



دانشگاه صنعتی امیرکبیر  
(پلی تکنیک تهران)  
دانشکده مهندسی برق

## گزارش پروژه کنترل تطبیقی خودرو خودران

نگارش  
ستایش خاصه تراش

استاد راهنما  
دکتر حیدرعلی طالبی

تیر ۱۴۰۴

## چکیده

با توجه به پیچیدگی و پویایی محیط‌های رانندگی واقعی، طراحی سامانه‌های کنترلی برای خودروهای خودران نیازمند رویکردهایی است که توانایی سازگاری با شرایط متغیر را داشته باشند. یکی از چالش‌های اصلی در این زمینه، عدم قطعیت در مدل‌سازی دقیق دینامیک وسیله نقلیه و تغییرات ناگهانی شرایط محیطی همچون وضعیت جاده، شرایط جوی و فاصله با سایر وسایل نقلیه است. در این پروژه، یک کنترل‌کننده تطبیقی برای کنترل سرعت خودروی خودران طراحی شده است که می‌تواند به‌صورت بلادرنگ با تغییرات پارامترهای سیستم و شرایط محیطی سازگار شود.

ابتدا مدل دینامیکی طولی خودرو استخراج شده و سپس با فرض عدم دسترسی دقیق به برخی پارامترها مانند جرم و مقاومت‌های حرکتی، یک شناساگر تطبیقی جهت تخمین این پارامترها طراحی شده است. در ادامه، یک قانون کنترل مبتنی بر روش تطبیقی مستقیم توسعه یافته است تا پایداری سیستم و رسیدن به سرعت مطلوب تضمین شود. نتایج شبیه‌سازی در محیط MATLAB/Simulink نشان می‌دهند که کنترل‌کننده طراحی‌شده می‌تواند در برابر تغییرات ناگهانی پارامترها و اغتشاشات خارجی عملکرد قابل قبولی داشته باشد و پاسخ سریع و بدون نوسان به تغییرات مرجع ارائه دهد.

**واژگان کلیدی:** کنترل تطبیقی، خودرو خودران، کنترل سرعت، عدم قطعیت، شناساگر تطبیقی

صفحه	فهرست مطالب
أ	چکیده.....
1	فصل اول مقدمه (دستور العمل).....
3	فصل دوم آماده سازی محیط.....
4	آماده سازی محیط.....
4	1-2- شبیه سازی در Simulink.....
4	2-2- شبیه سازی در Carla.....
6	فصل سوم شبیه سازی در Simulink.....
7	1-3- مقدمه کنترل کننده MRAC.....
7	2-3- مسئله کنترل.....
9	3-3- معرفی مدل ها.....
9	1-3-3- مدل عرضی اسمی (Lateral Nominal Model).....
10	2-3-3- مدل طولی اسمی (Longitudinal Nominal Model).....
10	3-3-3- تعریف مدل مرجع (Reference Model).....
11	4-3-3- شرایط اولیه سیستم.....
11	1-3-4-3- شرایط اولیه مدل اسمی.....
11	4-3- ساختار کنترل کننده MRAC.....
13	5-3- برآورد عدم قطعیت.....
16	6-3- نوآوری.....
18	7-3- بلوک MRAC در Simulink.....
19	9-3- نتایج.....
21	فصل چهارم شبیه سازی در Carla.....
22	1-4- طراحی مدل مرجع.....
23	2-4- مدل سیستم.....
24	3-4- شرح طراحی کنترل کننده.....
24	4-4- نوآوری (کنترل تطبیقی به همراه فازی).....
25	فصل پنجم نوآوری.....
27	فصل ششم جمع بندی و نتیجه گیری و پیشنهادات.....
29	منابع و مراجع.....

\_\_\_\_\_

\_\_\_\_\_

## فهرست علائم

### علائم لاتین

نیروی مقاومت غلتشی غیرخطی و مقاومت آیرودینامیکی	$k_f u $
اغتشاش ناشی از شیب جاده	$d_{long}(x)$
اثر بخشی سیستم گاز/ترمز (علامت آن شناخته شده است)	$B_l(x)$
سرعت طولی خودرو	$u$
ورودی کنترلی مربوط به سیستم گاز/ترمز	$u_c$

## فصل اول

### مقدمه (دستور العمل)

## مقدمه

پیشرفت‌های اخیر در حوزه هوش مصنوعی، الکترونیک و سامانه‌های کنترل، زمینه را برای توسعه خودروهای خودران فراهم کرده است. یکی از اجزای حیاتی در طراحی این خودروها، سامانه‌های کنترلی هوشمند هستند که باید بتوانند عملکرد وسیله نقلیه را در شرایط متنوع و پیش‌بینی‌ناپذیر محیطی تضمین کنند. برخلاف خودروهای سنتی که توسط راننده کنترل می‌شوند، در خودروهای خودران نیاز است که سامانه کنترلی، تصمیمات بلادرنگ اتخاذ کرده و پایداری و ایمنی سیستم را در مواجهه با عدم قطعیت‌ها و اغتشاشات خارجی حفظ نماید.

در این راستا، کنترل تطبیقی به‌عنوان یکی از روش‌های قدرتمند در کنترل سامانه‌هایی با پارامترهای نامشخص یا متغیر شناخته می‌شود. این نوع کنترلرها می‌توانند با شناسایی پارامترهای ناشناخته و اعمال اصلاحات لازم، عملکرد مطلوب سیستم را بدون نیاز به مدل دقیق حفظ کنند. از آنجایی که پارامترهایی مانند جرم خودرو، شیب جاده، نیروی مقاومت هوا و شرایط ترافیکی همواره در حال تغییر هستند، بهره‌گیری از یک کنترل‌کننده تطبیقی می‌تواند راهکاری مؤثر برای کنترل سرعت خودروهای خودران ارائه دهد.

در این پروژه، تمرکز بر طراحی یک کنترل‌کننده تطبیقی برای تنظیم سرعت خودرو هوشمند با در نظر گرفتن تغییرات محیطی و دینامیکی بوده است. هدف آن است که این کنترل‌کننده در شرایط واقعی بتواند پایداری سیستم را تضمین کرده و عملکردی ایمن و کارآمد ارائه دهد.



## فصل دوم

### آماده سازی محیط

## آماده سازی محیط

برای این پروژه دو محیط برای پیاده سازی فرضیات استفاده شده است. محیط اول Simulink می باشد که در آن با استفاده از بلوک های مربوطه شبیه سازی صورت میگیرد و فرضیات مورد آزمایش قرار می گیرند. محیط دوم نرم افزار Carla می باشد که شبیه سازی به صورت دقیق می باشد و کنترل تطبیقی به صورت کد های python مورد استفاده قرار می گیرد. به علت محدودیت های این برنامه فرضیات بیان شده بار دیگر در Simulink آزمایش می شوند.

### 1-2- شبیه سازی در Simulink

در این محیط بلوک های فرضی به صورت زیر قرار میگیرند:

1. بلوک تابع تبدیل سرعت خودرو
2. بلوک تابع تبدیل فرمان خودرو
3. بلوک نویز رندوم به عنوان شرایط محیطی که محیط خودرو را از حالت ایده آل خارج می کند.
4. ۲ ورودی به صورت منحنی که نمایانگر مسیر خودرو و چرخش ایده آل فرمان باشند.
5. خروجی که سرعت و نحوه چرخش فرمان است.

### 2-2- شبیه سازی در Carla

در این محیط، کنترل تطبیقی برای هدایت یک خودروی خودران پیاده سازی شد. شبیه ساز CARLA یک پلتفرم متن باز مبتنی بر موتور Unreal Engine است که برای آزمون و ارزیابی الگوریتم های رانندگی خودکار مورد استفاده قرار می گیرد.

ابتدا با استفاده از Python API مربوط به CARLA، اتصال به سرور شبیه ساز برقرار شد و محیط شبیه سازی شامل یک نقشه شهری، خودروی عامل (ego vehicle) و حسگرهایی مانند GPS و IMU پیکربندی شد.

سپس با تعریف یک مدل دینامیکی ساده برای خودرو، کنترلر تطبیقی بر مبنای این مدل طراحی گردید. در این کنترلر، ضرایب مدل به صورت برخط تخمین زده می شوند تا عدم قطعیت های سیستم و شرایط متغیر مسیر جبران گردد. الگوریتم تطبیقی MRAC یا برای این منظور قابل استفاده اند.

کنترلر طراحی شده در حلقه کنترلی شبیه سازی قرار گرفت. خروجی الگوریتم کنترل تطبیقی شامل فرمان هایی مانند زاویه فرمان (steering angle) و شتاب (throttle) بود که مستقیماً به خودروی شبیه سازی شده اعمال شد. در طول شبیه سازی، با ثبت داده های موقعیت و سرعت، عملکرد کنترلر تطبیقی ارزیابی گردید.

در نهایت، نتایج به صورت نمودارهایی از مسیر طی شده، خطای تعقیب مسیر و تغییرات پارامترهای تطبیقی ارائه شدند که نشان دهنده ی توانایی کنترلر در انطباق با شرایط متغیر محیطی و دینامیکی خودرو می باشد.

## فصل سوم

## شبیه سازی در Simulink

## شبیه سازی در Simulink

در این بخش به شبیه سازی در Simulink پرداخته می شود. کد های ارسالی نیز شامل گزارش کامل تر می باشد و از آوردن کد در این گزارش خودداری شده است.

### 3-1- مقدمه کنترل کننده MRAC

کنترل کننده تطبیقی برای سیستم بدین منظور تعریف می شود که سیستم در برابر تغییرات ناگهانی و غیر قابل پیش بینی مقاوم باشد یا به بیانی خود را با تغییرات وفق دهد و از عملکرد پایدار خود خارج نشود. در این سیستم فرضی تغییرات غیر قابل پیش بینی اعم از دینامیک غیر قطعی خودرو و یا تغییرات جوی می باشد. کنترل کننده هایی مانند کنترلر مقاوم و یا کنترلر های gain scheduling نیز مناسب برای همچنین سیستمی هستند اما برتری کنترلر تطبیقی در این است که gain های این کنترل کننده با تغییرات به وجود آمده در سیستم تغییر کرده و تطبیق می یابند.

کنترلر تطبیقی مدل مرجع (MRAC) نوعی کنترل کننده تطبیقی می باشد که هدف آن این است که سیستم مورد بررسی رفتاری مانند سیستمی ایده آل داشته باشد. سیستم ایده آل رفتار دلخواه را به صورت پایدار باید دنبال کند.

### 3-2- مسئله کنترل

در یک خودروی خودران، به دلیل وجود عدم قطعیت در دینامیک سیستم و تأثیر عوامل خارجی مانند تغییرات شرایط آب و هوایی، شیب جاده، یا وزش باد، نوسانات ناخواسته ای در سرعت طولی و جابجایی عرضی خودرو ایجاد می شود. این نوسانات می توانند عملکرد خودرو را مختل کرده و ایمنی آن را به خطر اندازند. هدف اصلی این پروژه، طراحی یک کنترل کننده تطبیقی مبتنی بر مدل مرجع (MRAC) است که بتواند این نوسانات را حذف کرده و رفتار دینامیکی خودرو را به گونه ای تنظیم کند که سرعت طولی و جابجایی عرضی خودرو از یک مدل مرجع از پیش تعریف شده پیروی کنند. برای دستیابی به این هدف، ابتدا یک کنترل کننده پایه طراحی می شود تا رفتار مرجع مطلوب را محقق سازد.

برای توصیف حرکات خودروی خودران، یک مدل دینامیکی غیرخطی با عدم قطعیت پیشنهاد شده است که شامل دو بخش اصلی است: دینامیک طولی و دینامیک عرضی.

### دینامیک طولی

معادله دینامیک طولی به صورت زیر تعریف می شود:

$$\dot{u} = -k_f |u| + d_{long}(x) + B_1(x)u_c$$

### دینامیک عرضی

معادله دینامیک عرضی به صورت زیر تعریف می شود:

$$\ddot{y} = d_{lat}(x) + B_2(x)\delta$$

که در آن:

$d_{lat}(x)$ : اغتشاشات عرضی (مانند وزش باد یا انحنای جاده)

$B_2(x)$ : ثربخشی سیستم فرمان (علامت آن شناخته شده است)

$y$ : جابجایی عرضی خودرو

$\delta$ : ورودی کنترلی مربوط به سیستم فرمان

این مدل ها نشان دهنده رفتار دینامیکی پیچیده خودرو در شرایط واقعی هستند که تحت تأثیر عوامل خارجی و عدم قطعیت های سیستم قرار دارند.

### مدل مرجع

برای دستیابی به رفتار مطلوب، سرعت طولی و جابجایی عرضی خودرو باید از یک مدل مرجع مرتبه دوم پیروی کنند. مدل های مرجع پیشنهادی به صورت زیر تعریف شده اند:

### مدل مرجع برای سرعت طولی

مدل مرجع برای سرعت طولی به صورت زیر است:

$$\ddot{y}_m + 4\dot{y}_m + 4y_m = 4r(t)$$

### مدل مرجع برای جابجایی عرضی

مدل مرجع برای جابجایی عرضی به صورت زیر است:

$$\ddot{y}_m + 4\dot{y}_m + 4y_m = 4r_y(t)$$

این مدل‌های مرجع رفتار مطلوب خودرو را در شرایط ایده‌آل تعریف می‌کنند و هدف کنترل‌کننده MRAC، تنظیم ورودی‌های کنترلی  $u_c$  و  $\delta$  به گونه‌ای است که رفتار واقعی خودرو به این مدل‌های مرجع نزدیک شود.

### 3-3- معرفی مدل ها

در طراحی کنترلرهای تطبیقی مبتنی بر مدل مرجع (MRAC)، تعیین یک مدل نامی برای سیستم واقعی و همچنین یک مدل مرجع پایدار و مطلوب ضروری است. در این بخش، مدل‌های اسمی و مرجع برای دینامیک‌های عرضی (Lateral) و طولی (Longitudinal) یک وسیله نقلیه تعریف شده‌اند. هدف این است که کنترلر تطبیقی رفتار سیستم را به سمت مدل مرجع سوق دهد.

#### 3-3-1- مدل عرضی اسمی (Lateral Nominal Model)

مدل عرضی به صورت یک انتگرال گیر مرتبه دوم ساده شده تعریف شده که ورودی آن زاویه فرمان  $\delta$  است:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = {}_{lat}B, \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = {}_{lat}A$$

که در آن:

- $y$  جابه‌جایی عرضی (lateral displacement)
- $v$  سرعت عرضی (lateral velocity)

خروجی سیستم، همان جابه جایی عرضی در نظر گرفته شده.

### 2-3-3- مدل طولی اسمی (Longitudinal Nominal Model)

مدل طولی نیز به صورت یک انتگرال گیر مرتبه دوم تعریف شده است که ورودی آن شتاب  $a$  است:

$$\begin{bmatrix} 0 \\ 1 \end{bmatrix} = {}_{\text{long}}B, \quad \begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix} = {}_{\text{long}}A$$

که در آن:

- $u$ : سرعت طولی (longitudinal velocity)
- مشتق آن (شتاب) به عنوان ورودی دینامیک در نظر گرفته شده است.

### 3-3-3- تعریف مدل مرجع (Reference Model)

مدل های مرجع، سیستم هایی پایدار از مرتبه دوم هستند که دینامیک مطلوب را مشخص می کنند. کنترلر تطبیقی تلاش می کند تا رفتار مدل اسمی به این مدل ها نزدیک شود.

#### مدل مرجع عرضی

مدل مرجع برای حرکت عرضی به شکل زیر تعریف شده است:

$$\begin{bmatrix} 0 \\ 4 \end{bmatrix} = {}_{m,\text{lat}}B, \quad \begin{bmatrix} 1 & 0 \\ 4- & 4- \end{bmatrix} = {}_{m,\text{lat}}A$$

این مدل دارای قطب های با پاسخ سریع و میرایی مناسب است که پایداری سیستم را تضمین می کند.

#### ۲.۲ مدل مرجع طولی

برای دنبال کردن سرعت طولی مطلوب، از مدل زیر استفاده شده است:

$$\begin{bmatrix} 0 \\ 2 \end{bmatrix} = {}_{m,\text{long}}B, \quad \begin{bmatrix} 1 & 0 \\ 2- & 2- \end{bmatrix} = {}_{m,\text{long}}A$$



این مدل نیز یک سیستم پایدار مرتبه دوم با پاسخ نرم و مناسب برای کنترل سرعت است.

### 3-3-4- شرایط اولیه سیستم

#### 3-3-4-1- شرایط اولیه مدل اسمی

در زمان  $t=0$ ، مقادیر متغیرهای حالت برای مدل‌های اسمی به صورت زیر مقداردهی شده‌اند:

• عرضی:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} {}_0y \\ {}_0v \end{bmatrix} = x_{lat}(0)$$

• طولی:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = \begin{bmatrix} {}_0u \\ 0 \end{bmatrix} = x_{long}(0)$$

#### شرایط اولیه مدل مرجع

مدل‌های مرجع نیز با شرایط اولیه صفر مقداردهی شده‌اند:

$$\begin{bmatrix} 0 \\ 0 \end{bmatrix} = x_{m,long}(0) \quad , \quad \begin{bmatrix} 0 \\ 0 \end{bmatrix} = x_{m,lat}(0)$$

### 3-4- ساختار کنترل کننده MRAC

هدف از طراحی کنترل کننده MRAC، تنظیم ورودی کنترل به گونه‌ای است که حالت‌های سیستم واقعی، حالت‌های مدل مرجع را دنبال کنند. در این ساختار، یک مدل مرجع طراحی می‌شود که رفتار مطلوب سیستم را مشخص می‌کند، و سپس با استفاده از قانون کنترل تطبیقی، تلاش می‌شود تا خروجی‌های سیستم به مدل مرجع نزدیک شوند.

معادله کلی کنترل در MRAC به صورت زیر تعریف می شود:

$$u(t) = -K_x x(t) + K_r r(t) + u_{ad}(t)$$

که در آن:

- $x(t)$ : بردار حالت سیستم واقعی است.
- $r(t)$ : ورودی مرجع به مدل است.
- $K_x$ : ماتریس بهره بازخورد (feedback gain)
- $K_r$ : بهره پیش خور (feedforward gain)
- $u_{ad}(t)$ : بخش تطبیقی کنترل است که برای جبران عدم قطعیت های مدل به کار می رود.

**هدف طراحی:**

هدف آن است که سیستم واقعی، به کمک این قانون کنترل، رفتاری مشابه با مدل مرجع از خود نشان دهد، به گونه ای که:

$$e(t) = x(t) - x_m(t)$$

$$\lim_{t \rightarrow \infty} \|x(t) - x_m(t)\| = 0$$

**انتخاب بهره ها**

برای آنکه سیستم اسمی (nominal plant) بدون در نظر گرفتن  $u_{ad}$  نیز تا حد ممکن به مدل مرجع نزدیک باشد، بهره های  $K_x$  و  $K_r$  به گونه ای انتخاب می شوند که سیستم حلقه بسته، دینامیک مشابه مدل مرجع داشته باشد.

**کنترل جانبی - زاویه فرمان:  $\delta$**

در کنترل جانبی هدف کنترل موقعیت جانبی  $y$  و سرعت جانبی  $v$  خودرو است. بنابراین بردار حالت به صورت  $x=[y \ v]$  در نظر گرفته می شود.

### کنترل طولی - شتاب $a$ یا throttle/brake

در کنترل طولی، هدف کنترل سرعت طولی  $u$  و مشتق آن (یعنی شتاب) است. بنابراین حالت به صورت  $x=[u \ \dot{u}]$  تعریف می شوند.

### نحوه به دست آوردن بهره ها (Gain Tuning Method)

فرایند تنظیم بهره ها (tuning) به شرح زیر انجام شد:

#### 1. مدل سازی سیستم دینامیکی خودرو:

از مدل تقریبی دو درجه آزادی (bicycle model) برای مدل سازی دینامیک جانبی و طولی خودرو استفاده شد.

#### 2. تعریف مدل مرجع:

برای هر زیرسیستم (طولی و جانبی)، یک مدل مرجع با پاسخ گذرای دلخواه (به طور معمول، مرتبه دوم با مشخصات  $\zeta$  و  $\omega_n$ ) تعریف شد.

#### 3. به کارگیری رابطه تطابق مدل مرجع:

با استفاده از روش تطبیق مدل (Model Matching Conditions)، بهره های  $K_x$  و  $K_r$  به گونه ای انتخاب شدند که معادلات حلقه بسته ی سیستم اسمی، با معادلات مدل مرجع مطابقت یابد. این کار با فرض عدم وجود عدم قطعیت ها انجام شده است.

## 3-5- برآورد عدم قطعیت

در کنترل تطبیقی (MRAC (Model Reference Adaptive Control)، یکی از چالش های مهم، حضور عدم قطعیت ها و عدم قطعیت های مدل در سیستم کنترل شونده است. هدف از برآورد عدم قطعیت به صورت آنالین، شناسایی این عدم قطعیت ها و جبران آن ها توسط سیگنال کنترلی تطبیقی  $u_{ad}(t)$  است، به گونه ای که کنترل کننده پایه بتواند عملکردی مشابه سیستم ایده آل (نومینال) داشته باشد.

## مدل برآورد عدم قطعیت

عدم قطعیت سیستم با استفاده از مدل زیر تقریب زده می شود:

$$w^T \phi(x) = u_{ad}(t)$$

که در آن:

- بردار وزن های برآورد شده است که به صورت آنلاین تطبیق داده می شود.
  - $\phi(x)$  بردار ویژگی ها است که نقش تقریب زننده عدم قطعیت های ناشناخته سیستم را دارد.
- انتخاب بردار ویژگی  $\phi(x)$  با استفاده از توابع پایه شعاعی (RBF)
- برای تقریب دقیق تر عدم قطعیت ها، از بردار ویژگی مبتنی بر توابع پایه شعاعی استفاده شد. توابع RBF به دلیل ویژگی های زیر برای این کاربرد مناسب هستند:
- **تقریب زننده عمومی:** با تعداد کافی تابع RBF می توان تقریب خوبی از توابع غیرخطی صاف ارائه داد.
  - **قابلیت بیان اثرات غیرخطی:** توابع RBF می توانند به خوبی تأثیرات غیرخطی دینامیک سیستم را مدل کنند، بخصوص زمانی که فقط استفاده از متغیرهای حالت مانند  $x$  کافی نباشد.
  - **محلی بودن:** هر تابع RBF پاسخ دهی خود را محدود به یک ناحیه مشخص در فضای حالت دارد که موجب تطبیق بهتر رفتار کنترل کننده در شرایط مختلف می شود.

## فرمول توابع RBF با هسته گاوسی

تابع RBF با هسته گاوسی به صورت زیر تعریف می شود:

$$\left( \frac{2\|x - c\|^2}{2d} \right) \exp = \phi_i(x)$$

که در آن:

- $C_i$  مرکز تابع پایه  $i$ ام در فضای حالت است.
- $d$  پهنای باند (bandwidth) یا پراکندگی تابع است.
- $\|x - c_i\|$  فاصله اقلیدسی بین حالت فعلی سیستم و مرکز  $i$ ام است.

پارامترهای توابع پایه شعاعی در این پروژه

- تعداد توابع پایه شعاعی 30
- محدوده مراکز توابع: از -2 تا 2 برای هر بعد از فضای حالت
- پهنای باند  $d=4$ : (bandwidth)

شرایط رانندگی سیستم کنترل شونده

مراحل انجام کار

### 1. تعریف فضای حالت و بردار ویژگی‌ها:

فضای حالت تعریف و 30 مرکز توابع RBF در محدوده  $[-2, 2]$  به صورت یکنواخت توزیع شدند. برای هر حالت  $x$  بردار ویژگی  $\phi(x)$  شامل مقادیر 30 تابع RBF محاسبه می‌شود.

### 2. محاسبه توابع RBF:

برای هر تابع  $i$ ام، مقدار  $\phi_i(x)$  با توجه به فاصله حالت فعلی تا مرکز  $c_i$  و پهنای باند  $d=4$  محاسبه می‌شود.

### 3. برآورد وزن‌های تطبیقی:

وزن‌های  $w$  به صورت آنلین توسط قانون تطبیق MRAC بروزرسانی می‌شوند تا عدم قطعیت مدل سیستم را پوشش دهند.

### 4. محاسبه سیگنال کنترلی تطبیقی:

سیگنال تطبیقی  $u_{ad}(t) = w^T \phi(x)$  تولید شده و به کنترل کننده پایه اضافه می‌شود تا سیستم را به رفتار مدل رفرنس نزدیک کند.

### 5. در نظر گرفتن شرایط رانندگی:

پارامترهای شرایط محیطی (جاده، باد، باران و وزن خودرو) به عنوان ورودی به مدل دینامیکی و شرایط عدم قطعیت وارد شده تا کنترل کننده بتواند به تغییرات محیطی پاسخ مناسب دهد.

## 3-6- نوآوری

در این پروژه برای بهبود عملکرد کنترل کننده تطبیقی MRAC، از منطق فازی برای تعیین پارامترهای مهم الگوریتم یادگیری یعنی نرخ یادگیری (Learning Rate) و گاما (Gamma) استفاده شده است. هدف این است که این پارامترها به صورت تطبیقی و بر اساس شرایط محیطی تغییر کنند تا کنترل کننده بتواند عملکرد بهتری در شرایط مختلف داشته باشد.

### مراحل انجام پروژه

#### 1. تحلیل مسئله و شناسایی متغیرها

در ابتدا متغیرهای موثر بر عملکرد کنترل کننده و پارامترهای یادگیری شناسایی شدند. این متغیرها شامل موارد زیر هستند:

- شرایط جاده (Road condition)
- سرعت باد (Wind speed)
- میزان بارش باران (Rain rate)
- وزن خودرو (Car weight)

این پارامترها به عنوان ورودی های سیستم فازی انتخاب شدند تا تاثیرات محیطی را به صورت دقیق تر در تعیین نرخ یادگیری و گاما لحاظ کنند.

#### 2. فازی سازی متغیرهای ورودی

برای هر یک از متغیرهای ورودی، مجموعه های فازی متناسب تعریف شد.

- ابتدا مقادیر دقیق (crisp) ورودی ها گرفته شد.

- سپس با طراحی توابع عضویت (Membership Functions) مناسب، این مقادیر به مقادیر فازی تبدیل شدند.
- توابع عضویت به گونه‌ای طراحی شدند که بتوانند دامنه‌های مختلف هر متغیر (مثلاً باد کم، متوسط، زیاد) را به خوبی نمایش دهند.

### 3. ساخت قوانین فازی

قوانین شامل مجموعه‌ای از قواعد IF-THEN طراحی شد که ورودی‌ها را به مقادیر خروجی دلخواه (Learning Rate) و  $\Gamma$  مرتبط می‌کند.

مثال‌هایی از قواعد تعریف شده:

- اگر سرعت باد زیاد است و میزان بارش باران زیاد است، آنگاه نرخ یادگیری بزرگ است.
  - اگر شرایط جاده نامساعد است، نرخ یادگیری باید کم شود.
- همچنین قوانین مربوط به پارامتر گاما نیز با توجه به نقش آن در الگوریتم یادگیری تعریف شدند.

### 4. مکانیزم استنتاج فازی

از موتور استنتاج فازی Mamdani استفاده شد تا بر اساس ورودی‌های فازی و قوانین تعریف شده، نتایج میانی به دست آیند.

- قوانین به صورت ترکیبی با استفاده از عملگرهای فازی مانند AND و OR اعمال شدند.
- نتایج میانی هر قانون به صورت مجموعه‌های فازی ترکیب شدند.

### 5. دفازی‌سازی (Defuzzification)

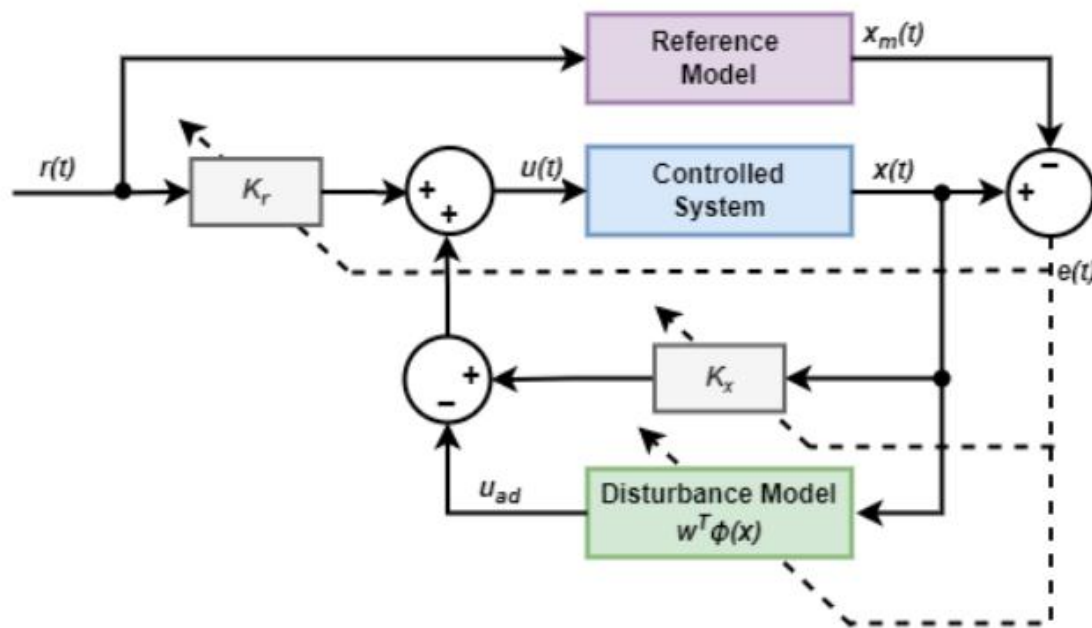
خروجی‌های فازی به مقادیر دقیق (crisp) تبدیل شدند تا بتوان از آن‌ها در کنترل کننده MRAC استفاده کرد.

روش دفازی‌سازی انتخاب شده، روش مرکز ثقل (Centroid) یا میانگین وزن دار (Weighted Average) بود.

مقادیر به دست آمده برای نرخ یادگیری و گاما در طراحی کنترل کننده به صورت تطبیقی اعمال شدند تا عملکرد سیستم بهبود یابد.

### 7-3- بلوک MRAC در Simulink

در این بخش از پروژه فرض شد که دینامیک سیستم شناخته شده می باشد و در نتیجه کنترل به صورت مستقیم انجام می پذیرد. این کنترل کننده شامل ساختار زیر می باشد:



که خروجی کنترلی را به شکل زیر ارائه می دهد.

$$u(t) = k_x x(t) + k_r r(t) - u_{ad}$$

$$u_{ad} = w^T \phi(x)$$

این بلوک به صورتی طراحی شده است که می تواند قوانین تطبیق را به توجه قانون لیاپانف به شکل زیر پیاده کند:



$$\begin{aligned}\dot{k}_x &= \Gamma_x x(t) e^T(t) P B \\ \dot{k}_r &= \Gamma_r r(t) e^T(t) P B \\ \dot{w} &= \Gamma_w \phi(x) e^T(t) P B \\ A_m^T P + P A_m + Q &= 0\end{aligned}$$

### 3-8- نتایج

با دادن سیگنال سینوسی به صورت مرجع به سیستم خروجی سیستم و تخمینی که از اغتشاش زده شد است به شکل زیر حاصل می شود که نشان دهنده عملکرد درست کنترل کننده می باشد.

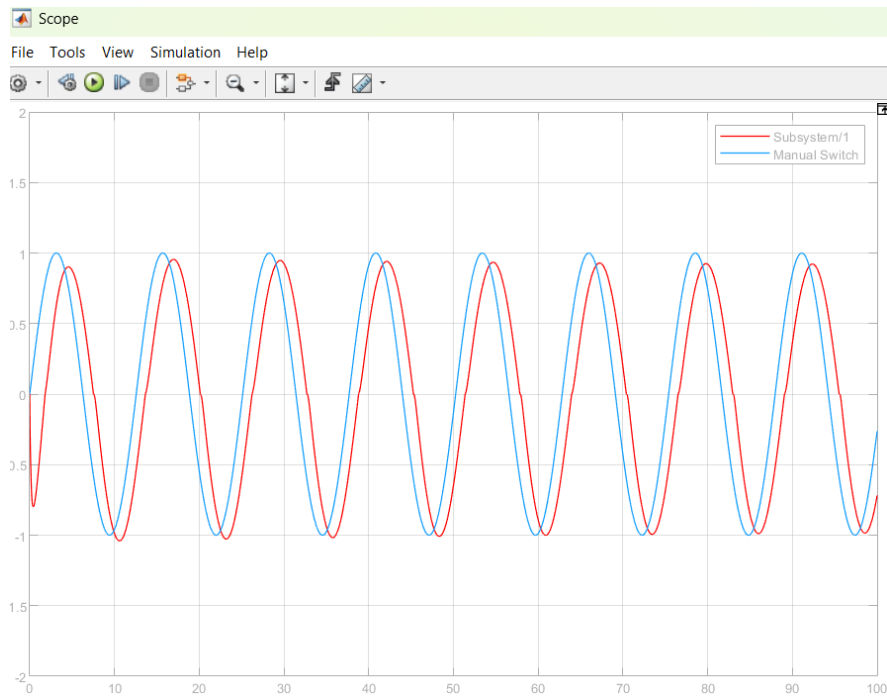


Figure 1) خروجی سیستم و سیگنال مرجع

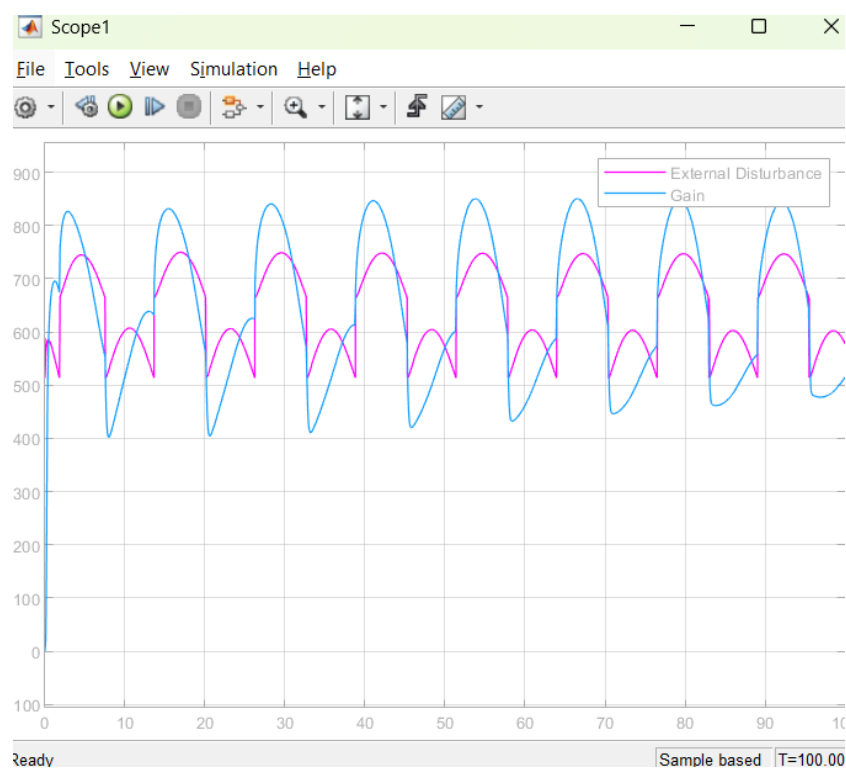


Figure 2) اغتشاش اصلی و اغتشاش تخمینی

## فصل چهارم شبیه سازی در Carla

## شبیه سازی در Carla

در این بخش نحوه شبیه سازی در Carla تشریح می گردد.

### 1-4- طراحی مدل مرجع

تا پیش از این در تکالیف و شبیه سازی های Simulink مدل مرجع به عنوان تابع تبدیلی پایدار مطرح میشد. در محیط های واقعی شرایط کمی پیچیده تر است و نمی توان مدل مرجعی با تابع تبدیل ساده و دلخواه داشت که سیستم رفتاری مشابه به آن داشته باشد. در این بخش از پروژه خودرویی ایده آل که در محیطی ایده آل ( سطح هموار و شرایط جوی پایدار) در نظر گرفته می شود که هدف آن دنبال کردن مسیری فرضیست. خودروی ایده آل خودرویی فرضیست که به صورت autopilot در شبیه ساز Carla به حرکت در می آید و مسیری را به انتخاب خود دنبال می کند. ورودی مرجع در این بخش از پروژه مسیری می شود که این خودرو ایده آل فرضی در شبیه ساز طی میکند. اطلاعات مربوط به مختصات خودرو سرعت و ورودی هایی که در هر لحظه داشته است تا مسیر را طی کند با step های زمانی 0.05 ثانیه در فایل CSV ذخیره می شود. بخشی از این داده های ذخیره شده در پایین نمایش داده شده اند:

time	speed_mph	throttle	steer	brake	x	y	z	pitch	yaw	roll
0	0.057318	0	0	0	-64.6448	24.47101	-0.01491	0	0.159198	0
0.05	0.04431	0	0	0	-64.6448	24.47101	-0.01245	0	0.159198	0
0.1	0.034007	0	0	0	-64.6448	24.47101	-0.01055	0	0.159198	0
0.15	0.025937	0.85	-0.15	0	-64.6448	24.47101	-0.0091	0	0.159198	0
0.2	0.019679	0.85	-0.15	0	-64.6448	24.47101	-0.008	0	0.159198	0
0.25	0.014862	0.85	-0.15	0	-64.6448	24.47101	-0.00716	0	0.159198	0
0.3	0.011178	0.85	-0.15	0	-64.6448	24.47101	-0.00654	0	0.159198	0
0.35	0.008377	0.85	-0.10903	0	-64.6448	24.47101	-0.00606	0	0.159198	0
0.4	0.006258	0.85	-0.10903	0	-64.6448	24.47101	-0.00571	0	0.159198	0
0.45	0.004661	0.85	-0.10903	0	-64.6448	24.47101	-0.00545	0	0.159198	0
0.5	0.003463	0.85	-0.10903	0	-64.6448	24.47101	-0.00525	0	0.159198	0
0.55	0.000988	0.85	-0.10903	0	-64.6448	24.47101	-0.00522	0	0.159198	0
0.6	0.001345	0.85	-0.10903	0	-64.6448	24.47101	-0.00516	0	0.159198	0
0.65	0.101293	0.85	-0.10903	0	-64.6438	24.47093	-0.00511	0.006058	0.15573	0.002318
0.7	0.51304	0.85	-0.10903	0	-64.6255	24.46981	-0.00534	0.085698	0.090863	0.024691
0.75	0.74541	0.85	-0.10903	0	-64.5927	24.46785	-0.00559	0.171854	0.004367	0.04314
0.8	1.002934	0.85	-0.10903	0	-64.546	24.46494	-0.00584	0.252184	-0.1106	0.06187

Figure 3 (جدول اطلاعات ثبت شده از مدل مرجع ( حاصل کد datacollector.py

حال با استفاده از داده های موجود در جدول تخمینی ساده از این سیستم زده می شود. این تخمین با استفاده از الگوریتم RLS می باشد. برای این کار معادله ای از سیستم مرجع به صورتی فرض می شود که

تابع حالت های پیشین و همچنین ورودی های فرض شده می باشد. با کمک اطلاعات جدول و تابع فرض شده ماتریس های  $A_m$  و  $B_m$  استخراج میگردند.

معادلات:

$$X_T = A * X_{T-1} + B * THROTTLET + C * STEERT + D * BRAKET$$

$$Y_T = A * Y_{T-1} + B * THROTTLET + C * STEERT + D * BRAKET$$

$$Y_{AWT} = A * Y_{AWT-1} + B * THROTTLET + C * STEERT + D * BRAKET$$

$$SPEEDT = A * SPEEDT-1 + B * THROTTLET + C * STEERT + D * BRAKET$$

$$A_m = [A_x \ 0 \ 0 \ 0; 0 \ A_y \ 0 \ 0; 0 \ 0 \ A_{yaw} \ 0; 0 \ 0 \ 0 \ A_{speed}]$$

$$B_m = [B_x \ B_y \ B_{yaw} \ B_{speed}; C_x \ C_y \ C_{yaw} \ C_{speed}; D_x \ D_y \ D_{yaw} \ D_{speed}]$$

با استفاده از کد `modelestimator.py` فضای حالت به شکل زیر تخمین زده می شود:

```
AM =
[[1.00176536 0.          0.          0.          ]
 [0.          0.99738321 0.          0.          ]
 [0.          0.          0.9287813  0.          ]
 [0.          0.          0.          0.97971263]]

BM =
[[ 0.07075442  0.04591283  3.61660205  0.23144604]
 [ 0.0584045   0.19913708 -7.21824105 -0.01847172]
 [ 0.08392217  0.06304376  3.46838312 -0.08955975]]
```

## 2-4- مدل سیستم

در این بخش خودروی مورد نظر تنظیم می شود که هدف پروژه طراحی کنترل کننده برای این خودرو می باشد. در تنظیمات نرم افزار می توان ویژگی های فنی خودرو از جمله .... تغییر یابند. شرایط جوی نیز به بارانی تغییر کند. ورودی های کنترلی برای خودرو مورد کنترل `brake`, `throttle` و `steer` می باشند. برای دوباره این موضوع لحاظ می شود که پارامتر های سیستم قابل دسترسی نیستند. این بار بر خلاف شبیه سازی سیمولینک از کنترل غیر مستقیم استفاده می شود. بدین صورت که:

$$\dot{\hat{A}} = \Gamma_a x(t) e^T(t) P$$

$$\dot{\hat{B}} = \Gamma_b u(t) e^T(t) P$$

$$\dot{w} = \Gamma_w \phi(x) e^T(t) P B$$

$$\dot{k}_x = \Gamma_x x(t) e^T(t) P B + \sigma k_x$$

$$\dot{k}_r = \Gamma_r r(t) e^T(t) P B + \sigma k_x$$

$$\dot{w}_x = \Gamma_w \phi(t) e^T(t) P B + \sigma w$$

### 3-4- شرح طراحی کنترل کننده

برای کنترل کننده تطبیقی MRAC نیاز به مدل مرجع می باشد. این مدل مرجع ورودی مرجع را دریافت کرده و خروجی دلخواه را تحویل می دهد. سیستم هدف نیز باید با دریافت مدل ورودی مرجع به خروجی یکسان دست پیدا کند که این خود نیازمند کنترلی هوشمند می باشد. ورودی مرجع سرعت و چرخش فرمان ایده آلیست که خودرو باید در هنگام حرکت روی مسیر فرضی داشته باشد. مراحل عملکرد کنترل کننده مدل مرجع در بخش ۴-۱ استخراج شد. مدل تخمینی سیستم نیز در بخش قبل استخراج شد حال با کمک دینامیک به دست آورده شده از این دو سیستم کنترل کننده MRAC مشابه بخش Simulink به دست می آید و روی سیستم جدید اعمال می شود.

### 4-4- نوآوری (کنترل تطبیقی به همراه فازی)

فرض می شود سنسورهای موجود در خودرو قابلیت اندازه گیری وزن خودرو و یا نرخ بارش باران را دارد. حال برای این شرایط مشابه آنچه در Simulink وجود داشت نرخ یادگیری با قوانین فازی تعیین می شود.

## فصل پنجم نوآوری

## نوآوری

در مقاله مرجع کنترلر MRAC طراحی شده انعطاف کافی را نداشت. برای شرایط جوی بسیار بد و بد با سرعت یکسان عمل میکرد در حالی که باید احتیاط در نظر گرفته می شد. در هوای بسیار بارانی سرعت ایده آل به اندازه ای که ماشین ایده آل در حرکت است نباید باشد. به همین دلیل خودرو هدف باید انعطاف داشته باشد تا سرعت ایده آل خود را دنبال کند. پارامترهای تغییر یافته در تنظیمات فرض می شود که با سنسورهای خودرو قابلیت حس شدن دارند. بر روی پارامترهای تغییر یافته سازی انجام می شود که در ادامه به کنترل کننده اضافه می شود. همچنین در مقاله مرجع فرض می شد که مدل مرجع در اختیار گذاشته شده و مدلی بسیار ایده آل بود ولی در این پروژه مدل مرجع خود حاصل تخمینی از سیستمی آزمایش شده بود ( در بخش شبیه سازی در Carla)



## فصل ششم

### جمع‌بندی و نتیجه‌گیری و پیشنهادات

## جمع‌بندی و نتیجه‌گیری

در این پروژه هدف دنبال کردن مسیری مشخص مانند مدل مرجع برای خودرویی خودران بود. شبیه سازی در دو نرم افزار Simulink و Carla انجام گرفت. در Simulink شبیه سازی کاملاً به صورت ایده آل انجام شد ولی در CARLA به محیط واقعی نزدیک تر و عملکرد دقیق تری داشت. نوآوری انجام گرفته در این پروژه استفاده از قوانین فازی برای دستیابی به نرخ یادگیری دقیق تر بود.

## پیشنهادهات

برای تخمین پارامترهای به شیوه کلاسیک و قوانین لیاپانف می توان از شبکه عصبی برای تخمین پارامترهای کنترل کننده استفاده کرد که از سرعت و دقت بهتری برخوردار می باشد.

## منابع و مراجع

- [1] Abolfazl Simorgha, Abolhassan Razminiab,, Arash Marashianb, Adaptivevelocity control of an autonomousvehicleusing input-error model reference approach, Journal of the Franklin Institute 361 (2024) 106700

