

# قوانین لیگ شبیه‌سازی فوتبال\*

دانشکده مهندسی کامپیوتر  
دانشگاه صنعتی شریف

اردیبهشت ۱۳۸۲

## ۱ مقدمه

کارگزار فوتبال یا SoccerServer سیستمی است که مأمورهای<sup>۱</sup> خود که به زبان‌های مختلف نوشته شده‌اند را قادر می‌سازد که در مقابل یکدیگر یک بازی فوتبال را انجام دهند. مسابقه به صورت کاربر-کارگزار برگزار می‌شود، به این صورت که یک کارگزار (کارگزار فوتبال) یک زمین بازی فوتبال را به صورت مجازی فراهم می‌آورد و حرکت‌های بازیکنان و توپ را در آن شبیه‌سازی می‌کند. هر کاربر حرکت یک بازیکن را کنترل می‌کند. ارتباط مابین کارگزار و هر کاربر از طریق پروتکل UDP/IP انجام می‌شود. بنابراین کاربران می‌توانند از هر سیستم برنامه‌نویسی که امکانات UDP/IP را دارد، استفاده کنند.

کارگزار فوتبال شامل دو برنامه می‌باشد: کارگزار و SoccerMonitor. کارگزار فوتبال یک برنامه‌ی کارگزار است که حرکت‌های توپ و بازیکنان را شبیه‌سازی می‌کند و با کاربران ارتباط برقرار کرده و یک بازی فوتبال را مطابق قوانین خاصی که برایش مشخص شده کنترل می‌کند. SoccerMonitor برنامه‌ای است که زمین بازی فوتبال که به صورت مجازی توسط کارگزار فوتبال فراهم شده را بر روی مانیتور نمایش می‌دهد. چندین برنامه‌ی SoccerMonitor می‌توانند به یک برنامه کارگزار

---

\*به منظور دریافت اطلاعات بیشتر به توضیحات کارگزار فوتبال که هر سال به‌روز می‌شود مراجعه کنید. آدرس توضیحات کارگزار سال ۲۰۰۲ از طریق آدرس [http://www.cs.mu.oz.au/~۲۵۷/pearce\\_lectures/robocup/docs/manual.pdf](http://www.cs.mu.oz.au/~۲۵۷/pearce_lectures/robocup/docs/manual.pdf)  
Agent<sup>۱</sup>

متصل شوند. بنابراین این امکان را داریم که یک بازی را بر روی چندین صفحه‌ی نمایش نشان دهیم.

یک کاربر با استفاده از یک درگاه<sup>۲</sup> UDP به کارگزار فوتبال متصل می‌شود. با استفاده از این درگاه کاربر فرمان‌هایی را جهت کنترل بازیکن به کارگزار می‌فرستد و اطلاعاتی را از حس‌گرهای بازیکن دریافت می‌کند. به عبارت دیگر یک برنامه‌ی کاربر مغز بازیکن را تشکیل می‌دهد. هر کاربر فقط می‌تواند یک بازیکن را کنترل کند. بنابراین یک تیم به تعداد بازیکنانش کاربر دارد. ارتباط ما بین کاربرها نیز بایستی از طریق کارگزار فوتبال و با استفاده از قراردادهای hear و say انجام شود. یکی از اهداف کارگزار فوتبال، ارزیابی سیستم‌های MultAgent یا چند مأموره می‌باشد و طبیعی است که ارتباط مابین مأموران می‌تواند به عنوان معیاری برای ارزیابی مورد استفاده قرار بگیرد.

---

<sup>۲</sup>Port

توضیح	مقدار استفاده شده	مقدار پیش گزیده	نام پارامتر
عرض دروازه	۱۴.۳۲	۷.۳۲	goali_width
شعاع یک بازیکن	۰.۳	۱.۰	player_size
وزن یک بازیکن	۶۰	۶۰	player_waight
حداکثر توان یک بازیکن	۳۵۰۰	۲۵۰۰	stamina_max
شماره درگاه UDP/IP که کارگزار از طریق آن با بازیکنان و SoccerMonitor در ارتباط است.	۶۰۰۰	۶۰۰۰	port
شماره درگاه UDP/IP که به کاربر مربی متصل است.	۶۰۰۱	۶۰۰۱	coach_port
شماره درگاه UDP/IP که به کاربر مربی online متصل است.	۶۰۰۲	۶۰۰۲	olcoach_port
حداکثر تعداد پیغام‌های کاربر مربی online	۱۲۸	۱۲۸	say_coach_cnt_max
حداکثر طول پیغام کاربر مربی online	۱۲۸	۱۲۸	say_coach_msg_size

جدول ۱: پارامترهای قابل تنظیم کارگزار شبیه‌سازی فوتبال

## ۲ پارامترها

کارگزار فوتبال و SoccerMonitor دارای پارامترهای زیادی هستند که قابل تنظیم می‌باشند. در زیر به برخی از این پارامترها اشاره می‌کنیم. برخی از پارامترهای کارگزار فوتبال که قابل تنظیم هستند، در جدول ۱ آمده است.

### ۱.۲ تنظیم پارامترهای کارگزار فوتبال با استفاده از خط فرمان

کاربر می‌تواند پارامترهای کارگزار فوتبال را که در بخش قبلی معرفی شدند، با استفاده از خط فرمان<sup>۳</sup> به صورت زیر تنظیم کند:

---

Command Line<sup>۳</sup>

```
\% SoccerServer [-ParameterName Value]*
```

کارگزار فوتبال همچنین گزینه‌های زیر را نیز می‌پذیرد.

```
-f ConfFileName
```

```
-coach-w-referee
```

## ۲.۲ فایل پیکربندی کارگزار فوتبال

یک کاربر می‌تواند پارامترهای کارگزار فوتبال را با استفاده از فایل پیکربندی کارگزار فوتبال که معمولاً server.conf نامیده می‌شود، به صورت زیر تنظیم کند:

```
\% SoccerServer -f ConfFileName
```

در فایل پیکربندی، هر خط به صورت یک زوج پارامتر-مقدار می‌باشد:

Paramter : Value

در این فایل، هر خطی که با '#' آغاز شده باشد به عنوان Comment در نظر گرفته می‌شود.  
مثال:

```
log_file : game.log
```

```
record: game.rec
```

با استفاده از این دو فرمان، log یک بازی در فایل game.log ذخیره می‌شود و فرامین فرستاده شده از کلیه بازیکنان در فایل game.rec ذخیره می‌شود.

## ۳.۲ پارامترهای SoccerMonitor

برخی از پارامترهای SoccerMonitor که قابل تنظیم هستند در جدول ۲ آمده است.

نام پارامتر	مقدار پیش گزیده	مقدار استفاده شده	توضیح
host	localhost	localhost	نام host که کارگزار بر روی آن در حال اجراست.
port	۶۰۰۰	۶۰۰۰	شماره درگاه UDP/IP که از طریق آن می توان به کارگزار متصل شد
team_l_color	Gold	Gold	رنگ تیم سمت چپ در زمان آغاز بازی
team_r_color	Red	Red	رنگ تیم سمت راست در زمان آغاز بازی

جدول ۲: پارامترهای قابل تنظیم SoccerMonitor

## ۴.۲ تنظیم پارامترهای SoccerMonitor

مشابه تنظیم پارامترهای کارگزار فوتبال، پارامترهای SoccerMonitor را نیز می توانیم با استفاده از خط فرمان و یا با استفاده از فایل پیکربندی تنظیم کنیم. برای تنظیم پارامترهای SoccerMonitor با استفاده از خط فرمان، از فرمان زیر استفاده می شود:

```
\% Soccermonitor [-ParameterName Value]*
```

برای تنظیم پارامترهای SoccerMonitor با استفاده از فایل پیکربندی که معمولاً monitor.conf است، از این فرمان استفاده می شود:

```
\% Soccermonitor -f ConfFileName
```

در فایل پیکربندی، هر خط به صورت یک زوج پارامتر-مقدار می باشد:

```
Paramter : Value
```

در این فایل، هر خطی که با '#' آغاز شده باشد به عنوان Comment در نظر گرفته می شود.

## ۳ قوانین مسابقه

یک مسابقه‌ی شبیه‌سازی روبات فوتبالیست شامل گام‌های زیر می‌باشد:

- ۱) هر یک از بازیکنان دو تیم با استفاده از فرمان Init به کارگزار وصل می‌شوند.
- ۲) وقتی که تمام بازیکنان آماده بازی شدند، داور مسابقه با فشردن دکمه Kick\_off مسابقه را شروع می‌کند. با این عمل، نیمه نخست بازی آغاز می‌شود.
- ۳) نیمه نخست بازی ۵ دقیقه طول می‌کشد. وقتی که این نیمه به پایان رسید، کارگزار بازی را به حالت تعلیق در می‌آورد.
- ۴) زمان بین دو نیمه ۵ دقیقه است که در طول آن شرکت‌کنندگان می‌توانند برنامه‌ی بازیکنان را عوض کنند.
- ۵) قبل از آغاز نیمه‌ی دوم، هر کاربر با استفاده از فرمان reconnect به کارگزار وصل می‌شود.
- ۶) وقتی که تمام کاربران حاضر شدند، داور مسابقه مجدداً با فشردن دکمه Kick-off بازی را آغاز می‌کند.
- ۷) نیمه‌ی دوم بازی نیز ۵ دقیقه طول می‌کشد. بعد از این نیمه، کارگزار بازی را متوقف می‌کند.
- ۸) در صورت مساوی شدن دو تیم، یک نیمه‌ی اضافی آغاز می‌شود. در این نیمه از قانون گل‌تلاشی استفاده می‌شود، یعنی هر تیمی که زودتر گل بزند، برنده مسابقه خواهد بود.

## ۱.۳ ویژگی‌های زمین بازی

زمین بازی و تمام اشیای موجود در آن به صورت دو بعدی می‌باشند. زمین بازی  $105m \times 68m$  می‌باشد. عرض دروازه  $14.6m$ ، یعنی دو برابر اندازه‌ی دروازه در مسابقات فوتبال واقعی می‌باشد. تجربه نشان داده است که گل‌زدن به دروازه با اندازه‌ی واقعی بسیار مشکل خواهد بود. توپ و بازیکنان به شکل دایره می‌باشند و تمام زوایا و فاصله‌ها با توجه به مرکز این دایره‌ها مشخص می‌شوند. توپ و بازیکنان به عنوان اشیاء متحرک زمین بازی، شناخته می‌شوند. قوانین مسابقه دو دسته هستند. قوانینی که توسط کارگزار و به صورت خودکار داوری می‌شوند و قوانینی که توسط داور انسانی داوری می‌شوند.

## ۲.۳ داور خودکار و داور انسانی

قوانینی که توسط کارگزار فوتبال داوری می‌شوند، به صورت زیر هستند:

- قبل از شروع بازی (زمان Kick-Off یا بعد از ردو بدل شدن گل)، همه بازیکنان باید به زمین خود منتقل شوند. بدین منظور، بعد از هر گل، داور خودکار بازی را به مدت ۵ ثانیه متوقف می‌کند تا بازیکنان هر تیم به مکان دلخواه در زمین خود حرکت (Move) کنند. اگر در پایان این مدت بازیکنی در زمین حریف بماند، کارگزار او را به یک مکان تصادفی در نیمه خود منتقل می‌کند.
  - زمانی که تیمی گل می‌زند، کارگزار ابتدا یک پیغام به همه ارسال می‌کند. سپس نتیجه بازی را به روزرسانی می‌کند و توپ را به مرکز زمین منتقل می‌کند. سپس حالت بازی را به Kick-off-x که در آن x چپ یا راست است تغییر می‌دهد. سپس بازی را به مدت ۵ ثانیه متوقف می‌کند تا بازیکنان به زمین خود بازگردند.
  - هنگامی که توپ به خارج از زمین برود، داور توپ را به مکان مناسبی انتقال می‌دهد و وضعیت بازی را به یکی از حالت‌های `kick_in`، `corner_kick`، یا `goal_kick` تغییر می‌دهد.
  - وقتی که بازی در یکی از وضعیت‌های توقف مانند `corner` قرار داشته باشد، داور بازیکنان حریف را به بیرون از شعاع مشخصی از توپ هدایت می‌کند. ضمناً در این حالت، با اولین فرمان `kick` وضعیت بازی به حالت `play_on` تغییر داده می‌شود.
  - در انتهای هر نیمه که مدت آن ۵ دقیقه یا ۳۰۰۰ چرخه شبیه‌سازی است، داور ماشینی مسابقه را به حالت تعلیق در می‌آورد و شروع مجدد توسط داور انسانی انجام می‌شود.
- کشف خطاهایی مانند سد کردن برای داور خودکار مشکل است، زیرا آنها به نیت بازیکنان ارتباط دارند. بنابراین، کارگزار راهی را برای اعمال داوری یک انسان روی چنین خطاهایی آماده کرده است. چنین خطاهایی با استفاده از امکانات خاص روی برنامه‌ی کارگزار، توسط داور انسانی قابل اعلام هستند. از خطاهایی که به داور انسانی احتیاج دارند می‌توان به موارد زیر اشاره کرد:
- احاطه کردن توپ.
  - پرکردن دروازه با استفاده از تعدادی بازیکن.
  - سد کردن راه عبور سایر بازیکنان از روی عمد.

- سوء استفاده دروازه بان از امکان گرفتن توپ با استفاده‌ی پشت سرهم از دستورات kick و catch.

- اشباع کردن کارگزار فوتبال و شبکه توسط پیغام‌های زیاد.

### ۳.۳ فرمان‌های کنترلی بازیکنان

مجموعه فرمان‌های کنترلی بازیکنان عبارتند از:

- turn Moment

با توجه به Moment جهت بازیکن را تغییر می‌دهد. Moment می‌تواند بین minmoment و maxmoment باشد. به صورت پیش‌گزیده این مقدار بین  $180^\circ$  و  $-180^\circ$  می‌باشد. اما در هنگام حرکت، با توجه به اینرسی حرکتی، ممکن است که میزان واقعی که بازیکن تغییر جهت می‌دهد متفاوت با مقداری باشد که در آرگومان Moment برای دستور turn مشخص شده است. اندازه‌ی واقعی زاویه‌ای که بازیکن تغییر جهت می‌دهد، با توجه به فرمول زیر به دست می‌آید:

$$actual\_angle = Moment / (1.0 + inertia\_moment \times player\_speed)$$

*player\_speed* طول بردار سرعت بازیکن و *inertia\_moment* پارامتری است که مقدار پیش‌گزیده‌ی آن  $5.0$  می‌باشد.

- turn\_neck Angle

هر بازیکن یک گردن دارد که می‌تواند مستقل از بدنش گردش کند. زاویه‌ی سر بازیکن، زاویه‌ی دید بازیکن است. فرمان turn\_neck زاویه‌ی سر بازیکن را نسبت به بدنش تغییر می‌دهد. Angle در بازه‌ی  $[90, -90]$  می‌تواند تغییر کند.

- dash Power

سرعت بازیکن را در جهتی که بازیکن قرار دارد به اندازه‌ی  $Power \times dash\_power\_rate$  افزایش می‌دهد. Power بایستی در بازه‌ی minpower و maxpower (به صورت پیش‌گزیده:  $[100, -30]$ ) باشد.

- kick Power Direction

اگر که توپ به اندازه‌ی کافی نزدیک به بازیکن باشد آن‌گاه بازیکن در جهت Direction و با توانی به اندازه‌ی Power به توپ ضربه‌ای وارد می‌کند. Power بایستی بین



minpower و maxpower باشد و Direction بایستی بین minmoment و maxmoment باشد (به صورت پیش گزیده:  $[-180, 180]$ ). فاصله‌ی توپ و بازیکن نیز بایستی کمتر از  $kickable\_margin + ball\_size + player\_size$  باشد.

● move X Y

بازیکن را به مکان  $(X, Y)$  منتقل می‌کند. مبدأ ما مرکز زمین خواهد بود و محور  $X$  به سمت دروازه‌ی حریف و محور  $Y$  به سمت راست می‌باشد. از این فرمان فقط در زمانی که بازی در وضعیت before\_kick\_off است می‌توان استفاده کرد و یا زمانی که دروازه‌بان توپ را گرفته است.

● catch Direction

سعی می‌کند که توپ را در جهت Direction بگیرد. Direction بایستی که در بازه‌ی  $[-180, 180]$  باشد. فقط دروازه‌بان می‌تواند از این فرمان استفاده کند.

● say Message

پیغام Message را به تمام بازیکنان انتشار می‌دهد. Message بایستی که یک رشته با طول کمتر از ۵۱۲ نویسه باشد. البته یک حداکثر فاصله داریم که از آن فاصله بیشتر، دیگر پیغام شنیده نخواهد شد. این پیغام از طریق کارگزار و با استفاده از فرمان hear به گوش دیگر کاربران می‌رسد.

● change\_view ANGLE\_WIDTH QUALITY

این فرمان زاویه‌ی دید و کیفیت اطلاعات بصری را تغییر می‌دهد. ANGLE\_WIDTH بایستی که یکی از مقادیر  $narrow = 45$ ،  $normal = 90$ ، و یا  $wide = 180$  باشد. QUALITY می‌تواند یکی از مقادیر low و یا high را داشته باشد. اگر QUALITY=high باشد، آنگاه کارگزار اطلاعات دقیقی راجع به مکان اشیاء به کاربر می‌فرستد و اگر QUALITY=low باشد، آنگاه کارگزار اطلاعات کمتری راجع به مکان اشیاء به کاربر می‌فرستد (فقط جهت‌ها را می‌فرستد و فاصله‌ها را نمی‌فرستد). مقادیر پیش‌گزیده‌ی ANGLE و QUALITY به ترتیب normal و high می‌باشند.

از طرف دیگر، بسامد فرستادن اطلاعات بصری که به کاربر فرستاده می‌شود، بستگی به مقادیر ANGLE و QUALITY دارد.

### ۴.۳ توان (Stamina)

هر بازیکن دارای توانی می‌باشد. در این ارتباط سه پارامتر زیر تعریف می‌شوند:

(۱) stamina: در زمانی که بازیکن عمل dash یا kick انجام می‌دهد از آن کاسته می‌شود و در هر چرخه‌ی شبیه‌سازی به تدریج بازیابی می‌شود.

(۲) effort: مشخص می‌سازد که عمل dash تا چه اندازه مؤثر بوده است.

(۳) recovery: کنترل می‌کند که پارامتر stamina تا چه میزان در هر چرخه‌ی شبیه‌سازی بازیابی شود.

یک بازیکن می‌تواند عمل dash را با یک آرگومان Power کمتر از پارامتر stamina فعلی خود انجام دهد و پارامتر stamina بعد از عمل dash به اندازه‌ی Power کاسته می‌شود. یعنی اگر یک بازیکن عمل dash انجام دهد، آن‌گاه:

$$stamina := \max(stamina - Power, 0)$$

$$Effective\_Power := effort \times (Power, stamina)$$

اگر علامت Power منفی باشد، باعث می‌شود که از پارامتر stamina به اندازه‌ی دو برابر کاسته شود. در هر چرخه‌ی شبیه‌سازی این سه پارامتر به صورت زیر تغییر می‌کنند:

• Stamina: stamina به تدریج افزایش می‌یابد:

$$stamina = \min(stamina\_max, stamina + recovery \times stamina\_inc\_max)$$

• Effort: ایده‌ی اصلی این است که اگر stamina کم شود، effort کاسته شود و اگر stamina افزایش یابد، effort افزایش یابد:

$$effort = \begin{cases} \max(effort\_min, effort - effort\_dec) & \text{if } stamina \leq (effort\_thr \times stamina\_max) \\ \min(1, 0, effort + effort\_inc) & \text{if } stamina \geq (effort\_thr \times stamina\_max) \\ effort & \text{otherwise} \end{cases}$$

• Recovery: مشابه پارامتر effort می‌باشد، با این تفاوت که recovery هرگز افزایش نمی‌یابد:

$$recovery = \begin{cases} \max(recovery\_min, recovery - recovery\_dec) & \text{if } stamina \leq (recovery\_dec\_thr \times stamina\_max) \\ recovery & \text{otherwise} \end{cases}$$

ترتیب تغییر پارمترهای فوق به این صورت است که ابتدا پارامتر stamina کاسته می‌شود (اگر یک بازیکن عمل dash را انجام داده باشد) و سپس مقدار recovery تغییر می‌کند و سپس effort و در نهایت stamina تا حدودی بازیابی می‌شود.

## ۴ اطلاعات حس گرها

### ۱.۴ اطلاعات حس گرینایی

این اطلاعات با استفاده از فرمان زیر، از کارگزار به برنامه‌های کاربران منتقل می‌شود:

(see Time ObjInfo ObjInfo ...)

در این فرمان، Time زمان فعلی را نشان می‌دهد. ObjInfo اطلاعاتی راجع به یک شیء قابل رؤیت در میدان بازی را می‌دهد. برای ObjInfo از قالب زیر استفاده می‌شود:

(ObjName Distance Direction DistChng DirChng BodyDir HeadDir)

```
ObjName := (player Teamname UniformNumber)
          | (goal [l|r])
          | (ball)
          | (flag c)
          | (flag [l|c|r][t|b])
          | (flag p[l|r][t|c|b])
          | (flag g[l|r][t|b])
          | (flag [l|r|t|b] 0)
          | (flag [t|b][l|r][10|20|30|40|50])
          | (flag [l|r][t|b][10|20|30])
          | (line [l|r|t|b])
```

مقادیر Distance، Direction، DistChng، و DirChng به صورت زیر محاسبه می‌شوند:

$$P_{rx} = P_{xt} - P_{x_0}$$

$$P_{ry} = P_{yt} - P_{y_0}$$

$$v_{rx} = v_{xt} - v_{x_0}$$

$$v_{ry} = v_{yt} - v_{y_0}$$

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

$$Direction = \arctan\left(\frac{P_{ry}}{P_{rx}}\right) - a_0$$

$$e_{rx} = \frac{P_{rx}}{Distance}$$

$$e_{ry} = \frac{P_{ry}}{Distance}$$

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

$$DirChng = \left[ \frac{-(v_{rx} \times e_{rx}) + (v_{ry} \times e_{ry})}{Distance} \right] \times \left( \frac{180}{\pi} \right)$$

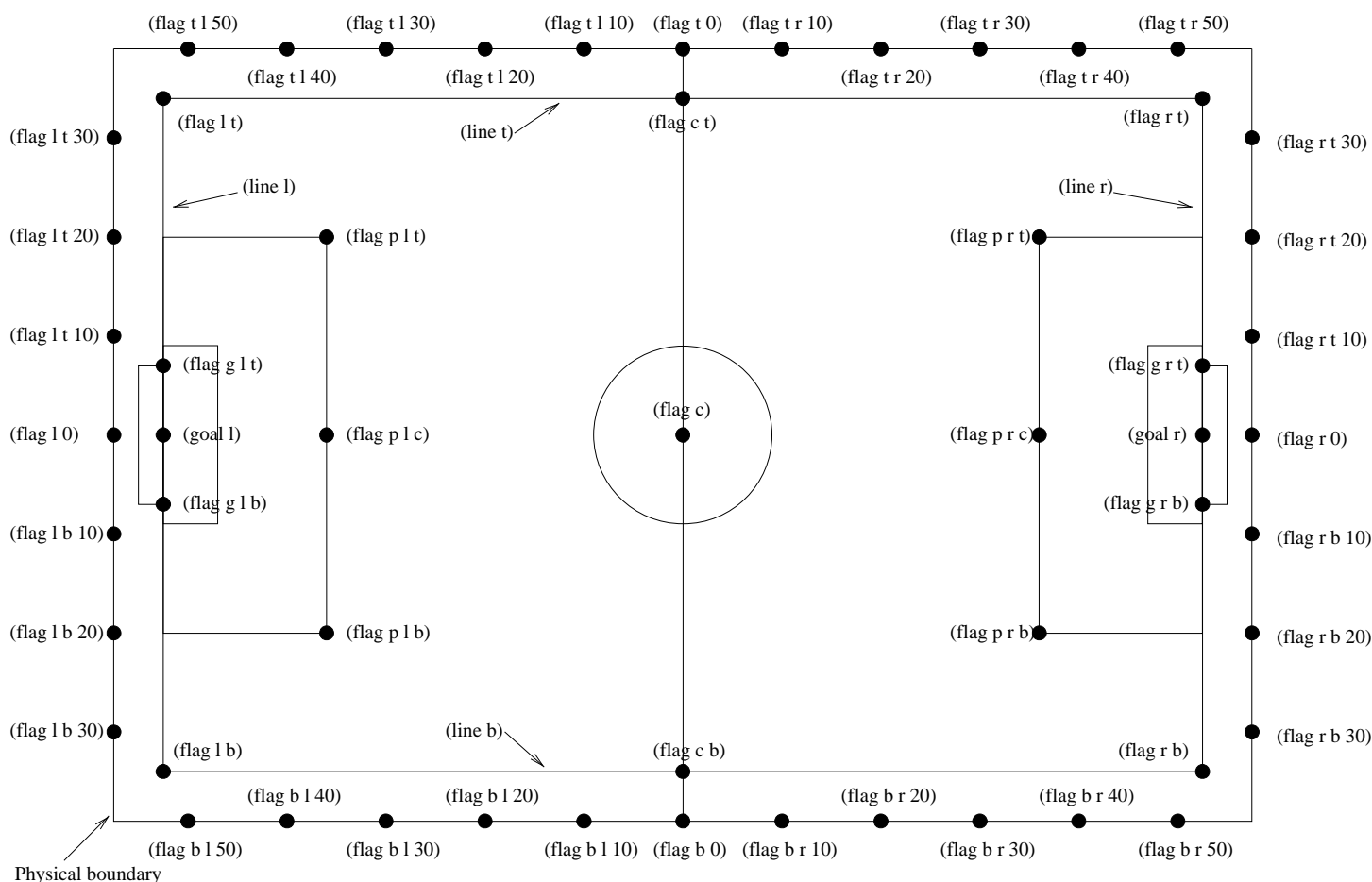
$(P_{xt}, P_{yt})$  مکان شیء قابل رؤیت و  $(P_{x_0}, P_{y_0})$  مکان بازیکن،  $(v_{xt}, v_{yt})$  سرعت شیء قابل رؤیت و  $(v_{x_0}, v_{y_0})$  سرعت بازیکن، و  $a_0$  جهت بازیکن است. به عبارت دیگر،  $a_0$  برابر مجموع BodyDir و HeadDir آن بازیکن است. توجه شود که HeadDir و BodyDir به صورت نسبی تعریف می‌شوند. یعنی اگر بازیکن رؤیت شونده و بازیکن رؤیت کننده هر دو بدنشان به یک سمت باشد، آن‌گاه BodyDir برابر صفر است.

در مورد خط این نکته قابل ذکر است که Distance برابر است با فاصله‌ی بازیکن تا محل برخورد آن خط با خطی که از مرکز دید بازیکن می‌گذرد.

زاویه دید بازیکن به عوامل زیادی بستگی دارد. اول پارامترهای sense-step و visible-angle در داخل کارگزار است که فاصله زمانی بین دو عمل دیدن و زاویه دید بازیکن را نشان می‌دهد است (مقادیر اولیه در حال حاضر ۱۵۰ میلی ثانیه و ۹۰ درجه است). عامل دیگر، خود بازیکن است که می‌تواند نرخ و کیفیت بینایی‌اش را با تغییر پارامترهای ViewWidth و ViewQuality تغییر دهد. برای محاسبه نرخ بینایی و زاویه دید کنونی بازیکن، از معادلات زیر استفاده می‌شود:

$$view - frequency = sense - step * view - quality - factor * view - width - factor$$

که در آن view-quality-factor برابر ۱ است اگر و فقط اگر ViewQuality برابر high و برابر ۰/۵ است اگر ViewQuality برابر low باشد. مقدار view-width-factor هم ۲ است اگر و فقط اگر



شکل ۱: پرچم‌ها و خطوط در محیط شبیه‌سازی فوتبال

ViewWidth برابر narrow و ۱ است اگر فقط اگر برابر normal و ۰/۵ است اگر wide باشد.

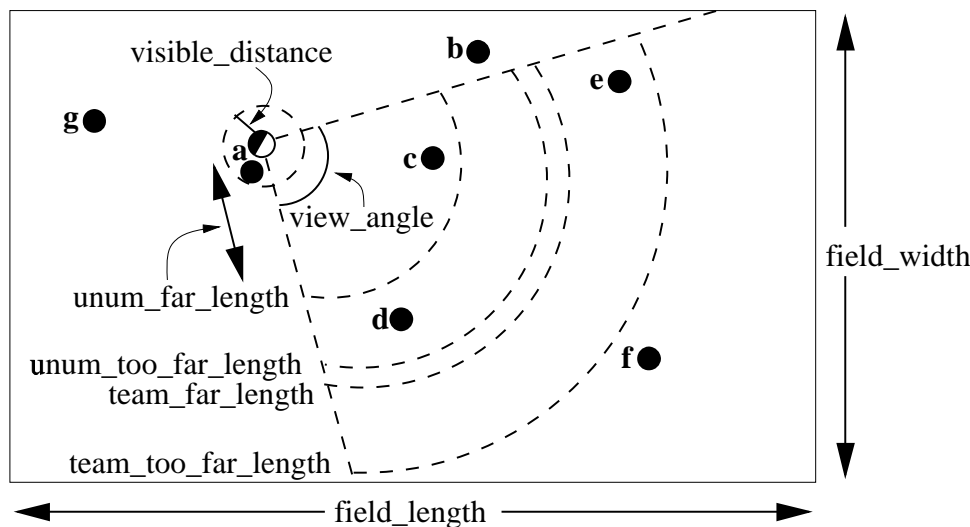
$$view - angle = visible - angle * view - width - factor$$

که در آن view-width-factor برابر ۰/۵ است اگر فقط اگر ViewWidth برابر narrow، ۱ است اگر normal و ۲ است اگر wide باشد.

بازیکن همچنین می‌تواند یک شیء که در فاصلهٔ visible-distance متری آن است را ببیند. اگر شیء در داخل این فاصله قرار داشته باشد ولی در زاویهٔ دید بازیکن نباشد، بازیکن تنها قادر به فهمیدن نوع آن شیء است (توپ، بازیکن، دروازه یا پرچم) ولی نام دقیق آن قابل دستیابی نیست. نمونه‌ای از مفهوم زاویهٔ دید در شکل ۲ آمده است.

به منظور کاهش دقت اطلاعات بینایی رسیده از کارگزار، مقادیر ارسال شده از طریق کارگزار به بازه‌هایی کوانتیزه می‌شوند. به عنوان مثال، فاصلهٔ یک شیء توپ یا بازیکن با فرمول زیر کوانتیزه

⊙ Client whose vision perspective is being illustrated



شکل ۲: مثالی از محدوده دید یک بازیکن

می شود:

$$d' = \text{Quantize}(\exp(\text{Quantize}(\log(d), \text{quantize} - \text{step})), 0.1)$$

که در آن  $d$  و  $d'$  فاصله دقیق و فاصله تقریبی هستند. تابع  $\text{Quantize}$  نیز به صورت

$$\text{Quantize}(V, Q) = \text{ceiling}(V/Q) \cdot Q$$

می باشد.

بدین ترتیب، بازیکنان قادر به دریافت مکان دقیق اشیای دور نخواهند بود. مثلاً زمانی که فاصله حدود ۱۰۰ متر است، مقدار نویز حداکثر ۱۰ متر و زمانی که فاصله کمتر از ۱۰ متر باشد، نویز حداکثر ۱ متر است.

## ۲.۴ مدل حس‌گر شنوایی

پیغام‌های شنوایی زمانی ارسال می‌شوند که یک بازیکن و یا مربی پیغام say به کارگزار بفرستد. پیغام‌های داور نیز از طریق شنوایی به بازیکنان می‌رسد. تمامی پیغام‌ها بلافاصله به بازیکنان می‌رسد.

یک پیغام شنوایی در قالب زیر است:

(hearTimeSender" Message")

در صورتی که منبع صدا یک بازیکن باشد، پارامتر Sender جهت نسبی او است و در غیر این صورت یکی از مقادیر زیر را خواهد داشت:

- self: اگر ارسال کننده پیغام خود بازیکن باشد.
- referee: اگر ارسال کننده پیغام داور باشد.
- online-coach-right یا online-coach-left: اگر ارسال کننده پیغام یکی از داورهای خط‌نگه‌دار باشد.

حداکثر طول پیغام (Message) برابر say-msg-size است. پیغام‌های مشخصی که از طرف داور ارسال می‌شود در بخش مربوطه خواهد آمد. پارامترهای کارگزار که در کیفیت این حس‌گر تاثیرگذار است در جدول ۳ آمده است.

پارامتر در soccer.conf	مقدار
audio-cut-dist	۵۰/۰
hear-max	۲
hear-inc	۱
hear-delay	۲
say-msg-size	۵۱۲

جدول ۳: پارامترهای تاثیرگذار در حس‌گر شنوایی

یک بازیکن تنها زمانی می‌تواند یک پیغام را بشنود که ظرفیت شنوایی او حداقل hear-decay باشد. مقدار این پارامتر با شنیده شدن یک پیغام کاهش پیدا می‌کند. در هر چرخه بازی، این ظرفیت به اندازه hear-inc اضافه می‌شود. بیشترین مقدار پارامتر ظرفیت شنوایی hear-max است. به منظور جلوگیری از مختل یا بلااستفاده شدن ارتباطات یک تیم توسط بازیکنان حریف، کانال‌های ارتباطی دو تیم از یکدیگر مجزا است. در پیکربندی کنونی کارگزار، هر بازیکن در هر سیکل حداکثر یک پیغام از هر تیم را می‌شنود. اگر پیغام‌های زیادی در یک چرخه به یک بازیکن برسند، اولین پیغام شنیده می‌شود. این قانون شامل پیغام‌های کارگزار یا داورها نمی‌شود.

### ۳.۴ مدل حرکت اشیاء

در هر چرخه‌ی شبیه‌سازی، پارامترهای مربوط به اشیاء متحرک به روش زیر محاسبه می‌شود:



$$\begin{aligned}
(u_x^{t+1}, u_y^{t+1}) &= (v_x^t, v_y^t) + (a_x^t, a_y^t) \\
(p_x^{t+1}, p_y^{t+1}) &= (p_x^t, p_y^t) + (u_x^{t+1}, u_y^{t+1}) \\
(v_x^{t+1}, v_y^{t+1}) &= decay \times (u_x^{t+1}, u_y^{t+1}) \\
(a_x^{t+1}, a_y^{t+1}) &= (0, 0) : \text{reset acceleration}
\end{aligned}$$

در فرمول‌های فوق،  $(u_x^t, u_y^t)$  معرف میزان جابجایی،  $(p_x^t, p_y^t)$  معرف مکان و  $(v_x^t, v_y^t)$  نشان دهنده‌ی سرعت شیء می‌باشد.  $decay$  ضریب اضمحلال نامیده می‌شود و توسط پارامترهای  $ball\_decay$  و یا  $player\_decay$  مشخص می‌شود.  $(a_x^t, a_y^t)$  شتاب حرکت شیء است و با استفاده از پارامتر  $Power$  در فرمان‌های  $dash$  (اگر شیء مورد نظر بازیکن باشد) و یا  $kick$  (اگر شیء مورد نظر توپ باشد) به صورت زیر محاسبه می‌شود:

$$(a_x^t, a_y^t) = Power \times power\_rate \times (Cos(\theta^t), Sin(\theta^t))$$

$\theta^t$  نشان دهنده‌ی جهت شیء در زمان  $t$  می‌باشد و  $power\_rate$  از طریق پارامترهای  $dash\_power\_rate$  و یا  $kick\_power\_rate$  به دست می‌آید. اگر شیء مورد نظر یک توپ می‌باشد، جهت آن را به صورت زیر محاسبه می‌کنیم:

$$\theta_{ball}^t = \theta_{kicker}^t + Direction$$

$\theta_{ball}^t$  جهت توپ،  $\theta_{kicker}^t$  جهت بازیکنی است که ضربه را زده است و  $Direction$  دومین پارامتر فرمان  $kick$  است.

- تصادم  
اگر که در انتهای چرخه شبیه‌سازی، دوشیء با یکدیگر هم‌پوشانی<sup>۴</sup> داشته باشند، آن‌گاه اشیاء آن‌قدر به عقب رانده می‌شوند که با یکدیگر هم‌پوشانی نداشته باشند و سپس سرعت آن‌ها در ۰.۱ - ضرب می‌شود.

- نویز  
به منظور منعکس کردن حرکت‌های ناخواسته‌ی اشیاء در دنیای واقعی، کارگزار فوتبال به حرکت‌های اشیاء و پارامترهای دستورات مقداری نویز اضافه می‌کند. در مورد حرکت اشیاء، نویز را به صورت زیر در معادله‌ی حرکت اضافه می‌کنیم:

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

---

Overlap<sup>f</sup>

به گونه‌ای که  $\tilde{r}_{r_{max}}$  یک عدد تصادفی با توزیع یکنواخت در بازه‌ی  $[-r_{r_{max}}, r_{r_{max}}]$  می‌باشد.  $r_{r_{max}}$  پارامتری است که وابسته به سرعت حرکت شیء می‌باشد، یعنی:

$$r_{r_{max}} = rand \times |(v_x^t, v_y^t)|$$

به گونه‌ای که  $rand$  ضریبی است که توسط پارامترهای  $ball\_rand$  و  $player\_rand$  مشخص می‌شود.

برای اضافه کردن نویز به آرگومان‌های  $Power$  و  $Moment$  یک فرمان، به صورت زیر عمل می‌شود:

$$argument = (1 + \tilde{r}_{rand}) \times argument$$

## ۴.۴ مدل حس لامسه

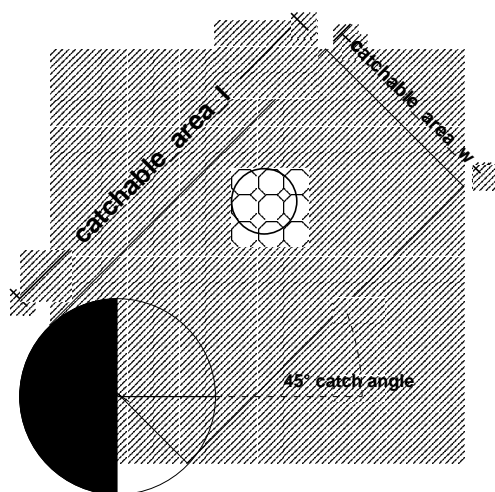
حس لامسه<sup>۵</sup> وضعیت فیزیکی بازیکن را نشان می‌دهد. این اطلاعات در هر sense-body-step میلی‌ثانیه به بازیکن ارسال می‌شود. قالب این پیغام به صورت زیر است:

(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))

در این پیغام

---

Body Sensor<sup>۵</sup>



شکل ۳: منطقه‌ای در زاویه ۴۵ درجه که دروازه‌بان می‌تواند توپ را در آن دریافت کند.

- ViewQuality یکی از مقادیر high و low است.
- ViewWidth یکی از مقادیر narrow و normal و wide است.
- AmountOfSpeed تخمینی از سرعت بازیکن است.
- DirectionOfSpeed تخمینی از جهت حرکت بازیکن است. متغیر Count تعداد تمامی دستوراتی از این نوع است که کارگزار اجرا کرده است. به عنوان مثال،  $DashCount = 134$  یعنی بازیکن ۱۳۴ بار دستور dash را اجرا کرده است.

## ۵.۴ مدل دریافت توپ

دروازه‌بان تنها بازیکنی است که قادر به گرفتن توپ است. این بازیکن تنها در حالت play-on می‌تواند توپ را از هر جهتی تصاحب کند به شرط اینکه در در منطقه جریمه باشد و توپ نیز به حد کافی به او نزدیک باشد. اگر دروازه‌بان در زاویه  $\varphi$  توپ را بگیرد، منطقه‌ای که توپ در آن قابل دستیابی است، یک مستطیل در جهت  $\varphi$  می‌باشد به طول catchable-area-l و catchable-area-w می‌باشد (شکل ۳). اگر توپ در این مستطیل قرار داشته باشد، توپ به احتمال catch-probability توسط دروازه‌بان دریافت می‌شود. جدول ۴ مقادیر پارامترهای دستور catch را نشان می‌دهد. اگر دستور catch توسط دروازه‌بان با موفقیت انجام نشود، دروازه‌بان می‌تواند دستور دیگر catch را پس از گذشت حداقل catch-ban-cycle سیکل بعد اجرا کند. اگر دریافت توپ با موفقیت انجام

مقدار	پارامتر در server.conf
۲/۰	catchable-area-l
۱/۰	catchable-area-w
۱/۰	catch-probability
۵	catch-ban-cycle
۲	goalie-max-moves

جدول ۴: پارامترهای تاثیرگذار در دریافت دروازه‌بان

شود، حالت بازی به  $goalie-catch-ball-[l-r]$  و سپس در همان سیکل به  $free-kick-[l-r]$  تغییر می‌کند. دروازه‌بان می‌تواند از دستور `move` به منظور حرکت در منطقه جریمه استفاده کند. این دستور حداکثر می‌تواند `goalie-max-moves` بار توسط دروازه‌بان اجرا شود و `move` های بعد از آن بدون تاثیر خواهند بود.

## ۶.۴ مدل شتاب‌گیری بازیکن

دستور `dash` به منظور شتاب‌گیری یک بازیکن در یک جهت خاصی نسبت به بدنش استفاده می‌شود. مقدار معتبر پارامتر شتاب در این دستور در فایل‌های پیکربندی کارگزار توسط `minpower` و `maxpower` مشخص می‌شود. جدول ۴ مقادیر پیش‌فرض این پارامترها را نشان می‌دهد. هر بازیکن یک مقدار مشخصی توان (`stamina`) دارد که با دستور `dash` کاهش می‌یابد. در پایان هر نیمه، میزان توان هر بازیکن با `stamina-max` مقادیردهی می‌شود. اگر بازیکن به سمت جلو شتاب بگیرد ( $power > 0$ )، مقدار `stamina` به اندازه `power` کاهش می‌یابد. شتاب منفی با کم‌شدن دو چندان `stamina` همراه است. اگر `stamina` یک بازیکن کمتر از مقدار لازم برای یک `dash` باشد، `power` به نحوی کاهش می‌یابد که دستور `dash` با `stamina` موجود قابل اجرا باشد و به میزان بیش از آن احتیاج نداشته باشد. بازیکنان غیریکسان هر بار توان موجودشان کمتر از `stamina` لازم باشد، از مقداری `stamina` اضافه استفاده می‌کنند که به نوعشان بستگی دارد و با پارامترهای `extra-stamina-delta-min` و `extra-stamina-delta-max` مشخص می‌شود. پس از کاهش `stamina`، کارگزار توان موثر `dash` را که وابسته به پارامترهای `dash-power-rate` و تلاش کنونی بازیکن (که مقداری بین `effort-min` و `effort-max`) است محاسبه می‌کند:

$$effectivedashpower = effort * dash - power - rate * power$$

مقدار `edp` و سوی بازیکن به صورت برداری به بردار شتاب بازیکن اضافه می‌شوند.

Basic Parameters server.conf		Parameters for heterogeneous Players player.conf		
Name	Value	Name	Value	Range
<i>minpower</i>	-100			
<i>maxpower</i>	100			
<i>stamina_max</i>	4000			
<i>stamina_inc_max</i>	45	<i>stamina_inc_max_delta_factor</i>	-100.0	25
		<i>player_speed_max_delta_min</i>	0.0	— 45
		<i>player_speed_max_delta_max</i>	0.2	
<i>extra_stamina<sup>a</sup></i>	0.0	<i>extra_stamina_delta_min</i>	0.0	0.0
		<i>extra_stamina_delta_max</i>	100.0	— 100.0
<i>dash_power_rate</i>	0.006	<i>dash_power_rate_delta_min</i>	0.0	0.006
		<i>dash_power_rate_delta_max</i>	0.002	— 0.008
<i>effort_min</i>	0.6	<i>effort_min_delta_factor</i>	-0.002	0.4
		<i>extra_stamina_delta_min</i>	0.0	— 0.6
		<i>extra_stamina_delta_max</i>	100.0	
<i>effort_max<sup>a</sup></i>	1.0	<i>effort_max_delta_factor</i>	-0.002	0.8
		<i>extra_stamina_delta_min</i>	0.0	— 1.0
		<i>extra_stamina_delta_max</i>	100.0	
<i>effort_dec_thr</i>	0.3			
<i>effort_dec</i>	0.005			
<i>effort_inc_thr</i>	0.6			
<i>effort_inc</i>	0.01			
<i>recover_dec_thr</i>	0.3			
<i>recover_dec</i>	0.002			
<i>recover_min</i>	0.5			
<i>player_accel_max</i>	1.0			
<i>player_speed_max</i>	1.0	<i>player_speed_max_delta_min</i>	0.0	1.0
		<i>player_speed_max_delta_max</i>	0.2	— 1.2
<i>player_rand</i>	0.1			
<i>wind_force</i>	0.0			
<i>wind_dir</i>	0.0			
<i>wind_rand</i>	0.0			
<i>player_decay</i>	0.4	<i>player_decay_delta_min</i>	0.0	0.4
		<i>player_decay_delta_max</i>	0.2	— 0.6

<sup>a</sup>Not in server.conf, but compiled into the server

شکل ۴: پارامترهای دستور dash

#### ۷.۴ مدل طاق (stamina)

سه متغیر مهم در stamina یک بازیکن تاثیر دارند: مقدار stamina، میزان بهبودیابی (recovery) و میزان تلاش بازیکن (effort). همانطور که گفتیم میزان stamina مربوط به یک بازیکن در هر دستور dash کاهش می‌یابد و در هر سیکل نیز اندک اندک به آن اضافه می‌شود. متغیر recovery مشخص می‌کند که میزان افزایش stamina در هر سیکل چقدر باشد و پارامتر effort نشان می‌دهد که ضربه به توپ چقدر تاثیرگذار است (جدول ۴). اصولاً در هر گام شبیه‌سازی که میزان stamina کمتر از یک مقدار آستانه‌ای است، مقدار effort و recovery آنقدر کاهش می‌یابد که توانی برای stamina بدست آید. در هر سیکل نیز که مقدار stamina یک بازیکن بالاتر از یک مقدار آستانه‌ای باشد، مقدار پارامتر effort به یک مقدار ماکزیمم افزایش می‌یابد. مقدار پارامتر recovery فقط در آغاز هر نیمه به مقدار ۱ تغییر می‌کند و در طول بازی افزایش پیدا نمی‌کند.

#### ۸.۴ مدل شوت (kick)

دستور Kick دو پارامتر می‌گیرد: قدرت شوت (که عددی بین minpower و maxpower است) و زاویه‌ای که بازیکن به توپ ضربه می‌زند (بین minmoment و maxmoment). جدول ۵ مقادیر کنونی این پارامترها را نشان می‌دهد.

زمانی که فرمان Kick به کارگزار می‌رسد، در صورتی که توپ قابل شوت کردن باشد، و بازیکن نیز در offside نباشد، شوت می‌شود. توپ در صورتی قابل شوت کردن است که فاصله آن با بازیکن کمتر از kick-able-margin باشد. بازیکنان غیریکسان می‌توانند مقادیر متفاوتی برای این پارامتر داشته باشند. به این نکته دقت داشته باشید که منظور از فاصله، فاصله مرکز بازیکن و مرکز توپ منحنای شعاع توپ و شعاع بازیکن است.

اولین پارامتری که برای یک شوت محاسبه می‌شود، قدرت موثر آن است:

$$effectivepower = kick - power * kick - power - rate$$

اگر توپ مستقیماً روبروی بازیکن قرار نداشته باشد، قدرت موثر شوت به میزان ضریبی از فاصله توپ با بازیکن کم می‌شود. در ضربه، هم جهت و هم فاصله مهم هستند.

اگر زاویه نسبی توپ ۰ درجه نسبت به روبروی بازیکن باشد، قدرت ضربه تغییر نمی‌کند. هرچه زاویه زیادتر می‌شود، قدرت ضربه کمتر می‌شود. بدترین حالت زمانی اتفاق می‌افتد که توپ پشت سر بازیکن قرار گرفته باشد که قدرت ضربه به میزان ۲۵ درصد کاهش می‌یابد.

پارامتر موثر دیگر، فاصله بازیکن با توپ است. اگر این فاصله صفر باشد، قدرت ضربه تغییر نمی‌کند و اگر توپ در حداکثر فاصله‌اش یعنی kickable-margin وجود داشته باشد، قدرت موثر توپ

Basic Parameters server.conf		Parameters for heterogeneous Players player.conf		
Name	Value	Name	Value	Range
<i>minpower</i>	-100			
<i>maxpower</i>	100			
<i>minmoment</i>	-180			
<i>maxmoment</i>	180			
<i>kickable_margin</i>	0.7	<i>kickable_margin_delta_min</i>	0.0	0.7
		<i>kickable_margin_delta_max</i>	0.2	— 0.9
<i>kick_power_rate</i>	0.027			
<i>kick_rand</i>	0.0	<i>kick_rand_delta_factor</i>	0.5	0.0
		<i>kickable_margin_delta_min</i>	0.0	— 0.1
		<i>kickable_margin_delta_max</i>	0.2	
<i>ball_size</i>	0.085			
<i>ball_decay</i>	0.94			
<i>ball_rand</i>	0.05			
<i>ball_speed_max</i>	2.7			
<i>ball_accel_max</i>	2.7			
<i>wind_force</i>	0.0			
<i>wind_dir</i>	0.0			
<i>wind_rand</i>	0.0			

شکل ۵: پارامترهای مؤثر در شوت

به میزان ۲۵ درصد کاهش می‌یابد. نهایتاً برای محاسبه قدرت مؤثر، به فرمول زیر می‌رسیم:

$$ep = ep * \left( 1 - .25 * \frac{dir - diff}{180} - .25 * \frac{dist - ball}{kickable - margin} \right)$$

قدرت ضربه نهایتاً منجر به محاسبه  $a_{n_i}$  می‌شود که به بردار حرکت سراسری توپ اضافه می‌شود. پارامتر دیگری به نام *kick-rand* در کارگزار وجود دارد که جهت تولید نویز در ضربه به کار می‌رود. برای بازیکنان غیریکسان، این پارامتر بستگی به *kick-rand-delta-factor* دارد که در فایل *player.conf* تعیین می‌شود.

## ۹.۴ مدل حرکت (move)

این دستور جهت جای‌دهی بازیکنان در یک محل مشخص در زمین در موقع آرایش تیم (before-kick-off و goal-r-n یا goal-l-n) و نه در زمان بازی مورد استفاده قرار می‌گیرد. در این حالت، بازیکنان می‌توانند به تعداد دلخواه این دستور را اجرا کنند و به هر جایی از زمین خود حرکت کنند.

کاربرد دوم این دستور، حرکت دروازه‌بان در داخل منطقه جریمه است که بعد از دریافت توپ توسط او انجام می‌گیرد. دروازه‌بان می‌تواند حداکثر goalie-max-oves بار قبل از ضربه زدن به توپ از دستور move استفاده کند.

## ۱۰.۴ مدل صحبت (say)

با استفاده از این دستور بازیکنان می‌توانند پیام‌هایی با حداکثر طول say-msg-size به بقیه اعلام کنند. پیام‌ها می‌توانند حاوی کارکترهای در محدوده  $[- <> ? * / ( ) - z A - 9 a - 0 -]$  باشند. پیام‌هایی که دورتر از فاصله audio-cut-dist از بازیکن گفته می‌شوند نمی‌توانند توسط بازیکن شنیده شوند. قدرت شنوایی بازیکنان فقط با محدودیت ظرفیت شنوایی همراه است. جدول ۵ پارامترهای مربوط به این دستور را نشان می‌دهد.

مقدار	پارامتر در server.conf
۵۱۲	say-msg-size
۵۰	audio-cut-dist
۲	hear-max
۱	hear-inc
۲	hear-decay

جدول ۵: پارامترهای دستور say

## ۱۱.۴ مدل چرخش (turn)

همانطور که دستور dash جهت شتاب‌دادن به بازیکن در جهت کنونی حرکتش مورد استفاده قرار می‌گیرد، دستور turn نیز جهت تغییر جهت بدن بازیکن به کار می‌رود. پارامتر این دستور، زاویه چرخش است که بین minmoment و maxmoment قرار دارد. اگر بازیکن در حال حرکت نباشد، بازیکن به میزان moment می‌چرخد؛ ولی وجود اینرسی، همواره از تغییر جهت یک بازیکن در حال



حرکت جلوگیری می‌کند. بدین ترتیب زاویه واقعی که یک بازیکن می‌چرخد به صورت زیر محاسبه می‌شود:

$$actual - angle = moment / (1.0 + inertia - moment * player - speed)$$

پارامتر inertia-moment با مقدار پیش فرض ۵/۰ در کارگزار مشخص می‌شود. بنابراین زمانی که بازیکن با سرعت ۱ در حال حرکت است، بیشترین چرخش موثری که می‌تواند انجام دهد، ۳۰ درجه در جهت راست یا چپ خواهد بود. به این نکته دقت کنید که از آنجا که یک بازیکن نمی‌تواند در یک سیکل هم dash کند و هم turn، حداکثر سرعت یک بازیکن در زمان چرخش player-decay . player-speed-max خواهد بود که به معنی ۶۰ درجه چرخش بیشینه برای بازیکن است.

برای بازیکنان ناهمسان، اینرسی پیش فرض inertia-value بعلاوه مقداری که حداقل inertia-moment-delta-factor . player-decay-delta-min و حداکثر player-decay-delta-max است می‌باشد.

## ۱۲.۴ مدل چرخش سر (TurnNeck)

با اجرای این دستور، بازیکن می‌تواند سرش را مستقل از بدنش بچرخاند. در واقع زاویه سری یک بازیکن زاویه دیدش را مشخص می‌کند. دستور turn-neck می‌تواند همزمان با اجرای دستورات turn، dash و kick اجرا شود. این دستور برخلاف دستور turn، تحت تاثیر اینرسی بازیکن قرار نمی‌گیرد؛ میزان چرخش سر باید بین minneckmoment و maxneckmoment قرار داشته باشد.

## ۵ کاربر مربی (The Coach Client)

### ۱.۵ معرفی

مشابه مسابقات فوتبال واقعی، وقتی که یک بازی در حال انجام است، از خارج زمین نمی‌توانیم آنرا دچار وقفه کنیم. بنابراین هیچ کسی به جز بازیکنان و داور مسابقه نمی‌تواند بر روی بازی تأثیر بگذارد و یا آنرا کنترل کند. به هر حال بسیار مفید خواهد بود که در زمانیکه در حال نوشتن برنامه‌های بازیکنان هستیم، بتوانیم کنترل بیشتری بر روی بازی داشته باشیم.

به عنوان مثال، این امکان وجود دارد که در زمانیکه یک عمل خاص، مثلاً dribbling را آزمایش می‌کنیم، بتوانیم به صورت خودکار از روش‌های یادگیری هوش مصنوعی استفاده کنیم. بنابراین یک کاربر که آنرا به عنوان مربی می‌شناسیم معرفی گردیده است. کاربر مربی امکانات زیر را دارد:

(۱) می‌تواند وضعیت بازی را کنترل کند.

(۲) می‌تواند پیغام‌های شنوایی را انتشار دهد. این پیغام می‌تواند شامل یک فرمان و یا اطلاعات مورد نیاز یک یا چند بازیکن باشد. معنای این پیغام و در نتیجه تفسیر آن بستگی به نظر کاربر دارد.

(۳) می‌تواند اطلاعاتی راجع به تمام اشیای قابل حرکت (توپ و بازیکن) را از کارگزار بگیرد.

### ۲.۵ مربیگری در حضور و یا عدم حضور داور

به صورت پیش‌گزینه، در کارگزار فوتبال یک داور داریم که در زمین مسابقه فعال می‌باشد. اگر بخواهیم یک کاربر مربی کنترل کاملی بر روی مسابقه داشته باشد، بایستی که از کارگزار فوتبال بخواهیم که پیمانانه<sup>۱</sup> داور را فعال کند. در این حالت وظیفه مربی است که وضعیت بازی را کنترل کند و وضعیت بازی را مطابق قوانین خاص خودش تغییر دهد.

همچنین این امکان وجود دارد که هم پیمانانه داور و هم مربی بر روی بازی کنترل داشته باشد. در این حالت، مربی به عنوان مثال می‌تواند یک session یادگیری را کنترل کند و یا به یک بازیکن اطلاعاتی راجع به راندمان فعلی آن بازیکن بدهد و آن بازیکن بتواند با استفاده از این اطلاعات، رفتار خود را تنظیم کند.

در ابتدای مسابقه، کارگزار فوتبال را بایستی مطلع سازیم که از کدامیک از این وضعیت‌ها می‌خواهیم استفاده کنیم. اگر کارگزار فوتبال را با استفاده از گزینه ی coach- در خط فرمان اجرا کنیم،

<sup>1</sup>Module

آنگاه پیمانهای داور غیر فعال می‌شود و ما فقط کاربر مربی خواهیم داشت و اگر آنرا با گزینه‌ی coach-w-referee اجرا کنیم، آنگاه هم پیمانهای داور و هم کاربر مربی فعال خواهند بود. در هر دو وضعیت فوق، کارگزار فوتبال یک درگاه UDP (به صورت پیش‌گزیده: درگاه ۶۰۰۱) را آماده می‌کند و کاربر مربی از طریق آن به کارگزار فوتبال متصل می‌شود.

### ۳.۵ مربیگری در یک مسابقه‌ی واقعی

تا قبل از سال ۱۹۹۸ این امکان وجود نداشت که کاربر مربی در طول مسابقات واقعی به کارگزار فوتبال متصل شود. اما از سال ۱۹۹۸ این امکان وجود دارد که کاربر مربی در مسابقات واقعی شرکت کند. ایده‌ای که پشت سر این کار است این است که یک مربی می‌تواند بازی را مشاهده و تحلیل کند و اطلاعات استراتژیکی را به بازیکنان خود بدهد. امکاناتی که یک مربی در طول یک بازی واقعی می‌تواند داشته باشد، محدود به دو مورد زیر است:

(۱) دریافت اطلاعاتی راجع به مکان توپ و بازیکنان از کارگزار.

(۲) انتشار و دریافت پیغام‌های شنوایی.

### ۴.۵ فرمان‌های مربی

در طول یک مسابقه واقعی از فرامینی که با علامت  $\otimes$  مشخص شده است، نمی‌توانیم استفاده کنیم:

$\otimes$  change\_mode\_PLAY\_MODE

با استفاده از این فرمان، می‌توان وضعیت بازی را به یکی از حالت‌های before\_kick\_off، play\_on، kick\_off\_Side، kick\_in\_Side، goal\_kick\_Side، free\_kick\_Side، corner\_kick\_Side و time\_over تغییر داد. در اغلب این وضعیت‌ها، مکان توپ ثابت باقی می‌ماند و فقط وضعیت بازی عوض می‌شود.

$\otimes$  move OBJECT X Y [Vdir]

این فرمان مکان OBJECT که می‌تواند یک توپ و یا بازیکن باشد را به مکان (X, Y) جابجا می‌کند و اگر Vdir را مشخص کرده باشیم، جهت آنرا به Vdir تغییر می‌دهد.

$\otimes$  check\_in\_ball

راجع به موقعیت توپ، از کارگزار فوتبال سوال می‌کند. چهار وضعیت می‌تواند بروز کند:

- (۱) in\_field: توپ در داخل محدوده زمین است.
- (۲) goal\_l: توپ در داخل محدوده دروازه سمت چپ زمین است.
- (۳) goal\_r: توپ در داخل محدوده دروازه سمت راست زمین است.
- (۴) out\_of\_field: توپ در جای دیگری غیر از مکان‌های فوق است.

look

از کارگزار، اطلاعاتی راجع به موقعیت اشیای زیر بر روی زمین می‌گیرد:

(۱) دروازه‌های سمت راست و سمت چپ

(۲) توپ

(۳) تمام بازیکنان فعال

say MESSAGE

پیغام MESSAGE را به تمام کاربران انتشار می‌دهد. فرمت و غالب پیغام MESSAGE همان گونه‌ای است که برای کاربران بازیکن استفاده می‌شود.

ear MODE

امکان ارسال اطلاعات شنوایی به کاربر مربی را فعال و یا غیر فعال می‌سازد. MODE می‌تواند یکی از مقادیر on و یا off را داشته باشد. اگر ear on باشد، آنگاه کارگزار کلیه اطلاعات شنوایی را که به بازیکنان می‌فرستد، به کاربر مربی هم می‌فرستد و اگر ear off باشد، آنگاه کارگزار اطلاعات شنوایی را به کاربر مربی نمی‌فرستد.

## مراجع

- [1] Robocup Soccer Simulation Committee, *Rules and Regulations for Soccer Simulation League*, <http://www.robocup.org>, accessed: Dec. 2002.