



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عنوان

روش فیلتر ذره در موقعیت یابی ربات های انسان نما در
زمین فوتبال

نگارش

دانیال علی حسینی

استاد راهنما

دکتر رضا صفابخش

خرداد ۱۳۹۵

به نام خدا



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشگاه صنعتی امیرکبیر

(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

عنوان

روش فیلتر ذره در موقعیت‌یابی ربات‌های انسان‌نما در

زمین فوتبال

نگارش

دانیال علی حسینی

استاد راهنما

دکتر رضا صفا بخش

درس

روش تحقیق و گزارش نویسی

خرداد ۱۳۹۵

تقدیر و تشکر

در ابتدا لازم می‌دانم از دکتر صفابخش بابت تمام زحمات و انتقال نکات مهم و کلیدی در راستای درس، دکتر همایون‌پور، جناب آقای خادمیان و تمام افرادی که در تهیه این گزارش کمک و مساعدت نموده‌اند، تشکر نمایم.

چکیده

همه ما از برنامه جهانی سال ۲۰۵۰ ربوکاپ باخبر هستیم؛ ربات‌ها تا به سال ۲۰۵۰، باید به حدی رسیده باشند که بتوانند در یک مسابقه فوتبال، تیم انسان‌ها را شکست دهند. از این رو ربات‌ها باید بسیاری از توانایی‌های انسان‌ها را در خود داشته باشند؛ یکی از این توانایی‌ها استفاده از محیط برای تخمین موقعیت خود نسبت به بقیه اشیای محیط است.

در این گزارش سعی شده است تا روشی برای حل مسئله موقعیت‌یابی ارائه شود. ربات مورد نظر، ربات انسان‌نما^۱ در زمین فوتبال بوده و به همین دلیل روش‌هایی بیان خواهد شد که قابل استفاده در این کلاس از ربات‌ها باشد؛ در واقع هدف، ارائه روشی است که با استفاده از آن بتوان در هر زمانی از مسابقه موقعیت ربات را در زمین نسبت به نقطه‌ای قراردادی حدس زد. روشی که برای حل این مسئله در این گزارش ارائه شده است روش فیلتر ذره^۲ است. این فیلتر از جهات بسیاری قدرتمند بوده و توانایی حل مسائل با پیچیدگی بالا را دارد.

کلمات کلیدی: فیلتر ذره، موقعیت‌یابی، ربات انسان‌نما، مونت کارلو^۳، فوتبال، ربوکاپ

^۱ Humanoid

^۲ Particle filter

^۳ Monte Carlo

فهرست مطالب

عنوان	صفحه
۱ مقدمه	۷
۲ شرایط مسئله	۸
۲-۱ زمین بازی	۸
۲-۲ برنامه ربات	۸
۲-۲-۱ بینایی	۹
۲-۲-۲ تصمیم‌گیری	۱۰
۲-۲-۳ اجرایی	۱۰
۳ روشهای حل مسئله	۱۱
۳-۱ ناوبری کورکورانه	۱۱
۳-۲ فیلتر کامن و فیلتر ذره	۱۱
۳-۲-۱ x متغیر هدف	۱۲
۳-۲-۲ y مشاهدات	۱۲
۳-۲-۳ مدل سیستم f	۱۲
۳-۲-۴ g مدل اندازه‌گیری	۱۳
۳-۲-۵ خروجی فیلتر	۱۳
۳-۳ فیلتر کالمن	۱۴
۳-۴ فیلتر ذره	۱۴
۴ روش فیلتر ذره	۱۷
۵ نحوه استفاده از روش فیلتر ذره	۲۰

۵-۱	انتخاب توزیع اهمیت مناسب.....	۲۰
۵-۱-۱	روش فیلتر ذره بوتاسترپ	۲۰
۵-۲	مدلسازی	۲۱
۵-۲-۱	مدل سیستم	۲۲
۵-۲-۲	مدل اندازه گیری	۲۲
۵-۳	استفاده از خروجی فیلتر ذره	۲۳
۶	نتیجه گیری	۲۴
	مراجع و منابع مفید	۲۵

فهرست اشکال

<u>عنوان</u>	<u>صفحه</u>
شکل ۱. شکل زمین و ابعاد آن	۸
شکل ۲. برنامه ربات	۸
شکل ۳. جایگاه بخش موقعیت‌یابی در مقایسه با بقیه واحدها در برنامه ربات	۹
شکل ۴. نمونه‌های از تابع $p(x_t y_{0:t})$ که به عنوان خروجی ارائه میشود.	۱۳
شکل ۵. استفاده از مونت کارلو برای شبیه‌سازی تابع احتمال $p(x_t y_{0:t})$ که در شکل ۴ در صفحه ۱۳	۱۳
نشان داده شد	۱۵
شکل ۶. نحوه کلی عملکرد فیلتر کالمن و فیلتر ذره؛ فرآیند مارکوفی نوع اول.	۱۶
شکل ۷. روند کلی نحوه عملکرد الگوریتم فیلتر ذره بوتاسترپ	۲۱

۱ مقدمه

از آن جا که هر روزه ربات ها کاربرد بیشتری در زندگی ما پیدا می کنند، مسئله موقعیت یابی جایگاهی ویژه یافته است. با ذکر یک مثال، مسئله را واضح تر خواهیم کرد. تصور کنید در یک زمین فوتبال تاریک هستید و گاهی توپ یا دروازه را می بینید؛ با دیدنشان، می توانید فاصله خود را نسبت به آن ها بدست آورید. حال فرض کنید در حین بازی کردن، توپ را گم کرده اید و باید به دنبال توپ بگردید؛ می دانید توپ در فاصله ۲ متری از مرکز زمین است. به این دلیل که از موقعیت خود در زمین خبر ندارید، موقعیت توپ در زمین هیچگونه کمکی به نحوه جست و جوی شما نخواهد کرد و باید به دور خود بچرخید تا توپ را در اطراف خود پیدا کنید. بنابراین نمی توانید استراتژی قوی برای بازیتان طراحی کنید.

هدف این گزارش بررسی نحوه حل مسئله موقعیت یابی در ربات های انسان نما می باشد که منظور از عبارت مسئله موقعیت یابی، تخمین موقعیت ربات نسبت به بقیه اشیای محیط است. به منظور حل این مسئله، چندین روش بیان خواهد شد که بهترین آن ها روش فیلتر ذره بوده و تمرکز گزارش بر روی این روش می باشد.

روند مطالب به شرح زیر است:

۱. توضیح شرایط حاکم بر مسئله

۲. ارائه چند روش برای حل مسئله

۳. ارائه روش فیلتر ذره به عنوان یک راه حل مناسب

۴. نحوه استفاده از روش فیلتر ذره

لازم به ذکر است که تمامی روش های ارائه شده در این گزارش تقریبی بوده و تنها تخمینی از جواب مسئله

را به عنوان خروجی در اختیار ما می گذارند.

۲ شرایط مسئله

در این فصل، توضیحاتی مختصر راجع به محیط بازی ربات و نحوه کار کردن آن مطرح خواهد شد تا شرایط ربات در هنگام بازی بهتر درک شود.

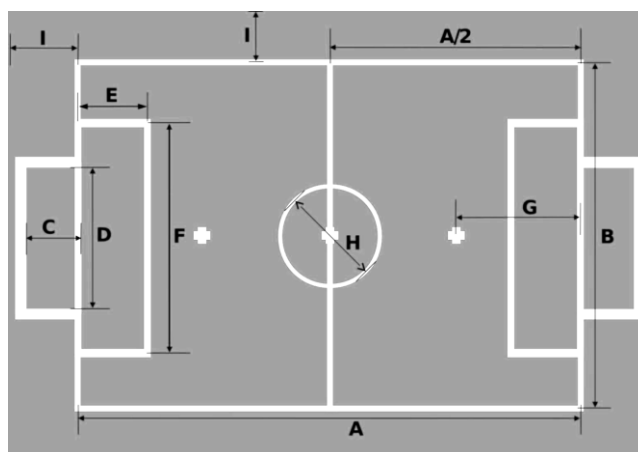
۲-۱ زمین بازی

همان‌طور که در شکل ۱ مشخص شده است، زمین بازی مانند زمین بازی فوتبال بوده ولی با مقیاسی

جدول ۱. ابعاد زمین

علامت	معنی علامت	کلاس اندازه نوجوان
A	طول زمین	۹۰۰
B	عرض زمین	۶۰۰
C	عمق دروازه	۶۰
D	عرض دروازه	۲۶۰
E	طول محوطه جریمه	۱۰۰
F	عرض محوطه جریمه	۵۰۰
G	فاصله نقطه پنالتی	۲۱۰
H	قطر دایره میانی	۱۵۰
I	حداقل حاشیه دور زمین	۷۰

کوچک‌تر. ابعاد زمین در جدول ۱ آورده شده است.



شکل ۱. شکل زمین و ابعاد آن

لازم به ذکر است که ربات زمین بازی را به خوبی می‌شناسد؛ برای مثال می‌داند کرنر سمت چپ در موقعیت

(۱۰۰، ۱۰۰) نسبت به نقطه وسط زمین قرار دارد. این

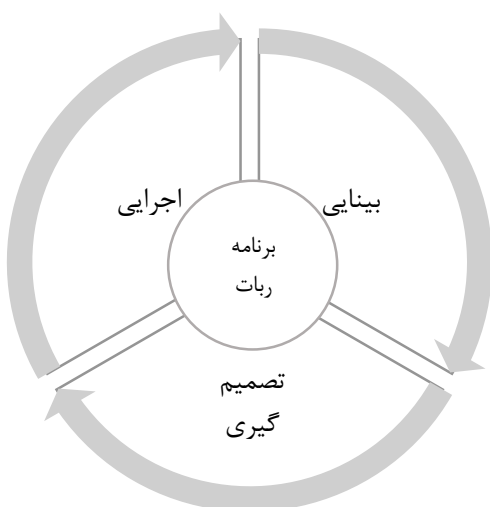
اطلاعات کمک زیادی در روش فیلتر ذره خواهد کرد.

۲-۲ برنامه ربات

برنامه ربات در یک حلقه تکرار کلی قرار دارد. الگوی

برنامه ربات در شکل ۲ نشان داده شده است. هر دور شامل

سه بخش کلی است:



شکل ۲. برنامه ربات

- بینایی
- تصمیم گیری
- اجرایی

۲-۲-۱ بینایی

این بخش از تعدادی بخش دیگر تشکیل شده است:

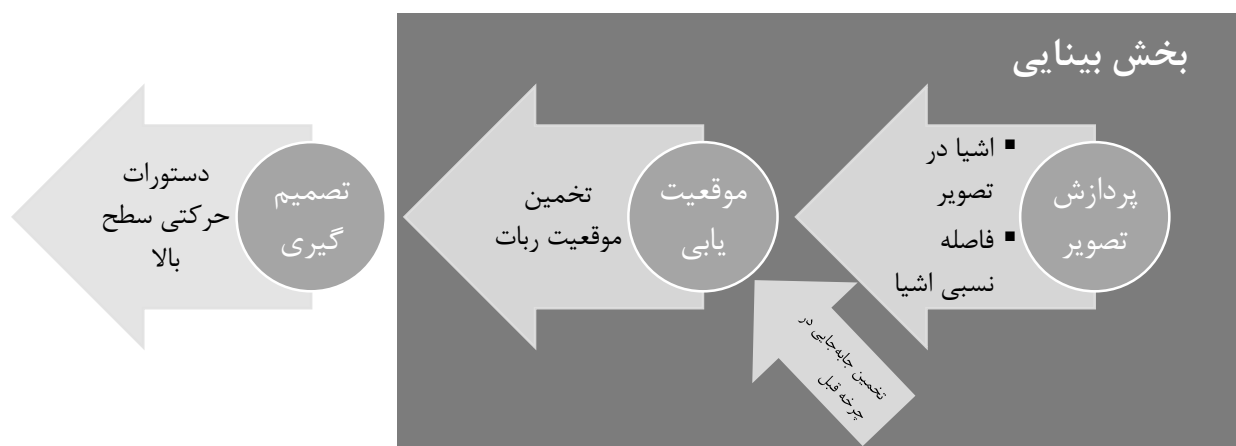
- پردازش تصویر
- موقعیت یابی

۲-۲-۱-۱ پردازش تصویر

در این بخش ابتدا عکسی از محیط گرفته شده و تمام اشیای قابل شناسایی توسط ربات شناسایی می‌شوند. خروجی این واحد نوع اشیا و فاصله نسبی آن‌ها بوده که در اختیار بقیه واحدها قرار می‌گیرد.

۲-۲-۱-۲ موقعیت یابی

همان‌طور که از نام این بخش برمی‌آید، در این بخش موقعیت ربات نسبت به نقطه‌ای قراردادی (برای مثال نقطه وسط زمین) تخمین زده می‌شود. نوع اشیا، فاصله نسبی آن‌ها و میزان جابه‌جایی در دور قبلی به عنوان ورودی در اختیار این واحد قرار گرفته تا تخمینی از موقعیت ربات را به عنوان خروجی در اختیار سایر واحدها قرار دهد.



شکل ۳. جایگاه بخش موقعیت یابی در مقایسه با بقیه واحدها در برنامه ربات

۲-۲-۲ تصمیم‌گیری

این واحد دستورات حرکتی سطح بالا را صادر می‌کند؛ برای مثال خروجی این واحد می‌تواند سه سرعت در راستای محور x ، y و سرعت دورانی باشد که به واحد اجرایی ارسال می‌شود. اشیای دیده شده در آخرین عکس و فاصله نسبی آن‌ها و تخمین موقعیت ربات به عنوان ورودی در اختیار این واحد قرار می‌گیرد.

۲-۲-۳ اجرایی

این بخش بیشتر مربوط به محاسبات مکانیکی بوده و دستورات صادر شده توسط واحد تصمیم‌گیری را اجرا می‌کند.

در این بخش سعی شد توضیحاتی راجع به محیط بازی ربات و نحوه کار کردن آن ارائه شود تا بیشتر با وضعیت و شرایط ربات در هنگام بازی آشنا شوید.

۳ روش‌های حل مسئله

برای حل مسئله مورد نظر، روش‌های متفاوتی ارائه شده است که بعضی بسیار ساده و بعضی بسیار پیچیده-اند. در این فصل، تعدادی از روش‌های معروف موجود شامل روش ناوبری کورکورانه^۴، فیلتر کالمن^۵ و فیلتر ذره مطرح خواهند شد.

۱-۳ ناوبری کورکورانه

در این روش تمام جابه‌جایی‌های بدست‌آمده از لحظه اولیه تا به لحظه کنونی با هم جمع شده و نتیجه‌ی جمع به عنوان خروجی در اختیار سایر واحدها قرار می‌گیرد. به دلیل این که هر جابه‌جایی به تنهایی و جدا از بقیه جابه‌جایی‌ها در زمان‌های دیگر، دارای مقداری خطا می‌باشد، در اثر جمع خطاها با یکدیگر، جواب ارائه شده مقدار زیادی خطا خواهد داشت. البته باید این روش پیاده‌سازی شده و آزمایش شود؛ چرا که ممکن است برای بعضی کاربردها مناسب باشد. از آن‌جا که خطای اندازه‌گیری میزان جابه‌جایی ربات، بزرگ می‌باشد، استفاده از این روش برای حل مسئله مورد بحث در گزارش مناسب نیست [۱].

۲-۳ فیلتر کالمن و فیلتر ذره

فیلتر کالمن و فیلتر ذره سعی دارند تا مسئله را به صورت آماری حل کنند. خروجی این دو الگوریتم یک تابع احتمال می‌باشد که معرف نقطه حضور ربات در زمین است؛ هر نقطه‌ای که احتمال بیشتری در تابع احتمال داشته باشد، احتمال حضور ربات در آن نقطه بیشتر خواهد بود.

برای اینکه بهتر بتوان این فیلترها را بررسی کرد، نیاز است تا با پارامترهای مورد استفاده در این فیلترها آشنا شد.

^۴ Dead reckoning

^۵ Kalman filter

۳-۲-۱ x متغیر هدف

x متغیر هدفی است که می‌خواهید مقدار آن را در لحظه t حدس بزنید. در مسئله مورد بحث، x به صورت

$\begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ خواهد بود؛ x و y موقعیت ربات را در زمین مشخص خواهند کرد و θ زاویه آن را در زمین مشخص می‌کند [۲، ۳].

۳-۲-۲ y مشاهدات

مشاهده می‌تواند هر آن چیزی که توضیحی درباره x بدهد باشد. در مورد ربات‌های انسان‌نما، فاصله نسبی اشیا و نوع آن‌ها می‌تواند به عنوان مشاهده مورد استفاده قرار گیرد؛ اما چطور این داده‌ها می‌توانند اطلاعاتی راجع به x بدهند؟ برای روشن شدن موضوع، فرض کنید که نوع شیء نقطه وسط زمین و فاصله نسبی آن ۲ متر تشخیص داده شده است؛ با داشتن این مشاهده می‌توانید بگویید در دایره‌ای به شعاع ۲ متر و به مرکز نقطه وسط زمین حضور دارید [۲، ۳].

۳-۲-۳ f مدل سیستم

مدل سیستم وظیفه حدس وضعیت x در لحظه بعدی را دارد؛ به طور دقیق‌تر، f با گرفتن مقدار x_{t-1} (مقدار x در لحظه $t-1$) مقدار x_t را حدس می‌زند. نمایش ریاضی این مدل به صورت $f(x_t|x_{t-1})$ می‌باشد؛ همان‌طور که از شکل تابع مشخص است، این مدل به صورت یک تابع احتمال شرطی تعریف شده است [۲، ۳].

اگر بخواهید f را برای مسئله مورد بحث تعریف کنید، می‌توان به شکل غیر دقیق تعریف زیر را ارائه کرد:

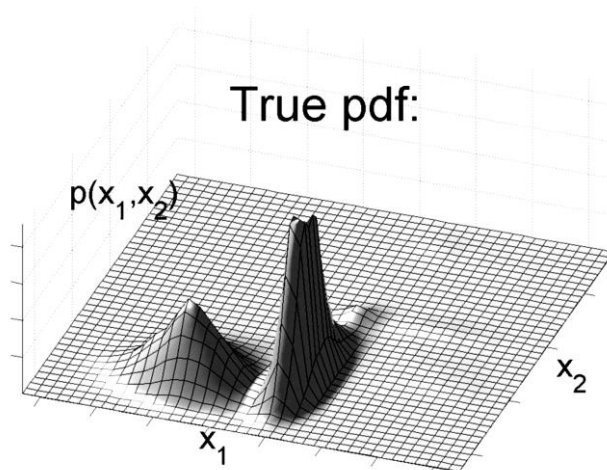
"تابع f با گرفتن میزان جابه‌جایی بدست‌آمده در چرخه قبل و مقدار x_{t-1} ، $x_t = \begin{bmatrix} x \\ y \\ \theta \end{bmatrix}$ را حدس می‌زند".

۳-۲-۴ g مدل اندازه‌گیری

این مدل با دریافت مقدار x_t بیان می‌کند که اگر در موقعیت x_t باشید، چه مشاهده‌ای بدست خواهد آمد؛ برای مثال اگر ربات روی دایره وسط زمین باشد و x_t ای مناسب با موقعیت ربات به g دهید، این مدل نقطه وسط زمین و فاصله نسبی ۱ متر را به عنوان خروجی به ما می‌دهد؛ به شکل دقیق‌تر این خروجی را به صورت یک تابع احتمالی بیان می‌کند. نمایش ریاضی این مدل به صورت $g(y_t|x_t)$ می‌باشد که باز هم به صورت یک تابع احتمال شرطی تعریف شده است [۲, ۳].

۳-۲-۵ خروجی فیلتر

همان‌طور که توضیح داده شد، خروجی این دو فیلتر یک تابع احتمال است که معرف احتمال حضور ربات در نقاط مختلف زمین است. خروجی فیلتر ذره به زبان ریاضی به صورت تابعی به شکل $p(x_t|y_{0:t})$ تعریف می‌شود؛ به عبارت دیگر فیلتر سعی دارد به ازای دریافت تمام مشاهدات به دست آمده از لحظه صفر تا t مقدار x را در لحظه t تخمین بزند. همان‌طور که از شکل تابع مشخص است، این تابع هم به صورت یک تابع احتمال شرطی تعریف شده است. برای مثال شکل ۴ می‌تواند خروجی فیلتر باشد. در مناطقی از نمودار که قله ایجاد شده است، احتمال حضور x در آن مناطق بیشتر است.



شکل ۴. نمونه‌ای از تابع $p(x_t|y_{0:t})$ که به عنوان خروجی ارائه می‌شود.

۳-۳ فیلتر کالمن

آن طور که در تعریف فیلتر کالمن آمده است، f و g باید تابعی خطی و گوسی باشند [۱، ۲]. برای مثال در تعریف (۱) هیچ توانی بالاتر از یک یا پارامتری در مخرج دیده نمی‌شود. چنین تابعی، یک تابع خطی است. اما تابعی به شکل (۲)، تابعی خطی نیست.

$$f(x_t|x_{t-1}) = x_{t-1} + \Delta x_{t-1} \quad (۱)$$

$$f(x_t|x_{t-1}) = x_{t-1} + (\Delta x_{t-1})^2 \quad (۲)$$

با توجه به شرایط مسئله مورد نظر ما در این گزارش و نوع ربات، ممکن است f به صورت (۳) تعریف شود:

$$f(x_t|x_{t-1}) = \begin{bmatrix} x_{t-1} \\ y_{t-1} \end{bmatrix} + \max(\lambda_1 \Delta x_{t-1}, \lambda_2 \Delta y_{t-1}) \quad (۳)$$

این تعریف از f تعریفی غیر خطی می‌باشد چراکه از تابع \max در تعریف آن استفاده شده است. علت تعریف f به شکل بالا یا شکل‌های پیچیده‌تر از آن این است که ممکن است ربات به طور ناگهانی توسط داور جابه‌جا شود. بنابراین امکان جابه‌جایی ناگهانی و پیش‌بینی نشده در ربات وجود داشته و نمی‌توان f را به گونه‌ای خطی تعریف کرد.

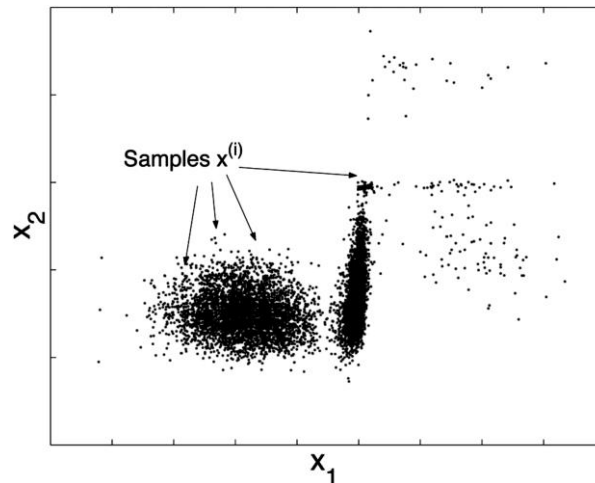
با توجه به توضیحات ارائه شده، این فیلتر برای مسئله مورد بحث مناسب نبوده و بیش از این راجع به این فیلتر صحبت نخواهد شد.

۳-۴ فیلتر ذره

این فیلتر برخلاف فیلتر کامن نیازی به f و g خطی ندارد؛ f و g می‌توانند غیر خطی و غیر گوسی باشند. بنابراین این فیلتر می‌تواند پاسخ‌گوی نیاز ما برای حل مسئله موقعیت‌یابی باشد.

ایده روش فیلتر ذره برای بدست آوردن تابع احتمال $p(x_t|y_{0:t})$ استفاده از روش مونت کارلو می‌باشد. اگر بخواهیم در یک جمله نحوه کارکرد مونت کارلو را توضیح دهیم، می‌توان به این جمله اشاره کرد که مونت

کارلو سعی دارد تا با استفاده از تولید تعداد زیادی نقاط تصادفی یک تابع احتمال را شبیه‌سازی کند؛ به این صورت که هر منطقه‌ای که تراکم نقاط در آن منطقه بیشتر است، احتمال حضور x در آن منطقه بیشتر است. برای مثال شکل ۴ را می‌توان با شکل ۵ شبیه‌سازی کرد. همان‌طور که توضیح داده شد، مناطقی که تراکم نقاط در آن‌ها بیشتر است، تابع احتمال هم در آن مناطق دارای قله است.

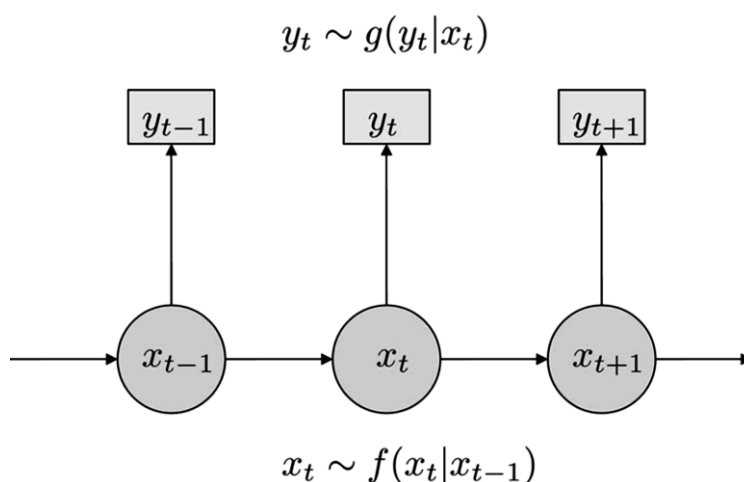


شکل ۵. استفاده از مونت کارلو برای شبیه‌سازی تابع احتمال $p(x_t|y_{0:t})$ که در شکل ۴ در صفحه ۱۳ نشان داده شد

روند کلی نحوه عملکرد فیلتر ذره را می‌توان با شکل ۶ توضیح داد. همان‌طور که در شکل ۶ مشخص است، فیلتر ذره ذاتاً خاصیت بازگشتی دارد. مقدار x در هر لحظه، به لحظه قبل خود وابسته است که نمایانگر خاصیت بازگشتی آن است. برای تخمین مقدار x_t ، t بار باید الگوریتم فیلتر ذره را اجرا کرد. در هر بار اجرای فیلتر ذره، با استفاده از مدل f (مدل سیستم) و مقدار x_{t-1} ، مقدار x_t و سپس با استفاده از مدل g (مدل اندازه‌گیری) اینکه چه مشاهده‌ای باید در موقعیت x_t بدست آید، حدس زده می‌شود. در نهایت از مقایسه مشاهده بدست‌آمده از محیط و مشاهده‌ای که دیدن آن حدس زده شد برای تصحیح خطای حدس بدست‌آمده از مدل سیستم استفاده می‌گردد. همین روند در هر بار اجرای فیلتر ذره تکرار می‌شود و در هر بار اجرای آن، موقعیت x در لحظه بعدی تخمین زده می‌شود.

لازم به ذکر است که ساختار شکل ۶ ساختاری خاص و مشخص است. مقدار x در هر لحظه، تنها به مقدار x در لحظه قبل وابسته است؛ مشاهده y بدست‌آمده در هر لحظه به مقدار x در همان لحظه بستگی دارد و نه

مقدار x در دو لحظه قبل. سیستمی که دارای چنین ساختاری باشد، دارای خاصیت فرآیند مارکوفی نوع اول^۶ است. فیلتر ذره تنها در محیطهایی پاسخگو است که ساختار آنها دارای ویژگی فرآیند مارکوفی نوع اول باشد [۲]. در مورد مسئله ما، این که چه مشاهده‌ای از محیط دریافت شود، کاملاً به موقعیت ربات در زمین وابسته است و نه موقعیت‌های قبلی ربات در زمین؛ یا این که اکنون ربات در چه موقعیتی از زمین قرار دارد، تنها تابعی از موقعیت قبلی او در زمین است و نه موقعیت او در دو لحظه قبل؛ در واقع موقعیت ربات در لحظه قبل اطلاعات کافی را برای حدس موقعیت ربات در لحظه بعد فراهم می‌کند و نیازی به دانستن موقعیت ربات در دو لحظه قبل نیست.



شکل ۶. نحوه کلی عملکرد فیلتر کالمن و فیلتر ذره؛ فرآیند مارکوفی نوع اول.

در این فصل چندین روش حل مسئله بررسی شد:

۱. روش نوابری کورکورانه
۲. فیلتر کامن
۳. فیلتر ذره

در نهایت به دلیل پیچیده بودن مدل‌های مورد نیاز برای حل مسئله، تنها روش فیلتر ذره مناسب شناخته شد که در فصل بعد، به توضیح دقیق‌تر نحوه کارکرد آن پرداخته خواهد شد.

^۶ First order Markov process

۴ روش فیلتر ذره

تا به اینجا، نحوه عملکرد فیلتر ذره به صورت کلی توضیح داده شد. در این فصل قصد داریم تا به صورت دقیق نحوه عملکرد آن را بیان کنیم.

می‌دانید برای بدست آوردن $p(x_t|y_{0:t})$ می‌توان از روابط بازگشتی حدس-تصحیح^۷ استفاده کرد:

• حدس:

$$p(x_t|y_{0:t-1}) = p(x_{t-1}|y_{0:t-1})f(x_t|x_{t-1}) \quad (۴)$$

• تصحیح:

$$p(x_t|y_{0:t}) = p(x_t|y_{0:t-1})g(y_t|x_t) \quad (۵)$$

هدف این است که بتوان از $p(x_t|y_{0:t})$ نمونه برداری کرد. به دلیل اینکه در حالت کلی، نمونه برداری از این توزیع امکان پذیر نیست، باید از روش‌های نمونه برداری و نمونه برداری مجدد استفاده کرد. نمونه‌هایی از این روش‌ها توضیح داده شده است [۲، ۳]. بدین منظور N نقطه یا ذره $\tilde{x}_t^{(i)}$ که $i = 1, \dots, N$ از یک توزیع اهمیت^۸ مناسب $q(x_t|y_{0:t})$ ، نمونه برداری می‌شود. وزن هر نقطه از (۶) قابل محاسبه است.

$$\tilde{\omega}_t^{(i)} = \frac{p(\tilde{x}_t|y_{0:t})}{q(\tilde{x}_t|y_{0:t})}, \quad i = 1, \dots, N \quad (۶)$$

هر قدر q به p نزدیک‌تر باشد، نمونه برداری دقیق‌تر خواهد بود. به این منظور q را به صورت (۷) تجزیه کرده تا مقداری شبیه p گردد.

$$q(x_t|y_{0:t}) = q(x_{t-1}|y_{0:t-1})q(x_t|x_{t-1}, y_t) \quad (۷)$$

حال اگر (۶) را با استفاده از روابط (۷)، (۵) و (۴) بازنویسی کنید، (۸) بدست می‌آید.

^۷ Prediction-Correction

^۸ Importance distribution

$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \times \frac{f(\tilde{x}_t^{(i)} | \tilde{x}_{t-1}^{(i)}) g(y_t | \tilde{x}_t^{(i)})}{q(\tilde{x}_t^{(i)} | \tilde{x}_{t-1}^{(i)}, y_t)} \quad (۸)$$

در گام آخر هم با استفاده از (۹)، به نرمال سازی وزن ها پرداخته می شود؛ چرا که لزوماً با اعمال روش بالا، جمع وزن ها یک نخواهد شد.

$$\omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}} \cdot i = 1. \dots N \quad (۹)$$

تا به اینجا مبانی نظری مبحث فیلتر ذره توضیح داده شد؛ حال می توان الگوریتم فیلتر ذره را بیان کرد. الگوریتم ۱ تحت عنوان فیلتر ذره (حالت کلی) ارائه شده است [۳].

```

for i = 1. .... N do
    Sample  $\tilde{x}_0^{(i)} \sim p(x_0)$ 
    Assign initial importance weights
     $\tilde{\omega}_0^{(i)} = \frac{p(\tilde{x}_0^{(i)}) g(y_0 | \tilde{x}_0^{(i)})}{q(\tilde{x}_0^{(i)} | y_0)}$ 
end for
for t = 1. .... T do
    Resample( $\{x_t^{(0)}. \omega_t^{(0)}\}. \dots \{x_t^{(N)}. \omega_t^{(N)}\}$ )
    for i = 1. .... N do
        propagate
         $\tilde{x}_t^{(i)} \sim q(\tilde{x}_t^{(i)} | x_{t-1}^{(i)}, y_t)$ 
        compute weight
         $\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \times \frac{f(\tilde{x}_t^{(i)} | x_{t-1}^{(i)}) g(y_t | \tilde{x}_t^{(i)})}{q(\tilde{x}_t^{(i)} | x_{t-1}^{(i)}, y_t)}$ 
    end for
    normalize weights
     $\omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}} \cdot i = 1. \dots N$ 
end for

```

الگوریتم ۱. الگوریتم فیلتر ذره در حالت کلی

در الگوریتم ۱ $p(x_0)$ حدس اولیه از مقدار x در لحظه صفر است که آن هم به عنوان ورودی در اختیار این الگوریتم قرار می گیرد. $p(x_0)$ به شکل یک تابع احتمالاتی تعریف شده است.

در این فصل سعی شد تا با جزئیات نظری فیلتر ذره و الگوریتم آن آشنا شوید. حال اینکه چطور باید از این الگوریتم استفاده کرد، در فصل بعد شرح داده خواهد شد.

۵ نحوه استفاده از روش فیلتر ذره

تا کنون، حالت کلی الگوریتم فیلتر ذره مورد بررسی قرار گرفت. در این فصل، چگونگی استفاده از این الگوریتم توضیح داده خواهد شد.

۵-۱ انتخاب توزیع اهمیت مناسب

برای اینکه بتوان از الگوریتم فیلتر ذره استفاده نمود باید یک توزیع اهمیت q را انتخاب کرد؛ هر چه این توزیع به توزیع p نزدیکتر باشد، نتیجه فیلتر ذره دقیقتر خواهد بود. در ساده‌ترین نوع فیلتر ذره که بوت-استرپ^۹ نامیده می‌شود، توزیع اهمیتی را پیشنهاد می‌کند که باعث ساده شدن بسیاری از روابط می‌شود.

۵-۱-۱ روش فیلتر ذره بوت‌استرپ

در این روش پیشنهاد شده است که از مدل f (مدل سیستم) به عنوان q استفاده شود. با جایگزین کردن q با f ، شبه کد به شکل الگوریتم ۲ در می‌آید [۳].

```

for  $i = 1 \dots N$  do
    Sample  $\tilde{x}_0^{(i)} \sim p(x_0)$ 
    Assign initial importance weights
    
$$\tilde{\omega}_0^{(i)} = \frac{p(\tilde{x}_0^{(i)})g(y_0|\tilde{x}_0^{(i)})}{f(\tilde{x}_0^{(i)})}$$

end for
for  $t = 1 \dots T$  do
    Resample( $\{x_t^{(0)}, \omega_t^{(0)}\}, \dots, \{x_t^{(N)}, \omega_t^{(N)}\}$ )
    for  $i = 1 \dots N$  do
        propagate
        
$$\tilde{x}_t^{(i)} \sim f(\tilde{x}_t^{(i)}|x_{t-1}^{(i)})$$

        compute weight
        
$$\tilde{\omega}_t^{(i)} = \omega_{t-1}^{(i)} \times g(y_t|\tilde{x}_t^{(i)})$$

    end for

```

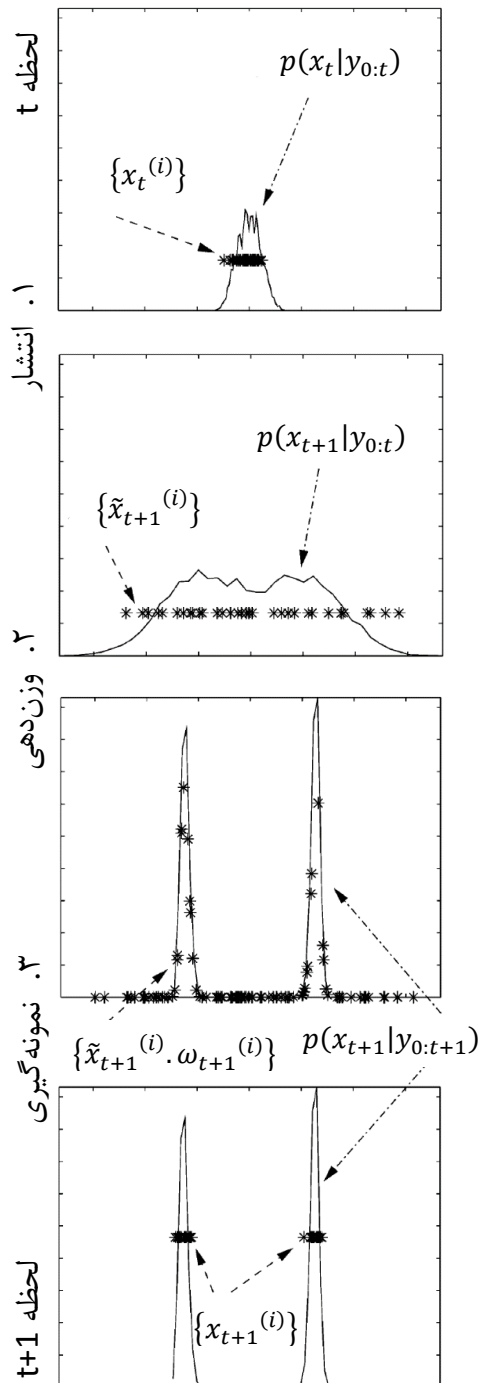
^۹ Bootstrap

normalize weights

$$\omega_t^{(i)} = \frac{\tilde{\omega}_t^{(i)}}{\sum_{j=1}^N \tilde{\omega}_t^{(j)}} \cdot i = 1, \dots, N$$

end for

الگوریتم ۲. الگوریتم فیلتر ذره بوت استرپ



شکل ۷.

روند کلی نحوه عملکرد الگوریتم فیلتر ذره
بوت استرپ

اگر بخواهیم نحوه عملکرد الگوریتم فیلتر ذره بوت استرپ را با شکل توضیح دهیم، به این صورت است که ابتدا مجموعه ای از نقاط $\{x_t^{(i)}\}$ به الگوریتم داده می شود؛ سپس این نقاط انتشار می یابند؛ به عبارت دیگر با استفاده از مدل f موقعیت بعدی تمام نقاط حدس زده می شود. در مرحله بعد به این نقاط با توجه به مدل اندازه گیری g ، وزن داده شده و در نهایت عمل نمونه گیری مجدد روی این نقاط انجام می گیرد. تمام مراحل به طور کامل در شکل ۷ نشان داده شده اند.

۲-۵ مدل سازی

همان طور که در فصل های قبل توضیح داده شد، دو مدل در فیلتر ذره مطرح هستند؛ مدل سیستم و مدل اندازه گیری. هنگام بیان نحوه عملکرد فیلتر ذره، فرض بر این است که این دو مدل آماده بوده و تنها از آن ها استفاده می شود؛ اما این که این دو مدل چگونه طراحی شوند، خود مسئله ای فارغ از الگوریتم فیلتر ذره است و در صورت انتخاب نادرست آن ها، نتیجه ی مورد انتظار بدست نخواهد آمد.

این که این دو مدل چگونه انتخاب شوند، خود نیازمند اندازه‌گیری‌ها و طراحی‌های خاصی است که از حوصله این گزارش خارج می‌باشد؛ از این رو تنها نمونه‌ای از این دو مدل (ارائه شده توسط تیم بی‌هیومن^{۱۰} از دانشگاه برمن^{۱۱} آلمان) در این فصل آورده شده است [۱].

۵-۲-۱ مدل سیستم

$$\rightarrow \begin{bmatrix} x_t^{(i)} \\ y_t^{(i)} \end{bmatrix} = \begin{bmatrix} x_{t-1}^{(i)} \\ y_{t-1}^{(i)} \end{bmatrix} + R_{t-1}^{(i)} \begin{bmatrix} \Delta x_t + \text{sample} \left(\max \left(\lambda_1 \Delta x_t + \lambda_2 \Delta y_t + \lambda_3 \bar{\omega}_{t-1}^{(i)} \right) \right) \\ \Delta y_t + \text{sample} \left(\max \left(\lambda_1 \Delta x_t + \lambda_2 \Delta y_t + \lambda_3 \bar{\omega}_{t-1}^{(i)} \right) \right) \end{bmatrix} \quad (10)$$

$$\rightarrow \theta_t^{(i)} = \theta_{t-1}^{(i)} \Delta \theta_t + \text{sample} \left(\max \left(\lambda_4 \Delta \theta_t + \lambda_5 |(\Delta x_t, \Delta y_t)| + \lambda_6 \bar{\omega}_{t-1}^{(i)} \right) \right) \quad (11)$$

$$\bar{\omega}_t^{(i)} = \max \left(\frac{\sum_i \omega_t^{(i)}}{N \omega_t^{(i)}} - 1.0 \right)^2 \quad (12)$$

متغیرهای $\lambda_1, \lambda_2, \dots, \lambda_6$ در (۱۰) و (۱۱) ضرایب ثابتی هستند که برای شبیه‌سازی خطای جابه‌جایی و جابه‌جایی ناگهانی استفاده شده‌اند. همان‌طور که قبلاً هم توضیح داده شد، دلیل پیچیدگی رابطه‌ها، مدل کردن اتفاقات مختلف در زمین است؛ برای مثال برای مدل کردن جابه‌جایی ربات توسط داور و به صورت ناگهانی، مدل باید به صورتی طراحی شده باشد که احتمال جابه‌جا شدن ناگهانی ربات را در هر لحظه در نظر گرفته باشد. بنابراین ممکن است برای مدل کردن تعداد بیشتری از اتفاقات، مدل پیچیده‌تری را ارائه کرد.

۵-۲-۲ مدل اندازه‌گیری

در این قسمت تنها نحوه عملکرد مدل اندازه‌گیری را توضیح داده خواهد شد. در این مدل، ابتدا ربات خود را جای یکی از نقطه‌ها گذاشته و مشاهده‌ای که باید از این موقعیت دریافت شود را محاسبه می‌کند. در نهایت

^{۱۰} BHuman

^{۱۱} Bremen

با استفاده از محاسبه اختلاف مشاهده حدس زده شده و مشاهده بدست آمده در چرخه فعلی و استفاده از توزیع نرمال، عددی را به عنوان احتمال دریافت این مشاهده در این نقطه ارائه می‌کند.

۳-۱۵ استفاده از خروجی فیلتر ذره

مسئله آخر در استفاده از فیلتر ذره نحوه استفاده از $p(x_t|y_{0:t})$ است. در این مورد هم روش‌های متفاوتی ارائه شده است؛ به عنوان مثال می‌توان از میانگین وزن دار (امید ریاضی متغیر x) که در (۱۳) تعریف شده است استفاده کرد.

$$E(x) = \int x_t p(x_t|y_{0:t}) \approx \sum_{i=1}^N x_t^{(i)} \omega_t^{(i)} \quad (13)$$

در این فصل هم سعی شد تا با نحوه استفاده از روش فیلتر ذره و جزئیات کاربردی آن آشنا شوید. همان‌طور که توضیح داده شد، نحوه عملکرد این فیلتر با میزان مناسب انتخاب کردن مدل‌ها و توزیع اهمیت رابطه مستقیم داشته که عدم توجه به این موضوع می‌تواند منجر به گرفتن نتیجه‌ای دور انتظار گردد.

۶ نتیجه‌گیری

در این گزارش سعی شد روش‌های مختلف موقعیت‌یابی در ربات‌های انسان‌نما در مسابقه فوتبال ارائه گردد. اولین و ساده‌ترین روش، ناوبری کورکورانه بود که به علت خطای زیاد مورد استفاده قرار نگرفت. در گام بعدی فیلتر کالمن معرفی شد که به دلیل مشروط بودن به داشتن مدل سیستم و اندازه‌گیری خطی، با مسئله مورد نظر بحث در این گزارش مطابقت نداشت. در نهایت فیلتر ذره به عنوان راه حل نهایی معرفی شد که محدودیتی در مدل سیستم و اندازه‌گیری نداشت. در انتها هم مثال‌هایی از مدل سیستم و اندازه‌گیری و نحوه استفاده از خروجی الگوریتم ارائه شد.

لازم به ذکر است که مهم‌ترین نکته در استفاده از روش فیلتر ذره، انتخاب توزیع اهمیت، مدل سیستم و اندازه‌گیری مناسب می‌باشد. هر چه توزیع اهمیت به توزیع $p(x_t|y_{0:t})$ نزدیک‌تر باشد، جواب الگوریتم، دقیق‌تر خواهد بود؛ به همین صورت اگر مدل سیستم و اندازه‌گیری به درستی انتخاب شوند، موجب بهتر شدن جواب می‌شوند و بالعکس.

به عنوان کارهای آینده می‌توان در زمینه انتخاب توزیع اهمیت، مدل سیستم و اندازه‌گیری مناسب پژوهش کرد؛ چرا که تاثیر مستقیم در جواب الگوریتم دارند.

مراجع و منابع مفید

1. T. Laue and T. Röfer, "Particle filter-based state estimation in a competitive and uncertain environment," *Proceedings of the 6th International Workshop on Embedded Systems, Vaasa, Finland, 2007*.
2. M.S. Arulampalam, et al., "A tutorial on particle filters for online nonlinear/non-Gaussian Bayesian tracking," *Signal Processing, IEEE Transactions on*, vol. 50, no. 2, 2002, pp. 174-188.
3. O. Cappé, et al., "An overview of existing methods and recent advances in sequential Monte Carlo," *Proceedings of the IEEE*, vol. 95, no. 5, 2007, pp. 899-924.
4. C.M. Bishop, *Pattern recognition and machine learning*, springer, 2006.
5. T. BO, "Self-Localization of Humanoid Robot in a Soccer Field," 2010.
6. S.-B. Han, et al., "Landmark-based particle localization algorithm for mobile robots with a fish-eye vision system," *Mechatronics, IEEE/ASME Transactions on*, vol. 18, no. 6, 2013, pp. 1745-1756.
7. E. Hashemi, et al., "Particle filter based localization of the nao biped robots," *Proc. System Theory (SSST), 2012 44th Southeastern Symposium on*, IEEE, 2012, pp. 168-173.
8. T. Laue, et al., "Efficient and reliable sensor models for humanoid soccer robot self-localization," *Proceedings*

of the Fourth Workshop on Humanoid Soccer Robots in conjunction with the, 2009, pp. 22-29.

9. E. Munera Sánchez, et al., "A Reliability-Based Particle Filter for Humanoid Robot Self-Localization in RoboCup Standard Platform League," *Sensors*, vol. 13, no. 11, 2013, pp. 14954-14983.

10. B. Tian, et al., "Self-localization of humanoid robots with fish-eye lens in a soccer field," *Proc. Robotics Automation and Mechatronics (RAM), 2010 IEEE Conference on*, IEEE, 2010, pp. 522-527.