



## نقشه برداری و مکان‌یابی هم‌زمان با استفاده از داده لیزر و روش نقشه برداری سریع بدون ردیابی با انطباق پویایی

مرضیه زمانی علویجه<sup>1</sup>، شهرام هادیان جزی<sup>2\*</sup>

1- دانشجوی کارشناسی ارشد، مهندسی مکترونیک، دانشگاه اصفهان، اصفهان

2- استادیار، مهندسی مکانیک، دانشگاه اصفهان، اصفهان

\* اصفهان، صندوق پستی 8174673441، s.hadian@eng.ui.ac.ir

### اطلاعات مقاله

مقاله پژوهشی کامل

دریافت: 19 بهمن 1394

پذیرش: 30 اردیبهشت 1395

ارائه در سایت: 02 تیر 1395

کلید واژگان:

مکان‌یابی و نقشه برداری هم‌زمان

الگوریتم سریع بدون ردیابی

لیزر

انطباق پویایی

### چکیده

مکان‌یابی و نقشه برداری هم‌زمان یکی از مسائل بنیادی در رباتیک است و تاکنون الگوریتم‌های بسیاری از جمله الگوریتم نقشه برداری سریع که بسیار پرکاربرد است و الگوریتم نقشه برداری سریع بدون ردیابی که جدیدتر است، برای حل این مسئله به کار گرفته شده‌اند. اگرچه در مقایسه‌های علمی، عملکرد این الگوریتم بهتر از الگوریتم سریع ارزیابی شده است، هنوز ظرفیت‌های بررسی نشده زیادی در رابطه با این الگوریتم وجود دارد. لذا در این مقاله به موضوع ارایه اصلاحات برای بهبود الگوریتم سریع بدون ردیابی و بررسی عملکرد آن پرداخته شده است. یکی از راه‌های افزایش دقت تخمین موضع ربات، مقایسه و تطبیق دو اندازه‌گیری متوالی و اصلاح موضع ربات به نحوی که بهترین تطابق میان دو اندازه‌گیری برقرار شود، است. این روش که انطباق پویایی نامیده می‌شود برای بهبود عملکرد الگوریتم‌ها موثر بوده ولی تاکنون برای الگوریتم سریع بدون ردیابی مورد آزمون قرار نگرفته است. در این مقاله برای اولین بار از ترکیب الگوریتم سریع بدون ردیابی با انطباق پویایی برای مکان‌یابی و نقشه برداری هم‌زمان استفاده می‌شود. برای ارزیابی تاثیر انطباق پویایی، این الگوریتم روی دو مجموعه داده لیزر به دست آمده در محیط شبیه‌سازی و آزمایش‌های عملی، پیاده‌سازی و نتایج با حالت بدون انطباق پویایی مقایسه می‌شوند. نتایج مقایسه نشان می‌دهند که افزودن انطباق پویایی به الگوریتم سریع بدون ردیابی، دقت تخمین مسیر و دقت نقشه را به مقدار قابل ملاحظه‌ای بهبود می‌دهد. همچنین با بهبود سرعت الگوریتم انطباق داده، زمان اضافی لازم برای انجام انطباق پویایی جبران شده و سرعت کلی الگوریتم سریع بدون ردیابی بهبود یافت.

## Simultaneous Localization and Mapping Using Laser Data and Unscented FastSLAM with Scan Matching

Marzieh Zamani Alavijeh, Shahram Hadian Jazi\*

Department of Mechanical Engineering, University of Isfahan, Isfahan, Iran.

\* P.O.B. 8174673441, Isfahan, Iran, s.hadian@eng.ui.ac.ir

### ARTICLE INFORMATION

Original Research Paper  
Received 08 February 2016  
Accepted 19 May 2016  
Available Online 22 June 2016

#### Keywords:

Simultaneous localization and mapping  
Unscented FastSLAM algorithm  
Laser  
Scan Matching

### ABSTRACT

Simultaneous localization and mapping (SLAM) is a fundamental problem in autonomous robotics. Many algorithms have been exploited to solve this problem, among these algorithms, FastSLAM is one of the most widely used and Unscented FastSLAM is one of the newest. Although in several scientific researches it is stated that Unscented FastSLAM outperforms FastSLAM, there are still unexamined potentials regarding Unscented FastSLAM. Therefore, this paper seeks to improve the overall performance of Unscented FastSLAM. Map accuracy and quality directly depend on the accuracy of localization and observations. In SLAM algorithms, robot pose is predicted using motion model, and then corrected using the difference between map features and recently observed features. Accuracy of pose estimation may improve by comparing two sequential observations and modifying robot pose to result in best match between them. This method is called scan matching and has been successfully combined with FastSLAM algorithm and some other SLAM algorithms not including Unscented FastSLAM. Therefore, this paper seeks to investigate the performance of Unscented FastSLAM combined with scan matching. Simulation results show that combining Unscented FastSLAM with scan match significantly improves accuracy of localization and mapping.

### 1- مقدمه

به‌خصوص مکان‌یابی و نقشه برداری هم‌زمان<sup>1</sup> از محیط صورت گرفته است. در سال 1985 یکی از اولین تلاش‌ها برای انجام مکان‌یابی و نقشه برداری به صورت هم‌زمان با استفاده از حسگر پویسگر لیزری و شفت انکودر و با توجه به عدم قطعیت‌های موجود در مسئله توسط چتیلا و همکارانش صورت گرفت

به دست آوردن مدل فیزیکی و تهیه نقشه محیط اطراف یکی از مسایل مهم و اساسی در رباتیک است و کاربردهای بسیاری از قبیل هدایت خودکار ربات، ردیابی و تشخیص اجسام و اشخاص و انجام عملیات جستجو و نجات دارد. طی ده سال گذشته پیشرفت چشمگیری در زمینه هدایت خودکار ربات و

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

Please cite this article using:

M. Zamani Alavijeh, S. Hadian Jazi, Simultaneous Localization and Mapping Using Laser Data and Unscented FastSLAM with Scan Matching, *Modares Mechanical Engineering*, Vol. 16, No. 6, pp. 217-225, 2016 (in Persian)

برای ارجاع به این مقاله از عبارت ذیل استفاده نمایید:

M. Zamani Alavijeh, S. Hadian Jazi, Simultaneous Localization and Mapping Using Laser Data and Unscented FastSLAM with Scan Matching, *Modares Mechanical Engineering*, Vol. 16, No. 6, pp. 217-225, 2016 (in Persian)

در الگوریتم‌های مکان‌یابی و نقشه‌برداری هم‌زمان موضع لحظه‌ای ربات با استفاده از مدل حرکت پیش‌بینی شده و سپس با استفاده از تفاوت ویژگی‌های ثبت شده در نقشه و ویژگی‌هایی که به‌تازگی مشاهده شده‌اند، اصلاح می‌شود. یکی از راه‌های افزایش دقت تخمین موضع ربات، مقایسه و تطبیق دو اندازه‌گیری متوالی و اصلاح موضع ربات به‌نحوی که بهترین تطابق میان دو اندازه‌گیری برقرار شود، است. این روش که انطباق پوی<sup>8</sup> نامیده می‌شود برای بهبود عملکرد الگوریتم‌های مکان‌یابی و نقشه‌برداری هم‌زمان نظیر نقشه‌برداری سریع موثر بوده است [9].

نتایج یک تحقیق نشان می‌دهد که انطباق پوی می‌تواند برای مکان‌یابی با دقت مناسب حتی در محیط 3 بعدی مورد استفاده قرار گیرد [10]. همچنین در [11] از ترکیب انطباق پوی و روش نقشه‌برداری سریع با اولویت مکان‌یابی نسبت به نقشه‌برداری برای پردازش داده لیزر استفاده شده و نتایج نشان‌دهنده بهبود مکان‌یابی بودند. در [12] یک روش هدایت سریع ربات ارایه شده که نقاط به‌دست‌آمده از هر پوی را ابتدا در سلول‌های نقشه شبکه‌بندی تصویر کرده و سپس مکان‌یابی دقیق به کمک الگوریتم انطباق پوی انجام می‌شود. در [13] روش انطباق پوی به‌دست داده شده و برای پردازش ابرنقطه 3 بعدی مورد استفاده قرار گرفته است.

با وجود تاثیر انطباق پوی در افزایش دقت مکان‌یابی، این روش تاکنون به همراه نقشه‌برداری سریع بدون ردیابی مورد آزمون قرار نگرفته است. در این مقاله برای اولین بار از ترکیب روش نقشه‌برداری سریع بدون ردیابی با انطباق پوی برای مکان‌یابی و نقشه‌برداری هم‌زمان استفاده می‌شود. برای ارزیابی تاثیر انطباق پوی بر عملکرد نقشه‌برداری سریع بدون ردیابی، این الگوریتم روی دو مجموعه داده لیزر به‌دست آمده در محیط شبیه‌سازی و همچنین در آزمایش عملی، پیاده‌سازی و نتایج آن با الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه (بدون انطباق پوی) مقایسه می‌شوند. نتایج مقایسه نشان می‌دهند که افزودن انطباق پوی به الگوریتم نقشه‌برداری سریع بدون ردیابی، دقت تخمین مسیر و دقت نقشه را به مقدار قابل ملاحظه‌ای بهبود می‌دهد. همچنین با بهبود سرعت الگوریتم انطباق داده، زمان اضافی لازم برای انجام انطباق پوی جبران شده و سرعت کلی الگوریتم نقشه‌برداری سریع بدون ردیابی نیز بهبود می‌یابد.

## 2- ترکیب انطباق پوی با نقشه‌برداری سریع بدون ردیابی

الگوریتم تکرار شونده نزدیک‌ترین نقطه<sup>9</sup>، یکی از بهترین روش‌های انطباق پوی برای هم‌تراز کردن دو مجموعه نقطه است و کاربرد زیادی در اصلاح موقعیت ربات با بهره‌گیری از مقایسه و انطباق دو پوی متوالی دارد. این الگوریتم با دو مجموعه نقطه و یک حدس اولیه برای انتقال بین آن‌ها آغاز می‌شود، سپس انتقال مکرر با استخراج نقاط متناظر بین دو مجموعه و محاسبه خطای اختلاف این نقاط تصحیح می‌شود. برای تعیین حدس اولیه گزینه‌های متنوعی وجود دارد: مقادیر تصادفی، ردیابی مکان حسگر، جستجو برای نقاط متناظر بدون اعمال انتقال اولیه و با گرفتن ورودی از کاربر. در این مقاله الگوریتم تکرار شونده نزدیک‌ترین نقطه که در [14] معرفی شده و کد آن توسط میان پیاده‌سازی شده [15, 16]، به عنوان الگوریتم و کد مبنا مورد استفاده و بازنویسی قرار گرفته است. در این مقاله الگوریتم مکان حسگر که با استفاده از مدل حرکت ردیابی می‌شود و جابجایی حسگر به‌عنوان انتقال اولیه مورد استفاده قرار می‌گیرد.

[1]. در سال 1990 اسمیت و همکارانش برای اولین بار با استفاده از مشخصه‌های محیط به نقشه‌برداری پرداختند و از فیلتر کالمن<sup>1</sup> برای حل مسئله استفاده کردند [2].

در دهه گذشته تمرکز بسیاری از محققان بر روی یافتن راه‌حل‌های مناسب برای مکان‌یابی و نقشه‌برداری هم‌زمان در زمان واقعی بوده است. در این میان محبوب‌ترین روش‌ها برای مسئله مکان‌یابی و نقشه‌برداری هم‌زمان فیلتر کالمن توسعه‌یافته<sup>2</sup> [3] و فیلتر ذره رانو-بلک‌ولایزد<sup>3</sup> بوده‌اند [4]. فیلتر کالمن توسعه‌یافته از تقریب خطی استفاده می‌کند و فرض می‌کند که تابع چگالی احتمال توزیع گوسی دارد. اما در تحقیقات مختلفی ثابت شده‌است که نقشه به‌دست آمده به این روش به‌دلیل خطای ناشی از خطی‌سازی چندان دقیق نیست و همچنین پیچیدگی محاسباتی این روش زیاد است و در زمان واقعی قابل اجرا نیست. تحقیقات بسیاری برای برطرف کردن این مشکلات صورت گرفته است.

در تلاش برای افزایش دقت مکان‌یابی و نقشه‌برداری هم‌زمان، فیلتر کالمن جدیدی تحت عنوان فیلتر کالمن ردیابی‌نشده<sup>4</sup> معرفی شد که از تقریب خطی استفاده نمی‌کند [5]. فیلتر کالمن ردیابی‌نشده از یک روش قطعی نمونه‌برداری استفاده می‌کند که متوسط و کواریانس تخمین‌ها را به‌دست می‌آورد. فیلتر کالمن ردیابی‌نشده به‌جای تابع غیرخطی سیستم، تابع چگالی احتمال را تخمین می‌زند و هرچقدر میزان غیرخطی بودن سیستم بیشتر باشد تخمین قابل اطمینان‌تری نسبت به فیلتر کالمن توسعه‌یافته ارایه می‌دهد اما پیچیدگی محاسباتی آن مشابه فیلتر کالمن توسعه‌یافته است. از طرف دیگر الگوریتم مکان‌یابی و نقشه‌برداری هم‌زمان سریع<sup>5</sup>، که از این به بعد به اختصار نقشه‌برداری سریع نامیده می‌شود، با به‌کارگیری هم‌زمان فیلتر ذره و فیلتر کالمن توسعه‌یافته حجم محاسبات را به‌طرز قابل توجهی نسبت به فیلتر کالمن توسعه‌یافته و فیلتر کالمن ردیابی‌نشده کاهش می‌دهد [6]. در نهایت الگوریتم مکان‌یابی و نقشه‌برداری هم‌زمان سریع بر مبنای فیلتر کالمن ردیابی‌نشده تحت عنوان مکان‌یابی و نقشه‌برداری هم‌زمان سریع بدون ردیابی<sup>6</sup>، که از این به بعد به اختصار نقشه‌برداری سریع بدون ردیابی نامیده می‌شود، به‌عنوان یکی از جدیدترین روش‌های مطرح شده در زمینه مکان‌یابی و نقشه‌برداری هم‌زمان با بهره‌گیری از تخمین قابل اطمینان فیلتر کالمن ردیابی‌نشده، دقت را افزایش و حجم محاسبات را کاهش می‌دهد [7]. نشان داده شده‌است که نقشه‌برداری سریع بدون ردیابی در مقایسه با روش‌های نقشه‌برداری سریع بر مبنای فیلتر کالمن توسعه‌یافته و نقشه‌برداری سریع<sup>7</sup> از نظر دقت و کیفیت نقشه‌برداری عملکرد مطلوب‌تری دارد [8]. با وجود این برتری، هنوز ظرفیت‌های بررسی‌نشده زیادی در رابطه با الگوریتم نقشه‌برداری سریع بدون ردیابی وجود دارد.

مکان‌یابی و نقشه‌برداری کاملاً به هم وابسته هستند. به این معنا که برای یافتن مکان دقیق ربات در یک محیط، نقشه صحیحی از محیط مورد نیاز است و از طرف دیگر نقشه مجموعه‌ای از مشخصه‌های محیط است که مختصات این مشخصه‌ها نسبت به ربات اندازه‌گیری شده است و در نتیجه دقت نقشه به دقت مکان‌یابی ربات بستگی دارد و افزایش دقت تخمین موضع، نقش به‌سزایی در بهبود دقت و کیفیت نقشه دارد.

<sup>1</sup> Kalman Filter (KF)

<sup>2</sup> Extended Kalman Filter (EKF)

<sup>3</sup> Rao-Blackwellized Particle Filter (RBPF)

<sup>4</sup> Unscented Kalman Filter (UKF)

<sup>5</sup> FastSLAM (FS)

<sup>6</sup> Unscented FastSLAM (UFS)

<sup>7</sup> FastSLAM 2.0

<sup>8</sup> Scan Matching

<sup>9</sup> Iterative closest point (ICP)

اولیه ( $\alpha_{init}$ ) به‌صورت زیر تعیین می‌شود:

$$A = \{\alpha_i\} = \left\{ \alpha_{init} + k \frac{\pi}{180} \right\}; k = -15, -14, \dots, 14, 15 \quad (7)$$

با توجه به این‌که زاویه  $\alpha_{init}$  یک زاویه تصادفی نیست و از رابطه (5) به‌دست آمده، برای جستجوی دقیق‌تر زاویه یک بازه 30 درجه‌ای، از 15 درجه قبل از زاویه  $\alpha_{init}$  تا 15 درجه بعد از  $\alpha_{init}$ ، کافی و مناسب است. زوایای آزمایشی در این بازه با اختلاف 1 درجه تعیین شده و مورد بررسی قرار می‌گیرند.

2- محاسبه و اعمال دوران  $\alpha_i$  به پویا جدید

اعمال دورانی معادل هر زاویه  $\alpha_i$  از مجموعه زوایای A به پویا جدید مطابق زیر صورت می‌گیرد:

$$R_{\alpha_i} = \begin{bmatrix} \cos \alpha_i & -\sin \alpha_i \\ \sin \alpha_i & \cos \alpha_i \end{bmatrix} \quad (8)$$

$$Z'_{new} = R_{\alpha_i} Z_{new} \quad (9)$$

3- تطابق پویا جدید دوران‌یافته ( $Z'_{new}$ ) با پویا مبنا و استخراج نقاط متناظر

گام بعدی جستجوی نقاط متناظر بین پویا مبنا و پویا جدید ( $Z'_{new}$ ) است که براساس معیار نزدیک‌ترین فاصله انجام می‌شود. ابتدا برای هر نقطه از پویا جدید، نزدیک‌ترین نقطه از پویا مبنا جستجو می‌شود و فاصله آن‌ها محاسبه می‌گردد. این امکان وجود دارد که برای چندین نقطه از پویا جدید، یک نقطه از پویا مبنا یافت شود که در این صورت تنها جفت نقطه‌ای که کمترین فاصله را دارد حفظ شده و بقیه حذف می‌شوند. همچنین جفت نقطه‌هایی که فاصله آن‌ها بیشتر از حد مجاز،  $d_{max}$  باشد، حذف می‌شوند و در نهایت ماتریس مطابقت<sup>2</sup> (CM) مطابق زیر تشکیل می‌گردد:

$$CM = \begin{bmatrix} m_1 & n_1 & d_1 \\ \vdots & \vdots & \vdots \\ m_K & n_K & d_K \end{bmatrix} \quad (10)$$

هر سطر این ماتریس مربوط به یک تناظر بین پویا جدید و پویا مبنا است، یعنی نقطه  $m_1$  از  $Z_{ref}$  متناظر با نقطه  $n_1$  از  $Z_{new}$  است و  $d_1$  فاصله میان این دو نقطه است.

به منظور راحتی در نوشتن ماتریس مطابقت به فرم ستونی زیر بازنویسی می‌شود:

$$CM = [M_{ref} N_{new} d]_{K \times 3} \quad (11)$$

4- محاسبه انتقال با استفاده از نقاط متناظر و اعمال انتقال به پویا

جدید ( $Z'_{new}$ )

بعد از اعمال دوران  $\alpha_i$ ، با محاسبه تفاضل میانگین پویا مبنا و پویا جدید دوران یافته، مناسب‌ترین بردار انتقال  $T_i$  تعیین می‌شود:

$$T_i = \text{mean}(Z_{ref}(M_{ref})) - \text{mean}(Z_{new}(N_{new})) \quad (12)$$

تابع  $\text{mean}$  برای محاسبه نقطه میانگین مجموعه نقاط یک پویا به‌کار می‌رود.

سپس پویا جدید انتقال یافته،  $Z''_{new}$ ، از رابطه (13) به‌دست می‌آید:

$$Z''_{new} = Z'_{new} + T_i \quad (13)$$

5- تطابق پویا جدید دوران و انتقال یافته با پویا مبنا و استخراج نقاط متناظر

در این مرحله، ماتریس تطابق جدید مطابق مرحله 3 تشکیل می‌شود.

در این الگوریتم اطلاعات دو پویا لیزر<sup>1</sup> متوالی که در دستگاه مختصات کارترزین محلی ربات بیان شده‌اند با هم مقایسه می‌شوند. نزدیک‌ترین نقاط متناظر از دو پویا که فاصله آن‌ها از یک مقدار ثابت (براساس دقت زاویه و برد حسگر) کمتر باشد، به‌صورت دو مجموعه نقاط متناظر از دو پویا ذخیره می‌شوند. سپس نقاط ذخیره شده پویا جدید، مکرراً مورد انتقال‌های خطی و چرخشی قرار می‌گیرند و بعد از هر انتقال مجموعه نقاط متناظر باز محاسبه می‌شود و این چرخه تا زمانی که خطای انطباق به مقدار قابل قبولی برسد و یا تعداد تکرارها از حدی بیشتر شود، ادامه می‌یابد. در نهایت انتقال به‌دست آمده که بهترین انطباق میان دو پویا را نتیجه می‌دهد، جابجایی از موضع مبنا به موضع جدید را نشان می‌دهد. با اعمال این انتقال به موضع مبنا، موضع جدید با دقت خوبی به‌دست می‌آید. در اکثر موارد برای کاهش تکرارهای مورد نیاز، جابجایی به‌دست آمده از مدل حرکت به‌عنوان اولین انتقال مورد استفاده قرار می‌گیرد.

الگوریتم تکرارشونده نزدیک‌ترین نقطه در این مقاله مطابق مراحل زیر انجام می‌شود:

1- تعیین ورودی‌های الگوریتم

ورودی‌های الگوریتم عبارتند از دو پویا متوالی پویا مبنا و پویا جدید و همچنین انتقال اولیه.

پویا مبنا ( $Z_{ref}$ ) و پویا جدید ( $Z_{new}$ ) که به‌ترتیب در زمان‌های  $t-1$  و  $t$  به‌دست آمده و نسبت به دستگاه مختصات کارترزین محلی ربات بیان می‌شوند:

$$Z_{ref} = \begin{bmatrix} x_1^{ref} & x_2^{ref} & \dots & x_K^{ref} \\ y_1^{ref} & y_2^{ref} & \dots & y_K^{ref} \end{bmatrix} \quad (1)$$

$$Z_{new} = \begin{bmatrix} x_1^{new} & x_2^{new} & \dots & x_L^{new} \\ y_1^{new} & y_2^{new} & \dots & y_L^{new} \end{bmatrix} \quad (2)$$

که در این‌جا هر جفت  $\begin{bmatrix} x \\ y \end{bmatrix}$  موقعیت یک نقطه از محیط در دستگاه مختصات کارترزین محلی ربات است.  $K$  تعداد نقاط مربوط به پویا مبنا و  $L$  تعداد نقاط پویا جدید است. تفاوت این دو عدد مشکلی در الگوریتم ایجاد نمی‌کند، چرا که الگوریتم تنها با جفت نقاط متناظر سروکار دارد.

بردار انتقال اولیه از تفاضل موضع مبنا و موضع جدید محاسبه می‌شود. از آن‌جا که موضع ربات در زمان  $t-1$  در مرحله قبل اصلاح شده است، به‌عنوان موضع مبنا ( $X_{ref}$ ) و موضع ربات در زمان  $t$  به‌عنوان موضع جدید ( $X_{new}$ ) خوانده می‌شوند:

$$X_{ref} = [x_{ref}, y_{ref}, \theta_{ref}]^T \quad (3)$$

$$X_{new} = [x_{new}, y_{new}, \theta_{new}]^T \quad (4)$$

که در این رابطه  $x, y$  موقعیت ربات و  $\theta$  جهت‌گیری ربات است. با توجه به دو بعدی بودن مسئله، انتقال نیز همانند موضع، شامل یک زاویه دوران  $\alpha$  و یک بردار جابجایی  $T = [x, y]$  است که از روابط زیر محاسبه می‌گردند:

$$\alpha_{init} = \theta_{new} - \theta_{ref} \quad (5)$$

$$T_{init} = [x_{init}, y_{init}] = [x_{new} - x_{ref}, y_{new} - y_{ref}] \quad (6)$$

در این الگوریتم تنها از دوران اولیه  $\alpha_{init}$  استفاده شده و  $T_{init}$  در مرحله بعد با استفاده از همین دوران محاسبه می‌شود، لذا تنها  $\alpha_{init}$  به‌عنوان انتقال اولیه به الگوریتم وارد می‌شود. مجموعه زوایای دوران آزمایشی حول زاویه دوران

<sup>2</sup> Correspondence Matrix

<sup>1</sup> Laser Scan

یکی از متداول‌ترین روش‌های ارتباط داده، روش بیشترین احتمال است. در این روش ابتدا ویژگی‌های نقشه که در مختصات جهانی ذخیره شده‌اند، در دستگاه کارترین محلی ربات در زمان  $t$  بیان می‌شوند تا قابل مقایسه با مشاهده‌های جدید باشند. سپس برای هر ویژگی از پویا  $Z_{new}$  شبیه‌ترین ویژگی نقشه را جستجو کرده و چنانچه اختلاف این دو ویژگی کمتر از آستانه پذیرش تناظر و مقدار تابع احتمال قابل قبول باشد، متناظر شناخته می‌شوند و اگر مقدار تابع احتمال بالاتر از آستانه رد تناظر باشد، به‌عنوان یک ویژگی جدید ثبت می‌شود.

برای پیاده‌سازی این روش معمولاً از کد نوشته شده توسط بیلی [17] استفاده می‌شود که برای یافتن ویژگی متناظر احتمالی با هر مشاهده، فاصله تمام ویژگی‌های نقشه با آن مشاهده را محاسبه کرده و نزدیک‌ترین فاصله را ذخیره می‌کند و سپس تابع احتمال را محاسبه کرده و براساس مقدار آن ویژگی را به عنوان شناخته‌شده، جدید و یا نامعلوم طبقه‌بندی می‌کند.

مشکل این روش این است که محاسبه فاصله تعداد زیادی زوج ویژگی مشاهده و نقشه، بسیار زمان‌بر است. به‌طور مثال فرض کنید که تعداد 300 ویژگی در نقشه ذخیره شده باشند و در پویا جدید نیز 100 مشاهده ثبت شده باشد، چنانچه قرار باشد با این روش نزدیک‌ترین ویژگی متناظر با هر مشاهده پیدا شود، باید  $100 \times 300$  یعنی 30000 محاسبه فاصله انجام شود. از طرف دیگر به‌دلیل همین تعداد زیاد محاسبات، زمان قابل ملاحظه‌ای صرف مرحله ارتباط داده‌ها می‌شود. در واقع مرحله ارتباط داده‌ها، با اختلاف زیادی، زمان‌برترین مرحله از الگوریتم‌های نقشه برداری سریع و نقشه برداری سریع بدون ردیابی است.

در این تحقیق تلاش شده تا با انجام اصلاحاتی فرآیند ارتباط داده‌ها با دقتی مشابه ولی در زمانی کمتر و بدون تکرار بررسی شود. برای این منظور، برای بررسی نزدیک‌ترین فاصله، ابتدا از دستور `dsearchn` در نرم‌افزار متلب<sup>3</sup> استفاده شده است که با مقایسه دو مجموعه نقاط دو بعدی، برای هر نقطه از مجموعه دوم، نزدیک‌ترین نقطه از مجموعه اول را یافته و فاصله آنها را نیز محاسبه می‌کند. نقاط مجموعه دوم غیر تکراری هستند ولی ممکن است برای چندین نقطه از این مجموعه، یک نقطه مشترک از مجموعه اول به‌عنوان نزدیک‌ترین نقطه یافته شود. در این صورت، جفت نقطه‌ی دارای کمترین فاصله حفظ شده، و بقیه حذف می‌گردند. سپس تابع احتمال برای تناظرهای باقی‌مانده محاسبه می‌شود و سایر مراحل مطابق پیاده‌سازی آقای بیلی انجام می‌شوند. متوسط زمان انجام هر مرحله ارتباط داده با این روش (در شرایط مشابه) 0.02 ثانیه است که در حدود 50 بار سریع‌تر از روش اول است.

الگوریتم ارتباط داده مورد استفاده در این مقاله دارای مراحل زیر است:

- 1- تعیین ورودی‌ها: ویژگی‌های ذخیره شده در نقشه،  $Z_{map}$ ، و پویا جدید،  $Z_{new}$
- 2- جستجوی نزدیک‌ترین نقطه از ویژگی‌های نقشه برای هر نقطه از پویا جدید با استفاده از دستور `dsearchn`:

$$[corr, D] = dsearchn(Z_{map}, tri, Z_{new}) \quad (18)$$

در این رابطه،  $[corr, D]$  ماتریسی مشابه ماتریس CM در رابطه (10) است و هر سطر  $corr$  مربوط به اندیس یک جفت نقطه متناظر بین پویا جدید و نقشه و همان سطر از  $D$  فاصله بین دو نقطه است.

همچنین `tri` با استفاده از دستور `delaunay` در نرم‌افزار متلب محاسبه می‌شود:

6- محاسبه و ذخیره خطای انتقال برای نقاط متناظر

برای ارزیابی دقت انطباق دو پویا در هر تکرار و برای هر  $\alpha_i$  خطای تناظر که معادل مجموع فاصله‌های جفت نقطه‌های متناظر تقسیم بر تعداد آن-هاست محاسبه می‌شود:

$$error_i = \frac{1}{K} \sum_{j=1}^K d_j \quad (14)$$

7- تکرار الگوریتم برای کمینه کردن خطا

مراحل 2 تا 6 برای تمام مجموعه زوایای دوران آزمایشی  $\alpha_i$  انجام شده و دوران و انتقالی که کمترین خطا را داشته باشد، انتخاب می‌شود:

$$T_{icp} = T_j; \alpha_{icp} = \alpha_j \quad (15)$$

زاندیس کمترین خطای به‌دست آمده است و در نتیجه  $\alpha_j$  و  $T_j$  مناسب‌ترین دوران و انتقال هستند.

8- محاسبه موضع اصلاح شده ربات

موضع اصلاح شده از اعمال دوران و انتقال بهینه به موضع جدید ربات به‌دست می‌آید:

$$R_{\theta_{new}} = \begin{bmatrix} \cos\theta_{new} & -\sin\theta_{new} \\ \sin\theta_{new} & \cos\theta_{new} \end{bmatrix} \quad (16)$$

$$X'_{new} = \begin{bmatrix} [x_{new}, y_{new}]^T + R_{\theta_{new}} T_{icp} \\ \theta_{new} + \alpha_{icp} \end{bmatrix} \quad (17)$$

شبه کد الگوریتم انطباق پویا بصورت زیر است:

ICP ( $Z_{ref}, Z_{new}, \alpha_{init}$ )

Generate sample  $A$

$$A = \{\alpha_i\} = \left\{ \alpha_{init} + k \frac{\pi}{180} \right\}; k = -15, -14, \dots, 14, 15$$

for  $\alpha_i \in A$  do

$$R_{\alpha_i} = \begin{bmatrix} \cos\alpha_i & -\sin\alpha_i \\ \sin\alpha_i & \cos\alpha_i \end{bmatrix}; Z'_{new} = R_{\alpha_i} Z_{new}$$

Calculate Correspondence Matrix

$$T_i = \text{mean}(Z_{ref}(M_{ref})) - \text{mean}(Z_{new}(N_{new}))$$

$$Z''_{new} = Z'_{new} + T_i$$

Recalculate Correspondence Matrix

$$error_i = \frac{1}{K} \sum_{j=1}^K d_j$$

endfor

$$j = \arg \min\{error_i, i\}$$

$$T_{icp} = T_j; \alpha_{icp} = \alpha_j$$

$$R_{\theta_{new}} = \begin{bmatrix} \cos\theta_{new} & -\sin\theta_{new} \\ \sin\theta_{new} & \cos\theta_{new} \end{bmatrix}$$

$$X'_{new} = \begin{bmatrix} [x_{new}, y_{new}]^T + R_{\theta_{new}} T_{icp} \\ \theta_{new} + \alpha_{icp} \end{bmatrix}$$

همان‌طور که در شبه کد نیز مشخص است پیچیدگی محاسباتی الگوریتم به تعداد زوایای مورد بررسی در مجموعه زوایای دوران آزمایشی بستگی دارد. لذا اگر تعداد این زوایا  $K$  باشد، پیچیدگی محاسباتی  $O(K)$  است. بنابراین هرچه بازه زوایای مورد جستجو بزرگتر و دقت بیشتر باشد، الگوریتم پیچیده‌تر و زمان‌برتر خواهد بود.

### 3- ارتباط داده‌ها<sup>1</sup> به روش بیشترین احتمال<sup>2</sup>

ارتباط داده‌ها فرآیند بررسی ارتباط میان ویژگی‌هایی که به‌تازگی مشاهده شده‌اند با ویژگی‌های ذخیره شده در نقشه است. تشخیص درست این که کدام ویژگی‌ها جدید هستند و کدام ویژگی‌ها قبلاً مشاهده شده‌اند و این ویژگی‌های مشاهده شده با کدام ویژگی‌های نقشه متناظر هستند، برای به‌روزرسانی موضع ربات و تهیه نقشه محیط ضروری است.

<sup>1</sup> Data Association

<sup>2</sup> Maximum Likelihood (ML)

<sup>3</sup> MATLAB

برای بررسی تاثیر انطباق پویایی روی الگوریتم نقشه برداری سریع بدون ردیابی، باید ابتدا الگوریتم نقشه برداری سریع بدون ردیابی اولیه مورد استفاده قرار گیرد. برای این منظور از کد نوشته شده توسط کیم با همان الگوریتم ارتباط داده اولیه استفاده شده و تنها پارامترهای آن تنظیم شده اند و قابلیت تولید نقشه شبکه بندی به آن اضافه شده است. "شکل 2" نقشه ویژگی مینا به همراه مسیر واقعی و مسیر تخمین زده شده ربات و "شکل 3" نقشه شبکه بندی محیط را نشان می دهد.

به دلیل خطای تخمین مسیر، مرزها دارای خطا بوده و زاویه دار هستند. در برخی نواحی، خطای موضع به قدری زیاد بوده که موجب تکرار یک وجه از محیط در نقشه ویژگی مینا و شبکه بندی شده است.

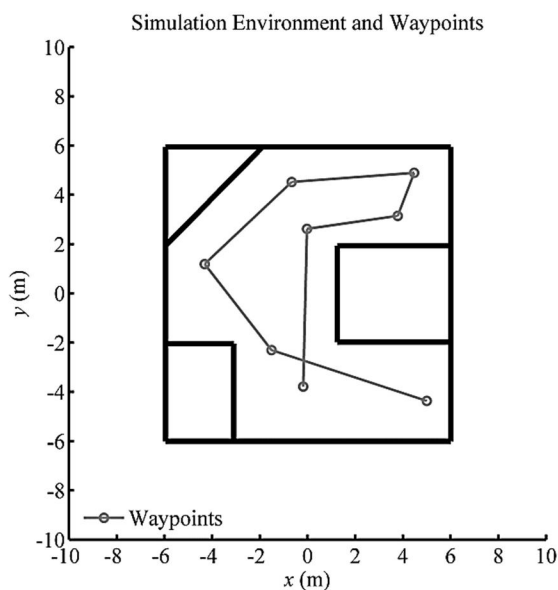


Fig. 1 Indoor environment and robot path simulated in MATLAB  
شکل 1 محیط و مسیر هدایت ربات شبیه سازی شده در نرم افزار متلب

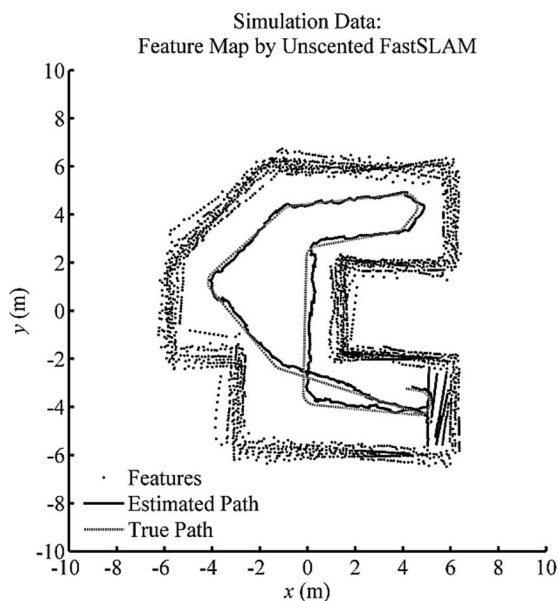


Fig. 2 Simulation Data: Feature map, true path and estimated path by Unscented FastSLAM

شکل 2 داده شبیه سازی: نقشه ویژگی مینا، مسیر واقعی و مسیر تخمین زده شده توسط الگوریتم نقشه برداری سریع بدون ردیابی اولیه

$$tri = delaunayn(Z_{map}) \quad (19)$$

- 3- حذف نقاط تکراری از ویژگی های نقشه و حفظ نزدیک ترین نقطه
- 4- محاسبه احتمال ارتباط جفت نقطه و مقایسه با آستانه پذیرش و آستانه رد ویژگی
- 5- ذخیره نقطه ی پویا جدید به عنوان ویژگی جدید، ویژگی قدیمی و یا هیچ کدام

#### 4- پیاده سازی و ارزیابی عملکرد الگوریتم نقشه برداری سریع بدون ردیابی با انطباق پویایی

در این مقاله مکان یابی و نقشه برداری هم زمان از طریق اعمال الگوریتم های نقشه برداری سریع بدون ردیابی اولیه و نقشه برداری سریع بدون ردیابی با انطباق پویایی روی داده های به دست آمده از شبیه سازی و نیز داده های لیزر واقعی انجام شده است. کد الگوریتم نقشه برداری سریع بدون ردیابی در نرم افزار متلب توسط کیم نوشته شده [18] و به عنوان کد مینا در این مقاله مورد استفاده قرار گرفته و تغییرات لازم متناسب با شرایط مسئله روی آن اعمال شده است. برخی از مهم ترین این تغییرات عبارتند از:

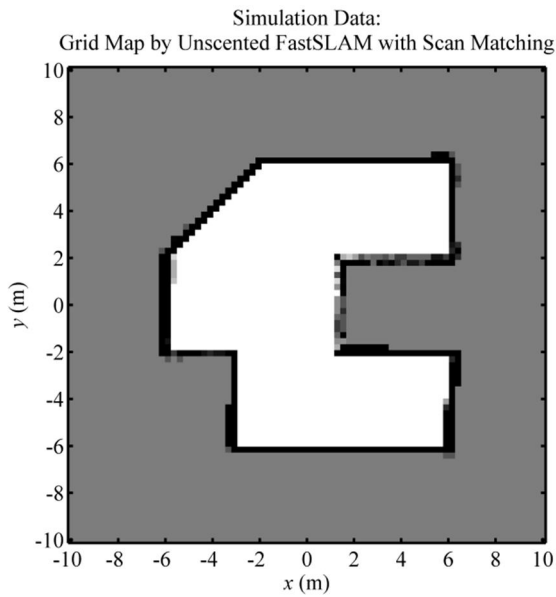
- افزودن انطباق پویایی
- بهبود سرعت الگوریتم ارتباط داده ها
- تولید نقشه شبکه بندی محیط
- بهینه سازی مقدار متغیرهای الگوریتم از جمله آستانه پذیرش و رد ویژگی

خروجی های این الگوریتم ها عبارتند از: نقشه شبکه بندی محیط، نقشه ویژگی مینا به همراه مسیر واقعی طی شده توسط ربات و مسیر تخمین زده شده توسط الگوریتم ها، و نیز میانگین خطای موقعیت و جهت گیری تخمین زده شده نسبت به موقعیت و جهت گیری واقعی ربات که در یک جدول بیان شده اند. همچنین برای بررسی تاثیر بهبود سرعت الگوریتم ارتباط داده ها و افزودن انطباق پویایی، زمان اجرای کل الگوریتم و زمان اجرای مراحل ارتباط داده ها و انطباق پویایی به صورت جدول ارائه شده است.

#### 4-1-1- پیاده سازی و ارزیابی الگوریتم نقشه برداری سریع بدون ردیابی با انطباق پویایی روی داده لیزر به دست آمده از شبیه سازی

در این مرحله، محیط پیرامون ربات مطابق "شکل 1" شبیه سازی شده و نقاطی به عنوان راهنمای مسیر در آن تعبیه شده اند که ربات با دنبال کردن این نقاط در محیط حرکت می کند و اطلاعات سرعت هر لحظه برای استفاده در مدل حرکت ثبت می شود. خطای کنترل سرعت 0.1 متر بر ثانیه و خطای کنترل جهت ربات 1 درجه در ثانیه فرض شده اند. در این مرحله الگوریتم روی داده لیزر اعمال می شود که توسط حسگر پویسگر لیزری شبیه سازی شده به دست می آید. برد این حسگر 6 متر، میدان دید آن 180 درجه و رزولوشن زاویه ای آن برابر 1 درجه شبیه سازی شده است. همچنین نویز اندازه گیری آن به ترتیب 0.03 متر و 1 درجه در نظر گرفته شده اند. از نظر زمان بندی، ارسال فرمان کنترل سرعت به ربات و پویا محیط هم زمان با هم و هر 0.1 ثانیه صورت می گیرند. پس از ذخیره اطلاعات حرکت ربات و اطلاعات به دست آمده از حسگر شبیه سازی شده، در مرحله بعد این اطلاعات به صورت ورودی به الگوریتم داده شده و مورد پردازش برون خط قرار می گیرند.

#### 4-1-1- عملکرد الگوریتم نقشه برداری سریع بدون ردیابی اولیه

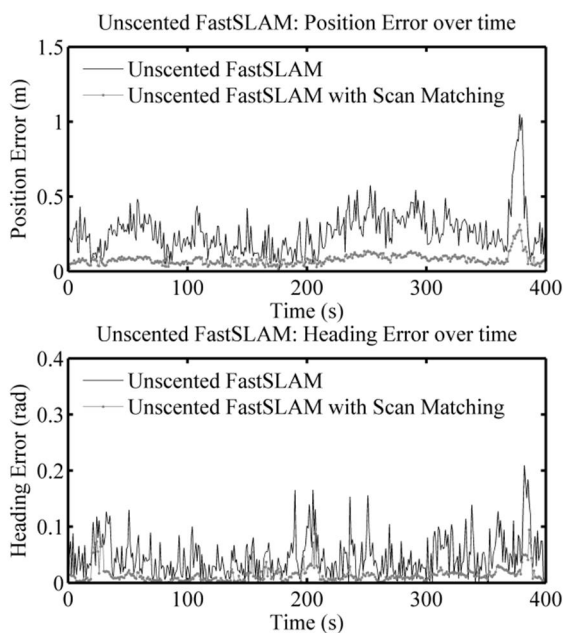


**Fig. 5** Simulation Data: Grid map by Unscented FastSLAM with scan matching

شکل 5 داده شبیه‌سازی: نقشه شبکه‌بندی توسط الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی

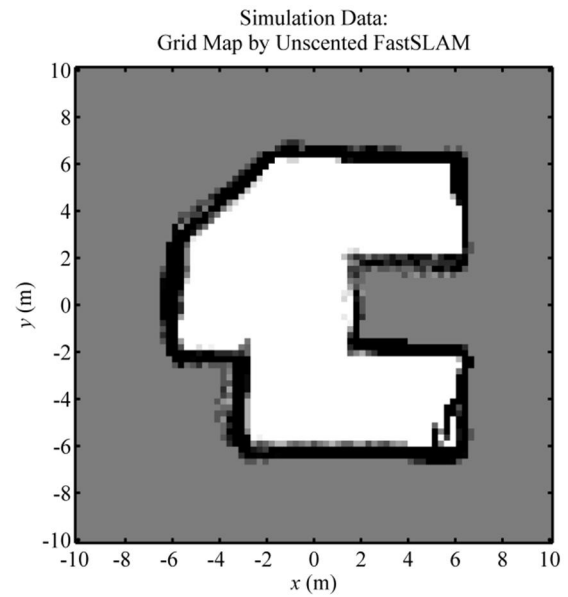
"شکل 6" خطای موقعیت و جهت‌گیری در طول مسیر را برای هر دو الگوریتم نشان می‌دهد. مقدار خطای تخمین توسط نقشه‌برداری سریع بدون ردیابی با انطباق پویایی در تمام گام‌ها کمتر از نقشه‌برداری سریع بدون ردیابی اولیه است.

همچنین مقادیر میانگین خطای موقعیت و جهت‌گیری برای هر دو الگوریتم در جدول 1 آورده شده‌اند. متوسط خطای موقعیت 0.26 متر و متوسط خطای جهت‌گیری 0.046 رادیان است. مقایسه میانگین خطای این الگوریتم با روش قبل نشان می‌دهد که انطباق پویایی، تخمین موقعیت و



**Fig. 6** Position and heading estimation error over time

شکل 6 خطای تخمین موقعیت و جهت‌گیری در طول زمان

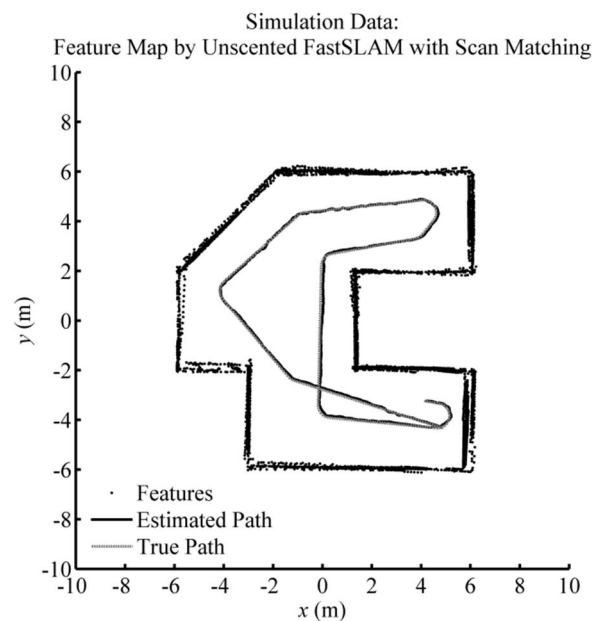


**Fig. 3** Simulation Data: Grid map by Unscented FastSLAM

شکل 3 داده شبیه‌سازی: نقشه شبکه‌بندی توسط الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه

**4-1-2- عملکرد الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی**

در این مرحله الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی روی داده لیزر حاصل از محیط شبیه‌سازی اعمال شد. نقشه‌های ویژگی‌منا و شبکه‌بندی به‌دست آمده به ترتیب در "شکل 4" و "شکل 5" آورده شده‌اند. همان‌طور که در "شکل‌های 4 و 5" مشخص است، دقت تخمین مسیر بیشتر شده و مسیر تخمین‌زده شده در الگوریتم دوم بسیار به مسیر واقعی نزدیک‌تر است. به‌همین دلیل نقشه‌های ویژگی‌منا و شبکه‌بندی دقیق‌تر با مرزهای باریک‌تر به‌دست آمده است.



**Fig. 4** Simulation Data: Feature map, true path and estimated path by Unscented FastSLAM with scan matching

شکل 4 داده شبیه‌سازی: نقشه ویژگی‌منا، مسیر واقعی و مسیر تخمین‌زده شده توسط الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی

بین زمان داده کنترل حرکت و داده لیزر وجود ندارد. بنابراین در هر گام، با مقایسه زمان متناظر با آن موضع و زمان متناظر با پوشش‌های لیزر، نزدیک‌ترین پوشش از نظر زمانی انتخاب و پردازش می‌شود. به دلیل آن‌که بازه‌های زمانی بسیار کوچک هستند، اختلاف بین زمان حرکت و زمان پوشش در حدود 0.1 ثانیه و قابل چشم‌پوشی است.

در این مرحله نتایج نقشه‌برداری سریع بدون ردیابی با انطباق پوی با نقشه‌برداری سریع بدون ردیابی اولیه از نظر دقت نقشه و زمان اجرا مقایسه می‌شود. متغیرهای الگوریتم متناسب با شرایط آزمایش تنظیم شده‌اند.

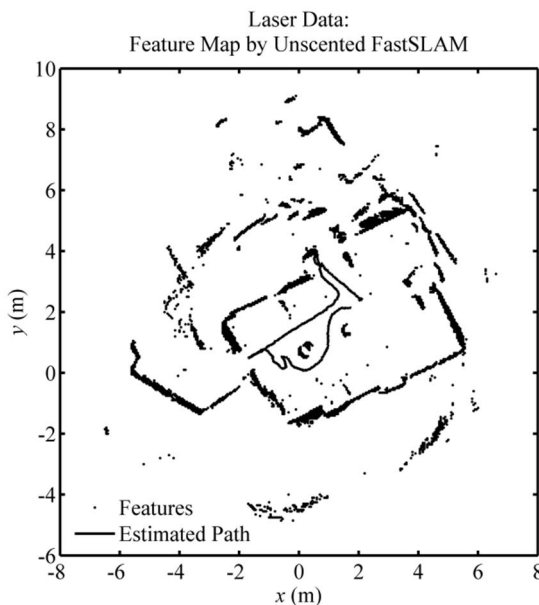
**4-2-1- پیاده‌سازی الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه روی داده لیزر به‌دست آمده از آزمایش عملی**

مشابه قسمت قبل، ابتدا عملکرد نقشه‌برداری سریع بدون ردیابی اولیه روی داده لیزر واقعی بررسی می‌شود. نقشه‌های ویژگی‌منا و شبکه‌بندی به‌دست آمده به ترتیب در "شکل 7" و "شکل 8" نشان داده شده‌اند. به دلیل آنکه مسیر واقعی ربات در اختیار نبوده، امکان مقایسه مسیر تخمین زده شده با مسیر واقعی وجود ندارد؛ بنابراین ارزیابی عملکرد الگوریتم از طریق کیفیت نقشه صورت می‌گیرد.

همان‌طور که در "شکل‌های 7 و 8" مشخص است، مرزها و موانع در حد قابل قبولی در نقشه‌های ویژگی‌منا و شبکه‌بندی نشان داده شده‌اند. اما به دلیل خطا در تخمین مسیر، مرزها چندان باریک نیستند و حتی گوشه بالای اتاق دوبار تکرار شده است. بیرون‌زدگی‌های موجود در نقشه، در اثر درهای باز ایجاد شده‌اند ولی چندان واضح نشان داده نشده‌اند.

**4-2-2- پیاده‌سازی الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پوی روی داده لیزر به‌دست آمده از آزمایش عملی**

پس از اجرای الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پوی، نقشه‌های ویژگی‌منا و شبکه‌بندی به‌دست آمدند که در "شکل 9" و "شکل 10" نشان داده شده‌اند. مقایسه این نقشه‌ها با نتایج قسمت قبل نشان



**Fig. 7** Experimental Data: Feature map and estimated path by Unscented FastSLAM

شکل 7 داده آزمایشگاهی: نقشه ویژگی‌منا و مسیر تخمین زده شده توسط الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه

**جدول 1** متوسط خطای تخمین موقعیت و جهت‌گیری در الگوریتم‌های نقشه‌برداری سریع بدون ردیابی اولیه و نقشه‌برداری سریع بدون ردیابی با انطباق پوی

**Table 1** Average position and heading estimation

نقشه‌برداری سریع بدون ردیابی بدون انطباق	نقشه‌برداری سریع بدون ردیابی با انطباق پوی	متوسط خطای موقعیت
0.078	0.260	خطای موقعیت نرمالیزه
0.020	0.066	متوسط خطای جهت‌گیری
0.014	0.046	خطای جهت‌گیری نرمالیزه
0.009	0.029	

جهت‌گیری را به ترتیب 70% و 68% بهبود داده و در نتیجه نقشه دقیق‌تری ارائه می‌دهد. لازم به ذکر است خطای نرمالیزه از تقسیم مقدار خطا بر مقدار میانگین به‌دست می‌آید.

در جدول 2 زمان اجرای الگوریتم‌های نقشه‌برداری سریع بدون ردیابی برای داده شبیه‌سازی آورده شده است. زمان اجرای الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه برای 100 پوشش 236 ثانیه است که 219 ثانیه (93%) از آن مربوط به مرحله ارتباط داده‌ها است. همین درصد قابل توجه انگیزه‌بخش تلاش برای بهبود سرعت ارتباط داده‌ها در این مقاله بوده است. مدت زمان ارتباط داده‌ها در الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه شده در این مقاله تنها 4 ثانیه است که 54 بار سریع‌تر است. در نتیجه با وجود اضافه کردن انطباق پوی، نقشه‌برداری سریع بدون ردیابی ارائه شده در این مقاله، 3 بار سریع‌تر از نقشه‌برداری سریع بدون ردیابی اولیه است.

**4-2-2- پیاده‌سازی الگوریتم نقشه‌برداری سریع بدون ردیابی روی داده لیزر به‌دست آمده از آزمایش عملی**

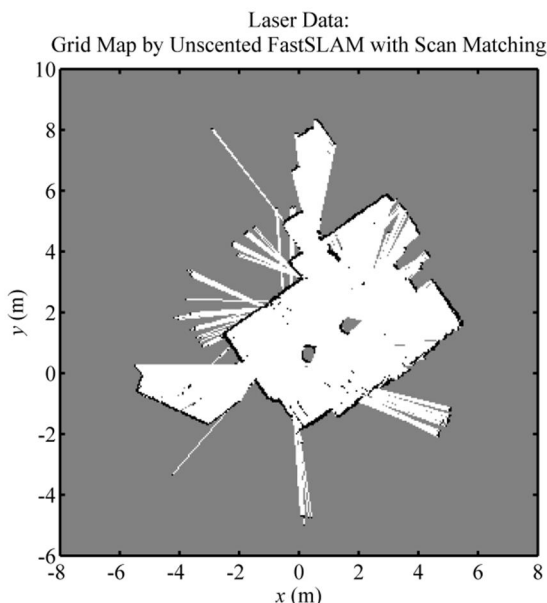
برای بررسی عملکرد الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پوی در شرایط واقعی، آزمایش عملی نیز انجام شد. در این آزمایش ربات مجهز به پوششگر لیزری و انکودر در یک محیط بسته به‌صورت دستی حرکت داده شد و مجموعه اطلاعات شامل دور چرخ‌های چپ و راست به‌دست آمده از حسگر انکودر و داده لیزر به‌دست آمده از پوششگر لیزری در محیط دانشگاه صنعتی اصفهان و به کمک رباتی که در همین دانشگاه ساخته شده به‌دست آمده است.

پوششگر لیزری مورد استفاده در این آزمایش دارای برد 6 متر، میدان دید 180 درجه و رزولوشن 0.3 درجه است. نویز اندازه‌گیری این مدل پوششگر 0.03 متر و 1 درجه است. مدت انجام آزمایش 120 ثانیه بوده و به‌طور تقریبی در هر ثانیه 10 پوشش لیزر و 29 دور چرخ چپ و راست ضبط شده است. از تعداد دور چرخ چپ و راست برای محاسبه مقدار و جهت سرعت ربات استفاده می‌شود. ربات به صورت دستی حرکت داده شده و هماهنگی خاصی

**جدول 2** داده شبیه‌سازی: زمان اجرای الگوریتم نقشه‌برداری سریع بدون ردیابی برای صد پوشش

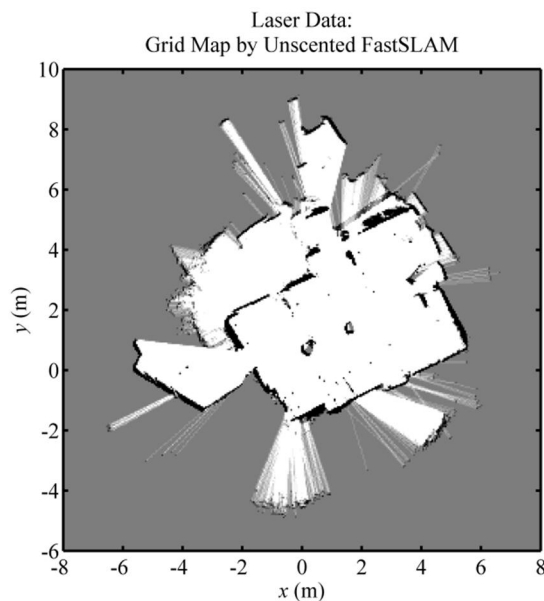
**Table 2** Simulation data: Execution time of Unscented FastSLAM for 100 scans

نقشه‌برداری سریع بدون ردیابی بدون انطباق پوی	نقشه‌برداری سریع بدون ردیابی با انطباق پوی	زمان کل الگوریتم [ثانیه]
75	236	زمان تطابق پوشش [ثانیه]
58	0	زمان ارتباط داده [ثانیه]
4	219	



**Fig. 10** Experimental Data: Grid map by Unscented FastSLAM with scan matching

شکل 10 داده آزمایشگاهی: نقشه شبکه‌بندی توسط الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی



**Fig. 8** Experimental Data: Grid map by Unscented FastSLAM

شکل 8 داده آزمایشگاهی: نقشه شبکه‌بندی توسط الگوریتم نقشه‌برداری سریع بدون ردیابی اولیه

جدول 3 داده آزمایشگاهی: زمان اجرای الگوریتم نقشه‌برداری سریع بدون ردیابی برای صد پویایی

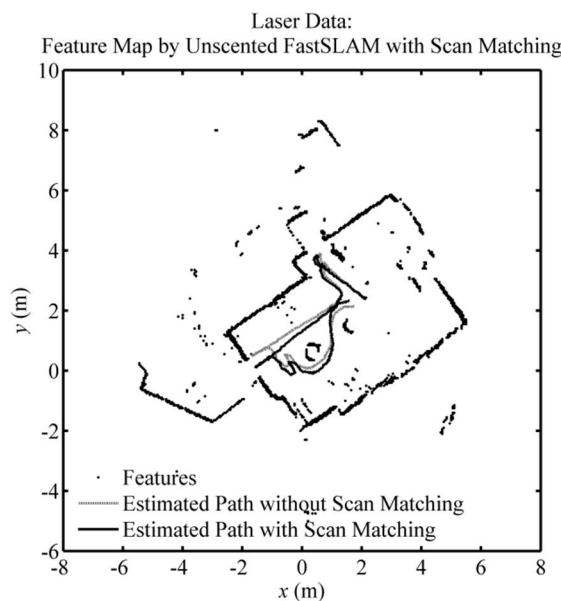
**Table 3** Experimental data: Execution time of Unscented FastSLAM for 100 scans

نقشه‌برداری سریع بدون ردیابی با انطباق پویایی	نقشه‌برداری سریع بدون ردیابی بدون انطباق پویایی	
93	385	زمان کل الگوریتم [ثانیه]
68	0	زمان تطابق پویایی [ثانیه]
6	343	زمان ارتباط داده [ثانیه]

نقشه‌برداری سریع بدون ردیابی ارائه شده در این مقاله تنها 6 ثانیه و 57 بار سریع‌تر است. بنابراین با وجود اضافه کردن انطباق پویایی، نقشه‌برداری سریع بدون ردیابی ارائه شده در این مقاله، 4 بار سریع‌تر از نقشه‌برداری سریع بدون ردیابی اولیه است.

### 5- نتیجه‌گیری و پیشنهادات

در این مقاله برای اولین بار از ترکیب الگوریتم نقشه‌برداری سریع بدون ردیابی با انطباق پویایی استفاده شد و سرعت مرحله ارتباط داده به میزان قابل ملاحظه‌ای بهبود یافت. سپس نقشه‌برداری سریع بدون ردیابی با انطباق پویایی روی داده لیزر به دست آمده از شبیه‌سازی و داده لیزر واقعی پیاده‌سازی شد و نتایج آن با نتایج نقشه‌برداری سریع بدون ردیابی اولیه مقایسه شدند. نتایج مقایسه نشان داد که با افزودن یک مرحله انطباق پویایی به الگوریتم، دقت تخمین مسیر و نقشه به مقدار قابل ملاحظه‌ای بهبود می‌یابند. مقدار کاهش خطای تخمین موضع در شبیه‌سازی 68% است و در آزمایش عملی به دلیل در دست نداشتن مسیر واقعی قابل اندازه‌گیری نیست. همچنین مشکل زمان‌بر بودن مرحله ارتباط داده‌ها، با طراحی بهتر حلقه و استفاده از دستورهای مناسب‌تر برطرف شد. ارتباط داده‌ها در حدود 50 بار سریع‌تر از قبل شده و زمان کل الگوریتم جدید در شبیه‌سازی 68% و در



**Fig. 9** Experimental Data: Feature map and estimated path by both Unscented FastSLAM algorithms

شکل 9 داده آزمایشگاهی: نقشه ویژگی‌مبنا، مسیر تخمین زده شده توسط هر دو الگوریتم نقشه‌برداری سریع بدون ردیابی

می‌دهد که انطباق پویایی باعث شده مرزها با دقت بیشتر و باریک‌تر از قبل ترسیم شوند. موانع نیز کامل‌تر از قبل در نقشه ظاهر شده‌اند. در "شکل 9" علاوه بر نقشه ویژگی مبنا و مسیر تخمین‌زده شده توسط هر دو الگوریتم نشان داده شده است.

همچنین در جدول 3 زمان پیاده‌سازی الگوریتم‌های نقشه‌برداری سریع بدون ردیابی روی داده لیزر واقعی و برای 100 پویایی آورده شده است. در نقشه‌برداری سریع بدون ردیابی اولیه، 89% از زمان کل الگوریتم مربوط به مرحله ارتباط داده‌ها است، در حالی که این مدت زمان در الگوریتم



آزمایش واقعی 75% کمتر از الگوریتم اولیه است.

## 6- تقدیر و تشکر

در این‌جا لازم است از آزمایشگاه رباتیک و میکاترونیک پیشرفته دانشکده مهندسی مکانیک دانشگاه صنعتی اصفهان و جناب آقای مهندس مرتضی بدلی به سبب همکاری بی‌دریغ در تهیه امکانات و محیط مورد نیاز تقدیر و تشکر شود.

## 7- مراجع

- FastSLAM2. 0, and UKF-based FastSLAM algorithms, *Proceeding of 16th International Conference on Intelligent Engineering Systems (INES)*, Lisbon, Portugal: IEEE, pp. 37-43, 2012.
- [9] D. Hahnel, W. Burgard, D. Fox, S. Thrun, An efficient FastSLAM algorithm for generating maps of large-scale cyclic environments from raw laser range measurements, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Las Vegas, USA: IEEE, pp. 206-211, 2003.
- [10] G. Michalíček, D. Klimentjew, J. Zhang, A 3D simultaneous localization and mapping exploration system, *Proceeding of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, Phuket, Thailand: IEEE, pp. 1059-1065, 2011.
- [11] J. Röwekämper, C. Sprunk, G. D. Tipaldi, C. Stachniss, P. Pfaff, W. Burgard, On the position accuracy of mobile robot localization based on particle filters combined with scan matching, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Algarve, Portugal: IEEE, pp. 3158-3164, 2012.
- [12] T. Furukawa, L. Dantanarayana, J. Ziegler, R. Ranasinghe, G. Dissanayake, Fast global scan matching for high-speed vehicle navigation, *Proceeding of IEEE International Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI)*, San Diego, USA: IEEE, pp. 37-42, 2015.
- [13] Z. Zheng, Y. Li, LIDAR data registration for unmanned ground vehicle based on improved ICP algorithm, *Proceeding of 7th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, Hangzhou, China: IEEE, pp. 554-558, 2015.
- [14] P. J. Besl, N. D. McKay, Method for registration of 3-D shapes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 2, pp. 239-256, 1992.
- [15] A. S. Mian, M. Bennamoun, R. Owens, Three-dimensional model-based object recognition and segmentation in cluttered scenes, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 10, pp. 1584-1601, 2006.
- [16] A. S. Mian. *icp2*, Accessed on 16 December 2015; <http://staffhome.ecm.uwa.edu.au/~00053650/code/icp2.m>.
- [17] T. Bailey. *SLAM Package of Tim Bailey*, Accessed on 16 December 2015; <https://openslam.org/bailey-slam.html>.
- [18] C. Kim, *Unscented FastSLAM*, Accessed on 16 December 2015; <https://openslam.org/ufastslam.html>.
- [1] R. Chatila, J. P. Laumond, Position referencing and consistent world modeling for mobile robots, *Proceeding of IEEE International Conference on Robotics and Automation*, Silver Spring, USA: IEEE, pp. 138-145, 1985.
- [2] C. G. Harris, J. Pike, 3D positional integration from image sequences, *Image and Vision Computing*, Vol. 6, No. 2, pp. 87-90, 1988.
- [3] R. Smith, M. Self, P. Cheeseman, *Estimating Uncertain Spatial Relationships in Robotics*, pp. 167-193, New York: Springer, 1990.
- [4] A. Doucet, N. De Freitas, K. Murphy, S. Russell, Rao-Blackwellised particle filtering for dynamic Bayesian networks, *Proceeding of the Sixteenth Conference on Uncertainty in Artificial Intelligence*, San Francisco, USA: Morgan Kaufmann Publishers, pp. 176-183, 2000.
- [5] R. Martinez-Cantin, J. Castellanos, Unscented SLAM for large-scale outdoor environments, *Proceeding of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Edmonton, Canada: IEEE, pp. 3427-3432, 2005.
- [6] M. Montemerlo, S. Thrun, D. Koller, B. Wegbreit, FastSLAM: A factored solution to the simultaneous localization and mapping problem, *Proceeding of the AAAI National Conference on Artificial Intelligence*, Edmonton, Canada: AAAI, pp. 593-598, 2002.
- [7] C. Kim, R. Sakthivel, W. K. Chung, Unscented FastSLAM: a robust and efficient solution to the SLAM problem, *IEEE Transactions on Robotics*, Vol. 24, No. 4, pp. 808-820, 2008.
- [8] Z. Kurt-Yavuz, S. Yavuz, A comparison of EKF, UKF,