**实验五 特殊比赛模式的设计**

**1. 实验目的**

（1）掌握 Robocup 仿真机器人足球比赛中特殊比赛模式发生的条件；

（2）掌握 Robocup 仿真机器人足球比赛特殊比赛模式的规则要求；

（3）了解 Robocup 仿真机器人足球比赛特殊比赛模式的战术设计思想；

（4）进一步熟悉 WorldModel 类。

**2. 实验设备**

硬件环境：PC 机

软件环境：操作系统 Linux

**3. 实验内容**

(1) 概述

Robocup 仿真机器人足球比赛特殊比赛模式包括角球（corner\_kick）、界外球（kick\_in）、定位球/任意球（free\_kick）以及球门球（goal\_kick）。

任务：

(2) 角球（corner\_kick）

当防守方球员将球踢出底线时，由进攻方开角球。Server 一旦接收到发球队员发出的 kick 命令后，就将比赛模式设为正常的 play\_on 模式。注意，和国际足联的规则类似，发球队员在其他球员接触球之前不能再触球，否则判犯规。此时，另一方球员须在一定时间（根据 Manual 的规定为 300 周期）内将球开出，否则算发球失误，由对方发定位球。

在 UVA\_trilearn 的原代码中，球员是不管这些特殊比赛模式的，统统是最近的球员跑向球，一脚踢向球门。这就需要我们手工编码完成这些细节地方。一种简单的设计思路是：（1）如果比赛进入角球模式，则离球最近的球员 A 跑向球，而次近的球员 B 跑到某个接应点，等待 A 将球传过来；（2）A 跑到球跟前（即进入可踢范围）时，不必立即将球开出，可以先看看场上环境，等 B 跑到预定位置并且体力恢复得差不多时再开球。

以上设计思路同学们在实验过程中只须完成（1）即可，对于学有余力的同学可以考虑（2）的实现。下面列出几个可能用到的函数：

PlayModeT WorldModel::getPlayMode() const 返回比赛模式（或者直接用 bool WorldModel::isCornerKickUs() 判断是不是我方开角球）；

ObjectT WorldModel::getClosestInSetTo(ObjectSetT objectSet, ObjectT o, double \*dDist=NULL, double dConfThr=-1.0) 返回在对象集合 objectSet 中距离对o 最近的对象，只有当对象的可信度高于给定的阈值才被考虑，如果没有给出阈值则使用 PlayerSettings 中定义的阈值，同时 dDist 返回距离；

ObjectT WorldModel::getSecondClosestInSetTo(ObjextSetT objectSet, ObjectT o, double \*dDist=NULL, double dConfThr=-1.0) 返回在对象集合 objectSet 中距离对

* o 次近的对象，只有当对象的可信度高于给定的阈值才被考虑，如果没有给出阈值则使用 PlayerSettings 中定义的阈值，同时 dDist 返回距离。

下面具体描述一下实现过程：

首先，我们知道 demeer5()主要是围绕三句话展开的，即（1）如果球可踢，则用最大力量踢球；（2）如果球不可踢且我是队友中最快到球的队员，则去截球；（3）其他情况按战略点跑位。我们可以围绕这三句话来实现角球策略。即在角球模式下（1）如果球可踢，则传球给接应球员；（2）如果球不可踢且我是队友中最快到球的队员，则去发球；（3）其他情况下，如果我是离球次近的队员，那么我去接应。

接着，有了这个基本思想后，我们开始编写代码。

角球是要求向场内踢。编写pointKickTo()返回在不同角时应该踢向的位置。

当球在自己半场时，将位置的X坐标调大，尽量向对方半场踢；而在对方半场时，X设置得比较小，为夺球攻门创造机会。

位置：basicplayer

VecPosition BasicPlayer::pointKickTo()//计算角球应该踢向的点

{

VecPosition posBall, posKickTo;

double x, y;

posBall = WM->getBallPos();

x = posBall.getX();

y = posBall.getY();

posKickTo.setX(- x / fabs(x) \* 5 + x );

posKickTo.setY(- y / fabs(y) \* 12 + y );

return posKickTo;

}

位置：demeer5

else if( WM->isCornerKickUs())//我方角球

{

if(WM->isBallKickable() )

{

soc = kickTo(pointKickTo(), SS->getBallSize());

}

else if(WM->getFastestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL, &iTmp)

== WM->getAgentObjectType())

{

soc = moveToPos(posBall, PS->getPlayerWhenToTurnAngle());

}

else if(WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES, OBJECT\_BALL)

== WM->getAgentObjectType())

{

soc = moveToPos(pointKickTo(), PS->getPlayerWhenToTurnAngle());

}

else

{

soc = moveToPos(WM->getStrategicPosition(), PS->getPlayerWhenToTurnAngle());

}

ACT->putCommandInQueue( soc ); // 放入命令队列

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

(3) 界外球（kick\_in）

当一方球员将球踢出边线时，比赛即进入界外球模式。比赛的规则要求基本同角球。

简单的设计思路和实验内容与角球也差不多，在此不再赘述。

要求同学们编写一段程序实现界外球策略，要求能够根据发球点的位置给出合适的接应球员的接应位置。

可能用到的函数：

bool WorldModel::isKickInUs() 返回是否是我方开界外球。

界外球和角球类似，区别在于比赛中很少有机会在对方底线发界外球。所以大部分情况都将接应点设置得更加靠近敌方半场，增加进攻机会。

位置：demeer5

else if( WM->isKickInUs() )//我方界外球

{

if( WM->isBallKickable() )

{

soc = kickTo(pointKickToK(), SS->getBallSize());

}

else if(WM->getFastestInSetTo( OBJECT\_SET\_TEAMMATES, OBJECT\_BALL, &iTmp)

== WM->getAgentObjectType())

{

soc = moveToPos(posBall, PS->getPlayerWhenToTurnAngle());

}

else if(WM->getSecondClosestInSetTo(OBJECT\_SET\_TEAMMATES, OBJECT\_BALL)

== WM->getAgentObjectType())

{

soc = moveToPos(pointKickToK(), PS->getPlayerWhenToTurnAngle());

}

else

{

soc = moveToPos(WM->getStrategicPosition(), PS->getPlayerWhenToTurnAngle());

}

ACT->putCommandInQueue( soc ); // 放入命令队列

ACT->putCommandInQueue( turnNeckToObject( OBJECT\_BALL, soc ) );

}

位置：basicplayer

/\* 此函数返回界外球应该踢向的点

\* 策略是把球尽可能往敌方半场踢

\*

\*/

VecPosition BasicPlayer::pointKickToK()//计算界外球应该踢向的点

{

VecPosition posBall, posKickTo;

double x, y;

posBall = WM->getBallPos();

x = posBall.getX();

y = posBall.getY();

posKickTo.setX( x + 12 );

posKickTo.setY(- y / fabs(y) \* 12 + y );

return posKickTo;

}

(4)定位球/任意球（free\_kick）

当一方球员犯规或违例时，由对方开任意球。跟国际足联的规定略有不同的是，Robocup 的任意球没有直接任意球跟间接任意球之分。

比赛的规则要求基本同界外球。

（1）如果是对方开球，我方在场上等待对方发球，比赛进入正常模式

（2）如果是我方发球，

如果球可踢 并且离对方球门比较近，尝试直接射门，否则把球传给队友

如果球不可踢，则离球最近的球员去发球，第二近的去接应（代码基本同 边界球）

所以在上面的基础上，在发球时加入判断离球门的距离：

Double dis;

Vection goal = {52.5,0};

Dis = getRelativeDistance(OBJECT\_GOAL\_R);

if (dis < 20)

ACT->putCommandInQueue(kickTo(goal, SS->getBallSpeedMax()\*0.8 ));

else

ACT->putCommandInQueue(kickTo(pointToKickTo(),SS->getBallSpeedMax()\*0.8));

(5).球门球（goal\_kick）（选做）

让守门员沿最近敌方球员、自己、次近敌方球员三人构成夹角的角平分线开球。

先求出最近敌方球员、次近敌方球员的位置，然后结合自己位置分别和他们连线，调用平分线函数求出平分点，往这个点开球。