

بسمه تعالی



دانشکده مهندسی کامپیوتر

دانشگاه صنعتی شریف

آزمایشگاه سخت افزار

پروپوزال پروژه شماره ۳

سامانه کمک نگهدارنده خط

استاد: جناب آقای دکتر اجلالی

گروه ۵: ترلان بهادری - همیلا میلی

فهرست

1	فهرست
2	مقدمه
3	روش انجام پروژه
3	بررسی کلی پروژه
4	سنسورها
4	سخت افزار
5	نرم افزار
6	معماری سیستم
6	سخت افزار
7	نرم افزار
8	جدول هزینه ها
9	سخت افزار
9	زمان بندی
10	۴ آبان
10	۱۸ آبان
10	۲ آذر
10	۱۶ آذر
10	۳۰ آذر
10	۷ دی
11	منابع

مقدمه

هدف از انجام این پروژه طراحی یک سامانه کمک نگهدارنده خط است. با پیشرفت تکنولوژی خودروها و خودران شدن آنها، لازم است سیستم‌های نگهدارنده خط دقیق‌تر از گذشته عمل کنند و به همین جهت این پروژه با هدف پیاده‌سازی هر چه بهتر این سیستم‌ها طراحی خواهد شد. برای پیاده‌سازی می‌توان از هر دو روش سخت‌افزاری و نرم‌افزاری استفاده نمود. به دلیل در دسترس بودن حالت نرم‌افزاری، این روش را برای پیاده‌سازی پروژه در نظر خواهیم گرفت.

به طور کلی برای انجام پروژه لازم است با در دست داشتن تعدادی ورودی از روی دوربین‌ها یا سنسورهای متصل شده در نقاط مختلف خودرو، درباره نحوه ادامه حرکت خودرو تصمیم‌گیری شود. بنابراین به طور خلاصه مهم‌ترین ورودی‌ها و خروجی‌های برنامه در جدول زیر آورده شده‌اند.

نوع داده	دریافت‌کننده	توضیحات
ورودی	دوربین‌های ثبت حرکت	نتیجه نهایی مورد نیاز از این دوربین‌ها، مختصات به دست آمده از ادامه خطوط مسیر و زاویه ماشین در مسیر است.
ورودی	سنسورهای فاصله (مادون قرمز)	این داده از نزدیک شدن بیش از حد خودروها به دیواره مسیر و سایر ماشین‌ها جلوگیری می‌کند. همچنین می‌تواند در حفظ فاصله ماشین از جلو و عقب کمک‌کننده باشد.
خروجی	زاویه فرمان	در صورتی که چرخش در مسیر توسط ورودی‌ها شناسایی شود، زاویه فرمان باید به گونه‌ای تغییر کند که خودرو از میان دو خط عبور نکند.

جدول ۱ - معرفی ورودی‌ها و خروجی‌های مورد نیاز و دریافت‌کننده‌های آنها

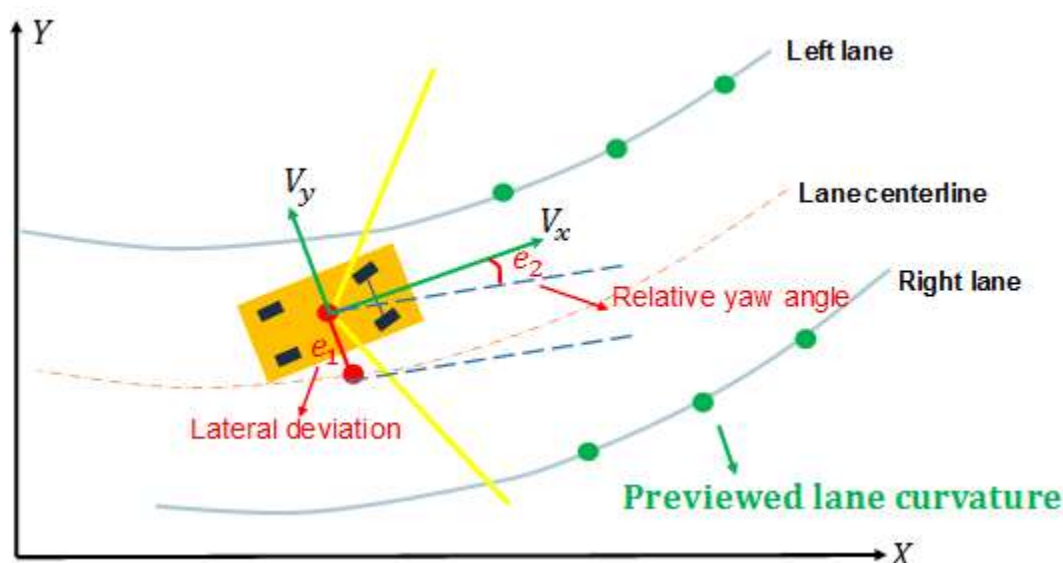
پس از مشخص شدن زاویه خروجی و اعمال آن، ماشین حرکت کرده و وارد حالت بعدی می‌شود و ورودی‌های لازم را از حالت جدید به دست آمده مجددا دریافت می‌کند.

روش انجام پروژه

بررسی کلی پروژه

نرم افزاری

در پیاده‌سازی پروژه به صورت نرم‌افزاری، از برنامه MatLab به همراه Simulink برای شبیه‌سازی حرکت ماشین استفاده خواهیم کرد. همان‌طور که در مقدمه به آن اشاره شد، در هر لحظه با دریافت اطلاعاتی از ورودی‌ها درباره زاویه فرمان برای لحظات بعد تصمیم‌گیری خواهد شد. شکل زیر نمای جامع‌تری از اطلاعات لازم برای تعیین زاویه فرمان را نمایش می‌دهد:



شکل ۱- شمایی از اطلاعات لازم برای تعیین زاویه فرمان

در هر مرحله، state ماشین از روی سرعت در راستای محور y و نرخ زاویه انحراف تعیین می‌شود. با ورودی گرفتن زاویه فرمان، مقادیر مورد نیاز برای state بعدی تعیین می‌شوند.

سخت افزاری

در صورت پیاده‌سازی سخت‌افزاری، لازم است که ورودی‌ها هر لحظه از سنسورها و دوربین دریافت شوند و با استفاده از یک میکروکنترلر Raspberry Pi 3b می‌توانیم آنها را پردازش کنیم و مقدار خروجی را به یک میکروکنترلر Arduino Uno که با استفاده از سیم به Raspberry Pi متصل است (اتصال سریال) منتقل کنیم (به علت اینکه مقادیر آنالوگ خوانده شده از سنسورها در ورودی را بتوانیم به آنالوگ تبدیل کنیم). این میکروکنترلر Arduino Uno باید به سیستم فرمان و موتور خودرو متصل باشد و مقادیر خروجی مثل زاویه فرمان را به آنها منتقل کند.

سنسورها

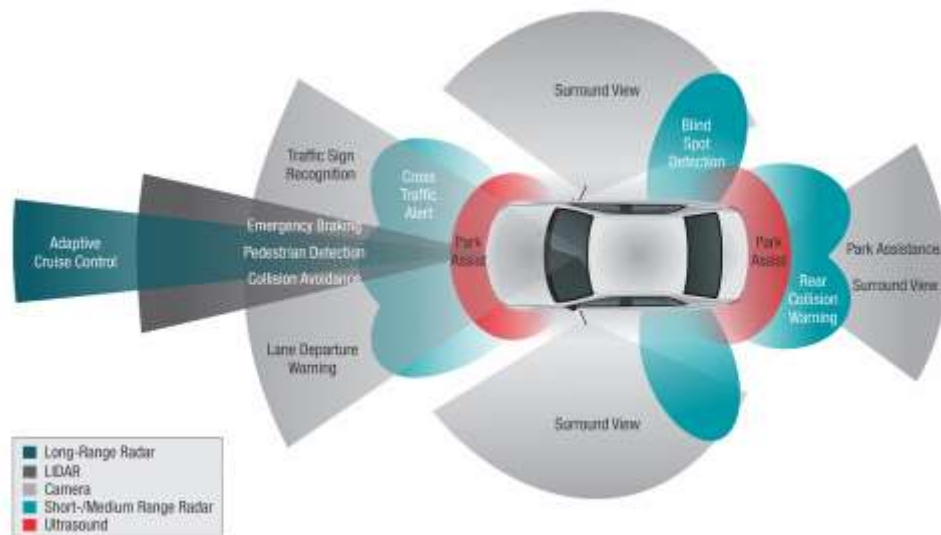
سخت‌افزاری

با توجه به اینکه این پروژه یک پروژه تحقیقاتی است، طرح ارائه شده صرفاً برای سخت‌افزاری است که می‌تواند جهت آزمایش الگوریتم نرم‌افزاری مورد استفاده قرار بگیرد و به همین علت از سخت‌افزار با کیفیت قابل قبول برای استفاده در خودرو واقعی معرفی نخواهد شد.

در خودروهایی که مجهز به سیستم کمک نگهدارنده خط هستند، 6 سنسور مادون قرمز برای تشخیص ترافیک در اطراف خودرو به کار می‌رود که مطابق تصویر زیر، 4 سنسور در قسمت عقب خودرو و 2 سنسور در قسمت جلو تعبیه می‌شود و همچنین یک دوربین لازم است که ترافیک جلوی خودرو را تشخیص دهد.



شکل ب- موقعیت و عملکرد سنسورهای مادون قرمز در خودروهای مجهز به سیستم کمک نگهدارنده خط

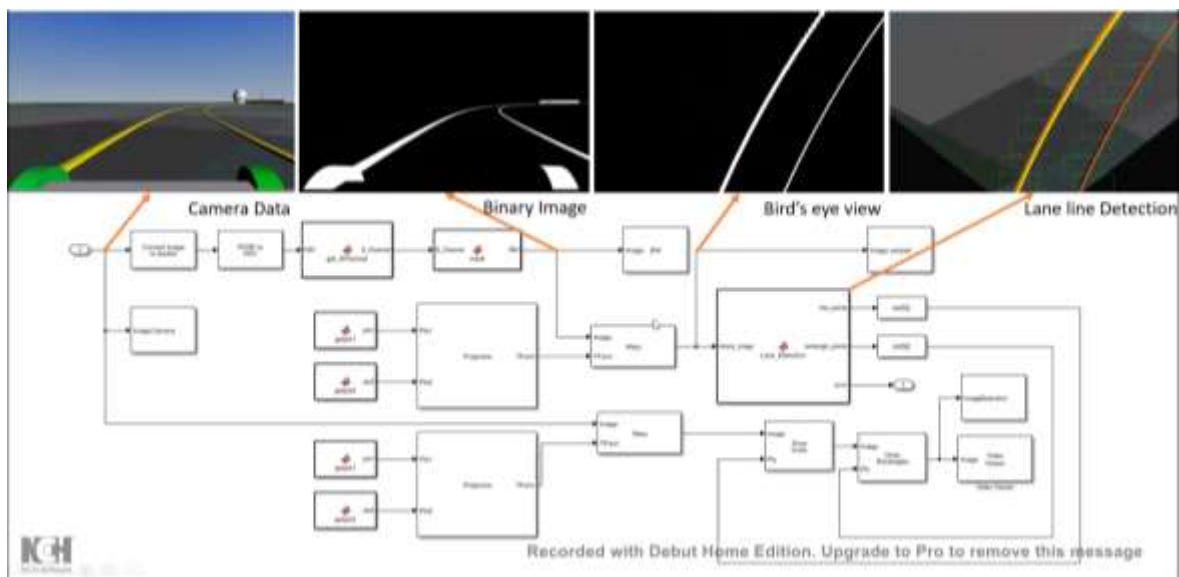


شکل ج - انواع سیستم‌های اطلاعاتی استفاده شده در خودروهای مجهز به سیستم کمک نگهدارنده خط. همان طور که از شکل مشخص است، در برخی مناطق تنها سنسورهای کوتاه‌برد مورد استفاده هستند. این در حالی است که سنسورهای دوربرد و حتی دوربین در برخی مکان‌ها مورد استفاده قرار می‌گیرند.

در واقع ما فقط از short-medium range radar و Camera در این پروژه استفاده خواهیم کرد و نیازی به سایر این سنسورها برای سیستم نگهدارنده خط نداریم.

نرم‌افزاری

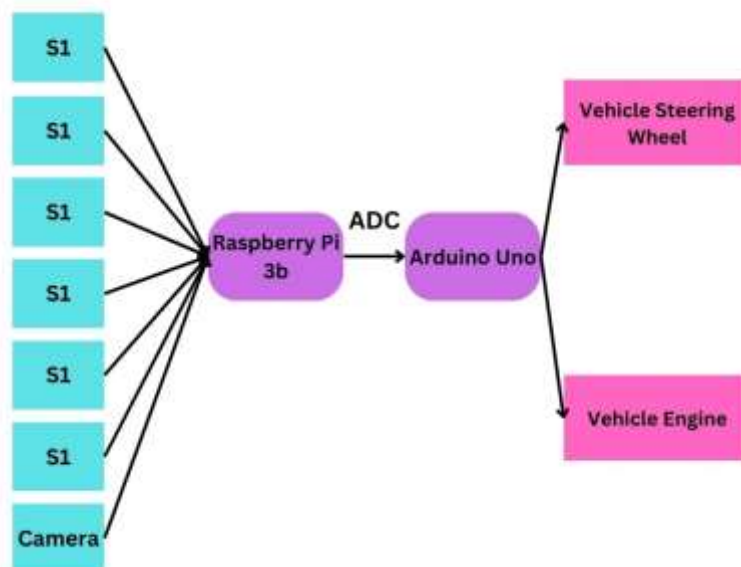
در حالت نرم‌افزاری، برای دریافت اطلاعات مورد نیاز سیستم روش‌های گوناگونی وجود دارد. می‌توان با استفاده از ماژول دوربین و با اعمال فیلترهای مشخص بر روی عکس و تغییر زاویه دوربین، خطوط و فاصله ماشین از آنها را شناسایی کرد. همچنین می‌توان از ماژول‌های آماده موجود در Simulink نیز استفاده نمود که در این صورت کار برنامه‌نویسی بسیار آسان‌تر می‌شود و ماژول‌های کمتری نیز مورد نیاز خواهند بود. در رابطه با سنسورهای فاصله پیچیدگی زیادی برای تصمیم‌گیری وجود نخواهد داشت چرا که داده‌های خروجی از این اجزا به طور مستقیم مورد استفاده قرار خواهند گرفت. در صورت ساخت واحد ادراک، طراحی زیر برای بلوک ذکر شده مد نظر قرار داده خواهد شد.



شکل د- ساختمان واحد پردازش تصویر و تشخیص خط در روش مستقل از مازول متلب

معماری سیستم

سخت‌افزاری



شکل ه- معماری سیستم در حالت سخت‌افزاری

نرم‌افزاری

شمای کلی برنامه در شبیه‌ساز به صورت زیر خواهد بود:

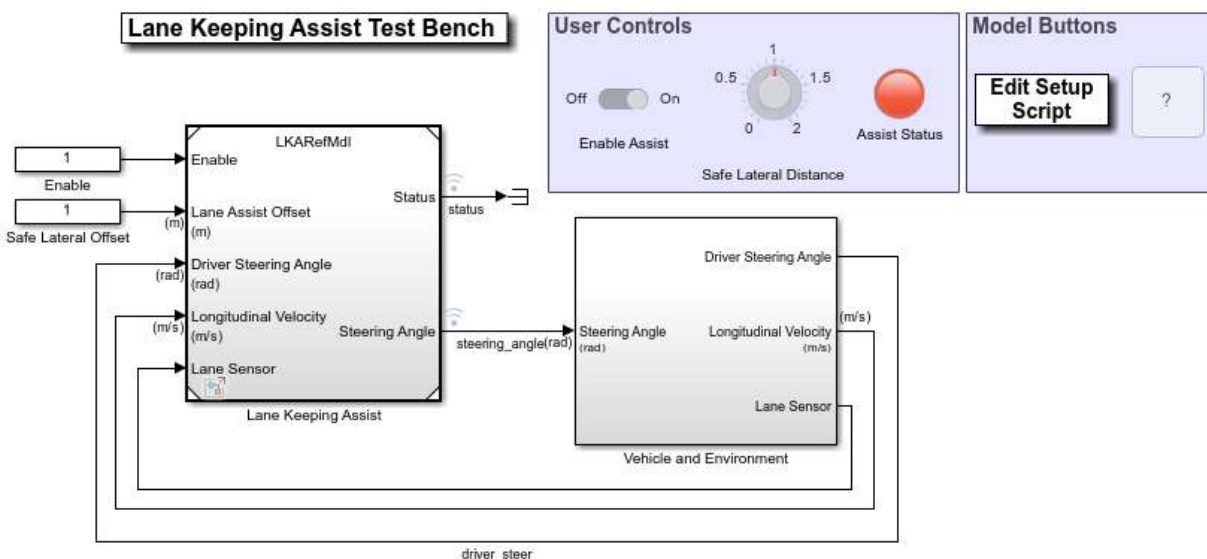
در حالت نرم‌افزاری همان‌طور که در مقدمه نیز به آن اشاره شد، از MatLab و Simulink برای شبیه‌سازی استفاده خواهیم کرد. ماژول‌های آماده‌ای در این زمینه در برنامه متلب وجود دارند که در اینجا به اختصار درباره آنها توضیح خواهیم داد.

Lane Keeping Assist with Lane Detection

این سیستم به طور خودکار تشخیص می‌دهد که چه زمانی خودرو در آستانه خروج از خط قرار می‌گیرد و در همین زمان زاویه فرمان را برای جلوگیری از این اتفاق تغییر می‌دهد. این ماژول از دو بخش اساسی تشکیل شده است:

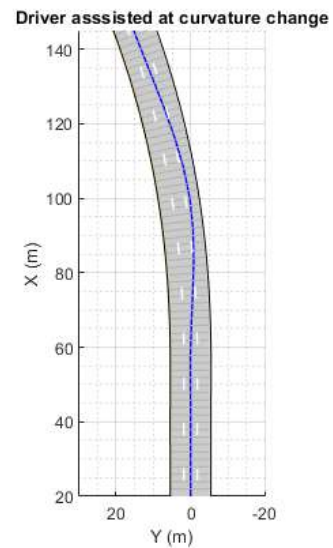
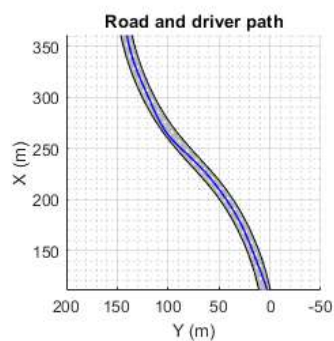
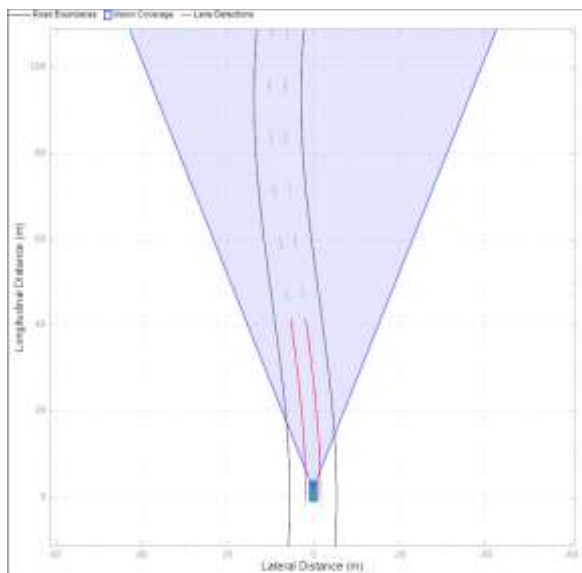
1. سامانه کمک نگه‌دارنده خط که زاویه فرمان خودرو را تنظیم می‌کند.
2. زیرمجموعه خودرو و محیط که برای شبیه‌سازی موقعیت خودرو و مشخص نمودن فاصله خودرو از خطوط طراحی می‌شود.

از مزایای استفاده از این ماژول آن است که عملکرد خودرو برای یک مسیر از پیش طراحی شده از دو نمای عمودی و Bird's-Eye قابل مشاهده است. همچنین قابلیت‌های متلب این امکان را در اختیار ما قرار می‌دهد تا با آزمایش الگوریتم‌ها و مقادیر ثابت مختلف، نمودارهای مختلفی از یک مسیر را رسم کرده و راهکارها را با هم مقایسه کنیم.

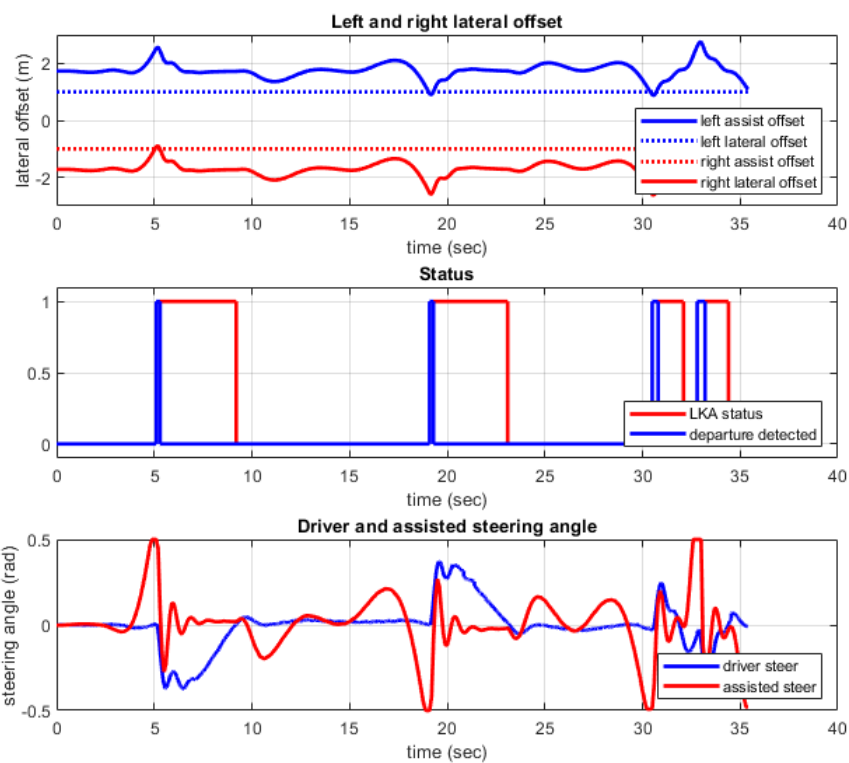


Copyright 2017-2020 The MathWorks, Inc.

شکل و - ساختار سیستم LKA در متلب



شکل ز- نمای عمودی (دو شکل واقع در سمت راست) و نمای Bird's-Eye از مسیری طی شده توسط خودرو در شبیه‌سازی



شکل ح- مقایسه راهکارهای مختلف در زمینه‌های متفاوت (این مقادیر توسط سازندگان ماژول به دست آمده است)

جدول هزینه‌ها

سخت‌افزاری

در جدول زیر تخمینی از هزینه مورد نیاز برای پیاده سازی پروژه به صورت محصول سخت افزاری آمده است.

نام قطعه	فی (هزار تومان)	تعداد مورد نیاز	قیمت کل (هزار تومان)	لینک فروشنده
سنسور مادون قرمز	260	6	1560	لینک
دوربین	1350	1	1350	لینک
Raspberry Pi 3b	790	1	3872	لینک
ATMega 328 Microcontroller (Arduino Uno)	289	1	289	لینک
LCD	51	1	51	لینک
SD Card	190	1	190	لینک
			جمع: 7312	

زمان بندی

۴ آبان

ارائه پروپوزال و تصویب آن

۱۸ آبان

ارائه گزارش میانی اول – راه اندازی سیستم خودرو شبیه سازی شده و ارتباط میان اجزا

۲ آذر

ارائه گزارش میانی دوم – نصب سنسورها بر روی خودروی شبیه سازی و تعیین ورودی و خروجی ها

۱۶ آذر

ارائه گزارش میانی سوم – یافتن بهترین مکان برای نصب سنسور از طریق آزمون و خطا

۳۰ آذر

تحویل اولیه پروژه

۷ دی

تحویل نهایی پروژه به همراه مستندات نهایی و فیلم

از توجه شما متشکریم

1. <https://www.aparat.com/v/KdzC6>
2. <https://digiato.com/article/2016/07/18/%D8%AA%DA%A9%D9%86%D9%88%D9%84%D9%88%DA%98%DB%8C-lane-keeping-assist%D8%9B-%D8%B1%D9%88%D8%B4-%D9%85%D8%B1%D8%B3%D8%AF%D8%B3-%D8%A8%D9%86%D8%B2-%D8%A8%D8%B1%D8%A7%DB%8C-%D9%BE%DB%8C%D8%B4%DA%AF%DB%8C>
3. https://mej.aut.ac.ir/article_667.html
4. https://www.researchgate.net/publication/316681953_Hardware_Simulation_of_Active_Lane_Keeping_Assist_Based_on_Fuzzy_Logic
5. <https://uk.mathworks.com/help/mpc/ug/lane-keeping-assist-system-using-model-predictive-control.html>
6. <https://uk.mathworks.com/help/mpc/ug/lane-keeping-assist-with-lane-detection.html>
7. <https://www.youtube.com/watch?v=hSiTNyfp3T0>