

doi: 10.7690/bgzdh.2015.05.024

一种全局“水中救援”比赛策略

周伟诚, 夏庆锋

(南京大学金陵学院机器人实验室, 南京 210089)

摘要: 针对水中机器人水球比赛中的水中救援项目, 提出一种全局“水中救援”比赛策略。基于比赛规则的要求, 制定了机器鱼救援的整体策略及路径规划, 并设计了以尾部甩动的 Hammer 函数算法, 替代了比赛中常用的头部撞击算法。实践结果表明: 该策略提高了机器鱼成功撞倒目标的概率, 在 2013 年 RoboCup 中国机器人大赛全局视觉组的水中救援比赛中取得了亚军。

关键词: 机器鱼; 甩尾算法; 路径规划

中图分类号: TP242.6 **文献标志码:** A

A Global Game Strategy in “Water Rescue” Competition

Zhou Weicheng, Xia Qingfeng

(Laboratory of Robotics, Jinling College, Nanjing University, Nanjing 210089, China)

Abstract: As for the water rescue game in underwater robot water polo game, puts forward a global “Water Rescue” competition strategy. Based on the requirements of the rules of a game, develops the overall strategy and path planning of robot fish rescue, and designed a slapping-tail Hammer function algorithm, replacing the common head-butting algorithm in the game. Practice results show that: this strategy greatly improves the probability of hitting target down for robot fish, and get the runner-up in Global Vision Group “Water Rescue” Game of 2013 China RoboCup Robot Competition.

Keywords: robot fish; slapping-tail algorithm; path planning

0 引言

水中机器人比赛是中国机器人大赛的正式项目, 同时也是中国人的首创, 更有望成为世界标准赛事^[1]。比赛以仿生机器鱼在水池之中进行竞速、追逐和水球等形式展开。目前, 水中机器人比赛主要分为全局视觉组、自主视觉组、2D 仿真组和自由创意组 4 大类。机器人开发的一个重要因素就在于其能代替人力自主地完成一系列任务, 因此人的辅助干预在机器人完成整个任务过程中所占重要性应该越小越好, 并以机器人能够完全脱离外界辅助而完成任务作为最高目标。笔者在构思项目的整体策略时也逐渐意识到这一理念的先进性, 舍弃了原先利用同一个冲撞函数对手动切换指定的不同目标进行反复攻击的方案, 改为自主性较高的整体流程方案, 虽然与前者比起来代码结构更为复杂, 但根据测试结果显示执行效率有较高的提升。

1 比赛规则介绍^[2]

参赛队派一条机器鱼按照抽签顺序参加比赛。水池中安放 5 个支柱, 其中编号 1、2、3 的支柱为最低难度的支柱, 编号为 4 为中等难度的支柱, 编号 5 为最高难度的支柱。每个支柱顶部放置一个装

有真鱼的透明盒子, 盒子上方贴有与支柱直径相当的红色色标, 整个盒子代表受困待救的小鱼。支柱的初始位置如图 1 所示。机器鱼从水池右侧池壁中心点出发, 逐一冲撞支柱, 使得支柱顶部的盒子能够落入水中(水球偏离设定位置), 代表成功营救受困的小鱼。

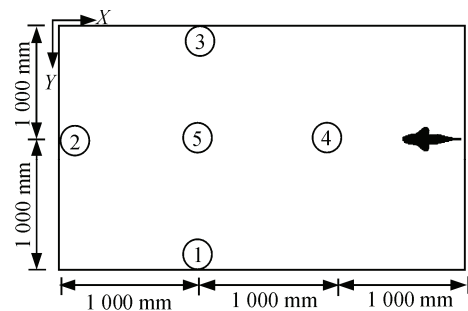


图 1 全局视觉愤怒的小鱼示意图

比赛的计分规则如下: 编号 1、2、3 的支柱上的盒子落入水中分别记 1 分, 编号 4 支柱上的盒子落入水中记 2 分, 编号 5 支柱上的盒子落入水中记 3 分。若在 5 min 内完成比赛, 则记录完成比赛所用时间。以所用时间长短排序, 时间最短者获得全局视觉水中救援比赛项目第 1 名, 以此类推。若在 5 min 内未完成比赛, 则比较各队得分高低进行排名。若得分相同, 则依据第 1 个落入水中盒子所用

收稿日期: 2015-03-06; 修回日期: 2015-04-15

作者简介: 周伟诚(1993—), 男, 江苏人, 学士, 从事软件工程研究。

的时间长短计算名次，以此类推。

2 整体策略

如图 1 所示，5 个目标呈十字型分布于水池中，编号为 1、2、3 的紧靠水池边缘的 3 个目标难度最低，通过实验得知，机器鱼以正常前进游动的方式就能将其撞倒；4 号目标离机器鱼起始位置最近但难度中等，并且被机器鱼以正常直线游动的方式撞倒的概率较小^[3]；5 号机器鱼的目标难度最高，以往每届比赛中只有一两支参赛队依靠复杂的头部撞击函数才能将其撞倒，可谓是比赛的制胜点。根据以上对目标的分析，笔者采取了先易后难的路径方案，即先以较快速度的方式将难度最低的 3 个目标撞倒后，再利用复杂有效的撞击函数依次向难度递增的 4、5 号目标发起进攻。其中笔者考虑到难度中等的 4 号目标位于机器鱼起始方向的正前方，并且距离较近，所以设计了比赛开始后首先直线撞击 4 号，但无论是否成功都立即执行撞击 1、2、3 号目标的统筹路线方案，以期达到提高整体效率的目的。

笔者为了实现项目整体流程自动化的目标，同时结合控制平台执行代码的原理(即平台在很短的周期时间内遍历执行策略代码，并周而复始地循环该周期)，对多种代码结构进行试验，最终采取了效率最高的以 Step 标志位作为 case 的 switch 语句结构。代码片段如下：

```

switch(Step)
{ case 0:
    If(MissionCompleted(Ball3,m_action[fishId],fish0,fishId,24)) Step = 1;
    break;
  case 1:
    If(MissionCompleted(temp[1],m_action[fishId],fish0,fishId,24)) Step = 2;

```

考虑到平台的周期执行性工作原理，而局部变量的生命周期仅能存在于一个执行周期中，新的执行周期开始时局部变量会被重新初始化，故将 Step 设为静态变量以方便其在项目流程中持续作用。

Bool MissionCompleted()是笔者自定义的用于判断机器鱼是否到达目标点的函数，基本原理是当机器鱼的鱼头点与目标点距离小于某阈值时(即最后一个参数)，则认为已到达，并返回 true 值，增加 Step 值并跳出等待下一个执行周期，反之返回 False 并跳出等待。

3 路径规划

如以上比赛规则所述，1、2、3 这 3 个目标被

撞倒的难度最低，仅由 2 个笔筒重叠而成，没有使用胶带将其捆绑固定，也没有在内部放置配重。经过多次实际测试得出的结果表明：机器鱼如果能以中档速度直线前进，只需从侧面碰撞 1、2、3 号目标即可将其击倒。基于以上结论，笔者在规划机器鱼撞击 1、2、3 号目标的路径时，更侧重于时间最优的路径方案。考虑到撞击多个目标路线的连贯性和快捷性，笔者决定以 1 号、2 号、3 号和尝试撞击的 4 号目标为顶点，依次连接成为一个菱形，使得任意两个连续目标点之间的路线为最短的直线，且整体路线总和最短。原路径设计如图 2 所示。

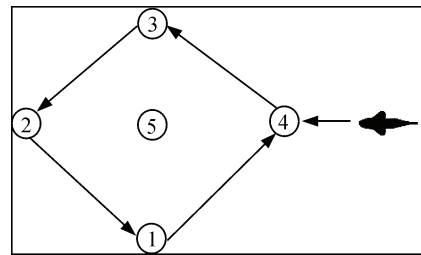


图 2 原路径设计

然而在实际测试中却发现该方案并不理想，主要原因有 2 个：第一，实际情况下机器鱼按照路径游动时，在顶点处的动作处理并不能稳定地到达理论预期，即实际的顶点会有一定概率偏离目标所在位置；第二，忽略了目标点处若以菱形顶角形状路径前进，则机器鱼需要在目标点转动菱形内角的补角角度($\geq 90^\circ$)，这会造成较多的时间耗费。而且鱼体转动过程中的线速度很低，不利于对目标的撞击。

基于以上 2 个缺陷，笔者修改了原本的游动路径，采用微分的思想将需要转动的角度分解成多个较小的部分，统筹分布于整体路径中。具体做法为：第一，将目标点处的路径由顶角改为平角(即直线)，增加机器鱼与目标碰撞的期望面积以及撞击的速率，提高碰撞概率和撞击效率；第二，按照一定的阈值在目标点前后设置临时点，为鱼体角度调整和直线撞击过程提供缓冲机制。改进后的路径设计如图 3 所示。

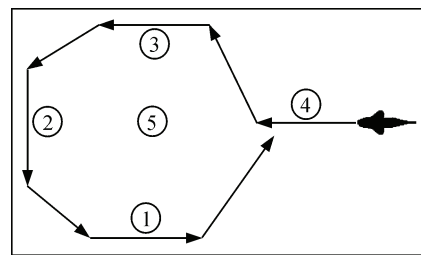


图 3 改进后的路径设计

4 甩尾撞击算法

如第1章的比赛规则所述,4、5号目标难度依次增大,以针对1、2、3号目标使用的撞击方式将其全部撞倒的可能性为零,而其中难度最高的5号目标更是比赛的制胜点,故需要设计攻击性和复杂度更高的撞击算法。笔者在该项目首次比赛的经验基础上,经过理论改进和反复试验,最终设计出了一套效率较高的甩尾撞击算法,封装后命名为Hammer函数,原型如下:

```
void Hammer(CFishInfo fish,CPoint aim,
CFishAction &action,int fishId,long AttackRange),
其中 AttackRange 参数为自定义阈值。
```

4.1 状态划分

笔者在设计 Hammer 函数时,将机器鱼撞击目标的整体动作分为3个阶段,并且将分别对应的状态定义为准备态、攻击态和撤离态。各状态的功能概述和联系为:当机器鱼处于准备态时,主要行为是通过调整位姿以瞄准并靠近目标;当目标处于攻击范围内时便切换成攻击态,对目标做出甩尾动作;攻击结束后切换成撤离态,离开目标一定距离后切换为准备态进入下一个动作周期,以此实现自动化循环过程。

为了使状态划分的效果得到具体的保障,笔者以目标为圆心,不同距离为半径构造了2个同心圆,这样就将整个水池区域粗略划分为3个区域^[4]。

在此区域划分的基础上,笔者又结合机器鱼在实际撞击过程中运动的路径,将以上区域划分结果细化为4个不同编号的分区,以提供机器鱼判定本身状态的具体基准。区域划分情况如图4所示。

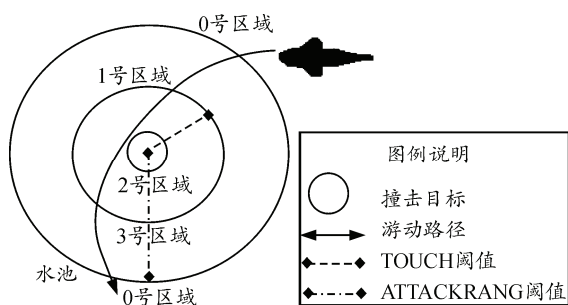


图4 区域划分示意图

为了解决上图中各个分区的划分问题(尤其是1、3分区),笔者采用了2个布尔型变量——Attacked、Touched作为标志位,对以上4个分区对应的3种状态进行编码。编码规则如表1所示。

其中,当机器鱼鱼头点与目标点的距离大于ATTACKRANGE时(即机器鱼鱼头处于0号区域),

Attacked标志位则被置为False,反之置为True;当机器鱼鱼头点与目标点的距离小于TOUCH时(即鱼头处于2号区域),Touched则被置为True,反之置为False。

表1 状态编码

机器鱼位置	Attacked	Touched	对应状态
0号区域	F	F	准备态
0号区域→1号区域	F	F	准备态
1号区域	F	F	准备态
1号区域→2号区域	F→T	F→T	准备态→攻击态
2号区域	T	T	攻击态
2号区域→3号区域	T	T→F	攻击态→撤离态
3号区域	T	F	撤离态
3号区域→0号区域	T→F	F	撤离态→准备态

4.2 状态策略

如区域状态划分图所示,1、2、3号区域分别对应准备态、攻击态和撤离态。其中每个状态对应的策略思路及算法如下所述。

4.2.1 准备态

准备态的主要策略思想是通过调整位姿以瞄准并靠近目标,其关键参数是ATTACKRANGE,即笔者自定义的宏,当机器鱼向目标游动并且两者距离递减直到小于该阈值时,则判断机器鱼进入动作周期并处于第一阶段准备态。准备态中使用的主要算法为Roundp2p函数,这是比赛平台提供的封装函数,主要作用是根据目标点和机器鱼的距离和角度的实时关系,不断调整机器鱼的速度和方向,使其向目标点前进。

4.2.2 攻击态

当机器鱼鱼头与目标点的距离小于TOUCH值时(即鱼头处于2号区域中)就被判断处于攻击态。攻击态属于最重要的状态,其策略设计是极其重要的。笔者在测试中发现,鱼尾甩动对目标产生的撞击效果平均优于通常使用的头部撞击方式的效果。于是笔者决定采用尾部甩动的方式作为攻击态的主要行为。攻击态整体策略为:当机器鱼从准备态转换为攻击态时,将游动方向设为直行,骤降速度,即让机器鱼按照状态切换前瞬时的游动状态依靠惯性向前漂动,使鱼身紧靠目标直线漂过,当鱼尾接触目标时,以最大角度向目标方向大力甩尾。要提高该算法的效率,需要处理好以下两处细节:

1) 准确判断甩尾的时机。

笔者应对该问题时采取的是以距离判断的方案,即当鱼尾与目标距离到达合适的值时甩尾(距离过大或过小都会导致撞击面积小而影响效果)。但比赛平台无法直接获取鱼尾的坐标点,考虑到鱼体总

长是固定的,所以笔者将鱼尾最大面积接触目标时,鱼头与目标点的最小距离设为 ACTION 阈值。当机器鱼处于攻击态,并且鱼头与目标距离大于 ACTION 值时,更改其模态并设置方向,使鱼尾以最大的力度甩动。机器鱼甩尾动作位置如图 5 所示。

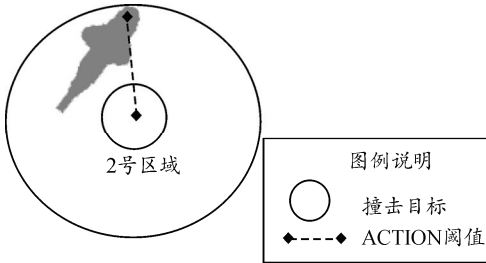


图 5 机器鱼甩尾动作位置示意图

2) 判断甩尾方向。

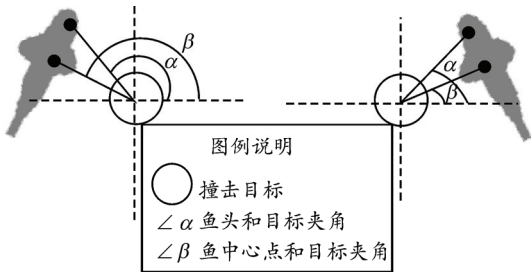


图 6 判断甩尾方向示意图

当鱼尾处于合适的位置进行甩动时,需要判断目标相对鱼尾的方位(即目标处于鱼尾左侧或右侧),以确定甩动方向。由于比赛平台使用的是以逆时针方向作为正方向的坐标系,所以笔者采用的处理方案是,根据鱼头和目标点形成的角度(α)与鱼

中心点和目标点的角度(β)相减所得角度的正负进行判断。如图 6 所示,角度差值为正则目标点位于鱼尾左侧,角度差值为负则目标点位于鱼尾右侧。

4.2.3 撤离态

撤离态的主要作用是,使机器鱼撞击动作结束后,向目标反方向前进以远离目标,到达 0 号区域并进入新的 Hammer 函数周期,撤离期间设置较小的前进角度使鱼头缓慢调转,为新的 Hammer 动作做位姿准备。

5 结束语

水中救援项目对于撞击算法和比赛用时具有较高的要求,经过试验与比赛测试,笔者提出的包含甩尾算法的“水中救援”比赛策略表现优于以往其他策略。该算法在南京大学金陵学院参加的多次比赛中已得到成功应用,并在 2013 年 RoboCup 中国机器人大赛全局视觉组的水中救援比赛中取得了亚军,证明了该策略的有效性。

参考文献:

[1] 北京大学智能控制实验室. 水中机器人竞赛推介书 [M/OL]. <http://robot.pku.edu.cn/>, 2012.
 [2] 谢广明. 机器人水球比赛项目推介书[M]. 北京: 北京大学工学院, 2009: 1-5.
 [3] 孙琴, 武海健, 夏庆锋. 机器鱼游动性能改善方法[J]. 兵工自动化, 2014, 33(12): 67-71.
 [4] 张轩, 周映慧, 史豪斌. 基于二次区域划分的机器鱼比赛策略[J]. 兵工自动化, 2014, 33(12): 63-66.

 (上接第 76 页)



图 5 智能避障图

5 结语

笔者提出了一种基于 Arduino Mega 2560 的智能小车的总体设计,探讨了系统的核心处理器等关键模块的选型,并初步介绍了智能小车关键模块的

基本原理和应用情况,在此基础上进行了小车软件设计。试验结果表明:该智能小车操作方式灵活、开发成本低、易于推广、适用于教育、家庭等多种场合,也可作为科研机构研究平台进行二次开发,具有较好的应用前景。

参考文献:

[1] 曹永彦, 王培俊, 毛茂林, 等. 基于单片机的机器人设计与制作[J]. 实验科学与技术, 2010(1): 22-23.
 [2] 赵津, 朱三超. 基于 Arduino 单片机的智能避障小车设计[J]. 自动化与仪表, 2013(5): 1-4.
 [3] Wheat D. Arduino Internals[J]. Springer Berlin, 2013, 11-12.
 [4] 张睿. 基于 K60 的智能小车控制系统设计与实现[D]. 杭州: 杭州电子科技大学, 2014.
 [5] 汤国望. 交互式移动机器人系统的设计与实现[D]. 武汉: 华中科技大学, 2012.
 [6] 石晶, 王宜怀, 苏勇. 基于 ARM Cortex-M4 的 MQX 中断机制分析与中断程序框架设计[J]. 计算机科学, 2013, 40(6): 41-45.