

FIRA 仿真足球机器人及其射门动作的智能化设计

赵学伟 赵英凯

南京工业大学 江苏 南京 210009

E-mail: gloomyccloud@163.com zhaoyk@njut.edu.cn

摘要: 本文主要简单介绍机器人足球 FIRA 系列仿真机器人的设计步骤以及神经网络在射门动作中的应用。

关键词: 机器人足球 FIRA 神经网络

FIRA SimuroSot Robot Soccer and the Intelligentized Designment of Shooting

Zhao Xue-wei Zhao Ying-kai

Nanjing University of Technology, Nanjing, China 210009

E-mail: gloomyccloud@163.com zhaoyk@njut.edu.cn

Abstract: The disquisition mostly introduces the designment process of SimuroSot in the FIRA series robot soccer and the application of Neurotics Nomine(NN) in the designment of shooting.

Key Words: robot soccer FIRA Neurotics Nomine (NN)

1. 足球机器人简介

机器人足球赛是由硬件或仿真机器人进行的足球赛，比赛规则与人类正规的足球赛类似。硬件机器人足球队的研发涉及计算机、自动控制、传感与感知融合、无线通讯、精密机械和仿生材料等众多学科的前沿研究与综合集成。仿真机器人足球的研究重点是球队的高级功能，包括动态不确定环境中的多主体合作、实时推理—规划—决策、机器学习和策略获取等当前人工智能的热点问题。

国际上机器人足球赛有两个分支，一个是 FIRA，一个是 RoboCup。它们的最大区别就是 FIRA 系列的是集中控制机器人的动作，策略。而 RoboCup 系列是分布式控制。另外，FIRA 是小车的碰撞模型，RoboCup 是以人为模型。图 1 是一般 FIRA 硬件机器人的原理图。

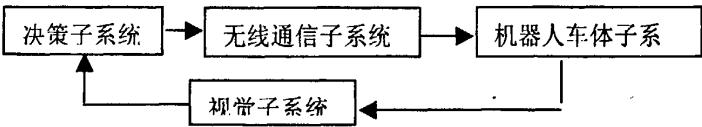


图 1 足球机器人系统

2. FIRA 仿真机器人设计

FIRA 仿真一般分为 5vs5 和 11vs11 两种，有服务器和客户端，其中服务器由主办方提供，现在 11vs11 用哈尔滨工业大学提供的服务器，5vs5 用的是澳大利亚的一所大学所提供的三维服务器。而客户端是由参赛方自己编程，11vs11 的 Client 与 Server 通过 UDP/IP 协议通信，5vs5 是通过 DLL 的方式与 Server 连接。现在大多数是用 Visual C++ 语言编写。一般的，编制控制策略都要经历以下步骤：

1) 入信息预处理

包括区域划分、判断控球者、计算实体距离、区域检测数、选取传球者等。

2) 态势分析与策略选择

对输入信息进行预处理后, 根据赛场形势, 及时调整比赛策略。一般球场上使用的全攻全守、自由人战术和人盯人策略等。

3) 队形确定与角色分配

在确定好策略以后, 为了使足球机器人在比赛中相互配合、协调作战, 更好的完成各自的任务而不至于出现冲突, 需要给每个机器人分配不同的角色。一般根据球位置及球员位置进行角色分配。确定好机器人角色后, 就要对机器人的动作进行设计和安排, 根据动作函数调用关系及复杂程度把动作分为三层: 基本动作, 技术动作, 战术动作。

在足球机器人比赛中, 跑位和配合是很重要的。这就需要自己很好进行球场划分和队形选择。在 5vs5 中, 球场划分如下图虚线所示。

假设己方机器人是向右方进攻。当采用的是全攻全守策略时, 一个机器人带球主攻, 其他机器人要抓紧时间站好位, 一般都要站到 O1、O2、A、B、C 几个点上。假设带球的机器人在 O1 点, 其他机器人应该在 O2、A、B 点上, 那么带球的机器人就可以直接带球射门, 或者传给 A 点或 B 点的机器人, 由其射门, 也可以传给 O2 点的机器人, 由它进行传球或者传给其他机器人射门。防守时, 机器人退到 O、O3、O4 以及 G 点进行防守, 若对方机器人已经到达 F 点, 则 G 点机器人和 O3 点机器人可以很快的到达 F 点进行防守, 若是防守成功, 己方得球, 又可以很快的发动进攻。

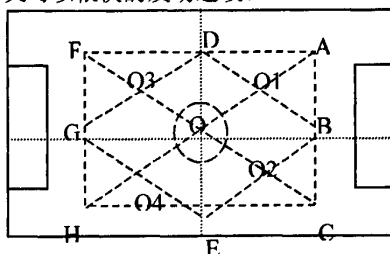


图2 机器人规划

当把机器人分为进攻机器人和防守机器人时, 应该使机器人迅速抢到 O1、O2、O3、O4 四个点。进攻时, 假设 1 号进攻队员在 O1 点得到球, 同时 2 号队员迅速到达 B 点, 由 2 号队员射门。防守时, 站在 O3、O4 点的防守机器人可以很快的到达 F、G、H 等出防守。按上述图形中的站位可以进行许多技术动作的实现。

3. 足球机器人射门函数的智能化

足球机器人的许多函数需要自己构建, 在构建函数和策略的过程中, 可以利用智能控制方法来实施。在算法设计和路径规划上, 可以考虑用最优控制、自适应控制、模糊控制以及神经网络等方法。下面简要说明用神经网络实现机器人射门动作。

多层前项神经网络是最常用最流行的神经网络模型, 它的逼近能力和训练算法是其应用的关键。误差反向传播 (BP) 算法是著名的多层前馈网络训练算法, 尽管存在收敛速度慢、存在局部极值和泛化能力有限等缺点, 但因其简单、易行、计算量小、并行性强等特点, 目前仍是多层前项神经网络训练的常用算法。

如果要求实现射门动作, 应该要求球在所划分的射门区内, 然后要求离球最近的机器人到达合适的位置, 选择合适的角度射门。此时需要考虑的因素包括球的位置、守门员的位置、以及对方队员的位置等。下面以 5vs5 为例来实现 BP 网络的机器人射门动作。

先让各个机器人在球场上排好, 把要射门的机器人摆到要射门的区域, 当机器人离球的距离最近时, 机器人会到合适的位置 (指机器人在球与球门的某一点的连线上, 而且连线不经过对方队员与守门员), 调整角度射门, 球进就算射门成功。射门的角

虑球的速度。然后就可以让机器人以最大的速度冲向球，就能完成射门动作。如果射门成功，就将本次样本存入训练样本库，如果射门不成功，则进行下一次训练。

通过上述方法进行 3000 次实验，总共射门成功为 500 次，则训练样本为 500。训练样本越多越好，底下用 BP 网络进行权值选择的普遍性会更好，能适应更多的球场变化。

选择好要射门的机器人后就可以利用 BP 网络进行训练。该 BP 网络的输出层的维数应该为三维，即准备射门的机器人要运动到的合适的位置坐标 $pos_e.x$ 、 $pos_e.y$ 和射门的角 ang_e 。

输入的维数包括球的位置坐标 $pos_ball.x$ 、 $pos_ball.y$ ，球的速度 vol_ball ，球运动的方向 ang_ball ，对方守门员的位置坐标 $pos_goalie.x$ 、 $pos_goalie.y$ 、运动方向 ang_gaolie 和运动速度 vol_gaolie ，四个对方队员的位置 $pos_opp[num].x$ 、 $pos_opp[num].y$ 及其运动速度 $vol_opp[num]$ 和运动方向 $ang_opp[num]$ ，四个己方队员的位置 $pos_home[num].x$ 、 $pos_home[num].y$ 及其运动速度 $vol_home[num]$ 和运动方向 $ang_home[num]$ ，（其中 $num=1, 2, 3, 4$ ）等 40 个参数，故是 40 维的。

维数太多会影响计算的速度，增加网络的复杂性，因此我们应该把那些对射门影响不大的值删去。既然是射门动作，己方的其他机器人的运动状态可以不考虑，这样就可以把维数减为 24 维。

隐含层是 BP 网络一个重要的特征，而隐含层节点数一直没有好的方法可以确定，现在都是实验的方法再加上个人的经验来选择。这里，我们选择节点数为 10。

确定好输入、隐含层和输出的维数分别为 24、10 和 3 后，就可以进行样本学习了。

1) 样本学习法

假设输入为 $X = (pos_ball.x, pos_ball.y, vol_ball, ang_ball, pos_goalie.x, pos_goalie.y, vol_goalie, ang_gaolie, pos_opp[num].x, pos_opp[num].y, vol_opp[num], ang_opp[num], pos_home[num].x, pos_home[num].y, vol_home[num], ang_home[num])$ ，其中 num 为 1, 2, 3, 4。输出为 $Y = (pos_e.x, pos_e.y, ang_e)$ 。

设节点之特性为 Sigmoid 型，即满足函数 $f(x) = \frac{1}{1+e^{-x}}$ 。任一节点 i 之输出为 O_i 。设

其 n 个输入为 X_{ij} ，权值为 W_{ij} ， $j=1, 2, \dots, n$ ，阈值为 θ_i ，则可认为其输入为

$net_i = \sum_{j=1}^n W_{ij} X_{ij} - \theta_i$ ，于是， $O_i = f(net_i)$ 。为方便起见，将阈值 θ_i 并入到 W 中，令

$W_{i(n+1)} = -\theta_i$ ， $X_{i(n+1)} = 1$ ，这样 $net_i = \sum_{j=1}^{n+1} W_{ij} X_{ij}$ 。

有 $N=500$ 个样本 (X_k, Y_k) ， $k=1, 2, \dots, N, N+1$ 。使用平方型误差函数

$$E = \frac{1}{2} \sum_{k=1}^N (y_k - \hat{y}_k)^2 \quad (1)$$

其中 \hat{y}_k 为网络之实际输出，定义单个样本 k 的误差

$$E_k = \frac{1}{2} (y_k - \hat{y}_k)^2 \quad (2)$$

对某一输入 X_k ，隐含层每个节点 i 之输出为 O_{ik} ，则

$$net_{ik} = \sum_{l=1}^{24+1} W_{il} X_{lk} \quad (3)$$

$$O_{ik} = f(net_{ik}) \quad (4)$$

$$\delta_{ik} = \frac{\partial E_k}{\partial X_{ik}}$$

于是

$$\frac{\partial E_k}{\partial W_{il}} = \frac{\partial E_k}{\partial net_{ik}} \frac{\partial net_{ik}}{\partial W_{il}} = \frac{\partial E_k}{\partial net_{ik}} X_{ik} = \delta_{ik} X_{ik}$$

所以有

$$\begin{cases} \delta_{ik} = f'(net_{ik}) \sum \delta_{jk} W_{ji} \\ \frac{\partial E_k}{\partial W_{il}} = \delta_{ik} X_{ik} \end{cases} \quad (5)$$

其中 $i=1,2,\dots,10$, $l=1,2,\dots,24,25$ 。

输出层的每个节点 j 的输出为 Y_{jk} , 则

$$net_{jk} = \sum_{m=1}^{10+1} W_{jm} O_{mk} \quad (6)$$

$$Y_{jk} = f(net_{jk}) \quad (7)$$

$$\delta_{jk} = \frac{\partial E_k}{\partial net_{jk}}$$

$$\frac{\partial E_k}{\partial W_{jm}} = \frac{\partial E_k}{\partial net_{jk}} \frac{\partial net_{jk}}{\partial W_{jm}} = \frac{\partial E_k}{\partial net_{jk}} O_{mk} = \delta_{jk} O_{mk}$$

于是有

$$\delta_{jk} = \frac{\partial E_k}{\partial Y_{jk}} \frac{\partial \hat{Y}_{jk}}{\partial net_{jk}} = -(Y_{jk} - \hat{Y}_{jk}) f'(net_{jk}) \quad (8)$$

其中 $j=1,2,3$, $m=1,2,\dots,10,11$ 。

在此, 网络分为 3 层, 第 3 层为输出节点, 第一层为输入节点, 则反向学习算法之步骤如下:

- (1) 选定权值系数之初值为 W ;
- (2) 重复下述过程直到收敛。

① 对 $k=1$ 到 N

a、先利用公式 (3)、(4) 计算 net_{ik} 、 O_{ik} , 再利用公式 (6)、

(7) 计算 net_{jk} 、 \hat{Y}_{jk} (正向过程)

b、先由公式(8)计算输出层的 δ_{jk} ，再由公式(5)计算隐含层的 δ_{lk}

(反向过程)

② 修正权值

$$W_{il} = W_{il} + \mu \frac{\partial E}{\partial W_{il}} \quad (9)$$

$$W_{mj} = W_{mj} + \mu \frac{\partial E}{\partial W_{mj}} \quad (10)$$

其中

$$\frac{\partial E}{\partial W_{il}} = \sum_{k=1}^N \frac{\partial E_k}{\partial W_{il}} \quad (11)$$

$$\frac{\partial E}{\partial W_{mj}} = \sum_{k=1}^N \frac{\partial E_k}{\partial W_{mj}} \quad (12)$$

其中 $i=1,2,\dots,10$ ， $l=1,2,\dots,24,25$ ， $m=1,2,\dots,10,11$ ， $j=1,2,3$ 。

2) 随机学习法

当样本很多时，上面的方法的计算量是很大的，主要是 $\frac{\partial E_k}{\partial W}$ 的计算。它必须计算每一

个样本的 $\frac{\partial E_k}{\partial W}$ ，然后相加。为了简化计算，每步仅计算随机抽取样本的 $\frac{\partial E}{\partial W}$ ，这就是随机

学习的思路，其步骤如下：

(1) 任选权值系数的初值 W ；

(2) 重复直到收敛：

a、随机选取任一整数 $k_t \in \{1,2,\dots,N\}$ ；

b、计算瞬时梯度向量 $\frac{\partial E_{k_t}}{\partial W}$ ；

c、修正权值： $W_i = W_i - \alpha_t \frac{\partial E_{k_t}}{\partial W_i}$

其中 t 为迭代次数， α_t 随 m 值减少，一般取 $\alpha_t = \frac{1}{t}$ 。

3) 集中学习法

与此相关的是 Widrow-Hoff 法，设输入按时间为一个序列 $x_1, x_2, \dots, x_k, \dots$ ，计

算 $\hat{y}_k = \sum_{i=1}^N x_{ki} W_i$ ，它是要求输出 y_k 的近似值，则权值按下式修正：

$$W_i = W_i - \mu x_{ki} (\hat{Y}_k - Y_k) \quad (8)$$

4) 加快反传算法的收敛速度
反传学习中, 权系数修正公式为

$$\Delta W_i(k+1) = \eta \delta_i o_j + \alpha \Delta W_i(k) \quad (9)$$

其中 k 为所加样本数, η 为学习步长, δ_i 是误差, 对输出单元和内部单元 δ_i 是不一样的, 按照这种方法学习有如下缺点:

- (1) 有的步对某些样本会使误差减少, 而对其他样本则不然, 因而使迭代次数增加。
- (2) η 值不易选, 太大会导致振荡, 太小又使收敛速度减低。

改进的算法如下:

第一种方法是动量修改法。并不是在每给一个样本后就立即修改权值系数, 而是所有样本施加后, 对每个样本效果计算误差, 权值系数调节量与误差总和有关。

$$\Delta W_i(m+1) = \eta \sum_k \delta_{ki} o_{kj} + \alpha \Delta W_i(m) \quad (10)$$

注意此处 m 为迭代次数而不是样本号, 右端是对所有样本求和。

第二种方法是学习系数的最优化选择。 η 选择根据误差是否降低而定。先取 $\eta = 0.01$, $\alpha = 0.8$ 。若某次修正使误差减少, 则在下一步时 η 乘以 1.2; 若某一步使误差增大, 则放弃这一次修正, 把 η 乘以小于 0.8, 并置 $\alpha = 0$ 则重复一次, 直到又得到使误差减少时, α 恢复原值。

下表是用自带的射门函数进行射门和用了优化 η 的神经网络之后的射门函数进行射门射门成功率的比较。由此可以看出用了神经网络后射门动作的成功率大大提高。

表 1 射门函数比较表

	普通射门函数	优化 η 的神经网络的射门函数
射门次数	300	300
射门成功数	87	213
射门成功率	29%	71%

4、结束语

足球机器人涉及学科广泛, 是进行信息学科研究和教学的很好的平台, 又是人工智能的很好的切入点。它是以体育竞赛为载体的前沿科研竞争和高科技对抗, 是培养信息—自动化科技人才的重要手段, 同时也是展示高科技进展的生动窗口和促进科技成果实用化和产业化的新途径。

参考文献

- [1] The Laws of the Game, Micro-Robot World Cup Soccer Tournament, 1997
- [2] 韩学东, 基于人工神经网络的足球机器人分层学习研究, 计算机工程与应用, 2001.23
- [3] 王耀南著, 智能控制系统, 湖南大学出版社
- [4] 佟国峰 赵姝颖 吴国刚, 足球机器人——科技竞技场, 电子科技导报, 1999 年第 4 期
- [5] 洪炳熔 薄喜柱, 机器人足球比赛策略仿真系统的开发, 计算机应用研究, 1999 年第 3 期

FIRA仿真足球机器人及其射门动作的智能化设计

作者: 赵学伟, 赵英凯

作者单位: 南京工业大学 江苏 南京 210009

本文链接: http://d.g.wanfangdata.com.cn/Conference_6737632.aspx