

RoboMaster 冬令营技术报告（挑战赛）

1、项目规划

1.1 项目规划

两题对算法要求基本上搜索加上优化，机械上而言，连连看可以复用一部分结构，考虑到机械工作量问题，本组选择连连看作为课题，规则要求第一人称视角，这将会对操作手造成巨大影响，因此算法、机械需要考虑上这一点进行设计。

Day7: 制作连连看机器

9: 00-10: 00 基本方案

10: 00-14:00 完成基本结构

14: 00-21: 00 装车

21:00-5:00 理线，调试

任务完成情况：任务完成。

Day8: 比赛

1.2 方案设计

我们初始的方案：通过一圈的气动推杆或电推杆伸缩夹子（像个小水车那样），把所有箱子收齐，然后沿着 123456 行走，到各个位置把相应的箱子转到底部，放在位置上。这个方案可以直接忽略复杂的算法计算，通过结构解决问题。但这个方案，通过高度，体积计算和难度估计，被直接推掉了。

正式方案：

此机器人最终版本的设计亮点在于没有爪子对块进行抓取，而是直接将块吞入肚内，省去了抓块儿的时间。

最初的版本我们打算将机器人用爪子夹取起来然后可以直接从块上越过，省去一大部分时间，但这种方法只适用于半自动或全自动的方案，对于手动来说，这个方案就略慢，加上本身是第一人称视角，就是的这种方案有了极高的难度。

在比赛前几个小时，我们放弃了整辆车的上半部分，这保留了底盘和框架，这样的效率其实也很高。

1.3 理论计算

1.准备时间有三分钟，使用文件输入大致需要 10s（从接收文件到输入），放置机器时间大概 30s，因此算法结果必须要在 120s 内输出，因此算法迭代深度不超过 7 层，机器遍历全场需要 30s，取放方块大致需要 10s，在此基础上 8 分钟能完成 12 块，若算上失误大致能完成 9 块，因此在设计算法时不能考虑整体最优的解法，而应该考虑当前情况下对自己最优的解法

2.臂的抬升：机械臂首先要能高于一个箱子（高 20cm），并且夹取抬升后要能高于一个箱子，再给一个容错的距离，，滑轨至少需要 38cm。

3.臂的夹取：用两个连杆连接大扭矩舵机向外撑开。舵机扭矩 $25\text{kg}\cdot\text{cm}$ ，连杆 5cm ，与箱子接触约 45° ，因此单边能提供约 53.3N 的法向力。两边加起来就有 106.6N 的法向力。由于箱子本身的粗糙程度较高，与夹子的摩擦系数（虽然没有具体数值）也很高，夹子完全能将箱子抬起。

4.车架大小：能让箱子轻松进入内部。车架宽度至少大于箱子对角线的长度 28.3cm ，因此我们将车架内部宽度定为 30cm 。为了让机械臂抬起箱子后还能直接从上方穿过箱子，高度至少 40cm ，这里我们保守地定为 50cm 。

2、研发历程

2.1 方案预演

由于时间原因，我们只进行了爪子的实验，我们装好了一个简易的爪子（事实证明没有时间再装一个了）用手将舵机上链接的爪子与箱子接触，是箱子发生形变，发现这时候提爪子就已经可以抓起整个箱子了。于是直接用了这种方案。

2.2 方案的细化

我们对于主体结构采用了 2020 的铝型材，其原因是结实，好连接（连接件和 T 型螺母的短缺当时不在我们的考虑范围之内），并且调整起来会更方便。对于主体框架和轮子的连接方式，我们用碳板

做了连接件，如果不用碳板会导致整个结构的不稳定。麦轮的放置我们采用了外八字方法以提高机器人的稳定性。

对于爪子的选择，我们认为从外方抓取会严重影响我们整个车的空间利用，因此我们决定利用箱子上的洞从内部抓取，经过实验发现可行。我们的爪子因为没有是时间进行 3 D 打印，直接用了 m a k e b l o c k 的标准件进行搭建，一共有三版，第一版是一根梁上装有一个绑了橡皮筋的长螺丝，但这样会使箱子因为手里不均衡而发生倾斜于是第二版改成了短螺丝，发现有点虚，于是第三版改成了一个三角形的片状结构，利用斜边对块内施加一个线状的力，可以完美解决问题。（但是最后没有用上……）

抬升机构采用了 m a k e b l o c k 步进电机，一是没有其他好的电机可以用了，二是考虑到嵌入式同学时间有限，三是方便安装。

走线我们就打算直接从型材内部走，这样做其实没毛病，只是苦了嵌入式同学焊了大半夜的线……

为了方便进块，我们用复写板做了两个向外开的结构，引导方块进入内部。

2.3 算法

1. 课题选择：选择课题时算法考虑了机械的工作量问题，因此考虑复用第一阶段结构以完成连连看任务，而选择放弃华容道任务。

2. 模型构建：连连看问题中，方块移动到 RFID 位置的路程是固定的，因此求最优解即为求 RFID 到下一块方块的距离之和的最小值，但是本题类似旅行商问题（NP 问题），在多项式时间复杂度内无解，因此考虑进行搜索。

3. 算法选择：被考虑过的主算法有：贪心，深度优先。以下为算法比较：(1) 贪心，贪心在运算过程中有不同贪心策略，如：选择最近的方块放置、选择周围四块中最近的方块放置……但是贪心存在局部最优解的问题，探测步数较少。(2) 深度优先搜索，深搜算法能得出最佳方案，但是深搜算法计算量大，计算时间长，遍历整个图是不现实的。

经过测试，在第一人称视角下，估算 8 分钟内能完成数量大概为 7-8 块。因此考虑结合以上两种算法，对当前点往下探测一定步数，然后贪心选择花费最小的方案。计算机 1s 内能运行的计算次数大概是 10^8 (CPU: 赛扬双核 3.0 GHz)，经过估算 3 分钟内深度搜索 5-7 步最为合适，因此选定 6 步搜索预测，然后进行贪心。

在以上基础上，考虑通过某些优化、剪枝，降低运算量以及保证一定精度，因此选定了两个辅助算法：人工势场、A*。人工势场法构建一个势场图，势能公式如下：

$$W = \text{Pre}[i-1][j] + \text{Pre}[i+1][j] + \text{Pre}[i][j-1] + \text{Pre}[i][j+1]$$

$\text{Pre}[i][j]$ 表示 $[i][j]$ 处方块往前预测所能得分的花费

构建这个人工势场目的在估计这个方块完成的价值，当 W 越小表示在这个方块搬运其他方块的价值越高，这个场的估值作为 A* 的估计值

进行运算。（两个辅助算法结合后未证明、优化至完全可行，因此正式比赛时仅使用了两个主算法）

4. 算法效果：输入后 1 分钟内能得出结果，算法输出结果显示，需要就近完成只需要移动一步的方块然后进行其他考虑。

5. 期望：完成本算法未完成部分，并根据第一版算法得出结果尝试优化。

2.4 联调

在调试过程中，我们的步进电机带动法兰盘会滑轴，导致机械臂不能正常伸缩。我们尝试很多方法，如拧紧顶丝，加个顶死，把法兰盘焊在轴上。最后把线缠在轴上，并改了控制方式就解决了问题。

3、感想感悟

3.1 技术收获

1.我对 S W 的模拟系统有了更进一步的了解，我们运用这方米娜的知识对我们的机器人进行了应力分析，这样可以了解它的刚性，强度等方面的参数，帮助我们设计加工机器人。我倒没觉得比别的队有多棒，但是其实是让我们相较于其他队对我们的机器人更有信心，更有把握。

我们组机器人的很多地方其实都很值得我骄傲，比如说原型设计

中的用爪子提块，从块上越过来争取时间这个想法，即是说到最终机器人的状态，直接从块上开过去进行收取其实也是一种非常省时间的方案。

但现实和理想的差距其实就是这么小，哪些你觉得应该可以出成绩的结构到了事实比赛中就会出现各种各样的问题，再随着规则的改变，操作手的原因（是我是我），比赛时各种不稳定因素等，导致机器人无法按照理想中的那样去运行。

我们的爪子其实还是比较好的设计（我认为），但仅限于全自动或半自动模式，并且对机器人的稳定性要求非常高，这极大的挑战了程序员的功力，而我作为机械组的，明显高估了程序员的水平。这是我在“一天做出机器人”这样的项目中不应该做的，时间太紧了。

我们最后的处理方法是放弃了整辆车的上半部分，保留下半部分用来推块。有时候，作为一名设计者，你很难放弃精心设计出来的东西，但该放手时还得放手，得分要紧，设计其次，于是我们把中部主梁进行了位置调整，以满足推块的需求。

如果我在进行一次设计，我会坐一辆与最终版类似，但体积更小，机动性更强，有一个舵机再开口控制一根大梁旋转的小型机器人，他会很完美的解决我们所有在场上遇到的问题。