

青少年机器人足球比赛系统决策编程的可视化

张伯泉, 杨宜民

(广东工业大学自动化学院, 广州 510090)

摘要: 分析了机器人足球比赛系统中决策子系统的一般结构, 建立了产生式推理模型和决策的表达模型, 构造了机器人足球比赛系统的决策程序的结构; 定义了决策存储的结构体变量类型, 设计了菜单, 并以此形式实现了决策编程的可视化。

关键词: 机器人足球; 可视化编程; 决策前件; 决策后件

Visualization of Teenage Robot Soccer Decision Programming

ZHANG Boquan, YANG Yimin

(Institute of Automation, Guangdong University of Technology, Guangzhou 510090)

【Abstract】 This paper analyses the general structure of robot soccer decision subsystem, and makes the production reasoning model and the expressional model of decision-making. Thus it constructs decision-making program's structure for this robot soccer system. A new structural variable type for decision storage is also defined. Eight menus are created, through which decision programming is visualized.

【Key words】 Robot soccer; Visualized programming; Decision former part; Decision rear part

机器人足球比赛是近年来国际上迅速开展起来的一种高科技对抗活动, 该项活动在我国各大高校里也日渐活跃。为了培养学生从小爱科学、用科学的兴趣, 锻炼学生在实践中发现问题、思考问题、解决问题的能力, 全面提高学生的综合素质, 贯彻《2001-2005年中国青少年科学技术及活动指导纲要》, 近年来我国把智能机器人教育引入了中小学教育, 积极开展了青少年机器人足球比赛^[1,2]。我国于2001年举办了首届中国青少年机器人竞赛, 其中包括了机器人足球比赛项目, 现已连续举行了4届。目前, FIRA和RoboCup是两大国际机器人足球比赛组织。机器人足球比赛对研究多智能体的合作与竞争理论具有重要的实践与指导意义。通常足球机器人系统可以划分为机器人车体子系统、通信子系统、视觉子系统和决策子系统4个部分^[3], 通过计算机视觉子系统闭环而构成智能决策和控制系统。视觉子系统负责识别球和机器人, 得到现场信息; 通信子系统负责传送现场获得的信息和发送机器人车体的运动指令; 机器人车体系统负责接收指令并驱动机器人车体运动; 决策子系统则根据现场信息推理计算从而得到机器人运动控制指令。其中决策子系统组织机器人协作, 做出适当的战术配合, 是取得胜利的关键^[4]。但机器人足球比赛决策子系统的编制比较复杂^[5], 需要高层次的知识和计算机编程水平, 不适于青少年机器人足球比赛。本文设计了一种可视化的比赛决策的编制方法。参赛者只要按照一定的步骤, 即可生成比赛用的决策及决策库, 在比赛时只要将自己编制的决策库与客户程序中的决策子程序连接即可达到比赛的目的。决策及决策库的生成可以独立完成, 从而使青少年将注意力集中于比赛决策开发, 以达到理想的比赛成绩。

1 机器人足球比赛系统决策子系统的一般结构

决策子系统根据由视觉子系统获得的现场信息, 分析当前比赛形势, 确定攻防策略, 然后给机器人分配角色, 决定各个机器人的动作, 最后将运动指令发送给机器人。根据这

一特点, 机器人足球控制决策系统中普遍采用了分层控制的结构^[6]。东北大学提出了由信息预处理、态势分析和策略选择、队形确定、角色分配、目标位姿确定和左右轮速确定等6个步骤组成的决策推理系统, 并运用到足球机器人决策系统设计中; 中南大学提出过由协调层、运动规划层和动作层组成的3层决策模型^[7]。图1是Mirosoft机器人足球比赛决策子系统的一般结构。

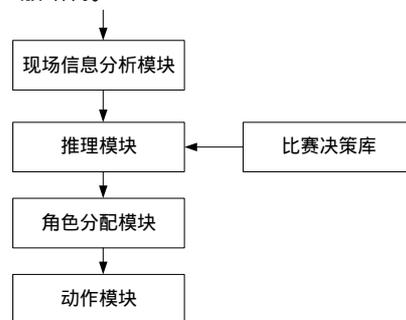


图1 机器人足球比赛系统决策子系统的一般结构

现场信息是视觉子系统对现场某周期信息识别的结果, 包括球与双方队员的位姿等信息。

推理模块是根据视觉子系统传递来的现场信息, 对场上的形式进行判断和推理, 并根据比赛决策库产生相应的下一周期的比赛决策。比赛决策确定了本方球员在下一周期的角色、任务以及执行任务的位置。例如, 在3Vs3的比赛中, 当采用全攻全守策略, 在球到对方球门附近, 产生的决策是离球近的机器人负责去抢球, 称为主攻机器人; 另一机器人向对方球门区运动, 伺机得分, 称为助攻机器人。主攻机器人

基金项目: 广州市科技资金资助项目(2004Z2-D0151)

作者简介: 张伯泉(1974-), 男, 博士生, 主研方向: 机器人控制, 计算机视觉; 杨宜民, 教授、博导

收稿日期: 2005-12-25 **E-mail:** zbq-01@163.com

得球可以将球传给协攻机器人，也可直接射门；第3个机器人为守门员，只在球门内活动，完成扫球动作。在本客户程序中采用产生式推理模型。

角色分配模块确定每个队员在某种策略下的下个周期的角色。如在3Vs3的比赛中，在全攻全守策略下，当球在对方半场时，确定下个周期的队员的主攻、协攻或守门员等角色。

动作模块是根据每个机器人当前周期的角色与任务确定相应的动作。基本动作有踢球、射门、拦截、守门、避障等。

2 决策模型和产生式推理模型

在机器人足球比赛中，我们首先选定某个策略，然后在此策略的基础上，生成一系列决策，从而形成决策库。为方便推理，决策由前件与后件组成。前件也称为产生式推理的条件，后件也称为产生式推理的结果。决策的前件即产生式推理的条件由球的位置(Ball_location)、本方球门受威胁程度(Danger)、对方球员位置(Opponent_location)构成；决策的后件即产生式推理的结果由本方球员的角色(Role)、任务(Task)以及执行任务的位置(Location)构成。根据产生式推理模型：if A then B，把MiroSot推理模型和决策表示为：

```
if (Ball_location & Danger & Opponent_Location)
then Decision {
    (Role1, Task1, Location1),
    (Role2, Task2, Location2),
    ....
    (RoleN, TaskN, LocationN)
}
```

在MiroSot 3Vs3比赛中，所用的场地的尺寸为150cm×130cm的长方形，所用机器人的尺寸为7.5cm×7.5cm的正方形。为了便于编制决策又不降低位置精度，将比赛场地划分为9×9的81个长方形格子，球的位置(Ball_location)用Ball_location(x,y)表示(其中1 ≤ x, y ≤ 9的无符号整数)；本方球门受威胁程度(Danger)用于考虑防守，现定义为3个值的一维变量：危险度高(3)，危险度中(2)，危险度低(1)；对方球员位置(Opponent_Location)的信息量虽然很大，但在比赛中起的作用较弱，现只考虑对方除守门员外的其余队员与对方底线的距离，定义为3个值的一维变量：靠前(3)，中间(2)，靠后(1)。由此可得相应的决策有9×9×3×3=729条。推理模型和决策也可表示为：

```
if (Ball_location(x,y) & Danger & Opponent_Location)
then Decision{
    (Role1, Task1, Location1(x1,y1)),
    (Role2, Task2, Location2(x2,y2)),
    ....
    (RoleN, TaskN, LocationN(x3,y3))
}
```

角色(Role)指定相应机器人的角色，其值定义为主攻队员-1、协攻队员-2、防守队员-3、守门员-4；任务 Task 指定相应机器人的任务，其值定义为踢球-1、射门-2、拦截-3、守门-4、避障-5；位置 Location(x,y)的定义与 Ball_location(x,y)的定义相同。

用户只要根据可视化的编程要求输入决策的前件即产生式推理的条件和相应的决策后件即产生式推理的结果，就可编制完成一条决策。而决策在决策库中以何种形式存储以及推理的过程是如何进行的，对用户来说是完全透明的。

对于本方球门受威胁程度(Danger)，做出如下规定：

规则1 若球在对方半场，则 Danger = 1；

规则2 若球在本方半场，守门员没有封住球的射门角度，且无防守队员位于球和球门之间，则 Danger = 3；

规则3 若球在本方半场，守门员挡住了球的射门角度，且防守队员位于球和球门之间，则 Danger = 1；

规则4 其余情况，Danger = 2。

对于对方球员位置(Opponent_Location)，做出如下规定：

规则1 若除守门员外的其余2个对方队员之间距离大于X_long/2，则 Opponent_distance 为离球最近的对方球员与对方底线距离，否则 Opponent_distance 为该2个对方球员与对方底线距离的平均值。

规则2 若 Opponent_distance > x_long/2，则 Opponent_Location = 3；

规则3 若 x_long/4 < Opponent_distance < x_long/2，则 Opponent_Location = 2；

规则4 若 Opponent_distance < x_long/4，则 Opponent_Location = 1；

其中，变量 Opponent_distance 定义为对方球员与对方底线的距离，X_long 定义为球场总长。

由于决策库中的决策很多，用户编制的决策库可能不能包含全部的729条决策。当进行推理时，可能没有相匹配的决策，此时会出现某个周期的推理中断，影响比赛。为了避免此种情况的发生，我们编写了基本决策集。当进行推理时，先在用户决策集中进行，当在用户决策集中未找到相应的决策时，再在基本决策集中推理。因此，把决策子系统的推理模型修改为图2所示。

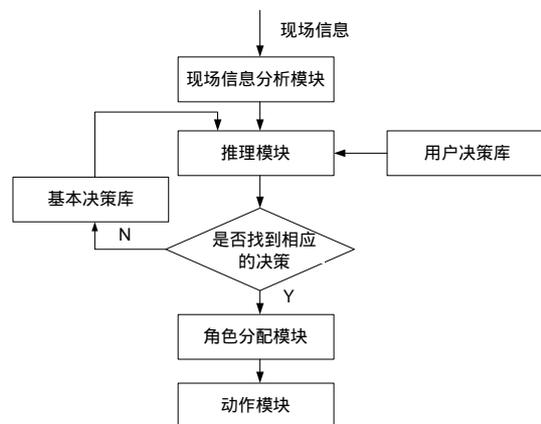


图2 机器人足球比赛系统决策子系统模型

3 决策编程的可视化

为了接收可视化界面中输入的决策的前件与后件，即推理的条件与结果，在客户程序中定义了两个结构体类型：

```
struct decision_result
{
    unsigned role;
    unsigned task;
    unsigned location_x;
    unsigned location_y;
};

struct decisiontype
{
    unsigned ball_location_x;
    unsigned ball_location_y;
    unsigned danger;
    unsigned opponent_location;
    struct decision_result role[3];
};
```

利用 struct decisiontype 结构体类型的变量，可以存储界面中输入的决策数据，从而把决策存储于决策库中。在进行比赛时，打开用户决策库，从决策库中传入决策进行推理。当用户决策库中没有对应的决策时，从基本决策库里选择相应的决策进行推理。

由图 3 可见，决策可视化编程由几个菜单项构成。菜单“新建决策集”用来新建一个决策集。点击该菜单，出现如图 4 所示的界面。此时，可以输入包含已存在的路径在内的文件名。如果路径不存在或文件已存在，则出现错误提示。缺省的路径为程序安装的路径。



图 3 决策的可视化编程界面



图 4 新建决策集界面

菜单“打开决策集”用来打开一个已存在的决策集，如果路径或文件不存在，则会出现错误提示。界面与图 4 类似。

菜单“关闭决策集”用来关闭一个已打开的决策集，如果路径或文件不存在，则会出现错误提示。界面与图 4 类似。

菜单“删除决策集”用来删除一个已存在的决策集，如果路径或文件不存在，则会出现错误提示。界面与图 4 类似。

菜单“添加决策”用于在某个决策集已打开的情况下，向此决策集添加决策。其界面如图 5 所示。



图 5 添加决策的界面

在“球的位置[1~9]”中，用于输入球的位置对(x,y)，其中 1 ≤ x,y ≤ 9，且 x,y 为无符号整数。球的位置对(x,y)可以有多个，表示球在这些位置时，决策是相同的。为简化输入，可以用数字字符串“11”表示球所在位置(1,1)，用“1234”表示球所在位置(1,2)与(3,4)。但表示球的位置的数字字符串中的数字不能超出9，也不能小于1，且字符串中的数字个数不能为0或奇数。其它的输入项目中的数字也各有其含义，已在上文中说明。假设我方的半场在球场右侧，图 5 所示的数据表示：当球在位置(4,4)或(2,3)、危险程度为1(危险程度低)、对手位置为1(对手位置靠近它自己的半场)时，机器人1的角色为1(主攻队员)、任务为2(射门)、位置为(2,3)，机器人2的角色为2(协攻队员)、任务为1(踢球)、位置为(4,

4)，机器人3的角色为4(守门员)、任务为4(守门)、位置为(8,4)。当添加的决策已包含在决策集中，则会出现提示；当添加的决策的前件即推理条件已存在时，也会给出提示，以防生成决策前件(推理的条件)相同而决策后件(推理的结果)不同的两条决策。

菜单“删除决策”用来删除一个已打开的决策集中的某条决策。其界面如图 6 所示。图 6 所示的数据表示删除与球的位置为(4,4)、危险程度为1、对方球员的位置为1的条件所对应的决策。如果没有与该条件相符合的决策则给出提示。

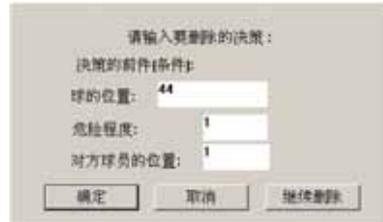


图 6 删除决策的界面

菜单“显示决策”用来显示一个已打开的决策集中的某条决策。其界面如图 7 所示。当输入球的位置、危险程度、对方球员的位置等数据后，点击确定按钮，在“决策”显示栏中会显示出与条件相对应的决策。显示的决策是决策库中的原型，它是标准的 C++ 语言形式，可以直接从决策库中调入客户程序供推理使用，这对客户来说是透明的。decision 是客户程序中定义的具有 struct decisiontype 类型的结构体变量。如果没有符合条件的决策，则给出提示。



图 7 显示决策的界面

菜单“显示决策集”用来显示一个决策集中的全部决策。其界面如图 8 所示。从结果中可以看出，显示的决策已按决策的前件(推理条件)中的球的位置、危险程度、对方球员的位置排序，这样对决策库中的决策一目了然。



图 8 显示决策集的界面