





یادگیری تقویتی در کنترل
پروژه شماره ۲

**Information Theoretic MPC for Model-Based
Reinforcement Learning**

سیده ستاره خسروی
مبینا لشگری

زمستان ۱۴۰۳

فهرست

۱. مقدمه.....	۵
۱.۱. چالش‌های اصلی در یادگیری تقویتی مبتنی بر مدل:.....	۵
۱.۲. راهکار پیشنهادی:	۶
۱.۳. اهداف تحقیق:	۶
۲. کنترل پیش‌بینی مدل (Model Predictive Control - MPC)	۷
۲.۱. اصول کنترل پیش‌بینی مدل.....	۷
۲.۲. چالش‌های کنترل پیش‌بینی مدل	۸
۲.۳. کنترل انتگرال مسیر پیش‌بینی مدل (MPPI)	۸
۲.۴. کنترل پیش‌بینی مدل مبتنی بر شبکه عصبی.....	۱۰
۲.۵. جمع‌بندی.....	۱۰
۳. کنترل مبتنی بر نظریه اطلاعات	۱۰
۳.۱. فرموله‌سازی مسئله کنترل.....	۱۱
۳.۲. انرژی آزاد و کنترل بهینه.....	۱۲
۳.۳. حداقل‌سازی واگرایی کولبک-لیبلر (KL-Divergence)	۱۲
۳.۴. نمونه‌گیری اهمیت (Importance Sampling)	۱۳
۳.۵. قانون به‌روزرسانی کنترل.....	۱۳
۳.۶. جمع‌بندی	۱۳
۴. کنترل پیش‌بینی مدل با استفاده از شبکه‌های عصبی.....	۱۴
۴.۱. انگیزه استفاده از شبکه‌های عصبی در MPC	۱۴
۴.۲. فرمول‌بندی مسئله	۱۴
۴.۳. فرآیند آموزش شبکه عصبی	۱۵
۴.۴. پیاده‌سازی MPC با مدل شبکه عصبی.....	۱۶
۴.۵. الگوریتم پیشنهادی.....	۱۷
۴.۶. نتایج شبیه‌سازی.....	۱۸
۴.۷. جمع‌بندی	۱۸
۵. نتایج شبیه‌سازی.....	۱۸
۵.۱. وظیفه تاب دادن معکوس پاندول روی چرخ (Cart-Pole Swing-Up)	۱۸

۵.۲ وظیفه هدایت کوادروتور (Quadrotor Navigation)	۲۰
۵.۳ تحلیل خطای چندمرحله‌ای (Multi-Step Prediction Error)	۲۱
۵.۴ تأثیر تنظیمات شبکه عصبی بر عملکرد کنترل	۲۱
۵.۵ جمع‌بندی	۲۲
۶ نتایج تجربی	۲۲
۶.۱ پلتفرم آزمایشی	۲۲
۶.۲ جمع‌آوری داده‌های آموزشی	۲۳
۶.۳ پیاده‌سازی الگوریتم MPPI با شبکه عصبی	۲۳
۶.۴ نتایج آزمایش‌های رانندگی	۲۴
۶.۵ تحلیل خطای مدل شبکه عصبی	۲۵
۶.۶ جمع‌بندی	۲۵
۷ نتیجه‌گیری و پیشنهادات آینده	۲۶
۷.۱ جمع‌بندی یافته‌ها	۲۶
۷.۲ محدودیت‌ها و چالش‌ها	۲۷
۷.۳ پیشنهادات برای تحقیقات آینده	۲۸
۷.۴ نتیجه‌گیری نهایی	۲۹

۱. مقدمه

در حوزه رباتیک، بسیاری از وظایف را می‌توان در قالب مسائل یادگیری تقویتی مطرح کرد. در این چارچوب، هدف سیستم، بهینه‌سازی یک تابع هزینه از طریق داده‌های جمع‌آوری شده از تعاملات آن با محیط است. مسائل یادگیری تقویتی در سیستم‌های رباتیک اغلب دارای فضای حالت-کنش پیوسته و ابعاد بالا هستند که چالش‌های متعددی را از نظر پیچیدگی محاسباتی و تعمیم‌پذیری به همراه دارند.

به‌طور کلی، رویکردهای RL را می‌توان به دو دسته روش‌های بدون مدل (Model-Free) و روش‌های مبتنی بر مدل (Model-Based) تقسیم‌بندی کرد:

۱. روش‌های بدون مدل:

در این روش‌ها، مستقیماً از طریق تعامل با محیط و جمع‌آوری داده، سیاست بهینه‌سازی می‌شود. روش‌هایی مانند گرادیان سیاست (Policy Gradient) و Q-Learning از این دسته هستند. هرچند که این روش‌ها در وظایف پیچیده موفق عمل کرده‌اند، اما معمولاً نیازمند داده‌های زیاد و زمان آموزش طولانی هستند. همچنین، محدودیت دیگر این است که راه‌حل نهایی معمولاً به سیاست اولیه وابسته بوده و انعطاف‌پذیری کمی در کشف راهکارهای جدید دارند.

۲. روش‌های مبتنی بر مدل:

در مقابل، رویکردهای مبتنی بر مدل ابتدا یک مدل دینامیکی از محیط را یاد می‌گیرند و سپس از آن برای محاسبه سیاست بهینه استفاده می‌کنند. روش‌های متعددی در این حوزه پیشنهاد شده‌اند که شامل بهینه‌سازی مسیر (Trajectory Optimization) و مدل‌سازی اغتشاش (Disturbance Modeling) هستند. مزیت اصلی این روش‌ها در کاهش تعداد تعاملات واقعی با محیط و بهره‌گیری از دانش موجود از سیستم است. با این حال، یکی از چالش‌های اصلی این روش‌ها دقت مدل یادگیری شده و چگونگی مدیریت عدم قطعیت‌های مدل است.

۱.۱. چالش‌های اصلی در یادگیری تقویتی مبتنی بر مدل:

۱. تعمیم‌پذیری:

سیستم‌های رباتیکی که در محیط‌های متغیر و تصادفی فعالیت می‌کنند، نیاز به انطباق سریع با شرایط جدید دارند.

۲. پیچیدگی محاسباتی:

روش‌های MPC معمولی به دلیل استفاده از تکنیک‌های بهینه‌سازی مقید، نیاز به تقریب‌های درجه اول یا دوم دارند که می‌تواند کارایی را در سیستم‌های پیچیده محدود کند.

۱.۲. راهکار پیشنهادی:

برای مقابله با این چالش‌ها، کنترل پیش‌بینی مدل (MPC) به عنوان یکی از ابزارهای مؤثر شناخته می‌شود. MPC با انجام بهینه‌سازی بلادرنگ (Online Optimization) روی افق پیش‌بینی مشخص، امکان تعمیم بهتر را فراهم می‌کند. یکی از انواع پیشرفته MPC، کنترل انتگرال مسیر پیش‌بینی مدل (Model Predictive Path Integral - MPPI) است که از نمونه‌گیری‌های تصادفی برای بهینه‌سازی مسیر استفاده می‌کند و می‌تواند هزینه‌های پیچیده را بدون نیاز به محاسبه حل کند.

تحقیقات پیشین MPPI عمدتاً بر سیستم‌های دارای دینامیک کنترل-آفین متمرکز بودند که توانایی‌های آن را در کاربردهای عمومی محدود می‌کرد. در این مطالعه، روش MPPI گسترش یافته است تا برای سیستم‌های تصادفی با دینامیک‌های عمومی نیز قابل استفاده باشد.

رویکرد پیشنهادی بر مبنای چارچوب اطلاعاتی نظری توسعه یافته و بدون نیاز به فرضیات کنترل آفین، یک راهکار مبتنی بر داده را برای یادگیری مدل ارائه می‌دهد. به منظور مدل‌سازی دینامیک سیستم، از شبکه‌های عصبی چندلایه (Multi-Layer Neural Networks) استفاده شده است که امکان تقریب دقیق رفتارهای سیستم را فراهم می‌آورد.

۱.۳. اهداف تحقیق:

- ارائه یک الگوریتم MPC انعطاف‌پذیر که قابلیت مدیریت معیارهای هزینه غیرمحدب و پویایی‌های غیرخطی را داشته باشد.
- استفاده از مدل‌های یادگیری ماشین به‌ویژه شبکه‌های عصبی برای تقریب دینامیک سیستم.
- ارزیابی عملکرد الگوریتم روی وظایف شبیه‌سازی و سخت‌افزار واقعی شامل ربات‌های چرخ‌دار و کوادروتورها.

۲. کنترل پیش‌بینی مدل (Model Predictive Control - MPC)

کنترل پیش‌بینی مدل (MPC) یک روش بهینه‌سازی مبتنی بر مدل است که در آن دنباله‌ای از ورودی‌های کنترلی به صورت بازگشتی محاسبه و اجرا می‌شود. در هر مرحله، یک افق زمانی مشخص برای پیش‌بینی در نظر گرفته شده و بهترین مسیر کنترلی بر اساس حداقل‌سازی یک تابع هزینه بهینه می‌شود. این فرآیند با اجرای اولین ورودی از دنباله محاسبه‌شده و بازنگری مجدد مدل در مراحل بعدی ادامه می‌یابد.

۲.۱. اصول کنترل پیش‌بینی مدل

به طور کلی، فرآیند MPC شامل مراحل زیر است:

۱. مدل‌سازی سیستم:

یک مدل دینامیکی از سیستم در قالب معادلات حالت غیرخطی یا خطی در نظر گرفته می‌شود:

$$x_{t+1} = f(x_t, u_t) + w_t$$

در این رابطه:

- x_t بردار حالت سیستم در زمان t است.
- u_t بردار کنترل ورودی اعمال شده به سیستم است.
- w_t نویز سیستم با میانگین صفر و واریانس مشخص است.

۲. تعریف تابع هزینه:

یک تابع هزینه طراحی می‌شود که هدف آن به حداقل رساندن هزینه‌های مربوط به وضعیت‌های نهایی سیستم، مسیر طی‌شده، و انرژی مصرفی است:

$$J = \sum_{t=0}^{T-1} C(x_t, u_t) + \Phi(x_T)$$

که در آن:

- $C(x_t, u_t)$ هزینه لحظه‌ای بر اساس وضعیت و ورودی کنترل است.
- $\Phi(x_T)$ هزینه نهایی مربوط به وضعیت انتهایی سیستم است.

○ T افق پیش‌بینی است.

۳. بهینه‌سازی دنباله کنترلی:

بهینه‌سازی به صورت بلادرنگ انجام شده و ورودی کنترلی بهینه u_t^* محاسبه می‌شود.

۴. اجرای ورودی کنترلی:

اولین مقدار از دنباله کنترل به سیستم اعمال شده و سیستم به‌روزرسانی می‌شود.

۲.۲. چالش‌های کنترل پیش‌بینی مدل

علی‌رغم موفقیت‌های MPC در سیستم‌های صنعتی و فرآیندهای پیچیده، به‌کارگیری آن در رباتیک با چالش‌های متعددی روبروست، از جمله:

۱. پیچیدگی محاسباتی:

حل مسئله بهینه‌سازی در زمان واقعی (Real-Time) نیاز به قدرت محاسباتی بالایی دارد.

۲. مدل‌های غیرخطی:

سیستم‌های غیرخطی و دارای عدم قطعیت، نیازمند مدل‌سازی دقیق و رویکردهای خاص برای مدیریت خطاها هستند.

۳. محدودیت‌های افق پیش‌بینی:

با افزایش افق پیش‌بینی، هزینه محاسباتی افزایش یافته و نیاز به روش‌های عددی سریع‌تری وجود دارد.

۲.۳. کنترل انتگرال مسیر پیش‌بینی مدل (MPPI)

به‌عنوان جایگزینی برای روش‌های سنتی MPC، رویکرد کنترل انتگرال مسیر پیش‌بینی مدل (Model Predictive Path Integral - MPPI) مطرح شده است. MPPI یک روش مبتنی بر نمونه‌گیری تصادفی است که با استفاده از شبیه‌سازی‌های متعدد، توزیع احتمالاتی برای ورودی کنترلی ارائه می‌دهد.

ویژگی‌های کلیدی MPPI:

- توانایی بهینه‌سازی معیارهای هزینه غیرمحدب.
- امکان استفاده از روش‌های موازی‌سازی در سخت‌افزارهای GPU.
- انعطاف‌پذیری بالا در مواجهه با سیستم‌های غیرخطی.

روش کلی MPPI:

رویکرد MPPI مبتنی بر اصول نظریه اطلاعاتی است که در آن بهینه‌سازی از طریق حداقل‌سازی واگرایی کولبک-لیبلر (KL-Divergence) بین توزیع ورودی کنترلی اعمال‌شده و توزیع بهینه انجام می‌شود. روش کلی شامل مراحل زیر است:

۱. نمونه‌گیری از فضای ورودی:

مجموعه‌ای از نمونه‌های ورودی تصادفی V با توزیع گاوسی انتخاب می‌شوند.

۲. محاسبه وزن‌های اهمیت:

برای هر مسیر نمونه‌برداری‌شده، یک وزن اهمیت بر اساس تابع هزینه محاسبه می‌شود:

$$w(V) = \frac{1}{\eta} \exp \left(-\frac{1}{\lambda} S(V) \right)$$

که در آن:

- λ ضریب دمایی برای کنترل تأثیر نویز در فرآیند تصمیم‌گیری است.
- $S(V)$ تابع هزینه مسیر مربوط به دنباله کنترلی است.

۳. به‌روزرسانی دنباله کنترلی:

ورودی کنترل بهینه از میانگین وزن‌دار نمونه‌ها استخراج می‌شود:

$$u^* = \sum_{i=1}^N w_i v_i$$

۲.۴. کنترل پیش‌بینی مدل مبتنی بر شبکه عصبی

در این تحقیق، برای ارتقای عملکرد MPPI، از شبکه‌های عصبی چندلایه به‌عنوان مدل دینامیکی سیستم استفاده شده است. هدف از این کار، جایگزینی مدل‌های سنتی با یک مدل مبتنی بر داده است که می‌تواند ویژگی‌های پیچیده دینامیکی سیستم را به دقت تقریب بزند.

مزایای استفاده از شبکه‌های عصبی در MPC:

- افزایش دقت مدل: مدل‌های مبتنی بر داده توانایی تطبیق بهتر با شرایط پیچیده محیطی را دارند.
- کاهش وابستگی به مدل‌های تحلیلی: نیاز به مدل‌سازی فیزیکی دقیق کاهش می‌یابد.
- انعطاف‌پذیری بالا: توانایی یادگیری رفتارهای غیرخطی سیستم.

فرآیند یادگیری مدل دینامیکی:

۱. جمع‌آوری داده از تعاملات سیستم با محیط.
۲. آموزش شبکه عصبی با استفاده از روش‌های بهینه‌سازی مانند RMSProp.
۳. ارزیابی مدل یادگیری‌شده در وظایف کنترلی.

۲.۵. جمع‌بندی

در این بخش، مفاهیم اساسی کنترل پیش‌بینی مدل و چالش‌های آن بررسی شد. همچنین، روش MPPI به‌عنوان رویکردی کارآمد برای کنترل سیستم‌های غیرخطی ارائه شد و کاربرد شبکه‌های عصبی در بهبود مدل‌سازی دینامیک سیستم تشریح گردید. در بخش‌های بعدی، روش پیشنهادی با جزئیات بیشتری بررسی شده و نتایج تجربی ارائه می‌شود.

۳. کنترل مبتنی بر نظریه اطلاعات

در این بخش، مبانی نظری الگوریتم کنترل پیش‌بینی مدل مبتنی بر نمونه‌گیری (MPC) را که بر اساس اصول نظریه اطلاعات توسعه داده شده است، بررسی می‌کنیم. این رویکرد از دو مفهوم کلیدی در نظریه اطلاعات بهره می‌برد: واگرایی کولبک-لیبلر (KL-Divergence) و انرژی آزاد (Free Energy). استفاده از این مفاهیم منجر به

توسعه یک روش کنترل بهینه می‌شود که می‌تواند بدون نیاز به محاسبات گرادیانی سنگین و با قابلیت پردازش موازی، دنباله کنترلی مناسب را تعیین کند.

۳.۱. فرموله‌سازی مسئله کنترل

در نظر بگیریم که یک سیستم دینامیکی تصادفی در زمان گسسته با معادله زیر تعریف شود:

$$x_{t+1} = F(x_t, v_t)$$

که در آن:

- x_t بردار حالت سیستم در زمان t است.
- v_t متغیر کنترل تصادفی است که از یک توزیع نرمال با میانگین u_t و کواریانس Σ نمونه‌برداری می‌شود:

$$v_t \sim \mathcal{N}(u_t, \Sigma)$$

ورودی کنترلی به‌عنوان یک متغیر تصادفی با توزیع احتمال $P(V)$ در نظر گرفته می‌شود، که در آن:

$$V = \{v_0, v_1, \dots, v_{T-1}\}$$

در فرآیند کنترل، هدف این است که دنباله کنترلی بهینه $U = \{u_0, u_1, \dots, u_{T-1}\}$ را به گونه‌ای انتخاب کنیم که هزینه کل سیستم حداقل شود. هزینه مسیرهای نمونه‌برداری شده با استفاده از یک تابع هزینه به‌صورت زیر محاسبه می‌شود:

$$C(x_1, x_2, \dots, x_T) = \varphi(x_T) + \sum_{t=0}^{T-1} q(x_t)$$

که در آن:

- $\varphi(x_T)$ هزینه نهایی است.
- $q(x_t)$ هزینه لحظه‌ای در هر زمان t است.

۳.۲. انرژی آزاد و کنترل بهینه

در چارچوب نظریه اطلاعات، می توان هزینه کلی سیستم را بر اساس مفهوم انرژی آزاد به صورت زیر تعریف کرد:

$$F(V) = \lambda \log \mathbb{E}_P \left[\exp \left(-\frac{1}{\lambda} S(V) \right) \right]$$

که در آن:

- λ یک پارامتر دمایی است که حساسیت سیستم به هزینه ها را کنترل می کند.
- $S(V)$ مجموع هزینه مسیر است.
- $P(V)$ توزیع احتمالاتی مسیرها است.

به منظور ساده سازی محاسبات، انتظارگیری در معادله فوق با استفاده از توزیع احتمال دیگری که پارامتر کنترل را شامل می شود، انجام می گیرد:

$$F(V) \leq -\lambda \mathbb{E}_Q \left[\log \frac{Q(V)}{P(V)} + \frac{1}{\lambda} S(V) \right]$$

هدف کنترل کننده این است که با انتخاب یک توزیع کنترل مناسب $Q(V)$ ، مقدار انرژی آزاد را به حداقل برساند.

۳.۳. حداقل سازی واگرایی کولبک-لیبلر (KL-Divergence)

واگرایی کولبک-لیبلر بین توزیع کنترل $Q(V)$ و توزیع بهینه $Q^*(V)$ به صورت زیر تعریف می شود:

$$D_{KL}(Q^* \parallel Q) = \sum_V Q^*(V) \log \frac{Q^*(V)}{Q(V)}$$

برای حداقل سازی این واگرایی، باید دنباله کنترلی U به گونه ای به روزرسانی شود که این اختلاف کمینه شود. این منجر به یک قانون به روزرسانی برای کنترل می شود که به صورت زیر است:

$$u_t^{(i+1)} = u_t^{(i)} + \sum_{k=1}^N w_k \epsilon_t^{(k)}$$

که در آن:

- $\epsilon_t^{(k)}$ نویز کنترلی در نمونه k است.
- w_k وزن مربوط به هر مسیر است که با استفاده از تخمین‌های مونت کارلو محاسبه می‌شود.

۳.۴. نمونه‌گیری اهمیت (Importance Sampling)

از آنجایی که نمونه‌برداری مستقیم از توزیع بهینه امکان‌پذیر نیست، از تکنیک نمونه‌گیری اهمیت (Importance Sampling) برای تقریب آن استفاده می‌شود. در این روش، نمونه‌هایی از یک توزیع گوسی ساده گرفته شده و وزن آن‌ها بر اساس هزینه مسیر محاسبه می‌شود:

$$w(V) = \frac{1}{Z} \exp \left(-\frac{1}{\lambda} S(V) \right)$$

که در آن Z یک ثابت نرمال‌سازی است.

۳.۵. قانون به‌روزرسانی کنترلی

پس از محاسبه وزن‌های اهمیت، ورودی کنترلی بهینه از میانگین وزن‌دار نویزهای اعمال شده محاسبه می‌شود:

$$u_t = \sum_{k=1}^N w_k v_t^{(k)}$$

این فرآیند به‌صورت تکراری انجام می‌شود تا زمانی که مسیر کنترل بهینه پیدا شود.

۳.۶. جمع‌بندی

در این بخش، چارچوب اطلاعاتی نظری برای کنترل پیش‌بینی مدل ارائه شد که شامل روش‌های حداقل‌سازی واگرایی KL، محاسبه انرژی آزاد، و نمونه‌گیری اهمیت برای بهینه‌سازی مسیرهای کنترلی است. این رویکرد به ما این امکان را می‌دهد که سیاست‌های کنترلی پیچیده را بدون نیاز به گرادیان‌گیری‌های مستقیم محاسبه کنیم.

در بخش بعدی، پیاده‌سازی الگوریتم MPC با استفاده از شبکه‌های عصبی به‌عنوان مدل دینامیکی سیستم ارائه خواهد شد.

۴. کنترل پیش‌بینی مدل با استفاده از شبکه‌های عصبی

در این بخش، نحوه استفاده از شبکه‌های عصبی چندلایه (Multi-Layer Neural Networks) به‌عنوان مدل دینامیکی سیستم در کنترل پیش‌بینی مدل (MPC) بررسی می‌شود. هدف از این رویکرد، جایگزینی مدل‌های دینامیکی تحلیلی سنتی با مدل‌های یادگیری‌شده از داده است که قابلیت تطبیق بهتر با سیستم‌های پیچیده و غیرخطی را فراهم می‌آورد.

۴.۱. انگیزه استفاده از شبکه‌های عصبی در MPC

مدل‌های دینامیکی سنتی معمولاً مبتنی بر معادلات فیزیکی هستند که پارامترهای آن‌ها باید از طریق فرآیندهای شناسایی سیستم استخراج شوند. این فرآیند می‌تواند پیچیده و زمان‌بر باشد و دقت آن نیز به کیفیت داده‌های شناسایی بستگی دارد. در مقابل، مدل‌های یادگیری‌شده با استفاده از شبکه‌های عصبی، قابلیت یادگیری از داده‌های تجربی را دارند و می‌توانند رفتارهای پیچیده و غیرخطی سیستم را با دقت بیشتری تقریب بزنند.

مزایای استفاده از شبکه‌های عصبی در MPC عبارت‌اند از:

- **افزایش دقت مدل:** یادگیری مستقیم از داده‌های تجربی منجر به مدل‌سازی دقیق‌تر رفتارهای پیچیده سیستم می‌شود.
- **انعطاف‌پذیری بالا:** امکان تعمیم به شرایط عملیاتی مختلف بدون نیاز به تنظیم دستی پارامترهای فیزیکی.
- **قابلیت بهینه‌سازی سریع‌تر:** اجرای بهینه‌سازی بلادرنگ با بهره‌گیری از پردازش‌های موازی در سخت‌افزارهایی مانند GPU.

۴.۲. فرمول‌بندی مسئله

فرض می‌کنیم که دینامیک سیستم به‌صورت زیر تعریف شود:

$$x_{t+1} = f(x_t, u_t) + \varepsilon_t$$

که در آن:

- x_t بردار حالت سیستم در زمان t است.
 - u_t ورودی کنترلی سیستم است.
 - $f(x_t, u_t)$ یک تابع غیرخطی است که توسط یک شبکه عصبی تخمین زده می‌شود.
 - ε_t نویز مدل است که فرض می‌شود دارای توزیع نرمال با میانگین صفر باشد.
- شبکه عصبی با مجموعه داده‌های جمع‌آوری شده به صورت زیر آموزش داده می‌شود:

$$D = \{(x_t, u_t, x_{t+1})\}_{t=0}^N$$

هدف از آموزش شبکه، یافتن تابع تقریب $\hat{f}(x_t, u_t, \theta)$ است که با کمینه‌سازی تابع هزینه زیر بهینه می‌شود:

$$\mathcal{L}(\theta) = \sum_{t=0}^N \|x_{t+1} - \hat{f}(x_t, u_t, \theta)\|^2$$

که در آن:

- θ پارامترهای شبکه عصبی است.
- N تعداد نمونه‌های آموزشی است.
- $\mathcal{L}(\theta)$ تابع هزینه آموزش مدل است که اختلاف بین خروجی شبکه و مقادیر واقعی را کمینه می‌کند.

۴.۳. فرآیند آموزش شبکه عصبی

آموزش شبکه عصبی به صورت زیر انجام می‌شود:

۱. جمع‌آوری داده:

- از طریق اجرای اولیه سیستم با یک کنترلر ساده یا توسط یک اپراتور انسانی.
- شامل زوج‌های (حالت، ورودی، خروجی) که رفتار سیستم را نشان می‌دهند.

۲. پیش‌پردازش داده:

- نرمال‌سازی داده‌ها برای تسریع یادگیری.
- حذف نویزهای غیرضروری از داده‌های جمع‌آوری شده.

۳. انتخاب معماری شبکه:

- استفاده از شبکه‌های عصبی کاملاً متصل (Fully Connected) با دو لایه مخفی.
- استفاده از توابع فعال‌سازی مانند تابع تانژانت هایپربولیک که برای تخمین سیستم‌های دینامیکی مناسب است.

۴. آموزش با روش‌های بهینه‌سازی:

- استفاده از روش گرادیان نزولی با بهینه‌ساز RMSProp.
- استفاده از تکنیک‌هایی مانند dropout و regularization برای جلوگیری از بیش‌برازش.

۴.۴. پیاده‌سازی MPC با مدل شبکه عصبی

پس از آموزش شبکه عصبی، مدل یادگیری شده برای پیش‌بینی دینامیک سیستم در چارچوب MPC به کار گرفته می‌شود. فرآیند پیاده‌سازی به شرح زیر است:

۱. اجرای MPPI:

الگوریتم کنترل انتگرال مسیر پیش‌بینی مدل (MPPI) برای نمونه‌گیری از مسیرهای کنترلی بهینه با استفاده از مدل شبکه عصبی اجرا می‌شود.

۲. پیش‌بینی دینامیک:

در هر گام، شبکه عصبی مقدار حالت بعدی x_{t+1} را بر اساس ورودی u_t و حالت فعلی x_t پیش‌بینی می‌کند.

۳. انتخاب کنترل بهینه:

مسیرهایی که کمترین هزینه را دارند، انتخاب شده و اولین ورودی از آن‌ها به سیستم اعمال می‌شود.

۴.۵. الگوریتم پیشنهادی

الگوریتم کنترل پیشنهادی در چارچوب MPPI با استفاده از مدل شبکه عصبی به صورت زیر پیاده سازی می شود:

الگوریتم ۱: MPPI با شبکه عصبی

۱. ورودی:

- مدل دینامیکی f^A
- افق پیش بینی T
- تعداد نمونه گیری ها K
- دنباله کنترل اولیه U

۲. تکرار:

- برای هر نمونه k در بازه K :

۱. تولید مسیر کنترلی تصادفی.

۲. پیش بینی مسیر با مدل شبکه عصبی.

۳. محاسبه هزینه مسیر.

۴. تعیین وزن های اهمیت.

۳. به روز رسانی:

- محاسبه ورودی کنترل بهینه با میانگین گیری وزن دار مسیرها.

۴. اجرا:

- اعمال اولین ورودی از دنباله کنترل به سیستم.

۴.۶. نتایج شبیه‌سازی

مدل‌های شبکه عصبی مختلف با اندازه‌های متفاوت برای ارزیابی عملکرد آزمایش شده‌اند. شبکه‌های کوچک‌تر با ۱۶ و ۳۲ نورون در هر لایه نتایج بهتری در مقایسه با شبکه‌های پیچیده‌تر نشان داده‌اند. همچنین، آزمایش‌ها نشان می‌دهند که استفاده از داده‌های بوت‌استرپ (Bootstrap Data) باعث بهبود سرعت همگرایی و دقت کنترل می‌شود.

۴.۷. جمع‌بندی

در این بخش، نحوه استفاده از شبکه‌های عصبی برای مدل‌سازی دینامیک سیستم و پیاده‌سازی کنترل پیش‌بینی مدل بررسی شد. نتایج اولیه نشان می‌دهند که استفاده از این مدل‌ها در کنار الگوریتم MPPI منجر به عملکرد مناسبی در محیط‌های شبیه‌سازی و سخت‌افزار واقعی می‌شود. در بخش‌های بعدی، آزمایش‌های تجربی و نتایج مربوط به اجرای الگوریتم روی سیستم‌های فیزیکی ارائه خواهد شد.

۵. نتایج شبیه‌سازی

در این بخش، نتایج حاصل از آزمایش الگوریتم کنترل پیش‌بینی مدل (MPC) مبتنی بر شبکه عصبی در محیط‌های شبیه‌سازی بررسی می‌شود. آزمایش‌ها بر روی دو وظیفه کنترلی پیچیده شامل «تاب دادن معکوس پاندول روی چرخ (Cart-Pole Swing-Up)» و «هدایت کوادروتور در یک محیط با موانع (Quadrotor Navigation)» انجام شده است. هدف از این آزمایش‌ها ارزیابی دقت، کارایی و قابلیت تعمیم الگوریتم در مواجهه با شرایط متغیر است.

۵.۱. وظیفه تاب دادن معکوس پاندول روی چرخ (Cart-Pole Swing-Up)

هدف آزمایش:

در این وظیفه، سیستم باید پاندول را که در ابتدا در وضعیت رو به پایین قرار دارد، به حالت عمودی برساند و آن را در آن وضعیت نگه دارد. کنترل باید از طریق اعمال نیروهای افقی به واگن صورت گیرد، در حالی که تعادل پاندول حفظ می‌شود.

مدل سازی سیستم:

معادلات دینامیکی سیستم به صورت زیر تعریف شده است:

$$\dot{x} = v$$
$$\dot{v} = \frac{F}{m} - \frac{g}{l} \sin(\theta)$$

که در آن:

- x موقعیت افقی واگن،
- v سرعت واگن،
- F نیروی کنترل،
- m جرم پاندول،
- l طول پاندول،
- g شتاب گرانش.

تابع هزینه:

تابع هزینه برای بهینه سازی در این آزمایش به صورت زیر در نظر گرفته شده است:

$$C(x) = 10x^2 + 500(\cos(\theta) + 1)^2 + v^2 + 15\dot{\theta}^2$$

این تابع هزینه تلاش می کند تا موقعیت افقی واگن را در محدوده ای مشخص نگه دارد و زاویه پاندول را در وضعیت عمودی تثبیت کند.

نتایج:

- آزمایش ها نشان دادند که استفاده از مدل شبکه عصبی با ابعاد کوچک (۱۶ نورون در هر لایه) در مقایسه با شبکه های بزرگ تر، عملکرد بهتری در کاهش خطا داشت.
- تعداد دفعات موفقیت در رسیدن به تعادل، در شبکه های آموزش دیده با داده های بوت استرپ به طور قابل توجهی افزایش یافت.

- مقایسه هزینه نرمال سازی شده مسیرهای کنترل نشان داد که عملکرد الگوریتم پیشنهادی به مدل فیزیکی واقعی نزدیک است.

۵.۲. وظیفه هدایت کوادروتور (Quadrotor Navigation)

هدف آزمایش:

در این وظیفه، یک کوادروتور باید از یک نقطه مشخص به نقطه مقصد حرکت کند و از برخورد با موانع اجتناب کند. این آزمایش توانایی کنترل الگوریتم را در شرایط پیچیده و محیط های دارای موانع بررسی می کند.

مدل سازی سیستم:

معادلات دینامیکی کوادروتور به صورت زیر در نظر گرفته شده است:

$$\ddot{x} = \frac{T}{m} (\sin(\phi) \cos(\theta) \cos(\psi) + \sin(\psi) \sin(\theta))$$

که در آن:

- x, y, z مختصات موقعیت کوادروتور،
- T نیروی پیشران،
- M جرم کوادروتور،
- ϕ, θ, ψ زوایای اوایلر.

تابع هزینه:

تابع هزینه برای این وظیفه شامل دو بخش است:

$$q(x) = (x - x_d)^T Q (x - x_d) + 100000C$$

که در آن:

- x_d موقعیت مطلوب،
- C جریمه برای برخورد با موانع.

نتایج:

- استفاده از مدل‌های شبکه عصبی با اندازه متوسط (۳۲ نرون در هر لایه) بهترین عملکرد را در جلوگیری از برخورد با موانع نشان داد.
- در مرحله آموزش، استفاده از داده‌های بوت‌استرپ باعث تسریع همگرایی شبکه و کاهش تعداد برخوردها شد.
- اجرای الگوریتم با تعداد بیشتری از نمونه‌های کنترلی منجر به مسیرهای بهینه‌تری شد، اما زمان محاسباتی را افزایش داد.

۵.۳. تحلیل خطای چندمرحله‌ای (Multi-Step Prediction Error)

در روش پیشنهادی، شبکه عصبی برای پیش‌بینی یک مرحله‌ای آموزش دیده است، اما در کنترل MPC، پیش‌بینی‌های چندمرحله‌ای نیز مورد نیاز است. این موضوع ممکن است به رشد خطای تجمعی منجر شود.

نتایج تحلیل خطا:

- خطای پیش‌بینی چندمرحله‌ای در سیستم پاندول نشان داد که مدل‌های کوچک‌تر دقت بیشتری دارند و خطای تجمعی را بهتر کنترل می‌کنند.
- برای کوادروتور، افزایش خطا با افزایش افق پیش‌بینی مشهود بود؛ با این حال، در اجرای کنترل، خطاها به خوبی توسط مکانیسم‌های تصحیح MPC جبران شدند.
- خطای میانگین مربعات (RMSE) برای مدل‌های مختلف ارزیابی شد و نتایج نشان دادند که مدل‌های یادگیری شده توانایی تخمین دقیق دینامیک سیستم را دارند.

۵.۴. تأثیر تنظیمات شبکه عصبی بر عملکرد کنترل

به منظور بررسی تأثیر اندازه شبکه عصبی بر عملکرد کنترل، پیکربندی‌های مختلفی از شبکه با اندازه‌های متفاوت آزمایش شدند. نتایج نشان دادند که:

- مدل‌های کوچک‌تر دقت بالاتری در پیش‌بینی دینامیک سیستم داشتند.

- مدل‌های بزرگ‌تر نیاز به داده‌های آموزشی بیشتری داشتند اما در نهایت دقت مشابهی ارائه دادند.
- استفاده از توابع فعال‌سازی مانند Tanh نسبت به ReLU نتایج بهتری در مدل‌های دینامیکی ارائه داد.

۵.۵. جمع‌بندی

در این بخش، عملکرد الگوریتم پیشنهادی در دو وظیفه کنترلی بررسی شد. نتایج نشان داد که:

۱. استفاده از مدل‌های شبکه عصبی به‌عنوان جایگزین مدل‌های تحلیلی می‌تواند منجر به کنترل مؤثر و بلادرنگ شود.

۲. دقت مدل به اندازه شبکه و مقدار داده‌های آموزشی بستگی دارد.

۳. روش MPPI در ترکیب با مدل‌های یادگیری‌شده عملکردی نزدیک به مدل‌های فیزیکی ایده‌آل ارائه می‌دهد.

در بخش بعدی، آزمایش‌های تجربی روی سخت‌افزار واقعی مورد بررسی قرار خواهد گرفت.

۶. نتایج تجربی

در این بخش، عملکرد الگوریتم پیشنهادی بر روی سخت‌افزار واقعی بررسی شده است. هدف از این آزمایش‌ها ارزیابی دقت، کارایی و قابلیت اجرایی الگوریتم کنترل پیش‌بینی مدل (MPC) مبتنی بر شبکه عصبی در یک سیستم فیزیکی واقعی است. آزمایش‌ها بر روی پلتفرم خودروبی AutoRally در یک مسیر مسابقه‌ای انجام شده است.

۶.۱. پلتفرم آزمایشی

آزمایش‌های تجربی بر روی خودروی خودران Georgia Tech AutoRally انجام شده است. این خودرو یک پلتفرم تحقیقاتی است که برای آزمایش الگوریتم‌های کنترل حرکتی در محیط‌های خارج از جاده طراحی شده است. مشخصات این خودرو به شرح زیر است:

- ابعاد: مقیاس یک‌پنجم خودروهای واقعی

- سرعت حداکثری: ۱۳ متر بر ثانیه (حدود ۴۷ کیلومتر بر ساعت)
- سنسورها: شامل GPS ، IMU، و دوربین‌های استریو
- پردازنده: NVIDIA GTX 750 Ti با ۶۴۰ هسته CUDA برای انجام محاسبات کنترلی بلادرنگ

۶.۲. جمع‌آوری داده‌های آموزشی

برای آموزش مدل شبکه عصبی، مجموعه داده‌ای شامل رفتارهای خودرو در مسیر جمع‌آوری شد. این داده‌ها شامل پنج نوع مانور رانندگی مختلف بودند:

۱. رانندگی در سرعت‌های پایین (۴ تا ۶ متر بر ثانیه): برای ثبت رفتارهای عادی در شرایط کنترل‌شده.
 ۲. مانورهای زیگزاگ: جهت بررسی پاسخ دینامیکی خودرو به تغییرات ناگهانی مسیر.
 ۳. شتاب‌گیری و ترمزگیری خطی: برای یادگیری تأثیر نیروی پیشران بر سرعت و شتاب خودرو.
 ۴. مانورهای لغزش: برای مدل‌سازی رفتار خودرو در شرایط کاهش اصطکاک.
 ۵. رانندگی با سرعت بالا: برای ثبت داده‌های عملکردی در حداکثر توان خودرو.
- هر نوع مانور به مدت ۳ دقیقه در هر جهت (ساعت‌گرد و پادساعت‌گرد) اجرا شد و در مجموع ۳۰ دقیقه داده جمع‌آوری گردید.

۶.۳. پیاده‌سازی الگوریتم MPPI با شبکه عصبی

پس از آموزش مدل شبکه عصبی، الگوریتم کنترل انتگرال مسیر پیش‌بینی مدل (MPPI) در زمان واقعی اجرا شد. پارامترهای کنترل برای این آزمایش به شرح زیر تنظیم شدند:

- افق پیش‌بینی: ۲.۵ ثانیه
- فرکانس کنترلی: ۴۰ هرتز
- تعداد نمونه‌گیری‌ها: ۱۲۰۰ مسیر کنترلی در هر گام

مراحل پیاده‌سازی:

۱. پیش‌بینی مسیر:

الگوریتم با استفاده از مدل شبکه عصبی، مسیرهای کنترلی مختلف را شبیه‌سازی کرده و هزینه هر مسیر را محاسبه می‌کند.

۲. انتخاب مسیر بهینه:

بر اساس معیار حداقل‌سازی هزینه، دنباله کنترل مناسب انتخاب و اولین ورودی به خودرو اعمال می‌شود.

۳. بازبینی مدل:

داده‌های جدید از حسگرها دریافت شده و برای اصلاح مدل مورد استفاده قرار می‌گیرند.

۶.۴. نتایج آزمایش‌های رانندگی

سناریوهای آزمایشی:

آزمایش‌ها در مسیر مسابقه‌ای به ابعاد ۳۰ متر اجرا شدند. چهار سناریوی مختلف شامل رانندگی در شرایط عادی، رانندگی با مانع، رانندگی در سرعت‌های بالا، و رانندگی در شرایط لغزنده بررسی شدند.

تحلیل نتایج:

۱. دقت مسیر:

نتایج نشان دادند که الگوریتم پیشنهادی توانسته است با دقت بالا مسیر را دنبال کند. زمان‌های ثبت‌شده برای آزمایش‌های مختلف نشان‌دهنده توانایی کنترل در حفظ مسیر بوده است.

۲. زمان اجرای دورها:

مقایسه زمان اجرای دورهای کامل در شرایط مختلف نشان داد که با افزایش سرعت، زمان‌های دور به طور متوسط ۶٪ کاهش یافت. این کاهش بیانگر بهبود عملکرد کنترل در بهینه‌سازی مسیرها است. جدول زیر عملکرد را در تکرارهای مختلف نشان می‌دهد:

	Avg. Lap (s)	Best Lap (s)	Top Speed (m/s)	Max. Slip
10 m/s	10.34	9.93	8.05	38.68
11 m/s	9.97	9.43	8.71	34.65
12 m/s	9.88	9.47	8.63	43.72
13 m/s	9.74	9.36	8.44	48.70

۳. کنترل در شرایط لغزنده:

کنترلر توانست در مسیرهایی که شامل سطوح لغزنده بودند، بدون انحراف قابل توجه حرکت کند. با این حال، میزان لغزش در سرعت‌های بالاتر افزایش یافت و در سرعت ۱۳ متر بر ثانیه به حداکثر مقدار ۴۸.۷۰ رسید. این نشان‌دهنده نیاز به تنظیمات دقیق‌تر برای کنترل لغزش در سرعت‌های بالا است.

۴. مقایسه با کنترل مبتنی بر مدل فیزیکی:

عملکرد الگوریتم پیشنهادی با استفاده از شبکه عصبی در مقایسه با مدل فیزیکی کلاسیک، عملکردی مشابه ارائه داد. دقت مسیر و زمان اجرای دوره‌ها نشان داد که کنترلر یادگیری‌شده می‌تواند به‌خوبی جایگزین روش‌های مبتنی بر مدل تحلیلی شود، بدون تفاوت قابل توجه در عملکرد.

۶.۵. تحلیل خطای مدل شبکه عصبی

برای ارزیابی دقت مدل یادگیری‌شده، خطای پیش‌بینی مدل در برابر مقادیر واقعی ثبت‌شده در آزمایش‌ها مورد بررسی قرار گرفت. نتایج تحلیل خطا نشان دادند که:

- خطای میانگین مربعات (RMSE) در پیش‌بینی موقعیت در بازه ۲ ثانیه‌ای حدود ۰.۱۵ متر بود.
- خطای زاویه‌ای در حدود ۰.۳ رادیان مشاهده شد که برای کنترل خودرو در سرعت‌های بالا قابل قبول بود.
- تأثیر نویز سنسورها به‌ویژه در سرعت‌های بالا محسوس بود، اما کنترل تطبیقی MPPI این نویزها را تا حد زیادی جبران کرد.

۶.۶. جمع‌بندی

در این بخش، نتایج حاصل از اجرای الگوریتم پیشنهادی بر روی خودروی AutoRally ارائه شد. نتایج آزمایش‌های تجربی نشان می‌دهند که:

۱. الگوریتم MPPI مبتنی بر شبکه عصبی عملکردی نزدیک به مدل‌های فیزیکی داشته و توانایی اجرای کنترل پیچیده را در شرایط مختلف داراست.

۲. **زمان محاسباتی مناسب:** با استفاده از پردازنده‌های گرافیکی، اجرای الگوریتم در زمان واقعی با ۱۲۰۰ نمونه در هر گام کنترلی امکان پذیر شده است.

۳. **افزایش دقت و پایداری در شرایط مختلف:** الگوریتم توانسته است رفتارهای پیچیده‌ای مانند دریافت کنترل شده و کنترل دقیق مسیر را اجرا کند.

در بخش بعدی، نتایج به دست آمده مورد تحلیل قرار گرفته و پیشنهاداتی برای بهبود آینده ارائه خواهد شد.

۷. نتیجه گیری و پیشنهادات آینده

در این بخش، نتایج کلی تحقیق مورد بررسی قرار گرفته و پیشنهاداتی برای بهبود و توسعه آینده ارائه می شود. الگوریتم پیشنهادی کنترل پیش بینی مدل (MPC) مبتنی بر شبکه عصبی و کنترل انتگرال مسیر پیش بینی مدل (MPPI) عملکرد مناسبی را در وظایف شبیه سازی و سخت افزار واقعی نشان داده است. این مطالعه نشان می دهد که استفاده از مدل های مبتنی بر داده می تواند به عنوان جایگزینی قدرتمند برای مدل های تحلیلی در سیستم های کنترلی پیچیده به کار گرفته شود.

۷.۱. جمع بندی یافته ها

نتایج تحقیق نشان دادند که:

۱. کارایی روش MPPI با شبکه عصبی:

- ترکیب MPC با شبکه عصبی منجر به کاهش نیاز به مدل سازی فیزیکی دقیق و افزایش انعطاف پذیری سیستم شد.
- عملکرد کنترلر در هر دو وظیفه شبیه سازی (پاندول روی چرخ و کوادروتور) قابل مقایسه با مدل های دقیق فیزیکی بود.
- خودرو در محیط های پیچیده با مسیرهای ناهموار و شرایط لغزنده قادر به کنترل دقیق و حفظ مسیر بود.

۲. دقت مدل یادگیری شده:

- مدل شبکه عصبی توانست دینامیک سیستم را با دقت قابل قبولی تخمین بزند.
- در وظیفه رانندگی، خطای RMS در پیش‌بینی مسیر کمتر از ۱۰ سانتی‌متر بود.
- مدل شبکه عصبی نسبت به اغتشاشات محیطی مانند نویز سنسورها انعطاف‌پذیری خوبی از خود نشان داد.

۳. قابلیت اجرایی در زمان واقعی:

- استفاده از GPU برای نمونه‌گیری موازی MPPI امکان اجرای کنترل در فرکانس ۴۰ هرتز را فراهم کرد.
- تعداد نمونه‌های استفاده‌شده (۱۲۰۰ نمونه) بهبود قابل توجهی در دقت کنترل ایجاد کرد.

۴. مقایسه با روش‌های کنترلی دیگر:

- کنترل مبتنی بر MPPI با مدل‌های یادگیری‌شده عملکردی نزدیک به روش‌های مبتنی بر مدل فیزیکی داشت.
- MPPI توانایی کشف مسیرهای کنترل بهینه را حتی در حضور نویزهای سیستم بهبود بخشید.

۷.۲. محدودیت‌ها و چالش‌ها

با وجود نتایج مثبت، روش پیشنهادی با برخی چالش‌ها و محدودیت‌های زیر روبرو بود:

۱. افزایش هزینه محاسباتی:

استفاده از شبکه‌های عصبی بزرگ برای مدل‌سازی دینامیک، هزینه‌های محاسباتی بالایی به همراه دارد. بهینه‌سازی ساختار شبکه و کاهش پیچیدگی مدل می‌تواند تأثیر مثبتی بر زمان محاسبات داشته باشد.

۲. دقت مدل در افق‌های زمانی بلندمدت:

مدل شبکه عصبی به دلیل آموزش بر اساس پیش‌بینی تک‌مرحله‌ای، ممکن است در پیش‌بینی چندمرحله‌ای دچار افزایش خطا شود. راهکارهایی مانند تنظیم دوره‌ای مدل و استفاده از روش‌های یادگیری توزیع احتمالی (Bayesian Neural Networks) می‌توانند این چالش را کاهش دهند.

۳. محدودیت در تعمیم به شرایط جدید:

شبکه عصبی بر اساس داده‌های آموزشی موجود آموزش دیده است و ممکن است در شرایط جدید یا غیرمنتظره نیاز به داده‌های جدید برای به‌روزرسانی مدل داشته باشد.

۴. نیاز به داده‌های اولیه:

فرآیند بوت‌استرپ اولیه داده‌های زیادی نیاز دارد که در محیط‌های عملی ممکن است زمان‌بر باشد.

۷.۳. پیشنهادات برای تحقیقات آینده

با توجه به یافته‌های تحقیق، پیشنهادهای زیر برای بهبود و توسعه روش ارائه می‌شوند:

۱. بهینه‌سازی ساختار شبکه عصبی:

استفاده از روش‌های یادگیری انتقالی (Transfer Learning) و شبکه‌های عصبی مبتنی بر توجه (Attention Mechanisms) می‌تواند باعث افزایش کارایی مدل در وظایف مختلف شود.

۲. ادغام با روش‌های کنترل تقویتی:

ترکیب روش MPPI با یادگیری تقویتی می‌تواند منجر به تولید سیاست‌های کنترلی تطبیقی‌تر شود که با شرایط مختلف محیطی سازگار شوند.

۳. افزایش استحکام کنترل:

استفاده از مدل‌های مبتنی بر عدم قطعیت، مانند روش‌های یادگیری احتمالاتی و مدل‌های گوسی، می‌تواند منجر به افزایش پایداری سیستم در شرایط نویزی شود.

۴. افزایش بهره‌وری محاسباتی:

استفاده از تکنیک‌های بهینه‌سازی مانند کوانتیزاسیون مدل‌های عصبی و محاسبات پراکنده می‌تواند هزینه محاسباتی را کاهش دهد و امکان اجرای کنترل در سیستم‌های با منابع محدودتر را فراهم کند.

۵. آزمایش در شرایط دنیای واقعی:

انجام آزمایش‌های بیشتری در محیط‌های ناهموار، مانند رانندگی در مسیرهای خارج از جاده و در شرایط آب‌وهوایی مختلف، می‌تواند قابلیت تعمیم روش را افزایش دهد.

۶. استفاده از فیلترهای پیش‌بینی برای اصلاح مدل:

ادغام روش‌هایی مانند فیلتر کالمن یا فیلتر ذره‌ای می‌تواند به اصلاح پیش‌بینی‌های مدل شبکه عصبی در زمان اجرا کمک کند.

۷.۴. نتیجه‌گیری نهایی

در این مطالعه، یک الگوریتم کنترل پیش‌بینی مدل بر اساس شبکه‌های عصبی و رویکرد اطلاعاتی MPPI ارائه شد. نتایج به‌دست‌آمده نشان می‌دهند که این روش می‌تواند به‌طور مؤثر برای کنترل سیستم‌های پیچیده رباتیکی در شرایط شبیه‌سازی و عملیاتی به کار گرفته شود.

نکات کلیدی تحقیق:

- کنترل مبتنی بر MPPI با مدل شبکه عصبی می‌تواند به عنوان جایگزینی مؤثر برای روش‌های مبتنی بر مدل‌های تحلیلی مورد استفاده قرار گیرد.
 - قابلیت اجرای کنترل در زمان واقعی با نرخ نمونه‌گیری بالا محقق شده است.
 - روش پیشنهادی می‌تواند در مواجهه با شرایط متغیر محیطی، عملکرد مناسبی از خود نشان دهد.
- نتایج این تحقیق گام مهمی در جهت استفاده از مدل‌های یادگیری ماشین برای کنترل سیستم‌های رباتیکی پیچیده محسوب می‌شود و می‌تواند در کاربردهای مختلفی مانند رباتیک خودران، خودروهای بدون سرنشین و سیستم‌های صنعتی هوشمند به کار رود.