DOI：10.13973/j.cnki.robot.2017.0008

基于改进 RRT 算法的 RoboCup 机器人动态路径规划

刘成菊，韩俊强，安　康
（同济大学电子与信息工程学院，上海　201804）

摘　要：针对足球机器人在场上采用反应式方法避障时存在的速度慢、效果差的问题，采用改进的快速扩展随机树（RRT）算法设计了一种能够适应机器人足球赛场动态移动障碍环境的路径规划器．首先，引入基本的快速扩展随机树算法，针对其随机性强、路径过长的缺点，提出了以一定概率选择目标点、增加引力分量以及路径平滑处理等改进方式；引入路径缓存区以及动态扩展随机树的方法来解决移动障碍物环境中的路径规划问题．复杂障碍物环境中的仿真实验表明，改进的规划路径长度比基本快速扩展随机树算法所得路径缩短约 20%．最终将策略移植到实体 NAO 机器人上参加 RoboCup 比赛，取得世界八强的成绩．

关键词：仿人机器人；路径规划；快速扩展随机树；动态环境
中图分类号：TP242.6　　**文献标识码**：A　　**文章编号**：1002-0446(2017)-01-0008-08

Dynamic Path Planning Based on an Improved RRT Algorithm for RoboCup Robot

LIU Chengju，HAN Junqiang，AN Kang
(*School of Electronics and Information Engineering*, *Tongji University*, *Shanghai* 201804, *China*)

Abstract: To solve the problem of slow speed and low efficiency of the soccer robots using reactive obstacle avoidance strategy at soccer game, an improved rapidly-exploring random tree (RRT) algorithm is adopted, and a path planner based on this algorithm is designed to adapt to the dynamic moving obstacles environment in the robot soccer field. Firstly, the basic RRT algorithm is introduced, and some improvement ways to solve the disadvantages of strong randomness and long path length are proposed, such as probabilistically selecting the target node, adding the gravity component, and smoothing the path. The path planning in dynamic moving obstacles environment is solved by using the methods of path buffer and dynamic-expanding random tree. The simulation in complex obstacle environment shows that the path generated by the improved algorithm is about 20% shorter than the path generated by the basic RRT algorithm. Finally, the strategy is applied to the physical NAO robot to participate in the RoboCup competition and the top eight grade is achieved.

Keywords: humanoid robot; path planning; rapidly-exploring random tree; dynamic environment

1　引言（Introduction）

实时动态环境下的路径规划是机器人足球中很重要的一部分，能够使机器人在赛场上避免与对方或者己方机器人发生碰撞，从而节省起立的时间，减少对于机器人本身的损伤，能够以更快的时间到达球，赢得对球的控制权．路径规划算法分为全局与局部路径规划算法两类，这两类方法各有优点，同时也都存在弊端．机器人足球场上的路径规划较为合理的方式是结合这两种方法的优点，将局部路径规划的思想结合到全局路径规划之中，这样既能够搜索出全局路径，也能适应实时变化的环境．

RRT 算法 [1-4] 是一个在路径规划中应用比较广泛的方法，但是其固有的随机性决定了这种方法不能直接应用到路径规划之中，不论是在静态环境下还是动态移动障碍物环境下．静态环境中利用该方法的改进方法进行规划路径已经较成熟 [5-6]，但是将该方法应用到机器人足球世界杯的动态环境中还很少．本文利用 RRT 算法在全局搜索中的优势，将一些局部搜索方法的思想加入到基本 RRT 算法中，改善其随机性带来的弊端．RRT 算法的优势还在于其在算法结构中提供了很多接口，可以根据实际情况采取不同的实现方法，在不同的层面上对算法作改进 [7-10]．本文中首先在接口实现上作不同的改进，然后尝试不同的随机树的结构观察其效果．在静态环境仿真通过的基础上，针对动态移动障碍物环境，作了进一步的改进以适应环境的需要．基于 Matlab 和 SimRobot 对提出的算法进行了验证，并

基金项目：国家自然科学基金（61673300，61573260），中央高校基本科研业务费专项
通信作者：安康，ankang526@foxmail.com　收稿/录用/修回：2016-11-01/2016-12-05/2016-12-09

将该算法应用到标准平台组的 RoboCup 比赛中验证了该算法的优点与局限性，同时提出进一步优化的方向.

2　改进的 RRT 算法（The improved RRT algorithm）

2.1　增加引力分量

基本 RRT 算法具有很强的随机性，为了解决这个问题，将人工势场法[11-14] 中引力的思想引入到 RRT 算法中，引导随机树朝着目标方向生长，无需对全局环境进行准确的建模，大大减少规划时间，对提高算法实时性有很大的帮助，同时对随机树的生长方向作了修正，改进路径不光滑的缺点，避免了产生局部极小，使算法在规划路径方面的能力得到很大的提高.

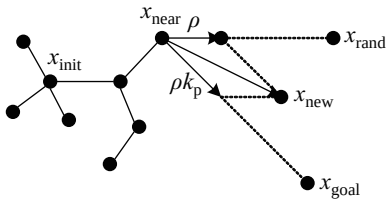


图 1　加入引力思想构建 RRT 图示

Fig.1　RRT construction with gravitation thought

如图 1 所示，这种改进的核心思想是在路径中的每一个节点 n 都引入一个目标引力函数 $\boldsymbol{G}(n)$，其中 x_{rand} 表示随机节点，x_{goal} 表示路径终点，ρ 表示搜索步长，此处的节点 n 表示由起点 x_{init} 向外扩展的第 n 个 x_{new} 节点，可表示为

$$\boldsymbol{F}(n)=\boldsymbol{R}(n)+\boldsymbol{G}(n) \tag{1}$$

其中，$\boldsymbol{F}(n)$ 表示为从节点 n 到新节点的生长指导函数，$\boldsymbol{R}(n)$ 表示节点 n 的随机生长函数，$\boldsymbol{G}(n)$ 为目标引力函数.

由引力势能

$$U=\frac{1}{2}\cdot k_{\text{p}}\cdot\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\|^2 \tag{2}$$

得到目标点与最近节点间的引力：

$$G=k_{\text{p}}\cdot\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\| \tag{3}$$

其中，ρ 为随机树的生长步长，k_{p} 为引力系数，$\boldsymbol{x}_{\text{goal}}$ 为机器人目标位置向量，$\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\|$ 表示节点向量 $\boldsymbol{x}_{\text{near}}$ 与目标点向量 $\boldsymbol{x}_{\text{goal}}$ 几何距离的绝对值. 根据式 (3) 构造出节点 n 处的目标引力函数 $\boldsymbol{G}(n)$ 为

$$\boldsymbol{G}(n)=\rho\cdot k_{\text{p}}\cdot\frac{\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\|} \tag{4}$$

在通过 RRT 算法增加新的叶节点时，目标引力函数会通过计算每个节点到目标的引力来影响新节点的选取引导随机树向目标方向生长.

由基本 RRT 算法中新节点 $\boldsymbol{x}_{\text{new}}$ 的构造公式[1-2] 可得：

$$\boldsymbol{R}(n)=\rho\cdot\frac{\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}\|} \tag{5}$$

将式 (4)、(5) 代入式 (1) 中可以得到：

$$\boldsymbol{F}(n)=\rho\cdot\frac{\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}\|}+\rho\cdot k_{\text{p}}\cdot\frac{\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\|} \tag{6}$$

根据式 (6) 得到引入引力思想后新节点的生成公式：

$$\boldsymbol{x}_{\text{new}}=\boldsymbol{x}_{\text{near}}+\rho\cdot\left(\frac{\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{rand}}-\boldsymbol{x}_{\text{near}}\|}+k_{\text{p}}\cdot\frac{\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}}{\|\boldsymbol{x}_{\text{goal}}-\boldsymbol{x}_{\text{near}}\|}\right) \tag{7}$$

引力系数对于随机树的生长起着关键的作用. 在无障碍物的时候可以增大引力系数，这样随机树的生长方向在很大程度上被引力函数所影响，减小了随机树生长的随机性；与障碍物距离较近时，减小引力系数，使随机树迅速绕开障碍物，朝无障碍区域生长. 在接近终点的时候，较大的引力系数将会导致随机树叶节点在终点的周围生长，从而生成的轨迹也会在终点附近震荡. 此时适当降低生长步长，为叶节点靠近终点提供最大的支持.

2.2　路径平滑处理

增加引力分量以后，规划出的路径有可能发生震荡，为了滤出路径中的冗余节点，引入路径平滑处理的方法. 考虑到足球机器人比赛场上的障碍物数量，能够与机器人发生碰撞的障碍只有其他 9 个机器人以及球门柱，由于 RRT 算法本身的随机性，很容易在选取节点时选出很多没有必要的节点，如图 2 所示. 实际上从 Step1 到 Step3 的过程是没必要的，路径平滑的想法就是想办法滤掉这种无效的路径点.

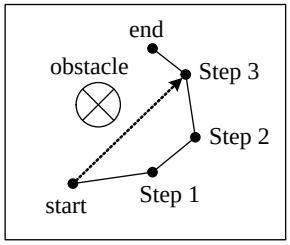


图 2　路径平滑操作

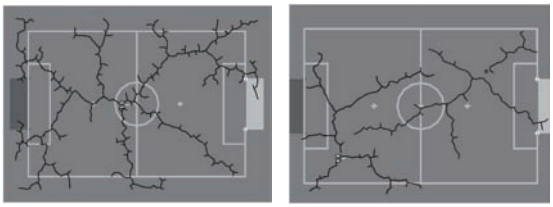
Fig.2　Path smoothing operation

路径起始节点附近可能会有冗余节点可以删除，同样，末尾节点附近也可能存在可删除的冗余节点. 平滑操作从第 2 个节点开始，依次向后检查起始节点与当前节点间的连线是否与障碍物发生冲

突，若未发生冲突，则删除该节点；若发生冲突，则保留该节点，扫描结束．对于路径末尾节点，从倒数第 2 个节点开始，依次向前检查．2 个过程的检测结束以后，可有效删除路径中的冗余节点．与障碍物发生冲突或者碰撞是指，障碍物距离路径的距离小于某个阈值，此阈值由障碍的几何尺寸来确定．

2.3 双向扩展随机树

双向扩展随机树 [13] 对 RRT 算法结构作出了改变，在搜索空间中定义了 2 根随机树，一根从起点开始生长，另外一根从终点开始生长，单根随机树与双向扩展树结构如图 3 所示．



(a) 单根随机树 (b) 双向扩展随机树

图 3 单根树与两根树结构比较

Fig.3 Structures of single tree and double tree

在随机树 1 开始搜索自由空间建立随机树的同时，随机树 2 也开始建立，2 根随机树交替进行叶节点的生长，在创建新的叶节点以后，检测该节点是否与另外一根树足够接近（2 个节点间的欧氏距离是否小于某个预设值）．当满足上述条件时，这 2 个节点连接起来，即 2 根随机树随之连接起来．双向扩展随机树的构造方法如表 1 伪码所示．

从参与连接 2 根随机树的 2 个节点开始，在各自的随机树上依次取前一个节点，直到随机树的根节点，得到 2 条子路径，连接即可得到由起点到终点的路径．双向扩展随机树在增加新节点时，要检查是否可以将 2 根随机树连接起来，增加了资源的消耗，但是提高了搜索的效率和精度．单根随机树在搜索建立的过程中，是以搜索到终点，或者到达终点附近的邻域为终止条件的．双向扩展随机树则拓宽了这种条件，2 根随机树有机会在搜索空间中的任一点结合，从而大大增加了搜索到路径的可能性．第 2 根随机树的起点为路径规划的终点，则终点一定是输出路径的最后一个节点，相对于单根随机树只能到达靠近终点的区域，双向扩展随机树提高了规划精度．

2.4 动态移动障碍物下的重规划

在障碍物可以移动的环境中，基于快速扩展随机树的路径规划方法需要作适当的发展 [15-16]．针对这个问题，最简单的方式就是重新进行路径规划，而新规划的路径要既能够避开障碍物，又能沿着原来的路径的趋势．

表 1 双向扩展随机树伪码

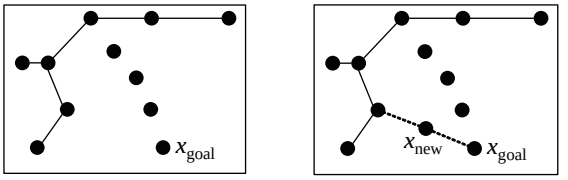
Tab.1 Pseudo code of the bidirectional extended random tree

```
doubleTreeMain()
 1  firstTree.init(x_init);
 2  secondTree.init(x_goal);
 3  useFirstTree = true;
 4  while searchCount < MAXCOUNT
 5    currentTree = getCurrentTree(useFirstTree);
 6    theOtherTree = getTheOtherTree(useFirstTree);
 7    grow(currentTree);
 8    for each node p_i ∈ theOtherTree
 9      if distance(currentTree.x_new, p_i) < ε_0
10        connectFlag = true;
11        break;
12      endif
13    endfor
14    if connectFlag == true
15      break;
16    endif
17    useFirstTree =! useFirstTree;
18  endwhile
19  if connectFlag == true
20    completeTree = connectTrees(firstTree,secondTree)
21    searchTreeForPath();
22  endif
```

2.4.1 建立路径点缓存

为了使在线规划更加高效，引入路径缓存的方法．所谓路径缓存是指建立一个缓存区域，区域里面存储了之前成功规划的路径上的点，当重新建立随机树搜索路径时，以一定的概率选择路径缓存区中的点．定义随机树朝向终点生长的概率为 p_{goal}，定义路径缓存点概率为 p_{way}，以概率 p_{way} 选择路径缓存区域中的点．p_{goal} 使随机树的生长具有偏向目标点的特性，p_{way} 使随机树的生长具有偏向上个规划路径的特性，假如 p_{goal} 和 p_{way} 的和小于 1，随机树将会以随机生长概率 p_{rand} 随机生长，则这 3 种概率的和为 1．这 3 种概率所代表的取点方式如图 4 所示．图 4(a) 中的离散点表示的是路径缓存点里的路径点，其中 x_{new} 表示新生成的节点，通过实线连接的是重新构造的随机树，虚线连接的节点表示的是选取新节点的过程．随机选取路径点是通过改写

RRT 算法中的 *chooseTarget*() 函数实现的，其伪码如表 2 所示.



(a) 带有路径缓存的随机树　(b) 以概率 p_{goal} 朝向终点生长

(c) 以概率 p_{way} 朝向路径缓存点生长　(d) 以概率 p_{rand} 随机生长

图 4　不同概率偏向性生长示意图

Fig.4　Sketch map of growth with different probabilities

表 2　选取路径缓存点伪码

Tab.2　Pseudo code for selecting path cache point

```
chooseTarget()
1   p = RandomReal in [0.0, 1.0];
2   i = RandomInt in [0, N − 1];
3   if 0 < p < p_goal
4       return x_goal;
5   else if p_goal < p < p_goal + p_way
6       return wayPointCache[i];
7   else if p_goal + p_way < p < 1
8       return RandomState();
```

该操作开始的时候先选择 2 个随机数，其中 p 是一个随机小数，范围在 0 ～ 1 之间，i 是一个随机整数，范围在 0 ～ $N-1$，其中 N 是路径缓存区中节点的数量. 概率性选取缓存区中的点是通过随机数 p 的值来实现的. 若 p 小于 p_{goal}，新节点沿着路径终点的方向选取；若 p 大于 $p_{goal}+p_{way}$，新节点沿着随机节点的方向选取；否则，新节点沿着路径缓存中的一个随机节点的方向，该随机节点由随机整数 i 确定.

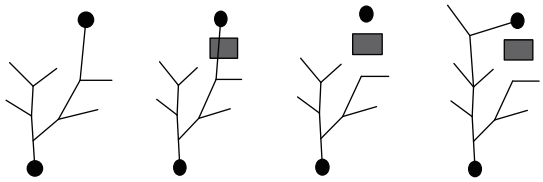
确定 2 个概率 p_{goal} 和 p_{way} 比较困难，需要实际的调试才能得到较好值. 在调试过程中本文按照文 [17] 中提到的 $p_{goal}=0.1$，$p_{way}=0.6$ 作为调试初值，在这组值的基础上进一步调试.

每一次搜索路径的操作成功得到路径以后，就把当前得到的路径中的点，存入路径缓存中，若路径点数量大于缓存的容量，则利用缓存大小对路径进行截断，前一部分路径点存入缓存.

2.4.2　动态扩展随机树

路径缓存的方法放弃了之前所有的随机树节点，虽然能够找到当前情况下的较好路径，但浪费了计算资源. 动态扩展随机树的方法保留了当前随机树，对其适当修剪，以适应移动障碍物的新位置，搜索出当前环境下的规划路径.

动态扩展随机树的执行分为初始规划和重规划两部分. 初始规划是指在自由空间中搜索建立随机树，搜索路径. 重规划是指在随机树已经建立以后，障碍物发生了移动，并且与已有随机树的树枝发生了碰撞，对随机树进行修剪，重新搜索路径. 剪枝以及动态重规划的方法如图 5 所示.



(a) 初始随机树　(b) 新的障碍物干扰到已规划路径　(c) 受影响无效的子树被剪掉　(d) 修剪的树重新生长搜索路径

图 5　动态扩展随机树构造示意图

Fig.5　Sketch map of the dynamically extended random tree

如图 5(a) 所示，按照改进 RRT 算法构造出随机树到达终点. 当搜索空间中障碍物的位置发生变化并且影响当前随机树时，标记出所有不可用的枝和节点. 对随机树进行修剪，去除不可用的部分，如图 5(c) 所示，此时所有的节点和边保证是可用的，但是随机树不一定能够到达终点. 使随机树继续生长，至少能够到达终点一次. 动态扩展随机树继承了设置路径缓存区的优点，同时又能够根据障碍物与随机树碰撞的具体情况进行合理的剪枝操作，保留有效部分，大大减少了重规划时搜索节点的资源消耗，很适合作为动态移动障碍物环境下的路径规划器.

3　RoboCup 路径规划（Path planning for RoboCup）

针对 RoboCup 赛场上的实际情况，需要对算法作具体的条件限制. 根据实际比赛经验，当机器人接近目标点时，如果目标点周围存在障碍物，此时进行路径规划，输出的路径往往会使机器人绕着障碍物一圈而到达目标点，这样的绕路容易失去控球的先机. 讨论 3 种情况下的路径搜索措施. 图 6 中 4 个图都表示障碍物在起点或者终点的附近：图 6(a) 表示障碍物与起点和终点的连线没有发生冲突；图 6(b) 表示障碍物与起点和终点的连线发

生了冲突，但是障碍物距离起点与终点的连线大于 500 mm；图 6(c) 表示障碍物距离起点很近，但是起点到障碍物与到终点的夹角大于 40°；图 6(d) 表示障碍物距离终点很近，但是终点到障碍物与起点的夹角大于 40°. 在这些特殊情况下，机器人已经很接近目标点，如果为了避开障碍而沿着规划路径绕开会变得非常被动. 机器人接近目标点的途中如果遇到障碍物，也可以通过超声波检测到障碍的存在从而被动绕开，与障碍之间最少有 40° 的夹角，被动绕开障碍不会耗费很多资源.

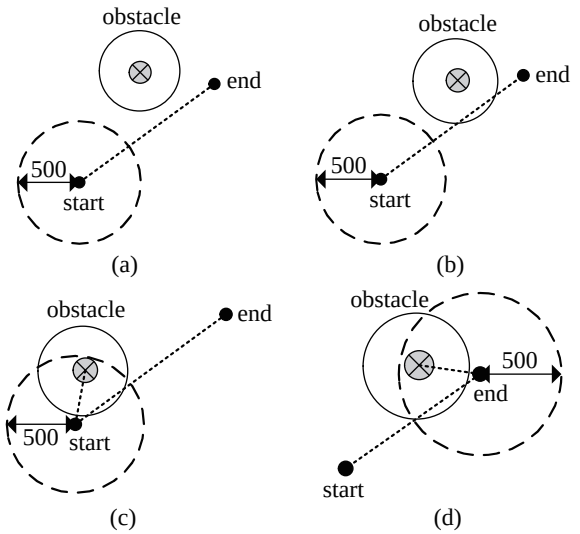


图 6 特殊情况下的路径选择

Fig.6 Path selection in exceptional cases

图中所描述的 500 mm 与 40° 是根据 NAO 机器人的几何尺寸以及实际比赛的经验得到的，可以作为调试参数参与调试. 场上的障碍物位置信息会被定时更新，障碍物是否与当前随机树有冲突的信息也将被更新. 机器人只关心到后面几个目标点的路上会不会遇到障碍物. 假如障碍物影响到了更后面的路径点，不会触发重规划，理由是引发碰撞的障碍物是在时刻移动着的，当前时刻的碰撞很有可能在此后被消除，所以对于距离碰撞点一定距离的机器人来说，该路径很大概率上还是有效的. 频繁的重规划会使计算量增加很多，同时由于不同规划路径的方向都有一些差异，减少规划次数会减少路径的抖动.

动态扩展随机树需要进行剪枝的操作，针对赛场上的动态环境，随着随机树的生长，赛场空间中树枝会很多，如果每次树枝发生碰撞都要修剪会耗费很多资源，所以只有在前方路径点上发生碰撞，需要进行路径重规划的时候，才去检查需要修剪的树枝，然后进行重规划.

4 仿真与实验验证（Simulations and experiments）

4.1 Matlab 仿真平台

图 7 是 Matlab 静态仿真环境的显示界面，其中的黑色圆圈表示的是随机生成的障碍物信息，是由地图构建模块生成的，同时生成的还有随机的起点和终点信息. 如果此时存在随机树以及由随机树生成的路径，显示模块会同时显示出随机树的枝干以及路径信息，包括路径的起点和终点，随机树的枝干用细线表示，输出路径用粗实线表示. 地图生成的场地信息是完全随机的，但是为了使实验结果具有可比性，文中都采用同样的地图作算法的验证.

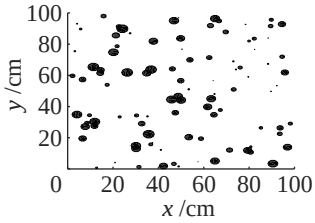


图 7 静态仿真环境

Fig.7 Static simulation environment

4.1.1 基本 RRT 算法

图 8 是用基本 RRT 算法搜索的路径，可见虽然 2 个地图的障碍物位置以及数量是完全一样的，起点以及终点的位置也是一样的，但是这种情况下搜索出来的路径完全不一样，而且图中很多的细线表示有很多没有用到的节点被搜索插入到随机树中，此实验中搜索步长 ρ 取 5 cm，朝向终点生长概率 p_{goal} 取 0.3.

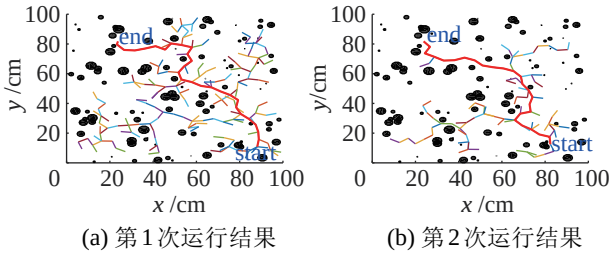


图 8 基本 RRT 算法仿真图

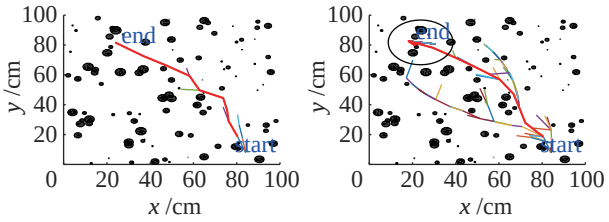
Fig.8 Simulation result based on the basic RRT algorithm

4.1.2 增加引力分量

对于随机树中的每一个节点增加一个引力分量，有助于增强随机树的平滑性，因为随机搜索方法本身不是一个最优的算法，增加引力分量后将进一步缩短路径，减少随机性. 仿真测试如图 9 所示.

如图 9(a) 所示，实验中搜索步长 ρ 取 5 cm，朝向终点生长概率 p_{goal} 取 0.3，引力系数 k_{p} 取 1.5. 增

加引力分量后，路径缩短，而且平滑程度增强，但是图 9(b) 中，路径在终点附近发生了震荡，这是因为引力系数为常量，在靠近终点时，随机树的生长会被引力强制拉过终点，又被拉回来．引力系数应该随着随机树的生长而逐渐变小，当靠近终点的时候，引力系数应该接近 0.



(a) 第 1 次运行结果　　(b) 第 2 次运行结果

图 9　增加引力分量仿真结果

Fig.9　Simulation results based on gravitation thought

4.1.3　路径平滑操作

如果给每个节点都增加引力分量，容易在终点附近造成路径的震荡，路径平滑操作可以检测出这些异常点，在路径中删除．比赛中场上机器人数量比上述仿真图中的障碍物要少很多，平滑处理可以删除很多没有必要的节点，缩短路径的同时减少机器人转体的次数．路径平滑处理的仿真测试结果如图 10 所示．

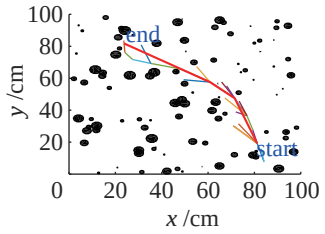


图 10　路径平滑操作仿真结果

Fig.10　Simulation result based on path smoothing operation

此实验中，搜索步长 ρ 取 5 cm，朝向终点生长概率 p_{goal} 取 0.3，引力系数 k_{p} 取 1.5．图 10 的路径生成在增加了路径平滑处理以后，路径的平滑度有很大的提高，节点数也明显减少．

表 3　不同方法生成路径的长度与计算时间对比

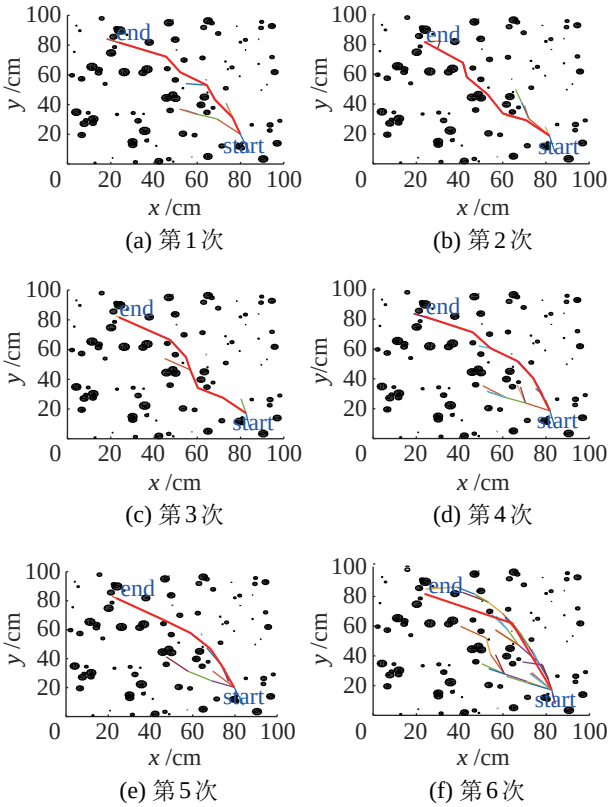
Tab.3　Path length and computation time of different methods

方法	路径长度 /cm	计算时间 /s
基本 RRT 方法	114.035	0.343
增加引力分量	97.681	0.3941
路径平滑	91.032	1.02

表 3 所示为不同方法在相同障碍物信息、相同起点信息和相同终点信息时，在 Matlab 仿真环境中的性能对比．对于每种方法，在仿真环境中运行 10 次，得到 10 组路径长度和计算时间值，取其平均值，每个方法的参数与上文所述相同．可见，随着方法的改进，计算复杂度增加，用更多的计算资源可以换取更短的路径．

4.1.4　路径缓存

路径缓存方法是为了在动态环境下减少路径抖动而设计的方法，可以通过测试同一地图多次规划是否能够规划出相似的路径来验证，图 11 为连续运行 6 次得到的结果．



(a) 第 1 次　　(b) 第 2 次

(c) 第 3 次　　(d) 第 4 次

(e) 第 5 次　　(f) 第 6 次

图 11　路径缓存区仿真实验

Fig.11　Simulation result of path cache

图 11(a) ～ (f) 是应用路径缓存区方法连续规划 6 次所得到的同一地图中的 6 次规划结果，6 条路径在大致走向上是一样的，符合之前对于动态障碍物的要求．

4.2　SimRobot 环境仿真

在 SimRobot 平台上主要测试的是路径规划器生成的路径是否满足场上实际情况．应用基本 RRT 算法规划出的路径如图 12 所示．图 12 中，黑色实线为基本 RRT 算法的输出路径，中间“×”标注的地方表示的是对方机器人障碍．可见基本 RRT 算法的输出路径光滑程度差、路径长，凡是在转折点处机器人都需原地转体，会浪费很多时间并且失去控球的先机．

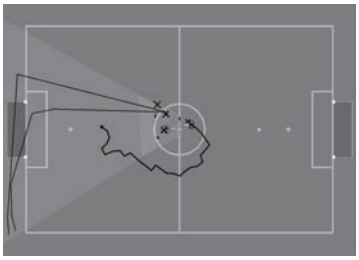
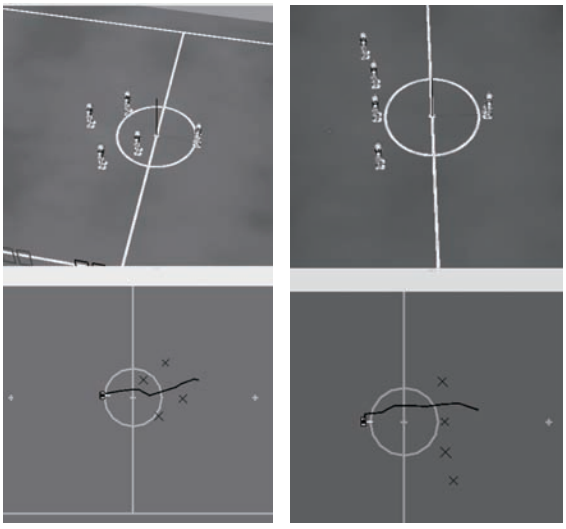


图 12 基本 RRT 算法在 SimRobot 仿真结果

Fig.12 Simulation result of the basic RRT algorithm in SimRobot

SimRobot 里面对方机器人是不会移动的，可以用鼠标手动移动障碍物的位置，观察此时的规划路径情况．应用改进动态扩展随机树的方式规划出的路径如图 13 所示．



(a) (b)

图 13 动态扩展随机树在 SimRobot 中的仿真结果

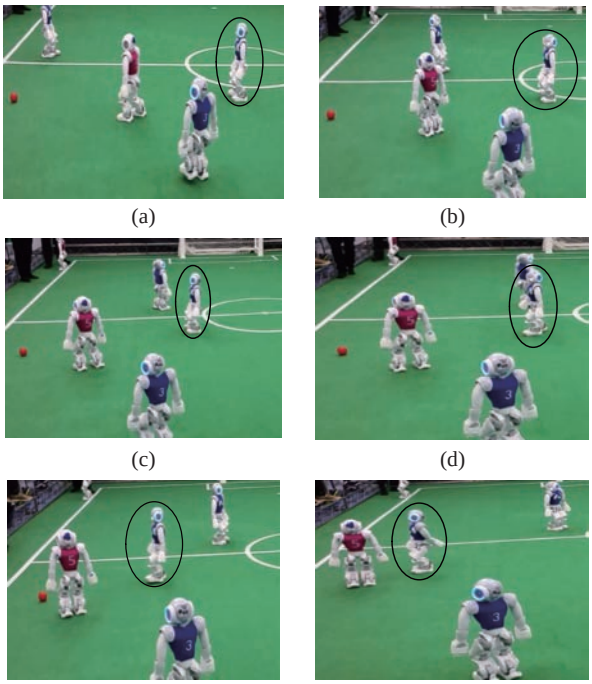
Fig.13 Simulation result of the dynamically extended random tree algorithm in SimRobot

图 13(a) 中机器人前方有 4 个机器人，这是赛场上比较复杂的情况，在这种情况下所规划出的路径使机器人能够在对方 4 个机器人的间隙中走过，并且保持一定的安全距离，完全满足赛场上的避开障碍的需要．图 13(b) 对方的 4 个机器人位置发生移动，此时有障碍物干扰到了机器人的前进路径，路径进行了重规划，由于路径缓存区的引入，以及随机树的剪枝操作，重规划出的路径与原路径没有发生很大差异，减少了路径的抖动，能够满足赛场上的需求．

4.3 实体机器人测试

在通过 SimRobot 测试以后，虽然代码结构上不需要再作其他调整，但是针对实际机器人要进行一些参数调整．实体实验结果如图 14 所示．

图 14 中被椭圆圈起来的蓝色机器人从图 14(a) 的位置朝向球的位置走，即路径规划的起点为当前机器人的位置，终点为球的位置，与当前机器人距离最近的障碍机器人是前方的红色机器人，为了避开它，规划了一条路径．双向随机树从机器人当前位置以及球的位置同时开始生长，当 2 根随机树的叶节点很接近的时候，2 根随机树被结合起来，随即生成路径．当前位置蓝色机器人规划出的路径按照其行走方向来看是一条弧线，能够绕过对方的红色机器人到达球的位置，参与球的拼抢，获得控制球的能力．



(a) (b) (c) (d) (e) (f)

图 14 实际比赛规划路径

Fig.14 Path planning in actual competition

实际比赛中，搜索步长 ρ 取 50 mm，朝向终点生长概率 p_{goal} 取 0.3，朝向路径缓存点生长概率 p_{way} 取 0.6，引力系数 k_{p} 取 1.1，在引入路径缓存之后，引力系数不能设置得过大，否则，在选取新节点的时候，会减弱路径缓存的作用．

5 结论（Conclusion）

针对快速扩展随机树的缺点，本文提出了几点改进方案，以一定概率选择目标点，能够使随机树生长过程中概率性地偏向终点选择节点，减弱了其随机性；引力分量使搜索出的路径更加平滑，路径长度进一步缩短；路径平滑处理使得路径上无用的节点被删掉，同时消除了由于引力分量所造成的路径震荡问题．最后移植到 NAO 机器人上的动态扩

展随机树算法所规划出的路径能够满足赛场上较复杂障碍物环境的需要，在比赛的时候表现很好.

动态扩展随机树还有很多需要改进的地方. 通过这种选取随机点的方式生成的路径，本身是由一个个随机树节点组成的，这样就会导致生成的路线是一条折线，并非理想的平滑曲线，这种路径对于机器人来说是不利的. 当机器人从一个节点向另外一个节点运动时，首先要完成原地的转体运动，当转到下一个节点的方向的时候再继续往前走，NAO 机器人的身体结构可以完成这样的运动，但是很多机器人不能，而且会耗费时间和能量. 进一步的研究工作希望能够在当前算法的基础上解决生成平滑路径的问题并在仿人机器人平台上展开实验验证.

参考文献（References）

[1] LaValle S M. Rapidly-exploring random trees: A new tool for path planning[R]. Ames, USA: Computer Science Department, Iowa State University, 1998.

[2] Kuffner J J, LaValle S M. RRT-connect: An efficient approach to single-query path planning[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2000: 995-1001.

[3] Esposito J, Kim J, Kumar V. Adaptive RRTs for validating hybrid robotic control systems[M]//Algorithmic Foundations of Robotics VI. Berlin, Germany: Springer, 2005: 107-121.

[4] Karaman S, Walter M R, Perez A, et al. Anytime motion planning using the RRT[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2011: 1478-1483.

[5] 李威洲. 基于 RRT 的复杂环境下机器人路径规划 [D]. 哈尔滨：哈尔滨工程大学，2012.
Li W Z. Robot motion planning based on rapid-exploring random trees in complex environment[D]. Harbin: Harbin Engineering University, 2012.

[6] 王全. 基于 RRT 的全局路径规划方法及其应用研究 [D]. 长沙：国防科学技术大学，2014.
Wang Q. Research on rapidly-exploring random trees based global path planning and its application[D]. Changsha: National University of Defense Technology, 2014.

[7] Du M B, Chen J J, Zhao P, et al. An improved RRT-based motion planner for autonomous vehicle in cluttered environments[C]//IEEE International Conference on Robotics and Automation. Piscataway, USA: IEEE, 2014: 4674-4679.

[8] 宋金泽，戴斌，单恩忠，等. 一种改进的 RRT 路径规划算法 [J]. 电子学报，2010，38(2A)：225-228.
Song J Z, Dai B, Shan E Z, et al. An improved RRT path planning algorithm[J]. Acta Electronica Sinica, 2010, 38(2A): 225-228.

[9] Wu D, Sun Y J, Wang X, et al. An improved RRT algorithm for crane path planning[J]. International Journal of Robotics and Automation, 2016, 31(2): 84-92.

[10] Kala R. Rapidly exploring random graphs: Motion planning of multiple mobile robots[J]. Advanced Robotics, 2013, 27(14): 1113-1122.

[11] Khatib O. Real-time obstacle avoidance for manipulators and mobile robots[J]. International Journal of Robotics Research, 1986, 5(1): 90-98.

[12] 郝利波. 基于改进 RRT 与人工势场混合算法的足球机器人路径规划研究 [D]. 西安：西安科技大学，2011.
Hao L B. Based on improved RRT with artificial potential field hybrid algorithm of robot soccer path planning research[D]. Xi'an: Xi'an University of Science and Technology, 2011.

[13] Yuan X, Zhu Q D, Yan Y J. Collision avoidance planning in multi-robot system based on improved artificial potential field and rules[J]. Journal of Harbin Institute of Technology, 2009, 16(3): 413-418.

[14] Hu Y L, Zhang Q S. Multi-robots path planning based on improved artificial potential field method[M]//Advanced Materials Research. Clausthal-Zellerfeld, Germany: Trans Tech Publications, 2012: 937-940.

[15] Abbadi A, Matousek R, Minar P, et al. RRTs review and options [M]//Computational Engineering in Systems Applications (vol. II). WSEAS Press, 2011: 194-199.

[16] Wang W, Li Y. A multi-RRTs framework for robot path planning in high-dimensional configuration space with narrow passages[C]//IEEE International Conference on Mechatronics and Automation. Piscataway, USA: IEEE, 2009: 4952-4957.

[17] Bruce J, Veloso M M. Real-time randomized path planning for robot navigation[M]//Lecture Notes in Computer Science, vol.2752. Berlin, Germany: Springer, 2002: 288-295.

作者简介：

刘成菊（1980 –），女，博士，副教授. 研究领域：仿生技术，机器人运动控制，进化计算.

韩俊强（1992 –），男，硕士. 研究领域：机器人运动控制.

安　康（1981 –）男，博士，讲师. 研究领域：机器人运动控制，被动行走.