Slovenská technická univerzita Fakulta informatiky a informačných technológií Ilkovičova 2, 842 16 Bratislava

Tímový projekt

Korekcia dynamických vlastností virtuálnych modelov komponentov vozidiel

Dokumentácia k inžinierskemu dielu

Vedúci práce:	Ing. František Horvát, PhD.		
Členovia tímu:	Bc. Katarína Juhásová		
	Bc. Lukáš Anda		
	Bc. Dávid Baári		
	Bc. Šimon Ferančík		
	Bc. Dávid Kecskés		
	Bc. Roman Slonskyi		
	Bc. Mykola Yanko		
Akademický rok:	2020/2021		

Obsah

1 Úvod	2
2 Globálne ciele na zimný semester	3
3 Architektúra systému	4
3.1 Moduly	5
3.1.1 Desktopová aplikácia	5
3.1.2 Analýza najazdených okruhov	5
3.1.3 Optimalizácia steering commandu	5
4 Moduly systému	6
4.1 Desktopová aplikácia	6
4.2 Analýza najazdených okruhov	6
4.3 Optimalizácia steering commandu	7
5 Výsledky testovania	8
Tabuľka testovania	8
6 Nastavenie prostredia na vývoj	9
7 Inštalačná príručka	10
8 Používateľská príručka	11
Analýza dráhy	11
Optimalizácia komponentov	13

1 Úvod

Tento dokument obsahuje dokumentáciu k inžinierskemu dielu projektu s témou *Korekcia dynamických vlastností virtuálnych modelov komponentov vozidiel [CarComponents]* v rámci predmetu Tímový projekt.

Cieľom projektu je návrh a implementácia desktopovej aplikácie, ktorá má slúžiť ako simulátor správania vybraných komponentov používaných v automobilovom priemysle. Tento projekt je riešený v spolupráci so SjF STU. Simulácia správania komponentov má byť prevedená kombináciou matematických opisov komponentov a strojového učenia.

2 Globálne ciele na zimný semester

Cieľ projektu je definovaný ako návrh a implementácia desktopovej aplikácie, ktorá má slúžiť ako simulátor správania vybraných komponentov používaných v automobilovom priemysle. Zo SjF STU budú poskytnuté namerané časové dáta opisujúce správanie viacerých komponentov v reálnom živote spolu s matematickými opismi pre dynamické správanie týchto komponentov. Celkovo má byť simulované správanie troch komponentov, v zimnom semestri je však plánované zahrnutie iba jedného z nich - volantu. V rámci projektu bude teda optimalizovaný tzv. *steering command*.

Funkcionality aplikácie, ktoré majú byť aplikované v zimnom semestri je možné zhrnúť do dvoch bodov:

- 1. analýza najazdených okruhov,
- 2. optimalizácia steering command-u.

V rámci analýzy najazdených okruhov má byť možné nahrať súbor s najazdenými okruhmi a k nim príslušný referenčný okruh. Tieto okruhy majú byť vykreslené v diagrame a zároveň majú byť poskytnuté ďalšie štatistické informácie pre jednotlivé okruhy. Štatistické informácie by mali zahŕňať počet bodov zaznamenaných na okruhoch, dĺžky najazdených okruhov a metriku vyjadrujúcu podobnosti/odchýlky okruhov v porovnaní s referenčným okruhom. Tieto informácie má byť možné uložiť a vytlačiť. Zároveň je požadované rozdelenie pôvodného súboru s viacerými okruhmi na jednotlivé okruhy a export týchto okruhov ako samostatných súborov.

Na optimalizáciu *steering command-u* bolo pôvodne plánované využitie neurónových sietí, avšak po analýze dostupných dát bola zvolená alternatívna metóda strojového učenia - učenie posilňovaním (angl. *Reinforcement Learning*). Pomocou tejto metódy by malo byť možné upraviť výstupy z matematického opisu komponentu tak, aby sa výsledky čo najviac priblížili k reálnym hodnotám. V rámci používateľského pre rozhrania túto časť aplikácie má byť vizualizovaný priamo komponent, pre ktorý bude prebiehať simulácia a diagramy, na ktorých budú znázornené reálne namerané hodnoty, výstupy simulácie a korekcie (rozdiely medzi nameranými a simulovanými dátami).

V prípade, že všetky vyššie opísané požiadavky budú splnené v predstihu, zvážime zahájenie zapracovania ďalších komponentov už v zimnom semestri.

3 Architektúra systému

Projekt je rozdelený na 2 menšie časti. Pôvodne bol projekt rozdelený na 2 samostatné podprojekty ale rozhodli sme sa ich zlúčiť a vytvoriť jeden väčší, pretože vznikla potreba robiť trénovanie modelu na mieste používania.

Prvá časť sa zaoberá nastavením reinforced learningu a jeho trénovaním. Používa jazyk Python, pričom jeho vstupom sú záznamy z najazdených trás a referenčné trasy. Tie sa následne spracujú a pomocou knižnice Keras a OpenAI Gym sa vyhodnotia a trénujú. Ako inšpiráciu sme použili toto nastavenie prostredia: <u>Car racing</u>. Na vyhodnotenie nasledujúceho kroku sa používa model bicyklu z balíku funkcii z Matlab-u.

Druhá časť sa zaoberá samotnou aplikáciou. Je rozdelená na štatistickú analýzu, ktorá je spracovaná v jazyku Python avšak nepoužíva úplne rovnaký preprocessing ako prvá časť, pretože nevyžaduje všetky dáta. Spracované dáta sa zobrazia v aplikácii. Predná časť aplikácie, teda frontend, je vytvorená pomocou frameworku Electron. Je navhnutá ako tenký klient, to znamená že len zobrazuje dáta a reaguje na interakcie používateľa. V prípade potreby je veľmi jednoduché vytvoriť webovú stránku z danej aplikácie.



Obr. 1: Architektúra systému

3.1 Moduly

Z pohľadu architektúry je projekt rozdelený na viacero častí. Moduly a ich funkcionality sú podrobne opísané v nasledujúcej kapitole 4.

3.1.1 Desktopová aplikácia

Jednou z požiadaviek bola desktopová aplikácia. Rozhodli sme sa ju vytvoriť pomocou jazyku JavaScript vo frameworku Electron. Backend je tvorený v jazyku Python, aby bola zjednodušená spolupráca s učením posilňovaním.

3.1.2 Analýza najazdených okruhov

Rovnako dôležitou požiadavkou bola analýza okruhov z dát, ktoré sú vložené do aplikácie. Táto analýza je vykonávaná v jazyku Python a ponúka ohodnotenie trasy z viacero pohľadov.

3.1.3 Optimalizácia steering commandu

Kľúčovou požiadavkou pre tento projekt bola optimalizácia steering commandu pre konkrétnu trasu. Tá funguje vďaka natrénovaniu modelu pomocou učenia posilňovaním. Následne sa do modelu vložia trasy a vypočíta sa optimálnejší steering command.

4 Moduly systému

4.1 Desktopová aplikácia

Analýza

Na základe požiadaviek zákazníka je potrebné vytvoriť aplikáciu, ktorá spĺňa dve kľúčové funkcionality. Prvou z nich je výber komponentov, ktoré je potrebné optimalizovať a analyzovanie samotnej jazdnej dráhy po vložení referenčných dát a najazdených okruhov. Je potrebná aby aplikácia bola spustiteľná vo forme .exe súboru.

Návrh

Po spustení aplikácie sú používateľovi poskytnuté dve možnosti a to analýza dráhy a optimalizácia komponentov. Po rozkliknutí analýza dráhy je potrebné vložiť súbor s referenčnou dráhou a log súbor. Po nahratí súborov aplikácia v tejto časti vykreslí okruh a vypíše zoznam kôl a ich ohodnotenia v bodoch.

Implementácia

Aplikácia komunikuje s backendovou časťou, ktorá je napísaná v jazyku Python pomocou externých súborov typu JSON, CSV a LOG, ktoré sú pred spracované na backende v Pythone a zobrazené na frontende v Electrone.

Testovanie

Testovanie používateľského rozhrania prebieha bude prebiehať za účasti zákazníka, ktorý program pripomienkuje a snaží sa vyzdvihnúť hlavné chyby čo sa týka user experience ale aj samotnej funkcionality.

4.2 Analýza najazdených okruhov

Analýza

Je potrebné aby v aplikácii bolo možné nahrať súbor s dátami o trase, a zistiť z neho koľko bolo prejdených okruhov, vypísať počet bodov pre každý okruh a zistiť dĺžky jednotlivých okruhov. Ďalej je potrebné vypočítať odchýlku najazdenej trasy od referenčnej a taktiež vypočítať priemernú odchýlku od referenčnej trasy. Nakoniec je potrebné tieto dáta exportovať a zobraziť, resp. vytlačiť.

Návrh

Keďže front end aplikácie je napísaný v jazyku Javascript, je potrebné dáta nejakým spôsobom predspracovať, a práve na to slúži model analýza najazdených okruhov, kde sa dáta znormalizujú, odstránia sa stĺpce s nepotrebnými údajmi a logy, pri ktorých auto úplne

zastavilo. Separujú sa jednotlivé kolá okruhu a exportujú sa do súboru JSON aby ich bolo možné zobraziť na frontende.

Implementácia

Dáta je v prvom rade potrebné transformovať do dátového rámca, ktorý využíva Python v knižnici pandas. Dáta ďalej musia byť normalizované a niektoré atribúty upravené tak aby sa dalo s nimi lepšie pracovať. Zbytočné stĺpce obsahujúce nepotrebné informácie je vhodné odstrániť. Taktiež sú odstránené všetky riadky, kde auto úplne zastavilo. Aby boli riadky medzi sebou správne poprepájané je vytvorený dodatočný stĺpec odkazujúci na pozíciu, ktorá sa nachádza v ďalšom riadku. Jednotlivé kolá okruhu sú oddelené podľa počiatočnej pozície a normalizované na zobrazenie v grafe. Nakoniec sú spracované dáta vyexportované v súbore typu JSON a pripravené na zobrazenie na frontende aplikácie.

Testovanie

Pre testovanie budeme využívať jednotkové testy, ktorými budeme testovať jednotlivé čiastkové funkcionality aplikácie.

4.3 Optimalizácia steering commandu

Analýza

Na správne určenie steering commandu je potrebné, využiť metódu strojového učenia, pomocou ktorej vieme auto naučiť udržiavať referenčnú trasu v reálnych podmienkach.

Návrh

Pre našu aplikáciu bolo potrebné navrhnúť vhodnú metódu strojového učenia, ktorá by bola schopná správne určovať polohu steering comandu a v čo najkratšom čase. Na základe podrobnejšieho skúmania rôznych prístupov pomocou strojového učenia sme sa rozhodli zvoliť prístup pomocou učenia posilňovaním.

Implementácia

Strojové učenie je implementované pomocou reinforcement learningu, teda učenia posilňovaním podľa vzoru šoférovania auta. Na výpočet správnej polohy po aplikovaní steering commandu je využitý model bicykel.

Testovanie

Počas strojového učenia nie je možné testovať funkčnosť ani overiť korektnosť riešenia. Výsledky bude možné zhodnotiť na základe reálneho správania v teréne.

5 Výsledky testovania

Keďže náš projekt je datovo-orientovaný, naša aplikácie spracováva relatívne veľké množstvo dát. Preto počas testovania sme sa sústredili na to aby dáta boli spracované v optimálnom časovom úseku, aby používateľ dlho nečakal na výsledok po request-e. Na sledovanie toho, či naše čas odozvy našich funkcionalít je optimálny, sme spravili výkonnostné testovanie. Hlavné testovanie sa týkalo funkcionalít analýzy najazdených okruhov. Čiže, rozdelenie najazdených okruhov a zistenie odlišností okruhov od referenčnej trasy podľa metriky priemerná vzdialenosť bodov od trasy.

Na výkonnostné testovanie sme použili nástroj **cProfile** a knižnicu **timeit** v programovacom jazyku Python. **cProfile** nám dovolil vykonať podrobné profilovanie funkcií. Knižnicu **timeit** sme využili na zistenie času behu jednotlivých funkcií.

Tabuľka testovania

	Starý čas výkonania (s)	Nový čas výkonania (s)	Percentuálne zrýchlenie
Rozdelenie okruhov	40	0.032	125000%
Zistenie odlišností okruhov	171	0.069	247826%

Rozdelenie okruhov prebiehalo nad 3000 bodmi.

Zistenie odlišností prebiehalo nad 306 bodmi.

Čiže, podľa výsledkov vieme povedať, že podarilo sa nám získať obrovské zrýchlenie, a používateľ sa bude cítiť komfortne pri použití funkcionalít.

6 Nastavenie prostredia na vývoj

Na spustenie aplikácie je potrebný Python minimálne vo verzii 3, Npm Package Manager, a Matlab s komponentami Automated Driving Toolbox, Robotics System Toolbox a ich závislosťami. Aplikáciu odporúčame otvoriť v IDE prostredí PyCharm.

V príkazovom riadku je následne potrebné spustiť tieto príkazy:

npm install pip install -r requirements.txt

7 Inštalačná príručka

Pre inštaláciu aplikácie CarComp stačí stiahnuť súbor .zip, v ktorom sa nachádza celá aplikácia. Zip súbor si uložte na známe miesto v počítači, ideálne na pracovnú plochu. Po kliknutí pravým tlačidlom myši na .zip súbor sa Vám otvorí menu, v ktorom kliknite na "Extrahovať do...". Po zvolení sa vám na rovnakom mieste ako uložený .zip súbor vytvorí adresár s rovnakým menom. Po otvorení adresára dvojitým kliknutím ľavého tlačidla myši môžete vidieť všetky potrebné súbory k správnemu fungovaniu aplikácie CarComp. Pre spustenie aplikácie stačí dvojité kliknutie ľavým tlačidlom myši na .exe súbor, ktorý spustí aplikáciu, ktorú môžete používať.

8 Používateľská príručka

Po dokončení inštalácie programu a po úspešnom spustení podľa inštalačnej príručky, môžete vidieť hlavné menu aplikácie, na ktorom môžete vidieť 2 tlačidlá "Analýza dráhy" a "Optimalizácia komponentov". (Viz. obr. 2.)



Obr. 2: Hlavné menu CarComp

Analýza dráhy

Po kliknutí na tlačidlo "Analýza dráhy" môžete vidieť nové kontextové okno pre analýzu dráhy. Okno obsahuje 2 tlačidlá "Vložiť referenčnú dráhu" a "Vložiť LOG súbor". (viz. obr. 3)

"Vložiť referenčnú dráhu" – Po kliknutí na tlačidlo môžete vidieť kontextové okno, kde musíte zadať umiestnenie "log" súboru, ktorý obsahuje dáta o referenčnej dráhe. Po vložení súboru môžete na spodku kontextového okna vidieť vizualizáciu referenčnej dráhy. (viz. obr. 4)

"Vložiť LOG súbor" – Po kliknutí na tlačidlo môžete vidieť kontextové okno, kde musíte zadať umiestnenie ".log" súboru, ktorý obsahuje dáta o zaznamenanej jazde na referenčnej dráhy. Po vložení súboru môžete na spodku kontextového okna vidieť vizualizáciu záznamu a zároveň na pravej strane pod nadpisom "List" môžete vidieť detailnejšie informácie o jednotlivých okruhoch ako jeho dĺžku a iné. Po kliknutí na určitý okruh môžete vidieť zmenu vizualizácie, ktorá teraz ukazuje Vami vybraný okruh na referenčnej trase. (Viz. obr. 4.)

CarComp				-	o x
< Dráha				Ŧ	ē
Vložiť refere	nčnú dráhu	Viožiť LOG súbor	List		
Počet	Štatistika bodov na referenčnej trase Počet	trás v analyzovanom súbore			

Obr. 3: Analýza dráhy

CarComp				-	o x
< Dráha				ŧ	•
	Vložiť referenčnú dráhu	Vložiť LOG súbor	List		
	kartfinal36pr3r	200629130554_gps	 Kolo 1 - 4140 bodov, priemerná odchýlka: 0.728m Kolo 2 - 1551 bodov, priemerná odchýlka: 0.96m Kolo 2 - 200 kato priemerná odchýlka: 0.96m 		
	Štatis Počet bodov na referenčnej trase 1421	stika Počet trás v analyzovanom súbore 34	 Kolo 3 - 3790 bodov, priemerňa odchlyka: 0.42km Kolo 4 - 1994 bodov, priemerňa odchlyka: 0.74m Kolo 5 - 1820 bodov, priemerňa odchlyka: 0.689m Kolo 6 - 1442 bodov, priemerňa odchlyka: 0.689m Kolo 7 - 1833 bodov, priemerňa odchlyka: 0.680m Kolo 7 - 1833 bodov, priemerňa odchlyka: 0.803m 		
1421 34			 Kolo 9: 1652 bodov, priemenia dodržika. 0.742m Kolo 11: 172 bodov, priemenia dodržika. 0.782m Kolo 11: 1512 bodov, priemenia dodržika. 0.782m Kolo 12: 1675 bodov, priemenia dodržika. 0.882m Kolo 13: 1281 bodov, priemenia dodržika. 0.882m Kolo 15: 1282 bodov, priemenia dodržika. 0.882m Kolo 15: 1184 bodov, priemenia dodržika. 0.812m Kolo 15: 1182 bodov, priemenia dodržika. 0.912m Kolo 12: 1155 bodov, priemenia dodržika. 0.912m Kolo 22: 1255 bodov, priemenia dodržika. 0.812m Kolo 23: 1255 bodov, priemenia dodržika. 0.812m Kolo 24: 1255 bodov, priemenia dodržika. 0.813m Kolo 34: 1312 bodov, priemenia dodržika. 0.813m Kolo 34: 1135 bodov, priemenia dodržika. 0.813m Kolo 34: 1315 bodov, priemenia dodržika. 0.813m Kolo 34: 1135 bodov, priemenia dodržika. 0.813m Kolo 34: 1135 bodov, priemenia dodržika. 0.8142m 	n 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	

Obr. 4: Oranžová čiara predstavuje dáta z referenčeéj dráhy a modré čiary predstavujú dáta z jednotlivých okruhov

"List" – V liste môžeme vidieť štatistiku pre jednotlivé okruhy z vloženého .log súboru. Štatistiky obsahujú počet bodov pre daný okruh a priemernú odchýlku v metroch. Na každý okruh môžeme kliknúť pre vizualizáciu konkrétneho kola, ktorá sa zobrazí pod pôvodným grafom. Schovať vizualizáciu môžeme opätovným kliknutím na vybraný okruh. (Viz. obr. 5.)



Obr. 5: Modrá čiara vizualizuje dáta vybraného 33. okruhu z listu

Optimalizácia komponentov

Funkcionálna časť "Optimalizácia komponentov" je v momentálnom štádiu len v dizajnovom návrhu. Plná funkcionalita bude dodaná v rámci letného semestra.

Po kliknutí na tlačidlo "Optimalizácia komponentov" môžete vidieť nové kontextové okno pre optimalizáciu komponentov. Okno obsahuje 3 tlačidlá "Koleso", "Volant" a "Brzdy". (Viz. obr. 6.)



Obr. 6: Optimalizácia komponentov

Po vybraní komponentu, ktoré chcete optimalizovať sa Vám zobrazí nové kontextové okno pre daný komponent s tlačidlom "Vložiť LOG súbor". Po kliknutí na tlačidlo môžete vidieť kontextové okno, kde musíte zadať umiestnenie ".log" súboru, ktorý obsahuje dáta o zvolenom komponente. Po vložení .log súboru môžete vidieť vizualizáciu dát pomocou grafu. Graf obsahuje 2 čiary – modrú a slabo modrú. Modrá farba predstavuje vizualizáciu nameraných dát a slabo modrá predstavuje vizualizáciu simulovaných dát. (Viz. obr. 7.)



Obr. 7: Vizualizácia dát pre komponent

V pravom hornom rohu môžeme vidieť tlačidlo so šípkou ukazujúcou dole a ikonu tlačiarne. **Šípka ukazujúca dole** – Toto tlačidlo Vám stiahne vizualizáciu, list a štatistiku vytvorenú z vložených dát do pamäte vášho počítača

Ikona tlačiarne – Toto tlačidlo Vám dokáže vytlačiť vizualizáciu, list a štatistiku vytvorenú z vložených dát za predpokladu, že máte funkčnú a zapojenú tlačiareň.