

09 蓝鹰暑期培训

robocup2D server

祝元宠

redsky0802@gmail.com

(本文稿按照 rcssserver13.2.2 编写)

什么是 robocup2D server

- 二维仿真场地
- C/S 结构
- UDP/IP 通讯
- 物理运动模型
- 感知模型
- 球员异构模型
- 自动裁判
- 在线教练

我们可以这么理解.....

- 二维仿真场地 → 虚拟世界，注意球是不能飞起来的.....
- C/S 结构 → 开源 server 保证公平， client= 球队
- UDP/IP 通讯 → 网络通信不保证完全可靠
- 物理运动模型 → 动作指令 + 离散化物理仿真 + 随机误差
- 感知模型 → 信息不完整，伴随随机误差
- 球员异构模型 → 人与人不同
- 自动裁判 → 精确，“越位 1 毫米也是越位”.....
- 在线教练 → 球队的必要补充，“10 号，那边有个空当”.....

离散化仿真

- 球员可以通过视角的改变以及听觉，手势等获取赛场信息，并且要求在每个模拟周期结束前发送规定语法的字符串表示本周期决策的最终命令。而后 server 将这些命令汇总，通过一定规则的运算，得到下一个周期的世界模型，再根据每个球员的视角、听力等发送相应的信息，球员收到信息后，更新自身的世界模型，随即进行下一个周期的决策过程。
- 比赛上下半场各 **5 分钟**，每个模拟周期只有 **100ms**，要求程序的**实时性**非常高。

两种坐标系

- 全局坐标：
 - 平面直角坐标系
 - 向对方球门为 x 轴正方向
 - x 轴顺时针转 90 度为 y 轴正方向
 - 球场为 FIFA 标准场地 105m × 68m
- 个人坐标（相对坐标）：
 - 平面极坐标系，以身体正方向为极轴
 - 合法角度范围： $-180^{\circ} \sim 180^{\circ}$ ， 顺时针为正

server 的几个模型

- 运动仿真模型
- 基础动作模型
- 感知模型
- 球员异构模型
- 裁判和教练

运动仿真模型

- 设 t 表示周期数
- u 表示有效速度
- 速度公式: $u(t+1) = v(t) + a(t) + r(t) + w(t)$
- v 表示上一个周期的速度经过衰减在这个周期中的表现
- a 表示加速度
 - 球员的加速度由加速命令产生
 - 球的加速度由踢球或者铲球命令产生，如果有多个球员同时踢或者铲球，则球的加速度由这些命令产生的加速度进行**矢量叠加**而得到
- r 表示环境噪声， w 表示风力的影响（目前为 0）

运动仿真模型

- 环境噪声 r 的大小和 $\text{ball_rand}(0.1), \text{player_rand}(0.05)$ 有关，方向随机
- 球速大小的最大值是 3.0，球员则为 1.05
- 速度衰减公式： $v(t) = \text{decay} * u(t)$
 - 球员的 decay 参数的标准值是 0.4
 - 球的 decay 值是 0.94
- P 表示位置
- 位置公式： $p(t+1) = p(t) + u(t+1)$

运动仿真模型

- 在 2D 仿真比赛中，球员是一个半径为 0.3 米的实心圆，球是一个半径为 0.085 的实心圆
- 如果球员或球在某个周期有重合，则视作发生一次碰撞
- 碰撞双方的速度各自乘以 -0.1 ，作为下一个周期的速度
- 碰撞仅仅局限于位置重合的情况，如果仅仅是运动轨迹交叉则不发生碰撞

基础动作模型（上）

- dash （ 加速 ）
- turn （ 身体转向 ）
- kick （ 踢球 ）
- tackle （ 铲球 ）
- catch （ 扑球 ）
- move （ 瞬移 ）
- 以上命令不相容，在同一个仿真周期中只可以任选其一发出，不可重复

dash

- 在本周期给自己在某个特定方向上一个加速度
- 参数: power , angle
 - power 为发力的大小, $|power| < 100.0$
 - angle 为加速的方向, 取相对自身坐标系的值
 - power 和 angle 的组合有 8 种, 其中 4 种是有意义的
 - angle 为 0 时, $power > 0$ 表示向前加速, 反之向后
 - angle 为 ± 90 时, $power > 0$ 表示侧向移动

dash 的物理模型

- $\text{dash_power} = \text{effort} * \text{power} * \text{dir_rate} * \text{dash_power_rate}$
- power 是参数
- dash_power_rate 是球员的异构参数，参考值为 0.006
- dir_rate 是 angle 和 power 的函数
 - angle = 0 , power > 0 时, dir_rate 为 1
 - angle = 0 , power < 0 时, dir_rate 为 1
 - |angle| = 90 时, dir_rate 为 0.25

dash 的物理模型

- effort 是和球员状态以及其异构参数 effort_max、effort_min 相关的一个数值。这个数值直接关系到球员在场上的跑动能力，具体将在后面介绍
- 加速度 $accel = (dash_power, angle + angleBody)$
 - angleBody 是身体方向

turn

- 在本周期内让自己转过一定角度
- 参数: angle
 - angle 是转身的角度, $|angle| < 180.0$, 取**相对**坐标
 - 转身时候的实际角度由球员速度和 inertia 这个参数一起决定
 - 球员速度越快, 实际转动的角度越小
 - 球员静止时实际转动的角度最大

turn 的物理模型

- $$\text{new_body_angle} = \text{body_angle} + \text{angle} / (1.0 + \text{inertia_moment} * \text{vel})$$
- angle 是参数
- 可以看出，转身的角速度和异构参数 inertia_moment 与速度大小 vel 的乘积成反比关系
- vel = 0 时，最多可以转 180 度
- vel 的极限是 1.05，若取 inertia_moment = 5，可知在极限速度下，每个周期的转动角度小于 30 度！
- turn 的执行过程中将会产生**随机误差**！

turn 的误差

- Turn 的执行过程中将会产生**随机误差!**
- 实际转过的角度在
$$\text{angle} / (1.0 + \text{inertia_moment} * \text{vel}) * (1.0 \pm \text{player_rand})$$
之间
- $\text{player_rand} = 0.1$ 是常数

kick

- 在本周期给球在某个特定方向上一个加速度
- 参数： power , angle
 - power 为发力的大小， $|power| < 100.0$
 - angle 是发力的方向， angle 取相对坐标
 - 球和球员的边界距离小于 kickable_margin 时可以踢到球，此时的 kick 命令有效
 - 踢球的有效加速度随着球的距离和相对角度的增加而减少

kick 的物理模型

- $\text{eff_power} = \text{power} * \text{kickPowerRate} * (1.0 - 0.25 * \text{dir_diff} / 180.0 - 0.25 * \text{dist_ball} / \text{kickableMargin})$
- power 为参数，kickPowerRate 是常数（0.027）
- dir_diff 是球和身体方向的夹角（绝对值）
- dist_ball 是球和人的边界距离
- kickableMargin 是队员特有的异构参数（见异构模型）
- 球的加速度 $\text{accel} = (\text{eff_power}, \text{angle} + \text{angleBody})$
 - angle 是参数，angleBody 是身体方向
 - accel 在产生过程中有**随机误差**！

tackle

- 在本周期以一定**概率**，在特定范围（与可踢范围部分重叠）内给**球**在某个特定方向上一个加速度
- 参数： angle
 - angle 是发力的方向，取**相对**坐标
 - **power 的值由 angle 唯一确定，不可更改**
 - 球处于球员正前方左右 2.5 米、向前 2 米的区域内时可以得到非零的铲球概率，铲球概率的值由球相对球员的位置唯一确定
 - tackle 完成后，球员有 **10 个周期**不能移动。

tackle 的物理模型

- $(|player2ball.x| / tackle_dist)^{tackleExponent} + (|player2ball.y| / tackle_width)^{tackleExponent} = fail_prob$
- 铲球失败的概率是两个指数式的和
- 指数 tackleExponent 是 server 定义的常数，等于 6
- player2ball 是指球员到球的相对向量
- tackle_dist 和 tackle_width 也是常数，分别是前述的 2.0 和 1.25 (= 2.5 / 2 , 分成左右两边)
- 要求 player2ball.x > 0

tackle 的物理模型

- $\text{tackle_eff_power} =$
 maxTacklePower
 $\times (1.0 - (| \text{angle} | / 180))$
 $\times \text{tacklePowerRate}$
- maxTacklePower 现在被定义为 100
- 其它的参数理解和 kick 指令相同
- tacklePowerRate 也等于 0.027 , 所以**单人单周期踢球或者铲球**, 在极限情况可以得到 2.7 的加速度
- 同理: accel
 $= (\text{tackle_eff_power}, \text{angle} + \text{angleBody})$
- 与 kick 一样, 这里的 accel 也有**随机误差**!

kick 和 tackle 的加速度误差

- Kick 和 tackle 的加速度误差算法相同
- 误差为极坐标形式的矢量
 - 误差的方向随机
 - 误差的大小和 kick_rand 这个异构参数有关
 - 误差的大小也和球相对球员的位置和速度大小有关

catch

- 在本周期内扑到球
- 参数: angle
 - angle 是扑球的方向, 可以是任意方向
 - angle 取**相对**坐标
 - 球处于相对球员 angle 方向的横向 2.0 米、纵向 1.2 米的区域内时可以扑到球, **概率为 1**
 - 守门员相邻两次扑球的最短间隔为 **5 个周期**, 以模拟真实世界中的扑救动作

move

- 在己方半场的自由移动
 - 上下半场开场前
 - 任意一方进球后
- 守门员扑到球后可以在己方禁区内使用两次 move 命令以避免对方球员的紧逼围堵
- 参数： x , y
 - x , y 为全局坐标

能量何处来？

- 不知道大家是否注意到了，在前面六条不相容的基本指令中，不少指令里面有一个 **power** 值
- 其中人的跑动（dash），将要消耗体力！
 - move 指令的意义：减少体力的无谓消耗
- Robocup2D 最新体力模型（电池模型）如下：
 - 每个球员在每个半场开始前将补充满他们的“体力池”，总计 148600 点
 - 每个球员即时体力还有 8000 点（上限值）

能量何处来？

- 每个 dash 执行后，消耗的体力为 power 的值，如果 $\text{power} < 0$ ，则消耗 $-2.0 * \text{power}$
- 每个周期，球员通过自身属性中的 staminc 从体力池中补充相应数值的即时体力，补充到上限处为止
- 一些动作体力消耗巨大，如长时间带球、长距离的全力冲刺、倒退跑（ $\text{power} < 0$ ）等
- 体力下降到 2400 点，将导致 effort 的下降，从异构参数的 effort_max 降到 effort_min
- 体力回升到 4800 点，effort 将会随之回升

基础动作模型（下）

- turn_neck （ 头部转向 ）
- change_view （ 改变视角 ）
- pointto （ 手臂指向 ）
- attentionto （ 听觉集中，只注意某个人的喊话 ）
- say （ 喊话 ）
- score （ 数据请求 ）
- 以上命令可以相容，在同一个仿真周期中可以发出一次，但是总数不能太多以堵塞 server 的通信

turn_neck & change_view

- turn_neck 的参数: angle
- change_view 的参数: width
- 其中 $|\text{angle}| \leq 90.0$, 取相对坐标
- width 表示视野宽度, 可以是 narrow、normal、wide 三种。
 - 在同步视觉模型中, 上述视角和刷新周期分别为 60、120、180 度和 1、2、3 周期, 并且视觉信息总是在更新周期开始的时刻发送到球员
 - 还有一种异步视觉模型, 但是不推荐使用, 这里就不介绍了

say & attentionto

- say 的参数: msg
- msg 是一个用双引号引出的字符串, 如“abc”
- say 成功后, 会有返回消息 (ok say)
- attentionto 的参数: team , num
- team 表示哪一支球队, num 为球员号码
- attentionto 可以单独使用一个参数 off , 用以停止动作

pointto & score

- pointto 的参数: dist , angle
- angle 取相对坐标, dist 表示距离
- Score 没有参数
- score 会返回 (score Time Our Their)
 - Time 为时间, Our : Their 为比分

如何获取信息？

- 以上讲了不少动作指令，可是，做出决策之前，**信息的获取和分析，某种程度上说比决策方法本身更加重要！**
- 球员的信息获取将会通过**视觉**、**听觉**和**感知**进行
 - 感知？大家可以认为是对自身信息的获取。主要用于提取自身的状态。

视觉模型

- 相对坐标系
- 地标定位
- 信息不完整性
 - 距离加大，如果对象是球员，则首先丢失球员号码信息，然后是所属球队信息，最后是完全看不见。
- 信息噪声
 - 11 个人可以得到 11 个不同的球的位置，与现实十分接近。
 - 极坐标系的误差：距离和方向
- 刷新周期和视野范围的矛盾.....不可调和 ~~~

听觉模型

- 拥挤的单通道低带宽环境
- 信息公开、通信不可靠、噪音大
- 听力总能力有限，听一次少一次.....
- 那这个东西有什么用？
 - 裁判的“哨音”将会通知比赛状态的改变，这种信息绝对可靠.....
 - 尽管每个周期只能听到一条长度极其有限的信息，但是在信息高度缺乏的球场上，可以认为极其珍贵.....
 - 通信加密 -> 特殊战术 -> 致命一击

感知模型

- 球员能够感知到的信息：
 - 视角
 - 体力
 - 身体速度（大小和方向）
 - 头部相对身体前方的角度
 - 发出各种命令的次数，可以检查 server 是否 miss 了一些命令（在系统负荷很高的时候会发生这种情况）

球员异构模型

- 18 人大名单，编号 id 0 ~ id 17
- 规定 id 0 是守门员
- 异构参数：
 - (player_type (id 0)(player_speed_max 1.05)(stamina_inc_max 45)(player_decay 0.4)(inertia_moment 5)(dash_power_rate 0.006)(player_size 0.3)(kickable_margin 0.7)(kick_rand 0.1)(extra_stamina 50)(effort_max 1)(effort_min 0.6))
 - player_speed_max == 1.05 常数
 - player_size == 0.3 常数

球员异构模型

- 用途：
 - 最大速度： dash_power_rate 、 player_decay 、 effort_max
 - 爆发力： dash_power_rate 、 effort_max
 - 耐力： stamina_inc_max
 - 转身速度： inertia_moment
 - 控制范围： kickable_margin
 - 踢球的准确性： kick_rand
- 教练选择 id1~id17 中的 10 个上场比赛，中途不得换人

教练还能干啥？

- 教练在指定上场队员后，可以断开和 server 的连接，也可以继续在场地上指挥（但是有 **50 周期的时间延迟**，基本上没有作用）
- 获取**无噪声的全局信息**
- 用 say 指令和球员交互
- 消息的总量、大小、类型都有限制
- 死球时，通讯限制较小，可以传输大量信息，如对对方阵形和球员个体的分析等
- 离线教练：用于训练和调试，场景重现等

作业

- 下载 rcssserver 的源代码（蓝鹰 2D 主页）
- 阅读 src/player.cpp 中 4 个物理模型的具体代码，加深印象：
 - dash、kick、tackle、turn
- 由于 server 要向下兼容，所以 server 中的代码，除了带有很强的 C++ 风格外，还和这个文档的表述方式略有不同
- 读 server 代码是蓝鹰 2D 每年的必修课

今天就讲到这里，谢谢！