Zborník vybraných prác študentskej vedeckej a odbornej činnosti

Študentská vedecká a odborná činnosť 27. apríl 2022

Organizátor:

Slovenská technická univerzita v Bratislave Fakulta elektrotechniky a informatiky

Editori:

Peter Benko Eva Miklovičová

Vydala Slovenská technická univerzita v Bratislave vo Vydavateľ stve SPEKTRUM STU. Náklad 50 ks CD nosičov

ISBN 978-80-227-5191-9

Obsah

V	yho	dnotenie prehliadky prác ŠVOČ	6
1	Me	echatronika a Aplikovaná informatika	12
	1.1	Informačná platforma pre vizualizáciu regionálnych dát Michal Czapala, Miroslav Behúl	13
	1.2	Kprobe-based Kernel Event Trace Logging for Exploit Detection Kristína Hrebeňárová, Peter Švec	19
	1.3	Servopohon s meničom SINAMICS Igor Bélai, Igor Bélai	24
	1.4	Monitorovanie a riadenie udalostného systému s využitím PLC a cloudu Simona Lopatniková, Oto Haffner	30
	1.5	Prepojenie DC motora so simulačným prostredím MATLAB Jaroslav Hajda, Danica Rosinová	36
2	Jao	drové a fyzikálne inžinierstvo, Elektronika a fotonika	41
	2.1	Technológie prípravy OLED (organických elektroluminiscenčných diód) s emisným dopantom Ondrej Kokavec, Michal Mičjan	42
	2.2	Ako dlho sa vybíja kondenzátor cez LED diódu? Kristína Gažová, Peter Bokes	46
	2.3	Teplotné pole na povrchu Lesieho kocky Karolína Kovalčíková, Peter Bokes	50
	2.4	Vybrané spätnoväzobné koeficienty reaktivity rýchleho plynom chladeného reaktora ALLEGRO Ivan Pančiak, Amine Bouhaddane	55
3	Ro	botika a kybernetika	61
	3.1	Aplikácia pre návrh diskrétneho stavového riadenia Kristína Okienková, Jana Paulusová	62
	3.2	Robustné riadenie polohy MIMO laboratórneho pohybového systému Róbert Málik, Ján Kardoš	68

3.3	Segmentation of retinal images damaged by diabetic retinopathy Michal Kováč, Jarmila Pavlovičová	74
3.4	Neuroevolúcia autonómneho vozidla Marko Chylík, Ivan Sekaj	80
3.5	Vizuálna odometria pre lietajúce robotické systémy Martin Sedláček, Jozef Rodina	87

Členovia programového a recenzného výboru

doc. Ing. Oto Haffner, PhD. Ing. Igor Bélai, PhD. Ing. Miroslav Behúl, PhD. Ing. Peter Švec Ing. Vladislav Novák prof. Ing. Márius Pavlovič, PhD. prof. Ing. Martin Weis, DrSc. Ing. Katarína Sedlačková, PhD. Mgr. Marek Vančo, PhD. prof. Ing. Peter Hubinský, PhD. Ing. Slavomír Kajan, PhD. Ing. Marián Tárník, PhD. Ing. Jana Paulusová, PhD.

Vyhodnotenie prehliadky prác ŠVOČ

Úvod

Na Fakulte elektrotechniky a informatiky STU v Bratislave sa dňa 27. apríla 2022 konala prehliadka prác študentskej vedeckej a odbornej činnosti (ŠVOČ). Táto študentská vedecká konferencia dáva príležitosť študentom bakalárskeho a inžinierskeho štúdia prezentovať výsledky, ktoré vypracovali samostatne, alebo pod vedením pedagogických a výskumných pracovníkov ústavov FEI STU, prípadne významných odborníkov z praxe.

Aj v roku 2022 bolo vyhlásených osem sekcií zodpovedajúcich študijným programom akreditovaným na FEI STU. Počet prác prihlásených do jednotlivých sekcií je uvedený v nasledovnej tabuľke.

	Sekcia	Ústav	Počet prác	Počet študentov
1	Aplikovaná informatika	ÚIM	2	2
2	Elektrotechnika	ÚE	0	0
3	Elektroenergetika	ÚEAE	0	0
4	Elektronika a fotonika	ÚEF	1	1
5	Jadrové a fyzikálne inžinierstvo	ÚJFI	3	3
6	Mechatronika	ÚAMT	3	3
7	Robotika a kybernetika	ÚRK	5	5
8	Telekomunikácie	ÚMIKT	0	0
			14	14

Prihlásené práce boli prezentované v troch sekciách nasledovne:

- Mechatronika a Aplikovaná informatika: 5 príspevkov,
- Robotika a kybernetika: 5 príspevkov,
- Jadrové a fyzikálne inžinierstvo a Elektronika a fotonika: 4 príspevky.

V dôsledku obmedzenia prezenčnej metódy štúdia spôsobeného globálnou pandémiou ochorenia COVID–19 aj revitalizáciou priestorov fakulty prebiehalo rokovanie v prvých dvoch sekciách online formou a v tretej sekcii prezenčnou formou. Príspevky v jednotlivých sekciách hodnotili najmenej trojčlenné komisie.

V každej sekcii boli najlepšie práce ocenené Cenou dekana FEI STU v Bratislave a ďalšie vynikajúce práce získali Diplom dekana FEI STU v Bratislave. Obe ocenenia sú spojené s jednorazovým mimoriadnym štipendiom.

Československá sekcia IEEE venovala do súť aže polročné členstvo v IEEE a tričko s logom spoločnosti pre jednu prácu z každej technickej sekcie. Ceny odovzdala členka predstavenstva československej sekcie IEEE Ing. Elena Cocherová, PhD.

Spoločnosť HUMUSOFT s.r.o. ocenila Cenou HUMUSOFTu tri najlepšie príspevky vypracované s využitím systému MATLAB a COMSOL Multiphysics. Autor víťaznej práce získal voucher na online školenie podľa vlastného výberu, pričom autori ďalších prác získali vecné dary. Ceny odovzdal zástupca spoločnosti HUMUSOFT na Slovensku Ing. Martin Foltin, PhD.

Po dvojročnej prestávke bola obnovená spolupráca so Zväzom slovenských vedeckotechnických spoločností (ZSVTS). Cena ZSVTS bola určená pre najlepšiu prácu študentskej vedeckej konferencie, pričom nositeľ ceny dostal možnosť stať sa Ambasádorom ZSVTS, ktorého úlohou bude sprostredkovávať

informácie o svojich vedeckých aktivitách aj o ZSVTS nielen na univerzite, ale aj vo svojom okolí. Práce nominované na toto ocenenie navrhli komisie v jednotlivých sekciách a následne najlepšiu prácu vybral a ocenenie v mene ZSVTS odovzdal predseda Slovenskej svetelnotechnickej spoločnosti prof. Ing. Dionýz Gašparovský, PhD.

Ocenené práce a ďalšie práce odporučené komisiami v sekciách boli zaradené do zborníka, ktorý bol vydaný v elektronickej forme.

Zoznam ocenených prác je uvedený v nasledujúcich tabuľ kách.

Prehľad ocenení

Cena dekana FEI STU v Bratislave

Meno	Názov práce	Vedúci práce	Sekcia
Ondrej Kokavec	Technológie prípravy OLED (organických elektroluminiscenčných diód) s emisným dopantom	Ing. Michal Mičjan, PhD.	Jadrové a fyzikálne inžinierstvo a Elektronika a fotonika
Bc. Michal Czapala	Informačná platforma pre vizualizáciu regionálnych dát	Ing. Miroslav Behúl, PhD.	Mechatronika a Aplikovaná informatika
Bc. Martin Sedláček	Vizuálna odometria pre lietajúce robotické systémy	Ing. Jozef Rodina, PhD.	Robotika a kybernetika

Diplom dekana FEI STU v Bratislave

Meno	Názov práce	Vedúci práce	Sekcia
Kristína Gažová	Ako dlho sa vybíja kondenzátor cez LED diódu?	doc. Ing. Peter Bokes, PhD.	Jadrové a fyzikálne inžinierstvo a Elektronika a fotonika
Kristína Hrebeňárová	Kprobe-based Kernel Event Trace Logging for Exploit Detection	Ing. Peter Švec	Mechatronika a Aplikovaná informatika
Bc. Kristína Okienková	Aplikácia pre návrh diskrétneho stavového riadenia	Ing. Jana Paulusová, PhD.	Robotika a kybernetika

Meno	Názov práce	Vedúci práce	Sekcia
Ivan Pančiak	Vybrané spätnoväzobné koeficienty reaktivity rýchleho plynom chladeného reaktora ALLEGRO	Ing. Amine Bouhaddane, PhD.	Jadrové a fyzikálne inžinierstvo a Elektronika a fotonika
Bc. Igor Bélai	Servopohon s meničom SINAMICS	Ing. Igor Bélai, PhD.	Mechatronika a Aplikovaná informatika
Bc. Róbert Málik	Robustné riadenie polohy MIMO laboratórneho pohybového systému	doc. Ing. Ján Kardoš, PhD.	Robotika a kybernetika

Cena Československej sekcie IEEE

Meno	Názov práce	Vedúci práce	Sekcia
Jaroslav Hajda	Prepojenie DC motora so simulačným prostredím MATLAB	prof. Ing. Danica Rosinová, PhD.	Mechatronika a Aplikovaná informatika
Bc. Igor Bélai	Servopohon s meničom SINAMICS	Ing. Igor Bélai, PhD.	Mechatronika a Aplikovaná informatika
Bc. Kristína Okienková	Aplikácia pre návrh diskrétneho stavového riadenia	Ing. Jana Paulusová, PhD.	Robotika a kybernetika

Cena ZSVTS

Meno	Názov práce	Vedúci práce	Sekcia
Bc. Marko Chylík	Neuroevolúcia autonómneho vozidla	prof. Ing. Ivan Sekaj, PhD.	Robotika a kybernetika

Vybrané práce v jednotlivých sekciách

V nasledujúcich kapitolách sú uvedené všetky ocenené práce ako aj ďalšie vybrané práce v jednotlivých sekciách.

Sekcia 1

Mechatronika a Aplikovaná informatika

Webová platforma pre vizualizáciu regionálnych dát

Bc. Michal Czapala¹

¹Ústav informatiky a matematiky, FEI STU v Bratislave

xczapalam@stuba.sk

Abstrakt – Práca sa zaoberá problematikou vizualizácie regionálnych dát z verejne dostupných zdrojov alebo dát poskytnutých používateľmi platformy. Prvá časť práce sa venuje prieskumu aktuálnych riešení na Slovensku a v zahraničí. V druhej časti práce je popísaný návrh riešenia a implementácia vizualizačnej platformy. Výsledná platforma má pomôcť menším inštitúciám a obciam zapojiť sa do trendu inteligentných miest a transparente informovať obyvateľov o dianí v ich okolí.

1 Úvod

V dnešnej dobe sa často skloňujú pojmy *transparentnosť* samospráv alebo inteligentné (smart) mestá. Medzi podstatné prvky potrebné k dosiahnutiu týchto cieľ ov patria dáta a ich dostupnosť. Mnohé samosprávy sa snažia poskytovať na svojich webových stránkach rôzne štatistické údaje, najčastejšie napríklad rozpočet alebo miera vytriedenia odpadu. Dostupnosť otvorených dát je však na Slovensku stále na nízkej úrovni, o čom svedčí aj správa Európskeho dátového portálu [1], ktorá skúmala mimo iného dostupnosť a kvalitu otvorených dát v štátoch Európskej únie. Slovensko sa v celkovom hodnotení umiestnilo až na predposlednom mieste, čo je najhoršie umiestnenie od začiatku vykonávania týchto prieskumov.

Zverejňovanie dát napomáha k informovanosti občanov, no nespracované textové údaje môžu byť častokrát náročnejšie na pochopenie. Pohľad na dlhú postupnosť číselných dát je dostačujúci napríklad v prípade, že hľadáme jednu alebo dve konkrétne hodnoty. Hľadanie trendov alebo hraničných hodnôt (tzv. *outliers*) v takomto formáte je však dosť nepraktické. Cieľom mojej práce je vytvorenie platformy na vizualizáciu regionálnych dát, ktorá bude uľahčovať pochopenie zverejňovaných dát a prispievať tak k informovanosti občanov.

2 Aktuálny stav problematiky

V zahraničí môžeme nájsť viacero príkladov inteligentných miest. Írska spoločnosť EirGrid Group prevádzkuje smart grid systém a údaje z neho poskytuje širokej verejnosti prostredníctvom online stránky [2]. Dostupné sú vizualizácie aktuálnej spotreby elektrickej energie, množstva produkovanej energie z veterných elektrární, podielu rôznych typov elektrární na výrobe energie alebo množstva generovaného CO₂.

Mesto Londýn poskytuje na svojich stránkach prístup k datasetom, ktoré sú spojené s hlavným mestom. Dostupné sú

dáta týkajúce sa zdravotníctva, ubytovania, životného prostredia, dopravy alebo ekonomiky. Dáta sú na webe prezentované pomocou prehľadných grafov a dostupných je viac ako 700 datasetov [3].

Napriek tomu, že Slovensko zaostáva v oblasti otvorených dát a inteligentných miest, tu nájdeme mestá, ktoré v tejto oblasti napredujú. Žilina prostredníctvom portálu City Dashboard informuje o aktuálnom počasí, dopravnej situácii, polohe vozidiel verejnej dopravy a poskytuje aj náhľad snímkov niektorých dopravných kamier [4].

Magistrát mesta Bratislava prevádzkuje portál OpenData Bratislava [5], na ktorom uverejňuje rôzne typy dát, napríklad o doprave, ekonomike, zdravotníctve a rozpočte. Dáta sú dostupné na stiahnutie a niektoré datasety obsahujú aj vizualizáciu napríklad vo forme grafov. Portál taktiež poskytuje prístup cez API rozhranie, k tomu je však potrebné požiadať o pridelenie API kľúča.

Vo všetkých spomenutých prípadoch bolo zjavné, že systém zverejňovania a vizualizácie dát si musel vytvoriť sám prevádzkovateľ, prípadne požiadal o vytvorenie systému na mieru externého dodávateľa. Výhodou takéhoto riešenia je, že systém je vytvorený podľa požiadaviek prevádzkovateľa a dokáže pokryť všetky potrebné prípady použitia.

Obstarávacie náklady podobného systému môžu byť značne vysoké, čo môže predstavovať prekážku najmä pre menšie subjekty ako sú obce alebo neziskové organizácie. V ď alšej časti práce sa budem venovať návrhu platformy, ktorá by mala pomôcť tento problém vyriešiť.

3 Návrh riešenia

Cieľ om je vytvoriť platformu, ktorá bude jednoduchá na implementáciu a používanie z pohľadu koncového užívateľa. Okrem vizualizácie dát by mala poskytovať možnosť autentifikácie a autorizácie užívateľ ov, ktorí budú môcť upravovať konfiguráciu vizualizácie a taktiež spravovať dáta, ktoré budú zobrazované. Od záujemcov sa preto bude vyžadovať vytvorenie účtu, pomocou ktorého budú platformu spravovať. Predpokladá sa, že záujemcov bude viacero a preto je vhodné platformu prevádzkovať ako službu, ktorú môže naraz využívať veľké množstvo inštitúcií. Tento typ prevádzky softvéru zodpovedá Software-as-a-Service modelu, kde si záujemcovia danú službu prenajímajú.

Stránky nájomcov by mali byť izolované, aby sa nemohli navzájom negatívne ovplyvňovať. Inštancie týchto stránok bude

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 1. Mechatronika a Aplikovaná informatika

nutné dynamicky vytvárať pri registrácii nových nájomcov a naopak odstraňovať pri zrušení účtu nájomcu. Túto požiadavku je ľahké splniť využitím kontajnerizácie – pri registrácii nového účtu sa z obrazu vytvorí inštancia nového kontajnera a pri zrušení účtu sa tento kontajner jednoducho odstráni.

Keď že stránky nájomcov bude obsluhovať viacero webových serverov, je nutné zabezpečiť správne smerovanie požiadaviek na konkrétne servery. Platforma bude prevádzkovaná na vlastnej verejnej doméne a presmerovanie na stránky konkrétneho nájomcu je možné vyriešiť s použitím sub-domén. Pred konkrétne webové servery platformy sa umiestni proxy server, ktorý na základe použitej sub-domény presmeruje každú požiadavku na správny server.

Dáta budú získavané a spracovávané z viacerých zdrojov a nájomcovia si budú môcť konfigurovať jednotlivé úlohy. Spracovávanie údajov môže byť časovo aj výkonovo náročné a ak by zdieľ alo prostriedky s webovým serverom, mohlo by to mať negatívny dopad na používateľ ov. Úlohy by sa mali preto vykonávať v samostatnom kontajneri.

Ukladanie spracovaných dát a údajov potrebných k prevádzke platformy bude zabezpečovať databázový systém. Väčšinu údajov budú predstavovať časozberné (*timeseries*) dáta, ale ukladať bude treba aj relačné dáta. Z týchto dôvodov som zvolil databázový systém PostgreSQL s rozšírením TimescaleDB, ktorý tieto požiadavky dokáže splniť.

Výsledná platforma bude pozostávať z viacerých nezávislých komponentov a bude musieť spĺňať niekoľko funkčných a nefunkčných požiadaviek.

3.1 Funkčné požiadavky

- Stránky nájomcov budú podporovať prihlásenie administrátorov
- Weby nájomcov bude možné dynamicky vytvárať a odstraňovať
- Orchestrátor po aktivácii účtu vytvorí kontajner so stránkou nájomcu
- Proxy server zabezpečí presmerovanie požiadaviek na správny webový server
- Databáza bude zabezpečovať sumarizáciu a odstraňovanie historických dát
- Úlohy na pozadí automaticky sťahujú dáta z nastavených zdrojov
- Stránky nájomcov obsahujú vizualizácie stiahnutých dát

3.2 Nefunkčné požiadavky

- Platforma bude realizovaná ako SaaS služba
- Stránky nájomcov a orchestrátor budú používať HTTPS protokol
- Stránky nájomcov budú od seba izolované
- Vytvorenie účtu nájomcu bude vyžadovať overenie
- Databáza bude optimalizovaná na prácu s údajmi v reálnom čase (time series data)

- Dáta nájomcov sú navzájom izolované
- Úlohy na pozadí budú naprogramované v jazyku C#

4 Implementácia

Výslednú platformu tvorí niekoľko dôležitých komponentov, ktoré sú od seba navzájom oddelené vo forme docker kontajnerov. V tejto kapitole sa budem venovať podrobnému popisu ich implementácie a spôsobu fungovania.



Obr. 1: Komponenty platformy

4.1 Perzistencia údajov

Kontajnery v sebe neobsahujú žiadne perzistentné dáta. Takáto separácia umožňuje napríklad veľmi jednoduchú aktualizáciu platformy – starý kontajner je nahradený novým s novou verziou aplikácie a všetky dáta zostanú zachované. Údaje sa ukladajú na niekoľko miest, ktorým sa budem venovať v nasledujúcich podkapitolách.

4.1.1 Hostiteľ ský systém

Priamo na hostiteľ skom systéme sú uložené zdrojové súbory potrebné k spusteniu platformy. Celý proces spustenia je vysoko automatizovaný, vďaka čomu prevádzkovateľ ovi platformy stačí získať zdrojové kódy platformy (napríklad stiahnutím z git repozitára), vyplniť niekoľ ko konfiguračných parametrov a zahájiť spustenie platformy. Všetky potrebné konfiguračné parametre sa nachádzajú v zložke *secrets* a pre úspešné spustenie sú vyžadované nasledujúce konfiguračné údaje:

Prihlasovacie údaje k databáze – databázový systém vytvorí administrátorský účet podľa zadaných prihlasovacích údajov.

Prihlasovacie údaje k proxy serveru – Pri prvom spustení platformy sa na serveri vytvorí administrátorský účet so zadanými údajmi.

Konfigurácia emailového klienta – Platforma využíva na zasielanie emailovej komunikácie externú smtp službu. V

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

tomto súbore sa nachádzajú údaje potrebné pre nastavenie SMTP klienta (adresa servera, port, heslo...).

SSL certifikáty – tieto certifikáty sú použité na zabezpečenie šifrovanej komunikácie medzi užívateľmi a serverom pomocou HTTPS protokolu.

Okrem týchto údajov sú na hostiteľ skom systéme uložené aj konfiguračné súbory pre docker, ktoré definujú ako sa majú jednotlivé komponenty vytvoriť a spustiť.

Posledné úložisko ktoré platforma využíva priamo na hostiteľ skom systéme sú zložky *proxydata* a *proxyletsencrypt*, v ktorých sú uložené súbory proxy servera. Ich obsah je priamo pod správou proxy servera a teda po ich vytvorení nie je potrebná žiadna ď alšia konfigurácia.

4.1.2 Docker jednotka

Údaje z databázy sa ukladajú do docker jednotky (*docker volume*). Je to špecializované úložisko pod správou Dockera a v porovnaní s ukladaním údajov na hostiteľ ský systém poskytuje vyšší výkon a lepšiu podporu pre paralelný prístup [6]. Databáza bude zapisovať väčšie množstvo údajov od viacerých nájomcov a docker jednotky pre tento prípad použitia lepším riešením ako vyššie spomínané prepojenie s hostiteľ ským systémom, ktoré je pomalšie.

4.2 Databázový systém

Za správu takmer všetkých dát platformy zodpovedá kontajner, na ktorom je spustený databázový systém PostgreSQL s rozšírením TimescaleDB, ktoré pridáva podporu práce s časozbernými dátami. Platforma využíva tri typy databáz:

Konfiguračná databáza – V tejto databáze sú uložené konfiguračné dáta potrebné na prevádzku platformy. Obsahuje informácie o registrovaných nájomcoch a ich kontajneroch, dostupných úlohách, konfigurácie konkrétnych inštancií úloh pre každého nájomcu a zoznam migrácií pre danú databázu.

Databáza užívateľ ských účtov – Pre každého nájomcu je vytvorená jeho vlastná databáza, v ktorej sa nachádzajú používateľ ské účty. Túto databázu si spravuje knižnica ASP.NET Core Identity, ktorú používam na účely autentifikácie a autorizácie. V databáze je niekoľ ko tabuliek, ktoré uchovávajú informácie o užívateľ och, ich rolách a právach a taktiež je prítomná aj tabuľ ka s informáciami o migráciách.

Databáza údajov – Každý nájomca má k dispozícii svoju vlastnú databázu, do ktorej sa ukladajú údaje získané z úloh na pozadí spolu s údajmi priamo nahratými od nájomcu. Táto databáza predstavuje hlavné úložisko údajov, preto je jej štruktúra najkomplexnejšia. Údaje sa ukladajú do tzv. hypertabuliek, ktoré sú optimalizované na prácu s časozbernými dátami. V databáze sú taktiež nakonfigurované automatizované úlohy, ktoré dáta spracovávajú pomocou agregačných funkcií a staré neagregované údaje postupne z databázy odstraňujú.

4.3 Proxy server

Na spravovanie presmerovaní medzi webovými stránkami rôznych nájomcov je nutné správne nastavenie proxy servera. V mojej práci som využil nginx proxy manager [7], ktorý je dostupný ako docker obraz, čo je ideálne pre môj prípad použitia. Obsahuje intuitívne grafické rozhranie, v mojom prípade som sa ale viac zameral na využitie API prístupu na správu konfigurácie. Koncové body API aktuálne nie sú nikde zdokumentované, preto som pre ich využitie najprv musel sledovať internetovú komunikáciu medzi grafickým rozhraním a back endom a týmto spôsobom zistiť koncové adresy a parametre volaní. Získané poznatky som neskôr využil pri konfigurácii proxy servera pomocou API prístupu z orchestrátora.

4.4 Úlohy na pozadí

Kontajner s úlohami na pozadí je zodpovedný za pravidelné vykonávanie zadaných úloh, ktoré spočívajú v získaní, spracovaní a následne uložení požadovaných údajov do databázy. Program na začiatku skontroluje migrácie konfiguračnej databázy a v prípade neaktuálnej verzie databázy počká na vykonanie migrácií. Keď je databáza aktuálna, nasleduje registrovanie úloh, ktoré je možné vykonávať. Úlohy sú postupne zapisované do databázy do tabuľ ky *Jobs*. V prípade, že úloha v databáze už existuje, tak sú aktualizované jej parametre.

Následne program prechádza zoznam zadaných úloh v tabuľ ke *JobDefinitions* pre každého nájomcu a v prípade čakajúcich migrácií databázy nájomcu počká na jej dokončenie. Vytvorí sa nová úloha podľa zadaných parametrov, pridá sa do zoznamu spustených úloh a začne vykonávať prácu s dátami.

Každá úloha počas svojho životného cyklu prejde niekoľkými stavmi. Po úspešnom uložení dát do databázy sa úloha uspí na nájomcom stanovený čas a po uplynutí tejto doby znova začne spracovávať dáta. V prípade, že sa počas vykonávania úlohy vyskytne neočakávaná chyba, udalosť sa zapíše do databázy a úloha skončí v chybovom stave. Ak nájomca vypne vykonávanie úlohy, pri nasledujúcom vykonávaní sa úloha prepne do vypnutého stavu a jej vykonávanie sa ukončí. Pri vypnutí programu spravujúceho vykonávané úlohy sa všetky úlohy prepnú do zrušeného stavu a vynúti sa ich ukončenie.

Po spustení všetkých úloh nájomcov program spustí na pozadí ešte jednu úlohu, ktorej účel je pravidelne kontrolovať stav spustených úloh a zmeny v databáze zadaných úloh. V prípade, že nejaká úloha skončila chybou alebo bola vypnutá nájomcom, program ju odstráni zo zoznamu spustených úloh. Po tomto kroku sa z databázy získa zoznam všetkých úloh, ktoré sa nenachádzajú v zozname spustených úloh a zároveň nie sú nastavené ako vypnuté. Úlohy prejdú rovnakým procesom ako pri počiatočnom spustení programu, spustia sa a naplánuje sa ďalšia kontrola spustených úloh.

4.5 Orchestrátor

Orchestrátor zohráva dve dôležité úlohy. Prvou úlohou je hosť ovanie úvodnej webovej stránky platformy, na ktorej nájomcovia nájdu informácie o platforme a jej funkciách a taktiež tu nájdu aj registračný formulár. Druhou, omnoho zložitejšou úlohou, je riadenie platformy ako celku.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

4.5.1 Webová stránka platformy

V prípade záujmu o využívanie služieb platformy vyplní budúci nájomca registračný formulár. Po jeho úspešnom vyplnení sa zobrazí správa informujúca o úspešnosti registrácie. V tomto momente sa v databáze platformy v tabuľ ke *Tenants* vytvoril záznam o novom nájomcovi spolu s potrebnými údajmi, ale zatiaľ je účet nájomcu vedený ako neaktivovaný. Aktiváciu je potrebné uskutočniť kliknutím na odkaz, ktorý bol odoslaný emailom do schránky nájomcu.

Po kliknutí na aktivačný odkaz v emaile sa začne proces aktivácie účtu nájomcu, aktivácia je z pohľadu nájomcu ukončená a orchestrátor spustí proces prípravy účtu. Aktivácia prebieha nasledovne:

- 1. Overenie správnosti aktivačného kódu
- 2. Zápis aktivácie účtu do databázy
- 3. Nastavenie presmerovania na proxy serveri
- 4. Vytvorenie kontajnera s webom nájomcu
- 5. Spustenie kontajnera nájomcu
- 6. Vytvorenie administrátorského účtu pre nájomcu

V prípade, že počas aktivácie nastane neočakávaná chyba, účet nájomcu sa opäť deaktivuje, nájomca je oboznámený s problémom ktorý nastal a účet môže skúsiť aktivovať neskôr, keď bude chyba odstránená. Po úspešnej aktivácii sa nájomcovi zobrazí odkaz na jeho novovytvorený web, ktorý môže začať spravovať.

4.5.2 Riadenie platformy

Orchestrátor je riadiacim centrom celej platformy a zodpovedá za jej spustenie, prevádzku aj ukončenie. S tým súvisí aj spravovanie kontajnerov, ktoré obsahujú webové stránky nájomcov. Tieto úlohy je možné splniť pomocou takzvanej *docker-out-of-docker* techniky, ktorej podstatou je pripojenie sa z vnútra kontajnera na docker, ktorý je spustený na hostiteľ skom systéme. Týmto spôsobom je možné z jedného kontajneru vytvárať, spravovať a mazať ostatné kontajnery spustené na systéme.

Pri spustení platformy sa orchestrátor najprv napojí na docker API na hostiteľ skom systéme, aby mohol spravovať ostatné kontajnery. Po úspešnom pripojení nasleduje pokus o spojenie s proxy serverom. Najprv program odošle *ping* dotaz na adresu proxy servera a čaká na odpoveď po dobu dvoch sekúnd. Ak odpoveď nedostane, pokus opakuje ešte trikrát a v prípade neúspechu vyvolá výnimku. V prípade úspešného spojenia so serverom sa začne proces autentifikácie a autorizácie.

Pri prvom spustení platformy má proxy server nastavené dočasné prihlasovacie údaje, ktoré orchestrátor využije na prihlásenie a následne sú tieto údaje okamžite zmenené. Nové údaje sú zadané podľa konfigurácie v súbore *secrets/npmCredentials.txt*, ktorý musí administrátor platformy pred prvým spustením vytvoriť, inak dôjde počas spúšťania platformy k výnimke a program sa ukončí. Tieto údaje môže neskôr administrátor využiť k prihláseniu do grafického rozhrania proxy servera, ktoré umožňuje manuálne spravovať konfiguráciu. Po zmene prihlasovacích údajov a prihlásení je nutné nakonfigurovať SSL certifikát potrebný pre zabezpečenie šifrovanej komunikácie. Program načíta súbory certifikátu zo zložky *secrets/ssl* a odošle požiadavku na proxy server, pomocou ktorej overí správnosť certifikátu. Po overení sú dáta o certifikáte odoslané na proxy server a tento certifikát sa ďalej použije na zabezpečenie šifrovaného spojenia.

Posledným krokom pri štarte platformy súvisiacim s proxy serverom je konfigurácia presmerovaní užívateľ ov na správny webový server. Program získa zoznam všetkých subdomén, ktoré sú zadefinované na serveri. Pri prvom spustení sú vytvorené presmerovania na hlavnú stránku platformy a do administrátorskej časti proxy servera. Následne je kontrolovaný zoznam presmerovaní so zoznamom nájomcov a v prípade, že nejaký nájomca nemá zadané presmerovanie na svoj web orchestrátor toto presmerovanie vytvorí.

Pre zasielanie emailových upozornení je nutné pri spustení nakonfigurovať emailového klienta, ktorý bude správy odosielať. Platforma vyžaduje na odosielanie správ emailový server tretej strany. Konfigurácia tohto servera je uložená v súbore *secrets/emailConfig.txt* z ktorého ju orchestrátor pri štarte prečíta.

Posledným krokom pri štarte orchestrátora je migrácia konfiguračnej databázy. Program sa pripojí k databáze a v prípade nutnosti aktualizuje jej schému. Po tomto poslednom kroku sa spustí webový server s domovskou stránkou platformy.

4.6 Web nájomcu

Kontajner s webom nájomcu obsahuje webový server hosť ujúci webové stránky konkrétneho nájomcu. Tieto kontajnery sú dynamicky vytvorené alebo odstránené pri registrácii nového nájomcu, prípadne zrušenie účtu existujúceho nájomcu. Webové stránky sú rozdelené na dve časti a ich obsahu sa budem venovať v nasledujúcich kapitolách.

4.6.1 Verejná časť

Verejná časť je prístupná všetkým návštevníkom stránky a jej hlavnou úlohou je vizualizácia dát v databáze. Na domovskej stránke sa nachádzajú dlaždice zobrazujúce aktuálne informácie (napríklad o teplote a vlhkosti vzduchu alebo meteorologických výstrahách), ktoré si vybral nájomca. Príklad domovskej stránky je zobrazený na obrázku 2.

Platform =	🖬 test org		🚨 Login
Main	Dashboard		
Dashboard			
8 Temperatures	There are no weather arens reported.		
lumidities	Temperature Nitra	PM10 Nitra	Humidity Nitra
n Pressures	10 20 30	100 150	40 50 60
इड ी Winds	0 40		30 70 80
📥 Pollution 🔹 👻	-10 50	50 200	10 90
🌲 Weather alerts 🛛 🕕	-20 11 °C 60	0 10 μg/m ³ 250	0 65 % 100
🚯 Custom Datasets 🛛 👻	Last update: 21.4.2022 19:00	Let update: 21.4.2022 20:00	Last update: 21.4.2022 19:00

Obr. 2: Domovská stránka s dlaždicami

Medzi rôznymi podstránkami sa používatelia môžu prepínať pomocou l'avého bočného menu. Po kliknutí na odkaz je užívateľ presmerovaný napríklad na podstránku s údajmi

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 3: Prehľ ad teploty vzduchu za posledných 30 dní

o teplote vzduchu, ktoré sú rozdelené podľa zadanej lokality. Zobrazovaná je aktuálna teplota ako aj historické údaje za posledné tri dni, sedem dní a 30 dní. Príklad vizualizácie 30 dňového prehľadu je na obrázku 3.

Podobným spôsobom sú vizualizované aj dáta poskytnuté nájomcom. Podporované sú timeseries dáta aj kategorické dáta, ktoré môže nájomca rozdeliť do rôznych kategórií. Príklad vizualizácie kategorických dát je zobrazený na obrázku 4.



Obr. 4: Príklad vizualizácie kategorických dát

Platforma taktiež podporuje vizualizáciu meteorologických výstrah pre lokality, ktoré si špecifikoval nájomca. Výstrahy sú farebne odlíšené, zoradené podľa stupňa ohrozenia a obsahujú informácie o lokalite, type, začiatku a trvaní výstrahy a odporúčanie vydané Slovenským hydrometeorologickým ústavom. Tieto dáta sú získavané z portálu Meteoalarm, kde ich priamo uverejňuje SHMÚ.

Platforma tiež obsahuje administrátorskú sekciu, ktorá je prístupná len prihláseným používateľ om. Po kliknutí na tlačidlo *Login* v hornej časti stránky je užívateľ presmerovaný na stránku s prihlasovacím formulárom, kde má možnosť prihlásiť sa alebo obnoviť svoje zabudnuté heslo.

4.6.2 Administrátorská časť

Po prihlásení má administrátor prístup do správcovskej časti webovej stránky. V hornom menu sa administrátorovi zobrazí

Weather Alerts • Time delays between this website and the website are possible, for the most up to date information about all published by the participating National Meteorological Services please use Severe flooding warning Flooding is expected Location from Michalors 11.4.2022 18:00 Vehidation na of-balaxies please use Michalors 11.4.2.202 18:00 Vehidation na of-balaxies please use Michalors 11.4.2.202 18:00 Vehidation na of-balaxies please use Vehidation ma of-balaxies please use Michalors 11.4.2.202 18:00 Vehidation ma of-balaxies please use Intermediations and water supplies are possible. Some evacuations may be required. weet 11.2.220 12:00 Moderate low temperature warning No detailed description available Location Tram Lotel Lovel Type Brezero 12.4.2022 0.000 12.4.2022 0.000 Moderate Low Tem					
Severe flood	ing warning				
Flooding is expecte	d				
Location	From	Until 12.4.2022 10:00	Level	Type Flooding	Certa Like
Michalovce Vzhľadom na očaka aktivity.	ivané zrážky a pohyb ľadov možni	o očakávať v danej oblasti vzostu	ip vodných hladín s možr	nosťou prekročenia Stupňov po	vodňovej
Michalovce Vzhľadom na očaki aktivity. Instructions: BE PF communications ar Imed: 11.42022 12.43	ivané zrážky a pohyb ľadov možní REPARED to protect yourself and y ad water supplies are possible. Sor	o očakávať v danej oblasti vzostu our property. Some flooding of p ne evacuations may be required.	ip vodných hladín s možr	nosťou prekročenia Stupňov po networks are possible. Disruptic	vodňovej in to power, iource Meteodam/SP
Severe flooding warning Less Less Tops Curative Elocation from Less Tops Curative Michalore 11.4.2022 18:00 12.4.2022 10:00 Severe Tops Curative Validation na odakkiwate źrakty a połtyb fladov możno odakkiwat v danej oblasti wostup vodných hladin s możnostou prekročenia Stuphor povedňovej akółty. Instructione BLE PREPAREID to protect yourself and your property. Some flooding of properties and transport networks are possible. Disuption to power, communications and water supplies are possible. Some evacuators my be required. Instruction Instruction <t< td=""></t<>					
Michalovce Vzhľadom na očaki aktivity. Instructions: BE Pf communications ar Inued: 11.42022 19.40 Moderate lov No detailed descrip	ivané zrážky a pohyb ľadov možní kEPARED to protect yourself and y di water supplies are possible. Sor w temperature warning titon available	o očakávať v danej oblasti vzostu o ur property. Some flooding of p ne evacuations may be required.	ip vodných hladín s možr	nosťou prekročenia Stupňov po networks are possible. Disruptic	vodňovej in to power, iouce: Meteodern/Si

Obr. 5: Vizualizácia meteorologických výstrah

možnosť nastavenia úloh na pozadí, nahratie vlastného súboru s dátami, konfigurácia dlaždíc na domovskej stránke alebo zrušenie účtu nájomcu.

Po kliknutí na možnosť spravovania úloh na pozadí sa administrátorovi zobrazí zoznam aktuálne zadaných úloh spolu so základnými informáciami o úlohe: Poradové číslo, názov, dátum vytvorenia, stav, interval vykonávania, časová pečiatka posledného vykonania úlohy a informácia o prípadných chybách počas vykonávania úlohy.

Na obrázku 6 taktiež vidieť, že administrátor môže pridávať nové úlohy a pri každej úlohe má možnosť ju editovať a meniť jej parametre, prípadne je možné jej vykonávanie dočasne pozastaviť. V prípade potreby je možné úlohu úplne odstrániť. Slúžia na to príslušné tlačidlá v stĺpci *Actions*. Po odstránení úlohy sa jej vykonávanie zastaví a záznam o úlohe spolu s jej parametrami je z databázy odstránený.

Add	job						
#	Actions	Name	Created	State	Interval	Last run	Error
1	i	Počasie	05.03.2022 18:54	Sleeping	11 min	12.03.2022 19:17	-
2	i Z	Počasie	05.03.2022 20:41	Sleeping	15 min	12.03.2022 19:16	—
3	i	Košice	07.03.2022 14:45	Sleeping	15 min	12.03.2022 19:16	-
4	i Z	Kvalita ovzdušia	09.03.2022 08:23	Sleeping	10 min	12.03.2022 19:09	-
5	i Z	Ovzdušie-centrum	02.03.2022 09:52	Sleeping	5 min	12.03.2022 19:12	-
6	[]	Teplota-centrum	08.03.2022 20:41	Sleeping	5 min	12.03.2022 19:15	-
7	[]	Teplota	10.03.2022 11:22	Sleeping	30 min	12.03.2022 19:02	-

Obr. 6: Administrácia úloh nájomcu

Nájomca má taktiež možnosť zrušiť svoj účet, čo má za následok vymazanie všetkých údajov o nájomcovi, ako aj údaje získané z úloh alebo vlastných údajov nahratých na platformu. Zrušenie účtu je nutné pre bezpečnosť ešte potvrdiť kliknutím na odkaz zaslaný na email nájomcu.

5 Záver

Výsledkom mojej práce je Software-as-a-Service platforma na vizualizáciu dát, ktorá podporuje viacero nájomcov – tenantov. Táto platforma by mala pomôcť menším subjektom Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

ako sú napríklad obce alebo neziskové organizácie pridať sa k ostatným smart mestám na Slovensku alebo v zahraničí a poskytnúť nástroj na zverejňovanie dát o svojom regióne širokej verejnosti. Platforma je ľahko rozšíriteľná a škálovateľná vďaka využitiu kontajnerizácie a je možné ju v budúcnosti ľahko aktualizovať, aby držala krok s aktuálnymi trendmi a požiadavkami používateľov. Výhodou oproti konkurencii je, že ponúka zároveň zverejňovanie vlastných datasetov nájomcu spolu s možnosť ou vizualizácie dát o vybranom regióne, ktoré sú dostupné z verejných zdrojov.

Pod'akovanie

Chcel by som sa poďakovať Ing. Miroslavovi Behúlovi, PhD. za odbornú pomoc a rady, ktoré mi poskytol na konzultáciách. Zároveň by som sa chcel poďakovať Slovenskému hydrometeorologickému ústavu za spoluprácu a poskytnutie údajov pre vypracovanie tejto práce.

Literatúra

- D. van Hersteren, L. van Knippenberg, R. Weyzen, E. Huyer, G. Cecconi data.europa.eu *OpenData Maturity Report 2021*, str. 5, 2022
- [2] EirGrid Group Smart Grid Dashboard Dostupné z: https://www.smartgriddashboard.com/
- [3] Greater London Authority London Datastore Dostupné z: https://data.london.gov.uk/
- [4] Mestský úrad v Žiline Smart Žilina City Dashboard Dostupné z: https://smart.zilina.sk/desktop/
- [5] Magistrát hlavného mesta SR Bratislavy OpenData Bratislava Dostupné z: https://opendata.bratislava.sk/
- [6] Docker, Inc. Use volumes Dostupné z: https://docs.docker.com/storage/volumes/
- [7] jc21 & contributors Nginx Proxy Manager Dostupné
 z: https://github.com/NginxProxyManager/nginx-proxymanager

Kprobe-based Kernel Event Trace Logging for Exploit Detection

Kristína Hrebeňárová¹, Peter Švec¹

¹ Institute of Computer Science and Mathematics, Faculty of Electrical Engineering and Information Technology, Slovak University of Technology, Ilkovičova 3, Bratislava, Slovakia

{xhrebenarova, peter.svec1}@stuba.sk

Abstract: With an increasing number of successful mitigations against memory corruption vulnerabilities taking place in user space, kernel itself has become an interesting target for many adversaries to exploit. As it runs with higher privileges, protection against malicious attacks is ought to be the kernel's uttermost priority. We devoted our time to examine several stack-based buffer overflow exploitation techniques with the aim of identifying the most popular execution paths taken. As our results do have a potential use in later automated exploit detection, a loadable kernel module (LKM) was developed with the ability to trace and log our findings - interesting system events repeatedly occuring in many exploits - via native Linux kernel framework - Kprobes.

1 Introduction

The C programming language was created in 1970s and has been gaining its traction since then [1]. Many times referred to as a high-level assembly language, it has become so popular mostly due to its speed, portability, and flexibility. Memory manipulation in C language, however, lacks automated bounds checking and allows for arbitrary memory address access. The fact that the C language relies in such a manner on the programmer's ability to not make mistakes whilst working with memory, introduces a whole separate class of vulnerabilities, bugs with the potential of being exploited by the attacker, so called memory corruption vulnerabilities. The class comprises vulnerabilities such as stack-based or heap-based overflows, double frees, memory leaks, and reading from not properly initialized memory.

An operating system is a complex software whose main purpose is to manage computer's hardware - such as central processing unit, memory and peripheral input/output devices, and to serve as an intermediary between user, one's applications and computer hardware [7]. Due to security reasons, a certain level of isolation between user applications and code of operating system, kernel, is needed, the CPU's privilege level controlling access to resources has been introduced. CPU can be running in multiple privilege modes out of which the most intriguing are kernel mode and user mode. When CPU runs in kernel mode, it has access to higher privilege instructions compared to ones available in user mode. Kernel runs in kernel mode, in so called *Ring 0*. User applications run in restricted mode which is referred to as *Ring 3*. Also, besides the aforementioned hardware separation, kernel protects itself at the software level via restricting virtual memory access to kernel space (available solely to kernel) and user space. User applications can interact with kernel via an interface implemented by kernel - the *system calls*. By these means, user space program can ask kernel to perform a specified service on its behalf.

Owing to the fact that C is very fast and efficient, virtually all operating systems are written in a mix of low-level machine language and C [6], and are, therefore, prone to suffer from either known or unknown presence of memory corruption vulnerabilities. We implemented a custom loadable kernel module - a driver for a special character device - containing a kernel stack-based buffer overflow vulnerability. User is able to interact with our character device through standard system call API, via open(), write() and read() system calls. By overwriting the return address, attacker is able to obtain arbitrary code execution exploit primitive.

The first release of Linux kernel dates back to 1991 [7]. Since then it has become the most widely used operating system kernel [3]. Being of an open-source nature, its source code is freely and publicly available. In recent years, it has become a tremendously appealing target for exploiting as more than 1850 Common Vulnerability Exposures (CVEs) regarding vulnerabilities in Linux kernel were reported during the last decade. Primary location of most of these was in kernel's drivers and modules [2]. Even more striking is the fact, that a lifetime of a security flaw in kernel is roughly 5.5 years [4]. Hence it is safe to say that, unfortunately, kernel developers give plenty of time to attackers to find and exploit numerous vulnerabilities before it gets patched.

In this paper, we propose a solution utilizing a Linux kernel framework Kprobes with which we are able to trace and log Linux kernel execution flow by placing hooks onto internal kernel functions of our choice.

This paper is structured as follows. In Section 2, we dive into stack-based buffer overflow vulnerability fundamentals. Section 3 is devoted to analysis of Linux kernel exploitation basics as well as current state of Linux kernel protection. We also identify subset of kernel events we want to trace for later automated exploit detection purposes. Our results are then contemplated and evaluated in section 5 and, lastly, in section

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Figure 1: Stack-based buffer overflow

6 we briefly discuss our results and their potential use in further work.

2 Stack-based buffer overflow

One of the most notoriously known and exploited vulnerability is stack-based buffer overflow. The exploitation that takes place in such adversary activity is often referred to as stack smashing. The prevalence of stack-based buffer overflow exploitation stems in the characteristic nature of the vulnerability with the simplicity of the attack itself.

Stack is a data structure with sequentially ordered data that is especially known for its last in, first out (LIFO) principle of data manipulation. Call stack is a crucial concept regarding program's execution as it is comprised of smaller building blocks - call stack frames. Interestingly, it grows downwards. New call stack frame is created upon another function's execution and it contains function's local variables, passed arguments, and, most importantly, the return address. Return address points to the exact place where execution flow ought to return after function finishes.

Buffer, in terms of programming, is a block of contiguous memory location with its main purpose of serving as a temporary storage for variables. Problem arises when the size of an input to be copied into buffer is not properly checked against intended buffer's size. As depicted in figure 1, insufficient (or absent) array bounds checking allows for much bigger input to be copied into buffer. As discussed earlier, the layout of the stack is fairly deterministic and local variables are stored before the return address. Without bounds checking, it is even possible for input to overwrite contents of memory addresses past its own call stack frame. However, from the exploitation perspective, the most interesting data to overwrite is the return address. It is possible to overwrite it with specially crafted input - so called payload in order to redirect program's control flow by attacker's will.

Linux kernel creates 2 separate call stacks for each thread - a path of execution within a process - to use in both user space and kernel space. Kernel stack has, similarly to user space one, very deterministic layout which makes it very favourable to exploit. The consequences of successful exploitation in kernel space is, unlike the one in user space, much substantial as it allows for full compromise of target machine.

Sekcia 1. Mechatronika a Aplikovaná informatika

3 Analysis

In this section, we discuss Linux kernel exploitation fundamentals and kernel functions worth of tracing.

3.1 Credentials

To make a gross simplification, each process has a corresponding set of credentials determining its rights to access various system resources as well as files in the system. Credentials are made up of (not only, but most importantly) user id (UID) and effective user id (EUID). Linux kernel interprets a UID of 0 as a root user user who can execute privileged commands and has a full control over the machine. Therefore, it is an attacker's main aim to tamper these credentials via exploiting a vulnerability in kernel code in order to obtain a UID of 0 and thus gain a full control over the target machine. Internally, Linux manages credentials with namely 2 functions - prepare_kernel_cred() and commit_creds(). The first one prepares a set of credentials based on the passed argument. Furthermore, if the argument equals to NULL, function prepares root's credentials. Later updates process's credentials with a set of new one.

3.2 ret2usr attack

One of the simplest form of exploiting kernel stack-based buffer overflow vulnerability is with redirection of control flow back to the user space memory which is controlled by the attacker [8]. Generally, as shown in figure 2, this is done by triggering a stackbased buffer overflow with a custom payload bigger than the originally intended local buffer size and overwriting the return address located on stack with an address of data residing in user space. The file /proc/kallsyms contains all the symbols kernel uses and knows about. By reading the file, adversary is able to obtain addresses on which the previously mentioned credential managing functions are stored at. With such knowledge, one can easily write a function to elevate his privileges to root ones and spawn a root shell. It is important to mention that this type of attack would not be functional if right kernel protections mitigations such as SMEP - would take place.

3.3 Kernel protections

Kernel implements numerous architecture-dependent protections against the stack- based buffer overflow exploitation. Here, we mention some of them.

3.3.1 Stack protector

The fundamental idea behind a stack protector is adding a random value - called stack canary - onto the stack right after the return address has been pushed [9]. Its value is then checked at function's prologue. While overflowing the local buffer with data (figure 3), canary value which is sitting right before it, gets overwritten and kernel can detect attacker's efforts of stack smashing. However, this protection can be easily bypassed if combined with an information leak. In kernel, the value of stack canary is generated randomly for each thread and its value stays

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Figure 2: ret2usr attack diagram

the same throughout the thread's lifetime [10]. If the attacker is able to read stack contents with the stack canary being present, one can then alter payload to include thread's kernel stack canary value and elegantly bypass stack protector. Again, to escalate one's privileges, it is necessary to call aforementioned internal kernel functions.

3.3.2 Other mitigations

In recent years, kernel developers tried hardening Linux kernel by introducing other protections.

SMEP. Kernel can be easily prohibited from executing code located in user space. This easily implementable mitigation is called **Superuser Mode Execution Protection** (SMEP) and would make our exploits malfunction terribly [11].

SMAP. **Supervisor Mode Access Prevention** (SMAP) prevents kernel from accessing pages from user space. If this mitigation is enabled, kernel will no longer be able to read or write from and to user space pages [11].

KASLR. Similarly to **Address Space Layout Randomization** (ASLR), the mitigation used in user space, there is **Kernel ASLR** (KASLR) which randomizes base address of kernel image. Its entropy of randomization is questionable and when combined with an information leak, it can be easily bypassed by computing the offset which makes it ineffectual [12].

KPTI. Kernel also implements a feature which completely separates user space and kernel space called **Kernel Page-Table Isolation** [13].

When trying to exploit a stack-based buffer overflow on a system with these protections enabled, one'd better switch from return-2-user attack to Return-Oriented Programming (ROP) approach [11].

Sekcia 1. Mechatronika a Aplikovaná informatika



stack frame 0

Figure 3: Overwriting of stack canary

3.4 System events identification

Whilst an exploit executes, numerous system events occur. Conveniently, we were able to pinpoint certain interesting system events ranging from process creation to the root shell spawning.

New process (thread) is created by fork() (clone()) system call. Both of these are internally in kernel implemented with kernel_clone() function. Reason for this is that Linux kernel does not distinguish between process and thread. Instead it refers to both as *tasks*.

Every task has a set of credentials associated with it. The functions manipulating with credentials were already mentioned - prepare_kernel_cred() and commit_creds().

System calls such as write() and open() with which the process can communicate and interact with our character device file do internally use _copy_from_user() (when writing to file) and _copy_to_user() (when reading from file).

4 Implementation

Our aim was to trace the execution of interesting system events and produce a log file with all necessary information about the kernel's execution flow at specific times via custom loadable kernel module which could be inserted into running system.

4.1 Kprobes

Trace-logging LKM of ours uses a mechanism native to kernel - Kprobes. Firstly introduced in Linux kernel v2.6.9-rc2, it is a lightweight subsystem for instrumentation of kernel code execution [5]. Kprobes - kernel probes - work dynamically by modifying the code in memory as they place a breakpoint - so called probepoint - upon an instruction of our choice. When this probepoint is hit, a kprobe handler associated with it will be executed. With this mechanism, we are able to trace various system events, investigate registers and function's arguments as well as their return values after they finish their execution.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

5 Results

We have successfully implemented a kernel module utilizing kprobe mechanism in order to log system events of our choice. We also wrote programs whose goal was to exploit a vulnerability located in another kernel module. Successfully, we have been able to trace the execution of those events and print log messages containing crucial data into console in real time.

5.1 Event trace logging

Segment depicted in figure 4 refers to new process creation. Parent process - shell - with its **process id** (PID) of 146, creates new (child) process with PID = 147 via fork() system call. Allocation of PID for new process is then followed by commit_creds() which sets process' new credentials based on parent process.

Reading from a file (figure 5) is handled by read() system call which internally calls function _copy_to_user() in order to get data from kernel space to user space. Noticeably, reading of as few as 16*B* of information leak from kernel stack were enough to read the value of stack canary.

In figure 6, one can see triggering of buffer overflow as user passes one's input into character device. _copy_from_user() is called with specified arguments. After the payload is written onto stack, we can observe the emerge of ret2usr attack as the privilege escalation comprising of prepare_kernel_cred() with NULL argument and commit_creds() which eventually sets process' credentials to root's. Lastly, when spawning root shell with UID = 0, we are able to notice new process (PID = 148) being created.

6 Conclusion

Prioritizing efficiency over security is not uncommon. Linux kernel, mostly written in unsafe C programming language, is not an exception. Throughout the years, kernel developers have been trying to harden Linux kernel as much as possible. Their continuous efforts have brought results in form of mitigations which are, however, still bypassable under certain conditions. In this paper, we proposed a set of key system events that take place during exploitation of stack-based buffer overflow vulnerability in kernel space. The outcome of our efforts is a functional loadable kernel module which utilizes Linux kernel subsystem Kprobes to trace and log those events. Produced results - logs containing crucial data - can later be used in further research and development of automated Linux kernel exploit detection including proper machine learning techniques.

References

[1] Ritchie, Dennis M and Kernighan, Brian W and Lesk, Michael E. The C programming language. *Prentice Hall Englewood Cliffs*, 1988. Sekcia 1. Mechatronika a Aplikovaná informatika

- [2] Shameli-Sendi, Alireza. Understanding Linux kernel vulnerabilities. *Journal of Computer Virology and Hacking Techniques*, p. 265-278, 2021.
- [3] www.linuxfoundation.org. 2020 Linux Kernel History Report. Retrieved April 21, 2022, from https://www.linuxfoundation.org/wp-content/uploads/ 2020_kernel_history_report_082720.pdf, 2020.
- [4] Cook, Kees. Kernel Self-Protection Project. Retrieved April 21, 2022, from https://www.outflux.net/slides/2021/lss/kspp.pdf, 2021.
- [5] Mavinakayanahalli, Ananth and Panchamukhi, Prasanna and Keniston, Jim and Keshavamurthy, Anil and Hiramatsu, Masami. Probing the guts of kprobes. *Linux Symposium*, Vol. 6. 2006.
- [6] Tanenbaum, Andrew. Modern operating systems. *Pearson Education, Inc.*, 2009.
- [7] Silberschatz, Abraham and Galvin, Peter B and Gagne, Greg Operating system concept. *John Wiley & Sons*, 2006.
- [8] Kemerlis, Vasileios P and Polychronakis, Michalis and Keromytis, Angelos D ret2dir: Rethinking kernel isolation. 23rd USENIX Security Symposium (USENIX Security 14), 2014.
- [9] Davis, Matt Sacrifice a canary upon the stack of the gods: on canaries, coal mines and stack sanity. *Linux Journal, Belltown Media Houston, TX*, 222 (2012): 4, 2012.
- [10] Sun, Jiadong and Zhou, Xia and Shen, Wenbo and Zhou, Yajin and Ren, Kui PESC: A Per System-Call Stack Canary Design for Linux Kernel. *Proceedings of the Tenth ACM Conference on Data and Application Security and Privacy*, 2020.
- [11] Gruss, Daniel and Maurice, Clémentine and Fogh, Anders and Lipp, Moritz and Mangard, Stefan Prefetch side-channel attacks: Bypassing SMAP and kernel ASLR. *Proceedings* of the 2016 ACM SIGSAC conference on computer and communications security, p.368-379, 2016.
- [12] Gruss, Daniel and Lipp, Moritz and Schwarz, Michael and Fellner, Richard and Maurice, Clémentine and Mangard, Stefan Kaslr is dead: long live kaslr. *International Symposium on Engineering Secure Software and Systems*, p.161-176, 2017.
- [13] Müller, Lars Kpti a mitigation method against meltdown. *Advanced Microkernel Operating Systems*, 2018.

Študentská vedecká a odborná činnosť Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 1. Mechatronika a Aplikovaná informatika

```
> ./bypass_canary
[alloc_pid] TIME = [13.846602106] PID = 146, UID = 1000, RET = {{NEW} PID = 147}
[commit_creds] TIME = [13.867079420] PID = 147, UID = 1000, ARG = {new cred -> uid = 1000, euid = 1000}
[commit_creds] TIME = [13.868535347] PID = 147, {NEW} UID = 1000, {NEW} EUID = 1000
```

Figure 4: Creation of process

Figure 5: Reading from /dev/vulnerable character device

Figure 6: Writing payload into character device

Servopohon s meničom SINAMICS

Igor Bélai

Ústav automobilovej mechatroniky, FEI STU v Bratislave

xbelaii@stuba.sk

Abstrakt – V príspevku je opísané pracovisko s frekvenčnými meničmi SINAMICS S120, jeho príprava pre použitie na cvičeniach a experimentálne výsledky rýchlostného servopohonu, ktoré potvrdili využiteľnosť pracoviska vo výuke.

1 Úvod

Na ÚAMT bolo pri riešení výskumného projektu "Platoon modeling and control for mixed autonomous and conventional vehicles: a laboratory experimental analysis" vytvorené pracovisko semi-fyzikálnych modelov vozidiel, ktoré používa meniče SINAMICS S120 a štyri synchrónne motory s permanentnými magnetmi. Toto pracovisko doposiaľ nebolo uvedené do prevádzky. Mojou úlohou bolo uviesť do prevádzky pohony pracoviska a pripraviť ich pre použitie na cvičeniach. Na uvedenie pracoviska do prevádzky bol využitý program Starter [1].

Príspevok sa zaoberá opisom pracoviska z hľadiska technického a programového vybavenia v kapitole 2, ďalej opisom navrhnutého a implementovaného programového vybavenia v kapitole 3, opisom identifikácie parametrov mechanického subsystému pohonov v kapitole 4, návrhom regulačného obvodu rýchlosti v kapitole 5 a výsledkami experimentov na reálnom servopohone v kapitole 6.

2 Opis pracoviska

2.1 Technické vybavenie

Frekvenčné meniče radu SINAMICS S sú určené pre komplexné úlohy riadenia pohybu so synchrónnymi a asynchrónnymi motormi. Umožňujú dosiahnuť vysokú presnosť regulácie a rýchlu odozvu.

Komunikácia frekvenčných meničov s inými prvkami riadiaceho systému je realizovaná prostredníctvom PROFINET IO.

Riadiace funkcie pohonu so SINAMICS S120 sú implementované v samostatnej riadiacej jednotke pohonu, ktorá môže riadiť niekoľko motorov. Riadenie pohonu môže využívať vektorové riadenie, skalárne riadenie V/f a reguláciu polohy alebo rýchlosti.

Systém d'alej obsahuje striedače, ktoré generujú striedavé napätia na napájanie motorov. Striedače konvertujú jednosmerné napätie vytvorené pomocou riadeného usmerňovača na striedavé fázové napätia motorov. Vstupom do usmerňovača je striedavé napätie napájacej siete. Striedače napájajú štyri trojfázové synchrónne motory s permanentnými magnetmi typu SIMOTICS S 1FT7 s integrovaným absolútnym snímačom polohy. Tieto motory majú nasledovné parametre:

nominálny výkon $P_N = 880$ W, nominálne otáčky $N_N = 6000$ ot/min, nominálny moment $M_N = 1.4$ Nm.



Obr. 1. Konfigurácia pracoviska s meničom SINAMICS S120

Pracovisko s frekvenčnými meničmi SINAMICS S120 je vybavené aj programovateľným logickým automatom (PLC) typu SIMATIC Open Controller. V PLC je implementovaný algoritmus riadenia pohonu pre použitie v pedagogickom procese.

SIMATIC Open Controller je priemyselný počítač, ktorý je určený pre riadiace a vizualizačné aplikácie. Používa operačný systém Windows 10 IoT Enterprise. Funkcionalita PLC je implementovaná programom S7-1500 Software Controller.

2.2 Programové vybavenie

Riadiaca jednotka pohonu sa konfiguruje z počítača, ktorý je s ňou prepojený prostredníctvom rozhrania Ethernet. Na samotnú konfiguráciu dodáva výrobca meniča program Starter, ktorý umožňuje načítať z meniča aktuálnu konfiguráciu

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

pohonov, vykonávať v nej zmeny a pomocou funkcie Autotuning nastaviť parametre regulátorov pohonov. Parametre pohonov je možné modifikovať v tabuľke parametrov alebo pomocou grafických dialógových okien, ktoré majú formu blokových schém.



Obr. 2. Dialógové okno programu Starter na nastavenie parametrov PI regulátora rýchlosti

Na konfiguráciu PLC je použitý program TIA Portal, ktorý umožňuje nastaviť konfiguráciu hardvéru PLC a komunikačnej siete (Profinet), vytvoriť riadiaci program v niektorom zo štandardizovaných programovacích jazykov (LAD, FBD, SCL), zaviesť ho do pamäte PLC, odladiť ho a vytvoriť grafickú vizualizáciu procesu.

využitia Z hľadiska pracoviska študentami sa predpokladá, že budú k nemu pristupovať pomocou programu Matlab, ktorý má slúžiť na spúšťanie experimentov a zobrazovanie nameraných veličín. Pri tomto sa využíva schopnosť Matlabu komunikovať s inými zariadeniami pomocou OPC UA (Open Platform Communication Unified Architecture). Ide o najnovšiu generáciu technológií od OPC Foundation pre bezpečný, spoľahlivý a nezávislý prenos nespracovaných údajov a predbežne spracovaných informácií od senzorov a polí až po riadiaci systém a do systémov plánovania výroby [2].

3 Vytvorené programové vybavenie pre PLC

Používateľský program, ktorý bol implementovaný do pamäte PLC, umožňuje využívať dvojicu pohonov pracoviska, pričom jednému pohonu je regulovaná rýchlosť a druhý pohon slúži ako zaťažovací. Hriadele motorov týchto pohonov sú prepojené tuhou spojkou. Na reguláciu rýchlosti bol navrhnutý PI regulátor s dvomi stupňami voľnosti, ktorého výstupom je žiadaný moment motora. Na vysielanie akčného zásahu z PLC do meniča boli použité funkcie MC_TORQUELIMITING a MOVE_VELOCITY z knižnice TIA Portal [3]. Aktuálna rýchlosť pohonu bola načítaná z poľa NIST_B telegramu 105, ktorý je použitý na prenos údajov medzi PLC a riadiacou jednotkou pohonu. Implementovaný PID regulátor s dvomi stupňami voľnosti bol naprogramovaný v jazyku SCL podľa blokovej schémy na Obr. 3, kde je:

w – žiadaná hodnota rýchlosti

- x aktuálna hodnota rýchlosti
- y výstup regulátora (žiadaný moment motora)
- Kp zosilnenie proporcionálnej zložky regulátora
- Tn časová konštanta integračnej zložky
- Tv časová konštanta derivačnej zložky
- Tf časová konštanta filtra derivačnej zložky
- Kwp dopredná väzba proporcionálnej zložky
- Kwv dopredná väzba derivačnej zložky
- Kfbk zosilnenie spätnej väzby anti-windupu



Obr. 3. Štruktúra spojitého PID regulátora s dvomi stupňami voľnosti

Perióda vzorkovania algoritmu regulácie rýchlosti je $T_s =$ 1 ms. S touto periódou vzorkovania sú načítavané hodnoty aktuálnej rýchlosti motora a vysielané akčné zásahy do riadiacej jednotky pohonu. Prenos veličín cez OPC UA do Matlabu s tak malou periódou vzorkovania nie je možný. Preto sa hodnoty všetkých veličín, ktoré majú byť zobrazované (spracovávané) v Matlabe zaznamenávajú do poľa premenných a ukladajú sa v pamäti PLC. Za týmto účelom bol vytvorený a implementovaný funkčný blok slúžiaci na záznam vzoriek zvolených veličín. Po uskutočnení experimentu sú tieto polia s hodnotami vzoriek veličín načítané cez OPC UA do Matlabu. Záznam hodnôt veličín je spúšťaný z Matlabu na to určenou premennou.

Pri experimentoch sa nemusí zaťažovací moment generovať len skokovo, ale môže mať aj iný priebeh. Kvôli tomu bol navrhnutý a implementovaný generátor žiadaného momentu záťaže, ktorý je schopný generovať nasledovné periodické priebehy momentu záťaže:

- Trojuholník
- Píla
- Sínus
- Obdĺžnik

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Používateľ môže nastaviť amplitúdu a frekvenciu signálu generátora žiadaného momentu záťaže.

3.1 Prepojenie PLC s Matlabom

Na komunikáciu Matlabu cez OPC UA sú v Matlabe implementované funkcie, ktoré umožňujú zriadiť spojenie typu OPC UA, prenášať údaje do/z Matlabu cez OPC UA a ukončiť spojenie OPC UA. Pre našu aplikáciu bol využitý matlabovský skript, ktorý tieto funkcie používa a umožňuje nastaviť potrebné hodnoty v pamäti PLC, spustiť experiment a načítať z PLC zaznamenané údaje. Podstatné časti skriptu sú uvedené nižšie.

Pri komunikácii OPC UA vystupuje Matlab ako klient a PLC ako server. V Matlabe je nutné zriadiť objekt typu klient, na čo slúži príkaz:

uaclient = opcua('192.168.0.1',4840);

Následne je možné zriadiť spojenie so serverom: connect(uaclient);

Po zriadení spojenia sa môžu prenášať údaje medzi serverom a klientom. Pred samotným prenosom údajov je potrebné vytvoriť uzly OPC UA, ktoré umožnia zápis/čítanie do/z premennej na strane servera. Pri zápise hodnoty zosilnenia regulátora rýchlosti sa uzol OPC UA vytvorí nasledovne:

Node_Kp =
opcuanode(3,"""OPC DB"".""Kp""",uaclient)

Samotná hodnota sa zapíše príkazom: writeValue(Node Kp, Kp)

Pri načítaní hodnoty sa využije príkaz readValue ('Uzol', 'premenná').

Po uskutočnení experimentu sa spojenie OPC UA zruší a to nasledovným príkazom:

disconnect(uaclient);

4 Identifikácia parametrov mechanického subsystému

V modeli mechanického subsystému pohonu bolo identifikované suché trenie M_{z0} a celkový moment zotrvačnosti *J*. Pri identifikácii suchého trenia bol plynule zvyšovaný moment motora *M* a vyhodnocovaná rýchlosť otáčania *N*. Moment motora, pri ktorom došlo k roztočeniu motora, bol identifikovaný ako suché trenie M_{z0} . Boli vykonané tri merania. Výsledná hodnota suchého trenia bola vypočítaná na základe ich aritmetického priemeru. Na Obr. 4 je priebeh momentu a rýchlosti motora z jedného merania. Identifikovaná hodnota $M_{z0} = 0.1017$ Nm.



Obr. 4. Priebeh momentu a rýchlosti motora pri identifikácii suchého trenia

Pri výpočte parametrov regulátora rýchlosti je potrebné poznať aj moment zotrvačnosti sústavy J (dva motory + spojka). Tento parameter je možné po konfigurácii nájsť v programe Starter – jeho hodnota v našom prípade je 2.2551866*10⁻⁴ kgm². Rozhodol som sa urobiť vlastnú identifikáciu momentu zotrvačnosti. Vychádzal som zo vzťahu pre uhlové zrýchlenie motora ε .

$$\varepsilon = \frac{M_m - M_{z0}}{J} \to J = \frac{M_m - M_{z0}}{\varepsilon} \tag{1}$$

Identifikácia J prebehla tak, že do motora, ktorý slúži na generovanie záťaže, bol vysielaný moment záťaže s obdĺžnikovým priebehom. Veľkosť momentu záťaže sa menila z hodnoty 0 Nm na 0.4 Nm. Perióda signálu momentu záťaže bola T = 0.125s. Z priebehu rýchlosti zaťažovaného motora bol vypočítaný rozdiel medzi rýchlosťou motora pri najväčšej záťaži a rýchlosťou motora pri najmenšej záťaži (Δ N). Bol zaznamenaný čas tejto zmeny rýchlosti (Δ T) a následne vypočítané zrýchlenie ε podľa (2) a moment zotrvačnosti J podľa (1).

$$\varepsilon = \frac{\Delta N}{\Delta T} \frac{\pi}{30} \tag{2}$$



Obr. 5. Priebeh rýchlosti motora pri identifikácii momentu zotrvačnosti

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Tento postup som zopakoval 5-krát. Z identifikovaných hodnôt momentov zotrvačnosti bola vypočítaná stredná hodnota, ktorá bola stanovená ako výsledná hodnota momentu zotrvačnosti J. Táto hodnota bola ďalej používaná pri experimentoch.

$$J = \frac{\sum_{i=1}^{5} J_i}{5} = 2.52 * 10^{-4} \, kgm^2 \tag{3}$$

5 Návrh regulačného obvodu rýchlosti

Na reguláciu rýchlosti bol použitý PI regulátor s dvomi stupňami voľnosti podľa Obr. 3 pre Tv = 0. Na Obr. 6 je štruktúra regulačného obvodu rýchlosti, s prenosovou funkciou (4). Táto štruktúra bola použitá aj pri simulácii.



Obr. 6. Simulačná schéma regulačného obvodu rýchlosti

$$F(s) = \frac{n}{n^*} = \frac{\frac{K_p}{JT_n} \frac{30}{\pi} (K_{wp}T_n s + 1)}{s^2 + s\frac{K_p}{J} \frac{30}{\pi} + \frac{K_p}{JT_n} \frac{30}{\pi}}$$
(4)

Metódou rozmiestňovania pólov boli odvodené vzťahy na výpočet parametrov regulátora rýchlosti. Póly ROR sú rovnaké ako má prenosová funkcia druhého rádu:

$$G_2(s) = \frac{\omega_0^2}{s^2 + 2\xi\omega_0 s + \omega_0^2}$$
(5)

Odvodené vzťahy pre výpočet parametrov regulátora $K_{\rm p}$ a $T_{\rm n}$:

$$K_p = \frac{\pi}{30} 2\xi \omega_0 J$$

$$T_n = \frac{30K_p}{\pi I \omega_0^2} = \frac{2\xi}{\omega_0}$$
(6)

Pomocou K_{wp} je možné ovplyvniť hodnotu nuly v čitateli prenosovej funkcie ROR. V prípade, že platí $\xi = 1$, je možné správnou voľbou K_{wp} kompenzovať jeden pól prenosovej funkcie F(s) a to za podmienky:

$$\xi = 1 \to \frac{1}{\omega_0} = K_{wp} T_n \to K_{wp} = \frac{1}{\omega_0 T_n} = \frac{1}{2}$$
 (7)

6 Experimentálne overenie vlastností regulačného obvodu rýchlosti

Pre regulačný obvod s uvedeným regulátorom rýchlosti bolo vykonaných 5 meraní. Merania sledovali vplyv zmeny

parametrov regulátora na priebeh rýchlosti motora pri skokovej zmene žiadanej rýchlosti z 1000 ot/min na 1500 ot/min v čase 1 s a pri pôsobení momentu záťaže, ktorý nadobúdal hodnoty v rozsahu <0 ; 0.4> Nm. Moment záťaže začal pôsobiť v čase 2 s.

Pri prvom meraní bol sledovaný vplyv ω_0 na priebeh rýchlosti motora, pričom moment záťaže mal obdĺžnikový priebeh.



Obr. 7. Priebeh rýchlosti motora pre rôzne hodnoty ω_0



Obr. 8. Priebeh žiadaného momentu motora pre rôzne hodnoty ω_0

Z Obr. 7 vyplýva, že čím väčšie je ω_0 , tým rýchlejšie sa ustáli rýchlosť motora na žiadanej hodnote. Pri vyššej hodnote ω_0 je aj vplyv poruchy, resp. zmeny momentu záťaže, menší – regulátor rýchlejšie reaguje na poruchu a teda rýchlejšie odstráni jej vplyv – avšak za cenu väčšieho akčného zásahu (Obr. 8).

Druhé meranie bolo zamerané na vyhodnotenie vplyvu zmeny tlmenia ξ na priebeh rýchlosti motora, pričom moment záťaže mal obdĺžnikový priebeh.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 9.Priebeh rýchlosti motora pre rôzne hodnoty &



Obr. 10. Priebeh žiadaného momentu motora pre rôzne hodnoty $\boldsymbol{\xi}$

Z Obr. 9 je vidieť, že pri $\xi = 0.7$, dochádza k malému preregulovaniu. Takisto aj vplyv momentu záťaže je väčší. Regulátor s takýmto tlmením je však rýchlejší. Zároveň je však potrebné vyvíjať väčší akčný zásah pri zmene rýchlosti (Obr. 10).

Tretie meranie sledovalo vplyv zmeny K_{wp} na priebeh rýchlosti motora, pričom moment záťaže mal obdĺžnikový priebeh.



Obr. 11. Priebeh rýchlosti motora pri rôznych hodnotách Kwp



Obr. 12. Priebeh žiadaného momentu pri rôznych hodnotách $K_{\rm wp}$

Z Obr. 11 je vidieť, že zmena hodnoty K_{wp} má vplyv iba na zmenu hodnoty žiadanej rýchlosti. Pri hodnote $K_{wp} = 0.5$, reaguje regulátor rýchlejšie na zmenu žiadanej rýchlosti, tým pádom rýchlosť motora dosiahne žiadanú hodnotu za kratší čas. V tomto prípade regulátor generuje väčší akčný zásah, tzn. žiadaný moment motora bude väčší pri skokovej zmene rýchlosti (Obr. 12). Pri pôsobení momentu záťaže nezáleží na hodnote K_{wp} - veľkosť odchýlky rýchlosti od žiadanej hodnoty spôsobenej vplyvom momentu záťaže je rovnaká, takisto je rovnaký aj čas, za ktorý dôjde k odstráneniu vplyvu momentu záťaže.

V štvrtom meraní sú porovnávané priebehy rýchlosti motora z experimentu a zo simulácie pri pôsobení rôznych priebehov momentu záťaže. Toto meranie bolo vykonané pre štyri rôzne priebehy momentu záťaže – trojuholník, píla, sínus, obdĺžnik. Na Obr. 13 je zobrazený priebeh žiadanej rýchlosti motora, rýchlosti motora zo simulácie, rýchlosti motora z experimentu, takisto aj priebeh momentu záťaže a žiadaného momentu motora pre experiment, pričom ide o meranie so sínusovým priebehom momentu záťaže. Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 13.Priebeh rýchlosti motora zo simulácie a experimentu so žiadaným momentom pre experiment a s momentom záťaže so sínusovým priebehom

Pri piatom meraní bol porovnávaný priebeh rýchlosti motora zo simulácie a z experimentu pri rôznych hodnotách tlmenia ξ .



Obr. 14. Priebeh žiadanej rýchlosti a rýchlosti motora zo simulácie a z experimentu pri rôznych hodnotách tlmenia ξ

Z Obr. 14 vidieť, že priebehy rýchlosti sa značne nelíšia pri simulácii aj pri experimente. Rozdiely medzi priebehmi rýchlosti z experimentu a medzi priebehmi rýchlosti zo simulácie sú pravdepodobne spôsobené tým, že v simulačnom modeli sú zanedbané niektoré vlastnosti reálneho systému.

7 Záver

Navrhnuté a prezentované riešenie umožňuje využitie pohonov s frekvenčným meničom SINAMICS S120 na cvičeniach. Používateľ môže overovať vlastnosti rýchlostného servopohonu s PI regulátorom rýchlosti, pričom jeho parametre môže nastaviť ním zvolenou metódou. V práci je uvedený príklad nastavenia parametrov regulátora rýchlosti metódou rozmiestňovania pólov. Rýchlostný servopohon môže byť zaťažovaný momentom záťaže, so štyrmi predvolenými Sekcia 1. Mechatronika a Aplikovaná informatika

priebehmi. Frekvenciu a amplitúdu momentu záťaže môže používateľ nastaviť.

Literatúra

- [1] SINAMICS S120 With Starter Commissioning Manual. 11/2017. Siemens AG, Norimberg, Nemecko
- [2] OPC Jednotná Architektúra Interoperabilita pre Industrie
 4.0 a Internet of Things. Verzia 07/ November 2017. OPC
 Foundation. Dostupné na: https://elearning.mechatronika.cool/wpcontent/uploads/2018/11/
 OPC_UA_Celkový_prehľad_Brožúra_Verzia_nov_2017 _SK.pdf
- [3] SIMATIC S7-1500 with SINAMICS S120: Torque limiting with MC_TorqueLimiting. 04/2021. Siemens AG. Dostupné na: < https://cache.industry.siemens.com/dl/files/023/10979502 3/att_1064127/v2/109795023_S120-torquereduction_en.pdf>

Monitorovanie a riadenie udalostného systému s využitím PLC a cloudu

Simona Lopatniková Ústav informatiky a matematiky Fakulta elektrotechniky a informatiky Slovenská technická univerzita v Bratislave Email: xlopatnikova@stuba.sk

Abstrakt—Príspevok sa venuje problematike modelovania, simulácie a riadenia virtuálneho udalostného systému s využitím softPLC a cloudovej aplikácie. Využitie virtuálneho - simulovaného udalostného systému prináša viacero výhod. Nakoľko výstup je vo forme prípadovej štúdie využiteľnej pre oblasť vzdelávania odborníkov v odbore aplikovanej mechatroniky a inžinierskej informatiky, bolo potrebné zabezpečiť, aby takýto študijný podklad bol pre záujemcov ľahko dostupný. Virtuálny model nie je nutné kupovať, servisovať a nezaberá žiadny fyzický priestor, a teda ho je možné ponúknuť širšej skupine záujemcov.

I. Úvod

Riadený udalostný systém v reálnej praxi pozostáva napríklad zo série výrobných liniek [1] . V školskom prostredí pre účely vzdelávania sú využívané rôzne zmenšené modely (napr. od firmy F ischertechnik a lebo L EGO [2]). Vhodnou alternatívou sú tiež virtuálne udalostné systémy, ktoré sa dajú realizovať napríklad pomocou Factory IO, pričom táto aplikácia je využívaná v predloženom príspevku.

V našej prípadovej štúdii (obrázok č.1) by sme namiesto tradičných platených PLC editorov a runtime chceli demonštrovať využitie voľ ne dostupného open-source aplikačného systému OpenPLC. OpenPLC však nepodporuje moderný komunikačný štandard OPC UA, ale len konvenčný protokol Modbus. OpenPLC síce vie komunikovať s Factory I/O pomocou tohto protokolu a realizovať pomocou neho riadiace procesy, chceme však dáta posielať aj do cloudu [3] a prípadne ich poskytnúť ď alším klientom cez OPC UA. Preto potrebujeme prostredník Node-RED, ktorý nie je potrebný k riadeniu, ale máme ho tam tiež kvôli zdieľ aniu dát do cloudu cez MQTT a tiež cez OPC UA server, ku ktorému vieme pripojiť širokú paletu klientov. V našom prípade budeme testovať OPC UA klienta UAExpert.

II. ŠPECIFIKÁCIA A SPRÁVANIE UDALOSTNÉHO SYSTÉMU

Udalostný systém by sa dal vo všeobecnosti zadefinovať ako systém, ktorý môže nadobúdať viacero stavov, pričom prechody medzi jednotlivými stavmi sú vyvolávané udalosť ami (angl. event-driven system). V tomto prípade uvažujeme virtuálny model udalostného systému vo forme výrobného systému, ktorý pozostáva z výrobného pásu a obrábacieho centra. Model sa skladá z viacerých dielov, ktoré by dynamicky mali reagovať na udalosti, ktoré v systéme nastávajú. Na

Oto Haffner

Ústav automobilovej mechatroniky Fakulta elektrotechniky a informatiky Slovenská technická univerzita v Bratislave Email: oto.haffner@stuba.sk



Obr. 1. OpenPLC v prepojení s Node-RED a MS Azure



Obr. 2. Schéma výrobného systému

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 3. Celkový pohľad na virtuálnu výrobnú linku - uhol č.1

vstupe (emitter) sú v istých časových intervaloch generované kusy materiálu (polovýrobky), ktoré je nutné dopraviť do obrábacieho centra. Tu sa z nich procesom obrábania stávajú výrobky pripravené na expedíciu zákazníkom. Premiestňovanie materiálu a výrobkov zabezpečujú pásové dopravníky. Tých máme konkrétne šesť (C1 - C6). Funkčnosť dopravníkov zabezpečuje sedem svetelných senzorov (R1 - R7). Pomocou emittera vyšleme materiál na pás, kde ho zachytí senzor R1 a následne sa spustí pás C1. Až daný produkt zaznamená senzor R3, spustí sa nám rohový pás C2 a následne aj pás C3, ktorým materiál poputuje do obrábacieho centra. Tu sa spracuje na konečný produkt, ktorý postupuje na pás C4, ktorý sa spustí po zaznamenaní senzorom R5. Následne senzorom R6 spustíme pásy C5 a C6. Konečný výrobok zaznamenáme senzorom R7, ktorý by mal tiež zabezpečovať rátanie počtu hotových výrobkov. Je vhodné využiť zarovnávač, ktorým je docielené, aby produkt správne vstúpil do obrábacieho centra. Je potrebné tiež zabezpečiť, aby v systéme nedochádzalo ku kolíziám medzi kusmi materiálu a tiež medzi hotovými výrobkami.

Na vytvorenie výrobnej linky vo Factory I/O sme potrebovali viacero dielov. V nasledujúcich bodoch si ich všetky vymenujeme: pásový dopravník (angl. Belt conveyor), zakrivený pásový dopravník (angl. Curved belt conveyor), zarovnávač (angl. Aligners), kĺzačka (angl. Chute cnveyor), Neopracovaný materiál / polovýrobok (angl. Raw Material), retroreflexný senzor a odrazka (angl. Retroreflective S ensor a nd Reflector), vysielač (angl. Emitter), odstraňovač (angl. Remover) a obrábacie centrum (angl. Machining Center).

Na obrázku č.3 vidíme celkové pohľady na virtuálny model výrobnej linky.

III. RIADENIE UDALOSTNÉHO SYSTÉMU

Aby sme dokázali riadiť udalostný systém, používame na to v tejto prípadovej štúdii open-source editor a runtime OpenPLC. V OpenPLC používame jazyk Ladder logic (tzv. rebríkový diagram).

V hlavnom okne v hornej časti sa nám zobrazí vstupné pole, ktoré nám umožňuje vytvárať premenné pre PLC program. Programy PLC sú cyklické, čo znamená, že začnú prvým





Obr. 4. Rebríkový diagram - časť 1.

príkazom, skončia posledným príkazom, chvíľ u počkajú a následne sa príkazy opakujú. Predvolená hodnota čakania je 20 ms. Nechali sme predvolenú hodnotu, aby nám program nespotreboval 100% CPU zariadenia, keď že naše PLC chápeme len ako softPLC a pre demonštračné účely je to postačujúce. Používame globálne premenné RetroreflectiveSensor od 1 po 7, ktoré nám snímajú produkt na páse, na základe ktorého zapíname alebo vypíname pás, prípadne počítame počet polovýrobkov alebo výrobkov. Hodnoty výrobkov a následne prepočty riešime cez globálne premenné Result 1 až 3. Result1 nám počíta koľko polovýrobkov sa pustilo na pás. Result2 hovorí, koľko je hotových koncových výrobkov a Result3 nám udáva, koľko výrobkov je aktuálne vo výrobe. Pre beh výrobných pásov používame premenné Conveyor 1 **až 6**. Ďalej používame rôzne časovače a matematické funkcie. Časovačmi príklad docieľ ujeme aby nedošlo ku kolíziám. A senzormi tiež rátame dobu výrobkov na páse, aby opäť nedošlo k ďalším kolíziám.

V strednej časti tvoríme diagram. Diagram tvoríme za pomoci panelu s nástrojmi. Program je "skenovaný" alebo spúšť aný procesorom zľ ava doprava a zhora nadol. Pre zhrnutie, náš diagram nám zabezpečuje chod celého výrobného pásu. Obsahuje 11 priečok, 11 kontaktov, 15 cievok, 3 SR funkčné bloky, 6 TOF funčných blokov, 2 ADD aritmetické funkcie a 1 SUB aritmetickú funkciu.

Prvá časť LD diagramu nám zabezpečuje chod prvých troch pásových dopravníkov. Ak nám **RetroreflectiveSensor1** a **RetroreflectiveSensor3** zaznamená materiál, tak pásy **Conveyor1**, **Conveyor2** a **Conveyor3** sa nám spustia na dobu

Študentská vedecká a odborná činnosť Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 5. Rebríkový diagram - časť 2.

10 sekúnd a materiál putuje po výrobných pásoch smerom k stroju. Avšak ak **RetroreflectiveSensor1** a **RetroreflectiveSensor3** nezaznamená materiál na páse dlhšie ako 10 sekúnd, pásy sa vypnú. Máme tam ešte ďalšiu podmienku a to takú, že ak **RetroreflectiveSensor3** z achytí materiál, tak **Conveyor2** a **Conveyor3** sa spustí na 3 sekundy a to z toho dôvodu, že to môžu byť posledné kusy a **RetroreflectiveSensor1** by už materiál nezachytil a nastala by kolízia. Rovnaký postup nastáva aj pri druhej časti s **RetroreflectiveSensorom5** a **RetroreflectiveSensorom6** a **Conveyorom4**, **Conveyorom5** a **Conveyorom6**.

Druhú časť LD diagramu tvoria bloky a aritmetické funkcie. **RetroreflectiveSensor2** nám vchádza do funkcie ADD a slúži nám na počítanie kusov surového materiálu (polovýrobkov), ktoré sa položili na daný pás a hodnotu nám zapíše do premennej **Result1. RetroreflectiveSensor7** nám slúži na počítanie už vyrobených podstavcov (výrobkov) a hodnotu si zapisujeme do premennej **Result2**. Aritmetická funkcia SUB potom tieto dva premenné **Result1** a **Result2** odčíta a táto hodnota nám určuje, koľko výrobkov / kusov materiálu sa momentálne nachádza na pásových dopravníkoch.

IV. KOMUNIKÁCIA

Komunikácia prebieha najprv medzi OpenPLC a Factory IO. Komunikáciu zabezpečujeme za pomoci Modbus protokolu a to z toho dôvodu, že OpenPLC nepodporuje modernejšie protokoly (napr. OPC UA). OpenPLC sa v tomto prípade správa ako Modbus server a Factory IO sa správa ako Modbus klient. Keď že chceme dáta posielať do cloudu, prípadne ich poskytnúť ď alším klientom pomocou OPC UA servera, potrebujeme prostredníka vo forme Node-REDu. Node-RED nie je teda potrebný pre samotné riadenie výroby, ale využívame ho pre možnosť zdieľ ania dát pre OPC UA server a do cloudu. Z Node-REDu do cloudu Microsoft Azure a naspať komunikujeme pomocou komunikačného protokolu MQTT.

A. Prepojenie OpenPLC runtime s Node-RED

Aby sme mohli správne prepojiť OpenPLC runtime s Node-RED, tak je potrebné pochopiť, ako funguje protokol Modbus a čo je v našom prípade server a čo je klient. OpenPLC dokonca dokáže súčasne fungovať aj ako server a aj ako klient, čo však aktívne využívať nebudeme.

Modbus ponúka 4 typy prenášaných dát.

- **Discrete Input** Jeden bit (BOOL), ktorý sa používa na binárny vstup (napríklad zo senzorov). V našom prípade to sú adresy typu %IX. Dokáže ho zapisovať len Modbus server.
- **Coil** Jeden bit (BOOL), ktorý sa väčšinou používa na binárny výstup. V našom prípade sú to adresy typu %QX. Dokáže ho zapisovať nielen server, ale aj klient!
- **Input Register** 16-bitový register určený len na čítanie. Je to niečo ako Discrete Input, akurát to nie je BOOL, ale je to INT s veľkosťou 16-bit, ktorý môže byť unsigned alebo signed.
- Holding Register 16-bitový register určený na čítanie a aj zapisovanie. Je to niečo ako Coil, akurát to nie je BOOL, ale je to INT s veľkosť ou 16-bit, ktorý môže byť unsigned alebo signed.

Ako je to teda s OpenPLC a jeho fungovaním? Ako sme uviedli, OpenPLC dokáže fungovať ako Modbus klient a aj ako Modbus server. On tak funguje vlastne súčasne, akurát pre každý mód má vyhradené iné adresy.

Ďalej nám treba nastaviť Factory I/O na Modbus klienta. Nastavujeme *localhost*, kde nám beží OpenPLC, čo je 127.0.0.1. Ďalej je treba nastaviť, že digitálne vstupy využívame na Coiloch. Logickejšie by bolo využiť Inputy, ale nakoľko Modbus klient dokáže zapisovať len do Coils, tak pre vstupy musíme využiť Coils. A posledné treba nastaviť I/O Points a tam nastavujeme vstupy, výstupy podľa toho, koľko priestoru potrebujeme.

A ako posledné vo Factory I/O treba priradiť správne k súčastiam Factory I/O vstupné a výstupné premenné (obrázok č.6). Pozor dávame na adresy, ktoré nám musia sedieť s OpenPLC.

Do OpenPLC runtime si nahráme náš program z OpenPLC editora, ktorý si spustíme. Prejdeme teda na obrázok č. 7, kde si ukážeme, ako načítavame hodnoty z OpenPLC runtime a ako sa napájame cez protokol Modbus. Možnosť komunikácie cez Modbus sme získali nainštalovaním knižnice *node-red-contrib-modbus 5.14.1*.

Cez uzol, ktorý sme nazvali **Modbus Read - %QX0.0-7**, načítavame **vstupy (zo senzorov)** a cez uzol **Modbus Read - %QX0.7+** načítavame **výstupy (aktuátory)**. My začíname vstupmi od adresy 0, teda %QX0.0. Výstupy máme od adresy %QX1.0. Uzol Modbus číta vždy celý bajt, čo v tomto prípade je v poriadku, keď že máme presne 8 vstupných premenných. Premenné sú typu BOOL (true / false). Modbus Read funguje ako klient a musíme ho napojiť na server. My ho napojíme na server, ktorý sme si nazvali OpenPLC local a nastavili sme príslušnú adresu 127.0.0.1 a port 502. Ďalej si určime, že budeme čítať coils, čo je štandardný príkaz Modbus protokolu

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 6. Factory I/O - Modbus klient 2



Obr. 7. Node-RED ako Modbus klient

(FC1: Read Coil Status). V prípade našich vstupných premen-

Sekcia 1. Mechatronika a Aplikovaná informatika

ných nastavíme adresu na 0, keď že načítavame od %QX0.0 (v prípade %QX1.0 by to bolo teda 8). Kvantitu sme nastavili na 1, nakoľko čítame 1 bajt. Poll rate sme nastavili na 2 sekundy, čo znamená, že hodnota sa načítava každé 2 sekundy.

Podobný postup sme použili aj na **Modbus Read** - %QX0.7+, kde však čítame výstupné premenné od PLC adresy %QX1.0 a naša Modbus adresa je 8.

V PLC programe máme aj 3 hodnoty typu INT, ktoré sú uložené v registroch, ktoré majú iné adresy ako coils (tie sú typu BOOL). Ide napríklad o počet vyrobených výrobkov. Tieto hodnoty načítavame pomocou uzla **Modbus Read Hold-ing**.

B. OPC UA server a klient

Vzhľadom k tomu, že chceme hodnoty z výrobnej linky ponúknuť aj ďalším užívateľom, ktorí môžu disponovať OPC UA klientmi, bolo nutné v Node-REDe realizovať OPC UA server, ktorý by tieto dáta poskytoval.

Ako už bolo povedané, OpenPLC runtime nedokáže fungovať ako OPC UA server, nakoľ ko ide o open-source bezplatný nástroj. Toto je výsadou pokročilejších PLC riešení, ako je napríklad Codesys alebo Siemens TIA Portal. Preto bude OPC UA server vytvorený pomocou Node-RED, do ktorého dáta prichádzajú z OpenPLC runtime cez protokol Modbus.

Na vytváranie servera OPC UA bola využitá knižnica nodered-contrib-opcua 0.2.256.

Pre vytvorenie servera je potrebné využiť uzol OPC UA server, kde sa nastaví port (v našom prípade 53880) a voliteľne aj jeho názov. My využívame predvolený názov. Je možné nastaviť aj možnosti autentifikácie, keď že sme už uvádzali, že komunikačný štandard OPC UA podporuje viacero bezpečnostných profilov. My pre jednoduchosť (a nakoľko fungujeme na localhoste) využívame anonymný prístup.

S vytvorením servera súvisí aj vytvorenie **adresného priestoru** (angl. address space), teda premenných, ktoré na úvod budú prázdne alebo budú mať preddefinovanú hodnotu, pričom neskôr tieto premenné naplníme hodnotami, ktoré nám OpenPLC runtime posiela pomocou Modbus protokolu.

Aby sme naše vstupy a výstupy mali v adresnom priestore prehľadne oddelené, tak sme si vytvorili v adresnom priestore priečinky *FIOOutputs* a *FIOInputs*.

Je treba poznamenať, že jednotlivé direktívy, ktoré posielame uzlu OPC UA servera, sú posielané pomocou uzla typu *Inject*. Treba teda nastaviť, aby sa tieto príkazy vykonali automaticky po spustení Node-RED programu, pričom je logické, že časovanie treba nastaviť tak, aby sa najprv spustili príkazy vytvárajúce priečinky a až potom sa vytvorili premenné. Takto úspešne vytvoríme OPC UA server so želaným adresným priestorom.

My budeme využívať tieto direktívy (podporovaných je však viacero):

- addFolder vytvára priečinok v adresnom priestore
- setFolder nastavuje priečinok v adresnom priestore

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 8. Vytváranie OPC UA servera a jeho adresného priestoru v Node-RED

 addVariable - pridáva premennú do nastaveného priečinka

Direktívy sú viazané na *msg.payload* správ v Node-RED a samotný obsah na *msg.topic* správ. Toto môžeme vidieť aj na obrázku č. 8, kde je ukázané vytváranie priečinka *FIOOutputs*. V *msg.payload* je príkaz na pridanie priečinka a v *msg.topic* vidíme definovaný namespace a názov priečinka.

Pridanie premennej *RetroreflectiveSensor1* by vyzeralo takto, pričom vidíme, že premennej nastavujeme aj dátový typ:

- msg.payload: {"opcuaCommand": "addVariable"}
- msg.topic: ns=1; s=RetroreflectiveSensor1; datatype=Boolean

Na to aby sme vedeli napĺňať premenné dátami, tak sme si potrebovali vytvoriť vlastnú funkciu v jazyku JavaScript, nakoľko v Node-RED nie je nutné využívať len vstavané uzly, ale vieme si vytvárať aj vlastné.

Všimnime si na obrázku č. 7 uzol s názvom Function: Modbus to OPC UA namespace, teda uzol predstavujúci našu vlastnú funkciu. Konkrétne si opíšeme ten uzol, ktorý je spojený s uzlom Modbus Read - %QX0.0-7. Uzol Modbus Read - %QX0.0-7 posiela do uzla Function: Modbus to OPC UA namespace pole 8 hodnôt (typu BOOL). Na obrázku č. 9 vidíme obsah uzla. Pre každú hodnotu získanú z Modbusu sme predpripravili prázdnu premennú (msg0,msg1,msg2,...), kde ukladáme jednotlivé prvky poľa. Do ich payloadu ukladáme samotnú hodnotu (typu BOOL), zaujímavejšou časť ou je však msg.topic. Tu podľa dokumentácie OPC UA klienta nainštalovaného v Node-REDe je potrebné definovať, do ktorej premennej v adresnom priestore OPC UA premenná pôjde. Máme teda definované ID namespacu (skrátene ns), názov premennej a dátový typ, ktorý sa zhoduje so vstupnými premennými získanými z Modbus servera.

Následne je uzol *Function: Modbus to OPC UA namespace* spojený s uzlom **OPC UA client**, ktorý hodnoty uloží do nášho

Sekcia 1. Mechatronika a Aplikovaná informatika

it funct	function node state Cancel Or Properties Image Cancel Or Name Function: Modbus to OPC UA namespace Image On Stop Image On Start On Message On Stop Image On Start On Message On Stop Image On Start On Message On Stop Image Oil Status On Status On Message On Stop Image Oil Status On Message On Stop Image Oil Status Oil Other Status On Stop Image Oil Status Status Oil Status Oil Image Oil Status Status Status Oil Status Oil Status Image Oil Status Status S							
Delete						Cancel		Done
Prope	rties						٥	E
Nam	Fun	ction: Mod	dbus to OPC U	Anamespace				₽.
Ø Se	tun	On S	tart	On Message		On Stop		
	tub.	0.110				onorep		
1	// msg[0] ;	is %0X0.	.0					
2	// msg[1] :	is %QX0.	.1					
3								
4	msg0 = {};							
5	msg0.paylo	ad = msg	g.payload[0];				
6	msg0.topic	= "ns=:	l;s=Retrore	FlectiveSensor	l;data	type=Boole	an"	;
7								
8	msg1 = {};			1.				
10	msg1.topic	au = msg	g.payroau[1]; ElectiveSencer		type-Reels		
10	msgr.copic	- 115	r,s-ketrore	rectivesensor	c,uaca	cype=booie	an	·
12	$m \le \sigma^2 = \{\}$:							
13	msg2.paylo	ad = msg	g.payload[2]	1;				
14	msg2.topic	= "ns=3	l;s=Retrore	FlectiveSensor	3;data1	type=Boole	an"	
15								
16	msg3 = {};							
17	msg3.paylo	ad = msg	g.payload[3];				
18	msg3.topic	= "ns=:	l;s=Retrore	FlectiveSensor	4;data1	type=Boole	an"	
19	maga - ().							
20	msg4 = {};	ad = mer	navload[4]	1.				
22	msg4.topic	= "ns="	1:s=Retrore	D ElectiveSensor	5:data1	vpe=Boole	an"	
23			-,		.,	-,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,		
24	msg5 = {};							
25	msg5.paylo	ad = msg	g.payload[5]	1;				
26	msg5.topic	= "ns=3	l;s=Retrore	FlectiveSensor	5;data1	type=Boole	an"	;
27								
28	msg6 = {};							
29	msg6.paylo	ad = msg	g.payload[5					
30	msg6.topic	= "ns=:	1;s=ketrore	FlectiveSensor	;data	cype=Boole	an"	
31								
33								
34	return Ims	20.msg1	.msg2.msg3.r	nsg4.msg5.msg6	1:			
	· · · · fund	a - 1 (b.s.)						

Obr 9. Funkcia na napĺňanie OPC UA servera hodnotami získanými pomocou Modbus protokolu z OpenPLC - Function: Modbus to OPC UA namespace

OPC UA servera.

Hodnoty z nášho OPC UA servera vieme poskytovať aj iným klientom, ako je klient nainštalovaný v Node-RED. Teoreticky by mohlo ísť o klienta, ktorý nemusí bežať len na počítači, ale aj na smartfóne, tablete, v cloude a podobne. My sme pre otestovanie využili OPC klienta **UaExpert**.

C. Aplikácia v cloude Microsoft Azure

Jedným z hlavných cieľ ov diplomovej práce bola realizácia cloudovej aplikácie, pomocou ktorej by bolo možné monitorovať udalostný výrobný systém - sledovať hodnoty vybraných veličín, tieto dáta vhodne vizualizovať, účelne ich spracovať a prípadne do systému aj zasahovať.

Na začiatku sme si určili, čo chceme, aby sa v našej cloudovej aplikácii nachádzalo. Určili sme si teda **funkcionálne požiadavky**:

- zobrazovanie načítaných hodnôt a akčných tlačidiel v prehľadnom paneli (angl. dashboard)
- komunikácia s Node-RED pomocou protokolu MQTT
- zobrazovanie počtu vyrobených (finálnych) výrobkov odovzdaných do expedície; počtu polovýrobkov (kusov surového materiálu), ktoré prišli do výroby; počtu polovýrobkov / výrobkov, ktoré aktuálne sú na dopravníkov (respektíve v systéme ako takom)
- grafické znázornenie aktuálnej teploty vo výrobnej hale
- jednoduché spracovanie hodnôt aktuálnej teploty vo výrobnej hale za účelom vyvolania alarmu, ak teplota vystúpi nad istú hodnotu
- možnosť núdzového zásahu do systému pozastavenie a spustenie výrobnej linky

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Vytýčené charakteristiky, ktoré sú definované vyššie majú za úlohu opísať to, čo vytváraný aplikačný systém (cloudová aplikácia) bude schopná realizovať.

Počas riešenia práce bolo zistené, že najefektívnejším riešením bude využiť pomerne novú komplexnú službu typu aPaaS (application Platform as a Service) **Azure IoT Central**, ktorá v sebe spája rôzne funkcionality.

Microsoft Azure IoT Central sa používa na pripojenie a správu zariadení vo veľkom rozsahu a poskytuje spoľahlivé údaje pre obchodné štatistiky, vieme ho teda zaradiť medzi ERP (angl. enterprise resource planning) systémy. Zahŕňa v sebe viacero služieb typu PaaS, čím umožňuje vytvoriť jednoducho konfigurovateľné, komplexné a bezpečné IoT riešenia. Webové používateľ ské rozhranie umožňuje rýchlo prepojiť zariadenia, monitorovať stav zariadení, vytvárať pravidlá a spravovať milióny zariadení a ich údaje počas ich životného cyklu.

Štruktúra nášho zariadenia s názvom *conveyor-system1* je nasledovná, pričom prvý údaj je **systémový názov** a druhý údaj je **typ schopnosti**: turnOn (*command*), turnOff (*command*), produced (*telemetry*), entered (*telemetry*), on_line (*telemetry*), temperature (*telemetry*), location (*property*).

Pripojenie zariadenia je riešené pomocou autentifikácie typu SAS (angl. skratka Shared access signature). Náš model zariadenia v cloude prepájame s Node-RED pomocou údajov ID scope, Device ID a Primary key.

Následne si vieme prezerať dáta, ktoré prijíma náš cloud. Všimnime si obrázok č. 10, kde vidíme 3 hodnoty, ktoré náš cloud prijíma - Totally produced, Entered on line a Currently on line. Totally produced znamená, koľko výrobkov bolo vyrobených a odovzdaných do expedície. Entered on line je počet polovýrobkov (kusov surového materiálu) položených na pás. Currently on line je počet aktuálnych polovýrobkov a výrobkov na páse. Vidíme tiež graf, ktorý nám zobrazuje teplotu ovzdušia v danej továrni. Nakoľko naša výrobná linka nie je reálna, tak teplotu len simulujeme pomocou generovania náhodných hodnôt v Node-REDe v rozmedzí 18 až 25. Vieme z Node-RED tiež vyslať signál počas behu aplikácie, že teplota ovzdušia je 100°C. V tomto prípade máme realizovanú na strane Azure IoT Centralu udalosť (alarm), ktorá zabezpečí zaslanie e-mailu. Týmto demonštrujeme základnú formu spracovania a vyhodnotenia dát, na základe ktorého je vykonaná udalosť. Udalosti sa v Azure IoT Central môžu vyberať z preddefinovaných typov alebo si je možné naprogramovať vlastnú funkciu pomocou Azure Functions (serverless architektúra).

Ako prebieha v Node-RED posielanie dát, stručne opíšeme práve na príklade generovania hodnôt teploty (obrázok č. 11). Začíname použitím uzla *timestamp*, ktorý nám zabezpečuje cyklický beh daného programového toku. Ďalej používame *random* uzol (na obrázku s názvom *Temperature*), ktorý generuje celé náhodné čísla od 18 po 25. Tento uzol vchádza do vlastnej funkcie *temperature*, kde do *msg.payload* našej správy vkladáme identifikátor, ktorý je zhodný s identifikátorom premennej *temperature* v cloude na danom modeli (dvojčati) zariadenia. Máme tu pripravený aj uzol *inject 100*,



Obr. 10. Dashboard v Azure IoT Central



Obr. 11. Tok posielanie dát v Node-RED (teplota ovzdušia)

ktorého manuálnym zatlačením vieme zaslať hodnotu teploty 100°C, aby sme mohli simulovať požiar vo výrobnej hale. Celá vetva nakoniec vchádza do uzla *Azure IoT Central*, kde je definované ID zariadenia na cloude a kľúč k prístupu. Okrem tohto je tu definovaný a j k omunikačný p rotokol, v našom prípade je to MQTT. Dá sa komunikovať aj cez AMQP alebo HTTPS. Používame knižnicu *azure-iot-central 1.6.0*.

Pomocou cloudovej aplikácie vieme do nášho virtuálneho výrobného systému aj **zasahovať**. Toto sme demonštrovali pomocou implementácie možnosti núdzovo **pozastaviť** a **spustiť linku**. Na obrázku č.10 vidíme v pravom hornom rohu tlačidlá, ktoré toto zabezpečujú.

V. ZÁVER

V príspevku bola predstavená prípadová štúdia, ktorá demonštruje možnosti riadenia virtuálneho udalostného systému pomocou softPLC (PLC runtime bežiaci na bežnom PC). Virtuálny udalostný systém je modelovaný vo Factory IO. Realizovaná je tiež komunikácia od senzorov a aktuátorov až po cloud.

POĎAKOVANIE

Chceli by sme sa poďakovať Erikovi Kučerovi za pomoc pri implementácii cloudovej aplikácie.

Referencie

- Operational technology. [Online]. Available: https://www.gartner.com/en/ information-technology/glossary/operational-technology-ot
- [2] M. Pajpach, O. Haffner, E. Kučera, and P. Drahoš, "Low-cost education kit for teaching basic skills for industry 4.0 using deep-learning in quality control tasks," *Electronics*, vol. 11, no. 2, 2022. doi: 10.3390/electronics11020230. [Online]. Available: https: //www.mdpi.com/2079-9292/11/2/230
- [3] Cloud. [Online]. Available: https://www.investopedia.com/terms/c/ cloud-computing.asp

Prepojenie DC motora so simulačným prostredím MATLAB

Jaroslav Hajda

Ústav automobilovej mechatroniky FEI STU v Bratislave

xhajda@stuba.sk

Abstrakt – Cieľom príspevku je prepojiť DC motor s enkodérom s prostredím MATLAB. S týmto zámerom sme vytvorili experimentálnu zostavu pozostávajúcu vývojovej dosky, motora a ďalších súčiastok potrebných na napájanie alebo riadenie. Túto sústavu sme následne prepojili pomocou nadstavby Simulink a ďalších doplnkov s MATLABom. Prepojenie sme realizovali cez simuláciu v externom móde (HIL). Po prepojení sme pomocou dostupných nástrojov vykonali niekoľko experimentov, ktorých výsledky sme spracovali. Úspešne sme tak prepojili a plne ovládali DC motor z prostredia MATLAB.

1 Úvod

MATLAB a jeho nadstavba Simulink ponúkajú prostredie na modelovanie a simuláciu dynamických systémov. Jednosmerný motor je jeden z najbežnejších dynamických systémov využívaných v praxi. Na jeho modelovanie ponúka MATLAB viacero spôsobov, či už pomocou matematického modelu alebo pomocou rozšírenia Simscape. Práca v teoretickej rovine je však len časťou experimentálneho procesu a poznatky v nej nadobudnuté je vhodné využiť v praktickej rovine. Zámerom nášho príspevku je preskúmať možnosť prepojenia reálneho DC motora s prostredím MATLAB. Prepojením chceme dosiahnuť, aby bolo motor možné plne ovládať z prostredia MATLAB a Simulink. Takto by sme vedeli využiť nástroje, ktoré MATLAB ponúka, na reálnom zariadení. Naším zámerom je toto prepojenie dosiahnuť využitím základných funkcií vývojovej dosky, bez použitia akéhokoľvek špeciálneho hardvérového vybavenia. Takto chceme dosiahnuť to, aby riešenie bolo možné využiť s použitím akejkoľvek, dostatočne výkonnej, vývojovej dosky podporovanej prostredím Simulink.

2 Hardvérové vybavenie

Pracujeme s doskou z rodiny STM 32 Nucleo-64, s modelom F411RE [1], ktorý je podporovaný doplnkom pre Simulink. Dosku sme si vybrali pre jej vysoký výkon, ktorý sme predpokladali, že bude potrebný pre prácu s krátkymi intervalmi enkodéra a externým módom v Simulinku.

Parametre Nucleo F411RE:

- 100 MHz maximálna frekvencia CPU ARM®32-bit Cortex®-M4
- 512 KB Flash
- 128 KB SRAM

prevádzkové napätie 3,3V

Keďže je jasné, že DC motor nemôžeme napojiť priamo na dosku, na ovládanie napätia na motore využívame modul ovládača DC motora. Samotný modul je postavený na základe vysoko-prúdového BTS 7960 [2] polmostíka (H-Bridge). Ten dokáže pracovať s prúdmi 43A a s napájacím napätím od 5,5 do 27,5V, čo znamená že je vhodný pre DC motory s nominálnym napätím 12 a 24V. Má ochranu proti vysokému a nízkemu prevádzkovému napätiu, proti vysokým prúdom a proti prehrievaniu. Modul je ovládaný PWM signálom z Nuclea a priamoúmerne nám poskytuje rovnaké plnenie pre napätie motora. Napríklad ak má PWM signál z Nuclea striedu 50% a modul je napájaný 12V, na výstupe pre motor bude napätie 6V.

Na napájanie používame batériu s kapacitou 1800 mAh a nominálnym napätím 7,4V. Keďže to je nielen málo pre napájanie modulu ovládača, ale napätie batérie sa mení podľa nabitia, využívame na zosilnenie a stabilizáciu napätia menič. Konkrétne modul DC-DC buck/boost meniča LTC3780 [3]. Na napájanie mikrokontrolérov zabudovaných v moduloch a enkodéra motora využívame jednoduchý buck DC-DC menič HW-613. Parametre motora uvádzame v Tab.1

Tab. 1. Parametre motora [5]

nominálne napätie	12	[V]
prevodový pomer	43,8	[-]
otáčky bez zaťaženia (hriadeľ prevodovky)	251	[1/min]
otáčky bez zaťaženia (hriadeľ motora)	11000	[1/min]
krútiaci moment	1,8	[N*m]
prúd bez zaťaženia	350	[mA]
špičkový prúd	7	[A]
izolačný odpor motora	20	[MΩ]
rozlíšenia enkodéra (hriadeľ prevodovky)	≈700	[CPR]
rozlíšenia enkodéra (hriadeľ motora)	16	[CPR]
napájacie napätie enkodéra	5	[V]
Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

3 Prepojenie Simulink-Nucleo

Pre prácu na reálnom zariadení je potrebné nainštalovať nasledujúce doplnky:

- Simulink Coder Support Package for STMicroelectronics Nucleo Boards
- MATLAB Coder
- Simulink Coder

Pre prepojenie Nuclea so Simulinkom potrebujeme nakonfigurovať dosku v okne Model Configuration Parameters. Po inštalácii doplnku pre Nucleo, Hardware board v karte Hardware Implementation ponúka na výber niekoľko z podporovaných dosiek. My si vyberieme dosku STM32 Nucleo F411RE. Pre fungovanie Externého módu je potrebné pod Target hardware resources, nastaviť na karte External mode správny COM port pre našu dosku. Ten zistíme po pripojení dosky k počítaču vo Windows v Správcovi zariadení, respektíve v Device Manager. Aby sme videli výstup pre externý mód v Diagnostic Viewer pre pre riešenie ďalších prípadných problémov, zapneme si aj funkciu Verbose. V prostredí Simulink nastavíme pre schémy typ simulácie External a čas zastavenia nekonečno.

Inštaláciou doplnku pre Nucleo získame aj nové bloky v knižnici, ktoré budeme využívať v schémach. Bloky nájdeme v knižnici pod kartou Simulink Coder Support Package for STMicroelectronics Nucleo Boards. Pre naše potreby využijeme bloky Digital Read, Digital Write a PWM Output. Pre blok Digital Read je potrebné nastaviť pin a čas vzorkovania, pre Digital Write stačí nastaviť pin a pre PWM Output nastavíme pin, frekvenciu PWM signálu a počiatočnú striedu. Každý z týchto blokov ponúka možnosť zobraziť mapu pinov. Každý z bežných I/O pinov na doske je možné použiť pre PWM signál.

Pri kompilácii schém a generovaní kódu v externom móde sa niekedy vyskytli chyby. Tie nastali napríklad vtedy, keď bol v MATLABe nastavený iný adresár než lokácia spúšťanej Simulink schémy. Chyba sa v tomto prípade neopravila po presunutí do správneho adresára. Zistili sme, že najlepším riešením, ak vieme že chyba nie je v schéme alebo jej konfigurácii, je vymazať priečinok *slprj* a prípadne aj priečinok s vygenerovaným kódom (*názov schémy*)_ert_rtw. Priečinok *slprj* obsahuje cache kód pre Simulink, ktorý v bežnom prípade urýchľuje spúšťanie, ale v prípade chyby je lepšie vygenerovať nový.

4 Experiment

Po úspešnom prepojení dosky s prostredím MATLAB môžeme začať s konkrétnymi experimentmi na motore. Ak

chceme motor akýmkoľvek spôsobom riadiť, musíme najprv vedieť spracovať dáta z enkodéra. My sme sa rozhodli výstup motora sledovať v jednotkách RPM.

4.1 Spracovanie dát enkodéra podľa počtu pulzov

Jednou z možností spracovania dát z enkodéra je počítanie impulzov. Podľa tohto princípu sme zostavili schému na Obr. 1. Základom tejto schémy je počítadlo, ktoré je nastavené tak, aby pripočítalo 1 vždy, keď zaznamená rastúcu hranu. Počítadlo vyšle na výstup aktuálnu hodnotu pri každom pripočítaní. To je však problém, pretože my chceme súčet impulzov za danú periódu. Riešime to cez triggered subsystem, ktorý pomocou minimálneho oneskorenia spúšťame krátko predtým než sa počítadlo resetuje. Tento subsystém len prepúšťa na výstup to, čo ma na vstupe vtedy, keď dostane impulz. Počítadlo resetujeme pomocou digitálnych hodín, ktoré vyšlú impulz každú desatinu sekundy. Oneskorenie je nastavené na desatinu milisekundy. Na výstupe subsystému teda dostávame frekvenciu v počte pulzov za desatinu sekundy. Pri prepočte na RPM musíme tieto vynásobiť 600, aby sme dostali počet pulzov za minútu a vydeliť, 700 aby sme získali počet otáčok za minútu. Kde 700 je počet impulzov na jednu otáčku hriadeľa prevodovky.

Táto metóda je menej presná ako počítanie periódy pulzu, no oveľa jednoduchšia a lepšie implementovateľná do reálneho systému. Už z princípu tejto metódy si môžeme všimnúť, že vyššiu presnosť dosiahneme len menším vzorkovaním. Na druhú stranu ak chceme optimalizovať riadenie, vyžaduje si to aj dostatočne rýchlu odozvu systému. Musíme hľadať kompromis medzi presnosťou a hustotou vzorkovania.

4.2 Riadiaca schéma

Pre prvotné experimenty sme sa rozhodli vytvoriť jednoduchú riadiacu schému. Schému, ktorú vidíme na Obr. 3. využijeme na zoznámenie sa s prácou v externom móde a zároveň s jej pomocou môžeme namerať prevodovú a prechodovú charakteristiku experimentálnej sústavy.

Konštanta *Enable* posiela logickú jednotku na piny 6 a 7, ktoré sú spojené s enable pinmi na H-bridge. Konštanta *Plnenie*, je spojená s premennou *strieda* a upravuje striedu PWM signálu na pine 5. Dáta enkodéra z pinu 9 spracovávame pomocou schémy z kapitoly 4.1, ktorú sme umiestnili do subsystému.



Obr. 1. Schéma pre spracovanie dát enkodéra podľa počtu pulzov

Cez bloky To Workspace ukladáme hodnoty otáčok do premennej *rpm2* a čas simulácie do premennej *t*. Aby sa dáta zaznamenávali, musíme dať signály logovať. To v prostredí Simulink vieme spraviť napríklad cez kontextové menu po kliknutí na konkrétny signál, kde vieme vybrať Log Selected Signals alebo Stop Logging Selected Signals. Signály zaznamenávame s periódou vzorkovania 0.01s, aby sme zbytočne nezať ažovali schému. To nám úplne postačuje, keďže informácia o otáčkach sa aktualizuje každých 0.1s. Čas vzorkovania pre čas sme nastavili priamo v bloku Digital Clock, ktorý má na výstupe čas simulácie. Pre otáčky sme čas vzorkovania nastavili ako Sample Time parameter Output bloku subsystému na spracovanie dát enkodéra. To si vyžiadalo Transfer Rate blok pred Output blokom, kvôli konfliktu v čase vzorkovania.

4.2.1 Ovládanie experimentu v externom móde z MATLAB skriptu

Ako už naznačuje schéma na Obr. 2., rozhodli sme sa schému nespúšťať priamo z prostredia Simulink, ale zo skriptu v MATLABe. Na rozdiel od normálnych schém však nie je možné pre schému v externom móde spustiť cez príkaz *sim*. Na ovládanie schémy používame príkaz *set_param*. Ten dokáže priamo z prostredia MATLAB meniť parametre schém v Simulinku. V našich skriptoch budeme využívať túto funkciu na zadávanie príkazov *SimulationCommand* podľa vzoru:

set_param('názovschémy', 'SimulationCommand', 'príkaz')

Príkazy, ktoré v skriptoch použijeme sú *Start, Update, Stop* a *WriteDataLogs*. Start a Stop slúžia na začatie a ukončenie behu schémy a externého módu. Príkaz Update nám dovoľuje meniť premenné používané schémou za jej behu. Posledný príkaz, ktorý využívame, WriteDataLogs zapisuje najaktuálnejšie logované hodnoty signálov do Workspace prostredia MATLAB.

Ak však využijeme len tieto príkazy, kód bude funkčný len vtedy, ak bude ovládaná schéma otvorená v prostredí Simulink. Aby sme sa tomuto problému vyhli využili sme funkciu *load_system*, ktorá načíta vybranú schému bez toho, aby sa otvorilo okno Simulink.

Ďalším problémom, na ktorý sme narazili je to, že potrebujeme ovládať čas, kedy sa jednotlivé príkazy v skripte spúšťajú. Ak by sme totiž položili príkazy rovno za sebou, celý skript by zbehol tak rýchlo ako to počítač dovolí a ak využívame na ukončenie experimentu príkaz Stop, chod experimentu sa okamžite po štarte zastaví. Využívame preto funkciu *pause*, ktorá umožňuje chod skriptu pozastaviť a experiment po túto dobu beží na nastavených hodnotách. Druhou možnosťou je pri každej zmene premennej experiment vypnúť a znova spustiť a nastaviť konečný čas v konfigurácii schémy. Táto možnosť je však oveľa viac časovo náročná a na naše účely produkovala rovnaké výsledky, alebo nebola použiteľná, kvôli tomu, že parametre je potrebné meniť za behu.

4.2.2 Prevodová charakteristika experimentálnej sústavy

Prevodovú charakteristiku sme robili pomocou jedného merania v jedenástich bodoch so striedou od 0 do 100%, čo predstavuje napätie 0 až 12V.



Obr. 2. Prevodová charakteristika DC motora



Obr. 3. Riadiaca schéma pre externý mód

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Na grafe na Obr. 2 vidíme, že prevodová charakteristika je výrazne lineárna okrem jemnej nelinearity na začiatku medzi 0 až 2V. To zodpovedá tomu čo vieme o kefových DC motoroch. Na grafe sme vyznačili pracovný bod, v ktorom budeme vypracovávať prechodovú charakteristiku pre motor.

Zaujímavý je tiež celý priebeh experimentu na Obr. 4., kde vidíme ako sa rýchlosť motora v čase graduálne menila so zvyšujúcou sa zadávanou striedou.



Obr. 4. Priebeh experimentu na meranie prevodovej char.

4.2.3 Prechodová charakteristika experimentálnej sústavy

Počas práce so schémami v externom móde sme si uvedomili potrebu overiť správnosť údajov zo Simulinku. Keďže údaje enkodéra je potrebné čítať s vysokou frekvenciou, vznikol predpoklad, že Nucleo nemusí stíhať spracovať kód generovaný externým módom v rámci jedného cyklu. To by sa prejavilo hlavne pri prechodovej charakteristike. Rozhodli sme sa preto pre referenciu vytvoriť čo najjednoduchší program v C++ v prostredí Visual Studio Code.



Obr. 5. Prechodová charakteristika DC motora

Merania boli vykonané pre jednotkový skok z napätia 6V na 7V. Pri meraní prechodovej charakteristiky sme vykonali tri merania pre nami vytvorený program v C++ a tri merania v prostredí MATLAB=Simulink. Na Obr. 5. vidíme, že priebehy sa približne zhodujú. Môžeme teda konštatovať, že program vygenerovaný externým módom pracuje rovnako dobre ako náš jednoduchší program. Keďže bolo ťažké identifikovať čo je dopravné oneskorenie sústavy a čo je spôsobené skriptom rozhodli sme sa dopravné oneskorenie pri tvorbe prechodovej charakteristiky neuvažovať.

4.2.4 Identifikácia

Tri priebehy namerané v externom móde sme spriemerovali a na základe vzniknutej krivky sme aproximovali priebeh ako systém prvého rádu. Výsledkom je prenosová funkcia zapísaná v rovnici 1. Z rovnice môžeme vidieť, že ustálenú hodnotu odozvy na jednotkový skok napätia sme určili na 21.25 RPM.

$$G(s) = \frac{21,25}{0.17s+1} \tag{1}$$

Odozvu aproximácie na jednotkový skok napätia v porovnaní so spriemerovanými meraniami z externého módu vidíme na Obr. 6. Hoci je dynamika motoru ovplyvnená prevodovkou, vidíme, že aproximácia ako systém prvého rádu je postačujúca.



Obr. 6. Prechodová charakteristika DC motora a jej aproximácia

Na Obr. 7 porovnávame priebeh aproximácie pri rôznych hodnotách napätia v prevodovej charakteristike. Vidíme, že v oblasti pracovného bodu sa odozva aproximácie a zariadenia líši najmenej. Čím viac sa vzďaľujeme od pracovného bodu tým viac sa hodnoty líšia. Najväčšie rozdiely pozorujeme pri nelinearite zariadenia pri nízkych hodnotách napätia. Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 7. Prevodová charakteristika DC motora a jej aproximácia

5 Záver

Cieľom príspevku bolo vytvoriť funkčné prepojenie medzi prostredím MATLAB a DC motorom. Začali sme výberom a zostavením hardvéru experimentálnej zostavy. Pri výbere hardvéru sme hľadeli nie len na parametre a výkon, ale aj na cenovú dostupnosť. Pri výbere dosky sme si zároveň všímali, či je doska podporovaná doplnkami MATLAB a či ponúka aspoň základné ovládacie prvky v knižnici Simulink. Ak doska nie je podporovaná doplnkami akékoľvek ďalšie prepojenie s MATLABom je natoľko komplikované, že sa nedá považovať za vhodné. Hardvérovú zostavu sme prepojili pomocou dostupnej technickej dokumentácie jednotlivých súčastí.

Softvérová časť prepojenia zahŕňa inštaláciu potrebných doplnkov, správne nastavenie solvera pre každú schému a ďalšie úpravy schém spojené so simuláciou v externom móde. Najväčšie problémy spôsobovalo zaznamenávanie údajov a ich ukladanie. To bolo potrebné riešiť cez nastavenie logovania jednotlivých signálov a vhodným umiestnením blokov Rate Transfer. Blokmi Rate Transfer sme riešili prípadné konflikty medzi periódami vzorkovania údajov. Problémy nastávali tiež pri cache systému MATLAB, ktorý ak uchoval zlé nastavenia, pri ďalšom spustení, hoci už so správnymi nastaveniami, spôsoboval error. Ak sme mali podozrenie, že error je spôsobovaný len cache, riešili sme to manuálnym vymazaním príslušného priečinka.

Po úspešnom prepojení sme začali s experimentovaním. Ako prvá sa ukázala nutnosť spracovania dát z enkodéra. Keďže pracujeme s mikropočítačom a reálnym zariadením, snažili sme sa o čo najjednoduchšiu, no efektívnu schému. Vytvorená schéma plne postačovala na experimenty, ktoré sme na zariadení vykonávali. Keďže prostredie Simulink ponúka len obmedzené prostriedky na spracovanie údajov, rozhodli sme sa so schémou pracovať z prostredia MATLAB skriptu a Command Window. Toto spojenie medzi skriptom, Simulink schémou a reálnym zariadením, nám umožňuje interaktívne pracovať so zariadením a sledovať ako menené parametre ovplyvňujú jeho chod. Ešte za chodu experimentu spracovávať dáta schémy a sledovať výsledky experimentu nie len Sekcia 1. Mechatronika a Aplikovaná informatika

empiricky, ale priamo sledovať aj namerané dáta. Na meranie a zápis údajov pre experimenty sme si preto vytvorili vlastné skripty, ktorých výstupmi sú nameraná prevodová a prechodová charakteristika.

Prevodovú charakteristiku sme namerali bez výrazných problémov. Pri prechodovej charakteristike mohli nastať problémy ak by bol nameraným otáčkam priradený nesprávny čas. Preto sme pre referenciu vytvorili druhý program v C++, v prostredí Visual Studio Code. Po porovnaní výsledkov sme zistili, že program vygenerovaný MATLABom funguje správne, respektíve rovnako dobre ako nami naprogramovaný program v C++. Podľa nameranej prechodovej charakteristiky sme systém identifikovali ako systém prvého rádu a aproximáciou získali prenosovú funkciu. Odozvu aproximácie a zariadenia sme následne porovnali a zistili, že systém bol v rámci možností aproximovaný správne. Nami vykonané experimenty môžu poslúžiť ako základ pre ďalšie riadenie motora. MATLAB v tomto ohľade ponúka mnoho nástrojov. Naším zámerom je ďalej pokračovať s reguláciou rýchlosti pomocou PID regulátora s automatickým ladením.

Literatúra

- [1] *NUCLEO-F411RE*. [Online]. Mbed. 20. 4. 2022. Available: <u>https://os.mbed.com/platforms/st-nucleo-f411re/</u>
- [2] Data Sheet BTS 7960. [Online]. Infineon Technologies. 20. 4. 2022. Available: <u>https://www.infineon.com/dgdl/BTS7960_Datasheet.pdf?</u> <u>folderId=db3a304412b407950112b408e8c90004&fileId=</u> <u>db3a304412b407950112b43945006d5d</u>
- [3] Data Sheet LTC3780. [Online]. ANALOG DEVICES. 20. 4. 2022. Available: <u>https://www.analog.com/media/en/technical-</u> documentation/data-sheets/LTC3780.pdf
- [4] HW-613 / MP2315 Synchronous buck DC-DC converter for 1.8, 2.5, 3.3, 5, 9 or 12 V. [Online]. PCHEY. 20. 4.
 2022. Available: <u>https://www.pchey.com/2021/12/hw-613-mp2315-synchronous-buck-dc-dc.html</u>
- [5] Gear Motor w/Encoder, model No.GB37Y3530-12V-251R. [Online]. DFROBOT. 20. 4. 2022. Available: <u>https://wiki.dfrobot.com/12V DC Motor 251rpm w Encoder_SKU_FIT0186_#target_0</u>

Sekcia 2

Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Technológie prípravy OLED (organických elektroluminiscenčných diód) s emisným dopantom

Ondrej Kokavec, Michal Micjan

Ústav elektroniky a fotoniky (Oddelenie organickej elektroniky a lekárskej elektroniky), Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave

xkokaveco@stuba.sk

Abstrakt – Táto práca sa zaoberá technológiami prípravy organických elektroluminiscenčných diód za pomoci emisných dopantov, na báze irídia, pre dosiahnutie efektívnejších výsledkov a získanie nového spôsobu realizácie rôznych farieb. Pre overenie predstavenej teórie bolo skonštruované funkčné zariadenie vyžarujúce červenú farbu, spĺňajúce všetky elektrické aj optické predpoklady, slúžiace ako odrazový mostík pre ďalšie pokračovanie výskumu v oblasti organických elektroluminiscenčných diód.

1 Úvod

V dnešnom modernom svete je priam nemožné sa vyhnúť elektronickým zariadeniam, ako sú televízie, monitory, obrazovky notebookov, tabletov, smartfónov, inteligentných hodiniek a pod. V priebehu rokov sa objavili rôzne princípy, na ktorých môžu obrazovky pracovať, počínajúc CRT (*z angl.* Cathode Ray Tube), LCD (*z angl.* Liquid Crystal Display), LED (*z angl.* Light Emitting Diode) aj s aktuálne novou technológiou µLED, PDP (*z angl.* Plasma Display Panel) atď. Väčšina spomenutých princípov pracuje na báze anorganickej elektroniky. Tu však, ako aj v iných oblastiach, vstupuje práve organická elektronika, ktorá dokáže svojimi vlastnosťami klasickým riešeniam konkurovať, no v niektorých prípadoch priniesť aj lepšie alebo úplne nové vlastnosti.

Keďže OLED je aktívny zdroj žiarenia, každý pixel generuje svoje vlastné svetlo a tým pádom displej spotrebuje (vo väčšine prípadov) menej energie. Taktiež vďaka tomu možno vytvoriť dokonale čiernu farbu úplným zhasnutím pixelu. Štruktúry sú jednoduchšie, odolnejšie voči mechanickému napätiu a majú rýchlejšiu odozvu ako napríklad LCD displeje. Vyznačujú sa vysokým rozlíšením, ostrosťou kvalitným vykreslením farieb a kontrastom, a širokým pozorovacím uhlom. Okrem toho ale ponúkajú aj nové a lákavé vlastnosti, ako ohybnosť a priehľadnosť a výsledné zariadenie môže mať až neuveriteľne tenkú hrúbku oproti klasickým technológiám. Majú však aj svoje špecifické nevýhody a to je ich vysoká citlivosť voči okolitému prostrediu, relatívne kratšiu životnosť a zatiaľ pomerne vysokú cenu [1].

Prvá OLED bola skonštruovaná v roku 1987 v spoločnosti Eastman Kodak Dr. Ching Wan Tangom a Stevenom Van Slyckom. Princípy teda boli položené pomerne dávno a OLED technológie sú v neustálom vývoji už vyše 30 rokov [2]. Či už komerčné spoločnosti alebo vedci z celého sveta, neustále usilovne pracujú na objavovaní a vylepšovaní všetkých vlastností, ktoré nám táto prelomová technológia môže ponúknuť a zároveň ju tak postupne sprístupniť širokej verejnosti.

2 Problematika a jej riešenie

Štruktúra OLED prvku je tvorená tenkými vrstvami organických materiálov umiestnenými medzi dvoma elektródami nanesenými na priehľadnom substráte. Týchto vrstiev môže byť väčšie množstvo, pričom každá má svoje špecifické vlastnosti a úlohy. Väčšina z nich slúži na transportovanie voľných nosičov náboja (dier zo strany anódy a elektrónov zo strany katódy), ktoré sa snažia čo najefektívnejšie dopraviť do strednej vrstvy, označovanej ako emisná (skr. EML). Tu dochádza k elektroluminiscencii, čo je žiarivá rekombinácia elektrónov a dier v materiáli s polovodičovými vlastnosťami. Vplyvom elektrického poľa na prvok dochádza ku injekcií a následnej rekombinácií náboja, čo má za dôsledok emisiu elektromagnetického žiarenia, najčastejšie vo viditeľnej časti spektra [1][3]. Príkladová štruktúra takéhoto viacvrstvového prvku, aj so znázorneným princípom činnosti, je zobrazená na Obr. 1.



Obr. 1. Energetický pásmový diagram viacvrstvovej OLED.

Vlastnosti emitovaného žiarenia závisia od emisnej vrstvy, jej zloženia a použitých materiálov. Z hľadiska emisných materiálov prebieha samostatný rozsiahly výskum, z ktorého v priebehu rokov vzišli rôzne zaujímavé riešenia. Podľa ich spoločných princípov činnosti, vlastností a času Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

aplikovania, došlo k ich rozdeleniu na 3 generácie, pričom v tejto práci sú rozobrané dve: 2. generácia, ktorej sa táto práca týka a 1. generácia, s ktorou ju porovnávame. Viac o 1. a 2. generácií je spomenuté v nasledujúcej podkapitole.

2.1 Teória

Pri vyššie spomenutej rekombinácii voľných nosičov náboja je systém vybudený na určitú energiu. Prirodzene sa snaží vrátiť späť na nejakú z jeho stabilných energetických hladín a pri tom uvoľňuje túto energiu von, napríklad žiarivými prechodmi vo forme generácie fotónov. Pri tomto procese sú generované singletové a tripletové excitóny (viazané elektrón-dierové páry) v pomere 25 ku 75 %. Pre čo najlepší výsledok a dosiahnutie čo najvyššej kvantovej účinnosti je potrebné plnohodnotne využiť všetky tieto stavy a práve to je hlavným aspektom nasledujúceho vývoja [2].

kontext, 1. generácia emisných materiálov, Pre označovaná ako fluorescenčná, pracuje s materiálmi obsahujúce organické farbivá. Je schopná využiť len prechody singletových excitónov do základného energetického stavu, čiže jej vnútorná kvantová účinnosť (z angl. Internal Quantum Efficiency) IQE = 25 %. Najvyššia prirodzená vonkajšia kvantová účinnosť (z angl. External Quantum Efficiency) EQE zariadenia tejto generácie je 5 %. Sem patrí aj už spomenutá prvá OLED z roku 1987, ktorá vyžarovala zelenú farbu (vlnovej dĺžky 550 nm) a využívala na to materiál Tris-(8hydroxyquinoline)aluminum (Alq₃) [2].

Nahradila iu 2. generácia, označovaná ako fosforescenčná, ktorá je ústrednou témou tejto práce. Aby sa dokázalo využiť zostávajúcich 75 % prechodov (uchovaných v excitónoch v tripletovom stave) boli skúmané a vyvinuté emisné materiály na báze ťažkých kovov ako je irídium (Ir) platina alebo (Pt). Energia z excitónov najnižšieho singletového stavu S1 dokáže prekonvertovať do najnižšieho tripletového stavu T1 a fosforescenciou vykonať prechod do základného stavu S₀. Tento princíp umožňuje dosiahnuť IQE až 100 %, čo je rozhodujúce pre to, aby bolo výsledné EOE zariadenia vysoké (to prakticky dosahuje niečo od 20 cez 30 %). Táto technológia sa objavila okolo roku 2000 a bola (aj stále je) široko využívaná [2]. Vyššie popísaný princíp fungovania emisných materiálov 2. generácie je znázornený na Obr. 2.



Obr. 2. Princíp fungovania 2. generácie emisných materiálov.

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

2.2 Experiment

Sklenené substráty, od výroby pokryté tenkou vrstvou indiumcín-oxidu (*z angl.* Indium Tin Oxide) ITO, boli narezané na rozmery 18x18 mm.

Následne prešli ultrazvukovou čističkou v niekoľkých fázach, pričom každá z nich trvala 15 minút pri teplote 39 °C. V prvom kroku bol použitý 20 % vodný roztok etanolamínu, potom izopropylalkohol a nakoniec deionizovaná voda. Medzi každým krokom boli substráty prepláchnuté deionizovanou vodou a na úplnom konci čistenia osušené dusíkom. Toto čistenie zabezpečilo odstránenie nečistôt z procesu rezania a iných, z povrchu substrátov.

Na tvarovanie anódy (ITO) bol využitý fotolitografický proces. Použil sa pozitívny fotorezist S1813, ktorý bol po kvapnutí na substrát odstredený s programom v dvoch fázach: 500 RPM (ot/min) po dobu 3 sekúnd s akceleráciou 500 slúžiaci na homogénne rozliatie fotorezistu po RPM/s, substráte a 3000 RPM po dobu 30 sekúnd s akceleráciou 500 RPM/s, slúžiaci pre následné vytvorenie vrstvy a odstredenie nadbytočného materiálu. Substrát bol žíhaný 1 minútu na piecke pri 110 °C za účelom odparenia zvyškov rozpúšťadiel. Pripravené vrstvy boli osvecované (UV žiarením) cez požadovanú tieniacu fotolitografickú masku 20 sekúnd, po čom sa dala osvetlená časť rozpustiť do vývojky MF-320, zriedenej s deionizovanou vodou v pomere 1:2, na 1 minútu. Tento proces zaručil, že pri ďalšom kroku zostanú miesta pokryté fotorezistom chránené pred leptaním. Na konci boli substráty jemne opláchnuté prúdom deionizovanej vody a osušené dusíkom.

Mokré leptanie pre permanentné dosiahnutie požadovaného tvaru, v 35 % kyseline chlorovodíkovej, trvalo približne 7 minút. Po procese sa kvôli zmytiu kyseliny substráty namočili do deionizovanej vody, odkiaľ putovali do acetónu pre odstránenie zvyšného fotorezistu, po čom ich bolo opäť potrebné opláchnuť deionizovanou vodou a osušiť dusíkom.

Pred samotným vákuovým naparovaním sa povrchy substrátov ešte raz upravili [4]. Najprv boli vystavené UV-O₃ (ozón) na 10 minút, po čom nasledovalo ultrazvukové čistenie (odmastenie) v acetóne a izopropylalkohole (oba kroky po dobu 10 minút pri teplote 39 °C).

Vákuové naparovanie prebiehalo pri tlaku rádovo 10⁻⁶ mBar a nižšom (bez prerušenia vákua) za použitia pripravených masiek pre jednotlivé vrstvy. Všetky použité organické materiály boli zakúpené od dodávateľa *Ossila*. Štruktúra vytváraného prvku bola [4] ITO/ TAPC (20 nm)/ TCTA (10 nm)/ TCTA:B3PYMPM:Ir(dmpq)₂(acac) (30 nm)/ B3PYMPM (45 nm)/ Al (100 nm).

Kde TAPC je materiál 1,1-Bis[(di-4tolylamino)phenyl]cyclohexane, TCTA je Tris(4-carbazoyl-9ylphenyl)amine, B3PYMPM je 4,6-Bis(3,5-di(pyridin-3yl)phenyl)-2-methylpyrimidine, Ir(dmpq)₂(acac) je Bis(2-(3,5dimethylphenyl)quinolineC2,N')(acetylacetonato)iridium(I II) [5] a Al je hliníková katóda. Táto štruktúra je pre lepšiu predstavu graficky znázornená na Obr. 3. Pri experimente sa ju a jej parametre podarilo okrem drobných zanedbateľných defektov dodržať.

Emisná vrstva bola vytvorená za princípu súbežného vákuového naparovania materiálov TCTA a B3PYMPM,

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

slúžiacich ako hosťujúci materiál a pridaním emisného dopantu Ir(dmpq)₂(acac) v pomere 3 wt% [4].



Obr. 3. Štruktúra navrhnutého a vyrobeného OLED prvku.

Meranie prebiehalo v dusíkovom boxe a v špeciálnom prípravku pre testovanie OLED prvkov. Elektrické charakteristiky boli merané vymeriavačom *Keysight B1500A* a použitím programu *Keysight EasyEXPERT*. Spektrálne charakteristiky boli merané spektrometrom *Ocean Optics Red Tide USB650* a použitím programu *SpectraSuite*.

2.3 Výsledky

Pre úplnosť je na Obr. 4 zobrazená štruktúra nami-vytvoreného prvku vo forme energetického diagramu.



Obr. 4. Energetický pásmový diagram navrhnutého a vyrobeného OLED prvku.

Volt-ampérové charakteristiky boli merané v intervale od -43 do 43 V s krokom 1 V. Prvky začali znateľne vyžarovať svetlo od hodnoty 30 V a vyššie, prierazné napätie bolo experimentálne určené na 50 V. Z nameraných dát je vidieť, že volt-ampérová charakteristika má v priepustnom smere charakter klasický usmerňujúci a v závernom smere prúd blokuje. Prúdy dosahovali hodnoty desiatky μA (konkrétne reprezentatívna vzorka dosiahla najvyšší prúd 74,5 µA pri napätí 43 V).

Volt-ampérová charakteristika reprezentatívnej vzorky je graficky znázornená na Obr. 5.



Obr. 5. Nameraná volt-ampérová charakteristika vyrobeného OLED prvku.

Spektrálne charakteristiky boli merané súbežne s voltampérovými charakteristikami, čiže spolu priamo korešpondujú. Nameraná intenzita elektroluminiscencie (*skr*. EL) bola normovaná a je vztiahnutá k príslušným vlnovým dĺžkam z intervalu 350 až 1000 nm s krokom 1 nm.

Spektrálna charakteristika reprezentatívnej vzorky je graficky znázornená na Obr. 6. Následne bola fitovaná Gaussovou funkciou, ktorá matematicky udala hodnoty uvedené v pridruženej tabuľke (automaticky implementovaná v Obr. 6). Najpodstatnejším parametrom je *xc*, ktorý označuje stred maxima funkcie. Inými slovami hovorí, že daná OLED vyžaruje na vlnovej dĺžke 620 nm, čo je podľa všeobecnej definície na spodnej hranici červenej farby.



Obr. 6. Nameraná a fitovaná spektrálna charakteristika vyrobeného OLED prvku.

Reálne vizuálne výsledky možno vidieť na Obr. 7 v priloženej fotografii.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 7. Fotografía rozsvieteného OLED prvku v testovacom prípravku.

3 Záver

použitý v tejto práci.

Vytvorením funkčného prototypu OLED na princípe emisnej dopácie sme potvrdili správnosť teórie. Doteraz sme boli na Oddelení organickej elektroniky schopní vyrábať iba referenčné zelené OLED na báze emisných materiálov 1. generácie (napr. Alq₃), takže sa jedná o veľký posun vpred. Druhá generácia emisných materiálov zlepšuje a aj otvára nové vlastnosti nami-vyrobených OLED prvkov a tým pádom rozširuje možnosti ich nasledovného použitia.

Vyrobené prvky vykazovali veľmi malé prúdy, ale boli riadené znateľne vyšším napätím. Tieto skutočnosti vytvárajú nové výzvy pre vylepšenia technológie do budúcna, napríklad za účelom zníženia prahového napätia. To je možné dosiahnuť napríklad úpravou hrúbky jednotlivých vrstiev, použitím iných materiálov, alebo zmenou geometrie prvkov. Úmerne tomu však môže narastať zložitosť prípravy prvku a tým sa zvyšovať riziko vzniku chýb.

Hlavným lákadlom je určite lepšia (v niektorých prípadoch aj úplne nová) realizácia farieb. Výsledná farba prvku závisí primárne od materiálu nadopovaného do emisnej vrstvy, pričom keď použijeme dopanty z rovnakej skupiny (napríklad komplexy Ir), zvyšok štruktúry môže jednoducho ostať zachovaný. Ako teoretické príklady do ďalšieho výskumu uvádzame zelený dopant Bis[2-(2-pyridinyl-N)phenyl-C](acetylacetonato)iridium(III) (Ir(ppy)₂(acac)) [4], ďalej oranžový dopant Bis(2-(9,9-diethyl-fluoren-2-yl)-1-phenyl-1H-benzo[d]imidazolato)(actylacetonate)iridium(III) (Ir(fbi)₂(acac)) [6], ako aj červený dopant Bis[2-(3,5-dimethylphenyl)-quinoline](acetylacetonate)iridium(III) (Ir(mpq)₂(acac)) [4], ktorého ekvivalent Ir(dmpq)₂(acac) bol

Všetko má aj svoje nevýhody, v tomto prípade hlavne technologického charakteru a to mierne vyššiu cenu materiálov (čo súvisí aj s obmedzenou dostupnosťou na našom trhu) a komplikovanejší výrobný proces, spôsobený hlavne nutnosťou súbežného vákuového naparovania viacerých materiálov pre vytvorenie emisnej vrstvy.

Vo výsledku však možnosti, ktoré nám táto technológia môže ponúknuť a kam nás ďalej posunúť, výrazne prevyšujú relatívne negatíva a je výhodné sa ňou ďalej zaoberať. Minimálne pokiaľ nám nebude prístupné lepšie riešenie, napríklad 3. generácia emisných materiálov, ktorá je momentálne na vzostupe.

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Pod'akovanie

Moje úprimné poďakovanie patrí Ing. Michalovi Mičjanovi, PhD. a prof. Ing. Martinovi Weisovi, DrSc. za vedenie tejto práce, cenné rady a pripomienky. Ďalej Ing. Miroslavovi Novotovi, PhD. za veľkú pomoc vo viacerých technologických oblastiach pri realizácii experimentu. A v neposlednom rade Ing. Michalovi Sobotovi za asistenciu pri meraní elektrických a optických vlastností realizovaného zariadenia.

Tento článok vznikol vďaka podpore v rámci projektu grantovej agentúry Ministerstva školstva, vedy, výskumu a športu SR APVV-20-0564.

Literatúra

- [1] MERTENS, Ron. What is an OLED?. MERTENS, Ron. The OLED Handbook: A guide to the OLED industry, technology and market. 2021 ed. Lulu, 2021, p. 7 - 54. ISBN 1716256925.
- [2] HONG, Gloria, Xuemin GAN, Céline LEONHARDT, Zhen ZHANG, Jasmin SEIBERT, Jasmin M. BUSCH a Stefan BRÄSE. A Brief History of OLEDs—Emitter Development and Industry Milestones. Advanced Materials [online]. 2021, 33(9) [cit. 2022-04-12]. ISSN 0935-9648. Dostupné z: doi:10.1002/adma.202005630
- [3] MIČJAN, Michal a Martin WEIS. Organic electronic materials and devices. YURISH, Sergey Y. Advances in Microelectronics: Reviews. Vol. 3. Barcelona: IFSA Publishing, 2021, p. 19-74. ISBN 978-84-09-33338-7.
- [4] LEE, Sunghun, Daniel LIMBACH, Kwon-Hyeon KIM, Seung-Jun YOO, Young-Seo PARK a Jang-Joo KIM. High efficiency and non-color-changing orange organic light emitting diodes with red and green emitting layers. Organic Electronics [online]. Amsterdam (NL): Elsevier, 2013, July 2013, 2013(Vol. 14, Iss. 7), p. 1856 - 1860 [cit. 2022-04-12]. Dostupné z: doi:https://doi.org/10.1016/j.orgel.2013.04.020
- [5] Ossila: enabling materials science [online]. Sheffield: Ossila, c2022 [cit. 2022-04-12]. Dostupné z: https://www.ossila.com/
- [6] ZHANG, Baohua, Guiping TAN, Ching-Shan LAM, et al. High-Effi ciency Single Emissive Layer White Organic Light- Emitting Diodes Based on Solution-Processed Dendritic Host and New Orange-Emitting Iridium Complex. Advanced Materials [online]. Weinheim: Wiley, 2012, 13 March 2012, 2012(Vol. 24, Iss. 14), p. 1873 - 1877 [cit. 2022-04-12]. Dostupné z: doi:https://doi.org/10.1002/adma.201104758

Ako dlho sa vybíja kondenzátor cez LED diódu?

Kristína Gažová¹, Peter Bokes¹

¹Ústav jadrového a fyzikálneho inžinierstva, Fakulta elektrotechniky a informatiky, Slovenská technická univerzita v Bratislave, Slovenská Republika

xgazovak1@stuba.sk, peter.bokes@stuba.sk

Práca sa zaoberá teoretickým analytickým aj numerickým modelovaním vybíjania kondenzátora cez sériovo zapojenú LED diódu, ktorá predstavuje nelineárny elektronický prvok. Predpovede teoretických modelov sú porovnané s meraním časového priebehu poklesu napätia na kondenzátore v zodpovedajúcom elektronickom obvode. Numerický výpočet, ktorý zahŕňa nelineárnu volt-ampérovú charakteristiku, poskytuje mierne lepší opis reality ako jednoduchý lineárny model LED.

1 Úvod

Kondenzátor, ktorý je zaradený v obvode sériovo s rezistorom by sa mal vybíjať podľa časovej konštanty $\tau = RC$, kde *C* je kapacita kondenzátora a *R* je odpor rezistora [1]. Podľa teoretických znalostí po uplynutí času by malo byť na kondenzátore nulové napätie.

Tento jav je možné jednoducho demonštrovať, keď bude v obvode sériovo s ostatnými prvkami zaradená LED dióda (z ang. Light-Emitting Diode). Po uplynutí doby $\tau = 3\tau$ je kondenzátor na 95% vybitý, LED diódou by nemal prechádzať prakticky žiaden elektrický prúd a teda LED dióda by nemala svietiť. Pri praktickej realizácií tohto javu je možné odsledovať, že LED dióda svieti dlhšie ako udáva teoreticky určená časová konštanta τ a teda na kondenzátore je nenulové napätie výrazne dlhšiu dobu.

2 Teoretický model vybíjania kondenzátora

2.1 Opis vybíjacieho obvodu pre kondenzátor



Obr. 1: Schéma zapojenia obvodu, v ktorom dochádza k vybíjaniu kondenzátora cez LED.

Obvod pozostáva zo sériovo zapojeného kondenzátora,

rezistora a LED diódy, Obr. 1. Po zopnutí spínača v čase t = 0 začne obvodom prechádzať elektrický prúd. LED dióda svieti a kondenzátor sa postupne vybíja. Mohlo by sa zdať, že keď dióda prestane svietiť, bude kondenzátor vybitý. V skutočnosti napätie na kondenzátore nie je nulové. Napätie na kondenzátore U_C je porovnateľ né s prahovým napätím U_f (z ang. Forward voltage) na LED dióde, čo vyplýva z VA charakteristiky (volt - ampérová charakteristika) LED diódy.

Na základe druhého Kirchhoffovho zákona môžeme pre vybíjací obvod písať:

$$\frac{q}{C} + R\frac{\mathrm{d}q}{\mathrm{d}t} + U_{\mathrm{D}} = 0 \tag{1}$$

kde q je náboj na kondenzátore, C je kapacita kondenzátora a $U_{\rm D}$ napätie na dióde, ktoré sa určuje podľ a VA charakteristiky LED diódy.

2.2 Voltampérová charakteristika LED diódy

VA charakteristika (Obr. 2 opisuje správanie sa diódy nelineárneho prvku v obvode.



Obr. 2: Fialovou krivkou je znázornená meraná VA charakteristika LED diódy. Zelená krivka zobrazuje lineárny model, ktorý znázorňuje hodnoty prúdu vyššie ako je hodnota prúdu I_c . Sú to hodnoty, ktoré sa nachádzajú v zobrazenom grafe za kolenom VA charakteristiky. Modrá krivka zobrazuje lineárny model pre hodnoty prúdu, ktoré sú nižšie ako hodnota prúdu I_c a v grafe sa nachádzajú pred kolenom VA charakteristiky.

Ak je napätie na LED dióde väčšie ako prahové napätie $U_{\rm f}$, tak obvodom prechádza elektrický prúd a LED dióda svieti. Pokles napätia na prahové napätie $U_{\rm f}$ zapríčiní, že dióda prestáva

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

prepúšť ať prúd a pri nízkych napätiach môžeme považovať prúd za nulový. Mieru nárastu prúdu v okolí vybranej hodnoty prúdu I charakterizuje diferenciálny odpor diódy R_d ,

$$R_{\rm d} = \left. \frac{\mathrm{d}U}{\mathrm{d}I} \right|_{I} \tag{2}$$

V rámci najjednoduchšieho *lineárneho modelu* sa predpokladá, že diferenciálny odpor od hodnoty prúdu *I* nezávisí ak je napätie na dióde väčšie ako je prahové napätie $U_{\rm f}$. Napätie na LED potom môžeme vyjadriť jednoduchým vzť ahom

$$U_{\rm D}(I) = U_{\rm f} + R_{\rm d}I. \tag{3}$$

Takýto najjednoduchší model ale neodzrkadľuje to, že pre veľmi malé hodnoty prúdu napätie na dióde predsa len klesá a nakoniec dosiahne nulovú hodnotu pre nulový prúd. Toto možno modelovať dodatočnou lineárnou časťou modelu VA charakteristiky vychádzajúcou z počiatku, a majúcu veľmi veľký diferenciálny odpor R_{d2} .

Napätie na dióde $U_D(I)$ môžeme v rámci tohto druhého *počastiach lineárneho modelu* vyjadriť vzť ahom

$$U_{\rm D}(I) = \begin{cases} R_{d2}I & I < I_c \\ U_f + R_dI & I > I_c \end{cases}$$
(4)

Kritická hodnota prúdu I_c je taká hodnota, v ktorej sa lineárne modely zobrazené v grafe pretínajú. táto hodnota sa dá určiť výpočtom z rovnosti vyjadrení napätia na dióde,

$$I_c = \frac{U_f}{R_{d2} - R_d}.$$
 (5)

Výhodou po-častiach lineárneho modelu je to, že preň dokážeme nájsť analytické riešenie diferenciálnej rovnice (1). Na druhej strane, konečná hodnota R_{d2} neopíše VA charakteristiky veľmi spoľ ahlivo a preto najlepším modelom VA charakteristikou by bola samotná tá, ktorú nameriame. Preto v našej práci nájdeme aj plne numerické riešenie rovnice (1) s využitím lineárnej interpolácie meranej VA charakteristiky červenej LED. Takýto model vybíjania budeme nazývať *interpolovaným* modelom.

2.3 Analytický výpočet vybíjania kondenzátora

Pri analytickom výpočte vybíjania kondenzátora budeme vychádzať z po-častiach lineárnej VA charakteristiky LED diódy. VA charakteristika sa skladá z dvoch lineárnych častí. Tie sa rozdeľ ujú podľa prechádzajúceho prúdu diódou. Vzhľadom na to, že vrámci výpočtu sú graficky zobrazované hodnoty náboja od času, je vhodnejšie udávať ako hraničný parameter pre lineárne modely hodnotu času t_c , v ktorom prúd prechádzajúci LED poklesne na kritický prúd I_c .

Lineárny model pre čas vyšší ako je hodnota času t_c

Kirchhoffov zákon opisujúci obvod, ktorým prechádza prúd s vyššími hodnotami ako je hodnota prúdu I_c je v tvare:

$$\frac{q}{C} - U_{\rm f} + R_d \frac{\mathrm{d}q}{\mathrm{d}t} + R \frac{\mathrm{d}q}{\mathrm{d}t} = 0 \tag{6}$$

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Nakoľko ide o lineárnu diferenciálnu rovnicu prvého rádu, riešenie dokážeme nájsť štandardným postupom [2] s výsledkom:

$$q(t) = U_{\rm f}C + (U_0 - U_{\rm f})Ce^{-t/\tau_1}, \tag{7}$$

kde časová konštanta au_1 sa rovná:

$$r_1 = (R_d + R)C \tag{8}$$

Lineárny model pre čas nižší ako je hodnota času t_c

τ

Pre $t < t_c$ použijeme prvú časť VA charakteristiky LED v rámci rovnice (1),

$$\frac{q}{C} + R_{d2}\frac{\mathrm{d}q}{\mathrm{d}t} + R\frac{\mathrm{d}q}{\mathrm{d}t} = 0 \tag{9}$$

Výpočet lineárneho modelu sa tiež realizuje pomocou metódy variácie konštánt pre základné diferenciálne rovnice, s výsled-kom:

$$q(t) = \left(U_{\rm f} C + (U_0 - U_{\rm f}) C e^{-t_c/\tau_1} \right) e^{-(t-t_c)/\tau_2}$$
(10)

kde časová konštanta τ_2 je rovná:

$$\tau_2 = (R_{d2} + R)C.$$
(11)

Vo vyjadrení vystupuje hodnota premennej t_c , ktorú vypočítame derivovaním predchádzajúceho výrazu podľa času – výpočtu náboja q(t) pre hodnoty času vyššie ako je hodnota t_c a následným upravovaním nájdeme

$$t_c = -\tau_1 (U_0 - U_f) C \ln \left(\frac{I_c \tau_1}{(U_0 - U_f)C} \right)$$
(12)

Zobrazené lineárne modely



Obr. 3: Analytické znázornenie vybíjania kondenzátora. Zelenou krivkou je zobrazený lineárny model pre hodnoty prúdu nižšie ako je hodnota prúdu I_c , platný pre $t < t_c$. Fialovou krivkou je znázornený lineárny model pre hodnoty prúdu vyššie ako je hodnota prúdu I_c , platný pre $t > t_c$.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

V grafe na Obr. 3 sú zobrazené oba výsledky pre obe časti po-častiach lineárneho modelu, rovnica (7) aj (10) pre parametre VA charakteristiky $U_{\rm f} = 10$ V, $R_{\rm d} = 44,5$ Ω , a $R_{\rm d2} = 50000$ k Ω . Počiatočné napätie kondenzátora je $U_0 = 10$ V, kapacita kondenzátora C = 0.001 C a R = 10000k Ω .

V skutočnosti je graf vybíjania kondenzátora vybranou kombináciou zobrazených kriviek. Od začiatku vybíjania po čas t_c priebeh kopíruje fialová krivka. Po čase t_c vybíjanie kondenzátora kopíruje zelená krivka.

2.4 Numerická implementácia riešenia pre všeobecnú nelineárnu VA charakteristiku

Ak chceme použiť presný nelineárny model VA charakteristiky LED je nevyhnutné riešiť diferenciálnu rovnicu (1) numericky. Eulerova metóda predstavuje najjednoduchšiu z numerických metód, pričom jej stabilitu možno výrazne zlepšiť voľbou času pre výpočet jej jednotlivých výrazov [3].

V rámci tu použitej numerickej aproximácie možno prúd vyjadriť v tvare:

$$I = \frac{q_{(t+\Delta t)} - q_{(t)}}{\Delta t} \tag{13}$$

Potom pre obvod vyjadrený druhým Kirchhoffovým zákonom platí:

$$\frac{q_{(t+\Delta t)}}{C} + R \frac{q_{(t+\Delta t)} - q_{(t)}}{\Delta t} + U_{\rm D} \left(\frac{q_{(t+\Delta t)} - q_{(t)}}{\Delta t}\right) = 0 \tag{14}$$

Problém je analyticky vyjadriť hodnotu $q_{(t+\Delta t)}$ z nelineárnej funkcie $U_{\rm D}(I)$. Pre riešenie tohto problému sme použili iteratívny postup v rámci ktorého sme numericky hľadali postupné aproximácie tejto hodnoty $q_{(t+\Delta t)}^{(n)}$ pomocou funkcie Findqnew (Obr. 4).

```
double Findqnew(double q, double R, double C, double dt, double t;
{
    double q_new,q_new0;
    double mix=0.2;
    double error=1.e-6;
    q_new=q-dt*(q/(C*R));
    q_new0 = q_new;
    do {
        q_new0 = mix*q_new+(1-mix)*q_new0;
        q_new0 = mix*q_new+(1-mix)*q_new0;
        q_new0 = q_new;
        do {
            q_new0 = q_new;
            do {
            q_new0 = q_new;
            do {
            q_new0 = q_new;
            do {
            q_new0 = q_new0/C + Udx((q_new0-q)/dt ));
            }
        while(fabs((q_new-q_new0)/q_new) > error);
        return(q_new);
```

Obr. 4: Ukážka funkcie Findqnew z programu

Opis iteračnej funkcie Findqnew

Pri prvom volaní funkcie sa do premenných priradí hodnota, ktorá zodpovedá veľkosti náboja, keď nie je obvode zaradená LED dióda,

$$\frac{q_{(t+\Delta t)}^{(1)}}{C} + R \frac{q_{(t+\Delta t)}^{(1)} - q_{(t)}}{\Delta t} = 0$$
(15)

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

 $q_{(t+\Delta t)}^{(1)}$ je počiatočný odhad náboja v nasledujúcom časovom okamihu, ktorý sa spresňuje iteratívnym riešením rovnice

$$\frac{q_{(t+\Delta t)}^{(n)}}{C} + R \frac{q_{(t+\Delta t)}^{(n+1)} - q_{(t)}}{\Delta t} + U_{\rm D} \left(\frac{q_{(t+\Delta t)}^{(n)} - q_{(t)}}{\Delta t}\right) = 0$$
(16)

Vrámci cyklu je nová hodnota po každej iterácii získaná ako vážený priemer z posledných dvoch iterácií, pričom váha hodnoty z poslednej iterácie je daná premennou mix a váha hodnoty z predposlednej iterácie (1-mix),

$$q_{(t+\Delta t)}^{(n+1)}$$
mix + $(1 - mix)q_{(t+\Delta t)}^{(n)} \to q_{(t+\Delta t)}^{(n+1)}$. (17)

Dôvodom na použitie váhového priemer z posledných dvoch iterácií je tlmenie oscilácií priebežných výsledkov, ku ktorým by mohlo dôjsť počas chodu cyklu a teda, aby iteračná podmienka mohla byť splnená a cyklus bol ukončený.

Cyklus sa opakuje pokial' je relatívny rozdiel hodnôt dvoch nasledujúcich iterácii väčší ako predpísaná chyba výpočtu. Hodnota predpísanej chyby je na začiatku funkcie programu. Ak je podmienka splnená cyklus sa skončí a funkcia programu vráti hodnotu premennej $q_{(t+\Delta t)}^{(n+1)}$, ktorá je využitá v programe na ďalšie výpočty.

2.5 Diskusia presnosti numerického výpočtu

V predchádzajúcich častiach boli opísané dva výpočty náboja na kondenzátore, ktorý sa vybíja v čase. Prvý spôsob je analytický, ktorý sa odvíja od modelu po-častiach lineárnej VA charakteristiky LED diódy. Druhý spôsob je numerický a umožňuje využitie experimentálne určenej VA charakteristiky. Aby sme overili presnosť numerického výpočtu použijeme miesto meranej VA charakteristiky jednoduchší model počastiach lineárnej charakteristiky (4).



Obr. 5: Porovnanie numerického a analytického výpočtu. Fialová krivka zobrazuje výpočet vybíjania kondenzátora numerickou metódou. Zelená krivka zobrazuje výpočet vybíjania kondenzátora analytickou metódou. Vložený graf zobrazuje relatívny rozdiel oboch metód vo vypočítanom náboji pre daný čas.

V grafe na Obr. 5 sa zobrazené krivky popisujúce stav náboja na kondenzátore pre oba modely prekrývajú. Vidíme, že obe závislosti sú vo veľmi dobrom súhlase a teda môžeme dôverovať našej numerickej implementácii. Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

3 Meranie vybíjania kondenzátora pomocou osciloskopu

Pre porovnanie teoretických modelov výpočtu náboja na kondenzátore so skutočnosťou sme využili meranie pomocou osciloskopu, ktorý sníma ako sa mení napätie na kondenzátore v čase. Použitý bol USB osciloskop PicoScope 4224 IEPE od firmy Pico Technology.

V obvode bol požitý elektrolytický kondenzátor s kapacitou C = 1 mF, odpor $R = 10 \text{ k}\Omega$, modrá LED dióda. Na nabíjanie bola použitá batéria s napätím $U_0 = 9,3 \text{ V}$. Na realizáciu merania sme zapojili dva obvody. V prvom obvode s batériou a pomocným rezistorom sa kondenzátor nabíja. Druhý "vybíjací" obvod sa skladá z rezistora, LED diódy a paralelne ku kondenzátoru sme zapojili osciloskop.

V Obr. 6 sú uvedené časové priebehy vybíjania kondenzátora pri použití troch rôznych LED. Pre porovnanie s numerickým výpočtom v projekte je dôležitá len charakteristika s modrou LED diódou, no počas merania bolo zobrazených viac charakteristík vybíjania kondenzátora aj pre iné farby LED diódy, pre porovnanie vplyvu konkrétnej VA charakteristiky danej LED a zostatkového napätia na kondenzátore. Z nameraných priebehov je vidieť, že zostatkové napätie závisí od farby použitej LED, čo zodpovedá aj odlišným hodnotám ich prahového napätia.



Obr. 6: Vybíjanie kondenzátora pre červenú (RED), zelenú (GREEN) a modrú (BLUE) LED diódu.

4 Porovnanie výsledkov z výpočtu a merania

Porovnanie nameraného priebehu napätia na kondenzátore (Obr. 6) s vypočítaným priebehom náboja na kondenzátore pre po-častiach lineárny model (Obr. 5) jasne naznačuje, že zvolená hodnota R_{d2} vôbec nezodpovedá skutočnosti. Preto pri ďalšej diskusii budeme s meraním porovnávať len predpovede jednoduchého lineárneho modelu a numerického modelu s nameranou VA charakteristikou.

Tieto výsledky sú ukázané na obrázku 7. Zelenou krivkou je zobrazené meranie napätia osciloskopom na kondenzátore.



Obr. 7: Porovnanie merania, analytického a numerického výpočtu pre modrú LED.

Žltá krivka zobrazuje lineárny model pre hodnoty $U_0 = 9,3$ V, $U_f = 2,44$ V a $R_d = 44,84 \Omega$. Fialovou krivkou je znázornený výsledok pre numerický model výpočtu vybíjania kondenzátora v čase s nameranou VA charakteristikou. Zo zobrazeného priebehu je vidieť, že analytický výpočet ako aj numerický výpočet skoro presne kopíruje nameraný priebeh.

Pri vybíjaní kondenzátora cez LED je proces vybíjania pomalší ako keby LED v obvode nebola, nakoľko časová konštanta zahŕňa aj vnútorný diferenciálny odpor LED a naviac, napätie na kondenzátore a teda aj na LED sa dlho udržiava nad hodnotou prahového napätia $U_{\rm f}$.

Cieľ om zapojenia LED do vybíjacieho obvodu je to, aby čas vybíjania kondenzátora bolo možné odhadnúť aj jednoduchým zmeraním doby svietenia diódy. Pri pohľade na LED diódu zboku, je čas jej svietenia približne zhodný s hodnotou $3\tau \approx 30$ s, t.j. časom vybitia kondenzátora na 90%. Znamená to, že pri pozorovaní zboku je ľudské oko schopné zachytiť svetlo z LED diódy, ak ňou prechádza prúd $I = 34,5 \,\mu$ A . Pri pohľade na LED diódu zvrchu cez jej šošovku je možné registrovať svetlo z LED až počas t = 60 s, kedy je hodnota prúdu, ktorá prechádza LED diódou len $I = 2 \,\mu$ A. Interpretácia doby svietenia diódy len pomocou časovej konštanty τ_1 teda nie je jednoznačná.

Literatúra

- [1] D. Halliday, R. Resnick, and J. Walker. *Fundamentals of Physics*, New York: Wiley, 1997.
- [2] A. Grega, D. Kluvanec a E. Rajčan. *Matematika pre fyzikov*, Slovenské pedagogické nakladateľ stvo Bratislava, 1975.
- [3] William H. Press et. al. Numerical Recipes in C : the Art of Scientific Computing. Cambridge [Cambridgeshire]; New York :Cambridge University Press, 1992.

Teplotné pole na povrchu Leslieho kocky

Karolína Kovalčíková, Peter Bokes

Ústav jadrového a fyzikálneho inžinierstva, FEI STU v Bratislave

xkovalcikova@stuba.sk

V práci sa zaoberáme meraním homogenity teplotného poľa na povrchu Leslieho kocky a emisivity jednotlivých stien kocky prostredníctvom termokamery, fotometra, termistora a termočlánku s cieľom overenia ich presnosti. Porovnaním teplôt vo vybraných miestach povrchu kocky pomocou presnejších senzorov môžeme vylepšiť kalibráciu termokamery.

1 Úvod

Zohriate telesá vyžarujú elektromagnetické žiarenie z celého spektra, no podstatná časť je sústredená do intervalu v pásme infračerveného žiarenia, konkrétne $\lambda = (8 - 12)\mu m$. Toto žiarenie označujeme ako tepelné žiarenie.

Analýzou rozloženia tepelného poľa na povrchu telesa sa zaoberá termografia. Termografickým meracím systémom je možné zobraziť tepelné pole meraného objektu na jeho povrchu tak, aby bolo viditeľné ľudským okom. Výstupom z termokamery je termogram, ktorý je možné analyzovať a určiť teplotu v jednotlivých bodoch.

Na overenie presnosti merania termokamery vzhľadom na emisivitu povrchu bolo uskutočnené meranie na Leslieho kocke [1]. Leslieho kocka je uzavretá nádoba s tvarom kocky, umiestnená na otočnom podstavci. Takáto kocka je zdrojom infračerveného žiarenia, pričom jej štyri bočné steny majú rôzne emisivity. Kocka je vyhrievaná volfrámovou žiarovkou umiestnenou vnútri kocky. Teplota kocky sa ovplyvňuje regulátorom príkonu žiarovky, umiestneným na podstavci kocky. Okamžitá teplota sa určuje meraním elektrického odporu termistora zabudovaného v kocke, pomocou ohmmetra.

Motiváciou tohto výskumu bola predovšetkým kalibrácia termokamery na účely diplomovej práce, ktorej obsahom je pokračovania v skúmaní prestupu tepla pozdĺž trubice [2] a porovnanie teoretického výpočtu s nameranými hodnotami.



2 Termokamera

Konštrukcia termokamery sa skladá z optiky a detektora. Optika fungujúca zároveň ako filter selektuje časť elektromagnetického žiarenia, dôležitú z hľadiska merania teploty t.j. infračervené žiarenie a sústreďuje toto žiarenie na detektor. Tvorená je šošovkou ošetrenou antireflexnou vrstvou zabraňujúcou odrazu žiarenia čím zvyšuje jej priepustnosť.

Detektor (polovodič) prevádza dopadajúce infračervené žiarenie na elektrický signál. Signál je ďalšou elektronikou prevedený na výsledný termogram, ktorý je prezentovaný užívateľovi termokamery ako výsledok merania. Z hľadiska princípu funkcie rozlišujeme dva základné typy detektorov, a to tepelné (pyroelektrické) a fotónové.

Tepelné detektory fungujú na princípe zmeny elektrických vlastností v závislosti na intenzite dopadajúceho infračerveného žiarenia. Takéto detektory sú využívané vo väčšine termokamier, majú vysokú využiteľnosť vzhľadom k širokopásmovej spektrálnej citlivosti, takže sú schopné detegovať žiarenie v širokom rozsahu vlnových dĺžok.

Fotónové detektory pracujú na princípe počítania fotónov t.j. kvánt žiarenia. Sú úzkopásmové t.j. schopné detegovať žiarenie v úzkom spektre vlnových dĺžok, avšak pri vyššej citlivosti. Na rozdiel od tepelných detektorov vyžadujú chladenie.

Elektrický signál vychádzajúci z detektoru je spracovaný mikroprocesorom. Užívateľské rozhranie slúži primárne k zobrazovaniu termogramov na displeji termokamery a ovládanie samotnej termokamery prostredníctvom ovládacích tlačidiel alebo dotykovým displejom.

Kvalita výstupu termokamery závisí od rôznych parametrov ako napríklad rozlíšenie detektoru udávajúce počet bodov, do ktorých je rozdelená snímaná plocha, teplotný rozsah alebo citlivosť definovaná ako najmenší rozdiel teplôt, ktorý je schopná daná termokamera zachytiť. [3]

Meranie bolo realiyované pomocou termokamery *FLUKE PTi120 Pocket Thermal Imager*. Je to ručná termokamera s tepelným detektorom. Výrobcom udávaná presnosť merania teploty je ± 2 °C, rozlíšenie IR čipu je 120×90 pixelov, rozsah žiarenia 8-14 µm teplotná citlivosť ≤ 60 mK. Termokamera zobrazuje termosnímky na dotykovej LCD obrazovke a ukladá ich do internej pamäti, z ktorej je možné preniesť ich do počítača cez USB rozhranie a následne analyzovať softvérom *FLUKE Connect*.

Táto termokamera má predvolenú hodnotu emisivity 0,94 a nastavená je na meranie vo vzdialenosti 80 cm.

Obr. 1. Leslieho kocka

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 2. FLUKE PTi120 Pocket Thermal Imager

3 Meranie

Pre žiarivú energiu E_0 predstavujúcu energiu elektromagnetických vĺn dopadajúcich na jednotku povrchu telesa za jednotku času platí [4]:

$$E_0 = E_\alpha + E_r + E_d \tag{1}$$

pričom E_{α} je časť energie absorbovanej telesom, E_r je časť energie odrazenej telesom a E_d je časť energie prejdenej telesom. Po následnom vydelení rovnice celkovou dopadajúcou energiou:

$$1 = \frac{E_{\alpha}}{E_0} + \frac{E_r}{E_0} + \frac{E_d}{E_0}$$
(2)

vieme vyjadriť veličiny:

$$\alpha = \frac{E_{\alpha}}{E_0} \tag{3}$$

$$r = \frac{E_r}{E_0} \tag{4}$$

$$d = \frac{E_d}{E_0} \tag{5}$$

kde α je pomerná pohltivosť (absorptancia), r je pomerná odrazivosť a d pomerná priepustnosť. Vďaka týmto veličinám môžeme prepísať rovnicu na tvar:

$$1 = \alpha + r + d \tag{6}$$

Na základe hodnoty daných parametrov rozlišujeme tri modely telies:

- dokonale čierne (a = 1, r = 0, d = 0),

- dokonale biele (a = 0, r = 1, d = 0),

- dokonale priepustné (a = 0, r = 0, d = 1).

Intenzita vyžarovania povrchu *M* udáva množstvo energie vyžiarenej jednotkovým povrchom telesa za jednotku času a je určená Stefanovým-Boltzmannovým zákonom [2,3]:

$$M = \varepsilon \sigma_{SB} T^4 \tag{7}$$

kde ε je emisivita povrchu, $\sigma_{SB}=5,67\cdot10^{-8}~\rm W\cdot m^{-2}\cdot K^{-4}$ je Stefanova-Boltzmannova konštanta a T je termodynamická

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

teplota povrchu. Intenzita vyžarovania rastie so štvrtou mocninou povrchovej teploty telesa.

Emisivita vyjadruje mieru schopnosti povrchu telesa vyžarovať elektromagnetické žiarenie. Je definovaná ako podiel vyžiarenej energie telesa a vyžiarenej energie dokonale čierneho telesa (emisivita dokonale čierneho telesa $\varepsilon = 1$).

Meranie, ktorým overujeme Stefanov-Boltzmannov zákon, sa uskutočňuje fotometrom - senzorom intenzity žiarenia.



Obr. 3. Fotometer

Fotometer sa pri každom meraní oprie k stene kocky o dištančné opierky a uvedie do činnosti stlačením mechanickej clony, ktorá mimo merania chráni termočlánok fotometra pred nežiaducim ohriatím. Jednotlivé merania nemajú trvať viac ako 5 sekúnd. Elektrické napätie ktoré na senzore vzniká, sa meria citlivým voltmetrom. Nastavenie nuly na voltmetri pred každým meraním sa robí otáčaním červeného gombíka, ktorý je umiestnený na bočnej stene senzora, pričom fotometer opierame o čiernu plastovú platňu, ktorá má teplotu okolia. Údaj na voltmetri nevyjadruje intenzitu vyžarovania, no toto napätie je veličine priamoúmerné. Násobiaci koeficient však nepoznáme.



Obr. 4. Aparatúra merania tepelného žiarenia na Leslieho kocke

Merali sme intenzity vyžarovania čiernej, bielej a kovovej matnej steny kocky, ktorá bola vyhriata na teplotu približne 70°C. Pri konštantnej teplote sa zisťovali intenzity vyžarovania rôznych povrchov kocky. Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

V nasledujúcej tabuľke môžeme vidieť priemernú hodnotu napätia na fotometri U_i , strednú kvadratickú odchýlku σ_i priemernej hodnoty napätia U_i určenej z troch meraní a hodnotu emisivity ε_i . Emisivitu sme určili pomocou vzťahu:

$$\varepsilon_i = \varepsilon_{\check{c}} \left(\frac{U_i}{U_{\check{c}}} \right) \tag{8}$$

kde $\varepsilon_{\varepsilon} = 0.94$ je emisivita čierneho povrchu kocky a U_{ε} napätie na fotometri pre tento povrch pri 70°C. Táto hodnota emisivity čierneho povrchu vedie na dobrú zhodu merania teploty v strede čiernej steny kocky termokamerou a pomocou termočlánku (obr. 10).

Tab. 1. Porovnanie vyžarovania rôznych stien Leslieho kocky

VELIČINY	ČIERNA	BIELA	KOVOVÁ MATNÁ
Ui	0,2575	0,2434	0,1130
σ_{i}	0,0091	0,0213	0,0049
εί	0,94	0,89	0,41

Ďalším pokusom bolo meranie vyžarovania čierneho povrchu kocky. Kocka sa zohriala na približne 80°C. Vyhrievanie sa vyplo a kocka sa pomaly ochladzovala. Počas tohto procesu sa zisťovala intenzita vyžarovania pri rôznych teplotách kocky, konkrétne pri každom znížení teploty o 5°C pomocou termokamery a fotometra. Údaje odčítané z voltmetra boli vynesené do grafu ako funkcia rozdielu štvrtých mocnín teploty kocky a teploty okolia. Týmito bodmi sme preložili priamku lineárnou regresiou čím sme overili funkcionálny tvar Stefanovho-Boltzannovho zákona (7).



Obr. 5. Závislosť elektrického napätia fotometra od štvrtej mocniny absolútnej teploty

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Na zistenie teploty z nameraných hodnôt odporov termistora sme využili tabuľku závislosti teploty od jeho elektrického odporu, ktorá obsahovala hodnoty odporov pre celočíselné teploty od 10°C do 150°C. Na výpočet teplôt sme využili interpoláciu týchto hodnôt. Z grafu na obr. 6 možno posúdiť povahu interpolácie vzhľadom na počet údajov, z ktorých bola vyhotovená. Vzhľadom na túto závislosť sme na ďalšie výpočty interpoláciou použili len dva najbližšie body aktuálnej hodnoty odporu termistora.



Obr. 6. Závislosť interpolovanej teploty termistora od počtu okolitých údajov, z ktorých bola interpolácia počítaná



Obr. 7. Porovnanie merania termistora a termokamery

Meranie teploty sme overili aj termočlánkom, ktorý predstavuje spoj vodičov z chromelu (TFCY-005) a alumelu (TFAL-005) s priemerom 0,13 mm. Merali sme v piatich polohách, konkrétne rohy strán kocky a stred. Meranie bolo uskutočnené pre bielu a čiernu stenu kocky a teplota termočlánku bola vztiahnutá k teplote ľadu v izolovanej nádobe. Napätie vznikajúce na termočlánku bolo merané citlivým voltmetrom a následne boli tieto hodnoty prepočítané podľa tabuľky od výrobcu vodičov na hodnoty teploty.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Interpolácia bola počítaná z dvoch hodnôt pred/za nameranou hodnotou.



Obr. 8. Závislosť interpolovanej teploty termočlánku od počtu okolitých údajov, z ktorých bola interpolácia počítaná



Obr. 9. Meranie teploty pomocou termočlánku

Okrem merania termočlánkom sme samozrejme merali teplotu povrchov čiernej a bielej steny kocky aj termokamerou a následne sme tieto merania porovnali.



Obr.10. Porovnanie merania teploty čiernej steny kocky pri teplote 58,35°C na termistore termokamerou (vľavo) a termočlánkom (vpravo).



Obr. 11. Porovnanie merania teploty bielej steny kocky pri teplote 58,40°C na termistore termokamerou (vľavo) a termočlánkom (vpravo).

4 Emisivita povrchov Leslieho kocky

V programe *Fluke Connect* alebo priamo v termokamere je možné v rámci spracovania snímky zmeniť predpokladanú emisivitu snímaného povrchu, čo vedie na zmenu teploty na termosnímke. Vychádzajúc z rovnice (7) sa možno domnievať, že tento prepočet využíva vzťah:

$$\varepsilon_2 = \varepsilon_1 \frac{(T_1^4 - T_0^4)}{(T_2^4 - T_0^4)} \tag{9}$$

kde T_1 predstavuje teplotu odmeranú termokamerou pri jej nastavení na predpokladanú emisivitu ε_1 a ε_2 je emisivita, ktorú by sme mali nastaviť na termokamere, aby sme ňou získali skutočnú teplotu povrchu T_2 . Skutočnú teplotu povrchu T₂ meriame pomocou termočlánku. Túto domnienku sme chceli otestovať. Prepočet sme aplikovali na bielu a sivú stenu. V tabuľke 2 môžeme vidieť porovnanie emisivít určených pomocou fotometra s predpokladom, že emisivita čierneho povrchu je 0,94, čo je aj prednastavená hodnota keďže takéto hodnoty korešpondovali v termokamere, s hodnotami nameranými termočlánkom. Rozdiely mohli vzniknúť nepresnosťou merania pomocou fotometra, o čom svedčí pomerne veľká stredná kvadratická odchýlka strednej hodnoty napätia (tab. 1), prehriatím fotometra pri nedostatočnom vychladnutí po jednotlivých meraniach, nesprávnym nastavením hodnoty nuly pri snímaní referenčnej platne alebo snímaním len malej časti plochy strany kocky.

Tab. 2. Porovnanie emisivity vypočítanej zo vzťahu (9) a emisivity určenej podľa údajov z fotometra pre bielu a sivú stenu Leslieho kocky

	ε _B [-]	ε _s [-]
PREPOČET	0,91	0,39
FOTOMETER	0,89	0,41

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

5 Homogenita teplotného poľa na Leslieho kocke

Detailne bola skúmaná termosnímka pri teplote 80°C, na ktorej bol podrobne analyzovaný celý povrch čiernej steny kocky. V programe *FLUKE Connect* bolo možné pomocou markerov preskúmať teploty v každom mieste steny kocky. Najvyššie teploty boli dosahované v stredných častiach vo vyššej polovici steny.



Obr. 12. Teplotné pole na povrchu čiernej steny Leslieho kocky pri teplote 81,21°C na termistore

6 Záver

Vzhľadom na presnosť získaných výsledkov môžeme konštatovať, že rôznosť emisivít čiernej a bielej steny kocky je zanedbateľný. Vopred nastavená emisivita 0,94 pre meranie termokamerou je vhodne zvolená a relatívne presná v oboch prípadoch. Čo sa týka matnej kovovej steny, tu je potrebná korekcia. Presnosť termokamery vzhľadom na meranie teploty termistorom je uspokojivá. Nepresnosti mohli nastať stratami alebo polohou meracej aparatúry (priame slnečné žiarenie) prípadne nepresným ohmmetrom, ktorý snímal termistor. Zhoda merania teploty termistorom a termočlánkom je tiež uspokojivá. Tento rozdiel možno vysvetliť tým, že termistor je umiestnený bližšie k zdroju vyhrievania kocky a ďalej od steny kocky v porovnaní s termočlánkom. Taktiež pri meraní fotometrom sa snímal stred stien, nie celý povrch a z obrázku 12 je možné vidieť, že tepelné pole na povrchu nie je homogénne. Kvantitatívne, výsledky namerané termistorom sú o 1,1% vyššie a výsledky nasnímané termokamerou o 1,1% nižšie ako údaje z termočlánku, ktorý sme určili ako referenčný prvok.

7 Literatúra

- [1] Infračervené žiarenie (Laboratórna úloha ÚJFI FEI STU)., 2011. z: tp://www.ujfi.fei.stuba.sk/fyzika/navody/
- [2] KOVALČÍKOVÁ, K. Modelovanie a meranie vedenia tepla v kvázi-1D sústavách (Bakalárska práca). Bratislava : STU, 2021. 46 s.

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

- [3] Princip a funkce., 2022 z:
- http://www.termokamera.cz/princip-a-funkce/ [4] KALOUSEK, M., HUČKO, B. Prenos tepla. 1.vyd.
- Bratislava : STU, 1996. 186 s. ISBN 80-227-0881-X [5] ILKOVIČ, D. *Fyzika 1*. 4.vyd. Bratislava : Alfa, 1969.
- [5] ILKOVIC, D. *Fyzika 1*. 4.vyd. Bratislava : Alfa, 1969.
 496 s.
- [6] HOLMAN, J. P. *Heat transfer*. 10.vyd. New York : McGraw-Hill, 2010. 462 s. ISBN 978-0-07-352936-3.

Vybrané spätnoväzobné koeficienty reaktivity rýchleho plynom chladeného reaktora ALLEGRO

Ivan Pančiak¹, Ing. Amine Bouhaddane, PhD.²

¹ Ústav jadrového a fyzikálneho inžinierstva, FEI STU v Bratislave ² VUJE a.s., Okružná 5, 91864 Trnava

xpanciak@stuba.sk

Abstrakt - V našej práci sa zaoberáme výpočtami Dopplerovho spätnoväzobného koeficientu, void efektu – vnosu reaktivity od straty chladiva a spätnoväzobným efektom reaktivity od teplotnej rozťažnosti palivového pokrytia pri dvoch uvažovaných palivách štartovacej aktívnej zóny reaktora ALLEGRO – UOX (UO₂) a MOX (U,PuO₂) na 2D geometrickom modeli v nekonečnej mreži. Skúmame pri tom prejav vybraných spätnoväzobných efektov pri UOX a MOX palive a vyhodnocujeme, pri ktorom palive je efekt dominantnejší.

1 Úvod

Reaktor ALLEGRO je prototypom rýchleho, héliom chladeného reaktora IV. generácie (GFR). Technológia GFR spája výhody využitia rýchleho neutrónového spektra spoločne s vysokou teplotou chladiva na výstupe (ALLEGRO 530 °C, komerčný GFR 850 °C). Tento typ reaktora môže byť súčasťou uzavretého palivového cyklu, čím prispeje k udržateľnosti jadrovej energetiky v budúcnosti. Vďaka možnosti transmutácie minoritných aktinoidov je možné znížiť dlhodobú rádiotoxicitu vyhoreného jadrového paliva. [1] Ciele projektu ALLEGRO sú overenie technológie GFR a zisk užitočných prevádzkových skúseností s touto technológiou, výskum a vývoj pokročilého karbidového paliva s SiC pokrytím, výskum možností využitia GFR na priemyselné aplikácie a výrobu vodíka vďaka vysokej teplote chladiva na výstupe. Cieľom našej práce je analýza vybraných spätnoväzobných efektov reaktivity v bezúnikovom prostredí - nekonečnej mreži s materiálmi, ktoré budú využité v reaktore ALLEGRO. Takáto analýza má význam pri overení metodiky výpočtu pred vyšetrovaním spätnoväzobných efektov reaktivity na celej AZ v 3D geometrii s uvážením úniku neutrónov.

2 Zloženie aktívnej zóny reaktora ALLEGRO

V reaktore ALLEGRO sú skúmané dve možné konfigurácie štartovacej aktívnej zóny (AZ) – s MOX (U,PuO₂) palivom, v ktorom bude hmotnostný podiel PuO₂ dosahovať 26,374 % [2] a s UOX palivom UO₂ s obohatením 19,5 % ²³⁵U.

Celkový výkon reaktora ALLEGRO bude 75 MWt. AZ reaktora ALLEGRO podľa dizajnu vytvoreného francúzskou organizáciou pre atómovú energiu a alternatívne energie CEA, pozostáva z 81 palivových kaziet, 6 experimentálnych kaziet, ktoré budú slúžiť na testovanie pokročilého karbidového paliva vyvíjaného pre GFR a z regulačného systému reaktora, pozostávajúceho zo 4 kaziet diverzifikovaných bezpečnostných zariadení a 6 kaziet regulačných zariadení. V palivovej kazete je umiestnených 169 palivových prútikov s oceľovým 15-15 Ti pokrytím [2].

Okolo palivovej časti AZ bude rozmiestnených 174 kaziet reflektora usporiadaných v 3 radoch, vyrobených z ocele 15-15 Ti. Po obvode reflektora bude umiestnených 198 tieniacich kaziet usporiadaných v 4 radoch (Obr. 1). Vrstva reflektora a tienenie bude umiestnená aj zvrchu a zospodu palivovej časti AZ [2].



Obr. 1 Aktívna zóna reaktora ALLEGRO [3]

3 Výpočtový kód Serpent

Pri výpočtoch a simuláciách vybraných spätnoväzobných efektov reaktivity sme využili stochastický, Monte Carlo (MC) výpočtový kód Serpent, vyvinutý vo VTT výskumnom centre vo Fínsku. Serpent umožňuje skrátenie potrebného výpočtového času pri zachovaní vysokej presnosti výpočtov v porovnaní s inými MC výpočtovými kódmi [4].

4 Výpočty spätnoväzobných efektov reaktivity v bezúnikovom prostredí

Na základe fyzikálnych vlastností jednotlivých materiálov AZ sa v reaktore prejavujú tzv. spätnoväzobné efekty, kedy dochádza k reakcii na akúkoľvek zmenu teploty, rozmerov či výkonu. Pri výpočte v bezúnikovom prostredí overujeme najmä materiálové vlastnosti, ktoré vplývajú na správanie sa reaktora.

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Bezúnikové prostredie sme vytvorili voľbou odrazovej – reflektívnej okrajovej podmienky vo výpočtovom kóde Serpent. V takom prípade nedochádza k úniku neutrónov zo sledovanej oblasti. V tomto prostredí je model navrhnutý v 2D geometrii. Výpočty v nekonečnej mreži (v bezúnikovom prostredí) slúžia na overenie metodiky výpočtov pred návrhom celej AZ s konkrétnou geometriou. Vstupné údaje o rozmeroch a atómových hustotách nuklidov, potrebné pre vytvorenie 2D modelu, pochádzajú z metodiky ESNII+ [2]. Na obrázku (Obr. 2) je znázornený náš geometrický model palivovej kazety v nekonečnej mreži.



Obr. 2 2D geometrický model kazety ALLEGRO v nekonečnej mreži

4.1 Dopplerov teplotný koeficient

Dopplerov efekt je dominantným spätnoväzobným efektom, ktorý sa prejavuje pri zmene teploty paliva v AZ reaktora. Pri zmene teploty paliva dochádza k zmenám tepelného (Brownovho) pohybu atómov, ktorý sa prejavuje kmitaním (reazonanciami) okolo ich rovnovážnych polôh. Jadrá paliva sú terčovými jadrami pre neutróny. Ak sa terčové jadro a neutrón k sebe približujú v dôsledku kmitavého pohybu jadra, ich energie sa sčítajú. Ak sa naopak, terčové jadro pohybuje v dôsledku kmitavého pohybu rovnakým smerom ako neutrón, ich energie sa odčítajú. Pri zvýšení teploty paliva dochádza k väčším kmitom terčových jadier okolo ich rovnovážnych polôh. Tým sa zvyšuje pravdepodobnosť, že výsledná energia terčového jadra a neutrónu po zrážke dosiahne niektorú energetickú hladinu vzbudenia zloženého jadra, kedy môže dôjsť k rezonančnému zachyteniu neutrónu terčovým jadrom. Pri náraste teploty paliva teda rastie mikroskopický účinný prierez parazitnej absorpcie neutrónu. Takto zachytený neutrón nevyvolá štiepenie jadra, neprispeje tak k udržovaniu sebestačnej štiepnej reťazovej reakcie Error! Reference source not found.. To má za následok zníženie multiplikačného koeficientu, reaktivity a následne teploty paliva v AZ. Týmto spôsobom sa reaktor sám stabilizuje. Naopak, pri poklese výkonu a teploty paliva v reaktore klesá pravdepodobnosť rezonančného zachytenia neutrónu a rastie pravdepodobnosť

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

štiepenia. Následne dochádza k opätovnému zvýšeniu reaktivity, teploty paliva a k stabilizácii reaktora.

Pri výpočte sme zvyšovali teplotu paliva vždy o 150 K oproti predchádzajúcemu stavu a pozorovali zmenu multiplikačného koeficientu, pričom sme nemenili geometrické rozmery v modeli. Pri výpočte Dopplerovho efektu sme zvolili štatistiku 1 000 000 neutrónov v 500 aktívnych a 50 neaktívnych cykloch. Výsledky simulácie sú spracované v nasledujúcej tabuľke (Tab. 1).

T 1	1	с ·	,	1 11	/ v,	D I	1 1	C 1 /
Lan		Nnracowanie	N WWG	ledkov	www.octov	Llonn	lerovho	etektu
rau.	1	Spracovanic		ICUKOV				UTURIU

UOX						
T [K]	\mathbf{k}_{inf}	σ	ρ [pcm]	dρ/dT [1/K]	dp [pcm]	
900	1,37386	9,60E-06	27212		0	
1050	1,37277	1,10E-05	27155	-3,85E-06	-58	
1200	1,37175	9,90E-06	27100	-3,61E-06	-112	
1350	1,37091	9,80E-06	27056	-2,98E-06	-157	
1500	1,37018	9,80E-06	27017	-2,59E-06	-195	
		Ν	MOX			
T [K]	$\mathbf{k}_{\mathrm{inf}}$	σ	ρ [pcm]	dρ/dT [1/K]	dρ [pcm]	
900	1,60268	1,60E-05	37605		0	
1050	1,60167	1,50E-05	37565	-2,62E-06	-39	
1200	1,60082	1,70E-05	37532	-2,21E-06	-72	
1350	1,60007	1,70E-05	37503	-1,95E-06	-102	
1500	1,59937	1,60E-05	37475	-1,82E-06	-129	

Hodnoty k_{inf} a σ sú hodnoty vypočítané kódom Serpent. Reaktivitu ρ [%] počítame podľa vzťahu (1):

$$\rho \left[\text{pcm} \right] = \frac{k_{inf} - 1}{k_{inf}} * 10^5 \tag{1}$$

Hodnotu dp [%] počítame podľa vzťahu (2):

$$d\rho_i \left[pcm\right] = \rho_i(T_i) - \rho_1(T_1) \tag{2}$$

Koeficient dp/dT [%/K] počítame podľa vzťahu (3):

$$d\rho/dT_i [1/K] = \frac{\rho_{i+1} - \rho_i}{T_{i+1} - T_i}$$
(3)

Vyhodnotenie výsledkov je v nasledujúcich grafoch (Obr. 3, Obr. 4, Obr. 5, Obr. 6):

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 3 Graf závislosti kinf od teploty UOX paliva



Obr. 4 Graf závislosti kinf od teploty MOX paliva

Pri MOX aj UOX palive pozorujeme z grafov (Obr. 3, Obr. 4) pokles k_{inf} s narastajúcou teplotou paliva, pričom vidíme, že pri vyšších teplotách paliva pokles k_{inf} nie je taký strmý ako pri nižších teplotách.



Obr. 5 Graf závislosti zmeny reaktivity od teploty

Na Obr. 5 je znázornená zmena reaktivity oproti počiatočnému (nominálnemu prevádzkovému) stavu v závislosti od teploty paliva. Zmeny reaktivity sú záporné, čo znamená, že reaktivita s rastúcou teplotou paliva klesá. Strmší pokles reaktivity pozorujeme pri UOX palive. To znamená, že pri UOX palive je Dopplerov efekt výraznejší ako pri MOX palive.

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika



Obr. 6 Graf závislosti zmeny reaktivity [1/K]

Z Obr. 6 vidíme, že derivácia – strmosť zmeny reaktivity na sledovaných teplotných intervaloch je najmenšia pri nižších teplotách. V týchto teplotných intervaloch dochádza k výraznejšiemu poklesu reaktivity pri zmene teploty. S rastúcou teplotou sa strmosť zmeny reaktivity v závislosti od teploty približuje k nule. To znamená, že zmeny reaktivity pri vyšších teplotách už nie sú také prudké ako pri nižších teplotách.

Pri obidvoch uvažovaných typoch štartovacieho paliva pozorujeme záporný Dopplerov teplotný spätnoväzobný koeficient. Dopplerov efekt sa výraznejšie prejavil v UOX palive ako v MOX palive. To znamená, že spätná väzba systému bude silnejšia pri zmene teploty UOX paliva. Myslíme si, že by to mohlo byť spôsobené množstvom izotopu ²³⁸U v palive. V MOX palive je menej ²³⁸U (64,51 % hm.) [2] ako v UOX palive (70,94 % hm.), teda v UOX palive bude dochádzať k väčšiemu množstvu rezonančných záchytov neutrónu na ²³⁸U, čo bude mať väčší vplyv na zmeny reaktivity. Túto hypotézu je potrebné overiť, resp. vyvrátiť ďalšími výpočtami.

4.2 Void efekt – vnos reaktivity od straty chladiva

Pri výpočte void efektu sme skúmali zmeny reaktivity pri 3 tlakoch chladiva – 0,1 MPa (atmosférický tlak), 3,5 MPa a 7 MPa (nominálny prevádzkový tlak). Výsledky výpočtov sú znázornené v grafoch (Obr. 7, Obr. 8, Obr. 9, Obr. 10). Štatistika pri výpočte void efektu zostala rovnaká ako pri výpočte Dopplerovho efektu.

Výsledky zo simulácie sú spracované v nasledujúcej tabuľke (Tab. 2):

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Tab. 2 Spracovanie výsledkov výpočtov void efektu

UOX						
p [MPa]	$\mathbf{k}_{\mathrm{inf}}$	σ	ρ [pcm]	dρ/dp [1/MPa]	dp [pcm]	
7	1,37175	9,90E-06	27100		0	
3,5	1,37276	9,90E-06	27154	1,53E-04	54	
0,1	1,37328	1,00E-05	27208	1,59E-04	108	
МОХ						
p [MPa]	$\mathbf{k}_{\mathrm{inf}}$	σ	ρ [pcm]	dρ/dp [1/MPa]	dp [pcm]	
7	1,60082	1,70E-05	37532		0	
3,5	1,60325	1,60E-05	37627	2,71E-04	95	
0,1	1,60567	1,60E-05	37721	2,77E-04	189	



Obr. 7 Závislosť kinf od tlaku chladiva pri UOX palive



Obr. 8 Závislosť kinf od tlaku chladiva pri MOX palive

V grafoch závislosti k_{inf} od tlaku chladiva (Obr. 7, Obr. 8**Error! Reference source not found.**) vidíme pri obidvoch typoch uvažovaného paliva – UOX aj MOX nárast k_{inf} pri poklese tlaku.

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika



Z Obr. 9 vidíme, že zmena reaktivity s poklesom tlaku rastie, pričom strmší priebeh je pri MOX palive. Z toho možno usúdiť, že void efekt sa prejaví výraznejšie pri využití MOX paliva.



Obr. 10 Graf závislosti zmeny reaktivity vztiahnutej na 1 MPa [%/MPa]

Na Obr. 10 pozorujeme strmosť zmeny reaktivity – deriváciu v závislosti od tlaku. Prvý vykreslený bod v grafe symbolizuje interval od 7 do 3,5 MPa a druhý vykreslený bod interval od 3,5 do 0,1 MPa. Pri oboch typoch paliva vidíme, že strmosť zmeny reaktivity s poklesom tlaku mierne rastie.

Pri obidvoch typoch paliva pozorujeme pozitívny void efekt. Pri vyšetrovaní void efektu sme sledovali veľké zmeny tlaku. Myslíme si, že pozitívny charakter void efektu je spôsobený moderáciou neutrónov na héliu. Hélium sa vyznačuje nízkym mikroskopickým účinným prierezom rozptylu ($\sigma_s = 0.79$ barn) [6]. Neutrón sa bude na l'ahkom jadre hélia rozptyľovať pružným potenciálovým rozptylom, pričom odovzdá časť svojej kinetickej energie a spomalí sa. Pri strate chladiva koncentrácia hélia klesne, čím sa zníži aj miera moderácie neutrónov héliom. Neutrónové spektrum sa týmto posunie do vyšších energií. To môže spôsobiť nárast mikroskopického účinného prierezu štiepenia 238U (a ďalších štiepiteľných izotopov - ²⁴⁰Pu, ²⁴²Pu v MOX palive), kvôli čomu rastie kinf. Túto hypotézu však bude nutné overiť, resp. vyvrátiť ďalšími výpočtami už na modeli AZ reaktora ALLEGRO s konkrétnou 3D geometriou, keďže na objektívne zhodnotenie void efektu je potrebné uvážiť aj únik neutrónov zo systému konečných rozmerov.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

4.3 Spätnoväzobný efekt reaktivity od teplotnej rozťažnosti palivového pokrytia

V reaktore dochádza vplyvom zmeny teploty aj k zmenám rozmerov konštrukčných materiálov v dôsledku teplotnej rozťažnosti. Palivové pokrytie je dôležitou bezpečnostnou bariérou pred únikom produktov štiepenia. Je navrhnuté tak, aby malo minimálny vplyv na neutrónovo-fyzikálne vlastnosti reaktora. V reaktore ALLEGRO bude v UOX a MOX štartovacej zóne vyrobené pokrytie paliva z 15-15 Ti ocele AIM1 [2]. Našim cieľom je zistiť, ako bude radiálna tepelná rozťažnosť palivového pokrytia vplývať na reaktivitu v MOX a UOX aktívnej zóne.

Pri výpočte pozorujeme zmeny rozmerov palivového pokrytia pri rôznych teplotách v 300 stupňových intervaloch. Z metodiky projektu ESNII+ [2] sú známe geometrické rozmery palivového pokrytia v studenom stave (pri teplote 20 °C). Je nutné tieto rozmery prepočítať pri našich sledovaných teplotách, počínajúc nominálnou prevádzkovou teplotou 447,5 °C [2]. Pri zmene rozmerov akéhokoľvek materiálu sa zmenia aj atómové hustoty, t.j. počet atómov v jednotke objemu. Je preto nutné okrem rozmerov prepočítať aj atómové hustoty jednotlivých izotopov stavebných prvkov pokrytia pri sledovaných teplotách. Postupovali sme nasledovne: Najskôr sme vypočítali koeficient teplotnej dĺžkovej rozťažnosti α [1/°C] pri sledovaných teplotách podľa vzťahu (4) [2]:

$$\alpha = 0,0000178 + (T - 400) * \\ * \frac{0,0000182 - 0,0000178}{100}$$
(4)

Kde T [°C] je sledovaná teplota palivového pokrytia. Ďalej vypočítame nové geometrické rozmery – vnútorný a vonkajší polomer pokrytia podľa vzťahu (5):

$$R(T) = R(T_0) * [1 + \alpha * (T - T_0)]$$
⁽⁵⁾

Kde R(T) [cm] je polomer pokrytia pri sledovanej teplote, R(T₀) [cm] je polomer pokrytia za studena, t.j. pri teplote 20 °C. Po zmene geometrických rozmerov sme prostredníctvom kódu Serpent zistili objemy palivového pokrytia v sledovaných stavoch, ktoré sme porovnali s objemom pokrytia za studena, čím sme získali relatívnu objemovú zmenu palivového pokrytia V_r podľa vzťahu (6):

$$V_r = \frac{V(T)}{V_0} \tag{6}$$

Kde V(T) [cm³] je objem pri danej teplote, V₀ [cm³] je objem pokrytia za studena. Nakoniec vypočítame zmenu atómovej hustoty konkrétneho izotopu stavebného prvku n_j [atom/cm³]podľa vzťahu (7):

$$n_j = \frac{n_0}{V_r} \tag{7}$$

Kde n_0 [atom/cm³] je atómová hustota daného izotopu vo východiskovom stave – za studena a V_r je relatívna zmena objemu palivového pokrytia.

Výsledky výpočtov sú uvedené v grafoch (Obr. 11, Obr. 12, Obr. 13, Obr. 14) a tabuľke (Tab. 3).

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Štatistika zostala rovnaká ako pri predošlých výpočtoch – 1000000 neutrónov v 500 aktívnych a 50 neaktívnych cykloch.

Tab. 3 Spracova	anie výsledł	kov výpočtov	spätnoväzol	oného
efektu reaktivit	y od tepelne	j rozťažnosti	palivového	pokrytia

UOX						
V [cm ³]	T [°C]	k _{inf}	σ	ρ [pcm]	dp [pcm]	dρ/dV [1/cm ³]
138,01	447,5	1,37176	1,0E-05	27101	0	
139,14	747,5	1,37184	1,0E-05	27105	4	3,76E-05
140,36	1047,5	1,3719	1,0E-05	27108	7	2,61E-05
141,67	1347,5	1,37195	9,2E-06	27111	10	2,03E-05
143,08	1647,5	1,37203	1,0E-05	27115	14	3,01E-05
144,58	1947,5	1,37212	9,4E-06	27120	19	3,19E-05
МОХ						
V				ρ	dρ	dp/dV
$[cm^3]$	T [°C]	\mathbf{k}_{inf}	σ	[pcm]	[pcm]	[1/cm3]
138,01	447,5	1,6009	1,5E-05	37535	0	
139,14	747,5	1,60105	1,5E-05	37541	6	5,18E-05
140,36	1047,5	1,60114	1,6E-05	37544	9	2,88E-05
141,67	1347,5	1,6013	1,5E-05	37551	16	4,76E-05
143,08	1647,5	1,60143	1,5E-05	37556	21	3,60E-05
144,58	1947,5	1,60161	1,7E-05	37563	28	4,68E-05



Obr. 11 Graf závislosti kinf od objemu palivového pokrytia pri UOX palive



Obr. 12 Graf závislosti kinf od objemu pokrytia pri MOX palive

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

Sekcia 2. Jadrové a fyzikálne inžinierstvo, Elektronika a fotonika

Na Obr. 11 a Obr. 12 vidíme, že aj pri MOX aj pri UOX palive rastie k_{inf} so zväčšujúcim sa objemom.



Obr. 13 Graf závislosti zmeny reaktivity od objemu pokrytia

Z Obr. 13 pozorujeme nárast reaktivity so zväčšujúcim sa objemom pokrytia pri UOX aj pri MOX palive. Krivka pri MOX palive má strmší priebeh, z čoho môžeme usúdiť, že daný efekt sa prejaví výraznejšie práve pri využití MOX paliva.



Obr. 14 Graf závislosti zmeny reaktivity [1/cm³] od objemu pokrytia [cm³]

Pri skúmaní spätnoväzobného efektu od radiálnej teplotnej rozťažnosti palivového pokrytia sme z výpočtov zistili, že reaktivita narastá vplyvom rozmerových zmien na palivovom pokrytí. Tento efekt sa výraznejšie prejaví pri využití MOX paliva. Myslíme si, že kladný trend tohto efektu by mohol byť spôsobený vytesnením chladiva zo systému vplyvom rozťažnosti palivového pokrytia. Tým, že rastú rozmery palivového pokrytia, klesá objem, v ktorom sa nachádza chladivo. To znamená, že vzdialenosť medzi palivovými prútikmi sa zmenší, čo môže spôsobiť nárast k_{inf}. Túto hypotézu je však potrebné overiť ďalšími výpočtami na 3D modeli AZ reaktora ALLEGRO, kedy musíme uvažovať aj s únikom neutrónov zo systému.

5 Záver

Pri výpočtoch v programe Serpent sme zistili vplyv vybraných spätnoväzobných efektov na reaktivitu pri použití MOX a UOX hnacej AZ.

Dopplerov teplotný koeficient vyšiel záporný aj pri MOX aj pri UOX palive. Výraznejšie sa prejavil pri použití UOX paliva. Myslíme si, že je to spôsobené rozdielnym obsahom izotopu ²³⁸U v palive. V UOX palive je väčšie množstvo ²³⁸U (70,94 % hm.) ako v MOX palive (64,51 % hm.) Z tohto dôvodu mohlo dôjsť k väčšej miere rezonančných záchytov neutrónov izotopov ²³⁸U v UOX palive, čo malo väčší vplyv na zmeny k_{inf}. Túto hypotézu je potrebné overiť ďalšími výpočtami a objektívne vyšetriť Dopplerov efekt na 3D modeli AZ reaktora ALLEGRO s uvážením úniku neutrónov.

Void efekt - vnos reaktivity od straty chladiva vyšiel pri oboch uvažovaných hnacích zónach kladný, t.j. reaktivita rastie pri poklese tlaku chladiva. Výraznejšie sa void efekt prejavil pri použití MOX paliva. Myslíme si, že to môže byť spôsobené moderáciou neutrónov héliom. Neutróny sa na ľahkých jadrách rozptyľujú pružným potenciálovým rozptylom hélia s mikroskopickým účinným prierezom $\sigma_s = 0,79$ barn [6]. Pri tom strácajú svoju energiu a dochádza k moderácii. Pri strate chladiva sa koncentrácia hélia zmenší, čím sa zmenší aj moderácia neutrónov. Spektrum neutrónov sa vtedy posunie do vyšších energií, čo má za následok nárast mikroskopického účinného prierezu štiepenia štiepiteľných izotopov (²³⁸U, ²⁴⁰Pu, ²⁴²Pu), čo by následne zvýšilo reaktivitu. Túto hypotézu je potrebné overiť ďalšou analýzou a vyšetrovať void efekt už na 3D modeli AZ s uvážením úniku neutrónov z prostredia.

Spätnoväzobný teplotný efekt od teplotnej rozťažnosti palivového pokrytia je pozitívny pri oboch uvažovaných palivách. Dominantnejšie sa tento efekt prejavil pri MOX palive. Rastúci trend tohto efektu by mohol byť spôsobený vytesnením chladiva (zmenšením objemu, v ktorom sa nachádza) pri náraste rozmerov pokrytia paliva, pričom dôjde aj k zmenšeniu vzdialenosti medzi prútikmi, čo zvýši reaktivitu. Túto hypotézu je nutné potvrdiť (resp. vyvrátiť) ďalšími analýzami a výpočtami na 3D modeli s uvážením úniku neutrónov.

Výpočty boli realizované v nekonečnej mreži, t.j. bez úniku neutrónov. Na objektívne vyšetrenie spätnoväzobných koeficientov reaktivity je nutné ich vyšetriť na 3D modeli, s uvážením úniku neutrónov zo systému. Cieľom do budúcna je porovnanie týchto výpočtov s výpočtami uskutočnenými na 3D návrhu celej AZ ALLEGRA a skúmanie vplyvu úniku neutrónov na tieto spätnoväzobné efekty. Rovnako je naším cieľom aj potvrdiť (resp. vyvrátiť) vyslovené hypotézy.

6 Literatúra

- [1] IRSN. Review of Generation IV Nuclear Energy Systems, April 2015
- [2] TEMESVARI, E. ALLEGRO Core Specifications, in ESNII+ Deliverable D6.1.1-2, MTA EK, 30.10.2014
- [3] SLUGEŇ, V. et al. ALLEGRO demonštrátor rýchleho plynom chladeného reaktora IV. generácie. STU, Júl 2014, p.70
- [4] LEPPÄNEN, J. et al (2015). The Serpent Monte Carlo code: Status, development and applications in 2013. Ann. Nucl. Energy, 82 (2015), p. 142-150
- [5] REUSS, P. Neutron Physics. EDP Sciences, 2008, ISBN 978-2-7598-0041-4
- [6] MUGHABGHIB, S.F. Atlas of Neutron Resonancies. ORNL, 2018, ISBN 978-0-4445-2035-7

Sekcia 3

Robotika a kybernetika

Aplikácia pre návrh diskrétneho stavového riadenia

Kristína Okienková, Jana Paulusová

Ústav robotiky a kybernetiky, FEI STU v Bratislave

xokienkova@stuba.sk

Abstrakt – Práca sa zaoberá návrhom diskrétneho stavového riadenia pre vybrané modely riadeného systému. Aplikácia je vytvorená v prostredí programu MATLAB a poskytuje používateľovi možnosť linearizácie nelineárneho modelu chemického reaktora v ľubovoľne zvolenom pracovnom bode, návrh riadenia pre vybrané metódy diskrétneho stavového riadenia, a následne porovnanie navrhnutých riadení. V práci je opísaný teoretický návrh riadenia a opis funkcionalít aplikácie.

1 Úvod

Diskrétne riadenie zohráva dôležitú úlohu v oblasti automatizácie a riadenia od chvíle, kedy úlohu regulátorov prevzali počítače pracujúce v určitých periodicky sa opakujúcich momentoch. Stavová oblasť, v ktorej sa navrhuje riadenie, predstavuje širšiu informáciu o riadenom procese, pretože stavové veličiny poskytujú úplný obraz o vnútornej dynamike riadeného procesu, čím skvalitňujú navrhnuté riadenie.

Cieľom práce je vytvorenie vhodnej aplikácie pre návrh diskrétneho stavového riadenia. V rámci aplikácie je realizovaných 5 typov diskrétneho stavového riadenia.

Práca sa zaoberá teoretickým opisom návrhu spomínaných typov diskrétneho stavového riadenia. Následne je podrobne opísaná aplikácia s funkcionalitami, ktoré poskytuje používateľovi. Aplikácia je rozdelená do 3 základných okien, a to linearizácia Van de Vusseovej reakcie, stavového diskrétneho riadenia a porovnanie návrh navrhnutého riadenia. Používateľ aplikácie má možnosť si zvoliť pracovný bod, v ktorom sa bude linearizovať model chemického reaktora. Následne môže pre linearizovaný systém navrhnúť diskrétne stavové riadenie, ktoré sa overí aj na nelineárnom modeli daného systému. Aplikácia používateľovi ponúka informácie o vlastnostiach linearizovaného systému, stabilite systému a kvalite regulácie. Okrem iného poskytuje používateľovi možnosť porovnať navrhnuté riadenia.

2 Návrh diskrétneho stavového riadenia

V danej kapitole opíšeme vybrané metódy návrhu diskrétneho stavového riadenia, ktoré je následne použité v aplikácii.

2.1 Návrh riadenia so stavovou spätnou väzbou

Riadenie so stavovou spätnou väzbou je riadenie založené na znalosti stavov riadeného systému, tzv. spätnoväzbové riadenie [1]. Diskrétny stavový opis systému je definovaný nasledovne

$$\mathbf{x}(k+1) = \mathbf{A}_{d}\mathbf{x}(k) + \mathbf{B}_{d}\mathbf{u}(k)$$
(1)

$$\mathbf{y}(k) = \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k)$$

kde x je stavový vektor rozmeru n, u je vstupný vektor rozmeru r, y je výstupný vektor rozmeru m, A_d je diskrétna matica systému s rozmermi $(n \times n)$, B_d je diskrétna matica vstupov s rozmermi $(n \times r)$, matica C je matica výstupov s rozmermi $(m \times n)$ a matica D je matica pôsobenia vstupu na výstup s rozmermi $(m \times r)$ [2].

K diskrétnemu stavovému opisu systému (1) sa pridá riadenie u(k) definované nasledovne

$$\boldsymbol{u}(k) = -\boldsymbol{K}_{\boldsymbol{C}}\boldsymbol{x}(k) \tag{2}$$

kde matica K_C s rozmermi ($m \times n$) predstavuje proporcionálne zosilnenie stavov $\mathbf{x}(k)$ v zápornej spätnej väzbe. Po dosadení rovnice riadenia (2) do stavového opisu (1) vyjadríme diskrétny stavový opis uzavretého regulačného obvodu nasledovne

$$\mathbf{x}(k+1) = (\mathbf{A}_d - \mathbf{B}_d \mathbf{K}_c) \mathbf{x}(k) + \mathbf{B}_d \widetilde{\mathbf{w}}(k)$$
(3)

$$\mathbf{y}(k) = (\mathbf{C} - \mathbf{D} \mathbf{K}_c) \mathbf{x}(k) + \mathbf{D} \widetilde{\mathbf{w}}(k)$$

kde $\tilde{w}(k)$ predstavuje posun na žiadanú veličinu w(k), pretože riadenie so stavovou väzbou nezabezpečí sledovanie žiadanej veličiny w(k) výstupnou veličinou y(k) [3]. To znamená, že takéto riadenie neodstráni trvalú regulačnú odchýlku e(k), ale zabezpečí riadenie stavov x(k) na nulovú hodnotu, resp. na ustálené hodnoty stavov systému v pracovnom bode po vychýlení systému z týchto stavov.

Maticu zosilnení K_C môžeme získať pomocou dvoch metód, a to metódou Pole Placement alebo pomocou LQ kritéria.

2.1.1 Metóda Pole Placement

Metóda Pole Placement je metóda umiestnenia pólov uzavretého regulačného obvodu (URO). Póly URO sú vlastné čísla matice dynamiky uzavretého regulačného obvodu A_c , ktorá je definovaná nasledovne

$$A_{\mathcal{C}} = (A_d - B_d K_{\mathcal{C}}) \tag{4}$$

Vlastné čísla matice A_c vyjadríme pomocou determinantu z nasledujúcej charakteristickej rovnice URO

$$\det(zI - (A_d - B_dK_c)) = 0$$
⁽⁵⁾

kde I je jednotková matica s rozmermi ($n \times n$).

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Pri návrhu riadenia je dôležité, aby URO bol stabilný, a teda bola splnená podmienka stability v diskrétnej oblasti, ktorá hovorí, že systém je stabilný, ak všetky póly systému sa nachádzajú v jednotkovej kružnici [4].

Hodnoty matice zosilnení K_C vieme vypočítať porovnaním členov pri rovnakých mocninách operátora Z-transformácie charakteristickej rovnice URO a želaného polynómu m(z), ktorý reprezentuje želané umiestnenie pólov URO v komplexnej rovine. Druhou možnosťou je použitie Ackermannovho vzorca, ktorý je definovaný nasledovne [5]

$$\boldsymbol{K}_{\boldsymbol{C}} = \boldsymbol{e}_{\boldsymbol{n}}^{\mathrm{T}} \boldsymbol{Q}_{\boldsymbol{C}}^{-1} \boldsymbol{m}(\boldsymbol{A}_{\boldsymbol{d}}) \tag{6}$$

 $(\cap$

kde $e_n^T = [0, 0, ..., 0, 1]$, Q_C je matica riaditeľnosti a $m(A_d)$ je želaný charakteristický polynóm URO m(z), do ktorého je matica A_d dosadená za operátor Z-transformácie.

2.1.2 LQ kritérium

Riadenie s použitím LQ kritéria je tzv. LQ riadenie (lineárne kvadratické riadenie), ktoré vychádza z minimalizácie kvadratického kritéria, ktoré je definované nasledovne [6]

$$J(\boldsymbol{x_0}) = \sum_{k=0}^{\infty} \boldsymbol{x}(k)^T \boldsymbol{Q} \boldsymbol{x}(k) + \boldsymbol{u}(k)^T \boldsymbol{R} \boldsymbol{u}(k)$$
⁽⁷⁾

kde x_0 je začiatočná podmienka stavového vektora x(k), matica $Q \ge 0$ a matica $R \ge 0$.

Maticu zosilnení K_C vypočítame použitím nasledujúceho vzťahu

$$K_{c} = (\boldsymbol{B}_{d}^{T} \boldsymbol{P} \boldsymbol{B}_{d} + \boldsymbol{R})^{-1} \boldsymbol{B}_{d}^{T} \boldsymbol{P} \boldsymbol{A}_{d}$$
⁽⁸⁾

kde **P** je pozitívne definitná symetrická matica, ktorú vypočítame z Riccatiho rovnice v nasledujúcom tvare [6]

$$0 = -\mathbf{P} + \mathbf{A}_{d}^{T}\mathbf{P}\mathbf{A}_{d} -$$
(9)
$$-\mathbf{A}_{d}^{T}\mathbf{P}\mathbf{B}_{d}(\mathbf{B}_{d}^{T}\mathbf{P}\mathbf{B}_{d} + \mathbf{R})^{-1}\mathbf{B}_{d}^{T}\mathbf{P}\mathbf{A}_{d} + \mathbf{Q}$$

kde matica Q s rozmermi $(n \times n)$ a matica R s rozmermi $(r \times r)$ sú voliteľné.

2.2 Návrh dopredného stavového riadenia

Dopredné stavové riadenie výrazne zmenší trvalú regulačnú odchýlku e(k) a zabezpečí ustálenie výstupnej veličiny y(k) na žiadanej hodnote w(k), pokiaľ poznáme riadený systém.

Dopredné stavové riadenie sa skladá z doprednej časti regulátora a zápornej stavovej spätnej väzby a je definované nasledovne

$$u(k) = -K_{c}x(k) + K_{w}w(k)$$
⁽¹⁰⁾

kde K_w je zosilnenie doprednej časti regulátora.

Dosadením rovnice riadenia (10) do diskrétneho stavového opisu (1) vyjadríme zosilnenie doprednej časti regulátora v nasledujúcom tvare

$$K_w = \frac{1}{\boldsymbol{C}(\boldsymbol{I} - \boldsymbol{A}_c)^{-1}\boldsymbol{B}_d}$$
(11)

kde matica I je jednotková matica s rozmermi $(n \times n)$ a matica $A_C = A_d - B_d K_C$.

Sekcia 3. Robotika a kybernetika

2.3 Návrh riadenia so stavovou spätnou väzbou a integračnou zložkou

Riadenie so stavovou spätnou väzbou a integračnou zložkou zabezpečí dosiahnutie žiadanej veličiny w(k) bez trvalej regulačnej odchýlky e(k).

Pre zabezpečenie ustálenia sa výstupnej veličiny y(k) na žiadanej veličine w(k) potrebujeme pridať integrátor regulačnej odchýlky pre každý výstup systému, pričom regulačná odchýlka systému je definovaná v nasledujúcom tvare

$$\boldsymbol{e}(k) = \boldsymbol{w}(k) - \boldsymbol{y}(k) \tag{12}$$

kde *e* je vektor regulačnej odchýlky rozmeru *m*.

Integračné stavy výstupu sú potom definované nasledovne [6]

$$\begin{aligned} \boldsymbol{x}_i(k+1) &= \boldsymbol{x}_i(k) + T_s \boldsymbol{e}(k) = \\ &= \boldsymbol{x}_i(k) + T_s \boldsymbol{w}(k) - T_s \boldsymbol{C} \boldsymbol{x}(k) \end{aligned}$$
(13)

kde T_s je perióda vzorkovania. Pridaním integračných stavov výstupu $x_i(k)$ je riadený systém rozšírený nasledovne

$$\begin{bmatrix} \boldsymbol{x}(k+1) \\ \boldsymbol{x}_{i}(k+1) \end{bmatrix} = \begin{bmatrix} \boldsymbol{A}_{d} & 0 \\ -T_{s}\boldsymbol{C} & \boldsymbol{I} \end{bmatrix} \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{x}_{i}(k) \end{bmatrix} + \begin{bmatrix} \boldsymbol{B}_{d} \\ 0 \end{bmatrix} \boldsymbol{u}(k) + \\ + \begin{bmatrix} 0 \\ T_{s}\boldsymbol{I} \end{bmatrix} \boldsymbol{w}(k)$$
(14)

kde I je jednotková matica s rozmermi ($m \times n$). Rovnicu (14) môžeme zapísať aj v nasledujúcom tvare

$$\widetilde{\mathbf{x}}(k+1) = \widetilde{\mathbf{A}}_{d}\widetilde{\mathbf{x}}(k) + \widetilde{\mathbf{B}}_{d}\mathbf{u}(k)$$
⁽¹⁵⁾

(1 5)

kde \tilde{x} je rozšírený stavový vektor a \tilde{A}_d , \tilde{B}_d sú rozšírené matice, ktoré získame porovnaním vzťahov (14) a (15).

Riadenie s integračnou zložkou a stavovou spätnou väzbou predstavuje PI regulátor, ktorý je definovaný v nasledujúcom tvare

$$\boldsymbol{u}(k) = -[\boldsymbol{K}_{\boldsymbol{c}} \quad \boldsymbol{K}_{\boldsymbol{i}}] \begin{bmatrix} \boldsymbol{x}(k) \\ \boldsymbol{x}_{\boldsymbol{i}}(k) \end{bmatrix} =$$

$$= -\boldsymbol{K}_{\boldsymbol{c}} \boldsymbol{x}(k) - \boldsymbol{K}_{\boldsymbol{i}} \boldsymbol{x}_{\boldsymbol{i}}(k)$$
(16)

kde K_i je matica zosilnenia integračnej zložky. Pomocou Ztransformácie rovnice (13) a jej následnou úpravou môžeme integračné stavy výstupu prepísať v rovnici riadenia (16) nasledovne

$$\boldsymbol{u}(k) = -\boldsymbol{K}_{c}\boldsymbol{x}(k) - \boldsymbol{K}_{i}\frac{T_{s}z^{-1}}{1 - z^{-1}}\boldsymbol{e}(k)$$
(17)

kde T_s je perióda vzorkovania.

2.4 Návrh dopredného stavového riadenia s integračnou zložkou

Integračná zložka spôsobuje oneskorenie regulácie, ktoré sa prejaví na priebehoch riadenia oneskorením o jeden krok periódy vzorkovania T_s . Pridaním doprednej časti regulátora do riadenia sa odstráni oneskorenie regulácie.

Dopredné stavové riadenie s integračnou zložkou je definované nasledovne

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

$$u(k) = -K_c \mathbf{x}(k) - K_i \frac{T_s z^{-1}}{1 - z^{-1}} e(k) + K_w w(k)$$
⁽¹⁸⁾

kde zosilnenie doprednej časti regulátora K_w vypočítame pomocou rovnice (11).

2.5 Návrh stavového riadenia s anti-windupom

Windup efekt sa zvykne objavovať pri obmedzení akčného zásahu u(k), kedy akčný zásah dosiahne saturáciu a integrátor začne integrovať až po opustení saturácie, čo vnesie do regulácie oneskorenie [7]. Na odstránenie oneskorenia sa zavedie do regulačného obvodu anti-windup, ktorý zabezpečí okamžitú reakciu integrátora na dosiahnutie saturácie akčnej veličiny u(k), a tak odstráni windup efekt.

Uvažujme diskrétny stavový opis systému s riadením v nasledujúcom tvare [6]

$$\begin{aligned} \boldsymbol{x}_{c}(k+1) &= \boldsymbol{A}_{c}\boldsymbol{x}_{c}(k) + \boldsymbol{B}_{c}\boldsymbol{z}(k) \\ \boldsymbol{u}(k) &= \boldsymbol{C}_{c}\boldsymbol{x}_{c}(k) + \boldsymbol{D}_{c}\boldsymbol{z}(k) \end{aligned} \tag{19}$$

kde matica A_c je nestabilná, akčný zásah u(k) je saturovaný a z(k) predstavuje regulačnú odchýlku. Akčný zásah u(k), keďže je saturovaný, zadefinujeme ho nasledovne

$$u(k) = sat(v(k))$$
(20)
$$v(k) = C_c x_c(k) + D_c z(k)$$

kde v(k) predstavuje nesaturovaný akčný zásah.

Pre odstránenie nestability URO, pridáme anti-windup do riadenia nasledovne

$$\begin{aligned} \boldsymbol{x}_{c}(k+1) &= \boldsymbol{A}_{c}\boldsymbol{x}_{c}(k) + \boldsymbol{B}_{c}\boldsymbol{z}(k) + \\ &+ \boldsymbol{K}_{A}\big(\boldsymbol{u}(k) - \boldsymbol{v}(k)\big) \end{aligned}$$
(21)

kde K_A je zosilnenie anti-windupu, ktoré je voliteľné a aktívne iba počas saturácie. Rovnicu (21) vyjadrením jednotlivých veličín upravíme do nasledujúceho tvaru

$$\begin{aligned} \mathbf{x}_{c}(k+1) &= (\mathbf{A}_{c} - \mathbf{K}_{A}\mathbf{C}_{c})\mathbf{x}_{c}(k) + \\ &+ (\mathbf{B}_{c} - \mathbf{K}_{A}\mathbf{D}_{c})\mathbf{z}(k) + \mathbf{K}_{A}\mathbf{u}(k) \end{aligned}$$
(22)

ktorého veličiny pri porovnaní s rovnicami (13) a (16) sú nasledujúce

$$\begin{aligned} \mathbf{x}_{c} &= \mathbf{x}_{i} & \mathbf{B}_{c} &= [T_{s}\mathbf{I} \quad -T_{s}\mathbf{C}] \\ \mathbf{A}_{c} &= \mathbf{I} & \mathbf{D}_{c} &= [\mathbf{0} \quad -\mathbf{K}_{c}] \\ \mathbf{C}_{c} &= -\mathbf{K}_{i} & \mathbf{z} &= \begin{bmatrix} \mathbf{W} \\ \mathbf{x} \end{bmatrix} \end{aligned}$$
 (23)

kde I je jednotková matica, K_i je matica zosilnení integračnej zložky a K_c je matica zosilnení stavovej spätnej väzby.

3 Aplikácia pre návrh riadenia

Vytvorená aplikácia je naprogramovaná v užívateľskom prostredí programu MATLAB App Designer. Tento program poskytuje používateľovi možnosť vytvoriť aplikáciu jednoducho a efektívne s využitím funkcií, ktoré ponúka prostredie MATLAB.

Grafické rozhranie programu sa skladá z dvoch hlavných častí – grafická a programová. V grafickej časti si používateľ zvolí objekty a ich rozmiestnenie v rámci vytváranej aplikácie. Program ponúka rôzne interaktívne objekty na vkladanie a zobrazovanie údajov, ako napr. tlačidlá, grafy, textové polia alebo tabuľky. Výhodou programu je automatické vkladanie programového kódu grafických komponentov do programovej časti. V tejto časti má možnosť používateľ naprogramovať kompletnú funkcionalitu danej aplikácie, pričom App Designer uľahčuje programovanie pomocou predpripravených funkcií k daným objektom. Ďalšou výhodou je vzájomné prepojenie týchto dvoch častí. Keď používateľ vykoná zmenu v grafickej časti, zmena sa prejaví aj v programovej a naopak. Okrem vytvárania nezávislých aplikácii, program MATLAB poskytuje aj prepojenie aplikácie s iným podprogramom ako je napríklad Simulink, ktorý môže vykonávať operácie na pozadí aplikácie [8].

Aplikácia sa skladá z troch základných okien:

- linearizácia Van de Vusseovej reakcie,
- návrh diskrétneho stavového riadenia,
- porovnanie navrhnutého riadenia.

Po spustení aplikácie sa zobrazí okno - Linerizácia Van de Vusseovej reakcie (Obr. 1), v ktorom si používateľ môže zvoliť pracovný bod F_0 , v ktorom sa bude linearizovať Van de Vusseova reakcia prebiehajúca v chemickom reaktore. Okrem pracovného bodu si používateľ môže zvoliť minimálnu a maximálnu hodnotu vstupného prietoku (Fmin, Fmax). Spomínané vstupné hodnoty sa nachádzajú na paneli s názvom Vstupný prietok, ktorý je podfarbený žltou farbou signalizujúcou interakciu používateľa s aplikáciou. Aplikácia je naprogramovaná tak, aby po zmenení vstupných údajov aktualizovala všetky údaje v aplikácii. Pri spustení aplikácie sa načítajú inicializačné hodnoty všetkých vstupných premenných.

Vedľa spomínaného panela sa nachádza panel s názvom *Pracovný bod*, v ktorom sa vypisujú ustálené hodnoty vybraného pracovného bodu, a to vstupný prietok F_0 , koncentrácia vstupnej suroviny C_{A0} a koncentrácia žiadaného produktu C_{B0} .

Pod panelmi sa zobrazuje obrázok predstavujúci prietokový izotermický chemický reaktor tzv. Continous Stirred Tank Reactor (CSTR), v ktorom prebieha exotermická Van de Vusseova reakcia. Do reaktora priteká vstupná surovina A (cyklopentadién) a za dokonalého miešania vplyvom katalyzátora zreaguje na žiadaný produkt B (cyklopentenol). Pri reakcii vzniká aj nežiadúci produkt C (cyklopenténdiol) a prebieha nežiadúca paralelná reakcia, ktorej výsledný produkt je D (dicyklopentándiol) [9]. Pod obrázkom reaktora sa nachádza nelineárny model systému, ktorý je definovaný rovnicami hmotnostnej bilancie pre koncentrácie vstupnej suroviny C_A a žiadaného produktu C_B . Nižšie je uvedená tabuľka s kinematickými parametrami rovníc.

V prvom okne sa ďalej nachádza graf s prevodovou charakteristikou, na ktorej sa zobrazuje zvolený pracovný bod. Pod grafom je umiestnená skupina prepínajúcich sa panelov s vlastnosťami linearizovaného systému. V nadradenom paneli s názvom *Spojitá oblasť* sa nachádza panel s názvom *Stavový opis a prenosová funkcia*, v ktorom sa zobrazuje stavový opis s hodnotami matíc A, B, C, D a prenosová funkcia G(s) linearizovaného systému v okolí pracovného bodu v spojitej oblasti.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 1. Užívateľské grafické prostredie aplikácie – Linearizácia Van de Vusseovej reakcie

Ďalej sa nachádza podradený panel s názvom Vlastnosti *systému*, v ktorom sa vypisujú hodnoty časových konštánt (T_1 , T_2 , τ), zosilnenia K, pólov a núl systému (p_1 , p_2 , n). Taktiež sa zobrazujú informácie o stabilite systému, ráde systému, kauzalite, fázovosti a periodicite systému. V ďalšom nadradenom paneli s názvom Diskrétna oblasť sa nachádza panel s názvom Stavový opis, v ktorom si používateľ môže zvoliť periódu vzorkovania Ts zo zadaného intervalu, a následne sa vypočítajú matice diskrétneho stavového opisu Ad, Bd, C, D linearizovaného systému v okolí pracovného bodu. V ďalšom podradenom paneli s názvom Prenosová *funkcia* sa vypisuje diskrétna prenosová funkcia G(z), hodnoty časových konštánt (T_1 , T_2 , τ), zosilnenia K, pólov a núl systému (p_1, p_2, n) v diskretnej oblasti, a taktiež stabilita systému. V poslednom podriadenom paneli s názvom Riaditeľnosť a pozorovateľnosť sa vypisuje, či je systém a pozorovateľný, a zároveň hodnoty riaditeľný matíc riaditeľnosti Q_C a pozorovateľnosti Q_O .

V ďalšej časti prvého okna sa zobrazuje prechodová charakteristika ako priebeh vybranej stavovej veličiny (x_1 , x_2) pre zvolený pracovný bod. Používateľ má možnosť výberu hodnoty času simulácie a hodnoty delta, čo je veľkosť skoku simulovanej veličiny. Taktiež si môže zvoliť priebehy, ktoré sa budú zobrazovať na grafe, a to priebeh stavovej veličiny pre nelineárny model, linearizovaný spojitý model, linearizovaný diskrétny model alebo ustálené hodnoty pracovného bodu.

Nakoniec si môže používateľ zvoliť pre akú oblasť (spojitá/diskrétna) sa bude vykresľovať graf s pólmi a nulami systému v komplexnej rovine.

Druhé okno s názvom *Návrh diskrétneho stavového riadenia* (Obr. 2) umožňuje používateľovi navrhnúť parametre riadenia pre vybrané metódy, ktorých teoretický návrh riadenia je opísaný v predchádzajúcej kapitole.

Používateľ môže navrhnúť riadenie pre nasledujúce typy riadenia:

- 1. Riadenie so stavovou spätnou väzbou
- 2. Dopredné stavové riadenie
- 3. Riadenie so stavovou spätnou väzbou a integračnou zložkou
- 4. Dopredné stavové riadenie s integračnou zložkou
- 5. Stavové riadenie s anti-windupom

Výber riadenia sa nachádza na paneli s názvom Výber diskrétneho stavového riadenia. Na základe zvoleného riadenia sa zobrazuje vo vedľajšom paneli dané riadenie s objektami pre zadávanie parametrov riadenia. Používateľ má možnosť výberu metódy pre vybrané riadenie, a to Pole Placement a LQ kritérium. Pokial' si používateľ vyberie metódu Pole Placement, na paneli sa objavia polia na zadávanie žiadaných pólov URO (p_1, p_2) . Taktiež pre túto metódu používateľ môže zaškrtnúť políčko s názvom Kvalita regulácie, čím zvolí výpočet žiadaných pólov URO na základe hodnôt maximálneho preregulovania η_{max} a času regulácie t_{reg} , ktoré si môže zvoliť. Pri výbere druhej metódy s názvom LO kritérium sa zobrazia na paneli polia na zadávanie hodnôt matíc R a O. Pre riadenia č. 3, 4 a 5 sa zadáva tiež hodnota tretieho žiadaného pólu p_3 alebo hodnoty matíc **R** a **Q**, pričom rozmer matice Q sa zväčší na (3×3) . Pri poslednom riadení tiež pribudne voliteľná hodnota zosilnenia anti-windupu K_A .

V druhom okne sa nachádza panel s názvom Zosilnenia navrhnutého riadenia, v ktorom sa vypisujú hodnoty zosilnení zvoleného riadenia. Pod týmto panelom sa nachádza prepínací panel so zobrazovaním parametrov pre stabilitu URO a kvalitu riadenia. Na paneli *Stabilita systému* sa vypisujú póly URO a či je daný systém stabilný. Taktiež sa tieto póly v diskrétnej oblasti zobrazujú na grafe v komplexnej rovine.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 2. Užívateľské grafické prostredie aplikácie - Návrh diskrétneho stavového riadenia

Na paneli Kvalita regulácie sa vypisujú hodnoty ukazovateľov kvality regulácie pre zvolený model, a to čas regulácie t_{reg} , čas nábehu t_n , čas oneskorenia t_d , čas maximálnej prechodovej charakteristiky hodnoty maximálne t_p , podregulovanie preregulovanie maximálne η_{max} , σ_{max} a maximálna hodnota prechodovej charakteristiky P. Na paneli sa nachádza aj obrázok s otáznikom, na ktorý keď používateľ klikne, otvorí sa nové okno s názvom Ukazovatele kvality regulácie. V okne sa zobrazí graf, na ktorom sú graficky zobrazené spomínané ukazovatele kvality regulácie a ich podrobné vysvetlenie.

Ďalším väčším panelom je panel s názvom *Simulácia* simulačnej schémy. Používateľ si v rámci tohto panela môže voliť hodnotu času simulácie a hodnotu delta, čo je veľkosť skoku žiadanej veličiny w. Ďalšou možnosťou pre používateľa je výber modelov, pre ktoré sa zobrazia priebehy na grafoch, a to nelineárny model a linearizovaný model. Používateľ má tiež možnosť výberu zobrazenia žiadanej hodnoty w, pretože priebeh výstupnej veličiny y ju nedosiahne v prvom riadení ako je aj vidieť na Obr. 2. Na základe zvoleného modelu (linearizovaný/nelineárny) vo výbere s názvom *Simulačná* schéma a kvalita regulácie sa pod daným panelom zobrazuje obrázok simulačnej schémy vybraného riadenia pre vybraný model. Ako už bolo spomenuté vyššie, zvolený model slúži aj ako model, pre ktorý sa vypočítajú ukazovatele kvality regulácie.

V pravej polovici daného okna sa nachádzajú grafy na zobrazenie priebehov výstupnej veličiny y, stavovej veličiny x_1 , akčného zásahu u a regulačnej odchýlky e.

Tretie okno s názvom *Porovnanie navrhnutého riadenia* (Obr. 3) poskytuje používateľovi možnosť porovnať navrhnuté riadenia. Používateľ si môže zvoliť možnosť porovnania – *Metódy riadenia* alebo *Typov riadenia*. Na základe vybraného porovnania sa na paneli zobrazia dva výbery s rôznymi možnosťami, a to Zobrazenie na grafoch a Výber diskrétneho stavového riadenia.

Ak si používateľ zvolí možnosť Porovnanie metódy riadenia, vo výbere Zobrazenie na grafoch si môže zaškrtnúť metódy (Pole Placement – Žiadané póly URO, Pole Placement – Kvalita regulácie, LQ kritérium), pre ktoré sa zobrazia priebehy na grafoch. Porovnanie metód riadenia je možné len pre vybrané riadenie, ktoré si používateľ môže zvoliť vo výbere s názvom Výber diskrétneho stavového riadenia, ktorý obsahuje už 5 spomínaných riadení.

Ak si používateľ zvolí možnosť *Porovnanie typov riadenia*, vo výbere *Zobrazenie na grafoch* sa zobrazí 5 spomínaných riadení, pre ktoré sa zobrazia priebehy na grafoch. Ako v predchádzajúcom prípade porovnanie typov riadenia je možné len pre vybranú metódu riadenia, ktorú si používateľ môže zvoliť vo výbere s názvom *Výber metódy riadenia*, ktorý obsahuje metódy: *Pole Placement – Žiadané póly URO*, *Pole Placement – Kvalita regulácie* a *LO kritérium*.

Používateľ si ďalej môže zvoliť hodnotu času simulácie a hodnotu delta, čo je veľkosť skoku žiadanej veličiny w. Taktiež má možnosť výberu modelu (nelineárny/linearizovaný), pre ktorý sa budú zobrazovať simulované priebehy veličín a možnosť výberu zobrazenia žiadanej veličiny w na grafe výstupnej veličiny y.

V treťom okne sa nachádzajú 4 grafy. Každý graf je určený na zobrazenie časového priebehu jednej z nasledujúcich veličín: výstupná veličina y, stavová veličina x_1 , akčný zásah u a regulačná odchýlka e.

Aplikácia ponúka používateľovi pre zlepšenie komunikácie tzv. *Tooltipy*, ktoré sa zobrazujú pri podržaní myši na objekte a detailnejšie opisujú funkcionalitu objektu.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 3. Užívateľské grafické prostredie aplikácie - Porovnanie navrhnutého riadenia

4 Záver

Cieľom práce bolo vytvorenie aplikácie pre návrh diskrétneho stavového riadenia. Vybrané metódy riadenia, ktoré boli aplikované v aplikácii, boli teoreticky opísané v prvej časti práce. Následne bola podrobne opísaná funkcionalita aplikácie, ktorá bola vytvorená v prostredí MATLAB App Designer.

V aplikácii bol použitý ako riadený systém nelineárny model Van de Vusseovej reakcie prebiehajúcej v chemickom reaktore, ktorý sa linearizoval vo vybranom pracovnom bode. Aplikácia poskytovala používateľovi informácie o linearizovanom systéme a používateľ mal možnosť okrem návrhu riadenia pre vybrané metódy, vzájomne porovnať navrhnuté riadenia.

Aplikácia by mohla slúžiť ako vhodná učebná pomôcka pre oboznámenie sa s navrhovaním diskrétneho stavového riadenia, linearizáciou nelineárneho systému, ako aj s vlastnosť ami systému a riadenia v spojitej a diskrétnej oblasti.

Pod'akovanie

Ďakujem vedúcej práce Ing. Jane Paulusovej, PhD. za poskytnutú pomoc a informácie pri tvorbe príspevku. Taktiež chcem poďakovať doc. Ing. Jánovi Kardošovi, PhD. za cenné rady a poznámky k práci.

Literatúra

 FIKAR, M. TAR2 2021 02 25 Stavové riadenie spojitých systémov [cit. 2022-04-06]. Dostupné na internete:
 https://www.youtube.com/watch?v=e8X6XccyKDw&li <u>st=PLXyoII3CJDek16aCBRvRhWSr3xxn4y5bl&index=2</u> > Bratislava: UIAM FCHPT, 2021.

- [2] *Stavový popis, linearizace, Teorie dynamických systémů* [cit 2022-04-06]. Dostupné na internete: <<u>http://matlab.fei.tuke.sk/orhs/subory/podklady/Roubal_s</u> tavovy_priestor.pdf>
- [3] FIKAR, M. TAR1_20161107 & Úloha sledovania [cit. 2022-04-06]. Dostupné na internete: <<u>https://www.youtube.com/watch?v=oRIB0pZm7g0&list</u> =PLXyoII3CJDelJkUdnyvm04kC6VjSL7Zqy&index=4> Bratislava: UIAM FCHPT, 2017.
- [4] PAULUSOVÁ, J. 2019. Úvod do teórie diskrétneho riadenia, určenie diskrétnej prenosovej funkcie. Prednáška predmetu Číslicové riadenie, FEI STU.
- [5] FIKAR, M. TAR2 2021 02 25 Stavové riadenie spojitých systémov [cit. 2022-04-06]. Dostupné na internete: <<u>https://www.youtube.com/watch?v=e8X6XccyKDw&li</u> st=PLXyoII3CJDek16aCBRvRhWSr3xxn4y5bl&index=2
 > Bratislava: UIAM FCHPT, 2021.
- [6] FIKAR, M. TAR2 2021 04 15 Návrh stavového riadenia: regulátor [cit. 2022-04-06]. Dostupné na internete: <<u>https://www.youtube.com/watch?v=ZCPC3zeu3vI&list</u> =<u>PLXyoII3CJDek16aCBRvRhWSr3xxn4y5bl&index=9</u>> Bratislava: UIAM FCHPT, 2021.
- [7] KARDOŠ, J. 2019. *Regulátory a ich špeciálne štruktúry*. Prednáška predmetu Pohybové systémy, FEI STU.
- [8] PAULUSOVÁ, J. 2019. App Designer v MATLABe. Prednáška predmetu Úvod do inžinierstva a bezpečnosť v elektrotechnike, FEI STU.
- [9] LUZ, E. M. L. SANTOS, B. F. 2019. Development of Intelligent Models for the Prediction the Dynamics of Nonlinear Process. In *Chemical Engineering Transactions*, ISSN 2283-9216, 2019, roč. 74, s. 763 – 768.

Robustné riadenie polohy MIMO laboratórneho pohybového systému

Róbert Málik¹, Ján Kardoš¹

¹Ústav robotiky a kybernetiky, FEI STU v Bratislave

xmalikr@stuba.sk

Abstrakt – Príspevok je zameraný na návrh robustného riadenia polohy MIMO dynamického systému. Zákon riadenia je zložený z dvoch zložiek. Jedna zložka je navrhnutá metódou vypočítaných momentov a jej hlavnou úlohou je spätnoväzobná linearizácia dynamického systému. Druhá zložka je založená na riadení s premenlivou štruktúrou a zabezpečuje robustnosť zákona riadenia voči parametrickým a signálovým poruchám spôsobených pohybom dynamického systému, ako aj nepresnostiam v odhade parametrov dynamického systému. Navrhnutý zákon riadenia je otestovaný na laboratórnom pohybovom systéme.

1 Úvod

Návrh riadenia pre nelineárne MIMO systémy je často komplikovaná, avšak z hľadiska priemyselných aplikácii nevyhnutná úloha. V príspevku predstavíme návrh robustného riadenia tvoreného kombináciou spätnoväzobnej linearizácie prostredníctvom metódy vypočítaných momentov a riadenia s premenlivou štruktúrou.

Riadenie s premenlivou štruktúrou zabezpečuje invariantnosť systému voči parametrickým a signálovým poruchám vďaka špeciálnemu stavu dynamických systémov - kĺzavému režimu. Kĺzavý režim v systémoch vzniká pri vysokofrekvenčnom oscilovaní akčnej veličiny, takáto forma riadenia je však často pre reálne systémy nevyhovujúca. Preto vytvoríme spojitú náhradu takéhoto nespojitého riadenia s použitím pravidla dosiahnutia prepínacej funkcie.

Výsledný zákon riadenia potom implementujeme na dvojmotorovú sústavu HSM150, ktorá je ale SISO systém. Preto prvý stupeň voľnosti vytvoríme simulačne a druhý stupeň voľnosti bude predstavovať jeden motor zo sústavy HSM150, pričom druhý motor sústavy HSM150 bude simulovať nelineárne interakcie medzi jednotlivými stupňami voľnosti.

2 Dynamický model

Zákon riadenia navrhneme pre MIMO systém s dvomi stupňami voľnosti, ale použité postupy by boli použiteľné aj pri viacerých stupňoch voľnosti. Demonštráciu riadenia MIMO laboratórneho pohybového systému vykonáme na dvojmotorovej sústave HSM150, ktorá je však SISO systém. Požadovaný MIMO systém vytvoríme tak, že prvý stupeň voľnosti budeme simulovať v Simulink-u a prvý motor sústavy HSM150 bude predstavovať druhý stupeň voľnosti. Na druhý motor sústavy HSM150 privedieme žiadaný moment záťaže, ktorý vypočítame pomocou simulovaného stupňa voľnosti, a bude predstavovať signálové poruchy medzi jednotlivými stupňami voľnosti. Pohybové rovnice dynamického modelu predpokladáme v tvare:

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{q}} = \mathbf{\tau} - \mathbf{B}\dot{\mathbf{q}} - \mathbf{c}(\mathbf{q},\dot{\mathbf{q}}) - \mathbf{g}(\mathbf{q}) \tag{1}$$

kde jednotlivé matice obsahujú zložky:

$$j_{11} = J (J_{N_{11}} + J_{N_{12}} \cos (k_N q_2))$$

$$j_{12} = 0$$

$$j_{21} = 0$$

$$j_{22} = J$$

$$b_{11} = B$$

$$b_{12} = 0$$

$$b_{21} = 0$$

$$b_{21} = 0$$

$$b_{22} = B$$
(2)

$$c_{1} = C_{N_{1}} \sin(k_{N}q_{1}) \left(2\dot{q}_{1}\dot{q}_{2} + \dot{q}_{2}^{2} \right)$$

$$c_{2} = C_{N_{2}} \sin(k_{N}q_{1}) \dot{q}_{1}^{2}$$

$$g_1 = G_{N_{11}} \cos(k_N q_1) + G_{N_{12}} \cos(k_N (q_1 + q_2))$$

$$g_2 = G_{N_2} \cos(k_N (q_1 + q_2))$$

Pohybová rovnica (1) s maticovými zložkami (2) je obdobou dynamického modelu dvojčlánkového robota [1–3, 8–10]. Keď že ale nemáme k dispozícii takýto laboratórny MIMO systém, tak sme kvôli realizovateľ nosti maticové zložky (2) prispôsobili vlastnostiam sústavy HSM150 normalizačnými konštantami (konštanty s indexom *N*).

Dynamický model (1) môžeme dekomponovať na jednotlivé stupne voľnosti [1,2]:

$$j_{mm}\left(\mathbf{q}\right)\ddot{q}_{m}=\tau_{m}-\tau_{z_{m}}-B\dot{q}_{m} \tag{3}$$

kde τ_{z_m} označuje moment záťaže:

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

$$\tau_{z_m} = c_m \left(\mathbf{q}, \dot{\mathbf{q}} \right) + g_m \left(\mathbf{q} \right) + \sum_{n \neq m} j_{mn} \left(\mathbf{q} \right) \ddot{q}_n \tag{4}$$
$$m, n = 1, 2$$

Ďalej v rovnici (3) zavedieme redukovaný vstup u_m , časovú konštantu T_m a zosilnenie K_m :

$$u_m = \tau_m - \tau_{z_m} \tag{5}$$

$$T_m = \frac{j_{mm}\left(\mathbf{q}\right)}{B} \tag{6}$$

$$K_m = \frac{1}{B} \tag{7}$$

Potom rovnica (3) prejde do tvaru:

$$T_m \ddot{q} = K_m u_m - \dot{q}_m \tag{8}$$

Vidíme, že podľa rovnice (6) časová konštanta systému T_m je premenlivá podľa polohového vektora **q** dynamického systému. Teda hodnota časovej konštanty sa pohybuje v intervale $T_m \in \langle T_{m_{min}}, T_{m_{max}} \rangle$. Z hľadiska návrhu robustného riadenia bude dôležitá práve hodnota časovej konštanty $T_{m_{max}}$, keď systém dosahuje najhoršiu dynamiku.

3 Riadenie s premenlivou štruktúrou

Robustný zákon riadenia založíme na nespojitom nelineárnom riadení využívajúcom kĺzavý režim, takzvanom riadení s premenlivou štruktúrou (VSC - Variable Structure Control) [1–6]. Kĺzavý režim je špeciálny stav nelineárnych dynamických systémov, v ktorom dynamický systém kopíruje definovanú krivku vo fázovom priestore, systém sa tak správa invariantne voči parametrickým a signálovým poruchám [5,6]. Kĺzavý režim sa vyznačuje vysokofrekvenčnými osciláciami akčného zásahu. Riadenie predpokladá oscilovanie akčného zásahu medzi jeho hraničnými hodnotami $-\tau_{max}$ a τ_{max} , kde $\tau_{max} > 0$ vyjadruje maximálnu možnú hodnotu akčného zásahu. Matematicky je možné takýto zákon riadenia vyjadriť v tvare [2,3]:

$$\tau = \tau_{max} \mathrm{sgn}\left(F\right) \tag{9}$$

kde F označuje zvolenú prepínaciu funkciu:

$$F = \alpha e + \dot{e} \tag{10}$$

Zákon riadenia (9) a prepínacia funkcia (10) sú kvôli prehľadnosti uvedené pre jeden stupeň voľnosti. Rovnako budeme uvádzať pre jeden stupeň voľnosti aj výpočty jednotlivých parametrov zákona riadenia.

Požadujeme regulačný pochod bez preregulovania v čo najkratšom čase pre najhoršiu dynamiku systému T_{max}. Takéto časovo suboptimálne riadenie získame, pokiaľ sa systém bude rozbiehať s maximálnym hnacím momentom a po dosiahnutí hodnoty prepínacej funkcie F = 0 (prepínacia priamka) začne brzdiť s maximálnym brzdným momentom. Zvolená prepínacia funkcia (10) je parametrizovaná parametrom α (sklon prepínacej kde δ [%] je požadovaná presnosť sledovania.

priamky). Zo znalosti fázových trajektórii systému (3) môžeme vo fázovej rovine vyjadriť parameter α v tvare [1,2]:

$$\alpha = \frac{1}{T_{max} \left(1 - \frac{\tau_2}{\tau_1} \ln \left(\frac{\tau_1}{\tau_2} + 1 \right) \right)} \tag{11}$$

kde τ_1 označuje najnepriaznivejší hnací moment pri rozbehu a τ_2 označuje najnepriaznivejší hnací moment pri brzdení [1,2]:

$$\tau_1 = \tau_{max} + \tau_{z_{max}} + \frac{\dot{q}_{max}}{K} \tag{12}$$

$$\tau_2 = \tau_{max} - \tau_{z_{max}} - \frac{q_{max}}{K} \tag{13}$$

Člen $\frac{\dot{q}_{max}}{K}$ vo výrazoch (12) a (13) zohľadňuje vplyv vývoja dynamiky referenčného signálu.

Dôležitou vlastnosť ou riadenia s premenlivou štruktúrou je maximálna uhlová frekvencia referenčného signálu ω_{max} , ktorú je systém schopný sledovať v kĺzavom režime. Hodnotu maximálnej uhlovej frekvencie ω_{max} , môžeme získať aplikovaním metódy fázového vektora rýchlosti [1,2]:

$$\omega_{max} = \frac{-1 + \sqrt{1 + 4\frac{KT_{max}}{A}\tau_2}}{2T_{max}}$$
(14)

Zákon riadenia s vysokofrekvenčným prepínaním akčnej veličiny je ale nevhodný na riadenie reálnych pohybových systémov. Podmienkami vzniku kĺzavého režimu sú konvergencia fázových trajektórií k prepínacej krivke a jeho vznik v konečnom čase [4-6]. Vhodným zvolením pravidla dosiahnutia [4-7], ktoré určuje dynamiku vývoja hodnôt prepínacej funkcie, ale môžeme zamedziť vzniku klzavého režimu bez straty robustnosti [1,2]. Teda zákon riadenia môžeme vyjadriť ako ekvivalentné časovo suboptimálne riadenie vo forme PD-regulátora:

$$\Delta \tau = K_p e + K_d \dot{e} \tag{15}$$

kde:

$$K_p = \frac{T_{max}\alpha k}{K} \tag{16}$$

$$K_d = \frac{T_{max}\left(\alpha + k\right) - 1}{K} \tag{17}$$

Po aplikovaní pravidla dosiahnutia v zákone riadenia pribudol nový parameter k. Daný parameter súvisí s presnosť ou sledovania referenčného signálu. Hodnotu parametra k môžeme vyjadriť použitím frekvenčnej metódy kontrolného bodu [1,2]:

$$k = \frac{100\omega_{max}}{\mu\delta} \sqrt{\frac{\frac{1}{T_{max}^2} + \omega_{max}^2}{\alpha^2 + \omega_{max}^2}}$$
(18)
$$\mu = \frac{\tau_2}{\tau_{max}}$$

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

4 Metóda vypočítaných momentov

V zákone riadenia (19) použijeme metódu vypočítaných momentov založenú na spätnoväzobnej linearizácii [8–10]. Spätnoväzobná linearizácia spočíva v transformovaní nelineárneho systému na ekvivalentný linearizovaný systém pomocou akčnej veličiny:

$$\boldsymbol{\tau} = \boldsymbol{\tau}_0 + \boldsymbol{\Delta}\boldsymbol{\tau} \tag{19}$$

kde τ_0 je príspevok od metódy vypočítaných momentov, ktorá zabezpečuje linearizáciu systému a $\Delta \tau$ predstavuje navrhnuté ekvivalentné časovo suboptimálne riadenie (15), ktoré zabezpečuje stabilitu systému a kvalitu riadenia.

Pohybovú rovnicu (1) prepíšeme do tvaru:

$$\ddot{\mathbf{q}}^* = (\mathbf{J}(\mathbf{q}))^{-1} \left(\mathbf{\tau} - \mathbf{B}\dot{\mathbf{q}} - \mathbf{c}\left(\mathbf{q}, \dot{\mathbf{q}}\right) - \mathbf{g}\left(\mathbf{q}\right) \right)$$
(20)

Na základe výrazu (20) zvolíme:

$$\tau_0 = \mathbf{J}(\mathbf{q}) \,\ddot{\mathbf{q}}^* + \mathbf{B} \dot{\mathbf{q}}^* + \mathbf{c}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{g}(\mathbf{q}) \tag{21}$$

Po dosadení zákona riadenia (19) do výrazu (20) získame linearizovaný systém v tvare:

$$\mathbf{J}(\mathbf{q})(\ddot{\mathbf{q}}^* - \ddot{\mathbf{q}}) + \mathbf{B}(\dot{\mathbf{q}}^* - \dot{\mathbf{q}}) + \Delta \tau = \mathbf{0}$$
(22)

alebo pre vektor regulačnej odchýlky $\mathbf{e} = \mathbf{q}^* - \mathbf{q}$:

$$\mathbf{J}(\mathbf{q})\ddot{\mathbf{e}} + \mathbf{B}\dot{\mathbf{e}} + \Delta\tau = \mathbf{0}$$
(23)

kde ė a ë označujú časové derivácie vektora regulačnej odchýlky.

V reálnych systémoch je často potrebné pri návrhu zákona riadenia zohľadniť obmedzenie rýchlosti. Do zákona riadenia (19) dosadíme výraz (15) a rozpíšeme regulačnú odchýlku rýchlosti $\dot{e} = \dot{q}^* - \dot{q}$ pre jeden stupeň voľnosti:

$$\tau = \tau_0 + K_p e + K_d \dot{q}^* - K_d \dot{q} \tag{24}$$

Obmedzenie rýchlosti môžeme v zákone riadenia (24) zabezpečiť pokiaľ predpokladáme, že rýchlosť systému sa správa ako výstup systému 1. rádu. V ustálenom stave pri obmedzení rýchlosti $v_{lim} > 0$ potom musí platiť:

$$\tau_{v_{lim}} = \frac{v_{lim}}{K} \tag{25}$$

Použitím ustáleného stavu (25) a zákona riadenia (24) môžeme po niekoľ kých úpravách odvodiť obmedzenie:

$$-\left(\frac{1}{K}+K_d\right)v_{lim} \le \tau_0 + K_p e + K_d \dot{q}^* \le \left(\frac{1}{K}+K_d\right)v_{lim} \qquad (26)$$

kde v_{lim} predstavuje zvolenú hodnotu obmedzenia rýchlosti.

Na základe obmedzenej rýchlosti môžeme tiež uvažovať nový sklon prepínacej priamky α [3]. Aby bol systém schopný sledovať referenčný signál, tak rýchlosť referenčného signálu nesmie prekročiť obmedzenie v_{lim} . Teda maximálna regulačná odchýlka rýchlosti môže dosahovať hodnotu $|\dot{e}| = 2v_{lim}$ a potom môžeme uviesť:

$$\alpha = \frac{1}{T_{max} \left(1 - \frac{K\tau_2}{2\nu_{lim}} \ln\left(\frac{2\nu_{lim}}{K\tau_2} + 1\right) \right)}$$
(27)

Treba poznamenať, že riadenie by fungovalo aj s pôvodnou hodnotou parametra α , ale dosahovalo by zbytočne nižšiu dynamiku.

5 Implementácia zákona riadenia na dvojmotorovú sústavu HSM150

Riadenie otestujeme na dvojmotorovej sústave HSM150 pre skokový referenčný signál a harmonický referenčný signál pre *m*-tý stupeň voľnosti v tvare:

$$q_m^* = A_m \sin(\omega_m t + \frac{\pi}{2}) \tag{28}$$

Tab. 1: Parametre systému, zákona riadenia a harmonického referenčného signálu

DOF (m)	1	2
B $[\mathbf{N} \cdot \mathbf{m} \cdot \mathbf{s}]$	$7,03 \cdot 10^{-5}$	$7,03 \cdot 10^{-5}$
$J\left[kg\cdot m^{2}\right]$	$1,2\cdot 10^{-4}$	$1,2\cdot 10^{-4}$
J _{N11} [-]	1,75	-
$J_{N_{12}}$ [-]	0,25	-
C_{N_1} $\left[N \cdot m \cdot s^2\right]$	$3,8095 \cdot 10^{-7}$	-
$C_{N_2} \left[N \cdot m \cdot s^2 \right]$	-	$2 \cdot 10^{-6}$
$G_{N_{11}} \ [N \cdot m]$	0,06	-
$G_{N_{12}} \ [N \cdot m]$	0,04	-
$G_{N_2} \ [N \cdot m]$	-	0,04
k_N [-]	0,0419	0,0419
$j_{mm_{max}} \left[kg \cdot m^2 \right]$	$2, 4 \cdot 10^{-4}$	$1, 2 \cdot 10^{-4}$
T _{mmax} [s]	3,4139	1,7070
$\mathbf{K}_{\mathbf{m}} \left[\mathbf{N}^{-1} \cdot \mathbf{m}^{-1} \cdot \mathbf{s}^{-1} \right]$	14225	14225
$K_{p_m} \ [N \cdot m]$	0,1413	0,0875
$K_{d_m} \ [N \cdot m \cdot s \]$	0,0586	0,0181
$\tau_{max_m}~[N\cdot m]$	0,2925	0,2925
$\tau_{z_{max_m}}~[N\cdot m]$	0,18	0,12
$v_{lim_m} \left[rad \cdot s^{-1} \right]$	200	300
$\alpha_{\rm m}$ [s ⁻¹]	2,4333	4,9711
$\omega_{\max_m} \left[rad \cdot s^{-1} \right]$	1,5136	1,9727
δ_m [%]	1	1
$k_m \left[s^{-1} \right]$	241,9655	146,6285
A _m [rad]	150	250
$\omega_m \left[rad \cdot s^{-1} \right]$	1	1

Fakulta elektrotechniky a informatiky STU v Bratislave



Obr. 1: Priebeh fázových veličín pri skokoch referenčného signálu.



Obr. 2: Priebeh akčného zásahu pri skokoch referenčného signálu: a) referenčný akčný zásah zo zákona riadenia. b) meranie akčného zásahu z motora HSM150

Hodnoty momentu zotrvačnosti J a koeficientu viskózneho trenia B motora HSM150 v tabuľ ke 1 boli získané z katalógových údajov dvojmotorovej sústavy HSM150.

Na obrázku 1 a 3 môžeme vidieť priebehy fázových veličín a ich obmedzení počas riadenia systému. V obmedzeniach



Obr. 3: Priebeh fázových veličín pri sledovaní harmonického referenčného signálu.



Obr. 4: Priebeh akčného zásahu pri sledovaní harmonického referenčného signálu: a) referenčný akčný zásah zo zákona riadenia. b) meranie akčného zásahu z motora HSM150

rýchlosti sa vyskytujú mierne preregulovania vplyvom nelinearít systému, môžeme ale zhodnotiť, že rýchlosť systému je obmedzená na požadovanú hodnotu. Z grafického porovnania tiež vidíme, že riadená veličina veľmi dobre sleduje referenčný signál.

Sekcia 3. Robotika a kybernetika

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave





Obr. 5: Priebeh regulačnej odchýlky pri skokoch referenčného signálu: a) priebeh regulačnej odchýlky, b) detail priebehu regulačnej odchýlky

Obr. 6: Priebeh regulačnej odchýlky pri sledovaní harmonického referenčného signálu: a) priebeh regulačnej odchýlky, b) detail priebehu regulačnej odchýlky

Vysokofrekvenčné oscilácie ako aj mierne presiahnutia obmedzenia akčného zásahu (obrázok 2 a 4) sú spôsobené šumom pri meraní a samotným generátorom momentu. O ekvivalentnom časovo suboptimálnom riadení teda môžeme tvrdiť, že dokázalo odstrániť reléový charakter časovo suboptimálneho riadenia. Skoky akčného zásahu nastávajú len z dôvodu skokových zmien referenčného signálu, pretože pri dosahovaní referenčného signálu požadujeme aby sa systém pohyboval s maximálnou možnou dynamikou.

Podľa grafického porovnania regulačných odchýlok (obrázok 5 a 6) vidíme, že riadenie dosahuje požadovanú presnosť sledovania s veľkou rezervou, kvalita riadenia je tak na veľmi dobrej úrovni. Zákon riadenia sa teda správa robustne voči vyskytujúcim sa parametrickým a signálovým poruchám.

6 Záver

Cieľ om príspevku bol návrh a implementácia robustného zákona riadenia na nelineárny MIMO laboratórny pohybový systém. Ako robustné riadenie bola zvolená kombinácia metódy vypočítaných momentov s riadením s premenlivou štruktúrou vyznačujúcim sa kĺzavým režimom, v ktorom sa systém správa invariantne voči parametrickým a signálovým poruchám. Riadenie s kĺzavým režimom je charakterizované vysokofrekvenčnými osciláciami akčného zásahu, takáto forma riadenia je často nevhodná pre aplikáciu na reálne systémy. Voľ ba vhodného pravidla dosiahnutia prepínacej funkcie umožnila potlačiť vznik kĺzavého režimu bez straty robustnosti, získali sme tak ekvivalentné časovo suboptimálne riadenie. Hlavnou úlohou ekvivalentného časovo suboptimálneho riadenia bolo zabezpečenie stability a požadovanej kvality riadenia a metóda vypočítaných momentov pôsobila ako spätnoväzobná linearizácia systému.

Navrhnutý zákon riadenia sme potom aplikovali na reálnu dvojmotorovú sústavu HSM150. Sústava HSM150 je však SISO systém, MIMO systém sme vytvorili tak, že prvý stupeň voľ nosti bol simulovaný v Simulink-u a druhý stupeň voľ nosti reprezentoval prvý motor sústavy HSM150. Nelineárne interakcie medzi jednotlivými stupňami voľ nosti sme simulovali pomocou druhého motora sústavy HSM150 tým, že sme naň priviedli žiadaný moment záť aže, ktorý sme vypočítali pomocou simulovaného stupňa voľ nosti.

Z výsledkov experimentov môžeme zhodnotiť splnenie žiadanej presnosti sledovania referenčného signálu. Celkovú kvalitu riadenia môžeme vyhodnotiť ako veľmi dobrú, napriek pôsobeniu parametrických a signálových porúch, ako aj nelinearít vyskytujúcich sa v reálnom systéme vplyvom jeho opotrebovania.

Literatúra

- KARDOŠ, J. Robust Computed Torque Method of Robot Tracking Control. 22nd International Conference on Process Control. 2019. Štrbské Pleso, Slovensko. DOI: 10.1109/PC.2019.8815088.
- [2] KARDOŠ, J. Robust Robot Control Driven by a Variable Structure State Observer. 19th International Carpathian Control Conference, ICCC 2018. 2018 Szilvásvárad, Maď arsko. DOI: 10.1109/carpathiancc.2018.8399665.
- [3] KARDOŠ, J. Robust Point-to-Point Control with Velocity Limitation. 21st International Conference on Process Control. 2017. Štrbské Pleso, Slovensko. DOI: 10.1109/pc.2017.7976193.
- [4] HUNG, J. Y. GAO, W. HUNG, J. C. Variable Structure Control: a Survey. IEEE Transactions on Industrial Electronics, Vol. 40, No. 1. 1993. DOI: 10.1109/41.184817.
- [5] EDWARDS, C. SPURGEON, S. K. Sliding Mode Control (Theory and Applications). 1998. ISBN-13: 978-0-7484-0601-2.
- [6] UTKIN, V. GULDNER, J. SHI, J. Sliding Mode Control in Electro Mechanical Systems (Automation and Control Engineering). 2009. ISBN-13: 9781420065602.
- [7] XIAOYUAN, W. YAOPENG, Z. PENG, G. Design and Analysis of Second-Order Sliding Mode Controller for Active Magnetic Bearing. 2020. URL: https://www.google.com/url? sa=t&rct=j&q=&esrc=s&source=web&cd=&ved= 2ahUKEwj7u9eJ78HzAhWKC-wKHbC0BEsQFnoECDEQAQ& url=https%3A%2F%2Fwww.mdpi.com% 2F1996-1073%2F13%2F22%2F5965%2Fpdf&usg= AOvVaw32h2yFgBPEpJF4G--Z1H12. [cit. 2022-04-08].
- [8] KELLY, R. SANTIBÁŃEZ, V. LORÍA, A. Control of Robot Manipulators in Joint Space. 2005. Londýn, Spojené kráľ ovstvo. ISBN-13: 978-1-85233-994-4.
- [9] WANG, X. HOU, B. Trajectory tracking control of a 2-DOF manipulator using computed torque control combined with an implicit Lyapunov function method. Journal of Mechanical Science and Technology. 2018. DOI: 10.1007/s12206-018-0537-6.
- [10] NEUROMEKA. Computed Torque Method. URL: http://docs.neuromeka.com/2.3.0/en/IndySDK/ section_ctm/. [cit. 2022-04-08].

Segmentation of retinal images damaged by diabetic retinopathy

Michal Kováč¹

¹Slovak University of Technology, Bratislava 812 43, Slovakia

xkovacm12@stuba.sk

This article deals with the methods of pre-processing and semantic segmentation of significant structures of retina. We work with publicly available database of fundus images E-Optha Ex and DDR; these databases contain images with marked exudates - symptoms of diabetic retinopathy (DR). The images were being pre-processed to be suitable for the segmentation of exudates by U-Net. The images were divided into patches and processed via augmentation. Subsequently, the patches were classified into healthy and pathological to accomplish the best balance of the training data. The result of the segmentation was evaluated by metrics, which are important in medical use: sensitivity, specifity and Dice coefficient.

1 Introduction

Diagnostics is the key to the correct medical treatment of various diseases. Identification of the patient's diagnosis depends on doctor's knowledge and long-time experience. To perform automatic diagnostics by computer, a mathematical description of the relation between the symptoms and diseases is required; a process that would be extremely complicated. The applicable solution is neural network [1]. The goal of our task is processing and segmentation of digital images of fundus oculi. These images are used to identify several diseases, including diabetic retinopathy (DR). One of the identifiable DR symptoms are hard exudates that are composed of lipids and proteinaceous material Early diagnostics is especially important, because the [7]. formation and presence of exudates is not necessarily painful for the patient. By means of the method we designed, we determine the individual exudates from the digital images. In our work, we use U-net architecture, that was developed especially for purposes of medical diagnostics [5]. We trained and tested the Unet on 2 different databases and compare the achieved outcomes to similar works. Our contribution is organized as follows; Chapter II introduces the related works related to segmentation of exudates. The image databases and their preparation for Unet training and testing are described in part III. The U-net used for segmentation of exudates is presented in the part IV. Part V contains experiments and results evaluation.

2 Related works

Image segmentation is often used to process medical data to diagnose various diseases. An article by Kaura und Mittala (2018) compares different methods of segmentation of exudates, and also mentions advantages and disadvantages of each method. Global and adaptive thresholding was one of the first methods of segmentation of fundus images and it was mentioned by Phillips (1993). The basic methods for segmentation of exudates used computer vision features such as: morphological operations, thresholding, edge detection. Later, neural networks came in use and nowadays, mainly convolutional neural networks are being employed. Kou et al. (2020) [1] worked with convolutional neural network U-Net in segmentation of fundus images, and they also introduced their own neural network ERU-Net. Their work was focused on diagnostics of the diabetic retinopathy by looking for the disease signs on fundus images. Microaneurysms and exudates were segmented from these images by convolutional neural networks. To increase the accuracy of the U-Net segmentation, they modified the U-Net architecture by adding two expansion paths and residual blocks. E-Optha-Ex, IDRiD and DDR databases were used as dataset for the segmentation of exudates [4].

3 Data preparation

3.1 Databases

For U-net training and for correct evaluation of segmentation outcomes were needed pathological images of retina with already marked exudates on pixel level. We used images of 2 publicly available databases - the E-Optha Ex and the DDR [5] [6].

E-Optha Ex database offers 47 color fundus images with already identified exudates and a number of images without exudates. It was designed especially for research of diabetic retinopathy. All the findings were marked by several ophthalmologists. Overall, 2 278 exudates are marked on the 47 images. The images differ in size, they have following resolutions: 2 048 x 1 360, 2 544 x 1 696, 1 440 x 960, and 1 504 x 1 000 pixels. Apart from the pathological images, the database includes 35 images of healthy retina.

DDR database was firstly introduced in article. Main purpose of this database is DR screening, lesion segmentation

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

and lesion detection. This database consists 13 673 fundus images with image-level annotations, and among them, 757 images are provided with pixel-level and bounding-box-level annotations. Images in this database also differ in size from 1 380×1382 to 5184×3456 pixels. [6]

3.2 Data pre-processing

For the automatic segmentation of exudates, we use E-Optha database and DDR database. E-Optha Ex database contains training dataset and test dataset; to obtain validation dataset, we split the training dataset. On the other hand, DDR database already includes all required datasets. Each dataset must be correctly pre-processed to obtain properly trained network. The input data for neural network has to be images of same dimension, and the size of images is important, because of the memory load. The number of training data had to be increased, and the pixels had to be on the required scale. The data had to be normalised, to obtain the best results of neural network learning. On images prepared in such way, the neural network could learn; however, the obtained training results might not be ideal. One of the most important factors of the training dataset is balance of individual segmentation classes. In our case, there are only two classes - exudates and background.



division of images into patches. From these 2 386 images, there were 1 846 color images and 540 black-background images. All black images came from the image corners where retina was not displayed. There were 265 images with exudates. The data was to a large extent imbalanced, because there were far more images without exudates (only background) than with exudates. For this reason, all black images were discarded, see image Fig. 2. Such adjustments of input data had to be done also on mask-images with marked exudates (masks). We removed only black images. The DDR test dataset multiplied from 225 images into 19 869 images by division of the images. After dividing these data into patches, it created 32 287 images and without black patches it was 27 190. The validation dataset consists of 149 images and after dividing them into patches, we had 14 528 at our disposal.



Fig. 2: Example of pre-processing on E-Optha Ex database

Fig. 1: Process of the data pre-processing

There are also two types of imbalanced data – first in overall dataset, and the second refers to individual images and their pixels. It is crucial that the overall data set contains mainly images with sought objects and fewer images without the exudates. The issue with the second type of imbalanced data, the individual images, and their pixels, is caused by the disproportion between a few pixels containing exudates (first class) and many pixels containing background (second class), the irrelevant parts. Once the input data are pre-processed, they need to be divided into batches. When dividing the data, it is crucial that various types of data are represented in each batch equally – stratification is necessary.

3.2.1 Division into patches

The input images had to be converted, into equal size, acceptable for available memory size. For this purpose, we divided the original images from the database into smaller, but equal pieces of 256 x 256 pixels. In case that the size of original image was not divisible by the size of smaller image, the original image was enlarged by zero padding to desired size. In practice, almost all the original images had to be processed in such way. By dividing the images, we acquired size of image suitable for training, and we acquired larger quantity of input data. The E-Optha Ex training dataset swelled from 47 images into 2 386 images by

Augmentation

3.2.2

The purpose of the augmentation of dataset was to obtain better results even with little data. We increased the data in two different ways. First method - the offline augmentation, we augmented the data and added them to the whole dataset. We also tried to augment the data randomly directly during the training process (online augmentation), i. e. in the input The advantage of the second method is of the network. that the augmented images are not saved, and therefore, the memory is not utilised. Initial 47 images increased in the first step by dividing them into smaller equal blocks. After this augmentation, the number of images increased to 1 846. This first way of augmentation was based on artificial increase of the training dataset by flipping every image horizontally and vertically. Afterwards, we obtained triple of the training data, from 1 846 images to 5 538 images, sized 256 x 256 pixels. The rotations were performed with the input images and masks as well. We can see the original image from database of size 2 544 x 1 696 divided into smaller images of size 256 x 256 pixels, which were subsequently rotated on horizontal and vertical (see Fig. 2) axis. This augmentation methods was used only for E-Optha Ex database, DDR training dataset was too large for another offline augmentation.

4 Segmentation of vascular exudates from images of retina by u-net

4.1 Methodology

We decided to solve the segmentation of exudates of medical images by convolutional neural network of type U-Net. As mentioned above, this type of network is designated specifically for such data. Our goal was to segmentate the images by pixels and mark those images that include the exudates. Similarity evaluation of neural network prediction and ground true exudates was determined by Dice coefficient. The model of the network had to be validated directly in the process of training after one finished epoch on different than training data.

The network had to be tested when it finished training, and the output images had to be depicted – the predictions. Besides the predictions, it was important to control selected metrics of the output evaluation. During the training, the values of loss function and metrics were listed: TPR (True positive rate), TNR (True negative rate), TP (True positive), FP (False positive), IoU (Intersection over union). The same metrics had to be monitored also on the validation and the network testing. After we reviewed these data, we could adjust the parameters of the network, or detect potential mistakes. After the network was trained, we selected the best trained model on the basis of minimal value of loss function, and we saved the model for further use.

We approached the training of the data with two different approaches. By the first approach, we trained the network on all training data - on images with and without exudates. As mentioned above, the number of images without the exudates was significantly bigger than the images with the exudates. This approach could cause a problem for the network, resulting in incorrect training, because the dataset was too imbalanced, and the network would train on images without exudates. Therefore, we decided to adjust the dataset, so that is contains only images with pathological symptoms. We manually removed the images without the exudates and their masks from the original dataset. This adjustment is only allowed for training datasets. The same model of network was used in both types of training. We expected better success rate of segmentation when training the network only on relevant images (images containing both classes - background and exudates). Both types of training approaches had to be compared and evaluated. The second approach (training on adjusted dataset) was implemented only on E-Optha Ex database, because removing images without lesions on DDR database would be significantly exacting, considering the amount of images.

4.2 Network architecture

The model of the network was based on the original design according to Fig. 3, on which we exercised several adjustments. In the original architecture, the ReLu was used as an activation function between convolutional layers. We substituted this activation function with function ELU or function hyperbolic tangens. When we used the ReLu activation function, the features set to zero; therefore, the final predictions of the network were black masks of exudates – we were losing features due to the ReLu activation function. When we used the ELU (exponential ReLu) or hyperbolical tangens, the trained features were no set to zero; therefore, the predictions were non-zero – we were not losing features. Consequently, we adjusted the input and output layers of the network according to the parameters of our images. In order to calculate the loss function, we used the relation 1 - Dice coefficient, which we tried to minimalize in the process of training by means of optimization algorithm.



Fig. 3: Original U-Net architecture [5]

4.3 Training

We needed to train our network, i. e. in the course of several epochs, we put images in batches in the input of the network, and we evaluated similarity between prediction of the network and the ground truth through the loss function.

The change of weights was evaluated by optimizer on the basis of the loss function, and the cycle repeated after the change of weights. In order to achieve the best results of the image segmentation, we had to set the training parameters correctly; therefore, we exercised several experiments with various network parameters. Thus, we changed the algorithm of the network optimalization, gradient, batch size, and to a certain extent also the model of the network (by changing the activation functions) in the course of training.

5 Experiments

5.1 Network Training on the whole E-Optha Ex Dataset

After several tests, we found the best parameters concerning the network training, by which we achieved the best courses of the Dice coefficient. The process of training with the optimizer being Nadam with the gradient $1 \cdot 10^{-5}$, batch size 32 and 400 epochs. The Dice coefficient on training dataset reached 80 % with this adjustment, but as seen in Fig. 4, the Dice coefficient on validation dataset saturated on around 50 %. Saturation of the Dice coefficient on validation data signified that the training data was no longer sufficient to train the network. Comparison of the network prediction and ground truth (specific part of color

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

image) can be seen in Fig. 5. Together with the prediction and ground truth was depicted also sum of all pixels, i.e., total of 29 pixels containing exudates. After the analysis of the predictions of the network, we found that the predicted images, which truly contained exudates, looked like the ground truth of an image at first sight, but some pixels of exudates were identified by the network as background, and the background pixels were falsely identified as exudates. The issue lied with images containing no exudates, see Fig. 5; such predictions contained many falsely positive images. According to the ground truth, the image does not contain any exudates, but the network falsely segmented several pixels as exudates.



Fig. 4: Dice coeficient on E-Optha Ex database



Fig. 5: Two Examples of U-Net Prediction on E-Optha Ex test Dataset

Saturation of the Dice coefficient on validation data signified that the network was over-trained; therefore, we used online augmentation of the images directly in the network input. As mentioned above, the images are adjusted randomly. Thus, when the same image enters the network, the network considers it a new image. After several adjustments to the network, we achieved the best outcomes with such type of augmentation, however, the same as without the augmentation. When shearing and shifting the images, the process of training worsened significantly; therefore, we used only rotation and zoom in onward. Furthermore, we tested the network by crossvalidation in such manner that we used the network parameters, with which we achieved the best outcomes, and we repeated the process of training and testing five times. The final Dice coefficient from testing was noted in chart, while the training and test data changed randomly every cycle. The outcomes of cross-validation can be seen in Tab.1.

Tab. 1: Results of Cross-validation of U-Net on the whole Dataset

Cross-validation						
Number [-]	1.	2.	3.	4.	5.	
Dice coefficient [%]	59.3	62.5	58.8	60	60	

5.2 Network Training on the whole DDR Dataset

In case of DDR database, we proceeded in the same way as with the previous database when training. With the use of adaptive optimizers as Adam and Nadam, the final course of loss function oscillated and the value of loss function resulted in local minimum and the resulting network predictions were black images. Due to this result, we used SGD optimizer. With this optimizer, the learning rate does not change. To prevent the value of the loss function resulting in local minimum, we needed to change the learning rate in the process of training as follows: at the beginning of the training, we needed to ensure higher value of the learning rate. If the course of the loss function stopped to converge, we reduced the learning rate in such a way that the course of the loss function converges again. As we can see in Fig. 6a, the Dice coefficient value was increasing, saturated around the 200th epoch, at which point we reduced the learning rate, and the course of the loss function started to converge again. We repeated this procedure for the whole training process. After 500 epochs at the end of the training, we noticed that the course tended to converge further, we therefore finished training the network with the initial weights from the previous training. Such progression of the Dice coefficient in the course of training can be seen in Fig. 6b; the value of the Dice coefficient keeps increasing, but slower, because the learning rate is reduced.

Tab. 2: Comparison of the success rate of segmentation on two different databases of retina

Database	Metric	Validation	Test
E Onthe Ex	IoU [-]	-	-
E-Opula Ex	Dice coeff. [-]	0.5	0.6
מחח	IoU [-]	0.0911	0.2773
DDK	Dice coeff. [-]	0.397	0.4342

To compare our outcomes of segmentation of exudates from fundus images from DDR database using the U-Net network (see Tab. 2) with the outcomes mentioned in the article by Tao et al. (2019) [6] - in segmentation of exudates using the HED model, they achieved on testing data value IoU = 18.74%, and on validation dataset value IoU = 9.4%. With DeepLab-v3 + model, they achieved outcomes on testing data IoU = 31.18%, and on validation dataset IoU = 29.1%.



(a) The first 500 epochs of the training process



(b) Training of the network from 500 to 1 000 epochs

Fig. 6: Process of training U-Net on DDR Database evaluated by Dice coefficient



Fig. 7: Examples of Network Prediction on DDR test Data.

5.3 Network training on the adjusted E-Optha Ex dataset

Firstly, the training dataset of images had to be adjusted in such manner that all images without exudates had to be removed. This process was similar to the removal of totally black images from the dataset. We went step by step through all masks of images, and we saved values 1 or 0 to the created vector, in which case 1 meant that image contains exudates, and 0 that image contains only background. In order to decide on the existence of the exudates, we summed up the pixels of the whole image. If the sum of pixels was zero, the mask did not contain any exudates. Subsequently, we went through the created vector, input images



Fig. 8: The training process of the network on adjusted dataset of E-Optha Ex database

and masks, and we saved only such images that contained 1 in vector in the same place. After we removed the images without exudates, the dataset decreased to 795 images, and 20 % of the images were used for validation. This dataset already contained the augmented images.

The best adjustment of the network was tested by crossvalidation, where we repeated the training and testing of the network five times, always on different group of data, and the numbers were noted in chart. The length of training in crossvalidation was selected on the basis of the Fig. 8, which depicts the duration of epochs, during which the Dice coefficient reached the maximum on validation data. Subsequently, we added a certain reserve to this number, because it may take various amount of time to achieve similar results by different network training. Due to this fact, we selected the training to last 150 epochs during the cross-validation. It was in our interest that during this test the network trained on the minimum epochs (for shorter training time), but enough for the network to duly train to obtain relevant results of the tests.

Tab. 3: Results of cross-validation on the adjusted dataset

Cross-validation						
Number [-]	1.	2.	3.	4.	5.	
Dice coefficient [%]	59.2	62.1	60.0	62.0	62.3	

The results of the validation correspond with the graphic course from the Fig. 8, when the Dice coefficient reached around 60 % on validation data. From the chart referring to the results of the cross-validation is obvious that the Dice coefficient is in testing around 60 % \pm 2 % by various training and test data from the whole adjusted dataset, see Tab. 3.

6 Conclusion

The aim of our work was processing and subsequent segmentation of digital medical images. Firstly, we got acquainted with the common methods of processing and segmentation of images. When studying the existing methods, we found out that

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave



Fig. 9: Examples of U-Net Prediction on E-Optha Ex test Dataset

with the use of convolutional networks were reached interesting outcomes. Our solution was therefore aimed this direction. We decided on that account to use convolutional neural network of U-Net type. We chose to use the U-Net architecture, because this architecture was designed for the segmentation of medical data. The designed model of convolutional network was trained and tested on images from DDR and E-Optha Ex databases. The procedure was the same when pre-processing the images from these two databases.

When training the network, we created two datasets. The first dataset contained all images, the second dataset was adjusted in such a way that all images without exudates were removed. Since only relevant data entered the network, therefore a great number of images with exudates, we expected better outcomes when training on the whole dataset. We determined the parameters of the network as optimizer, learning rate and batch size experimentally. In case of DDR database, we created only one dataset that contained all images.

In case of E-Optha Ex and both approaches to training, the network over-trained and the Dice coefficient was bigger on the training data than on the validation or testing data. When evaluating the best outcomes by training the network on the original dataset, the Dice coefficient reached 83 %, but it got to 51 % on validation data immediately after 50 epochs and kept oscillating with minor changes around 50 %. After displaying the predictions of the network, we found out that the marked exudates seemed to be similar to the actual mask of the exudates, but some were not accurately segmented, and the network described many pixels as false positive. False positive pixels were found also on images, which did not contain exudates, meaning on healthy parts of retina. In case of DDR database, we had at disposal more training data. However, we achieved worse outcomes with adaptive optimizers as in case of E-Optha Ex database. Therefore, we used the SGD optimizer, which extended the training process. We gradually reduced the learning rate in the process of training to prevent saturation of the loss function. This resulted in a prolonged training process, and we had to train the network for 1 000 epochs. The outcomes of this process (see Table 2) were comparable with the outcomes in the article by Tao et al. (2019) [6]. As can be seen in the Fig. 6b, the Dice coefficient tended to keep increasing, so to improve the success rate, a longer training time is required.

The outcomes of the network training only on images with exudates turned out better; although, the network over-trained

again. The Dice coefficient reached 95 % on training data during 400 epochs; however, in case of validation it reached only 62 %, that is better than previously. That the network validation outcomes were better when training on only images with exudates suggests that the original dataset was heavily imbalanced. Saturation of Dice coefficient happened when training on both datasets. For this reason, we thought that the problem is in the network or in the data itself. After detailed inspection of the network structure and double-checking the individual layers, we did not find the error. Therefore, we proceeded to analysis of the training dataset.

After closer analysis of the images, we found that in some images the exudates marked by doctor were not in exactly the same place - they were slightly displaced than in the actual image, or were not clearly visible; the network was, therefore, unable to find the correct signs. To improve the outcomes of the segmentation, leaving out such blocks from the training dataset might probably help. Another possible adjustment of the training dataset might be leaving out the blocks that contain only small amount of pixels marked as exudates (for example less than 5). And of course, further augmentation of the dataset might help to improve the outcomes.

References

- Tsiknakis N et al. Learning for Diabetic Retinopathy Detection and Classification Based on Fundus Images: A Review. *Computers in Biology and Medicine* Vol. 135, 2021, p. 104599.
- [2] Kurilová V, Goga J, Oravec M et al. Support vector machine and deep-learning object detection for localization of hard exudates. *Sci Rep* 11, 16045 (2021).
- [3] Kaur Jaskirat, Mittal Deepti. A generalized method for the segmentation of exuda-tes from pathological retinal fundus images. *Biocybernetics and Biomedical Engineering*. 2018 (38) (1), pp. 27–53. ISSN 0208-5216.
- [4] Kou C et al. An Enhanced Residual U-Net for Microaneurysms and Exudates Segmentation in Fundus Images. *IEEE Access.* 2020 (8), 185514–185525.
- [5] Ronneberger 0 et al. U-Net: Convolutional Networks for Biomedical Image Segmentation. Medical Image Computing and Computer-Assisted Intervention (MICCAI). Springer, 2015, Vol. 9351, pp. 234–241. Available from: http://lmb.informatik. uni-freiburg.de/Publications/2015/RFB15a.
- [6] Tao Li et al. Diagnostic assessment of deep learning algorithms for diabetic retinopathy screening. *ScienceDirect*. 2019
- [7] Codenotti M et al. GLOBAL PERSPECTIVES: Treatment of Hard Macular Exudates in Diabetic Retinopathy. 2010, Available from: https://retinatoday.com/articles/2010jan-feb/global-perspectives-treatment-of-hard-macularexudates-in-diabetic-retinopathy

Neuroevolúcia autonómneho vozidla

Bc. Marko Chylík, prof. Ing. Ivan Sekaj, PhD.

Ústav robotiky a kybernetiky, FEI STU v Bratislave Ústav robotiky a kybernetiky

xchylik@stuba.sk

Abstrakt:

Cieľom práce je vytvoriť model autonómneho vozidla v simulačnom prostredí CARLA, ktoré je navrhnuté priamo na účely výskumu autonómnych vozidiel. Vozidlo bude schopné vnímať okolité prostredie a na základe podnetov z neho upravovať svoje jazdné vlastnosti tak, aby splnilo vopred preddefinovanú úlohu. V rámci vnemu prostredia sa venujeme spracovaniu obrazu prednej kamery, ktorý použijeme na detekciu čiar pomocou hlbokej segmentačnej neurónovej siete.

Práve čiary budú oporným bodom pre orientáciu vozidla v priestore. Priestor budeme zároveň monitorovať radarom, ktorý bude hľadať prekážky v okolí vozidla. V neposlednom rade musíme vhodne integrovať navigačný systém, na základe ktorého bude vozidlo dostávať informácie o bodoch, ktoré vedú k žiadanému cieľu.

Riadenie pohybu bude mať za úlohu klasická viacvrstvová neurónová sieť, trénovaná evolučnými algoritmami, ktoré optimalizujú správanie siete tak, aby čo najviac korelovala s definovanými kritériami.

Takto natrénované vozidlo následne vyskúšame nechať prejsť rôznymi testovacími scenármi aby sme overili robustnosť riadiacej neurónovej siete.

Kľúčové slová: Autonómne vozidlo, neuroevolúcia, segmentačná neurónová sieť, CARLA, evolučné algoritmy

1 Úvod

Autonómne vozidlá. Niečo, čo je tak blízko a ďaleko zároveň. Ľudstvo sa už dlhšiu dobu zaoberá tým, ako zvýšiť bezpečnosť na cestách. V dobe expandujúcich technológii sa veľká nádej vkladá do vývoja autonómnych vozidiel. Ich vývoj sa rok od roka zintenzívňuje a je len otázkou času, kedy budú realitou.

Tieto autá zároveň prinesú okrem bezpečnosti väčší komfort a ak sa podarí overiť ich stopercentnú funkcionalitu, eliminujú tým chybovosť vodiča. Stroje nepoznajú únavu, či zlé rozpoloženie. Kladných dôvodov, prečo takéto auto má zmysel, je určite mnoho a každý si vie povedať, prečo práve jemu autonómne vozidlo uľahčí život.

Cieľom našej práce bude priblížiť problematiku a simulačne vyvinúť autonómne vozidlo, ktoré bude schopné reagovať na podnety okolitého sveta správne a bez zásahu ľudského šoféra dopraviť bezpečne posádku na požadované miesto. V práci budeme používať také riešenia, ktoré budú odzrkadľovať reálne možnosti aby bolo možné tento simulačný model čo najjednoduchšie preniesť do reálneho sveta. Kľúčom k autentickosti je určite dobrý simulátor – CARLA. Ten ponúka ovládanie vozidla ako v realite. To zahŕňa reálnu fyziku vozidiel a v neposlednom rade niekoľko kamier, či snímačov s vlastnosťami reálnych zariadení.

Technológie ponúkajú omnoho lepšie možnosti vnímania okolitého sveta (rôzne kamery, senzory, či iné dáta) ako ľudské vnemy, avšak ich fúzia je často komplikovaná. Práve na tomto zlyhávajú striktné algoritmy, ktoré riadia vozidlá pomocou preddefinovaných pravidiel. Niekedy to ale nestačí a rozhodnutie na základe týchto algoritmov nie je správne. Čo ak by sme autu dali schopnosť sa na základe jeho vlastných predošlých znalostí rozhodnúť sa aký krok bude ten správny?

Použitie neuroevolúcie pri učení riadiacej neurónovej siete odsimuluje prirodzenú evolúciu pri riadení vozidla podobne, ako to je v autoškole pri učení riadenia vozidiel ľuďmi. Zároveň pomocou nej vieme presne určiť dôležitosť kritérií pri riadení (dodržiavanie pruhov, vyhýbanie sa prekážkam, ...)

Zároveň trošku poodkryjeme aj metódy spracovania obrazu, či hlbokej neurónovej siete, ktorú použijeme na detekciu čiar v reálnom čase a bude tvoriť majoritnú časť vnemu prostredia vozidla.

2 Vnem prostredia

Každé reálne vozidlo, ktoré disponuje autonómnym riadením potrebuje k svojej činnosti viditeľné čiary. Technologickí giganti už dlhé roky používajú čiary ako ohraničenie zóny, v ktorej sa môže pohybovať, pričom na zmenu pruhu musí mať vozidlo dôvod (analógia lokálnej navigácie v CARLE).

Na detegovanie aktuálneho pruhu použijeme prednú kameru pripevnenú k vozidlu. Obraz z nej použijeme ako vstupný obraz pre segmentačnú neurónovú sieť. Cieľom je získať pozíciu čiar na tomto vstupnom obraze. V projekte využívame dve segmentačné siete, **MobileNetV3Small** a **Unet++**. Každá ponúka iné výhody.

Ďalej sa budeme venovať aj radarovému diaľkomeru a ďalším údajom o jazde vozidla, ktoré pomôžu v celkovom merítku k zvládnutiu nečakaných situácii.

2.1 Dataset

Dataset je základ pre natrenovanie neurónových sietí. Tento obsahuje cez tritisíc trénovacích a okolo stovky testovacích obrázkov. Na trénovanie segmentačnej siete potrebujeme dataset v tvare vstupný obraz – žiadaná segmentovaná maska. Na zvýšenie robustnosti natrénovaných sietí je vhodné obrázky

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

vhodne augmentovať. Na augmentáciu používame najmä metódy spracovania obrazu, ktoré sa venujú **zmenám farebnej** škály, ostrosti a kontrastu a tak simulujeme reálny priebeh, kedy nevieme úplne stopercentne garantovať kvalitu obrazu podobne, ako je tomu v datasete. Otáčanie obrazu či priblíženie augmentáciou nezvýšilo presnosti siete.



Obrázok 1: Ukážka datasetu

2.2 Metrika siete

V prípade segmentačných sietí nám klasické metrické metódy na vyjadrenie chyby neurónovej siete nebudú stačiť. Ako metriku kvality preto použijeme **Dice multi**. Je to metóda na určenie presnosti segmentačnej siete, ktorá hodnotí prekrytie masky (pre nás ideálnej segmentácie) a zároveň aj reálnej segmentácie. Vieme ju vyjadriť ako:

$$Dice_{MULTI} = 2 * \frac{S_p}{S_M + S_S} \tag{1}$$

, kde S_p je plocha prekrytia segmentovanej oblasti a masky, S_M je plocha masky a S_S plocha segmentácie.

2.3 UNet++

Táto sieť bola vyvinutá pre medicínske účely, kde sa pomocou nej detegovala prítomnosť rôznych chorôb na snímkach. UNet (prvá verzia) je autoenkóder na segmentáciu obrazu, rozdelený na úrovne (stupne) kódovania, prepojenie medzi kódovacou a dekódovacou časťou je na úrovni stupňa kódovania. Ako jeden z mála modelov využíva transponovanú konvolúciu – *dekonvolúciu* pri dekódovaní. Jej vylepšená verzia, ktorú využívame aj my používa ešte tri dodatočné vylepšenia oproti pôvodnej verzii – *premosťovacie cesty* medzi jednotlivými konvolučnými vrstvami, *prepoje* medzi vrstvami nerovnakej úrovne a najdôležitejším prídavkom je *šírenie čiastkových výsledkov segmentácie* z podvrstiev. V praxi to znamená, že celkový výsledok segmentácie sa štatisticky spracúva z viacerých segmentácii, nie len z finálnej vrstvy. Štruktúra siete je znázornená na obrázku nižšie:





Obrázok 2: Unet++ L4

Výhodou tejto siete je možnosť použitia enkóderu z inej, už existujúcej siete. V našom prípade využívame ako enkóder použijeme *ResNet34* s predtrénovanými váhami zo siete *ImageNet*. Trénovanie potom netrvá dlho, keďže sieť stačí vyladiť do vhodnej podoby. Od Unet++ očakávame presnú segmentáciu, avšak kvôli veľkému počtu konvolučných filtrov bude segmentácia zrejme nedostatočne rýchla.

2.4 MobileV3Small

MobileV3Small je typologicky úplne inou sieťou, ako predošlá UNet++. Bola vyvinutá najmä pre vnorené systémy, prípadne mobilné telefóny, kde sa dala použiť pri detekcii objektov v reálnom čase. Potrebuje veľmi malý výpočtový výkon a napriek tomu dosahuje presné výsledky vo veľkej rýchlosti. Prelomový prvok Mobile siete (už vo verzii V1) je *hĺbková konvolúcia*. Tá zmenší počet parametrov siete a urýchli tok dát. Ďalším významným prvkom je využitie SE (Squeeze and excitation) blokov, ktoré umožňujú dynamickú rekalibráciu medzi kanálmi obrazu skoro bez zvýšenia výpočtovej náročnosti. Sieť vyzerá nasledovne:



$28^2 \times 24$	bneck, 3x3	88	24	-	RE	1
$28^2 \times 24$	bneck, 5x5	96	40	1	HS	2
$14^2 \times 40$	bneck, 5x5	240	40	1	HS	1
$14^2 \times 40$	bneck, 5x5	240	40	1	HS	1
$14^2 \times 40$	bneck, 5x5	120	48	1	HS	1
$14^{2} \times 48$	bneck, 5x5	144	48	1	HS	1
$14^2 \times 48$	bneck, 5x5	288	96	1	HS	2
$7^2 \times 96$	bneck, 5x5	576	96	1	HS	1
$7^2 \times 96$	bneck, 5x5	576	96	1	HS	1
$7^2 \times 96$	conv2d, 1x1	-	576	1	HS	1
$7^2 \times 576$	pool, 7x7		-	ੁ	-	1
$1^2 \times 576$	conv2d 1x1, NBN	-	1024		HS	1
$1^{2} \times 1024$	conv2d 1x1, NBN	-	k	-	-	1

Obrázok 3: MobileNet V3

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Z obrázka vyššie vidíme, že majoritná časť siete je zložená z blokov typu "bottleneck" – sú to konvolučné filtre upravené na extrahovanie príznakov a zároveň redukciu toku dát. V tabuľke vidíme, či je použitá technológia squeeze and excitation (**SE**), použitú nelinearitu (**NL**) a veľkosť konvolučného kroku (**s**).

Od siete si sľubujeme rýchlejšiu detekciu, pričom rozdiel v kvalite nemôže byť veľký.

2.5 Výsledky segmentácie



Obrázok 4: Priebeh trénovania sietí



Obrázok 5: Porovnanie segmentácii na validačných dátach

Na predošlých obrázkoch vidíme porovnanie presností sietí. Z neho vyplýva, že mierne lepšie obstála **Unet++**, ktorá dosiahla vyššiu presnosť počas trénovania a aj pri optickej kontrole výsledkov na Obrázok 5 sa javí ako presnejšia. Druhá sieť síce nedisponuje až takou presnosťou, na základe Tab. 1, kde vidíme porovnanie časov segmentácii vieme povedať, že je omnoho rýchlejšou, ako prvá menovaná. V real-time aplikácii potrebujeme, aby bola segmentácia čo najrýchlejšia a preto budeme ako segmentačnú sieť používať práva **MobileNet V3 Small.** Priemerný čas segmentácie je 30 – 40 milisekúnd, čo je vyhovujúce a môžeme prejsť ďalej.

T 1	1	ъ.	,	~		
Tah		Prie	merne	Casy	segmen	tacie
I uo.	1.	1 1 1 1	/menne	ousy	beginen	uuuu

Názov siete	Priemerný čas [s]	Najrýchlejší [s]	Najpomalší [s]
Unet++	0,119	0,095	0,181
MobileNet V3 Small	0,033	0,017	0,062

2.6 Interpretácia čiar

V súčasnej podobe máme informáciu, kde sa čiary nachádzajú na vstupnom obraze – dvojrozmernom priestore o veľkosti $\mathbf{u} \times \mathbf{v}$. Pomocou inverzného mapovania a predpokladu, že čiary budú ležať na zemi (poznáme tým pádom z súradnicu) pretransformujeme tento dvojrozmerný priestor do trojrozmerného (X, Y, Z), kde pozíciu čiar budeme vedieť vyjadriť v presnej vzdialenosti voči vozidlu v metroch:

$$\begin{pmatrix} X \\ Y \\ Z \end{pmatrix} = \frac{h}{R_C(0, 1, 0)^T (u, v, 1)^T} \begin{pmatrix} u \\ v \\ 1 \end{pmatrix}$$
(2)

, kde h je výška kamery umiestnenia kamery, R_c je rotačná matica natočenia kamery. Tento prepočet je však správny len pre objekty umiestnené na zemi, keďže sme aplikovali spomínané zjednodušenie.

Takto prepočítame teda všetky body z dvojrozmerného priestoru do trojrozmerného. Následne polynomiálnou aproximáciou tretieho stupňa získame predpis funkcie, ktorá aproximuje ľavú a pravú čiaru.

$$y_{LINE}(x_{LINE}) = a_3 x_{LINE}^3 + a_2 x_{LINE}^2 + a_1 x_{LINE} + a_0$$
(3)

Po zisku tejto aproximácie vieme vypočítať prislúchajúcu súradnicu y pre ľubovoľné x v okolí vozidla. Musíme si uvedomiť, že bod (0, 0, h) prislúcha umiestneniu kamery a tak všetky prepočty budú vzhľadom na tento bod.



Obrázok 6: Ukážka aproximácie

3 Neuroevolúcia riadenia

Po tom, ako sme schopní vnímať okolité prostredie, musíme nastaviť simulátor. Ten nastavíme do synchrónneho režimu s fixnou periódou vzorkovania. Tým zabezpečíme, že máme plnú kontrolu nad simuláciou a neunikajú žiadne dáta. Zároveň vieme špecifikovať, akú časovú periódu znamená jeden simulačný krok.

Autonómia vozidla bude spočívať v klasickej viacvrstvovej (MLP) sieti, ktorá bude riadiť natáčanie kolies. Ako trénovací scenár použijeme úsek z diaľnice, kde bude cieľom vozidlo naučiť sledovať čiary a držať sa v nich a na základe navigačných dát sa dostať zo štartu do cieľa.

3.1 Ovládanie vozidla

Vozidlo sa v CARLA simulátore ovláda na koncepte inšpirovanom realitou. To znamená, pridávanie plynu na zrýchlenie vozidla, brzdenie na spomalenie a samozrejme natáčanie kolies na riadenie smeru.

Riadenie rýchlosti bude prebiehať za pomoci integrovaného riadenia agentom, ktorý využíva PID regulátor na udržovanie žiadanej konštantnej rýchlosti za použitia kombinácie pridávania plynu a brzdenia. Použijeme automatickú prevodovku a natáčanie vozidla (riadenie kolies) vieme vyjadriť pomocou vzťahu (4):

$$s_k = s_{k-1} + s_{NN}, \ s_k \in <-0.8; \ 0.8 >, \ s_{NN} \in <-0.1; \ 0.1 > (4)$$

Pričom vieme, že s_k je natočenie v aktuálnom kroku, s_{k-1} natočenie v minulom kroku a s_{NN} je výstupné natočenie z neurónovej viacvrstvovej siete.

3.2 MLP siet'

MLP sieť si môžeme predstaviť ako na Obrázok 7 – v práci používame rôzne vstupy, pričom sú všetky normalizované do < -1; 1 > na základe ich vlastných kritérii. V experimentoch následne overíme vplyvy jednotlivých vstupov na kvalitu riadenia. MLP sieť je zložená z väčšieho počtu neurónov, pričom každý neurón vieme vyjadriť ako:

$$o_j = tanh\left(\sum_{i=1}^n w_{ji}x_i + b_j\right)$$
(5)

Zo vzťahu (5) vyplýva, že ako aktivačnú funkciu použijeme **hyperbolický tangens** – jeho definičný obor je < -3; 3 >.



Obrázok 7: Štruktúra MLP siete

3.3 Vstupy

- *Detekcia čiar* (6) tri údaje o ypsilonovej pozícii l'avej a pravej čiary v rovine kamery, 10 metrov pred vozidlom a 20 metrov pred vozidlom.
- RADAR (3) po jednom údaji o priemernej dĺžke nameranej radarom, lúče, ktoré majú smerujú do zeme boli ignorovné: (v zátvorke rozsah azimutov):
 - pred vozidlom (< -10° ; 10° >)
 - vľavo (< -45° ; 25° >)
 - vpravo (< 25°; 45° >)
- *Navigácia (2)* spracované navigačné dáta zo simulátora. Normalizovaná chyba v osiach x a y.
- *Binárne vstupy (4)* vstupy, ktoré môžu mať len hodnoty -1, 0 alebo 1:
 - Podľa pozície čiary. Ak je auto bližšie k ľavej, výstup je 1, ak k pravej, výstup je -1.
 - Podľa najbližšej prekážky. Ak je vľavo, -1, ak vpravo, 1 a ak pred vozidlom, tak 0.
 - Podľa rýchlosti v smere k cieľu. Ak auto mieri na cieľ, výstup je 0, inak opačne podľa smeru ktorým by mal ísť
 - Rozdiel medzi odporúčaným natáčaním a našim aktuálnym. Ak je odporúčané viac, výstup je -1, ak menej, tak 1.
- Metrika (2) aktuálne natáčanie a natáčanie kolies 10 krokov späť. Normalizované podľa povoleného rozsahu natáčania.
- Pozícia v rámci pruhu (1) vzdialenosť od stredu pruhu. Normalizované podľa šírky pruhu.

3.4 Použitie evolučných algoritmov na učenie NS

Ako sme spomínali v predošlých kapitolách, silnou stránkou použitia evolučných algoritmov v tomto prípade bude, že vieme jednoduchou parametrizáciou zmeniť želané správanie vozidla. Našim hlavným cieľom bude, aby vozidlo bolo schopné prejsť vopred určenú trasu a popri tom sa držať v pruhu, k čomu mu pomôžu najmä detegované čiary.

Hľadané parametre, ktoré budú tvoriť jedinca, budú váhy synapsii a biasy neurónov. Ich počet samozrejme závisí od zvolenej štruktúry NS a vybraných vstupov. V prípade použitia MLP siete z Obrázok 7 pôjde o **311** neznámych parametrov, zložených z 21 biasov a 290 váh. Rozsah parametrov závisí od definičného oboru využitej nelineárnej funkcie v neuróne.

Vyskladáme populáciu genetického algoritmu o veľkosti N jedincov. Pre rýchlejšiu konvergenciu budeme generovať parametre len v 10% veľkosti def. oboru. Genetický algoritmus následne prispôsobí potrebné váhy/biasy tak, aby bola fitness funkcia optimalizovaná, na čo môže použiť celý definičný obor aktivačnej funkcie *tanh*. Blokovú schému nášho genetického algoritmu uvádzame na Obrázok 8. Fitness funkcia má nasledovný tvar:

$$fit = 5 * crossings + 5000 * bad - (3)$$
$$* range + 2500 * goals)$$
(6)

, pričom *crossings* je počet prejdení cez čiaru do iného pruhu, *bad* je kolízia, prípade iná situácia z ktorej sa auto nevie dostať. *range* je prejdená vzdialenosť a *goals* počet dosiahnutých (čiastkových) cieľov.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

Start Počiatočné populácia N Procovid 1 Najlepší jedine: z posiednej populácia Pracovid 1 Nuřevělaváne: Zjednotnie populácie n + k + m = N

Obrázok 8: Bloková schéma genetického algoritmu

3.5 Priebeh trénovania

Aby sme overili vplyv vstupov na kvalitu riadenia, vykonáme krátky experiment, pri ktorom vylúčime niektoré hlavné vstupy, viac o nastaveniach GA uvádzame v tabuľke:

Tab. 2: Porovnanie experimentov

Najlepšia fitness	Počet cyklov	Detekcia čiar	Radar	Navigácia
-15721	30	áno	nie	nie
-15720	50	nie	áno	nie
-15723	50	nie	nie	áno
-15708	25	áno	nie	áno
-15722	30	áno	áno	áno

Ako naznačuje tabuľka vyššie, vozidlo sa nám podarilo naučiť vždy aj pri použití jedného z hlavných vstupov. Ostatné vstupy, ktoré sme bližšie opisovali v 3.3, boli použité pri všetkých experimentoch. Vyplýva to z fitness funkcie nižšej ako -15000, čo znamená, že auto dostalo odmenu za dosiahnutie všetkých čiastkových cieľov. Na obrázku nižšie znázorňujeme priebeh optimalizácie.



Obrázok 9: Porovnanie optimalizácie genetických algoritmov

Sekcia 3. Robotika a kybernetika

3.6 Výsledky

Najlepším kritériom porovnania je však jazda na testovacej dráhe. To, že sa auto naučí jazdiť na trénovacej dráhe s vysokou kvalitou, neznamená, že toto riešenie je aj robustné. Najprv overíme, či auto zvládne trénovaciu dráhu a následne experimenty porovnáme aj na testovacích dráhach.



Obrázok 10: Porovnanie jázd vozidla na trénovacej dráhe

Ako vidíme na Obrázok 10, všetky experimenty prešli trénovaciu dráhu bez problémov, čo sme očakávali už na základe fitness funkcii týchto riešení.



Obrázok 11: Porovnanie jázd vozidla na testovacej dráhe

Keď sa však pozrieme na Obrázok 11, kde sme použili základnú testovaciu dráhu, vidíme, že väčšina z navrhnutých riešení nedisponuje dostatočnou robustnosťou. Z neho vyplýva, že využitie fúzie detekcie čiar a navigácie zabezpečí žiadanú robustnosť pre túto dráhu. Tieto dva hlavné vstupy sú tak minimálne potrebné na autonómne riadenie natáčania vozidla bez okolitých objektov. Zaradenie radaru však nezhorší kvalitu riadenia pri testovacej trase a zároveň je priam potrebný pre ďalšie testovacie scenáre na detekciu potenciálnych nečakaných kolízii s inými objektami.

Na overenie tejto hypotézy použijeme pokročilú dráhu, ktorá sa nachádza v mestskej časti. Pri nej overíme, či sme

Fakultná prehliadka prác, 27. apríl 2022

Fakulta elektrotechniky a informatiky STU v Bratislave

skutočne schopní vnímať neznáme prostredie a riadiť dostatočne kvalitne natáčanie vozidla. V mestskej časti sa nachádzajú semafory, stĺpy, či budovy, ktoré môžu spôsobiť nečakané kolízie. Zároveň akčné zásahy natáčania budú musieť byť rádovo vyššie ako pri predošlých experimentoch. Rýchlosť vozidla upravíme na 40 km/h.



Obrázok 12: Porovnanie jázd vozidla na pokročilej testovacej dráhe

Výsledky na Obrázok 12 potvrdzujú hypotézu, ktorú sme vyjadrili po poslednom experimente. Iba fúzia všetkých vstupov vedie k dostatočne robustnému riadeniu natáčania, ktoré je aplikovateľné v ktoromkoľvek mieste na mape. Tento scenár zahrňoval aj problematiku nedetekovateľných čiar, v tej chvíli sú tieto vstupy "vypnuté" a natáčanie funguje priamo podľa ostatných vstupov.



Obrázok 13: Vozidlo so všetkými vstupmi na pokročilej testovacej dráhe

Obrázok 13 ukazuje detail cesty najlepšieho riešenia. Vidíme tu prvý krát aj optimálnu cestu, ktorá je vypočítaná na základe navigačného systému. Tento obrázok v určitých momentoch ukazuje veľmi zaujímavé výsledky. Auto totiž nadobudlo schopnosť akejsi predikcie, a vidíme, že k zabáčaniu dochádza častokrát už skôr, ako auto dostane takýto údaj z lokálnej navigácie. Zároveň je pekne viditeľný vplyv detekcie čiar. Auto sa priblížilo na ľavú, (a potom aj pravú) čiaru po prvej, najostrejšej zákrute. Tam detekcia čiar nebola možná, avšak po jej opätovnej aktivácii sa auto opäť spoľahlivo dostalo do stredu pruhu a dokázalo prejsť celou testovacou trasou.

4 Záver

V práci sme overili možnosť vnemu prostredia pomocou segmentačnej neurónovej siete, ktorá dokázala vo veľkej rýchlosti a zároveň veľmi presne ohraničovať jazdný pruh, v ktorom sa vozidlo nachádza. Úspešne sme taktiež integrovali ďalšie senzory a dáta zo simulátora CARLA a vykonali ich postupnú fúziu, ktorá slúžila ako základ pre riadiacu viacvrstvovú neurónovú sieť.

Pomocou princípu neuroevolúcie sme túto sieť naučili na základe rôznych vstupov reagovať na podnety z prostredia a prejsť trénovacou dráhou bez problémov, pričom neuroevolúcia zabezpečila veľmi rýchlu konvergenciu k minimu fitness funkcie.

Aby sme však overili robustnosť natrénovaných sietí, použili sme testovaciu dráhu, ktorá spočívala prechodom cez diaľničný úsek, avšak v opačnom smere. To znamená, že auto muselo zatáčať aj opačným smerom ako bolo natrénované, čo spôsobilo (očakávane) komplikácie pri niektorých konfiguráciách. Najlepšie podľa tejto testovacej dráhy vyšla fúzia detekcie čiar s navigáciou, v tesnom závese nasleduje fúzia všetkých vstupov. Ostatné konfigurácie neboli schopné prejsť takouto dráhou.

Preto sme na záver zaradili náročnejšiu testovaciu dráhu. Tá spočívala v prechode mestskou časťou, kde je potrebné zabáčať omnoho častejšie a prudšie. Sú tu ďalšie nepoznané objekty, ktoré môžu spôsobiť kolíziu a v neposlednom rade v niektorých miestach nedisponujeme čiarami.

Pri nej sme overili, že najrobustnejšou konfiguráciou sa stala podľa očakávaní **fúzia všetkých vstupov** (All). Tá zabezpečila bezproblémový prechod všetkými dráhami, pričom v niektorých prípadoch zvládol dosiahnuť ciele lepšie ako boli plánované lokálnou navigáciou, čo je presne jedným zo základných prvkov autonómie. Tým sme overili, že koncept neuroevolúcie je vhodný na riešenie takéhoto typu úlohy.

Ďalším krokom k plnej autonómii vozidla by bolo riadenie rýchlosti a zabránenie kolíziam vhodnou úpravou rýchlosti vozidla. Úspešnou fúziou takéhoto autonómneho systému s tým, ktorý sa nám podaril navrhnúť teraz, by sme docielili, že auto dosiahlo plnú autonómiu.

Pod'akovanie

Ďakujem prof. Ing. Ivanovi Sekajovi, PhD. za jeho odborné rady, ktoré výraznou mierou prispeli k nájdeniu riešení pre dané úlohy.

Literatúra

- KVASNIČKA, V. BEŇUŠKOVÁ, Ľ. POSPÍCHAL, J. – FARKAŠ, I. – TIŇO, P. – KRÁĽ, A. 1997. Úvod do teórie neurónových sietí. Bratislava, IRIS, 1997. 285 s. ISBN 80-887-77830-1
- [2] MACH, M. 2009. *Evolučné algoritmy: prvky a princípy*. Košice: TUKE, 2009. 250s. ISBN 978-80-8086-123-0.
- [3] SEKAJ, I. 2005. Evolučné výpočty a ich využitie v praxi. Bratislava, IRIS, 2005. ISBN: 80-89018-87-4.
- [4] SEKAJ, I. 2021. TOOLBOX v4 GENETICKÉ ALGORITMY, používateľká príručka
- [5] SHER, GENE I. 2013. Handbook of Neuroevolution Through Erlang, New York: Springer-Verlag 2013. ISBN 978-1-4614-4463-3. Chapter 4
- [6] THEERS, M. 2021. Algorithms for automated driving. Chapter 1: LANE DETECTION. Dostupné online: https://thomasfermi.github.io/Algorithms-for-Automated-Driving/LaneDetection/LaneDetectionOverview.html
- [7] THEERS, M. 2021. Lane Detection for Carla Driving Simulator: DATASET. Dostupné online: https://www.kaggle.com/datasets/thomasfermi/lanedetection-for-carla-driving-simulator
- [8] VASILEV I., 2019. Advanced Deep Learning with Python: Design and implement advanced next-generation AI solutions using TensorFlow and PyTorch, 2019. ISBN: 978-1-78995-617-7
- [9] Albumentations.ai: Mask augmentation for segmentation, 2022. Dostupné online: https://albumentations.ai/docs/getting_started/mask_aug mentation/
- [10] Understanding semantic segmentation with UNET. 2019. Dostupné online: https://towardsdatascience.com/understanding-semanticsegmentation-with-unet-6be4f42d4b47
- [11] RONNEBERGER O., FISCHER P., BROX T., 2015. U-Net: Convolutional Networks for Biomedical Image Segmentation. arXiv: 1505.04597, 2015. Dostupné online: https://arxiv.org/pdf/1505.04597.pdf
- [12] ZHOU Z., 2018. UNet++ A Nested U-Net Architecture for Medical Image Segmentation. arXiv: 1807.10165, 2018. Dostupné online: https://arxiv.org/pdf/1807.10165.pdf
- [13] LAMBA H., 2019. U-Nets with ResNet Encoders and cross connections. TowardsDataScience 2019. Dostupné online: https://towardsdatascience.com/u-nets-withresnet-encoders-and-cross-connections-d8ba94125a2c
- [14] JAIN V., 2019. Everything you need to know about MobileNetV3, TowardsDataScience 2019. Dostupné online: https://towardsdatascience.com/everything-youneed-to-know-about-mobilenetv3-and-its-comparisonwith-previous-versions-a5d5e5a6eeaa
- [15] HOWARD A., SANDLER M., CHU G., CHEN L.C., CHEN B., TAN M., WANG W., ZHU Y., PANG R., VASUDEVAN V., LE Q., ADAM H., 2019. Searching for MobileNetV3, arXiv: 1905.02244, 2019. Dostupné online: https://arxiv.org/pdf/1905.02244.pdf
- [16] VAN GANSBEKE W., DE BRABANDERE B., NEVEN D., PROESMANS M., VAN GOOL L., 2019. End-to-end Lane Detection through Differentiable Least-Squares Fitting, arXiv: 1902.00293, 2019. Dostupné online: https://arxiv.org/pdf/1902.00293.pdf

Vizuálna odometria pre lietajúce robotické systémy

Martin Sedláček¹

¹Inžiniersky program robotika a kybernetika, FEI STU v Bratislave

xsedlacek@stuba.sk

Abstrakt – Vizuálna odometria môže byť spoľahlivým a presným zdrojom lokalizácie pre lietajúce robotické systémy. Základom funkčnej realizácie vizuálnej odometrie je správne nastavený kamerový systém a následné spracovanie obrazu. Kalibráciu, spracovanie obrazu a nastavovanie kamier v kontexte s vizuálnou odometriou v tejto práci teoreticky hodnotíme a aj prakticky realizujeme. Naše postupy a riešenia sú overené experimentom na NVIDIA Jetson platforme.

1 Úvod

Vizuálna odometria (VO) je proces určovania polohy robota, je založená na spracovávaní obrazových streamov z jednej alebo viacerých kamier. VO ako taká je podrobne skúmaná problematika v robotike. Medzi hlavnú motiváciu vo výskume tejto metódy určite patrí jej potencionálna vysoká presnosť, nižšia cena potrebného hardvéru a úplná nezávislosť od iných systémov. Pojem VO bol prvý krát použitý v práci pod vedením Nistér [1]. [2]

VO je vhodná pre lietajúce robotické systémy využívané vo vnútorných priestoroch, kde nie je možné sa navigovať pomocou globálneho navigačného systému. VO dokáže zabezpečiť dostatočnú robustnosť, presnosť a kvalitu relatívnej lokalizácie vo vnútorných priestoroch, čo je nevyhnutné pre navigáciu a úspešné plnenie misie.

Cieľom našej práce bude vyriešiť obsluhu kamerového systému, tak aby výstupný obraz bol vhodný pre využitie vo VO, ktorú bude možné nasadiť do autonómnej kvadrokoptéry určenej do vnútorných priestorov. Správne implementovaný kamerový systém, je základom každého úspešného riešenia VO.

Najprv uvedieme všeobecný kamerový systém so zameraním na charakteristiky, s ktorými budeme pracovať v našej práci. Okrem realizovania kalibrácie kamerového stereo páru ponúkneme taktiež naše praktické skúsenosti s konkrétnymi kalibračnými riešeniami a osvedčeným kalibračným postupom. Mimo určovania kalibračných parametrov sa budeme venovať aj nastaveniam kamery ovplyvňujúce finálnu kvalitu obrazu.

Naše návrhy a postupy budeme realizovať na platforme NVIDIA Jetson NX (integrovaná do systému AI Drone Core od firmy Airvolute s.r.o.). Pri využívaní kalibračných údajov kamery taktiež poukážeme na možnosti akcelerácie výpočtovej práce s obrazom na našej platforme. V našej práci budem využívať aj ROS, ktorý zabezpečuje kompatibilitu s ostatným systémom kvadrokoptéry. Aj napriek tomu, že sa v práci nebudeme veľmi explicitne venovať tomu, ako realizovať celý návrh VO. Tak ako demonštráciu funkčnosti kamerového systému použijeme výsledky odhadu trajektórie z takéhoto systému.

2 Vizuálna odometria a lokalizácia lietajúcich robotických systémov

Realizácia VO, ktorá bola použitá v tejto práci je zobrazená na Obr. 1. Systém využíva viaceré softvérové balíky, z ktorých boli niektoré vyvinuté od začiatku, niektoré výrazne upravené opensource riešenia s vhodnou licenciou a niektoré boli vyvinuté v spolupráci s firmou Airvolute s.r.o.. Realizácii celého systému sa nebudeme venovať, keďže to je náplňou inej práce. V krátkosti môžeme zhrnúť, že ako implementáciu algoritmov pre VO bol zvolený Elbrus vyvinutý firmou NVIDIA, ostatné obslužné balíky slúžia najmä na získavanie a spracovanie dát pre túto implementáciu VO.

Pre správnu funkčnosť celého tohto návrhu sme museli zaistiť obsluhu kamier a spracovanie ich dát do takej formy, ktorá bude vhodná pre použitie vo VO. Pri implementácii softvérových balíkov pre obsluhu kamier sme sa museli zamerať na výpočtovú nenáročnosť, správnu synchronizáciu dát a spoľahlivosť.



Obr. 1 Bloková schéma - návrh systému VO, našu prácu sústreďujeme na zelené bloky

Ako sme už naznačili v tejto práci sa budeme venovať najmä kamerovému systému, kalibrácii kamier, nastaveniu kamier a následnému spracovaniu obrazu. Softvérové bloky, ktoré boli využívané v riešení daných tém sú na Obr. 1 zvýraznené zelenou farbou.

2.1 Kamerový systém

Kamerový systém je základ pre každú realizáciu VO. Kamera je vo všeobecnosti senzor, ktorý prevádza 3D priestorové body (scéna) do 2D roviny (snímka). Prevod scény do snímky je realizovaný zachytávaním odrazu svetla z prostredia na snímací čip. Pri dopade svetla na fotobunku snímacieho čipu je generovaný elektrický náboj, ktorý je typicky zosilnení a vyhodnotení. Časový interval počas, ktorého dopadá svetlo na snímací čip nazývame expozičný čas. Expozičným časom je možné priamo korigovať jas snímky. Veľkosť zosilnenia elektrického náboja z fotobuniek je často krát tiež možné korigovať, nastavovaním tohto parametra taktiež upravujeme jas snímky, ale pri príliš vysokých hodnotách zosilnenia môže byť do snímky vnesený zbytočný šum.

Pri využívaní kamier v aplikáciách, kde je potrebné z obrazu zistiť merateľné údaje vzťahujúce sa na snímaný priestor. Je nevyhnutné poznať aj parametre kamery, ktoré nám hovoria o tom ako je 3D bod z priestoru projektovaný na 2D rovinu snímku. Pri uvažovaní kamerového modelu tzv. pinhole, je 3D bod ($P = [X, Y, Z]^T$) zo scény perspektívne projektovaný do roviny obrazu ($p = [u, v]^T$). Perspektívne zobrazenie v tomto prípade môžeme vyjadriť:

$$s * p' = K * [R | t] * P'$$
 (1)

kde *K* je matica intrinzických (vnútorných parametrov kamery), R a t tvoria maticu extrinzických parametrov (R je natočenie a t je translačný vektor), $p' = [u, v, 1]^T$, $P' = [X, Y, Z, 1]^T$, s je váhový koeficient, u a v sú koordináty v prípade snímku pixelov.

Na obrázku Obr. 2. môžeme vidieť zobrazenú perspektívnu projekciu (ktorá platí pre pinhole kamerový model). Optická os kamery vychádza z centra projekcie C, pretína snímku (zobrazovaciu rovinu) v bode c, bod c nazývame hlavný bod.



Obr. 2 Pinhole kamerový model (perspektívna projekcia), fokálna dĺžka f, principiálny bod c, kamerové centrum C [3]

Projekciu bodu P (3D bod v priestore) do zobrazovacej roviny, na Obr. 2 bod p, môžeme vyjadriť takto (ak predpokladáme, že sa kamera nachádza v stredu globálneho súradnicového systému, teda extrinzické parametre kamery by nepredstavovali posun ani rotáciu):

$$p_x = u = f_x * \frac{X}{Z} + c_x \tag{2}$$

$$p_y = v = f_y * \frac{y}{z} + c_y \tag{3}$$

Pinhole kamerový model (1) môžeme rozpísať do kompletného vyjadrenia,

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_x & c_y \\ 0 & 0 & 1 \end{bmatrix} \\ * \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{12} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix}$$
(4)

kde r_{xx} sú prvky rotačnej matice, t_{xx} sú prvky translačného vektora, X a Y a Z predstavujú pozíciu na snímanej scéne, f_{xx} fokálna dĺžka, c_{xx} principiálny bod, u a v je pozícia zobrazeného bodu na zobrazovacej rovine.

Pinhole kamerový model nie je platný pre kamery, ktoré obsahujú šošovku (prípadne objektív so zakrivením). V obraze z takýchto kamier vzniká geometrické skreslenie, ktoré je nutné odstrániť, aby sme dokázali získať skutočnú polohu bodu v priestore. Podľa typu zakrivenia línii obrazu môžeme geometrické skreslenie rozdeliť na súdkovité skreslenie a poduškovité skreslenie. V poduškovitom skreslení sa línie obrazu deformujú smerom do obrazu. V súdkovitom skreslení sa línie obrazu deformujú smerom von z obrazu. [4]



Obr. 3 Typy geometrického skreslenia obrazu [5]

Vyjadrenie (1) môžeme upraviť pridaním distorčnej matice kamery D, tak dané vyjadrenie bude platné aj pre kamery so šošovkami, ktoré obraz skresľujú (váhový koeficient s už neuvádzame).

$$p' = K * D * [R | t] * P'$$
(1)

kde D je matica odstraňujúca skreslenie obrazu, *K* je matica intrinzických (vnútorných parametrov kamery), R a t tvoria maticu extrinzických parametrov (R je natočenie a t je translačný vektor), $p' = [u, v, 1]^T$, $P' = [X, Y, Z, 1]^T$, u a v sú koordináty v prípade snímku pixelov.

Medzi najviac používané distorčné modely patrí tzv. radiálno-tangenciálny model [6] s rôznym počtom radiálnych koeficientov (najčastejšie 2 a 3) a ekvidistančný model. Ekvidistančný model je primárne určený pre kamery s veľmi širokým zorným poľom, najčastejšie kamery tzv. rybie oko [7].

Proces identifikácie intrinzických, extrinzických a distorčných parametrov kamery nazývame kalibrácia kamery. Kalibráciou kamery sa zaoberáme prakticky v nasledujúcich sekciách Správna kalibrácia kamery je veľmi dôležitá pre presné fungovanie vizuálnej odometrie, parametre kamery priamo ovplyvňujú výpočty spojené s určovaným bodov v priestore zo snímok kamery.

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

2.2 OV9281 stereo pár

Základ nášho kamerového systému tvoria kamery OV9281. Kamery sú umiestnené na karbónovej konštrukcii, s horizontálnou vzdialenosťou 10cm. Hrúbka karbónovej konštrukcii je 4mm, čo eliminuje skrúcanie, priehyb a iné deformácie, ktoré by kvalitu stereo páru znižovali.



Obr. 4 OV9281 kamerový stereopár

OV9281 využíva CMOS optický čip a globálnu uzávierku. Kamera je schopná fungovať vo viacerým módoch, ktoré ovplyvňujú rozlíšenie a počet snímok za sekundu. My sme pracovali s módom 0 (rozlíšenie 1280x800 pixelov) módom 2 (rozlíšenie 640x400 pixelov). Výrobca kamery uvádza diagonálny FoV 166°. Stereo pár na synchronizáciu ľavej a pravej kamery používa externý synchronizačný signál. [8]

Kamery OV9281 využívajú objektív, ktorý do ich obrazu vnáša skreslenie. Pre využitie tohto stereo páru vo VO musíme skreslenie identifikovať a odstrániť ho.

2.3 Kalibrácia OV9281 a práca s obrazom

Pri prvej kalibrácii senzora OV9281 sme využívali nástroj Kalibr [9] s tzv. aprilgrid kalibračnou tabuľkou. Výsledky danej kalibrácie, z pohľadu kvalitatívnych ukazovateľov (epipolárna chyba a reprojekčná chyba), vyzerali byť dostatočné, avšak keďže náš stereo pár nie je dokonale zarovnaný sú potrebné aj extrinzické parametre medzi ľavou a pravou kamerou. Extrinzické parametre kamery sú taktiež identifikované nástrojom Kalibr, avšak nedajú sa využiť bez úpravy ďalej. Nástroj Kalibr je určený pre vedeckú komunitu, preto sa môže jeho obsluha zdať trochu komplikovanejšia. Zároveň nástroj Kalibr bol vydaný pre Ubuntu 12.04, čiže inštalácia pre novšie verzie Ubuntu prináša so sebou určité problémy s kompatibilitou, ktorých riešenie je na koncovom užívateľovi a dá sa očakávať, že pre ďalšie verzie Ubuntu bez natívnej podpory Python2 sa kompatibilita Kalibr s daným operačným systémom ešte zhorší.

Na základe uvedených nedostatkov Kalibr sme sa rozhodli hľadať alternatívny kalibračný nástroj. Po prieskume dostupných možností sme vybrali ako vhodný nástroj pre nás camera_calibrate [10], ide o ROS node, ktorý je súčasťou väčšieho balíka ROS modulov image_pipeline. Nástroj je vo svojom jadre postavený na OpenCV moduloch, ktoré postupne extrahujú potrebné údaje pre kalibráciu z obrazových dát a následne vyriešia optimalizačný problém hľadania vhodných kalibračných parametrov kamery. Nástroj camera calibrate taktiež ponúka jednoduché GUI, kde je užívateľ informovaný o aktuálnom stave kalibrácie a jej výsledkov.

Nástroj camera_calibrate podporuje kalibračné tabuľky tzv. šachovnica a ChArUco, zobrazené na Obr. 5. Pri riešení fyzického vyhotovenia tabuľky sme hľadali také vyhotovenie, aby sme zabezpečili stále rovný povrch kalibračného vzoru, nízku odrazivosť a vyhotovenie kalibračného vzoru potlačou (nie fóliou, ktorá sa môže rozťahovať a sťahovať na základe teploty).

Finálne vyhotovenie kalibračnej tabuľky je z materiálu dibond o rozmeru 1.2m výška a 1m šírka. Väčší rozmer tabuľky by mal zaistiť presnú detekciu kalibračného tvaru z obrazu pri rôznych vzdialenostiach. Jedna strana tabule bola využitá na vzor šachovnice, druhá na ChArUco. Na kalibrovanie sme využívali šachovnicu. Kalibráciu kamery OV9281 sme vykonávali na rôznych datasetoch, celkovo najlepšie výsledky sme dosiahli pri postupnom prechádzaní kalibračnej tabuľky celým FoV kamery v vzdialenosti 1.5m od kamery, následne sme sa s kalibračnou tabuľkou približovali ku kamere tak, aby pokryla celý jej FoV. Ako distorčný model kamery sme použili radiálne tangenciálny model s počtom parametrov 4[6]. Testovali sme aj ekvidistančný distorčný model, avšak výsledky boli horšie ako pri prvom uvedenom.



Obr. 5 Kalibračné tabuľky pre kamery (vľavo šachovnica a vpravo ChArUco)

Výsledky kalibrácie aplikované na obraz sú vizualizované na Obr. 6. Obraz C, čiže rektifikovaný a neskreslený obraz využívame v návrhu VO na odhad polohy. Je dobré upozorniť nato, že aplikovaním kalibračných parametrov na pôvodný obraz z nášho stereo páru sme stratili časť zorného poľa kamery za cenu dobre zarovnaných (rektifikovaných) a neskreslených snímok. Avšak táto strata FoV kamery nie je príliš veľká, čiže daný obraz je stále vhodný pre odhad pohybu. Obraz C je taktiež možné použiť na výpočet disparity a následnú generáciu hĺbkovej mapy, dané spracovanie obrazu sme použili ako čiastočnú validáciu našej kalibrácie.

Pri analýze epipolárnej chyby stereo snímkov, sme zistili, že daná chyba je menšia pri strede obrazu (hodnoty oscilujúce v okolí 0.1 závislé na pozícii kalibračného obrazu) a postupne sa zväčšuje smerom ku stranám obrazu (hodnoty oscilujúce v okolí 0.55 závislé na pozícii kalibračného obrazu). Tento jav sme sa snažili odstrániť využitím radiálne tangenciálneho modelu s viacerými parametrami, avšak aj pri 5 parametrovom distorčnom modely sme dosahovali podobných výsledkov ako pri pôvodnom 4 parametrovom modely. Ak by zvýšená chyba pri okrajoch obrazu bola nevhodná pre výpočet vizuálnej odometrie, tak jedno z možných riešení je orezať

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

pôvodný obraz o krajné časti (tie kde je skreslenie najvýraznejšie) a takto orezaný obraz použiť ďalej v procese kalibrácie a na záver vo výpočte VO.



Obr. 6 Vizualizácia výsledku kalibrácie stereo páru OV9281 (A – obraz nerektifikovaný a skreslený, B – celý obraz rektifikovaný a neskreslený, C – platný obraz po rektifikácii s odstráneným skreslením)

Kamery OV9281 je nutné nastavovať (najmä ich zosilnenie a expozičný čas), vždy pred ich použitím, pretože nedisponujú žiadnym automatickým nastavením týchto hodnôt. To, že nastavenie týchto parametrov je kriticky dôležité, môžeme vidieť na Obr. 7, oba tieto snímky nie sú úplne vhodné pre VO, ale jeden je použiteľný a druhý vôbec.



Obr. 7 Príklad rôznych konfigurácií kamery OV9281

V začiatkoch vývoja sme kamery nastavovali manuálne podľa charakteristiky prostredia, avšak časom sa ukázalo, že automatické nastavenie týchto parametrov by bolo veľmi praktické. Zároveň v lietajúcom robotickom systéme, nie je prípustné manuálne nastavovanie parametrov kamery, pred každým štartom misie. Pri skúmaní tejto problematiky sme taktiež analyzovali prácu [11], kde je online automatické nastavovanie parametrov kamery využité k zvýšeniu presnosti systémov závislých od kvality snímok a teda aj vizuálnej odometrie. Na základe prieskumu dostupných možností sme naše riešenie automatického nastavenia kamery postavili na už existujúcom riešení nastavovania expozičného času kamery podľa požadovanej kvality obrazu pomocou jednoduchého regulátora [12].

Naše riešenie sme nazvali easy_auto_exposure. V easy_auto_exposure sa najprv vstupný obraz zredukuje na

240 pixelov na šírku a 160 pixelov na výšku. Ďalej sa vypočíta histogram hodnôt jasu. Daný histogram sa spriemeruje, spriemerovaná hodnota v regulátore vystupuje ako meraná veličina. Jednoduchý KI regulátor reguluje spriemerovanú hodnotu histogramu jasu na požadovanú hodnotu, len pre zaujímavosť naša požadovaná hodnota je 2.5, ale táto hodnota je typická pre naše konkrétne kamery s konkrétnym nastaveným zosilnením. V regulátore vystupuje aj pásmo necitlivosti, ktoré zabezpečuje, aby nebol generovaný akčný zásah v prípade iba malých odchýlok regulovanej veličiny od požadovanej hodnoty. Akčný zásah je realizovaný generovaním ROS servisu, ktorý je posielaný z easy_auto_exposure do gscam (softvérový balík, ktorý je súčasťou využitého riešenia VO na Obr. 1). Gscam daný ROS servis prijme a vykoná zníženie alebo zvýšenie expozičného času kamery. Kvalitu obrazu automaticky nastaveného z easy_auto_exposure vidíme na Obr. 6. Easy_auto_exposure pracuje iba s expozičným časom, čiže zosilnenie kamery musí byť stále nastavované manuálne, avšak toto manuálne nastavenie, podľa našich skúseností, sa dá realizovať dostatočne všeobecne pri prvotnom testovaní kamerového stereo páru (napríklad vo výrobe) a určite nie je nevyhnutné ho meniť pred každým štartom misie. Pri zistení nedostatkov v aktuálnom riešení, existujú aj pokročilejšie metódy online nastavovania kvality obrazu, napríklad v [13] upravujú expozičný čas aj zosilnenie kamery. Ako metriku vyhodnocovania správneho nastavenia kamery, okrem jasu obrazu využívajú aj spätnú väzbu z vizuálnej odometrie o tom, koľko príznakov na obraze je dostupných.

z kamerového stereo páru Obraz rektifikujeme a odstraňujeme jeho skreslenie. Na dané operácie bol použitý balík ae_isaac_ros_image_pipeline, tento balík bol vyvinutý vývojármi firmy Airvolute s.r.o., vychádzali z [14]. Balík vytvára akýsi most medzi ROS a NVIDIA VPI , NVIDIA VPI (Vision Programming Interface) je softvérový balík, ktorý sprístupňuje rôzne algoritmy akcelerované na výpočtových procesoroch NVIDIA Jetson. V našej práci boli využité akcelerované funkcie pre rektifikovanie a odstránenie distorcie z obrazu. Do daného balíku sme museli implementovať podporu pre náš distorčný model (4 parametrový radiálne tangenciálny doplnení funkcionality model). Po sme balík ae isaac ros image pipeline mohli využiť v našom návrhu kamerového systému, príklad vstupu a výstupu obrazu z tohto balíka vidíme na Obr. 6.

2.4 Optimalizácia prenosu obrazových dát v systéme ROS a NVIDIA Jetson

Realizácia systému VO (a realizácia kamerového systému) je určená pre výpočtovú jednotku NVIDIA Jetson, ktorá disponuje okrem CPU aj inými výpočtovými procesormi. V našom návrhu využívame GPU v balíku ae_isaac_ros_image_pipeline na akceleráciu rektifikovania a odstránenie distorcie z obrazu. Taktiež v implementácii VIO Elbrus (od firmy NVIDIA) je do veľkej miery využívaná akcelerácia na GPU.

Po analýze nášho riešenia sme prišli nato, že veľkú časť výpočtového výkonu naše riešenie spotrebuje na prenos video streamom (topicov) v systéme ROS. Pri štandardnom prenose dát v systéme ROS sa prenášajú celé dáta, ktoré musia byť serializované a deserializované, čiže dochádza ku kopírovaniu pomerne veľkých dát, čo zaťažuje CPU. Lepšie riešenie by nemuselo prenášať celé objemné dáta, ale len smerník na už existujúce dáta v pamäti. Avšak pri využití ROS

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

node v systéme ROS takéto prenášanie dát medzi ROS nodami nie je možné. Pri skúmaní problému sme ale narazili na tzv. ROS nodelet, ide o nadstavbu ROS nodu, ktorá umožnuje posielanie dát odkazom na ich adresu bez kopírovania. Funguje to tak, že z ROS nodu sa vytvorí knižnica alebo plugin, ktorý je spustení pod hlavným procesom nazývajúcim sa nodelet manažér. Medzi nodeletami, ktoré sú spustené v rámci jedného nodelet manažéra je umožnené posielanie dát cez odkaz na pamäťové miesto. V našom riešení sme pridali nadstavbu nodeletu pre ROS balíky, ktoré pracujú s obrazovými dátami. Týmto krokom sme zabezpečili, aby úpravy na obraze a jeho prenos mohol byť realizovaný v reálnom čase. Zároveň sme uvoľnili výpočtový výkon pre iné systémy.

2.5 Overenie riešenia

Využitie kamerového systému a práce s obrazom v systéme vizuálnej odometrie budeme demonštrovať experimentom na kvadrokoptére (zobrazená na Obr. 7)., ktorá obsahuje kompletný systém od firmy Airvolute s.r.o. a realizáciu návrhu systému VO z Obr. 1. Experiment sme vykonali v laboratóriu dronov na FEI STU. Keďže v práci sme sa viac zamerali na kamerový systém, tak daný experiment nebudeme kvalitatívne vyhodnocovať exaktnou metrikou.



Obr. 8 Testovacia konfiguráciaUAV s AI Drone Core s kamerovým stereo párom OV9281 na experimenty

Experiment ktorý prezentujeme, je jednoduchá trajektória medzi regálmi. Stereo pár kvadrokoptéry smeruje kolmo na smer pohybu, čiže vždy do regálov. Pre lepší popis sme danú trajektóriu znázornili aj v digitálnej dvojičke laboratória dronov. Dané zobrazenie nám ponúka taktiež lepší kontext, v akom prostredí sme experiment vykonali. Červenou farbou je zobrazená trajektória a zelenou štartovací a končiaci bod.



Obr. 9 Testovacia trajektória v zobrazená v digitálnom dvojčati laboratória dronov

Na základe výsledku experimentu zobrazeného na Obr. 10 môžeme konštatovať, že prezentované postupy a riešenie kamerového systému určeného pre VO boli správne zvolené a realizované. Pri analýze výsledku, vidíme, že trajektória nekopíruje úplne presne trajektóriu definovanú experimentom, avšak takéto odchýlky sú tolerovateľné a stále pomerne dobré. V trajektórii taktiež môžeme vidieť určitú formu menšieho driftu VO, predpokladáme, že tento jav môže vznikať v dôsledku znižujúcej sa presnosti kalibrácie na okrajoch obrazu. Avšak, keďže chyba nie je príliš veľká, nie je nutné upravovať riešenie. Ak by bola chyba väčšia pristúpili by sme k orezaniu obrazu o krajné časti.



Obr. 10 Výstup so systému VO s použitím OV9281 kamerového systému

3 Záver

V tejto práci sme do detailu rozobrali postupy, riešenia a nutné úpravy obrazu a kamerového systému tak, aby ich bolo možné použiť v systéme VO. Okrem teoretickej prípravy sme taktiež zrealizovali konkrétne kroky, ktoré viedli k tomu, aby kamerový systém založený na kamerách OV9281, mohol byť použitý v lietajúcom robotickom systéme.

Keďže palubný počítač je v podstate vnorený systém s obmedzeným výpočtovým výkonom, tak sme sa zamerali aj na výpočtovú optimalizáciu kamerového systému. Kde sme pomocou softvérových balíčkov využili akceleráciu na GPU

Fakultná prehliadka prác, 27. apríl 2022 Fakulta elektrotechniky a informatiky STU v Bratislave

a pomocou nodeletov realizovali posielanie video streamov len ako adresy na pamäť bez kopírovania dát. Aj týmito optimalizáciami sme zaistili to, aby kamerový systém mohol pracovať v reálnom čase a teda mohol byť použitý v systéme VO pre lietajúce robotické systémy.

Prezentované softvérové riešenia a postupy môžu byť využité aj s iným kamerovým stereo párom. Čo je výhoda, keďže v prípade zistenia nedostatkov kamier OV9281 je možné ich pomerne rýchlo nahradiť iným modelom kamier.

Veríme, že sme funkčnosť riešení dostatočne demonštrovali praktickými ukážkami v hlavnej textovej časti práce a nakoniec overili aj praktickým experimentom. Obsluha kamerové systému so zameraním na využitie vo VO bude využitá v inej akademickej práce, ktorá sa zaoberá kompletným návrhom VO pre kvadrokoptéru.

Pod'akovanie

Chceli by sme vyjadriť vďaku vedúcemu práce (Ing. Jozef Rodina, PhD.) za poskytnuté cenné rady. Kolektívu z firmy Airvolute s.r.o. ďakujeme taktiež za sprostredkovanie cenných znalostí a za materiálneho zabezpečenia.

Literatúra

- NISTER, D., O. NARODITSKY a J. BERGEN. Visual odometry. In: Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004 [online]. IEEE, 2004, s. 652-659 [cit. 2022-01-08]. ISBN 0-7695-2158-4. Dostupné z: doi:10.1109/CVPR.2004.1315094
- [2] AQEL, Mohammad O. A., Mohammad H. MARHABAN, M. Iqbal SARIPAN a Napsiah Bt. ISMAIL. Review of visual odometry: types, approaches, challenges, and applications. *SpringerPlus* [online]. 2016, 5(1) [cit. 2022-01-08]. ISSN 2193-1801. Dostupné z: doi:10.1186/s40064-016-3573-7
- [3] PAHLBERG, Tobias. *Registration of multiple ToF camera point clouds*. Luleå University of Technology, 2010.
- [4] SALVI, Joaquim, Xavier ARMANGUÉ a Joan BATLLE. A comparative review of camera calibrating methods with accuracy evaluation. *Pattern Recognition* [online]. 2002, **35**(7), 1617-1635 [cit. 2022-04-17]. ISSN 00313203. Dostupné z: doi:10.1016/S0031-3203(01)00126-1
- [5] RAZALI YUSOFF, Ahmad a Mohd FARID MOHD ARIFF. The effect of varies camera calibration fields on camera parameters. *Jurnal Teknologi*. 2015. Dostupné z: doi:10.11113/jt.v77.6853
- [6] BROWN, Duane C. Close-Range Camera Calibration. *Photogrammetric Engineering*. 1971, (378), 855-866.
- [7] KANNALA, J. a S.S. BRANDT. A generic camera model and calibration method for conventional, wide-angle, and fish-eye lenses. *IEEE Transactions on*

Pattern Analysis and Machine Intelligence [online]. 2006, **28**(8), 1335-1340 [cit. 2022-04-17]. ISSN 0162-8828. Dostupné z: doi:10.1109/TPAMI.2006.153

- [8] *OMNIVISION OV9281-Ov9282*. 2017. Dostupné také z: http://www.ovt.com
- [9] FURGALE, Paul, Jérôme MAYE, Jörn REHDER, Thomas SCHNEIDER a Luc HINZMANN. Kalibr [online]. [cit. 2022-01-13]. Dostupné z: https://github.com/ethz-asl/kalibr
- [10] BOWMAN, James a Patrick MIHELICH. Camera_calibration. *ROS.org* [online]. 2020 [cit. 2022-04-15]. Dostupné z: http://wiki.ros.org/camera_calibration
- [11] ZHANG, Zichao, Christian FORSTER a Davide SCARAMUZZA. Active exposure control for robust visual odometry in HDR environments. In: 2017 IEEE International Conference on Robotics and Automation (ICRA) [online]. IEEE, 2021, s. 3894-3901 [cit. 2022-01-14]. ISBN 978-1-5090-4633-1. Dostupné z: doi:10.1109/ICRA.2017.7989449
- [12] ZHU, Alex. Auto_exposure_control. *Github* [online]. [cit. 2022-04-16]. Dostupné z: https://github.com/alexzzhu/auto_exposure_control
- [13] BÉGIN, Marc-André a Ian HUNTER. Auto-Exposure Algorithm for Enhanced Mobile Robot Localization in Challenging Light Conditions. *Sensors* [online]. 2022, 22(3) [cit. 2022-04-16]. ISSN 1424-8220. Dostupné z: doi:10.3390/s22030835
- [14] *Isaac_ros_image_pipeline* [online]. [cit. 2022-04-16]. Dostupné z: https://github.com/NVIDIA-ISAAC-ROS/isaac_ros_image_pipeline