Journal of Applied And Basic Machine Intelligence Research (ABMIR)

پژوهش‌های نظری و کاربردی هوش ماشینی

سال چهارم، شماره اول، بهار و تابستان ۱۴۰۵، صفحات ۲۷ – ۱۵

CAR-FMP الگوریتم بهبودیافته برنامه‌ریزی حرکت مبتنی بر نیرو برای خودروهای خودران در محیط‌های شهری

شکیبا برنجکوب[۱]، سمانه حسینی[۲*]، علی تمیزی فر[۳]

[۱] دانشجوی کارشناسی دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران

[۲] استادیار دانشکده مهندسی برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران

[۳] دانشجوی کارشناسی ارشد دانشکده برق و کامپیوتر، دانشگاه صنعتی اصفهان، اصفهان، ایران

مقاله پژوهشی

تاریخ دریافت:
۱۴۰۴/۰۷/۱۹

تاریخ پذیرش:
۱۴۰۴/۰۹/۰۸

کلیدواژه‌ها:
خودروهای خودران، برنامه‌ریزی حرکت، الگوریتم FMP، ناوبری شهری، کنترل توزیع‌شده، سبقت ایمن، شبیه‌ساز CARLA، تصمیم‌گیری بلادرنگ

نویسنده‌مسئول:
samaneh.hoseini@iut.ac.ir

چکیده

در سال‌های اخیر، توسعه سامانه‌های هوشمند حمل‌ونقل توجه بسیاری از پژوهشگران را به خود جلب کرده است. یکی از مهم‌ترین چالش‌های این حوزه، ناوبری خودروهای خودران در محیط‌های شهری است که به‌دلیل نیاز به تصمیم‌گیری بلادرنگ در شرایط ترافیکی پویا، تعامل با سایر وسایل نقلیه و واکنش مناسب به موانع غیرقابل پیش‌بینی، پیچیدگی بالایی دارد. در این پژوهش، نسخه‌ای ارتقایافته از الگوریتم برنامه‌ریزی حرکت مبتنی بر نیرو (FMP) با عنوان CAR-FMP برای ناوبری خودروهای خودران شهری ارائه شده است. در الگوریتم پیشنهادی، با معرفی یک نیروی کمکی جدید به‌منظور انجام مانورهای سبقت ایمن، حفظ فاصله مناسب با سایر خودروها و تنظیم پویای سرعت در پیچ‌ها، پایداری و روانی حرکت خودرو به‌صورت محسوسی بهبودیافته است. به‌منظور ارزیابی عملکرد، آزمایش‌ها در محیط شبیه‌سازی CARLA و در دو سناریوی استاندارد CoRL2017 و NoCrash انجام‌شده‌اند. نتایج نشان می‌دهد که روش پیشنهادی نسبت به الگوریتم‌های یادگیری‌محور متداول، به‌طور میانگین موجب افزایش ۶ درصدی نرخ موفقیت در سناریوی CoRL2017 و افزایش ۱۲ درصدی در سناریوی NoCrash با ترافیک متراکم می‌شود. درمجموع، چارچوب CAR-FMP با تکیه بر کنترل توزیع‌شده و عدم وابستگی به مسیرهای ازپیش‌تعیین‌شده، رویکردی مؤثر و مقیاس‌پذیر برای ناوبری ایمن و کارای خودروهای خودران در محیط‌های شهری فراهم می‌سازد.

: 10.22034/ABMIR.2025.23796.1178

E-ISSN: 2821-2037
/The Author 2026. Published by Yazd University This is an open access article under the CC BY 4.0 License (https://creativecommons.org/licenses/by/4.0/).

۱- مقدمه

وسایل نقلیه خودران به‌عنوان یکی از دستاوردهای تحول‌آفرین در حوزه فناوری حمل‌ونقل، نویدبخش ارتقای ایمنی، بهبود بهره‌وری و افزایش رفاه کاربران هستند [۱]. در این سامانه‌ها، «برنامه‌ریزی حرکت[1]» از جمله مهم‌ترین و در عین حال چالش‌برانگیزترین اجزاء به شمار می‌رود [۲]. برنامه‌ریزی حرکت به توانایی وسیله نقلیه در تصمیم‌گیری مستقل درباره مسیر حرکت، اجتناب از برخورد با موانع و رسیدن ایمن و بهینه به مقصد اشاره دارد. این موضوع در محیط‌های شهری، به دلیل تراکم بالای ترافیک، حضور موانع متحرک و شرایط محیطی متغیر، با پیچیدگی بیشتری همراه است. روش‌های برنامه‌ریزی حرکت به دو دسته کلی تقسیم می‌شوند: روش‌های مبتنی بر یادگیری [۳ و ٤]، نظیر یادگیری تقلیدی[2] [۵-۸] و یادگیری تقویتی[3] [٦ و ۹-۱۳] با بهره‌گیری از داده‌های آموزشی یا تعامل مستقیم با محیط، سیاست‌های رانندگی را فرا می‌گیرند. در روش‌های مبتنی بر یادگیری تقلیدی، یک شبکه عصبی عمیق از روی داده‌های رانندگی انسان آموزش می‌بیند تا تصمیم‌های رانندگی مثل فرمان، ترمز و گاز را پیش‌بینی کند. این رویکردها به دلیل برخورداری از قابلیت یادگیری تطبیقی، در برخی سناریوها عملکرد مطلوبی نشان می‌دهند؛ با این حال، به حجم قابل‌توجهی از داده نیاز دارند و معمولاً در مواجهه با سناریوهای جدید و ناشناخته با محدودیت و ضعف همراه می‌شوند.

از سوی دیگر در الگوریتم‌های یادگیری تقویتی عامل رانندگی از طریق تعامل مستقیم با محیط و بر اساس دریافت پاداش‌ها و جریمه‌ها آموزش می‌بیند. ورودی شامل تصاویر دوربین و وضعیت خودرو است و خروجی فرمان‌های سطح پایین رانندگی (گاز، ترمز، فرمان) است. مزیت اصلی روش یادگیری تقویتی بی‌نیازی از داده انسانی است، اما آموزش آن پرهزینه و زمان‌بر بوده و نیازمند قدرت پردازشی بالا است [۱٤]. برخی از الگوریتم‌های این حوزه مانند الگوریتم CIRL [۸] رویکردی ترکیبی اتخاذ کرده‌اند. این الگوریتم از یادگیری تقلیدی برای یادگیری سیاست اولیه رانندگی و از یادگیری تقویتی برای بهبود سیاست در شرایط پیچیده و پاداش‌محور استفاده می‌کند. گرچه این الگوریتم‌ها نسبت به روش‌های قبلی عملکرد بهتری دارند اما همچنان در سناریوهای ناشناخته ممکن است دچار خطا شوند.

روش‌های غیرمبتنی بر یادگیری شامل رویکردهایی نظیر درخت‌های تصادفی با کاوش سریع (RRT) [۱۵]، کنترل پیش‌بین مدل (MPC) [۱٦] و الگوریتم‌های مبتنی بر نیرو مانند FMP [۱۷] هستند که بر اساس مدل‌های ریاضی و فیزیکی، مسیر حرکت را تعیین می‌کنند. این روش‌ها هرچند در محیط‌های ساختاریافته قابل‌اعتماد هستند، اما در محیط‌های شهری پویا و واقعی از انعطاف‌پذیری کافی برخوردار نیستند. در این میان، الگوریتم برنامه‌ریزی حرکت مبتنی بر نیرو (FMP) به‌عنوان روشی شناخته‌شده، حرکت عامل‌ها (مانند خودروها) را به‌صورت برآیند نیروهای مختلف مدل می‌کند؛ از جمله نیروی دافعه از موانع و سایر عامل‌ها و نیروی جاذبه به سمت هدف. این الگوریتم در محیط‌های چندعامله و متراکم توانسته است حرکت بدون برخورد و بهینه‌ای ارائه دهد، با این حال، کاربرد مستقیم آن در خودروهای خودران شهری با چالش‌هایی همراه است؛ از جمله نیاز به تغییر مسیر آنی، اجرای مانورهایی نظیر سبقت و رعایت قوانین راهنمایی و رانندگی.

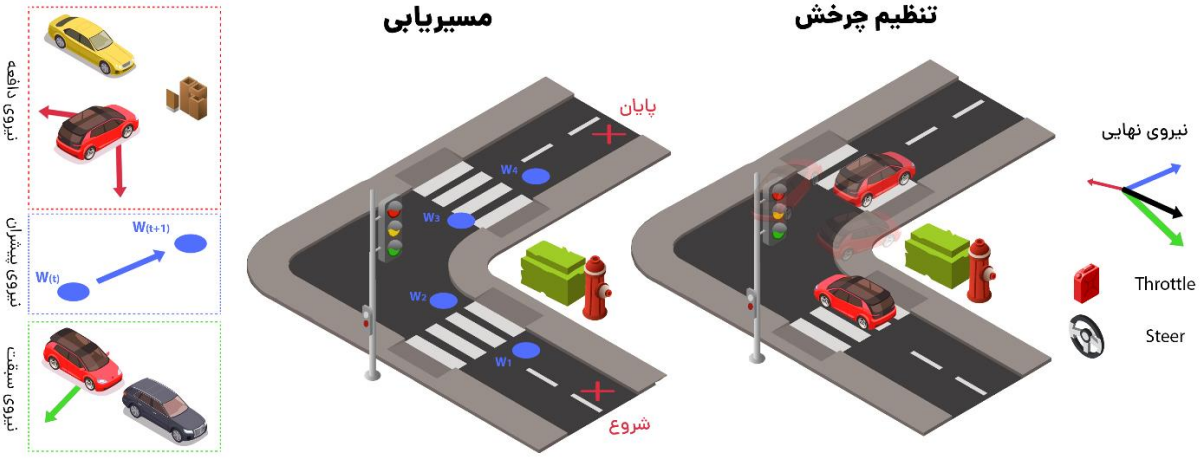
در این مقاله، نسخه‌ای ارتقایافته از الگوریتم FMP با عنوان CAR-FMP معرفی می‌شود که متناسب با نیازهای ناوبری خودروهای خودران در محیط‌های شهری طراحی‌شده است. در این نسخه، با افزودن یک نیروی سوم ویژه برای سبقت ایمن، امکان تنظیم دینامیکی سرعت هنگام گردش در پیچ‌ها و ادغام با سامانه‌های مسیریابی لحظه‌ای، عملکرد الگوریتم به‌طور قابل‌توجهی نسبت به نسخه کلاسیک FMP بهبودیافته است. بدین‌ترتیب، خودرو قادر خواهد بود بدون نیاز به مسیرهای ازپیش‌محاسبه‌شده، در لحظه مسیر بهینه خود را تعیین کرده و در برابر موانع و تغییرات ترافیکی واکنش مناسب نشان دهد. برای ارزیابی عملکرد الگوریتم CAR-FMP، از شبیه‌ساز CARLA 0.9.6 [۱٤] استفاده‌شده است که

[1] Motion planning
[2] Immitation Learning
[3] Reinforcement Learning

یک محیط متن‌باز و واقع‌گرایانه برای بررسی الگوریتم‌های رانندگی خودکار است. آزمایش‌ها در شرایط مختلف جوی و ترافیکی، نشان می‌دهند که این الگوریتم نسبت به روش‌های پایه‌ای مانند کنترل خودکار پیش‌فرض CARLA و دیگر الگوریتم‌های برنامه‌ریزی حرکت، عملکرد روان‌تر، سریع‌تر و ایمن‌تری ارائه می‌دهد. همچنین در سناریوهایی مانند سبقت‌گیری و تغییر مسیر ناگهانی، نرخ موفقیت خودرو به شکل قابل‌توجهی افزایش‌یافته است.

درمجموع، الگوریتم پیشنهادی CAR-FMP (شکل ۱) با بهره‌گیری از ترکیب کنترل توزیع‌شده، مدل‌سازی فیزیکی و مسیریابی لحظه‌ای، الگویی مؤثر و کارآمد برای کاربردهای واقعی رانندگی خودکار در محیط‌های شهری ارائه می‌دهد.

ساختار ادامه این مقاله به‌صورت زیر سازمان‌دهی شده است. در بخش دوم، پیشینه پژوهش و رویکردهای موجود درزمینه برنامه‌ریزی حرکت خودروهای خودران مرور می‌شود. بخش سوم به تشریح الگوریتم پیشنهادی CAR-FMP و اجزای اصلی آن اختصاص دارد. در بخش چهارم، نتایج شبیه‌سازی و ارزیابی عملکرد الگوریتم در محیط CARLA ارائه و با روش‌های مرجع مقایسه شده است. درنهایت، بخش پنجم به جمع‌بندی یافته‌ها و پیشنهاد مسیرهای آینده پژوهش اختصاص‌یافته است.



شکل (۱): ناوبری خودکار در شهر با الگوریتم CAR-FMP

۲- پیشینه پژوهش

برنامه‌ریزی حرکت در خودروهای خودران نقش اساسی در تضمین ایمنی و کارایی ناوبری در محیط‌های پیچیده و غیرقابل پیش‌بینی ایفا می‌کند. تاکنون رویکردهای متعددی برای حل این مسئله ارائه‌شده‌اند که به‌طورکلی در دو دسته مبتنی بر یادگیری و غیر مبتنی بر یادگیری قرار می‌گیرند. در این بخش، ابتدا روش‌های مبتنی بر یادگیری و سپس روش‌های غیر مبتنی بر یادگیری مرور شده‌اند و در پایان، الگوریتم FMP و محدودیت‌های آن در محیط‌های شهری مورد بررسی قرارگرفته است.

الف) روش‌های مبتنی بر یادگیری

روش‌های مبتنی بر یادگیری یکی از رایج‌ترین رویکردها برای ناوبری خودکار خودروها هستند و امروزه توجه بسیاری از پژوهشگران را به خود جلب کرده‌اند. به‌طورکلی، این روش‌ها را می‌توان در دو دسته اصلی یادگیری تقویتی و یادگیری تقلیدی تقسیم‌بندی کرد.

پژوهش‌های اخیر به‌مرور جامع این روش‌ها و ارائه رویکردهای ترکیبی پرداخته‌اند. برای نمونه، مقاله مروری چن و همکاران [۹] کاربردهای یادگیری تقویتی عمیق (DRL) را در برنامه‌ریزی مسیر و کنترل خودروهای خودران بررسی کرده و چالش‌های کلیدی، مانند ایمنی و انتقال از شبیه‌سازی به دنیای واقعی را تشریح می‌کند. در یک رویکرد عملی‌تر، وانگ و همکاران [۱۰] مدل ترکیبی یادگیری تقویتی و کنترل پیش‌بینی مدل (MPC) را برای مانور تعقیب خودرو ارائه داده‌اند. در این ساختار، عامل RL سیاست بهینه رانندگی را فرا می‌گیرد، درحالی‌که MPC تضمین می‌کند

محدودیت‌های ایمنی و دینامیک خودرو رعایت شود تا مانوری ایمن و بهینه حاصل گردد.

هرچند روش‌های مبتنی بر یادگیری تقویتی [٦ و ١١] توانایی تولید رفتار تطبیقی را دارند، اما به زمان آموزش طولانی، منابع محاسباتی بالا و داده‌های گسترده نیازمندند. از سوی دیگر، روش‌های یادگیری تقلیدی [٥ و ٦] با بهره‌گیری از مشاهدات رانندگی انسان، مدل هدایت خودرو را فرا می‌گیرند. این روش‌ها آموزش سریع‌تری دارند، اما به کیفیت و تنوع داده‌های آموزشی حساس‌اند و در مواجهه با سناریوهای ناشناخته عملکرد ضعیفی از خود نشان می‌دهند.

در یکی از رویکردهای متداول، وظایف رانندگی خودران به سه بخش اصلی ادراک، برنامه‌ریزی و کنترل تقسیم می‌شوند که به آن Pipeline ماژولار [١٤] گفته می‌شود. در این روش، ماژول ادراک با استفاده از شبکه‌های قطعه‌بندی مانند RefineNet، [١٨] اطلاعات معنایی شامل جاده، پیاده‌رو و موانع پویا و ایستا را استخراج می‌کند. سپس، با ترکیب این داده‌ها و طبقه‌بند مبتنی بر شبکه AlexNet [١٩] موقعیت تقاطع‌ها تخمین زده می‌شود. علی‌رغم سادگی و ساختار ماژولار، وابستگی شدید این روش به دقت ماژول ادراک و فقدان انطباق‌پذیری در شرایط پیچیده، از محدودیت‌های آن است.

یکی از رویکردهای متداول در آموزش سامانه‌های رانندگی خودکار، استفاده از روش‌های یادگیری تقلیدی است که در آن شبکه عصبی بر اساس داده‌های رانندگی انسان آموزش می‌بیند. برای نمونه، در پژوهش ارائه شده توسط چنگ و همکاران [٥] از داده‌های تصویری، فرمان‌های سطح بالا و اعمال متناظر راننده برای آموزش شبکه استفاده‌شده است تا بتواند رفتار راننده را در شرایط مشابه پیش‌بینی کند. این رویکرد نسبت به مدل‌های مبتنی بر قوانین، توانایی بیشتری در انطباق با محیط دارد؛ با این حال، در سناریوهای ناشناخته و شرایط خارج از داده‌های آموزشی ممکن است دچار ناپایداری شود.

روش مبتنی بر یادگیری تقویتی ارائه می‌دهد بدون نیاز به داده‌های رانندگی انسانی، سیاست رانندگی را صرفاً از طریق تعامل با محیط و دریافت پاداش می‌آموزد. نمونه‌ای از این روش توسط لیلیکراپ و همکاران [١١] ارائه شده است. در این روش، خودرو در هر مرحله بر اساس فرمان سطح بالا مسیر بهینه‌ای به سوی هدف طی می‌کند. با وجود توانایی بالای این رویکرد در یادگیری رفتار بهینه، حجم بالای نمونه‌های آموزشی و دشواری همگرایی پایدار، از چالش‌های اصلی آن به‌شمار می‌رود.

رویکرد CIRL [٦] ارائه‌شده توسط Xiaodan Liang و همکاران، با ترکیب یادگیری تقلیدی و تقویتی، سیاست رانندگی بهینه را در محیط‌های پیچیده شهری می‌آموزد. این مدل با استفاده از الگوریتم DDPG و ورودی‌های تصویری قادر به انجام مانورهایی مانند پیچیدن، توقف و پیروی از مسیر است.

با وجود پیشرفت‌ها، تمامی این روش‌ها در شرایط غیرقابل پیش‌بینی مانند حضور عابر پیاده، تغییر ناگهانی مسیر خودروهای دیگر یا شرایط آب‌وهوایی متغیر، دچار ناپایداری عملکرد می‌شوند. از این رو، رویکردهای ترکیبی و هیبریدی مطرح‌شده‌اند که در آن‌ها یک بخش از سیستم مبتنی بر یادگیری داده‌محور و بخش دیگر مبتنی بر کنترل قطعی عمل می‌کند. اگرچه این روش‌ها به لحاظ نظری مزایای هر دو رویکرد را ترکیب می‌کنند، اما چالش‌هایی همچون تعادل بین قابلیت تطبیق و تضمین ایمنی در زمان واقعی همچنان باقی است.

ب) روش‌های غیر مبتنی بر یادگیری

این دسته شامل الگوریتم‌هایی مانند درخت‌های تصادفی با کاوش سریع [١٥] کنترل پیش‌بین مدل (MPC) [١٦]، و نقشه‌های احتمالی [٢٠] (PRM) است که بر پایه مدل‌سازی ریاضی و کنترل قطعی عمل می‌کنند. در این روش‌ها، مسیر از پیش محاسبه می‌شود و سامانه بدون نیاز به داده آموزشی خاص، حرکت خودرو را برنامه‌ریزی می‌کند.

برای مثال، الگوریتم RRT در فضاهای با ابعاد بالا کارآمد است اما زمان‌بر بوده و برای ناوبری بلادرنگ در محیط‌های شهری مناسب نیست. از سوی دیگر، MPC با تعریف تابع هزینه و محدودیت‌ها، قادر است تصمیم‌های دقیق برای کنترل لحظه‌ای بگیرد اما نیازمند مدل‌سازی دقیق از دینامیک خودرو و محیط است.

در میان روش‌های کنترل قطعی، الگوریتم‌هایی نیز ارائه شده‌اند که ترکیب یادگیری و MPC را پیشنهاد می‌دهند. برای نمونه،

Keuntaek Lee [۱۲] و همکارانش، با استفاده از Deep Inverse Reinforcement Learning، یک نقشه هزینه قابل تفسیر را برای رانندگی در ترافیک سنگین پیشنهاد کرده‌اند که به‌خوبی با MPC ترکیب شده و نتایج قابل توجهی ارائه داده‌اند. در میان روش‌های غیر یادگیری، الگوریتم FMP [۱۷] به‌دلیل سادگی و قابلیت اجرای توزیع‌شده، به‌عنوان پایه این پژوهش انتخاب شده است و در ادامه با جزئیات بررسی خواهد شد.

الف) الگوریتم برنامه‌ریزی حرکت مبتنی بر نیرو (FMP)

الگوریتم برنامه‌ریزی حرکت مبتنی بر نیرو، رویکردی مبتنی بر فیزیک است که حرکت عامل‌ها را به‌عنوان نتیجه‌ای از برآیند نیروهای مختلف مدل‌سازی می‌کند. این الگوریتم نخستین بار برای کنترل دسته‌جمعی عامل‌ها در محیط‌های چندعامله توسعه یافت و از رفتار گروهی پرندگان و موجودات زنده الهام گرفته‌شده است. درFMP ، هر عامل (نظیر خودرو) با اعمال نیروهایی نظیر نیروی هدایت‌گر به سمت هدف و نیروی دافعه از سایر عامل‌ها و موانع مسیر حرکت خود را تعیین می‌کند.

سه نیروی اصلی در این مدل عبارت‌اند از:

۱) نیروی دافعه[1]

این نیرو نقش جلوگیری از برخورد با سایر عامل‌ها یا موانع را بر عهده دارد. به‌صورت ریاضی، برای عامل که در موقعیت $\boldsymbol{p_i}$ قرار دارد، نیروی دافعه محاسبه‌شده در زمان t از سایر همسایگان $\boldsymbol{j \in N_i}$ در رابطه (۱) تعریف‌شده است:

$$\mathbf{f_t^R} = \sum_{\mathbf{j \in N_i}} (\boldsymbol{\varphi}(\|\mathbf{p_j} - \mathbf{p_i}\|) \frac{\mathbf{p_j} - \mathbf{p_i}}{\|\mathbf{p_j} - \mathbf{p_i}\|}) \qquad (۱)$$

که در آن $\boldsymbol{\varphi(z)}$تابع پتانسیل دافعه است و در رابطه (۲) آمده است:

$$\boldsymbol{\varphi}(\mathbf{z}) = \begin{cases} -\boldsymbol{\rho} \times (\mathbf{z} - \mathbf{r})^2 & \mathbf{0} < \mathbf{z} < \mathbf{r} \\ \mathbf{0} & \mathbf{z} \geq \mathbf{r} \end{cases} \qquad (۲)$$

در این رابطه، $\boldsymbol{r}$ شعاع تأثیر نیروی دافعه (شعاع تعامل)، و $\boldsymbol{\rho}$ ضریب شدت نیروی دافعه است. به این ترتیب، هرچه عامل $\boldsymbol{i}$ به عامل دیگری نزدیک‌تر شود، نیروی دافعه‌ای که احساس می‌کند بیشتر خواهد بود و آن را از برخورد بازمی‌دارد.

۲) نیروی هدایت‌گر[2]

این نیرو عامل را به سمت موقعیت هدف که در زمان t با $\boldsymbol{\tau_t}$ نشان داده‌شده است هدایت می‌کند. تعریف ریاضی آن در زمان t در رابطه (۳) آمده است:

$$\mathbf{f_t^Y} = -\mathbf{c_1}(\mathbf{p_t} - \boldsymbol{\tau_t}) - \mathbf{c_2}(\mathbf{v_t}) \qquad (۳)$$

که در آن $\boldsymbol{c_1}$، ضریب نیروی جاذبه به سمت هدف، $\boldsymbol{c_2}$، ضریب مقاومتی وابسته به سرعت و $\boldsymbol{v_t}$ ، سرعت فعلی عامل است.

این نیرو باعث می‌شود عامل با سرعت و جهت مناسب به‌سوی هدف حرکت کند، درحالی‌که هم‌زمان ملاحظاتی مانند کاهش نوسانات یا توقف ناگهانی را نیز در نظر می‌گیرد.

۳) نیروی نهایی و فاصله ایمن تضمین‌شده

با جمع دو نیروی بالا، برآیند نیروی اعمال‌شده به عامل به دست می‌آید. این نیرو جهت و شتاب عامل را مشخص می‌کند. افزون بر این، برای اطمینان از حفظ فاصله ایمن میان عامل‌ها، در تحلیل نظری الگوریتم FMP اثبات‌شده است که این الگوریتم قادر به حفظ حداقل فاصله مجاز از پیش تعیین‌شده بین دو عامل است.

٤) ویژگی‌های ساختاری الگوریتم FMP

توزیع‌شده و مقیاس‌پذیر: هر عامل به‌صورت مستقل و فقط با اطلاعات محلی تصمیم‌گیری می‌کند که نیاز به کنترل مرکزی را حذف کرده و کارایی الگوریتم را در محیط‌های شلوغ افزایش می‌دهد.

عدم نیاز به مسیر از پیش محاسبه‌شده: عامل‌ها باتوجه به موقعیت هدف و موقعیت لحظه‌ای سایر عامل‌ها، مسیر خود را در لحظه تعیین می‌کنند.

پشتیبانی از هزاران عامل: به‌دلیل ساده‌سازی ساختار کنترلی، الگوریتم قادر است حتی در حضور هزاران عامل نیز بدون برخورد و با حفظ بازدهی، عملکرد مناسبی ارائه دهد.

با وجود تمام این مزایا، نسخه پایه FMP برای شرایط واقعی شهری کافی نیست؛ چراکه از عهده سناریوهایی مانند سبقت، توقف در تقاطع، کنترل فرمان و سرعت در پیچ‌ها، یا رعایت علائم رانندگی برنمی‌آید. در ادامه مقاله، نسخه ارتقا یافته‌ای با نام CAR-FMP معرفی می‌شود که با افزودن نیروهای اضافی، بهبود در محاسبات

[1] Repulsive Force

[2] Navigational Force

سرعت و مسیر، و استفاده از شبیه‌ساز CARLA، بر این محدودیت‌ها غلبه می‌کند.

۳- روش پیشنهادی

در این پژوهش، الگوریتم FMP به‌گونه‌ای توسعه‌یافته که بتواند در شرایط پویا، پیچیده و غیرقابل پیش‌بینی محیط‌های شهری، ناوبری دقیق، ایمن و سازگار با ترافیک را برای خودروهای خودران ممکن سازد. در این روش، با الهام از مدل‌های کنترل توزیع‌شده و شبیه‌سازی فیزیکی، سه نیروی اصلی شامل نیروی هدایت‌گر، نیروی دافعه و نیروی سبقت طراحی‌شده‌اند. هر خودرو به‌صورت مستقل، و بدون نیاز به اطلاعات جهانی، از این نیروها برای هدایت خود در مسیر استفاده می‌کند. این رویکرد به‌ویژه در سناریوهای چندعاملی که هماهنگی بین وسایل نقلیه اهمیت دارد، عملکردی منعطف و پایدار ارائه می‌دهد.

فرآیند ناوبری خودرو به پنج مرحله پیوسته تقسیم می‌شود:

گام اول: مسیریابی

در روش پیشنهادی در اولین مرحله، مسیر کلی حرکت خودرو به مجموعه‌ای از هدف‌های موقت[1] تقسیم می‌شود. این اهداف موقتی همان نقاطی هستند که در امتداد مسیر نهایی قرار می‌گیرند و خودرو باید به ترتیب به سمت آن‌ها حرکت کند. این نقاط در شکل ۱ قسمت (ب) با نام‌های W1 تا W4 نشان داده‌شده‌اند. با حرکت از یک نقطه موقت به نقطه بعدی، خودرو به‌تدریج به مقصد نهایی نزدیک می‌شود. برای تولید این مسیر در این پژوهش از الگوریتم A* [۲۱] استفاده‌شده است. A* الگوریتمی مبتنی بر جست‌وجوی آگاهانه است که با استفاده از تابع هزینه‌ای شامل فاصله طی‌شده و تخمین فاصله باقی‌مانده تا مقصد، بهینه‌ترین مسیر را میان گره‌های موجود در گراف انتخاب می‌کند. این الگوریتم با بهره‌گیری از اطلاعات توپولوژیکی نقشه محیط که از شبیه‌ساز CARLA دریافت شده‌اند، از نقاط ورودی و خروجی شروع کرده و با یافتن بهترین دنباله از گره‌های میانی، مسیری هموار و قابل دنبال‌کردن برای هدایت خودرو فراهم می‌سازد. مسیر بین هر جفت از این نقاط به‌صورت رشته‌ای از نقاط میانی تولید می‌شود تا با حفظ یکنواختی در فاصله، حرکت روان و ایمن خودرو تضمین شود. برای مثال، اگر مسیر هدف دارای ۶۵۰ متر طول و ۶ پیچ باشد، الگوریتم ۵۶ نقطه بین‌راهی تولید می‌کند که هر کدام به‌عنوان یک گام موقت در مسیر در نظر گرفته می‌شوند. این طراحی باعث می‌شود خودرو بتواند با انعطاف بالا، در برابر موانع و شرایط متغیر جاده واکنش مناسب نشان دهد.

گام دوم: محاسبه نیروها

پس از مشخص‌شدن مسیر، خودرو باید با استفاده از نیروهای فیزیکی به‌صورت بلادرنگ خود را در مسیر هدایت کند. در این مرحله، سه نیروی اصلی زیر به‌طور هم‌زمان محاسبه می‌شوند:

۱) نیروی هدایت‌گر:

این نیرو که در شکل ۱ قسمت (الف) (تحت عنوان نیروی پیشران) نشان داده‌شده است، خودرو را به سمت نقطه بین‌راهی فعلی هدایت می‌کند. هنگامی که فاصله خودرو از نقطه هدف ازحد مشخصی[2] کمتر شود، هدف بعدی انتخاب می‌شود و جهت نیرو به‌روزرسانی می‌گردد. این فرآیند تا رسیدن به نقطه پایانی مسیر تکرار می‌شود .برای محاسبه این نیرو در الگوریتم CAR-FMP از فرمولی مشابه رابطه (۳) در الگوریتم FMP استفاده می‌شود.

۲) نیروی دافعه:

این نیرو برای جلوگیری از برخورد با سایر وسایل نقلیه، موانع ثابت، جدول‌ها و دیگر عناصر محیطی فعال می‌شود و در شکل ۱ قسمت (الف) (تحت عنوان دافعه) نشان داده‌شده است. طبق مختصات موانع که از طریق شبیه‌ساز دریافت می‌شود فاصله هر خودرو از سایر اجسام محاسبه‌شده و درصورتی‌که کمتر از یک آستانه ایمنی باشد، نیرویی در جهت عکس مانع اعمال می‌کند. شدت این نیرو نیز متناسب با فاصله تنظیم می‌شود؛ یعنی هرچه خودرو به مانع نزدیک‌تر باشد، نیروی دافعه قوی‌تر خواهد بود. این نیرو با استفاده از رابطه‌ای شبیه به رابطه (۱) در الگوریتم FMP محاسبه می‌شود.

[1] Waypoint

[2] Threshold

۳) نیروی سبقت‌گیر[1]:

یکی از نوآوری‌های مهم الگوریتم CAR-FMP، تعریف نیرویی ویژه برای انجام مانور سبقت است. در محیط‌های شهری، خودرو ممکن است با وسایل نقلیه کندتر مواجه شود که ادامه مسیر مستقیم را مسدود کرده‌اند. در چنین شرایطی، نیروی سوم فعال می‌شود. این نیرو با تحلیل سرعت نسبی، فاصله جانبی، و امکان حرکت در سمت چپ مسیر، شرایط سبقت ایمن را فراهم می‌کند این نیرو که در شکل ۱ قسمت (الف) (تحت عنوان نیروی سبقت) نشان داده‌شده است بر اساس رابطه (٤) محاسبه می‌شود. در این رابطه $\varphi(z)$ تابع پتانسیل دافعه است که در رابطه (۲) تعریف‌شده است و f_{t-1} برآیند نیروی ما در لحظه t-1 است همچنین α زاویه میان عامل و ایجنت دیگر یا مانع است. p_i و p_j به ترتیب محل فعلی خوردو ما و محل فعلی خودرو یا مانع موجود در مسیر است.

رابطه (٤) بر پایه منطق مدل‌های مبتنی بر نیرو (Force-based Models) توسعه‌یافته است که در آن جهت و شدت نیروها متناسب با زاویه دید و فاصله عامل از موانع تعیین می‌شود. در نسخه کلاسیک FMP ، برآیند نیروها صرفاً شامل نیروی دافعه از و نیروی جاذبه به سمت هدف است و هیچ مؤلفه‌ای برای تغییر مسیر فعال در راستای سبقت وجود ندارد. در الگوریتم CAR-FMP، نیروی سبقت $\vec{f}_t^{\,o}$ به‌عنوان مؤلفه‌ای جدید معرفی‌شده که جهت آن عمود بر بردار نیروی هدایت‌گر فعلی در سمت چپ مسیر قرار دارد تا رفتار سبقت واقعی شبیه‌سازی شود.

منشأ هندسی این نیرو از مشاهده تجربی در محیط شبیه‌سازی CARLA به‌دست آمده است: هنگامی‌که زاویه بین جهت حرکت فعلی خودرو و خط اتصال آن با خودروی مانع (زاویهα) بیش از ۳۰ درجه می‌شود، خودرو در موقعیتی قرار دارد که از دید رانندگی انسانی نیز شرایط برای آغاز سبقت ایمن فراهم است. شدت نیرو نیز بر اساس تابع پتانسیل مشابه نیروی دافعه FMP تعریف‌شده، اما با یک ضریب مقیاس کوچک‌تر تا مسیر سبقت نرم و کنترل‌پذیر باقی بماند. به‌عبارت‌دیگر، رابطه (٤) تعمیمی از مدل کلاسیک FMP با افزودن مؤلفه جانبی کنترل‌شده است که به‌صورت نیمه‌تجربی و با تنظیمات متوالی در محیط CARLA بهینه‌سازی شده است.

$$\vec{f}_t^{\,o}=\begin{cases}\varphi(z)\times \overrightarrow{\perp f_{t-1}}\times\cos\alpha & \text{if } \dfrac{\overrightarrow{f_{t-1}}\cdot\overrightarrow{(p_j-p_i)}}{|f_{t-1}|\,|(p_j-p_i)|}>\cos 30^\circ\\ 0 & \text{otherwise}\end{cases} \qquad (٤)$$

گام سوم: ترکیب نیروها و کنترل نهایی

پس از محاسبه سه نیروی ذکرشده f_t، بردار نهایی حاصل از برآیندگیری تمام نیروها، نیروی وارد شده برای حرکت خودرو را تعیین می‌کند (شکل ۱ قسمت (ج) برآیند نیروها با فلش سیاه‌رنگ نشان داده‌شده است). نحوه محاسبه این نیرو در رابطه (۵) نشان داده‌شده است.

$$f_t = f_t^R + f_t^\gamma + f_t^o \qquad (۵)$$

سپس متغیرهای کنترل مانند شتاب[2]و زاویه فرمان[3] براساس این نیرو تنظیم می‌شوند. پس از محاسبه نیروی نهایی حاصل از برآیند همه نیروها، بسته به شرایط ممکن است تغییرات کنترلی برای تطابق با محدودیت‌های فیزیکی وسیله نقلیه و قوانین رانندگی روی نیروی نهایی اعمال شود. برای مثال اگر سرعت خودرو از مقدار مجاز فراتر رود (به عنوان مثال ٤٥ کیلومتر بر ساعت)، مقدار نیرو صفر می‌شود تا سرعت از سرعت مجاز بالاتر نرود. در پایان این گام وقتی زاویه نهایی و جهت فعلی خودرو اختلاف اندکی (به عنوان مثال کمتر از ٤ درجه) داشته باشند، زاویه فرمان صفر شده و خودرو به‌صورت پایدار در مسیر حرکت می‌کند.

گام چهارم: کنترل عملکرد خودرو در پیچ‌ها

در این مرحله، الگوریتم عملکرد خودرو در هنگام پیچیدن را تنظیم می‌کند تا ایمنی آن حفظ شود. این کار با تحلیل زاویه فرمان انجام می‌شود. اگر زاویه فرمان بیشتر از ۰/۲ باشد، یعنی خودرو وارد یک پیچ شده است.

[1] Overtaking Force
[2] Throttle
[3] Steer

در این حالت، بسته به سرعت فعلی خودرو، مقدار شتاب تنظیم می‌شود:

- در سرعت‌های بالا (بیش از ۳۰ کیلومتر بر ساعت): شتاب به مقدار بسیار پایین (مثلاً ۰/۱) کاهش می‌یابد تا خودرو به‌آرامی کاهش سرعت دهد.
- در سرعت‌های پایین‌تر: شتاب به نصف کاهش می‌یابد تا خودرو با ایمنی بیشتری گردش کند.

این تنظیمات کمک می‌کند تا خودرو از خطرات ناشی از چرخش سریع یا ازدست‌دادن کنترل در پیچ‌ها جلوگیری کند. این مرحله در شکل ۱ قسمت (ج) نشان داده‌شده است.

گام پنجم: حرکت نهایی و تثبیت

در پایان، نیروی نهایی حاصل از مجموع نیروهای محاسبه‌شده، به‌صورت پیوسته در سیستم اعمال می‌شود. این نیرو سرعت، شتاب و جهت خودرو را در هر لحظه تنظیم کرده و واکنش مناسب به موانع، پیچ‌ها و مانورها را فراهم می‌سازد.

در الگوریتم پیشنهادی پس از محاسبه بردار نیروی نهایی مشابه الگوریتم پایه FMP سرعت و مکان بعدی خودرو به ترتیب طبق روابط (٦) و (٧) محاسبه می‌شوند که در آن Δt گام زمانی استفاده‌شده در الگوریتم برای به‌روزرسانی شتاب، سرعت و مکان است و در این مقاله مقدار آن ۰,۰۲ ثانیه در نظر گرفته‌شده است. f_t، v_t و p_t به ترتیب نشان‌دهنده برآیند نیروها (که نحوه محاسبه آن در رابطه (٥) نشان داده شد)، سرعت و مکان در لحظه t است.

$$v_t = v_{t-1} + f_t \Delta t \quad (٦)$$

$$p_t = p_{t-1} + v_t \Delta t \quad (٧)$$

مکانیزم کنترل طوری طراحی‌شده است که از نوسانات غیرضروری جلوگیری می‌کند و با تثبیت فرمان و سرعت در لحظات حساس، مصرف انرژی و پیچیدگی محاسباتی را کاهش می‌دهد.

درنهایت پیش از رسیدن به هر یک از اهداف موقت، هدف موقت بعدی جایگزین می‌شود تا از کاهش سرعت عامل جلوگیری شود و سرعت عامل مجدداً طبق روند گفته‌شده محاسبه می‌شود و هر بار تا رسیدن به مقصد نهایی این روند تکرار می‌شود.

درمجموع، الگوریتم CAR-FMP با ادغام سه نوع نیروی هوشمند، و استفاده از معماری توزیع‌شده، روشی مؤثر و پایدار برای ناوبری خودروهای خودران در سناریوهای واقعی و پیچیده شهری ارائه می‌دهد. این رویکرد برخلاف روش‌های کلاسیک، بدون نیاز به مسیرهای ایستا، توانایی واکنش به محیط پویا را فراهم می‌سازد.

٤- نتایج و ارزیابی

برای ارزیابی عملکرد الگوریتم پیشنهادی CAR-FMP، از شبیه‌ساز CARLA [١٤] به‌عنوان یکی از پیشرفته‌ترین و دقیق‌ترین محیط‌های شبیه‌سازی برای تحقیق درزمینه رانندگی خودران استفاده‌شده است. در این شبیه‌ساز، سناریوهایی متنوع شامل شرایط آب‌وهوایی مختلف، تراکم ترافیک متفاوت، موانع متحرک و ایستا، و انواع نقشه‌های شهری در نظر گرفته‌شده است تا عملکرد الگوریتم در محیط‌های نزدیک به واقعیت مورد بررسی قرار گیرد. مقادیر نهایی انتخاب‌شده برای پارامترهای الگوریتم به‌صورت $c_1 = 1,\ c_2 = 1,\ r = 10,\ \rho = 0.7$ هستند. این مقادیر بر اساس انجام آزمایش‌های متعدد در شبیه‌ساز CARLA و ارزیابی دقیق عملکرد مدل، به‌صورت تجربی تعیین‌شده‌اند.

٤-١ معیار ارزیابی

برای اعتبارسنجی عملکرد مدل و الگوریتم‌های طراحی‌شده در این پژوهش، از روش ارزیابی کمی مبتنی بر تحلیل رفتار عامل در شرایط شبیه‌سازی‌شده استفاده‌شده است. مشابه با مقاله اصلی CARLA [١٤] معیار اصلی مورداستفاده در این ارزیابی درصد موفقیت الگوریتم در هدایت ماشین به سمت هدف نهایی تعیین‌شده بدون برخورد با موانع ثابت یا متحرک محیطی است. در این معیار، برای کسب موفقیت عامل (خودروی خودران) باید در بازه زمانی مشخص و بدون تصادف به مقصد برسد در غیر این صورت عدم موفقیت خودرو برای این سناریو در نظر گرفته می‌شود. این ارزیابی به‌گونه‌ای طراحی‌شده‌اند که قابلیت تعمیم‌پذیری الگوریتم به سناریوهای مختلف رانندگی و میزان تطابق آن با اهداف رفتاری سیستم را مشخص نمایند.

٤-٢ سناریوهای ارزیابی

برای ارزیابی دقیق‌تر و مقایسه‌پذیر عملکرد، از دو سناریو استاندارد در حوزه رانندگی خودران تعریف‌شده در این شبیه‌ساز استفاده‌شده است:

سناریوی CoRL2017 در شبیه‌ساز CARLA، اولین بنچ‌مارک رسمی برای ارزیابی سامانه‌های رانندگی خودران در محیط‌های شهری است که در مقاله اصلی CARLA معرفی شد [١٤]. این بنچ‌مارک شامل چهار سناریوی مختلف با سطوح دشواری افزایشی است که برای ارزیابی عملکرد الگوریتم‌های مختلف طراحی‌شده‌اند:

١) رانندگی در مسیر مستقیم (Straight Driving) : در این سناریو، خودرو باید بدون تغییر مسیر و در شرایط ترافیکی ساده حرکت کند.

٢) رانندگی با یک پیچ (One Turn Navigation) : در این سناریو، خودرو باید پس از حرکت در مسیر مستقیم، یک پیچ را با موفقیت طی کند.

٣) ناوبری در مسیر پیچیده‌تر(Navigation) : در این سناریو، خودرو باید در یک مسیر پیچیده‌تر با چندین پیچ و تغییر مسیر حرکت کند.

٤) ناوبری با موانع دینامیک (Navigation with Dynamic Objects): در این سناریو، خودرو باید در مسیری حرکت کند که در آن موانع متحرک مانند خودروهای دیگر وجود دارند و باید با آن‌ها تعامل داشته باشد.

سناریوی دوم NoCrash است که یکی از چالش‌های پیشرفته محیط CARLA محسوب می‌شود. NoCrash برای شبیه‌سازی سه سناریوی مختلف از ترافیک طراحی‌شده است: مسیر بدون ترافیک، مسیر با ترافیک معمولی، و مسیر با ترافیک متراکم. این سناریوها شرایط متنوع و چالش‌برانگیزی را فراهم می‌آورند که امکان سنجش پایداری و دقت سیستم در محیط‌های واقعی‌تر را فراهم می‌سازند. هدف اصلی در این سناریو جلوگیری از وقوع تصادف است. سامانه‌های ارزیابی‌شده باید توانایی حفظ ایمنی خودرو را در طول مسیرهای مختلف نشان دهند.

٤-٣ محیط‌های مورداستفاده

آزمایش‌ها در دو شهر مجازی «Town02» و «Town10» از محیط CARLA اجرا شده‌اند. همچنین سناریوها در شش وضعیت آب‌وهوایی مختلف برای افزایش واقع‌گرایی اجرا شده‌اند: ظهر آفتابی، ظهر پس از باران، ظهر با بارش شدید باران، غروب آفتابی، ظهر ابری پس از باران، غروب بارانی ملایم. برای هر یک تست‌های انجام‌شده، مشابه با شرایط در نظر گرفته‌شده در مقاله اصلی CARLA [١٤]، تعداد ٢٥ آزمایش انجام‌شده و میانگین نتایج این ٢٥ آزمایش در جدول شماره ١ و ٢ گزارش‌شده است. در هر یک از این ٢٥ تست انجام‌شده مقصد ماشین به تصادف از بین نقاط مجاز انتخاب‌شده است. این تعداد در تمام الگوریتم‌های مورد مقایسه به‌صورت یکسان در نظر گرفته‌شده است.

٤-٤ مقایسه با روش‌های موجود

عملکرد الگوریتم CAR-FMP با چهار روش شناخته‌شده درزمینه رانندگی خودران (که در بخش ٢ معرفی شدند) و در محیط CARLA به عنوان الگوریتم پایه پیاده‌سازی شده‌اند مقایسه شده است:

- **MP**: الگوریتم کلاسیک ماژولار پایپلاین [١٤]
- **IL**: الگوریتم مبتنی بر یادگیری تقلیدی [١٤]
- **RL**: الگوریتم مبتنی بر یادگیری تقویتی [١٤]
- **CIRL**: الگوریتم ترکیبی یادگیری تقلیدی و یادگیری تقویتی [١٣]

جدول ١ میزان موفقیت الگوریتم CAR-FMP در مقایسه با چهار روش دیگر در سناریو CoRL2017 را نشان می‌دهد (اعداد بر اساس درصد موفقیت)

همان‌طور که مشاهده می‌شود، الگوریتم CAR-FMP در تمامی وظایف عملکرد بهتری نسبت به روش‌های دیگر ارائه داده است. خصوصاً در سناریوهای پیچیده مانند گردش در پیچ‌ها و مسیر با مانع متحرک نرخ موفقیت افزایش چشم‌گیری نسبت به الگوریتم‌های یادگیری‌محور دارد. دلیل این نتایج را می‌توان به اضافه کردن دو نیروی مربوط به سبقت ($\vec{f}_t^{o}$) و اصلاح نیرو در چرخش‌ها مرتبط دانست. $\vec{f}_t^{o}$ به الگوریتم کمک می‌کند به خوبی از موانع متحرک مانند خودروی جلوتر عبور کند و اصلاح نیرو در

چرخش کمک می‌کند تا در مسیرهای پیچ‌دار خودرو از مسیر منحرف نشود.

جدول (۱): مقایسه درصد موفقیت روش CAR-FMP با روش‌های دیگر در سناریو CoRL2017

وظیفه	MP	IL	RL	CIRL	CAR-FMP
مسیر مستقیم	۵۰	۸۰	٦٨	۹۸	۱۰۰
مسیر با یک پیچ	۵۰	٤٨	۲۰	۸۲	۸۸
مسیر دلخواه	٤٧	٤٤	٦	٦٨	٧٦
مسیر با مانع متحرک	٤٤	٤٢	٤	٦٢	٧٢

در سناریوی NoCrash نیز الگوریتم CAR-FMP با روش‌های پیشرفته LBC [۱۰] و Model-Free RL [٤] که پیاده‌سازی هر دو به عنوان الگوریتم پایه درمحیط شبیه‌ساز CARLA وجود دارد مورد مقایسه قرار گرفت. نتایج به‌دست‌آمده از درصد موفقیت CAR-FMP در مقایسه با دو الگوریتم بالا در جدول ۲ خلاصه‌شده‌اند:

جدول (۲): مقایسه درصد موفقیت روش CAR-FMP با روش‌های دیگر در سناریو NoCrash

وظیفه	LBC	Model-Free RL	CAR-FMP
بدون ترافیک	۱۰۰	۹۹	۱۰۰
ترافیک معمولی	۹٤	۸۷	۸۸
ترافیک متراکم	۵۱	٤۲	۵٤

همان‌گونه که در جدول ۲ مشاهده می‌شود، الگوریتم CAR-FMP در اکثر شرایط ترافیکی خصوصاً در سناریوهای متراکم که تصمیم‌گیری بلادرنگ و اجتناب از برخورد حیاتی است، توانسته است عملکردی بهتر یا برابر نسبت به روش‌های دیگر داشته است، این در حالی است که چون مدل مبتنی بر نیرو است نیاز به مرحله آموزش ندارد. عملکرد الگوریتم CAR-FMP در شرایط «ترافیک معمولی» اندکی پایین‌تر از برخی روش‌های یادگیری‌محور مانند LBC است. علت اصلی این تفاوت به ماهیت مدل مبتنی بر نیرو بازمی‌گردد. در ترافیک معمولی، خودروهای اطراف در فاصله‌های متوسط و با سرعت‌های متغیر حرکت می‌کنند، در نتیجه نیروی دافعه و نیروی سبقت در وضعیت‌های ناپایدار فعال و غیرفعال می‌شوند. این نوسانات کوچک در برآیند نیروها می‌تواند موجب تغییرات جزئی در تصمیم‌های کنترلی خودرو شود و در برخی موارد منجر به کاهش نرخ موفقیت گردد.

از سوی دیگر، در ترافیک متراکم، رفتار خودروها پیش‌بینی‌پذیرتر بوده و فاصله‌ها کوتاه‌ترند، بنابراین نیروهای فیزیکی غالب و پایدارتر عمل می‌کنند؛ در نتیجه، مدل CAR-FMP توانسته است پایداری بالاتری نشان دهد. به‌طورکلی، می‌توان گفت که افت جزئی در شرایط ترافیک معمولی ناشی از حساسیت نیروهای میان‌عامل به فاصله‌های متوسط است و می‌تواند در نسخه‌های آینده با معرفی یک تابع تطبیقی برای تنظیم خودکار ضرایب نیروها برطرف شود. به‌طورکلی نتایج به‌دست‌آمده نشان می‌دهد الگوریتم CAR-FMP در مسیرهای پیچیده با شرایط محیطی پویا (همچون موانع متحرک و پیچ‌های تند)، عملکردی پایدار و قابل‌اعتماد دارد. توانایی تطبیق لحظه‌ای با شرایط مسیر بدون نیاز به مسیر از پیش محاسبه‌شده، باعث افزایش نرخ موفقیت شده است. در مقایسه با الگوریتم‌های یادگیری‌محور، الگوریتم پیشنهادی دارای پیاده‌سازی ساده‌تر، بدون نیاز به آموزش سنگین، و قابل‌اعتماد در شرایط ناشناخته است.

٤-۵ مقایسه تحلیلی با الگوریتم پایه FMP

ازآنجاکه الگوریتم پیشنهادی CAR-FMP بر پایه ساختار الگوریتم کلاسیک FMP توسعه‌یافته است، مقایسه مفهومی و عملکردی میان این دو ضروری است. الگوریتم پایه FMP در اصل برای مدل‌سازی حرکت جمعی عامل‌ها در محیط‌های ساده و بدون قوانین خاص طراحی‌شده و تمرکز آن بر اجتناب از برخورد و رسیدن به هدف با استفاده از برآیند نیروها است.

با این حال، در محیط‌های شهری واقعی، رانندگی خودران نیازمند تصمیم‌گیری‌های سطح بالاتر مانند حفظ خط حرکت، رعایت قوانین رانندگی، انجام مانور سبقت ایمن و تنظیم سرعت در پیچ‌ها است؛ قابلیت‌هایی که در نسخه اصلی FMP وجود ندارند. به همین دلیل، استفاده مستقیم از FMP در شبیه‌ساز CARLA منجر به رفتارهای ناپایدار و افزایش نرخ برخورد در ترافیک پویا می‌شود. در نسخه پیشنهادی CAR-FMP، با افزودن نیروی کمکی سبقت و تنظیم تطبیقی سرعت، این محدودیت‌ها برطرف شده‌اند. به‌طور

مشخص، نیروی سوم معرفی‌شده باعث می‌شود خودرو بتواند در شرایطی که مسیر توسط وسایل نقلیه کندتر مسدود شده است، مانور سبقت ایمن و پیوسته انجام دهد. علاوه‌براین، مکانیزم کنترل سرعت در پیچ‌ها از لغزش و ناپایداری جلوگیری کرده و موجب افزایش نرخ موفقیت در سناریوهای واقعی‌تر شده است. بنابراین، عملکرد ضعیف الگوریتم پایه FMP در مقایسه با CAR-FMP طبیعی است، زیرا نسخه اولیه اساساً برای کاربرد در رانندگی خودران طراحی نشده است.

۵- نتیجه‌گیری

در این پژوهش، یک الگوریتم پیشرفته برای برنامه‌ریزی حرکت خودروهای خودران با نام CAR-FMP معرفی شد که بر پایه مدل مبتنی بر نیرو و ساختار کنترل توزیع‌شده طراحی‌شده است. الگوریتم پیشنهادی با افزودن قابلیت‌هایی نظیر تنظیم سرعت در پیچ‌ها، امکان انجام مانور سبقت‌گیری ایمن، و استفاده از نقاط بین‌راهی موقت، توانست محدودیت‌های نسخه کلاسیک FMP را در سناریوهای واقعی شهری برطرف سازد.

ارزیابی‌های انجام‌شده در شبیه‌ساز CARLA و در سناریوهای متنوع شامل شرایط آب‌وهوایی مختلف، تراکم ترافیک، و موانع متحرک نشان داد که CAR-FMP عملکردی به‌مراتب بهتر از روش‌های یادگیری‌محور دارد. این الگوریتم بدون نیاز به آموزش داده‌محور، توانست در شرایط واقعی‌نما، نرخ موفقیت بالاتری نسبت به سایر روش‌ها ثبت کند.

ویژگی‌های شاخص الگوریتم CAR-FMP شامل موارد زیر است:

- ناوبری ایمن و پایدار در محیط‌های شهری پویا؛
- افزایش نرخ موفقیت در دستیابی به مقصد؛
- پیاده‌سازی سبک، قابل‌توسعه و بدون نیاز به منابع سنگین محاسباتی؛
- عدم نیاز به مسیرهای از پیش‌تعریف‌شده یا داده‌های آموزشی گسترده.

با توجه به نتایج امیدوارکننده الگوریتم CAR-FMP، مسیرهای توسعه آینده می‌تواند شامل موارد زیر باشد:

- افزودن درک معنایی از محیط مانند تشخیص تابلوها، چراغ‌های راهنمایی و عابرین پیاده به‌کمک بینایی ماشین؛
- ادغام با سیستم‌های ادراک بصری یا یادگیری عمیق برای افزایش دقت تصمیم‌گیری در شرایط غیرقابل‌پیش‌بینی؛
- پیاده‌سازی بر روی پلتفرم واقعی به‌منظور ارزیابی عملی در محیط‌های شهری واقعی و نه صرفاً شبیه‌سازی‌شده؛
- بهینه‌سازی مصرف انرژی از طریق تطبیق هوشمند شتاب و مسیر در سناریوهای پرترافیک.

درمجموع، نتایج این پژوهش نشان می‌دهد که الگوریتم‌های مبتنی بر مدل‌سازی فیزیکی و کنترل توزیع‌شده همچنان می‌توانند جایگزین مؤثر و قابل‌اعتمادی برای روش‌های پیچیده یادگیری‌محور درزمینه رانندگی خودکار باشند، به‌ویژه در سناریوهایی که ایمنی، سادگی پیاده‌سازی و واکنش سریع در زمان واقعی اهمیت دارند.

References

[1] L. Samimi-Dehkordi, " Network Attacks Detection in Autonomous Vehicles using Deep Nerul Network", Applied and basic Machine intelligence research, 2, 22025, pp. 83-93 10.22034/abmir.2025.22510.1083

[2] S. Teng, X. Hu, P. Deng, B. Li, Y. Li, Y. Ai, D. Yang, L. Li, Z. Xuanyuan and F. Zhu, "Motion planning for autonomous driving: The state of the art and future perspectives," IEEE Transactions on Intelligent Vehicles, vol. 8, no. IEEE, pp. 3692--3711, 2023.

[3] D. Dauner, M. Hallgarten, A. Geiger and K. Chitta, "Parting with misconceptions about learning-based vehicle motion planning," in Conference on Robot Learning, PMLR, 2023, pp. 1268--1281.

[4] E. Ohn-Bar, A. Prakash, A. Behl, K. Chitta and A. Geiger, "Learning situational driving," in Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 11296--11305.

[5] J. Cheng, Y. Chen and Q. Chen, "Pluto: Pushing the limit of imitation learning-based planning for autonomous driving," arXiv preprint arXiv:2404.14327, 2024.

[6] X. Liang, T. Wang, L. Yang and E. Xing, "Cirl: Controllable imitative reinforcement learning for vision-based self-driving," in Proceedings of the European conference on computer vision (ECCV), 2018, pp. 584--599.

[7] D. Chen, B. Zhou, V. Koltun and P. Krähenbühl, "Learning by cheating," in Conference on robot learning, PMLR, 2020, pp. 66--75.

[8] Codevilla, Felipe, et al. "End-to-end driving via conditional imitation learning." 2018 IEEE international conference on robotics and automation (ICRA). IEEE, 2018.

[9] Y. Chen, C. Ji, Y. Cai, T. Yan and B. Su, "Deep reinforcement learning in autonomous car path planning and control: A survey," arXiv preprint arXiv:2404.00340, 2024.

[10] L. Wang, S. Yang, K. Yuan, Y. Huang and H. Chen, "A combined reinforcement learning and model predictive control for car-following maneuver of autonomous vehicles," Chinese Journal of Mechanical Engineering, vol. 36, no. Springer, p. 80, 2023.

[11] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver and D. Wierstra, "Continuous control with deep reinforcement learning," arXiv preprint arXiv:1509.02971, 2015.

[12] K. Lee, D. Isele, E. A. Theodorou and S. Bae, "Spatiotemporal costmap inference for MPC via deep inverse reinforcement learning," IEEE Robotics and Automation Letters, vol. 7, no. IEEE, pp. 3194--3201, 2022.

[13] Toromanoff, M., Wirbel, E., & Moutarde, F. (2020). End-to-end model-free reinforcement learning for urban driving using implicit affordances. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 7153-7162)

[14] A. Dosovitskiy, G. Ros, F. Codevilla, A. Lopez and V. Koltun, "CARLA: An open urban driving simulator," in Conference on robot learning, PMLR, 2017, pp. 1--16.

[15] LaValle, S. M., & Kuffner, J. J. (2001). Randomized kinodynamic planning. The International Journal of Robotics Research, 20(5), 378–400.

[16] Rawlings, J. B., & Mayne, D. Q. (2009). Model predictive control: Theory and design. Nob Hill Publishing.

[17] S. H. Semnani, R. de, H. Anton and H. H. Liu, "Force-based algorithm for motion planning of large agent," IEEE Transactions on Cybernetics, vol. 52, no. IEEE, pp. 654--665, 2020.

[18] G. Lin, A. Milan, C. Shen, and I. Reid. RefineNet: Multi-path refinement networks for high-resolutionsemantic segmentation. In Computer Vision and Pattern Recognition (CVPR), 2017

[19] A. Krizhevsky, I. Sutskever, and G. E. Hinton. ImageNet classification with deep convolutional neural networks. In Neural Information Processing Systems (NIPS), 2012.

[20] Kavraki, L. E., Švestka, P., Latombe, J. C., & Overmars, M. H. (1996). Probabilistic roadmaps for path planning in high-dimensional configuration spaces. IEEE Transactions on Robotics and Automation, 12(4), 566–580.

[21] P. E. Hart, N. J. Nilsson, and B. Raphael, “A formal basis for the heuristic determination of minimum cost paths,” IEEE Transactions on Systems Science and Cybernetics, vol. 4, no. 2, pp. 100–107, Jul. 1968, doi: 10.1109/TSSC.1968.300136.

Journal of Applied And Basic Machine Intelligence Research (ABMIR)

Applied and Basic Machine Intelligence Research

Vol. 4, No. 1, Spring & Summer 2026, pp. 15 - 27

CAR-FMP: Improved Force-Based Motion Planning for Autonomous Vehicles in Urban Environments

Shakiba Berenjkoub [1], Samaneh Hoseini [2*], Ali Tamizifar[3]

[1]B.Sc. Student, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

[2]Assistant Professor, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

[3]M.Sc. Student, Department of Electrical and Computer Engineering, Isfahan University of Technology, Isfahan, Iran

Article Information

Original Research Paper

Received:
2025 October 11

Accepted:
2025 November 29

Keywords:
Autonomous vehicles, Motion planning, FMP algorithm, Urban navigation, Distributed control, Safe overtaking, CARLA simulator, Real-time decision-making

Corresponding Author*:
samaneh.hoseini@iut.ac.ir

Abstract

Urban autonomous vehicle navigation remains one of the fundamental challenges in intelligent transportation systems, as it requires real-time decision-making in dynamic traffic conditions, continuous interaction with other vehicles, and appropriate responses to unpredictable obstacles. In this study, an enhanced version of the Force-based Motion Planning (FMP) algorithm, entitled CAR-FMP, is proposed for autonomous urban driving. In the proposed approach, an additional auxiliary force is introduced to enable safe overtaking maneuvers, maintain appropriate distances from surrounding vehicles, and dynamically adjust speed while turning, thereby improving the overall smoothness and stability of the vehicle's motion. To evaluate performance, the algorithm was tested in the CARLA simulation environment under two standard benchmark scenarios, CoRL2017 and NoCrash. Experimental results indicate that the proposed CAR-FMP method achieves, on average, a 6% higher success rate in the CoRL2017 scenario and a 12% increase in the NoCrash scenario with dense traffic, compared with typical learning-based approaches. Overall, the CAR-FMP framework, relying on distributed control and independent of pre-defined paths, provides an effective and scalable solution for safe and efficient navigation of autonomous vehicles in complex urban environments.

: 10.22034/ABMIR.2025.23796.1178

E-ISSN: 2821-2037 /The Author 2026. Published by Yazd University This is an open access article under the CC BY 4.0 License (https://creativecommons.org/licenses/by/4.0/).