مقایسه روشهای مختلف خانواده فیلترهای کالمن در بهبود دقت تخمین برای موقعیت یابی و ساختن نقشه همزمان در یک ربات متحرک

علیرضا محمد شهری، خوشنام شجاعی، کاوه احمدی

دانشگاه علم و صنعت ایران، دانشکده مهندسی برق، پژوهشکده الکترونیک

shahri@iust.ac.ir, kh_shojaii@ee.iust.ac.ir, kaveh_ahmadi@ee.iust.ac.ir

چکیده - در این مقاله، مسأله *موقعیت یابی و ساختن نقشه همزمان* (*SLAM*) را بر روی یک ربات متحرک خودمختار با استفاده از روشهای مختلف در خانواده فیلترهای کالمن بررسی می‌کنیم. روش معمول برای موقعیت یابی و ساخت نقشه‌های هندسی در ربات‌های متحرک خودمختار فیلتر کالمن توسعه یافته است. این مقاله، با به کار گیری نسخه‌های اصلاح شده فیلتر کالمن برای الگوریتم *SLAM* برای اولین بار نشان خواهد داد که خطای ناشی از خطی سازی مدلها در فیلتر کالمن توسعه یافته یک بار از طریق خطی سازی تحلیلی مجدّد مدل اندازه گیری و بار دیگر از طریق خطی سازی آماری مدلهای پروسه و اندازه‌گیری فوق العاده بهبود پیدا می‌کند. نهایتاً، با ارائه نتایج شبیه سازی، برتری و کارآیی بهتر این نسخه‌های اصلاح شده در خانواده فیلترهای کالمن برای حل مسأله *SLAM* را نسبت به فیلتر کالمن توسعه یافته اثبات خواهیم کرد.

کلید واژه- تخمین حالت، ساختن نقشه، فیلتر کالمن توسعه یافته تکراری، فیلتر کالمن نقطه سیگما، موقعیت یابی ربات متحرک.

۱- مقدمه

مسأله موقعیت یابی و ساختن نقشه بطور همزمان[1] در ربات‌های متحرک (SLAM) عبارت است از فرآیند ساختن یک نقشه از مشخصه‌های قابل درک در محیط و استفاده از این نقشه بطور همزمان برای تخمین موقعیت سیستم متحرک، که برای این کار ربات به توانایی‌اش در بدست آوردن و استخراج اطلاعات ناوبری مفید از داده‌های سنسورهای خارجی تکیه دارد و هیچ دانشی از محیط ندارد.

روش معمول مورد استفاده برای این مسأله فیلتر کالمن توسعه یافته[2] (EKF) است که به صورت بازگشتی تخمین پیوسته‌ای از حالت‌های ربات و مشخصه‌های محیط را بدست می‌دهد [۱]، [۲]. مدل ربات و سیستم اندازه‌گیری طبیعتاً غیر خطی هستند. به همین سبب، خطی سازی این مدلها در الگوریتم EKF برای محاسبه ماتریس فرآیند و مشاهده از طریق محاسبه ژاکوبین سبب ایجاد بایاس در فیلتر می‌شود و همگرایی فیلتر کالمن را به خطر می‌اندازد [۶]. هر چه میزان غیر خطی بودنِ مدلها بیشتر باشد، این خطای خطی سازی بزرگ تر خواهد بود. با به کار گیری فیلتر کالمن توسعه یافته تکراری[3] (IEKF) می‌توان دقت تخمین حالتها را در الگوریتم SLAM افزایش داد و مانع ناسازگاری و واگرایی الگوریتم شد. این فیلتر با خطی سازی تحلیلی مجدد مدل مشاهده، خطای خطی سازی را کاهش می‌دهد.

خولیر و اولمن در سال ۱۹۹۷ مقاله‌ای را ارائه کردند که اساس شکل گیری نسخه دیگری از فیلترهای کالمن به نام *فیلترهای کالمن نقطه سیگما*[4] (SPKF) محسوب می‌شود و برتری فوق‌العاده‌ای را نسبت به EKF نشان می‌دهد. این فیلترها به جای خطی سازی تحلیلی مدلهای فرآیند و مشاهده، تابع توزیع احتمال متغیر حالت تصادفی را تا مرتبه دوم تقریب می‌زنند [۳]، [۴]، [۵]. این نسخه جدید در

۱ Simultaneous Localization And Mapping (SLAM)
۲ Extended Kalman Filter (EKF)
۳ Iterated Extended Kalman Filter (IEKF)
۴ Sigma Point Kalman Filters (SPKF)

خانواده فیلترهای کالمن یک نوع خطی سازی آماری را روی مدل غیر خطی انجام می‌دهد و دقت تخمین را فوق‌العاده بهبود می‌بخشد. در این مقاله، ابتدا در بخش (۲) مروری بر حل مسأله SLAM با استفاده از فیلتر کالمن توسعه یافته خواهیم داشت. سپس در بخش (۳) فیلتر کالمن توسعه یافته تکراری را بررسی می‌کنیم. در بخش (۴)، فیلتر کالمن نقطه سیگما را معرفی می‌کنیم. در بخش (۵)، الگوریتم SPKF را برای حل مسأله SLAM ارائه می‌کنیم. نتایج شبیه سازی در بخش (۶) ارائه می‌شود تا عملکرد بهتر نسخه‌های اصلاح شده در خانواده فیلترهای کالمن را برای حل مسأله SLAM نسبت به EKF نشان دهد. نهایتاً، در بخش (۷) از مقاله نتیجه گیری می‌شود.

۲- مروری بر الگوریتم EKF-SLAM

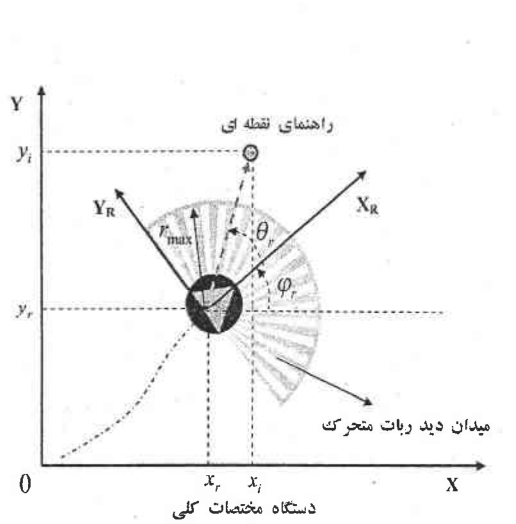
در این بخش بطور خلاصه مسأله SLAM را برای یک ربات متحرک خودمختار مرور می‌کنیم. ساختار الگوریتم SLAM مبتنی بر مشخصه‌های هندسی با به کار گیری روش فیلتر کالمن توسعه یافته به صورت زیر است. فرض می‌کنیم که ربات دارای سنسورهای اندازه گیری فاصله و زاویه مشخصه-ها نسبت به دستگاه محلی (دستگاه متصل به بدنه ربات) باشد. بردار حالت ربات را با $X_r(k)$ و بردار حالت همه مشخصه‌های محیط (راهنماها) را با $X_m(k)$ نشان می-دهیم. هدف تخمین توأم بردار حالت ربات و موقعیت مشخصه‌های محیط است. برای این منظور، بردار حالت ربات و مشخصه‌ها را در یک بردار حالت کامل، $X(k)$، قرار می-دهیم و به دنبال تخمین این بردار حالت کامل، یعنی $\hat{X}(k)$ می‌گردیم [۱].

$$X(k) = [X_r(k)^T \quad X_m(k)^T]^T, \quad i = 1,\dots,N \qquad (۱)$$

که در آن بردار حالت ربات مشخص کننده موقعیت و سمت گیری ربات در نمایش دو بعدی است. راهنماها نیز به صورت نقطه‌ای در نظر گرفته می‌شوند.

N تعداد راهنماهای محیط است. مدل سیستم متحرک (ربات) که با فرض ورودی کنترلی $u(k)$، $X_r(k-1)$ را به $X_r(k)$ می‌نگارد، را به صورت زیر قرار می‌دهیم:

$$X_r(k) = f(X_r(k-1), u(k)) + w(k) \qquad (۲)$$



شکل ۱: نمایش دستگاه مختصات محلی، کلی و میدان دید یک ربات مجهز به یک فاصله یاب برای اندازه گیری فاصله و زاویه.

مدل مشخصه‌ها نیز با توجه به فرض ایستا بودن آنها در محیط به صورت زیر در نظر می‌گیریم:

$$X_m(k) = X_m(k-1) \qquad (۳)$$

نویز پروسه، $w(k)$ ، را با نویز سفید با توزیع احتمال گاوسی با میانگین صفر وکوواریانس Q مدل می کنیم:

$$E[w(k)] = 0 \quad , \quad E[w(k).w^T(k)] = Q \qquad (۴)$$

مدل اندازه گیری نیز به صورت زیر در نظر گرفته می‌شود:

$$Z(k) = h(X(k)) + v(k) \qquad (۵)$$

نویز اندازه‌گیری، $v(k)$ ، را نیز با نویز سفید با توزیع احتمال گاوسی با میانگین صفر و کوواریانس R مدل سازی می-کنیم:

$$E[v(k)] = 0 \quad , \quad E[v(k).v^T(k)] = R \qquad (۶)$$

نویز اندازه گیری از نویز پروسه مستقل است. یعنی با هم هیچ همبستگی ندارند:

$$E[v(k).w^T(k)] = 0 \qquad (۷)$$

الگوریتم SLAM با استفاده از فیلتر کالمن از چهار بخش پیش بینی حالت، مشاهده، بهنگام سازی و افزودن مشخصه جدید تشکیل می‌شود.

۲-۱- مرحله پیش‌بینی حالت

با توجه به اطلاعات بدست آمده از اُدومتری، و بردار حالت کامل بهنگام شده در مرحله زمانی قبلی $(k-1)$ و ورودی کنترلی، یک پیش بینی از موقعیت ربات و راهنماها را با توجه به مدل پروسه بدست می‌آوریم [۱]. پیش بینی بردار حالت را با بالانویس (-) و بهنگام سازی آن را با بالانویس (+) نشان می‌دهیم. مدل پروسه به صورت زیر نوشته می‌شود:

$$\begin{aligned}\hat{X}_r(k) &= f(\hat{X}_r(k-1), u(k-1)) \\ \hat{X}_m(k) &= \hat{X}_m(k-1)\end{aligned} \qquad (۸)$$

بنابراین پیش‌بینی بردار حالت به صورت زیر خواهد بود:

$$\hat{X}^-(k) = \begin{bmatrix} f(\hat{X}_r^+(k-1), u(k-1)) \\ \hat{X}_m^+(k-1) \end{bmatrix} \qquad (۹)$$

ماتریس کوواریانس پیش‌بینی نیز به صورت زیر محاسبه می‌شود:

$$P^-(k) = \nabla_X f_{k-1} . P^+(k-1) . \nabla_X f_{k-1}^T + Q \qquad (۱۰)$$

که در آن $\nabla_X f$ ماتریس پروسه است. ساختار ماتریس کوواریانس نیز به صورت زیر خواهد بود:

$$P^-(k) = \begin{bmatrix} P^{rr-}(k) & P^{ri-}(k) \\ P^{ir-}(k) & P^{ij-}(k) \end{bmatrix} \qquad (۱۱)$$

که در آن P^{rr-} ماتریس کوارایانس حالتهای ربات است، P^{ri-} ماتریس کواریانس ربات- راهنما است که همبستگی میان حالتهای ربات و راهنماها را نشان می‌دهد و P^{ij-} نیز ماتریس کواریانس میان حالتهای راهنماها با همدیگر را نمایش می‌دهد. با توجه به مدل اندازه‌گیری در رابطه (۵) پیش بینی مشاهدات به صورت زیر خواهد بود:

$$\hat{Z}^-(k) = h(\hat{X}^-(k)) \qquad (۱۲)$$

۲-۲- مرحله مشاهده

با استفاده از یک سنسور اندازه‌گیری فاصله و زاویه، بردار مشاهده واقعی نیز بدست می‌آید:

$$Z(k) = \begin{bmatrix} Z_R(k) \\ Z_\theta(k) \end{bmatrix} \qquad (۱۳)$$

سپس ، خطای اندازه گیری که تفاضل مشاهدات واقعی و پیش‌بینی شده است، را محاسبه می‌کنیم:

$$v(k) = Z(k) - \hat{Z}(k) \qquad (۱۴)$$

عدم قطعیت در خطای اندازه گیری با رابطه (۱۵) بدست می‌آید:

$$S(k) = \nabla_X h(k) . P^-(k) . \nabla_X h^T(k) + R(k) \qquad (۱۵)$$

۲-۳- مرحله بهنگام سازی

چنانچه همبستگی *داده‌ها* که فرآیند نسبت دادن مشاهدات پیش بینی شده به مشاهدات واقعی است را حل شده فرض کنیم، بردار حالت پیش بینی شده با روابط (۱۶) تا (۱۸) بهنگام سازی خواهد شد:

$$W(k) = P^-(k) . \nabla_X h^T(k) . S^{-1}(k) \qquad (۱۶)$$

که در آن، $\nabla_X h^T$ ماتریس مشاهده است و W بهره کالمن است. معادلات بهنگام سازی حالت و ماتریس کوواریانس عبارتند از:

$$\hat{X}^+(k) = \hat{X}^-(k) + W(k) . v(k) \qquad (۱۷)$$

$$P^+(k) = P^-(k) - W(k) . S(k) . W^T(k) \qquad (۱۸)$$

۲-۴- افزودنِ راهنمای جدید[۱]

چنانچه راهنمای مشاهده شده قبلاً توسط سنسورهای ربات دیده نشده باشد، به انتهای بردار حالت افزوده می‌شود و ماتریس کوواریانس آن نیز محاسبه می‌شود [۲].

با توجه به روابط (۱۰)، (۱۵) و (۱۶) محاسبه ژاکوبین‌ها برای خطی سازی مرتبه اول مدلهای پروسه و مشاهده حول تخمین بردار حالت منجر به یک خطای بزرگ می شود و این خطای خطی سازی در بدترین حالت به خصوص وقتی که درجه غیر خطی بودنِ مدلها خیلی زیاد باشد، می‌تواند منجر به واگرایی فیلتر و نهایتاً نقشه گردد. علاوه بر این، وجود ماتریس‌های ژاکوبین در فیلتر کالمن توسعه یافته باعث می‌شود که پیاده‌سازی عملی الگوریتم اندکی مشکل شود [۳]. در بخش ۴، فیلتر کالمنِ نقطه سیگما بعنوانِ یک فیلتر قدرتمند که هیچ گونه خطی سازی تحلیلی روی مدلها انجام نمی‌دهد و فقط توزیع احتمال تخمین را تا مرتبه دوم

۱ New Landmark Augmentation

تقریب می‌زند، را معرفی خواهیم کرد.

۳- فیلتر کالمن توسعه یافته تکراری

فیلتر کالمن توسعه یافته مدلهای پروسه و مشاهده را به ترتیب حول متوسط توزیع احتمال پسین مرحله زمانی قبلی (x_{k-1}^{+}) و متوسط توزیع احتمال پیشین مرحله زمانی کنونی (x_k^-) خطی سازی می‌کند. وقتی درجه غیر خطی بودن مدلها قابل توجه باشد، بهتر است که از فیلتر کالمن توسعه یافته تکراری استفاده کنیم. فیلتر کالمن توسعه یافته تکراری (IEKF) چندین بار پیش‌بینی تخمین، x_k^-، را بهنگام سازی می‌کند تا به تخمین بهتری برسد. IEKF، این کار را با محاسبه مکرر بهره کالمن K_k و یک تخمین حالت پسین میانی $\hat{x}_k^i$ که i تعداد تکرار بهنگام سازی است، انجام می‌دهد [۷]. برای این کار، محاسبات زیر مکرراً انجام می‌شود:

$$K_k = P_k^- H_k^T \,(H_k P_k^- H_k^T + R_k)^{-1} \qquad (۱۹)$$

$$\hat{x}_k^{i+1} = \hat{x}_k^- + K_k(z_k - h(\hat{x}_k^i) - H_k(\hat{x}_k^- - \hat{x}_k^i)) \qquad (۲۰)$$

که ماتریس ژاکوبین H_k حول تازه ترین تخمین حالت میانی $\hat{x}_k^i$ محاسبه می‌شود. بعد از تعداد معینی تکرار مرحله بهنگام سازی یا هنگامیکه تغییرات $\hat{x}_k^i$ تقریباً ثابت شد، IEKF تخمین حالت بهنگام شده $\hat{x}_k^+$ را برابر آخرین مقدار تخمین $\hat{x}_k^i$ قرار می‌دهد و P_k^+ را با استفاده از آخرین مقادیر K_k و H_k محاسبه می‌کند. در نتیجه بخاطر محاسبه مجدد تابع اندازه گیری h و ژاکوبین H_k، تخمین پسین بهبود پیدا می‌کند، ولی حجم محاسبات افزایش می‌یابد. در واقع، IEKF خطای خطی سازی مدل مشاهده را با خطی سازی مجدد مدل حول تخمین حالت پسین میانی $\hat{x}_k^i$ کاهش می‌دهد و به دنبال بهترین تخمین می‌گردد.

۴- الگوریتم SPKF-SLAM

همانطور که قبلاً ذکر شد، خطی سازی مدلهای غیر خطی پروسه و مشاهده می‌تواند منجر به خطای تخمین قابل ملاحظه در حالت ربات و موقعیت راهنماهای محیط شود. ابتدا در قسمت ۴-۱ فیلتر کالمن نقطه سیگما را بعنوان یک راه حل برای جلوگیری از این خطی سازی معرفی می‌کنیم و سپس در قسمت ۴-۲ از آن برای حل مسأله SLAM استفاده می‌کنیم.

۴-۱- فیلتر کالمن نقطه سیگما

قلب فیلتر کالمن نقطه سیگما مجموعه‌ای از نقاط نمونه وزن دهی شده هستند که نقاط سیگما نامیده می‌شوند. این نقاط بطور کاملاً قطعی انتخاب می‌شوند و متوسط و کوواریانس توزیع احتمال پیشین متغیر تصادفی را بطور کامل بدست می‌آورند [۴]. متغیر تصادفی N بُعدی X را در نظر بگیرید که از طریق تابع غیر خطی f به متغیر تصادفی y نگاشته می‌شود:

$$y = f(X) \qquad (۲۱)$$

توزیع احتمال X دارای متوسط $\hat{X}$ و ماتریس کوواریانس P است. ابتدا نقاط سیگما را با طرح زیر انتخاب می‌کنیم. تعداد این نقاط سیگما $2N+1$ است بطوریکه اولین نقطه متوسط توزیع احتمال در نظر گرفته می‌شود. بقیه نقاط با جابجایی متوسط به اندازه مثبت و منفی مضربی از جذر ماتریس کوواریانس بدست می‌آیند. بنابراین، این نقاط حول متوسط متقارن هستند.

$$\chi_0 = \hat{X} \qquad (۲۲)$$

$$\chi_i = \hat{X} + (\gamma\sqrt{P})_i \;,\quad i = 1,\ldots,N \qquad (۲۳)$$

$$\chi_i = \hat{X} - (\gamma\sqrt{P})_i \;,\quad i = N+1,\ldots,2N \qquad (۲۴)$$

که γ یک پارامتر وزن‌دهی است که پراکندگی نقاط سیگما حول متوسط را کنترل می‌کند. برای محاسبه مؤثر جذر ماتریس کوواریانس از فاکتور چولسکی استفاده می‌کنیم، مشروط بر اینکه ماتریس مثبت نیمه معین باشد [۳]، [۴]. ماتریس نقاط سیگما را هم به صورت زیر تشکیل می‌دهیم:

$$\chi(k) = [\,\chi_0(k), \chi_i(k), \chi_{i+N}(k)\,] \;,\quad i = 1,\ldots,N \qquad (۲۵)$$

سپس با انتشار نقاط سیگما از تابع غیر خطی f، نقاط سیگمای تبدیل یافته را بدست می‌آوریم:

$$\gamma_i = f(\chi_i) \quad i = 0,\ldots,2N \qquad (۲۶)$$

متوسط و ماتریس‌های کوواریانس توزیع احتمال به صورت زیر محاسبه می‌شوند:

$$\hat{y} \approx \sum_{i=0}^{2N} w_i^m \, \gamma_i \qquad (۲۷)$$

$$P_y \approx \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^c \, \gamma_i \, \gamma_j^T \qquad (۲۸)$$

$$P_{xy} \approx \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^c \, \chi_i \, \gamma_j^T \qquad (۲۹)$$

این معادلات تا مرتبه دوم بسط سری تیلور دقت دارند. پارامترهای w_i^m و w_{ij}^c وزنهای عددی مثبت مقداری هستند که برای مقادیر معینی از $\{i, j ; i \neq j\}$ صفر هستند. مقدار پارامترهای w و γ نیز به نوع روش نقطه سیگما وابسته‌اند. این روشها عبارتند از *تبدیل نامحسوس و تبدیل تفاضل مرکزی*. همانطور که در معادلات اخیر دیده می‌شود، هیچ ژاکوبین یا مشتق تحلیلی وجود ندارد. این روشها می‌توانند به صورت خطی سازی آماری تابع غیر خطی با تکنیکی به نام *رگرسیون خطی آماری وزن‌دهی شده*[1] (WSLR) توصیف شوند. این نوع خطی سازی بسیار دقیق تر از خطی سازی تحلیلی موجود در فیلتر کالمن توسعه یافته است [۴]، [۵].

۴-۲- حل مسأله SLAM مبتنی بر SPKF

در این بخش با به کار گیری فیلتر کالمن نقطه سیگما مسأله SLAM را حل می‌کنیم. با در نظر گرفتنِ فرضهای انجام شده در قسمت (۲) برای مسأله SLAM در رباتیک متحرک، ما یک بردار حالت افزوده شده جدید تعریف می‌کنیم که متشکل از $X(k)$ و بردار نویز پروسه است. ماتریس کوواریانس جدید نیز یک ماتریس قطری بلوکی است که بلوکهای آن را $P(k)$ و ماتریس کوواریانس نویز پروسه $Q(k)$ تشکیل می‌دهند [۵].

$$X_k^a = [X_k \;\; V_k]^T \quad , \quad P_k^a = \begin{bmatrix} P_k & 0 \\ 0 & Q_k \end{bmatrix} \qquad (۳۰)$$

سپس ماتریس نقاط سیگما را هم به صورت زیر محاسبه می‌کنیم:

$$\chi^{a+} = [\chi_0^+ , \chi_i^+ + \gamma\sqrt{P_k^a} , \chi_i^+ - \gamma\sqrt{P_k^a}\,] \qquad (۳۱)$$

این نقاط سیگما را از مدل پروسه می‌گذرانیم و نقاط سیگمای تبدیل یافته را محاسبه می‌کنیم:

$$\chi_i^{x-}(k+1) = f(\chi_i^{x+}(k), u(k), \chi_i^{V+}(k)) \qquad (۳۲)$$

که $\chi_i^{x+}(k)$ نقاط سیگما برای بردار حالت کامل اصلی $X(k)$ و $\chi_i^{V+}(k)$ نقاط سیگما برای نویز پروسه است. در این صورت، متوسط و توزیع احتمال پیشین (پیش بینی شده) را با روابط زیر می‌توان محاسبه نمود:

$$\hat{X}_{k+1}^{x-} = \sum_{i=0}^{2N} w_i^m \, \chi_{i,\,k+1}^{x-} \qquad (۳۳)$$

$$P_{k+1}^- = \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^c \, (\chi_{i,\,k+1}^{x-})(\chi_{j,\,k+1}^{x-})^T \qquad (۳۴)$$

نقاط سیگمای تبدیل یافته را از مدل اندازه گیری می گذرانیم و نمونه‌های مشاهده پیش بینی شده را بدست می آوریم:

$$Z_i^-(k+1) = h(\chi_i^{x-}(k+1)) \qquad (۳۵)$$

پس از آن می‌توانیم پیش بینی اندازه گیری و ماتریس‌های کوواریانس را از روی نقاط سیگمای تبدیل یافته محاسبه کنیم:

$$\hat{z}_{k+1}^- = \sum_{i=0}^{2N} w_i^m \, Z_{i,\,k+1}^- \qquad (۳۶)$$

با فرض صحیح بودنِ همبستگی داده‌ها در مسأله SLAM خطای اندازه گیری و ماتریس کوواریانس متناظر با آن، P_{ZZ}^-، و همچنین ماتریس کوواریانس میان بردار حالت و اندازه گیری، P_{XZ}^-، به صورت زیر محاسبه می‌شوند:

$$v_{k+1} = z_{i,\,k+1} - \hat{z}_{i,\,k+1}^- \qquad (۳۷)$$

$$P_{ZZ,k+1}^- = R + \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^c \, (Z_{i,\,k+1}^-).(Z_{j,\,k+1}^-)^T \qquad (۳۸)$$

$$P_{XZ,k+1}^- = \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^c \, (\chi_{i,\,k+1}^{x-}).(Z_{j,\,k+1}^-)^T \qquad (۳۹)$$

سپس متوسط و ماتریس کوواریانس توزیع احتمال پسین را با روابط زیر بدست می‌آوریم:

$$\hat{X}_{k+1}^+ = \hat{X}_{k+1}^- + K_{k+1}.v_{k+1} \qquad (۴۰)$$

$$\hat{X}_{k+1}^+ = \hat{X}_{k+1}^- + K_{k+1}.v_{k+1} \qquad (۴۱)$$

[1] Weighted Statistical Linear Regression (WSLR)

$$P_{k+1}^{+} = P_{k+1}^{-} - K_{k+1} \cdot P_{ZZ,k+1}^{-} \cdot K_{k+1}^{T} \quad (۴۲)$$

در روابط (۳۷) و (۳۸)، K_{k+1} بهره کالمن است که از رابطه زیر محاسبه می‌شود:

$$K_{k+1} = P_{XZ,\,k+1}^{-} \cdot [P_{ZZ,\,k+1}^{-}]^{-1} \quad (۴۳)$$

وقتی یک راهنمای جدید توسط سنسورهای ربات آشکار می‌شود، فرآیند افزودنِ مشخصه جدید به انتهای بردار حالت کامل انجام می‌شود. اگر z_{k+1} بردار اندازه گیری واقعی با ماتریس کوواریانس R_{k+1} باشد، خواهیم داشت:

$$X_{k+1}^{AUG} = [X_{k+1}^{-} \;\; z_{k+1}]^{T} \;,\; P_{k+1}^{AUG} = \begin{bmatrix} P_{k+1}^{-} & 0 \\ 0 & R_{k+1} \end{bmatrix} \quad (۴۴)$$

پس از تشکیل نقاط سیگما، داریم:

$$\chi_{i,\,k+1}^{AUG+} = g_i(\chi_{i,\,k+1}^{AUG}) \quad (۴۵)$$

$$\hat{X}_{k+1}^{+} = \sum_{i=0}^{2N} w_i^{m}\, \chi_{i,\,k+1}^{AUG+} \quad (۴۶)$$

$$P_{k+1}^{+} = \sum_{i=0}^{2N} \sum_{j=0}^{2N} w_{ij}^{c}\, (\chi_{i,\,k+1}^{AUG+}).(\chi_{j,\,k+1}^{AUG+})^{T} \quad (۴۷)$$

که در معادله (۴۵)، g مدل اضافه سازی بردار حالت جدید است. در بخش بعدی، ما با ارائه نتایج شبیه سازی برتری الگوریتم‌های IEKF و SPKF را برای حل مسأله SLAM نسبت به الگوریتم EKF نشان خواهیم داد.

۵- نتایج شبیه سازی

در این بخش، برای مقایسه عملکرد نسخه‌های اصلاح شده فیلتر کالمن توسعه یافته برای حل مسأله SLAM، نتایج شبیه‌سازی را ارائه می‌کنیم. برای شبیه سازی یک ربات با درایو تفاضلی را در نظر می‌گیریم. مدل پروسه درایو تفاضلی به صورت زیر است:

$$\hat{X}_R(k+1) = \hat{X}_R(k) + \begin{bmatrix} \frac{\Delta s_r + \Delta s_l}{2} \cos(\hat{\theta}_R(k) + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r + \Delta s_l}{2} \sin(\hat{\theta}_R(k) + \frac{\Delta s_r - \Delta s_l}{2b}) \\ \frac{\Delta s_r - \Delta s_l}{b} \end{bmatrix} \quad (۴۸)$$

که در آن b فاصله میان دو چرخ درایو تفاضلی است. Δs_l و Δs_r نیز به ترتیب مسافتهای طی شده برای چرخ‌های سمت چپ و راست هستند. در اینجا ورودی کنترلی میزان جابجایی چرخها در نظر گرفته شده است. در این شبیه سازی، برای ورودی کنترلی نیز نویز در نظر می‌گیریم و ماتریس کوواریانس آن را به صورت زیر محاسبه می کنیم:

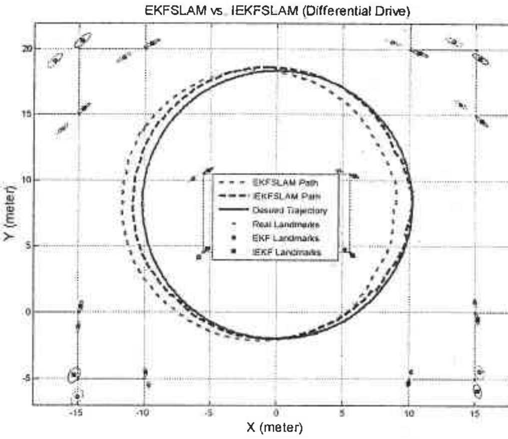
$$U = \mathrm{cov}(\Delta s_l, \Delta s_r) = \begin{bmatrix} k_r \Delta s_r & 0 \\ 0 & k_l \Delta s_l \end{bmatrix} \quad (۴۹)$$

که k_l و k_r ثابتهای خطا هستند که نشان دهنده پارامترهای غیر اُدومتری درایو موتور و تعامل بین چرخها و زمین هستند. یک محیط داخل ساختمان با ابعاد ۲۹ × ۳۴ را شبیه‌سازی کرده‌ایم. در اینجا گوشه‌ها و لبه‌ها را بعنوان مشخصه‌های محیطی در نظر می‌گیریم. ربات مجهز به یک حلقه سونار پلاروید است که به کمک روش *ترکیب مبتنی بر سه گوش‌سازی*[1] (TBF) می‌تواند مختصات و ماتریس کوواریانس مشاهده را برای مشخصه‌های لبه و گوشه بدست آورد. ربات با حرکت روی یک مسیر دایره‌ای، از الگوریتم SLAM استفاده می‌کند تا حالت و موقعیت مشخصه‌های محیط را تخمین بزند و نقشه محیط را بسازد. برای مسأله همبستگی داده‌ها در الگوریتم SLAM نیز از روش فیلتر استاندارد نزدیکترین همسایگی (NNSF) استفاده کرده‌ایم.

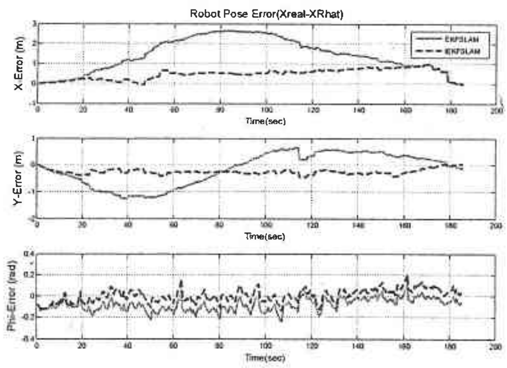
ابتدا تأثیر خطی سازی تحلیلی مجدد در فیلتر کالمن توسعه یافته را بررسی می‌کنیم. شکل ۲ مسیر و نقشه تخمین زده شده توسط دو فیلتر EKF و IEKF را نشان می‌دهد. همانطور که مشخص است، فیلتر کالمن توسعه یافته تکراری دقت تخمین را در الگوریتم SLAM فوق‌العاده بهبود داده است. برای مقایسه دقت تخمین، خطای تخمین حالت ربات توسط هر دو فیلتر را در شکل ۳ آورده‌ایم. جدول ۱ نیز دقت تخمین را بر حسب میانگین مربعات خطا برای هر دو فیلتر نشان می‌دهد. همچنانکه ملاحظه می‌شود، میانگین مربعات خطا برای فیلتر کالمن توسعه یافته تکراری کمتر بوده و بهبود دقت تخمین در نسخه تکراری EKF با نتایج شبیه‌سازی تأیید می‌شود.

حال تأثیر خطی سازی آماری موجود در فیلتر کالمن نقطه سیگما را برای بهبود دقت تخمین در الگوریتم SLAM بررسی می‌کنیم.

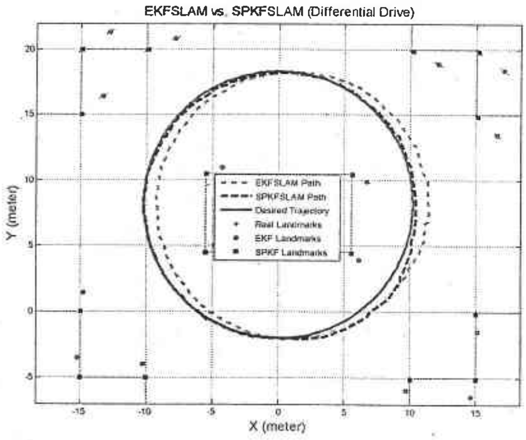
۱ Triangulation Based Fusion (TBF)



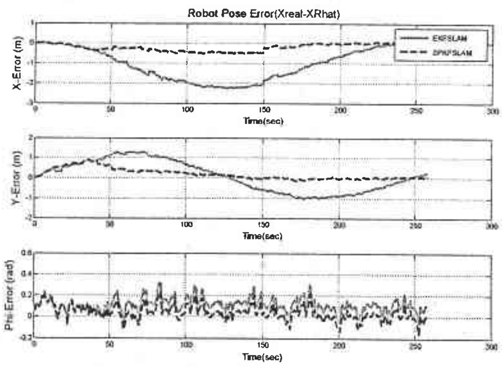
شکل ۲: نقشه نهایی بدست آمده از الگوریتم EKFSLAM و IEKFSLAM . مسیر تخمین زده شده توسط IEKF با خط چین نشان داده شده است.



شکل ۳: مقایسه خطای تخمین حالت ربات برای الگوریتم EKFSLAM و IEKFSLAM . خطای تخمین فیلتر کالمن توسعه یافته تکراری با خط چین نشان داده شده است.



شکل ۴: نقشه نهایی بدست آمده از الگوریتم EKFSLAM و SPKFSLAM . مسیر تخمین زده شده توسط SPKF با خط چین نشان داده شده است.



شکل ۵: مقایسه خطای تخمین حالت ربات برای الگوریتم EKFSLAM و SPKFSLAM . خطای تخمین فیلتر کالمن نقطه سیگما با خط چین نشان داده شده است.

شکل ۴ نقشه و مسیر تخمین زده شده توسط الگوریتم‌های EKFSLAM و SPKFSLAM را نشان می‌دهد. مسیر تخمین زده شده توسط فیلتر کالمن نقطه سیگما با خط چین نمایش داده شده است و مسیر تخمین زده شده توسط فیلتر کالمن توسعه یافته با نقطه چین مشخص شده است. مشخصه‌های تخمین زده شده به کمک فیلتر کالمن نقطه سیگما با مربع نشان داده شده‌اند و تقریباً روی محل واقعی مشخصه‌ها قرار گرفته‌اند. برتری فیلتر کالمن نقطه سیگما از نتایج شبیه‌سازی کاملاً مشخص است. خطای تخمین حالتهای ربات در شکل ۵ آورده شده است. جدول ۲ نیز دقت تخمین را بر حسب میانگین مربعات خطا برای هر دو فیلتر نشان می‌دهد.

بنابراین، بهبود دقت تخمین در فیلتر کالمن نقطه سیگما برای حل مسأله SLAM نیز با نتایج شبیه‌سازی تأیید شده است. نتایج جداول ۱ و ۲ بطور کمی برتری نسخه‌های اصلاح شده فیلتر کالمن توسعه یافته را نشان می‌دهند. توجه کنید که در این شبیه‌سازی، ما برای اینکه نتایج را بطور واضح نشان دهیم، ماتریس های کوواریانس پروسه و مشاهده را بزرگ انتخاب نموده‌ایم. وگرنه، دقت تخمین هر دو فیلتر نسبتاً بالاست. ولی همگرایی و سازگاری نسخه‌های اصلاح شده نسبت به فیلتر کالمن توسعه یافته فوق‌العاده بیشتر است. این نسخه‌ها اصلاح شده می‌توانند جایگزین‌های خوبی برای تخمین‌گر کالمن در آینده به شمار روند.

جدول ۱: مقایسه عملکرد و دقت تخمین حالت‌های ربات بر حسب میانگین مربعات خطا برای الگوریتم EKFSLAM و IEKFSLAM.

میانگین مربعات خطا	حالت x	حالت y	حالت φ
EKFSLAM	۲/۷۴۶۸	۰/۴۱۲۱	۰/۰۱۰۳
IEKFSLAM	۰/۳۰۱۲	۰/۰۷۴۰	۰/۰۰۳۹

جدول ۲: مقایسه عملکرد و دقت تخمین حالت‌های ربات بر حسب میانگین مربعات خطا برای الگوریتم EKFSLAM و SPKFSLAM.

میانگین مربعات خطا	حالت x	حالت y	حالت φ
EKFSLAM	۱/۶۹۳۷	۰/۵۵۱۸	۰/۰۱۶۹
SPKFSLAM	۰/۰۷۶۹	۰/۰۹۹۵	۰/۰۰۶۱

۶- نتیجه‌گیری و تحقیقات آینده

هدف از ارائه این مقاله جستجوی تخمین‌گرهای مناسب‌تر و سازگارتر برای حل مسائل مهم ناوبری نظیر الگوریتم SLAM است. نسخه‌های اصلاح شدهٔ فیلتر کالمن توسعه یافته، ضعف این فیلتر در خطی سازی مدلها را نسبتاً پوشش می‌دهند. این فیلترها دقت تخمین و سازگاری الگوریتم SLAM را برای ساخت نقشه‌های هندسی از محیط و استفاده توأم آنها برای موقعیت یابی سیستم متحرک خودمختار افزایش می‌دهند. نتایج شبیه‌سازی ارائه شده در بخش قبل نیز تأیید کرد که با خطی سازی تحلیلی مجدد و با خطی سازی آماری در نسخه‌های اصلاح شده می‌توان همگرایی الگوریتم را افزایش داد.

نویسندگان این مقاله در حال پیاده‌سازی الگوریتم SLAM با روشهای مذکور در پژوهشکده الکترونیک دانشگاه علم و صنعت ایران هستند تا نتایج شبیه‌سازی را با پیاده‌سازی واقعی الگوریتم بر روی یک ربات متحرک خودمختار تأیید کنند. بعلاوه، سعی بر آن است تا تکرار در خطی سازی آماری موجود در فیلتر کالمن نقطه سیگما، برای بهبود دقت تخمین در این فیلتر بررسی شود و نسخه اصلاح شده دیگری به نام فیلتر کالمن نقطه سیگمای تکراری نیز برای حل مسأله SLAM ارائه شود.

مراجع

[1] M. W. M. G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, nd M. Csorba, "A solution to the simultaneous localization and map building (SLAM) roblem," IEEE Trans. Robot. Automat., vol.17 , no.3 ,pp.229 – 241, June 2001..

[2] T. Bailey. "Mobile Robot Localisation and Mapping in Extensive Outdoor Environments." PhD thesis, University of Sydney, Australian Centre for Field Robotics, 2002.

[3] Julier, Simon J. and Jeffery K. Uhlmann. "A New Extension of the Kalman Filter to Nonlinear Systems." In The Proceedings of AeroSense: The 11th International Symposium on Aerospace/Defense Sensing, Simulation and Controls, Multi Sensor Fusion, Tracking and Resource Management II, SPIE, 1997.

[4] R. van der Merwe. "Sigma-Point Kalman Filters for Probabilistic Inference in Dynamic State-Space Models." PhD thesis, OGI School of Science & Engineering, Oregon Health & Science University, April 2004.

[5] E. A. Wan and R. van der Merwe. "The unscented kalman filter for nonlinear estimation." In Symposium 2000 on Adaptive Systems for Signal Processing, 2000.

[6] Yaakov Bar-Shalom X.-Rong Li Thiagalingam Kirubarajan,"Estimation with Applications To Tracking and Navigation," John Wiley and Sons Inc. 2001.

[7] Rudy Negenborn. "Robot Localization and Kalman Filters on finding your position in a noisy world." M.S thesis, Utrecht University, September 2003.