

第二章 RoboCup 仿真2D Server平台



本章概要



- **Server**平台概要
- **Server**之球场仿真
- **Server**之消息板
- **Server**之裁判模块
- **Simple Client**
- **教练-Trainer和Coach**

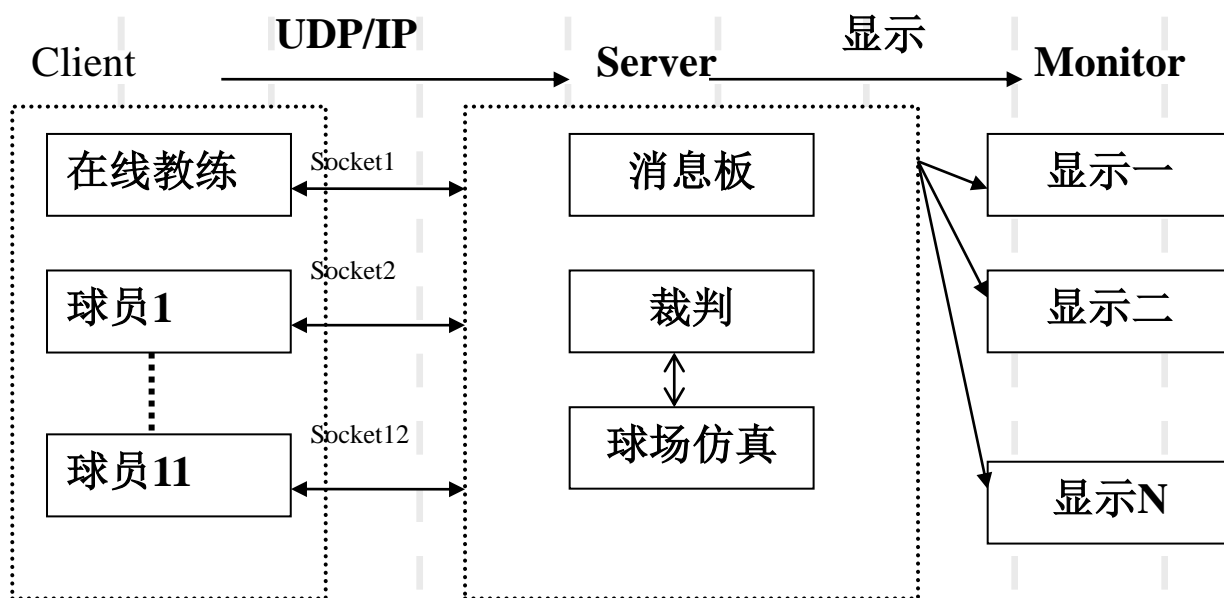


2.1 Server平台 (1)



仿真系统由3个部分组成的：

- Soccer Server
- Monitor
- Client



2.1 Server平台（2）



- 比赛平台包括两个主要程序：**soccerserver**和**soccermonitor**。**Soccerserver**作为一个服务器程序模拟所有球员和球的移动、和球员通讯以及根据比赛规则控制比赛进程。**Soccermonitor**是一个程序将从**soccerserver**那里获得场上信息显示到一个虚拟场地上，可以有許多**soccermonitor**连接到**soccerserver**上。
- 比赛平台提供了一个虚拟的足球场地，由客户端程序控制的队员可以在场上跑动，踢球。**Soccerserver**主要由球场仿真模块、裁判模块和消息板模块三个部分组成。



2.1 Server平台 (3)



- 球场仿真模块

- 计算球场上对象的运动，检测他们之间的碰撞。球场上的对象包括每队各11名队员、球、球门、标记及标志线等。其中球和球员都具有大小、位置、速度、加速度等属性，球员则还有方向、耐力等属性。球员与球的属性每个周期末计算更新一次，计算的依据是动力学定律。如果球员与球员或球之间发生重叠，则作碰撞处理。

- 裁判模块根据规则来控制比赛的进程

- 自动裁判可以检测一些简单的形势，如进球、界外球、越位等

- 消息板模块负责客户端之间的通讯

- 通过socket，客户端程序可以发送命令来控制球员，
- 通过socket，客户端程序可以接收球员的感知信息。



2.1 Server平台（4）

- Soccerserver采用的离散化模式运行。每个仿真周期是100ms
- 在每个周期的开始server根据各个球员的状态（包括可视范围、获得时间等）发送相应的已更新的场上信息给Client
- 每个Client在每个仿真周期结束前，发送自己的行为到Server
 - 如果一个球员在一个周期内发送了多于一条的独立行为请求，server将随机选择一个执行（很多时候是使用后面一条命令）
 - 另一方面，如果球员在一个周期内没有发送行为请求，它将失去该周期的行动机会，对于这样一个实时对抗的环境这无疑是很不利的
- Server收集所有球员程序的行为请求，直到每个周期末才统一执行并更新场上信息

2.1 Server平台(5) - server获取

- 一个完整的机器人足球仿真源码包包括以下几个部分，当然每个部分也可以独立获得并安装、升级。
 - **rcssbase** 供其他各种仿真机器人足球程序包使用的基本代码。
 - **rcssserver** 是该仿真源码包的主要部分，执行实际的仿真工作。客户端程序和server通过UDP/IP协议发送命令和接收感知信息。
 - **rcsslogplayer** 可以重放rcssserver录制的比赛录像（*.rcg文件）。要注意的是rcsslogplayer只是用于控制录像的回放，显示还是需要通过monitor。
 - **rcssmonitor**和**rcssmonitorclassic** 通过连接rcssserver或rcsslogplayer来显示现场比赛或比赛录像。

2.1 Server平台(6) - server获取

- 获取Soccer server

- www.robocup.org

- <https://github.com/rcsoccersim/>

- robot.hfut.edu.cn

2.1 Server平台(7) – server安装

- 以rcsoccersim-*.tar.gz源码包为例，说明如何安装比赛平台。
- -> tar xf rcsoccersim-*.tar.gz
- 同时一个目录rcsoccersim-*会自动会建立。进入该目录：
- -> cd rcsoccersim-*
- -> ./configure
- -> make

rcsoccersim-*.tar.gz -> rcsoccersim-15.0.1.tar.gz

2.1 Server平台(8) – server启动

- 分别启动server、monitor和client.

- ./rcssserver

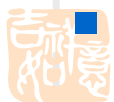
- ./rcsslogplayer (./rcssmonitor)

- ./soccerserversim启动server和monitor.

本章概要



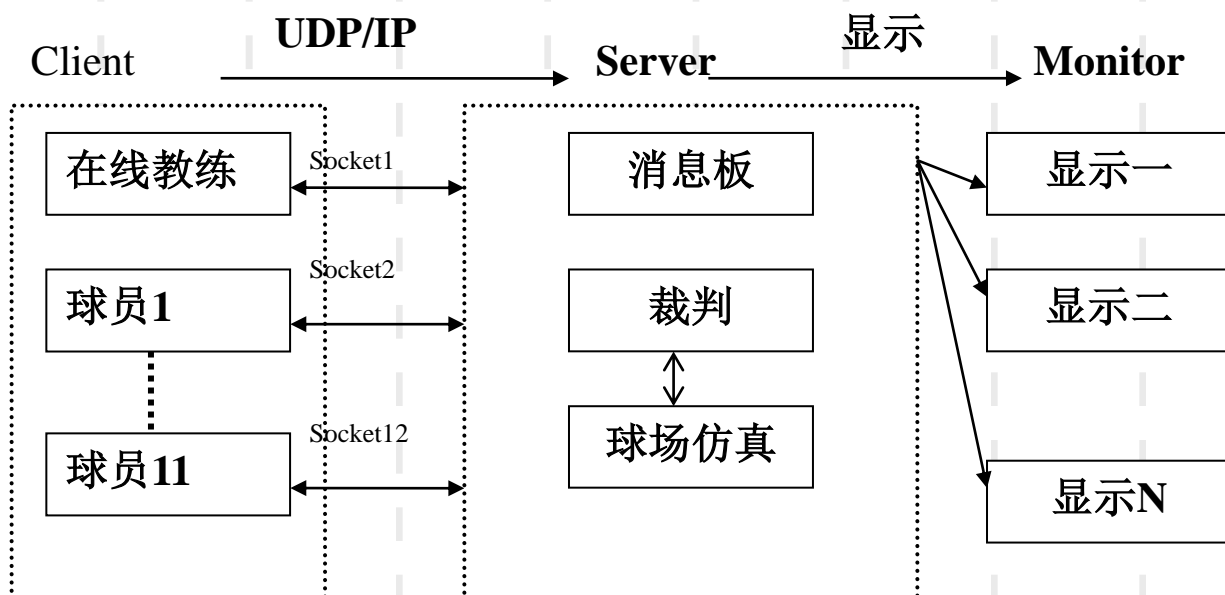
- Server平台概要
- Server之球场仿真
- Server之消息板
- Server之裁判模块
- Simple Client
- 教练-Trainer和Coach



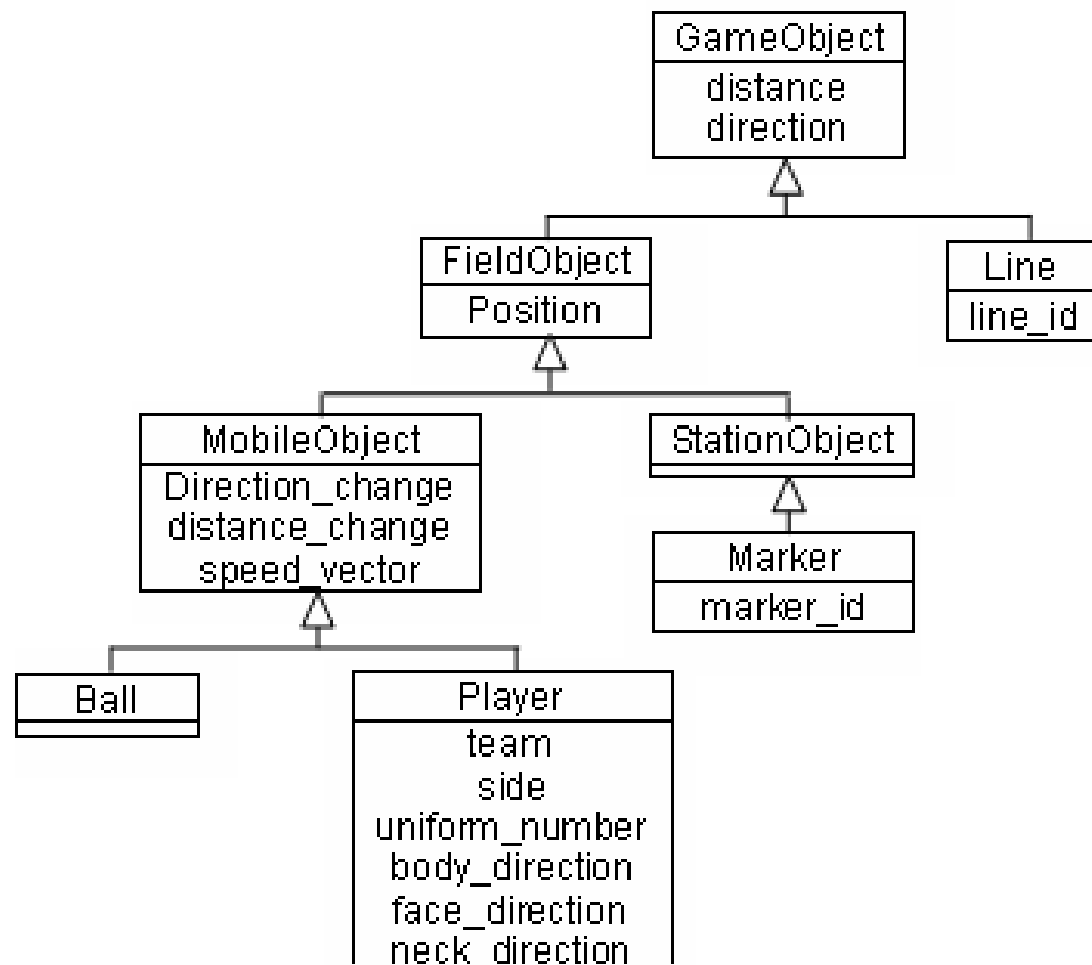
2.2 Server之球场仿真



- 球场上的对象
- 场地和球员
- 对象的运动模型
- 球员的体力模型
- 异构球员



2.2.1 球场上的对象



2.2.2场地和球员



- 仿真环境中足球场和其中的全部对象都是二维的
 - 105 (m) * 68 (m)
 - 14.64 (m)，是实际的两倍
 - 球员和球都使用圆圈来表示
- 仿真比赛使用2个坐标系(球场全局坐标系和球员个人坐标系)
 - 球场上全局坐标系为平面直角坐标系，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。
 - 球员个人坐标采用平面极坐标系，以face正方向为极轴，合法角度范围：-180~ 180，顺时针为正。



2.2.3对象的运动模型

$$(u_x^{t+1}, u_y^{t+1}) = (v_x^t, v_y^t) + (a_x^t, a_y^t): \text{加速}$$

$$(p_x^{t+1}, p_y^{t+1}) = (p_x^t, p_y^t) + (u_x^{t+1}, u_y^{t+1}): \text{移动}$$

$$(v_x^{t+1}, v_y^{t+1}) = \text{decay} \times (u_x^{t+1}, u_y^{t+1}): \text{衰减速率}$$

$$(a_x^{t+1}, a_y^{t+1}) = (0, 0): \text{复位加速度}$$

$$(a_x^t, a_y^t) = \text{Power} \times (\cos(\theta^t), \sin(\theta^t))$$

噪音

$$(u_x^{t+1}, u_y^{t+1}) = (v_x^t, v_y^t) + (a_x^t, a_y^t) + (\tilde{r}_{r_{\max}}, \tilde{r}_{r_{\max}})$$

2.2.4球员的体力模型

- 每个球员都有自己的体力值，存储在虚拟的“体力池”里。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。球员还有一个即时体力（最大值为8000），在执行dash、kick、tackle等命令时，消耗即时体力。同时，每个周期球员通过计算出当前可以即时恢复的体力，从体力池中补充相应数值的即时体力，补充到即使体力的最大值为止。



dash时体力的消耗、体力使用效用及补充过程

- Soccerserver通过限制球员的体力来阻止队员始终以最大速度（*player_sp_max*）跑动。其中涉及到三个方面：
 - *stamina* ([0, *stamina_max*]) 表示体力值，它限制了***dash***命令的*power*参数。
 - *effort* ([*effort_min*, 1.0]) 表示球员体力使用效用。
 - *recovery* ([*recovery_min*, 1.0]) 表示即时可补充的体力。

2.2.4球员的体力模型（1）

■（1）可消耗的体力

- 在dash时，给出的 $power$ 参数是消耗的体力，此执行完dash命令后，即时体力值都要减少 $power$ ，因此很显然该值必须要小于当前球员的体力值 $stamina$ 。需要注意的是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 $-2.0 * power$ 。

- 即实际消耗的体力为： $power = \min(power, stamina)$

- 消耗了以后的新体力为： $stamina = stamina - power$

2.2.4球员的体力模型（2）

（2）体力使用效用

- 球员在不同体力条件下使用消耗相同的体力值的实际效果是不一样的，如：在体力充沛时，消耗值全部被转化成了实际值，体力比较差的时候只有部分转化成了实际值，这个转化的效率被称为体力使用效用，即**effort**。**effort**的取值和当前的实时体力有关，随着实时体力的变化而变化。在每个循环周期内，如果**stamina**低于某个阈值，则**effort**减少，否则增加：

- if **stamina** **effort_dec_thr****stamina_max** and **effort** > **effort_min** then
➤ **effort** = **effort** - **effort_dec**

- if **stamina** **effort_inc_thr****stamina_max** and **effort** < **effort_max** then
➤ **effort** = **effort** + **effort_inc**

- Server13版本以后，**stamina_max**为8000，**effort_dec_thr**取值为0.3，**effort_dec**取值为0.005，**effort_inc_thr**取值为0.6，**effort_inc**取值为0.01，**effort_min**的取值为0.6，**effort_max**取值为1.0。

- 当即时体力**stamina**下降到2400点以后，每个周期**effor**下降0.005。

- 当即时体力**stamina**回升到4800点以后，每个周期**effort**回升0.01。

2.2.4球员的体力模型 (3)

■ (3)体力的补充

➤ 每个周期，球员都从体力池中补充一定数值的体力，补充的值是由`recovery`参数控制的。同体力使用效用一样，体力越差补充的越慢。具体为：

➤ `if stamina < stamina_max and recovery > recover_min then`

➤ `recovery = recovery - recover_dec`

➤ 其中`stamina_max`为8000，`recover_dec_thr`取值为0.3，`recover_min`取值为0.5，`recovery`取值为1.0，`recover_dec`取值为0.002。

➤ 球员实时体力值`stamina`为：

➤ `stamina = stamina + recovery*stamina_inc`

➤ `if stamina > stamina_max, then stamina = stamina_max`

➤ `stamina_inc`取值为50。

2.2.4球员的体力模型（4）

- 需要指出的是在“体力池”里面的体力储备耗尽时，球员的体力也就没有补充源泉了，所以在开发程序的时候要谨慎使用一些体力消耗巨大的动作。如长时间带球、长距离的全力冲刺、倒退跑（ $\text{power} < 0$ ）等。



2.2.5异构球员



- 所谓异构球员，即个人特点及能力不一样的球员。
- 球员的属性
 - player_type //球员类型
 - player_speed_max //最大速度
 - stamina_inc_max //每周期最大即时体力恢复值
 - player_decay //球员速度衰减
 - inertia_moment //惯性因子
 - dash_power_rate //加速力量转化率
 - player_size //球员自身直径大小
 - kickable_margin //可踢球范围
 - kick_rand //在踢球方向上的随机噪音
 - extra_stamina //异构球员的额外体力
 - effort_max //最大体力使用效用
 - effort_min //最小体力使用效用
 - kick_power_rate //踢球力量转化率
 - foul_detect_probability //异构球员犯规被判罚的几率
 - catchable_area_l_stretch //异构球员扑球范围的扩展部分



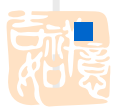
2.2.5异构球员

- 0号类型球员（也是标准球员）的具体参数如下：
(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_margin0.7)(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))
- 其中，球员的player_speed_max和player_size属性为常数，目前的值分别为1.05和0.3。player_type的值为18，也就是可以出现18种不同类型的球员。比赛时每支球队可以使用其中的10种类型，在比赛时可以实现3次换人。

2.2.5 异构球员



- 调节dash_power_rate、player_decay和effort_max变大可以实现球员的更大的速度
- 调节dash_power_rate、effort_max变大实现球员更大的爆发力
- 调节stamina_inc_max变大实现球员更大的耐力
- 调节inertia_moment变小实现更快的转身
- 调节kickable_margin变大实现球员的更大踢球范围
- 调节kick_rand变小实现球员踢球更准确。



本章概要



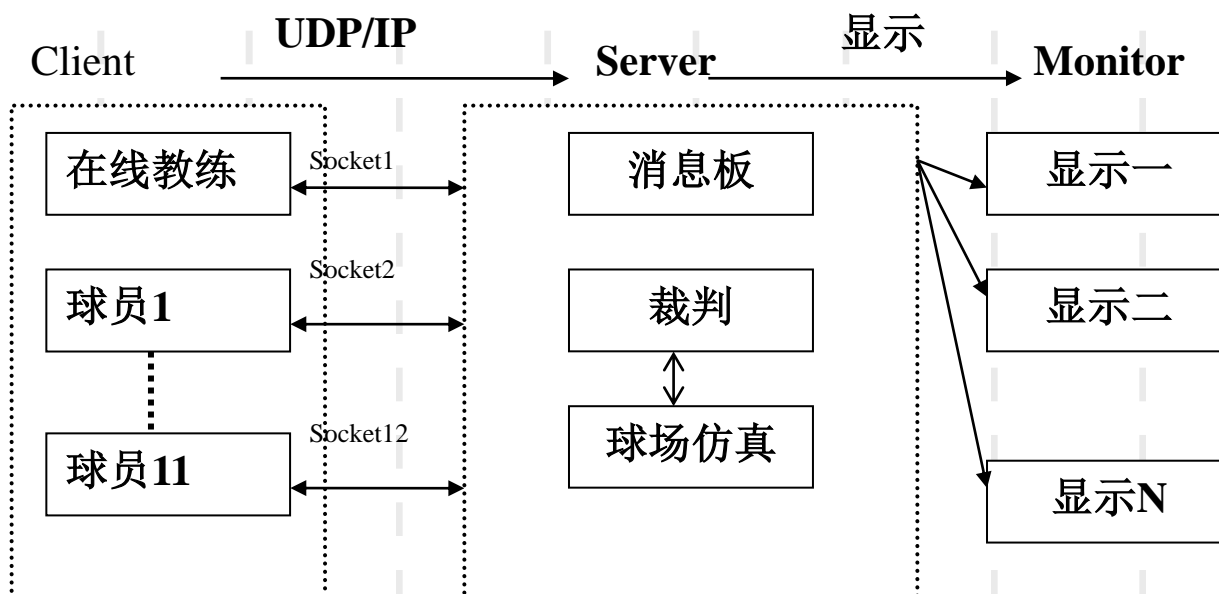
- Server平台概要
- Server之球场仿真
- **Server之消息板**
- Server之裁判模块
- Simple Client
- 教练-Trainer和Coach



2.3 Server之消息版



- 球员的感知信息模型
- 球员的动作模型



2.3.1 球员的感知信息模型

- 一个RoboCup智能体有3种不同的感觉。
- **听觉感知模型**感觉来自于裁判, 教练, 和其他的球员发送的(语音)消息
- **视觉感知模型**感觉场地的可视化信息, 诸如球员当前视野内的距离, 物体的方向. 视觉传感的工作也是象亲近传感器, 看到近的物体, 而看不到背对球员的世界.
- **自身感知模型**感觉当前球员本身的状态, 象他的体力, 速度, 颈部的转角. 这些传感器(感觉)一起给球员构成一个完整的世界模型.

(1) 听觉感知模型



- (hear Time Sender "Message")

- Time: 前的仿真周期。
- Sender: 如果是其他球员发送的消息，那么是发送者的相当方向 (Direction)，否则就是下面的选项：
- self: 发送者是自己本人。
- referee: 裁判是发送者。
- online_coach_left 或者online_coach_right: 发送者是在线教练。
- Message: 消息内容。

■ (hear 2021 self "I'm numb5") 表示在第2021周期听到自己说的话 "I'm numb5";

■ (hear 2023 -30 "I'm numb5") 表示在第2023周期听到来自相对自己身体-30度方向说的话 "I'm numb5";

■ (hear 2025 referee corner_kick_r) 表示在第2025周期听到裁判说的话当前比赛模式为corner_kick_r，即右边球队要发角球。



(2) 视觉感知模型



- (see Time ObjInfo ObjInfo ...)

➤ Time: 当前时间。

➤ ObjInfo表示了可视对象的信息。其格式为:

(ObjName Distance Direction DistChng DirChng BodyDir HeadDir)

- ObjName = (player Teamname Unum)

- | (goal Side)
- | (ball)
- | (flag c)
- | (flag [l | c | r][t | b])
- | (flag p [l | r][t | c | b])
- | (flag [t | b][l | r][10 | 20 | 30 | 40 | 50])
- | (flag [l | r][t | b][10 | 20 | 30])
- | (flag [l | r | t | b] 0)
- | (line [l | r | t | b])

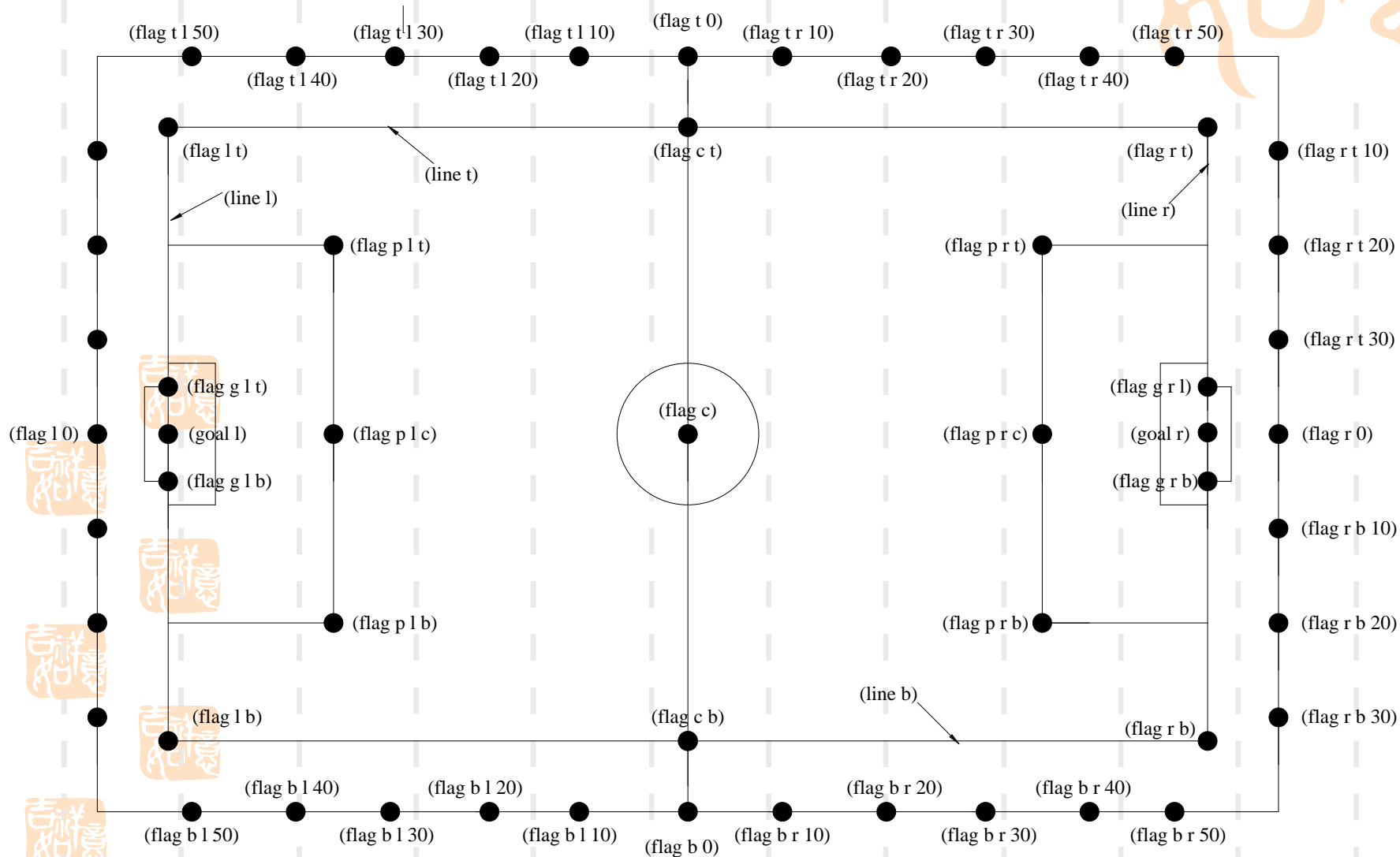
- Distance, Direction表示目标的相对距离和相对方向

- DistChng和DirChng分别表示目标距离和方向的相对变化

- BodyDir 和 HeadDir, 分别是被观察球员相对观察者的身体和头部的相对角度
(*)



场上标志




各对象信息计算公式

$$P_{rx} = P_{xt} - P_{x0}$$


$$P_{ry} = P_{yt} - P_{y0}$$

$$v_{rx} = v_{xt} - P_{x0}$$


$$v_{ry} = v_{yt} - P_{y0}$$


$$Distance = \sqrt{P_{rx}^2 + P_{ry}^2}$$

$$Direction = \arctan(P_{ry} / P_{rx}) - a_0$$


$$e_{rx} = P_{rx} / Distance$$

$$e_{ry} = P_{ry} / Distance$$


$$DistChng = (v_{rx} * e_{rx}) + (v_{ry} * e_{ry})$$

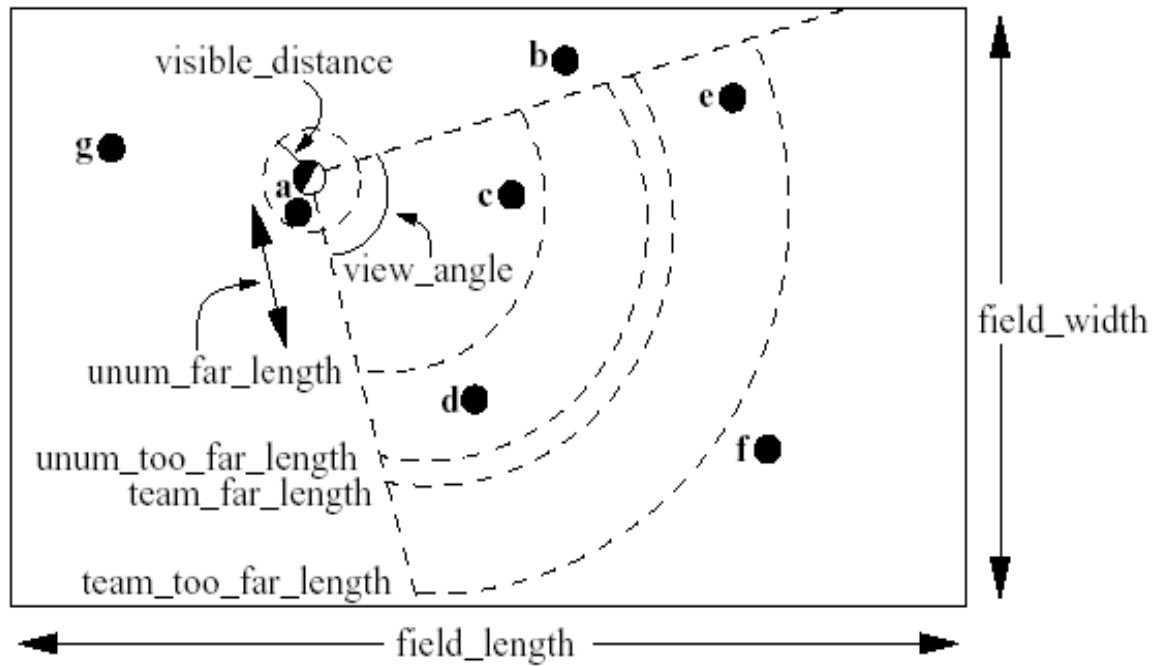
$$DirChng = [(-(v_{rx} * e_{rx}) + (v_{ry} * e_{ry})) / Distance] * (180 / \pi)$$

吉祥意

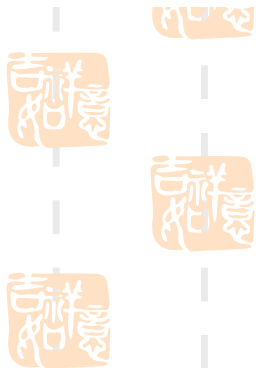


视野范围

● Client whose vision perspective is being illustrated



吉祥如意



同步视觉模式



- | mode | view width(degree) | see frequency |
- | narrow | 60 | every cycle |
- | normal | 120 | every 2 cycles |
- | wide | 180 | every 3 cycles |



视觉感知噪声



(3) 自身感知模型

■ (sense_body *Time*

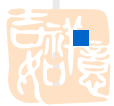
- (view_mode *ViewQuality ViewWidth*)
- (stamina *Stamina Effort*)
- (speed *AmountOfSpeed DirectionOfSpeed*)
- (head_angle *HeadDirection*)
- (kick *KickCount*)
- (dash *DashCount*)
- (turn *TurnCount*)
- (say *SayCount*)
- (turn_neck *TurnNeckCount*)
- (catch *CatchCount*)
- (move *MoveCount*)
- (change_view *ChangeViewCount*))

其中`Time`表示当前时间，`view_mode`表示视觉模式。`ViewQuality`的取值是high或low。`ViewWidth`取值是narrow, normal, wide。`Stamina`表示体力值，`Effort`表示体力使用效用。`Speed`表示速度，`AmountOfSpeed`是球员速度的近似值。`DirectionOfSpeed`是球员速度的近似方向。`head_angle`表示头部方向。`HeadDirection`是球员头部的相对方向。变量`Count`是由Server执行的对应命令的总量。如：`DashCount=134`说明球员其时已经执行了134次dash命令。

2.3.2 球员的动作模型



- (catch Direction) 扑球
- (turn Moment) 转身
- (turn_neck Angle) 转脖子
- (move X Y) 瞬移
- (dash Power,angle) 加速
- (kick Power Direction) 踢球
- (say Message) 说话
- (tackle power angle) 铲球
- (score) 咨询比赛
- (change_view angle_width quality) 改变视觉模式
- (sense_body)
- (attentionto team playernumber) 表示听觉专注于某个球员
- (change_player_type playernumber playertype) 更换球员类型
- (synch_see) 改变视觉模式为同步视觉



2.3.2 球员的动作模型 (1)

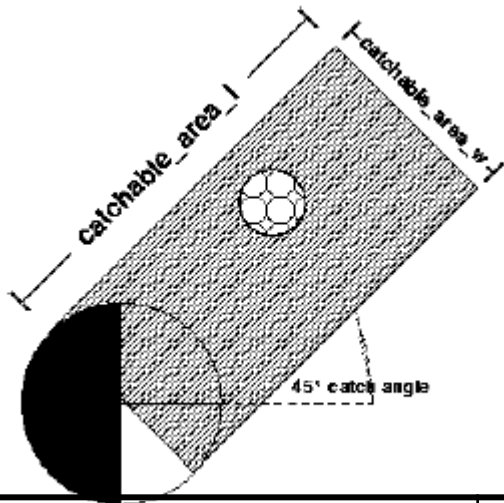
1、(*catch direction*)

- 球员扑球动作。守门员是唯一能执行**catch**命令的球员。守门员可以从任何方向扑到足球，只要足球在可扑范围内，守门员在罚球区内，且比赛模式是“**play_on**”。如果守门员以 δ 角度去**catch**足球，那么可扑范围是长宽分别是**catchable_area_l**和**catchable_area_w**的矩形区域。如果足球在这个矩形区域内，能够被扑到的可能性是**catch_probability**，在外面则不能扑到。



2.3.2球员的动作模型（1）

- 守门员与球夹角为45°时的扑球有效区域



Parameter in serer.conf	Value
Catchable_area_l	2.0
Catchable_area_w	1.0
Catchable_probability	1.0
Catch_ban_cycle	5
Goalie_max_moves	2

- 如果扑球命令失败，必须经过 *catch_ban_cycle* 周期才能执行下一次的扑球命令。
- 如果守门员成功扑住球，经过同样的周期比赛模式会变成 *goalie_catch_ball_[l|r]* 和 *free_kick_[l|r]*
- 一旦守门员扑住球，他可以带球在禁区内执行 *move* 命令移动。踢出球之前守门员可以执行 *goalie_max_moves*(2) 次的 *move* 命令，多余的 *move* 命令将不会有效并且 *server* 回返回错误 *too_many_moves*。
- 抓住球，转圈，轻轻踢球后立刻将球再次抓住后移动（不超过 *goalie_max_moves* 次的 *move* 命令），这样拖延时间被认为是不道德的。

2.3.2 球员的动作模型 (2)

■ 2、(*turn Moment*)

➤ 球员的转身动作，其参数 *moment* 属于 *minmoment* 和 *maxmoment*（缺省为到180）组成的区间。在球员运动的过程中，由于惯性的存在，转身更为困难。一般地，球员的实际转身角度由下式计算：

➤
$$\text{Actual_angle} = \text{moment} / (1.0 + \text{inertia_moment} * \text{player_speed})$$

➤ *inertia_moment* 是一个参数，缺省值为5.0。由上式可知，当一个球员以最大速度（1.05）跑动时，他最大可以转过的角度略小于30。

2.3.2 球员的动作模型 (3)

■ 3、(*turn_neck angle*)

- 球员转脖子动作。使用turn_neck命令，从某种角度上，是在独立于球员身体的转动头颈。球员的头部角度是他的视野角度。命令turn改变球员的身体角度，而命令turn_neck则改变了球员相对他的身体的颈部角度。球员颈部的相对角度介于 minmoment 和 maxmoment之间)。切记头颈角度是相对于球员身体的相对角度，如果球员执行了turn命令，而没有执行turn_neck命令，球员的视野角度也是会改变的。

2.3.2 球员的动作模型 (4)

■ 4、(*move* X Y)

- 球员瞬移动作。该动作可以把球员移动到场上的任何一个地方。只有在设置整个球队时，**move**命令有效，在正常比赛期间是没有效果的。在上下半场开始前（比赛模式是‘**before_kick_off**’）以及进球后（比赛模式是‘**goal_r_n**’和‘**goal_l_n**’）才可以使用**move**命令。在这种情况下，只要比赛模式没有改变，球员可以被移动到自己半场的任何地点（就是说 $x < 0$ ），而且可以被移动任意多次。如果球员还在对方半场的话，那么将会被**Server**移动到己方半场的随机位置。

2.3.2 球员的动作模型 (5)

■ 5、(*dash power, angle*)

- 球员的加速动作。*dash*是指在球员所面对的方向上的一个冲力。它只是一个小地向前冲的动作。它不是持续的跑。为了能持续地跑，必须要发送多个*dash*命令。
- *dash*有2个参数：*power*和*angle*，其中*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，解释一下其中的4种，*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；

2.3.2 球员的动作模型 (5)

- 球员力量的有效值dash_power:

- $\text{dash_power} = \text{effort} * \text{power} * \text{dir_rate} * \text{dash_power_rate}$

- power是力量参数

- 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

- effort表示体力使用效率。

- (dash_power, angle + angleBody) 带入公式2.5, 计算最后球员的加速度。

2.3.2 球员的动作模型 (6)

■ 6、(*kick Power Direction*)

- 球员踢球动作。**kick**命令同**dash**命令非常相似，只不过它用在球上，而非球员。球员是不能踢到**kickable-area** ($= \text{player_size} + \text{ball_size} + \text{kickable_margin}$) 以外的球。
- 令**dir_diff**表示球相对球员身体方向夹角的绝对值，**dist_ball**表示球员到球的距离，**power**为给定踢球力量，则**kick**命令的力量有效值计算方法为：

$$\begin{aligned} \text{kick_power} = & \text{kick_power_rate} * (1 - 0.25 * \text{dir_diff} / 180 - 0.25 \\ & * (\text{dist_ball} - \text{player_size} - \text{ball_size}) / \text{kickable_margin}) \\ & * \text{power} \end{aligned}$$

2.3.2 球员的动作模型 (7)

■ 7、(**say Message**)

- 球员可以使用**say**命令对其他球员进行广播消息。消息最长**say_msg_size(10)**字节，有效字母是[-0-9a-zA-Z().+*/?<>_]（不包括方括号）。球员说的消息能够内**audio_cut_dist(50)**距离内的任一支球队的球员听到。发到**server**的消息会被马上发送到可听范围内的球员处。命令**say**的使用仅仅受球员听觉能力的限制。

2.3.2 球员的动作模型 (8)

■ 8、tackle(power, angle)

- 以一定概率，在特定范围内给球在某个特定方向上一个加速度，该范围比踢球范围更大。
- 其参数为power和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失败概率：

$$fail_prob = (|player2ball.x| / tackle_dist)^{tackleExponent} + (|player2ball.y| / tackle_width)^{tackleExponent}$$

- 其中，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），同时要求player2ball.x>0。指数tackleExponent是server定义的常数，等于6。

- tackle完成后，球员有10个周期不能移动。

2.3.2 球员的动作模型 (8)

- 铲球的有效力量 etp 的计算方法为:

$$etp = tacklepower * (1.0 - (|angle| / 180)) * tacklePowerRate$$

- 其中为铲球力量，一般使用 $maxTacklePower$ (100)，其它的参数理解和 $kick$ 指令相同。 $tacklePowerRate$ 也等于0.027。
- 铲球使球产生的加速度把 etp 、 $angle + angleBody$ 代入式2.5计算得出。理论上单人单周期铲球，在极限情况可以得到2.7的加速度。



2.3.2 球员的动作模型 (9)

■ 9、score

- 球员咨询实时比分动作。球员可以通过发score来询问即时球场双方的比分。



2.3.2 球员的动作模型 (10)

■ 10 (change_view Angle_width quality)

- 球员改变视觉模式动作。**change_view**可以改变当前球员的视觉模式，2个参数分别为**angle_width**和**quality**，表示下个周期要改变的目标视觉模式，
- 如**change_view wide high** 表示从下个周期开始采用宽视觉和高质量的视觉模式，即要求视野范围为**180**度，且视觉数据是高质量的。

2.3.2 球员的动作模型 (11)

- **(sense_body)**

- 球员自身感知动作。

- **(attentionto team playernumber)**

- 球员听觉集中的动作。表示听觉专注于某个球员。包含2个参数，team和playernumber，分别表示所属球队和球员号码，当team参数为off时，表示停止该动作。

- **(change_player_type playernumber playertype)**

- 球员要求换球员类型的动作。该动作是Server在引入异构球员以后加入的。有2个参数，playernumber和playertype。表示把playernumber号球员更换为playertype类型的异构球员。

- **(pointto distance dir)**

- 球员手臂指向某一点的动作。distance 和dir分别表示该点的相对距离和角度。

- **(synch_see)**

- 球员改变视觉模式为同步视觉的动作。

2.3.2 球员的动作模型（相容-互斥动作）

- 互斥动作：不能同时发送给server
 - dash、kick、tackle及turn、catch、move
- 相容动作：可以同时发送给server
 - turn_neck、chang_view、pointto、attentionto、say、score
 - 不宜发送太多，堵塞server

球员长时间带球为什么容易被人追上？

本章概要



- Server平台概要
- Server之球场仿真
- Server之消息版
- **Server之裁判模块**
- Simple Client
- 教练-Trainer和Coach



2.4 Server之裁判模块

- 比赛模式
- 比赛模式及切换规则



2.4.1 比赛模式

- PM_BEFORE_KICK_OFF, PM_KICK_OFF_LEFT, PM_KICK_OFF_RIGHT,
- PM_KICK_IN_LEFT, PM_KICK_IN_RIGHT
- PM_CORNER_KICK_LEFT, PM_CORNER_KICK_RIGHT,
- PM_GOAL_KICK_LEFT, PM_GOAL_KICK_RIGHT,
- PM_GOAL_LEFT, PM_GOAL_RIGHT,
- PM_FREE_KICK_FAULT_LEFT, PM_FREE_KICK_FAULT_RIGHT,
- PM_FREE_KICK_LEFT, PM_FREE_KICK_RIGHT,
- PM_INDIRECT_FREE_KICK_RIGHT PM_INDIRECT_FREE_KICK_LEFT
- PM_BACK_PASS_LEFT, PM_BACK_PASS_RIGHT,
- PM_OFFSIDE_LEFT, PM_OFFSIDE_RIGHT,
- PM_PLAY_ON,
- PM_TIME_OVER,
- PM_PENALTY_SETUP_LEFT, PM_PENALTY_SETUP_RIGHT,
- PM_PENALTY_READY_LEFT, PM_PENALTY_READY_RIGHT,
- PM_PENALTY_TAKEN_LEFT, PM_PENALTY_TAKEN_RIGHT,
- PM_FROZEN, PM_QUIT

2.4.2 比赛模式及切换规则(1)

1. 开球Kick-Off

- 在kick off之前（比赛开始前，和进球后），所有的球员都必须在她自己的半场。为了能够达到这样，在每次进球得分后，采取把比赛挂起5s时间。在这个间隔中，球员可以使用move命令移动到某点，而不是跑向这个点，这样会既费时又费体力stamina。如果在5s结束后，球员还呆在对方的半场，裁判会把她移到她们自己半场的随机位置。

2. 进球Goal

- 如果一个球队进球得分，裁判要作很多事情。首先，她向所有的球员广播进球信息。然后更新比分，将球移到中点，并把比赛模式置为kick_off_x（x是left或right，代表左半场球队或右半场球队）。最后，她将比赛挂起5s，在此期间球员回到自己的半场。

2.4.2 比赛规则(2)

3. 出界Out of Field

- 当足球出界时，裁判把足球放到一个合适的位置（边线，角球区或球门区）并且把比赛模式置为界外球kick_in，角球coner_kick，或者球门球goal_kick。如果是角球的话，裁判将把足球放到场内正确的角球区坐标（1m，1m）处。

4. 清场Player Clearance

- 当比赛模式是开球kick_off，界外球kick_in，或角球corner_kick时，防守队员移出以足球为圆心，9.15m为半径的圆形区域。被移出的球员随机放置在圆形区域的周围。如果比赛模式是越位offside，所有的进攻球员被移回到没有越位的位置。这种情况下的进攻球员是指在越位范围内的所有球员以及位于以足球为圆心9.15m为半径的圆形区域内。如果比赛模式是球门球goal_kick，所有的进攻球员会被移到罚球区外。当球门球发生时，进攻球员不能重新进入罚球区。当足球被踢出罚球区后，比赛模式马上被设置为正常play_on。

2.4.2 比赛规则(3)



5. 比赛模式控制Play-Mode Control

- 当比赛模式是开球kick_off，界外球kick_in，或者角球corner_kick时，裁判在足球因为kick命令产生移动时，马上把比赛模式设置为正常play_on。

6. 中场时间和终场时间Half_Time and Time_Up

- 当上半场或下半场结束时，裁判暂时挂起比赛。每个半场的默认值是3000个仿真周期（大约5分钟）。如果在下半场结束后，还是平局的话，会进行加时赛。



2.4.2 比赛规则(4)

- 某些故意犯规动作，如故意阻挡等，很难由裁判(referee)自动判断，因为它与球员的意图有关。因此，server 为通过人来判断这些犯规动作提供了一种方式。人类裁判可以挂起比赛，赋予某个球队任意球。



2.4.2 比赛规则(5)



- 故意包围足球(Surrounding the ball)
- 故意用多名队员阻挡球(Blocking the goal with too many players)
- 故意长时间持球(Not putting the ball into play after a given number of cycles)
- 故意阻挡其他队员的移动(Intentionally blocking the movement of other players)
- 守门员滥用catch 命令（守门员不允许在罚球区内重复使用kick 和 catch 命令，因为这样为移动足球提供了安全的方式）.

■ 向server 发送过多命令Flooding the Server with Messages每个Client 在一个仿真周期内不能发送超过3 或4 个命令。过多的命令会使server 阻塞。如果server 发生阻塞，将在赛后进行检查。

■ 不恰当的行为Inappropriate Behaviour



本章概要



- Server平台概要
- Server之球场仿真
- Server之消息版
- Server之裁判模块

- Simple Client

 教练-Trainer和Coach



2.5 Client



- 每个球员程序(Client)都是独立的进程，通过一个特定的端口（默认是6000）和比赛平台连接
- 当一个球员程序和比赛平台建立好连接后，所有通讯信息都通过这个端口传输。
- 通过这个接口，客户端程序可以发送命令来控制场上的一个球员行动以及接收到这名球员的感知信息。
- 一个球队最多可以连接 12名队员包括 11 名球员（其中一个守门员）和一个场上教练。



2.5 Client(1)



- 示例程序——RcssClient，其基本功能如下：
 - 与server 建立连接，从 monitor 上即可看到球员正常上场
 - 通过终端用户可以直接看到 client从 server 接收到的消息
 - 可以通过终端输入向 server 发送命令，直接控制场上球员。



2.5 Client(2)



- 初始化球员要以下面的格式来发送init命令:

- (init TeamName [(version VerNum)] [(goalie)])

- 如:

- 1 (init MyTeam (version 9))

- 2 (init L 2 before_kick_off)

- 3 (**server_param** (audio_cut_dist 50)(auto_mode 0)(back_dash_rate 0.6)(back_passes 1)(ball_accel_max 2.7)(ball_decay 0.94)(ball_rand 0.05)(ball_size 0.085)(ball_speed_max 3)(ball_stuck_area 3)(ball_weight 0.2)(catch_ban_cycle 5)(catch_probability 1)(catchable_area_l 1.2)(catchable_area_w 1)(ckick_margin 1)(clang_advice_win 1)(clang_define_win 1)(clang_del_win 1)(clang_info_win 1)(clang_mess_delay 50)(clang_mess_per_cycle 1)(clang_meta_win 1)(clang_rule_win 1)(clang_win_size 300)(coach 0)(coach_msg_file "")(coach_port 6001)(coach_w_referee 0)(connect_wait 300)(control_radius 2)(dash_angle_step 45)



2.5 Client(3)

- (dash_power_rate 0.006)(drop_ball_time 200)(effort_dec 0.005)(effort_dec_thr 0.3)(effort_inc 0.01)(effort_inc_thr 0.6)(effort_init 1)(effort_min 0.6)(extra_half_time 100)(extra_stamina 50)(forbid_kick_off_offside 1)(foul_cycles 5)(foul_detect_probability 0.5)(foul_exponent 10)(free_kick_faults 1)(freeform_send_period 20)(freeform_wait_period 600)(fullstate_l 0)(fullstate_r 0)(game_log_compression 0)(game_log_dated 1)(game_log_dir ".")(game_log_fixed 0)(game_log_fixed_name "rcssserver")(game_log_version 5)(game_logging 1)(game_over_wait 100)(goal_width 14.02)(goalie_max_moves 2)(golden_goal 1)(half_time 300)(hear_decay 1)(hear_inc 1)(hear_max 1)(inertia_moment 5)(keepaway 0)(keepaway_length 20)(keepaway_log_dated 1)(keepaway_log_dir ".")(keepaway_log_fixed 0)(keepaway_log_fixed_name "rcssserver")(keepaway_logging 1)(keepaway_start -1)(keepaway_width 20)(kick_off_wait 100)(kick_power_rate 0.027)(kick_rand 0.1)(kick_rand_factor_l 1)(kick_rand_factor_r 1)(kickable_margin 0.7)

2.5 Client(4)

- (landmark_file “~/.rcssserver-
landmark.xml”)(log_date_format “%Y%m%d%H%M-
”)(log_times 0)(max_back_tackle_power 0)(max_dash_angle
180)(max_dash_power 100)(max_goal_kicks
3)(max_monitors -1)(max_tackle_power 100)(maxmoment
180)(maxneckang 90)(maxneckmoment 180)(maxpower
100)(min_dash_angle -180)(min_dash_power -
100)(minmoment -180)(minneckang -90)(minneckmoment -
180)(minpower -100)(nr_extra_halfs 2)(nr_normal_halfs
2)(offside_active_area_size 2.5)(offside_kick_margin
9.15)(olcoach_port 6002)(old_coach_hear
0)(pen_allow_mult_kicks 1)(pen_before_setup_wait
10)(pen_coach_moves_players 1)(pen_dist_x
42.5)(pen_max_extra_kicks 5)(pen_max_goalie_dist_x
14)(pen_nr_kicks 5)(pen_random_winner 0)(pen_ready_wait
10)(pen_setup_wait 70)(pen_taken_wait
150)(penalty_shoot_outs 1)

2.5 Client(5)

- (player_accel_max 1)(player_decay 0.4)(player_rand 0.1)(player_size 0.3)(player_speed_max 1.05)(player_speed_max_min 0.75)(player_weight 60)(point_to_ban 5)(point_to_duration 20)(port 6000)(prand_factor_l 1)(prand_factor_r 1)(profile 0)(proper_goal_kicks 0)(quantize_step 0.1)(quantize_step_l 0.01)(record_messages 0)(recover_dec 0.002)(recover_dec_thr 0.3)(recover_init 1)(recover_min 0.5)(recv_step 10)(say_coach_cnt_max 128)(say_coach_msg_size 128)(say_msg_size 10)(send_comms 0)(send_step 150)(send_vi_step 100)(sense_body_step 100)(side_dash_rate 0.4)(simulator_step 100)(slow_down_factor 1)(slowness_on_top_for_left_team 1)(slowness_on_top_for_right_team 1)(stamina_capacity 130600)(stamina_inc_max 45)(stamina_max 8000)(start_goal_l 0)(start_goal_r 0)(stopped_ball_vel 0.01)

2.5 Client(6)

➤ (synch_micro_sleep 1)(synch_mode 0)(synch_offset 60)(synch_see_offset 0)(tackle_back_dist 0)(tackle_cycles 10)(tackle_dist 2)(tackle_exponent 6)(tackle_power_rate 0.027)(tackle_rand_factor 2)(tackle_width 1.25)(team_actuator_noise 0)(team_l_start "")(team_r_start "")(text_log_compression 0)(text_log_dated 1)(text_log_dir "./")(text_log_fixed 0)(text_log_fixed_name "rcssserver")(text_logging 1)(use_offside 1)(verbose 0)(visible_angle 90)(visible_distance 3)(wind_ang 0)(wind_dir 0)(wind_force 0)(wind_none 0)(wind_rand 0)(wind_random 0))

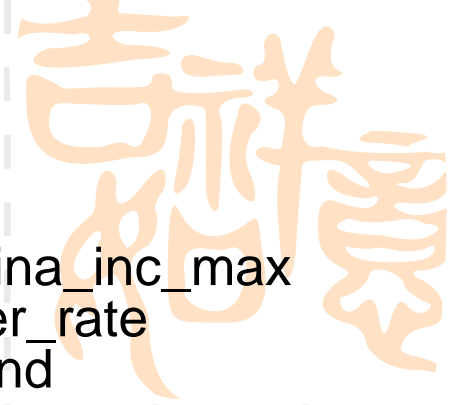
2.5 Client(7)



- > (**player_param** (allow_mult_default_type 0)(catchable_area_l_stretch_max 1.3)(catchable_area_l_stretch_min 1)(dash_power_rate_delta_max 0)(dash_power_rate_delta_min 0)(effort_max_delta_factor - 0.004)(effort_min_delta_factor -0.004)(extra_stamina_delta_max 50)(extra_stamina_delta_min 0)(foul_detect_probability_delta_factor 0)(inertia_moment_delta_factor 25)(kick_power_rate_delta_max 0)(kick_power_rate_delta_min 0)(kick_rand_delta_factor 1)(kickable_margin_delta_max 0.1)(kickable_margin_delta_min - 0.1)(new_dash_power_rate_delta_max 0.0008)(new_dash_power_rate_delta_min - 0.0012)(new_stamina_inc_max_delta_factor - 6000)(player_decay_delta_max 0.1)(player_decay_delta_min - 0.1)(player_size_delta_factor -100)(player_speed_max_delta_max 0)(player_speed_max_delta_min 0)(player_types 18)(pt_max 1)(random_seed 1290616676)(stamina_inc_max_delta_factor 0)(subs_max 3))



2.5 Client(8)



- > (**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))
- > (**player_type (id 1)**)(player_speed_max 1.05)(stamina_inc_max 48.5389)(player_decay 0.44929)(inertia_moment 6.23224)(dash_power_rate 0.00541018)(player_size 0.3)(kickable_margin 0.770318)(kick_rand 0.170318)(extra_stamina 64.5555)(effort_max 0.941778)(effort_min 0.541778)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.04856))
- > (**player_type (id 2)**)(player_speed_max 1.05)(stamina_inc_max 48.2706)(player_decay 0.423059)(inertia_moment 5.57648)(dash_power_rate 0.0054549)(player_size 0.3)(kickable_margin 0.650066)(kick_rand 0.0500655)(extra_stamina 54.8205)(effort_max 0.980718)(effort_min 0.580718)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.26727))



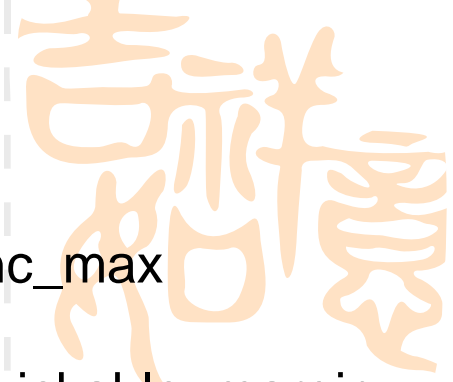
2.5 Client(9)

- > (**player_type (id 3)**)(player_speed_max 1.05)(stamina_inc_max 48.7949)(player_decay 0.359476)(inertia_moment 3.98689)(dash_power_rate 0.00536752)(player_size 0.3)(kickable_margin 0.723659)(kick_rand 0.123659)(extra_stamina 56.4249)(effort_max 0.974301)(effort_min 0.574301)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.25835))
- > (**player_type (id 4)**)(player_speed_max 1.05)(stamina_inc_max 48.861)(player_decay 0.465079)(inertia_moment 6.62698)(dash_power_rate 0.0053565)(player_size 0.3)(kickable_margin 0.726635)(kick_rand 0.126635)(extra_stamina 60.3498)(effort_max 0.958601)(effort_min 0.558601)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.1413))
- > (**player_type (id 5)**)(player_speed_max 1.05)(stamina_inc_max 47.4337)(player_decay 0.349233)(inertia_moment 3.73083)(dash_power_rate 0.00559439)(player_size 0.3)(kickable_margin 0.621817)(kick_rand 0.0218168)(extra_stamina 52.294)(effort_max 0.990824)(effort_min 0.590824)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.10691))

2.5 Client(10)

- > (**player_type (id 6)**)(player_speed_max 1.05)(stamina_inc_max 43.2308)(player_decay 0.331801)(inertia_moment 3.29504)(dash_power_rate 0.00629487)(player_size 0.3)(kickable_margin 0.618407)(kick_rand 0.0184071)(extra_stamina 58.258)(effort_max 0.966968)(effort_min 0.566968)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.28349))
- > (**player_type (id 7)**)(player_speed_max 1.05)(stamina_inc_max 41.9615)(player_decay 0.469033)(inertia_moment 6.72583)(dash_power_rate 0.00650642)(player_size 0.3)(kickable_margin 0.696455)(kick_rand 0.0964553)(extra_stamina 91.9435)(effort_max 0.832226)(effort_min 0.432226)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.12625))
- > (**player_type (id 8)**)(player_speed_max 1.05)(stamina_inc_max 45.5818)(player_decay 0.489811)(inertia_moment 7.24526)(dash_power_rate 0.00590304)(player_size 0.3)(kickable_margin 0.653601)(kick_rand 0.0536015)(extra_stamina 79.2477)(effort_max 0.883009)(effort_min 0.483009)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.12525))

2.5 Client(11)



- > (**player_type (id 9)**)(player_speed_max 1.05)(stamina_inc_max 41.3817)(player_decay 0.367623)(inertia_moment 4.19058)(dash_power_rate 0.00660305)(player_size 0.3)(kickable_margin 0.66719)(kick_rand 0.0671904)(extra_stamina 64.8683)(effort_max 0.940527)(effort_min 0.540527)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.25511))
- > (**player_type (id 10)**)(player_speed_max 1.05)(stamina_inc_max 44.9312)(player_decay 0.381243)(inertia_moment 4.53108)(dash_power_rate 0.00601146)(player_size 0.3)(kickable_margin 0.770862)(kick_rand 0.170862)(extra_stamina 65.2737)(effort_max 0.938905)(effort_min 0.538905)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.2152))
- > (**player_type (id 11)**)(player_speed_max 1.05)(stamina_inc_max 47.6851)(player_decay 0.465885)(inertia_moment 6.64713)(dash_power_rate 0.00555248)(player_size 0.3)(kickable_margin 0.736104)(kick_rand 0.136104)(extra_stamina 99.7224)(effort_max 0.80111)(effort_min 0.40111)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.2991))

2.5 Client(12)

- > (**player_type (id 12)**)(player_speed_max 1.05)(stamina_inc_max 41.1404)(player_decay 0.343584)(inertia_moment 3.58961)(dash_power_rate 0.00664327)(player_size 0.3)(kickable_margin 0.677948)(kick_rand 0.0779481)(extra_stamina 81.2578)(effort_max 0.874969)(effort_min 0.474969)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.07423))
- > (**player_type (id 13)**)(player_speed_max 1.05)(stamina_inc_max 44.3644)(player_decay 0.305605)(inertia_moment 2.64012)(dash_power_rate 0.00610593)(player_size 0.3)(kickable_margin 0.662066)(kick_rand 0.062066)(extra_stamina 61.3986)(effort_max 0.954406)(effort_min 0.554406)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.28253))
- > (**player_type (id 14)**)(player_speed_max 1.05)(stamina_inc_max 47.9259)(player_decay 0.387175)(inertia_moment 4.67936)(dash_power_rate 0.00551235)(player_size 0.3)(kickable_margin 0.684607)(kick_rand 0.0846072)(extra_stamina 88.8498)(effort_max 0.844601)(effort_min 0.444601)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.03135))

2.5 Client(13)



- > (**player_type (id 15)**)(player_speed_max 1.05)(stamina_inc_max 47.2627)(player_decay 0.358798)(inertia_moment 3.96995)(dash_power_rate 0.00562288)(player_size 0.3)(kickable_margin 0.67457)(kick_rand 0.0745702)(extra_stamina 60.6932)(effort_max 0.957227)(effort_min 0.557227)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.13539))
- > (**player_type (id 16)**)(player_speed_max 1.05)(stamina_inc_max 41.6674)(player_decay 0.425703)(inertia_moment 5.64258)(dash_power_rate 0.00655544)(player_size 0.3)(kickable_margin 0.74953)(kick_rand 0.14953)(extra_stamina 89.9971)(effort_max 0.840012)(effort_min 0.440012)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.21888))
- > (**player_type (id 17)**)(player_speed_max 1.05)(stamina_inc_max 47.584)(player_decay 0.426363)(inertia_moment 5.65907)(dash_power_rate 0.00556934)(player_size 0.3)(kickable_margin 0.663107)(kick_rand 0.0631074)(extra_stamina 73.1811)(effort_max 0.907276)(effort_min 0.507276)(kick_power_rate 0.027)(foul_detect_probability 0.5)(catchable_area_l_stretch 1.11204))
(msg 0 1 "(result 201011241637 HFUT_FINAL_0-vs-null)")



2.5 Client(14)

- (init MyTeam (version 9))
- (init L 2 before_kick_off)
- (sevrer_param Parameters...) 当前server
使用的比赛相关参数
- (player_param Parameters...) 当前server
使用的球员相关参数
- (player_type id Parameters...) 异构球员信息

2.5 Client(15)

- 重新连接是很有用的，因为无需重启比赛就可以修改球员的程序。只能在无PlayOn的比赛模式中使用（如：在半场时间）。
- 使用下面的格式进行重新连接：
 - (reconnect TeamName UniformNumber)
- Sever的反馈
 - (reconnect Side PlayMode)
 - 否则，根据不同情况，可能会是如下消息：
 - (can't reconnect) 表示比赛处于PlayOn模式,不允许reconnect;
 - (error reconnect) 表示由于某种错误，球员未能成功重新上场；
 - (error no_more_team_or_player_or_goalie) 类似于init 命令的响应。

2.5 Client(16)



- 断开连接
 - 发送给server一个bye命令。这个命令将会把球员从场上移出。
 - Server没有回应



本章概要



- Server平台概要
- Server之球场仿真
- Server之消息版
- Server之裁判模块
- Simple Client



教练-Trainer和Coach



2.6 教练



- 教练coach 是为其他球员提供帮助的特殊client。有两种coach：在线教练online coach 和训练者trainer。后者也常被叫做离线教练。
- 一般而言，训练者能对比赛实行更多的控制，仅仅在开发阶段使用；而在线教练则可以在正式比赛时使用。在自学习或者自动控制开发时，训练者是非常有用的。在线教练则是在比赛时为球员提供更多的建议和信息。



2.6 教练(1)



- **Trainer**
- 在开发球员**Client** 时，比如对带球、踢球技术进行机器学习时，以自动方法构造训练场景是非常有效的
- 训练者**Trainer**应该有以下功能：
 - (1)能控制比赛模式play-mode。
 - (2)能广播听觉消息。消息应该可以包括一个命令或者一些其他球员的信息。它的语法和语义是用户自定义的。
 - (3)能将球员和足球移动到场上的任何位置，同时可以设定他们的方向和速度。
 - (4)能得到运动物体的无噪声信息



2.6 教练(2)



- Online Coach (在线教练)
- 在线教练则是企图观察比赛然后为球员提供建议和信息。因此，他的能力有所限制：
 - (1)能和球员通讯
 - (2)能得到运动对象的无噪声信息
- 为了防止教练集中式的控制每一个球员，通讯能力被限制了，只能发有限次的消息（隔**300**个周期才能发）给球员，并且发送以后要经过一定的周期（**50**个）才能被球员接收。
- 在线教练是一个很好的工具，可以进行对方建模，比赛分析以及给己方球员战略指导。
- 由于教练能得到场上的无噪声全部信息，而且实时要求不高，因此教练可以花费更多的时间考虑战略。

