

گزارش فنی تیم GRCG

صالح خزاعی , سامان پرهمت

دبیرستان دکتر هشترودی مشهد

Saleh.khazaei@gmail.com

چکیده

این بخش دید کلی در مورد تیم شبیه سازی دویعدی GRCG است.

ما در سال‌های اخیر بر روی بیس‌های Agent2D و UVA_Trilearn کد نویسی کرده بودیم و تصمیم گرفتیم برای مسابقات IranOpen 2011 با نوشتن بیس GRCG گامی فراتر در این عرصه برداریم.

تلاش ما در افزایش سطح کیفیت الگوریتم‌های هوش مصنوعی (پاس دادن , شوت کردن و ...) در رابطه با شبیه سازی دویعدی فوتبال بوده است.

مقدمه

در تیم ما سعی کردیم که با استفاده از الگوریتم‌های هوش مصنوعی و تلاش در ارائه الگوریتم‌های بهتر و پیاده سازی آنها در تیم , استفاده از استراتژی‌های مختلف و تصمیم گیری در شرایط مختلف تیمی قوی بوجود بیاوریم.

در بیس GRCG از قسمت‌هایی از ریاضیات بیس Agent2D و توابعی برای کاهش نویز سرور از قسمت لوکالیزیشن بیس Agent2D (estimateSelfFace , getNearestMarker , localizePlayer) , set_entry , localizeBallRelative , getFaceDirByMarkers , getFaceDirByLines , getMovableObjInfo , getStaticObjInfo) استفاده شده است.

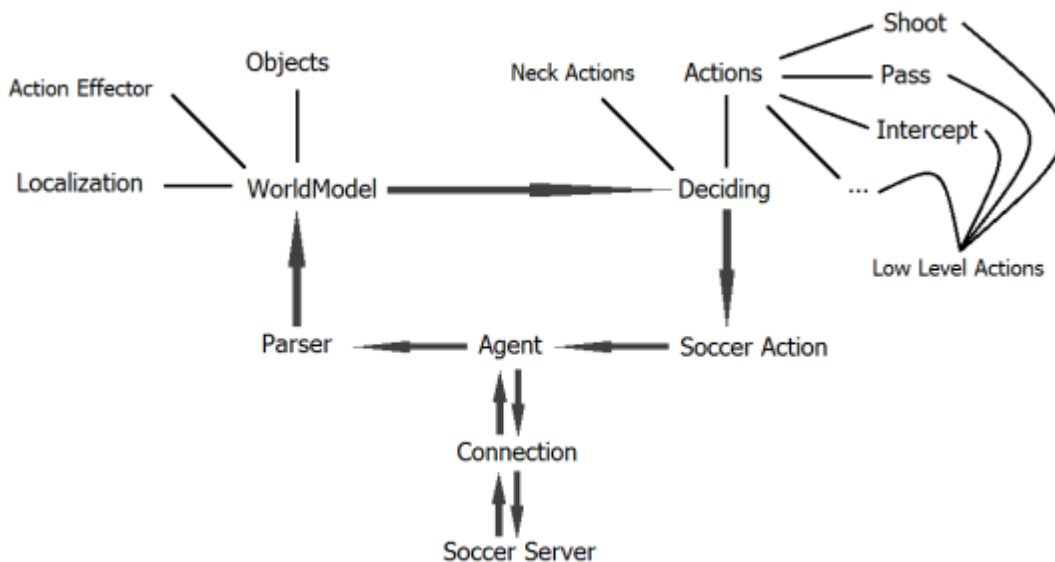
در مورد شبیه سازی دو بعدی فوتبال

مسابقات جهانی روبات‌ها تلاشی در جهت ارتقا الگوریتم‌های هوش مصنوعی و تحقیقات مربوط با روبات‌های هوشمند می باشد که در قالب یک مسئله استاندارد با قابلیت آزمایش و انعطاف بسیار بالا مطرح شده است. هدف نهایی و بلند مدت روبات‌ها عموماً به صورت زیر بیان می شود:

"در نیمه قرن ۲۱, سال ۲۰۵۰ میلادی, یک تیم از روبات‌های هوشمند یک مسابقه رسمی فوتبال را با رعایت کلیه قوانین فیفا, در مقابل فاتح آخرین قهرمان جام جهانی فوتبال انسانی پیروز خواهند شد."

هدف فوق یکی از بزرگترین مبارزات علمی است که در تاریخ بشری مطرح شده است. [1] در راستای اجرای این هدف , چندین لیگ رباتیک بوجود آمده است که لیگ شبیه سازی دو بعدی فوتبال نیز یکی از این لیگها است.

دید کلی در مورد بیس GRCG



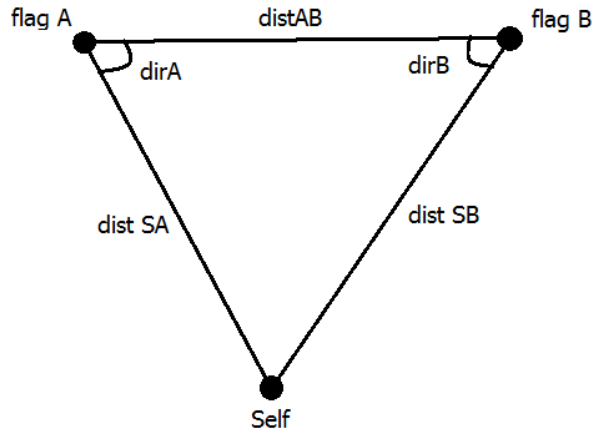
شکل 1) روند کلی انجام عملیات ها در بیس GRCG

لوکالیزیشن (Localization)

1- پیدا کردن مختصات خود بازیکن (Self Localization)

در این بیس با استفاده از الگوریتم پیدا کردن موقعیت با استفاده از دو پرچم و استفاده از تمامی پرچم ها با در نظر گرفتن شرایط , موقعیت خود را با تقریب خوبی بدست می آوریم.

برای انجام این کار ابتدا بررسی می کنیم که چگونه می توان با استفاده از دو پرچم موقعیت خود را بدست آورد.



شکل ۲) لوکالیز با استفاده از دو پرچم

شکل بالا را در نظر می گیریم. در شکل بالا مختصات پرچم ها و فاصله پرچم ها تا بازیکن را داریم پس فاصله بین پرچم ها را نیز داریم. برای پیدا کردن مختصات نقطه Self می توان بردار AB (بردار (flagB - flagA)) را به اندازه dirB چرخاند سپس طول بردار را به اندازه distSB قرار می دهیم .

برای بدست آوردن dirB می توانیم از فرمول زیر استفاده کنیم

$$d = \sqrt{A^2 B^2 - 2 \cdot A \cdot B \cdot \cos \theta}$$

که با جایگذاری مقادیر فرمول به صورت زیر در می آید.

$$distSA = \sqrt{distAB^2 distSB^2 - 2 \cdot distAB \cdot distSB \cdot \cos dirB}$$

$$\rightarrow distSA^2 = distAB^2 distSB^2 - 2 \cdot distAB \cdot distSB \cdot \cos dirB$$

$$\rightarrow distSA^2 - distAB^2 distSB^2 = -2 \cdot distAB \cdot distSB \cdot \cos dirB$$

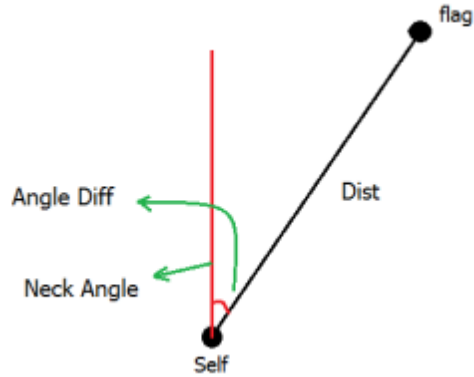
$$\rightarrow \frac{distSA^2 - distAB^2 distSB^2}{-2 \cdot distAB \cdot distSB} = \cos dirB$$

$$\rightarrow dirB = \text{acos}\left(\frac{distSA^2 - distAB^2 distSB^2}{-2 \cdot distAB \cdot distSB}\right)$$

سپس برای پیدا کردن مختصات Self , مختصات flagB را با بردار بدست آمده جمع می کنیم تا مختصات Self را بدست آوریم. همین کار را برای flagA انجام می دهیم. سپس میانگین دو مختصات را برای مختصات Self در نظر می گیریم (برای کاهش نویز سرور).

2- پیدا کردن زاویه گردن

با استفاده از این الگوریتم برای تمامی پرچم هایی که آنها را می بینیم می توانیم زاویه گردن خود را با تقریب بسیار خوبی بدست آوریم.



شکل ۳) محاسبه زاویه گردن با استفاده از مختصات خود، پرچم‌ها و اختلاف زاویه گردن و پرچم

سرور Angle Diff را به بازیکن ارسال می‌کند. با استفاده از این زاویه و مکان خود (در قسمت قبل توضیح داده شد.) و مکان پرچم‌ها می‌توان با الگوریتم زیر زاویه گردن را بدست آورد.

```
Double getNeckAngle ( Position flag , Position self , Angle angle_diff )
```

```
{
```

```
Double dir = (flag – self).getAngle();
```

```
Return dir + angle_diff;
```

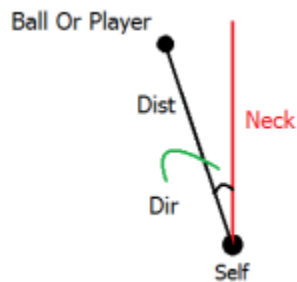
```
}
```

برای کاهش نویز سرور این کار را برای تمام پرچم‌هایی که در محوطه دید هستند انجام می‌دهیم و میانگین آنها را زاویه گردن خود قرار می‌دهیم.

3- پیدا کردن زاویه بدن

سرور برای بازیکنان اختلاف زاویه بدن و گردن را می‌فرستد. برای پیدا کردن زاویه بدن کافیسیت این زاویه را از زاویه گردن خود که در قسمت قبل محاسبه کردیم کم کنیم.

4- محاسبه مختصات توپ و بازیکنان دیگر



شکل ۴) محاسبه مختصات توپ و بازیکنان دیگر

محاسبه مختصات توپ یا بازیکنان دیگر با استفاده از زاویه گردن و اختلاف زاویه (این مقدار از طرف سرور ارسال می شود) از طریق الگوریتم زیر امکان پذیر است.

$$\text{ballX} = \text{dist} * \cos_deg (\text{dir} + \text{neck});$$

$$\text{ballY} = \text{dist} * \sin_deg (\text{dir} + \text{neck});$$

ضربه زدن به توپ (Kick)

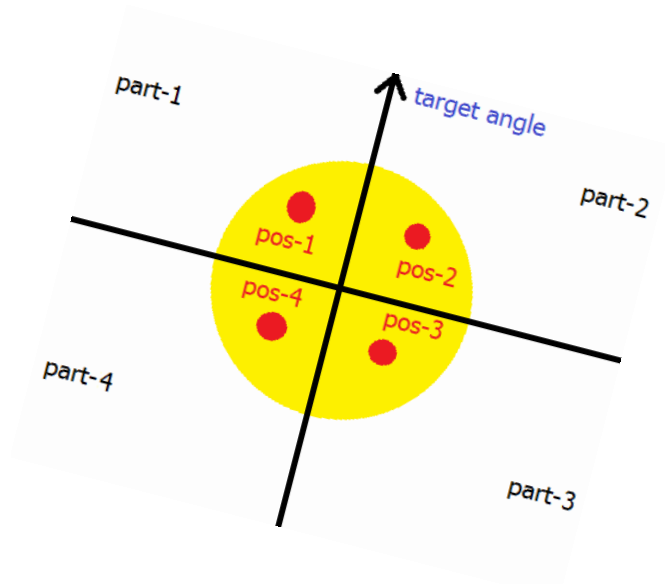
برای ضربه زدن به توپ و رساندن سرعت توپ به سرعت دلخواه با استفاده از چند ضربه (multi - kick) می توان این عمل را انجام داد زیرا رساندن سرعت توپ به سرعت زیاد با یک ضربه ممکن نیست. برای این کار ابتدا بررسی می کنیم که آیا می توانیم توپ را با یک ضربه به سرعت دلخواه برسانیم !
برای اینکار از فرمول های زیر استفاده می کنیم.

$$ep = ep \cdot \left(1 - 0.25 \cdot \frac{\text{dir_diff}}{180^\circ} - 0.25 \cdot \frac{\text{dist_ball}}{\text{kickable_margin}} \right)$$

$$\text{MaxKickSpeed} = ep \cdot \text{MaxKickPower}$$

چنانچه سرعت مورد نظر ما از MaxKickSpeed کمتر بود می توانیم با یک ضربه سرعت توپ را به سرعت دلخواه برسانیم.

چنانچه نمی توانستیم توپ را با یک ضربه به سرعت دلخواه برسانیم توپ را دور بازیکن می چرخانیم. برای چرخاندن توپ دور بازیکن از الگوریتم زیر استفاده می کنیم.



شکل ۵) نحوه منطقه بندی برای چرخاندن توپ دور بازیکن

Method MultiKick ()

```
{  
If ( ball is in part-1 ) kick ball to pos-2  
If ( ball is in part-2 ) kick ball to pos-3  
If ( ball is in part-3 ) kick ball to pos-4  
If ( ball is in part-4 ) kick ball to pos-1  
}
```

با استفاده از الگوریتم زیر می توان زمان زدن ضربه نهایی را محاسبه کرد.

Method FinalKick()

```
{  
maxKickSpeed = ep . maxKickPower;  
Position target_rel = ( target_point – ball_point );  
If ( fabs ( target_rel.getAngle() – vel.getAngle() ) < 20.0 && first_speed <  
maxKickSpeed )  
    Kick to target_point  
Else  
    MultiKick();  
}
```

در آینده

تیم GRCG سعی بر این دارد که برای مسابقات ایران اوپن ۲۰۱۱ با نوشتن اکشن هایی مانند یارگیری , موضع گیری , دریبل بدنی و ... تیمی سریع و قدرتمند در دفاع و حمله تشکیل دهد. توضیح مختصری در مورد این اکشن ها در زیر آمده است.

1- یارگیری (Mark)

هنگامی که بازیکن حریف صاحب توپ است به صورتی بین بازیکن موردنظر و بازیکن صاحب توپ قرار می گیریم که مانع رسیدن پاس از سوی بازیکن صاحب توپ به بازیکن مورد نظر شویم. این اکشن در دو حالت اجرا می شود.

PlayOn Mark -1-1

در این حالت بازیکنانی از حریف را mark می کنیم که توانایی دریافت پاس و ایجاد موقعیتی خطرناک را داشته باشند. بازیکنان خودی که این عمل را انجام می دهند نباید بازیکنان دفاع یا حمله باشند زیرا استفاده از بازیکنان دفاع خود باعث ایجاد موقعیتی خطرناک می شود و استفاده از بازیکنان حمله نیز باعث کاهش سرعت ضد حمله و حتی ناموفق شدن ضد حمله شود.

DeadBall Mark -1-2

در این حالت تمامی بازیکنان حریف را (غیر از بازیکنان دفاع) mark می کنیم. این کار را با بازیکنان دفاع انجام نمی دهیم زیرا بازیکنی که ضربه را می زند می تواند با پاس در عمق , توپ را با موفقیت به بازیکنان حریف دیگر برساند. با ترکیب این دو حالت mark در شرایط مناسب می توانیم سیستم دفاعی قوی ای تشکیل دهیم.



شکل ۶) نحوه mark کردن (بازیکن ۲ زرد , بازیکن ۲ آبی را mark کرده است)

2- دربیبل بدنی (Body Dribble)

برای انجام این عمل می توانیم با انتخاب مقصدی مناسب برای دربیبل و انجام مراحل زیر دربیلی قوی را تشکیل دهیم.

- 1- چنانچه نیازی به تغییر مکان توپ بود (بازیکنی از حریف در موقعیتی قرار داشت که می توانست توپ را از بازیکن بگیرد) توپ را به امن ترین منطقه از بدن انتقال می دهیم.
- 2- اگر برای حرکت به سمت نقطه مقصد نیاز به چرخش زاویه بدن داشتیم:
 - ۲-۱- اگر توپ در سیکل بعد از محوطه بدن بازیکن خارج می شد سرعت توپ را کاهش می دهیم.
 - ۲-۲- در غیر اینصورت بدن خود را به سمت مقصد می چرخانیم.
- 3- تا موقعی که توپ از بدن خارج نمی شود dash می کنیم.
- 4- توپ را به سمت مقصد به صورتی kick می کنیم که در سیکل بعد در محوطه بدن بازیکن باشد.

منابع

[1] نیما کاویانی , مصطفی رفایی , روبوسینا از صفر , ۱۳۸۴

[2] Server Manual