

Observability-Constraint EKF for SLAM Navigation Systems; Design and Comparison

Yusef Khayaat¹, Saeed Khankalantary²

¹ Ph.D., Faculty of Electrical Engineering, University of Kurdistan, Sanandaj, Iran

y.khayat@uok.ac.ir

² Assistant Prof., Faculty of Electrical Engineering, K. N. Toosi University of Technology, Tehran, Iran

s.kalantary@kntu.ac.ir

Abstract:

Recently, the use of different visual navigation algorithms as a navigation-aided system has become of interest by many researchers. In most of these approaches, to improve the accuracy of the navigation problem, undesired predicted landmarks are used again in comparison with the output of the algorithm by appropriate filtering units for fusion and integration in the system. However, the number of points with a specific feature, the amount of noise, and the presence or absence of color noise have a direct impact on the performance of the conventional filters such as extended Kalman filter (EKF). In this paper, in addition to demonstrating how to model SLAM-based video navigation systems and systematically design and implement filtering unit, an observability-constrained EKF (OC-EKF) has been proposed for improving the performance of the system and minimizing the system errors. The main contribution of the proposed scheme is that it has provided the SLAM filtering mechanism as a constrained optimization issue that considers the observability constraints in the estimation problem. Besides, the performance of OC-EKF is compared with the standard EKF (STD-EKF) and the first estimate Jacobean EKF (FEJ-EKF), H_{inf} -based Kalman Filter (KF), and Robust KF; where, the robust performance of the proposed scheme against uncertainties, white and color noises have been verified through comprehensive simulation results. As shown in the Monte Carlo simulation results, the proposed OC-EKF not only provides more accurate navigation, but also behaves more robustly against system uncertainties and noises.

Keywords: Navigation systems, Visual-Inertial odometry, Data fusion, Extended kalman filter, Simultaneous localization and mapping (SLAM).

Article Type: Research

Received: 01. May. 2022

Revised: 24. Dec. 2022

Accepted: 19. April. 2023

Corresponding author: Saeed Khankalantary

Corresponding author's address: Faculty of Electrical Engineering, KN Toosi University of Technology, Tehran, Iran



1. Motivation of the work

Appropriate performance of SLAM-based navigation systems in noisy environments, or environments where the number of feature points is significantly reduced, is severely affected and the proper filtering method to integrate information has a significant impact on the performance of the system navigation. To do so, in this article, after proposing an in-detailed modeling of SLAM-based navigation systems, appropriate filtering in noisy system to improve the performance of the system is introduced as the main motivation of the work.

2. Contributions

As one of the main contribution of this article, we have shown how observability constraints can be applied to SLAM-based navigation systems. Although the main feature of the FEJ-EKF filtering method is to improve the performance, but this method is not robust and is also dependent on the initial estimation value. While the OC-EKF filter optimizes the expected square error of the linearization points while satisfying the observability conditions. In fact, the most important contribution of the proposed scheme is that it transforms the filtering problem into a constrained optimization problem that includes observability constraints in the estimation problem.

3. Procedures

SLAM includes the following three basic operations that are repeated at each step time:

- 1) When a robot moves and reaches a new point of the scene, due to the noises and unavoidable errors, this movement increases the uncertainty in the accurate positioning. A viable solution requires a mathematical model for this step, which is called a movement model.
- 2) Through a series of image processing processes, the robot processor has discovered interesting features from the environment that should be included in the map. We call these features signs, marks, or points with special features. Due to the errors in external sensors, the location of these signs is unclear. Furthermore, since the location of the robot is currently uncertain, these two uncertainties must be properly combined. A viable solution at this stage requires a mathematical model to determine the position of points with special characteristics, which is called the inverse observation model.
- 3) The robot observes the points with special features that have already been mapped and uses them to correct its positioning by considering all the points with special features in its visible space; Therefore, in this case, both localization and signal uncertainty are reduced. A viable solution is the need for a mathematical model to predict the measured values from the predicted marked location and the positioning of the robot, which is called direct observation.

By using these three models plus an estimation and filtering unit for data fusion, we can create a solution for the proper use of visual navigation systems.

4. Findings

In addition to showing how to accurately model SLAM navigation systems and systematically implement their filtering, to compare the performance of the STD-EKF, the generalized Kalman filter based on the Jacobian first estimate. FEJ-EKF, H_{inf} -based Kalman filter, robust Kalman filter and the proposed method of observable-constrained extended Kalman filter OC-EKF are discussed. As shown in the results of the Monte Carlo simulations, the OC-EKF filter is more resistant to the uncertainties of the system by maintaining better performance and compared to other nonlinear filters investigated in this research, and even with various uncertainties, it has provided better accuracy for accurate navigation of the complex.

5. Conclusion

This article deals with the design of OC-EKF in order to increase the accuracy of navigation systems based on SLAM. The usual filtering method of these systems is based on linear filters or the standard Kalman filter. Although the EKF filter has been introduced as one of the most appropriate solutions to improve the data fusion of the SLAM-based navigation systems, however, the number of marks with special characteristics, the amount of noise, and the presence or absence of color noise have a direct impact on the performance of conventional EKF.

فیلتر کالمن تعمیم یافته مشاهده پذیر مقید برای سامانه های ناوبری SLAM؛ طراحی و مقایسه

یوسف خیاط^۱، سعید خان کلانتری^۲

۱- فارغ التحصیل دکتری- دانشکده مهندسی برق- دانشگاه کردستان- سنندج- ایران

y.khayat@uok.ac.ir

۲- استادیار- دانشکده مهندسی برق- دانشگاه صنعتی خواجه نصیرالدین طوسی- تهران- ایران

s.kalantary@kntu.ac.ir

چکیده:

اخیراً، استفاده از الگوریتم های متفاوت ناوبری تصویری به عنوان یک سیستم کمک ناوبری مورد علاقه بسیاری از محققان بوده است. در اکثر این رویکردها، به منظور بهبود دقت مساله ناوبری، نقاط ویژگی پیشگویی شدهی نامطلوب در مقایسه با خروجی الگوریتم دوباره توسط واحدهای فیلترینگ مناسب جهت ادغام و تلفیق در سامانه استفاده می شوند. با این حال، تعداد نقاط با ویژگی خاص، میزان نویز و وجود یا عدم وجود نویز رنگی تاثیر مستقیمی بر روی کارایی فیلترهای مرسوم همانند فیلتر کالمن تعمیم یافته دارد. به همین منظور در این مقاله، علاوه بر نمایش نحوه مدل سازی سامانه های ناوبری تصویری و پیاده سازی سیستماتیک چگونگی فیلترینگ آن ها، فیلتر کالمن تعمیم یافته مشاهده پذیر مقید به منظور بهبود عملکرد سامانه ناوبری و کمینه ساختن خطاهای سامانه ارائه شده است. مهمترین نوآوری طرح پیشنهادی این است که مساله فیلترینگ را به عنوان یک مسئله بهینه سازی مقید که قیود مشاهده پذیری را در مساله تخمین لحاظ می کند، تبدیل کرده است. علاوه بر آن، عملکرد فیلتر کالمن تعمیم یافته مشاهده پذیر با فیلتر کالمن تعمیم یافته استاندارد، فیلتر کالمن مبتنی بر ژاکوبین نخستین تخمین، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم نیز مقایسه شده است و عملکرد مقاوم فیلتر ارائه شده در مقابل با عدم قطعیت ها، نویزهای سفید و رنگی به وسیله نتایج شبیه سازی نمایش داده شده است. همچنان که در نتایج مونت کارلوی شبیه سازی ها هم نشان داده شده است، فیلتر کالمن مشاهده پذیر مقید نه تنها ناوبری دقیق تری را نتیجه می دهد، بلکه نسبت به نامعینی ها و نویزهای سیستم مقاوم تر نیز می باشد.

کلمات کلیدی: سامانه های ناوبری، ناوبری بصری-اینرسی، تلفیق اطلاعات، فیلتر کالمن تعمیم یافته، مکان یابی و نقشه برداری همزمان

نوع مقاله: پژوهشی

دریافت: ۱۴۰۱/۰۲/۱۱

بازنگری: ۱۴۰۱/۱۰/۰۳

پذیرش: ۱۴۰۲/۰۱/۳۰

نام نویسندهی مسئول: سعید خان کلانتری

نشانی نویسندهی مسئول: دانشگاه صنعتی خواجه نصیرالدین طوسی - تهران - ایران

۱- مقدمه

امروزه سیستم‌های ناوبری بصری-اینرسی، به دلیل فراگیری بیش از پیششان، در طیف وسیعی از کاربردهایی همانند واقعیت افزوده تلفن همراه تا ناوبری هوایی و رانندگی خودکار بکار گرفته شده‌اند. در این روش‌ها، هدف ترکیب اطلاعات چندین حسگر و رسیدن به سیستمی است که کارایی آن از تک‌تک حسگرها بیشتر بوده و بتواند با دقت بیشتری، اهداف ناوبری مدنظر طراح را ارضا نماید [۱]. سیستم‌های بکار گرفته‌شده و یا در حال توسعه و تحقیق در زمینه ناوبری تصویری در سامانه‌های مختلف ناوبری را می‌توان بر اساس مسئله‌ی داشتن، نداشتن یا نحوه تشکیل نقشه مرجع، شرایط محیطی که در آن مورد بهره‌برداری قرار می‌گیرند و یا حتی نوع دوربین و کاربرد آن دسته‌بندی نمود [۲]. در یک دسته‌بندی دیگر می‌توان به سیستم‌های ناوبری تصویری مبتنی بر شار نوری، ناوبری تصویری بر اساس ظاهر، ناوبری تصویری بر اساس استخراج ویژگی‌های کیفی تصویر و تفسیر آن از محیط مورد جستجو، ناوبری تصویری بر پایه ردیابی ویژگی‌ها و ناوبری تصویری بر اساس ادومتری تصویری اشاره کرد [۳-۴].

در ناوبری تصویری مبتنی بر نقشه‌های توپولوژیکی، متخصصین پردازش تصویر با استفاده از یکسری صفات از گوشه‌ها و محل اتصال گره‌ها، ارتباط بین گره‌ها و خطوط تصویر را توسط الگوریتم‌های خاص پردازش تصویر استخراج می‌کنند [۲]، که تصویر حاصل معرف نقشه محیط پیرامون بوده و ارتباط بین صفات بر روی تصویر به صورت یک نقشه جامع در خواهد آمد. به همین منوال نیز تکنیک‌های مبتنی بر تطبیق ظواهر بر پایه ذخیره تصاویر قبلی استوار بوده به‌صورتی که تصاویر ذخیره‌شده به‌عنوان الگو به کار رفته و وسیله نقلیه مورد مطالعه، تعیین موقعیت و ناوبری خود را بر اساس تطبیق تصویر فعلی با تصاویر الگو انجام خواهد داد [۴]. در مرجع [۵] مروری بر روش‌های تطبیق تصاویر بر اساس استخراج ویژگی انجام شده است. یکی از معروف‌ترین این الگوریتم‌ها، الگوریتم SIFT^۱ است که به تحلیل ویژگی‌های مقاوم در برابر تغییرات روشنایی و تغییرات مقیاس می‌پردازد [۶]. الگوریتم SIFT و مشتقات آن بعدها مرجع بسیاری از کارهای تحقیقاتی در زمینه تطبیق و پردازش تصاویر گردید. نویسندگان در مرجع [۷] الگوریتم SIFT بهبود داده‌شده را ارائه داده‌اند، به‌صورتی که نقاط تطبیق اشتباه را به کمک موقعیت مکانی‌های ثابت در تصویر و از قبل تعریف کردن نقاط کلیدی موقعیت‌های ثابت تصویر، بتوان به راحتی تشخیص داد.

اساس روش‌های ادغام در سامانه‌های تصویری با سامانه‌های ناوبری مبتنی بر عوارض زمین، ردیابی مشخصه‌های تصویر و هموگرافی محیط به‌وسیله سامانه‌ی تصویری و سپس تلفیق مشاهدات تصویر و عوارض زمین با هم به‌وسیله یک روش فیلترینگ مناسب است [۸]. در این روش‌ها، چنانچه بیش از سه نقطه بین دو تصویر مرجع و تصویر دریافت شده توسط دوربین نصب‌شده بر روی هواپیما، تطبیق داده شود، موقعیت سه بعدی و وضعیت دوربین را می‌توان به کمک

الگوریتم‌های فتوگرامتری محاسبه کرد [۹-۱۰]. به بیانی دیگر، می‌توان گفت که ابتدا تطبیق صحنه بین تصویر لحظه‌ای و تصویر مرجع که داده‌های ارتفاع را نیز دارد انجام شده و در مرحله بعد به کمک چندین نقطه به‌دست‌آمده از خروجی تطبیق و با استفاده از الگوریتم فتوگرامتری، موقعیت سه بعدی و وضعیت در دستگاه مختصات جغرافیایی محاسبه می‌شود، تا سرعت‌های خطی و زاویه‌ای را بوسیله فیلترینگ مناسب محاسبه کرده و در صورت نیاز شتاب‌ها را نیز به دست آورد [۱۱].

فرآیند تخمین تغییر موقعیت یک وسیله به کمک ورودی یک یا چند دوربین که روی وسیله قرار گرفته‌اند، ادومتری تصویری می‌باشد. ادومتری تصویری به کمک بررسی تغییرات حرکت به‌وسیله تصاویر دریافتی از دوربین، موقعیت وسیله را به تدریج تخمین می‌زند. البته لازم به ذکر است که مسئله تعیین وضعیت نسبی دوربین و ایجاد یک ساختار سه بعدی از مجموعه تصاویر دوربین در بحث بینایی ماشین با نام ایجاد ساختار از حرکت شناخته می‌شوند [۱۲-۱۳]، که ادومتری تصویری حالت خاصی از ساختار حرکت می‌باشد. ساختار حرکت در فرم کلی مربوط به چگونگی ساخت صحنه سه بعدی از تصاویر دوربین در موقعیت‌های مختلف و متوالی مرتبط و نامرتبط می‌باشد. استخراج ساختار نهایی اغلب به‌وسیله روش‌های مختلف بهینه‌سازی صورت گرفته که بار محاسباتی آن با افزایش تعداد تصاویر نیز افزایش می‌یابد [۱۴].

از معروف‌ترین روش‌های ناوبری تصویری، روش ناوبری تصویری مبتنی بر مکان‌یابی و نقشه‌برداری همزمان و یا SLAM^۲ می‌باشد. هدف اصلی از این روش ناوبری تهیه نقشه محیط، بدون داشتن هیچ دانش اولیه‌ای نسبت به همان محیط و موقعیت‌یابی وسیله در همان نقشه به‌صورت هم‌زمان می‌باشد. مزیت اصلی ناوبری SLAM این بوده که احتیاجی به نقشه‌های از پیش تعیین‌شده برای عمل انطباق نداشته و وسیله بدون داشتن هیچ اطلاع قبلی از محیط قادر خواهد بود مسیر و موقعیت خود را تشخیص دهد. در [۱۵] با ادغام فیلتر کالمن تعمیم‌یافته با الگوریتم‌های فازی، سعی در بهبود دقت این سامانه‌ها داشته است. منبع [۱۶] نحوه پیاده سازی فیلتر کالمن خطی برای این سامانه‌ها را نمایش داده است. در مرجع [۱۷] ترکیب ناوبری اینرسی و دوربین با دو رویکرد تشکیل نقشه و بدون تشکیل نقشه بررسی شده است و در ادامه به پیاده سازی فیلتر کالمن تعمیم‌یافته برای سیستم SLAM پرداخته است. مسئله بهبود ناوبری در این سامانه‌ها با پیاده سازی فیلتر کالمن بی‌اثر در مرجع [۱۸] دنبال شده است. همچنین در [۱۹] مقایسه‌ای جامع بین فیلتر کالمن و فیلتر کالمن تعمیم‌یافته برای محیط‌های نویری انجام شده است.

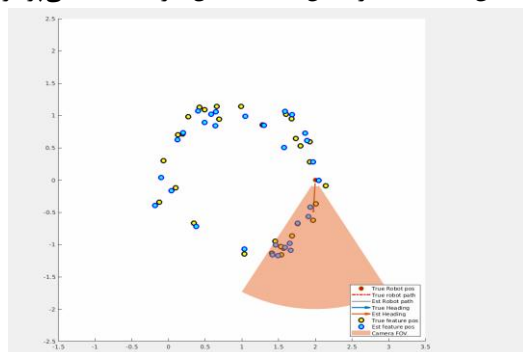
عملکرد درست و دقیق سامانه‌های مبتنی بر SLAM در محیط‌های نویری و یا محیط‌هایی که تعداد نقاط ویژگی به طور محسوسی کاهش یافته، به شدت تحت تاثیر قرار گرفته و نحوه فیلترینگ مناسب جهت تلفیق اطلاعات تاثیر بسزایی در ناوبری دقیق آن‌ها دارد. به همین

مشاهده معکوس می‌نامند.

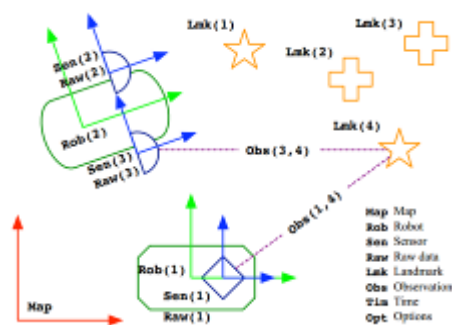
(۳) ربات نقاط با ویژگی‌های خاص که قبلاً نقشه‌برداری شده است را مشاهده کرده و از آن‌ها برای تصحیح مکان‌یابی خود با در نظر گرفتن همه نقاط با ویژگی‌های خاص در فضای قابل مشاهده‌اش استفاده می‌کند؛ بنابراین، در این مورد، هر دو عدم قطعیت مکان‌یابی و نشانه‌ای کاهش می‌یابد. یک راه‌حل شدنی نیاز به یک مدل ریاضی برای پیش‌بینی مقادیر اندازه‌گیری از محل علامت‌گذاری شده پیش‌بینی شده و مکان‌یابی ربات است، که این مدل را مشاهده مستقیم می‌نامند.

با استفاده از این سه مدل به‌علاوه یک واحد تخمین و فیلترینگ می‌توان راه‌حلی برای بهره‌برداری مناسب از سامانه‌های ناوبری تصویری بسازیم. واحد تخمین و فیلترینگ مسئول انتشار مناسب عدم قطعیت‌ها در هر بار رخ دادن یکی از سه حالت فوق است. شکل (۱) مکان‌یابی ربات با در نظر گرفتن نقاط با ویژگی‌های خاص در فضای قابل مشاهده‌اش بوسیله SLAM را نمایش می‌دهد، که در ادامه بیشتر بر آن تمرکز می‌کنیم.

همچنان که از شکل (۲) قابل مشاهده است، واحدهای مربوط به یک سامانه‌ی ناوبری مبتنی بر SLAM عبارت‌اند از نقشه، سنسورها، نقاط به‌دست‌آمده از واحد پردازش تصویر که بیانگر ویژگی‌های خاص هستند، مشاهده‌گرها، و واحد تخمین. در بخش بعد و با استفاده از تصویر نشان داده شده در شکل (۲)، به مدل‌سازی سامانه می‌پردازیم.



شکل (۱): مکان‌یابی ربات با در نظر گرفتن نقاط با ویژگی‌های خاص در فضای قابل مشاهده‌اش



شکل (۲): واحدهای مربوط به یک سامانه‌ی ناوبری مبتنی بر SLAM

منظور در این مقاله بعد از مدل‌سازی دقیق این سامانه‌ها، به بررسی نحوه پیاده‌سازی فیلترینگ آن‌ها می‌پردازیم. در ادامه به پیاده‌سازی فیلتر کالمن تعمیم‌یافته استاندارد^۳ (STD-EKF)، فیلتر کالمن تعمیم‌یافته مبتنی بر ژاکوبین نخستین تخمین^۴ (FEJ-EKF) و ارائه فیلتر کالمن تعمیم‌یافته مشاهده‌پذیر مقید^۵ (OC-EKF) می‌پردازیم و به عنوان یکی از اصلی‌ترین نوآوری‌های این مقاله نشان خواهیم داد چگونه می‌توان قیود مشاهده‌پذیری را در قالب یک الگوریتم به سامانه‌های SLAM گسترش داد. هرچند ویژگی اصلی روش فیلترینگ FEJ-EKF بهبود عملکرد در عین حال محاسبات ساده‌تر می‌باشد، اما این روش زیاد مقاوم نبوده و به مقدار اولیه تخمین نیز بسیار وابسته می‌باشد. در حالی که فیلتر OC-EKF به بهینه‌سازی خطای مربعی مورد انتظار نقاط خطی‌سازی در عین برآوردن شرایط مشاهده‌پذیری می‌پردازد. در واقع مهم‌ترین نوآوری طرح پیشنهادی این است که در روش پیشنهادی، مساله فیلترینگ را به‌عنوان یک مسئله بهینه‌سازی مقید که قیود مشاهده‌پذیری را در مساله تخمین لحاظ می‌کند تبدیل می‌نماید. همچنان که در شبیه‌سازی‌ها نیز به مقایسه فیلتر پیشنهادی با فیلترهای FEJ-EKF، STD-EKF، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم پرداخته شده است، فیلتر ارائه شده OC-EKF علاوه بر بهبود ناوبری سیستم، عملکرد مقاوم‌تر و بهتری در مقایسه با فیلترهای بررسی شده دارد.

۲- سامانه‌های ناوبری SLAM

همچنان که گفته شد، سامانه‌های ناوبری مبتنی بر SLAM از مهم‌ترین نوع سامانه‌های تصویری بوده که به‌منظور ناوبری با اهداف مختلف تجاری و یا نظامی می‌توانند بکار گرفته شوند. با این حال، همانند دیگر سامانه‌های ناوبری، به‌منظور ادغام و یا افزایش دقت ناوبری نیازمند فیلترینگ مناسبی می‌باشند. SLAM شامل سه عملیات اساسی زیر بوده که در هر مرحله زمانی تکرار می‌شوند:

(۱) هنگامی که ربات یا پرند حرکت کرده و به نقطه جدیدی از صحنه می‌رسد. به دلیل نویزها و خطاهای اجتناب‌ناپذیر، این حرکت عدم قطعیت در مکان‌یابی دقیق ربات را افزایش می‌دهد. یک راه‌حل شدنی نیازمند مدل ریاضی برای این مرحله بوده که آن را مدل حرکت می‌نامند.

(۲) به‌وسیله‌ی یک سری از فرایندهای پردازش تصویری، پردازنده‌ی ربات ویژگی‌های جالبی را از محیط کشف کرده که باید در نقشه گنجانده شوند. ما این ویژگی‌ها را نشانه‌ها یا نقاط با ویژگی خاص می‌نامند. به دلیل خطاها در حسگرهای بیرونی، مکان این نشانه‌ها نامشخص است. علاوه بر این، از آنجاکه مکان ربات در حال حاضر نامشخص است، این دو عدم قطعیت باید به‌درستی ترکیب شوند. یک

راه‌حل شدنی در این مرحله نیازمند یک مدل ریاضی برای تعیین موقعیت نقاطی با ویژگی‌های خاص در صحنه بوده که از داده‌های به‌دست‌آمده توسط سنسورها قرار است مشخص شود، که آن را مدل

۳- مدل سازی سامانه

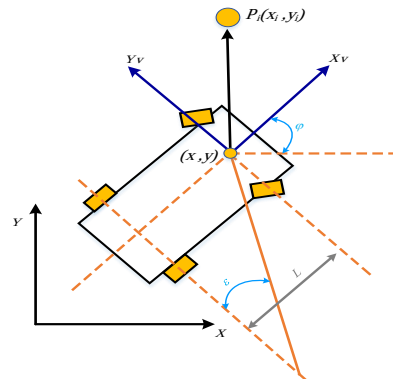
به منظور مدل سازی مناسب سامانه، ابتدا دینامیک سیستم را بدست آورده، سپس مدل مشاهدات را استخراج کرده و در آخر به تعریف متغیرهای حالت می پردازیم.

۳-۱- دینامیک سیستم

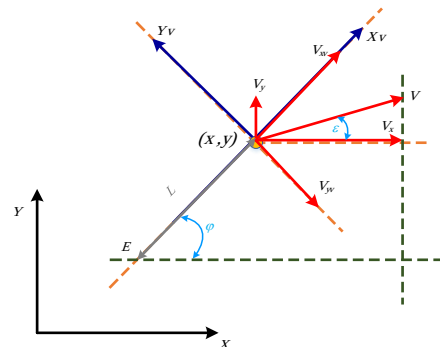
شکل (۳) مدل یک ربات را نشان می دهد که می تواند نشانگر یک ماشین متحرک مستقل در یک صفحه ی دو بعدی باشد. چرخ های جلو آن می توانند عملکرد فرمان را انجام دهند و چرخ های عقب چرخ های متحرک ثابتی هستند. سیستم مختصاتی سنسور با وسیله نقلیه متحرک سازگار است و می تواند علائم ویژه ای را برای به دست آوردن داده های حسگر مشاهده کند. بر اساس مدل سینماتیکی نشان داده شده، وضعیت وسیله نقلیه به صورت

$$x_v = [x \ y \ \varphi]^T \quad (1)$$

تعریف می شود که در آن x_v وضعیت وسیله نقلیه، x موقعیت در راستای محور افقی، y موقعیت در راستای محور عمودی، φ زاویه بردار حرکت با محور افقی می باشد. ورودی کنترل زاویه سکان γ و سرعت V است که معادل زاویه فرمان و دریچه گاز وسیله نقلیه است.



شکل (۳): مدل یک ربات متحرک مستقل در یک صفحه ی دو بعدی



شکل (۴): نمایش زاویه سکان در ربات

برای سهولت در تحقیق، مدل بدنه خودرو در شکل (۳) را می توان به میله ای به طول L مدل کرد. همان طور که در شکل (۴) نشان داده شده است سرعت چرخ ها در دو طرف قسمت جلویی به نقطه

مرکزی V نسبت داده می شود، علاوه بر این مشکل مثبت و منفی زاویه سکان است را می توان به صورت زیر و به راحتی در مدل سازی، جای داد.

با توجه به شکل (۴) هنگامی که زاویه سکان منفی است، مدل سینماتیک وسیله نقلیه خودران مدل سازی شده است. تجزیه سرعت V به جهات عرضی و طولی، روابط زیر را نتیجه خواهد داد

$$\begin{aligned} \dot{x} &= V_x = V \cos(\varepsilon) \\ \dot{y} &= V_y = V \sin(\varepsilon) \end{aligned} \quad (2)$$

که ε مجموع زوایای سکان و حرکت بوده و برابر است با:

$$\varepsilon = \varphi + \gamma \quad (3)$$

پس می توان نوشت که:

$$\begin{aligned} \dot{x} &= V_x = V \cos(\varphi + \gamma) \\ \dot{y} &= V_y = V \sin(\varphi + \gamma) \end{aligned} \quad (4)$$

که با ادغام دو معادله فوق می توان معادله حرکت در دو بعد را به فرم زیر بازنویسی کرد:

$$\begin{cases} \dot{x} = V \cos(\varphi + \gamma) \\ \dot{y} = V \sin(\varphi + \gamma) \\ \dot{\varphi} = \frac{V \sin(\gamma)}{L} \end{cases} \quad (5)$$

که فرم گسسته با گام گسسته سازی ΔT آن را اگر به فرم اندازه گیری با نویز بازنویسی کنیم، خواهیم داشت:

$$\begin{cases} x(k+1) = x(k) + \Delta T V(k+1) \cos(\varphi(k) + \gamma(k+1)) + q_x(k+1) \\ y(k+1) = y(k) + \Delta T V(k+1) \sin(\varphi(k) + \gamma(k+1)) + q_y(k+1) \\ \varphi(k+1) = \varphi(k) + \Delta T \frac{V(k+1) \sin(\gamma(k+1))}{L} + q_\varphi(k+1) \end{cases} \quad (6)$$

و همچنین فرم برداری آن به شکل زیر خواهد بود:

$$\begin{aligned} \begin{bmatrix} x(k+1) \\ y(k+1) \\ \varphi(k+1) \end{bmatrix} &= \begin{bmatrix} x(k) \\ y(k) \\ \varphi(k) \end{bmatrix} + \begin{bmatrix} \Delta T V(k+1) \cos(\varphi(k) + \gamma(k+1)) \\ \Delta T V(k+1) \sin(\varphi(k) + \gamma(k+1)) \\ \Delta T \frac{V(k+1) \sin(\gamma(k+1))}{L} \end{bmatrix} \\ &+ \begin{bmatrix} q_x(k+1) \\ q_y(k+1) \\ q_\varphi(k+1) \end{bmatrix} \end{aligned} \quad (7)$$

به منظور نزدیک شدن به مدل واقعی و برای سهولت شبیه سازی، می-

در نهایت معادله سیستم را به فرم زیر بازنویسی کرد

$$\begin{aligned} x_t &= g(x_{t-1}, u_t) + \varepsilon_t \\ z_t &= h(x_t) + \delta_t \end{aligned} \quad (13)$$

و با توجه به بردار حالت x_t سیستم، باید ماتریس میانگین حالت μ و ماتریس کواریانس Σ به دست آورده شوند، که داریم:

$$\mu = \begin{bmatrix} x \\ y \\ \varphi \\ m_{1,x} \\ m_{1,y} \\ \vdots \\ m_{n,x} \\ m_{n,y} \end{bmatrix} \quad \left\{ \begin{array}{l} \text{Robot's Pose} \\ \text{Landmarks} \end{array} \right. \quad (14)$$

و همچنین برای ماتریس انتشار کواریانس‌ها نیز [۱۹]:

$$\Sigma = \begin{bmatrix} \Sigma_{x_v x_v} & \Sigma_{x_v m} \\ \Sigma_{m x_v} & \Sigma_{mm} \end{bmatrix} \quad (15)$$

که در آن داریم:

$$\Sigma_{x_v x_v} = \begin{bmatrix} \sigma_{xx} & \sigma_{xy} & \sigma_{x\varphi} \\ \sigma_{yx} & \sigma_{yy} & \sigma_{y\varphi} \\ \sigma_{\varphi x} & \sigma_{\varphi y} & \sigma_{\varphi\varphi} \end{bmatrix} \quad (16)$$

$$\Sigma_{x_v m} = \begin{bmatrix} \sigma_{xm_{1,x}} & \sigma_{xm_{1,y}} & \cdots & \sigma_{xm_{n,x}} & \sigma_{xm_{n,y}} \\ \sigma_{ym_{1,x}} & \sigma_{ym_{1,y}} & \cdots & \sigma_{ym_{n,x}} & \sigma_{ym_{n,y}} \\ \sigma_{\varphi m_{1,x}} & \sigma_{\varphi m_{1,y}} & \cdots & \sigma_{\varphi m_{n,x}} & \sigma_{\varphi m_{n,y}} \end{bmatrix}$$

$$\Sigma_{m x_v} = \begin{bmatrix} \sigma_{m_{1,x}x} & \sigma_{m_{1,x}y} & \sigma_{m_{1,x}\varphi} \\ \sigma_{m_{1,y}x} & \sigma_{m_{1,y}y} & \sigma_{m_{1,y}\varphi} \\ \vdots & \vdots & \vdots \\ \sigma_{m_{n,x}x} & \sigma_{m_{n,x}y} & \sigma_{m_{n,x}\varphi} \\ \sigma_{m_{n,y}x} & \sigma_{m_{n,y}y} & \sigma_{m_{n,y}\varphi} \end{bmatrix}$$

$$\Sigma_{mm} = \begin{bmatrix} \sigma_{m_{1,x}m_{1,x}} & \sigma_{m_{1,x}m_{1,y}} & \cdots & \sigma_{m_{1,x}m_{n,x}} & \sigma_{m_{1,x}m_{n,y}} \\ \sigma_{m_{1,y}m_{1,x}} & \sigma_{m_{1,y}m_{1,y}} & \cdots & \sigma_{m_{1,y}m_{n,x}} & \sigma_{m_{1,y}m_{n,y}} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ \sigma_{m_{n,x}m_{1,x}} & \sigma_{m_{n,x}m_{1,y}} & \cdots & \sigma_{m_{n,x}m_{n,x}} & \sigma_{m_{n,x}m_{n,y}} \\ \sigma_{m_{n,y}m_{1,x}} & \sigma_{m_{n,y}m_{1,y}} & \cdots & \sigma_{m_{n,y}m_{n,x}} & \sigma_{m_{n,y}m_{n,y}} \end{bmatrix}$$

در ادامه به پیاده‌سازی تخمین گر مبتنی بر EKF می‌پردازیم.

توان حداکثر زاویه سکان و سرعت چرخش آن را محدود کرد، به‌صورتی که سرعت V به‌عنوان یک مقدار ثابت تعیین‌شده و محدودیت‌های زیر اعمال می‌شود [۱۹].

$$\begin{cases} |\gamma| < \max G & \max G \in R^+ \\ |\dot{\gamma}| < \text{rate } G & \text{rate } G \in R^+ \\ V = V_{\text{rate}} & V_{\text{rate}} \in R^+ \end{cases} \quad (8)$$

۳-۲- مدل مشاهدات

با فرض اینکه سنسور ربات متحرک یک لیدار است، اندازه‌گیری‌ای که برمی‌گرداند فاصله نسبی $z_r^i(k)$ و زاویه آزیموت $z_\theta^i(k)$ مربوط به علائم نقاط با ویژگی خاص i -ام است. در این صورت، مطابق شکل (۴)، مدل مشاهده را می‌توان به شرح زیر بیان کرد:

$$z_i(k) = \begin{bmatrix} z_r^i(k) \\ z_\theta^i(k) \end{bmatrix} = h_i(x(k)) + r_i(k) = \begin{bmatrix} \sqrt{(x_i - x(k))^2 + (y_i - y(k))^2} \\ \arctan(\frac{y_i - y(k)}{x_i - x(k)})\varphi(k) \end{bmatrix} + \begin{bmatrix} r_i^i(k) \\ r_0^i(k) \end{bmatrix} \quad (9)$$

جایی که $r_i(k)$ بردار نویز مشاهده شده است. به‌منظور نزدیک شدن به سناریوی کاربرد واقعی، محدوده مشاهده سنسور تنها نقاط با ویژگی خاص در یک نیم‌دایره با شعاع MAX_RANGE در جلوی وسیله نقلیه است.

۳-۳- تعریف متغیرهای حالت

با فرض در نظر گرفتن مختصات نقاط با ویژگی‌های خاص به‌صورت $m_{i,x}$ و $m_{i,y}$ است؛ ربات دارای سه متغیر حالت X و Y و زاویه جهت φ ، و ورودی کنترل سرعت V و زاویه سکان γ است؛ همچنان که قبلاً نیز اشاره شد ربات می‌تواند فاصله نسبی $z_r^i(k)$ و زاویه جهت و یا آزیموت $z_\theta^i(k)$ بین خود و نقطه با ویژگی خاص اول را از طریق لیدار به دست آورد. پس می‌توان بردار حالت سیستم را به‌صورت زیر تعریف کرد [۱]:

$$x_t = \underbrace{[x, y, \varphi]}_{\text{Robot's Pose}}, \underbrace{[m_{1,x}, m_{1,y}, \dots, m_{n,x}, m_{n,y}]}_{\text{Landmark } n}^T \quad (10)$$

در همین راستا، بردار کنترل سیستم را به‌صورت زیر می‌توان تعریف کرد

$$u_t = [V, \gamma]^T \quad (11)$$

و بردار مشاهده سیستم را نیز می‌توان به‌صورت زیر تعریف کرد

$$z_t = \underbrace{[z_r^1, z_\theta^1]}_{\text{Landmark 1}}, \dots, \underbrace{[z_r^n, z_\theta^n]}_{\text{Landmark } n}^T \quad (12)$$

۴- پیاده سازی تخمین گر مبتنی بر EKF

۴-۱- مرحله پیش بینی

در مرحله پیش بینی EKF، بر اساس الگوریتم EKF، مقادیر Σ_t و μ_t باید با توجه به رابطه زیر به روز شود. از آنجا که نقاط با ویژگی های خاص که در شبیه سازی استفاده می شوند ایستا هستند، m_t در μ_t و Σ_{mm} در مقدار Σ_t مقدار $\Sigma_{x_v x_v}$ و $\Sigma_{x_v m}$ باید به روز شود.

$$\begin{aligned}\bar{\mu}_t &= g(u_t, u_{t-1}) \\ \bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t\end{aligned}\quad (17)$$

X_v را با توجه به مدل حرکت ربات متحرک و به صورت زیر به روز کرده

$$\begin{aligned}\bar{x}_{vt} &= g_{x_v}(x_{vt}, u_t) \\ &= \begin{bmatrix} x(t-1) \\ y(t-1) \\ \phi(t-1) \end{bmatrix} + \begin{bmatrix} \Delta TV(t) \cos(\phi(t-1) + \gamma(t)) \\ \Delta TV(t) \sin(\phi(t-1) + \gamma(t)) \\ \Delta T \frac{V(t) \sin(\gamma(t))}{L} \end{bmatrix}\end{aligned}\quad (18)$$

از آنجا که نقاط با ویژگی های خاص ثابت هستند، ماتریس ژاکوبین مدل سیستم را می توان به صورت زیر نوشت [۱۹]:

$$G_t = \frac{\partial g(\mu_{t-1}, u_t)}{\partial \mu_{t-1}} = \begin{bmatrix} G_{x_v}^{x_v} & 0 \\ 0 & I \end{bmatrix}\quad (19)$$

که در

$$\begin{aligned}G_{x_v}^{x_v} &= \frac{\partial g_{x_v}(x_{v_{t-1}}, u_t)}{\partial x_{v_{t-1}}} \\ &= I + \frac{\partial}{\partial (x, y, \phi)^T} \begin{bmatrix} V \cdot dt \cdot \cos(\phi + \gamma) \\ V \cdot dt \cdot \sin(\phi + \gamma) \\ \frac{V \cdot dt \cdot \sin(\gamma)}{l} \end{bmatrix} \\ &= \begin{bmatrix} 1 & 0 & -V \cdot dt \cdot \sin(\phi + \gamma) \\ 0 & 1 & V \cdot dt \cdot \cos(\phi + \gamma) \\ 0 & 0 & 1 \end{bmatrix}\end{aligned}\quad (20)$$

طبق الگوریتم EKF، فرمول به روز رسانی Σ_t به صورت زیر است [۱۹]

$$\begin{aligned}\bar{\Sigma}_t &= G_t \Sigma_{t-1} G_t^T + R_t \\ &= \begin{bmatrix} G_{x_v}^{x_v} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \Sigma_{x_v x_v} & \Sigma_{x_v m} \\ \Sigma_{m x_v} & \Sigma_{mm} \end{bmatrix} \begin{bmatrix} (G_{x_v}^{x_v})^T & 0 \\ 0 & I \end{bmatrix} + R_t \\ &= \begin{bmatrix} G_{x_v}^{x_v} \Sigma_{x_v x_v} (G_{x_v}^{x_v})^T & G_{x_v}^{x_v} \Sigma_{x_v m} \\ (G_{x_v}^{x_v} \Sigma_{x_v m})^T & \Sigma_{mm} \end{bmatrix} + R_t\end{aligned}\quad (21)$$

که در آن:

$$R_t = \begin{bmatrix} R_{x_v} & 0 \\ 0 & 0 \end{bmatrix}$$

$$R_{x_v} = \begin{bmatrix} \sigma_x^2 & 0 & 0 \\ 0 & \sigma_y^2 & 0 \\ 0 & 0 & \sigma_\phi^2 \end{bmatrix}\quad (22)$$

اگر واریانس متغیر کنترل σ_v^2 و σ_γ^2 مشخص باشد، ماتریس کوواریانس ورودی کنترل را می توان با فرمول زیر به متغیر حالت ربات متحرک ترسیم کرد.

$$\begin{aligned}G_t^u &= \frac{\partial g_{x_v}(x_{t-1}, u_t)}{\partial u_t} = \begin{bmatrix} \frac{\partial g_{x_v}(x_v, u)}{\partial v} & \frac{\partial g_{x_v}(x_v, u)}{\partial \gamma} \end{bmatrix} \\ &= \begin{bmatrix} dt \cdot \cos(\gamma + \phi) & -V \cdot dt \cdot \sin(\gamma + \phi) \\ dt \cdot \sin(\gamma + \phi) & V \cdot dt \cdot \cos(\gamma + \phi) \\ \frac{dt \cdot \sin(\gamma)}{l} & \frac{dt \cdot \cos(\gamma)}{l} \end{bmatrix}\end{aligned}\quad (23)$$

$$R_{x_v} = G_t^u \begin{bmatrix} \sigma_v^2 & 0 \\ 0 & \sigma_\gamma^2 \end{bmatrix} G_t^{uT}$$

۴-۲- مرحله به روز رسانی

در مرحله به روز رسانی، μ_t و Σ_t با توجه به مقدار به دست آمده از سنسور مطابق رابطه زیر باید اصلاح شوند [۱۹]:

$$\begin{aligned}K_t &= \bar{\Sigma}_t H_t^T (H_t \bar{\Sigma}_t H_t^T + Q_t)^{-1} \\ \mu_t &= \bar{\mu}_t + K_t (z_t - h(\bar{\mu}_t)) \\ \Sigma_t &= (I - K_t H_t) \bar{\Sigma}_t\end{aligned}\quad (24)$$

با توجه به مدل مشاهده ربات متحرک، اجازه دهید متغیرهای زیر را تعریف کنیم:

$$\delta_x = m_{i,x} - x\quad (25)$$

$$\delta_y = m_{i,y} - y$$

که فاصله مختصات نقاط ویژگی از محور X و Y هستند، و سپس

$$q = \delta_x^2 + \delta_y^2\quad (26)$$

پس مدل مشاهده را می توان به صورت نوشت:

$$\bar{z}^i = \begin{pmatrix} \bar{z}_r^i \\ \bar{z}_\theta^i \end{pmatrix} = h(\bar{\mu}) = \begin{pmatrix} \sqrt{q} \\ \arctan 2(\delta_x, \delta_y) - \phi \end{pmatrix}\quad (27)$$

و در ادامه نیز ماتریس ژاکوبین H_t^i مدل مشاهده را می توان به صورت زیر محاسبه کرد:

هر مرحله زمانی مشاهده شد، ژاکوبین اندازه‌گیری را به صورت زیر محاسبه می‌کنیم:

$$H'_{k+1} = \begin{bmatrix} H'_{k+1} & H'_{k+1} \\ \text{Robot} & \text{Landmarks} \end{bmatrix} \quad (32)$$

$$= (\nabla h_{k+1}) C^T (\hat{x}_{v_{k+1}|k})$$

$$\times \begin{bmatrix} -\mathbf{I}_2 & -\mathbf{J}(\hat{x}_{L_{k+1}|k}) & -\hat{x}_{R_{k+1}|k-1} & \mathbf{I}_2 \end{bmatrix}$$

که در آن، C ماتریس دوران است.

۵- OC-EKF پیشنهادی

روش‌های جدیدی برای بهبود سازگاری SLAM مبتنی بر EKF نیز پیشنهاد شده است که شرایط مشاهده‌پذیری در مواقعی که تعداد نقاط با ویژگی‌های خاص کم شود را نیز پوشش دهد. در این مقاله به طور خاص، پیشنهاد شده است تا نقاط خطی سازی EKF به گونه‌ای انتخاب شود که اطمینان حاصل شود که زیر فضای غیرقابل مشاهده مدل سیستم EKF دارای ابعاد مناسب باشد. به این منظور ثابت شده که با محاسبه تخمین اولین ژاکوبین EKF برای هر یک از متغیرهای حالت می‌توان به این مهم دست یافت [۲۰]. رویکرد پیشنهادی در این مقاله که با نام مشاهده‌پذیر مقید OC-EKF پیشنهاد شده است که تحت همان چارچوب کلی قرار می‌گیرد. در این رویکرد خطاهای مورد انتظار نقاط خطی‌سازی (تفاوت بین نقطه خطی سازی و حالت واقعی) را به عنوان یک قید در تابع بهینه‌سازی سیستم لحاظ کرد، که در ادامه به آن می‌پردازیم.

۵-۱- الگوریتم پیشنهادی

عملکرد FEJ-EKF به شدت به مقدار اولیه تخمین حالت متکی است، زیرا از آن در تمام مراحل فیلترینگ برای محاسبه ژاکوبین استفاده می‌کند. اگر تخمین‌ها از حالت واقعی فاصله داشته باشند، خطاهای خطی شده ممکن است زیاد باشد و می‌تواند عملکرد تخمین‌گر را تضعیف کند. در زمینه SLAM، دو هدف اصلی وجود دارد که باید با هم تطبیق داده شوند: کاهش خطاهای خطی سازی در هر مرحله و خواص قابل مشاهده مدل سیستم خطی شده؛ بنابراین نیاز است تا خطای مربعی مورد انتظار نقاط خطی سازی در عین برآوردن شرایط مشاهده‌پذیری به حداقل برسد. این مسئله را می‌توان به عنوان یک مسئله بهینه‌سازی مقید فرموله کرد، که در آن قیود الزامات مشاهده‌پذیری را بیان می‌کند. در این صورت، فیلتر حاصل از قیود مشاهده‌پذیری فیلتر OC-EKF را می‌توان به صورت زیر بیان کرد:

$$\bar{\mu}_t = (x, y, \varphi, m_{1,x}, m_{1,y})$$

$$H_t^i = \frac{\partial h(\bar{\mu}_t)}{\partial \bar{\mu}_t} = \frac{\partial}{\partial \bar{\mu}_t} \left(\arctan 2(\delta_x, \delta_y) - \varphi \right) \frac{\partial \sqrt{q}}{\partial \varphi} \quad (28)$$

$$= \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix}$$

در فرایند مشاهده واقعی، در بسیاری از موارد، (احتمالاً) بیش از یک نقطه با ویژگی خاص مشاهده خواهد شد، یعنی وقتی دنباله نقطه با ویژگی خاص مشاهده شده Z_t است، بردار حالت فعلی μ_t است

$$Z_t = \{z_1, z_2, \dots, z_m\}$$

(29)

$$\mu_t = (x, y, \varphi, \underbrace{m_{1,x}, m_{1,y}}_{\text{robot's pose}}, \dots, \underbrace{m_{n,x}, m_{n,y}}_{\text{landmark n}})^T$$

هنگامی که نقاط با ویژگی خاص متعددی مشاهده می‌شود، از آنجا که نقاط با ویژگی خاص ثابت هستند و هیچ گونه ارتباطی بین هیچ نقطه با ویژگی خاصی وجود ندارد، برای ماتریس ژاکوبین H_t^i مشاهدات نقاط با ویژگی خاص، مراحل مشتق جزئی مربوط به سایر نشانه‌ها همه برابر صفر هستند، و بنابراین، H_t^i را می‌توان به صورت زیر نوشت [۱۹]

$$H_t^i = \frac{1}{q} \begin{pmatrix} -\sqrt{q}\delta_x & -\sqrt{q}\delta_y & 0 & \sqrt{q}\delta_x & \sqrt{q}\delta_y \\ \delta_y & -\delta_x & -q & -\delta_y & \delta_x \end{pmatrix} F_j$$

$$F_j = \begin{pmatrix} 1 & 0 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 1 & 0 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 1 & 0 \dots 0 & 0 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 1 & 0 & 0 \dots 0 \\ 0 & 0 & 0 & 0 \dots 0 & 0 & 1 & 0 \dots 0 \end{pmatrix} \quad (30)$$

۴-۳- ژاکوبین اولین تخمین فیلتر کالمن تعمیم یافته یا FEJ-EKF

به منظور محاسبه این فیلتر نیاز است تا دو تغییر زیر در فیلترینگ انجام شود [۲۰]:

(۱) به جای محاسبه ژاکوبین به فرم $G_t^{x_v}$ برای انتشار حالت‌ها، کافی است از روش زیر برای محاسبه ژاکوبین استفاده نمود:

$$G_t^{x_v} = \begin{bmatrix} \mathbf{I}_2 & \mathbf{J}(x_{v_{k+1}|k} - x_{v_{k|k-1}}) \\ \mathbf{0}_{1 \times 2} & 1 \end{bmatrix} \quad (31)$$

$$\text{where } \mathbf{J} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix}$$

(۲) در ارزیابی ماتریس ژاکوبین اندازه‌گیری ما همیشه از تخمین نقاط شاخص استفاده می‌کنیم؛ بنابراین، اگر نقطه شاخصی برای اولین بار در

بسته از طریق بررسی مکان هندسی مقادیر ویژه ماتریس.

جدول (۱): پارامترهای مربوط به شبیه‌سازی‌ها			
پارامتر	نماد	مقدار	واحد
سرعت	V	۸	متر بر ثانیه
واریانس سرعت	σ_V	۲	متر بر ثانیه
واریانس زاویه سمت	σ_γ	۰/۲	رادیان
شعاع دید	R	۱۰	متر

۶- تحلیل نتایج

به منظور نمایش مزایای و اثربخشی روش فیلترینگ پیشنهادی در مقایسه به فیلتر کالمن تعمیم‌یافته و فیلتر کالمن تعمیم‌یافته مبتنی بر ژاکوبین نخستین تخمین، در این بخش به بررسی شبیه‌سازی‌های سامانه‌ی SLAM مدل‌شده در حضور فیلترهای معرفی شده می‌پردازیم. جدول (۱) مقادیر مربوط به پارامترها و محدوده‌ی نامعینی آن‌ها را نمایش می‌دهد. شکل (۶) نتایج شبیه‌سازی برای ناوبری تصویری یک ربات با استفاده از الگوریتم معرفی شده و در مقایسه با حالت بدون استفاده از واحد فیلترینگ و در مقایسه با دیگر روش‌های فیلترینگ را نمایش می‌دهد.

واحد محورهای طولی و عرضی متر بوده و ستاره‌های آبی‌رنگ نشان‌دهنده نقاط با ویژگی‌های خاص می‌باشند؛ نقاط آبی نشان‌دهنده نقاط کلیدی در مسیر هستند؛ خط سیر سیاه رنگ مسیر درستی که ربات لازم است طی کند را نمایش می‌دهد. خط‌چین آبی رنگ مسیر طی شده با استفاده از فیلترپیشنهادی، خط‌چین زرد رنگ مسیر طی شده با استفاده از فیلترکالمن تعمیم‌یافته استاندارد و خط‌چین قرمز مسیر طی شده‌ی بدون استفاده از واحد فیلترینگ را نمایش می‌دهد؛ دایره نازک آبی‌رنگ نشان‌دهنده شعاع مشاهده سنسور تصویری ربات بوده که فقط نقاط با ویژگی خاص روبرو خود با زاویه ۱۸۰ درجه را مشاهده می‌کند. در این حالت ۱۰۰ عدد نقطه با ویژگی خاص موجود می‌باشد. همچنان که مشاهده می‌شود، روش OC-EKF خطای بسیار کمتری را نسبت به STD-EKF برای عبور از مسیر واقعی دارد. شکل (۷) نشان‌دهنده میزان خطا در راستاهای طولی، عرضی بر حسب متر و خطای جهت‌گیری تصویری بر حسب درجه برای ناوبری همان ربات در مسیر طی شده در شکل (۷) با استفاده از فیلترهای OC-EKF و STD-EKF، فیلتر کالمن مبتنی بر H_{inf} ، فیلتر کالمن مقاوم و بدون حضور سامانه‌ی فیلترینگ را نمایش می‌دهد که تأثیر بسزای استفاده از فیلتر ارائه شده OC-EKF قابل‌مشاهده است. طبق نتایج نشان داده شده کاملاً مشهود است که میزان خطا به وسیله روش فیلترینگ پیشنهادی OC-EKF نه تنها در جهت‌گیری، بلکه در راستاهای طولی و عرضی نیز بسیار کمتر از روش‌های دیگر فیلترینگ می‌باشد.

$$\min \left(\int \|x_r - \hat{x}_r\|^2 \mu(x_r | z_{0:k}) dx_r + \int \|x_{k+1} - \hat{x}_{k+1}\|^2 \mu(x_{k+1} | z_{0:k}) dx_{k+1} \right) \quad (33)$$

$$\text{s. t.} \quad H_{k+1} \sum_k \dots \sum_{k_0} U = 0 \quad \forall k \geq k_0$$

که مساله بهینه‌سازی فوق بوسیله جعبه‌ابزار *ylimp* به راحتی قابل حل است. البته به منظور افزودن اطلاعات نقطه با ویژگی خاص مشاهده‌شده به بردار حالت و ماتریس کوواریانس برای تغییر ابعاد μ و Σ نیاز به افزایش بعد بردار حالت می‌باشد. اگر مقدار مشاهده یک نقطه با ویژگی خاص مشاهده شده Z باشد، پس طبق مدل مشاهده می‌توان آن را با توجه به موقعیت کنونی ربات به صورت زیر نوشت:

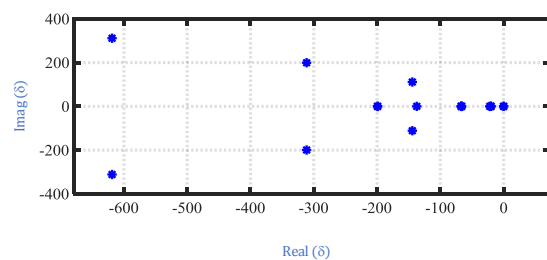
$$m = f(z, x_v) = \begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} x + z_r \sin(\varphi + z_\theta) \\ y + z_r \cos(\varphi + z_\theta) \end{pmatrix} \quad (34)$$

که با خطی کردن آن، می‌توان ماتریس ژاکوبین را به صورت زیر پیدا کنید

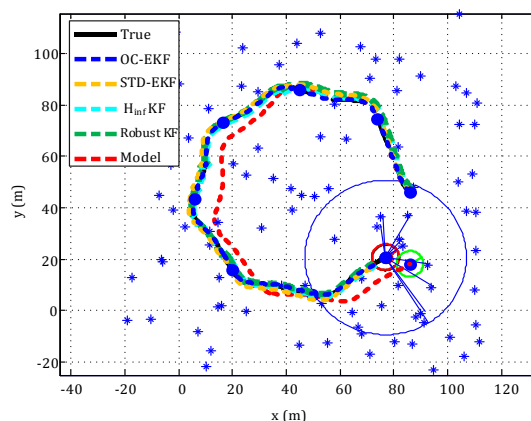
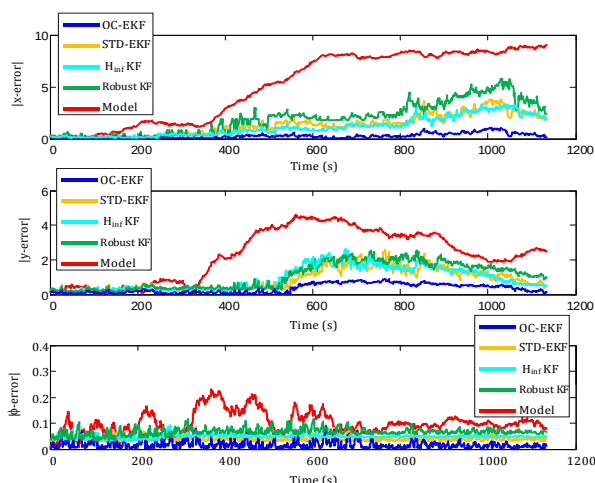
$$F_{x_v} = \frac{\partial f(z, x_v)}{\partial x_v} = \begin{pmatrix} 1 & 0 & -z_r \cos(\varphi + z_\theta) \\ 0 & 1 & z_r \sin(\varphi + z_\theta) \end{pmatrix} \quad (35)$$

$$F_z = \frac{\partial f(z, x_v)}{\partial z} = \begin{pmatrix} \cos(\varphi + z_\theta) & -z_r \cos(\varphi + z_\theta) \\ \sin(\varphi + z_\theta) & z_r \sin(\varphi + z_\theta) \end{pmatrix}$$

باید در نظر داشت هنگام به‌روزرسانی ماتریس کوواریانس، لازم است قسمت Σ_{mxv} ، Σ_{xvm} و Σ_{mm} مربوط به نقطه با ویژگی خاص در Σ به‌روز شود. ماتریس کوواریانس این قسمت را می‌توان از طریق ژاکوبین‌های F_{xv} و F_z به دست آورد تا ماتریس کوواریانس مدل مشاهده ماتریس کوواریانس موقعیت ربات به ضرب همبستگی مربوطه تبدیل می‌شود. اگر نقطه Z با ویژگی خاص تازه مشاهده شده باشد، روند استفاده از آن برای افزودن بردار حالت جدید به شرح الگوریتم (۱) است. بر خلاف روش FEJ-EKF که نیاز به اثبات پایداری سیستم بوسیله تابع لیپانوف با استفاده از ماتریس ژاکوبین مشخص شده در رابطه (۲۸) را دارد، پایداری روش پیشنهادی OC-EKF به راحتی و با بررسی مقادیر ویژه ماتریس Σ_k قابل بررسی است. شکل زیر، پایداری سیستم حلقه بسته با بررسی مکان هندسی قطب‌های سیستم در حضور الگوریتم پیشنهادی را نمایش می‌دهد.



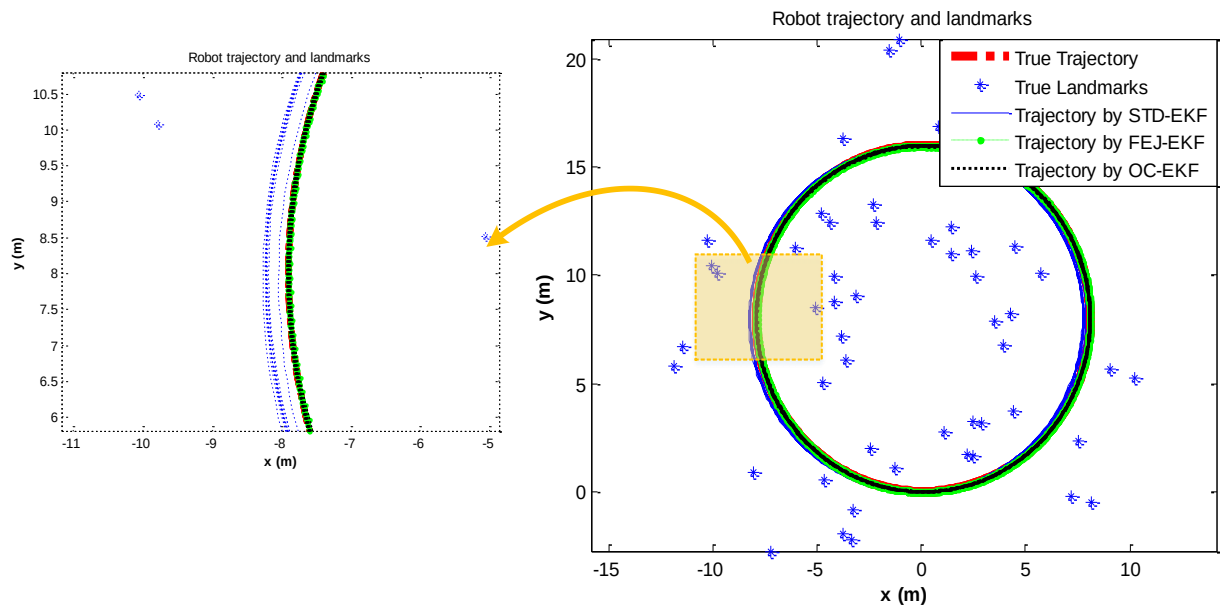
شکل (۵): نمایش پایداری سیستم با بررسی قطب‌های سیستم حلقه



شکل (۶): نتایج شبیه‌سازی برای ناوبری تصویری ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با STD-EKF، FEJ-EKF، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم

شکل (۷): میزان خطا در راستاهای طولی، عرضی و خطای جهت گیری با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با STD-EKF، FEJ-EKF، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم

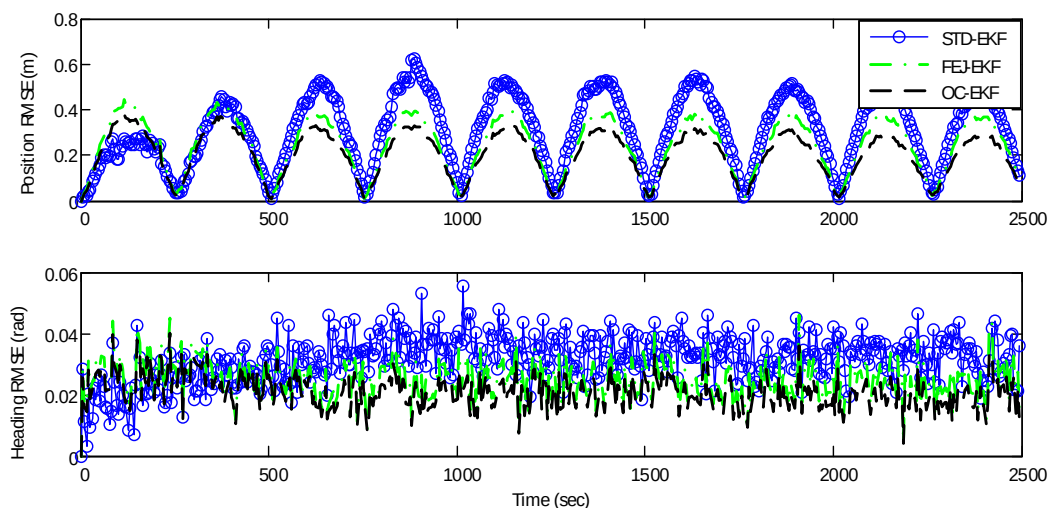
الگوریتم (۱): الگوریتم پیشنهادی	
Augmented $(\mu_{k-1}^{Optimal}, \Sigma_{k-1}^{Optimal}, z)$	مدل افزوده‌ی مشاهده‌پذیر مقید
$m = f(z, x_v) = \begin{pmatrix} m_x \\ m_y \end{pmatrix} = \begin{pmatrix} x + z_r \sin(\varphi + z_\theta) \\ y + z_r \cos(\varphi + z_\theta) \end{pmatrix}$	۱. مشاهدات در موقعیت کنونی
augment μ $\mu_k = \begin{pmatrix} \mu_{k-1} \\ m \end{pmatrix}$ $\Sigma_{k-1} = \begin{bmatrix} \Sigma_{x_v x_v, k-1} & \Sigma_{x_v m, k-1} \\ \Sigma_{m x_v, k-1} & \Sigma_{mm, k-1} \end{bmatrix}$ $F_{x_v} = \begin{pmatrix} 1 & 0 & -z_r \cos(\varphi + z_\theta) \\ 0 & 1 & z_r \sin(\varphi + z_\theta) \end{pmatrix}$ $F_z = \begin{pmatrix} \cos(\varphi + z_\theta) & -z_r \cos(\varphi + z_\theta) \\ \sin(\varphi + z_\theta) & z_r \sin(\varphi + z_\theta) \end{pmatrix}$ $P_{mm}^4 = F_{x_v} \Sigma_{x_v x_v, k-1} F_{x_v}^T + F_z Q F_z^T$ $P_{mm}^2 = F_{x_v} \Sigma_{x_v m, k-1}$ $P_{mm}^3 = P_{mm}^2 T$ $P_{x_v m} = F_{x_v} \Sigma_{x_v x_v, k-1}$	۲. مدل افزوده میانگین حالت‌ها با توجه به مشاهدات جدید
augment Σ $\Sigma_k = \begin{bmatrix} \Sigma_{x_v x_v, k-1} & \Sigma_{x_v m, k-1} & P_{x_v m} \\ \Sigma_{m x_v, k-1} & \Sigma_{mm, k-1} & P_{mm}^2 \\ P_{x_v m} & P_{mm}^3 & P_{mm}^4 \end{bmatrix}$	۳. مدل افزوده کواریانس حالت‌ها با توجه به مشاهدات جدید
return $\mu_k^{Optimal}, \Sigma_k^{Optimal}$ from equation (37)	۴. بازگرداندن مقادیر مقید بهینه‌شده



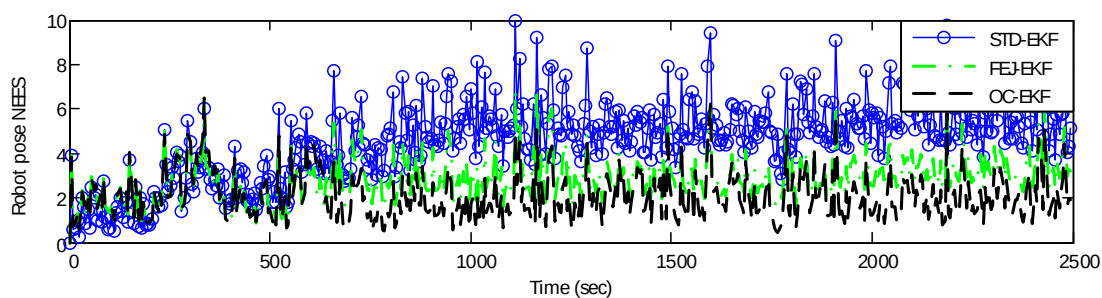
شکل (۸): نتایج شبیه‌سازی مونت کارلو برای ناوبری تصویری ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با STD-EKF و FEJ-EKF

میانگین مربعات را برای فاصله از مقدار واقعی موقعیت و همچنین زاویه سمت را نمایش می‌دهد. شکل (۱۰) نیز میزان خطای تخمین نرمالایز شده‌ی حالت‌ها برای ناوبری تصویری ربات با استفاده از اشکال مختلف فیلتر کالمن تعمیم‌یافته شده را نمایش می‌دهد. همچنان که مشخص است، فیلتر مشاهده‌پذیر مقید ارائه شده تخمین مناسب‌تری را برای سامانه‌ی ناوبری ارائه می‌دهند.

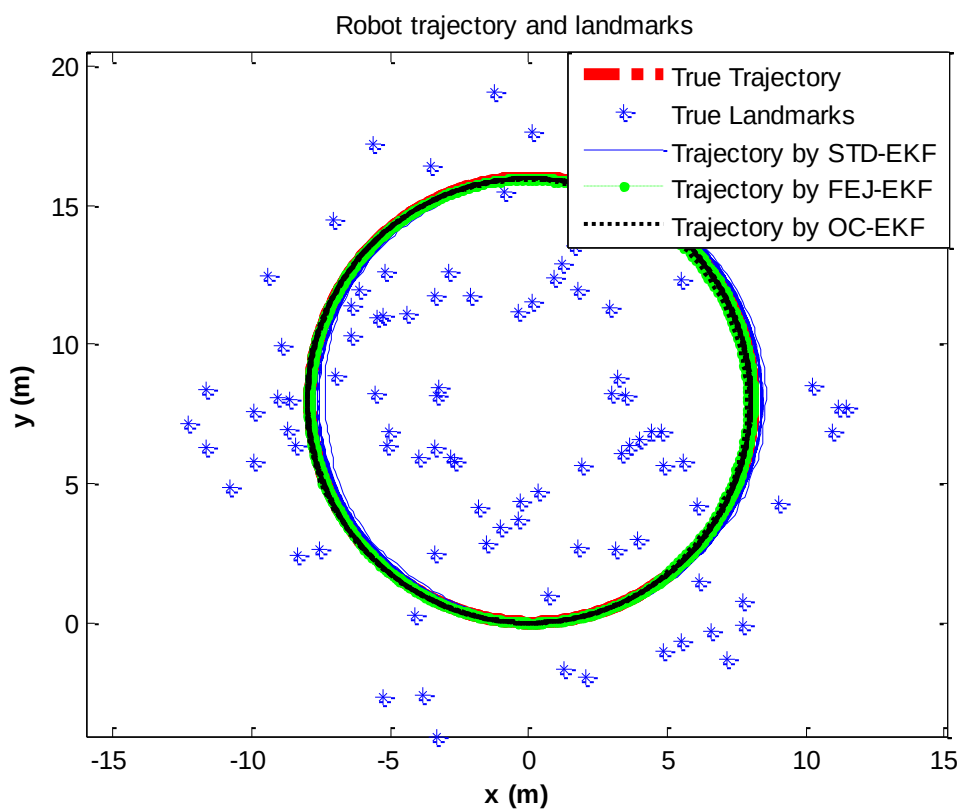
شکل (۸) نتایج شبیه‌سازی مونت کارلو ناوبری تصویری برای روش‌های مبتنی بر فیلتر کالمن تعمیم‌یافته استاندارد STD-EKF، تخمین اولیه ژاکوبین FEJ-EKF و روش پیشنهادی مشاهده‌پذیر مقید OC-EKF را نمایش می‌دهد که همگی توانسته‌اند حتی با وجود تعداد نقاط با ویژگی مشخص کم (۶۰ عدد)، با دقت بالایی ناوبری مربوطه را انجام دهند. به منظور بررسی عملکرد این فیلترها، شکل (۹) خطای ریشه



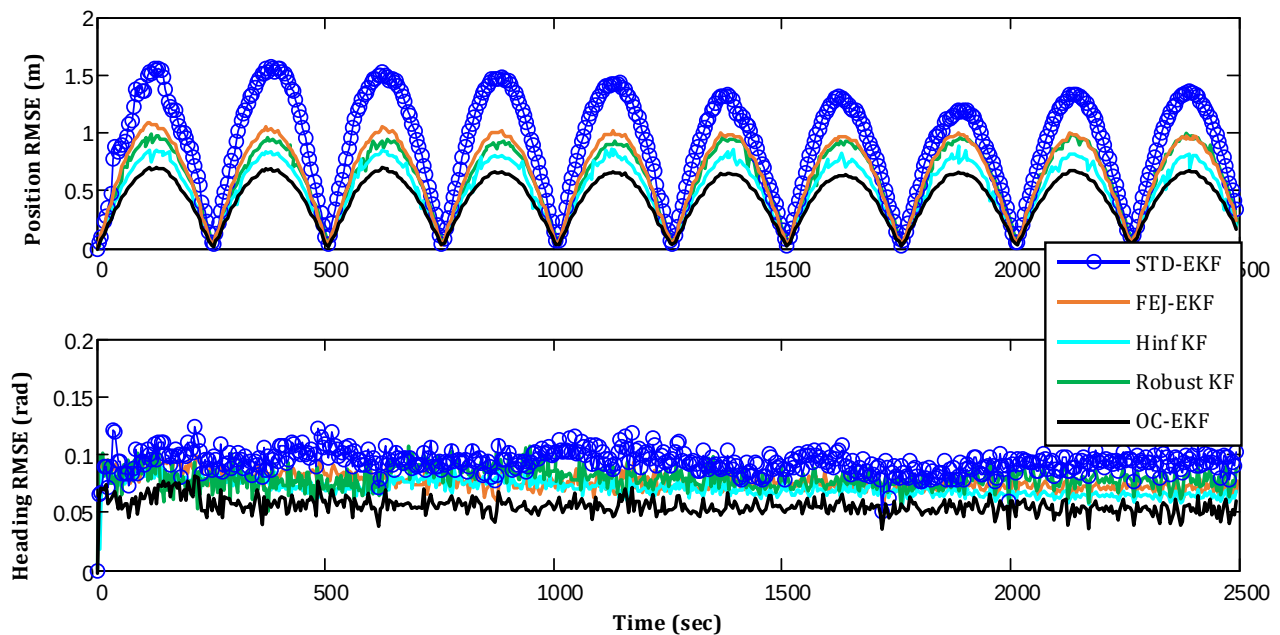
شکل (۹): میزان خطای موقعیت و خطای جهت‌گیری برای ناوبری تصویری یک ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با FEJ-EKF و STD-EKF



شکل (۱۰): میزان خطای تخمین نرمالایز شده‌ی حالت‌ها برای ناوبری تصویری یک ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با FEJ-EKF و STD-EKF



شکل (۱۱): نتایج شبیه‌سازی مونت کارلو برای ناوبری تصویری ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با FEJ-EKF و STD-EKF در حضور نویز رنگی



شکل (۱۲): میزان خطای موقعیت و خطای جهت‌گیری برای ناوبری تصویری یک ربات با استفاده از فیلتر OC-EKF ارائه شده و مقایسه آن با فیلتر STD-EKF، FEJ-EKF، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم در حضور نویز رنگی

چگونگی پیاده‌سازی سیستماتیک فیلترینگ آن‌ها، به مقایسه عملکرد فیلتر کالمن تعمیم‌یافته استاندارد STD-EKF، فیلتر کالمن تعمیم‌یافته مبتنی بر ژاکوبین نخستین تخمین FEJ-EKF، فیلتر کالمن مبتنی بر H_{inf} ، فیلتر کالمن مقاوم و روش پیشنهادی فیلتر کالمن تعمیم‌یافته مشاهده‌پذیر مقید OC-EKF پرداخته شده است. همچنان که در نتایج مونت‌کارلو شبیه‌سازی‌ها نشان داده شده است، فیلتر OC-EKF با حفظ عملکرد بهتر، نسبت به نامعینی‌های سیستم مقاوم‌تر بوده و در مقایسه با دیگر فیلترهای غیرخطی بررسی شده در این پژوهش، و حتی با وجود نامعینی‌های مختلف، دقت بهتری را برای ناوبری دقیق مجموعه فراهم کرده است.

مراجع

- [1] S. A. S. Mohamed, M.-H. Haghbayan, T. Westerlund, J. Heikkonen, H. Tenhunen, and J. Plosila, "A survey on odometry for autonomous navigation systems", IEEE Access, vol. 7, pp. 97466–97486, 2019.
- [2] N. Dinarvand, M. Norouzi, M. Dosarianian-Moghadam, "A real time optimization-based SLAM for indoor UAV flying robots", Journal of Iranian Association of Electrical and Electronics Engineers, vol. 18, no. 4, pp. 197–205, 2021.
- [3] Farazkish R. Reliability modeling in bio-nano robots. "Journal of Iranian Association of Electrical and Electronics Engineers", 17 (3), pp. 11-16, 2020.
- [4] F. Amiri, Kahaei M H. Enhancing Supervised Hyperspectral Unmixing using Spatial Correlation under Nonlinear Mixing Model. Journal of Iranian Association of Electrical and Electronics Engineers. 2020; 17 (1): 35-44.

شکل (۱۱) نتایج شبیه‌سازی مونت‌کارلو ناوبری تصویری به منظور مقایسه جامع فیلتر OC-EKF با روش‌های STD-EKF، FEJ-EKF و شکل (۱۲) مقایسه آن‌ها با فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم در حضور نویز رنگی به هنگام کاهش قابل‌ملاحظه‌ی تعداد نقاط با ویژگی مشخص (۵۰ عدد) را نمایش می‌دهد. خطای ریشه میانگین مربعات را برای فاصله از مقدار واقعی موقعیت و همچنین زاویه سمت برای این حالت نیز در شکل (۱۲) قابل‌مشاهده می‌باشد. همچنان که مشخص است، فیلتر کالمن مشاهده‌پذیر مقید تخمین بسیار مناسب‌تری برای سامانه‌ی ناوبری در حضور انواع نامعینی‌ها نسبت به STD-EKF، FEJ-EKF، فیلتر کالمن مبتنی بر H_{inf} و فیلتر کالمن مقاوم ارائه می‌دهد.

۷- نتیجه‌گیری

این مقاله به طراحی فیلتر کالمن توسعه‌یافته مشاهده‌پذیر مقید OC-EKF به منظور افزایش دقت سامانه‌های ناوبری مبتنی بر SLAM پرداخته است. روش معمول فیلترینگ این سامانه‌ها مبتنی بر فیلترهای خطی و یا فیلتر کالمن تعمیم‌یافته استاندارد می‌باشد. اگرچه فیلتر EKF به عنوان یکی از مناسب‌ترین راه‌حل‌ها به منظور بهبود تلفیق مناسب داده‌ها برای افزایش دقت مساله ناوبری در سامانه‌های SLAM معرفی شده است، با این حال، تعداد نقاط با ویژگی خاص، میزان نویز و وجود یا عدم وجود نویز رنگی تاثیر مستقیمی بر روی کارایی فیلترهای مرسوم مبتنی بر EKF داشته است. در این مقاله، علاوه بر نمایش نحوه مدل‌سازی دقیق سامانه‌های ناوبری SLAM و

consistency”, in *Experimental Robotics*, 2009, pp. 373–382.

زیر نویس ها

^۱Scale Invariant Feature Transform

^۲Simultaneous Localization and Mapping (SLAM)

^۳Standard Extended Kalman Filter (STD-EKF)

^۴First Estimate Jacobean Extended Kalman Filter (FEJ-EKF)

^۵Observability Constrained Extended Kalman Filter (OC-EKF)

- [5] J. Zeraatkarmoghadam, H. Farrokhi, N. Neda, Interference Management in Cognitive Radio Networks using Cooperative Beamforming with Imperfect CSI. *Journal of Iranian Association of Electrical and Electronics Engineers*. 2017; 14 (2):1-9.
- [6] F. Mirzaei, A. A. Pouyan, “A Review of Swarm Robotic and Its Place in Multi -Robotic Systems”, *Journal of Iranian Association of Electrical and Electronics Engineers*. 17 (2), pp. 53-72, 2020.
- [7] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, “ORB-SLAM: a versatile and accurate monocular SLAM system”, *IEEE Trans. Robot.*, vol. 31, no. 5, pp. 1147–1163, 2015.
- [8] D. Xu, Y. F. Li, and M. Tan, A general recursive linear method and unique solution pattern design for the perspective-n-point problem”, *Image Vis. Comput.*, vol. 26, no. 6, pp. 740–750, 2008.
- [9] M. Nezhadshahbodaghi and M. R. Mosavi, “A loosely-coupled EMD-denoised stereo VO/INS/GPS integration system in GNSS-denied environments”, *Measurement*, 183, 109895, 2021.
- [10] J. Gui, D. Gu, S. Wang, and H. Hu, “A review of visual inertial odometry from filtering and optimisation perspectives”, *Adv. Robot.*, vol. 29, no. 20, pp. 1289–1301, 2015.
- [11] M. A. Fischler and R. C. Bolles, “A paradigm for model fitting with applications to image analysis and automated cartography (reprinted in readings in computer vision, ed. ma fischler,””, *Comm. ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [12] Parsa P, Safabakhsh R. A New Method for Image Segmentation based on Multi-Objective Differential Evolution Fuzzy Clustering . *Journal of Iranian Association of Electrical and Electronics Engineers*. 2016; 13 (2) :103-114.
- [13] kalantari S, Abdollahifard M J, ahmadi S. Image Super-Resolution Using Analytical Edge Model. *Journal of Iranian Association of Electrical and Electronics Engineers*. 2018; 15 (2) :45-54.
- [14] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, “MonoSLAM: Real-time single camera SLAM”, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 6, pp. 1052–1067, 2007.
- [15] Z.-L. Ren, L.-G. Wang, and L. Bi, “Improved extended kalman filter based on fuzzy adaptation for slam in underground tunnels”, *Int. J. Precis. Eng. Manuf.*, vol. 20, no. 12, pp. 2119–2127, 2019.
- [16] M.-Y. Ju, Y.-J. Chen, and W.-C. Jiang, “Implementation of odometry with ekf in hector slam methods”, *Int. J. Autom. Smart Technol.*, vol. 8, no. 1, pp. 9–18, 2018.
- [17] Y. Song, Z. Zhang, J. Wu, Y. Wang, L. Zhao, and S. Huang, “A Right Invariant Extended Kalman Filter for Object based SLAM”, *arXiv Prepr. arXiv2109.05297*, 2021.
- [18] T. Nemoto, K. Onodera, R. E. Mohan, M. Iwase, and K. Wood, “An application of the simultaneous localization and mapping (slam) method based on the unscented kalman filter (ukf) to a reconfigurable quadruped robot with crawling locomotion”, in *2018 International Conference on Reconfigurable Mechanisms and Robots (ReMAR)*, 2018, pp. 1–8.
- [19] I. Ullah, X. Su, X. Zhang, and D. Choi, “Simultaneous localization and mapping based on Kalman filter and extended Kalman filter”, *Wirel. Commun. Mob. Comput.*, vol. 2020, 2020.
- [20] G. P. Huang, A. I. Mourikis, and S. I. Roumeliotis, “A first-estimates Jacobian EKF for improving SLAM