

RoboCup Simulation 2D Server

江森

mjiang@mail.ustc.edu.cn

(基于 rcssserver15.0.1 编写)

一场仿真 2D 的比赛过程

- 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。
- 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。
- 然后，Server 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。
- 这些工作都要求在 100ms 之内完成。比赛上下半场默认各位 3000 周期。

Server-The World of Player

- Stadium
- Referee

Stadium-The Stage of Player

- 2D
- 离散化
- 运动模型
- 感知模型
- 动作模型
- 球员异构模型

2D 环境及其描述

- 整个球场只是一个平面。球是不能飞起来的。
- 2D 中采用的坐标系有 2 种：
- 平面直角坐标系——用于全局信息
 - 对方球门为 x 轴正方向
 - X 轴顺时针旋转 90 度为 y 轴正方向
- 平面极坐标系——用于个人信息
 - 身体的朝向为极轴
 - 顺时针为正，逆时针为负
 - 角度大小 (-180 , 180)

离散化

- 时间的离散化
 - 每 100ms 为一个周期，工作按照周期循环
- 空间的离散化
 - Server 发过来的信息，精度都只有 0.01
- 动作的离散化
 - 比如 Dash，仅能够向 8 个方向 Dash
 - 而且 Dash 和 Turn 不相容，不可能在一周期内又 Dash，又 Turn

运动模型

- 速度公式

- $u(t+1) = v(t) + a(t) + r(t) + w(t)$

- u 为“有效速度”

- v 为衰减之后的上周期速度

- $v(t) = v(t-1) * \text{decay}$

- 其中: $\text{ball_decay}(0.94), \text{player_rand}$ (异构参数)

- a 为本周期其得到加速度，如果没有，则其为 0

- 球员的加速度由 dash 产生

- 球的加速度由 kick 或 tackle 产生。多人踢球进行矢量叠加

- r 为随机矢量

- 大小: 和 $\text{ball_rand}(0.1), \text{player_rand}(0.05)$ 有关。

- 方向: 随机

- $w(t)$ 为风速，目前为 0

运动模型

- 位置

- $P(t+1)=P(t)+u(t+1)$

- 碰撞

- 球是一个半径为 0.085m 的实心圆
 - 球员是一个半径为 0.3m 的实心圆
 - 如果某周期球员和球或者和另一个球员有部分重合，视为发生一次碰撞
 - 碰撞后， $u(t+1)=v(t)*(-0.1)$
 - 注：仅当重合才发生碰撞。若仅仅是轨迹交叉则不发生！

视觉

- 相对坐标系
- 通过地标定位自身
- 信息的不完整性
 - 首先丢失号码信息。然后是所属球队信息。最后完全看不见。
 - 注意，不论多么近，我是看不见对方做了什么动作的。
- 信息噪声
 - 得到的各种位置都有误差，距离越远误差越大。
- 刷新周期和视野范围
 - 1 周期 (60°)、2 周期 (120°)、3 周期 (180°)。视觉信息会在更新周期结束后发送给球员。

听觉

- 单通道—我不知道他听没听见
- 低带宽—只有 10 字节
- 信息公开—敌人也能听见—如果他听
- 信息不可靠—我说了他不一定能听见
- 听力有限—我一次只能注意两个人，一个对手的，一个队友的。

听觉

但是：

- 裁判通知比赛状态的改变用的方式就是“哨音”。
- 对配合的影响，是视觉的重要辅助：
 - “11号，我要给你传球啦！”
 - “执行B计划！”

所以重要性不言而喻

感知

- 感知自身

- 视角范围
- 体力
- 身体速度
- 头部位置
- 发出各种命令的次数

- 感知周围

- 能够知道周围特定距离内的物体的位置，不过不知道速度。这个感知信息是和视觉信息一起发出的。

信息的获取

- Turn neck
- Change view
- Point to
- Attention to
- Say
- Score
- 以上动作在一个周期之内可以各发出一次（只要在最大信息长度的限制之内）

Turn neck

- 参数： Angle
- 作用： 转动头部
- $|\text{Angle}| < 90$ ， 取相对坐标
- 这是唯一一个不会产生误差的动作， 转多少就是多少！

Change view

- 参数， width
- 作用：改变视角大小
- Width 包括 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
- 视觉信息总是在满足更新条件后才发送到球员
 - 譬如之前是 wide ， 则本周期变成了 narrow/normal ， 也需要再过一个周期才能接到视觉。

Point to

- 参数： dist ， angle
- 作用： 指向某个点。 不管自己转身还是别的， 仅仅指向那一个点。
- Angle 取相对坐标， dist 是到自己的距离。
- 可以发出 off 指令让自己不指。
- 不管是队友还是对手都只能够看到手臂方向。

Attention to

- 参数: team num
- Team 是哪一只球队
- Num 是哪个队员
- 还可以单独使用 off , 然后终止关注
- 关注的效果就是, 将会听到该队员说的一句话, 如果该队员没有说话, 将会听到那个队伍里随机的一句话
-

Say

- 参数： msg
- 实际上就是喊话。
- 如果有人注意你（ Attention to ）他下一周期就能够听到。
- 如果没人注意，也有一定概率被听到。

Score

- 无参数
- score 会返回 (score Time Our Their)
- Time 为时间， Our : Their 为比分

动作模型

- Dash
- Turn
- Kick
- Tackle
- Catch
- Move

Dash

- 参数： power ， angle
- 在本周期给自己在某个特定方向上一个加速度
- power 为发力的大小， $|power| < 100.0$
- Power 的值可正可负。当为负值的时候，即为倒跑。
- angle 为加速的方向，取相对自身坐标系的值。
- 注意，这个 angle 仅有 8 个方向 0° ， $\pm 90^\circ$ ， $\pm 45^\circ$ ， $\pm 135^\circ$ ， 180°
- 显然， 90° 正跑和 -90° 倒跑效果是一样的。其余的类似。

Dash

- $\text{dash_power} = \text{effort} * \text{power} * \text{dir_rate} * \text{dash_power_rate}$
- power 是参数
- dash_power_rate 是异构参数
- dir_rate 是 angle 的函数
 - angle = 0 , dir_rate 为 1
 - |angle| = 90 时 , dir_rate 为 0.4
 - |angle| = 45 或者 135 时 , dir_rate 为 0.7 和
 - |angle| = 180 时 , dir_rate 为 0.6

Dash

- effort 是和球员状态以及其异构参数相关的一个数
- 加速度 $\text{accel} = (\text{dash_power}, \text{angle} + \text{angleBody})$
 - angleBody 是身体方向

Stamina – The Power of Dash

- Robocup2D 最新体力模型（电池模型）如下：
 - 每个球员在每个半场和加时赛开始前将补充满他们的“体力池”，总计 130600 点
 - 每个球员即时体力还有 8000 点（上限值）
 - 每个 dash 执行后，消耗的体力为 power 的值，如果 $power < 0$ ，则消耗 $-2.0 * power$ ，这就是倒跑的代价
 - 每个周期，球员通过自身属性中的 stamina_inc 从体力池中补充相应数值的即时体力，补充到上限 8000 为止

Stamina – The Power of Dash

- 一些动作体力消耗巨大，如长时间高速带球、长距离的全力冲刺、倒退跑（ $power < 0$ ）等
- 体力下降到 2400 点，将导致 effort 的下降，从异构参数的 effort_max 逐渐降到 effort_min，同时 stamina_inc 也会逐渐下降体力回升到 4800 点，effort 将会随之逐渐回升，stamina_inc
- 则不会随之回升，只是在半场开球时恢复为默认的最大值
- 当体力池内的体力耗尽后，每次 Dash 仅能使用 extra_stamina 的体力。extra_stamina 是异构参数，参考值为 50。

Turn

- 在本周期让自己转过一定角度
- 参数： angle
- angle 是转身的角度， $|\text{angle}| < 180.0$ ， 取相对坐标
- 转身时候的实际角度由球员速度和 inertia_moment 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

- $\text{new_body_angle} = \text{body_angle} + \text{angle} / (\text{inertia_moment} * \text{vel})$ (1.0 +
- 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，相对误差为 10%

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

- $\text{eff_power} = \text{power} * \text{kickPowerRate} * (1.0 - 0.25 * \text{dir_diff} / 180.0 - 0.25 * \text{dist_ball} / \text{kickableMargin})$
- 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 是异构参数
- 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，给球在某个特定方向上一个加速度
- 参数：angle，foul
- 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（倒地）

Tackle

- 参数 foul 决定是否犯规，取字符串 true/false
 - foul = true 时，铲球后没有 10 个周期的屏蔽时间，相反，在危险情况下会导致对方的“倒地”（如果铲球成功）。若被“发现”（取决于概率，目前为 0.5），则会被判黄牌。
 - 两张黄牌下场。
- 危险情况定义：
 - 持球者（可以踢球的人）位于铲球者和球之间（二者连线会穿过持球者）。
 - 持球者背对铲球者（其身体角度跟自己与球连线方向夹角小于 90 度）。
 - 持球者在铲球者 Tackle “前”发出了 Dash 命令。

Kick 和 Tackle 的随机误差

- 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，取相对坐标，可以是任意方向
- 球处于相对球员 angle 方向的横向 2.0 米、纵向 0~UnrelCatch 最小值的区域内时可以扑到球，概率为 1
- 球处于相对球员 angle 方向的横向 2.0 米、纵向 UnrelCatch 区间的区域内时可以扑到球，概率为由内向外，从 1 开始，沿径向线性衰减到 0
- UnrelCatch 是球员的异构信息，其区间中心是 1.3 米。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
- 参数： x , y
- x , y 为全局坐标
- 上下半场开场前
 - 用以避免无谓的体力消耗
- 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

球员异构模型

- 总共有 18 人，编号 id 0 ~ id 17
- 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)(kick_power_rate 0.027)
(foul_detect_probability 0.5)(catchable_area_l_stretch
1))
- 以上为各个异构的参考值，也就是说，这些异构参数的设置不会与这些值相差太大
- player_speed_max == 1.05 常数
- player_size == 0.3 常数

教练

- 指定上场的球员的类型
 - 教练选择 id1~id17 中的 10 个上场比赛
 - 中途还可以在死球状态（非 play_on）下换人，一场比赛最多 3 个
- 获取无噪声的全局信息。
 - 包括一些球员无法得到的信息，如手臂指的距离。
 - 即使是教练也不知道球场上球员做了什么动作。
- 用 say 指令和全局交互
 - PlayOn 下有 50 周期的延迟
 - SetPlay 下没有延迟
- 消息的总量、大小、类型（info, advice 等等）都有限制
- 离线教练：用于训练和调试，场景重现等。

Referee-The Rule of Game

- Play on—— 正常比赛
- Kick off—— 开球——为节省体力，可以 Move
- Kick in—— 发界外球——只能用踢
- Corner kick—— 角球
- Goal kick—— 球门球
- Goalie free kick—— 守门员开球
- Free kick—— 任意球
- Indirect free kick—— 间接任意球

Foul

- Tackle（铲球）时，进入危险情况，则有 0.5 的概率被判定为犯规
- 如果是危险情况下的故意铲球（参数 foul 为 true），则会判定为犯规，有 0.5 的概率吃到黄牌
- 满两张黄牌则转为红牌，直接被清理下场
- 而铲球也有一定几率直接被判红牌，不过当前该值被设置为 0

Offside

- 传球时，接球队员的 x 坐标大于对方倒数第二个防守球员和
- 球的最大 x 坐标，则接球队员越位
- 越位线最少不超过中线，最大不超过底线
- 接球队员的判定，是接触球（和球相撞）或者对球发出 Kick 或者 tackle 指令即算作接球。

Back pass

- 守门员在禁区内用 catch 指令接住了之前最后一次触球人为队友的球，即判定为回传球违例
- 如果最后一次踢球时，本方球员和对方球员同时踢到球，则不算回传球违例
- 30 周期后，自动转换成间接任意球

Kick fault

- 一般特殊模式只要踢球即完成开球
- 门球要求球被传出禁区才完成开球
- 球在禁区内被对方踢到则重发
- 若球被本方队员带球（踢 - 跑 - 踢），则违例，应坠球
- 若是在进行 freekick 或者 kickin 的时候违例，则球权交给对方，由对方发球

发球超时

- 发球超时及处理：
- 一方发球，在 100 周期后没有进入 play_on，即球没有发出，则判发球违例（ free_kick_fault ）
- 此时在发球位置坠球（ drop_ball ），重新开始比赛
- 如果坠球点在禁区内，则改为在禁区的左右上角坠球

Forbidden area

- 真人足球的人墙距离是 9.15 米
- 因此，一般在特殊模式时， server 会把对手的人直接“扔出”以球为圆心， 9.15 米为半径的圆外，并将阻止对手进入
- 门球模式下，这个区域是开球方的整个禁区
- 越位时， server 会将越位方所有球员“扔回”越位线以后
- 开球时， server 会将滞留在对方半场的球员“扔回”本方半场。

平局的处理

- 比赛有两场加时赛，每场 1000 周期。
- 采用传统赛制，当两场加时赛全部结束时比赛才结束。
 -
- 若加时赛结束，双方比分仍然相同，则采用点球。

Thanks!