دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی برق

کنترل و عیب‌یابی هوشمند

پیاده‌سازی روش‌های هوشمند برای کنترل و عیب‌یابی سیستم تعلیق خودروی پیشرفته

نگارش

علی نظامی وند چگینی

استاد

دکتر فرزانه عبدالهی

بهار 1404

چکیده

مقدمه

سیستم‌های تعلیق نقش مهمی در پایداری، راحتی سواری و عملکرد هندلینگ خودرو دارند .**سیستم تعلیق چهارگانه (Quad Car Suspension)** یکی از بخش‌های مهم در مهندسی خودرو است که بررسی روش‌های مختلف آن می‌تواند به بهبود دینامیک کلی وسیله نقلیه کمک کند.

در این گزارشکار، مدل‌های مختلفی برای شناسایی و کنترل این سیستم مورد بررسی و آزمایش قرار می‌گیرند. روش‌های متنوعی از جمله مدل‌سازی ریاضی، شبیه‌سازی و مقایسه عملکرد سیستم‌های تعلیق مختلف ارزیابی می‌شوند تا تأثیر هر روش بر کیفیت سواری، پایداری و بازدهی سیستم تحلیل گردد.

هدف این گزارشکار، ارائه یک تحلیل جامع از روش‌های مختلف سیستم تعلیق و بررسی تأثیر آن‌ها در شرایط مختلف جاده‌ای و بارگذاری است.

واژه‌های کلیدی:

کنترل، عیب یابی، سیستم های هوشمند، شبکه های عصبی، سیستم تعلیق هوشمند خودرو

فهرست مطالب

صفحه

فصل اول: مقدمه و مدل‌سازی سیستم..... 1
معرفی سیستم..... 2
مدل‌سازی سیستم..... 3
معادلات سیستم:..... 4
1-1-1- پارامترهای سیستم..... 5
فصل دوم: کنترل مبتنی بر شبکه MLP..... 7
معرفی روش MLP..... 8
طراحی شبکه MLP..... 12
ساختار شبکه عصبی پرسپترون چندلایه (MLP) در این پیاده‌سازی..... 14
نتایج شبیه سازی..... 17
فصل سوم: کنترل مبتنی بر شبکه RBF..... 20
معرفی شبکه RBF..... 21
ساختار کنترلر RBF..... 21
ساختار شناسایی کننده RBF..... 24
سیستم حلقه باز..... 26
سیستم حلقه بسته..... 30
نتایج شبیه سازی..... 31
انجام آزمایش به ازای مقادیر مختلف sampling rate..... 33
فصل چهارم: کنترل مبتنی بر شبکه LSTM..... 34
معرفی شبکه LSTM..... 35
ساختار شبکه LSTM..... 35
طراحی مدل و پیاده سازی مدل غیر خطی..... 38
تولید مجموعه داده برای آموزش مدل در سیستم تعلیق فعال خودرو..... 43
مدل پیشنهادی برای شناسایی و رویت سیستم..... 46
نتیجه گیری کلی..... 48
فصل پنجم: کنترل مبتنی بر RL..... 51
معرفی روش کنترل مبتنی بر RL..... 52
معرفی الگوریتم SAC..... 53
توضیح بخش های مختلف برنامه..... 54

آموزش شبکه..... 65
نتایج آموزش..... 68
نتیجه گیری کلی..... 77

فصل ششم: جمع‌بندی و نتیجه‌گیری..... 79
نتیجه گیری روش های مختلف..... 80
MLP (Perceptron چندلایه)..... 80
RBF (شبکه عصبی شعاعی)..... 80
LSTM (شبکه‌های عصبی بازگشتی طولانی‌مدت کوتاه‌مدت)..... 80
SAC (الگوریتم یادگیری تقویتی Soft Actor-Critic)..... 81
مقایسه روش‌ها..... 82
نتیجه‌گیری کلی..... 82

منابع و مراجع..... 83

فهرست اشکال

صفحه

شکل 1-1 تصویر کلی از سیستم تعلیق خودرو با 4 درجه آزادی 2
شکل 1-2 تصویر کلی از مدل سازی سیستم تعلیق خودرو با 4 درجه آزادی 3
شکل 1-3 مدل سازی سیستم تعلیق پیشرفته 4
شکل 2-1 ساختار کلی شبکه 12
شکل 2-2 خروجی شبکه 17
شکل 2-3 خروجی شبکه به ازای مقادیر مختلف 18
شکل 3-1 ساختار کلی کنترل کننده RBF 22
شکل 3-2 ساختار شناسایی کننده RBF 25
شکل3-3 خروجی سیستم حلقه باز 27
شکل3-4 پاسخ حالات سیستم به ورودی پله 28
شکل3-5 خروجی سیستم نسبت به ورودی سینوسی 30
شکل3-6 نتایج شبیه سازی حلقه بسته 31
شکل 3-7 تغییرات وزن شناسایی کننده و کنترل کننده 32
شکل3-8 نتایج شبیه سازی با نرخ نمونه برداری 0.01 ثانیه 33
شکل4-1 ساختار شبکه عصبی LSTM 37
شکل 4-2 مدلسازی سیستم غیر خطی 38
شکل 4-3 نتایج مدلسازی سیستم (حالات سیستم) 39
شکل 4-4 نتایج مدلسازی سیستم (سیگنال کنترلی) 41
شکل 4-5 خروجی دیتا سازی 44
شکل 4-6 ورودی کنترلی 45
شکل 4-7 نتایج آموزش مدل 47

شکل 5-1 بررسی پاداش و حرکت در طی آموزش 68
شکل 5-2 میزان هزینه شبکه 71
شکل 5-3 میانگین critic loss 72
شکل 5-4 میزان احتمالات اکتور 73
شکل 5-5 میزان تغییر alpha temperature 73
شکل 5-6 تغییرات critic loss 75
شکل 5-7 مانیتور فرکانس هر حالت 76

فهرست جداول

صفحه

جدول 1-1- پارامترهای مورد نیاز برای مدلسازی کوادروتور 5
جدول 2-2- مقایه MLP با روش های کلاسیک 11
جدول 6-1 مقایسه روش های مختلف 82

فصل اول:

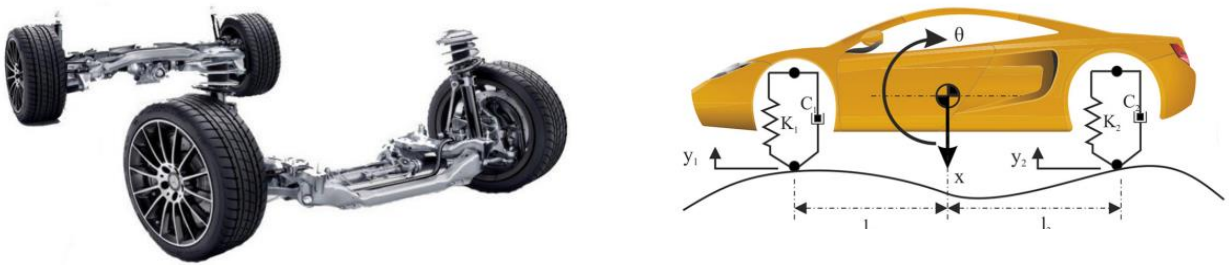
مقدمه و مدل‌سازی سیستم

معرفی سیستم

سیستم تعلیق خودرو یکی از مهم‌ترین بخش‌های آن محسوب می‌شود که وظیفه اصلی آن، جذب ضربات و نوسانات ناشی از ناهمواری‌های جاده و حفظ پایداری و کنترل خودرو در شرایط مختلف رانندگی است. سیستم تعلیق چهارگانه (Quad Car Suspension) یک ساختار پیشرفته است که از چهار مجموعه فنر و دمپر مستقل یا نیمه‌مستقل تشکیل شده و به منظور بهبود کیفیت سواری، کنترل دینامیکی و افزایش ایمنی خودرو طراحی شده است.

این سیستم علاوه بر کاهش لرزش‌ها و ارتعاشات، نقش مهمی در توزیع بار بین چرخ‌ها، بهینه‌سازی چسبندگی تایرها به سطح جاده و بهبود پاسخگویی خودرو به تغییرات مسیر دارد. طراحی و عملکرد آن می‌تواند بسته به نوع خودرو، کاربری آن و شرایط محیطی متفاوت باشد.

یکی از مهم‌ترین ویژگی‌های سیستم تعلیق، رفتار غیرخطی آن است. این غیرخطی بودن ناشی از عواملی مانند سختی متغیر فنرها، رفتار وابسته به سرعت و جابجایی در دمپرها، اصطکاک بین قطعات متحرک، تغییرات زاویه‌ای و وابستگی نیروهای تایر به شرایط جاده است. این پیچیدگی‌ها باعث می‌شوند که مدل‌سازی و کنترل این سیستم چالش‌برانگیز باشد و نیاز به استفاده از روش‌های پیشرفته شناسایی و کنترل داشته باشد.



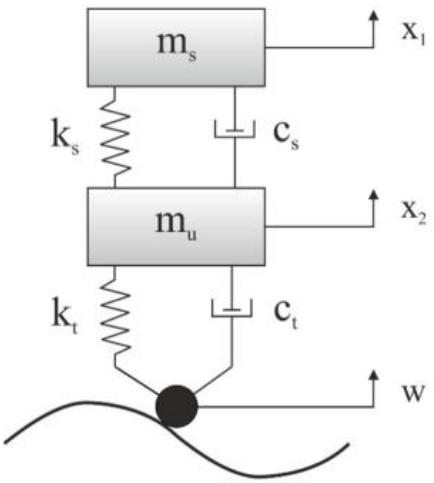
شکل 1-1 تصویر کلی از سیستم تعلیق خودرو با 4 درجه آزادی

مدل‌سازی سیستم

سیستم تعلیق چهارگانه خودرو یک مدل دینامیکی با چهار درجه آزادی دارد که شامل جرم فنربندی‌شده، جرم غیرفنربندی‌شده، فنر و دمپرهای سیستم تعلیق و تأثیر تایرها است. در این مدل، سختی تایر به صورت غیرخطی درجه دوم و سختی فنرهای تعلیق به صورت غیرخطی درجه سوم در نظر گرفته شده است.

معادلات دینامیکی سیستم بر اساس اصل دالامبر استخراج شده‌اند و نیروهای اعمال‌شده بر جرم‌های سیستم، شامل سختی و میرایی تعلیق و تایر، در نظر گرفته شده‌اند. این معادلات برای تحلیل و شبیه‌سازی رفتار سیستم در شرایط مختلف پیاده‌سازی شده‌اند.

با توجه به رفتار غیرخطی سیستم، مدل‌سازی به گونه‌ای انجام شده که بتواند تأثیرات ناشی از تغییرات سختی فنر و تایر را به‌طور دقیق شبیه‌سازی کند. همچنین، از روش‌های بهینه‌سازی برای تنظیم بهینه پارامترهای سیستم تعلیق و بهبود عملکرد آن در زمینه کاهش ارتعاشات و بهبود راحتی سرنشین استفاده شده است.

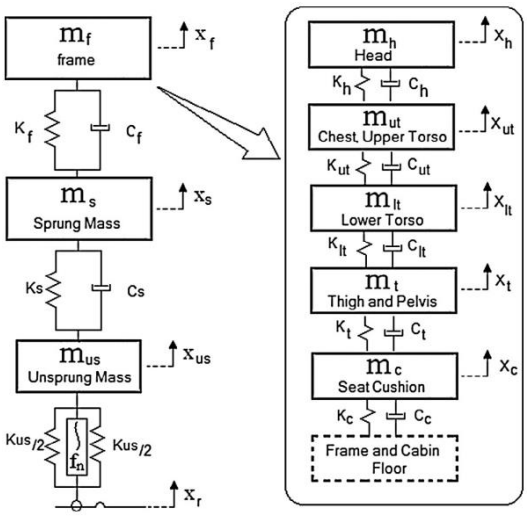


شکل 1-2 تصویر کلی از مدل سازی سیستم تعلیق خودرو با 4 درجه آزادی

معادلات سیستم:

به طور کلی معادلات غیر خطی سیستم به شکل زیر می باشد.

$$m_{us}\ddot{x}_{us} = -k_t(x_{us} - x_r) + k_s(x_s - x_{us}) + c_s(\dot{x}_s - \dot{x}_{us}) + k_{tnl}(x_{us} - x_r)^2 + k_{snl}(x_s - x_{us})^3$$
$$m_s\ddot{x}_s = -k_s(x_s - x_{us}) - c_s(\dot{x}_s - \dot{x}_{us}) - k_{snl}(x_s - x_{us})^3 + k_f(x_f - x_s) + c_f(\dot{x}_f - \dot{x}_s)$$
$$m_f\ddot{x}_f = -k_f(x_f - x_s) - c_f(\dot{x}_f - \dot{x}_s) + k_c(x_c - x_f) + c_c(\dot{x}_c - \dot{x}_f)$$
$$m_c\ddot{x}_c = -k_c(x_c - x_f) - c_c(\dot{x}_c - \dot{x}_f) + k_{tp}(x_t - x_c) + c_{tp}(\dot{x}_t - \dot{x}_c)$$
$$m_t\ddot{x}_t = -k_{tp}(x_t - x_c) - c_{tp}(\dot{x}_t - \dot{x}_c) + k_{lt}(x_{lt} - x_t) + c_{lt}(\dot{x}_{lt} - \dot{x}_t)$$
$$m_{lt}\ddot{x}_{lt} = -k_{lt}(x_{lt} - x_t) - c_{lt}(\dot{x}_{lt} - \dot{x}_t) + k_{ut}(x_{ut} - x_{lt}) + c_{ut}(\dot{x}_{ut} - \dot{x}_{lt})$$
$$m_{ut}\ddot{x}_{ut} = -k_{ut}(x_{ut} - x_{lt}) - c_{ut}(\dot{x}_{ut} - \dot{x}_{lt}) + k_h(x_h - x_{ut}) + c_h(\dot{x}_h - \dot{x}_{ut})$$
$$m_h\ddot{x}_h = -k_h(x_h - x_{ut}) - c_h(\dot{x}_h - \dot{x}_{ut})$$



شکل 1-3 مدل سازی سیستم تعلیق پیشرفته

پارامترهای سیستم

پارامترهای مورد نیاز جهت مدل‌سازی این سیستم تعلیق به صورت زیر می‌باشد.

جدول 1-1- پارامترهای مورد نیاز برای مدلسازی کوادروتور

Symbol	Description	Unit
A	System matrix	-
AR_h	Amplitude ratio of head RMS acceleration to seat RMS acceleration	-
AR_{ut}	Amplitude ratio of upper torso RMS acceleration to seat RMS acceleration	-
A_{wh}	Frequency-weighted RMS head acceleration	m/s^2
A_{w_spr}	Frequency-weighted RMS sprung mass acceleration	m/s^2
a_{wh}	Frequency-weighted head acceleration	m/s^2
c	Damping coefficient	$N \cdot s/m$
c_{lt}	Lumbar spine damping	$N \cdot s/m$
c_{ut}	Thoracic spine damping	$N \cdot s/m$
c_h	Cervical spine damping	$N \cdot s/m$
f_{obj}	Objective function	-
k	Stiffness	N/m
k_{lt}	Lumbar spine stiffness	N/m
k_{ut}	Thoracic spine stiffness	N/m
k_h	Cervical spine stiffness	N/m
k_{snl}	Nonlinear spring stiffness	N/m^3
k_t	Tire stiffness	N/m
k_{tnl}	Nonlinear tire stiffness	N/m^2
m	Mass	kg
VDV_h	Vibration dose value at the head	$m/s^{1.75}$
x_r	Road profile	m
$x, \dot{x}, \ddot{x}$	Displacement, velocity, and acceleration	$m, m/s, m/s^2$

- اندیس ها

Subscript	Meaning
s	Sprung mass
us	Unsprung mass
f	Frame
c	Seat cushion
tp	Thigh and pelvis
lt	Lower torso (lumbar spine)
ut	Upper torso
h	Head

- مقدار دهی مقادیر:

Parameter	Value	Parameter	Value	Parameter	Value
m_h	5.31	c_h	400	k_h	310,000
m_{ut}	28.49	c_{ut}	4750	k_h	183,000
m_{lt}	8.62	c_{lt}	4585	k_{lt}	162,800
m_t	12.78	c_t	2064	k_t	90,000
m_c	1	c_c	200	k_c	18,000
m_f	15	c_f	830	k_f	31,000
m_s	290	c_s	700	k_s	23,500
m_{us}	40	k_t	190,000	k_{tnl}	$1.5k_t$
		k_{snl}	$100k_s$		

فصل دوم:

کنترل مبتنی بر شبکه MLP

معرفی روش MLP

سیستم تعلیق چهارگانه خودرو یک سیستم پیچیده و غیرخطی است که طراحی کنترل‌کننده‌های کارآمد برای آن نیازمند درک دقیق دینامیک سیستم و واکنش آن نسبت به ورودی‌های مختلف است. از آنجا که مدل‌سازی تحلیلی این سیستم به دلیل وجود نامعینی‌ها، عوامل غیرخطی و تغییرات دینامیکی دشوار است، روش‌های یادگیری ماشین مانند **پرسپترون چندلایه Multi-Layer - MLP) Perceptron)** گزینه‌ای مناسب برای شناسایی و کنترل این سیستم محسوب می‌شوند.

1) شبکه عصبی MLP و ویژگی‌های آن

MLP یکی از پرکاربردترین شبکه‌های عصبی مصنوعی (ANN) است که شامل چندین لایه متصل به هم شامل لایه ورودی، یک یا چند لایه پنهان و لایه خروجی می‌باشد. ویژگی کلیدی این شبکه، توانایی یادگیری نگاشت‌های غیرخطی پیچیده بین متغیرهای ورودی و خروجی است. هر نورون در لایه‌های پنهان دارای یک تابع فعال‌سازی (مانند **سیگموید، تانژانت هیپربولیک یا ReLU**) است که باعث افزایش قدرت یادگیری شبکه در مسائل پیچیده می‌شود.

مزایای استفاده از MLP برای سیستم تعلیق چهارگانه خودرو

- ✓ قابلیت شناسایی رفتارهای غیرخطی سیستم
- ✓ توانایی تعمیم به ورودی‌های جدید و مقابله با نامعینی‌ها
- ✓ امکان استفاده در طراحی کنترل‌کننده‌های تطبیقی و مقاوم
- ✓ کاهش نیاز به مدل‌سازی دقیق تحلیلی

2) شناسایی سیستم تعلیق خودرو با استفاده از MLP

مرحله 1: جمع‌آوری داده‌ها

در این مرحله، مجموعه‌ای از داده‌های تجربی یا شبیه‌سازی‌شده از رفتار سیستم تحت ورودی‌های مختلف جمع‌آوری می‌شود. این داده‌ها شامل:

- **ورودی‌ها :**نیروی جاده، دست‌اندازها، سرعت خودرو و نیروهای تعلیق
- **خروجی‌ها :**جابه‌جایی چرخ، شتاب سرنشین، نیروی واکنش تعلیق

مرحله 2: آموزش شبکه MLP

- **ساختار شبکه :**تعداد نرون‌ها و لایه‌های پنهان با استفاده از روش‌های بهینه‌سازی مانند **آزمایش و خطا (Trial and Error)** یا **جستجوی شبکه‌ای (Grid Search)** انتخاب می‌شوند.
- **الگوریتم آموزش :**از الگوریتم‌های مبتنی بر **پس‌انتشار خطا (Backpropagation)** مانند **بهینه‌سازی گرادیان نزولی (SGD)** یا **آدام (Adam)** برای تنظیم وزن‌های شبکه استفاده می‌شود.
- **داده‌های آموزشی و تست :**داده‌ها به دو بخش آموزشی و تست تقسیم شده و مدل روی داده‌های آموزشی یاد می‌گیرد، سپس روی داده‌های تست ارزیابی می‌شود.

مرحله 3: ارزیابی مدل شناسایی

پس از آموزش، مدل باید بتواند رفتار سیستم تعلیق را به‌طور دقیق شبیه‌سازی کند. برای ارزیابی عملکرد آن از معیارهایی مانند:

- **میانگین مربع خطا(MSE)**
- **ضریب همبستگی(R^2)**

استفاده می‌شود. در صورت عدم دقت کافی، ساختار شبکه یا پارامترهای یادگیری تنظیم می‌شوند.

3) طراحی کنترل‌کننده مبتنی بر MLP برای سیستم تعلیق خودرو

کنترل‌کننده‌های مبتنی بر MLP می‌توانند به دو روش پیاده‌سازی شوند:

الف) کنترل مستقیم(Direct Control)

در این روش، شبکه عصبی مستقیماً یک نگاشت بین ورودی‌های سیستم (مانند دست‌انداز جاده و نیروهای تعلیق) و سیگنال‌های کنترلی (مانند گشتاور یا نیروی اعمالی به کمک‌فنرها) یاد می‌گیرد. این روش نیازمند حجم زیادی از داده‌های آموزشی برای یادگیری یک سیاست کنترلی کارآمد است.

ب) کنترل غیرمستقیم(Indirect Control)

در این رویکرد، ابتدا یک مدل دینامیکی از سیستم با استفاده از شبکه MLP به‌دست می‌آید، سپس از این مدل برای طراحی یک کنترل‌کننده مناسب (مانند **کنترل فیدبک تطبیقی** یا **کنترل پیش‌بین مدل ((MPC)** استفاده می‌شود.

الگوریتم‌های کنترل رایج در کنار MLP

✓**کنترل پیش‌بین مدل :(MPC)** از مدل آموزش‌دیده‌شده برای پیش‌بینی رفتار آینده سیستم و بهینه‌سازی سیگنال‌های کنترلی استفاده می‌شود.

✓**کنترل تطبیقی :**از شبکه MLP برای به‌روزرسانی پارامترهای کنترلی در شرایط نامشخص استفاده می‌شود.

✓**کنترل مقاوم :(Robust Control)** ترکیب MLP با روش‌های مقاوم برای کاهش حساسیت به تغییرات پارامترهای سیستم.

4) مقایسه MLP با روش‌های کلاسیک کنترل

جدول 2-2- مقایه MLP با روش های کلاسیک

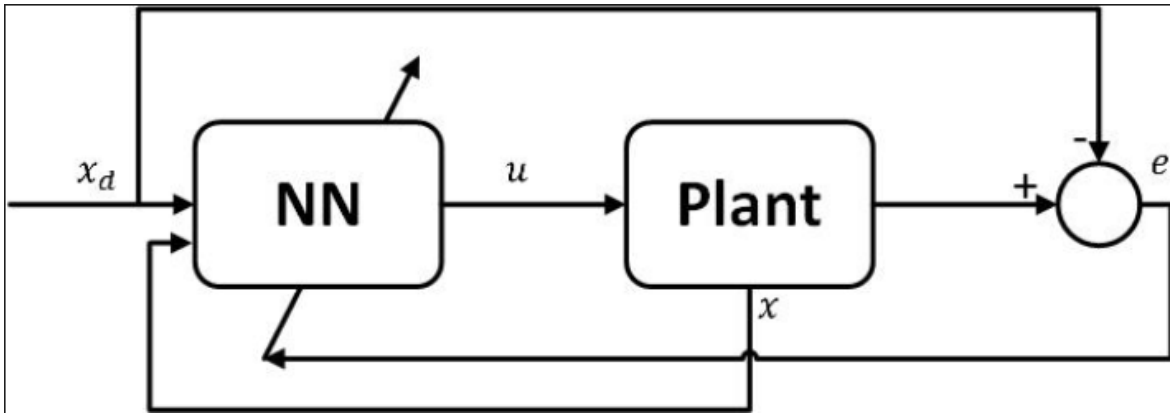
ویژگی	کنترل کلاسیک (PID, LQR)	کنترل مبتنی بر MLP
نیاز به مدل ریاضی	بله، نیاز به مدل دقیق دارد	خیر، می‌تواند بدون مدل تحلیلی یاد بگیرد
قابلیت انطباق	ضعیف در برابر تغییرات سیستم	قوی، می‌تواند به‌صورت آنلاین به‌روزرسانی شود
شناسایی سیستم	مبتنی بر معادلات دینامیکی	مبتنی بر داده‌های تجربی
پیچیدگی محاسباتی	کم، مناسب برای پیاده‌سازی سریع	بالا، نیازمند توان پردازشی بیشتر
عملکرد در سیستم‌های غیرخطی	ضعیف یا نیازمند خطی‌سازی	قوی، بدون نیاز به خطی‌سازی

5) چالش‌های استفاده از MLP در کنترل و شناسایی سیستم تعلیق

- ✓ نیاز به داده‌های آموزشی گسترده برای دستیابی به دقت بالا
- ✓ پیچیدگی در انتخاب معماری بهینه شبکه و تنظیم پارامترهای یادگیری
- ✓ هزینه محاسباتی بالاتر نسبت به روش‌های کلاسیک

طراحی شبکه MLP

ساختار کلی که ما در این پروژه داریم به شکل زیر می باشد.



شکل 2-1 ساختار کلی شبکه

در این پیاده‌سازی، یک شبکه عصبی پرسپترون چندلایه (MLP) طراحی شده است که می‌تواند در مسائل شناسایی و کنترل سیستم‌های مختلف مورد استفاده قرار گیرد. این نوع شبکه عصبی از مجموعه‌ای از نورون‌ها در چندین لایه تشکیل شده است که با استفاده از ارتباطات وزنی بین آن‌ها، فرآیند یادگیری را انجام می‌دهد. ساختار شبکه شامل لایه ورودی، چندین لایه پنهان و یک لایه خروجی است که هرکدام نقش مهمی در پردازش داده‌ها و یادگیری الگوهای مختلف ایفا می‌کنند.

این مدل بر اساس اصول یادگیری ماشین و روش‌های مبتنی بر شبکه‌های عصبی طراحی شده است. در این روش، ورودی‌های شبکه پس از عبور از لایه‌های مختلف، با استفاده از توابع فعال‌سازی پردازش شده و در نهایت خروجی مناسب تولید می‌شود. توابع فعال‌سازی نقش مهمی در تعیین پاسخ شبکه به ورودی‌های مختلف دارند و به آن امکان می‌دهند تا روابط پیچیده بین داده‌ها را بیاموزد. انتخاب تابع فعال‌سازی مناسب تأثیر بسزایی در عملکرد شبکه دارد و می‌تواند رفتار غیرخطی سیستم را به خوبی مدل کند.

در فرآیند یادگیری، ابتدا وزن‌ها و بایاس‌های شبکه به‌صورت مقداردهی اولیه تنظیم می‌شوند. سپس از طریق فرآیند انتشار رو به جلو، مقدار خروجی شبکه محاسبه شده و با مقدار هدف مقایسه می‌شود. اختلاف بین خروجی واقعی و مقدار مطلوب، خطا را تشکیل می‌دهد که باید در فرآیند یادگیری کاهش یابد. برای این منظور، روش پس‌انتشار خطا استفاده می‌شود که از طریق محاسبه گرادیان‌ها، وزن‌های شبکه را به‌روزرسانی کرده و باعث بهبود عملکرد شبکه در تطبیق با داده‌های ورودی می‌شود.

این روش یادگیری، مبتنی بر بهینه‌سازی وزن‌ها از طریق الگوریتم‌های گرادیان نزولی انجام می‌شود. نرخ یادگیری در این فرآیند نقش مهمی ایفا می‌کند و تعیین مقدار مناسب آن می‌تواند تأثیر قابل توجهی بر سرعت همگرایی و دقت نهایی مدل داشته باشد. در این پیاده‌سازی، پس از محاسبه گرادیان‌ها، وزن‌ها و بایاس‌های شبکه با اعمال تغییرات کوچک به سمت مقادیر بهینه هدایت می‌شوند. این فرآیند به‌صورت تکراری انجام شده و تا زمانی که شبکه به دقت مطلوبی برسد ادامه می‌یابد.

شبکه‌های عصبی پرسپترون چندلایه به دلیل توانایی در یادگیری الگوهای پیچیده و مدل‌سازی روابط غیرخطی، کاربردهای گسترده‌ای در زمینه‌های مختلف دارند. از جمله کاربردهای آن می‌توان به شناسایی سیستم‌های غیرخطی، کنترل تطبیقی، پردازش سیگنال، تشخیص الگو، و پیش‌بینی داده‌ها اشاره کرد. این مدل می‌تواند در سیستم‌هایی که دارای رفتارهای پیچیده و غیرخطی هستند، به‌عنوان یک ابزار کارآمد برای تحلیل و کنترل مورد استفاده قرار گیرد.

در این پیاده‌سازی، ساختار شبکه به گونه‌ای طراحی شده است که انعطاف‌پذیری بالایی در تنظیم تعداد لایه‌ها و نورون‌های هر لایه دارد. این ویژگی امکان تنظیم معماری شبکه بر اساس نیازهای خاص هر مسئله را فراهم می‌کند. همچنین از روش‌های مختلف فعال‌سازی برای بهبود عملکرد شبکه در شرایط مختلف استفاده شده است.

در نظر داریم که روابط استفاده شده در این پیاده سازی به شکل زیر می باشد:

$$\dot{x} = A_c x + f(x) + g(x)u, \qquad where\ f(x) = f_0(x) - A_c x$$

$$e = x - x_d, \qquad J = \frac{1}{2}(e^T e) + u^T R u$$

$$\frac{\partial J}{\partial e} = e$$

$$considering\ static\ approximation: \dot{x} = 0 \rightarrow \frac{\partial e}{\partial u} = -A_c^{-1} g(x) \approx -A_c^{-1} I_{n\times 1}$$

$$\frac{\partial J}{\partial u} = \frac{\partial J}{\partial e}\frac{\partial e}{\partial u} + \frac{\partial (u^T R u)}{\partial e} = -e A_c^{-1} I_{n\times 1} + u^T R$$

ساختار شبکه عصبی پرسپترون چندلایه (MLP) در این پیاده‌سازی

شبکه عصبی پرسپترون چندلایه (MLP) طراحی‌شده در این کد، برای کنترل یک سیستم تعلیق طراحی شده است. این شبکه دارای یک ساختار استاندارد با لایه ورودی، دو لایه پنهان و یک لایه خروجی است. هدف از این طراحی، استفاده از یک کنترل‌کننده عصبی برای کاهش نوسانات سیستم تعلیق و بهبود پایداری آن است.

۱ .معماری کلی شبکه

این شبکه از چندین لایه تشکیل شده است که هر یک دارای نقش مشخصی در فرآیند یادگیری و کنترل دارند:

- **لایه ورودی :**شامل **۴ نورون** است که نمایانگر وضعیت سیستم تعلیق می‌باشد. این ورودی‌ها شامل:
  - جابه‌جایی جرم اصلی(x_1)
  - جابه‌جایی چرخ(x_2)

- سرعت جرم اصلی(v_1)
- سرعت چرخ(v_2)

این مقادیر وضعیت لحظه‌ای سیستم را نشان می‌دهند و ورودی شبکه عصبی را تشکیل می‌دهند.

- لایه‌های پنهان :دو لایه پنهان با ۳۲ نورون در هر لایه در نظر گرفته شده‌اند. این لایه‌ها وظیفه استخراج ویژگی‌های پیچیده از ورودی‌ها را بر عهده دارند. در این لایه‌ها از تابع فعال‌سازی ReLU استفاده شده است که یک انتخاب مناسب برای مدل‌سازی روابط غیرخطی در سیستم‌های دینامیکی است.
- لایه خروجی :این لایه دارای ۱ نورون است که مقدار سیگنال کنترلی (u) را تولید می‌کند. این مقدار، نیرویی است که توسط سیستم کنترل به سیستم تعلیق اعمال می‌شود تا نوسانات را کاهش دهد. برای این لایه، تابع فعال‌سازی خطی (Linear) استفاده شده است که مقدار کنترلی را بدون محدودیت مستقیماً ارائه می‌دهد.

۲ .مکانیسم یادگیری و بهینه‌سازی

- **انتشار رو به جلو:(Forward Propagation)**

ورودی‌ها از طریق وزن‌ها و بایاس‌های لایه‌های مختلف عبور کرده و با استفاده از توابع فعال‌سازی پردازش می‌شوند تا خروجی نهایی) نیروی کنترلی (u تولید شود.

- **محاسبه خطا:**

خروجی شبکه با مقدار مطلوب مقایسه شده و خطا محاسبه می‌شود. در اینجا، مقدار مطلوب برابر با صفر در نظر گرفته شده است که نشان‌دهنده هدف کاهش نوسانات سیستم تعلیق است.

- **پس‌انتشار خطا:(Backpropagation)**

خطا از طریق الگوریتم پس‌انتشار به لایه‌های قبلی منتقل شده و وزن‌های شبکه بر اساس آن اصلاح می‌شوند.

- **به‌روزرسانی وزن‌ها:**

از طریق روش **گرادیان نزولی**، وزن‌ها و بایاس‌ها با استفاده از نرخ یادگیری مشخص، تنظیم می‌شوند تا شبکه به مرور زمان پاسخ بهتری ارائه دهد. در این پیاده‌سازی، نرخ یادگیری برابر با **۰.۱** در نظر گرفته شده و تعداد مراحل به‌روزرسانی در هر گام یادگیری **۵۰ بار** است که به همگرایی سریع‌تر مدل کمک می‌کند.

۳ .تعامل شبکه عصبی با سیستم تعلیق

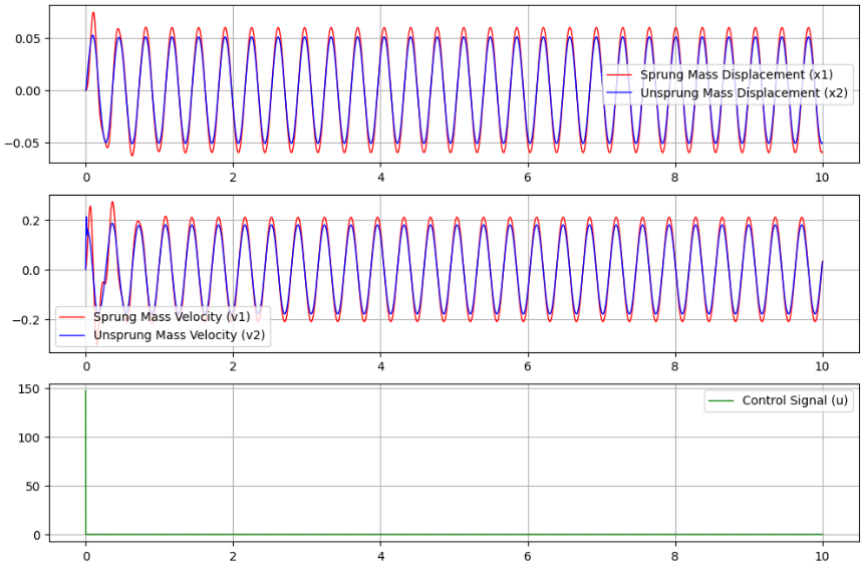
- در هر لحظه، وضعیت سیستم تعلیق اندازه‌گیری شده و به‌عنوان ورودی به شبکه عصبی داده می‌شود.
- شبکه عصبی مقدار نیروی کنترلی (u) را پیش‌بینی کرده و به سیستم تعلیق اعمال می‌کند.
- سپس، سیستم تعلیق با این نیروی جدید، وضعیت خود را به‌روزرسانی کرده و مقدار جدید جابه‌جایی و سرعت را ارائه می‌دهد.
- این فرآیند به‌صورت تکراری اجرا شده و شبکه به‌مرور عملکرد خود را بهینه می‌کند.

۴ .کاربردها و مزایای این شبکه

- **توانایی یادگیری روابط پیچیده و غیرخطی :**سیستم تعلیق دارای رفتار غیرخطی است و استفاده از یک کنترل‌کننده عصبی به‌جای روش‌های کلاسیک می‌تواند عملکرد بهتری ارائه دهد.
- **بهبود عملکرد سیستم تعلیق :**این شبکه باعث کاهش نوسانات و بهبود پایداری خودرو در مواجهه با ناهمواری‌های جاده‌ای می‌شود.
- **انعطاف‌پذیری بالا :**معماری شبکه می‌تواند به‌راحتی تغییر کند تا برای سیستم‌های دینامیکی مختلف بهینه شود.

نتایج شبیه سازی

با توجه به اینکه تعداد دور بروز رسانی وزن ها را برابر 50 و زمان نمونه برداری را 0.01 ثانیه و نرخ یادگیری را 0.1 در نظر گرفتیم خروجی شبکه به شکل زیر شده است.



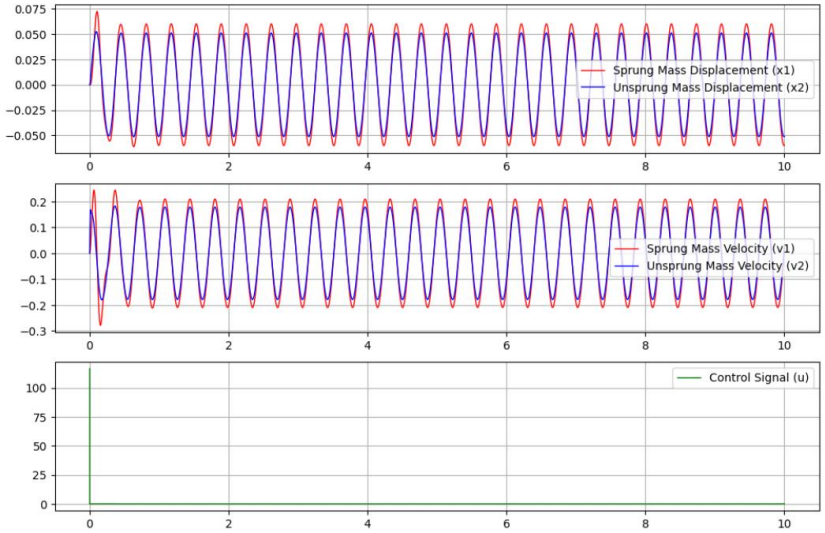
شکل 2-2 خروجی شبکه

همانظور که مشاهده میکنید با خطای بسیار کمی سیگنال مرجع دنبال شده است.

تاثیر هایپر پارامتر ها بر نتیجه tracking :

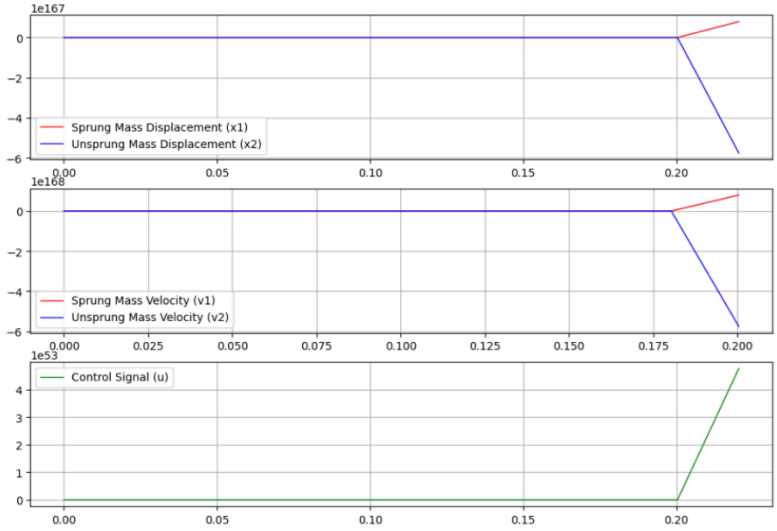
Lr = 0.01

Td = 0.001



Lr = 0.01

Td = 0.1



شکل 2-3 خروجی شبکه به ازای مقادیر مختلف

همانطور که مشاهده میکنید با افزایش sampling rate دقت افزایش می یابد اما حجم محاسبات شبکه به شدت بالا می رود اما از طرفی با کاهش این مقدار شبکه ممکن است حتی رو به ناپایداری برود .

لذا در این بخش ما مقادیر اولیه را به عنوان بهینه ترین حالت در نظر میگیریم.

فصل سوم:

کنترل مبتنی بر شبکه RBF

معرفی شبکه RBF

در این بخش، از شبکه عصبی پایه شعاعی **(RBF)** به‌عنوان یک ابزار قدرتمند برای شناسایی و کنترل سیستم استفاده شده است. این شبکه با بهره‌گیری از توابع پایه شعاعی، قادر است الگوهای پیچیده و غیرخطی را به‌خوبی مدل‌سازی کند. در فرآیند شناسایی سیستم، شبکه RBF ورودی‌ها و خروجی‌های سیستم را دریافت کرده و با یادگیری روابط میان آن‌ها، مدلی تقریبی از رفتار سیستم ارائه می‌دهد. این ویژگی به‌ویژه در سیستم‌های دینامیکی غیرخطی که مدل‌سازی ریاضی آن‌ها دشوار است، اهمیت بالایی دارد.

در زمینه کنترل، شبکه RBF به‌عنوان یک کنترل‌کننده هوشمند به کار گرفته می‌شود که با استفاده از داده‌های به‌دست‌آمده از شناسایی سیستم، ورودی‌های کنترلی مناسبی تولید می‌کند تا خروجی سیستم به مقدار مطلوب نزدیک شود. این روش نه‌تنها باعث افزایش دقت کنترل می‌شود، بلکه توانایی سازگاری با تغییرات دینامیکی سیستم را نیز دارد. به دلیل ساختار ساده و توانایی بالای شبکه‌های RBF در تقریب توابع پیچیده، این روش کنترل و شناسایی می‌تواند عملکرد بهتری نسبت به روش‌های کلاسیک ارائه دهد و در بسیاری از کاربردهای مهندسی مورد استفاده قرار گیرد.

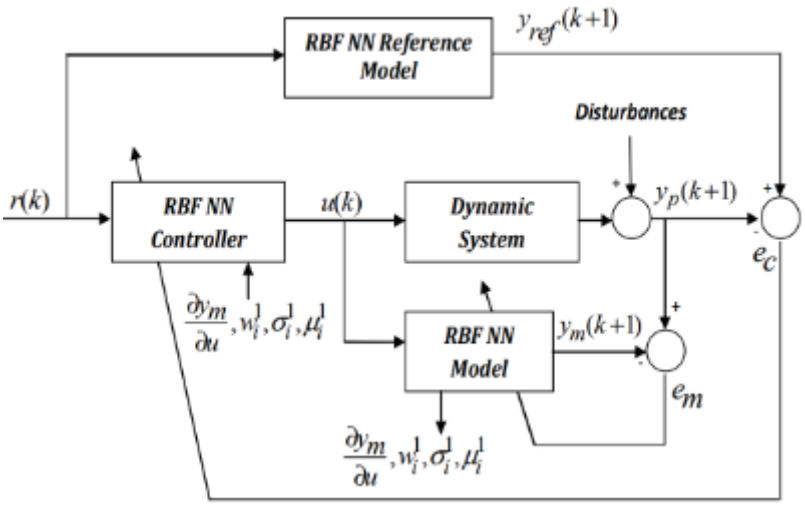
ساختار کنترلر RBF

کنترل‌کننده مبتنی بر **شبکه عصبی پایه شعاعی (RBF)** یکی از روش‌های قدرتمند در کنترل سیستم‌های دینامیکی است که از قابلیت‌های یادگیری شبکه‌های عصبی برای تخمین و تنظیم ورودی‌های کنترلی استفاده می‌کند. این کنترل‌کننده با استفاده از توابع پایه شعاعی، داده‌های ورودی را به یک فضای ویژگی تبدیل کرده و از طریق یک ساختار ساده اما مؤثر، ورودی کنترلی مناسب را تولید می‌کند.

در طراحی این کنترل‌کننده، لایه پنهان شامل تعدادی نورون با مراکز و پهنای مشخصی است که تعیین‌کننده پاسخ شبکه به ورودی‌های مختلف هستند. در مرحله یادگیری، این پارامترها به‌گونه‌ای تنظیم می‌شوند که شبکه بتواند الگوی مناسبی از رفتار سیستم را مدل کند. فرآیند یادگیری شامل تنظیم وزن‌های خروجی، مراکز توابع پایه و پارامترهای گوسی توابع پایه است که از طریق قوانین یادگیری مبتنی بر گرادیان و سیگنال خطا بهینه‌سازی می‌شوند.

یکی از ویژگی‌های مهم این کنترل‌کننده، **نرمال‌سازی ورودی‌ها** است که باعث بهبود عملکرد شبکه و جلوگیری از ناپایداری در یادگیری می‌شود. سپس، با استفاده از یک مکانیزم انتشار پیشرو، مقدار خروجی شبکه محاسبه می‌شود. این مقدار نشان‌دهنده پاسخ سیستم عصبی به ورودی داده‌شده است. علاوه بر این، در فرآیند یادگیری و به‌روزرسانی وزن‌ها، از یک تابع خطا برای تنظیم پارامترهای شبکه استفاده می‌شود که منجر به افزایش دقت و بهبود همگرایی در عملکرد کنترل‌کننده می‌شود.

به‌طور کلی، این کنترل‌کننده با داشتن ساختار ساده اما انعطاف‌پذیر، توانایی بالایی در مقابله با **سیستم‌های غیرخطی و پیچیده** دارد. استفاده از آن در کاربردهای کنترلی می‌تواند موجب بهبود عملکرد سیستم، کاهش خطا و افزایش تطبیق‌پذیری با تغییرات محیطی شود.



شکل 3-1 ساختار کلی کنترل کننده RBF

روابط کنترلر RBF:

$$\psi_i = \exp\left(-\frac{||x_{n_c}(k) - c_i||^2}{2\hat{\sigma}_i^2}\right)$$

$$u(k) = \sum_{i=1}^{n_2} v_i \psi_i$$

$$e_c = y_{ref}(k) - y(k)$$

$$v_i(k+1) = v_i(k) + \eta_{c_1} e_c \psi_i \sum_{i=1}^{n_1} w_i \phi_i \left(-\frac{u(k) - \mu_{i1}}{\sigma_i^2}\right)$$

$$\hat{\sigma}_i(k+1) = \hat{\sigma}_i(k) + \eta_{c_2} e_c v_i \frac{||x_{n_c}(k) - c_i||^2}{\hat{\sigma}_i^3} \psi_i \sum_{i=1}^{n_1} w_i \phi_i \left(-\frac{u(k) - \mu_{i1}}{\sigma_i^2}\right)$$

$$c_i(k+1) = c_i(k) + \eta_{c_3} e_c v_i \frac{x_{n_c}(k) - c_i}{\hat{\sigma}_i^2} \psi_i \sum_{i=1}^{n_1} w_i \phi_i \left(-\frac{u(k) - \mu_{i1}}{\sigma_i^2}\right)$$

$$0 \leq \eta_{c_i} \leq 1$$

ساختار شناسایی کننده RBF

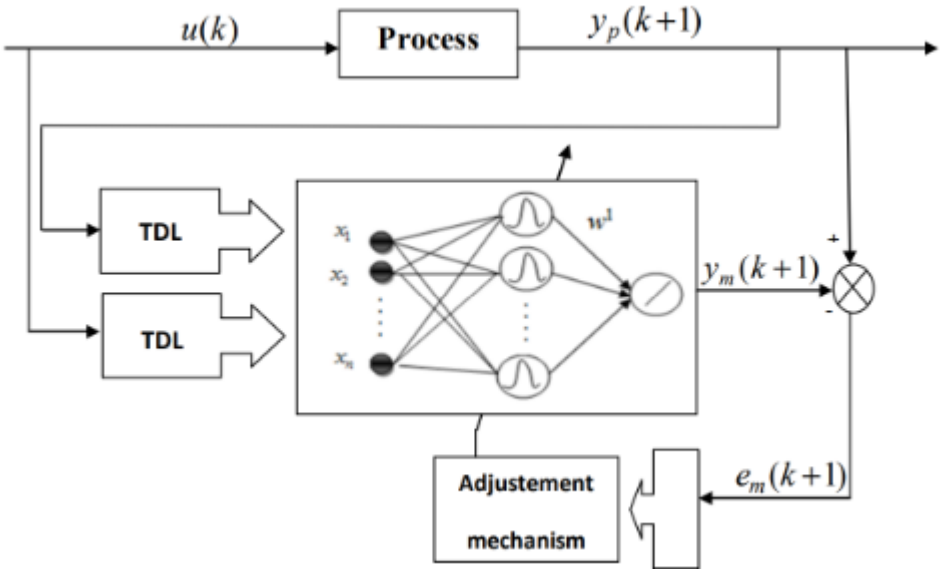
شناسایی‌کننده مبتنی بر **شبکه عصبی پایه شعاعی (RBF)** یک مدل هوشمند برای تخمین و شناسایی رفتار سیستم‌های دینامیکی است که بر اساس ویژگی‌های توابع پایه شعاعی عمل می‌کند. این شبکه قادر است **روابط غیرخطی پیچیده** را بین ورودی‌ها و خروجی‌های یک سیستم مدل کند و یک تخمین دقیق از رفتار سیستم ارائه دهد.

در این شناسایی‌کننده، از یک مجموعه **نورون‌های RBF** در لایه پنهان استفاده می‌شود. هر نورون دارای مرکز **(μ)** و پهنای **(σ)** مخصوص به خود است که میزان تأثیر آن بر ورودی‌ها را تعیین می‌کند. این مقادیر در طول فرآیند یادگیری تنظیم می‌شوند تا تطابق بهتری با داده‌های ورودی و خروجی سیستم برقرار شود. لایه خروجی شبکه نیز شامل **وزن‌های W** است که خروجی نهایی را با ترکیب پاسخ نورون‌های پنهان ایجاد می‌کند.

ورودی‌های شناسایی‌کننده ابتدا **نرمال‌سازی** می‌شوند تا تأثیر مقیاس متغیرهای مختلف کاهش یابد. سپس، مقدار **φ**، که نشان‌دهنده فعال‌سازی هر نورون نسبت به ورودی است، محاسبه می‌شود. این مقدار بر اساس فاصله ورودی از مراکز نورون‌ها و تابع گوسی تعیین می‌شود. در نهایت، خروجی شبکه از طریق ترکیب وزن‌دار مقادیر φ تولید می‌شود.

فرآیند یادگیری شامل **به‌روزرسانی وزن‌ها، مراکز و پهنای نورون‌ها** بر اساس سیگنال خطا است. سیگنال خطا، اختلاف بین مقدار واقعی و مقدار تخمین‌زده‌شده توسط شبکه است. با استفاده از قواعد یادگیری مبتنی بر گرادیان، پارامترهای شبکه تنظیم شده و عملکرد آن در طول زمان بهبود می‌یابد.

این شناسایی‌کننده با **قابلیت تعمیم بالا و توانایی یادگیری رفتارهای غیرخطی**، به‌طور گسترده در مسائل **کنترل تطبیقی، تخمین پارامترهای سیستم و پردازش سیگنال** استفاده می‌شود. ساختار انعطاف‌پذیر و توانایی همگرایی سریع از ویژگی‌های بارز این مدل محسوب می‌شود که آن را به یک ابزار قدرتمند در تحلیل و مدل‌سازی سیستم‌های پیچیده تبدیل کرده است.



شکل 3-2 ساختار شناسایی کننده RBF

روابط شناسایی کننده RBF:

$$\phi_i = \exp\left(-\frac{||x_{n_{id}}(k) - \mu_i||^2}{2\sigma_i^2}\right)$$

$$y_m = \sum_{i=1}^{n_1} w_i \phi_i$$

$$e_m = y(k) - y_m$$

$$w_i(k+1) = w_i(k) + \eta_1 e_m \phi_i$$

$$\sigma_i(k+1) = \sigma_i(k) + \eta_2 w_i \frac{||x_{n_{id}}(k) - \mu_i||^2}{\sigma_i^3} e_m \phi_i$$

$$\mu_i(k+1) = \mu_i(k) + \eta_3 w_i \frac{x_{n_{id}}(k) - \mu_i}{\sigma_i^2} e_m \phi_i$$

$$0 \leq \eta_i \leq 1$$

سیستم حلقه باز

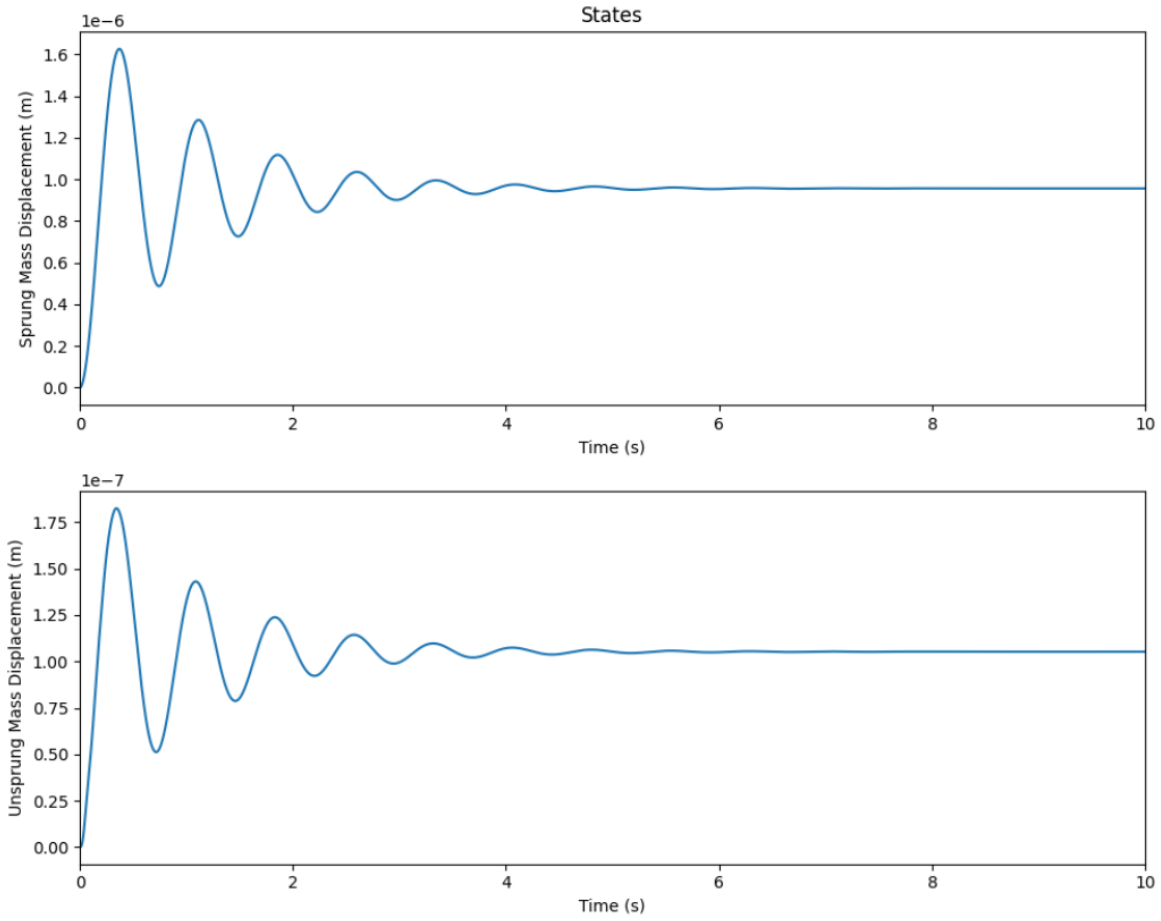
شبیه‌سازی حلقه باز **(Open-Loop Simulation)** یک روش مهم برای بررسی رفتار سیستم‌های دینامیکی بدون اعمال کنترل‌کننده خارجی است. این روش به ما امکان می‌دهد تا واکنش سیستم را تنها بر اساس ورودی‌های مشخص و معادلات دینامیکی آن تحلیل کنیم.

در این شبیه‌سازی، سیستم تعلیق به همراه ورودی‌های مرجع مانند موقعیت، سرعت و شتاب جاده مورد بررسی قرار می‌گیرد. ابتدا شرایط اولیه سیستم تنظیم شده و مقداردهی اولیه متغیرها انجام می‌شود. سپس در هر مرحله از شبیه‌سازی، بر اساس مقادیر ورودی، وضعیت جدید سیستم محاسبه شده و مقادیر حالت‌ها و خروجی ثبت می‌شوند. این فرآیند تا انتهای شبیه‌سازی ادامه می‌یابد و نتایج آن برای تحلیل رفتار سیستم ذخیره می‌شود.

در دو حالت مختلف، می‌توان شبیه‌سازی را اجرا کرد:

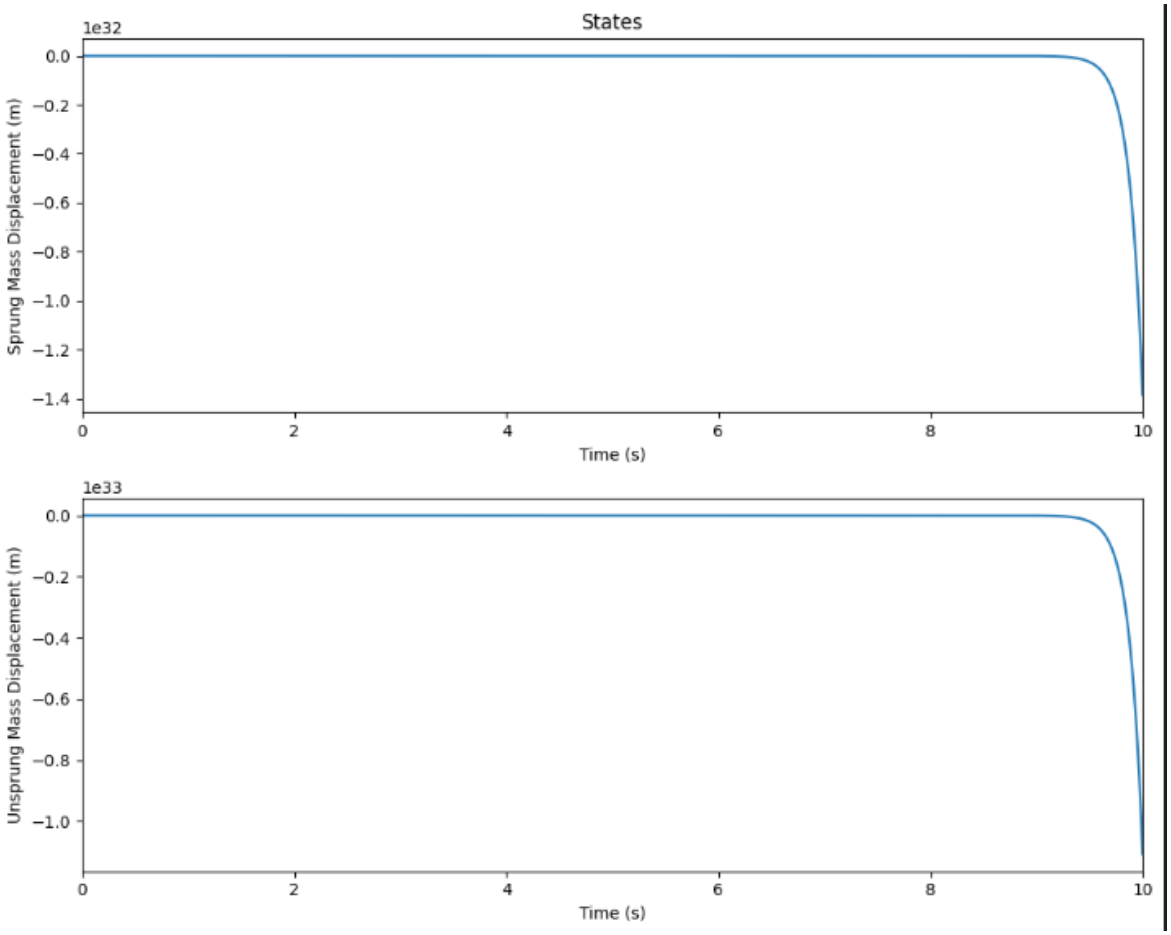
1. **حالت مرجع (Reference Mode)** که در آن خروجی مدل مرجع به‌عنوان مقدار ایده‌آل در نظر گرفته می‌شود.
2. **حالت عادی** که در آن سیستم بدون در نظر گرفتن مدل مرجع تنها تحت تأثیر ورودی‌های داده‌شده واکنش نشان می‌دهد.

در نهایت، داده‌های شبیه‌سازی شامل **تاریخچه زمانی، مقادیر حالت‌ها و خروجی سیستم** در قالب یک دیکشنری جمع‌آوری شده و قابل استفاده برای تحلیل‌های بعدی می‌شوند. این نوع شبیه‌سازی، مبنای مقایسه عملکرد سیستم قبل و بعد از اعمال کنترل‌کننده را فراهم کرده و در طراحی سیستم‌های کنترلی مورد استفاده قرار می‌گیرد.

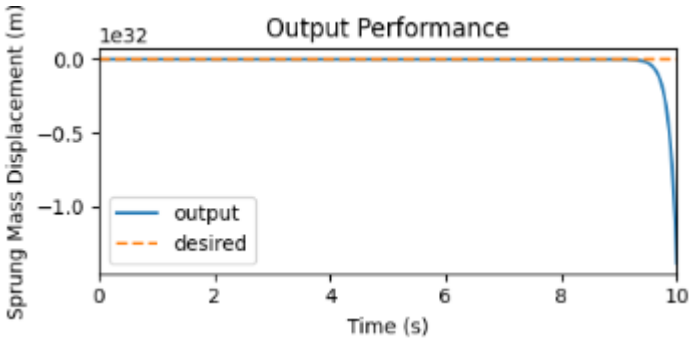


شکل3-3 خروجی سیستم حلقه باز

همانطور که مشاهده میشود هم جسم متصل به فنر و هم جسم جدا از فنر میزان جابجایی میرا شنوده‌ای دارند.

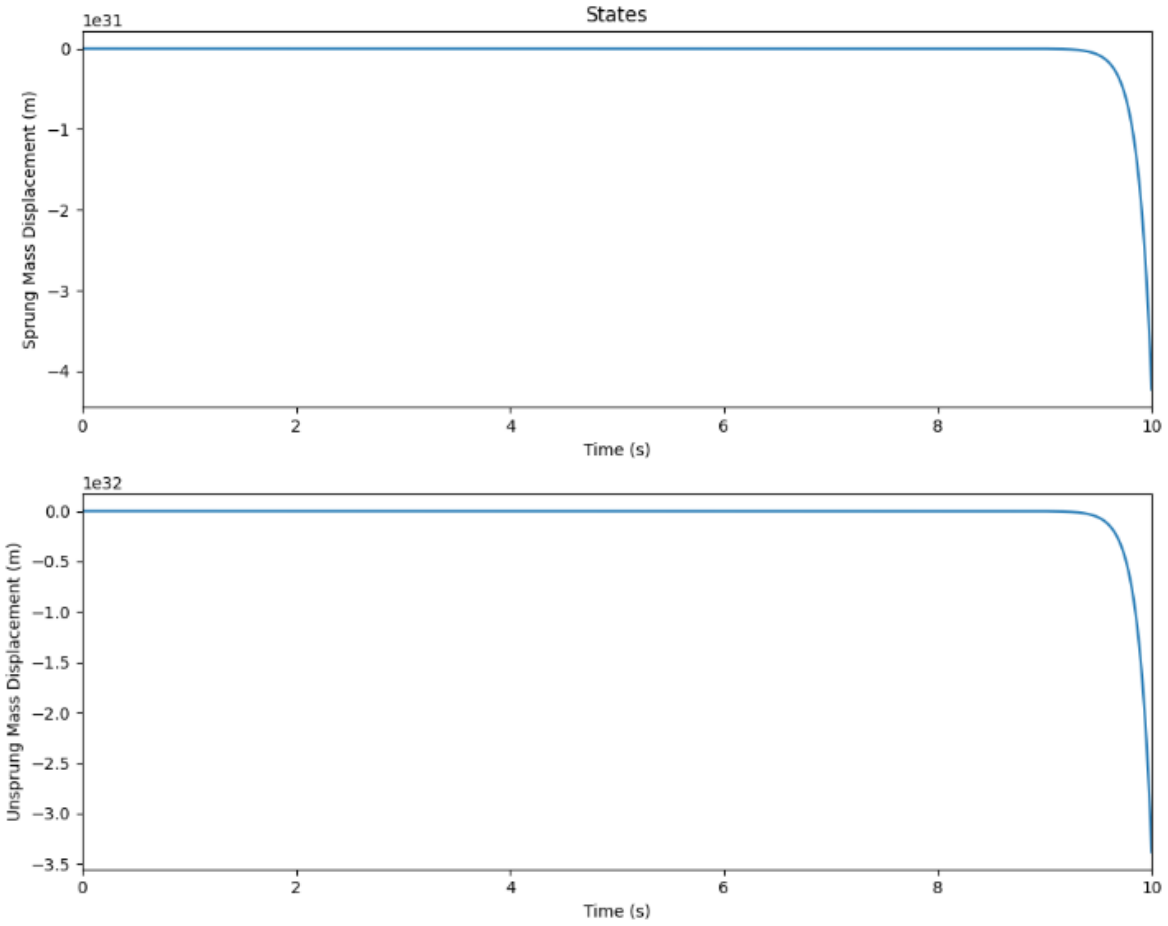


شکل3-4 پاسخ حالات سیستم به ورودی پله

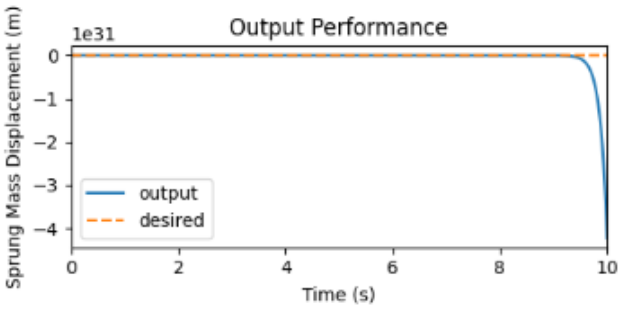


شکل 3-5 خروجی سیستم به ورودی پله

همانطور که مشاهده میکنید سیستم نسبت به ورودی پله ناپایدار می باشد.



شکل3-5 پاسخ سیستم به ورودی سینوسی



شکل3-5 خروجی سیستم نسبت به ورودی سینوسی

همانطور که مشاهده میشود سیستم نسبت به ورودی سینوسی ناپایدار است.

سیستم حلقه بسته

شبیه‌سازی حلقه بسته **(Closed-Loop Simulation)** یک روش ضروری در تحلیل و طراحی سیستم‌های کنترلی است که در آن **سیستم، کنترل‌کننده و شناسایی‌کننده** به‌صورت یکپارچه عمل می‌کنند. در این روش، خروجی سیستم به‌طور مداوم اندازه‌گیری شده و به کنترل‌کننده بازخورد داده می‌شود تا ورودی کنترلی مناسب را تولید کند. این فرآیند موجب بهبود عملکرد سیستم و کاهش خطا نسبت به مقدار مرجع می‌شود.

در این شبیه‌سازی، ابتدا مقادیر اولیه سیستم تنظیم شده و متغیرهای موردنیاز مقداردهی اولیه می‌شوند. در هر گام زمانی، سیستم ورودی مرجع را دریافت کرده و کنترل‌کننده مبتنی بر شبکه عصبی **RBF**با توجه به مقدار **خطای بین خروجی سیستم و مقدار مرجع**، سیگنال کنترلی مناسب را تولید می‌کند. این سیگنال سپس به سیستم اعمال شده و باعث تغییر وضعیت آن می‌شود. علاوه بر این، یک **شناسایی‌کننده RBF** نیز در کنار سیستم قرار دارد که وظیفه یادگیری رفتار سیستم و بهبود عملکرد کنترل را بر عهده دارد.

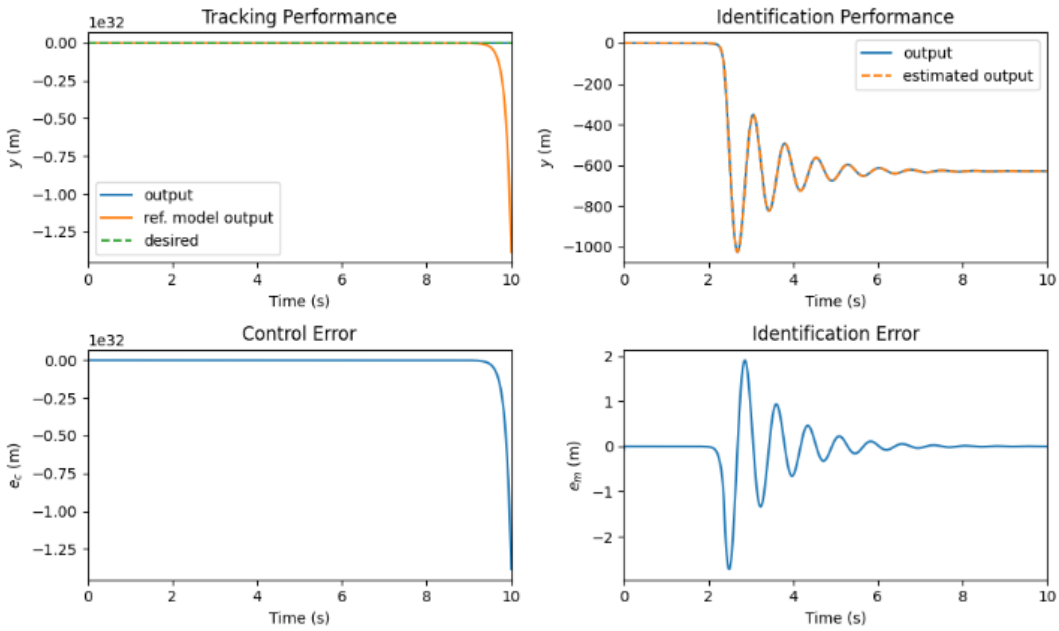
یکی از ویژگی‌های مهم این شبیه‌سازی، **استفاده از ورودی‌های تاخیری برای کنترل و شناسایی** است. برای این منظور، خروجی‌های گذشته و ورودی‌های کنترلی قبلی در قالب بردارهای تأخیری ذخیره

شـــده و به مدل کنترل‌کننده و شـــناســـایی‌کننده داده می‌شـــوند. این تکنیک باعث بهبود دقت کنترل و پایداری سیستم می‌شود.

در هر گام، وزن‌های کنترل‌کننده و شـناسـایی‌کننده بر اسـاس خطاهای محاسـبه‌شـده به‌روزرسـانی شـده و سـیسـتم به‌مرور عملکرد بهتری پیدا می‌کند. در پایان شـبیه‌سـازی، **تاریخچه زمانی، حالات سـیسـتم، خروجی‌ها، مقادیر کنترلی و تغییرات وزن‌ها** ذخیره شـــده و برای تحلیل‌های بعدی قابل اســـتفاده خواهد بود.

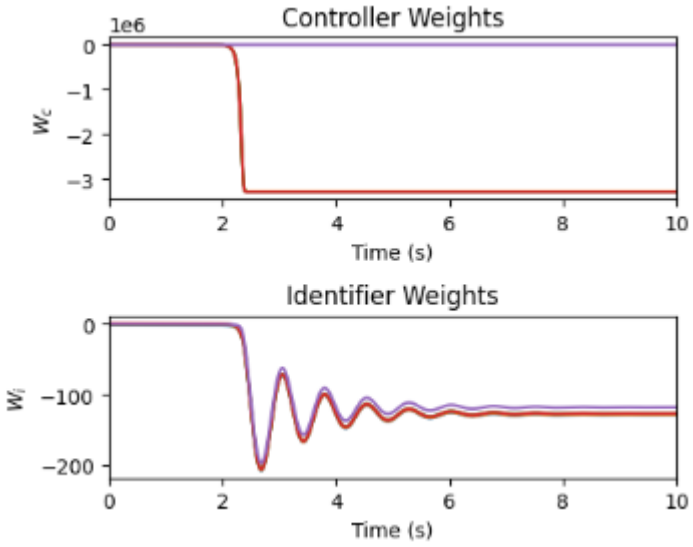
نتایج شبیه سازی

نتایج شبیه سازی با sampling_rate = 0.001 به شکل زیر می باشد.



شکل3-6 نتایج شبیه سازی حلقه بسته

طبق خروجی ای که در بالا مشاهده میکنید شناسایی گر توانسته به خوبی سیستم را شناسایی کند و خطای آن میرا شـونده می باشـد اما کنترلر ناپایدار می باشـد و نتیجه مطلوب را در tracking کسـب نکردیم.

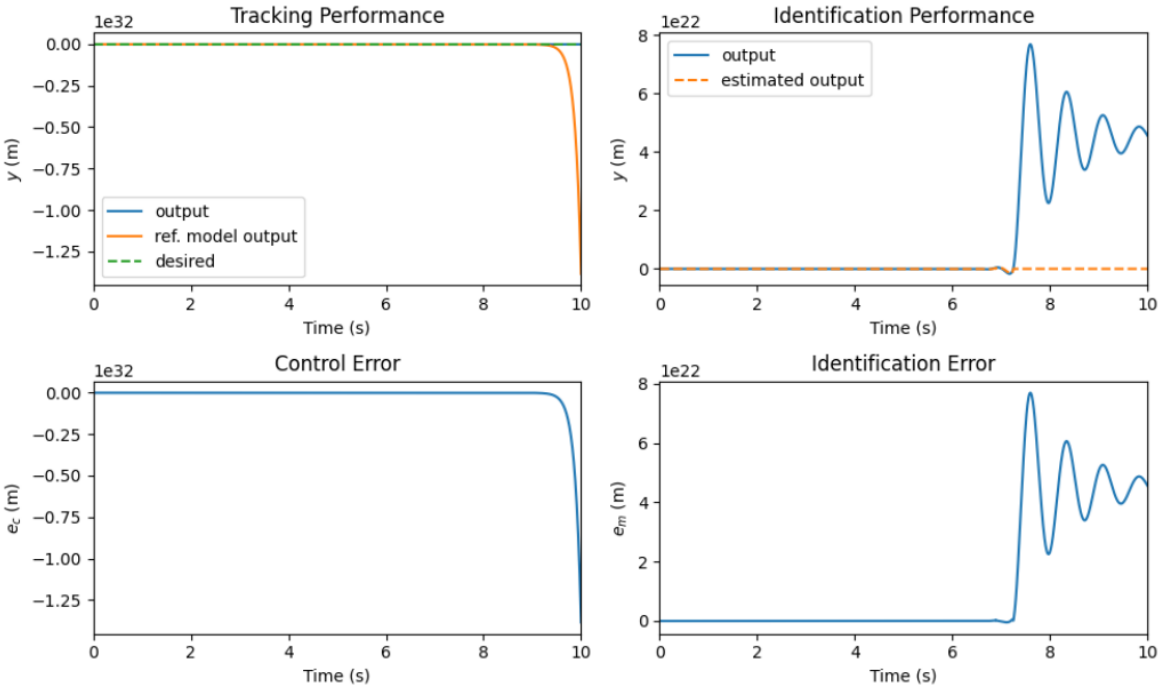


شکل 3-7 تغییرات وزن شناسایی کننده و کنترل کننده

همانطور که مشاهده میکنید وزن های شناسایی گر به صورت مطلوبی تغییر پیدا کرده اند اما از طرفی تغییرات وزن کنترلر نشان دهنده ناپایداری آن می باشد.

انجام آزمایش به ازای مقادیر مختلف sampling rate

Td = 0.01



شکل3-8 نتایج شبیه سازی با نرخ نمونه برداری 0.01 ثانیه

همانطور که مشاهده میکنید نتایج شناسایی گر همچنان میرا شونده می باشد اما در اینجا خروجی مطلوب نیست و کنترلر همچنان نا پایدار است.

لذا بهینه ترین میزان sampling rate را همان 0.001 در نظر می گیریم.

فصل چهارم:

کنترل مبتنی بر شبکه LSTM

معرفی شبکه LSTM

در این پروژه، یک مدل کنترلی مبتنی بر LSTM برای یک سیستم تعلیق **چهار درجه آزادی** توسعه داده شده است. هدف از این تحقیق، طراحی یک کنترل‌کننده هوشمند است که بتواند عملکرد سیستم تعلیق را بهبود بخشیده و نوسانات ناشی از ناهمواری‌های جاده را به حداقل برساند. مدل پیشنهادی از داده‌های ورودی شامل نیروهای اعمالی و پاسخ‌های سیستم استفاده کرده و از طریق یادگیری توالی‌های زمانی، یک استراتژی کنترلی بهینه را ارائه می‌دهد.

در ادامه، ابتدا به معرفی مدل دینامیکی سیستم تعلیق پرداخته شده، سپس روش طراحی و آموزش شبکه LSTM تشریح می‌شود. در نهایت، عملکرد کنترل‌کننده پیشنهادی با استفاده از شبیه‌سازی‌های عددی ارزیابی می‌گردد.

این نکته حائز اهمیت می باشد که برای دریافت نتایج بهتر از 4 درجه آزادی راننده در مدل صرف نظر شده است و صرفا 4 درجه آزادی مربوط به خودرو در نظر گرفته شده است.

ساختار شبکه LSTM

LSTMیک نوع شبکه عصبی بازگشتی (RNN) است که برای مدل‌سازی داده‌های دنباله‌دار و وابسته به زمان طراحی شده است. برخلافRNN های معمولی که با مشکل **ناپدید شدن گرادیان** مواجه هستند، LSTMدارای **سلول‌های حافظه** و **دروازه‌های کنترلی** است که امکان یادگیری روابط طولانی‌مدت را فراهم می‌کنند.

ساختارLSTM

در هر واحدLSTM ، چهار لایه اصلی وجود دارد که ورودی و وضعیت قبلی را پردازش می‌کنند:

1. **دروازه فراموشی(Forget Gate)**

- تصمیم می‌گیرد که چه اطلاعاتی از حالت قبلی حذف شود.

- با استفاده از یک تابع سیگموید، مقدار بین ۰ تا ۱ را تولید کرده و تعیین می‌کند که چه مقدار از اطلاعات قبلی باید نگه داشته شود.

2. **دروازه ورودی(Input Gate)**

- مشخص می‌کند که چه اطلاعات جدیدی به حافظه اضافه شود.
- از دو بخش تشکیل شده: یک لایه سیگموید که تصمیم می‌گیرد چه بخش‌هایی از داده‌های جدید مهم هستند، و یک لایه تانژانت هایپربولیک که مقدار جدید را ایجاد می‌کند.

3. **به‌روزرسانی حافظه سلولی**

- با ترکیب اطلاعات جدید و اطلاعات قبلی، مقدار جدیدی برای حافظه سلولی محاسبه می‌شود.

4. **دروازه خروجی(Output Gate)**

- مشخص می‌کند که چه مقداری از حالت سلولی به خروجی ارسال شود.
- از ترکیب لایه سیگموید و تابع تانژانت هایپربولیک برای محاسبه خروجی استفاده می‌کند.

نمایش گرافیکی ساختارLSTM

در تصویر زیر، معماری یک واحد LSTM نمایش داده شده است:

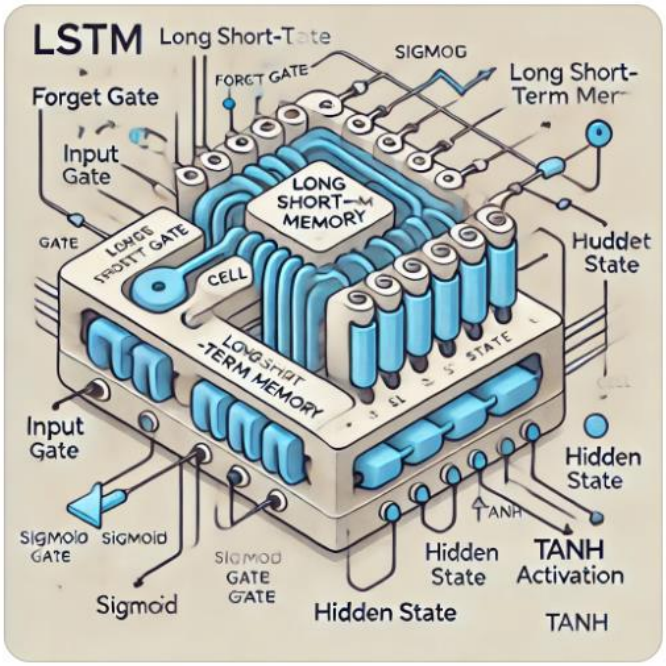
C: بردار حافظه سلولی

h: بردار خروجی

x: ورودی در هر زمان

σ: تابع سیگموید

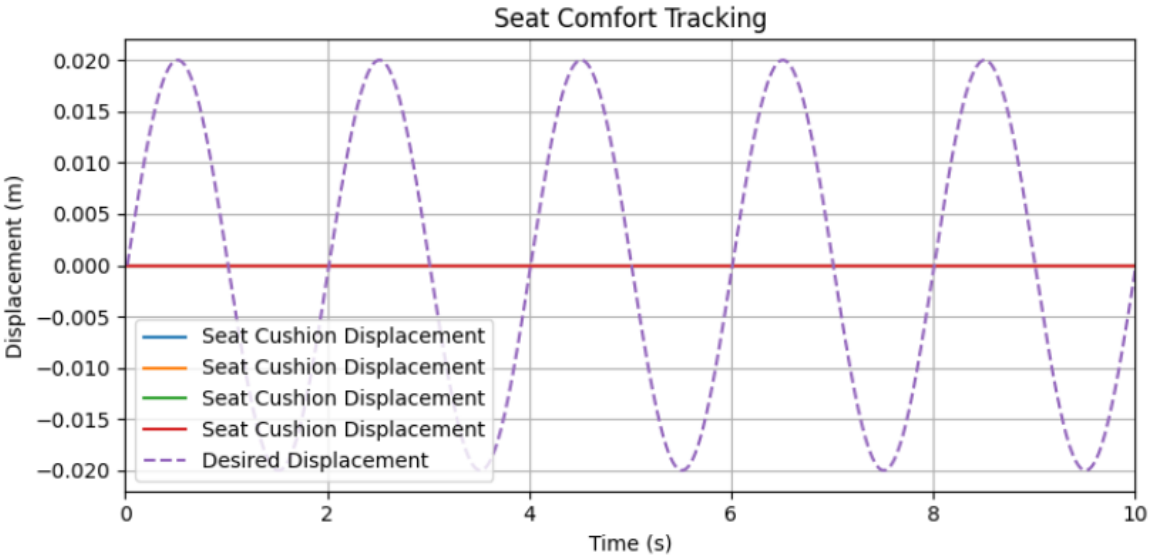
tanh: تابع تانژانت هایپربولیک



شکل4-1 ساختار شبکه عصبی LSTM

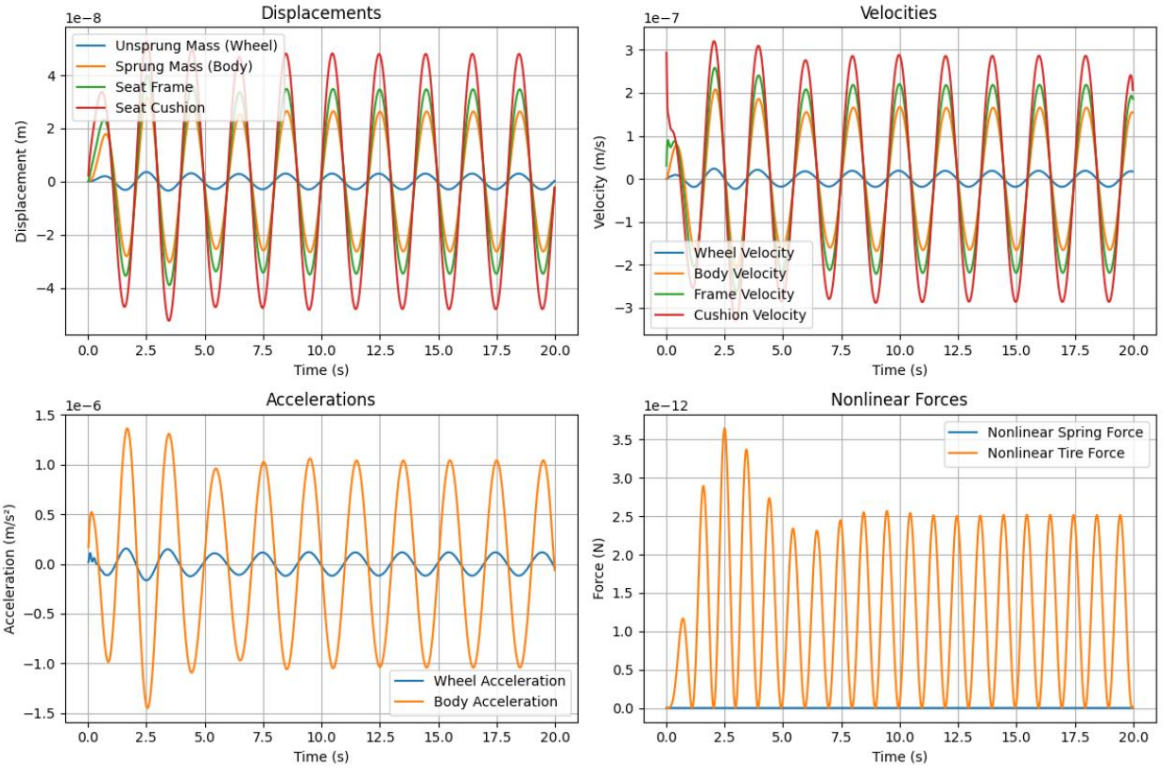
طراحی مدل و پیاده سازی مدل غیر خطی

در این بخش با استفاده از معادلات دینامیکی سیستم که در مقاله موجود می باشد ، سیستم را مدلسازی کردیم



شکل 4-2 مدلسازی سیستم غیر خطی

همانطور که مشاهده می کنید جابجایی صندلی نسبت به حالت مطلوب نا مناسب بوده و نیازمند این می باشد تا با پیاده سازی کنترلر و رویتگر مناسب تعقیب به خوبی انجام شود.



شکل 4-3 نتایج مدلسازی سیستم (حالات سیستم)

با توجه به تصویر بالا می توان گفت که مدلسازی به نحو خوبی انجام شده است.

این تصویر مجموعه‌ای از نمودارهای شبیه‌سازی دینامیکی یک سیستم تعلیق خودرو را نشان می‌دهد که قبل از پیاده‌سازی کنترلر و رویت‌گر انجام شده است. این سیستم شامل جرم‌های فنربندی‌شده (بدنه)، جرم‌های فنربندی‌نشده (چرخ)، قاب صندلی و بالشتک صندلی است. چهار نمودار مجزا برای نمایش تغییرات جابجایی، سرعت، شتاب و نیروهای غیرخطی در طول زمان ارائه شده‌اند.

تحلیل نمودارها:

1. **نمودار بالا-چپ(Displacements)**

- نشان‌دهنده میزان جابجایی‌های اجزای مختلف سیستم تعلیق در واحد متر است.

- مشاهده می‌شود که بالشتک صندلی (Seat Cushion) بیشترین میزان نوسان را دارد، در حالی که چرخ (Unsprung Mass) دارای کمترین دامنه جابجایی است.
- این نمودار نشـــان‌دهنده رفتار کلی ســـیســـتم تعلیق در برابر تحریک ورودی (مانند ناهمواری‌های جاده) است.

2. **نمودار بالا-راست(Velocities)**

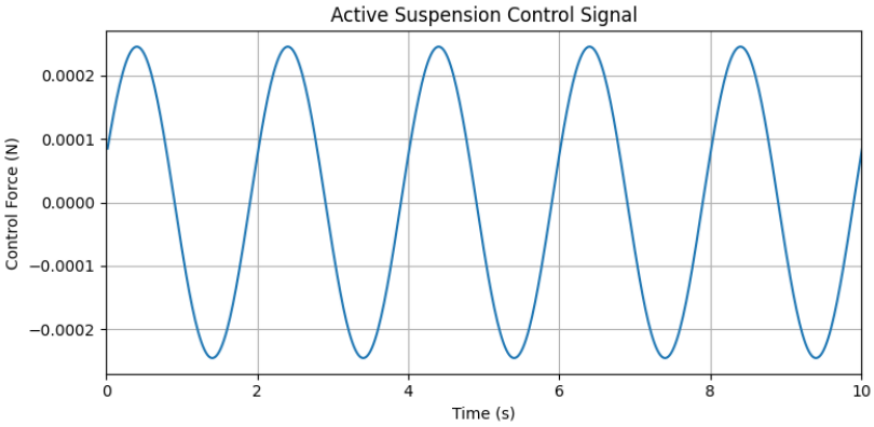
- سرعت‌های اجزای مختلف سیستم نمایش داده شده‌اند.
- مشاهده می‌شود که بیشترین دامنه سرعت مربوط به بالشتک صندلی و کمترین آن مربوط به چرخ است.
- شکل موج‌های مشابه نمودار جابجایی‌ها اما با تغییر فاز مشاهده می‌شود.

3. **نمودار پایین-چپ(Accelerations)**

- شتاب بدنه (Body Acceleration) و شتاب چرخ (Wheel Acceleration) نمایش داده شده‌اند.
- مشـــاهده می‌شـــود که شـــتاب بدنه نســـبت به شـــتاب چرخ دامنه بیشـــتری دارد، که نشان‌دهنده تأثیر مستقیم نوسانات جاده بر روی بدنه خودرو است.
- نوسانات شتاب، عامل تأثیرگذار بر راحتی سرنشینان خودرو هستند.

4. **نمودار پایین-راست(Nonlinear Forces)**

- نمایش‌دهنده نیروهای غیرخطی موجود در ســـیســـتم، از جمله نیروی فنر غیرخطی و نیروی تایر غیرخطی است.
- مشاهده می‌شود که نیروی تایر غیرخطی دارای نوسانات شدیدتری نسبت به نیروی فنر غیرخطی است.
- این نیروها به دلیل وجود المان‌های غیرخطی مانند لاستیک و فنرهای سیستم تعلیق ایجاد می‌شوند.



شکل 4-4 نتایج مدلسازی سیستم (سیگنال کنترلی)

این تصـویر نمودار **سـیگنال کنترلی سـیسـتم تعلیق فعال** را نمایش می‌دهد. محور افقی نشـان‌دهنده زمان (Time) بر حســـب ثانیه اســـت و محور عمودی مقدار **نیروی کنترلی** (Control Force)را بر حسب نیوتن نمایش می‌دهد.

تحلیل نمودار:

1. **ماهیت نوسانی سیگنال کنترلی:**

- نیروی کنترلی دارای یک شکل موج سینوسی متناوب است که نشان‌دهنده اعمال یک نیروی متغیر برای تنظیم عملکرد سیستم تعلیق می‌باشد.
- این نوع ســـیگنال معمولاً در ســـیســـتم‌های کنترلی که از کنترلرهای بازخوردی مانند **کنترلر (Linear Quadratic Regulator) LQR یا PID** استفاده می‌کنند، دیده می‌شود.

2. **دامنه نیروی کنترلی:**

- مقـدار نیروی کنترلی در بـازه‌ای بین **تقریبـاً -0.0002 تـا 0.0002 نیوتن** تغییر می‌کند.

- این مقدار نشـان می‌دهد که کنترلر نیروی محدودی را برای کاهش نوسـانات سـیسـتم تعلیق اعمال می‌کند.

3. **دوره تناوب و فرکانس سیگنال:**

- دوره تناوب (زمان یک سـیکل کامل نوسـان) حدود **2ثانیه** اسـت که نشـان‌دهنده **فرکانس پایین سیگنال کنترلی** است.
- این امر معمولاً نشـان‌دهنده تنظیمات کنترلری اسـت که هدف آن **بهبود پایداری و راحتی سرنشینان** بدون اعمال نیروهای شدید است.

تولید مجموعه داده برای آموزش مدل در سیستم تعلیق فعال خودرو

روش تولید داده

در این روش، داده‌های مورد نیاز با اجرای شبیه‌سازی‌های متعددی از سیستم تعلیق خودرو تولید می‌شوند. شبیه‌سازی‌ها با شرایط اولیه تصادفی انجام شده و ورودی‌های کنترلی مشخصی به سیستم اعمال می‌شود. سپس، خروجی‌های سیستم و وضعیت‌های داخلی آن در طول زمان ثبت شده و به عنوان مجموعه داده مورد استفاده قرار می‌گیرند.

ساختار مجموعه داده

مجموعه داده تولید شده شامل سه بخش اصلی است:

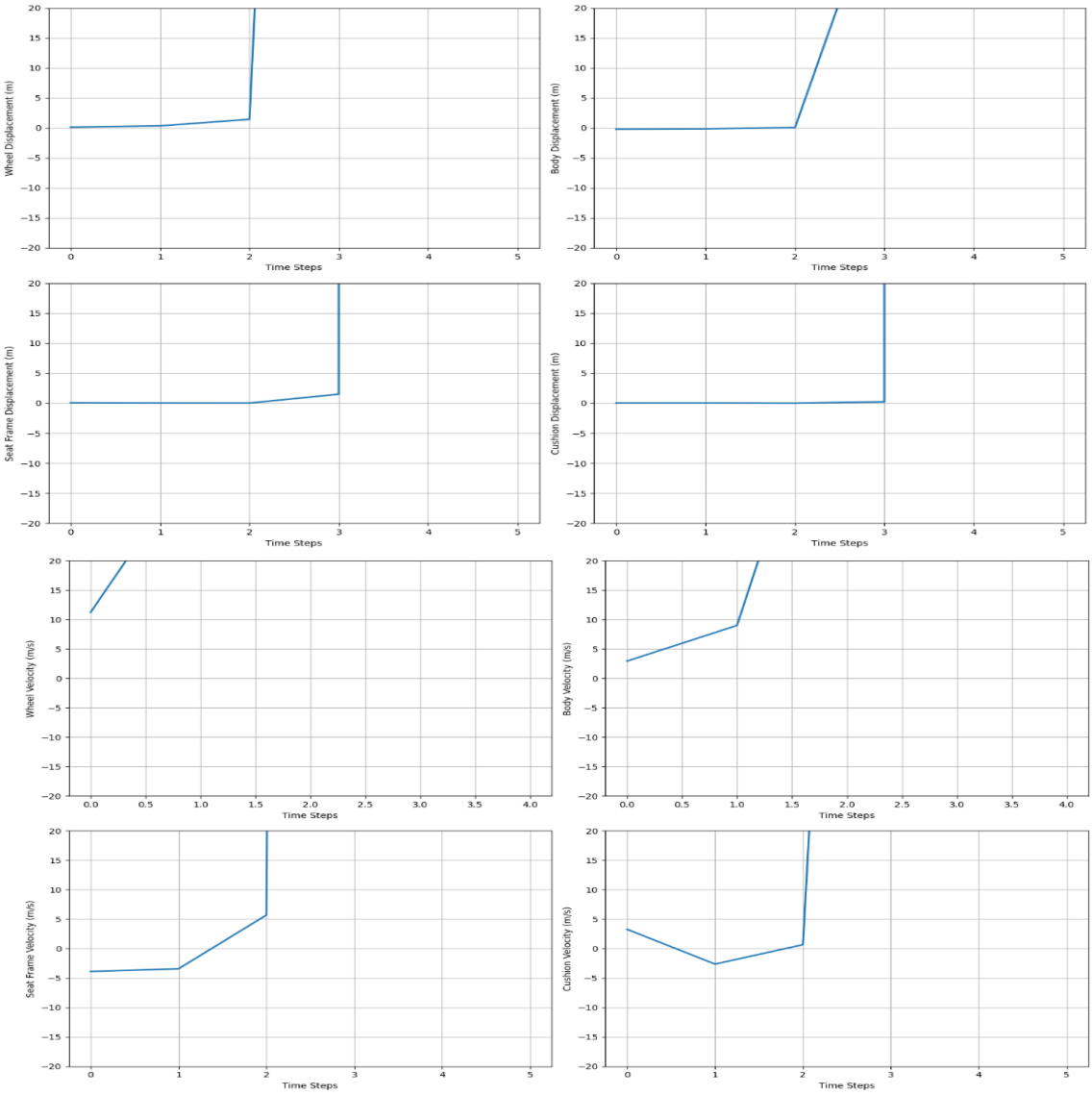
1. **ورودی‌های کنترلی :**نیروی اعمال شده توسط سیستم تعلیق فعال که به دو صورت **جاده‌ای (Road Disturbance)**و **پله‌ای (Step Input)** تولید می‌شود.
2. **وضعیت‌های داخلی :**مقادیر مربوط به متغیرهای داخلی سیستم تعلیق که شامل جابجایی‌ها، سرعت‌ها و شتاب‌های بخش‌های مختلف سیستم است.
3. **خروجی‌های اندازه‌گیری‌شده :**داده‌هایی که از حسگرهای سیستم جمع‌آوری می‌شوند و نشان‌دهنده رفتار کلی سیستم در پاسخ به ورودی‌های کنترلی هستند.

پیش‌پردازش داده‌ها

به منظور بهبود کیفیت داده‌ها و افزایش دقت مدل‌های یادگیری ماشین، داده‌های تولید شده نرمال‌سازی می‌شوند. این فرآیند باعث می‌شود که متغیرهای ورودی و خروجی در یک بازه مشخص مقیاس‌بندی شده و تأثیر مقادیر بسیار بزرگ یا بسیار کوچک بر آموزش مدل کاهش یابد. علاوه بر این، داده‌ها به‌گونه‌ای سازماندهی می‌شوند که بتوان از آن‌ها برای دو هدف استفاده کرد:

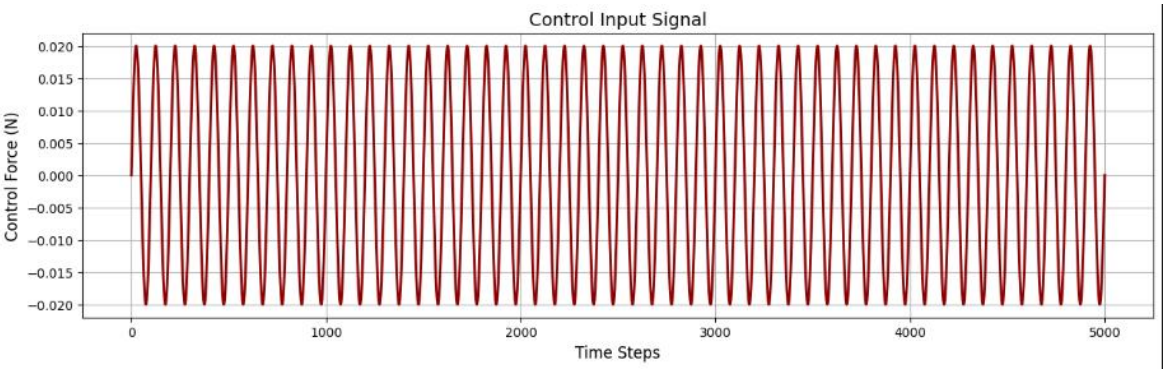
- **مدل‌سازی دینامیکی :**پیش‌بینی وضعیت آینده سیستم بر اساس وضعیت‌های قبلی و ورودی‌های کنترلی.

- **طراحی رویت‌گر :**تخمین وضعیت‌های داخلی سیستم بر اساس خروجی‌های اندازه‌گیری‌شده و ورودی‌های کنترلی.



شکل 4-5 خروجی دیتا سازی

پس از تولید مجموعه داده‌ها، نتایج شبیه‌سازی برای بررسی پاسخ سیستم تعلیق خودرو تحلیل شده‌اند. نمودارهای ارائه‌شده تغییرات جابه‌جایی و سرعت اجزای مختلف سیستم، از جمله چرخ، بدنه، فریم صندلی و کوسن را در طول زمان نشان می‌دهند. رفتار سیستم در واکنش به ورودی‌های کنترلی و تحریکات جاده‌ای بررسی شده و ناپیوستگی‌های موجود در داده‌ها نشان‌دهنده تغییرات ناگهانی ناشی از ورودی‌های کنترلی است. این تحلیل‌ها برای ارزیابی دقت مدل‌سازی و عملکرد سیستم تعلیق در شرایط مختلف ضروری هستند.



شکل 4-6 ورودی کنترلی

این نمودار سیگنال ورودی کنترلی را نشان می‌دهد که به سیستم تعلیق اعمال شده است. این سیگنال یک موج متناوب با دامنه محدود در محدوده [−0.02,0.02][-0.02, 0.02][−0.02,0.02]نیوتن است که با یک فرکانس ثابت تغییر می‌کند. هدف از اعمال این سیگنال بررسی واکنش سیستم تعلیق به تحریک‌های دوره‌ای و ارزیابی عملکرد کنترل‌کننده در کاهش ارتعاشات و بهبود پایداری خودرو می‌باشد.

همانطور که مشاهده می کنید مدل به شدت ناپایدار بوده و حتی با حذف 4 درجه آزادی مربوط به راننده خروجی نمودار های حالات بشدت نا پایدار می باشد و در آزمایش اول سیستم موفق به تعقیب سیگنال مورد انتظار نشده است.

این نتیجه نشان می دهد مدل سازی سیستم با این روش تقریبا غیر ممکن می باشد.

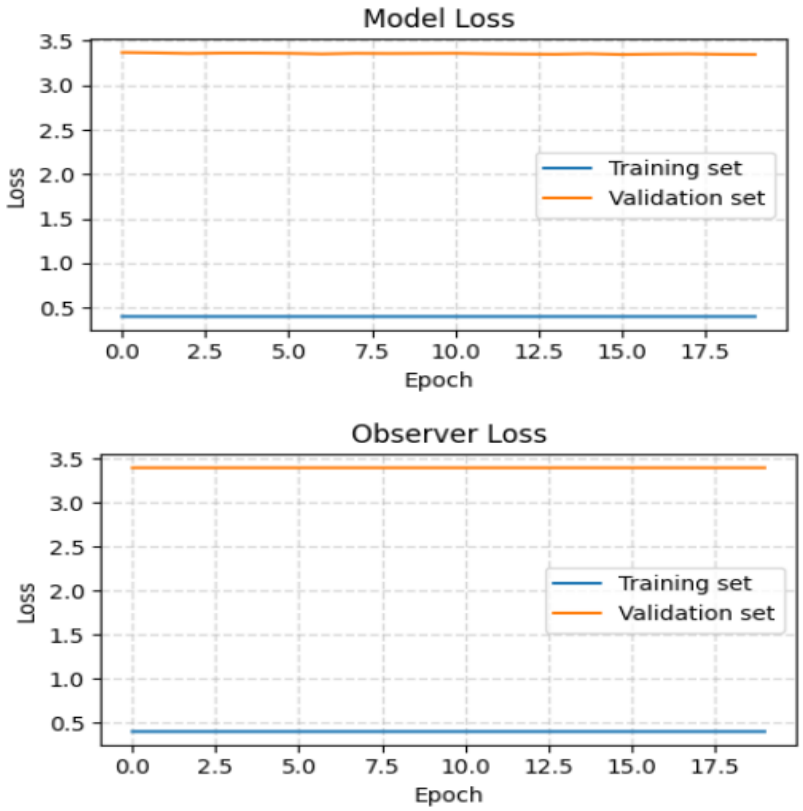
مدل پیشنهادی برای شناسایی و رویت سیستم

در این پژوهش، از شبکه حافظه‌دار بلندمدت (LSTM) به عنوان یک مدل داده‌محور برای شناسایی سیستم و طراحی رویتگر استفاده شده است. دو مدل مجزا توسعه داده شده‌اند: یک مدل LSTM برای شناسایی رفتار دینامیکی سیستم و یک مدل دیگر برای طراحی رویتگر جهت تخمین متغیرهای غیرقابل مشاهده.

مدل شناسایی سیستم بر اساس ورودی‌های مشاهده‌شده و خروجی‌های متناظر، ساختاری عمیق شامل یک لایه LSTM و لایه‌های متراکم (Dense) برای استخراج ویژگی‌های وابسته به زمان دارد. این مدل به‌منظور پیش‌بینی حالات آینده سیستم با استفاده از داده‌های آموزشی مورد استفاده قرار می‌گیرد.

در مقابل، رویتگر مبتنی بر LSTM وظیفه تخمین حالات سیستم را بر اساس ورودی‌های مشاهده‌شده بر عهده دارد. این مدل با ساختاری مشابه مدل شناسایی، اما با معماری مناسب برای تخمین حالات پنهان، امکان بازسازی متغیرهای دینامیکی سیستم را فراهم می‌کند.

برای بهبود دقت یادگیری، داده‌ها قبل از ورود به شبکه نرمال‌سازی شده‌اند و از تکنیک‌هایی مانند تنظیم نرخ یادگیری و کاهش نوسانات گرادیان در بهینه‌سازی استفاده شده است. نتایج اولیه نشان می‌دهد که این روش می‌تواند در مدل‌سازی سیستم‌های دینامیکی عملکرد قابل قبولی ارائه دهد، اما همچنان محدودیت‌هایی در مقابله با ناپایداری‌ها و پیچیدگی‌های سیستم تعلیق پیشرفته مشاهده شده است.



شکل 4-7 نتایج آموزش مدل

نتایج آموزش مدل LSTM برای شناسایی و رویت سیستم نشان‌دهنده عملکرد نامناسب این مدل در تخمین پارامترهای سیستم است. همان‌طور که در نمودارهای **Model Loss** و **Observer Loss** مشاهده می‌شود، مقدار **خطای اعتبارسنجی** (Validation Loss)در تمامی دوره‌های آموزشی تقریباً ثابت و در سطح بالایی باقی مانده است. این موضوع بیانگر آن است که مدل نتوانسته است الگوی مناسبی از داده‌های آموزشی استخراج کند و در نتیجه، تعمیم‌پذیری آن برای داده‌های جدید بسیار ضعیف خواهد بود.

از ســـوی دیگر، اختلاف قابل توجه میان **خطای آموزش** (Training Loss)و **خطای اعتبارســـنجی** بیانگر آن است که مدل دچار **overfitting**نشده، بلکه اساساً در یادگیری الگوهای سیستم ناتوان بوده است. این ضعف نشان می‌دهد که معماری انتخاب‌شده، میزان داده‌ها، و یا فرآیند تنظیم مدل، نتوانسته‌اند مدل مناسبی برای سیستم مورد نظر ارائه دهند.

بنابراین، اســتفاده از این مدل برای کاربردهای کنترلی یا شــناســایی ســیســتم نامناســب بوده و نیاز به روش‌های جایگزین مانند **مدل‌های کلاســـیک کنترل، تخمینگرهای مبتنی بر فیلتر کالمن، یا روش‌های یادگیری عمیق پیشـــرفته‌تر**)مانند شـــبکه‌های Transformer یا (Hybrid Models احساس می‌شود.

نتیجه گیری کلی

در این پژوهش، از شبکه‌های بازگشتی **LSTM**برای شناسایی و رویت سیستم تعلیق پیشرفته استفاده شد. هدف از این مطالعه، بررسی کارایی این روش در تخمین دینامیک سیستم و طراحی یک رویتگر کارآمد برای تخمین متغیرهای حالت بود. با این حال، نتایج حاصل از آموزش مدل نشان داد که روش پیشنهادی دارای ضعف‌های جدی بوده و نمی‌تواند عملکرد مطلوبی را در این زمینه ارائه دهد.

.1عملکرد ضعیف در فرآیند آموزش

نمودارهای مربوط به **میزان خطای مدل و رویتگر** نشان داد که مقدار خطای اعتبارسنجی (Validation Loss)در طول کل دوره‌های آموزشی تقریباً ثابت باقی مانده و در سطح بالایی قرار دارد. این پدیده نشان‌دهنده آن است که مدل نتوانسته است الگوی مناسب و معناداری را از داده‌های آموزشی استخراج کند. به عبارت دیگر، **فرآیند یادگیری مدل ناقص بوده و شبکه از لحاظ تعمیم‌پذیری ناتوان است.**

.2ضعف در تعمیم‌پذیری و همگرایی مدل

یکی از مشکلات اساسی مشاهده‌شده در نتایج، اختلاف ناچیز بین میزان خطای آموزش (Training Loss)و اعتبارسنجی بود که معمولاً بیانگر عدم یادگیری صحیح شبکه است. در یک مدل بهینه، انتظار می‌رود که با افزایش تعداد دوره‌های آموزشی، میزان خطای مدل کاهش یافته و همگرایی در یادگیری اتفاق بیفتد. اما در اینجا مدل در **مرحله آموزش حتی به یک حداقل موضعی مناسب نیز نرسیده است** .این نشان می‌دهد که شبکه **LSTMبرای مدل‌سازی دینامیک سیستم تعلیق پیشرفته مناسب نبوده** و نیازمند معماری‌های جایگزین است.

.3ناکارآمدی در تخمین متغیرهای سیستم

مدل طراحی‌شده نه‌تنها در پیش‌بینی متغیرهای خروجی سیستم دچار مشکل است، بلکه رویتگر مبتنی بر LSTM نیز قادر به ارائه تخمین مناسبی از متغیرهای حالت سیستم نیست. این مسئله در عمل موجب عدم اطمینان در پیاده‌سازی واقعی می‌شود، چرا که رویتگرها معمولاً باید بتوانند نویزها و ناپایداری‌های محیطی را نیز در نظر گرفته و عملکردی پایدار ارائه دهند.

راهکارهای پیشنهادی

با توجه به نتایج به‌دست‌آمده، پیشنهاد می‌شود که به جای استفاده از **LSTM**، روش‌های جایگزین مورد بررسی قرار گیرند. برخی از گزینه‌های پیشنهادی عبارت‌اند از:

- **روش‌های کلاسیک کنترل و فیلترگذاری :**استفاده از **فیلتر کالمن توسعه‌یافته (EKF)** یا **فیلتر کالمن غیرخطی (UKF)** برای بهبود رویت سیستم‌های پیچیده. این روش‌ها برای سیستم‌های دینامیکی مانند سیستم تعلیق عملکرد بهتری نسبت به مدل‌های یادگیری عمیق خواهند داشت.
- **شبکه‌های ترکیبی :**ترکیب **مدل‌های کلاسیک کنترلی با یادگیری عمیق**، مانند استفاده از **شبکه‌های عصبی ترکیبی (Hybrid Models)** که داده‌های استخراج‌شده از مدل‌های فیزیکی را بهبود می‌بخشند.
- **روش‌های یادگیری عمیق پیشرفته :**استفاده از مدل‌هایی نظیر **Transformer**یا **Attention-based Networks**که در پردازش سری‌های زمانی عملکرد بهتری نسبت به LSTMدارند.
- **بهینه‌سازی معماری مدل :**بررسی **افزایش تعداد لایه‌ها، تغییر در توابع فعال‌سازی، استفاده از مکانیزم‌های توجه (Attention Mechanisms)** و تنظیم دقیق‌تر هایپرپارامترها برای بهبود فرآیند یادگیری.

فصل پنجم:

کنترل مبتنی بر RL

معرفی روش کنترل مبتنی بر RL

پروژه کنترل هوشمند سیستم‌های تعلیق خودرو یکی از چالش‌های پیچیده در مهندسی خودرو است که می‌تواند تاثیر زیادی بر راحتی راننده و پایداری خودرو داشته باشد. یکی از روش‌های نوین و پیشرفته در این زمینه استفاده از الگوریتم‌های یادگیری تقویتی Reinforcement Learning) یا (RL است. این الگوریتم‌ها با امکان یادگیری از محیط و بهینه‌سازی مستمر عملکرد سیستم، می‌توانند راه‌حل‌های بهینه‌تری برای کنترل سیستم‌های پیچیده مانند سیستم تعلیق خودرو ارائه دهند.

در این گزارش، به تحلیل و طراحی یک سیستم کنترل هوشمند برای یک مدل سیستم تعلیق 4 درجه آزادی خودرو پرداخته شده است. برای این منظور، از الگوریتم Soft Actor-Critic (SAC) به عنوان یک روش پیشرفته یادگیری تقویتی استفاده شده است. این الگوریتم به دلیل توانایی بالای آن در یادگیری سیاست‌های بهینه در محیط‌های پیوسته و پیچیده، انتخاب مناسبی برای کنترل سیستم تعلیق است. سیستم تعلیق 4 درجه آزادی به دلیل پیچیدگی‌های مدل و نیاز به دقت بالا در تنظیمات، چالش‌های زیادی را در کنترل به همراه دارد که استفاده از متدهای هوشمند مانند SAC می‌تواند به بهبود عملکرد آن کمک کند.

در این گزارش، ابتدا به معرفی سیستم تعلیق 4 درجه آزادی و ویژگی‌های آن پرداخته می‌شود، سپس به بررسی الگوریتم SAC و نحوه پیاده‌سازی آن برای کنترل سیستم تعلیق خودرو خواهیم پرداخت. هدف از این تحقیق بهبود کارایی سیستم تعلیق از طریق یادگیری ماشین و الگوریتم‌های پیشرفته به منظور ارتقاء راحتی و ایمنی خودرو است.

معرفی الگوریتم SAC

الگوریتم Soft Actor-Critic (SAC) یک الگوریتم پیشرفته یادگیری تقویتی است که به طور خاص برای محیط‌های پیوسته طراحی شده است. این الگوریتم به عنوان یک الگوریتم actor-critic شناخته می‌شود، که در آن دو شبکه عصبی مجزا برای مدل‌سازی سیاست (actor) و تابع ارزش (critic) استفاده می‌شود. یکی از ویژگی‌های برجستهSAC ، استفاده از استراتژی‌های نرم (soft) برای بهبود پایداری و بهره‌وری در یادگیری است.

SACبرخلاف الگوریتم‌های کلاسیک که از سیاست‌های قطعی (deterministic) استفاده می‌کنند، از سیاست‌های تصادفی (stochastic) بهره می‌برد. این به این معناست که در هر گام از فرآیند یادگیری، سیاست انتخابی به صورت تصادفی از توزیع‌های احتمالی انتخاب می‌شود که به طور موثر باعث افزایش تنوع و توانایی کشف محیط می‌شود.

یکی دیگر از ویژگی‌های کلیدیSAC ، استفاده از مفهوم "entropy" در فرآیند یادگیری است. این مفهوم باعث می‌شود که الگوریتم به طور همزمان نه تنها به حداکثر کردن پاداش‌ها بلکه به حداقل کردن پیش‌بینی‌پذیری سیاست (یعنی افزایش تصادفی بودن آن) نیز توجه داشته باشد. این رویکرد کمک می‌کند تا الگوریتم بتواند بهتر به اکتشاف محیط بپردازد و از بیش‌برازش (overfitting) جلوگیری کند.

SACبه طور گسترده در مسائل پیچیده با محیط‌های پیوسته و غیرخطی مورد استفاده قرار می‌گیرد، زیرا توانایی یادگیری بهینه سیاست‌ها و بهینه‌سازی عملکرد سیستم‌ها را در چنین محیط‌هایی به خوبی دارد.

برای اطلاعات بیشتر از روابط به نوت بوک نهایی که به پیوست فایل ارسال شده است مراجعه کنید.

توضیح بخش های مختلف برنامه

توضیح عملکرد کلاس ReplayBuffer

کلاس **ReplayBuffer** وظیفه ذخیره‌سازی و مدیریت تجربیات محیط را بر عهده دارد. در فرآیند یادگیری تقویتی، این تجربیات شامل وضعیت‌های مختلف سیستم، اقدامات انجام شده، پاداش‌های دریافتی، وضعیت‌های بعدی و اطلاعاتی در مورد پایان تعامل است. این بافر به الگوریتم کمک می‌کند تا از تجربیات گذشته برای بهبود سیاست یادگیری استفاده کند. عملکرد کلی این کلاس به شرح زیر است:

1. ذخیره تجربیات

این بخش به‌طور مداوم تجربیات جدید را ذخیره می‌کند. هر تجربه شامل وضعیت فعلی، اقدام انجام‌شده، پاداش دریافت‌شده، وضعیت بعدی و مشخصه پایان تعامل است. این اطلاعات به‌طور مرتب در یک فضای ذخیره‌سازی ذخیره می‌شود تا در مراحل بعدی برای یادگیری از آن‌ها استفاده شود.

2. نمونه‌برداری تصادفی

در این بخش، تجربیات ذخیره‌شده به‌طور تصادفی انتخاب می‌شوند تا در فرآیند یادگیری استفاده شوند. انتخاب تصادفی تجربیات به الگوریتم این امکان را می‌دهد که یادگیری را بدون وابستگی به ترتیب تجربیات انجام دهد و از همبستگی میان نمونه‌ها جلوگیری کند.

3. محدود کردن اندازه بافر

برای جلوگیری از مصرف بیش از حد حافظه، تعداد تجربیات ذخیره‌شده در بافر محدود به یک مقدار مشخص است. وقتی تعداد تجربیات به حد مجاز برسد، تجربیات قدیمی‌تر جایگزین تجربیات جدیدتر می‌شوند.

4. میزان تجربیات ذخیره‌شده

این بخش تعداد تجربیات ذخیره‌شده در بافر را برمی‌گردلند، که به الگوریتم این امکان را می‌دهد تا از تعداد تجربیات ذخیره‌شده مطلع شود و تصمیمات خود را براساس آن تنظیم کند.

در مجموع، این سیستم حافظه‌ای به الگوریتم یادگیری کمک می‌کند تا از تجربیات گذشته خود برای بهبود عملکرد و یادگیری بهتر استفاده کند.

توضیح عملکرد شبکه CriticNetwork

شـــبکه **CriticNetwork**یکی از اجزای مهم الگوریتم‌های یادگیری تقویتی اســـت که وظیفه ارزیابی سـیاسـت‌ها را بر عهده دارد. در این شـبکه، دو تابع **Q**به طور همزمان برای تخمین ارزش وضـعیت‌ها و اقدامات مختلف اسـتفاده می‌شـود. در اینجا، از دو شـبکه مجزا برای هر یک از توابع **Q1**و **Q2**اسـتفاده می‌شود تا تنوع و دقت تخمین‌ها افزایش یابد.

1. ساختار شبکه

این شبکه شامل دو بخش اصلی است:

- **Q1**:این شـبکه برای تخمین مقدار Q1 که ارزیابی اولیه از ارزش وضـعیت-اقدام اسـت طراحی شده است.
- **Q2**:شـــبکه دوم برای تخمین مقدار Q2 که ارزیابی متفاوتی از همان وضـــعیت-اقدام را فراهم می‌کند طراحی شده است.

هر دو شبکه به‌طور مشابه از لایه‌های **Dense**برای پردازش اطلاعات استفاده می‌کنند:

- لایه‌های **Dense**با فعال‌سازی‌های **swish**و **relu**برای پردازش ویژگی‌های وضعیت و اقدام.
- در نهایت، خروجی نهایی به‌صورت تک‌بعدی برای هر شبکه **Q1**و **Q2**تولید می‌شود که نمایانگر ارزش آن وضعیت-اقدام است.

2. ورودی‌های شبکه

ورودی‌های شبکه ترکیبی از وضعیت‌ها و اقدامات هستند که به هم متصل می‌شوند:

- وضـعیت (state) و اقدام (action) در طول بعد ویژگی‌ها با هم ترکیب می‌شـوند تا یک ورودی مشترک برای هر دو شبکه ایجاد شود.

3. پیش‌پردازش و محاسبات

در این بخش:

- ابتدا وضعیت و اقدام با هم ترکیب می‌شوند.
- سپس از دو شبکه **Q1**و **Q2**برای پردازش این ورودی و تخمین ارزش آن استفاده می‌شود.

4. خروجی شبکه

در نهایت، این شـــبکه دو خروجی به شـــکل **q1**و **q2**تولید می‌کند که نمایانگر تخمین‌های مختلف از ارزش وضـــعیت-اقدام هســـتند. این مقادیر به الگوریتم کمک می‌کنند تا تصـــمیمات بهتری برای انتخاب سیاست اتخاذ کند.

توضیح عملکرد شبکه ValueNetwork

شـبکه **ValueNetwork**یکی از اجزای مهم در الگوریتم‌های یادگیری تقویتی اسـت که وظیفه تخمین تابع ارزش (Value Function) را بر عهده دارد. هدف این شـبکه این اسـت که برای هر وضـعیت موجود در محیط، یک تخمین از ارزش آن وضـــعیت ارائه دهد که نمایانگر کیفیت آن وضـــعیت برای اتخاذ تصمیمات مناسب در راستای بهینه‌سازی پاداش باشد.

1. ساختار شبکه

شـــبکه **ValueNetwork**شـــامل چندین لایه **Dense**به همراه لایه‌های اضـــافی اختیاری برای بهبود عملکرد است:

- **fc1**:اولین لایه **Dense**با ابعاد **fc1_dims**که برای پردازش ورودی‌ها اســتفاده می‌شــود. این لایه از تابع فعال‌سازی **ReLU**برای معرفی غیرخطی بودن به مدل استفاده می‌کند.
- **fc2**:دومین لایه **Dense**با ابعاد **fc2_dims**که پردازش‌های پیچیده‌تر را انجام می‌دهد.
- **v**:لایه نهایی که یک مقدار تک‌بعدی به‌عنوان خروجی شـــبکه تولید می‌کند. این مقدار نمایانگر تخمین ارزش وضعیت است.

2. لایه‌های اختیاری

برای بهبود پایداری فرآیند آموزش و جلوگیری از بیش‌برازش(overfitting) ، دو لایه اختیاری در این شبکه وجود دارند:

- **Batch Normalization (BN)**: این لایه‌ها برای نرمال‌سازی داده‌ها پس از هر لایه **Dense** استفاده می‌شوند تا پایداری فرآیند آموزش را افزایش دهند. این لایه‌ها می‌توانند به بهبود سرعت و کیفیت آموزش کمک کنند.
- **Dropout**: این لایه به‌طور تصادفی برخی از نرون‌ها را در حین آموزش غیرفعال می‌کند تا از بیش‌برازش جلوگیری کند و به مدل کمک می‌کند تا ویژگی‌های عمومی‌تری یاد بگیرد.

3. پیش‌پردازش ورودی‌ها

در این بخش، ورودی‌های شبکه که وضعیت‌های محیط هستند، به ترتیب از طریق لایه‌های مختلف پردازش می‌شوند:

- ابتدا از لایه اول **Dense** عبور می‌کنند.
- در صورت فعال بودن **Batch Normalization** و **Dropout**، این لایه‌ها به ترتیب بر روی داده‌ها اعمال می‌شوند.
- پس از پردازش در لایه دوم **Dense**، دوباره نرمال‌سازی و در صورت لزوم، **Dropout** اعمال می‌شود.
- در نهایت، خروجی نهایی به وسیله لایه **v** تولید می‌شود که یک تخمین از ارزش وضعیت است.

4. خروجی شبکه

خروجی شبکه یک مقدار تک‌بعدی است که نشان‌دهنده تخمین ارزش وضعیت ورودی است. این تخمین برای انتخاب سیاست‌های بهینه در الگوریتم‌های یادگیری تقویتی مورد استفاده قرار می‌گیرد.

توضیح عملکرد شبکه ActorNetwork

شبکه **ActorNetwork**وظیفه انتخاب سیاست در الگوریتم‌های یادگیری تقویتی را بر عهده دارد. این شبکه به‌طور خاص برای یادگیری سیاست‌های پیوسته در مسائل کنترل، مانند سیستم تعلیق خودرو، طراحی شده است. در این شبکه، هدف این است که یک توزیع نرمال از اقدامات (actions) را برای هر وضعیت (state) محاسبه کنیم، به‌طوری که مدل بتواند از آن برای انتخاب اقدام بهینه استفاده کند.

1. ساختار شبکه

شبکه **ActorNetwork**شامل چندین لایه **Dense**است که ورودی وضعیت‌ها را پردازش می‌کند:

- **fc1**:اولین لایه **Dense**با ابعاد **fc1_dims**برای پردازش ویژگی‌های وضعیت و یادگیری الگوهای غیرخطی.
- **fc2**:دومین لایه **Dense**با ابعاد **fc2_dims**برای افزایش ظرفیت شبکه و انجام پردازش‌های پیچیده‌تر.
- **mu**:لایه‌ای که میانگین توزیع نرمال (μ) را برای انتخاب اقدام پیش‌بینی می‌کند.
- **sigma**:لایه‌ای که انحراف معیار توزیع نرمال (σ) را محاسبه می‌کند، که میزان عدم قطعیت یا پراکندگی در انتخاب اقدامات را تعیین می‌کند.

2. محاسبه میانگین و انحراف معیار

شبکه ابتدا ورودی وضعیت را از طریق لایه‌های **Dense**عبور می‌دهد و سپس میانگین (mu) و انحراف معیار (sigma) را به‌طور مستقل محاسبه می‌کند:

- **mu**:تخمین میانگین توزیع احتمال اقدامات.
- **sigma**:تخمین انحراف معیار توزیع، که میزان پراکندگی و تنوع در انتخاب اقدامات را تعیین می‌کند.

برای جلوگیری از ناپایداری عددی، انحراف معیار **sigma**با استفاده از تابع **exp**و مقدار **noise**(برای اطمینان از مثبت بودن آن) محاسبه می‌شود و سپس با **clip**محدود می‌شود تا از مقادیر خارج از محدوده جلوگیری شود.

3. نمونه‌برداری از توزیع نرمال(Sampling)

در این بخش، دو روش برای نمونه‌برداری از توزیع نرمال پیش‌بینی‌شده وجود دارد:

- **Reparameterization Trick**: این تکنیک به الگوریتم اجازه می‌دهد که از مشتقات استفاده کرده و به‌طور کارآمدتری آموزش داده شود. در این روش، با استفاده از **epsilon** (یک متغیر تصادفی با توزیع نرمال) و انحراف معیار **sigma**، اقدام‌ها محاسبه می‌شوند.
- **Sampling for Inference**: در این روش، نمونه‌ها به‌طور مستقیم از توزیع نرمال با میانگین **mu** و انحراف معیار **sigma** گرفته می‌شوند.

4. اعمال tanh و محاسبه احتمال لاگ (Log Probability)

برای محدود کردن دامنه اقدامات به بازه معین، از تابع **tanh** استفاده می‌شود. این اطمینان می‌دهد که اقدامات در دامنه محدود و متناسب با مقدار **max_action** قرار بگیرند.

همچنین، برای محاسبه **log probability** اقدام‌ها، از **log** احتمال توزیع نرمال استفاده می‌شود. سپس یک اصلاح کوچک با استفاده از **tanh** انجام می‌شود تا احتمال لاگ درست‌تری محاسبه شود.

5. خروجی شبکه

- **Action**: اقدام انتخاب‌شده بر اساس میانگین و انحراف معیار تخمینی شبکه.
- **Log Probabilities**: احتمال لاگ اقدامات که برای محاسبات بازگشتی در الگوریتم‌های یادگیری تقویتی مورد استفاده قرار می‌گیرد (مثل به‌روزرسانی سیاست).

توضیح عملکرد کلاس Agent

کلاس **Agent** وظیفه مدیریت یادگیری و تصمیم‌گیری در یک محیط یادگیری تقویتی را بر عهده دارد. این کلاس برای پیاده‌سازی الگوریتم **Soft Actor-Critic (SAC)** طراحی شده است که از چندین

شبکه عصبی برای یادگیری سیاست، ارزیابی ارزش، و به‌روزرسانی پارامترها استفاده می‌کند. در اینجا، به طور مفصل توضیح می‌دهیم که هر بخش از این کلاس چگونه عمل می‌کند.

1. مقداردهی اولیه و تنظیمات

کلاس **Agent** در ابتدا با مقادیر پیش‌فرض تنظیم می‌شود که شامل موارد زیر است:

- **alpha** و **beta**: این مقادیر نرخ یادگیری برای شبکه‌های **actor** و **critic** را تعیین می‌کنند.
- **gamma**: نرخ تخفیف برای پاداش‌های آینده.
- **tau**: پارامتر برای به‌روزرسانی شبکه‌های هدف.(target networks)
- **memory**: حافظه برای ذخیره تجربیات، که با استفاده از کلاس **ReplayBuffer** پیاده‌سازی شده است.
- **actor, critic, value networks**: چهار شبکه عصبی برای **actor**، **critic_1**، **critic_2** و **value** تعریف می‌شوند.

2. مقداردهی شبکه‌ها

- شبکه **actor**: این شبکه برای پیش‌بینی سیاست (یعنی انتخاب اقدام‌ها) استفاده می‌شود. شبکه **actor** با توجه به وضعیت، توزیع احتمال اقدامات را تولید می‌کند.
- شبکه‌های **critic_1** و **critic_2**: این شبکه‌ها برای ارزیابی کیفیت اقدامات انتخاب‌شده توسط سیاست استفاده می‌شوند. این ارزیابی‌ها در حقیقت **Q-values** هستند که نشان‌دهنده ارزش یک اقدام خاص در یک وضعیت خاص است.
- شبکه **value**: این شبکه برای تخمین ارزش کل وضعیت‌ها استفاده می‌شود. به‌طور خاص، این شبکه تخمین می‌زند که یک وضعیت چقدر مطلوب است.
- شبکه‌های هدف (**target networks**): این شبکه‌ها نسخه‌های کپی از شبکه‌های اصلی هستند که به‌طور آهسته به‌روزرسانی می‌شوند تا نوسانات در یادگیری را کاهش دهند.

3. به‌روزرسانی پارامترهای شبکه‌ها

شـبکه‌های **target value**و **target critic**با اسـتفاده از پارامتر **tau**به‌طور آهسـته به‌روزرسـانی می‌شـوند. این به‌روزرسـانی‌ها با اسـتفاده از میانگین ترکیبی از وزن‌های شـبکه‌های اصـلی و هدف انجام می‌شوند.

4. انتخاب اقدام

شـبکه **actor**برای هر وضـعیت اقدام مناسـب را پیش‌بینی می‌کند. در صـورت نیاز به انتخاب تصـمیمات قطعی(deterministic) ، فقط میانگین اقدام (mu) انتخاب می‌شود. در غیر این صورت، از **sampling** نرمال برای انتخاب اقدامات استفاده می‌شود.

5. یادگیری و به‌روزرسانی شبکه‌ها

فرآیند یادگیری به‌طور عمده شامل به‌روزرسانی شبکه‌های **value**، **critic**، و **actor**است:

- **Value Network**:این شـبکه برای پیش‌بینی ارزش یک وضـعیت بر اسـاس اقدامات و-Q valuesاستفاده می‌شود. شبکه **value**به‌روزرسانی می‌شود تا خطای بین ارزش پیش‌بینی‌شده و هدف ارزشی جدید به حداقل برسد.
- **Critic Networks**:شـبکه‌های **critic_1**و **critic_2**برای تخمین Q-values اسـتفاده می‌شوند. این شبکه‌ها به‌روزرسانی می‌شوند تا خطای میانگین مربعی بین پیش‌بینی‌های فعلی و اهداف جدید) حاصل از پاداش‌ها و تخمین‌های شبکه (value به حداقل برسد.
- **Alpha**:پارامتر) **alpha**که با اسـتفاده از **log_alpha**نگهداری می‌شـود) برای تنظیم دمای آنتروپی سـیاسـت اسـتفاده می‌شـود. این پارامتر تأثیرگذار در میزان جسـتجو و تصـادفی بودن اقدامات است. شبکه **alpha**به‌طور مجزا به‌روزرسانی می‌شود.

6. مدیریت تاریخچه‌ها

کلاس **Agent**اطلاعات مختلفی از روند یادگیری مانند خسـارت‌های شـبکه‌هاactor) ، critic، (value، ضـرر آلفا، و پیش‌بینی‌های مختلف شـبکه‌ها را ذخیره می‌کند تا در فرآیندهای بعدی به تحلیل و بررسـی عملکرد الگوریتم پرداخته شود.

7. فرآیند یادگیری

فرآیند یادگیری شامل مراحل زیر است:

- **یادگیری شبکه value**: ابتدا با استفاده از تجربیات از حافظه، هدف‌های **value** محاسبه می‌شوند و سپس شبکه **value** با استفاده از این اهداف به‌روزرسانی می‌شود.
- **یادگیری شبکه‌های critic**: سپس شبکه‌های **critic_1** و **critic_2** با استفاده از اهداف جدید به‌روزرسانی می‌شوند.
- **یادگیری شبکه alpha**: این شبکه برای تنظیم مقدار **alpha** به‌طور مستقل به‌روزرسانی می‌شود تا آنتروپی سیاست بهینه باقی بماند.

8. خروجی و ذخیره‌سازی تجربیات

تجربیات) وضعیت‌ها، اقدامات، پاداش‌ها، وضعیت‌های جدید و (done با استفاده از متد **remember** در حافظه ذخیره می‌شوند تا در مراحل بعدی از آن‌ها برای یادگیری استفاده شود.

توضیح کلاس VehicleSuspensionEnv

کلاس **VehicleSuspensionEnv** یک محیط سفارشی برای شبیه‌سازی سیستم تعلیق خودرو در **OpenAI Gym** است. این محیط شامل ویژگی‌هایی است که به شما امکان می‌دهد سیستم تعلیق خودرو را شبیه‌سازی کرده و از آن برای یادگیری تقویتی استفاده کنید. در اینجا جزئیات هر بخش از این کلاس آورده شده است:

1. پارامترهای سیستم

در ابتدا، پارامترهای سیستم تعلیق تعریف شده‌اند. این پارامترها شامل مقادیر مختلفی هستند که رفتار سیستم تعلیق خودرو را شبیه‌سازی می‌کنند:

- **ms**: جرم قسمت‌های کشیده‌شده(sprung mass)
- **mus**: جرم بخش‌های غیرفعال(unsprung mass)
- **kf, kc, cs, cf, cc**: ضرایب سختی و میرایی مختلف سیستم.
- **kt_nl, ks_nl**: سختی‌های غیرخطی تایر و فنر.

2. فضای اقدامات(Action Space)

فضای اقدامات یک **فضای پیوسته** است که در آن یک مقدار بین -1 و 1 می‌تواند وارد شود. این مقدار با ضرب در 1000 برای اعمال نیروی کنترل به سیستم تغییر مقیاس می‌یابد.

3. فضای وضعیت(Observation Space)

فضای وضعیت 8 بعدی است که وضعیت‌های مختلف سیستم تعلیق را نمایان می‌سازد. این 8 بعد به ترتیب شامل:

- **x_us, dx_us**: موقعیت و سرعت بخش‌های غیرفعال.
- **x_s, dx_s**: موقعیت و سرعت بخش‌های کشیده‌شده.
- **x_f, dx_f**: موقعیت و سرعت فریم.
- **x_c, dx_c**: موقعیت و سرعت کوسن صندلی.

4. متد step

متد **step** برای یک گام از محیط است که در آن نیروی کنترل (بر اساس ورودی) اعمال می‌شود و وضعیت جدید سیستم محاسبه می‌شود. محاسبات مربوط به شتاب‌ها بر اساس معادلات غیرخطی صورت می‌گیرد و وضعیت جدید با استفاده از **عددگیری عددی** (numerical integration)به دست می‌آید.

- **محاسبه شتاب‌ها** :شتاب‌ها برای بخش‌های مختلف سیستم (مانند بخش‌های کشیده‌شده، بخش‌های غیرفعال و فریم) بر اساس نیروها و سختی‌ها محاسبه می‌شوند.
- **به‌روزرسانی وضعیت** :وضعیت سیستم با استفاده از شتاب‌های محاسبه‌شده به‌روزرسانی می‌شود.
- **مقیاس‌بندی وضعیت** :وضعیت به مقیاس موردنظر برای آموزش یادگیری تقویتی تبدیل می‌شود.
- **پاداش :(Reward)** پاداش به‌طور منفی محاسبه می‌شود تا به مدل کمک کند تا رفتار مطلوبی پیدا کند. این پاداش بر اساس معیارهای مختلف مانند شتاب‌ها، اختلاف موقعیت‌ها، و اعمال نیروی کنترل محاسبه می‌شود.
- **پایان :(Done)** در صورتی که شرایط خاصی (مانند موقعیت‌های خاص بخش‌های مختلف سیستم) برقرار شود، فرآیند به پایان می‌رسد.

5. متد reset

متد **reset** برای راه‌اندازی مجدد محیط و تنظیم وضعیت اولیه به کار می‌رود. این وضعیت اولیه معمولاً صفر است و از آن به‌عنوان نقطه شروع استفاده می‌شود.

6. متد render

متد **render** برای نمایش گرافیکی محیط استفاده می‌شود. در اینجا این متد پیاده‌سازی نشده است، اما معمولاً در شبیه‌سازی‌های پیچیده‌تر برای نمایش بصری وضعیت محیط به کار می‌رود.

7. کاربرد

این محیط برای استفاده در الگوریتم‌های یادگیری تقویتی طراحی شده است، به‌ویژه در الگوریتم‌های کنترل با بازخورد، مانند **Soft Actor-Critic (SAC)** که قبلاً در کد شما به آن اشاره شده است. با استفاده از این محیط، مدل می‌تواند سیاست بهینه‌ای برای کنترل سیستم تعلیق خودرو بیاموزد و به‌طور مؤثری به تعامل با محیط بپردازد.

آموزش شبکه

در این بخش، یک حلقه آموزشـــی برای یک عامل یادگیری تقویتی طراحی شـــده اســـت که در محیط شبیه‌سازی سیستم تعلیق خودرو فعالیت می‌کند. هدف این حلقه، آموزش عامل برای پیدا کردن سیاست بهینه‌ای اســـت که به آن اجازه می‌دهد تا نیروی کنترل مناســـبی را به ســـیســـتم اعمال کرده و پاداش بیشتری دریافت کند. مراحل آموزشی به‌صورت زیر انجام می‌شود:

1. **شروع هر قسمت:(Episode)**

- در ابتدای هر قسمت، وضعیت محیط (state) با استفاده از متد **reset**بازیابی می‌شود.
- وضعیت اولیه به عامل داده می‌شود و قسمت شروع می‌شود.

2. **عملیات داخل قسمت:(Episode)**

- در هر گام از قسمت، عامل یک اقدام (action) انتخاب می‌کند. این اقدام بر اساس سیاستی است که عامل در طول زمان یاد می‌گیرد.
- عامل به محیط ارسال می‌کند که چه اقدامی انجام دهد و سپس محیط وضعیت جدید، پاداش و وضعیت اتمام را باز می‌گرداند.
- عامل این اطلاعات را ذخیره می‌کند تا در آینده از آن برای به‌روزرسانی شبکه‌های عصبی خود استفاده کند.

3. **یادگیری عامل:**

- پس از انجام هر گام، عامل از داده‌های ذخیره‌شده استفاده می‌کند تا شبکه‌های عصبی خود (که نماینده سیاست و ارزش‌ها هستند) را به‌روزرسانی کند.
- یادگیری شامل محاسبه و به‌روزرسانی ضررهای مربوط به شبکه‌های مختلف (مقدار، منتقدها و آلفا) است.

4. **محاسبه پاداش:**

- پاداش در هر گام از قسمت محاسبه می‌شود و به عامل داده می‌شود. این پاداش از ترکیب فاکتورهای مختلف مانند انحرافات از مقادیر هدف، شتاب‌ها و میزان نیروی کنترل است.

5. **پیگیری تاریخچه عملکرد:**

- برای ارزیابی پیشرفت عامل، نمره هر قسمت ذخیره می‌شود.

- به‌طور دوره‌ای (هر 5 قسمت)، میانگین نمرات 100 قسمت آخر محاسبه می‌شود تا عملکرد عامل در طول زمان ارزیابی شود.

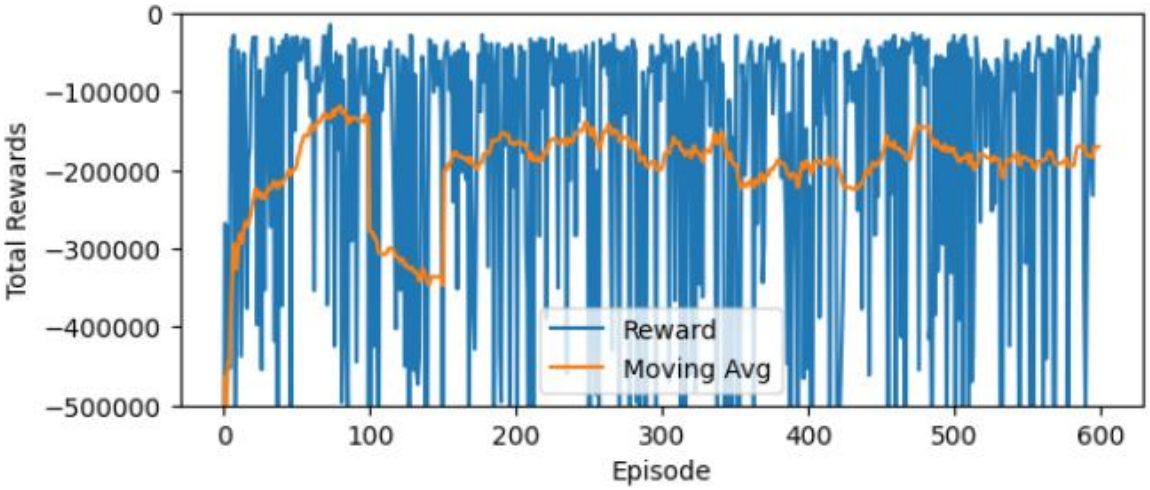
6. **پایان قسمت:**

- قسمت زمانی به پایان می‌رسد که شرایط خاصی مانند اتمام زمان یا رسیدن به وضعیت غیرمطلوب (مثل انحراف‌های بیش از حد از مقادیر هدف) محقق شود.

هدف آموزش:

هدف این فرایند، بهینه‌سازی سیاست عامل است تا با کاهش پاداش منفی (که ناشی از شتاب‌های بالا، انحرافات از هدف، و نیروی کنترل زیاد است) سیستم تعلیق خودرو به‌طور مؤثری کنترل شود. این مدل به مرور زمان یاد می‌گیرد که بهترین اقدامات را در شرایط مختلف برای کاهش پاداش منفی و بهبود عملکرد سیستم انتخاب کند.

نتایج آموزش



شکل 5-1 بررسی پاداش و حرکت در طی آموزش

همانطور که در نمودار فوق مشاهده میکنید در طول آموزش میزان پاداش ها بشدت نوسانی میباشند اما حرکت به میزان نسبتا مناسبی در برخی اپیزود ها پیشرفت داشته است

دلیل نوسانات شدید در میزان پاداش‌ها و پیشرفت نسبی در برخی اپیزودها می‌تواند به عوامل مختلفی برگردد که به روند یادگیری عامل و پیچیدگی محیط مرتبط هستند. این نوسانات و پیشرفت‌ها ممکن است ناشی از موارد زیر باشند:

1. **آغاز آموزش و کاوش در محیط:**

- در مراحل اولیه آموزش، عامل بیشتر در حال کاوش در محیط است تا یادگیری یک سیاست بهینه. در این دوره، عامل ممکن است تصمیمات اشتباهی بگیرد که منجر به پاداش‌های منفی و نوسانات بالا در پاداش‌ها شود.

- در این مرحله، عامل هنوز استراتژی مناسبی برای مواجهه با محیط پیدا نکرده است و در نتیجه ممکن است در برخی اپیزودها پاداش‌های خیلی پایین یا حتی منفی دریافت کند. این امر باعث نوسانات زیاد در نمودار پاداش‌ها می‌شود.

2. آموزش با سیاست‌های تصادفی:

- عامل در ابتدا با سیاست تصادفی اقدام می‌کند، که منجر به پاداش‌های متغیر و نوسانات شدید در هر اپیزود می‌شود. برخی اپیزودها ممکن است به‌طور تصادفی بهتر از دیگران عمل کنند و منجر به پاداش‌های بالاتر شوند، در حالی که در برخی دیگر عملکرد ضعیفی از خود نشان می‌دهد.
- این پدیده باعث ایجاد نوسانات در پاداش‌ها می‌شود و در عین حال، عوامل تصادفی می‌توانند باعث ظهور نتایج تصادفی و غیرقابل پیش‌بینی شوند.

3. الگوریتم یادگیری و زمان‌بندی به‌روزرسانی‌ها:

- الگوریتم‌های یادگیری تقویتی معمولاً از به‌روزرسانی‌های تدریجی استفاده می‌کنند. این به‌روزرسانی‌ها در ابتدا ممکن است خیلی کوچک و ناپایدار باشند و نوسانات زیادی در پاداش‌ها ایجاد کنند.
- یادگیری در ابتدا ممکن است در دنیای واقعی به کندی پیش برود، زیرا عامل هنوز بهینه‌ترین سیاست را پیدا نکرده است. تغییرات کوچک در پارامترهای شبکه عصبی و اشتباهات جزئی در انتخاب‌های اولیه می‌توانند باعث نوسانات در پاداش‌ها شوند.

4. چالش‌های محیطی و پیچیدگی دینامیک‌های سیستم:

- محیط پیچیده‌ای مانند سیستم تعلیق خودرو می‌تواند موجب ایجاد نوسانات شود. به دلیل وجود پارامترهای غیرخطی و پیچیدگی‌های فیزیکی مانند شتاب‌ها، نیروهای کنترل و تعاملات پیچیده میان اجزاء مختلف سیستم، عامل ممکن است نتواند در همه شرایط عملکرد یکنواختی داشته باشد.

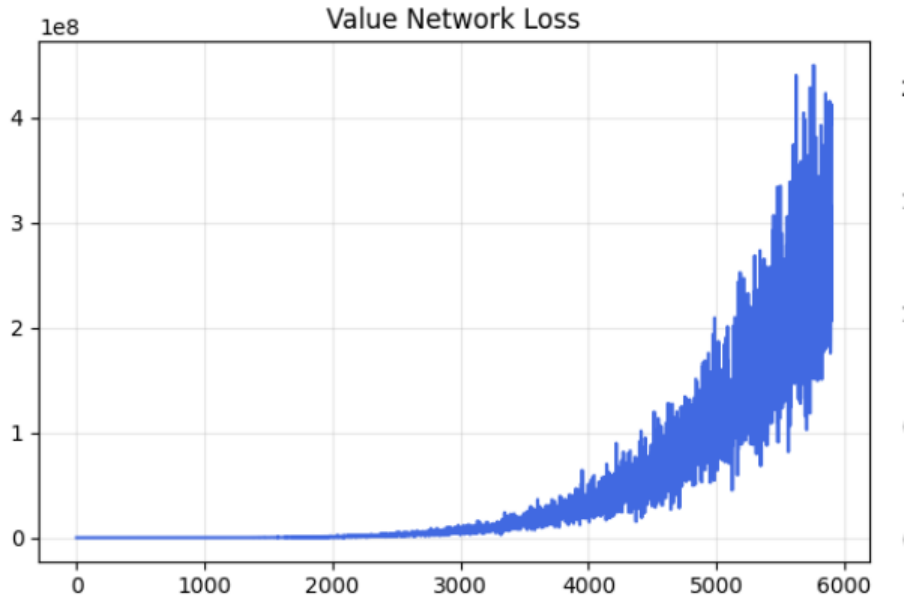
- در برخی اپیزودها، به‌ویژه زمانی که سیستم نیاز به واکنش سریع به تغییرات دارد، عامل ممکن است به‌طور غیرمنتظره‌ای به نتایج مثبت دست یابد و در برخی دیگر، به دلیل محدودیت‌های مدل و رفتار غیرقابل پیش‌بینی سیستم، عملکرد ضعیف‌تری داشته باشد.

5. نوسانات در پاداش به دلیل پاداش‌دهی پیچیده:

- پاداش‌دهی که بر اساس معیارهایی مانند انحرافات از هدف یا شتاب‌های بالا صورت می‌گیرد، ممکن است در ابتدا باعث نوسانات زیاد شود. به‌ویژه اگر عامل هنوز سیاست مناسبی برای تعادل بین این معیارها پیدا نکرده باشد.
- این نوسانات می‌توانند در اثر عدم تطابق کامل با اهداف پاداش‌دهی در آغاز آموزش رخ دهند. برای مثال، ممکن است عامل در برخی اپیزودها در تلاش برای کاهش شتاب‌های بزرگ، به نتایج بهتری دست یابد، در حالی که در اپیزودهای دیگر به دلیل تمرکز بیش از حد بر روی جنبه‌های خاص سیستم، از عملکرد بهینه دور شود.

6. پیشرفت‌های تدریجی در یادگیری:

- با گذر زمان، عامل شروع به یادگیری و بهبود سیاست خود می‌کند، اما این پیشرفت‌ها به‌طور یکنواخت اتفاق نمی‌افتد. در برخی اپیزودها، پیشرفت قابل توجهی در یادگیری سیاست ایجاد می‌شود که موجب بهبود پاداش‌ها می‌شود.
- در این مراحل، عامل ممکن است درک بهتری از محیط پیدا کرده و عملکردش را به‌طور چشمگیری بهبود دهد، که باعث می‌شود در برخی از اپیزودها پیشرفت قابل توجهی مشاهده شود.



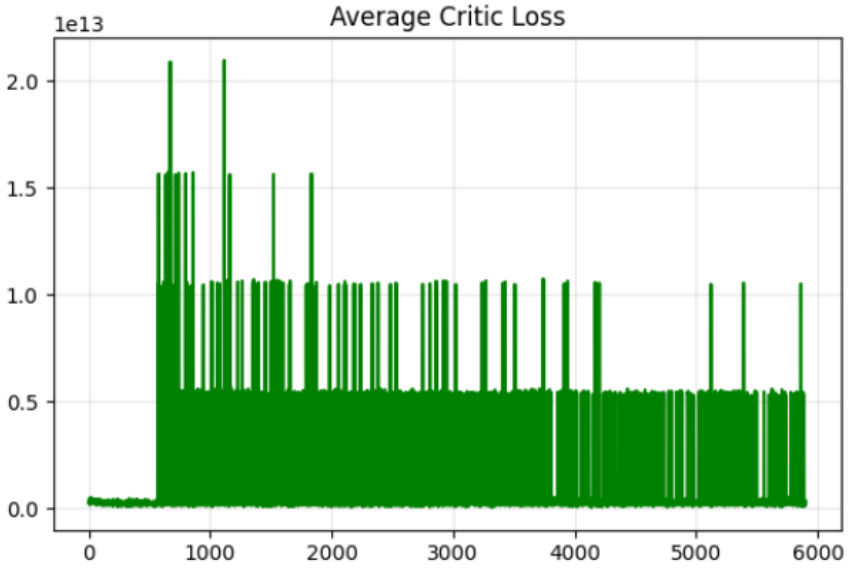
شکل 5-2 میزان هزینه شبکه

نمودار هزینه (Value Loss) نشان می‌دهد که میزان خطای پیش‌بینی ارزش وضعیت‌ها به‌طور نامنظم و ناپایدار افزایش می‌یابد. این افزایش ناپایداری ممکن است به دلایل متعددی رخ دهد:

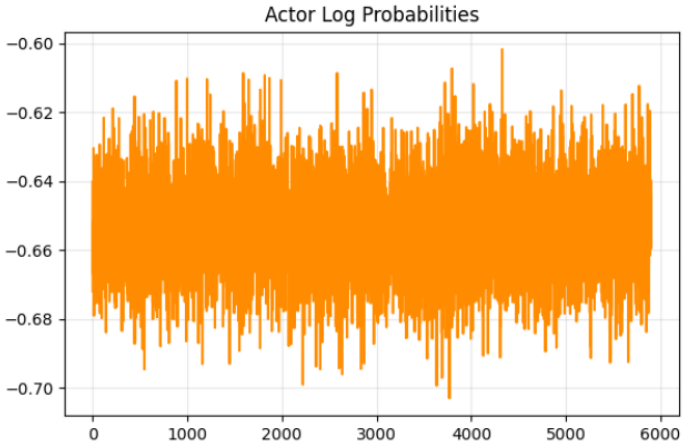
- **عدم ثبات در یادگیری :**در مراحل اولیه یا در نقاطی که عامل هنوز به سیاست بهینه دست نیافته، تغییرات ناگهانی در وزن‌های شبکه ممکن است باعث افزایش موقت هزینه شود. این موضوع به‌ویژه در سیستم‌های پیچیده با دینامیک‌های غیرخطی مشهود است.
- **تنوع داده‌های ورودی :**با تغییر وضعیت‌ها و تجربیات متفاوت در محیط، مدل ارزش ممکن است با توزیع‌های متنوعی از داده مواجه شود که در نتیجه به‌روزرسانی‌های شبکه به‌طور نامنظم انجام می‌شود و هزینه به صورت ناپایدار تغییر می‌کند.

- **تنظیمات نرخ یادگیری و پارامترها :**استفاده از نرخ یادگیری نسبتاً بالا یا به‌روزرسانی‌های ناگهانی پارامترها می‌تواند باعث شود که مدل به بهبود تدریجی نرسد و هزینه به طور ناگهانی افزایش پیدا کند.
- **اثر اکتشاف (Exploration):** در فرآیند یادگیری، عامل ممکن است به دلیل استراتژی‌های کاوش، گاهی اقداماتی انجام دهد که باعث می‌شود مقدار پیش‌بینی شده ارزش با هدف واقعی فاصله بگیرد. این موضوع می‌تواند به صورت موقت باعث افزایش هزینه شود.

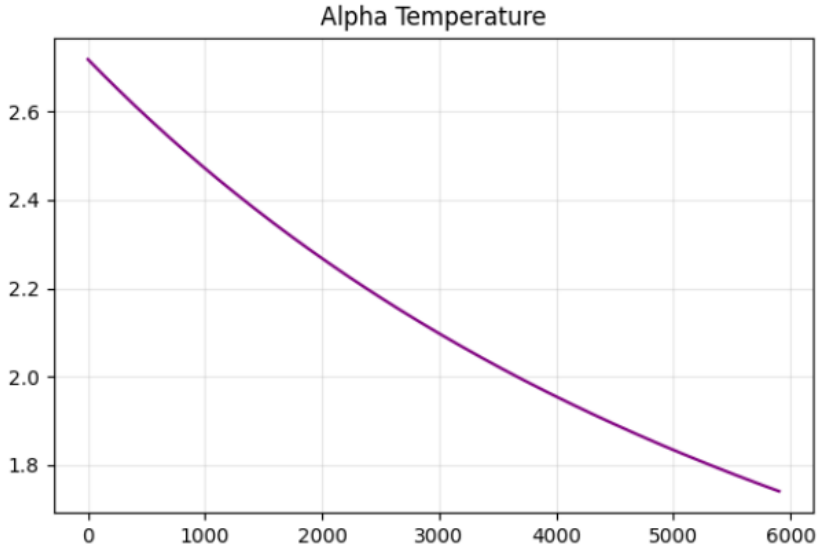
در مجموع، این نوسانات در نمودار هزینه بیانگر چالش‌های موجود در تخمین صحیح ارزش وضعیت‌ها و بهبود تدریجی مدل در مواجهه با محیط‌های پیچیده هستند. به مرور زمان و با بهبود سیاست عامل، انتظار می‌رود که این نوسانات کاهش یافته و هزینه به یک سطح پایدارتر نزدیک شود.



شکل 5-3 میانگین critic loss



شکل 5-4 میزان احتمالات اکتور



شکل 5-5 میزان تغییر alpha temperature

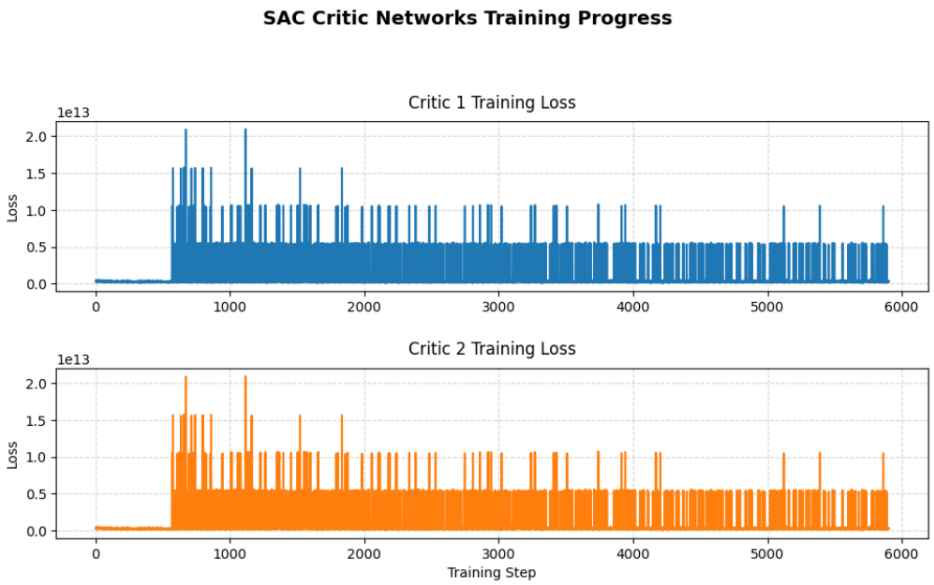
در الگوریتمSAC ، پارامتر دمای آلفا (α) نقشی کلیدی در تنظیم تعادل بین اکتشاف (exploration) و بهره‌برداری (exploitation) ایفا می‌کند. کاهش یکنواخت این دما در نمودار نشـــان‌دهنده روند تطبیق خودکار این پارامتر بر اساس سیاست یادگرفته‌شده است. به عبارت دیگر:

تنظیم خودکار آنتروپی :در ابتدای آموزش، به دلیل نیاز به اکتشـاف گسـترده، دمای آلفا ممکن اسـت بالاتر باشد تا عامل بتواند با سیاست تصادفی، محیط را به خوبی کاوش کند.

کاهش تدریجی با بهبود سـیاسـت :با پیشـرفت یادگیری، عامل به تدریج سـیاسـت خود را به سـمت تصـــمیم‌گیری‌های هدفمندتر و قطعی‌تر هدایت می‌کند. در این حالت، مقدار آنتروپی (عدم قطعیت) کاهش می‌یابد و نیاز به میزان بالای اکتشاف کمتر می‌شود.

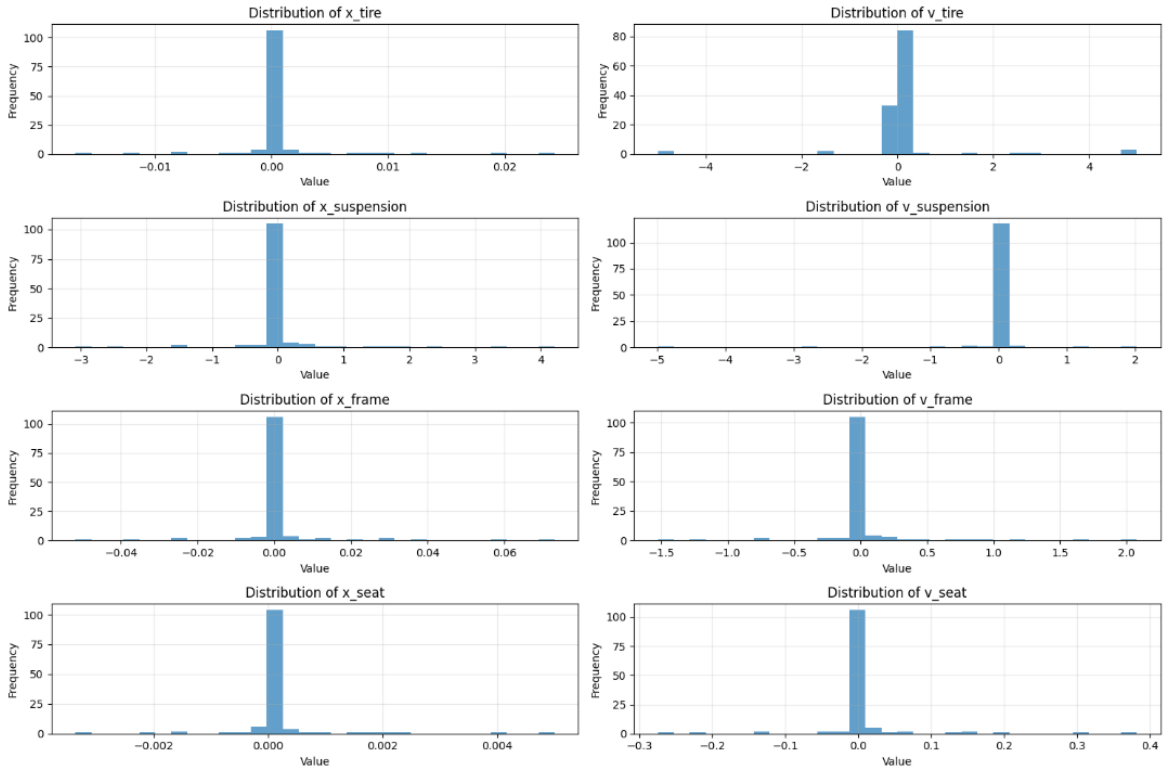
تنظیم دما به سـمت هدف آنتروپی :در فرآیند به‌روزرسـانی آلفا، اگر مقدار آنتروپی سـیاسـت از مقدار هدف (target entropy) بالاتر باشـد، به‌روزرسـانی گرادیان منجر به کاهش دمای آلفا می‌شـود تا سـطح آنتروپی به مقدار مطلوب نزدیک شود.

بنابراین، کاهش یکنواخت دمای آلفا نشـان‌دهنده‌ی این اسـت که عامل به مرور زمان به سـیاسـتی نزدیک می‌شود که سطح آنتروپی آن مطابق با هدف تعیین‌شده است و نیازی به اکتشاف گسترده کمتری دارد. این روند، نشانه‌ای از پایداری و تثبیت سیاست یادگرفته‌شده در طول آموزش محسوب می‌شود.



شکل 5-6 تغییرات critic loss

در نمودار critic loss می‌بینیم که هزینه شبکه‌های منتقد نسبت به ابتدای آموزش کاهش یافته است. این بهبود نشان‌دهنده افزایش دقت تخمین‌های Q-value است؛ یعنی شبکه‌های منتقد توانسته‌اند به مرور زمان با استفاده از تجربیات جمع‌آوری‌شده، ارزش وضعیت-اقدام را بهتر پیش‌بینی کنند. کاهش critic lossبیانگر تثبیت و بهبود عملکرد سیستم ارزیابی و در نتیجه پیشرفت کلی در بهینه‌سازی سیاست عامل است.



شکل 5-7 مانیتور فرکانس هر حالت

این نمودار نشان می‌دهد که مقادیر وضعیت‌ها عمدتاً حول نقطه تعادل (equilibrium) یا فرکانس تعادل متمرکز هستند. به عبارت دیگر، سیستم در طول زمان به حالت پایدار نزدیک شده و نوسانات زیاد کاهش یافته‌اند. این موضوع بیانگر کارایی بهبود یافته سیستم کنترل است، به گونه‌ای که عامل قادر است رفتار سیستم را به نحوی تنظیم کند که وضعیت‌ها حول نقطه تعادل باقی بمانند و از تغییرات ناخواسته و شدید جلوگیری شود.

نتیجه گیری کلی

در این آزمایش، با بهره‌گیری از الگوریتم یادگیری تقویتیSAC ، عامل قادر شد تا کنترل نسبتاً خوبی بر سیستم تعلیق خودرو اعمال کند. نتایج به دست آمده نشان می‌دهد که به مرور زمان، وضعیت‌های سیستم حول نقطه تعادل متمرکز شده و نوسانات زیاد کاهش یافته‌اند. کاهش هزینه شبکه‌های منتقد و شبکه ارزش، بیانگر بهبود دقت تخمین عملکرد سیستم و تثبیت سیاست یادگرفته‌شده عامل است.

از یک سو، این نتایج نشان می‌دهد که عامل توانسته است با استفاده از تکنیک‌های پیشرفته یادگیری تقویتی، به درک بهتری از دینامیک سیستم دست یابد و کنترل مناسبی را ارائه دهد. از سوی دیگر، باید توجه کرد که سیستم تعلیق خودرو دارای پیچیدگی‌های بسیار بالایی است؛ به دلیل وجود دینامیک‌های غیرخطی، اثرات نیروهای مختلف و تعامل پیچیده اجزاء مختلف سیستم، دستیابی به یک کنترل کاملاً بهینه چالشی جدی محسوب می‌شود.

برای بهبود عملکرد کنترل و رسیدن به نتایج مطلوب‌تر، می‌توان از راهکارهای زیر استفاده کرد:

- **بهینه‌سازی ساختار شبکه‌های عصبی :**استفاده از معماری‌های عمیق‌تر یا اعمال تکنیک‌های پیشرفته تنظیم وزن می‌تواند به بهبود دقت تخمین‌ها و کاهش خطاهای یادگیری کمک کند.
- **تنظیم دقیق هایپرپارامترها :**بهینه‌سازی نرخ‌های یادگیری، پارامترهای آلفا (دمای آنتروپی) و پارامترهای مرتبط با به‌روزرسانی شبکه‌های هدف می‌تواند باعث پایداری بیشتر در فرایند آموزش شود.
- **استفاده از تکنیک‌های پیشرفته کنترل :**ترکیب الگوریتم‌های یادگیری تقویتی با روش‌های کنترل تطبیقی یا الگوریتم‌های تکاملی می‌تواند به دستیابی به سیاست‌های کنترل بهینه‌تر و تطبیق بهتر با تغییرات دینامیکی سیستم کمک کند.
- **افزایش کیفیت داده‌های آموزشی :**جمع‌آوری تجربیات بیشتر و استفاده از استراتژی‌های پیشرفته بازپخش تجربیات می‌تواند به کاهش نوسانات و افزایش پایداری سیستم منجر شود.

در مجموع، اگرچه کنترل نسبتاً مناسبی در این آزمایش به دست آمده است، اما با توجه به پیچیدگی بیش از حد سیستم، همچنان فضای بهبود و پیشرفت وجود دارد. با اعمال راهکارهای پیشنهادی و

بهینه‌سازی بیشتر مدل، می‌توان انتظار داشت که عملکرد عامل در کنترل سیستم تعلیق به صورت چشمگیری ارتقا یابد.

فصل ششم:

جمع‌بندی و نتیجه‌گیری

نتیجه گیری روش های مختلف

در این پروژه، روش‌های هوشمند مختلف برای کنترل و عیب‌یابی سیستم تعلیق خودروی پیشرفته با استفاده از مدل‌های مختلف شبکه عصبی و الگوریتم‌های یادگیری تقویتی مورد بررسی قرار گرفتند. نتایج حاصل از هر روش به‌طور جداگانه تحلیل شده و در نهایت مقایسه‌ای دقیق بین این روش‌ها انجام خواهد گرفت.

MLP (Perceptron چندلایه)

شبکه عصبی چندلایه (MLP) به‌عنوان یک مدل ساده و قدرتمند برای شبیه‌سازی سیستم‌های پیچیده در این پروژه مورد استفاده قرار گرفت. نتایج نشان‌دهنده عملکرد بسیار خوب این مدل در شناسایی و تعقیب دینامیک سیستم تعلیق خودروی پیشرفته بود. MLP توانست به‌خوبی روابط پیچیده در داده‌ها را شبیه‌سازی کرده و دقت قابل قبولی در پیش‌بینی متغیرهای حالت سیستم ارائه دهد. همچنین، به‌دلیل ساختار ساده‌تر و نیاز به منابع محاسباتی کمتر، این روش به‌عنوان گزینه‌ای مناسب برای پیاده‌سازی سریع و کارآمد در سیستم‌های مشابه معرفی می‌شود.

RBF (شبکه عصبی شعاعی)

در استفاده از شبکه عصبی RBF برای شناسایی سیستم، نتایج حاکی از مشکلاتی در عملکرد این مدل بود. به‌ویژه، پس از اعمال مدل بر روی داده‌های سیستم تعلیق، مشاهده شد که خروجی‌های مدل در مقایسه با پیش‌بینی‌های مورد انتظار همچنان ناپایدار و غیرمطلوب بودند. این ناپایداری در عمل باعث کاهش دقت کنترل سیستم و حتی ناتوانی در شبیه‌سازی دینامیک واقعی سیستم شد. از آنجایی که سیستم تعلیق خودرو از دینامیک‌های پیچیده و غیرخطی برخوردار است، این روش قادر به شبیه‌سازی این پیچیدگی‌ها به‌درستی نبود و به‌عنوان یک گزینه ناکارآمد در نظر گرفته شد.

LSTM (شبکه‌های عصبی بازگشتی طولانی‌مدت کوتاه‌مدت)

در این بخش، از شبکه‌های عصبی LSTM برای شناسایی و رویت سیستم تعلیق استفاده شد. نتایج این مدل نشان داد که LSTM نمی‌تواند به‌طور مناسب دینامیک سیستم تعلیق خودروی پیشرفته را مدل‌سازی کند. به‌ویژه در فرآیند آموزش، میزان خطای مدل بسیار بالا بود و مدل نتوانست همگرایی لازم

را به‌دست آورد. این مشکلات ناشی از ضعف در تعمیم‌پذیری مدل و ناکارآمدی آن در تخمین متغیرهای حالت سیستم بود. به‌دلیل این نتایج ضعیف، پیشنهاد می‌شود که از روش‌های جایگزین مانند فیلتر کالمن توسعه‌یافته (EKF) یا ترکیب شبکه‌های عصبی با مدل‌های فیزیکی (Hybrid Models) استفاده شود تا دقت و پایداری بالاتری در شبیه‌سازی دینامیک سیستم حاصل گردد.

SAC (الگوریتم یادگیری تقویتی Soft Actor-Critic)

در این بخش، از الگوریتم یادگیری تقویتی SAC برای کنترل سیستم تعلیق خودرو استفاده شد. نتایج به‌دست‌آمده نشان داد که این الگوریتم توانسته است به‌طور موفقیت‌آمیزی کنترل مناسبی بر سیستم اعمال کند. عامل یادگیری تقویتی SAC در طی فرآیند آموزش توانست نوسانات سیستم را کاهش داده و به‌طور تدریجی وضعیت‌های سیستم را به نقطه تعادل نزدیک کند. با این حال، به‌دلیل پیچیدگی‌های بالای سیستم تعلیق با چهار درجه آزادی، همچنان چالش‌هایی در دستیابی به کنترل بهینه و پایدار وجود دارد. به‌طور کلی، این روش می‌تواند عملکرد بهتری داشته باشد اگر که با استفاده از تکنیک‌های پیشرفته‌تر مانند تنظیم دقیق‌هایپرپارامترها یا ترکیب الگوریتم‌های یادگیری تقویتی با روش‌های کنترل تطبیقی بهبود یابد.

مقایسه روش‌ها

معیار	MLP	RBF	LSTM	SAC
دقت	بالا	پایین	ضعیف	متوسط
پایداری سیستم کنترل	پایدار	ناپایدار	ناپایدار	نسبتا پایدار
توانایی در شبیه‌سازی پیچیدگی‌های غیرخطی	خوب	ضعیف	ضعیف	متوسط
زمان آموزش	کوتاه	متوسط	طولانی	طولانی
پایداری در طول زمان	پایدار	ناپایدار	ناپایدار	پایدار
نیاز به منابع محاسباتی	کم	متوسط	زیاد	زیاد
کاربرد در سیستم‌های پیچیده	مناسب	نامناسب	نامناسب	مناسب

جدول 6-1 مقایسه روش های مختلف

نتیجه‌گیری کلی

با توجه به نتایج به‌دست‌آمده از بررسی روش‌های مختلف، می‌توان نتیجه گرفت که برای سیستم‌های پیچیده‌ای مانند سیستم تعلیق خودرو با دینامیک‌های غیرخطی، هر یک از این روش‌ها دارای مزایا و معایب خاص خود هستند. روش MLP به‌عنوان بهترین گزینه در شبیه‌سازی و شناسایی دینامیک سیستم‌های غیرخطی پیچیده با دقت بالا و پایداری مناسب شناخته شد. الگوریتم SAC نیز در کنترل سیستم تعلیق عملکرد مطلوبی داشت، اما نیاز به بهینه‌سازی و تنظیم دقیق‌تر دارد. در مقابل، مدل‌های RBF و LSTM در شبیه‌سازی و کنترل سیستم با مشکلاتی همراه بودند که کارایی آن‌ها را به‌شدت کاهش داد.

در نهایت، پیشنهاد می‌شود برای سیستم‌های مشابه از ترکیب روش‌های مختلف یادگیری عمیق و یادگیری تقویتی استفاده شود. استفاده از معماری‌های پیشرفته‌تر، بهینه‌سازی‌های بیشتر و ترکیب روش‌های کنترل کلاسیک با یادگیری عمیق می‌تواند به دستیابی به عملکرد بهینه‌تر و پایدارتر منجر شود.

منابع و مراجع

[1] Optimization of nonlinear quarter car suspension– seat–driver model Mahesh P. Nagarkar a,b, Gahininath J. Vikhe Patil c , Rahul N. Zaware Patil

[2] Simulation and Analysis of Passive and Active Suspension System Using Quarter Car Model for Different Road Profile Abdolvahab Agharkakli , Ghobad Shafiei Sabet , Armin Barouz