.

- [13] YUJIAO HU, YUAN YAO, WEE SUN LEE. A reinforcement learning approach for optimizing multiple traveling salesman problems over graphs[J]. Knowledge-Based Systems, 2020, 204:106244.
- [14] 梁晓辉, 慕永辉, 吴北华等. 关于路径规划的相关算法综述[J]. 价值工程, 2020, 39(03):295-299.
- [15] 于赫年. 仓储式多 AGV 动态调度系统研究[D]. 黑龙江:哈尔滨工业大学, 2019.
- [16] 夏清松. 复杂环境下多移动机器人协同路径规划[D]. 湖北:武汉科技大学, 2019.
- [17] 关英姿, 刘文旭, 焉宁等. 空间多机器人协同运动规划研究[J]. 机械工程学报, 2019, 55(12):37-43.
- [18] 张兴国, 张柏, 唐玉芝等. 多机器人系统协同作业策略研究及仿真实现[J]. 机床与液压, 2017, 45(17):44-51.
- [19] 张丹露. 基于智能仓库的多机器人动态路径规划关键技术研究[D]. 北京:中国科学院大学(中国科学院重庆绿色智能技术研究院), 2017.
- [20] LIANA MACHADO, ROBERTO SCHIRRU. The Ant-Q algorithm applied to the nuclear reload problem[J]. Annals of Nuclear Energy, 2002, 29(12):1455-1470.
- [21] AKSHAY KUMAR GURUJI, HIMANSH AGARWAL, D. K. PARSEDIYA. Time-efficient A* Algorithm for Robot Path Planning[J]. Procedia Technology, 2016, 23:144-149.
- [22] 肖绍. 基于改进蚁群算法的无人机航迹规划研究[D]. 江西:南昌航空大学, 2016.
- [23] 周松儒. 遗传算法的混合改进研究及其应用[D]. 广西:广西大学, 2014.
- [24] 李擎, 张超, 陈鹏等. 一种基于粒子群参数优化的改进蚁群算法[J]. 控制与决策, 2013, 28(06):873-878+883.
- [25] 夏炎, 隋岩. PRM 路径规划算法优化研究[J]. 应用科技, 2010, 37(10):1-5.
- [26] 曾庆立, 李丽华, 唐圣学. 基于神经网络路径规划的硬件设计[J]. 吉首大学学报(自然科学版), 2007(06):74-76.
- [27] 郑秀敏, 顾大鹏, 刘相术. 基于栅格法-模拟退火法的机器人路径规划[J]. 微计算机信息, 2007(05):247-248+279.
- [28] 田东平, 迟洪钦. 混合遗传算法与模拟退火法[J]. 计算机工程与应用, 2006(22):63-65.
- [29] 徐精明, 曹先彬, 王煦法. 多态蚁群算法[J]. 中国科学技术大学学报, 2005(01):62-68.
- [30] 谢云. 模拟退火算法综述[J]. 微计算机信息, 1998(05):63-65.
- [31] LYDIA E. KAVRAKI, PETR SVESTKA, JEAN-CLAUDE LATOMBE et al. Probabilistic roadmaps for path planning in high-dimensional configuration spaces. [J]. IEEE Trans. Robotics and Automation, 1996, 12(4):566-580.
- [32] CAI C H, FERRARI S. Information-driven sensor path planning by approximate cell decomposition[J]. IEEE Trans on Systems, Man, and Cybernetics, Part B: Cybernetics, 2009, 39(3): 672-689.
- [33] 李华, 张洪伟, 杨庆等. 基于 SSH+Ajax 框架技术的开放式实验预约系统的研究和实现[J]. 甘肃联合大学学报(自然科学版), 2011, 25(04):79-81+96.

修改记录

- (1) 毕业设计（论文）题目修改
- (2) 毕业设计（论文）内容重要修改记录

第二次修改记录：

第 16 页 3.4，**修改前：**现 3.4 内容为原 3.3.3 的内容

修改后：将“动态工作区划分算法”独立成 3.4

第三次修改记录：

第 7 页 2.1，**修改前：**现 2.1 内容为原 1.3 的内容

修改后：将“系统设计要求”移动到第二章中

- (3) 毕业设计（论文）外文翻译修改记录
- (4) 毕业设计（论文）正式检测重复比

记录人（签字）：

指导教师（签字）：

致 谢

2018年，我从化机转专业到计算机专业，从此人生的轨迹就发生了重要的转折，首先感谢自己做出的正确选择以及为了这个决定付出的努力。新学期开始，我加入嵌入式与物联网实验室，正式开始接触到计算机专业相关的项目。三年的时光匆匆流逝，虽然留下了一些遗憾，但依旧是人生中极为宝贵的经历。

感谢我的导师丁男副教授。从大二进入实验室以来，丁老师一直为我指明了学习方向，并且提供了丰富的实验平台和资源。2019年，在丁老师的指导下，我获得了全国物联网竞赛华为杯国家级二等奖，这是我本科生涯中有幸获得的最高荣誉。毕业设计期间，丁老师虽然远在新疆，依然悉心地远程指导我，从论文的选题、到项目整体的构建思路，丁老师都提出了宝贵的建议。感谢丁老师这些年为我所做的一切。

感谢物联网实验室的董校老师。刚进入实验室的我面对众多陌生的知识无所适从，董老师耐心地为讲解单片机开发的基础知识，解决了许多困扰我的难题。平日里董校老师平易近人，经常和我们探讨学习生活中的各种有趣经历，实验室里总是充斥着轻松愉悦的氛围。

感谢辅导员范苏月老师和戴士焱老师。除了认真负责地完成服务学生的工作外，他们在我意志消沉的时候耐心地听我倾诉，鼓励我克服逆境，在很多人生问题上提出了宝贵的意见。

感谢和我一起参加比赛、完成研究项目和各种课程设计的好朋友们。团队项目是最好的学习技能和交流经验的方式，虽然有时过程磕磕绊绊，但是我们始终互相信任，互相帮助，最终总能获得优秀的成果。

感谢实验室的师兄师姐和朋友们。感谢赵雅文师姐和王守灿师兄分享的宝贵经验以及提供的各种帮助；感谢高鑫治和夏恬凡同学，我们在完成毕业设计的过程中互相分享交流，顺利完成了研究项目和论文。

感谢我的室友，以及一起约饭游玩的朋友们。我们在一起快乐地度过了学业外的闲暇时光，日后有缘再相见，苟富贵，勿相忘。

感谢始终在背后支持我的家人们。

最后，感谢论文评审老师，感谢您能够在百忙之中为我的论文提出宝贵的意见。