

一种基于 MAS 的 Robocup 测试床

A Testbed based on Multi-Agent Systems

张双民 李 毅 石纯一

(清华大学计算机科学与技术系 北京100084)

Abstract Robocup is one of the hot topic in the field of Artificial Intelligence and Automatic Control. In order for a robot to play a soccer game reasonably well, many technologies need to be integrated and a number of technical breakthroughs must be accomplished. The range of technologies spans the gamut of intelligent robotics research, including design principles of autonomous Agents, multi-Agent collaboration, strategy acquisition, real-time reasoning, and sensor-fusion, etc. Just stand by the multi-Agent systems, building a test-bed for problem solving of Robocup can improve multi-Agent system research. This paper introduces the competition mode, two typical Robocup systems and the test-bed based on MAS we designed.

Keywords Robocup, MAS, Agent

1 Robocup

国际机器人足球赛(The Robot World Cup),简称 Robocup,是人工智能和机器人研究的一种集中体现,特点在于给研究者提供了一个把他们的研究成果以竞赛方式表现出来的机会,实验的环境是动态的、互相敌对的、又需符合一定的标准。Robocup,为不同的理论、算法和 Agent 体系结构提供了一种评价的手段,为了使一个机器人(一个真机器人或者软件模拟的实体)能够比较合理地进行足球比赛,要求具备许多成熟的技术并且还需完成技术上的突破,涉及到的技术涵盖了机器人研究的所有领域,以及自治 Agent 的原理,如多 Agent 的合作、Agent 体系结构、联系反应方法并为这些方法建模或规划、实时识别、规划和推理、在动态环境中推理和行动、传感系统、多 Agent 系统、复杂任务的行为学习、决策产生机制等。

Robocup 包括三种竞赛方式:软件仿真、小场地的真实机器人比赛和中场地真实机器人比赛,每种竞赛方式受自身体系结构的限制,因此研究的内容有别于其它的比赛方式。在软件仿真的方面,需研究 Agent 之间的 Teamwork,包括底层的技术如将球传给队友和高层的技术如执行球队策略。Agent 建模,从基本技能如识别 Agent 传球的意图到高层队伍策略的复杂规划识别。多 Agent 学习,包括在线或离线对简单足球技能的学习,如传球或干扰对方以及更多的复杂的对策学习等。

2 两个 Robocup 系统

2.1 CMUnited-97系统

CMUnited-97是卡内基-梅隆大学于1997年开发的参加 Robocup-97比赛的系统,使用了合理的视觉算法,实现了在不确定和动态环境下机器人之间相互合作的算法,机器人能够自我组织阵形,遵守特定规则,并执行目标动作。在比赛过程中,能根据各种不同的环境产生不同合作方式,并能利用规则使球队获得最大受益。其视觉跟踪算法能跟踪多个移动物体和预测运动轨迹,系统 Agent 的行为,包括低层的独立行为和协商合作,具有战略性质的团队行为。CMUnited-97在 Robocup-97大赛上赢得了小场地机器人比赛的冠军,图1是参赛的 CMU97的比赛队伍。



图1 CMUnited-97的队员

2.2 哈尔滨工业大学机器人足球赛仿真系统

哈尔滨工业大学的 HT-1系统,参加了国内首届机器人足球比赛,HT-1是机器人足球比赛仿真系统,建立了机器人足球比赛仿真模型、必需的控制策略以

及对策略有效性的评价,实现了修正的区域防守策略和全攻全守策略,可以用于1对1和3对3的机器人足球比赛,三维计算机仿真画面如图2所示。

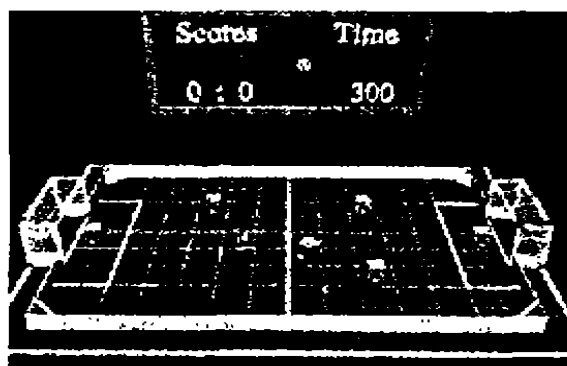


图2 HI-1型机器人仿真系统

3 Robocup 测试床

3.1 依 MAS 来研究 Robocup

以 MAS 为工具来研究 Robocup 是一种合理的选择。建立一个 Robocup 测试床有助于进一步分析研究。在 MAS 中要考虑 Agent 之间的差异性和协调控制的分散性,Agent 可以是各种具有自主性的系统,其行为方式可以相当复杂,在这种情况下 Agent 间的交互方式就由环境条件和 Agent 自身行为模式来确定。MAS 的研究任务之一就是确定何种条件会引导 Agent 用特定方法来行动,从而使 Agent 组成的系统表现出一定的特性。

Robocup 与 MAS 之间存在紧密的关系:

1) MAS 研究的对象是多个具有自治行为的 Agent,而作为 MAS 的基本元素 Agent,它能预测自己所处世界的发展趋势,能与其他 Agent 进行交互,能产生反应式行为从而达到自己的目的。Robocup 的元素是机器人,他同样具有交互和自主的行为能力,完全可以将一个机器人看作是一个 Agent 进行研究。

2) 在 MAS 研究中,Agent 之间的合作与对抗是普遍存在的。Agent 之间的通讯、Agent 之间的合作策略的执行、Agent 的合作规划以及对手模型的建立都是当前研究的热点问题。Robocup 同样存在合作与对抗,而且在比赛中,这种合作与对抗交织进行正好为研究 MAS 提供良好的测试环境。

3) 在 MAS 中每个 Agent 都有各自的特点,如何利用每个 Agent 不同的特点使 Agent 之间能更好合作也是 MAS 研究的重要课题。Robocup 每个队员也不相同,比如前锋速度快,后卫防守到位等,要掌握每个队员的特点,才能取得比赛胜利。

4) MAS 具有开放性,可依实际环境的变化有效实现其目标。Robocup 就是一个典型的开放式的 MAS 环境,场上的状态是实时改变的,机器人要根据场上状态自主修改规划型的开放式的 MAS 环境,场上的状态是实时改变的,机器人要根据场上状态自主修改规划,根据不同对手自主选择对抗策略和合作策略等。

Milind Tambe 等人认为国际象棋一直是人工智能的典型问题,但国际象棋是回合制、静态和集中控制的,而 Robocup 是动态、实时和分布控制的,因此可将 AI 的研究统一于以 Robocup 为背景的测试床之上。

建立一个 Robocup 测试床,为构造实际的 Agent 足球队,全面展开自主的机器人足球赛研究提供了可能。Robocup 测试床应是以机器人足球赛为运行环境,需考虑:

1) 规定标准的基本动作集合: {跑动,带球,踢球,断球,铲球,射门,扑救,掷界外球,...}。不同的动作有不同的属性,如跑动属性有速度、方向;踢球属性有力量、角度、高度等;扑救属性有方向、距离等。标准动作的设定为 Agent 设计提供了方便。

2) 规定比赛规则,规定越位、任意球、点球、角球、进球、中圈开球等规则,提供判定算法。

3) 提供标准的环境感知机制。为 Agent 提供标准的环境感知接口,Agent 可实现感知能力,以获得外部信息。测试床获得所有的 Agent 和外部环境的信息,并按一定的规则 (Agent 的位置、角度) 提供给各有关 Agent。

4) 运行平台。提供比赛环境,控制比赛进行、执行 Agent 产生的基本动作;设置环境的改变 (Agent 位置、足球的位置);按比赛规则,决定比赛事件的产生,如犯规、进球等。

5) 通讯机制。测试床应严格将各 Agent 分开,并消除中心控制存在的可能。Agent 间的信息交换提供三种环境: Agent 不进行显式的通讯,Agent 可进行含有噪音的无噪音的显式通讯。

3.2 一个简单的 Robocup 测试床

3.2.1 设计思想 本文设计了一个基于 MAS 的 Robocup 小型系统,进行了以下几个方面的工作: ①自主设计数据结构,提供良好的用户接口。在程序中,提供了良好的参赛平台,用户只需要设计球队规划,并提交规划即可。②能在程序中自主修改球队的指标参数,使用户能在不同的条件下进行各种实验。③能动态显示当前场上状态,有利于实时监控。

为了简化一些不必要的通讯操作,加强 MAS 的相互合作和对抗的特性,程序仍采用中央控制方式,所有队员的行为执行都由中央控制器进行处理,而队员的行为是自身向中央控制器统一提交的。简单来说,每

隔一段时间,场上的每名队员会根据自己的情况,产生行为或规划,同时把这些信息或行为提交给控制器,控制器根据提交的规划,统一执行每个队员的动作。再隔相同时间,仍进行相同的操作,中央控制系统与球队唯一的关联就是球员的规划请求,也就是说,如果想创建一只自己的球队,只需要每隔一定的时间段,提交每个队员的规划或动作即可,这样方便了用户的设计,使其能忽略大量无关的界面设计与具体执行,只需要关注球队的规划算法。

球队在场上可以获取的信息存储在中央控制器中,并且由中央控制器进行更新,球员可以根据这些信息做出自己的规划,这些信息是根据真实场上状态设计的,真实球员在场上能得到的信息基本上都得到反映,如球的位置和场上每个队员的位置等。在场上,队员还可以看见其他队员奔跑方向及球的运行轨迹,以一个数组存储各个时期球的位置和球速等特征。而持球队员是比较重要的信息,也是队员所应该知道的,所以对持球队员另加标注。这些信息基本上涵盖了真实场上的全局状态。

中央控制器负责维护这个全局状态结构,在每个时间片过后,中央控制器执行完所有队员的行动后,更新全局状态结构,使其符合当前场上的状态,如图3。

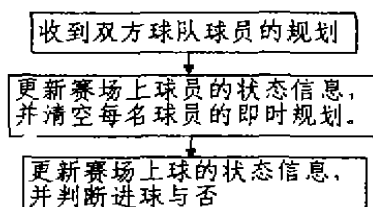


图3 中央控制流程图

3.2.2 程序体系结构设计

1) Agent 个性设计。在设计机器人球员的时候,考虑到不同位置的球员特性不同,对每个球员都设计了一套个性体系,包括球员的速度、传球技术、铲球技术、射门技术等。在计算中,会根据每个球员的不同素质,安排不同的规划。

2) 全局状态设计。因为 Robocup 是一个开放式的系统,要求球员能获悉场上的状态修改自身信念。球员所能获取的信息由于受到远近和前后的影响,只能是有限的信息。在设计中,考虑到这一点,根据球员场上的位置,设计出每个球员所能获得的场上状态信息,包括当前球的位置、球速估算、球的运动方向、所能看到的本方和对手的队员的场上位置等。

3) Agent 的规划设计。球员从状态结构中获取所需要的信息后,根据自己的规划算法,提出自己在这个时间片的规划。能选择的规划包括跑向目标点、接球、

抢球、射门等。中央处理器所能处理的动作有六种,基本涵盖了真实情况的所有行为。

3.2.3 Robocup 系统的 BDI 模型描述

1) Agent 模型。足球赛是典型的 MAS 问题,为有效表示 Agent 思维状态和过程引入 BDI 模型,Agent 模型关键还在于行为规划库 P 的表示。Agent 模型描述:

$$\begin{aligned}
 P &= P_1 \cap P_2, P_1 = \{p_i\}, p_i = \{a_{i1}, a_{i2}, \dots, a_{in}\}, n \geq 1, P_2 = \{g_i\} \\
 B &= \{f_i\}, f_i = (d_i, E, GS) \rightarrow R \\
 D &= \{d_i\}, d_i \in P, \text{即 } D = P \\
 I &= \{i_1, i_2, i_3\} \in P
 \end{aligned}$$

其中, P 是行为规划库, P_i 是个体行为规划, P 则是群体战术规划, p_i 是一个基本动作序列, g_i 为战术行为, B 为体现为判断规则和计算公式; D 为各种具有潜在可能的比赛行为; I 为决定实施的比赛行为, E 为环境, GS 为全局战术参数。在 Agent 思维过程中,利用 B 中的规则和当前环境 E 以及当前全局战术 GS , 对 D 中的各种可能行为评估,得到最优行为,并将其设定为 I , 并转化为行动 Action。如一个持球 Agent 可能的行为有向队友传球、向区域传球、射门、带球移动、战术执行等,利用 B 中的规则和公式对所有可能行为计算,得到最优行为,并执行。

2) Agent 动作。Agent 在场上产生的动作描述包括动作的形式化描述, Agent 产生该动作前的自身的必要状态 $Pre(A)$, Agent 成功执行该动作后的状态 $Post(A)$, 这样使该方法可以描述一些具体的 Agent 特定动作了。

Agent 接球:

- (1) $GetBall(A, Pos(x, y))$, 其中 A 表示 Agent, $Pos(x, y)$ 表示当前球在场上的坐标位置。
- (2) $IsNear(A, Ball) = 1$ and $IsBallBelong = 0$, 其中 $IsNear(A, Ball)$ 表示 Agent A 是否离球很近, 1 表示近, 0 表示远。 $IsBallBelong$ 表示当前球是否被控制, 1 表示球一定被其中一方控制, 0 表示没有。
- (3) $IsBallBelong = 1$ and $BallSpeed = 0$, 其中 $BallSpeed$ 表示球速。

Agent 传球:

- (1) $TransBall(A, Pos(x, y))$, 表示 Agent A 要把球传到 (x, y) 点。
- (2) $HasBall = A$ and $Know(B, TransBall(A, Pos(x, y)))$, 其中 $HasBall$ 表明当前球被哪个 Agent 控制, $Know(B, S)$ 表示 B 知道 S , 其中 S 是一种状态。
- (3) $Ball(x, y) = Pos(x, y)$, 其中 $Ball(x, y)$ 表示球的坐标位置。

Agent 射门:

- (1) $ShootBall(A)$,

(2) HasBall=A and IsNear(Gate)

(3) Score=Score+1 or IsBallBelong=Goalkeeper. 其中 IsBallBelong=GoalKeeper 表示球被守门员救起。

3) 对手模型。了解对手是对抗活动的基本前提,建立对手模型也成为了机器人足球赛的一个基本问题。基于 BDI 模型,在 Anand S. Rao 等人工作基础之上,本文给出了对手 Agent 模型和相应算法,以达到在动态开放环境下,实时跟踪对手的思维状态,为决策提供支持的目的。

将 Agent 自身视为 Agent1,将被跟踪的 Agent 视为 Agent2,则 Agent1 所建立的 Agent2 对手模型元组描述为 $O = \langle B, D, I, I_p, I_w, H \rangle$, O 在 Agent1 的 BDI 模型中应是一组 Belief,即 $Bel(\text{Agent1}, O)$,各元组分别为:

B 任意 $\Psi \in B = \{ \text{Bel}(\text{Agent2}, \Psi) \}$,集合 B 包含了 Agent1 知道的 Agent2 所具有的 Belief;

D 任意 $\Psi, Des(\text{Agent2}, \Psi) = \{ \Psi \in D$, Agent2 实际的集合是 D 的一个子集,即 D 包含了 Agent2 所有可能的 Desire;

I_s 任意 $\Psi \in I_s = \{ \text{Intention}(\text{Agent2}, \Psi) \}$, I_s 表示可以确定的 Agent2 的 Intention,包括先验的 Intention 和通过观察、通信等手段获得的 Intention;

I_p 对手的可能 Intention,通过已有 BDI 信息所推断出当前 Agent2 可能具有但无法确定的 Intention;

I_w 表示在 Agent2 所有可能的 Intention 中 Agent1 所关心的部分,可将 I_w 称为 Intention 的观察窗口,对于该集合之外 Intention 的,模型不考虑;

$H = \{ a_i \}$, a_i 表示 Agent2 在特定时间所执行的特定动作。 H 记录一定时间内的对手事件。

约定 O 中的 B, D, I 满足典型的 BDI 约束关系;对于任意 $\Psi \in I_p, B, D, I + \{ \Psi \}$ 也满足典型的 BDI 约束关系,但模型中, B, D, I_p 无须满足 DBI 约束,即 I_p 集中的 Intention 可以相互矛盾。

模型的实现算法:

Step0 初始化操作,将模型各元素置初始状态;

随后,系统按 Step1 至 Step4 顺序循环执行。

Step1 获得外界信息,修正对手模型中的 B, D, I_s 元素。

Step2 根据 B, D, I 的约束关系,修正 I_p ,消除思维状态的内部矛盾。

Step3 将可能的 Intention 加入 I_p 。

Step4 在 I_p 中存在多个对手 Intention,它们均是不确定的,为了有效跟踪对手,应对其进行排序,以便为自身决策提供依据。对可能意图的排序的方法应依不同具体情况而定,通常,我们可考虑意图实现的代价和对手的历史信息等因素。

参考文献

- Asada M. Robocup: Today and tomorrow—What we have learned. Osaka University, 1998
- Veloso M, Stone P, Han K. The CMUnited-97 robotic soccer team: Perception and multi-agent control. Carnegie Mellon University, 1998
- 洪炳熔, 薄喜柱. 机器人足球比赛策略仿真系统的开发. 哈尔滨: 哈尔滨工业大学计算机系智能机器人研究室, 1999
- Marsella S. On being a teammate: Experiences acquired in the design of Robocup teams, University of Southern California, 1998
- Kitano H, Asada M. Robocup: The Robot World Cup Initiative
- Tambe M. Building agent teams using an explicit teamwork model and learning. Artificial Intelligence, 1999 (110): 215~239
- Stone P, Veloso M. Task decomposition, dynamic role assignment, and low-bandwidth communication for real-time strategic teamwork. Artificial Intelligence, 1999(110): 241~273
- Werger B B. Cooperation without deliberation. A minimal behavior-based approach to multi-robot teams. Artificial Intelligence, 1999(110): 293~320
- Haddadi A, Sundermeyer K. Belief-Desire-Intention Agent Architectures, in Foundations of Distributed Artificial Intelligence, O'Hare G. M. P and Jennings N. R. (eds), John Wiley & Sons Inc, 1996. 169~185
- Rao A S. Multi-Agent Mental-State Recognition and its Application to Air-Combat Modelling, in Workshop on Distributed Artificial Intelligence, 1994. 283~304
- Han J. Conference tutorial notes: Data Mining Techniques, In: Proc. of ACM 'SIGMOD Int'l. Conf.' 96 on Management of Data (SIGMOD'96) Montreal, Canada, June, 1996
- Data Mining: Discovering Hidden Value in Your Data Warehouses. Pitwt Software. WWW files, 1996. Available at <http://www.pitwt.com/dmpaper/dmindex.html>
- C Zhenxin. Knowledge Discovery and System-user Adversary. Kybernetes: An International Journal of Cybernetics and General Systems, 1997, 26(4): 425~441
- Weldon Jay-Louise. Data Mining and Visualization. Database Programming and Design, 1996, 9(5): 21~24
- Fayyad U M, Piatetsky-Shapiro G, Smyth P. From Data Mining to Knowledge Discovery in Databases. AI Mag., 1996, 17: 37~54
- Fayyad U M, Piatetsky-Shapiro G, Smyth P. The KDD Proces for Extracting Useful Knowledge from Volumes of Data. Commun. ACM, 1996, 39: 27~41

(上接第81页)