分类号：_____ 密级：_____

UDC：_____ 编号：_____

工学硕士学位论文

基于 RRT 的复杂环境下机器人路径规划

硕士研究生：李威洲

指导教师：王能建 教授

学科、专业：机械制造及其自动化

论文主审人：钟宇光 副教授

哈尔滨工程大学

2012 年 3 月

哈尔滨工程大学

学位论文原创性声明

本人郑重声明：本论文的所有工作，是在导师的指导下，由作者本人独立完成的。有关观点、方法、数据和文献的引用已在文中指出，并与参考文献相对应。除文中已注明引用的内容外，本论文不包含任何其他个人或集体已经公开发表的作品成果。对本文的研究做出重要贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

作者（签字）：李成洲

日期：2018 年 12 月 30 日

哈尔滨工程大学

学位论文授权使用声明

本人完全了解学校保护知识产权的有关规定，即研究生在校攻读学位期间论文工作的知识产权属于哈尔滨工程大学。哈尔滨工程大学有权保留并向国家有关部门或机构送交论文的复印件。本人允许哈尔滨工程大学将论文的部分或全部内容编入有关数据库进行检索，可采用影印、缩印或扫描等复制手段保存和汇编本学位论文，可以公布论文的全部内容。同时本人保证毕业后结合学位论文研究课题再撰写的论文一律注明作者第一署名单位为哈尔滨工程大学。涉密学位论文待解密后适用本声明。

本论文（□在授予学位后即可 ☑在授予学位 12 个月后 □解密后）由哈尔滨工程大学送交有关部门进行保存、汇编等。

作者（签字）：李成洲　　导师（签字）：

日期：2018 年 12 月 30 日　　2018 年 12 月 30 日

分类号：_____ 密级：_____

U D C ：_____ 编号：_____

工学硕士学位论文

基于 RRT 的复杂环境下机器人路径规划

硕 士 研 究 生：李威洲

指 导 教 师：王能建 教授

学 位 级 别：工学硕士

学 科 、 专 业：机械制造及其自动化

所 在 单 位：机电工程学院

论 文 提 交 日 期：2011 年 12 月 20 日

论 文 答 辩 日 期：2012 年 3 月 14 日

学 位 授 予 单 位：哈尔滨工程大学

Classified Index:
U.D.C:

A Dissertation for the Degree of M.Eng

Robot motion planning based on Rapid-exploring Random Trees in complex environment

Candidate: Li Weizhou

Supervisor: Prof. Wang Nengjian

Academic Degree Applied for: Master of Engineering

Specialty: Mechanical Manufacture and automation

Date of Submission: Dec. 20, 2011

Date of Oral Examination: Mar. 14, 2012

University: Harbin Engineering University

摘　　要

机器人学是一门集理论、设计、制造和应用的综合学科。随着科技的进步，机器人的应用已经越来越广泛。其中移动机器人尤其受到人们的欢迎。路径规划技术在移动机器人中的地位十分重要。有效的路径规划技术需要特定环境下的机器人导航算法。快速扩展随机树是一种数据结构和算法，这种算法为了能够到达目标点能够不断地生成新的节点。这一属性使得它非常适合应用在非完整系统中。在研究中人们发现该算法在解决路径规划问题中有着不错的表现。

将基本快速扩展随机树算法加以改进，提出偏向目标型快速扩展随机树算法。并对两者进行了仿真实验对比。结果证明改进型能够避免基本快速扩展随机树搜索过于平均，用时较长等缺点，使搜索具有一定的倾向性，提高了搜索的速度与效率。

将偏向目标型快速扩展随机树算法加以应用并解决了两种复杂环境下的路径规划问题。这两种问题均从轮式移动机器人运动学模型出发，前者应用快速扩展随机树的改进算法，主要解决非完整约束下的移动机器人动态环境下的运动规划问题。而后者应用快速扩展随机树改进型与人工势场法结合的方法，主要针对人工势场法容易陷入局部极小等缺点，在结合了人工势场法局部搜索优点的基础上结合快速扩展随机树算法进行全局规划。避免了可能产生的局部极小。然后通过计算机仿真试验与对比试验，验证了以上两种方法的有效性。

关键词：偏向目标型快速扩展随机树；复杂环境；动态规划；非完整系统；移动机器人

ABSTRACT

Robotics is the research of the technology associated with the theory, design, manufacture and application of intelligent machines. With developments in this technology, the application of robotics is becoming more and more extensive; and mobile robot is one type of robot that is rapidly increasing in relevance, application and demand. Effective path planning techniques are required to develop navigation algorithms for successful mobility of robots in their particular environments. The Rapidly-Exploring Random Tree (RRT) technique is a search system used in path planning. This system can generate new nodes in order to reach the goal point. It makes the algorithm easily meet the needs of the kinetics and dynamics constraints of the system and it shows a good performance in tackling the path planning problems in recent years.

The basic RRT algorithm is modified and a new algorithm is proposed called Bias-goal RRT algorithm. The two algorithms are compared through computer simulation. Results show that the modified one has taken care of some of the shortcomings of the basic one such as long searching time and non-heuristic searching and it can increase the speed and efficiency compared with the basic one.

The modified RRT algorithm is applied to solve two kinds of path planning problems in complex environment. Both of the two problems consider the kinetic model of the wheeled robot. The former one uses the modified RRT algorithm mainly to solve the path planning problems in dynamic environment of the mobile robot with nonholonomic constraints. The later one uses the algorithm combined with modified RRT and artificial potential field to avoid the shortcomings of the artificial potential field method, such as local minimum. The combined algorithm uses artificial potential field for the local planner while the RRT algorithm is used for global path planning. With the help of the two algorithms, local minima are avoided. The simulation results show that the above two methods are valid and can be applied in path planning for wheeled mobile robots.

Keywords: Bias-goal rapidly-exploring random tree; complex environment; dynamic planning; nonholonomic system; mobile robot

目　录

第 1 章　绪论

1.1　引言

机器人学这门科学非常庞大，它包含了系统工程、机械工程、计算机技术、材料学等诸多领域。并且涉及理论、设计、制造和应用等诸多方面。路径规划研究的重点就是要使机器人自动并且安全的从起始位资移动到目标位姿。因此，路径规划问题是机器人研究中重要的组成部分。该问题在机器人的各种任务中经常出现。也就是怎样快速并准确的在各种复杂环境中找到期望的路径。

在经济、军事等很多方面机器人路径规划都十分重要。所以很多学者都对此开始了重点研究。研究主要包括算法与实验两部分。最近，美国科学家成功研制了 courage 与 opportunity 两辆火星登陆车使得路径规划技术应用于外太空领域。如今，路径规划已经应用到机动车的导航中，各大汽车生产商已经开发出多种导航软件与自主泊车软件。这说明我们的生活已经与这门学科密不可分[1]。

1.2　机器人技术的发展与意义

科学家们从上世纪 40 年代着手研制机器人。上世纪 50 年代美国通用公司研制并开发了第一部用于生产的机器人。从那个时候开始机器人开始了生产活动。市场经济的发展使得机器人可以被自由买卖以使得普通人都有机会使用。在上世纪七、八十年代机器人技术与生产出现了飞速发展的趋势。主要应用于汽车的生产。企业应用机器人用于工业生产降低了生产成本，提高了工作效率。减轻了一线工人的负担。八十年代以后机器人开始稳定发展。从机器人诞生之日起到现在主要经历了三次技术改革[2]：

一代机器人非常简单。没有传感器的帮助，用开关进行控制与编程控制。机器人内部装有记忆存储器，各种工作程式和边界条件都需编程给定。一代机器人并不具学习能力。也没有智能算法的支撑，只是机械的执行命令。

二代机器人已经具有部分感知环境功能与部分的自适应能力。机器人身上装有普通的简易搜索装置，该装置能够是机器人感知简单的外部环境与自身状态。并以这些状态为参数进行内部控制。简单的传感器可以感知光线与作用力，所以二代机器人能部分感知环境。

三代机器人即发展到今天的智能机器人，智能机器人已经趋于复杂化与智能化。这

种机器人安装了各种不同的环境感知装置以学习外部的环境变化。能够自觉处理从周围环境获得的信息并自主的完成人们交给的任务。这一切都由智能算法作为技术支撑。在智能算法的帮助下，机器人能根据环境的变化自主的调整自己的行为方式。同时形成自己的知识库，具有强大的学习能力。

移动机器人是机器人技术的一个主要研究领域与重要分支，到如今科学家们研究移动机器人已经有几十年的历史了。移动机器人可以感知外部信息并自主的完成人们交给的任务。机械电子学、材料学与计算机科学为移动机器人的发展提供了强大的技术支持。

移动机器人因工作方式不同，控制方式也各异。比如说在汽车制造领域应用广泛的机械手，它们运动方式比较单一，位置也相对固定，外界干扰也很少。所以控制方法也比较固定。但是移动机器人的工作环境非常复杂。经常能遇到很多意想不到的事件，运动环境的复杂性使得运动本身成为一个非常复杂的课题。即怎样摆脱运动中的干扰与机器人内部的不确定性。

研究者的最终目的是要是机器人在传感器的辅助之下充分的学习外部信息与机器人本体的运动状态。做到在不确定的外部条件中能正确的决定自身的运动方式并成功的到达人们期望的目标点。

由于自主移动机器人在经济、军事、科研等诸多领域具有不可估量的研究价值。这一课题一直以来都吸引着大量科研人员积极探索并且相继被列入世界各国的高科技发展计划[6-7]。

如果根据任务类型划分可以将移动机器人归为野外任务型和建筑物任务型。建筑物任务型机器人的工作环境主要是建筑物中，如学校、居民区、工厂等城市中建筑物，还有如电站、大坝等城市外建筑物[8]。在城市中建筑物中机器人主要执行打扫环境、物品传递、危险物品拆除等任务；在城市外建筑物中，由于建筑物自身的设计以及危险性的限制使得人类难以接近目标区域。这就需要有特殊构造机器人去执行任务。比如在本世纪初的埃及金字塔科学考察中，科学家们利用一个小型机器人进入人类不能通过的小走廊里进行探测。受到了全世界的瞩目[9]。城市外建筑类型任务的另一个重要表现就是核电站的原料与废料的搬运工作。

在野外型任务中，移动机器人也有重要的应用。比如在战争遗留下来的各种弹药的搜寻以及各种爆炸物的拆除都不能够用人来执行。机器人在这方面的作用显而易见。而且移动机器人在军事领域危险地段的巡逻作战方面也有重要应用，如军队中使用的无人机就是很好的例子。

1.3 移动机器人运动规划研究概述

移动机器人是机电一体化研究中的最高峰，它汇聚了各个尖端学科的最先进的研究成果。运输功能是移动机器人在生产领域的主要功能，其次还有装配，涂装等等。移动机器人在粮食生产、医学等领域也有不同程度的应用[1]。

路径规划是机器人学研究的主要分支，最主要的问题集中在如何解决不确定条件下的移动机器人路径规划问题。已知的解决方案有很多，如人工势场法、随机采样算法、以及诸多启发式算法等。在最近的研究中，科学家们改进了很多算法并提出了诸如仿生算法的新算法。路径规划算法已经趋于复杂化与智能化[3]。

现存的算法因为种种的原因在功能上以及应用上受到很大的限制。这种限制体现在例如：环境范围过大或者搜索的耗时过长等等。这些因素都限制了算法的发挥。特别是机器人处于复杂环境中的时候，算法处理的信息量太大以至于无法工作。这种问题在路径规划算法的研究中经常出现。因此，路径规划的算法要不断改进和创新，尤其是在未知条件下的路径规划更需要创新。创新的目标是要使机器人快速、准确的适应环境的变化并调整自身的运动状态并寻求一条最优或次优的路径。

路径规划主要分为全局路径规划和局部路径规划。在下文中主要针对路径规划的发展进行了叙述，然后例举了很多路径规划的算法并比较了他们的长处与短处。结尾展望了未来对路径规划技术的需求及其发展前景[3,4]。

1.3.1 移动机器人路径规划的分类

在深入了解这些不同的方法之前，首先要对这些算法进行区分。最重要的是对全局避障方法和局部避障方法进行区分。

全局路径规划方法通常是指如路径划或运动规划方法。他们把周边环境的信息存储在一张图里，并且利用这张图找到可行路径，目标点可以是一个坐标集，也可以是对车辆状态的完整描述，包括速度，方向角等等。

全局路径规划方法是一种利用环境的全局信息的方法。这包括车辆在当前状态下探测不到的信息。这种全局方法的不足就是要花费很长时间进行运算还有很高的内存使用率。为了等待计算结果，计算机可能会把原本很短的运算时间加以延长，所以从这一点上来说，全局路径规划法不适用于快速变化的动态环境，原因就是它的运算时间太长了。

而局部路径规划法可以大大缩短运算时间。它们对变化环境的反映非常迅速，这使他们非常适用于动态环境，它们只考虑机器人的瞬时环境是局部路径规划法速度快的主

要原因。局部路径规划能够很好的朝目标前进，但不能保证一定能够到达目标点。换句话说：局部路径规划不能全局收敛。

全局路径规划法经常被称作“深思熟虑”的方法，而局部路径规划法则被称为“交互法”或“发射法”。

全局路径规划法也可能找不到可行的路径，但是它们相比与局部路径规划法来说更容易找到可行路径。

全局和局部路径规划法两者都有他们的局限性。混合方法试着结合全局路径规划法和局部路径规划法来解决这些问题。如图 2.1 所示的“双层系统”：全局规划法规划出一条路径然后局部路径规划法尝试着跟踪这条路径[10]。

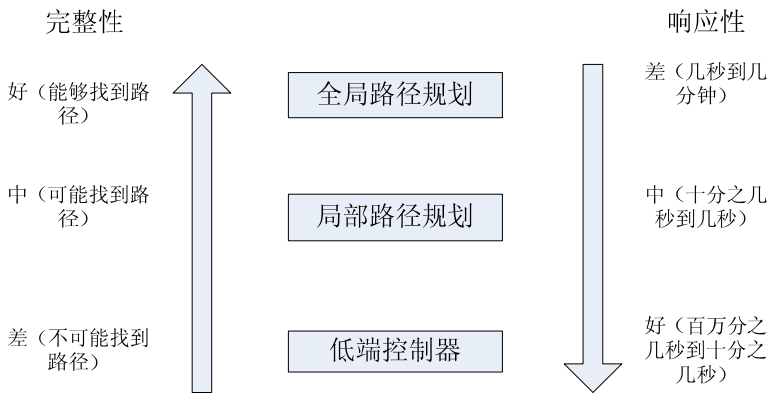

图 1.1 全局法与局部法的关系

局部法比全局法有更快快的运行速度，它能够发现并且回应全局路径规划法不能预见的事件。这使得混合方法比单一的全局方法具有更强的鲁棒性，然而它保持着全局路径规划法很多温和的属性。混合方法的示例见[11,12,30]。

1.3.2 移动机器人的全局路径规划法

这种路径规划方法的过程可以分为三个步骤，这三个步骤构成了所有全局路径规划法有效运行的基础。具体过程见表 1.1。

表 1.1 全局路径规划法的步骤

（1）	利用相关环境建模技术划分环境空间
（2）	形成包含环境空间信息的搜索空间
（3）	在搜索空间上应用各种搜索策略进行搜索

常见的搜索算法有：A*算法[13]、D*算法[14]、遗传算法[15,16]等。

全局路径规划法主要包括：栅格法、可视图法、拓扑法与自由空间法等。

(1)栅格法

受到网格思想的启发，科学家们有意的将机器人的运动空间划分为很多的栅格。然后利用规划算法来寻找路径。这种方法称为栅格法[14]。

栅格的划分是为了存贮机器人周围的环境信息。如果环境中存在障碍物，那么表示障碍物的地方的栅格就被划分的比较密集。而其他的地方就被划分的相对稀疏。这样，机器人就可以按照环境中栅格的疏密程度不同而做出判断是否有障碍物。主动避开栅格密集区从而达到避障的目的。

栅格的划分大小非常关键，它决定着算法的效率。如果分得过大，那么算法效率会很高但是功能会受到影响。因为机器人没有足够的信息来发现路径；如果分得太小的话，算法的功能会有所提高但规划效率会大大下降[17]。在当今的机器人路径规划领域中栅格法以及它的改进算法已经得到了广泛的应用[18]。

(2)可视图法

这种算法非常简单，它将移动机器人用一个点来表示。结下来用直线连接机器人与各个障碍物的各个顶点，然后连接目标点与障碍物的各个顶点。前提是所有的直线都不得穿越障碍物。线的连接构成了图，这个图就叫做可视图[14]。

因为我们可以观察到每条直线的顶点，并且所有的路径都是可行路径。这样，就产生了一个路径规划的优化问题。也就是要得出全部的与起始点和目标点相连直线距离的最小值。然后再优化算法的帮助下可以计算出距离最短的路径。

该算法的缺点是将机器人看作一个点，这样就会产生机器人距离障碍物太近的问题。有时该方法的运行效率也不高，缺乏灵活性。

(3)拓扑法

拓扑法[15]的主要思想是建立网络，然后在网络上搜索路径，这一点与可视图法类似。首先将机器人的运行环境划分为若干的子空间，这些子空间具有拓扑属性。这样就将原来的高维空间降低为低维的拓扑空间。这缩小了搜索的复杂度。只有障碍物的个数的增加才能增加拓扑法的复杂度。拓扑法的误差小，因为它不需要精确的机器人坐标[16]。

拓扑法的缺点是网络的建立非常复杂，更复杂的是随着环境的改变如何修正原来的拓扑网络。

(4)自由空间法

自由空间法[14]应用设置好的图形来建立自由空间，如凸多边形和锥形。然后将其用联通图表示，接下来搜索联通图寻找最优路径。

它的构造方式[19]为：以障碍物的某一定点出发作与其他顶点的连线。确保每条连接

线与障碍物的边线围成的凸多边形面积最大；机器人的可行路径极为每条连接线的中点连线。自由空间法的长处在于联通图不受起始点与目标点位置变化的影响。自由空间法的短处在于障碍物越多，算法复杂度越高，甚至失效。另外，自由空间法会导致路径的不确定[20]。

1.3.3 移动机器人的局部路径规划法

局部路径规划算法有很多，具有代表性的主要有：人工势场法(artificial potential field)、模糊逻辑(fuzzy logic)、神经网络法(neuro network)、遗传算法(genetic algorithm)等等。

(1)人工势场法

人工势场法[l4,22]的提出者叫做 Khatib，其精髓是把移动机器人的运动环境比作是受力场。机器人在该受力场中主要收到两个力的作用，一个是障碍物对机器人的斥力；一个是目标点对机器人的吸引力。在这两个力的合理的作用下机器人进行局部避障。

人工势场法的优点在于结构简单，方便人们进行控制。其提供的轨迹平滑，应用广泛。人工势场法的缺点在于容易产生局部极小而是机器人产生卡死。机器人在局部极小点停留、徘徊不前而不是向目标点运动。这使得人工势场法很难应用于复杂环境。人工势场法的改进算法有很多[23]，主要是通过改进势函数来摆脱局部极小所带来的问题。但是效果并不明显。

(2)模糊逻辑算法

模糊逻辑算法[10,24,26]起源于机动车司机的驾驶实践。司机对车辆的驾控不是基于对周围信息的准确判断。这一点也很难实现，因为周围的环境信息具有模糊性。该算法正是利用该模糊性，继而完成规划过程。模糊逻辑算法的优点在于没有局部极小的情况发生。适用于未知的环境。适合于信息供应并不是十分充足与确定的环境中。模糊逻辑算法的缺点是准确性差，不能提供最优路径。而且随着控制量的增加，算法的复杂度不断增大[27]。

(3)神经网络法

神经网络法[21,28,29]可以弥补模糊逻辑算法的不足，并且可以通过与模糊逻辑算法的结合来弥补后者的缺陷[31]。它浅显直观，主要思想就是该算法将传感器传入的数据网络化，按照预先给定的转角增量作为该网络的输出。首先选定一个初始样本集，这个样本集由许多预先设定的运动姿态的数据构成。然后对该样本集进行筛选得到标准样本集。机器人通过对外界信息按模糊规则实施处置，然后对信息进行汇总并归纳出模糊制度，然后

对标准样本集实施该模糊制度。通过不断的学习新的信息，机器人逐渐变得智能化。此外神经网络法与模糊逻辑算法的结合在动态空间中也有着很好的应用[32]。

(4)遗传算法

该算法模仿生物体在繁殖后代中所体现出的种种遗传变异的现象为根本，是一种仿生算法[33]。将自然界中的选择、交叉和变异等理论应用于计算机算法，具有很好的表现与前景。

遗传算法的主要理论是[34,35]：第一步将路径点看成一个种群并用二进制的字符串来表示。第二步是对这个种群进行选择、交叉于变异以提高种群的表现。最后形成最优的种群。遗传算法的优点在于它更容易的到全局最优解，因为区别于其他单点搜索算法，遗传算法采用多点搜索，这使得该算法不容易陷入局部最优解。即可以摆脱人工市场法的不足，十分适合于未知条件下的机器人路径规划。遗传算法的缺点在于速度慢，对计算机的需求非常高，这也是限制其应用的主要因素。遗传算法的改进方法有很多，许多学者通过简化路径号码并总结出不同的适应度函数[36]。

1.4 快速扩展随机树算法研究综述

在详细分析了以上全局与局部方法优缺点的基础上，科学家们在仿生算法领域找到了新的灵感。以往的路径规划算法在处理简单的规划问题时优势很明显，也得到了很好的应用。但是碰到另一类问题，即分完整约束问题时就显得捉襟见肘。而非完整约束下的路径规划又是路径规划问题的非常重要的组成部分，以往的方法对于环境的完整性与准确性要求非常高，如果机器人的自由度增加，那么算法的复杂度也呈指数增长。所以以往的算法在处理此类问题时有很强的局限性。

RRT[44-46]算法的全称是(Rapidly-exploring Random Tree)，翻译成中文为快速扩展随机树算法。它是今年来发展起来的，并且应用比普遍。该算法属于随机采样类型，具有搜索未知区域的功能。重点是，RRT 算法在处理分完整路径规划问题有着相当大的优势，因为它可以将各种约束集成到算法本身之中，因此对环境的要求就比较低。该算法效率高，速度快。可以直接在分完整系统中加以应用。随机性事该算法的固有属性，所以该算法概率完整。也就是说理论上肯定能够找到可行路径。

快速扩展随机树算法是一种可以将机器人的动力学约束考虑其中的运动规划方法。和其它的随机采样的算法一样，RRT 算法搜索空间的速度远远快于完整搜索算法。RRT 算法提供的路径并不是最优的，但是它在大多数实例中被证明是非常有效的。Tan 将该

方法应用在无人机的运动规划中[62]，而 Frazzoli 则把该算法应用于宇宙飞船的运动规划[63]。快速扩展随机树算法(RRT)就像它的名字所形容的那样，能够在执行算法过程中生成一个树状结构。这个树状结构的起点就是我们的机器人的起始位置，然后经过迭代扩展到机器人构型空间中的其它部分。RRT 算法的一个重要属性就是它不但能规划出机器人的坐标点，而且还包括首向和速度。如果一条路径是由考虑这些约束的 RRT 算法生成的，那么我们就知道那是一条满足约束的可行路径。

1.5 本文的主要研究内容

论文主要针对移动机器人运动规划的相关算法进行研究，主要内容有：

第一章绪论。这一章的主要描述的对象为：叙述了机器人技术的产生与发展；阐述了移动机器人研究的意义；将移动机器人运动规划的研究进行了概述；例举并分析了已有典型算法的特点。

第二章为建模部分，从运动学角度对机器人进行探讨。主体是它的车轮描述及运动学约束。同时对仿真环境进行建模和运动障碍物建模，运动障碍物建模包括匀速运动模型、随机运动模型。

第三章研究了快速扩展随机树算法以及其改进型。实现了以上两种算法的路径规划器。这一章的主要内容为：对基本的 RRT 算法进行了描述，包括算法的发展现状、具体的扩展方式与行为特性；针对基本 RRT 算法的工作效率低下，搜索方式过于平均与随机的缺点进行算法改进。提出了基于启发式函数的基本 RRT 算法的改进算法 Bias-goal RRT 算法；对 Bias-goal RRT 算法进行了详细的描述，通过计算机仿真实验将其与基本 RRT 算法进行对比分析。最后总结出 Bias-goal RRT 算法的优势所在。

第四章介绍选择性快速随机搜索树算法(Bias-goal RRT)在动态环境下的路径规划。首先对 BIAS-RRT 的直线扩展过程做适当调整，使改进的 RRT 算法(Bias-goal RRT)后能满足移动机器人的滚动约束，并给出动态环境下非完整系统中 Bias-goal RRT 的路径规划仿真结果。

第五章研究了 Bias-goal RRT 算法与人工势场法的结合应用。该章主要包含的内容有：简单介绍了人工势场法的基本原理和基本步骤；分析人工势场法的主要缺陷和一般的解决方案；提出 Bias-goal RRT 与势场法相结合的方法来解决势场法的缺陷并给出试验结果验证结合算法的有效性。

第 2 章　轮式移动机器人与环境建模

2.1　引言

轮式移动机器人的路径规划法是本文研究的主题，所以我们首先介绍移动机器人的运动学模型与相关的环境建模。

2.2　轮式移动机器人的运动学模型

无论是前轮驱动还是后轮驱动的轮式机器人，都必须要符合非完整约束条件[37-38]。如图 2.1 所示：

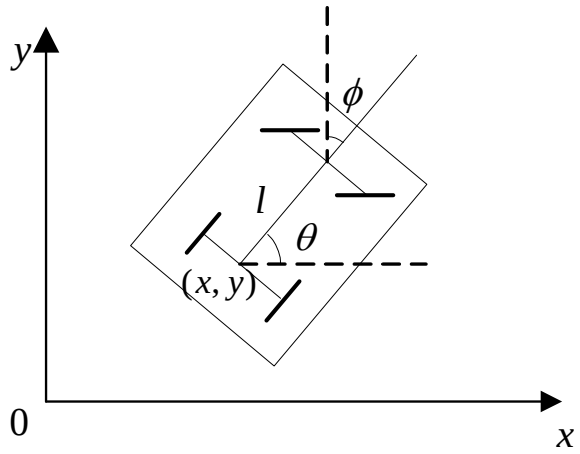

图 2.1 移动机器人运动学模型图

车的前轴和后轴上各有两个轮子。车的中心轴的中点是机器人所在的当前位置，前轮具有转弯功能，后轮方向固定。该机器人系统的广义坐标定义为 $q=(x,y,\theta,\phi)$。后轮的笛卡尔坐标为 (x,y)。θ 为机器人相对于 x 坐标轴的方向角。ϕ 为机器人的导航角。其中：

前轮、后轮受到的约束分别为：

$$\dot{x}_f \sin(\theta+\phi)-\dot{y}_f \cos(\theta+\phi)=0 \tag{2-1}$$

$$\dot{x}\sin\theta-\dot{y}\cos\theta=0 \tag{2-2}$$

式中：x_f ——轮式移动机器人的前轮的横坐标；

y_f ——轮式移动机器人的前轮的纵坐标；

该坐标具有刚体约束：

$$x_f = x + l\cos\theta \tag{2-3}$$

$$y_f = y + l\sin\theta \tag{2-3}$$

式中：l——为车前后轮之间的距离；

那么第一个运动学上的约束为：

$$\dot{x}_f \sin(\theta+\phi) - \dot{y}_f \cos(\theta+\phi) - \dot{\theta}\cos\phi = 0 \tag{2-4}$$

其中 Pfaffian 约束矩阵为：

$$C(q) = \begin{bmatrix} \sin(\theta+\phi) & -\cos(\theta+\phi) & -l\cos\phi & 0 \\ \sin\theta & -\cos\theta & 0 & 0 \end{bmatrix} \tag{2-5}$$

该矩阵的秩恒为 2。

若机器人为后轮驱动，系统受到的约束为：

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \tan\phi / l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{2-6}$$

式中：v_1——车的驱动速度输入；

v_2——为导航角速度输入；

该模型在$\phi = \pm\pi/2$处奇异，那是因为它的第一个矢量部分并不连续。而实际情况是当机器人前轮法线与纵轴垂直的时候，机器人就会卡死。因此，导航角需要控制在$\pm\pi/2$之内，这样就不会发生奇异。

驱动轮为前轮的机器人所受的约束为：

$$\begin{bmatrix} \dot{x} \\ \dot{y} \\ \dot{\theta} \\ \dot{\phi} \end{bmatrix} = \begin{bmatrix} \cos\theta\cos\phi \\ \sin\theta\cos\phi \\ \sin\phi / l \\ 0 \end{bmatrix} v_1 + \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} v_2 \tag{2-7}$$

或者表示成：

$$\begin{cases} dx\sin\theta - dy\cos\theta = 0 \\ \dfrac{dx}{dt} = v_1\cos\phi\cos\theta \\ \dfrac{dy}{dt} = v_1\cos\phi\sin\theta \\ \dfrac{d\theta}{dt} = (v_1/l)\sin\phi \\ \dfrac{d\phi}{dt} = v_2 \end{cases} \tag{2-8}$$

机器人前轮的驱动速度是v_1。此时即使$\phi = \pm\pi/2$时该机器人系统也不会发生奇异。真实世界中的轮式机器人一般采用前轮驱动，故本文主要研究的是前轮驱动的机器人。上文中轮式机器人的运动学约束是非完整约束的一种，称为滚动约束。

2.3 环境建模

在一个移动机器人运动规划的问题的研究中，核心目标是想让机器人从起始点顺利的绕过静止的或运动的障碍物并运动到目标点。为了实现这一目标，第一步要对这个运动规划问题进行严谨的、系统的与专业的描述。要能够科学的定义与描述在运动规划中出现的如机器人、起始点、目标点、静止的或移动的障碍物等对象。以上就是环境建模的概念。

在环境建模的过程中，路径规划问题描述转变成具体的数学求解或物理求解。例如在某一个环境中，机器人的位姿状态可以用(x, y, θ)的形式来表达出来。但是在没有规范的环境建模的理论指导下，人们对这一形式的理解与表达是多种多样的。所以，科学的，系统的对运动规划中各个对象的表达是至关重要的。环境建模的意义就在于统一观念，让人们更科学的认识与分析运动规划问题。从这个角度来说，环境建模对路径规划问题进行了规范表达。与此同时将数学方法应用于路径规划问题使得问题更容易理解。本文将路径规划的算法以仿真的形式表达出来。而仿真环境也将通过环境建模以对象的形式在仿真中演示出来。

本文的主要工作是要在仿真系统中表示出障碍物与非障碍物区域，演示算法的求解过程。

使用像素表示地图中的各种目标，每个像素为一个单位。暂且规定白色为自由空间，其他颜色为障碍物空间。

在含有多种约束的环境中，像素还可以包含机器人的诸多运动学约束。将像素确定

为地图的最小单位的理由很多，见表 2.1

表 2.1 选择像素用于环境建模的原因

(1)	像素简单明了，可以降低数据结构的复杂度。
(2)	像素可以兼容栅格法，能够得到普遍应用。
(3)	像素对自由空间和障碍物空间的划分使搜索效率提高。

2.4 移动障碍物建模

移动障碍物需要考虑时间轴，这是移动障碍物与静态障碍物不同的地方。机器人要能够正确的判断移动障碍物的坐标及其运动趋势以完成避障。判断的重点即运动模型的使用。下文主要研究了若干不同的运动模型。

2.4.1 匀速障碍物模型及其预测

若障碍物无加速度，速度是一个常数的话：

$$\begin{cases} a(t) = 0 \\ v(t) = c \end{cases} \tag{2-9}$$

则对于障碍 $p(x, y, t+\Delta t) = p(x, y, t) + ct$ 其中 c 为常量。那么它的位置坐标应该成线性变化。

$$\begin{cases} x = at + b \\ y = ct + d \end{cases} \tag{2-10}$$

这个模型是障碍物发生的全部运动的根本。在某一个间隔 $[t, t+\Delta t]$ 中，可以大致的认为障碍物是以匀速的状态运动的。而对这种模型而言，基于最小二乘法的线性预测法是最主要的预测方法。

2.4.2 随机障碍物模型及其预测

随机运动模型体现了障碍物状态的改变，我们可以用概率分布函数来估计这种状态[41]。

$$a(t) = \beta w(t) \tag{2-11}$$

其中 $w(t)$ 为随机向量，$a(t)$ 为加速度，很多时候我们无法掌握其概率的具体分布情况。因此适当的假设是比不可少的，可以举典型的分布函数为例，如高斯分布与联合分布。

所以我们可以假设其为高斯分布函数亦或是联合分布函数，那么，分布函数可以如

下表达：

$$prob(w(t)) = f_p(\mu(w(t)), \sigma^2(w(t)) \tag{2-12}$$

式中：$\mu(w(t))$——平均值向量；

$\sigma^2(w(t))$——方差向量；

那么障碍物的速度就可以得出

$$v(t) = v(t-\Delta t) + \int_{-\Delta t} w(t)d(t) \tag{2-13}$$

随机运动模型也可以以如下形式描述：

$$a(t) = \alpha a(t-\Delta t) + \beta w(t) \tag{2-14}$$

式中：$w(t)$——随机向量；

α——原有加速度的权值；

β——随机向量 $w(t)$ 的权值；

因此可以得出：

$$v(t) = v(t-\Delta t) + \int_{-\Delta t} (\alpha a(t-\Delta t) + \beta w(t))d(t) \tag{2-15}$$

2.5 机器人对障碍物运动的预测

移动机器人能够学习障碍物自身的信息来判断障碍物的移动规律。本节分析几种方法来感知障碍物的移动规律。

匀速直线运动时所有运动的基础。最小二乘法是最常用的方法，准确与简单是最小二乘法的特性。实时最小方差预测算法(LMSE)[42]是本文的选用方法。设 $p(x, y)$ 为机器人的位置。

设 x， y 与时间 t 成线性关系，则有

$$\begin{cases} x = at + b \\ y = ct + d \end{cases} \tag{2-16}$$

式中：a, b, c, d ——未知待估参数。

通过 n 次观察之后得出：

$$\begin{cases} x_i = at_i + b \\ y_i = ct_i + d \end{cases} \quad i = (1, 2 \ldots n) \tag{2-17}$$

把上式用矩阵形式描述：

$$\begin{cases} x_n = T_n A_n \\ y_n = T_n B_n \end{cases} \tag{2-18}$$

其中：

$$x_m = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} \quad y_m = \begin{bmatrix} y_1 \\ y_2 \\ y_3 \end{bmatrix} \quad T_m = \begin{bmatrix} t_1 \\ t_2 \\ t_3 \end{bmatrix} \quad A_m = \begin{bmatrix} a \\ b \end{bmatrix} \quad B_m = \begin{bmatrix} c \\ d \end{bmatrix}$$

待估参数 a，b 与 c，d 的推导过程相近，所以我们之推导 a，b 的求解过程。我们先定义一个误差向量：

$$E_n = (e_1, e_2 \ldots\ldots e_n)^T \tag{2-19}$$

推出：

$$E = X_n - T_n A_n^{'}$$

式中：$A_n^{'}$——估计解。

定义误差方程：

$$J_n \sum_{i=1}^{n} \lambda^{n-i} e_i^2 \qquad 0 \le \lambda \le 1 \tag{2-20}$$

定义 ： $p_{n-1} = \frac{1}{\lambda}(T_{n-1}^T T_{n-1})^{-1}$

由系统辨识理论可以得出：

$$\begin{cases} \hat{A}_n = \hat{A}_{(n-1)} + \gamma_n p_{n-1} t_n \{x_n - t_n^T \hat{A}_{(n-1)}\} \\ p_n = \frac{1}{\lambda}\{p_{n-1} - \gamma_n p_{n-1} t_n t_n^T p_{n-1}\} \end{cases} \tag{2-21}$$

其中：

$$t_m = \begin{bmatrix} t_m \\ 1 \end{bmatrix} \qquad \gamma_m = \frac{1}{1 + t_m^T p_{m-1} t_m}$$

机器人通过自身传感器得到 x_n，求解(2-21) $\hat{A}_n$ 可以被解出。$\hat{A}_n$ 为最优估计量。机器人通过感知障碍物的运动趋势完成对障碍物运动的预测。为避障打下基础。

2.6 本章小结

本章简要地介绍了移动机器人运动学模型以及环境建模作用与意义。并提出本文仿

真系统中环境表示的方法。然后对移动障碍物进行建模，包括匀速运动模型与随机运动模型，并分别给出了两种模型的预测方法。最后给出了移动机器人对障碍物预测的方法，为后文的仿真试验打下了理论基础。

第 3 章　快速扩展随机树算法及其改进型的比较研究

3.1　引言

上一章主要对移动机器人进行了运动学建模与相关的环境建模，本章主要针对快速扩展随机树算法及其改进型进行比较研究。

3.2　RRT 算法的流程与实现

最近十几年来，基于随机采样的路径规划算法渐渐增多，该类算法不但适用于不确定环境的高维空间，而且能够很好的解决如人工势场法的局部极小问题。

快速扩展随机树算法作为一种高效的数据结构和算法近年来得到了广泛的应用，该算法适用于解决未知环境下的完整性规划、以及涉及动力运动学约束的非完整性规划。

3.2.1　基本 RRT 的扩展方式

RRT 算法是以种随机算法[47]，并不需要特定的启发式函数的帮助。该算法对未知空间有着强烈的搜索倾向[48]。这是 RRT 算法的一个非常显著的优点。

该算法的扩展方式非常特殊，这中扩展方式能够渐渐的较少树节点与目标点之间的距离[49-50]。基本 RRT 的运行过程如下图所示：

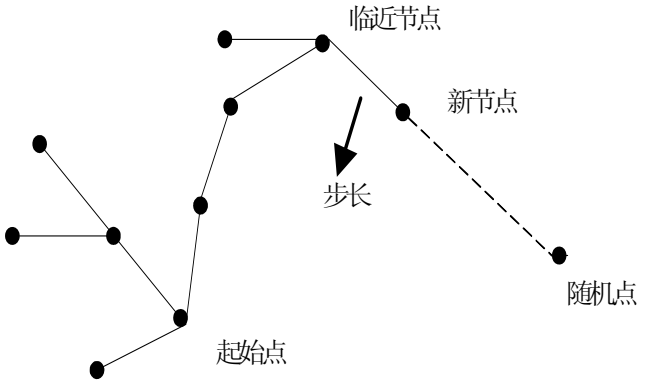

图 3.1 快速扩展随机树的扩展过程

在一个普通的路径规划问题中，其任务就是要在态空间Q中寻找一条从q_{start}到q_{goal}的连续路径。这其中我们要保证机器人在Q_{free}，即自由状态区域中运动，同时要绕过障碍物区域$Q_{obst} \subset Q$。我们可以将问题做如下描述：

表 3.1 路径规划问题描述

q_{start}	起始点即初始状态
q_{goal}	终点即目标状态
Q	状态空间
$Q_{goal} \subset Q$	目标区域
$Q_{obst} \subset Q$	障碍物区域
Q_{free}	自由状态区域

RRT 算法如图 3.1 所示。

$GENERATE\_RRT(q_{start}, K, \Delta t)$

1 $T.init(q_{start})$;

2 for k =1 to K do

3 $q_{rand} \leftarrow Random\_Configuration()$;

4 $q_{near} \leftarrow Nearest\_Neighbor(q_{rand}, T)$;

5 $u \leftarrow Select\_Input(q_{near}, q_{rand})$;

6 $q_{new} \leftarrow New\_Configuration(q_{near}, u, \Delta t)$;

7 $T.add\_vertex(q_{new})$;

8 $T.add\_edge(q_{near}, q_{new}, u)$;

9 return T

图 3.1 快速扩展随机树算法

下图所示为二维空间中在完整约束条件下快速扩展随机树的扩展过程。图中红色点为起始点，无目标点与障碍物。蓝色线条表示随机树的扩展结构。可以观察到随机树从起始点向四面八方扩展，最终布满整个二维空间。

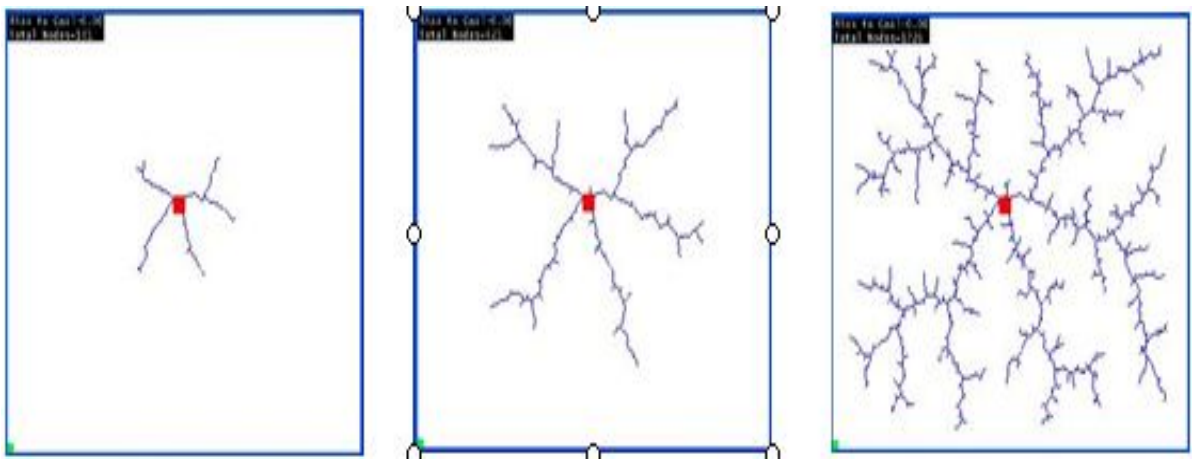

图 3.2 2D 空间中对完整约束规划问题 RRT 的扩展过程

3.2.2 RRT 算法流程

RRT 算法流程与 3.2 节叙述的大致相同，然而简单起见，我们用更加简洁的方式表示该流程。如表 3.2 所示：

表 3.2 构型空间中 RRT 算法各参数的意义

C	所有空间
C_{free}	无障碍空间
q_{start}	起始点
q_{goal}	目标点
ε	步长
$Dis(x_1, x_2)$	C 中任意两点间的距离
T_k	随机树有 k 个节点

RRT 的构建过程见表 3.3：

表 3.3 RRT 算法的构建过程

(1)	给出 T_{init} (即 q_{start})。
(2)	选择随机点 q_{rand}， $q_{rand} \in C_{free}$。
(3)	找出距离 q_{rand} 最近的节点 q_{near}。即 $Dis(q_{near}, q_{rand}) \le Dis(q, q_{rand})$。
(4)	在 q_{near} 与 q_{rand} 的连线上求 q_{new}， q_{new} 必须满足 $q_{new} \in C_{free}$。且 $Dis(q_{new}, q_{near}) = \varepsilon$ 的条件。如果存在 q_{new}，且 q_{new} 可避障，转到步骤 5，否则转到步骤 7。
(5)	在 T_k 上增加一个新节点。令 T_{k+1} 表示新的 RRT，则 $T_{k+1} = T_k + q_{new}$。
(6)	判断条件 $q_{goal} = q_{new}$，若是，转到步骤 7，否则转到步骤 2。
(7)	结束。

具体的随机树的流程图如图 3.3 所示，该图从另一个角度更加直观的表示出了随机树的具体扩展流程。

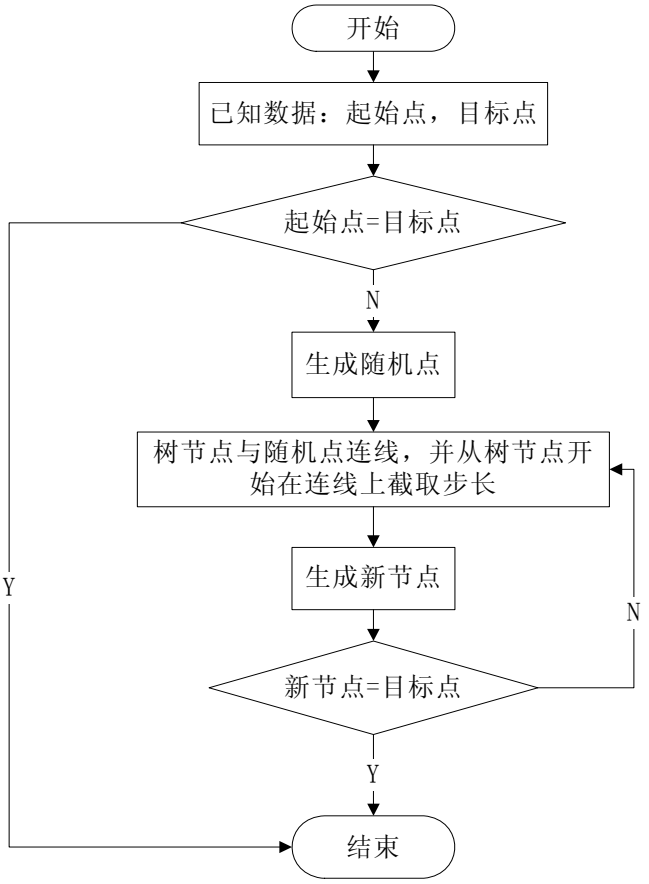

图 3.3 基本 RRT 算法流程图

找出T_k中距离q_{rand}最近的节点q_{near}的具体过程在 Nearest-Neighbor()函数中实现；它的伪代码描述如下所示：

$NEAREST\_NEIGHBOR(T, q_{rand})$

1 $d_{\min} \leftarrow Max;$

2 for $Iter = T.begin$ to $T.end$

3 $d \leftarrow Dis(Iter, q_{rand})$;

4 if $d \le d_{\min}$

5 $q_{near} \leftarrow Iter$;

6 $d_{\min} \leftarrow d$;

7 return q_{near};

New-Configuration()函数如下所示：

$NEW\_CONFIGURATION(q_{near}, q_{rand})$

1 $y_1 \leftarrow y_{x_{near}}$;

2 $x_1 \leftarrow x_{x_{near}}$;

3 $x_2 \leftarrow x_{x_{rand}}$;

4 $y_2 \leftarrow y_{x_{rand}}$;

5 if $STEP \geq Dis(q_{near}, q_{rand})$

6 $q_{new} \leftarrow q_{rand}$;

7 return q_{new};

8 $x_{q_{new}} \leftarrow x_1 + \frac{STEP \cdot (x_1 - x_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$;

9 $y_{q_{new}} \leftarrow y_1 + \frac{STEP \cdot (y_1 - y_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$;

10 if $Dis(q_{rand}, q_{near}) \leq Dis(q_{rand}, q_{new})$

11 return q_{new};

12 else

13 $x_{q_{new}} \leftarrow x_1 - \frac{STEP \cdot (x_1 - x_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$;

14 $y_{q_{new}} \leftarrow y_1 - \frac{STEP \cdot (y_1 - y_2)}{\sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}}$;

15 return q_{new};

树 T 与 q_{rand} 为两个传递到 Nearest-Neighbor 的参数，并返回 q_{near}， q_{near} 为树 T 中距离随机状态点 q_{rand} 最近的点。具体方法是要多次算出树节点与 q_{rand} 的距离并返回 q_{near}。

q_{near} 与 q_{rand} 为传递到 New-Configuration 的参数并返回 q_{new}。 q_{new} 表示新的树节点。具体的方法是：连接 q_{near} 与 q_{rand} 两个点并连接，然后再在连线上截取距离 q_{near} 长度为步长的坐标点。该坐标点即为 q_{new} 的坐标点。

实现过程如下：如步骤 1-4 所示，分别取 q_{near} 与 q_{rand} 的坐标。然后进入步骤 5-7，即如果两者的距离小于步长，则 q_{rand} 即为新的 q_{new}。如果距离大于步长，则如步骤 8-9 算出 q_{new} 的坐标。然后根据步骤 10 判断 q_{new} 是在 q_{near} 与 q_{rand} 的连线上还是反向延长线上。若在连线上则根据步骤 11 返回 q_{new}；如果在反向延长线上则根据步骤 12-15 计算出 q_{new} 的坐标。

3.2.3 算法实现

本章使用 OpenGL 技术[51]作为 RRT 算法的仿真技术。OpenGL 技术应用广泛、使用方便并且易于开发与修改。用该技术作为图形实现工具优点非常明显，该技术的特点如下：

(1)画面质量高

(2)具有很高的稳定性

(3)安全可靠

(4)便于使用

3.1 节描述了基本 RRT 算法的工作流程。为了实现这一流程，本节应用 VC++6.0 作为编程工具。中间采用 OpenGL 技术用于图形的表达，使得图形界面的实现更加简单。

3.3 Bias-goal RRT 算法的提出与实现

即便 RRT 算法由很多优点，如速度快、不会出现卡死以及概率完备等等。但该算法的固有规划方式限制了其进一步应用，需要进行改进以提高算法性能。基本 RRT 算法的缺陷见表 3.4：

表 3.4 基本 RRT 算法的不足

(1)	扩展方式过于平均
(2)	算法实时性很差
(3)	路径的质量不高

针对以上的缺陷，我们要改进基本的 RRT 算法以提高算法的效率。针对扩展方式过于平均这一缺点，我们可以从启发式算法的思想中汲取灵感以改进原有的算法。即改进的算法不是单一的搜索全部的空间，而是有一个搜索的启发式策略。这个策略就是使树的扩展有一个趋于目标点的趋势。这样就会提高搜索的时效，同时，理论上讲，这样的改进也会相应提高路径的质量。

基于以上想法，本文提出了基本 RRT 的改进算法，即有偏向目标属性的(Bias-goal RRT)算法。

3.3.1 Bias-goal RRT 算法

偏向目标型快速扩展随机树算法(Bias-goal RRT)提出的原因是基本RRT存在着上文提到的三种主要缺陷。在启发式思想的影响下，我们使该改进算法具有偏向目标搜索的趋势从而提高算法效率与路径质量。

如上文提到的，RRT 算法在扩展的时候要随机的吸纳一个点作为一个下一步扩展中转站。我们称这个随机点为 target。也可以称该 target 点为局部目标点。搜索树结点的分布状况取决于 target 的产生方式。

基本 RRT 的缺点是在选取 target 点的时候过于随机。这导致了对周围的环境的搜索

过分的平均。这时基本 RRT 算法效率低下的根本原因。

由于基本 RRT 算法没有启发性，所以在该算法工作的过程中时间消耗很长。本文改进的重点在于 target 点的产生方式。为避免搜索过于平均，已有学者提出改进 target 点产生方式的方法[52-53]。本文利用参数 Bias-goal 来完成 target 产生方式的改变。

3.3.2 Bias-goal RRT 算法实现

Bias-goal RRT 的算法流程如下：

$$BIAS\_RRT(q_{start}, K, \Delta t)$$

1 $T_{init}(q_{start})$;

2 for $k=1$ to $K < MAX\_TREESIZE$ do

3 $x_{target} \leftarrow BIAS\_Configuration(q_{goal}, BIAS)$

4 $q_{near} \leftarrow Nearest\_Neighbor(q_{target}, T)$;

5 $u \leftarrow Select\_Input(q_{near}, q_{target})$;

6 $q_{new} \leftarrow New\_Configuration(q_{near}, u, \Delta t)$;

7 $T_{add}\_vertex(q_{new})$;

8 $T_{add}\_edge(q_{near}, q_{new}, u)$;

9 return T

该算法和基本 RRT 算法大体类似，但它对基本 RRT 算法作了修改，在 Bias-goal RRT 里，q_{target} 为过程中目标点，q_{target} 的构造方式影响着 RRT 的形状。Bias-goal RRT 算法是通过随机函数加参数 Bias-goal 来产生局部目标点，这样就有能更有效率的找到目标点。

确定基点 q_{target} 的算法如下：

$$BIAS\_GOAL(q_{goal}, BIAS)$$

If $Math\_Random() < BIAS$

$$q_{target} = q_{goal}$$

else $q_{target} = q_{rand}$

具体方法是：选取参数 Bias-goal，该值的选取是为了表示算法认定 q_{goal} 为 q_{target} 的几率。判定方法是：如果 Bias-goal 的值大于算法产生的随机值的时候就选取目标点为 q_{target}。如果 Bias-goal 的值小于算法产生的随机值就选取一个随机点当作 q_{target}。

如此的 target 点产生方式使改进算法于基本 RRT 算法相比较有了偏向目标的趋势，但这种趋势并不影响其概率完整性，也就是说新的改进算法继承了基本 RRT 算法的全部优点，同时，收敛速度快，路径的质量也会有很大的提高。

具体 Bias-goal RRT 算法的流程图如图 3.4 所示：

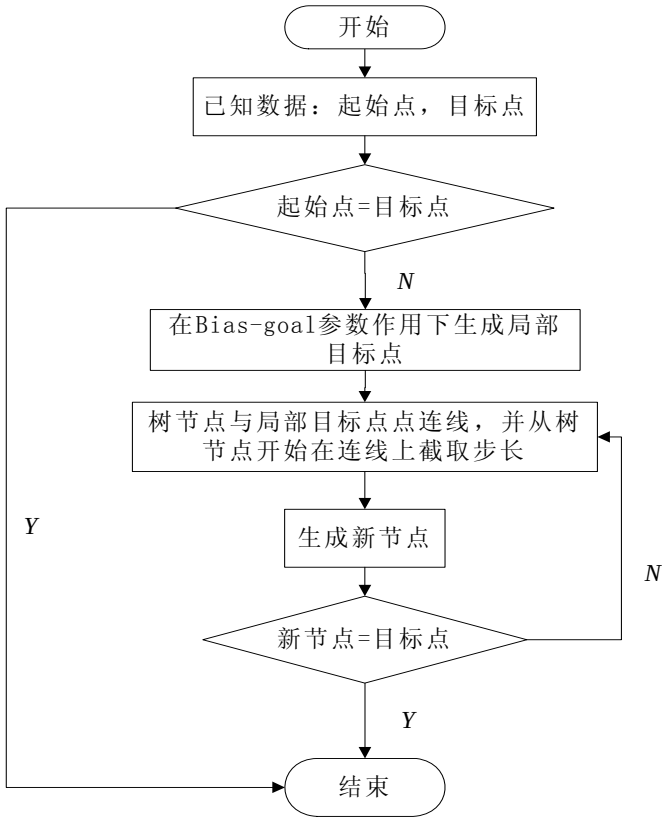

图 3.4 Bias-goal RRT 算法流程图

3.4 仿真试验与结果分析

这一节主要针对上文中提到的两种算发进行计算机仿真，然后通过仿真结果的比较，真实、直观、明了的对比出基本 RRT 算法与其改进型 Bias-goal RRT 算法的优缺点。

3.4.1 不同障碍物分布下的仿真实验

进行计算机仿真的电脑具体参数为见表 3.5：

表 3.5 仿真实验中电脑参数

处理器	Intel(R) Pentium(R) 4 CPU 3.00GHz
内存	512MB
磁盘驱动器	500GB
操作系统	WindowsXP ServerSP4
显卡	Intel(R) 82945G Express Chipset Family

本实验一共做了两组，目的在于比较基本 RRT 算法与 Bias-goal RRT 算法的表现，选择了四组典型的地图环境。具体的地图尺寸是 480×413 的环境下步长为 2，仿真如图 3.5 所示：

图 3.5 所示的图展现了基本 RRT 的生长过程与其改进型 Bias-goal RRT 的扩展对比试验。使用地图一，即随机分布障碍物。其中绿色点为目标点，红色点为起始点。黑色区域为障碍物，白色区域为自由区域，蓝色线条即为 RRT 算法的扩展过程，红色线条为最终可行路径。左图表示的是基本 RRT 的扩展过程，扩展时间为 4.625 秒，扩展节点数为 800，右图表示的是 Bias-RRT 的扩展过程，扩展时间为 2.643 秒，扩展节点数为 111。

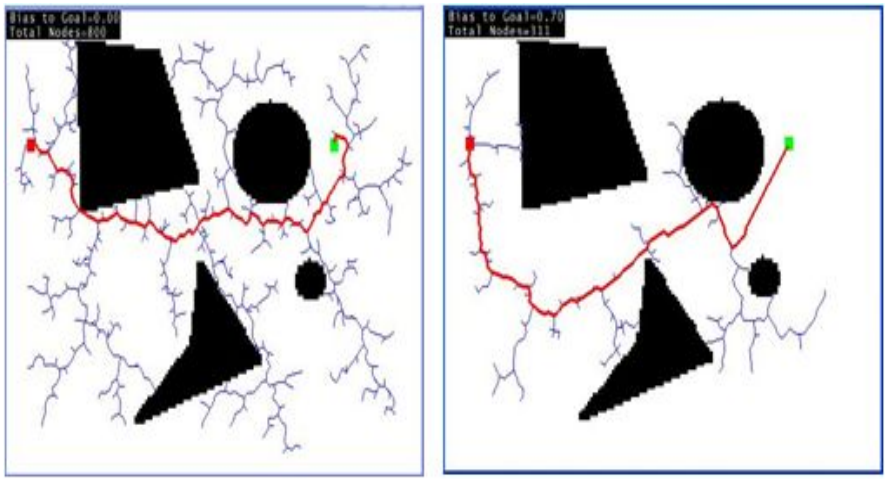

图 3.5 分布障碍类型

图 3.6 所示仿真实验使用地图二，即迷宫类型。设置步长为 2，其中基本 RRT 耗时 26.623 秒，扩展节点数为 4782。Bias-RRT 耗时 10.465 秒，扩展节点数为 1225。

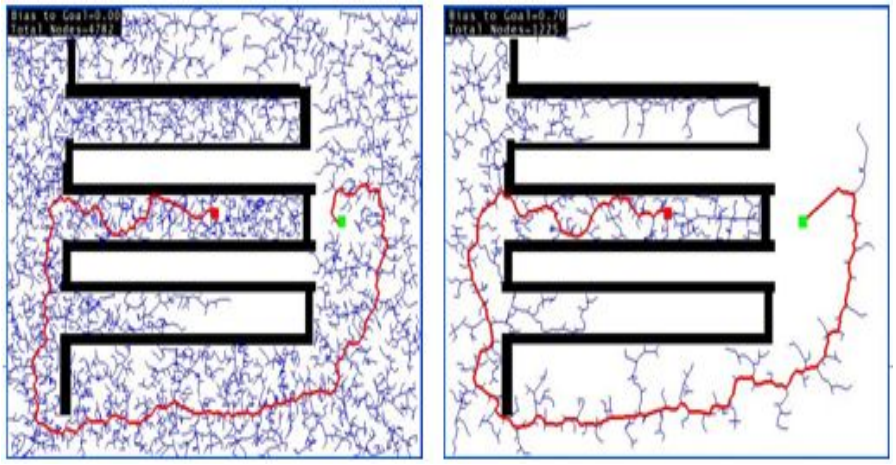

图 3.6 迷宫类型

图 3.7 所示仿真实验使用地图三，即混杂障碍类型。设置步长为 2 做的仿真实验。其中基本 RRT 耗时 10.654 秒，扩展节点数为 2875。Bias-RRT 耗时 3.765 秒，扩展节点数为 971。

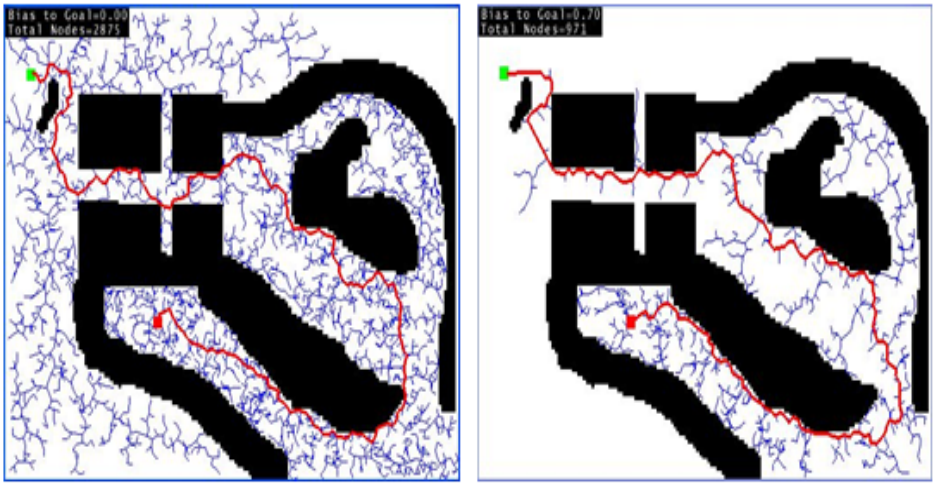

图 3.7 混杂障碍类型

图 3.8 所示仿真实验使用地图四，即狭窄通道类型。设置步长为 2 所做的仿真实验。其中基本 RRT 耗时 9.454 秒，扩展节点数为 2593。Bias-RRT 耗时 2.361 秒，扩展节点数为 479。

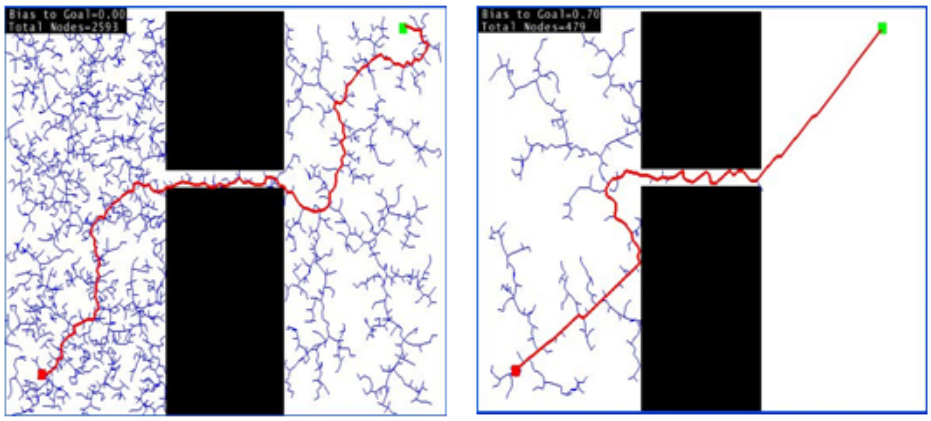

图 3.8 狭窄通道类型

我们在图 3.5、图 3.6、图 3.7、图 3.8 展示了快速扩展随机树算法及其改进型 Bias-goal RRT 算法在不同障碍环境中所表现出来的特点，所选取的 4 种环境比较典型所以被挑选使用。包括随机分布障碍类型、迷宫类型、混杂障碍类型和狭窄通道类型。

3.4.2 实验结果分析

除了以上的几个典型的环境，我们还做了大量的实验工作，随着工作的深入逐步总结出 RRT 算法在障碍物分布不同时的不同表现。RRT 算法倾向于探索未知环境，收敛速度快，有很大的改进空间。

在试验中主要比较了基本 RRT 与 Bias-goal RRT 在不同障碍物条件下的工作状态，结果显示 Bias-goal RRT 却具有强烈的搜索倾向。收敛速度更快。并且可以得出结论，即 Bias-goal RRT 算法通过具有倾向性的搜索方法，其表现要远远好于基本的 RRT 算法。

这一点无论是从搜索时间的长短和扩展节点树的多少都可以体现出来。

表 3.6 基本 RRT 与 Bias-RRT 算法的扩展时间以及扩展节点数的对比

地图类型	RRT 扩展时间(s)	Bias-RRT 扩展时间(s)	RRT 扩展节点数	Bias-RRT 扩展节点数
分布障碍类型	4.625	2.643	800	111
迷宫类型	26.623	10.465	4782	1225
混杂障碍类型	10.654	3.765	2875	971
狭窄通道类型	9.454	2.361	2593	479

由此可以对 Bias-goal RRT 算法的优点进行以下总结：

Bias-goal RRT 算法能适应不同的障碍物分布环境。如分布障碍类型、迷宫类型、混杂障碍类型、狭窄通道类型中的表现均非常出色。在以上环境中 Bias-goal RRT 算法都成功的找到了路径，并且路径质量都由于基本 RRT 算法。Bias-goal RRT 算法的耗时少于基本 RRT 算法，其消耗的时间一般是基本 RRT 算法的一半以内，在某些特定的环境中，如混杂障碍类型与狭窄通道类型所用的时间相对于基本 RRT 算法甚至更短。多次大量的仿真实验体现了该算法良好的运行速度。

3.5 本章小结

本章主要对比分析了基本 RRT 算法与其改进型 Bias-goal RRT 算法。描述了 RRT 算法的研究现状与发展前景。并对基本 RRT 的扩展方式进行了详细的阐述；描述了 RRT 算法的实现过程，首先对算法的流程进行了更加简洁的第二次描述。然后介绍了应用 OpenGL 技术实现的算法，详细分析了算法中具有代表性的两个函数；提出了基本 RRT 算法的主要缺陷，并提出改进的 Bias-goal RRT 算法，主要思想就是通过参数 Bias-goal 来实现偏向目标的搜索策略；通过仿真试验将基本 RRT 算法和 Bias-goal RRT 算法进行对比，挑选了 4 个比较典型的仿真环境，即障碍物随机分布类型、迷宫类型、混杂障碍类型和狭窄通道类型。通过对比表现出两种算法在搜索效率上的差异，主要包括搜索时间和搜索范围。

第 4 章　动态环境下 RRT 算法在非完整系统中的实现

4.1　引言

第二章提到过在移动机器人的运动学模型中提到了非完整约束[54]。非完整约束的意思是在系统中考虑机器人的种种运动参数。面对这类问题，科学家们已经并没有研究出了更好的办法。对于轮式移动机器人来说。这些方法大多将其视为一个点，或者是非常规则的形体来应用。也就是说解决的更多的是几何问题。而对于机器人面临的动力运动学问题并没有什么更好的解决办法。

然而为数不多的科学家针对这类问题也提出了自己的方法:如科学家 Launlond[55-56]的方法是首先并不考虑机器人所受的种种约束，即将机器人看做一个点先找到可行路径。然后处理这些可行路径。处理的方法是想将路径分解为一个个小的路径段，然后用预先设置好的标准路径来替代这些小的路径段。比如在转弯处用 Reeds&shepp 线段[57]。这种方法的主要思想就是将规划的路径用已知的路径替代。在这种思想的指导下，科学家们提出了多种替代方式：如 Sekhavat[58]提出了概率路标法(Probabilistic Roadmap)；还有 Barraquand 与 Latombe[59]提出的迭代法；Donald 与 Xavier[60]提出的标准栅格法与 Kindel 和 Hus[61]提出的不规则栅格法等等。

4.1.1　RRT 算法的优势

快速扩展随机树算法是一种可以将机器人的动力学约束考虑其中的运动规划方法。和其它的随机采样的算法一样，RRT 算法搜索空间的速度远远快于完整搜索算法。RRT 算法提供的路径并不是最优的，但是它在大多数实例中被证明是非常有效的。Tan 将该方法应用在无人机的运动规划中[62]，Frazzoli 则把该算法应用于宇宙飞船的运动规划[63]。

快速扩展随机树算法(RRT)就像它的名字所形容的那样，能够在执行算法过程中生成一个树状结构。这个树状结构的起点就是我们的机器人的起始位置，然后经过迭代扩展到机器人构型空间中的其它部分。RRT 算法的一个重要属性就是它不但能规划出机器人的坐标点，而且还包括首向和速度。如果一条路径是由考虑这些约束的 RRT 算法生成的，那么我们就知道那是一条可行路径。

4.1.2　机器人在状态空间中的表达

机器人在状态空间中受到种种的微分约束，所以其在状态空间中主要用微分约束来

表达。

而在描述一个完全运动学问题时，理想状态是使状态空间能够包含机器人的各种位姿属性。为了达到这个状态，本文利用微分约束来转换机器人的几何模型。具体方法如下：

设C为系统中机器人所用位姿的集合。$q \in C$表示机器人的每一个具体的位姿。设状态空间用字母X来表示。$x \in X$代表状态空间中的某一个特定的状态。

定义：$x = (q, \dot{q})$。

非完整系统中机器人也遵循着某种的守恒定律。比如在状态空间运动时，机器人的角动量是守恒的。在使用状态空间表达时，这可以被简单表达成m个隐式方程的形式，例如$g_i(x, \dot{x}) = 0$，$i = 1, \ldots, m$且$m < 2n$。n代表空间的维数。而约束可以表示为：

$$\dot{x} = f(x, u) \tag{4-1}$$

式中：$u \in U$；

U——控制量的集合；

然后要做的就是逼近表达式(4-1)，在一个时间段中，当前的状态为$x(t)$，$\left\{u(t^\alpha) \mid t \le t^\alpha \le t + \Delta t\right\}$，我们要求$x(t + \Delta t)$。使用标准形式的4阶龙格-库塔积分，假定输入量为常量u。这样只要已知系统的当前状态与控制量，就可以沿着时间轴逐步仿真。

$$x' = f(x(t) + \frac{\Delta t}{2} f(x(t), u), u),$$

$$x'' = f(x(t) + \frac{\Delta t}{2} x', u),$$

$$x''' = f(x(t) + \frac{\Delta t}{2} x'', u),$$

$$x(t + \Delta t) \approx x(t) + \frac{\Delta t}{6}(f(x(t), u) + 2x' + 2x'' + x''')$$

4.2 RRT 扩展算法在非完整系统的应用

非完整系统被定义为具有一系列约束方程的系统。这些约束方程包含系统配置变量的时间导数。这些方程是不可积分的。当系统的控制变量少于配置变量的时候就多用到这些方程。例如当车型机器人在三维构型空间移动的时候具有两个控制变量（线速度与角速度）。而结果是在构型空间中的任何一条路径在该系统中都不是适合的路径。这就是在完整约束系统中运动规划中单纯的几何算法并不能直接应用于非完整系统的原因。

4.2.1 Bias-goal RRT 算法在非完整系统中的应用

当障碍物产生的约束被直接传递到构型中的时候，非完整约束处理的是切空间。由一系列机器人参数和它们的导数代表。甚至当没有障碍物的时候，规划非完整运动也不是一件容易的事。如今还没有一个通用的算法能保证非完整运动规划能准确的到达给定的目标点。仅有的方法叫做估计方法（仅能保证该系统能到达目标点临近的点）。还有的就是为了特殊系统服务的准确方法。幸运的是，这些特殊系统几乎能够为所有现有的机器人服务。

避障将运动规划的难度又上升了一个级别。在该级别中我们既要考虑障碍物所带来的约束（例如处理系统的构型参数）还要考虑与非完整约束相关的参数导数。它将描述障碍物的几何技术与描述非完整运动的特殊结构的控制理论结合了起来。这样的结合使非完整系统中的机器人避障成为了可能。

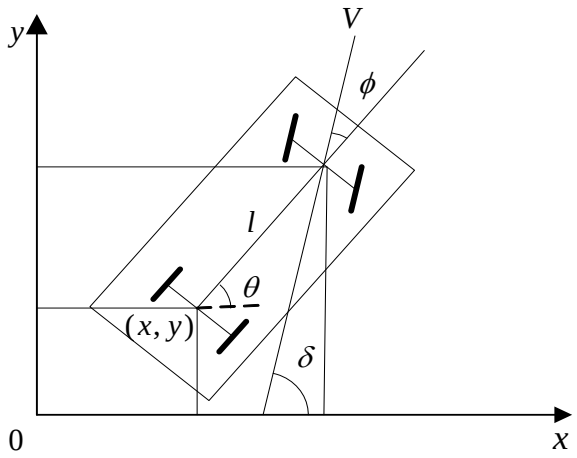

图 4.1 轮移式机器人模型图

具体的参数所代表的意义如表 4.1 所示

表 4.1 轮式机器人中各个参数的意义

θ	机器人与 X 轴夹角
ϕ	首向角
V	前轮速度
L	前后轮距离
δ	V 相对 X 轴正向的夹角

机器人的状态可以用 $q(x,y,\theta)$ 来表示。用 $inputu(V,\phi)$ 来表示输入的控制变量。系统受到的非完整约束可以表示为：

$$dx\sin\theta - dy\cos\theta = 0$$

$$\frac{dx}{dt}=V\cos\phi\cos\theta$$
$$\frac{dy}{dt}=V\cos\phi\sin\theta$$
$$\frac{d\theta}{dt}=(V/L)\sin\phi$$

C中任意两个位姿的间距可以表示为：

$$\text{distance}\quad(p,q)\quad=\quad[A1(x_p-x_q)^2+A2(y_p-y_q)^2+A3\min((\theta_p-\theta_q)^2,(\theta_p-\theta_q+2\pi)^2)]^{0.5} \tag{4-2}$$

式中：$A1$，$A2$，$A3$——权值参数

取$A1=1$，$A2=1$，$A3=L^2$。应用 Bias-goal RRT 算法。在随机目标点qr的附近找到距离qr最近的节点vn，在此基础上，Bias-goal RRT 算法寻找最优控制输入(V,ϕ)，从而使新节点qn距离qr最近。从此，一个优化问题就产生了：

$$\begin{cases}\min\limits_{qn\in C,qr\in C}(\text{distance}(qn,qr))\\ x_{qn}=x_{vn}+Vdt\cos\phi\cos\theta_{vn}\\ y_{qn}=y_{vn}+Vdt\cos\phi\cos\theta_{vn}\\ \theta_{qn}=\theta_{vn}+(Vdt/L)\sin\phi\end{cases} \tag{4-3}$$

对上式求解，即可得出最优输入，然后依照得出的ϕ与V推出新节点qn。这里举一个例子说明：

然后对函数$\text{distance}(qn,q)$求导航角ϕ的偏导数。假设控制量V是仅取两个值的集合$V=[-1,1]$。

当$V=-1$时，

$$\frac{\partial(\text{distance}(qn,q)^2)}{\partial\psi}=a\cos\phi+b\sin\phi+c\cos\phi\sin\phi=0 \tag{4-4}$$

当$V=1$时，

$$\frac{\partial(\text{distance}(qn,q)^2)}{\partial\psi}=a\cos\phi+b\sin\phi-c\cos\phi\sin\phi=0. \tag{4-5}$$

其中：

$$\begin{cases}a=(A3/L)\min[(\theta_{vn}-\theta_{qr}),(\theta_{vn}-\theta_{qr}+2\pi),(\theta_{vn}-\theta_{qr}-2\pi)]\\ b=(x_{vn}-x_{qr})\cos(\theta_{vn})+(y_{vn}-y_{qr})\sin(\theta_{vn})\\ c=A3/(L^2)-1\end{cases} \tag{4-6}$$

解方程 (4-4)，(4-5)，得出两个最优导航角 Φ。然后将控制对 (V,ϕ) 代入 $\text{distance}(qn,qr)$，取使 $\text{distance}(qn,qr)$ 较小控制对 (V,ϕ) 并得到导航角 $\phi=\arctan(a/b)$。根据 ϕ 与 V 推出新节点 qn。这是 Bias-goal RRT 算法扩展新节点的关键步骤。

受非完整约束的机器人 Bias-goal RRT 路径规划算法

$$BUILD\_BIAS-GOAL\_RRT(q_{start},K)$$

Step1：$T_{init}(q_{start})$ for $K=1$ to $K<MAX\_TREESIZE$ do

Step2：$q_{target}\leftarrow BIAS\_Goal(q_{goal},BIAS)$

Step3：$q_{near}\leftarrow Nearest\_Neighbor(q_{target},T)$

Step4：$Inputu\leftarrow Optm\_Input(q_{near},q_{target})$

Step5：$q_{new}\leftarrow New\_State(q_{near},Inputu,\Delta t)$

Step6：$T_{add}\_vertex(q_{new})$

Step7：$T_{add}\_edge(q_{near},q_{new},u)$

Step8：return T

这里的步骤 5 得到的 q_{new} 是可避障的。如果 q_{new} 与障碍物由碰撞则返回步骤 2 并重新产生局部目标点 q_{target}，直到 q_{new} 可避障为止。

在该算法中，最佳控制量输入算法的步骤如下：

Step1：用最优函数 $\min_{qn\in C,qr\in C}(\text{distance}(qn,qr))$ 以算出控制量的输入：

$$u=Optimization(V,\phi)$$

Step2：导航角范围为 $\phi<|MAXfi|$，判断导航角值的范围的方法如下：

$$if\quad \phi>MAXfi$$
$$\phi=MAXfi$$
$$else\quad if\quad \phi<-MAXfi$$
$$\phi=-MAXf$$

Step3：如果得到的控制量 (V,ϕ) 所确定的新节点能避障则转到转 step4，如果不能避障则采用逐步加导航角逐步检测法，即给出一个最小角度 $\min\_angle$，加上 $\min\_angle$ 再检测是否可避障，直到可以避障为止。

$$\phi_1=\phi \qquad \phi_2=\phi$$

$$while(Collision(\phi_1,Obsts))$$

$$\phi_1=\phi_1+\min\_angle$$

$$q_{n1} = New\_State(V, \phi_1)$$
$$while(Collision(\phi_2, Obsts))$$
$$\phi_2 = \phi_2 - small\_angle$$
$$q_{n2} = New\_State(V, \phi_2)$$
$$d_2 = dis\tan ce(q_{n2}, q)$$
$$u_2 = (V, \phi_2)$$

Step4：取两个控制量作用得到的较小结点为新结点，并记下相应的控制量。

$$if \quad d_1 < d_2$$
$$optimal\_u = u_1$$
$$else$$
$$optimal\_u = u_2$$
$$return(optimal\_u)$$

4.2.2 贝塞尔曲线路径的平滑处理

贝塞尔曲线在路径平滑方面的非常有效的工具[64,65]，它具有很多优秀的属性。其中的一点就是贝塞尔曲线可以重建局部路线而不更改整条路径的形状。所以本节我们应用贝塞尔曲线来完成 Bias-goal RRT 算法执行后的后续工作，创建一条更加圆滑的路径。一般来说 n 阶贝塞尔曲线可以定义为以下形式：

$$B_p(t) = \sum_{i=0}^{n} B_i^n(t) P_i \qquad t \in [0,1] \tag{4-7}$$

式中：$B_i^n(t)$——伯恩斯坦多项式；

其定义如下：

$$B_i^n(t) = (\frac{n!}{i!(n-1)!}) t^i (1-t)^{n-1} \qquad i = 0, \ldots, n \tag{4-8}$$

本为主要考虑 n=3 的情况，也就是三次贝塞尔多项式 $B_p(t)$。随着伯恩斯坦多项式的扩展，我们能得到贝塞尔曲线的参数方程。

$$\begin{cases} B_p(t) = At^3 + Bt^2 + Ct + P_1 \\ A = P_4 - P_1 + 3P_2 + 3P \\ B = 3(P_1 + P_3 - 2P_2) \\ C = 3(P_2 - P_1) \end{cases} \tag{4-9}$$

如(4-9)式所表示的，四个点 P_1、P_2、P_3 和 P_4 确定了一个三次的贝塞尔曲线；这里点 P_1 和 P_4 称为曲线的端点，而 P_2 和 P_3 成为控制点。该曲线有如下属性：

(1) P_1 和 P_4 为贝塞尔曲线的端点， $B_p(0)=P_1$ 并且 $B_p(1)=P_4$。

(2)在贝塞尔曲线中与线段 $\overline{P_1P_2}$ 相切于点 P_1，与线段 $\overline{P_3P_4}$ 相切于点 P_4。

应用三次贝塞尔曲线的第一个属性，路径段 G_p 可以被构建成为 $P_1=q_{init}$ 与 $P_4=q_{goal}$，即随机树在扩展的时候每一个扩展的端点 q_{init} 可以定位 P_1 而每一个扩展的目标点 q_{goal} 可以定为 P_4。这样随机树就可以按照三次贝塞尔曲线的形式进行扩展了。

具体的扩展过程如图 4.2 所示：

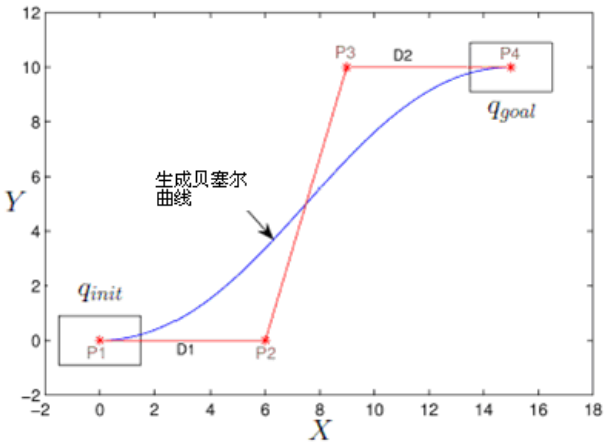

图 4.2 贝塞尔曲线路径处理

总的来说，该方法在目标点可以通过机器人连续的向前运动到达而且不需要机器人的其他机动动作的时候表现非常不错。特别值得一提的是，如果起始点和目标点距离很小的时候，贝塞尔曲线可以提供一个大的转弯。这一点并不是我们所期望的。但是，全局路径中的大的转弯对移动机器人的表现并没有决定性的影响。因为全局路径通常是为了为局部路径的生成提供帮助。

4.3 Bias-goal RRT 算法动态环境下的路径规划

在考虑非完整约束的基础上，机器人的动态避障亦是问题的难点。动态环境就意味着障碍物是运动的，或者目标点是运动的，本文考虑的只是障碍物的运动。由于障碍物的运动不规则，所以机器人对移动障碍物的预测就显得尤其重要。在对障碍物准确预测的基础上，应用规划算法进行实时避障是本节的主要内容。

4.3.1 移动机器人在动态环境下避障策略

在考虑非完整约束的情况下，我们可以将构型空间中的各种参数描述如下，见表 4.1 所示：

表 4.2 动态环境中的各种参数表示

τ	规划周期，在此周期内障碍物可视为不动的
S_v	传感器可感知的范围
S_p	机器人规划路径的区域
$V_i(t)$	t 时刻各个障碍物 $DObs_i(i=1,\cdots m)$ 的运动速度。
$\theta_i(t)$	t 时刻各个障碍物 $DObs_i(i=1,\cdots m)$ 的运动方向。
$x_i(t)$	t 时刻各个障碍物 $DObs_i(i=1,\cdots m)$ 的运动方向。
$y_i(t)$	t 时刻各个障碍物 $DObs_i(i=1,\cdots m)$ 的垂直坐标。
$SObs(t)$	障碍物的集合

机器在 τ 的时间段内首先探测感知障碍物各种运动信息 $SObs(t)$。这个工作是在 S_v 中完成的。接下来利用感知到的信息 $V_i(t)$，$\theta_i(t)$，$x_i(t)$，$y_i(t)$ 判断障碍接下来的运动。再得出障碍下一步的运动数据之后，机器人利用 Bias-goal RRT 算法规划实时路径。这一工作是在 S_p 内完成的。规划路径时 RRT 规模受到限制，节点数目不可以超过一个临界值。如果 RRT 的节点数超过了这个临界值它就不再扩展了。一棵随机树的构建停止后，τ 的寿命也就到了，接着自动进入下一个时间间隔的工作。

局部规划与碰撞检测：

局部规划的示意图如图 4.2 所示，在这个图中障碍物既有动态有用静态，黑色矩形为机器人的所在位置，小一点的圆 S_p 表示机器人规划路径的区域。大一点的圆 S_v 表示机器人传感器可感知的区域。在图中可以看到障碍物 $(Obst1)$ 从 t 时刻的位置运动到 $(t+1)$ 时刻的位置。机器人所要做的工作就是预测其从一个时间点 t 开始到下一个时间点 $(t+1)$ 的运动趋势。即确定 $(t+1)$ 时刻障碍物的位置与坐标。然后从 $(t+1)$ 时刻开始将所有环境看成是静态的，进而规划路径。

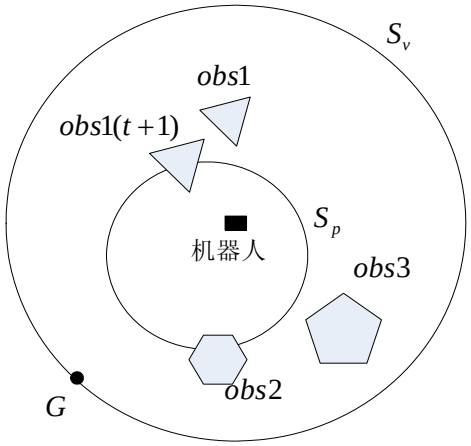

图 4.2 局部规划

这里将局部规划的方法作如下介绍， 首先设随机函数产生的随机值为 N_{random}。则局部子目标的选取方法分为两种情况讨论：

1)如果 N_{random} >Bias-goal，则在表示 S_v 的范围的圆上任意跳出一个点。

2)如果 N_{random} <Bias-goal，那么就连接机器人矩形的中心点与全局目标点。两者连线与 S_v 的范围所在圆的交点即为所求。

文章中应用的检测两物体是否发生碰撞的方法非常简单，文献[66]对该方法作了详细的介绍。现作如下简单介绍与说明。

在该碰撞检测法中我们将障碍物的边看作是有方向的，以五边形障碍物为例，如图 4.3 所示：

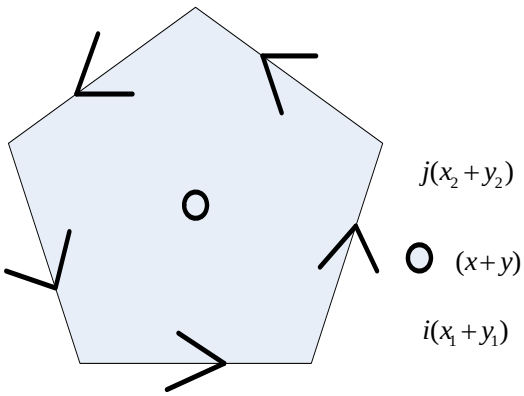

图 4.3 碰撞测原理图

检测的方法是如果机器人的一个点运动到有向直线的左边时就代表发生了碰撞。举个例子来说，如图 4.3，如果点 $i(x_1,y_1)$ 与点 $j(x_2,y_2)$ 分别代表多边形的任意一条边上的两个顶点。任意点的位置为 (x,y)，那么判断这个任意点是否与该多边形发生碰撞可以根据表达式(4-10)的正负来确定：

$$(y-y_1)-((y_1-y_2)/(x_1-x_2))(x-x_1) \tag{4-10}$$

如果上式为正，那么可以判定点在线的左边，即点在障碍物内部与障碍物发生碰撞。如果上式为负时，则可以判定点在线的右边，即不发生碰撞。如果机器人不是一个点而是一个多边形的时候，就涉及到判定两个多边形是否有重叠的情况发生。如果发生重叠现象就可以说他们的某条边一定有交点。这时就可以取两条相交线的两顶点按照上述的方法进行碰撞检测的判断。

对碰撞检测方法可以进行的优化，而优化的目标是为了提高速度。优化的思想是将多边形障碍物看成圆，该圆包含了整个多边形。

具体分两种情况：当机器人是点状的，如果该点不在圆中则不发生碰撞。如果机器

人为多边形，按上述方法作圆，两圆如果有没有交点则不发生碰撞。以上两点中如果发生碰撞就按照优化之前的方法进行碰撞检测。

4.3.2 Bias-goal RRT 算法在动态环境下应用

对处于动态环境下的移动机器人来说，最重要的是对动态信息的掌握与判断，如移动障碍物的运动趋势等等。然后根据这些信息调整自身的运动。在动态环境中快速随机搜索树的算法流程的流程与静态环境不同，它要根据动态障碍物位置的变化不断的对原有路径进行修正以完成动态避障。在动态环境下 Bias-goal RRT 算法的各种状态如表 4.2 所示：

表 4.3 动态环境下随机树的状态表示

G_{end}	机器人所要到达的终点
T	随机树
x_{init}	随机树的根结点
x_{goal}	临时目标点
$\dot{x} = f(x,u)$	微分约束

具体扩展步骤如下表 4.4 所示：

表 4.4 动态环境下随机树的具体扩展步骤

(1)	初始化，S_v，S_p，τ，RRT 的$T_{init}(x_{init})$
(2)	在S_v范围内搜索$SObs(t)$的位置，并在该范围内规划出局部目标点。
(3)	在S_p范围内用 Bias-goal RRT 算法规划路径进行障碍物的规避。
(4)	机器人移动固定的步长
(5)	在下一个时间段中再次探测障碍物信息与环境信息
(6)	机器人是否移动到G_{end}，如果是就终止算法，不是就返回执行步骤(2)

规划过程是按照时间段划分的，在一个时间段的规划过程中算法生成一棵随机树。但是这棵随机树的节点数目与规模是由时间段的长短决定的。时间段越长，节点数目越多，数的深度越大。当搜索树的深度超过时间段时，算法停止搜索树的生长。

每当开始一个新的时间段的时候，传感器就会感知目标点的位置，也同时会感知移

动障碍物的运动参数。如果移动障碍物的运动速度与运动方向保持恒定。机器人就可以预测在本时间段内移动障碍物的运动趋势。每个时间段分为若干个子时间段，算法在运行过程中通过预测子时间段内的信息进行规划。

子时间段的长度决定了在该段时间内搜索树的扩展规模。搜索树的节点都包含了移动机器人在该子时间段内的运动参数。如位置坐标、速度大小与运动方向等等。以此为基础，算法可以计算移动机器人出下一步的运动状态。如转角的大小与速度该变量的大小等等。以此时间段依次循环，一直到机器人找到目标点，算法结束。

容易看出，算法不是只运行一次就结束了，而是在每一个周期内都要运行。每一个周期结束时机器人所处的节点成为下一个周期内随机树算法的根节点。在每一个周期内由于系统实时性与机器人预测准确性的要求，随机树的生长要有所限制。在实际情况中，即处于动态障碍物的环境中，这种限制有其合理性。也就是说随机树的扩展需要一定的时间，在此时间周围环境是不断变化的。随机树的生成要考虑周围环境的变化。如果生长时间过长了，机器人在此段时间里并不能进行避障而有可能与移动障碍物发生碰撞。所以随机树的生长时间要取一个合适的值以适应周围环境的变化。大量的仿真实验也证明了这一点。

规划的具体流程图如图 4.4 所示：

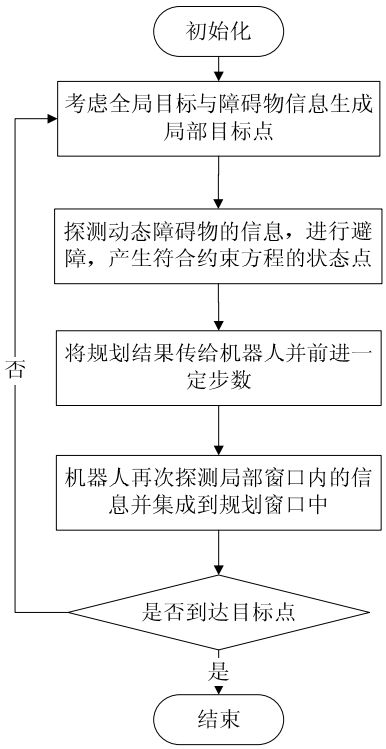

图 4.4　RRT 动态扩展流程图

对于机器人的定位我们只考虑它的几何中心，即几何中心坐标 $A(x,y)$ 为机器人所在的位置。机器人沿直线 AM 运动。并且机器人的速度的最大值不是无限的，这里规定速度的最大值为 $v_{\max}$，它的方向与 X 轴所夹的角度为 β。

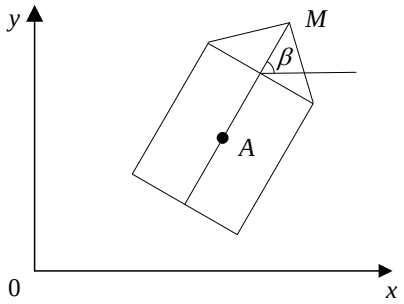

图 4.5 机器人的模型表示

这里在某一个时间段内，机器人在获得了这个时间段内障碍物的运动趋势与即将到达的位置之后利用算法 Bias-goal RRT 搜索路径，完成以后机器人利用规划的路径向前运动一个步长并判断是否到达目标点。如果到达目标点则规划结束。

如果没有到达目标点则自动开始下一个周期的规划。其中 Bias-goal RRT 树建立的算法流程下，

$BUILD\_BIAS\_RRT(x_{init},K)$

1 $T_{init}(x_{init})$;

2 for $k=1$ to $K<MAX\_TREESIZE$ do

3 $x_{target} \leftarrow BIAS\_Configuration(qgoal,BIAS)$

4 $x_{near} \leftarrow Nearest\_Neighbor(x_{target},T)$;

5 $x_{new} \leftarrow New\_State(x_{near},x_{target})$;

6 $u \leftarrow Output\_Control(x_{new},x_{near})$;

7 $T_{add}\_vertex(x_{new})$;

8 $T_{add}\_edge(x_{near},x_{new},u)$;

9 return T

在步骤 5 中必须保证 x_{new} 是处在自由空间中的。如果与障碍物发生了碰撞，返回步骤 3 重新生成 x_{target}，直到保证 x_{new} 是出于自由空间中的。

接下来讨论在非完整约束条件下 x_{near} 的生成步骤：

$Nearest\_Neighbor(T,x_{\text{target}})$

1 $d_{\min} \leftarrow Max$;

2 for $k=T_{begin}$ to T_{end}

3 $d \leftarrow \text{Distance}(x_k,x_{\text{target}})$;

4 if $d \le d_{\min}$

5 $x_{near} \leftarrow x_k$;

6 $d_{\min} \leftarrow d$;

7 return x_{near}

新节点的获取在本节中需要相应的获取算法，要能够适应动态环境的变化。所以新节点的获取方法与非动态环境下有一定区别。现在介绍动态环境下的新节点 New-State 的获取方法及算法流程。

New-State 的获取方法如图 4.6 所示：

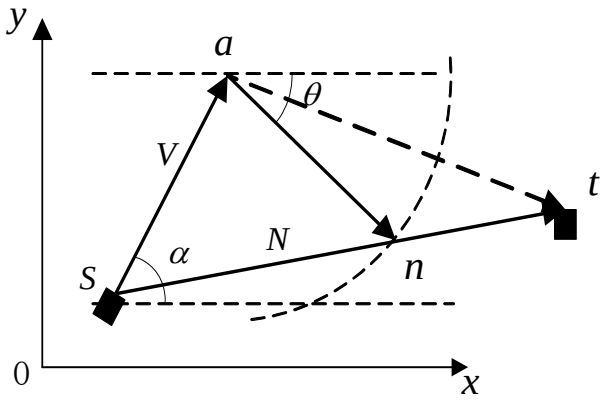

图 4.6 新节点的产生

机器人在图中用矩形块表示。与基本 RRT 的以步长为单位的构造方法不同，这里的 RRT 算法不但要考虑长度，还要考虑方向。即通过一种矢量计算的到移动机器人的下一步的可行点。像图所表示的那样，s 点是机器人的当前坐标点。t 表示 x_{target}。a 表示一个时间段过去后机器人运动到的方位。$\vec{V}$ 表示是使机器人运动到 a 点的速度大小与方向。该矢量值与 x 轴正向夹角为 α。但是机器人要运动到 t 位置，所以要选取一个控制量输入来使机器人向 t 点靠近。这个改变机器人运动状态的控制量我们设为 $\vec{A}$。表示机器人在一个时间段内速度的改变量。

在图中控制量 $\vec{A}$ 可以用矢量 $\vec{at}$ 表示，这个矢量可以使机器人很准确的到达随机点 t。然而由于种种因素的限制，如加速度与速度对控制量的限制使得输入矢量达不到矢量 $\vec{at}$。因此简便起见，本文就将速度的改变量设为固定的最大加速度 Q，这样机器人就可以最为迅速的接近 t 点了。

这样很显然，机器人不一定准确的到达随机点 t。实际节点 n 的确定方式为：以 a 为圆心，以 Q 长画弧，该弧与 st 的交点即为新的节点 n。而矢量 $\vec{N}$ 的方向就是新节点 n 的

速度方向。

$$New\_State(q_{near}, q_{\text{target}}, \Delta t)$$

Step1：$x_s \leftarrow x_{x_{near}}$

$y_s \leftarrow y_{x_{near}}$

Step2：$x_a \leftarrow x_{q_{near}} + xspeed_{q_{near}} \Delta t$

$y_a \leftarrow y_{q_{near}} + yspeed_{q_{near}} \Delta t$

Step3：$\theta = angle(a)$

$\alpha = angle(sa)$

$\vec{N} = \vec{V} + \vec{A}$

Step4：$x_n = sa\cos\alpha + an\cos\theta$

$y_n = sa\sin\alpha - an\sin\theta$

$x_{q_{new}} = x_n$

$y_{q_{new}} = y_n$

Step5：$xspeed_{q_{new}} = \dfrac{x_n - x_{q_{new}}}{\Delta t}$

$yspeed_{q_{new}} = \dfrac{y_n - y_{q_{new}}}{\Delta t}$

4.4 仿真试验与结果分析

本章成功实现了动态环境下将 Bias-goal RRT 算法应用于非完整约束系统下移动机器人路径规划中。进行了仿真实验并对实验结果进行了分析。

本章仿真实验所使用的电脑配置与第三章相同。本章试验用 MATLAB 语言实现，版本为 R2007a。试验结果如图所示。其中白色区域为自由联通空间，绿色区域代表静态障碍物，黑色矩形为移动障碍物，该移动障碍物将在某一时刻将机器人与目标点分割开来。红色矩形为移动机器人，粉色圆形为目标点。Bias-goal RRT 算法刷新时间段为 10 秒。

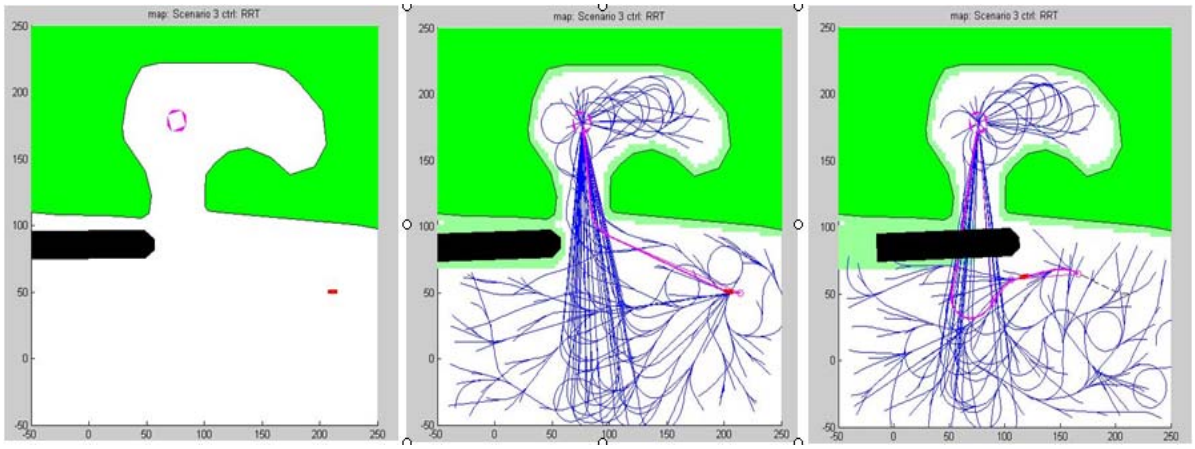

图 4.7 动态路径规划仿真

图 4.7 表示动态环境下移动机器人从起始位置搜索路径，并且在搜索树遇到移动障碍物的时候自动调整路径并且绕过移动障碍物的过程。

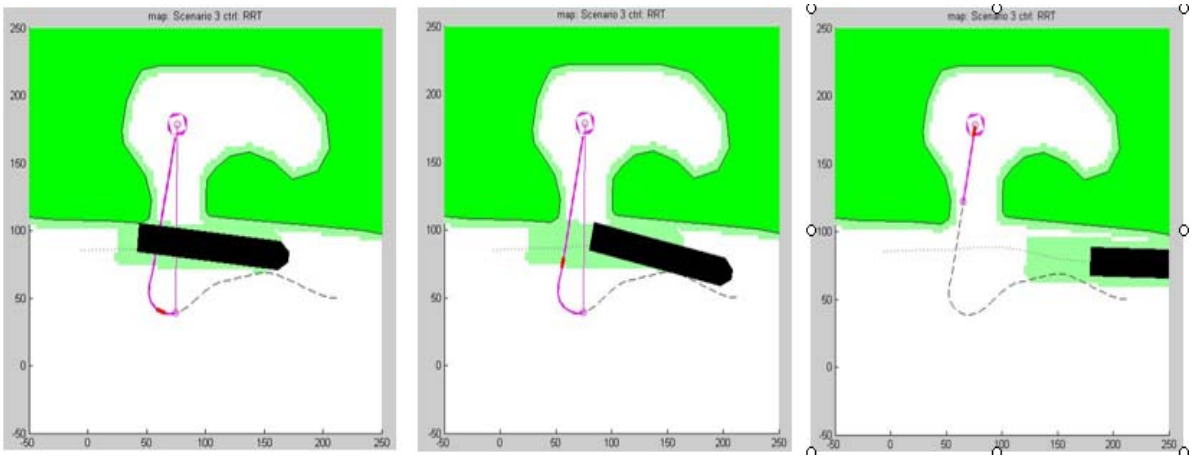

图 4.8 动态路径规划仿真

图 4.8 表示动态环境下移动机器人在确认不会与移动障碍物发生碰撞的情况下逐渐跟踪规划好的路径，并最终到达目标点的过程。

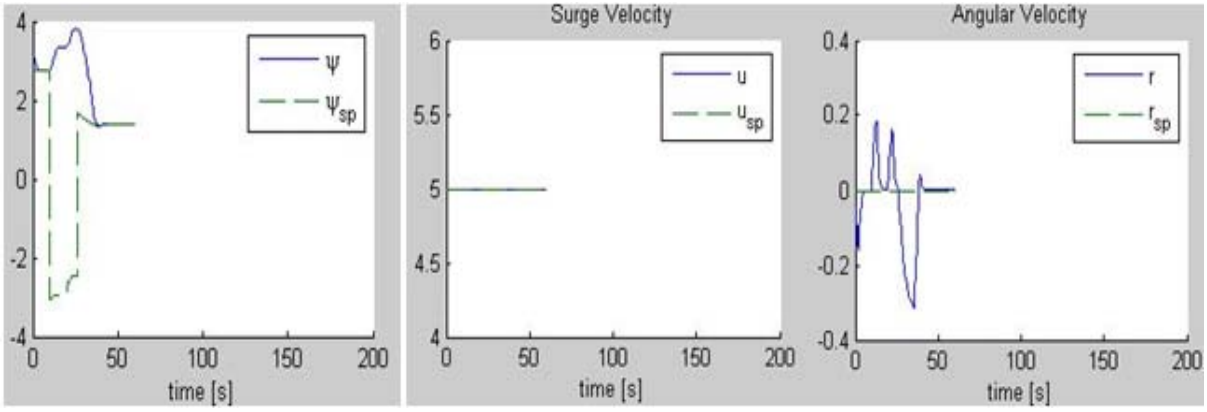

图 4.9 机器人首向角、线速度与角速度曲线

图 4.9 表示移动机器人从起始点运动到目标点的过程中首向角，线速度，角速度的变化曲线。可以看出：

(1) 实际线速度 u 与输入的线速度 u_{sp} 曲线重合，说明在仿真过程中移动机器人的速度大小并没有改变，所以该算法并不涉及到减速过程，机器人都是保持固定速度移动，这样对实际的运动时间的缩短有积极作用。

(2) 实际角速度 r 与输入角速度 r_{sp} 还有实际首向角 Ψ 与输入首向角 Ψ_{sp} 有偏差说明在动态环境下移动机器人面对环境的变化（包括障碍物的变化，路径的变化）不断调整原来的路径，说明该算法能较好的适应不断变化的动态环境。

在图 4.7 和图 4.8 中，机器人用改进的 Bias-goal RRT 算法不断的在每一个时间段里规划路径，使得机器人能够尽可能的做到对移动障碍物的规避。同时将非完整约束包含到 Bias-goal RRT 的节点的扩展过程中，使得扩展的新节点满足非完整约束。实验结果表明 Bias-goal RRT 算法能够迅速的找到目标点，并且在搜寻目标点的时候具有偏向性。提高了机器人的工作效率。在避障中，Bias-goal RRT 算法能够很好的预测移动障碍物的运动趋势与下一步位置，然后根据这些条件提供符合这个时间段里避障的可行路径。在下一个时间段中则重新运行 Bias-goal RRT 算法，对路径进行更新，保证满足动态条件下路径规划的需求。

由于 RRT 算法是一种随机采样算法。所以每次运行的结果的差别很大。所以我们进行的多次的仿真实验。如表 4.4 所示。

表 4.5 动态环境下的仿真记录

仿真次数	全部节点数	仿真时间(s)
1	4436	25
2	4587	26
3	4657	23
4	6912	37
5	4546	21

本章所论述的非完整系统中动态环境下的路径规划对于机器人执行任务有着深远的意义。传统的算法并不能很好的解决这一问题，原因就是它们无法适应地图环境的变化。

文中应用的 Bias-goal RRT 算法能轻松适应为完整约束系统，并且在动态障碍物面前有着不错的表现。RRT 算法的概率完整性使得地图中的任何一个节点都有被利用的可能。所以该算法有着不俗的搜索能力。该算法的另一个特性就是收敛迅速。Bias-goal 参数使得扩展节点的过程有偏向目标的趋势。机器人在每一个时间段中判断移动障碍物的位置并提供该段时间内的可行路径。实验证明该算法能够适应动态环境与非完整系

统。能有效的解决此类路径规划问题。

4.5 本章小结

本章实现了考虑非完整约束条件下 Bias-goal RRT 算法在动态环境中的路径规划。对轮式移动机器人复杂的运动学系统进行建模，并对该算法进行了处理，如利用贝塞尔曲线进行路径平滑处理等措施以使之适应为完整系统。使 Bias-goal RRT 具有更高的适应能力。

第 5 章　快速扩展随机树算法与人工势场法的结合应用

5.1　引言

上一章我们提出并实现了一种 RRT 算法的改进算法偏向目标型快速扩展随机树算法(Bias-goal RRT)。本章我们将用(Bias-goal RRT)算法与另一种算法人工势场法(Artificial Potential Field)相结合。

5.2　人工势场法

前面已经介绍过，人工势场法[22,70]是科学家 Khatib 在上世纪五十年代发明出来的。它的主要理念是将机器人所处的环境看成是一个受力场。机器人在这个受力场中受到来自障碍物的斥力与目标点的引力。在这两个力的合力作用下机器人能够实现避障与朝向目标点运动。

人工势场法具有很多优点，其中算法结构简单是最重要的一条，其次，该算法的运行效率非常高，并且对机器人本身的硬件系统要求非常低。这使得人工势场法普遍的被人们当作机器人的路径规划算法

人工势场法也存在着许多不足，如该算法是局部规划算法，这就导致该算法缺乏全局的环境信息。虽然局部的表现非常优异，但也导致了局部极小值得产生。机器人陷入局部极小点以后的表现非常糟糕，有时静止不前，有时前后抖动，有时来回摇摆。这都阻碍这机器人成功的到达目标点。这也是让人工势场法得以广泛应用的主要瓶颈之一。

为了发扬人工势场法的优点，克服其不足。许多科学家提出了许多不同的方法以摆脱局部极小所带来的困境。Barrquand 与 Latombe 研究的 RPP[68,69](Randomized Path Planner)算法，Hwang 等研究的 MPV(minimum potential valleys)算法等等。但是以上两种算法在动态环境下表现非常一般。

5.2.1　人工势场法的主要思想与基本原理

人工势场法是在避障中最依靠直觉的一种方法。这种算法非常简单，使用它可以我们可以在很短的时间内得到很不错的结果。Khatib 在 1985 提到了这一点[70]。

势场法的主要思想就是每个障碍物都产生对机器人的一个排斥力，而目标点对机器人产生一个吸引力。如果机器人按照这些力来指引它的前进的话，它就非常有希望到达目标点。但是这种方法远没有那么完美。

虚拟力

由障碍物产生的势场定义如下：

$$U_0(p)=\begin{cases}\frac{1}{2}(\frac{1}{\rho}-\frac{1}{\rho_0})^2 & if \quad \rho\le\rho_0\\ 0 & if \quad \rho>\rho_0\end{cases} \tag{5-1}$$

式中：ρ——从障碍物上的一个点到小车上的矢量

$\rho=\|P\|_2$，而 ρ_0 代表从障碍物产生的势场传播的有限距离。在最原始的势场法中一个点要经常被用到。这个点就是障碍五种离小车最近的点。

这个势场产生的力为：

$$F_0(p)=\begin{cases}\eta(\frac{1}{\rho}-\frac{1}{\rho_0})^2\frac{p}{\rho^3} & if \quad \rho\le\rho_0\\ 0 & if \quad \rho>\rho_0\end{cases} \tag{5-2}$$

目标点产生的吸引势场在(5-1)中表示出来，(5-1)中的 x 表示小车的并列矢量 x_d 是目标点的并列矢量：

$$\begin{cases}U_{X_d}(X)=k_p e^2\\ e=X-X_d\end{cases} \tag{5-3}$$

由这个场产生的力起到一个控制器的作用。在应用这个力之前必须做两点修改。首先，要提高这个方法的稳定性，要加入一个与机器人速度 $\dot{x}$ 成比例的一个损耗力。第二，这个成比例的力的最大值要有一个上限。这一点保势场法表现的连续性并且不依靠与目标点之间的距离。最终目标点产生的势场力为：

$$F_{x_d}(x)=-k_v\dot{x}-k_v\begin{cases}\frac{k_p}{k_v}e & if \quad \frac{k_p}{k_v}\|e\|\le V_{\max}\\ V_{\max}\frac{e}{\|e\|} & else\end{cases} \tag{5-4}$$

成比例部分产生力的上限为 $k_v V_{\max}$，而我们期望加在机器人上的力就是这些力的合力。具体的人工势场法流程图如图 5.1 所示。

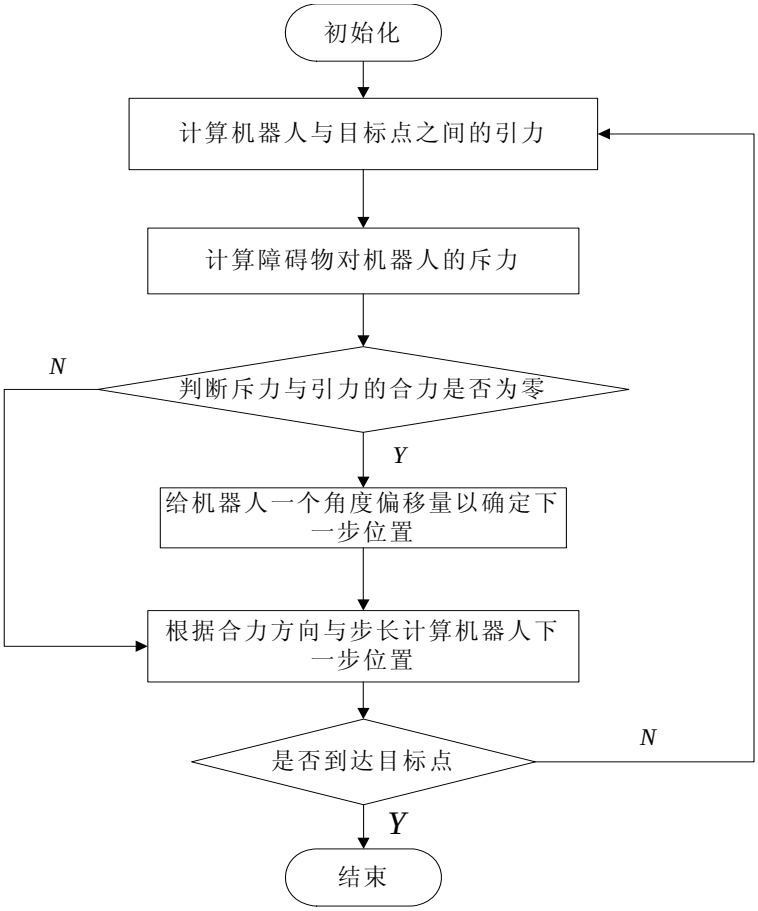

图 5.1 人工势场法流程图

从人工势场法的主要思想可以了解：在机器人所处的受力场中，机器人至始至终受到一个来自目标点的吸引力，该吸引力决定着机器人能够始终朝向目标点运动。而机器人并不是一直都受到障碍物的斥力，因为机器人离障碍物远的时候并不受到斥力，只有在距离障碍物一定距离的时候才受到障碍物的斥力。所以人工势场法只具有局部特性而全局特性不足。

因此，人工势场法的效率和优化程度不如于全局规划算法，但人工势场法也有避障反应速度快，局部搜索效率高等优点。所以人工势场法在搜索复杂度较低的地形的时候会有较好的表现，如下图 5.2 所示即为人工势场法在简单地形中成功搜索路径的情形。

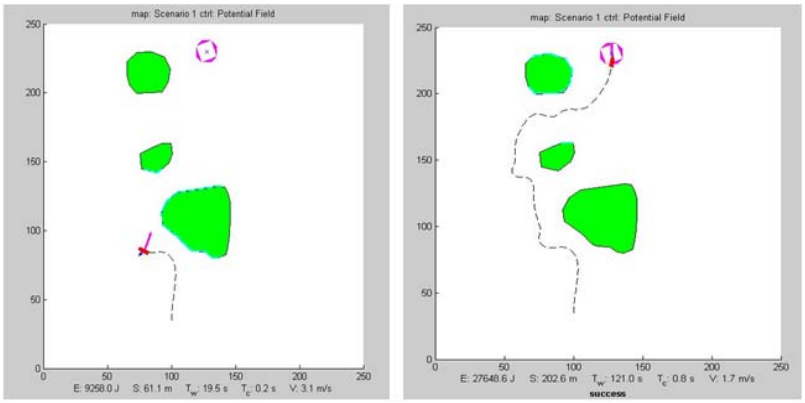

图 5.2 人工势场法静态环境下成功搜索路径的情形

图中环境表示：红色矩形为移动机器人，绿色区域为静态障碍物，白色联通区域为可自由移动的空间。粉色圆形为目标点。机器人上的红色箭头表示目标点对机器人的吸引力，蓝色箭头表示障碍物对机器人的斥力。

5.2.2 人工势场法的缺陷

人工势场法是一种简单而经典的避障算法，但它主要有四点不足，这使它很难被应用[28]。

(1) 由局部极小而产生的锁死现象

如同所有的局部路径规划法，势场法也会有产生局部极小的可能性。如果移动机器人在某处的引力与斥力相抵消的话，它就卡住了。这就是局部极小。算法找到了解，但解不正确。在障碍物混乱复杂的环境中更容易产生局部极小。

(2) 无法穿越紧密相邻的障碍物

该算法无法在紧密相邻的障碍物中间找到路径，如：一扇门。当小车接近通道时，斥力的合力会比吸引力更大，小车可能会产生局部极小或者绕过通道而选择一条其它的路径。

(3) 在障碍物之间摆动

在原始的算法中，由障碍物产生的斥力随着小车与障碍物距离的增加而剧烈的减少。这就意味着小车必须距离障碍物相对近一些才能感受到力。在多障碍物的环境中就会导致摆动。

(4) 在狭窄通道内摆动

在进入狭窄通道之后小车的摆动会更加明显[28]。当小车运行在混乱复杂的环境中时这些缺点会表现得更加明显，而构型空间如果简单有序的话这种算法会很有效率。有很多方法可以用来摆脱局部极小从而改进该算法；如随机路径规划法。

表 5.1 人工势场法的几种典型的失效情况

(1)	由局部极小产生的锁死现象
(2)	无法穿越紧密相邻的障碍物
(3)	在障碍物之间摆动
(4)	在狭窄通道内摆动

以上就是人工势场法的主要失效情形，其根本原因就是算法陷入了局部极小点。而且随着环境复杂度的增加，局部极小点也就更容易形成。环境越复杂，局部极小点就越多。

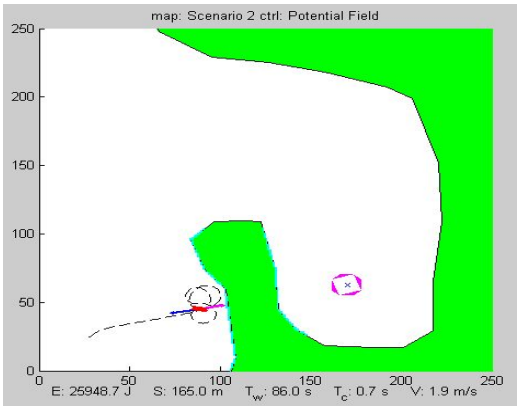

图 5.4 人工势场法陷入局部极小的典型情况

如上图 5.4 所示，机器人陷入局部极小。目标点的引力无法使机器人摆脱中间突起的障碍物，机器人陷入往复摆动的境地。

针对人工势场法的缺陷，人们提出了很多相关的改进算法，如虚拟力场法(VFF)是势场法的微小变种[67]，利用传感器搜集资料来创建一个全局地图。全局地图用二维的笛卡尔柱状网格来表示。这个柱状网格包含了障碍物被定位在环境中不同位置的可能性。然后计算地图中每个网格产生的斥力，但是要与网格的概率值相乘。

和谐势函数法(Harmonic Potential Function Fields)该算法来解决势场法产生的局部极小问题[71-72]。这些场叫做导航函数，这使得除了目标点之外没有局部极小点。但是和谐势函数法有一个弊端。那就是只有点状小车才能摆脱局部极小。而对非点状小车来说该算法不能保证找到全局解。该算法还需要关于所有障碍物的如形状，位置和运动等的完整信息，因此它不太适合用于动态避障。

5.3 人工势场法与 Bias-goal RRT 算法的结合应用

算法结合在路径规划的研究中的作用非常明显，目的是针对一种算法的缺点与不足而采用另外一种算法来进行弥补与改进。或者是利用另一种算法来提高已有算法的效率。

为了解决人工势场法的以上种种弊端与不足，本文提出了一种新的结合算法，即 RRT 算法与人工势场法的结合算法。

5.3.1 两种算法结合的主要思想

这两种算法的结合并不是偶然，因为势场法确实有着固有缺点。即缺乏全局特性，容易陷入局部极小点。所以势场法需要一个全局算法作为补充。同样 Bias-goal RRT 也不是完美的。作为一种全局算法 Bias-goal RRT 算法实时性差，所以也需要与人工势场法的结合。

所以在分析两种算法的优缺点的基础上，我们用 Bias-goal RRT 算法作为全局规划器，而局部规划采用人工势场法完成整个规划任务。

5.3.2 结合算法在复杂环境下的应用

正如上文提到的，在复杂环境中人工势场法可以适应动态环境，不断采集环境信息并可以进行实时避障等优点，也提到了当环境特殊时人工势场法容易陷入局部极小点的情况。所以在复杂环境中单纯的人工势场法已经不能满足机器人路径规划的需求。本节提出的结合算法是在人工势场法的基础上结合 Bias-goal RRT 算法。为的是在发扬人工势场法优势的基础上弥其不足，使其发挥更好的效果。

算法结合的方法是人工势场法进行局部规划，Bias-goal RRT 算法进行全局规划。当人工势场法陷入局部极小点的时候 Bias-goal RRT 算法进行全局规划。机器人沿 Bias-goal RRT 算法规划的全局路径运动。目的是使机器人脱离局部极小点，保证路径规划的顺利进行。具体规划过程如图 5.5 所示。

在每个周期的起始时刻，移动机器人实时地探测的障碍物信息，利用人工势场法进行局部路径规划。判断下一个时间段里机器人与障碍物的位置关系，然后再进行局部避障运动。

具体过程为：首先计算机器人与目标点之间的引力，然后计算机器人与障碍物之间的斥力，最后根据两者的合力方向与步长决定机器人在下一时刻的位置。当机器人运动一个步长之后判断其是否陷入局部极小点。如果没有陷入局部极小点，人工势场法继续运行。如果陷入了局部极小值点，那么 Bias-goal RRT 算法开始工作并为机器人提供全局路径，机器人将暂时利用 Bias-goal RRT 提供的全局路径运动并绕开局部极小值点。

当机器人脱离了局部极小值点之后，机器人将脱离 Bias-goal RRT 算法提供的全局路径并回到人工势场法的控制之中。并继续按人工势场法提供的路径运动。如果机器人再次陷入局部极小点，我们将重复上述过程。直到机器人最后到达目标点。

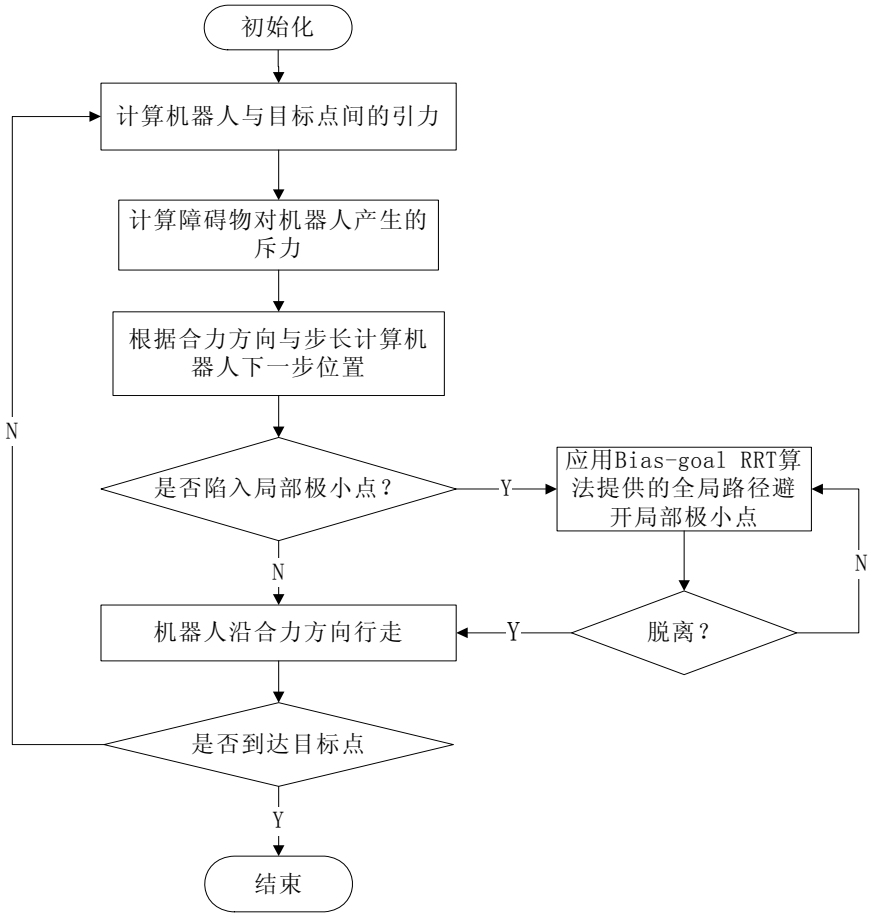

图 5.5 结合算法流程图

5.4 仿真试验与结果分析

本章成功实现了动态环境下将 Bias-goal RRT 算法与人工势场法结合运用到机器人路径规划中。进行了仿真实验。然后分析了仿真结果。

仿真实验中的电脑的主要参数与第三章与第四章相同。本章试验用 MATLAB 语言实现，版本为 R2007a 试验结果如图所示。其中白色区域为自由联通空间，绿色区域代表静态障碍物，黑色矩形为移动障碍物，红色矩形为移动机器人，粉色圆形为目标点。人工势场法的刷新时间为 0.5 秒，Bias-goal RRT 的刷新时间为 10 秒。

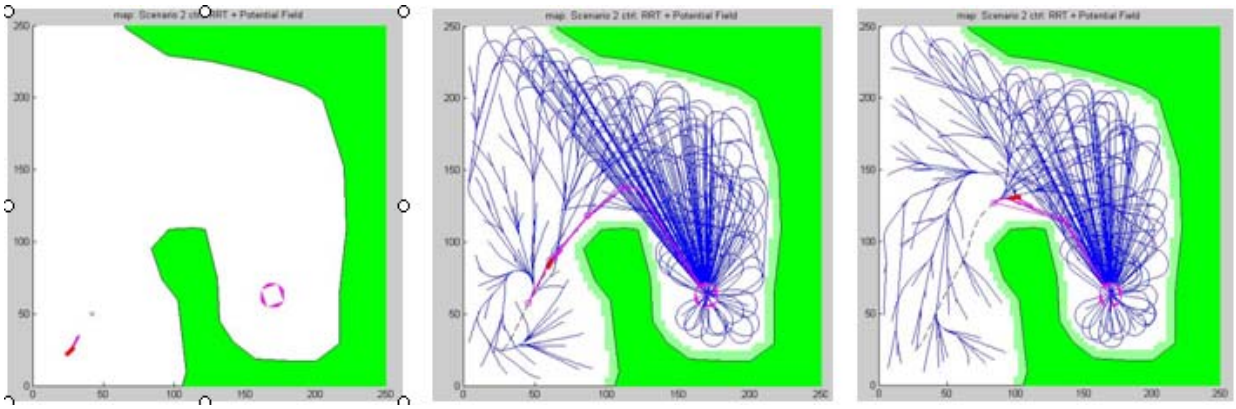

图 5.6 结合算法仿真

图 5.6 表示应用结合算法移动机器人从起始位置搜索路径，并且在搜索树遇到静态障碍物的时候自动调整路径并且绕过移动障碍物的过程。可以看到一共机器人也受到人工势场法的作用，因为我们可以看出来移动机器人并没有完全按照 Bias-goal RRT 算法所提供的路径进行跟踪。这说明在结合算法中机器人利用人工势场法作为局部规划算法，受到障碍物的斥力与障碍物的吸引力。当势场法陷入局部极小而失效时，机器人则利用 Bias-goal RRT 作为全局路径规划算法使自己摆脱局部极小的处境然后顺利到达目标点。

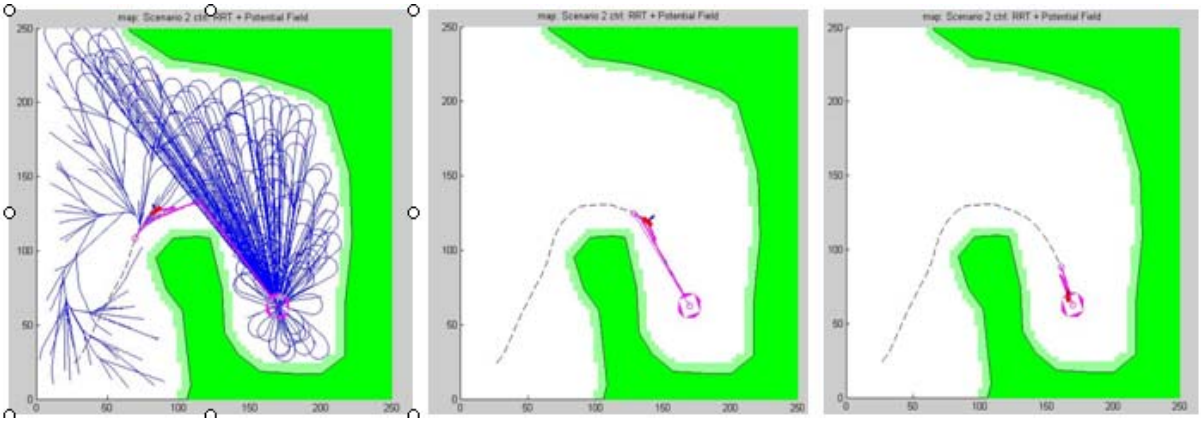

图 5.7 结合算法仿真

图 5.7 表示该仿真环境下移动机器人在摆脱局部极小点的情况下逐渐跟踪规划好的路径，并最终到达目标点的过程。

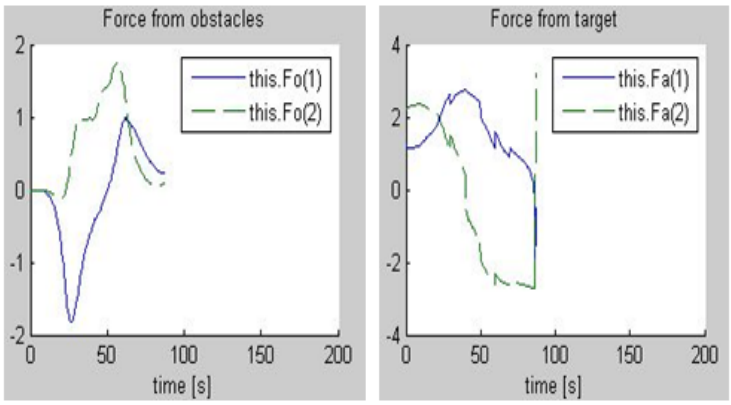

图 5.8 机器人受到的斥力与引力变化曲线

图 5.8 表示移动机器人所受来自障碍物的斥力和来自目标点的引力的结果图。可以看出理论值和实际值并不重合，这说明在 Bias-goal RRT 算法做为全局规划算法的情况下，移动机器人并没有完全按照人工势场法的规划路径前进。它受到了全局路径规划法的影响，所以出现曲线不重合的情形。

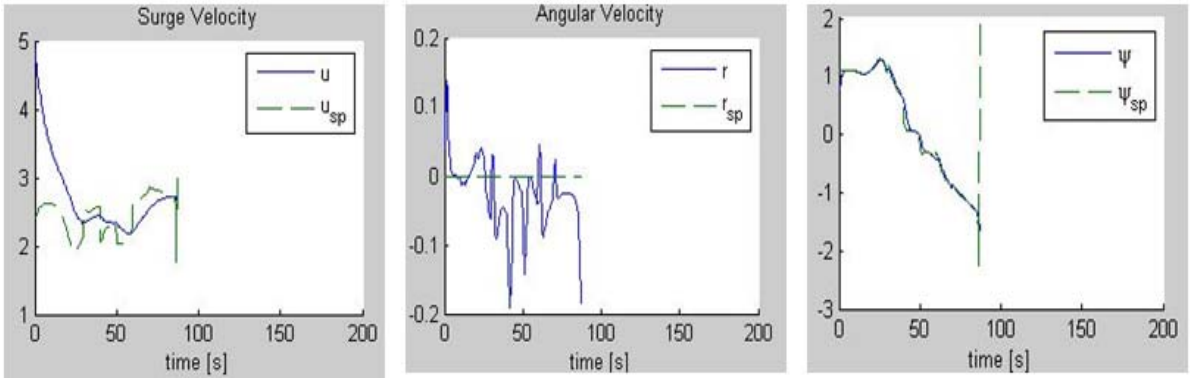

图 5.9 机器人的线速度、角速度与方向角变化曲线

图 5. 9 所示表示移动机器人从起始点运动到目标点的过程中首向角，线速度，角速度的变化曲线。可以看出由于机器人处在静态环境中没有受到环境突变的影响。所以机器人的首向角 Ψ 并没有与输入首向角 Ψ_{sp} 形成较大的偏差，基本上处于重合的状态。而机器人的输入线速度 u_{sp} 并不是一条直线，这是因为这是按照人工势场法为基准进行速度输入的，而人工势场法是局部算法，会出现速度的大小不同。而实际线速度 u 和输入线速度 u_{sp} 在仿真开始阶段出入较大是因为：为了避开局部极小值点，机器人采用 Bias-goal RRT 作为全局规划算法从而绕开局部极小点。而在仿真的后半段实际线速度 u 和输入线速度 u_{sp} 出入不大是因为：机器人已经绕开了局部极小点，Bias-goal RRT 算法已经完成了其全局路径规划的功能，进而在结合算法的作用下机器人最终到达目标点。

人工势场的缺点在对全局环境的探测不足，使机器人经常因为局部极小点而前后摆

动、徘徊不前或静止。因此结合 Bias-goal RRT 算法使得人工势场法能够脱离局部极小点。而且可以使机器人迅速的找到目标点。所以两者的结合非常有意义。Bias-goal RRT 算法概率完整，有很强的全局搜索能力。且它因为它是有偏于目标的。故搜索速度较快。在没有陷入局部极小点的时候，机器人一直利用人工势场法提供的局部路径运动。然后在重要时刻进行全局规划控制，绕过局部极小点并跟踪到达目标。仿真结果表明该算法是有效的。

5.5 本章小结

本章研究了 Bias-goal RRT 算法与人工势场法的结合应用。该章主要简单介绍了人工势场法的基本原理和基本步骤，然后分析人工势场法的主要缺陷和一般的解决方案。进而提出 Bias-goal RRT 与势场法相结合的方法来解决势场法的缺陷并给出试验结果验证结合算法的有效性。

结　论

(1)总结

1)改进传统的概率平均的快速随机搜索树算法，引入参数 Bias-goal 使搜索更加具有选择性。算法的收敛速度明显加快，机器人的路径搜索具有更强的目标性。

2)本文实现并比较了两种基于快速扩展随机树算法的路径规划器。一种是基本的快速扩展随机树规划器。仿真结果表明该规划器有积极搜索环境中的未知空间与概率完备等优点。另一种是偏向目标型快速扩展随机树规划器。它是基本快速扩展随机树算法的改进型。它针对基本快速扩展随机树搜索过于平均，用时较长等缺点进行改进。引入 Bias-goal 参数，使搜索具有一定的倾向性，提高了搜索的速度与效率。

3)根据非完整系统与动态环境的特性，将适用于非完整系统的快速扩展随机树改进算法即偏向目标型快速扩展随机树算法应用非完整系统中。应用 Bias-goal RRT 算法来引导机器人躲避动态障碍物。在每个周期中都用 Bias-goal RRT 算法来提供可以躲避移动障碍物的路径。这种方法提高了避障的效率，缩短了动态避障的时间。较以往的方法的表现有了显著的进步。

4)本文实现了人工势场法与快速扩展随机树算法的结合。结合算法主要针对人工势场法容易陷入局部极小等缺点，在结合了人工势场法局部搜索优点的基础上结合快速扩展随机树算法进行全局规划。避免了可能产生的局部极小，新的结合算法将选择性参数 Bias-goal 引入 RRT 算法。使得机器人的运动更加有偏向目标的趋势，缩短了机器人到达目标点的时间。

(2)展望

在本人对移动机器人运动规划的学习与研究中，我认为在不确定的条件下，动态的环境中的路径规划是值得研究的主要方向。同样非完整系统中的路径规划也是路径规划研究中非常重要的组成部分。这两个方向的研究对移动机器人的发展有着至关重要的意义。因为它们能够提高机器人执行各种复杂任务的能力，而且能够给人类以启迪与帮助，促进生成新的智能算法。所以对于机器人的运动规划的研究必须矢志不移的进行。然而，在研究过程中的种种局限，本文还有很多问题需要人们进一步加以探索：

1)本文对于路径规划算法的研究只进展到了计算机仿真层面，并没有和实际的移动机器人相结合，也就是理论没有应用于实际，只从理论上的研究不足以解决实际中的问题。所以希望后续的学者能够将算法应用于实际。

2)机器人建模时没有考虑机器人的动力学层面，而只有运动学层面的建模仿真实验

结果输出是相应的首向与速率。所以进一步的研究要对机器人的动力学约束作更多的考虑，以适应现实世界中的真实环境。

3)本文的路径规划环境主要是二维静态与动态的环境。而现实中的环境通常是动态的三维环境，所以目前的研究领域过于局限，因此三维环境中的路径规划研究是将来研究的另一个重要方向。

参考文献

[1] 王耀南. 机器人智能控制工程. 北京：科学出版社. 2004 年：5−7 页

[2] 熊宁，邵世煌，王直杰. 多变地形下机器人路径规划.控制与决策. 1995，10(5)：1-12 页

[3] 周明，孙树栋，彭炎午. 基于遗传模拟退火算法的机器人路径规划. 控制与决策. 1998，19(1)：118-120 页

[4] 李枚毅，蔡自兴. 改进的进化编程及其在机器人路径规划中的应用. 机器人. 2000，22(6)：490-494 页

[5] 张松灿. 移动机器人的智能导航和跟踪控制研究. 合肥王业大学硕士论文. 2005 年：3-10 页

[6] Crowley J L. Intelligent Robotics System-IRS’93.Robotics and Autonomous Systems.1994(12):109-111P

[7] 乔永兴. 自主式移动机器人的路径规划. 广西大学硕士论文. 2003 年：4-6 页

[8] 李磊，叶涛，谭民等. 移动机器人技术研究现状与未来. 机器人. 2002，24(5)：475-480 页

[9] 焦立男. 地面移动机器人运动规划与运动协调的若干算法研究. 南京理工大学博士论文. 2007 年：5-9 页

[10] Oivind Loe. Collision Avoidance Concepts for Marine Surface Craft. Project report. Norwegian University. 2007(12):5-7P

[11] James Colito. Autonomous mission planning and execution for unmanned surface vehicles in compliance with the marine rules of the road. Master’s thesis. University of Washington． 2007: 23-30P

[12] Spawar. http://www.spawar.navy.mil/robots/surface/usv. Accessed. 2007.12.10

[13] C Alexopoulos,P M Griffin. Path planning for a mobile robot. IEEE Trans on Sys Man Cybern. 1992,22(2):318-322P

[14] Yahja A, Singh S Stentz. An efficient on-line path planner for outdoor mobile robot. Robotics and Autonomous systems. 2000,32(2-3):129-143P

[15] 陈刚，沈林成. 复杂环境下路径规划问题的遗传路径规划方法. 机器人. 2001.23(1)：40-50 页

[16] 孙秀云. 基于遗传模拟退火算法的机器人全局路径规划. 微电子力学与计算机. 2005，22(6)：98-100 页

[17] 李磊，叶涛. 移动机器人技术研究现状与未来. 机器人. 2002，24(5)：475-480 页

[18] 于红斌，李孝全. 基于栅格法的机器人快速路径规划. 微电子学与计算机. 2005，22(6)：98-100 页

[19] Pere. Automatic planning of manipulator movements. IEEE Trans on Sys Man Cybern. 1981,11(11):681-698P

[20] J C Latombe. Robot Motion Planning. Kluwer, Norwell. MA. 1991

[21] 孙增圻. 智能控制理论与技术. 清华大学出版社，1997 年：34-37 页

[22] Khatib. Real-time obstacle for manipulators and mobile robot. The International Journal of Robotic Research. 1986,(1):90-98P

[23] J.C. Fraile, C.H. Wang,C.J.J. Paredis, P.K. Khosla. Agent-based control and planning of a multiple manipulator assembly system. IEEE Int. Conf. on Robotics and Automatation,1999,(2):1219-1225P

[24] J. Barraquand, J. C. Latombe. Robot Motion Planning: A Distributed Representation Approach. Int. Journal of Robotics Research,1991,10(6):628-649P

[25] Barraquand J, Langlois B, Latombe J C. Robot Motion Planning with many degrees of freedom and dynamic constraints. In: Miura H, Arimoto S, Robotics Research. Cambridge, MA, MIT Press,1990,435-440P

[26] Min Gyu Park, Jae Hyun Jeon, Min Cheol Lee. Obstacle avoidance for mobile robots using artificial potential field approach with simulated annealing. IEEE International Symposium on Industrial Electronics, Pusan, Korea. 2001,(3):1530-1535P

[27] 张培艳，吕恬生. 基于模拟退火-人工势场法的足球机器人路径规划研究. 机械科学与技术，2003；22(4)：547-555 页

[28] Jung H K, Hong S K, Chon H C, Yoo Y D. Path planning and obstacle avoidance of robot using new wall-following algorithm. Proc Asian Control Conf. Shanghai, 2000:1628-1633P

[29] 张明开，李龙澍. 关于人工势场法局部极小问题的一种解决方法. 计算机拘束与发展，2007，17(5)：137-139 页

[30] Jacoby Larson,Michael Bruch,and John Ebken. Autonomous navigation and obstacle avoidance for unmanned surface vehicles．Proceedings of the International Society for Optical Engineering. 2006: 36-47P

[31] Barraquand J, Langlois B, Latombe J C. Numerical Potential field techniques for robot path planning. IEEE Trans on system, Man and Cybemetics. 1992,22(2):224-241P

[32] 禹建丽，孙增圻. 一种快速神经网络规划算法. 机器人，2001，23(3)：201-205 页

[33] 王梅. 多关节自治机器人系统分布式协作运动规划方法研究. 杭州，浙江大学. 博士论文. 2008 年：55-58 页

[34] Nelson H C, Yung Cang Ye. An intelligent mobile vehicle navigator based on fuzzy logic and reinforcement learning. IEEE Trans on Sys Man and Cybern. Part B: Cyberneics,1999,29(2):314-320P

[35] Fierro R, Lewis F L. Control of nonholonomic mobile robot using neural networks. IEEE Trans on Neural Networks. 1998,9(4):589-600P

[36] Gerke M. Genetic path planning for mobile robots. Proc of American control conference. San Diego, CA, USA, 1999:596-601P

[37] J. P. Launond, S. Sekhavat, F. Lamiraux. Guidelines in nonholonomic mobile planning for mobile robots. In J. P. Laumond, enditor, Springer-Verlag, Berlin,1998: 1-53P

[38] J. P. Laumond, P. E. Jacobs, M. Taix et al. A motion planner for nonholonomic mobile robots. IEEE Transactions on Robotics and Automations. 1999: 577-593P

[39] 朱金寿，朱琪，王进等. 最小费用最大流算法在路径规划中的应用. 武汉理工大学学报（交通科学与工程版）. 2001，23(2)：172-174 页

[40] Chsia T. System identification: Least-squares methods. Lexington Books, Lexington, MA,1977: 124-129P

[41] 吴文清，刘嘉，崔世钢. 基于方向随机的障碍物实时预测的机器人路径规划. 机器人技术与应用，2003，2：37-40 页

[42] 吴勉，吉小宁，尹朝万. 基于修正的最小均方误差分类算法的移动机器人路径规划. 机器人技术与应用. 1997，19(3)：52-55 页

[43] 董立志，孙茂相，朱枫等. 基于实时障碍物预测的机器人运动规划. 机器人. 2000，22(2)：12-16 页

[44] LaValle S M. Planning Algorithms. University of Illinois. 2004: 256-268P

[45] S.M. LaValle. Rapidly-exploring Random Trees: A New Tool for Path Planning. Iowa State University. 1998: 34-46P

[46] LaValle S M, Kuffner J. Rapidly-exploring Random Trees: Progress and Prospects. Int'l Workshop on Algorithmic Foundations of Robotics(WAFR). 2000: 55-59P

[47] P. Cheng, Z. Shen, S.M. Lavalle. RRT-based trajectory design for autonomous automobiles and spacecraft． Archives of Control Sciences． 2001,11(3-4):167-194P

[48] S.M. LaValle,J.J. Kuffner． Rapidly-exploring random tree: Progress and Prospects. In B.R. Donald, K.M. Lynch, and D. Rus, editors, Algorithmic and Computational Robotics: New Directions, A K Peters. Wellesley, MA, 2001: 293-308P

[49] Peng Cheng, Zuojun Shen, Steven M. LaValle. Using Randomization to Find and Optimize Feasible Trajectories for Nonlinear Systems. Proc.38th Annual Allerton Conference on Communication, Control, and Computing. 2000: 43-48P

[50] S.M. LaValle. From dynamic programming to RRTs: Algorithmic design of feasible trajectories. In A. Bicchi, H.I. Christensen, and D. Prattichizzo, editors, Control Problems in Robotics. Springer-Verlag. Berlin. 2002: 19-37P

[51] 徐明亮，卢红星，王琬. OpenGL 游戏编程. 机械工业出版社. 北京. 2008 年：4-10 页

[52] S. R. Lindemann, S. M. LaValle.Incremantal low-discrepancy lattice methods for motion planning. In Proc. IEEE International Conference on Robotics and Automation. 2003: 45-68P

[53] M. S. Branicky, S. M. LaValle, L. Yang． Quasi-randomized path planning. In Proc. IEEE Int'l Conf. on Robotics and Automation． 2001: 1481-1487P

[54] J. P. Laumond, S. Sekhavat, F. Lamiraux. Guidelines in nonholonomic mobile planning for mobile robots. In J. P. Laumond. Editor. Robot motion planning and control. Springer-Verlag. Berlin. 1998: 1-53P

[55] J. P. Laumond. Feasible trajectories for mobile robots with kinematics and environmental constrints. In Proc. Int. Conf. On Intelligent Autonomous Systems. 1986: 346-354P

[56] J. P. Laumond, P. E. Jacobs, M. Taix. A motion planner for nonholonomic mobile robots. IEEE Transactons on Robotics and Automations. 1999: 577-593P

[57] J. A. Reeds, L. A. Shepp. Optimal paths for a car that goes forward and backwards. Pacific Journal of Mathematics. 1990,145(2): 367-398P

[58] S. Sekhavat, P. Svestka, J. P. Laumond etc. Multi-level path planning for nonholonomic robots using semi-holonomic subsystems. In J. P. Laumond et al. editors. Algorithms for robotic motion and manipulation:1996 workshop on the algorithmic foundations of robotics.A.K.Peters,Wellesley,MA,1997: 79-96P

[59] J. Barraquand, J. C. Latombe. Robot motion planning: A distributed representation approach. International Journal of Robotics Research.1991,10(6): 628-649P

[60] B. Donald, P. Xavier, J. Canny etc. Kinodynamic motion planning.Journal of the ACM. 1993,40(5): 1048-1066P

[61] R. Kindel, D. Hsu, J. C. Latombe etc. Kinodynamic motion planning amidst moving obstacles. In Proc. IEEE International Conference on Robotics and Automation. 2000: 537-543P

[62] Chiew Seon Tan, Robert Sutton, and John Chudley. An incremental stochastic motion planning technique for autonomous underwater vehicles. Proceedings of IFAC Control Applications in Marine Systems Conference. 2004: 483-488P

[63] Emilio Frazzoli. Quasi-random algorithms for real-time spacecraft motion planning and coordination. Acta Astronautica. 2003,53:485-495P

[64] James J. Kuffner and Steven M. LaValle. Rrt-connect: An efficient approach to single query path planning. IEEE International Conference on Robotics and Automation. 2000: 995-1001P

[65] Mark B. Miliam. Real-Time optimal trajectory generation for constrained dynamical systems. PhD Thesis. California Institute of Technology. Pasedena. CA. 2003: 56-79P

[66] Steven. LaValle. Planning Algorithms. University of Illinois. 2006: 715-745P

[67] Y. Koren and J. Borenstein. Analysis of control methods for mobile robot obstacle avoidance. Proceedings of the IEEE International Workshop on Intelligent Motion Control. 1990: 457–462P

[68] J. Barraquand, B. Langois, J .C. Latombe. Potential fields for robot motion planning. IEEE Transactions on System． Man and Cybernetics. 1992,22(2): 224-241P.

[69] J.Barraquand,J.C.Latombe. Robot motion planning: A distributed representation approach. International journal of robotics research. 1991,10: 628-649P

[70] O. Khatib. Real-time obstacle avoidance for manipulators and mobile robots. Proceedings of the ICRA’85. 1985: 500–505P

[71] J. Guldner, V. I. Utkin, H. Hashimoto, and F. Harishima. Obstacle avoidance in R^n based on artificial harmonic potential fields. Proceedings of the ICRA’95. 1995: 51-56P

[72] Jin-Oh Kim and Pradeep K.Khosla. Real-time obstacle avoidance using harmonic potential functions. IEEE Transactions on Robotics and Automation. 1992,12: 338-349P

攻读硕士学位期间发表的论文和取得的科研成果

一、攻读硕士学位期间发表的论文（含已录用待发表的）

二、攻读硕士学位期间取得的科研成果

致谢

值此论文即将完成之际，向我的导师王能建教授表达我最诚挚的谢意！感谢导师王能建教授对本人硕士论文的悉心指导，感谢导师对本人学习上的指导、生活上的关心和精神上的鼓励。导师严谨求实、锐意创新的治学态度、渊博的知识、敏捷的思维、开阔的视野和始终饱满的工作热情使我受益匪浅！

感谢邱长华老师，张家泰老师对我孜孜不倦的教诲，他们的学者气质与大家风范深深的影响了我，让我对做人与做事的方法与态度有了新的认识，感谢二位对我的帮助。

感谢史冬岩老师、薛开老师、祝海涛老师、朱世范老师、钟宇光老师、潘文林老师、曲东越老师、刘钦辉老师、展勇老师、刘崇老师、王伟琳老师以及滕晓艳老师在我硕士学习期间对我孜孜不倦的教导，感谢他们在我硕士课题的选题、方案的制定及改进方面对我的大力指导和无私帮助！

感谢张德福博士、周立杰博士在我硕士课题研究过程中给与的倾力帮助！感谢李立新同学、马登云同学、潘林同学、陆洋同学、王少帅同学、周金阳同学、在我论文撰写过程中给与的各种帮助！感谢韩喜、刘元庆、刘宏博等师弟、师妹们对我工作的支持和帮助！

感谢我伟大的父母亲，感谢他们对我的养育之恩，感谢他们对我的理解、关心和鼓励！感谢我的朋友们对我的支持和鼓励。

我非常高兴能够成为哈尔滨工程大学 CIMS 研究所的一名硕士研究生，在这里我感受到了实验室快速发展的澎湃动力，感受到了师生之间的深深情意，感受到了踏实认真、锐意创新、追求卓越的研究精神，这一切是如此的美好，我定会终生难忘，永记在心！