

## بهبود عملکرد الگوریتم EKF-SLAM در محیط‌های پویا با استفاده از ANFIS

اسماعیل خان میرزا\*

دانشیار، گروه مهندسی مکانیک، دانشگاه علم و صنعت، تهران، ایران، khanmirza@iust.ac.ir

فریدون جعفری بابوکانی

دانشجوی کارشناسی ارشد، مهندسی مکانیک، دانشگاه علم و صنعت، تهران، ایران، ferydoon.jafari@gmail.com

### چکیده

مسئله‌ی مکان‌یابی و نقشه‌کشی همزمان در محیط‌های پویا موضوع مهمی در مبحث ناوبری ربات‌های خودمختار می‌باشد که هنوز تحقیقات زیادی در این زمینه انجام نشده است. در این مقاله، با ارائه‌ی روشی جدید، اشیاء پویا در محیط را همزمان با وجود اشیاء ساکن ردیابی می‌نماییم. برای این منظور الگوریتم EKFSLAM برای محیط‌های پویا توسعه داده شده است به طوری که مکان‌یابی و نقشه‌کشی همزمان و ردیابی اشیاء پویا در محیط در قالب یک مسئله حل می‌شود، که قبلاً به صورت مجزا این کار انجام می‌شد. همچنین با توجه به اینکه کارایی و عملکرد الگوریتم فیلتر کالمن توسعه‌یافته وابستگی زیادی به دانش صحیح در مورد ماتریس کواریانس نوین مشاهدات دارد از یک سیستم تطبیقی نرو-فازی (ANFIS) به منظور تنظیم ماتریس کواریانس نوین مشاهدات استفاده شده تا دقت و پایداری الگوریتم را نسبت به سایر روش‌های قدیمی (SLAM and DATMO, FastSLAM, EKF) تضمین نماید. نتایج آزمایشات حاکی از آن است که عملکرد الگوریتم پیشنهادی موجب ردیابی دقیق اشیاء در زمان اجرای SLAM در محیط پویا می‌شود و از دقت و پایداری خوبی برخوردار است. واژه‌های کلیدی: SLAM، EKF، محیط پویا، ANFIS، ردیابی چند هدفه.

## Improving the performance of the EKF-SLAM algorithm in dynamic environments using ANFIS

E. Khanmirza

Department of Mechanical Engineering, Iran University of Science & Technology, Tehran, Iran

F. Jafari Babookani

Department of Mechanical Engineering, Iran University of Science & Technology, Tehran, Iran

### Abstract

The issue of Simultaneous localization and mapping (SLAM) in dynamic environments is an important object in the navigation of autonomous robots, which has not yet been investigated much. In this paper, by presenting a new method, we track dynamic objects in the environment simultaneously with living static object. For this purpose, the EKFSLAM algorithm has been developed for dynamic environments in the way that simultaneous mapping and mapping of dynamic objects in the environment are solved in the form of a problem, which was previously done individually. Also, considering that the efficiency and performance of the developed Kalman filter algorithm depend heavily on the correct knowledge of the covariance matrix of observations, an adaptive Neuro-Fuzzy inference system (ANFIS) is used to adjust the measurement noise covariance matrices to ensure that the accuracy and consistency of the algorithm Than other older methods (SLAM and DATMO, FastSLAM, EKF) is guaranteed. The results of the experiments indicate that the proposed algorithm performs precise tracking of objects during the implementation of SLAM in a dynamic environment and it has good accuracy and consistency.

**Keywords:** SLAM, EKF, Dynamic environment, ANFIS, objective Tracking.

کوتاه [۷-۹] و همچنین ثابت فرض کردن ماتریس کواریانس نوین مشاهدات و فرایند انجام شده است [۱۰-۱۲]، اما حل مسئله برای محیط‌های پویا با زمان عملکرد زیاد ربات و تطبیقی نمودن ماتریس کواریانس نوین مشاهدات هنوز از مسائل قابل بحث در این موضوع می‌باشد. در روش‌های متداولی که تا امروز برای اصلاح و بهبود نتایج و عملکرد SLAM مورد استفاده قرار گرفته است عمدتاً شامل مدل‌های ریاضی از حرکت ربات و حسگرهاست که سعی می‌کنند تا رفتار واقعی ربات را با دقت خوبی پیش‌بینی کنند. اما خروجی حاصل از این مدل‌ها به وسیله فیلترهای مختلف و بر مبنای اندازه‌گیری حسگرها اصلاح می‌شود. راه حل‌های فراوانی برای حل مسئله SLAM در محیط‌های ساکن انجام شده است اما حل مسئله در محیط‌های پویا به دلیل پیچیدگی زیادی که دارد از مسائل قابل بحث در این موضوع می‌باشد. دلیل آن این است که اشیاء پویای موجود در محیط نیازمند تشخیص دسته‌بندی و ردیابی هستند. بنابراین در سال‌های اخیر پژوهشگران رو

### ۱- مقدمه

یکی از توانایی‌های اصلی مورد نیاز برای ربات‌های خودمختار تعامل با محیط می‌باشد. عموماً ربات‌ها برای داشتن تعامل با محیط از داده‌های حسگرهایی که دارای اغتشاشات هستند استفاده نموده تا به درکی از محیط دست یابند و در عین حال موقعیت خود در محیط را تعیین نمایند. این فرآیند تحت عنوان مکان‌یابی و نقشه‌کشی همزمان<sup>۱</sup> شناخته می‌شود. بهبود عملکرد ربات با استفاده از این روش نسبت به سایر روش‌های ناوبری [۱] بر هیچ‌کس پوشیده نیست. پژوهش‌های زیادی بر روی مسئله‌ی SLAM با در نظر گرفتن این فرض که محیط اطراف ربات به صورت ساکن [۲-۶]، زمان عملکرد ربات

<sup>1</sup> Simultaneous localization and mapping (SLAM)

\* نویسنده مکاتبه کننده، آدرس پست الکترونیکی: khanmirza@iust.ac.ir

تاریخ دریافت: ۹۷/۰۱/۲۲

تاریخ پذیرش: ۹۸/۰۴/۲۲

به سمت حل مسأله SLAM در محیط‌های پویا آورده‌اند. فرض اشیاء ساکن، در محیط‌های واقعی که اشیاء پویا وجود دارند باعث کاهش چشم‌گیر دقت و پایداری می‌شود بنابراین این لازم است تا محیط به صورت پویا در نظر گرفته شود.

با سخن گفتن از موانع پویا موضوع جالب تشخیص و دنبال کردن موانع پویا<sup>۱</sup> نیز به مسأله اضافه می‌شود، اولین بار در مسأله‌ی SLAM توسط وانگ و همکاران [۱۳] این روش ارائه داده‌شد. این محققان همزمان SLAM و DTMO را برای محیط داخلی و محیط خارجی حل نمودند. هدفشان حل این دو مسأله برای سرعت‌های بالا و محیط‌های بیرونی بود. که مسأله را به صورت دو بخش جداگانه‌ی ساکن و پویا در نظر گرفته و آن را حل کردند. روش‌های متعددی برای ردیابی اشیاء پویا با استفاده از یک ربات وجود دارد اما یکی از موفق‌ترین روش‌ها توسط بوبروک و همکاران [۱۴]، ارائه شده است. این محققان با استفاده از الگوریتم تقسیم‌بندی بلوک، پارامترهایی نظیر سرعت و موقعیت هر شیء را شناسایی می‌کند. در این روش یک مدل کلی حرکت به کار رفته است که امکان ردیابی انواع مختلفی از اشیاء (از انسان گرفته تا سایر پلتفرم‌های متحرک) را در یک سیستم فراهم می‌کند. ولف و همکاران [۱۵]، الگوریتم جدیدی ارائه دادند که به وسیله‌ی آن بین اشیاء پویا و ساکن موجود در محیط تمایز قائل می‌شود. به این صورت که اشیاء پویا را در یک نقشه و اشیاء ساکن را در نقشه‌ی دیگر مدل می‌کردند که با ترکیب این دو نقشه یک مدل کامل از توصیف محیط ایجاد می‌کردند. همچنین با ذخیره‌ی نقاط راهنما در نقشه‌ی سوم این نقاط راهنمای ساکن را برای حل مسأله‌ی SLAM بکار می‌گرفتند. نتایج این محققان نشان می‌دهد که تشخیص موانع پویا برای حل مسأله بسیار مهم است. خانمیرزا و همکاران [۱۶] با ارائه یک روش تطبیقی جدید سعی در تعمیم این مسأله برای ربات‌های همکاری در محیط‌های پویا داشتند، این محققان ابتدا مسأله SLAM را با استفاده از مشخصه‌های محیطی ساکن حل کرده و سپس ردیابی هدف به صورت مجزا با کمک فیلترهای موجود در حوزه ردیابی هدف تشخیص و ردگیری کرده‌اند. آه و همکاران [۱۷]، با مستقل اعلام کردن نقاط راهنما از مسأله سعی در کاهش خطا داشتند. به این صورت که با تجزیه‌کردن مسأله به حالت سنتی و جدید به استفاده از نقاط راهنمای پویا برای کاهش خطا پرداختند. بحرینی و همکاران [۱۸]، یک رویکرد جدید برای دسته بندی و ردیابی چندین شیء متحرک در محیط ارائه داده‌اند. مشخصه اصلی الگوریتم ارائه شده توسط آنان در پرداختن به اشیاء ثابت و متحرک در SLAM و تشخیص و ردیابی آنها بدون تقسیم مسأله به دو بخش (SLAM و DATMO) است.

به منظور بهبود الگوریتم فیلتر کالمن توسعه‌یافته در محیط‌های ساکن از روش‌هایی بر پایه‌ی الگوریتم‌های ابتکاری<sup>۲</sup> که به بهبود آن منجر می‌شود در برخی موارد استفاده شده است. اما برای محیط‌های پویا این کار صورت نگرفته است. از جمله مزایای شبکه‌ی عصبی تطبیقی توانایی یادگیری، بهینه‌سازی و انعطاف‌پذیری می‌باشد. در حالی که یک منطق فازی روشی مبتنی بر قواعد است که این قواعد توسط علم افراد خبره ساخته می‌شوند. سیستم استنتاج عصبی فازی

تطبیقی<sup>۳</sup> مزیت استفاده از شبکه عصبی تطبیقی و منطق فازی را با هم ترکیب می‌کند. در زمینه‌ی کاهش خطا با توجه به دانسته و ثابت فرض نمودن ماتریس کواریانس نویز مشاهدات در محیط ساکن نیز پژوهش‌هایی انجام شده است. به عنوان نمونه می‌توان به پژوهش آمیتا چاترج و ماتسونو [۱۹]، اشاره نمود این محققان از روش نروفازی برای تخمین عناصر ماتریس R در الگوریتم فیلتر کالمن توسعه‌یافته<sup>۴</sup> زمانی که مشاهدات بروز رسانی می‌شوند استفاده کردند، تا بدین وسیله الگوریتم فیلتر کالمن توسعه‌یافته که برای حل مسأله مکان‌یابی و نقشه‌کشی همزمان بکار می‌رود را بهبود بخشند. در پژوهشی دیگر آمیتا چاترج و ماتسونو [۲۰]، مسأله مکان‌یابی و نقشه‌کشی همزمان برای یک ربات متحرک زمینی را به کمک الگوریتم فیلتر کالمن توسعه‌یافته حل نمودند. در این مقاله آنان با استفاده از آموزش سیستم فازی با ناظر توسط یک الگوریتم جدید ازدحام ذرات، روش فیلتر کالمن توسعه‌یافته را بهبود بخشیدند. در سال ۲۰۱۰ هاونگی و همکاران [۲۱]، برای حل مسأله مکان‌یابی ربات‌ها از روش (EKF-ANFIS) استفاده نمودند. در این پژوهش از (ANFIS) جهت افزایش کارایی الگوریتم فیلتر کالمن توسعه‌یافته با هدف کاهش عدم تطابق بین کواریانس نظری با کواریانس واقعی، استفاده شده است. در پژوهشی دیگر هاونگی و همکاران [۲۲]، الگوریتم جدیدی تحت عنوان نروفازی به کمک فیلتر کالمن نامحسوس<sup>۵</sup> ارائه دادند. در این پژوهش نیز هدف کاهش عدم تطابق بین کواریانس نظری با کواریانس واقعی می‌باشد. نتایج نشان می‌دهد که الگوریتم پیشنهادی عملکردی بهتری دارد. در سال ۲۰۱۱ آنکیشان و همکاران [۲۳]، برای حل مسأله مکان‌یابی و نقشه‌کشی ربات از الگوریتم ANFIS به کمک فیلتر کالمن استفاده کردند. در این مقاله آنان با مقایسه روش‌های مختلف نظیر فیلتر کالمن نامحسوس ریشه‌دوم، FAST SLAM2.0 و چند روش دیگر به این نتیجه رسیدند که الگوریتم ANFIS عملکرد بهتری نسبت به سایر روش‌های آموزش دارد. در سال ۲۰۱۷ هاونگی [۲۴]، الگوریتم FAST SLAM2.0 را با استفاده از یک سیستم نروفازی (ANFIS) و الگوریتم ازدحام ذرات<sup>۶</sup> بهبود داده است در این روش که برای محیط ساکن می‌باشد ماتریس کواریانس نویز مشاهدات و فرآیند را با استفاده از ANFIS و بازنمونه برداری را با استفاده از PSO بهبود داده است. نتایج الگوریتم پیشنهاد شده کارآمد می‌باشد. لازم به ذکر است که روش‌های مورد بحث در زمینه‌ی بهبود ماتریس نویز مشاهدات تنها برای محیط‌های ساکن مورد استفاده قرار گرفته است و در محیط پویا تا به حال بررسی نشده است. در این مقاله با دنبال نمودن دو هدف یعنی حل مسأله برای محیط پویا و یافتن بهترین ماتریس نویز مشاهدات از طریق روش تطبیقی سعی در بهبود مسأله‌ی SLAM برای کاربردهای واقعی داریم. از این رو به منظور حل معضل اول مسأله‌ی ردیابی اشیاء پویا در کنار مسأله‌ی SLAM برای فیلتر کالمن توسعه‌یافته استفاده می‌شود و برای معضل دوم از سیستم نروفازی (ANFIS) که توسط الگوریتم ژنتیک آموزش دیده است استفاده می‌نماییم.

<sup>3</sup> Adaptive Neuro-Fuzzy Inference system (ANFIS)

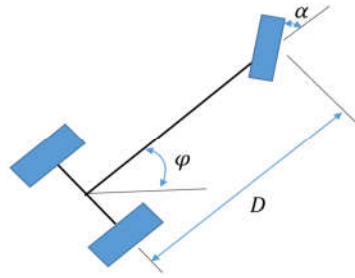
<sup>4</sup> Extended Kalman filter (EKF)

<sup>5</sup> Unscented Kalman Filter (UKF)

<sup>6</sup> particle swarm optimization (PSO)

<sup>1</sup> Detection and Tracking of Moving Objects (DTMO)

<sup>2</sup> Heuristic



شکل ۱- مدل ربات بکار گرفته شده جهت شبیه سازی

شامل وضعیت ربات، نقاط راهنما و اشیاء متحرک موجود در محیط شبیه سازی در آن ذخیره شده است به صورت زیر در نظر گرفته می شود:

$$X(k) = \begin{bmatrix} X_r(k) \\ X_s(k) \\ X_d(k) \end{bmatrix} \in \mathbb{R}^{3+2n+4m} \quad (5)$$

وضعیت ربات شامل موقعیت و جهت گیری آن در محیط می باشد که به صورت  $X_r(x_r, y_r, \theta_r)$  در نظر گرفته می شود. برای مدل نمودن اشیاء ساکن تنها نیاز به موقعیت این اشیاء داریم که برای شیء ساکن نام به صورت  $X_{si}(k)$  نمایش می دهیم. این اشیاء می توانند به بردار وضعیت ربات افزوده شود در این صورت برای n شیء ساکن در محیط داریم:

$$X_s(k) = \begin{bmatrix} x_{s1}(k) \\ y_{s1}(k) \\ x_{s2}(k) \\ y_{s2}(k) \\ \vdots \\ x_{sn}(k) \\ y_{sn}(k) \end{bmatrix}, i = 1, 2, \dots, n. \quad (6)$$

اما به منظور مدل کردن اشیاء پویا نیاز به موقعیت و سرعت اشیاء است. شیء متحرک لام به صورت  $X_{dj}$  نمایش داده می شود. در صورتی که در محیط m تا شیء متحرک وجود داشته باشد به صورت رابطه زیر به بردار اصلی وضعیت ربات مطابق رابطه ی اضافه می شوند.

$$X_d(k) = \begin{bmatrix} \begin{bmatrix} x_{d1}(k) \\ y_{d1}(k) \\ \dot{x}_{d1}(k) \\ \dot{y}_{d1}(k) \end{bmatrix}^T & \begin{bmatrix} x_{d2}(k) \\ y_{d2}(k) \\ \dot{x}_{d2}(k) \\ \dot{y}_{d2}(k) \end{bmatrix}^T & \dots & \begin{bmatrix} x_{dm}(k) \\ y_{dm}(k) \\ \dot{x}_{dm}(k) \\ \dot{y}_{dm}(k) \end{bmatrix}^T \end{bmatrix}^T \quad (7)$$

$j = 1, 2, \dots, m$

به طوریکه  $x_{dj}(k)$  و  $y_{dj}(k)$  نمایش دهنده ی موقعیت،  $\dot{x}_{dj}$  و  $\dot{y}_{dj}$  نشان دهنده ی سرعت در مختصات دکارتی شیء پویا نام می باشد. حرکت اشیاء پویا در محیط توسط یک مدل گسسته در مختصات دکارتی (مدل سرعت ثابت برای تخمین حرکت اشیاء در صفحه ی دو بعدی) به صورت زیر در نظر گرفته شده است.

$$X_{dj}(k+1) = A(\Delta t)X_{dj}(k) + v_d(k), j = 1, 2, \dots, m \quad (8)$$

در این مدل شتاب دنبال کردن شیء به صورت نویز فرایند مدل شده است. ماتریس A ماتریس انتقال برای زمان نمونه برداری  $\Delta t$  نویز  $v_d$  فرایند با ماتریس کواریانس  $Q_d$  است که به صورت گوسی مدل می شود. ماتریس کواریانس به صورت زیر می باشد:

$$Q_d = \sigma_Q^2 \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^2}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^2}{2} \\ \frac{\Delta t^2}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^2}{2} & 0 & \Delta t^2 \end{bmatrix} \quad (9)$$

## ۲- SLAM و DATMO

ربات را در نظر بگیرید که در یک محیط ناشناخته بدون هیچ دانش قبلی از محیط حرکت می کند. این ربات همزمان با اجرای فرمان کنترلی، توسط حسگرها در حال جمع آوری مشاهداتی از محیط است که دارای اغتشاش می باشد. مکان یابی و نقشه کشی همزمان (SLAM) فرایندی است که به باز یابی توصیف محیط و مسیر ربات به صورت به هنگام با استفاده از ورودی کنترلی و مشاهدات جمع آوری شده می پردازد [۲۵].

همانطور که قبلاً نیز ذکر شد، ناوبری در یک محیط واقعی در حضور اشیاء متحرک همچنان یک مشکل برای روبات های خودمختار است. پیاده سازی مسأله ی SLAM به همراه الگوریتم DATMO به ربات کمک می کند تا دانش کاملی از محیط اطرافش داشته باشد. در ادامه ی این بخش، مدل سینماتیکی یک ربات متحرک و مدل اشیاء ثابت و متحرک توصیف می شود.

از فیلتر کالمن توسعه یافته جهت خطی سازی سیستم ها استفاده می شود. در ادامه مسأله ی SLAM توسط فیلتر کالمن توسعه یافته به صورت کلی شرح داده می شود که جزئیات آن در [۲۶] ارائه شده است. شکل کلی مدل حرکتی و مدل مشاهدات برای مسأله به صورت زیر در نظر گرفته می شود.

$$X_k = f(X_{k-1}, u_k) + q_k, Z_k = h(X_k) + r_k \quad (1)$$

که  $f(0,0)$  و  $h(0)$  به ترتیب توصیف کننده ی مدل حرکتی ربات و مدل مشاهدات است. همچنین  $q_k$  و  $r_k$  بیانگر نویزهای گوسی سفید با میانگین صفر می باشند. در این صورت معادلات فیلتر کالمن توسعه یافته برای مدل تقریبی، به صورت زیر خواهد بود.

$$\begin{aligned} \bar{X}_k^- &= f(\bar{X}_{k-1}^+, u_k) + q_k \\ P_k^- &= \nabla f_x P_{k-1}^+ \nabla f_x^T + \nabla f_u Q_k \nabla f_u^T \\ K_k &= P_k^- \nabla g_x^T S_k^{-1} \\ \bar{X}_k^+ &= \bar{X}_k^- + K_k u_k \\ P_k^+ &= P_k^- - K_k S_k K_k^T \end{aligned} \quad (2)$$

که در آن

$$\begin{aligned} v_k &= Z_k - h(\bar{X}_k^-, 0) \\ S_k &= \nabla h_x P_k^- \nabla h_x^T + \nabla h_w R \nabla h_w^T \\ K_k &= P_k^- \nabla h_x^T S_k^{-1} \end{aligned} \quad (3)$$

که  $v_k$  باقی مانده ی تخمین و  $\nabla h_x, \nabla f_u, \nabla f_x$  به ترتیب، ماتریس های جاکوبی مدل حرکتی، نویز مدل حرکتی، مدل مشاهدات و نویز مدل مشاهدات هستند.

### ۲-۱- مدل فرآیند غیر خطی و ویژگی ها

ربات استفاده شده جهت شبیه سازی مطابق شکل ۱ می باشد. مدل غیرخطی ربات که موقعیت را از داده های ورودی اندازه گیری شده محاسبه می کند به صورت رابطه ی می باشد.

$$X_r(k+1) = \begin{bmatrix} x_r(k) + \Delta S * \cos(\varphi + \alpha) \\ y_r(k) + \Delta S * \cos(\varphi + \alpha) \\ \varphi_r(k) + \Delta S * \frac{\sin(\alpha)}{D} \end{bmatrix} \quad (4)$$

بطوریکه D فاصله ی بین دو چرخ،  $\Delta S = V * dt$  جابجایی بین دو گام متوالی و  $\alpha$  زاویه ی فرمان می باشد. بردار حالت ربات در نظر گرفته

$$Z_i(k) = [z_p(k) \quad z_0(k)] = g_2^{-1}(g_1^{-1}(x_v, x_{mi})) \quad (15)$$

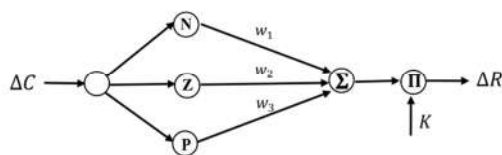
بطوریکه  $g_1^{-1}$  و  $g_2^{-1}$  معکوس توابع موجود در معادلات ۱۱ و ۱۰ است که به صورت زیر محاسبه می‌شود.

$$g_2^{-1} = \left[ \begin{array}{c} \sqrt{(x_i - x_v)^2 + (y_i - y_v)^2} \\ \arctan\left(\frac{y_i - y_v}{x_i - x_v}\right) - \varphi_v \end{array} \right] \quad (16)$$

$$g_1^{-1} = R_n^b(k)[x_{mi}^T(k) - P^n(k)]$$

### ۳-۲- ساختار نو-فازی بکارگرفته شده

معمولا برای حل مسائل مشخصات آماری از پیش‌دانسته شده نویز فرآیند و اندازه‌گیری معلوم فرض می‌شود. در حالی که این مشخصات در کاربردهای واقعی نامشخص و حتی ممکن است با زمان تغییر کند. اطلاعات از پیش دانسته شده نادرست نویز فرآیند و اندازه‌گیری ممکن است سبب کاهش عملکرد، واگرایی عملی و ناسازگاری شود. یک راه موثر برای غلبه بر این ضعف استفاده از الگوریتم‌های تطبیقی می‌باشد.



شکل ۳- سیستم نو-فازی بکار گرفته شده

مزیت الگوریتم تطبیقی این است که موجب سازگار شدن کواریانس‌های محاسبه شده با کواریانس واقعی می‌شود. در تحقیقات اخیر در مورد برای تطبیق خانواده‌ی کالمن فیلتر دو رویکرد اصلی وجود دارد. تخمینگر تطبیقی چند مدله<sup>۱</sup> و تخمینگر تطبیقی ابداعی<sup>۲</sup>. اگر چه اجرای این رویکردها کاملا متفاوت هستند، اما هر دو رویکرد مفهوم استفاده از اطلاعات آماری جدید را با استفاده از نوآوری (یا باقی مانده) توالی به اشتراک می‌گذارد [۲۷] در این مقاله از روش تخمینگر تطبیقی ابداعی برای فیلتر کالمن توسعه یافته استفاده شده است. جهت محاسبه‌ی عدم تطابق بین کواریانس نظری و واقعی داریم:

$$\Delta C_k = C_k - S_k \quad (17)$$

که در آن کواریانس واقعی ( $C_k$ ) با استفاده از رابطه‌ی  $C_k = vv^T$  و کواریانس نظری ( $S_k$ ) با استفاده از رابطه ۳ محاسبه می‌شود. سپس ماتریس بدست آمده را نرمالایز کرده و به عنوان ورودی، به شبکه‌ی ANFIS داده می‌شود، خروجی آن  $\Delta R$  به صورت رابطه ۱۲ به ماتریس اصلی  $R$  تطبیق داده می‌شود.

$$R_k = R_{k-1} + \begin{bmatrix} \Delta\sigma_r^2 & 0 \\ 0 & \Delta\sigma_\theta^2 \end{bmatrix} \quad (18)$$

هدف کاهش عدم تطابق بین کواریانس واقعی و کواریانس نظری می‌باشد برای این منظور در ادامه از یک سیستم تک ورودی تک خروجی الگوریتم نو فازی برای این منظور استفاده می‌نماییم. ایده‌ی اولیه برای تطبیق ماتریس به این صورت است که افزایش و کاهش  $R_k$  باعث افزایش و کاهش  $S_k$  در معادله‌ی (۳) می‌شود. بنابراین  $R_k$

و ماتریس انتقال نیز به صورت زیر در نظر گرفته می‌شود:

$$A(\Delta t) = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (10)$$

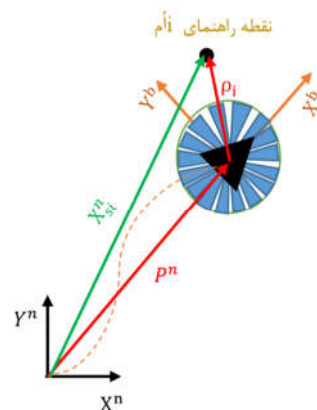
به طوریکه  $\sigma_0^2$  انحراف استاندارد از نویز فرآیند می‌باشد. در نهایت معادله‌ی نهایی سینماتیکی را می‌توان به صورت زیر در نظر گرفت.

$$X(k+1) = \begin{bmatrix} F & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & T \end{bmatrix} X(k) + \begin{bmatrix} v_r(k) \\ 0 \\ v_d(k) \end{bmatrix} \quad (11)$$

به طوریکه  $F$  ماتریس ژاکوبین می‌باشد که از خطی‌سازی مدل سینماتیکی ربات بدست می‌آید،  $v_r(k)$  نویز فرآیند بعد از خطی‌سازی و اماتریس واحد می‌باشد.

### ۲-۲- مدل مشاهدات

قبل از محاسبه‌ی مدل مشاهده، ارتباط غیر خطی بین مشاهده حسگر و موقعیت نقطه راهنمای باید بررسی شود. فاصله و جهت در ارتباط با نقطه‌ی راهنما در مختصات متصل به حسگر توسط حسگر فراهم می‌شود. ارتباط بین فاصله و جهت در مختصات حسگر و موقعیت نقطه‌ی راهنما در مختصات ناوبری در شکل ۲ نشان داده شده است. هدف بدست آوردن معادله‌ی غیرخطی موقعیت نقطه راهنما از حالت ربات و اطلاعات نقشه می‌باشد.



شکل ۲- ارتباط نقاط راهنما و ربات در دستگاه ناوبری

فاصله و جهت توسط حسگر در ارتباط با ربات و نقطه‌ی راهنما ارائه می‌شود که یک مدل غیرخطی به صورت زیر می‌باشد:

$$Z(k) = h(x(k), v(k)) \quad (12)$$

فاصله و جهت مشاهده شده در مختصات حسگر،  $Z^T = [\rho, \theta]^T$  در ارتباط با آمین موقعیت نقطه راهنما  $X_{si}^s$  به صورت یک تابع غیر خطی به فرم زیر می‌توان بیان نمود:

$$x_{mi}^n(k) = g_1(x(k), z(k)) = P^n(k) + R_n^b(k) p_{mb}^b(Z^b(k)) \quad (13)$$

به طوری که  $P^n$  موقعیت در مختصات ناوبری،  $p_{mb}^b$  فاصله‌ی مختصات کارترین نقطه‌ی راهنما از حسگر در مختصات جسم  $R_n^b(k)$  ماتریس انتقال از مختصات جسم به مختصات ناوبری می‌باشد. با توجه به تبدیل مختصات از قطبی به دکارتی داریم:

$$p_{mb}^b = g_2(Z^b(k)) = \begin{bmatrix} \rho \cos(\theta) \\ \rho \sin(\theta) \end{bmatrix} \quad (14)$$

در نهایت مدل مشاهدات به صورت زیر مورد استفاده قرار می‌گیرد:

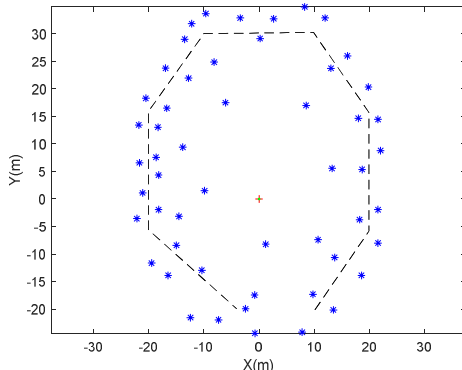
<sup>1</sup>multiple model adaptive estimation (MMAE)

<sup>2</sup>innovation adaptive estimation (IAE)

آموزش داده می‌شود سپس از پارامترهای موجود استفاده می‌شود.

جدول ۱- پارامترهای بکار رفته در الگوریتم ژنتیک

ردیف	توصیف پارامترها	مقدار پارامترها
۱	تعداد تکرار	۴۰۰
۲	جمعیت کل	۶۰
۳	درصد همگرایی	۰/۷
۴	درصد جهش	۰/۵
۵	نسبت جهش	۰/۱



شکل ۵- محیط در نظر گرفته شده جهت شبیه‌سازی

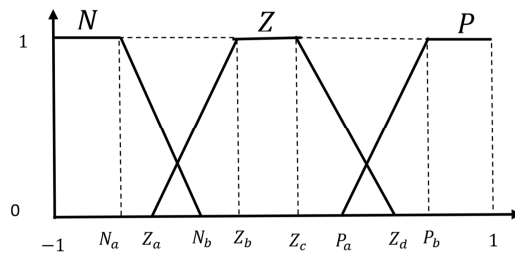
### ۳- نتایج

به منظور ارزیابی عملکرد سیستم پیشنهادی، مسأله‌ی SLAM را برای یک محیط پویا که شامل اشیاء ساکن و پویا هستند شبیه‌سازی می‌کنیم. محیط در نظر گرفته شده برای شبیه‌سازی به این صورت می‌باشد که فرض شده است دو شیء متحرک در محیط قرار دارند و در مسیریابی حرکت می‌کنند، حال هدف ما نقشه‌کشی و مکان‌یابی همزمان توسط ربات و همچنین تعیین و دنبال کردن اشیاء متحرک می‌باشد. با این عمل درک ربات در از محیط اطراف خود افزایش یافته و برای عملکرد خودمختار قادر به انجام تصمیم‌گیری مطمئن‌تری می‌شود. لازم به ذکر است که برخی از منابع کدهای اولیه‌ی مورد استفاده در شبیه‌سازی در [۲۸] موجود می‌باشد، که بعد از تغییر و بهبود در روش پیشنهادی این مقاله مورد استفاده قرار گرفته است. جهت شبیه‌سازی محیطی مطابق شکل ۵ طراحی شده است که مسیر و نقاط راهنمای موجود را نشان می‌دهد. نقاط ستاره (\*) موقعیت نقاط راهنما را در نقشه نشان می‌دهد که به صورت ساکن هستند. خط چین نشان داده شده نیز مسیر حرکت ربات را نشان می‌دهد، مسیری که ربات در حالت ایده‌آل باید حرکت کند با توجه به دینامیک حرکتی آن (بخش ۱-۲) می‌باشد. فرضیات در نظر گرفته شده به منظور شبیه‌سازی به این صورت می‌باشد، سرعت ربات برابر  $1.5 \frac{m}{s}$  و نرخ تغییرات چرخش ربات  $20 * \frac{\pi}{180}$  می‌باشد. فاصله‌ی بین محور دو چرخ برابر  $D = 0.2(m)$  فاصله زمانی بین سیگنال‌های کنترل برابر  $0.025$  ثانیه، اغتشاشات بر روی ورودی سرعت  $\sigma_v = 0.3 \frac{m}{s}$  و ورودی سرعت زاویه‌ای  $\sigma_\omega = 3 * \frac{\pi}{180}$  مشاهده‌ات توسط حسگر لیدار به این صورت انجام می‌گیرد که بیشترین فاصله‌ی قابل تشخیص برابر ۳٫۵ متر و هر

می‌تواند برای کاهش عدم تطابق استفاده می‌شود. قوانین کلی تطبیقی به صورت اگر- آنگاه به صورت زیر می‌باشد.

- اگر  $\Delta R_k = w_1$  منفی شود آنگاه  $\Delta C_k = w_1$
- اگر  $\Delta R_k = w_2$  صفر شود آنگاه  $\Delta C_k = w_2$
- اگر  $\Delta R_k = w_3$  مثبت شود آنگاه  $\Delta C_k = w_3$

که  $w_1$ ،  $w_2$  و  $w_3$  مقادیر متناسب با مقدار عدم تطابق است و از طریق آموزش بدست می‌آید. ساختار ANFIS معادل، مطابق شکل ۳ است که از چهار لایه تشکیل شده است. در لایه‌ی اول که لایه‌ی ورودی است، ماتریس عدم تطابق کواریانس واقعی و نظری به شبکه‌ی ANFIS وارد شده و با توجه به توابع عضویت تعریف شده مطابق شکل ۴، عمل فازی‌سازی را انجام می‌شود (لایه‌ی دوم). پس از آن، براساس قوانین ذکر شده مقادیر وزنی، در درصدتعلق ضرب شده و عمل نافازی‌سازی انجام می‌شود (لایه‌ی سوم)، در نهایت،  $(\Delta R_k)$  در ضرب  $k$  ضرب شده که خروجی شبکه‌ی ANFIS می‌باشد (لایه‌ی چهارم).



شکل ۴- سیستم نرو-فازی بکار گرفته شده

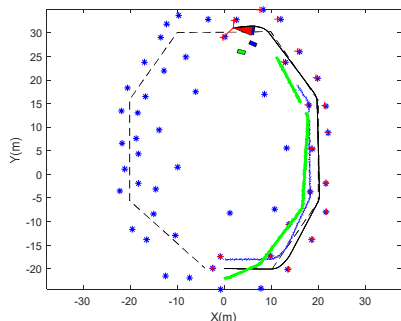
به منظور تعیین پارامترهای مناسب سیستم یعنی، پارامترهای توابع عضویت، وزن‌ها و ضریب خروجی، باید سیستم نروفازی آموزش داده شود. با این حال، آموزش در مدل با ناظر معمولی انجام نمی‌شود، زیرا سیستم به اندازه کافی شناخته شده نیست. بدین ترتیب، روش پس انتشار خطا و روش‌های آموزشی عادی نمی‌تواند مورد استفاده قرار گیرد. در این پژوهش از روش مرسوم الگوریتم ژنتیک (GA) برای آموزش الگوریتم (ANFIS) استفاده شده است. در ادامه به بررسی این روش می‌پردازیم.

الگوریتم ژنتیک یک روش جستجوی کلی و تصادفی است که برپایه‌ی نظریه تکامل شکل گرفته است. در دهه‌ی اخیر استفاده از الگوریتم ژنتیک در علوم مختلف به دلیل عدم نیاز به آنالیز کلاسیک و فرمول‌بندی مسأله افزایش چشم‌گیری داشته است. مواردی که در الگوریتم ژنتیک باید به آن توجه ویژه شود، انتخاب صحیح عملگرها و تابع برازش می‌باشد. نحوه‌ی اجرای الگوریتم GA جهت آموزش الگوریتم ANFIS به این صورت می‌باشد که ورودی داده‌های آموزش و خروجی سیستم‌فازی بهینه می‌باشد.

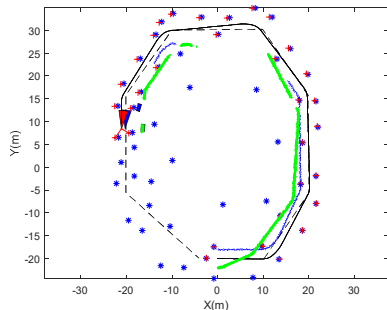
در این الگوریتم برای عملگرهای انتخاب و همگرایی به ترتیب از روش های مرسوم چرخ رولت و متوسط‌گیری استفاده شده است. تابع برازش براساس نسلی که کمترین خطا را ارائه می‌دهد ارزیابی می‌شود. پارامترهای بکار رفته در الگوریتم ژنتیک در جدول ۲ نشان داده شده است. هدف بدست آوردن پارامترهای بهینه است که ذره‌ی مورد نظر در الگوریتم ژنتیک را ایجاد می‌کند. این ذره به طول ۱۲ می‌باشد، که برای هر محیط  $(N_a, N_b, Z_a, Z_b, Z_c, Z_d, P_a, P_b, w_1, w_2, w_3, k)$

دو بار یکبار مشاهده انجام می‌گیرد و دارای اغتشاشات  $\sigma_R = 0.2 \text{ m}$  و  $\sigma_s = 1 * \frac{\pi}{180}$  می‌باشد. برای مسیر ربات این فرض در نظر گرفته شده است که موقعیت ربات در مختصات  $X_r(0) = [-1.9, -4.9]^T$  و زاویه آن با محور مختصات ناوبری برابر صفر می‌باشد. در صورت که فاصله‌ی ربات تا نقاط راهنمای مسیر کمتر از ۰.۲ متر شود، نقطه‌ی راهنمای بعدی را به عنوان هدف در نظر گرفته و به سمت آن حرکت می‌کند.

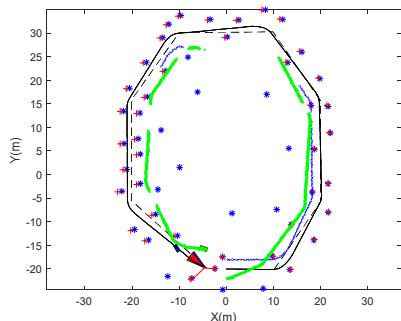
با اعمال فرضیات ذکر شده، شبیه‌سازی الگوریتم پیشنهادی مطابق شکل ۶ بدست می‌آید. در این شکل دو شیء در محیط با سرعت  $1.3 \frac{m}{s}$  در حرکت می‌باشند که توسط رباتی که در محیط قرار دارد، در حال ردیابی هستند. مسیر سبز رنگ مسیر تخمین زده‌شده‌ی شیء شماره ۲ و مسیر آبی رنگ، مسیر تخمین زده‌شده‌ی شیء شماره ۱ را نشان می‌دهد. البته در صورتی که اشیاء در دید حسگر باشد مسیر آن تخمین زده می‌شود. علت اینکه در برخی نقاط مسیر تخمین زده شده از اشیاء متحرک وجود ندارد این است که این شیء از دیده حسگرهای ربات خارج می‌شود و با دوباره دیده شدن این شیء مسیر آن تخمین زده می‌شود. موقعیت تخمین زده‌شده‌ی ربات و اشیاء متحرک که در حال حرکت می‌باشند مطابق شکل ۶ است. ربات در یک محیط ناشناخته با اشیاء ساکن و پویای مختلف شروع به حرکت می‌کند. در انتها همانطور که مشاهده می‌شود عملکرد الگوریتم پیشنهادی تخمین خوبی از موقعیت ربات و اشیاء را فراهم می‌کند. شکل ۷ RMSE<sup>۱</sup> موقعیت ربات و اشیاء متحرک در محیط نشان داده شده است. با توجه به نمودار مشاهده می‌شود که این خطا حد قابل قبولی کنترل شده است و نشان از عملکرد خوب الگوریتم پیشنهادی است. در شکل ۷ سرعت تخمینی اشیاء مشاهده می‌شود، علت اختلاف ناگهانی سرعت شیء از مقدار ایده‌آل آن، خارج شدن شیء متحرک از دید حسگر می‌باشد. زیرا با خارج شدن آن از بردار حالت حذف می‌شود.



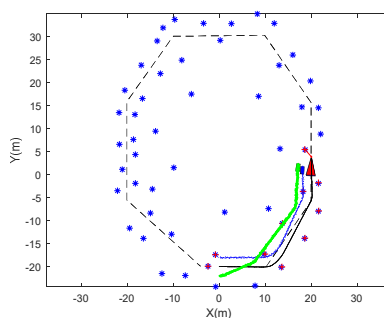
ب



ج



د



الف

شکل ۶- مراحل اجرای الگوریتم EKF بهبودیافته برای محیط‌های پویا.

(الف) گام ۱۰۰۰ (ب) گام ۲۰۰۰ (ج) گام ۳۰۰۰ (د) گام ۳۹۴۳

"+" نقاط راهنمای تخمین زده شده، "..." سیاه رنگ مسیر تخمین

زده شده ربات، "..." نازک (آبی رنگ) مسیر تخمین زده شده‌ی شیء

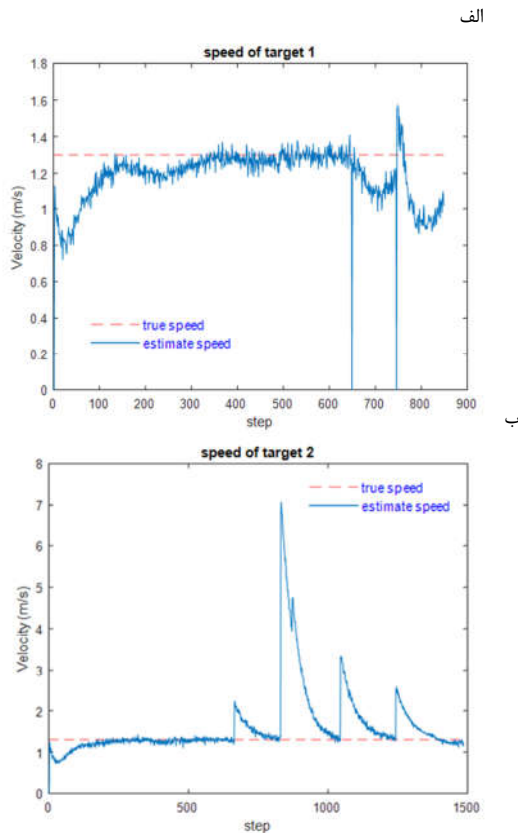
متحرک شماره یک و "..." ضخیم (سبز رنگ) مسیر تخمین زده

شده‌ی شیء متحرک شماره دو.

(الف)  $x=0$  (ب)  $x=1$

<sup>۱</sup> Root Mean Square Error (RMSE)

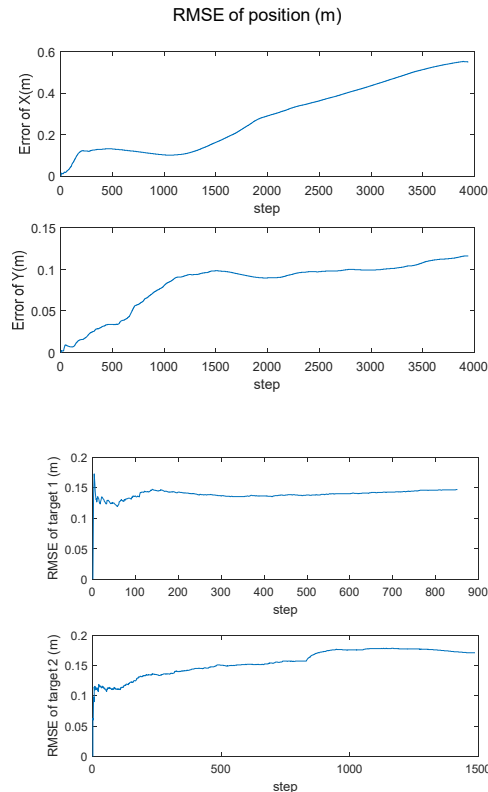
مجزا در نظر می‌گیرد که در این مقاله موقعیت و سرعت اشیاء متحرک به بردار حالت افزوده می‌شوند.



شکل ۸- سرعت واقعی و تخمین زده شده‌ی اشیاء متحرک در محیط. (الف) شیء شماره یک (ب) شیء شماره دو.

- تنظیم ماتریس کواریانس مشاهدات: ثابت و دانسته فرض کردن ماتریس کواریانس عامل اصلی واگرایی و کاهش دقت در نقشه‌ی تولیدی می‌باشد. در این مقاله با استفاده از روش ANFIS این ماتریس بهبود داده شده است. لازم به ذکر است با وجود مزایای ذکر شده‌ی الگوریتم پیشنهادی، دارای برخی چالش‌های نیز می‌باشد. یکی از این چالش‌ها، هزینه‌ی محاسباتی برای محیط‌های شامل اشیاء پویای با تعداد بالا می‌باشد. به دلیل اضافه شدن  $n$  شیء متحرک بردار حالت ربات  $4n$  افزایش می‌باشد و در نهایت ماتریس کواریانس آن، یک ماتریس  $4n * 4n$  می‌شود که هزینه‌ی محاسباتی آن برای محیط‌های با تراکم بالا، قابل توجه است. که به عنوان کارهای آینده می‌توان به آن توجه نمود. همچنین چالش دیگر را می‌توان کاهش عملکرد الگوریتم پیشنهادی برای حل مسأله‌ی SLAM برای چند ربات دانست زیرا همبستگی بین اشیاء متحرک و اشیاء ساکن برای چند ربات وقتی به منظور انتقال نقشه به یکدیگر نزدیک می‌شوند موضوع پیچیده‌ای می‌باشد.

الف



شکل ۷- خطای RMSE موقعیت ربات و اشیاء متحرک. (الف) خطای میانگین مربعات موقعیت ربات (ب) خطای میانگین مربعات موقعیت اشیاء متحرک در محیط.

ب

#### ۴- نتیجه‌گیری

در این مقاله با ارائه‌ی یک روش جدید مسأله‌ی EKF-SLAM به همراه اشیاء متحرک در محیط پیاده‌سازی شد. در این راستا با توجه به اینکه حل مسأله SLAM توسط الگوریتم‌های بی‌زین و به ویژه الگوریتم فیلتر کالمن توسعه‌یافته همواره با چالش ثابت فرض نمودن ماتریس کواریانس مشاهدات و فرآیند و ساکن بودن محیط آزمایش همراه است. در این مقاله با بکارگیری همه‌ی روش‌های زیر در یک مسأله سعی در از بین بردن ضعف هر یک از موارد ذکر شده را داشتیم. این موارد به صورت خلاصه به شرح زیر می‌باشد:

- محیط: عموماً در مسأله‌ی SLAM محیط را ساکن در نظر می‌گیرند که فرض بسیار نامناسبی است، زیرا محیط واقعی شامل اشیاء پویایی هستند که دائم در حال جابه‌جایی هستند. ردیابی این اشیاء و تخمین موقعیت و سرعت آنها توسط ربات باعث می‌شود تا ربات درک کاملی نسبت به محیط اطراف خود داشته باشد، و بتواند عملکرد صحیحی داشته باشد. همچنین این روش باعث پایداری و افزایش دقت در نقشه‌ی تولیدی را به همراه دارد.
- بیان مسأله در قالب یک بردار حالت: روش‌های موجود برای محیط‌های پویا عموماً اشیاء پویا را از حالت ربات، به صورت

## ۵- مراجع

- Dynamic Environments. In *IEEE Information Science and Applications (ICISA)*, pp. 1-4, , 2013.
- [18] Bahraimi M. S., Bozorg M., and Rad A. B. SLAM in dynamic environments via ML-RANSAC. *Mechatronics*, Vol. 49, pp. 105-118, 2018.
- [19] Chatterjee, A. and Fumitoshi M., A neuro-fuzzy assisted extended Kalman filter-based approach for simultaneous localization and mapping (SLAM) problems. *IEEE Transactions on Fuzzy Systems*, Vol. 15, No. 5, 2007.
- [20] Chatterjee, A., and Fumitoshi M., A Geese PSO tuned fuzzy supervisor for EKF based solutions of simultaneous localization and mapping (SLAM) problems in mobile robots. *Expert Systems with Applications*, Vol. 37, No. 8, 2010.
- [21] Havangi R., Nekoui M. A., and Mohammad Teshnehlab., Adaptive neuro-fuzzy extended Kaiman filtering for robot localization. In *14th International Conference (EPE/PEMC) on Power Electronics and Motion Control*, 2010.
- [22] Havangi R., Teshnehlab M A., and Nekoui M. A., A novel adaptive neuro-fuzzy unscented kalman filter for SLAM. *International Journal of Humanoid Robotics* Vol. 8, No. 01, 2011.
- [23] Ankişhan H., and Efe M., Adaptive neuro fuzzy supported Kalman filter approach for simultaneous localization and mapping. In *19th Conference on Signal Processing and Communications Applications (SIU)*, IEEE, pp. 266-270., 2011.
- [24] Havangi R., Improved FastSLAM2. using ANFIS and PSO. *Automatika*, Vol. 57, No. 4, 2016.
- [25] Montemerlo M., and Thrun S., FastSLAM: A scalable method for the simultaneous localization and mapping problem in robotics. Vol. 27, 2007.
- [26] Bailey T., Mobile robot localisation and mapping in extensive outdoor environments. Phd. Thesis, University of Sydney, 2002.
- [27] Loebis D., Sutton R., Chudley J. and Naeem W., Adaptive tuning of a Kalman filter via fuzzy logic for an intelligent AUV navigation system. *Control engineering practice*, Vol. 12, No. 12, 2014.
- [28] Bailey T. and Juan N., [Online] Available: [www-personal.acfr.usyd.edu.au/tbailey/software/index.html](http://www-personal.acfr.usyd.edu.au/tbailey/software/index.html), 2004.
- [1] نوروژ، م.، ابراهیمی، م.، ارباب میر، م.، مروری بر روش‌های تلفیق سیستم ناوبری اینرسی و موقعیت یاب جهانی و بررسی رویکردهای نو در این زمینه. مهندسی مکانیک دانشگاه تبریز، دوره ۴۸، شماره ۳ - شماره پیاپی صفحه ۳۶۵-۳۶۹، ۱۳۹۷.
- [2] Durrant-Whyte H., and Bailey T., Simultaneous localization and mapping: part I. *IEEE robotics & automation magazine* Vol. 13, No. 2, 2006.
- [3] Montemerlo M., Sebastian T., Daphne K., and Wegbreit B., FastSLAM: A factored solution to the simultaneous localization and mapping problem. *Aaai/iaai*, 2002.
- [4] Bailey T. and Durrant-Whyte H., Simultaneous localization and mapping (SLAM): Part II. *IEEE Robotics & Automation Magazine*, Vol. 13, No. 3, 2006.
- [5] Karlsson N., Di Bernardo E., Ostrowski J., Goncalves L., Pirjanian P. and Munich M., The vSLAM algorithm for robust localization and mapping. In *Proceedings of the IEEE International Conference on Robotics and Automation, ICRA*, 2005.
- [6] Lin M., Yang C. and Li D., An improved transformed unscented FastSLAM with adaptive genetic resampling, *IEEE Transactions on Industrial Electronics* Vol. 66, No. 5 pp. 3583-3594, 2018.
- [7] Tardós J. D., Neira J., Paul M. N. and Leonard J., Robust mapping and localization in indoor environments using sonar data. In *21th International Journal of Robotics Research*, No. 4, 2002.
- [8] Xi W., Ou Y., Peng J. and Yu G., A new method for indoor low-cost mobile robot SLAM. In *IEEE International Conference on Information and Automation (ICIA)*, 2017.
- [9] Sankrit H., Panwala B., Mudgal P. and Patil C., Indoor SLAM using Kinect Sensor. In *2th International Journal of Science Technology and Engineering (IJSTE)*, No. 10, 2016.
- [10] Shoudong H., and Gamini D., Convergence and consistency analysis for extended Kalman filter based SLAM. *IEEE Transactions on robotics*, Vol. 23, No. 5, 2007.
- [11] Mur-Artal R., Montiel J. M. and Tardos J., ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, Vol.31, No. 5, 2015.
- [12] Whelan T., Kaess M., Johannsson H., Fallon M., Leonard J., and McDonald J., Real-time large-scale dense RGB-D SLAM with volumetric fusion. *The International Journal of Robotics Research*, Vol. 34, No. 4-5, 2015.
- [13] Wang C., Thorpe C., Simultaneous localization and mapping with detection and tracking of moving objects. In *Proceedings 2002 IEEE International Conference on Robotics and Automation*, 2002.
- [14] Bobruk J., and Austin D., Laser motion detection and hypothesis tracking from a mobile platform. In *Conference on Robotics & Automation*, 2004.
- [15] Wolf D. F., and Sukhatme G., Mobile robot simultaneous localization and mapping in dynamic environments. *Autonomous Robots*, Vol. 19, No. 1, 2005.
- [16] اسپهبدی نیا، س.، و خان میرزا، ا.، الگوریتم C-SLAM تطبیقی در محیط دینامیک. مجله مهندسی مکانیک دانشگاه تبریز، دوره ۵۰، شماره ۱- شماره پیاپی ۹۰، صفحه ۹-۱۵، ۱۳۹۹
- [17] Oh S., Hahn M. and Kim J., Simultaneous Localization and Mapping for Mobile Robots in