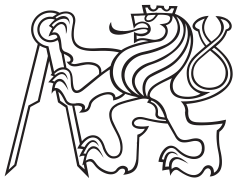


Bakalářská práce



České
vysoké
učení technické
v Praze

F3

Fakulta elektrotechnická
Katedra řídicí techniky

Využití bezdrátové komunikace při distribuovaném řízení kolon autodráhových autíček

Šimon Wernisch

Vedoucí: Ing. Zdeněk Hurák
Obor: Kybernetika a robotika
Studijní program: Systémy a řízení
Květen 2017

České vysoké učení technické v Praze
Fakulta elektrotechnická
katedra řídicí techniky

ZADÁNÍ BAKALÁŘSKÉ PRÁCE

Student: **Wernisch Šimon**

Studijní program: Kybernetika a robotika
Obor: Systémy a řízení

Název tématu: **Využití bezdrátové komunikace při distribuovaném řízení kolon autodráhových autíček**

Pokyny pro vypracování:

1. Navrhněte a implementujte softwarové nástroje pro zaznamenávání parametrů bezdrátové komunikace mezi jednotlivými auty v distribuované řízené koloně aut na autodráze. Zejména nechte je umožněno zaznamenávat zpoždění či přímo i ztráty paketů. Data získaná při experimentech následně analyzujte.
2. Analyzujte (i jen formou simulací) dopad komunikačního zpoždění na účinnost dopředného (angl. feedforward) regulátoru implementovaného na palubě každého auta. Pokuste se rozmyslet, zda by nebylo možné (předpokládané) negativní dopady při znalosti aktuálního zpoždění nějak účinně kompenzovat.

Seznam odborné literatury:

- [1] Liu, Xiangheng, A. Goldsmith, S. S. Mahal, and J. K. Hedrick. Effects of Communication Delay on String Stability in Vehicle Platoons. In ITSC 2001. 2001 IEEE Intelligent Transportation Systems. Proceedings (Cat. No.01TH8585), 625?30, 2001. doi:10.1109/ITSC.2001.948732.
- [2] Willke, T. L., P. Tientrakool, and N. F. Maxemchuk. A Survey of Inter-Vehicle Communication Protocols and Their Applications. IEEE Communications Surveys Tutorials 11, no. 2: 3?20. doi:10.1109/SURV.2009.090202.

Vedoucí: doc. Ing. Zdeněk Hurák, Ph.D.

Platnost zadání: do konce letního semestru 2017/2018

L.S.

prof. Ing. Michael Šebek, DrSc.
vedoucí katedry

prof. Ing. Pavel Ripka, CSc.
Děkan

V Praze dne 9. 2. 2017

Poděkování

V první řadě bych rád poděkoval svému vedoucímu panu inženýrovi Zdeňku Hurákovi za řádné vedení, Martinu Ládovi za četné rady a nápady a panu inženýrovi Ivu Hermanovi za uvedení do problematiky a vedení v době, kdy jsem k projektu přišel. Děkuji své široké rodině za psychickou, finanční a kulinářskou podporu ve všech chvílích.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

V Praze, 25. května 2017

Abstrakt

Práce se v rámci projektu Slotcar platooning, vedeného na katedře řídicí techniky, zabývá pochopením měřítek kvality a nedokonalosti bezdrátové komunikace, kterými jsou zpoždění a ztráta paketů. Navrhuje postup k jejich měření a provádí analýzu v zájmu určit pravděpodobné meze měřítek a v zájmu vyvodit směřodonné údaje pro návrh aplikace, ve které bude síť využita k číslicovému přenosu.

Dále práce popisuje návrh CACC systému pro řízení podélné vzdálenosti mezi vozidly k použití v projektu. Na navržený systém pohlíží z hlediska stability systému při uvažování nedokonalostí bezdrátové komunikace, která je při řízení využita, a navrhuje, jakým způsobem se dají problémy vzniklé nedokonalostmi kompenzovat.

Klíčová slova: komunikační zpoždění, řetězcová stabilita, VANET, CACC

Vedoucí: Ing. Zdeněk Hurák
katedra řídicí techniky,
FEL

Abstract

As part of the Slotcar Platooning project led at the Department of Control Engineering, this thesis examines network induced performance liabilities and imperfections, delay and packet loss. It proposes metrics and procedures for measuring them in order to determine the probable limits and in order to deduce indicators for the design of the application in which the network will be used for digital transmission.

Then the thesis continues by describing the design of a CACC system for longitudinal vehicle control usable in the project. The designed system is examined in terms of stability when considering imperfections in the network which is used for velocity control data transmission. Suggestions follow, how to compensate the noticed network induced problems.

Keywords: communication delay, string stability, VANET, CACC

Title translation: Wireless networking for distributed control of platoons of slotcars

Obsah

1 Úvod	1	3.1.1 Volba regulátorů	17
1.1 Projekt Slotcar platooning	1	3.1.2 Řetězcová stabilita	19
1.2 Cíl práce	2	3.2 Dopad zpoždění	20
2 Nástroje pro zaznamenávání parametrů a metrik bezdrátové komunikace	3	3.2.1 Kompenzace zpoždění	21
2.1 Komunikační zpoždění	4	4 Závěr	25
2.1.1 Implementace měřicí metriky zpoždění	6	A Algoritmy	27
2.1.2 Rozdělení do tříd a analýza	7	B Literatura	29
2.1.3 Vliv více zdrojů	9		
2.2 Ztráta paketů	10		
2.3 Uchovávání a synchronizace času	12		
2.3.1 Experimentální ověření synchronizace hodin	12		
2.4 Rozšíření GUI	13		
3 Využití komunikace při řízení	15		
3.1 CACC systém pro autíčka Slotcar platooning	15		

Obrázky

1.1 Kolona autíček, autor Jiří Zemánek	2	3.2 CACC řídicí smyčka k -tého vozidla v koloně	16
2.1 Topologie sítě použité v projektu Slotcar platooning	3	3.3 Porovnání skokové odezvy ACC přenosu autíčka za použití PD regulátoru a bez	17
2.2 Časový průběh měřených zpoždění s vedlejším provozem	5	3.4 Porovnání zesílení $F(j\omega)$ pro různé hodnoty h_d	18
2.3 Hlavička dat aplikační vrstvy autíček	7	3.5 Časový průběh rychlostí CACC kolony sledující referenční vozidlo	18
2.4 Roztřídění testovaných vektorů do tříd dle empirických rozdělení	8	3.6 Schéma přenosu reference skrz kolonu	19
2.5 Mapa pravděpodobnosti $P(\tau \leq \lambda)$	9	3.7 Zesílení přenosu $\Gamma_k(j\omega)$ s $h_d = 0,05$ pro různá zpoždění	20
2.6 Porovnání rozdělení pro konstantní celkový datový tok s n vysílajícími autíčky	9	3.8 Nalezený minimální rozestup $h_{d,min}$ v závislosti na komunikačním zpoždění τ	21
2.7 Ilustrační číslicový přenos sítě	11	3.9 Simulace predikce rychlosti za zpoždění $\tau = 200$ ms, vzorkování v síti $T = 30$ ms	22
2.8 Princip porovnávání PPS pulzů dvou hodin s probíhající synchronizací	12	3.10 Degradovaná CACC řídicí smyčka	23
2.9 Porovnávání synchronizace dvou hodin na osciloskopu	13		
2.10 Nový panel GUI aplikace	14		
3.1 Schéma řízení kolony	15		

Tabulky

2.1 Parametry sítě ovlivňující zpoždění 6

2.2 Deskriptivní statistika tříd 8

Kapitola 1

Úvod

Automobilová doprava je neodmyslitelnou součástí společnosti a je s ní spojeno několik problémů. Americká studie Schrank et al. (2015) prokazuje, že neefektivní cestování a čekání v dopravních zácpách vede k spotřebě o 3,1 bilionů galonů více paliva a stojí americký národ 160 bilionů dolarů ročně. To jsou čísla za rok 2014, která nadále rostou. V posledních dvaceti letech byl veden vývoj v polích inteligentních dopravních systémů (ITS) a automatizovaných dálničních systémů (AHS) na systému Adaptive cruise control (ACC), kde řízení rychlosti vozidla je upravováno radarovými měřeními vzdálenosti k předchozímu vozidlu. Tento asistenční systém začal pronikat na trh a nabízí automatické udržování vzdálenosti k předchozímu vozidlu. Řada studií ukázala, že použití automatizovaných systémů v automobilech a formování kolon sníží emise a spotřebu paliva až o třetinu (Park et al., 2011), zvýší bezpečnost jízdy a efektivitu využití silniční infrastruktury. Přidáním komunikace k řízení, kdy auta si budou průběžně sdělovat důležitá data, je možné dosáhnout ještě lepších výsledků, například auta řídí bezpečně i při menším rozestupu s ostatními vozidly. Toto rozšíření systému ACC se nazývá Cooperative adaptive cruise control (CACC). Vinou nedokonalostí síťové komunikace, jako je komunikační zpoždění, dlouhé periody vzorkování a ztráta paketů, však může CACC své kvality ztratit a provoz ohrozit.

1.1 Projekt Slotcar platooning

Cílem projektu je vytvořit platformu pro zkoumání a demonstraci řízení kolon, a to takovou, která bude cenově dostupná a rozšiřitelná. Pro tento záměr



Obrázek 1.1: Kolona autíček, autor Jiří Zemánek

je experimentální model sestaven z dráhových autíček Carrera, která jsou vybavena řadou senzorů, STM procesorem, H-můstkem a počítačovým modulem Raspberry Pi Compute Module. Díky tomuto vybavení je každé autíčko schopno regulovat rychlost na základě měření vzdálenosti k předchozímu autíčku, měření vlastní rychlosti a zrychlení, a je schopné také zpracovávat informace obdržené od ostatních autíček po bezdrátové síti (Lád et al., 2017). Celý projekt zaštiťuje skupina AA4CC katedry řídicí techniky.

V předchozích letech byla na projektu propracována identifikace systému jednotlivých autíček a návrh řídicích zpětnovazebních regulátorů na základě lokálních měření. Dále byl vytvořen program v jazyce Java pro nastavování parametrů, jako jsou udržovaný rozestup, referenční rychlost apod., vše v grafickém uživatelském rozhraní. Tento program také m. j. umožňuje rozesílat měření aut po síti mezi auty a do programu MATLAB.

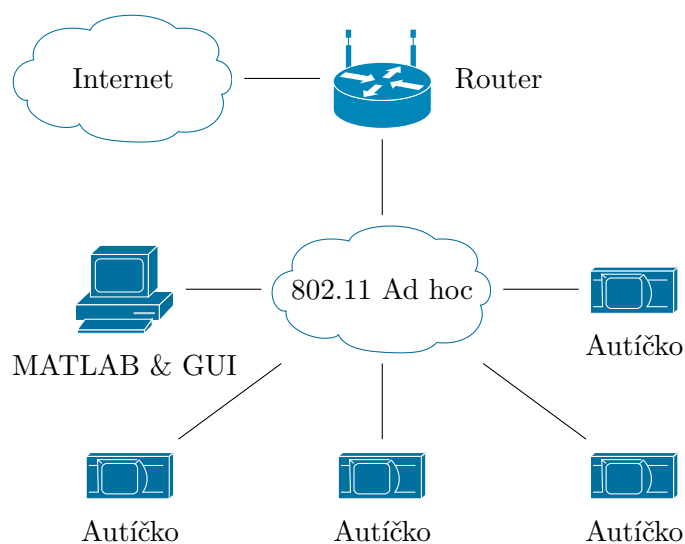
1.2 Cíl práce

Tato práce si v první fázi klade za cíl vytvořit nástroje pro zaznamenávání zpoždění, které vzniká při komunikaci po bezdrátové síti. Zaměření bude na aplikaci, ve které probíhá řízení, a nikoliv na nižších vrstvách OSI modelu či obecných modelech. Vytvořené nástroje budou navrženy pro reprodukovatelnost ve chvíli, kdy se v aplikaci změní podložené komunikační vrstvy či cílová rychlost komunikace. Změřená data budou dále statisticky analyzována pro současnou síť použitou v projektu. V druhé fázi se práce zaměří na syntézu dopředného filtru pro CACC řízení, analýzu dopadu komunikačního zpoždění na řízení autíčka právě za použití CACC, a v poslední řadě se pokusí navrhnout účinnou kompenzaci vlivu zpoždění.

Kapitola 2

Nástroje pro zaznamenávání parametrů a metrik bezdrátové komunikace

Jak bylo nastíněno v úvodu, síť zanáší do distribuovaných řízených systémů řadu nesnází. Je možné, že tyto problémy bude možno heuristicky či numericky řešit, a pro tento záměr je nutné implementovat nástroje, které za běhu programu vyhodnotí některé metriky a následně s nimi pracují, či je pouze zaznamenají pro dodatečné statistické zpracování a posouzení.



Obrázek 2.1: Topologie sítě použité v projektu Slotcar platooning

2.1 Komunikační zpoždění

Komunikační zpoždění je důležitou vlastností datových sítí a je jedním z měřítek kvality propojení dvou síťových uzlů. Před měřením je zapotřebí definovat, co pro konkrétní aplikaci zpoždění znamená. Laner (2013) pokládá při úvahách k zavedení měřících metrik tři otázky

1. Ve které síťové vrstvě se zpoždění projevuje?
2. Které síťové prvky by měly vytvářet události?
3. Co by mělo být definováno jako událost pro výpočet zpoždění?

Událostí je zde myšlen moment, ve kterém průběh aplikace projde významným bodem a při kterém je zaznamenán čas. Je důležité poukázat na fakt, že ve zkoumané aplikaci jsou síťové uzly v jedné ad hoc síti, viz obrázek 2.1, v důsledku čehož je zkoumaný model zjednodušený oproti ostatním síťovým studiím. Mezi autíčky, jejichž komunikaci zaznamenáváme, není žádný mezuzel a data putují jedním skokem. Je tedy triviální určit jako zdroje událostí právě dvě komunikující autíčka. Pro řízení jsou významná data, která jedno autíčko měří a odesílá a jiné autíčko přijímá. Tato data se nachází v nejvyšších vrstvách OSI modelu, které jsou realizovány uvnitř aplikace, a zde by také měly být zaznamenány události. Metrika pro zpoždění bude intuitivně definována jako

$$\tau_{XY,n} = t_{Y,n} - t_{X,n} \quad (2.1)$$

kde $\tau_{XY,n}$ je komunikační zpoždění n -té datové jednotky (paketu) mezi zdrojem paketu X a příjemcem paketu Y , $t_{Y,n}$ je čas, kdy je celá datová jednotka zrekonstruována v příjemci, a $t_{X,n}$ je čas, ve kterém byla datová jednotka vytvořena k odeslání ve zdroji.

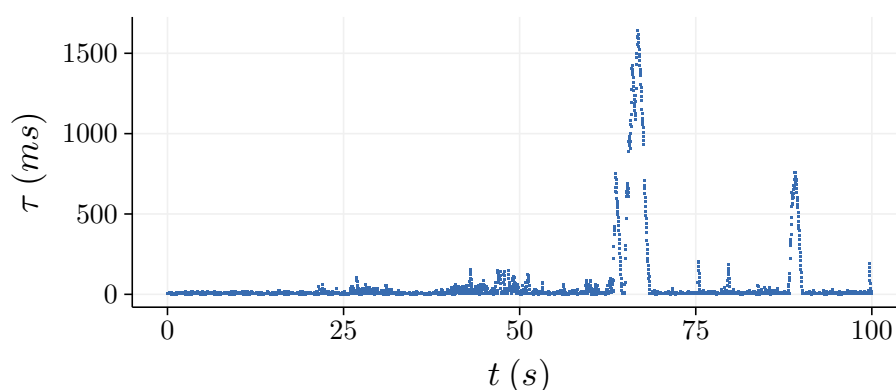
Zpoždění pak pro lepší pochopení můžeme rozepsat na následující díly

$$\tau = \tau_q + \tau_m + \tau_t + \tau_p$$

kde τ_q je doba potřebná k odeslání všech předchozích nedokončených paketů ve frontě zdroje, τ_m je doba strávená čekáním na uvolnění sdíleného komunikačního média, τ_t je doba, kterou trvá vyslání všech bitů, a τ_p je doba šíření jednoho bitu prostorem z vysílače k přijímači. Některé lze dále rozepsat. Pro dobu šíření je dán jasný vzorec $\tau_p = \frac{d}{c}$, kde d je prostorová vzdálenost mezi X a Y a c je rychlost šíření vlny vzduchem, která je rovna rychlosti světla. Vzdálenost, kterou vlnění nejdále dosáhne, se bude lišit podle vyzařovacího výkonu a směrovosti antény, ale u autíček můžeme očekávat nejvyšší vzdálenost

kolem patnácti metrů. Díky těmto číslům můžeme τ_p bezpečně zanedbávat. Podobně pro dobu vyslání všech bitů existuje vztah $\tau_t = \frac{n}{R}$, kde n je počet bitů a R je přenosová rychlost sítě. Tato složka zpoždění je neměnná a není třeba ji zkoumat.

Pro τ_q a τ_m není pragmatický vztah, který by za znalosti parametrů sítě určil hodnotu. Doba čekání ve frontě bude nulová, když vytížení sítě bude $U \ll 1$ a fronta zůstane bezpečně prázdná. Neurčitě poroste ve chvíli, kdy vytížení sítě bude vysoké a zprávy nebudou odcházet dostatečně rychle. V případě, kdy vysílač má volný kanál, je i čekání na uvolnění τ_m nulové. U bezdrátových sítí, kde je frekvenční pásmo využito v četných zařízeních, takovýto předpoklad nelze uvažovat. Navíc přístupová metoda CSMA/CA, využitá v sítích WiFi, se z principu chová náhodně a k uzlům, které se pokouší vysílat, nevykazuje spravedlivé rozdělení kapacity (Jian – Chen, 2008). Z tohoto důvodu byly zkoumány náhrady této přístupové metody například v Jian – Chen (2008); Khairnar – Kotecha (2013).



Obrázek 2.2: Časový průběh měřených zpoždění s vedlejším provozem

Ukázka vytížení sítě cizími zdroji je v grafu 2.2, ve kterém je znázorněn časový vývoj měřeného zpoždění při záměrně spuštěné vytěžovací aplikaci, která náhodně obsadila kanál. V době nečinnosti se zpoždění drží v malém intervalu, který považují, co se týče velikosti zpoždění, za dobrý. Vedle toho jsou špičky, kdy zpoždění roste k hodnotám výrazně vyšším. Každé zvýšení oproti předchozímu vzorku ukazuje čekání právě na uvolnění kanálu a vnímaná akumulativita ukazuje plnění front. Příčina těchto dvou jevů, vysoké vytížení sítě jedním, nebo více zdroji, je v anglické literatuře známá jako congestion (zahlčení, přetížení). Zahlčení je hlavní úskalí, se kterým se MAC vrstva bezdrátových sítí potýká, a řada studií se jím zabývá, například Dabrowski – Mills (2016); Jardosh et al. (2005), ale také i v obecném kontextu teorie grafů a analytické matematiky (Pizano, 2010).

Θ	Parametry experimentu
	velikost paketu η časový rozestup paketů λ datový tok r
Ψ	Rušivě proměnné parametry
	Šum bezdrátového kanálu změny ve vytížení kanálu neznámými zdroji vzdálenost uzlů d nejistoty měření
Φ	Konstantní parametry
	Kapacita kanálu C_{max}

Tabulka 2.1: Parametry sítě ovlivňující zpoždění

Tabulka 2.1 udává očekávané parametry pro modelovanou situaci. Ty jsou výběrem z Laner (2013, tabulka 2.3) s lehkými modifikacemi. Θ tvoří prostor, ve kterém je testováno. Měření jsou prováděna pro obecně proměnné dvojice (η_n, λ_n) , které dohromady dávají okamžitý datový tok $r_n = \frac{\eta_n}{\lambda_n}$ a průměrný datový tok $R = \frac{\bar{\eta}}{\bar{\lambda}}$ (Laner, 2013). Ψ jsou parametry, jejichž šum se snažíme odrušit správným provedením experimentu. Jediný konstantní parametr C_{max} lze změřit nástroji, jako je Iperf (Iperf), a to za dobrých podmínek. Pro autička je to $C_{max} = 11,2$ Mbit/s. Vytížení definujeme pomocí kapacity jako

$$U = \frac{r}{C_{max}} \quad (2.2)$$

2.1.1 Implementace měřicí metricky zpoždění

V minulé sekci bylo definováno, co je cílem měření, při kterých událostech mají být časové značky zaznamenány, a jakým způsobem je síť parametrizována. Výpočet zpoždění je žádoucí provést v příjemci datových jednotek. Ten je schopen určit čas přijetí, ale potřebuje znát i čas vzniku. Tuto informaci je tedy potřeba přenést spolu se samotnými daty. Obrázek 2.3 zobrazuje hlavičku dat vkládanou do UDP datagramu rozšířenou o časovou značku. Časová značka zde představuje celé číslo o velikosti čtyř bajtů, kde horní limit je dán jako $2^{31} - 1 = 2\,147\,483\,647$. Cílové rozlišení času a zároveň jednotka časové značky je jedna milisekunda. To znamená, že časová značka přeteče zhruba každých 25 dní. Pro jednodušší rekonstrukci a člověku jasnou představu, kdy k přetečení dojde, je možné implementovat časovou značku ne jako počet milisekund od začátku systémové epochy, která je pro Linux dána jako 1. ledna 1970, ale jako počet milisekund od vlastní klouzavé epochy, na

kteří se stroje se správným časem (rozebráno v sekci 2.3) shodnou. Takovou epochou může být například „půlnoc z minulé neděle na pondělí“. Z této časové značky není problém rekonstruovat čas a datum a vypočítat metriku pro zpoždění z rovnice 2.1. Ke skutečnému přetečení nedojde, protože dřív, než uplyne 25 dní, se epocha posune a časová značka „ přeteče“ uměle.

63	55	47	39	31	23	15	7	0
Časová značka				Číslo služby	Délka dávky	Cílová adresa	Odchozí adresa	

Obrázek 2.3: Hlavička dat aplikační vrstvy autíček

■ Sběr dat

Test spočívá ve vytvoření toku paketů, který bude parametrizován prostorem Θ , a zaznamenávání zpoždění. Pro jednoduchost bude uvažován jenom jeden typ toků, a to ten, ve kterém jsou parametry z Θ po dobu měření konstantní. K tomu bude zdroj paketů vytvářet a vysílat prázdné pakety o požadované velikosti a s požadovanou periodou a příjemce paketů bude nastaven k ukládání požadovaného množství vzorků. Těmto vzorkům lze přiřadit empirické rozdělení a to poté lze podrobit statistické analýze. V prostoru Θ je definována oblast zájmu pomocí intervalů parametrů z Θ . V těchto intervalech je vybráno několik rozprostřených hodnot, pro které se test provede.

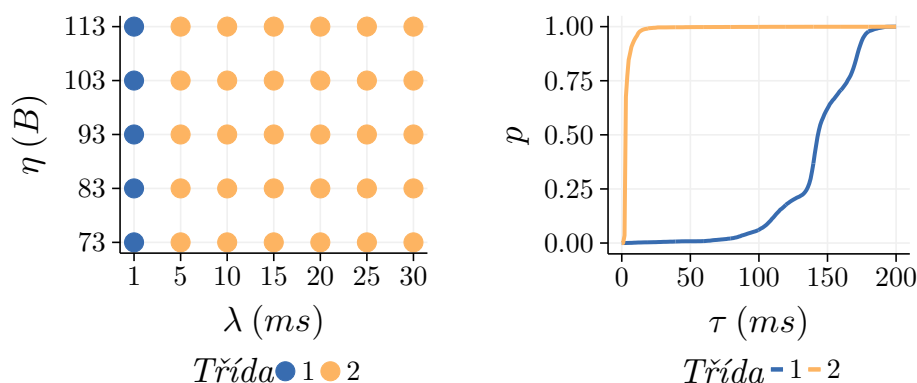
■ 2.1.2 Rozdělení do tříd a analýza

Již diskretizovaný prostor Θ obsahuje větší množství bodů a k nim změřených empirických rozdělení. V řadě z bodů se rozdělení neliší a algoritmy K-medoids a Partition Around Medoids (Maechler et al., 2017) lze sdružit body do tříd. K tomu bude definována vzdálenost empirických rozložení náhodných veličin (Laner, 2013)

$$D(F, G) = \sqrt{\int_0^\infty \frac{1}{\tau} (F(\tau) - G(\tau))^2 d\tau}$$

kde F a G jsou empirická rozložení zpoždění τ .

Grafy 2.4 jsou výsledkem předchozích kroků, ve kterých jsem vybral oblast zájmu na základě množství dat a periody, které byly v té době v aplikaci používány. Prostor zde byl rozdělen do tříd, viz 2.4a, a pro středy těchto



(a) : Mapování vektorů do tříd (b) : Empirické rozdělení středů tříd

Obrázek 2.4: Roztřídění testovaných vektorů do tříd dle empirických rozdělení

Třída	μ_τ	σ_τ	$q_\tau(0,5)$	$q_\tau(0,75)$	$q_\tau(0,90)$	$q_\tau(0,95)$
1	142,71	26,45	143	164	173	176
2	3,56	2,07	3	3	5	8

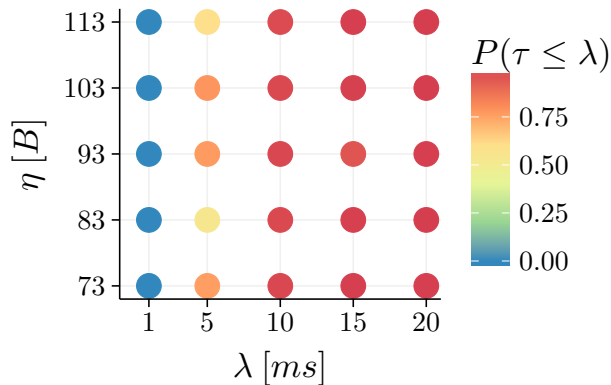
Tabulka 2.2: Deskriptivní statistika tříd

tříd byla vykreslena empirická rozdělení v grafu 2.4b. V těchto grafech lze pro zvolené (η, λ) najít očekávané rozložení zpoždění. Tabulka 2.2 udává pro jednotlivé třídy výsledky deskriptivní statistiky, jako jsou střední hodnota, směrodatná odchylka, medián a některé další kvantily.

Zájmová oblast pro oba parametry je taková, že λ pokrývá širší oblast a η je značně omezené, s minimální hodnotou danou velikostí UDP datagramu a maximální hodnotou, která se nepřibližuje limitu. Třídy jsou pouze dvě, protože jediný rozdíl v empirických rozloženích, který je významnější než chyby způsobené nekontrolovanou změnou parametrů Ψ , je u měření, kde autíčko nezvládlo odesílat s požadovaným λ . Návrh komunikace pak nehraje roli co do rozložení pravděpodobnosti zpoždění, trefí-li se do druhé třídy.

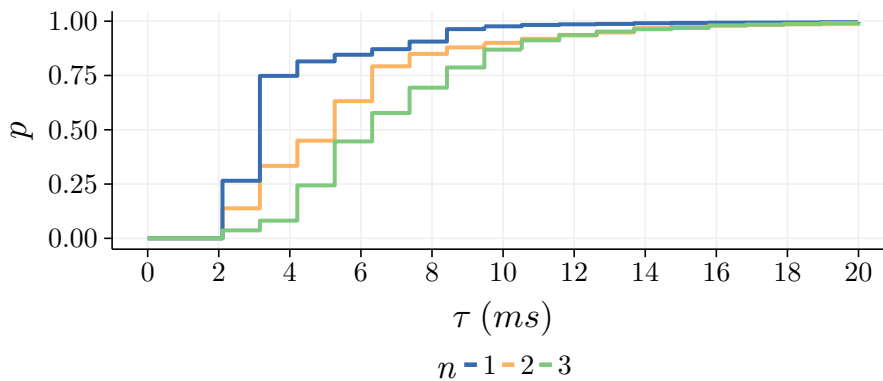
■ Doručení dat během jedné periody

Jedním z možných požadavků je, že při vzorkování dat s periodou $T = \lambda$ v obou zařízeních má datová jednotka změřená ve vzorkovaném čase k být odeslaná a připravená k zpracování v čase $k+1$. To by znamenalo, že zpracovaná informace

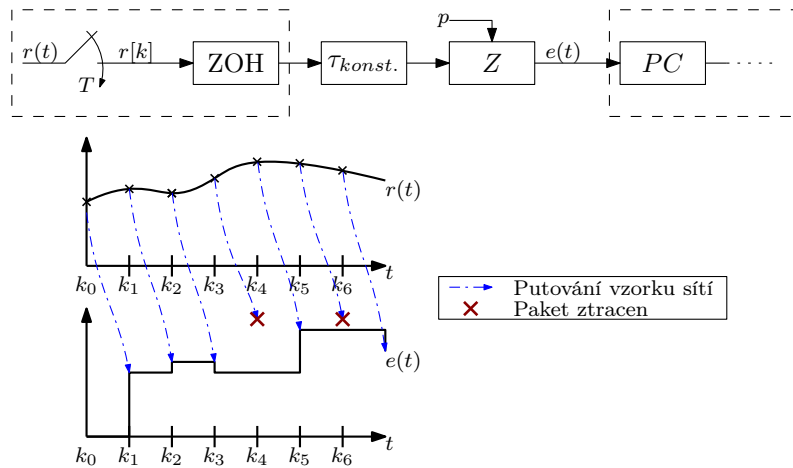
Obrázek 2.5: Mapa pravděpodobnosti $P(\tau \leq \lambda)$

bude nejméně stará. Pro tento požadavek lze z měření určit pravděpodobnost, se kterou zpoždění datové jednotky nebude větší než $T = \lambda$. Ta je barevně zobrazena v grafu 2.5. Z grafu je vidět, že při vhodném výběru T během návrhu není nereálné předpokládat, že paket vždy bude k dispozici v čase $k + 1$. Samozřejmě, nižší vzorkovací perioda s sebou nese řadu výhod, ale uvážíme-li rostoucí zatížení komunikace, je potřeba nalézt periodu uprostřed.

2.1.3 Vliv více zdrojů

Obrázek 2.6: Porovnání rozdělení pro konstantní celkový datový tok s n vysílajícími autičky

V předchozích podsekcích byla rozebrána zpoždění při uvažování jednoho vysílajícího zdroje dat, a to bylo pro analýzu zpoždění vhodnější. Jak však



Obrázek 2.7: Ilustrační číslicový přenos sítě

by přinesla dodatečné režijní náklady a k paketu by bylo nutné přidat další položku o velikosti několika bajtů. Měření z sekce 2.1.2 ukazuje, že přidání dalších dat nemá na přenos vliv. Přeposílání dat, kterým TCP zaručuje přenos paketu, je dokonce nežádoucí, a to z toho důvodu, že ve chvíli, kdy probíhá přeposílání již staré informace, je k dispozici informace nová, která trpí stárnutím ve frontě. Je pak žádoucí ponechat datový přenos na UDP, ignorovat ztracené pakety a pro ostatní pakety přejít na TCP spojení, je-li zapotřebí spolehlivost přenosu.

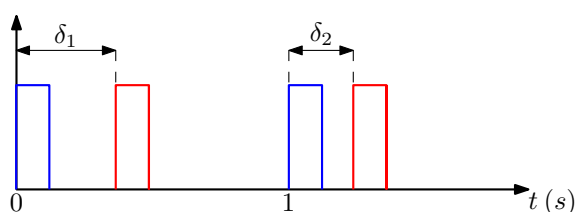
Jiným přístupem pro měření něčeho, co v podstatě má stejnou podkladnou myšlenku jako ztracený paket, je sledovat nepřítomnost nové informace ve chvíli, kdy nová informace má být k dispozici v rámci implementovaného vzorkovaného číslicového přenosu. Uvažujme model z obrázku 2.7. Měřená veličina je vzorkována, odeslána do sítě, která je zde nastíněna konstantním zpožděním a možnou ztrátou s pravděpodobností p , a doručena do jednotky, která s informací pracuje. Zpoždění uvážím menší než vzorkovací perioda, takže pokud je zpráva úspěšně doručena, je k dispozici v následujícím vzorkovaném čase. Zpožďovací člen se tedy chová jako člen z^{-1} . Ztrátový člen bude představovat paměť, která s pravděpodobností p nezmění svůj výstup ve vzorkovaném čase. Příjímá zprávy, zde označen jako PC , je schopen určit, že ve vzorkovaných časech k_4 a k_6 má k dispozici hodnoty staré dvě periody místo předpokládané jedné periody. To udělá buď 1) vedením logické hodnoty, která při přijetí nové hodnoty je nastavena do logické 1 a při zpracování ve vzorkovaném čase kT nastavena do logické 0 2) porovnáním časových značek, jak byly zavedeny v sekci 2.1, se vzorkovací periodou a současným časem. Toto je vesměs ekvivalentní se ztrátou paketu a je výběrově k dispozici pouze pro komunikaci číslicového přenosu. Navíc tento přístup sebou nese komunikační režii přibalováním dalších položek do paketů a odesíláním režijních zpráv pro navázání spojení.

2.3 Uchovávání a synchronizace času

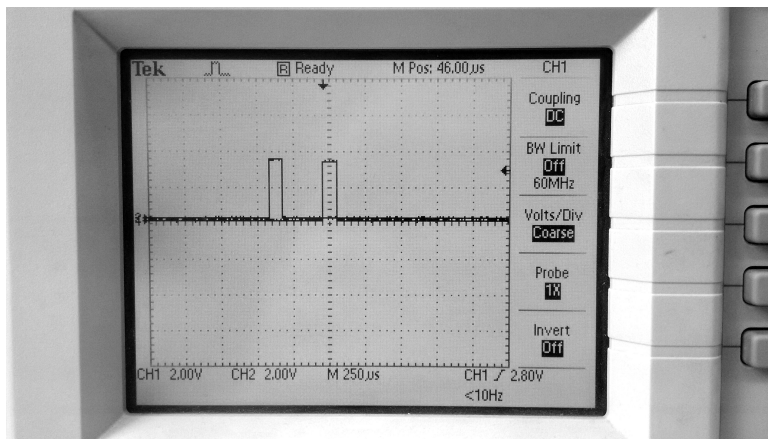
Pro správnou funkčnost aplikací, které pracují s informací o čase v rozptřených systémech mezi uzly sítě, je důležité, že čas uchovávaný v každém uzlu je správný. Tomu je rozumět tak, že čas přečtený na hodinách je ve všech okamžicích stejný jako čas jiných, referenčních hodin. Za referenční hodiny můžeme považovat například čas UTC (Coordinated Universal Time), ale pro měření zpoždění postačí čas libovolných hodin v síti, pokud tikají se relativně správnou rychlostí. Počítačové hodiny vykazují obecně různý rozptyl podle nedokonalostí oscilátorů i počáteční posun a je tedy na místě čas průběžně synchronizovat. Z dnes rozšířených řešení se nejvíce nabízejí Network Time Protocol, Precision Time Protocol, GPS a IRIG (Klecka, 2015). K nim lze přidat četné algoritmy pro bezdrátové senzorové sítě (Yadaf et al., 2015). Pro experimentální platformu autíček není GPS praktické kvůli nedostupnosti GPS signálu uvnitř budov a potřeby dodatečného hardware modulu. NTP a PTP jsou v principech podobné protokoly pracující v sítích LAN/WLAN/WAN s dostupnými implementacemi pro operační systém Linux. Otázkou je, která implementace těchto protokolů naplní požadavky aplikace.

2.3.1 Experimentální ověření synchronizace hodin

V projektu Slotcar platooning, kde interní hodiny v použitém Raspberry Pi operují s frekvencí 1 MHz, je rozlišení hodin rovné jedné milisekundě. Synchronizace, která dosáhne ustálené chyby v řádu stovek mikrosekund, by dostatečně naplnila požadavky aplikací pracujících s nejvyšším možným rozlišením. Protokoly NTP a PTP nebyly vyvinuty se zaměřením na síťové infrastruktury podobné té použité v Slotcar platooning, a tak nelze považovat experimentální výsledky o přesnosti prováděné kolem těchto protokolů za obecně platné i zde. Softwareové odhady, které implementace poskytují ve svých odhadech, mohou být vlivem problémů v síti (vysokým rozptylem zpoždění) nepravdivé. Proto jedním z implementovaných nástrojů je přesný generátor PPS signálu na fyzické GPIO piny. Za předpokladu, že testo-



Obrázek 2.8: Princip porovnávání PPS pulzů dvou hodin s probíhající synchronizací



Obrázek 2.9: Porovnávání synchronizace dvou hodin na osciloskopu

vané hodiny jsou již synchronizované pod jednu sekundu, lze na osciloskopu mezi dvěma PPS signály sledovat posun těchto hodin, jak je znázorněno v obrázku 2.8. Osciloskopy nemají nutně schopnost vypočítat a zaznamenat tento údaj, a tak je na člověku, pozorovat vývoj a zhodnotit použité synchronizační řešení. Fotografie na obrázku 2.9 zachycuje provedený experiment, ve kterém bylo použito NTPd, a posun hodin se ustálil na hodnotě $370\ \mu\text{s}$. Při experimentu s linuxptp a PTPd bez hardwareové podpory se ukázalo, že posun konvergoval k nule, ale v důsledku chyb se často vychýlil do řádu desítek milisekund. NTPd se tedy pro Slotcar platooning jeví jako vhodnější, avšak doba potřebná k synchronizaci je značně dlouhá, protože NTPd z důvodu primárního určení pro servery nepodporuje skok času. Alternativní implementace Chrony (Chrony) pracující s protokolem NTP toto řeší, a proto bude v projektu využita.

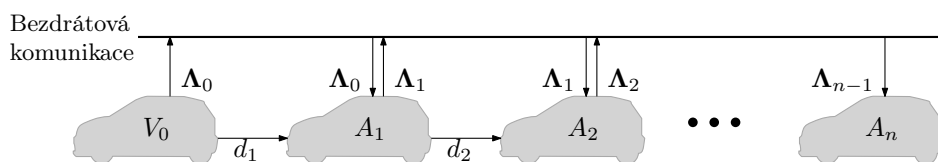
Tento experiment nelze považovat za ideální, například Laner (2013, příloha C) pracuje s mnohem specializovanějším vybavením a lepším postupem. Pro současnou aplikaci v projektu se však i tento jednodušší experiment jeví jako dostatečně dobrý za předpokladu, že je k dispozici rychlý osciloskop.

2.4 Rozšíření GUI

Pro demonstrační účely a zjednodušení provádění experimentů je vhodné mít k dispozici GUI, ze kterého lze aplikaci ovládat. K existující grafické části aplikace přibyl panel pro ovládání implementovaných nástrojů. Ukázka tohoto panelu je na obrázku 2.10. Přidává možnost spustit ukládání měřeného zpoždění v libovolném autíčku, zapnout a přerušit vytváření prázdný toku dat z autíčka pro zátěž sítě a zobrazit rychlý přehled o zpoždění ve formě

Kapitola 3

Využití komunikace při řízení



Obrázek 3.1: Schéma řízené kolony

Cooperative adaptive Cruise Control (CACC) zmíněné v úvodu je řídicí systém rychlosti. Spadá do inteligentních automatizovaných dálničních systémů, jejichž vývoji se věnuje řada iniciativ v Evropě, Americe a Japonsku.

Centrálním objektem pro zkoumání řízení podélné vzdálenosti vozidel je kolona, schematicky na obrázku 3.1. První vozidlo slouží pro kolonu jako vůdce a jeho řízení rychlosti nespadá pod CACC. Vůdce dokonce nemusí existovat, může být pro ostatní vozidla vytvořen virtuálně, a proto se literatura nezabývá jeho realizací. Vůdce slouží pouze jako zdroj vstupů do systému. Ostatní vozidla se již řídí pomocí CACC. Ve zbytku práce budou indexována pomocí $k \in \{1, 2, 3, \dots, n\}$.

3.1 CACC systém pro autíčka Slotcar platooning

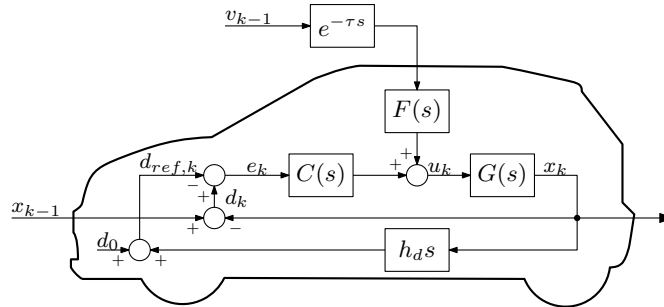
V koloně z obrázku 3.1 je pro každé auto cílem udržovat požadovanou vzdálenost, která je zavedena pro k -té vozidlo jako

$$d_{ref,k} = d_{0,k} + h_d \dot{x}_k \quad (3.1)$$

kde $d_{0,k}$ je požadovaná vzdálenost při zastavení vozidel, $h_{d,k}$ je časový rozestup a \dot{x}_k je rychlost k-tého vozidla. Takto určená vzdálenost má pozitivní vliv na stabilitu (Naus et al., 2010) a, jak bude odvozeno později, umožňuje určit kauzální přenos $F(s)$ a zaručit řetězcovou stabilitu kolony. Pro $h_{d,k} > 0$ roste udržovaná vzdálenost s rychlostí.

Uvažuje se topologie, kde po síti je do auta přenášena informace z předchozího auta. Zheng et al. (2017) toto nazývá predecessor-following (PF), přeloženo sledování předchozího vozidla, a Ploeg et al. (2015b) má jiný název, one-vehicle look-ahead. To je v kontrastu s topologiemi, kde je sledováno více aut složených z aut před a za řízeným autem a z vůdce. Další volbou při stanovování systému je, která veličina je vstupem do dopředného regulátoru F . Literatura se mezi stavy, kterými jsou pozice, rychlost a zrychlení, přiklání ke zrychlení, které nese informaci s nejrychlejší dynamikou. Zároveň uvažuje, že regulovaným vstupem auta je zrychlení. To odpovídá řídicí představě přidání nebo ubrání plynu. V návrhu CACC systému pro Slotcar platooning se přikloním k rychlosti jako „komunikované“ veličiny. To z důvodu, že měření zrychlení není v projektu zatím dobře provedeno a zpětnovazební smyčka pro sledování referenční rychlosti již byla pro autíčka kvalitně navržena.

Za všech předpokladů je CACC systém znázorněn na obrázku 3.2. Autíčka jsou homogenní a navržena stejným způsobem a tak, kde to nebude zapotřebí, je vynechána indexace. Přenos $G(s)$ zde představuje dynamiku autíčka, která



Obrázek 3.2: CACC řídicí smyčka k-tého vozidla v koloně

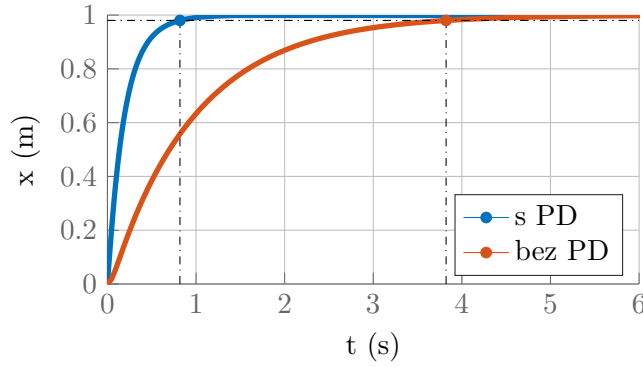
je navíc obalena zpětnou vazbou z rychlosti s PI regulátorem pro řízení na referenční rychlost

$$G(s) = \frac{k(s - z_1)}{s(s - p_1)(s - p_2)} \quad (3.2)$$

Druhý přenos, který je již určen, je přenos

$$H(s) = \frac{e_k(s)}{x_k(s)} = 1 + h_d s \quad (3.3)$$

který spojuje dvě cesty: 1) přes změnu požadované vzdálenosti z rovnice 3.1 2) přes měření vzdálenosti k předchozímu autu d_k . Zbývají dva bloky, a těmi jsou dopředný regulátor $F(s)$ a zpětnovazební regulátor $C(s)$.



Obrázek 3.3: Porovnání skokové odezvy ACC přenosu autíčka za použití PD regulátoru a bez

3.1.1 Volba regulátorů

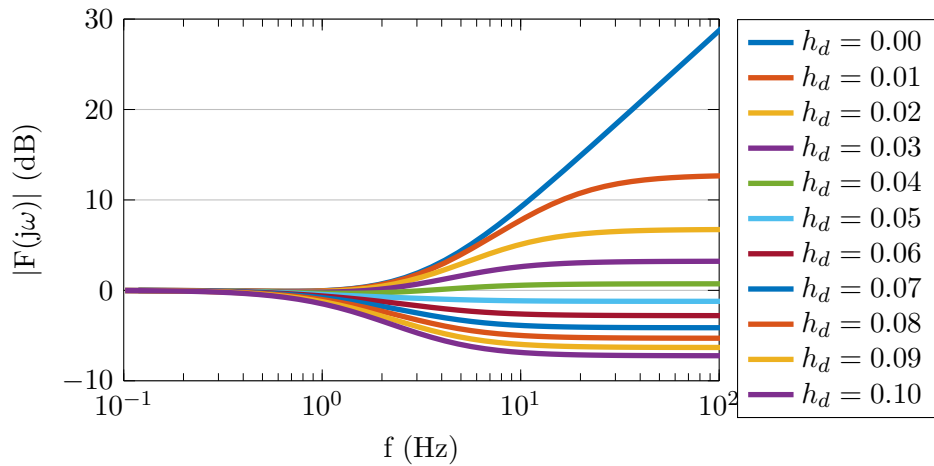
Návrh pokračuje volbou regulátoru C při odmyslení komunikace a časového rozestupu. Jedná se o návrh čistého ACC systému (Naus et al., 2010). Pak systém jednoho auta z obrázku 3.6 je jednoduchá zpětná vazba s přenosy $G(s)$ a $C(s)$. Přenos uzavřené smyčky je

$$T_{ACC,k}(s) = \frac{x_k}{x_{k-1}} = \frac{C(s)G(s)}{1 + C(s)G(s)} \quad (3.4)$$

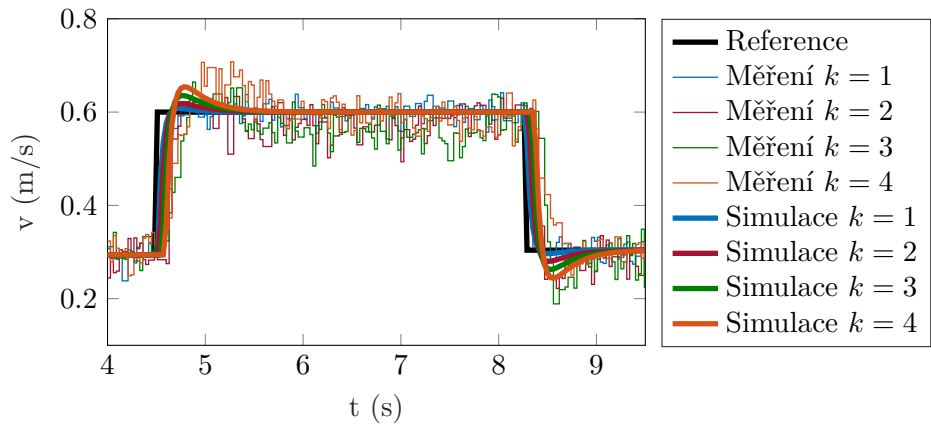
Volba regulátoru $C(s)$ umožňuje pozitivně změnit tento přenos. Požadavky na přenos F_{ACC} mohou být: 1) jednotkové ustálené zesílení 2) nízký nebo žádný překmit 3) rychlá odezva. První dva požadavky přenos splňuje i při použití $C(s) = 1$. Třetí je ve frekvenčním návrhu spojen s šířkou pásma ω_{BW} , kterou lze nastavit použitím PD regulátoru (Gao, 2003). Zlepšení odezvy na skok je patrné na obrázku 3.3. Odladěný PD regulátor s hodnotami $k_D = 0.2$ a $k_P = 5$ snížil dobu ustálení. Systém T_{ACC} je s astatismem typu 0 a zesílením $K_p = 1$. Jako takový sleduje skok reference (polohy) bez chyby v ustáleném stavu. Jinak tomu je při sledování rampy, t. j. skoku rychlosti, kde PD regulátor neudrží konečnou chybu. To je u kolony aut na dálnici nežádoucí. Řešením by bylo přidat integrační složku a v projektu je připraven takový regulátor. Bez uvažování dopředného regulátoru $F(s)$ je to jediným přímočarým řešením, jak řídit rychlost autíčka popsaného přenosem z rovnice 3.2. Přesto zůstává ACC systém schopný auto uřídit a je k dispozici ve chvíli, kdy komunikace selže.

Dopředný regulátor $F(s)$ CACC systému je jinou formou, jak zlepšit charakteristiky řízení. Ze struktury systému na obrázku 3.2 je vyvozen přenos

$$T_{CACC,k}(s) = \frac{e_k(s)}{x_{k-1}(s)} = \frac{1 - se^{-\tau s}F(s)G(s)H(s)}{1 + K(s)G(s)H(s)} \quad (3.5)$$



Obrázek 3.4: Porovnání zesílení $F(j\omega)$ pro různé hodnoty h_d



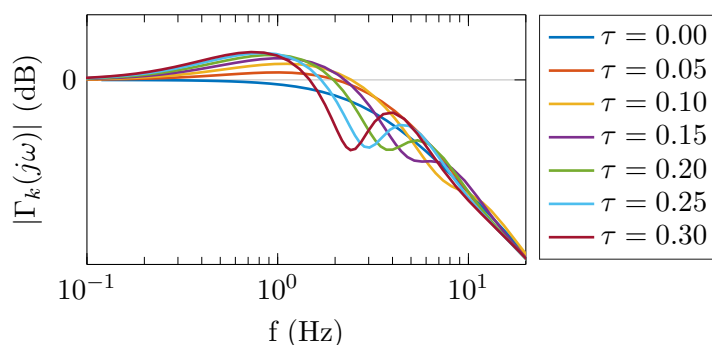
Obrázek 3.5: Časový průběh rychlostí CACC kolony sledující referenční vozidlo

Návrh regulátoru $F(s)$ vychází z požadavku na nulovou chybu e_k Naus et al. (2010). $F(s)$ je vybráno tak, aby čitatel přenosu z rovnice 3.5 byl nulový, a to za uvažování $\tau = 0$

$$F(s) = (sG(s)H(s))^{-1} = \frac{k_p(s - z_1)(s - z_2)}{(s - p_1)(s - p_2)} \quad (3.6)$$

Přenos je vázán na volbu časového rozestupu h_d . Pro nulové h_d je přenos nekauzální, pro příliš nízké h_d zesiluje vysoké frekvence, jak je vidět v grafu 3.4.

Na obrázku 3.5 je experimentální ověření implementace navrženého CACC systému vůči simulaci. Při implementaci byl použita hodnota $h_d = 0,05$ a systém byl diskretizován se vzorkovací periodou $T = 30$ ms a přepsán do Java kódu pro použití v autíčku. Přestože měření rychlosti je hrubé, je vidět, že CACC systém je aplikovatelný v praxi.



Obrázek 3.7: Zesílení přenosu $\Gamma_k(j\omega)$ s $h_d = 0,05$ pro různá zpoždění

Dosazení přenosu z pravé strany rovnice 3.12 do 3.11 za nulového zpoždění dá

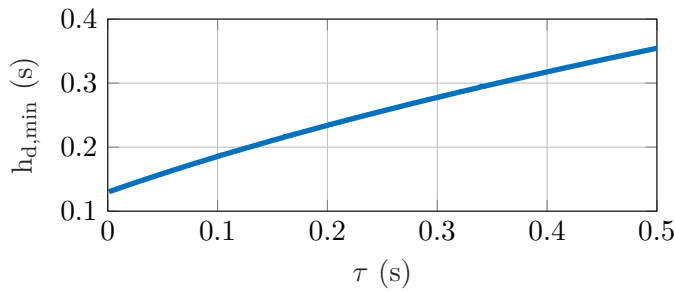
$$\sup_{\omega} \left| \frac{1}{j\omega h_d + 1} \right| \leq 1 \quad (3.13)$$

Podmínka je marginálně splněna nezávisle na h_d , v grafu na obrázku 3.7 je vidět, že pro hodnotu $h_d = 0,05$ je splněna pouze v případě nulového zpoždění.

3.2 Dopad zpoždění

Pro budoucí použití v automobilových sítích byla navržena a v roce 2010 schválena technologie IEEE 802.11p (802.11p), která se v řadě věcí liší od ad-hoc módů ostatních 802.11 sítí, avšak problémy zůstávají obdobné. V Khairnar – Kotecha (2013) byly prozkoumány formou simulací přístupové metody CSMA a STDMA v modelové situaci komunikace aut na dálnici. Hlavní závěr tohoto výzkumu je, že při vysokém zatížení sítě dochází k vážné degradaci. Nové STDMA má nižší propustnost, ale například neztrácí pakety ve frontách při zatížení. Místo toho implementuje přeposílání, které zvyšuje zpoždění. Ačkoli není vyvozen závěr o škále zpoždění, z grafů je vidět, že zpoždění vyšší jak 100 ms se při simulaci objevilo relativně často, takové pakety byly zahozeny a zahrnuty do ztracených.

Obrázek 3.7 ukazuje, že zpoždění zruší řetězcovou stabilitu pro systém navržený bez zpoždění. Podmínka pro řetězcovou stabilitu bude složitější než vztah 3.13, nebude-li zpoždění nulové. V takovém případě nelze z rovnice 3.11 analyticky vyvodit zajímavé závěry. (Ploeg et al., 2015b; Xing et al., 2016) adoptují strukturu, kde před dynamiku $G(s)$ je vložena inverze přenosu $H(s)$. V takovém případě je přenos $H(s)$ separovatelný z přenosu řetězcové stability $\Gamma_k(s)$. Bez takové úpravy je dostačujícím pozorováním, že se zvyšujícím



Obrázek 3.8: Nalezený minimální rozestup $h_{d,min}$ v závislosti na komunikačním zpoždění τ

zpožděním roste \mathcal{H}_∞ norma systému (viz obrázek 3.7 a se zvyšujícím h_d může \mathcal{H}_∞ norma klesat.

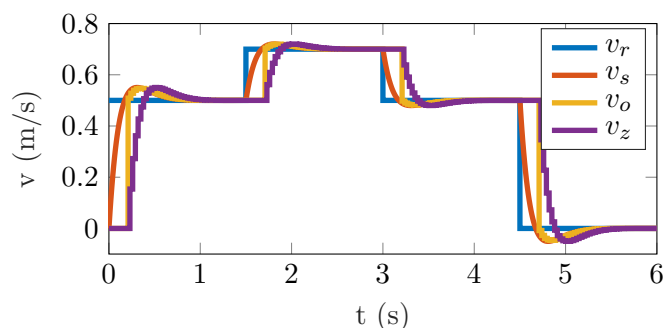
3.2.1 Kompenzace zpoždění

Dopravní zpoždění, nabývající tvaru $D(s) = e^{-\tau s}$, má vliv na řetězcovou stabilitu a představuje přenos nekonečného řádu. Pro umožnění zkoumání zpoždění v rámci stability bude zpoždění aproximováno pomocí Padého rozvoje 4. řádu (Xing et al., 2016). Důležité je, že aproximace neovlivní výpočet podmínky pro řetězcovou stabilitu. Cílem je zaručit stabilitu systému i při zpoždění.

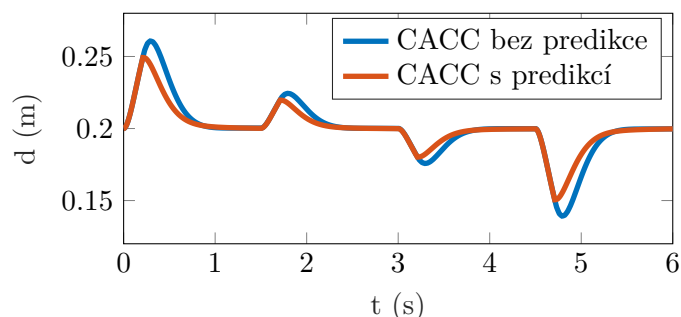
Nastavení h_d

Pomocí iterativního algoritmu 1, který postupně zvyšuje h_d , dokud není systém pro konkrétní zpoždění řetězcově stabilní, byla numericky nalezena křivka udávající závislost mezi minimálním potřebným časovým rozestupem $h_{d,min}$ a zpožděním komunikace, viz obrázek 3.8. Pokud je systém realizován s $h_d = h_{d,min}(\tau_h)$ pak je systém stabilní pro všechna $\tau \leq \tau_h$. Lze tedy využít dobrého odhadu nejvyššího zpoždění, které v systému nastane, například 95-tý kvantil změřený v první kapitole této práce, tabulce 2.2, jenom přeměřený pro konkrétní situaci. Samozřejmě se mohou vyskytnout zpoždění výrazně větší, a pak je potřeba uvažovat jinak než pouhým nastavováním časového rozestupu.

■ Určení současné rychlosti za znalosti systému v minulosti



(a) : Simulace rychlostí referenční, skutečné, odhadované a zpožděné



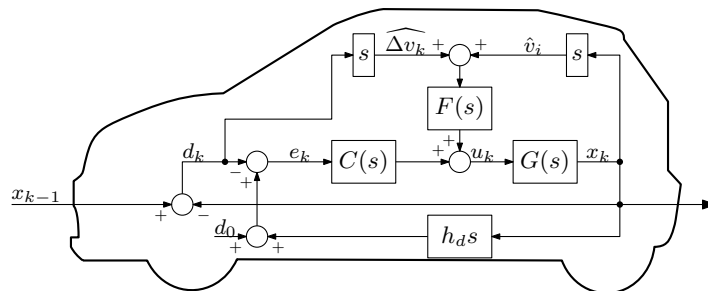
(b) : Simulace udržené vzdálenosti $d = 20$ cm CACC autíček s predikcí a bez predikce

Obrázek 3.9: Simulace predikce rychlosti za zpoždění $\tau = 200$ ms, vzorkování v síti $T = 30$ ms

Pokud nám v rámci datové jednotky přijde nejen rychlost předchozího auta, ale také ostatní stavy a všechny vstupy, a změří se zpoždění, pak by za znalosti stavového popisu systému bylo možné odhadnout současnou hodnotu libovolné veličiny ze známého popisu. Algoritmus 2 popisuje, jak by takovýto blok mohl být realizovatelný se synchronní částí, jejíž perioda by udávala vzorkovací periodu výstupu z bloku, a asynchronní částí, která by se spustila při přijetí paketu. Simulace použití tohoto estimátoru, bylo by mylné nazvat ho pozorovatel, je na obrázku 3.9. V Simulaci je vůdce, který sleduje referenční rychlost, a za ním vedle sebe dvě autíčka, která pomocí CACC udržují k vůdci vzdálenost 20 cm. Jedno autíčko je vybavené znalostí vůdčovi dynamiky a odhadujícím blokem před dopředným regulátorem $F(s)$ a druhé autíčko přijímá zpožděnou informaci přímo. Informace je přenášena s výrazným zpožděním 200 ms, a to má za následek, že především při náhlé změně vůdčovi rychlosti auta reagují zpožděně. Estimátor ze zpožděné informace vytáhne aktuální informaci, a tím autíčko reaguje lépe než to bez estimátoru. Vlastností tohoto algoritmu je, že pokračuje v odhadování, i když nedostane novou informaci v důsledku zpoždění nebo ztráty.

Navržený blok zatím zůstává pouze v simulaci, pro aplikování v praxi potřebuje dobře namodelovaný popis, ve kterém bude hrát významnou roli identifikovaný model tření, který by v současném algoritmu musel být lineární. Z popisu by mohl být model tření vypuštěn, pokud by v autíčku přibyl kompenzátor tření, jak jej navrhuje například van Geffen (2009). Pak by rozdíl mezi modelem bez tření a skutečností mohl být zanedbatelný a prediktor by se dal použít.

■ Degradovaný CACC systém



Obrázek 3.10: Degradovaná CACC řídicí smyčka

Ploeg et al. (2015a) popisuje degradovaný CACC systém (dCACC) tak, že v případě výpadku nebo nepoužitelnosti komunikace je vstup do přímé vazby nahrazen prediktorem zrychlení. Ten z měřené vzdálenosti a derivace této vzdálenosti (t. j. relativní rychlosti) odhaduje relativní zrychlení pomocí Singerova modelu zrychlení, ke kterému přičítá své vlastní zrychlení a získává zrychlení předchozího auta. Podobnou náhradu lze přidat k CACC systému navrženému v sekci 3.1. Systém na obrázku 3.10 znázorňuje ideální formu dCACC systému pro dráhová autíčka. Vstupem do přenosu $F(s)$ je odhad zrychlení předchozího vozidla na základě vlastního měření. To je součet relativní rychlosti předchozího vozidla a vlastní rychlosti. V této ideální formě by kolona s dCACC byla perfektně stabilní i pro nulový rozestup. Model zanedbává interní zpoždění, šum a fázový posun filtrů použitých při měření veličin. Tyto zanedbané prvky budou při sčítání odhadů hrát významnou roli, a jako takový je popis v současné formě nedostačující pro prozkoumání degradace CACC. Je možné experimentálně měřit frekvenční charakteristiky, vkládáním falešného signálu x_{k-1} , a na základě algoritmu podobného algoritmu 1 určit minimální rozestup. Tento rozestup bude nezávislý na zpoždění, a v momentu, kdy křivka z grafu 3.8 překročí hranici nalezeného rozestupu pro dCACC směrem nahoru, bude vhodné přepnout na dCACC systém (Ploeg et al., 2015a).

Kapitola 4

Závěr

Prvním úkolem práce bylo navrhnout a implementovat zaznamenávání zpoždění v bezdrátové síti, kterou autíčka využívala. V rámci tohoto úkolu byly popsány parametry, které v síti vystupují, a jaké mají vliv na zpoždění. Následně byl implementován nástroj, který síť z aplikace cíleně zatěžoval, pro zkoumání dopadu síťových parametrů, které patří do návrhu aplikace. Pro různé datové toky byl proveden experiment, jehož výsledkem je statistika, která návrháři dá přehled o očekávané míře zpoždění. Největším úskalím bezdrátových sítí se ukázal souběžný přístup více vysílačů, vedlejší provoz, nad kterým ve veřejných bezdrátových pásmech není kontrola. Přístup k médiu je u bezdrátových sítí silně nedeterministický, náhodný a náchylný a při přetížení sítě se veškerá komunikace začíná hroutit.

Pro použitelnost měření zpoždění bylo nutné synchronizovat hodiny počítačových modulů v autíčkách. Běžné synchronizační implementace se určením mýjely použití v autíčkách a synchronizace nebyla zaručená. Proto byl implementován generátor PPS signálu na piny počítačových modulů a několik implementací bylo otestováno pomocí osciloskopu. Implementace, která dosáhla nejlepších výsledků, se jmenuje Chrony a bude používána v projektu.

Dále byla v práci rozebrána ztráta paketů, důvody vzniku ztracených paketů a přístupy k měření. Stejně jako u zpoždění byla ztráta paketu spojena především s přetížením sítě. Bylo probráno, že měření ztráty za běhu není jednoduché ani zajímavé. Pro přenos důležitých paketů bylo navrženo používat TCP spojení nebo zavést požadované prvky z TCP virtuálně nad UDP. Pro přenos řídicích dat, který tvoří převážnou část toku dat mezi auty, není žádoucí zavádět mechanismy bojující se ztrátami paketů, a tak byl navržen

přístup, jak selektivně pro tento tok sledovat podobnou informaci, kterou je nepřítomnost nových dat. Později v druhé části práce byl navržen estimátor, který funguje i při nedostupnosti nových dat po delší dobu za cenu možné nestability celkového systému v důsledku ztráty přesnosti odhadu, a také byly popsány dvě strategie, když komunikace selže, a to přechod na ACC nebo dCACC.

Druhá část práce se zaměřila na návrh CACC systému a implementaci do autíček. Implementace byla usnadněna díky podkladům Martina Láda, který podobnou strukturu do projektu přidal v rámci své magisterské práce. Při implementaci a analýze bylo čerpáno z různých vědeckých článků posledních sedmi let, především nizozemské skupiny z Technické univerzity v Eindhoven. Byla probrána řetězcová stabilita kolony s CACC řízením a dopad komunikačního zpoždění. V rámci snahy o kompenzaci vlivů byl numericky určen vztah pro volbu časového rozestupu v závislosti na očekávaném zpoždění a dále byl navržen a simulačně ověřen člen, který co nejlépe odhaduje aktuální informaci ze staré, čímž do jisté míry odrušuje vliv zpoždění a ztráty paketů. Tento estimátor nebyl ověřen v praxi v důsledku nehotové práce na modelování a kompenzování tření. V poslední řadě je naznačen systém degradovaného CACC, který je možné použít jako náhradu CACC v případě, kdy komunikace selže. Práce na degradovaném CACC pro dráhová autíčka v projektu zůstává otevřená.

Příloha A

Algoritmy

Algoritmus 1 Numericky nalezení vztahu $h_{d,min}(\tau)$

Require: $G(s), C(s), F(s), \tau_{max}$

$hd \leftarrow 0.05$

$H \leftarrow$ transfer function $s \cdot hd + 1$

$D \leftarrow$ vector $0.001 : 0.001 : \tau_{max}$

$hds \leftarrow$ initialize vector size of D

$i \leftarrow 1$

for all τ from vector D (rising) **do**

$SS \leftarrow$ string stability transfer function of G, C, H, D, F, τ

$hnorm \leftarrow$ get peak gain of SS \triangleright Can be \mathcal{H}_∞ or \mathcal{L}_2 norm

while $hnorm > 1$ **do**

$hd \leftarrow hd + 0.0001$

$SS \leftarrow$ string stability transfer function of G, C, H, D, F, τ

$hnorm \leftarrow$ get peak gain of SS

end while

$hds(i) \leftarrow hd$

$i \leftarrow i + 1$

end for

Plot hds by D

Algoritmus 2 Odhad současného výstupu ze staré informace a znalosti systému

Require: Discrete state space of previous car M , N , O and sampling time T

Asynchronous on packet receipt:

$\mathbf{x} \leftarrow$ states read from packet

$\mathbf{u} \leftarrow$ inputs read from packet

$K \leftarrow \lceil \frac{\tau}{T} \rceil$ \triangleright τ is delay of packet, $K \cdot T$ is age of states \mathbf{x}

Synchronous with period T :

$y \leftarrow$ do STEP(K)

$K \leftarrow 1$ \triangleright Unless new packet arrives, next time do 1 step

function STEP(k)

$\mathbf{x} \leftarrow M^k \mathbf{x} + [M^{k-1}N + M^{k-2}N + \dots + N] \mathbf{u}$ \triangleright Simulate system

$y \leftarrow O\mathbf{x}$ \triangleright return output in time kT

return y

end function

Příloha B

Literatura

802.11p. IEEE Standard for Information technology– Local and metropolitan area networks– Specific requirements– Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications Amendment 6: Wireless Access in Vehicular Environments, 2010.

Chrony - Introduction [online]. Dostupné z: <https://chrony.tuxfamily.org/>.

DABROWSKI, C. – MILLS, K. The Influence of Realism on Congestion in Network Simulations. National Institute of Standards and Technology, 2016.

GAO, Z. Scaling and bandwidth-parameterization based controller tuning. In *Proceedings of the 2003 American Control Conference*. IEEE, 2003. doi: 10.1109/acc.2003.1242516.

iPerf - The ultimate speed test tool for TCP, UDP and SCTP [online]. 2017. Dostupné z: <https://iperf.fr/>.

JARDOSH, A. P. et al. Understanding congestion in IEEE 802.11b wireless networks. In *Proceedings of the 5th ACM SIGCOMM conference on Internet measurement - IMC '05*. ACM Press, 2005. doi: 10.1145/1330107.1330140.

JIAN, Y. – CHEN, S. Can CSMA/CA networks be made fair? In *Proceedings of the 14th ACM international conference on Mobile computing and networking - MobiCom '08*. ACM Press, 2008. doi: 10.1145/1409944.1409972.

KHAIRNAR, V. D. – KOTECHA, K. Performance of Vehicle-to-Vehicle Communication using IEEE 802.11p in Vehicular Ad-hoc Network Environment. *International Journal of Network Security & Its Applications*. 2013, 5, 2, s. 143–170. doi: 10.5121/ijnsa.2013.5212.

- KLECKA, R. Fundamentals of Precision Time Protocol. In *2015 ODVA Industry Conference*, 2015.
- KUROSE, J. F. – ROSS, K. W. *Computer Networking: A Top-Down Approach*. Addison-Wesley, 3. edition, 2000. ISBN 9780132856201.
- LANER, M. *Analyzing Packet Delay in Reactive Networks*. PhD thesis, Vienna University of Technology, 2013.
- LIANG, C.-Y. – PENG, H. String Stability Analysis of Adaptive Cruise Controlled Vehicles. *JSME International Journal Series C*. 2000, 43, 3, s. 671–677. doi: 10.1299/jsmec.43.671.
- LÁD, M. – HERMAN, I. – HURÁK, Z. Vehicular platooning experiments using autonomous slot cars. Dostupné z: <http://aa4cc.dce.fel.cvut.cz/sites/default/files/downloads/publications/ifac2017.pdf>. Ne-publikováno, 2017.
- MAECHLER, M. et al. *cluster: Cluster Analysis Basics and Extensions*, 2017. R package version 2.0.6.
- MARKOVSKI, V. Simulation and Analysis of Loss in IP Networks. Master's thesis, University Sts. Cyril and Methodius, 1996.
- MURRAY, D. et al. Measuring the reliability of 802.11 WiFi networks. In *2015 Internet Technologies and Applications (ITA)*. IEEE, 2015. doi: 10.1109/itecha.2015.7317401.
- NAUS, G. J. L. et al. String-Stable CACC Design and Experimental Validation: A Frequency-Domain Approach. *IEEE Transactions on Vehicular Technology*. 2010, 59, 9, s. 4268–4279. doi: 10.1109/tvt.2010.2076320.
- PARK, B. – MALAKORN, K. – LEE, J. Quantifying Benefits of Cooperative Adaptive Cruise Control Toward Sustainable Transportation System. Virginia Tech Transportation Institute, 2011.
- PIZANO, C. O. *Mitigating Network Congestion: Analytical Models, Optimization Methods and their Applications*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2010.
- PLOEG, J. – WOUW, N. – NIJMEIJER, H. Lp String Stability of Cascaded Systems: Application to Vehicle Platooning. *IEEE Transactions on Control Systems Technology*. 2014, 22, 2, s. 786–793. doi: 10.1109/tcst.2013.2258346.
- PLOEG, J. et al. Graceful Degradation of Cooperative Adaptive Cruise Control. *IEEE Transactions on Intelligent Transportation Systems*. feb 2015a, 16, 1, s. 488–497. doi: 10.1109/tits.2014.2349498.
- PLOEG, J. – WOUW, N. – NIJMEIJER, H. Fault Tolerance of Cooperative Vehicle Platoons Subject to Communication Delay. *IFAC-PapersOnLine*. 2015b, 48, 12, s. 352–357. doi: 10.1016/j.ifacol.2015.09.403.

- POSTEL, J. User Datagram Protocol. 1980.
- SCHRANK, D. et al. 2015 Urban Mobility Scorecard. Texas A&M Transportation Institute, 2015. Dostupné z: <https://mobility.tamu.edu/ums/report/>.
- GEFFEN, V. A study of friction models and friction compensation. Technische Universiteit Eindhoven, 2009. Dostupné z: <http://www.mate.tue.nl/mate/pdfs/11194.pdf>.
- XING, H. – PLOEG, J. – NIJMEIJER, H. Padé Approximation of Delays in Cooperative ACC Based on String Stability Requirements. *IEEE Transactions on Intelligent Vehicles*. 2016, 1, 3, s. 277–286. doi: 10.1109/tiv.2017.2662482.
- YADAF, G. K. – KUMAR, A. – RAGHUVANSHI, A. Analysis of Time Synchronization Protocols for Wireless Sensor Networks: A Survey. *IJCSMC*. 2015, 4, s. 1062–1068. ISSN 2320–088X.
- ZHENG, Y. et al. Distributed Model Predictive Control for Heterogeneous Vehicle Platoons under Unidirectional Topologies. *IEEE Transactions on Control Systems Technology*. 2017, 25, 3, s. 899–910. doi: 10.1109/tcst.2016.2594588.