

Západočeská univerzita v Plzni
Fakulta aplikovaných věd
Katedra kybernetiky



Bakalářská Práce

Návrh řídicího systému pro autonomní RC automobil

Autor: Adam Vodrážka

Studijní program: Kybernetika a řídicí technika

Vedoucí práce: Ing. Flídr Miroslav, Ph.D.

Plzeň, 2024

ZÁPADOČESKÁ UNIVERZITA V PLZNI

Fakulta aplikovaných věd
Akademický rok: 2023/2024

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

(projektu, uměleckého díla, uměleckého výkonu)

Jméno a příjmení: **Adam VODRÁŽKA**
Osobní číslo: **A21B0409P**
Studijní program: **B0714A150005 Kybernetika a řídicí technika**
Specializace: **Automatické řízení a robotika**
Téma práce: **Návrh řídicího systému pro autonomní RC automobil**
Zadávající katedra: **Katedra kybernetiky**

Zásady pro vypracování

1. Seznámení s matematickým modelem RC automobilu.
2. Návrh a realizace metody plánování cesty závodní tratí pro autonomní RC automobil.
3. Návrh a realizace řídicího systému pro průjezd trati autonomním RC automobilem.

Rozsah bakalářské práce: **30-40 stránek A4**
Rozsah grafických prací:
Forma zpracování bakalářské práce: **tištěná/elektronická**

Seznam doporučené literatury:

* Liniger, A., Domahidi, A., and Morari, M. (2015) Optimization-based autonomous racing of 1:43 scale RC cars. Optim. Control Appl. Meth., 36: 628–647. doi: 10.1002/oca.2123.

* Curinga, F. (2017). Autonomous racing using model predictive control (Dissertation). URL:
<https://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-222801>

* Liniger, A. (2018). Path Planning and Control for Autonomous Racing. ETH Zurich. doi: 10.3929/ethz-b-000302942

Vedoucí bakalářské práce: **Ing. Miroslav Flídr, Ph.D.**
Katedra kybernetiky

Datum zadání bakalářské práce: **17. října 2023**
Termín odevzdání bakalářské práce: **20. května 2024**



Doc. Ing. Miroslav Flídr, Ph.D.
děkan



Doc. Dr. Ing. Vlasta Radová
vedoucí katedry

V Plzni dne 17. října 2023

Prohlášení

Předkládám tímto k posouzení a obhajobě bakalářskou práci zpracovanou na závěr studia na Fakultě aplikovaných věd Západočeské univerzity v Plzni.

Prohlašuji, že jsem bakalářskou práci vypracoval samostatně a výhradně s použitím odborné literatury a pramenů, jejichž úplný seznam je její součástí.

V Plzni dne 20. května 2024

.....
Podpis autora

Poděkování

Tímto bych chtěl poděkovat panu Ing. Miroslavu Flídrovi, Ph.D. za odborné vedení mé práce, poskytování cenných rad a připomínek. V neposlední řadě za veškerý věnovaný čas.

Abstrakt

Tato bakalářská práce se zabývá návrhem řízení pro autonomní RC autíčko jezdící po trati, s účelem volby a sledování závodní trajektorie a minimalizace času, potřebného pro projetí tratě. Čtenáři je popsán model autíčka a je představeno prediktivní řízení (MPC - model predictive control), to je způsob řízení na základě predikce chování z modelu. Dále bude představen způsob získávání referenční trajektorie, kterou se bude autíčko snažit následovat.

Klíčová slova: MPC, řízení autonomního auta, prediktivní řízení, autonomní závodění, plánování trajektorie autonomního auta

Abstract

This bachelor's thesis focuses on designing control for an autonomous RC car driving on a track, aiming at selecting and tracking the racing trajectory while minimizing the time needed to complete the course. The reader is introduced to the car model, and a control method based on predicting the model's behavior (MPC - model predictive control) is presented. Furthermore, a method for obtaining a reference trajectory that the car will attempt to follow will be introduced.

Key words: MPC, predictive control, autonomous car control, autonomous racing, path planning, path tracking

Obsah

Obsah	7
1 Úvod	8
2 Model automobilu	11
2.1 Kinematický model	11
2.2 Dynamický model	14
3 Řídicí algoritmus	20
3.1 Dvě vrstvy regulátoru	21
3.2 Plánování referenční trajektorie	21
3.3 Sledování trajektorie - MPC regulace	22
4 Implementace algoritmu řízení - experiment	25
4.1 Parametry modelu	25
4.2 Generátor závodních tratí	26
4.3 Simulační prostředí (Simulation GUI)	27
4.4 Programové detaily implementace úlohy	28
4.5 Provedení experimentu	29
4.6 Volba parametrů optimalizační úlohy	32
4.7 Představení aplikace	32
4.8 Příklady realizace zajímavých tratí	34
5 Závěr	37
6 Reference	38

Kapitola 1

Úvod

Tato bakalářská práce se zaměřujeme na algoritmy autonomního řízení vozidel aplikované pro závodní účely, kde je potřeba kombinovat pokročilé technologie řízení se specifickými požadavky závodního prostředí. V následujícím odstavci bude představena autorova motivace k práci a její cíl.

Již od dětství mne vždy fascinovaly automobily, a to zejména kvůli jejich komplexnosti, která jako celek tvoří například dopravní vozidlo nebo hračku pro dospělé (sportovní vůz). Vždy jsem byl ohromen množstvím technologie z různých inženýrských odvětví vměstnaných na mobilní platformu o velikosti pár jednotek čtverečních metrů.

Přesuneme-li se k závodění, tak schopnost závodníků projet okruh extrémně vysokými rychlostmi s přetíženími, které se podobají letu ve stíhačce, si určitě taktéž zaslouží obdiv. Tito profesionálové trénují správné projetí zatáčky klidně i pár let a i přesto jsou to pořád jen lidé a může se stát, že se při závodě z různých důvodů například netrefí do nejrychlejší stopy, nebo důvodem smyku skončí dokonce mimo trať.

Řízení automobilu je skvělý příklad, na kterém lze aplikovat mnou studovanou teorii řízení a zároveň prohloubit své znalosti v závodění a technikách spojených například s výběrem optimálních trajektorií průjezdu zatáčkou, nebo předjíždějících manévrů. Analýzou matematického modelu vozidla rozšířit své znalosti o způsobech modelování reálných procesů, což je v mém oboru jeden z nejdůležitějších kroků pro návrh kvalitního řízení. Nakonec možnost pozorovat souvislosti mnou navrhovaného řízení a například autonomního řízení aut v závodních hrách. Cílem práce je navrhnout algoritmus řízení, který bude schopný projíždět trať stylem jízdy podobajícímu se závodění sportovních automobilů za účelem minimalizace času potřebného pro průjezd tratí. V dalších odstavcích bude následovat uvedení do tématu a představení, v práci používaných, odborných pojmů, pro lepší pochopení problematiky, rozebírané v následujících částech práce.

V nedávné době jsme byli svědky pozoruhodného pokroku v oblasti autonomních technologií vozidel, jak ve výzkumu, tak v komerčních aplikacích. Společnosti jako Tesla a Volvo představují neustále se zdokonalující autonomní řešení pro veřejné silnice. Tato práce se ale bude zabývat autonomními technologiemi v oblasti závodních scénářů. Závody vyžadují sofistikované systémy řízení, které dokážou v dynamickém prostředí navigovat vozidla blízko svých limitů, integrujíc pokročilé plánování trasy a vyhýbání se překážkám v reálném čase, přičemž dodržují přísné výpočetní omezení.

Tento vstup do světa autonomních závodů je ukázkově demonstrován prací [1], kde autoři zkoumají optimalizační řídicí algoritmy pro závodní autonomní vozidla. Jejich pozornost je zaměřena na vývoj regulátorů schopných pracovat v reálném čase na vestavěných platformách.

Experimentální výsledky na RC závodních autech značky Kyosho typu Dnano o měřítku 1:43 ukazují, že nasazené regulátory dosahují impozantních výsledků - rychlosti aut přesahující 3 m/s což je přibližně 465 km/h, přičemž operují na mezi kluzu - styl jízdy prováděny zkušenými lidskými řidiči s rozsáhlým tréninkem.

V uvedené práci autoři dále stanovili optimalizační úlohu, řešenou později implementovanými regulátory pro autonomní závodní vozy, které jsou schopny efektivně začleňovat vyhýbání se překážkám, provádět předjížděcí manévry, pracovat s omezeními tratě a umožňovat řízený drift. Tyto algoritmy řízení jsou navrženy s důrazem na výpočetní efektivitu v reálném čase, což bylo potvrzeno experimentálními výsledky na závodních RC autech.

Prediktivní řízení (MPC - Model Predictive Control) hraje klíčovou roli v kontextu autonomního závodění při optimalizaci řízení vozidla, díky zohlednění jeho budoucího chování. Tento způsob řízení realizuje minimalizaci kritéria v omezeném časovém horizontu, což je zásadní pro řízení autonomních vozidel na závodních tratích. Tato technika přináší efektivní způsob, jak řídit vozidlo v dynamickém prostředí, zohledňující jeho model a okolní podmínky. Pro dosažení optimálního řízení je klíčové ladění parametrů kritéria kvality a výběr vhodného solveru pro řešení optimalizačního problému konkrétní úlohy autonomního závodění.

V práci [9] je rozebráno téma hierarchického řízení, jedná se o spolupráci vyšší úrovně řízení, která plánuje trajektorii s konečným horizontem pomocí matematického modelu a poté druhá úroveň, která implementuje prediktivní řízení a sleduje naplánovanou trajektorii vygenerovanou vyšší vrstvou. Často pozorujeme, že nižší úroveň zahrnuje podrobný model vozidla, zatímco vyšší úroveň je postavena na zjednodušených modelech. Použití MPC regulátoru v nižší úrovni je běžné díky možnostem, jak lze do řízení zahrnout omezení daná jak modelem (maximální úhel natočení kol nebo například zrychlení), tak prostředím (překážky nebo okraje trati.). Hlavním rozdílem mezi různými variantami takových regulátorů je, jak je nelineární dynamika vozidla zohledněna v řešení MPC problému. Nejoblíbenější techniky jsou buď přímo používat nelineární dynamiku a spoléhat se na solver pro úlohu nelineárního programování (NLP) [16], nebo používat aproximační techniky, jako je například lineární časově variantní (LTV) model, což vede na úlohu konvexního kvadratického programování (QP). Přístup v této práci bude realizovat LTV aproximaci, převážně kvůli výhodě nižších výpočetních nároků.

Důležitou částí je výpočet referenční trajektorie. Ta je vypočtena na základě matematického modelu vozidla a zároveň s respektem ke všem omezením jako jsou překážky na trati, její okraje a dále omezení na řídicí vstupy vozidla. Tento přístup umožňuje autonomnímu vozidlu sledovat optimální trajektorii, maximalizovat svůj postup po trati a efektivněji reagovat na měnící se prostředí.

Cílem této práce je navrhnout autonomní řízení vozidla pro projetí závodní trati. Pro zjednodušení se u trati se předpokládá, že leží v rovině, je uzavřená a existuje její ohraničení. Zároveň by se na trati neměly objevovat příliš prudké zatáčky, které nedovolí mechanická konstrukce vozidla projet. Jelikož se v této práci věnujeme pouze simulaci, bude vždy známá centrální linie a ohraničení trati reprezentováno body se souřadnicemi $[x, y]$.

Tato práce se bude soustředit pouze na simulační prostředí, tudíž nejsou definované žádná pevná omezení na model vozidla. Stanovíme si tedy vlastní parametry vozidla. Na základě takto stanoveného modelu, je cílem navrhnout optimální trajektorii závodního charakteru, minimalizující čas projetí dané sekce. Tuto trajektorii poskytnout jako referenci implementovanému algoritmu řízení, který na základě minimalizace řídicího kritéria vypočte optimální řídicí vstupy, což je natočení kol a zrychlení vozidla, a zajistí sledování této trajektorie a udržení vozidla v ohraničené trati.

V práci bude nejprve představen kinematický matematický model vozidla, který popisuje

dostatečně přesně pohyb pro vozidlo, které není v blízkosti svých limitů přilnavosti k vozovce. Následně bude představena možnost jak kinematický model doplnit o dynamické složky. Bude tedy představen dynamický model, díky kterému lze dosáhnout přesnějšího popisu chování vozidla, které se pohybuje blízko svých mezí přilnavosti. K získání dynamického modelu je zapotřebí představit model pneumatiky, na jehož základě lze spočítat síly působící na pneumatiky, kdy právě tyto síly pak do modelu vnášejí dynamiku. Nevýhodou dynamického modelu je však velké množství neznámých parametrů popisující vlastnosti pneumatiky a také použitého motoru. Následně bude provedena linearizace a diskretizace použitého modelu. V další části práce bude představena technika plánování referenční trajektorie a následně její použití v řídicím algoritmu, který realizuje řízení použitím techniky prediktivního řízení. Na konec práce bude proveden experiment, kdy všechna zmíněná teorie bude naprogramována do simulačního prostředí, za účelem realizovat řízení autonomního vozidla projíždějící závodní tratí.

Kapitola 2

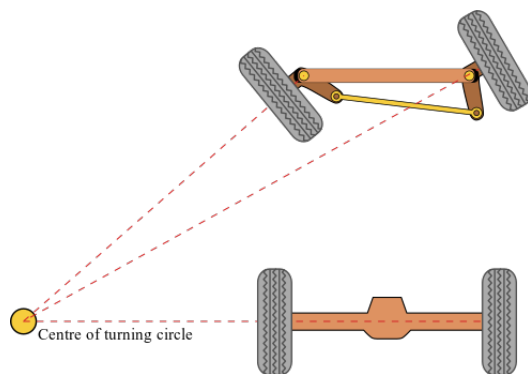
Model automobilu

V této kapitole bude představena problematika modelování automobilů. Pro simulaci pohybu automobilu na trati a řešení optimalizačního problému algoritmem prediktivního řízení (MPC) je klíčové zvolit vhodný model vozidla. Kapitola se zaměřuje na různé přístupy modelování, nejprve bude představen kinematický model a následně předvedené jeho možné doplnění o dynamické složky. Tyto modely se liší ve své složitosti, přesnosti a schopnosti simulovat chování vozidla při různých stylech jízdy různou rychlostí. Kapitola dále pojednává o modelování pneumatik a s nimi spojenými silami, působící na vozidlo, jejichž zavedením se přechází od kinematického modelu k dynamickému. Nakonec bude provedena linearizace a diskretizace kinematického modelu. Jelikož se tato práce zabývá pouze simulačním prostředím, bude vozidlo simulováno kinematickým modelem, z důvodu velkého počtu neznámých parametrů v případě použití dynamického modelu. Tyto parametry se zjišťují empiricky na reálném objektu.

2.1 Kinematický model

Pro efektivní řešení optimalizační úlohy je klíčové definovat model vozidla. Tato část se zaměřuje na představení modelu jízdního kola, který dostatečně přesně popisuje chování automobilu při pohybu na rovné ploše, jak je prezentováno v publikacích [1] a [2]. Omezením na dvourozměrný prostor (pohyb v rovině) se dopouštíme zanedbání náklonů (bočních i směrových). Model spočívá v zjednodušení karosérie vozidla na tuhé těleso s hmotností m a momentem setrvačnosti I_z , který je orientován vůči ose Z , tj. osu kolmou na rovinu XY . Model vozidla disponuje pouze jedním předním a jedním zadním kolem. Proměnné zatížení vnějších stran vozidla při zatáčení je také zanedbáno.

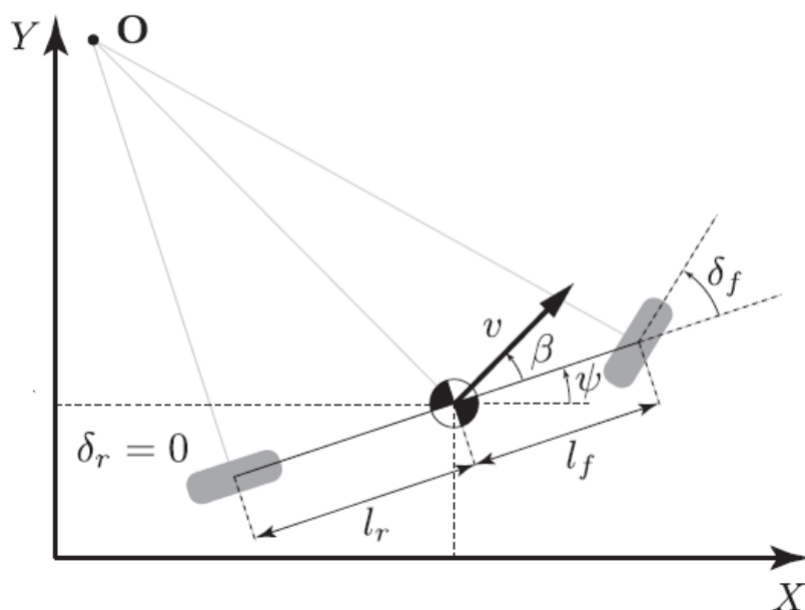
Použitím modelu bicyklu dochází ještě k jednomu výraznému zjednodušení. Tím že sjednocujeme obě kola nápravy do jednoho, nemusíme uvažovat složitosti spojené s Ackermannovo modelem zatáčení [10], který spočívá v odlišné rychlosti vnějších a vnitřních kol automobilu při průjezdu zatáčkou a dále také rozdílnému natočení každého z předních kol, jak lze vidět v náčrtu na obrázku 2.1.



Obrázek 2.1: Naznačení Ackermannova modelu řízení [10]

Model vozidla bude použit jak k simulaci chování automobilu na trati, tak při řešení optimalizačního problému MPC. Obecně lze prohlásit, že model používaný k výpočtu řízení bývá často zjednodušen oproti tomu simulačnímu, protože použití složitějších modelů s více stavy má významný vliv na složitost řešení optimalizačního problému.

Odvození diferenciálních rovnic kinematického modelu



Obrázek 2.2: Grafické schéma pro zavedení modelu bicyklu [3]

Rám spojuje kola bicyklu a jeho počátek je umístěn v jeho centroidu. Na obrázku 2.2 lze pozorovat základní parametry modelu. Vektor rychlosti v ve směru, kterým se model bicyklu pohybuje. Úhel mezi rámem a směrem pohybu v značený β . Úhel natočení rámu ψ v prostoru vzhledem k ose X . Vzdálenost zadních kol l_r a předních l_f od centroidu bicyklu. Úhel natočení předních kol δ_f a zadních kol δ_r . Uvedený kinematický model vychází z popisu uvedeném v [3] a [4].

Kinematika systému z obrázku 2.2 je pak dána těmito rovnicemi

$$\begin{cases} \dot{X} &= v \cos(\psi + \beta) \\ \dot{Y} &= v \sin(\psi + \beta) \\ \dot{\psi} &= \frac{v}{l_r} \sin(\beta) \\ \dot{v} &= a \end{cases} \quad (2.1)$$

Kde úhel β lze vypočítat následujícím vztahem

$$\beta = \tan^{-1} \left(\frac{l_r}{l_f + l_r} \tan(\delta_f) \right) \quad (2.2)$$

Stav systému je tedy definován jako vektor

$$\mathbf{x} = \begin{pmatrix} X \\ Y \\ \psi \\ v \end{pmatrix} \quad (2.3)$$

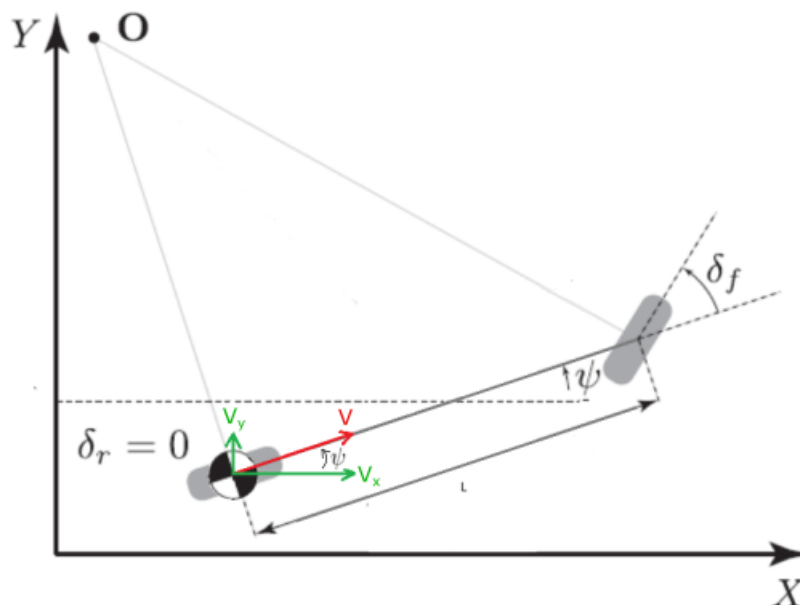
Vstup systému lze definovat jako vektor zrychlení a natočení předních kol

$$\mathbf{u} = \begin{pmatrix} a \\ \delta_f \end{pmatrix} \quad (2.4)$$

Kinematický model se zaměřuje na statické vlastnosti vozidla a podle definice nezahrnuje dynamiku vozidla jako je například modelování sil působících na pneumatiky. Předpokládá, že vozidlo perfektně drží na ploše, po které se pohybuje (žádné smýkaní). Tento model poskytuje uspokojivé výsledky při jízdě vozidla běžnou rychlostí.

Úprava modelu - zjednodušení

Předchozí formulace daná vztahy (2.1) může být zjednodušena, pokud bychom uvažovali přesun počátku souřadného systému bicyklu do zadního kola bicyklu. Myšlenka je zachycena následujícím obrázkem:



Obrázek 2.3: Model bicyklu se souřadným systémem v zadní kole

Uvažováním souřadného systému umístěným v zadním kole dojdeme k zjednodušenému modelu a dostáváme následující diferenciální rovnice popisující kinematický model:

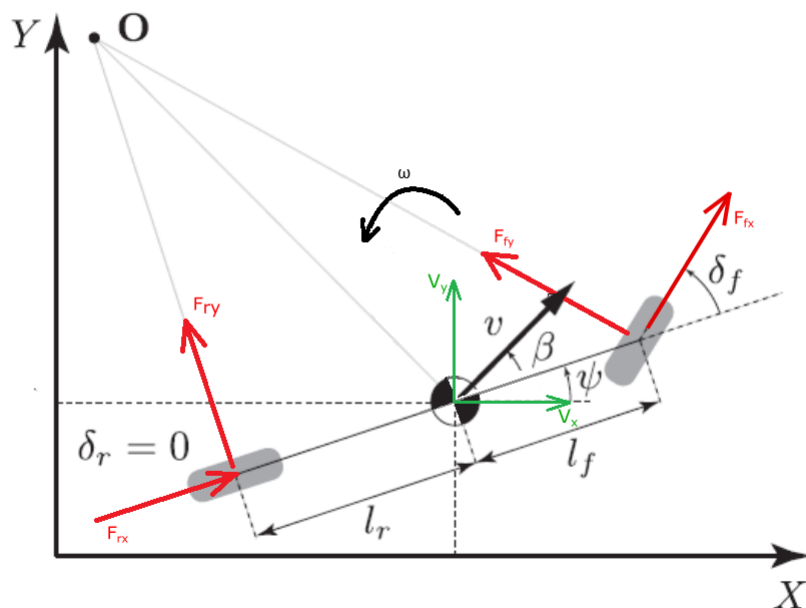
$$\begin{cases} \dot{X} = v \cos(\psi) \\ \dot{Y} = v \sin(\psi) \\ \dot{\psi} = \frac{v}{L} \tan(\delta_f) \\ \dot{v} = a \end{cases} \quad (2.5)$$

Kde L označuje rozvor. Ostatní veličiny nesou stejný význam jako v předchozím modelu, nově zde figurují V_x a V_y , které zobrazují složky vektoru rychlosti V vzhledem k osám X a Y . Vektor stavu společně se vstupním vektorem se nemění.

2.2 Dynamický model

Ačkoliv byl v praktické části této práce kvůli jednoduchosti a pohybu pouze v simulačním prostředí použit kinematický model, určitě stojí za zmínku jeho doplnění o dynamické složky. Kinematický model poskytuje uspokojivé výsledky, když není rychlost vozidla a úhel řízení extrémní. Může se stát nedostatečným při přivedení vozidla blízko svým rychlostním limitům, v takovém okamžiku začínají pneumatiky ztrácet přilnavost k silnici (to je označováno jako driftování). Pro simulaci realistického chování vozidla je vhodné do modelu zakomponovat síly působící na model, čímž přejdeme od kinematického modelu na dynamický. Navíc je v ideálním případě vhodné zjednodušenou verzi těchto dynamik integrovat i do modelu používaného pro MPC, tím by byla získána vlastnost regulátoru, například předpovídat kdy jsou pneumatiky na pokraji své přilnavosti k vozovce a na základě toho vhodněji generovat akční zásahy.

Na následujícím obrázku jsou k vidění použité značky pro popis dynamického modelu bicyklu.



Obrázek 2.4: Grafické schéma dynamického modelu bicyklu [1]

Stejně jako v případě uvedeného kinematického modelu 2.2 uvažujeme, že rám spojuje kola bicyklu a jeho počátek je umístěn v jeho centroidu. Tento model zavádí navíc ω což je označení úhlové rychlosti otáčení rámu bicyklu, které bylo již popsáno v předchozím modelu (2.5). Dále zde figurují síly působící na přední a zadní pneumatiky F_f a F_r , rozdělené na složky x a y vzhledem k osám X a Y.

Diferenciální rovnice modelu jsou odvozeny v [1] a [6] a mají následující tvar

$$\begin{cases} \dot{X} &= v_x \cos(\psi) - v_y \sin(\psi) \\ \dot{Y} &= v_x \sin(\psi) + v_y \cos(\psi) \\ \dot{\psi} &= \omega \\ \dot{v}_x &= \frac{1}{m}(F_{r,x} - F_{f,y} \sin(\delta_f) + mv_y \omega) \\ \dot{v}_y &= \frac{1}{m}(F_{r,y} - F_{f,y} \cos(\delta_f) - mv_x \omega) \\ \dot{\omega} &= \frac{1}{I_z}(F_{f,y} l_f \cos(\delta_f) - F_{r,y} l_r) \end{cases} \quad (2.6)$$

Vektor stavu odpovídající matematickému popisu modelu se pak skládá z těchto složek

$$\mathbf{x} = \begin{bmatrix} X \\ Y \\ \psi \\ v_x \\ v_y \\ \omega \end{bmatrix} \quad (2.7)$$

Vstupní vektor definujeme jako

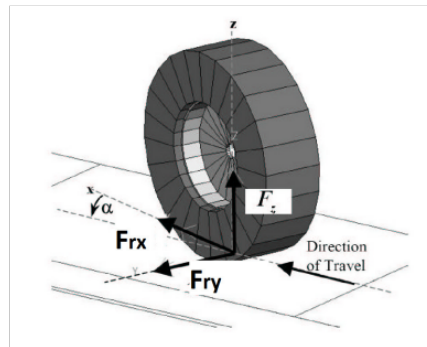
$$\mathbf{u} = \begin{pmatrix} F_{rx} \\ \delta_f \end{pmatrix} \quad (2.8)$$

Takto formulovaný model již uvažuje dynamické síly od pneumatik spojené s měnícím se natočením vozidla. Vektor stavu nyní obsahuje proměnné v_x , v_y a ω , které v sobě zahrnují působení zavedených sil, což vede na lepší přiblížení k realitě, když je vozidlo blízko svých hranic přilnavosti [1], [4]. Ve vstupním vektoru lze pozorovat zaměnění zrychlení a za sílu F_{rx} , to je způsobeno tím, že v dynamickém modelu samotné zrychlení nevystupuje, je zde schováno ve vztahu $F = m \cdot a$, vycházejícího z druhého Newtonova zákona. Také to znamená, že v takovém případě by byla poháněna pouze zadní náprava a ostatní síly by tedy vycházeli pouze ze samotného pohybu vozidla. Síly F modelují interakci pneumatik s vozovkou a jsou určeny zčásti empirickým způsobem, jak je podrobně popsáno v následující části.

Model pneumatiky

Pro praktickou realizaci dynamického modelu je zapotřebí znát síly působící na pneumatiku. V této sekci budou popsány dva základní přístupy modelování těchto sil. Nejprve budou graficky znázorněny sledované proměnné a následně představeny způsoby jejich výpočtu.

Následující obrázek představuje značení použité k popisu interakce pneumatik s vozovkou. Tento popis je nezbytný k úplné charakterizaci dynamiky vozidla a tedy k možnosti jeho ovládání na mezi stability. Za určitých podmínek může vozidlo ztratit přilnavost a začít klouzat (driftovat).



Obrázek 2.5: Zobrazení sil působících na pneumatiku (zadní kolo) [2]

V obrázku jsou znázorněny síly působící na zadní pneumatiku F_{rx} a F_{ry} , vzhledem k osám X a Y. Poté je zde naznačena osa pohybu (Direction of Travel) a možné vyosení vůči této ose, například v případě smyky, reprezentováno úhlem α .

Pacejkův model pneumatiky

V případě použití modelu pana Hans B. Pacejka jsou síly na pneumatiku popsány následujícím způsobem:

$$\begin{aligned}
 F_{f,y} &= D_f \sin(C_f \arctan(B_f \alpha_f)) \\
 F_{r,y} &= D_r \sin(C_r \arctan(B_r \alpha_r)) \\
 F_{r,x} &= (C_{m1} - C_{m2} v_x) d - C_r - C_d v_x^2
 \end{aligned} \tag{2.9}$$

kde:

$$\begin{aligned}\alpha_f &= -\arctan\left(\frac{\dot{\psi}l_f + v_y}{|v_x|}\right) + \delta \\ \alpha_r &= -\arctan\left(\frac{\dot{\psi}l_r - v_y}{|v_x|}\right)\end{aligned}\quad (2.10)$$

V těchto rovnicích figurují parametry B = tvrdost pneumatiky, C = koeficient tvaru pneumatiky, D = maximální hodnota krouticího momentu, kterou pneumatika může generovat. Dolní indexy f a r reprezentují přední a zadní pneumatiku. Tyto parametry jsou určeny empiricky, provedením sady experimentů na vozidle. Rovnice pro $F_{r,y}$ a $F_{f,y}$ modelu pneumatiky (2.9) popisují tření pneumatik se silnicí. Poslední rovnice používá parametry stejnosměrného motoru k popisu valivého a tažného odporu zadního kola. Parametry C_{m1} a C_{m2} definují model motoru, C_{rr} představuje valivý odpor a C_d odpor vzduchu [11].

Přesnost získání parametrů figurujících v těchto rovnicích je stěžejní, aby bylo možné zohlednit jev kluzu (driftu). Pokud tento jev věrohodně modelem předvídat, pak je možné vozidlo přivést na hranici stability a tím dosáhnout maximálních rychlostí a závodit efektivně. Nicméně vystihnout reálné chování pneumatik přiřazením odpovídajících parametrů může být obtížné. Navíc je potřeba zohlednit faktor opotřebení a ohřátí pneumatiky, který bude mít také vliv na hodnoty těchto parametrů.

Fialův model pneumatiky

Tento model volí jednodušší metodu přístup, vyžadující méně parametrů k přizpůsobení fyzické pneumatiky matematickému modelu [1]. Stěžejním předpokladem je, že laterální a podélná tuhost pneumatiky C_f a C_r si jsou rovny. Dále se předpokládá, že distribuce tlaku na kontaktní ploše pneumatiky je parabolická. Poté je celkový kluz σ vypočten jako:

$$\sigma = \sqrt{\sigma_y^2 + \sigma_x^2} \quad (2.11)$$

kde σ_x a σ_y jsou podélný a laterální kluz:

$$\sigma_x = \begin{cases} \frac{r\omega_w - v_x}{r\omega_w} & \text{během zrychlování} \\ \frac{r\omega_w - v_x}{V_x} & \text{během brzdění} \end{cases} \quad (2.12)$$

$$\sigma_y = \frac{v_x}{r\omega_w} \tan(\alpha) \quad (2.13)$$

Kde r je poloměr pneumatiky, ω_w je úhlová rychlost pneumatiky, v_x je podélná rychlost vozidla a α je natočení pneumatik vůči směru pohybu vozidla.

Síla F_t je definována vztahem:

$$F_t = \begin{cases} \mu F_z \left(3\theta\sigma - \frac{3}{2}(\theta\sigma)^2 + \frac{1}{3}(\theta\sigma)^3\right) & \text{pro } \sigma \leq \sigma_m \\ \mu F_z & \text{pro } \sigma \geq \sigma_m \end{cases} \quad (2.14)$$

kde $\theta = \frac{1}{\sigma_m} = \frac{C_r}{3\mu F_z}$, σ_m odpovídá mezní hodnotě, kdy ještě nedochází ke kluzu. Poznamenejme, že $C_f = C_r$. Jev kluzu (driftování) se začíná projevovat, když $F_t = \mu F_z$.

V případě čisté bočního skluzu platí $\sigma_y = \tan(\alpha)$ a $\sigma_x = 0$. V případě čisté podélného skluzu platí $\sigma_y = 0$. Poté lze dopočíst následující síly dosazované do původního modelu (2.6)

$$\begin{aligned} F_{r,x} &= \frac{\sigma_x}{\sigma} F_t \\ F_{r,y} &= \frac{\sigma_y}{\sigma} F_t \end{aligned} \quad (2.15)$$

Linearizace a diskretizace kinematického modelu

Výše uvedený dynamický model (2.6) dosahuje lepších výsledků při porovnání s realitou, problémem ale zůstává empirická identifikace parametrů. Jelikož se práce zaměřovala pouze na počítačovou simulaci a nebyl uvažován žádný fyzický model, tak byla úloha zjednodušena použitím kinematického modelu (2.5), který uvažuje konstantní rychlost a centroid umístěný na zadní nápravě vozidla.

Model vozidla k linearizaci je tedy popsán následujícími rovnicemi:

$$\begin{cases} \dot{X} = v \cos(\psi) \\ \dot{Y} = v \sin(\psi) \\ \dot{\psi} = \frac{v}{L} \tan(\delta_f) \\ \dot{v} = a \end{cases} \quad (2.16)$$

S vektorem stavu:

$$\mathbf{x} = \begin{pmatrix} X \\ Y \\ \psi \\ v \end{pmatrix} \quad (2.17)$$

Ve zbytku práce budeme pracovat s linearizovaným modelem (2.16), který je možný popsat následující ODE (Obyčejná diferenciální rovnice):

$$\dot{\mathbf{x}} = \frac{\partial}{\partial \mathbf{x}} \mathbf{x} = f(\mathbf{x}, \mathbf{u}) = \mathbf{A}' \mathbf{x} + \mathbf{B}' \mathbf{u} \quad (2.18)$$

Kde jsou matice \mathbf{A}' a \mathbf{B}' definovány následovně

$$\mathbf{A}' = \begin{bmatrix} \frac{\partial}{\partial x} v \cos(\psi) & \frac{\partial}{\partial y} v \cos(\psi) & \frac{\partial}{\partial v} v \cos(\psi) & \frac{\partial}{\partial \psi} v \cos(\psi) \\ \frac{\partial}{\partial x} v \sin(\psi) & \frac{\partial}{\partial y} v \sin(\psi) & \frac{\partial}{\partial v} v \sin(\psi) & \frac{\partial}{\partial \psi} v \sin(\psi) \\ \frac{\partial}{\partial x} a & \frac{\partial}{\partial y} a & \frac{\partial}{\partial v} a & \frac{\partial}{\partial \psi} a \\ \frac{\partial}{\partial x} \frac{v \tan(\delta)}{L} & \frac{\partial}{\partial y} \frac{v \tan(\delta)}{L} & \frac{\partial}{\partial v} \frac{v \tan(\delta)}{L} & \frac{\partial}{\partial \psi} \frac{v \tan(\delta)}{L} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cos(\bar{\psi}) & -\bar{v} \sin(\bar{\psi}) \\ 0 & 0 & \sin(\bar{\psi}) & \bar{v} \cos(\bar{\psi}) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{\tan(\bar{\delta})}{L} & 0 \end{bmatrix} \quad (2.19)$$

$$\mathbf{B}' = \begin{bmatrix} \frac{\partial}{\partial a} v \cos(\psi) & \frac{\partial}{\partial \delta} v \cos(\psi) \\ \frac{\partial}{\partial a} v \sin(\psi) & \frac{\partial}{\partial \delta} v \sin(\psi) \\ \frac{\partial}{\partial a} a & \frac{\partial}{\partial \delta} a \\ \frac{\partial}{\partial a} \frac{v \tan(\delta)}{L} & \frac{\partial}{\partial \delta} \frac{v \tan(\delta)}{L} \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 1 & 0 \\ 0 & \frac{\bar{v}}{L \cos^2(\bar{\delta})} \end{bmatrix} \quad (2.20)$$

Použitím Eulerovy diskretizační metody získáme diskrétní časový model s časovým krokem dt .

$$\mathbf{x}_{k+1} = \mathbf{x}_k + f(\mathbf{x}_k, \mathbf{u}_k) dt \quad (2.21)$$

Použitím Taylorova rozvoje prvního řádu kolem $\bar{\mathbf{x}}$ a $\bar{\mathbf{u}}$.

$$\begin{aligned}\mathbf{x}_{k+1} &= \mathbf{x}_k + (f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \mathbf{A}'\mathbf{x}_k + \mathbf{B}'\mathbf{u}_k - \mathbf{A}'\bar{\mathbf{x}} - \mathbf{B}'\bar{\mathbf{u}}) dt \\ \mathbf{x}_{k+1} &= (\mathbf{I} + dt\mathbf{A}') \mathbf{x}_k + (dt\mathbf{B}') \mathbf{u}_k + (f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) - \mathbf{A}'\bar{\mathbf{x}} - \mathbf{B}'\bar{\mathbf{u}}) dt\end{aligned}\quad (2.22)$$

Tak získáme diferenční rovnice popisující uvažovaný diskretizovaný systém.

$$\mathbf{x}_{k+1} = \mathbf{A}\mathbf{x}_k + \mathbf{B}\mathbf{u}_k + \mathbf{C} \quad (2.23)$$

Kde matici \mathbf{A} určíme jako:

$$\mathbf{A} = (\mathbf{I} + dt\mathbf{A}') = \begin{bmatrix} 1 & 0 & \cos(\bar{\phi})dt & -\bar{v}\sin(\bar{\phi})dt \\ 0 & 1 & \sin(\bar{\phi})dt & \bar{v}\cos(\bar{\phi})dt \\ 0 & 0 & 1 & 0 \\ 0 & 0 & \frac{\tan(\bar{\delta})}{L}dt & 1 \end{bmatrix} \quad (2.24)$$

a matici \mathbf{B} stanovíme následovně:

$$\mathbf{B} = dt\mathbf{B}' = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ dt & 0 \\ 0 & \frac{\bar{v}}{L\cos^2(\bar{\delta})}dt \end{bmatrix}$$

Na závěr je matice \mathbf{C} definována následujícím předpisem

$$\mathbf{C} = (f(\bar{\mathbf{x}}, \bar{\mathbf{u}}) - \mathbf{A}'\bar{\mathbf{x}} - \mathbf{B}'\bar{\mathbf{u}}) dt \quad (2.25)$$

$$= dt \left(\begin{bmatrix} \bar{v}\cos(\bar{\phi}) \\ \bar{v}\sin(\bar{\phi}) \\ \bar{a} \\ \frac{\bar{v}\tan(\bar{\delta})}{L} \end{bmatrix} - \begin{bmatrix} \bar{v}\cos(\bar{\phi}) - \bar{v}\sin(\bar{\phi})\bar{\phi} \\ \bar{v}\sin(\bar{\phi}) + \bar{v}\cos(\bar{\phi})\bar{\phi} \\ 0 \\ \frac{\bar{v}\tan(\bar{\delta})}{L} \end{bmatrix} - \begin{bmatrix} 0 \\ 0 \\ \bar{a} \\ \frac{\bar{v}\bar{\delta}}{\cos^2(\bar{\delta})} \end{bmatrix} \right) = \begin{bmatrix} \bar{v}\sin(\bar{\phi})\bar{\phi}dt \\ -\bar{v}\cos(\bar{\phi})\bar{\phi}dt \\ 0 \\ -\frac{\bar{v}\bar{\delta}}{L\cos^2(\bar{\delta})}dt \end{bmatrix} \quad (2.26)$$

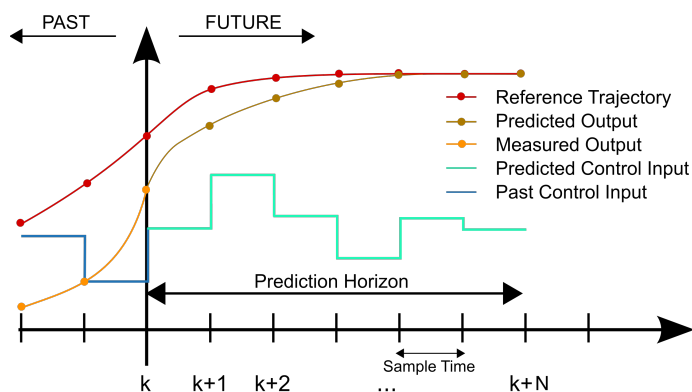
Takto získané linearizované matice \mathbf{A} , \mathbf{B} , \mathbf{C} dále budeme používat pro simulace chování vozidla na trati, tak pro výpočet MPC problému představeného v další kapitole

Kapitola 3

Řídicí algoritmus

Tato kapitola se zabývá pokročilou metodou řízení známou jako prediktivní řízení (MPC), která hraje klíčovou roli v moderních řídicích systémech díky své schopnosti předvídat budoucí chování systému a na základě toho optimalizovat řídicí vstupy v reálném čase. Hlavní pozornost je věnována konceptu dvouvrstvého regulátoru, který rozkládá problém řízení na dvě úrovně: plánování trasy a její následné sledování. Tento přístup umožňuje efektivněji řešit složitou úlohu navigace vozidel na závodní dráze, kde je klíčové najít optimální trajektorii a zároveň na ní co nejpřesněji udržet vozidlo. Prostřednictvím této kapitoly se seznámíme s technikou plánování trasy a teoretickými základy MPC, jeho aplikacemi a výzvami spojenými s jeho implementací v praxi.

Ke sledování trajektorie je použito metody prediktivního řízení neboli MPC. To je metoda používaná v automatizačních úlohách, která na základě předpovídání budoucího chování systému, které je ohodnoceno vhodně zvoleným kritériem optimality, generuje akční vstupy. Tento přístup je zejména vhodný pro systémy s více vstupními veličinami a pro systémy s omezenými hodnotami vstupu a stavu. Bude následovat obrázek zavádějící základní pojmy MPC, které budou vysvětleny v následujícím odstavci



Obrázek 3.1: Pojmy prediktivního řízení [12]

Trajektorie na obrázku 3.1 je vertikálně rozdělena na dvě části: minulost a budoucnost (past, future), to znamená, že vše před časovým okamžikem k se již stalo a trajektorie pro následující časové kroky je predikována. Dále lze pozorovat referenční veličinu (reference trajectory), predikované chování výstupu (predicted output), změřený výstup v minulosti (measured output), minulé akční zásahy (past control input) a predikované akční zásahy (predicted

control input) zajišťující vysledování reference

Predikční model je zásadní součástí celého MPC procesu, jelikož předpovídá chování systému a reakci na různé řídicí vstupy. Tato předpověď je prováděna v rámci určitého časového horizontu, známého jako horizont předpovědi (prediction horizon), který je definován jako konečná doba, po kterou se provádí predikce budoucího stavu systému.

Optimalizační úloha v MPC je řešena v každém okamžiku k na horizont řízení N , s cílem minimalizovat nákladovou funkci. Tato funkce zohledňuje chování systému, jako je sledování požadovaných hodnot, ale také penalizace za porušení různých omezení. Tato omezení mohou zahrnovat limity na vstupy a jednotlivé stavy systému. Výstupem optimalizační úlohy je akční zásah, aplikovaný na systém v okamžiku k .

3.1 Dvě vrstvy regulátoru

V [1] autoři navrhují, rozdělení problému řízení průjezdu tratí na dva menší podproblémy: plánování trasy a sledování trasy. Oba tyto problémy spočívají v řešení určité optimalizační úlohy za účelem minimalizování stanoveného kritéria a tím se snaží zaručit kvalitní řízení.

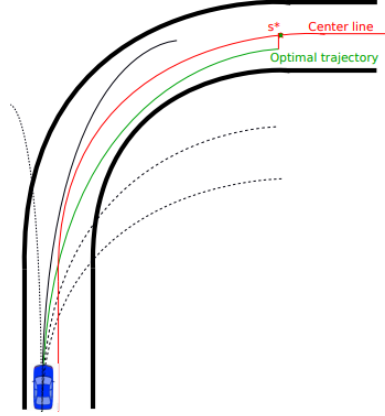
Jak je podrobněji popsáno v [7], pouhé následování středové čáry je špatnou strategií, protože v zatáčkách dochází k významným ztrátám (délka trati při průjezdu u kraje zatáčky je výrazně kratší) a zároveň k významnému poklesu rychlosti, pokud je zatáčka až moc prudká. Právě z těchto důvodů se stává správně naplánovaná trať stěžejní, pro optimální závodní strategii.

Po získání referenční trajektorie je nutné vypočítat sadu vstupů (úhlů řízení a zrychlení), které umožní vozidlu zůstat dostatečně blízko vygenerované referenční závodní linie. Tento problém je klasifikován jako servo problém. Což je řízení za účelem sledování požadované trajektorie.

3.2 Plánování referenční trajektorie

Jak je uvedeno v [8], úkolem plánování trasy je vygenerovat referenční trajektorii, kterou bude vozidlo následovat. V závodním kontextu by tato trasa měla minimalizovat čas průjezdu určité sekce, nebo ekvivalentně maximalizovat pokrok vzhledem ke středové čáře za určitou dobu. Referenční trajektorie je generována řešením optimalizačního problému s omezeními na parametry modelu vozidla a hranice trati.

Nicméně, jak je podrobně popsáno v [1] a [8], tento přístup vede k velmi nelineárním a nekonvexním optimalizačním problémům. Tím obdržíme optimalizační problém, který aktuální nástroje nedokáží v reálném čase vyřešit. Je tedy potřeba zvolit jiný přístup. V [1] autoři použili k modelování trajektorie tabulku konstantních vstupních veličin, zrychlení a úhlů řízení. Aplikací všech kombinací těchto vstupních veličin dostáváme sadu trajektorií, ze které následně vybereme tu optimální. Tímto způsobem sice nelze zaručit nejlepší možnou trajektorii, protože uvažujeme konstantní zrychlení a úhel zatáčení, tím pádem všechny generované trajektorie budou obloukového charakteru a navíc je množina vstupů omezená z důvodu výpočtu úlohy v reálném čase, ale navzdory těmto omezením lze získat referenční trajektorii, která úspěšně maximalizuje postup po dráze a minimalizuje potřebný čas. Ilustrace je zobrazena na následujícím obrázku



Obrázek 3.2: Zobrazení algoritmu plánovače trasy [1]

Na obrázku 3.2 lze pozorovat vizualizaci vozidla, ze které vedou přerušované černé čáry, ty zobrazují jednotlivé vygenerované trajektorie. Trať je ohraničena tlustými černými čarami, kdy uprostřed těchto čar pozorujeme červeně zobrazenou středovou čáru. Optimální trajektorie je v tomto případě označena zelenou barvou, důležité je, že tato trajektorie neopouští trať a zároveň ze všech vygenerovaných trajektorií maximalizuje postup vzhledem ke středové čáře.

K rozhodnutí jaká z vygenerovaných trajektorií bude vybrána zavedeme formulaci problému:

$$x^* = \max_{j \in [1, \dots, M]} P(x_j^k) \quad (3.1)$$

S vazebními podmínkami na stavové veličiny a rozsah vstupních veličin:

$$\begin{aligned} x_{k+1}^j &= f(x_k^j, u_k^j), \quad \forall k \in [t, \dots, t + N - 1] \\ x_k^j &\in X_{\text{track}} \\ u_k^j &\in U(x) \end{aligned} \quad (3.2)$$

Kde index j je z množiny $[1, \dots, M]$ všech možných trajektorií, které mohou vzniknout z množiny vstupů. Označení $x^* = [x_0^*, \dots, x_{N-1}^*]$ je optimální trajektorie, maximalizující postup podél středové čáry. Operátor $P : x \rightarrow P(x)$ je projekce stavu směrem k středové čáře. První podmínka $x_{k+1}^j = f(x_k^j, u_k^j)$ říká, že jsou jednotlivé body trajektorie generovány modelem vozidla na základě předchozího stavu a vstupu. Podmínka $x_k^j \in X_{\text{track}}$ zavádí podmínku, že jednotlivé body trajektorie musí ležet v ohraničené trati. Dále $U(x)$ představuje množinu všech možných vstupů. Tato množina je funkcí aktuálního stavu a respektuje omezení jako je maximální rychlost a zrychlení vozidla. Zjednodušeně lze říci, že na základě modelu vozidla jsou vygenerovány, pomocí tabulky vstupů, různé trajektorie. Tyto trajektorie jsou vyhodnoceny v závislosti na postupu podél středové čáry a vybrána je ta trajektorie, maximalizující postup po trati.

3.3 Sledování trajektorie - MPC regulace

Tato část se bude zabývat problémem držení vozidla na získané trajektorii. K vyřešení tohoto problému bude specifikována optimalizační úloha, jejím vyřešením budou získány akční zásahy, které budou dále aplikovány na vozidlo.

Optimalizační úloha MPC je definována následujícím kritériem J:

$$\begin{aligned}
J_k &= (\mathbf{x}_{k+N,\text{ref}} - \mathbf{x}_{k+N})^T \mathbf{Q}_N (\mathbf{x}_{k+N,\text{ref}} - \mathbf{x}_{k+N}) \\
&+ \sum_{i=k}^{k+N-1} (\mathbf{x}_{i,\text{ref}} - \mathbf{x}_i)^T \mathbf{Q} (\mathbf{x}_{i,\text{ref}} - \mathbf{x}_i) \\
&+ \sum_{i=k}^{k+N-2} (\mathbf{u}_{i+1} - \mathbf{u}_i)^T \mathbf{R}_d (\mathbf{u}_{i+1} - \mathbf{u}_i) \\
&+ \sum_{i=k}^{k+N-1} \mathbf{u}_i^T \mathbf{R} \mathbf{u}_i
\end{aligned} \tag{3.3}$$

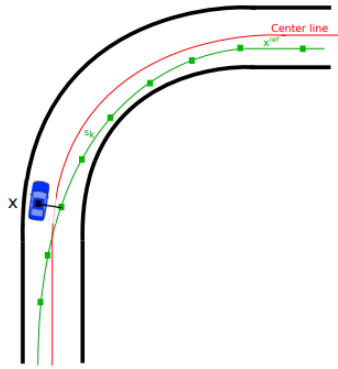
S vazebními podmínkami na stavové veličiny a rozsah vstupních veličin:

$$\begin{aligned}
\mathbf{x}_{k+1}^j &= f(\mathbf{x}_k^j, \mathbf{u}_k^j), \quad \forall k \in [t, \dots, t + N - 1] \\
\mathbf{x}_k^j &\in X_{\text{track}} \\
\mathbf{u}_k^j &\in U(\mathbf{x})
\end{aligned} \tag{3.4}$$

Kdy cílem optimalizační úlohy je nalézt minimální hodnotu kritéria J_k za splnění vazebních podmínek. Tím dostaneme optimální vstupní vektor \mathbf{u} .

- $(\mathbf{x}_{k+N,\text{ref}} - \mathbf{x}_{k+N})$ je rozdíl mezi referenční a aktuální hodnotou na konci predikčního horizontu. \mathbf{Q}_N je obecně pozitivně semidefinitní a ovlivňuje, jakou má tato odchylka váhu.
- $\sum_{i=k}^{k+N-1}$ značí sumu přes predikční horizont od $i = k$ do $i = k + N - 1$, kde $N =$ počet predikovaných kroků
- $(\mathbf{x}_{i,\text{ref}} - \mathbf{x}_i)$ představuje odchylku mezi referenční hodnotou stavových proměnných $\mathbf{x}_{i,\text{ref}}$ a aktuální hodnotou \mathbf{x}_i v každém kroku predikce. \mathbf{Q} je obecně pozitivně semidefinitní váhový faktor pro tyto odchylky.
- $(\mathbf{u}_{i+1} - \mathbf{u}_i)$ představuje odchylku mezi následující a aktuální hodnotou ovládacího vstupu v jednotlivých krocích predikčního horizontu. \mathbf{R}_d je pozitivně definitní váhový faktor pro tyto rozdíly. Tento člen kritéria penalizuje ostré změny vstupních veličin, které by mohli vést k destabilizování auta na vozovce.
- \mathbf{u}_i představuje vektor ovládacích vstupů v daný okamžik i . \mathbf{R} je pozitivně definitní váhový faktor pro tyto hodnoty ovládacích vstupů.

Vazební podmínky jsou shodné s podmínkami definovanými pro problém plánování trajektorie (3.2).



Obrázek 3.3: Ilustrace chyby sledování referenční trajektorie [1]

Prostřednictvím obrázku 3.3 lze nastínit možný stav systému, přičemž lze pozorovat odchýlení automobilu od vygenerované trajektorie. Takto velkých odchylek by se algoritmus za normálních podmínek neměl dopouštět, ovšem nelze podchytit všechny vnější vlivy, které takovou odchylku mohou způsobit.

Představená formulace problému (3.3) a (3.4) charakterizuje optimalizační úlohu MPC, kde je cílem minimalizovat celkový náklad (cost) při dodržování omezení a dosahování referenčních hodnot pro stavové proměnné v daném predikčním horizontu o délce N . Váhové faktory \mathbf{Q}_N , \mathbf{Q} , \mathbf{R} a \mathbf{R}_d stanovují "důležitost" jednotlivých členů (stavu a vstupu) v celkovém cíli optimalizace.

Kapitola 4

Implementace algoritmu řízení - experiment

V této kapitole bude popsána a testována implementace řídicího algoritmu prezentovaného v předchozí kapitole.

V prvním kroku je zapotřebí mít k dispozici část trasy, na které budeme úlohu realizovat. Kdyby se jednalo o reálnou situaci, získávali bychom tuto trasu z čidel a měřících přístrojů, jako je například lidar. Jelikož se budeme pohybovat pouze v simulačním prostředí, je tedy zapotřebí použít buď předem stanovenou trať, anebo generátor, který bude generovat trať vyhovující parametrům vozidla. To znamená, že zaručí průjezdnost tratě.

Jakmile máme obstaranou trasu, můžeme pokročit na problém plánování trajektorie. Trajektorii budeme generovat na základě modelu a s účelem maximalizace postupu podél centrální čáry, toho docílíme vygenerováním M možných trajektorií a jejich následným filtrováním za účelem výběru té optimální.

Další částí bude sledování této vygenerované trajektorie. K řešení tohoto problému použijeme prediktivní řízení na základě matematického modelu. Toto řízení na základě modelu vozidla vygeneruje $N - 1$ akčních zásahů pro $N - 1$ budoucích okamžiků, kdy minimalizací určitého kritéria, zajišťuje optimální řízení pro držení vozidla na referenční trajektorii. Z těchto $N - 1$ vygenerovaných akčních zásahů se vždy použije pouze ten první a tento stejný proces se bude neustále opakovat.

V neposlední řadě je zapotřebí se zamyslet, jak často potřebujeme obnovovat referenční trajektorii, ta je reprezentovaná simulovaným obloukem, který se snaží MPC sledovat. Z důvodů poruch a dalších neměřitelných veličin, které nejsou zahrnuté v modelu, se ale může stát, že referenční trajektorie je například nesplnitelná, nebo pro daný stav systému už není optimální, proto musíme stanovit nějaký kompromis mezi výpočetními nároky a přesností ve formě frekvence generování této referenční trajektorie. Ačkoliv zvýšíme výpočetní nároky, bude použito generování nové trajektorie v každém výpočetním kroku, pro dosažení co nejhladšího průjezdu tratí.

4.1 Parametry modelu

Jak bylo již zmíněno, tato práce se bude soustředit pouze na simulační prostředí, tudíž nejsou definované žádná pevná omezení na model vozidla. Stanovíme si tedy vlastní parametry vozidla: délka = 4.5m, šířka = 2m, rozvor (vzdálenost mezi přední a zadní nápravou) = 2.5m

příčemž vzdálenost od začátku vozidla k přední nápravě je stejná jako vzdálenost od konce vozidla k zadní nápravě = 1m, maximální úhel natočení předních kol = 55°, maximální rychlost 45km/h, maximální zrychlení = 1 m/s², maximální brzdný účinek = 3 m/s².

4.2 Generátor závodních tratí

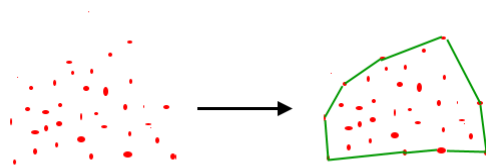
Proces navrhování závodní dráhy začíná generováním náhodných bodů, které položí základy pro následné tvarování a design trati. Tento přístup zaručuje náhodnost a za pomoci geometrických a matematických operací vytváří jak unikátní, tak technicky proveditelný závodní okruh. Algoritmus lze rozložit do několika následujících úloh

Generování Náhodných Bodů

Počáteční fáze procesu generování závodní dráhy zahrnuje vytvoření náhodného souboru bodů v předem definovaných mezích. Toto seskupení tvoří základ pro následující tvar dráhy. Algoritmus zajistí minimální vzdálenost mezi body, vytvářející základní soubor souřadnic, které později definují tvar závodní dráhy.

Výpočet konvexní obálky a úpravy tvaru

Na náhodně generované body je aplikován algoritmus konvexní obálky, který obalí vygenerované body do polygonu a vytvoří tak počáteční tvar závodní dráhy. Pro dosažení rozmanitosti a složitosti jsou následně vygenerovány body na půli vzdálenosti mezi vrcholy konvexní obálky, na tyto body jsou aplikovány náhodné posuny. Tyto posuny jsou ovlivněny faktorem obtížnosti, umožňující přizpůsobení celkové složitosti trati. Dráha je dále upravována úpravou úhlů mezi po sobě jdoucími úseky, aby nebyl překročen maximální úhel zatočení.

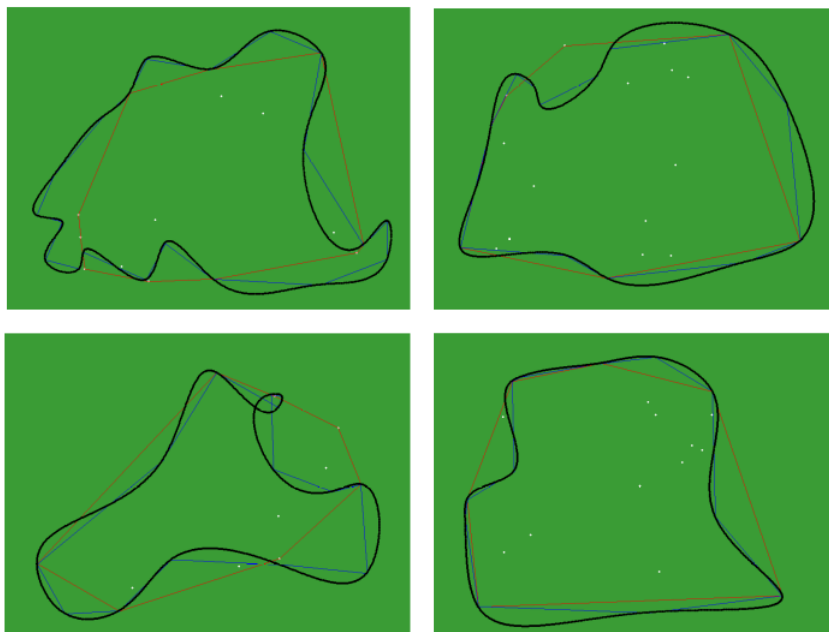


Obrázek 4.1: Vizualizace algoritmu konvexní obálky

Spline Interpolace a vytvoření krajích čar

Na vygenerovaný polygon je použita spline interpolace k vytvoření plynulé a spojitě křivky z upravených bodů polygonu. Tato fáze zajišťuje kontinuitu závodní dráhy. Nakonec jsou na obou stranách středové čáry pomocí knihovny Shapely vytvořeny krajní čáry. Tyto čáry definují šířku dráhy a poskytují jasnou hranici pro závodní vozidla.

Vytvoření trati



Obrázek 4.2: Vizualizace algoritmu vytváření trati

Na obrázcích lze pozorovat předešlé kroky, **bílé body** odpovídají náhodně vygenerovaným bodům, **červený polygon** znázorňuje algoritmus konvexní obálky, **modrý polygon** je upravený polygon zpestřující trať, ale zároveň respektující stanovená omezení, **černá linie** je výsledná středová čára, reprezentující finální tvar trati

4.3 Simulační prostředí (Simulation GUI)

Jelikož je rozsahem této práce pouze simulace realizovaného řízení, byla k algoritmu vytvořena primitivní aplikace

Framework PyQt

Protože jsem pro implementaci simulace algoritmu použil programovací jazyk Python, rozhodl jsem se pro vytvoření GUI (graphical user interface) použít PyQt, jelikož jsem se s Qt samotným seznámil prostřednictvím jiného předmětu. K čemu tento framework slouží je popsáno v následujících odstavcích.

PyQt je mostem, který integruje robustní multiplatformní framework Qt, určený pro realizaci GUI aplikací, napsaný v jazyce C++ s programovacím jazykem Python, zejména sloužící jako modul pro tvorbu grafického uživatelského rozhraní (GUI). Při vývoji aplikací s PyQt je výhodou možnost využívat síly obou technologií - Qt a Pythonu - k vytvoření efektivních a vizuálně atraktivních uživatelských rozhraní.

Jeden z klíčových prvků je práce s widgety, které představují různé části uživatelského rozhraní. Správné rozmístování a design těchto prvků je klíčové pro estetiku a efektivitu aplikace.

Dobrým návykem je oddělování logiky aplikace od grafické vrstvy. Model-View-Controller (MVC) architektura může být výhodná, kde je model odpovědný za data, view za grafické zobrazení a controller za interakci uživatele s aplikací. Toto rozdělení zvyšuje udržitelnost a škálovatelnost aplikace a podporuje jasnou strukturu organizace kódu.

Dalším důležitým pravidlem je efektivní správa událostí. PyQt poskytuje robustní systém signálů a slotů, který umožňuje propojování událostí s odpovídajícími funkcemi nebo metodami. Tímto způsobem lze zajistit plynulou interakci mezi uživatelem a aplikací.

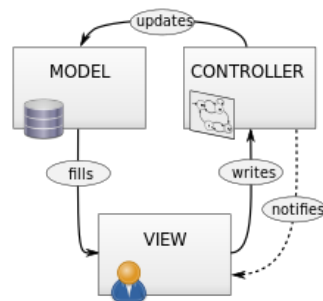
Efektivní správa událostí je dalším klíčovým aspektem. PyQt poskytuje robustní systém signálů a slotů, který usnadňuje propojování událostí s odpovídajícími funkcemi nebo metodami. Tímto způsobem se zajišťuje plynulá interakce mezi uživatelem a aplikací.

4.4 Programové detaily implementace úlohy

Jak bylo již zmíněno, pro programování úlohy jsem použil programovací jazyk python. Pro řešení optimalizační úlohy byla použita knihovna Cvxpy, ta je určena pro řešení konvexních optimalizačních problémů. Poskytuje vysoce úrovně rozhraní pro definování a řešení optimalizačních problémů v čitelné a intuitivní syntaxi, což ji činí přístupnou pro výzkumníky, inženýry a odborníky z různých oblastí.

Dále stojí za zmínku také známá knihovna Numpy a Shapely. Numpy usnadňuje práci s maticemi, přičemž zároveň zrychluje běh programu. Shapely je knihovna, sloužící pro práci s geometrickými objekty, díky které bylo možné snadně vytvořit trasu pro vozidla.

Při programování byla snaha od sebe oddělit funkční část kódu a zobrazení výsledků, této programové struktury se říká **Model-View-Controller (MVC)**. Právě díky tomuto přístupu bylo možné, po naprogramování samotné úlohy, jednoduše dovytvořit uživatelské grafické prostředí pro přehlednou vizualizaci a testování algoritmu. Zmíněný návrhový koncept rozděluje aplikaci na tři hlavní komponenty popsané v následujících odstavcích.



Obrázek 4.3: Grafická ilustrace konceptu MVC

Model reprezentující strukturu dat, logiku a funkce aplikace. Model je zodpovědný za přístup k datům, jejich ukládání a manipulaci. Obsahuje všechny potřebné informace pro aplikaci a pravidla, jak tyto informace zpracovat.

View (Pohled) zodpovídá za zobrazení dat uživateli, tedy za uživatelské rozhraní. Prezentuje data z modelu uživatelům a zobrazuje uživatelské vstupy, které ale nezpracovává přímo, pouze předává zpět do modelu pomocí řadiče.

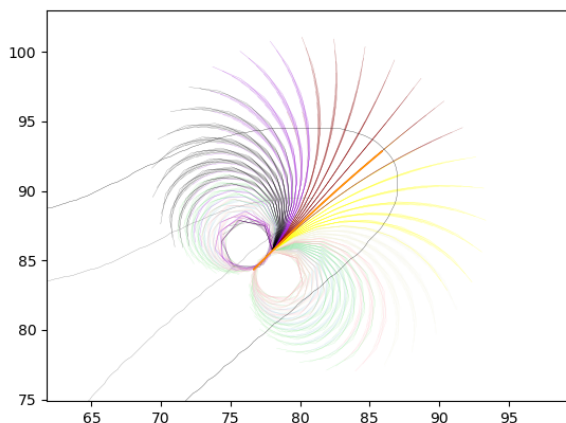
Controller (Řadič) funguje jako prostředník mezi modelem a pohledem. Zpracovává uživatelské vstupy, volá modelovou logiku a aktualizuje pohled s novými daty z modelu. Řídí tak tok dat v aplikaci a zpracovává uživatelské interakce.

4.5 Provedení experimentu

V této části bude předvedeno několik možných naladění algoritmu řízení. Nejprve budeme pozorovat vliv délky horizontu na tvar generovaných trajektorií, bude představen problém související s moc dlouhým predikčním horizontem a zároveň uvedeny nedokonalosti v případě použití krátkého horizontu. Následně bude uveden algoritmus s proměnnou délkou predikčního horizontu a představeny parametry optimalizační úlohy. Všechna řešení budou doplněna obrázky a daty získanými ze simulací.

Pozorování vlivu změny délky predikce referenční trajektorie

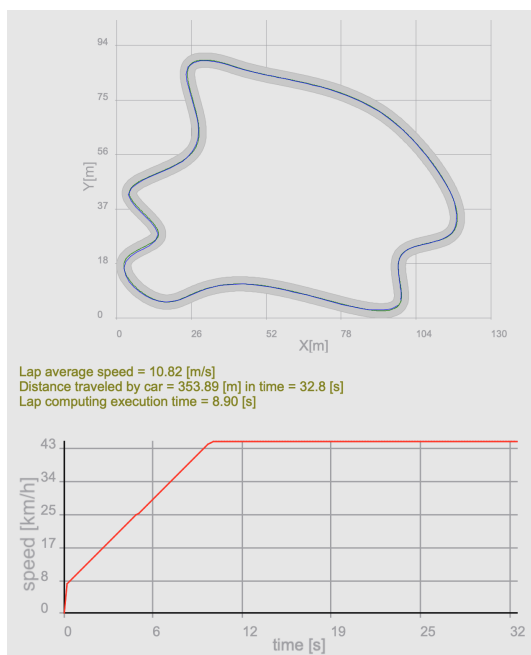
Budeme pozorovat, jak závisí průjezd vozidla dráhou na délce plánovaného oblouku neboli na délce referenční trajektorie. Toto je stěžejní zejména pro zkrácení dráhy při průjezdu zatáčkou. Plánování trasy spočívá v tom, že algoritmus zná středovou čáru a okraje dráhy, na základě těchto informací získává trajektorii ležící mezi okraji dráhy s největším postupem vzhledem ke středové čáře. Pokud bude plánovaný oblouk moc krátký, nebude docházet k odchýlení vozidla od středové čáry, naopak moc dlouhý oblouk může působit potíže v příliš prudkých zatáčkách, kdy by se mohlo stát, že vlivem délky oblouku by všechny trajektorie končily mimo dráhu.



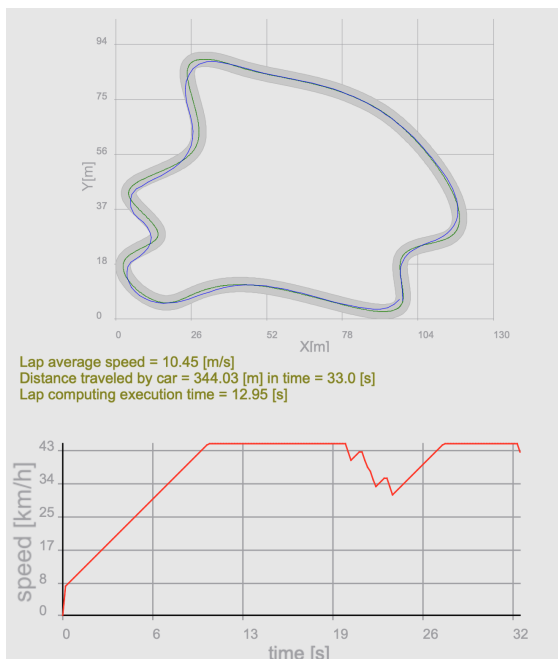
Obrázek 4.4: Grafická ilustrace plánování trajektorie v prudké zatáčce

Na obrázku si můžeme všimnout problému představeného v předchozím odstavci. Je-li délka plánovaného oblouku moc velká, může se stát, že se ocitneme v zobrazené situaci, kdy algoritmus nedokáže zvolit optimální trajektorii.

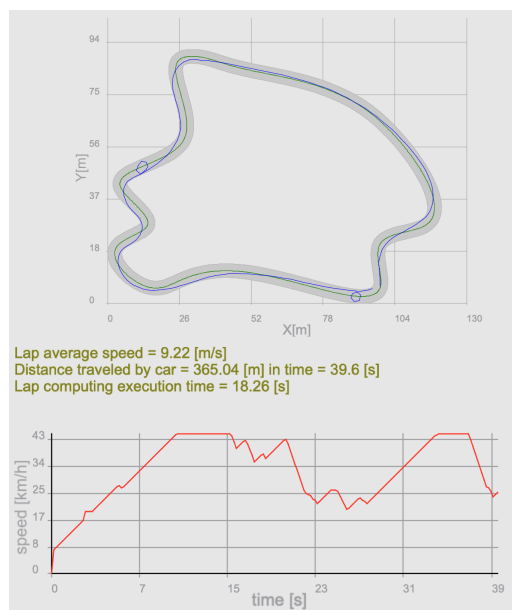
Logicky by se dalo uvažovat, že se tomuto předejde zkrácením plánovaného oblouku, to ovšem přinese negativní vliv v případě dlouhých táhlých zatáček, kdy délka plánovaného oblouku nedovolí odhalit možné zkrácení dráhy, protože generovaný oblouk prokládá vždy moc krátký úsek zatáčky. Problematika bude přiblížena pomocí následujících obrázků



Obrázek 4.5: Délka horizontu predikce zvolená jako 2 následujících kroků



Obrázek 4.6: Délka horizontu predikce zvolená jako 7 následujících kroků



Obrázek 4.7: Délka horizontu predikce zvolená jako 9 následujících kroků

Na obrázcích 4.5, 4.6 a 4.7 vidíme trasu po které vozidlo jezdí, zelenou barvou jsou označené okraje a středová čára, modrá linie představuje vozidlem projetou dráhu. Ve spodní části lze pozorovat vývoj rychlosti vozidla v čase. Obrázky jsou doplněny o text podávající informaci průměrné rychlosti, ujeté vzdálenosti, času jízdy a výpočetní náročnosti ve formě průměrného času běhu programu z pěti simulací.

Z obrázků plyne, že pro tuto trať je rychlejší plánování pomocí kratšího horizontu predikce, navíc je nutno zmínit, že v případě dalšího zvětšování horizontu predikce (více než 7 kroků), začne docházet k nežádoucímu jevu, kdy je jako optimální trajektorie vybrána otočka sloužící ke zpomalení a oddálení od kraje trasy (viz obrázek 4.7), nebo se dokonce začne stávat, že žádná z trajektorií nevede auto po trati a tím přestane řízení úplně fungovat. Přestože čas pro případ prezentovaný na obrázku 4.5 byl jednoznačně lepší, lze pozorovat, že projetá cesta je téměř shodná se středovou čarou a nedochází zde ke zkrácení dráhy v táhlých zatáčkách.

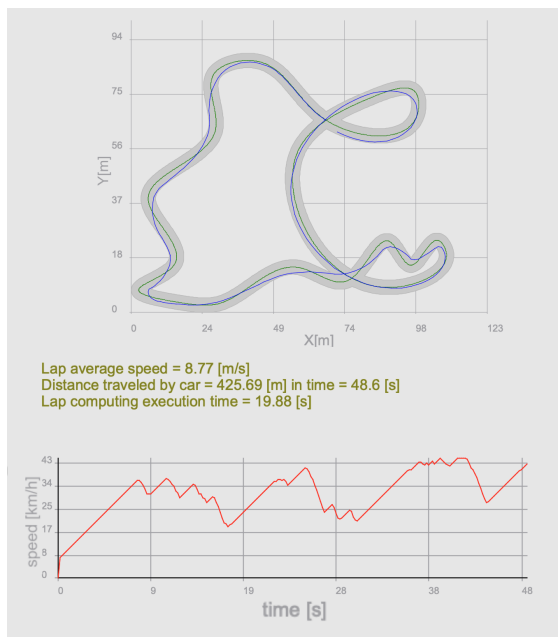
Plánování trajektorie s proměnnou délkou horizontu predikce

Problematika z předchozí sekce byla motivací k vývoji nové techniky plánování. Aby docházelo ke zkrácení trasy v táhlých zatáčkách a zároveň nedocházelo k nerozhodným momentům, kdy nelze vybrat optimální trajektorii, jako je nastíněno na obrázku 4.4, byla implementovaná proměnná délka horizontu predikce odvíjející se od aktuální rychlosti auta.

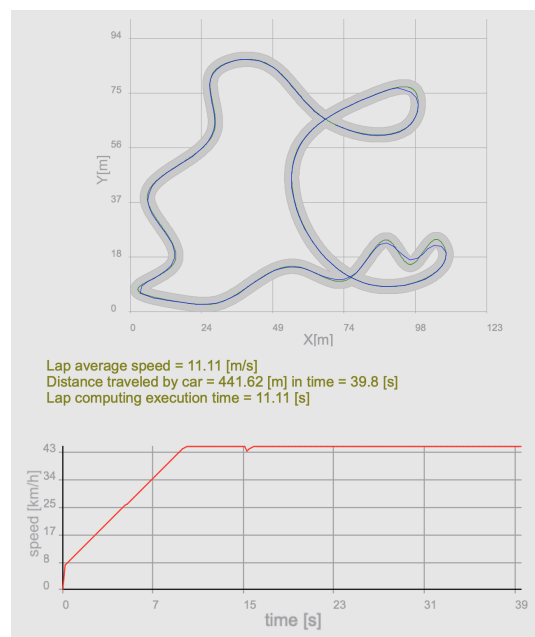
Tento regulátor mění predikční horizont na základě rychlosti auta, zavedeme proto 2 nové proměnné, minimální délku horizontu (HML) odpovídající 5 budoucím krokům a rychlostní koeficient (HLC) rovnající se hodnotě 1.1. Délka horizontu predikce je pak dána vztahem

$$horizon_length = \max(HML, \text{int}(state.v * HLC)) \quad (4.1)$$

Kde $state.v$ odpovídá aktuální rychlosti vozidla v metrech za sekundu, ze vztahu tedy vyplývá, že horizont predikce nebude nikdy kratší než 5 následujících kroků a při vysoké rychlosti se bude protahovat až na 13 kroků, to je omezení vycházející z maximální rychlosti vozidla je 12 m/s. Následující obrázky předvedou porovnání s krátkým fixním horizontem predikce.



Obrázek 4.8: Proměnná délka horizontu predikce



Obrázek 4.9: Délka horizontu predikce zvolená jako 2 následující kroky

Ačkoliv na čas jednoznačně vyhrává řízení s krátkým horizontem predikce, na obrázku 4.8 se ukazuje zajímavá závodní stopa vozidla, která aplikuje zkracování trasy v táhlých zatáčkách,

což lze pozorovat i na celkové ujeté vzdálenosti, ale zároveň zůstává robustní v těch ostrých. Co se týče časů běhu programu, tak vidíme, že řízení s delším horizontem predikce je výpočetně náročnější, tudíž program průměrně běží pro jedno kolo zhruba o 9 sekund déle. Kombinací těchto dvou přístupů by se dalo dojít k optimálnímu nastavení pro danou dráhu. Jelikož jsem měl ale k dispozici generátor takových drah, bylo mým cílem hledat spíše robustní obecné řešení, kdy bude vozidlo schopné projet jakoukoliv trať a zároveň volit závodní stopu.

4.6 Volba parametrů optimalizační úlohy

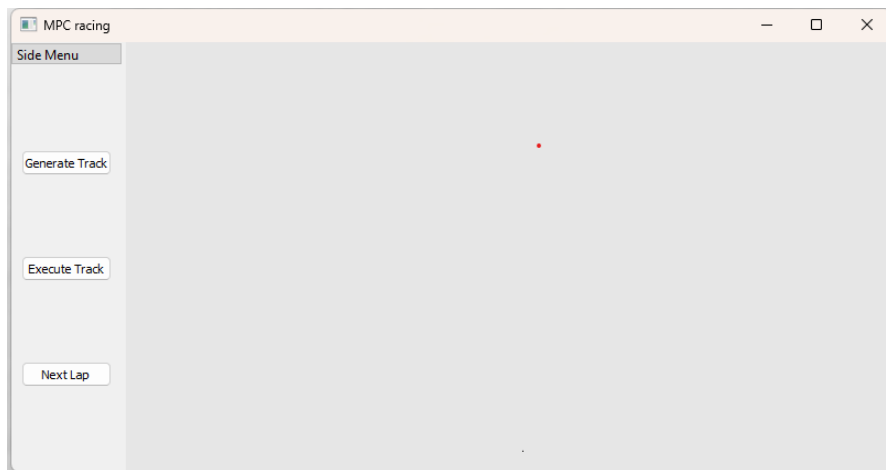
V této sekci bude uvedeno nastavení parametrů optimalizační úlohy. Předem stojí za to zdůraznit, že tato práce uvažuje pouze simulaci, kdy jak simulace, tak plánování a výpočet řízení všichni sdílí ten samý matematický popis vozidla a zároveň do simulace není vnesena žádná náhoda. Pro výpočet optimalizační úlohy bylo potřeba stanovit parametry \mathbf{Q} , \mathbf{Q}_f , \mathbf{R} , \mathbf{R}_D figurující ve výpočtu kritéria optimalizační úlohy (3.3), (3.4).

$$\begin{aligned} \mathbf{R} &= \begin{bmatrix} 0.01 & 0 \\ 0 & 0.01 \end{bmatrix} && \text{(input cost matrix)} \\ \mathbf{R}_d &= \begin{bmatrix} 0.01 & 0 \\ 0 & 1.00 \end{bmatrix} && \text{(input difference cost matrix)} \\ \mathbf{Q} &= \begin{bmatrix} 3.0 & 0 & 0 & 0 \\ 0 & 3.0 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix} && \text{(state cost matrix)} \\ \mathbf{Q}_N &= \mathbf{Q} && \text{(final state cost matrix)} \end{aligned}$$

Diagonální členy matice \mathbf{R} ovlivňují váhu odchylky predikovaných vstupních veličin od těch referenčních, člen R_{11} váží zrychlení vozidla a člen R_{22} váží velikost úhlu natočení předních kol. Prvky matice \mathbf{R}_d váží rozdíl plánovaných vstupů v po sobě jdoucích krocích. Zde jsem zvolil koeficient vážení rozdílu úhlu řízení roven 1 za účelem co nejhladší jízdy. Členy matice \mathbf{Q} poté postupně váží odchylky (predikovaných veličin od referenčních) jednotlivých prvků stavu a to souřadnici X, souřadnici Y, natočení vozidla v prostoru a poslední člen váží rychlost vozidla. Poslední matice \mathbf{Q}_N váží velikost rozdílu predikovaných a referenčních stavových souřadnic v posledním kroku horizontu predikce, pro mou úlohu jsem tuto matici zvolil shodnou s maticí \mathbf{Q} . Všechny hodnoty byly zvoleny na bázi experimentu.

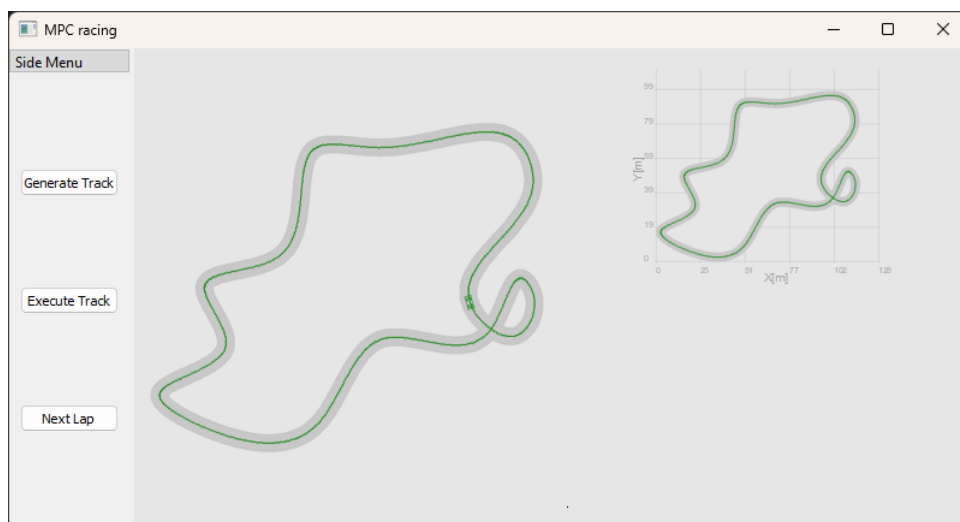
4.7 Představení aplikace

Aplikace aktivně mění velikost jednotlivých widgetů v závislosti na celkové velikosti okna.



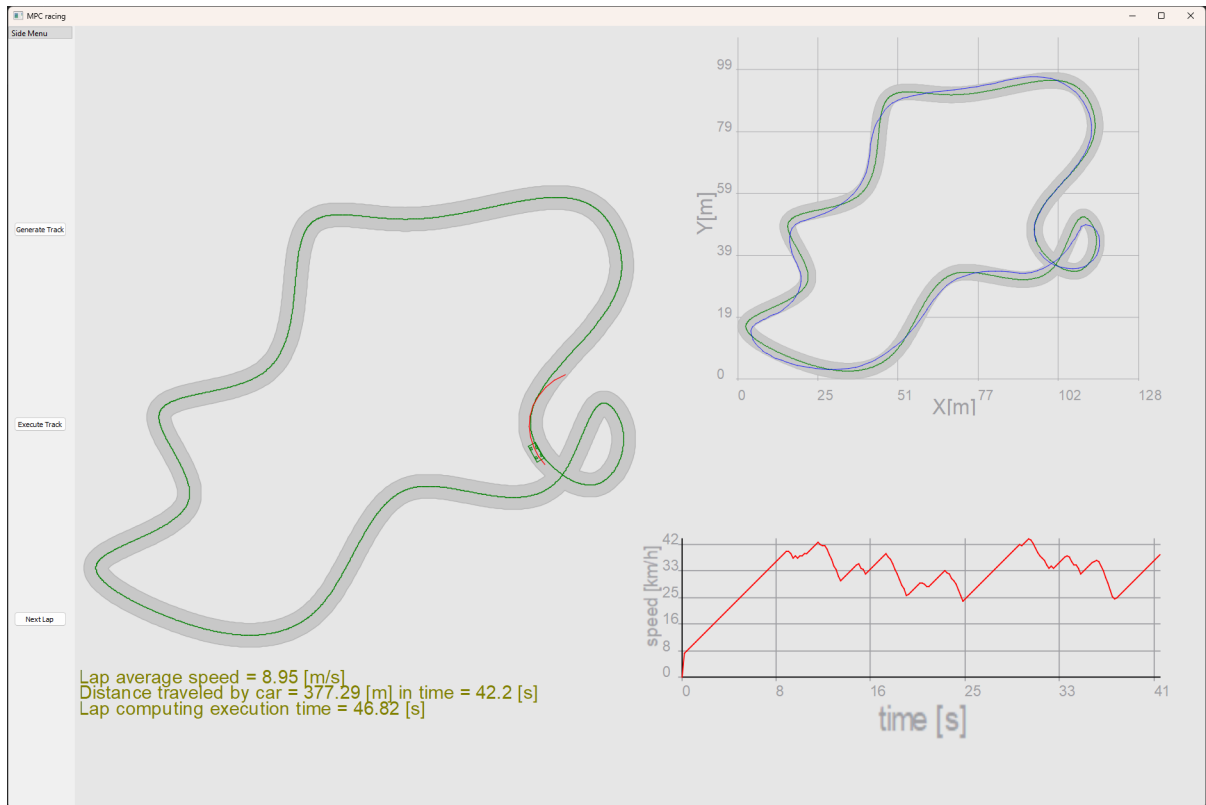
Obrázek 4.10: Defaultní vzhled aplikace po spuštění

Pomocí tlačítka "Generate Track" si uživatel může proklikávat mezi náhodně generovanými tratěmi. Kdy se trať zobrazuje v levé části grafické plochy vpravo od sloupce menu



Obrázek 4.11: Výběr náhodně generované tratě

Jakmile je uživatel se vzhledem a obtížností tratě spokojený má možnost pomocí tlačítka "Execute Track" spustit simulaci průjezdu tratí. V tento okamžik začne na pozadí běžet implementovaný algoritmus řízení. Vše je graficky zobrazeno v aplikaci. Generování referenční trajektorie představuje červený oblouk vycházející z vozidla. Vozidlo je reprezentováno čtvercem s viditelným natočením kol, pro sledování řízení.

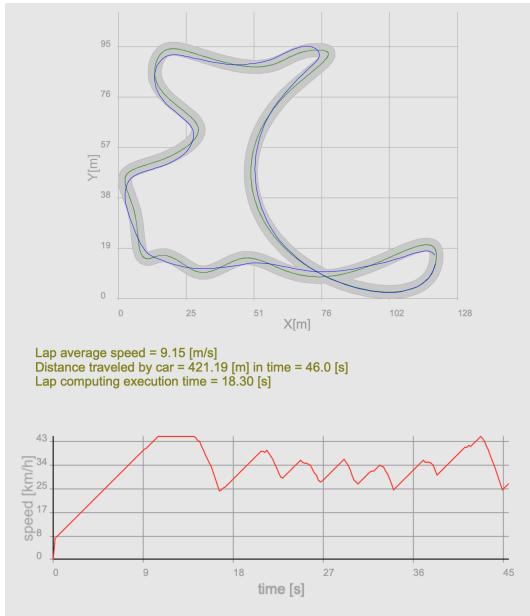


Obrázek 4.12: Průjezd tratě

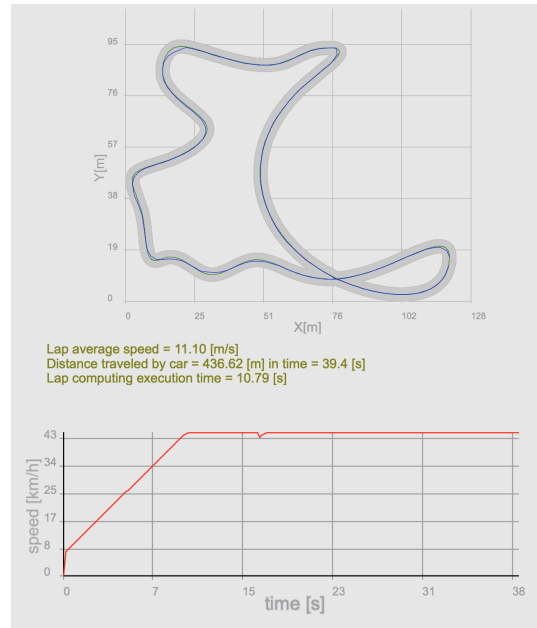
Pravá část aplikace se dělí na horní a dolní segment, kdy v horním segmentu je zobrazena trať a trajektorie vygenerována pohybem vozidla. Na této části můžeme pozorovat algoritmus plánování trati a zpětně hodnotit výsledky průjezdu. V dolní části je vykreslován graf závislosti rychlosti a času, na kterém můžeme pozorovat jak aktuální tak předešlou rychlost vozidla a analyzovat například pokles rychlosti při průjezdu zatáčkou

4.8 Příklady realizace zajímavých tratí

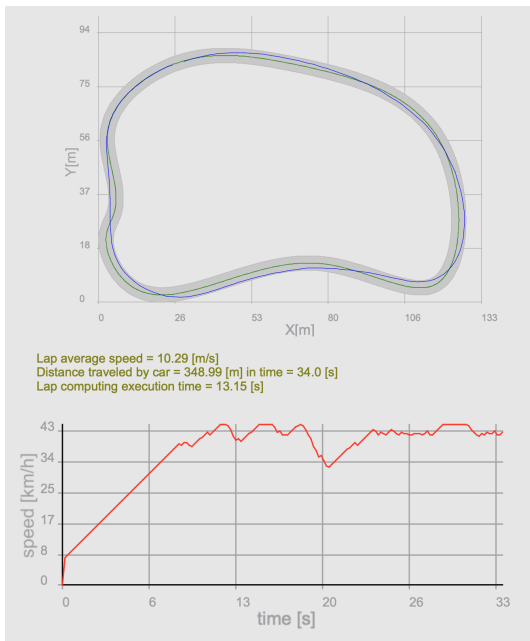
Následující obrázky ukazují provedení experimentu na různých tratích. Vlevo je vždy plánování trasy s proměnnou délkou horizontu predikce, zatímco na pravé části jsou realizace s fixním predikčním horizontem (2 kroky do budoucnosti). Obrázky jsou vždy doplněny textem podávajícím informaci o průměrné rychlosti vozidla, délce ujeté trasy, času jízdy a času exekuce simulace počítačem. Poslední hodnota je průměrem z pěti po sobě jdoucích simulací.



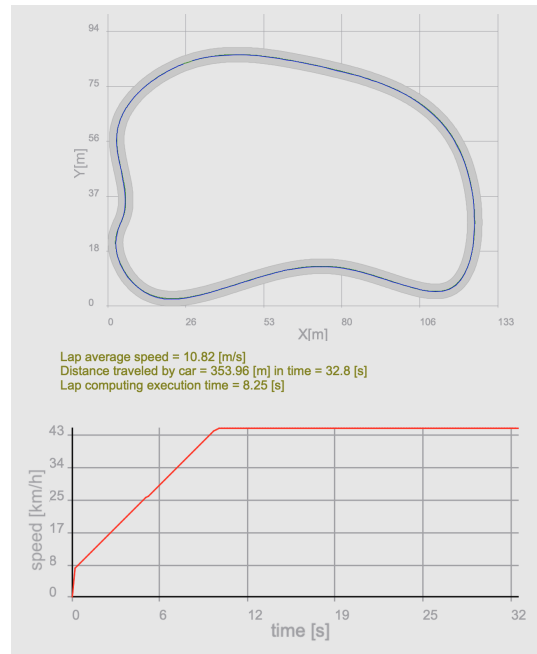
Obrázek 4.13: Proměnná délka horizontu predikce



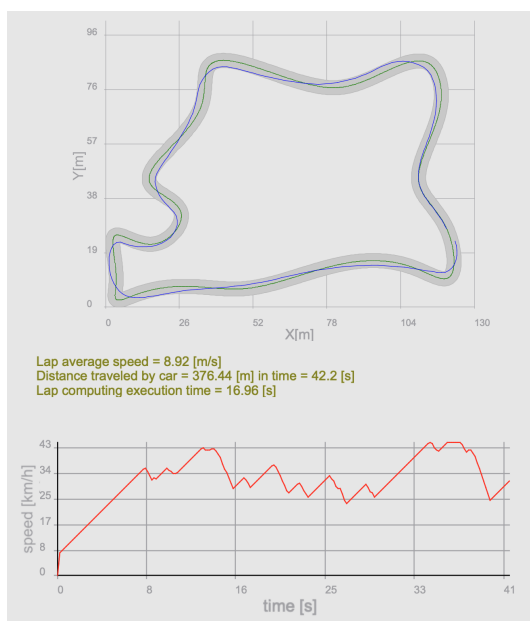
Obrázek 4.14: Délka horizontu predikce zvolená jako 2 následující kroky



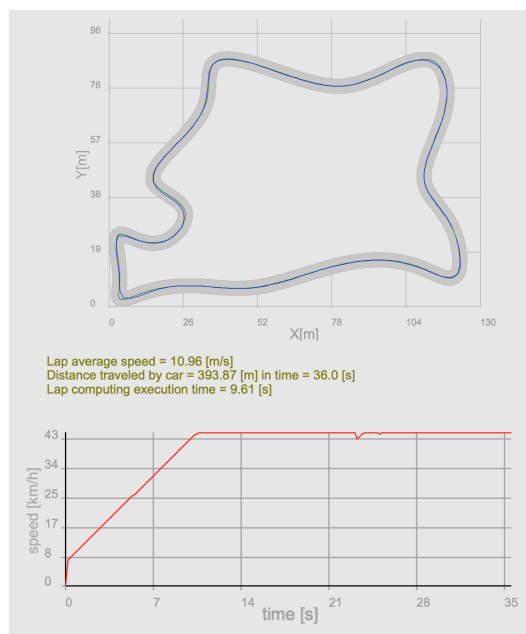
Obrázek 4.15: Proměnná délka horizontu predikce



Obrázek 4.16: Délka horizontu predikce zvolená jako 2 následující kroky



Obrázek 4.17: Proměnná délka horizontu predikce



Obrázek 4.18: Délka horizontu predikce zvolená jako 2 následující kroky

Jak bylo již řečeno v sekci 4.5 a co potvrzují naměřená data jako průměrná rychlost a čas kola, řízení s kratším horizontem predikce dosahuje lepších časů, zatímco plánování s proměnnou délkou predikčního horizontu generuje zajímavou závodní stopu, o tom vypovídá i ujetá vzdálenost. Lze si také všimnout, že v případě krátké reference se autíčko pohybuje téměř stále maximální rychlostí. V případě že bychom tuto rychlost chtěli pro zajímavost zvýšit, se dostáváme do problému pomalého vzorkování, kdy se mezi jednotlivými časy výpočtu referenční trajektorie vozidlo vlivem diskrétního systému ocitá mimo dráhu.

Kapitola 5

Závěr

V této práci byla prozkoumána a implementována strategie prediktivního řízení (MPC) pro autonomní závodní vozidlo v simulovaném prostředí. Celkovým cílem bylo vyvinout a otestovat algoritmus, který bude navigovat vozidlo po různě složitých závodních tratích, přičemž bude provádět minimalizaci času průjezdu a zároveň generovat závodní linii.

Práce nejprve představila teoretické základy MPC, včetně jeho aplikací, výhod a potenciálních výzev. Zabývala se také dvouvrstevným přístupem k MPC, kde se první vrstva soustředí na plánování optimální trajektorie a druhá na její následné sledování. Tento přístup umožňuje efektivní rozdělení úlohy a zjednodušuje složité problémy spojené s navigací vozidla.

K experimentální části byla vytvořena aplikace pro vizualizaci a testování implementovaného algoritmu řízení, zajišťující lepší porozumění chování algoritmu a efektivitě řízení. Součástí aplikace je i generátor náhodných závodních tratí, který umožňuje vytvářet rozmanité a technicky náročné tratě. Tato část práce poté prozkoumala vliv délky horizontu predikce na efektivitu řízení a schopnost vozidla adaptovat se na různé typy zatáček a úseků trati. Byly testovány různé nastavení algoritmu, od krátké až po proměnně dlouhé délky predikčního horizontu a byly zkoumány jejich dopady na kvalitu řízení vozidla.

Výsledky experimentů ukázaly, že kratší horizont predikce, umožňuje vozidlu dosáhnout kratších časů na ve většině tratí, avšak na úkor závodní strategie. Na druhou stranu, plánování s proměnnou délkou predikčního horizontu, která se přizpůsobuje aktuální rychlosti vozidla, přineslo zajímavější závodní stopu, zejména ve složitějších úsecích tratě, avšak za cenu delšího času průjezdu. Naskytuje se zde prostor pro výzkum a to například zkusit projetou trajektorii vygenerovanou plánováním s proměnnou délkou, použít celou jako referenční dráhu pro strategii s krátkým plánováním, tímto způsobem by mohlo dojít k projetí tratě v rekordních časech.

Tato práce poskytuje solidní základ pro další výzkum v oblasti autonomního závodění a otevírá možnosti pro pokročilé strategie řízení, které by mohly být aplikovány v reálných podmínkách a závodních scénářích.

Kapitola 6

Reference

- [1] Alexander Liniger, Alexander Domahidi, and Manfred Morari. Optimization-based autonomous racing of 1:43 scale RC cars. *Optimal Control Applications and Methods*, 36(5):628–647, 2015.
- [2] Tamás Keviczky, Paolo Falcone, Francesco Borrelli, Jahan Asgari, and Davor Hrovat. Predictive control approach to autonomous vehicle steering. 2006 American Control Conference, pages 4670–4675, 2006.
- [3] Jason Kong, Mark Pfeiffer, Georg Schildbach, and Francesco Borrelli. Kinematic and Dynamic Vehicle Models for Autonomous Driving Control Design. pages 2–7, July 2015.
- [4] Robin Verschueren, Stijn De Bruyne, Mario Zanon, Janick V. Frasch, and Moritz Diehl. Towards time-optimal race car driving using nonlinear MPC in real-time. In *Proceedings of the IEEE Conference on Decision and Control*, volume 2015-Febru, pages 2505–2510, 2014.
- [5] Mogens Graf Plessen, Pedro F. Lima, Jonas Martensson, Alberto Bemporad, and Bo Wahlberg. Trajectory Planning Under Vehicle Dimension Constraints Using Sequential Linear Programming. April 2017.
- [6] Emilio Frazzoli, Efstathios Velenis and Panagiotis Tsiotras. On steady-state cornering equilibria for wheeled vehicles with drift. 2009.
- [7] Brian Beckman. *The Physics of Racing*. April 1991. available at: <http://phors.locost7.info/contents.htm>.
- [8] Florent Althé, Philip Polack, and Arnaud de La Fortelle. A Simple Dynamic Model for Aggressive, Near-Limits Trajectory Planning. *IEEE IV 2017 conference*, June 2017.
- [9] Y. Lu, X. Zhang, X. Xu and W. Zhang, "A Two-layer MPC Approach for Consistent Planning and Control of Autonomous Vehicles," 2022 International Conference on Advanced Robotics and Mechatronics (ICARM), Guilin, China, 2022, pp. 717-724, doi: 10.1109/ICARM54641.2022.9959111.
- [10] Gautam, Puneet, Sahai, Shubham, Kelkar, Sachin, Reddy D, Mallikarjuna, Agrawal, Prajwal. (2021). Designing Variable Ackerman Steering Geometry for Formula Student Race Car. *International Journal of Analytical Experimental and Finite Element Analysis (IJAEFEA)*. 8. 1-11. 10.26706/ijaefea.1.8.20210101.

- [11] Bakker, E., Nyborg, L., and Pacejka, H., "Tyre Modelling for Use in Vehicle Dynamics Studies,"SAE Technical Paper 870421, 1987, <https://doi.org/10.4271/870421>.
- [12] Rawat, Pallav. "Model Predictive Control 101."Medium, 2021. Web. accessed 30.04.2024..
<https://pallavrawat.medium.com/model-predictive-control-101-28de17208c39>.