

Robocup3D 仿真中时间通报信息的使用

时长娥, 许元

(东南大学 RoboCup 仿真训练基地, 江苏 南京 210096)

Email: evelineshi@sohu.com, xychn15@yahoo.com.cn

摘要: 主要研究 RoboCup3D 仿真中时间通报信息的实现和其性能的测试。本文提供了时间通报信息的具体实现, 做了相关的测试将使用时间通报前后的特性作一比较, 测试证明使用时间通报信息可以提高仿真的效率和踢球的精度。

关键词: 三维仿真机器人足球, 时间通报

Usage of TimeNotifyMessage in RoboCup3D

Shi Change, Xu Yuan

(RoboCup Simulation Group, SouthEast University, Nanjing Jiangsu 210096)

Email: evelineshi@sohu.com, xychn15@yahoo.com.cn

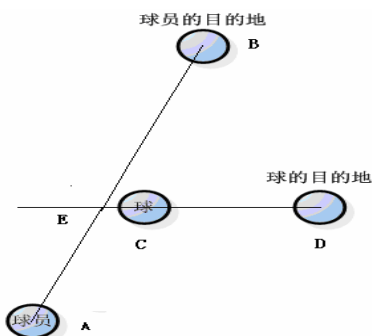
Abstract: This paper mainly describe the actualize of Time Notify Message in RoboCup3D simulation and tests of its performance. We did series of tests on simulation with Time Notify Message and without Time Notify Message. The results of tests show that using Time Notify Message can enhance the efficiency of simulation and precision of kick ball.

Keywords: RoboCup3D TimeNotify

1 引言

SPADES 实现“感觉—思考—动作”周期, 每个智能体接受感觉并返回动作。这就意味着智能体只能在接收到感觉信息之后有动作。因此, SPADES 提供了一个动作叫做请求时间通报 (request time notify) 的特性, 一个时间通报的实质是返回一个空的感觉(时间通报), 在接收到后智能体可以以一个动作回复。这允许智能体在一个特定的时间发送动作, 即使通常没有接收到感觉信息时。例如, 如果智能体接收到一个感觉在 100 周期并希望在 110 周期产生一个动作, 但是下一个感觉只有在 120 周期才会到达, 智能体可以要求在 110 周期接收到一个时间通报并在接收到之后发出命令。

这个特性有很多的应用, 比如 3D 比赛中是有高度的, 球员可以跳起来撞球 (这个特性在目前的服务器版本中还没有实现)。因为仿真是离散的, 智能体每 200ms 只能发送一条信息即只能完成一个动作, 可能在这个周期时球被踢出, 而下一个周期球已落地, 踢球的动作需要在两个感知信息之间完成, 这时候使用时间通报信息可以很好的解决这个问题。它的另一个典型应用是在踢球时。Robocup3D 比赛中踢球时球员、球、目的地必须满足三点一线, 也就是说球员必须首先运动球与目的地的延长线上并且在控球范围内才能踢球。如下图所示



在正常情况下，球员首先须运动至 E 点附近且满足精度要求时才能踢球，每一细微的调整将占用一个仿真周期，在情况较差时，这将浪费几秒的时间，大大影响了仿真效率。同时球员在到达目的地后需等待感知信息的到来，因此每一步的调整是近乎静止的，这也导致了球员速度和加速度的浪费。一个可行的方案是球员在往目标地运动的过程中完成踢球的动作，适当使用时间通报信息可以实现这个目标，这样既可减少资源的浪费，又可以在一定程度上提高精度。

2 实现

提供一个接口，参数是 SoccerCommand 类的对象。SoccerCommand 类的声明如下，它封装了一个时间参数，动作类型和相应的动作参数，并定义了相关的接口函数。

```
class SoccerCommand
{
float      time;
CommandT   CT;
Vector3f   drivePower;
double     kickAngle;
double     kickForce;
public:
void SetTime(float t){time = t;}
void SetCT(CommandT T){CT = T;}
void SetDrivePower(Vector3f p){drivePower = p;}
void SetKick(double a,double f){kickAngle = a;kickForce = f;}
float GetTime(){return time;}
CommandT   GetCommandT(){return CT;}
Vector3f    GetDrivePower(){return drivePower;}
double      GetKickAngle(){return kickAngle;}
double      GetKickForce(){return kickForce;}
};
```

在当前动作类型确定，动作参数确定，发送动作的时间确定的情况下，调用 SendActiong 函数，如下：

判断当前动作是否需要延时发送？

不需要：将当前动作发出

需要：1) 请求相应的时间通报信息

2) 将当前动作保存在一个全局变量中。

其中的请求时间通报信息的函数向 server 发送请求时间通报的“R”信息，其参数为发送当前感知信息的时间加上需要延时的时间。

在接收到服务器返回的“T”信息后，调用 SendTimeNotifyAction()函数将先前未到发送时间而保存下来的动作信息发送出去。在 SendTimeNotifyAction()函数中根据动作的类型获取相应的参数，并将动作发出。

3 测试

为了测试时间通报信息发送的正确性，在决策循环中加入请求时间通报信息的函数：

```
RequestTimeNotifyMessage(SEUWM. GetMsgSendTime());
```

观察记录的 log 文件，在每一次请求后均有一相应的“T”信息，并且没有发现“O”信息，也即已成功实现了发送和接收时间通报信息。

为检测使用这个特性的优越性，我们做了一系列的测试。考虑到仿真的复杂性，我们编写了测试队，让一个球员朝某一固定目的地踢球，在保证一定精度的情况下，比较调整时间。第一个测试是在不考虑球员速度的条件下测试的，即球员都是以近似于静止的方式朝目的地运动，测试结果如表一所示：

精度指三点一线的角度误差，单位是度；每组数据测试时间为半场时间（300S）；Sick 次数指半场时间里球员踢球的次数。总调整时间为球进入可踢范围内到球被踢出这一段时间。

表一：

方法 统 计 精 度	不使用时间通报			使用时间通报		
	Kick 次数	总调整时间	平均调整时间	Kick 次数	总调整时间	平均调整时间
90	55	7.4	0.1345	38	5.2	0.1368
60	54	16.2	0.3	45	15.6	0.3467
30	65	54.0	0.8308	57	30.4	0.5333
15	51	50.6	0.9922	40	85.8	2.145
10	49	73.4	1.4979	45	72.6	1.6133
5	55	82.8	1.5055	47	75.2	1.6
3	44	77.6	1.7636	47	94.0	2.0
2	36	97.8	2.7167	43	102.6	2.3860
1.8	30	108.2	3.6067	37	75.6	2.0432
1	31	118.0	3.8065	31	123.2	3.9742

由表格可以看到，使用时间通报并没有明显的效果。这是由于目前的位置调整动作在微调的时候，速度很小，时间通报的优点没有充分发挥，考虑可以适当提高力量。速度快了，

各仿真步的差别才能显示出来。所以在智能体踢球之前，即 kick 之前的运动，可在绕着球调整的状态下，提高速度。使用 GetPowerForVel 实现以恒定速度绕球转。

踢球的角度误差满足方差 0.02 的正态分布。经测试注意到，当精度要求过高时调整时间过长，这时失去意义。经多次测试，我们认为选择 1.8 作为角度精度的要求，调整时间适当且误差范围可以接受。

以下的测试设定精度 1.8，恒定速度 0.75，调整时间为球员到踢球一侧某一范围内到球被踢出的时间。测得的数据如表二所示：

表二：

使用时间通报				不使用时间通报			
比赛场次	总踢球次数	总调整时间	平均调整时间	比赛场次	总踢球次数	总调整时间	平均调整时间
4	143	6.6S	0.046S	4	53	83.8S	1.581S

4 结论

由表二可以看到使用时间通报信息可以大大减少踢球前的调整时间，从而提高了仿真效率。比较两种方法的总踢球次数，注意到不使用时间通报信息比使用时间通报信息少得多，平均每场比赛踢球 14 次，此时的精度要求是 1.8，如若进一步提高精度要求，调整时间将成倍增加，甚至出现震荡，无法踢出球的状况。因此在不使用时间通报信息时，精度有限制。而使用时间通报信息后，因为调整时间大大减少，精度要求可进一步提高，相关测试正在进行中。

参考文献：

- [1] Patrick F. Riley. SPADES (System for Parallel Agent Discrete Event Simulation) Users Manual[J]. <http://spades-sim.sourceforge.net>,2003.
- [2] Hugo Marques, Nuno Lau and Luís Paulo Reis FC PORTUGAL 3D SIMULATION TEAM: ARCHITECTURE, LOW-LEVEL SKILLS AND TEAM BEHAVIOUR OPTIMIZED FOR THE NEW ROBOCUP 3D SIMULATOR[J]. <http://caspian.iust.ac.ir/Downloads.htm>,2004.