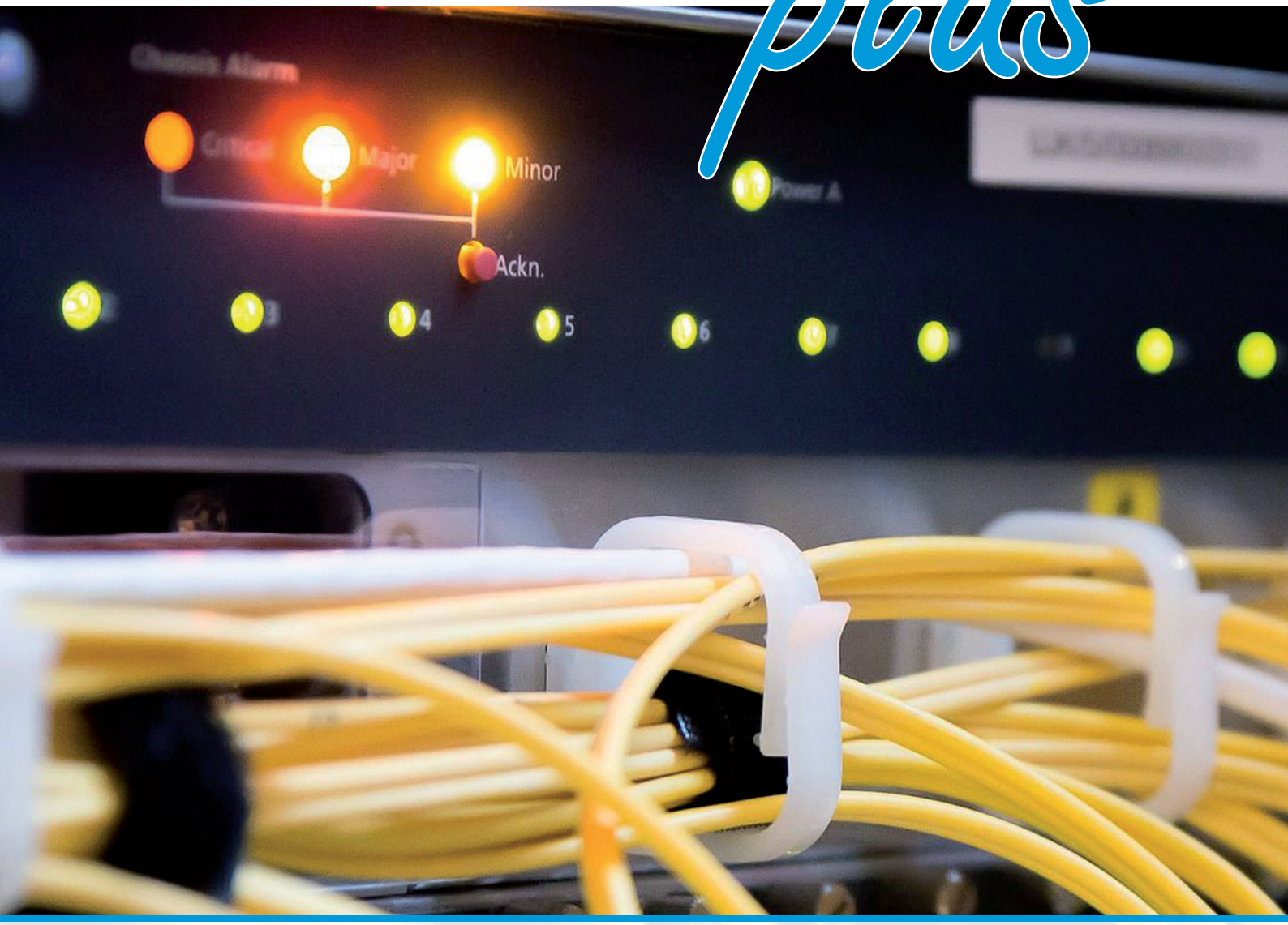
1/2019

atp journal plus



Activities of the Department of Control and Information Systems University of Zilina in the field of industrial and transport processes

reviewed slovak professional magazine for scientific and engineering issues

Aktivity Katedry riadiacich a informačných systémov Žilinskej univerzity v Žiline v oblasti riadenia priemyselných a dopravných procesov

recenzované periodikum vedeckých a inžinierskych publikácií

ISSN 1336-5010

Aktivity Katedry riadiacich a informačných systémov Žilinskej univerzity v Žiline v oblasti riadenia priemyselných a dopravných procesov

Activities of the Department of Control and Information Systems University of Zilina in the field of industrial and transport processes

Odborný garant

prof. Ing. Karol Rástočný, PhD.
Žilinská univerzita v Žiline,
Fakulta elektrotechniky a informačných technológií,
Katedra riadiacich a informačných systémov

Technical guarantee

prof. Ing. Karol Rástočný, PhD.
University of Zilina, Faculty of Electrical Engineering and information technology,
Department of Control and Information Systems

Vydavateľ

HMH s.r.o.
Tavarikova osada 39
841 02 Bratislava 42
IČO: 31356273

Publisher

Spoluzakladateľ

Katedra ASR, EF STU
Katedra automatizácie a regulácie, EF STU
Katedra automatizácie, ChtF STU
PPA CONTROLL, a.s.

Co-founder

IEEE
Czechoslovakia Section
www.ieee.cz

Redakčná rada

Draft committee

Redakcia

Editors office

ATP Journal
Galvaniho 7/D
821 04 Bratislava
tel.: +421 2 32 332 182
fax: +421 2 32 332 109
vydavatelstvo@hmh.sk
www.atpjournal.sk

Ing. Anton Gérer,
šéfredaktor – editor in chief
gerer@hmh.sk

Zuzana Pettingerová,
DTP grafik – DTP graphic designer
dtp@hmh.sk

Dagmar Votavová,
obchod a marketing – sales and Marketing
mediamarketing@hmh.sk

Mgr. Bronislava Chocholová
jazyková redaktorka – text corrector

Datum vydania: jún 2019

ISSN 1336-5010
ATP Journal *PLUS* 1/2019

Obsah

Riadenie priemyselných procesov

Riadenie dopravných procesov

Content

Control of industrial processes

Control of transport processes

HMATOVÁ INTERAKCIA S VIRTUÁLNYM SYSTÉMOM

Jozef Hrbček, Martin Paprčka

Abstrakt
Hmat je jedinečný ľudský zmysel. V porovnaní s ostatnými umožňuje interakciu, obojsmerný tok energie a výmenu informácií medzi prostredím a človekom. Slúži ľuďom na spoznávanie a pozorovanie prostredia, ale aj na manipuláciu s ním. Ľudia sa často pokúšajú manipulovať s objektmi, s ktorými to za normálnych okolností nie je možné. Objekt môže byť príliš ťažký, môže mať vysokú teplotu, vyžarovať radiáciu so smrteľnými účinkami, byť príliš ďaleko alebo neexistuje v hmatateľnej forme, ale len ako súbor informácií. Na dosiahnutie tohto cieľa si ľudia vytvorili rôzne pomocné prostriedky. Jeden z nich je hmatové (alebo haptické) rozhranie, ktoré nielenže umožňuje manipuláciu s takými objektmi, ale poskytuje aj hmatovú spätnú väzbu. Článok sa zaoberá návrhom a realizáciou hmatového rozhrania vo forme ovládacej páky. Pre konštrukčnú jednoduchosť a finančnú nenáročnosť je zvolená páka s jedným stupňom voľnosti. Takýto typ hmatového rozhrania sa v zahraničnej literatúre označuje ako „Haptic paddle“ vo voľnom preklade „páka s hmatovou odozvou“.

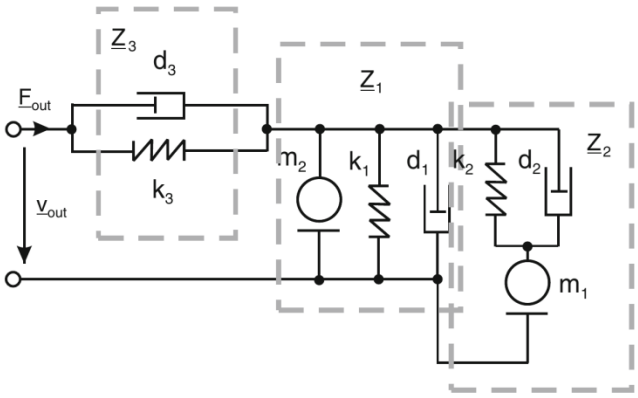
Kľúčové slová: páka s hmatovou odozvou, PLC, virtuálne objekty

Úvod

Úlohy hmatovej interakcie sa delia na dve hlavné triedy: hmatové vnímanie a kontrola motoriky. Tieto triedy môžeme ďalej deliť na základné operácie. Triedu hmatové vnímanie na: detekcia, rozlišovanie, identifikácia a škálovanie. Triedu kontrola motoriky delíme na: presun, selekcia objektu a funkcie a modifikácia. Príklady využitia hmatovej odozvy v rôznych oblastiach môžeme nájsť v [6], [7], [8] a [9]. Klasifikácia hmatového vnímania je fyziologicky postavená a definovaná výhradne na základe umiestenia hmatových receptorov. Je definovaná v norme ISO 9241-910 [1]. Prvý vnem kontaktu, keď ruka interaguje s určitým objektom, vytvárajú dotykové receptory (nervové zakončenia) v koži. Tieto receptory poskytujú informácie napr. o tvare, textúre, viskozite alebo teplote povrchu objektu. Práve týmto informáciám hovoríme taktilné. Keď ruka začne pôsobiť silou a pokúša sa uchopiť objekt, prichádzajú do hry kinestetické informácie (silová spätná väzba), informácie o pozícii a pohybe ruky vzhľadom na objekt [3]. Pri mechanickom návrhu hmatového rozhrania je dôležité uvažovať o používateľových mechanických vlastnostiach ako o mechanickej záťaži systému. Preto ich musíme vhodným spôsobom interpretovať. Jedna z možností je teória výmeny energie medzi človekom a prostredím (virtuálnym alebo reálnym). Fyzikálne dokážeme energiu pri interakcii reprezentovať pomocou sily *F* a kinematických veličín ako zrýchlenie *a*, rýchlosť *v*, alebo ohyb *d*. Vo všeobecnosti ide o závislosť medzi silou *F* a rýchlosťou *v*. Schopnosť človeka interagovať s prostredím je frekvenčne závislá, kvôli dynamickým mechanickým vlastnostiam ľudského tela.

1. Impedancia ľudskej motoriky

Thorsten A. Kern definuje osemprvkový model pre interpoláciu veličín charakterizujúcich impedanciu ľudskej motoriky. Model je možné charakterizovať pomocou troch impedančných skupín: Impedancia $\underline{Z}_3$ (rovnica 3) modeluje elasticitu a tlmenie pokožky pokiaľ je v priamom kontakte s manipulátorom. $\underline{Z}_1$ (rovnica 1) je centrálny element modelu a popisuje mechanické vlastnosti dominantných častí tela - často prstov. $\underline{Z}_2$ (rovnica 2) dáva náhľad na mechanické vlastnosti končatín, často rúk a umožňuje vyvodiť predpoklady o mechanickom predpätí pri určitom spôsobe uchopenia manipulátora.



Obr. 1 Osemprvkový model impedancie používateľa [1]
Fig. 1 Eight-element model of the user's impedance [1]

Parametre každého z prvkov (tlmič, struna a hmota) v modeli na obr. 1 sa určujú meraním. V literatúre [1] je detailne opísaný postup merania a použité meracie prístroje na stanovenie parametrov.

$$\underline{Z}_1 = \frac{s^2 m_2 + k_1 + d_1 s}{s} \left[\frac{\mathrm{Ns}}{\mathrm{m}}\right], \qquad (1)$$

$$\underline{Z}_2 = \left(\frac{s}{d_2 s} + \frac{1}{s m_1} \right)^{-1} \left[\frac{\mathrm{Ns}}{\mathrm{m}} \right], \tag{2}$$

$$\underline{Z}_3 = \frac{d_3 s + k_3}{s} \left[\frac{\mathrm{Ns}}{\mathrm{m}} \right]. \tag{3}$$

Dynamická výmena energie pri interakcii môže byť teda charakterizovaná pomocou mechanickej impedancie. Vo veľkej časti spektra je ľudskeá interakcia pasívna lebo pri veľkej impedancii nemá ľudská motorika žiadny akčný vplyv a ani nevníma spätnú reakciu.

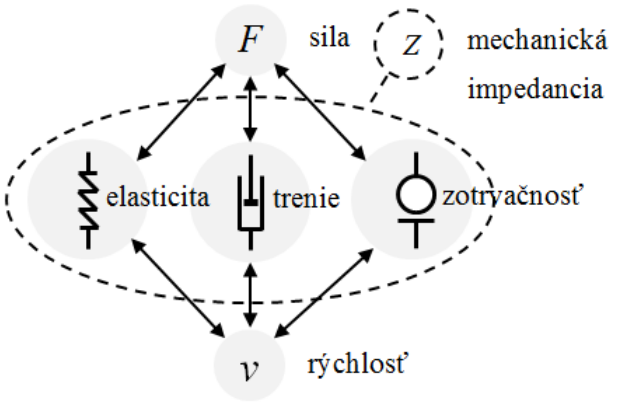
2. Riadenie hmatovej odozvy

Pri modelovaní mechanických systémov sa používajú tri základné prvky:

- struna (elasticita),
- tlmič (trenie),
- hmota (zotrvačnosť).

K realistickému modelu objektu z reálneho sveta sa dokážeme priblížiť ich kombináciou. Samostatne nepokrývajú realitu pretože sú ideálne. Sú pasívne, to znamená, že žiadnym spôsobom neprodukujú energiu. Všetky tri majú translačné a rotačné verzie, ktoré vieme opísať analogicky. Vlastnosti týchto prvkov zvyčajne opisujeme pomocou závislosti sily a pohybu vo forme polohy, rýchlosti a zrýchlenia. Každý z prvkov má jeden z dvoch typov správania sa k energii, ktorá naň pôsobí. Jeden typ správania je uchovanie a druhý rozptýlenie (premena na teplo) všetkej energie. Struna uchováva energiu ako potenciálnu energiu, hmota uchováva energiu ako kinetickú energiu a tlmič energiu rozptyľuje.

Na modelovanie mechanických systémov sa často používa mechanická impedancia. Na obr. 2 je znázornený jej význam.



Obr. 2 Ilustrácia významu mechanickej impedancie

Fig. 2 Illustration of the importance of mechanical impedance

Mechanickú impedancu môžeme považovať za dynamické rozšírenie tuhosti. Ide o mechanický odpor voči pohybu. Presnejšia definícia impedancie je pomocou Laplaceovho obrazu prenosovej funkcie, kde je vstupná veličina rýchlosť a výstupná sila [2]:

$$Z(s) = \frac{F(s)}{v(s)} \left[\frac{Ns}{m} \right] \tag{4}$$

$$Y(s) = \frac{1}{Z(s)} = \frac{v(s)}{F(s)} \left[\frac{m}{Ns} \right] \tag{5}$$

Pre obrazové prenosy jednotlivých prvkov potom platia nasledujúce vzťahy. Mechanická impedancia struny je:

$$Z_s(s) = \frac{f(s)}{v(s)} = \frac{k}{s} \left[\frac{Ns}{m} \right], \tag{6}$$

mechanická impedancia tlmiča:

$$Z_b(s) = \frac{f(s)}{v(s)} = b \left[\frac{Ns}{m} \right], \tag{7}$$

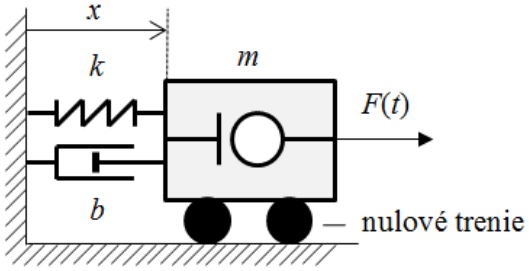
a mechanická impedancia hmoty:

$$Z_m(s) = \frac{f(s)}{v(s)} = ms \left[\frac{Ns}{m} \right]. \tag{8}$$

Medzi mechanickými a elektrickými systémami platí matematická analógia. Túto analógiu môžeme využiť pri modelovaní dynamických systémov a mechanické systémy zakresľovať formou elektrických obvodov.

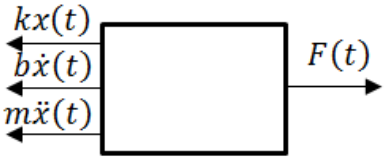
Pre systém na obr. 3 môžeme zakresliť obrazec pôsobenia síl, ktorý je na obr. 4, kde je jasne vidieť vektory všetkých síl, čo na teleso pôsobia. Na základe toho, že súčet síl v systéme je nula, dokážeme systém opísať lineárnou diferenciálnou rovnicou druhého rádu.

$$m\ddot{x}(t) + b\dot{x}(t) + kx(t) - F(t) = 0. \tag{9}$$



Obr. 3 Dynamický mechanický systém: struna, tlmič, teleso

Fig. 3 Dynamic mechanical system: string, damper, body



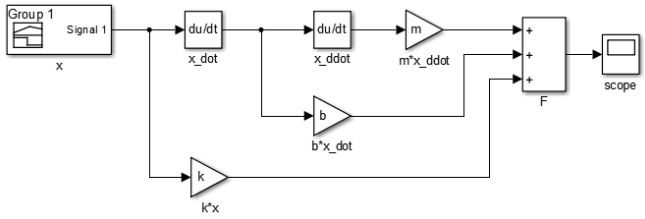
Obr. 4 Obrazec pôsobenia síl systému: struna, tlmič, teleso

Fig. 4 Block diagram of the system forces: string, damper, body

Takúto diferenciálnu rovnicu dokážeme v nástroji Simulink modelovať viacerými spôsobmi; pomocou stavového modelu, prenosovej funkcie, alebo grafu signálových tokov. My si zvolíme graf signálových tokov, čo je vlastne schéma v Simulinku, kde využijeme základné prvky ako zosilnenie, integrátor, derivátor a súčtový člen. Z hľadiska kauzality môžeme usporiadať schému dvoma spôsobmi.

Kauzalita $F(t)=f(x(t))$: Priamo z rovnice bez akýchkoľvek úprav je zrejmá závislosť sily od polohy. Na základe toho dokážeme vytvoriť model ktorý je na obr. 5. Tento model vyhovuje z hľadiska kauzality pri modelovaní impedancie. Nevýhodou je derivácia, ktorá vystupuje na vstupe a robí celý systém náchylný na rušenie. V praxi sa používa dis-

krétna náhrada derivácie, ktorá je konečná, ale aj tak niekoľko násobne zosilňuje šum na vstupe.



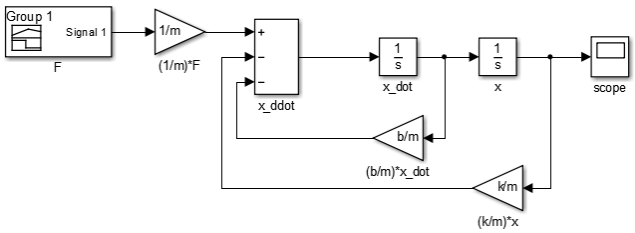
Obr. 5 Schéme dynamického systému vytvorená podľa závislosti sily od polohy

Fig. 5 Simulink model of the dynamic system created according to the force-to-position dependence.

Kauzalita $x(t)=f(F(t))$: Pre modelovanie závislosti polohy od sily si rovnicu môžeme upraviť na tvar, kde vyjadríme zrýchlenie:

$$\ddot{x}(t) = -\frac{b}{m}\dot{x}(t) - \frac{k}{m}x(t) + \frac{1}{m}F(t). \qquad (10)$$

Potom bude schéma vyzerať ako na obr. 6. Kvôli integrátorom a zápornej spätnej väzbe má takýto systém charakter systému prvého rádu. Je odolný voči šumu a je výhodný pri modelovaní mechanickej impedancie.

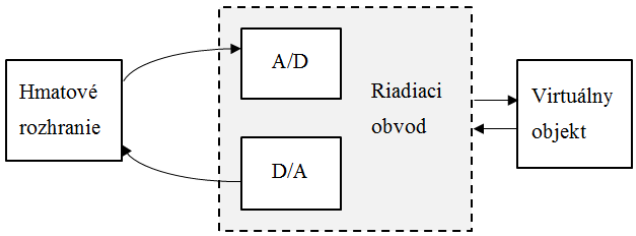


Obr. 6 Schéme dynamického systému vytvorená podľa závislosti polohy od sily

Fig. 6 Dynamic system schematic based on position-to-force dependence

3. Návrh hmatového rozhrania

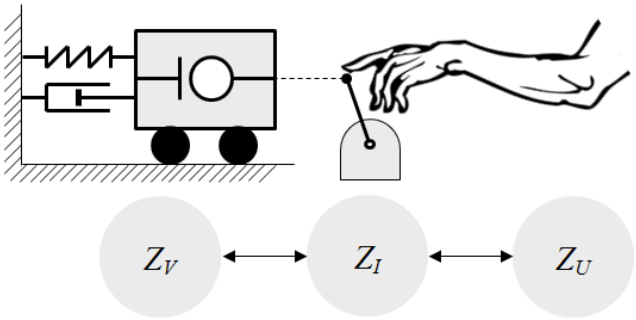
Hmatové rozhranie typu Haptic paddle (HP) je jednoduché hmatové rozhranie s jedným stupňom voľnosti (1 DOF). Využíva impedančné riadenie. Štandardne sa pri väčšine konštrukcií HP využíva elektromagnetický aktuátor a to konkrétne jednosmerný motor, ktorý generuje výstupný krútiaci moment. Užívateľ typicky interaguje so zariadením prostredníctvom joysticku ako rukoväte, kde sa transformuje krútiaci moment motora na výstupnú silu. Motor je riadený výkonovým zosilňovačom, ktorý je väčšinou realizovaný ako H-most. H-most je ovládaný pomocou riadiaceho obvodu, ktorý komunikuje s počítačom na ktorom beží riadiaci softvér. Keďže ide o impedančne riadený systém, tak všetky HP majú snímač polohy, niektoré majú aj snímač sily, aby bolo možné realizovať riadenie s uzavretou slučkou. Na obr. 7 je všeobecná bloková schéma HP.



Obr. 7 Všeobecná bloková schéma HP

Fig. 7 General block diagram of haptic control paddle

Pri návrhu mechanických častí tak ako aj pri výbere aktuátora musíme myslieť na dynamické správanie samotnej mechaniky, teda jej vlastnú impedanciu. Na obr. 8 je znázornené vzájomné pôsobenie troch impedancií, ktoré vzájomne pôsobia pri hmatovej interakcii s HP. Impedancia virtuálneho prostredia Z_V je samozrejme virtuálna, zodpovedá dynamickým vlastnostiam virtuálneho objektu a úlohou navrhovaného systému je čo najvernejšie ju interpretovať používateľovi. Môžeme ju považovať za žiaducu. Jej dôležitou vlastnosťou je pasivita. Jej význam je, že model pozostávajúci len z pasívnych prvkov nemôže sám vygenerovať energiu, len ju uchovať alebo rozptýliť. Zdroj energie pre systém je akčný zásah používateľa. S pasivitou je úzko spojená stabilita. Pokiaľ je pasívny systém spojený s iným pasívnym systémom, je nevyhnutne stabilný. Skúsenosti ukazujú, že ľudská hmatová interakcia je stabilná pri interakcii s pasívnym systémom. Preto je používateľ typicky považovaný za pasívnu impedanciu Z_U, a to najmä pri vysokých frekvenciách nad šírku pásma dobrovoľného pohybu. Výrazný vplyv na stabilitu má vzorkovacia frekvencia a oneskorenie presunu informácie zo vstupu na výstup [5].



Obr. 8 Ilustrácia vzájomného pôsobenia impedancie virtuálneho prostredia Z_V, impedancie hmatového rozhranie Z_I a impedancie používateľa Z_U

Fig. 8 Illustration of interaction of virtual environment impedance Z_V, impedance of the tactile interface Z_I and user impedance Z_U

Preto na výpočtovú časť elektroniky máme hlavnú požiadavku, aby bola schopná pracovať v reálnom čase s frekvenciou hodín minimálne 1000Hz. Čo sa týka výkonových prvkov, mali by byť schopné dodať energiu zhruba 20W, aby bolo možné vyvinúť silu v dynamickom rozsahu 0-10N.

3.1 Návrh hardvéru

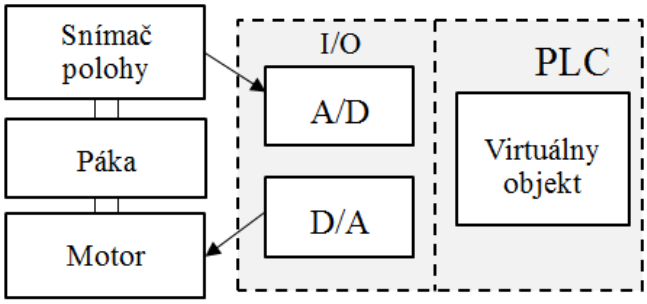
Na riadenie celého systému je použitý programovateľný logický automat (PLC) od firmy B&R. Modulárny systém s PLC obsahuje požadované vstupno-výstupné moduly a spĺňa minimálne hardvérové požiadavky na realizáciu HP. Okrem toho zabezpečujú robustnosť a teda využiteľnosť aj v priemyselnej praxi.

Na obr. 7 je uvedený všeobecný blokový diagram HP, kde je vidieť, že virtuálne prostredie nie je priamo simulované riadiacim obvodom. My sa chceme pokúsiť riadiť HP a súčasne aj simulovať virtuálne objekty na PLC. Na obr. 9 je blokový diagram usporiadania hardvéru.

Konkrétny typ použitého PLC je PowerPanel 400 4PP451.0571-75 od spoločnosti B&R.

Na riadenie motora je použitý modul s typovým označením X20MM2436. Výstupný obvod je tvorený dvojkanálovým H-mostom, ktorý umožňuje dva typy riadenia: Riadenie napätia pomocou šírky impulzov PWM a regulovanie prúdu. Napájať sa dá napätím od 24 do 39 VDC, nominálny prúd je 3A (3,5A maximálny), frekvencia PWM je od 15 Hz do 50 kHz s presnosťou na 16 bitov, rozlíšenie šírky impulzov je 15 bitov plus znamienko, minimálna šírka impulzu je 10ns,

minimálna dĺžka riadiaceho cyklu je 250 µs. Má aj štyri digitálne vstupy, ktoré sa dajú použiť ako dva čítače pre enkodér alebo stavové vstupy s ľubovoľným použitím [13].



Obr. 9 Blokový diagram usporiadania hardvéru HP

Fig. 9 Block diagram of the hardware configuration

Moduly pre digitalizáciu signálov z analógových snímačov, ktoré máme k dispozícii sú X20AI2632. Modul X20AI2632 má dva analógové vstupy, prúdové aj napäťové, 16 bitové rozlíšenie konverzie vrátane znamienka, napätie meria v rozsahu od -10 V do 10 V, minimálna dĺžka riadiaceho cyklu je 200 µs [13].

Použitý je diskový (flat) motor SSW-996 850807, je japonskej výroby a nominálne napätie má 24VDC Je konštruovaný bez oceľového jadra takže ma prakticky nulové zvlnenie momentu. Aby sme boli schopní pre tento motor navrhnúť parametre regulátora potrebujeme meraním zistiť jeho parametre. Dôležité parametre jednosmerného motora, ktoré sú potrebné pri jeho riadení sú:

- Odpor vinutia kotvy R_a,
- indukčnosť vinutia kotvy L_a,
- veľkosť spriahnutého magnetického toku Ψ_{PM},
- momentová konštanta motora k_M.

Meranie sme uskutočnili pomocou volt-ampérovej metódy. Rotor motora je počas merania zabrzdený. Výsledný zmeraný odpor R_a kotvy je 8,687 Ω.

Indukčnosť kotvy sme určili pomocou merania prechodového javu prúdu pri pripojení zdroja na svorky motora. Postupovali sme veľmi podobne ako pri meraní odporu. Rotor bol zabrzdený a zdroj sme pripájali na krátke okamihy pri ktorých sme na osciloskope zaznamenali prechodový jav. Zmerané dáta sme následne spracovali v nástroji Matlab. Indukčnosť motora dokážeme určiť pomocou vzťahu:

$$L_a = T_a . R_a \,[\mathrm{H}]. \tag{11}$$

Odpor vinutia už poznáme z predošlého merania a časovú konštantu odčítame zo zmeraného prechodového javu. Časovú konštantu T_a (63,2% z ustálenej hodnoty) alebo násobok T_a z neho vieme odčítať v niekoľkých bodoch, kde nadobúda určitú percentuálnu hodnotu z ustálenej hodnoty. Výsledná indukčnosť vinutia L_a = 1,898 mH.

Spriahnutý magnetický tok sme vykonali generátorickým meraním na prázdno. To znamená, že sme motor roztočili na určité otáčky externým mechanickým momentom a merali sme ich spolu s indukovaným napätím. Výsledný spriahnutý magnetický tok dokážeme určiť zo základného vzťahu pre výpočet indukovaného napätia:

$$U_{i=} \omega \, \psi_{PM} \,[\mathrm{V}]. \tag{12}$$

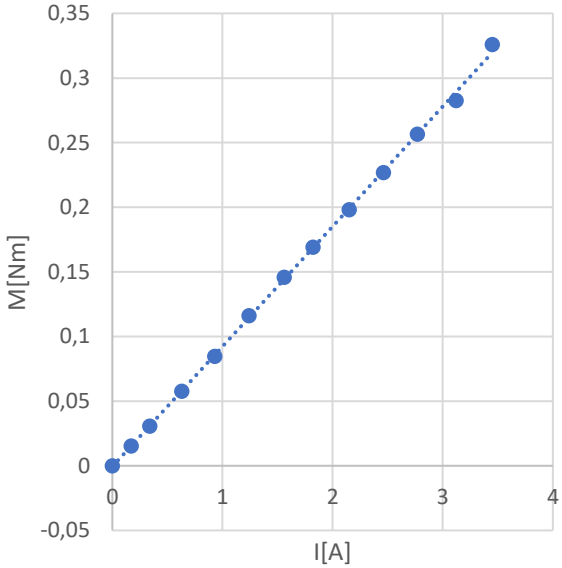
vypočítaná hodnota spriahnutého magnetického toku je 0.09444 [V].

Momentovú konštantu motora k_M určíme podľa vzťahu:

$$k_M = p\psi_{PM} \left[\frac{Nm}{A}\right], \tag{13}$$

kde p je počet pólových dvojíc.

Na riadenie krútiaceho momentu motora potrebujeme poznať jeho momentovú konštantu čo je závislosť krútiaceho momentu od prúdu tečúceho vinutím motora. Momentovú konštantu motora sme určili meraním, pretože sme nemali k dispozícii informáciu o počte pólových dvojíc. Zmerali sme silu vytváranú motorom na páke dĺžky 90 mm od osi rotora v závislosti od pretekajúceho prúdu jeho vinutím. Motor sme riadili prúdom pomocou PLC a modulu pre riadenie motorov X20MM2436. Pri meraní sme postupne nastavovali prúd tečúci motorom a odčítavali hodnoty zo silomera typu IPCouche 0663i. Merali sme aj skutočný prúd ktorý tečie motorom a vytvorili sme závislosť momentu motora od prúdu.



Obr. 10 Závislosť zmeraného momentu motora od prúdu tečúceho vinutím

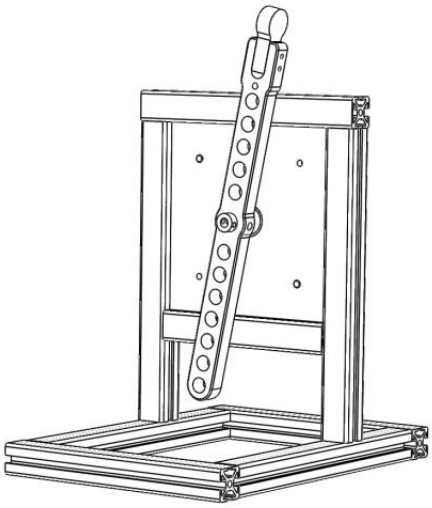
Fig. 10 Dependence of measured motor torque from winding current.

Momentová konštanta ktorú sme zmerali je k_M=0.091934134 [N_m/A]. Na základe známeho indukovaného napätia sme určili aj počet pólových dvojíc motora p=1. Konštanta ktorá definuje závislosť výstupného prúdu od hodnoty ktorá nastavuje prúd vo výstupnom module je k_{DAC}=6353.469.

3.2 Návrh mechanickej konštrukcie

Konštrukcia páky je prispôsobená tomu, že je použitý diskový motor, ktorý má špecifickú plochú konštrukciu a relatívne veľký priemer rotora. Výhodou takejto konštrukcie je väčší moment sily, na druhej strane musíme počítať aj s väčšou zotrvačnosťou. Kvôli tomu páku upneme priamo na hriadeľ rotora. Konštrukciu sme navrhli pomocou hliníkových x-profilov. Motor je umiestnený do takej výšky, aby mohla byť použitá symetrická páka s ťažiskom v strede (obr. 11).

Snímač uhla natočenia páky je uložený priamo na osi motora. Použili sme rotačný rezistívny snímač polohy od firmy ALPS RDC50. Má odpor 10kΩ, garantovanú linearitu ±2% v rozsahu 320° a životnosť 1000000 cyklov.

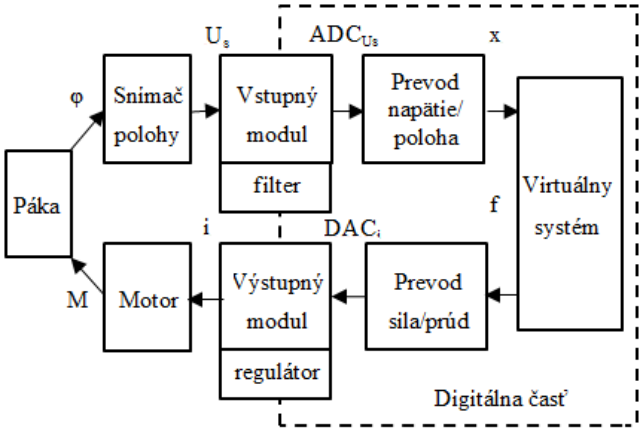


Obr. 11 Návrh konštrukcie páky pre diskový motor

Fig. 11 Design of the control paddle for disc motor

3.3 Návrh softvéru

Na radenie hmatovej odozvy páky s takto navrhnutou konštrukciou sa štandardne využíva impedančný typ riadenia, kde je vstup rýchlosť/poloha (snímanie polohy) a výstup sila (moment motora).



Obr. 12 Blokový diagram riadiacej slučky

Fig. 12 Control loop block diagram

Na blokovom diagrame riadiacej slučky je vidieť tok a transformácie informácií riadiaceho algoritmu. Na obr. 12 je bloková schéma riadiacej slučky, kde sa využíva impedančný riadiaci systém s otvorenou slučkou. Ako je zrejmé s hardvérového návrhu, analógová veličina, ktorá nesie informáciu o polohe, je elektrické napätie *Us* zo snímača polohy. Vstupný modul prevedie elektrické napätie na digitálnu hodnotu ADC_{Us}, vo forme 15 bitového celého čísla, čiže napätie v rozsahu 0 – 10 V na číslo v rozsahu 0 – 32768. Do meraného signálu polohy sa pravdepodobne bude vnášať šum, preto budeme musieť vzorkovaný signál digitálne filtrovať. Môžeme to realizovať priamo, pomocou vstupného modulu X20AI2632. Hodnotu ADC_{Us} prevedieme najskôr na uhol natočenia páky pomocou kalibračného vzťahu:

$$\varphi = \frac{(\varphi_{max} - \varphi_{min})(\mathrm{ADC_{Us}} - \mathrm{ADC_{Us\,min}})}{\mathrm{ADC_{Us\,max}} - \mathrm{ADC_{Us\,min}}} + \varphi_{min}\,[rad], \tag{14}$$

kde φ_{min} a φ_{max} sú hodnoty minimálneho a maximálneho uhlu natočenia v radiánoch. $ADC_{Us\,min}$ a $ADC_{Us\,max}$ sú maximálna a minimálna digitálna hodnota zo vstupného analógového modulu. Pokiaľ bude virtuálny systém rotačného charakteru, tak na jeho vstup privedieme priamo uhol φ. Ak bude systém tranzlačný, privedieme na jeho vstup polohu konca páky. Tú vypočítame jednoducho podľa vzťahu:

$$x = \varphi \cdot l\,[m], \tag{15}$$

kde *l* je dĺžka páky. Informáciu o polohe privedieme na vstup virtuálneho systému. Okrem polohy alebo uhla potrebujeme na vstup priviesť tranzlačnú rýchlosť alebo uhlovú rýchlosť páky. Tú vypočítame pomocou derivácie podľa času:

$$\dot{x} = \frac{dx}{dt}\left[\frac{m}{s}\right], \tag{16}$$

analogicky vypočítame aj uhlovú rýchlosť. Reakciou systému na vstupné veličiny bude informácia o výstupnej sile (v prípade translačného systému) alebo momente (v prípade rotačného systému). Pokiaľ bude výstupom sila, prepočítame ju na krútiaci moment *M* podľa vzťahu:

$$M = f \cdot l\,[Nm]. \tag{17}$$

S krútiaceho momentu sa vypočíta prúd pomocou momentovej konštanty motora k_m. Výpočet prúdu *I* bude podľa vzťahu:

$$I = \frac{M}{k_M}[A]. \tag{18}$$

Prúd je prepočítaný na hodnotu DAC_i, ktorá reprezentuje hodnotu prúdu tečúceho motorom.

Hodnota DAC_i je celé číslo v rozsahu od -32768 do 32768 reprezentujúce prúd v rozsahu -3,5 až 3,5A. Modul riadenia motora X20MM2436 je v prúdovom režime a využívame jeho vnútorný regulátor prúdu, takže nepotrebujeme programovať vlastnú prúdovú slučku.

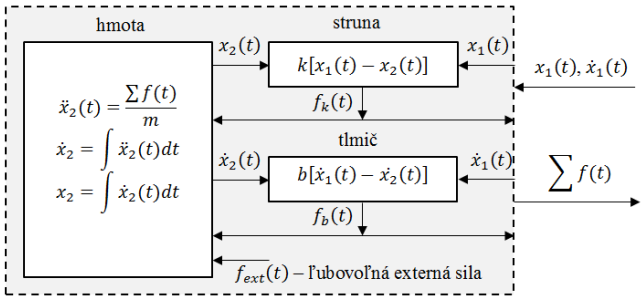
3.4 Matematický model virtuálneho systému

Ako je zrejmé z kapitoly 2, modelovanie mechanických systémov je postavené na diferenciálnych rovniciach druhého rádu. Na softvérové riešenie diferenciálnych rovníc v diskrétnom čase boli implementované diskrétne náhrady spojitej derivácie a integrálu [4].

Virtuálne objekty boli vytvorené na báze objektovo orientovaného programovania v jazyku C++. Vytvorené sú nasledovné virtuálne objekty:

- Dynamický mechanický systém dvoch vozíkov,
- model reálnej struny,
- dynamický model trecej sily presúvaného telesa,
- model virtuálnej steny [10],
- virtuálna lopta

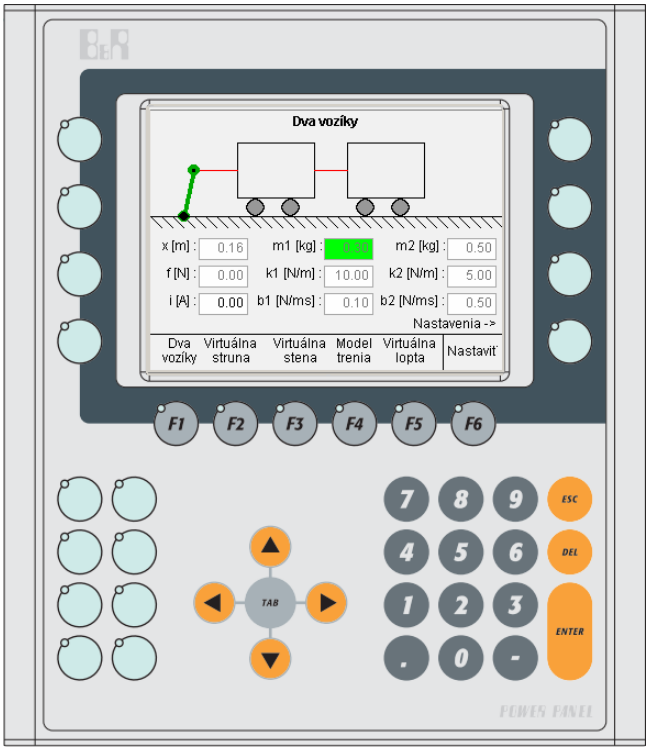
Pre každý zo základných prvkov modelu mechanického systému sme vytvorili triedu, ktorá reprezentuje ich vlastnosti a správanie.



Obr. 13 Logické usporiadanie objektovo orientovaného modelu

Fig. 13 Logical organization of object-oriented model

Na zistenie akou silou pôsobí struna použijeme metódu *getForce*. Trieda reprezentujúca tlmič s názvom *Damper* funguje veľmi podobne s tým rozdielom, že má nastavený koeficient trenia *b* pomocou metódy *setDamping* a rýchlosť začiatku a konca tlmiča pomocou metód *setStartSpeed* a *setEndSpeed*. Trieda, ktorá reprezentuje hmotu sa volá *Body*. Pri inicializácii sa nastaví hmotnosť *m* pomocou konštruktora alebo metódy *setWeight*. Pri aplikácii objektu je použitá naprogramovaná metóda *addForce*, ktorá pripočíta hodnotu sily celkovému súčtu síl *f_ext*, ktoré pôsobia na hmotu. Keď sú všetky sily sčítané, použije sa metóda *applyForces*, pomocou ktorej sa zo síl vypočíta zrýchlenie, rýchlosť a poloha, pokiaľ súčet síl pôsobiacich na teleso presiahol statickú treciu silu.



Obr. 14 Vizualizácia na PLC

Fig. 14 Visualization on PLC

Vizualizačný program slúži na sprostredkovanie vizuálnej spätnej väzby z modelu, konfiguráciu modelu a celkovo programu. Na vykresľovanie modelovaných objektov a charakteristík je použitá knižnica *VisAPI*, ktorá umožňuje vo vizualizácii vykresľovať základné geometrické tvary.

Prepočet digitalizovaných dát polohy uložených v premennej *adc* na fyzikálne veličiny polohu a rýchlosť funguje tak, ako je to opísané v návrhu. Dáta polohy však obsahujú kvantizačný šum. Tento šum sa po derivovaní výrazne zosilní, preto ho musíme filtrovať. Časovú konštantu filtra T_f sme nastavovali na 0,01s. Na výber modelu v programe slúži premenná *model*. Na zvolenie virtuálneho modelu slúžia funkčné tlačidlá vizualizácie F1 –F5, podľa obr. 14.

Na obr. 15 je zobrazená hardvérová realizácia HP.

Obr. 15 Hardvérová realizácia HP

Fig. 15 Hardware realization

Záver

Vývoj hmatových rozhraní je motivovaný použitím v systémoch na vzdialenú manipuláciu a ovládanie, použitím vo virtuálnej realite, v neinvazívnych lekárskych aplikáciách a v komunikačných systémoch. Podstata hmatovej odozvy spočíva v tom, že operátor príslušnej kategórie riadiaceho systému získava od riadeného systému odozvu nielen v podobe vizuálnych informácií na zobrazovacej jednotke doplnenej aktustickými informáciami, ale aj fyzickou reakciou ovládacieho prvku (ovládacej páky, joysticku, ...), ktorý napríklad operátorovi kladie mechanický odpor v závislosti od stavu riadeného systému, kmitá alebo inak fyzicky reaguje. Pri modelovaní mechanických systémov sme použili tri základné prvky: struna (elasticita), tlmič (trenie), hmota (zotrvačnosť). Pomocou týchto prvkov sú v PLC vytvorené virtuálne systémy: dva vozíky spojené pružinou, model trenia kvádra na podložke, model pružiny, model pevnej steny a loptičky. Virtuálnemu systému nastavujeme parametre modelovaných prvkov ako napríklad hmotnosť telies, konštanty strún a tlmičov a aj nelinearít. Vytváranie modelov funguje na báze objektovo orientovaného programovania. Na riadenie a vizualizáciu slúži PLC typu PowerPanel 400 od firmy B&R s funkčnými tlačidlami a displejom.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] HATZFELD, CH., KERN, T. A.: Engineering Haptic Devices - A Beginners Guide. London : Springer-Verlag, 2014. ISBN 978-1-4471-6517-0

[2] SAMUR, E.: Haptics Performance Metrics for Haptic Interfaces. London : Springer-Verlag, 2012. ISBN 978-1-4471-4224-9

[3] SADDIK, A. EL, OROZCO, M., EID M., CHA, J.: Haptics Technologies Bringing Touch to Multimedia, Springer Series on Touch and Haptic Systems. Berlin, Heidelberg : Springer-Verlag , 2011. ISBN 978-3-642-22657-1

[4] BALÁTĚ, J.: Automatické řízení. BEN, Praha, 2004, ISBN 80-7300-148-9

[5] COLGATE, J. E., WEIR, D. W.: Stability of Haptic Displays. [Online] [Dostupné na internete: 10. 5. 2019.] http://web.stanford.edu/class/me327/readings/4-Weir08-Rendering-Stability.pdf.

[6] POORTEN, V.E., DEMEESTER, E., LAMMERTSE, P.: Haptic feedback for medical applications, a survey, in Proc. Actuator Conf., 2012, s. 1–6

[7] LOGITECH COMPANY: Extreme 3D pro joystick. [Online] [Dostupné na internete : 10. 5. 2019] https://www.logitechg.com/en-us/products/gamepads/extreme-3d-pro-joystick.html

[8] STEELE, M., GILLESPIE, R.B.: Shared control between human and machine: using a haptic steering wheel to aid in land vehicle guidance. Proc Human Fact Ergonom Soci Annual Meet 2001, 45 (23): 1671–1675. DOI: 10.1177/154193120104502323

[9] YIN, X., GUO, S., HIRATA, H., ET AL.: Design and experimental evaluation of a teleoperated haptic robot–assisted catheter operating system. J Int Mater Syst Struct 2014; 27(1): 3–16. DOI: 10.1177/1045389X14556167

[10] HRBČEK, J., BOŽEK, P., SVETLÍK, J., ŠIMÁK, V., HRUBOŠ, M., NEMEC, D., JANOTA, A., BUBENÍKOVÁ, E.: Control system for the haptic paddle used in mobile robotics, In: International Journal of Advanced Robotic Systems, Sage journals, Vol. 14, No. 5, 2017, 1-11 s., ISSN 1729-8814

[11] PAPRČKA, M.: Hmatová odozva ovládacej páky, diplomová práca, 2016, ID 28260220162021

[12] HRBČEK, J., ŠIMÁK, V., HRUBOŠ, M.: Riadenie motorov použitím systému B&R, EDIS 2017, ISBN 978-80-554-1327-3

[13] B&R: Help programu Automation Studio, 2019

Abstract

This paper discusses the haptic paddle design and realisation for the reason of virtually touching objects and feeling its forces. A haptic interface is a kinaesthetic link between a human and some real or virtual environment. Control systems based on PLC have desired input–output modules and are fulfilling minimal hardware requirements for the realization of HP. They are also ensuring robustness and applicability in industrial praxis. The models of five virtual objects were implemented into the PLC using object-oriented programming. The parameters of the virtual objects can be changed in the visualization (like stiffness *k*, damping constant *b* and wall position).

Ing. Jozef Hrbček, PhD.

Ing Martin Paprčka

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológii
Katedra riadiacich a informačných systémov
Univerzitná 1
010 26 a Žilina
E-mail: jozef.hrbcek@fel.uniza.sk

RIADENIE POLOHY LOPTIČKY NA KOTÚČI POMOCOU PLC A FREKVENČNÉHO MENIČA

Jozef Hrbček, Vojtech Šimák

Abstrakt
Táto práca pojednáva o doprednom riadení laboratórneho systému loptičky na kotúči, ktorý sa skladá z riadiaceho systému, frekvenčného meniča, indukčného motora, snímača vzdialenosti, kotúča a remenicového prevodu. Na začiatku je vysvetlené riadenie indukčného motora frekvenčným meničom, ďalej je opísaná stratégia riadenia a implementácia riadiaceho algoritmu do PLC. Posledné časti ukazujú výsledky a možné zlepšenia systému.

Kľúčové slová: Prediktívne riadenie, PID, identifikácia, frekvenčný menič, trojfázový asynchrónny motor

Úvod

Vytvorený systém loptičky na vertikálne položenom kotúči sa líši od podobných zariadení svojimi špecifickými (nezvyčajnými) vlastnosťami, ktoré vyžadujú väčšie nároky na samotný riadiaci algoritmus. Systém loptičky a kotúča je v odbornej literatúre označovaný ako BOW (The Ball and Wheel system). Vyznačuje sa svojou nelinearitou, nestabilitou, vplyvom náhodných veličín a ohraničeniami, podobne ako mnohé systémy v priemyselných aplikáciách. Vplyv na dynamické správanie má nielen pomer veľkosti použitého kotúča a loptičky, ale aj prevod medzi hnacou jednotkou a hnaným systémom. Miesto položenia loptičky a požiadavka na umiestnenie loptičky má tiež vplyv na regulačný pochod. Ak bude použitý algoritmus riadenia schopný riadiť systém loptičky a kolesa s požadovanými vlastnosťami, môžeme o ňom predpokladať, že jeho využitie bude uplatniteľné aj v priemysle. V tomto systéme sú aplikované regulátory PSD a MPC.

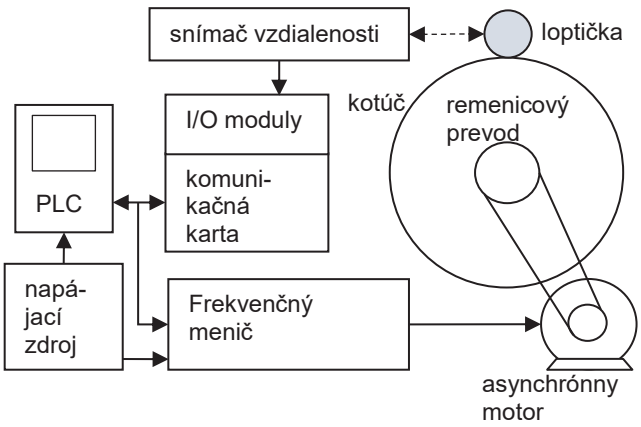
1. Hardvérové zloženie systému

Programovateľný logický automat (PLC) Power Panel od rakúskej firmy Bernecker + Rainer spolu so vstupno - výstupnými a komunikačnými modulmi tvoria základ riadiaceho systému. Riadený systém loptičky na kotúči obsahuje akčný člen, ktorým je trojfázový asynchrónny motor o výkone 550W a optický odmeriavací systém od firmy SICK s odozvou merania 15ms s presnosťou ±7mm. Nevyhnutnou súčasťou je frekvenčný menič, aby sme mohli ovládať trojfázový asynchrónny motor.

Podľa hore uvedeného opisu je zrejmé o aké nezvyčajné špecifické vlastnosti systému ide:

- Použitie lacného asynchrónneho motora, ktorý nie je určený na rýchle zmeny pohybu – prejavuje sa sklz a aj zotrvačnosť. Vhodným motorom by bol napríklad servomotor.
- Odmeriavací systém s veľkou periódou odozvy a malou presnosťou.
- Frekvenčný menič (0.75 kW) s možnosťou zmeny frekvencie po kroku 0,5Hz.

Dokážeme riadiť takýto systém? O to sa postará softvérová časť. Pre riadenie základného pohybu motora využívame štandardizované PLCopen funkcie. PLCopen je medzinárodná organizácia pre štandardizáciu softvérových prostriedkov v automatizácii. Takto je zaručená lepšia implementácia a prenositeľnosť kódu. Štandard obsahuje aj časť riadenie pohybu (PLCopen Motion control) pre programovanie systémov s motormi. Použitý softvérový nástroj Automation Studio umožňuje programovanie riadiacich systémov v deviatich programovacích jazykoch. Obsahuje aj podporu všetkých piatich programovacích jazykov, ktoré definuje norma IEC 61131 v časti 3 pre programovanie riadiacich systémov v priemyselnej automatizácií.



Obr. 1 Bloková schéma systému BOW

Fig. 1 The block diagram of BOW system

1.1 Asynchrónny motor

Asynchrónny motor je točivý elektrický motor pracujúci na striedavý prúd. V súčasnosti je najviac využívaným poho-

nom v elektrotechnike vôbec. Výhodami asynchrónneho motora sú:

- jednoduchá konštrukcia;
- nízke náklady na výrobu;
- vysoká spoľahlivosť a nenáročnosť údržby.

Medzi nevýhody patrí veľký prúdový nárast po pripojení do siete a nemožnosť plynulej regulácie otáčok lacnými a jednoduchými prostriedkami. Tok energie medzi hlavnými časťami motora statorom a rotorom je realizovaný výhradne pomocou elektrickej indukcie, vďaka čomu sa mu niekedy hovorí aj indukčný motor. Najčastejšie používaným asynchrónnym motorom v elektrických strojoch je trojfázový motor, ktorý je používaný ako základná pohonná jednotka. Základom činnosti asynchrónneho motora je vytvorenie točivého magnetického poľa statorom v okolí rotora. Toto magnetické pole vytvorí v rotore napätie a vyvolá točivú silu. Otáčky točivého magnetického poľa n_{mag} sú dané pomerom frekvencie f napájacieho napätia a počtu pólových dvojíc p trojfázového motora. Väčšina asynchrónnych motorov má dve alebo tri pólové dvojice. Výsledný vzťah je potom daný nasledovne:

$$n_{mag} = \frac{f.50}{p}\left[min^{-1}\right]. \quad (1)$$

Pri bežnej pasívnej záťaži sa motor nikdy nemôže otáčať rovnakými otáčkami akými sa otáča magnetické pole statora. Pri synchrónnych otáčkach by sa magnetické pole a rotor voči sebe vôbec nepohybovali a nevznikala by tak žiadna točivá sila. Miera rozdielu otáčok poľa a rotora je nazývaná sklz s, udávaná je v [%] a je definovaná vzťahom:

$$s = \frac{n_{mag} - n}{n_{mag}}.100\left[\%\right], \quad (2)$$

kde n sú otáčky rotora. Podľa hodnoty sklzu sa môže asynchrónny motor rozdeľovať na oblasti jeho práce. A to buď generátor $s \in (-\infty, 0)$, motor $s \in (0, 1)$ alebo brzda $s \in (1, \infty)$.

Otáčky rotora sa môžu počítať pomocou nasledujúceho vzťahu:

$$n = \frac{f.60}{p}\left(1 - s\right)\ \left[rpm\right]. \quad (3)$$

Otáčky motora sú teda dané jeho sklzom, frekvenciou napájacieho napätia a počtom pólových dvojíc statora.

Regulácia zmenou frekvencia sa realizuje pomocou frekvenčného meniča, ktorý riadi otáčky magnetického poľa statora. To je možné dvoma spôsobmi: skalárne alebo vektorovo. Skalárne riadenie sa používa u motorov s malými nárokmi na dynamické vlastnosti. Pri vektorovom riadení sa dá nastaviť nielen veľkosť magnetického poľa, ale ešte aj jeho smer a tým môžeme docieliť plynulú zmenu otáčok.

1.2 Frekvenčný menič

ACOPOSinverter X64 je frekvenčný menič určený pre trojfázové asynchrónne motory s napäťovým rozsahom od 200 do 500V a výkonovým delením od 0,18 do 15kW. So svojimi integrovanými funkciami je obzvlášť vhodný pre splnenie požiadaviek nielen jednoduchých priemyselných aplikácií. Priamo v meniči je obsiahnutých niekoľko funkcií, ako je napríklad prepäťová ochrana pre motor, lokálne ovládanie pomocou tlačidiel, možnosť pripojenia brzdového rezistora, automatické vyrovnanie rotačného zaťaženia s detekciou rýchlosti rotácie a automatické reštartovanie, tak isto je možné uložiť konfiguráciu priamo do meniča.

Základné parametre frekvenčného meniča:

- Typ: ACOPOSinverter x64
- Inštalovaný výkon [kW]: 0.75
- Nominálny prúd [A]: 4,8
- Maximálny prúd [A]: 7,5
- Rozlíšenie [Hz]: 0,5
- Frekvenčný rozsah [Hz]: 0,5 ... 500

1.3 Snímač vzdialenosti

Optické snímače krátkej vzdialenosti pracujú na princípe snímania vyslaného modulovaného infračerveného žiarenia. Poskytujú presné merania vzdialenosti pre aplikácie, ktoré vyžadujú vysoký stupeň presnosti. Snímač má kovové puzdro s polymetylmetakrylátovou šošovkou (PMMA). Má merací rozsah 1000 mm.

Základné parametre:

- Typ: DT20-P130B1000
- Merací rozsah [mm]: 100... 1000
- Doba odozvy [ms]: 10
- Rozhranie: analógový výstup [mA]: 4 … 20
- Presnosť [mm]: ± 4

Tento senzor je možné vymeniť za iný, ktorý má rýchlejšiu odozvu. Na určenie vzdialenosti (polohy) loptičky môžeme tiež použiť digitálne spracovanie obrazu [1], [2].

1.4 Riadiaci systém

Na riadenie systému je použitý priemyselný logický automat (PLC) PowerPanel 400. Tento PLC obsahuje displej a integruje technológiu riadenia. Obsahuje komunikačnú kartu pre rozhrania priemyselných sietí X2X a Ethernet POWERLINK. Na ukladanie sa používa Compact Flash karta s kapacitou 8 GB, čo umožňuje, aby bol riadiaci systém úplne bez rotujúcich častí a dáva mu schopnosť pracovať v priemyselných aplikáciách. Programový softvér Automation Studio slúži na tvorbu jednotnej konfigurácie a programu, vrátane diagnostiky a vizualizácie.

Základné parametre PLC:

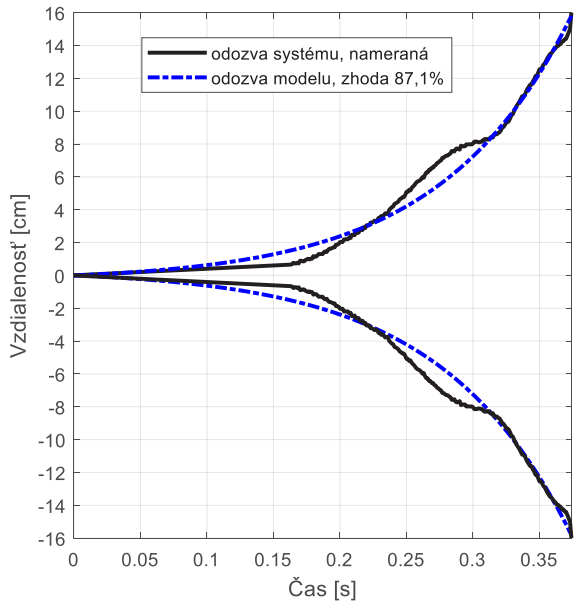
- Typ: 4PP451.0571-75
- Zobrazovacia jednotka: 5.7" QVGA TFT displej
- Procesor a pamäte: Geode LX800 500MHz; 128 MB SDRAM; 512 KB SRAM
- Rozhrania: Ethernet, USB, RS232, X2X, POWERLINK

2. Parametrická identifikácia systému

Na vytvorenie matematického modelu systému, ktorý sa používa v riadiacom algoritme založenom na modeli, boli použité vstupné a výstupné dáta zostrojeného systému. V tejto práci bola na získanie modelu systému použitá parametrická diskrétna identifikácia. Hlavným dôvodom identifikácie je dlhší čas odozvy snímača oproti času cyklu riadiaceho programu. Pri použití modelu máme informácie o dynamickom správaní systému v každej perióde vykonania programu. Nasledujúca časť opisuje postup identifikácie.

Prechodovú charakteristiku sme získali privedením jednotkového skoku na vstup systému. Vstupom systému je otočenie kotúča, ktoré sa realizuje vstupným signálom s hodnotou 15000 unitov v kladnom a -1500 unitov v zápornom smere pre frekvenčný menič. Hodnota 500 unitov zodpovedá frekvencii 1 Hz na výstupe frekvenčného meniča. Namerali sme vzdialenosť padajúcej loptičky smerom od vertikálnej polohy kotúča. Presnosť bola vyhodnotená v MATLABE (príkaz *compare*) pomocou hodnoty zhody, ktorá opisuje zhodu medzi modelovanými a meranými výstupmi [3]. Získaná hodnota zhody je 87,1%. Táto hodnota je postačujúca vzhľadom na vytvorenie modelu z čo najjed-

noduchšou štruktúrou. Okrem dôkazu vhodnosti pomocou hodnoty zhody bola vykonaná validácia reziduálnym testom.



Obr. 2 Odozva systému a modelu

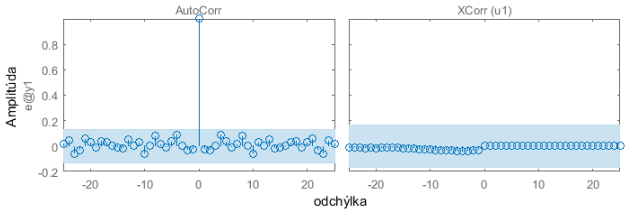
Fig. 2 System response and model response

Obrazový prenos identifikovaného systému je:

$$F(z)=\frac{\pm 0,00004593z^2}{z^4-0,88z^3-0,26z^2-0,18z+0,32}. \quad (4)$$

Vzorkovacia perióda T =0,001 s.

Reziduálna analýza, zobrazená na obr. 3, sa skladá z dvoch častí: test belosti a test nezávislosti. Podľa testu belosti má dobrý model reziduálnu autokorelačnú funkciu vnútri intervalu spoľahlivosti príslušných odhadov, čo naznačuje, že reziduá sú korelované. Podľa testu nezávislosti má dobrý model zvyšky korelované s minulými vstupmi. Dôkaz korelácie ukazuje, že model neopisuje časť výstupu, ktorý sa týka daného vstupu. Napríklad, vrchol mimo interval spoľahlivosti pre oneskorenie k znamená, že výstup $y(t)$, ktorý pochádza od vstupu $u(t\text{-}k)$, nie je riadne opísaný pomocou modelu.



Obr. 3 Reziduálna analýza

Fig. 3 Residual test

Reziduálne odchýlky sú náhodne rozmiestnené okolo nuly, nie sú korelované s akoukoľvek inou premennou, vrátane vstupov a výstupov, a preto je tento model správny.

3. Algoritmy riadenia

Táto kapitola opisuje dva riadiace algoritmy; PSD a MPC, ktoré boli implementované do PLC.

3.1 PSD

PSD je diskrétna forma PID regulátora. Na to, aby sa použil v takýchto zariadeniach, musí byť algoritmus riadenia PID diskretizovaný. Použitím nahradenia integračnej metódy sumačnou metódou a derivácie diferenciou bol získaný PSD algoritmus.

PSD regulátor má definovaný akčný zásah $u_R(t)$ len v časových okamihoch $t = kT$, kde k = 0,1,2,…, tj:

$$u_R(kT)=r_0\left[e(kT)+\frac{1}{T_I}I(kT)+T_D D(kT)\right], \quad (5)$$

kde r_0 je zosilnenie, $I(kT)$ je hodnota integrálu v diskrétnom časovom okamihu kT a $D(kT)$ je hodnota derivačnej zložky v diskrétnom časovom okamihu kT. V tejto práci bol použitý inkrementálny algoritmus s filtrom. Aplikovaním derivácie na $u_R(t)$.

$$u_R(t)=K\left[e(t)+\frac{1}{T_I}\int_0^t e(t)dt+T_D\frac{de(t)}{dt}\right], \quad (6)$$

dostaneme:

$$\frac{du(t)}{dt}=K\left[\frac{de(t)}{dt}+\frac{1}{T_I}e(t)+T_D\frac{d^2e(t)}{dt^2}\right]. \quad (7)$$

Náhrady derivácií diferenciou:

$$\frac{de(t)}{dt}=\frac{e(k)-e(k-1)}{T}, \quad (8)$$

$$\frac{de^2(t)}{dt^2}=\frac{e(k)-2e(k-1)+e(k-2)}{T^2}, \quad (9)$$

kde T je perióda vzorkovania. Dosadením týchto vzťahov do rovnice 7 dostaneme diferenčnú rovnicu:

$$u(k)=u(k-1)+q_0e(k)+q_1e(k-1)+q_2e(k-2), \quad (10)$$

kde:

$$q_0=K\left(1+\frac{T}{T_I}+\frac{T_D}{T}\right), q1=-K\left(1+\frac{2T_D}{T}\right), q_2=K\frac{T_D}{T}. \quad (11)$$

Rovnica 10 bola implementovaná do PLC s cyklickým časom vykonávania 1 ms.

3.2 MPC

Metódy prediktívneho riadenia s modelom (MPC – Model Predictive Control) vychádzajú zo spoločnej základnej myšlienky a majú spoločné črty, ktorými sú: matematický model riadenia systému je použitý na predikciu budúceho riadeného výstupu systému, je známa trajektória žiadanej veličiny do budúcnosti, výpočet postupnosti budúcich riadiacich zásahov zahŕňa minimalizáciu vhodnej účelovej funkcie (obvykle kvadratickej) s budúcimi trajektóriami prírastkov riadenia a regulačnej odchýlky, iba prvý akčný zásah je realizovaný a celý postup minimalizácie sa opakuje v ďalšej perióde vzorkovania. Použiteľnosť prediktívnych riadiacich algoritmov je pomerne široká a kvalita riadenia je zvyčajne vyššia ako v prípade PID regulátorov. Sú použiteľné na nestabilné, viac-rozmerné procesy alebo procesy s dopravným oneskorením a kompenzujú účinky merateľných a nemerateľných porúch [4], [5]. Aplikácie použitia MPC v tunelových systémoch sú opísané v [6], [7] a [8].

V tejto práci sme použili metódu DMC (Dynamic Matrix Control), ktorá je jednou z najrozšírenejších prístupov a vytvára základ mnohých komerčne dostupných produktov MPC. Riadiaci algoritmus je založený na modeli získanom z reálneho systému:

$$y(k) = \sum_{i=1}^{N} h_i u(k-i), \tag{12}$$

kde h_i sú koeficienty FIR (konečnej impulznej odozvy) modelu riadeného systému. Predikované hodnoty môžeme vyjadriť:

$$\begin{aligned} \hat{y}(n+k \mid n) &= \sum h_i \Delta u(n+k-i) + \hat{d}(n+k \mid n) = \\ &= \sum h_i \Delta u(n+k-i) + \sum h_i \Delta u(n+k-i) + \hat{d}(n+k \mid n), \end{aligned} \tag{13}$$

predpokladáme, že aditívna porucha je počas predikčného horizontu p konštantná, kde k=0 ...p.

$$\hat{d}(n+k \mid n) = \hat{d}(n \mid n) = y(n) - \hat{y}(n \mid n). \tag{14}$$

Dekompozícia odozvy na zložku závislú na budúcich hodnotách riadenia a na zložku určenú stavom systému v čase n,

$$\hat{y}(n+k \mid n) = \sum h_i \Delta u(n+k-i) + f(n+k), \tag{15}$$

kde $f(n+k)$ je tá zložka odozvy systému, ktorá nezávisí na budúcich hodnotách akčnej veličiny. A výraz pre predikciu potom môžeme písať v maticovom tvare:

$$\hat{y} = Gu + f, \tag{16}$$

kde $\hat{y}$ je p rozmerný vektor obsahujúci predikcie výstupnej veličiny systému s horizontom predikcie, u je vektor príspevkov akčnej veličiny, f je voľná odozva systému a G je matica dynamiky systému. Úlohou DMC regulátora je dopredne riadiť výstupy podľa nastavenej hodnoty v zmysle najmenších štvorcov s možnosťou zahrnutia penalizácie vstupov a výstupov. Manipulované premenné sú zvolené na základe minimalizovania kvadratickej účelovej funkcie, ktorá zhodnocuje budúce regulačné odchýlky [9]:

$$\begin{aligned} J = &\sum_{j=1}^{p} \left[\lambda_y (\hat{y}(n+j \mid n) - w(n+j)) \right]^2 \\ &+ \sum_{j=1}^{m} \left[\lambda_u \Delta\hat{u}(k+j-1) \right]^2 = \lambda_y \hat{e}\hat{e}^T + \lambda_u \hat{u}\hat{u}^T \\ &= \lambda_y (G\hat{u} + f - w)(G\hat{u} + f - w)^T + \lambda_u \hat{u}\hat{u}^T, \end{aligned} \tag{17}$$

kde $\hat{e}$ je vektor budúcich chýb v rámci predikčného horizontu a $\hat{u}$ je vektor prírastkov akčnej veličiny počas horizontu riadenia. Ak nie sú prítomné ohraničenia, môžeme optimalizačnú úlohu riešiť analyticky. Dostaneme celú trajektóriu budúcich prírastkov akčnej veličiny

$$\Delta u = (G^T \Lambda_y{}^T \Lambda_y G + \Lambda_u{}^T \Lambda_u)^{-1} G^T \Lambda_y{}^T \Lambda_y (w - f), \tag{18}$$

kde Λ_y je diagonálna váhovacia matica a Λ_u je výstupná diagonálna matica pre nastavenie rýchlosti zmien akčnej veličiny. Z vypočítaných prírastkov $\Delta u(k)$ je použitý len prvý člen. Je to preto, lebo nie je možné dokonale odhadnúť poruchový vektor a predpovedať dlhodobé správanie systému [4]. Celý postup sa opakuje v ďalšej perióde vzorkovania. Ide o princíp pohyblivého horizontu. V reálnych aplikáciách sú prítomné aj ohraničenia, ktoré je možné zahrnúť priamo do riadiaceho algoritmu.

$$\begin{aligned} &u_{\min} \le u(n) \le u_{\max}, \Delta u_{\min} \le u(n) - u(n-1) \le \Delta u_{\max}, \\ &y_{\min} \le y(n) \le y_{\max}. \end{aligned} \tag{19}$$

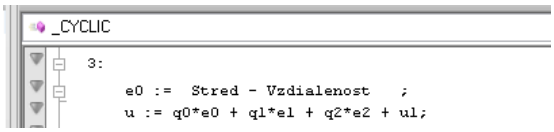
Ohraničenia sú spôsobené rozsahom snímačov a akčných členov, ohraničenou rýchlosťou zmeny polohy akčných členov (daná vlastnosťami členov alebo požiadavky riadeného procesu), požiadavkami na reguláciu bez prekmitov, monotónnym priebehom regulačného priebehu a pod.

Ďalšia kapitola sa zaoberá implementáciou PSD a DMC algoritmu do PLC.

4. Implementácia riadiacich algoritmov

Programový nástroj Automation Studio podporuje všetky základné programovacie jazyky používané v priemyselnej automatizácii podľa normy IEC 61131.

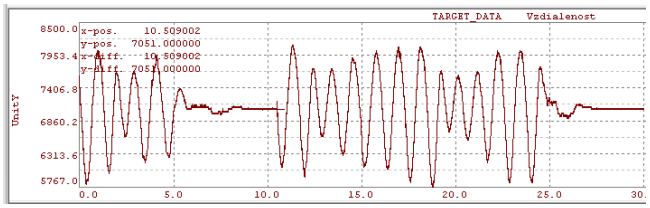
Na implementáciu riadiacich algoritmov PSD bol použitý programovací jazyk ST. Podľa rovnice 10 bol regulátor PSD implementovaný tak, ako je znázornené na obr. 4.



Obr. 4 Implementácia PSD algoritmu

Fig. 4 PSD implementation

Obr. 5 znázorňuje odozvu systému s dvoma vonkajšími zmenami vertikálnej polohy loptičky. V tomto systéme pôsobia aj nemerané rušenia spôsobené nepresne zaobleným povrchom a chybou merania. Referenčná poloha bola nastavená na 7051 jednotiek.

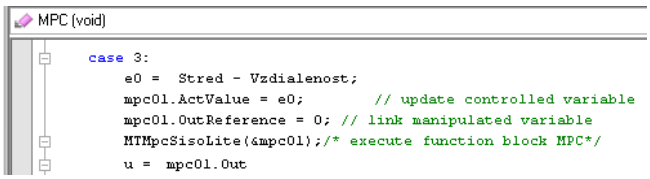


Obr. 5 Správanie systému s PSD regulátorom

Fig. 5 PSD controller behaviour– ball position

Knižnica MTMpcSiso sa používa na implementáciu riadiacich algoritmov založených na modeli, ktoré sa používajú na reguláciu jednorozmerného systému a je napísaná v jazyku ANSI C (obr. 6).

Predikčný horizont p bol nastavený na 60 a riadiaci horizont na 30. Regulátor je hlavne konfigurovaný (ladený) pomocou diagonálnej matice váhových faktorov Λ_y (s nastavenou hodnotou 0,85) a diagonálnej matice pre nastavenie rýchlosti zmien akčnej veličiny Λ_u (s hodnotou 0,1). Váhové faktory sa používajú na hodnotenie regulačnej odchýlky. Čím je zvolená väčšia hodnota parametra Λ_y, tým agresívnejšie sa regulátor chová a robustnosť klesá. Pre nastavenie rýchlosti zmien akčnej veličiny sa používa váhovací faktor Λ_u. Čím väčší je tento parameter, tým sa zvyšuje presnosť a robustnosť sa znižuje. Softvérové bloky môžu obmedziť absolútne hodnoty riadiacej veličiny (*OutMin*, *OutMax*) a rýchlosť jej zmeny (*OutDeltaMin*, *OutDeltaMax*) vzhľadom na manipulované premenné. Tieto sa v algoritme považujú za ohraničenia, ktoré musí regulátor striktne dodržiavať [9].



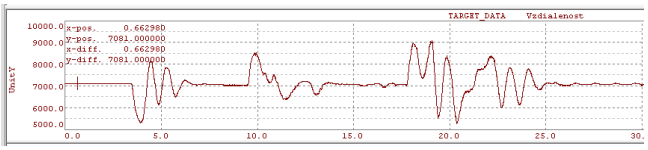
Obr. 6 Implementácia DMC – časť zdrojového kódu

Fig. 6 DMC controller –part of source code

Ohraničenia boli nastavené podľa vlastností reálneho systému na: *OutDeltaMin* = 500 (Rozlíšenie je 0,5 Hz);

OutDeltaMax = 10000 (Maximálna zmena riadiacej veličiny);

OutMin = - 22000;*OutMax* = 22000 (Max. frekvencia 22Hz).



Obr. 7 Správanie systému s DMC regulátorom

Fig. 7 System behaviour with DMC controller

Obr. 7 znázorňuje odozvu systému s tromi vonkajšími zmenami vertikálnej polohy loptičky s rovnakými podmienkami ako pre PSD regulátor. Referenčná poloha *w*(*t*) bola tiež nastavená na 7051 jednotiek. MPC regulátor má rýchlejšiu odozvu a lepšiu kvalitu riadenia. Okrem štandardných riadiacich algoritmov systém obsahuje aj bezpečnostný riadiaci systém. To sa realizuje pomocou bezpečnostného PLC SafeLogic8100 s bezpečnostnými I/O modulmi. Viac informácií o bezpečnostných požiadavkách na riadenie bezpečnostne kritických procesov je uvedených v [10] a [11].

Záver

V tomto príspevku je opísaný systém loptičky na vertikálne položenom kotúči (BOW) a jeho stratégia riadenia. Na otestovanie riadiaceho systému boli úspešne implementované regulátory PSD a DMC. Vytvorený systém sa líši od podobných zariadení svojimi špecifickými nezvyčajnými vlastnosťami ako napríklad: použitie trojfázového asynchrónneho motora a použitie snímača vzdialenosti s dlhou dobou odozvy, ktoré vyžadujú väčšie nároky na samotný riadiaci algoritmus. Aj keď sú komponenty systému odlišné od bežne používaných BOW systémov [12], sme schopní konštruovaný systém riadiť. Systém je možné vylepšiť nahradením snímača vzdialenosti za nový s kratšou dobou odozvy - čas odozvy by sa mal rovnať času programového cyklu. Okrem toho by riadiaci algoritmus mal obsahovať viacero modelov systému pre rôzne rýchlosti otáčania kotúča. V našej práci využívame parametrickú identifikáciu reálneho systému. To bolo nevyhnutné na získanie prechodovej charakteristiky s periódou 1 ms namiesto 10 ms, ktoré poskytuje snímač. Výsledky ukazujú, že DMC má lepšie vlastnosti ako PSD regulátor.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] BUBENÍKOVÁ, E., PIRNÍK, R., HOLEČKO, P., FRANEKOVÁ, M.: The ways of streamlining digital image processing algorithms used for detection of lines in transport scenes video recording, 3th IFAC and IEEE Conference on Programmable Devices and Embedded Systems, Volume 48, Issue 4, 2015, 174–179 pp., ISSN 2405-8963

[2] HALGAŠ, J., PIRNÍK, R.: Monitoring of parking lot traffic using a video detection. In: Acta Technica Corviniensis - Bulletin of engineering. Tom 8, fasc. 3, 2015, 17-20 pp, ISSN 2067-3809

[3] KIVILUOTO, S., WU, Y., ZENGER, K., GAO, X. Z.: Identification of actuator model in an electrical machine by prediction error method and cultural particle swarm optimization, International Conference on System Engineering and Technology (ICSET), 74-78 pp., ISBN 978-1-4577-1256-2

[4] CAMACHO, E. F., BORDONS, C.: Model Predictive Control. 2nd ed., Springer-Verlag, ISBN 1-85233-694-3

[5] MACIEJOVSKI, J. M.: Predictive Control with constrains. 1st publ., Harlow, England: Prentice Hall, p. 331, ISBN 0-201-39823-0

[6] HRBČEK, J., SPALEK, J., ŠIMÁK, V.: Mathematical description of tunnel systems for the purpose of design the predictive algorithm, Acta Electrotechnica et Informatica, Vol. 10, No. 2, pp. 52–56, ISSN 1335-8243

[7] TAN, Z., XIA, Y., YANG, Q., ZHOU, G.: Adaptive Fine Pollutant Discharge Control for Motor Vehicles Tunnels under Traffic State Transition. In: IET Intelligent Transport Systems, 2015,Vol.: 9, Issue: 8, 783-791 pp., ISSN 1751-956X

[8] MIKLÓŠIK, I., SPALEK, J.: Acquisition of Meteorological Data for the Tunnel Simulator, In: Proc. of the 10th International Conference ELEKTRO 2014, 459-464 pp., ISBN 978-1-4799-3720-2

[9] Automation Studio B&R Help Explorer, 2016: MTMpcSiso, Bernecker + Rainer Industrie Elektronik GmbH

[10] ŽDÁNSKY, J., NAGY, P.: Influence of the control system structure with safety PLC on its reliability and safety. Proceedings of the 9th international conference ELEKTRO 2012, Rajecké Teplice, ISBN 978-1-4673-1178-6

[11] ŽDÁNSKY, J.: Using PLC for control of safetycritical processes. In: Proceedings of International conference OWD 2004. Gliwice: Silesian University of Technology, 2004, 421–426 pp., ISBN 83-915991-8-3.

[12] HO, M. T., TU, Y. W., LIN, H. S.: Controlling a ball and wheel system using full-state-feedback linearization [Focus on Education], Control Systems, IEEE, Vol.: 29, Issue: 5, 93 – 101 pp., ISSN 1066-033X

Abstract

This paper discusses control of the laboratory ball on the wheel system (BOW) that consists of PLC, frequency inverter, induction motor and distance sensor. At the beginning the frequency inverter control of induction motor is explained, next the control strategy and implementation is descripted. The last parts show the results and possible improvement.

Ing. Jozef Hrbček, PhD.

Ing. Vojtech Šimák, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológii
Katedra riadiacich a informačných systémov
Univerzitná 1
010 26 a Žilina
E-mail: jozef.hrbcek@fel.uniza.sk

Obsah

KYBERBEZPEČNOSŤ AUTONÓMNYCH VOZIDIEL A UMELÁ INTELIGENCIA

Aleš Janota, Roman Michalík

Abstrakt
Článok je venovaný problematike vzťahu umelej inteligencie a kybernetickej bezpečnosti autonómnych a prepojených vozidiel. Obsahuje prehľad súčasného stavu ich rozvoja, existujúce problémy a analyzuje podmnožinu umelej inteligencie - metódy strojového učenia, ktoré sa javia ako použiteľné a prínosné pre riešenia kybernetickej bezpečnosti automobilových aplikácií.

Kľúčové slová: umelá inteligencia, strojové učenie, inteligentný systém, autonómne vozidlo, kybernetická bezpečnosť

Úvod

Umelá inteligencia (UI) od svojho zrodu na konferencii v Dartmouthe v roku 1956 urazila dlhú cestu, počas ktorej sa priebežne menila jej definícia, obsah a očakávania jej tvorcov. V súčasnosti prevláda definícia, že UI je „teória a vývoj počítačových systémov schopných vykonávať úlohy, ktoré bežne vyžadujú ľudskú inteligenciu, ako je vizuálne vnímanie, rozpoznávanie reči, rozhodovanie a preklad medzi jazykmi“ (The New Oxford American Dictionary, 3. vyd.). Bežná populácia si pojem UI spája viac s rôznymi sci-fi filmami ako s aplikáciami UI, ktoré ale často a nevedomky využíva vo svojich smartfónoch, inteligentných zariadeniach, na webe a pod. [1]. Bez ohľadu na argumenty niektorých filozofov, že skutočná inteligencia nemôže byť nikdy dosiahnutá obyčajným strojom [2], jednou z hlavných vedeckých tém pre budúcnosť ľudstva sa javí dosiahnutie nadľudskej inteligencie (tzv. super-inteligencie). Súčasný výskum sleduje viaceré cesty, čo zvyšuje naše šance, že jedna z nich bude úspešná [3]:

- emulácia celého mozgu (tzv. uploading): vytvorenie dôveryhodnej kópie ľudského mozgu, ktorá vyžaduje použitie niekoľkých, ale zatiaľ technologicky nezvládnuteľných, krokov, ako je skenovanie, transformácia naskenovaných údajov na neuro-výpočtovú štruktúru a nakoniec simulácia pomocou super-počítača;
- biologické poznanie: posilňovanie fungovania biologických mozgov s cieľom zvýšiť našu inteligenciu nad súčasnú úroveň; z tradičných metód s okrajovými účinkami (nové formy vzdelávania a odbornej prípravy), neurologických vylepšení (ovplyvňovanie materskej a detskej výživy, prevencie chorôb atď.) cez aplikácie špeciálnych chemikálií (liekov zlepšujúcich pamäť, výkon, duševnú energiu) až po genetické manipulácie, DNA syntézu alebo klonovanie ľudí;
- rozhrania mozog – počítač: menej pravdepodobný prístup na dosiahnutie superinteligence, avšak potenciálne zlepšujúci výkon nášho mozgu (rýchlejšie výpočty, vyššia presnosť, viac prenášaných údajov, širšie senzorické schopnosti) a eliminujúci (alebo aspoň redukujúci) výskyt niektorých chorôb (v súčasnosti implantáty na zníženie frekvencie epileptických záchvatov, paralýzy Parkinsonovou chorobou, na diagnostiku alebo prevenciu viacerých zdravotných komplikácií);
- siete a organizácie: prepojenie a posilnenie individuálnej inteligencie (sietí a organizácií) tak, aby sa dosiahla určitá forma „kolektívnej super-inteligencie“ (Internet ako budúci super-inteligentný umelý organizmus?);
- umelá inteligencia: jej rozvoj v súčasnosti výrazne poháňa najmä priemysel, pričom čelíme 2 technologickým mega trendom - globálnemu prepojeniu sveta a dynamickému vývoju technológií UI.

Všetky vyššie uvedené prístupy pravdepodobne prinesú rôzne výsledky a tiež veľa kontroverzií, etických problémov, porušení aktuálne platných zákonov, závažných zdravotných komplikácií alebo iných, zatiaľ ani netušených problémov. Jedna z koncepcií hovorí o vývoji UI v troch vlnách – vlne expertných systémov, vlne štatistického alebo strojového učenia odvodzujúceho z poskytnutých dát pravidlá alebo rozhodovacie postupy a budúce vlne spájajúcej predošlé dve s cieľom dosiahnuť kontextovú sofistikovanosť, abstrakciu a vysvetlenie. Samostatne jazdiace automatizované vozidlá sú odrazom druhej vlny. Vo vedeckej komunite zatiaľ neexistuje konsenzus o budúcom vývoji UI, o jej budúcich formách, časových harmonogramoch a potenciálnych dopadoch na ľudskú spoločnosť. Najčastejšie sa skloňujú 3 formy UI [1]:

- UI úzka alebo slabá (*ANI - Artificial Narrow Intelligence*): úzko špecializovaná na jedinú oblasť;
- UI všeobecná, silná alebo na ľudskej úrovni (*AGI - Artificial General Intelligence, Strong AI, Human-Level AI*): s počítačmi inteligentnými tak ako ľudia;
- Umelá super-inteligencia (*ASI - Artificial Superintelligence*): na základe definície Nicka Bostroma je oveľa múdrejšia ako najlepší ľudský mozog prakticky v každej oblasti, vrátane vedeckej tvorivosti, všeobecnej múdrosti a sociálnych zručností.

Zatiaľ používame technológie na zdokonalenie našich fyzických a psychických daností. Stroje môžu robiť veci rýchlejšie, nie je ale nič, čo by mohol robiť stroj a človek nie – stroje nemajú tzv. zdravý rozum a stále sa pohybujeme na úrovni slabej UI. Veľmi široký úvod k základným konceptom UI (zrozumiteľný aj pre čitateľov s malým až žiadnym technickým zázemím) a ich praktickému využitiu od začiatku roka 2018 je k dispozícii v [4]. O dosiahnutí AGI sa hovorí ako o Živote 3.0, nikto však nevie definovať, čo to bude pre ľudstvo znamenať; k dispozícii sú optimistické vízie (pozri napr. „Omega príbeh“ v [5]) až absolútne katastrofické. Vynárajú sa mnohé etické aspekty, diskutované v kontexte bezpečnostných otázok jednotlivých technológií založených na UI, a s nimi rad nových výrazov ako napr. umelá morálka, výpočtová etika, počítačová etika, etika kyborgov, stro-

jová etika, morálka strojov, etika robotov, práva robotov, atď. Podrobný zoznam odkazov a zdrojov uvedených termínov je k dispozícii napr. na str. 135, [6].

Podľa [5] existujú tri odlišné myšlienkové prúdy (školy) reprezentované nasledujúcimi hľadiskami:

- techno-skeptici: „Nemali by sme sa báť, pretože v dohľadnej budúcnosti UI nedosiahne úrovne ľudí (najmenej stovky rokov)“;
- digitálni utopisti: „Nemali by sme sa báť – dôjde k tomu, ale je prakticky zaručené, že to bude dobrá vec (digitálny život je prirodzený a žiaduci ďalší krok v kozmickom vývoji)“;
- členovia hnutí hovoriacich o úžitku UI: „Záujem je opodstatnený a užitočný, pretože výskum a diskusia o bezpečnosti UI teraz zvyšuje šance na dobrý výsledok (cieľom by malo byť vytvorenie užitočnej inteligencie)“.

V akademických kruhoch sa môžeme stretnúť s 2 názorovými prúdmi - jedna skupina sa domnieva, že UI predstavuje existenčnú hrozbu pre ľudstvo, ale jej prínos je vyšší ako náklady; ďalšia skupina oceňuje výhody technológií UI a sústreďuje sa na spravodlivosť, zodpovednosť a transparentnosť jej vývoja. Nájdenie bezpečného správania pre UI je oveľa zložitejším problémom, ako sa pôvodne zdalo - aby bola UI bezpečná, bude pravdepodobne potrebné poskytnúť ju ako veľmi presnú a úplnú definíciu správneho správania, čo sa ale dá urobiť veľmi ťažko [2]. Pri hodnotení negatívnych dopadov by sme mali rozlišovať medzi dvomi situáciami:

- UI je naprogramovaná tak, aby robila niečo devastujúce (je zneužitá)
- UI je naprogramovaná tak, aby robila niečo prospešné, ale vyvíja a používa deštruktívnu metódu na dosiahnutie svojho cieľa (je príliš nedokonalá).

Dalo by sa namietať, že náš budúci vzťah a bezpečnú koexistenciu s inteligentnými strojmi kodifikujú Asimovove zákony, mnohé diskusie však poukazujú na ich nedostatočnosť [7]. Ako najvyspelejšia iniciatíva založená na zhromažďovaní, udržiavaní a šírení princípov UI sa javí iniciatíva založená na princípoch [8]. Tie by mohli usmerňovať vývoj UI počas nasledujúcich desaťročí a storočí a v čase, kedy viaceré výskumné oblasti spolu nespolupracujú a neexistuje zjednocujúca všeobecná teória, mohlo by ísť pomyselné svetlo na konci tunela.

1. Autonómne a prepojené vozidlá

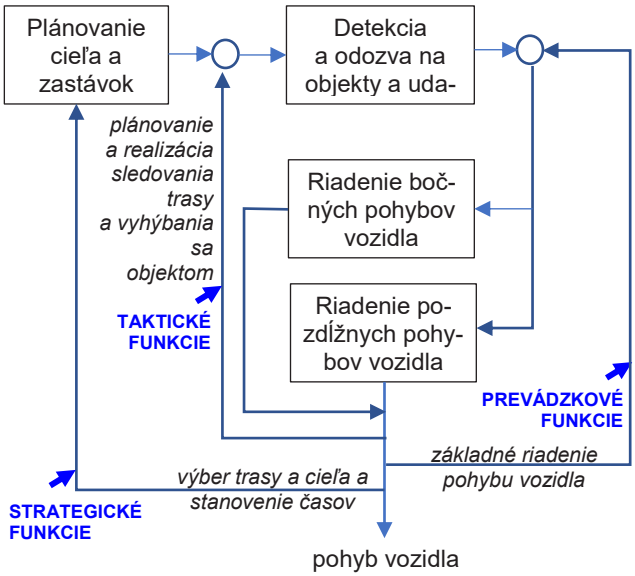
Spôsob, akým sa UI darí alebo nedarí riešiť technické problémy je najviditeľnejší v oblasti vývoja autonómnych vozidiel a robotov, do ktorej sa premietajú najnovšie technologické výdobytky a uplatňujú najnovšie metódy a algoritmy UI. Prínosy UI smerujú najmä do týchto 5 oblastí:

- asistencia a/alebo automatizácia vedenia vozidla;
- poskytovanie služieb pomocou cloudových platforiem;
- aplikácie v poisťovaní vozidiel a riešení nehôd;
- výroba vozidiel,
- monitorovanie vodiča (identifikácia, rozpoznanie, pozorovanie, dokumentovanie).

1.1 Úrovne automatizácie autonómnych vozidiel

Do procesu riadenia vstupujú 3 primárne faktory - vodič, systém riadenia vozidla a ďalšie systémy a komponenty vozidla, ktoré pri uvažovaní prostredia realizujú úlohu dynamického riadenia (obr. 1). Všeobecne akceptovaný [9] zadefinoval 5 úrovní automatizácie (tab. 1), pričom na trhu už sú k dispozícii úrovne 0 až 3, vyššie stupne sú v procese testovania. Kategorizácia je založená na posúdení:

- či systém automatizácie riadenia vozidla realizuje podúlohy bočného a pozdĺžneho pohybu v úlohe dynamického riadenia,
- či systém automatizácie riadenia vozidla realizuje podúlohy bočného a pozdĺžneho pohybu v úlohe dynamického riadenia súčasne,
- či systém automatizácie riadenia vozidla tiež realizuje podúlohu detekcie a odozvy na objekty a udalosti,
- či systém automatizácie riadenia vozidla tiež realizuje odovzdanie úlohy dynamického riadenia,
- či je systém automatizácie riadenia vozidla limitovaný návrhom operačného prostredia.



Obr. 1 Úloha dynamického riadenia [9]

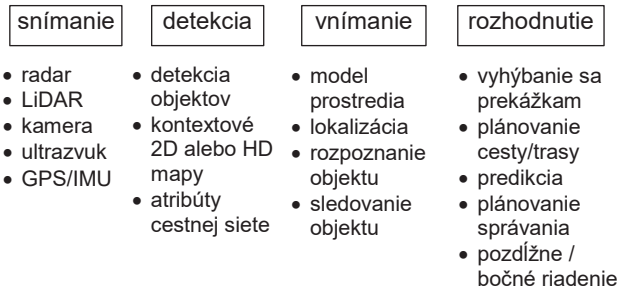
Fig. 1 Dynamic Driving Task [9]

Automatizácia		Vodič	Monitoring
0	iba vodič	pohľad upretý na cestu, ruky na volante	monitorované riadenie
1	vodič s pomocou		
2		ruky dočasne mimo volantu	
3	podmienená automatizácia		nemonitorované riadenie
4	vysoká automatizácia	pohľad mimo cesty, ruky mimo volantu	
5	plná automatizácia		

Tab. 1 Úrovne automatizácie riadenia vozidla podľa [9]

V úrovni 0 má vodič výhradnú kontrolu nad svojím vozidlom, aj keď môžu existovať systémy aktívnej podpory (napr. parkovacie senzory, tempomat na udržiavanie rýchlosti). Úroveň 1 obsahuje asistenčné systémy umožňujúce vozidlu čiastočne reagovať na podnety z okolia (napr. adaptívny tempomat či systém na udržiavanie vozidla v jazdnom pruhu). Realizujú sa podúlohy bočného alebo pozdĺžneho riadenia, nie však obidve súčasne (príklad: 2018 Honda Civic). Úroveň 2 využíva prepojenie technológií prvej úrovne, vďaka čomu je vozidlo schopné na obmedzený čas prebrať úplnú kontrolu nad rýchlosťou a riadením, pri riešení problémov sú však zásahy vodiča nevyhnutné (príklad: 2018 Tesla Model S). V úrovni 3 je vozidlo schopné pri zadaní trasy nahradiť vodiča, v prípade problémov však musí dôjsť k okamžitému odovzdaniu riadenia naspäť vodičovi, čo je hlavná slabina a nebezpečný aspekt tohto konceptu (príklad: 2019 Audi 8). V úrovni 4 sa vyžaduje zásah vodiča iba

v kritických situáciách, zväčša po predošlom automatizovanom zastavení (príklad: Hyundai NEXO – v rámci testovania). Všetky úrovne 1 až 4 sa odohrávajú v obmedzenom operačnom prostredí. Najvyššia úroveň 5 zodpovedá plne autonómnemu vozidlu, ktoré sa pohybuje v ľubovoľnom prostredí a nepredpokladá zásahy vodiča (preto v princípe nemusia ani existovať ovládacie prvky – volant, pedále a pod.). Jej dosiahnutie predpokladá zvládnutie 3 základných prekážok – videnie, vnímanie (porozumenie videného) a konanie.



Obr.2 Kľúčové stavebné bloky autonómneho riadenia

Fig.2 Key building blocks of autonomous driving

obr. 2 uvádza kľúčové stavebné bloky úplnej automatizácie [13]. Vzhľadom na nedokonalosť senzorov musia byť využívané viaceré detekčné princípy, výstupy ktorých sa prekrývajú a dopĺňajú. Snímané dáta musia byť správne interpretované s cieľom v reálnom čase detegovať stacionárne aj pohyblivé objekty v danom prostredí. Zohľadňovať sa musia dopravné podmienky, výskyt chodcov, poveternostné podmienky, stav vozovky, komunikácia s ostatnými subjektmi (V2X), atď. Najťažšou úlohou je prijímať bezpečné rozhodnutia, čo si vyžaduje otestovať a vyhodnotiť miliardy možných aj nepravdepodobných scenárov.

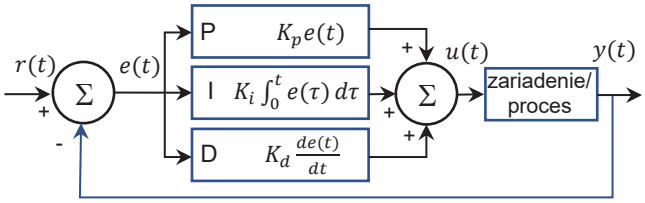
V zmysle vyššie uvedeného môžeme uviesť niekoľko príkladov klasifikácie:

- vozidlo je vybavené adaptívnym tempomatom, pri poruche systému vodič dokončí úlohu dynamického riadenia – úroveň 1,
- vodič je mimo vozidla a iniciuje (bezdrôtovo) manéver automatického parkovania vozidla – úroveň 2,
- vozidlo vybavené systémom automatizácie riadenia vykonáva celú úlohu dynamického riadenia počas dopravnej zápchy, ale nie je schopné v tom pokračovať, ak sa dostane na miesto nehody, vtedy žiada o intervenciu vodiča – úroveň 3,
- vozidlo vybavené systémom automatizácie riadenia vozidla je navrhnuté na prevádzku v rámci areálu univerzity alebo letiska, kde zbiera a vysadzuje cestujúcich na konkrétnej trase – úroveň 4,
- vozidlo vybavené systémom automatizácie riadenia je schopné automatickej navigácie na všetkých cestách za všetkých poveternostných a dopravných podmienok pri zadaní cieľa používateľom – úroveň 5.

V rámci jednotlivých podúloh úlohy dynamického riadenia možno identifikovať rôzne prístupy, napr. pri riadení natočenia kolies (*steering*) prístup založený na:

- metódach nevyužívajúcich UI,
- metódach využívajúcich UI,
- kombinácii oboch vyššie uvedených možností.

Prvý prístup používa teóriu riadenia na výpočet uhla natočenia kolies na udržanie vozidla v požadovanej trajektórii, čo sa deteguje zväčša prostredníctvom algoritmov počítačového videnia. Jedna z najobľúbenejších metód je založená na použití PID radičov. Radič v riadiacej slučke počíta regulačnú odchýlku $e(t)$ ako rozdiel medzi signálom spätnej väzby vozidla a signálom ďalšieho povelu. Korekčná hodnota $u(t)$ vypočítaná podľa (1) obsahuje všetky tri zložky (proporcionálnu, integrálnu, derivačnú) a aplikuje sa na riadený proces (obr. 3). Ako príklad použitia tohto prístupu v oblasti automatizácie riadenia vozidiel možno odporučiť napr. zdroje [14][15].

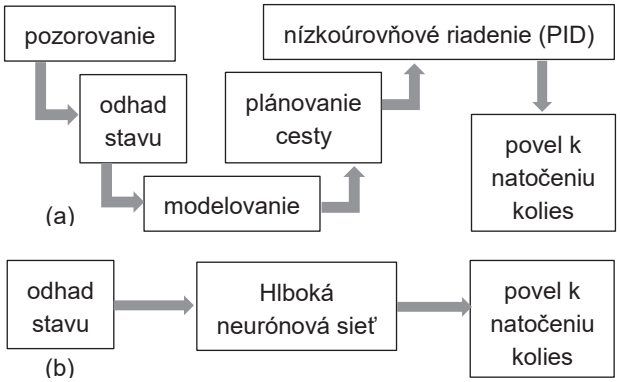


Obr. 3 Výpočet regulačnej odchýlky v PID slučke

Fig. 3 Calculation of correction value in a PID loop

$$u(t) = K_p e(t) + K_i \int_0^t e(\tau)d\tau + K_d \frac{de(t)}{dt} \quad (1)$$

Pri použití UI nepočítame presný uhol natočenia pomocou matematických rovníc, namiesto toho sa spoliehame na inteligentného agenta vyberajúceho najlepší akčný zásah. Agent môže byť trénovaný pomocou hlbokého učenia na veľkých dátových súboroch s cieľom naučiť sa rozpoznávať rysy cesty a predpovedať smer tak, aby sa vozidlo udržalo na vozovke. Štandardne sa úloha dekomponovala do niekoľkých krokov typu detekcia chodcov, detekcia čiar na vozovke, plánovane cesty/pohybu, riadenie motora a pod. (obr. 4a). S využitím metód hlbokého učenia možno dosiahnuť menší a elegantnejší systém (obr. 4b) [16].



Obr. 4 Rozdiel medzi priamym a modularizovaným prístupom

Fig. 4 Difference between and-to-end and modularized approaches

2. Kybernetická bezpečnosť

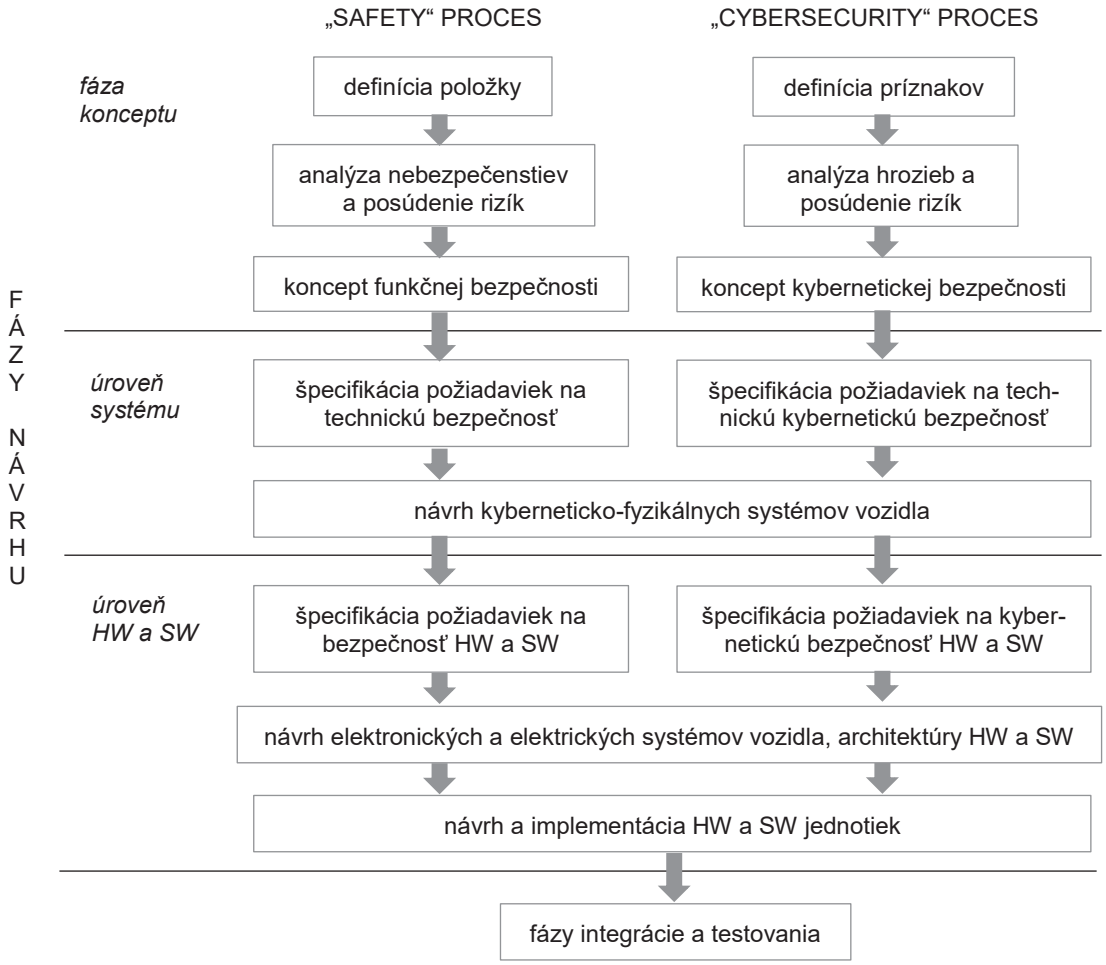
2.1 Legislatívny rámec

Automobilový priemysel podobne ako iné odvetvia priemyslu čelí veľkej výzve v dôsledku digitalizácie. V snahe napomôcť rozvoju autonómnej mobility prijala Európska komisia (EK) stratégiu [10], ktorá vytvára rámec pre ďalší vývoj v zmysle vyššie uvedených skutočností. Okrem technologického zvládnutia viacerých procesov potrebných pre autonómne riadenie tak do popredia čoraz náhlivejšie vystupuje otázka zaistenia bezpečnosti - chápanej jednak v zmysle anglického výrazu „safety“, ale najmä v zmysle výrazu „security“ (fyzickej, informačnej, kybernetickej). Problematika kybernetickej bezpečnosti v súvislosti s autonómnymi vozidlami je vyvolaná novými hrozbami kybernetických útokov vyplývajúcimi z prepojiteľnosti vozidiel a systémovej integrácie tisícok komponentov [10]. Podľa [17] bude 75% z celkového počtu 92 mil. automobilov celosvetovo predaných v roku 2020 pripojených na Internet. V súčasnosti neexistuje odvetvový prístup k ochrane vozidla pred kybernetickými

útokmi. Preto EK prijala v roku 2017 balík opatrení v oblasti kybernetickej bezpečnosti [11], uverejnila usmernenia týkajúce sa bezpečnostnej a certifikačnej politiky potrebné na bezpečnú a dôveryhodnú komunikáciu správ súvisiacich s bezpečnosťou cestnej premávky a riadením dopravy medzi vozidlami a infraštruktúrou [12] a uznesením A8-0036/18/P8_TA-PROV(2018)0063 ju Európsky parlament zaviazal zverejniť legislatívny návrh na zabezpečenie rovnakých podmienok pre prístup k palubným údajom a zdrojom, ktoré majú obrovský potenciál na vytváranie nových a personalizovaných služieb a výrobkov (pomoc, poistenie, oprava, prenájom, represia a pod.). V zámorí bol iniciovaný a neskôr re-iniciovaný obdobný proces (tzv. Security and Privacy in Your Car (SPY) Car Act of 2015, 2017), ktorý zatiaľ viedol k publikovaní návodov [18] a osvedčených postupov [19] pre aplikácie v Spojených štátoch. V rámci vytváraného legislatívneho a štandardizačného procesu však existujú viaceré základné dilemy [17], predovšetkým:

- kto je vlastníkom dát zbieraných z vozidla?
- ide o osobné dáta?
- kto zodpovedá za ich bezpečnosť?
- komu k nim umožniť prístup?
- aká je úloha štandardov a relevantných organizácií?

V súvislosti s informáciami stojí sa pozornosť pripomenutie rozdielov medzi kybernetickou bezpečnosťou a informačnou bezpečnosťou, rovnako dobre aplikovateľné aj v automobilovom sektore, pozri [20].



Obr. 5 Safety a security v návrhu systémov autonómnych inteligentných vozidiel a inteligentnej infraštruktúry [29]

Fig. 5 Safety and security through the design of autonomous intelligent vehicle systems and intelligent infrastructure [29]

Funkčná a prevádzková bezpečnosť prepojených a autonómnych vozidiel je predmetom dlhodobého výskumu. Zložité kyberneticko-fyzikálne systémy v ňom (okrem fyzickej infraštruktúry a fyzických vozidlových systémov) začínajú hrať čoraz dôležitejšiu úlohu. Z hľadiska vývojového procesu nie je kybernetická bezpečnosť iba jedným komponentom navyše, musí byť integrálnou súčasťou fázy plánovania, od samotného konceptu, cez výrobu, prevádzku a údržbu až po vyradenie systému z činnosti, ako je zjednodušene naznačené na obr. 5 [29]. Požiadavky na funkčnú bezpečnosť cestných vozidiel sa definujú v zmysle ISO 26262, ktorá používa prístup založený na rizikách špecifických pre automobilový sektor pre určenie úrovní automobilovej integrity bezpečnosti *ASIL* (*Automotive Safety Integrity Level*). Oblasť kybernetickej bezpečnosti riadi najmä ISO 21434, problematiku ochrany údajov ISO 27001 a mnohé ďalšie.

2.2 Súčasný stav - dôvody na riešenie problematiky

Dostupné štatistiky jednoznačne preukazujú prudký nárast bezpečnostných incidentov. Podľa [21] v prvom kvartáli 2019 bolo identifikovaných 51 incidentov, čo je v porovnaní s rovnakým obdobím minulého roku (15 incidentov) viac ako 300% nárast (za celé obdobie 2018 išlo o 66 incidentov). Berúc do úvahy posledný prieskum spoločnosti Cisco, existuje analógia s viac ako 350% celoročným nárastom ďalšieho javu v kybernetickej priestore známeho ako ransomware. V niektorých prípadoch bol cieľom tohto typu útoku už aj automobilový sektor. V rámci rozdelenia hackerov na etických (tzv. *white hat*) a „zlých“ (tzv. *black hat*) v prvom štvrťroku 2019 narástol ich percentuálny pomer z 45/55 na 28/72 v prospech *black hat* hackerov, čo je dôkazom rastúcej

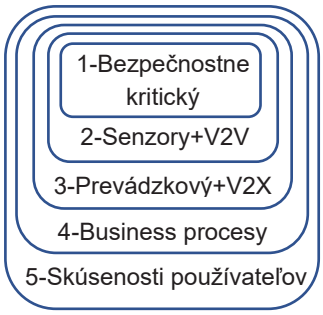
miery rizík. Prehľad cieľov najčastejších vektorov útoku je v tab. 2.

systém bezkľúčového vstupu	47%
servery	17%
mobilné aplikácie	6%
CAN (Controller Area Bus) zbernica	4%
ECU (Engine Control Unit)	4%
Infotainment (informačno-zábavný systém)	4%
OBD (On-board Diagnostics) port	4%
Bluetooth	2%

Tab.2 Najčastejšie vektory útoku v 1. kvartáli 2019 [21]

Štatistické údaje z rovnakého zdroja hovoriace o motivácii útokov uvádzajú snahu o odcudzenie vozidla (40%), narušenie/poškodenie aktivít súvisiacich s podnikaním a poskytovaním služieb (17%), získanie kontroly nad vozidlom (14%), porušenie údajov (10%) či snahu o realizáciu podvodných aktivít (5%).

Predikcie pre rok 2019 nie sú o nič optimistickejšie [22]:
- s každou novou službou nový vektor útoku,
- narušeniu ochrany osobných údajov výmenou / zdieľaním vozidiel,
- nárast a zdokonaľovanie hackerských útokov na autonómne senzory,
- nárast počtu podvodov a zneužitia v oblasti smart mobility,
- nárast útokov na bezkľúčový vstup do vozidla,
- potenciálna možnosť útoku na celú flotilu vozidiel.



LEGENDA:
1 priamy dopad vplyv na bezpečnosť (*safety*) cestujúcich
2 všeobecné systémy vozidla, prostredie
3 speňaženie údajov, telematika, služba riadenia flotily vozidiel
4 hlavné vstupy do bezpečnostne kritických systémov
5 informačne-zábavné systémy, aplikácie, príslušenstvo

Obr. 6 Dekompozícia ekosystému prepojeného vozidla na 5 vzájomne spojených podsystémov [24]

Fig. 6 Decomposition of the connected vehicle ecosystem into 5 interlinked subsystems [24]

Hrozby v oblasti kybernetickej bezpečnosti v ostatných rokoch významne menia svoj charakter [27]:
- *zmena motivácie útočníka* – motivácie môžu byť veľmi rôznorodé (ideologické dôvody, snaha ukázať prevahu, nespokojnosť, donútenie, náhoda, osobné uspokojenie, závislosť, zvedavosť, finančný profit, apod.); principiálne dochádza k posunu od jednotlivca poháňaného motívom zvedavosti k útokom dobre financovaných a vyškolených útočníkov v rámci kybernetickej vojny alebo sofistikovaných aktivít kriminálnych organizácií,
- *nárast rýchlosti a šírky záberu útokov* – od ručného nájdenia slabého miesta softvéru na konkrétnom počítači sa prechádza k automatizovanému vyhľadávaniu slabých miest, ich šíreniu cez Internet a globálnemu ovplyvňovaniu všetkých pripojených zariadení,
- *podstatný nárast potenciálnych dopadov narušenia* – globálne prepojenie zariadení a ľudí znamená, že útoky neovplyvnia iba digitálny svet (ako v minulosti), ale tiež fyzický svet prostredníctvom Internetu vecí a sveta spoločnosti cez všadeprítomné platformy sociálnych médií.

2.3 Potenciál a úloha UI

UI je základným komponentom automatizácie vozidiel a jej využitie ovplyvní mnohé procesy [25]:
- na politickej úrovni musí byť stanovené, za akých podmienok sú/budú vozidlá (s rôznym stupňom automatizácie) považované za bezpečné; obzvlášť dôležité môžu byť „okrajové prípady“ alebo nezvyčajné situácie, ktoré sa nemusia vyskytnúť ani pri testovaní,
- podobne ako v prípade konvenčných vozidiel bude vo verejnom záujme identifikovať problémy s bezpečnosťou a vyšetrovať príčiny nehôd – zatiaľ čo pre vývojárov UI je vznik nehody príležitosťou na vylepšenie techník strojového učenia, pre vyšetrovateľov a zodpovedné orgány bude na stole otázka, akým dielom UI prispela k nehode – nájsť odpoveď však môže byť pri UI systémoch viac ako veľmi zložité,
- automatizované vozidlá a UI systémy musia byť chránené pred škodlivými kybernetickými útokmi; útok hackerov je najsamozrejmejším príkladom, slabinou systému však môže byť napr. oklamanie systému počítačového videnia falošným svetelným zdrojom alebo iným klamlivým signálom, v dôsledku čoho zareaguje systém nebezpečným spôsobom.

V tab. 3 je pre ilustráciu uvedená knižnica klasifikujúca agentov ohrození v automobilovom priemysle podľa [30], rozdeľujúcich ich podľa zámerov na nie-priateľské a nepriateľské. Ten istý zdroj uvádza aj knižnicu metód a cieľov (*MOL – Methods and Objectives Library*) a knižnicu všeobecných expozícií – vystavení sa rizikám (*CEL – Common Exposure Library*), tab. 4, aplikované na autonómne a prepojené vozidlá. Knižnice sa neustále vyvíjajú a nemožno ich považovať za kompletné.

Cieľom bezpečnostných útokov môžu byť všetky piliere, na ktorých stojí kybernetická bezpečnosť. Podľa [23] sú to v prípade automobilových aplikácií 4 základné piliere:
- základňa pre UI návrh a implementáciu,
- vývojová infraštruktúra podporujúca hlboké učenie,
- riešenie dátového centra pre robustnú simuláciu a testovanie,
- pervazívny bezpečnostný program.

V tab. 1 je ako vnútorná vozidlová sieť uvedená CAN, ktorá je najrozšírenejšia. Ide o centralizovanú sieťovú zbernicu, po ktorej sa posielajú všetky dáta vo vozidle. Okrem nej však existujú aj ďalšie: *LIN* (*Local Interconnect Network*) – cenovo priaznivá alternatíva na prepojenie prepínačov, inteligentných aktuátorov, snímačov, malých motorčekov, svetiel, riadenie kúrenia a pod; *Flex-Ray* – sieť podporujúca nové typy *drive-by-wire* systémov (t.j. bez fyzického spojenia vstupného a výstupného zariadenia), ktorá si vyžaduje dobrý manažment chýb spolu s vysokými prenosovými rýchlosťami; alebo *MOST* (*Media Oriented Systems Transport*) majúca najväčšiu šírku pásma, čo ich predurčuje pre audio, video, navigačné a telekomunikačné systémy.

Vyšší počet prepojení (Car2Cloud, LTE, WiFi, atď.) s vonkajším svetom tak zvyšuje bezpečnostné hrozby. Problémy sú v hlavných 4 oblastiach [30]:
- bezdrôtové aktualizácie (*OTA – Over-The-Air updates*),
- nízky výpočtový výkon (na rozdiel od potenciálne vysokého výkonu útočiacich počítačov),

ATRIBÚTY AGENTA OHROZENIA		nie nepriateľský zámer				nepriateľský zámer														
		Nepozorný zamestnanec	Neškolený zamestnanec	Sympatizant smerom navonok	Informačný partner	Hacker aktivista (*Hacktivist*)	Konkurent	Kybernetický vandal	„Dolovač" dát (*Data miner*)	On-line sociálny hacker	Amatér (*Script kiddie*)	Štátny kybernetický bojovník	Organizovaný zločin	Radikálny aktivista	Senzáciechtivý	Kybernetický terorista	Kybernetický zločinec	Vládny špión	Vnútorný špión	Nahnevaný zamestnanec
prístup	zvnútra	■	■	■	■													■	■	■
	zvonka					■	■	■	■	■	■	■	■	■	■	■	■			
výstup	získanie/odcudzenie							■		■	■		■				■		■	
	obchodná výhoda						■		■									■		
	materiálna škoda	■	■					■				■	■	■		■	■			■
	poškodenie cestujúcich											■	■			■		■		
	poškodenie reputácie	■	■	■	■	■		■			■	■		■	■					■
	technická výhoda						■		■									■	■	
	15 min slávy										■				■					
zdroje	jednotlivec	■	■	■													■			■
	spolok, fórum										■				■					
	súťaž							■												
	tím								■	■										
	organizácia				■	■	■						■	■		■			■	
	vláda											■						■		
zručnosti	žiadne																			
	minimálne		■								■				■		■			
	prevádzkové				■			■		■										■
	odborné	■		■		■	■		■			■	■	■		■		■	■	
viditeľnosť	verejné		■											■	■					
	skryté	■		■		■		■					■			■	■			
	tajné				■		■		■	■								■	■	
	„nestará sa"										■	■								■
limity	etický kódex		■		■															
	v rámci zákona	■																		
	mimo zákona – mierne narušenie			■		■	■	■	■	■	■			■	■		■		■	
	mimo zákona – závažné narušenie											■	■			■		■		■
cieľ	skopírovať					■	■		■	■								■	■	
	poprieť											■								
	ublížiť											■	■	■		■		■		
	zničiť											■				■				■
	poškodiť			■								■				■				■
	zobrať												■			■				
	všetko / „nestará sa"	■	■		■			■		■	■			■	■		■			
motivácia	náhodná	■	■		■															
	prinútenie																			
	nespokojnosť																			■
	prevaha							■				■								
	ideológia					■								■		■		■		
	preslávenie sa														■					
	zisk pre organizáciu						■		■				■							
	finančný zisk pre jednotlivca									■							■		■	
	osobné uspokojenie			■							■									
	nepredpovedateľná																			

Tab. 3 Knižnica agentov hrozieb pre automobilový priemysel [30]

- obťažné monitorovanie stavu elektroniky certifikovanou autoritou (vozidlo nie je vždy pripojené na Internet),
- náklady na zabezpečenie SW (v zmysle slova security),
- bez security nie je ani safety – jedno infikované vozidlo môže predstavovať potenciálne nebezpečenstvo pre všetky ostatné.

Uvedené problémy a snaha o ich riešenie sa významným spôsobom premietajú do tvorby otvorenej systémovej architektúry automobilov AUTOSAR (*Automotive Open System Architecture*), ktorá je v súčasnosti de facto štandardom pre výrobcov originálnych zariadení (OEM) a ich dodávateľov v automobilovom priemysle a je v súlade s „automobilovými štandardmi“ ako ISO 15767, ISO 14229, ISO 27145, atď. Vnorený SW je v tejto architektúre hierarchicky rozdelený do 5 vrstiev (podobné OSI modelu) – základný softvér (*MCAL - Microcontroller Abstraction Layer, ECAL - ECU Abstraction Layer, Services Layer*), runtime prostredie a aplikačná vrstva (www.autosar.org). Tab. 5 uvádza taxonómiu potenciálnych zraniteľností a rôzne úrovne vážnosti ich dôsledkov.

ÚROVEŇ	Expozície (vystavenie sa riziku)	Typ prístupu		Potenciálny dopad na:		
		fyzický	bezdrôtový	bezpečnosť	ochrana osobných údajov	násilné odcudzenie
VYSOKÁ	OBD II port	■		■		
	WiFi		■	■		
	3G/4G spojenie		■	■		
	OTA aktualizácia		■	■		
	Infotainment systém		■	■		
	smartphone	■		■		
STREDNÁ	Bluetooth		■	■		
	vzdialený typ app		■	■		
	KeyFob, immobilizéry		■			■
	USB	■		■		
	ADAS systémy		■	■		
	DSCR prijímač (V2X)		■	■		
NÍZKA	DAB rádio		■	■		
	TPMS		■		■	
	GPS		■		■	
	eCall		■	■		
	EV nabíjací port	■		■		
	CD/DVD prehrávač	■		■		

Tab. 4 Tabuľka expozícií [30]

3. Metódy UI pre kybernetickú bezpečnosť

UI má potenciál pomôcť pri identifikovaní zraniteľných miest a ich odstraňovaní, pri detekcii útokov a obrane proti aktívnym útokom. Pomáha nám riešiť zložité problémy spôsobom, akým by to robil sám človek. Mechanizmus rozhodovania podobný ľudskému mechanizmu rozhodovania sa tak snažíme modelovať pomocou nejakých algoritmov. Využívajú sa neurónové siete (pri detekcii DoS, červov, spamu, zombie a klasifikácii škodlivého SW a pod., kedy sa cení najmä ich vysoká rýchlosť), expertné systémy (napr. pri pri výbere bezpečnostných opatrení), inteligentné agenty (ochrana proti DDoS útokom), prehľadávanie (najmä informované), strojové učenie (SU) a pod.

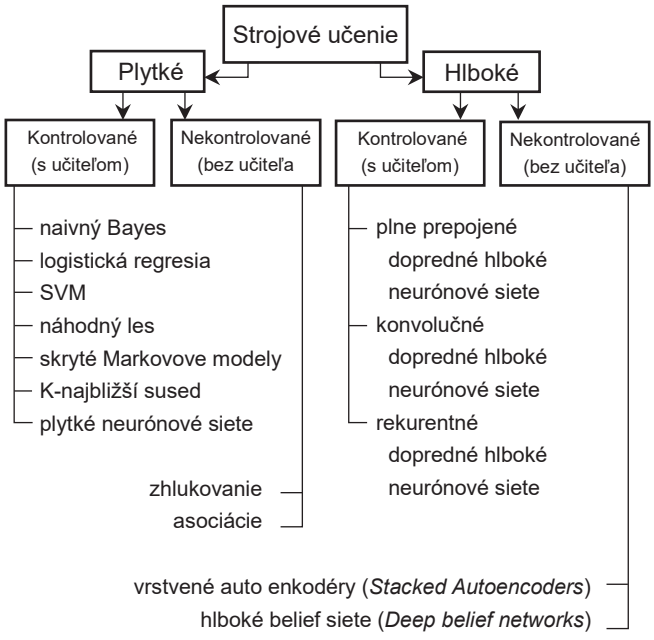
SU je podoblasťou UI, ktorá dáva počítačom možnosť učiť sa bez explicitného naprogramovania. Počítačové programy sa tak po vystavení novým údajom môžu meniť. Dôvodom súčasnej popularity SU je skutočnosť, že máme také množstvo údajov ako nikdy pred tým (veľké dáta) a potrebujeme, aby dávali nejaký zmysel, pretože už nie je v ľudských silách ich analyzovať. SU tak spolu s ľudskou expertízou pomáhajú budovať prediktívne modely kybernetických útokov.

Os rizika:	Závažnosť útoku		
	nízka	stredná	vysoká
počet ovplyvnených vozidiel	útok na individuálne vozidlo	všetky vozidlá daného typu alebo modelu	vozidlá viacerých výrobcov
ohrozené ECU	systémy zábavy, kúrenia a chladenia, pomocné	časti rozhrania riadiaceho spracovanie paliva, rýchlomer alebo navigáciu	Priama zraniteľnosť životne dôležitých systémov ako motor, funkcie brzdenia a riadenia nápravy
Útočiaca entita	sólo hacker	koordinovaná skupina hackerov	národný štát, teroristická organizácia na kritickej dopravnej infraštruktúre
Motivácia útoku	Experimentovanie, nešťastie alebo nuda	Finančné vydieranie (prostredníctvom ransomware) alebo priemyselná špionáž	Úmyselný úmysel poškodiť, kybernetická vojnu na kritickej dopravnej infraštruktúre
Schopnosť replikácie	vyžaduje obrovskú technickú prepracovanosť pre každý prienik	Požadovaná stredná až nízka úroveň technickej sofistikovanosti	Automatizovaný skript, ktorý by po svojom vytvorení a distribúcii komukoľvek dovolil prístup k tej istej zraniteľnosti
Obťažnosť opravy	Trvalá SW bezdrôtová aktualizácia SW môže zaplátať zraniteľnosť	Bezdrôtová nočná aktualizácia vyžadovaná na zaplátanie zraniteľnosti	Fyzické odvolanie vozidla potrebné na odstránenie zraniteľnosti
Odcudzené dáta	obmedzené, anonymizované súbory dát	externé videozáznamy, dáta o správaní sa vodiča	Personalizované dáta o mieste a ceste, interný video / audio záznam, citlivé finančné informácie

Tab. 5 Taxonómia rizík zraniteľnosti [31]

3.1 Strojové učenie

SU zahŕňa veľké množstvo neustále sa vyvíjajúcich paradigiem, ktoré majú vzájomné vzťahy a niekedy prekrývajúce sa hranice. Rôzne pohľady a aplikácie tak môžu viesť k rôznym klasifikáciám. Na obr. 7 je klasifikácia ML algoritmov pre účely kybernetickej bezpečnosti podľa [26].



Obr. 7 Klasifikácia algoritmov strojového učenia pre aplikácie kybernetickej bezpečnosti [26]

Fig. 7 Classification of machine learning algorithms for cyber security applications [26]

Tradičné algoritmy SU sú uvádzané ako „plytké učenie“ (*Shallow Learning*), ako opak k „hlbokému učeniu“ (*Deep Learning*). Prvá kategória si vyžaduje doménového experta, ktorý je schopný identifikovať kritickú úlohu identifikovania relevantných charakteristík dát pred vykonaním samotného algoritmu. Druhá kategória sa spolieha na multi-vrstvovú reprezentáciu vstupných dát a môže vybrať príznaky autonómne v procese nazvanom učenie reprezentácií (*representation learning*). Treba poznamenať, že každá kategória môže obsahovať desiatky rôznych techník (pre aktualizovaný zoznam pozri napr. https://cran.r-project.org/web/views/MachineLearning.html).

3.1.1 Naivný Bayes (*Naive Bayes*)

Tieto algoritmy sú pravdepodobnostné klasifikátory, ktoré vychádzajú z apriórneho predpokladu, že príznaky vstupného súboru údajov sú navzájom nezávislé. Sú škálovateľné a nevyžadujú veľké trénovacie množiny dát na to, aby dosiahli slušné výsledky.

3.1.2 Logistická regresia (*Logistic Regression*)

Ide o kategorické klasifikátory, ktoré využívajú diskriminačný model. Podobne ako v predošlom prípade, aj tieto algoritmy predpokladajú apriórnu nezávislosť prevzatia vstupných funkcií. Ich výkon závisí vo veľkej miere od veľkosti trénovacej množiny.

3.1.3 Metóda podporných vektorov (*Support Vector Machines*)

Ide o nie-pravdepodobnostné klasifikátory, ktoré mapujú vzorky údajov v priestore objektov príznakov s cieľom maximalizovať vzdialenosť medzi každou kategóriou vzoriek. Nevytvárajú žiadny predpoklad o vstupných príznakoch, v klasifikáciách s viacerými triedami však pracujú zle. Preto by sa mali používať ako binárne klasifikátory. Ich obmedzená škálovateľnosť môže viesť k dlhým časom spracovania.

3.1.4 Náhodný les (*Random Forest*)

Ide o súbor rozhodovacích stromov, ktorý berie do úvahy výstup každého stromu pred poskytnutím jednotnej konečnej odpovede. Každý rozhodovací strom je podmienený klasifikátor: strom je prechádzaný zhora nadol a v každom uzle sa kontroluje daná podmienka voči jednému alebo viacerým príznakom analyzovaných údajov. Tieto metódy sú účinné pre veľké súbory údajov a vynikajú pri problémoch s viacerými triedami. Hlbšie stromy môžu viesť k preučeniu.

3.1.5 Skryté Markovove modely (*Hidden Markov Models*)

Modelujú systém ako súbor stavov produkujúcich výstupy s rôznou pravdepodobnosťou; cieľom je určiť postupnosť stavov, ktorá vedie k pozorovaným výstupom. Umožňujú pochopiť časové správanie pozorovaní a vypočítať pravdepodobnosti danej sekvencie udalostí. Hoci môžu byť trénované na klasifikovaných a neklasifikovaných súboroch údajov, v kybernetickej bezpečnosti, sa zväčša používajú s klasifikovanými údajmi.

3.1.6 *K*-najbližší sused (*K-Nearest Neighbour*)

Metóda sa používa na klasifikáciu a možno ju použiť na riešenie problémov s viacerými triedami. Ich učiaca a testovacia fáza sú však výpočtovo náročné, pretože pri klasifikovaní každej testovanej vzorky ju porovnávajú so všetkými vzorkami v trénovacej množine.

3.1.7 Plytká neurónová sieť (*Shallow Neural Network*)

Algoritmy sú založené na neurónových sieťach, ktoré pozostávajú z množiny neurónov organizovaných v dvoch alebo viacerých komunikujúcich vrstvách. Zahŕňajú všetky typy neurónových sietí s obmedzeným počtom neurónov a vrstiev. Napriek existencii nekontrolovanej plytkej neurónovej siete (bez učiteľa) sa v oblasti kybernetickej bezpečnosti používajú najčastejšie na klasifikačné úlohy.

3.1.8 Zhlukovanie (*Clustering*)

Metóda zhlukuje dátové body, ktoré majú podobné charakteristiky. K dobre známym prístupom patrí metóda *k*-priemerov (*k-means*) a hierarchické zhlukovanie. Metódy zhlukovania majú obmedzenú škálovateľnosť, ale predstavujú flexibilné riešenie, ktoré sa typicky používa ako predbežná fáza pred prijatím kontrolovaného algoritmu (s učením) alebo na účely detekcie anomálií.

3.1.9 Asociácie (*Associations*)

Cieľom je identifikovať neznáme vzory medzi údajmi, čo ich robí vhodnými na účely predikcie. Majú však tendenciu produkovať nadmerný výkon nie nevyhnutne platných pravidiel, a preto musia byť kombinované s presnými kontrolami zo strany ľudského experta.

3.1.10 Plne prepojené dopredné hlboké neurónové siete (*Fully-connected Feedforward Deep Neural Networks*)

Variant hlbokej neurónovej siete, kde každý neurón je pripojený ku všetkým neurónom predchádzajúcej vrstvy. Nevytvára žiadny predpoklad o vstupných údajoch a poskytuje flexibilné a univerzálne riešenie pre klasifikáciu za cenu vysokých výpočtových nákladov.

3.1.11 Konvolučné dopredné hlboké neurónové siete (*Convolutional Feedforward Deep Neural Networks*)

Variant hlbokej neurónovej siete, kde každý neurón prijíma svoj vstup len z podmnožiny neurónov predchádzajúcej vrstvy. Táto vlastnosť robí sieť efektívnou pri analýze priestorových údajov, ale jej výkon sa znižuje, keď sa aplikuje na nie-priestorové dáta. Má nižšie výpočtové náklady ako predošlá sieť.

3.1.12 Rekurentné hlboké neurónové siete (*Recurrent Deep Neural Networks*)

Variant hlbokej neurónovej siete, neuróny ktorej môžu posielať svoj výstup aj do predchádzajúcich vrstiev; tento dizajn spôsobuje, že sa učia horšie plne-prepojené neurónové siete. Vynikajú ako sekvenčné generátory, najmä ich posledný variant, dlhá krátkodobá pamäť.

3.1.13 Vrstvené auto enkodéry (*Stacked Autoencoders*)

Skladajú sa z viacerých auto enkodérov, triedy neurónových sietí, kde je počet vstupných a výstupných neurónov rovnaký. Vynikajú v úlohách pred-učenia a v malých súboroch údajov dosahujú lepšie výsledky ako nasledovná metóda.

3.1.14 Hlboké belief siete (*Deep Belief Networks*)

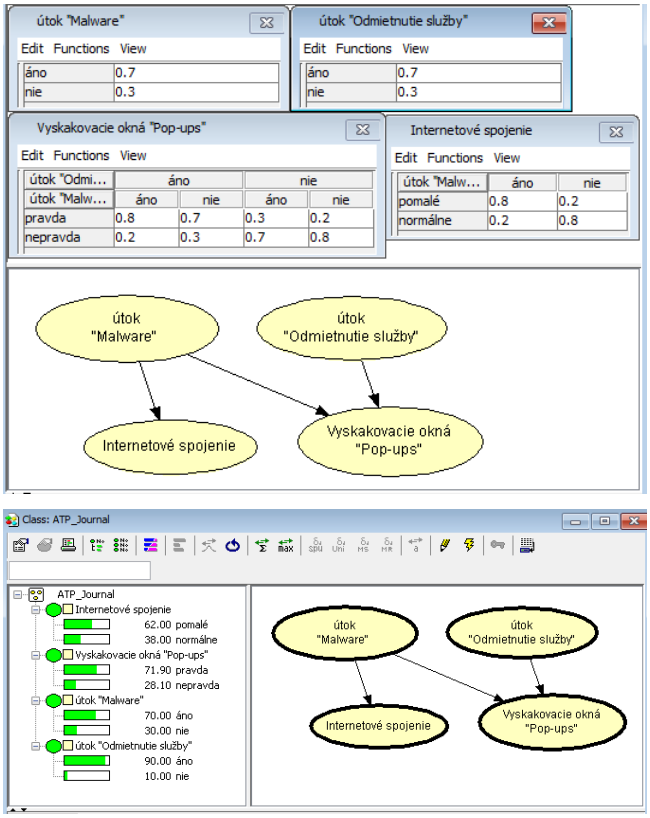
Sú modelované prostredníctvom zostavy tzv. *Restricted Boltzmann Machines*, triedy neurónových sietí bez výstupnej vrstvy. Môžu sa úspešne použiť na úlohy pred-učenia, pretože vynikajú vo funkcii extrakcie vlastností. Vyžadujú si

fázu učenia, ale so súbormi neklasifikovaných údajov (bez učiteľa).

Pokiaľ ide o najčastejšie aplikácie v oblasti kybernetickej bezpečnosti, kde možno vyššie uvedené metódy SU v súčasnosti nájsť, ide o detekciu nepovoleného vstupu (*intrusion detection*), analýzu škodlivého softvéru (*malware analysis*) a detekciu spamu a neoprávneného získavania údajov (*spam and phishing detection*). Podrobnejší pohľad na aplikáciu vyššie uvedených algoritmov pre ten ktorý typ uvažovaného útoku je k dispozícii v [26]. Autonómne schopnosti algoritmov SU sa nesmú preceňovať, pretože absencia ľudského dohľadu môže kvalifikovaným útočníkom uľahčiť preniknutie, odcudzenie údajov a dokonca sabotovanie činnosti celku (systému, podniku, služby). Šesť rôznych dimenzií, ktoré prináša prienik UI, SU a kybernetickej bezpečnosti, je diskutovaných v [27] – legislatívne a politické otázky, ľudský faktor, dáta, hardvér, softvér a algoritmy, a uvedenie do prevádzky (sfunkčnenie). Ďalší pohľad na techniky UI aplikované v oblasti kybernetickej bezpečnosti (v členení na expertné systémy, neurónové siete a inteligentných agentov) možno nájsť napr. v [28].

3.2 Bayesovské siete

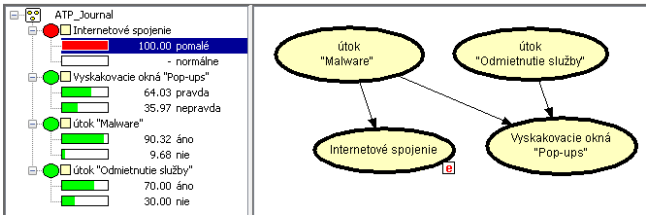
Častou kritikou metód SU je konštatovanie, že je nemožné alebo obťažné predikovať budúcnosť na základe historických dát, namiesto toho by sa mali využívať poznatky. Prekážkou pri tvorbe realistických modelov je nedostatok dostatočného množstva historických dát opisujúcich bezpečnostné narušenia, incidenty a hrozby. Jedným z prístupov ponúkajúcich riešenie sú bayesovské siete (*BBN – Bayesian Belief Networks*) z rodiny pravdepodobnostných grafických modelov, ktoré v sebe spájajú kvantitatívne a kvalitatívne poznatky.



Obr. 8 Príklad modelu bayesovskej siete

Fig. 8 Sample of BBN

Obr. 8 ukazuje jednoduchý model BBN definujúci vzťahy medzi vybranými 4 náhodnými diskrétnymi premennými. Horná časť obsahuje definíciu siete, dolná časť reprezentuje spustený model (v prostredí Hugin Expert A/S 7.6), pričom zelené hodnoty (v percentách) na ľavej strane zobrazujú marginálne hodnoty pravdepodobnostných očakávaní výskytov jednotlivých javov v prípade, keď nebolo realizované žiadne pozorovanie (nemáme žiadny dôkaz o výskyte toho ktorého javu). Model umožňuje realizovať rôzne pozorovania (jednotlivo aj v skupinách) a vyhodnocovať, ako sa menia subjektívne pravdepodobnosti očakávaných javov. Z porovnania obr. 8 a obr. 9 napríklad vidíme, ako nám narástla pravdepodobnosť útoku škodlivého kódu (malware) z hodnoty 0.7 na hodnotu 0.932, keď sme detegovali jeden z jeho možných prejavov – spomalenie internetového spojenia (pozn. - východzie hodnoty pravdepodobností boli v tomto prípade stanovené náhodne).



Obr. 9 BBN pri pozorovaní „pomalého internetového spojenia“

Fig. 9 BBN with “slow Internet connection” observed

Viac o teoretickom pozadí daného formalizmu sa môže čitateľ dozvedieť napr. v učebnici [32]. Systematický prehľad BBN modelov v oblasti kybernetickej bezpečnosti je v [33].

Záver

Existuje jemný, ale pritom zásadný rozdiel medzi „myslieť ako človek“ („silná UI“) a „vykonávať intelektuálne úlohy ako človek“ („všeobecná UI“). Zatiaľ prevládajú aplikácie „slabej UI“. Je zrejmé, že zväzku UI, kybernetickej bezpečnosti a automatizovaných vozidlám patrí budúcnosť, to však platí v dobrom aj v zlom - na jednej strane sa môže UI stať efektívnym nástrojom v rukách útočníkov, na druhej strane si bez UI ťažko predstaviť úspešný rozvoj automatizácie automobilového priemyslu, návrh, testovanie, overovanie, prevádzku a zaistenie bezpečnosti používateľov automatizovaných vozidiel. Napriek tomu, že vznikajúce algoritmy sú čoraz „inteligentnejšie“, je súčasná UI stále veľmi nedokonalá a ľudskí experti sú stále dôležitejší ako ona. Metódy kontrolovaného SU (s učiteľom) sa učia na príkladoch správania sa škodlivého SW a prejavoch rôznych techník útoku a pomáhajú tak pri klasifikácii škodlivého SW, identifikácii spamu či analýze veľkých objemov firewall dát s cieľom predikovať a hodnotiť škodlivé IP adresy. Metódy nekontrolovaného SU (bez učiteľa) sa využívajú na najrôznejšie analýzy (klasifikáciu doménových mien, početnosti vyhľadávaní, prioritizácia IOC), zväčša s využitím prístupov založených na štatistike a pravidlách, a určujú, čo je normálne pre jedinečné charakteristiky chráneného prostredia. Existujú mnohé technologické výzvy, ktorým sa treba v oblasti kybernetickej bezpečnosti venovať:

- dostupnosť a kvalita trénovacích dát – obsahujú množstvo citlivých informácií (intelektuálne vlastníctvo, osobné identifikačné dáta a pod.);
- overovanie a testovanie naučených modelov v reálnych podmienkach;
- neustály vývoj kybernetickej bezpečnosti (potreba neustálej aktualizácie, rozširovania a opakovaného trénovania modelov).

Ukázaný jednoduchý model bayesovskej siete ukazuje jeden z možných prístupov, ktorý je v ostatnom čase často skloňovaný a vedený snahou o zapracovanie a využitie

expertízy ľudských expertov do rozhodovacích (v tomto prípade pravdepodobnostných) modelov.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] URBAN, T.: The AI Revolution: The Road to superintelligence. Wait But Why? Blog https://waitbutwhy.com/2015/01/artificial-intelligence-revolution-1.html

[2] ARMSTRONG, S.: Smarter than us. The rise of machine intelligence. MIRI, 2014, ISBN 978-1-939311-06-1

[3] BOSTROM, N.: Superintelligence. Paths, Dangers, Strategies. Oxford University Press, 2014, ISBN 978–0–19–967811–2

[4] ROUHIAINEN, L.: Artificial Intelligence. 101 things you must know today about our future. 2018, ISBN 1982048808

[5] TEGMARK, M.: Life 3.0: being human in the age of artificial intelligence. Alfred A. Knopf, 2017, ISBN 9781101946602

[6] YAMPOLSKIY, R.V.: Artificial Superintelligence. A Futuristic Approach. CRC Press, 2016, ISBN 978-1-4822-3444-2

[7] BARRAT, J.: Our final invention: artificial Intelligence and the end of the human era. St. Martin Press, 2013, ISBN 978-0-312-62237-4

[8] Asilomar AI Principles. https://futureoflife.org/ai-principles/?submitted=1#confirmation

[9] SAE J3016™ „Levels of Driving Automation“, SAE International, 2018

[10] COM(2018) 283 final: Na ceste k automatizovanej mobilite: stratégia EÚ pre mobilitu budúcnosti. Oznámenie komisie Európskemu parlamentu, Rade, Európskemu hospodárskemu a sociálnemu výboru, výboru regiónov, 2018

[11] Digital Single Market. Policy. Cybersecurity, EK, 2018, https://ec.europa.eu/digital-single-market/en/cyber-security

[12] Intelligent Transport Systems: Cooperative, connected and automated mobility (CCAM). EK, 2019, https://ec.europa.eu/transport/themes/its/c-its_en

[13] SOVANI, S.: Top 3 Challenges to Produce Level 5 Autonomous Vehicles. December 2018, https://www.ansys.com/blog/challenges-level-5-autonomous-vehicles

[14] CHANDNI, C. et al.: Vision based closed loop PID controller design and implementation for autonomous car. International Conference on Advances in Computing, Communications and Informatics (ICACCI), IEEE, Udupi, India: 13-16 Sept.2017

[15] ZHAO, P. et al.: Design of a Control System for an Autonomous Vehicle Based on Adaptive-PID. International Journal of Advanced Robotic Systems, vol. 9, issue 2, 2012

[16] XU, H. et al.: End-to-end learning of driving models from largescale video datasets. arXiv:1612.01079, 2017

[17] The state of security regulation in the connected car ecosystem. Challenges, dilemmas and stakeholders' interests. E-book, Upstream Security Ltd., 2018

[18] Accelerating the Next Revolution in Roadway Safety. Federal Automated Vehicles Policy, NHTSA, Sept 2016

[19] Cybersecurity best practices for modern vehicles. Report No. DOT HS 812 333, NHTSA, Oct 2016

[20] Understanding difference between Cyber Security and Information Security - CISO platform. 2016 http://www.cisoplatform.com/profiles/blogs/understanding-difference-between-cyber-security-information

[21] Q1 2019 sees rapid growth of automotive cyber incidents. Upstream Security Ltd., 2019

[22] Upstream security global automotive cybersecurity report 2019. Research into smart mobility cyber attacks trends. Upstream Security Ltd., 2019

[23] Self-driving Safety Report. NVIDIA Report, 2018

[24] GOLDBERG, J: Traditional IT Cyber Security vs. Automotive Cyber Security Explained. Guard Knox, March 2018

[25] Artificial Intelligence. Emerging Opportunities, Challenges, and Implications. Report GAO-18-142SP, 2018

[26] APRUZZESE, G. et al.: On the Effectiveness of Machine and Deep Learning for Cyber Security. 10th Int. Conference on Cyber Conflict (CyCon), IEEE, 2018

[27] Artificial Intelligence and Machine Learning Applied to Cybersecurity. The results of an intensive three-day IEEE Confluence. IEEE, 6-8 October 2017

[28] PANIMALAR, A.S. et al.: Artificial Intelligence Techniques for Cyber Security. International Research Journal of Engineering and Technology (IRJET), Vol. 05, Issue 03, 2018

[29] TOKODY, D. et al.: Safety and security through the design of autonomous intelligent vehicle systems and intelligent infrastructure in the smart city. Interdisciplinary Description of Complex Systems (16(3-A), 384-396, 2018

[30] KARAHASANOVIC, A.: Automotive Cyber Security. Threat modelling of the AUTOSAR standard. MSc. thesis, Chalmers University of Technology, Gothenburg, 2016

[31] WATNEY, C., DRAFFIN, C.: Addressing New Challenges in Automotive Security. R Street Policy Study, No.118, Nov. 2017

[32] GREGOR, M., JANOTA, A., HRUBOŠ, M.: Kompendium vybraných metód umelej inteligencie 1. VŠ učebnica, Žilinská univerzita v Žiline, 2018, 211 s.

[33] CHOCKALINGAM, S. et al.: Bayesian Network Models in Cyber Security: A Systematic Reviews. Proc. of the Nordic Conference on Secure IT Systems (Nordic 2017), TU Delft, 2017

Abstract

This article deals with the relationship between artificial intelligence and cyber security of autonomous and connected vehicles. It contains an overview of the current state of their development, existing problems and analyzes a subset of artificial intelligence – machine learning methods that appear to be useful and beneficial to cyber security solutions for automotive applications.

prof. Aleš Janota, PhD.

Ing. Roman Michalík

ŽILINSKÁ UNIVERZITA V ŽILINE
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 8215/1
010 26 Žilina
Tel.: +421 41 513 3356
E-mail: ales.janota@fel.uniza.sk

Obsah

SENZOROVÁ SIEŤ, ZBER A VYHODNOTENIE DÁT

Alžbeta Kanáliková

Abstrakt
Bezdrôtové senzorové siete (WSN) sú priestorovo distribuované autonómne senzory používané na monitorovanie fyzikálnych alebo environmentálnych veličín. Článok sa zaoberá popisom technológií WSN a zberom dát a vyhodnotením dát z existujúcej senzorovej siete na báze technológie Libelium v areáli Žilinskej univerzity. Malá bezdrôtová sieť je vybudovaná na základe komunikácie prostredníctvom XBee protokolu alebo wifi protokolu s vybudovaným dátovým rozhraním a aplikáciou, ktorá prezentuje dáta formou grafov a grafických vizualizácií nad mapou areálu Žilinskej univerzity.

Kľúčové slová : Senzorová sieť, Libelium, senzorový uzol, REST architektúra, webová aplikácia.

Úvod

Bezdrôtové senzorové siete (WSN) sú štruktúry zložené z niekoľkých typov uzlov (stovky alebo tisíce). Uzly WSN sú rozptýlené a obsahujú vyhradené senzory na monitorovanie a zaznamenávanie fyzikálnych podmienok prostredia a organizovanie zozbieraných údajov v centrálnom mieste. WSN meria podmienky prostredia, ako je teplota, zvuk, úrovne znečistenia, vlhkosť, rýchlosť a smer vetra, tlak atď. Na vybudovanie senzorovej siete boli použité jednotlivé prvky od španielskej firmy Libelium, ktorá poskytuje základné senzory rôznych typov a modulov a router Meshlium. V areály Žilinskej univerzity boli konkrétne použité senzory radiácie, senzory na meranie kvality ovzdušia, senzory na rozlíšenie typu signálu, napr. wifi signálu, resp. GPS signálu a parkovacie senzory. Senzorová sieť sa neustále rozvíja, Dátové rozhranie je prispôsobené na možné rozšírenie siete o ďalšie vznikajúce senzorové uzly. Dáta sú zaznamenávané do databázy a prostredníctvom aplikačného rozhrania prenášané na webový portál kde je možné dáta analyzovať podľa jednotlivých parametrov.

1. Senzorová sieť

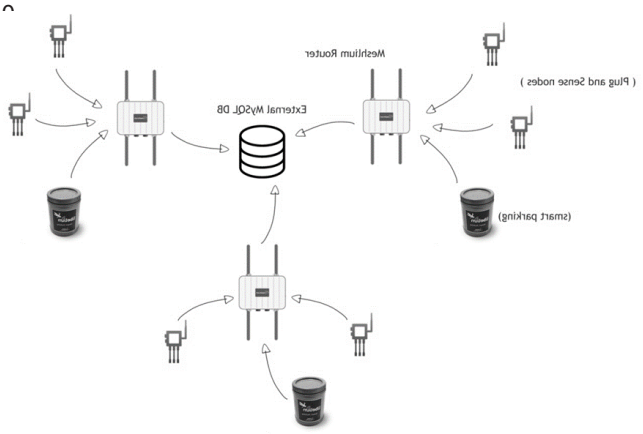
Bezdrôtové senzorové siete sú tvorené veľkým množstvom senzorových uzlov schopných komunikovať bezdrôtovo. Sieťová topológia nie je vopred definovaná a vytvorí sa po inicializácii siete, prípadne sa mení v priebehu životnosti siete vplyvom zmien v počte uzlov, prostredia, ovzdušia a pod.

Senzorová sieť na ŽÚ (Žilinskej univerzite) je navrhnutá prostredníctvom komponentov od španielskej firmy *Libelium.* Sieť sa skladá z viacerých uzlov, ktoré sú vytvorené senzormi, konkrétne napríklad modulárna doska s názvom *Waspmote* v rôznych prevedeniach s možnosťou osadenia vlastných senzorov alebo riešenia, ktoré kombinujú hotové senzorové dosky s doskami na osadenie vlastných senzorov, nazývanými *Plug & Sense.* Ďalším typom senzora je parkovací senzor, ktorý deteguje obsadenie parkovacieho miesta [1]. Všetky typy senzorov sú na obr. 1.

Obr. 1 Rôzne typy senzorov
Fig. 1 Different types of sensors

V areály ŽÚ je vytvorených viacero malých senzorových sietí zložených z niekoľkých senzorov (uzlov). Vstupnou bránou pre každú sieť je router nazvaný *Meshlium*, ktorý obsahuje operačný systém *Linux*. Jeho prevedenie je modulárne a podľa zvolenej hardvérovej konfigurácie môže obsahovať až štyri z ponúkaných komunikačných rozhraní: WiFi 2.4GHz, WiFi 5GHz, 3G/GPRS, Bluetooth alebo *ZigBee* (868/915/2400 MHz).*Meshlium* okrem smerovania, resp. posielania dát do siete *Wifi, Ethernet,* Bluetooth dokáže dáta aj ukladať do databázy a parsovať. Router beží na plnohodnotnom operačnom systéme a je možné doň nahrať vlastné programy, ktoré pobežia v predinštalovaných *runtimoch* (C++, Java, Ruby, PHP, Perl a C) alebo doinštalovať vlastný *runtime*. Dokonca *Meshlium Xtreme* je nastavené ako skener a umožňuje detegovať prítomnosť zariadení napr. iPhone, Android a vo všeobecnosti, hocijaké zariadenie pracujúce s Wifi alebo Bluetooth rozhraním [2], [3].

Komunikácia v senzorovej sieti závisí od počtu uzlov. Uzly sú osadené komunikačnými modulmi rodiny XBee s protokolmi XBee-PRO ZB a XBee-PRO 802.15.4, XBee-PRO 868, XBee-PRO DigiMesh a XBee Wi-Fi [3]. Názorná topológia siete je znázornená na obr. 2.



Obr. 2 Topológia senzorovej siete (*Meshlium* a jednotlivé senzory)

Fig. 2 Sensor Network Topology *(Meshlium* and Individual Sensors)

Vzhľadom na rôznorodosť používaných komunikačných protokolov na prenos dát vo WSN sa prispôsobili komunikačné parametre pre konkrétny protokol (štandard). Na oživenie siete bolo nutné naprogramovať senzorové uzly v súlade s dodržaním logickej hierarchie siete a nastaviť databázový výstup [4].

2. Dátové rozhranie

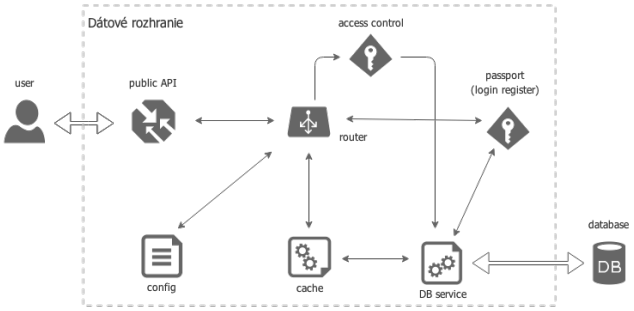
Ako už bolo spomínané routre *Meshlium* poskytujú vlastnú vnútornú databázu - MySQL a parser rámcov, ktorý rozkladá ich dátový obsah a ukladá do tabuľky. Dátová časť všetkých rámcov použitých koncových zariadení je štrukturálne jednotná a výstupom parseru je unifikovaný výstup [2]
Na prístup k dátam bolo potrebné vybudovať jednotné rozhranie, ktoré umožní aj ďalšiu rozšíriteľnosť do budúcnosti. Konkrétne aplikačné dátové rozhranie (API) na báze jednoduchej webovej služby je budované prostredníctvom architektúry REST (*Representational State Transfer*). Má prístup k databáze a používa jednoduché metódy HTTP (GET, POST, PUT, PATCH, DELETE) [4].

2.1 Databáza

Databáza obsahuje základné dáta zo senzorov, ktoré sú rozdelené podľa prijatých paketov z wifi, alebo scanovacích zariadení z *Meshlia Xtreme*, ktoré sníma jednotlivé typy signálov. Obsahuje aj ďalšie dáta potrebné pre webové rozhranie, napr. informácie o užívateľoch, typoch senzorových dosiek, uzloch jednotlivých sietí a ich protokoloch. Tieto dáta sú potrebné na konfiguráciu sietí. Všetky dáta sú umiestnené v databázových tabuľkách s rovnomenným názvami. Dáta zo senzorov sa do databázy pridávajú automaticky, priamo z jednotlivých senzorov. Ostatné dáta sa do tabuliek databázy pridávajú prostredníctvom webového rozhrania.

2.2 Spracovanie dát z databázy

Spracovanie dát z databázy zabezpečuje spomínané API rozhranie. Na prístup k dátam sú zadefinované 3 stupne autorizácie: verejný prístup, prihlásený používateľ, administrátor.
Rozhranie pri vyžiadaní dát (pomocou metódy GET) vracia všetky dostupné dáta požadovanej kategórie z databázy vo formáte JSON (dáta sú navyše dostupné aj vo formáte CSV). Aby bolo možné pristupovať len ku relevantným údajom, napríklad len k údajom určitej meranej veličiny alebo údajom z časového rozsahu sú zadefinované *query* parametre, pomocou ktorých je možné dáta filtrovať v rámci výsledkov z dopytov. Výsledné dáta sú vo formáte *JSON*, alebo aj vo formáte csv. Formát *JSON (JavaScript Object Notation)* sa používa na získavanie prostriedkov zo servera na pozadí kódu vykonávaného v klientovi (webovom prehliadači). Spomínaná REST web aplikácia, konkrétne *RESTfull API* aplikácia používa platformu *NodeJS* s modulom *ExpressJS* na strane servera. Používatelia majú prístup len k povrchovej vrstve dátového rozhrania, ktoré tvoria URL cesty a vnútorné závislosti a procesy ostávajú zaobalené. Vnútornú štruktúru API rozhrania je možné vidieť na obr. 3.



Obr. 3 Vnútorná štruktúra API rozhrania

Fig. 3 Internal API Interface Structure

Každá požiadavka na API (aplikačné rozhranie) prechádza cez router (obsahuje databázu), ktorý volá priradené obslužné funkcie roztriedené podľa HTTP metód. V prípade, že prístup ku zdroju (akékoľvek dáta v tabuľkách databázy) vyžaduje prihlásenie alebo aj autorizáciu, obsahuje cesta navyše *middleware* ("prostredníka"), ktorý predspracuje požiadavku - napríklad overí prístupové práva.

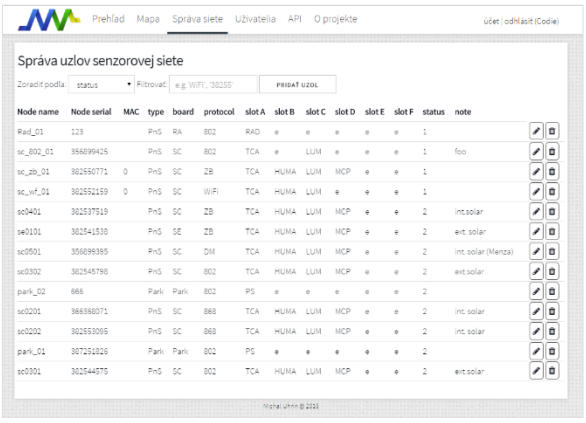
3. Webová aplikácia

Webová aplikácia sa skladá z:

- Dátového rozhrania, ktoré už bolo spomínané postaveného na platforme *NodeJS* a frameworku *ExpresJS* s technológiou *Javascrip*t. Tvorí stranu servera.
- Užívateľská časť nazýva sa aj tzv. Front END a beží v prehliadači. Je podobne vytvorená v *Javasripte* a používa *framework AngularJS*. Na vizualizáciu dát je použitá aj knižnica *C3.js* a využíva aj knižnicu *Google Maps API* na zobrazovanie máp [4].

Hlavnou úlohou webovej aplikácie je zobraziť potrebné dáta z uzlov siete, t. z. filtrovať dáta, ktoré nie sú v dopyte, napríklad ak si vyberieme dáta z vybraného uzlu alebo dáta za určité obdobie a pod. V aplikácií sú definované nasledujúce obrazovky: prihlásenie, prehľad veličín, manažment siete (umožňuje riadiť danú senzorovú sieť), mapa siete – zobrazuje mapu senzorovej siete. Na obr. 4 je ukážka výstupu webovej aplikácie z úpravy uzlov v senzorovej sieti.

Webová stránka pracuje prostredníctvom webových služieb (konkrétne REST architektúra), ktoré napríklad pripoja web k databáze, prihlásia používateľov, obslúžia prichádzajúce alebo odchádzajúce požiadavky na server, zobrazia mapu s konkrétnymi uzlami siete, zobrazia grafy výstupov z jednotlivých uzlov a pod.

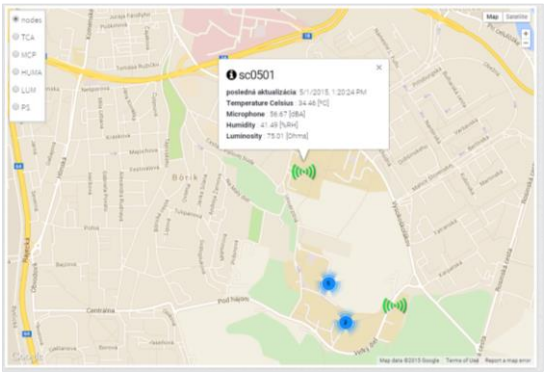


Obr. 4 Webová aplikácia – správa uzlov v senzorovej sieti

Fig. 4 Web application - node management in sensor network

3.1 Zobrazenie mapy uzlov senzorovej siete

Webová aplikácia na zobrazenie mapy jednotlivých uzlov používa mapy od spoločnosti *Google*, resp. *Googlemaps* kde prostredníctvom rozhrania - *Googlemaps API* zobrazuje jednotlivé uzly. Zobrazenie vybraného uzlu senzorovej siete je na obr. 5.

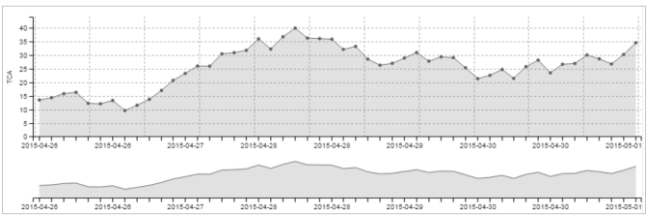


Obr. 5 Zobrazenie vybraného uzlu senzorovej siete na mape

Fig. 5 View selected node of sensor network on the map

3.2 Vyhodnotenie dát

Dáta z jednotlivých uzlov sú vo webovej aplikácií graficky vyhodnocované a zobrazené v grafoch. Graf zobrazuje namerané dáta podľa dní a hodín, čo umožňuje štatistické vyhodnotenie dát za určité časové obdobia (obr. 6).

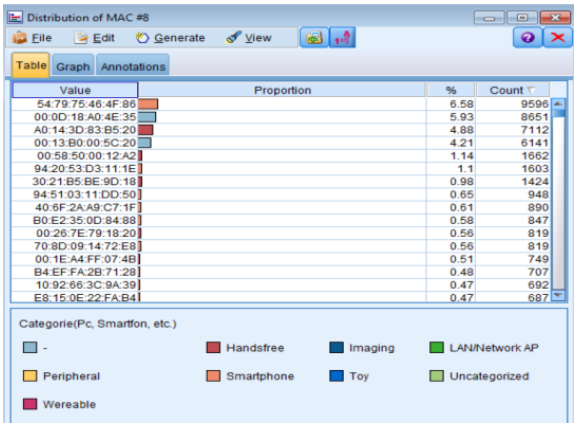


Obr.6 Grafické znázornenie dát zo senzora

Fig.6 Graphical representation of sensor data

Analýza dát zo senzorov je veľmi zaujímavá platforma pre ďalší výskum. Veľmi vhodné je prepojenie webovej aplikácie a databázy s platformou od firmy IBM, ktorá umožňuje prostredníctvom aplikácií a nástrojov na analýzu a dolovanie dát dáta analyzovať a získať z nich nové informácie. Konkrétne napr. *IBM Analytics, IBM Watson, IBM IOT* a pod.

Príklad analýzy a dolovania dát použitím nástrojov od firmy IBM na dátach zo senzorov siete ŽÚ, konkrétne MAC adresy zosnímaných zariadení používajúcich Wifi sieť, je zobrazený na obr. 7.



Obr. 7 Príklad - Analýza MAC adries s nástrojom od firmy IBM

Fig. 7 Example - MAC Address Analysis with IBM Tool

Záver

V dnešnej dobe je koncept IOT (*Internet of Things*) veľmi živým a reálnym konceptom súčasnosti a hlavne budúcnosti. Senzor, senzorové siete sú neodmysliteľnou súčasťou na budovanie inteligentných miest (*Smart cities*), inteligentných vecí, áut, zariadení atď. Navrhnutá sieť a aplikácia na jej ovládanie a vyhodnotenie dát umožňuje študentom a učiteľom spoznávať potrebné zručnosti ale aj vedomosti z implementácie senzorových sietí. Rovnako umožňuje získať reálne dáta zo senzorov, ktoré je možné ďalej analyzovať, dolovať prostredníctvom dataminingových nástrojov, techník, napr. aj pomocou platformy od firmy IBM. V budúcnosti sa plánuje rozšírenie senzorovej siete a hlbšia analýza a získavanie znalostí zo senzorových dát.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] Waspmonte technical Guide [Online] Libelium: http://www.libelium.com/downloads/documentation/waspmote_technical_guide.pdf

[2] Meshlium Xtreme Datasheet [Online] Libelium: http://www.libelium.com/downloads/documentation/v12/quickstart_guide_meshlium.pdf

[3] Waspmonte Zigbee networking Guide [Online] Libelium:

http://www.libelium.com/downloads/documentation/waspmote-zigbee-networking_guide.pdf

[4] UHRÍN M.: Návrh aplikácie na vyhodnocovanie dát zo senzorovej siete, diplomová práca 2015, vedúci práce: Ing. A.Kanáliková, PhD.

Abstract

Wireless Sensor Networks (WSNs) are spatially distributed autonomous sensors used for monitoring physical or environmental variables. The article deals with the description of WSN technologies and data collection and data evaluation from an existing sensor network based on Libelium technology at the University of Žilina. Small wireless network is built on the basis of communication via XBee protocol or wifi protocol with built-in data interface and application that presents data in the form of graphs and graphical visualizations over the map of the University of Žilina.

Ing. Alžbeta Kanáliková, PhD.

Žilinská univerzita
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 1
010 26 Žilina
alzbeta.kanalikova@fel.uniza.sk

Obsah

POUŽITIE STAVOVÉHO DIAGRAMU UML NA PROGRAMOVANIE SAFETY PLC

Milan Medvedík, Juraj Ždánsky

Abstrakt
Príspevok sa zaoberá systematickým prístupom k tvorbe programu pre safety PLC (Programmable Logic Controllers) na základe opisu požadovanej funkcie stavovým diagramom UML (Unified Modeling Language). Takýto postup možno použiť pri dosahovaní systematickej integrity bezpečnosti riadiaceho systému so safety PLC. Ako softvérová podpora UML je použitý nástroj Rhapsody a aplikačný príklad je realizovaný na safety PLC Simatic.

Kľúčové slová: UML, Rhapsody, safety PLC, stavový diagram, bezpečnostná funkcia

Úvod

Dosiahnutie požadovanej úrovne integrity bezpečnosti (SIL-Safety Integrity Level) je dané zaistením integrity bezpečnosti proti náhodným poruchám a integrity bezpečnosti proti systematickým chybám. Dosiahnutie požadovanej integrity bezpečnosti proti náhodným poruchám súvisí s voľbou vhodných opatrení (pri elektronických systémoch ide predovšetkým o voľbu architektúry systému), ktorých cieľom je zaistiť vopred definované správanie sa v prípade poruchy hardvéru. Voľbou vhodnej architektúry vzhľadom na požadovanú SIL sa podrobnejšie zaoberá napr. [1], [2], [3].

Systematické chyby sa týkajú hardvéru aj softvéru riadiaceho systému. Pri komerčne dostupných riadiacich systémoch (akými sú aj safety PLC), možno predpokladať, že prípadné systematické chyby hardvéru sa relatívne rýchlo zistia a odstránia (vďaka veľkému množstvu aplikácií komerčne dostupných riadiacich systémov). Problémom preto zostávajú systematické chyby softvéru, ktorý je jedinečný pre každú aplikáciu (s výnimkou niektorých štandardizovaných funkcií). Preto je nevyhnutné v jednotlivých fázach životného cyklu systému pamätať na opatrenia, ktoré vedú k minimalizácii systematických chýb.

Vo fáze vývoja riadiaceho systému je dôležité, aby bol na báze formálnych prípadne poloformálnych metód vytvorený prehľadný a zrozumiteľný model, ktorý umožní odstrániť prípadné nejasnosti alebo protirečenia v neformálnej špecifikácii a umožní preskúšať komplexnosť a bezchybnosť špecifikácie. Použitie vhodného formalizmu na opis správania sa riadiaceho systému (resp. opis správania sa bezpečnostných funkcií) výrazne zefektívni prácu programátora a minimalizuje systematické chyby softvéru.

Voľba metódy na opis správania sa systému bude závisieť, okrem iného, aj od vlastností aplikačného hardvéru. Pri tvorbe softvéru realizujúceho bezpečnostné funkcie sa vo všeobecnosti treba vyhnúť takým postupom, ktoré síce minimalizujú program (programové triky, zhusťovanie kódu, spätné skoky, ...), ale môžu byt zdrojom systematických chýb v programe.

Pri riadiacich systémoch so safety PLC, ktorými sa zaoberá aj tento článok, sú niektoré z týchto zásad implementované priamo v softvérovom prostredí na tvorbu aplikačného softvéru (napr. safety PLC majú obmedzený inštrukčný súbor).

Pre rôznorodosť bezpečnostných funkcií, na realizáciu ktorých sa safety PLC používajú, sa nedá jednoznačne určiť najvhodnejší postup na elimináciu systematických chýb. Pri jednoduchších bezpečnostných funkciách s nižšou požadovanou SIL výrobcovia safety PLC odporúčajú použiť predpripravené a certifikované funkcie a dôsledne testovať vytvorený aplikačný program (s cieľom odhaliť potenciálne systematické chyby).

Pri zložitejších bezpečnostných funkciách sú však predpripravené funkcie nedostačujúce a vytvorený softvér sa reálne nedá otestovať úplne. Preto je v takýchto prípadoch použitie systematických postupov pri tvorbe softvéru nevyhnutné. Systematický postup treba voliť tak, aby jeho použitie bolo podporené vhodným softvérovým nástrojom, umožňujúcim napr. modelovanie funkčnosti. Systematickým prístupom k tvorbe softvéru sa zaoberá napr. [4], [5] a verifikáciou vytvoreného softvéru PLC sa zaoberá napr. [6], [7], [8].

V tomto článku sa na modelovanie činnosti systému používa UML a softvérový nástroj Rhapsody. Príspevok sa zaoberá postupom tvorby programu pre safety PLC na základe modelu vytvoreného v softvérovom nástroji Rhapsody.

1. Konečný automat ako model funkčného správania sa safety PLC

Bezpečnostné funkcie realizované na safety PLC možno považovať za sekvenčné riadiace úlohy, preto sa dá ich správanie modelovať konečným automatom. Konečný automat je algebraický systém vhodný na opis správania diskrétneho, dynamického systému s pamäťou, ktorý možno definovať ako usporiadanú päticu

$$\boldsymbol{M} = (X, S, Y, u, v), \qquad (1)$$

kde X je množina vstupných slov (vstupov), S je množina stavov a Y je množina výstupných slov (výstupov); u a v sú zobrazenia, pričom zobrazenie u sa nazýva prechodová funkcia a zobrazenie v výstupná funkcia.

Pretože konečný automat ako model funkčného správania sa safety PLC chceme použiť na tvorbu aplikačnej časti safety programu, treba zaviesť analógiu množín a zobrazení z pätice (1) tak, aby množiny boli adresovateľné a zobrazenia realizovateľné safety programom.

Na obr. 1 je znázornený safety PLC ako konečný automat. Firmvér v F-CPU (Fail-safe Central Processing Unit) riadi vykonávanie operačného cyklu a v spolupráci so systémovou časťou safety programu zaisťuje kontrolné mechanizmy safety PLC. Perióda volania operačného cyklu predstavuje analógiu synchronizačnej frekvencie synchrónneho automatu.

Množinu vstupných slov možno definovať pomocou karteziánskeho súčinu nasledujúco:

$$X = \left\{\{x_1^I, x_2^I, \dots, x_q^I\} \times \{x_1^P, x_2^P, \dots, x_k^P\}\right\}, \quad (2)$$

kde $x_1^I, x_2^I, \dots, x_q^I$ sú vstupné slová prichádzajúce zo vstupného modulu (F-I) a q je počet týchto slov (tieto vstupné slová predstavujú informácie z riadeného procesu); $x_1^P, x_2^P, \dots, x_k^P$ sú vstupné slová umožňujúce prepojenie na systémové informácie safety PLC (napríklad možno pomocou nich realizovať časové závislosti safety PLC, prijatie informácií odoslaných z iného safety PLC a pod.; v konečnom dôsledku ide o vstupné slová z pamäte F-CPU) a k je počet týchto slov.

Vzhľadom na činnosť konečného automatu treba rozlíšiť aktuálny stav $s(t)$ od stavu v nasledujúcom časovom okamihu $s(t + t_t)$ (tento stav je výsledkom prechodovej funkcie; prechodovú funkciu realizuje safety program; t_t je interval vykonávania safety programu). Preto musia existovať dve množiny slov.

- Množina slov $\{x_1^S, x_2^S, \dots, x_r^S\}$. Slovo x_i^S je vstupným slovom safety programu a identifikuje stav s_i, v ktorom sa konečný automat nachádza v čase *t*.
- Množina slov $\{y_1^S, y_2^S, \dots, y_r^S\}$. Slovo y_j^S je výstupným slovom safety programu a identifikuje stav s_j, v ktorom sa konečný automat bude nachádzať v čase $t + t_t$.

Tieto výstupné slová tvoria spätnú väzbu konečného automatu a sú zapisované do pamäte F-CPU.

Množinu výstupných slov možno definovať pomocou karteziánskeho súčinu nasledujúco:

$$Y = \left\{\{y_1^O, y_2^O, \dots, y_h^O\} \times \{y_1^P, y_2^P, \dots, y_g^P\}\right\}, \quad (3)$$

kde $y_1^O, y_2^O, \dots, y_h^O$ sú výstupné slová zapisované na výstupný modul (F-O) a h je počet týchto slov (nastavenie výstupov safety programom realizuje výstupnú funkciu; týmito výstupnými slovami možno ovplyvňovať riadený proces podľa požiadaviek bezpečnostnej funkcie); $y_1^P, y_2^P, \dots, y_g^P$ sú výstupné slová ovplyvňujúce pamäťový priestor súvisiaci s realizáciou systémových funkcií safety PLC (napríklad odoslanie informácií do iného safety PLC) a g je počet týchto slov.

Stavový diagram je grafickou formou zápisu konečného automatu a možno ho považovať za poloformálny model správania sa bezpečnostnej funkcie.

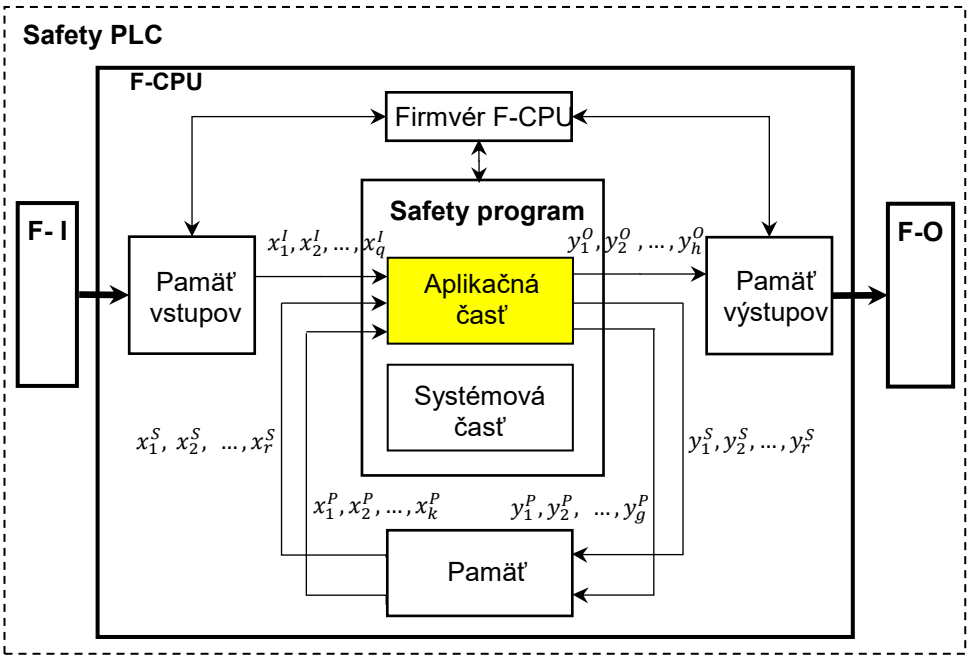
2. Aplikačný príklad

Praktické využitie konečného automatu ako modelu správania sa safety PLC je prezentované na konkrétnom aplikačnom príklade. Tento je zámerne zvolený tak, že realizuje jednu z certifikovaných funkcií ponúkaných výrobcom safety PLC Simatic. Certifikovaná funkcia je potom použitá na verifikáciu funkcie vytvorenej na základe stavového diagramu.

Predpokladajme funkciu, ktorá vyhodnocuje zhodu dvoch vstupov, pričom toleruje definovaný čas nezhody. Certifikovaná funkcia je v prostredí TIAportal označená ako EV1oo2DI a jej správanie je nasledovné.

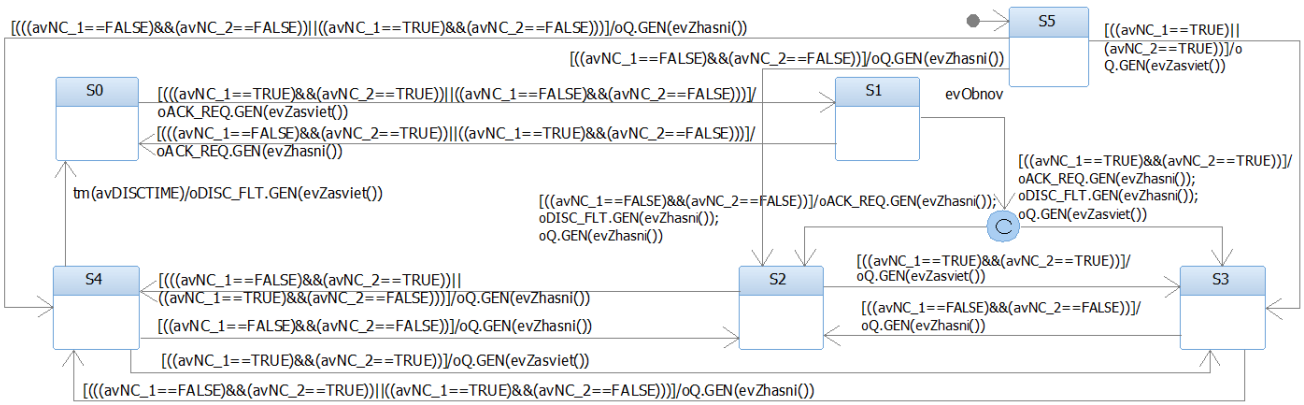
Výstup funkcie (Q) sa nastaví na hodnotu true, ak sú vstupné signály (*IN1* a *IN2*) rovné hodnote true a nie je zaznamenaná chyba prekročenia času nezhody (výstup *DISC_FLT* nie je v hodnote true). Ak je signál jedného alebo oboch vstupov rovný hodnote false, potom je výstup (Q) nastavený na hodnotu false.

Akonáhle sú vstupné signály (*IN1* a *IN2*) odlišné, začne sa merať čas nezhody (maximálny čas nezhody *DISCTIME* je



Obr. 1 Safety PLC ako konečný automat

Fig. 1 Safety PLC as finite automaton



Obr. 2 Stavový diagram

Fig. 2 Statechart

vstupný parameter funkcie). Ak sú signály oboch vstupov odlišné aj po uplynutí času nezhody, je detegovaná chyba prekročenia času nezhody (výstup *DISC_FLT* je nastavený na hodnotu true).

Ak chyba zanikne (vstupné signály *IN1* a *IN2* majú rovnaké logické hodnoty), tak je signalizovaná požiadavka na obnovu funkcie po detegovanej chybe (výstupný signál *ACK_REQ* sa rovná hodnote true). Obnova nastane po nábehovej hrane na vstupe (*ACK*).

2.1 Model správania sa funkcie EV1oo2DI v UML – stavový diagram

Na vytvorenie modelu správania sa funkcie EV1oo2DI je použitý modelovací jazyk UML. UML slúži na vizualizáciu, špecifikáciu, navrhovanie a dokumentáciu softvérových systémov. Skladá sa z diagramov, ktoré opisujú štruktúru systému (diagram tried, diagram objektov, diagram balíčkov,...) a správanie sa systému (diagram aktivít, diagram prípadov použitia, stavový diagram,...). Nie vždy je nevyhnutné použiť na opis systému všetky diagramy. Podrobnejšiu špecifikáciu UML možno nájsť na oficiálnych internetových stránkach [9].

Stav	Opis stavov
S0	Stav, v ktorom sa funkcia nachádza po prekročení dovoleného času nezhody vstupných signálov (bezpečný stav, tzv. stav pasivácie).
S1	Stav, v ktorom sa funkcia nachádza po prekročení dovoleného času nezhody vstupných signálov a následnom zániku nezhody (stav čakania na potvrdenie obnovy).
S2	Stav, v ktorom sa funkcia nachádza, ak majú vstupné signály hodnotu false (normálny pracovný stav - výstup *Q* má hodnotu false).
S3	Stav, v ktorom sa funkcia nachádza, ak majú vstupné signály hodnotu true (normálny pracovný stav - výstup *Q* má hodnotu true).
S4	Stav, v ktorom sa funkcia nachádza, ak sú vstupné signály v nezhode a ešte neuplynul dovolený čas nezhody (stav tolerovania povoleného času nezhody).
S5	Inicializačný stav, v ktorom sa funkcia nachádza po spustení systému.

Tab. 1 Význam stavov v stavovom diagram na obr. 2

Na opis správania sa funkcie EV1oo2DI je použitý stavový diagram (obr. 2) vytvorený v softvérovom nástroji Rhapsody. Výhoda tohto nástroja spočíva v tom, že po vytvorení príslušných diagramov umožňuje generovať softvér v jazyku C++ a následne vytvoriť grafické rozhranie. Pomocou neho možno simulovať správanie sa vytvoreného modelu a overiť správnosť neformálnej špecifikácie.

Význam stavov na obr. 2 je v uvedený v tab. 1.

Prechody medzi stavmi sú iniciované splnením podmienok uvedených pri prechodoch v stavovom diagrame na obr. 2. Po splnení podmienok dochádza ku generovaniu udalostí. Syntax použitá na obr. 2 je daná použitým softvérovým nástrojom (Rhapsody).

2.2 Vytvorenie aplikačného programu safety PLC

Po vytvorení stavového diagramu a otestovaní jeho správnosti možno pristúpiť k tvorbe aplikačného programu pre safety PLC.

Ako prvý krok je nevyhnutné zadefinovať analógie atribútov, udalostí a stavov objektov zo stavového diagramu s premennými v safety PLC. Tieto analógie sú uvedené v tab. 2. pre vstupy funkcie a v tab. 3 pre výstupy funkcie. V prvom stĺpci je názov premennej v stavovom diagrame (obr. 2) a v druhom stĺpci je jej dátový typ. V treťom stĺpci je názov premennej v aplikačnom programe vytváranom na základe stavového diagramu a v štvrtom stĺpci je jej dátový typ.

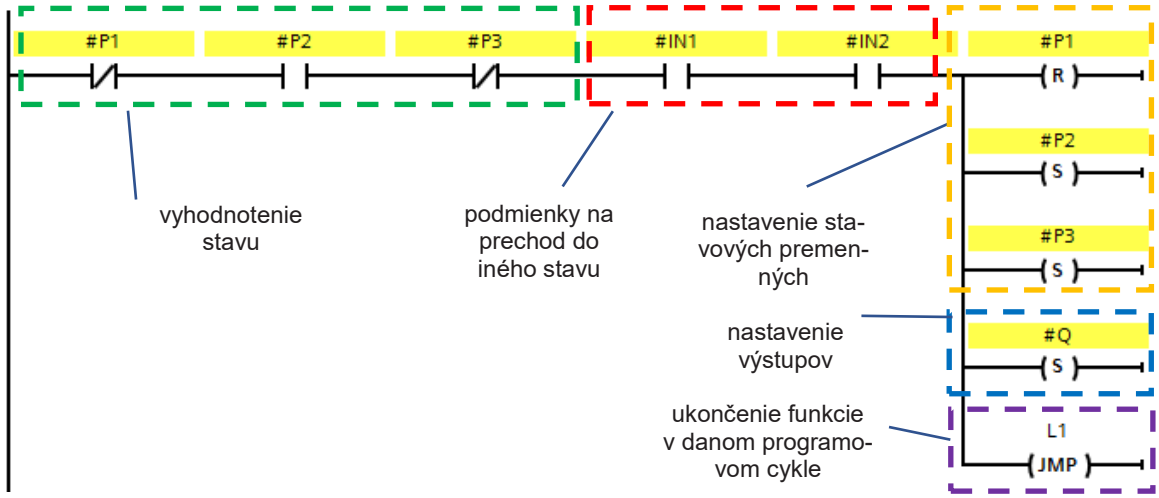
Rhapsody	typ	PLC	typ
avNC_1	bool	IN1	bool
avNC_2	bool	IN2	bool
evACK	event	ACK	bool
avDISCTIME	int	DISCTIME	time

Tab. 2 Analógie atribútov a udalostí zo stavového diagramu s premennými v safety PLC

V tab.3 sú analógie stavov objektov zo stavového diagramu s premennými v safety PLC. Poradie a význam stĺpcov je rovnaký ako v tab.2.

Rhapsody	typ	PLC	typ
stav oQ	objekt	Q	bool
stav oACK_REQ	objekt	ACK_REQ	bool
stav oDISC_FLT	objekt	DISC_FLT	bool

Tab. 3 Analógie stavov objektov zo stavového diagramu s premennými v safety PLC



Obr. 3 Štruktúra priečky safety programu vytvoreného zo stavového diagramu

Fig. 3 Structure of the safety program ladder created from the statechart

Na kódovanie stavov je použitý binárny kód (tab. 4), pretože vykonávanie bitových operácií je podstatne rýchlejšie ako vykonávanie operácií s číslami. Vo všeobecnosti možno použiť akýkoľvek kód, ktorý zaistí jedinečnú identifikáciu každého stavu konečného automatu a zároveň bude tento kód schopný spracovať safety PLC. Vzhľadom na obmedzené dátové typy a obmedzený inštrukčný súbor safety PLC, možno okrem použitia binárneho kódu spravidla ešte uvažovať s použitím dekadického kódu.

P1	P2	P3	Stav
0	0	0	S0
0	0	1	S1
0	1	0	S2
0	1	1	S3
1	0	0	S4
1	0	1	S5

Tab. 4 Kódovanie stavov

Na obr. 3 je štruktúra vybranej priečky z aplikačného programu. Priečka realizuje prechod zo stavu S2 do stavu S3. Prvá časť priečky (označená ako „vyhodnotenie stavu“) zisťuje, či sa safety PLC nachádza v stave S2. Druhá časť priečky (označená ako „podmienky na prechod do iného stavu“) zisťuje, či sú splnené podmienky na prechod do stavu S3. Tretia časť priečky (označená ako „nastavenie stavových premenných“) slúži na zapísanie stavových premenných, ktoré určujú stav S3. Štvrtá časť priečky (označená ako „nastavenie výstupov“) nastaví výstup *Q* na hodnotu true (táto hodnota zodpovedá akcii, ktorá sa vykoná pri prechode zo stavu S2 do stavu S3 v stavovom diagrame na obr. 2). Posledná časť priečky (inštrukcia *JMP*) slúži na ukončenie funkcie v danom programovom cykle, čo zaistí vykonanie maximálne jedného prechodu medzi stavmi počas jedného programového cyklu (to je nevyhnutné kvôli korektnému nastavovaniu výstupov safety PLC).

Na obr. 3 je znázornená len jedna priečka safety programu. Priečku s takouto štruktúrou je nevyhnutné vytvoriť pre každý prechod v stavovom diagrame, pričom použitie inštrukcií a premenných je analogické s uvedenou priečkou.

Prepis zo stavového diagramu na aplikačný program bol realizovaný manuálne, ale postup je algoritmizovateľný a dá sa automatizovať.

Záver

Štruktúra programu realizujúceho bezpečnostné funkcie sa musí vyznačovať jednoduchosťou a prehľadnosťou, aby sa v procese verifikácie a validácie dala overiť jeho správna funkcia. Jednou z možností ako dosiahnuť program s takýmito vlastnosťami je použiť na jeho tvorbu systematický postup založený na prepise stavového diagramu do programu. V článku je prezentovaný postup, ktorý umožňuje použiť stavový diagram UML vytvorený v softvérovom nástroji Rhapsody. Funkcia naprogramovaná týmto postupom bola verifikovaná komparáciou s pôvodnou certifikovanou funkciou. Uvedený postup možno použiť aj v prípade vytvárania zložitejších funkcií, pričom treba zvoliť vhodnú úroveň dekompozície funkcie na jednoduchšie funkcie. Všeobecne platí, že čím sú základné funkcie jednoduchšie, tým je jednoduchšia ich realizácia, ale na druhej strane je zložitejšie riadenie ich vzájomnej koordinácie. Preto treba dbať na to, aby základné funkcie mali čo najmenší počet vstupov a výstupov, aby rozhrania medzi funkciami boli presne definované a výmena informácií medzi nimi bola čo najmenšia.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] RÁSTOČNÝ, K., ŽDÁNSKY, J.: Availability and safety of typical SRCS architectures with safety PLC. In journal: ATP Journal Plus 2/2013, pp. 82-86, ISSN 1336-5010, 2013.

[2] ŽDÁNSKY, J., RÁSTOČNÝ, K., HRBČEK, J.: Influence of architecture and diagnostic to the safety integrity of SRECS output part, Proceedings of international conference Applied

Electronics, Pilsen, Czech Republic, pp. 297-301, ISBN 978-80-261-0385-1, ISSN 1803-7232, 2015.

[3] ROUSAND, M.: Reliability of Safety-Critical Systems, Theory and Applications, Published by John Wiley & Sons, Hoboken, New Jersey, ISBN 978-1-118-11272-4, 2014.

[4] HE, N., OKE, V., ALLEN, G.: Model-based Verification of PLC programs using Simulink Design, International Conference on Electro Information Technology, Dakota, Grand Forks, p. 211-216, ISBN 978-1-4673-9985-2, 2016.

[5] DARVAS D., MAJZIK I., BLANCO VIÑUELA E.: Formal Verification of Safety PLC Based Control Software. Integrated Formal Methods. Lecture Notes in Computer Science, vol 9681. Springer, pp. 508-522, ISSN 0302-9743, 2016.

[6] BIALLAS, S., KAMIN, V., KOWALEWSKI, S., SCHLICH, B., SEHESTEDT, S., STATTELMANN, S.: Verification of Safety-Critical PLC Programs using Safety Automata, 14th Branch Meeting of Measurement and Automation Technology - Automation, Baden, Germany, p. 75-79, ISBN 978-3-18-092209-6, ISSN 0083-5560, 2013.

[7] OVATMAN, T., ARAL, A., POLAT, D., OSMAN ÜNVER, A.: An overview of model checking practices on verification of PLC software. Software & Systems Modeling, Springer, pp. 937-960, ISSN 1619-1366, 2016.

[8] LAMPERIERE-COUFFIN, S., LESAGE, J.-J.: Formal Verification of the Sequential Part of PLC, In Discrete Event Systems, pp. 247-254, ISSN 0893-3405, 2000.

[9] Dostupné na internete: https://www.uml.org/.

Abstract

The paper deals with a systematic approach to the creation of a program for safety PLCs (Programmable Logic Controllers) based on the description of the required function by the UML (Unified Modeling Language) state diagram. Such a procedure can be used to achieve systematic safety integrity of the control system with safety PLCs. The Rhapsody tool is used as UML software support and the application example is implemented on the Simatic safety PLC.

Ing. Milan Medvedík

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univezitná 8215/1
010 26 Žilina
Tel.: +421 41 513 3306
E-mail: milan.medvedik@fel.uniza.sk

doc. Ing. Juraj Ždánsky, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univezitná 8215/1
010 26 Žilina
Tel.: +421 41 513 3342
E-mail: juraj.zdansky@fel.uniza.sk

Obsah

PLATFORMA IZOT A RIADENIE BEZPEČNOSTNE KRITICKÝCH PROCESOV

Tomáš Panáč, Radovan Svítek, Juraj Spalek

Abstrakt

Príspevok je venovaný aplikačným vlastnostiam platformy IzoT™ ako technologickej súčasti konceptu IIoT. Poukazuje na jeho rizikové atribúty a možné hrozby pri používaní komunikačných protokolov najmä z hľadiska bezpečnosti. V závere sú odporúčania pre praktické použitie tejto platformy v riadení bezpečnostne kritických procesov v rámci IIoT.

Kľúčové slová: platforma IzoT, bezpečnostne kritický proces, priemyselný internet vecí IIoT, analýza rizík, zabezpečenie.

Úvod

Platformu IzoT™ pre Industrial Internet of Things (IIoT) vyvinula americká spoločnosť Echelon v roku 2013. Vznikla na báze pôvodnej technológie LonWorks, ktorá je na trhu v množstve priemyselných aplikácií už od konca 20. storočia. Ide o rodinu čipov, zásobníkov, rozhraní a riadiaceho softvéru, ktorý umožňuje vývoj zariadení pre IIoT [0].

Technológia LonWorks vznikla z dôvodu lepšieho cenového riešenia pre inteligentné systémy, ktorá umožňuje integráciu zariadení a prístrojov od rôznych výrobcov a obslúži až 32 tisíc zariadení (uzlov) v sieti. Technológia LonWorks sa vyznačuje jednoduchou integráciou a adaptáciou na rôzne typy systémov, znížili sa týmto jednak finančné náklady, čas potrebný na vývoj a testovanie nového riešenia, ako aj čas na zavedenie systému do prevádzky. Z tohto dôvodu firma nebola nútená zaoberať sa vývojom nového systému a na báze staršej technológie LonWorks bola vytvorená nová platforma pre priemyselne účely - platforma IzoT [1].

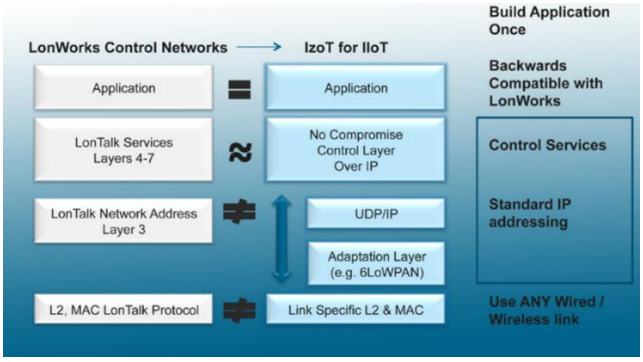
Možno povedať, že spoločnosť Echelon Corporation bola priekopníkom vo vývoji otvorených štandardných riadiacich sieťových platforiem, priniesla všetky prvky potrebné pre návrh, monitorovanie, inštaláciu a riadenie priemyselných "komunít zariadení" v rámci automatizácie budov, osvetlenia a iných trhov po celom svete. Echelon vyvíjala a predávala kompletné systémy a podsystémy pre cieľové aplikácie, integrovaný softvér, plus System-on-chips (SoC) a uvedenie do prevádzky a nástroje pre správu pre OEM. Dnes je na svete nainštalovaných viac než 100 miliónov zariadení, ktoré spoločnosť Echelon vyvinula a nainštalovala [1] [2].

V roku 2018 firmu Echelon odkúpila za 45 miliónov dolárov americká spoločnosť Adesto Technologies so sídlom v Santa Clare v Kalifornii. Všetky vytvorené aplikácie a platformy, teraz spadajú pod túto spoločnosť, ktorá je popredným poskytovateľom inovatívnych polovodičov a vstavaných systémov pre aplikácie, tvoriace bázu IoT. Jej široké portfólio polovodičových a vstavaných technológií je optimalizované pre pripojené zariadenia IoT v priemyselných, spotrebiteľských, komunikačných a lekárskych aplikáciách [3].

Hlavným cieľom príspevku je zistiť a zdôvodniť, či je platforma IzoT vhodná na riadenie bezpečnostne kritických procesov. Spracovaním analýzy rizík v zmysle safety a security dokážeme určiť, či je na to vhodná, prípadne aké má nedostatky a na základe týchto informácií formulovať odporúčania na zlepšenie bezpečnosti pre reálne aplikácie do IIoT.

1. Platforma IzoT

Platforma IzoT patrí do rodiny čipov, rozhraní a softvérových nástrojov pre správu, ktoré komunikujú pomocou Internet Protokolu (IP) a umožňujú vývoj zariadení pre IIoT. Delíme ich na kategórie:



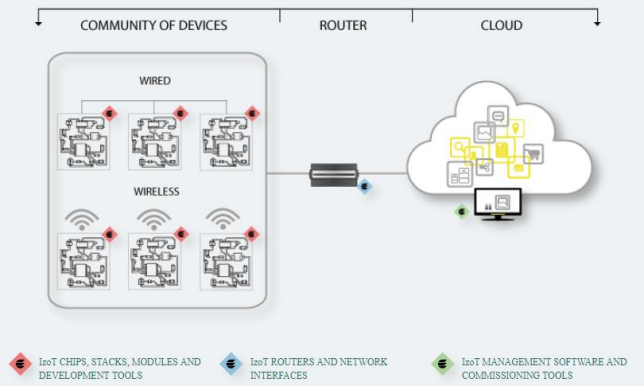
Obr. 1 IzoT ako nadstavba platformy LonWorks [4]

Fig. 1 IzoT as the LonWorks platform extension [4]

Na obr. 1 je schéma, akou Echelon opisuje funkčnosť vrstiev LonTalk 4-7 (ISO 14908-1) na vrchole UDP/IP vrstvy, ktorá umožňuje všetkým známym LonTalk službám, aby boli ponúkané prirodzene cez IP. Tento softvér beží na ľubovoľnom procesore v prostredí, ktoré poskytuje spojenie UDP/IP soketov a je používaný s akoukoľvek podvrstvou, ktorá podporuje takéto komunikačné spojenie. V dostupnom sortimente IzoT produktov sa prelínajú nové a staré technológie. Veľké množstvo priemyselných zariadení, používaných na úrovni pre-

vádzkových technológií, môže pomocou tejto platformy fungovať v moderných IT prostrediach. Medzi produkty tejto platformy patria (obr. 2) [5]:

- skupina zariadení (čipy, zásobníky/stacks, moduly a vývojové nástroje)
- router (routre a sieťové rozhrania)
- cloud (software pre správu a nástroje na uvedenie do prevádzky)



Obr. 2 Všeobecný náhľad zariadení v IzoT platforme [6]

Fig. 2 General view of devices in the IzoT platform [6]

Všetky dostupné komponenty platformy IzoT sú členené do troch podskupín na základe toho, v akom uzle pracujú. Existujú tri podskupiny platformy [3]:

- IzoT moduly, čipy a vývojove nástroje (tab.1);
- IzoT riadiace softvéry a komunikačné nástroje ako súčasť Cloudu (tab. 2);
- IzoT routere a sieťové rozhrania (tab. 3).

Zariadenie	Popis
CPM 4200 Wi-Fi Modul	vytvára komunikáciu bezdrôtových zariadení pre IIoT.
CPM 4200 Wi-Fi EVK	kompletná hardvérová a softvérová platforma na vytváranie alebo vyhodnocovanie bezdrôtových snímačov, radičov a akčných členov.
FT 6010/6050 Smart Transceivers	modernizácia a konsolidácia inteligentných kontrolných sietí; je to kľúčový produkt v platforme IzoT; podpora pre multiprotokolové čipy (podpora LonTalk aj BACnet protokolov).
Neuron 6050 Procesor	zahŕňa funkcie komunikácie a riadenia na jednom čipe ako v hardvéri, tak aj vo firmvéri, aby sa uľahčila konštrukcia LonTalk a BACnet zariadení.
IzoT FT 6000 EVK	kompletná hardvérová a softvérová platforma na vytváranie alebo vyhodnocovanie zariadení založených na transcieveroch a procesoroch zo série 6000.
IzoT SDK 2	softvérový balíček, ktorý umožňuje vývojárom vytvárať komunikačné zariadenia pre IIoT. IzoT SDK tiež umožňuje vývojárom vytvoriť webový aplikačný server pre siete IzoT.

Tab. 1 IzoT moduly, čipy a vývojove nástroje [6]

Zariadenie	Popis
IzoT Router 2	zariadenie na prepojenie zariadení LonTalk/IP a LON na ethernetovom kanáli so zariadeniami LonTalk/IP a LON na kanáloch FT alebo RS-485 a na prepojenie webových stránok a podnikových aplikácií k zariadeniam LonTalk/IP a LON.
IzoT U60 FT DIN USB	kompaktné USB sieťové rozhranie, ktoré slúži na pripojenie k hostiteľskému počítaču, radiču alebo na pripojenie k routeru IzoT.
IzoT U60 FT USB Modul	doskový modul s rozhraním USB, ktorý sa dá ľahko integrovať do ľubovoľného radiča alebo zariadenia s LonTalk/IP a LON FT zariadením s krútenou dvojlinkou; rovnako slúži na pripojenie k hostiteľskému počítaču, radiču alebo na pripojenie k routeru.

Tab. 1 IzoT riadiace softvéry a komunikačné nástroje [6]

Zariadenie	Popis
IzoT Router 2	zariadenie na prepojenie zariadení LonTalk/IP a LON na ethernetovom kanáli so zariadeniami LonTalk/IP a LON na kanáloch FT alebo RS-485 a na prepojenie webových stránok a podnikových aplikácií k zariadeniam LonTalk/IP a LON.
IzoT U60 FT DIN USB	kompaktné USB sieťové rozhranie, ktoré slúži na pripojenie k hostiteľskému počítaču, radiču alebo na pripojenie k routeru IzoT.
IzoT U60 FT USB Modul	doskový modul s rozhraním USB, ktorý sa dá ľahko integrovať do ľubovoľného radiča alebo zariadenia s LonTalk/IP a LON FT zariadením s krútenou dvojlinkou; rovnako slúži na pripojenie k hostiteľskému počítaču, radiču alebo na pripojenie k routeru.

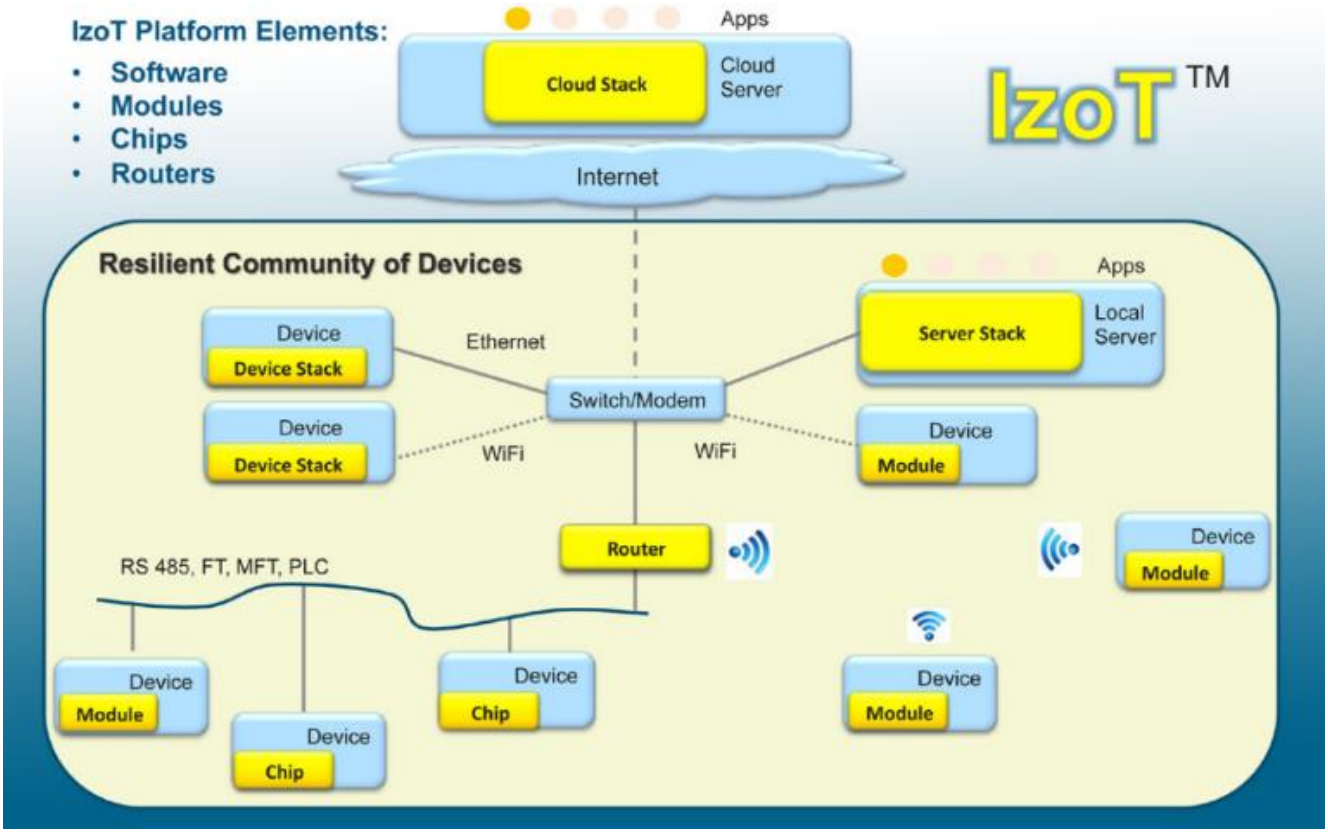
Tab. 3 IzoT routere a sieťové rozhrania [6]

2. Komunikácia v platforme IzoT

Platforma IzoT zahŕňa kompletné vývojové prostredie, sady protokolov, routery, čipy, nástroje pre uvedenie do prevádzky a moduly pre inžinierov, ktorí vytvárajú riadiace aplikácie a zariadenia pre IIoT [7]. Na obr. 3 je skupina zariadení typu peer-to-peer, ktorá využíva rôzne komunikačné možnosti založené na protokole IP, ako je Wi-Fi, RS-485, FT a Ethernet.

2.1 Multiprotokolová komunikácia

Echelon vylepšila svoje čipy tak, aby podporovali protokoly LonTalk/FT, LonTalk/IP, BACnet/IP a BACnet / MS-TP. Obr. 4 ilustruje, ako zariadenie vyvinuté pomocou čipu IPnabled FT 6000 môže pracovať ako zariadenie LonWorks, ako aj zariadenie BACnet s rovnakou aplikáciou. Aplikácia nad rámec služieb ISO 14908-1 L4-L6 sa môže prezentovať ako tradičný uzol LONWORKS (pomocou adresovania LonTalk ISO 14908-1 L2-L3) alebo ako uzol LON-over-IP (pomocou IP adresy namiesto 14908 -1 adresovanie L2-L3). Navyše tá istá aplikácia sa môže prezentovať ako uzol "BACnet-overIP" kvôli unikátnej mapovej vrstve BACnet, ktorá je zabudovaná do FT 6000. Výhodou je, že BACnet / IP je dostupný cez kanál FT, ktorý je ďaleko lepší ako menej spoľahlivý kanál RS-485, ako aj nižšie náklady a flexibilnejšie ako používanie siete Ethernet pre BACnet / IP [8].

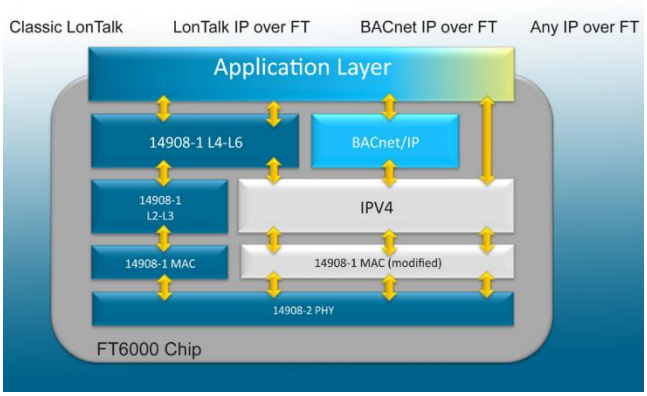


Obr. 3 Spôsoby komunikácie platformy IzoT [4]

Fig. 3 Modes of communication of IzoT platform [4]

Tým, že Echelon, resp. nástupnícka spoločnosť Adesto Technologies poskytuje spoločnú platformu pre vývoj multiprotokolových, peer-to-peer, bezdrôtových a káblových riadiacich produktov, vie platforma IzoT [5]:

- umožniť vývojárom vytvárať citlivé, spoľahlivé a škálovateľné riadiace zariadenia, ktoré sú IP-enable
- znížiť náklady na inštaláciu, údržbu a prevádzku pre vlastníkov budov
- nižšie náklady na vedu a výskum (R&D) pre OEM (Original Equipment Manufacter = pôvodný výrobca zariadenia)
- pomôcť OEM rozšíriť svoje adresovateľné trhy s použitím menšieho počtu skladových jednotiek produktu
- umožniť ovládanie sietí, ktoré majú byť použité na nové aktíva, ktoré predtým buď nepoužívali ovládanie sietí alebo používali patentované riadiace technológie.



Obr. 4 Multiprotokolová komunikácia čipu FT6000 [5]

Fig. 4 Multiprotocol communication of FT6000 chip [5]

3. Cloud Computing a bezpečnosť

Pripomeňme, že informačná bezpečnosť je komplexný prístup k ochrane informácií ako celku. Preto je dôležité chrániť informácie vo všetkých ich formách a počas ich životného cyklu, to znamená počas ich tvorby, zberu, spracovania, úschovy, prenosu a likvidácie. Pre účinnú ochranu treba určiť, aké informácie organizácia má a akú majú hodnotu. Čím citlivejšie dáta, tým je implementácia riadenia a kontroly informačnej bezpečnosti na úrovni Cloudu žiadanejšia, hlavne ak sa jedná o informácie, ktorých prípadná strata alebo odcudzenie by mohlo spôsobiť kritické následky. Treba však poznamenať, že cieľom nie je iba implementácia, ale aj ďalší dlhodobý vývoj a funkčnosť systému v reakcii na zmeny v organizácii a jej prostredí [9].

„Cloud computing je metóda prístupu k používaniu počítačovej technológie, ktorá je založená na poskytovaní zdieľaných výpočtových zdrojov a ich využívaní ako služba.“ Existujú rôzne modely služieb a možnosti dodávania, ale všetky typy cloud computingu sú schopné poskytovať zdieľané zdroje na požiadanie. Sú to elastické, samoobslužné služby s rozsiahlou štruktúrou prístupu k dátovým zdrojom [9].

Cloud computing je marketingový výraz pre webové aplikácie, úložné a komunikačné služby. V cloud computing dátové centrum uchováva informácie, ktoré tradične uložili koncoví používatelia na svojich počítačoch. To vyvoláva obavy v súvislosti s ochranou súkromia používateľov, pretože používatelia musia zadávať svoje údaje. Okrem toho prechod na centralizované služby by mohol ovplyvniť súkromie a bezpečnosť používateľských interakcií. Bezpečnostné hrozby sa môžu vyskytnúť pri zabezpečení zdrojov a počas distribuovaného nasadzovania aplikácií. Vznikajú aj nové hrozby. Hackeri

môžu napríklad použiť virtualizovanú infraštruktúru ako spúšťač nových útokov. Cloud by mal udržiavať integritu údajov a súkromie používateľov. V tejto súvislosti je potrebné preskúmať nové mechanizmy ochrany údajov s cieľom zabezpečiť ochranu osobných údajov, bezpečnosť zdrojov a autorské práva [9].

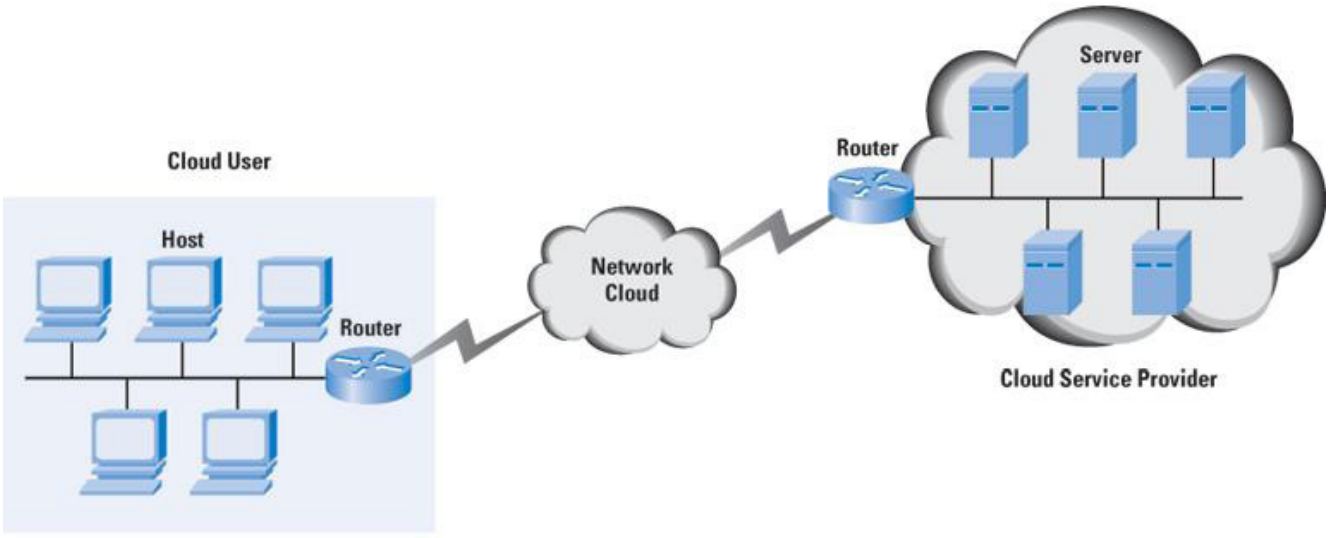
3.1 Interný Cloud

Interný cloud je v mnohých ohľadoch najbežnejší typ cloud computingu. Vnútorný cloud sa vyskytuje v rámci jedinej organizácie a umožňuje im implementovať virtualizáciu pre interné služby. Predpokladom je, že vnútorná infraštruktúra vrátane serverov, sietí, úložných zariadení a aplikácií bude prepojená a virtualizovaná, čo jej umožní dynamicky pracovať tak, aby sa dosiahla maximálna účinnosť. Toto sa odlišuje od jednoducho virtualizovanej situácie v tom, že umožňuje vyšší stupeň automatizácie [9]. Interný cloud môže byť tiež známy ako firemný cloud, pretože je v tejto sfére veľmi využívaný. Pre korporáciu pozostávajúcu z viacerých spoločností cloud platforma umožňuje minimalizovať náklady na IT podporu viacerých inštancií tej istej aplikácie. Napríklad, ak každá spoločnosť používa rovnaký inventárny softvér, namiesto vytvorenia viacerých inštancií môže byť použitá jedna viacnásobná inštancia.

Organizácia vytvára interný cloud tým, že uplatňuje model služby cloud computing a rámec poskytovania informácií prostredníctvom miestnych alebo externých zdrojov dátových centier. Vnútorný cloud potom poskytuje každému uzlu v rámci organizácie výpočtové, ukladacie a softvérové služby. Interný cloud poskytuje nasledujúce výhody [9]:

- zníženie nákladov pri správe viacerých inštancií tej istej aplikácie (minimalizácia nákladov na infraštruktúru);
- rýchle a súčasné aktualizácie aplikácií pre všetky spoločnosti;
- znížené hardvérové požiadavky;
- celková bezpečnosť cloudu (alebo aspoň kontrola zabezpečenia).

Tento koncept je veľmi podobný privátnemu cloudu a často sa tieto pojmy zamieňajú, pretože technológie cloud computing sa používajú pre jednu organizáciu. Rozdiel je v tom, že privátny cloud by mohol odkazovať aj na vyhradené zdroje u poskytovateľa služieb tretej strany, kde interný cloud odkazuje na využitie vlastnej infraštruktúry.



Obr. 5 Princíp externého cloudu [9]

Fig. 5 The principle of an external cloud [9]

3.2 Externý Cloud

- Tento typ modelu využíva externú službu prostredníctvom poskytovateľa cloudu a jeho prístup je sprostredkovaný cez internet. Služby ponúkané týmto typom cloudu sú dostupné najširšiemu okruhu zákazníkov, teda hovoríme hlavne o širokej verejnosti. Výpočtová infraštruktúra externého cloudu je vlastnená jeho poskytovateľom, zákazník k poskytovaným službám pristupuje vzdialene po sieti prostredníctvom klientskeho rozhrania. Práve podľa toho akým spôsobom sú služby poskytované koncovým užívateľom môžeme rozdeliť externé cloudy na [9]:
- verejný cloud - je k dispozícii širokej verejnosti alebo veľkej priemyselnej skupine a je vo vlastníctve organizácie, ktorá poskytuje cloudové služby; prostriedky sú poskytované od poskytovateľa tretej strany, ktorý zdieľa zdroje; poskytovateľ je schopný komukoľvek vytvoriť virtuálny cloud, kde je teda určitá množina oddelená nie fyzicky ale logicky; príkladmi verejného cloudu môžu byť napr. „Amazon Simple Storage Service“, „Google App Engine“ alebo „Microsoft Azure“;
- súkromný cloud - rieši niektoré problémy a riziká, ktoré vyplývajú z verejného cloudu; pre zákazníka to môže byť napr. znížená schopnosť rozhodovať o umiestnení dát, nutnosť zdieľať cloudovú infraštruktúru a riziká vyplývajúce z požiadaviek na dostupnosť, dôvernosť a bezpečnosť dát; významný poskytovateľ privátneho cloudu je napr. firma IBM pod označením „IBM SmartCloud Foundation“.
- hybridný cloud – je spojením modelu verejného a súkromného cloudu; tento typ cloudu umožňuje väčším firmám rozdeliť výpočtové prostriedky a dáta, s ktorými pracujú do dvoch skupín:
- využívanie služieb verejného cloudu širokou verejnosťou;
- využívanie služieb súkromného cloudu spoločnosťou.

Prostredie pozostáva z viacerých interných alebo externých poskytovateľov. Preto tento typ cloudu poskytuje najvyššiu úroveň flexibility.

Jednou z hlavných výhod verejných cloudových služieb je do určitej miery jednoduchosť škálovania. Pre malé a stredné podniky vo všeobecnosti je škálovateľnosť platená. Zdroje sú v podstate ponúkané „na požiadanie“, takže akékoľvek

zmeny úrovne aktivity je možné spracovávať veľmi ľahko. To zase prináša nákladovú efektívnosť. Obrovská sieť serverov zapojených do verejných cloudových služieb znamená, že môže profitovať z väčšej spoľahlivosti. Aj keby jedno dátové centrum úplne zlyhalo, sieť jednoducho prerozdeľuje záťaž medzi ostatné centrá, čo je veľmi nepravdepodobné, že verejný cloud niekedy zlyhá. Stručne povedané, výhody verejného cloudu sú:

- škálovateľnosť podľa platobnej doby;
- efektivita nákladov;
- zvýšená spoľahlivosť.

Existujú samozrejme nedostatky v používaní verejných cloudových služieb. Na začiatku zoznamu je skutočnosť, že bezpečnosť údajov v rámci verejného cloudu je dôvodom na obavy. Často sa považuje za výhodu, že verejný cloud nemá žiadne geografické obmedzenia, čo zjednoduší prístup bez ohľadu na to, kde sa nachádzate. Ale na druhej strane by to mohlo znamenať, že váš server je v inej krajine, ktorú riadi úplne iný súbor bezpečnostných a súkromných predpisov. Výkon môže byť tiež problémom. S geografiou súvisí aj doba odozvy, ktorá môže byť pre niektoré aplikácie kľúčová. Prenos dát môže byť ovplyvnený zvýšenou prevádzkou používateľov na internete.

3.3 Riziká spojené s Cloud Computingom

Bezpečnostné problémy súvisiace s cloud computingom spadajú do dvoch vymedzených kategórií: bezpečnostné problémy, s ktorými sa stretávajú poskytovatelia cloudu (organizácie poskytujúce softvér, platformu alebo infraštruktúru) a bezpečnostné problémy, ktorým čelia ich zákazníci (spoločnosti alebo organizácie, ktoré hosťujú aplikácie alebo ukladajú údaje v cloude). Jedná sa však o zdieľanú zodpovednosť. Poskytovateľ musí zabezpečiť bezpečnosť svojej infraštruktúry a ochranu údajov a aplikácií svojich klientov, zatiaľ čo používateľ musí prijať opatrenia na posilnenie svojej aplikácie a používať silné heslá a autentifikačné opatrenia [9]. Ak sa organizácia odhodlá ukladať svoje údaje alebo hostiť aplikácie vo verejnom cloude, stratí schopnosť mať fyzický prístup k serverom, kde uložili svoje dáta. Týmto pádom sa stávajú potenciálne citlivé údaje terčom útokov z vnútra. Poskytovatelia cloudu preto musia zabezpečiť, aby dátové centrá boli často sledované kvôli podozrivej aktivite a tiež by mali zabezpečiť správnu izoláciu údajov a segregáciu logického ukladania [9].

Riziká, ktoré sú spojené s cloud computing možno zaradiť do piatich hlavných skupín rizík:

- spojené s riadením
- spojené s manažmentom bezpečnosti
- spojené s dátami
- spojené so spoľahlivosťou
- spojené so schopnosťou zabezpečiť súlad v jurisdikcii používateľov

4. Rizikové atribúty platformy IzoT

Ako bolo uvedené, hlavná výhoda platformy IzoT je jej otvorenosť. Podporuje viacero komunikačných protokolov a zjednocuje staršie a aj nové zariadenia v sieti. Ak prenos bezpečnostne relevantných správ používajú bezpečný účastníci komunikácie, ale využíva sa nedôveryhodný prenosový systém, musíme rátať s určitými hrozbami, ktorými sú prenášané správy vystavené.

Za otvorený komunikačný systém sa považuje taký, ktorý neplní čo i len jednu podmienku uzavretosti systému. Čiže taký systém, v ktorom platí aspoň jeden z nasledujúcich bodov:

- pripojenie neoprávneného účastníka nie je vylúčené
- systémy riadenia siete môžu smerovať (a dynamicky presmerovať) tok prenášaných dát akoukoľvek cestou medzi koncami prenosového systému v súlade s programom, ktorý nie je používateľovi známy
- všetkých oprávnených účastníkov komunikácie nemožno explicitne vymenovať
- fyzikálne vlastnosti prenosových médií (vrátane ich prenosových charakteristík a odolnosti proti vonkajším vplyvom) nie sú používateľovi známe
- všetky vlastnosti prenosového systému nie sú používateľovi známe (jednotlivé elementy prenosového zariadenia môžu prenášané dáta čítať, ukladať do pamäti, spracovávať, zoskupovať, znovu prenášať a podobne, podľa programu, ktorý nie je používateľovi známy)

Z týchto uvedených bodov vyplývajú všeobecné hrozby, ktoré môžu nastať a integrita prenášanej správy je im vystavená. Okrem uvedených hrozieb sem patria aj takisto nebezpečné udalosti, ktoré môžu v otvorenom prenosovom systéme čiže aj v platforme IzoT vzniknúť:

- **neúmyselné** nebezpečné udalosti - patria sem napríklad elektromagnetické rušenie(EMI), blesk, ľudská chyba, oheň, prerušenie vodičov a ďalšie iné.
- **úmyselné** nebezpečné udalosti - patrí sem, úmyselné monitorovanie prevádzky, odpočúvanie, vloženie dát, zmena dát a vymazanie dát neautorizovaným subjektom. Ide predovšetkým o útoky sústredené na prenosový systém.

Na obr. 6 je štruktúra bezpečnostne relevantného prenosového systému, medzi aplikáciami, ktorý využíva otvorený komunikačný systém. Podľa správy o priemyselnom zlyhaní údajov spoločnosti Verizon za rok 2018 spôsobili útočníci sponzorovaní štátmi viac ako polovicu porušení údajov v priemysle. Spolu s takými útokmi správa Verizon odhalila, že cyberattack je hlavným motívom týchto porušení. V tejto správe kalifornská kybernetická spoločnosť Vectra, ktorá sa venuje detekcii skrytých kybernetických útokov a napomáha lovcom hrozieb, ako aj zvyšuje efektívnosť vyšetrovania incidentov, odhaľuje, že útočníci, ktorí sa vyhnú obvodovej bezpečnosti, môžu ľahko špehovať, šíriť a ukradnúť. Výrobný priemysel vykazuje vyššiu ako normálnu mieru prieskumu a bočnej pohybovej aktivity súvisiacej s kyberútokom [11].

5. Odporúčania pre praktické použitie platformy IzoT v IIoT

IzoT je najflexibilnejšou voľbou pre implementáciu siete zariadení v oblasti priemyselného internetu vecí, pretože niektoré jej vlastnosti sú na trhu jedinečné. Medzi tieto vlastnosti patrí [9]:

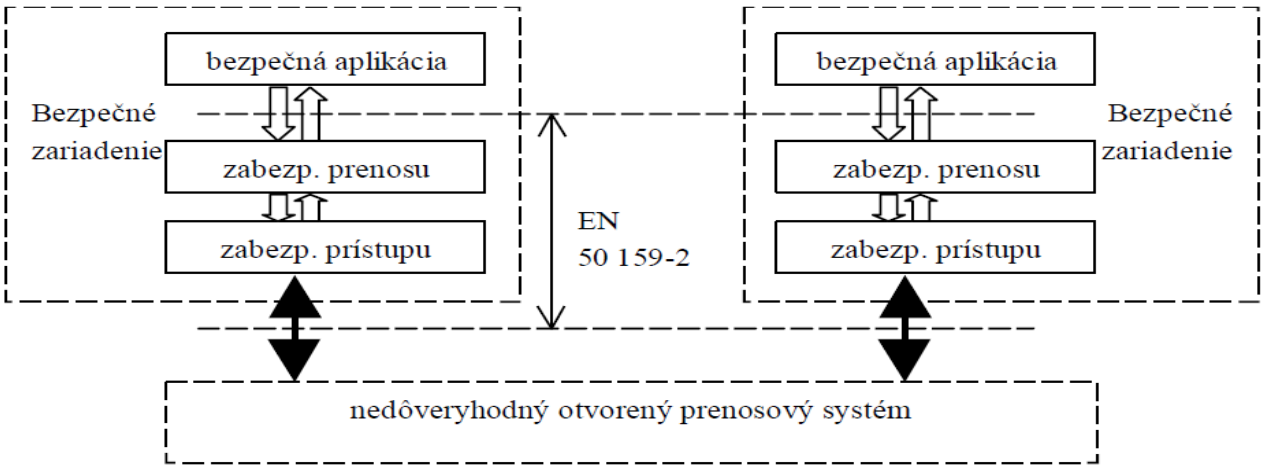
- zjednotenie viacero komunikačných liniek pod bežný aplikačný model;
- poskytuje osvedčený model interoperability pre aplikácie;
- transparentná podpora starších sieti ISO/IEC 14908;
- môže pracovať s bežne dostupnými UDP rozhraniami pre nízkonákladové zariadenia.

Avšak napriek veľkým výhodám IzoT nepredstavuje optimálne riešenie v oblasti riadenia bezpečnostne kritických procesov vďaka hrozbám, ktoré boli už spomenuté. Medzi osvedčené techniky a postupy, ktoré môžu eliminovať resp. aspoň minimalizovať tieto hrozby, patria:

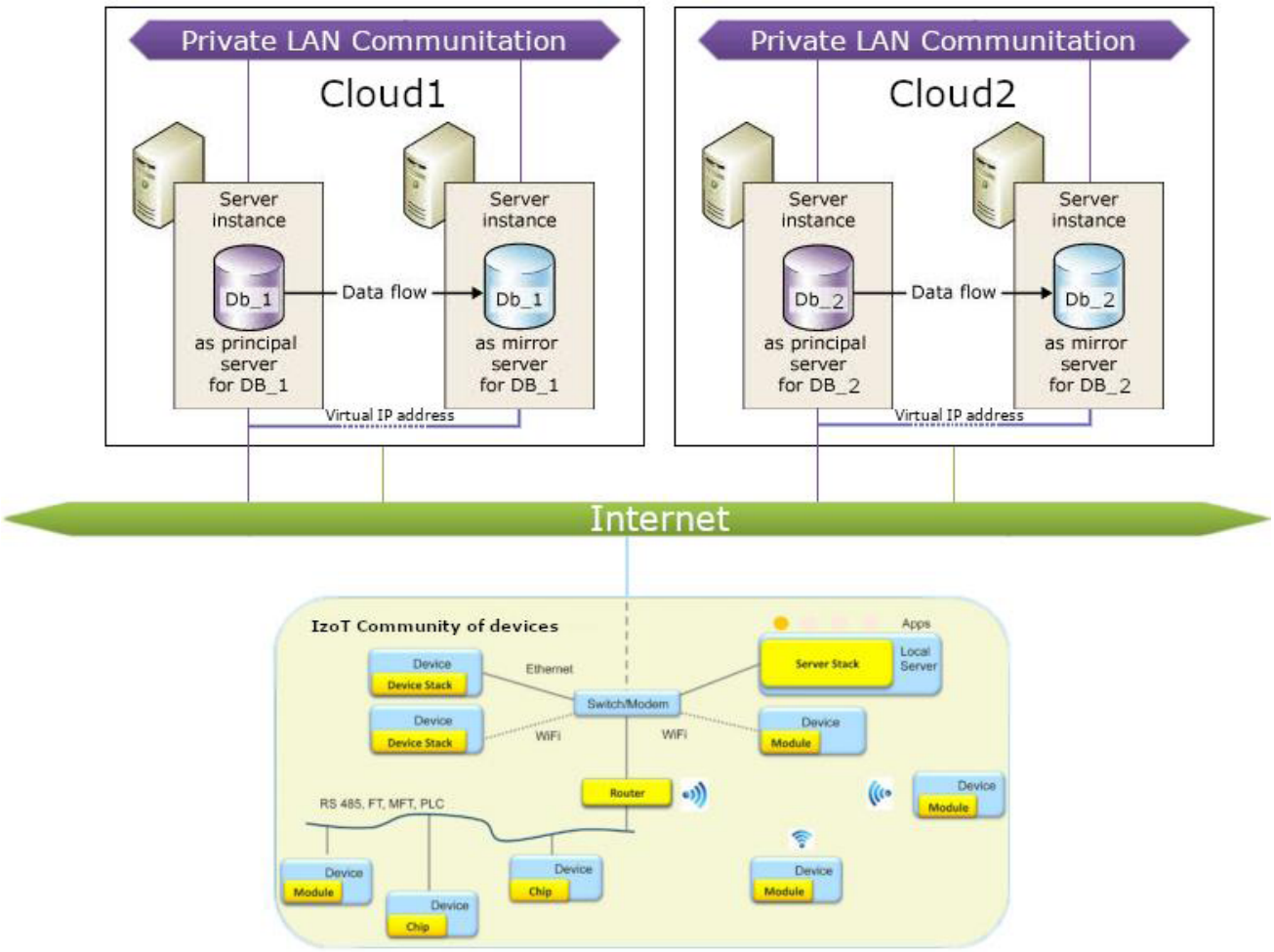
- použitie viackanálových systémov
- použitie optickej kabeláže pre prenos dát
- použitie viac nezávislých cloudových úložísk (obr. 6)

Medzi najväčšie výhody použitia viacerých nezávislých cloudových úložísk patrí väčšia miera kontroly, spoľahlivosti a vyriešenie pochybnosti a rizík ohľadom (ne)dostupnosti dát. Dôležité je taktiež vykonávať povinnú starostlivosť u všetkých

redundantných úložísk. Tak sa stanú bezpečnostné požiadavky ešte náročnejšími, ak na server ukladáme citlivé dáta, resp. dáta z kritického riadenia, čo vyžaduje ich zabezpečenie a ochranu pred útočníkmi



Obr. 6 Štruktúra bezpečnostne relevantného prenosového systému [10]
Fig. 6 Structure of safety relevant transmission system [10]



Obr. 7 Princíp dvoch nezávislých cloudových úložísk [9]
Fig. 7 The principle of two independent cloud storage [9]

Záver

Cieľom príspevku bolo ukázať, či je platforma IzoT vhodná na riadenie bezpečnostne kritických procesov a to najmä z hľadiska bezpečnosti protokolov, ktoré sú v platforme IzoT použité. Jednalo sa o protokoly LonTalk, ktorý platforma IzoT prebrala zo staršej platformy LonWorks a protokol BACnet, ktorý bol k novej platforme integrovaný, čím sa vytvorila nová multiprotokolová platforma. Z rozsiahlejšej analýzy rizík vyplynulo, že kryptografické algoritmy, ktoré sa používajú v protokole LonTalk sú veľmi zastarané a nepostačujú na to, aby dokázali zabezpečiť dostatočnú ochranu na zaistenie autenticity, súkromia a integrity dát. Protokol využíva algoritmus RC4, ktorý bol v minulosti niekoľko krát prelomený a veľmi malé dĺžky súkromných kľúčov, ktoré v dnešnej dobe už nepostačujú.

Protokol BACnet sa po úprave špecifikácie z roku 2006 ukázal oveľa bezpečnejší ako protokol LonTalk, čo vyplýva z použitia zabezpečených zariadení a routerov a taktiež z implementácie lepších kryptografických algoritmov. Protokol využíva na zabezpečenie dôvernosti dát algoritmus AES s kľúčom dĺžky 128 bitov, ktorého výpočtová bezpečnosť je v dnešnej dobe relatívne bezpečná, ale tým že sa výpočtová rýchlosť zariadení neustále zvyšuje môže byť bezpečnosť tohto algoritmu onedlho veľmi otázna [9].

Okrem toho sa ukázalo, že interné (firemné) cloudové úložiská sa javia ako najbezpečnejšie. Ale keďže väčšina aplikácií pre riadenie bezpečnostne kritických procesov sa nachádza mimo interných oblastí firmy alebo podniku, je zrejmé, že platforma by musela komunikovať s externým cloudom, čo sa zatiaľ vôbec nejaví ako bezpečné a spoľahlivé riešenie. Medzi hlavné riziká externého cloudu patria napr. odmietnutie služby, narušenie integrity a dôvernosti, čo sú hlavné podmienky na to aby takýto systém mohol byť implementovaný.

Keďže ide o otvorený prenosový systém, kde informácie putujú po nezabezpečených sieťach, je zložité a aj drahé riešiť zabezpečenie, hlavne z časového a finančného hľadiska. Avšak existujú opatrenia na zaistenie lepšej bezpečnosti ako napr. využitie VPN, IPsec, Kerberos atď., ale pre riadenie v priemysle a hlavne pre bezpečnostne kritické riadenie je vhodnejšie použiť niektorý z uzatvorený prenosový systémov. Možno o pár rokov budú otvorené prenosové systémy na toľko spoľahlivé, že budú môcť byť použité na bezpečnostne kritické riadenie a konkurovať uzatvoreným systémom, zatiaľ to ale neplatí. IzoT platforma nie je primárne určená na bezpečnostne kritické riadenie, preto ten, kto sa rozhodne využiť platformu IzoT na účely bezpečnostne kritického riadenia, bude zodpovedný za škody v dôsledku zlyhania systému, poškodenia majetku alebo úrazu osôb.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ

Literatúra

[0] Priemyselný ethernet prekonal priemyselné prevádzkové zbernice. ATP Journal/nové trendy, 24.1.2019

[1] IIoT. Platforma IzoT. [Online] [Dátum: 10. 12 2018.] https://www.wired.com/2014/02/spime-watch-echelon-inc-izot-platform-industrial-internet-things/.

[2] ECHELON. Platforma IzoT. [Online] [Dátum: 10. 12 2018.] http://news.echelon.com/press-release/corporate/echelon-expands-its-izot-platform-iiot-enable-multiprotocol-wired-and-wirele.

[3] ADESTO. Adesto Technologies. [Online] [Dátum: 11. 12 2018.] https://www.adestotech.com/about-us/about-adesto/

[4] ECHELON. IIoT. [Online] [Dátum: 13. 12 2018.] http://echelon-la.com/files/M2M-IIoT-Control-networking-2.0-whitepaper.pdf

[5] SVÍTEK, R.: Platforma IzoT pre Industrial Internet of Things. Bakalárska práca. EF UNIZA Žilina, 2017

[6] IzoT Zariadenia. Prevzaté: 02.12.2017Dostupné na: https://www.echelon.com/izot-platform.

[7] ECHELON. Platforma IzoT. [Online] [Dátum: 10. 12 2018.] http://news.echelon.com/press-release/corporate/echelon-expands-its-izot-platform-iiot-enable-multiprotocol-wired-and-wirele

[8] ECHELON. IIoT. [Online] [Dátum: 13. 12 2018.] http://echelon-la.com/files/M2M-IIoT-Control-networking-2.0-whitepaper.pdf

[9] PANÁČ, T.: Je platforma IzoT vhodná na riadenie bezpečnostne kritických procesov? Diplomová práca EF UNIZA, Žilina, 2018

[10] Riziká. Cyber Threat. [Online] [Dátum: 21. 3 2019.] https://ics-cert.us-cert.gov/sites/default/files/ICSJWG-Archive/QNL_JUN_17/IIOT_and_the_Cyber_Threat_S508C.pdf

[11] Riziká. Kyberšpionáž. [Online] [Dátum: 21. 3 2019.] https://www.industryweek.com/technology-and-iiot/industrial-iot-escalates-risk-global-cyberattacks

Abstract

The paper is devoted to application features of the IzoT™ platform as a technological part of the IIoT concept. It points out its risk attributes and possible threats in the use of communication protocols, especially in terms of security. In conclusion, there are recommendations for the practical use of this platform in managing safety critical processes within IIoT

Ing. Tomáš Panáč

Bc. Radovan Svítek

prof. Ing. Juraj Spalek, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných technológií
Ul. Univerzitná 8215/1
010 26 Žilina
Tel: +421 41 5133300
E-mail: juraj.spalek@fel.uniza.sk

Obsah

SIMULÁCIA STRATOSFÉRICKÝCH LETOV BALÓNA S RIADENÝM ZOSTUPOM

Vojtech Šimák, Filip Škultéty, Dušan Nemec, Marián Hruboš, Jozef Hrbček

Abstrakt
V tomto článku sme sa zaoberali matematickou simuláciou stratosférických letov balóna v rámci projektu SALSA. Stratosférický balón má niesť užitočnú záťaž (miniatúrny satelit) za účelom testovania v nepriaznivých podmienkach stratosféry pre overenie jeho funkcií. Let stratosférickým balónom je neporovnateľne lacnejší voči reálnemu umiestneniu satelitu na obežnú dráhu Zeme nosnou raketou.

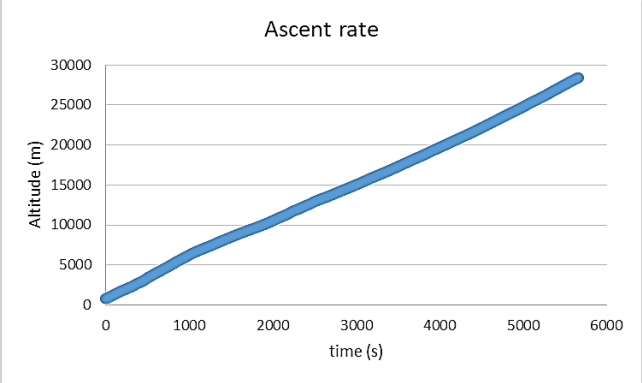
Kľúčové slová: matematický model, balón

Úvod

Projekt SALSA (Stratospheric Autonomous Landing System Application) je riešený pre Európsku Vesmírnu Agentúru (ESA), spoločnosťou Gospace a ich partnermi (vrátane Žilinskej univerzity v Žiline). Cieľom projektu je vyvinúť autonómny pristávací systém pre stratosférický balón (bezmotorový klzák). Tento autonómny systém bude mať za cieľ vyhnúť sa zakázaným zónam. Medzi zakázané zóny patria najmä letiská (civilné aj vojenské), elektrárne, chemické továrne, rafinérie... V prvej fáze je celý systém vynesený do stratosféry héliovým balónom. Táto fáza je najdlhšia v rámci letu a počas tejto fázy majú poveternostné vplyvy najväčší dopad na miesto pristátia. Po fáze stúpania balón buď sám praskne pôsobením síl, alebo je odpojený. Nasleduje fáza stabilizovaného pádu. Počas tejto fázy padá systém len s relatívne malým stabilizačným padákom, ktorý nemá za úlohu brzdiť systém, ale zabezpečiť, aby sa systém počas pádu neotáčal. V určitej výške je aktivovaný mechanizmus otvárania hlavného padáku (riadeného klzáka). Riadený klzák bude mať formu krídlového padáku. V tomto okamihu systém na základe informácií o polohe z GNSS systému, mapy zakázaných zón a smeru vetra dokáže rozhodnúť o mieste pristátia.

1. Prvá fáza: Stúpanie

V tejto fáze héliový balón stúpa pôsobením statickej vztlakovej sily. Táto sila je závislá od objemu, tlaku a teploty hélia. Proti tejto sile pôsobí gravitačná sila daná tiažovým zrýchlením a hmotnosťou všetkých komponentov. V prípade, že balón začne stúpať a pohybuje sa určitou rýchlosťou, pôsobí naň taktiež aerodynamický odpor pohybu. Ten však závisí od rozmerov a tvaru objektu. Balón počas stúpania prechádza vrstvami atmosféry s nízkou teplotou, tie balón ochladzujú a tým klesá objem a vztlaková sila. Proti tomuto javu pôsobí klesajúci tlak vzduchu, ktorý naopak spôsobuje rozpínanie balóna vplyvom vnútorného tlaku. Z uvedeného vyplýva, že sa jedná o veľmi komplexný fyzikálny jav s množstvom parametrov. Preto sme sa pri fáze modelovania zamerali na reálny let stratosférického balóna a snažili sme sa zistiť stúpavosť ako funkciu výšky. Prekvapujúcim zistením bolo, že balón stúpa konštantne od začiatku až po koniec stúpania (obr. 1).



Obr. 1 Graf stúpania héliového balóna v atmosfére
Fig. 1 Ascent of helium balloon in atmosphere

1.2 Matematický model teploty

Teplota je veľmi dôležitý parameter počas stúpania a klesania celého systému. Teplota vystupuje vo funkcii hustoty vzduchu a je zobraziteľná v ľubovoľnom bode letu. Teplotný model atmosféry je nelineárny. Matematický model teploty sa nachádza v [3]. Je tvorený z viacerých lineárnych častí podľa nadmorskej výšky, uvedených v tab. 1

Nadmorská výška v km	Teplotný koeficient v [K/m]	Počiatočná teplota na dolnej hranici vrstvy v [K]
0 - 11	-0,0065	288,15
11 - 20	0	216,65
20 - 32	0,001	216,65
32 - 47	0,0028	228,65
47 - 51	0	270,65

Tab. 1 Teplotný model atmosféry

Z uvedeného vyplýva, že teplota vo vrstvách medzi 11 km a 20 km nadmorskej výšky veľmi nízka (216,65 K je približne 56,5 °C).

1.3 Matematický model vetra

Vietor je kľúčový parameter, od ktorého závisí, kam celý systém poletí. Dáta boli poskytnuté pre Katedru leteckej dopravy Slovenským hydrometeorologickým ústavom. Dáta o vetre sú pre stredné Slovensko a pre letné mesiace (predpoklad je, že väčšina letov bude v lete). Atmosféra je rozdelená na 5 vrstiev a pre každú vrstvu je stredná hodnota a rozptyl rýchlosti vetra a stredná hodnota a rozptyl smeru (azimutu):

- Vrstva od 0 do 2000 m
  2,5 [m.s^{-1}] ± 2,5 [m.s^{-1}] azimut 212° ± 28,5°
- Vrstva od 2000 do 4000m
  10,5 [m.s^{-1}] ± 3,9 [m.s^{-1}] azimut 320° ± 21°
- Vrstva od 4000 do 7500m
  21,4 [m.s^{-1}] ± 5,1 [m.s^{-1}] azimut 258° ± 22,5°
- Vrstva od 7500 do 15000m
  42,4 [m.s^{-1}] ± 10,95 [m.s^{-1}] azimut 243° ± 26,5°
- Vrstva od 15000 do 45000m
  11,4 [m.s^{-1}] ± 2,65 [m.s^{-1}] azimut 96° ± 30°

2. Druhá fáza: voľný pád

Počas tejto fázy už systém neobsahuje héliový balón. Počas prvých cca 40s systém v podstate len zrýchľuje gravitačným zrýchlením. Počas tejto doby klesne systém až o cca 7 km a nadobudne takmer nadzvukovú rýchlosť. V prípade zoskoku Felixa Baumgartnera [1] dokonca nadzvukovú rýchlosť prekonal. Pre teleso padajúce vo výške 30km tento jav nie je až taký nezvyčajný, keďže rýchlosť pádu (vyrovnanie gravitačnej a aerodynamickej odporovej sily) je závislá od hustoty vzduchu a hustota vzduchu je závislá od nadmorskej výšky. Dôvodom pomerne rýchleho zostupu je letecká premávka. Tým je zaručený rýchly prechod najpoužívanejšími letovými hladinami (7 – 11 km nadmorskej výšky) a zníženie pravdepodobnosti kolízie s leteckou premávkou. Výsledná rýchlosť pádu je vypočítaná pomocou vzorca (1) a (2) uvedených v [2]

$$v = \sqrt{\frac{2.m.g}{\rho_{Vzduchu}.A.C_d}}, \quad (1)$$

kde:

v	je rýchlosť objektu pri páde s odporom vzduchu;
m	je hmotnosť objektu;
g	je gravitačné zrýchlenie (9,81 m.s^{-2});
A	je plocha pôdorysu objektu pri páde;
C_d	je koeficient aerodynamického odporu;
$\rho_{vzduchu}$	je hustota vzduchu (ďalší vzorec).

$$\rho_{vzduchu} = \rho_0 . \left(\frac{T_b}{T_b + L_b.(h - h_0)}\right)^{\left(1 + \frac{g.M}{R.L_b}\right)}, \quad (2)$$

kde:

ρ_0	je hustota vzduchu v nulovej výške (1,225 kg.m^{-3});
T_b	je teplota vzduchu v nulovej výške (288,15 °K);
L_b	je koeficient poklesu teploty s výškou (-0,0065 °K.m^{-1});
h	je nadmorská výška v m;
h_0	je nulová výška, rovná 0 m;
M	je molárna hmotnosť vzduchu (28,97 g.mol^{-1});
R	je univerzálna plynová konštanta (8.3144598 J.K^{-1}.mol^{-1}).

3. Tretia fáza: kontrolovaný zostup

V tomto okamihu je vo výške 3000 m.n.m. je vydaný pokyn na otvorenie hlavného riadeného padákového klzáka. V tejto výške je dostatok času na zníženie vertikálnej rýchlosti aj pri pristátí na najvyšší bod na Slovensku (2654 m.n.m.). Pri otvorení padáku sa podľa rýchlosti a smeru vetra a podľa miesta otvorenia vypočíta v prípade prekrytia možnej zóny pristátia so zakázanou zónou únikový vektor na bod s minimom únikovej funkcie. V našich podmienkach sa zakázané zóny neprekrývajú a v podstate sa systém snaží zo zakázanej zóny uniknúť. Smer tohto únikového vektora je vypočítaný ako spojnica stredu zakázanej zóny a momentálnej polohy.

3.1 Posledná zákruta

Poslednou fázou letu je otočenie systému proti vetru. Padákový klzák sa pohybuje smerom nadol rýchlosťou podľa vzorca (1) a zároveň sa pohybuje doprednou rýchlosťou danou klzavosťou. Napríklad pri klzavosti 3:1 bude pri vertikálnej rýchlosti 5 m.s^{-1} výsledná dopredná rýchlosť 15 m.s^{-1}. To je nezanedbateľná rýchlosť. Netreba zabúdať, že systém je stále unášaný vetrom s jeho rýchlosťou a smerom. Rýchlosť voči zemi je teda vektorovým súčtom rýchlosti vetra a doprednej rýchlosti systému. Zákruta má pri nulovej rýchlosti vetra tvar kružnice. Pri nenulovej rýchlosti vetra je opisovaná kružnica posúvaná vetrom. Výsledná krivka opisujúca pohyb voči zemi je teda cykloida. V prípade, že rýchlosť vetra je menšia ako dopredná rýchlosť, sa jedná o predĺženú cykloidu a v prípade, že rýchlosť vetra prevyšuje doprednú rýchlosť, sa jedná o skrátenú cykloidu. Dôležité je minimum rýchlosti voči zemi (nastáva, keď majú vektory doprednej rýchlosti a rýchlosti vetra opačný smer). Tým sa znižuje riziko poškodenia systému pri pristátí. V prípade, že systém uniká zo zakázanej zóny proti vetru, nie je nutná žiadna zákruta. V prípade, že systém uniká po vetre je nutné otočiť systém pred pristátím o 180°. Smer otáčania sa volí ako menšia hodnota s ohľadom na periodicitu 360°. Teda napríklad pri rozdiele uhlov 90° systém robí zákrutu doprava o 90° a pri rozdiele o 270° systém robí zákrutu o 90° doľava. Uhol otočenia teda nikdy nepresiahne 180°. Systém sa otočí za čas definovaný ako súčin uhlovej rýchlosti a uhla otočenia. Za tento čas však systém stratí výšku danú súčinom času otáčania a vertikálnej rýchlosti. Zákrutu je teda potrebné začať najneskôr v tomto rozdiele výšok nad terénom. V matematickom modeli bola táto výška nastavená ako parameter simulácie. Detail trajektórie poslednej zákruty sa nachádza na obr. 2.

Obr. 2 Posledná zákruta proti vetru

Fig. 2 Final turn oposite the wind

4. Prepis do Matlabu

V tejto kapitole vyberieme niektoré užitočné funkcie Matlabu a jeho modulov, ktoré by záujemcovi uľahčili podobné matematické modelovanie pohybu bodu na Zemi.

reckon

je funkcia, ktorá vráti zemepisnú šírku a dĺžku cieľa pri známej zemepisnej šírke a dĺžke štartu, známom smere pohybu (azimut) a prejdenej vzdialenosti. Vzdialenosť je nutné zadať ako uhol pohybu v sústave sférických súradníc, a preto používame ďalšiu funkciu *nm2deg*. Táto funkcia mení na povrchu Zeme vzdialenosť v námorných míľach na uhol pohybu v stupňoch.

cart2pol a *pol2cart*

sú funkcie pre zmenu z karteziánskych a polárne súradnice a naopak.

Google earth toolbox

je toolbox pre vytvorenie *.kml* súborov, ktoré slúžia ako vstup napríklad pre Google Earth Pro a je opísaný v [4]. V tomto programe je možné v 3D prostredí pozrieť trajektóriu a urobiť snímky pre nekomerčné použitie. Samotný Google Earth Pro je zadarmo. Tento toolbox obsahuje veľa funkcii pre vykreslenie úsečky, bodu, polygónu a podobne... Naklonením pohľadu je možné ľubovoľné vygenerovanie obrázka trajektórie. Najpoužívanejšie funkcie sú: *ge_plot3*, *ge_point* a *ge_circle*.

Pred týmto spôsobom sme používali štandardnú dobre známu funkciu *plot* s podkladom mapy z Google Maps (online vyžiadanie pre rozsah mapy). Použitie tohto spôsobu bolo obmedzené, pretože API-kľúč na prístup k mapám je spoplatnený.

Záver

Matematická simulácia bola jednou z požadovaných fáz projektu Európskej Vesmírnej Agentúry. Po dokončení matematických simulácií sme zhodnotili, že simulovanie letu bude pravdepodobne odlišné od reality, ale prinieslo zaujímavé zistenia a prínosy.

Tým prvým je definovanie samotných zakázaných zón ako kružníc s polomerom 5km v strede letísk (stred pristávacej dráhy) a vodných plôch.

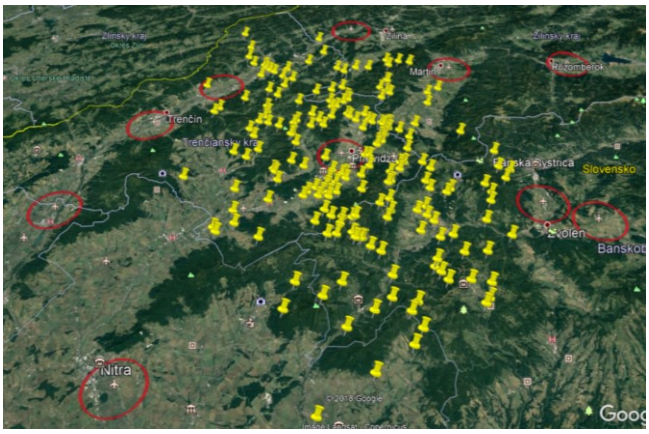
Ďalším záverom matematickej simulácie bol výber samotného miesta štartu. Na začiatok sme predpokladali štart systému v Žiline (Košice a Bratislava sú neprípustné z hľadiska blízkosti štátnej hranice). Na začiatku simulácií s prihliadnutím na štatistické údaje o prevládajúcom vetre sme však zistili, že s nenulovou pravdepodobnosťou môže dôjsť k pristátiu v Českej Republike. To je opäť neprípustné pre štart reálneho systému, ktorý by nemal prekročiť štátnu hranicu Slovenskej Republiky. Miesto štartu bolo teda posunuté do Ružomberka, kde k prekročeniu hranice už v žiadnej zo 400 simulácií nedošlo a stále sa jednalo o stredné Slovensko, pre ktoré platili štatistické údaje o vetre.

Tretí záver je vyhodnotenie samotnej funkcionality vyhýbania sa zakázaným zónam. Pri vypnutom systéme vyhýbania sa zakázaným zónam bolo 8 z 200 simulácii letu s pristátím priamo v zakázanej zóne. Pri zapnutom systéme vyhýbania sa zakázaným zónam došlo len k 1 z 200 pristátí vo vnútri zakázanej zóny a to bolo tesne na okraji (menej ako 200m od okraja 5km zóny). Systém sa snažil vyhnúť zóne, ale nadmorská výška pristátia znížila dolet od bodu otvorenia klzáku. Bolo uskutočnených len 400 simulácií, ale môžeme konštatovať, že systém vyhýbania sa zakázaným zónam je funkčný a znižuje pravdepodobnosť pristátia v zakázanej zóne (v našom prípade zo 4% na 0,5%) – obr. 3 a obr. 4.

Ďalším záverom simulácií bolo, že bude potrebné poznať výšku nad terénom. V tom bude určité úskalie reálneho systému. Systém totiž pozná svoju nadmorskú výšku z GNSS systému napríklad GPS, alebo GLONASS, prípadne z barometrického výškomera. To však nehovorí nič o výške nad terénom (je nutné poznať výšku terénu). Výšku nad terénom je teda možné získať:

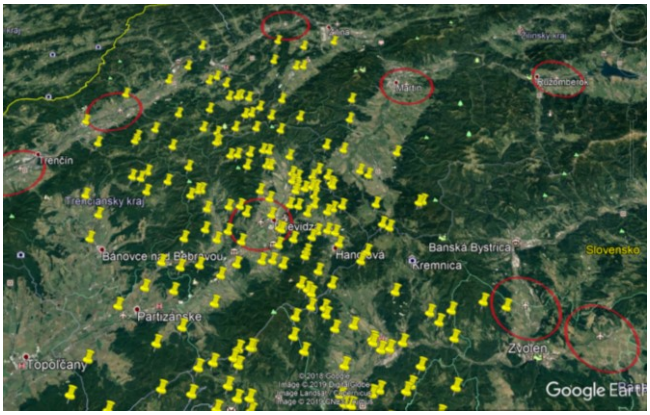
- Snímačom (napríklad laserovým), ktorý má na výstupe priamo výšku nad terénom. Tento snímač je však veľmi rozmerný a má hmotnosť rádovo stovky gramov. To je neprípustné, navyše snímač má dosah len necelých 200m a v prípade hmly vôbec nefunguje.
- Dátami o výške terénu uloženými v pamäti riadiacej jednotky. To sú ale dáta, ktoré zaberajú niekoľko GB pamäte (Slovenská Republika má rozlohu 49000 km^2 a pre každý bod v sieti sú potrebné 2 Bajty na údaj o výške).
- Poslednou možnosťou je mať komunikačný kanál s reálnym systémom a poslať systému výšku terénu pre požadovanú zemepisnú šírku a dĺžku. To je ale závislé od príjmu komunikačného signálu, ktorý s ohľadom na terén, prípadne mapu pokrytia mobilnými operátormi nemusí byť k dispozícii.

Nakoniec po skončení fáze projektu pre simuláciu a štartom reálneho systému bude matematický model upravený na reálne hodnoty skutočného systému a simulácia bude spustená bezprostredne pred štartom. Simulácia bude spustená pre aktuálne hodnoty vetra pre deň štartu a bude vypočítaná predpokladaná zóna pristátia. Tak bude možné vyslať pozemný tím na predpokladané miesto pristátia a po odvysielaní dát o mieste pristátia bude systém možné rýchlejšie nájsť. Napríklad pri štarte z Ružomberka sa pozemný tím po simulácii s predstihom vyšle do Prievidze a nebude čakať na signál o polohe pristátia. Je predpoklad, že systém pristane v okruhu niekoľko km od simulovaného miesta pristátia.



Obr. 3 Pri zapnutom systéme vyhýbania sa bolo iba 1 pristátie v zakázanej zóne (tesne na jej okraji)

Fig. 3 With the prohibited landing zone avoidance system turned on, only 1 landing occured inside the PLZ (directly on the PLZ border)



Obr. 4 Poloha bodov pristátia pri vypnutom systéme vyhýbania sa zakázaným zónam (8 z 200 pristátí bolo vo vnútri zakázaných zón)

Fig. 4 Landing points when avoidance for prohibited landing zones (PLZ) was turned off (8 of 200 landings occured inside of PLZ)

Posledným záverom fáze matematických simulácií je skutočnosť, že systém často pristával do pomerne neprístupných oblastí Veľkej Fatry, Malej Fatry, Vtáčnika a pod. Dané lokality síce neobsahovali zakázané zóny letísk, vodných plôch, ani ťažko prístupné Vysoké Tatry, ale do určitých lokalít by pozemnému tímu trvala cesta niekoľko hodín peši, prípadne cez hustú vegetáciu. Preto by bolo vhodné vytipovať preferované zóny pristátia, prípadne nechať systém dokĺzať napríklad na lúku na okraji blízkej dediny.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] CLARK, T.: Mathematical Analysis of the Equations of Motion for the Bloodhound SSC and Felix Baumgartner's Skydive, 2013, dostupné na: http://mccabeme.myweb.port.ac.uk/projects2013/thomasclarkbloodhound.pdf?fbclid=IwAR0OZvi8iDwI5WicNw64-YET8WAzgjDn0B4s1rNB8SiM16dmeOOpiyfBxk0

[2] NASA terminal velocity calculation, 2018, dostupné na: https://www.grc.nasa.gov/www/k-12/airplane/termv.html

[3] YAGER, R. J.: Calculating Atmospheric Conditions (Temperature, Pressure, Air Density, and Speed of Sound) Using C++, 2013, http://www.arl.army.mil/arlreports/2013/ARL-TN-543.pdf

[4] Dokumentácia ku Google Earth Toolboxu, 2012, dostupná na: https://www.mathworks.com/matlabcentral/fileexchange/12954-google-earth-toolbox

Abstract

This paper deals with mathematical simulation of balloon flight. The simulation was done in Matlab environment. The simulation is one of the project stages of the project SALSA (Stratospherical Autonomous Lansing System Application) The puspose of the simulation is to calculate the route of the flight before real system is built. The simulation concludes to reccomendations for real system design and for selection of the starting point.

Ing. Vojtech Šimák, PhD.

Ing. Dušan Nemec, PhD.

Ing. Marián Hruboš, PhD.

Ing. Jozef Hrbček, PhD.

Žilinská Univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 1
01026 Žilina
Tel.: +421415133304
E-mail: vojtech.simak@fel.uniza.sk

Ing. Filip Škultéty, PhD.

Žilinská Univerzita v Žiline
Fakulta Prevádzky a Ekonomiky Dopravy a Spojov
Katedra Leteckej dopravy
Univerzitná 1
01026 Žilina

Obsah

VPLYV APLIKAČNEJ DIAGNOSTIKY NA BEZPEČNOSŤ RIADIACEHO SYSTÉMU NA BÁZE SAFETY PLC

Juraj Ždánsky, Jozef Valigurský

Abstrakt
Takmer všetky komerčne dostupné riadiace systémy v súčasnosti umožňujú modulárne zostavovanie architektúry. Podobne je to aj pri riadiacich systémoch na báze safety PLC (Programmable Logic Controllers) určenými na realizáciu bezpečnostných funkcií. S rastúcou modularitou rastie aj počet možností ktorými možno realizovať bezpečnostnú funkciu, resp. funkcie. K dosiahnutiu požadovanej úrovne integrity bezpečnosti (SIL - Safety Integrity Level) bezpečnostných funkcií nevyhnutne patrí aj diagnostika. Tento príspevok sa venuje analýze vplyvu aplikačnej diagnostiky v riadiacom systéme na SIL realizovaných bezpečnostných funkcií.

Kľúčové slová: safety PLC, SIL, safety integrity level, diagnostika, bezpečnostná funkcia, Markovove reťazce

Úvod

Diagnostikou dnes disponuje každé elektronické zariadenie. Napriek tomu môže byť zámer, kvôli ktorému sa diagnostika realizuje, veľmi rozdielny pri jednotlivých zariadeniach. Pri riadiacich systémoch je diagnostika nevyhnutná nielen kvôli užívateľskému komfortu, ale je nevyhnutným nástrojom na dosiahnutie požadovaných spoľahlivostných a/alebo bezpečnostných vlastností systému. Ak porucha riadiaceho systému môže mať za následok významné škody na ľudskom zdraví, životnom prostredí alebo veľké materiálne škody, tak potom treba okrem funkčných vlastností riadiaceho systému sledovať aj bezpečnosť realizovaných funkcií. Takýto riadiaci systém sa označuje ako bezpečnostne relevantný riadiaci systém (SRCS-Safety Related Control System) a funkcie zaisťujúce bezpečnosť sa označujú ako bezpečnostné funkcie (SF-Safety Function) [1].

Spôsob vykonávania diagnostiky v SRCS je vo veľkej miere závislý aj od jeho architektúry [2]. SRCS so safety PLC (sPLC) sú modulárne riadiace systémy (ak nebude uvedené inak, tak budeme ďalej predpokladať, že SRCS je realizovaný na báze sPLC). Výrobcovi sPLC nie je vopred známe z akých komponentov bude sPLC tvoriaci SRCS zostavený, preto musí diagnostikou disponovať každý modul sPLC a zároveň musí byť zaistená aj možnosť výmeny potrebných diagnostických informácií medzi jednotlivými modulmi. To je zaistené pomocou firmvéru jednotlivých modulov. Diagnostika na úrovni firmvéru (vstavaná diagnostika; spravidla ide o funkčnú aj testovaciu diagnostiku) je nevyhnutná na dosiahnutie úrovne integrity bezpečnosti garantovanej pre každý modul sPLC.

Súčasťou SRCS sú aj snímače a koncové prvky. Tieto však už nemusia disponovať vlastným firmvérom (napr. stýkač, elektromagnetický ventil a pod.). Ak je na dosiahnutie požadovanej SIL realizovaných SF nevyhnutná ich diagnostika, tak musí byť realizovaná aplikačná diagnostika (diagnostika realizovaná v aplikačnom programe) [3], [4], [5].

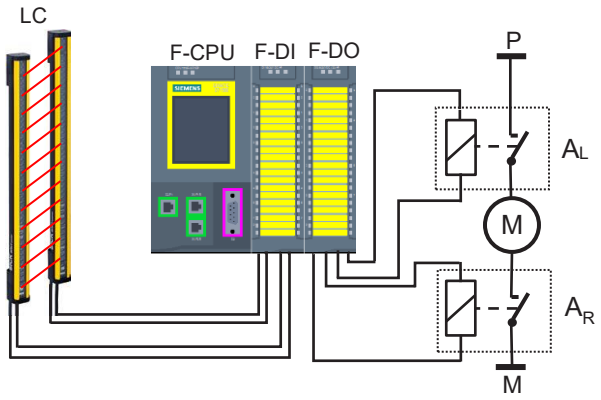
Modularita SRCS umožňuje variabilitu nielen v rámci výberu jeho komponentov, ale aj v rámci výberu architektúry SRCS. SRCS potom môže byť realizovaný nielen ako centralizovaný systém, ale vďaka priemyselným sieťam a protokolom umožňujúcim bezpečnú komunikáciu môžu byť jednotlivé časti SRCS rôzne poprepájané. Distribuovaný SRCS sa spravidla realizuje z dôvodu rozsiahlosti realizovaných funkcií (nielen bezpečnostných ale aj štandardných). Ak je SRCS realizovaný na báze sPLC, tak vstupno-výstupné safety moduly môžu byť kombinované so štandardnými modulmi v rôznych častiach distribuovaného SRCS. Safety moduly potom môžu byť použité na realizáciu SF a štandardné moduly na realizáciu bežných riadiacich funkcií.

Príspevok sa venuje porovnaniu bezpečnostných vlastností SF s aplikačnou diagnostikou koncových prvkov (stýkačov) realizovanou pomocou lokálnych a distribuovaných vstupno-výstupných modulov za predpokladu nepretržitej prevádzky (definícia nepretržitej prevádzky je uvedená v norme [1]). Pre jednoduchosť budeme predpokladať, že SRCS realizuje jednu SF, preto možno pravdepodobnosť nebezpečnej poruchy SRCS stotožniť s pravdepodobnosťou nebezpečnej poruchy realizovanej SF.

1. SRCS bez aplikačnej diagnostiky

Na obr. 1 je znázornené zapojenie SRCS bez aplikačnej diagnostiky (v zapojeniach sú použité grafické znázornenia tých komponentov, ktoré sú použité v experimentálnej časti). Safety PLC sa skladá z fail-safe centrálnej procesorovej jednotky (F-CPU), fail-safe digitálneho vstupného modulu (F-DI) a fail-safe digitálneho výstupného modulu (F-DO). Na obr. 1 nie sú znázornené komponenty, ktoré nemajú vplyv na bezpečnosť (pri dodržaní podmienok správnej aplikácie; napr. napájací zdroj). Predpokladajme, že SRCS realizuje jednoduchú SF: odpojenie motora od napájacieho napätia v prípade prerušenia optickej závory.

Motor poháňa pohyblivé časti stroja, ktoré predstavujú potenciálne nebezpečenstvo pre človeka. Preto je nevyhnutné priestor pohyblivých častí monitorovať optickou závorou (LC) a v prípade vstupu človeka do nebezpečnej zóny odpojiť motor (M) od napájacieho napätia, čím sa nebezpečná zóna uvedie bo bezpečného stavu (prestane byť nebezpečná pre človeka). Motor sa odpája pomocou dvoch stýkačov A_L a A_R. Stýkače použité v takomto zapojení nemusia mať žiadne špeciálne bezpečnostné vlastnosti.

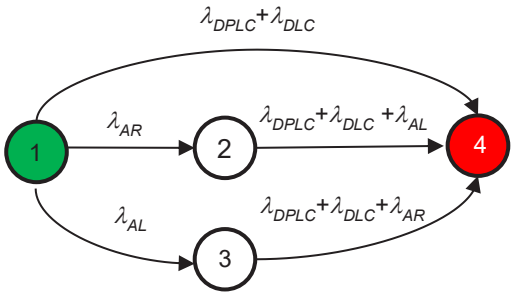


Obr. 1 SRCS bez aplikačnej diagnostiky

Fig. 1 SRCS without application diagnostics

Za nebezpečnú poruchu SRCS na obr. 1 treba považovať takú poruchu, ktorá spôsobí, že po detekcii prerušenia optickej závory motor nezastane v požadovanom čase. Časové hľadisko je veľmi dôležité, ale vzhľadom na rozsah tohto príspevku sa ním nebudeme ďalej zaoberať. Podrobnejšie sa časovému hľadisku venuje publikácia [6].

Pravdepodobnosť vzniku nebezpečnej poruchy SRCS (obr. 1) možno opísať pomocou Markovovho diagramu so spojitým časom (CTMC – Continuous Time Markov Chain) na obr. 2.



Obr. 2 CTMC – vznik nebezpečnej poruchy v SRCS bez aplikačnej diagnostiky (obr. 1)

Fig. 2 CTMC –dangerous failure occurence in SRCS without application diagnostics (fig. 1)

Stav 1 na obr. 2 predstavuje stav SRCS bez poruchy. Predpokladáme, že v tomto stave sa SRCS nachádza v čase t=0 (prakticky tento okamih možno stotožniť s okamihom uvedenia systému do prevádzky). Zo stavu 1 môže SRCS prejsť:

- do stavu 4 (nebezpečný stav) - po výskyte nebezpečnej poruchy sPLC alebo optickej závory;
- do stavu 2 – po výskyte poruchy stýkača A_R;
- do stavu 3 - po výskyte poruchy stýkača A_L.

Zo stavu 2, resp. 3 môže SRCS prejsť do stavu 4 po výskyte nebezpečnej poruchy sPLC alebo nebezpečnej poruchy optickej závory alebo po výskyte poruchy stýkača A_L, resp. A_R.

CTMC na obr. 2 možno opísať sústavou diferenciálnych rovníc a z nich odvodiť vzťah na výpočet pravdepodobnosti nebezpečnej poruchy SRCS bez aplikačnej diagnostiky (pravdepodobnosť stavu 4 v CTMC na obr. 2). Ide o pravdepodobnosť stavu 4 v CTMC na obr. 2.

$$P_4(t) = 1 - e^{-t.(\lambda_{DLC}+\lambda_{DPLC}+\lambda_{AL})} - e^{-t.(\lambda_{DLC}+\lambda_{DPLC}+\lambda_{AR})} + e^{-t.(\lambda_{DLC}+\lambda_{DPLC}+\lambda_{AL}+\lambda_{AR})}, \quad (1)$$

kde λ_{DLC} je intenzita nebezpečných porúch optickej závory LC, λ_{AL} je intenzita porúch stýkača A_L, λ_{AR} je intenzita porúch stýkača A_R a λ_{DPLC} je intenzita nebezpečných porúch sPLC. Vzťah (1) platí za predpokladu počiatočného rozdelenia pravdepodobností $\overrightarrow{P_0(t=0)} = \{1,0,0,0,0\}$.

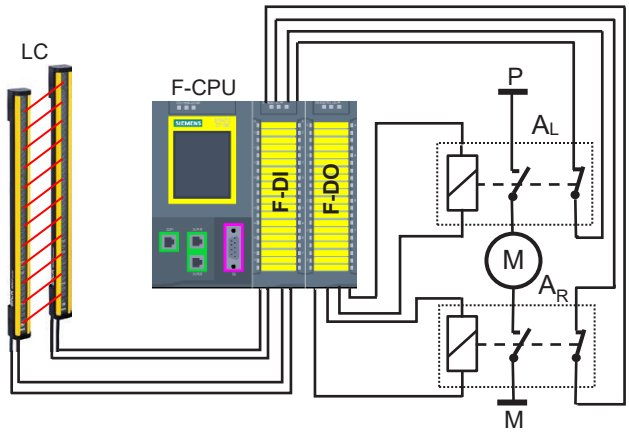
$$\lambda_{DPLC} = \lambda_{DCPU} + \lambda_{DFDI} + \lambda_{DFDO}, \quad (2)$$

kde λ_{DCPU} je intenzita nebezpečných porúch F-CPU, λ_{DFDI} je intenzita nebezpečných porúch F-DI modulu a λ_{DFDO} je intenzita nebezpečných porúch F-DO modulu.

2. SRCS s aplikačnou diagnostikou

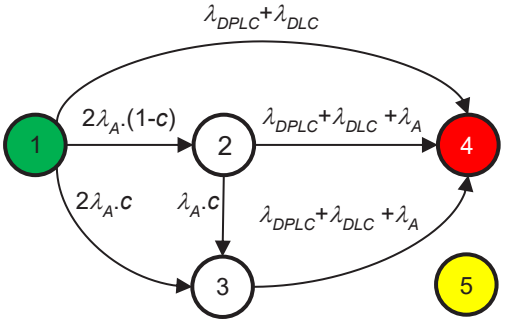
2.1 Aplikačná diagnostika realizovaná lokálnymi modulmi sPLC

Na obr. 3 je znázornené zapojenie SRCS so spätnou väzbou umožňujúcou snímať polohu stýkačov. Zapojenie je po funkčnej stránke schopné vykonávať rovnakú SF, ako zapojenie na obr. 1. Spätná väzba je nevyhnutná na realizáciu aplikačnej diagnostiky.



Obr. 3 SRCS s aplikačnou diagnostikou realizovanou lokálnymi modulmi sPLC

Fig. 3 SRCS with application diagnostics realized by local sPLC modules



Obr. 4 CTMC – vznik nebezpečnej poruchy v SRCS s aplikačnou diagnostikou (obr. 3)

Fig. 4 CTMC – dangerous failure occurence in SRCS with application diagnostics (fig. 3)

Vplyv aplikačnej diagnostiky na pravdepodobnosť nebezpečnej poruchy SRCS možno opísať kombináciou Markovových diagramov so spojitým a diskrétnym časom (DTMC – Discrete Time Markov Chain). Proces vzniku nebezpečnej

poruchy možno opísať CTMC znázorneným na obr. 4 a vplyv aplikačnej diagnostiky možno opísať pomocou DTMC na obr. 5.

CTMC na obr. 4 predpokladá, že $\lambda_{AL} = \lambda_{AR} = \lambda_A$. Tento predpoklad je splnený, ak sú použité dve identické stýkače (čo je prakticky spravidla vždy splnené). Z bezporuchového stavu 1 sa SRCS môže dostať do:

- stavu 4 (nebezpečný stav) po výskyte nebezpečnej poruchy sPLC alebo optickej závory;
- stavu 2 po výskyte nedetegovateľnej poruchy stýkača A_L, resp. A_R (c je miera diagnostického pokrytia);
- stavu 3 po výskyte detegovateľnej poruchy stýkača A_L, resp. A_R.

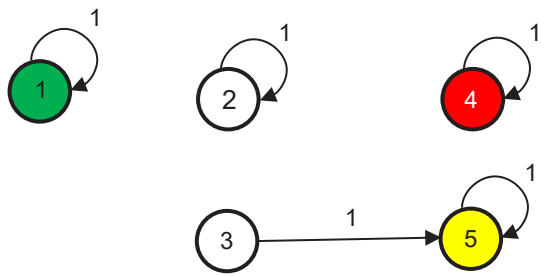
Zo stavu 2 sa SRCS môže dostať do:

- stavu 3 po výskyte detegovateľnej poruchy stýkača A_L, resp. A_R (ide o výskyt detegovateľnej poruchy toho istého stýkača na ktorom sa už vyskytla nedetegovateľná porucha);
- stavu 4 (nebezpečný stav) po výskyte nebezpečnej poruchy sPLC alebo nebezpečnej poruchy optickej závory alebo po výskyte poruchy stýkača A_R, resp. A_L (obidva stýkače majú poruchu).

Zo stavu 3 sa SRCS môže dostať do stavu 4 po výskyte nebezpečnej poruchy sPLC alebo nebezpečnej poruchy optickej závory alebo po výskyte poruchy stýkača A_R, resp. A_L (obidva stýkače majú poruchu).

Stav 5 predstavuje stav po detekcii a negácii poruchy SRCS (odpojenie motora od napájacieho napätia). Do stavu 5 neexistuje na obr. 4 žiadny prechod, pretože tento prechod môže nastať len počas vykonávania testu aplikačnej diagnostiky. Test spočíva v povele na rozpojenie stýkača a následnom vyhodnotení tejto zmeny pomocou spätnej väzby. V prípade realizácie testovacej diagnostiky je testovací interval pevne daný. V prípade realizácie funkčnej diagnostiky možno za testovací interval považovať maximálny čas medzi dvomi prevádzkovými povelmi na rozpojenie stýkačov.

Čas testu aplikačnej diagnostiky je vzhľadom na čas medzi dvomi testami zanedbateľný, preto možno vplyv aplikačnej diagnostiky na pravdepodobnosť nebezpečnej poruchy SRCS opísať pomocou DTMC (obr. 5).



Obr. 5 DTMC – vplyv aplikačnej diagnostiky v SRCS podľa obr. 3

Fig. 5 DTMC – application diagnostics influence shown in fig. 3 on SRCS parameters

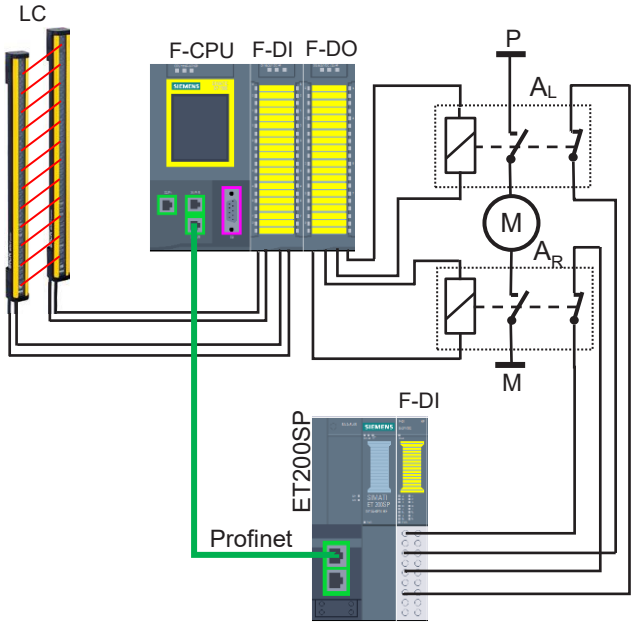
Aplikačná diagnostika je zameraná na testovanie stýkačov a je schopná odhaliť detegovateľné poruchy stýkača (stav 3 na obr. 4, resp. obr. 5). Po detekcii poruchy stýkača, SRCS prejde zo stavu 3 do stavu 5 (obr. 5). Ak sa SRCS nachádza v inom stave ako v stave 3, tak vykonanie testu stýkačov neovplyvní zmenu tohto stavu (SRCS zostáva v stavoch 1, 2, 4 alebo 5).

2.2 Aplikačná diagnostika realizovaná distribuovanými modulmi sPLC

Pri modelovaní aplikačnej diagnostiky v distribuovanom SRCS musíme zohľadniť aj vplyv ďalších hardvérových komponentov tvoriacich tento systém. Vzhľadom k tomu, že architektúra distribuovaného SRCS môže byť rôzna (rôzne architektúry môžu realizovať rovnakú SF), nie je možné vytvoriť univerzálny model zachytávajúci vplyv diagnostiky na pravdepodobnosť nebezpečnej poruchy v distribuovanom SRCS (model musí byť zostavený vždy vzhľadom na realizovanú SF a použitú architektúru).

Predpokladajme, že distribuovaný SRCS tvorí sPLC (F-CPU a lokálne moduly F-DI, F-DO) a distribuovaný fail-safe vstup (F-DI). Zapojenie takéhoto SRCS je na obr. 6. Tento jednoduchý distribuovaný SRCS realizuje rovnakú SF ako SRCS na obr. 1, pričom použitie modulov je nasledovné:

- F-CPU vykonáva aplikačný program realizujúci SF a aplikačnú diagnostiku;
- lokálny F-DI modul sníma informáciu o prerušení optickej závory LC;
- lokálny F-DO modul dáva povely na zopnutie, resp. rozopnutie pre stýkače A_L a A_R (na základe aplikačného programu v F-CPU);
- distribuovaný F-DI modul vyhodnocuje spätnú väzbu zo stýkačov A_L a A_R.

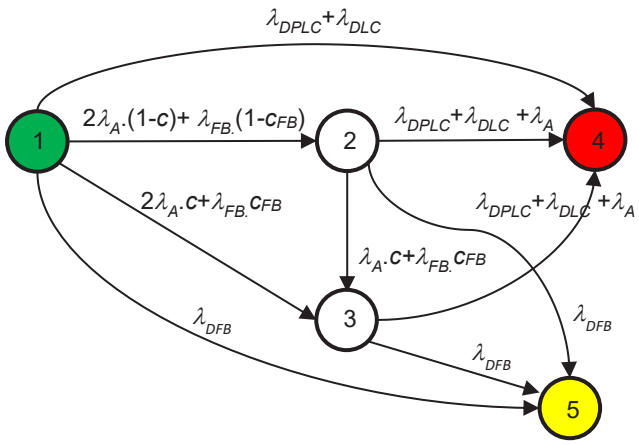


Obr. 6 SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi sPLC

Fig. 6 SRCS with application diagnostics realised by distributed sPLC modules

Proces vzniku nebezpečnej poruchy v distribuovanom SRCS (obr. 6) možno opísať CTMC znázorneným na obr. 7.

Pri aplikačnej diagnostike stýkačov dochádza nepriamo aj k diagnostike distribuovaného F-DI modulu a komponentov slúžiacich na komunikáciu F-DI modulu s F-CPU. Porucha v spätnej väzbe môže ovplyvniť aplikačnú diagnostiku stýkačov, preto musí byť zahrnutá do CTMC.



Obr. 7 CTMC – vznik nebezpečnej poruchy v SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi sPLC (obr. 6)

Fig. 7 CTMC – dangerous failure occurence in SRCS with application diagnostics realised by distributed sPLC modules (fig. 6)

Na rozdiel od CTMC na obr. 4 v CTMC na obr. 7 existujú prechody do stavu 5 zo stavov 1, 2 a 3. Intenzita týchto prechodov je daná intenzitou nebezpečnej poruchy v spätnej väzbe:

$$\lambda_{DFB} = \lambda_{DDI} + \lambda_{Dcom}, \tag{3}$$

kde λ_{DDI} je intenzita nebezpečných porúch distribuovaného F-DI modulu a λ_{Dcom} je intenzita nebezpečných porúch komunikácie.

Prechody do stavu 5 vychádzajú z predpokladu, že spätná väzba zo stýkačov sa sníma z pomocných rozpínacích kontaktov a bezpečný stav sPLC (a teda aj každého komponentu z ktorého sa sPLC skladá) je definovaný ako logická nula. Pri povele na zopnutie stýkača je aplikačnou diagnostikou očakávaná zo spätnej väzby log. 0. (a naopak pri povele na rozopnutie stýkača je očakávaná log. 1). V čase medzi vykonaním dvoch testov aplikačnou diagnostikou (stýkače dostávajú povel na zopnutie a zo spätnej väzby prichádza log. 0 – v súlade so špecifikáciou SF) sa nebezpečná porucha (hlásenie log. 1 namiesto log. 0) v spätnej väzbe okamžite prejaví, následkom čoho sPLC odpojí stýkače (negácia poruchy) a SRCS prejde do bezpečného stavu (stav 5).

Ak nastane porucha spätnej väzby, ktorá sa neprejaví okamžite, ale je detegovateľná pri teste aplikačnej diagnostiky, tak SRCS prejde do stavu 3 (rovnako ako pri detegovateľnej poruche stýkača). Ak táto porucha nie je detegovateľná, tak SRCS prejde do stavu 2. Prechod zo stavu 2 do stavu 3 zohľadňuje skutočnosť, že po výskyte nedetegovateľnej poruchy môže nastať detegovateľná porucha spätnej väzby.

Intenzitu porúch spätnej väzby možno stanoviť zo strednej doby medzi poruchami (MTBF – Mean Time Between Failures) pre moduly v spätnej väzbe a pre komunikáciu.

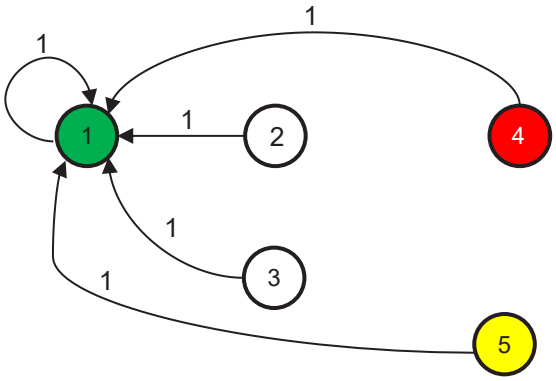
$$\lambda_{FB} = \frac{1}{MTBF_{FDI}} + \frac{1}{MTBF_{IM}} + \lambda_{com}, \tag{4}$$

kde $MTBF_{FDI}$ je stredná doba medzi poruchami F-DI modulu, $MTBF_{IM}$ je stredná doba medzi poruchami interface modulu ET200SP a λ_{com} je intenzita porúch pri komunikácii (túto je síce problematické stanoviť, ale vzhľadom na to, že proti intenzitám porúch F-DI modulu a interface modulu je rádovo menšia, možno ju zanedbať). Podrobnejšie sa prenosu informácií venujú publikácie [7] a [8].

Problémom však zostáva stanoviť mieru pokrytia porúch spätnej väzby (c_{FB}) prostredníctvom testu aplikačnej diagnostiky stýkačov. V najnepriaznivejšom prípade bude nevyhnutné predpokladať, že c_{FB}=0.

Čas testu aplikačnej diagnostiky je vzhľadom na čas medzi dvomi testami zanedbateľný, preto jeho vplyv na pravdepodobnosť nebezpečnej poruchy možno opísať pomocou DTMC (obr. 5).

Intenzity nebezpečných porúch jednotlivých modulov sPLC a optickej závory sú výrobcami udávané za predpokladu, že interval kontrolnej skúšky (tzv. proof test) je 20 rokov (snahou výrobcov je tento interval stotožniť so životnosťou systému). Vplyv dokonalej kontrolnej skúšky SRCS možno opísať DTMC znázorneným na obr. 8. Podrobnejšie sa vplyvu obnovy na bezpečnosť SRCS venuje publikácia [9].



Obr. 8 DTMC – vplyv proof testu na pravdepodobnosť nebezpečnej poruchy SRCS podľa obr. 6

Fig. 8 DTMC – proof test influence on dangerous failure probability of SRCS according to fig. 6

CTMC na obr. 7 možno opísať sústavou diferenciálnych rovníc

$$\begin{aligned}
&P_1\acute{}(t) = -(2\lambda_A + \lambda_{FB} + \lambda_{DPLC} + \lambda_{DLC} + \lambda_{DFB}).P_1(t)\ ,\\
&P_2\acute{}(t) = \left(2\lambda_A.(1-c) + \lambda_{FB}.(1-c_{FB})\right).P_1(t) - (\lambda_A.c + \\
&\lambda_{FB}.c_{FB} + \lambda_{DPLC} + \lambda_{DLC} + \lambda_A + \lambda_{DFB}).P_2(t),\\
&P_3\acute{}(t) = (2\lambda_A.c + \lambda_{FB}.c_{FB}).P_1(t) + (\lambda_A.c + \\
&\lambda_{FB}.c_{FB}).P_2 - (\lambda_{DPLC} + \lambda_{DLC} + \lambda_A + \lambda_{DFB}).P_3(t)\ ,\\
&P_4\acute{}(t) = (\lambda_{DPLC} + \lambda_{DLC}).P_1(t) + (\lambda_{DPLC} + \lambda_{DLC} + \\
&\lambda_A).P_2(t) + (\lambda_{DPLC} + \lambda_{DLC} + \lambda_A).P_3(t)\ ,\\
&P_5\acute{}(t) = (\lambda_{DFB}).P_1(t) + (\lambda_{DFB}).P_2(t) + (\lambda_{DFB}).P_3(t).
\end{aligned} \tag{5}$$

DTMC na obr. 5 možno opísať maticou pravdepodobnosti prechodu

$$\mathbb{P} = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{pmatrix}. \tag{6}$$

Predpokladajme periodické opakovanie testu stýkačov aplikačnou diagnostikou. Nech časový interval medzi dvomi testami je t_t. Potom možno počiatočné rozdelenie pravdepodobnosti sústavy diferenciálnych rovníc pre *n*+1 časový úsek vypočítať podľa vzťahu:

$$\overrightarrow{P_{n+1}(t=0)} = \overrightarrow{P_n(t=t_t)}.\mathbb{P}, \tag{7}$$

kde *n* je poradie testu počas života systému a $\overrightarrow{P_0(t=0)} = \{1,0,0,0,0\}$.

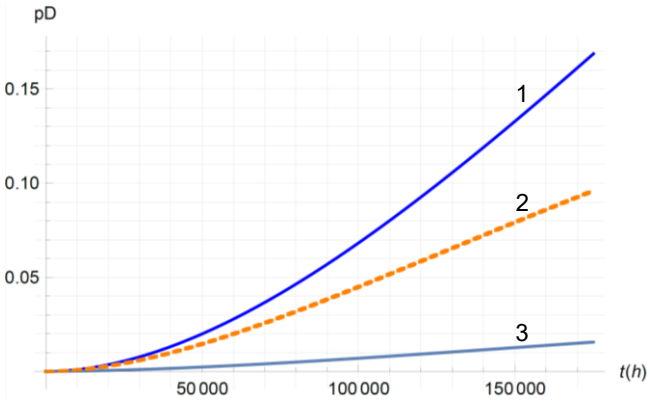
3. Experimentálne výsledky

Predpokladajme, že na realizáciu distribuovaného SRCS (obr. 6) sú použité nasledujúce komponenty:

- safety light curtain - $\lambda_{DLC} = 7.10^{-9}\ \mathrm{h}^{-1}$;

- fail-safe centrálna procesorová jednotka (CPU 1516F-3 PN/DP) - $\lambda_{DCPU} = 1.10^{-9}\ \mathrm{h}^{-1}$;
- fail-safe vstupný modul (F-DI 16x24V DC) - $\lambda_{DFDI} = 1.10^{-9}\ \mathrm{h}^{-1}$;
- fail-safe výstupný modul (F-DQ 8x24V DC/2A PPM) - $\lambda_{DFDO} = 2.10^{-9}\ \mathrm{h}^{-1}$;
- fail-safe vstupný modul (ET 200SP, F-DI 8x24VDC HF) - $\lambda_{DDI} = 1.10^{-9}\ \mathrm{h}^{-1}$, $MTBF_{FDI}$ = 65,04 rokov;
- komunikácia medzi CPU a ET200SP (pomocou Profisafe protokolu) $\lambda_{Dcom} = 1.10^{-9}\ \mathrm{h}^{-1}$;
- modul rozhrania ET200SP - $MTBF_{IM}$ = 74,08 rokov;
- stýkač $\mathrm{A_L}$, resp. $\mathrm{A_R}$ - $\lambda_{AL} = \lambda_{AR} = 3.10^{-6}\ \mathrm{h}^{-1}$.

Na obr. 9 sú znázornené časové priebehy pravdepodobnosti nebezpečnej poruchy SRCS bez aplikačnej diagnostiky (obr. 1) – priebeh 1, SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi sPLC (obr. 6) za predpokladu $t_t = 24\ h$, c=0,9 a c_{FB}=0 – priebeh 2 a SRCS s aplikačnou diagnostikou realizovanou lokálnymi modulmi sPLC (obr. 3) za predpokladu $t_t = 24\ h$, c=0,9 – priebeh 3.

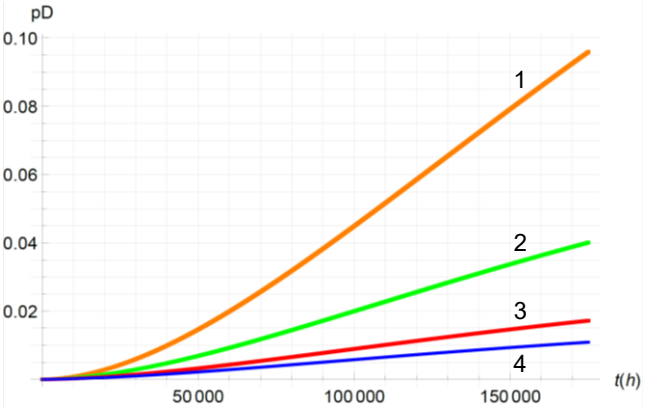


Obr. 9 Pravdepodobnosť nebezpečnej poruchy SRCS (1 – SRCS bez aplikačnej diagnostiky; 2 –SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi; 3 – SRCS s aplikačnou diagnostikou realizovanou lokálnymi modulmi)

Fig. 9 Dangerous failure probability of SRCS (1 – SRCS withou application diagnostics; 2 –SRCS with aplication diagnostics realised with distributed modules; 3 – SRCS with application diagnostics realised with local modules)

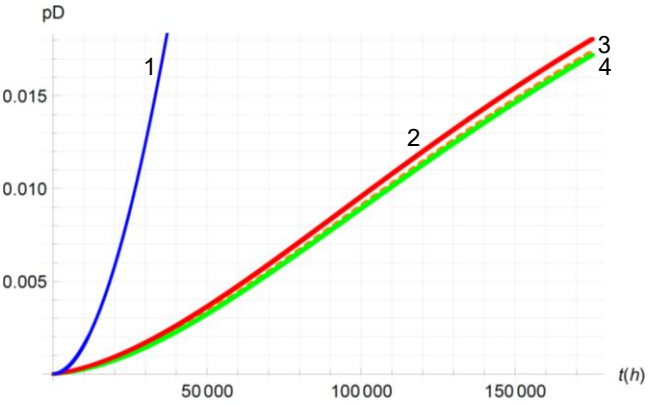
Z obr. 9 vidieť, že aplikačná diagnostika stýkačov cez distribuovaný F-DI modul pozitívne vplýva na pravdepodobnosť nebezpečnej poruchy SRCS (pravdepodobnosť nebezpečnej poruchy je v prípade distribuovaného SRCS s aplikačnou diagnostikou (obr. 6) nižšia ako pravdepodobnosť nebezpečnej poruchy SRCS bez aplikačnej diagnostiky (obr. 1)), je však silno závislá od miery pokrytia porúch spätnej väzby (c_{FB}) prostredníctvom testu aplikačnej diagnostiky stýkačov. Ak c_{FB}>0 (čo možno reálne predpokladať), tak pravdepodobnosť nebezpečnej poruchy bude nižšia. Vplyv c_{FB} na pravdepodobnosť nebezpečnej poruchy SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi je znázornený na obr. 10. Priebehy sú zostavené za predpokladu c=0,9, t_t=24 h a c_{FB}=0 - priebeh 1, c_{FB}=0,6 - priebeh 2, c_{FB}=0,9 – priebeh 3 a c_{FB}=0,99 - priebeh 4).

Vplyv časového intervalu medzi dvomi testami stýkačov (čas t_t) na pravdepodobnosť nebezpečnej poruchy SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi sPLC je znázornený na obr. 11. Priebehy sú zostavené za predpokladu c=0,9, c_{FB}=0,9 a t_t=24 h - priebeh 1, t_t=168 h (1 týždeň) - priebeh 2, t_t=744 h (1 mesiac) - priebeh 3, t_t=175 200 h (20 rokov; predpokladaná životnosť systému) - priebeh 4). Priebehy 3 a 4 na obr. 11 takmer splývajú, čo je spôsobené mierkou grafu (táto je zámerne zvolená tak, aby bolo možné vidieť vplyv časového intervalu medzi dvomi testami stýkačov v širokom rozsahu).



Obr. 10 Pravdepodobnosť nebezpečnej poruchy SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi (c=0,9, $t_t = 24\ h$ a c_{FB}=0 – priebeh 1, c_{FB}=0,6 – priebeh 2, c_{FB}=0,9 – priebeh 3, c_{FB}=0,99 – priebeh 4)

Fig. 10 Dangerous failure probability of SRCS with application diagnostics realised by distributed modules (c=0,9, $t_t = 24\ h$ a c_{FB}=0 – curve 1, c_{FB}=0,6 – curve 2, c_{FB}=0,9 – curve 3, c_{FB}=0,99 – curve 4)



Obr. 11 Pravdepodobnosť nebezpečnej poruchy SRCS s aplikačnou diagnostikou realizovanou distribuovanými modulmi (c=0,9, c_{FB}=0,9 a t_t=24 h - priebeh 1, t_t=168 h (1 týždeň) - priebeh 2, t_t=744 h (1 mesiac) - priebeh 3, t_t=175 200 h (20 rokov; predpokladaná životnosť systému) - priebeh 4))

Fig. 11 Dangerous failure probability of SRCS with application diagnostics realised by distributed modules (c=0,9, c_{FB}=0,9 and t_t=24 h - curve 1, t_t=168 h (1 week) - curve 2, t_t=744 h (1 month) - curve 3, t_t=175 200 h (20 years; required system lifetime) - curve 4))

4. Záver

Preukázanie úrovne integrity bezpečnosti bezpečnostných funkcií je nevyhnutným predpokladom ich reálneho použitia. Príspevok sa zaoberá vplyvom náhodných porúch hardvéru na SIL realizovaných bezpečnostných funkcií (tzv. hardvérová integrita bezpečnosti). Hardvérovú integritu bezpečnosti možno stanoviť na základe vhodného modelu.

Pri vytváraní modelu treba pamätať na to, aby bol model reálne použiteľný. Nie je problém zostaviť veľmi podrobný model, problémom však je získať údaje pre jeho kvantitatívne vyhodnotenie. Všetky CTMC v tomto príspevku sú zostavené tak, aby boli reálne použiteľné na výpočet pravdepodobnosti

nebezpečnej poruchy SRCS. Preto napríklad CTMC nijako nezohľadňujú počet vstupov, resp. výstupov F-DI, resp. F-DO modulov použitých na realizáciu SF. Výrobca totiž poskytuje len intenzitu nebezpečných porúch modulu ako celku. Z pohľadu bezpečnosti je preto nevyhnutné použiť tento údaj bez ohľadu na to, či SF používa všetky vstupy, resp. výstupy modulu, alebo len jeden vstup, resp. výstup. Takto získané výsledky sú síce pesimistickejšie, ale z pohľadu bezpečnosti akceptovateľné.

Okrem vplyvu náhodných porúch hardvéru treba venovať pozornosť aj integrite bezpečnosti proti systematickým chybám softvéru. Tejto problematike sa bližšie venujú napr. publikácie [10], [11].

Poďakovanie

Článok bol vypracovaný s podporou Kultúrnej a edukačnej grantovej agentúry MŠVVaŠ SR KEGA, projekt č. 008ŽU-4/2019: Modernizácia a rozšírenie možností vzdelávania v oblasti bezpečného riadenia priemyselných procesov pomocou safety PLC.

Literatúra

[1] EN IEC 61508: Functional safety of electrical/electronic/programmable electronic safety-related systems, 2010.

[2] ŽDÁNSKY, J., RÁSTOČNÝ, K., HRBČEK, J.: Influence of Architecture and Diagnostic to the Safety Integrity of SRECS Output Part, 20th International Conference on Applied Electronics, AE 2015; Pilsen, Czech Republic, September 8-9, p. 297-301, ISBN 978-802610385-1.

[3] RÁSTOČNÝ, K., ŽDÁNSKY, J., BALÁK, J., HOLEČKO, P.: Effects of Diagnostic on the Safety of a Control System Realised by Safety PLC, 11th International Conference, ELEKTRO 2016, Slovakia, May 16-18, p. 462-467, ISBN 978-1-4673-8698-2.

[4] ROUSAND, M.: Reliability of Safety-Critical Systems, Theory and Applications, Published by John Wiley & Sons, Hoboken, New Jersey, 2014, ISBN: 978-1-118-11272-4.

[5] SMITH, D.: The Safety Critical System Handbook, Published by Elsevier, 2016, ISBN 978-0-12-805121-4.

[6] ŽDÁNSKY, J., RÁSTOČNÝ, K.: Influence of Safety PLC Parameters to Response Time of Safety Functions, Proceedings of International Conference Applied Electronics, AE 2013, Pilsen, Czech Republic, Sep 10-12, p. 327-330, ISBN 978-80-261-0166-6, ISSN 1803-7232.

[7] RÁSTOČNÝ, K., FRANEKOVÁ, M., HOLEČKO, P., ZOLOTOVÁ, I.: Modelling of Hazards Effect on Safety Integrity of Open Transmission Systems, Computing and Informatics, Volume 35, Issue 2, p. 470-496, 2016, ISSN 1335-9150.

[8] FRANEKOVÁ, M., RÁSTOČNÝ, K., LULEY, P.: Practical Problems Within Safety Related Cryptography Communication Systems Assessment for Safety Critical Applications, 16th International Conference on Transport Systems Telematics, Ustron, Poland, Mar 16-19, 2016, p. 163-174, ISBN 978-3-319-49646-7, ISSN 1865-0929.

[9] RÁSTOČNÝ, K., ILAVSKÝ, J.: Effects of Recovery on the Safety of a Safety-related Control System, Proceedings of International Conference on Applied Electronics, AE 2011, Pilsen; Czech Republic; September 7-8, ISSN 1803-7232. ISBN 978-80-7043-987-6.

[10] DARVAS, D., MAJZIK, I., VINUELA, EB.: Formal Verification of Safety PLC Based Control Software, 12th International Conference on Integrated Formal Methods, Reykjavik, Iceland, Jun 01-05, 2016, p. 508-522, ISBN 978-3-319-33693-0, ISSN 0302-9743.

[11] BIALLAS, S., KAMIN, V., KOWALEWSKI, S., SCHLICH, B., SEHESTEDT, S., STATTELMANN, S.: Verification of Safety-Critical PLC Programs using Safety Automata, 14th Branch Meeting of Measurement and Automation Technology - Automation 2013, Baden, Germany, p. 75-79, ISBN 978-3-18-092209-6, ISSN 0083-5560.

Abstract

Nowadays almost all commercially available control systems, allow modularity of architecture. It is also similar for control systems based on safety PLC (Programmable Logic Controllers) intended for realization of safety functions. With more possibilities of modularity, we have more choices, how to realize safety functions. Diagnostics is also necessary to achieve the required safety integrity level (SIL) of safety functions. This paper addresses the analysis of the influence of diagnostics in control system on the safety integrity level of realized safety functions.

doc. Ing. Juraj Ždánsky, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univezitná 8215/1
010 26 Žilina
Tel.: +421 41 513 3342
E-mail: juraj.zdansky@fel.uniza.sk

Ing. Jozef Valigurský

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univezitná 8215/1
010 26 Žilina
E-mail: jozef.valigursky@fel.uniza.sk

Obsah

VYUŽITIE ČÍSLICOVÉHO SPRACOVANIA OBRAZU V INTELIGENTNÝCH DOPRAVNÝCH SYSTÉMOCH

Emília Bubeníková

Abstrakt
Moderné protikolízne systémy implementované v rámci inteligentných dopravných systémov pracujú na princípe analýzy rôznych typov senzorov. V mnohých aplikáciách je využité spracovanie získanej obrazovej informácie z prostredia okolo vozidla. Na kontrolu vybočenia vozidla z jazdného pruhu sa v kooperatívnych inteligentných systémoch používajú aj systémy sledovania opustenia jazdného pruhu. Veľmi dôležitým parametrom týchto systémov je spoľahlivosť použitej metódy číslicového spracovania obrazu, ktorá závisí od voľby optimálneho algoritmu na detekciu vodorovného dopravného značenia v rôznych svetelných podmienkach. Parametre algoritmov spracovania obrazovej informácie je vhodné nastaviť na základe SW simulácie z reálne získaných dopravných dát. Príspevok je zameraný na hľadanie a testovanie efektívnych metód číslicového spracovania obrazu v systémoch sledovania prekročenia čiary a ich využiteľnosť v prípadnej následnej medzivozidlovej komunikácii.

Kľúčové slová: inteligentné dopravné systémy, aplikácie zamerané na bezpečnosť, cestná doprava, LDWS systémy, VANET siete, Houghova transformácia

Úvod

Inteligentné dopravné systémy (IDS) sú pokročilé aplikácie, ktoré vďaka vývoju výkonnejších procesorov, komunikačných techník a rôznych typov senzorov, umožňujú v dopravných prostriedkoch zaviesť viac riadiacich, monitorovacích, bezpečnostných funkcií a tiež funkcií infotainmentu. Ich zavedenie zvyšuje pohodlie vodiča ale predo-všetkým sa snažia minimalizovať dôsledky dopravných nehôd prípadne im predchádzať [1]. Tieto systémy sú známe aj pod pojmami antikolízne systémy, systémy včasného varovania, asistenčné systémy alebo pre-crash systémy.

Mnohé z nich sú založené okrem využitia senzorovej techniky aj na spracovaní obrazových dát prostredníctvom metód počítačového videnia [2]. Snímanie obrazových dát získavaných pomocou senzorov obrazu a ich následné spracovanie predstavujú inováciu automobilového priemyslu a bezpečnosti cestnej premávky, pretože prispievajú k rozvoju asistenčných systémov pre vodičov [3].

V súčasnosti v oblasti automobilového priemyslu stále prebieha vylepšenie algoritmov, ktoré sú schopné napr. z obrazovej informácie určiť, či napríklad vozidlo neúmyselne opúšťa svoj jazdný pruh ešte predtým, ako by mohlo dôjsť k nehode. Vo väčšine prípadov sa navrhnuté metódy vyhľadávania jazdných pruhov overujú vo vhodne zvolenom modelovacom prostredí z reálne vytvorených videozáznamov za rôznych svetelných podmienok. Tento typ asistenčného systému je známy obvykle pod označením LDWS (Lane Departure Warning System) v tom prípade, ak ide o systém, ktorý varuje vodiča pred neúmyselným prekročením jazdného pruhu prostredníctvom zvukového, obrazového alebo haptického varovania alebo pod názvom LKS (Lane Keeping System) v prípade, ak dochádza aj k priamemu zásahu do vedenia vozidla a jeho udržaniu v jazdnom pruhu.

LDWS systémy sú navrhnuté tak, aby minimalizovali nehody tým, že upozorňujú vodiča na potencionálne problémy, ktoré môžu byť príčinou kolízií (napr. neúmyselné vychýlenie auta z jazdného pruhu v dôsledku rozptýlenia vodiča prípadne jeho ospalosti a podobne). Tento systém je obzvlášť účinný v situáciách, keď je cesta stále rovná a vodiči majú tendenciu nevenovať dostatočnú pozornosť vozovke.

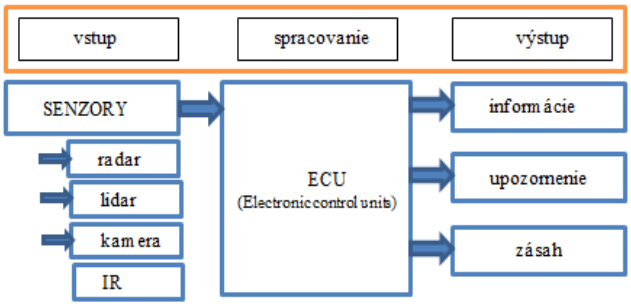
Príspevok sa podrobnejšie venuje systémom sledovania neúmyselného opustenia jazdného pruhu z pohľadu zistenia optimálnych nastavení vstupujúcich parametrov pre zaistenie spoľahlivej detekcie vodorovného dopravného značenia z videozáznamu na základe SW simulácie.

1. Asistenčné systémy a trendy v ITS

Služby spomínané v úvode príspevku zaraďujeme k službám pokročilých asistenčných systémov vodiča (Advanced Driver Assistance System, ADAS. Systémy ADAS používajú na zber fyzických údajov o vozidle a okolí vozidla rôzne typy snímačov (obr.1). Objekty ako sú blízke vozidlá, značenie jazdných pruhov, dopravné značky, prekážky na cestách a podobne sú zachytené rôznymi typmi snímačov prostredia. Patria k nim napr. ultrasonické senzory, RADARy [4], [5], LIDARy [6], IR senzory [7], senzory GPS (ktoré môžu napríklad detegovať pevné nebezpečenstvá, ako napríklad blížiace sa značky, cez databázu ich umiestnení) a najdôležitejšou skupinou snímačov sú kamerové systémy. Rôzne typy snímačov a ich prípadná fúzia tak umožňujú realizáciu funkcií ako ACC (Adaptive Cruise Control), EBA (Emergency Brake Assist, LDW (Lane Detection Warning), LKS (Lane Keep Assist),

FCW (Forward Collision Warning), rozpoznávanie dopravných značiek, detekcia chodcov atď. Moderné systémy ADAS pôsobia v reálnom čase prostredníctvom výstrah pre vodiča alebo ovládaním riadiacich systémov a sú predchodcami vozidiel budúcnosti. V súčasnosti má každé vozidlo v priemere 60-100 snímačov. Vzhľadom k tomu, že autá sa stávajú „inteligentnejšie“ predpokladá sa, že počet snímačov dosiahne až 200 snímačov na jedno auto. Tieto čísla predstavujú približne 22 miliárd senzorov používaných v automobilovom priemysle ročne do roku 2020 [8] .

Dnešné kamerové senzory používajú obrazové senzory na báze CMOS-HD s 1-2 Mpx. Vývoj kamerových systémov vedie k tomu, že z centralizovanej architektúry spracovania obrazovej informácie z viacerých kamier v centrálnej riadiacej jednotke (ECU) sa prejde na distribúciu spracovania obrazu priamo do inteligentných kamier. V prvej fáze (v rámci inteligentnej kamery) sa obraz spracuje a vykonajú sa geometrické transformácie, ako je napríklad EQ rybie oko, kompresia obrazu, ako aj spracovanie a streamovanie.

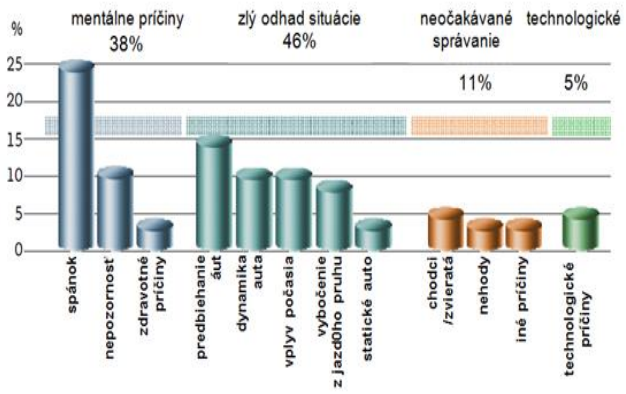


Obr. 1 Princíp práce ADAS systémov

Fig. 1 Principle of ADAS systems

V centralizovanom spôsobe spracovania je prvým krokom zber údajov, druhá fáza je predspracovanie a tretia etapa je následné spracovanie. Počas fázy predspracovania sa na celý nasnímaný obraz aplikujú rôzne funkcie vylepšenia vlastností, redukcia šumu, konverzia farieb a pod. Počas fázy po spracovaní sa vykonávajú rôzne funkcie, ako je rozpoznanie a interpretácia obrazového prostredia.

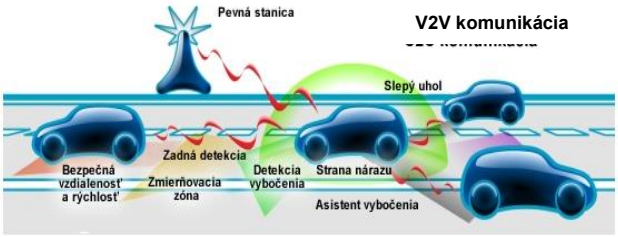
Príspevok sa venuje problematike spracovania obrazu a jeho využitia v hľadaní vodorovného dopravného značenia na vozovke. Niektoré časti problematiky boli už podrobnejšie uvedené v práci [9]. Význam systémov varovania pred neúmyselným vybočením z jazdného pruhu spočíva v odstránení dôsledkov nepozornosti vodiča pri riadení vozidla, ktorá patrí na prvé miesto v príčinách nehôd. Na obr. 2 je uvedená štatistika príčin fatálnych dopravných nehôd uverejnená automobilovou spoločnosťou Volkswagen [10].



Obr. 2 Príčiny dopravných nehôd [10]

Fig. 2 Causes of traffic accident [10]

K novej generácii inteligentných dopravných systémov patria kooperatívne inteligentné dopravné systémy (Cooperative Intelligent Transport Systems, C-ITS), ktoré poskytujú najmä služby, ktoré zvyšujú bezpečnosť cestnej premávky ako napríklad upozornenie na nebezpečné miesta, displeje s informáciami o obmedzení rýchlosti. V lokálnej bezdrôtovej sieti poskytujú hodnotné informácie vodičom priamo vo vozidle a to prostredníctvom prenosu rôznych typov varovných správ v dosahu prostredníctvom komunikácie V2V (Car to Car), resp. C2I (Car to Infrastructure) [11]. Technológie komunikácie krátkeho dosahu vo frekvenčnom pásme 5,9 GHz medzi vozidlami a dopravnou infraštruktúrou, sa označujú V2X (vozidlo-vozidlo, vozidlo-infraštruktúra) sú založené na WiFi, ETSI ITS-G5 a IEEE 802.11p. V porovnaní s inými komunikačnými technológiami štandard ETSI ITS-G5 je rozšírením všeobecného štandardu WiFi, ktorý bol upravený a optimalizovaný pre prevádzku v dynamickom automobilovom prostredí a využíva aplikácie súvisiace s bezpečnosťou. Komunikáciu medzi jednotlivými elementmi C-ITS sprostredkováva Ad-hoc sieť VANET (obr. 3), ktorá sa v súčasnosti považuje za jednu z najdôležitejších bezdrôtových komunikačných technológií v oblasti inteligentných dopravných systémov. Podporuje množstvo informačných, varovných a asistenčných služieb, ktoré budú postupne nasadzované v rôznych etapách inovácií v nasledujúcich rokoch.



Obr. 3 Princíp komunikácie medzi vozidlami prostredníctvom VANET siete

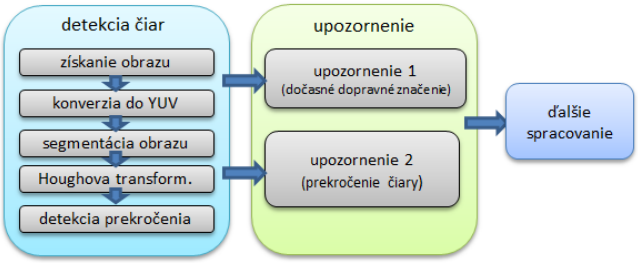
Fig.3 Principle of communication between vehicles via VANET

Aplikácie C-ITS využívajú prijaté správy z mobilných uzlov VANET (Vehicular ad hoc Network) siete a uzlov umiestnených popri cestnej infraštruktúre na bezpečné riadenie pohybujúcich sa vozidiel v zvislosti od vniknutej dopravnej situácie [12]. Správy typu broadcast, zasielané všetkým vozidlám v dosahu, obsahujú základné informácie o rýchlosti, rozmeroch, polohe vozidiel v dosahu. V závislosti od špecifickej udalosti (dopravná nehoda, zistené nebezpečenstvo na ceste), však môže každý uzol spustiť prenos ďalších správ (tzv. alert správy), ktoré informujú o vzniknutom nebezpečenstve. Takýto typ bezpečnostne-relevantných správ musí byť autorizovaný, t. j. doplnený o kryptograficky vytvorený digitálny podpis a časovú pečiatku [13], [14].

Na celom svete vzniklo za posledné desaťročie množstvo projektov, konzorcií a združení, ktoré zastrešovali mnohé medzinárodné projekty základného ako aj aplikovaného výskumu v spojení s renomovanými automobilkami. V Európe má významné postavenie konzorcium C2C-CC (Car 2 Car Communication Consortium) [15]. V súčasnosti má 88 členov, z toho 18 výrobcov automobilov, 39 výrobcov zariadení a 31 výskumných inštitúcií a úzko spolupracuje s európskymi a medzinárodnými normalizačnými organizáciami ako ETSI, CENELEC a CEN. Konzorcium C2C a platforma C-Roads podpísali v roku 2017 memorandum o porozumení, ktoré umožňuje úzku spoluprácu medzi automobilovým priemyslom, cestnými orgánmi a prevádzkovateľmi cestnej premávky na prípravu zavedenia počiatočných služieb a interoperability v oblasti IDS v celej Európe do roku 2019 [16], [17].

2. Detekcia čiar prostredníctvom algoritmov číslicového spracovania obrazu

Detekcia čiar v aplikáciách inteligentných dopravných systémov pracuje na základe strojového videnia (teda číslicového spracovania obrazovej informácie) z kamier umiestnených na vozidle. Jednotlivé etapy spracovania obrazovej informácie je možné vidieť na obr.4. Blok „ďalšie spracovanie“ môže predstavovať prepojenie systému napr. na priamy zásah do riadenia vozidla ako je to v prípade systémov pre udržanie vozidla v jazdnom pruhu (LKS) alebo pre prenos varovania do ďalších vozidiel prostredníctvom VANET sietí a medzi-vozidlovej komunikácie C2C [11], [12], [15]. Základný algoritmus detekcie čiar predpokladá, že cesty sú značené bielymi vodorovnými (plnými, prípadne prerušovanými čiarami) po oboch stranách cesty. Algoritmus je rozšírený aj o detekciu oranžových čiarami v prípade použitia dočasného dopravného značenia. Tento predpoklad platí pre väčšinu diaľnic a hlavných ciest v mestách.



Obr. 4 Postup realizácie systému detekcie prekročenia pruhu a jeho prepojenia na blok „ďalšieho spracovania“

Fig. 4 The sequence of steps in the implementation of the detection of exceeding the lane and its links to block "Next processing"

3. Získané výsledky

Pri detekcii vodorovného dopravného značenia bol využitý programový nástroj Matlab, v ktorom bola spracovaná získaná dopravná scéna pomocou záznamovej kamery Genius DVR-HD560 (HD Wide Angle Vehicle Recorder). Pri získavaní obrazových dát dopravnej scény bola kamera umiestnená v interiéri na čelnom skle automobilu. Obrazové dáta boli pred samotným spracovaním konvertované z formátu .AVI do formátu .MJPEG.

V procese segmentácie obrazu a hľadania čiar je možné použiť mnohé segmentačné metódy ako napr. [18], [19], [20] atď. Pri hľadaní hrán na ceste sa aj vo farebných obrázkoch dá 90% hrán zistiť zo šedotónového obrazu, s použitím klasických metód ako Houghova transformácia alebo techniky hľadania hranice (Canny, Kirch, Sobel).

V SW realizácii bolo uskutočnené hľadanie hrán pomocou Sobelovho hranového detektora, ktorý vykazoval v porovnaní s ostatnými detekčnými mechanizmami (Canny, Prewitt) najlepšie výsledky. Výstupom segmentácie bol binarizovaný obraz, u ktorého každý pixel, ktorý patrí nejakému objektu (v našom prípade hrane), má priradenú hodnotu *1* a všetky ostatné pixely majú hodnotu *0*. Získaný binarizovaný obraz bol ďalej spracovávaný pomocou Houghovej transformácie [21].

3.1 Detekcia čiar pomocou modifikovanej Houghovej transformácie

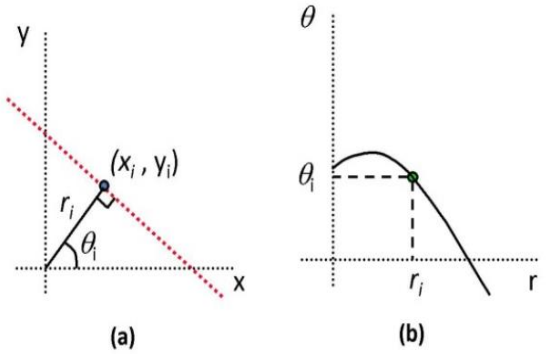
Jeden zo spôsobov detekcie čiar v obraze je použitie Houghovej transformácie. Originálna Houghova transformácia (HT) je segmentačná technika často používaná v prípadoch, keď je potrebné detegovať objekty so známym tvarom hranice (čiara, kružnica a pod.). Metóda je invariantná na otočenie, zmenu mierky atď. Navyše je necitlivá na šum a deformácie objektov. Pôvodne bola navrhnutá na detekciu rovných čiar. Aj keď originálna Houghova transformácia má rad výhod (napr. málo citlivá na šum, je necitlivá pri porušení hranice, je použiteľná i pri čiastočne zakrytých objektoch), bolo potrebné pre účely detekcie vodorovného značenia originálnu Houghovu transformáciu mierne modifikovať, keďže počas testovania a experimentov s HT bolo zaznamenaných niekoľko nedostatkov. Tieto nedostatky boli dodatočne upravené prostredníctvom podprogramov, čím vznikol modifikovaný Houghov algoritmus. Išlo hlavne o riešenie týchto čiastkových problémov [21]:

- Problém presnosti – vodorovné dopravné značenie tvorí v obraze pomerne širokú čiaru a algoritmus HT nájde niekoľko rovnobežných prípadne „skoro“ rovnobežných čiar, ktoré v Houghovom priestore vytvárajú niekoľko maxím. Navrhované riešenie: prostredníctvom algoritmu boli nájdené tie priamky, ktoré najlepšie reprezentujú jednu ľavú a jednu pravú vodorovnú čiaru.
- Problém skreslenia – vplyvom skreslenia sa priamka „zakriví“ a vo výsledku sa môže vyskytnúť niekoľko maxím a teda niekoľko priamok. Navrhované riešenie: Spoločné riešenie ako pri probléme presnosti.
- Problém začiatku a konca. Originálny algoritmus HT nerozlišuje začiatok a koniec kriviek. Tento problém však nebolo potrebné riešiť, pretože informácia o začiatku a konci čiar nie je relevantná v riešenej problematike.
- Problém použitia viacnásobných vnorených cyklov, ktoré sú súčasťou algoritmu HT, zvyšujú výpočtovú náročnosť. Preto je vhodné používať metódy na spresňovanie bodov záujmu. Navrhované riešenie: tento nedostatok bol riešený pomocou vytýčenia oblasti záujmu, v ktorej boli čiary detegované. Z celkovej získanej snímky bola spracovávaná iba časť s rozmermi 600x220 pixelov, v ktorej sa predpokladal najväčší výskyt detegovaných objektov.

Navrhnutý a realizovaný systém na vyhodnotenie prekročenia čiary, ktorý bol aplikovaný v SW realizácii je založený na využití Houghovej transformácie, pretože pri detekcii čiary bol pre zjednodušenie výpočtov zvolený lineárny model, v ktorom je možné opísať priamku v rovine aj v polárnych súradniciach (obr.5):

$$y = \frac{x.\cos\Theta}{\sin\Theta} + \frac{r}{\sin\Theta}, \qquad (1)$$

kde r predstavuje najmenšiu vzdialenosť priamky (v bode X [x, y]) od počiatku súradnicovej sústavy, Θ je veľkosť orientovaného uhla od kladnej x-ovej poloosi po polpriamku vedenú z počiatku súradnicovej sústavy kolmo na hľadanú priamku.

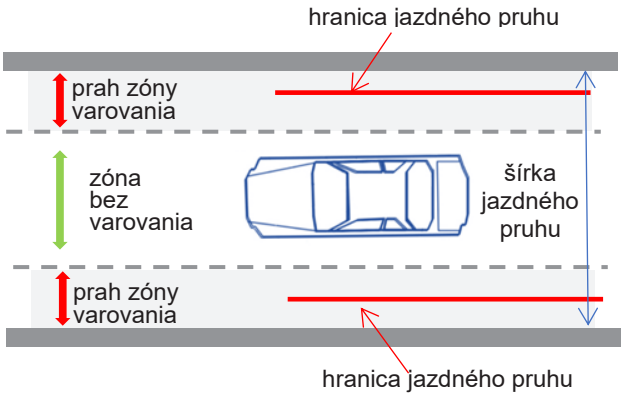


Obr.5 Houghova transformácia a) v kartézskych súradniciach, b) v polárnych súradniciach

Fig.5 Hough transformation a) Cartesian image space, b) polar Hough-space

3.2 Návrh konceptu umiestnenia zón prahu upozornenia v LDWS systéme

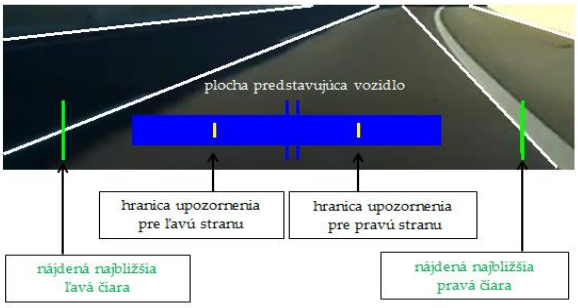
Pri softvérovej realizácii bolo vychádzané z medzinárodnej normy ISO 17361:2007, ktorá spresňuje definíciu systému, klasifikáciu, funkcie pre rozhranie človek-stroj a testovacie metódy systémov upozornenia pred neúmyselným opustením jazdného pruhu [22]. Systém LDWS považuje situáciu za bezpečnú, ak sa vozidlo pohybuje v okolí osi jazdného pruhu. Táto oblasť sa nazýva zóna bez upozornenia alebo bezpečná oblasť. Jazdný pruh je vymedzený vodorovným dopravným značením (pozri obr. 6). Okolo ohraničenia jazdného pruhu sa nachádza zóna varovania resp. upozornenia. Ak dôjde k vjazdu vozidla z bezpečnej zóny do zóny upozornenia, systém generuje upozornenie pre vodiča.



Obr. 6 Koncept prahu upozornenia a umiestnenie zón prahu upozornenia

Fig. 6 The concept of threshold notifications and location zone threshold alerts

Jazdný pruh a dráha vozidla sú snímané pomocou kamery. Z videosekvencie potom systém odhaduje pozíciu vozidla na vozovke a šírku jazdného pruhu. Detekcia prekročenia čiary v softvérovej realizácii funguje na princípe hľadania prieniku úsečiek. Pre každú snímku sa porovnáva poloha priesečníka najbližších nájdených čiar naľavo a napravo od vozidla a priamkou, ktorá predstavuje fiktívne vozidlo v snímke na obr. 7 sú tieto priesečníky znázornené zelenými zvislými čiarkami. Od nájdenia priesečníkov sa vypočíta pozícia stredu vozidla, resp. sú vypočítané dva stredy vozidla (stred ľavý, SL) a (stred pravý, SP), ktoré sú zobrazené dvomi modrými zvislými čiarkami. V prípade, že pri detekcii čiary dôjde k strate informácie o polohe vodorovného dopravného značenia na ceste, pamätá si systém 10 posledných hodnôt a poloha stredu sa vypočíta na základe nich.



Obr. 7 Princíp návrhu detekcie prekročenia čiary

Fig. 7 The principle of line crossing detection design

Na základe geometrie vozovky sa zo zosnímaných a definovaných údajov určia hranice zóny upozornenia vľavo (hranica upozornenia ľavá) a hranica zóny upozornenia vpravo (hranica upozornenia pravá). Na základe polohy stredov SL a SP fiktívneho vozidla a polohy hraníc upozornenia softvérová realizácia na základe geometrického modelu určuje možné prekročenie čiary [23]. V prípade prekročenia čiary je pre vodiča aktivované varovanie v podobe upozornenia, ktoré sa zobrazí na obrazovke v príslušnej snímke videosekvencie vo forme textu „ALARM >>“ alebo „<< ALARM“. Toto varovanie sa zobrazuje počas celého trvania vybočenia z jazdného pruhu, pozri obr.8. Upozornenie informuje vodiča, že je potrebná úprava trajektórie vozidla.

Pretože pri detekcii vodorovného dopravného značenia dochádza aj k situáciám, kedy príde v spracovávanej snímke k strate informácie o polohe čiary na ceste, navrhnutý algoritmus detekcie pracuje v dvoch častiach.

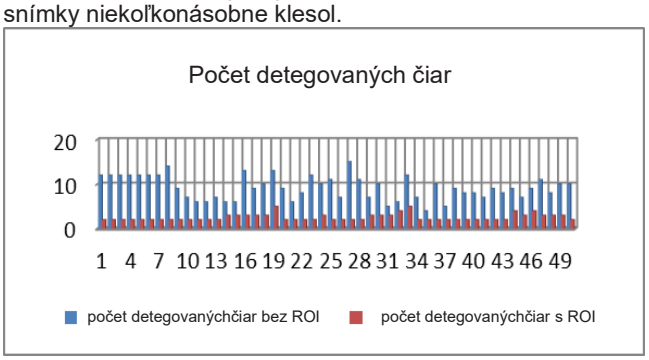
Obr. 8 Výsledok detekcie prekročenia čiary v smere doprava

Fig. 8 The result of line crossing detection in the right direction

V prvej časti sa porovnáva poloha oboch stredov (v prípade, že máme určené obe čiary) a ak nie sú nájdené obe čiary, porovnáva sa len jeden zo stredov vozidla s pozíciou hraníc zóny upozornenia [18].

4. Spoľahlivosť vyhodnotenia prekročenia čiary

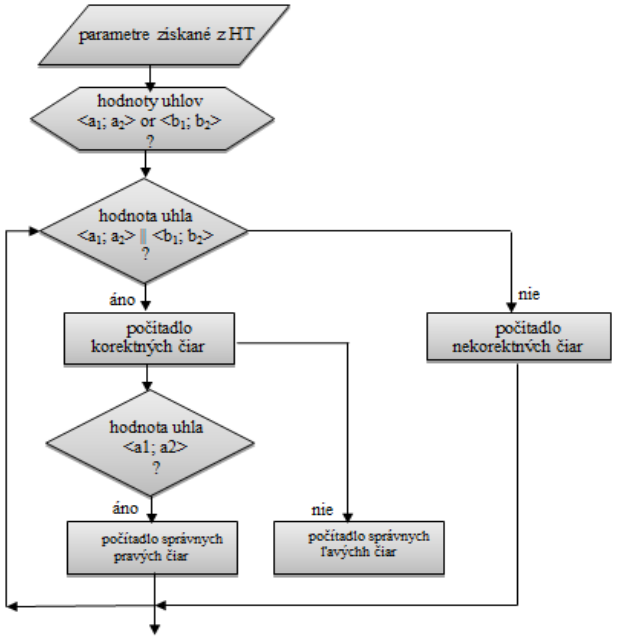
Pre algoritmy počítačového videnia je typická značná výpočtová náročnosť, pretože spracovávaná obrazová informácia je vo forme videozáznamu. Pri riešení úloh analýzy obrazu je potrebné hľadať spôsoby ako výpočty urýchliť. Dôležitým prvkom pri hľadaní čiar je nastavenie oblasti záujmu spracovania obrazovej informácie. Oblasť spracovávaných dát bola, ako už bolo spomenuté, redukovaná vhodným nastavením veľkosti oblasti záujmu (Region of Interest, ROI). ROI bol v SW aplikácii navrhnutý na základe geometrického modelu vozovky v dopravnej scéne. Pri segmentácii dopravnej scény sme vychádzali z faktu, že poznáme približnú polohu cesty a z toho, že vozovka bude v obraze začínať v jeho dolnej časti a smerom k horizontu sa bude zužovať. Tak bolo možné odhadnúť podiel plochy obrazu, ktorý obsahuje vozovku s vodorovným dopravným značením. Tiež rozmer a pozícia výrezu vzhľadom na celkový záber boli stále rovnaké. Veľkosť oblasti záujmu bola určená empiricky na základe testovania zosnímaných videosúborov. Na obr.9 sú v stĺpcovom grafe ilustračne znázornené počty nájdených čiar bez použitia ROI (240x480 pixelov) a s použitím ROI (65x412 pixelov). Je evidentné, že počet nájdených čiar pre tie isté porovnávané snímky niekoľkonásobne klesol.



Obr. 9 Počty nájdených čiar s použitím (240x480 pixelov) a bez použitia ROI (65x412 pixelov)

Fig. 9 Number of lines detected without using ROI (240x480 pixels), and using the ROI (65x412 pixels)

Základom varovania pred neúmyselným prekročením čiary je spoľahlivá detekcia čiar. Pri použití HT pri ich detekcii bolo potrebné uvažovať aj detekciu prípadných falošných čiar. Jedná sa o falošnú detekciu čiar získaných z objektov v obraze ako sú napr. zvodidlá prítomné na okraji cesty, vychodené koľaje na vozovke, horizont, stožiare osvetlenia a pod., teda miesta, kde v obraze dochádza k skokovej zmene jasu. Preto je potrebné pri procese detekcie čiar v obraze kategorizovať všetky nájdené čiary na správne, resp. nesprávne. Obrázok 10 ilustruje zjednodušený vývojový diagram, ktorý zobrazuje kategorizáciu čiar na „správne“ a „nesprávne“. Kategorizácia vychádza z nastavenia vhodnej tolerancie uhlov a ich porovnania s hodnotami uhlov u všetkých nájdených čiar.



Obr. 10 Zjednodušený vývojový diagram na kategorizáciu čiar

Fig. 10 Simplified flowchart for lines categorization

Nájdené čiary, ktoré sa nachádzajú v tesnej blízkosti čiar ohraničujúcimi jazdný pruh majú približne rovnaký uhol. Čiary s rovnakým alebo podobným uhlom, ako je uhol čiary na ceste, môžeme považovať za čiary nájdené správne. Algoritmus detekcie čiar v SW aplikácii uvažoval s rôznymi toleranciami uhlov. Vplyv nastavenia tolerancie uhla na počty nájdených správnych a nesprávnych čiar sú zobrazené v tab. 1.

číslo rámca tolerancie uhlov		1	2	3	4	5		44	45	46	47	48	49	50	spolu
<-50;-80> v <50;80>	správne	2	2	2	2	2	...	4	3	4	3	3	3	2	125
	nesprávne	0	0	0	0	0	...	0	0	0	0	0	0	0	0
<-70;-50> v <50;70>	správne	2	2	2	2	2	...	4	3	4	3	3	3	2	125
	nesprávne	0	0	0	0	0	...	0	0	0	0	0	0	0	0
<-50;-65> v <50;65>	správne	0	0	0	0	0	...	3	2	3	2	2	2	1	45
	nesprávne	2	2	2	2	2	...	1	1	1	1	1	1	1	80
<-50;-60> v <50;60>	správne	0	0	0	0	0	...	0	0	0	0	0	0	0	1
	nesprávne	2	2	2	2	2	...	4	3	4	3	3	3	2	124
<-55;-65> v <55;65>	správne	0	0	0	0	0	...	3	2	3	2	2	2	1	45
	nesprávne	2	2	2	2	2	...	1	1	1	1	1	1	1	80
<-60;-70> v <60;70>	správne	2	2	2	2	2	...	3	3	3	3	3	3	2	120
	nesprávne	0	0	0	0	0		1	0	1	0	0	0	0	5

Tab.1 Počty nájdených správnych a nesprávnych čiar pre rôzne nastavené tolerancie uhlov

Na obr. 11 sú znázornené detegované čiary. Tie čiary, ktoré boli vyhodnotené ako správne (tzn. patrili do stanovenej tolerancie uhlov) sú v obrázku označené zelenou farbou a čiary, ktoré nespĺňajú danú toleranciu sú označené červenou farbou.

Obr. 11 Príklad farebnej kategorizácie nájdených čiar pre konkrétnu snímku

Fig. 11 Example of colour categorization of found lines for a particular image

Pri následnom spracovaní v etape vyhodnotenia prekročenia čiary pracujeme už len s čiarami označenými ako „správne“. Aby sme v ďalšej etape spracovania obrazu zaručili spoľahlivú detekciu prípadného prekročenia čiary [21], [23], bolo potrebné zaistiť, že v každej spracovávanej snímke bude detegovaná minimálne jedna čiara, ktorá zodpovedá ľavému vodorovnému dopravnému značeniu a minimálne jedna čiara, ktorá zodpovedá pravému vodorovnému dopravnému značeniu. V reálnej nahrávke sa však vyskytujú medzery ako napr. v zosnímanej prerušovanej čiare.

Prítomnosť medzier v určitom čase spracovania môžu viesť k tomu, že v danej snímke (prípadne niekoľkých snímkach za sebou) chýba jedna z čiar, ktorá zodpovedá vodorovnému dopravnému značeniu. Tento problém bol riešený pomocou tzv. plávajúceho okna, veľkosť ktorého je možné v programovej realizácii meniť. Algoritmus „plávajúceho okna“ ešte pred začatím práce vlastného algoritmu detekcie čiar v zvolenom počte snímok vyhľadá snímku s maximálnou intenzitou a ten je následne použitý pre ďalšie spracovanie. Obrázok 12 ilustruje situáciu, na ktorej v analyzovanej snímke chýba ľavé dopravné značenie a stav snímky, ktorá bude analyzovaná po aplikácii plávajúceho okna s veľkosťou *W*=10 a *W*=15.

Obr. 12 Ilustrácia spracovávanej snímky bez (horný obrázok) a s použitím rôznych veľkostí plávajúcich okien

Fig. 12 Illustration processed images without (top figure) and using floating windows of different sizes floating windows

Z obr. 12 vyplýva, že zvýšenie spoľahlivosti detekcie čiar je možné ovplyvniť nastavením vhodne veľkého okna pre spracovanie vstupujúcich snímok.

Záver

Cieľom príspevku bolo opísať aplikáciu moderných techník počítačového videnia v číslicovom spracovaní obrazu so zameraním na dopravné procesy. Praktická realizácia a overenie výsledkov bola zameraná na nájdenie a overenie algoritmov umožňujúcich správnu detekciu vodorovného dopravného značenia v snímanej dopravnej scéne a ich kategorizáciu na „správne“ a „nesprávne“. Čiary kategorizované ako „správne“ boli následne použité v ďalšom kroku algoritmu, pre detekciu prípadného prekročenia čiary. Overenie správnosti a úspešnosti detekcie prekročenia čiar bolo vykonané na základe porovnania vopred určených hodnôt rámcov, u ktorých bola subjektívne detegované prekročenie čiary v skúmanom zázname dopravnej scény. Tieto údaje boli následne úspešne porovnané s hodnotami, ktoré boli určené na tom istom zázname pomocou vytvoreného algoritmu.

Ak je po detekcii prekročenia jazdného pruhu v procese ďalšieho spracovania využitý prenos upozornenia v rámci medzi-vozidlovej komunikácie v aplikáciách kooperatívnych inteligentných dopravných systémov, je potrebné následne riešiť okrem implementácie vhodných bezpečnostných mechanizmov pri prenose varovnej správy aj riešenie výpočtovej náročnosti. Proces realizácie digitálneho podpisu je obvykle založený na báze asymetrickej kryptografie a pre požadovanú bezpečnosť je to výpočtovo náročný problém. Hľadanie efektívnych schém digitálneho podpisu, u ktorých je zabezpečená garancia verifikácie prijatých správ v rámci komunikácie C2C je stále prioritnou výskumnou úlohou automobilových spoločností.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] EC initiatives eSafety (Mobility and Transport) [Online]. Available at: https://ec.europa.eu/transport/road_safety/specialist/knowledge/esave/ec_initiatives_on_esafety_en [Accessed 15 May 2019].

[2] BUBENÍKOVÁ, E., FRANEKOVÁ, M., HOLEČKO, P.: Security increasing trends in Intelligent Transportation Systems utilizing modern image processing methods, In: 13th International Scientific Conference, October 23-25 2013, Katowice, Ustroń, Poland, Berlin Heidelberg: Springer-Verlag, 2013, pp. 353-360 (Communications in computer and information science, 239. - ISSN1865-0929). Proceedings was published in an electronic version with ISBN 978-3-642-24660-9), (Springer Verlag, WoS), ISBN 978-3-642-41646-0.

[3] BUBENÍKOVÁ, E., PIRNÍK, R., HOLEČKO, P., FRANEKOVÁ, M.: The ways of streamlining digital image processing algorithms used for detection of lines in transport scenes video recording, In: PDES 2015: 13th IFAC and IEEE conference on Programmable devices and embedded systems: Cracow, Poland, 13-15 May 2015, [S. l.: Elsevier], SCOPUS, 2015, pp. 174-179, ISSN 2405-8963.

[4] HOFMANN, U., RIEDER, A., DICKMANNS, E.: Radar vision data function for hybrid adaptive cruise control on highways. Machine Vision and Applications, 14(1): 42-49, 2003.

[5] WIDMANN, G. R. et al.: Comparison of LIDAR based and radar based adaptive cruise control systems, Society, 109 (Part7): 126-139, 2000.

[6] HUANG, A. S. et al.: Finding multiple lanes in urban road networks with vision and LIDAR. In: Autonomous Robots, 26 (2): 103-122, 2009.

[7] PIRNÍK, R., HRUBOŠ, M., NEMEC, D., SVETLÍK, J. et al.: Integration of Inertial Sensor Data into Control of the Mobile platform, In: Advances in Intelligent Systems and Computing, 2016, pp. 271 - 282, (in English), ISBN 978-3-319-46534-0.

[8] Sensor Technologies for Intelligent Transportation Systems, [Online], Available at: https://www.ncbi.nlm.nih.gov/pubmed/29659524 [Accessed 15 May 2019].

[9] BUBENÍKOVÁ, E., FRANEKOVÁ, M., HOLEČKO, P.: Secure Solution of Collision Warning System Integration with Use of Vehicular Communications within Intelligent Transportation Systems. In: 12th IFAC Conference on Programmable Devices and Embedded Systems, Veľké Karlovice Czech Republic, September 25-27, 2013, pp. 78-83, ISSN:14746670, ISBN:978-390282353-3, and went out Proceedings of abstracts; pp. 159.

[10] LIENKAMP, M.: Intelligent, Connected Cars-Volkswagen's Vision of the Future, Electronics and Vehicle Research, VOLKSWAGEN. (2016, Jan. 25). [Online]. Available at: http://www.itu.int/dms_pub/itu-t/oth/06/1B/T061B0000020056PDFE.pdf [Accessed 15 May 2019].

[11] BUBENÍKOVÁ, E., FRANEKOVÁ, M., ĎURECH, J.: Security Solutions of Intelligent Transportation's Applications with using VANET Networks, In.: ICCC, In: 2014 15th International Carpathian Control Conference (ICCC), Veľké Karlovice 2014, Czech Republic, pp. 424-429, ISBN 978-1-4799-3528-4.

[12] ĎURECH, J: Security solution of VANET for control of intelligent transportation systems. Dissertation work, University of Žilina, Slovakia (in Slovak), 2016.

[13] ĎURECH, J., FRANEKOVÁ, M., HOLEČKO, P., BUBENÍKOVÁ, E.: Performance Analysis of Authentication Protocols Used Within Cooperative - Intelligent Transportation Systems with Focus on Security. 15th International Conference on Transport Systems Telematics (TST), Wroclaw, Poland, Apr 15-17, 2015. Selected paper In: Communications in Computer and Information Science Volume: 531, pp. 220-229, 2015.

[14] ROY, R. R.: Handbook of Mobile Ad Hoc Networks for Mobility Models, Springer, London, United Kingdom, 2011, ISBN: 978-1-4419-6048-1.

[15] Communication consortium Car 2 Car. (2016, Jan. 24) [Online]. Available at: https://www.car-2-car.org [Accessed 15 May 2019].

[16] C-roads – the platform of harmonised C-ITS deployment in Europe [Online]. Available at: https://www.c-roads.eu/platform.html [Accessed 15 May 2019].

[17] EVITA project. (2016, Jan. 23) [Online]. Available at: http://www.evita-project.org/ [Accessed 15 May 2019].

[18] HUH, K. et al.: Development of vision-based lane detection system considering configuration aspects. Optics and lasers in engineering, 43 (11): 1193-113, 2005.

[19] KOWSARSKI, T., BEAUCHEMIN, S. S., CHO, J.: Real-time vehicle detection and tracking using stereo vision and multi-view AdaBoost. In.: Intelligent Transportation Systems (ITSC), 2011, 14th International IEEE Conference, pp. 1255-1260, IEEE, 2011.

[20] SOTELO, A. M. et al.: A color vision based lane tracking system for autonomous driving on unmarked roads, Autonomous Robots, vol. 16 no. 1, pp. 95-116, October 2004.

[21] BUBENÍKOVÁ, E.: Detection of lines in applications of control within intelligent transport. Dissertation work, In: Slovak, University of Žilina, Slovakia, 2014.

[22] ISO 17361: 2007, Intelligent transport systems – Lane departure warning systems – Performance requirements and test procedures.

[23] BUBENÍKOVÁ, E., FRANEKOVÁ, M., HOLEČKO, P.: Evaluation of unwanted road marking crossing detection using real-traffic data for intelligent transportation systems. In: Telematics - support of transport: 14th international conference on Transport systems telematics, TST 2014: Katowice/Kraków/Ustroń, Poland, October 22-25, 2014: selected papers - Berlin: Springer-Verlag, 2014, pp. 137-145 (Communications in computer and information science, 471, ISSN 1865-0929), ISBN 978-3-662-45316-2.

Abstract
Modern anti-collision systems implemented within intelligent transport systems work on the principle of analysing different types of sensors. In many applications, processing of acquired image information from the environment around the vehicle is utilized. Lane Departure Warning Systems are also used in cooperative intelligent systems to control lane departure. A very important parameter of these systems is the reliability of the digital image processing method used, which depends on the choice of the optimal algorithm for detecting horizontal traffic signs. It is appropriate to set parameters of image processing algorithms based on SW simulation from real traffic data. The paper is focused on finding and testing effective methods of digital image processing in line crossing monitoring systems and their applicability in eventual inter-vehicle communication.

Ing. Emília Bubeníková, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 1
010 26 Žilina
Tel.: 0421425133344
emilia.bubenikova@fel.uniza.sk

MERANIE A VIZUALIZÁCIA DEFEKTOV V POVRCHU CESTNEJ VOZOVKY

Marián Hruboš, Dušan Nemec, Rastislav Pirník

Abstrakt
Predložený článok predstavenie výsledky v oblasti 3D merania a vizualizácie povrchu vozovky a následnej analýzy získaných modelov so zameraním na bezpečnosť premávky na daných vozovkách. Prezentovaná metóda umožňuje merať, generovať a následne analyzovať 3D model povrchu vozovky. Na základe analýzy modelu je možné určiť stav degradácie povrchu vozovky a poskytnúť základné údaje pre opravu vozovky. Metóda bola prakticky overená viacerými testovacími analýzami.

Kľúčové slová: 3D model, textúra, laserové skenovanie, degradácia

Úvod

Neustále sa zvyšujúca intenzita cestnej dopravy sa nepriaznivo prejavuje na kvalite komunikácii. Rôzne druhy porúch povrchu vozovky ako napr. trhliny, výtlky, pozdĺžne a priečne hrbole, zvlnenie povrchu, miestne poklesy alebo vyjazdené pozdĺžne koľaje nepriaznivo vplývajú na jazdný komfort a odzrkadľujú sa vyšším opotrebovaním niektorých častí motorových vozidiel. Tieto a iné deformácie vozovky a okolia tunela nepriaznivo ovplyvňujú bezpečnosť v okolí tunela, preto je nutné deformácie v pravidelných intervaloch analyzovať.

Pre meranie deformácií, ktoré sa vyskytujú na povrchu vozovky bolo vytvorených mnoho meracích metód a zariadení. Manuálne metódy merania vlastností povrchu v poslednej dobe nahrádzajú metódy založené na použití elektronických meracích zariadení. Jednou veľkou skupinou meracích zariadení, používaných na meranie deformácie vozovky, sú laserové meracie systémy. Tieto meracie systémy využívajú na meranie vzdialenosti, medzi laserovým meracím zariadením a samotnou vozovkou, bezkontaktnú meraciu metódu založenú na meraní doby letu laserového impulzu.

Laserové snímanie okolitého priestoru si už dlhšiu dobu nachádza uplatnenie vo viacerých odvetviach priemyslu a mnohé autorské kolektívy pracujú na spracovaní dát z týchto typov snímačov. Jednou veľkou skupinou snímačov používaných na vytvorenie 3D modelu priestoru sú terestriálne alebo taktiež 3D skenery. V mnohých prípadoch nie je možné využiť 3D skenery, vtedy je vhodné použiť 2D skenery. Pre vytvorenie 3D modelu priestoru rozľahlých vonkajších priestorov je nutné do výpočtu mračna bodov zahrnúť aj polohu merania.

Náš výskum zameriavame hlavne na cestnú a železničnú dopravu. V týchto odvetviach nie je možné zastaviť dopravný tok na niekoľko hodín kvôli zdĺhavému meraniu využitím 3D skenera. Preto sme sa rozhodli naše zariadenie zostaviť na základe 2D skenera, pričom výsledný 3D model povrchu vozovky v podobe mračna bodov získavame pomocou fúzie dát z laserového skenera, GPS prijímača a INS. Následne po spracovaní mračna bodov sme schopný na vygenerovaný povrch zmeraného priestoru aplikovať textúru, ktorá je automaticky generovaná z kamerových záznamov. Výsledný model je následne uložený vo formáte *.obj pričom sa skladá z jednotlivých rezov prostredím čo následne využívame na analýzu prostredia. [1]

1. Základné vlastnosti povrchu vozovky

Jednou zo základných vlastností povrchu je jeho drsnosť. Na meranie drsnosti povrchu a nerovností na ňom bolo vytvorených mnoho meracích metód a zariadení. Drsnosť, ako vlastnosť povrchu vozovky, zabezpečuje spolupôsobenie medzi vozovkou a kolesami. Drsnosť, z geometrického hľadiska, vyjadruje textúru povrchu resp. usporiadanie jednotlivých zŕn kameniva na povrchu tzv. makrotextúra a usporiadanie výbežkov a nerovností na povrchu zŕn tzv. mikrotextúra. Drsnosť sa dá definovať ako vlastnosť povrchu charakterizovaná odporom voči pošmyknutiu kolies vozidla na povrchu. Ide o tangenciálnu reakciu povrchu, slúžiacu k umožneniu prenosu hnacej resp. brzdnej sily z kolesa. Táto reakcia sa taktiež nazýva šmykové trenie, pričom súčiniteľ šmykového trenia je definovaný ako pomer šmykového trenia k normálovému zaťaženiu kolesa. Veľmi náročným problémom sa stáva zhodnotenie vozovky z hľadiska odolnosti voči šmyku.

Problém zhodnotenia vozovky vyplýva z veľkého počtu premenných, ovplyvňujúcich trenie medzi kolesom a vozovkou ako napr. profil a hĺbka dezénu pneumatiky, typ pneumatiky, tlak v pneumatike, typ a hmotnosť vozidla, rýchlosť vozidla, tlmenie a brzdný systém vozidla, rozloženie hmotnosti na vozidle, ročné obdobie, teplota, prítomnosť a hĺbka vodného filmu na vozovke, znečistenie vozovky, geometrické vedenie komunikácie, intenzita dopravy, vek vozovky, porušenie obrusnej vrstvy, druh kameniva a pod. Stanovenie miery vplyvu jednotlivých premenných na trenie medzi vozovkou a kolesom je veľmi náročné, z dôvodu nemožnosti separácie jednotlivých premenných a nemožnosti eliminácie ich vplyvov stanovením špeciálnych podmienok.

Treba však podotknúť, že najzásadnejší vplyv na trenie má povrchová textúra spolu s typom pneumatiky, jej stavom, rýchlosťou vozidla a prítomnosťou vody. Pre hodnotenie drsnosti povrchu je potrebné zaoberať sa problematikou textúry povrchu. Textúra sa delí podľa amplitúdy a vlnovej dĺžky nerovnosti na:

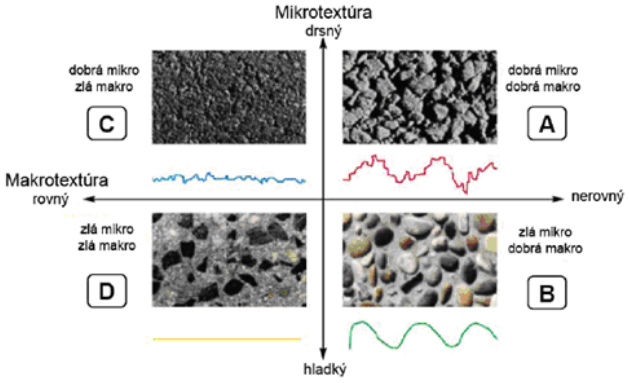
- mikrotextúry,

- makrotextúry,
- megatextúry.

Megatextúry tvoria irelevantnú zložku prvkov, vplývajúcich na drsnosť povrchu vozovky. Megatextúry pôsobia na komfort jazdy a opotrebovávanie častí vozidla. V interakcii kolies s vozovkou spôsobujú zníženie prítlaku kolesa. Vplyv makrotextúry sa najviac prejavuje na vlhkej vozovke pri strednej a vysokej rýchlosti vozidla. Jej hlavnou úlohou je odvod vody z povrchu vozovky, čím sa predchádza vzniku aquaplaningu.

Mikrotextúry zabezpečujú adhéziu t.j. úroveň trenia na mikroskopickej úrovni. Mikrotextúri majú najväčší vplyv na trenie pri nízkej rýchlosti vozidla a pri suchej vozovke. Pre vytvorenie optimálnych protišmykových vlastností povrchu vozovky je nutné, aby hodnoty mikrotextúry a makrotextúry boli za akýchkoľvek podmienok na primeranej úrovni.

Povrch vozovky môže byť začlenený do štyroch kategórií na základe hodnôt mikrotextúry a makrotextúry (obr. 1):
- A – drsný a nerovný , povrch má dobrú makrotextúrua aj mikrotextúru,
- B – vyhladený a nerovný, povrch má zlú mikrotextúru a dobrú makrotextúru,
- C – drsný a rovný, povrch má zlú makrotextúru a dobrú mikrotextúru,
- D – vyhladený a rovný, povrch má zlú makrotextúru aj mikrotextúru.



Obr. 1 Rôzne druhy povrchu v závislosti od mikrotextúri a makrotextúri [1]

Fig. 1 Categories of the road surface [1]

Pojem „nerovný“ povrch definuje makrotextúru obsahujúcu nerovnosti, ktorých priemerná hĺbka je väčšia ako 1,0 mm. „Drsným“ povrchom je považovaný povrch s priemernou hĺbkou mikrotextúry 50 μm.

Pre presnejšie definovanie vlastností profilu povrchu, je nutné povrch definovať nielen určením priemernej hĺbky nerovností, ale taktiež hustotu resp. distribúciu nerovností, ktorá je opísaná pomocou vlnovej dĺžky nerovnosti, t.j. vzdialenosť priľahlých výstupkov nerovností. Najlepšie hodnoty drsnosti povrchu sa dosahujú, ak je vlnová dĺžka 3 až 20 krát väčšia ako priemerná hĺbka nerovností. Z definícií súčiniteľa trenia a vlnovej dĺžky vyplýva, že hodnota drsnosti sa so zvyšujúcimi hodnotami mikrotextúry a makrotextúry zvyšuje a zároveň sa zvyšuje so znižujúcou sa hodnotou vlnovej dĺžky.

Všeobecne sa dá stanoviť, že zásadný vplyv na veľkosť trenia na styku vozovky s kolesom, má hodnota textúry. Pri jej definovaní je však okrem priemernej hĺbky a vlnovej dĺžky potrebné brať do úvahy morfológiu povrchu resp. rozloženie nerovností. Rozloženie nerovností sa delí na rozloženie s [1]:
- „pozitívnou“ textúrou (obr. 2 a),
- „negatívnou“ textúrou (obr. 2 b).

Obr. 2 Druhy textúr povrchu vozovky [1]

Fig. 2 Types of the surface texture [1]

Tieto a ďalšie vlastnosti povrchu vozovky je možné merať manuálnymi, alebo automatickými metódami. Medzi manuálne metódy patria:
- stanovenie drsnosti odmernou metódou,
- stanovenie drsnosti kyvadlom,
- meranie pomocou nivelačného prístroja,
- meranie pomocou laty na meranie nerovností,
- meranie pomocou totálnej stanice.

Medzi automatické metódy patria:
- meranie pomocou zariadenia VIDEOCAR,
- meranie pomocou zariadenia Profilograph GE.

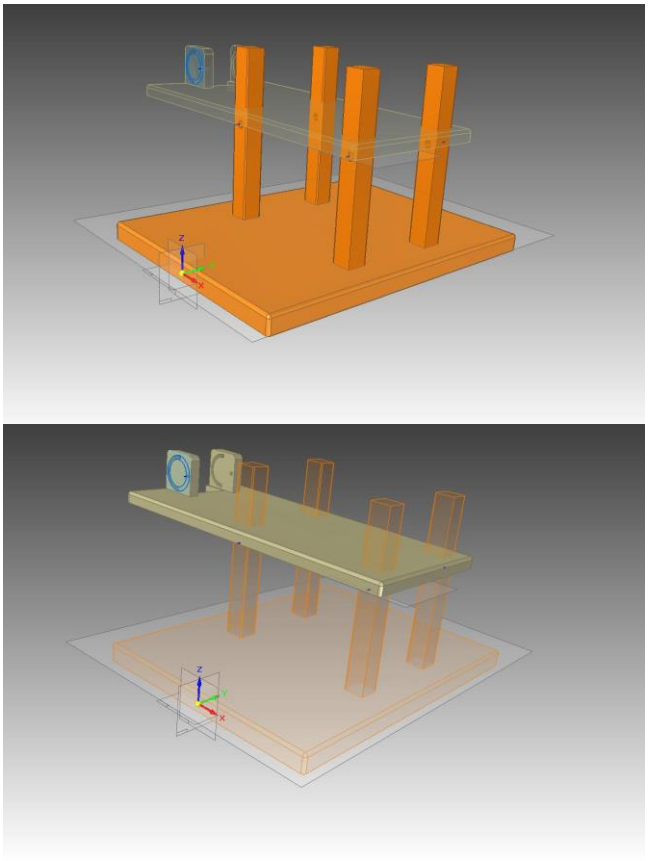
2. KONCEPCIA MERACIEHO ZARIADENIA

Základné požiadavky pre meranie povrchu vozovky sú:
- Mobilné meranie - požadujeme merať povrch vozovky v pohybujúcom sa dopravnom prúde, meranie aj vo vyšších rýchlostiach.
- Vhodné umiestnenie skenera – v súlade s legislatívou.
- Univerzálnosť - meranie by malo byť realizovateľné na rôznych platformách.
- Napájanie - nutnosť vyriešiť zdroj napájania pre skener - 24V DC.

Pri splnení všetkých týchto požiadaviek by malo byť meranie realizovateľné a opakovateľné. Preto bolo nutné poznať čo najviac detailov o meracom prístroji. Keďže má byť koncepcia vhodná pre mobilné merania, musíme ju vhodne umiestniť na vozidlo. Predpokladáme meranie na reálnych komunikáciách, teda vozidlo musí vyhovovať technickým predpisom. Pre pohyb meracej platformy využijeme automobil, ten spĺňa technické predpisy a je jednoducho dostupný.

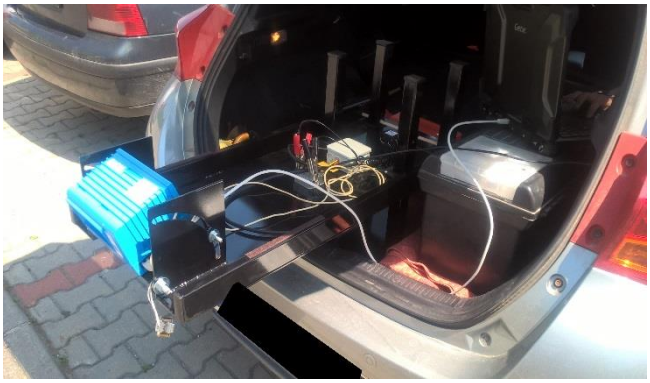
Bežný automobil má v kufri nakladaciu hranu. Je nutné túto hranu prekonať vhodnou konštrukciou so skenerom. Cestná vyhláška povoľuje presah nákladu vozidla o 40cm od hrany vozidla bez nutnosti používať označenie presahujúceho nákladu. Z dôvodu univerzálnosti je nutná realizovať aj nastavenie sklonu, pod ktorým skener meria. Z tohto dôvodu bude najlepšie použiť polohovateľný statív (obr. 3.), alebo obdobným spôsobom zabezpečiť vertikálnu manipuláciu.

Pre samotné meranie bol použitý skener LMS 400 firmy SICK. Skener LMS 400 je merací optoelektronický systém na meranie vzdialeností v 2D priestore s rozsahom 70°. Je to aktívny systém s červeným laserom (λ=650nm), s výstupným výkonom laseru 7,5mW, ktorý je zaradený do 2. triedy laserov (pre oko nebezpečný, pokožku bezpečný). MLS využíva ToF metódu na princípe fázového posunu (sínusová modulácia). Čas šírenia svetla a vlnová dĺžka sú použité na určenie fázového posunu medzi lúčom odoslaným a prijatým. Fázový posun je konvertovaný na frekvenciu, na základe ktorej dokáže systém určiť vzdialenosť. Skenovacia frekvencia systému je od 360 do 500Hz a uhlové rozlíšenie v rozsahu 0.1333° - 1°. Chyba merania pri tomto systéme je ±4mm. Tá sa môže líšiť v závislosti od odrazivosti meraného povrchu a vzdialenosti. Výrobcom udávané hraničné hodnoty sú 9mm chyby merania pre 6.5% odrazivosť a 3mm pre 100%. Vzdialenosť v rámci pracovného rozsahu je uvedená v rozmedzí 0,7– 3 m [2].



Obr. 3 Koncepcia konštrukcie pre uchytenie skenera [4]

Fig. 3 Virtual model of the stand for the laser scanner [4]



Obr. 4 Praktické meranie [4]

Fig. 4 Practical measurement setup [4]

Pre spracovanie dát z laserového skenera LMS 400 je možné využiť viacero prístupov. Či už online alebo offline spracovanie. Spracovanie dát použitím TCP/IP pripojenia na skener, alebo spracovanie dát zo súboru, do ktorého boli dáta uložené počas merania.

Pre tvorbu 3D modelu, generovaného z dát získaných z laserového skenera LMS 400, sme sa rozhodli použiť offline spracovanie dát zo súboru, v ktorom sú uložené jednotlivé merania. Zaznamenávanie merania pomocou programu SOPAS môže prebiehať dvoma spôsobmi. Prvý spôsob umožňuje zaznamenávať celú komunikáciu laserového skenera s aplikáciou a druhý umožňuje ukladať iba jednotlivé namerané hodnoty. Druhý spôsob je z pohľadu spracovania dát jednoduchší, lebo jeden záznam v súbore obsahuje priamo jednotlivé merania, ktorých počet je závislý od nastaveného uhlového rozlíšenia, pričom vzdialenosť je uložená priamo v milimetroch. Príklad:

„1250;1245;1245;1241;1238;1233;1229;1226;1224;1217;1217;1213;1207;1205;1204;1198;1197;1192;1191;1188;1185;1181; ..... ;1238“.

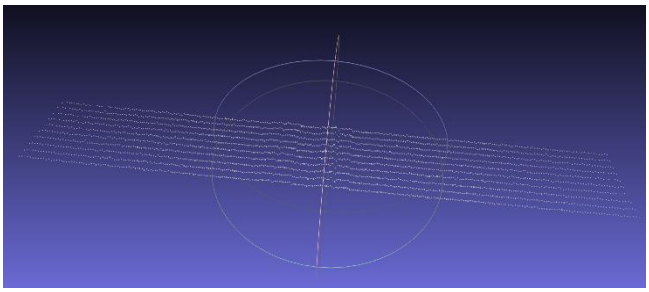
Nevýhodou tohto riešenia je nemožnosť kontroly. Prvý spôsob umožňuje kontrolu pomocou časovej pečiatky, ktorá je umiestnená na začiatku každej prijatej správy. Taktiež je možné prenos zabezpečiť kontrolným súčtom realizovaným funkciou XOR. Nevýhodou však je, že dáta sú uložené v hexadecimálnej sústave, takže pred samotným výpočtom polohy jednotlivých bodov mračna bodov je nutné zdrojové dáta predspracovať [3]. Príklad správy:

„RECEIVE 12:47:56.796 - SI A 425C0000 3E800000 0118 04ED 04E1 04E6 04D1 04CF 04E0 04C7 04CC 04C6 04C2 ..... 04D3“

Do algoritmu je zapracované aj vytváranie metadát pre tvorbu plôch objektu, pričom ak došlo k nesprávnemu zmeraniu bodu (zapríčineného napríklad zlým odrazom), tento bod nebude braný do úvahy pri výpočte plôch objektu a jeho hodnota bude rovnaká ako u predchádzajúceho bodu. Softvérovo je táto časť riešená nasledovne:

```
d = fscanf(fileIN, '%d',1);
pom = fscanf(fileIN, '%c',1);
if (d == 0)
    x=(cos((a+(j-1)*difa)*pi()/180)*(c/1000));
    y=(sin((a+(j-1)*difa)*pi()/180)*c/1000);
    fprintf(fileOUT, 'v %d %d %d\n',x, y,i*(20/1000));
    A((i-1)*N+j,1)=(i-1)*N+j;
    A((i-1)*N+j,2)=0;
else
    x=(cos((a+(j-1)*difa)*pi()/180)*(d/1000));
    y=-(sin((a+(j-1)*difa)*pi()/180)*(d/1000));
    fprintf(fileOUT, 'v %d %d %d\n',x, y,i*(20/1000));
    A((i-1)*N+j,1)=(i-1)*N+j;
    A((i-1)*N+j,2)=1;
    c=d;
end;
```

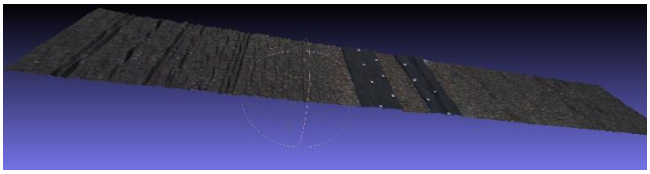
Výstupom tejto časti programu je model v podobe mračna bodov (obr. 5.).



Obr. 5 Praktické meranie 3D model – zameranie vytyčovacieho prvku

Fig. 5 Example of the processed cloud of points vizualized in 3D space

Na takýto model je možné aplikovať algoritmus na tvorbu plôch a textúr výsledkom je 3D model s príslušnou textúrou (obr. 6.).



Obr. 6 Praktické meranie 3D model s textúrou – zameranie vytyčovacieho prvku

Fig. 6 Example of the measured 3D model with applied visual texture

Záver

Z nášho pohľadu má táto práca potenciálny prínos pri údržbe vozoviek. Systém dokáže zaznamenávať profil vozovky aj pri vyšších rýchlostiach. Ak by sa výstupy využívali pri údržbe vozoviek, mohlo by to mať v konečnom dôsledku vplyv na priepustnosť dopravy po komunikácii, na zvýšenie bezpečnosti komunikácii a v konečnom dôsledku aj na životné prostredie. Vďaka ekonomickejšej jazde by autá mohli produkovať menej emisií, vodičom by ušetrili kvalitnejšie cesty peniaze za pohonné hmoty a súčasne aj čas strávený cestovaním. Pre SSC by mohlo byť meranie prospešné hlavne z dôvodu zistenia, ktoré úseky sú v nevyhovujúcom stave na prevádzku. V prípade pravidelného merania povrchu dokážeme z nameraných dát určiť aj kedy dochádza k opotrebovaniu komunikácie, poprípade pri rekonštrukcii použiť inú, vhodnejšiu zmes.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] HRUBOŠ, M.: Fúzia senzorických dát získaných prostredníctvom mobilnej meracej platformy, dizertačná práca 28260020153127

[2] HRUBOŠ, M., NEMEC, D., PIRNÍK, R., JANOTA, A., BUBENÍKOVÁ, E., NAGY, P.: Mobile two-dimensional laser scanner used for clasification of objects in the urban area, ACTA TECHNICA CORVINIENSIS – BULLETIN OF ENGINEERING, FASCICULE 2, ISSN: 2067 – 3809

[3] HRUBOŠ, M., SVETLÍK, J., NIKITIN, Y., PIRNÍK, R., NEMEC, D., ŠIMÁK, V., JANOTA, A., HRBČEK, J., GREGOR, M.: Searching for collisions between mobile robot and environment, INTERNATIONAL JOURNAL OF ADVANCED ROBOTIC SYSTEMS, DOI: 10.1177/1729881416667500

[4] HRUBOŠ, M., PIRNÍK, R., NEMEC, D.: Automatizovaný systém merania defektov vozovky v okolí tunela, ARTEP 2018, 7.2.-.9.2. 2019 Stará Lesná

Abstract

This article presents research results in the area of 3D scaning of the tunnel roadway surface and its surroundings. Then it describes analysis of the obtained models with respect to the safety of the road trafic inside tunnel. Presented method allows to create and analyze 3D model of the roadway surface. From the analysis of the model, it is possible to determine the degradation degree of the roadway and provide basic info required for its repair. Method has been verified by several testing analysis.

Ing. Marián Hruboš, PhD.
Ing. Dušan Nemec, PhD.
doc. Ing. Rastislav Pirník, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 8215/1
010 26 Žilina
+421 41 513 3301
{marian.hrubos;dusan.nemec;rastislav.pirnik}@fel.uniza.sk

RIADENIE VSTUPU VOZIDIEL DO DIAĽNIČNEJ SIETE

Aleš Janota, Jozef Hrbček

Abstrakt
Pri zvyšovaní objemov dopravy dosiahnu diaľničné siete svoje hraničné možnosti v okamihu, kedy už nebude možné extenzívne pridávať nové jazdné pruhy. Eliminovať alebo aspoň zmierniť následky vzniknutých kongescií možno nasadením pokročilých inteligentných dopravných systémov, konkrétne systémov regulujúcich vstup vozidiel na diaľnicu (známych ako *ramp metering*). Článok ukazuje, ako by bolo možné aplikovať dve vybrané stratégie riadenia (algoritmy ZONE a ALINEA) v zjednodušených podmienkach slovenskej diaľničnej siete. Výsledky modelovania a simulácií v prostredí PTV Vissim demonštrujú vlastnosti obidvoch algoritmov, a to prostredníctvom porovnania ich výsledkov navzájom ako aj porovnania výsledkov so situáciou bez akýchkoľvek regulačných opatrení.

Kľúčové slová: Zone, ALINEA, algoritmus, model, simulácia, ramp metering

Úvod

Počet osôb žijúcich vo veľkých metropolitných oblastiach trvalo narastá, čo so sebou prináša problémy so zaistením mobility ich trvalých či dočasných obyvateľov. Rozhodnutia o spôsobe, čase a mieste pohybu jednotlivcov spoluvytvárajú a ovplyvňujú prostredie, v ktorom žijeme. Pretože však vo väčších aglomeráciách uskutočňuje takéto rozhodnutie súčasne vyšší počet jedincov, musí byť celý proces organizovaný, aby sa dal realizovať a zaistila sa jeho bezpečnosť, efektívnosť, eliminovali sa dopady na životné prostredie, a pod. Postupné a nezadržateľné nasadzovanie informačných a komunikačných technológií predznamenáva prechod k tzv. „inteligentnej mobilite“, podmienenej prepojením fyzikálneho a digitálneho sveta. Veľké metropolitné oblasti sú prepojené sieťou diaľnic a železníc, pretože tieto druhy dopravy umožňujú dosahovať najvyššiu rýchlosť presunu osôb či tovaru. Problémom je, že diaľnice a ich parametre sú často limitované mestskou zástavbou, pretože urbanistické plánovanie v minulosti nerátalo s tak masívnou expanziou v budúcnosti. Nie je preto možné ďalej zvyšovať objem prevádzky výstavbou nových jazdných pruhov či rozširovaním súčasných, pretože už často nie je pre tento účel k dispozícii žiadny priestor. Jedným z riešení je použitie pokročilých inteligentných systémov (systémov dopravnej telematiky), ktoré môžu prispieť k zvýšeniu intenzity dopravných prúdov a tým plynulosti dopravy a vyššej efektívnosti využívania diaľničnej siete. Navrhované sú rôzne stratégie na zaistenie rovnováhy medzi počtom vozidiel vstupujúcich do diaľničnej siete a jej kapacitou. Tento článok predstavuje niektoré z nich, ich princíp, fungovanie ako aj benefity, ktoré od ich zavedenia možno očakávať.

1. Prehľad riadiacich stratégií

Za historicky prvý zdokumentovaný pokus o riadený vstup vozidiel na diaľnicu sa považuje rok 1963, kedy na chicagskej tzv. Eisenhowerovej diaľnici bolo vyskúšané manuálne ovládanie vjazdu vozidiel človekom. V roku 1967 nasledovali pokusy s automatickým uzatváraním príjazdovej komunikácie – tzv. *ramp closure* (v Detroite a L.A), v roku 1972 sa v Minneapolise začali objavovať riadené jazdné pruhy vyhradené pre autobusy, počet inštalácií začal postupne rásť, začali sa testovať preferenčné jazdné pruhy vyhradené pre plne obsadené vozidlá (tzv. *HOV – High Occupancy Vehicles*), a pod. [1]. V rámci Európskej únie sú uvedené prístupy predmetom výskumu a harmonizácie najmä od roku 2004 vďaka projektu EURAMP [2], ktorý riešil otázky regulácie vedľajšieho dopravného prúdu pripájajúceho sa k hlavnému dopravnému prúdu najmä počas dopravnej špičky a existencie dopravných kongescií.

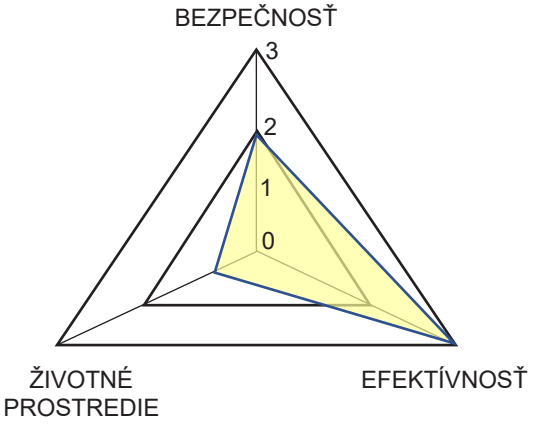
Vo všeobecnosti možno zhrnúť používané prístupy do 4 kategórií [3]:

- zatváranie vjazdu (tzv. *ramp* closure) – uzatvorenie môže byť dočasné, prerušované alebo trvalé;
- dávkovanie vozidiel (tzv. *ramp metering*) – rýchlosť vchádzajúcich vozidiel môže byť riadená dopravným návestidlom na základe monitorovania stávajúceho stavu, podľa potreby aj manuálne; niekedy býva tento prístup nazývaný aj ako “dávkovacia signalizácia” [4];
- riadenie vstupu špeciálnej triedy vozidiel: autobusové jazdné pruhy, pruhy pre plne obsadené vozidlá a pod.;
- špeciálne úpravy pripájacích terminálov.

K očakávaným benefitom uvedených opatrení patrí nepochybne vyššia bezpečnosť v mieste spájania sa komunikácie nižšej a vyššej triedy, kde možno predchádzať vzniku kolízií preventívnym rozpúšťaním kongescií. Zvyšuje sa mobilita vďaka zvýšeniu priepustnosti komunikácií zlepšením ich prevádzkových charakteristík (rýchlosti, cestovného času, meškania). Vyššia plynulosť premávky znamená menšiu spotrebu palív, ekonomické prínosy, znížené objemy emisií uvoľnené do ovzdušia a tým pozitívny dopad na životné prostredie. V neposlednom rade je prínosom aj pozitívne vnímanie stavu motoristickou verejnosťou a následne jej väčšia ochota súhlasiť s vynakladaním verejných prostriedkov na rozvoj cestnej infraštruktúry, jej prevádzku a údržbu. Celkový vplyv týchto technických opatrení býva znázorňovaný v súlade s grafom na obr. 1. Kriticky však treba poznamenať, že existujú isté obmedzenia – pri vzniku určitých typov kongescií

sa očakávaný prínos nemusí vôbec dostaviť. Typicky ide o situácie, kedy:

- intenzita regulovaného dopravného prúdu je v porovnaní s prúdom na hlavnej komunikácii veľmi nízka;
- vysoké dopravné prúdy sú realizované na šmykľavom povrchu vozovky;
- existujú priestorové obmedzenia na miestnej komunikácii, kde by mohli vozidlá čakať;
- kapacita diaľnice je vysoko prekročená (vznik lieviku).

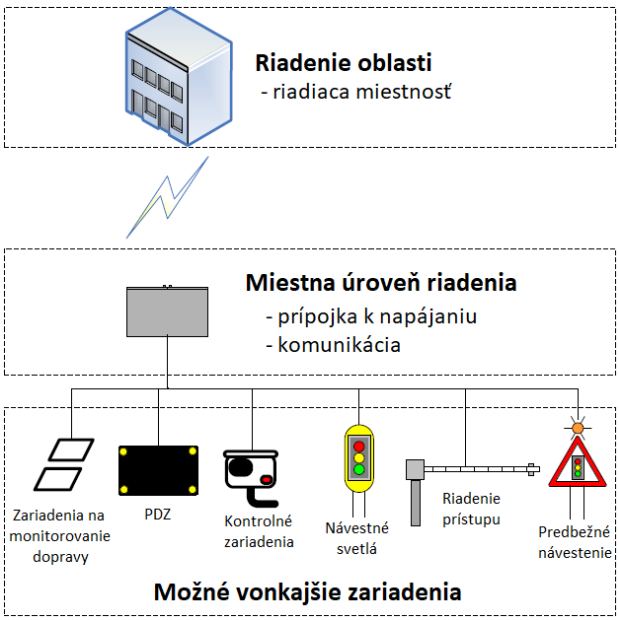


Obr. 1 Benefity očakávané po zavedení služby [5]

Fig. 1 Benefits expected after the service introduction [5]

1.1 Harmonizácia

V rámci dlhodobej snahy EÚ harmonizovať a dosahovať celoeurópske technické riešenia výnimkou nie je ani oblasť inteligentných dopravných systémov, v tomto konkrétnom prípade harmonizácia regulovania vstupu vozidiel na diaľnicu. Za jeden zo zásadných dokumentov možno považovať smernicu [5], ktorá bola pôvodne vypracovaná v rámci európskeho projektu EasyWay a neskôr aktualizovaná v rámci projektov európskej platformy pre inteligentné dopravné systémy EIP/EIP+.



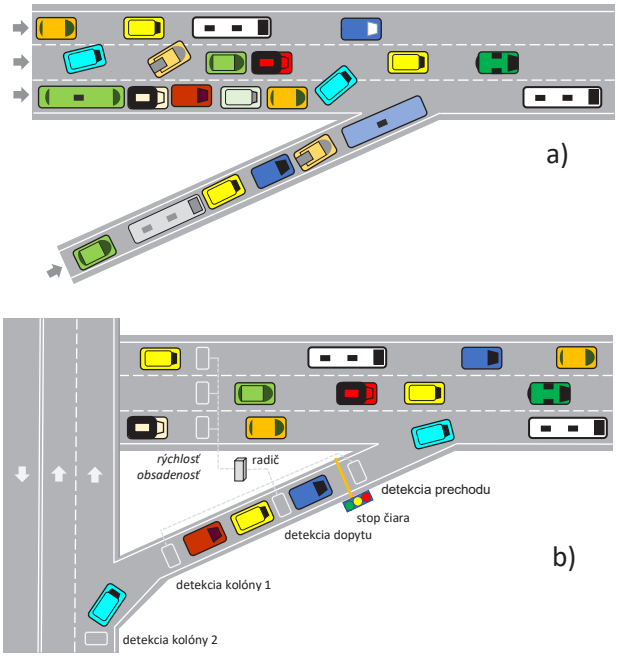
Obr. 2 Systémová architektúra [2]

Fig. 2 System architecture [2]

Z hľadiska požiadaviek na informačné a komunikačné technológie sa odporúča 3-úrovňová systémová architektúra podľa obr. 2 [5], pozostávajúca z nasledujúcich komponentov:

- *vonkajšie zariadenia*: detektory, videokamery, premenné dopravné značky (PDZ), informačné tabule, trvale svietiace dopravné značky, závory, svetelné návestidlá, a pod.; rozmiestnenie napr. podľa obr. 3b;
- *miestne riadenie*: radič na miestnej úrovni pripojený k vstupom/výstupom digitálnych vonkajších zariadení, k napájaniu a komunikačnému systému;
- *riadenie oblasti*: riadiace sub-centrum (podriadený dispečing), ktoré je hierarchicky členené a okrem radičov na miestnej úrovni môže byť pripojené k riadiacemu centru (hlavnému dispečingu) vybavenému počítačmi, GUI, atď.

Celá architektúra by mala fungovať na princípe „Plug & Play“, čo predpokladá zabezpečenie technickej interoperability. Implementácia služby si vyžaduje použitie európskeho štandardu DATEX II na výmenu dopravných informácií s využitím XML a UML.



Obr. 3 Situácia bez regulovania vjazdu (a) a po zavedení tzv. ramp meteringu (b)

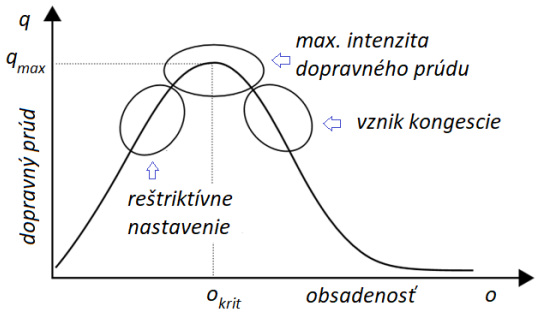
Fig. 3 Situation without regulation (a) and after introduction of so-called ramp metering (b)

Radiče na miestnej úrovni sa typicky nachádzajú v niektorom z nasledujúcich stavov:

- pohotovostný stav (*standby mode*) – návestné svetlá na príjazdovej rampe sú vypnuté;
- uvádzanie do prevádzky (*switching on*) – systém sa zapína;
- normálna prevádzka (*steady state*) – stav normálneho fungovania;
- rozpúšťanie kolóny (*queue override*) – stav, kedy sa systém snaží predchádzať kongescii v miestnej cestnej sieti; cieľom je dosiahnuť vyššiu priepustnosť;
- vypínanie systému (*switching off*);
- bezpečný režim (*fail-safe mode*) – predchádza alebo zmierňuje nie bezpečné dôsledky poruchy systému; v závislosti od situácie môže byť tento stav stavom vypínania systému (*switching off*) alebo stavom, kedy sa riadenie uskutočňuje na základe pevných časových intervalov (*fixed-time control*).

Aktivácia a zmeny vyššie uvedených stavov sa dejú pomocou vhodného riadiaceho algoritmu. Výstupy európskych projektov striktne nepredpisujú, ktorý riadiaci algoritmus treba

použiť pre daný dopravný scenár. Jeho činnosť by ale mala smerovať k priebežnému rozpúšťaniu kolón, ktoré majú tendenciu sa vytvárať najmä počas dopravných špičiek. Úloha nie je triviálna - treba nájsť optimálne nastavenie, ktoré by na jednej strane nebolo príliš reštriktívne a na druhej strane nevytváralo zbytočne kolóny na príjazdovej komunikácii. Na obr. 4 by takému nastaveniu zodpovedala maximálna hodnota intenzity dopravného prúdu q_{max} [voz/hod] pri tzv. kritickej obsadenosti o_{krit} [-].



Obr. 4 Závislosť intenzity dopravného prúdu q na obsadenosti o pre rôzne nastavenia ramp meteringu

Fig. 4 Depencency of traffic flow intensity q on occupancy o for various ramp metering settings

1.2 Klasifikácia

Systémy regulujúce vstup vozidiel na diaľnicu možno klasifikovať podľa rôznych kritérií. Podľa úrovne prevádzky na:

- *miestne* – riadenie na báze pevných časových intervalov alebo dopravne závislé (využívajúce informácie od dopravných detektorov);
- *systémové* (plošné) – riadenie dlhšieho úseku diaľnice (tzv. koordinovaná diaľnica) alebo riadenie väčšej oblasti (integrácia miestnych a hlavných komunikácií).

Podľa konfigurácie na:

- *jedno-pruhové* (bez alebo s detekciou plného obsadenia vozidiel);
- *viac-pruhové* (bez alebo s detekciou plného obsadenia vozidiel).

Podľa geografického určenia (lokalizácie):

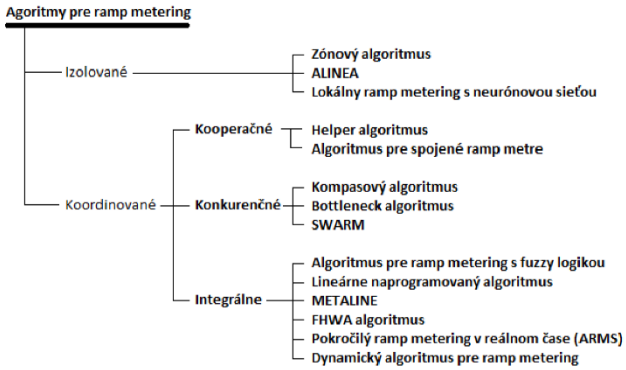
- vjazd na diaľnicu;
- prepojenie diaľnic;
- hlavný dopravný prúd na diaľnici.

Podľa spôsobu riadenia (t. j. podľa dopravných pravidiel):

- jedno vozidlo na zelenú;
- viac vozidiel na zelenú.

1.3 Riadiace algoritmy

Najdôležitejšou súčasťou systémov regulujúcich vstup vozidiel na diaľnicu (dávkovanie) je riadiaci algoritmus, ktorý určuje kedy a koľko vozidiel sa môže pripojiť z vedľajšej cestnej komunikácie k dopravnému prúdu na hlavnej komunikácii. Prehľad najčastejšie používaných alebo testovaných stratégií je na obr. 5. Skutočný počet je vyšší a neustále sa zvyšuje.
Vo všeobecnosti možno riadiace stratégie rozdeliť do 2 veľkých skupín – na lokálne (alebo izolované) a na koordinované (obr. 5). Lokálne berú do úvahy iba dopravné podmienky na príjazdovej komunikácii a v najbližšom segmente nadradenej komunikácie (diaľnice), kde je systém nasadený. Koordinované stratégie si všímajú dopravnú situáciu na dlhšom úseku nadradenej komunikácie (diaľnice) alebo diaľničnú sieť ako celok. Odborná literatúra ich ešte ďalej rozdeľuje na kooperačné, konkurenčné a integrálne (integrované). Obidve skupiny sú predmetom dlhodobého výskumu a možno sa tak stretnúť so snahami o využitie rôznych prístupov a o riešenie rôznych problémov (napr. multi-hierarchická stratégia koordinácie [6], využitie učenia posilňovaním (reinforcement learning) [7], kombinácia s navádzaním (route guidance) [8], nelineárny odhad pomocou revidovanej metódy číselnej diferenciácie [9], problém stability [10], pravdepodobnostný prístup [11], využitie Matlabu [12], a pod.).



Obr. 5 Klasifikácia riadiacich algoritmov

Fig. 5 Classification of control algorithms

2. Aplikácia algoritmov ZONE a ALINEA

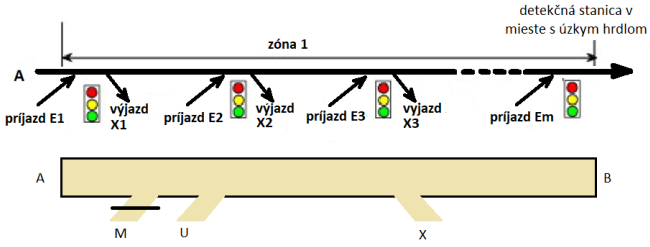
Väčšina aplikácií ramp meteringu predpokladá existenciu 3 a viacerých jazdných pruhov v jednom smere, kedy pre priestorové obmedzenia už nie je možné ďalšie rozširovanie počtu jazdných pruhov (celkovej šírky vozovky). Ide o situáciu vyskytujúcu sa v blízkosti rozľahlých veľkomiest, ktoré bojujú s nadmernou hustotou dopravných prúdov. Charakter diaľničnej siete na Slovensku zatiaľ nevykazuje podobné rysy, čo ale môže byť otázka blízkej budúcnosti v závislosti na rozširovaní diaľničnej siete a intenzifikácii dopravy. Z uvedeného dôvodu sa zatiaľ nejaví ako efektívne nasadzovanie zložitejších koordinovaných prístupov riadenia. V snahe priblížiť možnosti postupnej implementácie ramp meteringu sa v ďalšom texte zameriame na modelovanie, kalibráciu a simuláciu 2 lokálnych (izolovaných) typov algoritmu – ZONE a ALINEA, a na ich teoretickú implementáciu za zjednodušujúcich podmienok. Pre demonštračné účely sme zvolili konkrétnu časť slovenskej diaľnice D3. Cieľom bude ich vzájomné porovnanie ako aj porovnanie so situáciu, kedy zostane prístup vozidiel na diaľnicu neobmedzený. Vychádzame pritom z konfigurácie systému ako je uvedené na obr. 3b.

2.1 Algoritmus ZONE

2.1.1 Teoretické základy

Zónový algoritmus bol prvý raz nasadený v Minnesote (preto aj označenie „minnesotský algoritmus“) a snaží sa o vyváženie objemu vozidiel vstupujúcich a vystupujúcich do/z vymedzenej zóny s cieľom zachovať plynulý prejazd medzi zónami. Každá zóna môže pozostávať z viacerých čiastkových úsekov (obr. 6), z ktorých nie všetky musia mať vstupy s riadeným dávkovaním vozidiel, t. j. daný systém nemusí byť inštalovaný alebo nemusí byť práve v činnosti.
Výjazdy z diaľnice sú zväčša bez obmedzení a nie sú zdrojom častých problémov a incidentov. Prechod zo zóny do zóny môže obsahovať „úzke hrdlo“ (tzv. *bottleneck*) vzniknuté spájaním jazdných pruhov do menšieho počtu, čo môže prinášať komplikácie. Dĺžka zóny je približne od 4 do 8 km. Algoritmus tak spracúva viaceré vstupujúce a vystupujúce objemy vozidiel každej zóny a podľa nich nastavuje priepustnosť jednotlivých vstupných miest (dávkuje vozidlá).

Vyberá sa vždy jeden zo 6 možných časových intervalov zberu dát, v rozmedzí od žiadneho až po dĺžku cyklu 24 s (zodpovedá cca 150 voz/hod). Názov „stanica“ označuje skupinu detektorov na diaľnici.



Obr. 6 Návrh 1 zóny v ZONE algoritme
Fig. 6 Design of 1 zone in the ZONE algorithm

Algoritmus stráži, aby sa v prípojnom pruhu netvorili kvôli regulácii vstupujúcich vozidiel kolóny. Preto ich dĺžku monitoruje pomocou dopravných detektorov (na obr. 3b naznačené ako slučky). Minimálne dávkovanie vozidiel na príjazdovej komunikácii sa navrhuje tak, aby čakací čas pre jedno vozidlo nepresiahol stanovenú hodnotu (najčastejšie 4 min ≈ 240 s). K tomu je potrebné odhadnúť počet vozidiel (T), ktoré sa zmestia na príjazdovú komunikáciu, na základe priemerných dĺžok vozidiel, hustoty vozidiel (ako blízko seba môžu v kolóne stáť), rozmerov komunikácie a aktuálneho dávkovania. Pre max čas cyklu C_{max} potom platí [13]:

$$C_{max} = \frac{240\ (sekúnd\ v\ 4\ minútach)}{T}\ . \qquad (1)$$

Uvedenú hodnotu možno previesť na min. priepustnosť R_{min} podelením jednej hodiny hodnotou C_{max}:

$$R_{min} = \frac{3600\ (sekúnd\ v\ hodine)}{C_{max}}\ . \qquad (2)$$

Čas cyklu C_{max} nesmie byť väčší ako 15 s, preto absolútna minimálna priepustnosť na každom príjazde je 240 voz/hod. Je užitočné mať na pamäti, že časy cyklu a časy čakania sú v inverznom vzťahu – dlhší čas cyklu zodpovedá nižšej hodinovej priepustnosti.

Dáta z detektorov na hlavnom ťahu sa získavajú štandardne v 30 sekundových intervaloch, namerané objemy vozidiel a ich rýchlosť slúžia na nastavovanie prietoku v rámci stratifikačného (vrstvového) zónového algoritmu [3][13]. Algoritmus pracuje s nasledujúcim premennými (obr. 6), [3]:

- *A* – meraná hodnota: celkový počet vozidiel vstupujúcich do zóny zo susednej zóny (na obr. 6 zľava);
- *U* – meraná hodnota: koľko vozidiel vstúpilo do zóny z neregulovaných vjazdov;
- *X* – meraná hodnota: objem vozidiel, ktoré opustili zónu (všetky výjazdy súhrnne);
- *B* – meraná hodnota: celkový počet vozidiel vystupujúcich zo zóny v zúženej časti do susednej zóny;
- *M* – vypočítaná hodnota: maximálne množstvo vozidiel, ktoré smie vstúpiť do zóny cez všetky riadené príjazdové komunikácie (medzi bodmi A a B);
- *S* – vypočítaná hodnota: kapacita zóny; ak je doprava plynulá a jej intenzita (počet vozidiel za jednotku času) nízka, hlavná komunikácia (diaľnica) má nevyužitú kapacitu a systémy regulujúce vstup na ňu nemusia byť tak reštriktívne. Preto sa nevyužitá kapacita považuje za výstupnú hodnotu a počíta sa na základe priemernej hustoty pre plynulú dopravu na diaľnici porovnanej s aktuálnymi hustotami na diaľnici.

Všetky premenné sú z hľadiska fyzikálneho rozmeru vyjadrené v počte vozidiel za jednotku času (zväčša hodinu). Úlohou algoritmu je zabezpečiť, aby objem vozidiel vystupujúcich zo zóny (pravá strana rovnice (1)) bol vyšší alebo max. rovný ako objem vozidiel do nej vstupujúcich (ľavá strana rovnice(1)):

$$M + A + U \leq B + X + S, \qquad (1)$$

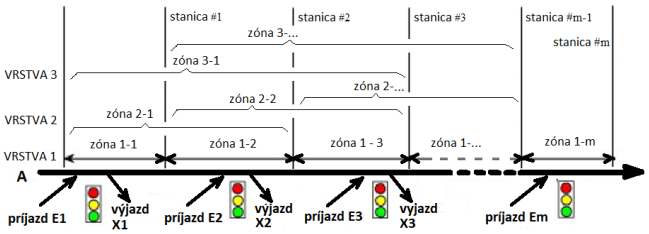
a odtiaľ

$$M \leq B + X + S - A - U. \qquad (2)$$

Ak R_n je navrhovaná dávka vozidiel na príjazdovej komunikácii so systémom riadeného dávkovania *n*; D_n je dopyt pre vjazd v mieste *n*; a D je požiadavka na vjazd na všetkých regulovaných vjazdoch danej zóny, potom platí:

$$R_n = \frac{M * D_n}{D}. \qquad (3)$$

Kľúčom k regulovaniu vjazdov je rozprestrieť objem M naprieč celou zónou vzhľadom na požiadavku D na regulovaných vjazdoch so závorami.

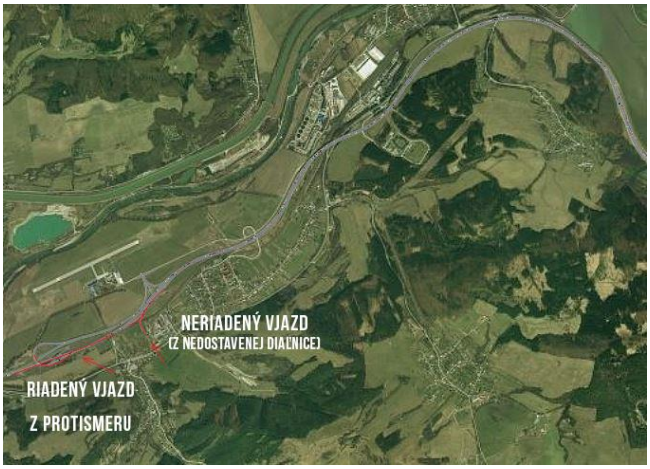


Obr. 7 Návrh vrstvenia zón pri SZM stratégii
Fig. 7 Design of zone layering in the SZM strategy

Vzhľadom na skutočnosť, že detekčné stanice sú rozmiestnené relatívne blízko seba a dĺžka jednotlivých zón môže byť aj väčšia, zóny sa v rámci diaľničného systému prekrývajú. Toto prekrývanie zón spôsobuje, že existuje tzv. „vrstvenie zón“ (obr. 7). Prvá vrstva obsahuje všetky zóny pozostávajúce presne z 2 staníc, druhá vrstva má všetky zóny s 3 stanicami, atď. – používa sa celkovo až 6 vrstiev. Vďaka tomuto konceptu možno aplikovať stratégiu dávkovania vozidiel v rámci vrstvených zón (*SZM – Stratified Zone Metering*). Pre príklady konkrétnych výpočtov vysvetľujúcich používanie jednotlivých vzťahov a vyčíslovanie jednotlivých premenných možno čitateľa odporučiť napr. na literatúru [13].

2.1.2 Aplikácia zónového algoritmu

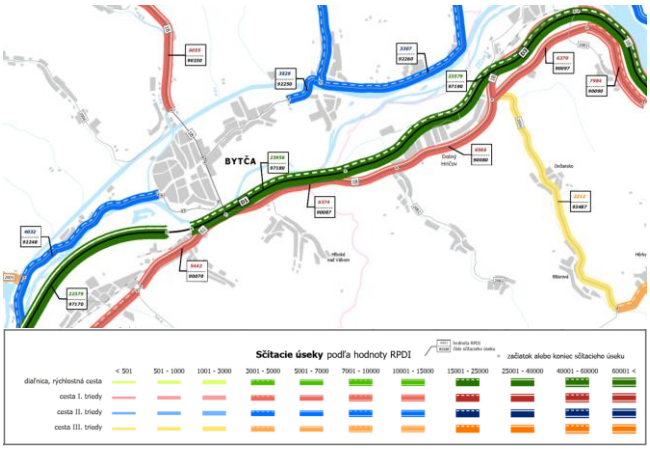
Zónový algoritmus bol zvolený z toho dôvodu, že nie je tak závislý na počte jazdných pruhov ako iné algoritmy a možno ho aplikovať aj na priestorovo menej rozvinutú infraštruktúru. Pre demonštračné účely bola zvolená časť diaľnice D3 v dĺžke cca 8000 m (obr. 8), ktorá v čase vypracúvania modelov bola v štádiu výstavby (časť diaľnice s tunelom Považský Chlmec bola medzičasom dokončená a sprevádzkovaná, úseky na Ovčiarsko a Višňové sú stále vo výstavbe – pozn.). Algoritmus bol zjednodušený pre účely práce s iba jedinou zónou. Z hľadiska modelovania a simulácie sa pri štúdiu a analýze ramp meteringu používajú viaceré nástroje, napr. Quadstone Paramics, S-Paramics, AIMSUN, PTV Vissim a mnohé ďalšie. V našej štúdii poslúžila plná verzia nástroja PTV VISSIM (vyžadujúca si aj Microsoft®.NET Framework) a v rámci neho najmä modul VisVAP uľahčujúci tvorbu, testovanie a kalibráciu algoritmov ramp meteringu. Vychádza sa z výsledkov a praktických skúseností získaných v práci [14].



Obr. 8 Voľba zóny (úsek D3 Žilina Strážov – Žilina Brodno)

Fig. 8 Selection of the zone (section of D3 Žilina Strážov – Žilina Brodno)

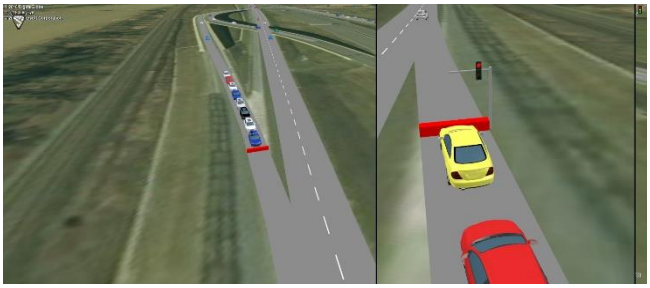
Pri určovaní objemov dopravy boli použité štatistické údaje zo Slovenskej správy ciest charakterizujúce stav v roku 2015 (Obr. 9.).



Obr. 9 Štatistické údaje – rok 2015 [15]

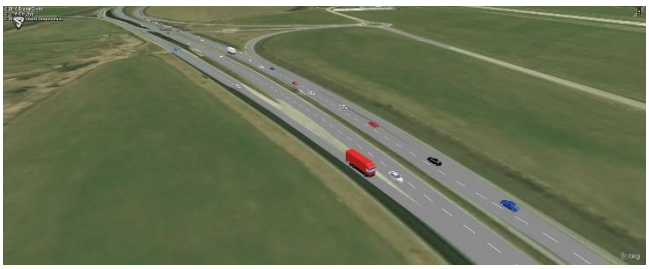
Fig. 9 Statistic data – 2015 year [15]

Priemerná hodnota objemu prevádzky na zvolenom úseku sa pohybovala v rozpätí od 15001 do 25000 vozidiel za deň. Ako horná hranica sa preto cvične zvolila hodnota 22379 voz/deň (Obr. 9), čomu zodpovedá po zaokrúhlení cca 1050 voz/hod pre smer Žilina Strážov – Žilina Brodno. Pre smer od Žiliny sa na modelovanie špičkových hodnôt nastavila dvojnásobná hodnota, pre vjazd v smere od tunela sa nastavilo 100 voz/hod.



Obr. 10 Riadený vjazd na D3 (s aktívnym návestidlo s červeným návestným znakom)

Fig. 10 The controlled entry to D3 (with an active signal having red signal aspect)



Obr. 11 Neriadený vjazd do zóny

Fig. 11 Uncontrolled entry to the zone

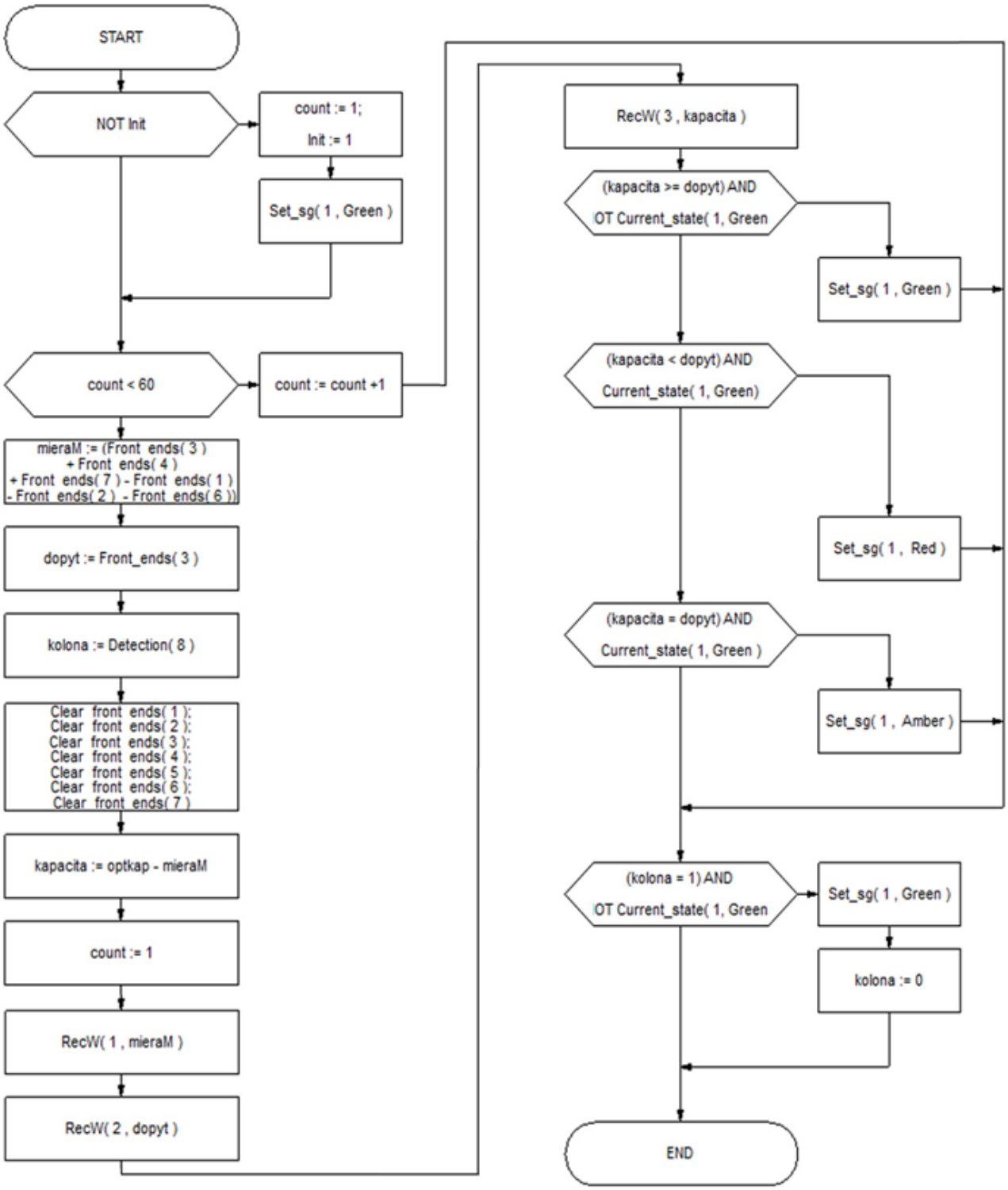
Príklady vzhľadu riadeného a neriadeného vjazdu v simulačnom prostredí sú na ilustráciu uvedené na obr. 10 a 11 (v danom poradí). V rámci nastavovania statického smerovania jazdy vozidiel boli definované odbočenia – ako žlté pruhy začínajúce červenou a končiace tyrkysovou farbou (obr. 12). Stanovenie odbočujúcich vozidiel bolo ponechané (vzhľadom na chýbajúce štatistiky) na náhodný výber.

Obr. 12 Ukážka rozloženia niektorých trás

Fig. 12 An example of the layout of some routes

Pre vytvorený model s nastaveným objemom vchádzajúcich vozidiel a odhadom ich správania bolo potrebné vytvoriť základný algoritmus na riadenie návestidla (dávkovanie vozidiel). Tomu predchádzalo vytvorenie signálovej skupiny a definícia základného správania sa návestidla. Potom bolo možné začať do modelu pridávať návestidlá, detektory a ďalšie potrebné prvky.

Originálny algoritmus bolo možné vzhľadom na jednoduchosť konfigurácie zjednodušiť (nepočíta sa s viacerými vjazdmi). Jeho grafické znázornenie je na obr. 13. Algoritmus začína iniciáciou, nastavením návestidla na zelenú a resetom počítadla (*count*) na hodnotu 1.Uvedený cyklus prebehne iba raz na samotnom začiatku. Ďalej cyklus počíta 60 cyklov, čo znamená, že odčítanie hodnôt z detektorov prebehne v našom prípade každú minútu rovnako ako aj nastavovanie nového stavu návestidla. Po dosiahnutí hodnoty *count* = 60 sa spustí cyklus (graficky vo vnútri počítadla). Hodnota premennej *mieraM* sa vypočíta ako rozdiel hodnôt načítaných z detektorov monitorujúcich počet vozidiel na výstupoch a vstupoch zóny. Následne sa načíta hodnota premennej *dopyt*. Túto hodnotu odovzdáva detektor na spočítavanie vozidiel umiestnený na začiatku riadeného vjazdu ústiaceho do zóny (obr. 14). Do premennej *kolona* sa priradí hodnota 0 alebo 1 podľa toho, či na detektore kolóny je prítomné vozidlo alebo nie. Po načítaní údajov z detektorov potrebných pre výpočty sa vnútorná pamäť detektorov vymaže a pripraví na načítanie nových hodnôt. Hodnota premennej *kapacita* sa určí ako rozdiel konštanty *optkap* a premennej *mieraM*. Nasleduje vynulovanie počítadla. Príkazy *RecW* (*record value*) slúžia na odoslanie momentálnych hodnôt premenných do programu Vissim.



Obr. 13 Blokový diagram zónového algoritmu

Fig. 13 The block diagram of the zone algorithm

Poslednými príkazmi v algoritme sú podmienky pre nastavovanie návestidla:

- ak je hodnota premennej *kapacita* vyššia alebo rovná ako hodnota premennej *dopyt* a farba návestidla nie je zelená, tak sa na zelenú zmení;
- ak je hodnota premennej *kapacita* nižšia ako hodnota premennej *dopyt* a aktuálny stav návestidla je zelená, tak sa jeho farba zmení na červenú;
- ak je hodnota premennej *kapacita* nižšia ako hodnota premennej *dopyt* a aktuálny stav návestidla je zelená, tak sa jeho farba zmení na žltú.

Poslednou súčasťou algoritmu je podmienka na vyhodnotenie prítomnosti kolóny, ktorá by dĺžkou už mohla ovplyvňovať plynulosť premávky na miestnej komunikácii pre vjazdom na diaľnicu. Ak je kolóna prítomná a dostatočne dlhá, zmení sa stav návestidla okamžite na zelenú farbu a dôjde k vyprázdneniu vjazdu.

Nevyhnutným krokom bola kalibrácia algoritmu – bolo potrebné sledovať správanie sa vozidiel na prvej odbočke za riadeným vjazdom (úzke hrdlo). Hodnota konštanty optimálnej obsadenosti bola zvolená 16 voz/min (pre rozpätie 15 až 20 voz/min sa ešte netvorili výrazné kongescie). Systém je však stochastický, takže vznik zápchy nie úplne eliminuje.

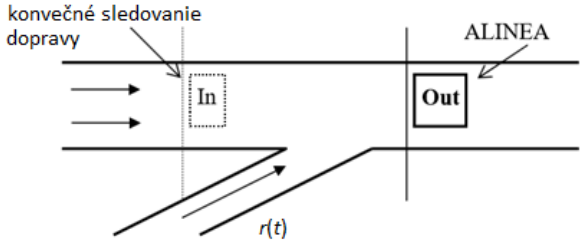
Obr. 14 Umiestnenie detektorov v zóne

Fig. 14 Detectors positioning in the zone

2.1 Algoritmus ALINEA

2.1.1 Teoretické základy

Ide opäť o jednoduchý, izolovaný a efektívny algoritmus, ktorý bol publikovaný už v roku 1991 [16]. Algoritmus využíva teóriu spätnej väzby a jeho cieľom je udržať maximálny prietok vozidiel na diaľnici za miestom vjazdu, vyjadrený prostredníctvom požadovanej hodnoty obsadenosti *o* [%] v mieste výstupných detektorov *out* (obr. 15).



Obr. 15 Umiestnenie vstupných a výstupných detektorov

Fig. 15 Positioning of input and output detectors

Dávkovanie vozidiel (intenzita dopravy) na vjazde $r(t)$ sa riadi vzťahom (4):

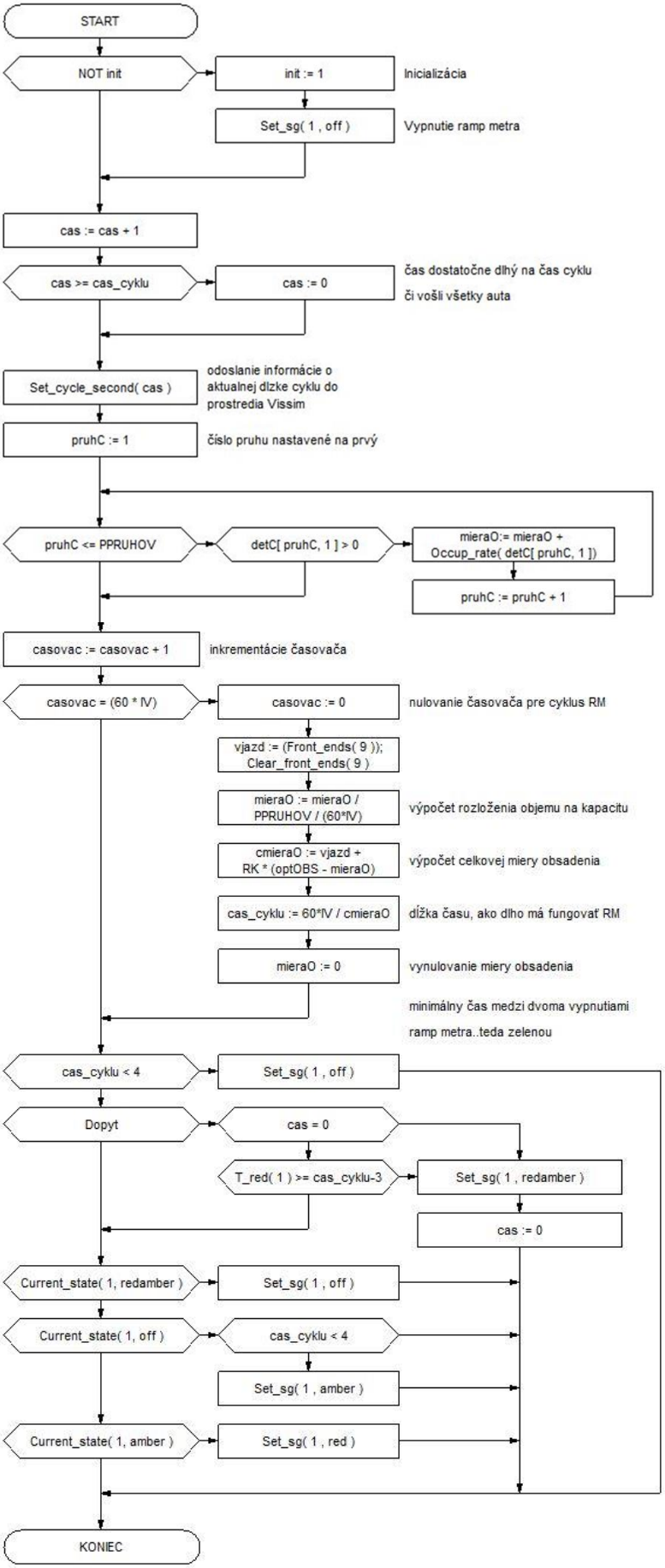
$$r(t) = r(t-1) + K_R \cdot (o - o_{out}\,(t)) \text{ [voz/hod], kde:} \qquad (4)$$

$r(t)$ – počet vozidiel vpustených na diaľnicu v časovom intervale t;
$r(t-1)$ [voz/hod] – nameraná intenzita dopravy v časovom intervale $(t-1)$;
K_R [voz/hod] - regulačný parameter ($K_R > 0$), typicky nastavovaný na 70 voz/hod;
o [%] – požadovaná hranica obsadenosti v mieste *out* detektorov – v krajinách EÚ nastavovaná typicky v rozpätí 18-30%;
$o_{out}(t-1)$ [%] – obsadenosť zmeraná v časovom intervale $(t-1)$ v mieste *out* detektorov.

Algoritmus je stále predmetom výskumu, čitateľa možno odkázať napr. na teoretické dokazovanie jeho efektívnosti [17] či prípadovú štúdiu o nedávnej implementácii v Istanbule [18]. Ideálne podmienky na aplikáciu lokálnej verzie algoritmu ALINEA sú na diaľnici s aspoň 3 pruhmi v jednom smere a s dostatočne dlhým pripájacím pruhom. Pretože sa však ľahko prispôsobuje na rôzne podmienky a rôzne tvary infraštruktúry, nebol problém použiť ho na situáciu modelovanú v našom prípade. Aby bolo možné následné porovnanie výsledkov, použili sme v modelu rovnaké nastavenia objemov dopravy ako v prípade zónového algoritmu. Vytvorenie modelu pre tento algoritmus je náročnejšie. Algoritmus bol vytvorený tak, aby v prípade plynulej dopravy zasahoval čo najmenej. Jeho aktivita začne narastať v momente, ak sa začne zvyšovať objem dopravy detegovaný detektormi na diaľnici. Pre nastavenie „agresivity“ alebo zvýšenie aktivity algoritmu sa používa konštanta pre optimálnu obsadenosť hlavného ťahu *optOBS*.

2.1.2 Aplikácia algoritmu ALINEA

Algoritmus začína iniciáciou (obr. 16) podobne ako zónový algoritmus s tým rozdielom, že vypne svetelnú signalizáciu (nenastaví zelenú). Algoritmus je postavený na podmienke dopytu vozidiel, ktoré chcú vstúpiť na hlavný ťah, preto sa spustí svetelná signalizácia až vzhľadom na dopyt a podmienky za vjazdom. Premenná *cas* slúži ako počítadlo celkového času simulácie. Postupne sa inkrementuje a odosiela do prostredia Vissim, kde môžeme sledovať hodnoty premenných v každej sekunde simulácie. Premenná *cas_cyklu* je v prvom behu nulová, potom sa jej hodnota navyšuje a záleží od nej výpočet min. trvania červeného signálu na návestidle. Ak premenná *cas* dosiahne rovnakú alebo vyššiu hodnotu ako je dĺžka cyklu, počítadlo sa vynuluje – riadiaca logika tak vie, že uplynul min. čas trvania svetelného signálu na návestidle. Premenná pre číslo jazdného pruhu *pruhC* sa definuje kvôli cyklickému napĺňaniu poľa pre detektory na hlavnom ťahu s premennou detektor číslo *detC*. Začína na jazdnom pruhu 1 a pokračuje, až kým nepresiahne celkový počet jazdných pruhov definovaný konštantou *PPRUHOV* (v našom prípade hodnota 2 – na hlavnom ťahu za vjazdom sú 2 jazdné pruhy, každý s 1 detektorom).
Súčasťou algoritmu je aj ďalší časovač pre výpočet miery obsadenia *mieraO* a celkovej miery obsadenia *cmieraO*. Cyklus (graficky vo vnútri) tohto časovača trvá minútu, čo zodpovedá 60 cyklom riadiacej logiky (t. j. realizácia algoritmu raz za sekundu). Konštanta *IV* je nastavená na 1 a predstavuje hodnotu časového intervalu medzi dvoma diskrétnymi časovými okamihmi (*t*-1). Keď časovač dosiahne hodnotu 60 sekúnd, vynuluje sa a prebehne načítanie hodnoty z detektora počítajúceho vozidlá, ktoré vstúpili z riadeného vjazdu, do premennej *vjazd*. Následne sa vymaže vnútorná pamäť tohto detektora. V cykle sa vypočíta hodnota miery obsadenia a celkovej miery obsadenia, ktorá je smerodajná pri nastavovaní minimálneho času červeného signálu návestidla na riadenom vjazde - prostredníctvom výpočtu vyššie spomenutého času cyklu. Ak je vypočítaný potrebný čas cyklu vyšší alebo rovný ako 4 s, návestidlo zostane vypnuté. Ak je nižší a na detektore umiestnenom na vjazde sa objaví vozidlo, splní sa podmienka *Dopyt* a cyklus ďalej vyhodnocuje, či je čas nulový alebo nie. Ak áno, návestidlo sa zmení na žltočervené a počítadlo času sa vynuluje. Ak nie, zisťuje sa, či bolo trvanie červeného signálu aspoň 3 s. Ak áno, bola splnená podmienka na min. dobu svietenia červenej a môže sa nastaviť žlto-červená a počítadlo času vynulovať. Podľa opísaného postupu sa nasledovne overuje aktuálny stav signálu na návestidle – ak bolo návestidlo z predchádzajúcich podmienok nastavené na červenožltý signál, tak sa návestidlo vypne (de facto zelený signál) a vjazd na diaľnicu je voľný.
Ak je návestidlo počas doby dopytu vypnuté a vypočítaný potrebný čas pre cyklus je nižší ako 4 s, signál na návestidle sa zmení na žltý. Potom sa cyklus zopakuje a zo žltej sa prejde na červenú. Pre správne fungovanie tohto algoritmu je kritická jeho kalibrácia, pri ktorej je najdôležitejšie zachovanie plynulosti premávky na hlavnom ťahu (regulácia konštanty optimálneho obsadenia *optOBS*). Vzhľadom na nižší počet pruhov v našej konfigurácii bolo potrebné hodnotu tejto konštanty znížiť až na hodnotu 0,18, kedy algoritmus pre nastavené objemy dopravy vykazoval najlepšie výsledky.



Obr. 16 Blokový diagram algoritmu ALINEA

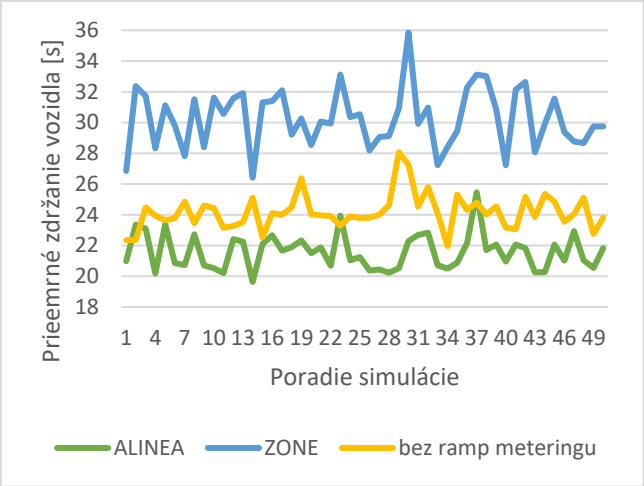
Fig. 16 The block diagram of the ALINEA algorithm

3. Vyhodnotenie výsledkov

Hodnotenie prínosu obidvoch riadiacich algoritmov bolo realizované na základe hodnôt priemerného zdržania, priemernej rýchlosti a celkového času, ktorý vozidlá strávili „cestovaním“ v modeli. Vyhodnocovaná bola aj situácia bez akéhokoľvek riadenia vjazdu vozidiel na diaľnicu. Simulácia bola nastavená rovnako pre obidva algoritmy na 50 opakovaní s náhodným priraďovaním hodnôt náhodným premenným s inkrementačným znakom 1.

3.1 Priemerné zdržanie

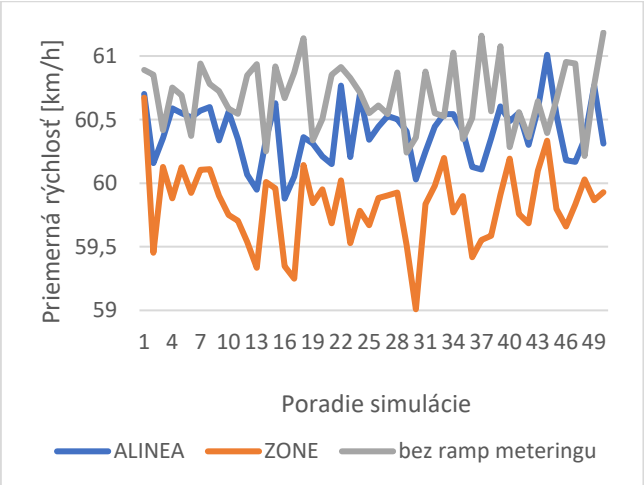
Ide o najkritickejší faktor vzhľadom na častú požiadavku, aby sa vozidlá pohybovali čo najrýchlejšie. Hodnota sa počítala ako podiel hodnôt celkového zdržania a počtu vozidiel, ktoré už modelom prešli plus vozidiel, ktoré sa ešte na konci simulácie v modeli nachádzali.



Obr. 17 Graf priemerného zdržania

Fig. 17 Graph of the average delay

Ako vidno na obr. 17, najmenšie zdržanie zabezpečil algoritmus ALINEA – priemerná hodnota 21,6 s (avšak len o 2,6 s lepšie ako v modelu bez ramp meteringu). Najvyššie zdržanie vykázal zónový algorimus (30 s), čo spôsobovalo čakanie na vjazde, kým sa uvoľní kapacita na hlavnom ťahu.



Obr. 18 Graf priemernej rýchlosti

Fig. 18 Graph of the average speed

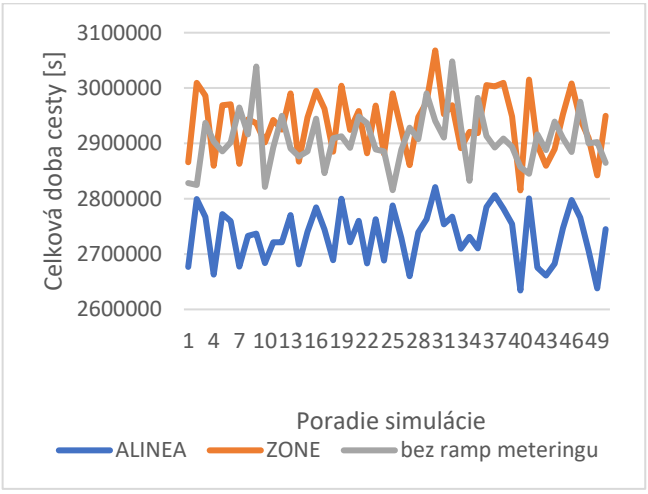
Poznámka: ak by viac ako o plynulosť išlo o využitie max. kapacity diaľnice a predchádzanie kongesciám, mohol by byť zónový algoritmus vhodnejší (nebolo simulované).

3.2 Priemerná rýchlosť

Hodnota priemernej rýchlosti vozidiel v modeli sa vypočítala ako podiel celkovej vzdialenosti, ktorú vozidlá prešli v modeli počas simulácie a celkového cestovného času všetkých vozidiel. Z grafu na obr. 18 vyplýva, že priemerná rýchlosť bez ramp meteringu bola 60,4 km/h. Zónový algoritmus v našich podmienkach spomaľoval premávku najviac (59,8 km/h). Algoritmus ALINEA spomalil premávku len o 2 desatiny.

3.3 Celková doba strávená na ceste

Pri simulácii sme sledovali aj dobu, ktorú v simulácii všetky vozidlá strávili na ceste. Z obr. 19 vyplýva, že v neriadenom modeli a v modeli so zónovým algoritmom išlo o priemerný čas 816 hodín oproti 759 hodinám pri riadení algoritmom ALINEA.



Obr. 19 Graf priemernej celkovej doby strávenej na ceste

Fig. 19 Graph of the total average time spent on the road

Záver

Inteligentné dopravné systémy zavedené do dopravnej infraštruktúry môžu pomôcť pri znižovaní alebo odstraňovaní rastúcich preťažení. Dávkovanie vozidiel pri vstupe na hlavné komunikácie (ramp metering) je jednou z aplikácií, ktoré majú potenciál odstrániť neefektívne využitie kapacity diaľnice a/alebo znížiť bezohľadné správanie vodičov. Tieto systémy však musia byť nastavené a kalibrované tak, aby spĺňali predpísané funkcie a očakávania prevádzkovateľov aj používateľov. Na nastavenie, kalibráciu a odskúšanie ešte pred vlastným nasadením dobre poslúžia modelovacie a simulačné nástroje.

Dva kontrolné algoritmy prezentované v príspevku ukázali, ako by mohlo vyzerať hodnotenie efektívnosti riadenia vstupu vozidiel na diaľnicu podľa troch rôznych parametrov. Algoritmus zóny najskôr vypočíta kapacitu a potom umožňuje vozidlám vstúpiť do zóny, čo má za následok oneskorenú odpoveď, čím sa zvyšuje oneskorenie pri vstupe a spomaľuje sa prevádzka. Alinetický algoritmus ALINEA umožňuje vozidlu vstúpiť do hlavnej trasy diaľnice

a potom vypočíta obmedzenie signalizované na vjazde nasledujúcim vozidlám. Preto je schopný zaistiť plynulejší a rýchlejší pohyb vozidiel.

Použitý aplikačný príklad (teoreticky aplikovaný na slovenské podmienky) predstavuje použitie uvedených riadiacich algoritmov za zjednodušených podmienok (menšie počty jazdných pruhov, objemy dopravy a pod.). Vo všeobecnosti môžu riadiace stratégie ramp meteringu priniesť uvedené benefity v situácii, kedy nie je možné zvyšovanie parametrov mobility extenzívnym spôsobom. Kriticky však treba poznamenať, že nemusia byť všeliekom a v niektorých špecifických situáciách môže byť ich použitie aj kontraproduktívne.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] Ramp Metering Status in North America. 1995 Update. Report DOT-T-95-117, US Department of Transportation, June 1995

[2] EURAMP (EUropean RAmp Metering Project). Európsky projekt 6. rámcového programu 507645 (03/2004 – 05/2007) https://trimis.ec.europa.eu/project/european-ramp-metering-project

[3] JACOBSON, L., STRIBIAK, J., NELSON, L., SALLMAN, D.: Ramp Management and Control Handbook. FHWA-HOP-06-001, 2006

[4] HOFFMANN, S.: Uplatnění zařízení pro regulaci vjezdu – tzv. dávkovací signalizace – na vysoce zatížených připojovacích pruzích. Dopravní inženýrství, 01/2009

[5] Traffic Management Services. RAMP METERING. Deployment Guideline. TMS-DG03, ver. 02-02-00, 59 s., December 2015,

[6] JIANG, R., LEE, J., CHUNG, E.: A Multi-Hierarchical Strategy for On-Ramp Coordination. International Journal of Intelligent Transportation Systems Research, vol. 15, issue 1, s. 50-62, 2017

[7] LU, C., HUANG, J., DENG, L. B., GONG, J.W.: Coordinated Ramp Metering with Equity Consideration Using Reinforcement Learning. Journal of Transportation Engineering Part A – Systems, vol. 143, issue 7, 2017

[8] PASQUALE, C., SACONE, S., SIRI, S., DE SCHUTTER, B.: A multi-class model-based control scheme for reducing congestion and emissionsin freeway networks by combining ramp metering and route guidance. Transportation Research Part C-Emerging Technologies, vol. 80, s. 384-408, 2017

[9] ABOUAISSA, H., MAJID, H., JOLLY, D.: Nonlinear state estimation and control for freeway on-ramp metering. Asian Journal of Control, vol. 19, issue 1, s. 233-244, 2017

[10] ALVAREZ-ICAZA, L., ROSAS-JAIMES, O., LARRAGA, M. E.: Stability of Local On-Ramp Metering Control Laws. Asisan Journal of Control, vol. 19, issue 2, s. 494-509, 2017

[11] MEHR, N., HOROWITZ, R. Probabilistic Freeway Ramp Metering. Proc. of the ASME 9th annual dynamic systems and control conference, Vol. 2, 2017

[12] LAGEREV, R., KAPSKI, D., BURINSKIENE, M., BARAUSKAS, A.: Reducing a possibility of transport congestion on freeways using ramp control management. Transport, vol. 32, issue 3, s. 314-320, 2017

[13] Stratified Zone Metering – The Minnesota Algorithm. https://www.dot.state.mn.us/trafficeng/modeling/dataextraction/Stratified%20Zone%20Metering.pdf

[14] SRNKA, T.: Riadenie toku vozidiel v aplikácii „Ramp Metering“. Diplomová práca, KRIS EF UNIZA, 2017

[15] Vysledky celoštátneho sčítania dopravy v SR v roku 2015: Žilinský kraj. Slovenská správa ciest. https://www.ssc.sk/files/documents/dopravne-inzinierstvo/csd_2015/za/scitanie_vuc_za_2015.pdf

[16] PAPAGEORGIOU, M., HADJ-SALEM, H., BLOSSEVILLE, J.-M.: ALINEA: A local Feedback Control Law for On-Ramp Metering. Transportation Research Board, issue 1320, s. 58-64, 1991

[17] ABOUAISSA, H., FLIESS, M., JOIN, C.: On ramp metering: towards a better understanding of ALINEA via model- free control. International Journal of Control, vol. 90, issue 5, s. 1018-1026, 2017

[18] ABUAMER, I. M., CELIKOGLU, H. B.: Local Ramp-Metering Strategy ALINEA: Microscopic Simulation Based Evaluation Study on Istanbul Freeways. 19th Euro Working Group on Transportation Meeting (EWGT2016), vol. 22, s. 598-606, 2017

Abstract

Increasing traffic volumes causes that freeway networks reach their limits when extending of transport infrastructure via adding new traffic lanes is not possible any more. Elimination or at least reduction of negative consequences resulting from congestions is then possible by deployment of advanced intelligent transport systems, particularly systems of ramp metering. The paper shows how to apply two control strategies (ZONE and ALINEA algorithms) under simplified conditions of the Slovak freeway network. Results of modelling and simulations obtained in the PTV Vissim environment demonstrate properties of both algorithms via their mutual comparison and comparison with situation when no regulation is applied as well.

prof. Ing. Aleš Janota, PhD, EurIng

Ing. Jozef Hrbček, PhD

Žilinská univerzita v Žiline
Fakulta a elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 8215/1, 010 26 Žilina
Tel.:+421 (0) 41 513 3300
E-mail: { ales.janota | jozef.hrbcek } @fel.uniza.sk

Obsah

EVOLUČNÁ OPTIMALIZÁCIA RIADENIA KRIŽOVATKY PEVNÝM SIGNÁLNYM PLÁNOM

Aleš Janota, Lukáš Slováček, Michal Gregor

Abstrakt
Článok ukazuje, ako možno pomocou použitia evolučných výpočtových techník a modelu optimalizovať riadenie svetelnej križovatky pevným signálnym plánom (s preddefinovaným trvaním fáz). Článok na príklade jednoduchej konfigurácie križovatky ilustruje kľúčové kroky v návrhu riadiaceho programu v jazyku Python. Kvôli stabilizácii hodnotení v rámci evolučnej optimalizácie sa pri optimalizácii parametrov riadiacej stratégie aplikuje determinizovaný model svetelnej križovatky. Riešenie sa však následne verifikuje pomocou plného modelu obsahujúceho aj stochastické prvky, čo umožňuje nielen verifikovať výsledné riešenie, ale ho i porovnať s pevným rozvrhom fáz vytvoreným štandardnými metódami.

Kľúčové slová: model, simulácia, križovatka, optimalizácia riadenia, evolučné metódy, evolučné stratégie, SUMO, Python

Úvod

Organizácia a riadenie dopravy v mestách predstavuje jeden z kľúčových problémov súčasnosti. Objem premávky neustále narastá a hoci sa mu postupne prispôsobuje aj cestná infraštruktúra, zásahy do nej sú veľmi nákladné a typicky sa nedajú realizovať dostatočne rýchlo na to, aby na meniacu sa situáciu pružne reagovali. Platí tiež, že v niektorých prípadoch nie je ďalšie rozširovanie cestnej infraštruktúry možné z hľadiska existujúcej zástavby a pod. Zo všetkých týchto dôvodov hrajú v súčasnosti mimoriadne dôležitú rolu metódy ponúkajúce alternatívne, efektívnejšie spôsoby, ako riadiť dopravu v rámci existujúcej infraštruktúry, ako zvýšiť jej plynulosť a bezpečnosť pre všetkých účastníkov (vodičov motorových vozidiel, cyklistov, chodcov a pod.) a tiež ako predchádzať zápcham a ďalším patologickým dopravným javom.

Aby sa zabezpečilo efektívne riadenie križovatiek, používa sa široké spektrum rôznych riadiacich systémov – od jednoduchých mechanických riešení až po sofistikované počítačové riadenie a koordinačné systémy so schopnosťou samonastavovania. Pri riadení svetelných križovatiek sa používa koncept fáz. Fáza predstavuje časť signálového cyklu alokovanú pre ľubovoľnú jednu kombináciu jedného či viacerých smerov, ktoré sú zároveň otvorené v rámci jedného či viacerých intervalov. Mnohé mestá stále najjednoduchšie riadenie svetelných križovatiek pomocou pevných signálnych plánov, ktoré sa aktivujú v rôznych časoch v priebehu dňa [1]. Výhodou tohto prístupu je, že sa riade-nie jednoducho navrhuje a sú naň nízke počiatočné aj prevádzkové náklady. Nevýhodou je, že systém nedokáže reagovať na zmeny v hustote dopravy a plytvá časom keď necháva otvorené smery, z ktorých do križovatky nevstupujú žiadne vozidlá. Vozidlá idúce z ostatných smerov sa kvôli tomu zbytočne zdržujú. Je veľmi ťažké vytvoriť optimálny signálny plán vychádzajúci z reálnych dopravných podmienok. Preto sa v minulosti venovalo veľa úsilia výskumu a štúdiám týkajúcim sa návrhu vhodných metód riadenia (napr. matematické analýzy, analýzy vychádzajúce z fuzzy logiky, umelé neurónové siete atď.) a optimalizácii získaných modelov.

Mnohé problémy riadenia v doprave sa dajú riešiť prístupmi založenými na simulácii a optimalizácii pomocou evolučných metód. Ako príklady dostupných optimalizovaných modelov môžeme spomenúť nasledovné - diskrétna udalostná simulácia a evolučné algoritmy boli použité na optimalizáciu komplexnej mestskej križovatky [2]; evolučné algoritmy boli tiež použité na optimalizáciu viac cieľového svetelného riadenia založeného na simulácii, ktoré vychádza z 3D mezoskopickej simulácie dopravy [3]; zdroj [4] zase navrhol inovatívny systém založený na agentnom riadení. Vo všeobecnosti patria evolučné algoritmy do triedy evolučných výpočtových techník [5] a používajú simulovanú evolúciu na hľadanie riešení pre komplexné praktické problémy. Stali sa populárnym nástrojom na optimalizáciu, ale aj na prehľadávanie a riešenie komplexných úloh [6].

Jedným z odvetví v oblasti evolučných metód sú tzv. evolučné stratégie (angl. *Evolution strategies*, ES). ES vychádzajú z princípu silnej kauzality: malé zmeny majú malé dôsledky. Dokáže nájsť takmer optimálne riešenie úlohy v rámci prehľadávaného priestoru a často sa používa v rámci empirických experimentov. Na rozdiel od inej triedy evolučných algoritmov – tzv. genetických algoritmov – sú väčšinou rýchlejšie, dokážu ľahko nájsť lokálne minimum a dávajú „dostatočne dobré“ riešenia, ktoré sú prijateľné, a operujú na priestoroch s reálnočíselnými parametrami (čo je vhodné pre mnoho bežných inžinierskych problémov). Naším cieľom je ukázať, ako sa dajú ES použiť na optimalizáciu fixných fáz pri riadení pomerne jednoduchej svetelnej križovatky.

Budeme využívať modely a softvérové nástroje na simuláciu dopravy, ktoré boli vyvinuté za účelom modelovania dopravy a tiež plánovania a analýzy rôznych dopravných stratégií riadenia dopravy v rámci simulácie [7]. V tejto štúdii využívame konkrétne simulačný nástroj SUMO – Simulation of Urban Mobility [8][9] – nástroj ktorý má otvorený zdrojový kód, je

prenositeľný medzi viacerými platformami, implementuje mikroskopickú a mezoskopiskú simuláciu dopravy a dokáže pracovať aj s veľkými cestnými sieťami. Používa sa v mnohých štúdiách, napr. aj v [10],[11],[12]. Základný prehľad možností a funkcionalít tohto nástroja, v porovnaní so 16 inými softvérovými nástrojmi na simuláciu dopravy, ktoré sú dostupné na trhu, sa dá nájsť v [7].

1. Evolučné algoritmy

Cieľom evolučných algoritmov je maximalizovať účelovú (hodnotiacu) funkciu vo vzťahu k určitej množine parametrov. Operujú na určitej populácii jedincov, ktoré predstavujú kandidátske riešenia danej úlohy. Jedince sa vo viacerých krokoch iteratívne modifikujú na základe nejakého metaheuristického pravidla. V prípade jedného z najznámejších typov evolučných algoritmov – tzv. genetických algoritmov – je modifikácia napr. založená na genetických operátoroch kríženia (typicky binárny operátor) a mutácie (unárny operátor).

My sme v tejto práci aplikovali evolučnú metódu známu pod názvom evolučné stratégie (ES). Špecifickou vlastnosťou ES je ich silný dôraz na samoadaptívnosť. Jedince obsahujú nielen parametre špecifikujúce vlastnosti daného kandidátskeho riešenia, ale aj tzv. endogénne parametre stratégie, ktoré určujú isté štatistické vlastnosti genetických operátorov a samy sú predmetom evolúcie. Samoadaptívnosť ES vyplýva práve z týchto endogénnych parametrov: evolúciou sa mení správanie samotných evolučných operátorov [13].

1.1 Model križovatky

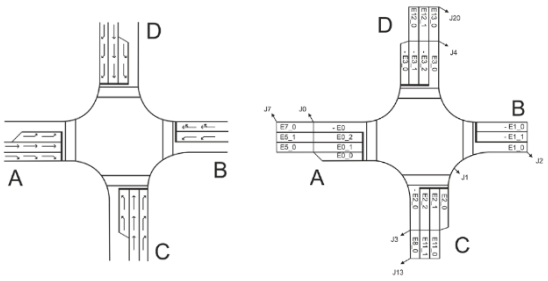
Cieľom tejto kapitoly je stručne predstaviť jednoduchý model cestnej svetelnej križovatky so známymi parametrami vytvorený v nástroji SUMO. Táto križovatka bude použitá ako referenčný model na porovnanie konvenčnej metódy riadenia s novo navrhnutým riešením. Schéma križovatky je na obr. 1 (vľavo), model v nástroji SUMO, obsahujúci identifikačné čísla jednotlivých jazdných pruhov ukazuje obr. 1 (vpravo). Maximálna rýchlosť v pruhoch je nastavená na 13.89 m/s, čo zodpovedá skutočnej predpísanej rýchlosti v rámci obce 50 km/h.

V ďalšom kroku sa nastavujú prepojenia medzi jednotlivými jazdnými pruhmi. Zhrnutie nastavení je v tab. 1.

Vstupná vetva	Vstupná hrana	Smer jazdy	Výstupná vetva	Výstupná hrana
A	E0_0	doprava	C	-E2_0
	E0_1	rovno	B	E1_0
	E0_2	doľava	D	E3_0
B	-E1_0	doprava	D	E3_0
	-E1_0	rovno	A	-E0
	-E1_1	doľava	C	-E2_0
C	E2_0	doprava	B	E1_0
	E2_1	rovno	D	E3_0
	E2_2	doľava	A	-E0
D	-E3_0	doprava	A	-E0
	-E3_1	rovno	C	-E2_0
	-E3_2	doľava	B	E1_0

Tab.1 Prepojenia jazdných pruhov

Čo sa týka riadenia križovatky, predpokladáme 3-fázový model riadenia, ktorý ukazujú obr. 2 a obr. 3.

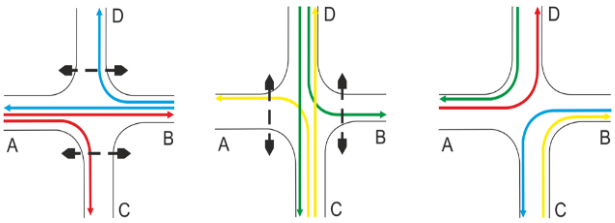


Obr. 1 Schéma križovatky (vľavo) a model križovatky v nástroji SUMO (vpravo)

Fig. 1 Intersection layout (left) and SUMO-based intersection scheme (right)

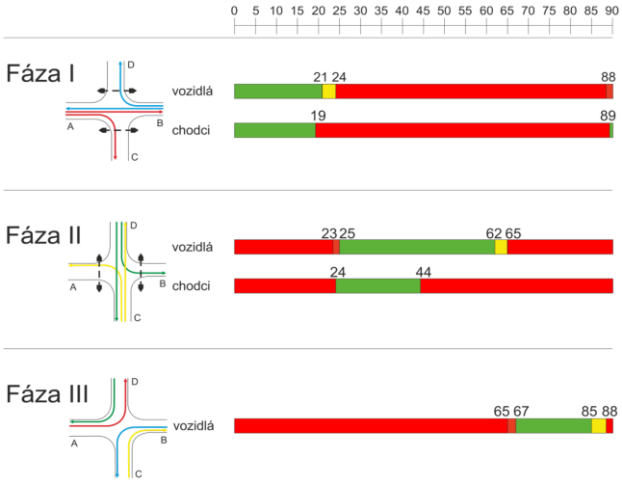
1.2 Riadenie križovatiek v nástroji SUMO

Jednou z výhod modelovacieho prostredia SUMO je možnosť využiť vstavané rozhranie na riadenie dopravy: tzv. TraCI (z angl. *Traffic Control Interface*). TraCI poskytuje prístup ku stavu bežiacej dopravnej simulácie, umožňuje získavať údaje o simulovaných objektoch a manipulovať so správaním sa objektov „online“ [7]. Pred spustením skriptu by sa mal užívateľ presvedčiť, že je správne nastavená premenná prostredia SUMO_HOME (podrobnejšie informácie je možné nájsť v návode).



Obr. 2 Definícia signálových fáz – fáza I (vľavo), II (uprostred) a III (vpravo)

Fig. 2 Definition of signal phases – Phase I (left), II (middle), and III (right)



Obr. 3 Definícia signálových fáz – fáza I (hore), II (uprostred) a III (dole)

Fig. 3 Definition of signal phases – Phase I (at the top), II (middle), and III (at the bottom)

1.3 Import Python modulov

Na riadenie križovatky s podporou nástrojov umelej inteligencie použijeme rozhranie TraCI z Python skriptu – jazyk Python je v súčasnosti v oblasti strojového učenia a umelej

inteligencie bezkonkurenčne najobľúbenejším jazykom. Skript musí začať deklaráciou ciest ku modulom súvisiacim s nástrojom SUMO a rozhraním TraCI. Príklad takej deklarácie (pre operačný systém Windows) nasleduje nižšie:

Zdrojový kód 1: Importovanie Python modulov pre operačný systém Windows

```
sys.path.append(os.path.join(os.path.dirname(__file__),'C:\Program-Files(x86)\DLR\Sumo', "tools"))

sys.path.append(os.path.join(os.environ.get("SUMO_HOME", os.path.join(

os.path.dirname(__file__), "C:\Program Files (x86)\DLR\Sumo")), "tools"))
```

1.4 Generovanie dopravy

Náš model križovatky zahŕňa dáta o jazdných pruhoch a riadení signálov svetelných križovatiek. Nehovorí však nič o doprave, ktorá by mala križovatkou prúdiť. Dopravné dáta sa budú generovať automaticky pomocou skriptu a budú zapísané do osobitného dočasného súboru, z ktorého sa v nástroji načítajú. Tento súbor obsahuje cesty definujúce všetky páry jazdných pruhov, medzi ktorými sa môžu vozidlá pohybovať a definície parametrov vozidiel prichádzajúcich z každého smeru, atď.

V našom prípade sú cesty určené geometriou križovatky a zostávajú v priebehu experimentov nezmenené. Definície týkajúce sa početnosti vozidiel a pod. sa však menia. My sme zvolili pre experimenty nasledujúci prístup: počas optimalizácie simuláciu determinizujeme, aby sa v hodnotiacej funkcii ES potlačili stochastické prvky. Keď však výsledné riešenie testujeme, používame prirodzene už úplnú, stochastickú simuláciu s randomizovanými konfiguráciami príjazdov vozidiel do križovatky, a pod. To nám pomáha určiť, či je nájdené riešenie dostatočne všeobecné.

Zdrojový kód 2: Vytvorenie dočasného súboru

```
with tempfile.NamedTemporaryFile('w', delete=False) as routes:

        print("""<routes>

<vType id="Car" sigma="0.5" length="5" minGap="2.5" guiShape="passenger"/>

…
```

Ako vidno, dočasný súbor sa vytvára pomocou balíčka „*tempfile*“. Koreňový tag obsahujúci cesty vozidiel sa vkladá do vnútra tagu <*routes*>. Tag <*vType*> nachádzajúci sa v jeho tele špecifikuje parametre vozidiel. Následne je možné vytvoriť samotné cesty:

Zdrojový kód 3: Vytvorenie ciest

```
<route id="CB" edges="E11 E2 E1" />""", file=routes)
```

V danom príklade vytvárame cestu pre vozidlá idúce z vetvy C do vetvy B. Obdobným spôsobom sa definujú cesty aj pre všetky ostatné smery.

Keď sú zadefinované parametre vozidiel a cesty, môžeme začať samotné vozidlá generovať:

Zdrojový kód 4: Generovanie vozidiel

```
N = 3600

for i in range(N):

    if random.uniform(0, 1) < pCB:

    print('<vehicle id="CB_%i" type="Car" route="CB" depart="%i" />' % (vehNr, i), file=routes)

    vehNr += 1

    lastVeh = i

print("</routes>", file=routes)
```

Simulácia je definovaná na trvanie jednej hodiny, t. j. 3600 sekúnd s 1-sekundovým simulačným krokom. Vozidlá je možné generovať v každom simulačnom kroku, t. j. raz za sekundu. Či v danom kroku generovať nové vozidlo sa rozhoduje na základe vzorky z Bernoulliho rozdelenia parametrizovaného na základe pomocnej premennej vyjadrujúcej koľko vozidiel za sekundu sa priemerne požaduje. Ak sa algoritmus rozhodne vozidlo generovať, zaregistruje sa v súbore spoločne so svojím identifikátorom, typom, cestou a časom, kedy opustí križovatku. Okrem toho sa inkrementujú určité pomocné premenné sledujúce koľko vozidiel sa vygenerovalo a kedy bolo generované posledné vozidlo.

Keď sa registrácia vozidiel dokončí, je potrebné uzatvoriť koreňový tag <*routes*>.

1.5 Nadviazanie spojenia s TraCI

Posledným krokom, ktorý je potrebné vykonať, je definícia atribútov simulácie. Keďže sa simulácie spúšťa zo skriptu, je potrebné spustiť SUMO server s TraCI knižnicou. Zdrojový kód potrebný na nadviazanie TCP spojenia so serverom je uvedený nižšie.

Zdrojový kód 5: Nadviazanie spojenia s TraCI

```
def start(cmd, port=None, numRetries=10, label="default"):

    if port is None:

        port = traci.sumolib.miscutils.getFreeSocketPort()

    sumoProcess = subprocess.Popen(cmd + ["--remote-port", str(port)],

            stdout=DEVNULL, stderr=DEVNULL)

    traci._connections[label] = traci.connect(port, numRetries, "localhost", sumoProcess)

    traci.switch(label)

    return traci.getVersion()

traci.start = start
```

V prípade, že má operačný systém aktívny firewall, môže tiež byť prirodzene potrebné pridať do jeho konfigurácie príslušné výnimky, aby komunikáciu neblokoval.

2. Simulácia

Keď sa vykonajú všetky potrebné nastavenia, môžeme vytvoriť funkciu na generovanie dopravy a zabezpečiť, že sa simulácia spustí so správnymi atribútmi.

Zdrojový kód 6: Spustenie simulácie

```
def run_simulation(params):

    routefile_name = generate_routefile()

    with contextlib.redirect_stdout(None):

        traci.start(["sumo","-c","map.sumocfg","-r",routefile_name,"--tripinfo- output","tripinfo.xml"])
```

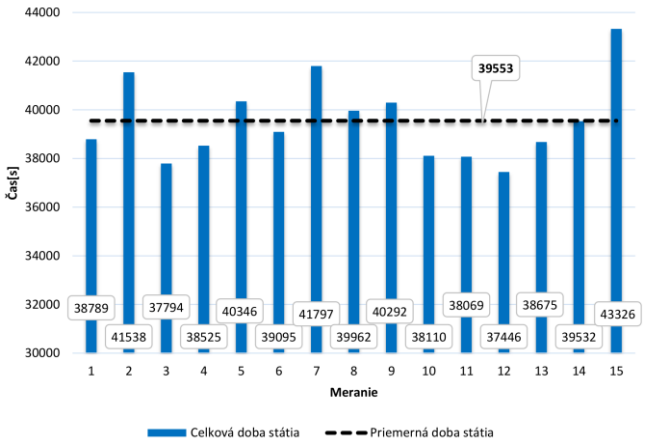
Naším cieľom je odhadnúť čakaciu dobu vozidiel prichádzajúcich z jednotlivých smerov. Keď sa úspešne nadviaže spojenie so SUMO serverom, môžeme ku parametrom simulácie pristupovať online.

Zdrojový kód 7: Čakacie časy vozidiel v jazdnom pruhu

```
while step < 3600:
traci.simulationStep()
    waitingTimeAC = traci.lane.getWaitingTime("E0_0")
    if (waitingTimeAC < pomE0_0 and pomE0_0 > pom0):totalTimeAC += pomE0_0
    pom0 = pomE0_0
    pomE0_0 = waitingTimeAC
    return (totalTimeAC + totalTimeAB + totalTimeAD + totalTimeBDA + totalTimeBC + totalTimeCB + totalTimeCD + totalTimeCA + totalTimeDA + totalTimeDC + totalTimeDB)
```

Skutočné hodnoty doby čakania sa zaznamenávajú v pomocných premenných (tu označených ako „*pom*“). Ten istý prístup sa používa pri všetkých jazdných pruhoch.

Simuláciou riadenia križovatky s pevnými časmi fáz rozumieme simuláciu takého riadenia, ktoré vychádza zo statických signálnych plánov. Keďže náš model je stochastický, musíme vykonať viacero simulačných behov a určiť priemerné čakacie doby (naprieč simulačnými behmi). Tieto následne použijeme ako základ pre porovnanie s evolučným prístupom. Priemerné čakacie doby ukazujú obr. 4 a obr. 5.



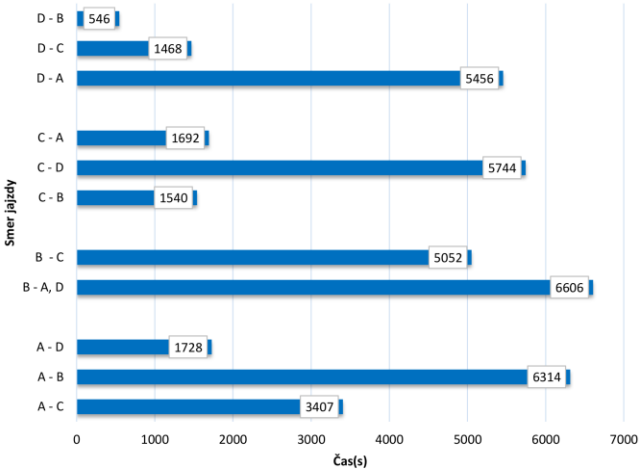
Obr. 4 Simulácia riadenia križovatky s pevnými časmi fáz

Fig. 4 Simulation of intersection control with fixed times

2.1 Evolučné ladenie časov fáz

Vyššie sme prezentovali výsledky simulácie pri riadení križovatky s pevnými časmi fáz určenými konvenčnou metodikou. Ďalším krokom je pokúsiť sa časy fáz vyladiť pomocou simulácie a evolučných stratégií. V našom prípade nebude vplyv evolučnej stratégie na riadenie križovatky priamy – evolúcia len optimalizuje nastavenia už existujúcich fáz.

Vytvoríme teda ďalší skript, v rámci ktorého bude evolučná stratégia navrhovať nastavenia preddefinovaných fáz a, sledujúc výstup hodnotiacej funkcie, bude aplikovať genetické operátory a vyberať najlepšie kandidátske riešenia s cieľom identifikovať čo možno najlepšie nastavenie fáz.



Obr. 5 Hodnoty priemerných čakacích dôb pre jednotlivé smery na križovatke s pevnými časmi fáz

Fig. 6 Values of the average waiting times for individual directions at the intersection with fixed times

Počiatočná populácia kandidátskych riešení sa generuje náhodným spôsobom podľa nižšie uvedeného kódu:

Zdrojový kód 8: Generovanie trvania fáz

```
def generate(random, args):
    phase_duration = []
    for i in range (args['num_phases']):
        if i == 1 or i == 2 or i == 8:
            phase_duration.extend([2])
        if i == 3 or i == 4 or i == 11 or i == 12:
            phase_duration.extend([1])
        if i == 7 or i == 10:
            phase_duration.extend([3])
        if i == 0 or i == 5 or i == 6 or i == 9:
            phase_duration.extend([random.randint(args['min_phase_duration'], args['max_phase_duration'])])
    return (phase_duration)
```

Musíme si uvedomiť, že opis fáz svetelnej signalizácie v rámci nástroja SUMO nezodpovedá tomu v reálnom svete. V SUMO sa novou fázou rozumie akákoľvek zmena v stave svetelnej križovatky. Preto je potrebné uvažovať prechody medzi fázami a nastaviť ich dĺžky štatisticky. Dĺžky fáz sa na začiatku zapíšu do súborov s príponou „.net.xml“. Preto sa musia zmeniť zakaždým, keď sa spustí simulácia. Dĺžky trvania fáz preto zapíšeme do dočasného súboru vždy, keď spúšťame simuláciu: podobne, ako to robíme pri generovaní premávky.

Zdrojový kód 9: Zápis fáz do dočasného súboru

```
def generate_tlLogic():
    with tempfile.NamedTemporaryFile('w', delete=False) as tlLogic:
        print("""<add>
            …
```

```
        """, file=tlLogic)
        for i in range (13):
            if i == 0:
                print('<phase duration="%d" state="rrrGGrrrrGGrGrGr"/>'% (params[0]), file=tlLogic)
            …
        print("""    </tlLogic></add>""", file=tlLogic)
    return(tlLogic.name)
```

Výstupom uvedenej funkcie je názov súboru s novo vygenerovanými dĺžkami fáz. Aby sa tento súbor úspešne prepojil so samotným simulačným procesom, je stále potrebné ho asociovať s parametrami, ktoré sa zadávajú pri spustení simulácie (konkrétne ide o parameter „a“).

Zdrojový kód 10: Parametre simulačného behu

```
traci.start(["sumo", "-c", "map.sumocfg", "-r", routefile_name, "-a", tlLogic_name])
```

Keď sme takto postupne prešli celý proces konfigurácie, zostáva nám už odprezentovať iba to, ako sme na predmetný problém aplikovali samotné evolučné stratégie. Evolučné stratégie – podobne ako väčšina ostatných evolučných prístupov – majú viacero menlivých častí, ktoré treba pri ich aplikácii na konkrétnu úlohu špecifikovať. Niektoré z nich sú problémovo agnostické: v tom prípade stačí typicky vybrať jednu z predprogramovaných alternatív. Iné komponenty evolučnej stratégie sú problémovo špecifické – môžu súvisieť napríklad s tým, akým spôsobom sa kódujú kandidátske riešenia a samozrejme s tým, ako sa ohodnocujú.

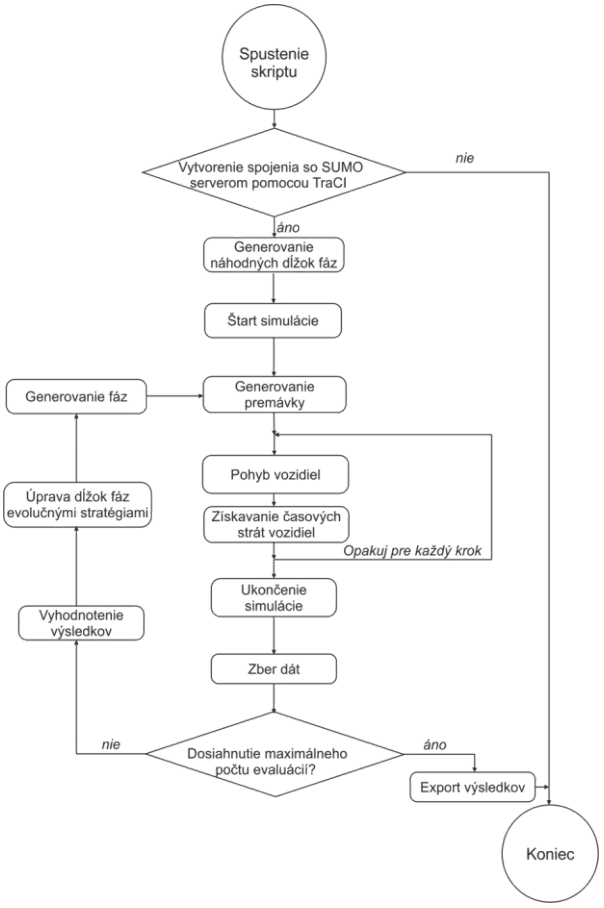
Ako uvidíme v nižšie priloženom listingu, definícia hodnotiacej funkcie je v tomto prípade jednoduchá – hodnotenie priamo závisí od výstupu simulačného programu, t. j. od celkovej doby čakania vozidiel. Ďalej je potrebné nastaviť ukončovacie kritérium, t. j. pri splnení akej podmienky evolúcia skončí. V našom prípade bude ukončovacie kritérium reprezentované jednoduchou funkciou s booleovskou návratovou hodnotou, pričom výstup „*True*“ znamená, že sa má vývoj ukončiť [14].

Ďalším komponentom, ktorý špecifikujeme, je pozorovacia funkcia (angl. *Observer*), ktorá nám umožňuje sledovať priebeh evolučného procesu, zaznamenávať a vizualizovať užitočné štatistické údaje o ňom, a pod. Pozorovacia funkcia dostáva na vstupe nasledujúce argumenty: aktuálnu populáciu jedincov, doterajší počet generácií a počet vyhodnotení hodnotiacej funkcie a prípadne ďalšie argumenty. Špecifikujú sa samozrejme aj ďalšie hyperparametre ako sú napr. veľkosť populácie, maximálny počet generácií a vyhodnotení hodnotiacej funkcie, či sa má pri výpočte využiť paralelizmus, a pod. Význam všetkých hyperparametrov možno nájsť v [14]. Evolučná stratégia na výstupe vracia zoznam, ktorého prvky predstavujú jedincov z finálnej populácie.

Zdrojový kód 11: Evolučná stratégia

```
def fitness(params, args):
    return run_simulation(params)
rand = random.Random()
rand.seed(int(time.time()))
es = inspyred.ec.ES(rand)
es.terminator = inspyred.ec.terminators.evaluation_termination
es.observer = inspyred.ec.observers.stats_observer
final_pop = es.evolve(generator=generate,
    evaluator=inspyred.ec.evaluators.parallel_evaluation_mp,
    mp_evaluator=fitness,
    mp_num_cpus=100,
    pop_size=32,
    maximize=False,
    max_evaluations=30000,
    num_phases=13,
    min_phase_duration=5,
    max_phase_duration=45)
```

Na obr. 6 prikladáme aj vizuálnu reprezentáciu celého riadiaceho skriptu. Pri výpočtoch využívame paralelizmus, ktorý nám umožňuje spustiť viacero nezávislých simulácií súčasne a vďaka tomu rýchlejšie určiť hodnotenia jedincov z populácie.



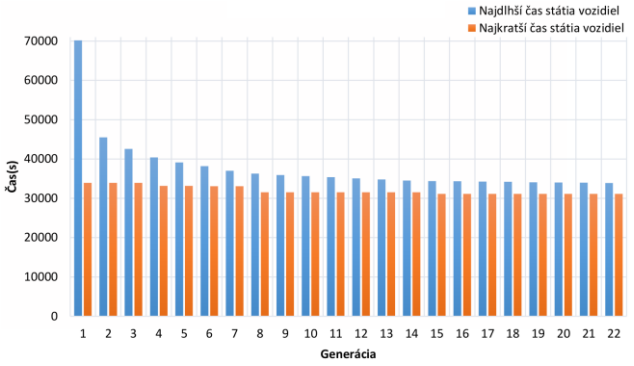
Obr. 6 Logika riadiaceho skriptu

Fig. 6 Logics of the control script

3. Výsledky evolučnej optimalizácie

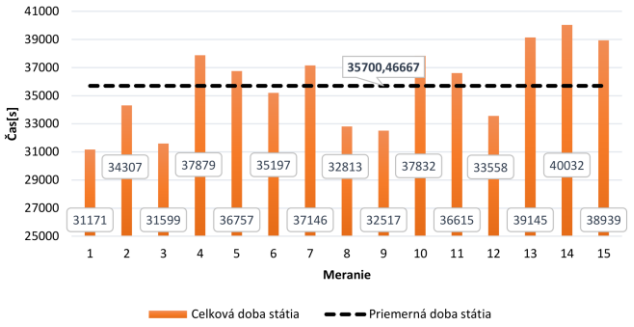
Výsledky optimalizácie pomocou evolučných stratégií prezentujú v grafickej podobe obr. 7 a obr. 8. Na obr. 7 vidno, že sa doba čakania s počtom generácií postupne znižuje. obr. 8 ukazuje celkové doby čakania z 15 rozličných simulačných behov pre dĺžky fáz nájdené evolučnou stratégiou.

Na obr. 9 sú vizualizované aj priemerné doby čakania pre jednotlivé jazdné pruhy. Najdlhšie čakacie doby sa vyskytli v úseku medzi uzlom C a D, t. j. na mieste, kadiaľ prechádza najintenzívnejšia doprava.



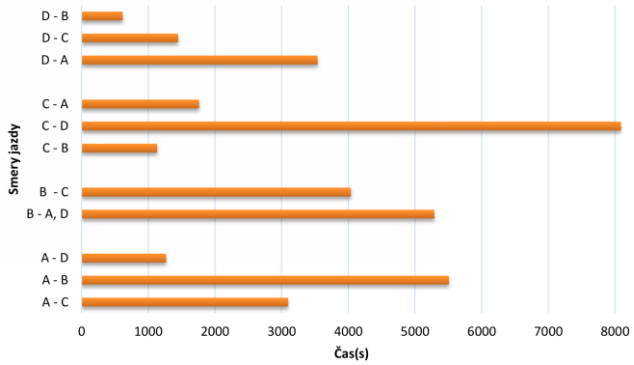
Obr. 7 Simulácie evolučných stratégií

Fig. 7 Simulation of evolution strategies



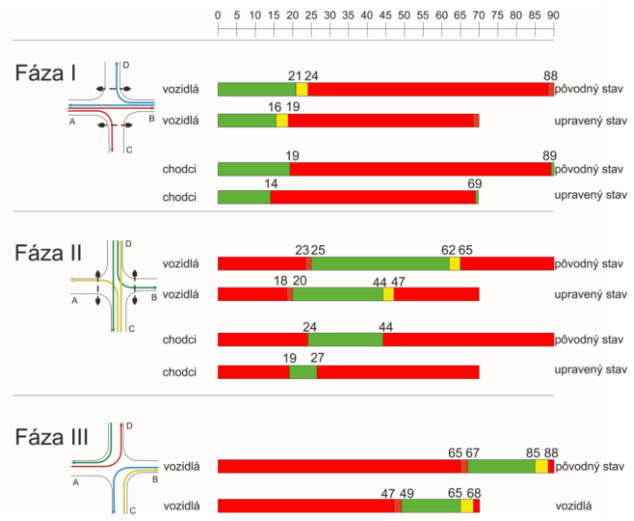
Obr. 8 Výsledky z 15 rôznych behov simulácie pri použití dĺžok fáz nájdených evolučnou stratégiou

Fig. 8 Results from 15 different simulation runs using the phase lengths found by the ES



Obr. 9 Priemerné časové straty pre jednotlivé smery

Fig. 9 Average waiting times for the individual directions



Obr. 10 Dĺžky fáz pred a po aplikácii evoluč. stratégií

Fig. 10 Phase lengths before and after usage of ES

Celkovo možno konštatovať, že načasovanie fáz navrhnuté evolučnou stratégiou je efektívnejšie než to navrhnuté štandardným výpočtom, čoho výsledkom sú nižšie časové straty pre jednotlivé vozidlá. obr. 10 ukazuje rozdiely medzi dĺžkami fáz pred a po optimalizácii pomocou evolučných stratégií.

4. Záver

Model prezentovaný v článku môže poslúžiť ako dobrý základ pre návrh nových algoritmov riadenia križovatky, či už pôjde o algoritmy z oblasti umelej inteligencie a strojového učenia alebo o iné, štandardnejšie prístupy. Samotný model by bolo možné ešte viacerými spôsobmi vylepšiť – napríklad zavedením ďalších zdrojov stochasticity, ktoré sú prítomné v reálnych dopravných systémoch a model ich zatiaľ neuvažuje. Tiež by bolo vhodné venovať osobitnú pozornosť modelovaniu správania chodcov, aby sa pri optimalizácii fáz minimalizovalo aj ich celkové zdržanie.

Vytvorené riešenie na báze evolučných stratégií je schopné optimalizovať dĺžky fáz v systéme so známymi parametrami, predovšetkým so známou intenzitou dopravy, ktorá je kľúčovým parametrom pri vytváraní spoľahlivého modelu.

Naše ďalšie výsledky (ktoré nie sú zahrnuté v tomto článku) indikujú, že ešte efektívnejšie riadenie križovatky by bolo možné dosiahnuť s použitím dynamického riadenia, t. j. takého, ktoré berie do úvahy aj súčasný stav križovatky. Dá sa predpokladať, že rozdiel bude ešte markantnejší v prípadoch, kedy sa aktuálna dopravná situácia bude výrazne vymykať z rozdelení, s ktorými sa počítalo pri pôvodnom návrhu riadenia.

Ďalšie práca by sa teda mohla sústrediť na dynamické metódy riadenia svetelnej križovatky s podporou umelej inteligencie. V poslednom období sa v tejto oblasti experimentuje napríklad aj s aplikáciou takých pokročilých metódy strojového učenia, ako je hlboké učenie s odmenou. Dá sa očakávať, že takéto prístupy budú schopné riadiť križovatku omnoho efektívnejšie, ale tiež bude pravdepodobne omnoho náročnejšie verifikovať ich správanie z pohľadu bezpečnosti.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] NIGARNJANAGOOL, S., DIA, H.: Evaluation of a dynamic signal optimisation control model using traffic simulation. IATSS Research, Vol. 29, No. 1, 2005.

[2] MIHĂIŢĂ, A. S., CAMARGO, M., LHOSTE, P.: Optimization of a complex urban intersection using discrete event simulation and evolutionary algorithms. IFAC Proceedings Volumes, Volume 47, Issue 3, 2014, 8768-8774.

[3] MIHĂIŢĂ, A. S., DUPONT, L., CAMARGO, M.: Simulation Modelling Practice and Theory 86, 2018, 120-138.

[4] ZHU, Y., DUAN, J., YIN, H.: A novel agent-based intersection control method for urban traffic. 2016 World Automation Congress (WAC), IEEE, Rio Grande, Puerto Rico: 31 July-4 Aug 2016, pp. 1-5.

[5] VIKHAR, P. A.: Evolutionary algorithms: A critical review and its future prospects. 2016 Int. Conference on Global Trends in Signal Processing, Information Computing and Communications (ICGTSPICC).

[6] YAO, X.: Global optimisation by evolutionary algorithms. IEEE Transactions on evolutionary computation, vol. 3, no. 2, 1999, pp. 82-102.

[7] PELL, A., MEINGAST, A., SCHAUER, O.: Trends in Real-time Traffic Simulation. Transportation Research Procedia 25, 2017, p. 1477-1484.

[8] SUMO – Simulation of Urban MObility. http://www.sumo.dlr.de/userdoc/Sumo_at_a_Glance.html

[9] GUDWIN, R. R.: Urban Traffic Simulation with SUMO. A Roadmap for the Beginners. DCA-FEEC-UNICAMP, 2016, 44 p.

[10] METEV, S. M., VEIKO, V. P., OSGOOD, R. M.: Urban Traffic Simulation with SUMO. Springer-Verlag Berlin, Germany, 1998.

[11] DIAS, J. C., ABREU, P. H., SILVA, D. C., FERNANDES, G., MACHADO, P., LEITAO, A.: Preparing Data for Urban Traffic Simulation using SUMO. The SUMO User Conference (SUMO2013), May, 2013.

[12] BEHRISCH, M., BIEKER, L., ERDMANN, J., KRAJZEWICZ, D.: Sumo-Simulation of Urban Mobility – An Overview. SIMUL 2011, the Third International Conference on Advances in System Simulation, 55-60.

[13] BEYER, Hans-Georg, SCHWEFEL, Hans-Paul: Evolution strategies. A comprehensive introduction. Natural Computing 1: 3-52, 2002.

[14] inspyred 1.0 documentation. Library Reference. https://pythonhosted.org/inspyred/reference.html

Abstract

The paper shows, how evolutionary methods and a model can be used to optimize static light intersection control (i.e. control using fixed duration of phases). It uses the case of a simple intersection to illustrate the key steps in designing a control programme in Python. A determined model of the intersection is used to ensure that the fitness values used by the evolutionary optimizer remain stable. However, the solution is subsequently verified using the full mod-el, including its stochastic elements, in order to be able to verify the resulting solution as well as to compare it with a static signal timing plan designed using standard methods.

prof. Ing. Aleš Janota, PhD., EurIng

Ing. Lukáš Slováček

Ing. Michal Gregor, PhD.

Žilinská univerzita
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 1, 010 26 Žilina
E-mail: ales.janota@fel.uniza.sk
E-mail: michal.gregor@fel.uniza.sk

BEZPEČNOSŤ STATICKEJ DOPRAVY

Rastislav Pirník, Dušan Nemec, Marián Hruboš

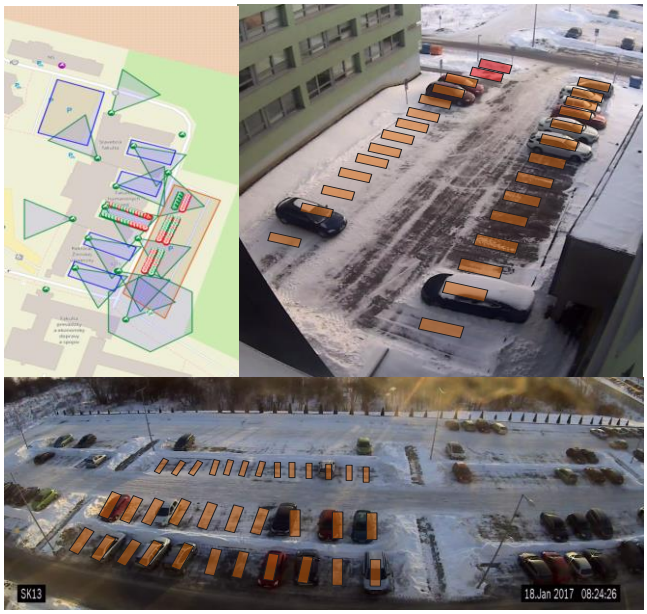
Abstrakt

Článok sa zaoberá návrhom a realizáciou systému na zabezpečenie dohľadu nad vozidlami na parkovisku. Návrh a realizácia takéhoto systému, za pomoci video kamier a príslušného SW a HW vybavenia, nám umožnila včas upozorniť majiteľa vozidla o pohybe (v nepriaznivom prípade ide o odcudzenie) jeho vozidla. Na základe schváleného úžitkového vzoru PUV 96-2015 "Automatizovaný systém monitorovania a stráženia dopravných prostriedkov na odstavných plochách" a známych poznatkov zo spracovania obrazu a počítačového videnia bol vytvorený detekčný systém založený na NNDetektore s podporou neurónových sietí.

Kľúčové slová: statická doprava, riadenie, IKT

Úvod

Hlavná myšlienka navrhnutého systému spočíva v poskytnutí nezávislej služby dohľadu nad vozidlom v miestach parkovacích zón, ktoré sú celé alebo aspoň sčasti pokryté kamerovými systémami. Na tieto účely by sa podľa doterajších znalostí dali použiť aj stávajúce dohľadové kamerové systémy nasadené v centrálnych zónach. Koncept bol rozpracovaný pre otvorené typy parkovísk, ktoré sú integrované do mestských parkovacích zón a takúto funkcionalitu doposiaľ neposkytujú. Na obr. 1 je vyznačené principiálne pokrytie časti parkoviska Žilinskej univerzity aj s vyznačenými detekčnými plochami (žltý obdĺžnik nad miestom vyznačeného státia).



Obr. 1 Koncepčné riešenie bezpečného systému parkovania na modelovom parkovisku UNIZA

Fig. 1 A conceptual solution for a safe parking system at UNIZA

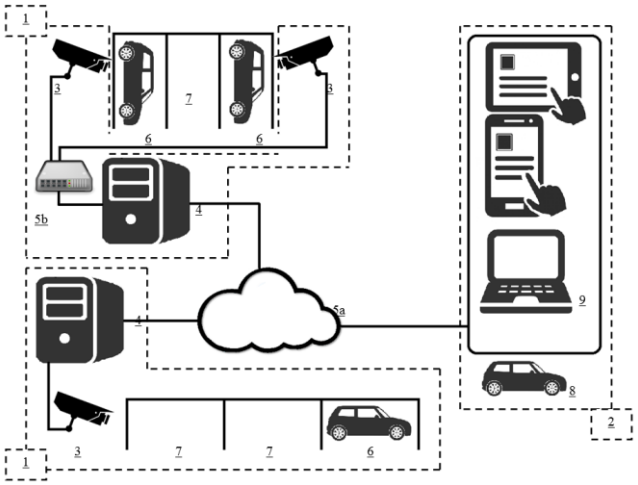
1. Systém RUNIZA

V súčasnej praxi sa na monitorovanie verejných priestranstiev, ulíc, odstavných plôch a dopravných prostriedkov používajú kamerové systémy. Kamerové systémy sú často prepojené komunikačným systémom/podsystémom, ktorý funguje centralizovane, rýchlosť odozvy signalizácie a monitorovania nie je vždy dostačujúca (nie je v reálnom čase) a flexibilita zobrazovania na výstupných zariadeniach je nízka (je predvolené výstupné zariadenie, bez možnosti adaptability). Zároveň je veľmi častým spôsobom monitorovania automobilov na parkovacích miestach a odstavných plochách fyzická kontrola.

Navrhnuté technické riešenie v podobe automatizovaného systému monitorovania a stráženia dopravných prostriedkov na odstavných plochách RUNIZA, umožňuje používateľovi flexibilitu a variabilitu za pomoci zobrazovacích terminálov (smartfón, tablet, ultrabook, atď.). Pripojenie zobrazovacích terminálov je možné prostredníctvom dátovej siete v reálnom čase, pričom špecializovaný softvér automaticky monitoruje a stráži dopravný prostriedok. Navrhnutý systém je nadstavbou klasických spôsobov stráženia prostredníctvom strážnej služby. Systém pomocou kamier a vyhodnocovacieho softvéru analyzuje situáciu na danom parkovacom mieste a v prípade nepovoleného manipulovania s automobilom upozorní oprávneného používateľa automobilu o tejto skutočnosti. Tak umožňuje používateľovi dopravného prostriedku pomocou mobilného zobrazovacieho zariadenia (smartfón, tablet, notebook, atď.) automatizovane monitorovať svoj zaparkovaný dopravný prostriedok.

Systém RUNIZA teda poskytuje koncovému používateľovi bezpečnostnú funkciu (službu) dohľadu nad jeho vozidlom. Výstupom systému (bezpečnostnej funkcie) je okamžité zaslanie informácie o odcudzení vozidla. Táto informácia je používateľovi zaslaná prostredníctvom emailu a SMS.

Automatizovaný systém obr. 2. (časť 1) umožňuje monitorovanie a stráženie dopravných prostriedkov (časť 8) parkujúcich na odstavných plochách (časť 6) a nachádzať voľné parkovacie miesta (časť 7) [4].

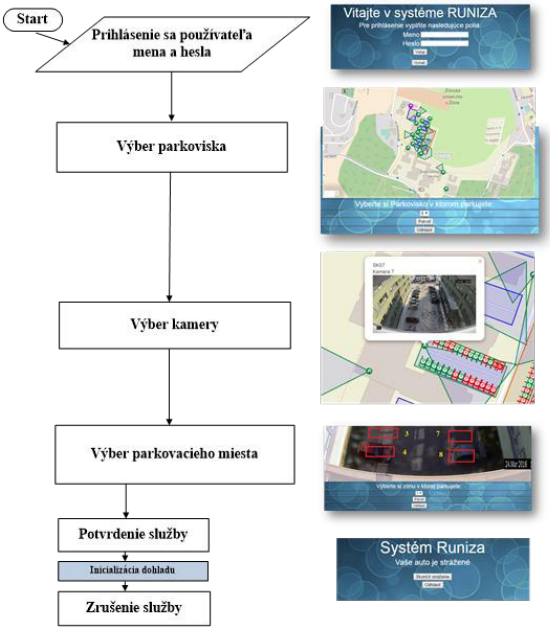


Obr. 2 Koncepcia systému RUNIZA

Fig. 2 Concept of the RUNIZA system

1.1 RUNIZA

Systém RUNIZA bol realizovaný na základe platného úžitkového vzoru **Chyba! Nenašiel sa žiaden zdroj odkazov.**. Používateľ sa pomocou webového klienta pripojí na webový server a na začiatku vyplní prihlasovacie údaje (obr. 3). Potvrdením prihlasovacích údajov si používateľ vyberie číslo kamery, ktorá robí dohľad nad jeho vozidlom. Potvrdením tohto výberu si systém doplní reťazec premenných MENO, KAMERA, ZONA, EMAIL, TEL.. Následne zmena hodnoty premennej STRAZ spustí podprogram VIDEODOHLAD pre dané parkovacie miesto.

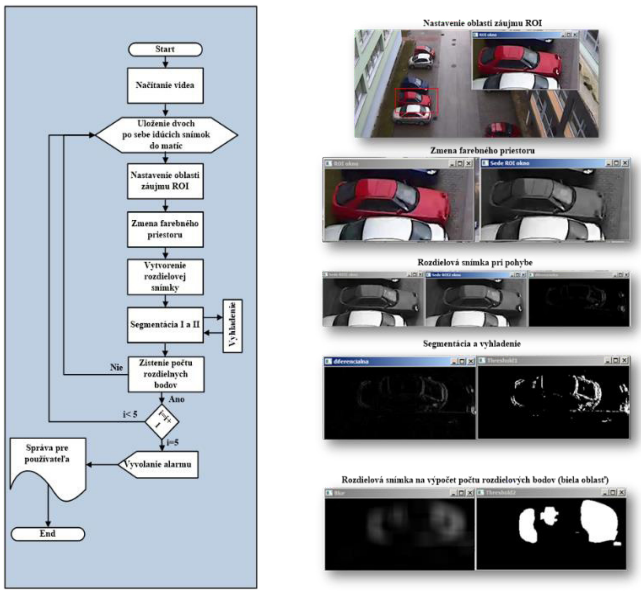


Obr. 3 Funkcionalita systému RUNIZA

Fig. 3 Workflow of the RUNIZA system

1.1.1 Detekcia vozidla na princípe ROI

Počas video analýzy (behu podprogramu VIDEODOHLAD) sa vykonávali potrebné kroky (obr. 4.) pre spracovanie obrazovej informácie, pričom základným krokom bolo porovnanie dvoch po sebe idúcich snímok (konkrétneho ROI - Region Of Interest – oblasť záujmu prideleného parkovaciemu miestu). Týmto sa získal počet rozdielnych bodov vo zvolených obrazových výrezoch. Ak bol počet rozdielnych bodov väčší ako 6,67% (1/15) počtu celkových bodov poľa ROI, vykonalo sa navýšenie premennej *i* o hodnotu 1 (*i*=*i*+1;). Ak sa takéto navýšenie vykonalo päťkrát po sebe, VIDEODOHLAD vyhodnotil potenciálny pohyb v obraze a vyvolal funkcie na odoslanie emailu a SMS správy.



Obr. 4 Algoritmus sekvencie VIDEODOHLAD

Fig. 4 Sequence of the VIDEODOHLAD algorithm

Počas skúšobnej prevádzky systému bolo identifikovaných niekoľko nepriaznivých faktorov, ktoré dokázali výrazne ovplyvniť kvalitu detekcie založenej na analýze ROI, ako sú zlé poveternostné podmienky, zníženie osvetlenia scény počas noci a pod.. Výrazný problém však vyvolávalo rozlíšenia snímača kamery a veľkosť dátového toku (podľa použitej kompresie) nevyhnutného na on-line prenos nespracovanej informácie. Pri vyššom počte ROI (viac ako 40 parkovacích miest) v jednom zábere kamery dochádza pri nízkych rozlíšeniach a vysokej kompresii k chybám neumožňujúcim spoľahlivú detekcie pohybu vozidla. Pri vysokých rozlíšeniach a vysokom dátovom toku naopak dochádzalo k výpadkom funkčnosti systému kvôli SW a HW prostriedkom nasadeného aplikačného servera (tab. 1.). Preto sa pri aktualizácii systému pristúpilo k zmene spôsobu detekcie vozidiel v podsystéme VIDEODOHLAD pomocou NNDetektora založeného na princípoch neurónovej siete.

Počet ROI - 40	Dátový tok [kb/s]					
Rozlíšenie	50	10	1500	200	500	10000
480p (768x432)						
720p (1280x720)						
1080p						
5MP						

Tab.1 Závislosť funkčnosti systému od rozlíšenia a dátového toku

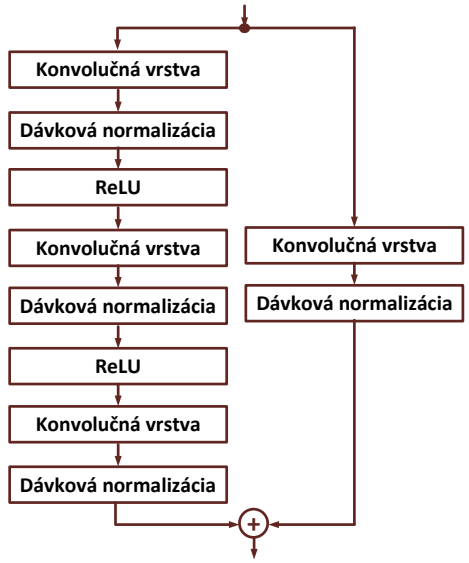
1.1.2 Detekcia s použitím neurónových sietí s hlbokým učením NNDetektor

Systémy detekcie voľných parkovacích miest (detekcie vozidla) na princípe klasifikátora založeného na princípoch neurónových sietí pre parkovacie plochy v extraviláne od roku 2015 získavajú veľkú popularitu a vzniká veľký počet praktických realizácií. Avšak spoločným problémom, s ktorým výskumníci a tvorcovia v tejto oblasti veľmi často čelia, je nedostatok reprezentatívneho súboru údajov pre potreby učenia/trénovania umelých neurónových sietí. Na natrénovanie a následné overenie činnosti NNDetektora sme preto použili súbor parkovacej sady PKLot - "A Robust Dataset for

Parking Lot Classication" http://web.inf.ufpr.br/vri/news/parking-lot-database), ktorý pozostáva zo snímok zachytených z dvoch parkovacích miest s tromi rôznymi pozíciami kamery.

V nami realizovanej neurónovej sieti (NNDetektore) sme využívali dva typy reziduálnych blokov, známe z architektúry ResNet50 (https://arxiv.org/abs/1512.03385):

- identitný reziduálny blok a
- konvolučný reziduálny blok (obr. 5).



Obr. 5 Konvolučný reziduálny blok

Fig. 5 Convolutional residual block

Identitný reziduálny blok pozostáva z podobnej štruktúry:

- konvolučná vrstva,
- dávková normalizácia,
- aktivačná funkcia,
- konvolučná vrstva,
- dávková normalizácia,
- aktivačná funkcia,
- konvolučná vrstva,
- dávková normalizácia.

K výstupu vymenovaných operácií sa na záver pripočítava pôvodný vstup reziduálneho bloku, čo nazývame skratkovým spojením. Ako aktivačná funkcia v našom prípade sa používa rektifikovaná lineárna funkcia (ReLU).

Zapojenia identitného a konvolučného reziduálneho bloku sa líšia len v tom, že v konvolučnom reziduálnom bloku sa v rámci skratkového spojenia aplikuje ešte konvolučná vrstva a dávková normalizácia. Pôvodný vstup sa pripočíta ku výstupu následne, až potom, ako sa naň aplikujú tieto dve operácie.

Okrem reziduálnych blokov sieť používa naviac združovacie vrstvy (pričom prvá vyberá maximum a druhá priemer hodnôt). V poslednej vrstve používame ako aktivačnú funkciu softmax funkciu, ktorá sa štandardne používa pri klasifikácii a zabezpečuje, že výstupy všetkých neurónov sa sčítajú na 1, takže sa následne dajú interpretovať ako pravdepodobnosti príslušných tried.

Získané dáta (celá dátová množina PKLot obsahuje 695851 obrázkov) boli delené v pomere 70%:25%:5% a to na trénovanie a testovacie dáta. Posledných 5% dát sa použilo ako validačná množina, pomocou ktorej sa realizovalo skoré ukončenie učenia. Z toho sa na učenie používa 487095 obrázkov, na testovanie 173963 obrázkov a na validáciu 34793 obrázkov. Dátová množina bola v tvare vhodnom na zostavenie nami požadovaného binárneho klasifikátora, pretože dáta sú rozdelené do dvoch tried (obsadené/neobsadené miesto).

V pôvodných dátach tvoria 48.54% obsadené a zvyšok neobsadené miesta. Rozdelenie na trénovacie a testovacie dáta sa realizovalo stratifikovaným výberom. Pomer tried tým zostal v tréningovej a testovacej množine rovnaký ako v pôvodných dátach.

Po natrénovaní NNDetektora sme dosiahli na testovacích dátach PKLOT nasledujúce výsledky:

	Správne klasifikované	Nesprávne klasifikované	
Voľné miesta	89465	53	**Celková správnosť**
Obsadené miesta	84365	81	**0,9992354**

Keďže databáza PKlot neobsahovala žiadne snímky zobrazujúce snehovú prikrývku na mieste parkovacích plôch (ide o klasické klimatické podmienky v strednej a severnej Európe), vytvorila sa nová databáza UNIZAS. Databáza UNIZAS sa tvorila od decembra 2017 do januára 2018 na parkovisku Žilinskej Univerzity v Žiline a zachytáva variability a scény: sneženie, slnečný jas, a hmla. Monitorované parkoviská UNIZA obsahujú viac ako 550 parkovacích miest (na obr. 6 je uvedený príklad jedného segmentu parkoviska).

Obr. 6 Parkovacie segmenty

Fig. 6 Parking segments

Zrealizovaný a natrénovaný NNDetektor sme následne nechali klasifikovať snímky zo súboru UNIZAS:

	Správne klasifikované	Nesprávne klasifikované	
Voľné miesta	283	153	**Celková správnosť**
Obsadené miesta	521	58	**0,7921182**

Je zrejmé, že úspešnosť detekcie výrazne klesla, keďže pôvodná dátová množina PKLot neobsahovala snímky zo snehovou prikrívkou. Aby dokázal detektor takéto vstupy správne klasifikovať, bolo potrebné NNDetektor dotrénovať na dátach UNIZAS. Pri trénovaní sa použilo delenie UNIZAS dát v známom pomere 70%:25%:5% na trénovacie, testovacie a validačné dáta pričom sa aplikovalo transfer učenie.

Výsledné hodnoty po dotrénovaní NNDetektora snímkami zo súboru UNIZAS:

	Správne klasifikované	Nesprávne klasifikované	
Voľné miesta	433	3	**Celková správnosť**
Obsadené miesta	566	13	**0,9842364**

Záver

Dosiahnuté výsledky potvrdzujú počiatočnú domnienku, že požadovaná schopnosť NNDetektora detegovať vozidlá v oblastiach s nepriaznivými poveternostnými podmienkami (snehovou prikrývkou) sa výrazne zvýši po dotrénovaní na lokálnej množine dát/snímok z pôvodných hodnôt pre správnosť detekcie v rozmedzí 80-89% až na prijateľné hodnoty presahujúce 98% úspešnosť. Problém nastal jedine pri snímkach, ktoré zachytávali situáciu na parkovisku pri zarosenej kamere, kde boli dosahované hodnoty úspešnej detekcie pod 80%.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] BUBENIKOVA, E., PIRNIK, R., HOLECKO, P.: Optimisation of video-data transmission in telematic system. In: Advances in electrical and electronic engineering. - ISSN 1336-1376. - Vol. 11, no. 2 spec. iss. (2013), p. 123-134

[2] BUBENÍKOVÁ, E., FRANEKOVÁ, M., HOLEČKO, P: Conceptual design of driving lane-crossing alarm threshold in C-ITS applications and its implementation, In.: 2016 Cybernetics & Informatics (K&I) 2016, 2-5 February 2016, Levoča, Slovakia, ISBN 978-1-5090-1832-1, p.p. 1-6,

[3] opencv.org. (2015, Dec.) opencv.org. [Online]. http://opencv.org/about.html [Dec. 12, 2016].

[4] ÚRAD PRIEMYSELNÉHO VLASTNÍCTVA SLOVENSKEJ REPUBLIKY Automatizovaný systém monitorovania a stráženia dopravných prostriedkov na odstavných plochách, Žilinská univerzita v Žiline, PIRNÍK R, a kol. Slovenský republika PUV 96-2015. 16.11.2016.

Abstract

The paper deals with design and realization of the system monitoring vehicles at the open type of parking area RUNIZA. Design and realization of such a system, with the use of video-cameras and relevant SW and HW equipment, help to warn a vehicle's owner of movement of his/her vehicle (in an adverse case of its theft). Information about movement is sent by email or SMS message. The system analysing images from cameras has been realized based on the approved utility model PUV 96-2015 "Automated system for monitoring and guarding vehicles on parking areas" and known knowledge about image processing and computer vision. The system has been implemented and tested within the University of Žilina Campus.

doc. Ing. Rastislav Pirník, PhD.

Ing. Dušan Nemec, PhD.

Ing. Marián Hruboš, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 8215/1
010 26 Žilina
+421 41 513 3301

{rastislav.pirnik;dusan.nemec;marian.hrubos}@fel.uniza.sk

Obsah

NIEKTORÉ PROBLÉMY SÚVISIACE S BEZPEČNOSŤOU PREVÁDZKY NA PRIECESTIACH ŽSR

Karol Rástočný, Peter Nagy

Abstrakt
Železničná doprava patrí medzi procesy, ktoré sú riadené s určitou mierou rizika. Je zrejmé, že vzhľadom na úroveň našich znalostí, dosiahnutú technickú úroveň riadiacich systémov a obmedzené finančné prostriedky nemôžeme počítať s absolútnou bezpečnosťou a teda požadovať nulové riziko dopravného procesu. Musíme pripustiť, že v reálnom technickom systéme sa môže vyskytnúť chyba alebo porucha a jej výskyt môže znamenať vznik určitého rizika pre riadený dopravný proces. Hlavná pozornosť autorov je venovaná návrhu a predstaveniu potenciálne využiteľných opatrení, ktoré by mohli zvýšiť bezpečnosť prevádzky na úrovňových železničných priecestiach prevádzkovaných na tratiach ŽSR. Samostatne sú diskutované technické a organizačné opatrenia. Niektoré z navrhovaných opatrení sú špecifické len pre slovenské podmienky, avšak do určitej miery možno niektoré zistenia zovšeobecniť a prípadne uplatniť aj v iných krajinách.

Kľúčové slová: bezpečnosť, železničné priecestie, ŽSR, technické opatrenia, organizačné opatrenia, železničná doprava

Úvod

V súčasnosti sa na tratiach ŽSR nachádza 2092 úrovňových priecestí (stav k 1.6.2018). Z toho je 1019 nezabezpečených (nie sú vybavené priecestným zariadením, resp. priecestným zabezpečovacím systémom) a 1073 zabezpečených (sú vybavené priecestným zariadením, resp. priecestným zabezpečovacím systémom) priecestí [1].

Ak úrovňové križovanie cestnej a železničnej komunikácie (ďalej len priecestie) nie je vybavené priecestným zabezpečovacím systémom (PZS), potom bezpečnosť pohybu účastníkov cestnej dopravy na takomto priecestí je zaisťovaná organizačnými opatreniami. Priecestie musí byť označené výstražným krížom a účastník cestnej dopravy je informovaný o tom, že sa blíži k priecestiu návestnou tabuľou. Podľa zákona [2] je vodič cestného vozidla pred priecestím povinný počínať si mimoriadne opatrne a presvedčiť sa, či môže bezpečne prejsť cez priecestie. Vo vzdialenosti 30 m pred priecestím a pri jeho prechádzaní je vodič povinný jazdiť rýchlosťou najviac 30 km.h^{-1}, resp. rýchlosťou najviac 50 km.h^{-1}, ak na výstražníku sa dáva aktívna signalizácia (prerušované biele svetlo).

Ak je priecestie vybavené PZS, potom účastník cestnej dopravy môže byť informovaný o blížiacom sa vlaku (vo všeobecnosti o blížiacom sa železničnom vozidle) k priecestiu:

- zvukovou výstrahou (akustický signál, ktorého zdrojom je mechanické alebo elektrické zariadenie – zvonec, húkačka);
- svetelnou výstrahou (dve červené striedavo prerušované svetlá umiestnené vedľa seba);
- mechanickou výstrahou (závory prehradzujúce pozemnú komunikáciu resp. jej časť).

V článku sa budeme zaoberať len PZS so základnou svetelnou výstrahou a s doplnkovou mechanickou výstrahou (mechanická výstraha môže a nemusí byť). Takéto PZS sa používajú na tratiach ŽSR s traťovou rýchlosťou do 140 km.h^{-1}. Na tratiach s traťovou rýchlosťou rovnou alebo väčšou ako 140 km.h^{-1} sa budujú PZS, ktoré musia mať aj mechanickú výstrahu, pričom rameno závory musí byť v dolnej polohe (vo výstražnom stave) blokované a musí byť kontrolovaná jeho celistvosť. Na tratiach ŽSR s traťovou rýchlosťou rovnou alebo väčšou ako 160 km.h^{-1} je budovanie PZS zakázané.

Aby účastník cestnej dopravy bol presne informovaný o pohybe železničných vozidiel v oblasti priecestia je nutné, aby:

- poskytované informácie boli jednoznačné;
- PZS vykonával svoju funkciu v zhode s funkčnou špecifikáciou (norma [3]);
- bola PZS mal čo najväčšiu pohotovosť.

Zaistenie týchto základných požiadaviek je spojené s určitými problémami špecifickými pre ŽSR. Na niektoré z nich poukazuje tento príspevok.

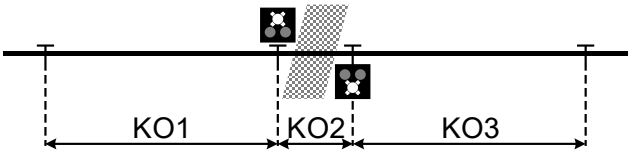
1. Ovládanie PZS jazdou vlaku

PZS vyžaduje pre svoju automatickú činnosť spoluprácu s technickými prostriedkami na vyhodnotenie prítomnosti železničného vozidla v obvode priecestia a na vyhodnotenie prejazdu železničného vozidla priecestím. Na tento cieľ možno použiť:

- líniové prostriedky (koľajové obvody, počítače osí),
- bodové prostriedky (koľajnicové spínače, koľajové slučky).

1.1 Koľajové obvody

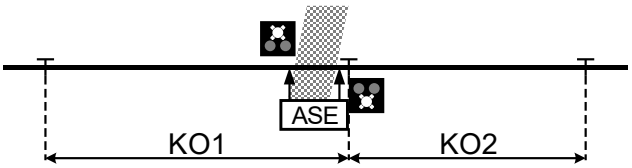
Jednoznačné informovanie účastníkov cestnej dopravy o stave železničnej dopravy v obvode priecestia na trati s obojsmernou prevádzkou si vyžaduje použitie minimálne troch koľajových obvodov (obr. 1). Spravidla sa v približovacích úsekoch (KO1 a KO3) používajú paralelné koľajové obvody, ktoré slúžia na detekciu voľnosti približovacieho úseku. Koľajový obvod v obvode priecestia (KO2), ktorý slúži na detekciu prejazdu vlaku cez priecestie, sa konštruuje tiež ako paralelný alebo sa prejazd železničného vozidla cez priecestie vyhodnocuje iným spôsobom.



Obr.1 Priecestie s 3 koľajovými obvodmi

Fig.1 Level crossing with 3 track circuits

Pre prevádzku ŽSR je typické použitie koľajových obvodov v konfigurácii podľa obr.2, kde koľajový obvod KO2, ktorý vyhodnocuje prejazd železničného vozidla cez priecestie je nahradený súborom ASE. V prípade súboru ASE ide o dva sériové neohraničené koľajové obvody, ktoré sa navzájom prekrývajú [8].



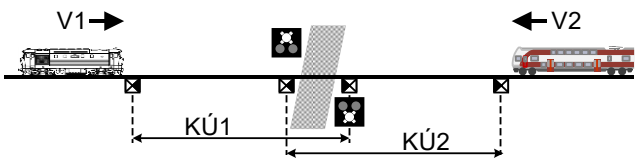
Obr.2 Priecestie s 2 paralelnými KO a ASE

Fig.1 Level crossing with 2 closed track circuits and an ASE

Správna činnosť koľajových obvodov závisí od ich šuntovej citlivosti a veľkosti pôsobiaceho vlakového šuntu. Činnosť koľajových obvodov môže byť ovplyvňovaná mnohými faktormi, medzi ktoré patria počasie, intenzita vlakovej dopravy, hmotnosť vlakov a ďalšie. Na vedľajších tratiach s nízkou intenzitou dopravy reálne hrozí možnosť straty šuntu a v dôsledku toho nespustenia výstrahy na priecestí pred prichádzajúcim vlakom. Z tohto dôvodu sa na tratiach ŽSR už viac ako 10 rokov nové PZS s koľajovými obvodmi nezriaďujú.

1.2 Počítače osí

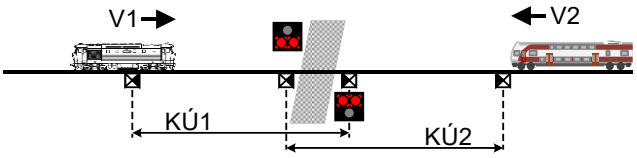
V nedávnej minulosti ŽSR prijali strategické rozhodnutie, že novo sa už koľajové obvody nebudú budovať. Preto pri rekonštrukciách tratí s priecestiami, bolo treba zaoberať sa problematikou náhrady koľajových obvodov používaných na ovládanie už existujúcich PZS. Ako reálna sa ukázala možnosť použitia dvoch počítačov osí tak, že ich koľajové (čítacie) úseky sa navzájom prekrývajú. Prekrývanie koľajových úsekov (KÚ) umožňuje vyhodnotiť prejazd vlaku cez priecestie a na základe tejto informácie ukončiť výstražný stav PZS.



Obr. 3 Jazda dvoch vlakov v obvode priecestia – základný stav

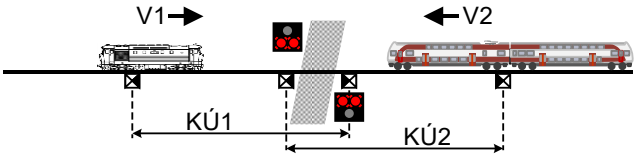
Fig. 3 The run of two trains in a level crossing area – initial state

Pri analýze rôznych prevádzkových situácií bolo identifikované nebezpečenstvo, vyplývajúce z protismernej jazdy dvoch vlakov v obvode priecestia (obr. 3.). Takáto prevádzková situácia je znázornená na obr. 4 až obr. 6.



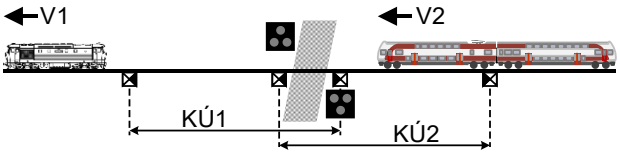
Obr. 4 Jazda dvoch vlakov v obvode priecestia – obsadený úsek KÚ1

Fig. 4 The run of two trains in a level crossing area – track section KÚ1 is occupied



Obr. 5 Jazda dvoch vlakov v obvode priecestia – obsadené úseky KÚ1 aj KÚ2

Fig. 5 The run of two trains in a level crossing area – track sections KÚ1 and KÚ2 are occupied



Obr. 6 Jazda dvoch vlakov v obvode priecestia – uvoľnenie úseku KÚ1, úsek KÚ2 zostáva obsadený

Fig. 6 The run of two trains in a level crossing area – clearing of the KÚ1 section while section KÚ2 remains occupied

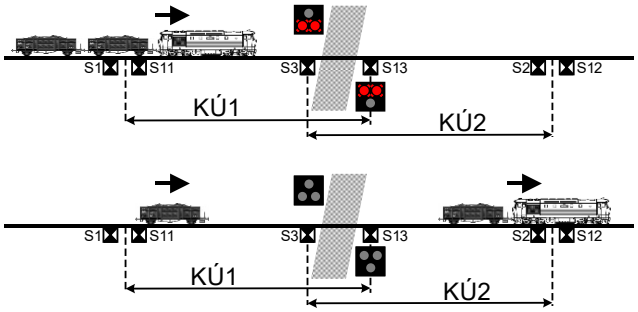
Nech vlak V1 obsadí koľajový úsek KÚ1 a nech v dôsledku tejto udalosti prejde PZS do výstražného stavu (obr. 4). Ak následne vlak V2 obsadí koľajový úsek KÚ2, výstražný stav pokračuje a logika PZS vyhodnotí túto situáciu tak, že vlak V1 obsadzuje koľajový úsek KÚ1 aj koľajový úsek KÚ2, čo však nezodpovedá skutočnosti (obr. 5). Ak z nejakých prevádzkových dôvodov vlak V1 zmení smer jazdy a uvoľní koľajový úsek KÚ1, PZS prejde do anulačného stavu a ukončí sa dávanie výstrahy. Vlak V2 príde na priecestie, na ktorom nie je dávaná výstraha – ide o potenciálne nebezpečný stav. Obdobná situácia môže nastať aj pre opačný smer dopravy. Takáto činnosť PZS je v rozpore s filozofiou zabezpečovacej techniky a aj v rozpore s požiadavkami na PZS [3].

Riziko spojené s opísanou prevádzkovou situáciou možno redukovať vyhodnotením smeru pohybu železničného vozidla a zapracovaním tejto informácie do logiky už existujúceho PZS tak, aby porucha samotného počítača osí nemohla spôsobiť potenciálne nebezpečnú situáciu.

1.3 Bodové prostriedky

Norma [3] vytvára priestor na použitie bodových technických prostriedkov na ovládanie PZS. Bodové technické prostriedky však neumožňujú bezpečne kontrolovať voľnosť približovacieho úseku a ani vyhodnotiť prejazd celého vlaku priecestím. Napríklad pri roztrhnutí vlaku (obr. 7) hrozí, že odtrhnutá časť vlaku príde na priecestie, ktoré je v anulačnom stave (nie je dávaná výstraha) po prejazde prvej časti vlaku. Určité zníženie rizika možno dosiahnuť spustením výstrahy, ak dôjde k neočakávanému obsadeniu snímača S3, resp.

S13 (v prípade jazdy vlaku v opačnom smere) Riziko vyplývajúce z tejto prevádzkovej situácie bolo zo strany ŽSR posúdené ako tolerovateľné riziko.



Obr. 7 PZS ovládané bodovými prostriedkami

Fig. 7 LCS controlled by point devices

2. Aktívna signalizácia

Určitou zvláštnosťou ŽSR je používanie aktívnej signalizácie. Aktívna signalizácia sa dáva prerušovaným svietením bieleho svetla umiestneného na výstražníku a informuje účastníka cestnej premávky o tom, že v obvode priecestia sa nenachádza železničné vozidlo. Používanie aktívnej signalizácie prináša so sebou niekoľko problémov:

- Väčšina zahraničných účastníkov cestnej premávky takúto signalizáciu nepozná.
- Norma [3] určuje, že aktívnou signalizáciou musia byť vybavené všetky PZS, až na výnimky uvedené v tejto norme. Faktom je, že na ŽSR cca 40 % PZS so svetelnou signalizáciou nemá aktívnu signalizáciu. V mnohých prípadoch zlý výklad predtým platnej normy (predchodkyňa normy [3]) viedol k tomu, že aktívna signalizácia bola argumentom na zdôvodnenie nedodržania rozhľadových pomerov na priecestí. Vyhláška o cestnej doprave, ktorá bola platná do roku 1990 predpokladala, že v prípade dávania aktívnej signalizácie za bezpečnosť na priecestí zodpovedá prevádzkovateľ dráhy a vodič cestného vozidla nie je povinný presvedčiť sa, či môže bezpečne prejsť cez priecestie. Mnohí vodiči (najmä starší) sú o platnosti tohto tvrdenia presvedčení doteraz. Faktom je, že v súčasnosti platný zákon [2] hovorí, že ak svieti prerušované biele svetlo (aktívna signalizácia), vodič je povinný vo vzdialenosti 50 m pred priecestím a pri jeho prechádzaní jazdiť rýchlosťou najviac 50 $km.h^{-1}$.
- Problémom je aj nejednoznačnosť významu informácií poskytovaných účastníkovi cestnej dopravy. Biele svetlo (návestenie aktívnej signalizácie) je umiestnené na výstražníku spoločne s dvoma červenými svetlami (návestenie svetelnej výstrahy). Ak PZS nie je vybavené aktívnou signalizáciou, tak prevádzky neschopný stav PZS (stav, keď PZS nie je schopné informovať účastníka cestnej dopravy o prítomnosti železničného vozidla v oblasti priecestia) je signalizovaný účastníkovi rovnako, ako základný stav (v obvode priecestia nie je železničné vozidlo, ktoré by mohlo ohroziť bezpečnosť cestnej premávky na priecestí). Vodič cestného vozidla v niektorých prípadoch (napríklad v noci) tak nemusí a ani nie je schopný rozlíšiť, či ide o PZS s aktívnou signalizáciou alebo bez aktívnej signalizácie.

Keďže aktívnou signalizáciou nie sú vybavené všetky PZS v železničnej sieti ŽSR, tak jej príspevok k bezpečnosti dopravy na priecestiach je diskutabilný.

3. Informovanie rušňovodiča o stave PZS

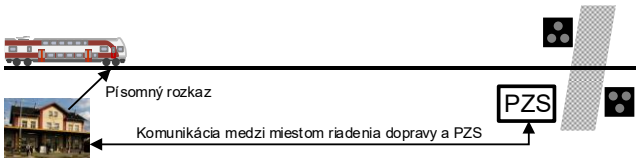
V prípade poruchy PZS musí rušňovodič znížiť rýchlosť jazdy vlaku a húkačkou upozorňovať účastníkov cestnej premávky o tom, že k priecestiu sa blíži vlak. K tomu je ale nevyhnutné informovať rušňovodiča o stave PZS. Rušňovodič môže byť o stave PZS informovaný tak, že informácia o stave PZS je prenášaná:

- dopravnému zamestnancovi do najbližšej dopravne (ten následne informuje rušňovodiča);
- na hlavné návestidlo kryjúce priecestie;
- na priecestník;
- na hnacie vozidlo (takéto riešenie sa na ŽSR nepoužíva).

Spoločným nedostatkom týchto riešení je to, že informácia je poskytovaná rušňovodičovi bodovo (okrem riešenia, keď informácia je prenášaná na hnacie vozidlo) v určitom mieste na trati. V prípade takej poruchy, že PZS nie je schopné prejsť do výstražného stavu a vlak sa už nachádza medzi miestom odovzdania informácie a PZS, rušňovodič príde na priecestie, ktoré nie je uzatvorené a pritom nezníži rýchlosť vlaku a neupozorňuje účastníkov cestnej dopravy.

3.1 Informovanie rušňovodiča prostredníctvom dopravného zamestnanca

Ide o najmenej dokonalý spôsob informovania rušňovodiča o stave priecestia, ktorý sa používa na tratiach bez traťového zabezpečovacieho zariadenia alebo s na tratiach vybavených poloautomatickým blokom. Komunikácia medzi miestom riadenia dopravy a PZS sa spravidla uskutočňuje po špeciálnom vedení. Dopravný zamestnanec v mieste riadenia dopravy je informovaný o stave PZS a má možnosť diaľkovo priecestie otvoriť alebo uzavrieť. Ak došlo k uzavretiu priecestia v dôsledku kritickej poruchy PZS, priecestie možno núdzovo otvoriť len za predpokladu, že rušňovodiči všetkých vlakov prichádzajúcich na priecestie sú o stave priecestia informovaní. Ak sa medzi miestom riadenia dopravy a PZS už nachádza vlak, priecestie možno núdzovo otvoriť, až po prijatí informácie o tom, že vlak už prešiel cez priecestie (napríklad prijatím informácie z nasledujúcej dopravne). V takomto prípade môže byť priecestie neúmerne dlho uzatvorené, keď zoberieme do úvahy, že takéto riešenia sa na ŽSR používa do vzdialenosti až 20 km medzi kontrolovaným PZS a miestom riadenia dopravy.



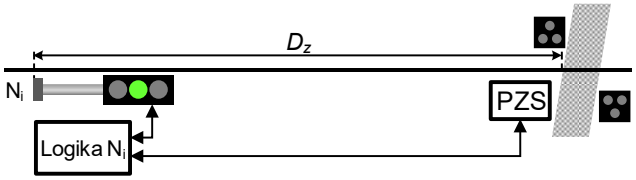
Obr. 8 Informovanie rušňovodiča prostredníctvom dopravného zamestnanca

Fig. 8 Informing an train-driver via an employee

3.2 Informovanie rušňovodiča prostredníctvom hlavného návestidla

Takéto riešenie (obr. 9) sa používa na tratiach s automatickým traťovým zabezpečovacím zariadením – automatickým blokom alebo automatickým hradlom. Najbližšie oddielové návestidlo umiestnené pred priecestím v smere jazdy vlaku má väzbu na PZS. V prípade automatického bloku s permisívnym významom návesti Stoj na oddielových návestidlách musí byť návestidlo umiestnené minimálne na zábrzdnú vzdialenosť. V prípade kritickej poruchy PZS sa na oddielovom návestidle dáva návesť Stoj. Obdobné riešenie sa využíva pre PZS, ktorých obvod priecestia sa prekrýva s obvodom stanice (väzba PZS sa realizuje na vchodové resp. odchodové návestidlá). V prípade kritickej poruchy PZS je na

príslušnom návestidle dávaná návesť Stoj s absolútnym významom a ďalšia jazda vlaku je podmienená súhlasom výpravcu v stanici. Rovnako sa návesť Stoj s absolútnym významom dáva v prípade kritickej poruchy PZS na oddielovom návestidle automatického hradla, pričom ďalšiu jazdu môže povoliť výpravca stanice prijímajúcej vlak daním privolávacej návesti na oddielovom návestidle kryjúcom PZS.



Obr. 9 Väzba PZS na hlavné návestidlo

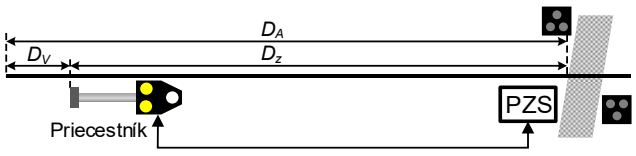
Fig. 9 Main signal and LCS coupling

3.3 Informovanie rušňovodiča prostredníctvom priecestníka

Jednou z možností, ako minimalizovať pravdepodobnosť príchodu železničného vozidla na nezabezpečené priecestie, je prenos informácií o stave PZS smerom k rušňovodičovi železničného vozidla. Na tento účel možno použiť zvláštne samostatné návestidlo tzv. priecestník, na ktorom sa dávajú tieto návesti:

- Pohotovostný stav. Informuje rušňovodiča, že železničné vozidlo smie ísť k priecestiu najvyššou dovolenou rýchlosťou. Technicky to znamená, že PZS je vo výstražnom stave alebo je schopný prejsť do výstražného stavu po obsadení približovacieho úseku železničným vozidlom.
- Poruchový stav. Prikazuje rušňovodičovi znížiť rýchlosť železničného vozidla tak, aby bol schopný bezpečne zastaviť pred prekážkou na priecestí.

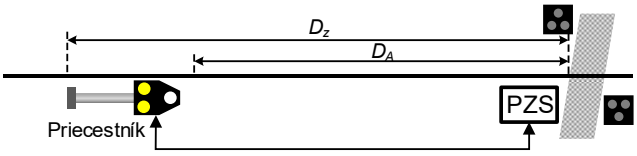
Ak návesť Pohotovostný stav sa dáva, ako dôsledok prechodu PZS do výstražného stavu, tak priecestník musí byť umiestnený vo vnútri približovacieho úseku na zábrzdnú vzdialenosť D_z od priecestia a na požadovanú dohľadnú vzdialenosť D_V od hranice približovacieho úseku (obr. 10). Splnenie tejto požiadavky v mnohých prípadoch vedie k nutnosti zväčšiť dĺžku približovacieho úseku D_A a tým aj predĺžiť čas uzavretia priecestia. Takéto riešenie je síce z bezpečnostného hľadiska správne, ale spôsobuje vážne prevádzkové problémy na priecestiach s vysokou intenzitou dopravy (či už cestnej alebo železničnej).



Obr. 10 Umiestnenie priecestníka v približovacom úseku

Fig. 10 Position of the gate signal inside the approaching section

V súčasnosti platné normy a predpisy umožňujú použiť priecestník tak, že návesť Pohotovostný stav sa dáva vtedy, ak PZS je v prevádzkyschopnom stave. V tomto prípade môže (ale nemusí) byť priecestník umiestnený aj pred približovacím úsekom (obr. 11), minimálne však na zábrzdnú vzdialenosť D_z pred priecestím. Výhodou tohto riešenia je to, že v niektorých prípadoch môže byť priecestie uzavreté pre účastníkov cestnej premávky na čas podstatne kratší ako pri predchádzajúcom riešení. Pozitívny efekt takéhoto riešenia sa prejaví výrazne vtedy, ak je viac priecestí krytých jedným priecestníkom.

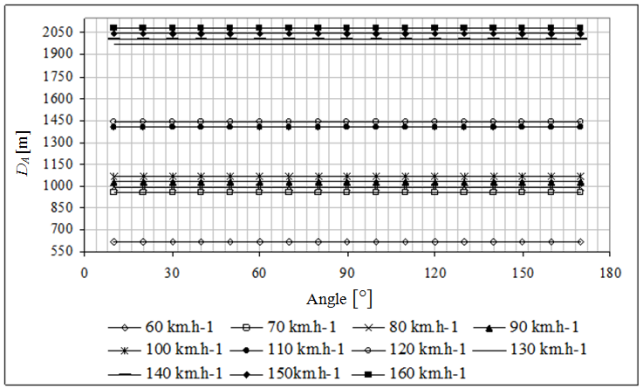


Obr. 11 Umiestnenie priecestníka mimo približovacieho úseku

Fig. 11 Position of the gate signal outside the approaching section

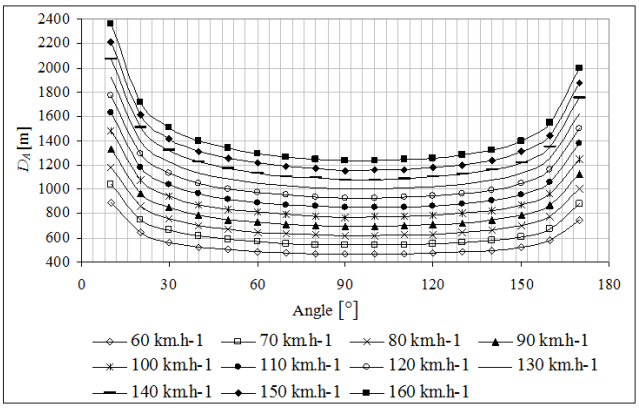
Nutnou podmienkou takéhoto riešenia (obr. 11) je, aby PZS vždy (presnejšie - s definovanou úrovňou integrity bezpečnosti) prešiel do výstražného stavu po obsadení približovacieho úseku železničným vozidlom. Súčasná úroveň techniky umožňuje splniť túto požiadavku. Príkladom môže byť PZS s viackanálovou architektúrou a pravidelným testovaním obvodov svetiel výstražníka, kde postačujúcou podmienkou na spustenie výstrahy je povel aspoň od jedného kanála. Túto podmienku spĺňajú len niektoré novobudované elektronické PZS.

Treba zvážiť, či takéto riešenie je vhodné aj pre reléové PZS, ktoré sú prevádzkované na tratiach ŽSR. Tieto PZS majú jednokanálovú architektúru a nedisponujú kontrolnými mechanizmami, ktoré by boli schopné v pravidelných časových intervaloch kontrolovať funkcieschopnosť obvodov svetiel výstražníka. Schopnosť PZS prejsť do výstražného stavu po obsadení približovacieho úseku musí byť doložená preukazom bezpečnosti.



Obr. 12 Dĺžka približovacieho úseku, ak priecestník informuje o výstražnom stave

Fig. 12 The length of the approaching section when the gate signal informs about warning state



Obr. 13 Dĺžka približovacieho úseku, ak priecestník informuje o prevádzkyschopnom stave

Fig. 13 The length of the approaching section when the gate signal informs about operating state

Na obr. 12 a obr. 13 sú zobrazené grafy znázorňujúce dĺžky približovacích úsekov v závislosti od toho, v akej funkcii je

priecestník použitý. Dĺžky približovacích úsekov boli vypočítané pre PZS bez závor na jednokoľajnej trati pre rôzne traťové rýchlosti a rôzne uhly kríženia trate a cestnej komunikácie. Je zrejmé, že čas uzatvorenia priecestia závisí od dĺžky približovacieho úseku a rýchlosti vlaku, akou sa pohybuje v približovacom úseku.

Záver

Bezpečnosť dopravy na priecestiach je podmienená aplikáciou technických opatrení na zníženie rizika a dodržiavaním organizačných opatrení najmä účastníkmi cestnej premávky. Postupy súvisiace s prevádzkou železničnej dopravy na priecestí určuje predpis [4]. Technické požiadavky na PZS stanovuje norma [3]. Zvýšenie bezpečnosti na železničných priecestiach možno dosiahnuť aplikáciou technických a organizačných opatrení.

Technické opatrenia

Funkcie, ktoré realizujú existujúce PZS sú realizované s úrovňou integrity bezpečnosti SIL4 a jej ďalšie zvyšovanie je veľmi neefektívne, pretože každé ďalšie zvýšenie technickej bezpečnosti si vyžaduje výrazné zvýšenie nákladov. Navyše by to viedlo len k nepatrnému zvýšeniu bezpečnosti dopravy na priecestiach. Zlyhanie PZS sa podieľa na nehodovosti na priecestiach veľmi malou mierou – na tratiach v správe ŽSR je to menej ako 0,1 % (zodpovedá to priemernej hodnote udávanej v rámci EÚ). K zvýšeniu bezpečnosti by prispelo doplnenie PZS o nové funkcie, resp. o úpravy, ktoré síce primárne nezvyšujú technickú bezpečnosť PZS, ale zlepšujú podmienky na dodržiavanie organizačných opatrení účastníkmi cestnej dopravy a tak v konečnom dôsledku výrazne môžu prispieť k bezpečnosti na priecestiach. K zvýšeniu bezpečnosti na priecestiach by prispelo:

- Dôsledné používanie závor. Aj keď používanie závor zvyšuje prevádzkové náklady (násilné poškodzovanie závor, náklady na údržbu) a tiež predlžuje čas uzavretia priecestia [5], je táto požiadavka oprávnená. Zo štatistiky uvedenej v [7] vyplýva, že nehodovosť na priecestiach so závorami je výrazne menšia ako nehodovosť na priecestiach bez závor.
- Uzavretie priecestia maximálne na čas nevyhnutný na bezpečný prechod najpomalšieho a najdlhšieho vlaku cez priecestie. Ak existujú priecestia na trati, kde sa pohybujú vlaky rôznymi rýchlosťami (napríklad vysokorýchlostné trate s kombinovanou dopravou) dochádza k situácii, že pri klasickom ovládaní PZS (s pevným spúšťacím bodom) je pre pomalé vlaky priecestie zbytočne dlho uzatvorené a vodiči majú nutkanie nerešpektovať výstrahu. Tento problém možno riešiť pomocou vyrovnávača približovacej doby [6].
- Informovanie rušňovodiča o stave PZS s cieľom minimalizovať pravdepodobnosť príchodu železničného vozidla na nezabezpečené priecestie. V prípade informácie o poruchovom stave PZS môže rušňovodič znížiť rýchlosť železničného vozidla tak, aby bol schopný pred prekážkou zastaviť. Na tento cieľ prednostne používať priecestník alebo väzbu na hlavné návestidlo. Optimálnym riešením je prenos informácie o stave PZS na hnacie vozidlo približujúce sa k priecestiu.
- Kontrolovanie priestoru priecestia. Použitie kamerového systému alebo iného technického systému na identifikáciu prekážky na priecestí má význam má len vtedy, ak je informácia o prekážke včas poskytnutá rušňovodičovi, aby mohol začať intenzívne brzdiť s cieľom zastaviť pred prekážkou na priecestí. V prípade takýchto systémov je veľmi problematická spoľahlivosť a bezpečnosť kontroly priecestia, pretože aj falošné hlásenie o prekážke je spojené s aktiváciou brzdného systému vlaku, čo môže spôsobiť úraz cestujúcich vo vlaku.
- Jednoznačný výklad informácií poskytovaných účastníkom cestnej dopravy. V prípade kritickej poruchy sa používa na priecestiach ŽSR také riešenie, že dovtedy, kým nie je o poruche PZS informovaný rušňovodič, tak je aktivovaná výstraha (ak je to technicky možné) a priecestie sa javí zo stany účastníka cestnej dopravy ako uzatvorené. Ak je o poruche PZS informovaný rušňovodič, výstraha sa nedáva a priecestie sa javí zo stany účastníka cestnej dopravy ako otvorené. Z bezpečnostného hľadiska by bolo žiaduce, aby v prípade poruchy zostalo priecestie uzatvorené až do okamihu, kým o tejto situácii nie je informovaný rušňovodič vlaku blížiaceho sa k priecestiu a to na takú vzdialenosť, aby dokázal znížiť rýchlosť na takú hodnotu, aby bol schopný zastaviť pred prekážkou. Dlhodobo zatvorené priecestie nevhodne pôsobí na psychiku vodiča cestného vozidla a vedie k situácii, že vodiči prechádzajú cez priecestie aj počas dávania výstrahy.
- Zjednotenie spôsobu informovania účastníkov cestnej dopravy o stave PZS v rôznych štátoch EÚ. V tomto prípade ide o problém, ktorý je síce technicky riešiteľný, ale praktické použitie je nereálne (naráža predovšetkým na legislatívne a ekonomické prekážky). predovšetkým legislatívnym problémom.
- Vysoká pohotovosť PZS, ktorá minimalizuje prípady, keď PZS je nefunkčné a bezpečnosť na priecestí je závislá len od dodržiavania organizačných opatrení, či už zo strany pracovníkov ŽSR alebo účastníkov cestnej dopravy. Navyše si treba uvedomiť, že v niektorých prípadoch nie sú vyhovujúce ani rozhľadové pomery na priecestí.

Organizačné opatrenia

Podľa zákona [2] je vodič pred priecestím povinný počínať si mimoriadne opatrne a za každej situácie sa musí presvedčiť, či môže bezpečne prejsť cez priecestie. Vzhľadom k takejto dikcii zákona už nie je prakticky možné ďalšie sprísňovanie organizačných opatrení. K zvýšeniu bezpečnosti na priecestiach by určite pomohla dôslednejšia kontrola a vymáhanie dodržiavania organizačných opatrení účastníkmi cestnej dopravy a zvýšené represívne postihy pri porušení týchto opatrení. V každom prípade sa treba systematicky venovať výchove a osvete účastníkov cestnej dopravy.

Poďakovanie

Tato publikácia vznikla vďaka podpore v rámci operačného programu Výskum a vývoj pre projekt:

Centrum excelentnosti pre systémy a služby inteligentnej dopravy II., ITMS 26220120050 spolufinancovaný zo zdrojov Európskeho fondu regionálneho rozvoja.

"Podporujeme výskumné aktivity na Slovensku/Projekt je spolufinancovaný zo zdrojov EÚ"

Literatúra

[1] ŽSR: Výročná správa 2018. http://www.zsr.sk.

[2] Zákon č 315/96 z.z. o premávke na pozemných komunikáciách a s ním súvisiace vyhlášky a predpisy.

[3] STN P 34 2651 Železničné priecestné zariadenia, 1999.

[4] Predpis ŽSR Z1 Pravidlá železničnej prevádzky, 2011.

[5] JANOTA, A.,RÁSTOČNÝ, K., ZAHRADNÍK, J.: Possible Measures for Safety Increase of ŽSR Level Crossing. In: 10. World Level Crossing Symposium Safety and Trespass Prevention. Paríž, 24. – 26. 6. 2008.

[6] KMEŤ, V., RÁSTOČNÝ, K.: Guarantee of Constant Train Approach Warning Time at Level Crossing System. In: Archives of transport system telematics, Vol.2, Issue 4, november 2009, pp. 7-12, ISSN 1899-8208.

[7] RÁSTOČNÝ, K., ZAHRADNÍK, J., JANOTA, A.: Particularities of level crossing installations at the Slovak railways. In: Proc. of the 1th Workshop of 6FP/ SELCAT, INRETS. pages. 29-34, Lille, Francúzsko, 16. 5. 2007. ISBN 978-2-85782-663-7; ISSN 0769-0266.

[8] RÁSTOČNÝ, K.., et al.: Prvky zabezpečovacích systémov. EDIS – vydavateľstvo ŽU, 2012, ISBN 978-80-554-0593-3.

Abstract

Railway transport is one of processes controlled with a certain level of risk. It is apparent that due to level of our knowledge, technical level and limited financial means we cannot calculate for absolute safety (zero risk) but we must admit that in a real technical system, some error or fault may occur and its occurrence may mean a certain risk for the controlled process. Main attention of authors is paid to proposing and presenting potentially usable measures that could increase safety of traffic operation at the level crossings LCs operated by the ŽSR (Slovak Railways). Separately there are discussed technical and organizational measures. Some of proposed measures are specific for Slovak conditions only, however to a certain extent some findings can be generalized and possibly applied in other countries, too.

prof. Ing. Karol Rástočný, PhD.

Ing. Peter Nagy, PhD.

Žilinská univerzita v Žiline
Fakulta elektrotechniky a informačných technológií
Katedra riadiacich a informačných systémov
Univerzitná 8215/1
010 26 Žilina
Tel.: +421 41 513 3320, 3357
E-mail: karol.rastocny@fel.uniza.sk
peter.nagy@fel.uniza.sk

Archív

Archive

2001 AT&P journal PLUS 1: Adaptívne a nelineárne riadenie systémov (tlačená verzia)
Adaptive and nonlinear control systems (printed version)

AT&P journal PLUS 2: Robotika, mechatronika, diskrétne výrobné systémy (tlačená verzia)
Robotics, mechatronics, discrete manufacturing systems (printed version)

2002 AT&P journal PLUS 3: Robustné systémy riadenia (tlačená verzia)
Robust control systems (printed version)

2003 AT&P journal PLUS 4: Samonastavujúce sa systémy v riadení procesov (tlačená verzia)
Selftuning systems in process control (printed version)

2004 AT&P journal PLUS 5: Robotické systémy (elektronická – CD verzia)
Robotics systems (electronic – CD version)

2005 AT&P journal PLUS 6: Mechatronika (elektronická – CD verzia)
Mechatronics (electronic – CD version)

AT&P journal PLUS 7: Umelá inteligencia v praxi (elektronická – CD verzia)
Artifical intelligence in Practise (electronic – CD version)

2006 AT&P journal PLUS 1: Mechatronické systémy (elektronická – CD verzia)
Mechatronic systems (electronic – CD version)

AT&P journal PLUS 2: Inteligentné meracie systémy (elektronická – CD verzia)
Intelligent measurement systems (electronic – CD version)

2007 AT&P journal PLUS 1: MMaMS'2007 (elektronická – CD verzia)
MMaMS'2007 (electronic – CD version)

AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process Control (electronic – CD version)

2008 AT&P journal PLUS 1: Mobilné robotické systémy (elektronická – CD verzia)
Mobile robotic systems (electronic – CD version)

AT&P journal PLUS 2: Riadenie v energetike (elektronická – CD verzia)
Control of Power Systems (electronic – CD version)

2009 AT&P journal PLUS 1: Inteligentné pohybové systémy (elektronická – CD verzia)
Intelligent motion control systems (electronic – CD version)

AT&P journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process control (electronic – CD version)

2010 AT&P journal PLUS 1: Systémy automatického riadenia (elektronická – CD verzia)
Systems of automatic control (electronic – CD version)

AT&P journal PLUS 2: Robotika vo vzdelávaní (elektronická – CD verzia)
Robotics in education (electronic – CD version)

2011 ATP Journal PLUS 1: Systémy automatického riadenia (elektronická – CD verzia)
Systems of automatic control (electronic – CD version)

ATP Journal PLUS 2: Riadenie procesov (elektronická – CD verzia)
Process Control (electronic – CD version)

2012 ATP Journal PLUS 1: Modelovanie mechanických a mechatronických sústav (elektronická – CD verzia)
Modelling of Mechanical and Mechatronic Systems (electronic – CD version)

2013 ATP Journal PLUS 1: Robtep 2012 (elektronická – CD verzia)
Robtep 2012 (electronic – CD version)

ATP Journal PLUS 2: Riadenie dopravných a priemyselných procesov (elektronická – CD verzia)
Control of transport and industrial processes (electronic – CD version)